

Comprehensive Project Documentation: Open-Source Radiation Hardening Simulator

Jacob Anderson, David Nichols, Collin Lambert, Parker Allred

August 6, 2024

Contents

1 Project Overview

This project aims to develop an open-source radiation hardening simulator using XSchem and NGSpice. The simulator provides a comprehensive library for simulating the effects of radiation on electronic circuits, including modules for fault injection and radiation effect simulation. The project is designed to be user-friendly and accessible to researchers and engineers working in the field of radiation hardening.

2 Library Structure and Core Features

2.1 Single Event Effect Simulation

Single event effects are radiation effects induced by a single radiation strike event, such as single event transients and single event upsets. The following three modules are used to simulate single event transients and, by extrapolation, single event upsets.

To use any of the following modules, if simulating a radiation strike on an NMOS, place the module such that the input of the current source is connected to the drain node, and the output of the current source is connected to the body node.

For a PMOS, place the module such that the input of the current source is connected to the body node, and the output of the current source is connected to the drain node.

2.1.1 Double Exponential

The double exponential current source has been a standard method of simulating the effects of a single radiation event for many years. To use, one must specify the rise time, fall time, rise time constant, fall time constant, and total amount of charge to inject.

2.1.2 Dual Double Exponential

Similar to the double exponential current source, the dual double exponential current source utilizes two double exponential current sources added together. This has been shown to be a more accurate representation of the actual photo-currents generated in a radiation strike. All the parameters specified in the double exponential are also present in the dual double exponential, but there are two sets of them that must be specified.

2.1.3 Adaptive Double Exponential

The adaptive double exponential current source addresses some issues that arise from the independent double exponential and dual double exponential current sources. These independent models don't take into account the circuitry surrounding the component on which a radiation strike is being simulated, leading to unrealistic values. The adaptive double exponential current source has circuitry that allows it to take into account the effects that surrounding circuitry create, preventing unrealistic values from being generated.

2.2 Other Radiation Effect Simulation

2.2.1 Total Ionizing Dose

TID, or total ionizing dose, is a radiation effect in which charge can build up in the insulating regions of a MOSFET. This buildup of charge can cause a number of undesirable effects, including changes in the threshold voltage of the MOSFET and increased leakage current. For this project, we model the change in threshold voltage.

TID effects on an NMOS transistor cause the threshold voltage to decrease, making the transistor transition out of the cutoff region more easily, raising the noise floor. For PMOS transistors, the threshold voltage increases, leading to similar noise floor issues.

We've modeled these effects as an inline voltage source that introduces a voltage bias into the signal driving the gate of the transistors. This bias effectively raises or lowers the threshold voltage of each type of transistor, respectively.

2.2.2 Rail Span Collapse

Rail span collapse is characterized by the voltage between the positive and negative voltage rails collapsing due to excess current draw. This phenomenon can cause severe problems in circuits, especially those with sequential logic, such as SRAMs.

We've modeled voltage drop due to excessive current draw as a logistic function. This function outputs a voltage drop for a given current draw input, allowing for accurate simulation of rail span collapse effects.

3 Installation Instructions

To install the simulator, follow these steps:

3.1 macOS

You can either download the repository and run the installation script directly from the GUI, or open the Terminal and run the following commands:

1. Clone the project repository from GitHub:

```
git clone https://github.com/Jacoba1100254352/RAD-HARD.git
cd RAD-HARD
```

2. Run the installation script:

```
./install_script.sh
```

The script will automatically install all necessary dependencies, including XScheme, NGSpice, Tcl, Tk, and BeSpiceWave.app, and set up the environment for you.

3.2 Windows

1. Download and install ActiveTcl:
 - ActiveTcl-8.6.13.0000-MSWin32-x64-47f84d9f.exe
2. Install XScheme:
 - Download and run XScheme-3.4.5-64bit.msi
3. (Optional) Install the waveform viewer app:
 - Download and run analog_flavor_eval_msw_64_24.06.24.00.msi

4 Usage Examples

4.1 Example Circuit: Operational Amplifier

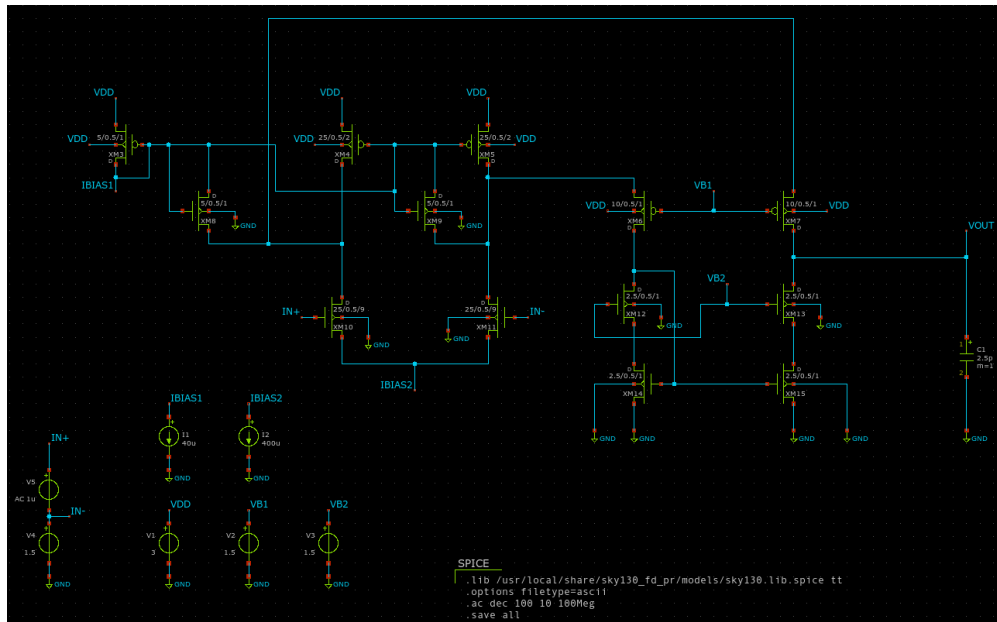


Figure 1: Example Operational Amplifier Circuit

Steps to Simulate:

1. Create the schematic in XScheme, or open an existing circuit by navigating to its location with File >Open. In this case the example circuit can be found in the repository under the Examples directory.
2. Export the netlist as `basic_opamp.spice` by pressing the “netlist” button at the top right. (Note: There will be no pop-up confirmation, but pressing the netlist button will create the .spice netlist file.)

3. Configure XScem to run the simulation:
 - Go to Simulation >Configure simulators and tools.
 - In the “spice” section, select the option that says: `ngspice -b -r "$n.raw"` “\$N”.
 - Press the “simulate” button at the top right of XScem.
4. View waveforms by pressing “waves >external viewer” in the top right of XScem or manually opening the file in BeSpiceWave.

5 Additional Resources

5.1 Documentation and Tutorials

- User Guide: Detailed user guide with step-by-step instructions.
- API Documentation: Comprehensive API documentation for all modules and functions.
- Tutorials: Various tutorials to help users get started with the simulator.

5.2 GitHub Repository

The source code and additional resources for the project can be found on GitHub:

- [GitHub Repository: RAD-HARD](#)

6 Conclusion

This document provides comprehensive documentation for the open-source radiation hardening simulator project. By following the instructions and utilizing the provided resources, users can effectively simulate and analyze the effects of radiation on electronic circuits.

Acknowledgment

We would like to thank Dr. Shiuh-hua Wood Chiang for his guidance and support throughout this project. This work was supported by [Funding Source].