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

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

May 2021

PI: David Wentzloff, Professor, UM

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

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

Approach:

Component library populated by ML-based datasheet scrubbing

User Specs

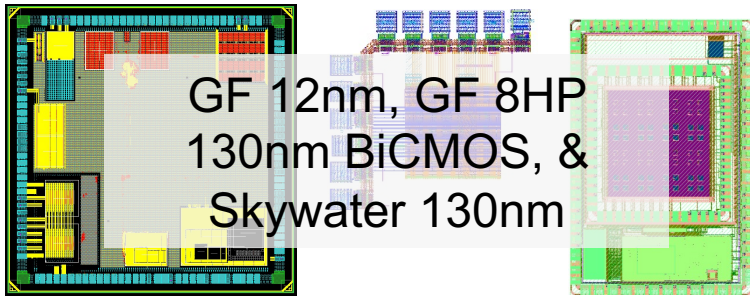
Correct-by-construction SoC synth

Digital Synth & APR Flow

First automated SoC Fab'd in TSMC65nm

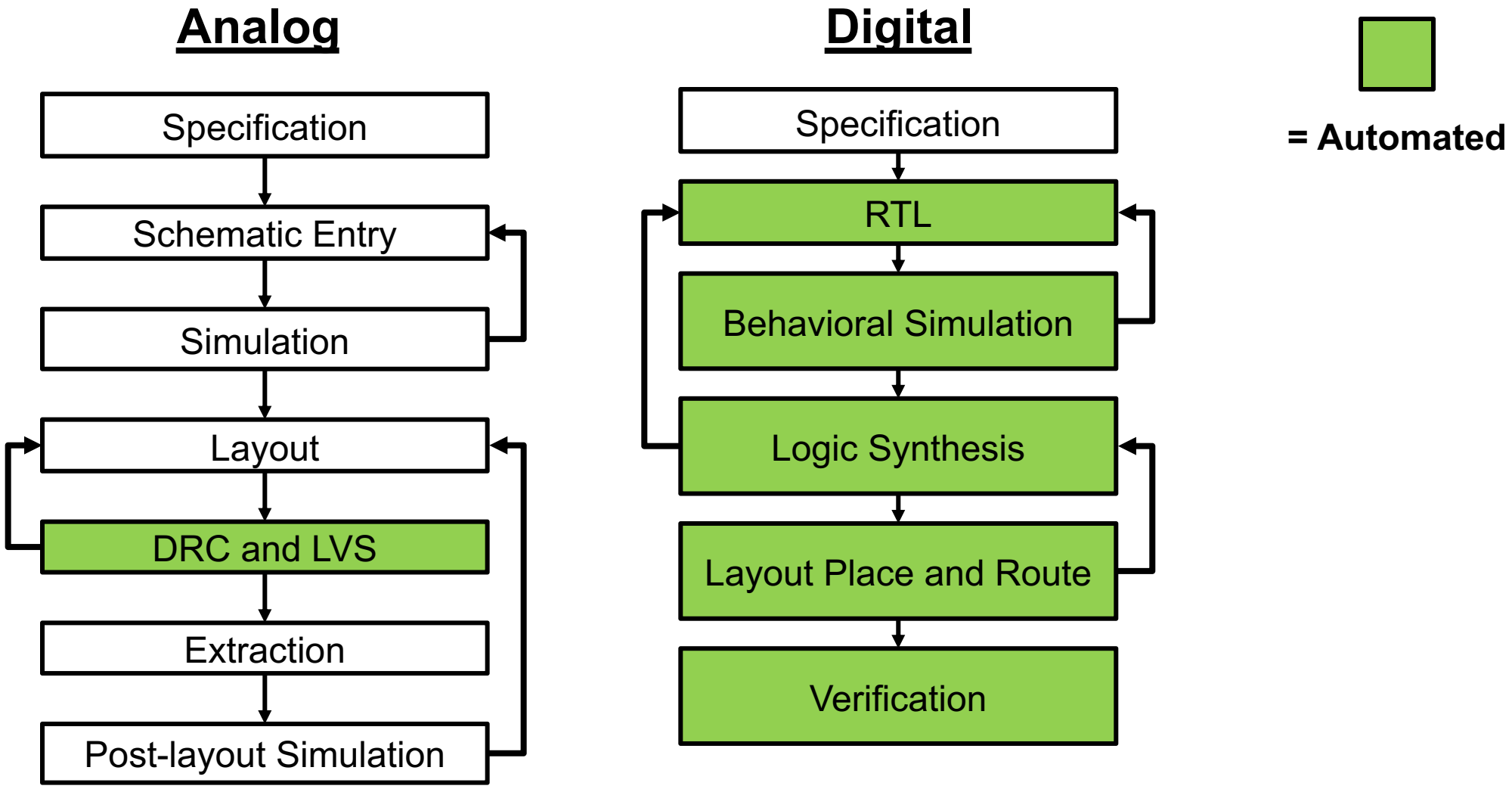
Cell-based analog generators using digital APR flow

- ADC_Gen
- LDO_Gen
- PLL_Gen
- Mem_Gen
- DCDC_Gen
- TSen_Gen
- CDC_Gen
- CP Amp
- ΣΔ ADC



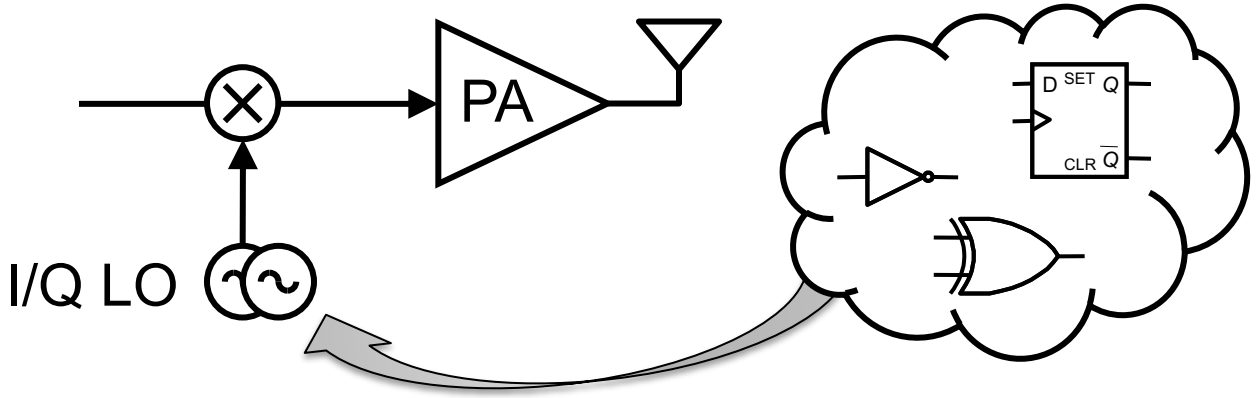
GF 12nm, GF 8HP
130nm BiCMOS, &
Skywater 130nm

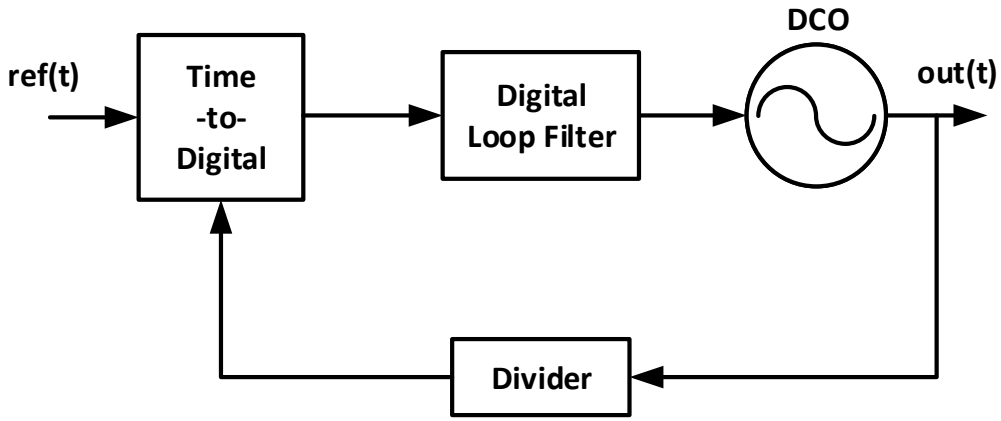
Analog vs. Digital Design Flow Today



Cell-Based Synthesizable Analog (VLSA)

