4/17 - Correctness & Security of Hardware

Guest lecture: Caroline Trippel
Work: using Alloy to verify hardware security

Hardware architecture is highly heterogeneous

- Each part interacts with shared resources differently
- We can measure power & performance
- How do we check security & correctness?

The idea:

- Software correctness & security problems can instead be mapped as looking at **problematic hardware event orderings**
    - We can check if these bad orderings are possible
- We can use analysis in Alloy to model this:
    - Happens-before analysis (cycles in a happens-before graph)
    - SAT solvers

Suppose we have a program in C:

- Suppose this program is correct & secure
- When this is compiled to instructions - this *can be executed in many possible ways*!
    - Different execution orders might translate to correctness/security issues in our software
    - We need to check the ways this can execute
        - This is where we tie in Alloy, SAT solvers, etc.
        - Microarchitectural Happens-Before Analysis

How do we model this?

- Microarchitectural Happens-Before Analysis
    - Model it as a directed graph for a particular architecture and program
    - Which events need to happen before other events happen
- We can check if a program is a possible by checking for no cycles in the graph
    - We want to eventually check that all these possible (acyclic) graphs are secure & correct.

We can try to auto-generate exploits:

- Check if it's susceptible to these classes of attacks
- Search for patterns of executions that are characteristic of a particular attack