Massachusetts Institute of Technology
32 Vassar Street, 32-G730
Cambridge, MA 02139
510.378.2615
mcarbin@csail.mit.edu
http://people.csail.mit.edu/mcarbin

Michael Carbin

Research Interests and Vision

Theory, design, and implementation of **programming systems**, including languages, program logics, static and dynamic program analyses, runtime systems, and mechanized verifiers. My research focuses on programming systems that deliver greater performance and resilience by changing the underlying semantics of the program in ways that are still sound and principled.

Education

Summer 2014 (Expected)
Ph.D., Electrical Eng. and Computer Science, Massachusetts Institute of Technology.
Thesis: Verifying the Reliability and Acceptability of Relaxed Approximate Programs
Advisor: Martin Rinard

August 2009
S.M., Electrical Eng. and Computer Science, Massachusetts Institute of Technology.
Thesis: Automatically Identifying Critical Behaviors in Programs
Advisor: Martin Rinard

June 2006
B.S., Computer Science, Stanford University.
Honors Thesis: Learning Effective BDD Variable Orders for BDD-Based Program Analysis
Award: Wegbreit Prize for Best Computer Science Undergraduate Honors Thesis
Advisor: Monica Lam

Research Experience

Massachusetts Institute of Technology.

Research Assistant

Reliability-Aware Optimization for Unreliable Hardware. (2013–present)
Design and implementation of the Rely programming language along with a compiler and program analysis for reliability-aware optimization of programs by compiling them to low-power unreliable hardware [1].

Reasoning about Acceptability Properties of Relaxed Programs. (2007–present)
Design and implementation of a language and verification system in Coq for verifying programs that have been extended with additional nondeterminsim to broaden the set of behaviors available for boosting performance and resilience [3, 14]. Design and implementation of dynamic execution and taint tracking analysis for detecting regions of a program that are tolerant to semantic change [7].

Modifying Programs for Increased Performance. (2010–present)
Design and implementation of program analysis system for verifying the safety of skipping loop iterations to increase performance [13]. Design and implementation of a dynamic program analysis for automatically reconfiguring a program through memory state transformations, thereby enabling dynamic reconfiguration of programs to address performance/power concerns [6].

Modifying Programs for Increased Resilience. (2008–present)
Design and implementation of the Jolt/Bolt system, a LLVM and Pin-based static/dynamic program analysis and runtime system for detecting and escaping infinite loops in end-user programs [5, 2]. Other techniques for dynamically enforcing program invariants to create secure programs by dynamically steering program execution [8, 4].

Microsoft Research.

Graduate Intern

Safe Interprocess Communication with Unbounded Queues. (Summer 2007)
Design and implementation of a language and compiler for safe, asynchronous interprocess communication with restricted unbounded queues in the Singularity Operating System.
Mentor: Manuel Fähndrich

Undergrad. Intern

Typesafe Compile-time Reflective Metaprogramming. (Summer 2005)
Design and implementation of a C# language extension for typesafe compile-time reflection in the Singularity Operation System [10].
Mentor: Jim Larus

Stanford University.

Undergrad. Researcher

Efficient BDD-based Program Analysis. (2004–2006)
Design and implementation of methods for optimizing Datalog-specified program analyses for bddbddb, including dataflow-based Datalog query optimization and learning effective BDD variable orders using machine learning [11, 12].
Advisor: Monica Lam

Language Design for Hardware/Software Transactional Memory. (2005–2006)
Design and implementation of Java language and virtual machine extensions for hardware/software transactional memory and transactional collection classes using Jikes RVM [9].
Advisor: Kunle Olukotun

Publication Overview

Published 12 peer-reviewed conference papers (1 best paper award) and 3 peer-reviewed workshop/short papers in venues such as PLDI, OOPSLA, SOSP, ASPLOS, ICSE, PODS, ECOOP, PPOPP, ISSTA, PEPM, APLAS, and GPCE. Given 13 conference/workshop/invited talks (1 best talk award) and received 2 patents.

