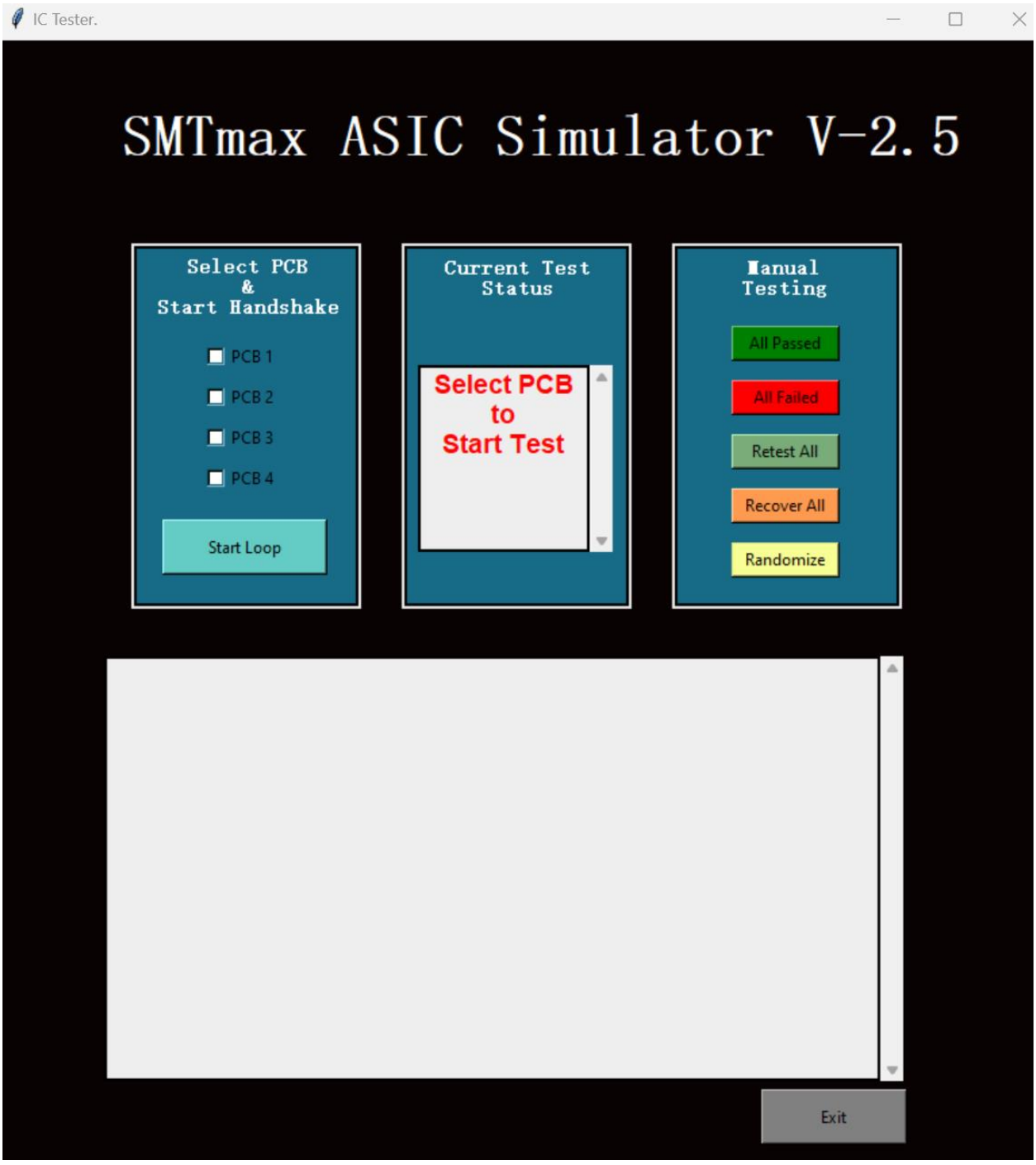
SMTmax ASIC Simulator V-2.5

Made by Mason Hernandez



Introduction:

This project, developed using Python and the Tkinter library, simulates the ASIC testing process, utilizing GPIO pins on Raspberry Pi and connected hardware components. The software part of this project enables users to select PCBs and view test results through a user-friendly GUI, while the hardware part consists of connected components as visualized in the included Fritzing project image.

Prerequisites:

- Python 3.x
- Tkinter
- RPi.GPIO
- Fritzing Image (For viewing the hardware schematic)

Installation:

1. Install Python and pip if not already installed.
2. Install the necessary Python libraries:

Usage

1. Run the main Python script: python main.py
2. The GUI will display allowing users to select PCBs and initiate tests.
3. The results of the tests will be displayed on the GUI.
4. Users can view the hardware connections and setup by opening the Fritzing image.

Features:

- GUI Interface: Enables users to interact with the system and view test results.

- Hardware Interaction: Communicates with hardware components through GPIO pins to conduct tests.

- Test Modes: Supports various test modes including All Passed, All Failed, Retest All, Recover All, and Randomize.

Hardware Schematic:

To understand the hardware setup and connections, refer to the provided Fritzing image.

Key Functions

- `signal_start()`: Initiates the testing process based on selected PCBs.

- `automate_handshake()`: Automates the testing process and displays random test results.

- `start_all_good()`, `start_all_bad()`, `start_all_retest()`, `start_all_recover()`: Conducts tests in various modes and displays the selected results.

Conclusion

This project provides an integrated approach to simulate ASIC testing, combining software logic with hardware components to emulate real-world scenarios.

License: