Faculty Portfolio: Promotion and Tenure

Date: 6/10/2015
Name: Phillip H. Jones III
Department: Electrical and Computer Engineering
Current Rank: Assistant Professor

I. Self Assessment of Scholarship

A. Summary Statement of Accomplishments in and Impact of Scholarship

Computing technology is a major driving force in society, impacting everything from how we interact with one another to the manner in which medical discoveries are made. Looking back only 20 years, the rapid advancements in computing has taken the idea of everyone (even elementary school children) having the ability in the palm of their hand to access inconceivable amounts of information from the realm of science fiction to an everyday reality. However, supporting the growing desires and needs of society for the next 20 years will required continued innovations in computing.

As an Assistant Professor within Iowa State University's Electrical and Computer Engineering department, my research has focused on the following computing areas to contribute to advancing the current state of the art:

Hardware support for embedded Cyber Physical Systems (CPS): CPS can be viewed as systems that have a tight coupling between their cyber (i.e. computing) and physical components. Often these systems have real-time and controls aspects. My research in this area has targeted the investigation, development and evaluation of computing hardware mechanisms to support these real-time and control aspects [3, 4, 5, 9, 14, 18, 19, 23, 29].

Reconfigurable computing: In this area, I have used reconfigurable computing technology as a research medium to explore the idea of using temperature feedback to improve computing performance [24, 28, 30, 31, 32, 33, 34], and leveraging the reconfigurable nature of this medium to perform fine-grained performance profiling [7, 35, 37, 42], and improving design resilience to faults within its computing fabric [2, 27].

Application specific hardware architectures: I have an interest in investigating hardware architectures for accelerating applications. Beyond the success I have found with accelerating sensor processing, controls, and real-time scheduling computations, I have found some initial success in the areas of matrix operation acceleration [6, 13, 16], graph processing[10, 17], security[1, 20, 25], and bioinformatics [21].

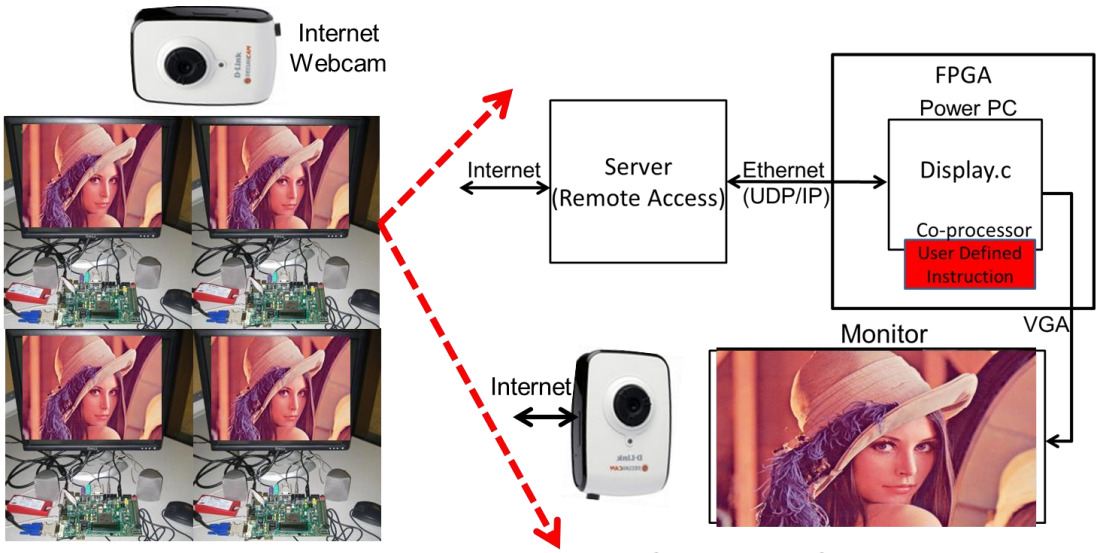
Summary of impact: In addition to the impact the above publications have had on the academic community through dissemination in journal and conference venues, the remainder of my portfolio highlights: 1) specific contributions of selected works, 2) accomplishments that have brought recognition to my department on an international stage, 3) innovations in the curriculum, and 4) contributions to broadening participation in the field of electrical and computer engineering.

B. Specific Statements of Accomplishments and Impact

1. Self Assessment of Accomplishments in and Impact of Teaching and Learning

I have enjoyed the opportunity to innovate within the curriculum. A number of these innovations have resulted in publications [12, 15, 21, 22, 26]. I have focused on five courses: 1) Reconfigurable Computing (CPRE 583), 2) Models and Techniques in Embedded Systems (CPRE 584), 3) Embedded Systems (CPRE 488), 4) Introduction to Embedded System (CPRE 288), and 5) Senior Design (CPRE 491/492).

Reconfigurable Computing (CPRE 583): The goal of this course is to help train new graduate students in conducting research in the area of computer architecture and reconfigurable computing. It is also offered as a distance education course to help service the needs of regional companies (e.g. Rockwell Collins, Lockheed Martin, and John Deere) in advancing their employees' skill set. Before my arrival at Iowa State University, this was already a successful course, however it was primarily lecture based. I developed a lab infrastructure that enabled both on-campus and distance education students to gain equivalent hands on experiences.



CPRE 583: Student Remote Development Infrastructure

CPRE 583: Co-processor Image Processing Assignment

Figure 1: a) Student Remote Development Infrastructure, b) Image Processing Assignment

Remotely accessible servers (as shown in Figure 1a) allowed distance students access to the same industrial strength tools as their on campus counter parts, local networking allowed distance students to interact with Reconfigurable Computing platforms, and Web cameras allowed them to remotely observe simple debugging mechanisms such as LEDs and small LCD screens.

Hands-on assignments were carefully designed to give distance students an equivalent experience as on-campus students. For example, Web cameras directed at computer monitors allowed students to observe the behavior of their image processing hardware, and network processing assignments allowed students to analysis and debug their hardware design by monitoring the local network traffic using tools such as TCPdump (Figure 1b).

Two unexpected outcomes of my efforts in developing hands-on components for this course were: 1) the on-campus students who had physical access to the lab setup began to abandon the physical lab in favor of using the distance student infrastructure. This could be considered the ultimate evidence that the distance infrastructure was fulfilling its intent to give on-campus and distance students an equivalent experience, 2) during my first semester at Iowa State, a major challenge was finding a way to allow distance students to remotely use the design tools, which are intensely graphical, over their low bandwidth internet connections. The solution that I came across was a remote access tool called NX. NX was so successful with my small class of 20 students that the department's computer support team has adopted it as the primary tool for having students remotely access computing resources, impacting on the order of 1,000 students.

I have an *ASEE* [22] and *MSE* [26] publication based on undergraduate courses that have leveraged the infrastructure I developed for CPRE 583.

Models and Techniques in Embedded Systems (CPRE 584): CPRE 584 was developed with the goal of training new graduate students for conducting research in the area of embedded computing systems and software/hardware codesign.

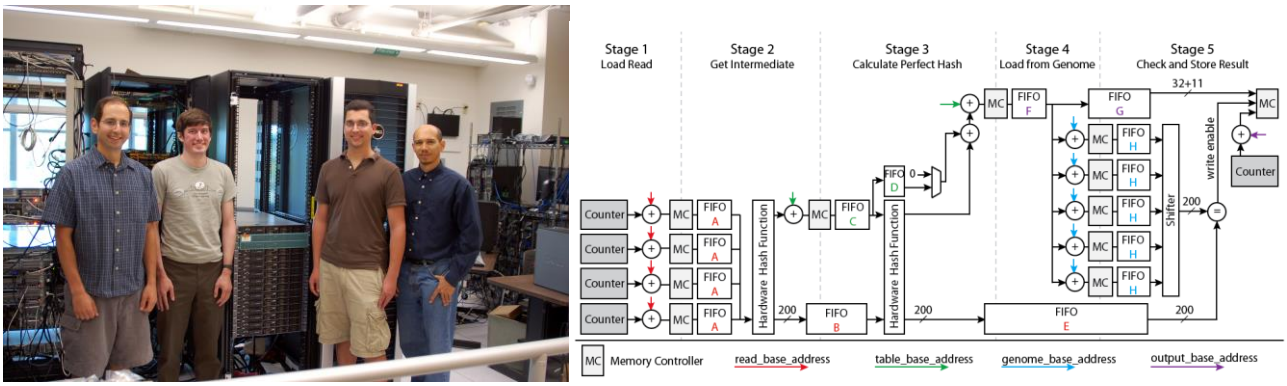


Figure 2: a) CPRE 584:MemoCODE team members (I am on the far right), b) winning hardware accelerated DNA mapping design [21]

Diving into research. A key idea behind this class was recreating the experience a new graduate student encounters in the field of embedded systems research. As part of the first third of the semester, students are exposed to a number of fairly complex design tools and are expected to ramp themselves up quickly. Since most of these tools have hundreds of pages of documentation, students are forced to learn to extract pertinent information efficiently. When students hit walls, I step in to help them navigate roadblocks. In order to show tool proficiency, they must demonstrate the implementation of a baseline experimentation infrastructure that can be used to evaluate research ideas. A common objective has been implementing an infrastructure that can be used to design, implement, and evaluate image processing algorithms (e.g. Sobel edge detection).

Teamwork and generating excitement. Synchronizing this class with an international design competition has been a successful innovation. The ACM-IEEE International Conference on Formal Methods and Models for Codesign (MemoCODE) design competition has been an excellent vehicle for testing the skills students develop during the first third of the course. On March 1st, MemoCODE releases a hardware/software codesign challenge, and teams have until March 31st to complete a fully functional solution. Winners are chosen in two categories: 1) Fastest, and 2) Most efficient use of computing resources. Since 2009 my class has had a few 2nd place finishes and in 2012 won 1st in

the category of Fastest (20X faster than 2nd place), and took second in computing efficiency. Members of the 2012 team are shown in Figure 2a, and the winning design's high-level architecture is shown in Figure 2b. Further detail can be found in the MemoCode conference proceedings [21]. As of 2014, this competition has moved to the summer, but has still been a great motivator for students to rally behind. In 2014 I had another team finish 1st. The details of the competition problem and our solution can be found in [15].

Embedded Systems (CPRE 488): This Junior/Senior level course was re-envisioned to provide students with design experiences they would find in modern embedded system development. The ZedBoard, which hosts a Xilinx Zynq FPGA, was used as the core of all laboratory assignments. The Zynq System on Chip (Soc) architecture embeds an ARM processor subsystem into an FPGA reconfigurable fabric, allowing students to gain experience with both embedded software and hardware design. Laboratories focused on topics such as integrating advanced third party Intellectual Property (IP) cores, developing their own hardware IP cores, Linux kernel and driver development, image processing, communication protocol analysis and implementation, and control system implementation. The heaviest lifting for revamping this course was the design and implementation of the laboratory assignments, shown in Figure 3. The work put forth in the development of this course also resulted in an MSE publication [12].