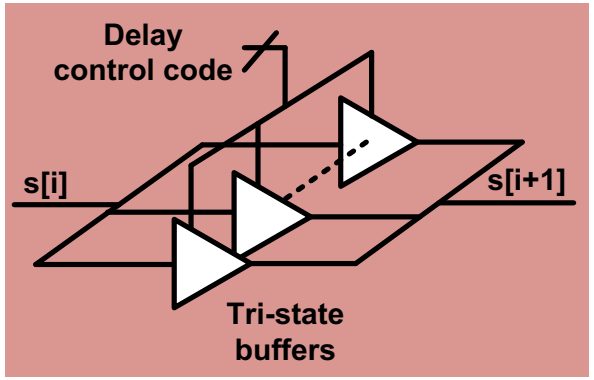
- **Goal: Described analog blocks in Verilog using standard and auxiliary cells**
- **[Optional] Add auxiliary cells to library**
- **Use existing digital synthesis and automatic place & route tools for physical design**

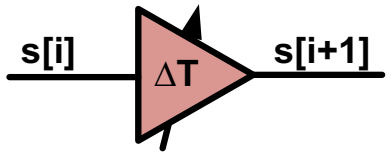




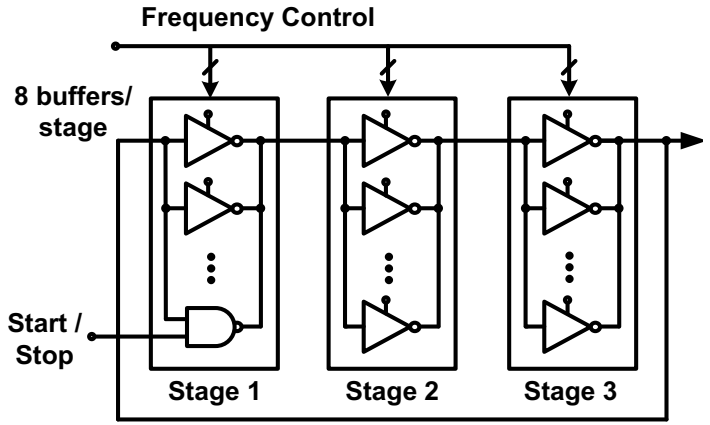
Example

Delay element using *N* tri-state buffers

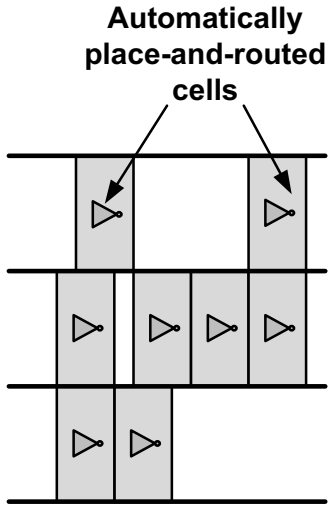


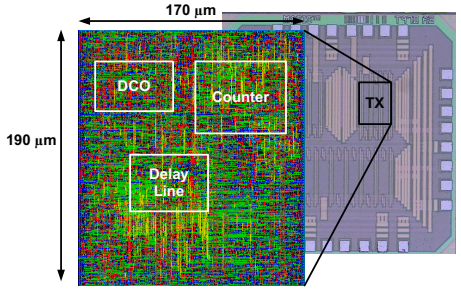


Structural Verilog to describe DCO with *S* stages



Synth & APR flow produces GDS



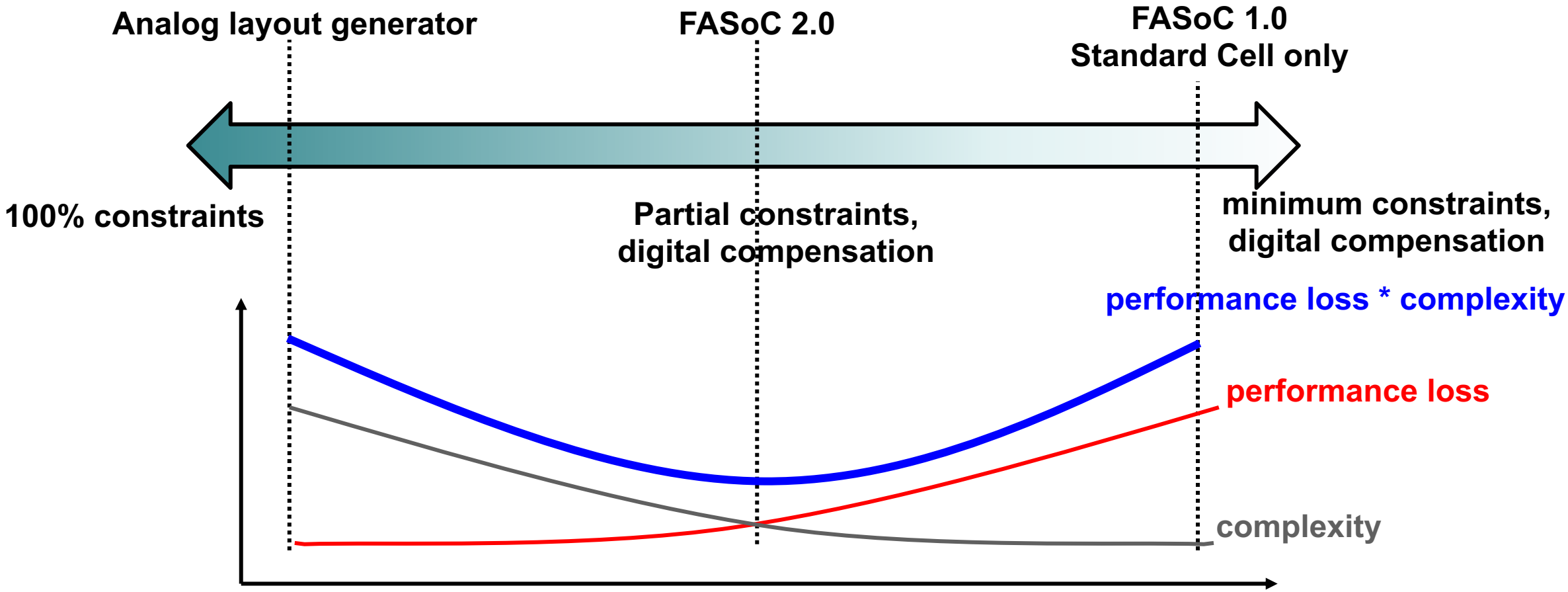


The Evolution of Cell-Based Design

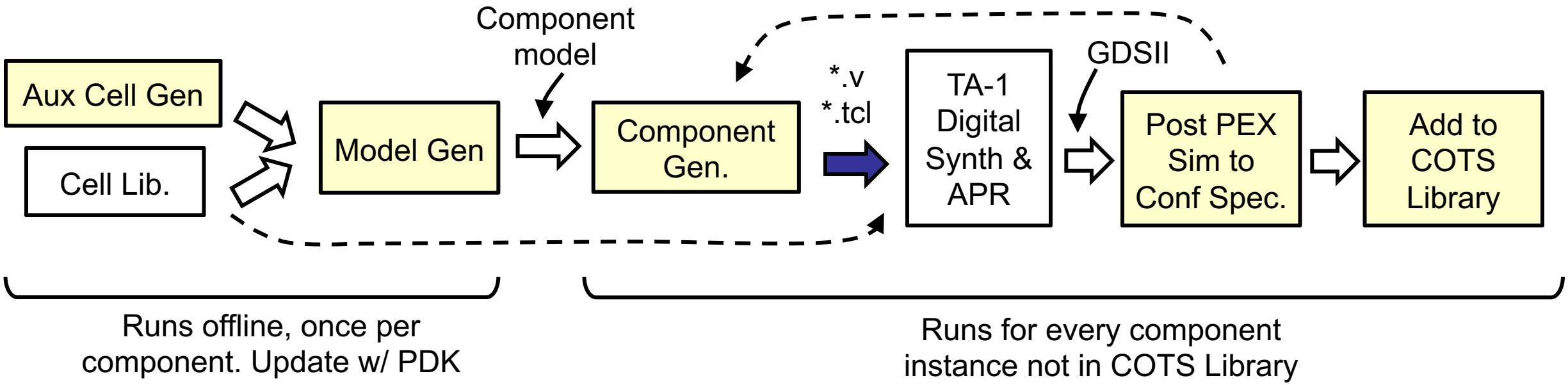
- **Standard tri-state buffer cells w/ unstructured layout (circa 2008)**
  - Simple, available in most standard cell libraries
  - Large variation in delay, and relative weights
  - Tuning range and resolution tradeoff is constrained
- **Differential tri-state and switch-cap auxiliary cells (circa 2011)**
  - Use P&R directives to structure layout (e.g. place stage cells together)
  - Use PWM to enhance resolution
- **Modeling to improve cell-based design flow (circa 2014)**
  - Numerical modeling to automate schematic design
- **FASoC – scripted cell placement for better performance (circa 2018)**
  - Added LDO, ADC, temp sense, memory, + more coming

