Techniques for Co-Design of Optically-Connected Embedded Multiprocessors

Neal Bambha, US Army Research Laboratory and University of Maryland, College Park

Shuvra Bhattacharyya, University of Maryland, College Park[1]

(In Proc. Annual Workshop on High Performance Embedded Computing, pages 97-99, Lexington, Massachusetts, September 2002)

Several trends in technology have important implications for future digital signal processing (DSP) systems. By the year 2010, integrated circuit technology will allow 800 million transistors on a single chip. Already, manufacturers are placing multiple DSP cores on a single chip. Multiprocessor systems will become increasingly important in the future. A significant challenge is to develop software and compiler techniques to effectively exploit multiple processors. Signal and image processing algorithms are among those applications that can benefit from multiprocessor systems. Optics provides unique advantages and opportunities for designers of embedded multiprocessor systems, including the ability to construct highly connected and irregular networks that are streamlined for particular applications. Using these networks, it is possible to implement application mappings that allow flexible, low-hop communication patterns between processors. This has advantages for reduced system latency and power. Such optically connected multiprocessors are particularly promising for embedded DSP applications, which are highly parallel, and typically have tight constraints on latency and power consumption. Several groups have demonstrated optically-connected multiprocessor systems (e.g. see[2] [3]). However, comparatively little work has been done to develop compiler technology and automated mapping tools to take advantage of these systems.

This work addresses the co-design of interconnect topologies and application mappings for DSP systems on optically connected multiprocessors. We demonstrate that existing DSP scheduling algorithms will deadlock for arbitrarily-connected networks, or when communication is restricted to a limited number of hops. We show that these low-hop communication schedules produce low power and low latency mappings. We demonstrate an effective algorithm for determining the set of feasible processors that will avoid schedule deadlock in a limited-hop schedule, and a useful metric, called communication flexibility, for the degree to which a given scheduling decision constrains future decisions (in the context of the given communication topology). We use this algorithm and the flexibility metric in conjunction with a dynamic level scheduling algorithm (DLS) to map several DSP applications across a wide range of interconnect topologies. These experiments demonstrate that the flexibility metric significantly improves scheduling performance.

We examine a set of DSP application benchmarks and schedule them using two different scheduling modes, one that incorporates only feasibility information (to avoid deadlock), and another that takes both feasibility and flexibility into account. We refer to these as the *feasibility-only* and *feasibility-flexibility* modes, respectively. To evaluate the performance across a range of connectivity levels, we schedule the applications onto networks with varying degrees of connectivity. Results for an FFT application are shown in Figure 1, showing a 30% relative schedulinng improvement when incorporating the flexibility metric. It is important to note that the DLS algorithm, by itself, will not work without at least using the feasibility metric to avoid deadlock.

The computational model used in this work is that of conventional acyclic *task graphs*, in which graph vertices (*tasks* or *nodes*) represent computations and each edge represents the communication of a packet of data from the source task to the sink task. Task graphs are particularly well-suited to DSP applications, which are frequently programmed as *synchronous dataflow* (SDF) graphs, a class of program representations for which efficient techniques exist for scheduling, communication synthesis, and power management. We outline a high-level architectural model of an optically-connected system based on dataflow. This model exposes the inherent parallelism of the application, and allows significant optimization with respect to reduced synchronization overhead and guarantees of deadlock avoidance.

We show that the optimal interconnect topology for low-hop communication is typically a very irregularly-connected network. We present an algorithm for determining the minimal interconnect for a given set of

[1] This work was supportedby the DARPA funded Optoelectronic Center for Innovative Photonic Chipscale Technologies (Contract number MDA972-00-1-0023).

[2] M. W. Haney, M. P. Christensen, P. Milojkovic, “Description and Evaluation of the FAST-Net Smart Pixel-Based Optical Interconnection Prototype,” *Proceedings of the IEEE*, vol. 88, no. 6, June 2000.

[3] P. S. Guilfoyle, “Digital Optical Computing Architectures for Compute Intensive Applications,” *Proceedings of the International Conference on Optical Computing, 1994.*

applications, given latency and/or power constraints. Optical interconnect technology is promising for this, since the interconnection patterns can flexibly be streamlined and reconfigured to match the target applications. Even in systems such as FAST-Net [2] which are designed to provide full connectivity, it is still desirable from the viewpoint of power and heat dissipation to construct minimal interconnect mappings, since for a given application, non-essential transmitters can be turned off.

