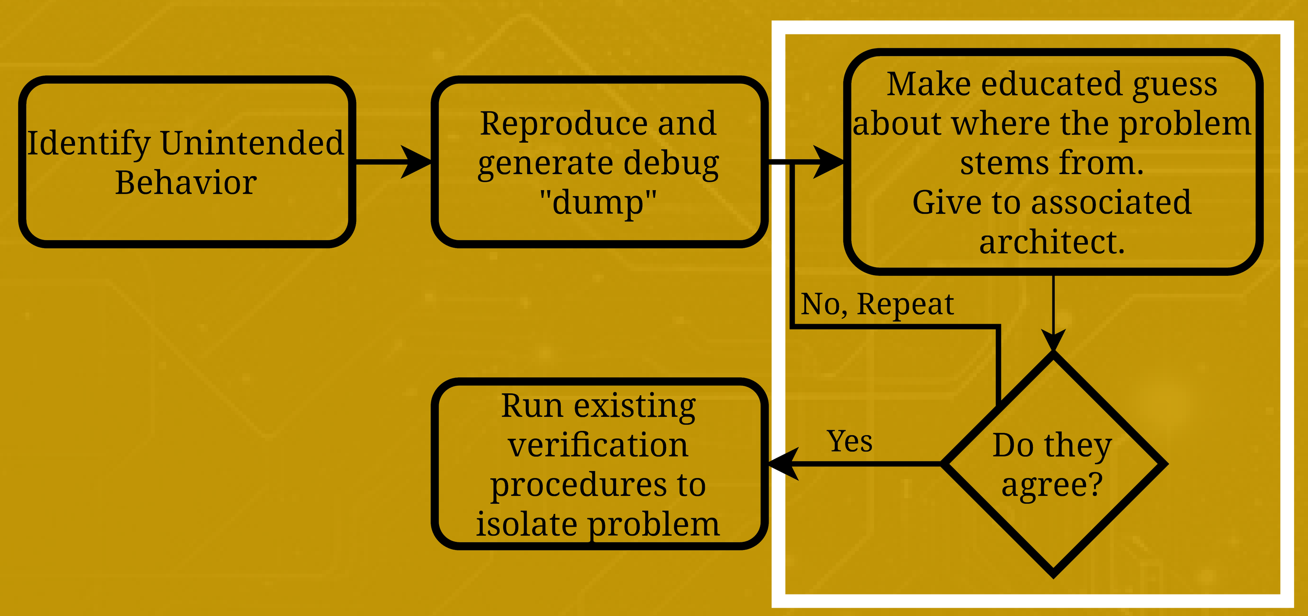
AI-DRIVEN POST-SILICON ROOT CAUSE ANALYSIS FOR HARDWARE DEFECTS

Abstract

When designing electronic hardware, design defects occasionally exist beyond standard verification procedures occurring prior to the design being fabricated and are instead found during testing afterwards.
While it's already difficult to find the root cause of these defects during any stage of the design process, the comparative lack of data available post-fabrication makes the process much more difficult.
Currently, engineers must use the collective knowledge and experience of experts within the group to make the best guess about which component contains the root cause defect, which is then tested. This cycle is repeated for as long as necessary.
Artificial Intelligence (AI) techniques are being explored to reduce the time needed within this cycle by having better inferences on the root cause component which can then be investigated further using traditional verification techniques. Additionally, the time needed for these traditional verification methods can be reduced by increasing the precision of the root cause inferences, which AI may also be able to assist with.

Motivation/Problem



After the first group of chips are produced for a given design, they are ran through many tests to ensure stability and to catch any bugs that were not found prior to production. When a bug is found that causes the system to "hang" or freeze, a "dump" of all of the data within the chip is produced for analysis.
As designs have gotten more complex, this amount of data has grown alongside it.
Currently, only the most skilled engineers, leveraging their decades of experience, have the ability to even make an educated guess about the root cause of the problem. This is passed around between many teams to finally find the bug with existing tools such as Formal Verification, which, in aggregate, can take a significant amount of time during which the product cannot be launched, therefore losing the company a lot of money and taking the time of engineers that would be better spent architecting the next design.

Hypothesis & Methods

Currently, it's known that AI neural networks are good at pattern recognition. The current hypothesis claims that with enough data, this ability can be leveraged to help a human engineer more quickly find the root cause of a known problem. This will be done by, after training the neural network with a large amount of data from the design in question, along with some aspects of the design, providing it with full data "dumps" for which it will attempt to narrow down the root cause area.

Dataset

The dataset will consist of prior solved problems, known good dumps, along with information about the layout of the given design.

