The Journey of Two Novice LSI Enthusiasts: Tape-Out of CPU+RAM in Just One Month

Yuki Azuma / Kazuhide Uchiyama

RISC-V Day Tokyo 2023 Summer 2023/06/20

Self Introduction

Kazuhide Uchiyama, University of Electro-Communications,
Information Security Engineering Program @Cra2yPierr0t

I started building my own CPU to understand assembly language.

I have been an FPGA lover since then. I was researching a side-channel attack.

The abstract Chisato Nisikigi is my Internet name.

Self Introduction

Yuki Azuma, University of Tsukuba, Department of Physics @heppoko_yuki

I love to know how things work by building stuff from scratch. ex) OS, CPU

Recent work is accelerating the simulation code of cosmic fluid dynamics in my belonging lab.

Workshop 1

* If you encounter menus do not work upon clicking, delete your browser's cache.

Open Source PDKs and EDAs, Community Experiences toward Democratization of Chip Design

Organizer : Makoto Ikeda (The University of Tokyo)å
Mehdi Saligane (University of Michigan)

Since its launch in 2020, the Open MPW shuttle program has received over 500 project submissions spanning 9 shuttles. This workshop will explore various topics related to designers' experiences, including measured results, foundry perspectives, and governmental expectations.

About Makoto Ikeda
Makoto Ikeda received his BE, ME, and Ph.D. degrees all in EE department of the University of Tokyo in 1991, 1993, and 1996, respectively. He is now a full professor at d.lab, the University of Tokyo. This workshop is co-organized with Dr. Mehdi Saligane of University of Michigan.

| 1. Design experience: "The Journey of Two Novice LSI Enthusiasts: Tape-Out of CPU+RAM in Just One Month", Kazuhide Uchiyama, University of Electro-Communications and Yuki Azuma, University of Tsukuba

| 2. From Zero to 1000 Open Source Custom Designs in Two Years, Mohamed Kassem, Co-founder and CTO, Efabless

| 3. The SKY130 Open Source PDK: Building an Open Source Innovation Ecosystem, Steve Kosier, Skywater technology

| 4. Open Source Chip Design on GF180MCU – A foundry perspective, Karthik Chandrasekaran, Global foundries

| 5. Japan Foundries' Perspectives on Silicon design democratization, Shiro Hara, Minimal Fab & AIST

| 6. Google's perspective on Open source PDKs, Open source EDA tools, and OpenMPW shuttle program, Johan Euphrosine and Tim Ansell, Google

| 7. The Nanofabrication Accelerator Project, Matthew Daniels, NIST

| 8. Japanese government perspective on Silicon design democratization, Yohei Ogino, The Ministry of Economy, Trade and Industry METI

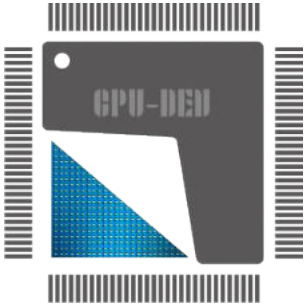
We are CPU Enthusiast

* CPU is BLACK-BOX

* We hate BLACK-BOX

* Shine the light and unravel BLACK-BOX!

* cpu-dev: The Group of CPU Enthusiasts



Application

Operating System

Computer Architecture

Micro Architecture

Logic

Digital Circuit

Analog Circuit

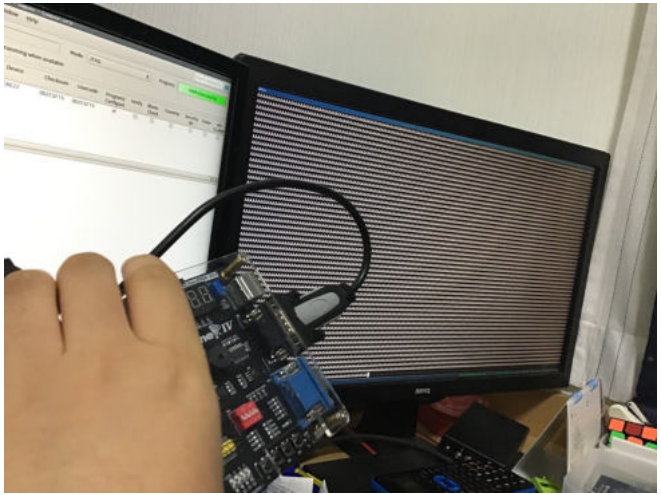
Device

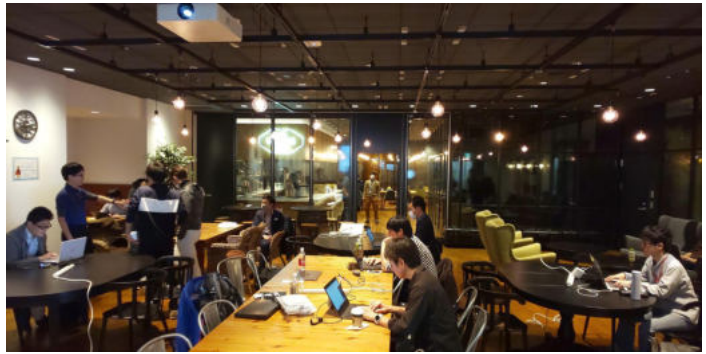
Physics

We are CPU Enthusiast

We used FPGA to make CPU then.

But FPGA couldn’t satisfy us, LSI is the truth.





cpu-dev group

Jacaranda-8: our original architecture

Tiny architecture

Implements interrupt

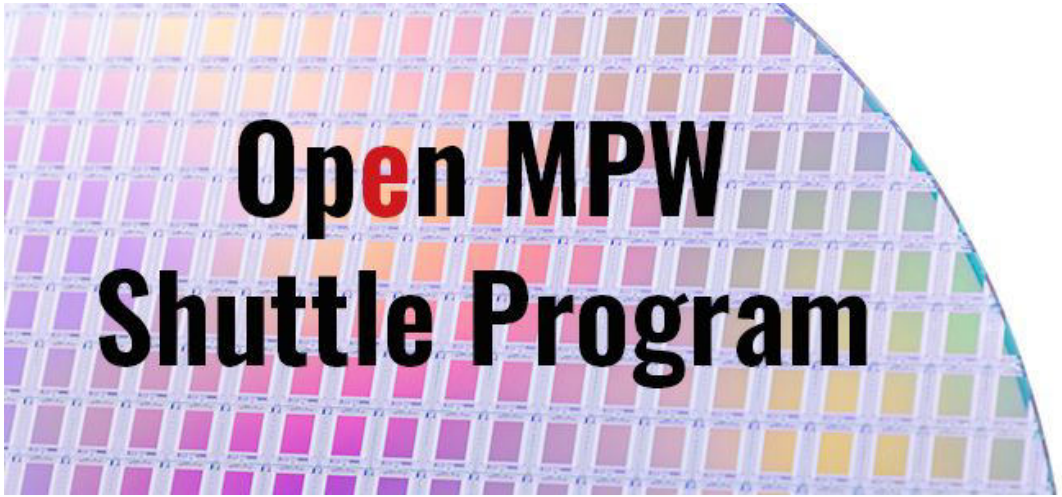
mov	add	and	or
not	sll	srl	sra
cmp	je	jmp	ldih
ldil	ld	st	iret

Jacaranda (fake Paulownia)
桐擬き

https://github.com/cpu-dev/cpu-dev.github.io/wiki/jacaranda-8_ISA_draft

Encounter with Open MPW Shuttle Program

- There's something called Open MPW, apparently.
- They say we can make LSI with it for free.
- Why not join!



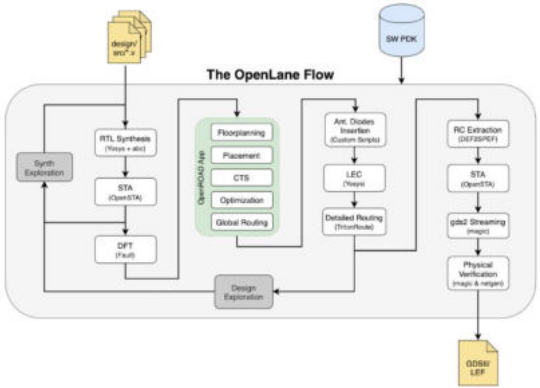
efabless.com

Sponsored by

Google

Open MPW Shuttle Program

- The shuttle program started by **Efabless**
- Skywater 130nm or GlobalFoundries 180nm
- No production and shipping costs
  - 224 WCSP chips and 2 evaluation boards
- All tools and PDK are open source
  - Skywater PDK and OpenLANE
- The 40 entries that pass the test among those submitted are randomly selected



PDK Announced

As part of the FOSSi Foundation's first talk in the dial up talk series, Tim 'mithro' Ansell announced the first, manufacturable 130nm PDK.

Submission deadline for the first run of the no cost MPW shuttle program.

