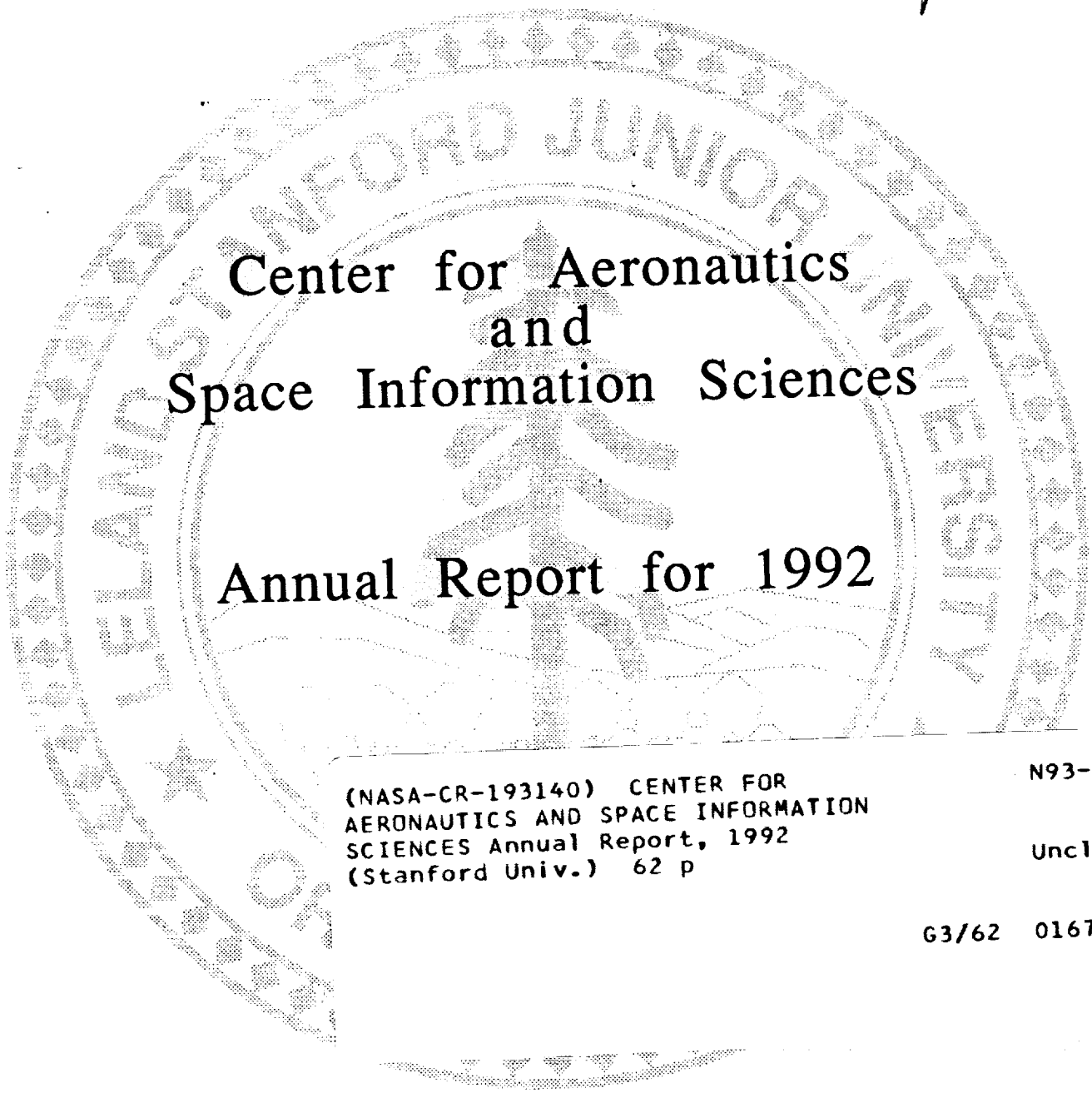
NASA-CR-193140



Center for Aeronautics and Space Information Sciences

Annual Report for 1992

Submitted by:
Michael J. Flynn, P.I.
Department of Electrical Engineering
Stanford University
Stanford, CA 94305-4055

Contents

List of Figures

Overview

This report summarizes the research done during 1991/92 under the CASIS program.

The Center for Aeronautics and Space Information Sciences was established in 1983 as a joint undertaking of NASA's Office of Aeronautics and Space Technology (OAST) and the Stanford School of Engineering. An important objective of this Center is the development of high risk, high payoff technology in areas related to computer and information science. Parallel to this, CASIS has the mission of training graduate students in areas that relate to NASA's long range needs and plans as they apply to new information system technologies.

CASIS is organized into individual research projects led by Stanford faculty in the departments of Electrical Engineering and Computer Science. Currently eight faculty, four postdoctoral staff, and approximately 14 graduate students are actively involved in the CASIS program. Within the Electrical Engineering Department, three different laboratories are involved in CASIS research: the Information Systems Laboratory, the Space, Telecommunications and Radioscience Laboratory, and the Computer Systems Laboratory. Each research team is led by a faculty member, with participation by Ph.D. research staff and graduate students. Each year, the majority of funds provided to CASIS is spent in graduate student support. Since the typical time to receive a Ph.D. degree is on the order of four years, we have found that we are able to produce four or five Ph.D. students each year. Depending on their degree level on entry into CASIS supported research, several (about two) other students are awarded M.S. or Engineer's degrees en route to their Ph.D. program. Owing to close contact with various NASA projects, the Ph.D. graduates often seek employment at NASA field centers or with NASA-supported aerospace contractors.

CASIS also provides a research environment for four to six postdoctoral research scholars. These visitors come from either universities or industry, both domestic and foreign. Of the university post-docs, usually from one to two receive partial support from CASIS. Industrial visitors are supported by their sponsoring companies. It is a long-term goal of CASIS to include visiting NASA personnel in this program.

Independent of CASIS, Stanford has an active instructional television program offering M.S. degrees in Electrical Engineering and Computer Science. There are direct TV links to NASA–Ames under this program, facilitating closer cooperation between CASIS facility and research programs and some NASA–Ames activities.

Each year, CASIS sponsors an annual meeting designed to promote the exchange of information and ideas between technical personnel from NASA field centers and CASIS personnel. In addition, attempts are made to draw other Stanford experts into the annual meeting so that the NASA attendees can have contact with a broad range of research programs. We plan to hold our Annual Review in May, 1993 at Stanford University.

Chapter 1

Computer Architecture

The computer architecture research under CASIS has four distinct themes:

1. The development of tools to evaluate the performance of high-performance computing systems.
2. The optical interconection of multiprocessors.
3. Basic hardware studies.
4. Software to support the efficient use of shared-memory multiprocessors.

Our first area concentrates on very high-level design tools and aids for systems designers and evaluators. We continue to develop our Computer Architect's Workbench and make it available to industry and other educational institutions. In a separate effort, we have developed tools to evaluate the performance of I/O subsystems. These tools, called *TIME*, include TLB simulators for I/O caches and disk subsystem simulators. During 1991/92 we have completed the prototype version of TIME, and are studying various disk array configurations using these tools.

In the area of optical interconnections for multiprocessors, we have developed a systems approach using multifaceted holograms to route messages among n processing elements. The communications medium is free-space optics, with each hologram facet providing a coupling between a source and a destination. During 1992 we have completed a feasibility study of this technology.

We have also continued some basic hardware studies. The first study was to find alternate hardware models that will support neural network-type pattern recognition applications. We particularly have looked at sparse adaptive memory, a version of sparse distributed memory originally developed at RIACS. The appeal of these approaches is that they can represent economical hardware realizations using conventional DRAM technology.

The other study in hardware used analog VLSI technology to study the possibility of reproducing the electronic equivalent of the biological cochlea in an ear. This study is now complete and provided interesting insight into the nature of redundancy in biological circuitry. Using similar approaches, it may be possible to perform computations (of at least certain types) in the presence of multiple faults.

The fourth area in architecture focused on compiler optimization of scientific code to improve the performance of large shared-memory multiprocessors. The key problem in moving scientific code to such organizations involves the blocking of data so that it fits properly in the processor cache. Scientific code frequently is ill-suited towards multiprocessors because of the presence of very large data structures that are accessed with a regular stride. Such access patterns have little spatial locality, and if the structures are large enough, insufficient temporal locality to fit in all but the very largest data cache. Thus, the processors are forced to run as if the caches were disabled. Recent work here indicates that it is possible to restructure accesses to such code so that the data structures are blocked into smaller pieces that properly fit in data cache.

1.1 The Computer Architect's Workbench

Brian Bray

Evaluating the usefulness of architectural/design decisions is complicated by the interdependence of architectural features, various levels of compiler sophistication, and the validity of benchmarks. At Stanford University, the Computer Architect's Workbench (AWB) has been developed to aid in evaluating designs. The Architect's Workbench is a set of tools to predict the relative performance of alternative computer and system architecture features. The Architect's Workbench runs actual programs for various hardware configurations.

The Architect's Workbench predicts the relative performance of alternative computer and system architectures. The system allows high-level tradeoff, including instruction format selection, instruction encoding, register set size and organization, and cache size and organization. The Compiler tools consists of compiler front ends, code optimizers, and register allocators. The remainder of the tools are used to simulate the architectures being studied. These consist of programs to do static analysis of the instruction and data stream and programs to simulate the memory hierarchy, i.e., instruction and data caches. Since simulation speed is crucial to examining many alternatives, the tools rely heavily on static analysis of the application to reduce simulation time.

As the issue rate of processors increases (from superscalar and/or superpipeline organizations), the demand on the memory system also increases. We have been extending the Architect's Workbench by developing new tools to further analyze the nature of access and use of memory. In this last year, with the use of these tools we have made several discoveries that can help optimize pipelined processors for memory access management.

- We have developed a two-level windowed register file to reduce the cycle time of to that of a smaller windowed register file while retaining the performance of a large windowed register file [2].

- We have developed a tag cache to eliminate the need to check the data cache before a perform write [4].

- We have developed subword caching to eliminate the need for subword access hardware being in the primary cache access path [3].

- We have developed translation hint buffers to eliminate the latency of address translation for a physically addressed instruction cache [5], [6].

- We have continued to explored page coloring; a simple modification to an operating system's page allocation algorithm to give physically addressed caches the access speed of virtually addressed caches [13].

In addition to our own research, we have continued to distribute the Architect's Workbench to other universities. We also have ported the AWB to the Sun4.

1.2 Input/Output and Memory Evaluation

Kathy Richardson

Disk I/O speed (latency and bandwidth) has and is expected to increase at a much slower rate than CPU speed and memory density. The gap in performance between the two continues to widen despite research and advancements in disk technology. I/O is emerging as a problem for all computer systems and application environments.

To compensate for the gap, we must fully understand the interaction between CPU, memory, and disk system, and make architectural and system enhancements accordingly. In order to do this, we have developed a set of tools (called *TIME*) that are designed to evaluate this interaction and to determine the role of new technologies in the storage system.

TIME is being used to investigate many issues involving I/O and its relationship to the memory system. The tools include a TLB simulator, virtual to real address translation via page tables, main memory allocation, I/O caches, and a disk system simulator.

The TIME tools mentioned are complete and are now being used to investigate I/O behavior. TIME is trace driven and designed to study real workloads rather than model statistical behavior. We have produced I/O traces from several workstation environments, using a modified operating system.

Progress

I/O caches (disk caches) are an important part of many I/O hierarchies. They are also useful for studying the I/O behavior of applications. The I/O cache simulator in TIME is suited for examining the referencing behavior of applications. It can keep track of the type of data requested, request sizes, and hit rates for each type. Request types can be selectively cached to evaluate interactions between requests in the cache, and to isolate and evaluate the spatial and temporal reference behavior.

Knowing how different types of data use an I/O cache can increase the effectiveness of the cache. Information from the system call level determines the type of requests. Since the trace is based on file names rather than disk block numbers, the filesystem maintenance, executable, and application data are all visible.

Each data type has different reference behavior. Each cache size supports different types to differing degrees. Properly exploiting these properties increases the reference hit rate in the I/O cache and reduces the number of references out of the cache to disk.

I/O caches reduce the I/O latency by eliminating disk accesses. Conventional approaches reduce latency by increasing the cache hit rate. But caches with the same hit rate, and the same disk, can have different miss latencies and different disk load characteristics. An alternate way to reduce the latency is to increase the efficiency of the load offered to the disk. The disk requests from both non-volatile and volatile cache configurations are evaluated. Neither configuration efficiently utilizes the cache to offer a low latency load to the disk. Ways to improve their load characteristics have been explored. The policies implemented in the cache impact the number of total disk requests as well as the time distribution of these requests. Figure 1.1 shows how typical management policies for non-volatile and volatile cache configurations affect the number of disk requests.

1.3 Sparse Distributed Memory

Brian Flachs

There are two approaches to pattern classification and automated control. Preprogrammed solutions like expert systems apply rules developed by humans to perform their tasks. Unfortunately, there many problems where no optimal algorithm or even reasonably good strategies are known. In these cases an adaptive system must be used. Adaptive systems attempt to organize themselves to develop an effective input to output relationship. Some examples of adaptive systems are high speed modems and some of the latest engine control computers. There are many different approaches to the problem. The neural network is one very common solution that can often perform well. Unfortunately, experience and trial-and-error are the only guides for architecting the network's levels, nodes, and connectivity. Information in the neural network is stored in the form of incomprehensible connection

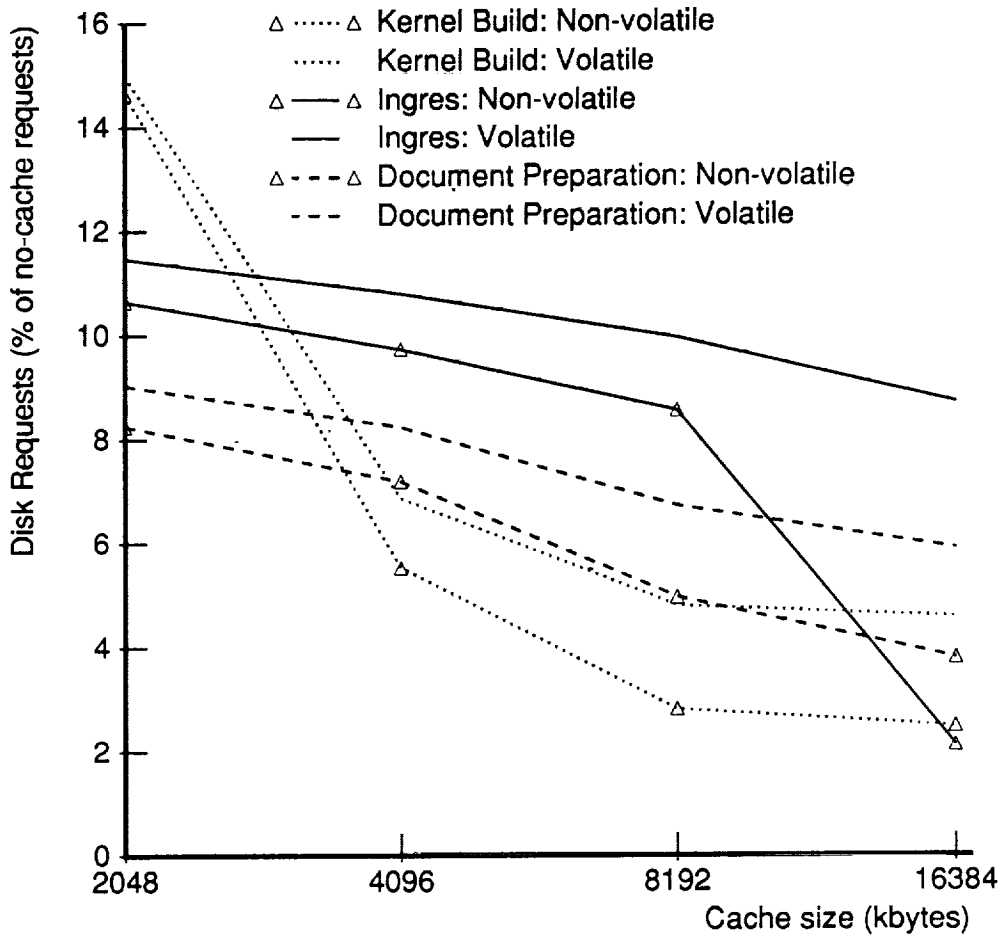


Figure 1.1: Disk requests generated by non-volatile and volatile caches, expressed as a percent of the original (or no-cache) requests.

strengths. Associative memory is another technique for solving these sorts of problems. Here the idea is to keep an explicit record of the training experiences. Unfortunately, this record can become unmanageably large and long periods of time may be required find the relevant experience.

Sparse Distributed Memory (SDM) is a memory-based approach to pattern recognition that incorporates the positive aspects of neural network and associative memory methods. On the one hand, there is a fixed recipe for SDM architecture. The designer decides how large to build the memory. On the other hand, SDM does not maintain a complete record of the training data.

Sparse Distributed Memory is a generalization of an associative memory. Like an associative memory, SDM has a list of vectors chosen from the input vector space. The vectors in this list are physically implemented address vectors. Unlike most associative memories, SDM does not require that the input address vector match one physically implemented vector exactly. It activates a set of physical addresses that are "nearby" the input vector. The original formulation's vectors were long bit vectors which were compared by Hamming distance although the vector space has been generalized to vectors with real components and a range of different distance metrics. Each physically implemented address vector has a corresponding data vector. Read and Write operations manipulate data vectors corresponding to physically implemented addresses in the active set. Read operations vector sum those data vectors to form a result, while write operations add the supplied data vector into each of the active data vectors.