Performance / Complexity Tradeoff

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



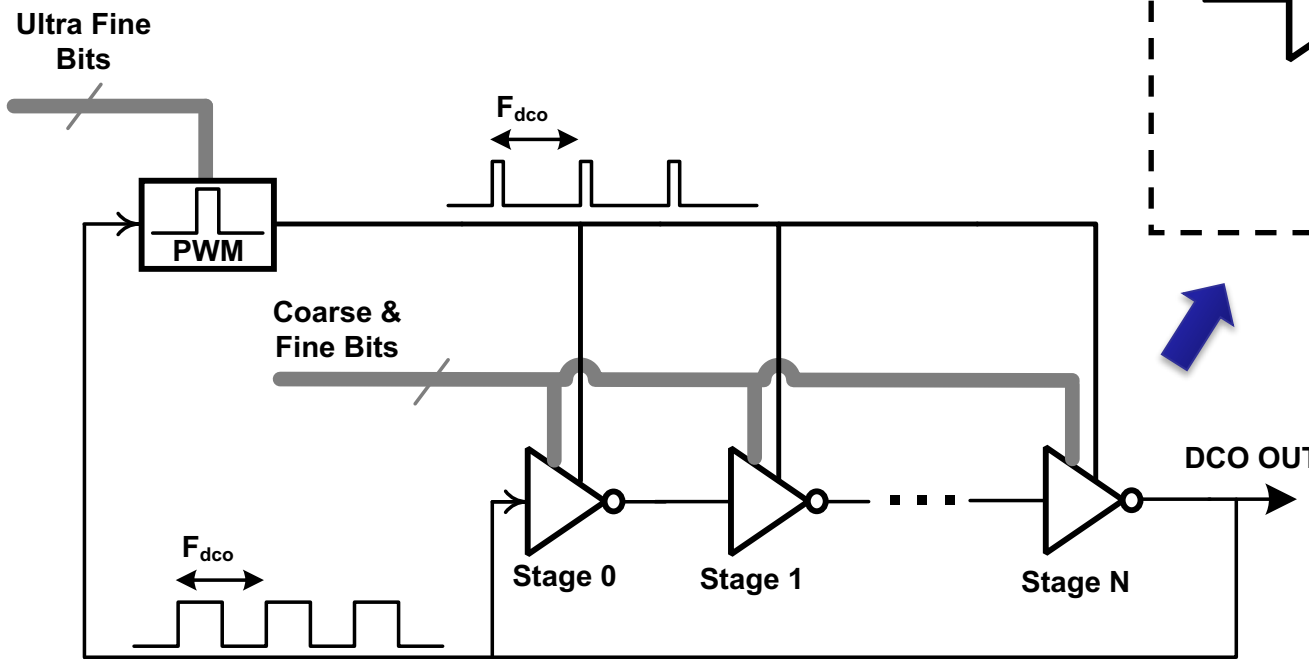
Cell-Based Analog Design Flow

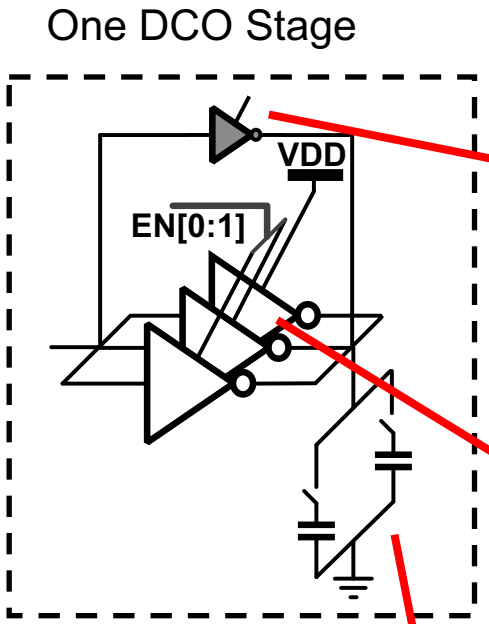


- **Same flow for ADPLL, ADC, CDC, DC/DC, LDO, Temp Sensor, Memory**
- **Structural and behavioral description of components**
- **Use "digital" synthesis and APR flow for physical design**
  - **No custom analog layout. No analog layout tool required.**

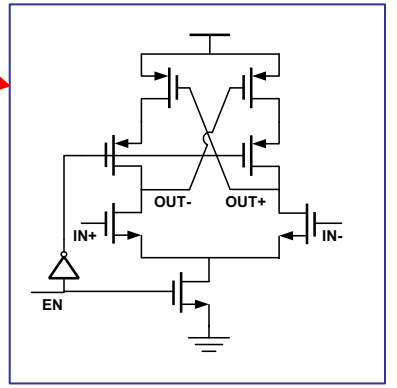
Differential DCO w/ PWM Fine-Tuning (circa 2011)

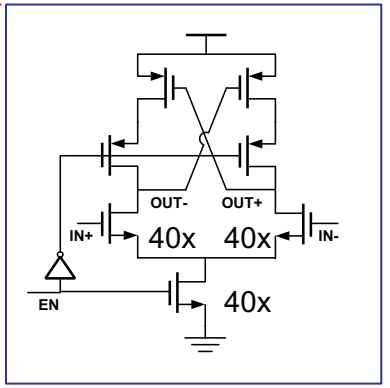
- Add auxiliary differential cells for better control of mismatch
- Two sized diff. current cells w/ feedback for symmetry
- Gate oxide cap cell w/ pass-gate

