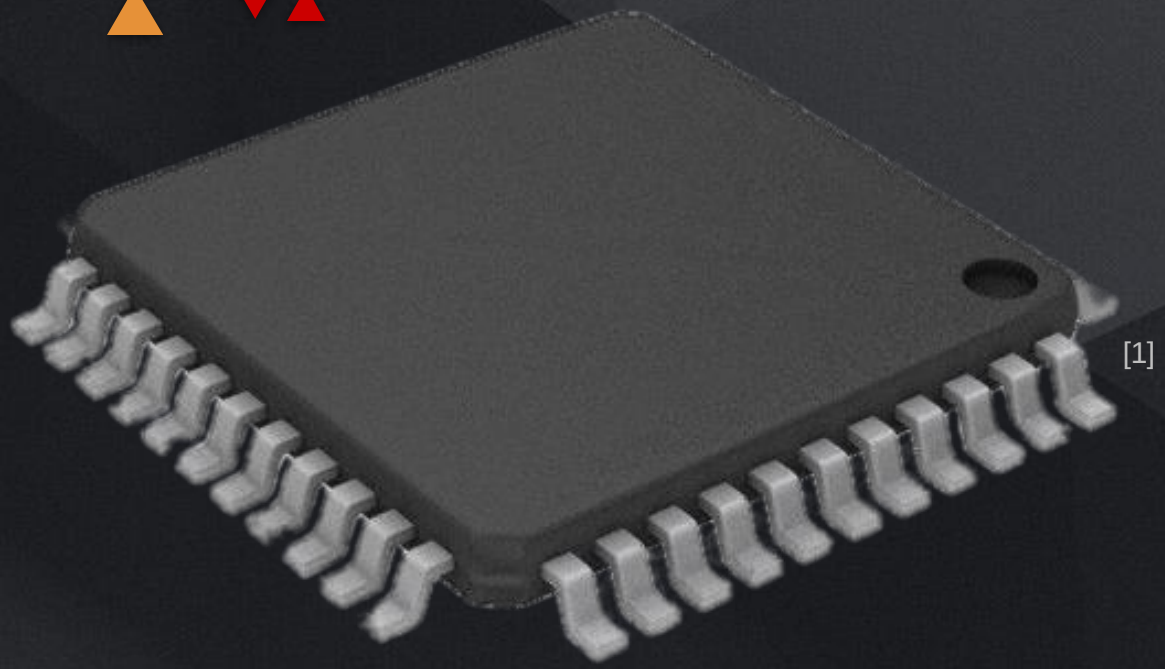

[1]

Digital ASIC Design
Industry Panel Presentation

Gregory Ling
Will Galles
Jake Hafele
Cade Breeding

Class: CPR E 492
Group: sddec23-06
Client/Advisor: Dr. Duwe
Website: http://sddec23-06.sd.ece.iastate.edu/

Introduction

Gregory Ling

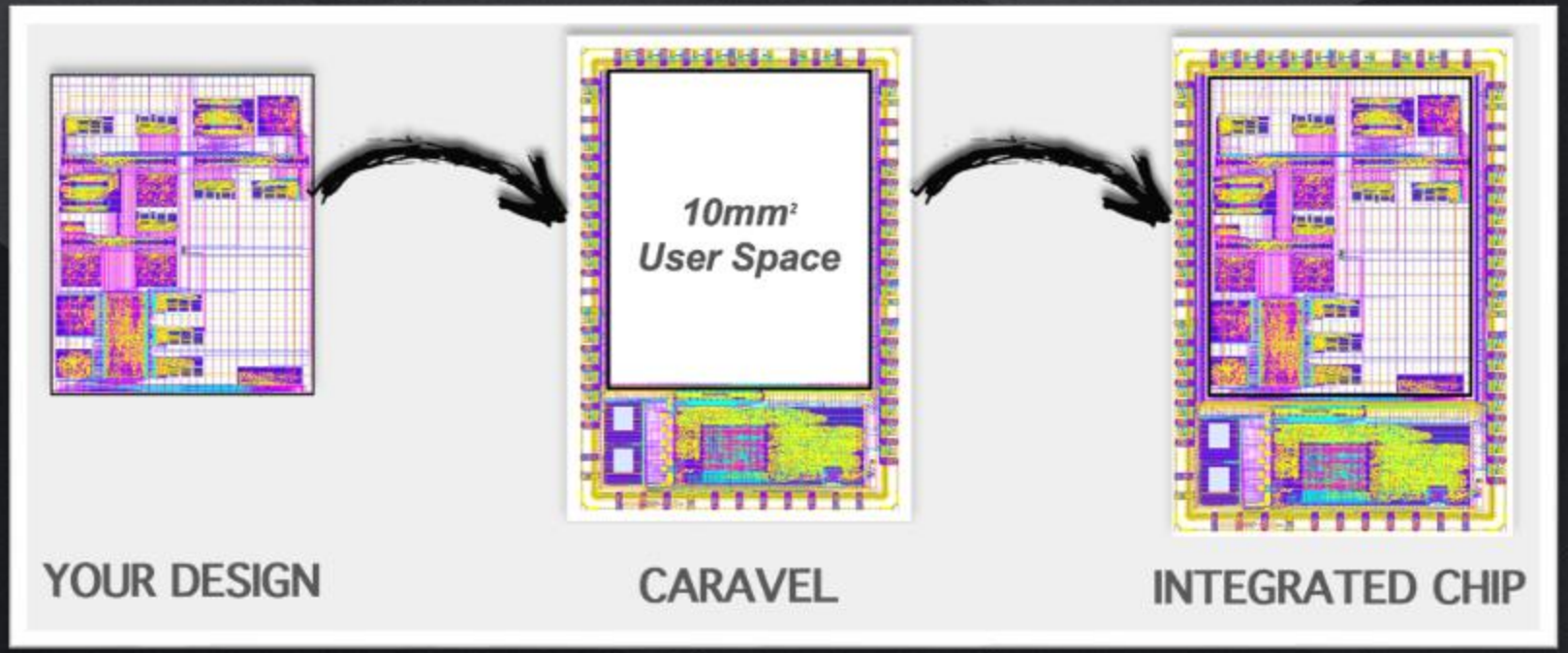
Problem Statement
Solution
Requirements
Users
Market Survey

Problem Statement

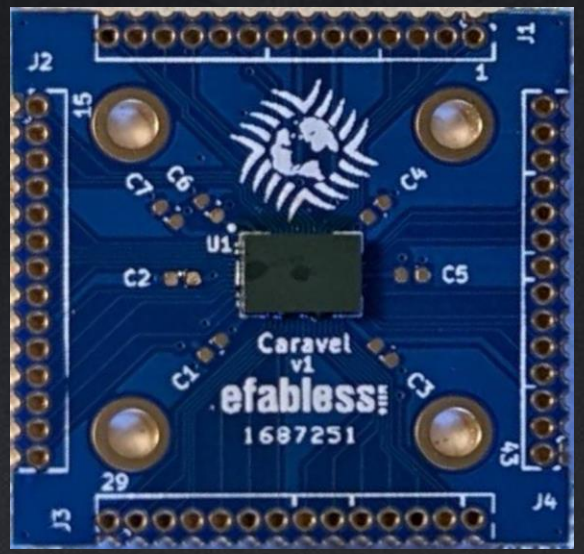
The ability to create a custom digital ASIC (Application Specific Integrated Circuit) is often locked behind high barriers to entry and traditionally restricted to teams of industry professionals.

