1 Introduction

1.1 Motivation

After spectacular growth over the past decade, wireless communication remains the single most important driver for the global semiconductor and electronics industry for the foreseeable future. With a compound annual growth rate of 11% over the next 4 years, wireless communication will remain a rapidly growing market and will reach total sales of $101 billion by 2015 [1]. It is expected that the number of mobile subscribers will reach over half of the world population by 2015 and the daily average received data volume per device will increase from less than 10 MB today to over 280MB by 2020 and over 600MB by 2028 [2–4]. As the mobile communication paradigm shifts from voice-centric to communication-centric, future mobile devices will be required to support pervasive Gbps internet access and a variety of applications related to multimedia, computer vision, and the human-machine interface. These features must be supported while delivering battery life that provides for days of active-use experience [5]. As a result, future mobile devices must implement increasingly diverse and complex communication baseband and multimedia/vision signal processing which imposes stringent cost, programmability/flexibility, and energy efficiency constraints. As CMOS technology scaling is quickly approaching its physical limit, it is evident that designers will have to increasingly rely on architectural and system level innovations to meet stringent energy and performance constraints in the coming decades.

Since signals from the physical world vary substantially and signal processing often aims to meet certain system-level specifications (e.g., signal-to-noise ratio and frame error rate), most signal processing tasks share the following two features: (1) The time-varying nature of analog signals measured in a physical environment tends to induce significant run-time signal processing computational workload variations; (2) Given a signal processing task, there is typically a large degree of freedom in signal processing algorithm implementation depending on signal characteristics. For example, different wireless signal detection algorithms could be used under different radio link conditions. These two features suggest that, given a signal processing task, the required amount of computation may vary significantly at run time. As a result, it is possible to improve energy efficiency by changing the algorithm implementation in response to changes in the physical environment, effectively providing *just enough* computation. In this context, a highly-programmable, memory intensive implementation platform which meets performance needs is required. Although microprocessors and digital signal processors (DSPs) provide a high level of programmability and flexibility, their relatively limited performance make them inadequate in many energy-constrained environments. Field-programmable gate arrays (FPGAs) provide significant fine-grain parallelism but suffer from several drawbacks. The highly-programmable nature of SRAM-based FPGAs leads to a 10× energy penalty [6] versus an equivalent ASIC implementation. Additionally, to exploit the run-time computational workload variation and algorithm changes that maximize signal processing energy efficiency, FPGAs must be dynamically reconfigured at run time [7]. Since existing FPGAs can only hold one set of configuration data on-chip, new configurations must be loaded from off chip, expending energy and causing a latency penalty.

A near-term opportunity to fundamentally remove the signal processing energy efficiency barrier is enabled by two industrial trends. (1) Coarse-grain reconfigurable architectures [8] [9], which have flexible routing similar to FPGAs but arithmetic processing blocks, have gained interest in recent years and have demonstrated superior energy efficiency for memory-limited computation-intensive tasks. These architectures demand a smaller amount of configuration data versus FPGAs. This feature provides an opportunity for the development of a coarse-grain reconfigurable device with multiple on-chip configurations which can quickly respond to run-time signal processing computational workload variation and algorithm design freedom. (2) Over the past few years there has been renewed interest in the search for highly scalable universal memory [10–12]. Phase-change RAM (PCRAM) and magnetoresistive RAM (MRAM) have received the most attention because of their scalability, endurance, and non-volatility. These technologies have experienced significant advancement over the past several years (e.g., see [13–25]). High-density on-chip memory could store many sets of configuration data, making it practically feasible to exploit just enough signal processing computation demand at run time. The non-volatile nature of the storage limits its impact on overall system power consumption. Additionally, the availability of high density on-chip memory directly supports the memory-intensive nature of communication baseband and multimedia/vision signal processing.

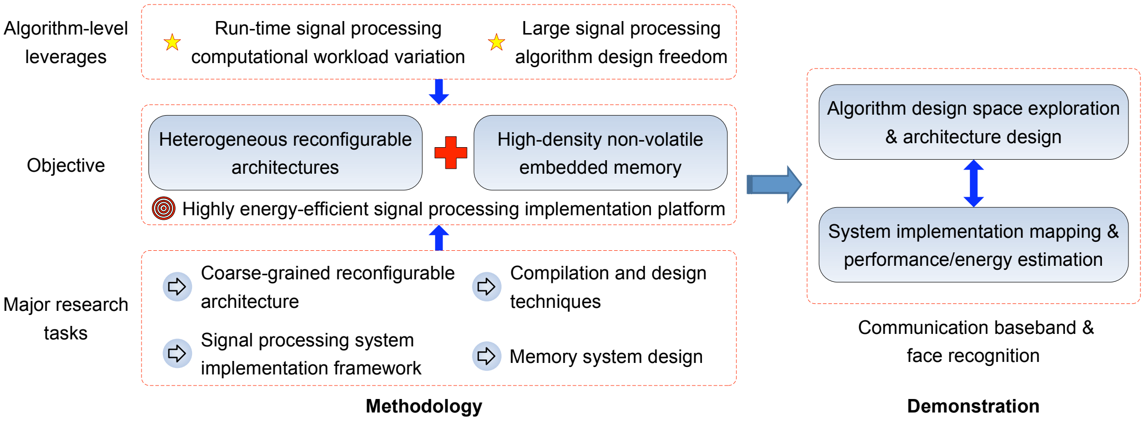

Algorithm-level leverages
Run-time signal processing computational workload variation
Large signal processing algorithm design freedom
Objective
Heterogeneous reconfigurable architectures
High-density non-volatile embedded memory
Highly energy-efficient signal processing implementation platform
Major research tasks
Coarse-grained reconfigurable architecture
Compilation and design techniques
Signal processing system implementation framework
Memory system design
Methodology
Algorithm design space exploration & architecture design
System implementation mapping & performance/energy estimation
Communication baseband & face recognition
Demonstration

Figure 1: Overview of the proposed research.

