Electrical Engineering
and Computer Science

DSP Design Group — Summary of Research Activities

This document contains abstracts from the 1993 Research Summary, summarizing individual research projects. The work divides into three categories: (1) design methodology for signal processing and communications, (2) parallel computation, and (3) signal processing and communications applications. The abstracts are grouped in this order.

University of
California at Berkeley

Design Methodology for Signal Processing and Communications:

Optimized Compilation of Dataflow Programs for Multirate Digital Signal Processing

STUDENT: **Shuvra S. Bhattacharyya**

ADVISOR: **Edward A. Lee**

SPONSORS: **DARPA (J-FBI-90-073) and Motorola**

We are investigating techniques to translate block diagram programs into optimized "C" code for multirate signal processing applications. In our system, a program is composed of an interconnection of functional blocks obtained from a predefined hand-coded library. The block diagram semantics are derived from synchronous dataflow (SDF), a version of dataflow in which the number of tokens produced and consumed by each functional block is fixed and known at compile time.

In an SDF description, only a partial ordering of the computations, based on data precedences, is specified. This allows full exposure of parallelism and also does not commit to a specific control-flow structure. We are examining techniques to determine the optimal loop hierarchy for an SDF program based on considerations of program compactness (instruction count), data space compactness, and throughput. We have developed heuristics to reduce program memory and data memory requirements simultaneously [1]. We have also determined how to construct the most compact loop hierarchy for a large class of SDF graphs. Currently we are investigating the integration of throughput considerations with these methods. Such considerations include vectorization [2], parallel processor scheduling, and instruction-level parallelism.

We have also developed systematic methods for reducing the amount of data memory required for a given loop hierarchy. We are currently investigating the integration of these approaches with the scheduling process. We are implementing our techniques in the framework of Ptolemy [3], an object-oriented prototyping environment that has been developed in our group.

[1] S. Bhattacharyya and E. A. Lee, "Scheduling Synchronous Dataflow Graphs for Efficient Looping", to appear in *Journal of VLSI Signal Processing*, 1992.

[2] S. Ritz, M. Pankert, and H. Meyr, "High Level Software Synthesis for Signal Processing Systems", *Proceedings of the International Conference on Application Specific Array Processors*, Berkeley, CA, August, 1992.

[3] J. Buck, S. Ha, E. A. Lee and D. G. Messerschmitt, "Ptolemy: A Framework for Simulating and Prototyping Heterogeneous Systems", invited paper to appear in the *International Journal of Computer Simulation*, special issue on "Simulation Software Development", 1992.

The Token Flow Model: Compile-Time Scheduling of Dynamic Dataflow Graphs

STUDENT: **Joseph Buck**

ADVISORS: **D. G. Messerschmitt and E. A. Lee**

SPONSORS: **SRC (92-DC-008) and Rockwell**

Dataflow graphs have proven to be a very effective representation for problems in digital signal processing, both because the representation is natural for DSP researchers and because the representation exposes the parallelism of the algorithm and does not impose excessive constraints on the order of execution. When the actors in the graph are restricted to be synchronous (meaning that the number of tokens produced and consumed by each actor is fixed and known at "compile time"), powerful techniques exist for demonstrating the consistency of the graph, determining memory requirements, and scheduling its execution on one or multiple processing elements. These techniques are used effectively in Ptolemy and its predecessor system, Gabriel.

However, the synchronous dataflow (SDF) model is overly restrictive for digital signal processing, since some data-directed decision-making is usually required. Accordingly, the token flow model (also known as "Boolean-controlled dataflow" or BDF) admits graphs containing dynamic actors — actors in which the number of tokens produced and consumed on each arc is a function of the data values of certain Boolean tokens. This larger class of graphs is Turing-equivalent, while SDF graphs are not. New results include necessary and sufficient conditions for such graphs to have bounded-length schedules, and sufficient conditions for such a graph to be scheduled in bounded memory. Bounded cycle length is important in hard-real-time problems to assure that deadlines are always met; bounded memory permits static allocation, which is particularly important when mapping DSP algorithms that have decision-making onto VLSI hardware. A clustering technique has also been developed that, given a dataflow graph, decomposes it into traditional control structures such as if-then-else, do-while, and iteration.

Code to perform clustering and scheduling for BDF graphs is currently being implemented in Ptolemy. Future plans include extending Ptolemy's code generation facilities to permit dynamic actors, and to schedule such graphs on multiple processors.

Mixed-Domain Scheduling in Ptolemy

POSTDOC: **Soonhoi Ha**

ADVISOR: **Edward A. Lee**

SPONSOR: **AT&T Bell Labs and NEC**

The purpose of this project is to develop the general rapid prototyping environment for multiprocessor architectures in Ptolemy. We provide a suite of specialized schedulers that can be mixed and matched for specific applications. This contrasts with other efforts that seek a single, all-encompassing scheduling approach. After the scheduling is performed, each processing element is assigned a set of blocks to be executed in a scheduler-determined order. We use the code generation scheme for single processor targets to generate code, including communication routines, for each processing element. Currently, we are targeting the Sproc from Star Semiconductor, the CM5 from Thinking Machines, and DSP3 from AT&T.

