REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)
05-02-2016

2. REPORT TYPE
Final Report

3. DATES COVERED (From - To)
1-Mar-2011 - 31-May-2014

4. TITLE AND SUBTITLE
Final Report: Hardware-Enabled Security Through On-Chip Reconfigurable Fabric

5a. CONTRACT NUMBER
W911NF-11-1-0082

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
611102

6. AUTHORS
G. Edward Suh

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES
Cornell University
373 Pine Tree Road

Ithaca, NY 14850 -2820

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
58256-CS-YIP.11

12. DISTRIBUTION AVAILIBILITY STATEMENT
Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT

The goal of this project was to enable hardware-based security techniques on future microprocessors in a way that they can be added and updated after fabrication, similar to software, while maintaining the efficiency and the security of hardware. For this purpose, the project investigated programmable architectures based on processing cores, on-chip reconfigurable fabric, and custom accelerators where security techniques can be implemented after chip fabrication. The study showed that such programmable architectures can indeed support a broad range of run-time monitoring techniques that detect errors and attacks with low overhead. The project also enabled fine-grained

15. SUBJECT TERMS
Security, Run-Time Monitoring, Hardware, Steganography

16. SECURITY CLASSIFICATION OF:

a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES
UU	UU	UU	UU	

19a. NAME OF RESPONSIBLE PERSON
G. Edward Suh

19b. TELEPHONE NUMBER
607-255-6856

Report Title

Final Report: Hardware-Enabled Security Through On-Chip Reconfigurable Fabric

ABSTRACT

The goal of this project was to enable hardware-based security techniques on future microprocessors in a way that they can be added and updated after fabrication, similar to software, while maintaining the efficiency and the security of hardware. For this purpose, the project investigated programmable architectures based on processing cores, on-chip reconfigurable fabric, and custom accelerators where security techniques can be implemented after chip fabrication. The study showed that such programmable architectures can indeed support a broad range of run-time monitoring techniques that detect errors and attacks with low overhead. The project also enabled fine-grained run-time monitoring for real-time systems by developing static and dynamic mechanisms to ensure that a monitored system still meets real-time deadlines. In addition to run-time monitoring, the project also investigated exploiting inherent variations and noise in off-the-shelf Flash memory to build hardware security functions, and showed that Flash memory can be used to securely hide information.

Enter List of papers submitted or published that acknowledge ARO support from the start of the project to the date of this printing. List the papers, including journal references, in the following categories:

(a) Papers published in peer-reviewed journals (N/A for none)

Received	Paper

TOTAL:

Number of Papers published in peer-reviewed journals:

(b) Papers published in non-peer-reviewed journals (N/A for none)

Received	Paper

TOTAL:

Number of Papers published in non peer-reviewed journals:

(c) Presentations

Number of Presentations: 0.00

Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received	Paper

TOTAL:

Number of Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received		Paper
02/02/2016	5.00	Daniel Y. Deng, G. Edward Suh. High-performance parallel accelerator for flexible and efficient run-time monitoring, 2012 42nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). 25-JUN-12, Boston, MA, USA. : ,
02/02/2016	7.00	Daniel Lo, G. Edward Suh. Worst-case execution time analysis for parallel run-time monitoring, the 49th Annual Design Automation Conference. 03-JUN-12, San Francisco, California. : ,
02/02/2016	6.00	Yao Wang, Andrew Ferraiuolo, G. Edward Suh. Timing channel protection for a shared memory controller, 2014 IEEE 20th International Symposium on High Performance Computer Architecture (HPCA). 15-FEB-14, Orlando, FL, USA. : ,
02/03/2016	8.00	Daniel Lo, Mohamed Ismail, Tao Chen, G. Edward Suh. Slack-aware opportunistic monitoring for real-time systems, 2014 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS). 15-APR-14, Berlin, Germany. : ,
02/03/2016	10.00	Mohamed Ismail, Daniel Lo, G. Edward Suh. Improving worst-case cache performance through selective bypassing and register-indexed cache, the 52nd Annual Design Automation Conference. 07-JUN-15, San Francisco, California. : ,
02/03/2016	9.00	Daniel Lo, Tao Chen, Mohamed Ismail, G. Edward Suh. Run-time monitoring with adjustable overhead using dataflow-guided filtering, 2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA). 07-FEB-15, Burlingame, CA, USA. : ,
06/12/2012	2.00	Daniel Y. Deng, G. Edward Suh. Precise Exception Support for Decoupled Run-TimeMonitoring Architectures, International Conference on Computer Design. 10-OCT-11, . : ,
09/01/2011	1.00	Daniel Lo, Greg Malysa, G. Edward Suh. FlexCache: Field Extensible Cache ControllerArchitecture Using On-Chip Reconfigurable Fabric, International Conference on Field Programmable Logic and Applications . 05-SEP-11, . : ,
09/04/2013	3.00	Mohamed Ismail, G. Edward Suh. Fast Development of Hardware-Based Run-Time Monitors Through Architecture Framework and High-Level Synthesis, the 30th International Conference on Computer Design (ICCD). 30-SEP-12, . : ,
09/04/2013	4.00	Yinglei Wang, Wing-kei Yu, Sarah Q. Xu, Edwin Kan, G. Edward Suh. Hiding Information in Flash Memory, IEEE Symposium on Security and Privacy. 20-MAY-13, . : ,

