N94-18351

Low Power Signal Processing Research at Stanford

J. Burr, P.R. Williamson and A. Peterson

Space, Telecommunications, and Radioscience Laboratory
Department of Electrical Engineering
Stanford University
Stanford, Ca. 94305
burr@mojave.stanford.edu

Abstract - **This paper gives an overview of the research being conducted at Stanford University's Space, Telecommunications, and Radioscience Laboratory in the area of low energy computation. It discusses the work we are doing in large scale digital VLSI neural networks, interleaved processor and pipelined memory architectures, energy estimation and optimization, multichip module packaging, and low voltage digital logic.**

1 Introduction

Our research in low energy computation for signal processing is being supported in large part by NASA. The neural network research is being funded by the Center for Aeronautics and Space Information Sciences (CASIS). Low energy computing research is being funded by NASA grant NAGW1910, "Low power signal processing technology for space flight applications".

2 Overall motivation

Our research in low energy computing is driven by the need to maximize computation rates in power constrained environments. Space based data systems and large scale neural networks both require low energy per operation; in flight systems, to minimize power consumption during data gathering, processing, storage, and communication; in neural networks, to achieve the necessary computation rates within manageable power budgets. These systems are characterized by high sustained levels of computational effort, unlike typical portable computer applications, which tend to have bursty, and much more modest, information processing requirements.

3 4:2 adder based architectures

We have been building deeply pipelined, parallel signal processors since 1985 [17,18,3,2,19]. We came up with a multiplier architecture which struck a balance between throughput, latch overhead, and regularity [18]. The multiplier consists of a tree of "4:2 adders" (see Fig

4.2.2

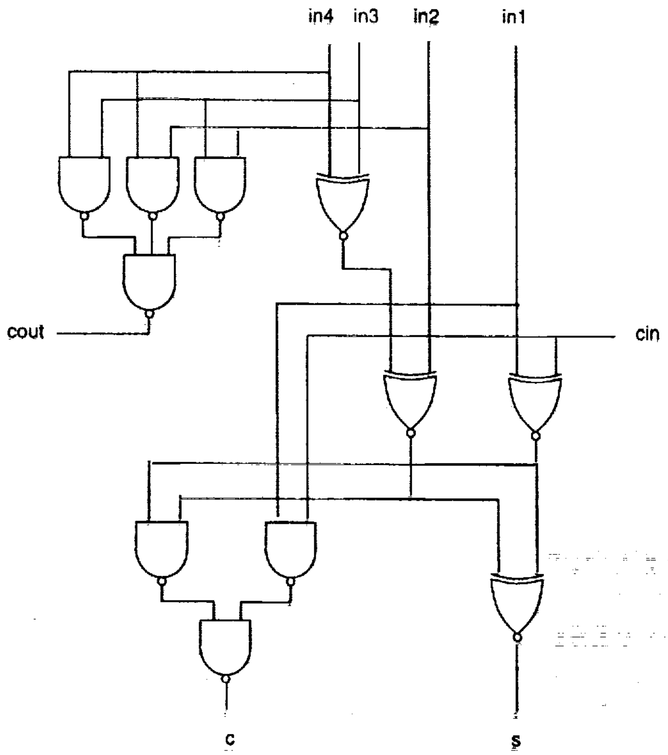

Figure 1: 4:2 adder. The critical path contains three xors in series.