We have implemented three scheduling techniques that map SDF graphs onto multiprocessors with various interconnection topologies: Hu's level-based list scheduling, Sih's dynamic level scheduling [1], and Sih's declustering scheduling [2]. For each target architecture, we define a C++ object, called Target. The Target class has several parameters for choosing a specific scheduling technique and other options. The Target class is required to provide the scheduler with the necessary information on interprocessor communication to enable both scheduling and code synthesis.

Furthermore, we have applied the idea of mixed domain scheduling to support dynamic constructs for code generation. Ptolemy defines a new domain for the dynamic constructs. By putting this new domain inside an "SDF Wormhole", the whole application can be scheduled statically. The dynamic constructs inside the SDF Wormhole change the runtime execution profile from the scheduled one. We have developed a technique that schedules dataflow graphs with run-time decision making, aiming to minimize the cost of run-time decisions [3]. We will define a specific Target class for this dynamic construct domain and generate suitable control code for the target architecture corresponding to the dynamic constructs.

[1] G. C. Sih and E. A. Lee, "Scheduling to Account for Interprocessor Communication Within Interconnection-Constrained Processor Networks," Proceedings of the International Conference of Parallel Processing, pp. 9-16, 1990.

[2] G. C. Sih, "Multiprocessor Scheduling to Account for Interprocessor Communication," Ph.D. dissertation, U.C.Berkeley, 1991.

[3] S. Ha, "Compile-Time Scheduling of Dataflow Program Graphs with Dynamic Constructs," Ph.D. dissertation, U.C.Berkeley, 1992.

Hardware/Software Codesign using Ptolemy

STUDENT: **Asawaree Kalavade**

ADVISOR: **Edward A. Lee**

SPONSOR: **SRC (92-DC-008)**

In a traditional design strategy, the hardware and software partitioning decisions are fixed at an early stage in the development cycle and the hardware and software designs are developed separately from then onwards. With advancements in technology, however, it becomes possible to obtain special-purpose hardware components (ASICs) at a reasonable cost and development time. Some designs also call for some programmability in the end product. This suggests a more flexible design strategy, where the hardware and the software designs proceed in parallel, with feedback and interaction between the two as the design progresses. The final hardware/software split can then be made after the evaluation of alternative structures with respect to performance, programmability, cost, and performance. This design philosophy, which helps reduce the time to market, is called hardware/software codesign.

We are studying the issues involved with the development of such a unified framework for the design of multiprocessor DSP systems, where the hardware consists of discrete components, ASICs, DSP cores, microprocessors, microsequencers, microcontrollers, or semi-custom logic developed using FPGAs or logic synthesis tools, and the software is the program running on the programmable processors. Towards this end, we are using the Ptolemy design environment. The long-term goal is directed towards systematizing the implementation of such a complex hardware/software mixture.

The feasibility of such codesign attempts under Ptolemy was ascertained with the help of a case study that involved the design of a telephone channel simulator [1]. Using the synchronous dataflow (SDF) and hardware simulation (Thor) domains of Ptolemy, a number of design options for this example were developed where different multiprocessor configurations and hardware/software partitions were explored. Assembly code corresponding to the algorithm was developed using the re-targetable code generation facilities. These designs were then evaluated with respect to cost/performance tradeoffs within the given environment.

Work is underway towards developing a "Silage" domain, which is a code generation domain for generating Silage (an applicative programming language for DSP applications) code. Silage simulations can be used for finite word-length analysis, and the generated Silage code can then be used for synthesis of custom hardware using pre-existing synthesis tools.

[1] Asawaree Kalavade, "Hardware/Software Design Using Ptolemy - A Case Study", Master's Report, EECS Dept., UC Berkeley, 1991.

Graphical Specification of Real-Time Control Flow

PROFESSOR: **Edward A. Lee**

SPONSOR: **Sony Corporation**

In the last few years, we have made considerable progress with graphical specification of signal processing systems. We are now extending this work to encompass real-time dynamic control flow for embedded systems. This enables specification of sophisticated user interaction, system adaptability, and real-time response to dynamic events. A key part of this effort is the development of a domain in Ptolemy with finite state machine (FSM) semantics. Such a domain would nicely complement our dataflow and discrete-event domains for system specification and design. We are currently exploring three approaches, one based on "statecharts" [1] (like the Statemate system from iLogix), another based on "speccharts" [2] (a variant from UC Irvine), and a third that combines straight FSMs with the hierarchical heterogeneity naturally supported by Ptolemy [3].

[1] D. Harel, "Statecharts: A Visual Formalism for Complex Systems," *Science of Computer Programming*, Vol. 8, No. 5, pp. 231-274, 1987.

[2] S. Narayan, F. Vahid, D. D. Gajski, "SpecCharts: A Language for System Level Specification and Synthesis", Proc. of Int. Symp. on Computer Hardware Description Languages, Marseille, April 1991.

