

# Ankush Pankaj Desai

337 Cory Hall, Berkeley, CA 94720  
ankush@eecs.berkeley.edu | ankushdesai . com  
+1-408-823-1269

---

## RESEARCH INTERESTS

I am mainly interested in ensuring *correctness*, *reliability*, and *security* of software systems. During my Ph.D., I have designed new techniques and tools that had an impact both in industry and academia, and are used for building reliable systems across domains like *device drivers*, *distributed systems*, *robotics*, and *cyber-physical systems*. My research draws from, combines, and contributes to the areas of *programming languages* and *systematic testing*, *runtime verification*, and *security*.

---

## EDUCATION

**University of California, Berkeley**  
Ph.D. candidate in Electrical Engineering and Computer Sciences 2013 - present  
**Thesis:** Modular and Safe Asynchronous Event-Driven Programming  
**Advisors:** Sanjit Seshia and Shaz Qadeer

**Indian Institute of Technology, Kanpur**  
Master of Technology in Computer Science and Engineering 2010  
**Thesis:** Fault-Tolerant On-Board Computers for Nanosatellite.  
**Advisor:** Arnab Bhattacharya

---

## CONFERENCE PUBLICATIONS (GOOGLE SCHOLAR)

- [\[c12\]](#) [A Runtime Assurance Framework for Programming Safe Robotics Systems.](#)  
*Ankush Desai, Shromona Ghosh, Sanjit Seshia, Natarajan Shankar, Ashish Tiwari.*  
International Conference on Dependable Systems and Networks (**DSN**) - 2019
- [\[c11\]](#) [Compositional Programming and Testing of Dynamic Distributed Systems.](#)  
*Ankush Desai, Amar Phanishayee, Shaz Qadeer, Sanjit Seshia.*  
Object-Oriented Programming, Systems, Languages, and Applications (**OOPSLA**) - 2018
- [\[c10\]](#) [DRONA: A Framework for Safe Distributed Mobile Robotics.](#)  
*Ankush Desai, Indranil Saha, Jianqiao Yang, Shaz Qadeer, Sanjit Seshia.*  
International Conference on Cyber-Physical Systems (**ICCPs**) - 2017
- [\[c9\]](#) [Combining Model Checking and Runtime Verification for Safe Robotics.](#)  
*Ankush Desai, Tommaso Dreossi, Sanjit Seshia.*  
Runtime Verification (**RV**) - 2017
- [\[c8\]](#) [Lasso detection using Partial State Caching.](#)  
*Rashmi Mudduluru, Pantazis Deligiannis, Ankush Desai, Akash Lal, Shaz Qadeer.*  
Formal Methods in Computer-Aided Design (**FMCAD**) - 2017
- [\[c7\]](#) [Systematic Testing of Asynchronous Reactive Systems.](#)  
*Ankush Desai, Shaz Qadeer, Sanjit Seshia.*  
Foundations of Software Engineering (**FSE**) - 2015
- [\[c6\]](#) [Approximate Synchrony: An Abstraction for Distributed Almost-synchronous Systems.](#)  
*Ankush Desai, Sanjit Seshia, Shaz Qadeer, David Broman, John Eidson.*  
Computer Aided Verification (**CAV**) - 2015
- [\[c5\]](#) [Natural Proofs for Asynchronous Programs using Almost-synchronous Invariants.](#)  
*Ankush Desai, Pranav Garg, P. Madhusudan.*  
Object-Oriented Programming, Systems, Languages, and Applications (**OOPSLA**) - 2014
- [\[c4\]](#) [Endlessly Circulating Messages in IEEE 1588-2008 Systems.](#)

David Broman, P. Derler, [Ankush Desai](#), John Eidson, Sanjit Seshia.  
International Symposium on Precision Clock Synchronization for Measurement, Control, Communication (ISPCS) - 2014

[c3] P: Safe Asynchronous Event-Driven Programming.  
[Ankush Desai](#), Vivek Gupta, Ethan Jackson, Shaz Qadeer, Sriram Rajamani, Damien Zufferey.  
Programming Language Design and Implementation (PLDI) - 2013

[c2] Depth Bounded Explicit-State Model Checking.  
Abhishek Udupa, [Ankush Desai](#), Sriram Rajamani.  
International SPIN Symposium on Model Checking of Software (SPIN) - 2011

