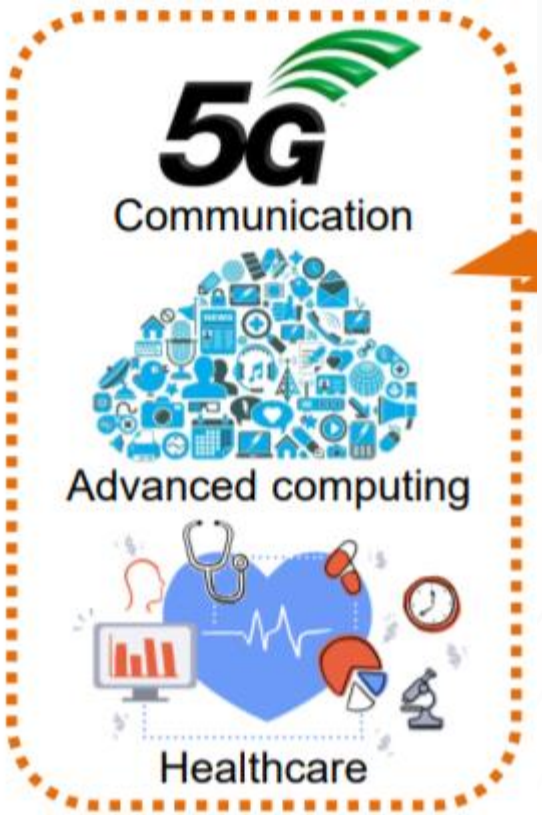
OpenSAR: An Open Source Automated End-to-end SAR ADC Compiler

Mingjie Liu, Xiyuan Tang, Keren Zhu, Hao Chen, Nan Sun, and David Z. Pan

The University of Texas at Austin

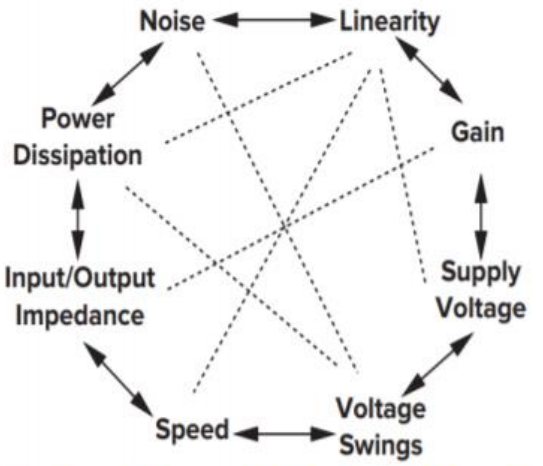
Introduction

- Analog-to-digital converters (ADCs)
  - Interface between analog front end to digital signal processing
  - Widely used from IoT, Automobile, Health, etc…
- Successive approximation register SAR ADCs
  - Digital friendly architecture
  - Great versatility: low power <-> high-speed
  - Moderate resolution and high energy efficiency



Analog Design and Challenges

- Heavily manual with limited automation
  - Compared with digital
- Difficulties
  - Complex performance trade-offs
  - Sensitive layout effects
  - No governing optimization targets



Courtesy [Razavi, Design of Analog IC]

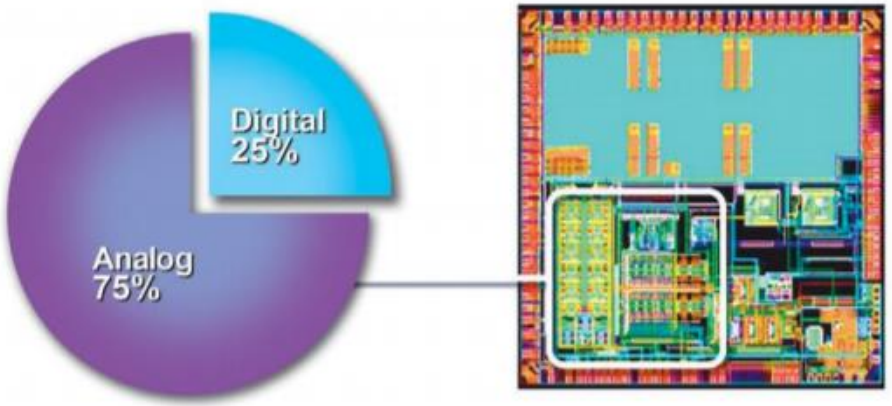


Image Sources: IBS and Dr. Handel Jones, 2012

Open Source EDA and Hardware

- Open source EDA and hardware are gaining momentum!
  - Skywater: 130nm PDK
  - OpenRoad (OpenLane): RTL to GDSII tool for digital
  - OpenRAM: Memory compiler
  - OpenSerDes: Serializer/Deserializer
  - FASoC: Autonomous SoC Synthesis with Cell-based Analog Circuits
- Analog circuits lacks open source hardware and designs:
  - Good for digital (mostly digital) designs: PLLs
  - Require additional support for custom cell designs: LDOs, etc..

