FORM 101 – PROPOSAL

1. Recent Progress

Mitigating computer system failure has been a challenge since the dawn of modern computing, but a number of factors are now converging such that resilience (a system's capacity to operate in the presence of failure due to manufacturing defect or faults in the field) is now a first-class design constraint. Semiconductor scaling has made it possible to introduce computers into every aspect of our lives. Scaling, which makes the wires and transistors of integrated circuits increasingly miniature, unfortunately also makes them increasingly vulnerable to myriad sources of failure: small, low-power, high-performance devices tend to be defect-prone, susceptible to transient error, and vulnerable to performance degradation as well as early total breakdown [1].

While the literature abounds with techniques for improving resilience, these techniques are poorly suited for today's emerging applications. Traditional approaches are expensive and power-hungry (*e.g.*, replicating software tasks [2] or hardware resources [3], or both [4]), while emerging applications are primarily cost- and power-constrained: applying the former to the latter is intractable, as the advantages of scaling, which make the applications possible, would be reversed. New computer engineering approaches are needed that enable designers to identify salient vulnerabilities and devise cost-effective solutions. If resilience is not addressed, the resulting costs will limit the introduction of new products and jeopardize the steady growth on which the information technology economy depends.

To these ends, I have done recent research developing both automation techniques for optimizing system lifetime and manufacturing yield, as well as computer system architectures for reducing the cost of safety-critical systems. In my dissertation, I developed a technique for automatically allocating *slack*—execution cycles and memory address space not required to meet initial performance requirements—in embedded network-on-chip multi-processor systems-on-chips (NoC-based MPSoCs). In the event that processors, memories or switches are defective at manufacturing time or fail in the field, slack accommodates task and data re-mapping and communication re-routing so systems may continue to function and satisfy performance constraints [5][6]. My technique, Critical Quantity Slack Allocation (CQSA), identifies those quantities of slack expected to have the most significant effect on opportunity to re-map tasks and data. Since tasks and data have specific computational and storage requirements, allocating too little achieves nothing; on the other hand, allocating too much degrades lifetime and yield by increasing temperature and area. I demonstrated that, by leveraging information about the quanta of slack needed to replace individual and groups of components, CQSA efficiently finds designs that achieve near-optimal lifetime-cost and yield-cost trade-offs.

More recently, I've conducted research developing system-level architectures for safety-critical systems that execute a mix of critical and non-critical tasks [7][8]. Traditional safety-critical systems employ *n*-modular redundancy (*n*MR), replicating resources and applications so errors are quickly detected by correctly executing copies. While *n*MR can ensure safe execution, it does so at great cost, traditionally requiring that (a) only critical tasks execute on critical task resources, and (b) critical tasks execute in lockstep. To reduce costs without compromising the low-latency error detection and correction that *n*MR techniques enjoy, I have developed *On-demand Redundancy* (ODR). ODR improves the efficiency and reliability of dual-modular redundancy (DMR) by relaxing the above constraints. Under ODR, multiple processors still execute multiple copies of critical tasks, but when processing units are not needed for critical tasks, they are used to execute non-critical tasks. This exposes more resources for non-critical task execution, reducing the overhead of traditional DMR.

2. Literature Review

There is an emerging body of research investigating approaches for addressing system resilience in embedded and general-purpose computer systems. New work has recently been done to address manufacturing defects and permanent failures in embedded and general-purpose computer systems. Zhu *et al*. [9]. Glass *et al*. [10], and Huang and Xu [11] propose design automation techniques for embedded systems that allocate and organize processor resources, and assign tasks to each, such that performance, cost and system lifetime are co-optimized. Shivakumar *et al*. [12] and Schuchman and Vijaykumar [13] both introduce architectural techniques for improving the yield of general-purpose processors. While these approaches are effective, our proposed research goes further: we aim to jointly optimize lifetime and yield, thereby exposing significant new challenges in the design- and run-time management of redundant resources, while uncovering new opportunity to improve both.

Other research focuses on addressing transient failure. Golander *et al*. improve the flexibility of spatially redundant systems by relaxing the constraint that pairs of redundant resources execute in lockstep [14]. Timor *et al*. improves the utilization of single-core temporal redundancy by extending the microarchitecture to support executing instructions twice [15]. Subramanyan *et al*. accelerate redundant software execution on chip multiprocessors by passing execution hints from leading cores to redundant tasks on a trailing core [16]. Though these techniques have important applications in high-performance computing, they fall short of providing the guarantees needed for safety-critical systems.