1.2 Outline of Proposed Research

A comprehensive new reconfigurable architecture which directly interfaces to banks of high-density non-volatile embedded memory forms the cornerstone of our energy-efficient signal processing solution. This architecture, in concert with compilation and simulation tools will be applied to time-varying communications algorithms and image processing algorithms to quantify the potential gains. Overall, the proposed work has a two-fold motivation: (1) **The addressed problem is important**: the use of a coarse-grain reconfigurable device with integrated non-volatile memory and a flexible compiler will allow for fast and efficient mapping and execution of signal processing applications, such as communication baseband signal processing and computer vision, two critical applications for mobile computing. (2) **The use of emerging embedded memory and coarse-grain architecture to facilitate application-dependent computation reconfiguration is largely unexplored**. The combination of coarse-grain reconfigurable architectures and high-density non-volatile embedded memory technologies provides unique opportunities to exploit the run-time signal processing computational variation and large algorithm design freedom needed to improve the energy efficiency of signal processing without compromising programmability and flexibility. The availability of high-density embedded memory readily embraces the memory-intensive nature of signal processing to further improve energy efficiency. Our proposed research is illustrated in Fig. 1. Major research tasks include:

(1) *Integration of a new coarse-grain reconfigurable architecture and high-density non-volatile memory*: A new coarse-grain reconfigurable architecture which is tuned for signal processing applications will be developed. Since mapping to these types of architectures is critical, a compiler which takes architecture reconfiguration based on changing environmental conditions will also be developed. An architectural simulator will be created to assist architecture exploration. A novel aspect of the new architecture is its direct access to large amounts of on-chip non-volatile memory. We will investigate how the embedded non-volatile memory fabric should be architected to support storage of on-chip configuration data or signal processing data. Users will have access to this memory for storage or architecture reconfiguration either through automatic compilation or through user directives specified prior to the compilation process. We will study the most appropriate approach to interface the memory fabric with the reconfigurable architectures.

(2) *Signal processing system framework*: We will develop a system design framework that enables designers to systematically apply the developed coarse-grain reconfigurable architecture and supporting design tools to realize energy-efficient implementations for various signal processing tasks. We envision that this design framework will contain two phases, off-line application-specific algorithm design space exploration and on-line reconfigurable architecture implementation scheduling. Run-time power management techniques will leverage dynamic reconfiguration to adaptively minimize overall system power consumption. Our analysis in Section 3.1 indicates that reconfiguration will have a negligible effect on signal processing performance.

(3) *Demonstration with real-life signal processing applications*: The new architecture and design methodology will be applied to two energy-hungry signal processing tasks that will be employed on future mobile devices: (1) wireless communication baseband signal processing for an emerging long-term evolution (LTE)

communication system, and (2) face recognition, the most important augmented reality application for future mobile devices. Besides their obvious importance, these two applications have been previously implemented using a number of signal processing algorithms. These algorithms will provide convenient algorithm foundations for our new architecture and design methodology. As demonstrated in Section 2.1, these signal processing algorithms exhibit over one order of magnitude run-time computational workload variation. Coupled with energy efficiency advantages of coarse-grain reconfigurable architectures, this suggests that the proposed solution can potentially achieve at least 10× energy efficiency improvement, effectively closing the energy efficiency barrier to programmable signal processing implementations.

1.3 Intellectual Merit and Broader Impact

The intellectual merit of this proposal lies in the cohesive integration and use of a coarse-grain reconfigurable device which is interfaced to high-density non-volatile memory technology. This interface will be exploited at the system level to dynamically adapt computation-intensive signal processing applications for significant energy savings. The PIs complementary skills in signal processing, reconfigurable architectures and compilation, integrated circuits, memory circuits and embedded systems are well matched to the objectives of this proposal. The proposed architecture will enable future mobile devices in commercial and defense-related sectors to continue to track their historic performance and functionality scaling by supporting energy efficiency.

Our proposed research is tightly coupled with educational opportunities at the University of Massachusetts, Amherst and Rensselaer Polytechnic Institute. Enhancements to graduate curricula will be developed based on the research results. Resources developed for the project will be made available to Mt. Holyoke College and the University of Puerto Rico, Mayaguez based on our previous relationships developed as part of an NSF Engineering Research Center. Further research infrastructure dissemination for education and training will be implemented through WiKi-based open access, and outreach activities will also be pursued to motivate high school students towards higher education in science and technology.

2 Background and Motivations

2.1 Signal Processing Algorithm-Level Variation

As described in Section 1, our proposed work will leverage the significant run-time computational workload variation of many signal processing algorithms and the wide algorithm implementation freedom of the algorithms. In this section we use applications from wireless communications (baseband signal processing) and computer vision (Adaboost-based face recognition) to demonstrate these features.

2.1.1 Wireless Communication Baseband Signal Processing

Energy-efficient implementation of baseband signal processing in future wireless communication devices will be increasingly challenging due to several industrial trends: (1) Link adaptation (or adaptive modulation and coding) [26] is widely used to maximize overall data transmission throughput by adapting the modulation and coding scheme according to the quality of the radio channels. Link adaptation is performed frequently. For example, high-speed downlink packet access (HSDPA) requires link adaptation every 2 ms. (2) Software-defined radios, cognitive radios [27,28] and emerging commercial smart phones are driving important global markets. Future related communication devices must be able to seamlessly support a variety of communication standards and self adapt to changing communication environments. (3) Iterative signal detection and error correction code (ECC) decoding based upon the Turbo principle [29] are pervasively used to minimize required channel signal-to-noise ratios (SNR). The iteration number of the decoders and associated computational complexity are directly related to many varying system parameters and the radio link quality.

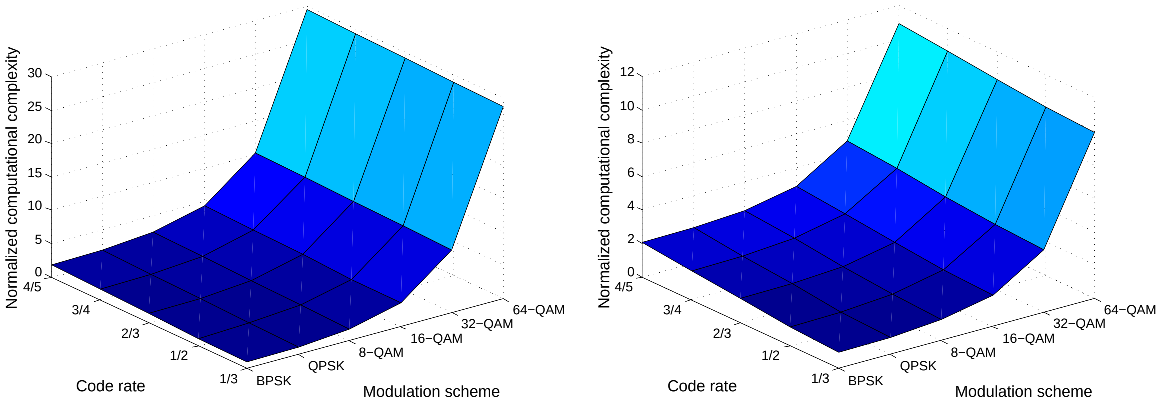
It is also expected that multiple-input multiple-output (MIMO) technology [30,31] will be pervasively used in virtually all communication standards such as long term evolution (LTE) and WiMAX. These trends inevitably will create significant run-time computational workload variations for baseband signal processing. Table 1 illustrates a set of possible parameters for use in representative MIMO wireless communication systems with link adaptation.