Prior Work on SAR ADC Automation

- Fully digitalized design [Seo TCAS-II 2018]:
  - Digital (Standard cell based) analog circuits: bootstrap sampling and NAND-based comparator
  - CDAC compiler: layout descriptive language
  - Layout generation: digital APR tools with custom constraints
- Template base layout generation [Wulff JSSC 2017]:
  - Need manual definition of layout templates
  - Fast layout generation and turn-around time
  - Manual design and device sizing

Prior Work on SAR ADC Automation

- Hybrid design methodology [Ding TVLSI 2018]:
  - Equation-based sizing + simulation finetuning
  - Library based comparator design selection

- Weakness:
  - Restrictions on design: use of digital cells (only), library design
  - Manual efforts required: layout template designs
  - Knowledge-based sizing: limited to circuit topology, not transferable

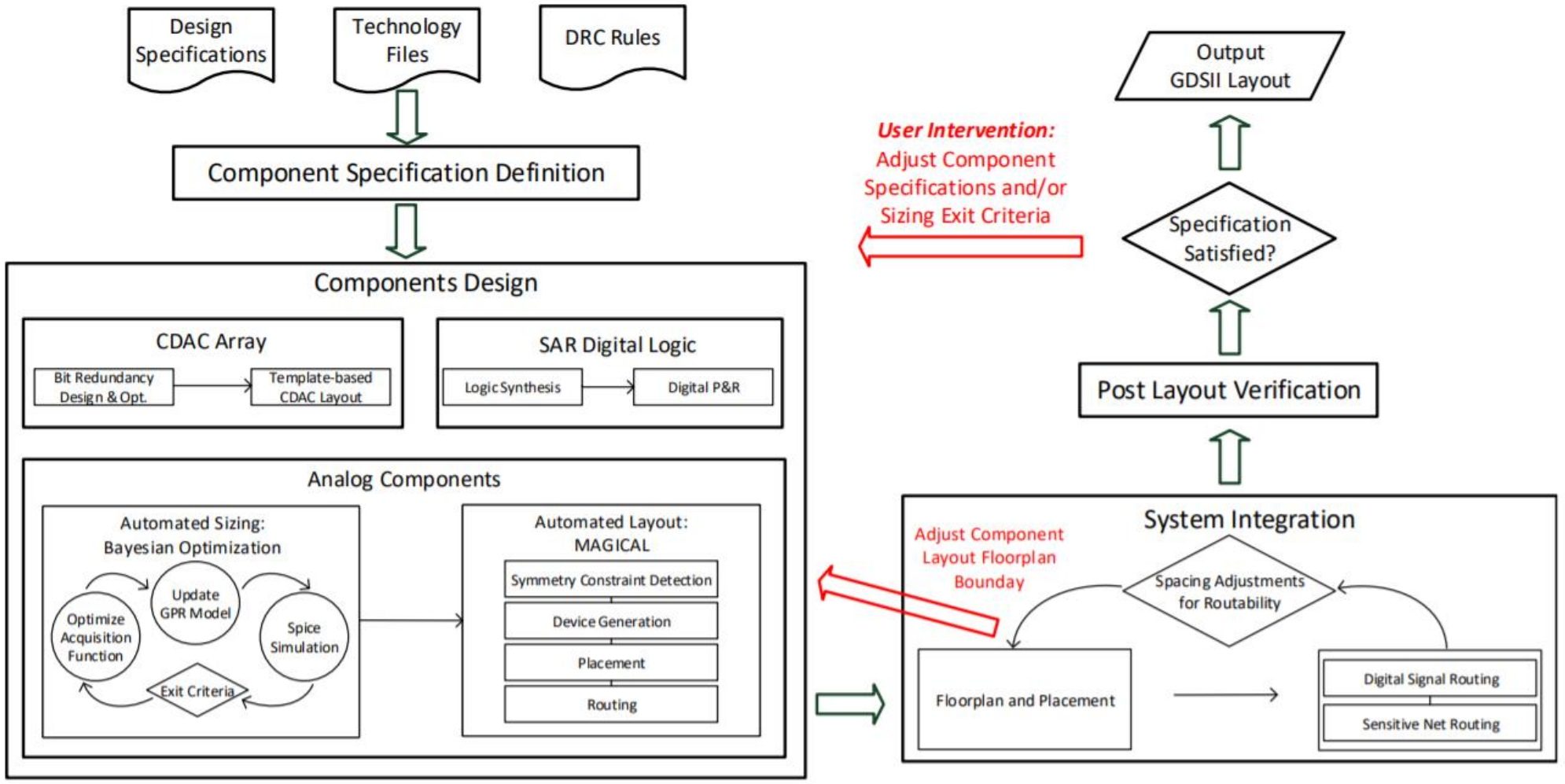
Our Contributions

- Ideally we want to achieve:
  - Flexible: Design methodology adapt to wide range of circuits
  - Automation: Little to no human in-the-loop
  - Quality: Achieve sign-off quality for tape-outs
- Our contributions:
  - Fully automated design generation: From Spec. to GDSII
  - Embed recent developments from analog sizing to layout generation
  - Achieves state-of-the-art performance based on post layout simulations
  - Open source framework for future design references