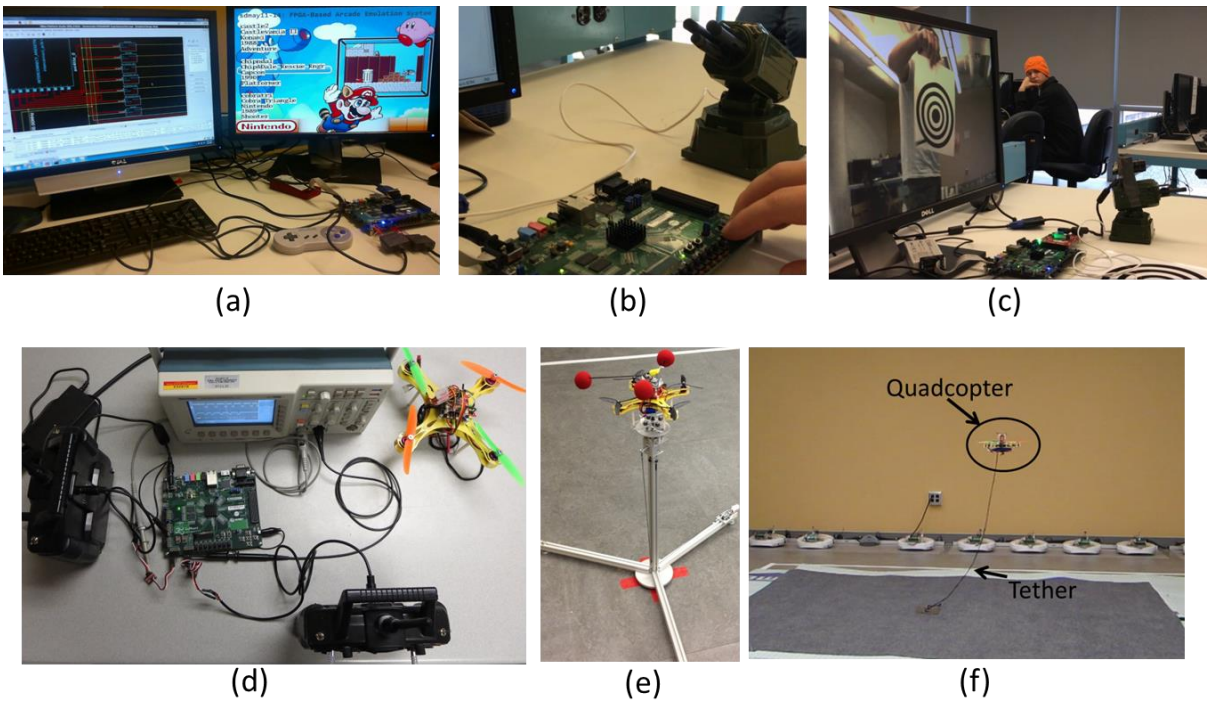


Figure 3: System on Chip (SoC) embedded system course (CPRE 488) based on the Xilinx Zynq FPGA. Lab assignments developed: a) Nintendo Entertainment System (NES) emulator, b) Linux driver development for a Nerf cannon, c) High definition image processing for object detection, d) Radio Control (RC) trainer port protocol capture and generation, and e/f) Introduction to controls.

Introduction to Embedded Systems (CPRE 288): This is primarily a sophomore-level undergraduate course. Through this course I found I greatly enjoy teaching undergrads at the early stages of their education. Currently I am helping perform a major updating of the lab infrastructure. This involves moving to a new I-Robot platform, and transitioning from an Atmel to an ARM based embedded processor architecture. I received the Warren B. Boast Undergraduate Teaching Award from my department in 2014 for this course.

2. Self Assessment of Accomplishments in and Impact of Research and Creative Activities

By their nature, cyber-physical systems (CPS) require examining the interaction of computing and physical components of a system in a holistic and cross disciplinary manner. Often these systems have real-time and controls aspects. The grand vision of my research is to drive the state-of-the-art in systems architecture and analysis techniques for engineer applications using such a cross disciplinary and holistic systems view. In particular, I have conducted investigations into cyber-physical interactions that occur at the intersection of hardware architecture, real-time systems, and controls design. Figure 4 captures my vision of the type of resource-constrained CPS I would like to help enable and an example of high-level architecture modifications that move toward this goal.

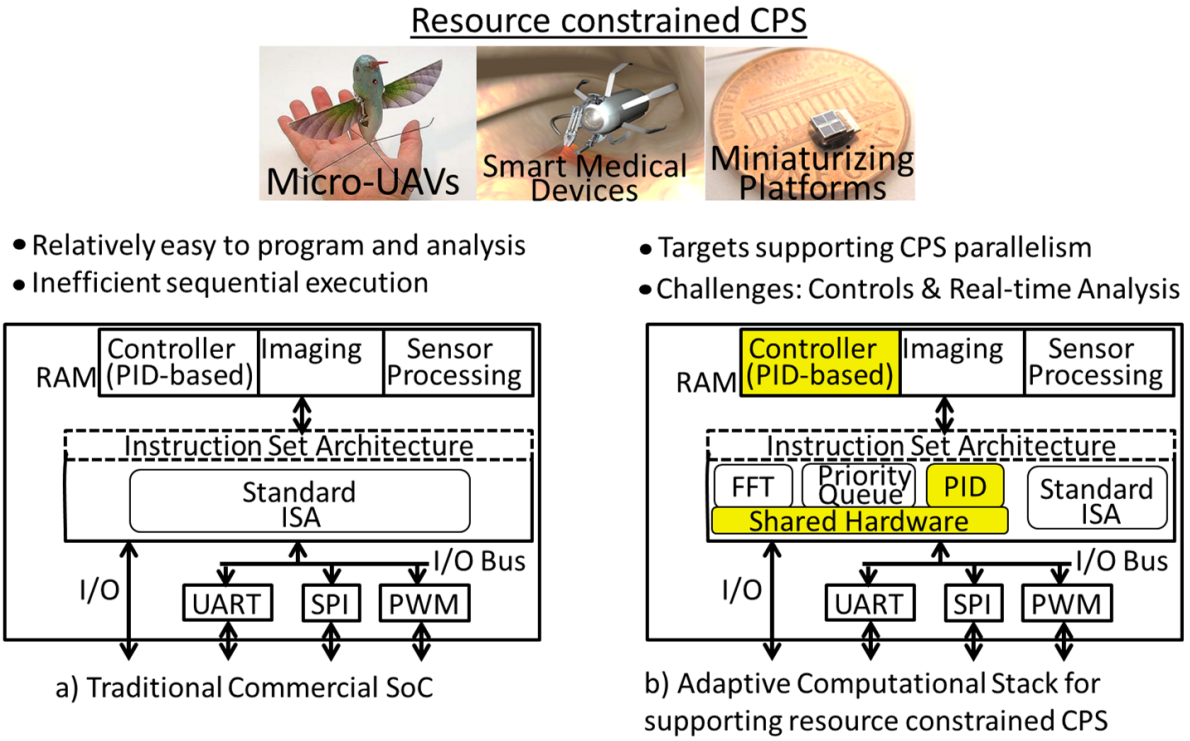


Figure 4: Vision for hardware architectural support of resource constrained CPS (Cyber Physical Systems)

In summary, the following presents three projects that reflect my efforts thus far in this endeavor. ***Project 1*** examines the use of hardware to assist in the implementation and design of **controls** aspects of embedded systems. ***Project 2*** explores the use of hardware to assist in reducing **real-time** scheduling overheads and improving execution time determinism. ***Project 3*** focuses on the specific **physical property of temperature** with respect to reliable measurement in reconfigurable fabrics to support thermal feedback for application performance improvement. These projects integrate into the computational stack hardware features that support controls and real-time aspects of CPS to better serve the dynamics of CPS applications.

Project 1: Controls Systems Hardware Support

Embedded systems and digital control theory have independently developed into mature fields, despite the clear connection between controllers and the embedded platforms that host them. The first digital controllers were implemented on dedicated microprocessor, thus helping maintain certain traditional simplifying assumptions for controls design (e.g. constant sensor sample rates, and constant computing time). However, increasing demands for tighter system integration, smaller

form-factors, and lower-power design has played a large role in making these simplifying assumptions less valid. In modern embedded systems, it is common for control tasks and non-control tasks to share a single processor via a scheduling algorithm or Operating System (OS), which in turn contributes to variations in task execution time. The **primary goals** of this project are: 1) characterizing the degree to which traditional controller design simplifying assumptions hold true and evaluating hardware mechanisms to help maintain these simplifying assumptions [5, 29], 2) developing a design approach using a Plant-on-Chip (PoC) hardware architecture to evaluate controllers on a given embedded platform earlier in the design process [4], and 3) investigating the hardware architecture space for supporting advanced control algorithms [9].

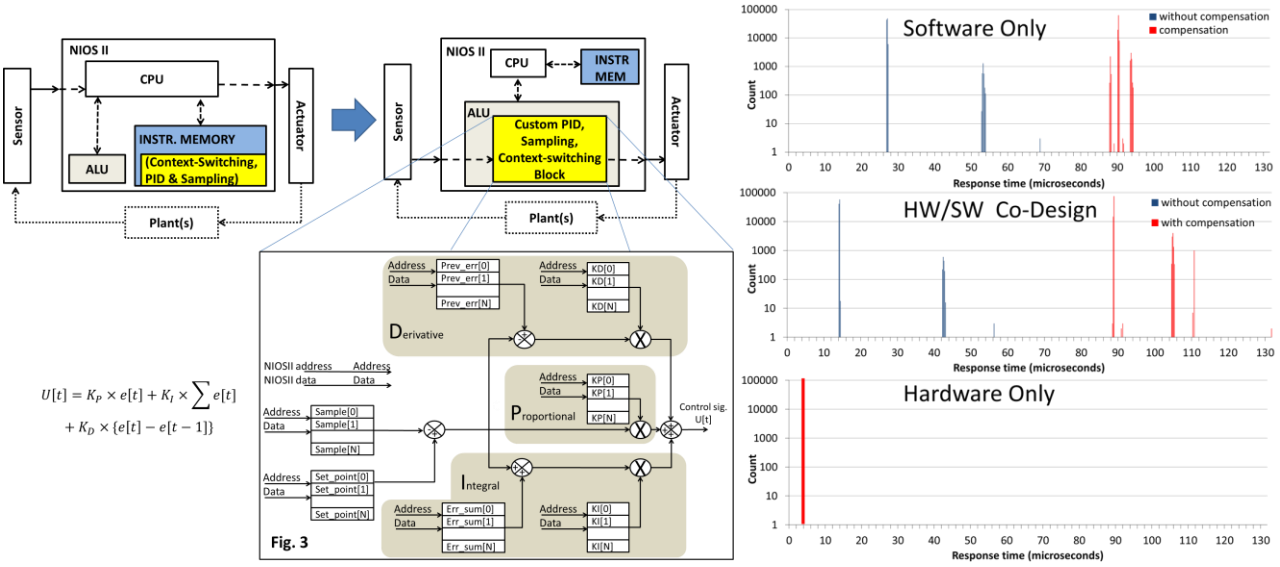


Figure 5: a) FPGA-based evaluation setup for moving functionality between software and hardware, and b) histograms representing the variation in response-time for executing a PID controller for several SW/HW distributions of functionality.

