VLSI PARALLEL PROCESSING FOR MUSICAL SOUND SYNTHESIS

John Wawrzynek
Thorsten von Eicken
Computer Science Division, EECS Department
University of California, Berkeley
Berkeley, CA 94720

ABSTRACT: In this paper, we present the architecture of a new musical sound synthesis machine. The machine is a parallel computer architecture designed specifically for the task of generating musical sounds based on simulating the physical behavior of musical instruments. It will be compact, inexpensive, and easily scalable, offering much higher performance per unit board area than would be possible using commercially available processors. Previous generation machines demonstrate that single-chip systems cannot offer sufficient performance to simulate ensembles of instruments. To overcome this problem our system includes a special purpose communication network managed by an on-chip unit. In addition, special processing is provided for calculations involving delay lines and table lookup operations.

Introduction

The goal of the MIMIC project is to develop a parallel computer architecture specifically for the task of generating musical sounds based on simulating the physical behavior of musical instruments. The machine will be compact, inexpensive, and easily scaled up or down in size and in performance to meet a variety of applications. A primary goal of the project is to design a system that achieves a much higher performance per board area than would be possible using commercially available processors.

This project continues work on custom architectures for music synthesis previously completed at Caltech [1]. That work provided promising results and validated the basic approach but was limited in its performance because no provisions for parallelism at the system (multi-chip) level were made. The MIMIC project addresses the issue of inter-chip communication in multi-chip systems and, at the same time, the basic processing function of each chip is extended to allow more general modeling techniques [2].

We have designed a multiple-instruction-stream multiple-data-stream (MIMD) multi-processor comprising identical pipelined processing chips connected via a network. Each chip contains a 20H ALU, 64Kbit of RAM, a network interface, and a special external DRAM (dynamic random access memory) interface supporting operations using digital delay lines and table lookup.

The machine is designed as a multi-processor and may be scaled to fulfill a range of performance goals and applications by adjusting the total number of processing nodes in the system. A standard workstation will be used as a host to the machine and serve the needs of application development and real-time user interaction.

Communication among processors is bi-serial. These bi-serial connections reduce the number of pins on each chip and consequently allow the use of smaller packages, achieving higher board density than would otherwise be possible.

A low-end configuration (see Figure 1) will include 27 processing nodes (in a 3 by 3 by 3 node cube), each computing about 20 million operations per second (MOPS) where each operation is a multiply-add and the entire system about 540 MOPS. Not counted are the other simultaneous activities on the chip which manage signal flow through the network, delay lines, and table lookup operations. In our applications the performance of
the system will come very close to these maximum bounds. A system of this size is sufficient for performing a realistic simulation of a concert grand piano or a small musical ensemble. A larger system of about 100 nodes on a board could implement an "orchestra in a box."

The processing node is designed to support a few basic primitives that form the computational elements of a particular class of sound synthesis algorithms. Figure 2 shows the set of computational primitives supported by our machine. These primitives form the lowest-level user interface to the machine. Computation graphs consisting of these primitives are automatically mapped by software to instructions and routing information for the MIMIC machine.

Model of Operation

Because the machine will simultaneously generate the sound of many different musical voices in real time, we provide each processing node with an independent control program so that each node works on different sets of instruments or different parts of a large complex instrument. The program for each node is loaded by the host computer prior to execution, and is stored in each node. This program is executed once every sample period, at a rate of approximately 20kHz, as controlled by an external signal. Our sound synthesis algorithms contain no conditional execution; therefore, the processing nodes contain no hardware for conditionals. The same set of instructions is repeated until changed by the host computer. Program changes, however, are intended to occur on a very slow time scale (e.g., when a new instrument model is desired). Instantaneous changes controlling the "playing of the instrument" are implemented by the host by sending packets that change coefficient values within nodes.

Communication

Because the control flow of MIMIC programs is linear (i.e., there are no conditional branches) the communication pattern for each program is predetermined. The routing information for packets between pairs of processing nodes can be determined at compile time and loaded (in the form of a program) into the MIMIC chips. Because no routing decisions are necessary at execution time, the routing hardware within each processing node is exceedingly simple. Also, because the communication patterns are static and routing is precomputed, no packet headers are needed and the packets can be routed through each node with only a single-bit time delay.

The communication bandwidth between processing nodes is used for sending and receiving signals captured for use in sending update messages from the host processor. Therefore, two separate types of communication are supported by the network: 1) inter-node communications implementing data links in the computation graph, and 2) host-to-node communication implementing update links. This network arrangement provides a graceful way to trade inter-node data communication bandwidth for host control bandwidth.

MIMIC Node

The requirements of the algorithms and the communications within the machine result in the following node structure. Each node has three major functional units: an arithmetic unit, a memory controller, and a communication switch. The arithmetic unit is similar to those typically found in standard signal processing engines. The memory controller performs the address calculations involved in implementing delay lines and performing table lookups. The delay line and table data are stored in standard memory chips. The memory controller relieves the relatively expensive arithmetic unit from simple address calculations. The third major unit, the communication switch, handles the flow of data in and out, and through the node. This switch is much simpler than ones used in general purpose multi-computer architectures.

ICMC GLASGOW 1990 PROCEEDINGS
Each processing node comprises a custom VLSI MIMIC chip and a commercial DRAM chip. The internal structure of a MIMIC chip is shown in Figure 3.

A design goal for our system was to keep the number of pins on the custom VLSI chip to a minimum. This way we can deploy a small package, thus maximizing the density of codes on the circuit board. The communication bandwidth requirements for the music synthesis algorithms indicate that bit-serial communication among chips and with the host computer is sufficient.

![Figure 1: MIMIC system level architecture. Processing nodes are typically arranged in a multidimensional mesh. Each processing node comprises a MIMIC chip and a commercial DRAM (dynamic random access memory) chip in approximately 1.5 square inches. Nodes connect to their nearest neighbors through bidirectional links. The links are used for transmitting signal data and control information injected into the network at multiple points. All connections from the host and between nodes are bit-serial.](image1)

![Figure 2: MIMIC user level computational primitives. Instrument models are made by building graphs of these primitives. Inputs may come from the output of other primitives or from the host computer.](image2)
Summary

In this paper we present the architecture of a novel machine, called MIMIC, designed specifically for the task of producing sound in real-time by simulating the physical behavior of musical instruments. The machine achieves high performance per unit board area by using custom VLSI processing chips in conjunction with commercial DRAM chips. The high system performance required to simulate complex instruments or small ensembles of instruments precludes single processor solutions. The MIMIC architecture is designed as a MIMD array of processing nodes interconnected by a special network and controlled by a host computer. The performance obtained per unit board area (29MOPS in approximately 1.5m²) cannot be matched by commercially available DSP processors. Those processors use large, high pin-count packages, and are not designed to work in a multi-processor configuration.

An expanded version of this paper may be found in [3].

Acknowledgments

We would like to thank the members of CS292 (Fall 1988) for their their work on the MIMIC project. In particular, Paul de Dood was partly responsible for the design of the processing node. This work was sponsored by the National Science Foundation (MIP-8955566) and DARPA (N00014-87-C-0182).

References


ICMC GLASGOW '90 PROCEEDINGS