OpenSAR Framework

- ADC spec. to GDSII
  - (1) Spec. breakdown to components (design knowledge)
  - (2) Component design optimization
    - Yield aware CDAC array design with template-based layout generation
    - Bayesian optimization for analog component sizing (S&H, comparator)
    - Automated analog P&R engine with MAGICAL
  - (3) System layout integration
    - Floorplanning and component layout adjustments
    - Coupling aware dual stage routing

OpenSAR Framework



CDAC Redundancy for Yield

- Redundancy recovers errors of MSB decisions
  - Sum of Bits 0-2: 1+1+2+2=6 > Bit3: 4
  - Code 1000 = Code 0110 = 4
  - Redundancy for each bit needs to be positive

$$r_i = w_R + \sum_{j=0}^{i-1} w_j - w_i.$$

	Row Interleaved LSBs				Column Interleaved MSBs			
Bit Index (i)	R	0	1	2	3	4	5	6
Bit Weight (w)	1	1	2	2	4	6	12	12
Redundancy (r)	--	--	0	2	2	4	4	16
Design Variable (d)	1	1	2	2	4	1	2	2
Interleave Pattern	1	1	2	2	1	1	2	2

CDAC Redundancy for Yield

- Assume unit capacitor of C_u has std. error of σ_u
- Then r_i follows the following normal distribution

$$\mu_i = (w_R + \sum_{j=0}^{i-1} w_j - w_i) \cdot C_u$$

$$\sigma_i = \sigma_u C_u \cdot \sqrt{w_R^2 + \sum_{j=0}^{i} w_j^2}$$

CDAC Redundancy for Yield

- We can define

$$k_i = \frac{\mu_i}{\sigma_i} = \frac{w_R + \sum_{j=0}^{i-1} w_j - w_i}{\sigma_u \sqrt{w_R^2 + \sum_{j=0}^{i} w_j^2}}$$

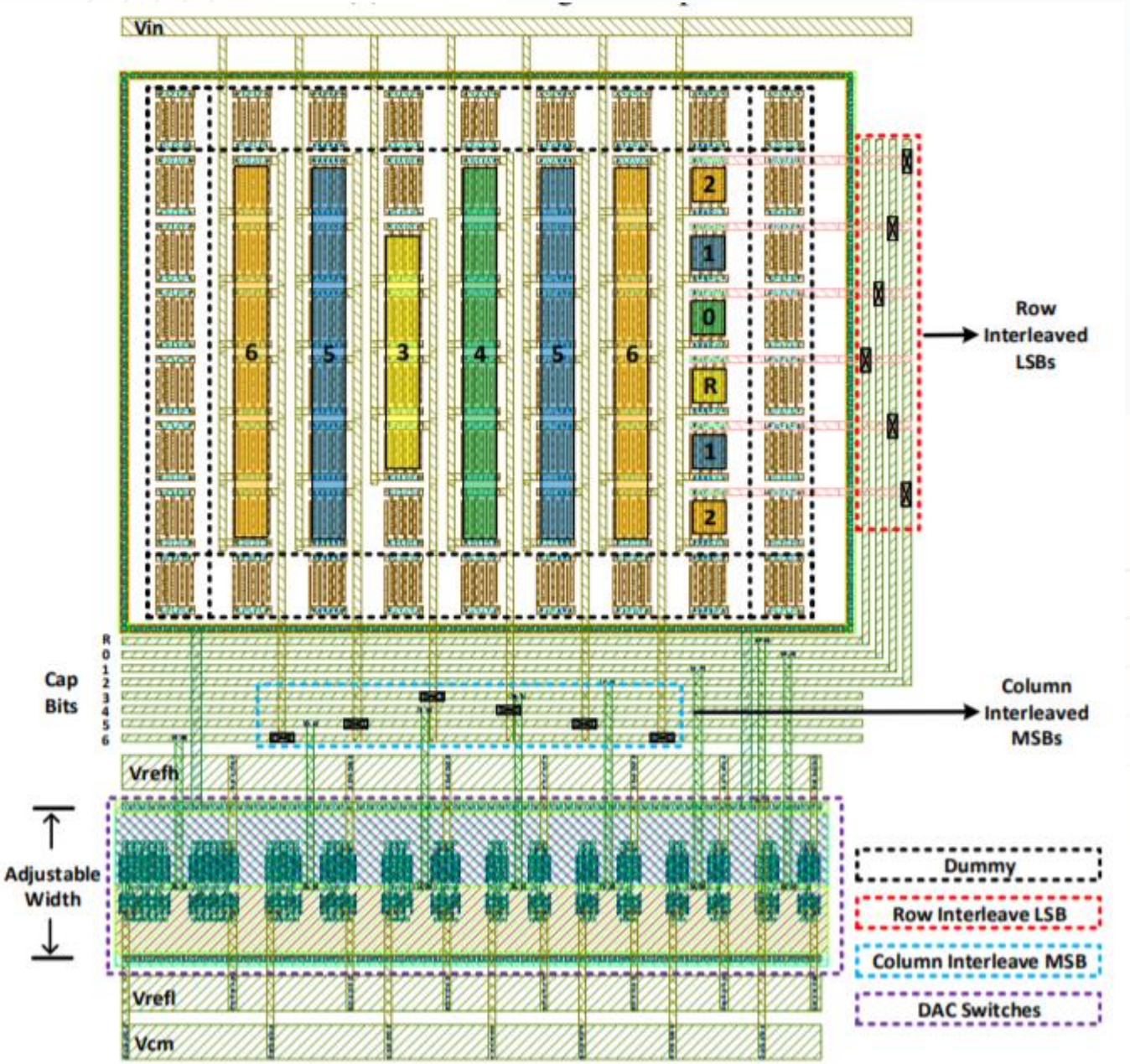
- Larger is better!

