OpenFASOC: Automated Open Source Analog and Mixed-Signals IC Generation

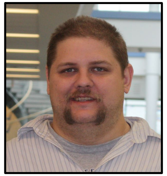
Mehdi Saligane
Research Scientist
University of Michigan

MICL MICHIGAN IN
CIRCUITS LA
UNIVERSITY OF M

FASoC: Fully-Autonomous SoC Synthesis

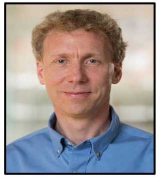
- DARPA IDEA Program
- Multi-University and Industry effort

D. Wentzloff



R. Dreslinski

D. Sylvester



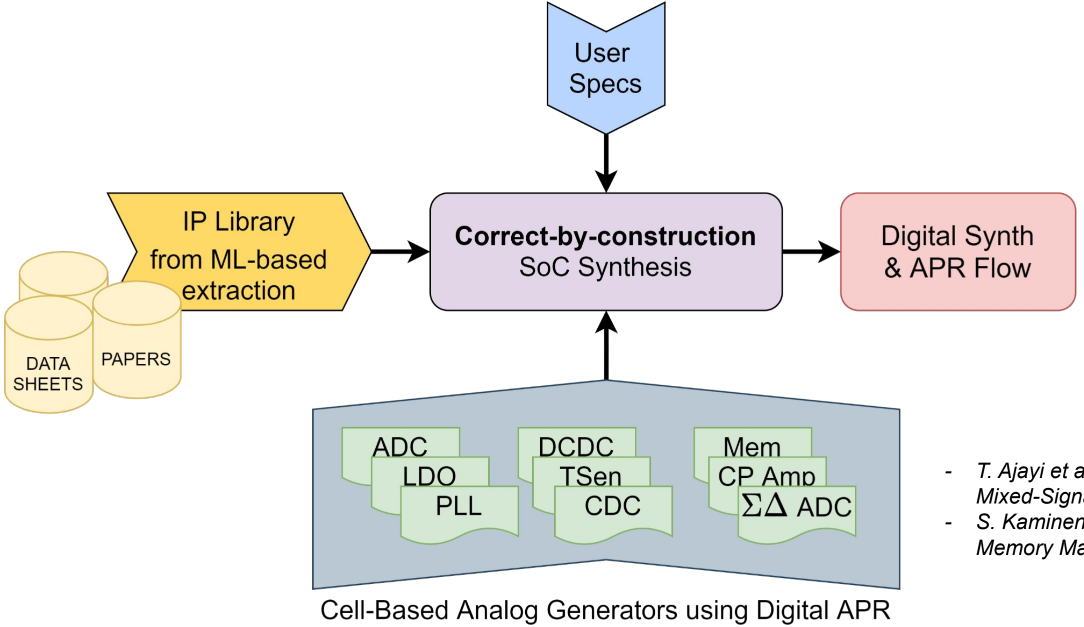
D. Blaauw

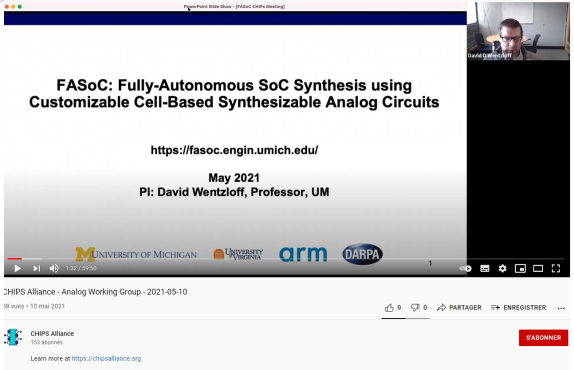
B. Calhoun

M. Coltella

FASoC: Fully-Autonomous SoC Synthesis

- Correct-by-construction SoC design leveraging IP-XACT and Arm Socrates
- Analog generation tools for xDC, PLL, SRAM, DCDC, temp sense, CP Amp, ΣΔ ADC


User Specs
IP Library from ML-based extraction
DATA SHEETS
PAPERS
Correct-by-construction SoC Synthesis
Digital Synth & APR Flow
ADC
LDO
PLL
DCDC
TSen
CDC
Mem
CP Amp
ΣΔ ADC
Cell-Based Analog Generators using Digital APR


PowerPoint Slide Show - [FASoC CHIPs Meeting]

FASoC: Fully-Autonomous SoC Synthesis using Customizable Cell-Based Synthesizable Analog Circuits

https://fasoc.engin.umich.edu/

May 2021
PI: David Wentzloff, Professor, UM

CHIPS Alliance - Analog Working Group - 2021-05-10

59 vues • 10 mai 2021

Learn more at https://chipsalliance.org

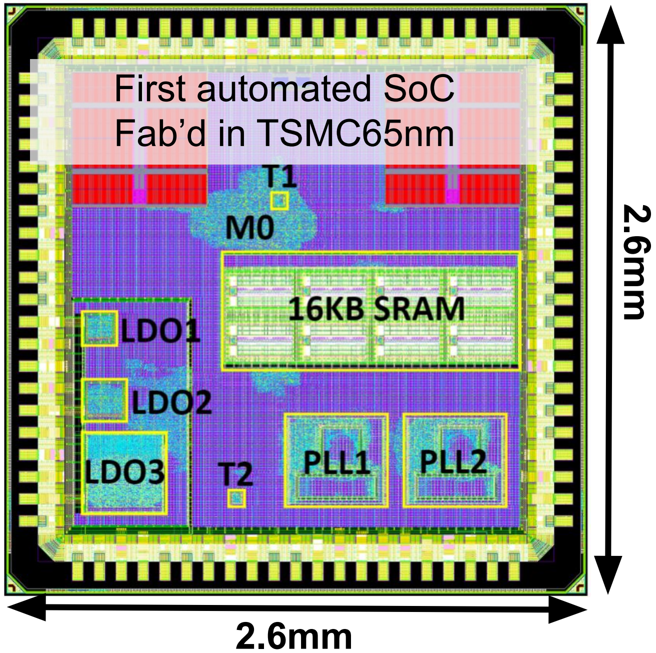
[CHIPS Alliance - Analog Working Group - 2021-05-10]

https://fasoc.engin.umich.edu/

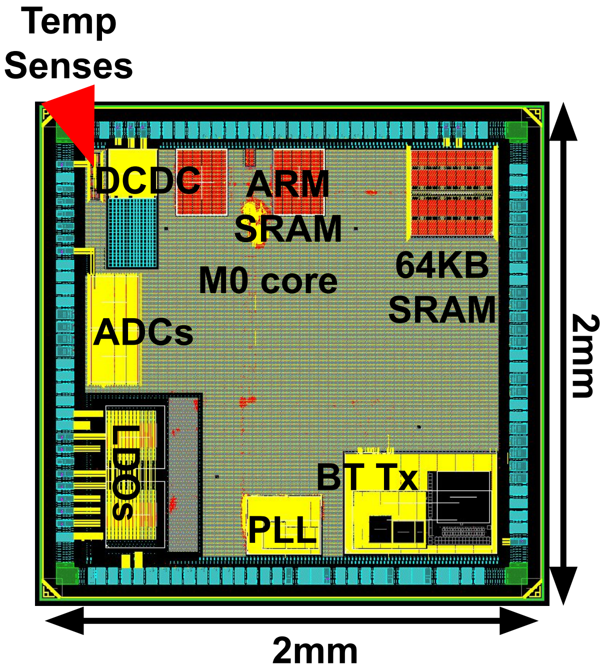
- T. Ajayi et al, “Fully-Autonomous SoC Synthesis Using Customizable Cell-Based Analog and Mixed-Signal Circuits Generation”, IFIP/IEEE VLSI SOC
- S. Kamineni et al, "MemGen: An Open-Source Framework for Autonomous Generation of Memory Macros," 2021 IEEE CICC, 2021

FASoC SoCs in TSMC 65 and GF12LP

- Multiple tape-outs in TSMC 65, GF12LP, SkyWater 130nm

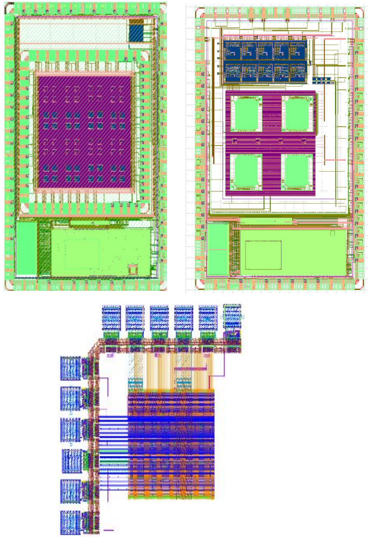


TSMC65LP SoC (2019-08)

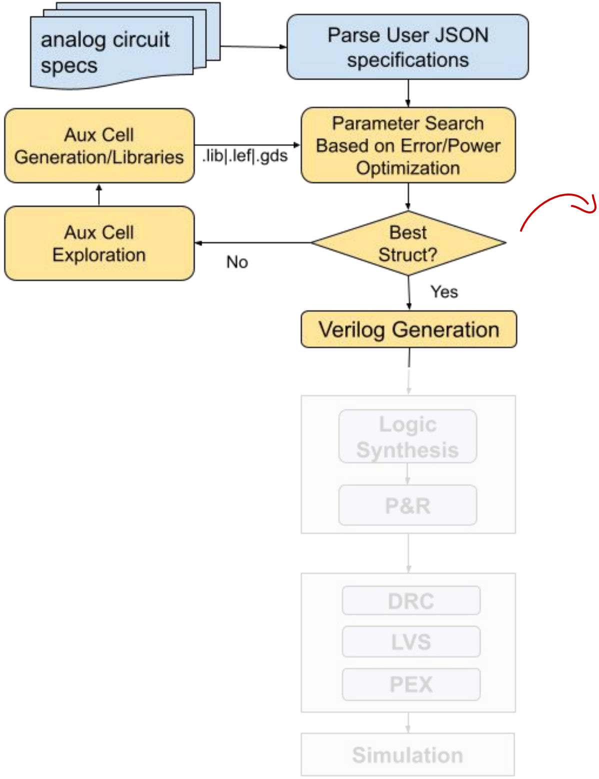


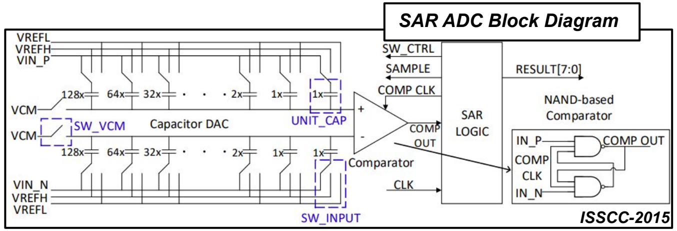
GF12LP SoC (2020-10)

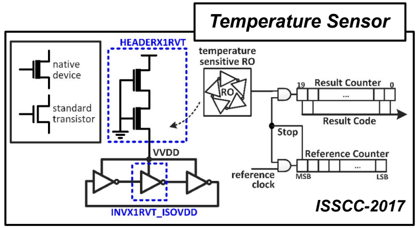
- GF12LP - 12nm FinFET
- GF 8HP - 130nm BiCMOS
- SKY130 - 130nm Bulk

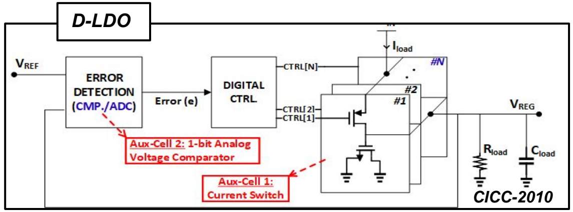


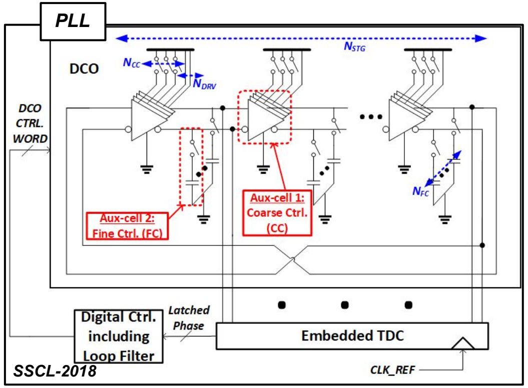
Our Cell-Based Approach to Analog Design