[3] J. Buck, S. Ha, E. A. Lee and D. G. Messerschmitt, "Ptolemy: A Framework for Simulating and Prototyping Heterogeneous Systems", invited paper to appear in the *International Journal of Computer Simulation*, special issue on "Simulation Software Development", 1992

Representing and Exploiting Data Parallelism in Graphical Signal Flow Representations of Algorithms

PROFESSOR: **Edward A. Lee**

SPONSOR: **Hitachi**

Signal flow graphs with dataflow semantics have been used in signal processing system simulation, algorithm development, and real-time system design. Dataflow semantics implicitly expose function parallelism by imposing only a partial ordering constraint on the execution of functions. They are also capable of representing data parallelism. We are investigating how the synchronous dataflow model can be used to graphically define algorithms while exposing their data parallelism to a compiler. A "recursive iterator" notation has been introduced to achieve scalable graphical representations for certain algorithms. The SDF model has also been extended to multidimensional streams to represent and exploit data parallelism in certain signal processing applications. The method appears to be by no means general, but to have broad enough applicability to be useful. The resulting semantics are related to reduced dependence graphs used in systolic array design [1] and to multidimensional streams in the declarative language Lucid [2]. It is more distantly related the stream-oriented functional language Silage [3], and streams in the dataflow language Id and the "synchronous" languages Lustre [4] and Signal [5].

[1] S. Y. Kung, *VLSI Array Processors*, Prentice-Hall, Englewood Cliffs, New Jersey, 1988.

[2] E. A. Ashcroft and W. W. Wadge, "Lucid, a Nonprocedural Language with Iteration", *Comm. of the ACM*, Vol. 20, No. 7, pp. 519-526, July 1977.

[3] P. N. Hilfinger, "Silage Reference Manual, Draft Release 2.0", Computer Science Division, EECS Dept., University of California, Berkeley, CA 94720, July 8, 1989.

[4] P. Caspi, D. Pilaud, N. Halbwachs, and J. A. Plaice, "Lustre: A Declarative Language for Programming Synchronous Systems", *Conference Record of the 14th Annual ACM Symposium on Principles of Programming Languages*, Munich, Germany, January, 1987.

[5] A. Benveniste, B. Le Goff, and P. Le Guernic, "Hybrid Dynamical Systems Theory and the Language SIGNAL", Research Report No. 838, Institut National de Recherche en Informatique at en Automatique (INRIA), Domain de Voluceau, Rocquencourt, B. P. 105, 78153 Le Chesnay Cedex, France, April 1988.

Real-Time Scheduling of Dataflow Programs

STUDENT: **Thomas M. Parks**

ADVISOR: **Edward A. Lee**

SPONSORS: **SRC (92-DC-008) and Bell Northern Research**

The purpose of this project is to apply real-time scheduling techniques to programs represented by dataflow graphs. During the past year, development of a prototype real-time dataflow (RTDF) domain under Ptolemy has continued. This domain uses preemptive scheduling of lightweight processes to simulate RTDF systems. In addition to this simulation domain, a code generation domain is also being developed. In this domain, code is generated in the C programming language from a dataflow graph description. The resulting code can be run on the “real-time workstation”, a single-board computer running a real-time operating system which provides a link between a non-real-time Unix workstation environment and hard-real-time signal processing systems.

Both of these domains rely on prioritized preemptive scheduling of real-time tasks. Future work on this project will focus on minimizing or avoiding the overhead of preemptive scheduling. This overhead can be minimized by reducing the number of tasks in a system, and it can be avoided by making use of static compile-time scheduling and non-preemptive run-time scheduling wherever possible.

Optimized Code Generation for DSPs

STUDENT: **José Pino**

ADVISOR: **Edward A. Lee**

SPONSOR: **ONR Fellowship, AT&T Bell Labs Fellowship, and Motorola**

The goal of this project is to create a new domain in Ptolemy that synthesizes efficient assembly code for programmable digital signal processors (DSPs). In this domain the user will be able to target various architectures, including stand-alone DSPs, workstations, and DSP/

workstation combinations. Direct assembly code generation for these processors is currently possible using Gabriel, the predecessor to Ptolemy. However, the code generated by Gabriel has too much overhead for many applications. This is particularly true in code constructed from fine-grain dataflow graphs.

Code generation in Ptolemy will incorporate architecture specific knowledge, such as hardware looping and the number of registers. This information will be used to eliminate excess transfers to and from memory. Architecture-specific knowledge will also be used to balance the use of loops and in-line code to efficiently use program memory while not sacrificing speed. This is particularly challenging in multirate systems.

During the course of this past year, all of Gabriel's assembly code generation functionality has been ported into Ptolemy. Some refinements of Ptolemy include proper interrupt handling, a more abstract assembly code generation model facilitating support for multiple DSP platforms, and a more robust loop scheduler.

The major focus for the coming year is to optimize the assembly code generated by Ptolemy with special attention given to buffer management and dynamic register allocation. Another goal will be to automate the code generation for host to DSP hardware interaction. A major focus will be to preserve the object-oriented style of Ptolemy simulations.

