CS486C – Senior Capstone Design in Computer Science

Project Description

Project Title: AI-ACETONE (AI-Automated CounterExample Test Objectives by Negating Expectations)	
	**Chris Ortiz**, Senior Technologist Tools & Infrastructure SanDisk Corp., Engineering & Product Management chris.ortiz@sandisk.com **Rex Jackson**, Vice President Tools & Infrastructure SanDisk Corp., Engineering & Product Management rex.jackson@sandisk.com

Project Overview:

At SanDisk, we test and validate our popular storage products to mark our seal of approval on the quality we uphold. In directed testing, we focus on specific, targeted features to verify whether they behave according to the design. We are checking whether the Product Team has implemented the intended functionality and whether it meets certain requirements—referred to as *properties* in TLA+ (Temporal Logic of Actions).

TLA+ is a mathematical language used for formally specifying system designs. It comes with tools such as TLC, the TLA+ model checker, which exhaustively explores the state space of a design.

We recognize that our Test Engineers cannot anticipate all possible combinations of test sequences. However, we aim to leverage the exhaustive exploration capabilities of TLC to verify that a TLA+ specification satisfies its defined properties. If a property is violated, TLC provides an error trace from the initial state, showing the shortest steps that lead to the violating state. This trace is known as a *counter-example*.

A TLA+ specification should only proceed in our development process if it passes TLC model checking. Otherwise, it would be logically unsound to develop a design that we know fails to meet its requirements. However, due to potential interpretation gaps between conceptual design and the development of the physical product, we must formulate *Test Objectives* during the design phase. These objectives provide a proactive set of ideas of test sequences to apply once the product is developed.

Formulating Test Objectives requires a deep understanding of the properties and how the product design satisfies them. This process is intricate and prone to gaps—known as *test holes*—which result from inadequate testing. Since a TLA+ specification captures all possible behaviors of the product design in relation to its requirement properties, we can systematically **negate each expected** property. TLC will then generate **counterexamples**, which effectively become our **Test Objectives**.

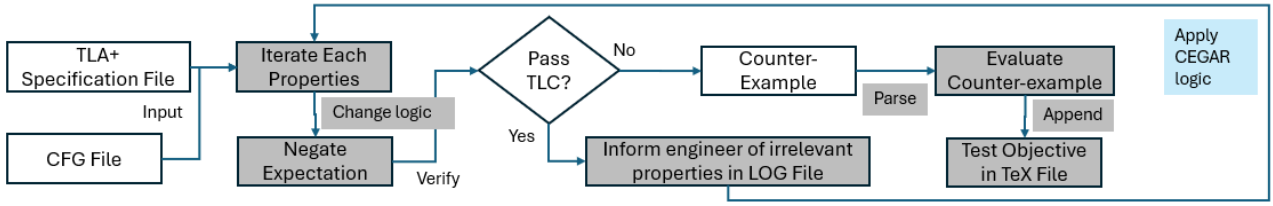
The proposal is to **automate** this process through Agentic **AI** in VSCode IDE, which would offer the following benefits:

1. May provide test sequences that cover **intricate corner cases that a Test Engineer might not have anticipated** or included as directed tests in the Test Objectives.
2. Provides an **automated list of Test Objectives** that are portable to SanDisk Design Document template, which is highly convenient for Test Engineers.

3. Maybe integrated into our future generative and **automated testing** that leverage the same TLA+ Specification source.

The figure below illustrates our proposed solution. The grayed boxes and arrow labels indicate the areas where we need NAU's assistance. Any light blue elements represent stretch goals, if applicable. This project will give NAU students exposure to PlusCal and TLA+, both invented by Leslie Lamport, an ACM Turing Award recipient whose contributions to computer science are widely used in universities and by major companies such as AWS, Microsoft, Oracle, Google, LinkedIn, MongoDB, Datadog, Intel, ARM, and Nvidia. Additionally, projects like this may be eligible for grants from the TLA+ Foundation if submitted and selected by the TLA+ committee.

A stretch goal if the students can add CEGAR (Counter-Example Guided Abstraction Refinement) in the methodology, like in this paper: https://www.sandia.gov/app/uploads/sites/222/2023/03/mayo_ftscs15_nominal.pdf



Knowledge, skills, and expertise required for this project:

- Knowledge of TLA+ and PlusCal using TLA+ VSCode extension.
  - Home Page: https://lamport.azurewebsites.net/tla/tla.html
  - Github: https://github.com/tlaplus
  - Conferences: https://conf.tlapl.us/home/
  - Foundation: https://foundation.tlapl.us/
- Coding skill in Rust (preferred) or Python: https://www.rust-lang.org/

Equipment Requirements:

Computer system with VSCode installed with the following extensions and internet connectivity.

- Rust-analyzer (The Rust Programming Language)
- TLA+ (Temporal Logic of Actions by TLA+ Foundation)
- Graphviz Interactive Preview (tintinweb)
- Live Share (Microsoft)
- Github Copilot with agent mode, Claude Sonnet 4.5 (preferred) or Opus 4.5

Other software:

- OpenJDK >= 11.0.6 (Please do not use Oracle's)
- Rust Installation and crates that might be needed

Software and other Deliverables:

Detailed, clear, and professionally composed documentation and literature which SanDisk product teams will use as reference manual.

Complete documentation and well-commented codebase in **Private** Github account, for which SanDisk can solely decide freely if it is beneficial for SanDisk to open-source this repository (or fork of it) or not in the future.