OpenMPW

- Multi-Project Wafer
- Skywater 130nm PDK
- OpenROAD
  - Yosys
  - Klayout
  - Magic
  - XSchem



[2]



[3]

p. 16-18

Goals

Explore

Modular design to test different limits of the fabrication process

Learn

Expand the knowledge base of open-source fabrication at ISU

Bring-Up Plan

Create a plan for a future group to bring-up our fabricated design

Users

Our Team

Research Groups

Future Teams

Open Source Community

Market Survey and Cost Estimate

Production Run

- ~$1M for full ASIC

Multi-Project Wafer

- OpenMPW – eFabless, open-source, funded by Google
- ChipIgnite – eFabless, paid ($9,750/project, 10mm^2, 4-5 months)
- Muse Semiconductor – TSMC, paid ($1,250/mm^2, 42 days)

Our Cost

- All design tools are open source
- Only real cost of the project is our time

Design

Will Galles

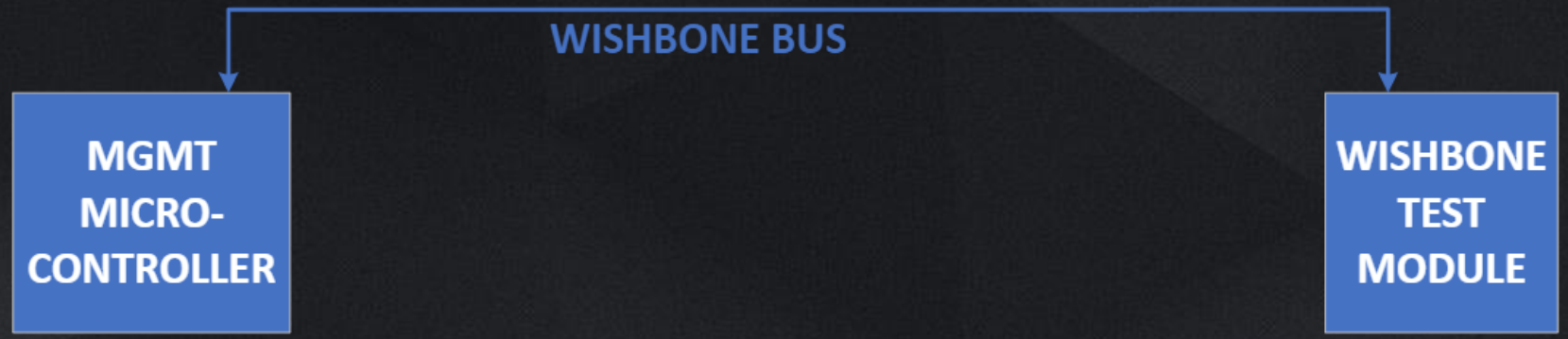
Design Requirements
Top Level Design
Module Descriptions

Technical Design Requirements

- Create a modular design that would allow for a wide array of operations to take place
- Create a redundant system that would still be able to operate if there was a manufacturing failure
- Explore new technologies and processes that had not been explored by previous teams