$$\max \min_{i \geq 2} k_i(w)$$

- Subject to ADC achieves resolution, and layout feasibility etc..

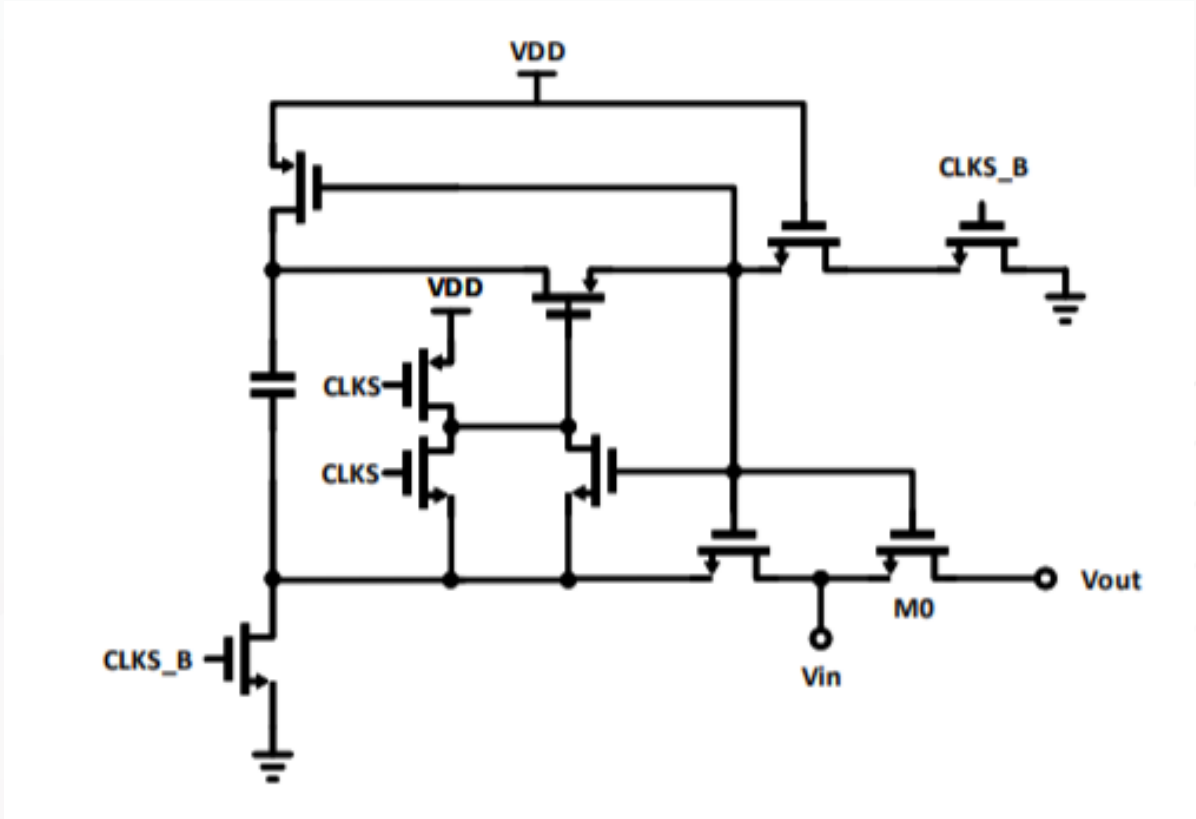
CDAC Design

- CDAC uses template-based layout generation
  - High layout quality and layout compactness
- Row and column based interleaving
  - Reduce random mismatch effects
- DAC switches: transient simulation ensure drive strength



