Working with the

Caravel Harness

ZEN team: Cole Blackman, Justin Zhang, Ceylan Morgul

ECE HPLP Lab UVA 2021

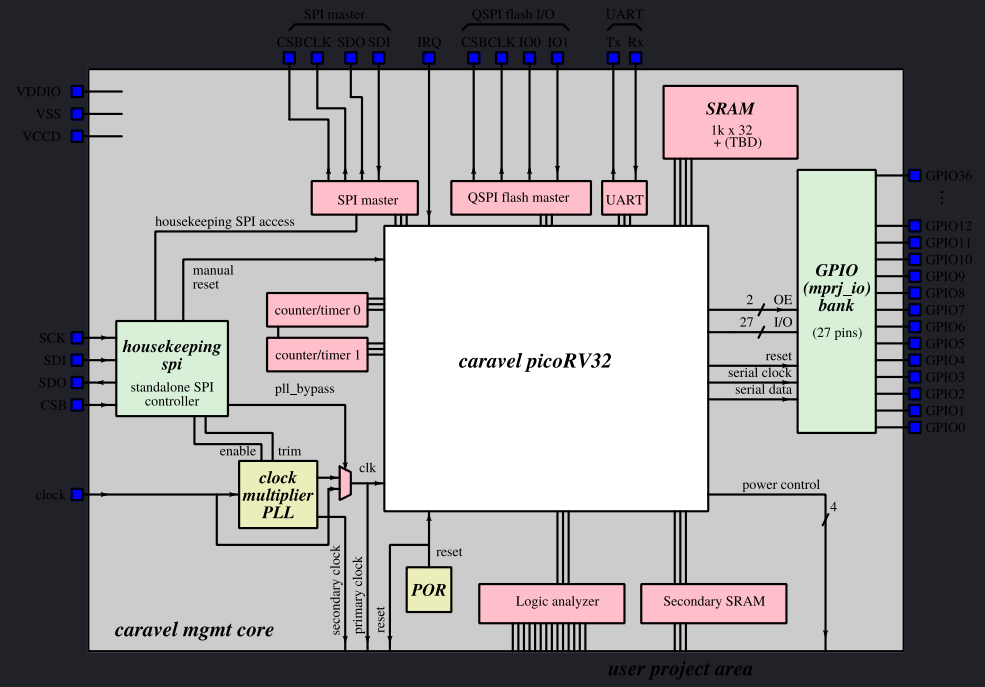


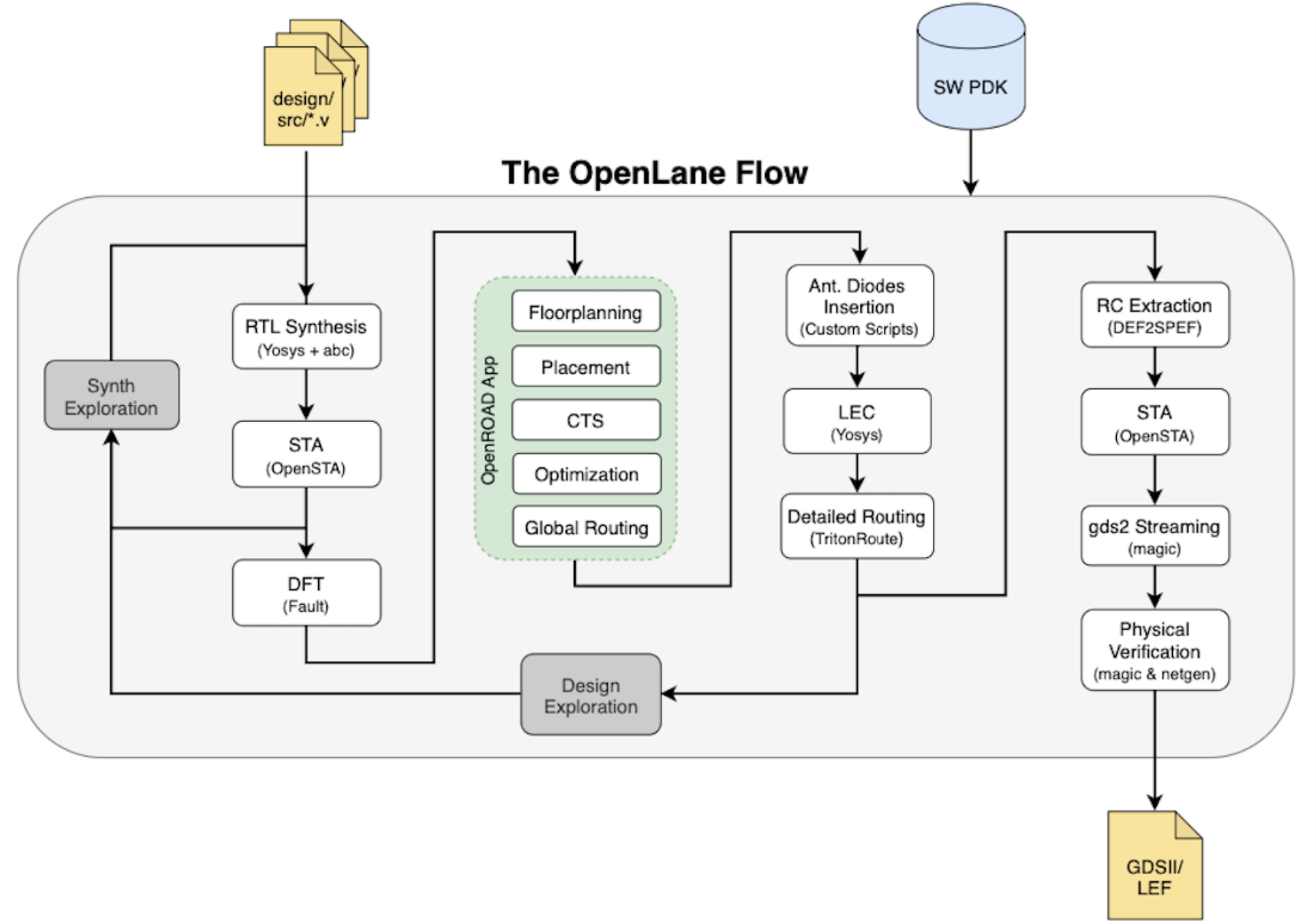
Table of Contents

Background: Skywater PDK and Sky130 process node

- Open source toolset
- Google and Skywater
- Design fabrication processes and ICs
- WIP (not supported in production yet/alpha)
- https://skywater-pdk.readthedocs.io/en/latest/

Background: OpenLane Flow

- Complete process for building GDSII database files (IC layouts)
- "Converts" designer's verilog files to GDSII instructions


The OpenLane Flow
design/
src/*.v
SW PDK
Synth
Exploration
RTL Synthesis
(Yosys + abc)
STA
(OpenSTA)
DFT
(Fault)
OpenROAD App
Floorplanning
Placement
CTS
Optimization
Global Routing
Ant. Diodes
Insertion
(Custom Scripts)
LEC
(Yosys)
Detailed Routing
(TritonRoute)
RC Extraction
(DEF2SPEF)
STA
(OpenSTA)
gds2 Streaming
(magic)
Physical
Verification
(magic & netgen)
Design
Exploration
GDSII/
LEF

Background: Efabless Open MPW Shuttle Program

- Fabricates your Skywater PDK 130nm process design using Openlane for free
- Your design must conform to submission standards (open source, git-compatible repo, etc)
- Design must pass checks