Advanced Filter Design

STUDENT: **Kennard White**

ADVISOR: **Edward A. Lee**

SPONSORS: **MICRO, Dolby Labs, and Comdisco Systems**

XPole is a graphical filter design program. It allows users to construct filters by manipulating singularities (poles and zeros) in the Z or S plane, while the program interactively computes the impulse and frequency responses. Standard filters (e.g., Butterworth, elliptical) of arbitrary order are supported, as well as several mappings from S to Z transform domain. XPole is intended to give users an intuitive understanding of the relationship between time-, frequency-, and transform-domain representations of filters. XPole is currently being integrated into the

Ptolemy environment. Future plans include support of root-locus and Nyquist plots, analysis of implementation and quantization effects, design of statistically derived filters, and support of frequency-based filter design algorithms. X11 with Tk is used for the graphical interface, and Tcl for the command and script language.

DSP Hardware Implementation Design through VHDL in the Ptolemy Environment

STUDENT: **Michael C. Williamson**

ADVISOR: **Edward A. Lee**

SPONSORS: **SRC (92-DC-008) and DoD National Defense Science and Engineering Graduate Fellowship (DoD NDSEG)**

The objective of this project is to provide a clearer design path between high-level DSP algorithm design and design of the digital hardware implementation. The Ptolemy design environment provides an excellent base for DSP algorithm design, but has only rudimentary hardware design capability. The Thor domain, while able to simulate digital hardware at a low level, does not easily allow for intermediate behavioral descriptions of components whose detailed functionality has yet to be specified. The VHDL standard for hardware description and simulation can serve as a bridge between abstract algorithmic descriptions and detailed low-level hardware designs

There are two ways in which a VHDL domain in Ptolemy could facilitate the transition between algorithm and implementation. The first is to incorporate an appropriate subset of VHDL into a VHDL scheduler, which would be similar to the schedulers in the discrete event or Thor domains. Block specifications of DSP algorithms in the synchronous dataflow and dynamic dataflow domains could be gradually expanded and substituted block by block with functionally equivalent blocks created in the VHDL domain in order to validate hardware implementation of the original algorithm and to experiment with various hardware design alternatives.

The second way is to take DSP designs which are specified in the VHDL domain and have Ptolemy generate VHDL descriptions of those designs, comparable to the way code-generation domains produce output source code that can be executed on various supported processors. These VHDL descriptions can then be passed to synthesis tools that generate gate-level circuit implementations. These two methods are being evaluated with the objective of designing a VHDL domain in Ptolemy.

Parallel Computation:

MIMD Synchronization and Coherence

STUDENT: **Philip Bitar**

ADVISOR: **E. A. Lee**

SPONSOR: **Hitachi Ltd. and Hitachi America**

We address the topic of architecture for dynamically synchronized multiprocessor computation. In this type of computation, synchronization and coherence cannot be completely resolved at compile time due to unpredictability in time of access by a process to an atom (a shared writable object). Hence, a major concern is to implement efficient synchronization and coherence at runtime.

We formulate a conceptual model for integrating the domain of synchronization and coherence concepts for a dynamically synchronized system. Our model applies to shared-memory architecture and message-passing architecture, multiprocessor or distributed. We also solve several outstanding problems in this domain. More specifically, our research entails the following topics and developments.

1. The Foundation. We identify the defining difference between shared-memory and message-passing architecture, and draw forth its implications.
2. Synchronization. We identify and organize the fundamental synchronization concepts, which provide the basis for formulating access order, combining trees, and coherence.
3. Access Order. In execution of a multiprocessor program, out-of-order memory access to atoms may obtain incorrect results. We reconceptualize this access-order problem, and we present the weakest possible solution to the problem.
4. Combining Trees. A combining tree is a parallel computation tree in which a parent combines requests from its children. For example, a parent may combine two requests “add R1 to X” and “add R2 to X” into “add R1+R2 to X”. A combining tree is necessary if the processor request rate to a memory location X exceeds the service rate that the memory system can provide. We analyze the motivation, structure,

and management of combining trees. We develop the concept of combining window, showing that temporal proximity of requests must be ensured.

5. Coherence. The problem of coherence arises if a writable object is shared among memory units, because a write to one copy must be reflected in the other copies. We formulate basic coherence options, and relate them to synchronization methods and to the access-order solution.
6. Trace-Driven Simulation. We identify the serious validity issues that multiprocessor trace-driven simulation faces.
7. Event-Driven Simulation. In order to evaluate tradeoffs that are present in the foregoing topics, we are developing discrete-event simulation software in Ptolemy, a schematic-based simulation system written in C++. For our dissertation, we are focusing on the topic of combining trees, and we are evaluating tradeoffs that are present in the choice of combining window discipline.

[1] Bitar, P., Despain, A.M. 1986. “Multiprocessor cache synchronization: issues, innovations, evolution.” *13th Int’l. Symposium on Computer Architecture*, 1986, 424-433.

[2] Bitar, P. 1989. “A critique of trace-driven simulation for shared-memory multiprocessors.” Presented at architecture workshop in 1989. In Dubois, M., Thakkar, S. (Eds.), C*ache and Interconnect Architectures in Multiprocessors*, Kluwer Academic Publishers, Norwell, Mass., 1990.