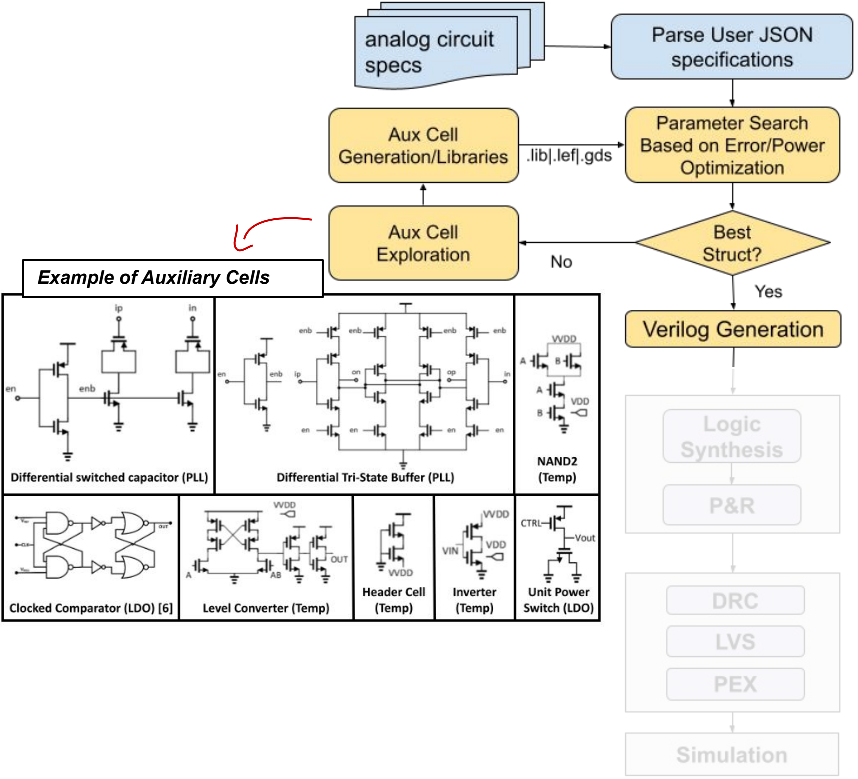


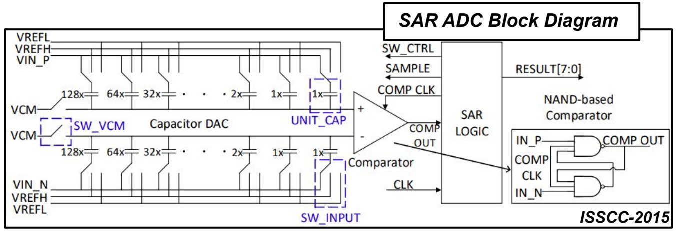


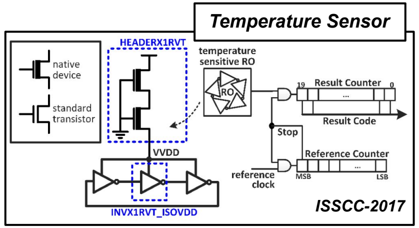


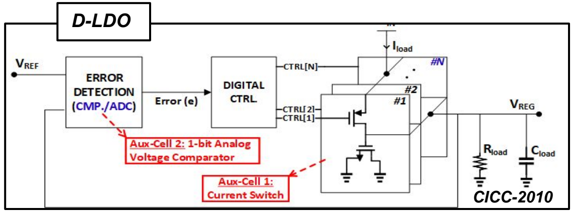


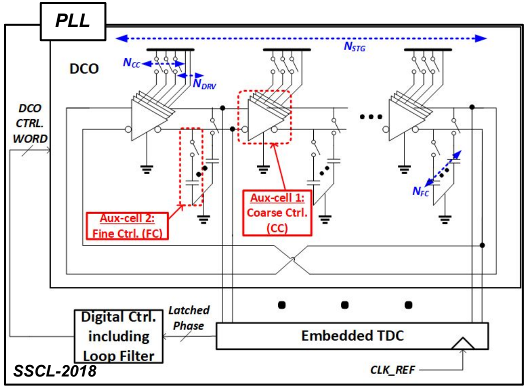
Our Cell-Based Approach to Analog Design



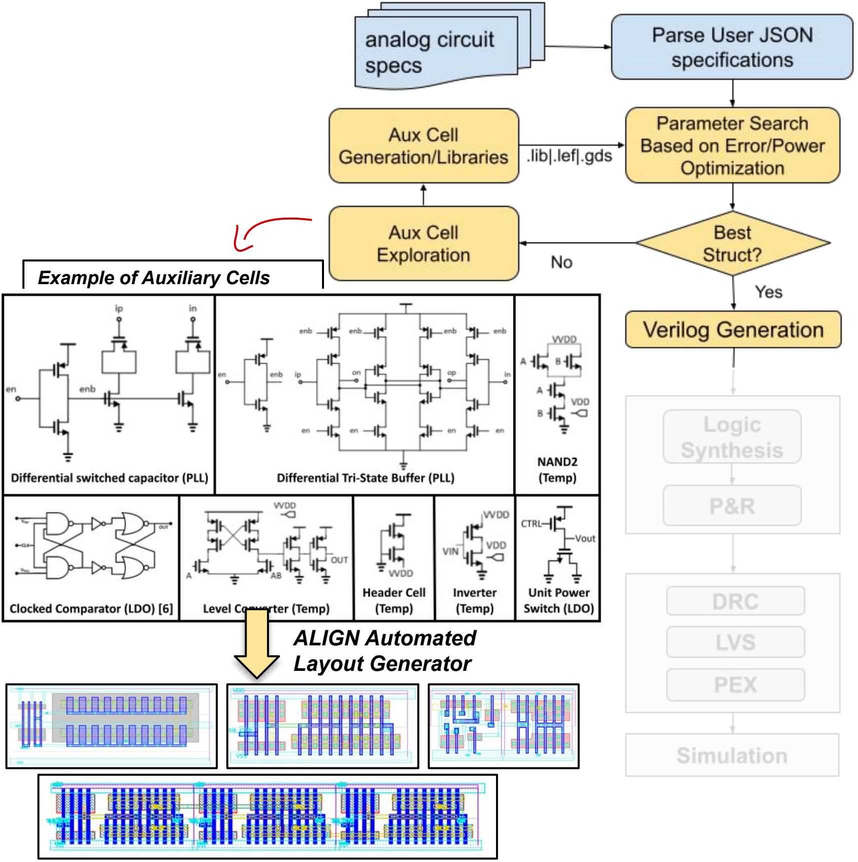








Our Cell-Based Approach to Analog Design



analog circuit specs

Parse User JSON specifications

Aux Cell Generation/Libraries

.lib|.lef|.gds

Parameter Search Based on Error/Power Optimization

Aux Cell Exploration

Best Struct?

No

Yes

Verilog Generation

Logic Synthesis

P&R

DRC

LVS

PEX

Simulation

Example of Auxiliary Cells

Differential switched capacitor (PLL)

Differential Tri-State Buffer (PLL)

NAND2 (Temp)

Clocked Comparator (LDO) [6]

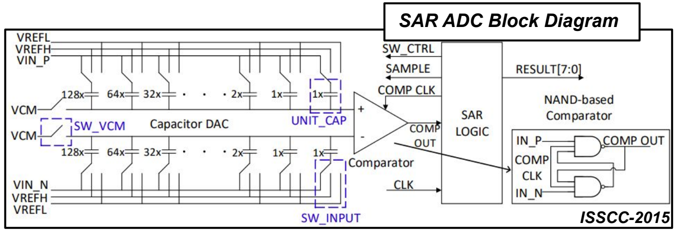
Level Converter (Temp)

Header Cell (Temp)

Inverter (Temp)

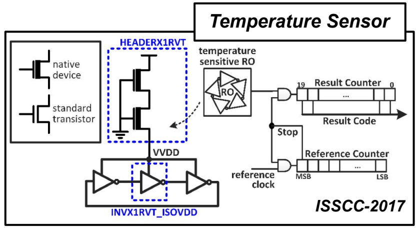
Unit Power Switch (LDO)

ALIGN Automated Layout Generator



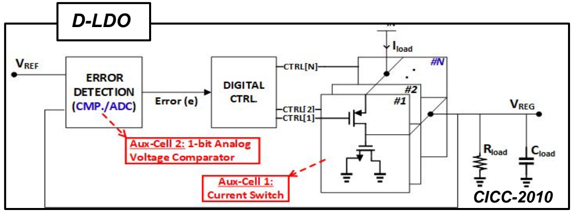
SAR ADC Block Diagram

ISSCC-2015



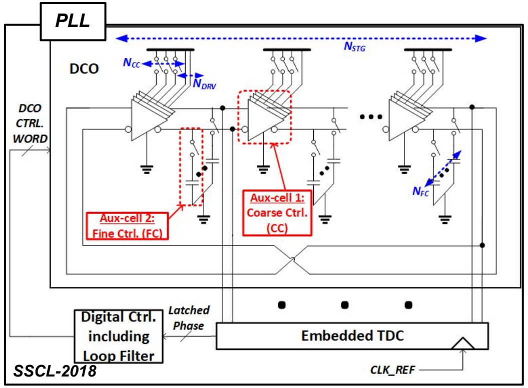
Temperature Sensor

ISSCC-2017



D-LDO

CICC-2010

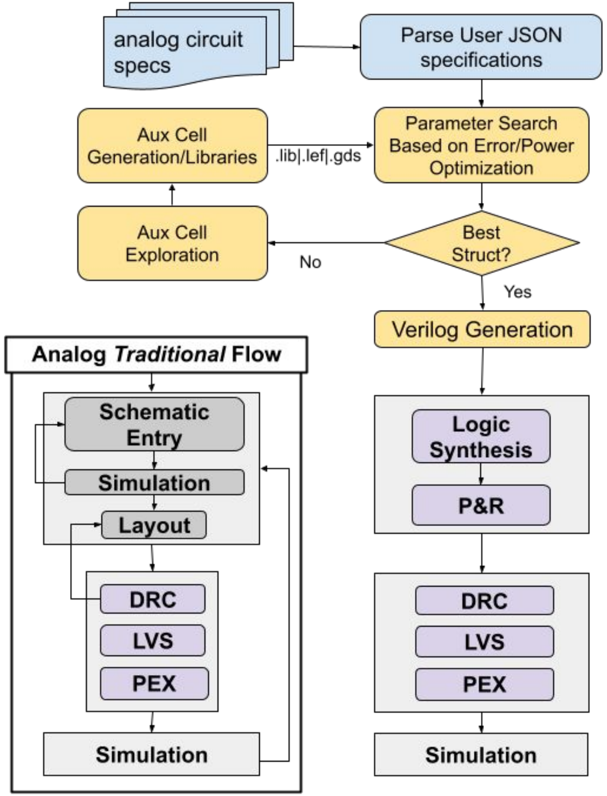


PLL

N_{STG}
DCO
N_{CC}
N_{DRV}
DCO CTRL. WORD
Aux-cell 2: Fine Ctrl. (FC)
Aux-cell 1: Coarse Ctrl. (CC)
N_{FC}
Digital Ctrl. including Loop Filter
Latched Phase
Embedded TDC
CLK_REF