[c1] Jugnu: Student Nanosatellite from Indian Institute of Technology, Kanpur.  
*Jugnu's Team*.  
Indian Small Satellite Systems Conference (ISSSC) at Indian Space Research Organisation (ISRO) Satellite Center, India - 2010

---

IN SUBMISSION  
OR PREPARATION

[u2] Systematic Testing of Actor Programs using Value Summaries.  
[Ankush Desai](#), Jianqiao Yang, Koushik Sen, Sanjit Seshia.  
(in preparation for CAV) - 2019

[u1] Kavach: Enforcing Privacy by Attaching Policies to Data.  
[Ankush Desai](#), Pramod Subramanyan, Sriram Rajamani, Sanjit Seshia.  
(in preparation for CCS) - 2019

---

INVITED &  
WORKSHOP  
PAPERS

[i3] Programming Safe Robotics Systems: Challenges and Advances.  
[Ankush Desai](#), Shaz Qadeer, Sanjit Seshia.  
International Symposium On Leveraging Applications of Formal Methods, Verification and Validation (ISoLA) (**invited**) - 2018

[i2] Formal Specification for Deep Neural Networks.  
S. A. Seshia, [Ankush Desai](#), T. Dreossi, D. J. Fremont, S. Ghosh, E. Kim, S. Shivakumar, M. Vazquez-Chanlatte, X. Yue.  
Automated Technology for Verification and Analysis (ATVA) (**invited**) - 2018

[w1] A New Reduction for Event-Driven Distributed Programs.  
[Ankush Desai](#), Pranav Garg, P. Madhusudan.  
International Workshop on Exploiting Concurrency Efficiently and Correctly (EC2) - 2014

---

TUTORIALS

[t1] P: Modular and Safe Asynchronous Programming  
International Conference on Runtime Verification (RV) - 2017

[t2] Safe Asynchronous Programming: Methodology, Language, and Tools  
SIGPLAN conference on Programming Language Design and Implementation (PLDI) - 2019

---

TECHNICAL  
REPORTS

[tr2] Iterative Cycle Detection via Delaying Explorers.  
[Ankush Desai](#), Shaz Qadeer, Sriram Rajamani, Sanjit Seshia.  
Microsoft Research Technical Report (MSR-TR-2015-28) - 2015

[tr1] Critical Path based Performance Models for Distributed Queries.  
[Ankush Desai](#), Kaushik Rajan, Kapil Vaswani.  
Microsoft Research Technical Report (MSR-TR-2012-121) - 2012

---

## AWARDS & ACHIEVEMENTS

- *Sevin Rosen Funds Departmental Award for Innovation*, 2017-18.
  - *Finalist* for the Microsoft Research Ph.D. Fellowship, 2016.
  - Head of the on-board-computers team that built India's first Nanosatellite **JUGNU**, 2009-11.
  - *Sir Ratan Tata Trust Merit Scholarship* for excellent academic performance, 2004-05.
  - *Sir J.R.D Tata Trust Merit Scholarship* for excellent academic performance, 2003-04.
- 

## RESEARCH DESCRIPTION

### **P Language**

<https://github.com/p-org/P>

*Modular and Safe Asynchronous Event-Driven Programming*

P is a domain specific language for writing safe event-driven asynchronous programs [c3]. P program is a collection of state-machines interacting with each other by exchanging messages. The P framework unifies modeling, verification, and programming into one activity. Not only can a P program be compiled into executable code, but it can also be validated using state-of-the-art systematic testing techniques like search prioritization [c7] [c2], natural proofs [c5], abstractions [c6], and symbolic execution [u2]. P also implements a module system that supports compositional programming and analysis of distributed systems (actor programs). We are the first to build a practical programming framework for distributed systems that allows decomposing system-level monolithic testing problem into a collection of component-level testing problems using sound principles of compositional refinement [c11].

P was first used to implement and verify USB 3.0 driver stack that shipped with Windows 8 (300+ bugs found and fixed). It has since been used for development of various other device drivers inside Microsoft: Windows Phone, HoloLens, and Bluetooth. Feedback - *"After a year of heavy daily usage across the entire team there have been no bugs in the layers that were modeled and verified"*

**P#**, a .Net runtime implementing P is used by engineers in Azure to design, implement and thoroughly test distributed systems and services. It uses prioritized portfolio search using delaying explorers inspired by [c7] for scalable testing.

