

Automated and Process-Portable Generation of Data Converters

Zhaokai Liu

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2025-21

<http://www2.eecs.berkeley.edu/Pubs/TechRpts/2025/EECS-2025-21.html>

May 1, 2025



Copyright © 2025, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

Automated and Process-Portable Generation of Data Converters

By

Zhaokai Liu

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering - Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Borivoje Nikolić, Chair

Professor Vladimir Stojanović

Professor Martin White

Summer 2023

Automated and Process-Portable Generation of Data Converters

Copyright 2023
by
Zhaokai Liu

Abstract

Automated and Process-Portable Generation of Data Converters

by

Zhaokai Liu

Doctor of Philosophy in Engineering - Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Borivoje Nikolić, Chair

High-speed analog-to-digital converters (ADCs) are critical components in wideband wireline and wireless communication systems. Rapid advancements in communication systems require medium-to-high resolution ADCs that can digitize a wide spectrum with high power efficiency. In addition to the strict performance demands, the design of ADC, as analog and mixed-signal circuits becomes progressively more complex and time-consuming at advanced technology nodes. The design challenges arise not only from the scaled supply voltage and increased variation, but also from the more complex layout, the explosion of design rules, and increased vulnerability to parasitic effects. To address these challenges, this dissertation presents an automated and process-portable ADC generator framework capable of creating instances that support a sampling rate of up to 4 GS/s and achieve a resolution of up to 9 effective number of bits (ENOB). The generator is developed using Berkeley Analog Generator (BAG), which enables automated circuit generation, speeds up the design iteration, and significantly improves design reusability.

To explore circuit-level techniques suitable for scaled processes with high programmability and reusability, both traditional voltage domain and time domain data conversion are explored. The proposed generator incorporates the successive approximation register (SAR) architecture, which has become prevalent in the field due to its ability to eliminate the need for precision analog components. Furthermore, the utilization of time-based data conversion techniques, such as ring oscillator-based voltage-controlled oscillator (VCO) data conversion and ring amplifiers, has shown promising results in terms of design metrics for implementing precise analog functions in scaled technologies. In the proposed generator, the VCO-based ADC is used for second-stage fine conversion, while the ring amplifier is used for residue amplification. A process-portable automated ADC generator, which supports a wide range of specifications, is developed by integrating and selectively enabling these techniques. The standalone SAR ADC generator has been ported to various processes and utilized to create designs with diverse specifications, showcasing the effectiveness of the generator-based

methodology. The complete circuit architecture of the proposed ADC generator, which achieves maximum performance, is based on a time-interleaved (TI) subranging ADC array. The sub-ADC uses a pipelined topology that combines both the SAR and the VCO-based ADCs to enhance the resolution. The ADC generator is fully automated and parameterized, generating designs that are compliant with design rules based on input parameters.

Finally, this thesis exemplifies the effectiveness of the generator-based design methodology through the creation of multiple generated prototypes of time-interleaved ADC designs using both BAG2 and BAG3 frameworks. As the main focus of this thesis, two prototypes were generated using the proposed generator to implement TI SAR-VCO ADCs with 4 and 8 channels, respectively. The 4-way interleaved design implemented in the Intel 22FFL process samples the input at a rate of 2 GS/s. And the 8-way time-interleaved prototype samples at 4 GS/s and is implemented using the Intel 16 process. The measurement setup and results of the latest prototype chip are presented to demonstrate the performance of the proposed ADC generator. The ADC achieves a peak SFDR of 72 dB and a resolution of over 9 bits within the 2 GHz Nyquist band. The total power consumption of the prototype under a 0.9 V supply is 124.6 mW. The prototype achieves a Schreier figure-of-merit (FOM) of 158.4 dB and a Walden figure-of-merit of 60.5 fJ/conv.-step. In summary, this thesis presents both circuit techniques and analog design automation. The proposed generator demonstrates promising ways of implementing automated and process-portable ADC designs with high reconfigurability using a generator-based methodology.

Contents

Contents	i
List of Figures	iii
1 Introduction	1
1.1 Motivation	1
1.2 Research Goals	5
1.3 Thesis Organization	7
2 High-Speed ADC Design	8
2.1 Introduction	8
2.2 Traditional ADC Architecture for High-Speed Operation	9
2.3 Time-Domain Data Conversion and ADC Architectures	12
2.4 Hybrid ADC Architectures	28
2.5 Time-Interleaved ADCs	30
2.6 Summary	35
3 Analog Design Automation and BAG Workflow	37
3.1 Introduction	37
3.2 Analog Circuit Automation	37
3.3 BAG Framework Overview	39
3.4 Schematic and Layout Generation	41
3.5 Design and Optimization Using BAG	44
3.6 Implementation of Generator-Based Design Flow and Prototype	45
3.7 Summary	46
4 Building Blocks of the Proposed ADC Generator	47
4.1 Introduction	47
4.2 Sampling Circuit Generator	49
4.3 SAR ADC Generator	60
4.4 Design of the VCO-based ADC Generator	82
4.5 Residue Amplification and Ring Amplifiers	95

4.6	Auxiliary Circuits	98
4.7	Summary	102
5	Generated Prototypes	104
5.1	Overview	104
5.2	LAYGO Time-Interleaved ADC Prototypes	105
5.3	Time-Interleaved SAR-VCO ADC Prototypes	108
5.4	Measurement	117
5.5	Measurement Results	120
6	Conclusion	127
6.1	Key Accomplishments	127
6.2	Future Work	128
	Bibliography	129
A	Layout Generation Engines in BAG2 and BAG3	146
A.1	LAYGO	146
A.2	XBase in BAG2	148
A.3	XBase in BAG3	150
B	Generator Examples	151
B.1	An Inverter Generator Example	151
B.2	Comparator Generator Examples	152

List of Figures

1.1	Diagrams of common receiver architectures: (a) superheterodyne, (b) direct conversion, and (c) direct RF sampling.	2
1.2	High-speed serial link transceivers with (a) a mixed-signal receiver architecture and (b) an ADC DSP-based receiver architecture.	2
1.3	Popular ADC architectures and the target coverage range of the proposed generator.	3
1.4	(a) Supply and threshold voltage vs. process nodes [9]. (b) The average growth in the number of design rules and the number of DRC operations over node progressions [10].	4
1.5	Diagram of a typical analog circuit design flow.	5
1.6	Diagram of the generator-based design flow using BAG.	6
2.1	Diagrams of (a) a flash ADC, (b) the resistive interpolation method, and (c) interpolation transfer curves.	9
2.2	Diagrams of (a) a typical SAR ADC and (b) its capacitive DAC.	10
2.3	(a) Diagram of a typical pipelined ADC and (b) the residue voltage transfer curve.	11
2.4	Diagrams of (a) a time-biased ADC consisting of a VTC and TDC and (b) a VCO-based ADC.	13
2.5	(a) Schematic of a typical differential VTC using current-starved inverters and (b) its timing diagram [35].	14
2.6	Diagrams of (a) a basic delay-line-based TDC and (b) a differential delay-line-based TDC that doubles the resolution.	15
2.7	Diagrams of (a) a recirculating TDC and (b) a TDC based on a delay-locked loop.	16
2.8	Diagrams of (a) a basic Vernier TDC and (b) a pulse-shrinking TDC.	17
2.9	Schematics of delay lines with (a) the passive interpolation and (b) the active interpolation. And the example waveforms of (c) the passive interpolation and (d) the active interpolation.	18
2.10	Schematic of resistively-coupled ring oscillators use both active and passive interpolations [48].	18
2.11	Schematics of stochastic TDCs with (a) sampling offsets and (b) unbalanced samplers.	19
2.12	(a) High-level block diagram of the one-phase VCO-based ADC and (b) its equivalent model.	20

2.13	Illustration of phase quantization scheme in the VCO-based ADC.	21
2.14	Diagrams of (a) the multiphase open-loop VCO-based ADC and (b) the open-loop VCO-based ADC structure with coarse-fine phase quantization.	22
2.15	Diagrams of VCO-based ADCs with linearization techniques including (a) the LUT-based calibration [55] and (b) the ‘split-ADC’ technique [59].	23
2.16	Diagram of the background VCO calibration process using a replica circuit.	24
2.17	Diagrams of the (a) digital modulator for linearizing the VCO and (b) inherently linear PFM-based VCO ADCs.	25
2.18	(a) Diagram of a general first-order continuous-time $\Delta\Sigma$ modulator and (b) the implementation with a VCO-based integrator.	26
2.19	(a) Diagram of a general second-order continuous-time $\Delta\Sigma$ modulator and (b) the implementation with VCO-based integrators.	27
2.20	Diagrams of (a) a pipelined SAR ADC, (b) a pipelined SAR-TDC, and (c) a pipelined SAR-VCO ADC.	28
2.21	Illustration of the time-interleaved ADC.	30
2.22	Comparison of energy per conversion between time-interleaved ADCs and single-channel ADCs.	31
2.23	Illustration of error sources in time-interleaved ADCs.	32
2.24	Illustration of time-interleaving spurs caused by offset, gain, and sampling time mismatch.	33
2.25	Illustration of time-interleaving errors in an 8-way time-interleaved ADC.	33
2.26	(a) Comparison of the time-interleaved VCO-based ADC quantization noise with $2\times$, $4\times$, and $8\times$ interleaved and (b) the location of zeros in the NTF of an $8\times$ interleaved VCO-based ADC.	34
2.27	Summary of performance for each ADC architecture including the SAR, pipelined, time-based, flash, $\Sigma\Delta$, hybrid and time-interleaved (TI) [103].	35
2.28	The numbers of ADC each architecture published over the years, including SAR, pipelined, time-based, flash, discrete-time $\Sigma\Delta$ (SDDT) and continuous-time $\Sigma\Delta$ (SDCT) [103].	36
3.1	The analog design flow using the BAG framework.	39
3.2	Diagram of a typical BAG workspace setup.	40
3.3	Examples of schematic and layout generations in BAG.	41
3.4	Illustration of the process-specific transistor primitives and routing grids setup.	43
3.5	Examples of parameterization and process portability in a generation-based design.	44
3.6	The design and optimization using BAG.	44
3.7	The integration flow of ADC prototype chips using BAG.	45
4.1	Diagram of the time-interleaved ADC implemented in the proposed generator.	48
4.2	Diagram of the two-stage pipelined hybrid sub-ADC and the conceptual timing diagram.	49

4.3	(a) Diagram of top-plate sampling and its waveform. (b) Diagram of bottom-plate sampling and its waveform.	50
4.4	Illustrations of (a) the conventional bootstrapped switch [120, 121] and (b) the bootstrap switch implemented in the proposed generator with (c) its equivalent circuit.	53
4.5	Diagrams demonstrating the implementation of (a) the bootstrapped sampler and (b) the sampling switches distributed in the SAR ADC.	54
4.6	Diagrams of (a) the top and bottom-plate sampler with a simple NMOS top-plate switch. (b) the top and bottom-plate sampler and waveform with top-plate bootstrapped. (c) the complete sampling scheme with a middle switch at the top plate.	55
4.7	The extracted simulation results for different sampling schemes, including (a) a simple NMOS top-plate switch, (b) an NMOS top-plate switch with a boosted clock, and (c) the completed sampling scheme in the generator.	56
4.8	Diagram of the unit capacitor and switch, along with its timing.	57
4.9	(a) Illustration of the signal feedthrough and (b) the amplitude of common-mode (CM) and differential-mode (DM) ripples before and after enabling the gate matching option.	58
4.10	(a) Diagram of the bootstrapped circuit, (b) example waveforms of critical nodes in the bootstrap circuit, (c) and example waveforms on the top plate of CDAC.	58
4.11	Extracted simulation results for the final design of the full sampler.	59
4.12	Diagram of a SAR ADC with asynchronous clocking and different sampling options.	61
4.13	Schematics of all supported comparator topologies: (a) strongarm, (b) triple-tail [126], (c) double-tail [127], (d) self-timed double-tail, (e) modified double-tail [128], (f) self-timed modified double-tail.	62
4.14	Diagrams of transistor grouping and layout hierarchy in comparator generators.	64
4.15	Illustrations of the integration steps of the comparator generator: (a) flow chart of the integration steps, (b) floorplan of the complete comparator generator, (c) floorplan of one comparator stage, (d) integration of the comparator, buffers, and the clock generator, and (e) power strapping.	66
4.16	Comparison of speeds for different comparators at various common-mode voltages.	67
4.17	Trade-off between speed and noise for different comparators.	68
4.18	Offset voltages of six comparator topologies using different supply voltages.	69
4.19	(a) Search algorithm of a 3-bit DAC with three steps (b) Search algorithm of a 3-bit DAC with four steps (c) Transfer curve for a 4-bit DAC.	70
4.20	Illustrations of the successive approximation searching and digital outputs of various methods.	72
4.21	(a) The conventional and (b) the monotonic capacitor switching schemes.	73
4.22	(a) The merged capacitor switching and (b) the split capacitor switching schemes.	74
4.23	Comparison of the switching energy in different schemes in a 10-bit SAR ADC.	75
4.24	Diagrams of the CDAC schematic generator.	76
4.25	Diagrams of the CDAC layout generator.	77

4.26	Illustration of CDAC array generation and the input parameters.	78
4.27	Design trade-offs of the overall SAR ADC generator.	79
4.28	Diagram of the asynchronous clock generator and example timing waveforms.	79
4.29	Schematics of logic cells and example timing waveforms.	80
4.30	Floorplan and generation steps of the SAR logic block.	81
4.31	Generated SAR ADCs using different processes.	81
4.32	The conceptual timing diagram of a VCO-based ADC.	82
4.33	The circuit architecture of the VCO-based ADC generator.	84
4.34	Comparison of the phase sampling in single-ended and differential ring oscillators [60].	89
4.35	Diagram of the ring oscillator generator.	90
4.36	The layout details of the ring oscillator stage and the floorplan of the ring oscillator generator.	90
4.37	Schematics of the asynchronous counter generator.	91
4.38	The layout details and floorplan of the counter generator.	92
4.39	Schematics of the sampling flip-flops for a single-stage VCO.	93
4.40	Diagram of the MUX-based thermometer-to-binary phase decoder.	94
4.41	Schematic of intermediate buffers between the RO and the counter.	94
4.42	Classification of popular residue amplifier topologies.	95
4.43	Schematics of (a) a dynamic amplifier, (b) an open-loop OpAmp, (c) an amplifier with floating supply, and (d) a ring amplifier.	96
4.44	Schematic of the complete ring amplifier generator.	97
4.45	Common-mode feedback paths and the stability calibration transistors in the closed-loop ring amplifier.	98
4.46	Diagram of the clock generation circuits for the time-interleaved ADC.	99
4.47	Diagram illustrating the skew calibration process in the ADC generator.	100
4.48	Diagrams of the supported resistor-based DAC generators.	101
4.49	The diagram of the generator architecture.	102
4.50	The diagram of the circuit architecture, floorplan, and available parameters.	103
5.1	Timeline and chip micrographs of the major generator-based prototype chips.	104
5.2	Architecture of the LAYGO time-interleaved SAR ADC generator.	105
5.3	Integration steps of the LAYGO SAR ADC generator.	106
5.4	Analog generators in the Hydra spine massive MIMO chip.	107
5.5	Diagram of the Time-interleaved SAR-VCO ADC prototype chip.	108
5.6	Timing diagram of clocking signals in the critical path.	109
5.7	The completed diagram of the prototype's front end.	110
5.8	The tuning resolution and range of the ring amplifier's biasing circuits along with its DNL and INL.	111
5.9	Simulated results of the ring amplifier linearity across different corners.	112
5.10	(a) Simulated tuning steps and range of the skew correction circuit. (b) Monte Carlo simulation of the LSB delay spread.	112

5.11	Simulation of VCO-based ADC, before (top) and after (bottom) calibration. . . .	113
5.12	Diagram of the data capture memory.	114
5.13	Chip micrograph of the first TI SAR-VCO prototype.	116
5.14	Chip micrograph of the second TI SAR-VCO prototype and sub-ADC layout details.	116
5.15	(a) Layouts of the custom BGA package design and bump maps of (b) the first and (c) the second prototypes.	117
5.16	Diagram of the board designs including a chip board for BGA package and an auxiliary board for the supply regulation and low-speed digital signal communication.	118
5.17	Top view of the 3D model of evaluation boards connecting to an FPGA.	118
5.18	Diagram and photograph of the measurement setup.	119
5.19	Measured output spectrum after calibration, sampling at 4 GS/s for low frequency (top) and Nyquist frequency (bottom) input signals.	120
5.20	Measured spectrum before and after time-interleaved error calibrations.	121
5.21	Measured SFDR and SNDR versus input frequency sampling at 4 GS/s for three ADC samples.	122
5.22	Measured SFDR and SNDR versus input frequency for a single channel (samples at 500 MS/s) and the time-interleaved channels (samples at 4 GS/s).	123
5.23	Measured SNDR and SFDR as a function of the input amplitude for (a) the SAR+VCO ADC and (b) the SAR ADC only.	123
5.24	Measured DNL and INL of ring oscillators in 8 channels.	124
5.25	The figure of merit of this work compared to other similar designs.	125
5.26	Power breakdown.	125
5.27	Performance summary and comparison with state-of-the-art ADC.	126
A.1	Illustration of the LAYGO layout generation flow.	147
A.2	Illustrations of (a) DigitalBase and (b) AnalogBase in the framework.	148
A.3	Illustration of the MOSBase in the BAG3 framework.	149
B.1	Example inverter generator in BAG3.	151
B.2	Generator code of the single-ended transistor group used in the comparator generators.	152
B.3	Generator code for the preamplifier.	153
B.4	Self-timed double-tail comparator generator.	154
B.5	Generated instance of a simple differential comparator in the cds-ff-mpt process.	155
B.6	Generated instance of a comparator with multiple groups in the cds-ff-mpt process.	155

Acknowledgments

I am deeply grateful for the extraordinary opportunity to conduct my research and academic exploration at the University of California, Berkeley, over the past few years. This journey has been enriched by the invaluable interactions with many talented individuals, particularly during the challenging times brought on by the pandemic. I want to take this opportunity to express my sincere gratitude to each and every one of them.

First and foremost, I would like to express my gratitude to my advisor, Professor Bora Nikolić. His expertise, understanding, and encouragement have been guiding throughout my PhD journey. He has been very supportive in all aspects of my graduate school studies, from steering me through my studies to providing valuable insights on research matters. He is also very understanding about all the difficulties that student might face and is willing to offer his support. I sincerely appreciate his unwavering support and assistance, which have been crucial in shaping this thesis and my academic journey.

Also, I would like to thank Professors Vladimir Stojanović, Ali Niknejad, and Martin White for being on my qualification exam and thesis committee. Their insights, valuable comments, encouragement, and consistent support were enormously beneficial. I also want to thank Professor Elad Alon for his inspiring discussions. I want to thank Chris Mangelsdorf for his valuable suggestions and meticulous help throughout the entire process, from the stage of my tapeout to the measurement. I am also truly thankful to my undergrad research advisor Professor Wengao Lu who introduced me to IC design.

My sincere thanks also go to all BWRC staff. In particular, I thank Candy Corpus for her willingness to assist and care for us, and Brian Richard, Jeffrey Anderson, James Dunn, and Mikaela Cavizo-Briggs for their tremendous support and providing the wonderful research environment. My gratitude goes to Anita Flynn for her tireless efforts in aiding my research.

I would also like to thank my fellow BWRC colleagues. First of all, I enjoyed being part of Bora's group with peers such as Yue Dai, Sean Huang, Harrison Liew, Daniel Grubb, Vighnesh Iyer, Felicia Guo, Nayiri Krysztofowicz, Jingyi Xu, Seah Kim, Dima Nikiforov, Dan Fritchman, Ken Ho, Amy Whitcombe, Sijun Du, Keertana Settaluri, and many others. This has enriched the experience and made the stay very memorable. Among them, I want to especially express my gratitude to Woorham Bae for his significant assistance during the early years of my study and his generous sharing of knowledge. Outside of the group, I would like to extend my gratitude to Zhongkai Wang for his invaluable technical discussions and enduring friendship. Besides, I would like to express my gratitude to Krishna Settaluri, Sidney Buchbinder, EKouros Hakhmaneshi, Eric Chang, and Nathan Narevsky for their help in my research and internship at Bluecheetah, particularly in relation to BAG-related subjects. I would also like to express my gratitude to the individuals I collaborated with during the crucial tapeouts: Kwanso Park, Bob Zhou, Paul Kwon, Ayan Biswas, Zhenghan Lin, Kunmo Kim, and Yi-Hsuan Shi. It is an honor for me to have the opportunity to work with them. I also thank Ruocheng Wang, Bozhi Yin, Luya Zhang, Yikuan Chen, Biqi Zhao, Meng Wei, and Benyuanyi Liu for their valuable feedback on my research and for their friendship.

My internship experience with Intel, Keysight, and Bluecheetah greatly enriched my knowledge. I thank John Keane for generously sharing his extensive experience in ADC design and Robert Neff for his guidance during my internship, as well as for the insightful discussions and assistance afterwards. I want to thank Charles Wu not only for his help during my internship but also for sharing his insights about research, career, and life. I would also like to express my gratitude to Hui Fu for offering me the opportunities to delve into diverse topics and for generously supporting my research.

Finally, I would like to express my deepest gratitude to my family for their boundless love and enduring support during the most challenging years of studying abroad and enduring the isolation brought on by the pandemic. I am grateful to my friends for their companionship. Most importantly, I want to thank my girlfriend Qingqing. Her love, patience, and support have made all my accomplishments possible, and I would not have been able to reach this far without her.

Chapter 1

Introduction

1.1 Motivation

Analog-to-digital converters (ADCs) convert inherently analog, real-world continuous signals into discrete, digital equivalents that can be stored and processed by digital systems. The ADCs essentially serve as the interface between the analog and digital domains, making them an indispensable component across numerous applications. The rapid advancement of digital integrated circuits has led to increasingly complex signal processing systems, which necessitate enhanced speed and precision in data processing across a broad range of applications. Therefore, the research in ADCs has been driven by the constant demand for faster and more accurate data processing in various applications. For instance, in the realm of advanced radar systems, high-speed Nyquist ADCs are implemented to enable precise target detection, tracking, and categorization. Another application is found in high-speed data acquisition systems, prevalent in industrial and scientific measurement contexts. These systems require high-speed ADCs to capture and digitize rapidly varying analog signals with high precision. Instruments such as oscilloscopes, spectrum analyzers, and network analyzers demand the use of high-speed ADCs for accurate digitization and analysis. High-speed ADCs enable these devices to provide precise measurements, diagnose complex electronic systems, and characterize the performance of devices under test. Moreover, as more devices become interconnected through the Internet of Things, the need for high-speed ADCs to efficiently process and transmit data will only continue to rise.

In the domain of communication systems, the demand for superior bandwidth and higher accuracy in both wireline and wireless communication systems has increased rapidly. These escalating demands have led to a surge in research and development efforts in the design of the ADC, which serve an indispensable role in these systems. In direct RF communication receiver systems, incorporating wideband ADCs contributes to the reduction of overall system complexity and enhances overall efficiency by capturing the full band of the signal, thereby simplifying the signal chain. The recent advancements in ADC design, with resolutions exceeding 10 bits and sampling rates extending into the gigahertz domain, have expedited

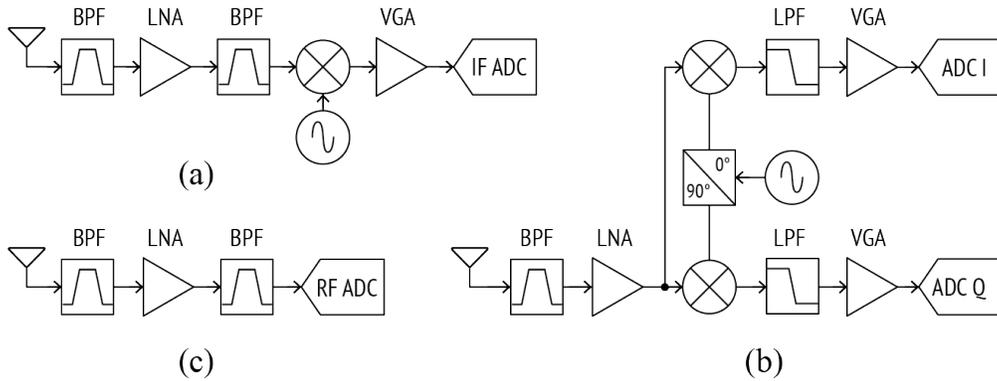


Figure 1.1: Diagrams of common receiver architectures: (a) superheterodyne, (b) direct conversion, and (c) direct RF sampling.

the practical application of software-defined radios [1, 2]. As depicted in Figure 1.1, a receiver architecture employs direct RF sampling, where the incoming signal is sent straight to the ADC for conversion [3, 4]. Contrasting this with the traditional superheterodyne architecture, a wideband ADC replaces a significant portion of the signal chain, consequently reducing system complexity, power consumption, and overall cost. In electrical and optical link designs, the integration of an ADC-based receiver front end (Figure 1.2) yields increased flexibility and adaptability with less complexity. These features enable enhanced back-end digital signal processing for equalization and symbol detection, offering a distinct advantage over mixed-signal receiver counterparts. As a result, it facilitates more spectrally efficient modulation schemes, as opposed to the binary non-return-to-zero (NRZ) configurations [5, 6, 7, 8].

The increasing demand for enhanced speed, precision, and efficiency in data processing across a broad range of industries and applications is posing significant challenges in the field of high-performance ADC design. The design of such high-performance ADCs requires a careful trade-off between speed, accuracy, and power consumption. As the speed of an

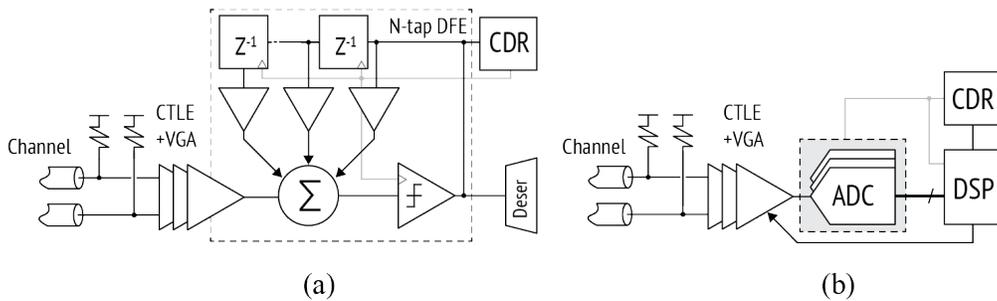


Figure 1.2: High-speed serial link transceivers with (a) a mixed-signal receiver architecture and (b) an ADC DSP-based receiver architecture.

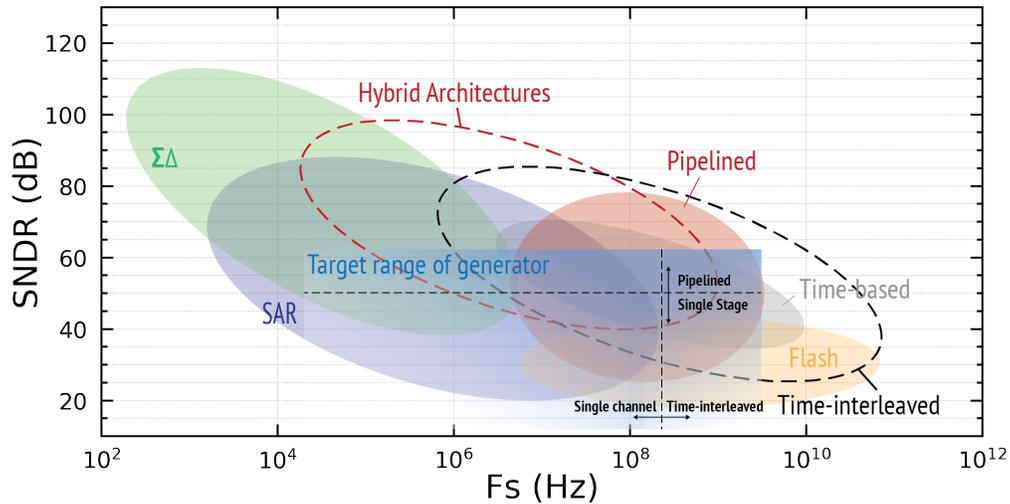


Figure 1.3: Popular ADC architectures and the target coverage range of the proposed generator.

ADC increases, maintaining high resolution can become increasingly difficult due to the inherent design trade-off between the sampling rate and the number of bits in most ADC architectures. Moreover, with increasing sampling rates, the circuit's power demand generally increases, making energy efficiency a critical design challenge. At the circuit level, implementing the optimal ADC core design involves a comprehensive understanding of the fundamental limitations inherent in each architecture, as well as the sources of various non-idealities. Hybrid converters have become more popular, combining the strengths of different architectures to achieve the best attainable performance beyond the capabilities of existing base topologies.

To improve the sample rate beyond that of a single converter, time-interleaving (TI) has been widely utilized in single-channel ADC cores. While the concept of interleaving is straightforward, the operation itself necessitates dedicated auxiliary circuits for clock distribution, buffers, and carefully designed sampling networks. The optimal integration of the ADC core with these peripheral circuits for optimal system trade-offs poses a significant challenge at the system level. Figure 1.3 illustrates the operating range of different base ADC architectures, along with hybrid and time-interleaved ADCs that exhibit superior performance in terms of resolution and speed.

Technology scaling has a significant influence on the design of ADCs as well. It is necessary to use ADCs with architectures that have scaling potential in order to fully leverage the advancements in digital processing power brought by CMOS scaling. Scaling provides several inherent advantages for analog circuits in general. As the size of transistors shrinks, it becomes possible to operate ADCs at higher speeds due to shorter transistor switching times. This scaling can also result in reduced power consumption per operation, which can

help mitigate one of the major challenges in high-speed ADC design.

However, technology scaling brings challenges to ADC design as well. For instance, the supply voltage scales at a more rapid pace than the threshold voltage, resulting in a reduced voltage headroom [9] (Figure 1.4 (a)). This restricts the attainable performance of the analog circuit, specifically limiting the available analog swing in ADC design and consequently narrowing the dynamic range. Additionally, the performance variability for minimum-sized transistors in a given process node increases as feature sizes shrink, and non-ideal effects pose increasingly significant design challenges. Therefore, as technology continues to advance, it becomes imperative to innovate design techniques and architectures that can address these challenges and facilitate the continuous progression of high-speed ADCs. One of the primary focuses of this research is hence the exploration of process-scalable converter architectures, implying a shift from architectures reliant on high-gain static amplifiers to architectures that predominantly require switches, capacitors, and digital blocks.

Alongside circuit-level challenges in achieving high performance at different process nodes, as an analog and mixed-signal circuit, the design of an ADC becomes increasingly complex, time-consuming, and error-prone in advanced processes. As device sizes shrank, the number of design rule checks (DRCs) grew exponentially, making it difficult to quickly prototype designs in modern process technologies (Figure 1.4 (b)). Also, the elevated sensitivity to parasitics, coupled with more complex layouts, necessitates more design iterations. Meanwhile, reliability concerns such as electromigration and dynamic voltage drop, further prolong the design process. Therefore, as technology continues to scale, it will be critical to develop innovative design techniques and architectures that can effectively address the performance challenges and enable the continued advancement of high-speed ADCs. In the meantime, agile design and fast prototyping are important for achieving the optimal choice with a given

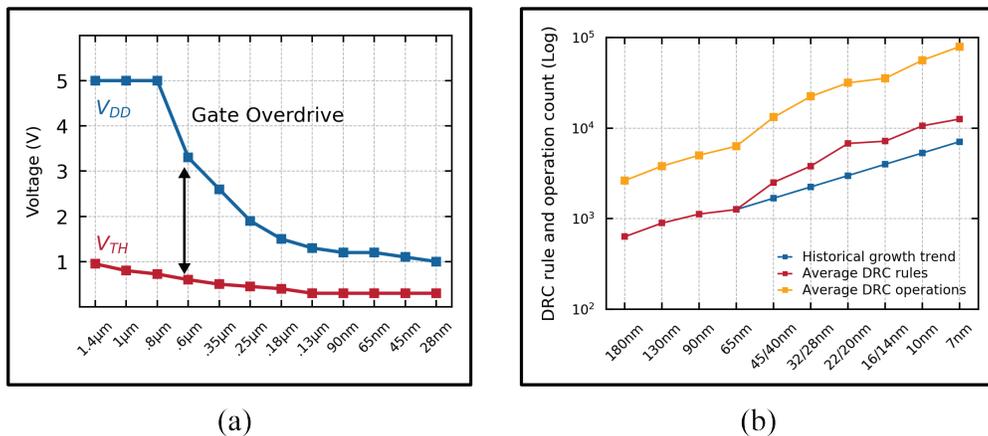


Figure 1.4: (a) Supply and threshold voltage vs. process nodes [9]. (b) The average growth in the number of design rules and the number of DRC operations over node progressions [10].

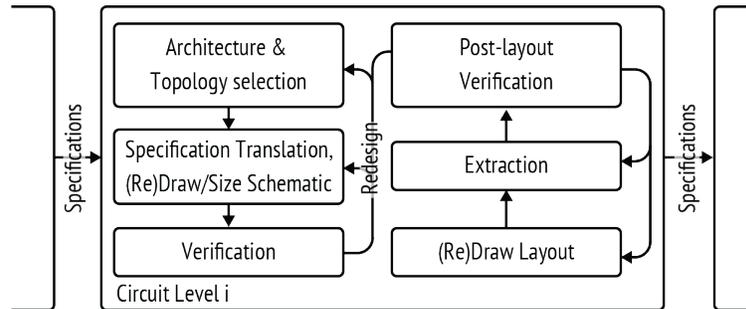


Figure 1.5: Diagram of a typical analog circuit design flow.

technology choice as an extra degree of freedom for potential exploration.

To address the concerns in the conventional analog circuit design flow (Figure 1.5) and improve the reusability of the design, the Berkeley Analog Generator (BAG) framework is introduced [11] to enable the design of analog circuit generators. The BAG framework provides various functions for drawing and modifying the layout and schematic using the BAG grid system and primitives. It includes APIs that connect the Python-based framework with various design tools, facilitating the design process. The BAG framework can automate circuit generation and encapsulate design concepts in the form of an executable generator. It can also produce DRC- and layout-versus-schematic (LVS)-clean schematics and layouts, along with verification testbenches. By using the generator-based design methodology, it is possible to automate the essential stages of the traditional analog design process. This automation includes schematic and layout generation, extraction, simulation, and resizing. These procedures can be incorporated into automatic design iteration loops. Designers can easily implement circuits and systems in various technologies merely by updating process-specific primitives and generator parameters. The agile design iterations achievable by updating the generator significantly shorten the design and validation time.

1.2 Research Goals

This research aims to create efficient and high-performance ADC designs while simplifying the design process, and reducing human effort by using a generator-based design methodology. The ADC generator framework implements various circuit generators and auxiliary scripts using the BAG framework. The benefits of ADC design using this methodology include:

- Automatic circuit generation: Automatic layout and schematic generation simplify the manual design process by capturing design ideas and methodologies in the generator scripts, reducing human effort and minimizing the possibility of errors. The agile design approach can be adopted during generator development to incrementally include more options and features in the circuit generator.

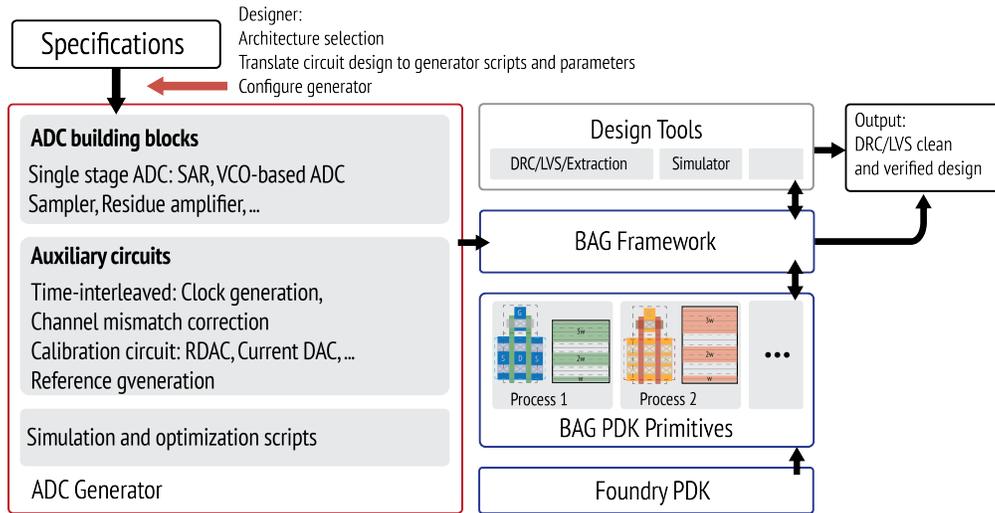


Figure 1.6: Diagram of the generator-based design flow using BAG.

- **Configurability and design reusability:** Different applications may require different ADC specifications, such as resolution, speed, and power consumption. The generator-based design allows for high reconfigurability, enabling the quick generation of customized designs with different targets. Another goal is to make the generators reusable across multiple technologies. Reusability enhances productivity and streamlines the design process. Automatic ADC design enables seamless scaling of designs across various process nodes, allowing designers to rapidly adapt to changing technology requirements.
- **Architectures exploration:** The ADC generator enables designers to explore and evaluate a wide range of ADC architectures, including traditional and hybrid alternatives. This process helps identify the most suitable architecture for specific applications, taking into account factors such as speed, resolution, power consumption, and technology constraints. This comprehensive exploration process leads to more informed decisions and the selection of optimal parameters and architectures that best suit the needs of the target application.

This research aims to develop a comprehensive ADC generator framework capable of generating a wide range of ADC instances that meet various specifications. Different base ADC architectures can be combined into a hybrid form to achieve higher resolution. Also, the supporting circuit for the time-interleaving architecture help enhances the sampling rate of ADC instances. The target resolution is up to 10 effective number of bits (ENOB) and up to 4 GHz sampling rate. The target range of generation is shown in Figure 1.3. To demonstrate the feasibility of the generator-based design methodology for high-performance ADC design, the prototypes are fabricated and measured.

1.3 Thesis Organization

Overall, this dissertation discusses and evaluates the ADC architectures that are suitable for high-speed applications. By utilizing a generator-based design methodology, this work ensures high reconfigurability and reusable designs across different process nodes to meet varying specifications.

Chapter 2 presents an overview of popular traditional and time-based ADC architectures. It explores their respective advantages, drawbacks, and scaling potential. Based on this analysis, potential hybrid architectures are examined, with successive approximation register (SAR) and voltage-controlled oscillator (VCO)-based architectures selected for further investigation in this work.

In Chapter 3, the focus shifts to existing generator-based design methods and analog design with the BAG framework, with a review and comparison of various approaches. Due to the desired reconfigurability and performance requirements of the target design, the scripted generation method using the BAG framework has been chosen to implement the proposed ADC generator. Examples of the BAG setup, circuit generation, and design flows are presented.

In Chapter 4, the generators of critical building blocks are presented, including the sampler, SAR ADC, VCO-based ADC and the residue amplifier. For each block, available options are introduced, taking into account various use cases. Chapter 4 provides a detailed examination of different topologies, design options and trade-offs. Additionally, the floorplan for each generator is presented to offer a clear understanding of implementation and layout.

Chapter 5 presents the prototypes implemented using the generator-based methodology. The first two prototypes employ an older version of the BAG framework to create a time-interleaved SAR ADC generator. As these initial prototypes are less relevant to the proposed ADC generator framework, the primary focus is placed on the latter two prototypes that utilize the ADC generator to implement a 4-way and 8-way time-interleaved SAR-VCO hybrid architecture. The measurement setup and results are also presented in this chapter.

Finally, Chapter 6 summarizes the dissertation by drawing several conclusions based on the results achieved and discusses potential future directions for further exploration in the realm of automatic ADC circuit design.

Chapter 2

High-Speed ADC Design

2.1 Introduction

The demanding performance requirements for high-performance ADC design in communication applications encourage the development of ADCs that use hybrid architectures. These architectures benefit from the integration of several sub-ADCs that utilize sub-ranging and pipelining architecture to achieve high-speed, high-resolution, and power-efficient operation. To explore the ADC architecture using the generator-based design methodology, this chapter describes key ADC architectures in both the voltage domain and the time domain. Scaling-friendly techniques are preferred to create a process-portable ADC generator. At the same time, architectures that are digital-centric and have low-voltage tolerance and high reconfigurability in resolution are more suitable for the generator-based methodology. Also, the time-interleaving architecture is discussed, as an effective method for increasing the speed of a standalone converter.

The function of an ADC is to generate an N -bit digital output D that approximates the analog signal as $V_{ADC} = D/2^N \cdot V_{ref}$, where V_{ref} represents the reference voltage. Depending on the approach used to obtain the final value, various categories of ADCs implement this conversion using different algorithms and in different domains. In the voltage domain, architectures such as flash, pipeline, and successive approximation register (SAR) sample the input signal at Nyquist frequency ($f_s = 2 \times f_{BW}$, where f_s and f_{BW} represent the sampling frequency and the maximum signal frequency, respectively). Other oversampling ADC architectures ($f_s \gg 2 \times f_{BW}$) are less relevant to the target performance of the ADC generator and are therefore not discussed here. On the other hand, advances in time-domain ADCs have shown competitive performance metrics for low-to-medium resolution using the delay line-based time-to-digital converter (TDC) and VCO-based architecture. The improvement in time resolution with CMOS scaling directly enhances the performance of ADCs. Additionally, converting the information to the time domain overcomes the limitations imposed by the available supply voltage. As a scaling-friendly trend in ADC design, time-based ADCs are investigated and integrated into the generator.

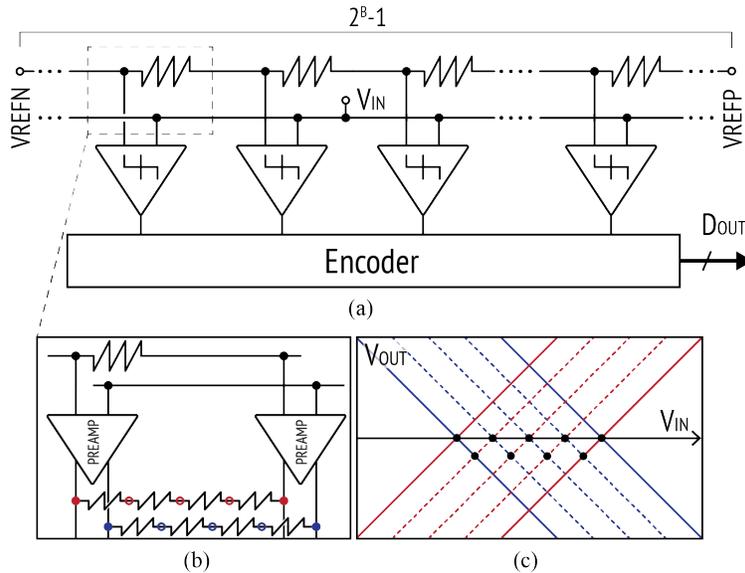


Figure 2.1: Diagrams of (a) a flash ADC, (b) the resistive interpolation method, and (c) interpolation transfer curves.

2.2 Traditional ADC Architecture for High-Speed Operation

This section briefly reviews the three most commonly used architectures in high-speed Nyquist-rate ADC designs: flash ADCs, successive approximation register (SAR) ADCs, and pipelined ADCs. An estimation of the power, speed, resolution, and complexity of these conventional topologies are presented to determine the most suitable topology to implement using the generator-based design method. The scaling potential is discussed as well.

2.2.1 Flash ADC

The diagram of a Flash ADC is shown in Figure 2.1. A typical flash ADC consists of an array of comparators that are calibrated to different reference voltages and are all clocked simultaneously. The reference voltages are often generated from a resistor ladder that includes 2^N resistors with identical values. The analog input voltage is compared to various reference voltages, and the outputs of comparators are collected. These outputs are then converted from thermometer code to binary code using an encoder [12]. Since all the comparators work simultaneously, the flash ADCs only need one clock period to complete the conversion, making them capable of achieving the fastest speed among all ADC architectures. The speed of the flash ADC is limited by the sampling process, as well as the delay of the comparator and the encoder.

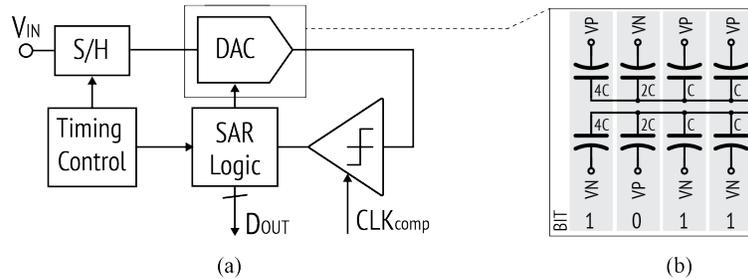


Figure 2.2: Diagrams of (a) a typical SAR ADC and (b) its capacitive DAC.