Selected Publications

Michael Carbin, Sasa Misailovic, and Martin Rinard. Verifying quantitative reliability for programs that execute on unreliable hardware. In *Object-Oriented Programming, Systems, Languages & Applications (OOPSLA)*, [**Best Paper Award**], 2013.

Michael Carbin, Deokhwan Kim, Sasa Misailovic, and Martin Rinard. Proving acceptability properties of relaxed nondeterministic approximate programs. In *Programming Languages Design and Implementation (PLDI)*, 2012.

Michael Carbin, Sasa Misailovic, Michael Kling, and Martin Rinard. Detecting and escaping infinite loops with Jolt. In *European Conference on Object-Oriented Programming (ECOOP)*, 2011.

Michael Carbin and Martin Rinard. Automatically identifying critical input regions and code in applications. In *International Symposium on Software Testing and Analysis (ISSTA)*, 2010.

Awards

2013	Best Paper Award, Object-Oriented Programming, Systems, Languages & Applications (OOPSLA)
2013	Best Student Talk, Principles of Programming Languages (POPL) Student Session
2008-2010	Microsoft Research Graduate Fellowship
2006-2007	MIT Lemelson Presidential Fellowship
2006	Stanford Wegbreit Prize for Best Computer Science Undergraduate Honors Thesis
2003	Winner of Stanford CS143 Compiler Contest

Press

Programming Unreliable Hardware [**1**].

- **Slashdot.** *New Framework for Programming Unreliable Chips.* http://bit.ly/JpvfQK
- **MITNews.** *How to Program Unreliable Chips.* http://bit.ly/1bSbkEQ

Detecting and Escaping Infinite Loops [**5, 2**].

- **Slashdot.** *Escaping Infinite Loops.* http://bit.ly/1dEEkyD
- **ArsTechnica.** *Framework lets users force hung programs to recover.* http://bit.ly/1bJaJEy
- **MITNews.** *Defibrillator for stalled software.* http://bit.ly/1dgrV3y

Reasoning About Relaxed Programs [**3**].

- **MITNews**. *New mathematical framework formalizes oddball programming techniques.* http://bit.ly/1idGVVl

Enforcing Program Invariants for Software Self-Healing [**8**].

- **Slashdot**. *Fixing Bugs, But Bypassing Source Code.* http://bit.ly/18PB4k8
- **MIT Technology Review**. *Software That Fixes Itself.* http://bit.ly/1973FUD
- **MITNews**. *Web sites that can take a punch.* http://bit.ly/1c65IbO

Teaching, Mentorship, Grants

Spring 2013 **Teaching Assistant for MIT undergraduate course 6.035: Computer Language Engineering**. Prepared compiler project materials, created and graded problem sets, created and graded two midterms and a final exam, and gave four lectures: *Introduction to Program Analysis*, *Dataflow Analysis*, and *Parallelization* (2 part).

2009-present **Student Mentor.** Advised multiple high school, undergraduate, masters, and younger graduate students while at MIT. Mentored two masters theses:

- Michael Kling, *Detecting and Escaping Infinite Loops Using Bolt* [2] (Feb. 2012).
- Tal Tchwella, *ComFIT: Comprehensive Fault Injection Tool* (expected Summer 2014).

2011-present Worked on grants, attended PI meetings, and presented results at site visits:

- **DOE**: X-Stack
- **DARPA**: Mission-Oriented Resilient Clouds

Service

Program Committee

- **CGO**: Code Generation and Optimization (2015)
- **ISSRE**: International Symposium on Software Reliability Engineering (2014)

Reviewer

- **PLDI**: Programming Language Design and Implementation (2012)
- **POPL**: Principles of Programming Languages (2013)
- **OOPSLA**: Object-Oriented Programming, Systems, Languages & Applications (2013)
- **MICRO**: Symposium on Microarchitecture (2013)
- **CGO**: Code Generation and Optimization (2013).

Member **MIT EECS Faculty Hiring Student Committee**. Attended faculty talks, interviewed candidates, and rendered opinions on each candidate (Spring 2013).

