ENEE 759E: Synthesis of Embedded Software

Course Goals:

Overview:
This course focuses on architectures, programming models, synthesis methods, and compiler technology for the design and implementation of digital signal processing (DSP) software. DSP is a core part of numerous embedded computing applications, ranging from low cost, high volume consumer-oriented applications to specialized, high performance systems for military and security applications. Examples of important DSP-related application areas include wireless communications, video conferencing, surveillance, medical imaging, digital audio, radar, and PC multimedia.

Programmability is a highly desirable feature of implementation platforms for very large scale integration (VLSI) systems for DSP. The trend toward programmable platforms is fueled by tight time-to-market windows, which in turn result from intense competition among DSP product vendors, and from the rapid evolution of technology, which shrinks the life cycle of state-of-the-art product designs. As a result of time-to-market pressure, designers are often forced to begin architecture design and system implementation before the specification of a product is fully completed. For example, a mobile communication product is often designed before the signal transmission standards under which it will operate are finalized, or before the full range of standards that will be supported by the product is agreed upon. In such an environment, late changes in the design cycle are mandatory. The need to quickly make such late changes requires the use of software. Furthermore, whether or not the product specification is fixed beforehand, software-based implementations using off-the-shelf processors take significantly less verification effort compared to custom hardware solutions.

This course provides an integrated treatment of several areas that are important for efficient design and implementation of DSP software on programmable platforms, in particular on single- and multiprocessor configurations of programmable digital signal processors (PDSPs), which are domain-specific embedded processors for DSP. The market for PDSPs has expanded dramatically in recent years due to the emergence of important consumer technologies such as wireless communications and PC multimedia, as well as due to advances in many other DSP-related application areas such as those listed above.

There are a number of distinguishing characteristics of DSP software implementations, and these characteristics must be handled adequately by the programming models, synthesis techniques, and compiler optimizations that are used in the process of developing DSP software. First, DSP software typically interacts continuously with the environment that surrounds it. Second, DSP software operates on effectively unbounded input data sequences, and similarly, produces output data sequences that are unbounded in length. Another important characteristic of DSP software implementations is the wide variety of critical metrics that must be taken into account when deriving these implementations. These metrics include throughput, latency, program and data memory requirements, memory partitioning constraints, energy consumption, power consumption, and financial cost. Furthermore, the architectures of PDSPs are significantly different from other types of processors, such as general-purpose microprocessors. This course reviews methods that have been developed and that continue to evolve to address these distinctive characteristics in the design and implementation of DSP software.

Texts:
• S. Sriram and S. S. Bhattacharyya, Embedded Multiprocessors, Marcel Dekker, Inc., 2000.
• Lecture notes: distributed in class.

Last Updated:

August, 2006