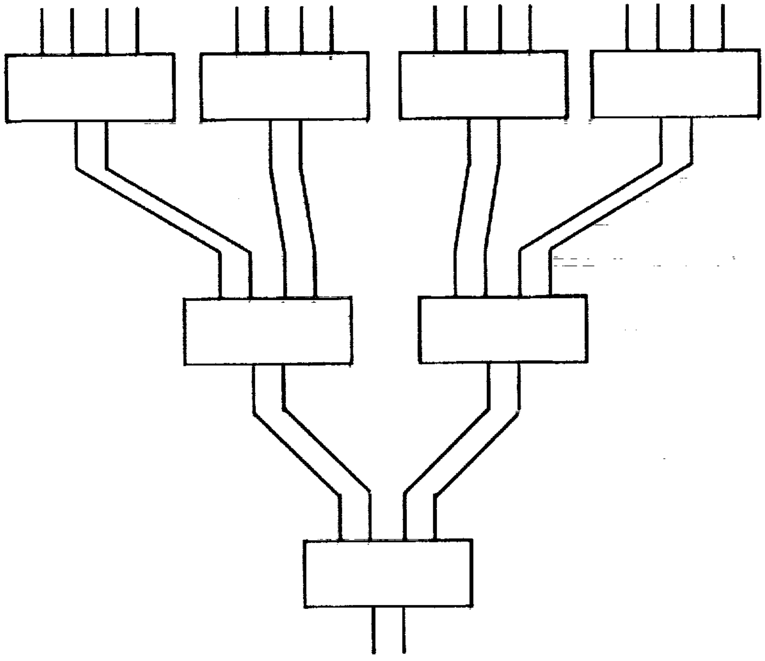

Figure 2: 4:2 multiplier. N partial products are reduced to 2 in $\log_2(N)/2$ stages of 4:2 adders.

2). A 4:2 adder (see Fig 1) has 4 inputs, a carry in, and generates two outputs and a carry out. The carry out does not depend on the carry in. The 4:2 adder can be implemented using two full adders, but a direct logic implementation can reduce the critical path from 4 xors in series to three. A multiplier built out of a tree of 4:2 adders has a much more regular structure than a Wallace tree [1], which uses a full adder to reduce three partial products to two at each stage. The 4:2 tree reduces 4 partial products to two at each stage, and has the self-similarity of a binary tree. A 4:2 adder can efficiently accumulate successive products in carry-save form. It can also be used in an ALU to perform arithmetic operations in time independent of the number of bits in the operands.

We have recently shown that power is minimized in a parallel multiplier when $ld = 11$ [10]. A 4:2 adder has a logic depth of 10, including latches. By comparison, RISC microprocessors typically have logic depths around 40.

We currently have a number of projects which are implementing architectures based on the latency in a 4:2 adder. We were becoming concerned about the feasibility of running systems at the clock rates implied by a logic depth of 10: in 0.8 micron CMOS, a 4:2 adder based clock generator circuit runs at 400MHz [20]. However, similar speeds have been reported elsewhere [21]. Recently, with the opportunity presented by tiled architectures and 3D multichip modules as discussed in [10,27], it appears that deeply pipelined architectures can also achieve good performance at very low energy.

4 Neural Nets

Large scale neural nets will require on the order of 10^{15} connections per second (CPS) [9]. Digital VLSI neurochips reported so far require around 1nJ per synaptic connection [22]; 10^{15} CPS would require a megawatt! Biological neurons require around 1fJ per synaptic connection, 6 orders of magnitude less. Attaining biological energy efficiency in silicon is a formidable challenge. We have identified a number of factors which together may reduce connection energy by 5 orders of magnitude to 10fJ per connection, permitting 10^{15} CPS at around 10 watts. These include: reduced arithmetic precision (10x), reduced feature size (10x), and low voltage operation (1000x).

In addition to investigating performance of large networks, we are implementing a digital Boltzmann machine [22] to demonstrate the viability of reduced precision, pipelined digital learning machines. The chip is being implemented in 2.0u CMOS, and consists of 32 5-bit neural processors, each supporting 1K 5-bit weights and capable of 80MHz operation. The chip will be capable of 2.5 billion connections per second, and 320 million connection updates per second.

5 Pipelined Memory

We are implementing a pipelined memory architecture (see Fig 3) which achieves high throughput by recursively subdividing the memory array into sections which can be traversed in a single cycle. Addresses are partially decoded in each section. The remaining