To quantitatively demonstrate the computational workload variation inherent in baseband signal processing, time varying computational complexity is illustrated using the parameters listed in Table 1. It is assumed

Table 1: A set of wireless communication system parameters.

Parameters	Permissible configurations
Number of TX antennas	1, 2, 3, 4
Number of RX antennas	1, 2, 3, 4
Modulation scheme	BPSK, QPSK, 8-QAM, 16-QAM, 32-QAM, 64-QAM
ECC code rate	1/3, 1/2, 2/3, 3/4, 4/5

that ECC capability is realized with widely-used low-density parity-check (LDPC) codes with a min-sum decoding algorithm [32] and soft-output MIMO signal detection is realized using the K-best detection algorithm [33]. The maximum number of internal LDPC decoding iterations is set to 16 and the value of K in K-best detection is set to be proportional to the modulation size and antenna number. By normalizing the complexity of various computational functions in terms of additions, the normalized computational complexity under different system configurations is estimated in Fig. 2. When coupled with the high frequency of link adaptation in real systems, the results demonstrate a significant degree of run-time baseband signal processing computational workload variation.



(a) 4×4 MIMO

(b) 2×2 MIMO

Figure 2: Estimated normalized computational complexities under different system configurations.

The signal processing modules, which perform MIMO signal detection, ECC decoding, channel estimation and OFDM demodulation, can be realized by several different specific algorithms covering a wide range of computational complexity versus performance trade-offs. Even for the same signal processing algorithm, different parameter configurations, such as finite word-length precision and iteration number, can cover a range of computational complexity versus performance trade-offs. As a result, algorithm implementation freedom can be leveraged by run-time system reconfiguration to improve overall energy efficiency.

2.1.2 Face Detection

Face detection has long been an important challenge in computer vision as it serves as a critical preprocessing step for other image recognition and tracking steps. Recently, Viola and Jones proposed an AdaBoost-based algorithm [34] using Haar-like features to carry out face detection. This algorithm is now widely used in practice. The AdaBoost-based detection scheme achieves a high detection rate by employing the cascade structure shown in Fig. 3 to ensure fast processing.

Each stage of the cascade is composed of a number of weak classifiers, which use Haar-like features for feature extraction. By combining these weak classifiers, a boosted strong classifier is constructed to detect complicated objects. An integral image is used in the AdaBoot-based detection algorithm [34] to reduce the computational cost for calculating Haar-like features. Detection is performed on a rectangular region, referred to as a sub-window, in the image. To detect objects with different sizes, different sub-window sizes are used on each image. A sub-window passes a stage if the stage classifier function returns a match, otherwise the

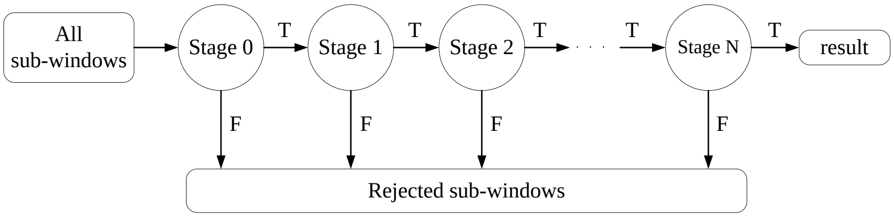


Figure 3: Cascade architecture of N classifier stages, where each stage is composed of a number of weak classifiers.

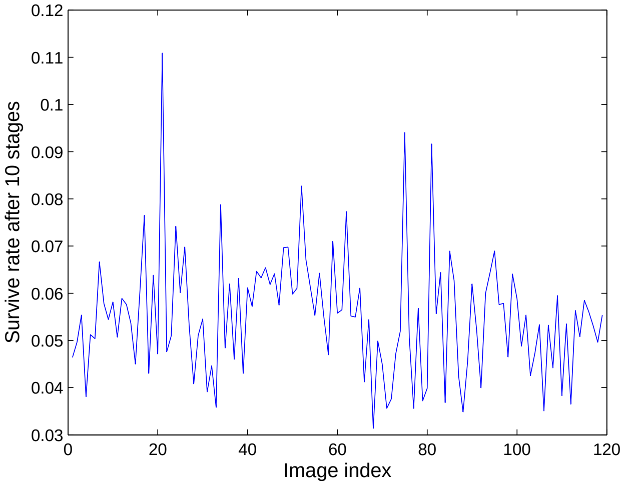


Figure 4: Percentage of sub-windows of each image passing the first 7 stages out of the total 25 stages.

sub-window is rejected and is not processed by the following stages. A sub-window only includes a face if it has been tested by every classifier and passes all stages. Typically, a majority of sub-windows are rejected by the first few stages, and only a small portion of the survivors reach the last stage. As a result, the computational workloads of the former stages and subsequent stages are highly imbalanced. To demonstrate the inherent run-time computational workload variation, we use the MIT+CMU test images [35] as a profiling data set and training data from the Intel OpenCV library [36] for the front face. Fig. 4 shows the fluctuation in sub-window pass rate for the first 8 out of 25 trained classifier stages in a test image. The large survival rate variance suggests that subsequent stages tend to experience significant run-time workload variations. In addition to the AdaBoost algorithm, other popular face detection algorithms include a statistical method based algorithm [37] and a neural network based algorithm [38]. These face detection algorithms may deliver better detection performance than AdaBoost under different physical environment and facial scenarios. To achieve the highest possible face detection performance, it is highly desirable that the implementation platform seamlessly supports most or all of the three different algorithms. Because these algorithms involve completely different computations, a reconfigurable architecture provides a desirable computing option that can achieve high energy efficiency while providing programmability and parallelism.

2.2 Coarse-Grain Reconfigurable Architectures

Coarse-grain reconfigurable computing generally contain tens or hundreds of ALU-based logic blocks which include small amounts of memory. These blocks are interconnected in one [9] or two dimensions [8, 39] using bus-based or switched wiring which can be configured on a per-application basis. Coarse-grain reconfigurable computing devices can be found between FPGAs and multi-core processors on the spectrum of digital computing architectures. In general, SRAM-based FPGAs consist of large numbers of fine-grain single-bit logic blocks which are interconnected using programmable wiring [40]. Although these devices can be programmed to implement any logic function and they can be easily reconfigured in the field, their power consumption is often more than an order of magnitude greater than a corresponding ASIC [40]. In contrast,

Table 2: Overview of current and emerging embedded memory technologies.