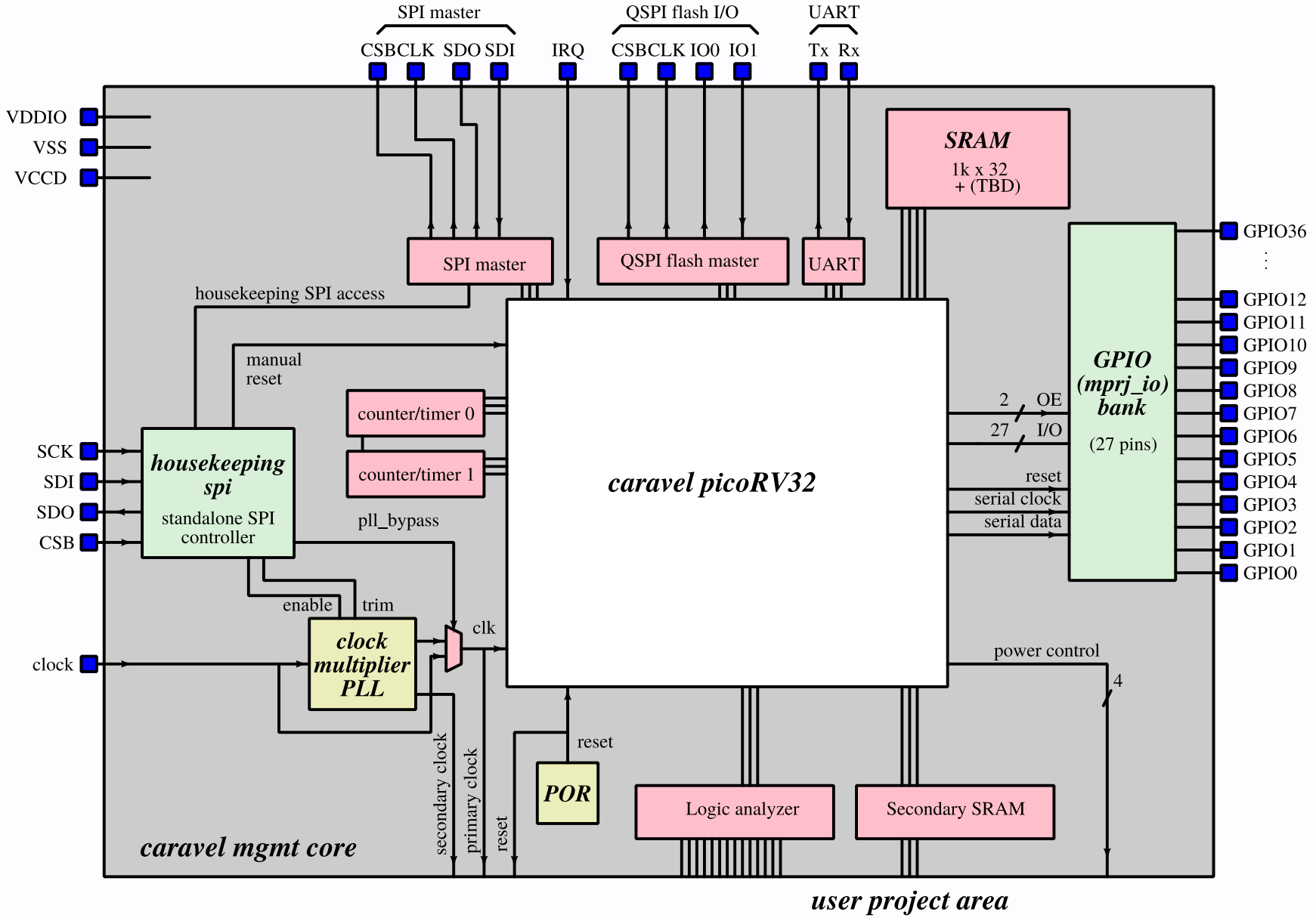
Caravel

- Test harness for Skywater 130nm OS PDK
- RISC-V RV32IMC IS running on PicoRV32 processor core
- **Provides a test "harness" consisting of tools like SRAM and GP I/O for open user area circuits**