Characterizing traditional simplifying assumptions for controller design. In [5] and [29], hardware mechanisms were designed and evaluated on an FPGA-based system-on-chip platform called RAVI (see Figure 6a) to support sensor processing and proportional-integration-derivative (PID) controller computations. Figure 5a illustrates the extreme configuration used while evaluating the impact of migrating PID controller functionality between a complete software implementation (far left of Figure 5a) and a complete hardware implementation (far right of Figure 5a). Figure 5b provides a flavor of the type of data collected from performing controller response-time performance evaluations.

Maintaining consistency is a key factor when comparing software/hardware tradeoffs for controller implementation. The RAVI development board allowed the use of a single platform for developing and evaluating each of the four software/hardware controller variations evaluated. The FPGA was used to implement the NIOS-II (Altera's soft-processor) based system-on-chip setup shown in the upper part of Figure 5a.

The metrics of interest for our evaluation were 1) response time, and 2) response-time jitter. The system variables varied to evaluate these metrics were:

(1) the architecture (I:Full SW, II: PID core in HW, III: PID core + sensor sampling in HW, IV: PID core + sensor sampling + plant context switching in HW),
(2) the number of plants controlled (10,100, 1000),
(3) the processor interrupt timer (1 ms, 100 ms),
(4) the sensor sampling rate in samples per second (SPS) (No Delay, 200KSPS, 819SPS),
(5) software-implemented jitter compensation (used or not used).

Result Summary. While keeping a fixed sampling rate and a fixed number of plants, and moving from Case I (completely software) to Case IV (completely hardware) we observed improved response times. Most interesting was the large improvement in response-time performance when moving plant context switching functionality from software to hardware. Hardware performing context switching more efficiently than software makes sense as it locally stores context in BlockRAM (i.e. on-chip memory), which can be accessed in a single clock cycle. In comparison, the main processor potentially has to fetch context information from main memory (taking many and a non-deterministic number clock cycles).

Contribution Summary. The primary contributions of this part of the project ([5] and [29]) were 1) the tight integration of a time multiplexed hardware PID controller within an embedded processor, 2) the characterization of the PID controller architecture for several alternative hardware/software hybrid-designs with respect to response time and jitter, and 3) the tight integration of a hardware-based sensor processing unit (SPU) within an embedded processor and its evaluation with respect to the software implementation of common sensor processing tasks in terms of response time.

Plant-on-Chip (PoC) approach for controller system- integration [4]. As the assumption of controller deterministic execution time on shared embedded platforms has continued to become less valid, interest has sparked in both the field of controls theory and the field of real-time scheduling to address this concern. Experts in real-time scheduling have explored new scheduling techniques to help maintain the assumption of periodic sample rates (i.e. reducing response-time jitter), and the controls community has developed advanced control algorithms and analysis techniques to better tolerate response time-jitter. Ideally the goal is to maximize CPU utilization, while maintaining a physical system's stability needs.

Additionally, a number of software tools (e.g. Jitter-Bug, TrueTime) have been developed to help analysis the impact of non-deterministic execution time on system stability. Though these analytical and simulation-based design tools provide a good approximation of a system's robustness to sample-period and delay, they work in environments and under assumptions that delay can be modeled as a probability distribution function. Research shows this to be not realistic and that computer elements (cache misses, bus arbitration) cause non-deterministic time variation in delay and sample-period.

Overview. In this work [4] an approach for controller system integration is presented that directly interfaces a Plant-on-Chip (PoC) emulator with an actual processor under test, the setup inherently incorporates the target system's sources of non-determinism, thus giving a more accurate result when characterizing the system's robustness against sample period and delay variation. Figure 6b illustrates the high-level system organization of the PoC directly interfaced to the embedded system on a single chip. Figure 6c provides an architectural view of the PoC, which implements a generic four state-space calculator, and Figure 6d provides an example of post-processed data used to relate system stability to various average sample times, and controller response times.

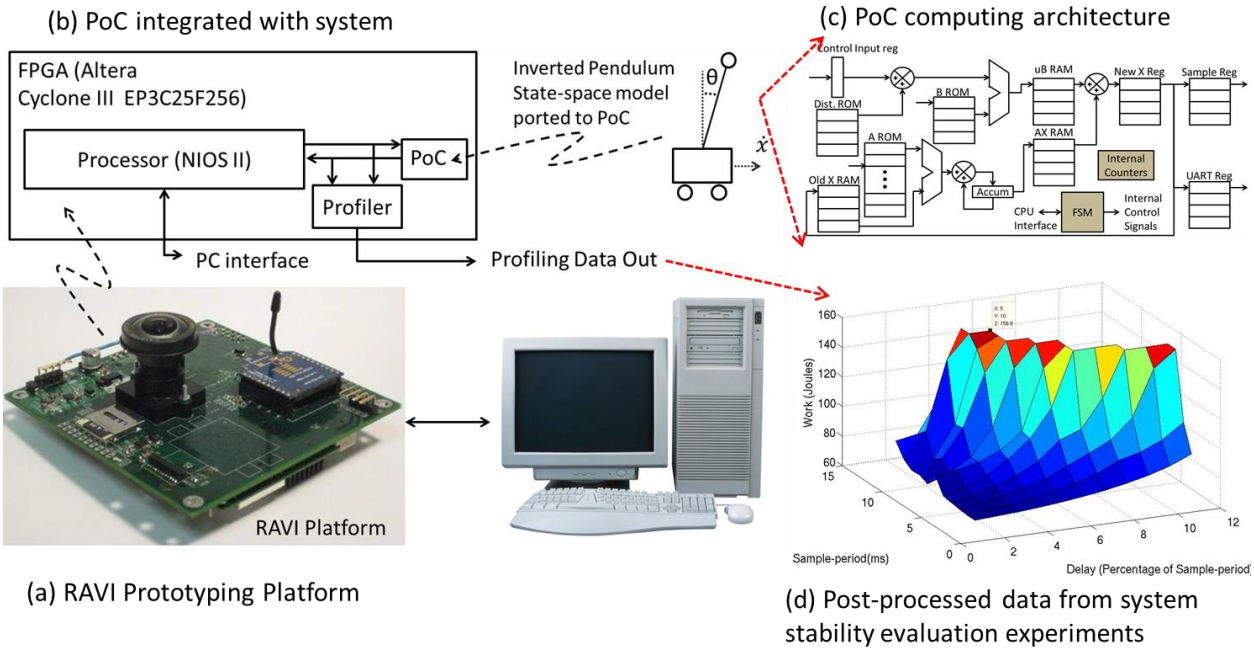


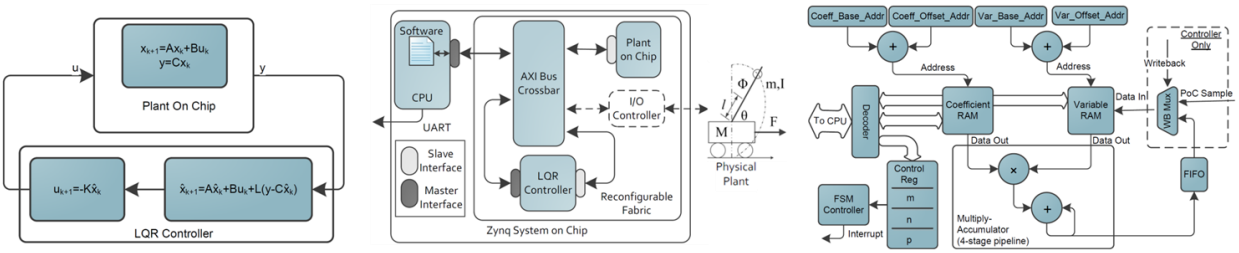
Figure 6: (a) RAVI FPGA-based prototyping platform, (b) PoC substituted for an inverted pendulum plant for evaluating system stability for variation in controller response time, (c) hardware computing architecture of the PoC, (d) example post-processed data showing actuator effort vs. sensors sample period and controller computation delay.

Contribution Summary. The experimental results indicate that this proposed framework both safely and accurately captures the non-deterministic effects of modern processor architecture on a physical plant. Comparing our results with those obtained by Jitter Bug (a popular tool for evaluating system stability); we demonstrate a more accurate representation of a real system.

Advanced control algorithm integration [9]. Implementing control algorithms on an FPGA can be challenging for engineers unfamiliar with hardware architecture design. This work presents a software-configurable FPGA co-processor architecture that can implement a wide range of linear state-space controllers, up to the complexity of a Linear Quadratic Regulator (LQR) coupled with a Luenberger observer. **Two goals** of this work are to 1) act as a stepping stone for separating embedded software concerns (e.g. real-time scheduling feasibility) from controls concerns (e.g. accounting for update-rate jitter), and 2) help envision a design methodology for bridging the communication gap between controls and embedded software engineering.

Overview. As can be seen in Figure 7a, the computations used to implement a state space PoC are similar to the computations required for state space controllers, thus the PoC was used as a starting point for this work. The state-space controller co-processor and PoC were extended to 1) support an arbitrary number of states, sensor inputs, and actuator outputs, and 2) use floating point instead of fixed point operations. Figure 7b illustrates the mapping of this architecture to the Xilinx Zynq FPGA computing medium, and Figure 7c provides a logical view of the state-space controller architecture. Figure 7d gives an example of how the PoC + state-space controller architecture can be used to analysis trade-offs between implementing a controller for a given plant in software vs. hardware.

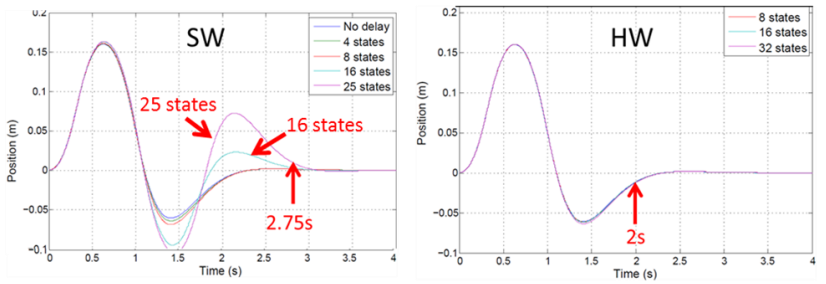
Contribution Summary. Embedded systems engineers can easily reconfigure this controller to suit a wide range of controls applications that can be represented as a state-space linear model.



(a) High level PoC + controller architecture.