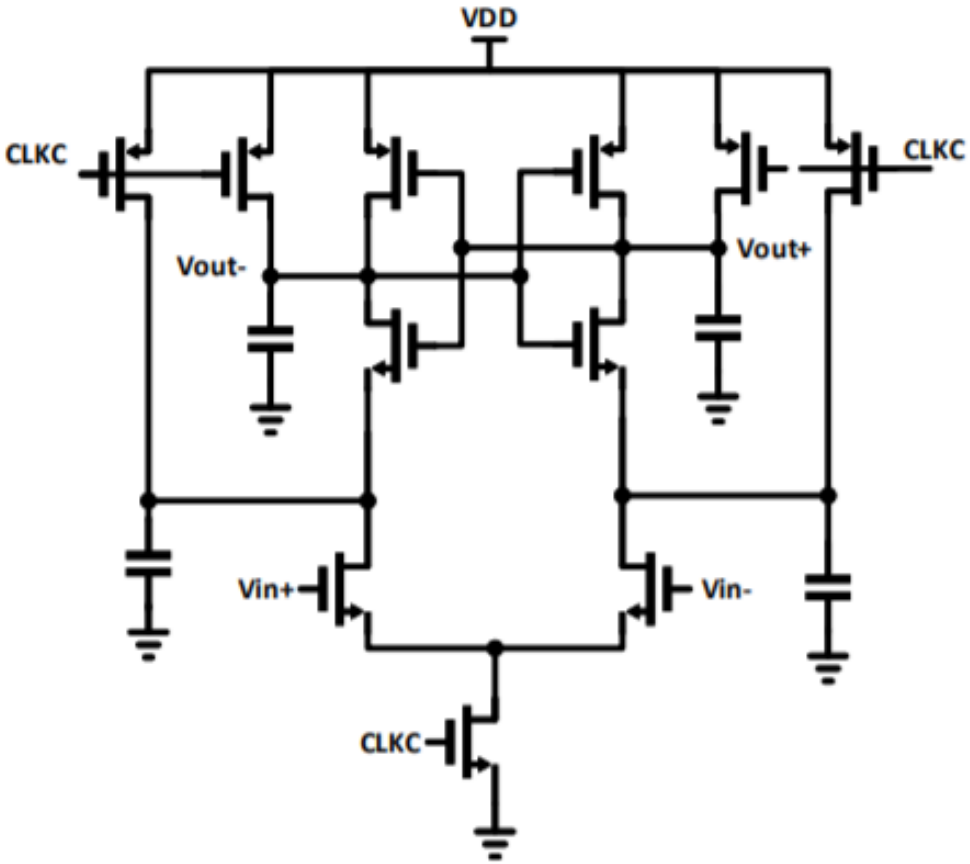
Sample and Hold Circuits (S&H)

- Linear sweep of transistor M0
- Ensure speed and linearity
- Other device sizing are fixed
- Automated analog P&R layout generation
- No additional efforts needed
  - Template design
  - Standard cells



Comparator Circuit

- Bayesian optimization for device sizing
- Ensure speed and noise requirements
- Minimize power consumption
- Automated analog P&R layout generation
- No additional efforts needed
  - Template design
  - Standard cells

