Dissertation Summary

- Scalable Wireless Application-specific Processors -
SWAPs for emerging wireless systems

Sridhar Rajagopal

My dissertation focuses on the design of scalable and programmable architectures to meet real-time requirements for the physical layer of high data rate wireless systems.

Motivation:
A plethora of standards for indoor and outdoor wireless environments are emerging for wireless systems such as W-CDMA, 802.11a W-LAN and Bluetooth. All standards seek to provide extremely high data rates (10-100 Mbps, depending on the standard) for multimedia applications. These standards also aim to provide better error rate performance by minimizing interference among users, using multiple antennas and providing higher spectral efficiency (in bits/sec/Hz) that requires the use of highly sophisticated algorithms. It is extremely challenging to attain real-time performance for implementing these computationally complex algorithms at these high data rates for wireless systems. Traditional designs for wireless systems have typically been as customized ASICs in order to meet high data rate requirements and to minimize power consumption. However, the need to rapidly prototype algorithms for these ever-evolving wireless standards, long design time and the flexibility needed to support multiple standards for ubiquitous wireless connectivity make programmable solutions increasingly important in emerging wireless systems.

Current programmable architectures fail to meet both the real-time and power consumption requirements for implementing proposed computation-intensive wireless algorithms at these high data rates. Hence, heterogeneous solutions incorporating co-processors have been proposed for wireless systems such as the recently released TI C6416 DSP, which has both a Viterbi and Turbo co-processor for decoding. In such heterogeneous systems, the role of the DSP is becoming increasingly diminished to that of a co-processor controller. As more co-processors are added for supporting advanced algorithms in the future, the benefits of the programmable solution will be lost and the system will become as inflexible as a custom ASIC implementation. Current programmable architectures also lack the scalability and robustness needed to support proposed algorithm designs with ever-increasing computational complexity for these evolving wireless standards, while still meeting real-time requirements for multimedia-based applications.

Dissertation Proposal:
For my dissertation, I have proposed a programmable and yet scalable architecture framework for designing architectures for wireless systems. These architectures aim to eliminate/minimize the need for co-processor support by explicitly exploiting data parallelism in hardware similar to ASICs. My dissertation evaluates the bottlenecks that occur in a programmable implementation for a chain of sophisticated receiver algorithms and seeks to quantify and minimize the gap between ASICs and programmable solutions for wireless systems. I am using a flexible, stream-based media processor simulator from Stanford that can exploit data parallelism by providing multiple clusters. This simulator is based on the ‘Imagine’ media processor architecture developed at MIT/Stanford. The base simulator architecture is modified to provide support for wireless communication applications by providing architecture scalability and support for efficient inter-cluster communication.
**Dissertation Contributions:**

(i) **A dynamically scalable architecture design for wireless systems:**

A fixed architecture design for programmable architectures that exploits data parallelism is not the right solution for wireless applications because the amount of data parallelism varies with algorithms and also with algorithm parameters. For example, a system may be required to support Viterbi decoding at constraint lengths 5 and 9, which have different levels of data parallelism. Choosing the lower parallelism compromises on the data rates while choosing the higher one wastes computational resources and hence, power. An architecture that allows scaling of the hardware so that the right amount of parallelism can be exploited while other units can be powered down is one of the contributions of my dissertation.

(ii) **Exploiting a new inter-cluster communication pattern:**

Many of the parallel implementations of physical layer algorithms show the same inter-cluster communication pattern. I have shown that operations such as matrix transposing, shuffling of the trellis during the Viterbi/Turbo/FFT computations, and packing and unpacking data to exploit sub-word parallelism – all show an odd-even permutation between two registers in the multi-cluster architecture. This inter-cluster communication forms a significant portion of the gap between the ASIC and programmable solutions for wireless systems. Providing hardware support for this communication pattern can provide significant speedups for multiple clusters.

(iii) **Exploring tradeoffs between functional unit utilization and memory stalls:**

High functional unit utilization and memory stall minimization is important in programmable designs for attaining low power consumption in wireless systems. I have found that a trade-off exists between an implementation of an algorithm that maximizes the functional unit utilization during execution of a kernel but uses memory to re-order data and an implementation that avoids memory stalls by re-ordering data within the kernels using the inter-cluster communication. The tradeoff seems to favor the “avoiding memory stalls” implementation of the algorithm. This is because memory stalls are found to have a larger impact on performance and also consume more power than if data re-ordering were to be done within the clusters. Also, packed data (exploiting sub-word parallelism) cannot be re-ordered in memory without unpacking in the clusters, re-ordering in memory and packing in the clusters again.

(iv) **Prototyping and designing architectures for future systems:**

This design framework proves to be a sound basis for investigating architectures as new algorithms get designed and evaluated for wireless systems. New algorithms can be prototyped and evaluated for performance in a high level language by simply exploiting more data parallelism if available. The compiler could also be enhanced to automatically search for the best cluster and functional unit configuration for the algorithms. If this fails to meet real-time, the software implementation and the simulator help in rapidly finding and solving new bottlenecks that are present in the algorithms under investigation.

My dissertation research has led to the SWAP designs that are able to meet real-time performance requirements in third generation wireless systems. The SWAP architectures, being scalable and programmable, can dynamically exploit parallelism levels for wide range of applications and can provide high performance computing for future embedded systems.