Fabricating a ReRAM Crossbar

Testing

Team 8
Joshua Thater, Aiden Petersen, Matt Ottersen, & Regassa Dukele

Problem Statement

Probem:

The client is interested in analog open source chip fabrication, and testing the limitations of Resistive RAM

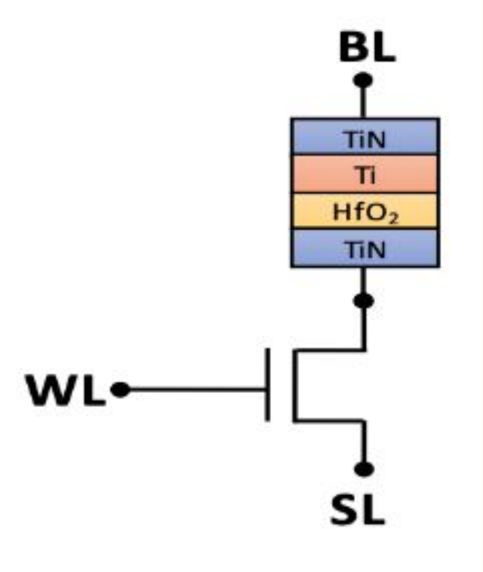
Solution:

Our goal is to use eFabless's MPW shuttle program to submit a ReRAM Crossbar for fabrication. Along the way, we would document our workflow, contributing to ISU's internal knowledge of analog fabrication in the Skywater 130nm process.

What is ReRAM

Resistive RAM is a type of non-volatile memory that does its computations in the analog domain

- Low Latency
- Power Efficient
- Area Efficient



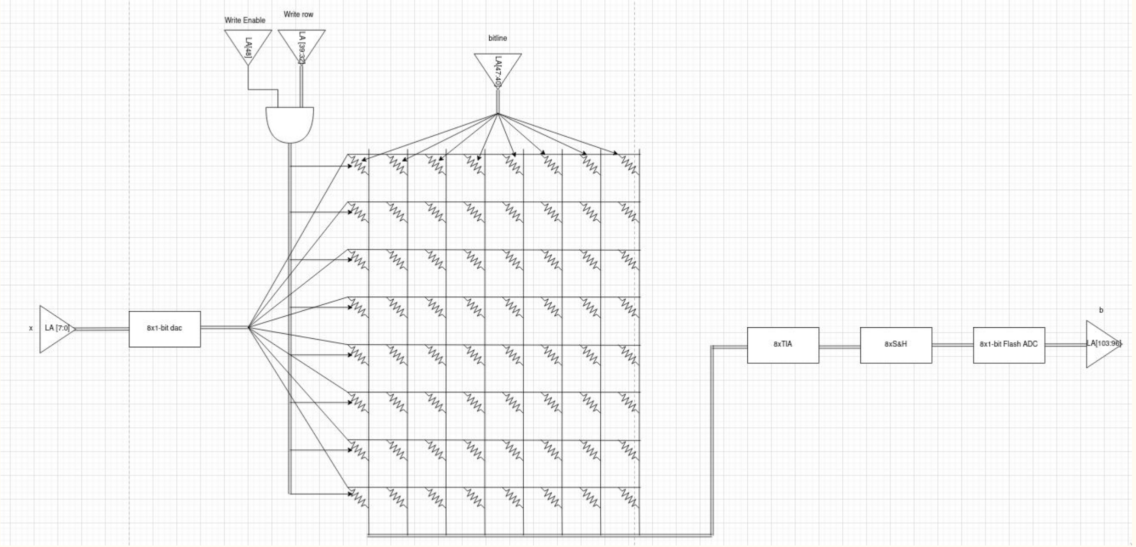
Unit Testing

Main components of the design

- DAC
- ReRam Cell (1T1R)
- Transimpedance Amp
- Sample and Hold
- ADC

Caravel Harness

- Allows us to communicate with the circuit



Unit Testing

Testing Process

- Create a Schematic and Testbench in Xschem
- Simulate the Testbench using Ngspice or Xyce
- Create layout using Magic🪄
- Use Netgen to run LVS
- Extract post-layout parasitics using magic
- Simulate the post-layout netlist using Ngspice or Xyce