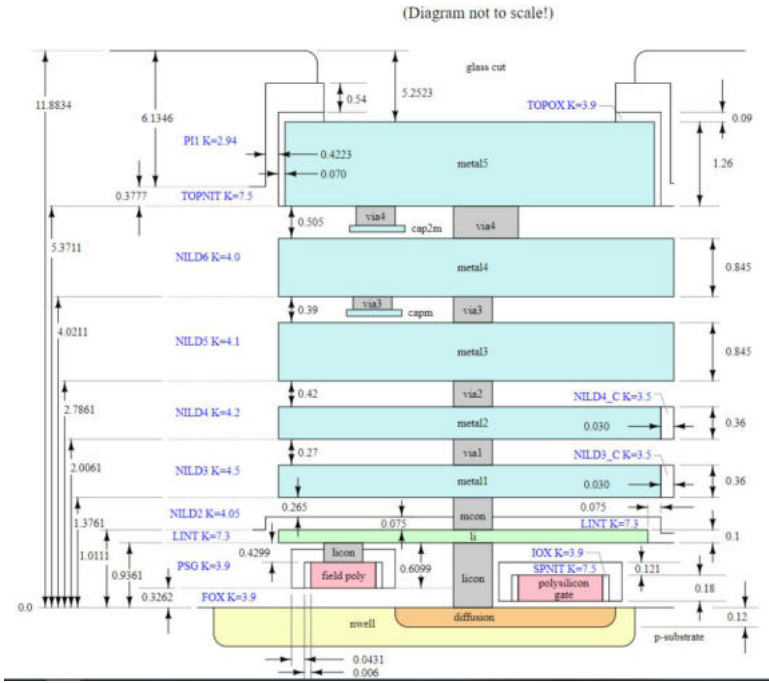
MPW-ONE

MPW-TWO

Submission deadline for the first run of the no cost MPW shuttle program.

sky130(skywater 130nm process)

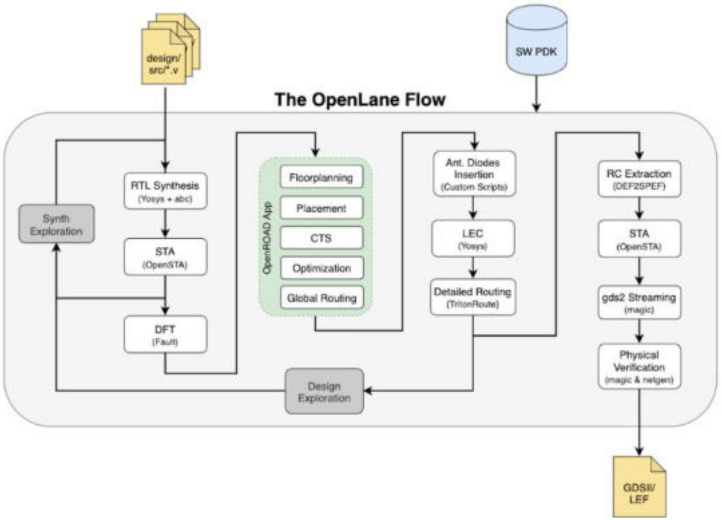
- Developed by Cypress and manufactured by Skywater
- Began to be used around 2001
- 2020 PDK released by Google and Skywater



OpenLANE Overview

OpenLANE

- RTL-to-GDSII compiler made from a combination of about 20 OSS
- It's all put together in a Docker container
- Write Verilog and config and run to generate GDSII



Configuration of OpenLANE

- Read the documentation and write the TCL

```
16 set script_dir [file dirname [file normalize [info script]]]
17
18 set ::env(DESIGN_NAME) user_proj_example
19
20 set ::env(VERILOG_FILES) "\
21     $::env(CARAVEL_ROOT)/verilog/rtl/defines.v \
22     $script_dir/../../verilog/rtl/user_proj_example.v"
23
24 set ::env(DESIGN_IS_CORE) 0
25
26 set ::env(CLOCK_PORT) "wb_clk_i"
27 set ::env(CLOCK_NET) "counter.clk"
28 set ::env(CLOCK_PERIOD) "10"
29
30 set ::env(FP_SIZING) absolute
31 set ::env(DIE_AREA) "0 0 900 600"
32
33 set ::env(FP_PIN_ORDER_CFG) $script_dir/pin_order.cfg
34
35 set ::env(PL_BASIC_PLACEMENT) 1
36 set ::env(PL_TARGET_DENSITY) 0.05
37
```

path to verilog files

clock port

clock period

die size

config.tcl

Floorplanning

Variable	Description
FP_CORE_UTIL	The core utilization percentage. (Default: 50 percent)
FP_ASPECT_RATIO	The core's aspect ratio (height / width). (Default: 1)
FP_SIZING	Whether to use relative sizing by making use of FP_CORE_UTIL or absolute one using DIE_AREA. (Default: "relative" - accepts "absolute" as well)
DIE_AREA	Specific die area to be used in floorplanning. Specified as a 4-corner rectangle "x0 y0 x1 y1". Units in um (Default: unset)
FP_IO_MODE	Decides the mode of the random IO placement option. 0=matching mode, 1=random equidistant mode

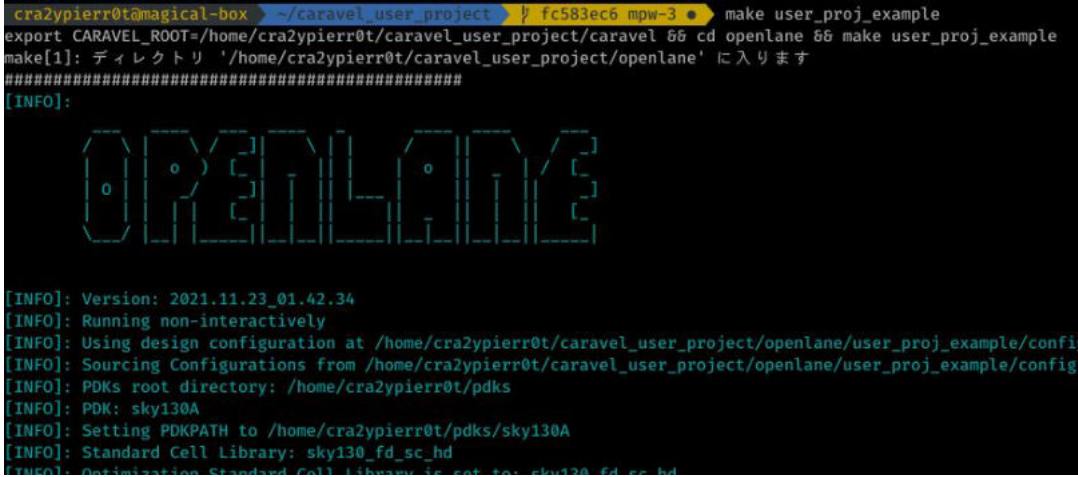
Placement

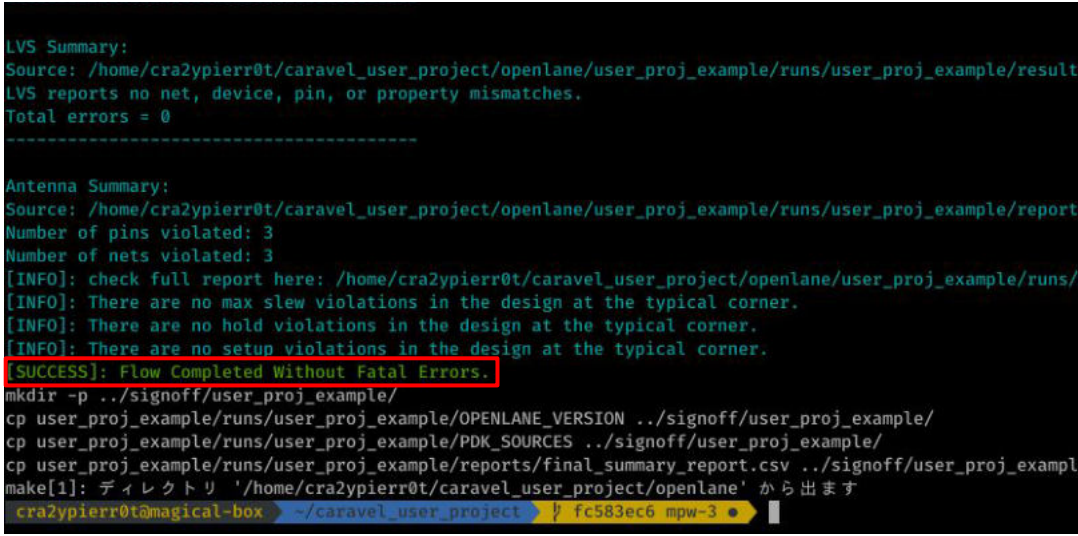
Variable	Description
PL_TARGET_DENSITY	The desired placement density of cells. It reflects how spread the cells would be on the core area. 1 = closely dense. 0 = widely spread (Default: 0.55)
PL_TIME_DRIVEN	Specifies whether the placer should use time driven placement. 0 = false, 1 = true (Default: 0)
PL_LIB	Specifies the library for time driven placement (Default: LIB_TYPICAL)
PL_BASIC_PLACEMENT	Specifies whether the placer should run basic placemer not (by running initial placement, increasing the minimu overflow to 0.9, and limiting the number of iterations to 0 = false, 1 = true (Default: 0)