Components of Caravel

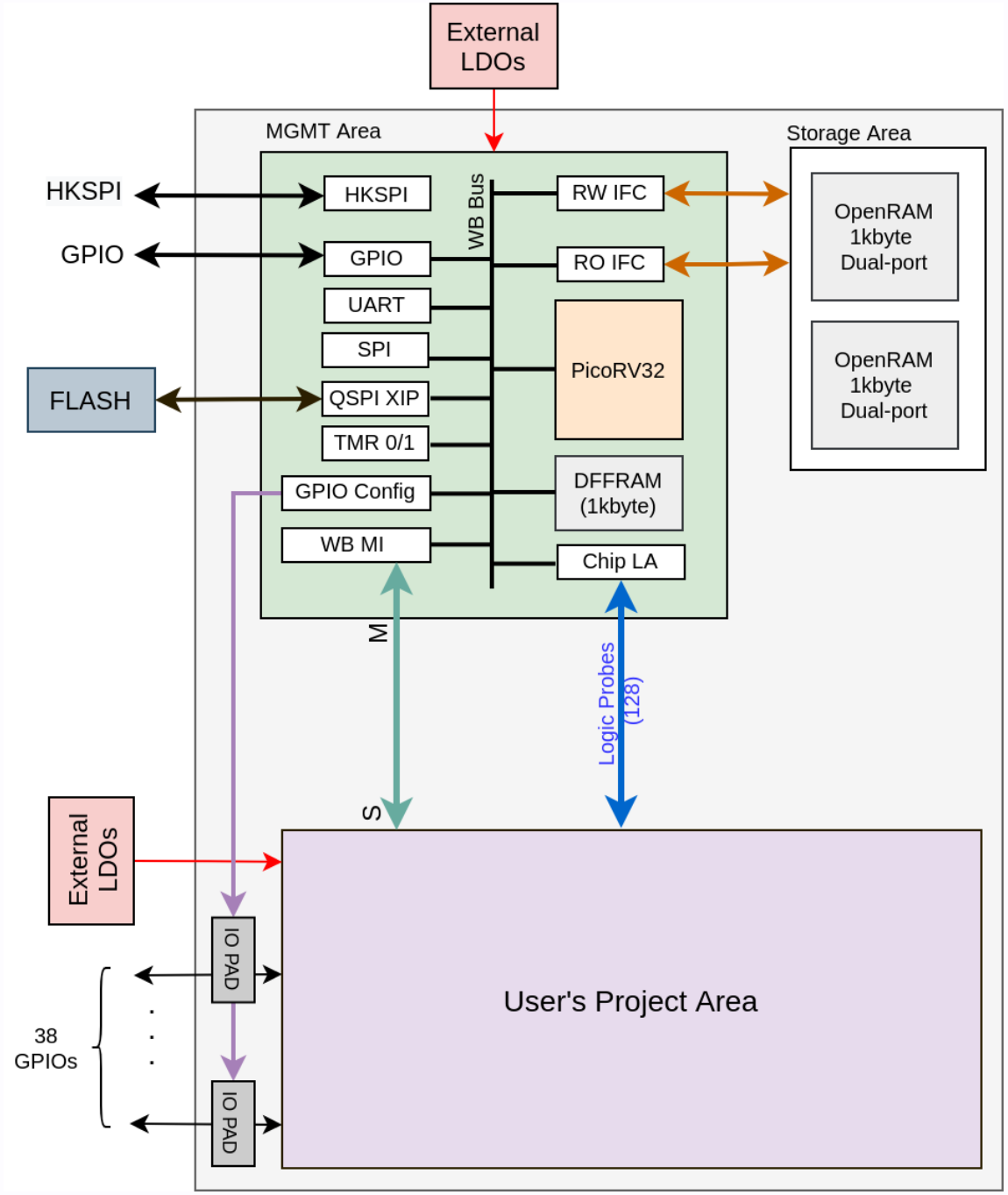
What does caravel provide the developer?

- Wishbone (32-bit)
- 128-bit logic analyzer
- UART
- SPI controller and SPI flash controller
- Large SRAM
- I/O
- And more...


SPI master
CSBCLK SDO SDI
IRQ
QSPI flash I/O
CSBCLK IO0 IO1
UART
Tx Rx
VDDIO
VSS
VCCD
SRAM
1k x 32
+ (TBD)
SPI master
QSPI flash master
UART
GPIO36
GPIO12
GPIO11
GPIO10
GPIO9
GPIO8
GPIO7
GPIO6
GPIO5
GPIO4
GPIO3
GPIO2
GPIO1
GPIO0
housekeeping SPI access
manual
reset
counter/timer 0
counter/timer 1
caravel picoRV32
GPIO
(mprj_io)
bank
(27 pins)
2 OE
27 I/O
reset
serial clock
serial data
SCK
SDI
SDO
CSB
housekeeping
spi
standalone SPI
controller
pll_bypass
enable
trim
clock
clock
multiplier
PLL
clk
power control
4
reset
POR
Logic analyzer
Secondary SRAM
secondary clock
primary clock
reset
caravel mgmt core
user project area