This effort began at Stanford by Pentti Kanerva (now at NASA–RIACS), who defined SDM [9]. This work continues at Stanford University and the Research Institute for Advanced Computer Science in two areas: investigation of Sparse Distributed Memory properties and implementation of a prototype system. The construction of the Stanford Prototype has been completed [7].

Progress

Inappropriate use of a Sparse Distributed Memory produces classifiers with very high error rate and limited capacity. The generic read and write operations are being explored in hopes of discovering SDM's basic mechanism. During this past year, effort has concentrated on the application of gradient descent optimization to SDM creating a new adaptive architecture named "Sparse Adaptive Memory" [8]. The fundamental adaptive algorithm has asymptotic properties compatible with the optimal algorithm. This means although there are sources of local minima, a sparse adaptive memory with an appropriate learning rate will approximate optimal decision boundaries. These results also indicate how the minimum size of SDM is related to the inherent complexity of the recognition environment and hold out the hope that very small memories with on the order of 10 or 100 physically implemented addresses will be capable of significant recognition.

Studies performed to better understand some parameters of the learning environment concluded that:

- A two dimensional, two class, synthesized pattern recognition problem allows changes in the decision surfaces to be visualized.
- Inverted pendulum stabilization is a classic problem in adaptive control and SDM is able to learn by trial and error to stabilize a linearized model of the pendulum.

Learning heuristics are under development to address some of the local minima and introduce a fast coarse learning mode related to the k nearest neighbor learning algorithm. This algorithm is based upon the analysis of sparse adaptive memory's gradient descent learning algorithm and its asymptotic properties and functions like a probabilistic Greedy algorithm.

Essentially, these heuristics control a pool recognition resources. Decisions are made

- Add resources to the sparse adaptive memory if patterns are received and performance is particularly poor, or
- Remove resources from the memory if some redundancy is detected or there is low utilization.

This algorithm vastly improves sparse adaptive memories learning time, probability of error, and probability of convergence when the complexity of the problem is close to the representational limit of a given sized sparse adaptive memory. A family of these algorithms will be presented in a CSL technical report "Sparse Adaptive Memory Learning Heuristics".

The Sparse Adaptive Memory is a fertile ground for research. Its recognition properties will be further studied. A whole new class of applicability is emerging with the neural network perception of SAM. Gradient descent and the ability to propagate errors backwards through SAM allows it and perhaps other look-up table based systems to replace multi-layer neural networks. SAM is computationally attractive since each computational element in the feed-forward net must produce a result while only a small portion of SAM is active at any time.

1.4 Studies of Optical Interconnects for Multiprocessors

Timothy M. Pinkston

The GLORI strategy is a multiprocessor interconnect scheme designed to aid the programmer, compiler, and/or run-time scheduler in facilitating locality, e.g., in partitioning, placement, and relocation of data and processes. It also aids the hardware architecture in

performing the functions useful in capturing and exploiting locality, e.g., clustering, combining, and caching. The impetus behind GLORI is to use the interconnect as an additional resource for taking advantage of multiprocessor locality and to use optics to achieve this goal. Among some of GLORI's non-optical features are hierarchical topology, clustering, combining, and minimal delay routing protocols.

Optics is a promising interconnect technology for multiprocessors. The usefulness of optics' many interconnect features are evaluated using GLORI as a framework. In the proposed GLORI organization, M processor-memory elements (PMEs) are clustered together onto shared buses at the local interconnect level, and N/M bus clusters are arranged in a hypercube at the global interconnect level. Reconfiguration of PME connectivity occurs system-wide to allow arbitrary groups of PMEs to be clustered, but the baseline shared bus-hypercube topology is assumed to remain intact. Interconnect links are established by fiber and free-space holographic diffractive optics.

In previous work, we found that cluster affinity among PMEs is maintained over intervals of tens to hundreds of thousands of execution cycles. Interconnect reconfiguration occurring once every millisecond is therefore sufficient to exploit this behavior. Ferroelectric liquid crystal spatial light modulators and experimental photorefractive holographic devices assumed by GLORI can operate well within this switch frequency.

Progress

Recently, we studied the feasibility of a GLORI implementation. A single-pass, static optical geometry composed of elementary guided and free-space holographic optical components was analyzed. A number of issues determine the feasibility of the GLORI system. Some issues indirectly impact feasibility, such as those relating to the holographic beam steering element. Other issues, such as those relating to optical geometry, directly impact how the system size can be feasibly implemented due to fan-out and fan-in limitations.

A multiprocessor system of $N = 256$ processor-memory nodes can be feasibly implemented with a static optical shared bus-hypercube GLORI network. By improving aspects of the optical system which most limit scalability, systems on the order of thousands of nodes can be built. The volume of GLORI's shared bus-hypercube optical routing unit was found to be about one cubic centimeter, which is far less than comparable electronic hard-wired implementations. The tradeoffs in implementing various GLORI strategy communication models in terms of resource count, implementation complexity, and latency reduction were also assessed.

Somewhat of an unexpected finding is that GLORI's block-frequent reconfiguration policy is not a general solution for improving performance. Block-frequent reconfiguration is shown to prove useful when it reduces global reference activity to levels that do not saturate the global network. In other words, if there is not sufficient cluster locality, reconfiguration is unlikely to reduce network latency.

1.5 A Variable-Resolution, Nonlinear Silicon Cochlea

Neal A. Bhadkamkar

The cochlea is an organ that sits behind the eardrum on each side of the head. Its fuction is to convert the sound-induced mechanical vibrations of the eardrum into electrical impulses that travel up the parallel fibers of the auditory nerve to the brain, where they are processed and *understood.*[1] A low-power silicon implementation of the cochlea could potentially be used to develop better cochlear implants, higher functionality hearing aids, novel consumer products, and robust front ends for speech recognition systems.

Subthreshold analog CMOS is a particularly interesting technology for the development of intelligent sensing and recognition systems. First commercially used in the development of Swiss watches by Vittoz, it has more recently been extended and popularized by Mead [16] for use in circuits that mimic biological systems in both function and sometimes form. The technology uses standard, and therefore inexpensive, CMOS transistors, operating with continuous currents and voltages rather than the binary voltages used in digital circuits. The current levels are tiny and are those of a transistor in the *off* state; large circuits can be constructed in this medium with very little power consumption.

Subthreshold analog VLSI implementations of cochlea-circuits were pioneered by Lyon and Mead [15] using a cascade of 2^{nd}-order resonant sections. Subseqent cascade implementations have been reported in the literature by Lyon [14] and by Liu *et al.* [10]. Our implementation is also a cascade implementation. It differs from the previously reported cascade implementations in two important respects. First, it is adjustable so that cascades with finer frequency resolution can be constructed without incurring a penalty in terms of the delay to a section with a given best frequency. Second, this implementation is nonlinear and implicitly models the saturating active mechanism of the outer hair cells that gives rise to basilar membrane response curves that become less sharply tuned as the input amplitude is increased.

Earlier work under this project had focused on the design and analysis of a circuit to model the behavior of a small length of the basilar membrane in the cochlea. During the past year we were able to successfully cast this design into silicon and construct a cochlea circuit by cascading 60 of these sections in series. The circuit is fabricated in a standard 2-micron CMOS process through MOSIS and occupies approximately 15% of the area of a standard 4.6 mm × 6.8 mm chip. The circuit is estimated to consume on the order of 300 nanowatts, not including off-chip communication requirements.

Experimental results demonstrate that the circuit is in fact adjustable, allowing cochleas with different resolutions to be simulated while maintaining the same gain and delay characteristics. An important aspect of this adjustability is the fact that the variation from

[1]The brain, in turn, sends control signals back to the cochlea, much as it sends signals to the eyes to focus them on items of interest. However, this aspect of the brain/cochlea interaction is poorly understood compared to the signal processing done by the cochlea itself.

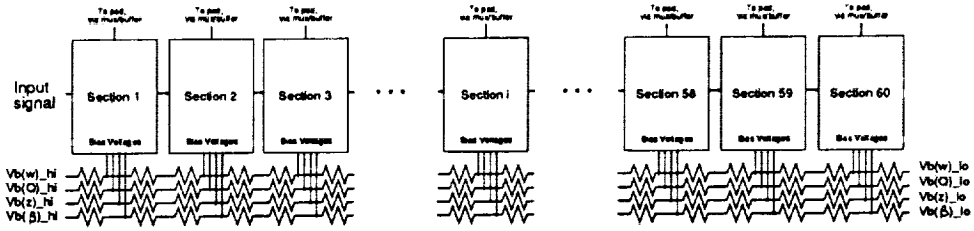


Figure 1.2: Schematic of 60 section cochlea cascade. In the physical circuit, the resistances are implemented by polysilicon wires, and the circuit is folded over at the midpoint, so that sections 31 to 60 are in a second row under sections 30 to 1 respectively.

nominal behavior of the cochlea-circuit due to circuit imperfections is reduced when the circuit is adjusted so that each section simulates a smaller rather than larger length of the basilar membrane. We suspect that a similar statistical mechanism is at work in the biological cochlea.

Experimental results also demonstrate that there are qualitative similarities between the large-input induced nonlinear response of the cochlea-circuit and that of the biological cochlea. Just like the vibration of the basilar membrane in a biological cochlea, the gain versus frequency curves at a given section of the cochlea-circuit become flatter and broader as the input amplitude is increased.

Figure 1.2 shows a schematic of the overall cochlea circuit, and Figure 1.3 shows the impulse response at every fifth section. As expected, the time period and delay increase exponentially as the section number increases.

The design has several limitations. First is the fact that it is not universally stable in its small signal linear region. This becomes an important consideration given the high degree of variation present in subthreshold circuits. A second limitation is the component count. Each section in the current design uses seven wide-range transconductance amplifiers and one buffer amplifier for driving the signal off the chip. Given the result that adjusting the circuit to get more sections per octave results in a lower variance system, one would like sections with the least number of components in order to limit the area requirements of high-resolution cochlea-circuits. A third limitation arises from the observation that this particular circuit was not designed with the goal of variance reduction in mind, which would have led to different choices of building blocks (such as simple rather than wide-range transconductance amplifiers), biasing techniques and layout.[2]

Experience with the first chip has led to several ongoing efforts. A second chip incorporating two 60 section cochlea-circuits and associated hair cell circuits has been designed

[2]Layout is not related to the design per se, but we mention it here anyway. The process of working with this and other chips has led to an appreciation of the special design rules that must be used with subthreshold technology.

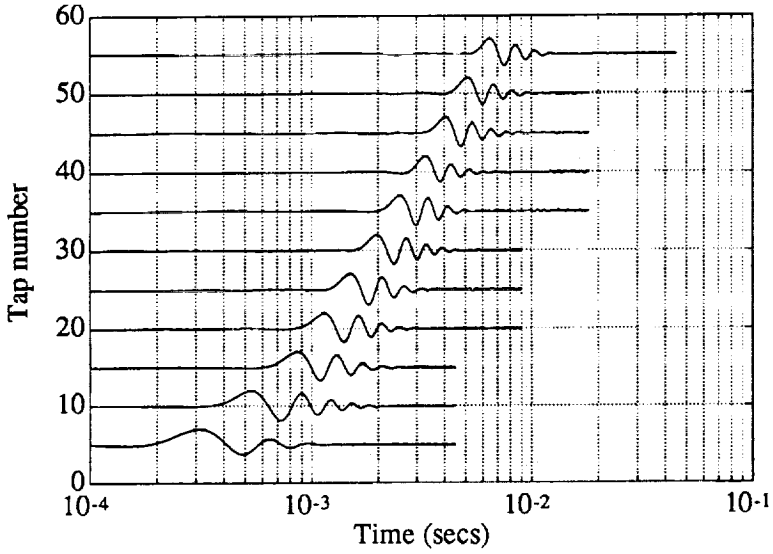


Figure 1.3: Normalized impulse response at every fifth tap.

and fabricated [1], but only partially tested as of this writing. It is meant to provide inputs to a cross-correlator circuit, so that the combined system can locate sound sources in the horizontal plane.

A third chip that has been designed but not fabricated is a finite element implementation of the same cochlear model that formed the basis of this work. It models the bulk of the fluid in the cochlear chambers using a two-dimensional resistive network based on the same principle used by Watts *et al.* [18]. More importantly, it has a circuit for the basilar membrane that explicitly incorporates a saturating active outer hair cell model. While this design has an even higher component count than the present design, it has the advantage of modeling the cochlea in form as well as function, so that it will be easier to incorporate other features such as feedback to the outer hair cells from higher processing centers.

Another effort that is in a much earlier stage is aimed at developing robust feature detectors and incorporating them onto the cochlea chip. The goal of this undertaking is to develop inexpensive but highly functional front ends for speech recognition systems.

An implicit result of our research is the viability of subthreshold analog VLSI as a medium for implementing dense analog computational systems. We designed a circuit to provide performance akin to the results obtained from a mathematical model, layed it out using vanilla design tools, fabricated it using a common process available through MOSIS, and obtained working chips on the first attempt. This should encourage other researchers to use this technology with a greater degree of confidence. However, we caution people that the design rules for obtaining low-variance systems based on subthreshold circuits are more stringent than those used in typical digital or analog circuit designs. Some of these rules are described in Vittoz [17]. However, because of the relative infancy of this design form,

these rules do not appear to be widely known.

1.6 Machine-Independent Parallel Programming

Monica Lam

The focus of our research is to improve the performance of caches on scientific code through compiler optimizations. While the effectiveness of caches has been well established for general-purpose code, their effectiveness for scientific applications has not. Simulation of a system modeled after the MIPS R4000 shows that it is not unusual for a processor to spend over 50% of its execution time in memory accesses on scientific code. Thus improving the memory subsystem of microprocessors is important for microprocessors to fulfill their promise as powerful building blocks for massive parallel machines.

Our approach is to transform the code via blocking and unimodular loop transformations (interchange, skew, and reversal) to maximize the reuse of data brought into the cache before they are replaced. Increasing the cache hit rate improves the system performance by reducing the effective memory access time as well as reducing the total number of memory accesses. Reducing the memory bandwidth requirement is especially important for large-scale parallel machines. After applying these transformations, the compiler then tries to hide the latency of remaining accesses through software-controlled prefetching. Prefetching is the technique of initiating memory accesses by explicit machine instructions before the actual use of the data; the memory latency is thus hidden by computations on other data.

In our research in previous years, we have developed a technique for blocking and loop transformations and demonstrated the effectiveness of blocking on the NASA7 benchmark. This year, we have developed a new algorithm for prefetching, implemented the technique in our SUIF (Stanford University Intermediate Format) compiler, and measured the performance of both blocking and prefetching together on a larger set of benchmarks. These programs were drawn from the NAS Parallel Benchmarks, SPEC benchmarks (which include the NASA7 program), the PERFECT CLUB benchmark and the SPLASH benchmark.

While the technique of prefetching has been studied previously, our work produces, as far as we know, the only compiler system that fully automated prefetching. Since machines today do not have any prefetch instructions, we rely on simulation to evaluate the performance of our algorithm. (The machine we simulated has roughly the same characteristics as the MIPS R4000.) By generating fully functional code, we have been able to measure not only the improvements in cache miss rates, but also the overall performance of a simulated system. Simulation results on a collection of benchmarks showed that the overhead in generating prefetch instructions could potentially negate the advantage of prefetching. Our compiler algorithm eliminates useless prefetches by using the same reuse analysis developed for blocking to issue prefetches only for those accesses that are likely to be cache misses.

Our empirical results are as follows. We found that the loop transformations which

include blocking and unimodular transformations (interchange, reversal, skewing) are applicable in many cases. They can improve the performance of many kernels on uniprocessors by a factor of two or three; however, due to possibly small data set sizes, the optimizations are not very effective in improving the PERFECT club benchmarks. The applicability of prefetching, on the other hand, is impressive. The speedup in overall performance ranges from 5 to 100%, with 6 of the 13 benchmarks improving by over 45%. When compared to an algorithm that indiscriminately prefetches all array accesses, our algorithm can eliminate many of the unnecessary prefetches without any significant decrease in the coverage of the cache misses. Since our prefetching algorithm works cooperatively with blocking, we can get the advantages of both prefetching and blocking in the same system.

Our study argues strongly for the inclusion of prefetch support in future microprocessors. The caches need to be lockup-free; that is, it must be able to handle multiple outstanding memory accesses at the same time. Our recommendation is not to include sophisticated hardware that guesses the data to prefetch. Our work shows that the overhead of software-controlled prefetches is tolerable and software techniques can be much more sophisticated in prefetching, for example, irregular data structures.

References

[1] Neal Bhadkamkar and Boyd Fowler. A sound localization system based on biological analogy. In *Proceedings of the 1993 IEEE International Conference on Neural Networks*, 1993. To be published March 1993.

[2] Brian K. Bray and M. J. Flynn. A Two-Level Windowed Register File. Technical Report CSL-TR-91-499, Computer Systems Laboratory, Stanford University, December 1991.

[3] Brian K. Bray and M. J. Flynn. Efficiently Supporting Subword Loads And Stores. Technical Report CSL-TR-92-514, Computer Systems Laboratory, Stanford University, April 1992.