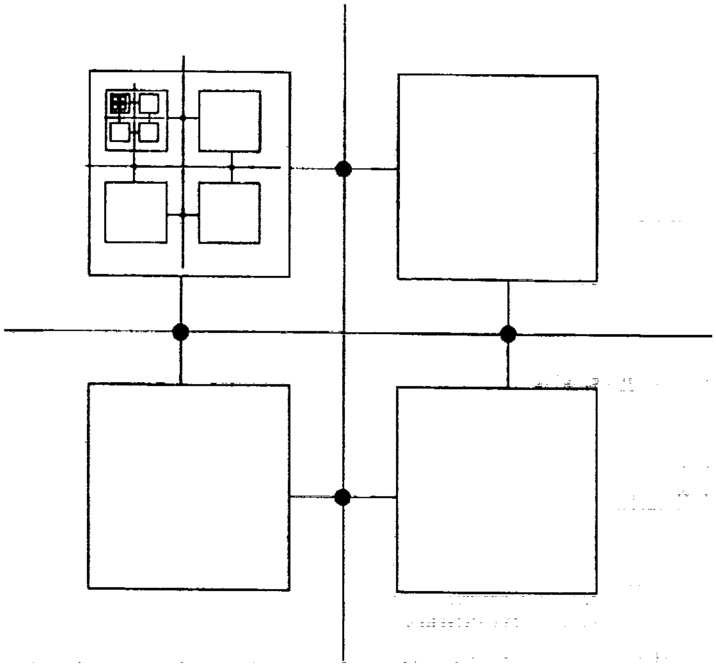

Figure 3: Pipelined memory

address bits are routed to the appropriate subsection where additional bits are decoded. At the lowest level, the remaining bits are decoded and data is read out of or written into a memory block. For read operations, the data is delivered back up through the subsections on subsequent cycles until it emerges at the pads. For write operations, the data accompanies the address down the tree.

The size of the memory block is matched to the propagation delay through a 4:2 adder. This turns out to be about 32 words x 32 bits. We have written an optimizer which sizes the transistors in this block for the minimum area and power that matches the delay [25]. We pipeline the address decode and data return, placing pipestages to minimize power dissipation. Power dissipation in the memory is greatly reduced by selectively clocking the portion of the memory which contains the data, leaving the rest of the system on standby.

Hierarchical memory organization first appeared in Mead and Conway [12,11], but this architecture was not pipelined. An unpipelined binary tree memory was described at the 1987 International Test Conference [8,26]. Hierarchical address decoding was reported in a 4Mb SRAM with selective enable to reduce power dissipation [7].

A pipelined memory architecture was discussed in [28]. The CT7C158 is a pipelined 64K SRAM offered by Cypress Semiconductor, who say: "Pipelined RAMs are used in writeable control store, DSP and logic analyzer/tester applications where throughput is the critical parameter."

Our pipelined memory is the first to combine hierarchical address decoding and selective clocking to maintain very high throughputs and very low power dissipation.

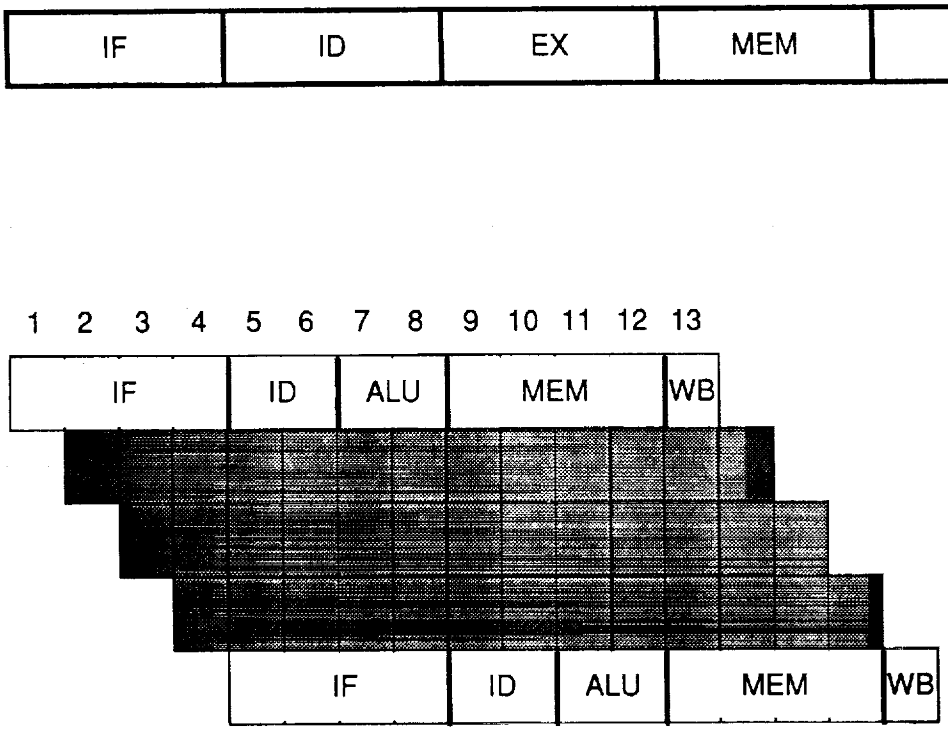

Figure 4: Interleaved processor pipeline, with a normal RISC pipeline for comparison. In this example, instruction fetches and memory accesses take four cycles. There are four independent instruction streams in various passes of execution.