Student Representative **MIT EECS Visiting Committee**. Met with the EECS Visiting Committee and gave personal perspective on the EECS Department and student life (Spring 2013).

Mentor **EECSCon.** Mentored undergraduate poster and presentation for the MIT EECS undergraduate conference, EECSCon (Spring 2013).

Volunteer **HackMIT**. Arranged graduate student supervisors for hackathon (Fall 2013).

Conference Publications

1. Michael Carbin, Sasa Misailovic, and Martin Rinard. Verifying quantitative reliability for programs that execute on unreliable hardware. In *Object-Oriented Programming, Systems, Languages & Applications (OOPSLA)*, [**Best Paper Award**], 2013.

2. Michael Kling, Sasa Misailovic, Michael Carbin, and Martin Rinard. Bolt: on-demand infinite loop escape in unmodified binaries. In *Object-Oriented Programming, Systems, Languages & Applications (OOPSLA)*, 2012.

3. Michael Carbin, Deokhwan Kim, Sasa Misailovic, and Martin Rinard. Proving acceptability properties of relaxed nondeterministic approximate programs. In *Programming Languages Design and Implementation (PLDI)*, 2012.

4. Fan Long, Vijay Ganesh, Michael Carbin, Stelios Sidiroglou, and Martin Rinard. Automatic input rectification. In *International Conference of Software Engineering (ICSE)*, 2012.

5. Michael Carbin, Sasa Misailovic, Michael Kling, and Martin Rinard. Detecting and escaping infinite loops with Jolt. In *European Conference on Object-Oriented Programming (ECOOP)*, 2011.

6. Hank Hoffmann, Stelios Sidiroglou, Michael Carbin, Sasa Misailovic, Anant Agarwal, and Martin Rinard. Dynamic knobs for responsive power-aware computing. In *Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2011.

7. Michael Carbin and Martin Rinard. Automatically identifying critical input regions and code in applications. In *International Symposium on Software Testing and Analysis (ISSTA)*, 2010.

8. Jeff Perkins, Sunghun Kim, Sam Larsen, Saman Amarasinghe, Jonathan Bachrach, Michael Carbin, Carlos Pacheco, Frank Sherwood, Stelios Sidiroglou, Greg Sullivan, Weng-Fai Wong, Yoav Zibin, Michael Ernst, and Martin Rinard. Automatically patching errors in deployed software. In *Symposium on Operating Systems Principles (SOSP)*, 2009.

9. Brian Carlstrom, Austen McDonald, Michael Carbin, Christos Kozyrakis, and Kunle Olukotun. Transactional collection classes. In *Principles and Practices of Parallel Programming (PPOPP)*, 2007.

10. Manuel Fähndrich, Michael Carbin, and James Larus. Reflective program generation with patterns. In *Generative Programming and Component Engineering (GPCE)*, 2006.

11. Monica Lam, John Whaley, Ben Livshits, Michael Martin, Dzintars Avots, Michael Carbin, and Christopher Unkel. Context-sensitive program analysis as database queries. In *Principles of Database Systems (PODS) (Invited)*, 2005.

12. John Whaley, Dzintars Avots, Michael Carbin, and Monica Lam. Using datalog with binary decision diagrams for program analysis. In *Asian Symposium on Programming Languages and Systems (APLAS)*, 2005.

Peer-Reviewed Workshop and Short Publications

13. Michael Carbin, Deokhwan Kim, Sasa Misailovic, and Martin Rinard. Verified integrity properties for safe approximate program transformations. Workshop on Partial Evaluation and Program Manipulation (PEPM), 2013.

14. Michael Carbin and Martin Rinard. (Relaxed) safety properties for relaxed approximate programs. Workshop on Relaxing Synchronization for Mulitcore and Manycore scalability (RACES), 2012.

15. Vijay Ganesh, Michael Carbin, and Martin Rinard. Cryptographic path hardening: Hiding vulnerabilities in software using cryptography. Off-the-Beaten Track (OBT), co-located with Principles of Programming Languages (POPL), 2012.