TOTAL: **10**

Number of Peer-Reviewed Conference Proceeding publications (other than abstracts):

(d) Manuscripts

Received	Paper

TOTAL:

Number of Manuscripts:

Books

Received	Book

TOTAL:

Received	Book Chapter

TOTAL:

Patents Submitted

Methods and systems for providing hardware security functions using flash memories, US 14/401,974, Filed May 17, 2013.

Patents Awarded

Awards

Graduate Students

NAME	PERCENT_SUPPORTED	Discipline
Tao Chen	0.75	
Mohamed Ismail	0.19	
FTE Equivalent:	**0.94**	
Total Number:	**2**	

Names of Post Doctorates

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Names of Faculty Supported

NAME	PERCENT_SUPPORTED	National Academy Member
G. Edward Suh	0.10	
FTE Equivalent:	**0.10**	
Total Number:	**1**	

Names of Under Graduate students supported

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Student Metrics

This section only applies to graduating undergraduates supported by this agreement in this reporting period

The number of undergraduates funded by this agreement who graduated during this period: 0.00

The number of undergraduates funded by this agreement who graduated during this period with a degree in science, mathematics, engineering, or technology fields:...... 0.00

The number of undergraduates funded by your agreement who graduated during this period and will continue to pursue a graduate or Ph.D. degree in science, mathematics, engineering, or technology fields:...... 0.00

Number of graduating undergraduates who achieved a 3.5 GPA to 4.0 (4.0 max scale):...... 0.00

Number of graduating undergraduates funded by a DoD funded Center of Excellence grant for Education, Research and Engineering:...... 0.00

The number of undergraduates funded by your agreement who graduated during this period and intend to work for the Department of Defense 0.00

The number of undergraduates funded by your agreement who graduated during this period and will receive scholarships or fellowships for further studies in science, mathematics, engineering or technology fields:...... 0.00

Names of Personnel receiving masters degrees

NAME	
Yinglei Wang	
Total Number:	**1**

Names of personnel receiving PHDs

NAME	
Daniel Lo	
Total Number:	**1**

Names of other research staff

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Sub Contractors (DD882)

Inventions (DD882)

Scientific Progress

The goal of this project was to enable hardware-based security techniques on future microprocessors in a way that they can be added and updated after fabrication, similar to software, while maintaining the efficiency and the security of hardware. To achieve this goal, the project investigated how to enable a wide range of hardware security techniques in a programmable fashion or on an existing platforms. The list includes cache extensions (FPL'11), instruction-grained run-time program monitors (ICCD'11, DSN'12, ICCD'12), and steganography in Flash memory (IEEE S&P'13). The project also developed techniques to enable run-time monitoring for real-time systems that require timely responses (DAC'12, RTAS'14, HPCA'15). Finally, the investigation of real-time systems also led to a new performance optimization (DAC'15) and timing channel protection (HPCA' 14).

The following paragraphs discuss each major technical development in more detail.

1. FlexCache: Extensible Cache Architecture (FPL'11)

In today's microprocessors, the cache architecture is highly optimized for one particular design and cannot be changed after fabrication. While allowing efficient implementations in dedicated logic, this inflexibility also implies that new techniques cannot be deployed in the field.

To solve this problem, we designed a new flexible cache architecture, named, FlexCache, which uses on-chip reconfigurable fabric to enable new extensions to be added in the field after fabrication. The key idea in this architecture is to perform frequent operations such as a cache hit with dedicated hardware, while allowing less frequent operations such as a cache miss to be customized using reconfigurable fabric.

We evaluated the flexibility and efficiency of the architecture through an RTL prototype implementation of the cache along with example extensions such as cache performance counters, side-channel protection, prefetching, various replacement policies and computation acceleration. The results show that various types of extensions can be realized on FlexCache with minimal impact on performance, power, and area.

2. Instruction-Grained Run-Time Program Monitoring

A large part of this project was focused on developing efficient and flexible run-time program monitoring framework. The run-time monitoring is designed to check individual instructions at run-time, and detect anomaly.