Despite its high-speed operation, the flash ADC has various limitations. The primary limitation is that an N -bit flash ADC requires $2^N - 1$ comparators, which significantly increases the area and power consumption as the target resolution increases. Not only is the number of comparators problematic, but the offset of the comparators also contributes to nonlinearity errors. Additionally, the input capacitance of the comparator is nonlinear and depends on the level of the input signal. This non-linear capacitor creates signal-dependent sampling time and distortion that is dependent on both frequency and amplitude. The comparator takes time to resolve the result, and the practical implementation must also address related issues such as metastability and sparkle codes. Interpolating [13, 14] and folding [15] techniques are proposed to alleviate some of the issues mentioned above.

The interpolation technique shown at the bottom of Figure 2.1 helps to reduce the input capacitance, area, and power dissipation. The linearity of the ADC is also improved by the averaging effect when interpolating the least significant bit (LSB) of the reference resistor. The folding technique maps the full-scale range of the ADC to a smaller range. The most significant bit (MSB) of the ADC determines which fold the input is in, while the LSB ADC determines the position within the fold. This approach significantly reduces the number of comparators and minimizes non-idealities related to input capacitance. Although combining interpolation and folding leads to improved area efficiency, the practical issue with the folding circuit, as well as the power efficiency and bandwidth of the preamplifier, limit its application in modern communication systems.

2.2.2 Successive Approximation Register ADCs

The SAR ADC utilizes a binary search algorithm to compare the generated analog signal with the input voltage successively. A typical implementation of a SAR ADC, as shown in Figure 2.2, consists of a digital-to-analog converter (DAC), digital logic, a comparator, and a clock generator. A sample-and-hold (S/H) block samples the input signal. The binary-weighted reference is generated using a DAC, the most common implementation being a capacitive DAC (CDAC). The DAC adds or subtracts a fraction of the reference voltage from the sampled voltage based on the result of the comparator. After evaluating from the MSB to the LSB, all N bits of digital outputs stored in the SAR ADC are collected. The comparator

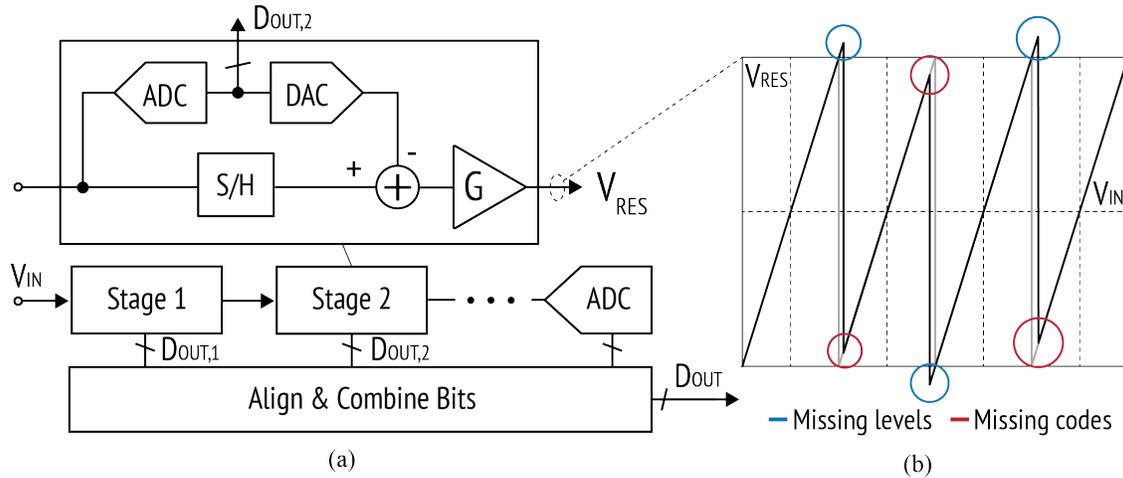


Figure 2.3: (a) Diagram of a typical pipelined ADC and (b) the residue voltage transfer curve.

clock can be provided externally. A synchronous clock is used to progress from MSB to LSB. Alternatively, asynchronous SAR ADC triggers the comparison internally, similar to dominoes [16]. This approach reduces the overhead of clock generation and provides a soft margin for worst-case comparison, thereby reducing the metastability rate. While the implementation of a single comparator is commonly used due to its simplicity in both circuit implementation and calibration, it is possible to use multiple comparators to replace the single comparator in Figure 2.2. SAR ADCs use two alternate comparators [17] and loop-unrolled SAR ADCs [18] introduce additional comparators to enhance conversion speed. A SAR ADC typically only requires clocked comparators with digital characteristics and digital logic gates. Those MOS switches and latches benefit greatly from aggressive technology scaling, leading to the success of modern SAR ADCs. Therefore, the digitally-intensive and highly efficient operation makes the SAR ADC a strong candidate in modern systems.

2.2.3 Pipelined ADC

Pipelined ADCs [19, 20, 21, 22] are popular for their unique combination of speed and resolution. Figure 2.3 illustrates the operation of the pipelined ADC through a series of stages. At each stage in the pipeline, the ADC handles a portion of the total conversion and resolves the MSB of the current state. The MSB result is subtracted from the input to generate a residue, which is then amplified and passed to the next stage. This process repeats until all bits are converted. In an N -bit, M -stage pipelined ADC, each stage resolves N_i bits, where $i = 0, 1, \dots, M - 1$. The total of $N_0 + N_1 + \dots + N_{M-1}$ bits are combined by the digital logic. The implementation of each stage consists of an S/H block, an N_i -bit flash ADC, an N_i -bit DAC, and a residue amplifier. The voltage that remains after each stage's conversion is then amplified by the residue amplifier. An amplifier with a gain equal to 2^{N_i} only works in

ideal conditions. Any errors in the sub-ADC decision levels will overload the backend stages and degrade the ADC transfer function. Figure 2.3 shows that the errors are translated into missing codes and levels. Therefore, a gain less than 2^{N_i} is typically employed to create inter-stage redundancy and allow for error tolerance. The pipelined ADC can operate at a higher speed than the SAR ADCs at the cost of more hardware, which consumes more area and power. While pipelined ADCs offer an attractive balance of speed and resolution, as the technology scales down, the reduced supply voltage and the need for precise analog components pose challenges to their design.

2.2.4 Summary

Among the three conventional architectures, SAR ADCs have the lowest complexity at the cost of lagging conversion speed behind flash and pipelined ADCs. The flash ADCs convert the signal within one clock cycle, but the complexity and power increase exponentially, making the design inferior to SAR and pipelined ADCs when the resolution is larger than 6 ENOB. The pipelined ADC can easily achieve a higher resolution than this. However, the implementation that includes an amplifier-like circuit does not benefit from aggressively scaled technology. The SAR ADC is much more compact, which makes it easier to design a time-interleaved system. This will be discussed later in this chapter. Overall, SAR ADCs can provide highly efficient, digital-intensive operations suitable for moderate conversion speeds and resolution levels [23]. The leading-edge performance covers a sampling rate range from tens of kilohertz to tens of gigahertz. For moderate frequency, SAR or SAR-assisted ADCs, such as [24] and [25] can provide medium resolution at low power levels. At the ultra-high-speed region, a 90 GS/s design [26] has been demonstrated to be suitable for optical and electrical data link applications. Considering the generator-based design methodology, the digital-like circuit and the highly regular, repetitive layout pattern in the SAR ADC greatly simplify the generator implementation. Moreover, the adjustment for resolution can be easily achieved by incorporating digital logic and expanding the DAC resolution. This makes it an ideal circuit for this research and provides a versatile building block for both standalone use cases and as one of the key components in higher speed and resolution designs.

2.3 Time-Domain Data Conversion and ADC Architectures

The scaling of CMOS technology is primarily driven by the optimization of digital circuits, which aims to improve switching speed, increase transistor density, and reduce the supply voltage. As a result, technology scaling creates challenges for designing critical components in the voltage-domain ADC. Moreover, the signal-to-noise ratio (SNR) of the ADC is degraded by the reduced signal swing when the supply voltage is decreased. In the meantime, the resolution of the time-domain ADC is not affected by the available supply voltage. The

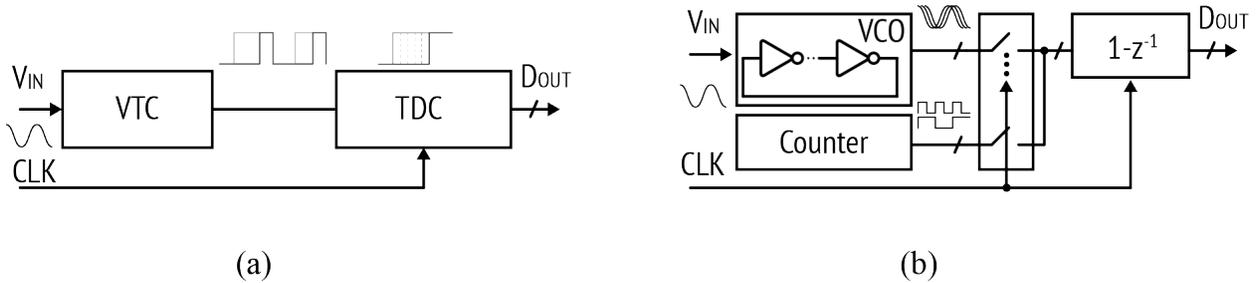


Figure 2.4: Diagrams of (a) a time-biased ADC consisting of a VTC and TDC and (b) a VCO-based ADC.

resolution can also be improved by the increased time resolution as the process scales. Moreover, the transition from voltage-domain to time-domain significantly simplifies the circuit's complexity due to the digital-centric nature of time-based ADCs. In a time-based ADC that utilizes a time-to-digital converter (TDC) and a voltage-to-time converter (VTC), the unknown input voltage is initially converted into the time difference between the occurrence of two digital signals. Then, the time signal is converted back to a voltage by a delay line that quantizes the time difference and converts it into a digital code. Another category of time-based data conversion involves the use of VCOs. The voltage-domain signal is converted to various oscillation frequencies and quantized by measuring phase. Figure 2.4 shows high-level diagrams of these two methods. Different techniques for time-domain data conversion are reviewed and compared in this section. In contrast to the well-known voltage-domain conversion, additional details are examined in this chapter to provide the context for discussion and illustrate the trade-offs in each time-based ADC implementation.

2.3.1 Voltage-to-Time Converters

In the first category of time-based ADCs described above, the front end consists of a VTC that receives an analog voltage to be converted and a clock signal. Most of the VTCs adopt the current-starved inverter topology [27, 28]. An example of a differential implementation is shown in Figure 2.5. It consists of two current-starved inverters and two comparators. After each reset period, the current-starved inverter takes an input voltage and discharges the output nodes proportionally. Once the output node exceeds the threshold voltage of the comparator, it generates a digital pulse. The timing diagram shown in Figure 2.5 (b) demonstrates that a difference in input voltage is translated to a relative delay Δt in the time domain. The TDC later quantizes this information into the digital domain.

The discharge rate controls the gain of the VTC. When a specific sensitivity of VTC is needed, a time-domain amplifier [29] can be cascaded to increase the gain and achieve a higher resolution. The VTC design tradeoff exists between its linearity and dynamic range. Although pseudo-differential topology can cancel out the second-order nonlinearity, higher-order harmonics still exist. Different techniques have been proposed to solve this

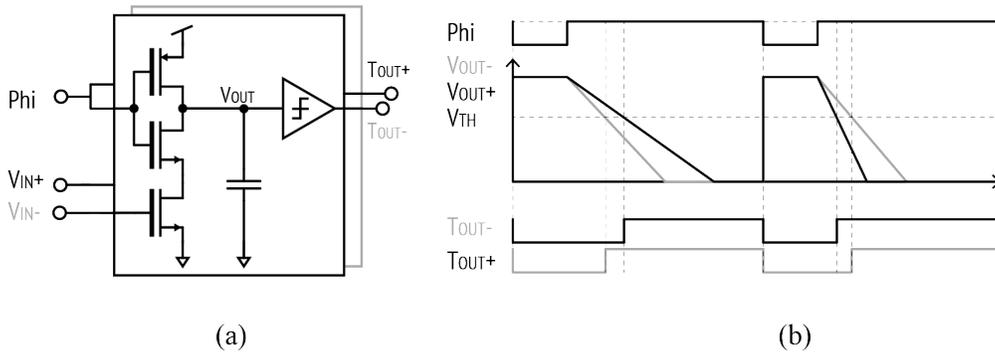


Figure 2.5: (a) Schematic of a typical differential VTC using current-starved inverters and (b) its timing diagram [35].

problem, including the use of adjustable biasing for optimal linearity [30], source degeneration linearization [31] or complementary implementation in [32, 33]. The dynamic range of the VTC can be increased by using folding techniques, which are similar to voltage folding in the design of the flash ADC [34, 35].

2.3.2 Time-to-Digital Converters

The back-end of time-based ADCs typically consists of a TDC that measures the time-domain information of a discrete amplitude signal. The time-domain information can be as follows: 1. The timing difference between the edge of a **START** and a **STOP** signal. 2. The location of signal edges relative to a reference signal. 3. A pulse width. The performance of a TDC design can be characterized by the minimum resolution (i.e., the smallest time step) and the dynamic range, which is defined as the lower and upper bounds of the time intervals within a given conversion time. Similar to voltage-domain ADCs, the nonlinearity in the transfer function of a TDC is the deviation of the time-to-digital transfer characteristic. The most straightforward implementation of a TDC is a digital counter that counts the number of signal edges during a given time. The time resolution T_{1sb} is determined by the period of the counter, and its upper bound is determined by the number of bits in the counter. The counter implementation can provide a resolution of tens of picoseconds in advanced process nodes. Any further increase in resolution results in a challenging design and excessive power consumption.

2.3.2.1 Delay Line

An alternative popular approach to quantizing the time-domain signal with higher resolution is to use delay-line-based techniques [36]. The diagram is shown in Figure 2.6 (a). Connecting the signal to the delay line creates multiple delayed versions, each of them having T_D compared to the previous one. The delayed signals are sampled by a set of parallel flip-flops

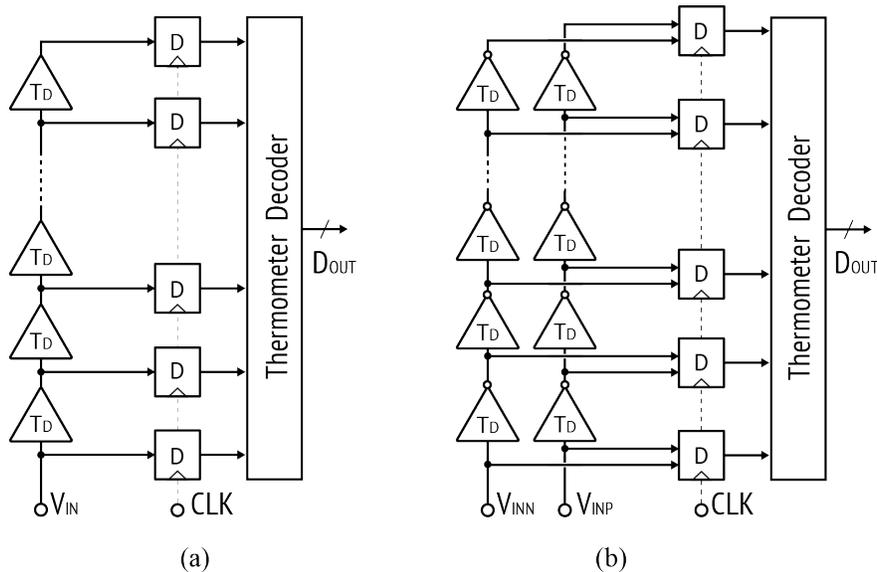


Figure 2.6: Diagrams of (a) a basic delay-line-based TDC and (b) a differential delay-line-based TDC that doubles the resolution.

using a reference clock. Similar to the flash ADC, the output code will be a thermometer code that indicates the transition occurring at the signal edge, assuming the signal is within the dynamic range. Therefore, the relative delay between the signal and clock edges is quantized with a resolution

$$T_{lsb} = T_d, \quad (2.1)$$

where T_d can be as small as two inverter delays ($2 \times t_{inv}$). This topology is simple and power efficient since it only consumes power during switching. Therefore, the delay-line-based TDC is widely used [28, 37, 38, 39]. One simple modification to the delay line can reduce the T_{lsb} by half [40]. The diagram is shown in Figure 2.6 (a). The delay cell is changed to an inverter and half of the LSB. Due to the change in signal polarity, different sampling flip-flops are used to capture the intermediate nodes. Specifically, they are used alternately in every other stage. Compared to the single-ended delay line, a differential implementation also matches the signal's rise and fall times and reduces mismatches.

2.3.2.2 Cyclic Delay Line

The simple delay line discussed above requires a sufficient number of stages to cover the entire period of the reference clock. Increasing the number of stages comes at the cost of more sampling flip-flops, which significantly impacts the power and complexity of the circuit. The number of stages can be significantly reduced by bending the delay line into a ring, allowing the signal to propagate cyclically. An example is shown in Figure 2.7 (a) [41, 42]. Similar to the folding techniques used in flash ADCs, a cyclic implementation not only

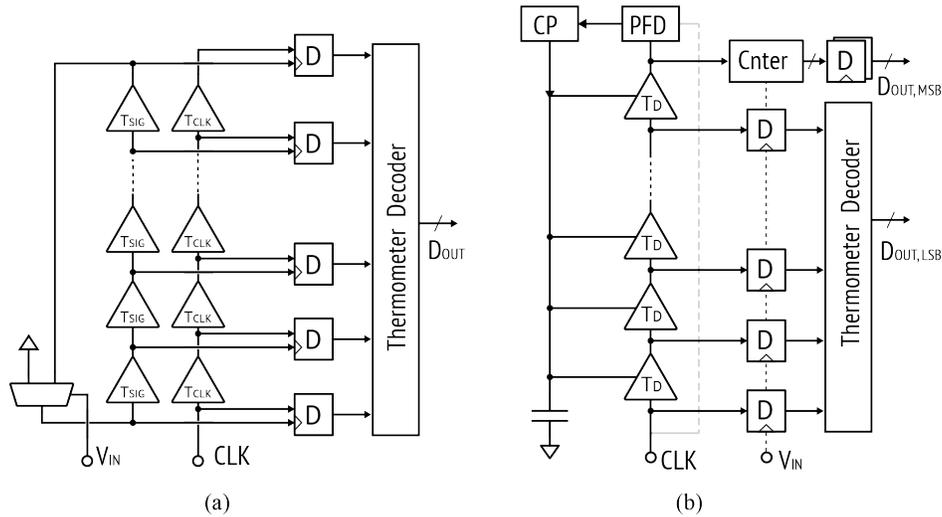


Figure 2.7: Diagrams of (a) a recirculating TDC and (b) a TDC based on a delay-locked loop.

reduces the number of critical elements but also improves the integral nonlinearity (INL) of the converter. An error is inevitably caused by the process-voltage-temperature (PVT) variation due to the delay mismatch between the multiplexer (MUX) gate and the inverter.

The output unit delay of the mentioned TDC topologies can be either a fraction of the reference clock period $(1/N) \cdot T_{CLK}$ or an absolute time step size T_{lsb} . Although the TDC inherently averages the error from the individual cells [28], additional calibration might be necessary for the TDC to correct the static error. Digital calibration techniques are typically implemented to achieve precise absolute time resolution by allowing for adjustability in each delay cell. When comparing the signal delay with a reference clock, a delay-locked loop or phase-locked loop is used to align the total delay of the delay line with the reference clock [42, 43]. An example is shown in Figure 2.7 (b).

2.3.2.3 Techniques to Improve the Resolution of Delay Lines

The delay-line-based TDCs reviewed so far limit the resolution to the minimum gate delay in a given process. To improve the resolution of the TDC, a sub-gate-delay resolution is necessary. The Vernier TDC and time interpolation techniques are used to enhance the time resolution beyond the minimum delay. The Vernier TDC measures the delay between the start and stop signals, similar to slide gauges [44]. The diagram of the Vernier delay line is shown in Figure 2.8 (a). The signal and reference are delayed by T_{SIG} and T_{CLK} , respectively. The resolution is then defined by the difference in delay:

$$T_{lsb} = T_{SIG} - T_{CLK}, \quad (2.2)$$

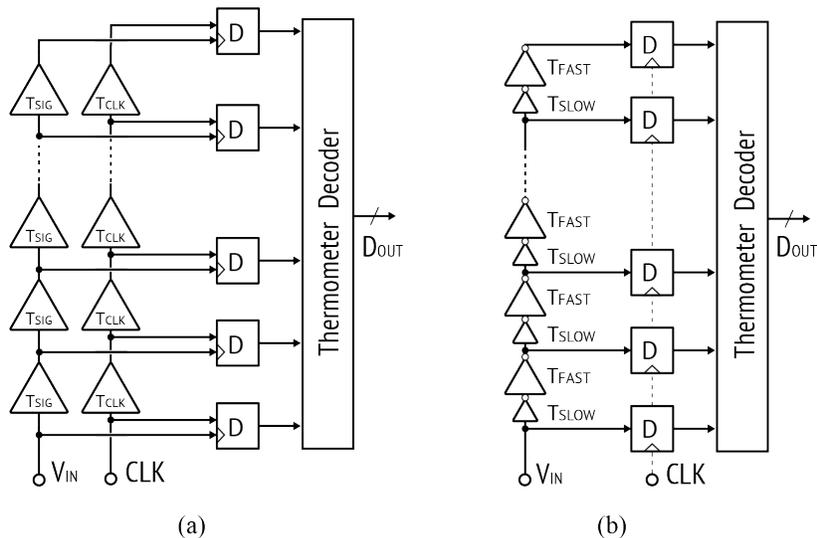


Figure 2.8: Diagrams of (a) a basic Vernier TDC and (b) a pulse-shrinking TDC.

where T_{SIG} and T_{CLK} are the delays in signal and clock paths. To ensure monotonicity, the minimum delay difference is constrained by the delay variations plus the input referred offset of the sampling flip-flops. A similar concept can be applied to the pulse-shrinking TDC shown in Figure 2.8 (b). The delay cell uses an asymmetrical inverter. After two inverters, the pulse width changes by

$$T_{\text{lsb}} = (T_{\text{FAST,rise}} - T_{\text{FAST,fall}}) - (T_{\text{SLOW,rise}} - T_{\text{SLOW,fall}}), \quad (2.3)$$

where T_{FAST} and T_{SLOW} are the delay of slow and fast inverters, respectively. The pulse width shrinks gradually as it passes through more stages. Then the input pulse width can be measured by detecting the stage where the pulse disappears.

Another possibility of improving the resolution to higher than the sub-gate-delay is by using [38], similar to the flash ADCs. Although guaranteed to be monotonic, passive interpolation has several disadvantages. The resistors, combined with the input capacitance of the flip-flops, form an RC circuit chain, which results in a nonuniform step. Also, the mismatch of passive elements and sampling flip-flops limits the minimum interpolation steps. Another approach is active interpolation. An example is shown in Figure 2.9 (a) [13, 45, 46, 47]. The differential signals are selected from various cells to generate a new phase. Redundant cross points can also be generated by intentionally creating different rise and fall times. Due to the large number of inverters, the power consumption significantly increases compared with the passive option. Taking the combination of the two methods mentioned above, a matrix of interpolated phase is created [48], and an example is shown in Figure 2.10.

Another way to improve the resolution of the delay line is by eliminating the delay cells [49, 50]. Unlike the TDC topologies discussed above, which increase the circuit size to

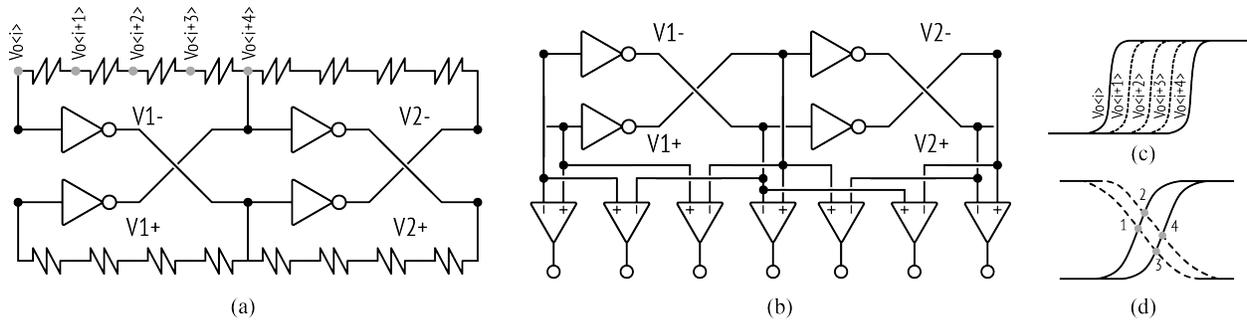


Figure 2.9: Schematics of delay lines with (a) the passive interpolation and (b) the active interpolation. And the example waveforms of (c) the passive interpolation and (d) the active interpolation.

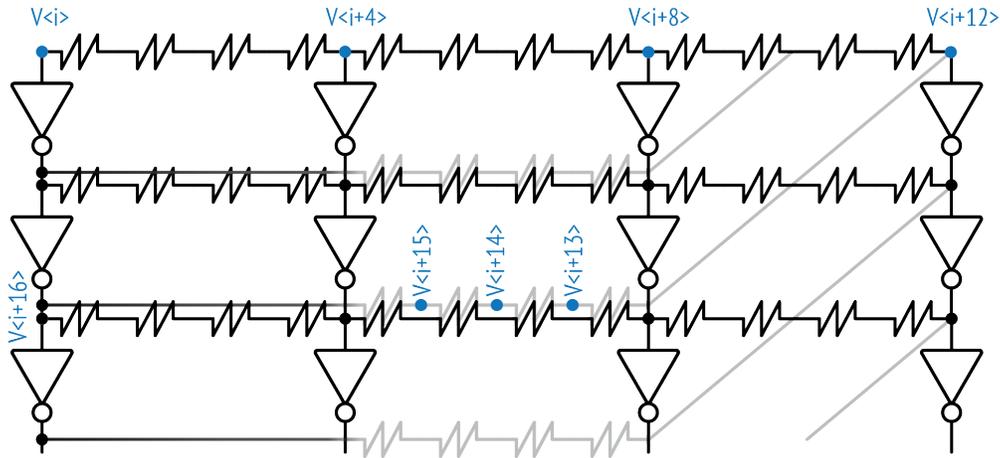


Figure 2.10: Schematic of resistively-coupled ring oscillators use both active and passive interpolations [48].

reduce random variation, the stochastic TDC leverages the random variation of the sampling flip-flops to eliminate the delay cells. Similarly, the offset of the sampling circuit can be intentionally adjusted, and the time-domain information is quantized by the resolving time of the comparator. The diagrams of such TDCs are shown in Figure 2.11

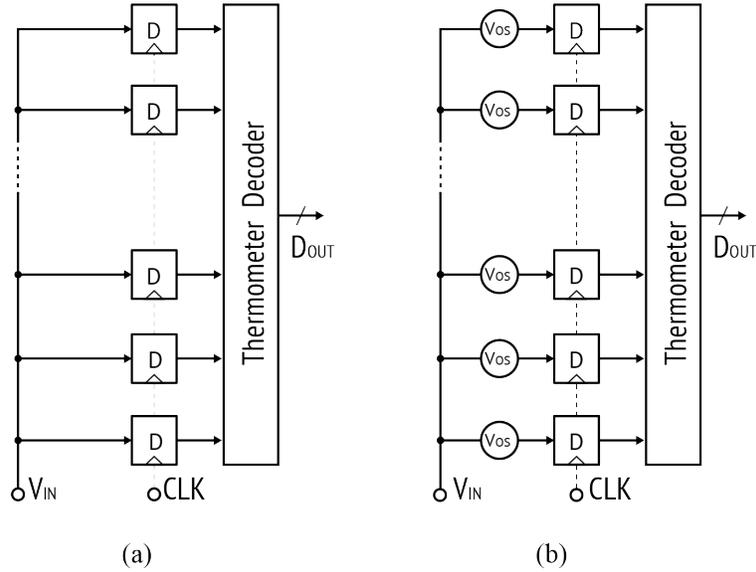


Figure 2.11: Schematics of stochastic TDCs with (a) sampling offsets and (b) unbalanced samplers.

2.3.3 VCO-Based ADC Design

2.3.3.1 Introduction

VCO-based ADC is another category of time-based ADC that utilizes a different method that combines voltage-to-time and time-to-digital conversion. Similar to the time-based ADC architecture, using a VCO for analog-to-digital conversion benefits from technology scaling and gate delay reduction. Therefore, VCO-based ADCs have gained significant attention in the last decade, primarily being used as quantizers or integrators in continuous-time sigma-delta ADCs. One of the earliest implementations of frequency sigma-delta modulation is proposed in [51]. The VCO is used as a high-speed quantizer in an oversampling sigma-delta ADC scheme [52]. Although most commonly used in the context of sigma-delta modulation, the VCO-based ADC can also be employed in open-loop architectures or hybrid ADCs.

2.3.3.2 Working Principle of the VCO-Based Core

Figure 2.12 shows a high-level diagram of a VCO-based ADC, in which the phase is measured by counting one phase of the VCO. The core of a VCO-based ADC is an oscillator. It consists of a control input and a ring oscillator (RO), that converts an analog input signal, denoted as $x(t)$, to the oscillation frequency of the RO, denoted as f_{VCO} . The controlled gain K_{VCO} is defined by

$$K_{VCO} = \frac{\partial f_{VCO}}{\partial V_{in}} \text{ (Hz/V)} \quad (2.4)$$

The RO output phase is defined as follows:

$$\Phi_{VCO} = 2\pi \int_0^{T_s} f_{VCO} dt, \quad (2.5)$$

where T_s represents the integration time. The transform function in the s-domain is

$$H(s) = \frac{2\pi K_{VCO}}{s}, \quad (2.6)$$

which is in the form of an ideal integrator with infinite gain. With a constant input voltage over the sampling period, the unknown voltage can be determined by measuring the change in phase $\Delta\Phi$:

$$V_{in} = \frac{2\pi\Delta\Phi}{K_{VCO}}. \quad (2.7)$$

Therefore, the most straightforward implementation of quantization is shown in Figure 2.12 (a). In this design, the oscillator is implemented as an N-stage RO. The phase measurement is performed by selecting one of the RO's outputs and observing the counter output at the beginning and end of the sampling period. The counter follows the VCO core and increments at the edges of the VCO output. Since only the edge of the output is detected, the output of the VCO is inherently quantized to the period of the RO.

The open-loop VCO-based ADC is inherently first-order noise-shaped. The signal and noise transfer functions can be derived as follows: Assuming the sampling period is T_s , the output of the VCO-based core during the n-th sampling period can be approximated as:

$$\Phi[n] = \int_{nT_s}^{(n+1)T_s} K_{VCO} x[n] dt + e[n-1]. \quad (2.8)$$

where $x[n]$ is the sampled input signal during the n-th period and the $e[n-1]$ denotes the leftover quantization error from the previous cycle, which can be considered as the starting phase. Each phase quantization generates a quantization error due to the residue phase $e[k-1] = \Phi(kT_s) - \Phi_q(kT_s)$, where Φ_q is the sampled phase, while $\Phi(t)$ represents the

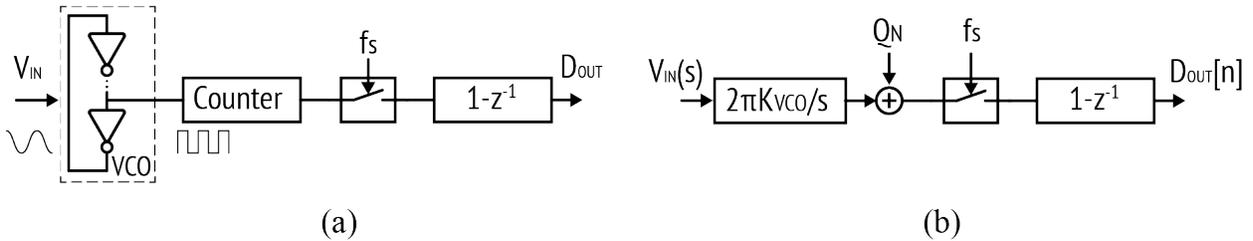


Figure 2.12: (a) High-level block diagram of the one-phase VCO-based ADC and (b) its equivalent model.

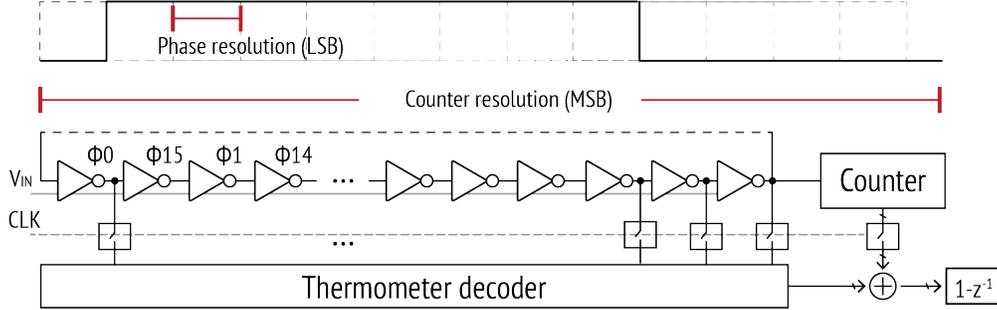


Figure 2.13: Illustration of phase quantization scheme in the VCO-based ADC.

continuous time phase. Assuming a constant input voltage, the equation above can be simplified to

$$\Phi[n] = Gx[n] + e[n - 1], \quad (2.9)$$

where $G = K_{VCO}T_s$ is the equivalent phase gain. And the final quantized output can be expressed as

$$y[n] = \frac{1}{2\pi}(Gx[n] + e[n - 1] - e[n]). \quad (2.10)$$

Taking the z-transform of the equation above gives

$$Y(z) = \frac{1}{2\pi}[GX(z) - (1 - z^{-1})e(z)]. \quad (2.11)$$

Therefore, the noise transfer function and the signal transfer function are

$$NTF = -\frac{1}{2\pi}(1 - z^{-1}) \quad (2.12)$$

and

$$STF = \frac{G}{2\pi}. \quad (2.13)$$

It can be seen that the quantization error of the VCO output is first-order shaped and equivalent to the first-order sigma-delta modulator. Therefore, the quantization error can be further shaped by increasing the oversampling ratio.

2.3.3.3 Open-Loop VCO-Based ADCs

In the simple example discussed above, the analog signal is digitized without any additional processing. The implementation is highly scalable due to its digital nature, while the main drawback is the nonlinearity of the oscillator. The resolution of the open-loop VCO-based ADC is difficult to define in the same manner as the traditional ADC, where the maximum achievable resolution is determined by the hardware implementation. Increasing the number of bits in the counter does not affect the resolution of the ADC at all. Instead, the resolution

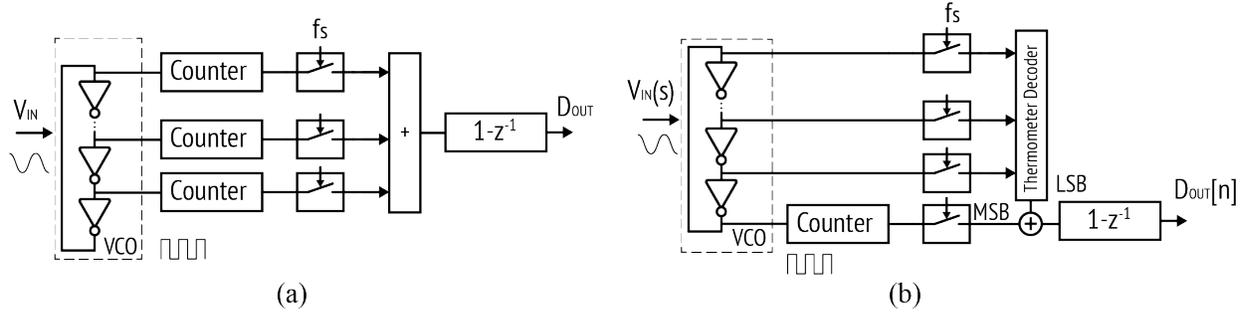


Figure 2.14: Diagrams of (a) the multiphase open-loop VCO-based ADC and (b) the open-loop VCO-based ADC structure with coarse-fine phase quantization.

can be defined by examining the minimum and maximum code it generates [53]. The code range R is defined as

$$R = \log_2\left(\frac{f_{VCO,max} - f_{VCO,min}}{f_s}\right). \quad (2.14)$$

Therefore, the ideal number of bits (ENOB) of the ADC is:

$$ENOB_{ideal} = \log_2\left(\frac{f_{VCO,max} - f_{VCO,min}}{f_s}\right). \quad (2.15)$$

The equations above show that increasing the resolution of the VCO-based ADC does not require additional hardware. By either halving the sampling rate or doubling the tuning range of the VCO, one extra bit can be achieved. These properties make the simple architecture shown above suitable for low-speed applications [54]. However, for higher speed requirements, the one-phase open-loop topology would need an unrealistic tuning range to obtain desired resolution.

One solution to this problem is to take advantage of the multiple phases in the RO. Conceptually, the idea is similar to the recirculating delay line discussed above. Figure 2.13 shows a diagram that demonstrates how the resolution can be improved from the counter's resolution (T_{per}) to T_{per}/N , where T_{per} represents the oscillation period and N is the number of stages in the RO. The output of the counter is the MSB of the quantizer, which is then combined with the decoded LSB from the RO's phase information. Such improvement can be implemented by taking the phases from each delay cell [55]. A diagram illustrating this is shown in Figure 2.14 (a). Assuming an N -stage RO is used as the VCO core, each stage has a uniform delay t_d . The resolution of the ADC can be improved to:

$$ENOB_{ideal} = \log_2\left(N \cdot \frac{f_{VCO,max} - f_{VCO,min}}{f_s}\right) = \log_2\left(\frac{1}{f_s t_{d,min}} - \frac{1}{f_s t_{d,max}}\right). \quad (2.16)$$

This multi-phase implementation requires a significant number of counters and reliable sampling and decoding of their output, which increases the complexity and power consumption.

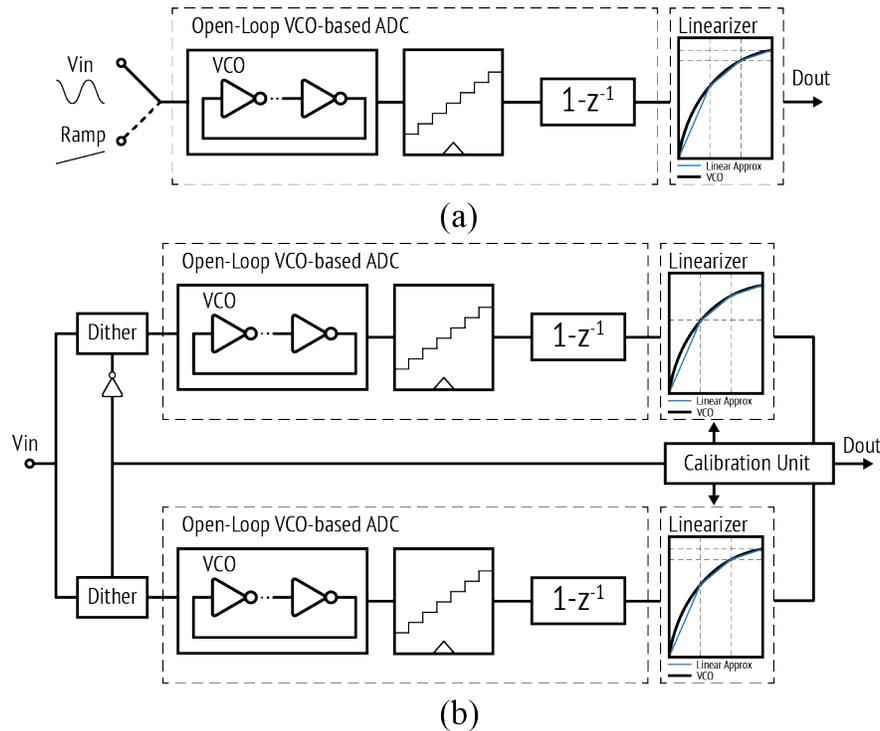


Figure 2.15: Diagrams of VCO-based ADCs with linearization techniques including (a) the LUT-based calibration [55] and (b) the ‘split-ADC’ technique [59].

Based on the observation that there are only two possible values at the output of each delay cell, all the counters attached to the internal phases can be replaced by a sample and decode block. This implementation allows for the ADC to have both coarse and fine quantization with minimum complexity [56, 57, 58]. A diagram illustrating the coarse-fine implementation is shown in Figure 2.14 (b). Moreover, the interpolation technique can further increase the quantization resolution [48]. Although a multi-edge counting scheme improves the resolution of the VCO-based ADC, the nonlinearity in the voltage-to-frequency tuning curve limits its performance. The pseudo-differential topology is commonly used to eliminate even-order harmonics, but higher-order harmonics persist.

2.3.3.4 VCO-Based ADC Non-linearity

Several techniques are proposed to improve the linearity of the VCO-based ADC. The most straightforward method is to use a look-up table (LUT) based linear interpolation [53, 48, 55], which is a relatively low-complexity technique. The diagram of the LUT-based calibration is shown in Figure 2.15 (a). The correct V-to-F curve is usually derived during foreground calibration using a ramp signal [55] or dithering injection [60]. A LUT stores the end points of each segment that is accessed by the MSB of the digital output. This process remaps

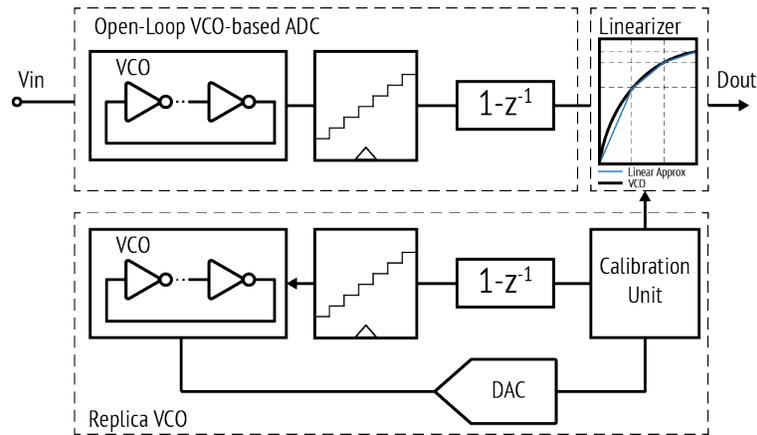


Figure 2.16: Diagram of the background VCO calibration process using a replica circuit.

the code using a linear interpolator to derive the correct voltage. However, the LUT-based methods do not track environmental variations and device aging. The work reported in [59] uses the ‘split-ADC’ concept, which implements an adaptation loop in the background to calibrate a pseudo-differential VCO-based ADC. The diagram of this calibration method is shown in Figure 2.15. A dithering signal is injected into the differential sides, and the calibration engine uses this information to align the V-to-F curve with the ‘correct’ linear curve.

Alternatively, a replica oscillator can be used in background calibration [61, 62, 63]. The concept of calibration relies on the use of the inverse of the V-to-F transfer function, denoted as $f^{-1}(\cdot)$, which is implemented by the replica calibration unit. A dedicated replica path runs in parallel with the quantization path to perform the reverse conversion. The diagram of the replica calibration method is shown in Figure 2.16.

2.3.3.5 Pulse Width Modulation, Pulse Frequency Modulation Based ADCs, and VCO-Based Sigma-Delta ADCs

For the completeness of the analysis, linearization techniques that are challenging to be applied in the target design are also reviewed in this section. A pulse width modulation (PWM) based ADC is proposed in [64] and [65]. A modulator first converts the analog signal into a two-level pulse train. Because only two points are used, the effective transfer curve is inherently linear. The diagram is shown in Figure 2.17 (a). The PWM signal is directly applied to the control voltage of the VCO, and the VCO quantizes the pulse width. While this approach relaxes the requirements of the oscillator, the design challenge shifts to signal modulation and sampling. The linearity and noise of a PWM-encoded signal rely on a robust PWM modulator [66]. Also, a high sampling frequency is required to prevent modulator harmonics from folding into the signal band. A slight mismatch between the

carrier frequency and the sampling frequency results in a dramatic decrease in performance [67]. A similar modulation approach that makes the VCO-based ADC a pulse frequency modulator (PFM) is proposed [68, 69]. The PFM modulation-based VCO is also intrinsically linear. The diagram of the PFM-based VCO ADC is shown in Figure 2.17 (b). The output of the VCO first goes through an edge detector. After that, a filter generates a pulse with a width of T_s at each rising edge of the VCO output waveform. Similar to the PWM-based scheme, the pulse is also sampled at a frequency of f_s . The modulation sidebands exhibit periodic nulls in the spectrum at multiples of the pulse width. By precisely matching the pulse width with the sampling frequency ($f_s = 1/T_s$), the folded harmonics in the signal band are first-order noise-shaped.