To demonstrate the efficacy of P, *we implemented and verified the IEEE 1588 distributed protocol*. We used a novel abstraction of "approximate-synchrony" [c6] to verify the protocol and also reproduced a long-debated bug which was well received by the IEEE 1588 standards community [c4].

### **Drona Framework**

<https://drona-org.github.io/Drona/>

*Programming Safe Robotics Systems*

We built Drona [c10], a framework for programming safe robotic systems. The Drona framework consists of two critical components: (1) A state machine-based programming language (P language) for specifying, implementing, testing, and verifying the concurrent, event-driven robotics software; (2) A run-time assurance system to ensure the safety of the system in the presence of untrusted components (e.g., off-the-shelf components and complex machine-learning techniques) [c12]. Combining the design-time analysis with run-time safety enforcement helps us bridge the gap between software testing and the actual execution of the software on a robotic platform in the physical world [c9]. The generated code from Drona can be deployed on ROS and other popular robotics platforms.

As a part of this project, we designed a generic & reliable software stack for mobile robots using the high-level modeling language P and used model-checking for systematically testing the entire stack. We also built a novel and *provably correct* decentralized asynchronous motion planner that can perform on-the-fly collision-free planning for multiple robots and also take into account the fact that distributed robots may have clocks that are only synchronized up to a tolerance, i.e., they are almost synchronous.

Drona has been used for safe programming of autonomous (multiple) drones at UC, Berkeley and at UPenn. in collaboration with Prof. George Pappas group. We also did several DARPA demo's as part of the TerraSwarm project. Demo videos: <https://drona-org.github.io/Drona/>

## ZING

*An explicit-state model checker*

<https://github.com/ZingModelChecker/Zing>

ZING is a state-of-art explicit state model checker that was used extensively for systematic testing of P programs. We implemented an efficient search prioritization technique based on delay-bounding for testing asynchronous reactive programs [c7]. The search algorithm is completely parallel [c2] and supports efficient iterative algorithms for finding both safety and liveness bugs [tr2]. Microsoft Product Groups used ZING for verification of device drivers and distributed systems.

Steven Sinofsky acknowledged the usefulness of ZING in finding bugs in Microsoft Products.  
<http://blogs.msdn.com/b/b8/archive/2011/08/22/building-robust-usb-3-0-support.aspx>

## PerfOrator

*Performance prediction for distributed queries*

One of the biggest challenges in doing resource optimization is to understand the impact of a change in resource allocation on the performance of a BigData job. In this project, we proposed a performance modeling approach for predicting the execution time of distributed queries. Our modeling approach is based on the combination of critical path method (analytical modeling), empirically generated black box models and cardinality estimation techniques used in databases. PerfOrator [tr1] helps in predicting and debugging performance of DryadLINQ queries submitted on a cluster. Performance parameters like job execution time, load imbalance and deviation because of outlier machines in the cluster can be predicted with high accuracy (less than 10%) using PerfOrator.

This project led the foundation for the analytical techniques used by PerfOrator tool developed at Microsoft.

## Jugnu Nanosatellite

*India's First Nano-satellite*

<http://www.iitk.ac.in/meold/jugnu/index.htm>

As a part of my Masters Thesis, I designed and implemented fault-tolerant software for onboard computers of the nanosatellite JUGNU. The onboard computer is the heart of a nanosatellite managing task like health monitoring, telemetry data handling, event scheduling, and satellite-payload control. Techniques like pre-emptive scheduling of monitoring task, checkpoint-based recovery, and watchdog timer based recovery were used for tolerance against single point failure.

We also designed and implemented an adaptive communication protocol for small LEO satellites, considering their low power and computational constraints. The protocol implemented features like adaptive baud-rate, adaptive power transmission, and variable packet length to increase the throughput of the data-link protocol. We were able to increase the throughput by 1.8 times using these adaptive schemes.

Finally, for protection against bit-flip faults caused due to radiations in space, we implemented triple modular redundancy (TMR) in program memory (Flash) and FAT16 file-system used for storing data on satellite. Because of this fault tolerance feature, our primary onboard computer was able to function correctly in space even after several reboots.

The proposed scheme for TMR was well appreciated by scientists from Indian Space Research Organization (ISRO) and was implemented first time in any nanosatellite.