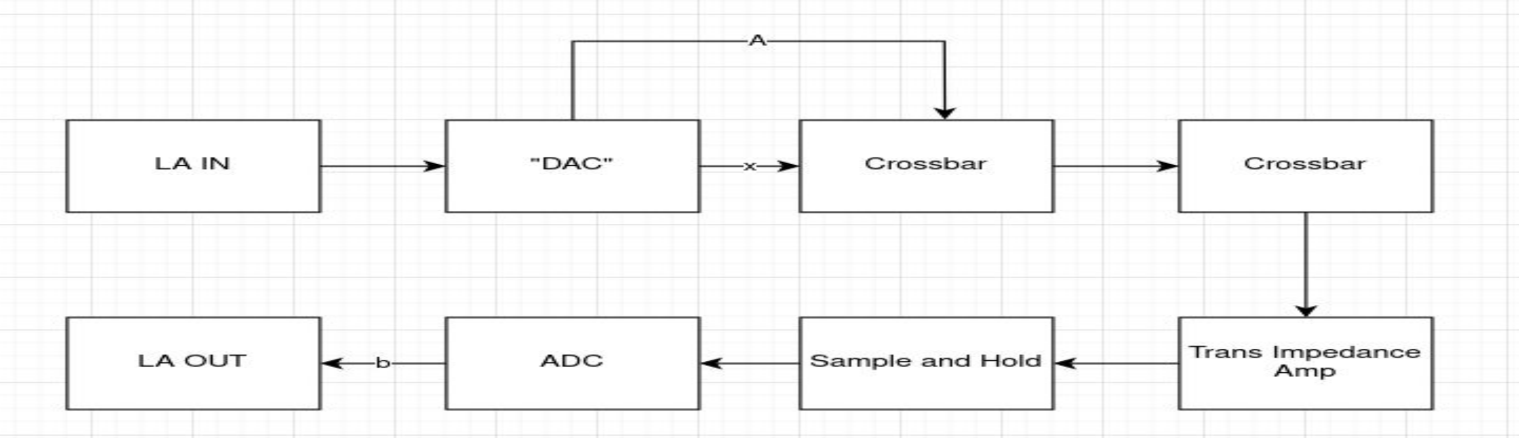
Interface Testing

- Our design consists of multiple units connected through wire interfaces, the user area and RISC-V CPU
- Communication from CPU to user area will be configured through 128-bit logic analyzer
- Each bit can be configured as input or output
- The same testing methodology is used for individual components and the composition of multiple units