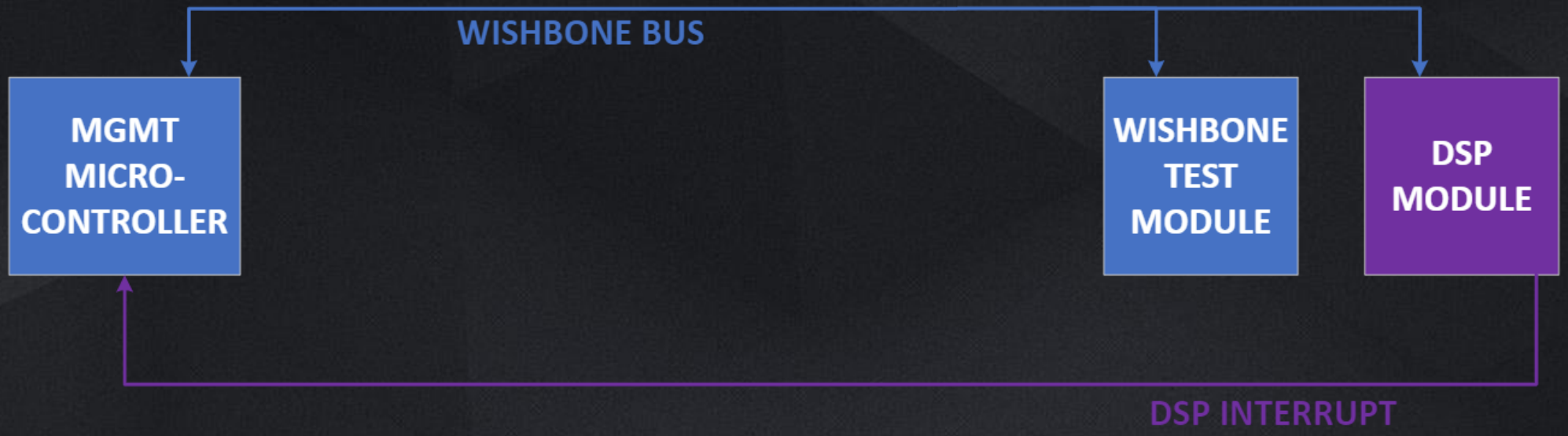
Wishbone Test Module

Purpose
Test on chip communication buses



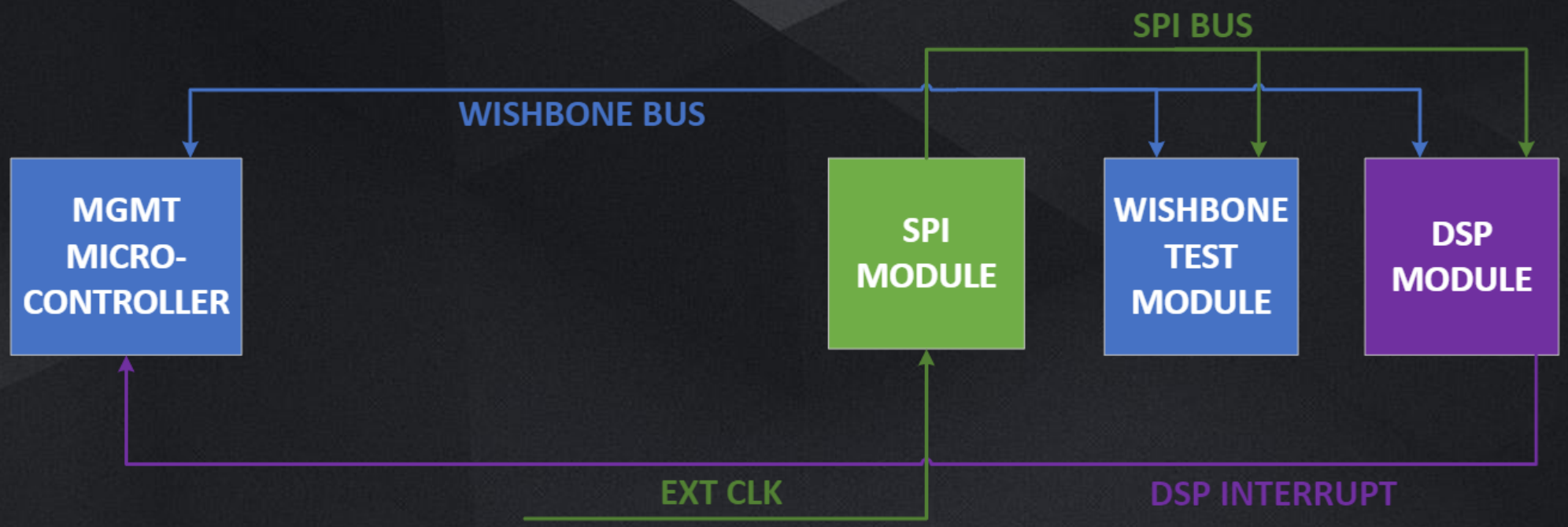
DSP Accelerator Module

Purpose
Utilize OpenRAM generated SRAM
Implement a more complex design



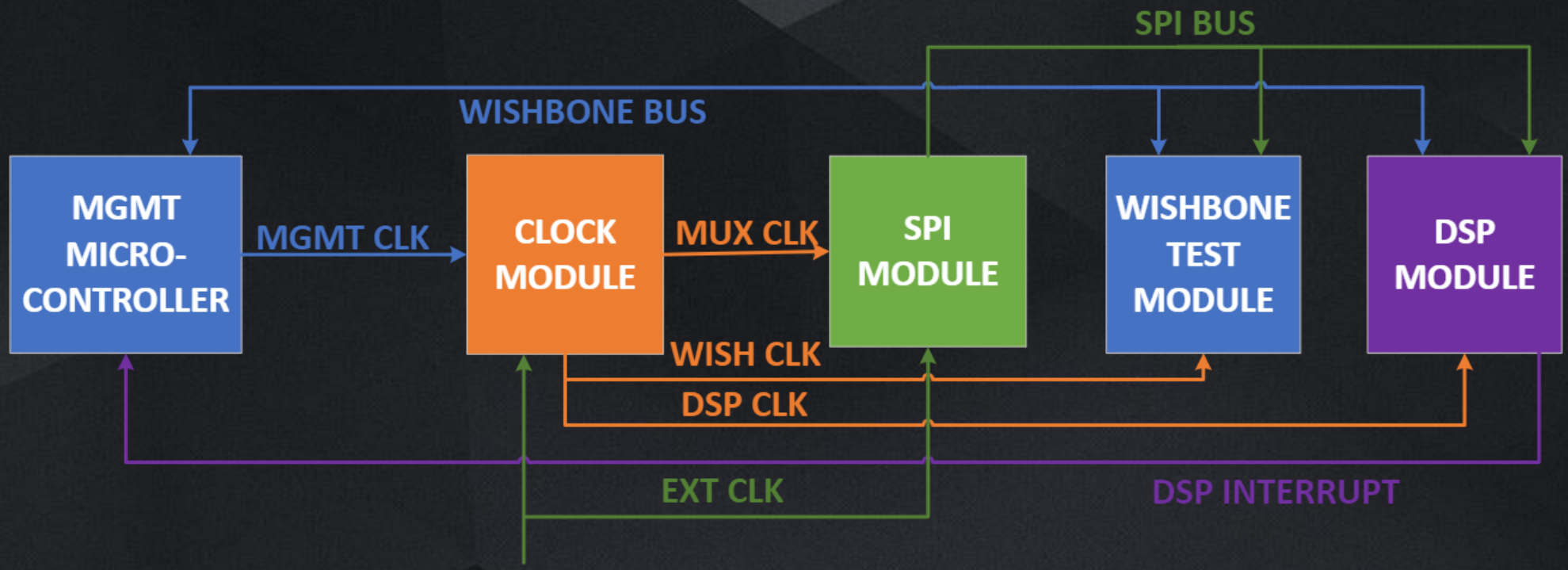
Backdoor SPI Module

Purpose
Backup data bus in case of failure
Test crossing clock domains



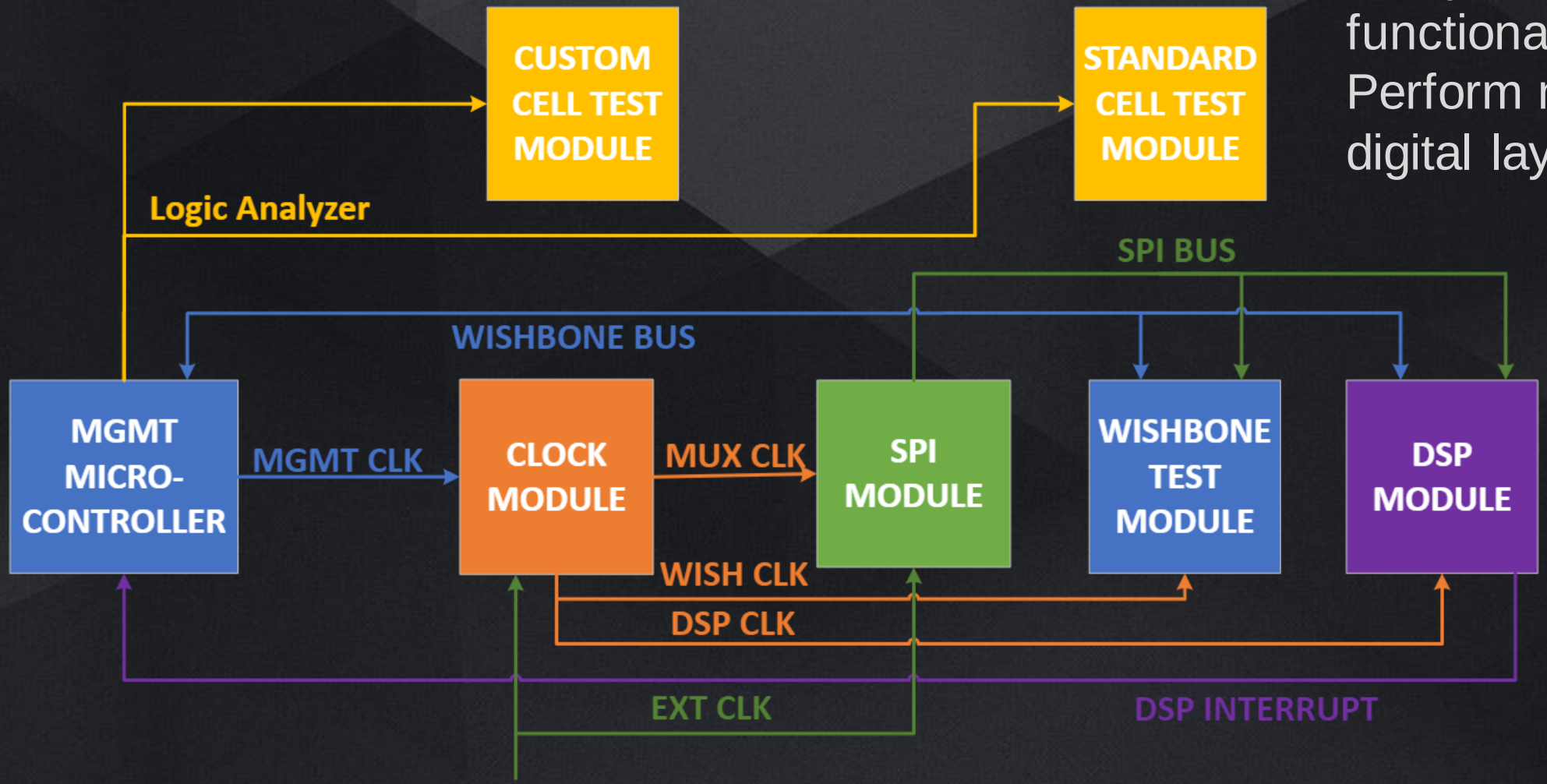
Clock Module

Purpose
