

Code Generation for Heterogeneous Multiprocessors

José Luis Pino

Edward A. Lee

AT&T Bell Labs Fellowship, ARPA(RASSP) F33615-93-C-1317 and the Ptolemy Project

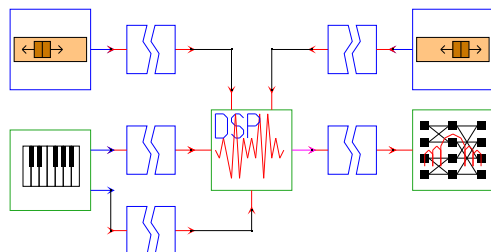


Figure 1. Top-level FM synthesis specification. The keyboard, DSP, and FFT blocks each contain an internal dataflow graph which is not shown.

Today, programming of signal processing algorithms on embedded digital signal processors is done in assembly language and scheduled by hand. If the processor configuration changes, the code must be redesigned. If the processors themselves change, the code must be completely rewritten. In this project, we are interested in rapid prototyping of signal processing applications. We have developed a code generation framework for heterogeneous multiprocessor DSP systems from a high-level block diagram specification [1]. Code generation requires partitioning and scheduling of the algorithm onto the multiprocessor architecture. Subsequently, the algorithm code is generated in the appropriate languages.

The algorithm is specified using multiple independent dataflow graphs as described in [2]. The independent graphs communicate over nondeterminate communication links. These links do not introduce data dependencies among the independent graphs. We have found that this communication mechanism is ideal for specifying run-time controls and displays for real-time signal processing applications.

An illustration of the use of such multiple dataflow graphs is shown in figure 1. Here we have the top-level specification of an FM music synthesis algorithm. This application has been targeted to a heterogeneous platform consisting of a Unix workstation and a Motorola 56001 DSP board. In this example, there are five independent dataflow graphs communicating over five nondeterminate links. The C and Tcl/Tk code is generated for the Unix workstation; Motorola 56001 assembly code is generated for the DSP board. The user interface, shown in figure 2, is generated from the user specification.

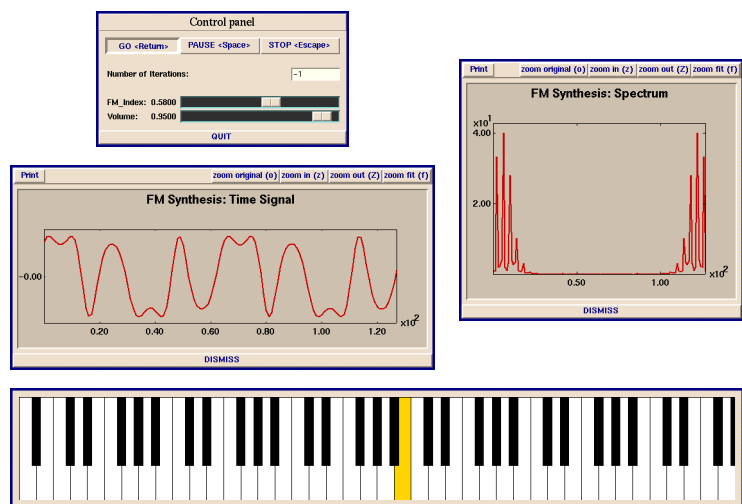


Figure 2. User interface generated for the FM synthesis application.

- [1] J. L. Pino, S. Ha, E. A. Lee, and J. T. Buck, "Software Synthesis for DSP Using Ptolemy," to appear in *Journal of VLSI Signal Processing*, special issue on *Synthesis for DSP*, vol. 9, no. 1, 1995. (http://ptolemy.eecs.berkeley.edu/papers/jvsp_codegen)
- [2] J. L. Pino, T. M. Parks and E. A. Lee, "Mapping Multiple Independent Synchronous Dataflow Graphs onto Heterogeneous Multiprocessors," *Proceedings of the IEEE Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, November 1994. (<http://ptolemy.eecs.berkeley.edu/papers/multiIndepGraph>)

Hierarchical Static Scheduling of Dataflow Graphs onto Multiple Processors

José Luis Pino

Edward A. Lee

AT&T Bell Labs Fellowship, ARPA(RASSP) F33615-93-C-1317 and the Ptolemy Project

The goal of this project is to reduce the complexity of scheduling synchronous dataflow (SDF) [1] graphs onto multiple processors. SDF semantics have proven to be useful in describing multirate digital signal processing algorithms. Furthermore, compile-time scheduling is possible from SDF block diagram descriptions. Many synchronous dataflow schedulers are available for both uniprocessor and multiprocessor architectures. Those for uniprocessor systems optimize for costs such as code and buffer memory usage while multiprocessor schedulers optimize the makespan of the application.

We are implementing a scheduling framework that can make use of heterogeneous schedulers. The core of this framework is a clustering technique that reduces the number of actors before expanding the SDF graph into a directed acyclic graph [2]. The internals of the clusters are then scheduled with uniprocessor SDF schedulers which can optimize for memory usage. The clustering is done in such a manner as to leave ample parallelism exposed for the multiprocessor scheduler.