Another approach to address the nonlinearity issue in the VCO-based ADC and, at the same time, make full use of the noise-shaping property, is to use a VCO in a closed-loop $\Delta\Sigma$ ADC. Figure 2.18 (a) shows a classic first-order noise shaping $\Delta\Sigma$ ADC. As mentioned earlier in this section, the VCO naturally exhibits continuous-time integration in the phase domain. This characteristic makes it an excellent substitute for the voltage integrator. The phase integrator continuously accumulates the phase at interstage nodes and can be easily read out by a digital logic circuit. Moreover, unlike the voltage integrator that eventually saturates to the supply voltage as the integration time increases, the phase integrator can automatically wrap around, which enables an open-loop $\Delta\Sigma$ implementation [51]. Figure 2.18 (b) shows the VCO-ADC embedded in a closed loop. The input amplitude seen by the VCO control voltage is reduced, which relaxes the linearity requirement. Moreover, a two-step ADC using separate VCO-based quantizers, acting as coarse and fine quantizers, is proposed in [70]

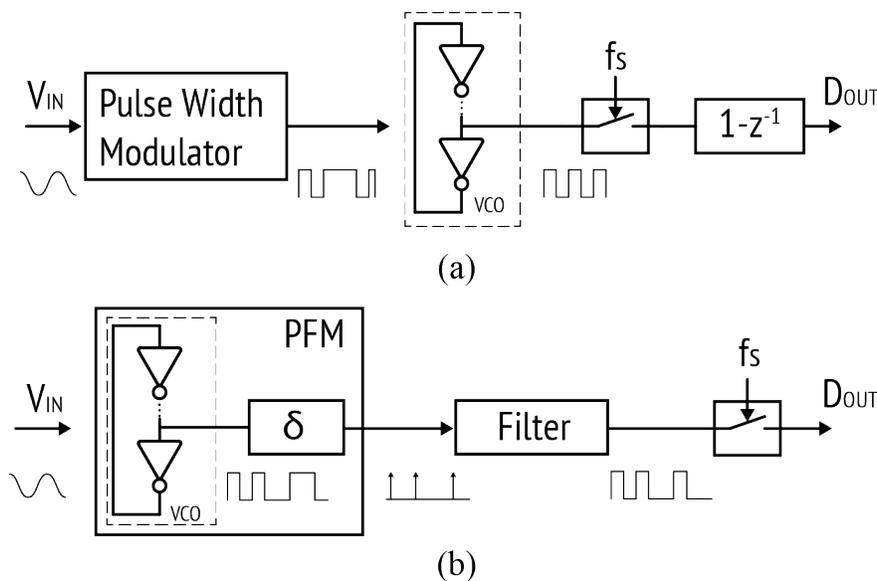


Figure 2.17: Diagrams of the (a) digital modulator for linearizing the VCO and (b) inherently linear PFM-based VCO ADCs.

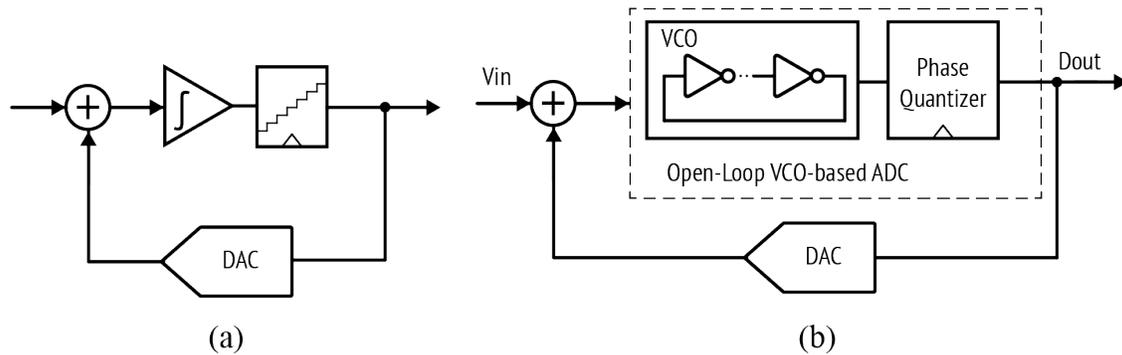


Figure 2.18: (a) Diagram of a general first-order continuous-time $\Delta\Sigma$ modulator and (b) the implementation with a VCO-based integrator.

to cancel out the nonlinearity of the first VCO. While the VCO provides first-order noise shaping, higher-order noise shaping can be achieved by using passive or active-RC filtering [71, 72, 73, 74]. Furthermore, higher-order noise-shaping can also be achieved by utilizing multiple VCO-based integrators [75, 76, 77, 78]. An example of a second-order architecture using two VCO-based integrators with up-down counters is shown in Figure 2.19 (b). An alternative method for implementing a high-order VCO-based ADC is through the use of multi-stage noise-shaping (MASH). For example, ADCs using VCO-based integrators in 0-1 MASH architectures [79, 80] and 1-1 MASH architecture [81] have been reported.

2.3.4 Summary

In this section, both delay-line-based time-domain data conversion and VCO-based data conversion are reviewed. Both techniques greatly benefit from technology scaling, resulting in a compact layout and competitive low power consumption compared to traditional architectures. In time-based converters implemented using the combination of VTCs and TDCs, the TDC consists mostly of digital circuits and benefits from high switching speed. However, the VTC is the remaining analog component that performs dynamic amplification and is thus limited by the reduction of the supply voltage. Therefore, VTCs often become limitations in design, hindering the achievement of sufficient linearity. Besides the difficulties in the VTC design, the resolution of TDCs is limited by the worse mismatch in scaled processes and the influence of PVT variations. Improving TDC resolution is another significant design challenge. The VCO-based ADC addresses the resolution issue differently. As the conversion time increases, the resolution of VCO-based ADC increases accordingly. Although the data conversion relies on the voltage-to-frequency transfer curve, which is an analog property of the circuit, the VCO scales well with digital logic gates. Calibration can be applied to correct the nonlinearity.

As for circuit generator implementations, both categories of architectures are suitable for the generator-based design methodology due to their highly digital circuit nature. The faster

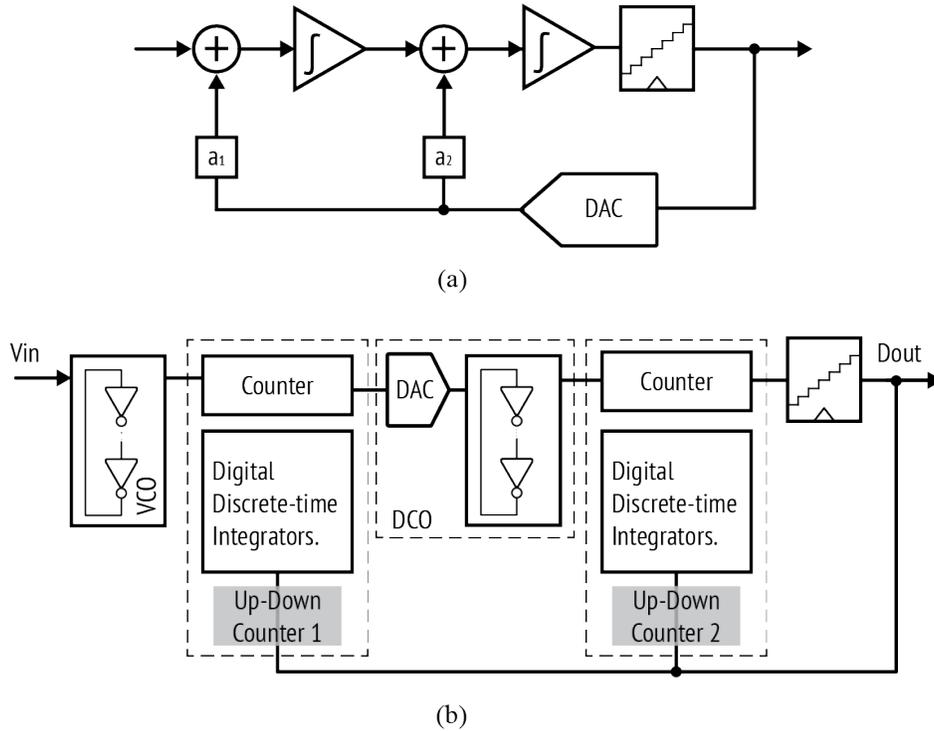


Figure 2.19: (a) Diagram of a general second-order continuous-time $\Delta\Sigma$ modulator and (b) the implementation with VCO-based integrators.

switching speed improves the resolution of time-based ADCs and makes such architecture attractive when porting to advanced process nodes. The VCO-based ADC is selected in this work due to its scalable resolution and straightforward implementation compared to the VTC design. Although mostly used in oversampled ADCs, the VCO-based ADC has also been proven to be suitable for high-speed conversion applications.

In summary, this work uses VCO-based topologies to develop an ADC generator for high-speed applications due to their scalability and digital-centric nature. The topology of the open-loop VCO-based ADC provides a regular floorplan that is highly repetitive and extendable, making the generator-based design a good option. Since the open-loop VCO-based ADC is more suitable for converting small amplitudes with medium-to-low frequencies, the VCO-based ADC generator is used as the second stage in the hybrid architecture.

2.4 Hybrid ADC Architectures

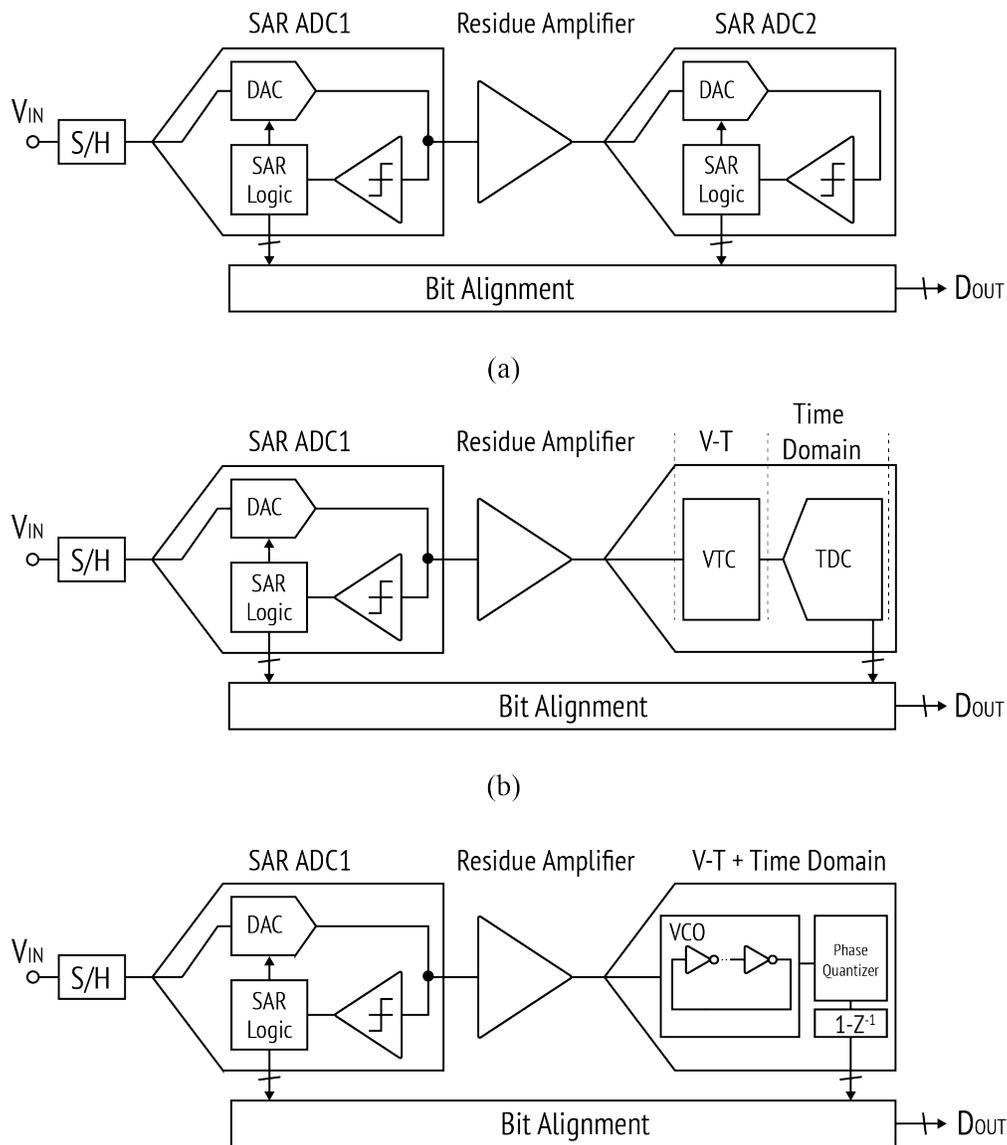


Figure 2.20: Diagrams of (a) a pipelined SAR ADC, (b) a pipelined SAR-TDC, and (c) a pipelined SAR-VCO ADC.

Through a careful blend of architectures, hybrid ADCs can optimize conversion speed, resolution, power consumption, and area requirements to meet the specific needs of an application. The constraints for each ADC architecture are reviewed in previous sections. The flash ADC, despite its high speed, is typically limited to low-to-medium resolution applications because its energy efficiency decreases significantly as the resolution increases. The SAR ADCs are limited by the power consumption of low-noise comparators when pushing toward higher resolution. Additionally, the sequential SAR operation limits the highest achievable conversion rate. The pipelined ADC, on the other hand, often requires precise amplification, which is challenging to implement within advanced nodes and requires extensive calibration. Regarding time-based ADCs, the intrinsic gate delay decreases as the process scales. However, the resolution of a time-domain ADC cannot continue to improve indefinitely due to mismatch and PVT variation limits. Consequently, this results in a resolution constraint when considering single-stage time-based ADCs. Pipelined and sub-ranging ADC architectures effectively address the inherent limitations of single-stage ADCs by combining the strengths of different types. Figure 2.20 shows several different hybrid implementations that are possible. The pipelined SAR ADC has emerged as a favored hybrid architecture [82, 83, 84, 85, 86, 87, 88, 89, 90, 91]. Figure 2.20 (a) shows a simplified diagram of the pipelined SAR architecture. The S/H block and the initial stage SAR implementation are identical to a traditional SAR ADC. The quantization error is amplified after the conversion in the first stage. In the second stage, the SAR ADC's S/H block can be absorbed into the residue amplification block. The data from these two stages are combined to produce the final digital output. The pipelined SAR is not limited to two stages; a higher number of stages [92] can also be implemented, presenting a design trade-off between power and speed. The pipelined SAR ADC offers several advantages compared to both conventional pipelined ADCs with flash sub-ADCs and conventional SAR ADCs. The integration of the SAR ADC into a pipeline structure eliminates the need for a dedicated S/H circuit in the pipelined ADC. By transforming the sequential SAR algorithm into a pipeline operation, the speed of the single-stage SAR ADC can be enhanced. Concurrently, the amplified voltage in the backend stage mitigates the noise requirement for the low-noise comparator. Moreover, the overall energy efficiency is improved, and the power dissipation of the amplifier is reduced as the resolution of the first stage increases [93]. Owing to its similarity to the MDAC in traditional pipelined ADCs, redundancy theory can also be directly applied to the pipelined-SAR architecture.

In parallel to the pipelined SAR, but with a different fine-ADC selection, the SAR and TDC architecture combines a coarse SAR ADC and a time-domain ADC [94, 95, 96, 97]. The diagram of such an architecture is shown in Figure 2.20 (b). As a second-stage ADC, the small input swing and less resolution requirement make the VTC and TDC converters an ideal choice. Similarly, the VCO-based ADC is often combined with other architectures to implement a high-resolution ADC [98, 99, 100, 101]. The noise-shaping properties of the VCO-based ADC are leveraged, as mentioned in the previous section. An example diagram of such implementation is shown in Figure 2.20 (c). Once the SAR conversion finishes, the residue is fed to the VCO and quantized. The small residue voltage is suitable for VCO

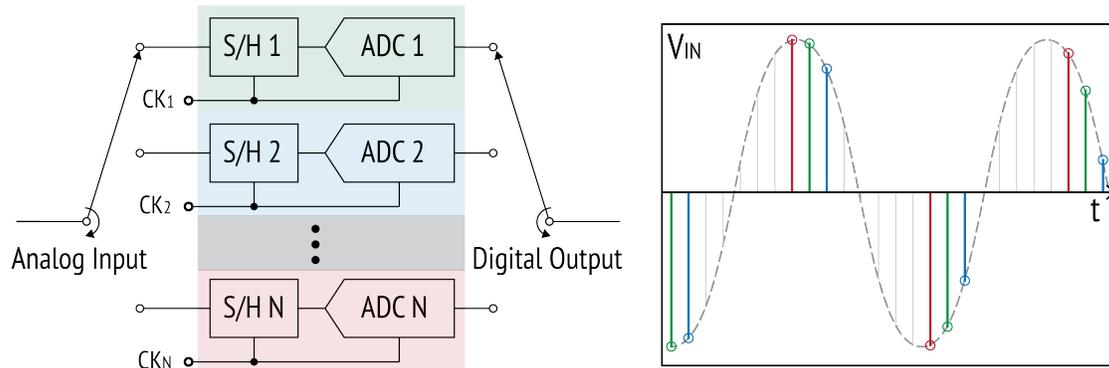


Figure 2.21: Illustration of the time-interleaved ADC.

operation because it alleviates the non-linearity issue. In these pipelined architectures, VCO-based ADCs have only been used in oversampling architectures. However, as mentioned in the previous chapter, the open-loop VCO-based ADC is fully capable of high-speed applications. Therefore, the proposed generator uses the VCO-based ADC as a high-speed ADC and utilizes it for the fine quantizer in subranging architectures.

2.5 Time-Interleaved ADCs

As shown in the preceding sections, a single-channel ADC often confronts a trade-off between speed and resolution. High-resolution ADCs typically operate at lower speeds. The time-interleaved technique is an effective solution for overcoming the speed limitations of single-channel ADCs. This is achieved by integrating multiple low-speed ADCs in parallel, interleaved in time. This approach allows for a higher aggregate sampling rate while maintaining the resolution of the sub-ADCs. The choice of architecture for sub-ADC implementation is determined by the specific application. Figure 2.21 illustrates the working principle of the time-interleaved ADC. In this configuration, multiple channels are running in parallel with each channel's sampling clock signal phase offset by $2\pi/N$ for adjacent slices. The input is sampled sequentially, and the digital output is multiplexed to construct the digital output. In this architecture, each channel is allocated more time for acquisition and conversion. Figure 2.22 compares the time-interleaved architecture with a single-channel ADC from the perspective of speed-energy trade-off. As the single-channel ADC approaches its speed ceiling, denoted as f_s , the power consumption escalates dramatically. Transitioning to the time-interleaved architecture raises this limit to $M \times f_s$, where M represents the number of ADC channels. This, however, introduces a power overhead due to the supporting circuits. For example, a dedicated clock generator is usually required to evenly and accurately distribute the sampling clock signal across 2π of the phase, along with other supporting circuits.

2.5.1 Error Sources in Time-Interleaved ADCs

Despite their speed benefits, time-interleaved ADCs suffer from mismatches among different channels, such as offset, gain, and time mismatches.

Offset mismatch: Sources of offset in the signal chain, such as the comparator offset, can be shifted forward to the input as an input offset for the ADC, annotated as V_{os} , as illustrated in Figure 2.23. The time-domain waveform exhibits a repetitive error occurring every N/f_s . Moreover, the average offset in each channel generates a DC component. The frequency response of the offset-induced spurs peak at

$$f_{\text{offset,spurs}} = \frac{N}{k} f_s (k = 1, 2, \dots, N). \quad (2.17)$$

Figure 2.24 shows the frequency domain spurs caused by the offset mismatch in an 8-way interleaved ADC.

Gain mismatch: In Figure 2.23, the gain mismatch is indicated by a different gain in each interleaved slice: G_i . Gain mismatch manifests itself as different slopes in the transfer curves of different sub-ADCs, which can be attributed to various sources such as the sampling process and changes in reference voltages. For an N-way time-interleaved ADC, the error in the frequency domain recurs every N/f_s , with the amplitude of the input signal modulating the amplitude of the error. This effect manifests as an amplitude-modulated noise located at

$$f_{\text{gain,spurs}} = \pm f_{\text{in}} \pm \frac{N}{k} \cdot f_s (k = 1, 2, \dots, N). \quad (2.18)$$

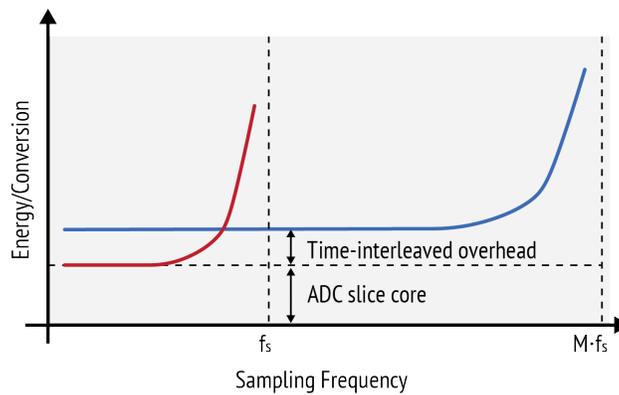


Figure 2.22: Comparison of energy per conversion between time-interleaved ADCs and single-channel ADCs.

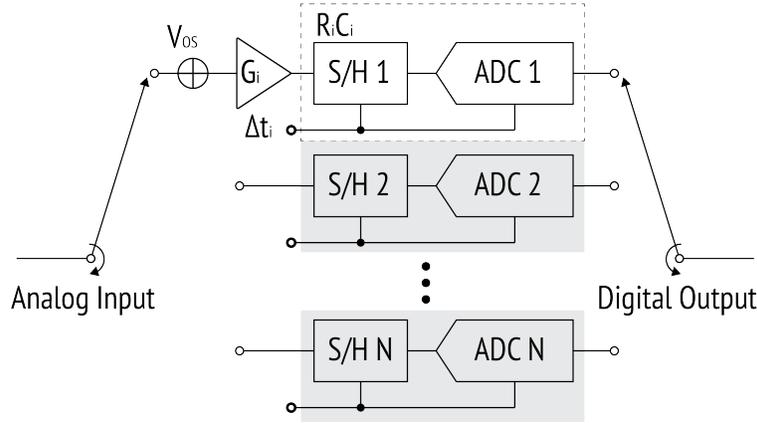


Figure 2.23: Illustration of error sources in time-interleaved ADCs.

The gain mismatch impairs the SNR, and the degree of performance impact also hinges on the amplitude of the input signal. Figure 2.24 shows the frequency domain spurs caused by the gain mismatch in an 8-way interleaved ADC.

Time mismatch: The mismatch caused by different sampling edges for each ADC is composed of both sampling clock skew (systematic error) and clock jitter (random error). It is shown in Figure 2.23 as Δt_i . This effect causes the largest error when the slope of the input signal is the steepest. In the frequency domain, it is essentially a phase-modulated noise, and the noise frequency peaks are also located at

$$f_{\text{time,spurs}} = \pm f_{\text{in}} \pm \frac{N}{k} f_s (k = 1, 2, \dots, N). \quad (2.19)$$

This error will overlap with the error caused by the gain mismatch. In contrast to the gain mismatch effect, such spurs are not obvious at low frequencies.

Bandwidth mismatch: Different sampling bandwidths for different channels can also lead to errors. The switches can be modeled as R-C networks. However, due to the variation in frequency response, the gain and phase differ for each channel during sampling. The equivalent effect of the process variation-induced bandwidth mismatch can be expressed as

$$V_{\text{sampled}} = G_i \cdot A \cos(2\pi f_{\text{in}} t + \theta_i), \quad (2.20)$$

where G_i and θ_i represent the gain and phase shifts, respectively, caused by sampling bandwidth mismatch. It causes both amplitude and phase-modulated effects and will also manifest itself similarly to gain and time mismatch. In the frequency domain, it also peaks at

$$f_{\text{BW,spurs}} = \pm f_{\text{in}} \pm \frac{N}{k} f_s (k = 1, 2, \dots, N). \quad (2.21)$$

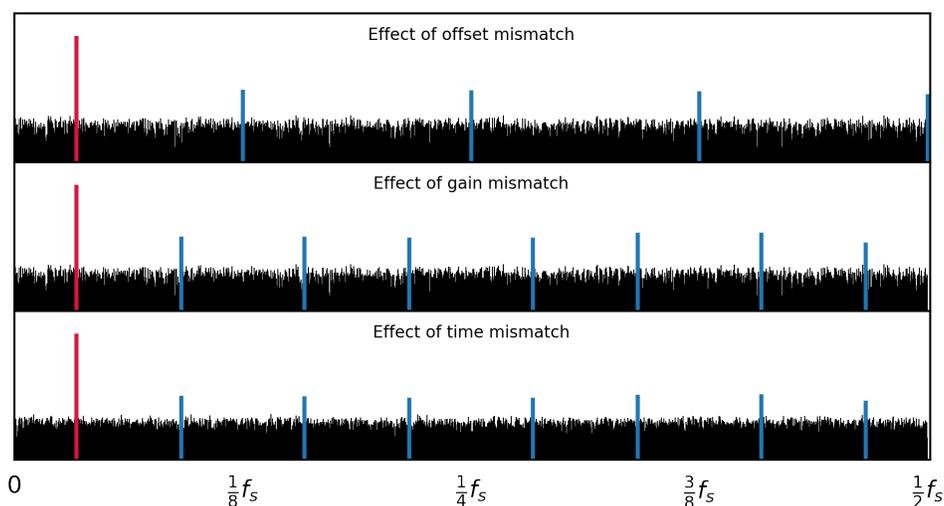


Figure 2.24: Illustration of time-interleaving spurs caused by offset, gain, and sampling time mismatch.

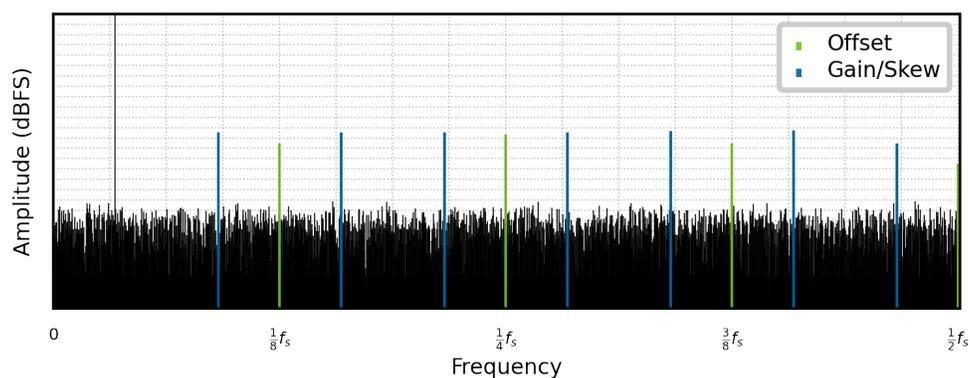


Figure 2.25: Illustration of time-interleaving errors in an 8-way time-interleaved ADC.

To summarize, the frequency response of a model for an 8-channel time-interleaved ADC is shown in Figure 2.25. Different frequency components are indicated in the plot. The effects of gain and timing skew mismatch are added together. It is clear that these non-ideal effects need to be calibrated out; otherwise, they will significantly degrade the performance of the time-interleaved ADC.

2.5.2 Time-Interleaved VCO-Based ADC

This work includes an open-loop VCO-based ADC, which naturally has a first-order noise-shaping property as analyzed in Section 2.3.3.2. Using the VCO-based ADC in the time-interleaved architecture leads to a modification to its transfer function [102] that will be

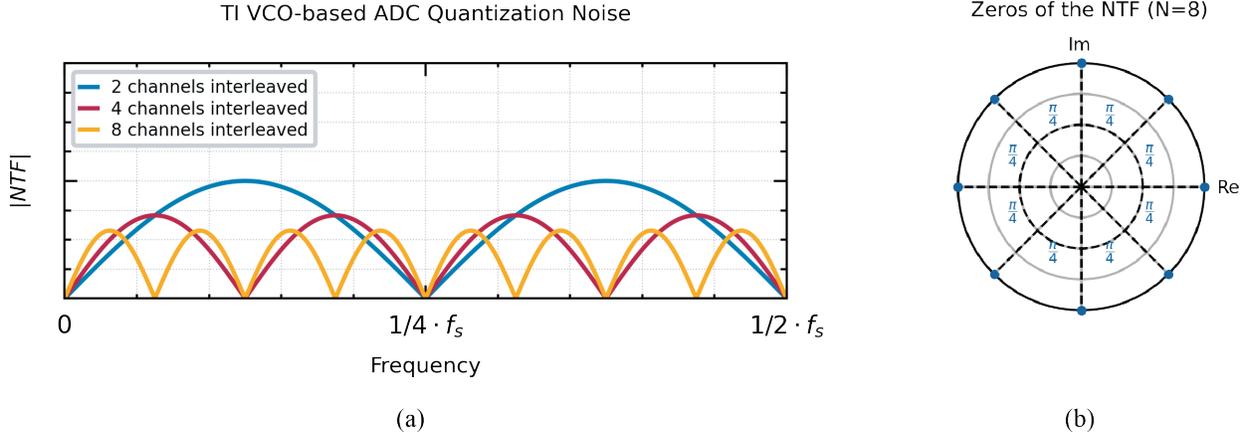


Figure 2.26: (a) Comparison of the time-interleaved VCO-based ADC quantization noise with $2\times$, $4\times$, and $8\times$ interleaved and (b) the location of zeros in the NTF of an $8\times$ interleaved VCO-based ADC.

discussed here. As mentioned in section 2.3.3.2, the open-loop VCO-based ADC is equivalent to a first-order $\Delta\Sigma$ ADC. In an M -way time-interleaved ADC, the transfer function is modified to

$$y[n] = \frac{1}{2\pi}(Gx[n] + e[n - M] - e[n]). \quad (2.22)$$

Therefore, the noise transfer function has been changed to

$$NTF = -\frac{1}{2\pi}(1 - z^{-M}). \quad (2.23)$$

which has first-order zeros at

$$z_k = \cos\left(\frac{2\pi k}{M}\right) + j \cdot \sin\left(\frac{2\pi k}{M}\right). \quad (2.24)$$

As examples, Figure 2.26 (a) shows the noise transfer function of $2\times$, $4\times$, and $8\times$ interleaved VCO-based ADC. And Figure 2.26 (b) shows an example of zeros in an 8-way time-interleaved VCO-based ADC architecture. In the frequency domain, the zeros create periodic nulls. Although this property can potentially be used in applications that require a band-pass ADC, it is not utilized in this work as this work focuses on a general-purpose ADC. Moreover, using the VCO-based ADC as a back-end stage for fine quantization typically results in the smearing of this shaped quantization in the spectrum due to thermal noise. As a result, the noise shaping becomes less noticeable in the spectrum.

2.6 Summary

This chapter reviewed the basic architectures for Nyquist-rate ADCs in both the voltage and time domains. As a summary of the performance each topology can achieve, Figure 2.27 [103] plots the signal-to-noise-and-distortion ratio (SNDR) against the sampling frequency of ADCs published at VLSI and ISSCC conferences, while Figure 2.28 shows the trend of ADC architecture selections over the years. As shown in Figure 2.27, hybrid and time-interleaved architectures are necessary for implementing a high-performance design. This requires incorporating optimized building blocks and supporting circuits in the ADC generator framework.

Both traditional SAR ADCs and VCO-based ADCs were selected as key building blocks in this work. These ADC architectures benefit from technology scaling and eliminate the need for accurate analog components, resulting in high energy efficiency. To achieve a higher resolution design, the amplification function is necessary. The ADC generator adopts the ring amplifier for the residue amplification, enabling the combination of SAR and VCO-based ADC into a pipelined architecture. The ring amplifier-based residue amplifier can provide a high input and output swing, as well as sufficient interstage gain, making it an ideal choice for the ADC generator. Design details will be presented in Chapter 4. As an effective method for improving conversion speed, the time-interleaving technique enables the ADC generator to cover a wider range of specifications. Therefore, it is also critical to support time-interleaving architectures in the ADC generator. Meanwhile, calibration circuits are included to help correct errors in such architectures. As a demonstration of the generator-based design methodology, prototypes implemented using the ADC generator have

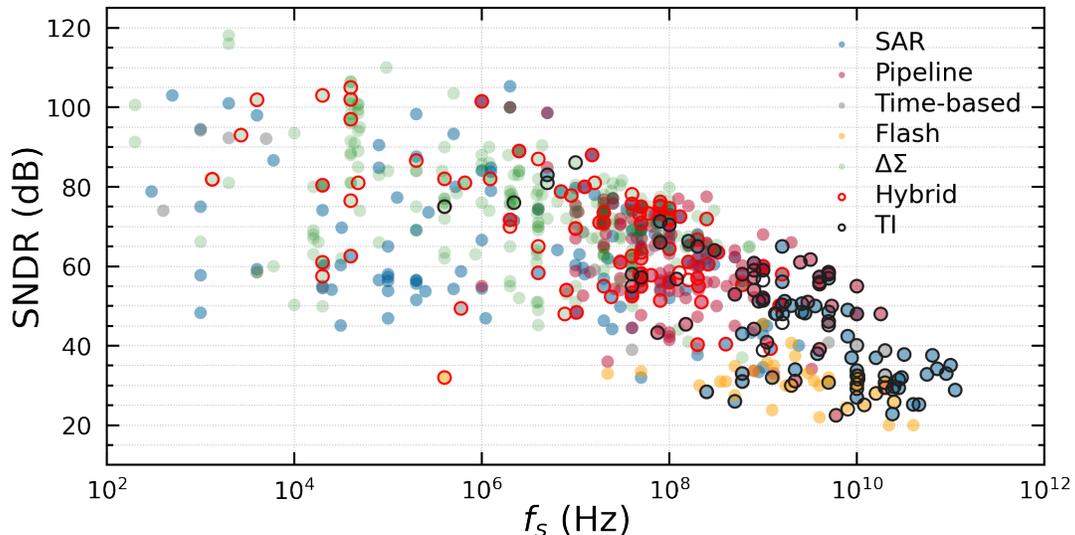


Figure 2.27: Summary of performance for each ADC architecture including the SAR, pipelined, time-based, flash, $\Sigma\Delta$, hybrid and time-interleaved (TI) [103].

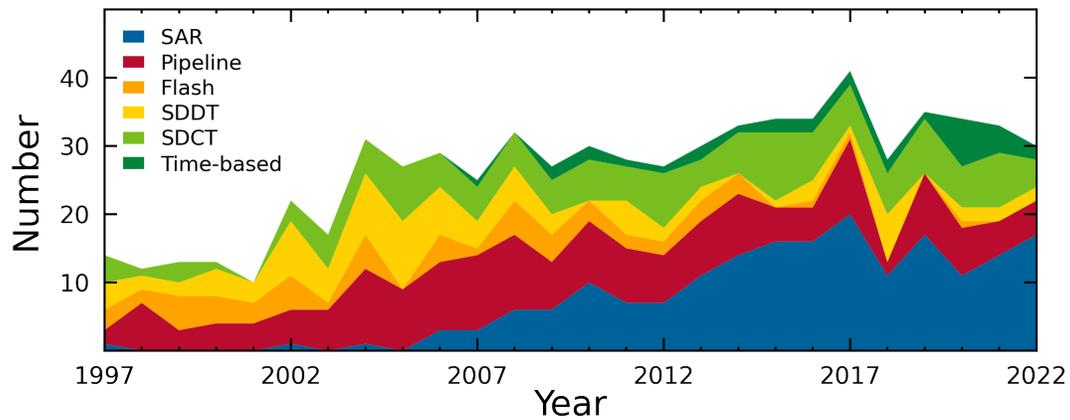


Figure 2.28: The numbers of ADC each architecture published over the years, including SAR, pipelined, time-based, flash, discrete-time $\Sigma\Delta$ (SDDT) and continuous-time $\Sigma\Delta$ (SDCT) [103].

a time-interleaved SAR-VCO architecture due to time constraints. More architectures can be implemented with little overhead in the generator code. For example, a pipelined SAR ADC can be supported by combining two or more low-resolution SAR ADCs with a dedicated amplifier and switch network. With this addition, the generator can support higher speed or resolution applications by leveraging the strengths of different architectures.

Chapter 3

Analog Design Automation and BAG Workflow

3.1 Introduction

In the realm of high-speed circuits, layout quality factors such as parasitic resistance and capacitance, matching, area, and other layout-dependent design trade-offs critically influence design choices, circuit functionality, and achievable performance. Therefore, implementing a process-portable automated ADC generator that encompasses various specifications and achieves performance comparable to a manual layout is challenging. Even with a validated circuit architecture, porting the same design to a similar process necessitates iterative optimization of the sizing and layout improvements. Moreover, formalizing the intricate design trade-offs in an ADC into a definitive set of constraints and objective functions is challenging. This typically requires the intuition and expertise of designers to comprehend the design problem and make appropriate changes. Therefore, the intricacies in analog circuit design and the goal of creating an automatic and process-portable generator motivate using the BAG [11, 104] to implement the design. This chapter briefly discusses various approaches to analog circuit generation. Additionally, the architecture and workflow of the BAG framework are introduced, with examples that demonstrate circuit generation, design optimization, and prototype integration flow.

3.2 Analog Circuit Automation

While digital circuit design tools have significantly improved over the decades, the analog design process has not changed much. A conventional analog design process consists of topology selection, device sizing, layout drawing, and repetitive verifications. Designers typically rely on empirical calculations and comprehensive simulations to iterate until the design meets the target specification. Although analog circuits have fewer transistors compared to

digital circuits, they present unique challenges that make them more complicated than their digital counterparts in several aspects:

- Highly customized layouts make analog circuits more time-consuming and susceptible to errors. Circuit performance is sensitive to layout-dependent effects. Numerous layout techniques are often necessary to maximize performance and prevent degradation.
- Apart from explicit constraints, such as technology constraints aimed at ensuring the design is ready for fabrication, there are also implicit constraints such as sensitive nodes in circuits that are sensitive to parasitics and transistors that require precise matching. These constraints are often based on underlying assumptions or the designer's experience. Design knowledge and expert inputs are difficult to repurpose, even when dealing with process porting and layout reuse.
- The circuit design progresses by eliminating feasible solutions at each design stage [105]. Unlike digital circuits, analog designs do not have continuously decreasing design freedom; each iteration may require a modification at a previous design stage. This broad feasible design space makes automation in analog circuit design challenging.

Although numerical methods [106, 107] have been proposed to assist in sizing, considering specific design constraints, the majority of analog design iterations are still manual. Consequently, these complexities in analog circuit design necessitate the development of an automation tool. Several methods for generating analog circuits have been proposed. Based on the degree of generality and the level of required human interaction during the generation process, generation tools can be categorized as follows:

Digital place and route: This method utilizes digital automation tools and involves the use of a custom analog design database. An array of analog designs is implemented, and the necessary files for the digital tool are prepared to make the analog blocks synthesis-friendly [108, 109]. Several mixed-signal circuits have been implemented using this method, including $\Delta\Sigma$ ADC [110], SAR ADC [111], LDO [112], and PLL [110]. While this method allows for the utilization of advanced commercial digital tools, it necessitates a significant amount of time to scale up the 'analog standard cells' and transfer them to various processes. Moreover, the resulting circuit instances have limited performance. Therefore, this approach is more suitable for circuits with higher tolerance to parasitics.

Optimization-based methods: The optimization-based approach [113, 114] targets generalized circuits and aims to minimize human involvement in order to achieve end-to-end circuit generation. Here, the circuit layout is synthesized using optimization techniques, and designers formalize the constraints as standard optimization problems. This method eliminates the need for manually creating the design database for synthesis or generator scripting.

Procedural or template-based methods: The template-based method [115, 116, 117] is commonly used for layout retargeting, process porting, or optimizing an existing design. In essence, procedural methods encode the layout based on a "stick diagram" floorplan. Circuit constraints and designers' knowledge are also embedded in the code. Due to the interaction between designers and the design tools in the procedural or template-based methods, the circuit performance can be enhanced, resulting in a high post-layout performance for a specific design. The ideal framework for such generators should enable the generation of fabrication-ready layouts that satisfy technology constraints. In addition, application programming interfaces (APIs) for verification execution and design script construction are beneficial for porting designs between different processes.

The BAG framework was chosen to develop the proposed generator in this work. As a procedural layout generation tool, designers have full control over the circuit implementations which allows the generated designs to meet the high-performance targets. The BAG framework provides APIs instead of a predefined generation flow, allowing designers to organize their circuit design in the form of generation scripts for varied preferences and applications. The generator-based design approach also allows for parameterization of layout and schematic, as well as rapid iterations. Verification and modeling functions are also available to facilitate the closed-loop design process. Designers can automate simulations and parameter updates by coding automatic sizing scripts for well-studied circuits. Additionally, the BAG framework can be combined with machine learning for automated circuit design [118, 119].

3.3 BAG Framework Overview

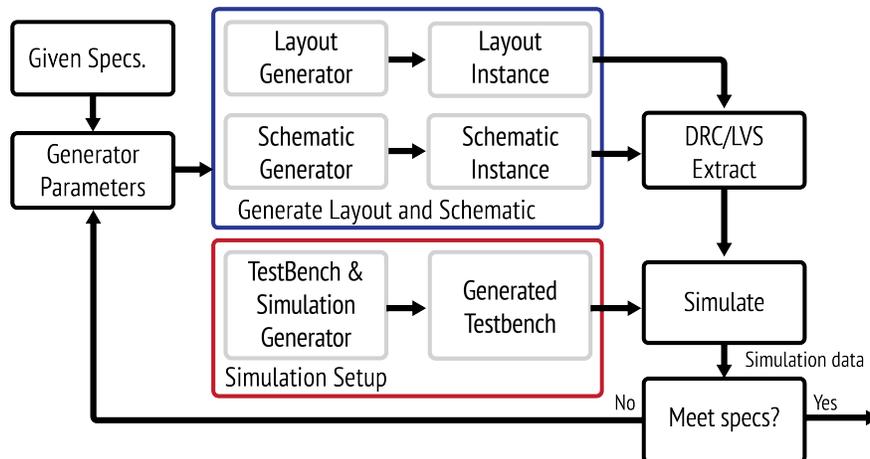


Figure 3.1: The analog design flow using the BAG framework.

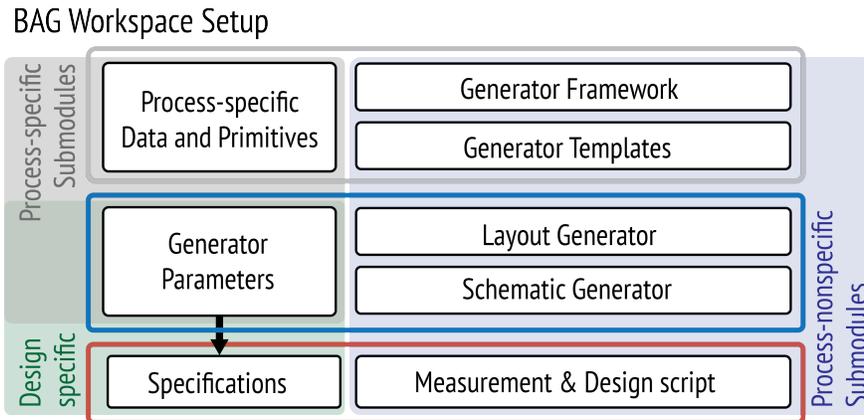


Figure 3.2: Diagram of a typical BAG workspace setup.

Figure 3.1 shows a typical BAG workspace setup. The generator framework, which consists of process-independent modules, provides APIs for communication between Python and commercial computer-aided design (CAD) tools. It enables the generation of schematics and layouts, the execution of DRC and LVS checks, and the running of simulations. The framework includes abstract layout classes in Python, which offer standard APIs for designers to implement the layout floorplan and schematic modification. Moreover, the track system enforces the use of an abstract routing grid with quantized widths and spaces. The generator framework hides the process-specific information under the standard interfaces and encapsulates the design rules within process-specific characterization. The process-specific modules include the setup of layout primitives, track system settings, and schematic templates for different processes. The generator parameters are also process-dependent. The design of parameterized and process-portable circuit generators relies on the fact that the circuit floorplan typically exhibits numerous invariant characteristics when implemented in various processes. Therefore, designers can develop automated and process-independent circuit generators by combining process-independent interfaces with technology-specific settings. Furthermore, by accurately configuring process-specific generator primitives, the generated instances are guaranteed to be DRC- and LVS-clean. This demonstrates the adaptability of generator scripts, as they encapsulate the design methodology and can accommodate varying input parameters. Therefore, generating distinct instances can be achieved simply by modifying these input parameters.