Recent research has also proposed techniques for addressing safety-critical systems executing software with mixed-criticality. Eles *et al*. presents a fault-tolerant scheduling technique for a single core that enforces hard deadlines while selectively enforcing soft deadlines [17]. Our proposed research additionally considers multi-core embedded systems, where the system architecture and scheduling techniques are expected to be a function of the application domain, enabling further optimization.

3. Objectives

The long-term objective of this research is to develop the architectures and automation techniques needed to support the efficient, effective, design of resilient computer systems. Because of the size and complexity of modern computer systems, design processes must be efficient, quickly identifying system instances based on designer inputs. The identified instances must also be effective, achieving application-specific near-optimal trade-offs of (a) system resilience to manufacturing defects, transient upsets, and performance degradation; and, (b) system performance, power and cost (*e.g.*, die area). This research will focus on the design of embedded systems, and in particular, the influence that differences in application domain have on differences in how best to achieve efficiency (*e.g.*, what architectures and resilience techniques to employ), and what constitutes effectiveness (*e.g.*, lifetime improvement under cost constraints, or power minimization under safety constraints). In pursuit of this goal, my research program will focus on three short-term objectives:

1. **Developing new, resilience-aware hardware/software (HW/SW) partitioning strategies;**
2. **Developing resilient system-level architectures and organizations, specifically targeting (a) lifetime and yield improvement and (b) low-cost safety-critical systems; and,**
3. **Developing parallel automation techniques which co-optimize systems at these various levels of design abstraction.**

The above research will lay a firm foundation of general principles of resilient computing for future research and development efforts targeting specific application domains, and additional opportunity for resilience co-optimization in the microarchitectural and circuit-level design of computer systems.

4. Methodology

4.1 Resiliency-aware Hardware/Software Partitioning and Assignment. We propose to develop new resilience-aware HW/SW partitioning techniques that, in addition to ensuring power and performance constraints can be satisfied by later stages of design, also consider resilience goals. Traditional HW/SW partitioning, given an application, first determines how best to divide an application into tasks, and then assigns either dedicated, special-purpose or software to perform those tasks. Dedicated hardware improves performance at the expense of die area, while software reduces system cost at the expense of performance and power. Our hypothesis is that without directly evaluating resilience, HW/SW partitioning can't cost-effectively achieve resilience goals.

HW/SW partitioning takes on new significance in the context of resilient system design. In my dissertation research, I observed that the initial assignment of tasks and data to resources played an important role in determining not only (a) the lifetime of the system, but also (b) what opportunity existed for improving lifetime or yield. By determining the granularity at which (i) slack could be allocated and (ii) tasks and data could be re-mapped in response to failure, the partitioning of an application into tasks and subsequent assignment to processors and memories determined what redundancy was required to survive different sorts of failures.

Extending tools I previously developed for the system-level design and evaluation of MPSoCs [5][6][8][18], PhD-A will explore ways of dividing applications into dedicated hardware tasks, and software tasks so that opportunity to improve system resilience is efficiently exposed to later design stages. Software tasks will be tunable for assignment to a spectrum of processor types, *e.g.*, general-purpose, digital signal processing, and domain-specific. PhD-A will also explore the application of techniques from reconfigurable computing (such as patchable ASICs [19]) to enable dedicated, special-purpose hardware to execute tasks beyond those it was specifically designed for.

PhD-A will adapt previously developed evaluation techniques in order to evaluate application partitioning and software task assignment for yield, reliability, and lifetime. This will in turn require high-level evaluation of system power dissipation [20] and thermal characteristics (*e.g.*, using HotSpot [21], as system lifetime is exponentially dependent on temperature. Early area estimation (available using floor planning) will also be important for the purposes of yield estimation [22].

4.2 Resilient System Architecture. To make resilient computer systems affordable, we propose developing system-level redundancy techniques that are specifically tailored for two distinct use cases: application-specific safety-critical systems, and embedded systems-on-chips. Our hypothesis is that in each case, specifically developing architectures and automation techniques suited to the particular resilience requirements of an application or domain will expose opportunity for cost-reduction.