[4] Brian K. Bray and M. J. Flynn. Tag Caching For Improving Write-Back Cache Bandwidth. Technical Report CSL-TR-92-509, Computer Systems Laboratory, Stanford University, February 1992.

[5] Brian K. Bray and Michael J. Flynn. Translation hint buffers to reduce access time of physically-addressed instruction caches. Technical Report CSL-TR-92-535, Computer Systems Laboratory, Stanford University, August 1992.

[6] Brian K. Bray and Michael J. Flynn. Translation hint buffers to reduce access time of physically-addressed instruction caches. In *Micro-25 Conference Proceedings*, pages 206–209, December 1992.

[7] Brian K. Flachs. Evaluation of a four fold sparse distributed memory prototype. Technical Report CSL-TR-90-444, Computer Systems Lab, Stanford University, August 1990.

[8] Brian K. Flachs. Sparse adaptive memory. Technical Report CSL-TR-92-530, Computer Systems Lab, June 1992.

[9] P. Kanerva. Self-propagating search: A unified theory of memory. Technical Report CSLI-84-7, Stanford Center for the Study of Language and Information, March 1984. Superceded by [Kan88].

[10] W. Liu, A. G. Andreou, and M. H. Goldstein, Jr. Voiced-speech representation by an analog silicon model of the auditory periphery. *IEEE Transactions on Neural Networks*, 3(3):477–487, May 1992.

[11] R. Lupas and S. Verdu. Linear multiuser detectors for synchronous code-division multiple-access channels. *IEEE Transactions on Information Theory*, 35(1):123–136, January 1989.

[12] R. Lupas and S. Verdu. Near-far resistance of multiuser detectors in asynchronous channels. *IEEE Transactions on Communications*, 38(4):496–508, April 1990.

[13] William L. Lynch, Brian K. Bray, and M. J. Flynn. The effect of page allocation on caches. In *Micro-25 Conference Proceedings*, pages 222–225, December 1992.

[14] Richard F. Lyon. CCD correlators for auditory models. In *1991 Asilomar Conference on Signals, Systems, and Computers*, 1991.

[15] Richard F. Lyon and Carver Mead. An analog electronic cochlea. *IEEE Transactions on Acoustics, Speech, and Signal Processing*,

[16] Carver Mead. *Analog VLSI and Neural Systems.* Addison Wesley, 1989.

[17] Eric Vittoz. Subthreshold design. *IEEE Circuits and Systems*, 22:147–153, 1986.

[18] Lloyd Watts, Richard F. Lyon, and Carver Mead. A bidirectional analog VLSI cochlear model. In *Advanced Research in VLSI 1991 : UC Santa Cruz*, pages 152–162. 1991.

[19] R. Ziegler, N. Al-Dhahir, and J. Cioffi. Nonrecursive adaptive decision-feedback equalization from channel estimates. In *IEEE Vehicular Technology Society* 42^{nd} *Conference*, pages 600–603, May 1992. Denver.

Chapter 2

Networking

CASIS networking research has developed as two separate projects: the first project, headed by Professor Cioffi, focuses on the development of techniques to minimize interference in high data rate message transmission. This research shows that it is possible to significantly improve the signal-to-noise ratio in wireless data communications by the use of advanced coding techniques that are closely coupled to the data modulation.

Our other research project in the networking area focuses on high-speed networking and fast packet switching. This research generally is directed at large ground-based networks where large numbers of messages must be routed under highly bursty traffic conditions to the appropriate destination. Professor Tobagi's group has made significant progress in demonstrating the ability to achieve high performance through the development of large switching modules capable of operating at very high speeds.

2.1 Wireless Data Transmission

John M. Cioffi

Reliable wireless transmission of increasing amounts of digital data has become of increased importance to communications in/from space. We have developed methods for sending data quickly and reliably. We have developed some techniques to reduce cross-talk and intersymbol interference in high data rate direct-sequence code-division multiple access methods (CDMA). We have developed fast equalization methods for time-division multiple access (TDMA) modulation schemes. We have started examining multicarrier as a wireless modulation technique. In particular, we are investigating the possiblity of tracking the gain and phase of individual tones in a multicarrier system. This would allow a higher data rate modulation scheme than the traditional phase-shift keyed (PSK) systems used in wireless transmissions.

Progress

Crosstalk rejection in Direct Sequence CDMA

Direct Sequence Code-Division Multiple Access (CDMA) has been proposed for commercial data networks. Its strength is that it can increase the capacity of a system due to the absence of a guardband requirement. Its weakness is that it can suffer from the near–far problem: a user experiencing strong interference from other users (near) while its own signal is relatively weak (far). Techniques such as power control have been proposed for combating the near–far problem. We have studied two additional signal processing methods for when power control is not feasible. The first technique is a linear equalizer, a modification of the decorrelating detector proposed by Lupas and Verdu [5, 6]. The second is a reduced-state sequence estimation (RSSE), an implementation of the Maximum-Likelihood Sequence Detector (MLSD) for CDMA channels with intersymbol interference (ISI) as well as interuser-interference [2].

CDMA is a modulation technique where codes or chip sequences separate users. The codes in CDMA perform the same function that time in Time-Division Multiple Access (TDMA) or frequency in Frequency-Division Multiple Access (FDMA) performs. However, in CDMA the codes are not orthogonal to each other as they are in TDMA or FDMA. With CDMA, the other users appear as background noise. When too many users are present, the CDMA signal starts to degrade. The CDMA signal can also suffer from the near–far problem, where the user-of-interest's signal is relatively weak compared to the other users in the system. At relatively low data rates, CDMA does not suffer from significant ISI due to the properties of the chip sequences. At higher data rates, close to 50 kbits per second or higher with indoor or outdoor data transmission, the CDMA system can suffer from intersymbol interference. We have developed techniques to address interuser interference (or cross-talk), the near–far problem, and ISI.

We have assumed channels where the data rate is high enough to cause both intersymbol interference as well as interuser-interference, or cross-talk. We modulate the chips with a square-root raised-cosine pulse. In simulation we have restricted ourselves to 2 and 3 users, but the techniques can be extended to multiple users.

At the data and chip rates and delay-spreads we have assumed, our single-user ISI channel is analogous to a $1 + aD$ channel, where D is a unit delay of one bit sampling period and a is a multipath coefficient. We also assume that the chip sequence is repeated each information bit and the multipath channels share the same group delay. In both the Linear Detector and the RSSE, we assume a good knowledge of the effective multi-dimensional channel.

In the Linear Detector, we match the received signal with each user's multipath channel and chip sequence. The effective channel response after matching has the form of an invertible matrix. We can invert the channel with either a zero-forcing or minimum-mean-square-error criteria. It can be shown using the matrix inversion lemma, that this solution

is equivalent to a linear equalizer at the chip rate. Implementation at the chip rate has the advantage of not explicitly requiring evaluation of the interfering users' channel responses. For the chip-rate equalizer we can treat the interfering users as noise, and estimate the covariance matrix via a training sequence [7]. The RSSE is a reduced complexity Maximum Likelihood Sequence Detection (MLSD).

We have simulated the performance of the linear equalizer and the RSSE and compared them to the output of a matched filter (RAKE) detector in multi-channel environments where one user experiences a near–far problem. Figure2.1 displays the three methods on 20 simulated channels, with a delay spread of 2.5 microseconds and a chip size of .5 microseconds, with 40 chips per information bit. In this realization, the relative received power of the user-of-interest is 20 dB below that of the interfering user. We see that in this case the traditional RAKE detector has poor performance. The Mimumum-Mean Square Error Equalizer and the RSSE show a 20 dB improvement over the RAKE in this instance. The RSSE in this case shows a 2-3 dB improvement over the Minimum-Mean Square Error Equalizer. In cases where there is no near–far problem, the Minimum-Mean Square Error Equalizer and the RSSE show equivalent performance.

Equalization methods

The minimum mean-square error decision feedback equalizer (MMSE-DFE), depicted in Figure 2.2, is a well-established ISI-mitigating receiver structure for data transmission on multipath dispersive channels. Assuming that the channel pulse response consists of $(\nu+1)$ nonzero taps $\{\mathbf{h}_0, \mathbf{h}_1, \cdots, \mathbf{h}_\nu\}$, then we need $(\nu+1)$ feedback taps $\{\mathbf{b}_0 = 1, \mathbf{b}_1, \cdots, \mathbf{b}_\nu\}$ to eliminate post cursor ISI completely, *assuming correct past decisions.* The anti-causal feedforward filter $\mathbf{w}^* = \left[\begin{array}{ccccc} \mathbf{w}^*_{-(N-1)} & \cdots & \mathbf{w}^*_{-1} & \mathbf{w}^*_0 \end{array} \right]$ then minimizes the mean square of the error sequence $e_k \stackrel{def}{=} x_k - z_k$ by achieving a compromise between precursor ISI and noise.

For many of the new wireless data transmission networks, e.g. the mobile radio standards in the U.S. (IS-54) and Europe (GSM), data is organized, transmitted, and received in finite-length packets. In such packet data applications, the optimal MMSE-DFE settings can be computed indirectly, by first estimating the channel from a known "training" sequence embedded in each packet, and then using the estimate to compute the equalizer settings for use in the recovery of the remaining unknown data in the packet. This indirect channel estimate-based equalization technique, as opposed to the traditional direct adaptation of the equalizer based on the channel output and known training sequence, is desirable from a performance point of view, especially for short packet and training durations [9]. The main obstacle towards popularizing this technique was its high computational requirements, especially in moderate to high speed applications.

By considering short blocks each consisting of N output symbols, as shown in Figure 2.3,

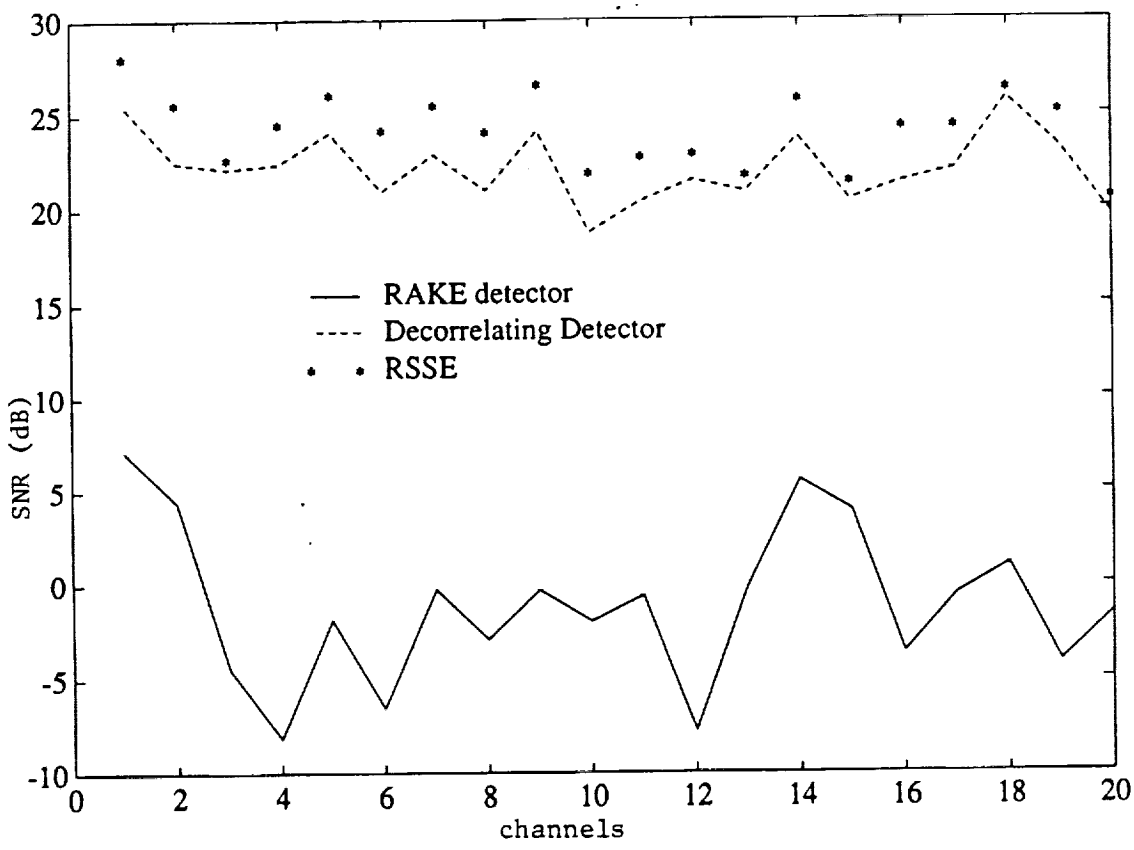


Figure 2.1: Performance of algorithms over 20 simulated channels in a severe near-far environment.

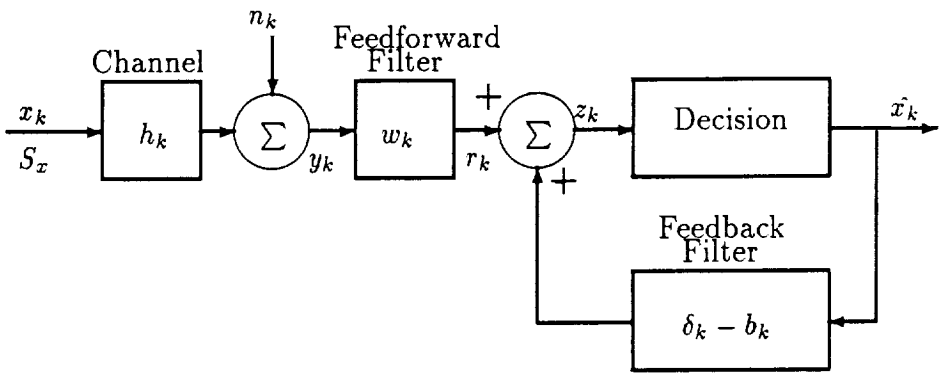


Figure 2.2: Block diagram of the MMSE-DFE.

the channel pulse response $\mathbf{h} = \begin{bmatrix} \mathbf{h}_0 & \mathbf{h}_1 & \cdots & \mathbf{h}_\nu \end{bmatrix}$ [1] can be safely assumed time-invariant over the duration of the block. Therefore, the input–output relation can be cast in matrix form as follows:

$$\begin{bmatrix} \mathbf{y}_{k+N-1} \\ \mathbf{y}_{k+N-2} \\ \vdots \\ \mathbf{y}_k \end{bmatrix} = \begin{bmatrix} \mathbf{h}_0 & \mathbf{h}_1 & \cdots & \mathbf{h}_\nu & 0 & \cdots & 0 \\ 0 & \mathbf{h}_0 & \mathbf{h}_1 & \cdots & \mathbf{h}_\nu & 0 & \cdots \\ \vdots & & & & & & \vdots \\ 0 & \cdots & 0 & \mathbf{h}_0 & \mathbf{h}_1 & \cdots & \mathbf{h}_\nu \end{bmatrix} \begin{bmatrix} x_{k+N-1} \\ x_{k+N-2} \\ \vdots \\ x_{k-\nu} \end{bmatrix} + \begin{bmatrix} \mathbf{n}_{k+N-1} \\ \mathbf{n}_{k+N-2} \\ \vdots \\ \mathbf{n}_k \end{bmatrix} \tag{2.1}$$

or more compactly

$$\mathbf{y}_{k+N-1:k} = \mathbf{H}\mathbf{x}_{k+N-1:k-\nu} + \mathbf{n}_{k+N-1:k} \tag{2.2}$$

The input sequence x_k is assumed complex, independent, and identically-distributed with energy S_x per complex dimension. The noise is white Gaussian with a power spectral density of (lN_0) per complex dimension, where l is the oversampling factor in the fractionally-spaced feedforward filter.

We showed in [1] that computation of the optimal length-N feedforward filter $\mathbf{w}^*$ and length-ν feedback filter $\mathbf{b}$ reduces to the following "Cholesky factorization" problem:

- Perform the triangular factorization

$$\frac{N_0}{S_x}\mathbf{I}_{N+\nu} + \mathbf{H}^*\mathbf{H} = \mathbf{L}\mathbf{D}\mathbf{L}^* \tag{2.3}$$

- The optimal feedback filter is given by

$$\mathbf{b}_{opt.} = \mathbf{L}\mathbf{e}_N \tag{2.4}$$

[1] The use of vector channel coefficients assumes an oversampled polyphase representation of the channel.

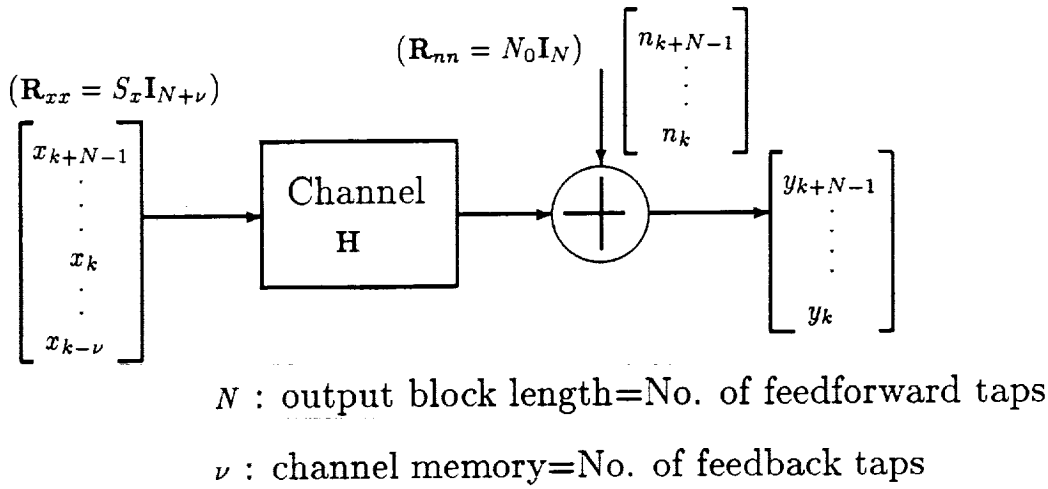


Figure 2.3: Input-output model for packet data transmission on dispersive channels.