	Current			Emerging			
Type	SRAM	eDRAM	eFlash	Cap-less DRAM	MRAM	PCRAM	FeRAM
Non-volatility	No	No	Yes	No	Yes	Yes	Yes
Cell structure	6T	1T-1C	1∼2T	1∼2T	1T-1MTJ	1T-1R	1T-1C
Cell size (F^2)	140	20∼30	8∼12	10∼30	10∼20	10∼20	10∼20
Scalability	Good	Bad	Good	Good	Good	Good	Medium
Incremental masks	0	4∼6	6∼8	0	3∼4	3∼4	3∼4
Endurance (cycles)	infinity	infinity	$10^4 \sim 10^5$	infinity	$> 10^{15}$	$> 10^8$	$> 10^8$
Read/write speed	1ns/1ns	5ns/5ns	100ns/5ms	5ns/5ns	10ns/10ns	20ns/50ns	10ns/10ns

multi-core processors generally contain tens to hundreds of full RISC processors with associated caches and memory management functionality.

Although not suited for general-purpose computation, coarse-grain reconfigurable devices can be effective for data-intensive applications which require custom data paths with time-varying connectivity. Over the past decade numerous academic [8,39,41] and commercial [42] coarse-grain architectures have been developed. The SmartCell architecture [8] consists of a two-dimensional array of 16-bit ALU-based cells. The blocks, which do not contain data memory, are grouped into four-cell blocks and interconnected using a series of partial crossbars. Although interesting, no compiler has been developed for the architecture and its application space has been limited to kernels such as FFT, DCT, and FIR filters. The similar AsAp array [39] uses a 2D array of 16-bit RISC processors which operate asynchronously. Each small core includes about 100 words of data and instruction memory. A circuit-switched interconnect which consists of multiplexers is used to exchange data values. A user programs each core individually and inter-core data exchanged is specified with a graphical user interface. In the Montium architecture [41], up to four compute blocks are grouped with 512 word memory banks. Each compute block contains five 16-bit ALUs and inter-block communication is performed via a crossbar. Only manual application mapping is reported. The commercial Ambric architecture [42] contains 336 32-bit RISC processors configured in a 2D mesh. Although this architecture does contain 128 Kbits per processor, this storage is unlikely to be sufficient for large image processing or communications applications. Also, the current programming tools require the definition of a Java program per processor and user definition of interprocessor communication.

In general, these architectures and similar previous coarse-grain reconfigurable architecture exhibit limitations in terms of compilation and access to significant amounts of local memory. These limitations lead to a significant energy penalty for the reconfigurations and storage needed for effective adaptive signal processing. In this proposal, we addresses these limitations through the use of high-density embedded non-volatile memory blocks which are interfaced to individual processors and the development of an automated compiler to map applications to the architecture using energy efficient configuration scheduling.

2.3 Emerging High-Density Embedded Memory Technologies

Table 2 compares the current and most promising emerging embedded memory technologies. SRAM has a large memory cell size due to its 6-T cell structure and is increasingly subject to leakage and process variability issues. Embedded DRAM (eDRAM) has a much smaller cell size and allows high speed accesses but it does not scale well due to capacitor fabrication concerns. Embedded flash (eFlash) is implemented with a NOR structure. This technology requires six to eight additional masks. Higher density NAND flash is typically not used as embedded memory due to a need for high voltage program/erase operations.

A variety of capacitor-less DRAM technologies are currently under consideration for next generation systems. These technoogies, which include thyristor RAM (T-RAM) [43] and Z-RAM [44], do not require capacitor fabrication leading to better scalability than conventional eDRAM. However, these memory technologies are inherently volatile and require costly refreshes. In the non-volatile emerging memory domain, the most promising candidates include phase-change RAM (PCRAM), magnetoresistive RAM (MRAM), and ferroelectric RAM (FeRAM). Because of their scalability, PCRAM and MRAM have received the most attention over the past several years and they have experienced significant technological progress. Compared with em-

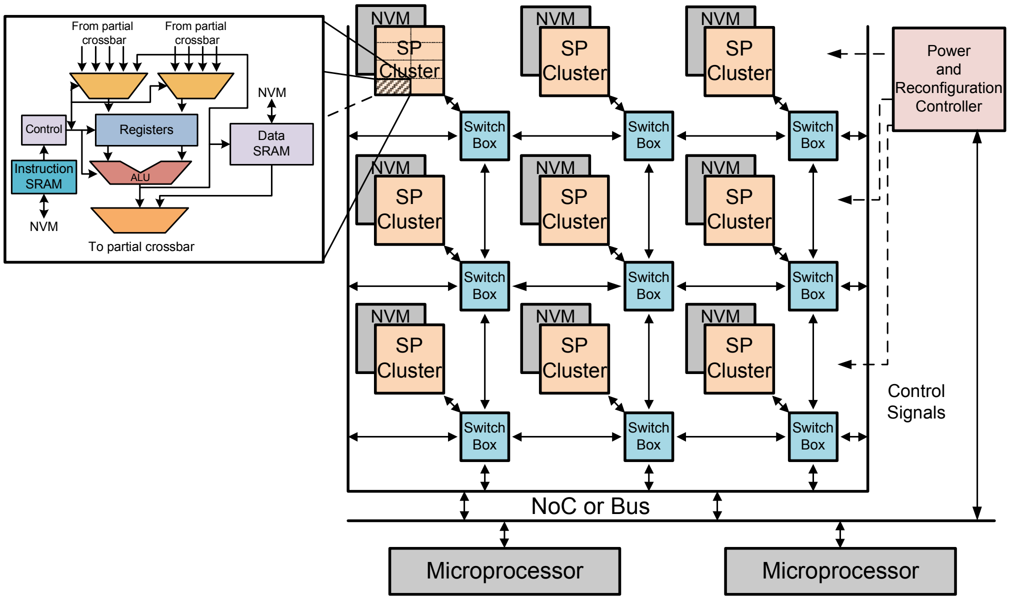


Figure 5: Proposed coarse-grain reconfigurable architecture with interface to embedded emerging non-volatile memory (NVM) such as MRAM or PCRAM. The dashed lines control the reconfiguration/shutdown of individual SP clusters and surrounding routing

bedded flash memory, both PCRAM and MRAM can achieve much higher storage density and demand 3∼4 fewer mask layers [45]. In addition, embedded flash memory write/erase operations require relatively high voltages (e.g., ∼10V) generated by charge pumps which can complicate power routing. In contrast, PCRAM and MRAM are readily programmed using a standard supply voltage. Since MRAM and PCRAM generally have higher scalability, higher endurance, and lower cost than eFlash, they are expected to play a crucial role in shaping future embedded memory, enabling unprecedented new opportunities in future integrated system design.