Jugnu is **India's first successful nano-satellite** built in collaboration with ISRO. It was launched in Oct. 2011, and the on-board-computers functioned correctly for 6 months sending health and image data to the ground station.

**SRI International, Menlo Park, USA.**

Summer 2017

Research Intern working with Dr. Natarajan Shankar and Dr. Ashish Tiwari

**Microsoft Research, Redmond, USA.**

Summer 2014, 2015, 2016

Research Intern working with Dr. Shaz Qadeer

**Microsoft Research, Bangalore, India.**

Sept. 2010 - May 2013

Research Fellow working with Dr. Sriram Rajamani

**Indian Institute Of Technology, Kanpur, India**

Sept 2008 - June 2010

Graduate Student Researcher working with Prof. Arnab Bhattacharya

---

TEACHING &  
MENTORING

**Teaching Assistant.**

- CS164: *Design and Implementation of Programming Languages* at UC, Berkeley (Spring 2016 and Spring 2017).
- CS330: *Operating Systems* at IIT, Kanpur (Autumn 2009).
- ESC101: *Fundamentals of Computing* at IIT, Kanpur (Autumn 2008 - 2009).

**Guest Lecture.**

- *Compositional Programming and Testing of Distributed Systems*  
UC Santa Cruz in Fall 2018, CMPS290S: *Languages and Abstractions for Distributed Programming*.
- *Runtime Assurance for Safe Robotics*  
UC Berkeley in Spring 2018, CS219C: *Formal Methods: Specification, Verification, and Synthesis*.

**Undergraduate Mentoring.**

- Cambridge Yang (UC Berkeley): I mentored Cambridge when working on a research project for designing and implementing symbolic execution engine for P (2016-2017). He is now a Ph.D. student at MIT.
- Sumukh Shivakumar (UC Berkeley): I am mentoring Sumukh on a research project for programming and testing machine learning components in robotics systems (2017-present). He will be joining UC Berkeley as a Masters Student.
- I was the head of the onboard computer's team (mentoring 8 undergraduate students) that launched India's first nanosatellite JUGNU.

---

SERVICES

**Program Committee Member**

I was a PC member in the Artifact Evaluation Committee for PPOPP 2016, CAV 2017, PLDI 2018, ECOOP 2018.

**External Expert Reviewer:** POPL 2019

**Sub-reviewer** for papers in conferences: CAV 2015, MODELS 2015, HSCC 2015, POPL 2016, CAV 2017, HVC 2017, ICRA 2018.

---

INVITED  
TALKS

*Modular and Safe Asynchronous Event-Driven Programming*

- CMU, MIT, Harvard, UCSC in Nov. 2018
- AGERE! Workshop on Programming Based on Actors, Agents, and Decentralized Control, 2018.
- Facebook in Nov. 2018 .

*DRONA: A Framework for Programming Safe Robotics Systems*

- Berkeley Artificial Intelligence Research Lab and the Berkeley DeepDrive in Oct. 2018
- Continental Automotive in Sep. 2018
- GRASP lab, the University of Pennsylvania in March 2017
- Online Seminar, TerraSwarm Research Center, Feb. 2017 .

*Building Reliable Distributed Systems using P*

- Google, Mountain View, CA, Feb. 2014.

- Infrastructure Team, Facebook, Menlo Park, CA, Sept. 2013.

*Approximate Synchrony: An Abstraction for Distributed Almost-synchronous Systems*

- Online Seminar, TerraSwarm Research Center, April, 2015.

---

REFERENCES

**Prof. Sanjit Seshia**

Professor, University of California, Berkeley ([sseshia@eecs.berkeley.edu](mailto:sseshia@eecs.berkeley.edu)).

**Dr. Shaz Qadeer**

Research Scientist, Facebook ([shaz@fb.com](mailto:shaz@fb.com)).

**Prof. Edward A. Lee**

Robert S. Pepper Distinguished Professor, University of California, Berkeley ([eal@eecs.berkeley.edu](mailto:eal@eecs.berkeley.edu)).

**Prof. George Pappas**

Joseph Moore Professor, University Of Pennsylvania ([pappasg@seas.upenn.edu](mailto:pappasg@seas.upenn.edu)).

**Dr. Sriram Rajamani**

Managing Director, Microsoft Research India ([sriram@microsoft.com](mailto:sriram@microsoft.com)).

---