Figure 3.2 shows a typical BAG circuit design flow. From the given high-level specifications, designers can convert them into structural generator parameters. These parameters are then passed to both the layout and schematic generators in order to produce corresponding schematic and layout instances. To ensure that generators can accommodate diverse circuit changes, designers should avoid using hard-coded parameters within the generator scripts. Moreover, the generator framework provides a set of standard APIs that enable the

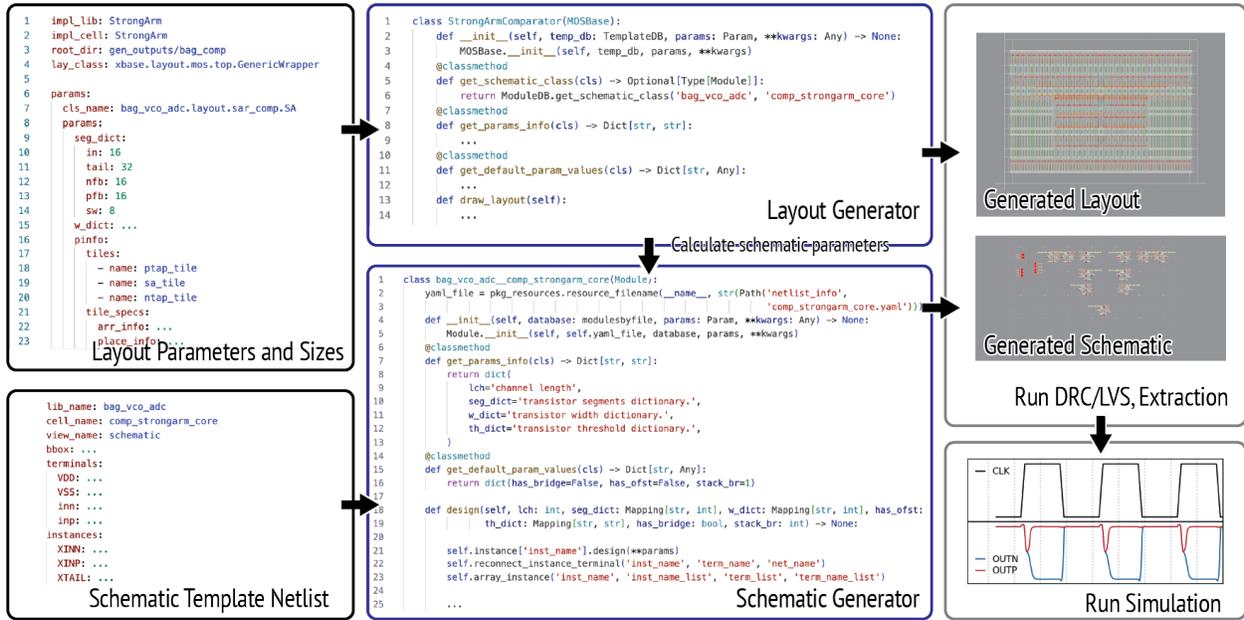


Figure 3.3: Examples of schematic and layout generations in BAG.

creation of testbenches and measurement scripts, thereby speeding up circuit design iterations. In addition, the framework allows designers to formalize their design procedures into design scripts. This feature integrates the procedures of schematic and layout generation, extraction, simulation, and resizing into automatic design iteration loops, thereby enhancing the efficiency and effectiveness of the design process.

3.4 Schematic and Layout Generation

An efficient, automatic, and process-portable circuit generator requires an appropriate process-specific setup that encapsulates the design rules and generator scripts capable of managing various parameters. Figure 3.3 demonstrates the schematic and layout generation using the BAG framework. Several layout generation engines are available in two versions of the BAG framework (BAG2 and BAG3). Details about each of the engines are shown in Appendix A. Similar to the traditional design process, an initial schematic is implemented as a template for the circuit, shown in the lower-left corner of Figure 3.3. This template netlist is extracted from pre-configured schematic templates that delineate ports, transistor names and locations, and symbol geometry. With the help of schematic parameters, the properties of devices in the schematic can be reconstructed and configured. For example, in the following code, transistors XN and XP in the schematic templates are configured based

on the given parameters.

```
self.instance['XN'].design(w=w_n, lch=lch, seg=seg_n, intent=th_n, stack=stack_n)
self.instance['XP'].design(w=w_p, lch=lch, seg=seg_p, intent=th_p, stack=stack_p)
```

This allows for replacing templates to accommodate changes in device type. When the number of instances is parameterized, the schematic generator can create an array of instances from a single instance template. Some useful APIs are shown in lines 21-23 of the Schematic Generator box inside Figure 3.3. Although passing the parameters of the schematic generator can enable direct schematic generation, the schematic parameters are typically derived from the parameters and calculations of the layout generator in the generator script to guarantee LVS correctness. Initially, the layout generator takes the required parameters and transistor sizes to implement the layout, and then it outputs parameters for the schematic generator. The parameters for layout generation typically include structural parameters such as transistor thresholds and sizes. An example of the input parameters is shown in the upper left corner of Figure 3.3. The generation of a process-portable layout relies on a circuit floorplan encoded within the generator, which remains unaffected by specific device sizes and design rules. The placement and width of connections are extrapolated from the circuit properties and the locations of transistors. This allows designers to place transistors according to the floorplan, rather than having to consider intricate geometric details while managing the connections without hard-coded values. The BAG framework's standard grid system for each process has been formulated to encapsulate metal design rules, with quantized widths and spaces derived from the same rules, ensuring compliance with design rules. Figure 3.4 presents a diagram of a simplified technology-specific setup. This configuration is dedicated to transistors, encapsulating their design rules and converting input parameters into layout geometries. Suitable boundaries and spaces are extracted from the information of the associated transistors. Routing-related data involves quantizing the width and spacing of all metal layers. Via geometries are defined in the primitive data and implemented based on the overlapping region.

Several base classes implemented in the Python framework are used to implement the layout generator:

- **TemplateBase** is the foundational class from which all layout generators inherit. It provides methods for unrestricted geometry drawing, as well as block and geometry placement within the BAG track system.
- **MOSBase** is a subclass of **TemplateBase**, which facilitates the drawing of transistor rows with varying properties and organizes lower metal layers that connect to transistors. **MOSBase** is used for all transistor-related generators, while passive devices are incorporated within the **TemplateBase**.
- **ArrayBase** is used to create device arrays, such as resistor arrays in the RDAC.

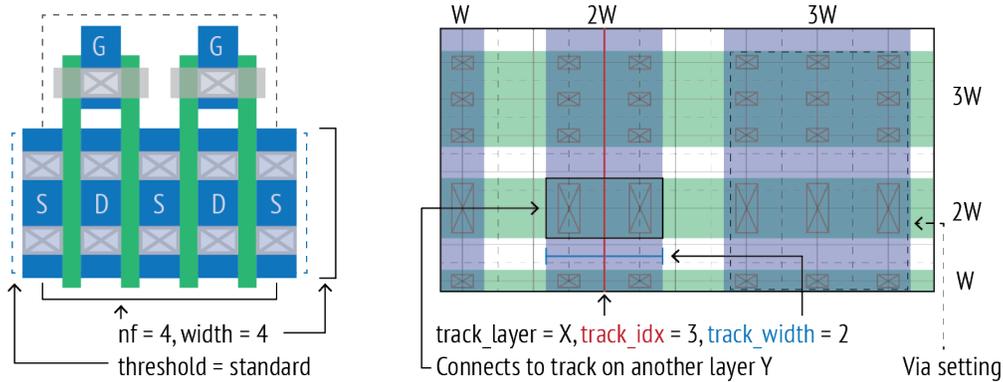


Figure 3.4: Illustration of the process-specific transistor primitives and routing grids setup.

Manual custom layouts can be integrated into the generator as black boxes. All sub-blocks are assembled within the `TemplateBase`. Various custom functions can construct passive devices by utilizing basic methods in `TemplateBase`. For instance, parallel wires can create a metal-oxide-metal (MOM) capacitor. Additionally, functions for tasks such as adding power straps and filling dummy structures are included to avoid repetitive coding. The implementations of layout classes are mostly inside the `self.draw_layout` methods, as shown in Figure 3.3’s Layout Generator box on line 13. An example code for the `MOSBase` is shown below that uses the `self.draw_base` method to initialize the floorplan and adds transistors using the `self.add_mos` method. Detailed examples can be found in Appendix B.

```
def draw_layout(self) -> None:
pinfo = MOSBasePlaceInfo.make_place_info(self.grid, self.params['pinfo'])
self.draw_base(pinfo)
# Add NMOS and PMOS
nports = self.add_mos(ridx_n, 0, seg_n, w=w_n, stack=stack_n)
pports = self.add_mos(ridx_p, 0, seg_p, w=w_p, stack=stack_p)
...
```

The power of a generator-based design methodology is illustrated in Figure 3.5, which displays different instances generated from the same SAR ADC generator script. For example, two instances with 8-bit and 5-bit are generated in the Intel 16 process. On the right, several instances generated in other processes using different process-specific primitive libraries are shown. Porting the generator across FinFET and planar processes can be challenging and relies on a well-established set of primitives. Pervasive options and flags are incorporated into the generator to handle various scenarios, including a restricted number of routing layers and different capacitor types.

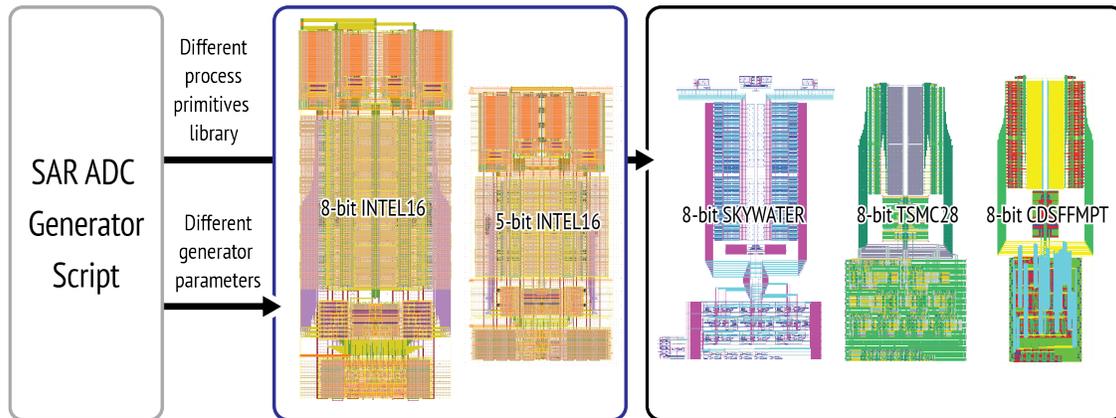


Figure 3.5: Examples of parameterization and process portability in a generation-based design.

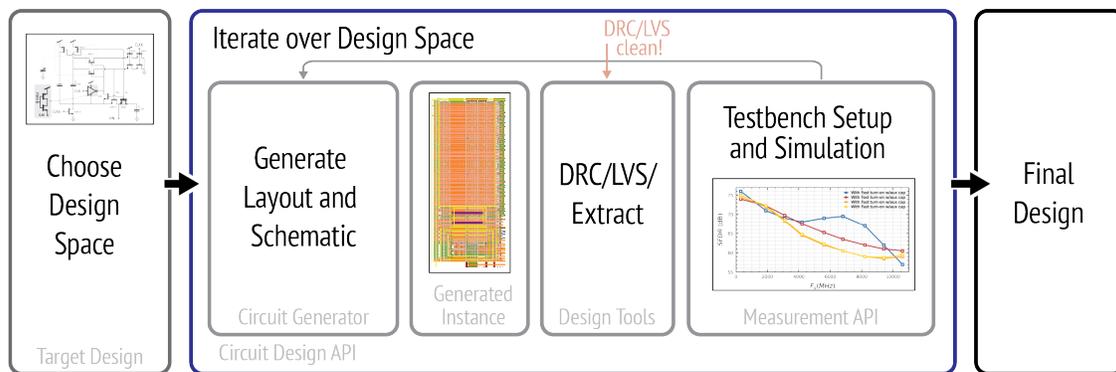


Figure 3.6: The design and optimization using BAG.

3.5 Design and Optimization Using BAG

In traditional analog design, circuit verification is typically specific to a design that utilizes a particular process. Furthermore, the design iteration process is primarily guided by the designer’s decisions, often without explicitly formalizing the design concepts and methods. However, the BAG framework provides APIs that allow designers to codify the design procedure into a design script. This script may include design equations that utilize technology-specific characterization data to calculate an initial starting point. This is followed by post-layout simulation iterations for parameter updates. In this process, BAG APIs assist in configuring measurement parameters, setting up testbenches, simulating instances, and post-processing simulation data. With the generation of schematics and the definition of the device under test (DUT)’s interfaces, verification testbenches, as well as design and

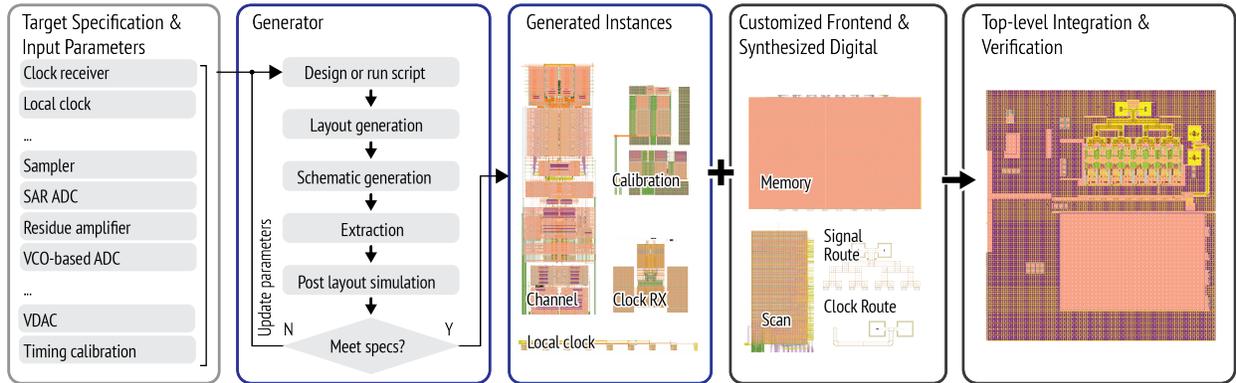


Figure 3.7: The integration flow of ADC prototype chips using BAG.

measurement scripts, become reusable. Using the BAG2 framework, studies in [119] and [118] exhibited circuit design methodologies using machine learning. While the work in this thesis does not focus on a closed-loop design algorithm for ADCs, several building blocks are optimized using BAG to fully exploit its measurement and design APIs. Figure 3.6 provides an example of sampler optimization using BAG, which will be discussed in detail in the next chapter. Various circuit techniques are implemented as generator options in the proposed generator. With a target sampling speed and load capacitor, a closed-loop script explores the provided design space and reports optimal performance sizes and options. The final optimal design is guaranteed to comply with DRC and LVS.

3.6 Implementation of Generator-Based Design Flow and Prototype

While the generator framework offers APIs to produce LEF and DEF files for easy instance integration in an SoC, the top-level integration of ADC generator prototypes is performed manually in this design. This section outlines the steps involved. Figure 3.7 outlines the steps in ADC design. Each block's specifications and initial parameters are defined. The generator scripts accept these parameters, create the design, and run the extraction. The performance of each block is verified manually or optimized by the design script to ensure that the target performance is achieved. Design iterations involve adjusting input parameters, regenerating the schematic and layout, and verification. The regeneration time for the circuit is usually in the range of tens of seconds, while for an ADC channel, it is only a few minutes. Thus, the design iteration process is significantly quicker than traditional flows. Manual work is limited to top-level integration and the design of prototype-specific and process-specific blocks, such as ESD, signal, and clock distribution. Additionally, the digital blocks are manually synthesized and integrated at the top level of the chip.

3.7 Summary

In this chapter, several methods of layout generation were reviewed, and the necessity of using the script-based methodology to achieve high circuit performance was discussed. The BAG framework was introduced. Schematic generation, layout generation, design, optimization, and the entire integration flow were presented.

Chapter 4

Building Blocks of the Proposed ADC Generator

4.1 Introduction

This chapter discusses the circuit design of process-portable ADC generators. The goal of the proposed generator is to accommodate various speeds, resolutions, and processes. The BAG framework enables automatic generation and fast technology porting. To support the design in advanced process nodes, topologies that are friendly to technology scaling are used. In contrast to traditional designs that depend on a specific hardware implementation, the generator provides multiple alternative options for each critical block in the ADC generator. This allows it to accommodate the ADC in various use cases.

This section first discusses the general considerations of ADC design within the context of the generator. The following sections focus on individual sub-ADC building blocks, such as the sampler, SAR, VCO-based ADC, and residue amplification. Available options for implementation in the literature are briefly reviewed, and the appropriate topologies are implemented as design options within the generator. Besides the optimized generator for a single-channel ADC, supporting circuits required for a complete time-interleaved architecture, ADC calibrations, and design prototyping are discussed. Lastly, the final section summarizes the architecture of the generator in terms of circuit and code construction. The building blocks described here were implemented in two prototypes, which will be discussed in the next chapter.

Generally speaking, the ADC generator is designed to support sampling rates of up to 4 GHz and resolutions greater than 10 ENOB. A passive sampling front end is used for low power and design simplicity, without an active input buffer. Therefore, the upper limit of the sampling rate is mainly determined by the linearity of the sampler. The architecture of the time-interleaved two-step hybrid ADC is shown in Figure 4.1. The diagram shows a minimum implementation of a prototype chip that allows the generated instances to be fabricated and the performance of the proposed generator can be measured. The analog

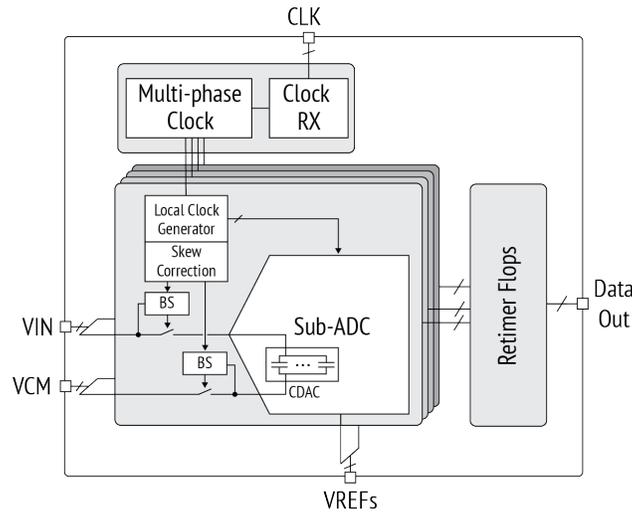


Figure 4.1: Diagram of the time-interleaved ADC implemented in the proposed generator.

blocks that need to be implemented as circuit generators are shown in gray. The design of the automatic portable process generator should include the following considerations:

- To support higher sampling rates, a time-interleaving architecture is necessary. Supporting circuits that enable time interleaving are included in the proposed generator, such as an ADC clocking circuit, a final retimer, reference generation circuits, and various calibration blocks. The sampler generator topologies are optimized to enable high-linearity sampling. Each slice in the TI ADC is designed to optimize conversion speed. With a higher conversion speed for each ADC slice, the number of interleaved channels is reduced and, consequently, decreases passive front end loading.
- For applications requiring a variable number of bits in the medium-to-low resolution range, the SAR ADC is an ideal solution. Moving toward higher resolution in a single-stage SAR significantly increases the power and necessitates precise matching or post-calibration circuitry. For this reason, a subranging topology is used. Time-domain data conversion techniques are explored, including the option of VCO-based ADC, which is also highly scalable. The VCO-based ADC can easily support various resolutions and sampling rates without the need for additional circuits. Combining VCO and SAR, the proposed generator creates a hybrid two-stage topology that supports higher resolution.
- Implementing the design as a circuit generator enables support for various speeds, resolutions, and high-level design constraints. For example, the SAR ADC can have varying numbers of reference voltages. More system design trade-offs can be explored by leveraging the high reconfigurability of this generator.
- Although only the single-stage SAR and SAR-VCO hybrid topologies are primarily used in the prototypes produced by the proposed generator, building blocks in the

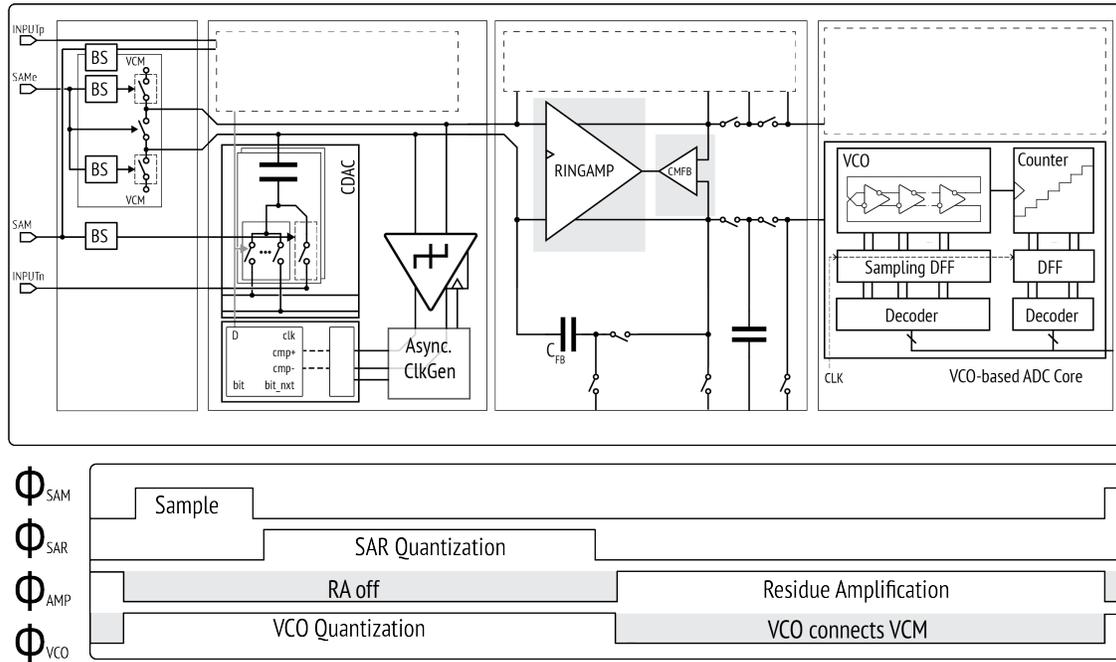


Figure 4.2: Diagram of the two-stage pipelined hybrid sub-ADC and the conceptual timing diagram.

generator are optimized and designed to incorporate more reconfigurability. This allows them to support different topologies by simply constructing an additional top-level circuit generator. For example, single-stage VCO-based ADCs, pipelined-SAR ADCs, and pipelined ADCs can be implemented by reusing residue amplification blocks.

The complete diagram of a two-stage hybrid ADC is shown in Figure 4.2, with all critical blocks assembled in a single-channel ADC. The timing diagram at the bottom of the figure shows the different phases of operation. During the sampling phase, the sampler tracks the input and samples the voltage on the CDAC of the first stage SAR ADC. Right after sampling, the first stage of the SAR initiates the conversion. The asynchronous SAR ADC is allocated a specific amount of time for conversion. The residue amplifier then converts the residue voltage on the CDAC and amplifies it before feeding it to the second-stage VCO-based ADC. During the amplification, the VCO-based ADC samples its input and then starts the conversion. The following section discusses these critical components in the order mentioned above.

4.2 Sampling Circuit Generator

Sampling switches are critical for the design of high-speed and high-resolution ADCs, particularly for time-interleaved architectures. The high-speed time-interleaved ADCs must

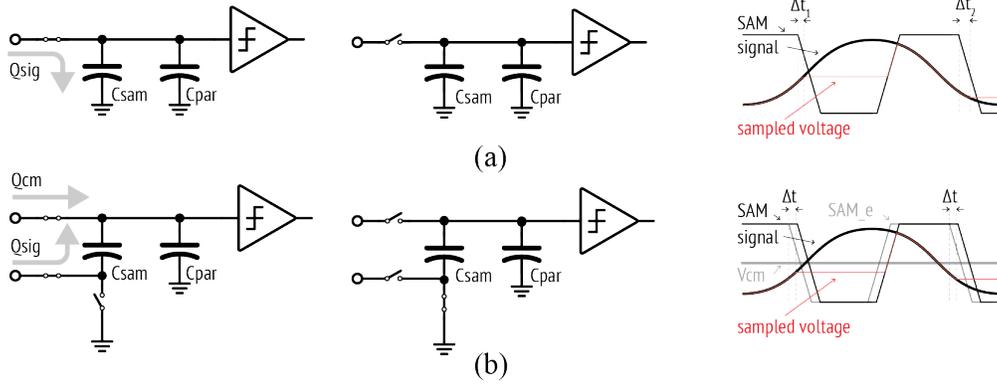


Figure 4.3: (a) Diagram of top-plate sampling and its waveform. (b) Diagram of bottom-plate sampling and its waveform.

sample the input signals within a short sampling window, typically on the order of hundreds of picoseconds. Concurrently, they must also ensure accurate charging of the sampling capacitor. With large input signal swings, the dominant nonidealities arise from distortions caused by the sampling switches. These distortions are primarily due to the switches' on-resistance during sampling and the charge injected from the channel of the transistors. Both of these factors vary with the gate-source voltages. Therefore, the proposed generator adopts the bottom-plate sampling technique and bootstrapped sampling switches to achieve sufficient sampling linearity when generating instance targets for high resolution.

4.2.1 Bottom-Plate Sampling

Figure 4.3 illustrates both the top-plate and bottom-plate sampling techniques. Only a single-ended model is shown for simplicity. Figure 4.3 (a) shows the top-plate sampling scheme, assuming that the sampling switch is implemented using an NMOS transistor. When the input signal is sampled on the top plate of the capacitor, the bottom plate is connected to ground. The actual disconnection instant of the sampling switch depends on the time when the gate voltage of the sampling transistor, V_G , drops to a value lower than $V_G - V_{IN} < V_{TH,n}$, where $V_{TH,n}$ represents the threshold voltage of the NMOS. As such, the delay between the sampling instant and the clock edge is variable. Figure 4.3 (a) depicts an example of different Δt_1 and Δt_2 . Additionally, the distorted channel charge Q_{sig} is sampled on the capacitor, and the on-resistance of the sampling transistor changes with the different input V_{IN} .

The bottom-plate sampling technique circumvents these problems by connecting the input signal to the bottom plate of the sampling capacitor and connecting the top plate of the sampling capacitor to a DC common-mode signal V_{CM} . The top plate of the capacitor disconnects slightly before the bottom plate switch turns off. Hence, the sampling charge is determined at the moment that $V_G - V_{CM} < V_{TH,n}$, which is a deterministic delay after the

actual sampling clock edge. Consequently, the two samples in Figure 4.3 (b) have the same delay, Δt . The technique also ensures a constant charge injection from the bottom-plate sampler, which can be canceled using a differential implementation. The charge stored on the top plate of the capacitor, C_{sam} , is determined after the top-plate switch is turned off, and Q_{sig} does not affect the final result. However, the bottom-plate sampling technique has drawbacks. For instance, it requires additional settling time for the capacitor DAC in certain switching schemes in a SAR ADC. Additionally, the presence of a parasitic capacitor at the top plate reduces the amplitude of the sampled voltage:

$$V_{\text{sampled,att}} = \frac{C_{sam}}{C_{sam} + C_{par}} \cdot V_{\text{sampled}}, \quad (4.1)$$

where C_{par} is the parasitic capacitance associated with the top plate of the CDAC.

4.2.2 Bootstrapped Switch Generator

For medium-to-low resolution ADCs with moderate speed requirements, a simple MOS or transmission gate sampler works adequately. The limitations of these samplers arise from their restricted conductance, which is dependent on the signal, and their signal-dependent charge injection. The output voltage during tracking can be expressed as

$$V_{out}(t) = V_{in}(t) - R_{on}C_{sam} \frac{dV_{out}(t)}{dt}. \quad (4.2)$$

This shows the effect of nonlinear conductance on modulating the output voltage. The bootstrapped switch is commonly used to achieve higher sampling linearity. The bootstrapped structure maintains a constant gate-to-source voltage, V_{GS} , for the sampling switch by applying a fixed voltage shift at the gate of the sampling transistor compared to the input signal V_S .

In this design, a bootstrapped switch generator is developed to meet the requirement for sampling linearity. Several modifications to the traditional design are offered as options in the generator to accommodate various operating conditions. The diagram of the generator is shown in Figure 4.4 (b); it is a modified version of the original design (Figure 4.4 (a) [120, 121]). The battery capacitor C_{boot} helps provide a constant voltage between the gate and source for the sampling switch. Therefore, it helps reduce the amplitude-dependent on-resistance modulation of the track-and-hold input. Moreover, the channel charge injected into the sampling charge when the switch is turned off is not amplitude-dependent. The transistor XSAM is the sampling switch, while the other components in the circuit are used to generate a gate-source voltage, V_{GS} , that is elevated by a constant value compared to the input voltage. During the hold phase, \overline{SAM} , XSAM is disconnected, and the battery capacitor C_{boot} is charged to V_{DD} by connecting the gate of transistor XCAP_P to ground. The C_{boot} is connected between the gate and source terminals of XSAM during the tracking phase (SAM). During the sampling phase, the gate voltage of XSAM rises above

the supply, necessitating the use of an additional transistor, XOFF1, to reduce the stress on the oxide of transistor XFF0. The gate of XON_P is connected to the input for the same reason. The bulk terminals of transistors XCAP_P and XON_P are tied together and connected to the top plate of C_{boot} to prevent forward-biasing when the source voltages of XCAP_P and XON_P exceed the supply. The voltage across C_{boot} is V_{DD} , and charge sharing results in $V_{GS} < V_{DD}$ after being connected to XSAM. Therefore, the gate oxide of the sampling switch remains within the reliability limits. The bootstrapping circuit eliminates the first-order signal-dependent modulation of V_{GS} and alleviates the nonlinear distortion of the on-resistance. The remaining nonidealities come from back-gate and higher-order on-resistance modulation, channel charge injection, and nonlinear parasitic loading in the critical circuit. These non-idealities become more pronounced at high sampling speeds and in time-interleaved architectures because the samplers have a shorter time to complete sampling. To analyze the non-ideality of the sampler, consider a sinusoidal input and a V_0 difference between consecutive hold phases as the boundary condition. The output voltage can then be solved as follows:

$$V_{out}(t) = \frac{A}{1 + \omega^2 R_{on}^2 C_{sam}^2} [\cos(\omega t) + R_{on} C_{sam} \sin(\omega t)] + V_0 e^{-\frac{t}{R_{on} C_{sam}}}. \quad (4.3)$$

Assuming the last term diminishes to a negligible value during the tracking phase and R_{on} also varies periodically with a sinusoidal input, the on-resistance can be expanded as a Fourier series:

$$R_{on} = R_0 + R_1 \cos(\omega t) + R_2 \cos(2\omega t) + \dots, \quad (4.4)$$

where R_i is the i -th Fourier coefficient. This suggests that the on-resistance of the sampler modulates the output voltage. The sampling circuit becomes nonlinear when the input frequency approaches the bandwidth of the sampler due to the memory effect. This mildly time-varying nonlinear system in discrete time can be analyzed using the Volterra series. In the proposed ADC generator, several techniques that improve the turn-on speed and sampler linearity are included in the sampler generator [122, 123] to address non-idealities and achieve high sampling linearity in different scenarios.

4.2.2.1 Fast Start-up Circuit

In the conventional bootstrapping circuit configuration, the critical loop comprises the series on-resistance of transistors XON_N, XON_P, and a combination of C_{boot} and the parasitic capacitance at node V_G . In the initial phase of tracking, the inverter pulls down the gate of XON_P, and the gate of XSAM is connected to the top plate of C_{boot} . The speed of V_G 's rising edge is limited by the parasitic capacitance associated with XON_P, XPD, and the output node of the inverter. The sampling switch adds significant overhead due to the large size required for the charging speed. Moreover, increasing the size of XON_P and the inverter does not work well due to the self-loading effect. Therefore, the transistor XON_N is not fully turned on until the voltage at its gate, V_G , reaches a sufficiently large value. Only after XON_N is partially turned on, the bottom plate of the C_{boot} is connected to

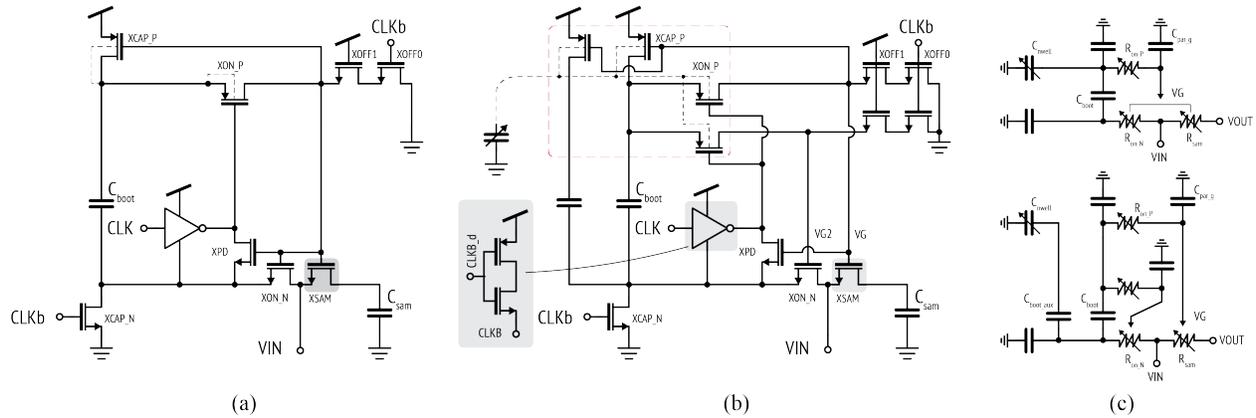


Figure 4.4: Illustrations of (a) the conventional bootstrapped switch [120, 121] and (b) the bootstrap switch implemented in the proposed generator with (c) its equivalent circuit.

the input, which raises the voltage of V_G . This operating sequence causes a modulation of the sampler’s on-resistance during the initial tracking phase, leading to a loss of sampling linearity at higher frequencies.

4.2.2.2 Nonlinear Parasitic Capacitance

Another problem of the conventional bootstrapped switch is that it connects the bulk and source nodes of the charging transistors XCAP_P and XON_P. This connection is designed to prevent forward biasing of the source-bulk junction during the sampling phase when the top-plate voltage of the capacitor exceeds the supply voltage. Also, this connection helps eliminate the body effect. However, doing so loads the top plate of the capacitor with the large nonlinear PN -junction capacitance C_{nwell} of the N-well (indicated by the dashed red rectangle). As a result, C_{nwell} modulates the gate voltage of the sampling switch, which leads to a degradation of the SFDR.

4.2.2.3 Modified Circuit and Generator Design

Several options are included in the generator to address the aforementioned issues. First, a parallel path is implemented to control the transistor XON_N in a way that decouples the activation speed of XON_N from the rise speed of V_G . This action results in an earlier turn-on of XON_N. Consequently, XON_N and XON_P track the input simultaneously in a bootstrapped manner, maintaining a maximum gate-source voltage at the beginning of the tracking phase. Also, the falling transient at V_G is improved because a substantial parasitic capacitance has been removed by disconnecting the gate of some transistors from node V_G . This results in a sharper falling edge and a better-controlled sampling moment. Second, a separate path is added to address the nonlinearity issue caused by the parasitic capacitor. The bootstrap circuit partitions the transistor XCAP_P and C_{boot} into two parts, creating

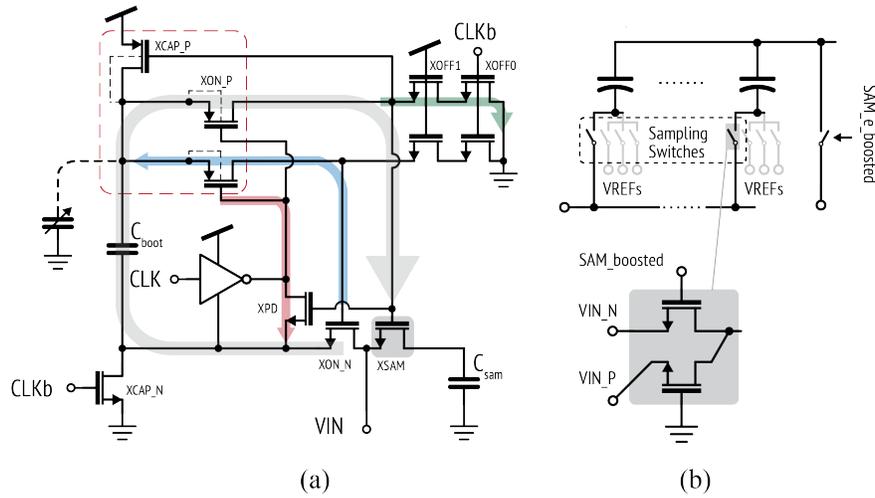


Figure 4.5: Diagrams demonstrating the implementation of (a) the bootstrapped sampler and (b) the sampling switches distributed in the SAR ADC.

the main and auxiliary paths. The input signal propagates to the gate of the sampling switch through the main path without being directly loaded by C_{nwell} , while the auxiliary path drives C_{nwell} . Additionally, when the bootstrapped switch is turned on, the gate of XON_P is mainly connected to the input through XPD instead of the inverter. This is due to the increasing voltage of the inverter's equivalent ground connection. Therefore, the drain of the NMOS in the inverter can be disconnected from the bottom plate of C_{boot} to reduce the nonlinear parasitic R and C [3]. Other techniques, such as pre-charging device [124], are also implemented but not shown in Figure 4.4. Figure 4.4 (c) shows the equivalent circuit of the original design (top) and the modified design (bottom). The parasitic C_{nwell} can be removed from the main path. Also, the parallel driving path removes some parasitic capacitance from the sampler's gate.

4.2.2.4 Implementation of the Sampling Switch in SAR ADCs

Figure 4.5 shows the implementation of the sampling switch (single-ended) integrated with the SAR ADC. Although depicted as a single transistor, the sampling switch highlighted in the gray rectangle represents 2^N switches (assuming a binary encoded CDAC) that are distributed within the CDAC shown in the diagram on the right side. All switches' sources are connected while the drains are tied to the 2^N unit capacitors in the CDAC. Each unit sampler has its corresponding dummy sampler that connects to the opposite signal to cancel the hold-mode feedthrough due to the source-drain coupling of the sampling switch.

The size of transistor XSAM is a trade-off between its on-resistance and the charge injection from its channel. Assuming t_s is allocated for sampling, the upper-bound of sampler

resistance for an N_{bit} resolution can be expressed as follows:

$$R_{on} \leq \frac{t_s}{(N_{bits} + 1)\ln(2)C_{sam}}, \quad (4.5)$$

where C_{sam} is the sampling capacitor. Also, the harmonics power caused by the sampler resistance decreases as the resistance value decreases. However, a larger sampler injects a more nonlinear charge into the sampling capacitor. Therefore, an optimal sampler size that balances these two effects can be obtained for maximum sampling linearity. In terms of the sizing of the other components within the bootstrap circuits, the critical turn-on path of the sampler (highlighted in gray) includes XON_P and XON_N. These must be large enough to ensure sufficient bandwidth for the turn-on path. XOFF1 and XOFF0, along with the corresponding transistors in the auxiliary path, are required to reset the gate voltage during the hold mode, thereby eliminating any hysteresis effect on the gate. The size of the inverter and XPD, which drive the gate of XON_P (highlighted in red), are relatively smaller. XCAP_N and XCAP_P are necessary for charging C_{boot} . The charging time in a time-interleaved ADC is sufficient, which allows for smaller transistor sizes and a reduction in nonlinear parasitic loading. The auxiliary path (highlighted in blue) activates XON_N, which is only a fraction of the main path, considering that XON_N is much smaller than XSAM.

4.2.2.5 Completed Sampler Generator

The optimized bootstrapped generator, combined with the bottom-plate sampling technique, is used to attain the desired goals of high resolution and speed for the proposed generator. The most straightforward implementation of bottom-plate sampling involves adding an additional NMOS transistor connecting to the top plate of the sampling capacitor. This switch requires low resistance during tracking and minimal leakage in the hold mode. Figure 4.6

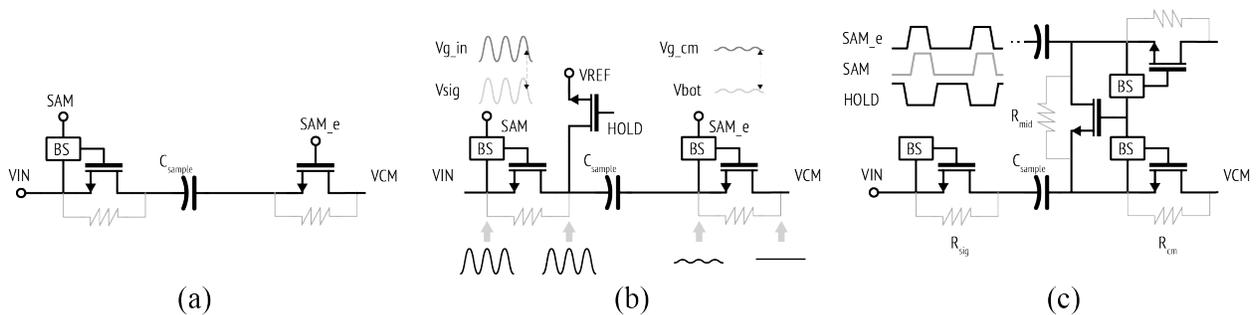


Figure 4.6: Diagrams of (a) the top and bottom-plate sampler with a simple NMOS top-plate switch. (b) the top and bottom-plate sampler and waveform with top-plate bootstrapped. (c) the complete sampling scheme with a middle switch at the top plate.

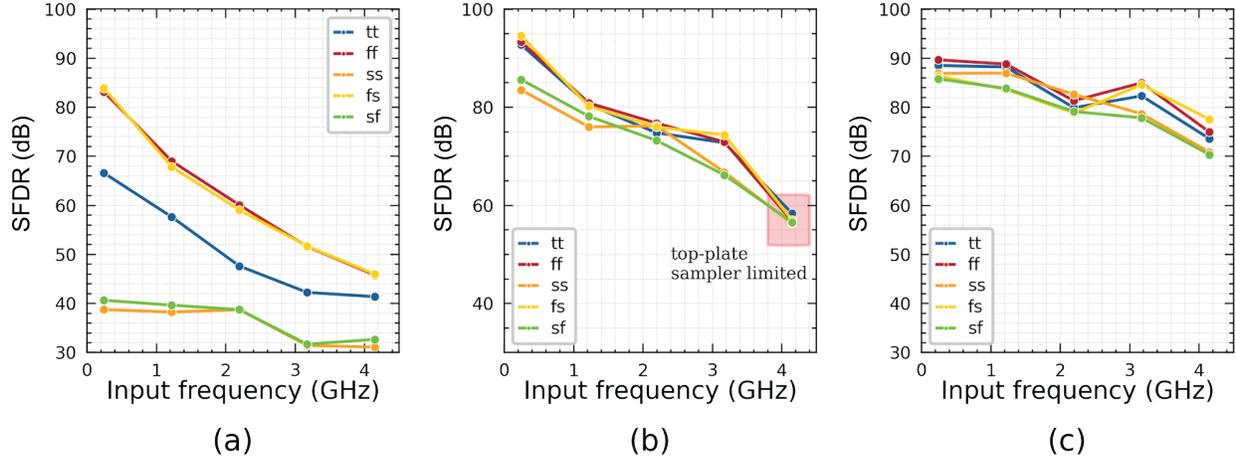


Figure 4.7: The extracted simulation results for different sampling schemes, including (a) a simple NMOS top-plate switch, (b) an NMOS top-plate switch with a boosted clock, and (c) the completed sampling scheme in the generator.

(a) illustrates the first implementation. Bootstrapped switches drive the bottom plate, while the top plate consists simply of an NMOS. However, the poor performance of a single NMOS with the gate connected to V_{DD} degrades its overall performance. The simulation result is shown in Figure 4.7 (a).