3 Proposed Research

Power savings in communication circuits may come from (i) dynamic algorithm selection (ii) efficient mapping of algorithms to hardware, (iii) VLSI design techniques and (iv) new low power circuit fabrics. In this proposal, we investigate low-power design techniques across the entire spectrum. The most novel aspects among these are the usage of MRAM/PCRAM memories to hold configuration bits for the reconfigurable architecture and primitive selection to optimize mapping of algorithms to hardware.

Our proposed work is split into several task sets. A new coarse-grain architecture with interfaces to high-density non-volatile memory will be developed. New memory architectural features which permit rapid retasking for signal processing applications are needed to support energy efficiency. Finally, several power management approaches for specific baseband and face recognition tasks on the new architecture will be developed.

3.1 Coarse-Grain Reconfigurable Architecture with Embedded High-Density Memory

Our envisioned coarse-grain architecture is designed with an eye towards high-density non-volatile memory access and straightforward compilation. As shown in Fig. 5, our proposed coarse-grain architecture is based on a two-dimension array of ALU-based signal processing (SP) clusters. The array interfaces to a network-on-chip or multiprocessor bus to provide access to standard multicore processors. A dedicated microcontroller or small state machine is located on the side of the array to provide control over power management activities in the array including the shutdown of unneeded clusters and voltage scaling of non-critical compute resources. The controller collects thermal, activity, and performance information from sensors located within the array clusters. Arrays containing 9 to 25 clusters will be initially evaluated via simulation. Inside the 2D array, clusters are connected using point-to-point wiring and multiplexer-based switchboxes (Fig. 5). Data are switched in 16-bit increments to lower the switch configuration memory overhead. As shown in Fig. 5, each SP cluster contains eight ALU-based cells connected using a partial crossbar. The outputs of each cell feed back into the partial crossbar and also have an interface to the inter-cluster interconnect. The inputs to the cluster connect to routing wires sourced by the switchboxes. Each cell contains a 32-bit ALU, a 512-word data memory, and a 512-word instruction memory. These small memories are directly connected to much larger multi-MB non-volatile memory (NVM) blocks which can be used to change their memory contents.

The interfaces between localized embedded (NVM) blocks and instruction and data memories and switchbox configuration storage are key innovations in this architecture. Unlike earlier coarse-grain reconfigurable architectures, these interfaces provide rapid access to large (multi-MB) non-volatile storage banks. Localized embedded NVM can dynamically reconfigure the small instruction and data memories under control of the external controller enabling changes in the signal processing algorithm implementation in response to the physical environment and/or signal characteristic changes (e.g. noise). Interconnections between clusters can be quickly customized in response to variations in cluster usage, allowing for rapid power reduction responses. The use of non-volatile memory provides a distinct power advantage for this architecture versus previous coarse-grain reconfigurable architectures. Not only is static power reduced, but portions of the array can be quickly reconfigured from power-saving shutdown mode.

One potential issue is the impact of the run-time reconfiguration latency on real-time signal processing system performance, i.e., the architecture reconfiguration may temporally pause the signal processing functions, leading to data loss and hence performance degradation. We have carried out a preliminary analysis which shows that such an impact is negligible and can easily be handled with an additional small buffer. First, we note that reconfiguration is only executed when the physical environment and/or signal characteristics significantly change. In the real world, such changes usually occur over a relatively long period, e.g., the wireless channel coherence time is typically at least on the order of ms and tens of ms, and the video signal characteristics vary at least on the order of tens of ms. In contrast, reconfiguration can be carried out very quickly by loading instructions from each individual NVM block to its associated SRAM block (as shown in Fig. 5), which can be done in parallel among all the clusters. For example, we assume each SRAM block is 64KB with a cache line size of 32B and an access latency of 2ns. Hence, only 4μs is needed to completely update 64KB SRAM. For high-speed wireless communication with a data rate of 200Mb/s, even assuming the channel coherence time is as low as 1ms, such a 4μs update latency is still over two orders of magnitude shorter and a buffer of a few KB can be readily employed to tolerate the 4μs update latency. We will further examine and fully take into account the performance impact of run-time reconfiguration in this research.

3.2 Compilation and Design Techniques

We will develop compilation and design techniques and tools to support the proposed coarse-grain reconfigurable architecture with interfaces to high-density embedded non-volatile memory. The proposed compilation environment builds upon existing compiler infrastructure and takes advantage of user interaction and communication estimation during the compilation process. The reconfigurable nature of the computation environment and novel power-aware features of the computation substrate necessitate two key areas of compiler innovation.

- Since the proposed architecture supports run-time reconfiguration, new communication dependence analyses and scheduling approaches are needed to determine appropriate computation balance.
- For best performance, computation tradeoffs should be evaluated in the context of feedback trace information based on sample data sets. While trace information is often used in processor instruction scheduling [46], to date little work has been performed in using trace information for energy-reduction for computation-intensive signal processing.

Our compiler infrastructure will be based on the SUIF-centric StreamIt [47] stream compiler from MIT and will support both traditional languages (C, C++) and stream-based languages (StreamIt). This initial infrastructure will be significantly enhanced to support the power-aware reconfiguration features. The proposed compiler contains a series of steps: front-end processing, functional block partitioning, computation and communication scheduling, and back-end, device-specific compilation. Notable aspects of the compiler include support for stream input language, an extensible library of independent, reorderable compilation optimizations, and output to a range of back-ends. Although the StreamIt compiler contains useful structures for front-end parsing, intermediate form representation, and back end processing, completely new mapping algorithms, estimation approaches, and internal representations will be necessary to support the proposed coarse-grain reconfigurable architecture.

Designs will be represented in a streaming language [47] where the overall computation is composed of several different filters organized into stream-graph control-flow structures. Filters represent the basic stream graph computational units. Edges represent the communication of values (messages) between filters. Following parsing, these input codes are translated to a unified abstract syntax tree (AST) representation. Following annotation, AST representations are converted to a graph-based intermediate form that represents functions at both a high and low level. This allows for rapid evaluation of individual constructs. Once estimates are obtained, optimizations, such as loop splitting across multiple cells may be considered to promote parallelism.

To analyze possible reconfiguration opportunities, a computation and communication graph will be extracted from the intermediate representation of the design to represent internal data dependencies. Each stream will have a source and destination cell and an associated maximum bandwidth requirement. High-bandwidth stream-based computation represented in the graph will be allocated using a heuristic scheduling algorithm. For data streams amenable to scheduled computation, data transforms follow a sequence of transactions between nearest neighbors. As a final compilation step, the internal code representation can be used to directly generate code for target cells. A modified version of the Computer System Description Language (CSDL) will be used to describe specific targets.