(b) Architecture mapped to Zedboard's Zynq FPGA.

(c) LQR state space controller with observer block diagram.



(d) Inverted pendulum on cart stability after a disturbance. Software vs. Hardware comparison for increasing number of pseudo controller states.

Figure 7

Project 2: Real-time Systems Hardware Support

Deploying increasing amounts of computation into smaller form factor devices is required to keep pace with the ever increasing needs of real-time and embedded system applications. The area of micro Unmanned Ariel Vehicles (UAVs) is an example of where such need exists. The size of these vehicles has rapidly decreased, while the capabilities users wish to deploy continue to explode. In June of 2011, the New York Times published several articles on the amazing work being pursued by Wright Patterson Air Force Base to develop micro-drones to aid soldiers on the battlefield. In February of 2011, the DARPA funded Nano Air Vehicle (NAV) program demonstrated a humming bird form-factor UAV weighing less than 20 grams (e.g. less than an AA battery) with video streaming capabilities. These real-time embedded applications can no longer rely on manufacturing advances to provide computing performance at Moore's law rates, due transistors approaching atomic scales and thermal constraints. Thus, more efficient use of the transistors available is needed.

In this project [3, 14, 19, 23], we assert the boundaries of software and hardware must be reexamined and believe a fruitful realm for research is the hardware-software co-design of real-time system functionality that has been traditionally implemented in software. Such a co-design is needed to balance the cost of dedicating limited silicon resources for high-performance fixed hardware functionally, with the flexibility and scalability offered by software. Additionally, we claim seamless migration between software and hardware implemented functionality is required to allow systems to adapt to the dynamic needs of applications. Specifically, this project examines hybrid architecture for priority queue management and evaluates this architecture within a real-time scheduling context.

The following motivates the importance of low processing overhead and timing predictably to a real-time scheduler's performance. Real-time operating systems (RTOSs) are designed to execute tasks within given timing constraints. An important characteristic of an RTOS is predictable response under all conditions. The core of the RTOS is the scheduler, which ensures tasks are completed by

their deadline. The choice of a scheduling algorithm is crucial for a real-time application. Online scheduling algorithms incur overhead, as the task queues must be updated regularly. This action is typically paced using a timer that generates periodic interrupts. The scheduler overhead generally increases with the number of tasks. A high resolution timer is required to distribute CPU load accurately based on a scheduling discipline in real-time systems, but such fine-grain time management increases the operating system overhead.

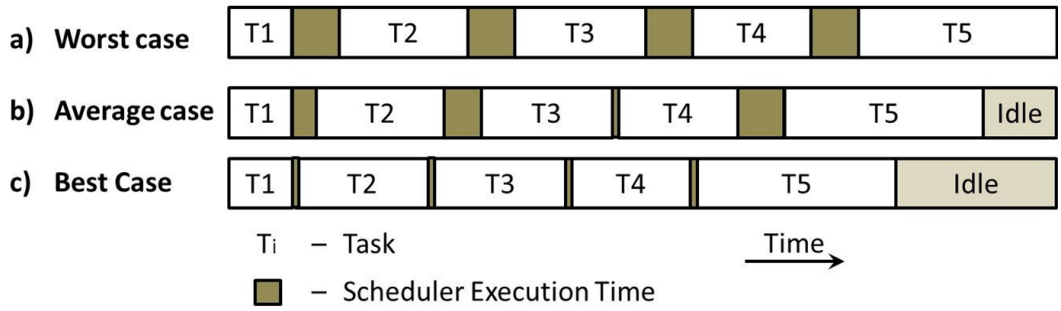


Figure 8: In order to allow analytical analysis of schedule feasibility, worst-case execution time (WCET) typically needs to be assumed. Thus, scheduler execution time variations that cause large differences between WCET and typical case execution time reduce utilization of system computing resources.

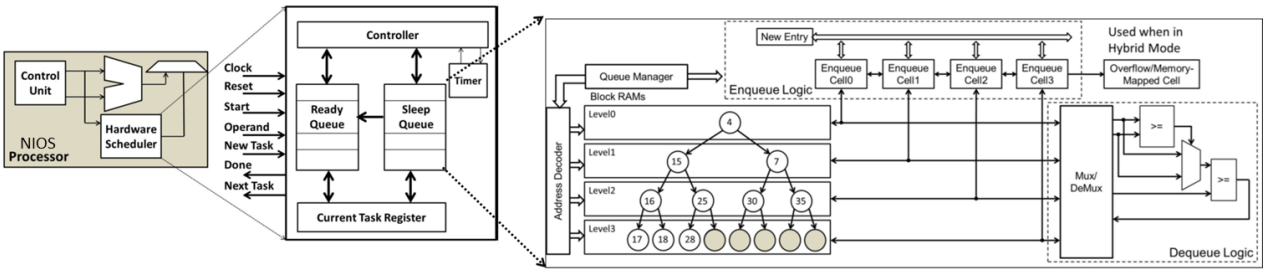
The extent to which a scheduler can ideally implement a given scheduling paradigm (e.g. Earliest Deadline First (EDF), Rate Monotonic (RM)), and thus provide the guarantees associated with that paradigm, is in part dependent on its timing determinism. A metric for helping quantify the amount of non-determinism that is introduced to the system by the scheduler is the variation in execution time among individual scheduler invocations. This can be roughly summarized by noting its best-case and worst-case execution times. Variations in scheduler execution time can be caused by system factors such as changes in task set composition, cache misses, etc. Reducing the scheduler's timing sensitivity to such factors can help increase deterministic behavior, which in turn allows the scheduler to better model a given scheduling paradigm.

Figure 8 illustrates how the variation in scheduler overhead affects processor utilization. To ensure that tasks meet their deadlines, the scheduler's worst-case execution times are often overestimated. This can cause a system to be underutilized and wastes CPU resources. In this project, we examine how the scheduler overhead and its variation can be reduced by migrating scheduling functionality (along with time-tick interrupt processing) to hardware logic. The expected results of our efforts are increased CPU utilization, better system predictability, and finer schedule and timing resolution.

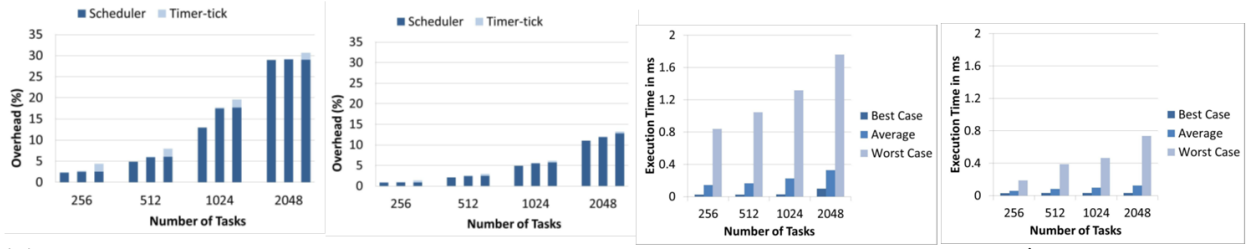
Overview. Figure 9a provides an overview of our software/hardware priority queue co-design implementation that supports up to 255 elements while under hardware management, and transitions to software-hybrid mode when this limit is surpassed, placing additional elements into main memory. A memory mapped interface provides software with access to priority-queue-structured on-chip memory, which enables quick and low overhead transitions between hardware and software management. As an application of this hybrid architecture, we implemented and evaluated a scalable task scheduler for real-time systems that reduces scheduler processing overhead and improves timing determinism of the scheduler.

Contribution Summary. The primary contributions made in this area have been 1) a hardware accelerated binary min heap that supports enqueue and peek operations in O(1) time, returns the top-priority element in O(1) time, and completes a dequeue operation in O(log(n)) time, 2) a scalable

hardware-software priority queue architecture that enables fast and low-overhead transitions of queue management between hardware and hybrid modes of operation, and 3) A hybrid scheduler architecture that reduces scheduling overhead and improves predictability [3, 23].



(a) SW/Hardware co-design of a Priority Queue implemented as a custom instruction



(b) Performance of a software implemented EDF scheduler compared to one based on SW/HW co-designed priority queue.

Figure 9

Results Summary. A new hybrid priority queue architecture has been implemented, which can be managed in hardware and/or software. The priority queue when managed in hardware supports enqueue and peek operations in O(1) time, returns the top-priority element in O(1) time, and completes a dequeue operation in O(log(n)) time. The design enables quick and low overhead transition between hardware and software management. It utilizes hardware logic to enhance the performance of queue operations even when managing the priority queue in software. As an application of the proposed priority queue architecture, a scalable hybrid scheduler is implemented that supports 255 tasks in hardware mode and up to an arbitrarily large number of tasks in hybrid mode. Figure 9b provides a brief summary of the results of the performance evaluation of our Field Programmable Gate Array (FPGA)-based system-on-chip prototype. It shows a 90% reduction in scheduling overhead and a 98% decrease in scheduler execution time variation, when the scheduler is managed by hardware as compared to software, thus giving more predictable execution times, which is necessary in high-performance real time systems.

Future Research. Avenues of future work include, 1) reducing the rate of performance degradation as the queue overflows into software, 2) evaluating the use of the hybrid priority queue in discrete event simulation and network optimization algorithms and 3) integrating the hybrid scheduler with Real-time Linux and characterizing the scheduler performance.

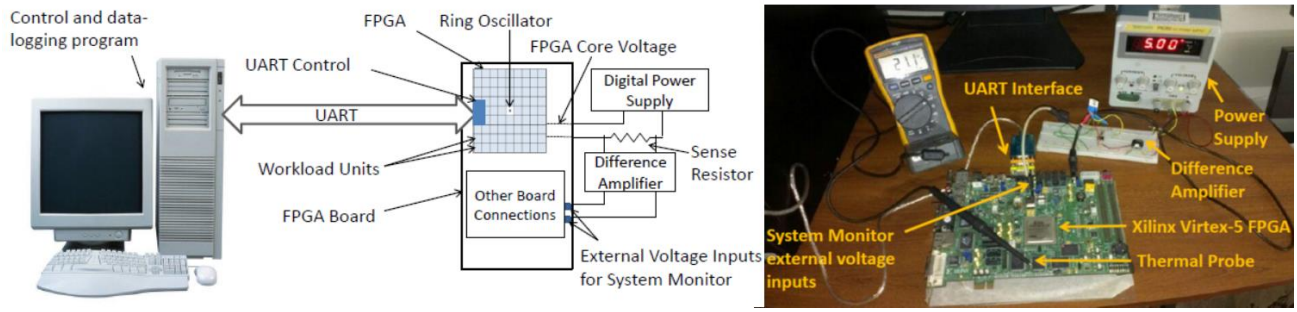


Figure 10: Experimentation setup for monitoring FPGA current draw to track workload intensity

Project 3: Temperature Feedback in Reconfigurable Computing

Embedded systems can be found in harsh and dynamic thermal environments. Environments such as deserts and outer space inherently exhibit large temperature changes over time. In general, embedded computers may need to operate in diverse thermal environments due the mobility of their host, or different instances of a system being deployed in significantly different thermal environments. Even large high-performance systems located in environmentally controlled rooms can be exposed to dynamic thermal conditions. For instance, a blade based computing system is one example, where densely packed computing nodes can cause constrained airflow and the thermal output of a node can impact its neighbors.