[3] Bitar, P. 1990. “MIMD synchronization and coherence.” Tech. report UCB/CSD 90/605, Computer Science Division, U.C. Berkeley, November 1990, Version 92/03/26.

[4] Bitar, P. 1990. “Combining windows: The key to managing MIMD combining trees.” Presented at architecture workshop in 1990. In Dubois, M., Thakkar, S. (Eds.), *Scalable Shared Memory Multiprocessors*, Kluwer Academic Publishers, Norwell, Mass., 1992.

[5] Bitar, P. 1992. “The weakest memory-access order.” *J. of Parallel & Distributed Computin*g, 15, August 1992, 305-331.

Heterogeneous Signal Processing Systems

STUDENT: **Praveen Murthy**

ADVISOR: **Edward A. Lee**

SPONSORS: **MICRO and Star Semiconductor**

In this project, a new domain in Ptolemy has been created which targets the Sproc chip made by Star Semiconductor. The Sproc is a multiprocessor DSP architecture which is unique in that a central memory is shared through time division multiplexing among four processors on the same die. Programming is via block diagrams with semantics closely related dataflow graphs.

The Star scheduling method is to temporally partition blocks among processors where block A will run on processor 1 for sample 1 and run on processor 2 for sample 2 and so on. Semaphores are used to ensure that no two processors simultaneously execute the same block. While this partitioning scheme is attractive for many DSP applications, it is not well suited for multirate graphs. By creating a new code-generation domain in Ptolemy, we are able to evaluate the effectiveness of other multiprocessor scheduling schemes such as Soonhoi Ha's quasi-static scheduler. The Sproc domain thus allows the user to automatically schedule and partition a ptolemy universe onto the Sproc chip using any of a number of schedulers developed by the Ptolemy group, and run the application in real time. Processor synchronization is achieved by using the Sproc's "trigger bus" mechanism which allows communication to occur without interrupts.

One of the challenges in creating this domain has been to maintain the object- oriented style of the Ptolemy program. Thus, the main target is a parent target which spawns a child target for each processor in the architecture. This approach ensures that arbitrary multiprocessor topologies can be easily created using single processor targets that already exist.

We are also examining a new real-time scheduling scheme that appears promising for scheduling multirate graphs. In this scheme, the graph is divided into timezones where each timezone consists of blocks operating at the same rate. A list of these timezones is created and each timezone is executed sequentially if blocks in the timezone are fireable. A counter is maintained at the inputs of each block to indicate whether enough data for that block is available. This scheme ensures that for real-time blocks like A/D converters, considerably less buffering is required than other schemes (such as loop scheduling). The drawback is that some scheduling decisions have to be made at runtime.

Real Time Video Signal Processing in Ptolemy

STUDENT: **Sun-Inn Shih**

ADVISOR: **Edward A. Lee**

SPONSOR: **DARPA (J-FBI-90-073) and Philips**

A programmable video signal processor (VSP), developed in the Philips Research Laboratories, can perform a broad range of video signal processing tasks in real time. The architecture of each processor is extensively parallel, and processors can be interconnected in large networks. The processing elements (PEs) operate in parallel at the rate of 27MHz. Development tools are provided for software development and hardware simulation of the VSP network.

The short-term objective of this project is to integrate the VSP tools into Ptolemy. The long term objective is to study and develop implementation methodologies for real-time video signal processing. Ptolemy's support for heterogeneity will provide an environment for studying the interaction of video algorithms and other signal processing tasks (for instance, signal transmission and network control).

Our initial approach is to define a VSP domain with synchronous dataflow (SDF) semantics. A block library will be developed that is compatible with the block library in existing Philips tools, enabling us to do functional verification by local simulation of the algorithm. A code generation mechanism will then couple this domain to existing VSP software development tools, which will be responsible for scheduling and code generation. In addition to existing SDF simulation targets, the following Targets will be implemented:

A Hardware Simulation Target will have a newly written scheduler that first invokes the SDF scheduler to compute the repetitions rate of each of the blocks, and then simply walks through the topology of the system and generates a file in Philips' intermediate ASCII netlist format. After generating this file, the Target invokes the existing VSP hardware simulator. This target can be used to debug applications, evaluate schedulers, and measure real-time performance.

A Hardware Target will consist of a VSP system connected to a workstation and associated video hardware.

A Hardware Emulator within the Thor or VHDL hardware simulation domains will be used to design application-specific configurations of VSPs. This will then be used as infrastructure for developing hardware/software codesign methodologies and evaluating retargetable scheduling techniques.

Restricted Application Parallel Architecture Design

STUDENT: **S. Sriram**

ADVISOR: **E. A. Lee**

SPONSOR: **SRC (92-DC-008) and Motorola**

In this project, we are investigating special purpose parallel architectures for real time signal processing. A systematic, low-cost method for communication between programmable components has been devised. This method is compatible with the assembly and microcode synthesis techniques in Gabriel and with the hardware/software codesign approach in Ptolemy.