Comments on Diagrams

- mprj_io I/O bank interfaces with caravel userspace
- User project area die size has decreased (from Efabless MPW Shuttle 1 to Shuttle 2)


External LDOs
MGMT Area
Storage Area
HKSPI
GPIO
FLASH
HKSPI
GPIO
UART
SPI
QSPI XIP
TMR 0/1
GPIO Config
WB MI
WB Bus
RW IFC
RO IFC
PicoRV32
DFFRAM (1kbyte)
Chip LA
OpenRAM 1kbyte Dual-port
OpenRAM 1kbyte Dual-port
M
S
Logic Probes (128)
External LDOs
IO PAD
IO PAD
38 GPIOs
User's Project Area

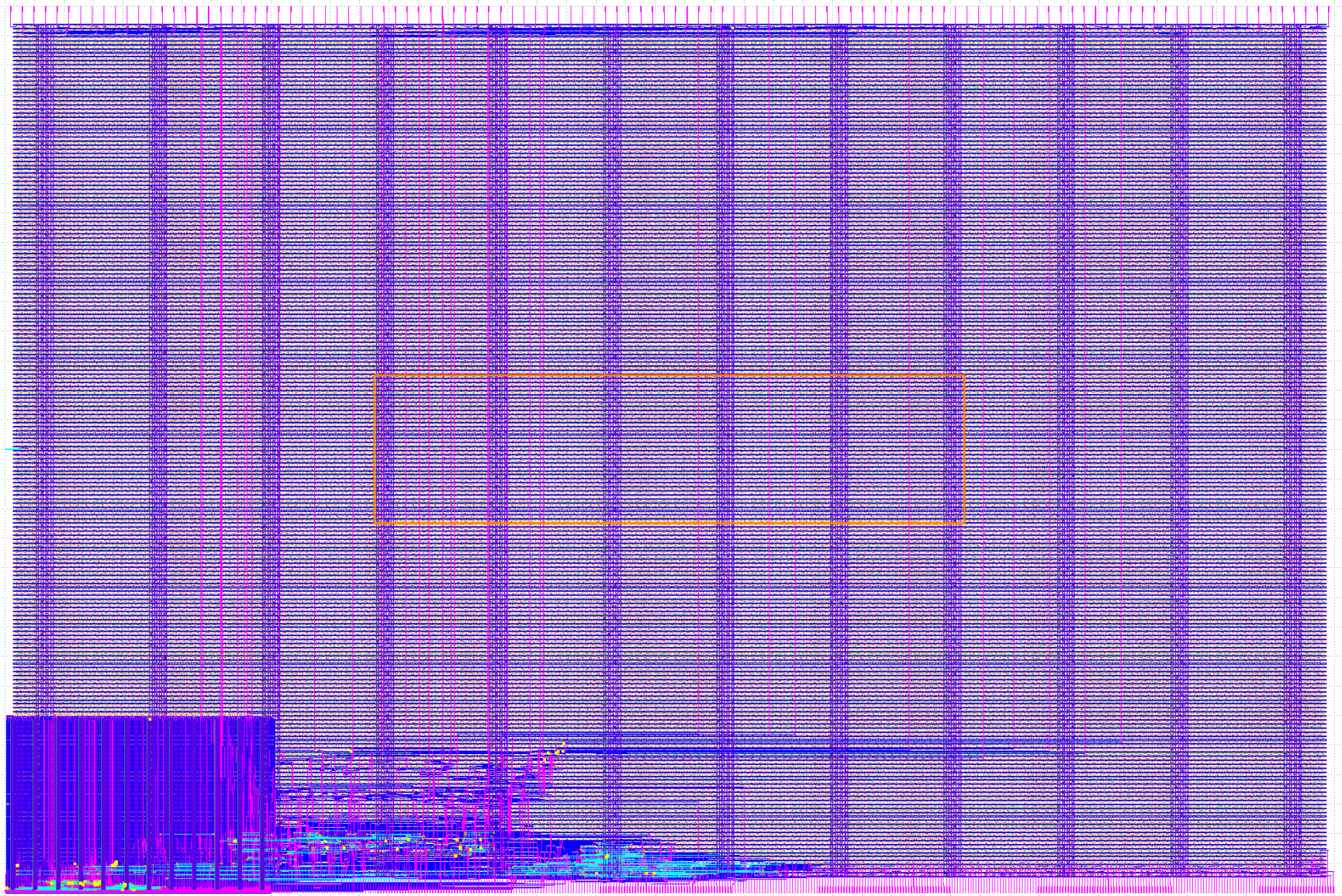
Openlane Hardening

Writing Verilog for Caravel

We already saw overhead diagrams of the circuit layout, but how can a designer interface with I/O provided by the harness and build functional chips on this platform?

Johnson Counter Example

- This example defines circuits for the caravel userspace, but does not require the use of I/O (see verilog code on our repository)



IO Example: SHA3-256

*Unlike the previous example, this was written by Jean Cyr, not by us. We changed sections of the code to be compatible with the new die size requirements of the MPW Shuttle 2.0 and rewrote small sections for optimization.

SHA-3 Keccak

- Hashing algorithm (secure hashing algorithm series of functions)
- Multi-step process based on manipulating the bits of the input data
- Computing SHA-3 hashes is a technique for Proof of Work in some cryptocurrencies

```
sha3_256_miner_regs #
(
   .DATA_WIDTH(DATA_WIDTH),
   .ADDR_WIDTH(ADDR_WIDTH)
)
s3r
(
   .clk(wb_clk_i),      // clock
   .reset(wb_rst_i),    // reset
   .addr(wbs_adr_i),    // address
   .ack(wbs_ack_o),     // acknowledge
   .read(active_cyc),   // read
   .write(write_cyc),   // write
   .rdata(wbs_dat_o),   // data in
   .wdata(wbs_dat_i),   // data out
   .sel(wbs_sel_i),     // select

   .header_o(header),               // header
   .difficulty_o(difficulty),      //
   .start_nonce_o(start_nonce),
   .control_o(control),
```