Application-specific Safety-critical Systems. In the context of safety-critical systems, we will investigate the relationships between application resilience constraints (*e.g,*. low-cost safety, vs. long-lifetime safety) and embedded system architecture (how much redundancy, at what level of granularity, and allocated according to what policy). A safety- or mission-critical system is one where failure has hazardous consequences for humans or other life and equipment. Examples include automotive embedded systems and medical devices. While these classes of systems have in common a requirement that no harm come to human users (a hard safety constraint), other aspects of resilience differ between the two application domains. For instance, automotive systems can often be safely shutdown when they fail (*e.g.*, with a switch to a mechanical backup system), and repaired at a later date. While repair is inconvenient and expensive, in addition to being a marketing liability, it is at the least practical. Furthermore, the repair process does not put human users at risk. Not so with implanted medical devices: it may not be possible to shut them down safely in the field; and, repair or replacement may involve surgery, which comes with its own risks in addition to costs.

The resulting differences in resilience constraints for each type of system have clear consequences for design. In an automotive context, minimizing cost is important: safety must be achieved with as few resources as possible; the rate of repair is a parameter to be tuned with other constraints. In the medical device context, cost is less important, but area and power are more important; the rate of repair must be minimized (long system lifetimes are desirable), but the systems must be small (for ease of implantation), and low power (as power dissipation heats, and therefore damages, surrounding tissue).

In prior work, we observed that the relative mix of critical- and non-critical tasks in an application determined what sort of architecture would be performance-cost optimal: different architectures were more or less easily utilized depending on the mix of tasks [7][8]. MEng-A will extend this research and explore the effect that widely divergent resilience goals (such as those of automotive and medical systems) have on safety-critical architecture. The essential observation to guide this research is that while the safety requirement is of principle importance, secondary objectives (performance, power, cost, lifetime) impose important constraints on system design. As a result, MEng-A will develop new architectures, and integrate a number of existing system evaluation techniques [5][8][18] in order to produce designs that satisfy resilience goals efficiently in contexts where the relative importance of these secondary objectives is different. Once we have identified architectural techniques appropriate for application domains (which have a specific set of relative constraints), PhD-A will extend my prior research [5][8][18] to further develop automation techniques for determining, given a particular application instance, the set of design instances that achieve the best trade-offs.

Improving the Lifetime and Yield of Embedded Systems-on-Chips. In the context of embedded systems without safety requirements, we will investigate architectures and redundancy allocation strategies for co-optimizing lifetime and yield. In my dissertation research, I observed that system-level redundancy techniques that improve lifetime (*e.g.*, slack allocation) tend to improve yield as well [23]. While differences in failure model lead to differences in the relative lifetime and yield improvement achieved by architecture (*e.g.*, SRAM memories are unlikely to fail in the field because of low-cost row and column redundancy, while defects in the control logic of memories can make entire arrays unusable), I showed that optimal lifetime improvement generally results in near-optimal yield improvement, and vice versa. This raises two important questions, however: (a) what system architectures are best suited to application-specific lifetime-yield co-optimization, and (b) how should redundant resources be utilized to address systems expected to face manufacturing defects as well as permanent failures?

We will investigate these issues by developing architectures and redundancy allocation approaches that simultaneously balance application-specific lifetime requirements and yield improvement. PhD-B will make use of and extend previously developed evaluation techniques and design automation approaches to explore the relationship between lifetime and yield improvement, focusing on architectural strategies that improve both without compromising either. The result of this research will be new system-level architecture strategies that determine: (a) on-chip network topology, *e.g.*, point-to-point (optimal performance, power and temperature), mesh (optimal recovery), or hierarchical (improving recovery relative to point-to-point networks without the performance, power and temperature penalty of a mesh); and, (b) system organization, especially in the context of custom- and semi-custom resources (*e.g.*, how should such resources be distributed throughout an on-chip network to ensure that common-mode failures, such as the loss of connectivity, don't eliminate both the primary and redundant copy of a resource). In order to achieve this, PhD-B will (a) develop system-level design strategies for interconnecting and organizing on-chip processors, memories and custom logic. Custom design tools [5][6][18] will subsequently be extended to consider these design strategies (including templates for on-chip network topologies, rules for the relative placement of redundant resources in the network, etc.). For the purpose of design evaluation, the tools will be further extended to automatically optimize the on-chip interconnect for (i) lifetime, (ii) yield, and (iii) both.