Integration Testing

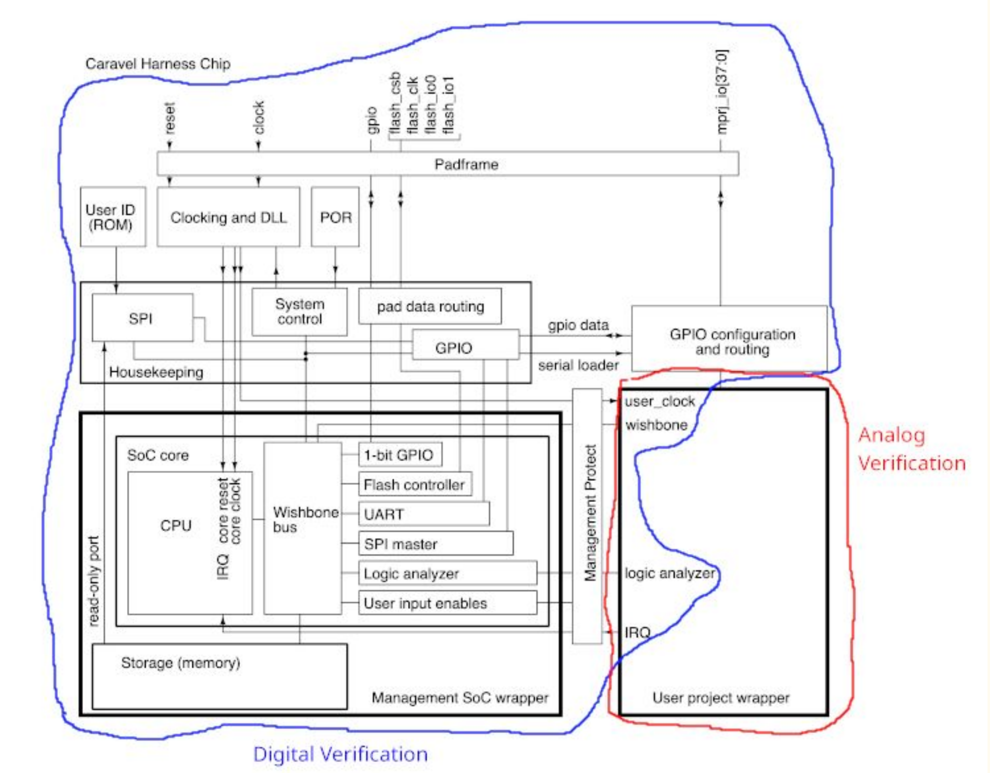
The main integration path in our design is the connection of analog components to the outside Caravel harness

- Create top-level schematic using Xschem
- Simulate its behavior using Ngspice and Xyce
- Simulate and tweak design until satisfied with behavior
- Using Magic, we make the layout and extract post-layout parasitics
- These parasitics will be simulated to ensure that the circuit still meets requirements
- Finally, we will connect our circuit to the outside harness and run further testing to ensure proper behavior



System Testing

- Two parts
  - Digital
    - Create digital behavioural version of ReRAM crossbar that matches functionality
    - Verify software functionality
    - Verify LA/port functionality
  - Analog
    - Verify user area functionality.
  - Hope it works if analog and digital portions both pass tests.
- Precheck

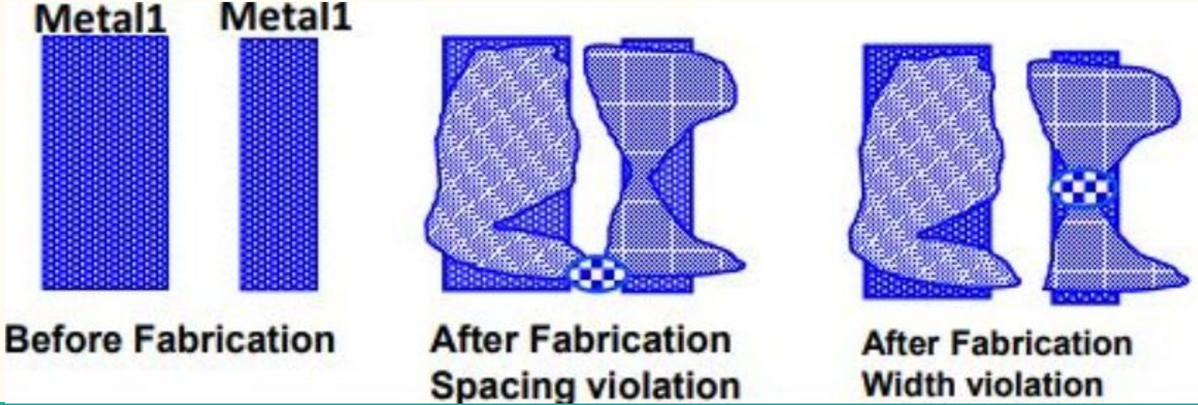


Regression Testing

- Comprehensive testbenches for every component
  - Create analog implementation in xschem
  - Create analog testbench (ngspice simulation)
  - Check testbench passes
  - Create Layout
  - Run LVS
  - Extract post-layout parasitics and rerun simulations
- Same process can be used on components that utilize other components.
- Same as unit testing but will include multiple components