6 Interleaved Processor

We are working on a processor architecture which achieves high performance by interleaving independent instruction streams on a deeply pipelined processor (see Fig 4). The number of independent streams is matched to the latency in the pipelined memory. The clock frequency is a multiple of a RISC clock, and is obtained by placing extra pipestages at critical points in a RISC architecture. The number of extra pipestages is smaller than expected because many of the normal RISC stages do not use up an entire clock cycle. Our objective is to achieve a 4x speedup over RISC in a given technology, and to implement a subset of the MIPS R3000 instruction set. We are experimenting with a variety of power reduction techniques at the circuit and system level in the processor design.

Multiple instruction stream processors have been built before (Burton Smith's work on HEP, Horizon, and Tera [16,15]), but only in the context of large supercomputers and not single integrated circuits, and not matched to the latency of a pipelined memory. Edward Lee at UC Berkeley proposed an interleaved architecture for use in signal processing [14], but his design is not pipelined as deeply as ours, and does not include pipelined memory. The only reference we have found so far which describes an interleaved processor and a pipelined memory is a Japanese paper on gate-level pipelined Josephson Junction circuits [28], which also describes a method to increase the throughput of CMOS memory by pipelining, but the two concepts are not synergized, and the memory organization is not discussed. Stone and Cocke say "some combination of long pipelines and multiple interleaved instruction streams may eventually prove effective for combining high speed and high efficiency" but give no details [23].

The RISC community is also investigating techniques for increasing performance. The two chief techniques are superscalar and superpipelined architectures [5]. In superscalar, more than one instruction may be in progress at a given time. In superpipelined, the RISC pipe is broken into a number of smaller stages with reduced logic depth. Both of these approaches result in added control complexity managing the potential hazards and resource conflicts which may result.

Superscalar machines, such as the Intel I860, fetch more than one instruction on each cycle, and execute in parallel whenever possible. There are restrictions in the combination of instructions which can be issued simultaneously. Superscalar increases resource utilization but does not increase the throughput of a given functional unit.

We reduce RISC logic depth by a factor of 4, and introduce 4 independent, interleaved instruction streams. The streams are kept independent to avoid the hardware complexities associated with managing a highly pipelined single thread of control. Each instruction stream executes its next instruction every fourth cycle. The control complexity is no worse than for a RISC machine but the throughput is 4 times greater on problems that can be parallelized. Fortunately, these are commonplace in signal processing. The architecture also supports zero-overhead context switching of up to 4 processes. This is very useful in embedded real time control applications.

6.1 Timing

Real time signal processing tasks often require "precise" timing. This is not easy in cache-based architectures, since cache miss recovery times can often be data dependent. The pipelined memory/interleaved processor behavior is precise: instruction latencies are fixed. Memory fetches always takes 4 cycles. There are never any cache misses. Branch timing is precise.

In a conventional RISC machine, the latency that takes place during a branch is unpredictable, because it depends on whether the target address is in the instruction cache, and if so, how it is aligned within the cache entry that contains it. Given a 4 cycle latency to fill a line in the cache, and a cache linewidth of 4 words, a branch target will only point to the first word in the line 25% of the time. The system must stall fetching the line following the line containing the branch target address. The AMD29000 "branch target cache" solves this problem by aligning cache lines to branch targets. This increases the complexity of the memory subsystem. The interleaved processor solves this problem by maintaining a fixed latency on every instruction fetch.

6.2 Energy

Our objective has been to maximize overall performance. With the advent of Multichip module technology, the performance of an individual chip must be considered in light of the system. We now are designing to maximize performance at minimum energy. The best way to do this is to obtain the maximum possible throughput, and use the performance margin to lower the supply voltage until all the available area is used and the power budget is met.