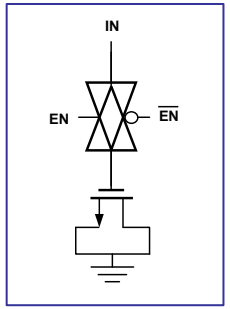




Aux Cells

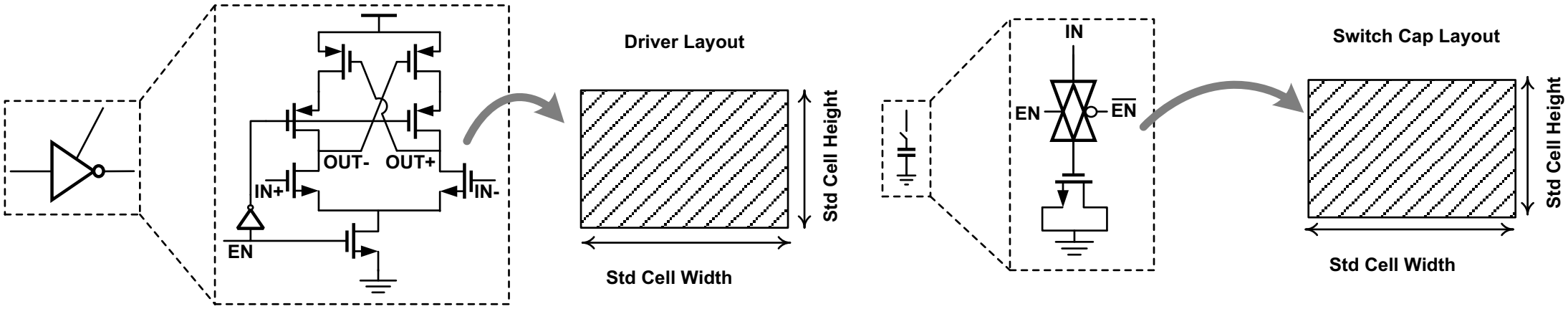






Design Methodology – Aux Cell Design

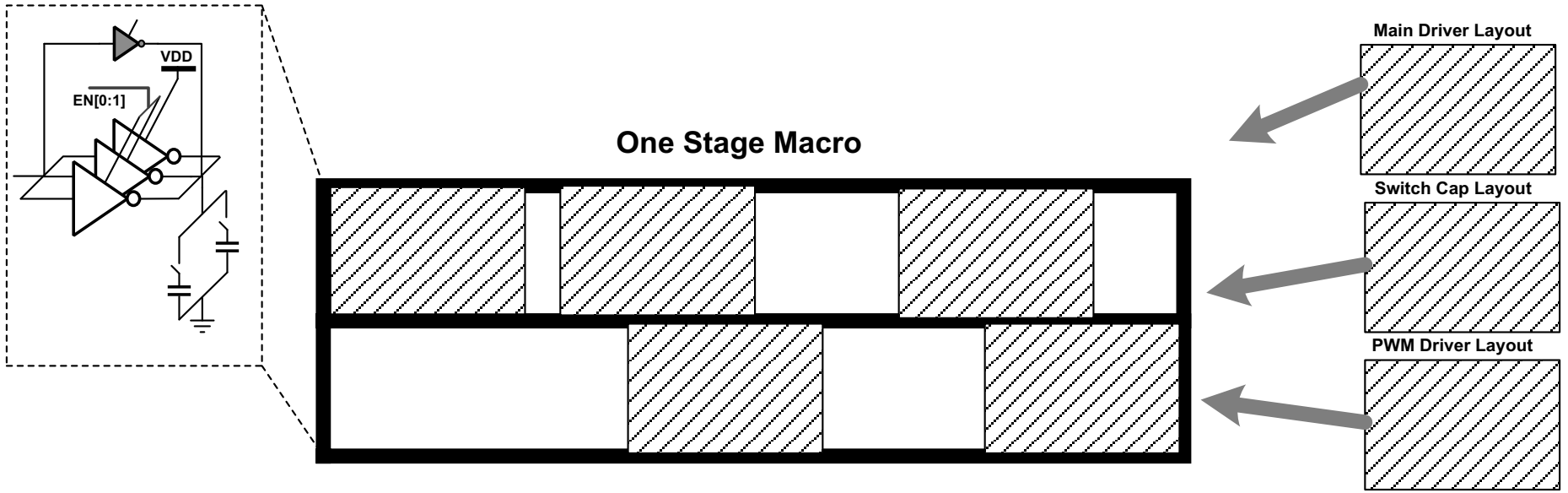
- Three custom cells in this prototype
  - Main DCO Driver
  - PWM Driver
  - Switch Cap
- Layout on standard cell grid and integrate with cell library



Cell layout automation with, for example, ALIGN or BAG

Design Methodology – Single Stage Marcos

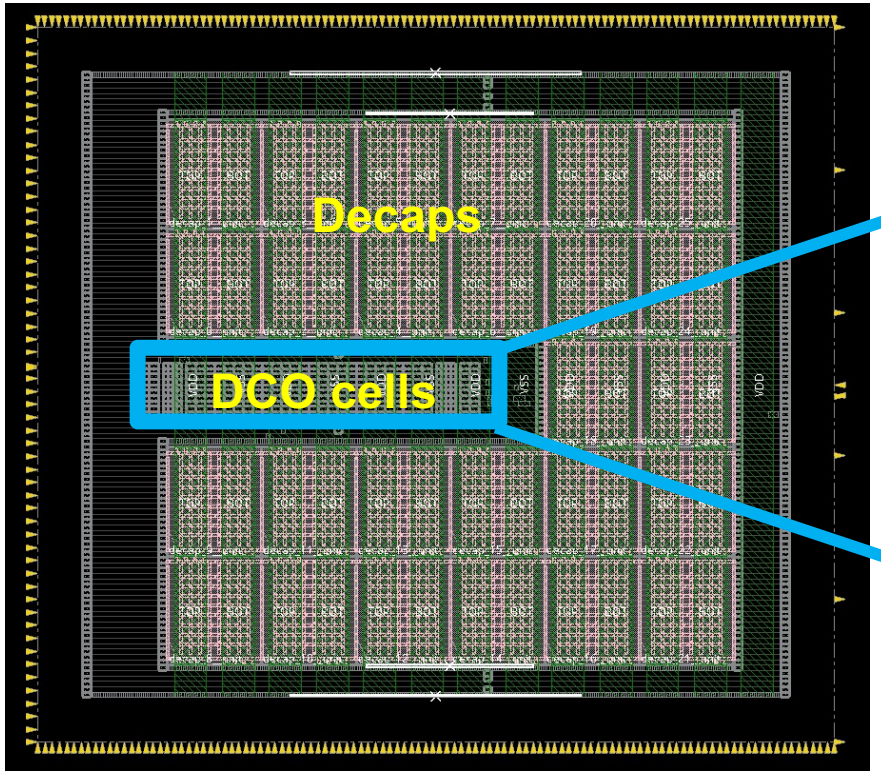
- APR single stage using macro (structured grouping of cells)
  - Standard capability of all P&R tools



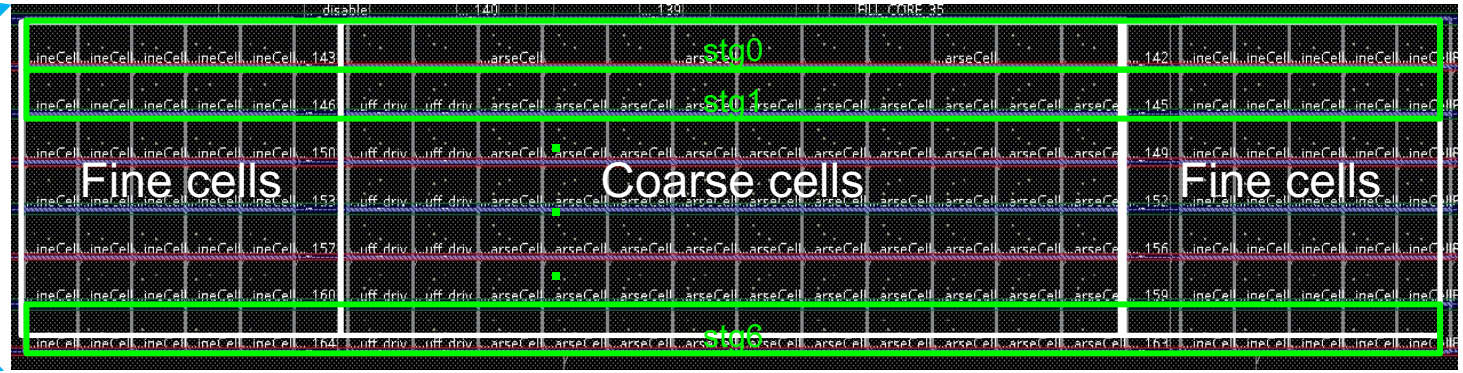
- Macros then assembled to form full DCO
- Entire APR process is scripted – no custom placement required