SSCL-2018

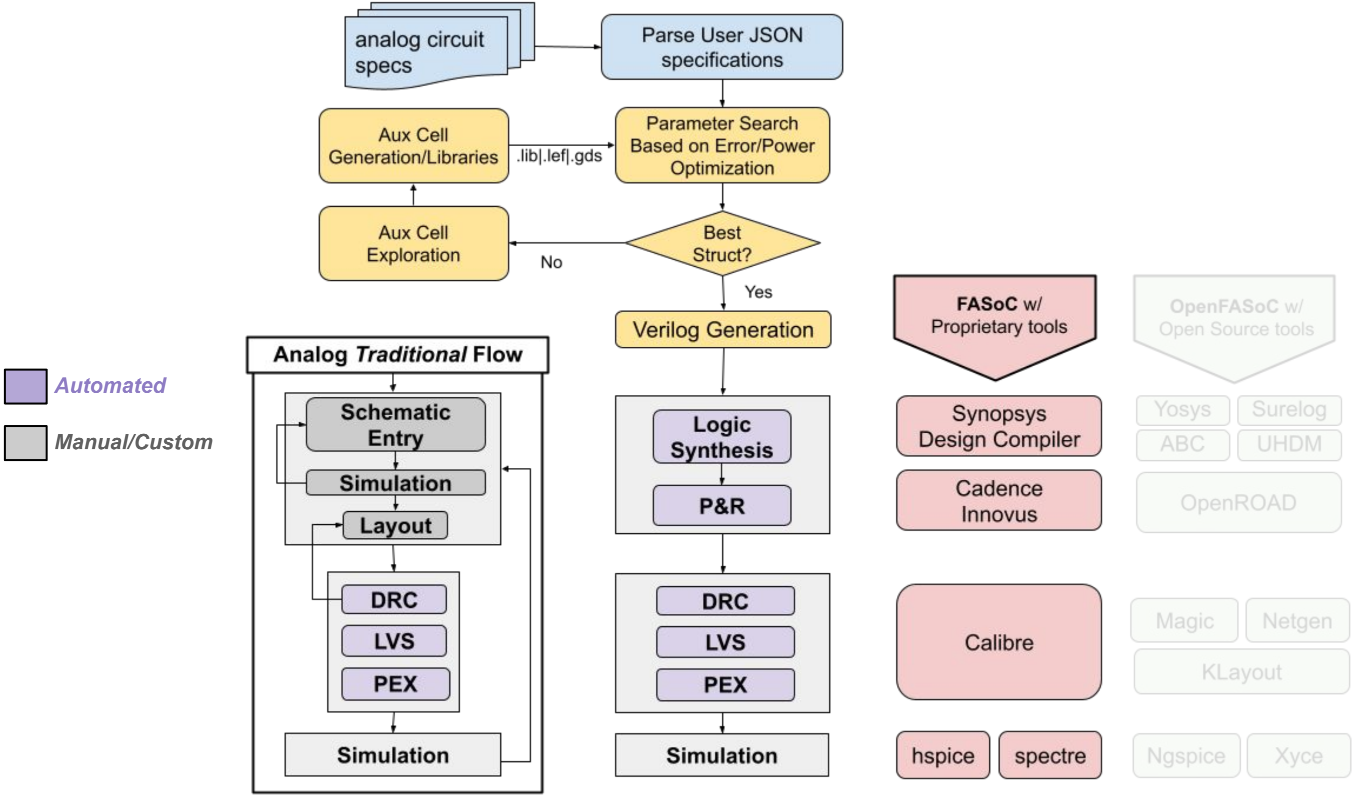
Analog vs. Digital Design Flow Today



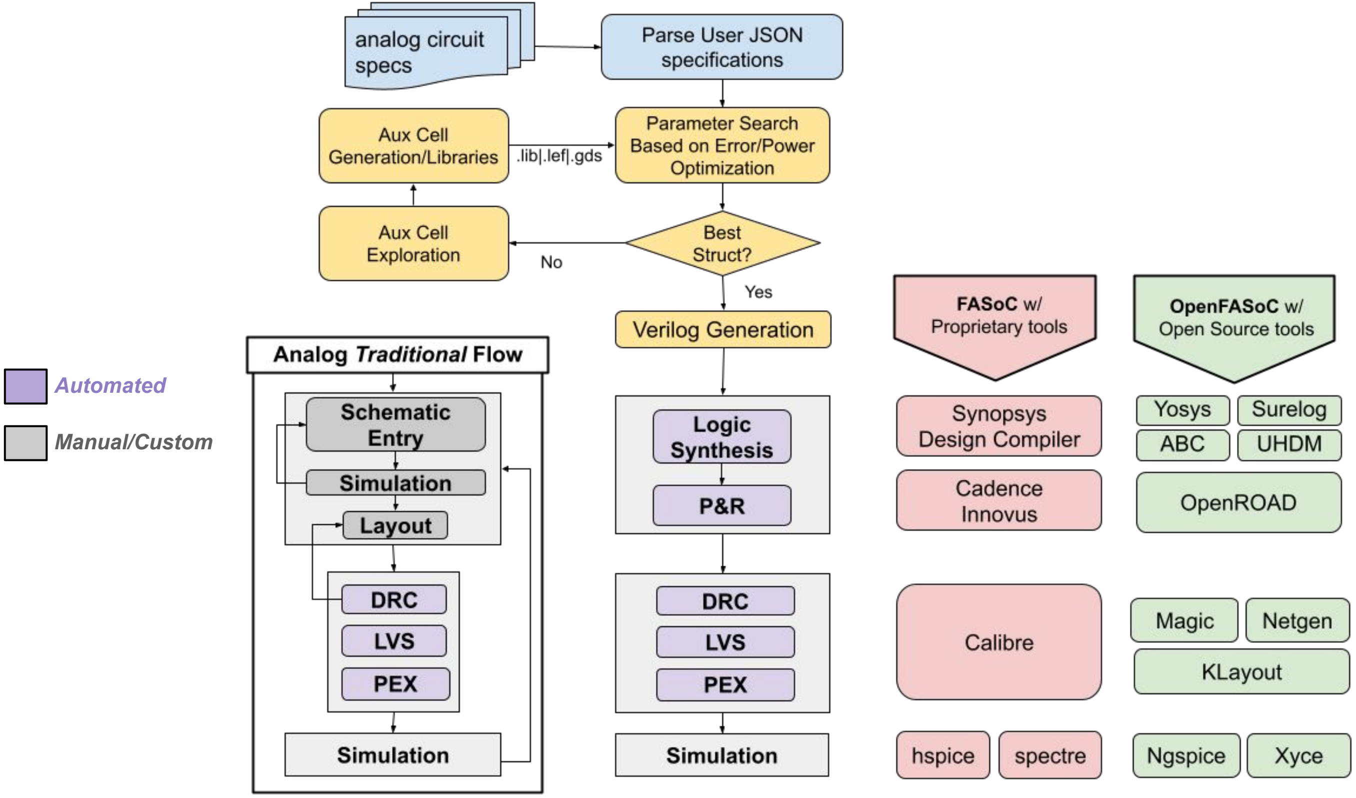
▨ *Automated*

▨ *Manual/Custom*

Proprietary vs. Open Source Design Flow

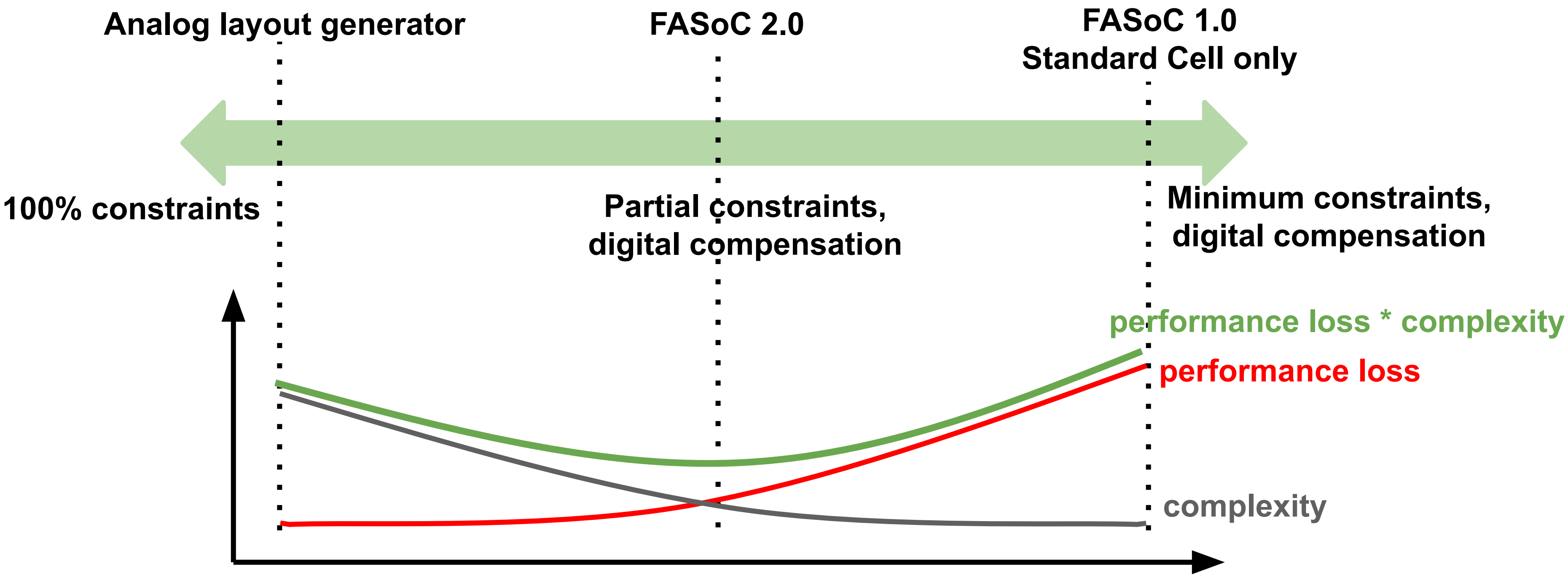


Proprietary vs. Open Source Design Flow



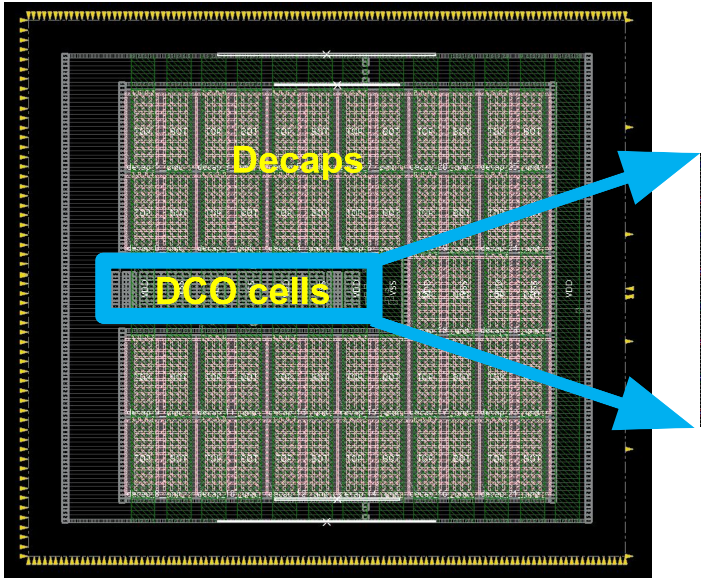
Performance / Complexity Tradeoff

- FASoC augments digital flow with APR tool placement/routing constraints and minimizes the (performance loss * complexity)

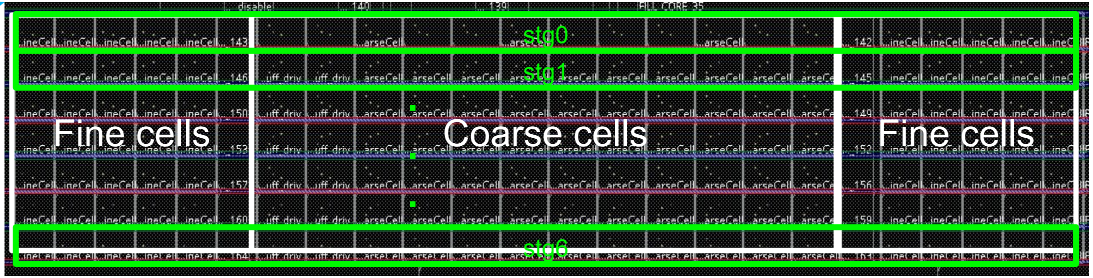


FASoC 2.0 – PLL example

- Patterned placement information generated by python code ⇒ reduce delay mismatch between stages, added Decaps
- Scalable with design parameters