Run OpenLANE

- OpenLANE is executed with $ make <design name>

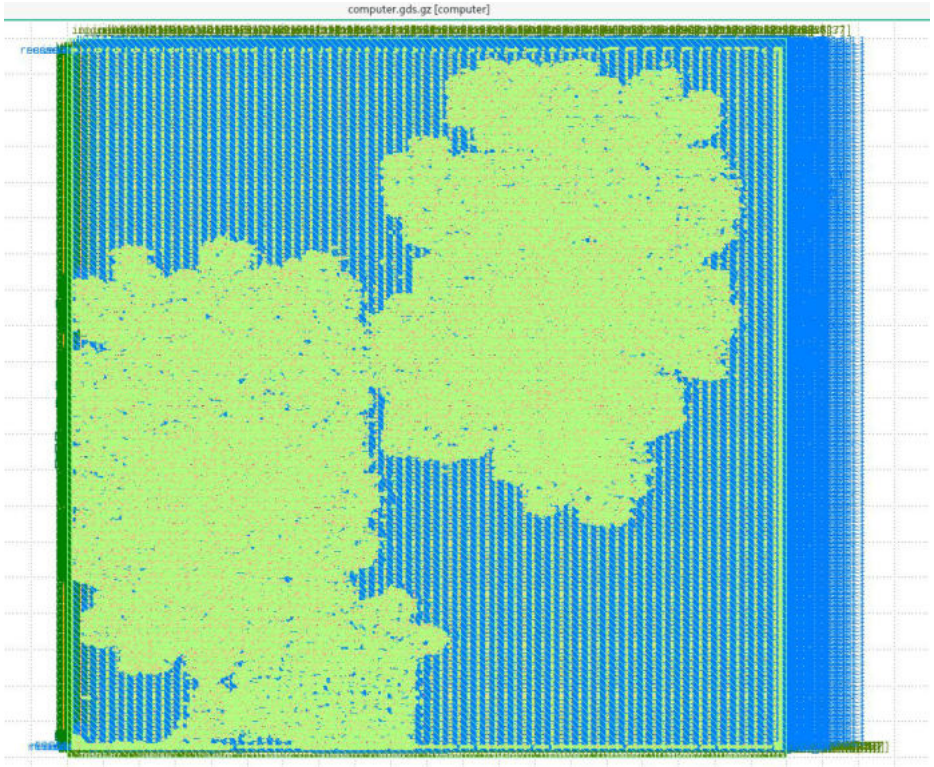
- If successful, GDSII is generated with the analysis results





Generating GDSII

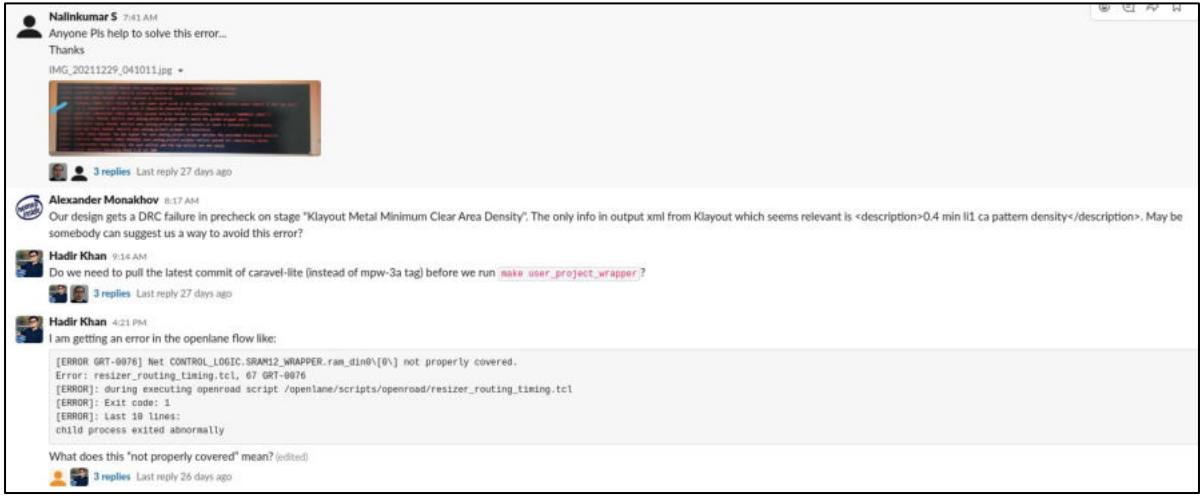
- Takes about 5 hours depending on the design
- Open MPW requirement is to be reproducible from the repository



Generated GDSII

Open Source Silicon Slack

- 5676 members, 154 channels
- Community for OpenMPW questions and EDA tools OSS
- Supreme Developers Community



Nalinkumar S 7:41 AM
Anyone Pls help to solve this error...
Thanks
IMG_20211229_041011.jpg

3 replies Last reply 27 days ago

Alexander Monakhov 8:17 AM
Our design gets a DRC failure in precheck on stage "Klayout Metal Minimum Clear Area Density". The only info in output xml from Klayout which seems relevant is <description>0.4 min li1 ca pattern density</description>. May be somebody can suggest us a way to avoid this error?

Hadir Khan 9:14 AM
Do we need to pull the latest commit of caravel-lite (instead of mpw-3a tag) before we run `make user_project_wrapper`?

3 replies Last reply 27 days ago

Hadir Khan 4:21 PM
I am getting an error in the openlane flow like:

```
[ERROR GRT-0076] Net CONTROL_LOGIC.SRAM12_WRAPPER.ram_din0\[0\] not properly covered.
Error: resizer_routing_timing.tcl, 67 GRT-0076
[ERROR]: during executing openroad script /openlane/scripts/openroad/resizer_routing_timing.tcl
[ERROR]: Exit code: 1
[ERROR]: Last 10 lines:
child process exited abnormally
```

What does this "not properly covered" mean? (edited)

3 replies Last reply 26 days ago

#general
✓ Joined · 2,374 members · General discussion channel around the Google SkyWat

#sky130
✓ Joined · 2,371 members · General discussion channel around the Google SkyWat

#announcements
✓ Joined · 945 members

#shuttle
✓ Joined · 525 members · mpw2 tags: OpenLANE: v0.15, caravel_user_project: mp

#openlane
✓ Joined · 643 members · Discussion about OpenLANE: An automated RTL to GDS

#analog-design ← analog circuit channel
766 members · Channel for discussion about designing mixed-signal integrated circui

#caravel
✓ Joined · 283 members · discussion on the Caravel test harness

#openram ← OpenRAM channel
✓ Joined · 313 members · Discussion channel for using the OpenRAM open-source

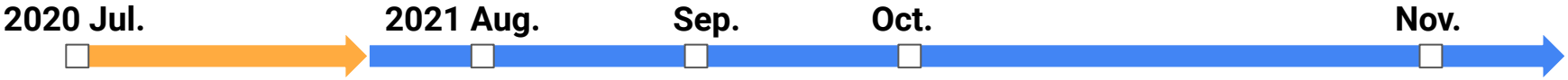
#efabless
✓ Joined · 291 members · Discussion channel for using the SKY130 process on the

#magic
359 members

One month to tapeout

Tapeout timeline

☆Death March to tapeout☆



7/11 OpenMPW Announced

8/13 Kazu wrote Japanese article of OpenMPW Introduction

9/20 MPW-3 Announced

10/4 Yuki and Kazu decided to submit our project

10/29 Deadline extended

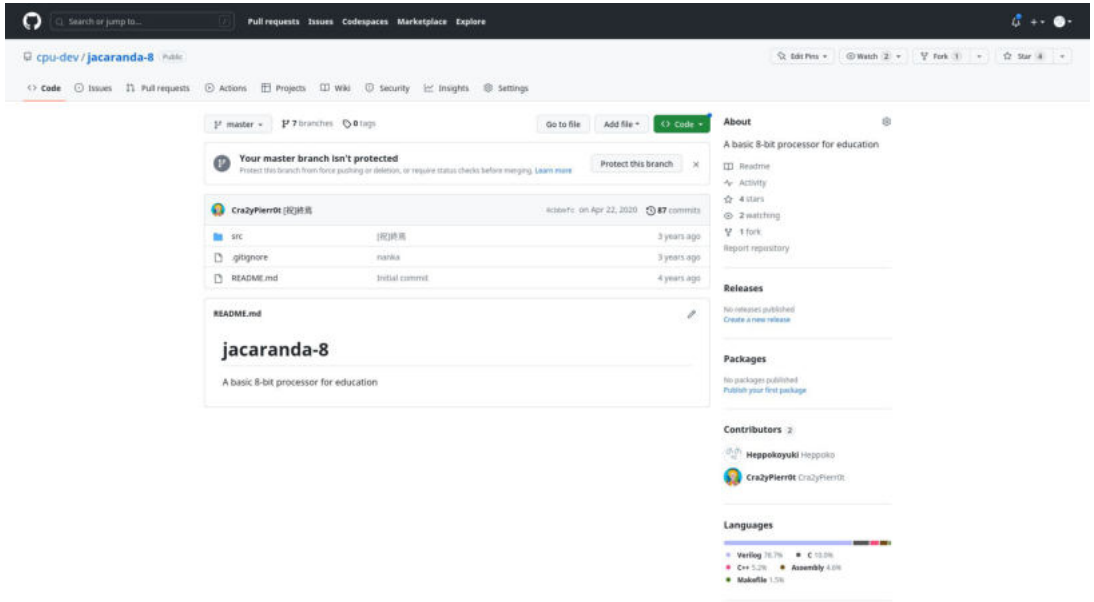
11/11 1:34 Tapeout test succeeded!

11/15 MPW-3 new deadline

2023 Jun 10

Our chips have arrived!

Day 0



We already had CPU HDL design then.