Test multiple clock domains
Backup clock in case of failure



Standard Cell & Custom Cell

Purpose
Verify base PDK functionality
Perform manual digital layout



Test Overview

Jake Hafele

Platforms Used
Test Plan

Platforms Used

OpenROAD

- RTL synthesis through GDS Layout

Magic

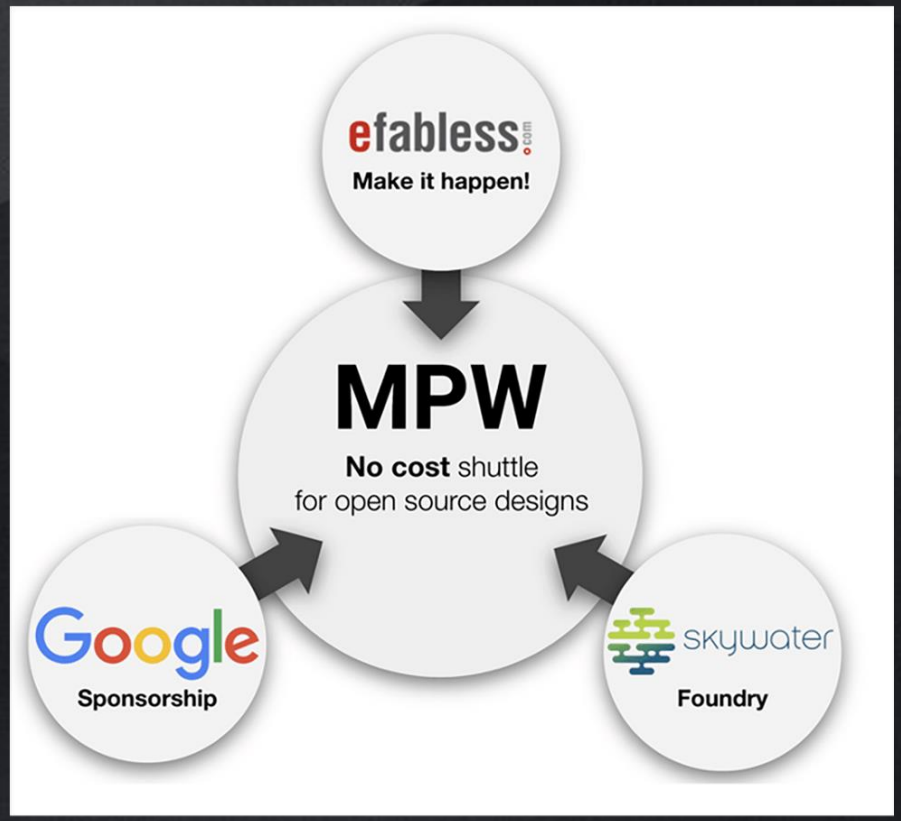
- Custom cell layout

KLayout

- Layout viewer

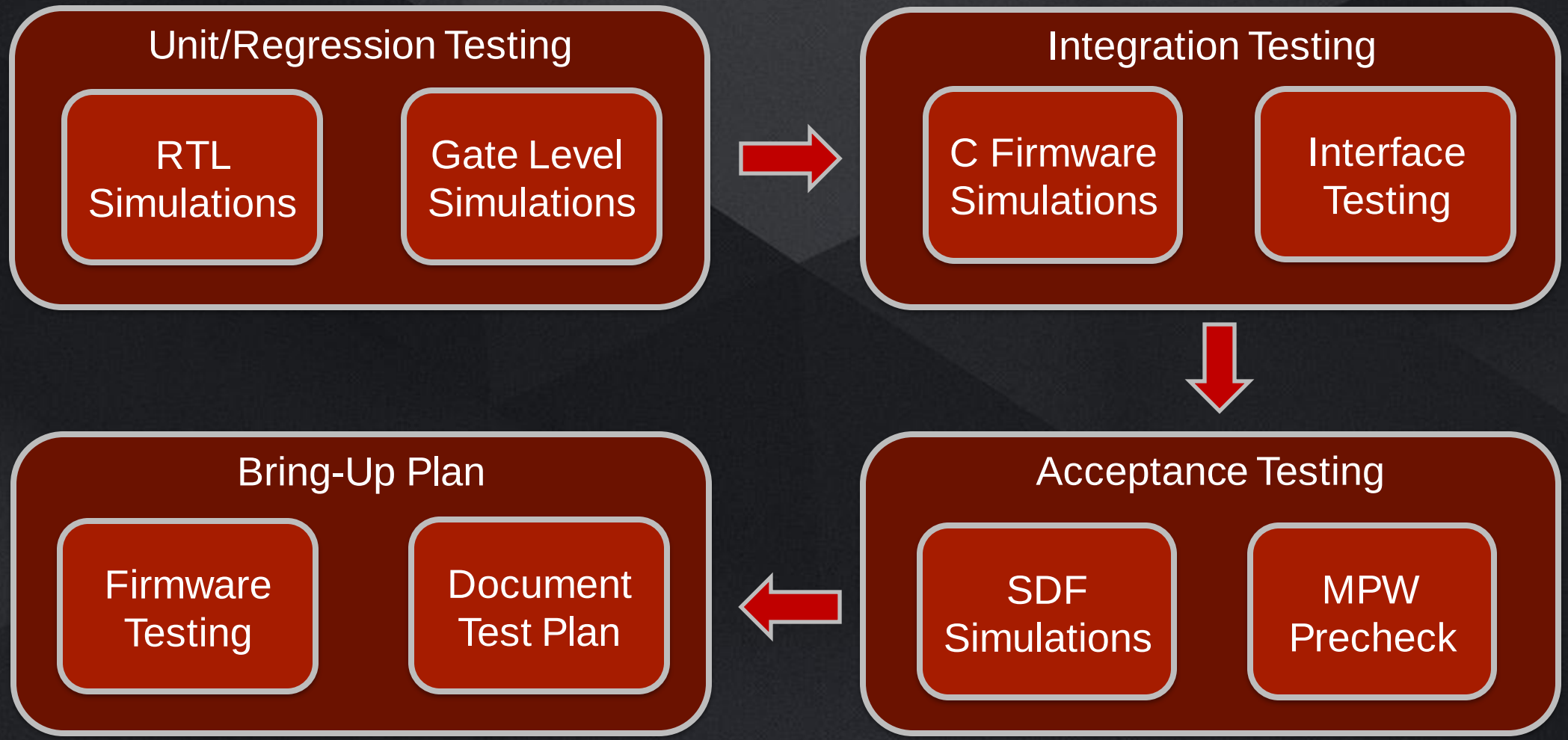
GTKWave

- Waveform viewer



[4]