To reduce the resistance seen at the top plate of the capacitor, a boosted clock can be used to drive the gate of the NMOS transistor. The simulation data shown in Figure 4.7 (b) is the result of the following setup: a clock drives the gate of the top-plate NMOS transistor with a swing of $V_{DD} + V_{CM}$, causing $V_{GS,NMOS}$ to be equal to V_{DD} . Higher overdrive voltage dramatically reduces the on-resistance and improves the overall linearity of the sampling. When the input frequency rises above 4 GHz, the SFDR of the sampler decreases to less than 60 dB. This is primarily due to the nonlinearity of the top-plate switch. The finite resistance of the top-plate sampler induces a minor swing ripple in the voltage at the capacitor's top plate, which is caused by the input signal. This modulation of the on-resistance occurs even though the gate of the top-plate switch is already boosted. In addition, the boosted gate voltage has the potential to exceed the limit of V_{GS} . To address these concerns, an additional bootstrapped switch is used for the top plate, which helps maintain a consistent V_{GS} . This bootstrapped circuit is interconnected between the gate of the NMOS and the top plate of the capacitor, ensuring a constant voltage across the source and gate of the top-plate NMOS, as shown in Figure 4.6 (b). To further reduce the resistance of the top plate, a common-mode transistor is included to connect the differential circuit. As a result, the common-mode resistance is expressed as

$$R_{cm} = (R_{mid}/2) \parallel R_{on}. \quad (4.6)$$

The signal SAM_e controls the top-plate sampler, while SAM , a delayed version of SAM_e , controls the signal switch. The delay between SAM_e and SAM is a trade-off between having

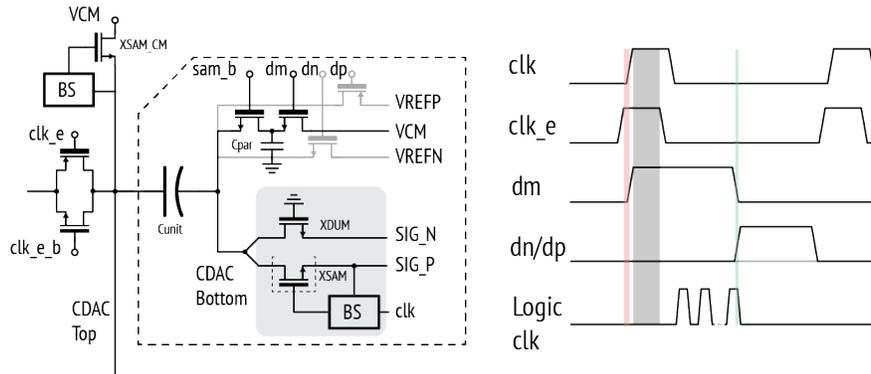


Figure 4.8: Diagram of the unit capacitor and switch, along with its timing.

enough tracking time in a time-interleaving architecture and ensuring a distinct separation between turn-off and the first bit’s flip-over operation in the CDAC.

The simulation results of the circuit, shown in Figure 4.6 (c), are presented in Figure 4.7 (c). The SFDR is augmented to 90 dB at relatively low frequencies, and it remains high at higher frequencies due to the constant V_{GS} provided by the bootstrap circuit. At low frequencies, linearity is limited by the signal-dependent residue charge. For rapid switching, the amount of charge injected into the sampling capacitor during bottom plate switching is determined by the impedance it encounters. This impedance is slightly input dependent because of the nonlinearity of the main sampler. At higher frequencies, the sampling performance is limited by the tracking bandwidth and back-gate modulation. Figure 4.8 shows the design and timing diagram of one unit capacitor with associated switches in the CDAC. A merged capacitor switching scheme is shown here for illustration. It has a pair of switches for sampling and several reference voltage switches. A stacked transistor is added on top of the common-mode voltage reference switch. It helps isolate the reference switch from the CDAC bottom plate. The benefits of this design include the simplification of logic design and a reduction in the first-bit flipping time in the bottom-plate sampling scheme, which accelerates the entire ADC operation.

4.2.3 Hold-Mode Feedthrough

A remaining concern in the sampling network is the hold-mode signal feedthrough from the sampler switches, which manifests as both common-mode and differential-mode ripples at the top plate of the CDAC. This differential-mode coupling from the signal is especially problematic because it could potentially disrupt comparator operations and lead to inaccurate decisions. The signal feedthrough mainly comes from the source-drain coupling of the sampling switches. To counteract this, a dummy transistor is paired with the sampling transistor (Figure 4.9). While this successfully cancels out the source-drain coupling, the signal coupled from the gate still exists. This necessitates careful gate resistance matching to cancel the coupling from the gate-source capacitance (C_{GS}). The resistance mismatch

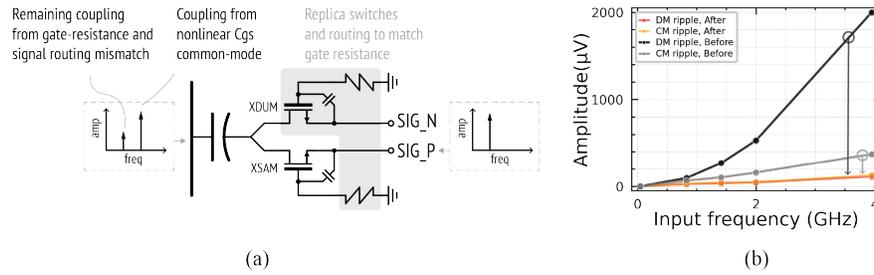


Figure 4.9: (a) Illustration of the signal feedthrough and (b) the amplitude of common-mode (CM) and differential-mode (DM) ripples before and after enabling the gate matching option.

becomes significant if the sampling switches are pulled to ground through XOFFs in the bootstrap circuit, while the dummy transistors are grounded through the nearest available ground nodes. This induces signal coupling on the top plate of the capacitor. To address this issue, a replica pull-down path is utilized to control gates of dummy transistors (Figure 4.10 (a)), which significantly suppresses differential mode signal coupling. However, the nonlinearity of C_{GS} also generates a second-order component at the bottom plate of the capacitor (Figure 4.9 (a)). While this component is unavoidable, it does not significantly impact ADC performance due to the differential implementation. Figure 4.9 (b) illustrates the amplitude of signal feedthrough at the top plate of the CDAC before and after enabling the gate resistance matching option in the generator. The remaining differential mode ripples are caused by the differential signal routing and the mismatches in gate resistance due to the limitations of the floorplan. The implementation in the bootstrap circuit generator is demonstrated in Figure 4.10 (a). Figure 4.10 (b) illustrates the internal node states during sampling. With the gate-matching path, the V_G and V_{off} nodes become differential, effectively canceling out their coupling to the capacitor DAC. Moreover, the waveform in Figure 4.10 (c) illustrates an example of the remaining second-order coupling.

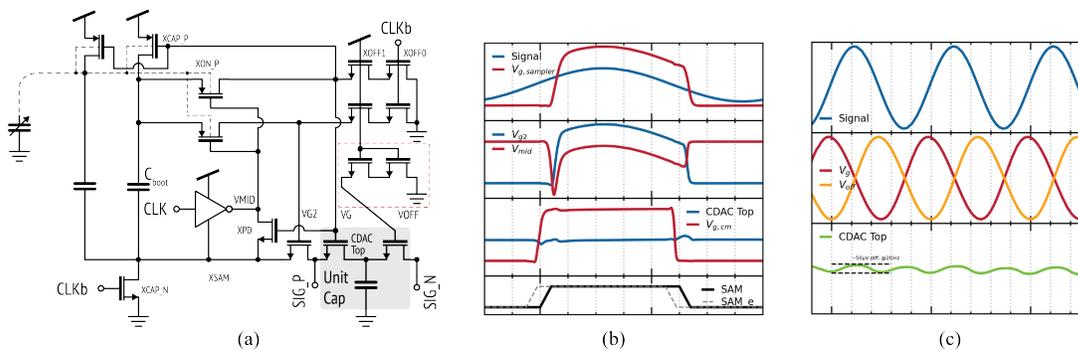


Figure 4.10: (a) Diagram of the bootstrapped circuit, (b) example waveforms of critical nodes in the bootstrap circuit, (c) and example waveforms on the top plate of CDAC.

4.2.4 Generated Instance and Simulation

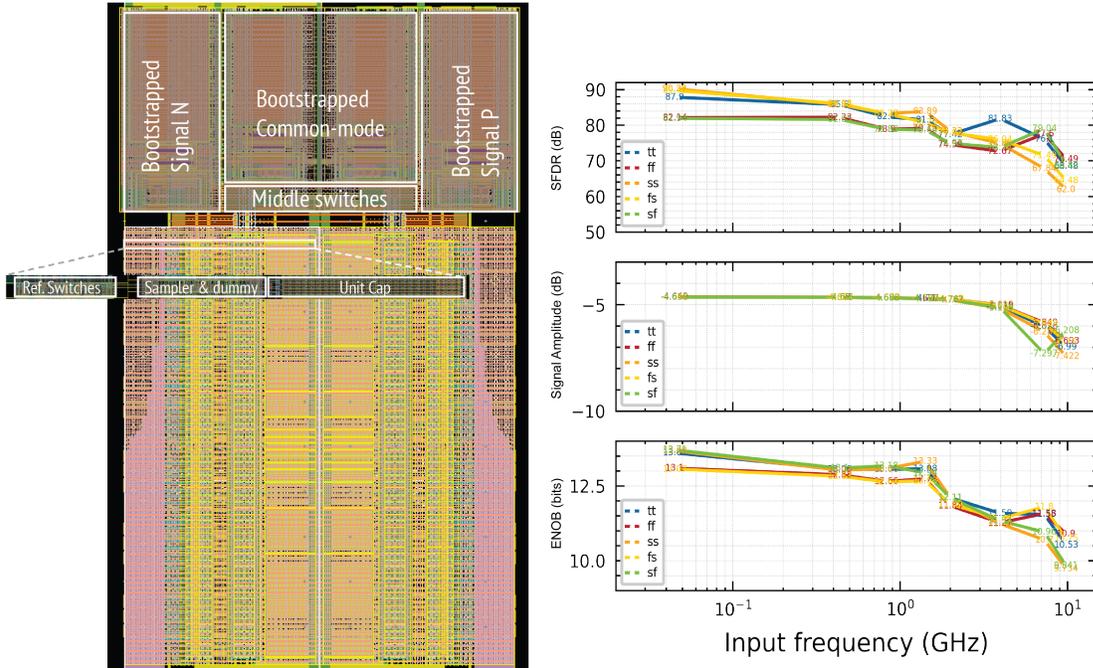


Figure 4.11: Extracted simulation results for the final design of the full sampler.

Figure 4.11 shows an instance generated from the complete sampling circuit generator. Bootstrap circuits are placed on the top, while sampling switches are distributed inside each capacitor unit. The simulation result is also shown in Figure 4.11. The sampler is evaluated in differential mode with a signal amplitude of $V_{amp} = 0.4V$ and a supply voltage of $V_{DD} = 1V$. The delay between SAM and SAM_e is approximately 20 ps. The sampling switch samples at a frequency of 500 MHz with a $\frac{1}{8}$ duty cycle clock used as the SAM signal. The bottom-plate sampler needs to have a sufficiently large size in order to reduce resistance. Meanwhile, the resistance of bottom-plate samplers is much more linear compared to the signal samplers', thereby experiencing smaller voltage variations at each node. Therefore, the resistance of common-mode samplers is intentionally increased to be comparable with the signal switches to suppress the nonlinearity caused by the signal sampler.

4.3 SAR ADC Generator

The SAR ADC has been widely used for medium-to-low resolution and speed applications. It uses a successive approximation (SA) algorithm to search for the unknown input voltage in a binary manner. A typical SAR ADC is comprised of four components: a CDAC, a comparator, SAR logic, and a clock generator. The SA algorithm can be implemented synchronously or asynchronously [16]. The synchronous implementation relies on a high-speed internal clock and takes into account the worst-case bit comparison and clock jitter to achieve the fastest conversion speed. In contrast, the asynchronous operation generates an internal ‘DONE’ signal once the comparator has resolved the result. This creates an oscillation loop that prompts the comparator to initiate the subsequent comparison and then reset. This loop delay determines the conversion speed of the SAR ADC. In this scenario, only one clock with a lower speed is needed for sampling, thus conserving the power that would be consumed by a high-speed clock generator. Furthermore, the asynchronous operation does not wait for the worst-case comparison, and the soft margin during the conversion phase accommodates potential metastable events. The proposed SAR ADC generator supports both clock generators to meet various design needs.

While a conventional SAR ADC employs a single comparator, alternative comparator architectures [17] and loop-unrolled topologies [125] have been proposed to capitalize on the reset phase of the comparator. In these structures, the comparator’s ‘DONE’ signal does not trigger the clock connected to the same comparator. Instead, it triggers the clock for a different comparator while the previous comparator is either reset (in the alternate comparator architecture) or kept static (in the loop-unrolled architecture). Unfortunately, this work does not support multiple comparators topologies due to system considerations in a two-stage hybrid architecture. This SAR generator supports both top-plate and bottom-plate sampling schemes to accommodate the requirements of different resolutions and system requirements. The diagram in Figure 4.12 depicts an asynchronous implementation with a generalized diagram that accommodates different CDAC choices and sampler choices. The paths activated in the bottom-plate sampling scheme are shown in red while the top-plate sampling scheme is shown in blue. Bootstrapped switches are used for critical blocks. Each critical block is discussed in detail in the following subsections.

4.3.1 Comparator Generator

4.3.1.1 Review of Dynamic Comparator Topologies

The comparator plays a crucial role in ADC design. Its performance in terms of speed, noise, offset, and power consumption directly impacts the overall ADC performance. It is uncommon to find comparators used in ADCs implemented as a series of high-gain amplifiers nowadays. While specific implementations may vary, most comparators used in ADCs typically incorporate dynamic amplification and positive feedback in a clocked circuit, thereby consuming zero static power. In such comparators, the preamplification stage supplies cur-

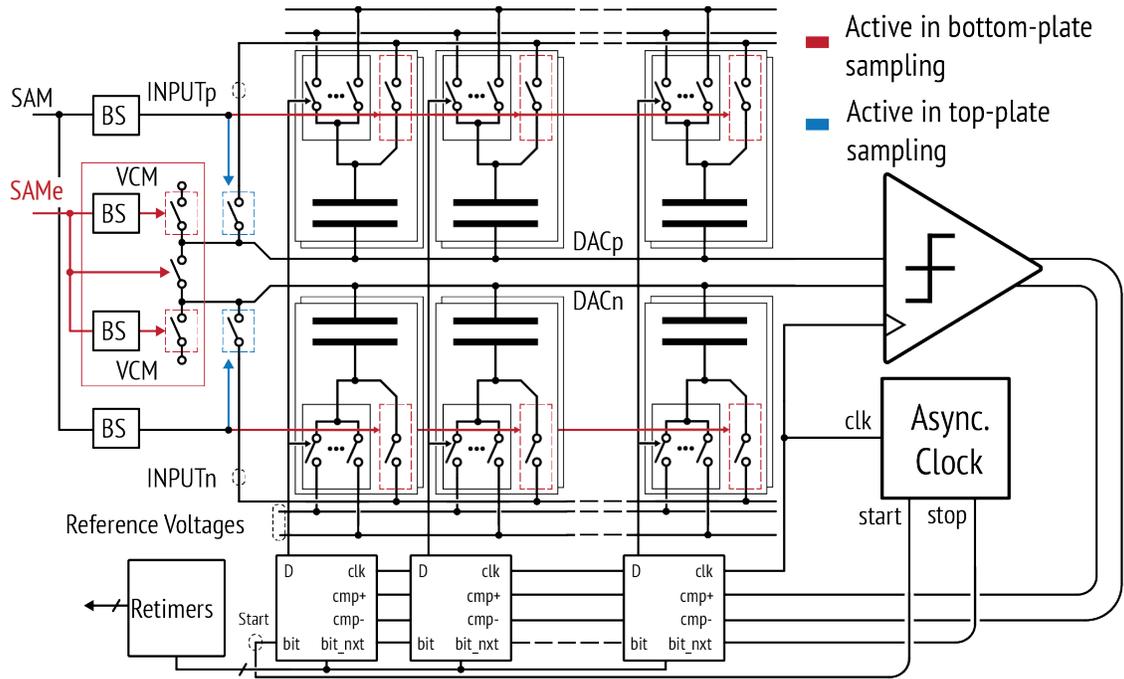


Figure 4.12: Diagram of a SAR ADC with asynchronous clocking and different sampling options.

rent to a regeneration stage consisting of cross-coupled inverters, which produce a digital output with a rail-to-rail swing.

The widely adopted strong-arm latch comparator [129] offers several advantages over static comparators in ADC design. Figure 4.13 (a) shows the schematic of the strong-arm latch comparator. The preamplification stage consists of a series-connected differential input pair and a regeneration latch. A brief description of its operation is as follows. Like other dynamic circuits, the strong-arm latch comparator in Figure 4.13 (a) first goes through a reset phase, where all the internal nodes are charged to a well-defined voltage (supply voltage in Figure 4.13 (a)). When the clock signal triggers, the tail transistor turns on, allowing current I_D to flow through the comparator. The common-mode current, which is half of the drain current ($\frac{1}{2}I_D$), discharges the drains of the input transistors from V_{DD} . Once the voltage drops to $V_{DD} - V_{TH,n}$, where $V_{TH,n}$ is the threshold of NMOS transistors, the NMOS transistors in the cross-coupled inverters turn on, initiating the next phase of operation. Any voltage difference between the differential inputs results in a voltage difference at the drains of the input pair, which initiates positive feedback. Once the output nodes are discharged to $V_{DD} - |V_{TH,p}|$, where $V_{TH,p}$ is the threshold of PMOS transistors, the PMOS transistors in the cross-coupled inverters turn on, thereby increasing the strength of positive feedback and amplifying a small input difference to a full-swing digital output.

Figure 4.13 illustrates some variants of the strong-arm latch comparator, including the optional bridge transistor across the differential sides when the latch is active. Additionally, a complementary version that utilizes PMOS as input transistors is included in the generator to accommodate different common-mode voltages. The primary drawbacks of the strong-arm latch and other single-stage dynamic comparators arise from the stack of three transistors, which restricts the headroom in low-supply designs. They are also susceptible to kickback noise due to the direct coupling between the input and output nodes. The preamplification phase of the strong-arm latch comparator is defined by the bias current and the threshold voltages. The bias current depends on the input common-mode voltage, making the comparator’s performance highly dependent on the input common-mode voltage. The I_D selection is also a design trade-off. A large g_m/I_D ratio is desired to achieve a larger gain in the preamplification phase, while a large tail current is preferred during the regeneration phase.

The double-tail comparators [127, 130, 128, 131, 132, 133, 134] are proposed to address the issues in single-stage comparators while maintaining their dynamic properties. The double-tail comparator, proposed in [127] is depicted in Figure 4.13 (c). It consists of separate preamplification and regeneration stages connected in a shunt configuration. Similar to the strong-arm latch comparator, the preamplifier performs integration on the intermediate nodes. The output of the first stage creates an imbalance in the static latch. Initially, the input transistors in the second stage have greater strength. As the output of the first stage ramps down, the cross-coupled pair initiates regeneration and amplifies the small difference into a full-swing signal. By separating preamplification and regeneration into two stages, designers have more freedom to optimize the gain of the preamplifier and the speed of re-

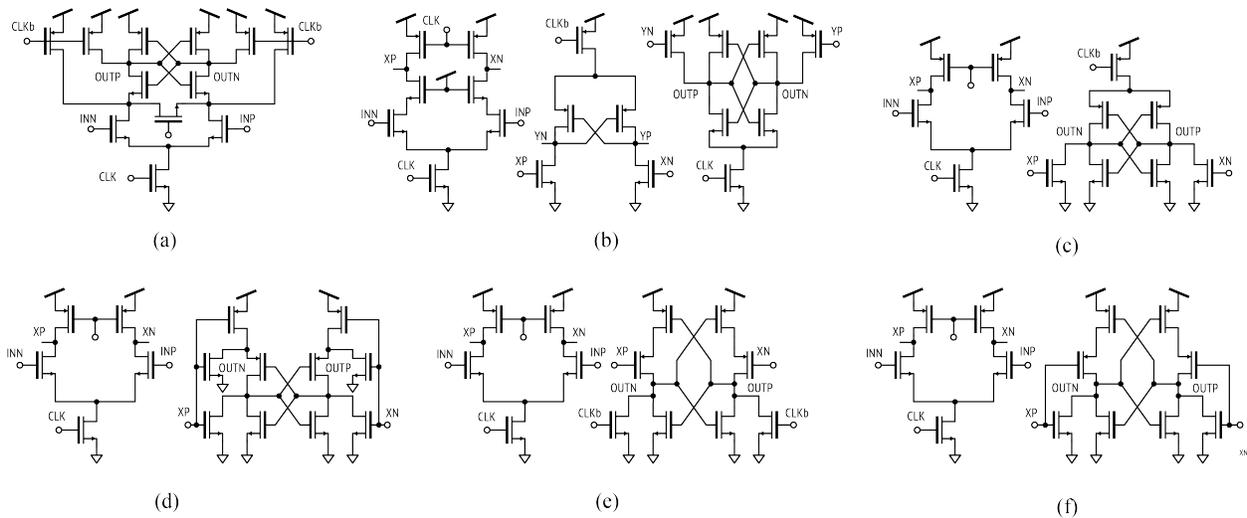


Figure 4.13: Schematics of all supported comparator topologies: (a) strongarm, (b) triple-tail [126], (c) double-tail [127], (d) self-timed double-tail, (e) modified double-tail [128], (f) self-timed modified double-tail.

generation separately. The preamplification stage naturally provides one layer of shielding between the input and output, resulting in less kickback noise. Overall, double-tail comparators offer faster regeneration speed and a wider common-mode range because the input pair does not degenerate the cross-coupled NMOS transistors.

Despite the advantages over single-stage comparators, the second stage of the double-tail comparator in Figure 4.13 (c) exhibits suboptimal performance. The input transistors in the second stage have very limited g_m throughout the entire operation of the second stage's operation [135, 130]. This is because during the propagation phase, which starts from the second stage's input transistors' activation and stops when the regeneration begins, the input transistors of the second stage experience a small drain voltage (V_D), causing them to operate in the deep triode region. Moreover, the gate voltage quickly drops below the threshold voltage ($V_{TH,n}$) shortly after V_D increases. A modification proposed in [130] addresses this issue by replacing the second stage with a strong-arm latch, where both the clock and signal are connected to the first stage output. The schematic is depicted in Figure 4.13 (d). This change enhances the sensitivity of the second stage by softly releasing the reset transistors in the second stage. As a result, critical transistors remain in saturation for a longer time, and the offset and noise of the comparator are defined by the preamplification stage. From an energy efficiency perspective, it is ideal for the regeneration stage to start with a more significant input difference, which also benefits noise and offset performance due to the larger gain provided by the first stage. However, for the amplifier in Figure 4.13 (c), it is not possible to extend the time at which the second stage initiates. The comparator proposed in [128] modifies the stacking orders of transistors. As a result, the input transistors of the second stage do not activate until the common-mode voltage is low enough to turn on PMOS transistors. Furthermore, positive feedback always starts with a strong inversion, resulting in a smaller loop constant, especially in critical near-metastable conditions.

A corresponding version that connects the clock signal (\overline{CLK}) of the second stage to the first stage output (depicted in Figure 4.13 (f)) has been shown to offer similar advantages as depicted in Figure 4.13 (d). In the subsequent discussion, we will refer to these single-phase double-tail comparator topologies as "self-timed." The corresponding self-timed version of the comparator abbreviates the need for an extra clock and avoids the requirement for precise timing, which ensures the correct operation of the second stage. Some other modified versions of two-stage comparators have also been proposed to address more specific issues [131, 132, 133, 134]. However, due to their similarity to the topologies discussed above, they will not be further discussed here. As an example of more stages, the comparator proposed in [126, 136] uses three stages to further increase the gain through a separate signal feedforward path. The topology shown in Figure 4.13 (b) is also implemented as a generator. These triple-tail comparators allow for even further optimizations across various aspects.

4.3.1.2 Considerations for ADC Generator Design

When designing SAR ADC generators, careful consideration must be given to the choice of comparators. Different topologies offer various design trade-offs, including speed, noise,

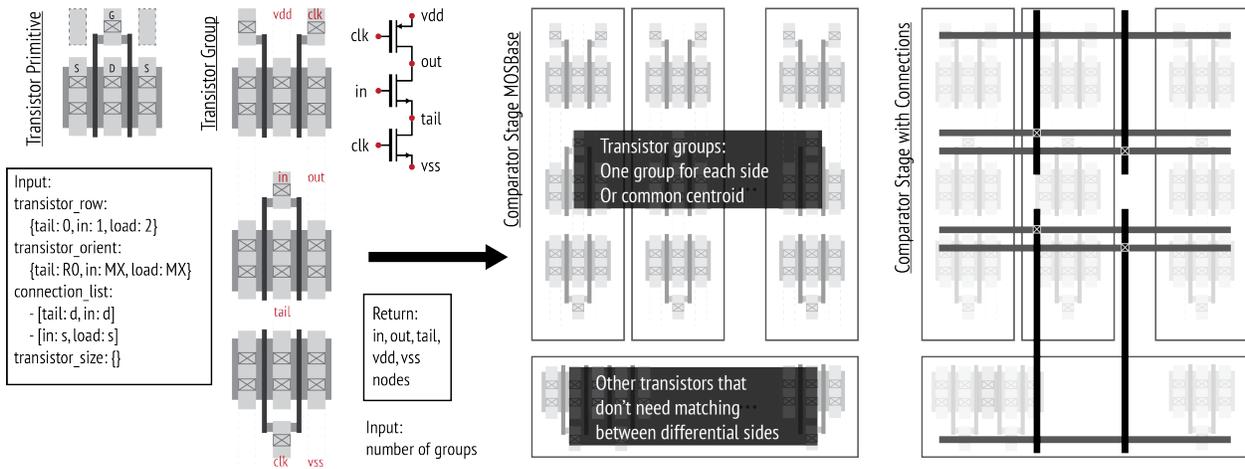


Figure 4.14: Diagrams of transistor grouping and layout hierarchy in comparator generators.

offset, power, common-mode voltage, and process node. Among these topologies, the strong-arm latch comparator stands out as the most compact option, working well with well-controlled common-mode voltage and supply voltage. Intuitively, a dynamic comparator with a separate preamplification stage followed by a regeneration stage is the most efficient choice with more design freedom and higher speed. Self-timed versions of dynamic comparators often exhibit better noise and offset performance due to a higher sensitivity of the second stage. While adding more stages can increase the speed, adding any more stages or branches than the double-tail comparators worsens the noise performance to some extent. To accommodate various performance requirements, the implementation of a SAR ADC generator necessitates selecting different comparator options. It is necessary to include complementary versions of each block, such as the preamplification stage and regeneration stage, along with their variants, to cover the entire range of common-mode voltage selections. Additionally, since some capacitor switching schemes do not provide a constant input common-mode voltage, additional verifications may be required to ensure performance at each bit step.

4.3.1.3 Design of Comparator Generators

The discussion above highlights that all stages in different comparators can be categorized into a dynamic amplification stage and a regeneration stage, with some modifications to the stacking orders of transistors and gate signals. The strong-arm latch comparator stands out due to its unique connections between the two stages. To cover different versions of comparators without creating a separate generator for each one, universal stages are created. These stages consist of smaller groups of unit transistors acting as a generator. Separate wrapper classes for different comparators utilize these universal stages and combine them to create a complete comparator generator. In the layout optimization of a SAR ADC

generator, it is common to integrate the comparator buffer, and clock generators near the comparator to minimize parasitic effects. To accomplish this, an integration wrapper class is employed.

The code example of building a double-tail comparator generator is provided in Appendix B. The construction of comparator layout generators is explained as follows. Figure 4.14 illustrates the smallest unit generator in the comparator generator library. The upper left corner of the figure depicts a conceptual diagram of transistor primitives with poly and lower metal connections. The order of transistors can be configured to encompass all possible combinations found in various latch and preamplifier designs. Adjacent transistors within the same branch can connect their source and drain, controlled by the `connection_list` input parameter. The orientation of transistors is determined by the `transistor_orient` input. Transistor grouping is implemented using the `SingleEndTxGroup` class, which allows access to all intermediate nodes. The transistor group in Figure 4.14 illustrates an example with input parameters displayed on the left side, and the connected metal is depicted with dashed gray lines. All the ports can be accessed in a higher-level generator. Moving up one level in hierarchy, comparator stage generators such as `SingleEndTxGroup`, `DynLatchMatch`, `HalfLatchMatch`, and `StrongArmMatch` inherit from `SingleEndTxGroup` and collect groups of `SingleEndTxGroup` for differential transistors. Shared transistors, such as the tail transistors, are placed in a single chunk and center-aligned with the groups. The simplest implementation of differential transistors allocates one group for the positive and negative sides, resulting in the most compact layout. For more stringent matching requirements, a common-centroid layout can be achieved by dividing transistors into more groups and defining their arrangement. By default, the grouping is one-dimensional, but higher matching accuracy can be attained by splitting transistors into multiple rows within the transistor groups. The final step involves collecting all the ports within the groups of transistors and establishing the necessary connections. Figure 4.14 presents a conceptual diagram illustrating the steps from a transistor to a connected comparator stage. Notably, no substrate connections are made within the comparator stage generator. Tap connections can be shared when multiple stages are included in the completed comparator. This step is included in the wrapper class for each topology. Appendix B provides example code for the double-tail comparator generator.

Figure 4.15 depicts the integration steps for a complete comparator layout, starting from the comparator stages. The generator takes parameters for each stage and other configuration options, such as complementary input. The schematic template for each stage is implemented with NMOS input, and the schematic generator handles the complementary implementation based on the input of the first stage. The flow chart in Figure 4.15 (a) outlines the steps involved in completing the comparator layout. In the case of multi-stage comparators, substrate connections are shared among stages. The completed comparator layout, as depicted in Figure 4.15, illustrates the floorplan of the layout. To ensure that each stage has the same width, dummy structures are used to fill the gaps. Depending on the overall width requirement of the SAR ADC, empty spaces can be added to the sides. These steps are implemented in the comparator wrapper class, allowing the comparator to function

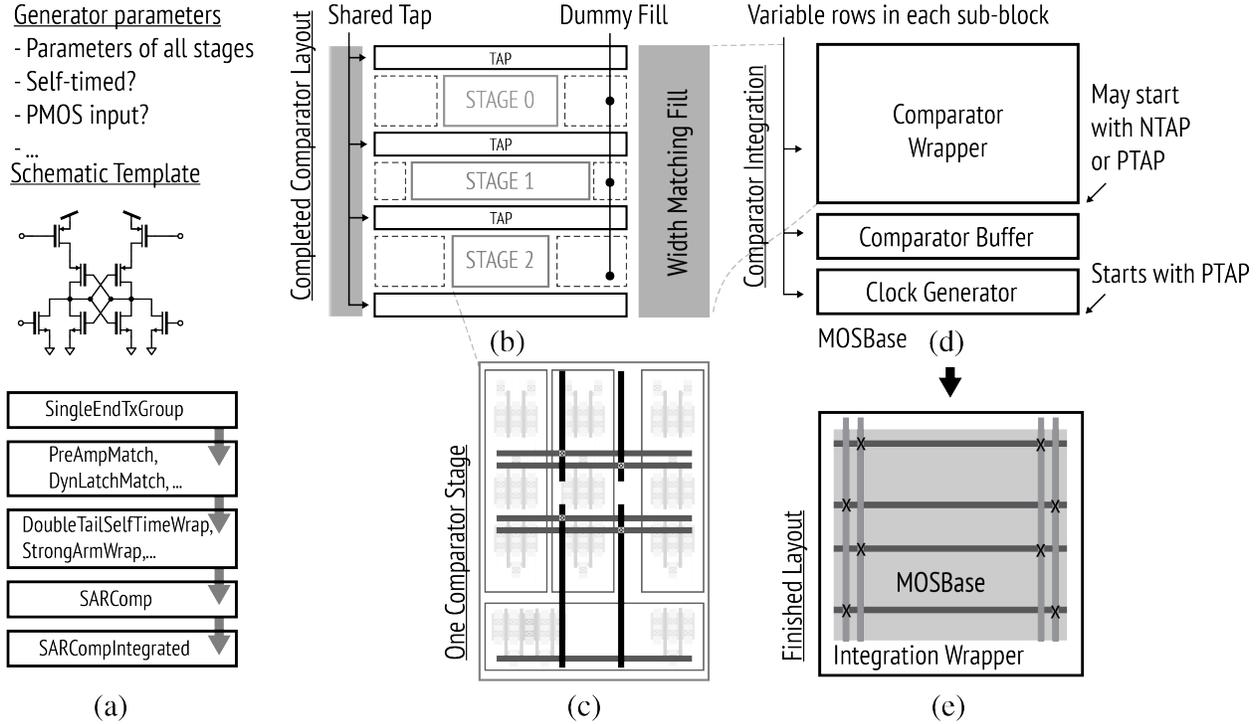


Figure 4.15: Illustrations of the integration steps of the comparator generator: (a) flow chart of the integration steps, (b) floorplan of the complete comparator generator, (c) floorplan of one comparator stage, (d) integration of the comparator, buffers, and the clock generator, and (e) power strapping.

as a standalone block. In the context of a SAR ADC, when utilizing the comparator generator, a separate `SARComp` class is implemented to encapsulate the comparator, as well as the necessary comparator buffers. In the case of an asynchronous SAR ADC, placing the clock generator near the comparator reduces parasitics and allows for higher clock speeds. The `SARComp` class instantiates an `MOSBase` class that starts from the clock generator and then proceeds to the buffers. The comparator is then placed at the top. All connections between blocks are also made at this level. A general `IntegrationWrapper` class completes the last step, exporting all supply tracks and bringing them to the specified top metal layer. With generators available for various comparator topologies, the SAR ADC generator can choose different topologies for different scenarios. Additionally, a scripted testbench can quickly evaluate the generated instance and assist designers in selecting the appropriate comparator topology. Simulation results from built-in testbenches within the generator exemplify the process of selecting the comparator in ADC design using generators. While the designer is responsible for making specific topologies and sizing decisions instead of relying on an automatic design script, the generator approach facilitates the design procedure and enables the

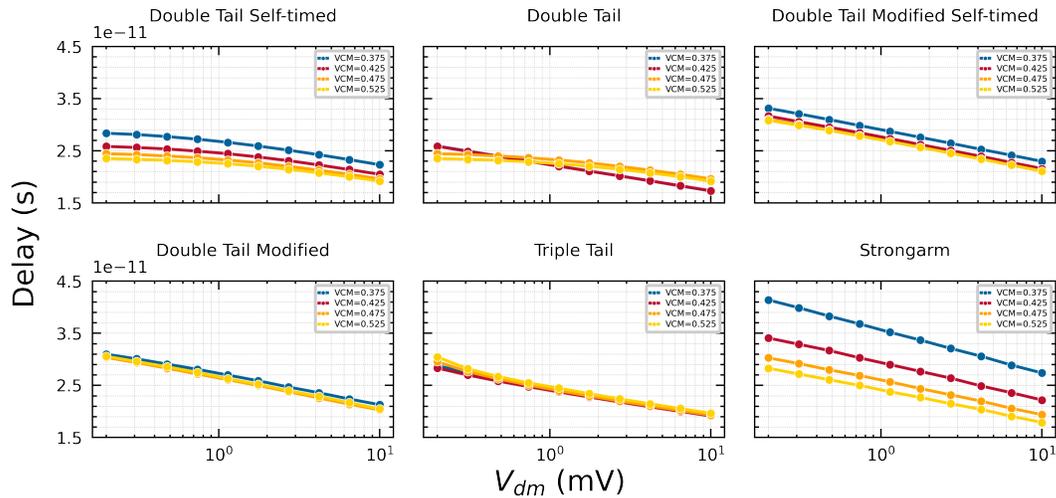


Figure 4.16: Comparison of speeds for different comparators at various common-mode voltages.

exploration and assessment of various design choices. For example, Figure 4.16 displays the speed of different comparator topologies with different input amplitudes and common-mode voltages. The strong-arm latch comparator exhibits significant delay variation at different common-mode voltages, with a nearly twofold increase in delay at $V_{cm} = 375$ mV compared to $V_{cm} = 525$ mV. The other topologies perform similarly under the evaluated conditions, although the modified version of the double-tail comparator is slightly slower due to a longer amplification time.

Figure 4.17 demonstrates the trade-off between noise and speed. Among the double-tail comparators, self-timed versions offer better noise performance at the expense of slightly lower speed at low common-mode voltages. The additional stage in the triple-tail comparator degrades noise performance, although it offers the fastest design for all common-mode voltages. Monte Carlo simulations of the input-referred offset voltage at different supply voltages are shown in Figure 4.18. The comparators exhibit a similar trade-off between offset and noise performance.

In conclusion, the built-in testbenches and generators for different topologies enable informed design choices for SAR ADCs. While specific topologies and sizing decisions require input from the designer, the generator-based approach simplifies the design process and facilitates the exploration and evaluation of various design choices.

4.3.2 Capacitor DAC Design

The CDAC performs two functions in the SAR ADC: it is used as the sampling capacitor and it executes binary search algorithms. In each conversion step, a fraction of the reference voltage is subtracted from the top plate of the CDAC, which decreases the voltage difference

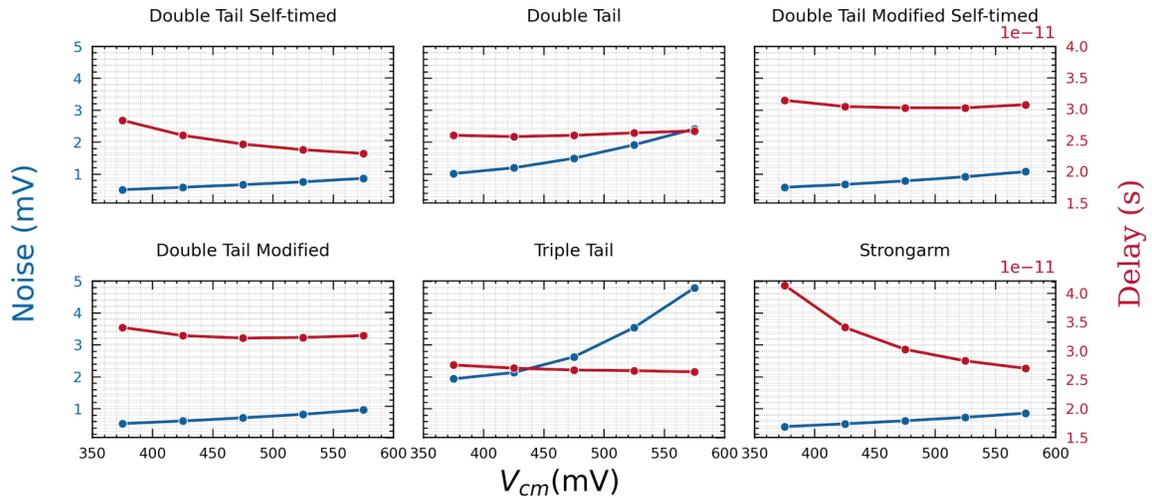


Figure 4.17: Trade-off between speed and noise for different comparators.

between the CDACs in a differential implementation. The capacitance of the CDAC in SAR ADCs directly affects ADCs' power consumption and resolution. Although power consumption in each conversion is highly dependent on the input voltage and the switching method, the power consumed by the CDAC is generally proportional to $C_{DAC}V_{REF}^2$, where C_{DAC} is the total capacitance of the DAC and V_{REF} is the maximum difference between positive and negative reference voltages. Considering thermal noise, the resolution requirement of the ADC sets a lower limit for CDAC capacitances. The thermal noise associated with the sampled signal is $k_B T / C_{DAC}$, where k_B denotes Boltzmann's constant and T represents absolute temperature. Increasing the SNR by an additional 6 dB (equivalent to a 1-bit increase in resolution) requires quadrupling the DAC's capacitance. Numerous CDAC implementations exist, including a wide variety of sampling methods, switching schemes, bit weights, and reference voltage counts. Each variant entails its own unique set of design trade-offs, which substantially influence the overall performance of the ADC. Therefore, designing a CDAC within the scope of a SAR ADC requires a wide range of options to achieve all the performance targets and usage cases that the ADC aims to support. In this context, this subsection discusses the sampling method, the DAC switching scheme, and the concept of redundancy in the ADC design.

4.3.2.1 Redundancy in SAR ADCs

Redundancy techniques have been widely used in pipelined ADC designs [137], as they assist in mitigating errors in pipeline stages, such as non-linearity, flash ADC errors, and comparator offsets. In pipelined ADC design, which incorporates redundancy, the aggregated resolution of the individual stages exceeds the target ADC resolution. This allows for the tolerance of minor errors in the sub-ADC stages. In SAR ADC design, the increasing

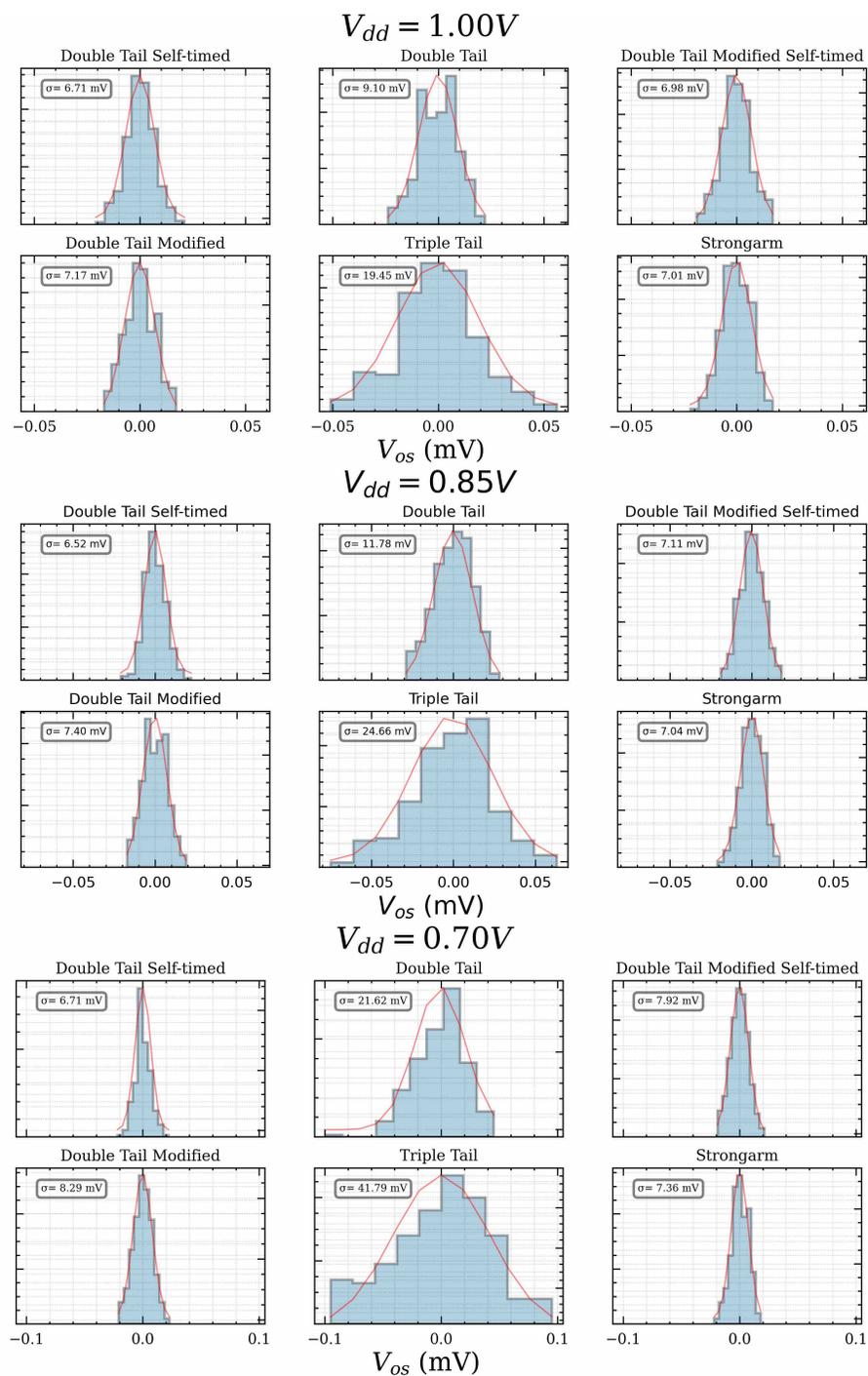


Figure 4.18: Offset voltages of six comparator topologies using different supply voltages.