SHA3-256 on Caravel (Optimizations)

INSERT CODE SNIPPET HERE

SHA3-256 on Caravel (Images)

INSERT IMAGES OF HARDENING HERE

Our Work

- Compiled documentation
- Wrote ECE Wiki pages
- Adapted SHA-3 for MPW Shuttle 2.0 and wrote SHA-3 documentation
- Wrote and hardened Johnson counter within Openlane/Caravel harness (doesn't use I/O)
- Wrote Verilog tutorial
- Adapted SHA-3 algorithm for a smaller die size

Possible improvements for SHA-3

- Adapt to smaller die area
- Change number of stages in pipeline
- Offload padding etc. from ASIC
- Variable message size

Our Future Work

- Implementation of SHA-2 (256) hashing algorithm in a custom user area ASIC ontop of the Caravel Harness (Bitcoin/Crypto mining applications)
- Entrance into IEEE SSCS design contest

Our Future Work: SHA2-256

- Similar to SHA3-256 in nature (SHA)
- Still widely used today (coexists with SHA-3)
- Used in Bitcoin (ASIC)

SSCS "PICO" Open-Source Design Contest

- First year members only, pre-college and undergraduates
- Encouraged to use an SSCS circuit
- Can reuse open-source circuits
- chipIgnite: “pre-designed carrier chip” and “open-source EDA tools”

PICO timeline

- July 30: submit proposal
- 15-20 selected round 1 design teams
  - Weekly meetings until October
- Sept. 24: 6 designs selected
- Can still apply for shuttles

Key Takeaways

- Skywater PDK - maturing and useful
- Caravel test harness - good for small applications and education
- Caravel makes it easy to realize Verilog designs
- These tools provide a good testbench for samples of more complex circuits like hashing algorithms

More Information /Sources /References

- Caravel documentation: https://caravel-harness.readthedocs.io/en/latest/
- Openlane documentation https://openlane.readthedocs.io/en/latest/
- Jean Cyr https://github.com/miscellaneousbits/
- Verilog by Example - Blaine Readler
- Efabless - https://efabless.com/