In such a compile-time (static) scheduling strategy, it is possible to predict the run time inter-processor communication (IPC) pattern. We present two techniques that make use of this compile-time determined communication pattern, for minimizing IPC and synchronization overhead in the parallel implementation. The first technique is aimed at eliminating arbitration and synchronization costs when using shared memory for IPC. We call this the Ordered Transactions strategy; the idea is to determine the order in which processors require access to shared resources and to enforce this order at run time. Enforcing such an order eliminates contention for shared resources and the need for explicit synchronization. We describe the design and hardware implementation details of a prototype multiprocessor board that was built as a proof-of-concept for the ordered transactions strategy.

The second technique we present in this thesis consists of efficient algorithms for minimizing synchronization costs in statically scheduled multiprocessors. These include procedures for detecting and eliminating redundant synchronization points in the schedule and systematically adding certain synchronization points with a view towards reducing the overall synchronization cost.

Edward A. Lee, Thesis Committee Chairman

Abstract

MINIMIZING COMMUNICATION AND SYNCHRONIZATION OVERHEAD IN MULTIPROCESSORS FOR DIGITAL SIGNAL PROCESSING

by

Sundararajan Sriram

Doctor of Philosophy in Electrical Engineering and Computer Sciences

Professor Edward A. Lee, Chair

This thesis is concerned with embedded systems for Digital Signal Processing (DSP) that consist of multiple programmable digital signal processors augmented with custom VLSI components; we will refer to such systems by the term "multiprocessor." The dataflow model of computation has been widely used for providing a formal methodology for specifying computations and mapping them to such multiprocessor systems.

In this thesis, we focus on DSP algorithms that can be specified as Synchronous Data Flow graphs and its extensions. Such algorithms can be efficiently scheduled onto multiple processing elements (a processor could be either programmable or a custom VLSI component) at compile time - computations in the graph are assigned to processors at compile time and the execution order of tasks assigned to each processor is also determined at compile time.

In such a compile-time (static) scheduling strategy, it is possible to predict the run time inter-processor communication (IPC) pattern. We present two techniques that make use of this compile-time determined communication pattern, for minimizing IPC and synchronization overhead in the parallel implementation. The first technique is aimed at eliminating arbitration and synchronization costs when using shared memory for IPC. We call this the Ordered Transactions strategy; the idea is to determine the order in which processors require access to shared resources and to enforce this order at run time. Enforcing such an order eliminates contention for shared resources and the need for explicit synchronization. We describe the design and hardware implementation details of a prototype multiprocessor board that was built as a proof-of-concept for the ordered transactions strategy.

The second technique we present in this thesis consists of efficient algorithms for minimizing synchronization costs in statically scheduled multiprocessors. These include procedures for detecting and eliminating redundant synchronization points in the schedule and systematically adding certain synchronization points with a view towards reducing the overall synchronization cost.

Edward A. Lee, Thesis Committee Chairman

Abstract

MINIMIZING COMMUNICATION AND SYNCHRONIZATION OVERHEAD IN MULTIPROCESSORS FOR DIGITAL SIGNAL PROCESSING

by

Sundararajan Sriram

Doctor of Philosophy in Electrical Engineering and Computer Sciences

Professor Edward A. Lee, Chair

This thesis is concerned with embedded systems for Digital Signal Processing (DSP) that consist of multiple programmable digital signal processors augmented with custom VLSI components; we will refer to such systems by the term "multiprocessor." The dataflow model of computation has been widely used for providing a formal methodology for specifying computations and mapping them to such multiprocessor systems.

In this thesis, we focus on DSP algorithms that can be specified as Synchronous Data Flow graphs and its extensions. Such algorithms can be efficiently scheduled onto multiple processing elements (a processor could be either programmable or a custom VLSI component) at compile time - computations in the graph are assigned to processors at compile time and the execution order of tasks assigned to each processor is also determined at compile time.