Algorithm 1 Successive Approximation Algorithm

Require: $V_{in} \in [V_{\text{ref},N}, V_{\text{ref},P}]$

- 1: $V_{\text{max}} = (V_{\text{ref},P} - V_{\text{ref},N})/2$
- 2: $V_{\text{ref}}[0] = V_{\text{max}}/2$
- 3: **for** $k = 0 \rightarrow N - 1$ **do**
- 4: **if** $V_{in} > V_{\text{ref}}[k]$ **then**
- 5: $V_{\text{ref}}[k + 1] = V_{\text{ref}}[k] + V_{\text{max}}/2^k$
- 6: $D_{\text{out}}[k] = 1$
- 7: **else**
- 8: $V_{\text{ref}}[k + 1] = V_{\text{ref}}[k] - V_{\text{max}}/2^k$
- 9: $D_{\text{out}}[k] = 0$

requirement for higher conversion speed reduces the available settling time for the CDAC, thus necessitating redundancy in CDACs in the proposed generator. To analyze the redundancy techniques and incorporate them into the proposed generator, the algorithm shown in Algorithm 1 demonstrates the process of binary search. The final result is calculated as follows:

$$\hat{V} = \sum_{i=0}^{N-1} w[i] D_{\text{out}}[i], \quad (4.7)$$

where N is the number of bits, $D_{\text{out}}[i]$ is the i -th bit at the output and $w[i]$ denotes the weight of the i -th bit in the CDAC. Errors can happen at the line 4,5, and 8 of the Algorithm 1

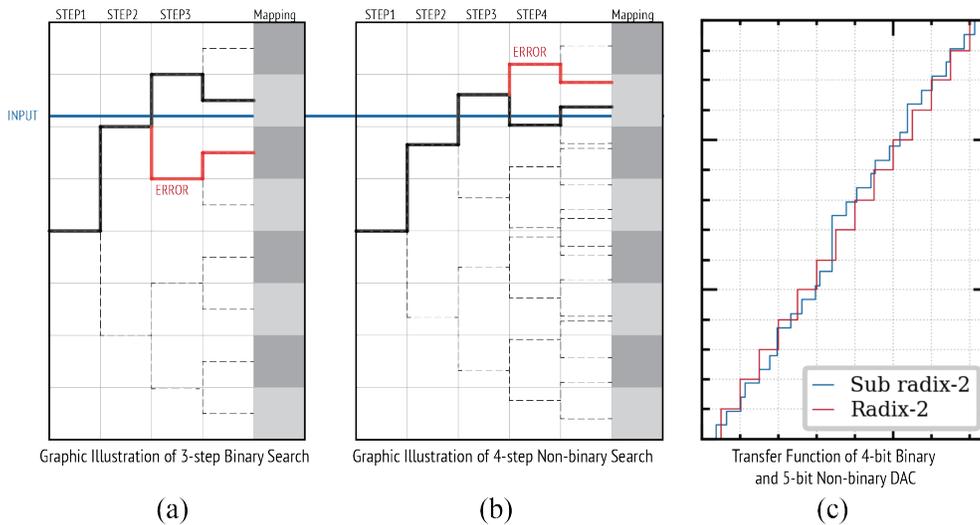


Figure 4.19: (a) Search algorithm of a 3-bit DAC with three steps (b) Search algorithm of a 3-bit DAC with four steps (c) Transfer curve for a 4-bit DAC.

Conceptually, redundancy in SAR ADCs can follow a methodology similar to pipelined ADCs. For instance, a 1.5-bit/step SAR ADC illustrated in [138] incorporates additional levels in the bit comparison. Nonetheless, it is more common for each bit conversion step to use a 1-bit comparator [139, 140, 141], with redundancy embedded either in DAC weights or conversion steps. The most apparent error that redundancy can alleviate in the SA algorithm 1 is the inequality evaluation in line 4. Including redundancy in the DAC makes it possible to correct the error when the comparator makes a mistake. Moreover, potential inaccuracies due to incomplete settling in the equations in lines 5 and 8 can be addressed by incorporating redundancy. Exploiting the incomplete settling with the help of DAC redundancy can enhance the ADC’s speed. Various methods of redundancy implementation have been reported.

The first category of redundant CDACs adopts non-binary weights with a radix less than 2. The final result of the conversion can be generalized to

$$\hat{V} = \alpha \cdot \sum_{i=0}^{N-1} w[i] D_{\text{out}}[i], \tag{4.8}$$

where α scales the result to the full-scale range, considering that a redundant CDAC usually has an over range. The fundamental mathematical explanation is provided in [142]. In general, a non-binary CDAC necessitates M steps for an N -bit resolution, and each step reduces the full range by a factor of less than 2. Assuming the SAR ADC is properly designed, only 2 out of N comparisons during the SAR operation are likely to be critical [143], including one being the last step. Any decision error that occurs at the step with redundancy still has a chance to be compensated in the subsequent step. However, restricting the radix to less than 2 poses difficulties in implementing a fractional DAC. In practice, measuring the radix is also challenging. The generalized non-binary algorithm only requires sufficient coverage to tolerate errors during quantization. Thus, with M ($M > N$) integer-weighted CDAC, the benefits of redundancy remain. CDAC designs in [24, 143] incorporating additional steps to compensate for potential errors in comparator decision and incomplete settling while maintaining the simplicity of binary encoding. The "redundancy in the k -th step" can be defined as:

$$q[k] = -w[k + 1] + \sum_{i=k+2}^M w[i] + 1. \tag{4.9}$$

Figure 4.19 (a) and (b) illustrate the search algorithm used in a 3-bit DAC. The blue line denotes the input signal, while the red line represents a potential erroneous decision path. The correct decision path in Figure 4.19 (a) is shown in black. An error occurs in step 2 that cannot be corrected in the final step, resulting in an error exceeding $\frac{1}{2}LSB$. In contrast, Figure 4.19 (b) depicts the decision path with a radix of $2^{3/4} = 1.68$. The subsequent steps correct the error that originated from step 3, thereby ensuring that the final result aligns with the correct path. Figure 4.19 (c) shows how redundancy assists in digital error correction. The transfer curve in the radix less than 2 scenario deviates from the ideal transfer function, featuring larger vertical segments indicating the missing output code. Unlike missing input decision levels (in the radix greater than 2 cases) that necessitate analog component calibration, missing codes can be corrected in the digital domain. It is

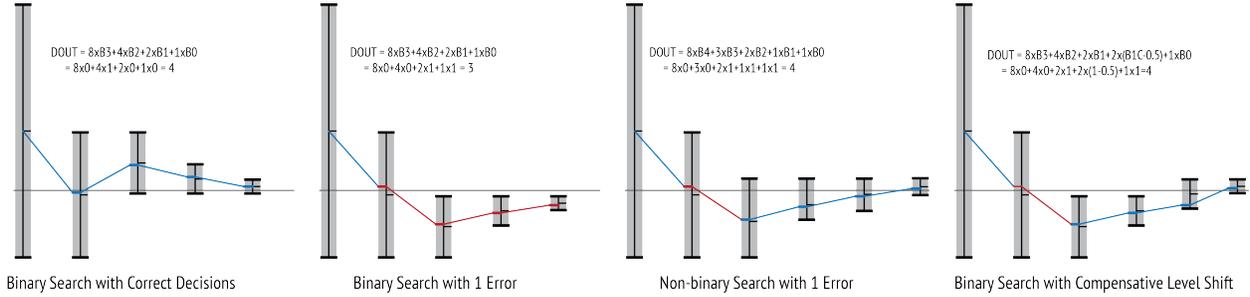


Figure 4.20: Illustrations of the successive approximation searching and digital outputs of various methods.

sufficient that the transfer function can be digitally corrected without any missing codes [144] when

$$C_i < C_0 + \sum_{k=0}^{i-1} C_k = C_0 + \sum_0^{i-1} \alpha^k C_0. \quad (4.10)$$

Figure 4.20 illustrates how incomplete settling is addressed in redundant CDAC designs. The horizontal gray line symbolizes the input signal level, while the vertical line boxes represent the remaining full range at the current bit.

The leftmost plot depicts a correct decision path, with the second decision requiring sufficient time for settling to obtain an accurate comparison result. To settle the first transition within half an LSB, a settling of $1 - 0.5/8 = 93.75\%$ or 2.77τ is required. The second plot depicts an erroneous decision path stemming from incomplete settling, resulting in an outcome that deviates by 1 LSB. The third plot employs a generalized non-binary DAC with $[8, 3, 2, 1, 1]$ weights. An error resulting from incomplete settling after the first decision can be corrected by all subsequent steps. The settling requirement reduces to $1 - 2.5/8 = 68.75\%$, equivalent to 1.16τ , thereby reducing the settling time requirement by over $2\times$ through redundancy. The last plot utilizes a binary DAC with level-shift compensation (an additional step) as proposed in [24]. The same error occurs, but the second-to-last step uses a shifting technique to correct the error.

In summary, integrating redundancy in the CDAC of SAR ADC can significantly reduce the need for settling speed. Rather than resorting to precise analog techniques or digital calibration to correct errors, redundancy techniques render these errors acceptable. Fortunately, even though there are various methods of implementing redundancy, and some of these techniques may use intriguing ways to absorb error, the circuit implementation is relatively straightforward, albeit with the cost of additional steps required for convergence. The ADC generator discussed in this work provides adequate redundancy options by adjusting the weights of capacitors and the number of steps.

4.3.2.2 CDAC Switching Schemes

In addition to speed and precision, the power consumption of the CDAC is also a critical specification. Although the lower limit of C_{DAC} value is often constrained by mismatch and

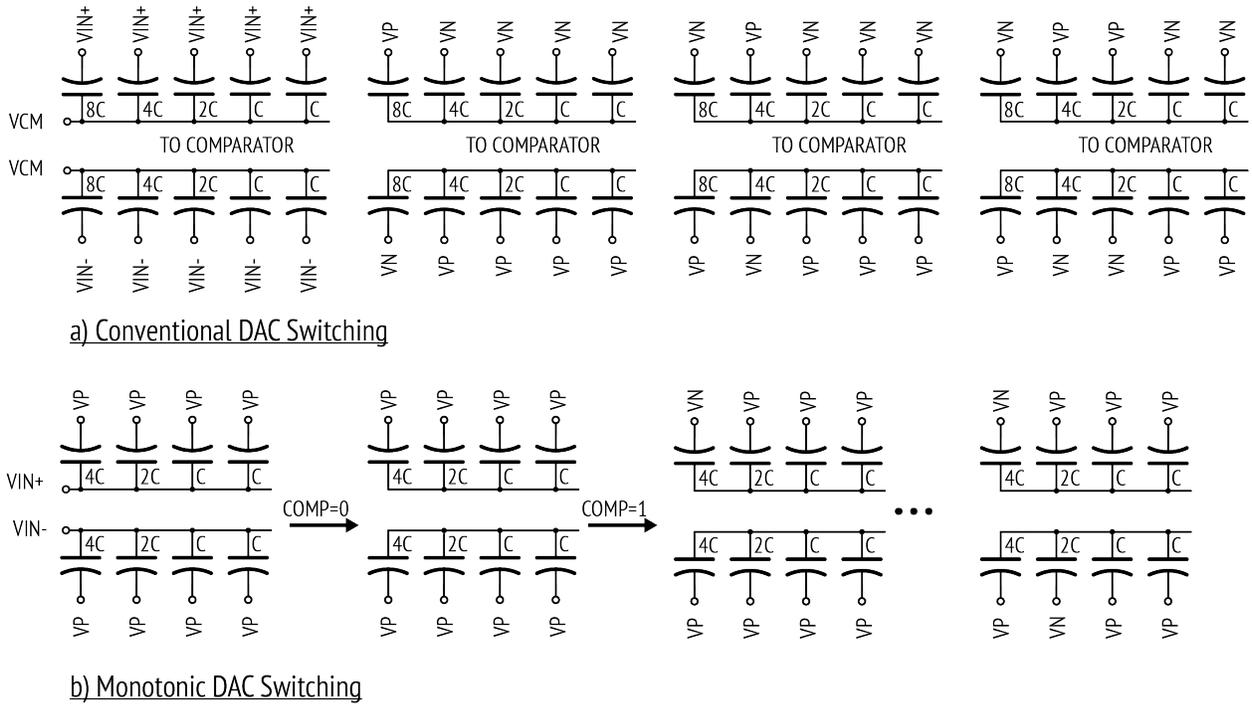
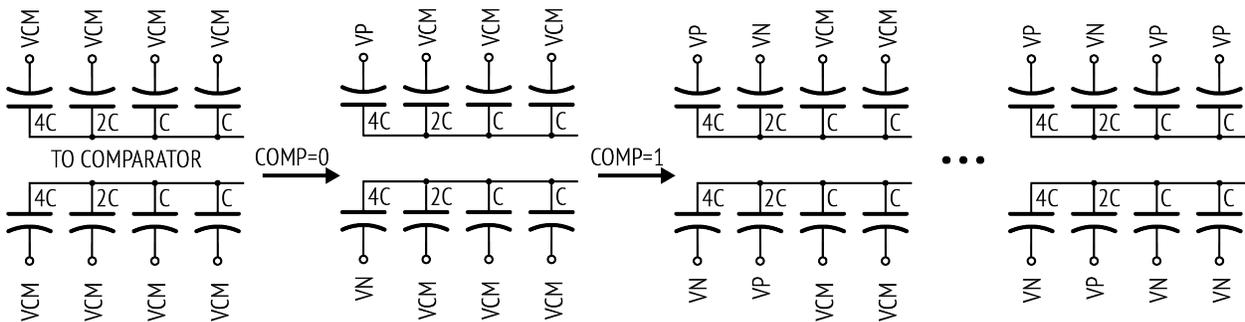


Figure 4.21: (a) The conventional and (b) the monotonic capacitor switching schemes.

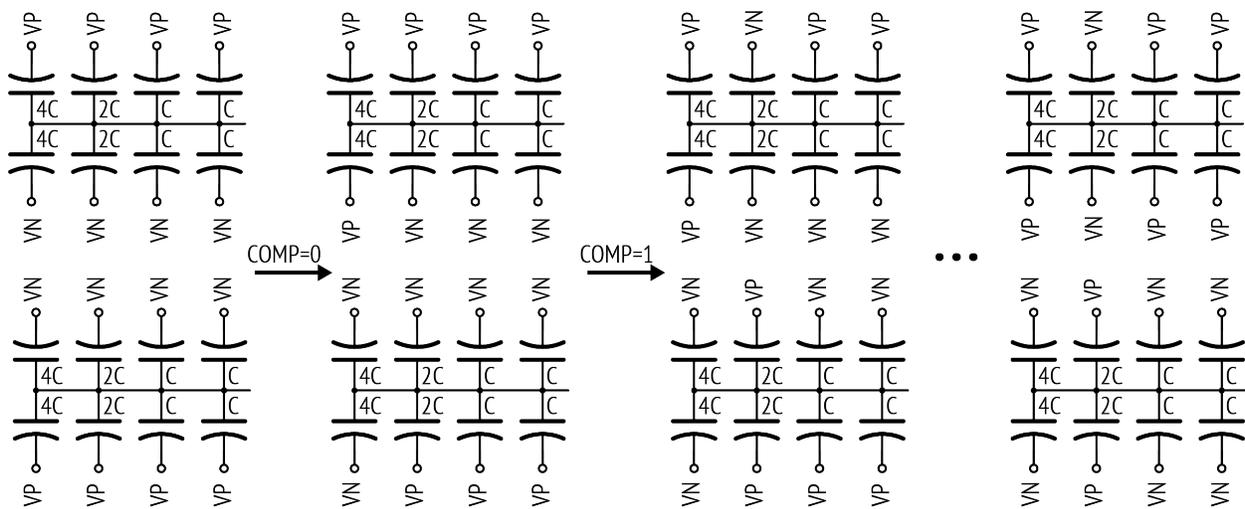
thermal noise requirements, different switching methodologies determine the proportionality constant of the energy consumed by the CDAC. In this section, several different capacitor-switching methods are discussed, comparing their energy efficiency using a Python simulation model of the CDAC.

First, Figure 4.22 (a) shows the conventional CDAC switching scheme. For simplicity, only the single-ended CDAC is described, and the top-plate and bottom-plate sampling schemes are used interchangeably. Despite most of the subsequent energy-efficient switching methods being demonstrated in the top-plate sampling scheme, corresponding bottom-plate versions can be designed with adaptations to the CDAC switches and the SAR logic.

After sampling, all the capacitor units are connected to the negative reference (V_N) while the MSB of the CDAC is switched to the positive reference voltage (V_P). The comparator then performs a comparison. If the output of the comparator is 0, the MSB switches to the negative reference V_N ; otherwise, it remains at the positive reference. This trial-and-error process repeats until all bits are resolved. However, the conventional CDAC exhibits inefficiency while executing this algorithm. Further, the energy consumed in ‘UP’ and ‘DOWN’ conversions presents significant asymmetry. As discussed in [145], the ‘DOWN’ conversion consumes five times more energy than the ‘UP’ conversion. The monotonic switching scheme was proposed to resolve the asymmetrical issue and improve energy efficiency [146]. The single-ended diagram is shown in Figure 4.21 (b). Unlike the conventional switching scheme, monotonic switching uses top-plate sampling with all the capacitor units connected to V_P .



a) Merged Capacitor Switching



b) Split Capacitor Switching

Figure 4.22: (a) The merged capacitor switching and (b) the split capacitor switching schemes.

After sampling, the CDAC with a higher voltage in the differential topology switches to the other reference from the MSB, depending on the output of the comparator. Since the voltages on the CDAC only reduce during the conversion, the common-mode voltage gradually steps down. This one-directional settling is favored when the speeds of the NMOS and PMOS transistors are different in a process. Also, the diminishing common-mode voltage requires the comparator to handle a wider range of the input common-mode voltages. Moreover, A constant common-mode voltage is critical in the design of a pipelined SAR or other pipelined ADC topologies. Therefore, constant common-mode capacitor schemes are required in such cases.

Figure 4.22 shows two alternative designs that have higher energy efficiency than the conventional design while maintaining a constant common-mode voltage. The split capacitor switching scheme is proposed in [147]. The single-ended diagram is shown in Figure 4.22 (a).

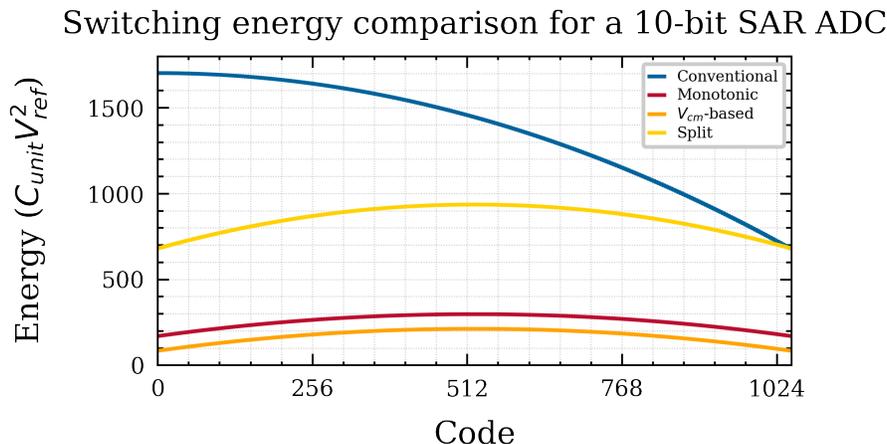


Figure 4.23: Comparison of the switching energy in different schemes in a 10-bit SAR ADC.

The entire CDAC is split into two halves. During sampling, half of the capacitors switch to VP while the remaining capacitors switch to VN . The top-plate voltage is incremented by connecting the negative half capacitor to VP and down-shifted by connecting the positive half capacitor to VN . Despite employing two reference voltages similar to the aforementioned schemes, this capacitor switching scheme maintains a constant common mode at the cost of doubling the number of capacitors.

Another capacitor switching scheme called merged capacitor switching, or V_{cm} -based switching, is proposed in [148], where an additional common-mode reference voltage (V_{cm}) is used (Figure 4.22 (b)). This switching scheme can be seen as merging the two parts of the capacitor in the split capacitor scheme. Combining two capacitors that are connected to VN and VP separately is equivalent to a capacitor connected to V_{cm} . The voltage shifting is simply done by switching the bottom plate from V_{cm} to either VP or VN . Note that accurate V_{cm} is not necessary for the search algorithm. However, if V_{cm} does not equal $(V_N + V_P)/2$ perfectly, the common mode voltage also varies.

To assess the energy efficiency of each capacitor switching scheme, a Python model is used. The model takes the expected DAC output code and CDAC weights and calculates the charge stored at each capacitor in every conversion step. Therefore, the energy is calculated. The result is shown in Figure 4.23, where the average energy of a uniformly spread input signal is compared. The figure shows that the energy consumed by the low analog input is significantly higher than the high input code in the conventional capacitor switching scheme. All three other schemes have symmetrical switching energy. In summary, compared to the conventional switching technique ($E \approx 1363 \cdot C_{DAC} V_{REF}^2$), the split-capacitor scheme achieves a 37.5% ($E \approx 850 \cdot C_{DAC} V_{REF}^2$), the monotonic scheme achieves 81.25% ($E \approx 255.5 \cdot C_{DAC} V_{REF}^2$), and the V_{cm} -based scheme have 87.5% ($E \approx 850 \cdot C_{DAC} V_{REF}^2$). Some other switching schemes have been proposed in the literature. [149, 150, 151, 152, 153] can achieve higher efficiency. Due to the complexity of the circuit in these implementations,

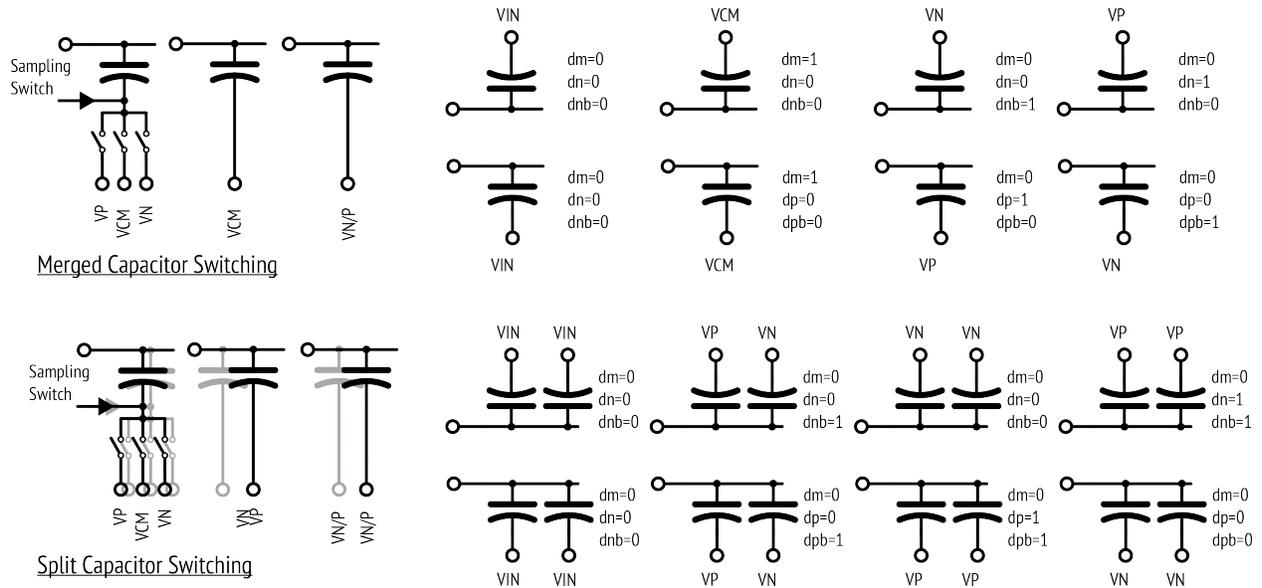


Figure 4.24: Diagrams of the CDAC schematic generator.

the generator does not incorporate these techniques. Also, no fractional reference voltage other than V_{cm} is used in the capacitor switching schemes. Using fractional reference voltages in the last few bits can create an equivalent LSB capacitor unit without further reducing the size of LSB CDAC. This approach can improve the matching of the CDAC. However, generating an accurate fractional reference voltage increases circuit complexity and reduces flexibility in generator implementation.

4.3.2.3 Design of CDAC Generators

Figure 4.24 shows the schematic generator of the V_{cm} -based and split capacitor switching schemes, as well as the corresponding logic signals. The CDACs shown in the figure are designed using the bottom-plate sampling scheme to improve sampling linearity. Figure 4.25 shows the structure of finger-typed MOM capacitors that are typically used in today’s SAR ADC designs [154, 155]. The generator supports a variable number of coupling layers. The capacitance is primarily defined by the lateral field and, therefore, depends on lithography rather than the thickness of the oxide film in metal-insulator-metal (MIM) capacitors. The generator supports different metal widths and lengths. The lower metal layers are favored because of high capacitor density with the trade-off of increased capacitor coupling to the substrate.

The unit capacitor generator takes a horizontal metal layer to construct a finger-shaped coupling structure. The parameters, such as the length of the coupling metal, the width of the terminal, and the number of fingers, are adjustable. The size of the unit capacitor for medium-to-low resolution SAR ADCs is reduced to less than 1 fF in high-speed designs,

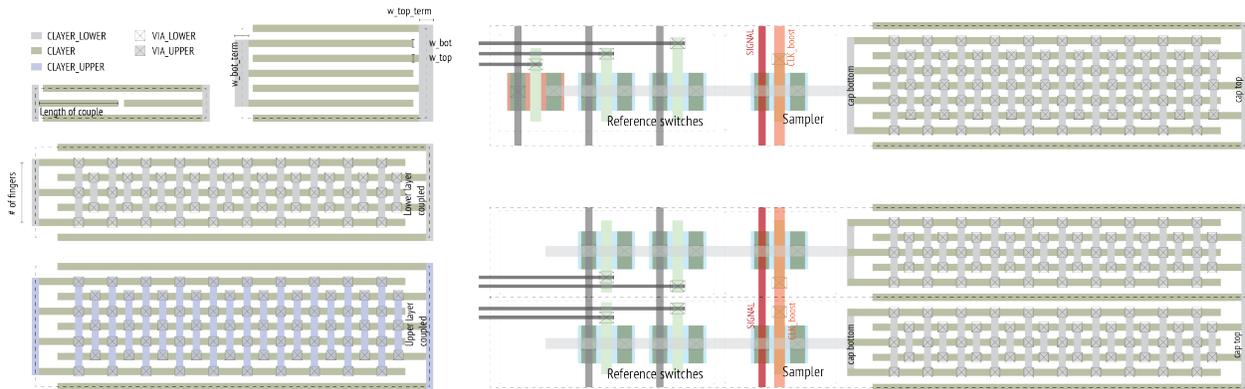


Figure 4.25: Diagrams of the CDAC layout generator.

with unit matching being the main limiting factor. Therefore, when a smaller unit capacitor is needed, a short coupling length with a single finger can be used. Figure 4.26 shows an example of a small LSB capacitor. When a higher capacitance density is required for thermal noise-limited design, additional coupling is achieved by incorporating vertical metal layers and vias. The capacitor can be accessed by a CDAC driver from the left side, while the right side is used for the top-plate connection to the comparator. The left diagram in Figure 4.25 shows how the unit capacitors are integrated with switches for the V_{cm} -based and split capacitor switching schemes, respectively. Besides the parameter for the capacitor unit, more details about the parameters of the CDAC generator are shown in Figure 4.26. The capacitor is symmetrically arranged to counteract the first-order gradient. The grouping of bit capacitors has fewer parasitics to ground, as shown in [122]. Even though the theory behind DAC redundancy is less straightforward, the implementation of redundancy in CDACs is simple. In the proposed generator, the generalized non-binary weights or compensative level-shifting steps can be implemented simply by specifying the DAC weights list. A dummy capacitor is also included in the array to improve matching and symmetry.

4.3.3 SAR Logic Generators

The implementation of SAR ADCs across various processes primarily utilizes the asynchronous architecture due to its aforementioned advantages. However, synchronous operation is implemented in certain processes when the matching of DAC settling speed and logic delay results in a suboptimal design. In either scenario, it is critical to assess the loop delay, which starts at the output of the comparator, traverses through logic gates and the CDAC, and then returns to the input of the comparator. The loop delay can be analyzed from an optimization perspective, as shown in Figure 4.27. In each bit comparison, the time durations for the signal takes to go through a comparator, a SAR logic cell, and settle the CDAC can be denoted as T_{COMP} , T_{LOGIC} , and T_{DAC} , respectively. T_{LOGIC} includes any

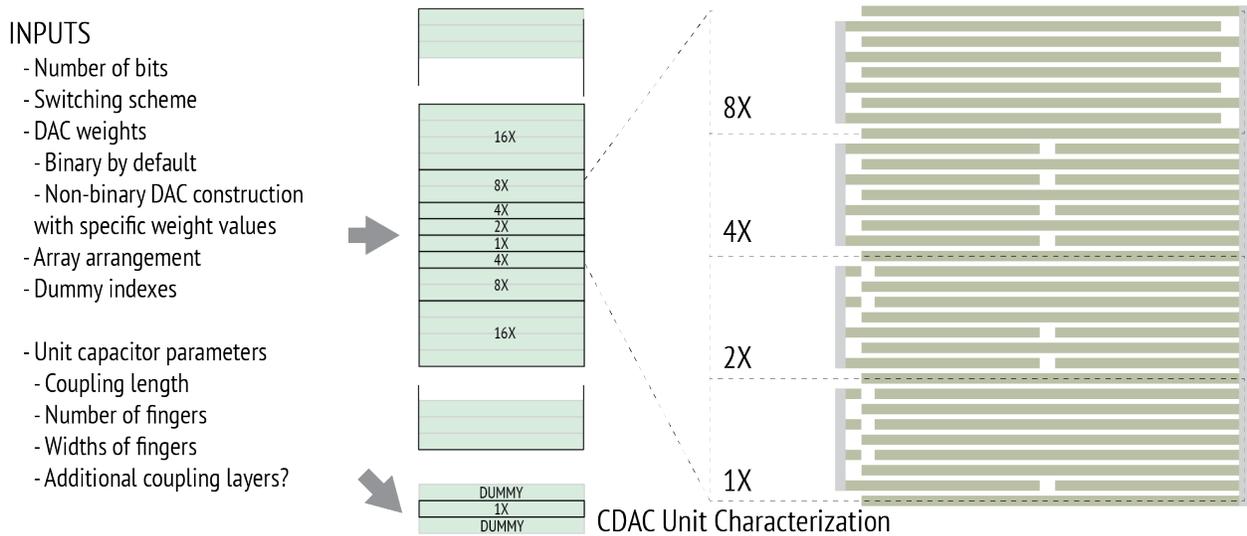


Figure 4.26: Illustration of CDAC array generation and the input parameters.

possible delay of CDAC drivers, T_{BUF} . And T_{COMP} includes the delay from the comparator clock triggers to the input of each logic cell.

The logic cell and buffers are sized depending on the size of the capacitor they drive. The initial sizing and number of stages are determined based on the calculation of logic effort. The delay in the SAR logic is minimized when it has an optimal path effort. However, since the generator only supports a single comparator SAR ADC, the T_{LOGIC} increases linearly with a larger logic cell. When combining T_{COMP} , T_{DAC} , and T_{LOGIC} , there will be a minimum delay point at which the overall loop speed can be optimized. This result can be interpreted as an alternative strategy for architecture selection. The main benefit of asynchronous operations is the elimination of wasted time spent waiting for the critical bit that does not require a long T_{LOGIC} . However, it is possible to carefully match delays for each loop so that each conversion takes approximately the same amount of time. The advantage of this approach is the reduction in the number of gates within the loop. This reduction eliminates the need for additional logic used for asynchronous clock trigger generation, which can introduce a delay of tens of picoseconds, even in advanced processes. Figure 4.27 shows the simplified diagram of the complete SAR generator, which illustrates the various configuration options available. The SAR logic cell must account for all comparator and CDAC design variations. The generator configurations are explored to maximize speed.

Considerations on CDAC switching schemes: While the sizing of the capacitor is primarily determined by thermal noise and matching requirements, the switching schemes are contingent on higher-level specifications and system design. The primary trade-off exists in the number of reference voltages, common-mode voltage requirements, and the switching

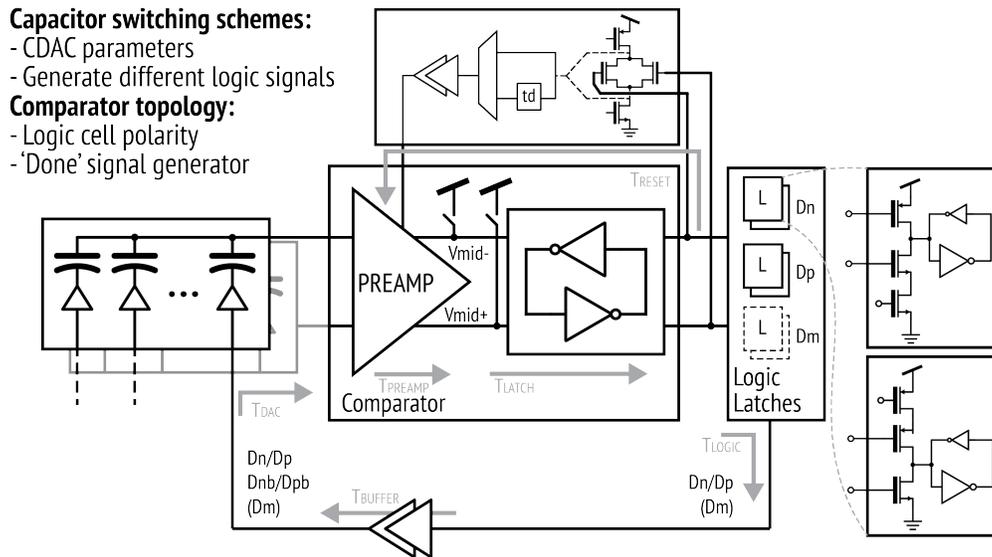


Figure 4.27: Design trade-offs of the overall SAR ADC generator.

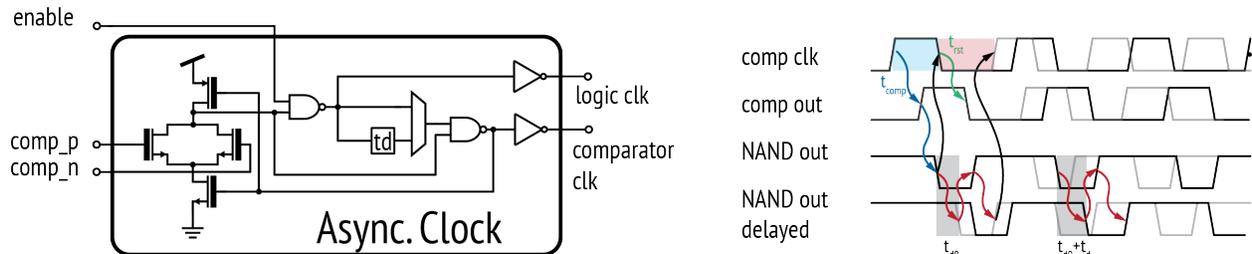


Figure 4.28: Diagram of the asynchronous clock generator and example timing waveforms.

energy. If a constant common-mode voltage is required, the split capacitor or V_{cm} -based scheme is selected, even though it requires an additional reference. If the ADC is sensitive to power consumption, a more aggressive switching scheme can reduce the energy of the V_{cm} -based scheme. The logic cells must support the generation of varying numbers of logic signals and different bit numbers. As shown in Figure 4.27, the D_m set of logic cells is optional and is only used in V_{cm} -based switching.

Considerations on the choice of a comparator: The selection of the comparator is primarily driven by the performance requirements of the ADC. Different comparator stages have corresponding reset values, which necessitate the logic cells to be sensitive to either low or high voltage levels. Removing any additional buffer for bit flipping is beneficial in order to achieve a minimal delay of the SAR logic. Therefore, two sets of complementary logic

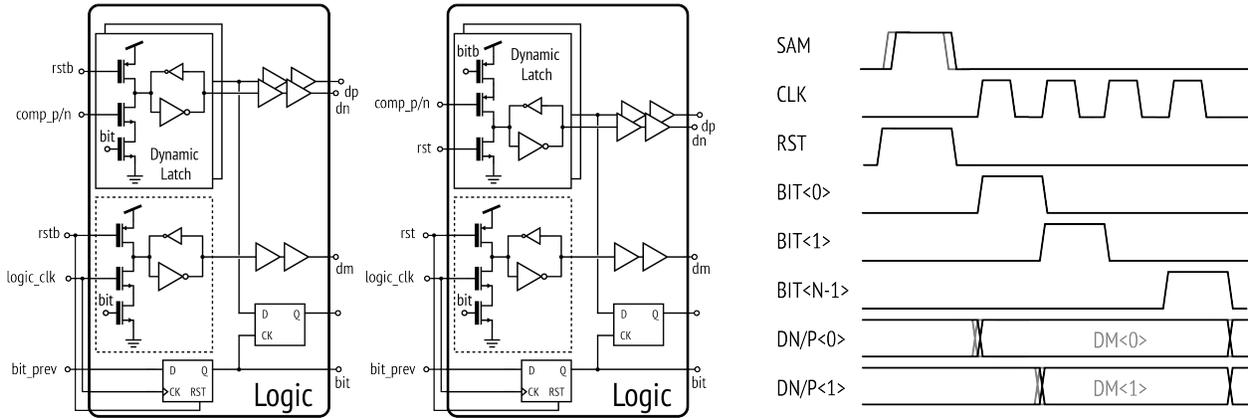


Figure 4.29: Schematics of logic cells and example timing waveforms.

cells are available to cover both cases. Also, incorporating common-mode insensitive topologies is necessary to accommodate varying common-mode voltage, such as in the monotonic switching scheme.

Asynchronous clock generator: The asynchronous clock generator, shown in Figure 4.28, oscillates to generate the clock signal for the comparator. A dynamic NAND/NOR gate is used to generate a ‘DONE’ signal and reset the comparator through the shortest path. The type of gate depends on the reset polarity of the comparator. An additional MUX selects between the original signal and the delayed version, which extends the clock period of the comparator, allowing for extra time for the settling of critical bits. Figure 4.28 shows that an additional delay, t_d , is introduced when the MUX is enabled.

The schematic of the logic cell generator is shown in Figure 4.29, along with an example timing diagram. The floorplan of the SAR logic generator is presented in Figure 4.30. Apart from the logic cells, clock buffers and data retimers are arranged on the bottom rows. Once the first bit is triggered, the bit window propagates through all the logic cells sequentially. As mentioned earlier, the logic cells need to be scaled for different capacitor weights. A list of scaling factors is passed to the generator, and the logic cell generator handles the factors. The variable number of cells is arranged from bottom to top. There are logic cells, state flops, local retimers, and buffers inside the logic cell. The SAR logic generator uses digital gates constructed in BAG, and an integration wrapper is used around the SAR logic to bring up the supply tracks to the specified top routing metal layer. The overall SAR floorplan necessitates width matching between the SAR logic and other components. Specified tracks can be used to fill the space and distribute power. Although the SAR logic provides high programmability, it does so at the expense of layout efficiency. The logic cell is wrapped in a rectangular block. The space is filled with either a tap cell or a dummy structure. Additionally, some gaps are left for logic cell alignment and are filled with tap cells at the top level of the SAR logic. The digital bits that come out from the SAR logic are retimed

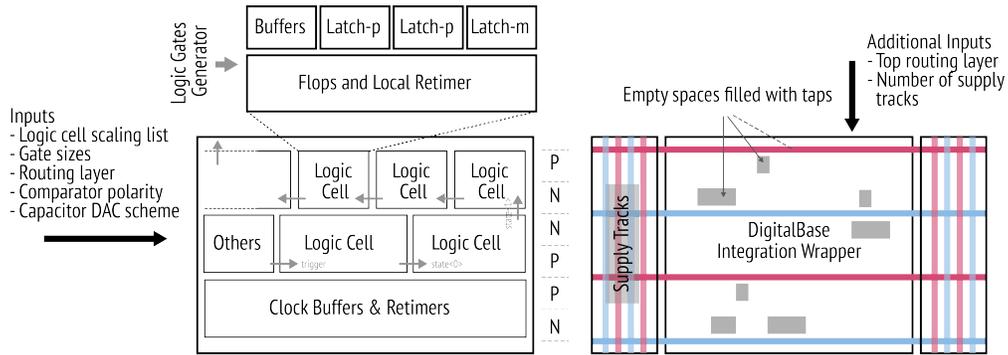


Figure 4.30: Floorplan and generation steps of the SAR logic block.

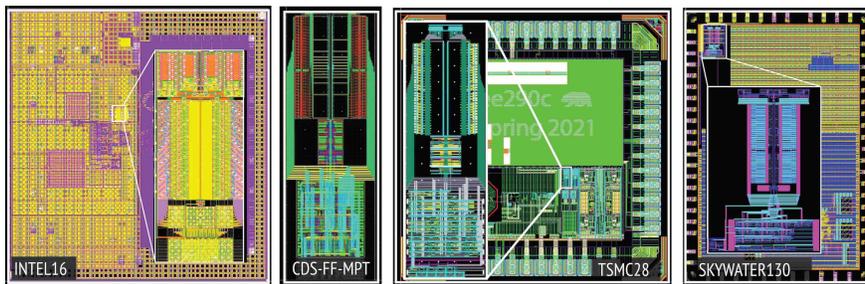


Figure 4.31: Generated SAR ADCs using different processes.

by the main clock and sent to the digital block for data capture or further processing.

4.3.4 Overall SAR ADC Generator and Floorplan

The top-level SAR ADC generator vertically assembles all the building blocks discussed above. The SAR logic block is placed at the bottom, with the comparator on top. Routings from the comparator to the logic cells are optimized to minimize parasitic effects. The capacitor DAC is positioned at the top of the comparator. Both the comparator and logic blocks are matched. Initially, all templates are created for each block, and the maximum width is used to calculate the dummy structure in narrower blocks. Integrated scripted testbenches are also included for fast verifications in the evaluation of the SAR ADC. These measurement scripts take the generated netlist or post-extraction netlist and connect it to stimuli specified in the given parameters. Both the static and dynamic properties of the ADC are evaluated. The script takes the simulation data and performs post-processing. Several generated instances using different processes are shown in Figure 4.31, which demonstrates the reusability of the generator-based design methodology.

4.4 Design of the VCO-based ADC Generator

The VCO-based ADC provides time-domain analog-to-digital conversion with first-order noise shaping, as discussed earlier. The open-loop VCO-based ADC provides higher speed compared to the closed-loop $\Delta\Sigma$ implementations. By using intermediate nodes in the ring oscillator, the resolution of the open-loop VCO-based ADC is enhanced to the gate-delay level. With interpolation techniques and cross-coupling, the resolution can be further extended to the sub-gate-delay level. This generator implements an open-loop VCO-based ADC, which can be used as a standalone generator or as a fine quantizer that follows the coarse quantization to form subranging ADCs with a hybrid architecture. Figure 4.32 shows a conceptual waveform of the open-loop VCO-based ADC. The ADC samples the input signal periodically. During the sampling phase, the control voltage (VCTRL in Figure 4.32) is connected to a constant voltage. The benefits of connecting the VCO core to a defined voltage will be analyzed later in this section. Generally speaking, it helps the VCO to start with a well-defined state that avoids dynamic effects during the transition between sampled voltages. Also, any kickback from the sampling flip-flops that occurs during the reset phase and the influence on the RO is minimized.

4.4.1 Design Considerations for the VCO-based ADC Generators

The diagram of the VCO-based ADC is shown in Figure 4.33 [53]. As mentioned in Chapter 2, the resolution of the ADC is defined as follows:

$$ENOB = \log_2\left(\frac{f_{VCO,max}}{f_s} - \frac{f_{VCO,min}}{f_s}\right), \tag{4.11}$$

where f_{VCO} denotes the oscillation frequency of the VCO. Depending on the range of oscillation frequencies, various methods of quantization, sampling, and differentiation can be employed. When $f_{VCO,max}$ is less than the ADC’s sampling frequency (f_s), at most one

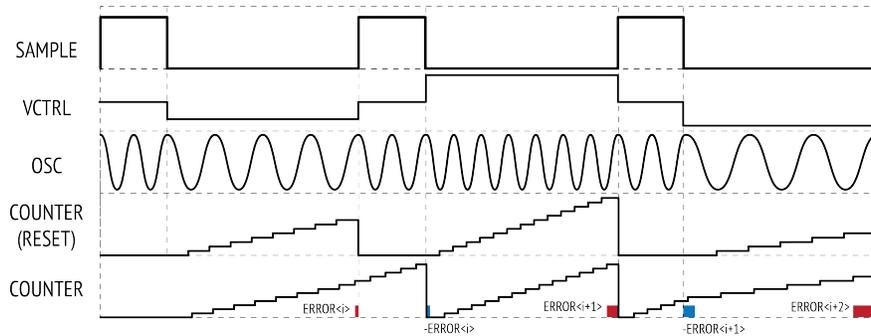


Figure 4.32: The conceptual timing diagram of a VCO-based ADC.

edge can appear at the input of the counter. The counter can then be simplified to two flip-flops and an XOR gate at the output of each oscillator phase. In the other cases where $f_{VCO,max} \geq f_s$, a multi-bit quantizer is required to handle multiple edges that appear at the VCO's output.

Moreover, the maximum speed of the oscillator affects the choice of counter implementations. Although a synchronous counter can simplify the sampling process, the delay in carry propagation limits its maximum speed. To this end, the asynchronous counter can catch up with an oscillation frequency greater than 10 GHz. However, the metastability concern in the sampling process of such counters complicates the overall design. In the design with a single-bit quantizer, sampling metastability will introduce a maximum error equal to one LSB of the ADC. The multi-bit quantizer, however, can have significant errors due to the sampling error. A double sampling technique is proposed to properly sample the output [48]. The design of the proposed generator does not include a counter reset. The comparison between the counter with and without reset is also shown in Figure 4.32. Whether resetting the counter does not bring any difference to the circuit performance. If the counter is reset, the counter output is the MSB of the ADC, while subtraction is required if the counter is not reset. Also, resetting and restarting the high-speed counter brings practical design challenges. Another design trade-off is the number of oscillator stages and the resolution of the counter. Although modifying the oscillator stages changes its frequency range, the previous analysis shows that it does not directly impact the achievable resolution in the design. Therefore, the design consideration mainly comes from other aspects, such as the stability of oscillation and the maximum speed of the counter. With differential implementation, an even number of stages can be used as long as it starts up correctly. Moreover, minimizing the number of stages reduces the power of the oscillator. The counter needs to be able to operate at the maximum oscillation frequency without consuming excessive power. In this subsection, various considerations are first examined to understand the design trade-offs in VCO-based ADC design. Implementations of critical generators in a VCO-based ADC are then discussed in following subsections.