- **Verilog design**
- RTL Synthesis
- Floorplanning
- Placement
- Static Timing Analysis
- GDS2 design
- Precheck
- Tapeout check
- LSI chip

VLSI.JP

To Article Index

- OpenMPW入門 改訂版
  - OpenMPWとは
  - 宣伝
  - 環境構築
  - OpenLANEの概要
    - OpenLANEのインストール
    - OpenLANEの動作確認
    - OpenLANEの設定
  - Caravelの概要
    - Caravelのインストール
    - Caravelの動作確認
    - Caravelのファイル構造
    - Caravelにおけるデザインのビルド
    - Caravelのドキュメント
  - OpenMPWで自分のデザインを焼こう！
    - 1. 焼きたいデザインを用意する
    - 2. 自分のデザインにCaravel用のインターフェースを生やす
      - WishboneとLogic Analyzerの違い
      - Wishbone interfaceを作る
      - Logic Analyzerを扱う
      - mprj_ioを扱う
      - 割り込みの扱い方
    - 3. 自分のデザインをGDSIIにする
      - 設定ファイルを書く
      - GDSIIを生成
    - 4. user_project_wrapperに自分のデザインを加える
      - 追記: user_defines.vを編集
    - 5. シミュレーションで動作を確認する
      - シミュレーション環境のインストール
      - 使うファイルを指定する
      - ファームウェアを書く
      - テストベンチを書く
      - Makefileをコピーする
      - シミュレーションを実行する
    - 6. OpenLANEの設定をする
      - CLOCK_NETの変更
      - CLOCK_PERIODの変更
      - FP_PDN_MACRO_HOOKSの

OpenMPW入門 改訂版

質問、修正案、その他連絡は@Cra2yPierr0tマデ

久々に以前書いた入門記事を読んだら日本語は雑だし内容は古いしこの世の終わりみたいな出来だったので90nmに備えて書きます。

オレオレLSIを焼きたいですよね？ 焼きましょう。 Skywater社がPDKを公開し、OSSなGDSIIコンパイラである**OpenLANE**も生まれました。そしてGoogleの出資によってEfablessが無料でLSIを作らせてくれるプログラム、**Open MPW Shuttle Program**をスタートしました。 今こそオレオレLSIを焼くチャンスです。あなたの作りたいチップをGoogleの金で作りましょう！

OpenMPWとは

OpenMPW(Open Multi Project Wafer)は**Efabless**にGoogleが出資して生まれたシャトルプログラムであり、ホームページには次の文言が書かれています。

The shuttle provides opportunities for designers to experiment and push the state-of-the-art without having to reconcile the risk associated with the cost of fabrication. The shuttle program is open to anyone, provided that their project is fully open source and meets the other program requirements. Costs for fabrication, packaging, evaluation boards and shipping are covered by Google for this program.

出典：https://efabless.com/open_shuttle_program

このシャトルはデザイナーに製造コストに纏わるリスクを負うことなく、実験し、最先端を追求する機会を提供します。 シャトルプログラムは、プロジェクトが完全にオープンソースであり、一定の要件を満たしていれば、誰でも参加することができます。 製造、パッケージング、評価ボード、そして送料は全てGoogleが負担します。

つまりデザインをオープンソースにすれば**完全無料**で自分の半導体を作れるプログラムという事ですね。これは熱い、参加するしかない、Google最高一生ついていきます。

と言っても提出された全てのデザインを焼いてくれるという訳ではなく、いくつかの条件が存在しています。

- 提出された中からランダムに40のプロジェクトが選ばれる
- デザインをオープンソースにすること
- レイアウトをリポジトリから再現可能にすること
- ライセンス諸々
- Caravel(後述)の使用
- プリチェックツールをパスしていること

今すぐにでも自分のHDLをLSIにしたい気持ちは分かりますが、必要な事がちょっと多いので少しずつやっていきましょう。

この次は各ツールの概要に入りますが、既に知っておりインストール済みなら飛ばして頂いて構いません。

宣伝

高位合成ツール、RgGenを使ったWalkthorughが生えました！

RgGen × OpenMPWでLSIを焼こう！

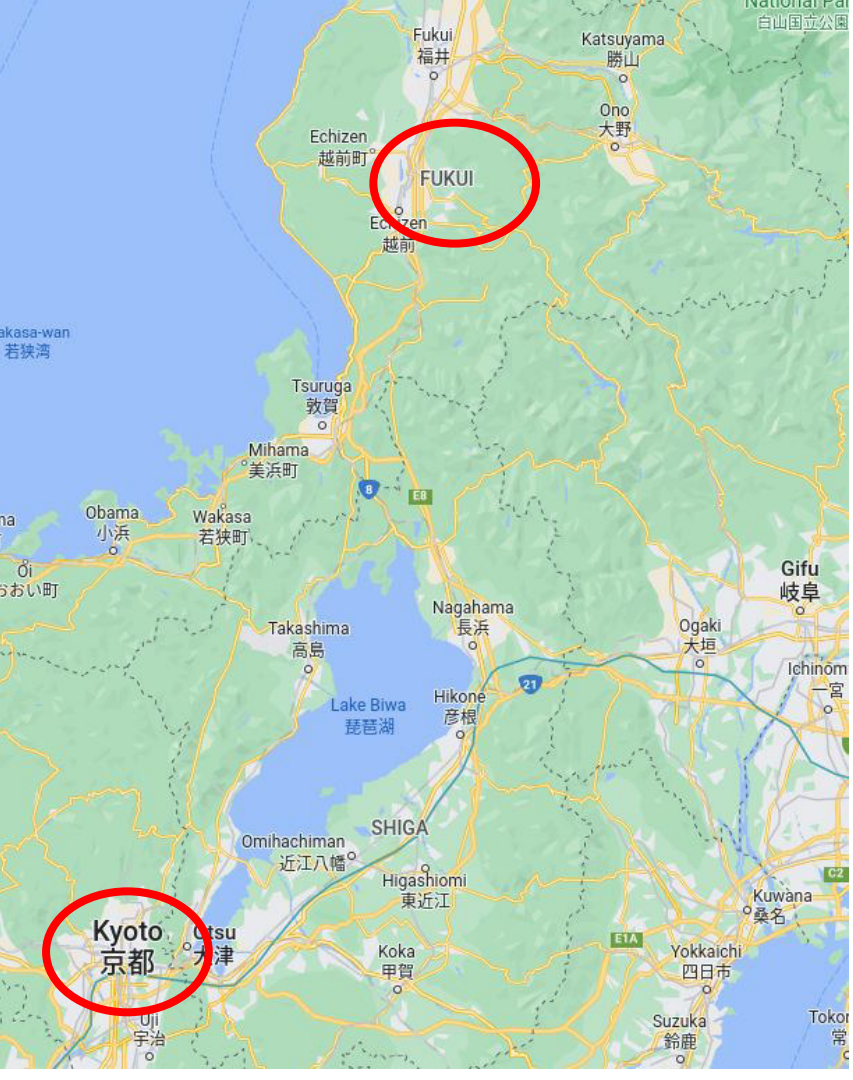
環境構築

まずはOpenMPWの開発を始めるための最低限の環境を構築しましょう。Linux/M1 Mac/Intel

Day 0

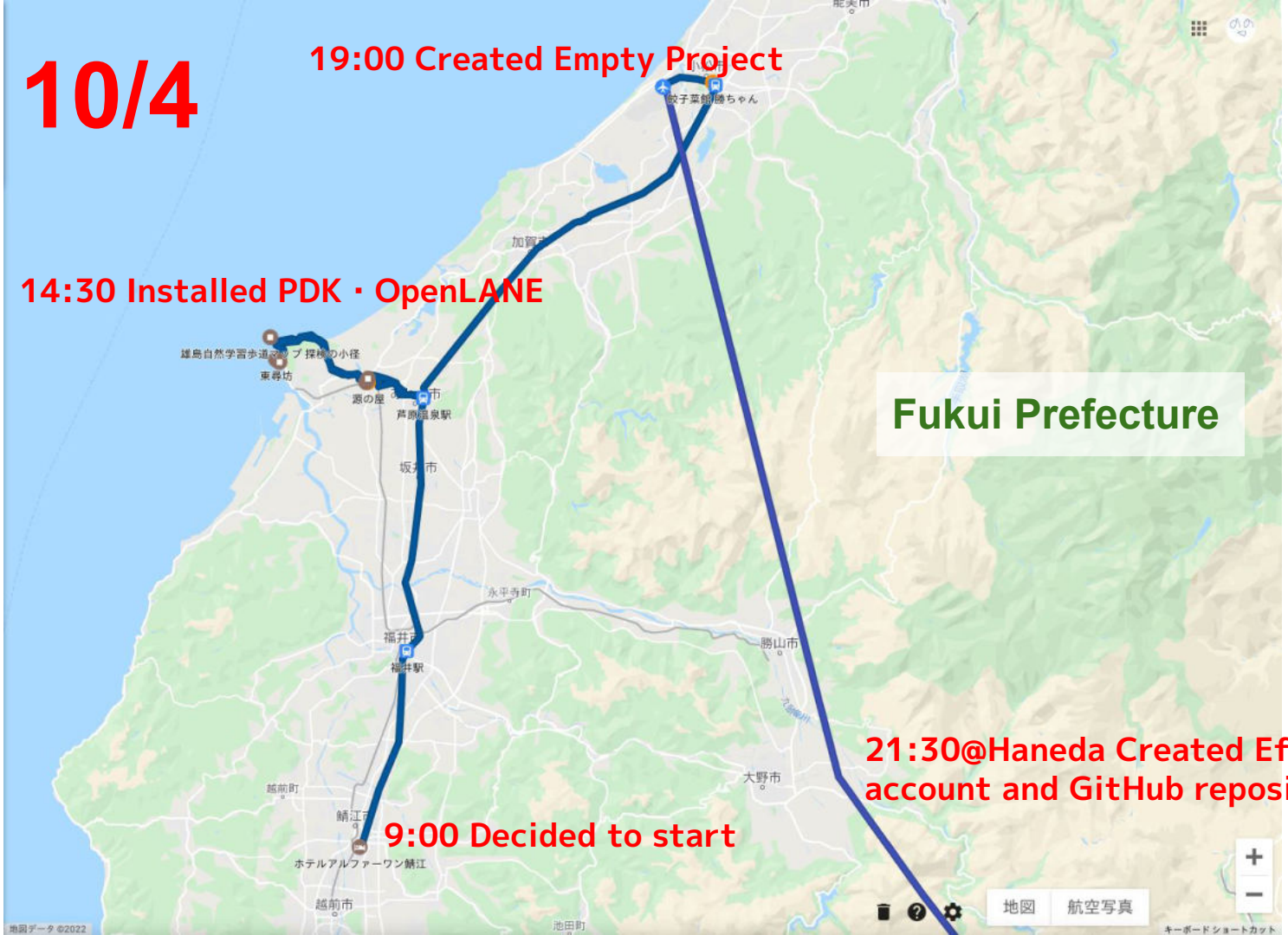
When Kazu knew the OpenMPW.
He was so fascinated.