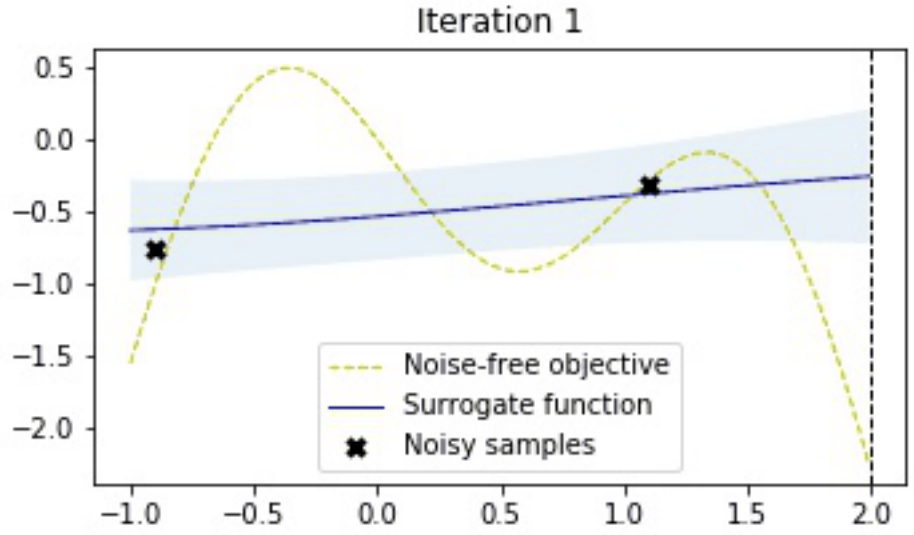


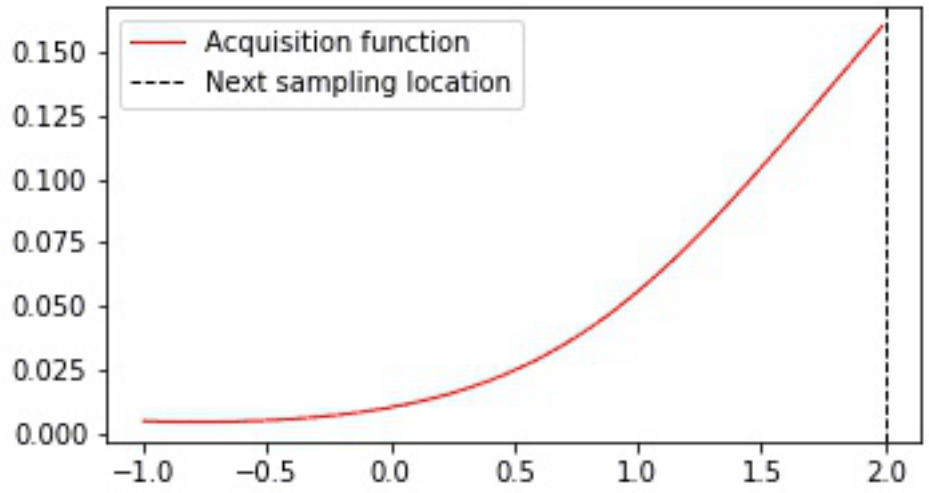
$$t_d < \frac{1}{2F_s N_{cycle}}$$

$$V_{noise} < \frac{1}{4} V_{LSB} = \frac{V_{DD}}{4\sqrt{2} \cdot 2^{N-1}}$$

Device Sizing with Bayesian Optimization

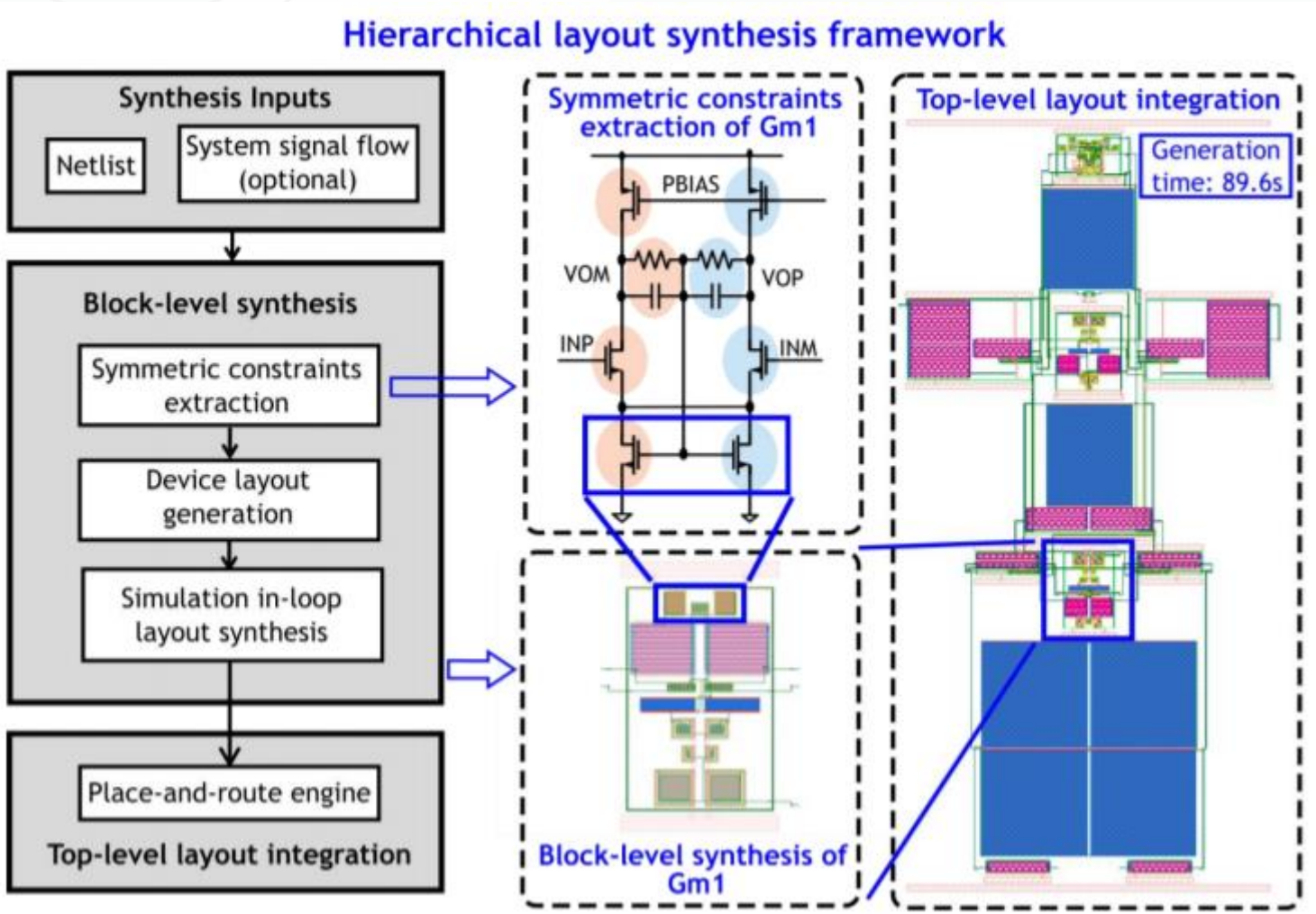
- Problem Formulation: Constrained Single-Objective
- Bayesian Optimization
  - Gaussian process regression: predict both mean and uncertainty
  - Acquisition function: determine where next to evaluate