The project started with a previously developed architecture, named FlexCore (MICRO'10), as the baseline. FlexCore is a hybrid processor architecture where an on-chip reconfigurable fabric (FPGA) is tightly coupled with the main processing core. FlexCore supports a broad range of run-time monitoring and bookkeeping techniques by programming the reconfigurable fabric.

- Precise Exception Support (ICCD'11)

Parallel monitoring framework such as FlexCore decouples the execution of the monitored program from the monitor in order to minimize performance impacts. Yet, such decoupled architectures lack support for precise exceptions and can only detect an exception (attack) after the monitored program completes an erroneous operation.

In this project, we developed a new architectural mechanism to support precise exceptions in non-speculative processors with decoupled monitors. The main is to create a shadow architecture state, which allows rolling back the processor state in case of an exception. Experimental results based on an RTL implementation show that our approach has low area, power, and performance overheads even when applied to simple, in-order processors.

- High-Performance Monitoring Accelerator (DSN'12)

While being flexible, typical reconfigurable fabric such as FPGAs has a rather limited clock frequency because of unpipelined routing. For example, we found that FlexCore could support monitoring for processors with moderate clock frequencies (<500MHz) with low overhead, but incurs high overhead for high-performance processors.

To address this limitation, we designed a high-performance monitoring accelerator, named Harmoni. The Harmoni architecture achieves much higher efficiency than software implementations and previously proposed monitoring platforms such as FlexCore by closely matching the common characteristics of run-time monitoring functions using the notion of tagging. In essence, the key idea is to encode monitoring operations as tagging operations where each instruction, register, and memory location carries a tag encoding security properties.

We implemented an RTL prototype of Harmoni and evaluated it with several example monitoring functions for security and programmability. The prototype demonstrates that the architecture can support a wide range of monitoring functions with different characteristics. Harmoni takes moderate silicon area, has very high throughput, and incurs low overhead on monitored programs even on processors with a high frequency (2.5GHz).

- Fast Design Framework (ICCD’12)

The parallel monitoring architecture requires each monitor to be designed as custom hardware module. Unfortunately, this cost and time for implementing each hardware monitor presents a major obstacle in deploying the run-time monitoring techniques in real systems.

This project addressed the design complexity problem by developing a common architecture framework and using high-level synthesis. Similar to customizable processors such as Tensilica Xtensa where designers only need to write a small piece of code that describes a custom instruction, our framework enables designers to only specify monitoring operations. The framework provides common functions such as collecting a trace of execution, maintaining meta-data, and interfacing with software. To further reduce the design complexity, we also explored using a high-level synthesis tool (Cadence C-to-Silicon) so that hardware monitors can be described in a high-level language (SystemC) instead of in RTL such as Verilog and VHDL.

To evaluate our approach, we implemented a set of monitors including soft-error checking, uninitialized memory checking, dynamic information flow tracking, and array boundary checking in our framework. Our results suggest that our monitor framework can greatly reduce the amount of code that needs to be specified for each extension, from 2105-to-2626 lines to 46-to-209 lines. The high-level synthesis was also found to be effective, achieving comparable area, performance, and power consumption to handwritten RTL.

3. Run-Time Monitoring for Real-Time Systems

The increasing safety-critical role of real-time systems such as automotive control systems requires increased attention to their security and reliability. Yet, the run-time monitoring techniques that we developed cannot be applied to hard real-time systems without a way to ensure that the monitoring will not lead to deadline misses. In this project, we developed a set of techniques to enable parallel program monitoring for real-time systems.

- Static WCET Analysis (DAC'12)

Traditional real-time system designs use the worst-case execution time (WCET) estimate of each task to design a system with timing guarantees. In this project, we developed the first analysis method for statically determining the impact of parallel monitoring on WCET using a mixed integer linear programming (MILP) formulation. The key innovation in this work was the method to model a FIFO between a monitored core and a monitor, and analytically capturing cases when the FIFO may be full. We used our method to estimate the WCET for seven benchmark programs and two possible monitoring techniques. This estimate was compared against observed execution times from simulation and a conservative upper bound based on sequential monitoring. The results showed that our method is far more accurate compared to the conservative bound. It estimates WCETs within 71% of worst-case observed execution times and up to 74% lower than the sequential bound.

- Run-Time Management for Hard Real-Time Systems (RTAS’14)

The WCET estimate above requires a system to be designed with an enough additional slack (margin) in order to deploy run-time monitoring. For the cases when such a slack is not an option or too expensive, we also developed a run-time mechanism to ensure the WCET of a monitored program.