3.3 Embedded Memory System Design

An emerging high-density non-volatile memory, such as MRAM and PCRAM, will be used in the envisioned signal processing implementation platform to both store configuration data and realize general-purpose embedded memory. As illustrated in Fig. 5, arrays of MRAM or PCRAM are used as general-purpose embedded memory for all the distributed SP clusters, hence the general-purpose embedded memory organization and structures are determined by the underlying coarse-grain reconfigurable architecture. In contrast, memory system design for configuration data storage has a much higher flexibility, which will be fully investigated in this proposal.

Existing FPGA devices generally use embedded flash memory for configuration data storage in two different ways: (i) fully distributed embedded flash memory cells which store logic and interconnect configuration data in place of SRAM (e.g., Actel FPGAs [48]), and (ii) standard SRAM-based FPGAs which are supplemented with an embedded flash memory block that stores configuration data. The data is loaded from this centralized memory into distributed SRAM cells when devices are powered up (e.g., nonvolatile FPGA devices from Lattice [49]). To integrate these emerging memory technologies with reconfigurable architectures, the tradeoff between configuration data stored in distributed and centralized memory must be addressed. In contrast to eFlash memory, the emerging memory technologies rely on resistance modulation to store data, while logic components and interconnect configuration rely on full-swing control voltage signals. Therefore, configuration data must be converted from the resistance domain to the voltage domain, which is harder to achieve when memory cells are distributed across the chip.

As a result, our work will mainly focus on the use of centralized memory for configuration data storage. To provide some distribution, we will consider the development of a *partially-centralized design style*. The reconfigurable computation fabric will be partitioned into several regions, where each region has its own private embedded non-volatile memory block. We believe that a partially centralized design strategy is the most promising option because: (1) It can naturally embrace the process variation and defects of highly scaled memory technologies through the use of traditional memory fault tolerance techniques; (2) It can readily store multiple sets of configurations, and even employ simple lossless data compression to further improve the effective capacity; (3) By using standard SRAM cells to directly hold the **active** configuration data, we can achieve the shortest run-time reconfiguration latency; (4) Such a partitioned approach can naturally support run-time power-gating to reduce static leakage power. In addition, such a partially centralized design strategy can potentially enable the use of distributed DRAM cells instead of SRAM cells to hold the **active**

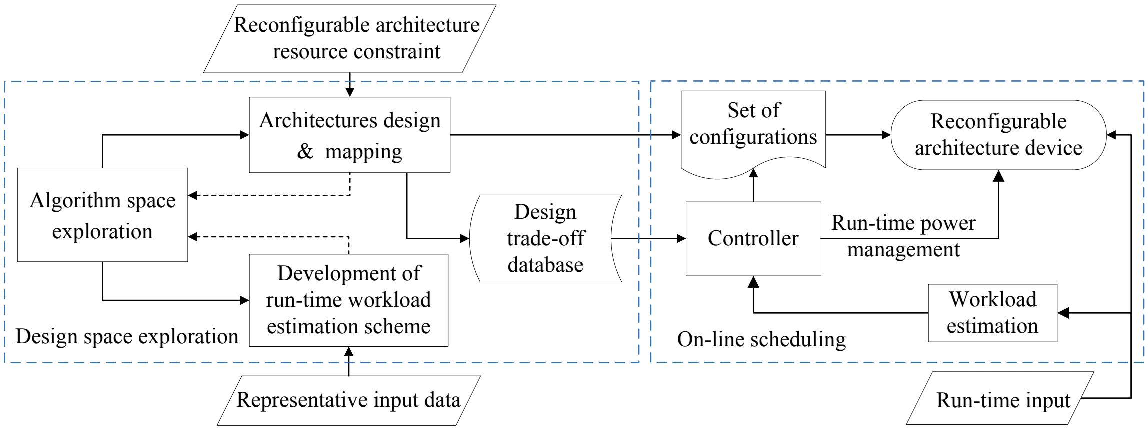

Reconfigurable architecture resource constraint
Architectures design & mapping
Algorithm space exploration
Design trade-off database
Development of run-time workload estimation scheme
Design space exploration
Representative input data
Set of configurations
Reconfigurable architecture device
Controller
Run-time power management
Workload estimation
On-line scheduling
Run-time input

Figure 6: Proposed design flow for implementing signal processing tasks on the reconfigurable architecture.

configuration. Although DRAM cells occupy much less silicon area than SRAM cells, DRAM cells are not used in current FPGA design practice because of the destructive nature of DRAM read operations and the need for refreshes due to charge leakage. The availability of localized, embedded non-volatile memory storage may help enable on-chip DRAM self-refresh.

To summarize, the following specific memory tasks will be targeted:

- *Memory circuits and systems modeling*: Based upon the widely used memory modeling tool Cacti [50] and the latest device and circuit models, a Cacti-based MRAM/PCRAM modeling capability will be developed. These models will estimate MRAM and PCRAM system metrics enabling effective system design space exploration. The interface of these circuits to the reconfigurable array and the associated impact on energy consumption will also be considered.
- *Configuration data storage system optimization*: Since configuration data are primarily read-dominated, the memory write latency is a far less important than the memory read latency. We will investigate how this imbalance can be exploited to optimize memory fault tolerance and performance. For example, MRAM cells are directly subject to a write latency versus read latency trade-off based on memory cell size. The partially centralized design style induces a trade-off between memory partition granularity and storage density. We will fully explore the memory granularity trade-offs and their impact on system performance and energy consumption.

3.4 Signal Processing System Implementation Framework

To evaluate the effects of energy-aware coarse-grain array reconfiguration based on signal processing algorithm variation, we will develop a systematic cross-layer design methodology. The concept, illustrated in Fig. 6, contains two phases: (1) off-line design space exploration, and (2) on-line scheduling with run-time power management.

Given a signal processing application, it is first necessary to select and implement specific algorithms that meet algorithm-level performance metrics. Due to run-time signal variations, algorithm-level signal processing performance metrics are typically specified under a worst-case scenario. Generally, algorithms which meet this scenario require the highest computational complexity. Although a lower complexity algorithm may not achieve the specified signal processing performance metric in the worst-case, it may be able to achieve or approach the metric in the average case. Algorithm implementations with different levels of parallelism may be developed to achieve a range of performance with different energy characteristics within available resource constraints. Multiple options will be available in the configuration database for rapid reconfiguration in the event of changing environmental conditions.