Gabriel allows only a subclass of DSP applications that can be expressed as a synchronous dataflow (SDF) graph. A compiler can automatically partition SDF graphs into tasks running on multiple processors and compute near optimal schedules for them. One consequence of this approach is that the ordering of transactions between the processors is known, or can be estimated with reasonable accuracy, at compile time. By determining such an order at compile time and enforcing it at run time, contention for shared resources is eliminated, and there is no need for explicit synchronization between processors.

Based on these ideas, we have designed and built a prototype multiprocessor design called the “ordered memory access” (OMA) architecture, comprised of four Motorola DSP96002 processors on a single printed circuit board. The processors communicate via shared memory. At runtime, processor accesses to shared memory are ordered by a transaction controller. The controller, implemented on a Xilinx FPGA, grants access to shared memory according to the schedule predetermined and downloaded into it at compile time. By ordering processor transactions in this manner, we have been able to achieve low overhead and low latency interprocessor communication with very little added hardware cost. Some music synthesis programs have been run and tested on the board. We are performing detailed benchmarking at present.

We are also exploring ways of extending these ideas to applications that involve some dynamics in their flow of control, and hence are not wholly SDF. The prototype architecture allows a processor to force the transaction controller to switch between schedules on the basis of runtime conditions. This will enable us to execute applications that fall into the dynamic dataflow (DDF) or Boolean dataflow (BDF) domains in Ptolemy.

[1] J. C. Bier, S. Sriram, and E. A. Lee, "A Class of Multiprocessor Architectures for DSP", *VLSI DSP IV*, IEEE Press, 1990.

[2] E. A. Lee and J. C. Bier, "Architectures for Statically Scheduled Dataflow", *Journal of Parallel and Distributed Computing*, December 1990.

Synthesis of Parallel Code for the CM-5

STUDENT: **Kennard White**

POSTDOC: **Soonhoi Ha**

ADVISOR: **Edward A. Lee**

SPONSOR: **NSF (MIP-9201605), MICRO, NSF Infrastructure Grant number CDA-8722788.**

We have developed the CM-5 from Thinking Machines Inc. as a target for Ptolemy's multiprocessor code generation. The CM-5's processing nodes (Sparc processors with dedicated network interfaces) match well with Ptolemy's large grain parallelization approach. Ptolemy will map an application fitting the synchronous dataflow (SDF) model of computation onto the CM-5's processing nodes and schedule the computation and interprocessor communication (IPC). A variety of different algorithms may be used to perform the mapping and scheduling. Ptolemy generates C code for the computation and IPC; this code is then compiled and executed. The IPC is performed using Active Messages (CMAM) from Prof. Culler's group [1].

The CM-5 target provides a good test bed for evaluating mapping and scheduling algorithms; in particular, we can evaluate how closely the models used by the scheduling algorithms match reality. Future plans includes general research of parallel scheduling algorithms, research into fine grain parallelization, and CM-5 specific research on alternative IPC protocols and finding weaker IPC synchronization methods.

[1] T. von Eicken, D. E. Culler, and S. C. Goldstein, and K. E. Schauser, "Active messages: a mechanism for integrated communications and computation," *Proc. of the 19th Int. Symp. on Computer Architecture*, Gold Coast, Australia, May 1992, also available as TR UCB/CSD 92/675, CS Division, University of California, Berkeley, CA 94720.

Signal Processing and Communications Applications:

Wireless Optical Communications for Local-Area Networks

STUDENT: **John Barry**

ADVISORS: **J. M. Kahn, E. A. Lee, and D. G. Messerschmitt**

SPONSOR: **NSF (MIP-8657523) and Sony**

The trend in computer technology is towards smaller yet more powerful computers, while the trend in telephony is towards mobile personal communication systems. It appears that the two trends are merging, as there is a growing interest in wireless data communication systems. The day may come when users of portable computers can access the same high-speed services offered to computers on wired networks.

We propose a wireless local-area network for portable computers based on nondirectional infrared communication [1]. The network consists of a wired backbone of base stations, with one or more base stations assigned to each cell. The base stations, which are fixed on the ceiling, fill the room with infrared light by transmitting a broad optical beam. The receivers have a wide field of view. The up link has similar characteristics but operates at a different wavelength. The technical challenge faced by the designer of such a network is to ensure a robust link while meeting the stringent cost, weight, and power-consumption constraints of a portable computer.

Our work to date has concentrated on the physical layer. We have characterized the primary impediments to communicating at high speeds — noise from ambient light and multipath dispersion — and proposed design strategies to counter them. We have developed a procedure for computing the multipath impulse response of indoor optical channels, accounting for multiple reflections of arbitrary order. The accuracy of this method has been verified experimentally [2]. We have performed a joint optimization of the optical transmitter, optical antenna and optical filter. We propose a hemispherical thin-film optical filter to achieve simultaneously a narrow bandwidth and wide field of view. We have developed a PIN-FET receiver design procedure which minimizes the required photodetector area while meeting the constraints imposed by bandwidth limitations, power consumption limitations, and signal-to-noise ratio requirements. The results show that speeds of 100 Mb/s or higher should be achievable.