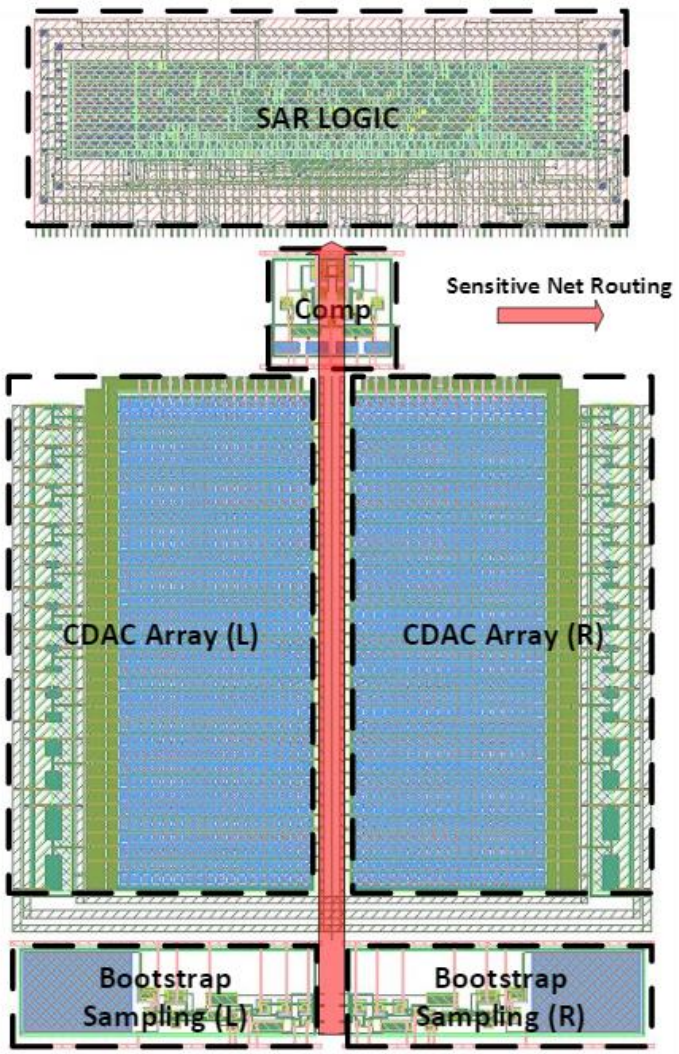
Automated Analog P&R with MAGICAL

- Hao et al. CICC'2021
- Fully automated layout synthesis
- Removes layout template design
- Fast turn-around time
- Silicon-proven results