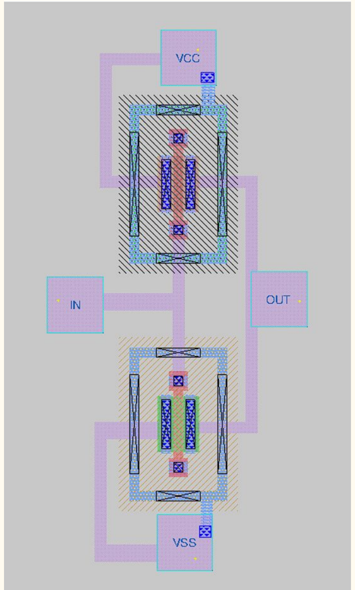
Acceptance Testing

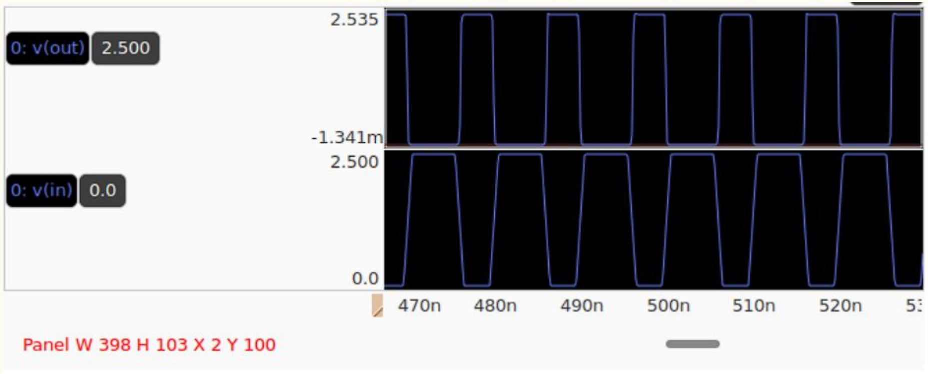
- Non-functionality acceptance will be done through a precheck system that Efabless has
  - Ensures that there are no DRC (design rule check) failures
- Functionality acceptance will be done through system testing and running tests cases on our design with the Caravel harness



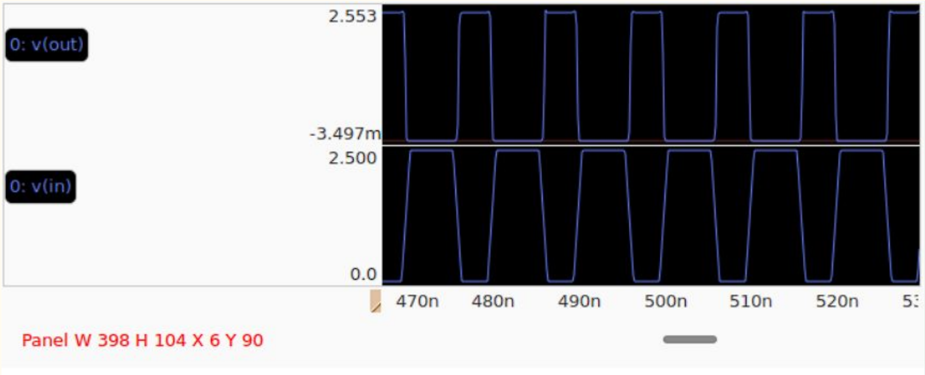
Results

- Do not have a ton of results now…
- Example demonstrating how parasitics can impact device performance





Original Simulation of Inverter (no parasitics)



Simulation of Inverter with parasitics

Questions?