EE/CprE/SE 492 Bi-WEEKLY REPORT 2

Digital ASIC Fabrication

9/11/2023 – 9/27/2023

Group number: sddec23-06

Client & Advisor: Dr. Henry Duwe

Team Members/Role

- Jake Hafele
- Gregory Ling
- Will Galles
- Cade Breedings

Past Week Assignments

- Get experience with open-source layout tool Magic
- Complete wishbone test
- Get Memory module for DSP to harden
- Get started on functional Verilog for DSP module
- Get SPI Interface hardened in user project wrapper
- Investigate different hardening methods with either macros or no macros

Meeting Notes from Previous Week with Dr. Duwe

9/20/23

Updates from us

- Gregory struggling with magic
- Plans on talking to Dr. Maruf
- Design issues are plaguing
- Suggests to talk to sddec-08 team about using magic
- They have gotten a 1 bit dac to work in magic
- Magic and Cadence behave very differently
- Gergory can not place labels in his design and that is his current blocker
- Using Tcl scripts to do everything
- Will got memory module to harden
- Endee up using the ones built into the open pdk
- No longer generating from scratch in open ram
- Brought up switching name of DSP module
- Decided that it wone be the best of ideas
- Changing the name wont remove the questions about the design
- It is a test vehicle that is meant to test xyz
- Big take away is that changing the name is not going to change the questions that we will have to field
- Just have to be ready to explain that the module was only a test bed
- We did not create a novel DSP design and must make that clear
- Cade is trying to wrap up his modules
- Start to integrate all of the different submodules together
- Jake has the whole SPI module done and hardened
- Jake is panning on starting to create test benches for the top level module

Notes

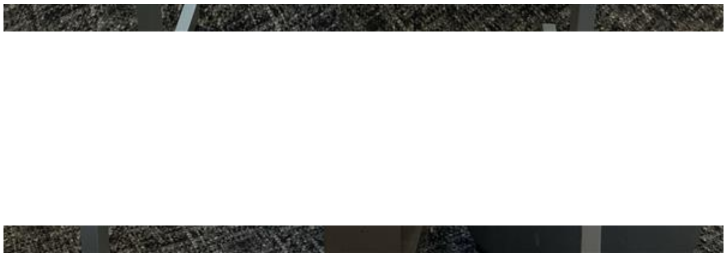
- Will needs to focus on creating the whole DSP module
- Would be great if we could get the whole thing done in a week 😊
-

TODO

- Start looking into the bring up plan
- Get a nucleo board?
- Show that we have the wrapper scripts running

Bi-Weekly Progress

- Gregory
  - Fighting Magic. I succeeded in getting a full CMOS inverter created in Magic using a tcl file.
  - I had significant issues because the colors of two paint layers in Magic had extremely similar coloring and didn't notice I was missing two layers in the design which caused it to fail DRC rules.
  - Potential todo is to create a python script which makes creating the tcl script easier or even writes the Magic file format directly as the .mag format is a simple text-based format.
  - It created an LEF and SPICE netlist so I can attempt simulation next week.
- Jake
  - Got SPI Interface and Module Control Verilog files to Harden together
  - Created hardening configs WITH prehardened macros and as one total macro
  - Hardened user_proj_final, which is under required user_project_wrapper
  - Created bit timing diagrams for SPI read and write transactions to Master SPI from uC
  - Updated documentation for SPI module with new testbenches and waveform results
- Will
  - Get the memory to harden
  - Start working on the sub components of the DSP module
- Cade
  - Completed and wrote test for standard cell module and completed clock gating module



Individual Contributions

Team Member	Contribution	Hours per report	Total Hours
Jake Hafele	Hardened top level design and SPI. Updated SPI documentation	15	90
Gregory Ling	Magic VLSI complete for example inverter.	10	61
Will Galles	Harden Memory and made subcomponents	15	90

	of DSP with test benches		
Cade Breeding	Finished modules	10	58

Plans for Upcoming Two Weeks

- Gregory
    - Create a D-flip-flop or other more complex design in Magic for our final design, run a Spice simulation on the resulting design. Also get the LEF to harden in the caravel wrapper.
- Jake
    - Fix syntax error with generics in top level modules preventing RTL simulations (But hardening works????)
    - Create a top level testbench for our wrapper user_proj_final
    - Import existing testbench tasks from SPI verification to automate SPI read and write transactions
    - Create a writeup for a Dos and Don'ts of OpenRoad hardening
- Will
    - Create the MAC unit for the DSP module and verify it with a test bench.
    - Start mapping the sub components of the DSP module together at the module level
    - Create top level module that will hold the state machine for the module
    - Create a basic VHDL test bench to test the modules basic functions
    - Explore the more complicated C test bench to utilize the wishbone in all of the test bench
- Cade
    - Finish writing test case and hardening module
    - Help with getting the overall integration started