CS486C – Senior Capstone Design in Computer Science

Project Description

Project Title: P2P2P (PlusCal to PlantUML to PDF)	
	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:

In SanDisk, we are serious about the Quality of our popular storage products, and we always strive to be best-in-class in upholding our global SanDisk brand. SanDisk has launched an initiative to write our designs using TLA+ (Temporal Logic of Actions) to enable formal verification through a model checker. TLA+ is a mathematical language based on set theory and logic, which most of our product team is not yet well-versed in. Currently, we rely on design documents written in English, supplemented with diagrams, tables, and pseudo code, which are reviewed iteratively by many stakeholders. However, we have found that this review process is inadequate for catching subtle bugs that were introduced during design if done without formal verification—hence our move toward adopting formal methods. We are trying to be more proactive by reducing, if not eliminating, those subtle bugs that can cost us significant expense for fixing the problem and may taint our company's reputation against Quality metrics from our customers.

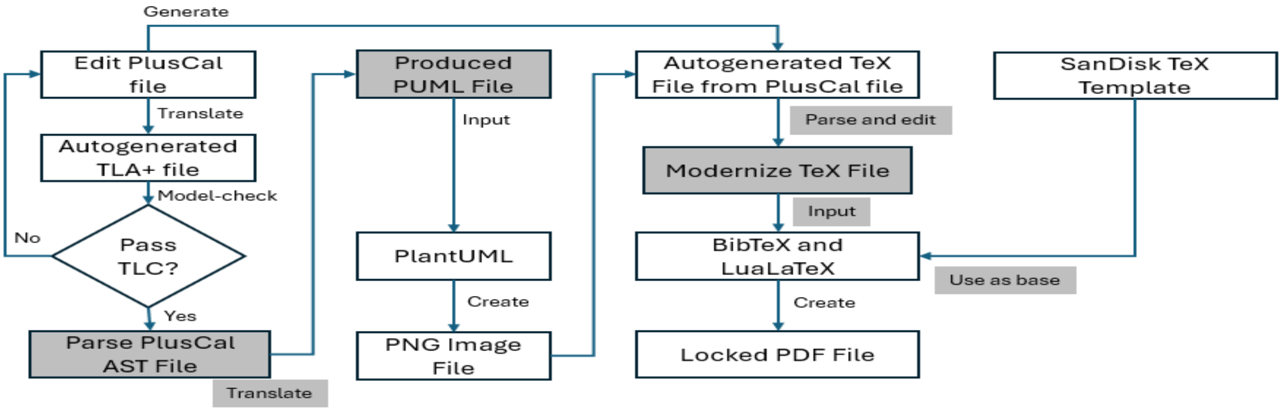
Most of our product teams are part of the Firmware Engineering group, where designs are typically expressed using flowcharts, UML sequence diagrams, and pseudo code. We discovered that **PlusCal**, a pseudo code language that compiles into TLA+, is more approachable for these engineers than writing TLA+ directly. It can also express nondeterminism and concurrency of our systems which our current tools cannot check. The TLA+ toolset can automatically translate PlusCal into TLA+, which can then be formally verified by the model checker.

However, we also found that design artifacts are often passed between teams, some of which are not firmware engineers and are unfamiliar with reading pseudo code. These teams are more comfortable with visual formats like flowcharts and UML sequence diagrams generated using **PlantUML**. Since most people are visual learners, we aim to automate the generation of flowcharts, UML sequence diagrams, and UML activity diagrams from PlusCal pseudo code. Synchronization of the PlusCal and PlantUML context is very important, that is why at the last step we intend to produce a locked **PDF** which can only happen when the PlusCal has passed the model-checker and the corresponding PlantUML figure associated with it is the precise representation to be confidently shared to the stakeholders.

This breakthrough will benefit us in three important ways:

1. **It will significantly reduce the time required to create visual figures,** which will be formally verified for correctness, giving our stakeholders greater confidence when reviewing our Design Specification documents.
2. **It will encourage firmware engineers** to adopt PlusCal as a stepping-stone to TLA+.
3. **The convenience of automated diagram generation** will help foster a culture of formal methods adoption within SanDisk's design process.

The figure below is our proposed solution where the grayed boxes and arrow's label are the ones we need NAU to help. This project will expose the NAU students to be familiar with LaTeX (Lamport TeX), PlusCal and TLA+ which Leslie Lamport invented. He is an ACM Turing Awardee where his contribution to Computer Science is use in universities, and by big companies like AWS, Microsoft, Oracle, Google, LinkedIn, MongoDB, Datadog, Intel, ARM, Nvidia, etc. Also, there is a potential that projects such as this can get grant from TLA+ Foundation if submitted and the committee selected it.



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/
- Generation of PlantUML Sequence and Activity Diagrams: https://plantuml.com/
- LaTeX and PDF generation using Lualatex. https://www.latex-project.org/
- 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.

- LaTeX Workshop (James Yu)
- Rust-analyzer (The Rust Programming Language)
- TLA+ (Temporal Logic of Actions by TLA+ Foundation)
- Graphviz Interactive Preview (tintinweb)
- Live Share (Microsoft)
- PDF Viewer (Mathematic Inc)
- PlantUML (jebbs)

Other software:

- OpenJDK >= 11.0.6 (Please do not use Oracle's)
- MiKTeX >= 25.3 (with bibtex and lualatex)
- Adobe Acrobat Reader
- 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 and well-commented codebase via ZIP file format.

Github private account for sharing development codes.