Systems with highly variable workload intensities can be thermally constrained by worst-case workload conditions. This is especially true of application specific integrated circuits (ASICs) that implement highly parallel and computationally intense applications. These applications can have many parallel modules active simultaneously under worst case workloads, while under light workloads most modules may be idle. Workload variation is an even larger concern for platforms that use reconfigurable devices. The diverse range of functionality that can be mapped onto these platforms, overtime, creates a wide variation in the amount heat that can be generated by the system.

Project Goals. This project investigates the use of temperature feedback for improving FPGA-based system performance. Specifically **two** aspects that have been explored are 1) obtaining reliable on-chip temperature measurements [24], and 2) quantifying the degree to which thermal feedback can be used to improved system performance [28, 30, 31].

Reliable Temperature Measurement.
Problem Investigated. During my previous work with using temperature feedback for improving the performance of FPGA-based systems [30, 31], it was observed that for variations in workload intensity that the output vs. temperature relationship of an on-chip ring oscillator based thermometer varies (Figure 11a). For applications that have extremely dynamic workload this can make using ring-oscillators based thermometers a challenge. Referring back to Figure 11a, varying FPGA utilization from 0% to 40% induces an error in temperate measurement of ~35 degrees. Ideally one wants the workload intensity vs. ring-oscillator frequency to be constant (Figure 11c).

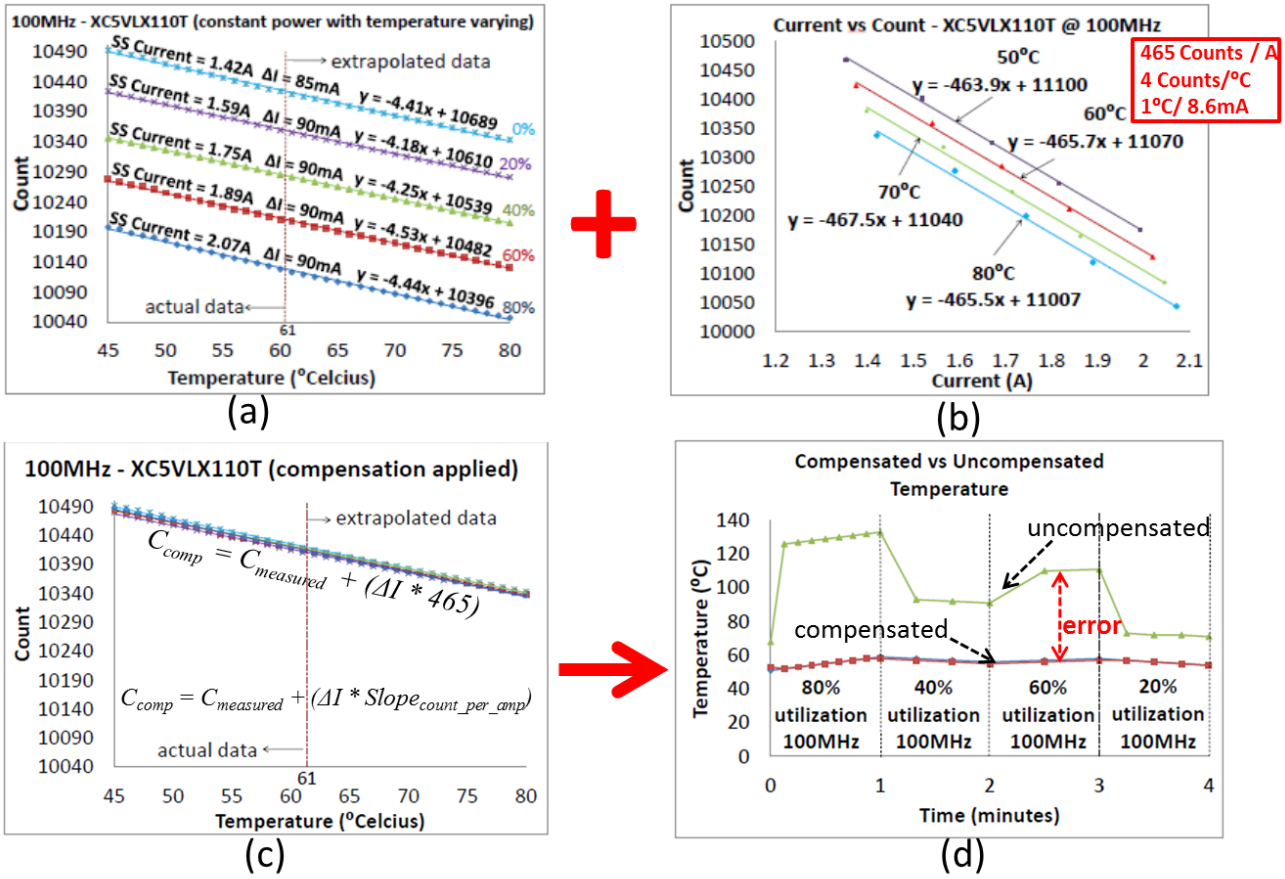


Figure 11: a) Observation that ring oscillator frequency vs. temperature relation varies with the workload intensity of the FPGA (0% - 80% utilization), b) results of experiments to characterize FPGA current draw vs. ring oscillator frequency for lines of constant temperature, c) the application of the results of characterization experiments to (a) to compensate for non-thermal effects of workload intensity on ring oscillator frequency, d) experimental evaluation of compensation technique, as can been seen the error between the compensated ring-oscillator thermometer and actual temperature is not noticeable compared to the uncompensated ring-oscillator thermometer as workload intensity is varied. .

Approach. The approach taken to mitigate this problem was to first characterize the non-thermal impact that workload intensity variation had on ring-oscillator frequency (typically caused by workload variation inducing microvolt level variation in the FPGA power source). Figure 10, shows the experimentation setup to perform this characterization, and Figure 11b provides the results of this characterization, a linear relation can be observed. This relation was then applied to the measured ring oscillator frequency to compensate for the non-thermal effects of workload intensity variation (Figure 11c). Figure 11d shows the resulting increase in measurement accuracy under workload variation.

Results Summary. For the specific FPGA characterized in Figure 11b, it was found that every 8.6mA change in workload current draw resulted in ~1 C error in temperature measurement (using a base workload as a reference). It should be noted that this linear relationship held when applied to a much larger FPGA hosted on a completely board, just with a different constant for slope. As can be seen in Figure 11d, before applying compensation, workload variation caused up to a 74 degree error in temperature measurement. The proposed compensation reduced the maximum error to 2 degrees [24].

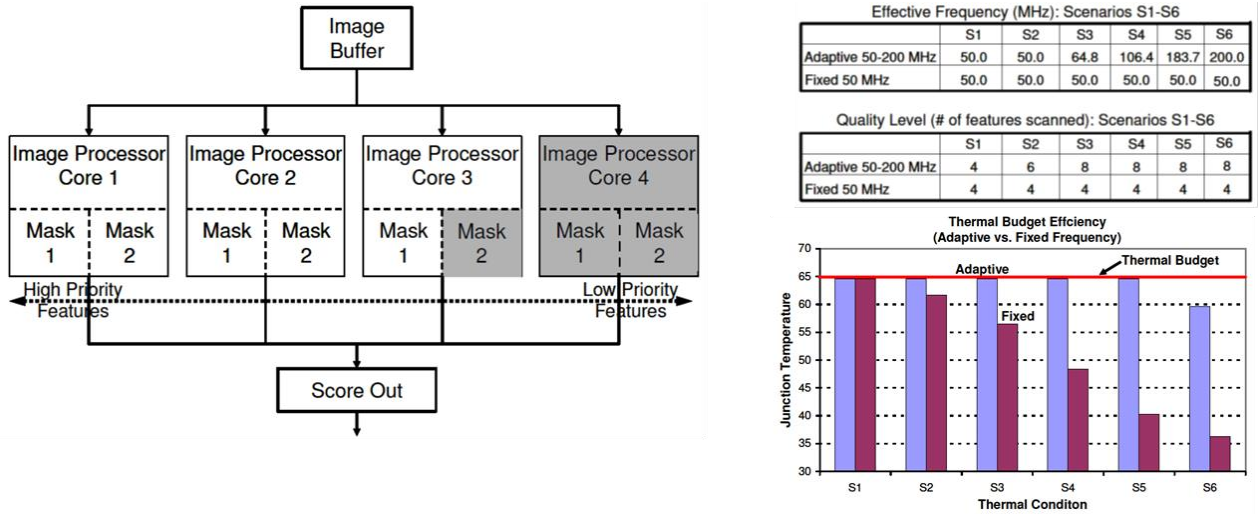


Figure 12: a) Run-time parameterizable FPGA-based image processing application, where application clock frequency, and number of Masks applied to an image (i.e. features scanned for) adjust to maximize performance for a given "Thermal Budget", b) Summary of experimental results that quantify performance improvement when using temperature feedback to dynamically adjust application parameters, as compared to a static design that must meet thermal constraints under worst-case operating conditions. Scenario S1 represents worst-case operation conditions (e.g. broken fan, high ambient temperature), while S6 represents best-case operating conditions (e.g. operational fans, and low ambient temperature) [30, 31].

Improving System Performance.

Problems Investigated. Two brute force approaches typically used to address thermal issues are: 1) overprovisioning through the use of heatsinks and fans to account for worst-case thermal operating conditions, and 2) constraining an application performance so that thermal thresholds will not be surpassed while operating under such conditions by limiting operating frequency, or the amount of parallelism (area) the application is allowed to utilize. Over provisioning a system can be expensive in terms of cost, performance, power, and weight (a major concern in micro UAVs). Constraining application performance to meet worst case thermal conditions can cause a system to perform significantly below its realizable potential during typical operating conditions.

Here I give a brief overview of two approaches that I have investigated to make use of temperature feedback to increase FPGA-based application performance for a given thermal constraint (i.e. "Thermal Budget"), as compared to a static (or fixed) implementation design to meet temperature constraints under worst-case thermal conditions.

Adapting Application parameters using temperature feedback [30, 31]. Figure 12a depicts an image processing application that can scan images for up to 8 features (one Mask per feature). The number of features scanned in parallel is run-time parameterizable based on temperature feedback. Additionally, the frequency at which the design runs is a function of temperature. Figure 12b summarizes the performance evaluation result when comparing this adaptive architecture to a static version that meets temperature constraints under worst thermal operating conditions. As can be seen, the adaptive architecture manages excess temperature margins to significantly increase application performance while operating in better than worst-case thermal operating conditions. Under best-case thermal operating conditions, we see a 4x factor increase in image throughput (operating at 50 MHz vs. 200 MHz), and a 2x improvement in the number of features that could be scanned in parallel (4 vs. 8).

