Project Plan

Sddec23-06

Class: CPR E 491
Advisor: Dr. Duwe

Problem Statement

Problem Description

The ability to create a custom digital ASIC (Application Specific Integrated Circuit) is often locked behind high barriers to entry and restricted to industry professionals.

Goals

- Gain insights on the open source design availabilities
- Test the eFabless capabilities including:
  - Clock Gating
  - Power Managment
  - Custom Cells

Management Style

Agile

- Weekly meetings to address requirements and documentation with group
- Code is on Git for version control everything else is integrated through teams

Task	Deadline	January	February	March	April	May	June	July	August	September	October	November	December
Install the open-source tools and simulate sample code	**2/19/2023**												
Build a sample user project with the Caravel tool flow													
Complete a Verilog simulation of a sample user project													
View the output waveforms of a sample user project in GTKWave													
Define our project specifications	**3/19/2023**												
Create list of many possible modules that could be useful to implement													
Investigate Skywater standard cell libraries to determine what design modules are possible													
From that list narrow down and select the most important few that we would want to develop													
Draw out a top-level diagram of the user area including each individual module	**3/26/2023**												
Determine how each module will interact with clock gating													
Determine how each module will interact with SPI slave													
Determine how each module will interact with included ARM microcontroller													
Determine how each module will interact with the wishbone bus													
Draw out detailed implementation of each module	**4/2/2023**												
Draw out module to include each of the needed subcomponents that will need to be created													
Define interactions between subcomponents for data and control paths needed for full functionality													
Write initial Verilog implementation of each module	**5/14/2023**												
Create a Verilog implementation of each module assigned to individual members													
Test and iterate using RTL simulations	**8/20/2023**												
Create thorough tests that will cover main functionality of module													
Create tests that verify module satisfies top level constraints of the module													
Create tests to cover edge cases outside of typical operating state													
Join modules together and verify final design as they are completed	**8/27/2023**												
Review Verilog modules designed by each team member													
Review Verilog testbenches designed by each team member													
Test and iterate using RTL simulations	**9/22/2023**												
Create timing tests to ensure that all individual modules satisfy the timing requirements of the system													
Create main functionality tests that verify each module gives correct results for main desired task													
Create tests to verify functional interactions between modules													
Verify using gate-level simulation	**9/22/2023**												
Create tests to ensure modules operate with the same functionality as our RTL simulations													
Verify submission using the provided verification tools	**10/1/2023**												
Ensure that final project passes Efabless' precheck tests													
Submit to MPW Shuttle	**10/8/2023**												
Create public repository to satisfy Efabless' open-source requirement													
Create a project on Efabless' website and point at our public repository													
Submit the design to the time applicable OpenMPW shuttle													
Create Software to run on embedded microcontroller	**10/22/2023**												
Create a repository of tested sample code which will verify the integrity of our system													
Create Documentation and bring up plan to test returned project in the future	**11/3/2023**												
Document the bring-up plan in a detailed form for a future user (at the level of a 288 student) could use to test our design when returned from eFabless													

Risks & Risk Mitigation

Risk	Estimated Probability
No MPW submission is available	40%
DSP does not fit in user area	40%
Wishbone bus does not work	5%
Fabrication error causes any individual module to fail	15%
Clock system contains errors before first round of testing	99.9%

Personal Effort Level

Task	Person Hours
Install the open-source tools and simulate sample code	20
Define our project specifications	20
Draw out a top-level diagram of the user area including each individual module	30
Draw out detailed implementation of each module	30
Write initial Verilog implementation of each module	80
Test and iterate using RTL simulations	80
Join modules together and verify final design as they are completed	40
Test and iterate using RTL simulations	70
Verify using gate-level simulation	70
Verify submission using the provided verification tools	40
Submit to MPW Shuttle	10
Create Software to run on embedded microcontroller	50
Create Documentation and bring up plan to test returned project in the future	80