DCO placements with Decap

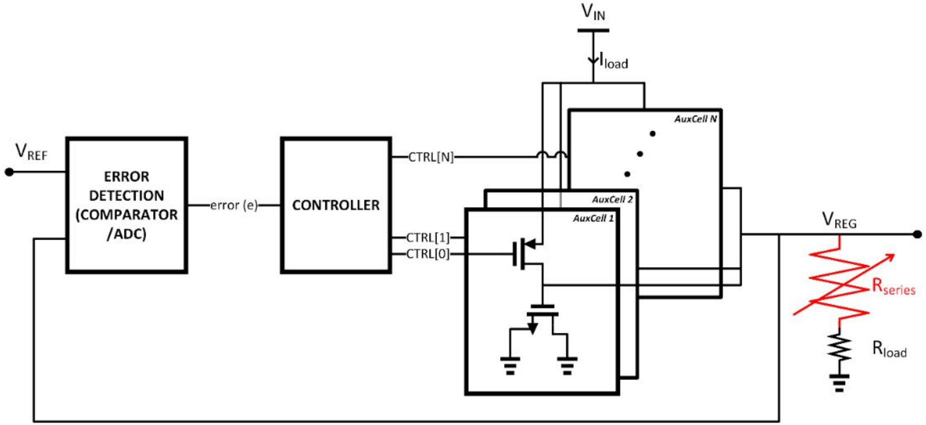


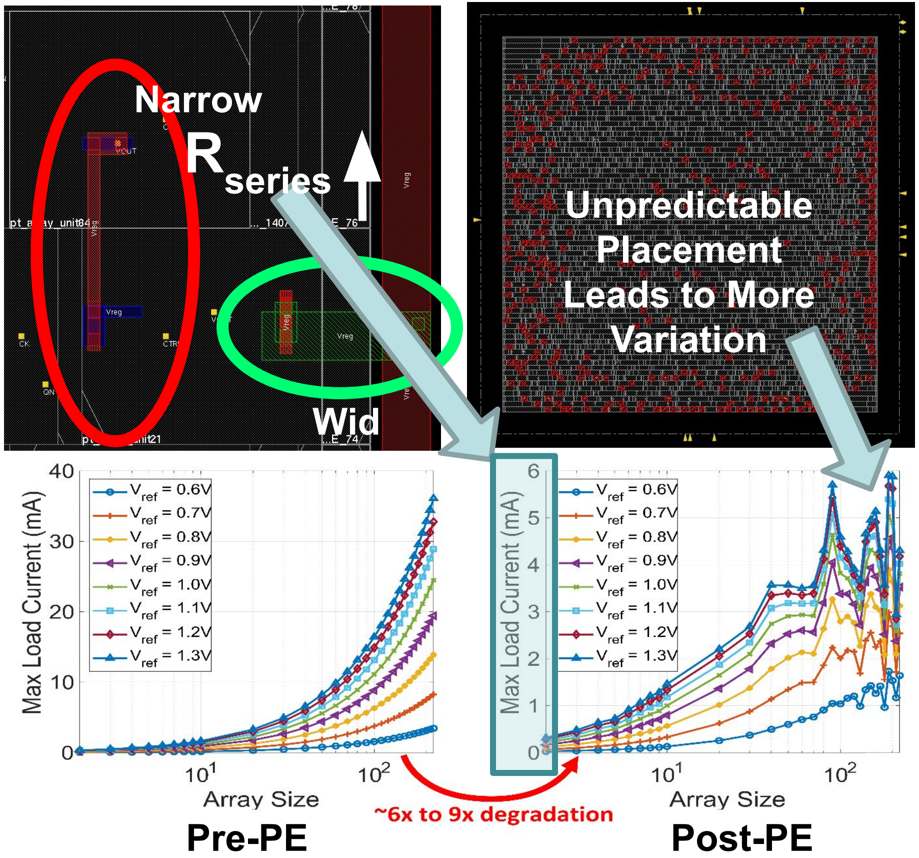
DCO cells

FASoC 2.0 – LDO example

Performance loss caused by PnR

- Large Series Resistance caused by wiring congestion for increased array size
- Unpredictable wiring due to random placement of power cells





Narrow R_{series}

Wid

Unpredictable Placement Leads to More Variation

~6x to 9x degradation

Pre-PE

X

Post-PE

X

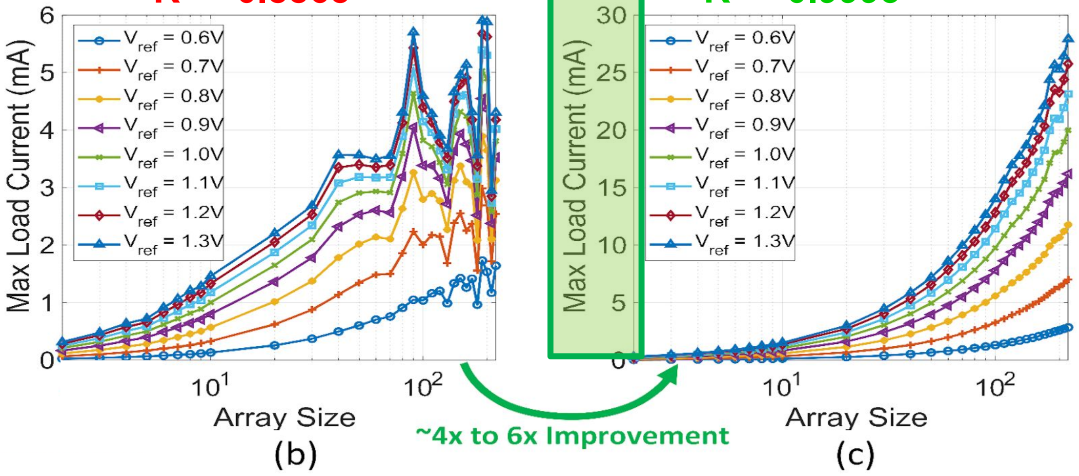
FASoC 2.0 – LDO example

Constraints to improve performance

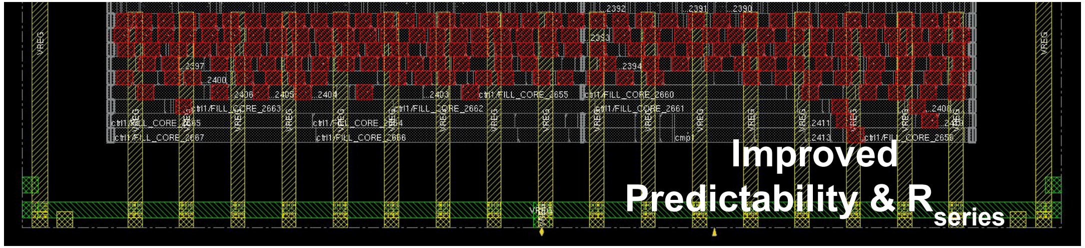
- Technology agnostic fencing to constraint placements
- Use power stripes to improve series R problem
- Automatic analysis of technology database file for determining the stripe metal layers

Constraint less Post-PEX $R^2 = 0.8865$

Scalable Constraint Post-PEX $R^2 = 0.9993$



~4x to 6x Improvement

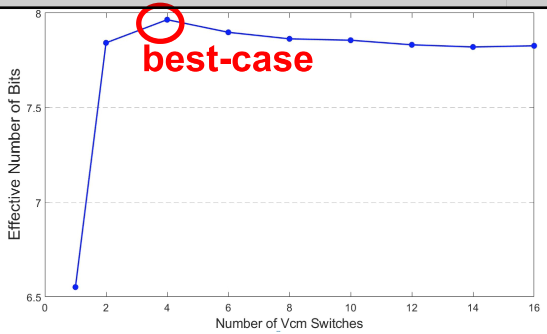


Improved Predictability & R_{series}

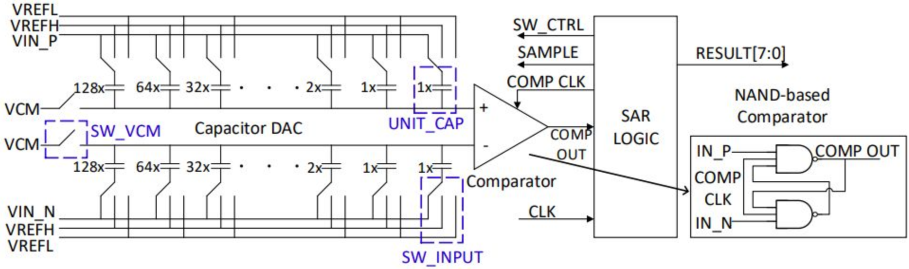
Added blocks: SAR ADC

- Symmetrical Placement of unit caps and switches

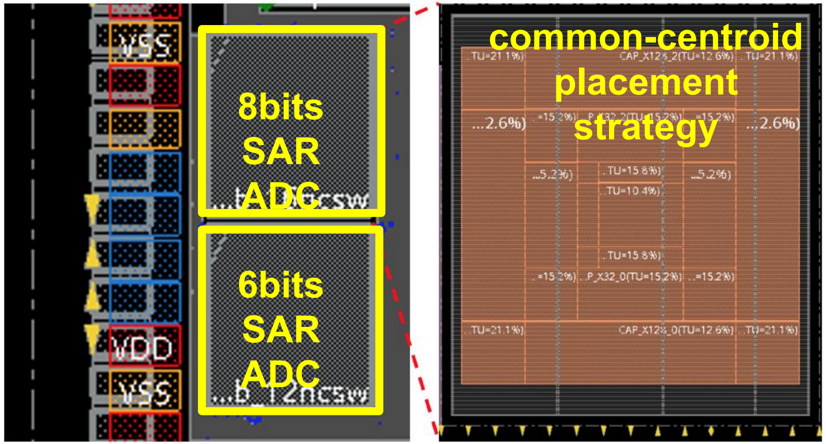
Output Spec.	CDL	PEX
$F_{SAMPLING}$ (MHz)	1	
Unit Cap Value (fF)	2.6	
Area (mm^2)	-	0.04
Power (µW)	6.72	11.2
Effective Number of Bits	7.86	7.75



Effective Number of Bits (ENOB) vs. Number of Vcm Switches



SAR ADC Block Diagram



OpenFASOC is part of the open source EDA and hardware community

- Very active open source community with over 2000 members!

FOSSi Foundation - Dial-Up Talk Series

youtube.com/c/**FOSSiFoundation**/videos

Tim Ansell - Fully open source manufacturable PDK for a 130nm process

The SkyWater Open Source Process Design Kit (PDK) is a joint project of Google and SkyWater Technology Foundry to provide a fully open source PDK.