- The optimal feedforward filter is given by

$$\mathbf{w}^*_{opt.} = d^{-1}_{N-1}(\mathbf{e}^*_N\mathbf{L}^{-1})\mathbf{H}^* \tag{2.5}$$

where $\mathbf{e}_N$ denotes the N^{th} unit column vector, d_{N-1} is the N^{th} diagonal elemnt of $\mathbf{D}$, and $(.)^*$ denotes the conjugate transpose operation.
Using the recent theory of "structured matrices", we derived the following efficient algorithm for computing $\mathbf{b}_{opt.}$:

Algorithm for computing $\mathbf{b}_{opt.}$

- Initial Condition: $\mathbf{G}_0 = \begin{bmatrix} \sqrt{\frac{N_0}{S_x}} & \mathbf{h}^* \\ \mathbf{0}_{(N+\nu-1)\times 1} & \mathbf{0}_{(N-1)\times 1} \end{bmatrix} \stackrel{def}{=} \begin{bmatrix} \mathbf{u}_0 & \mathbf{v}_0 \end{bmatrix}$

- Recursions: For $i = 0, 1, \cdots, N-1$

$$\begin{aligned} d_i &= |\,\mathbf{u}_i(1)\,|^2 + |\,\mathbf{v}_i(1)\,|^2 \quad : \text{diagonal elements} \\ k_i &= \frac{\mathbf{v}_i(1)}{\mathbf{u}_i(1)} \quad : \text{Schur coefficients} \\ \Theta(k_i) &= \frac{1}{\sqrt{1+|\,k_i\,|^2}} \begin{bmatrix} 1 & -k_i \\ k_i^* & 1 \end{bmatrix} \\ \tilde{\mathbf{G}}_i &= \mathbf{G}_i\Theta(k_i) \\ \mathbf{l}_i &= \frac{\tilde{\mathbf{G}}_i\mathbf{e}_1}{\sqrt{d_i}} \quad : \text{Columns of } \mathbf{L} \end{aligned}$$

$$\mathbf{G}_{i+1} = \tilde{\mathbf{G}}_i \begin{bmatrix} z & 0 \\ 0 & 1 \end{bmatrix} : \text{Updating } \mathbf{G}_i$$

- Output: $\mathbf{b}_{opt.} = \mathbf{l}_{N-1}$

Once $\mathbf{b}_{opt.}$ is calculated, $\mathbf{w}^*$ can be efficiently computed by **back substitution**, as shown in the flow chart of Figure 2.4.

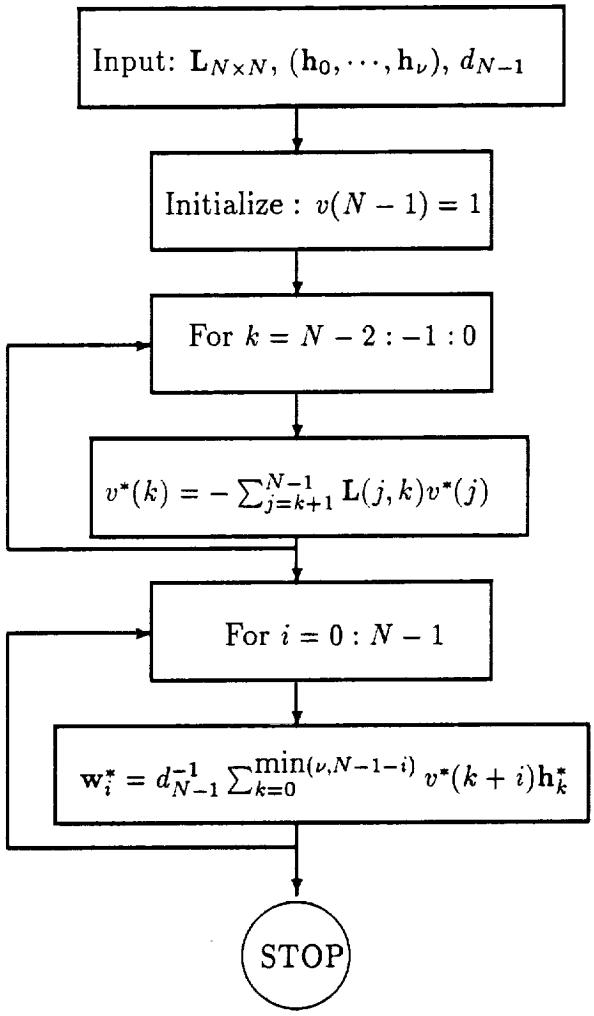


Figure 2.4: Flow chart for computing $\mathbf{w}$.

Figure 2.5 demonstrates the superiority of our proposed algorithm over the currently-most-efficient adaptive Recursive Least Squares (RLS) algorithm, in terms of total compu-

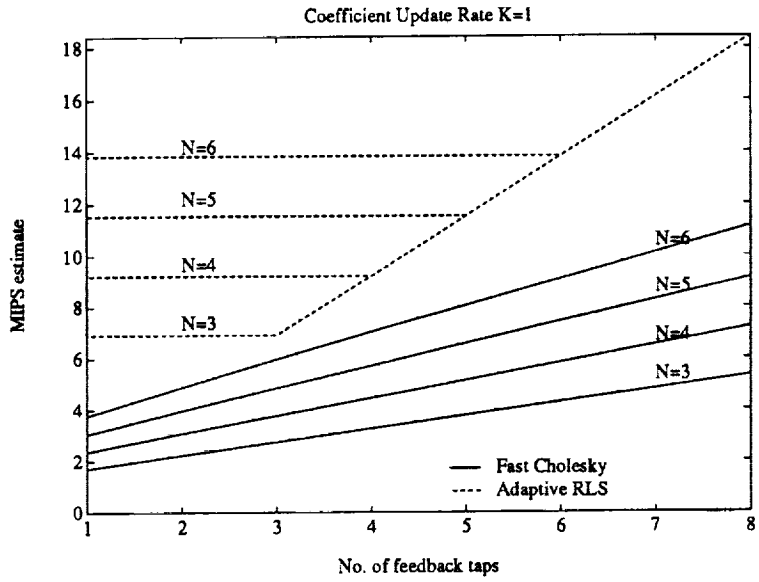


Figure 2.5: A MIPS estimate comparison between our proposed algorithm and the adaptive RLS algorithm.

tational complexity (measured in MIPS) for different choices of the number of feedforward and feedback taps.

For example, for $\nu = 6$ feedback taps and $N = 5$ feedforward taps, our algorithm requires about 7 MIPS compared to the 14 MIPS required by the RLS algorithm—a 50% saving!

Further reduction in computational cost can be achieved by updating the MMSE-DFE coefficients every K symbol periods instead of every symbol period, without significant performance degradation. As an example, for the $\nu = 6, N = 5$ situation mentioned above, the MIPS count drops to approximately 4.2 MIPS when $K = 2$, as shown in Figure 2.6.

Broadcast

The European Broadcast Union had developed a Digital Audio Broadcast (DAB) System that uses an Orthogonal Frequency Division Multiplexing (OFDM) or multicarrier system [3, 8]. This system, also known as the*Eureka* system, can provide CD quality. A similar approach to wireless transmission was put forth by Cimini in [4]. The multi-carrier system can mitigate interblock interference through the use of cyclic-extensions. The channel does not have to be estimated if differential encoding is used. High-data rate systems can suffer from frequency-selective fading. The DAB system assumes that some tones will be in error due to this frequency-selective fading, and uses convolutional codes to correct these errors. The advantage of this system is one does not need to estimate a moving channel.

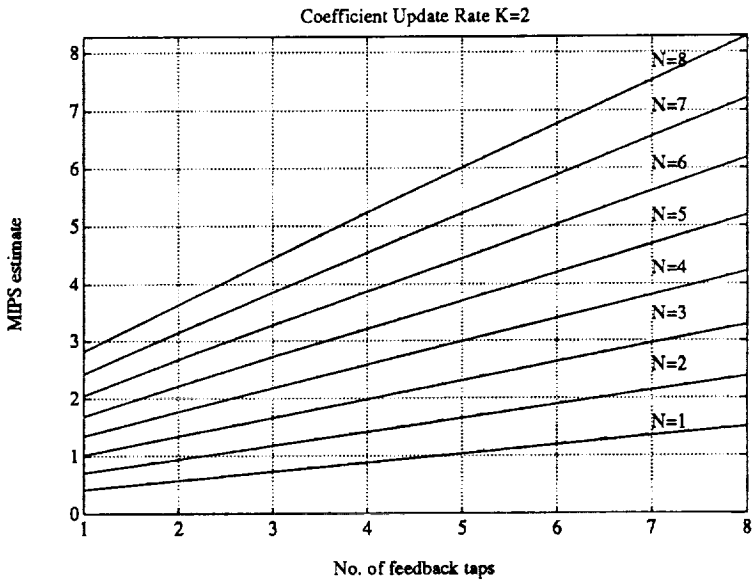


Figure 2.6: MIPS estimate for a coefficient update rate of 2.

Estimating a changing wideband channel is difficult in the time-domain. For this reason, wireless transmission has been restricted to phase-shift key (PSK) transmission. With a PSK system, one does not need to estimate the relative gain of the channel. Restriction to a PSK limits the rate of any system. With a multicarrier system, each tone is designed to have a relatively flat frequency response. To estimate the gain and phase of a single tone is relatively simple compared to estimating a wideband time domain channel. If we can estimate fairly accurately the gain and phase of each tone in a multicarrier system, we can avoid DPSK encoding. A more significant result of the ability to track the tones in a multicarrier system, is the ability to use a higer data rate quadrature amplitude modulation (QAM) modulation system. This could increase the rate of the system.

We are currently examining tracking methods employing a Frequency Equalizer or (FEQ). This is essentially a 1-tap adaptive Linear Mean Square (LMS) equalizer. We are simulating 16 QAM modulation scheme for a single tone that varies according to different Doppler frequencies. Preliminary results show this method to be promising.

References

[1] N. Al-Dhahir and J. Cioffi. Fast algorithms for the computation of the decision feedback equalizer. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 533–536, March 1992. San Francisco.

[2] M. V. Eyoboğlu and S. U. Qureshi. Reduced-state sequence estimation with set par-

titioning and decision feedback. *IEEE Transactions on Communications*, 36:13–20, January 1988.

[3] B. Le Floch J.C. Rault, D. Castelain. The coded orthogonal frequency division multiplexing (cofdm) technique, and its application to digital radio broadcasting towards mobile receivers. In *Globecom '89*, Dallas, Texas, November 1989.

[4] Leonard J. Cimini Jr. Analysis and simulation of a digital mobile channel using orthogonal frequency division multiplexing. *IEEE Transactions on Communications*, COM-33(7), July 1985.

[5] R. Lupas and S. Verdu. Linear multiuser detectors for synchronous code-division multiple-access channels. *IEEE Transactions on Information Theory*, 35(1):123–136, January 1989.

[6] R. Lupas and S. Verdu. Near-far resistance of multiuser detectors in asynchronous channels. *IEEE Transactions on Communications*, 38(4):496–508, April 1990.

[7] D. D. Falconer M. Abdulrahman and A. U. H. Sheikh. Equalization for interference cancellation in spread spectrum multiple acess systems. In *Vehicular Technology Conference VTC '92*, Denver, Colorado, April 1992.

[8] G. Plenge. Dab—a new sound broadcasting system: Status of development—routes to its introduction. *EBU Review*, (246), April 1991.

[9] R. Ziegler, N. Al-Dhahir, and J. Cioffi. Nonrecursive adaptive decision-feedback equalization from channel estimates. In *IEEE Vehicular Technology Society* 42^{nd} *Conference*, pages 600–603, May 1992. Denver.

2.2 Multimedia Computer Communications

Fouad A. Tobagi

In this section, we provide a description of accomplishments and results attained in the past year. This section covers work performed by the High Speed Networking Group under other sponsorship as well. It is provided here to give a comprehensive description of the activities in which the group has been engaging.

Progress

A Hybrid Shared-Memory/Space-Division Architecture for Large Fast Packet Switches

Recently, several architectures of fast packet switches have been prototyped at the rates required by ATM (155.52, 622.08 Mb/s). Given the high operating speeds involved, however, the size of these switching modules has been limited by implementation and technology constraints (the largest realizations support 32-64 I/O ports). Today, in order to deploy ATM in public nationwide networks, it is important to design and implement fast packet switches capable of serving large numbers of users. In addition, these switches must behave efficiently in the highly-bursty traffic conditions that future anticipated applications will generate.

Given the limited size of current switching modules, to build large fast packet switches, a common solution so far has been to interconnect several modules of identical type and size in a multistage configuration. Several multistage fabric architectures can be considered. A configuration which uses the smallest amount of resources would be one that provides a unique path from each source to each destination, as in the banyan multistage architecture. The throughput of such a fabric degrades due to heavy internal congestion under traffic scenarios that distribute the traffic unevenly among the available internal links. In order to avoid congestion and improve the performance, multiple paths for each input–output pair can be provided by increasing the number of stages. As soon as we have multiple paths from an input to an output, however, each path may experience different delays, due to buffering in the intermediate modules along the paths. Thus, if packets are routed independently from each other, out-of-order delivery may result. To prevent this problem, packets belonging to the same connection, or same virtual circuit, should be routed through the same path. In these conditions, given that the traffic for each call fluctuates, to achieve high bandwidth utilization while satisfying a given loss requirement for each call, an intelligent centralized routing algorithm may be needed to determine the path for each incoming call. Such an algorithm may become complex as the switch size increases, and thus constitute a limitation. Even more importantly, these configurations may be prone to internal blocking under multirate traffic, since a route is assigned to a connection and maintained for the whole duration of the call, and rearranging existing calls to accommodate a new one is not practical at the large sizes of interest.

In contrast with this solution of interconnecting identical modules in a multistage configuration, we have explored the use of components of different architectural types to capitalize on the distinctive features of each switching technique to build large fast packet switches. Specifically, we have proposed a hybrid architecture, referred to as the Memory/Space-division/Memory (MSM) switching fabric, which combines input and output shared-memory buffer components along with space-division banyan networks, making it possible to build a switch with several hundred I/O ports. The MSM achieves out-

put buffering, thus performing very well under a wide variety of traffic conditions, and is self-routing, thus adapting easily to different traffic mixes. From an implementation point of view, it is shown that the MSM requires less hardware than a comparable multistage configuration. We have demonstrated the feasibility of this architecture by designing and simulating the circuits of the critical components (e.g., banyan network, memory access, input control, system synchronization) of a large switching module at the 622.08 Mb/s data rate, using currently available semicustom 0.8-μm BiCMOS technology.

Under bursty traffic, in an output-buffer switch such as the MSM, the required buffer size at the outputs increases approximately linearly with the average burst length, eventually becoming unmanageable in case of long bursts. In the MSM, by implementing a backpressure mechanism to control the packet flow, we can evenly distribute the buffers between the inputs and the outputs; this reduces local buffer requirements, and achieves uniform utilization of the buffer capacity of the switch and sub-linear increase of the buffer requirements with the burst length. We have proposed and investigated several distributed buffer-management strategies which are based only on local knowledge of buffer occupation, and can therefore be implemented regardless of the switch size. Using these schemes, up to 70% reduction in total buffer requirements is achieved. These results apply in general to any switch with input and output queueing and backpressure, under bursty traffic conditions.

Performance Analysis of High-Speed Packet Networks in the Presence of Bursty Traffic

With the deployment of high speed networks, a wide range of applications involving the transmission of all types of information will emerge. As it is desirable that all the traffic be carried in an integrated fashion, the most appropriate switching technique to provide the flexibility required has emerged to be fast packet switching, referred to as asynchronous transfer mode (ATM) in the telecommunication world. The performance evaluation of future high speed packet networks is different from that of traditional packet networks in at least two respects: (i) the traffic generated by the sources exhibits time correlations. Thus, the traffic generated by the applications has to be characterized, and adequate traffic models have to be used for the evaluation of network performance; (ii) the measures used for the assessment of the quality of service provided by the network cannot be solely expressed in terms of packet loss rate and packet delay, and instead, have to be appropriately chosen so as to reflect the particular requirements of the applications.

In recent years, several contributions to the characterization of the traffic generated by such applications as voice with silence suppression or video with variable rate coding have appeared; it has been shown that a voice source with silence suppression, a still image or a data source, can be accurately modeled by a bursty source, defined as a traffic source alternating between the ON state, during which packets are generated according to some process, and OFF state, during which no packets are generated; furthermore, a variable rate video source can be modeled as the superposition of several independent bursty sources.