No Japanese, No Knowledge
➜ VLSI.JP !



Day 1

Our journey had started in Fukui.

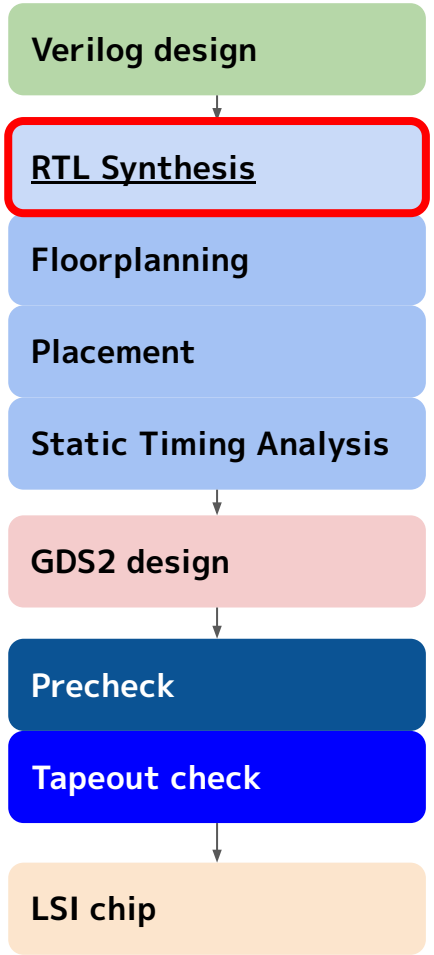

10/4
19:00 Created Empty Project
14:30 Installed PDK・OpenLANE
Fukui Prefecture
21:30@Haneda Created Efabless account and GitHub repository
9:00 Decided to start
餃子菜館勝ちゃん
東尋坊
源の屋
芦原温泉駅
福井駅
勝山市
大野市
越前町
鯖江市
ホテルアルファーワン鯖江
越前市
地図
航空写真

Oshima Island in Fukui: The beginning

~Day 9

Jacaranda-8 couldn't be synthesized as it was.

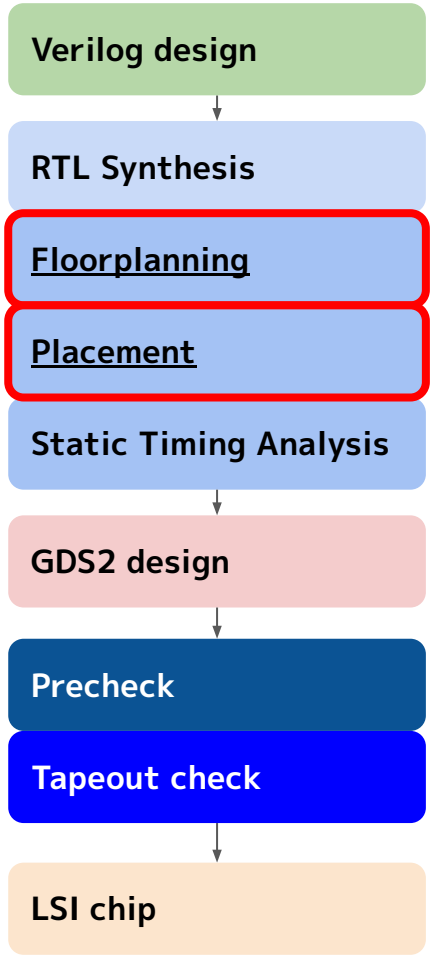
To enable synthesis, the battle with Yosys began...



Day 10

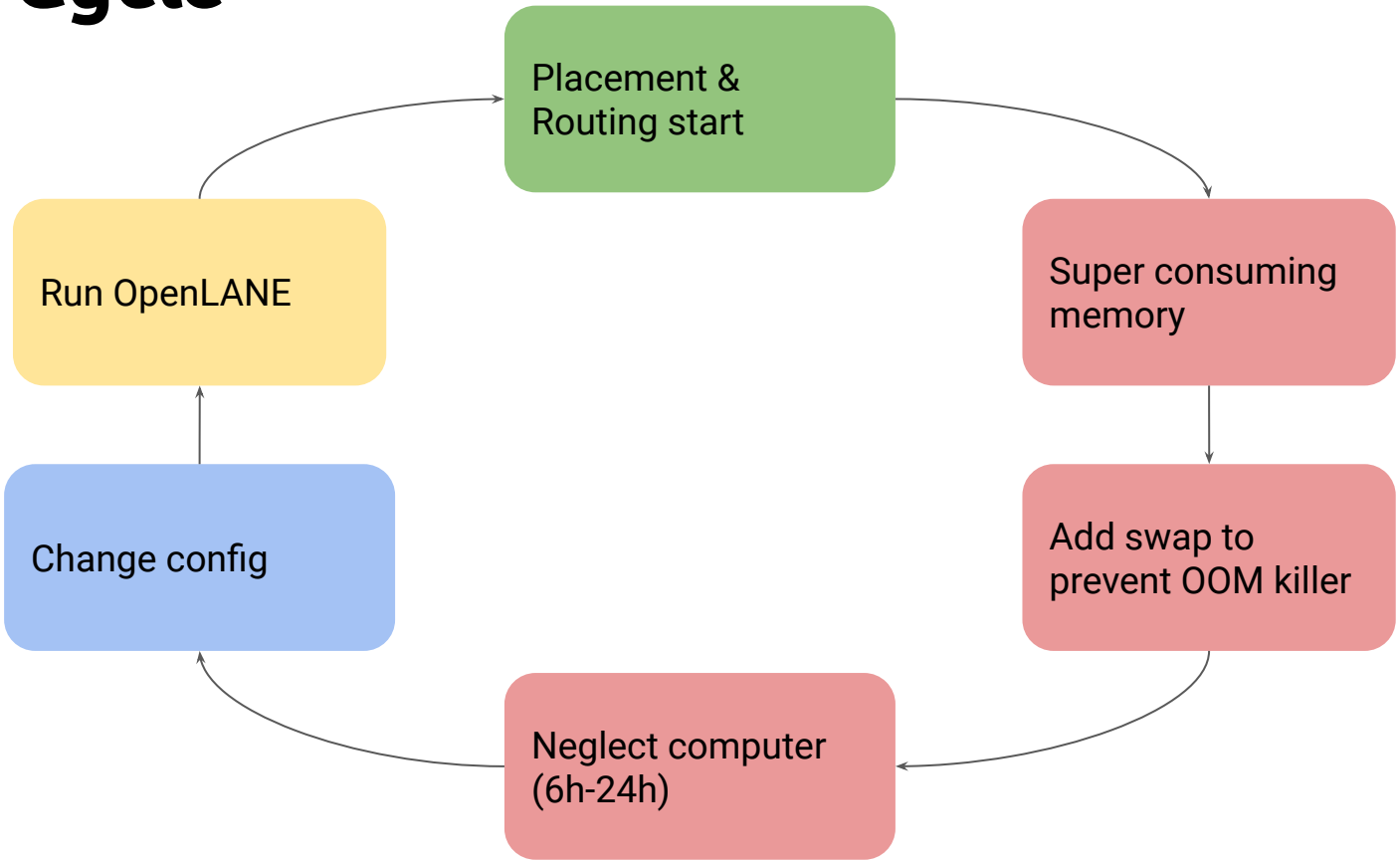
Ultimately, synthesis succeed.

Next battle target is... OpenROAD



OUT OF MEMORY!!!

Build Cycle


Placement &
Routing start
Super consuming
memory
Add swap to
prevent OOM killer
Neglect computer
(6h-24h)
Change config
Run OpenLANE

Day 11

SUPER Long Building time: 8 hours!

3 build/Day

Days remaining x 3

= Number of Build opportunities

Day 11

Integrate other macro was impossible...

> we didn't have knowledge and no documents about macro placement.