Test Plan



pp. 36-39

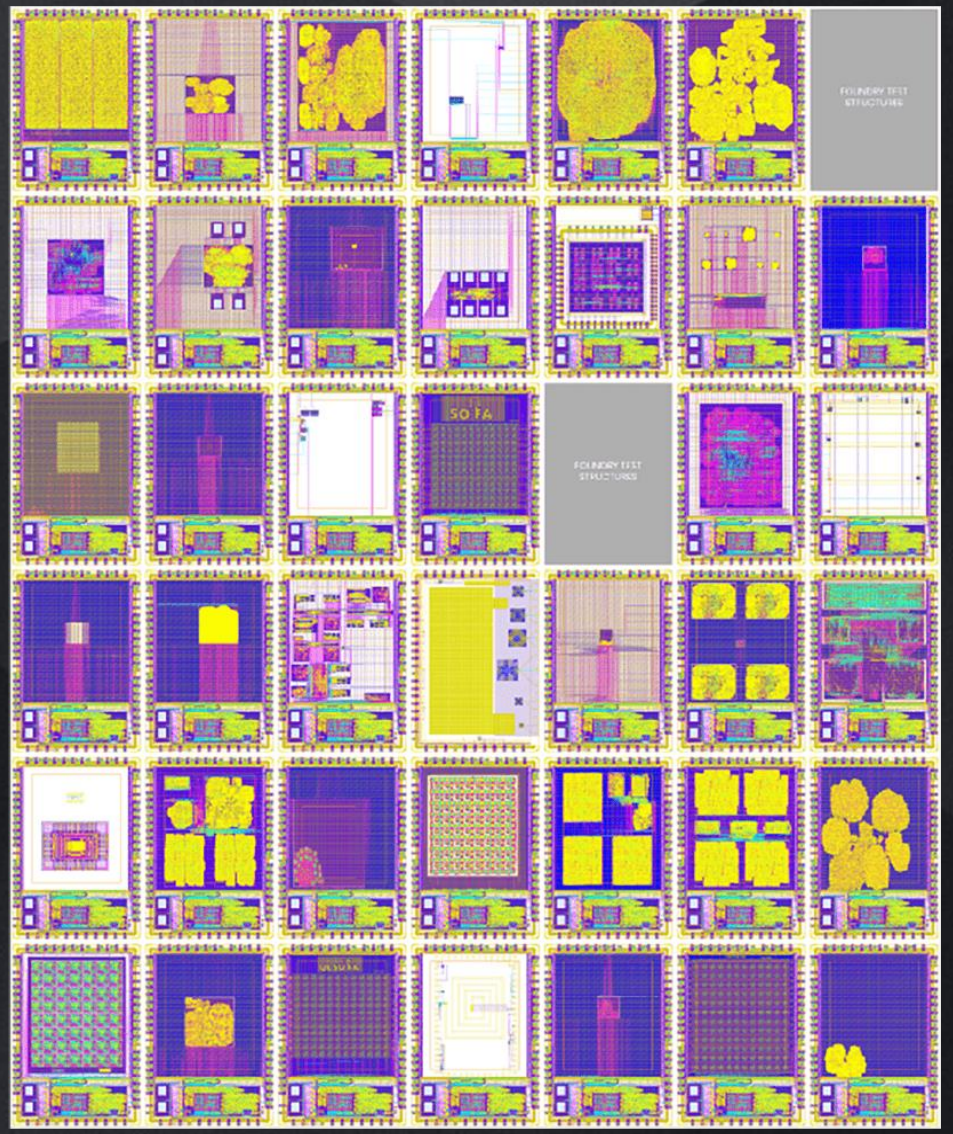
Implementation

Jake Hafele

Behavioral RTL
Custom Cells
SRAM

Overview

- **What is hardening?**
  - Provides compilation, synthesis, place and route, and signoff all in one package
  - Generates hardened macros for submodules AND hardened top-level design
- **Different implementation techniques**
  - Behavioral Verilog
  - Custom Cell
  - SRAM

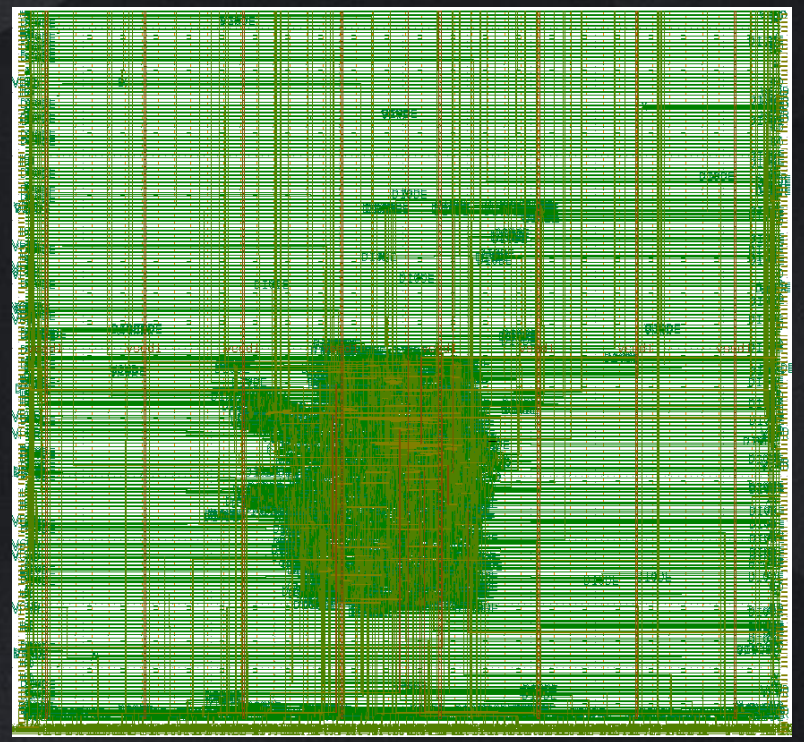


[4]

Behavioral RTL

- Define functionality of modules using Verilog
- "Common Case" for most designs
- Can be utilized for full design process

- **Used for:**
  - Standard Cell Test
  - Backdoor SPI
  - Wishbone Bus Test
  - DSP Accelerator
  - Custom Cell (functional verification only)
  - Clock MUX (functional verification only)

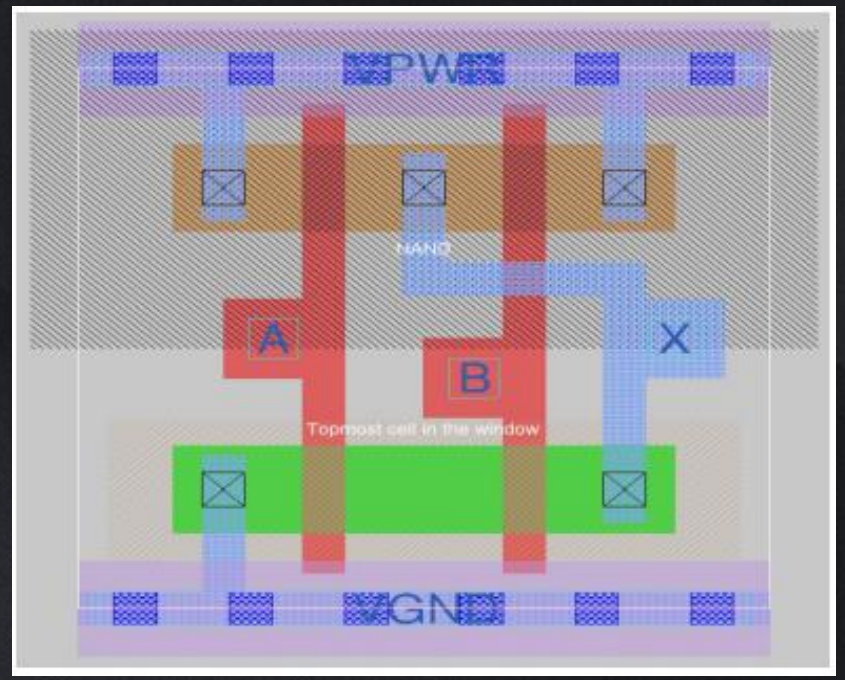


Custom Cell

- Create custom cell for use in parallel with SkyWater 130nm PDK
- Utilize open-source tools and new process for team and users