Because of the irregularity of the traffic generated by these sources, it is not desirable to permanently allocate a constant bandwidth to each source, as it would lead to poor bandwidth utilization. Instead, it is preferable to statistically multiplex the traffic generated by several sources; statistical multiplexing, however, leads to packet loss.

A great deal of attention has been devoted to the analysis of multiplexers carrying bursty traffic. The results of these analyses are in terms of long term average packet loss rate, packet delay distribution, average duration of lossy periods, loss rate during lossy periods, and average duration of lossless periods, and are insufficient to assess the quality of service provided to the users. In this work, we have concentrated on two applications: interactive still image transfer and interactive bulk data transfer. In each case, we have defined appropriate measures of quality of service. Using these measures, we have studied a number of network issues such as the effect of load on quality of service, buffer management, coding, forward error correction, bandwidth reservation, as appropriate for the various applications. Our contributions are the following:

a) Mathematical models and numerical techniques for the analysis of multiplexers carrying bursty traffic

Since a traffic source alternates between ON and OFF periods, the packet generation process of such a source exhibits time dependencies. In general, the analysis of multiplexers carrying bursty traffic is quite complex. A tractable Markovian model is obtained if the distribution of the ON and OFF periods are memoryless and independent, and the packet generation process during the ON periods is Markovian (Poisson, Bernoulli). In this case, the state description is two dimensional, and consists of the queue size and the number of sources in the ON state. Even with this tractable model, the state space for the system of interest is so large that appropriate techniques have to be devised to derive numerical results. Numerous such techniques have been published, and can be divided into two categories: (i) techniques which exploit the particular structure of the systems's equilibrium equations, and (ii) techniques which are based on approximating the original system with another system which is easier to solve (e.g., replacing the discrete time arrival and departure processes with their fluid flow approximations). We have surveyed these methods, and have compared them in terms of the accuracy of results they provide. Furthermore, we have devised a new technique which can handle larger size systems than possible with existing techniques, based on the fluid flow approximation and a proper exploitation of the structure of the equilibrium equations. Such a technique is necessary for the analysis of systems supporting highly bursty sources (such as data and still image sources) where the degree of multiplexing is large.

b) Study of the effect of packet loss on image quality

The effect of packet loss on the quality of service provided to such applications as voice, video, or still image transfer, should ideally be assessed by means of subjective evaluation in a realistic network testbed. Because of the high burstiness of the sources, however, the number of sources in the testbed has to be large in order for the experiments to be meaning-

ful; as a result, this approach is costly. Another approach consists of performing subjective evaluations in a testbed consisting of a single source subjected to a pattern of loss emulating that occurring inside a realistic network. As the number of issues affecting the quality of service of the above applications which need to be investigated is large (e.g., traffic characteristics, coding techniques, buffer management, etc), a large number of subjective tests are required to obtain statistically significant results. In order to alleviate this problem, it is necessary to resort to mathematical analysis and simulations, provided that the measures of quality used reflect accurately the requirements of human users.

In this work, we have focused on interactive still image communications and the effect of packet loss on image quality. An appropriate measure of image quality is obtained by dividing the image into regions of equal size (chosen appropriately), and by determining the maximum degradation (in terms of weighted mean square error) over all regions. We have evaluated a network consisting of a multiplexer and several image sources, modeled as bursty sources. Using both analysis and simulation, we have determined the fraction of images whose degradation exceeds a certain level for a number of cases pertaining to the traffic characteristics, the image coding technique (e.g., PCM, DPCM, DCT), the prioritization of the encoder output, the discarding policy implemented in the multiplexer.

c) Selective discarding for interactive bulk data transmission

In our third contribution, we have investigated the transfer of interactive bulk data between computers over ATM networks. With this application, data is typically generated by the source hosts in the form of messages, which are broken down into protocol data units (PDUs). Lost PDUs are retransmitted according to some protocol (e.g., Go-Back-N, selective retransmission). If one of the links between the source and the destination consists of an ATM network, PDUs entering the ATM network are divided into cells by the ATM adaptation layer, and are reassembled as they exit the network. If however, one cell is missing from a PDU, that PDU (and possibly other PDUs) has to be retransmitted.

We have studied the throughput achievable in an ATM network supporting such applications, using both mathematical analysis and simulations. The traffic sources are modeled as bursty sources. Because of the time dependencies in the cell loss process, the throughput is significantly better than if the loss process was assumed independent. It is, however, poor, unless the PDU size is on the order of a few cells. Since having small PDUs is undesirable because of the relatively high value of the overhead involved, we have investigated several means of improving the throughput: we have shown that forward error correction techniques are ineffective at improving the throughput, because of the time dependencies in the loss process. We have shown that selectively discarding of cells according to the PDU they belong to, aimed at concentrating cell loss into as little a number of PDUs as possible, is an effective means of improving the throughput.

d) Comparison between burst-by-burst bandwidth reservation and unrestricted network access for interactive bulk data transmission

In the previous work, we have investigated the effect of packet loss on the quality

of service when access by the sources to the network is unrestricted. It is possible to prevent the occurrence of packet loss by reserving sufficient bandwidth on the path from the source to the destination prior to the transmission of each burst. With burst by burst bandwidth reservation, additional delay is incurred for a burst to reach its destination, as a result of the propagation and waiting delay incurred in the reservation process. We have compared burst by burst bandwidth reservation and unrestricted network access in a network consisting of a single multiplexer, in terms of the delay needed for a burst to reach its destination. Additional considerations arise in the context of multihop networks. Indeed, in such networks, the throughput with burst by burst bandwidth reservation is hampered by the overhead involved in the reservation process. On the other hand, in the presence of applications which require that packets lost in the network be retransmitted, the throughput with unrestricted network access is hampered by the retransmission traffic when the loss rate is large. In both cases, the throughput achievable is dependent on the transmission time of the bursts. Modeling the traffic sources as bursty sources, we have analyzed and simulated a network consisting of several stages of identical switches. Using as a criterion of quality of service the delay necessary for an entire burst to reach its destination, we have studied the effect on quality of such parameters as the load in the network, the number of stages, the burst size, and the buffer size.

Optimal Routing of Video and Audio Streams in Packet Networks

a) Optimum Routing of Multicast Streams in a General Topology

We have considered the problem of optimum routing of a set of multicast streams in a network with a general mesh topology. Each of the N multicast streams is characterized by its source node, s_i, $i = 1, \ldots, N$, by its n_i destinations $d_{i1}, \ldots, d_{in_i}$, and by its bandwidth requirement, b_i. When routing a stream, there are two independent (and often contradictory) criteria: minimum cost (network usage) and minimum delay. Moreover, any combination of these two criteria is also possible.

We have formulated the optimum routing multicast routing problem as a minimization problem, using the objective function described above, subject to the capacity constraints on each link, and to a maximum permissible delay for each stream. Different priorities between the streams are implemented by adding weights for each stream to the objective function.

We have also shown that the optimum multicast routing problem can be formulated as an integer programming problem. The linear relaxation of this problem can be efficiently solved by applying the decomposition principle twice: the first time to decompose the global problem into individual multicast routing problems, and a second time to decompose the individual multicast routing into a series of unicast routing problems.

The modeling technique was tested for a number of small network configurations, and a computer program to solve the global problem is being written. We have also proposed a

heuristic routing algorithm for the minimum cost case, and shown that it finds the optimum solution for a number of simple cases.

b) Optimum Routing of Unicast Streams in WDM Reconfigurable Networks

In a WDM network, where the optical transmitters and/or receivers are tunable, the *logical network topology* can be made independent of the *physical network topology*. Therefore, when routing data one node to the other, there are two degrees of freedom: the actual route and the network topology. The problem we have considered is the optimum routing of unicast streams under those conditions, i.e., determining both the route and the network topology for a given set of requests.

It can be shown that the optimum routing/reconfiguration problem is NP-complete. For a given topology, the routing problem can be formulated as an integer programming problem, whose linear relaxation can be efficiently solved by applying the decomposition principle.

We have studied the problem of numerically finding an approximation of the optimum topology/routing. The problem was divided into two parts: searching in the topology space and finding the optimum routing for each topology. The latter task can be accomplished (exactly) by integer programming; the search on the topology space was implemented using simulated annealing. We derived also an upper bound for the achievable performance of the reconfigurable network and proposed an heuristic algorithm to identify a sub-optimum topology. By using the simulated annealing and the upper bound, we have shown that the proposed heuristics is close to the "true" optimum.

We have proposed a number of control algorithms for the reconfigurable network in the presence of delay, and performed simulation studies. The simulation studies were used to compare the WDM reconfigurable network (distributed switching) with a centralized ATM switch, and we have shown that for a number of scenarios the performance of the WDM network is similar or superior to the performance of a similar ATM switch.

Distributed Computing and Communications: Optimum Task Assignment and Dispatching to Processors Connected by Networks with Arbitrary Topologies

Today, there exist a large amount of computing capabilities in research and business environments in the form of workstations and personal computers networked together. We often have the situation that many of these computers sit idle for a considerable period of time, and thus wasting their computing capabilities. If we can harness this unused computing capability by executing large distributed programs, we would have considerable computing power available to us at relatively low cost.

There are two major elements that will determine the performance of a distributed computation: (i) characteristics of the distributed program to be executed; (ii) the distributed environment on which the program is being executed. These two elements are

independent of each other, however, the resultant performance depends on how they match with each other. The characteristics of a distributed environment is determined by the geographic dispersion and the processing capacity of the computers, the characteristics of the communications network, including its topology, bandwidth, and latency. Some of the important parameters that determine the characteristics of a distributed program are the degree of parallelism and its variation throughout the execution of the program, structure of the precedence relationships among subtasks and communication requirements. The distributed programs can be classified into two major categories according to the way the tasks of the program are related with each other: asynchronous and synchronous programs. In asynchronous programs, tasks are not blocked waiting for completion of other tasks. Conversely, in synchronous programs, tasks are related with others according to some precedence relationships. We can further divide asynchronous programs into two categories based on whether the tasks of the program communicate with each other or not. Similarly, we can classify the synchronous programs according to the structure of their precedence relationships, such as general, series-parallel, merge-splitting, tree, and pipelining structures.

Communication overhead is a major factor that limits the speedup of a distributed computation, and hence the effective gain that can be obtained by harnessing the unused computing capabilities. Given a distributed program and a set of networked processors to execute it, the communication overhead will depend on how the tasks are assigned and dispatched to the processors. It is well known that finding the optimum assignment for a program with a general precedence relationship among its tasks is NP-complete, even under the assumption that there is no communication overhead. There exist heuristics that give execution times within a constant of optimum time for the case where communication between two processors is assumed to be completely independent from all the other communications. However, so far there is no such heuristic to assign such arbitrarily structured tasks to processors connected over a network with an arbitrary topology.

Many distributed programs have task precedence relationships that are structured and somewhat simple. Therefore, it is important to determine an optimum or near-optimum task assignment for such well structured programs distributed across a network with an arbitrary topology. As a first step towards this goal, we have developed an algorithm, which, given a set of processors connected over a network with an arbitrary topology, and a program consisting of asynchronous tasks with identical execution times, determines the optimum assignment of these tasks across the processors, as well as the order and route on which the tasks are dispatched. Using our algorithm, we have examined the speedups obtainable on specific network topologies such as multiaccess bus, star, chain, mesh and completely connected topology.

Hierarchical Storage for Continuous Media

The support of continuous media such as video and audio will require significant changes to current storage systems. Even after compression, storage capacities for continuous media are

larger than for traditional data. For example, the current North American television format can be encoded and compressed to 1.5 Mbps; storing one minute of this requires about 10 MB. Bandwidth requirements for continuous media are more stringent than for traditional data. Although a file transfer may use higher peak bandwidths, audio and video require their bandwidths to be continuous and guaranteed. In order to provide a continuous data stream, data must be delivered to the user before the previous data has been consumed. In other words, real-time deadlines have to be satisfied. A continuous media server can provide continuity by buffering the data and by limiting the number of users. The data has to be buffered while, for example, a magnetic disk is switching tracks or other users are being served. If the number of users is not limited, a user may empty his buffer before it can be filled again. Research on continuous media servers has addressed the tradeoffs among buffer size, the number of users and the response time. Transferring large amounts supports more users because it makes more efficient use of the storage device. However, larger transfers increase buffer size and response time. Batching requests, instead of serving them as they arrive, supports more users because it allows more efficient disk scheduling. However, batching increases response time and possibly buffer requirements. Recent work has focussed on providing playback of fixed-rate video streams from magnetic disks. We have identified the following issues for further research: providing user functionality beyond playback-only, supporting variable bit-rate streams, scaling the storage capacity and the number of users beyond that available from magnetic disks, and integrating different media. Suppose we want to increase the storage capacity without necessarily increasing the number of users. For a server that uses a single magnetic disk-drive, the storage capacity is limited by the capacity of the disk. Capacity can be increased by adding disk-drives but this wastes bandwidth. Also, it can become prohibitively expensive for capacities that are orders of magnitude larger than the capacity of a single disk. Optical disks, magnetic tape and optical tape offer a lower cost per megabyte. Robotically-manipulated tape-cartridge libraries and optical-disk jukeboxes offer capacities on the order of terabytes. However, in general, tape-drives and optical disk-drives have longer response times and lower bandwidths. A storage hierarchy can combine the storage capacity of cheaper devices and the performance of more expensive devices. Through analysis, we have investigated a two-level hierarchy that uses high performance devices, such as magnetic hard disks, at one level and a large capacity device, such as a robotic magnetic-tape library, at the other level. As has been done for a single-level hierarchy, we have addressed the tradeoffs among buffer size, the number of users and the response time. Our first performance model of each hierarchy-level uses two parameters: the average latency and the sustained bandwidth. With this model, we (1) investigated the effect of varying the relative size of the transfer units from the two levels, (2) determined the relationship between the performances of the two levels, and (3) did a preliminary evaluation of different scheduling algorithms at both levels. One important issue was reducing the latency due to the copying and recopying of data as it traverses the hierarchy.

Chapter 3

Neural Nets

Our neural network research has two separate sub-projects. The first is a basic study on neural networks, including a rigorous understanding of the computational capability and limitations of neural networks. Closely associated with this is the need to understand the advantages of neural nets over more conventional control and coputational models.

Our second project is more oriented towards neural network architecure and hardware. In this project, directed by Professor Peterson, we have developed a particular neural network implementation called the *Stanford Boltzmann Engine.* Boltzmann machines are a special class of neural network whose learning algorithm can be shown to minimize a global energy measure using only local information. During 1993 we expect to complete a VLSI implementation of the Stanford Boltzmann Engine. It is projected to handle networks with up to 160 neurons and 20,485 bit weights.

3.1 Studies on Neural Networks

Thomas Kailath

CASIS support was used to help initiate studies in 1989 on the capabilities of artificial neural networks (ANNs). This work has now reached a measure of completeness.

The objectives of our research can be summarized as follows:

1. To provide a rigorous understanding of the computational capability and fundamental limitations of neural networks.
2. To explore the advantages of neural networks over conventional models of computation.
3. To develop systematic procedures for the efficient design of neural networks for various

applications.

The following sections give a summary of previous results and of our recent research on the above issues. We might mention that two of our papers were selected for plenary presentations at the NIPS (Neural Information Processing Society) meetings in Denver, Colorado, in November 1991 and November 1992 – each time there were only about 30 papers selected out of about 500 submitted.

Progress

Threshold Logic and a New Class of Arithmetic Circuits

The study of threshold logic was motivated by the modeling of the individual neurons in the brain with linear threshold elements (LTE's). The key issue in the study of threshold logic is the efficient realization of logic functions with circuits using LTE's, i.e., threshold circuits. Although threshold logic was intensively studied during the 60's [9], a rigorous theoretical groundwork for comparing the computational power of threshold circuits with logic circuits of AND/OR gates was not established until the beginning of the last decade. In their seminal paper [2], Furst, Saxe and Sipser showed that any logic circuit of AND/OR gates with a fixed number of levels (depth) requires an exponential number (2^{n^c}) of gates to compute the parity function of n inputs (even when the fan-in is allowed to be arbitrary). Their results in turn imply that a circuit of AND/OR gates with reasonable size computing the parity of many inputs must have many levels and therefore long computational delays, since the number of levels in a circuit roughly corresponds to the parallel time for computation. On the other hand, it was known in the 50's [8] that any symmetric function (including the parity) of n inputs can be realized by a two-level threshold circuit with n threshold gates. These theoretical results provide the evidence that threshold circuits are more powerful model of computation than logic circuits of AND/OR gates. Moreover, they also provide the motivation for the study of synthesis of logic functions with threshold circuits, since threshold circuits could potentially lead to more compact realization of logic functions. In [11], an algorithm for the synthesis of two-level threshold circuits was proposed.