Generating SRAM by Auto-Routing

Wanted to use OpenRAM

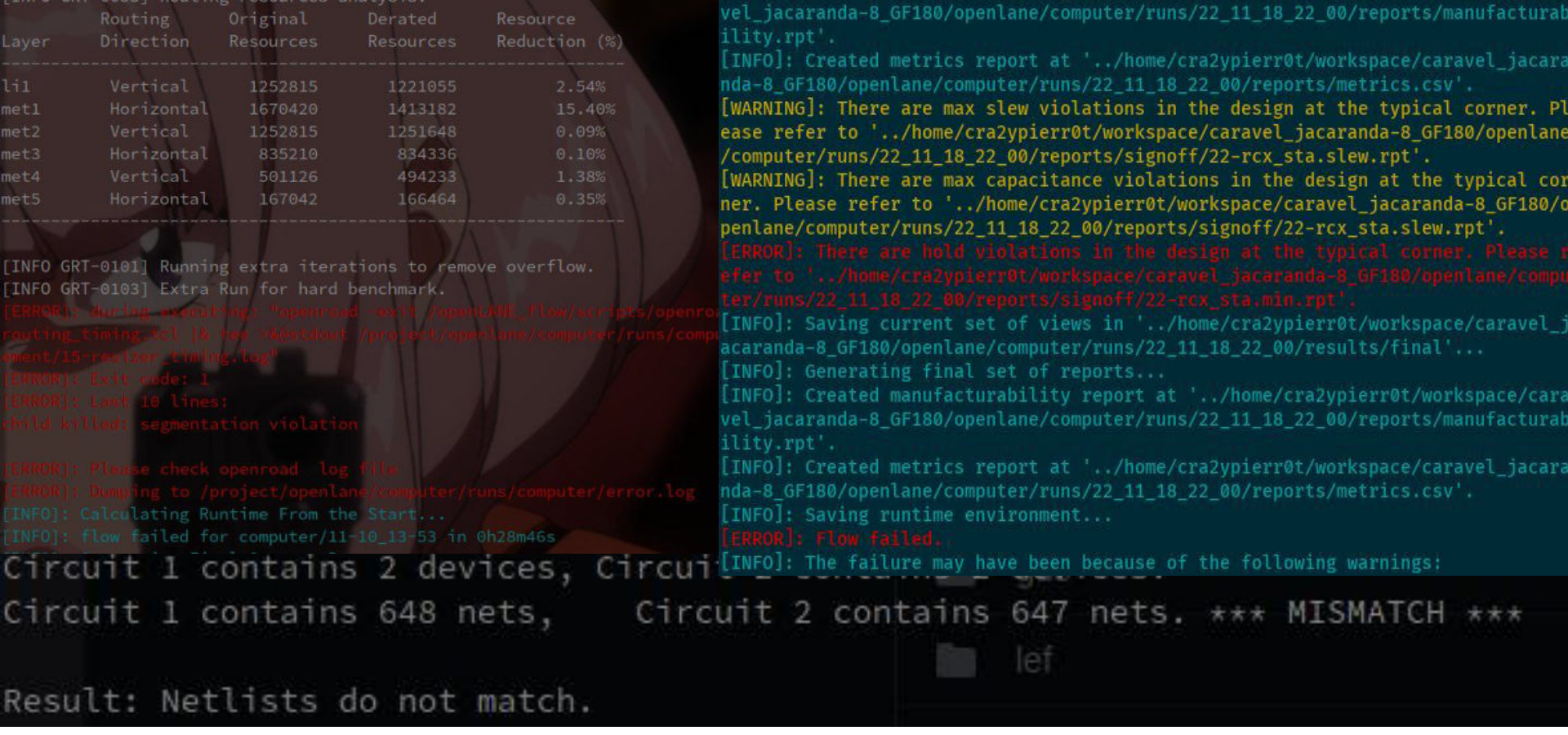
```
module data_mem(addr, w_data, w_en, r_data, clock);
    input [7:0] addr;
    input [7:0] w_data;
    input w_en;
    input clock;
    output [7:0] r_data;

    reg [7:0] mem[0:255];

    assign r_data = mem[addr];

    always @(posedge clock) begin
        if(w_en) begin
            mem[addr] <= w_data;
        end else begin
            mem[addr] <= mem[addr];
        end
    end
endmodule
```

Novice's RAM (Causes long build time)



FACING MANY ERRORS!!!

Day 12 ~ 20

Configuration Battle, We are beginner

```
set ::env(CLOCK_PORT) wb_clk_i
set ::env(CLOCK_NET) wb_clk_i
set ::env(CLOCK_PERIOD) 500

#set ::env(SYNTH_STRATEGY) "DELAY 2"

set ::env(PL_TARGET_DENSITY) 0.40
set ::env(FP_SIZING) absolute
set ::env(DIE_AREA) "0 0 1000 1000"
#set ::env(FP_CORE_UTIL) 6
#set ::env(FP_SIZING) relative

set ::(DECAP_CELL) "sky130_ef_sc_hd__decap_12"

set ::env(GLB_RESIZER_TIMING_OPTIMIZATIONS) 0
set ::env(PL_RESIZER_TIMING_OPTIMIZATIONS) 1
set ::env(PL_RESIZER_HOLD_SLACK_MARGIN) 0.2
set ::env(PL_RESIZER_ALLOW_SETUP_VIOS) 1
set ::env(QUIT_ON_HOLD_VIOLATIONS) 0
set ::env(GLB_RT_ADJUSTMENT) 0.30

set ::env(FP_PIN_ORDER_CFG) $script_dir/pin_order.cfg

set ::env(VDD_NETS) [list {vccd1}]
set ::env(GND_NETS) [list {vssd1}]

set ::env(DIODE_INSERTION_STRATEGY) 4
```

What is this

What is this

What is this

What is this

What is this

What is this

What is this

What is this

What is this

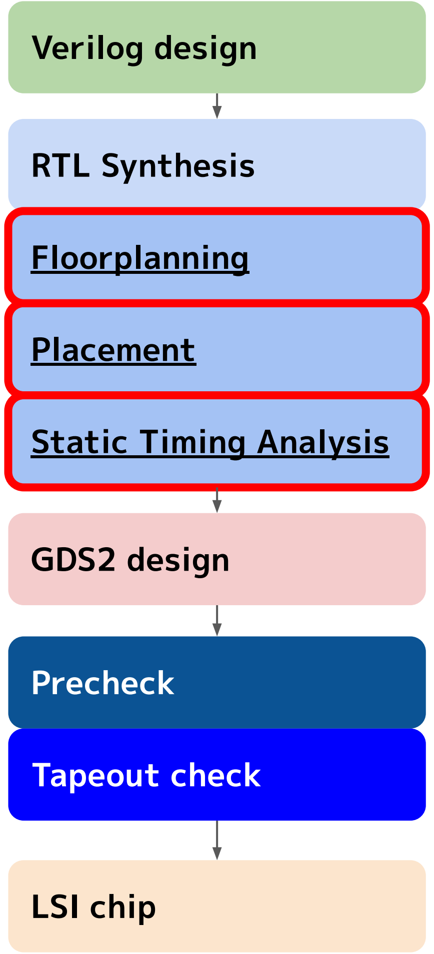
What is this

What is this

What is this

What is this

What is this

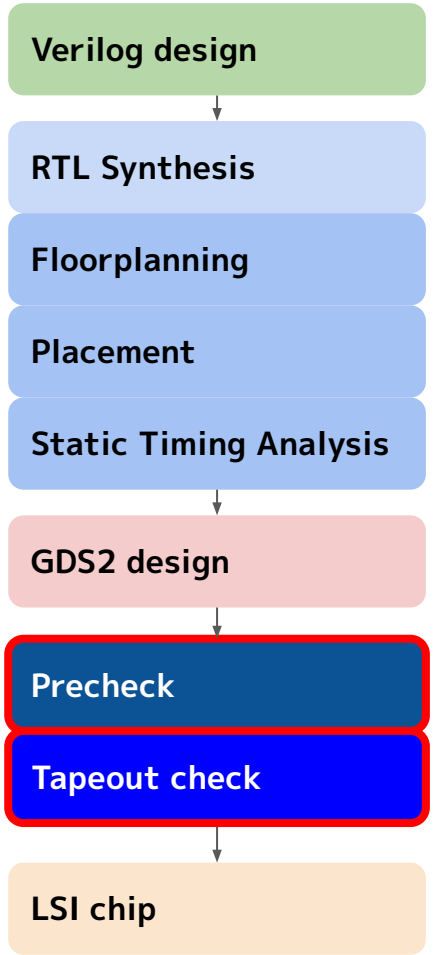


Day 21 ~ 30

Job Logs for 50b594c (eda7f068-4225-4a6e-9c6e-a2e073f7a53b)

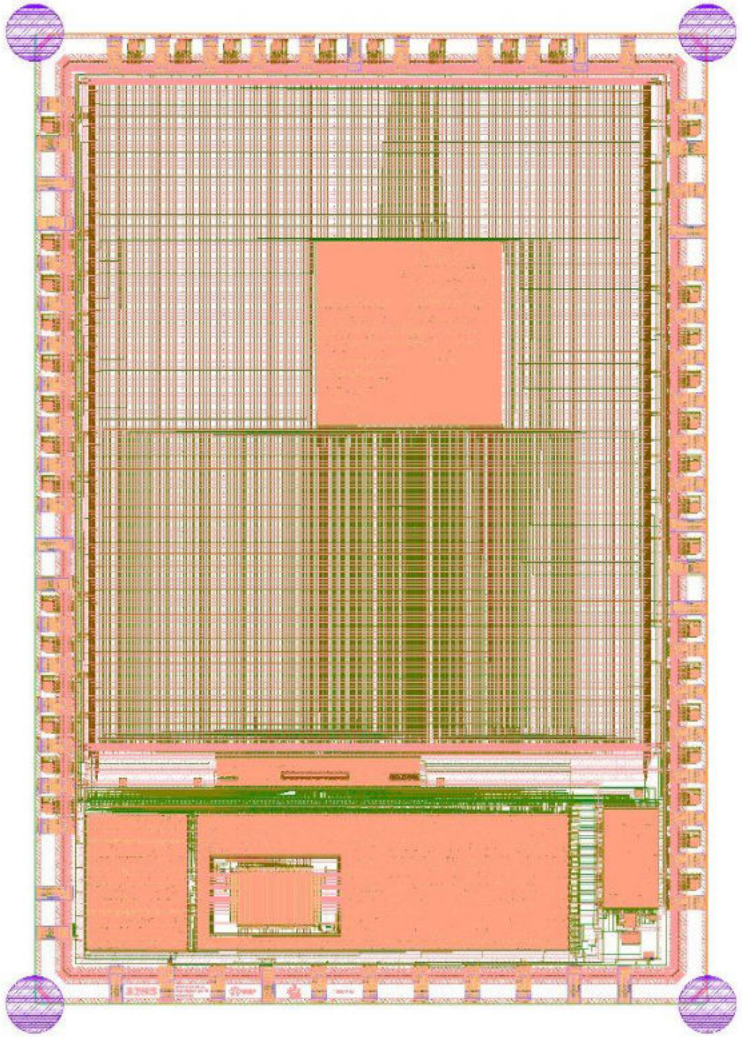
```
STDOUT: {{Project GDS Info}} user_project_wrapper: cddff5d8278d72b42fbbb990d31e1be59fd2284f
STDOUT: {{START}} Tapeout Started
STDOUT: {{Step Update}} Generating Final Layout: Step 1 of 5
STDOUT: {{Step Update}} Generating Final Layout: Step 2 of 5
STDOUT: {{Step Update}} Generating Final Layout: Step 3 of 5
STDOUT: {{Step Update}} Generating Final Layout: Step 4 of 5
STDOUT: {{Step Update}} Generating Final Layout: Step 5 of 5
STDOUT: Final GDS File caravel_0003ba28.gds sha1sum: b72ae30649d440f1e3167c353928214419920da0
STDOUT: {{Step Update}} Converting Final Layout from GDS to OAS
STDOUT: Final OAS File caravel_0003ba28.oas sha1sum: 4f9db9a6f297343ff940fba967e867665ab0ea32
STDOUT: {{Step Update}} Executing Check 1 of 2: Klayout Metal Minimum Clear Area Density
STDERR: [ WARN ] MET Density Check Result: GDS has 1 DRC violations.
STDOUT: {{Step Update}} Executing Check 2 of 2: Klayout Field Oxide Mask Density
STDERR: [ WARN ] FOM Density Check Result: GDS has 0 DRC violations.
STDOUT: {{FINISH}} Executing Finished, the full logs can be found in u6145_yuhkiya/design/caravel_jacaranda-8/jobs
/tapeout/eda7f068-4225-4a6e-9c6e-a2e073f7a53b/logs
```