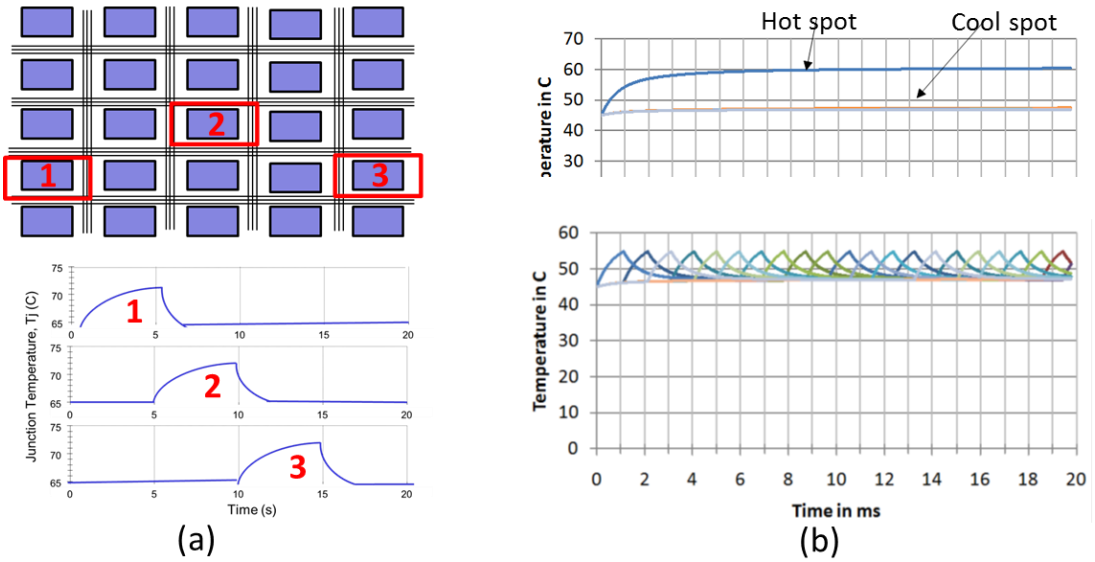


Figure 13: a) Illustration of FPGA junction temperature as a computing module is moved from location 1 to 2 to 3, migration is triggered based on temperature reaching a specified threshold, b) "HotSpot" based simulation results, showing about a 5 degree decrease in maximum hot spot temperature, as compared to a design with a static layout [28].

Mitigating Hot Spots. I have proposed, implemented, and evaluated a method for mitigating hotspots using the dynamic partial self-reconfiguration capabilities available in modern FPGAs. This approach swaps "hot" and "cool" modules at run-time using a temperature threshold to trigger reconfiguration, thereby allowing an FPGA floor plan to adjust to its current thermal conditions (Figure 13a). Using "Hotspot" to run simulation evaluation experiments, this approach shows a reduction in maximum hotspot temperature by up to 8 °C (Figure 13b), and an increase in application throughput by up to a factor of 2-3 times that of a fixed design for a given thermal budget [28].

3. Self Assessment of Accomplishments in and Impact of Institutional Service

I have been given a number of opportunities to contribute to the Electrical and Computer Engineering Department. I have served in a number of departmental administrative positions, volunteered to advise 2-3 Senior Design projects each year, and have been Iowa State University's proctor for IEEE's Xtreme Programming competition.

Departmental Administration.

Curriculum Committee. I served four years on the department's Undergraduate Curriculum Committee. During my tenure on this committee, I participated in discussions with respect to performing a complete redesign of the curriculum, with an end goal of deploying a curriculum that more tightly couples Electrical and Computer engineering content. The vision is to give both EE and CPRE students the same course work for their Freshman and Sophomore year, therefore not having to commit to Computer Engineering or Electrical Engineering until the start of their Junior year, and to give students the flexibility to look as much as an EE or CPRE as they want for their career goals.

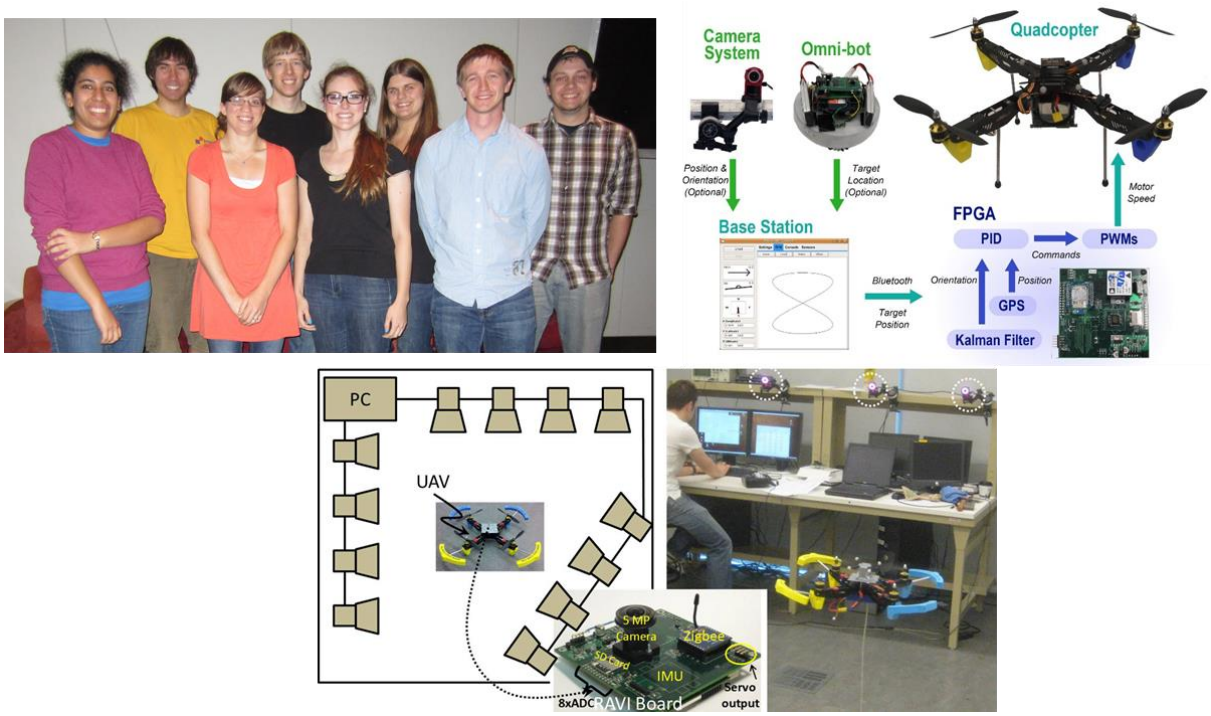


Figure 14: Since 2008 I have advised a Unmanned Aerial Vehicle (UAV) design team. The 2012-2013 team is shown in the upper left, a high-level system view of their project is given in the upper right, and the lower figure shows the UAV in flight. Note, the RAVI board that was developed for CPS research is being utilized for UAV control in combination with a high-speed camera system.

Computer and Networking Systems (CNS) Group Chair. I served as the chair of my research-area group from Spring 2011 to Spring 2014. As chair, my responsibilities included 1) driving group teaching assignments, 2) scheduling and staffing of PhD qualifying exams, and 3) representing my group's interests in the departmental Graduate Committee. Within the Graduate Committee, my responsibilities included giving input on Graduate Curriculum policies and evaluating students for College level recognition for excellence in graduate research, and excellence in teaching.

Other Administrative Service. For the department's most recent ABET accreditation, I gathered evidence of compliance for the undergraduate embedded systems course (CPRE 288) that I teach. I also served on the department's Graduate Admission committee for two semesters.

Senior Design Advising. I believe faculty involvement with undergraduate students outside of class is important not only in helping student development, but also as a means of direct feedback for how well the department's curriculum is meeting its goals. Since arriving at Iowa State University, I have been committed to hands-on mentoring of 2-3 Senior Design Projects each year. I have been most intensely involved with the ever evolving Microcontroller Controlled Aerial Robot Team (MicroCART) project. I have co-advised students with Dr. Nicola Elia (an expert in controls systems) to develop an indoor test infrastructure for developing autonomous flying battery powered vehicles. A photograph of one of my teams can be found in the upper left part of Figure 14. The remainder of Figure 14 shows the high-level system organization of their project and their (Unmanned Aerial Vehicle) UAV in flight.

This project exposes teams composed of both electrical and computer engineers to real world issues that must be considered when applying classroom knowledge to a complex systems. In addition to learning the value of team work, students learn (often the hard way through spectacular UAV

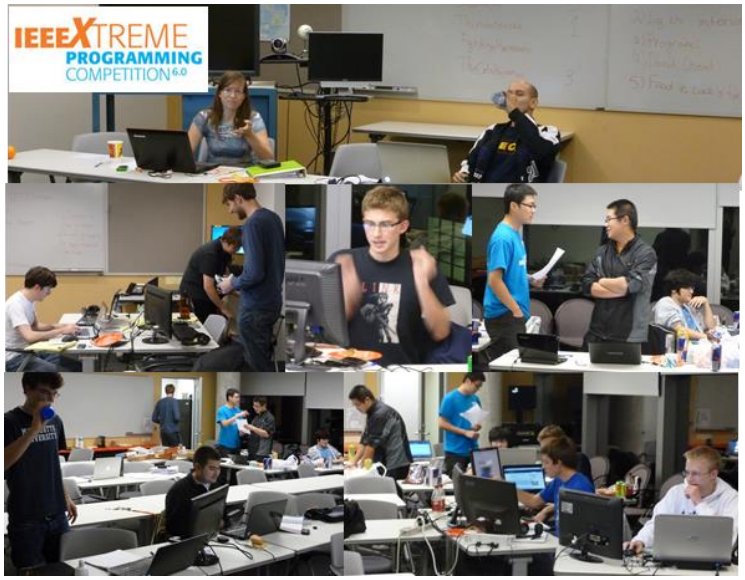


Figure 15: Over the last 4 years, I (upper right) have volunteered to proctor the IEEE 24 hour programing competition (IEEE Xtreme Programming Competition). My primary role is to make sure students follow the IEEE Code of Ethics during the competition, and stay energized and fed through the non-stop 24 hour period. This is a fantastic event for students to learn teamwork and perseverance in the face of challenging problems.

crashes) the value of testing software and hardware, appropriately characterizing noise associated with signals being processed, and systematically approaching and analyzing problems.

As the UAV infrastructure has become refined, it has been leveraged to have senior design teams use the infrastructure to design and implement solutions to problems using autonomous vehicles. For example, teams have had their UAV follow a small robot moving on the ground.