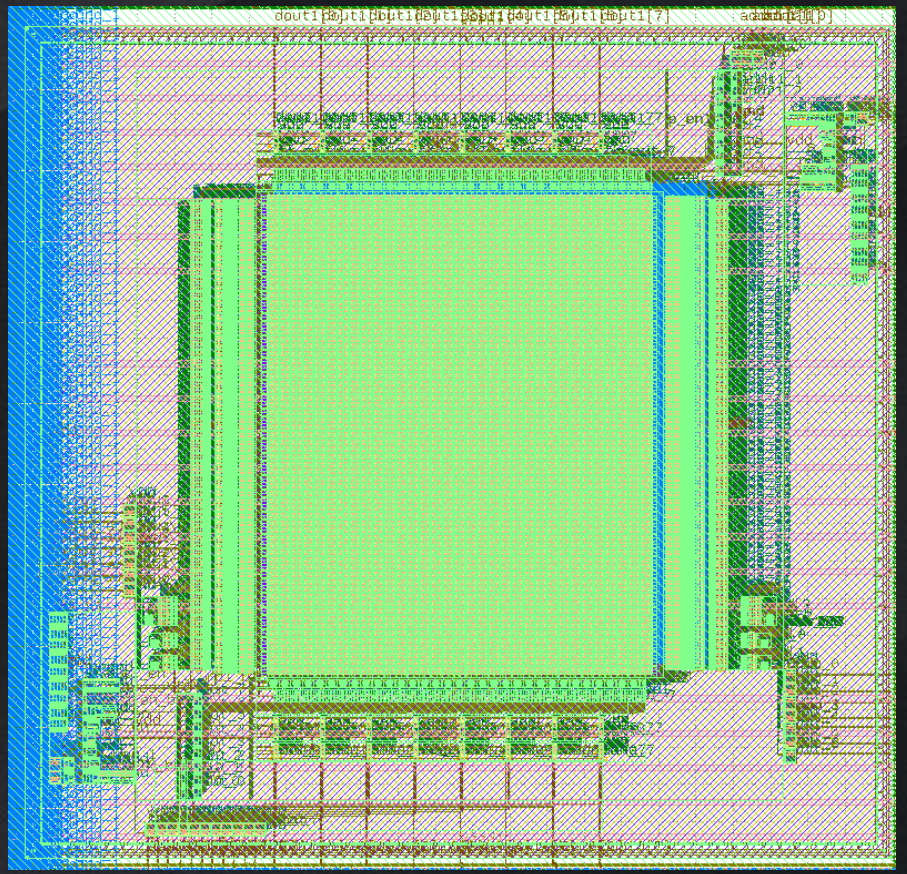
- **Used for:**
  - Custom Cell (GL verification, hardening)

- **Unique Process:**
  - Magic
  - XSchem



SRAM

- Utilizes OpenRAM project for SkyWater 130nm fabrication process
- **Requirement:**
  - Had to be placed in top level wrapper
- **Used for:**
  - DSP Accelerator (GL verification, hardening)
- **Unique Process:**
  - OpenRAM



Project Deliverables

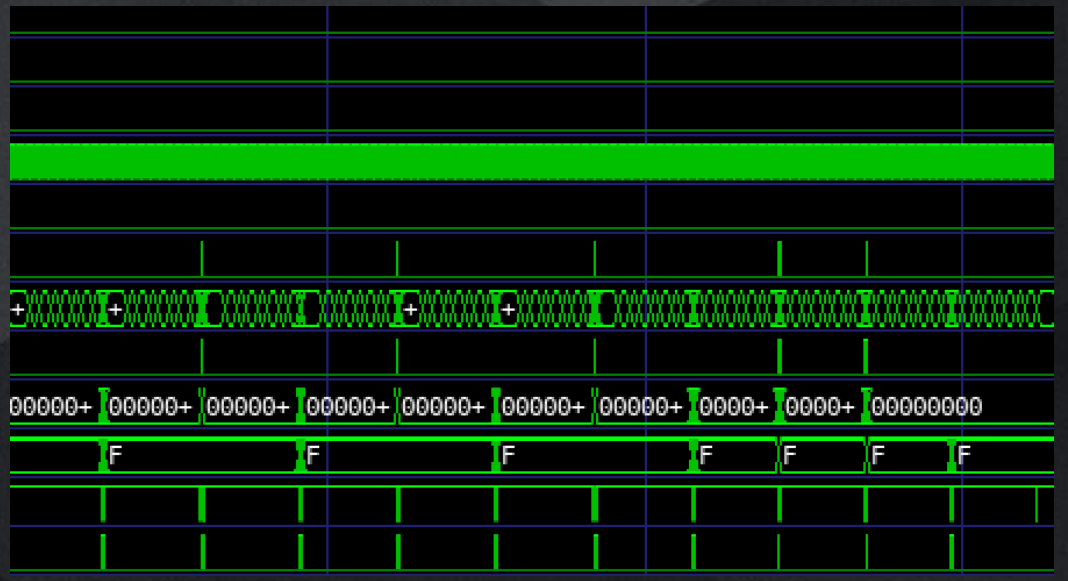
Cade Breeding

Functional Design
Precheck Pass
Caravel User Guide
Bring-Up Plan

Deliverable 1 – Functional Design

What was utilized?

- Modular Verilog designs
- Custom NAND cell
- Pre-hardened blackbox designs



Validated submodule and top-level designs

- RTL Verification
- GL Verification
- C Testing

```
c_test.hex loaded into memory
Memory 5 bytes = 0x6f 0x00 0x00 0x0b 0x13
FST info: dumpfile c_test.vcd opened for output.
                464215*** C Test Started! ***
               3598340*** C Test Finished!***
               3598340*** PASS ***
c_test_tb.v:70: $finish called at 3598339500 (1ps)
```

Deliverable 1 – Precheck Pass

Client Request

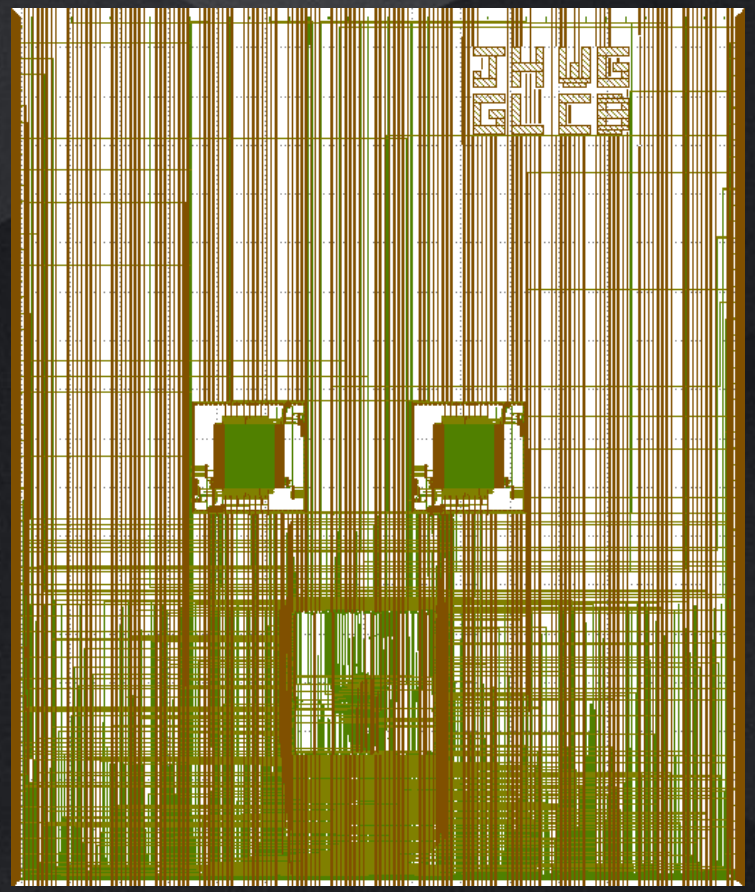
- Product passes the required precheck for MPW submission
- Perform signoff with SDF simulations