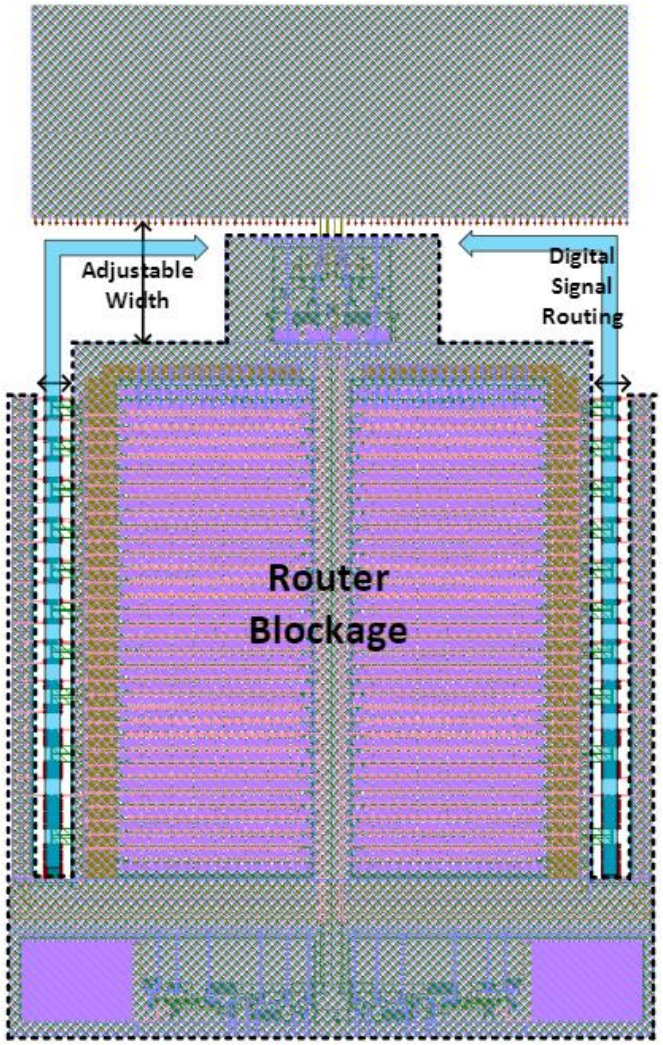


System Layout Integration

- SAR Logic: Digital tools
- Floorplan: Component layout adjustments
- Dual-stage Router:
  - Sensitive signal routed in middle first
  - Digital sigal routed with blockage
  - Mitigate coupling



(a) Floorplan and sensitive routing.



(b) Digital signal routing.

Experimental Results

- 2 SAR ADCs compiled in 40nm technology
  - High Speed Design: 10bit 100MS/s @ 1.2V
  - High Resolution Design: 12bit 1MS/s @ 0.7V
- Fast compilation
  - 2 hour total design time
  - 10min layout turnaround
  - Majority spent on simulations for sizing

TABLE II: Runtime Breakdown

	10bit		12bit	
	Design	Layout	Design	Layout
DAC Switch	6m33.6s	-	7m51.7s	-
CDAC Array	6.4s	11.7s	42.0s	39.1s
Bootstrap Switch	-	6.9s	-	8.4s
Comparator	1h45m47.3s	4.7s	1h52m32.7s	6.1s
SAR Logic	15.5s	2m56.1s	20.8s	2m49.3s
Top Floorplan	-	26.2s	-	1m4.2s
Top Routing	-	1m40.3s	-	3m15.6s
Total	1h52m42.8s	4m59.7s	2h1m27.2s	8m2.7s

Experimental Results

TABLE III: Performance Comparison with Prior Work on SAR ADC Automation

	Huang† [23]		Seo† [22]		Wulff† [16]		Ding† [17]		OpenSAR*	
Auto-Design	Yes		Yes (Digital)		No		Yes		Yes	
Auto-Layout	No		Yes		Yes		Yes		Yes	
Resolution	12	10	12	11	9		8	12	10	12
Technology (nm)	180	90	180	28	28 FDSOI		40		40	
ENOB	10.1	8.4	10.2	9.1	7.4	7.8	7.6	9.9	9.1	11.1
SNDR (dB)	62.7	52.1	63.3	56.8	46.4	48.8	47.4	61.1	56.3	68.8
BW(MHz)	0.3	25	0.05	25	1	10	16	0.5	50	0.5
FOM_S (dB)	152.2	159.7	155.3	164.8	166.8	166.8	156.7	165.8	166.8	176.0
FOM_W (fJ/c.step)	500	26.7	265.5	14.1	2.7	3.5	30.7	18.1	10.8	4.3

*Simulated results with layout R+C+CC parasitic extraction. †Tapeout measurement.

- Post layout simulation results:
  - Better resolution, higher speed
  - State-of-the-art figure of merits (FOMs)

Conclusions

- Embedding recent developments in analog CAD
- Custom circuit design with enhanced automation
  - Automation: removing manual efforts in sizing and layout
  - Flexible: not specified to certain design/topology etc..
  - Quality: DRC/LVS clean with verified post layout simulation
- Serve as a design reference for future agile IP design
- Scaling from component circuits into larger systems

Conclusions

- OpenSAR: https://github.com/magical-eda/OpenSAR
- MAGICAL: https://github.com/magical-eda/MAGICAL
- Future works
  - Missing files due to NDA restrictions
  - Working towards adapting to Skywater 130nm
  - Tape-out and silicon measurements

Thank You!