The clock frequency can be increased by a factor of 4, so that each stream can execute as fast as a RISC processor in the same technology, and the processor can achieve 4 times RISC performance. This implies 400MHz in 0.8 CMOS. Although this is feasible for small numbers of processors, we plan instead to lower voltage by a factor of 4, to 1.25V. This will give us the same 2D performance density as a RISC machine, but will require only 1/16 the energy per operation and 1/64 the power. We can capitalize on MCM technology to achieve 64 times the performance with 64 times the area for the same power budget.

Also, because resources are pipelined, more time is available to wake up an idle resource or put it on standby. Resources only need to be clocked if they are being used. If a resource is used by one stream, but not by the next, the inputs to that resource can retain their previous values.

Register files normally consume a significant portion of the power budget. Since each stream has its own register file, the access rate to a register file can be 1/4 the system clock frequency. Conventional SRAM is faster and lower power than multiported register files since the bitlines never have to swing more than 100mV for reading or writing. If the SRAM can be accessed in a single cycle, it can emulate a 4-port memory which can support any combination of up to 4 reads or writes every 4 cycles. In its standard configuration it would be accessed sequentially to fetch two operands and write back a third. Whether this results in less energy depends on how often operand addresses are repeated on successive instructions.

6.3 Area

The interleaved processor should require area comparable to a RISC processor because four sets of registers, program counters, and other state registers take no more area than on-chip instruction and data caches.

7 Multichip Modules

Multichip module packaging provides a number of significant new opportunities in system architecture and implementation. Bare die can be placed much closer together than packaged parts, leading to shorter wires and reduced communication energy. Area bonding reduces lead inductance, permitting higher frequency interchip communication. Small bonding pads and high connective capacity support seamless interchip communications optimized for propagating signals a few centimeters. Intrinsic bypass capacitance due to thin dielectric separation of Vdd and Gnd planes results in higher noise immunity.

The net result is the opportunity to reduce communication energy and increase system level performance by orders of magnitude compared to conventional packaging techniques. We are developing interconnect structures, data transmission circuits, and clock distribution structures for high performance (hundreds of MHz), low power (tens of mW) MCM systems. Much of our work in this area has been reported in [24].

We have designed a test module which is being fabricated by ATT. It includes passive structures for measuring capacitance, crosstalk, and characteristic impedance of a variety of conductor geometries. It also has two sites for MOSIS TinyChips which will test the interconnect by exchanging pseudorandom bitstreams through single ended and differential transceivers at data rates in excess of 200 MHz.

7.1 Tiled architectures for signal processing

The opportunity exists to extend the concept of regularity and locality so widely used in VLSI design to the multichip module level, and to identify a set of processor tiles which can tessellate the plane to generate massively parallel architectures. We are investigating a variety of "tiled" architecture opportunities. We have extended our neural net Boltzmann machine architecture to accommodate an arbitrarily large two dimensional array of chips.

8 Multiprocessing

The interleaved processor is inherently a symmetric shared memory multiprocessor. Memory consistency is guaranteed because there is no cache. We are investigating ways to interconnect interleaved processors for massively parallel multiprocessing.

8.1 Hierarchical pipelined ringbus

One possible organization of a massively parallel system is a "hierarchical ring bus" architecture which supports high bandwidth pipelined data exchange among multiple processors. The overall topology consists of rings of processors connected by gateways. Each local ring can sustain data transfers at the processor clock rate. Because the bus itself is pipelined, multiple transactions can be in progress concurrently, up to the number of processors in the ring. One of the nodes in the ring can be a gateway to another ring and can sustain the same I/O bandwidth. We plan to match the bus clock frequency to the latency of a 4:2 adder.

This architecture has been proposed elsewhere [15]. We think it is well matched to the performance and latency of the interleaved processors and multichip module based multiprocessors. In the spirit of interleaved instruction streams, the latency to complete a single bus transaction will be at least equal to the number of processors in the ring, but a separate bus transaction can be in progress simultaneously on each segment of the ring. This will result in substantially higher throughput than conventional bus architectures - in excess of 1 Gbyte/sec. This architecture is well suited to datastream oriented algorithms common in real time signal processing.

Although this approach introduces single point failures at each node in the ring, when placed in the context of 3D multichip module implementation we think the approach has some significant advantages.

The ringbus concept can be extended gracefully to large numbers of processors by recursively adding subrings connected by gateways. We will be analyzing the implementation complexity, energy, and performance of this approach in comparison to other processor communication networks.