In this framework, we leverage the fact that programs typically run faster than its WCET. Monitoring is only performed when enough dynamic slack exists in order to ensure that the monitoring does not impact the timing guarantees of tasks. If the slack is insufficient, our framework drops a monitoring operation while ensuring that there is no impact on the monitored task and that there is no false positive in the future. For efficient dropping, we developed a novel hardware architecture that can perform this dropping operation in a single cycle, matching the throughput of the task being monitored.

Thus, run-time monitoring can be applied opportunistically, with no impact on the worst-case execution time of tasks. Our experimental results for three different monitoring techniques verify that timing is never violated and that false positives never occur. In addition, on average, 15-66% of monitoring coverage is achieved on a multi-core platform with no impact on the worst-case execution times of tasks depending on the monitoring technique. With an FPGA-based monitor such as FlexCore, this average coverage of monitoring ranged from 62-86% depending on the monitoring technique.

- Coverage-Overhead Trade-off on Soft Real-Time Systems (HPCA’15)

The idea of only performing a subset of monitoring can be leveraged to not only enable monitoring of hard real-time systems, but also enable the trade-off between coverage and overhead. With this intuition, we extended our framework in RTAS’14 with a new hardware data-flow tracking engine that enables adjustable overhead through partial monitoring. This new framework

enables users to specify a desired overhead level for run-time monitoring. The dataflow engine was also extended to filter out monitoring operations associated with null metadata in order to reduce overhead. Given this architecture, we investigated how the dropping decisions should be made for partial monitoring and show that there exist interesting policy decisions depending on the target application of partial monitoring. Our experimental results show that overhead can be reduced significantly by trading off coverage. For example, for monitoring techniques with average overheads of 2-6x, the proposed architecture is able to reduce overhead to 1.5x while
still achieving 14-85% average coverage.

4. Flash Memory Steganography (IEEE S&P 2013)

Recently, researchers showed that analog characteristics of integrated circuits can be used to build new hardware security functions such as device fingerprints. Yet, these techniques require building new custom circuits.

In this project, we found that some of the analog characteristics of off-the-shelf Flash memory can be measured externally through a standard digital interface. From this observation, we developed a novel information hiding technique for Flash memory that can be implemented on off-the-shelf Flash memory chips without custom circuits. The method hides data within an analog characteristic of Flash, the program time of individual bits. The program time of each bit gets longer as the corresponding memory cell wears out, and we can change them individually by repeatedly writing the memory while controlling which value is written to each cell. Then, the relative differences in the program time can be used to store bits intentionally.

Because the technique uses analog behaviors, normal Flash memory operations are not affected and hidden information is invisible in the data stored in the memory. Even if an attacker checks a Flash chip’s analog characteristics, experimental results indicate that the hidden information is difficult to distinguish from inherent manufacturing variation or normal wear on the device. Moreover, the hidden data can survive erasure of the Flash memory data, and the technique can be used on current Flash chips without hardware changes.

5. Other Work

Our work on enabling the run-time monitoring for real-time systems also led to a few new techniques related to the timing of a system.

- WCET Optimization (DAC15)

Worst-case execution time (WCET) analysis is a critical part of designing real-time systems that require strict timing guarantees. However, data caches have traditionally been challenging to analyze in the context of WCET due to the unpredictability of memory access patterns. We developed a new cache structure, namely register-indexed cache, that is designed to be more amenable to static analysis compared to traditional caches. This is based on the idea that absolute addresses may not be known, but by using relative addresses, analysis may be able to guarantee a number of hits in the cache. In addition, we observed that keeping unpredictable memory accesses in caches could increase or decrease WCET depending on the application. Thus, we explored selectively bypassing caches in order to provide lower WCET. Our experimental results showed reductions in WCET of up to 35% over the state-of-the-art static analysis.

- Timing Channel Protection for Shared Memory (HPCA’14)

We found that that shared memory controllers are vulnerable to both side channel and covert channel attacks that exploit memory interference as timing channels. To address this vulnerability, we designed a secure memory controller that enables secure sharing of main memory among mutually mistrusting parties by eliminating memory timing channels. To eliminate timing channels, we identified the sources of interference in a conventional memory controller design, and proposed a protection scheme to eliminate the interference across security domains through two main changes: (i) a per security domain based queueing structure, and (ii) static allocation of time slots in the scheduling algorithm. Multi-programmed workloads comprised of SPEC2006 benchmarks were used to evaluate the protection scheme. The results show that the proposed scheme completely eliminates the timing channels in the shared memory with small hardware and performance overheads.

Technology Transfer

The techniques to use COTS Flash memory for security functions attracted some commercial interest. A company named CybKey Tech expressed interest in evaluating the technology. Researchers in CERT (CMU) also asked help in replicating the results – we provided necessary hardware and software.