An experimental prototype is being developed to demonstrate the practicality of wireless communication at 100 Mb/s [3]. In future work we will examine the effectiveness of distributed sources, such as a ceiling illuminated with a broad optical beam, to mitigate shadowing. We will also examine the advantages of photodetector arrays in shadowing and tilting environments. Finally, we will address the interaction between physical-layer issues and higher-level issues such as multiple access and cell hand-off.

[1] J. Barry, J. Kahn, E. Lee, D. Messerschmitt, "High-Speed Non-Directive Optical Communication for Wireless Networks," IEEE Network Magazine, November 1991, pp. 44-54.

[2] J. Barry, J. Kahn, W. Krause, E. Lee, D. Messerschmitt, "Simulation of Multipath Impulse Response for Indoor Wireless Optical Channels," accepted in IEEE Journal of Selected Areas in Communications, June 1992.

[3] See research abstract for Gene Marsh.

Capacity Penalty due to Ideal Zero-Forcing Decision-Feedback Equalization

STUDENT: **John Barry**

ADVISORS: **E. A. Lee and D. G. Messerschmitt**

SPONSOR: **NSF (MIP-8657523) and Sony**

Equalizers are used in digital communication systems because they make it easier for the receiver to recover the transmitted signal when the intervening channel induces temporal dispersion. In the process, some equalizers destroy information and hence cause a reduction in Shannon capacity. We examine the capacity penalty due to tail-canceling equalizers, a widely used class of equalizers that includes zero-forcing decision-feedback equalizers and Tomlinson-Harashima precoding systems.

We compare the capacity C_{ZF} of an idealized zero-forcing decision-feedback equalization (ZF-DFE) system to the capacity C of the underlying channel. We find that, for strictly bandlimited Gaussian-noise channels, the capacity penalty gets vanishingly small as the signal power constraint increases. For non-strictly bandlimited channels, on the other hand, the capacity difference $C - C_{ZF}$ grows without bound as the power constraint increases, and the asymptotic ratio C_{ZF}/C is as low as 93.6% for some channels. We thus conclude that ZF-DFE on loosely bandlimited channels is a suboptimal, information-lossy equalization technique, even in the limit of infinite signal power.

In a related work, Price [1] showed that, for a linear Gaussian-noise channel with input power constraint P and capacity $C(P)$, an uncoded PAM system using ZF-DFE with bit rate equal to $C(P)$ can achieve a 10^{-5} error rate when the average transmit power is about 8 dB larger than P. Many researchers have misinterpreted Price's result to mean that capacity can be achieved with ZF-DFE at high signal-to-noise ratios. Our results, however, show this to be true for strictly bandlimited channels only.

The tightness of C_{ZF} as an upper bound to achievable bit rates using a practical ZF-DFE system is still an open problem; a complete analysis should take into account the effects of error propagation, decoding delay, and finite symbol alphabet, all of which separate the ideal ZF-DFE from a practical one. In addition, the capacity penalty, if any, due to ideal minimum-MSE DFE has not yet been characterized.

[1] R. Price, "Nonlinearly Feedback-Equalized PAM versus Capacity for Noisy Filter Channels," Proc. ICC '72, pp. 22-12 — 22-17 (June 1972).

[2] J. R. Barry, E. A. Lee, and D. G. Messerschmitt, "Capacity Penalty due to Ideal Zero-Forcing Decision-Feedback Equalization," submitted to ICC '93, Geneva.

Ptolemy Simulation Tool for Mobile and Personal Radio

STUDENT: **John Davis**

ADVISOR: **Jean-Paul Linnartz**

SPONSOR: **NSF Minority Graduate Fellowship and Sony**

The design of mobile and personal communication systems requires consideration of various systems aspects, including for instance propagation impairments, modulation techniques, carrier synchronization, coding and decoding and access protocols. The simulation package Ptolemy provides a flexible and extensive platform for simulation of such systems. This project addresses the development of a Ptolemy library consisting of characteristic radio channels and appropriate spatial random-access schemes. Together with existing tools, such as an extensive set of building blocks for the simulation of Digital Signal Processing circuits, this offers a powerful facility for simulation of radio communication systems. This extension of Ptolemy is believed to yield a useful and powerful tool for the design and development of mobile and personal communication systems.

We use the tools being developed to study the performance of a digital cellular radio network using time division multiple access (TDMA). If the received signal-to-joint-interference ratio falls below a specified level due to multipath fading and shadowing of the desired signal during block transmission, an “outage” occurs. In our research we simulate the probability of block outage considering the combined shadowing and narrowband multipath fading. Our simulation takes into account the distribution of the average non-fade duration with respect to the duration of voice segments or data block. Previous studies always assumed that voice segments are short compared to the average duration of fades and non-fade periods. Our extended model and its results are relevant to the design and planning of digital cellular telephone networks and cellular data networks.