On-line scheduling selects a configuration at run time from the set obtained during the design phase. Configuration assignment to the coarse-grain array will be managed by an embedded controller based on two sets of input. As illustrated in Fig. 6, these inputs include estimated run-time computational complexity based on input signal characteristics, and information in the implementation trade-off database obtained during

	CLS100	CLS64	CLS32	CLS16
Number of LUTs	46104	28553	14509	7486
Number of registers	34101	25601	18801	15401
Number of DSPs	300	192	96	48
Power consumption (mW)	7957	5424	3276	2506

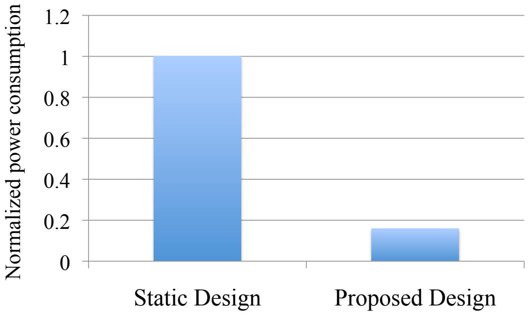


Figure 7: Preliminary case study on AdaBoost-based face detection to evaluate the energy saving potential.

the design space exploration phase. The controller will dynamically reconfigure the appropriate portions of the device based on the target application. A heuristic list scheduler will initially be used to determine when configuration changes should be considered. Another important task of the controller is to accordingly carry out run-time power management. Based upon the current configuration and run-time environment, the controller may adaptively apply clock-gating or power-gating on certain portion of the circuits in order to minimize power consumption. Therefore, the major research tasks in this context include:

- We will develop the on-line and off-line implementation framework. Although the design space exploration phase is highly application specific, appropriate design guidelines will be created for carrying out the recursive algorithm and architecture design procedures.
- To ensure optimized results, the on-line scheduling will consider implementation overheads induced by computation, system reconfiguration, and memory access and data transfer. We will extend Wattch [51] to combine microarchitectural power dissipation with power dissipation from dynamic reconfiguration. Since the reconfigurable device may be running several tasks for different applications at the same time, the available silicon source for any specific task may vary during run time, which should also be appropriately handled by the controller. It is expected that heuristic algorithms can be utilized for scheduling.
- We will investigate how to most effectively apply clock-gating and power-gating on the proposed coarse-grain reconfigurable architecture. The corresponding interface between the reconfigurable array and the embedded controller will also be investigated.

3.5 Preliminary Results

To quantitatively evaluate the potential of exploiting run-time workload variations to improve signal processing implementation energy efficiency both AdaBoost-based face detection and Adapative Viterbi decoding [52] were examined. In an initial experiment, we carried out a preliminary study that implements the AdaBoost-based face detection as described in Section 2.1.2 on an Altera Stratix III EP3 FPGA device consisting of 54,000 look-up tables (LUTs) and 488 DSPs. The detector contains 25 stages, which are implemented by two phases in a time-division multiplexed manner. A total of 100 classifiers were implemented for the first phase (denoted by *CLS100*). The workload of the second phase varies from image to image, thus we implement three extra FPGA configurations to cover a sufficiently large range of energy and speed trade offs. These implementations are referred to as *CLS64*, *CLS32*, and *CLS16*, which realize 64, 32, and 16 classifiers, respectively. The results shown in Fig. 7 assume the FPGA device contains enough embedded memory to store all the configurations. Run-time reconfiguration energy cost was determined using Spice simulations. The results in the figure are based upon the power estimation results and face detection run-time workload variations from MIT+CMU test images [35]. The proposed design, which leverages run-time workload variations shows a 5× energy reduction versus a static design.

Additional experiments using an FPGA [7] determined the possible power savings of dynamically reconfiguring a Adaptive Viterbi Algorithm decoder. As shown in Table 3 significant power savings can be achieved if the decoder is dynamically reconfigured in response to changing channel noise. Constraint lengths of between 4 and 14 are considered. In this case, channel decode rate is maintained by swapping in less (low noise) or more (higher noise) powerful decoders as channel noise characteristics change. Decoder perfor-

	Avg. Speed (Kbps)	Decode time (sec)	Reconfigs. required (out of 10,000)	Reconfig. Overhead (sec)	Avg. Power (mW)
XC4036XL-08					
Static	105.9	23617	0	0	135.7
Dynamic	105.9	23617	7065	282.6	98.8
XCV1000-04					
Static	61.7	40521	0	0	1611.0
Dynamic	61.7	40521	7007	107.2	505.3

Table 3: Static decoder versus dynamically-reconfigurable decoder power consumption

mance is maintained by using a small buffer during reconfiguration. Our coarse-grain architecture will allow for significantly reduced overall power characteristics versus the FPGA values noted here.

3.6 Putting Everything Together

This proposal aims to cohesively integrate emerging embedded memory technology and coarse-grain reconfigurable architecture to develop highly flexible and energy-efficient signal processing platform for future mobile devices. This new implementation vehicle will assessed via a comprehensive simulation and modeling infrastructure which incorporate (1) a CACTI-based tool set for MRAM and PCRAM modeling, (2) a SUIF-centric StreamIt compilation and simulation framework for the proposed coarse-grain architecture, and (3) a power estimation tool for the reconfigurable architecture and associated embedded memory which can be used to assess power management techniques As discussed in Section 1, wireless communication baseband and multimedia/vision signal processing will be evaluated with this simulation and modeling infrastructure. In particular, long term evolution (LTE), the latest 4G mobile communication standard, and face detection will be used as test cases. Specific tasks include:

- We will carry out comprehensive LTE baseband signal processing simulations in C covering a wide spectrum of environmental and algorithm configurations. These configurations will include radio link quality, user data rate, MIMO signal detection and ECC decoding algorithms for LTE. For face detection, we will carry out comprehensive simulations over a wide range of training and test images using well-established algorithms including the AdaBoost algorithm, statistical method based algorithm, and neural network based algorithm.
- Following comprehensive algorithm-level analysis, a reconfigurable coarse-grain architecture will be developed that interfaces to high-capacity non-volatile memory. A scheduler, which considers varying environmental and signal characteristics for dynamic reconfiguration will be integrated into StreamIt and the run-time system, as described in Section 3.4.
- Architecture-level simulations will be carried out using the simulation/modeling infrastructure to evaluate the energy consumption of the implemented LTE baseband signal processing and face detection tasks. Comparisons versus comparable ASIC and FPGA implementations are expected.

Based on our analysis in Section 3.1, we expect that reconfiguration will have negligible performance impact on communication and AdaBoost applications. As the research progresses we will maintain in communication with industrial contacts at leading companies for guidance. Initial contact with representatives at Broadcom and Qualcomm has been positive.

4 Research Team and Time Table

The PIs associated with this project have extensive research records in signal processing and memory circuits/systems (Zhang), reconfigurable architectures (Tessier), and power-aware computer architecture (Kundu). The close interaction of this team will facilitate the successful completion of the project. Table 4 shows the task partitioning among the PIs. The project will be conducted over a three year time period.

We plan to have regular “virtual” working group meetings via video teleconference (Skype) every two weeks to bring together the PIs and graduate students. Face-to-face meetings will take place every two months. The close proximity of UMass and RPI will facilitate these meetings.