FASoC 2.0 – PLL example

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



DCO placements with Decap

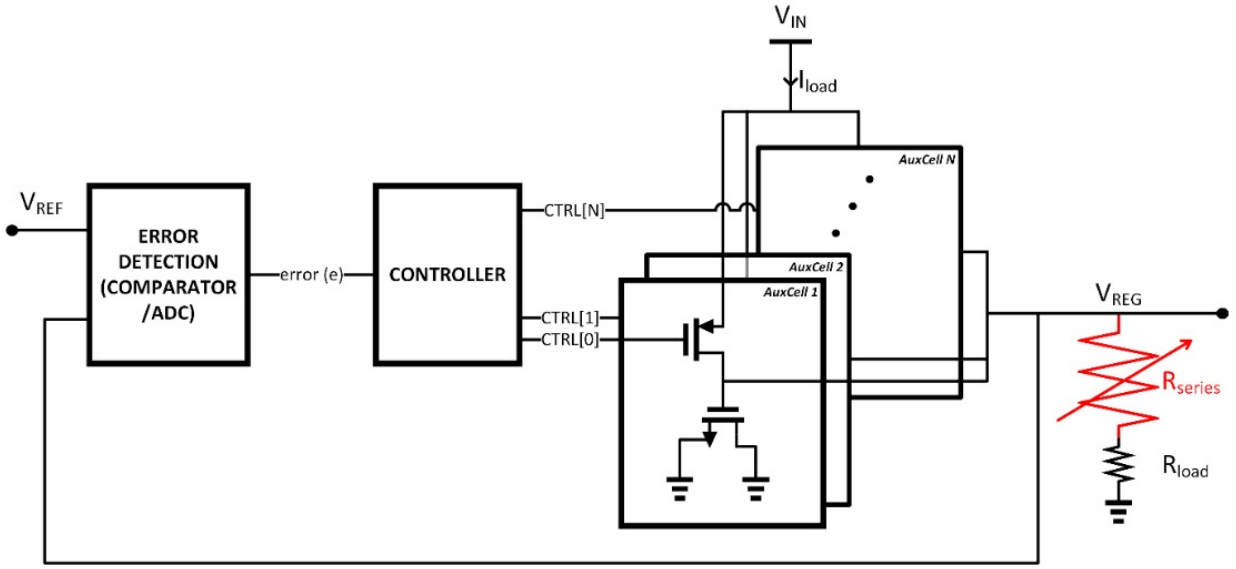


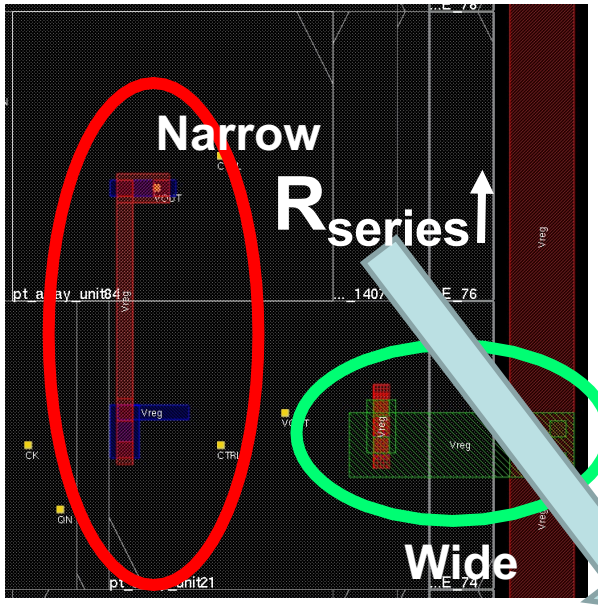
DCO cells

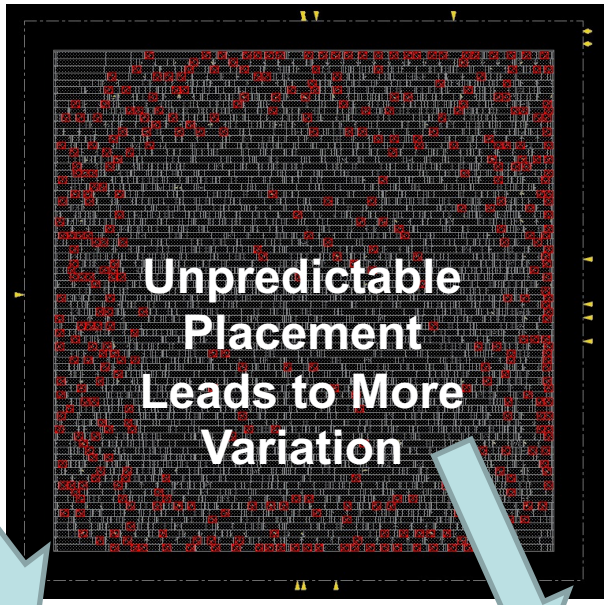
FASoC 2.0 – LDO example

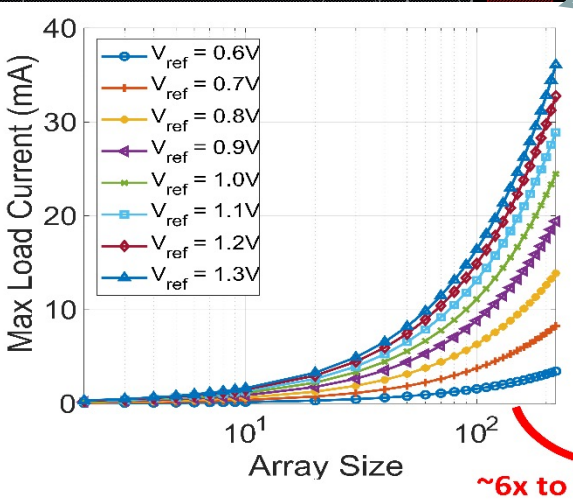
Performance loss caused by PnR

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

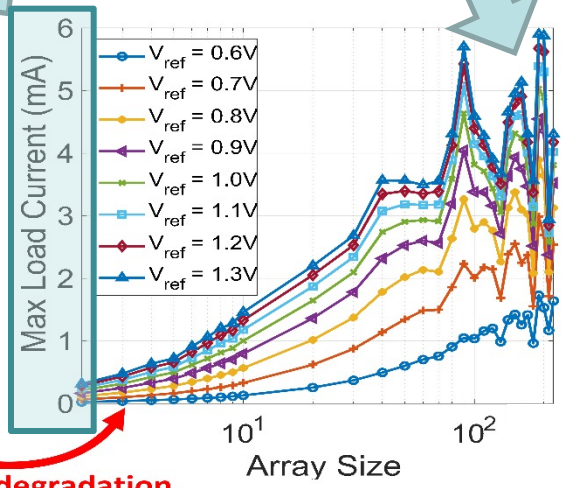








~6x to 9x degradation

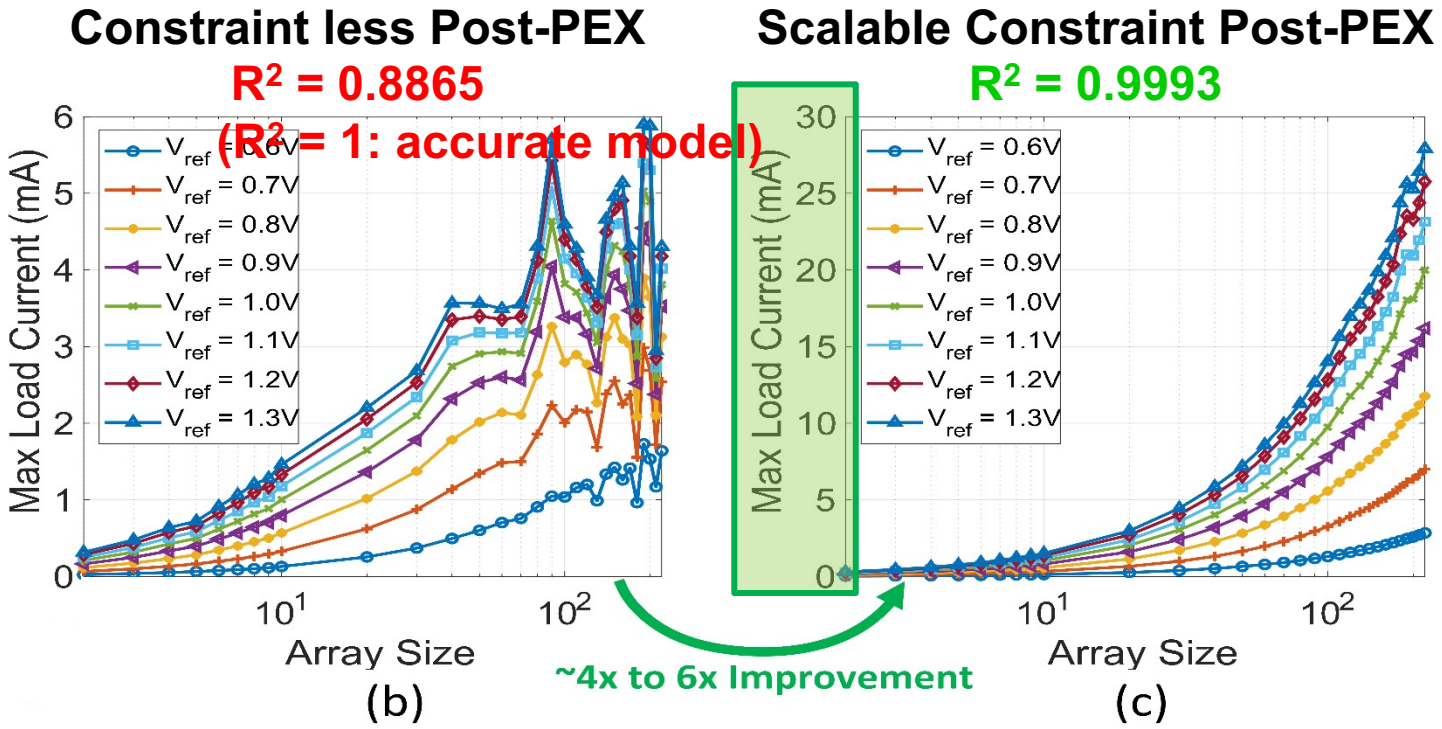