Phase noise: Since the VCO-based ADC is primarily used as a second stage for fine quantization in this work, it is essential to understand its noise limitations. As analyzed in [156], the spectral density of the phase noise difference between adjacent samples is defined as

$$\Delta\Phi = \Phi[n+1] - \Phi[n], \quad (4.12)$$

where Φ is the sampled phase. The influence of the oscillator's phase noise can be analyzed by integrating the noise over the quantization window T_s :

$$S_{\Delta\Phi} = |W_{T_s}(f)|^2 \times S_{\Phi}(f), \quad (4.13)$$

where $S_{\Phi}(f)$ represents the power spectrum density of the phase noise, and $W_{T_s}(f)$ denotes the Fourier transform of a rectangular window with a width of T_s . The spectrum density

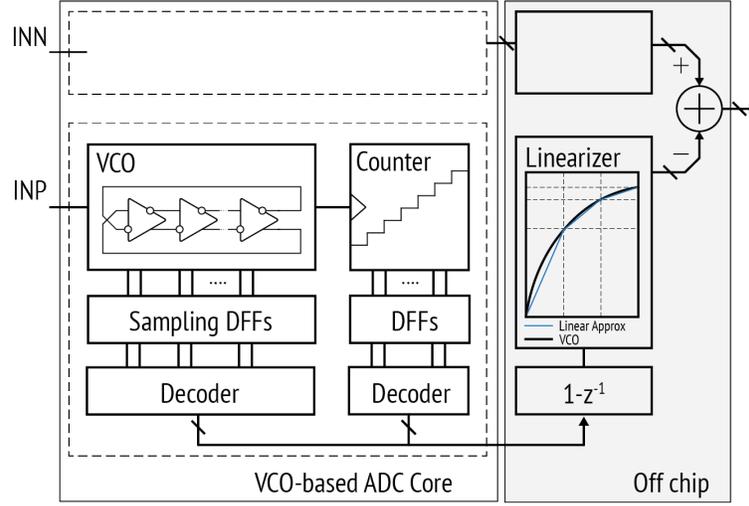


Figure 4.33: The circuit architecture of the VCO-based ADC generator.

$S_{\Delta\Phi}$ is then given by

$$S_{\Delta\Phi} = 4S_{\Phi}(f) \sin^2(\pi f / f_s), \quad (4.14)$$

where $f_s = 1/T_s$, which represents the sampling frequency of the ADC. Therefore, the mean-square value at the end of the integration window can be found using the Wiener-Khinchine theorem:

$$\sigma_{\Phi}^2 = \int_0^{\infty} 4S_{\Phi}(f) \sin^2(\pi f / f_s) df. \quad (4.15)$$

The single sideband phase noise caused by white noise is inversely proportional to the square of the frequency. The integration above with such phase noise shows that $\sigma_{\Phi}^2 \propto 1/f_s$. For the flicker noise-induced phase noise, the above integration does not converge due to the infinite phase noise as the frequency is close to the center frequency. As shown in [157] that with proper approximation, the above integration demonstrates that $\sigma_{\Phi}^2 \propto 1/f_s^2$ in this case. In the context of ADC design, the phase noise can be referred to as the input to the VCO as

$$\sigma_v^2 = \sigma_{\Phi}^2 \left(\frac{f_s}{2\pi K_{VCO}} \right)^2. \quad (4.16)$$

Therefore, σ_v^2 is proportional to f_s when the phase noise is thermal noise dominant, while f_s is not a relevant variable in the input-referred noise voltage when the flicker noise dominates.

Quantization noise and mismatch: The quantization noise of the VCO-based ADC can be calculated by modeling each delay cell in the ring oscillator as the LSB of the ADC. And the quantization errors can also be defined similarly. Assuming the power distribution function (PDF) of the quantization error $e(t)$ is uniformly distributed between $\pm\Delta/2$, where

$\Delta = 1\text{LSB}$, the quantization error causes an equivalent noise

$$\overline{e^2} = \int_{-\Delta/2}^{\Delta/2} \frac{e^2}{\Delta} de = \frac{\Delta^2}{12}. \quad (4.17)$$

As mentioned earlier, the quantization noise in an open-loop VCO-based ADC can be expressed as:

$$e = e[n+1] - e[n], \quad (4.18)$$

where e is the quantization error and $e[n]$, $e[n+1]$ are errors for samples n and $n+1$. The noise transfer function 2.12 shows that the quantization error is effectively doubled and first-order shaped. Using the VCO-based ADC as a high-speed ADC in a time-interleaved architecture does not benefit from noise shaping. This is because the in-band signal can be located near the Nyquist frequency of the sub-ADC, which is susceptible to most quantization noise. While resetting the phase brings another 3dB SNR increase due to reduced total quantization noise, shutting down and reactivating these circuits poses practical challenges in implementations. Therefore, both the RO and the counter are continuously running in the proposed generator.

The mismatches in the ring oscillator, caused by routing parasitics and device mismatches, can introduce spurious tones in the output spectrum, as analyzed in [55]. The effects of mismatch can also be analyzed statically. Assume each delay cell has differential nonlinearity (DNL) δt_i in the time domain, and the output phases are calibrated to be located at the mid-points of each decision level. The quantization noise with mismatches can then be calculated as:

$$\overline{e^2} = \frac{1}{N} \sum_{i=1}^N \int_{-\Delta/2-\Delta t_i/2}^{\Delta/2+\Delta t_i/2} \frac{e^2}{\Delta} de = \frac{1}{N} \sum_{i=1}^N \frac{(\Delta + \Delta t_i)^3}{12\Delta} = \frac{1}{N} \sum_{i=1}^N \frac{\Delta^2(1 + \delta t_i/\Delta)^3}{12}. \quad (4.19)$$

From another perspective, the assumption above holds because of the intrinsic dynamic element matching (DEM) of the ring oscillator. For the quantization of RO's phases, the same output can be obtained by traveling through different delay cells. Consequently, the INL of the RO starts from zero and then falls back to zero after one period of the oscillator. The randomized starting and ending points can be modeled as an additional noise source, σ_{DEM}^2 [35].

Sampling method: In contrast to conventional ADCs where the sampled voltage is directly quantized, VCO-based ADCs integrate the output of the S/H block using the VCO. Assume the input to the VCO is $x_{VCO}(t)$ and it can be decomposed as:

$$x_{VCO} = x[n] + e[n], \quad (4.20)$$

where $x[n]$ represents the ideal or final value it settles to, and $e(t)$ denotes the error between the transient and final values. Assume the VCO has a gain K_{VCO} :

$$\Delta\Phi[n] = \int_{nT_s}^{(n+1)T_s} K_{VCO}x_{VCO}(t)dt \quad (4.21a)$$

$$= \int_{nT_s}^{(n+1)T_s} K_{VCO}[x[n] + e[n]]dt \quad (4.21b)$$

$$= \Phi[n] + \int_{nT_s}^{(n+1)T_s} K_{VCO}e[n]dt \quad (4.21c)$$

The linear relationship reveals that the error does not degrade the performance of the ADC as long as the output is a linear function of the input [102]. The errors caused by return-to-zero (RZ) and non-return-to-zero (NRZ) sampling methods are compared here, assuming that $\alpha \cdot T_s$ is allocated for quantization in the RZ sampling scheme. In the NRZ sampling scheme, the input of the VCO switches between sampled voltages, and the phase change can be expressed as:

$$\Delta\Phi_{NRZ}[n] = \int_{nT_s}^{(n+1)T_s} K_{VCO}x[n](1 - e^{-\frac{t-nT_s}{\tau}})dt + \int_{nT_s}^{(n+1)T_s} K_{VCO}x[n-1]e^{-\frac{t-nT_s}{\tau}}(1 - e^{-\frac{-T_s}{\tau}})dt \quad (4.22a)$$

$$= K_{VCO}x[n](T_s - \tau(1 - e^{-\frac{T_s}{\tau}})) + K_{VCO}x[n-1]\tau(1 - e^{-\frac{T_s}{\tau}}) \quad (4.22b)$$

In the RZ sampling scheme, the input of the VCO connects to the sampled voltage for $\alpha \cdot T_s$ and uses the rest of the time for reset:

$$\Delta\Phi_{RZ}[n] = \int_{nT_s}^{(n+\alpha)T_s} K_{VCO}x[n](1 - e^{-\frac{t-nT_s}{\tau}})dt + \int_{(n+\alpha)T_s}^{(n+1)T_s} K_{VCO}x[n](1 - e^{-\frac{\alpha T_s}{\tau}})e^{-\frac{t-(n+\alpha)T_s}{\tau}}dt \quad (4.23a)$$

$$= K_{VCO}x[n][T_s - \tau(1 - e^{-\frac{\alpha T_s}{\tau}}) + \tau(1 - e^{-\frac{(1-\alpha)T_s}{\tau}})^2] \quad (4.23b)$$

It can be seen that the output only depends on $x[n]$. Furthermore, circuit nonlinearity issues can be easier to address.

Metastability: A large number of flip-flops are used to sample outputs from both the VCO core and the counter. Due to the asynchronous nature of the sampling process, especially for outputs that are not stable at the sampling instant, errors may occur in the sampling flip-flops due to metastability. The metastability problem has different effects when sampling the VCO phases and sampling the counter outputs. For the VCO phase sampling, the result caused by metastability is similar to that of the flash ADC in the voltage domain [158]. With proper circuit design techniques and a meticulous decoding scheme, the error can be reduced to 1 LSB. For the counter's outputs, sampling a binary counter can result in significant

errors in the sampled values. Although the Gray counter can help reduce such errors, the additional gates limit the maximum achievable speed. Therefore, a well-designed sampling flip-flop is required to minimize the probability of metastability. The metastability error in the VCO-based ADC can be derived from the sampling flip-flop's metastability. The metastability of the flip-flop can be expressed as:

$$t_{ms} = t_{ms,0} \cdot e^{-\frac{t_{res}}{\tau_{FF}}}, \quad (4.24)$$

where $t_{ms,0}$ is the asymptotic width of the metastability window when there is no resolution time, and τ_{FF} denotes the time constant of the feedback loop inside the sampling flip-flop [159]. The t_{res} is the time allocated for resolving the input, which is typically a fraction of the sampling clock cycle. For the outputs of the VCO, the probability of transitioning within the metastability window is

$$P(MS|x(t)) = \frac{2}{T_{VCO}} t_{ms} = 2t_{ms} f_{VCO}, \quad (4.25)$$

where $f_{VCO} = K_{VCO}x(t) + f_{fr}$ and f_{fr} is the free running frequency of the VCO. Consider a uniformly distributed input voltage:

$$P(MS) = \int_{-\frac{V_{FS}}{2}}^{\frac{V_{FS}}{2}} P(MS|x(t))P(x(t))dx \quad (4.26a)$$

$$= \int_{-\frac{V_{FS}}{2}}^{\frac{V_{FS}}{2}} 2t_{ms,0} \cdot e^{-\frac{t_{res}}{\tau_{FF}}} (K_{VCO}x(t) + f_{fr})dx \quad (4.26b)$$

$$= 2t_{ms,0} \cdot e^{-\frac{t_{res}}{\tau_{FF}}} f_{fr} \quad (4.26c)$$

Considering the allocated time ($\sim 1ns$) for sampling and the small time constant ($\sim 10ps$), the metastability is reduced to a negligible level. Moreover, metastability in the VCO's phase samples mixed with the effect of the sampling flip-flop's offset often manifests itself as bubbles of the cyclic thermometer code. Such errors can be further alleviated by proper decoder designs. In contrast, the error from the counter can be calculated as

$$\sigma_{MS,counter} = \sum_{i=0}^B P(MS)_{c,i} \times e_i^2 = \frac{1}{2^i} P(MS) \times (2\delta)^{i+1}, \quad (4.27)$$

where B is the number of bits in the counter, $P(MS)_{c,i}$ is the probability of metastability in the i -th bit, and e_i is the error caused at the i -th bit of the counter.

Integration time error: The sampling clock for the outputs of the VCO-based ADC also sets a time reference for the integration. Therefore, the effect of sampling clock jitter can translate to an error in the amount of time used for integration. The effect of the integration

time error can be analyzed as follows, the phase error, $\Phi_{e,int}$, caused by the integration time is:

$$\Phi_{e,int} = \int_{nT_s + \tau_{aj}[n]}^{(n+1)T_s + \tau_{aj}[n+1]} 2\pi(K_{VCO}x(t) + f_{fr})dt \quad (4.28a)$$

$$\approx 2\pi(K_{VCO}x(t) + f_{fr})(\tau_{aj}[n+1] - \tau_{aj}[n]) \quad (4.28b)$$

$$= 2\pi(K_{VCO}x(t) + f_{fr})\tau_{pj}[n] \quad (4.28c)$$

where τ_{aj} represents the absolute jitter and τ_{pj} is the period jitter. By calculating the auto-correlation on both sides and performing the discrete Fourier transform, the power spectral density (PSD) of the integration time error caused by period jitter is:

$$S_{\Phi_{e,int}}(f) = (2\pi)^2(K_{VCO}^2A^2/2 + f_{fr}^2)S_{\tau_{pj}}(f). \quad (4.29)$$

Therefore, the noise power can be expressed as

$$\sigma_{\Phi_{e,int}} = (2\pi)^2(K_{VCO}^2A^2/2 + f_{fr}^2)\sigma_{\tau_{pj}}^2, \quad (4.30)$$

where $\sigma_{\tau_{pj}}^2$ is the variance of the jitter. And $\sigma_{\tau_{pj}}^2$ can be estimated by:

$$\sigma_{\tau_{pj}}^2 = \int_0^\infty S_{\Phi,sam}(f) \frac{\sin^2(\pi f/f_s)}{(\pi f_0)^2} df, \quad (4.31)$$

where $S_{\Phi,sam}(f)$ is the power spectrum density of the sampling clock's phase noise. Therefore, minimizing the jitter of the VCO sampling clock is critical in order to reduce associated noise.

VCO control methods and parasitic pole in the control path: Most published VCO-based ADCs involve a trans-conductor driving current-starving transistor or a bias voltage controlling the speed of delay cells. Due to the large number of delay cells, a large parasitic capacitance is at the controlled node. The change in oscillation frequency can be divided into two steps: the change in the controlled voltage or current, and then the change in oscillation frequency. The delays caused by these two steps can be a serious issue in a closed-loop implementation, where the excessive loop delay causes loop stability issues. Therefore, the issue of the parasitic pole is typically needs to be addressed in closed-loop continuous-time $\Delta\Sigma$ modulator applications. However, in open-loop VCO-based ADC designs, as analyzed above in the sampling method, the extra delay is not a significant issue as long as a suitable sampling scheme is employed.

4.4.2 Voltage-Controlled Ring Oscillator Generator

A differential RO topology is used in the proposed generator, which offers many different possible configurations. The differential implementation is maintained across all possible design options. There are several advantages to using differential topology. First,

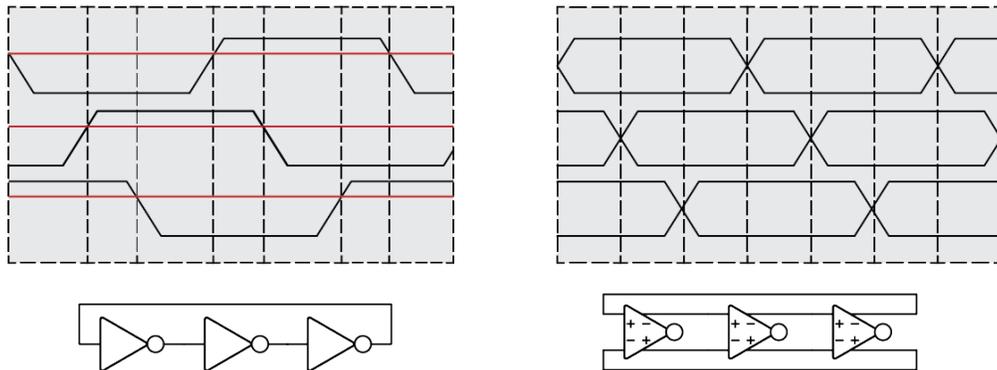


Figure 4.34: Comparison of the phase sampling in single-ended and differential ring oscillators [60].

the pseudo-differential or differential topology protects the RO from several common-mode non-idealities, such as the supply noise and the kickback noise from the comparators. Additionally, implementing RO in differential topology makes it possible to use even-order stages or non-inverting stages by swapping differential outputs and connecting them to inputs of the next stage. Moreover, the mismatch in rising and falling transition times in single-ended RO creates severe issues with phase quantization. As mentioned in [60], any mismatch between the falling and rising transitions directly translates to code nonuniformity. Because the rise and fall of nodes are primarily driven by PMOS and NMOS transistors separately, achieving matching delays for these two components across process corners and environmental variations is challenging. Considering the threshold difference between logic 1 and 0 for the sampling flip-flops, this disparity is further amplified by the difference between $V_{DD,RO} - V_{TH,FF}$ and $V_{TH,FF} - V_{SS,RO}$. Here, $V_{TH,FF}$ represents the flip-flops' threshold, while $V_{DD,RO}$ and $V_{SS,RO}$ denote the equivalent supply for RO. Implementing differential topology enables more robust sampling because the logic threshold voltage of the flip-flops is now the crossing point of the differential outputs, as shown in Figure 4.34.

Figure 4.35 shows the diagram of the VCO generator, including the RO discussed above. First of all, there are two options for controlling the frequency in the proposed generator. These two options both use a current-starved inverter to regulate the current of each stage. As shown on the left side of Figure 4.35, the difference lies in whether the tail nodes of the inverters are connected together. Besides the discussion about large capacitance at the controlled node in the previous subsection, the design trade-off of these connections is as follows. Using distributed tail current creates an imbalance between the pull-up and pull-down paths. Using a common tail transistor essentially provides a separate supply for the RO. Therefore, both PMOS and NMOS transistors in the ring see the same overdrive voltage and the transition time is better matched. One potential issue of this virtual supply is the swing of the RO, which could pose challenges in designing intermediate buffers that drive the counter. Also, depending on the specific ADC implementation, this effect might create

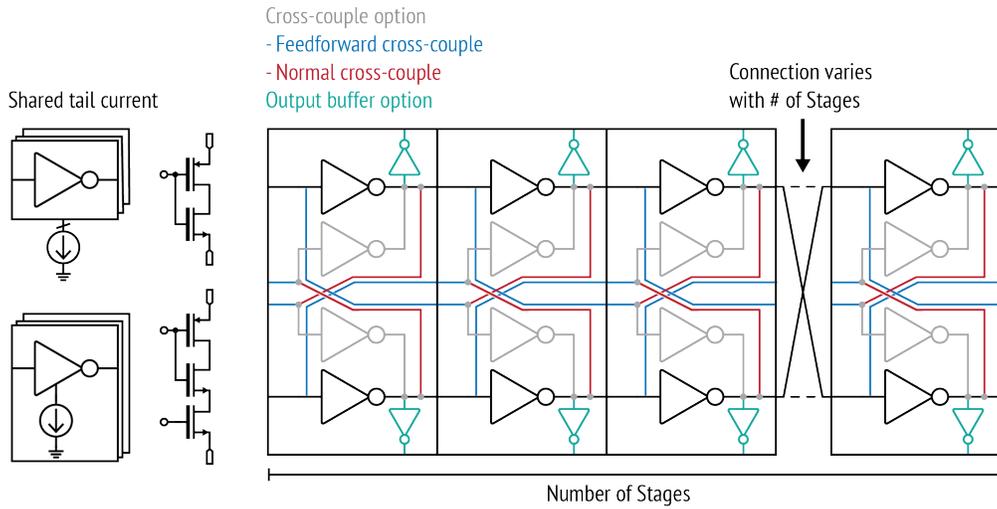


Figure 4.35: Diagram of the ring oscillator generator.

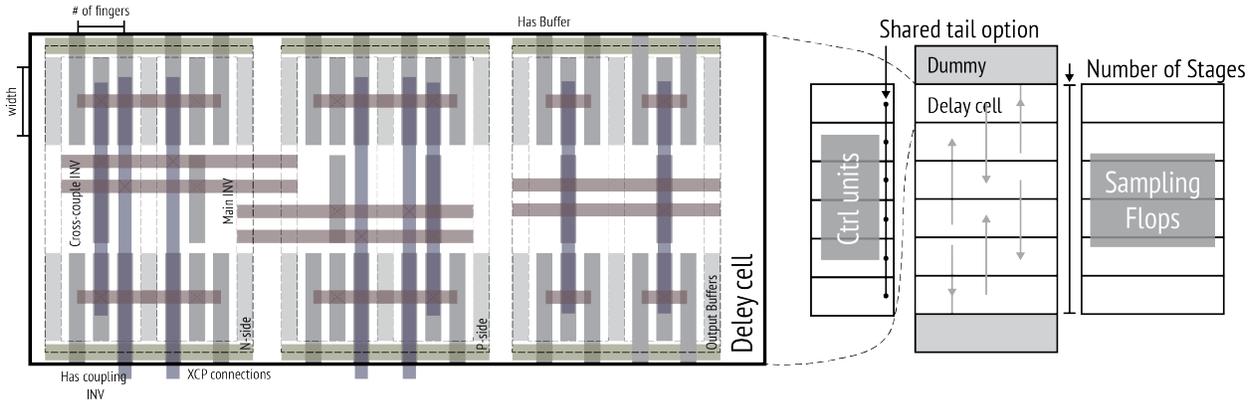


Figure 4.36: The layout details of the ring oscillator stage and the floorplan of the ring oscillator generator.

an input-dependent memory effect if the VCO is not properly reset [160, 161]. In contrast, the distributed tail connection has a much faster settling speed, and the step control input can instantaneously affect the oscillation frequency due to the much higher frequency of the parasitic pole.

Another option in the generator is the cross-coupling method. Coupling the output from previous stages can increase the RO frequency [72]. Also, cross-coupling helps to mitigate common-mode oscillation [53], which is the state in which the differential oscillator splits into two parallel oscillators with the same phases at each node. Two cross-coupling options are shown in Figure 4.35. The normal cross-coupling is shown in red, and the feed-forward

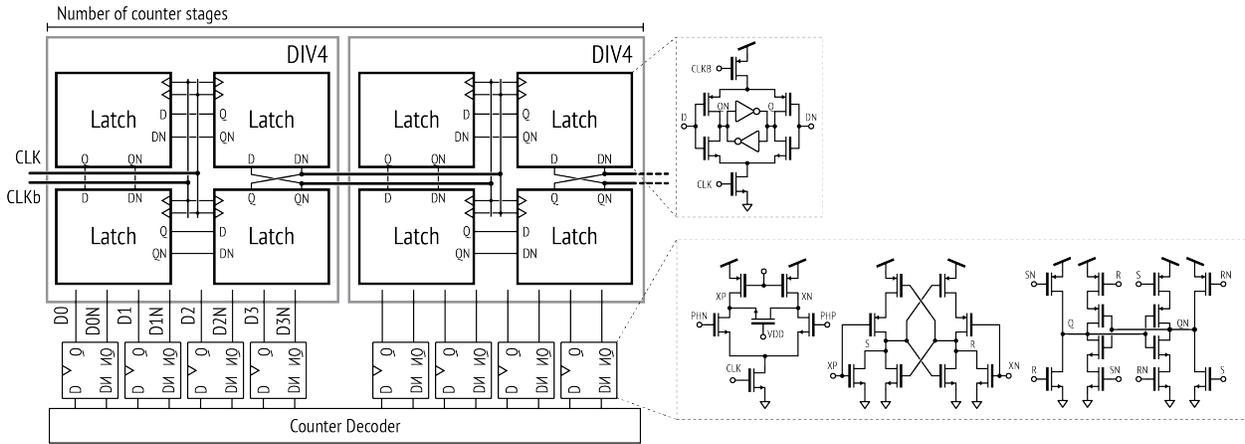


Figure 4.37: Schematics of the asynchronous counter generator.

coupling is shown in blue. Other than the two options mentioned above, the RO generator also provides output buffers. The design trade-off in the output buffers exists between the power consumption of the buffers and the isolation from the sampling flip-flops. Adding an output buffer to the RO before connecting it to the sampling flip-flops helps to mitigate the kickback noise from the flip-flops. When the clock is triggered, charge kickback from the parasitic capacitors in the flip-flops injects into the RO and disrupts the normal oscillation. Adding buffers absorbs the excess charge. But the buffers operate at the same frequency as the RO, which significantly increases the power consumption of the design. Depending on the specific implementation, the kickback may not be a severe issue for the overall operation. The number of stages in RO is also flexible and can be changed. In the case of even stages, the wire swapping is automatically handled by the generator script.

The layout floorplan of the generator is shown in Figure 4.36. On the left side, the figure shows the unit row of the ring, which consists of the main inverter pair, a coupling pair, and two output buffers (if the option is enabled). The RO is arranged vertically, with upward and downward signal propagation placed in an interleaved way to balance the parasitic effects at each node. Dummy rows are added at both ends to improve stage matching. The current tail transistors are designed to match the height of the unit stage and are connected together when the `shared_tail` option is enabled.

4.4.3 Counter Generators

As mentioned earlier, the design of the counter is also important. The counter needs to be fast enough to cover the entire target range of oscillation while maintaining low power consumption. Both synchronous [162, 100, 163, 52] and asynchronous [57, 164] designs are used. The carry propagation limits the design of synchronous counters. Therefore, running at frequencies higher than 10 GHz is challenging. Also, running all the flip-flops at the

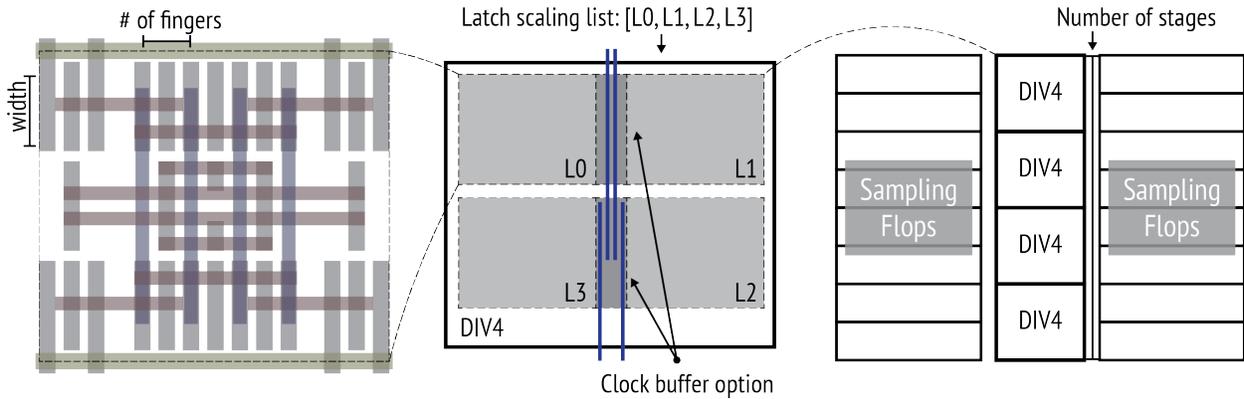


Figure 4.38: The layout details and floorplan of the counter generator.

full clock frequency consumes a significant amount of power and introduces noise into the oscillator.

An asynchronous counter that utilizes a chain of divide-by-2 blocks[165] is employed. Due to the long propagation delay, there is a possibility that the outputs of the counter are undergoing transition at the sampling moment. Therefore, a double sampling technique is used [164, 57] to reliably capture the output data. The current or delayed version of the outputs is selected based on the input clock state. The diagram of the counter and sampling flip-flops is shown in Figure 4.37. The generator takes input parameters such as the number of bits in the counter and divider, as well as various transistor sizes. The same sampling flip-flops used for the VCO output are reused here. Figure 4.38 shows the layout floorplan of the layout generator. Each stage in the divider and counter scales independently to provide sufficient drive strength. Additional inverters can be inserted between divide-by-2 stages as well.

4.4.4 Sampling Flip-Flops

As mentioned in section 4.4.1, sampling flip-flops with low metastability rate is critical in the VCO-based ADC design. A sense-amplifier-based differential flip-flop is used in the generator for sampling in the VCO to ensure reliable differential sampling. The sense-amplifier-based flip-flops also provide a higher gain, which helps to mitigate metastability and static errors. The sampling flip-flops are constructed based on the modified self-timed comparator shown in Figure 4.13 (f). This topology offers high energy efficiency due to the delayed starting time of the second stage. A self-timed version is used to avoid generating two clock phases and reduce the offset of the sampling flip-flops. An additional transistor is connected between the differential sides, which fully discharges the differential structure in each clock cycle. This additional transistor ensures that even if the input changes right after the clock edge, both nodes can still be pulled to ground. This extra transistor is minimized to prevent

degradation to the performance because of the race condition of simultaneous discharging at two sides. Ideally, a high-threshold transistor should be used here to address this issue. However, due to the limitation of the generator floorplan and the requirement for a compact layout to match the delay stage’s height, only the size of the transistor can be programmed. An S-R latch follows the double-tail comparator and stores the output value. A modified structure [166] is used to provide a compact layout and a symmetric pull-up and pull-down structure.

Figure 4.39 shows the schematic of the unit row on the right in Figure 4.36. The ring oscillator, buffers, and sampling flip-flops share the same `pinfo`, which is the row definition in the `MOSBase`. Unfortunately, the comparator generator discussed in the previous section is not reused here because it occupies a large amount of space. The sampling flip-flop generator is designed to provide a compact layout. Another set of comparator stages, which are compatible with the oscillator and counter stages’ floorplan, are used to build sampling flip-flops.

A MUX-based thermometer-to-binary decoder [12] is used for phase decoding. The design choice is made because of the simplicity of this topology compared to a ROM, a counter, or Wallace tree-based decoders. The MUX-based decoder incorporates a binary search algorithm, which effectively mitigates the bubble error caused by any residual sampling error. As mentioned earlier in this section, such a decoding scheme further lowers the metastability rate. The regular layout of this circuit is also favored by the generator-based methodology. The generator utilizes the MUX gate from the BAG digital gates library, and the floorplan is depicted on the right side of Figure 4.40. The generator takes input parameters for the number of bits and sizes for various gates.

4.4.5 The Counter Buffer Generator

The final component of the VCO-based ADC generator is the intermediate buffer, which connects the oscillator to the asynchronous counter. This buffer is necessary because the output waveform of the oscillator has a wide range of amplitude and frequency. And the counter needs to be driven by a full-swing square wave. According to the simulation, the

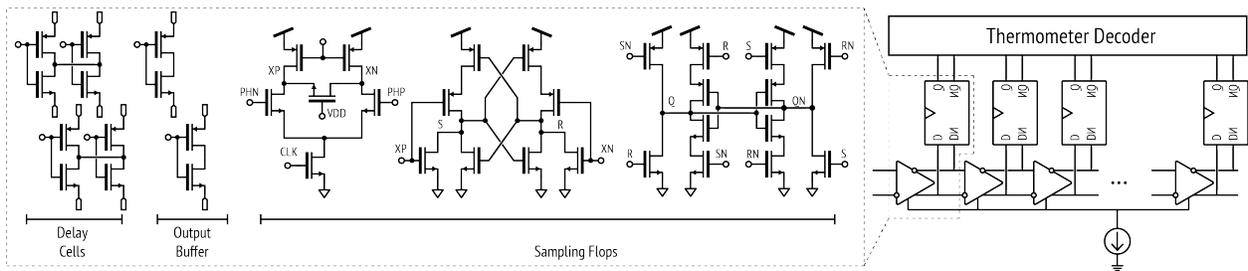


Figure 4.39: Schematics of the sampling flip-flops for a single-stage VCO.

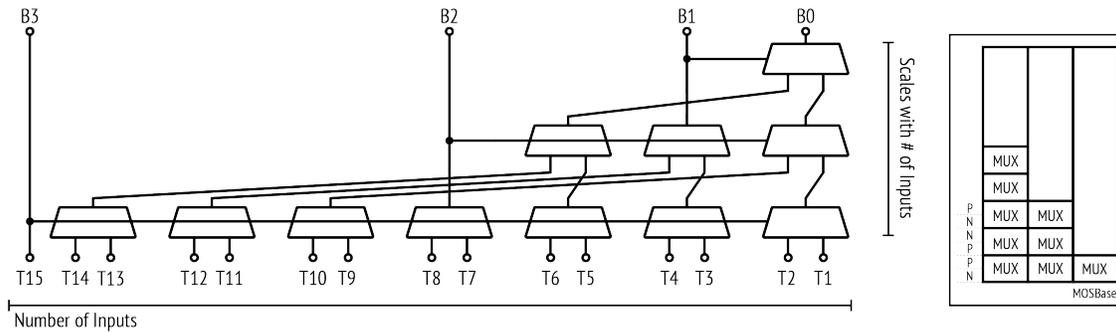


Figure 4.40: Diagram of the MUX-based thermometer-to-binary phase decoder.

ring oscillator has an output swing range of 300 mV to 800 mV under a 1 V supply. A basic inverter is unable to buffer such a signal reliably. A level shifter with cross-coupled transistors can be used to restore the swing. However, the power consumption of the buffer is high due to the partially turned-on transistors. To accomplish the buffer function while minimizing power consumption, a trans-impedance amplifier (TIA) is employed. First, the signal from the VCO is AC coupled to the TIA, and further cross-coupling is applied to correct the duty cycle and sharpen the edges, resulting in a full-swing square wave signal as the final output. The buffer generator simply combines several standard inverter generators. The resistor in the design is implemented by the transistor as well to simplify the generator floorplan. Because combining a resistor and transistor in a generator requires an additional step of integration using the `TemplateBase`. Due to a similar reason, the coupling capacitor is also constructed in the `MosBase` with lower metal layers, utilizing the grid system of the `MOSBase`. In this way, a compact design that is compatible with floorplans of the counter, flip-flops and the decoder is generated.

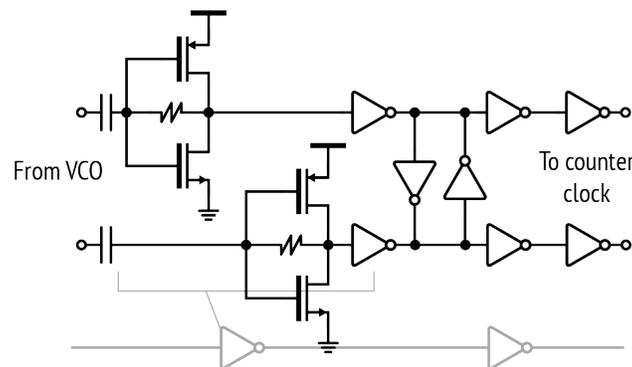


Figure 4.41: Schematic of intermediate buffers between the RO and the counter.

4.5 Residue Amplification and Ring Amplifiers

The residue amplification is an essential function in the generator. Building high-performance and low-power amplifiers becomes a challenge for any pipelined ADC design. Although it is possible to use passive residue transfer to perform the pipelining function, the noise and error contributions from the second and later stages become dominant in the overall ADC design, and their power consumption increases exponentially in order to attenuate the noise. Therefore, despite the high power consumption, residue application is a critical building block that mitigates errors in later stages and leads to substantial power savings overall. The design of a high-gain operational transconductance amplifier (OTA) becomes challenging with lower intrinsic transistor gain and lower supply voltage in modern processes. Although different gain-boosting techniques have been proposed, the limitations in the output swing and the speed make it challenging to utilize these techniques in high-speed ADC design. Therefore, the poor power efficiency and the limited bandwidth of conventional amplifiers have become the bottleneck in the design of pipelined ADCs.

Many alternative methods have been proposed to enhance the energy efficiency of residue amplifiers. Figure 4.42 shows a diagram illustrating the classification of possible residue amplifier options. The open-loop OTA-based amplifiers have also been used in residue amplification. Compared to the closed-loop implementation, the open-loop OTA offers better efficiency and lower nonlinear gain [89]. An example of the open-loop OTA is shown in Figure 4.43 (b) [167, 168]. In the amplification process of such amplifiers, the output goes through a slewing process first, and then the change rate of outputs decays and stabilizes exponentially. The inadequate time for exponential settling leads to the use of dynamic amplification or current integration mechanism. [169, 170, 171]. The schematic of a dynamic amplifier is shown in Figure 4.43 (a). The dynamic amplifiers are used in open-loop configurations for their simple implementation and noise-filtering features. The gain of a basic dynamic amplifier is defined by

$$A_v = \frac{g_m}{I_d}(V_{DD} - V_{CM}). \tag{4.32}$$

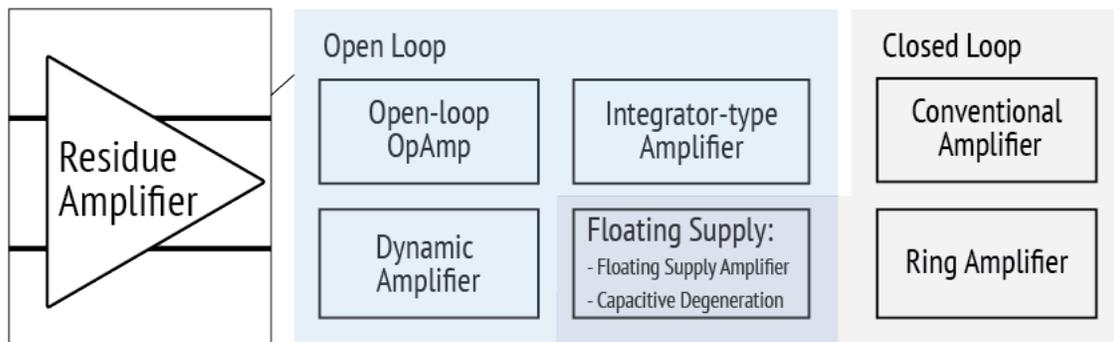


Figure 4.42: Classification of popular residue amplifier topologies.

This equation reveals that the gain of the amplifier depends on the fundamental properties of a given process. The drawbacks of the dynamic amplifier include low gain, poor linearity, and susceptibility to PVT variation. Different variants based on the basic topology have been proposed to improve gain [172, 173, 174, 175, 176, 84], linearity [177], and achieve PVT tracking [178, 87]. There are also integrator-type OTAs used in an open-loop configuration. Similar to the dynamic amplifier, it operates in the incomplete-settling region. Another category of amplifiers uses the dynamics of inverter operation with a floating supply (Figure 4.43 (c)). [179, 180]. The inverter works under a dynamically biased supply during the integration process, which boosts its g_m/I_D value. Also, the output capacitor is not fully discharged. All of the aforementioned techniques significantly improve energy efficiency compared to the dynamic amplifier. Although similar in topology, the capacitive degenerative topology exploits the compressive and expansive nonlinearities in common-source and source-degeneration configurations when the transistors operate in the sub-threshold region. The capacitive degeneration allows the circuit to dynamically transition between two topologies and cancels out the nonlinearity in this two-phase amplification. With precise control over the amplification time, this topology exhibits excellent linearity with $HD3 \leq -100$ dB. [181, 182, 82]. A floating supply inherently makes the circuit less sensitive to changes in the supply voltage and provides greater common-mode rejection. Therefore, similar topologies are also found to be used in closed-loop configurations. [183, 184, 185]. As for the closed-loop amplifier, the ring amplifier proposed in [186, 187] is one of the best alternatives to the conventional OTA and achieves high energy efficiency. The ring amplifier consists of several inverter stages with embedded offset voltage. The inverter chain is dynamically stabilized with this offset voltage. With a large signal swing, the device exhibits slewing behavior that allows rapid settling. The final stage is driven into the sub-threshold when the input signal swing diminishes. Therefore, the high output resistance significantly shifts the dominant pole towards low frequency and stabilizes the loop. Among these topologies, the ring amplifier provides large and accurate gains, as well as a compact and regular layout. The proposed ADC generator incorporates the ring amplifier as a critical block, enabling the subranging hybrid architectures and pipelined ADCs.

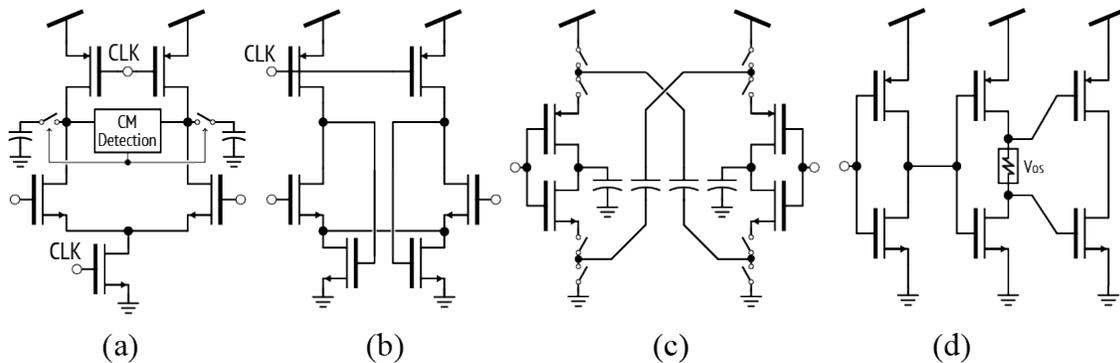


Figure 4.43: Schematics of (a) a dynamic amplifier, (b) an open-loop OpAmp, (c) an amplifier with floating supply, and (d) a ring amplifier.

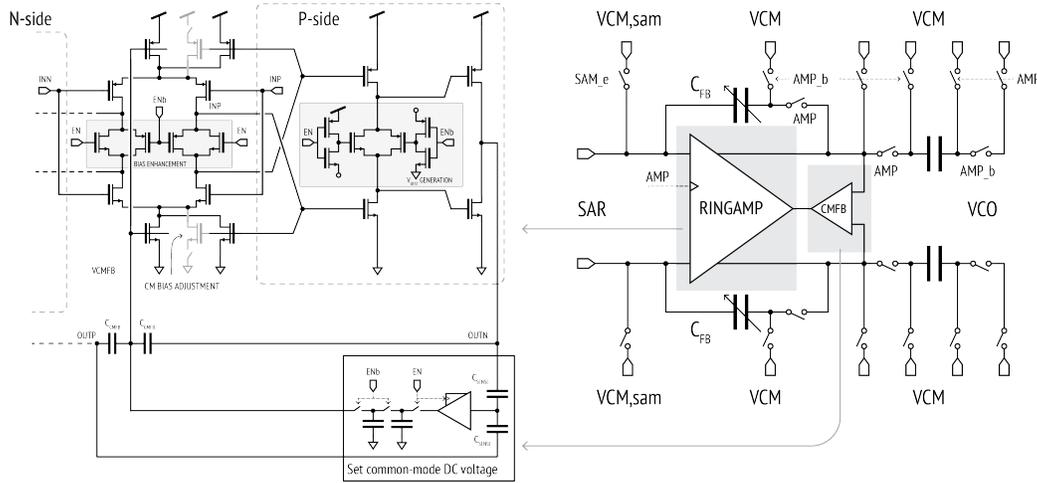


Figure 4.44: Schematic of the complete ring amplifier generator.

4.5.1 Ring Amplifier Generator

The generator selects the ring amplifier because of its fast amplification, which makes it suitable for the high-speed pipelined ADC. Additionally, the closed-loop operation provides a broad range of gains that are suitable for pipelined ADCs, pipelined SARs, and other hybrid topologies. The schematic of the ring amplifier generator is shown in Figure 4.44. A parallel arrangement of CMOS transistors [188] is embedded to create a voltage offset, which reduces the overdrive voltages of the third stage and pushes the dominant pole inward. The CMOS implementation provides extra tunability to the ring amplifier. The bias voltage of the CMOS resistor serves as the tuning knob for the embedded voltage and adjusts the settling behavior of the amplifier. Similar CMOS resistors are placed in the first stage [189, 190] to increase the overdrive voltage of the second stage. The biased enhancement also provides a way to disable the first stage during reset. Pulling the EN and ENb ports to the supply and the ground turns off the amplifier when it is not in use and saves power. The complete closed-loop residue amplifier is also shown in Figure 4.44 [191]. The amplifier's bias voltage is generated on-chip. More details will be discussed in the prototype implementation. Furthermore, the common-mode voltage is determined by sensing the average common-mode voltage during the amplification phase. This common-mode loop adaptively sets the DC voltage. Similar to [192], multiple additional common-mode feedback paths are used to stabilize the ring amplifier across process corners and different gain settings, as shown in Figure 4.45. Capacitive feedback (shown in green) senses the output during amplification and provides instantaneous feedback to the first stage. Local feedback paths (shown in red and blue) are implemented in the first stage. As the strength of PMOS and NMOS varies at different corners, a separate set of transistors (shown in gray) are added to adjust the feedback strengths, providing sufficient phase margin in different scenarios.