Mysterious Tapeout Check



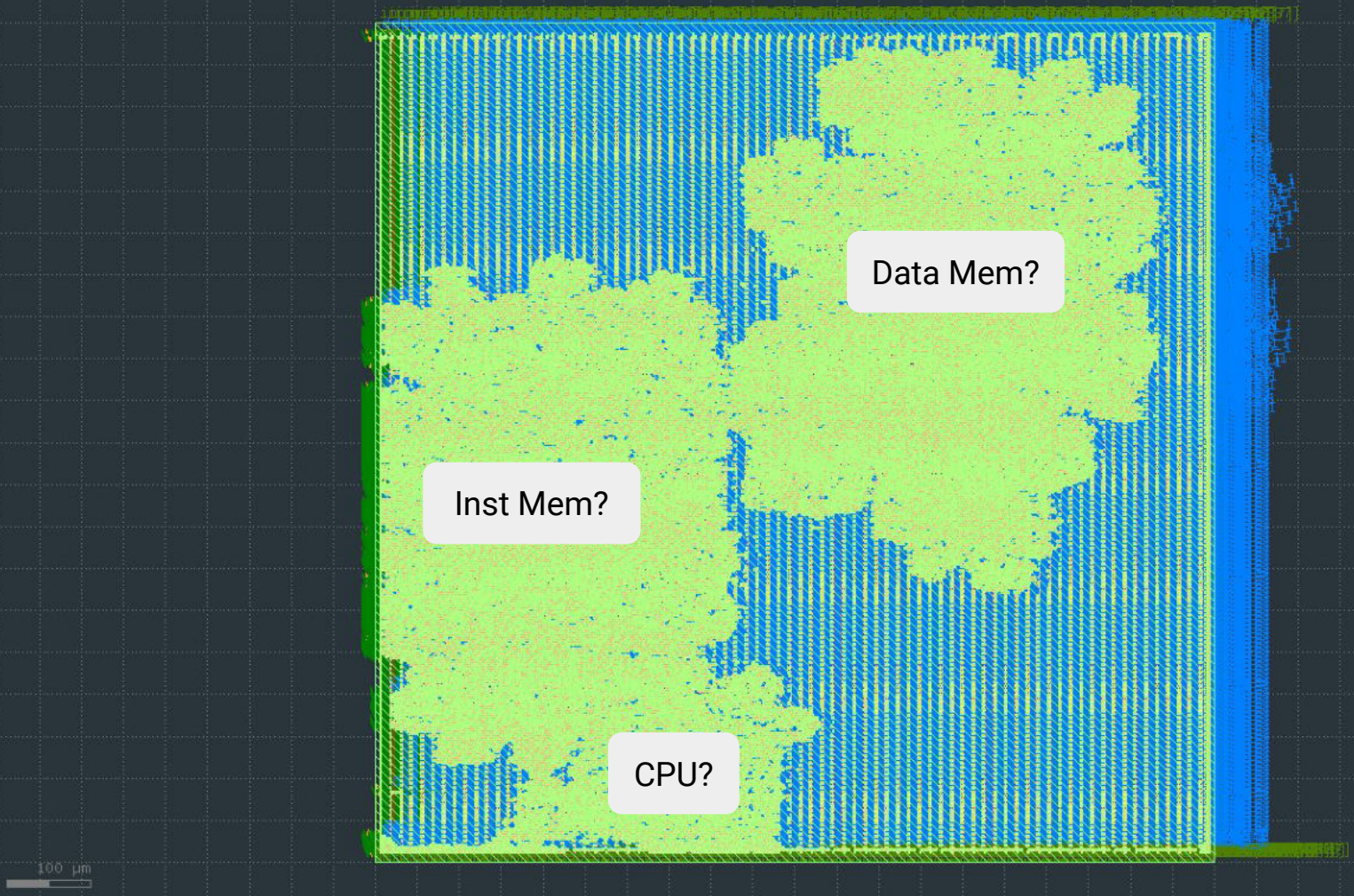
DO

完



NE

成

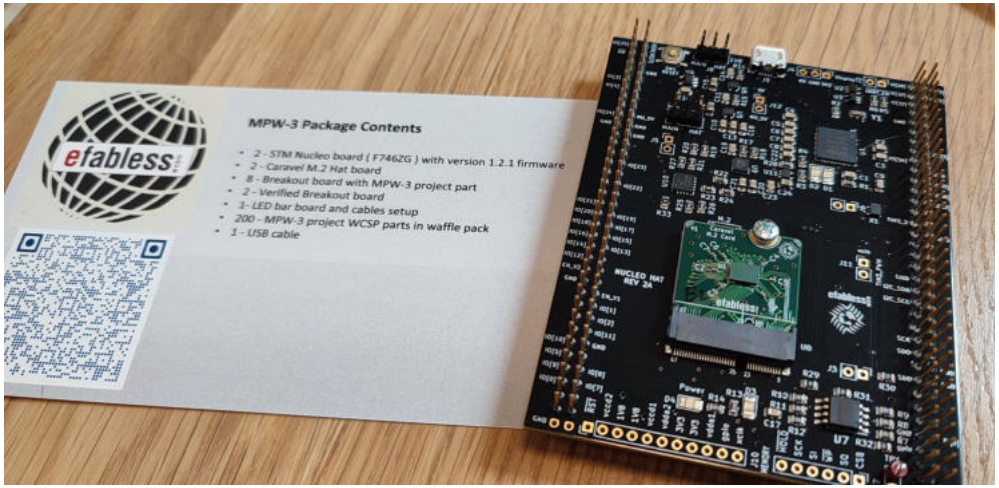

Data Mem?
Inst Mem?
CPU?
100 µm

Δyuk!
@heppoko_yuki

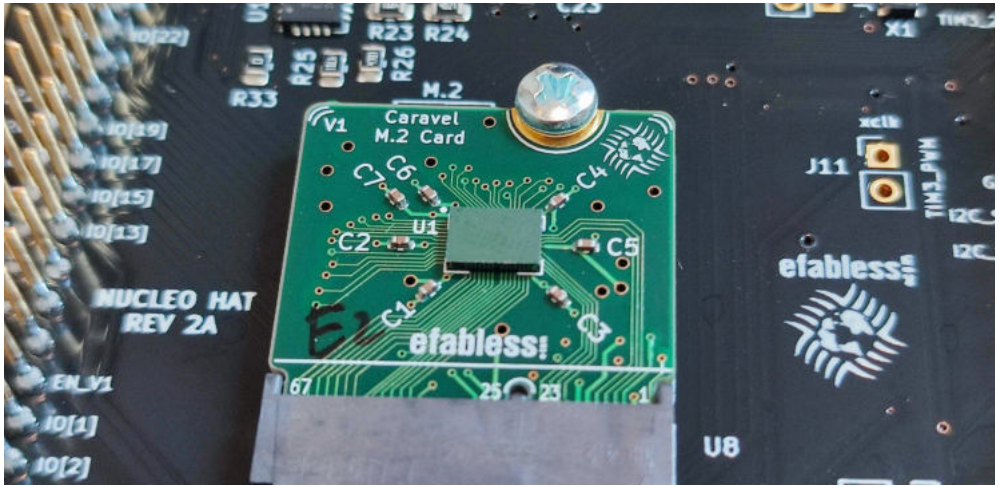
Jacaranda-8到着しました！
Jacaranda-8 has arrived!