Delivered Product

- Precheck passes the required tests for submission

Contents of Product

- Screenshots and code repository proving precheck pass



```
{{FINISH}} Executing Finished, the full log 'precheck.log'
{{SUCCESS}} All Checks Passed !!!
```

Deliverable 2 – Caravel User Guide

Client Request

- Document implementation processes including behavioral RTL, custom cell, and SRAM
- Document testing process for RTL, GL, SDF, and C testing

Delivered Product

- Detailed instruction manual with referenced implementation and verification processes

Contents of Product

- Appendix 8.4.1

Deliverable 3 – Bring-Up Plan

Client Request

- Plan that future groups can use to verify functionality of manufactured design

Delivered Product

- Bring up plan that will comprehensively test our design to ensure viability

Contents of Product

- C tests that can be run on the management soc to verify the device
- Written plan with the startup procedure

Conclusion

Cade Breeding

Challenges
Future Work
Final Summary

Potential Risks Identified in Semester 1

- No OpenMPW submission is available
- DSP module cannot both fit in user area and run at real time – Success
- Wishbone bus is unable to interact with user modules after fabrication
- Fabrication error causes an individual module to fail

pp. 24-25

Challenges

- LVS problems during hardening
- Updating the PDK version
- Open-source tools come with their own challenges and behave differently then propriety tools.
- Precheck DRC issues with the custom cell
- SDF Simulations are extremely slow

Future Work

The next step is to submit to the MPW9 shuttle and receive our fabricated design

Future testing groups can utilize our Bring-Up Plan deliverable and module specifications to validate our fabricated design.

Future senior design teams can utilize our Caravel User Guide deliverable to design more complex designs at a faster rate.

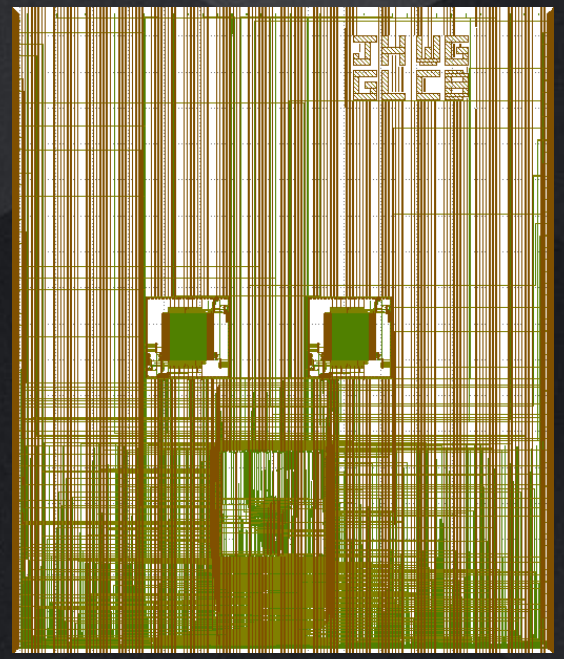
Final Summary

Design:

- Functional requirements met
- Precheck passed
- Ready for MPW submission and fabrication

Knowledge:

- Created bring up plan documentation
- Create user guide documentation
- Solved new and existing issues with design processes (Custom cell, SRAM)



```
c_test.hex loaded into memory
Memory 5 bytes = 0x6f 0x00 0x00 0x0b 0x13
FST info: dumpfile c_test.vcd opened for output.
                464215*** C Test Started! ***
               3598340*** C Test Finished!***
               3598340*** PASS ***
c_test_tb.v:70: $finish called at 3598339500 (1ps)
```

```
{{FINISH}} Executing Finished, the full log 'precheck.log'
{{SUCCESS}} All Checks Passed !!!
```

References

[1] https://www.pngall.com/microcontroller-png/
[2] https://efabless.com/open_shuttle_program
[3] https://www.zerotoasiccourse.com/post/mpw1-bringup/
[4] https://efabless.com/open_shuttle_program

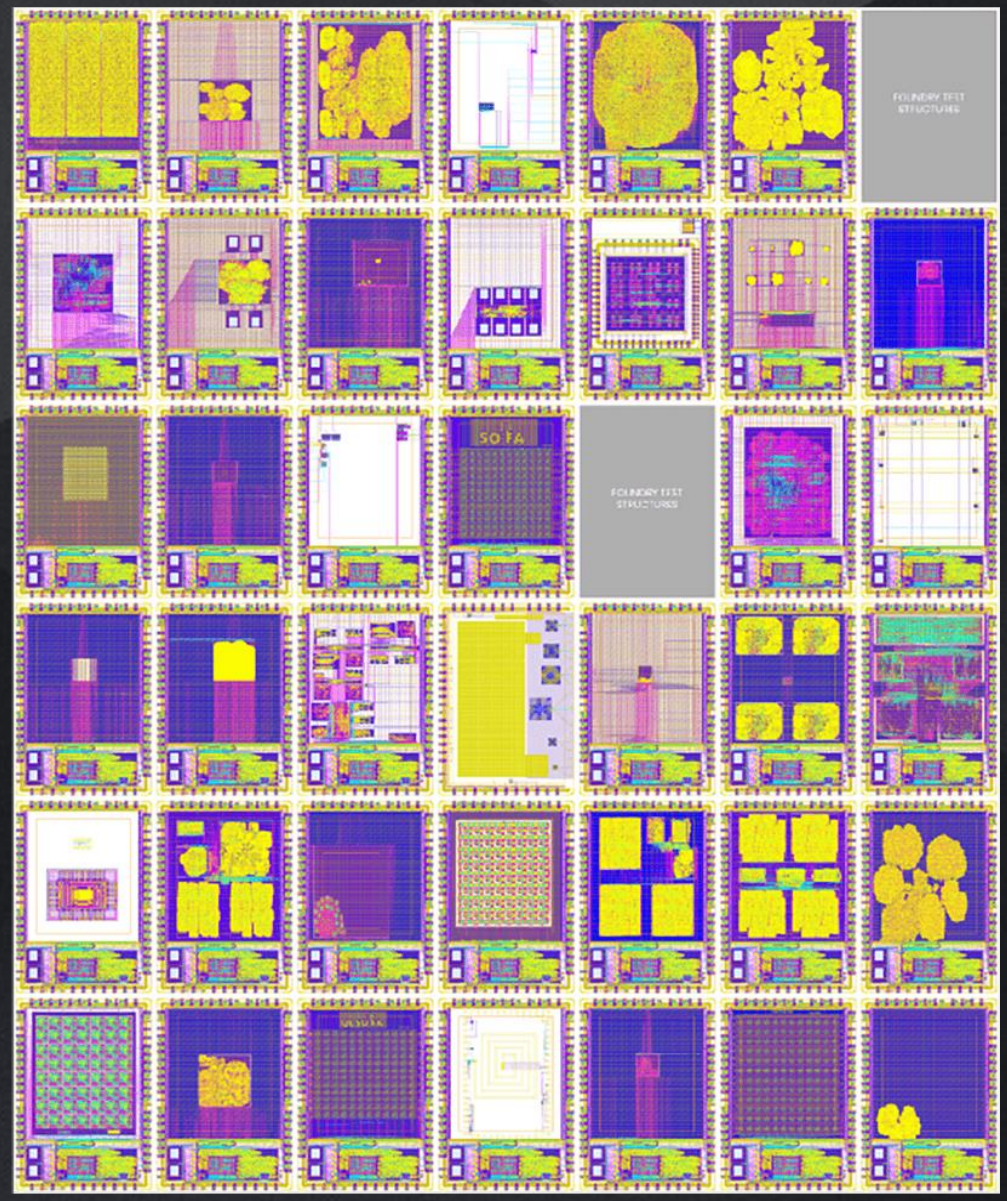
Questions?