Research interest in threshold circuits was given an impetus by the results of Chandra *et al.* [1]. It was implicit in their work that many commonly used functions including multiplication of two n-bit numbers, while require long computational delay in circuits of AND/OR gates, can be computed in threshold circuits of constant-depth (thus short delay). In our previous work [17, 18, 23], we have significantly improved and generalized the results of Chandra *et al.* to a wider class of arithmetic functions; for example, we have shown the following interesting results: *the multiplication of two n-bit numbers, and sorting of n n-bit numbers can all be done with three unit delays (independent of n) with threshold circuits of moderate size.* Moreover, we can prove that our threshold circuits are *optimal in depth i.e.,* that no solution can be found with less than three unit delays. We have also generalized

these results to more complicated functions and have shown for example that exponentiation and division can also be computed with depth-3 threshold circuits [23]. Thus *our recent results have provided some of the most efficient threshold circuits (both in terms of size and depth) for computing arithmetic functions.*

Of course, the technologically challenging task of building cheap and reliable threshold gates using physical devices still remain. Academic and industrial researchers have attempted to implement threshold gates with various technologies including electronic and optical devices. Perhaps the most successful attempt was made by Intel using the floating-gate nonvolatile memory technology. Intel's recent ultra high performance Electrically Trainable Analog Neural Network (ETANN) chip [6] provides an efficient implementation of neural network with threshold gates and sigmoidal elements for various applications. These technological advances have revived the study of threshold logic as a promising field of research.

On the Precision of Weights

Many experimental results on neural networks have indicated that the parameters or the weights usually grow exponentially with the size of the inputs. This appears to be a severe limitation on the capability of neural networks in practice. For example, since we would like to implement an artificial neuron or LTE using analog devices, from a practical point of view, it is important to know if the assumption of real valued (infinite accuracy) weights is necessary. For binary n-dimensional input vectors, it is known that each of the weights in an LTE requires at most $O(n \log n)$-bit accuracy [9]. Therefore after normalization, we can assume that all the weights are integers for analysis purpose. However, this still allows the weights to grow exponentially fast $2^{O(n \log n)}$ with the size of the inputs.

We have proved an interesting result [18] which states that *any neural network consisting of linear threshold elements with arbitrary real weights can be replaced by another neural network consisting of elements with $O(\log n)$-bit accuracy only at the expense of an increase in delay by a factor of two and a moderate increase in size.* Very recently, Goldmann, Håstad and Razborov [3] have strengthened our result and showed that the number of layers in the network only needs to be increased by 1 to reduce the precision of the weights to $O(\log n)$-bit accuracy. These results show that the limitation of exponentially large dynamic range of the weights can be removed by slightly increasing the depth. Similar issues on neural networks with sigmoidal elements have not been investigated in previous research.

Depth-Size Tradeoffs in Neural Networks

This issue arises naturally in the implementation of neural networks. Sometimes when the time for parallel computation is not as crucial, one might want to have a smaller network at the expense of a slight increase in the time for computation. In other words, one might want to know if increasing the depth by a small additive constant can significantly reduce the size of the network.

In fact, our recent work [22] has shown that for certain computations one can indeed show striking tradeoffs between depth and size. For example, we studied the depth-size tradeoffs in neural computation of symmetric Boolean functions and some arithmetic functions. Our results show that for these functions, a small increase in the depth can significantly decrease the size required in the network. In particular, we showed that *any* symmetric Boolean function (in n variables) can be computed with $O(\sqrt{n})$ threshold gates in a depth-3 network while the best known result required $O(n)$ threshold gates in a depth-2 network. Moreover, we can prove that our depth-3 network is almost optimal in the number of threshold gates and no significant reduction in the network size is possible (for arbitrary symmetric functions) even if unbounded depth networks are considered. *This result answers an open question outstanding since 1961.* Similar depth-size tradeoffs are also shown for functions such as *MULTIPLE SUM*, *ADDITION*, and *COMPARISON*. Our work also shows how to design almost optimal depth-d networks that compute specific useful symmetric functions such as the parity of n variables using $O(n^{1/(d-1)})$ threshold gates.

Lower Bound Techniques for Neural Networks

Previous techniques for deriving lower bounds on the size of neural networks have been restricted to the case of two layers [12, 4]. We have generalized previous work and developed new tools for establishing lower bounds on the size of neural networks; our results are the first known in obtaining lower bounds on the size of threshold circuits with depth more than two. For example, using results from the theory of rational approximation [10] and harmonic analysis [7], we have shown that the size of depth-3 networks computing arbitrary symmetric functions is $\Omega(n^{1/2-\epsilon})$, for any $\epsilon > 0$. This shows that the depth-3 networks mentioned above are almost optimal. Similarly we have shown that for depth-d networks implementing PARITY or COMPLETE QUADRATIC functions, at least $\Omega(dn^{1/d-\epsilon})$ LTEs are needed for any fixed $\epsilon > 0$. This again shows that our implementations of these functions are almost optimal in size. Recently, by developing more advanced approximation techniques, we were able to generalize our previous lower bound results on threshold circuits to more general neural networks including sigmoidal networks and radial-basis networks. Our lower bound techniques yield a unified approach to the complexity analysis of various models of neural networks with feedforward structures. Moreover, our results indicate that in the context of computing highly oscillating symmetric Boolean functions, networks of continuous-output units such as sigmoidal elements do not offer significant reduction in size compared with networks of linear threshold elements of binary outputs.

More recently we have been able to relate the communication complexity [16] of a Boolean function to the node complexity of threshold and related circuits implementing such a function. Our results [16] *lead to the best known lower bounds* on the size of threshold circuits *with unrestricted depth and unbounded weights.* Although these are the best known results, the lower bounds are at most linear in n (the number of input variables) and are matched by corresponding upper bounds only for a limited number of functions.

Analysis via a Geometric Approach

Formal approaches for analyzing and designing neural networks are slowly emerging. At the present time, however, there are very few effective tools for analyzing multilayer neural networks. The basic difficulty seems to be the lack of a quantitative understanding of the input-output behavior of every gate in the network.

In very recent work [13, 14] we have developed *a new geometric approach* for investigating the power of neural networks and their limitations. Our approach views a Boolean function of n variables as a vector in a 2^n-dimensional Euclidean space and invokes tools from linear programming and linear algebra to derive new results on the realizability of Boolean functions using LTEs. Some of our results lead to generalizations of key results concerning threshold circuit complexity, particularly those that are based on the so-called spectral or Harmonic Analysis approach.

Classifying Linearly Non-Separable Patterns using Perceptrons

The ability to "learn" is one of the most important features in neural networks. In our recent work [15], we have discovered new results on the computational properties of the perceptrons. Whereas learning and convergence properties of perceptrons are well understood for the case where the input vectors (or the training sets) to the perceptron are linearly separable, little is known about the behavior of a linear threshold element when the training sets are *linearly non-separable.* In our work, we have obtained the first known results on the *structure of linearly non-separable training sets* and on the *behavior of perceptrons* when the set of input vectors is linearly non-separable. More precisely, we have shown that using the well known perceptron learning algorithm a linear threshold element can learn the input vectors that are provably learnable, and identify those vectors that cannot be learned without committing errors. In order to develop our results, we first establish formal characterizations of linearly non-separable training sets and define learnable structures for such patterns. We also prove computational complexity results for the related learning problems. Based on such characterizations, we show that *a perceptron does the best* one can expect for linearly non-separable sets of input vectors and learns as much as is theoretically possible.

References

[1] A. K. Chandra, L. Stockmeyer, and U. Vishkin. Constant depth reducibility. *Siam J. Comput.*, 13:423–439, 1984.

[2] M. Furst, J. B. Saxe, and M. Sipser. Parity, Circuits and the Polynomial-Time Hierarchy. *IEEE Symp. Found. Comp. Sci.*, 22:260–270, 1981.

[3] M. Goldmann, J. Håstad, and A. Razborov. Majority Gates vs. General Weighted Threshold Gates. to appear in Seventh Annual Conference on Structure in Complexity Theory.

[4] A. Hajnal, W. Maass, P. Pudlak, M. Szegedy, and G. Turan. Threshold circuits of bounded depth. *IEEE Symp. Found. Comp. Sci.*, 28:99–110, 1987.

[5] T. Hofmeister, W. Hohberg and S. Köhling . Some notes on threshold circuits and multiplication in depth 4. *Information Processing Letters*, 39:219–225, 1991.

[6] M. Holler, S. Tam, H. Castro, and R. Benson. An Electrically Trainable Artificial Neural Network (ETANN) with 10240 "Floating Gate" Synapses. *International Joint Conference on Neural Networks*, vol.2, pp. 191–196, June 1989.

[7] R. J. Lechner. Harmonic Analysis of Switching Functions. In A. Mukhopadhyay, editor, *Recent Development in Switching Theory*. Academic Press, 1971.

[8] S. Muroga. The principle of majority decision logic elements and the complexity of their circuits. *Intl. Conf. on Information Processing, Paris, France*, June 1959.

[9] S. Muroga. *Threshold Logic and its Applications*. John Wiley & Sons Inc., 1971.

[10] D. J. Newman. Rational Approximation to $|x|$. *Michigan Math. Journal*, 11:11–14, 1964.

[11] A. Oliveira and A. Sangiovanni-Vincentelli. LSAT- An Algorithm for the Synthesis of Two Level Threshold Gate Networks. *International Conference on Computer-Aided Design*, pages 130–133, 1991.

[12] R. Paturi and M. Saks. On Threshold Circuits for Parity. IEEE Symp. Found. Comp. Sci., October 1990.

[13] V. P. Roychowdhury, K. Y. Siu, A. Orlitsky, and T. Kailath. A Geometric Approach to Threshold Circuit Complexity. Proc. of the Fourth Annual Workshop on Computational Learning Theory. San Mateo: Morgan Kaufman (publisher), pp. 97-111, August, 1991.

[14] V. P. Roychowdhury, K. Y. Siu, A. Orlitsky, and T. Kailath. On the Circuit Complexity of Neural Networks. Touretzky, D. S., Lippman, R., (eds): Advances in Neural Information Processing Systems 3. San Mateo: Morgan Kaufman (publisher), April 1991, pp. 953-959.

[15] V. P. Roychowdhury, K. Y. Siu, and T. Kailath. Classification of Linearly Non-Separable Patterns by Linear Threshold Elements. Technical report, Purdue University.

[16] V. P. Roychowdhury, K. Y. Siu, and A. Orlitsky. Lower Bounds For Threshold and Related Circuits and Communication Complexity. Technical Report, Purdue University, June 1992.

[17] K. Y. Siu and J. Bruck. Neural Computation of Arithmetic Functions. *Proc. IEEE*, 78, No. 10:1669–1675, October 1990. Special Issue on Neural Networks.

[18] K. Y. Siu and J. Bruck. On the Power of Threshold Circuits with Small Weights . *SIAM J. Discrete Math.*, 4(3):423–435, August 1991.

[19] K. Y. Siu, J. Bruck, T. Kailath, and T. Hofmeister. Depth-Efficient Neural Networks for Division and Related Problems . *IEEE Trans. Information Theory*. to appear.

[20] K. Y. Siu, V. P. Roychowdhury, and T. Kailath. Computing with Almost Optimal Size Threshold Circuits . In *IEEE International Symposium on Information Theory, Budapest, Hungary*, June 1991.

[21] K. Y. Siu, V. P. Roychowdhury, and T. Kailath. Rational Approximation, Harmonic Analysis and Neural Networks. In *Proc. Int. Joint Conf. on Neural Networks (IJCNN'92)*, Baltimore, June 1992.

[22] K. Y. Siu, V. P. Roychowdhury, and T. Kailath. Depth-Size Tradeoffs for Neural Computation . *IEEE Transactions on Computers* , Special Issue on Neural Networks , pp. 1402-1412, December 1991.

[23] K. Y. Siu and V. P. Roychowdhury. On Optimal Depth Threshold Circuits for Multiplication and Related Problems. Tech. Rep. No. ECE-92-05, University of California, Irvine, February, 1992. Accepted for publication in SIAM Journal on Discrete Mathematics

3.2 Neural Network Architecture and Hardware Design

Allen M. Peterson and James B. Burr

This section summarizes the research conducted during 1992 on architectural principles and the design of digital VLSI hardware for the implementation of neural networks. The principal group effort, now nearing completion, was the design of a digital VLSI chip which implements the "Boltzmann" learning algorithm. Additional effort was directed toward investigation of energy-efficient architectures for large nets.

Progress: Individual Research

This section describes individual student research results for 1992 over and above the substantial effort put in on the Stanford Boltzmann Engine (next section) by Mike Murray, Mike Leung, Kan Boonyanit, and Kong Kritayakirana.

Texture Classification – Michael M. Leung

This work focuses on the problem of classifying textures, with particular emphasis on the problem of scale and rotation invariance. Two techniques are under investigation. The first attempts to extract features which are inherently invariant. The second is to scale and rotate the data before classifying it.

Approximate Gradient Descent – Kan Boonyanit

This work focuses on a method to reduce the amount of computation in backpropagation by only propagating the largest errors backward through the network. It involves analyzing

the impact of learning rates and error thresholds on the amount of computation required. and the design of efficient VLSI structures for retaining the largest K samples in a data stream.

Wafer Defect Classification – Karen Huyser

This work focuses on applying neural networks to the task of classifying defect patterns on silicon wafers. The patterns can be classified by shape and size; each shape indicative of a different type of problem in the fabrication process. A noise model has been built and a variety of networks trained using synthetic data. Weight elimination is being investigated to improve generalization.

Precision, Learning and VLSI – Michael Murray

This work focuses on several key issues related to improving performance and energy efficiency of VLSI neural nets. Most of the effort during 1992 was directed toward converting the C-language behavioral description of the Boltzmann Engine, which had been optimized for low-precision learning the previous year, into silicon. This conversion process involved a number of steps as the control constructs and actions in C were translated into Boolean expressions which would meet the aggressive timing constraints in the design.

Mesh Communications – Yen-Wen Lu

This work involves investigating the communication properties of a number of different mesh topologies which are area-efficient and can "tile" well on 3D stacked multichip modules. Yen-Wen has developed a novel structure he calls the "segmented reconfigurable bus" which combines the flexibility of a reconfigurable bus architecture with the high throughput of a nearest neighbor mesh. The optimum segment length is derived from realistic evaluation of propagation delays in logic depth 10 architectures.

He has analyzed the performance of various meshes on variations of the permutation problem as a function of the connectivity profile. He has developed a statistical approach to performance analysis which can be applied to finite-duration problems like permutation as well as steady-state problems like constant injection. He has developed a simple hill-climbing strategy that avoids deadlock without requiring a separate buffer for each channel.

Mapping Neural Nets onto Meshes – Kong Kritayakirana

Kong is building on Yen-Wen's work by mapping neural network algorithms into segmented bus meshes. His objective is to find mappings which minimize communication energy. This involves optimizing the size of a processor's local memory, how to assign neurons

to processors, how to efficiently represent the network connectivity, and how to transmit updated activation data and weights to the places in the net where they are needed.

Progress: The Stanford Boltzmann Engine

The Stanford Boltzmann Engine has been taped out. First silicon is expected back in early February 1993. This section contains an overview of the chip, including a discussion of its distinguishing features.

Boltzmann machines

Boltzmann machines [HSA84, AH85, Hin89, PH89] are a special class of neural networks whose learning algorithm can be shown to minimize a global energy measure using only local information. The network contains recurrent connections (feedback), which can result in multiple responses to a single stimulus depending on the time evolution of the neural activations in the presence of noise. The Boltzmann learning algorithm, like back propagation, performs steepest descent in weight space, but can avoid local minima by blurring the error surface at high temperatures.

Network structure

Boltzmann networks consist of three types of units: "visible" units, which handle network input and output, "hidden" units, which handle internal connections, and a "true" unit, which supplies a "1" to all the other units. Connections between the units are usually symmetric ($w_{ij} = w_{ji}$), although this constraint can be relaxed if the weights are permitted to decay [Gal89, Al90]. Network response to an input vector is found by initializing the hidden and output activations to random values and then annealing the network according to a temperature schedule. Annealing is necessary due to the recurrent nature of the connections, and to avoid local minima.

There are two distinct types of Boltzmann machines: "stochastic" [HSA84, AH85], and "mean field" [PA87] or "deterministic" [Hin89]. The neurons in stochastic Boltzmann machines are binary valued stochastic decision elements. The neurons in deterministic Boltzmann machines are deterministic and multivalued. Our chip is a deterministic Boltzmann machine.

Boltzmann learning

Boltzmann learning consists of two phases: a "teacher" phase, and a "student" phase. During the teacher phase, all visible units are clamped to desired values and the neural activations are computed. During the student phase, the outputs are allowed to run free.

Annealing is accomplished by multiplying the sum of products by a temperature T before applying the usual sigmoid nonlinearity:

$$x_i = \frac{2}{1 + e^{-\frac{1}{T}\Sigma w_{ij} x_j}} - 1.$$

When T is large, the sigmoid response is very broad, and the error surface is blurred, obscuring all but the most prominent features. When T is zero, the sigmoid is a step function, revealing the fine structure of the error surface.

Weights are updated according to the correlations between a neuron's input and output during the student and teacher phases. The weight update rule depends only on information local to the neuron, which is often cited as an advantage for VLSI implementation.

In a stochastic network, each weight is changed by an amount given by the expected value of the student and teacher correlations averaged over time:

$$dw_{ij} = e(< t_i t_j > - < s_i s_j >)$$

which can be implemented stochastically as

$$\text{if } (t_i \& t_j \;\&\&\; rand() < e)\; w_{ij}{+}{+};$$

$$\text{if } (s_i \& s_j \;\&\&\; rand() < e)\; w_{ij}{-}{-};$$

where w_{ij} is the strength of the synaptic connection from neuron i to neuron j, dw_{ij} is the change to be made to weight w_{ij}, e is the learning rate, t_i is the output of neuron i during the teacher phase, and s_i is the output of neuron i during the student phase.