To support this effort, we will develop policies to assess life-cycle costs to determine how to utilize redundant resources to both improve yield and extend lifetime. Resources utilized to improve yield cannot subsequently be used to extend lifetime. In this case, designers must therefore balance the complex trade-offs between allocating and utilizing redundancy to improve yield, and holding redundant resources in reserve to extend lifetime. MEng-B will develop a framework for exploring the effect of various policies (*e.g.*, always repair defects, never repair defects, and policies in between). An essential part of this framework will be a new cost metric MEng-B will develop that captures the costs of repair and replacement when failures occur earlier than the required lifetime of a given system.

4.3 Parallel Cross-layer Co-optimization. To make the design of efficient, resilient computer system design tractable, we propose adapting existing evaluation and exploration techniques as well as those proposed above for parallel execution. As the number of integrated system-level components in a system grows exponentially, the scalability of system-level modeling and evaluation approaches becomes a critical challenge. Scaling is a greater problem for resiliency evaluation than other metrics, as its evaluation often relies on statistical estimation driven by repeated evaluation of the metrics on which it depends, such as performance, power, cost, and yield.

While the scalability of resilience estimation (and therefore design processes) poses a significant challenge, it also presents a unique opportunity: there is abundant parallelism in each design process, from the repeated calculations for each system element under evaluation (the inner loop of any design space exploration), to the various HW/SW partitioning alternatives available given a particular application (explored in the outermost loop). For multi-core embedded systems (which consequently have many possible organizations), it is easy (from a design perspective) to integrate redundancy, but difficult to do so cost-effectively (*e.g.*, from an area or power perspective): there are too many alternatives to evaluate without advanced exploration techniques. We will therefore aggressively parallelize design evaluation and design space exploration, making it possible to find the best possible designs without compromising an important business metric in embedded systems: time-to-market.

Using a framework such as OpenMP or OpenCL, each student will develop parallel algorithms. Thread-parallel HW/SW partitioning will be conducted, concurrently evaluating multiple alternatives (for instance, using distributed simulated annealing [24] or genetic programming [25]). For a given HW/SW partitioning (PhD-A), further thread-parallel design evaluation will be employed to (a) concurrently evaluate multiple network architectures and organizations (PhD-B), and (b) concurrently evaluate multiple instances of the same design (PhD-B, using Monte Carlo simulation). Each design evaluation will be computed in data-parallel fashion, leveraging general-purpose graphics processing units (GPGPUs) to accelerate component-level thermal modeling, and failure distribution manipulation (PhD-A, PhD-B, MEng-A, MEng-B), etc. The effectiveness of these techniques will be evaluated using state-of-the-art computing resources at the McGill site of the *Consortium Laval UQAM McGill and Eastern Quebec* (CLUMEQ), a Compute Canada center for High Performance Computing.

5. Impact

Today, most computers people interact with are embedded computers; they are the household and office appliances that enable our productivity, the communication devices that keep us in touch with colleagues and family, the transportation we rely upon, and the medical devices that support our health and assist our healing. This proliferation of embedded systems has been driven by the cost reduction born of miniaturization; however, the very cost improvement that has enabled this advancement is threatened by myriad resilience challenges. The proposed research program will ensure that costs remain low. Yield improvement research will apply redundancy to reduce the cost of manufacturing semiconductor systems; safety-critical architecture research will ensure that during their lifetime, systems will achieve application-specific reliability goals while minimizing redundancy; and, system lifetime extension research will ensure systems meet lifetime requirements at as low cost as possible. These results will be disseminated in high-profile peer-reviewed conference proceedings, journals.

The proposed program will also produce highly trained personnel, training graduate students in computer system architecture and organization, modeling and evaluation, and programming; graduate students will also learn parallel software development skills valuable to a wide variety of industries. Undergraduates completing their senior Design Project or Honors Design project will be given the opportunity to contribute toward the above projects as well. Additional details are available in the discussion of Contributions to the Training of Highly Qualified Personnel.

The research results of the proposed program will also be integrated with senior-level undergraduate courses at McGill University in Computer Organization and Architecture and Embedded Systems. A graduate-level course specifically focusing on stochastic design processes and the development of resilient systems will also be developed based on the results of the proposed research.