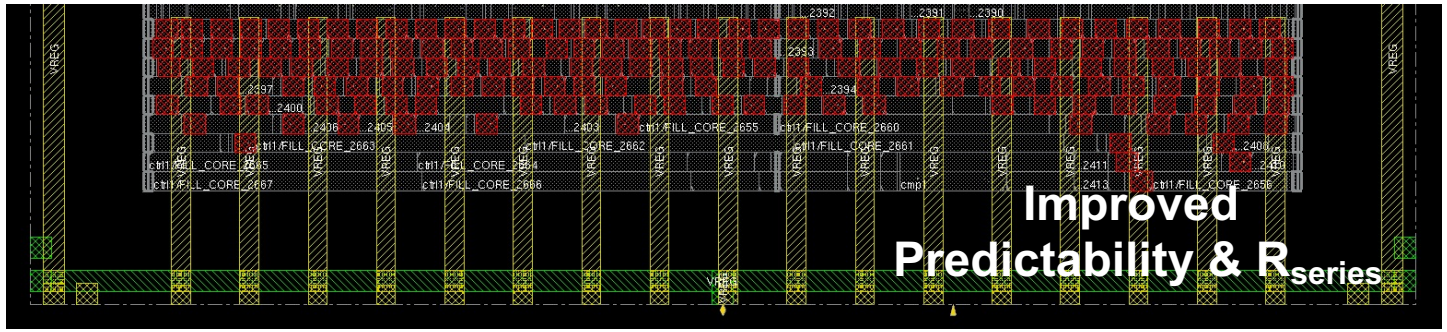
Pre-PEX **Post-PEX**

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**

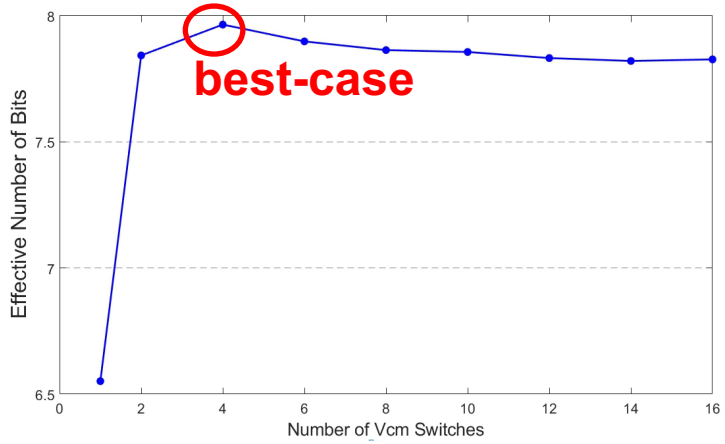




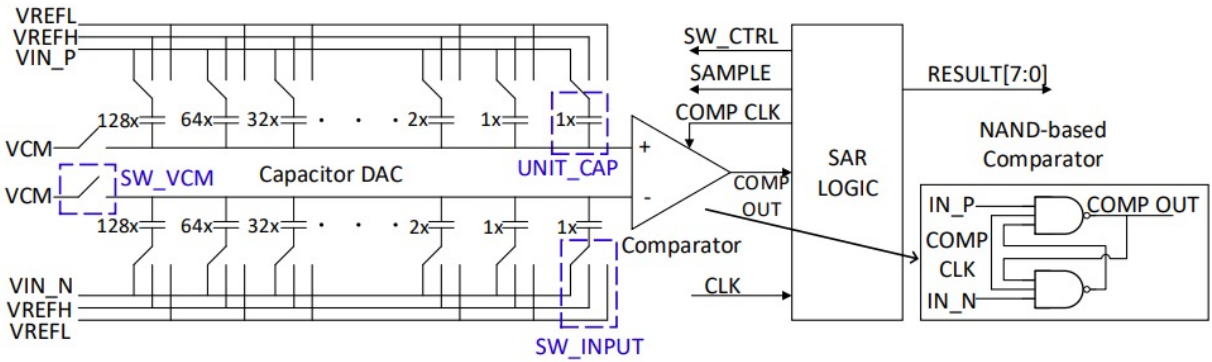
Added blocks: SAR ADC

- **Symmetrical Placement of unit caps and switches**

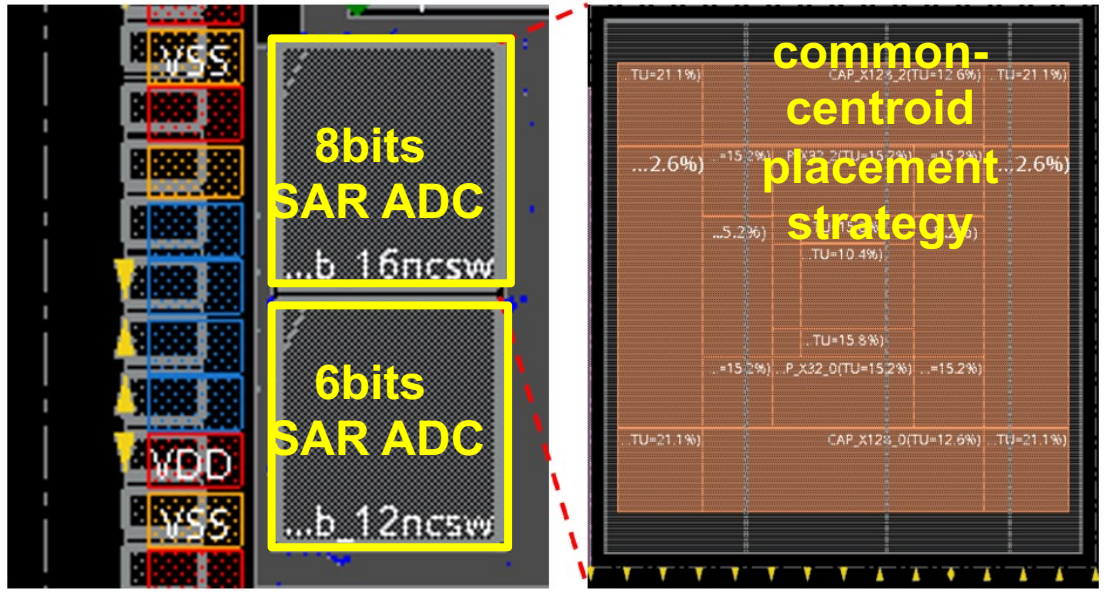
Output Spec.	CDL	PEX
$F_{SAMPLING}$ (MHz)	1	
Unit Cap Value (fF)	2.6	
Area (mm^2)	-	0.04
Power (μW)	6.72	11.2
Effective Number of Bits	7.86	7.75