In a deterministic network, the neural activations represent the expected values of the corresponding stochastic variables, so the weight update rule is

$$dw_{ij} = e(t_i t_j - s_i s_j)$$

Although this is a more complex expression, it only has to be computed once at the end of the anneal cycle for each weight, rather than once per iteration. Our chip implements a deterministic Boltzmann learning algorithm.

VLSI implementation

The Stanford Boltzmann Engine is a digital learning neurochip which implements the mean field Boltzmann learning algorithm. It can handle networks with up to 160 neurons and 20,480 5-bit weights. It is implemented in 1.2μ CMOS, runs at 180MHz, and dissipates 2.0W, achieving 5.76 GCPS (billion connections per second) and 720 MCUPS (million

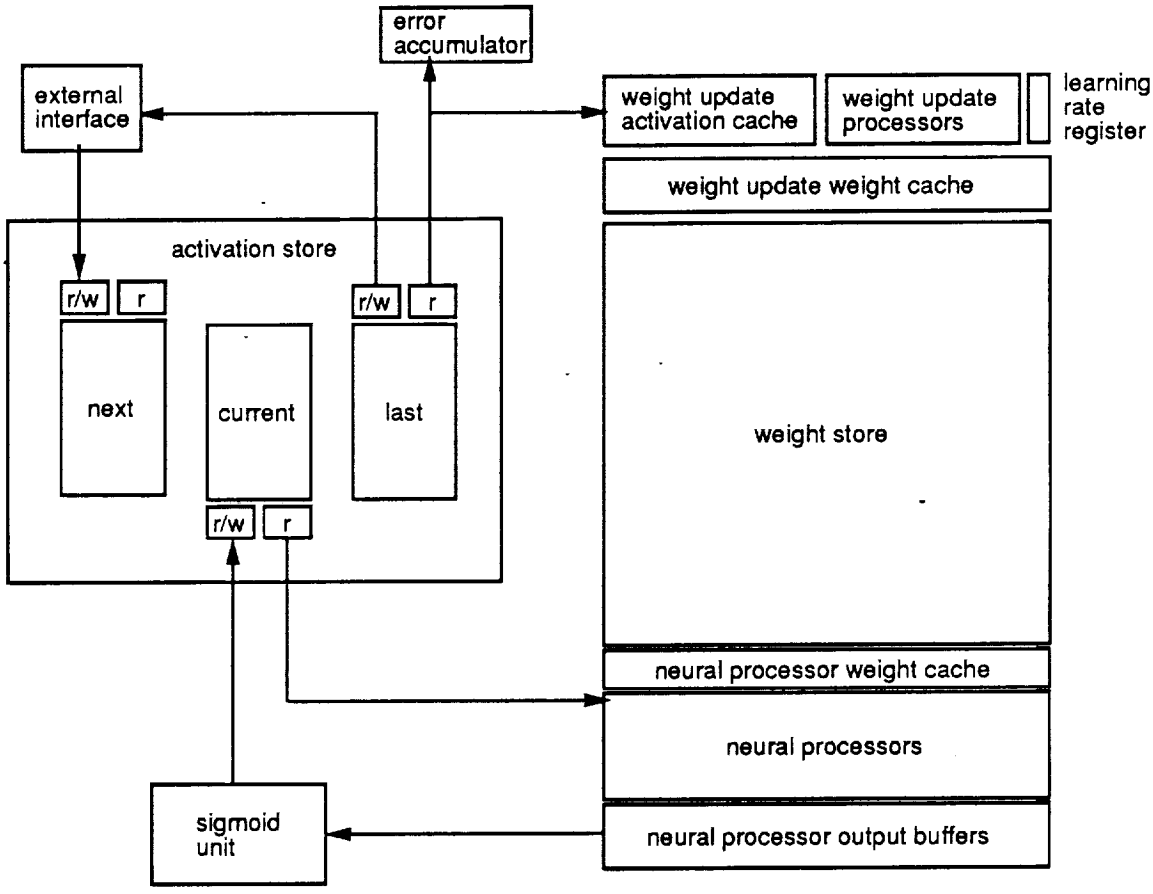


Figure 3.1: Stanford Boltzmann Engine block diagram.

connection updates per second) at 350pJ (picojoules, 10^{-12} joules) per connection. It achieves high performance at low energy per operation primarily through a combination of pipelining and reduced precision arithmetic.

The chip (see Figure 3.1) has 32 neural processors (NPs), 4 weight update processors (WPs), and 20,480 5-bit weights organized in 4 blocks. Each block has 8 NPs, one WP, and 5,120 weights. The chip has a sigmoid unit, three 160-word activation stores, and an external interface with an 8-word instruction cache. It also has a random number generator for initializing hidden units and an error accumulator for measuring error rates during learning. The NP to WP ratio was chosen to match the throughput requirements of a typical anneal schedule.

All paths in the system are balanced to match the latency through a 4:2 adder: 5.3ns in nominal 1.2μ CMOS at 5V and 27degC. This was particularly challenging in the control design, since the memories and datapaths were all pipelined. It was useful to ignore the cycle skew while designing the control, and to shift events to their real cycles afterwards.

Pad description

Figure 3.2 shows the pinouts of the Stanford Boltzmann Engine. The pins and their functions are tabulated below.

name	number	type	function
D	23	b	data
A	16	i	address
CS	1	i	chip select
RW	1	i	read/write
ACK	1	o	acknowledge
CDR	1	i	cold reset
WMR	1	i	warm reset
GO	1	i	go/stop
EA	1	o	error accumulator available
ER	1	o	error
OA	1	o	output available
BGO	1	i	BIST GO
BSTS	1	o	BIST status
CLKSEL	3	i	clock source select
CLKMD	2	i	clocking mode
CLKOEN	1	i	clock output enable
CLKOUT	1	o	clock output
CLKIN	1	i	clock input
CLKVCO	1	i	clock VCO input
TP	4	o	test points

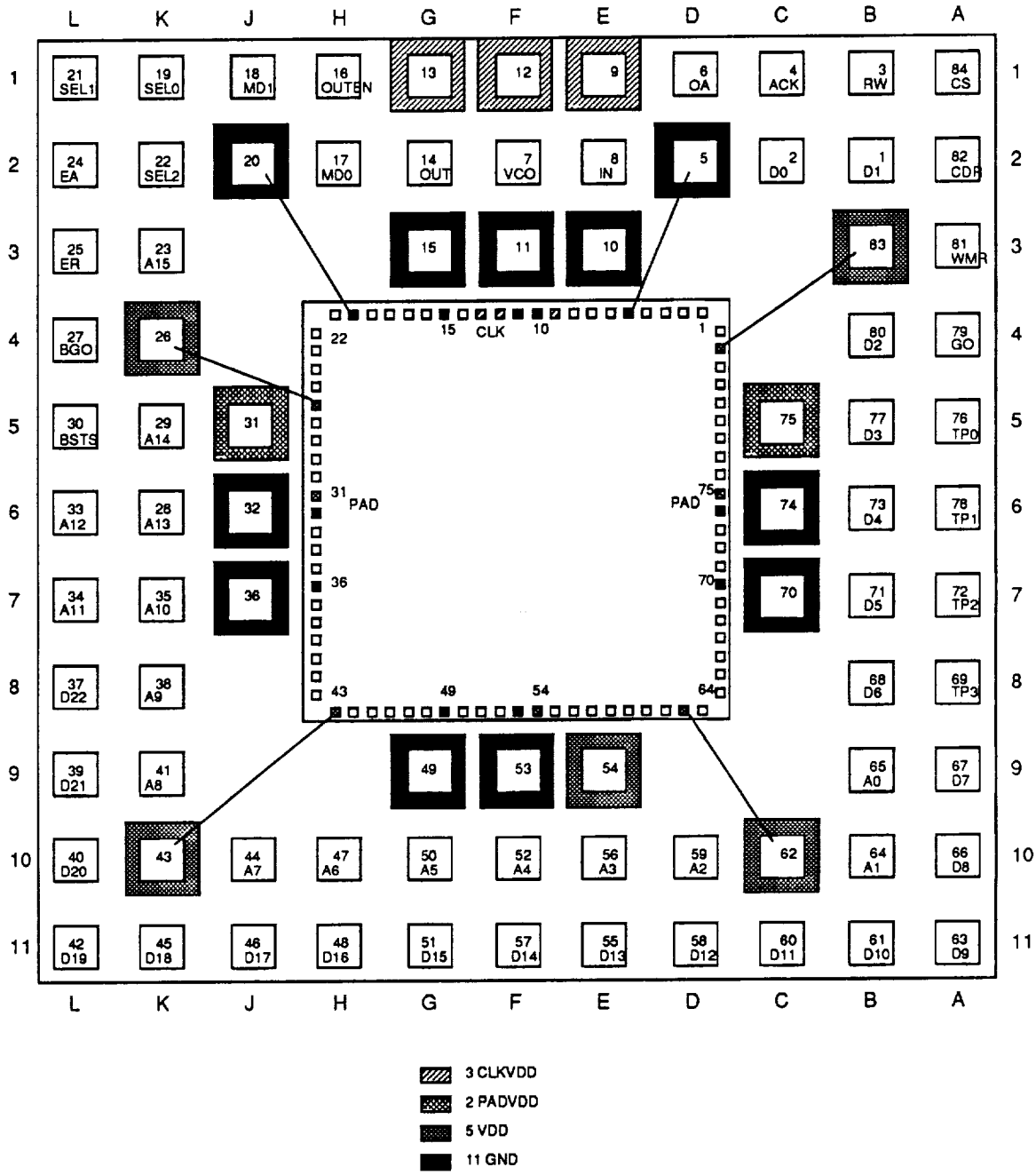


Figure 3.2: Stanford Boltzmann Engine Pin diagram.

types

b bidirectional
i input
o output

Unique features

The Stanford Boltzmann Engine has a number of key features which distinguish it from other neural network chips. These include:

Clocking

- The entire chip was designed to logic depth 10. This is what gives us a 180MHz clock in a 1.2μ technology. There are many paths on the chip which are near this length. The control, datapaths, and memory are all deeply pipelined to meet the timing constraints.
- The design minimizes power dissipation by only clocking active subsystems. This is done using a novel 3-tier gated clock hierarchy.

Arithmetic

- The chip uses 5-bit weights and activations, 5×5 bit multipliers, and 17-bit carry-save accumulators.
- Only the upper three bits of the activations are used in the weight update computation.
- The temperature is expressed in a reduced-precision floating point format, with a 2-bit mantissa and a 4-bit exponent.
- The neural and weight update processors are small. Each has about 1400 transistors and measures about $500\lambda \times 1000\lambda$.
- The chip has a single sigmoid unit consisting of a temperature multiplier and a 64-element lookup table.

Memory

- The chip uses 1-transistor dynamic RAM for weight storage, with the storage node implemented as a fet capacitor, and a simple single-ended, self-compensating sensing scheme.

- The chip uses 6-transistor SRAM for activation storage, with low power pipelined read/write timing.
- Each weight store has two 8-word caches so the weight memory is only accessed once every 4 cycles. The dual caches make the memory appear to be multiported.
- The three banks of activation store permit concurrent feedforward, weight update, and external pattern load without contention.

Control

- The control consists of state machines, counters, and comparators. The largest counter is only 10 bits.
- Meeting the timing constraints in the control was not too difficult.

External interface

- The external interface uses a 16-bit address bus, a 23-bit data bus (to read or write 4 activations or weights at a time), and a simple asynchronous handshake protocol using control signals **chip_select** and **acknowledge**.
- In most applications, the external datarate is much slower than the internal clock rate (1 MHz vs-180MHz). The key is to put the activation store on-chip (many other neural net chips have on-chip weights but few have on-chip activations).

Testability

- The internal state of the chip is memory mapped. All registers can be read and written from a single diagnostic bus.
- The chip includes a single-step mode, which was difficult to design due to the deep pipelines, but which we felt was essential for debugging. In single-step mode, the chip can go for N cycles and then stop.
- The chip has three clocking modes for flexibility in testing: external, on-chip VCO with off-chip bias, and on-chip VCO with on-chip bias.

Design methodology

- Our main design strategy was to define transistor-level netlists using **net**, simulate these netlists, lay out the circuits using **magic**, and compare layouts to their schematics using **gemini**. This worked very well, especially since some of the gate-level circuits we used in the layout worked fine using **spice**, but not in **irsim**. In these few cases we maintained two sets of cells - one to match the layout, and one for simulation.

- The "max-time" capability built into rsim was indispensable for meeting the timing constraints in the control.

Simulation with irsim

- DRAMs and SRAMs work fine with extra capacitance in the symbolic description.
- Latch feedthrough can also be controlled with extra capacitance.

Problems with the tools

- magic needs a better block router for hierarchical control design.
- irsim needs to support behavioral models to speed up full-chip simulation.

CAD tools

We built the chip using a mix of commercial and university-based CAD tools. Architecture exploration, including area, performance, and power estimation, were done using matlab. Algorithm development, precision studies, and behavioral modeling were done in C. Transistor level schematics were described using net [Ter82]. VLSI layout, extraction, and design rule checking were done with magic [2]. The layout was verified against the schematics using gemini [1]. Detailed timing simulation and analysis were done with spice [Na75]. Functional simulation was done using irsim[Lager88]. Coarse timing, critical path analysis, and power measurements were done using our own version of rsim [Ter82, Bur88].

Multichip networks

The chip is designed to be tiled in a regular array with nearest neighbor connections to implement multichip networks. The sigmoid units are deactivated in all chips except those along the major diagonal of the array. Activations are partially accumulated in each chip in the array and then shifted vertically toward the major diagonal. The sigmoids are performed on the diagonal, and the resulting activations shifted vertically and horizontally away from the major diagonal to make x_i and x_j available for updating w_{ij}. Each chip has four input and four output ports. This scheme permits network relaxation in constant time. A 4×4 array measures 3.2cm×3.7cm, supports up to 640 neurons, 327,680 weights, 64GCPS, and 16GCUPS at 3.3V and 10W.

Comparative performance

Table 3.1 shows how the Stanford Boltzmann Engine compares in capacitance, performance, and enrgy to other implementations reported in the literature.

Machine	C	CPS	CUPS	W	J/C	rel perf	rel time
Digital chips							
Sun4	-	1.5 M	400 K	40	26μ	1	1 month
DSP32C	-	40 M	10 M	1	250n	25	1 day
ASI CNAPS	256K	1.6 G	100 M	5	3n	250	2 hrs
SBE	20K	5.8 G	720 M	2	345p	1,800	16 min
SBE16	320K	93.0 G	12 G	32	345p	30,000	1 min
Analog chips							
Bellcore	496	100 M	100 M	0.5	5n	250	2 hrs
Mitsubishi	10K	500 M	500 M	1.5	3n	1,250	24 min

Table 3.1: Comparative performance of existing implementations. "SBE" is the Stanford Boltzmann Engine; "SBE16" is a 4×4 mesh of SBEs.

A Sun4 workstation has a feedforward computation rate of around 1.5 million connections per second, and can process one training vector every 100 msec (1,800 times slower than our Boltzmann Engine).

The first analog Boltzmann chip was designed at Bellcore by Joshua Alspector and colleagues in 1988 [Al88]. It was a test chip with only 6 neurons and 15 bidirectional synapses, but was the first VLSI neurochip with an on-chip learning capability. A more recent version of this chip [Al90] has 496 weights and 32 neurons and can accept a new pattern every 5μsec, implying 100 MCUPS.

An analog Boltzmann chip reported by Yutaka Arima et al. at the 1990 Symposium on VLSI Circuits [Ar90] is implemented in 1.0 micron CMOS, has 10K synapses, 125 neurons, is fully analog, processes one training vector every 20 μsec, and dissipates 1.5W.

The Adaptive Solutions CNAPS chip [Ham90] is fully digital, implemented in 0.8μ CMOS, and is over 5 cm^2. It has 64 processors, each with 4096 bytes of SRAM, and runs at 25MHz. It has a feedforward computation rate of around 1.6 GCPS, and a learning rate of about 100 MCUPS. In the same technology, our chip would run at 400MHz, achieving 12.8GCPS and 1.6GCUPS.

Our chip achieves its performance advantage by combining deep pipelining, massive parallelism, reduced precision, and high density synaptic storage.