24-hour IEEE Xtreme Programming Competition. As the name indicates, this is a non-stop 24 hour programming competition, sponsored by the IEEE Society. In summary, across the globe students form teams of 4 students or less and over a 24-hour period 1-2 challenging problems are released every hour. The goal is to solve and upload the solution to as many problems as possible. Over the last 4 years, I have served as the Iowa State University proctor. My primary responsibilities have included ensure teams comply with the IEEE code of ethics, stay well nourished, and keep their energy levels up. There are typically over 300 teams worldwide that compete, and the problems given are very challenging. From my proctoring experience, it is not uncommon for teams to go 5 or more hours before completing their first solution. I have found it to be a great opportunity for students to learn the importance of team work and perseverance. I look forward to proctoring this event each year; I use the student's energy to help me have a productive 24-hour Xtreme Research session. Figure 15 gives a snapshot of this event.

4. Self Assessment of Accomplishments in and Impact of Public, Extension and Professional Service

While conducting professional level service has not been my focus since arriving at Iowa State University, I have been able to find small ways to serve.

International Conference on Contemporary Computing (IC3). In the Spring of 2011, I took on a leadership role as co-chair of the Systems track for this conference. It was eye opening how much work goes into identifying and recruiting potential reviewers. I also gained a greater appreciation for how valuable to the community volunteering to participate in the peer review process is.

Academic Paper Reviewer. I have been involved in reviewing papers for major conferences (e.g. IPDPS, ISCAS) and journals (e.g. IEEE Transactions on Computers, TRETS, TECS, TPDS). Typically I review two batches of conference papers a year, and a few journal manual scripts.

National Science Foundation (NSF) panel member. Serving on a number of NSF grant panels has been among the most rewarding of my service experiences. I have enjoyed getting to meet new colleagues. Also gaining an understanding of what goes on behind the scenes during the review processes has made a profound impact on how I approach writing grants.

Future Plans. As I continue to advance in my academic career, actively seeking more opportunities for professional service is something that I am interested. For example, given the success that I have had with mentoring students for the ACM-IEEE sponsored MemoCODE design competition; this would be an activity that I would be interested in taking the lead running for a year or two.

5. Self Assessment of Accomplishments in Impact of Technology Transfer

N/A

6. Self Assessment of Accomplishments in and Impact of Outreach Activities and Efforts to Improve Intellectual, Gender and Ethnic Diversity

Since early in my career at Iowa State University, I have made efforts to support on-campus programs for increasing diversity and participation in STEM fields of study. As my career progresses, I would like to take on larger roles in support of this endeavor. The follow gives a brief description of four programs that I have been involved, and the manner in which I have participated.

Digital Women. The official stated charter of this University group is to: 1) encourage, support, and retain women in Computer Science, Software Engineering, Computer Engineering, Electrical Engineering, MIS, and any other field involved with electrical technology, 2) provide students with meaningful opportunities and social activities that promote positive relationships among students and faculty, and 3) encourage members of Digital Women to become an influence for good on campus and in the community.

Two and a half years ago Digital Women officially became recognized as a University Student group, and they recruited me to be their first official mentor. I came into this role when several female students who had recently taken my undergraduate Embedded Systems course (CPRE 288) indicated that they enjoyed the manner in which I conducted lecture, and felt I would make a good advisor for their group. My primary role as their advisor is to share my experiences as an engineer, and to convey a Faculty member's perspective on the topics they discuss. Attending their group meetings has given me an awareness of the needs and struggles that females in engineering majors face that I had never considered. I have also gained unexpected insight with respect to how they interact with one another. As compared to participating in group meetings that are primarily or entirely made up of male students, I have found Digital Women meetings have more of a social and collaborative quality to them, but still result in making great progress on a given topic. This group has become increasing active. For the past 2 years they have been sponsored by the department and Industry (e.g. Texas Instruments) to attend the annual Grace Hopper Conference for woman in technology. And this past year they coordinated with Texas Instruments to have two engineers host a soldering workshop on campus. This past year's members are shown in Figure 16.

Figure 16: Recent Digital Women group photograph. Being the mentor of this university recognized group has done much to increase my awareness female student concerns within the Electrical and Computer Engineering department.

Leadership through Engineering Academic Diversity (LEAD). This College of Engineering program aims to enhance the experience of undergraduate multicultural and women students in the College of Engineering through events and services geared toward Academic, Professional, and Social development and success. LEAD programs and services strive to increase the number of successful multicultural and female engineering graduates by helping students get connected, become leaders, improve grades, graduate sooner, and become better prepared for engineer careers.

My participation with LEAD has included: 1) speaking with students to share my experiences as a member of an underrepresented group while making my way through an undergraduate degree, 2) answering questions during round-table discussions with respect to what I do for research and what opportunities I see in industry after they complete a degree in Electrical or Computer engineering, and 3) participating in recruitment dinners for highly qualified high school Juniors and Seniors.

IT Olympics. This two day annual capstone event involves hundreds of high school students throughout Iowa for the purpose of 1) celebrating their year-long efforts toward developing IT based community service projects, and 2) competing in real-time challenges in the areas of Video Game Design, Robot Sumo, and Cyber Defense. The goal of the larger year-long "IT Adventures" program is to increase participation in technology throughout Iowa by getting high school students directly involved and using their projects to infuse technology into their local community.

I have helped organize the Game Design Competition during IT Olympics every year that I have been at Iowa State University. My responsibilities have included defining exciting real-time problems to keep students engaged during the event and judging their IT projects.

Make-to-Innovate (M:2:I). This is a program within the Aerospace Engineering Department that engages students in hands-on projects to augment their understanding of engineering fundamentals. It encourages participation of students from all engineering areas who have an interest in pursuing projects in spacecraft design, control systems, embedded systems, or robotics. By giving students the opportunity to build (and break) their creations, they are introduced to the complexity of systems and the concept of design optimization, concepts they will carry forth throughout their career.

This program has a legacy of mentoring student driven projects, and encouraging student leadership. In addition, it has attracted and retained female undergraduate participation in engineering projects. I have been a supporter of M:2:I, formally known as the Space Systems and Controls Laboratory, since I arrived at ISU. I have helped mentored their micro-satellite project called CySAT, and I have participated in technology open houses they have organized for local elementary schools.

II. Student, Peer, External and Other Assessments of Scholarship

A. Assessments of Teaching and Learning

1. Student Ratings of Teaching Effectiveness (Avg = 4.58/5.00)

4.48/5.00: Fall 2014, CPRE 288 Intro. Embedded Systems (47 students)
4.30/5.00: Spring 2014, CPRE 288 Intro. Embedded Systems (67 students)
4.52/5.00: Spring 2014, CPRE 488 Embedded Systems Design (50 students)
4.79/5.00: Fall 2013, CPRE 288 Intro. Embedded Systems (50 students)
4.50/5.00: Summer 2013, CPRE 288 Intro. Embedded Systems (35 students)
4.67/5.00: Spring 2013, CPRE 584 Models and Techniques in Embedded Systems (10 students)
4.01/5.00: Spring 2013, CPRE 288 Intro. Embedded Systems (142 students)
4.48/5.00: Fall 2012, CPRE 288 Intro. Embedded Systems (48 students)
4.75/5.00: Spring 2012, CPRE 584 Models and Techniques in Embedded Systems (8 students)
4.42/5.00: Fall 2011, CPRE 288 Intro. Embedded Systems (45 students)
5.00/5.00: Fall 2011, CPRE 583 Reconfigurable Computing (15 students: 6 of which distance)
4.63/5.00: Spring 2011, CPRE 288 Intro. Embedded Systems: Lab Instructor (29 students)
4.91/5.00: Fall 2010, CPRE 583 Reconfigurable Computing (20 students: 6 of which distance)
4.89/5.00: Spring 2010, CPRE 584X Models and Techniques in Embedded Systems(10 students)
4.58/5.00: Fall 2009, CPRE 583 Reconfigurable Computing (16 Students: 2 of which distance)
4.83/5.00: Spring 2009, CPRE 594 Embedded Systems Research Skills (9 students)
4.18/5.00: Fall 2008, CPRE 583 Reconfigurable Computing (21 students: 10 of which distance)

2. Peer Evaluations

In the Fall of 2013, two of my peers evaluated my teaching of CPRE 288 (Introduction to Embedded Systems). Below I have summarized their evaluation.

Methodology: This evaluation examined two aspects of my teaching: 1) course teaching material, and 2) my classroom presence during lecture.

Summary of Classroom Visit: the topic of the August 29th lecture for CprE 288 was on definitions and characteristics of embedded systems, and a summary of requisite background, while the topic of the September 3rd lecture was on an introduction for C programming for embedded systems. Both lectures made use of PowerPoint slides, although they were not exclusively relied on, as well as the whiteboard and a tablet computer for more detailed examples. There was a high level of interaction with the students, including several breakout sessions, and an innovative "random number" calling system for keeping students engaged with the current topic.

It was clear from these visits that Dr. Jones has a strong mastery of the subject material. CprE 288's focus on introductory embedded systems ties very close to his research interests and industry

experience, and this deeply-ingrained background comes across in lecture observations. The students in the classroom appeared to respect and appreciate his teaching style.

Feedback and Suggestions: It was noted during the class observation that the classroom itself is quite deep and narrow. Engaging students in the back rows when possible (by speaking slightly louder or walking through the room) would likely improve learning outcomes for those students who tend to gravitate to the back of a room.

Conclusion. The overall consensus of the committee is that Prof. Jones is an effective teacher. He has a firm grasp of the subject material of CprE 288, and is a highly energetic and engaging lecturer. He should be encouraged to consider the above-mentioned feedback, but otherwise to continue his effective teaching efforts.

3. Impacts

Iowa State University is a "Land-Grant University", and as such its primary directive is the teaching of practical agriculture, science, and engineering. During my time at Iowa State University, I have taken to heart the importance of conducting lectures and developing assignments that make knowledge accessible to my students. As I look back at the impact of my teaching, there are four areas where I see that my striving for excellence in teaching has yielded fruitful results: 1) academic publications, 2) research infrastructure, 3) graduate student recruitment, and 4) direct student impact.

Academic Publications. The effort I put forth when first arriving at Iowa State University to enable distance-students to have the same experience as on-campus students in my Reconfigurable Computing (CPRE 583) course helped put in place the framework that led to three publications, one in the *Proceedings of the Annual Conference of American Society for Engineering Education (ASEE)* [22], and two in the *IEEE International Conference on Microelectronic Systems Education (MSE)* [12, 26].

My Models and Techniques in Embedded Systems (CPRE 584) class' annual participation in the ACM/IEEE MemoCODE design challenge has led to a number of publications [15, 21], and a few of my student's Reconfigurable Computing (CPRE 583) class projects have led to publications [20, 23, 24]. Additionally, wining MemoCODE in 2012 and 2014 led to a couple press releases.