In this event, Tim Ansell will outline the collaboration and the goals of the project. He will get into the technical details of the PDK and outline the roadmap of the project.

j.mp/du20-pdk

OpenROAD

OpenLANE

Fully automated FOSS RTL to GDSII based on OpenROAD

Mohamed Shalan

Tuesday, July 28, 2020
16:00 GMT

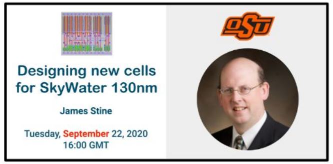
j.mp/du20-openlane

striVe SoC Family

Mohamed Kassem

Tuesday, August 25, 2020
16:00 GMT

j.mp/du20-strive

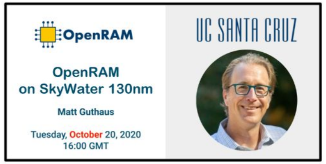


Designing new cells for SkyWater 130nm

James Stine

Tuesday, September 22, 2020
16:00 GMT

j.mp/du20-stdcells



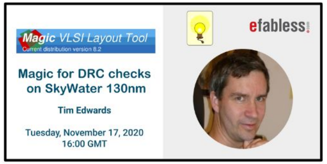
OpenRAM

OpenRAM on SkyWater 130nm

Matt Guthaus

Tuesday, October 20, 2020
16:00 GMT

j.mp/du20-openram



Magic VLSI Layout Tool
Current distribution version 8.2

Magic for DRC checks on SkyWater 130nm

Tim Edwards

Tuesday, November 17, 2020
16:00 GMT

j.mp/du20-magic

https://j.mp/esscxxrc21-sky130

Fully open source, *manufacturable*, PDK for 130nm process

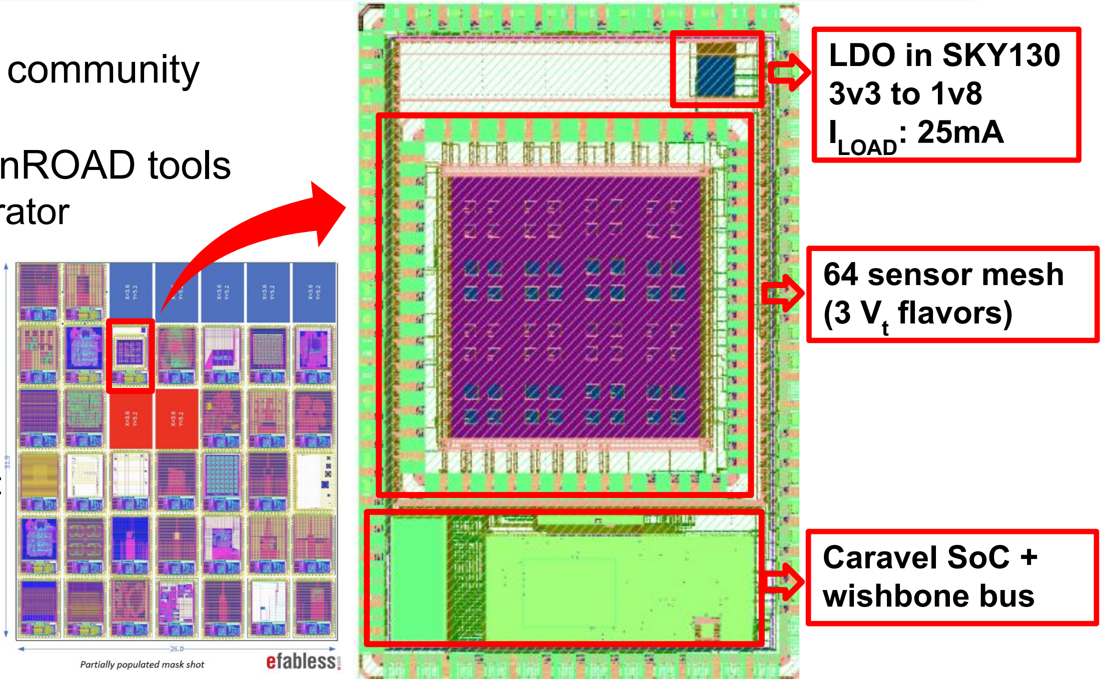
with fully open source tooling
AND no-cost MPW shuttle program!

Tim "mithro" Ansell <tansell@google.com>

- Publications:
  - T. Ansell and M. Saligane, "The Missing Pieces of Open Design Enablement: A Recent History of Google Efforts : Invited Paper," 2020 IEEE/ACM International Conference On Computer Aided Design (ICCAD), San Diego, CA, USA, 2020, pp. 1-8.

OpenFASOC on MPW-I: 64 sensors + D-LDO

- Actively contributing to the open source community
- 1st open FASoC flow built on top of OpenROAD tools
  - Focused on the Temp. Sensor Generator
- FASoC testchip in SKY130:
  - Includes Caravel SoC
  - 64 Temp. Sensor Mesh
  - LDO ported (~ a week)
    - Updated comparatc strongArm latch
    - 5v native NMOS switch



Partially populated mask shot

LDO in SKY130
3v3 to 1v8
I_{LOAD}: 25mA

64 sensor mesh
(3 V_t flavors)

Caravel SoC +
wishbone bus

Test-chip in MPW-I

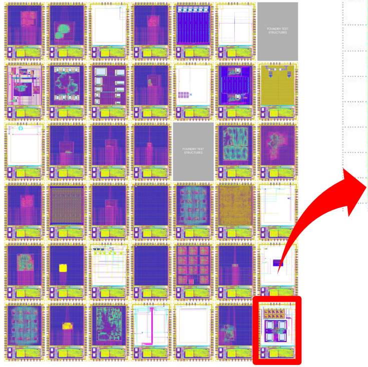
Open MPW Shuttle Program

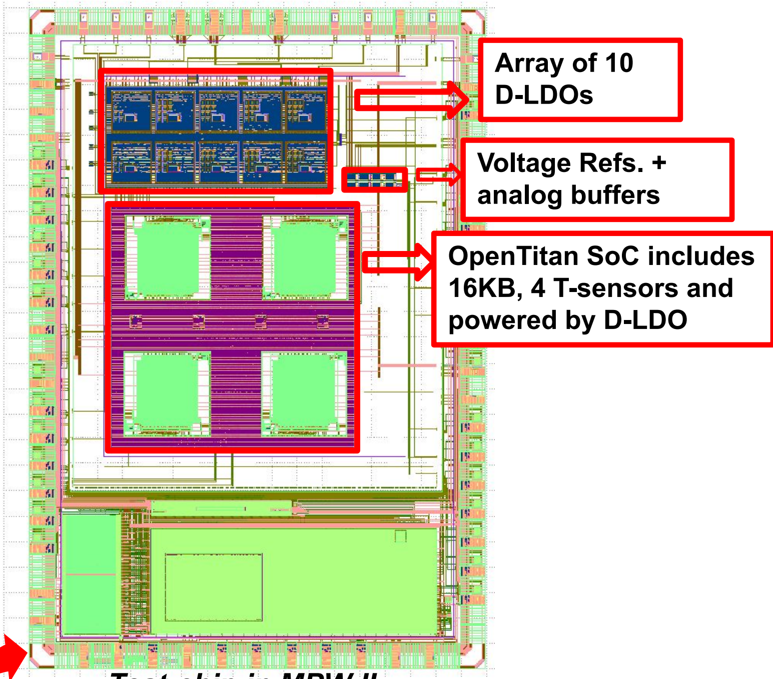
efabless Sponsored by Google

OpenFASOC on MPW-II: 1st Open Source AMS SoC

- Included initial support for voltage domains in OpenROAD
- Implementation of the OpenTitan SoC using an ECO flow to fix hold timing with degrading the F_{MAX}
- Temperature Sensor generator is using an end-to-end Open Source flow
- Updates to the D-LDO generator:
  - Embedded voltage references
  - Decap cells using MIM cap.
  - Multiple implementations and I_{LOAD}

- https://efabless.com/projects/239
- https://github.com/msaligane/caravan_openfasoc.git





Array of 10 D-LDOs

Voltage Refs. + analog buffers

OpenTitan SoC includes 16KB, 4 T-sensors and powered by D-LDO

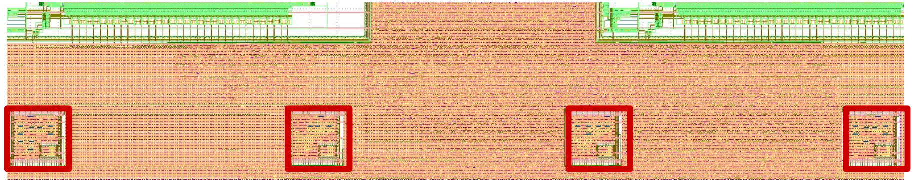
Test-chip in MPW-II

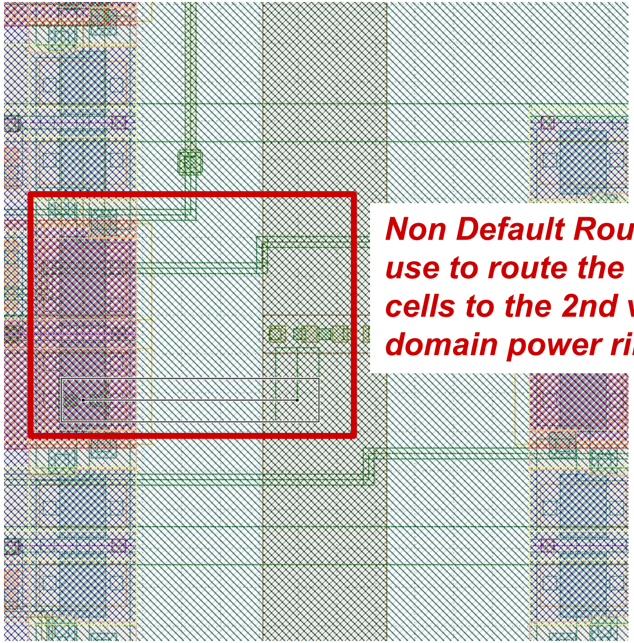
Open MPW Shuttle Program

efabless Sponsored by Google

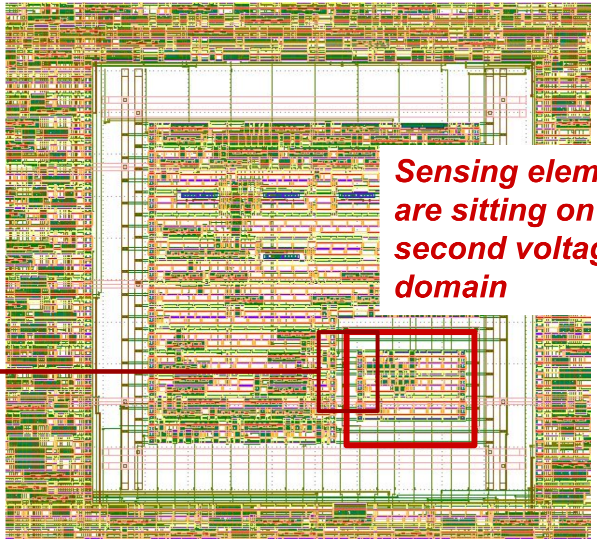
OpenFASOC on MPW-II: Integrated Temperature Sensors

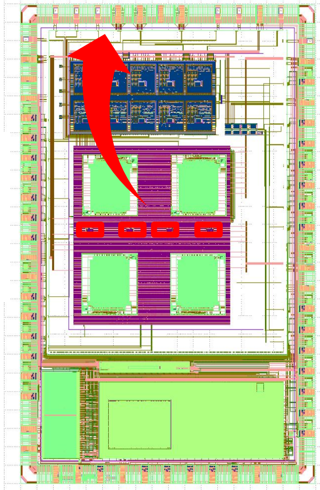
Sensors are embedded inside the OpenTitan SoC and connected through tilelink





Non Default Routing is use to route the header cells to the 2nd voltage domain power ring

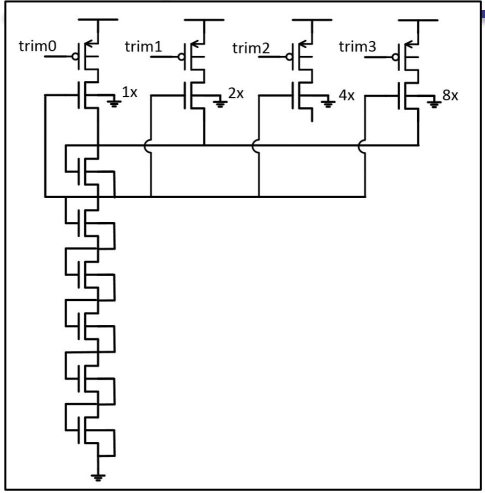




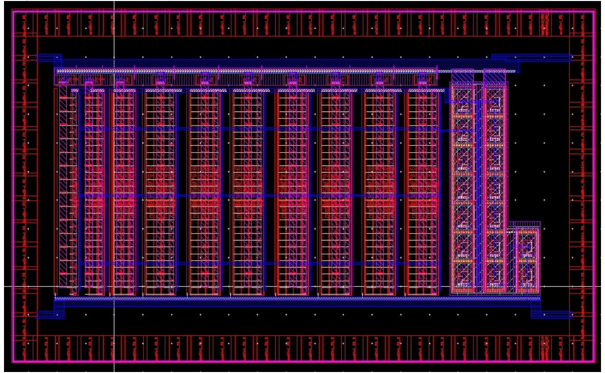
Sensing elements are sitting on a second voltage domain