#efabless #openMPW

9:12 AM · Jun 10, 2023 · **3,544** Views

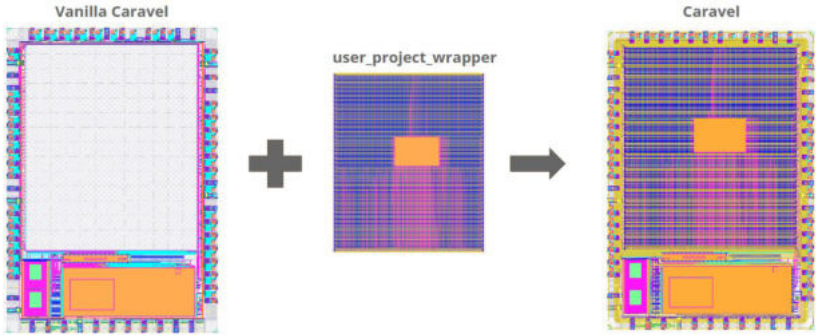

efabless
MPW-3 Package Contents
2 - STM Nucleo board (F746ZG) with version 1.2.1 firmware
2 - Caravel M.2 Hat board
8 - Breakout board with MPW-3 project part
2 - Verified Breakout board
1- LED bar board and cables setup
200 - MPW-3 project WCSP parts in waffle pack
1 - USB cable

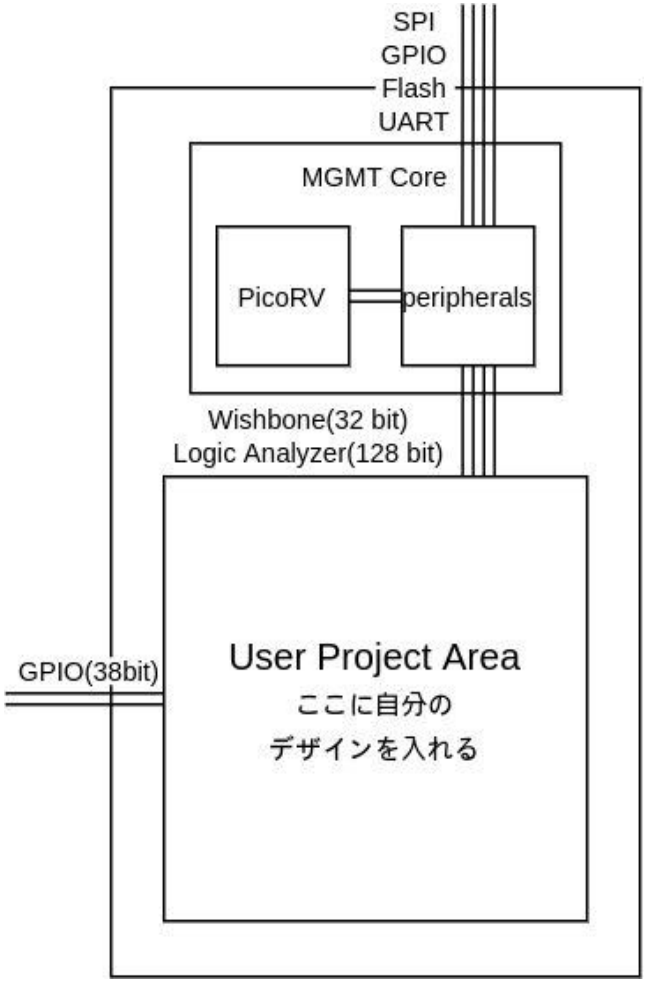

EFABLESS
PN#: E2
LOT#: 4206521
WAFER#: 01 – 12
QTY: 56
DATE: 2249
JOB#: 31431
Syagrus
EFABLESS
PN#: E2
LOT#: 4206521
WAFER#: 01 – 12
QTY: 56
DATE: 2249
JOB#: 31431
Syagrus
EFABLESS
PN#: E2
LOT#: 4206521
WAFER#: 01 – 12
QTY: 56
DATE: 2249
JOB#: 31431
Syagrus


M.2
Caravel
M.2 Card
V1
R23
R24
R25
R26
R33
C6
C7
C4
C2
C5
U1
J11
X1
efabless
NUCLEO HAT
REV 2A
U8

Caravel Harness

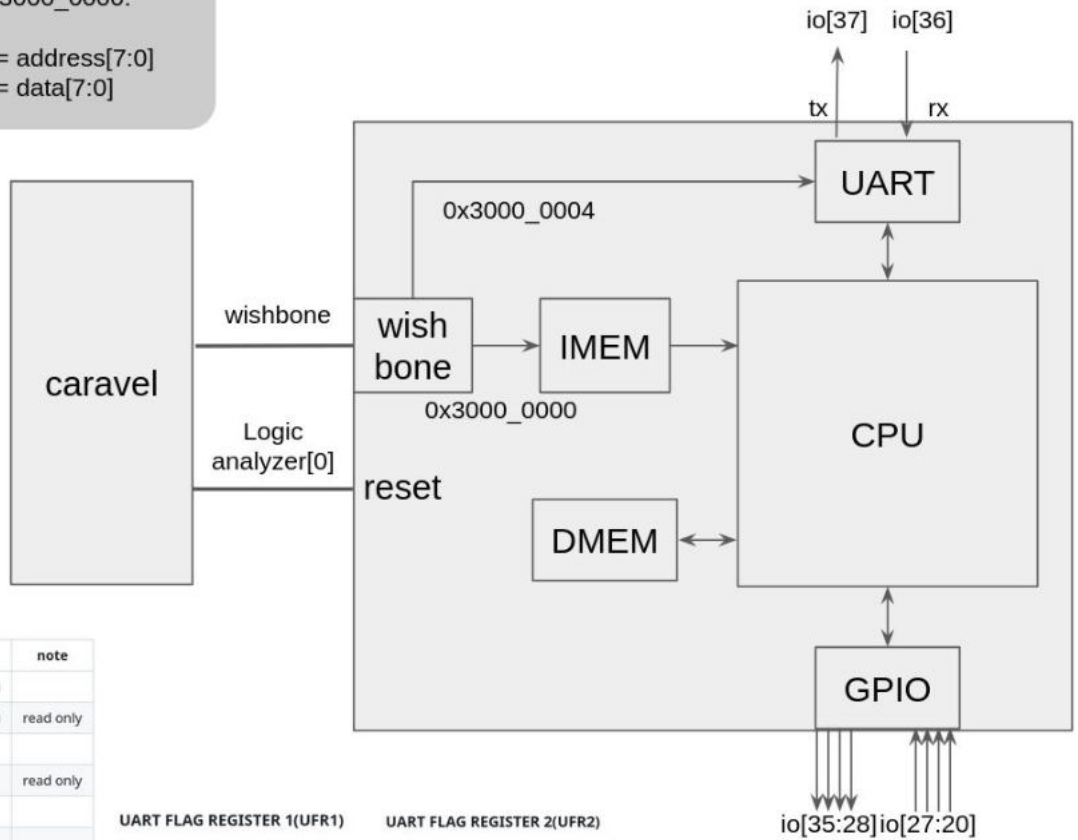
- Management SoC of the OpenMPW design
- We can connect it by wishbone and logic analyzer wire
- Simulation with caravel is also provided.
    - Simulation including MSoC behavior is available





instruction write on 0x3000_0000:

wishbone_data[15:8] = address[7:0]
wishbone_data[7:0] = data[7:0]



address	description	note
255	UART FLAG REGISTER 1(UFR1)	
254	UART FLAG REGISTER 2(UFR2)	read only
253	UART TX DATA(UTD)	
252	UART RX DATA(URD)	read only
251	GPIO OUT	
250	interrupt vector	
249	GPIO IN	read only

UART FLAG REGISTER 1(UFR1)

7-1	1	0
reserved	rx_en	tx_en

UART FLAG REGISTER 2(UFR2)

7-1	1	0
reserved	receive_flag	busy_flag

```
ldi 01
mov r0, r3
ldi ff
st r3, r0        // [0xff] ← 0x01 ; set status register
ldi 41
mov r0, r3
ldi fd
mov r1, r3
ldi fe
ld r2, r3
ldi 01
and r2, r3
cmp r2, r3
ldi 0c
je r3            // while([0xfe] == 0x01); ; wait until tx ready
st r1, r0        // set 'A' to TX register
ldi 06
jmp r3           // loop
```

```c
void
main()
{
    #include "instr.c"
    gpio_config_io();

    reg_gpio_mode1 = 1;
    reg_gpio_mode0 = 0;
    reg_gpio_ien = 1;
    reg_gpio_oe = 1;

    reg_gpio_out = 1;

    reset();

    // set reset to high
    reg_la0_data = 1;
    reg_mprj_xfer = 1;
    while (reg_mprj_xfer == 1);

    // set uart clk frequency
    write(UART_CLK_FREQ, 10000000);

    for(int i = 0; i < 29; ++i) {
        write(IMEM_WRITE, i << 8 | mem[i]);
    }

    reg_la0_data = 0;
    reg_mprj_xfer = 1;
    while (reg_mprj_xfer == 1);

    while(1) {
        reg_gpio_out = 0x0;
        delay(1000000);
        reg_gpio_out = 0x1;
        delay(1000000);
    }
}
```


1 2.00V/ 2
113.0s 10.00s/ Stop
1 1.18V
KEYSIGHT TECHNOLOGIES
Back
Cursors
Mode
Manual
Source
1
Cursors
X1
Units
X1: 87.800000us X2: 98.000000us Y1: -4.00000V Y2: 4.00000V
ΔX: +10.200000us 1/ΔX: +98.039kHz ΔY(1): +8.00000V
Auto Scale
Default Setup
Entry
Push to Select
Intensity
1


Fn
Ctrl
RYZEN
7

Summary

* Finally, we could make our LSI without knowledge of semiconductor engineering
* Open Source Silicon slack members helped us much
* Fundamental documentation for newbie is not enough
* Welcome to #japan-region

Expectations for OpenSourceSilicon

* **JAPAN!! COME ON!!**

* Please write documentation more friendly for Software Engineer

* We expect for widespread use in the field of computational science!

    expecting next system like GRAPE (N-body sim accelerator)

* Please give us more recent PDK!

Acknowledgement

* **Efabless, Skywater, OpenSourceSilicon Slack, Google**

Thank you for pleasant opportunities and support for us!

* **Shumpei Kawasaki-san and RISC-V Days Tokyo**

Thank you for inviting to precious Presentation time!

*** Kristopher Tate-san and Takahiro Kinoshita-san**

Thank you for rent an oscilloscope

* **And you…**

We wish you a pleasant RISC-V Day and
Good luck with the democratized LSI future!