Supplemental Material

Constraints

eFabless

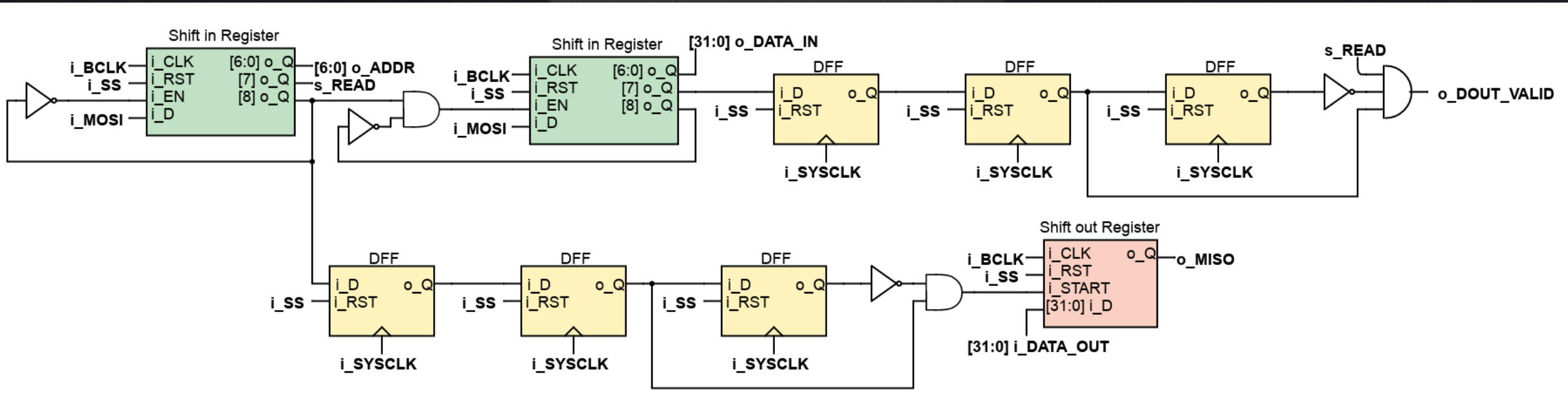
- 10 mm^2 user area
- SkyWater 130 nm fabrication process
- Verilog
- Specific folder structure
- MPW Precheck
- Open-source repository



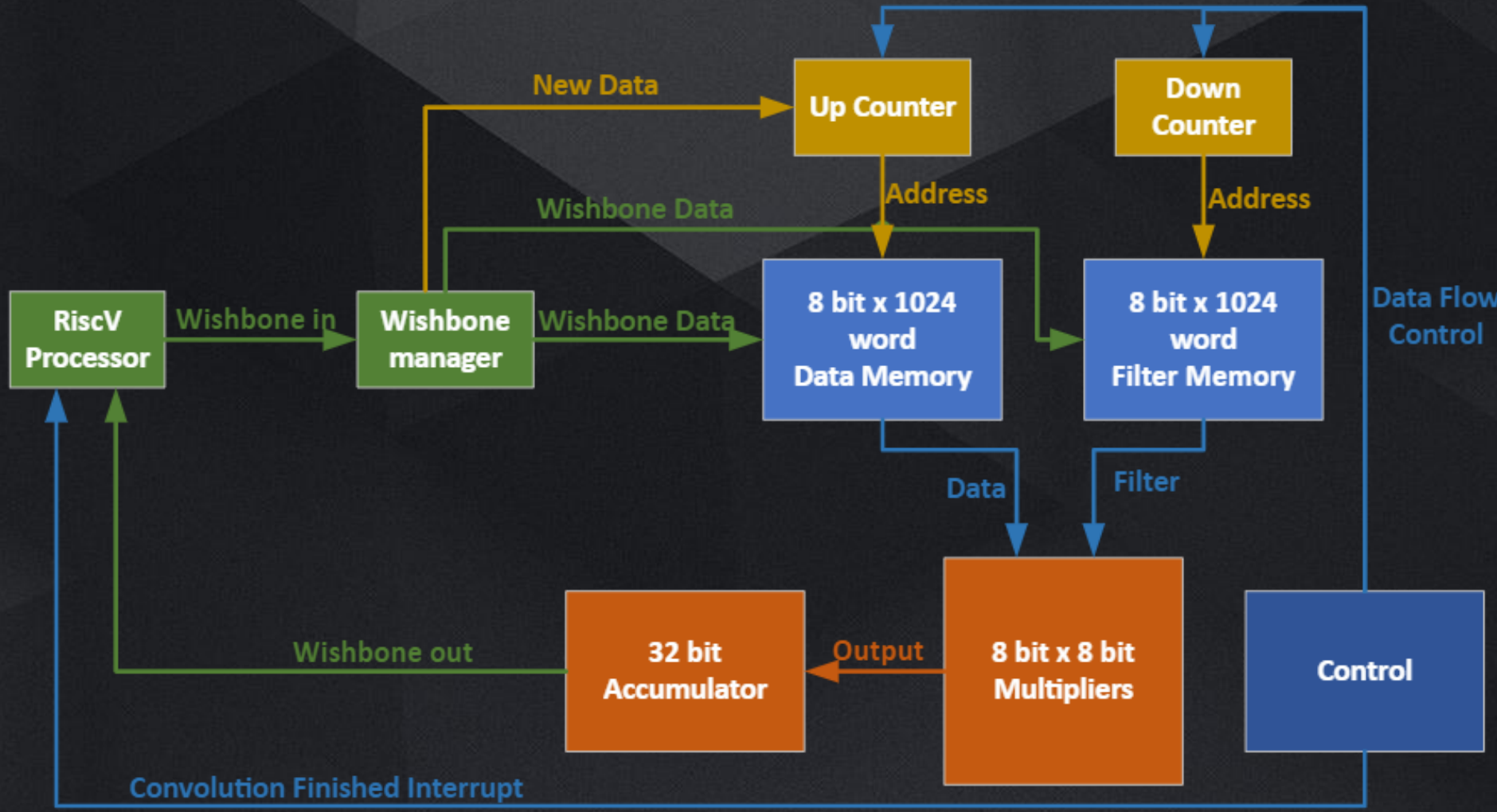
[4]

pp. 16-19

Backdoor SPI Module



Voice Road Noise Isolation Accelerator Module (DSP)



Does our design look like a Minecraft villager?

https://www.minecraft.net/en-us