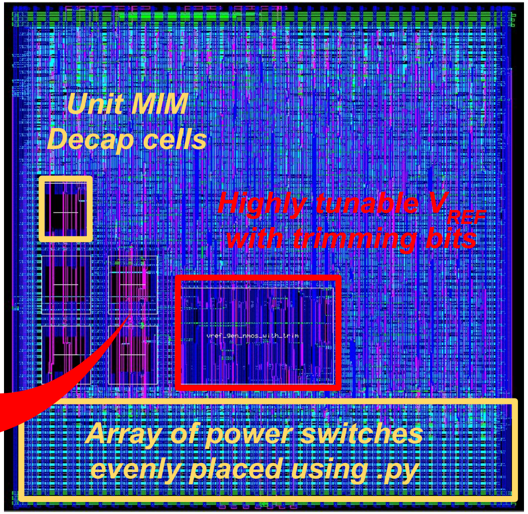
The temperature sensor generator uses a fully open source flow

OpenFASOC on MPW-II: D-LDO generator



Voltage Reference with symmetrical placement

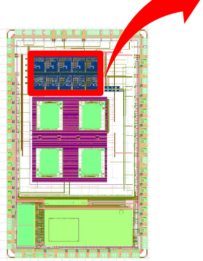


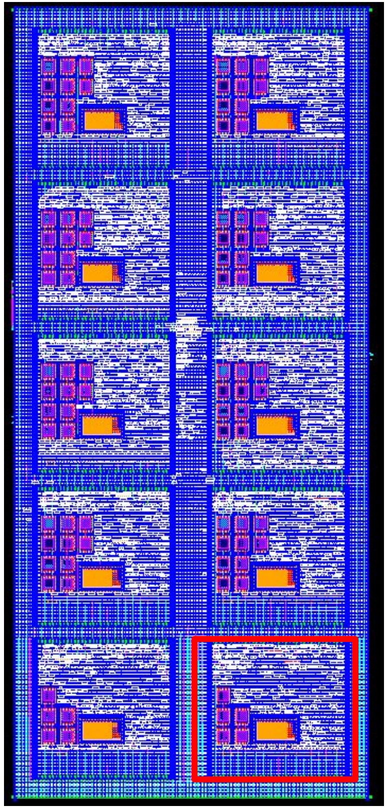


Unit MIM Decap cells

Highly tunable V_{REF} with trimming bits

Array of power switches evenly placed using .py



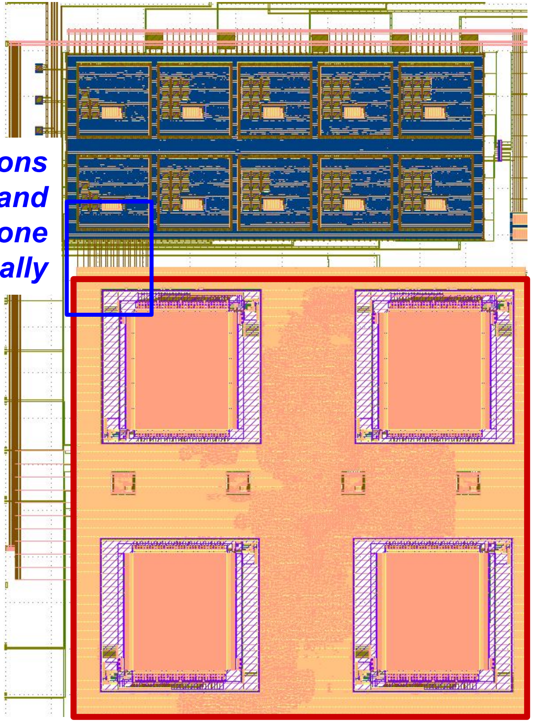


Array of D-LDOs

OpenFASOC on MPW-II: OpenTitan SoC

- 1st SoC using AMS components
-
- The Opentitan SoC contains
  - UART, SPI interfaces
  - 16KB of SRAM (OpenRAM)
  - D-LDO is used to power-up all the blocks
  - All Peripherals are connected through Tilelink

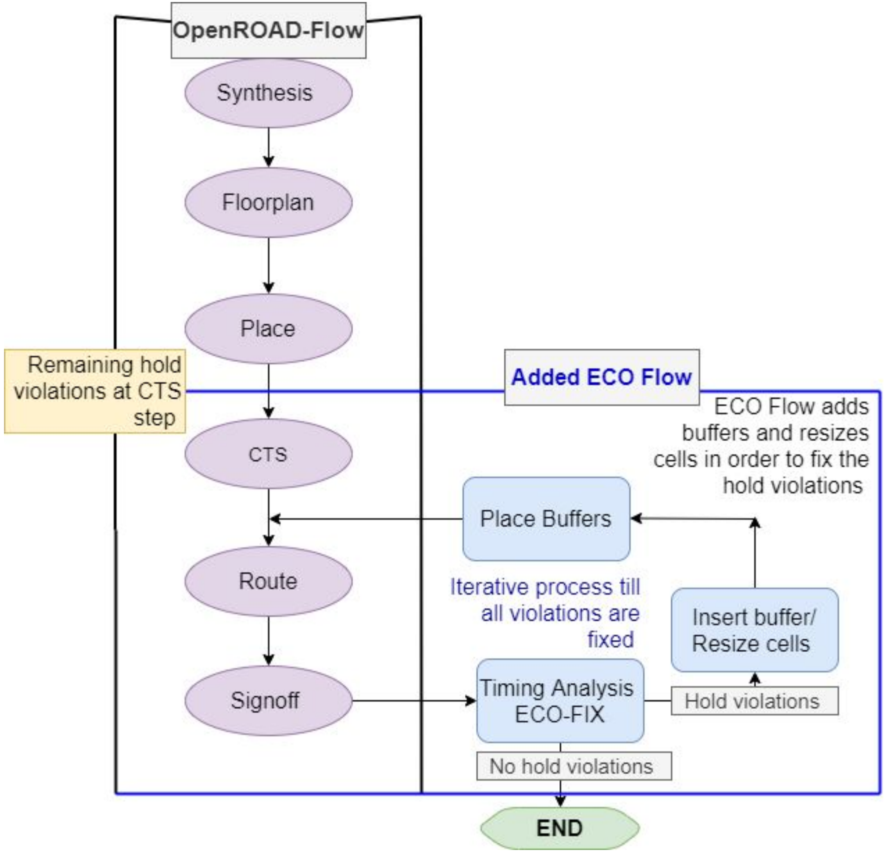
- Timing has been carefully checked and an ECO flow has been used to avoid altering the F_{MAX} while fixing hold violations



Power connections between LDO and OpenTitan are done Manually

OpenTitan AMS SoC - Die Photo

OpenFASOC on MPW-II: OpenTitan SoC - ECO flow


OpenROAD-Flow
Synthesis
Floorplan
Place
CTS
Route
Signoff
Remaining hold violations at CTS step
Added ECO Flow
ECO Flow adds buffers and resizes cells in order to fix the hold violations
Place Buffers
Iterative process till all violations are fixed
Insert buffer/ Resize cells
Timing Analysis ECO-FIX
Hold violations
No hold violations
END

- Custom automated ECO flow to close timing
- Modular flow using both proprietary and open source tool based flows

	OpenROAD flow	OpenROAD + ECO
Clock (ns)	48	48
Fmax (MHz)	20.8	20.8
Setup Time (ns)	11.01	9.44
Hold Time (ns)	-1.72	0.01
No of violations	1493	0
No of iterations for ECO	0	5

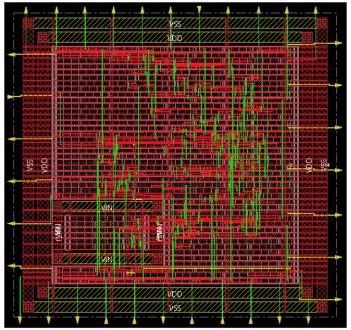
Hold fixing table with and without ECO

OpenFASOC on MPW-II: OpenROAD tooling

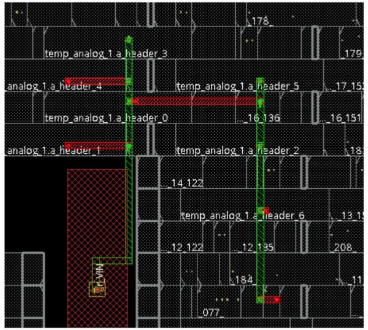
- The OpenROAD's team is actively improving their tools and adding new design features

- Closely working with UCSD and ARM to enable an AMS flow (power gating, UPF flow)

Custom nets python scripts are used for special routing



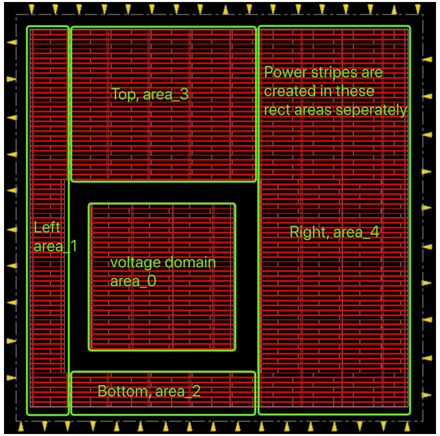
Layout view



Custom net

Example of code updates to create new PD features

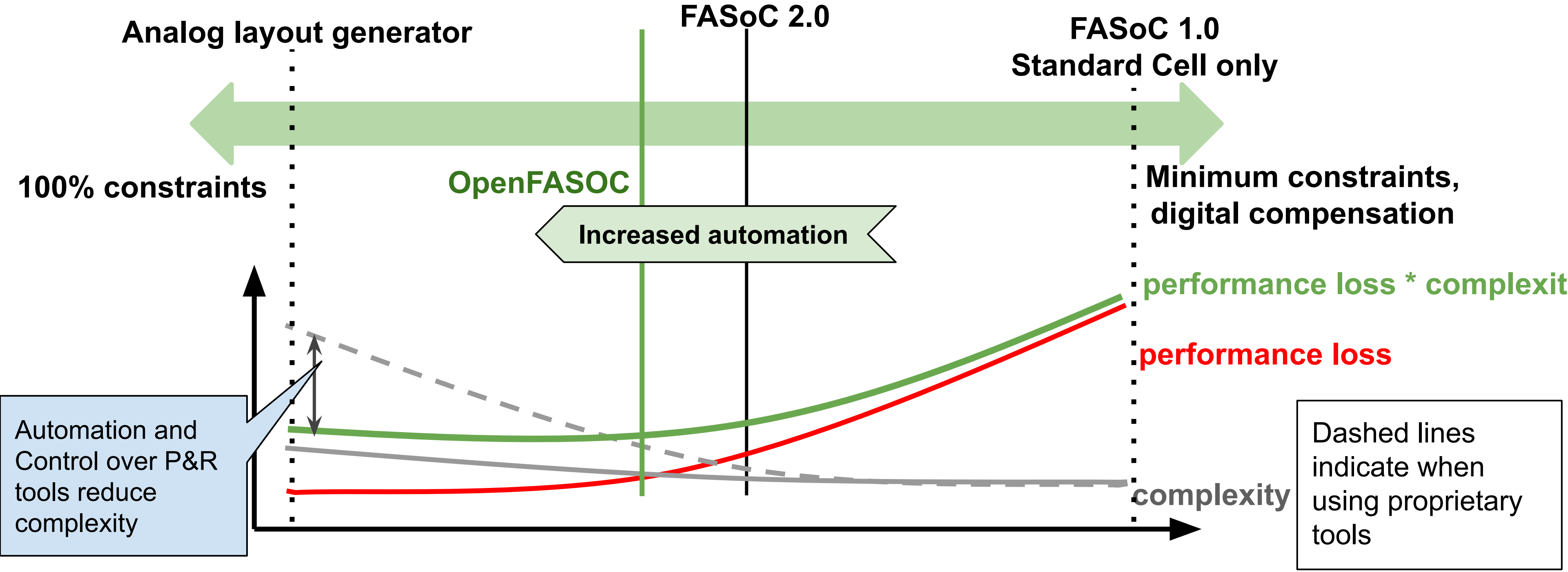
File name	**Func/Proc**
OpenROAD/src/init_fp/src/InitFloorplan.cc (Floorplan)	updateVoltageDomain()
OpenROAD/src/pdngen/src/PdnGen.tcl (Floorplan)	generate_stripes{}
OpenROAD/src/replace/src/replace.tcl (Placement)	global_placement{}