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



SAR ADC Block Diagram



Bluetooth Transmitter

- **ADPLL (gen. by PLL-GEN) + LDO (gen. by LDO-GEN) + PA (custom)**

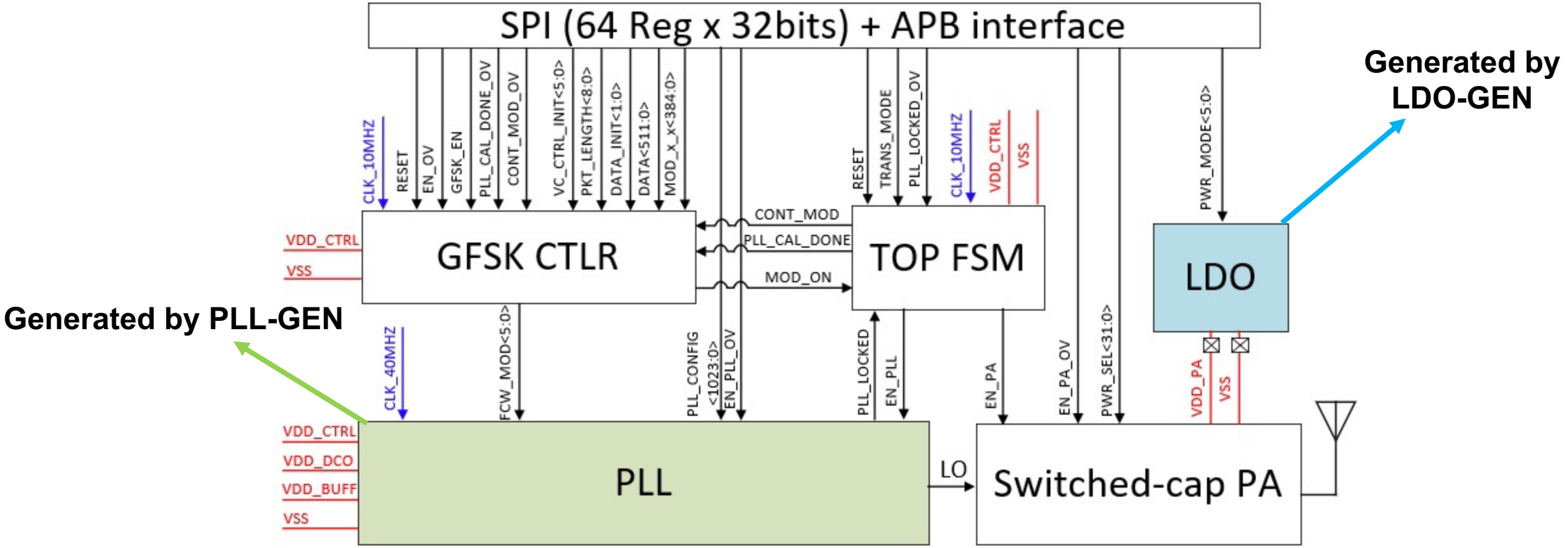
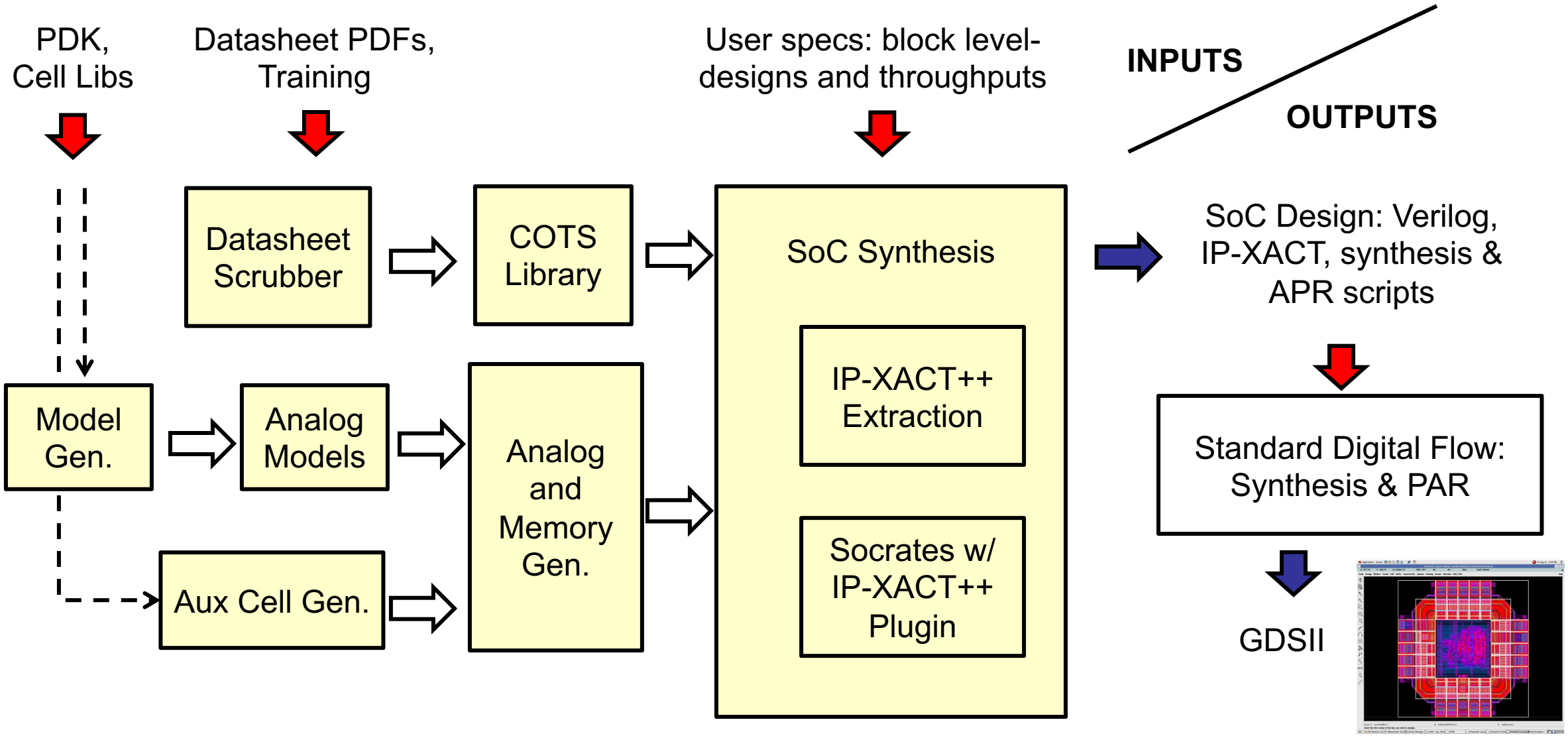
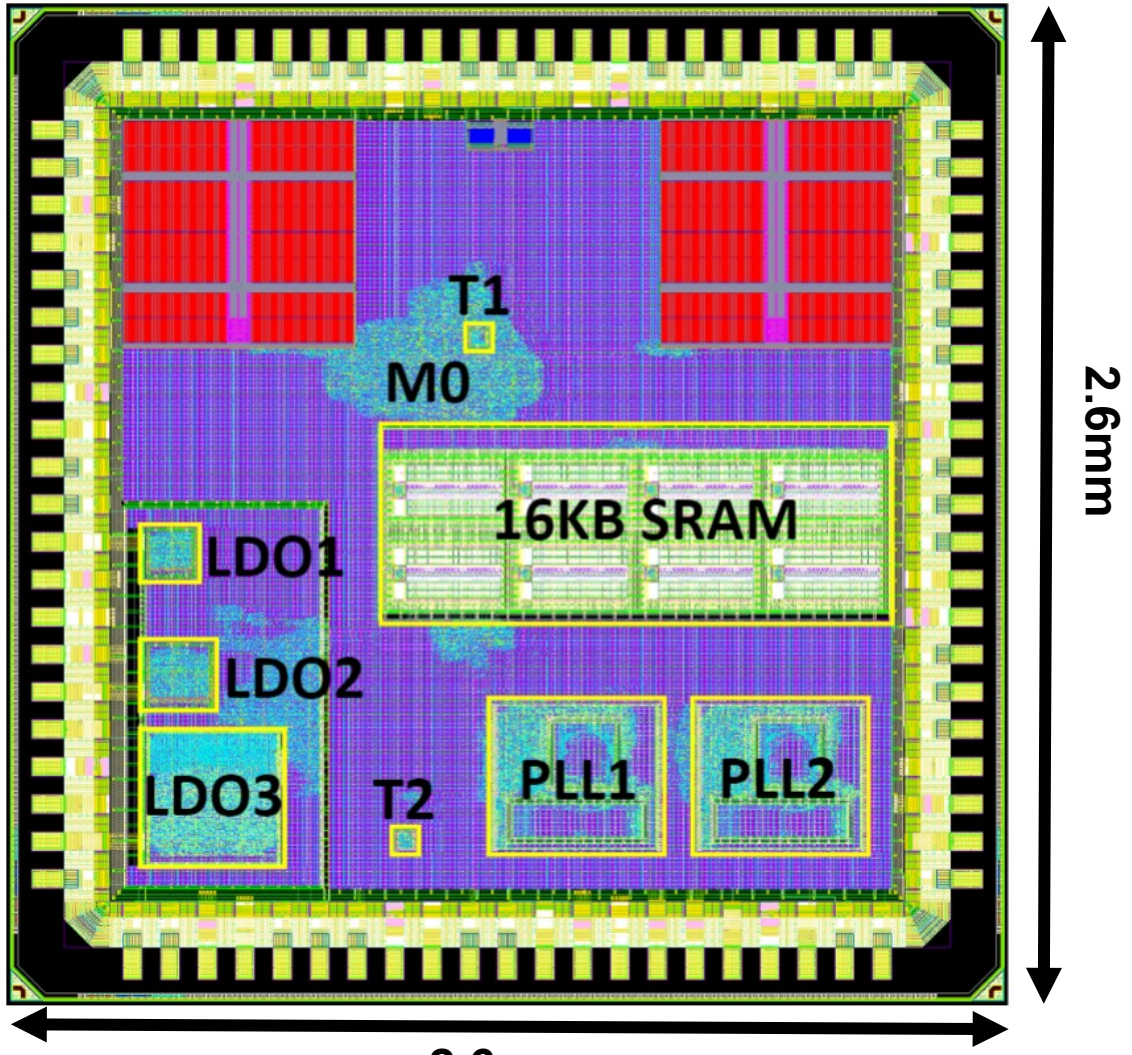