Tools Used

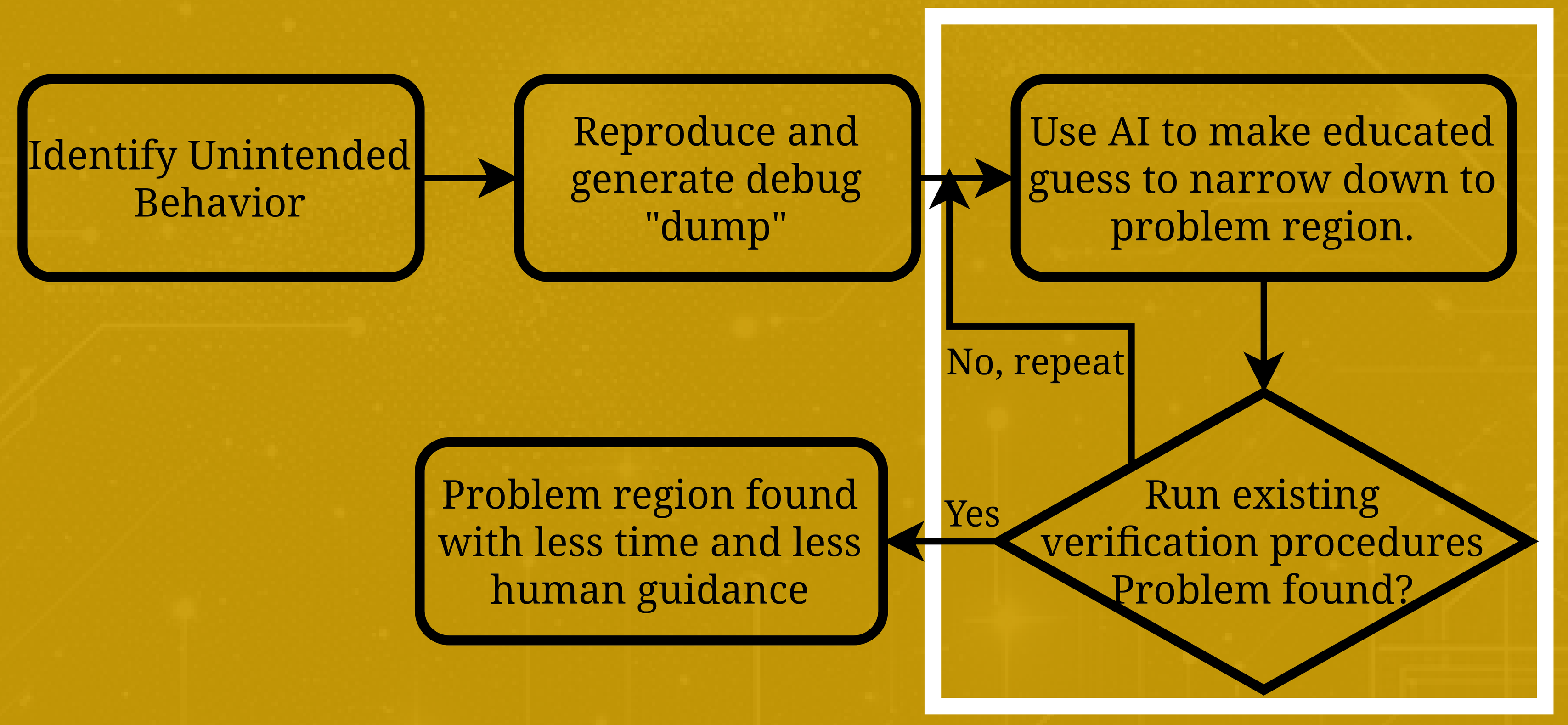
Neural networks will be trained with the data such that the model can narrow down the scope of the root cause for a given issue.

Evaluation

Given a problem that is solved without this tool, it will be evaluated for correctness in root cause determination. These results will also be used to further train the model.

Result Recording

A program or script will be made to record the accuracy of the model as it continues to be trained. A positive result would consist of a positive trend to a graph containing this information.



Electrical and Computer Engineering
UCF UNIVERSITY OF CENTRAL FLORIDA
DRACO

Acknowledgments

Thank you to the other members of the DRACO Lab for help creating this poster and assisting with certain portions of this project.

Discussion

In general, it appears that the data gathered from the "dump," when combined with information about the overall layout of the design, is structured in such a way that we can train an AI model to help the engineers find the problem areas faster.

This is because AI is best at detecting patterns in data, such that it can discern some of the best places to look within the design with enough data. We hope to leverage this functionality to get ever better results from this system

Future Work

Additional work will mostly be focused on increasing the amount of data, both of hung and normally operating systems, that the AI model is trained on. This is done to assess the efficacy of the designed implementation.

Being able to train the model on several different designs so that it can be more generalized would be ideal as a new model wouldn't need to be trained for each substantially different design, which would slow down the process of using the new tool. The ideal would be to make it as abstract as possible.

MICHAEL CASTIGLIA
MICHAELC@UCF.EDU

DRACO LAB | www.ece.ucf.edu/DRACO
Dr. Mike Borowczak, Lab Director

Dept. Of Electrical & Computer Engineering
College of Engineering and Computer Science
University of Central Florida