Example of* `create_voltage_domain` *usage

Performance / Complexity Tradeoff (OpenFASOC)

- FASoC augments digital flow with APR tool placement/routing constraints and minimizes the (performance loss * complexity)

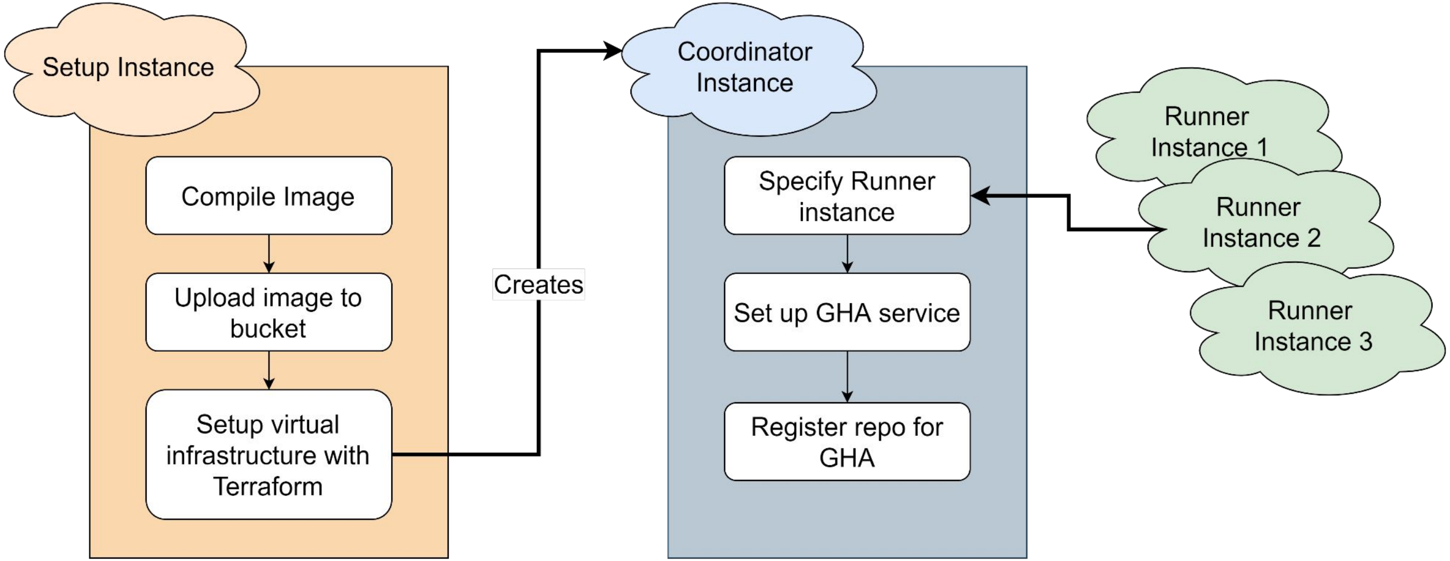


Automation and Control over P&R tools reduce complexity

Dashed lines indicate when using proprietary tools

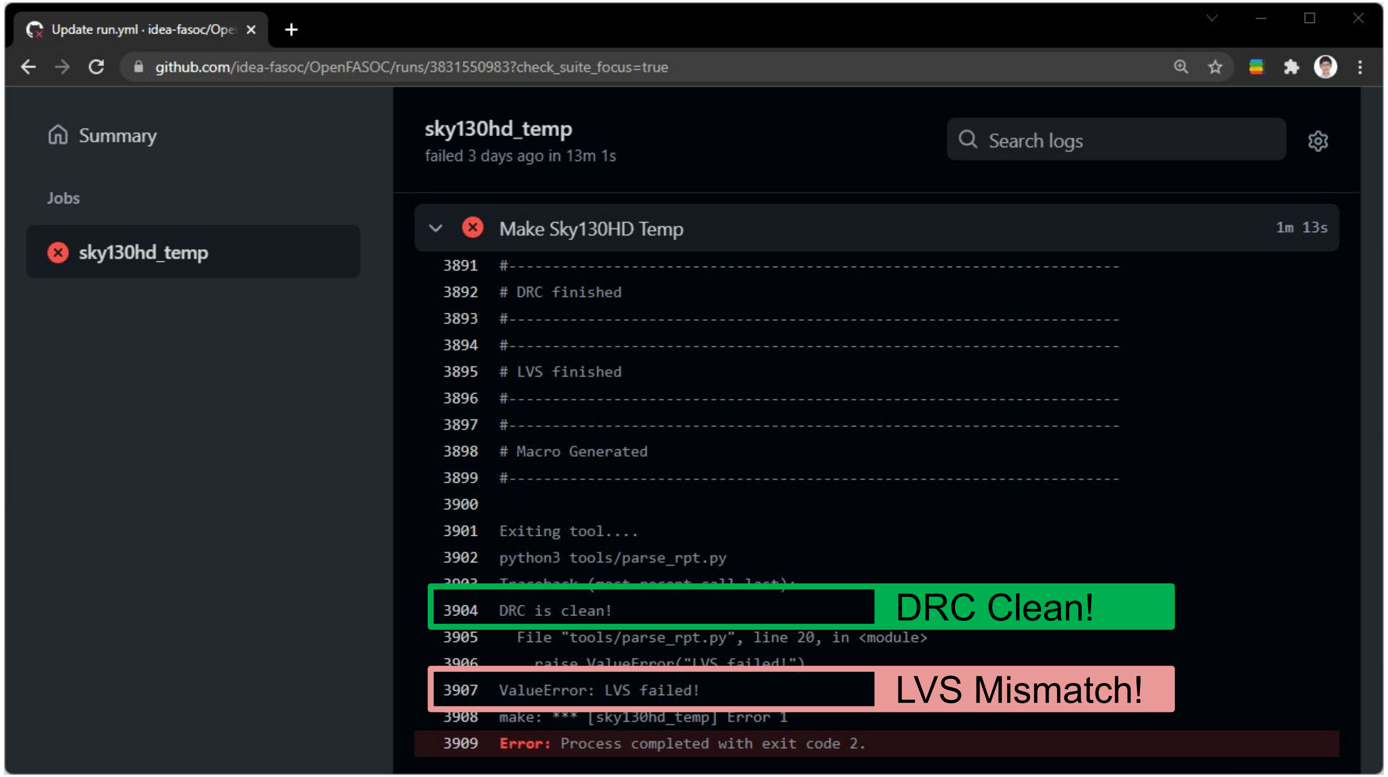
Continuous Integration (CI) for OpenFASoC

- First tested using GitHub actions (GHA) with Microsoft-hosted VMs
- Now using self-hosted VMs in GCP to run CI
  - Uses forked runner from Antmicro ([Antmicro · Open source custom GitHub Actions runners with Google Cloud and Terraform]())



Continuous Integration (CI) for OpenFASoC

- Includes automated checking of DRC and LVS errors


Update run.yml · idea-fasoc/Ope
github.com/idea-fasoc/OpenFASOC/runs/3831550983?check_suite_focus=true
Summary
Jobs
sky130hd_temp
sky130hd_temp
failed 3 days ago in 13m 1s
Search logs
Make Sky130HD Temp
1m 13s

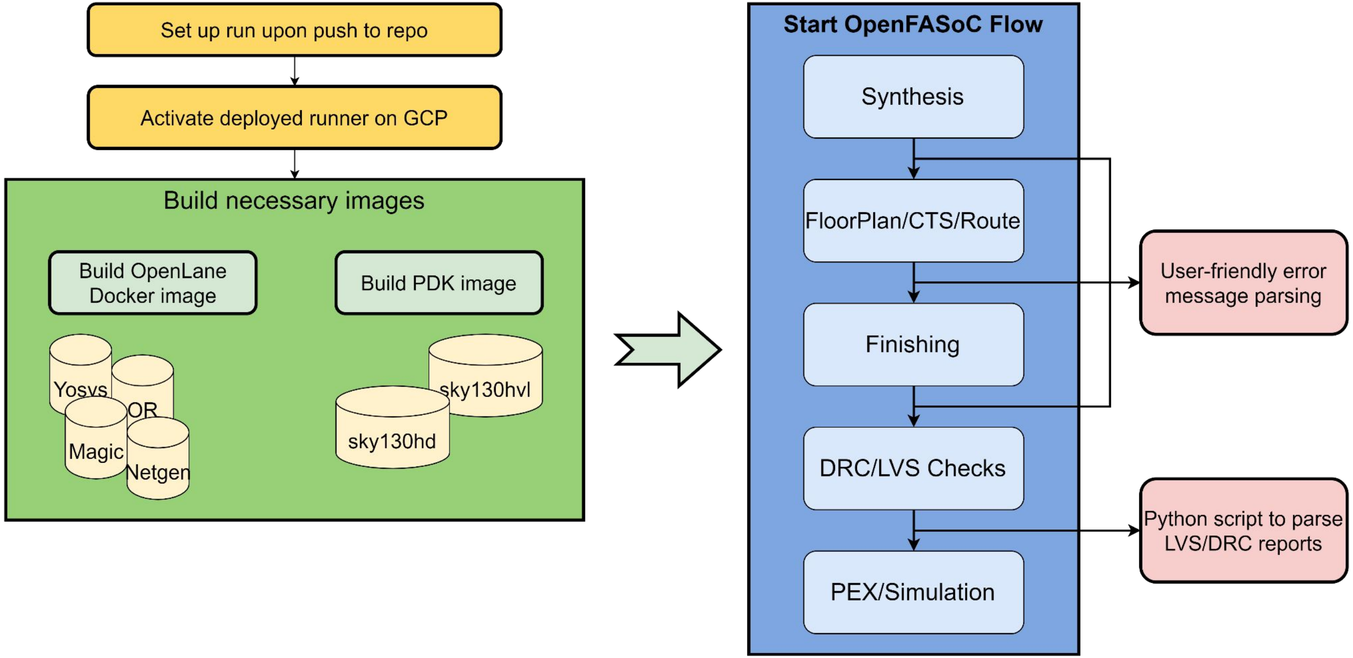
```
#------------------------------------------------------------------------
# DRC finished
#------------------------------------------------------------------------
#------------------------------------------------------------------------
# LVS finished
#------------------------------------------------------------------------
#------------------------------------------------------------------------
# Macro Generated
#------------------------------------------------------------------------

Exiting tool....
python3 tools/parse_rpt.py
Traceback (most recent call last):
DRC is clean!
  File "tools/parse_rpt.py", line 20, in <module>
    raise ValueError("LVS failed!")
ValueError: LVS failed!
make: *** [sky130hd_temp] Error 1
Error: Process completed with exit code 2.
```

DRC Clean!

LVS Mismatch!