Table 4: Project tasks and responsibilities.

Year	Task	PIs
1	Initial coarse-grain reconfigurable architecture, simulator, and compiler work.	Tessier
1	Initial mapping of baseband and face recognition applications.	Tessier/ Zhang
1	CACTI-based MRAM and PCRAM modeling tool development; MRAM/PCRAM-based configuration data storage implementation strategies and system optimization space exploration.	Zhang/ Kundu
1	Investigation of power management schemes and development of microarchitectural power estimation capability.	Kundu
2	Development and evaluation of coarse-grain reconfigurable architecture and supporting compilation and simulation tools.	Tessier
2	Development of signal processing system implementation framework with support of run-time power management.	Zhang/ Kundu
2	Design space exploration for baseband processing when using MRAM/PCRAM as general-purpose embedded memory for each SP cluster.	Kundu/ Tessier
3	Development of a comprehensive simulation and modeling infrastructure by integrating various simulation and modeling tool sets and developed design techniques.	All
3	Apply the developed design methodologies and simulation infrastructure to develop energy efficient implementation of LTE wireless communication baseband signal processing and face detection.	All
3	Course integration using developed simulator and compilation techniques.	All

5 Broader Impact

The broad nature of the proposal makes it appropriate for a diverse educational program that has impacts for the research community and society at large. Material from the project will be integrated in UMass course ECE636: Reconfigurable Computing, a graduate course taken by 20 students a year, and RPI course ECSE 6680: Advanced VLSI Design, a graduate course taken by 15 students a year. The availability of the tools will allow students to do experiments with state-of-the-art technology and develop new technologies.

(1) *Participation of Under-Represented Groups*: The University of Massachusetts offers significant opportunities for undergraduates to pursue independent research projects outside of the standard course flow. Since PI Tessier's arrival at UMass he has integrated several undergraduates into his research program. Recently, PI Tessier has supervised two undergraduate students that have been part of women and minority summer programs. The first of these efforts, performed in conjunction with the UMass Women in Engineering Program (WEP), resulted in a conference paper [53]. A second research project performed as part of the UMass Minority Engineering Program (MEP) is currently leading to a publication. In 2004, PI Tessier completed supervision of an African-American Master's degree student, Lilian Atieno, who was part of the NSF-funded Northeast Alliance Program (NEA). This program provides research opportunities to graduate students from under-represented minority groups at UMass. Lilian did an outstanding job on her research and recently published a paper [54] at the highly competitive International Symposium on FPGAs (paper accept rate of about 20%).

(2) *Research Infrastructure Dissemination for Education and Training*: We plan to actively promote tool re-use and further development both at UMass and RPI and by other researchers. We will make our developed comprehensive simulation and modeling infrastructure open to the public by posting the entire source code and associated document on a WiKi website. We plan to recruit undergraduate students, especially under-represented groups, to participate in the source code and document maintenance and WiKi website development.

(3) *Outreach to High School Students*: The PIs will actively initiate and enhance their participation in

outreach programs which will motivate high school students to consider higher education in science and engineering. Since 2004, PI Zhang has participated the New Visions Math, Engineering, Technology and Science (METS) program that brings high school seniors (mostly from rural areas) to the Rensselaer campus five mornings a week for the school year to attend first-year science and engineering classes and participate in research at School of Engineering laboratories. PI Zhang has given a series of presentations through this program entitled The Information Age: Where Integrated Circuits Meet Digital Signal Processing. These presentations show students how integrated circuits and signal processing has changed our life. The proposers are enthusiastic to expand this outreach activity to impact students from traditionally under-represented groups, such as minority students and female students. We plan to enhance the outreach presentations made by PI Zhang by incorporating more cutting-edge design examples and research challenges including the ones addressed by this proposal. The materials will be shared within the team and used at all appropriate outreach programs at UMass and RPI. Additionally, we will encourage and provide opportunities for the undergraduate students involved in this research project to actively participate in outreach activities.

6 Results from Prior NSF Support

PI Tessier is currently conducting research in field-programmable systems and embedded system design. Prof. Tessier is currently PI of NSF project CNS-083194: Network Virtualization Using Dynamic FPGA Reconfiguration (08/01/08 - 07/31/12, $300,000) which aims to implement multiple network routers in the same reconfigurable fabric. PI Tessier has also received major research funding from the Semiconductor Research Corporation and leading commercial research labs. Prof. Tessier received the UMass College of Engineering Outstanding Junior Faculty and Outstanding Teacher awards for the 2002-2003 academic year.

PI Kundu has recently concluded NSF funded projects on Improving Reliability and Availability of Chip Multiprocessors (August 2008-July 2010). These grants resuted in more than 20 publications. Kundu also has current NSF-SRC funding on A Design Framework for Improving Reliability, Debug and Security of Multi-Core Systems and has published recent papers on architectural debug techniques. Kundu recently co-authored a book on Design for Manufacturability (ISBN 978-0071635196) and won a best paper award at DATE09.

PI Zhang has served as PI of multiple NSF grants. His most recent completed grant is ECS-0522457, Baseband Signal Processing Algorithm and VLSI Architecture Co-Design Methodologies for Wireless MIMO Transceivers, $180,000, 09/01/05 to 08/31/08. This project focused on the development of algorithm and VLSI architecture co-design methodologies for implementing baseband signal processing building blocks in high data rate wireless multiple-input multiple-output (MIMO) transceivers. In this work he developed soft-output MIMO signal detector designs [55–58] and trellis-based decoder designs [59–61] in wireless MIMO communication systems. He also investigated an LDPC-based MIMO system design [62, 63].

7 Summary

The proposal aims to cohesively integrate dynamically-adaptable communications and image processing algorithms with coarse-grain reconfigurable architectures which contain high-density non-volatile memory to fundamentally improve energy efficiency in future mobile devices. The proposed coarse-grain architecture provides the capability to dynamically exploit significant run-time computational workload variation and large algorithm design freedom inherent in most signal processing tasks, effectively enabling just-enough computation. The approach embraces the memory-intensive and time-varying nature of many signal processing tasks, which can be leveraged to reduce overall system energy consumption. In addition, these approaches make it feasible to apply aggressive run-time power management on reconfigurable architectures to minimize static energy consumption. To fully evaluate the potential of these new opportunities, we propose to collaboratively develop a comprehensive design methodology including circuits, architectures, compiler, and run-time system. These technologies will be integrated together to demonstrate energy-efficient wireless communication baseband signal processing and computer vision. If successful, this collaborative research program will improve mobile device energy efficiency for these applications by one order of magnitude and provide a new breed of reconfigurable signal processing solution to enable future pervasive mobile devices.