Large Network Design

Our investigation into large, sparse networks continued to focus on energy and power considerations. Massively parallel systems based on 3D stacked multichip modules would have excessive power densities if implemented using standard design techniques. This catalyzed the development of an energy-driven design methodology and low voltage digital logic to

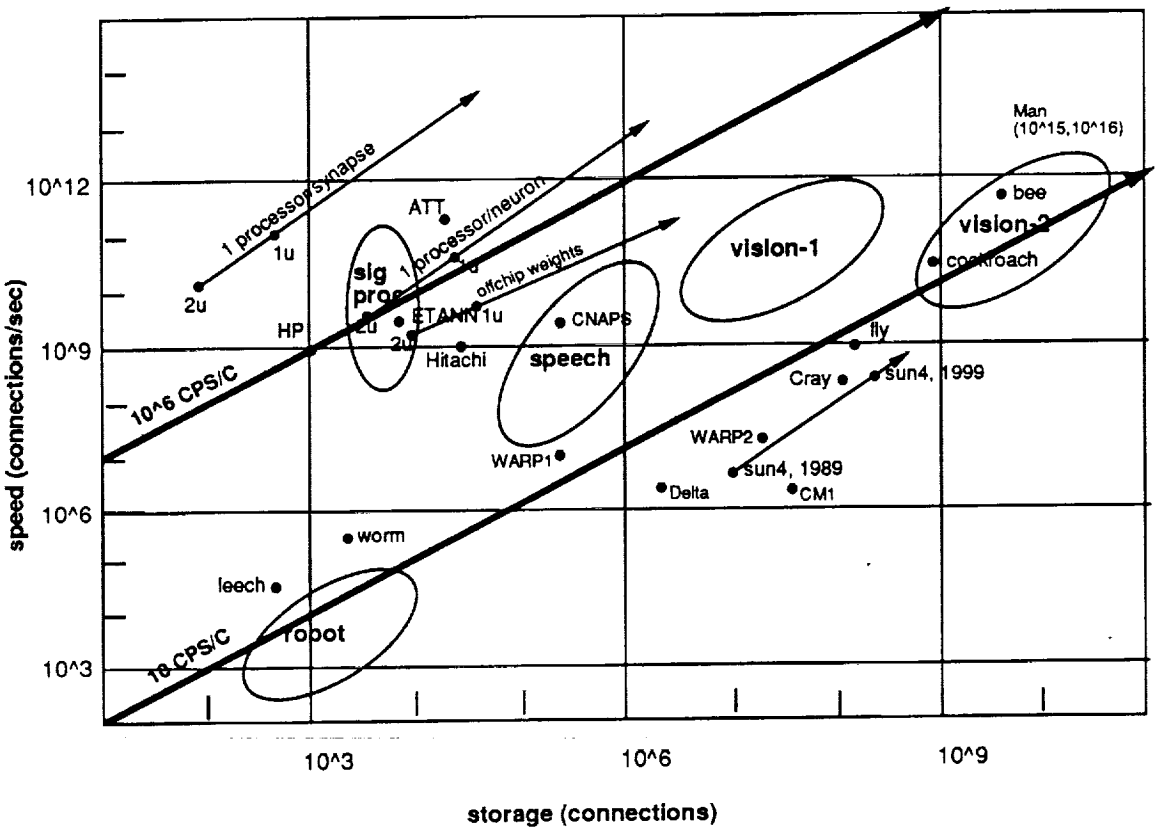


Figure 3.3: DARPA study application capacity and performance requirements. The biological trendline lies at 10 connections per second per connection (CPS/C); for signal processors, at 10^6 CPS/C.

support extremely high computation rates at modest power levels in massively parallel architectures. These observations are summarized in [BP91b, BP91a, Bur91b, Bur92], and to a lesser extent in [Bur91a].

The need for low energy is even more urgent in Irvine Sensors' Z-plane stacking technology, which can lap wafers to a thickness of 20 microns before stacking them, permitting up to 500cm^2 of silicon per cm^3, a factor of 50 denser than we had been considering.

The trendlines in Figure 3.3 suggest a wide range of possible scalable architectures. Our Boltzmann Engine supports 288,000 CPS/C.

Each neural processor in our Boltzmann Engine occupies about $500\lambda \times 1000\lambda$ and supports 180MCPS. This is enough area for about 1000 synaptic weights, so architectures with less than 10^4 CPS/C devote over 90% of the silicon area to memory. At 10 CPS/C, the memory to processor area ratio is about 10^4:1.

100CPS/C provides a good compromise between accelerated learning applications which would like 1000 CPS/C and biologically motivated feedforward applications which would like 10 CPS/C. 100 CPS/C can address both these domains with some loss in efficiency. In 0.5 micron CMOS a 1 cm^2 chip with a single processor can support 100MCPS and 1 million connections.

We have found that the most energy-efficient way to implement scalable architectures in two dimensions is to put the right balance of processors and memory on each chip, and then to tile the chips in a nearest neighbor mesh. One of the key requirements of this approach is energy-efficient interprocessor communications. The work that Yen-Wen Lu has been doing in this area is directly applicable.

The Irvine Sensors Z-plane stacking technology suggests a different mesh architecture for interprocessor communications and possibly a different partitioning of resources to take advantage of the proximity of vertical neighbors: the chip above you is only 20 microns away; the chip beside you is 1 cm away. Although we have been advocating architectures which combine processors and local memory on a single chip to reduce storage energy, it might be more efficient to place an array of processors on one chip and to stack local memory on separate chips directly above the processor array.

Conferences and Workshops

A number of people in the group were active in a number of conferences and workshops this year.

Jim Burr attended the Banff Workshop on Neural Networks Hardware in Banff in March. He also chaired the implementation section of NIPS*92. He served on the program committee of the 1992 NASA VLSI Symposium, the MASCOTS93 Workshop, and was on the Peer Review Team for the 5-year review of the NASA Idaho SERC.

Mike Murray presented the Boltzmann paper at the Application Specific Array Processors Conference in August.

Mike Leung presented a paper on texture recognition at the Asilomar Conference on Signal Processing.

Collaborative Research

The neural net research being funded by CASIS was substantially enhanced by the interaction with other projects and sponsors. Most significant of these were the Novel Devices Group and the Components Research Division of Intel Corporation for research in recognition architectures, and Stanford's own Center for Integrated Systems for work in Ultra Low Power CMOS.

Intel Neural RISC

We have been working closely with Intel on a number of issues related to the implementation of embedded neural coprocessors to perform recognition tasks. These engines need to be able to execute a variety of algorithms efficiently, including feedforward perceptron nets, elastic matching algorithms, and radial basis function networks. Intel has recognized the need for energy-efficient implementations of these functions since they are computationally intensive but have an extremely limited power budget in portable applications. They have been very supportive of our work in Ultra Low Power CMOS, providing advanced submicron fabrication facilities and engineering resources to evaluate performance and reliability of very low voltage circuits.

CIS Ultra Low Power CMOS

Stanford's Center for Integrated Systems has been funding our research in building CMOS circuits which can achieve good performance with supply voltages substantially below 1 volt. This year we have demonstrated good performance and reliability of a variety of simple circuits, and are continuing to design basic digital components for evaluation.

This work was originally motivated by the desire to approach biological efficiency with digital VLSI. The problem grew out of the challenge of exploiting the performance density available in "volume silicon" for packaging large networks. In this approach, a single neural tile would be designed and tiled on the surface of a multichip module. These modules would then be stacked to form a solid silicon structure. To achieve maximum memory density, such a system should be built from commercial DRAM and custom processors. However, this approach results in excessive communication energy because the weights are too far away from the processors. So, the minimum energy solution is to distribute the processors to the memory in a way that balances weight storage energy and activation communication energy. That is what Yen-Wen and Kong are investigating. However, once this is done, the energy density is still much too high using standard CMOS. In order to reduce the energy density to the point that heat can be extracted without exotic thermal management, the supply voltage had to be reduced substantially. However, this results in unacceptably slow circuits using standard CMOS technology. So we embarked on the Ultra Low Power CMOS project, whose objective is to reduce threshold voltages along with supply voltages so that reasonable performance can be achieved at much lower energy.

Originally motivated by our desire to build compact, high capacity, high performance neural nets, we have found widespread interest in our approach to low power, especially for space applications, terrestrial portable computers, and telecommunications equipment.

References

[AH85] David H. Ackley and Geoffrey Hinton. A learning algorithm for Bolzmann Machines. *Cognitive Science*, 9:147–169, 1985.

[Al90] R. B. Allen and J. Alspector. Learning of stable states in stochastic asymetric networks. Bellcore, 1990.

[Al90] J. Alspector. CLC—A cascadable learning chip. NIPS90 VLSI Workshop, December 1990.

[Al88] J. Alspector, B. Gupta, and R. B. Allen. Performance of a stochastic learning microchip. In *Advances in Neural Information Processing Systems*, pages 748–760, 1989.

[Baa91] Bevan Baas. A pipelined memory system for an interleaved processor. Technical report, Stanford University, September 1991.

[Bur88] James B. Burr, Advanced simulation and development techniques. In *IREE Australian Microelectronics Conference*, pages 231–238, 1988.

[BBP91] James B. Burr, James R. Burnham, and Allen M. Peterson. System-wide energy optimization in the MCM environment. In *IEEE Multichip Module Workshop*, pages 66–83, 1991.

[BP91a] James B. Burr and Allen M. Peterson. Energy considerations in multichip-module based multiprocessors. In *IEEE International Conference on Computer Design*, pages 593–600, 1991.

[BP91b] James B. Burr and Allen M. Peterson. Ultra low power CMOS technology. In *NASA VLSI Design Symposium*, pages 4.2.1–4.2.13, 1991.

[Bur91a] James B. Burr. Digital Neural Network Implementations. In P. Antognetti and V. Milutinovic, editors, *Neural Networks: Concepts, Applications, and Implementations, Volume 2*, pages 237–285. Prentice Hall, 1991.

[Bur91b] James B. Burr. Energy, capacity, and technology scaling in digital VLSI neural networks. NIPS91 VLSI Workshop, May 1991.

[Bur92] James B. Burr. Digital Neurochip Design. In K. Wojtek Przytula and Viktor K. Prasanna, editors, *Digital Parallel Implementations of Neural Networks*. Prentice Hall, 1992.

[BWP91] James B. Burr, P. Roger Williamson, and Allen M. Peterson. Low power signal processing research at Stanford. In *NASA VLSI Design Symposium*, pages 11.1.1–11.1.12, 1991.

[Gal89] C. Galland and G. Hinton. Deterministic Boltzmann learning in networks with asymetric connectivity. Technical report CRG-TR-89-6, University of Toronto, December 1989.

[1] Carl Ebeling. *The Gemini user's manual*, Carnegie-Mellon University, 1981.

[Ham90] D. Hammerstrom. A VLSI architecture for high-performance, low-cost, on-chip learning. In *IJCNN International Joiunt Conference on Neural Networks*, pages II:537–544, 1990.

[Hin89] Geoffrey E. Hinton. Deterministic Boltzmann learning performs steepest descent in weight-space. *Neural Computation*, 1:143–150, 1989.

[HSA84] G.E. Hinton, T.J. Sejnowski, and HD.H. Ackley. Bolzmann Machines: Constraint satisfaction networks that learn. Technical Report CMU-CS-84-119, Carnegie-Mellon University, May 1984.

[Lager88] University of California, Berkeley. *LagerIV distribution 1.0 silicon assembly system manual*, June 1988.

[LB87] Weiping Li and James B. Burr. An 80 MHz Multiply Accumulator. Technical report, Stanford University, September 1987.

[LBP88] Weiping Li, James B. Burr, and Allen M. Peterson. A fully parallel VLSI implementation of distributed arithmetic. In *IEEE International Symposium on Circuits and Systems*, pages 1511–1515, June 1988.

[LBP92] Yen-Wen Lu, James B. Burr, and Allen M. Peterson. Permutation on the mesh with reconfigurable bus. submitted to IPPS93, August 1992.

[Li88] Weiping Li. *The Block Z transform and applications to digital signal processing using distributed arithmetic and the Modified Fermat Number transform.* PhD thesis, Stanford University, January 1988.

[2] J. K. Ousterhout et al. *1985 VLSI tools: More works by the original artists.* University of California, Berkeley, VLSI Tools Distribution.

[MBS+92] Michael Murray, James B. Burr, David G. Stork, Ming-Tak Leung, Kan Boonyanit, and Allen M. Peterson. Scalable deterministic Boltzmann machine VLSI using multi-chip modules. In *Applications-Specific Array Processors*, 1992.

[Na75] L. W. Nagel. SPICE2: A computer program to simulate semiconductor circuits. Technical report, University of California, Berkekley. Tools Distribution.

[OP92] A.O. Ogunfunmi and A.M. Peterson. On the implementation of the frequency-domain LMS adaptive filter. *IEEE Transactions on Circuits and Systems*, 39(5):318–322, May 1992.

[PA87] C. Peterson and J.R. Anderson. A mean field theory learning algorithm for neural networks. *Complex Systems*, 1:995–1019, 1987.

[PH89] C. Peterson and E. Hartman. Explorations of the mean field theory learning algorithm. *Neural Networks*, volume 2, pages 475–494. Pergamon Press, 1989.

[Ter82] Chris J. Terman. User's guide to NET, PRESIM, and RNL/NL. Technical Report VLSI 82-112, Mass. Inst. Technology, 1982.

CASIS provides about half the research funding for our group. Several other projects round out the research effort and add strength to the neural net program. The students, their areas of investigation, and sources of support are

Amir Najmi	Dynamic programming	Intel
Yen-Wen Lu	Mesh communication architectures	Intel
Kong Kritayakirana	Mapping NNs onto meshes	Intel
Sabeer Bhatia	Ultra Low Power CMOS	Stanford CIS
Gerard Yeh	Interleaved architectures	NASA fellowship
Bevan Baas	Multiprocessor signal processing	EE Dept Fellowship
Jean-Francois Berne	Low Energy SRAM	Alcatel

The Intel-supported work is targeted toward embedded neural coprocessors for use in recognition tasks. The CIS-supported work is targeted toward very low energy computation.

3.3 Image Databases: Ramin Samadani

NASA has very large databases of image data from past and current missions. To make this vast quantity of image data useful it needs to be stored, retrieved and analyzed. Our research focuses on processing and analyzing the images to create more manageable abstractions from them. The goal is to extract information from images using tools from statistics, estimation, classification, optimization and signal processing. The information abstracted can then be used to automatically generate indeces for image databases. The extracted information can also be used as automatically generated primitives to be used as inputs to visualization processes.

Our work is closely tied to the various discipline sciences within NASA. In the past, we have had close working relationships with Goddard Space Flight Center and with the Jet Propulsion Laboratory. In order to ensure the applicability of the techniques, we test the information extraction techniques with NASA satellite data. For example, we have worked on motion extraction from SEASAT radar satellite images. Techniques related to these are incorporated in the Alaska SAR Facility for ERS-1, managed by the Jet Propulsion Laboratory. We have participated in JPL's design reviews. We have also worked on finding

and describing boundary and curvilinear features in these radar images. Finally, we have produced results on the problems of finding boundaries and regions in optical satellite images of the Aurora. This auroral work was first seeded by CASIS and then funded through the Center for Excellence in Space and Data Information Sciences.

3.4 CASIS Sponsored Publications

Kathy J. Richardson and M. J. Flynn. Strategies to Improve I/O Cache Performance. In *26th Hawaii International Conference on System Sciences.* IEEE, January 1993.

Timothy M. Pinkston, Uzi Efron, and Michael Campbell, "Optical Interconnects in the 3-D Computer for Fast Parallel Sorting," in *Proceedings of the ISMM International Conference on Parallel and Distributed Computing and Systems*, Pittsburgh, PA, pages 241–243, October, 1992.

"The GLORI Strategy for Multiprocessors: Integrating Optics into the Interconnect Architecture," *Technical Report: CSL-TR-92-552*, Stanford University, 177 pages, December, 1992.

R.A. Ziegler and J.M. Cioffi, "Estimation of Time-Varying Digital Mobile Radio Channels," *IEEE Transactions on Vehicular Technology, May 1992*, Vol 41, No. 2, pages 134–151

Robert A. Ziegler and John M. Cioffi "Estimation of Time-Varying Digital Mobile Radio Channels," *Globecom'91*, Phoenix, Arizona, December 1991, pages 1130–1134.

Naofal M. W. Al-Dhahir and John M. Cioffi, "Fast Algorithms for the Computation of the Decision Feedback Equalizer," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, San Francisco, California, March 23-26, 1992, Vol. IV, pages 533–536.

Robert A. Ziegler and John M. Cioffi, "Adaptive Equalization for Digital Wireless Data Transmission," *The Second Virginia Tech Symposium on Wireless Personal Communications*, Blacksburg, Virginia, June 17–19, 1992.

Sarah Kate Wilson and John M. Cioffi, "Equalization Techniques for Direct Sequence Code-Division Multiple Access Systems in Multipath Channels," *International Symposium on Information Theory*, San Antonio, Texas, January 17–22, 1993.

F.M. Chiussi, F.A. Tobagi, "A Hybrid Shared-Memory/Space-Divison Architecture for Large Fast Packet Switches," *Proceedings of ICC 1992*, Chicago, IL, July 1992.

Weijia Wang, "The Large Architectural Design and Performance Analysis of Large-scale Asynchronous transfer mode (ATM) Switches," Stanford University PhD thesis, August 1992.

James B. Burr. Digital Neurochip Design. In K. Wojtek Przytula and Viktor K. Prasanna, editors, *Digital Parallel Implementations of Neural Networks.* Prentice Hall, 1992.

Michael Murray, James B. Burr, David G. Stork, Ming-Tak Leung, Kan Boonyanit, and Allen M. Peterson. Scalable deterministic Boltzmann machine VLSI using multi-chip modules. In *Applications-Specific Array Processors*, 1992.

Leung, M.-T. and A. M. Peterson. "Scale and Rotation Invariant Texture Classification," Asilomar Conference on Signals, Systems, and Computers, Oct. 1992.

3.5 PhDs Awarded

Hui-ling Lou. The Study and Design of a Programmable Processor for Viterbi Detection, 1992.

Peter Okrah. Multichannel Modulation as a Technique to Combat Fading in Radio Channels, 1992.

Weijia Wang, Large Architectural Design and Performance Analysis of Large-scale Asynchronous transfer mode (ATM) Switches.