Continuous Integration (CI) for OpenFASoC

- CI is now set up for temp-sense
  - Uses Github actions to pull Docker images to run flow
  - Includes report-parsers for DRC/LVS logs

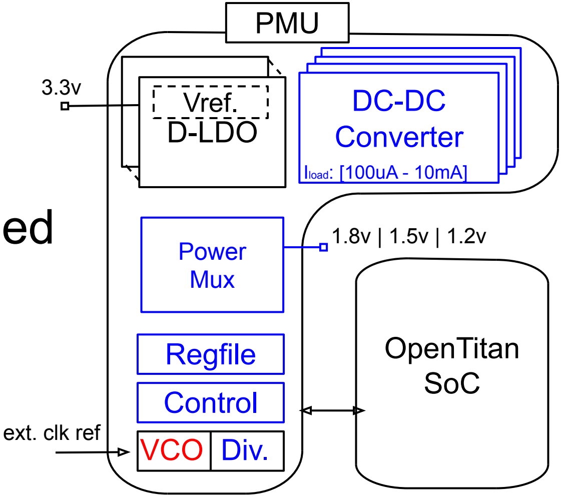

Set up run upon push to repo
Activate deployed runner on GCP
Build necessary images
Build OpenLane Docker image
Build PDK image
Yosys
OR
Magic
Netgen
sky130hvl
sky130hd
Start OpenFASoC Flow
Synthesis
FloorPlan/CTS/Route
Finishing
DRC/LVS Checks
PEX/Simulation
User-friendly error message parsing
Python script to parse LVS/DRC reports

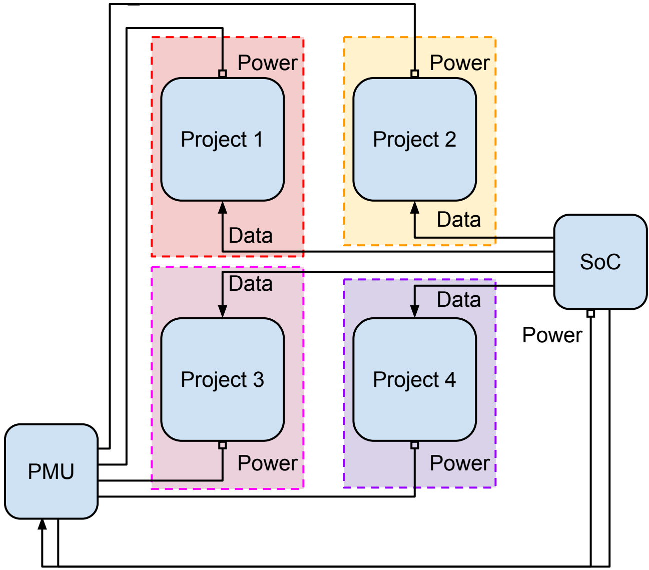
Future Work

- Power Management Unit generator which includes our switched cap. DC-DC converter

- PVTA Sensors

- SERDES using the cell based approach





Thanks!

- **Github links:**

https://github.com/idea-fasoc

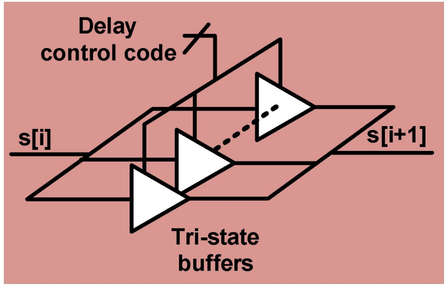
https://github.com/idea-fasoc/OpenFASOC

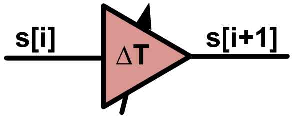
https://github.com/msaligane/caravan_openfasoc

https://github.com/msaligane/opentitan_soc

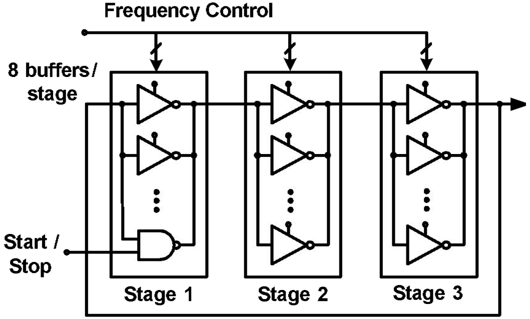
Example

Delay element using *N* tri-state buffers



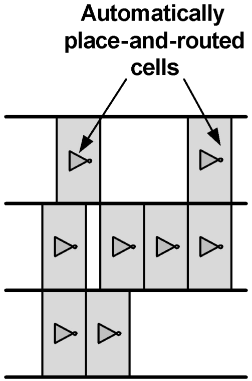


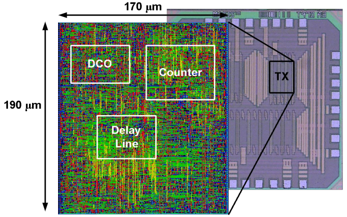
Structural Verilog to describe DCO with *S* stages



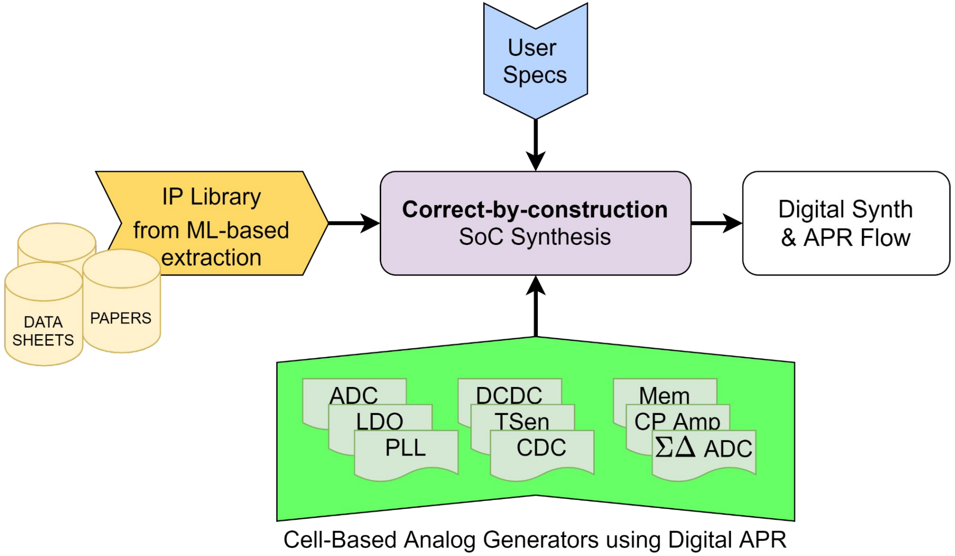
DCO
Frequency control

Synth & APR flow produces GDS





Placeholder



User Specs

IP Library from ML-based extraction

DATA SHEETS

PAPERS

Correct-by-construction SoC Synthesis

Digital Synth & APR Flow

ADC

LDO

PLL

DCDC

TSen

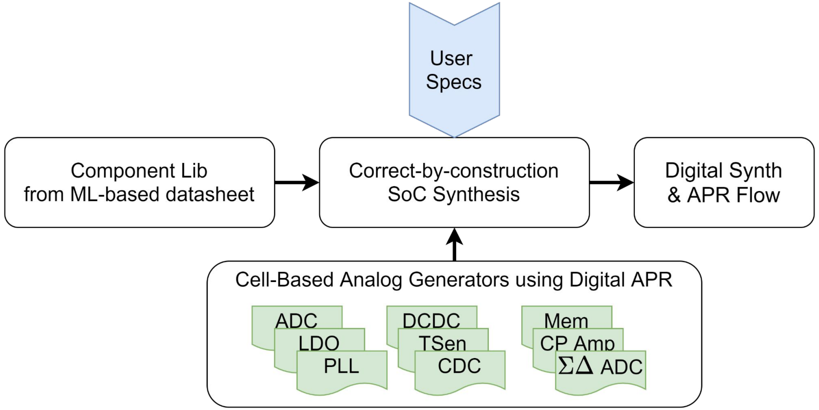
CDC

Mem

CP Amp

ΣΔ ADC

Cell-Based Analog Generators using Digital APR



User Specs

Component Lib from ML-based datasheet

Correct-by-construction SoC Synthesis

Digital Synth & APR Flow

Cell-Based Analog Generators using Digital APR

ADC

LDO

PLL

DCDC

TSen

CDC

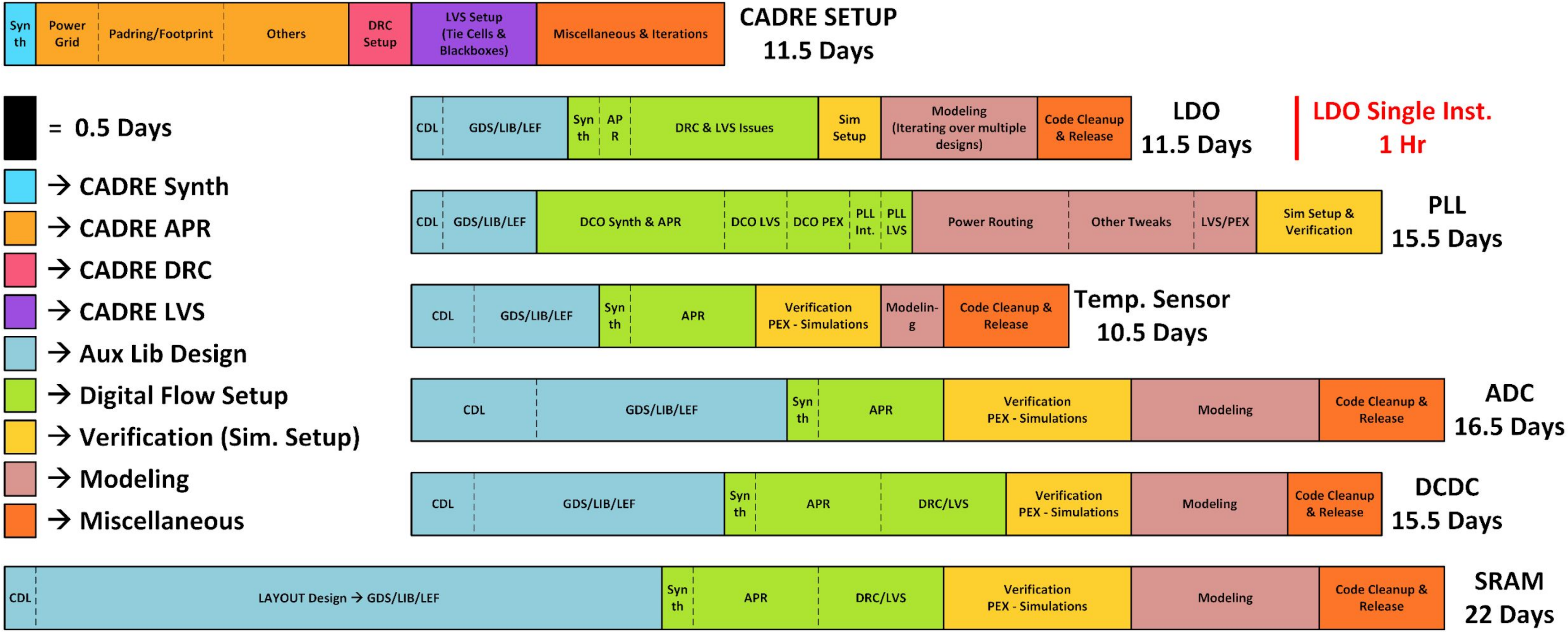
Mem

CP Amp

ΣΔ ADC

GF12 Porting Timeline – One Time Cost

- **Start with CADRE tool flow setup, followed by analog generators**



Syn th | Power Grid | Padring/Footprint | Others | DRC Setup | LVS Setup (Tie Cells & Blackboxes) | Miscellaneous & Iterations

CADRE SETUP
11.5 Days

⬛ = 0.5 Days

🟦 → CADRE Synth

🟧 → CADRE APR

🟥 → CADRE DRC

🟪 → CADRE LVS

🟦 → Aux Lib Design

🟩 → Digital Flow Setup

🟨 → Verification (Sim. Setup)

🟫 → Modeling

🟧 → Miscellaneous

CDL | GDS/LIB/LEF | Syn th | AP R | DRC & LVS Issues | Sim Setup | Modeling (Iterating over multiple designs) | Code Cleanup & Release

LDO
11.5 Days

LDO Single Inst.
1 Hr

CDL | GDS/LIB/LEF | DCO Synth & APR | DCO LVS | DCO PEX | PLL Int. | PLL LVS | Power Routing | Other Tweaks | LVS/PEX | Sim Setup & Verification

PLL
15.5 Days

CDL | GDS/LIB/LEF | Syn th | APR | Verification PEX - Simulations | Modelin-g | Code Cleanup & Release

Temp. Sensor
10.5 Days

CDL | GDS/LIB/LEF | Syn th | APR | Verification PEX - Simulations | Modeling | Code Cleanup & Release

ADC
16.5 Days

CDL | GDS/LIB/LEF | Syn th | APR | DRC/LVS | Verification PEX - Simulations | Modeling | Code Cleanup & Release

DCDC
15.5 Days

CDL | LAYOUT Design → GDS/LIB/LEF | Syn th | APR | DRC/LVS | Verification PEX - Simulations | Modeling | Code Cleanup & Release

SRAM
22 Days