Talks

Reasoning about Approximate Computing.

Microsoft Research Redmond, Invited Seminar	May 2014
University of Washington, Invited Seminar	May 2014
Cornell, Invited Seminar	May 2014
University of Illinois ECE, Invited Seminar	Apr. 2014
University of Michigan, Invited Seminar	Apr. 2014
Georgia Tech, Invited Seminar	Apr. 2014
MIT, Invited Seminar	Apr. 2014
Columbia, Invited Seminar	Mar. 2014
University of Pennsylvania, Invited Seminar	Mar. 2014
Princeton, Invited Seminar	Mar. 2014
UT Austin ECE, Invited Seminar	Mar. 2014
UCSD, Invited Seminar	Mar. 2014
UC Berkeley, Invited Seminar	Feb. 2014
Wisconsin, Invited Seminar	Feb. 2014
University of Chicago, Invited Seminar	Feb. 2014

Verifying Quantitative Reliability for Unreliably Executed Programs.

National Science Foundation Variability Expedition, UC Irvine, Irvine CA	Nov. 2013
Object-Oriented Programming, Systems, Languages, and Analysis (OOPSLA)	Oct. 2013

Reasoning about Relaxed Programs.

UC Berkeley OSQ	Aug. 2013
POPL Student Session **(Best Talk Award)**	Jan. 2013
SRI, Palo Alto, CA	Dec. 2012
Programming Languages Design and Implementation (PLDI)	Jun. 2012

Verified Integrity Properties for Safe Approximate Program Transformations.

Partial Evaluation and Program Manipulation (PEPM)	Jan. 2013

Bolt: Detecting and Escaping Infinite Loops in Unmodified Binaries.

Object-Oriented Programming, Systems, Languages, and Analysis (OOPSLA)	Oct. 2012

(Relaxed) Safety Properties for Relaxed Approximate Programs.

Relaxed Synchronization for Multicore and Manycore scalability (RACES)	Oct. 2012

Detecting and Escaping Infinite Loops with Jolt.

European Conference on Object-Oriented Programming (ECOOP)	Jul. 2011

Automatically Identifying Critical Input and Code Regions in Applications.

International Symposium of Software Testing and Analysis (ISSTA)	Jul. 2010
École Polytechnique Fédérale de Lausanne (EPFL)	Jul. 2010

Industry Experience

moka5.

Software Developer

Client-Host Shared Memory and Graphics Acceleration for the VMWare Hypervisor (Summer 2006)
Design and implementation of a client-host shared memory system for the VMWare Hypervisor that exploited the (undocumented) shared memory nature of VMware's DMA-based I/O. Design and implementation of an in-client DirectX accelerator for executing graphics commands on a host-based DirectX overlay using shared memory.

Stanford Academic Computing.

Software Developer

Web Development and Back-End Server Programming (2003–2006)
Implementation and maintenance of online trouble ticket systems for residential computing community. Implementation of content management system for dorm website assets.

3lbs.

Software Developer

Educational Game Development and Back-End Server Programming (2000–2006)
Implementation of Flash web games and back-end CGI support for children's educational website.

Patents

Configuration of Isolated Extensions and Device Drivers.
Galen C. Hunt, James R. Larus, Manuel A. Fähndrich, Orion Hodson, David R. Tarditi, Michael Spear, Michael Carbin, Steven P. Levi, Bjarne Steensgaard.
U.S. Patent Number 8,074,231. Filed: June 30, 2006. Issued: December 6, 2011

Bypass Virtualization.
Thomas Joseph Purtell, Won Chun, Michael Carbin
U.S. Patent Number 8,065,687. Filed: Jan 7, 2008. Issued November 22, 2011

References

Martin Rinard
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
rinard@csail.mit.edu
617-258-6922

James Larus
School of Computer and Communication Sciences
École Polytechnique Fédérale de Lausanne
james.larus@epfl.ch
+41 21 69 31403

Saman Amarasinghe
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
saman@mit.edu
617-253-8879

Srini Devadas
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
devadas@mit.edu
617-253-0454