Figure. BLE top-level diagram

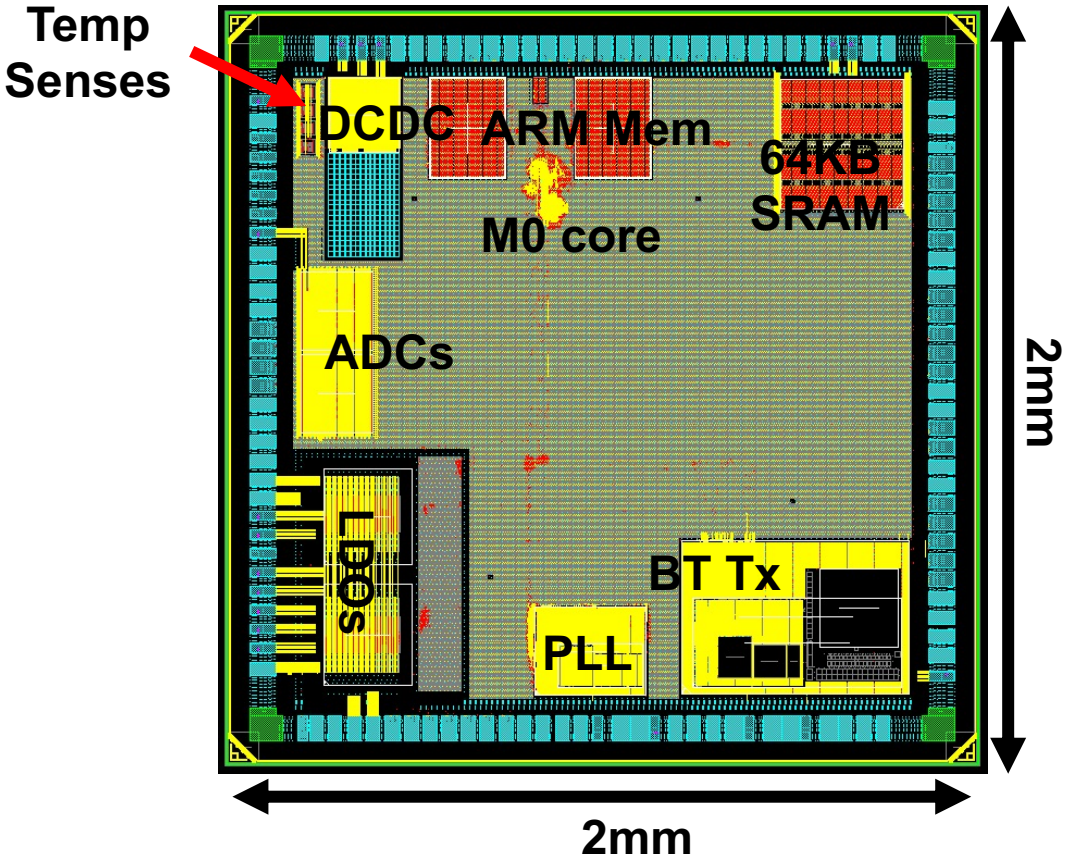
FASoC Tools Family and Flowchart



FASoC SoCs in TSMC 65 and GF 12



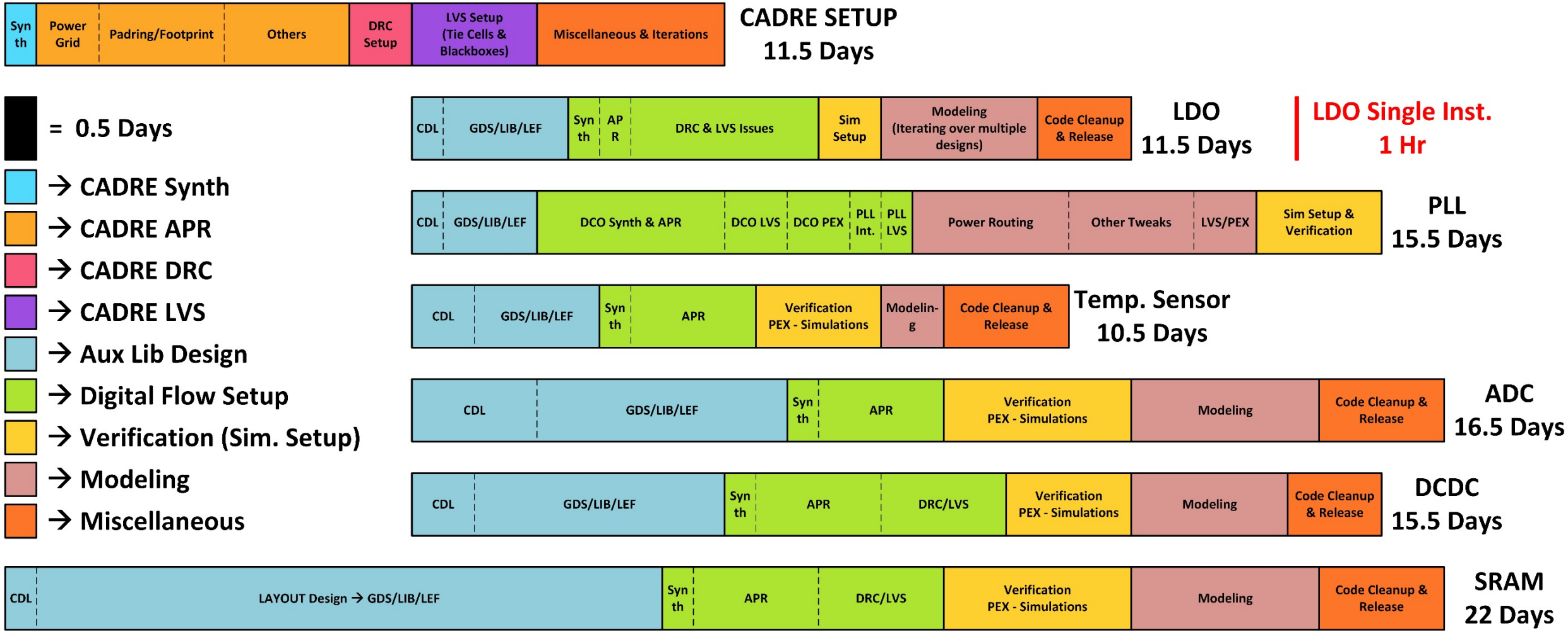
TSMC65LP SoC (2019-08)



GF12LP SoC (2020-10)

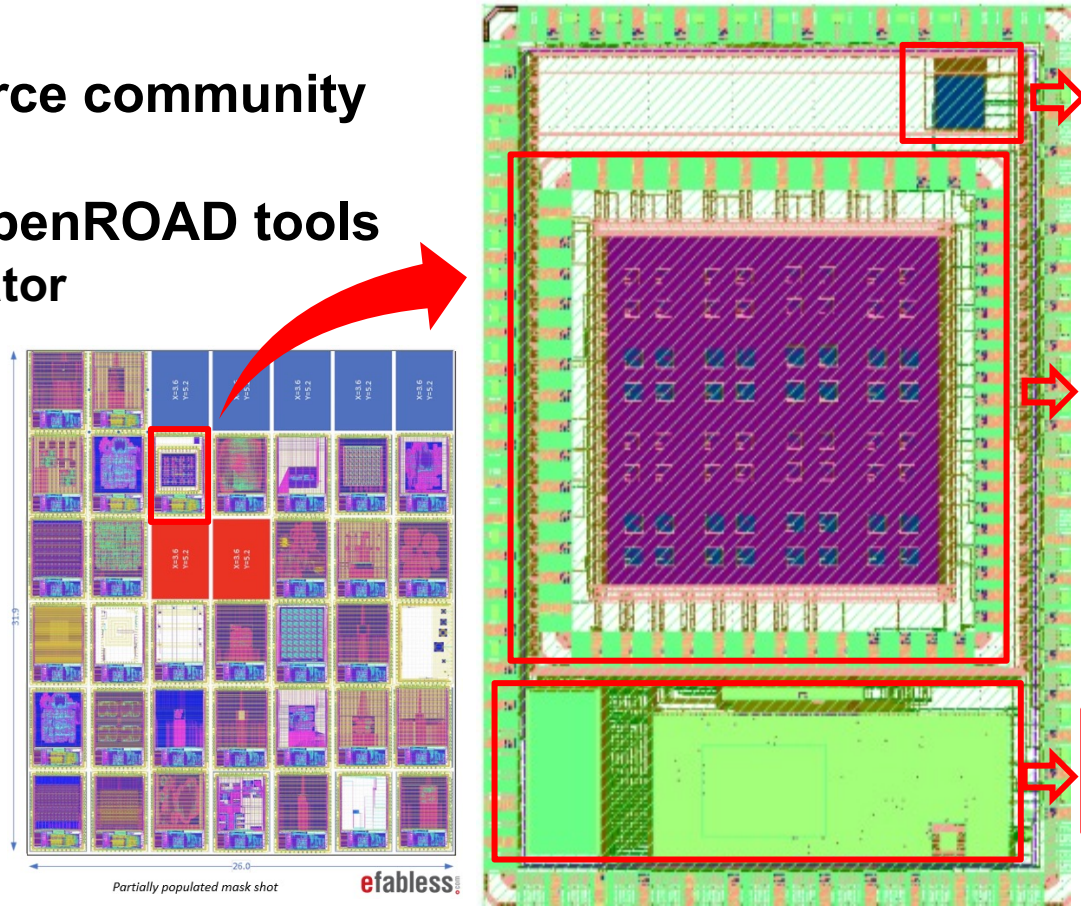
GF12 Porting Timeline – One Time Cost

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



SkyWater 130nm Effort and support of Google | Efabless

- **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 comparator to strongArm latch
    - 5v native NMOS switch



Test-chip in sky130 v3 – temp-sensors mesh SoC + LDO using the caravel SoC

- **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.

- **Main page: https://fasoc.engin.umich.edu/**
- **Git repo: https://github.com/idea-fasoc**
- **Google + eFabless**

Thanks!