Of key interest is mapping numerically intensive signal processing problems onto this architecture. A 1024 processor system might consist of 64 rings with 16 processors in each ring. At 400 MIPs per node and 1 Gbyte/sec per ring, total performance would be 400GIPS; total throughput would be 64 Gbytes/sec. Ring size can be optimized to balance instruction and communication bandwidth.

9 Energy estimation and optimization

We estimate energy using

$$\begin{aligned} E_{ac} &= \frac{1}{2}aCV^2 \\ E_{dc} &= I_{dc}V/f \end{aligned}$$

where a is the activity ratio, the fraction of transistors switching on each cycle, C is the capacitance being switched, V is the supply voltage, I_{dc} is the DC current, and f is the clock frequency.

This technique relies on short circuit current being a small fraction of the total.

We are investigating techniques for minimizing power dissipation by minimizing transistor sizes while minimizing short circuit current. These are conflicting constraints, and can lead to substantial power reductions over techniques which ignore short circuit current and assume minimum size devices result in minimum power.

We have modified our timing simulator to measure AC power dissipation by accumulating dumped charge. Preliminary results suggest good agreement with power measurements on fabricated chips. We are extending this technique to measure peak power. We have developed a memory block optimizer which sizes transistors in the pipelined memory to maximize a "merit" function which is a weighted combination of performance, power, and area. We are including the effects of short circuit current on both our transistor sizer and our memory block optimizer.

We have found that transistor sizing is important in optimizing highly pipelined designs. Balancing clock delays is especially important to minimize clock skew in the system. Transistors can also be sized to minimize energy, which involves balancing short circuit current against gate capacitance.

10 Low Voltage Digital Logic

Massively parallel architectures tiled on 3D stacked multichip modules can quickly exceed the ability to extract heat from the structure. Reducing the supply voltage promises substantial reductions in energy and power; we are investigating the practical limits to low voltage operation. This area is covered in depth in [10].

Our approach to low energy computation has attracted interest from a number of sources. More detailed investigation into the opportunity is being funded as a "Research thrust" by Stanford's Center for Integrated Systems. These research thrusts involve interaction with technical liaisons from CIS industrial partners. So far, the Ultra Low Power thrust has liaisons at DEC, GE, IBM, Intel, National Semiconductor, and TI.

11 Personnel

Who the group is:

Professor Allen M. Peterson, Principal Investigator

P. Roger Williamson, Senior Research Associate

James B. Burr, Senior Research Engineer

Low Energy Computing

Bevan Baas	computer architecture
Jim Burnham	high speed interconnect
Ely Tsern	interleaved algorithms
Gerard Yeh	Low energy VLSI circuits
Sabeer Bhatia	Low energy process design

Neural Networks

Kan Boonyanit	Approximate Gradient Descent
Karen Huyser	Wafer Defect Classification
Michael Leung	Texture Recognition
Michael Murray	Precision, Learning, and VLSI

Collaboration

ATT,	multichip modules
Sun,	energy optimization
Intel,	digital neural network architectures
Ricoh,	neural net coprocessors

12 Conclusion

Our research in low energy computation has been motivated by recent trends in VLSI technology, multichip module packaging, and application architectures. We believe the opportunity exists to achieve very high computation rates in power constrained environments by reducing decision, storage, and communication energy.

13 Acknowledgements

This research was supported in part by NASA grants NAGW1910 and NAGW419, by a gift from Intel Corporation, and by a grant from Stanford's Center for Integrated Systems. Multichip modules were provided by ATT, workstations by Sun Microsystems, and VLSI fabrication by MOSIS.

References

[1] Shlomo Waser and Michael J. Flynn, *Introduction to Arithmetic for Digital Systems Designers*, CBS College Publishing, 1982.

[2] Weiping Li and James B. Burr and Allen M. Peterson, " A fully parallel VLSI implementation of distributed arithmetic", *IEEE International Symposium on Circuits and Systems*,June, 1988, 1511-1515.

[3] Weiping Li, " The Block Z transform and applications to digital signal processing using distributed arithmetic and the Modified Fermat Number transform", 1988.

[4] Weiping Li and James B. Burr, " An 80 MHz Multiply Accumulator", PhD thesis, Stanford University, 1987.