Dynamic Assignment of Channel Resources in Multi-User Radio Networks

STUDENT: **Rolando Diesta**

ADVISOR: **Jean-Paul Linnartz**

SPONSOR: **SRC (92-DC-008)**

In radio data networks, efficient use of the available spectrum ensures high user capacity and low access delays for data messages. Previous studies of multiple-user wireless data networks mostly split the access strategy into a fixed cellular frequency reuse pattern and a dynamic multiple access scheme. We are pursuing the combined investigation of random access and spatial frequency reuse as a first step in the development of "spatial random access" methods for wireless data networks with mixed traffic patterns and various channel characteristics. In our analysis, we distinguish optimum strategies for the outbound and the inbound channel. In the outbound channel, transmissions can be coordinated between (spatially distributed) base stations to minimize harmful co-channel interference. For the inbound channel, we are evaluating collision-type random-access schemes with dynamic frequency reuse patterns.

The probability of successful reception (capture) given a certain interference situation has been studied previously for narrowband communication. Combined with specific data traffic characteristics, these conditional probabilities can be used to study the performance in terms of throughput per Hz per km2. We wish to estimate the achievable spectrum efficiency and spatial density of transmissions for the theoretical case that the position of each receiving terminal is known in advance and the characteristics of all propagation paths, including the path over which interfering signals travel, are known. A mobility model will also be considered, relaxing the assumption that the network always exactly knows in which cell a mobile terminal is located. Ptolemy will be used to simulate the developed protocols in comparison with more common protocols such as slotted ALOHA with fixed cellular frequency re-use.

A Toolkit for Real-time Physical Modeling Music Synthesis

STUDENT: **Brian Link**

ADVISORS: **David Wessel and Edward A. Lee**

SPONSOR: **Pauline Oliveros Foundation, Apple Computer (Grant INV 9004-011)**

A toolkit for real-time physical modeling synthesis and signal processing has been developed. The toolkit is designed to be both an environment for experimenting with new physical models and a musical instrument with real-time response. Real and imagined acoustic instruments and environments are modeled using digital waveguide filter networks. The toolkit runs on the Reson8, a parallel DSP architecture using eight Motorola 56001 chips. The kernel of the toolkit is hand coded in DSP 56k assembly language to optimize time-critical computations.

Max, a Macintosh application, is used to provide a flexible, real-time environment including event processing, DSP control, and a graphical user interface. This interface allows real-time control of many parameters and provides real-time filter network reconfiguration capabilities.

The toolkit provides composers with a fast way to construct physical or quasi-physical models. It takes advantage of the inherent modularity of waveguides to allow flexible construction of complex systems. Julius Smith and others have shown that waveguides permit good models of instruments with intuitive control parameters.

Overcoming Voice-Packet Loss in an ATM Network

STUDENT: **Gregory Walter**

ADVISOR: **Edward A. Lee**

SPONSOR: **NSF (MIP-9201605) and Bell Northern Research**

With the introduction of asynchronous transfer mode (ATM) networks, there will be a greater need for understanding the impact of replacing circuit-switched voice with packet-switched voice. In a packet-switched network, such as ATM, methodologies must be devised for dealing with voice packets never arriving or arriving too late at their destination. This problem is intensified when one considers the numerous ways speech may be encoded before being sent over the network. It is incorrect to assume that a packet-recovery method for one type of speech coding will be suitable for another type of speech coding. The goal of this project is to apply advanced voice-packet recovery techniques to an ATM network simulation using the Ptolemy design environment. Implementation will focus on 64 kbit/sec PCM and 32 kbit/sec ADPCM.

There are several voice-packet recovery methods available with 64 kbit/sec encoded speech. The most obvious, and simplest, involves substituting an all-zero packet where it is discovered a voice packet has been lost. Studies have shown, however, that only a 1% packet-loss rate is tolerable using this method. Another method, waveform substitution, has several variations: 1) replacing a missing packet with another packet; 2) using previously recovered samples to scan other packets for a probable match in sample sequence; and 3) duplicating the most recently received P ms of speech enough times to fill the missing packet, where P is an estimate of the pitch period. Two other methods include sample interpolation, where even and odd samples are packetized separately, and least significant bit (LSB) dropping which packetizes the LSBs of a sample separately from the most significant bits. When congestion occurs on the network, the LSB packets are the first to be discarded.

In adaptive coding techniques such as 32 kbit/sec ADPCM, many of the methods mentioned above are not applicable. For example, a simple waveform substitution technique will cause mistracking to occur between the coder and decoder due to the fact that samples in ADPCM are determined from previously coded samples, not simply by the currently coded sample. The most successful way of dealing with mistracking is to use an embedded ADPCM coding scheme and LSB dropping during congestion. The embedded ADPCM determines its coded samples with only the knowledge of the most significant bits of previous samples. Thus, if congestion occurs on the network, a lost LSB packet will not cause mistracking between the coder and decoder.

[1] D. Goodman, G. Lockhart, O. Wasem, W. Wong, “Waveform substitution techniques for recovering missing speech segments in packet voice communications,” *IEEE Transactions on ASSP*, vol. ASSP-34, no. 6, pp. 1440 - 1448, Dec 1986.

[2] J. Suzuki and M. Taka, “Missing packet recovery techniques for low-bit-rate coded speech,” *IEEE J. on Selected Areas in Communications*, vol. 7, no. 5, pp. 707 - 717, June 1989.