However, the freedom to streamline interconnection patterns opens up a vast design space, and thus the design of an optimal interconnect structure for a given application or set of applications is a significant challenge. We illustrate how our single-hop scheduling strategies, and the underlying concept of communication flexibility, can be used to guide the synthesis of application-specific interconnect structures. The main idea here is that for embedded multiprocessors, the interconnect topologies should be driven by the specific application mappings that will execute across them, and jointly designing the two is advantageous.

Specifically, we have developed a greedy, heuristic algorithm, called the *two-phase link adjustment* (TPLA) algorithm, to synthesize an interconnect and an associated multiprocessor schedule for a given application. The TPLA algorithm starts with a fully connected network, and operates in *down* and *up* phases. Input to the algorithm is either a makespan constraint for the application, or a constraint on the total number of links.

Each step of the down phase in TPLA removes one link, while each step of the up phase adds one link. One step of the down phase consists of assigning each existing link a score based on the schedule makespan resulting from its removal, and removing the link with the lowest score. A history of scores is kept for each link. For the first pass through the down phase, ties between links are broken randomly. On subsequent passes, the link history is used to break ties. Results for the TPLA algorithm are shown in Figure 2, which shows a 42% relative improvement over a random link removal strategy.

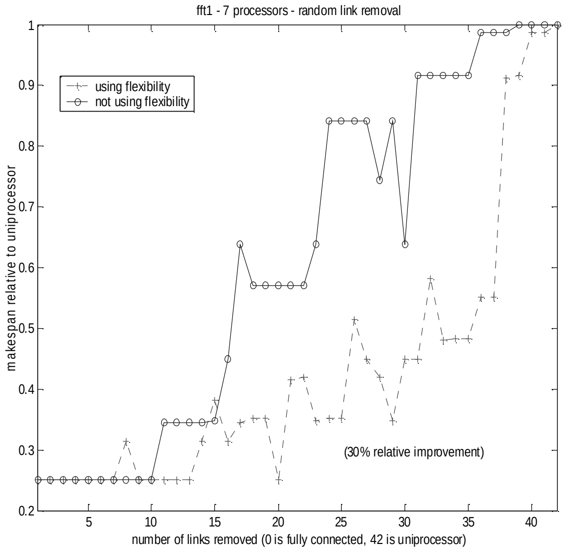

Figure 1. Schedules constructed using DLS with and without considering the processor flexibility metric.

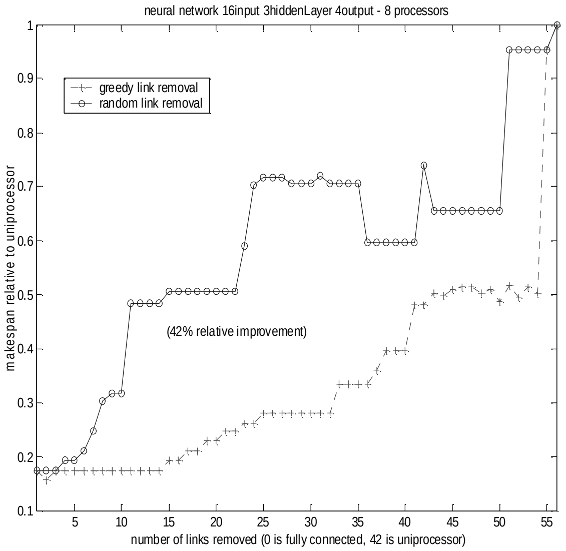

Figure 2. Link synthesis using the TPLA algorithm.

Optical interconnect technology is promising for global communication in embedded multiprocessors, since the interconnection patterns can flexibly be streamlined and reconfigured to match the target applications. However, due to the power consumption characteristics of optical links, it is useful to restrict communication across them to single-hop transfers. In this paper, we demonstrate an effective algorithm for determining the set of feasible processors that will avoid schedule deadlock in a single-hop schedule, and a useful metric, called communication flexibility, for the degree to which a given scheduling decision constrains future decisions (in the context of the given communication topology). We use this algorithm and the flexibility metric in conjunction with the DLS algorithm to map several DSP applications across a wide range of interconnect topologies. These experiments demonstrate that the flexibility metric significantly improves scheduling performance. We also demonstrate that these scheduling techniques can be used to effectively guide an algorithm for jointly synthesizing the interconnection network together with the mapping of an application onto the network.