[5] John L. Hennessy and Norman F. Jouppi, " Computer technology and architecture: An evolving interaction ", *IEEE Computer Magazine*, 9, 1991, 18-29.

[6] James B. Burr and James R. Burnham and Allen M. Peterson, " System-wide energy optimization in the MCM environment ", *IEEE Multichip Module Workshop*, 1991, 66-83.

[7] Toshihiko Hirose, Hirotada Kuriyama, Shuji Murakami, Kojiro Yuzuriha, Takao Mukai, Kazuhito Tsutsumi, Yasumasa Nishimura, Yoshio Kohno and Kenji Anami, " A 20ns 4Mb CMOS SRAM with hierarchical word decoding architecture ", *IEEE International Symposium on Circuits and Systems*, 1990, 132-133.

[8] Najmi T. Jarwala and D. E Pradhan, " An easily testable architecture for multimegabit RAMs ", *IEEE Test Conference*, 1987, 750-758.

[9] Carol Weiszmann, " DARPA Neural Network Study ", October 1987 - February 1988, *AFCEA International Press*, 1988.

[10] James B. Burr and Allen M. Peterson, " Ultra Low Power CMOS Technology ", *NASA VLSI Design Symposium*, 1991.

[11] Carver Mead and Lynn Conway, *Introduction to VLSI Systems*, Addison-Wesley, 1980.

[12] Carver A. Mead and Martin Rem, " Cost and performance of VLSI computing structures ", *IEEE Transactions of Electron Devices*, April, 1979, 533-540.

[13] Kentaro Shimizu, Eiichi Goto and Shuichi Ichikawa, " CPC (Cyclic Pipeline Computer) - an architecture suited for Josephson and Pipelined-Memory machines ", *IEEE Transactions on Computers*, Volume 38, Number 6, June, 1989, 825-832.

[14] Edward A. Lee and David G. Messerschmitt, " Pipeline interleaved programmable DSP's: Architecture ", *IEEE Transactions on Acoustics, Speech, and Signal Processing* , Sept, 1987, 1320-1333.

[15] Burton J. Smith, " The Horizon Supercomputer ", *Supercomputing*, Oct, 1988.

[16] Burton J. Smith, " Architecture and applications of the HEP multiprocessor computer system ", *SPIE, Real-Time Signal Processing IV*, 1981, 241-248.

[17] James B. Burr and others, " A 20 MHz Prime Factor DFT Processor ", Stanford University, Sept, 1987.

[18] Weiping Li and James B. Burr, " An 80 MHz Multiply Accumulator ", technical report, Stanford University, Sept, 1987.

[19] Alfred J. Eiblmeier, " A reduced coefficient FFT butterfly processor ", technical report, Staford University, Oct, 1988.

[20] Mark R. Santoro, " Design and Clocking of VLSI Multipliers ", PhD thesis, Sanford University, 1989.

[21] Y. Jiren, I. Karlsson and C. Svensson, " A true single phase clock dynamic CMOS circuit technique ", *IEEE Journal of Solid-State Circuits*, 1987, Volume SC-22, 899-901.

[22] James B. Burr, " Digital Neural Network Implementations ", *Neural Networks: Concepts, Applications, and Implementations, Volume 2*, Prentice Hall, 1991.

[23] Harold S. Stone and John Cocke, " Computer architecture in the 1990s ", *IEEE Computer Magazine*, Sept 1991, 30-38.

[24] James B. Burr, James R. Burnham and Allen M. Peterson, " System-wide energy optimization in the MCM environment ", *IEEE Multichip Module Workshop*, 1991, 66-83.

[25] Bevan Baas, " A pipelined memory system for an interleaved processor", technical report, Stanford University, Sept, 1991.

[26] Dhiraj K. Pradhan and Nirmala R. Kamath, " RTRAM: Reconfigurable and testable multi-bit RAM design ", *IEEE International Test Conference*, 1988, 263-278.

[27] James B. Burr and Allen M. Peterson, " Energy considerations in multichip-module based multiprocessors ", *IEEE International Conference on Computer Design*, 1991.

[28] Kentaro Shimizu, Eiichi Goto and Shuichi Ichikawa, " CPC (Cyclic Pipeline Computer) - an architecture suited for Josephson and Pipelined-Memory machines ", *IEEE Transactions on Computers*, Volume 38, Number 6, June, 1989, 825-832.