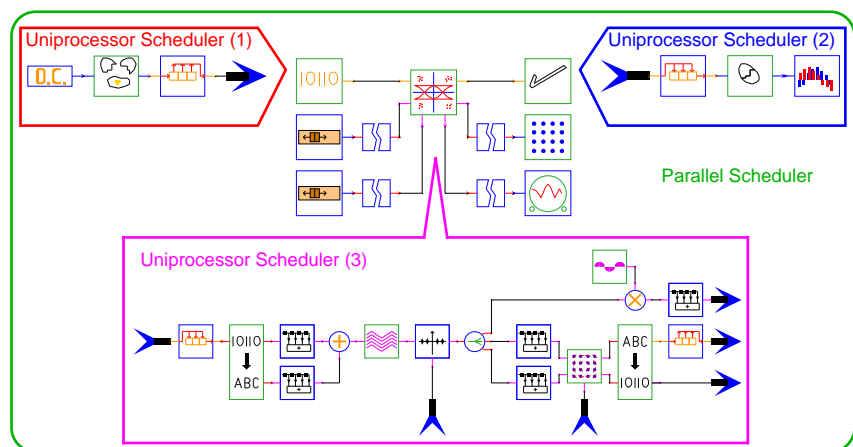


Figure 1. A 4-QAM Modem using 4 schedulers. There are 3 uniprocessor schedulers hierarchically embedded inside of a parallel schedule.

This framework has been tested on a number of practical applications

detailed in [2] and [3]. One of the applications, a 4-QAM modem, is shown in figure 1. For this modem, the use of our framework realized a 90x speedup in scheduling time with an 60x reduction of memory usage.

- [1] E. A. Lee and D. G. Messerschmitt, "Synchronous data flow," *Proceedings of the IEEE*, vol. 75, no. 9, 1987, p. 1235-1245.
- [2] J. L. Pino, S.S. Bhattacharyya and E. A. Lee, *A Hierarchical Multiprocessor Scheduling Framework for Synchronous Dataflow Graphs*, UCB/ERL M95/36, May 30, 1995. (<http://ptolemy.eecs.berkeley.edu/papers/erl-95-36>)
- [3] J. L. Pino, S. S. Bhattacharyya and E. A. Lee, "A Hierarchical Multiprocessor Scheduling System for DSP Applications," *Proc. IEEE Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, Oct. 29 - Nov. 1, 1995. (<http://ptolemy.eecs.berkeley.edu/papers/hierStaticSched-asilomar-95>)

Interface Synthesis in Heterogeneous System-Level DSP Design Tools

José Luis Pino

Edward A. Lee

AT&T Bell Labs Fellowship, ARPA(RASSP) F33615-93-C-1317 and the Ptolemy Project

We have developed a framework for automatic interface construction between prototyping and simulation engines in system-level DSP design tools. The techniques described below have been tested using the SDF (synchronous dataflow) model of computation in Ptolemy and can be extended to other models of computation. The framework provides incremental compilation, interfaces to foreign simulators, and interfaces between code generation domains.

Using incremental compilation, a compute-intensive subsystem in, for example, the SDF simulation domain can be retargeted to CGC (code generation in C) and compiled to become a single monolithic actor in SDF (figure 1). A similar capability is used to encapsulate a CG56 subsystem (which runs on the Motorola DSP56000) into an SDF actor. This new actor can then be added to the designer's actor library.

The interface mechanism also allows for easy incorporation of foreign simulators. For example, a VHDL subsystem can be analyzed to synthesize a fast customized C interface to a commercial VHDL simulator. The VHDL subsystem in turn can be interfaced with another system that executes on a DSP card.

Finally, the framework allows the combination of more than one code generation domain. For example, CGC can be mixed with CG56 to produce programs that execute concurrently on a host workstation and a DSP card.

A fundamental problem is that dataflow systems cannot always be incrementally compiled. The problem lies in the fact that dataflow systems lack the composition property. Thus subsystems of dataflow actors in an application specification do not necessarily have the same semantics as an individual actor.

- [1] J. L. Pino, M. C. Williamson, and E. A. Lee, "Interface Synthesis in Heterogeneous System-Level DSP Design Tools," submitted to *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Atlanta, GA, May 1996.

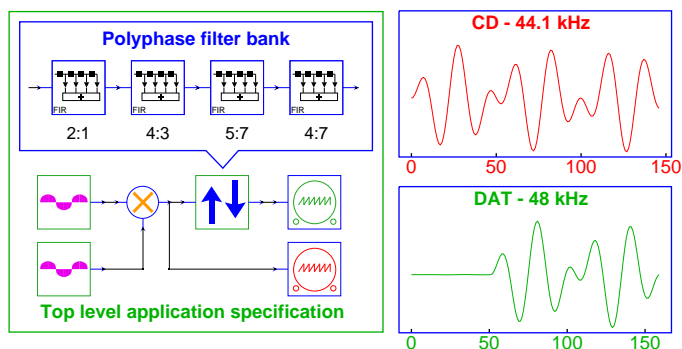


Figure 1. CD (44.1 kHz) to DAT (48 kHz) sample rate conversion. The polyphase filter bank is incrementally compiled from CGC into a monolithic simulation SDF actor. The input (CD) and output (DAT) signal plots are shown.