Research Infrastructure. My teaching efforts in CPRE 583 attracted a highly capable Master student (Matthew Clausman) to work with me for a semester. During that semester, a reconfigurable development platform called RAVI was designed and fabricated targeting research for CPS. This board has since been used to provide evidence of capability for obtaining a funded NSF-EAGER and AFOSR grant, has been used in a number of publications[3, 4, 5, 23], and has additionally been used by my MicroCART Senior Design team for controlling a UAV.

Another student from my first offering of CPRE583 (Kent Vander Velden), who has since received a PhD in computational biology, was asked by XtremeData Corporation (a high-end reconfigurable computing company) to be allowed to port and deploy his final class project (a hardware accelerated bioinformatics application) during their demo at the 2010 ACM/IEEE Super Computing conference. This lead to interactions with another high-end reconfigurable computing startup named Convey Computing. In turn my student set up initial communications between myself and Convey, which resulted in them providing their state of the art development system at a greatly reduced price with

$57,000 in donated parts from Xilinx Corporation. The industrial ties with Convey and the platform they have provided has increased the department's position for pursuing high quality research in the realm of high performance application acceleration using reconfigurable hardware technology. In 2014 I receive another $64,000 in donated parts from Xilinx to upgrade our Convey system.

The Convey Platform has been used to win the MemoCODE design competition in 2012 [15] and 2014 [21], aid in a number of publications [1, 10, 13, 16, 17], and now several PhD students are using this as their primary platform for conducting research.

Graduate Student Recruitment. My teaching of CPRE 288, CPRE 488, CPRE 583, CPRE 584, and mentoring Senior Design has resulted in a number of students deciding to attend graduate school, and/or continuing after their MS to pursue a PhD. Currently I have a couple undergraduate students conducting research with me, motivated by their experience in my under graduate embedded systems courses (CPRE 288 and CPRE 488), who will now likely continue their education at Iowa State University for an MS degree, and I am hoping to convince them to pursue PhD.

Direct Student Impact. Throughout my time at Iowa State University, I have had a number of students send me emails and letters in appreciation of my teaching and mentoring efforts. These communications have made clear the real-world impacts that my duties as an educator have on students' lives. I would like to share some of these comments.

Xin Zhao (email from a former CPRE 583 student, now at SanDisk): "I am staring to use FPGAs to do ASIC prototyping verifications… I am using 3 to 4 FPGA because the design is too big… I am using the muxing I/O technique I learned in your class… Thank you again".

Christy and Matt (letter from student leads of the IEEE Xtreme Programming Challenge): "Thank you for helping us again with IEEExtreme. We truly appreciate it! Enjoy the pie!"

Cimone Wright (letter from student mentor within the LEAD program): "Thank you for representing the CprE Dept. at the LEAD roundtable… The students appreciate your time."

Cimone Wright (letter): "Thank you for the writing book you gave me. It has helped tremendously!"

Sarah, Max, Tommy, Dillon (letter from Freshmen within "Take a Professor to lunch program"): "Thank you for meeting with us. Your answers to our questions is an invaluable resource"

4. Other Assessments

In 2014, I was awarded the Warren B. Boast Undergraduate Teaching Award from my department.

During the 2009-2010 academic year, I successfully completed the **Center for Excellence in Learning and & Teaching (CELT) Teaching Partners Program**. In this program, I was paired with a senior lecturer (Cinzia Cervato, Full Professor from Geology) and another Assistant Professor. I attended regular meetings with my senior partner to discuss her experiences in teaching, observed her teaching, received feedback from her on my teaching, and attended a number of CELT classes related to improving teaching effectiveness. This was a great experience and helped strengthen my lecturer presence.

B. Assessments of Research and Creative Activities

1. Summary of Citations for up to 10 Publications

H-index = 11,
Citations from: Google Scholar: "Phillip H. Jones" (6/2015),
Impact factors from: Journal Citation Reports (JCR) 2009,

1. An Experimental Evaluation of the REE SIFT Environment for Spaceborne Applications,
by K. Whisnant, R.K. Iyer, P. Jones, R. Some, D. Rennels;
International Conference on Dependable Systems and Networks (DSN), Washington, D.C., USA, 2002
(30 citations)

2. Extracting and Improving Microarchitecture Performance on Reconfigurable Architectures,
by Shobana Padmanabhan, Phillip Jones, David V. Schuehler, Scott J. Friedman, Praveen Krishnamurthy, Huakai Zhang, Roger Chamberlain, Ron K. Cytron, Jason Fritts, and John W. Lockwood;
International Journal of Parallel Programming, Volume 33, Issue 2 - 3, June 2005, Pages 115 - 136.
(20 citations, Impact factor = .82)

3. Liquid Architecture,
by Phillip Jones, Shobana Padmanabhan, Daniel Rymarz, John Maschmeyer, David V. Schuehler, John W. Lockwood, and Ron K. Cytron;
IEEE International Parallel and Distributed Symposium (IPDPS), Next Generation Software (NGS) Workshop, Santa Fe, New Mexico, April 26, 2004.
(19 citations)

4. The Effects of an ARMOR-based SIFT Environment on the Performance and Dependability of User Applications,
by K. Whisnant, R.K. Iyer, Z.T. Kalbarczyk, P.H. Jones III, D.A. Rennels, R. Some;
IEEE Transactions on Software Engineering (TSE), Volume 30, Issue 4, April 2004, Pages 257 - 277.
(18 citations, Impact factor = 3.57)

5. A Thermal Management and Profiling Method for Reconfigurable Hardware Applications,
by Phillip H. Jones, John W. Lockwood, and Young H. Cho;
IEEE International Conference on Field Programmable Logic and Applications (FPL), Madrid, Spain, Aug 28-30, 2006.
(17 citations)

6. Cycle-Accurate Microarchitecture Performance Evaluation,
by Richard Hough, Phillip Jones, Scott Friedman, Roger Chamberlain, Jason Fritts, John Lockwood, Ron Cytron;
IEEE Workshop on Introspective Architecture (WISA), Austin, TX, February 2006.
(15 citations)

7. Adaptive Thermoregulation for Applications on Reconfigurable Devices,
by Phillip H. Jones, James Moscola, Young H. Cho, and John W. Lockwood;

IEEE International Conference on Field Programmable Logic and Applications (FPL), Amsterdam, Netherlands, Aug 27-29, 2007.
(13 citations)

8. Characterizing Non-Ideal Impacts of Reconfigurable Hardware Workloads on Ring Oscillator-based Thermometers,
by Moinuddin Sayed and Phillip Jones;
IEEE International Conference on Reconfigurable Computing and FPGAs (Reconfig), Cancun, Mexico, Nov 3-Dec 2, 2011.
(11 citations)

9. An I/O Bandwidth-Sensitive Sparse Matrix-Vector Multiplication Engine on FPGAs,
by S. Sun, M. Monga, P. Jones, and J. Zambreno;
IEEE Transactions on Circuits and Systems-I (TCAS-I), Volume 59, no. 1, January 2012, Pages 113-123.
(11 citations, Impact factor = 2.3)

10. Hotspot Mitigation using Dynamic Partial Reconfiguration for Improved Performance,
by Adwait Gupte, and Phillip H. Jones;
IEEE International Conference on Reconfigurable Computing and FPGAs (Reconfig), Cancun, Mexico, Dec 9-11, 2009
(11 citations)

2. Evidence of Impact on Society of Research and Creative Activities

I would place my work in the area of hardware support for embedded CPS as what I hope to be the most impactful in the near future; more specifically, exploring hardware architectures to bridge the communication gap between controls and embedded software engineers. One reviewer of my recently accepted FPL paper [9] expressed best an aspect of this impact that I would like to see further propagate into the research community:

"The work demonstrates how the benefits of FPGA acceleration may be enabled in a more approachable way for traditional control engineers whose core competencies do not typically include embedded hardware-software co-design."

Recently I have also began talks with John Deere to see what collaboration opportunities may exist to apply this aspect of my research to the design and coordination of their agricultural machinery.

3. Other Assessments

1st Place, 2014 ACM-IEEE MemoCODE Design Competition:
A High Performance Systolic Architecture for k-NN Classification,
by K. Townsend, P. Jones and J. Zambreno;
Proceedings of the International Conference on Formal Methods and Models for Codesign (MEMOCODE), Lausanne, Switzerland, October, 2014

1st Place, 2012 ACM-IEEE MemoCODE Design Competition:
Shepard: A Fast Exact Match Short Read Aligner,
by C. Nelson, K. Townsend, B S. Rao, P. Jones and J. Zambreno;
Proceedings of the International Conference on Formal Methods and Models for Codesign (MEMOCODE), July, 2012

Best paper award VLSI Design 2007:
Dynamically Optimizing FPGA Applications by Monitoring Temperature and Workloads,
by Phillip H. Jones, Young H. Cho, and John W. Lockwood;
IEEE International Conference on VLSI Design (VLSI Design), Bangalore, India, Jan 6-10, 2007.

C. Assessments of Extension/Professional Practice and Service

N/A

III. Future Plans

As I continue my academic career, I have a number of near and long-term plans for both research and education.

Research. Cyber Physical Systems (CPS) is an exciting and growing field where I look forward to continuing to employ reconfigurable hardware to bring innovations to the computing stack. I have just scratched the surface of this research front with respect to HW/SW co-design approaches to support controls and real-time aspects of embedded CPS applications.

Three short term plans I have over the next 3-4 years are: 1) investigating more thoroughly the design space of control algorithm complexity and supporting hardware architectures, 2) developing and refining mechanisms and methodologies to bridge the gap between Controls engineers and Embedded Systems engineers, and 3) continuing to identify potential industrial partners (e.g. John Deere) to help magnify the impact of my work.

In the longer term, I would like to identify deeper generalizable results that can be more broadly applied to the design of embedded CPS.

Education. I am excited to continue curriculum innovation at Iowa State University. One vision I would like to drive into our curriculum is facilitating communication between our students focused on Embedded System design, and those focused on Controls design. Over the short term (2-3 years), I plan to leverage the infrastructure I have developed for research to infuse more hands-on implementation assignments into our two undergraduate Controls courses (EE 475 and EE 476). Over the slightly longer term (3-5 years), I would like to work toward coupling my newly re-envisioned Junior/Senior level Embedded Systems course (CPRE 488) with these Controls courses. Ideally, I would like 2-3 assignments that are common to both courses involving teams composed of students from both CPRE 488 and EE 476. The end goal being to expose students focused on Embedded HW/SW design to Controls concepts more deeply, and vise-a-verse for students focused on Controls. Also through this process, students would have an opportunity to learn the "language" of students from the complementary discipline.

Closing Thoughts. I can honestly say my time at Iowa State University has been among the most rewarding of my life. Both the opportunities I have been presented, and challenges I have faced while progressing through the tenure process have forced me to mature and grow in ways that I do not think any other experience would have achieved. I am thankful for the journey I have taken thus far, and look forward to continuing to grow and contribute to society through academia.