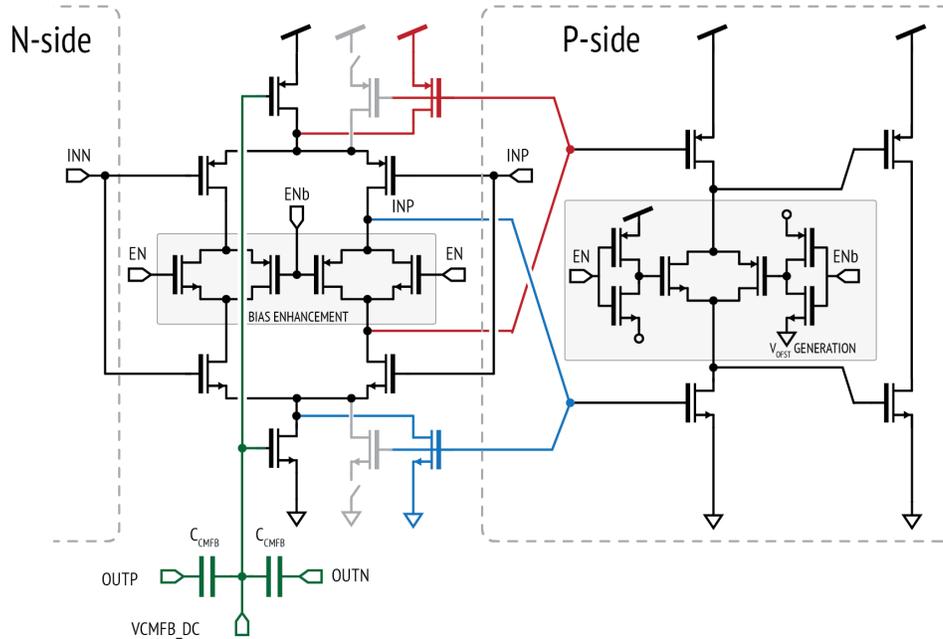


Figure 4.45: Common-mode feedback paths and the stability calibration transistors in the closed-loop ring amplifier.

4.6 Auxiliary Circuits

In addition to the critical components in single-channel ADCs, auxiliary circuits are also indispensable in the proposed ADC generator. As mentioned earlier, the time-interleaving technique is the key to extending the bandwidth of ADCs. Generating instances with different channels can easily meet various speed targets. Therefore, clocking circuits for the time-interleaving operation and generating different phases for the sub-ADCs are important. Besides, calibration circuits for the sub-ADCs errors and time-interleaving errors are included.

Clock generation circuit: In the time-interleaved architecture, it is crucial to distribute high-quality clock signals to sub-ADCs. The jitter in the sampling clock can directly translate into noise in the sampled signal and cannot be corrected later in the ADC. For a sinusoidal signal with amplitude A_{sig} , $V_{sig} = A_{sig} \sin(2\pi f_{sig} \cdot t)$, sampled by a clock with period $T = T_s$, the maximum error occurs at the zero crossing moment. The maximum jitter

$$v_{jitter,max} = 2\pi A_{sig} f_{sig} \delta t, \quad (4.33)$$

where δ_t is the difference between the actual sampling instant and the ideal instant. For a sampling clock with a standard deviation of jitter $\sigma_{t,rms}$, the average noise power can be calculated as

$$v_{n,jitter}^2 = \frac{1}{T_s} \int_0^{T_s} \left(\frac{dV_{in}}{dt}\right)^2 \sigma_{t,rms}^2 = 2(\pi f_{sig} A_{sig})^2 \sigma_{t,rms}^2. \quad (4.34)$$

Therefore, the resulting SNR is

$$SNR_{jitter} = 20 \log\left(\frac{1}{2\pi f_{in} \sigma_{t,rms}}\right). \quad (4.35)$$

Depending on the sampling scheme of the ADC, typically only one clock edge limits the performance and needs to be optimized with minimal additive jitter. Another important function of the clocking circuit is to correct the timing skew between sub-ADCs. As mentioned in Chapter 2, the skew between channels directly translates to the spurs in the spectrum and limits the ADC performance. As discussed in [193], the upper limits of the skew between channels can be approximated as follows:

$$\sigma_{skew} = \sqrt{\frac{M}{M-1} \cdot \frac{1}{2^{2N}} \cdot \frac{2}{3(2\pi f_{sig})^2}}, \quad (4.36)$$

where M is the number of channels and N represents the target ENOB of the sub-ADCs. The performance target of the proposed generator necessitates an effective suppression of

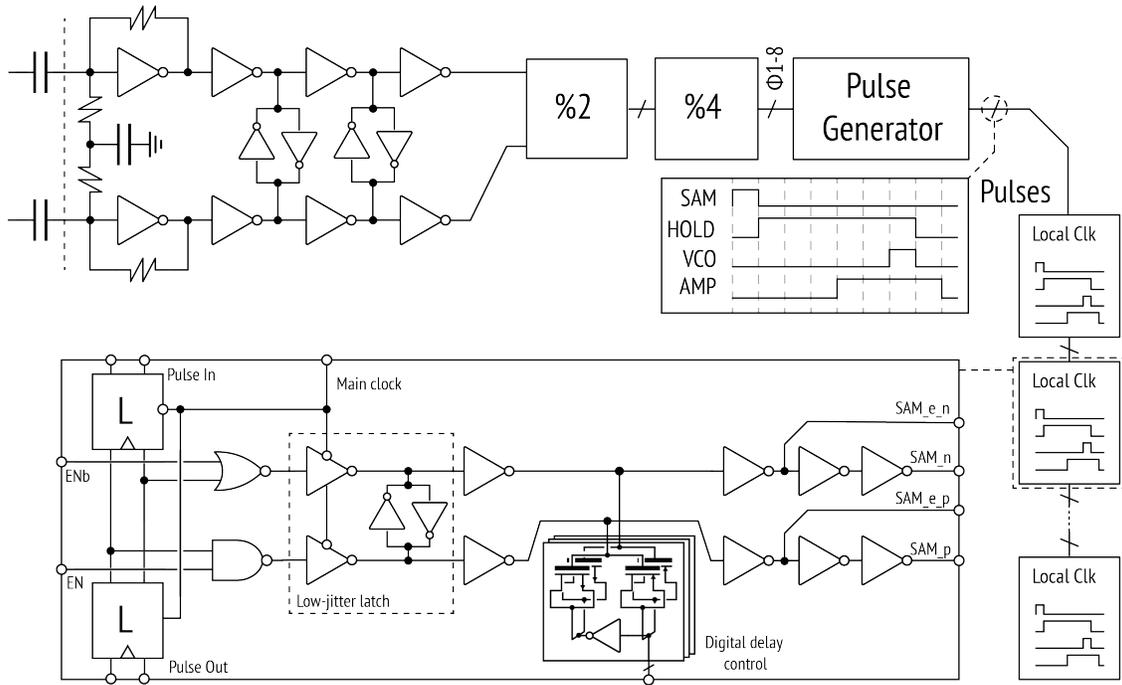


Figure 4.46: Diagram of the clock generation circuits for the time-interleaved ADC.

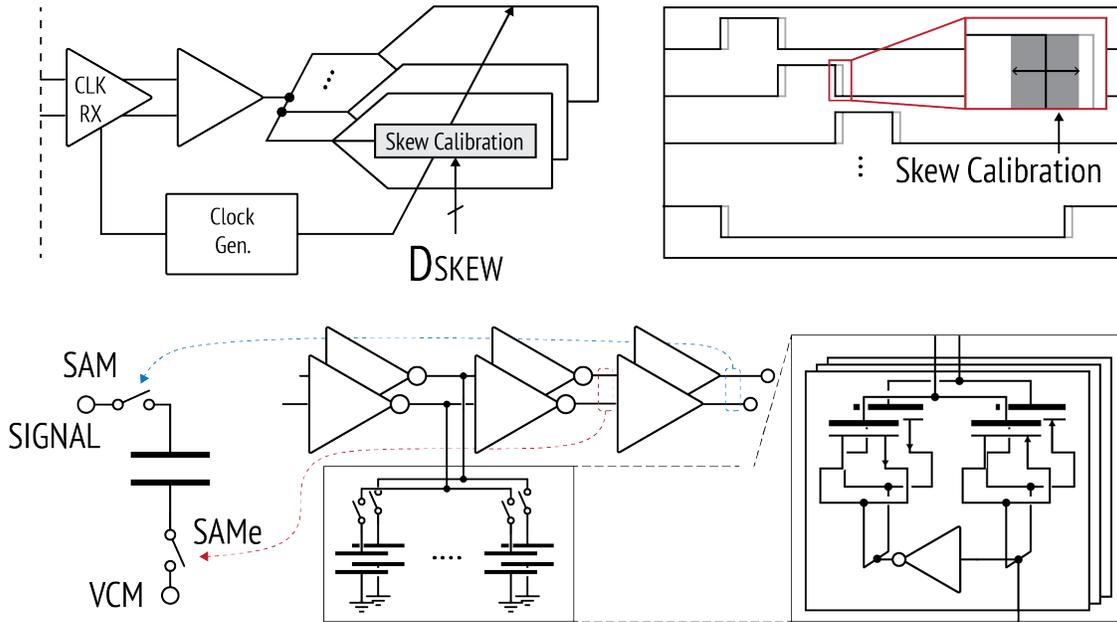


Figure 4.47: Diagram illustrating the skew calibration process in the ADC generator.

such interleaving spurs. For example, an 8-channel TI-ADC running at 4 GS/s needs less than 10 fs to maintain interleaving tones 80 dB lower than the fundamental tone. The clock generation circuit is shown in Figure 4.46. The clocking circuit generates clocks for multiple channels on the chip from the full-speed clock with a frequency of f_{clk} . A clock receiver is used to amplify the externally supplied full-speed clock. Large AC coupling capacitors are used to couple the clock signal in order to achieve wideband operation. Moreover, on-chip termination is used to improve matching. The TIAs in the clock receiver provide an internal bias point. When comparing the TIA with CML, CML has better supply noise immunity but consumes excessive power. The buffers in the following stages provide duty cycle correction and convert the sinusoidal input into a differential square wave with a sufficiently short rise and fall time. The buffered full-speed clock drives a divider, which is implemented as a divide-by-2 followed by a divide-by-4, generating eight phases from the main clock with a 12.5% duty cycle and $f_{clk}/8$ frequency by using NAND and NOR gates. The pulse generator takes divided phases and converts them into pulses for the operation of the ADC. The pulses are then passed to each channel, activating all the channels sequentially using a low-jitter latch and the full-speed clock.

Channel skew correction: Channel skew correction is achieved by tuning the critical early sampling clock using a single-stage floating transistor DAC. The diagram of the skew correction circuit is shown in Figure 4.47. The combination of a thermometer and a binary code DACs is used to achieve fine steps and a sufficient tuning range. Compared to a multistage approach, a single-stage approach allows for a more compact layout design [92].

The proposed circuit offers a solution for high-speed ADCs in the presence of timing skew, duty cycle distortion, and clock jitter.

Bias Voltage generation: In addition to the circuit described above, the generator incorporates several voltage DAC circuits intended for different calibration purposes. These circuits include the resistive DAC (RDAC) depicted in Figure 4.48, which supplies a general bias voltage. Additionally, the trapped-charge bias control proposed in [191], is specifically tailored for ring amplifier biasing, as it provides voltage beyond the supply rails. The single-ended RDAC implementation is shown in Figure 4.48 (a), which is used to generate the common-mode voltage for both the ring amplifier and the input to the reference buffer. Meanwhile, the differential RDAC (Figure 4.48 (b)) is designed to provide offset calibration for the comparators. Both the single-ended and differential RDAC implementations offer the option to incorporate sufficient decoupling capacitors at their outputs, which helps filter out the noise at the output. Both generators are highly scalable in terms of their resolution and tuning range. The total number of bits is partitioned into different numbers of bits for row and column decoders, ensuring a compact layout. The generator can take the required range and resolution parameters and automatically generate instances with the appropriate configurations.

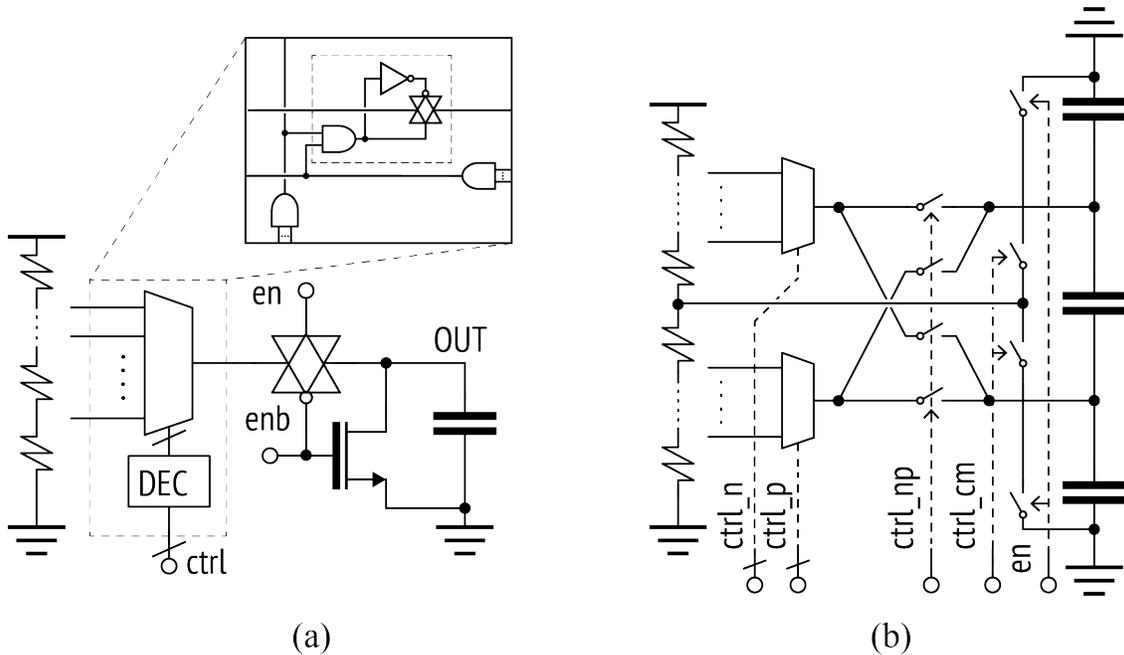


Figure 4.48: Diagrams of the supported resistor-based DAC generators.

4.7 Summary

This chapter describes the critical building blocks in the proposed ADC generator. The implementation of the generators involves the use of different Python base classes in the BAG framework to create various circuits. Figure 4.49 summarizes the code hierarchy of the proposed generator.

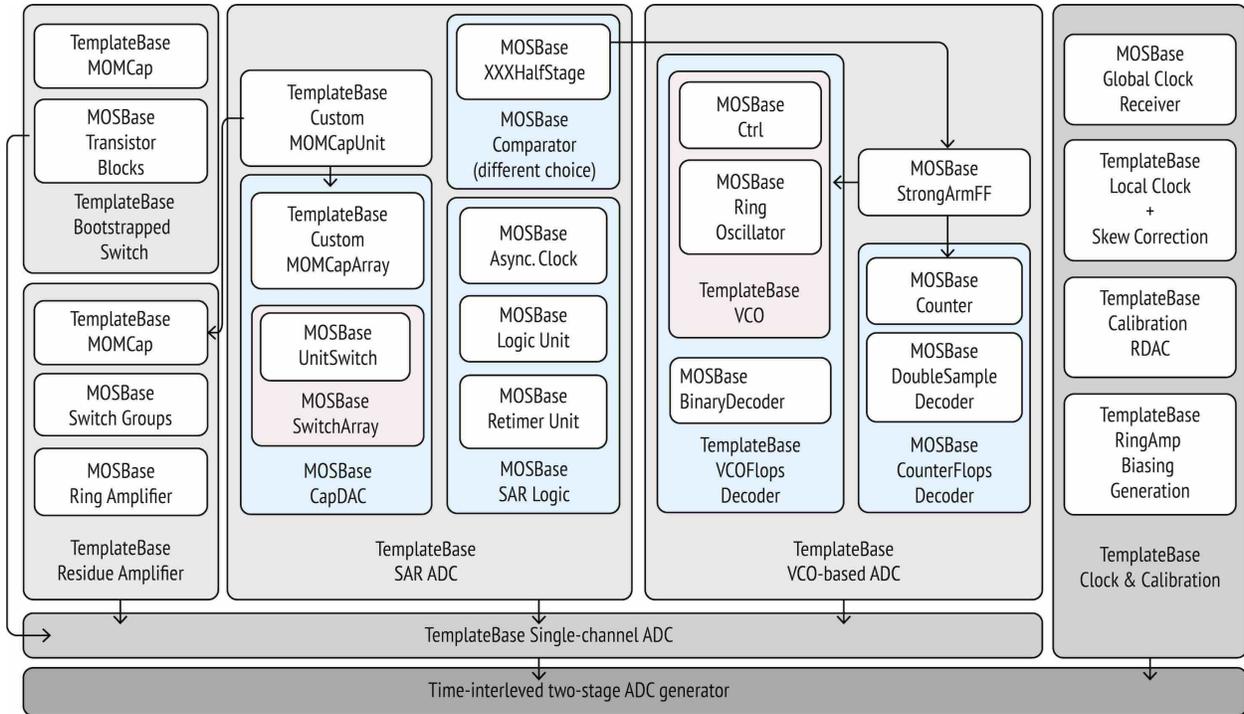


Figure 4.49: The diagram of the generator architecture.

Figure 4.50 provides a comprehensive summary of the crucial components of the ADC generator implemented using the BAG framework. This figure also highlights various design options and the floorplans of the circuit. The example layouts included in the figure were generated using the Intel 16 process technology. The lower left portion of the figure illustrates the circuit verification process executed within the generator framework. Although the primary focus of this research is not on developing a fully automated closed-loop design script for ADC, the generator framework incorporates various test and measurement scripts to expedite the circuit evaluation process and help designers make design iterations and decisions. The BAG framework is capable of initiating multiple post-layout extracted simulations with varying input parameters. Simple search algorithms have been implemented for some blocks to optimize the circuit’s performance. When a well-structured generator is employed, machine learning algorithms can be applied to the circuit design, as demonstrated in [118]. Critical block generators have been tested with machine learning algorithms, subsequently

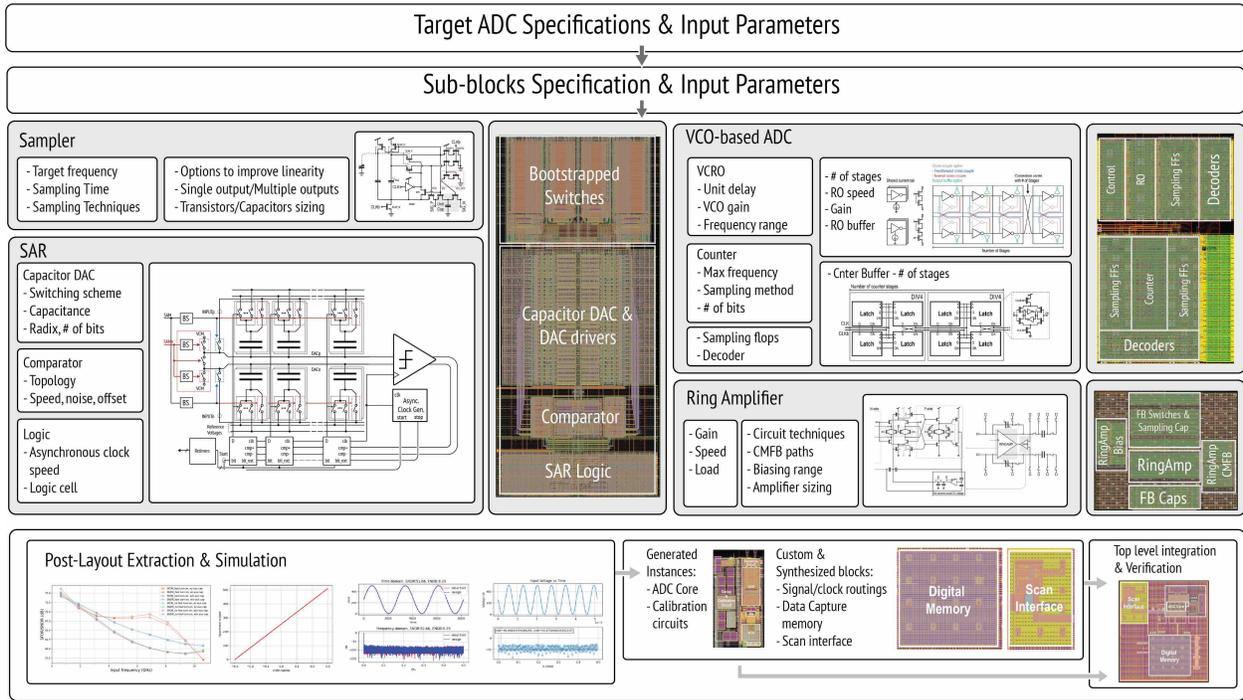


Figure 4.50: The diagram of the circuit architecture, floorplan, and available parameters.

inspiring the implementation of prototypes [194]. In the lower right section of the figure, a summary of the ADC prototype implementation steps is presented. Essential analog components are fully generated by the circuit generators and manually integrated with customized blocks and digital blocks to create completed chips for generator performance evaluation. Two prototypes have been developed using Intel 22FFL and Intel 16 process technologies, separately. Further details regarding these prototypes will be discussed in the next chapter.

Chapter 5

Generated Prototypes

5.1 Overview

Several prototypes have been fabricated in both Intel 22FFL and Intel 16 processes to demonstrate the generator-based design methodology. This chapter discusses several implementations of generated ADCs. Figure 5.1 shows the chip photos. As demonstrations of the generator-based design methodology, two tapeouts that use generated TI-SAR ADCs using the BAG2 framework are also presented.

The 4-channel and 8-channel time-interleaved SAR-VCO ADC prototypes were implemented separately using the proposed ADC generator and the BAG3 framework. Due to the less relevance of the LAYGO SAR ADC generator to the proposed ADC generator frame-

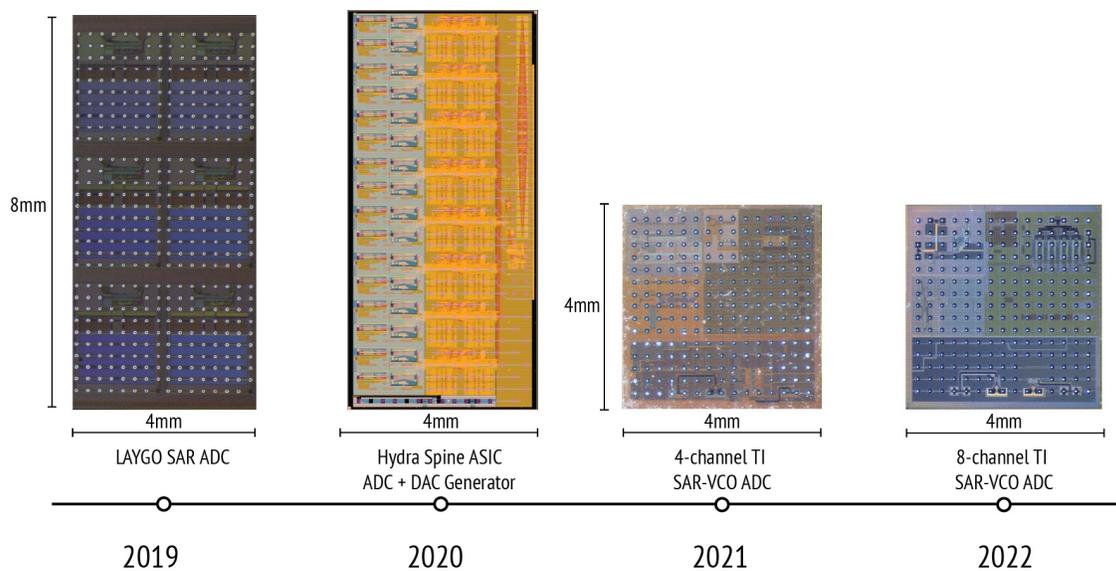


Figure 5.1: Timeline and chip micrographs of the major generator-based prototype chips.

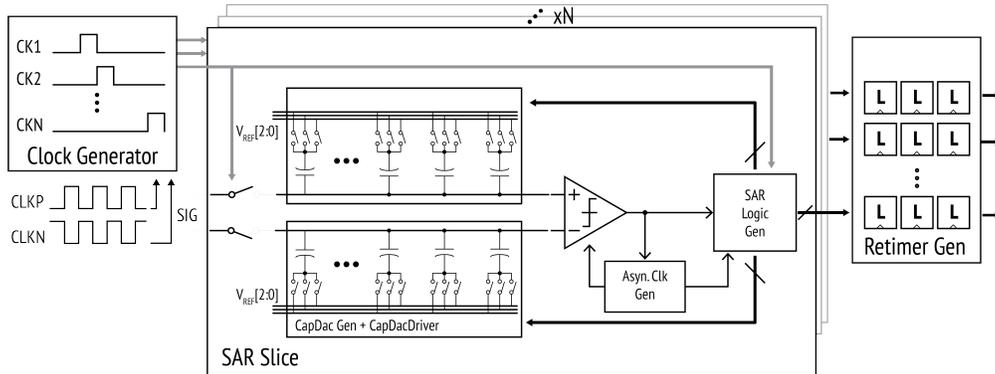


Figure 5.2: Architecture of the LAYGO time-interleaved SAR ADC generator.

work, the first two chips are briefly introduced, with a greater emphasis on the design details and measurement of the later two implementations. Section 5.2 introduces the LAYGO TI-SAR ADC generator and describes two chips used by the LAYGO ADC generator. Section 5.3 presents two prototypes implemented using the proposed ADC generator. Finally, the measurement results are presented.

5.2 LAYGO Time-Interleaved ADC Prototypes

The SAR ADC generator used in the first two chips employs the LAYGO layout generation engine to generate layouts based on given parameters automatically. LAYGO is one of the layout generator engines in the BAG2 framework that uses hand-crafted primitives and assembles layouts using Python scripts. Details about LAYGO layout generation engine can be found in Appendix A. Figure 5.2 shows an overview of the TI-SAR ADC generator architecture. This generator can provide 4-10 bits resolution and the time-interleaved architecture is used to achieve various sampling rates. The ADC comprises four main components: a multi-phase clock generation block, an array of SAR slice, a retimer at the output, and bias circuits (not shown here) which provides reference voltages for the capacitor DAC array. The blocks shown in the figure are explained as follows:

- **Clock generation:** It takes a half-rate differential clock input and uses a chain of delay cells to generate N different clock phases in parallel. These clock phases are distributed to the SAR array and trigger each SAR slice to sample the input signal sequentially.
- **ADC core:** The main component is a SAR ADC that takes an input signal, reference voltages, and a clock, and produces digital outputs. The SAR ADC consists of a capacitor DAC, a comparator, a SAR logic block, and an asynchronous clock generator. A fixed portion of time for each conversion is allocated for sampling. Moreover, it works in a self-timed, asynchronous way to convert a sampled signal at a frequency

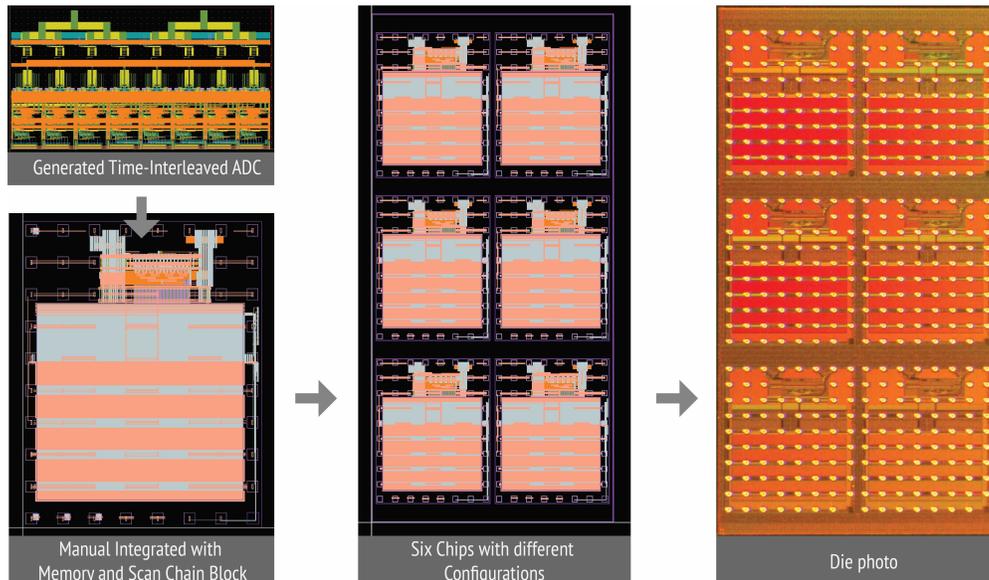


Figure 5.3: Integration steps of the LAYGO SAR ADC generator.

of $f_{s,\text{slice}} = f_s/N$, where f_s is the overall sampling frequency and N is the number of channels.

- Retimer: Because each ADC slice is timed to a different clock phase, a retimer is used to capture the digital output from each slice required. It aligns the output from each slice to the same clock so that the ADC can transfer data to the digital interface for further processing.

5.2.1 LAYGO ADC Prototype Implementations

This generator has been implemented in Intel 22FFL process with updated options that enhance the sampling rate by decoupling the charge sharing between channels. Figure 5.3 shows the steps involved in the chip integration. First, the generator generates a 9-bit 16-way time-interleaved SAR ADC. It operates at a sampling rate of 10 GS/s. A digital block includes data capture memory and a scan chain is manually integrated at the top level with the generated ADC. These blocks are used to set the configuration bits and read out the quantized results. Because of the rapid implementation using the ADC generator, six sub-chips with different configurations are generated and integrated into complete prototypes in a similar manner. Six designs have different sampler sizes, sampling strategies, and CDACs' radices. At the sub-chip level, the interface surrounding the ADC core is designed in such a way that different chips can be obtained by simply replacing the ADC core. Six sub-chips are integrated on the same die and diced after fabrication.

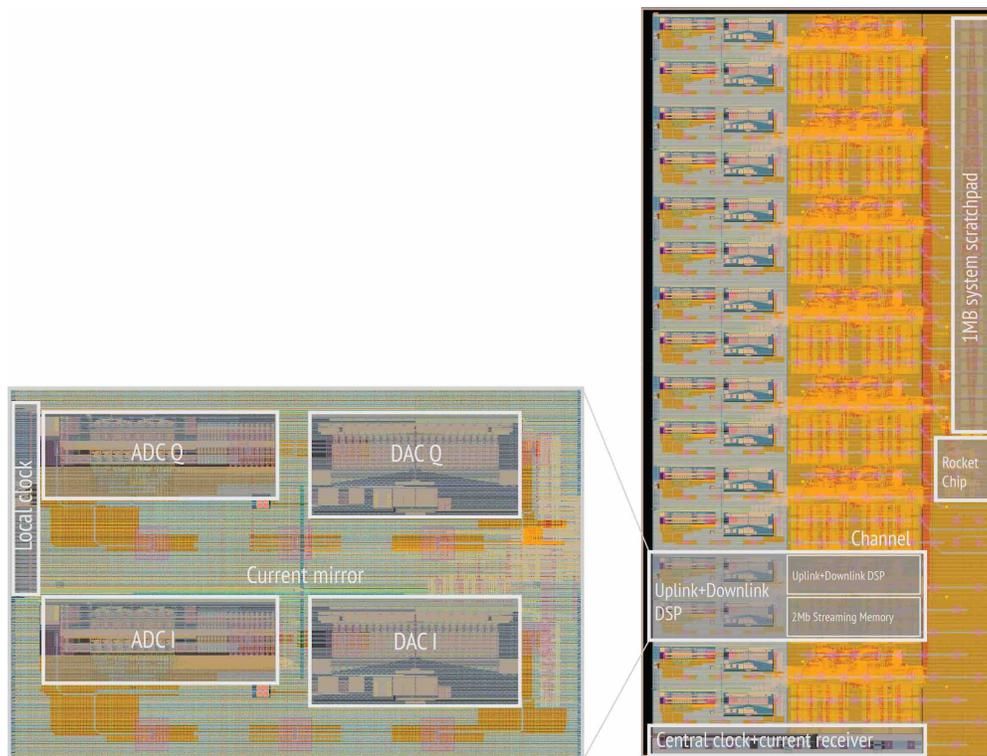


Figure 5.4: Analog generators in the Hydra spine massive MIMO chip.

5.2.2 Hydra Spine ASIC

Figure 5.4 presents an additional prototype chip of a multi-user massive multiple-input and multiple-output (MIMO) ASIC that leverages the same ADC generator. The chip has eight channels, each of which includes two ADC and DAC instances. The ADC and DAC are designed to achieve a sampling rate of 5 GS/s. Notably, the ADC generator in this configuration has only half the number of channels compared to the previously discussed prototype due to the lower speed requirement. The DAC design also utilizes the generator-based design methodology, implementing a two-way time-interleaved current steering DAC. The alternating approach facilitates the alternate steering of current from different DAC channels, thereby mitigating dynamic errors near the transition edge. The DAC generator is implemented using the XBase layout generation engine in the BAG2 framework. The same framework is also used to generate the circuits for clocking and bias current distribution. These generators have been repurposed directly from previous implementations in different processes and quickly ported from different process nodes. The generator-based design methodology enables the swift development and implementation of frequently used circuit blocks. By reusing and adapting pre-existing generators across varying process nodes, designers can save a significant amount of time and cost.

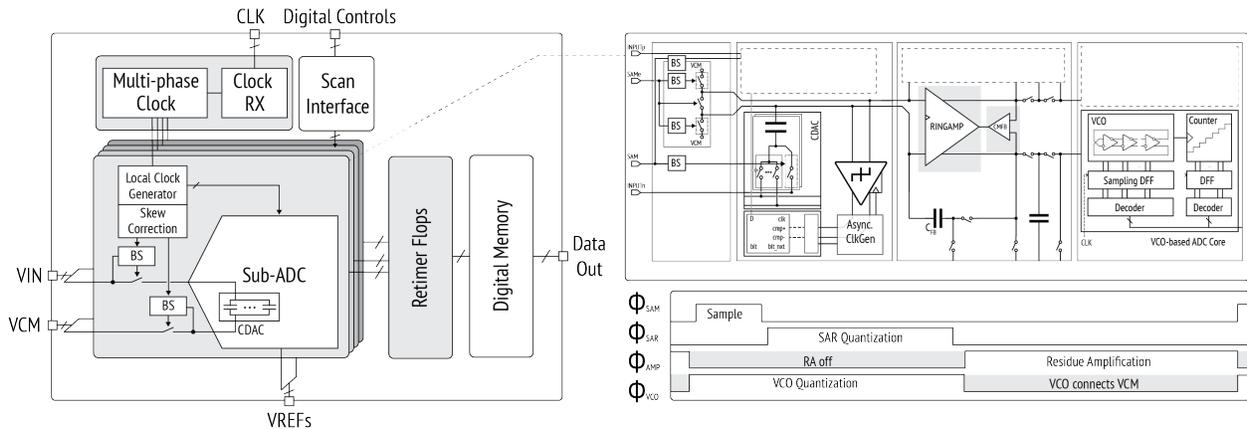


Figure 5.5: Diagram of the Time-interleaved SAR-VCO ADC prototype chip.

5.3 Time-Interleaved SAR-VCO ADC Prototypes

This section presents the implementation of the two prototypes that embody the work from previous chapters. Since the circuit topology details and design considerations were shown in the previous chapter, this chapter will only describe the generated instance configurations along with the verification results.

5.3.1 Prototype Overview

Based on the generator building blocks discussed in the previous chapter, two prototypes were implemented using Intel 22FFL and Intel 16 processes separately. The diagram in Figure 5.5 illustrates the design of both chips. The first prototype implements a 4-way time-interleaved SAR-VCO with a 6-bit SAR and a 7-bit VCO in the sub-ADC. And the second prototype implements an 8-way time-interleaved SAR-VCO ADC with sub-ADCs that have a 5-bit SAR and a 7-bit VCO. The critical building blocks have been optimized in the second prototype, and the design aims for a higher sampling rate. Due to the similarity between the two prototypes, only the design details from the latest version will be shown in the following sections. Besides the analog circuit, a scan interface and a data capture memory are implemented on the chip for the testability of both prototypes. Both versions utilize the same scan and memory design, with the only distinction being the number of bits required to accommodate the different requirements of the circuits.

5.3.2 Sub-ADC Design

The sampling capacitor for both versions is 300 fF, with a $1.6 V_{pp, diff}$ signal swing which ensures an SNDR of more than 70 dB. Both versions of the SAR ADC include 1 bit redun-

dancy in binary CDACs. The CDAC in the first prototype has capacitance weights of 32, 14, 8, 4, 2, 2, and 1, which provide 4, 2, 2, 2, 0, and 0 LSB redundancy from the second bit, respectively. Similarly, the second version uses 16, 7, 4, 2, 2, and 1 LSB for each decision. One less bit of comparison time allows for more time allocated for the amplification and the quantization of the VCO-based ADC, providing sufficient resolution for second-stage fine quantization. Two prototypes use different capacitor switching schemes: the first one adopts the MCS switching, while the second version uses a split-capacitor switching scheme. Due to the doubled number of unit capacitors in the latter switching scheme and one less bit in the second CDAC, the same size of the unit capacitor is used for both designs. The design choice was made to repurpose certain bumps to serve as the supply voltage for the VCO-based ADC, which will be presented in the measurement section of this chapter. The ring amplifier and the VCO-based ADC are mostly identical in both versions. Design optimization is achieved by incorporating more options into the generator and adjusting the generator parameters. The VCO-core in the second version adopts a differential topology with the same feedforward coupling as the first prototype to improve the minimum resolution and the common-mode rejection.

5.3.3 Clocking

The detailed timing diagram of the second prototype is shown in Figure 5.6, while the first prototype has a half divide ratio. Although the actual implementation is differential, only half of the circuit is shown for the clock receiver for simplicity. Since only the early sampling clock is critical in terms of jitter and skew, the timing diagram only shows the relevant signals in this path. At node X, the signal experiences significant delay and is susceptible to aggregated noise. First, the signal is synchronized by a latch at node Y. An enabling mechanism is used to turn on and off the channel for debugging purposes. The synchronized signal is passed through a low-jitter latch driven by the main clock. The main clock is carefully distributed from the clock receiver to each channel to ensure that the signal passes

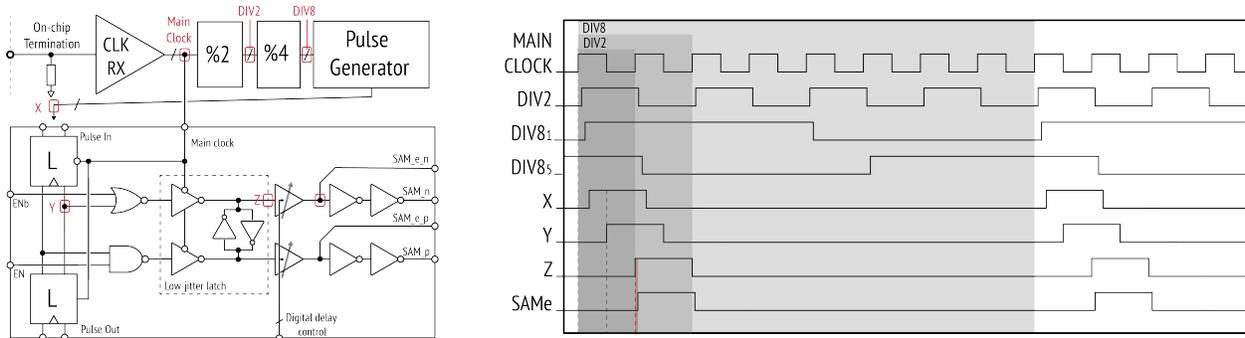


Figure 5.6: Timing diagram of clocking signals in the critical path.

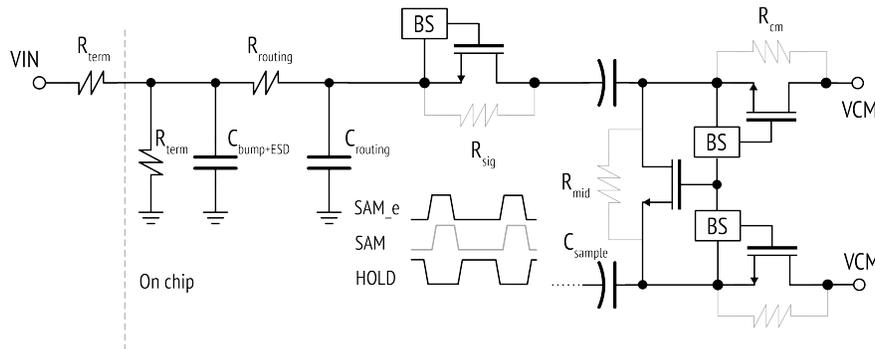


Figure 5.7: The completed diagram of the prototype's front end.

through as few stages as possible with a sufficiently sharpened edge. A skew correction block, which was presented in the previous chapter, is integrated into the final driver for tuning the critical sampling edge. Compared to the node Z signal, SAME adds a variable delay, shown in red. The primary goal of skew correction is to provide an effective solution that ensures the non-dominant nonlinearity is not dominated by skew. Therefore, no missing code has a higher priority than the linearity and monotonicity of the DAC.

5.3.4 Passive Frontend

The prototype design omits a dedicated front-end buffer; instead, a passive front end [4, 195] is used to both achieve low power consumption and a linear input network. A Y-tree distributes the signal to 8 channels, which are dummy filled along the way. Ground shielding is carefully optimized to reduce parasitic effects. Figure 5.7 shows a simplified diagram of the input signal path. Compared to standalone sampling switches, terminations, and routing resistances, ESD and routing capacitances degrade performance. Based on the simulation results, a guaranteed SFDR of more than 75 dB is achieved for the target signal bandwidth.

5.3.5 Circuit Calibration

A large number of tuning knobs are provided to ensure sufficient reconfigurability for the prototype design. The both TI errors and sub-ADC's circuits are foreground calibrated during the bringup process. This subsection summarizes all the configurable circuits on the prototype chip.

Ring amplifier biasing: Figure 5.8 shows the simulation results of the biasing voltage generation circuit. Both the NMOS and PMOS transistors in the CMOS resistor of the second-stage ring amplifier have a 7-bit resolution. The trapped-charge bias control circuit [196] provides sufficient resolution and tuning range. The ring amplifier also features a 3-bit stability calibration to ensure fast and stable settling across different corners and gain

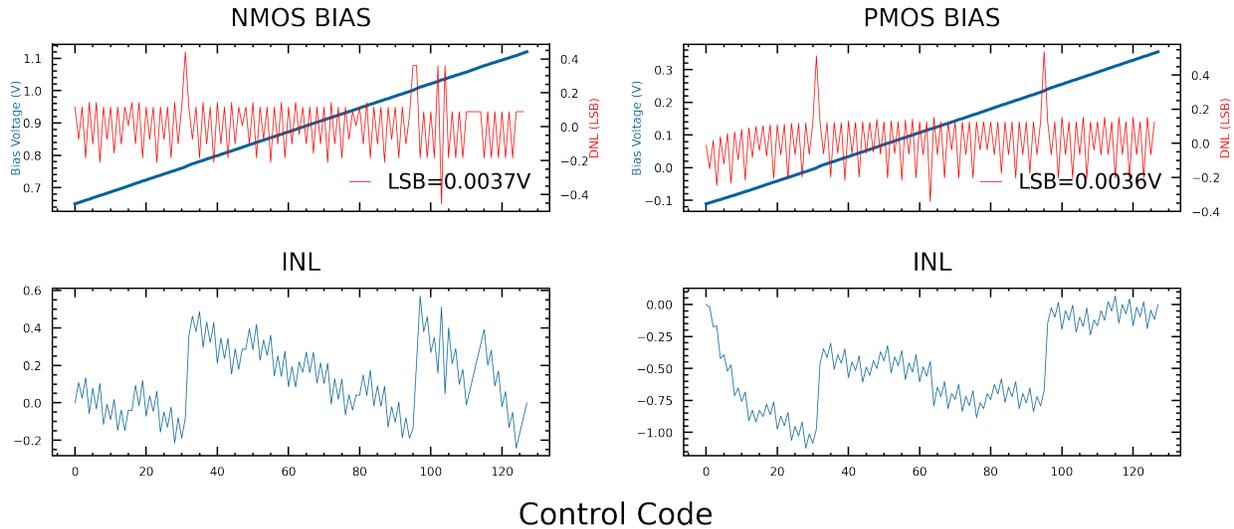


Figure 5.8: The tuning resolution and range of the ring amplifier’s biasing circuits along with its DNL and INL.

settings. The DC common-mode voltage is automatically adjusted through the closed-loop DC common-mode loop. The primary goal of the ring amplifier is to provide a sufficiently linear gain so that the residue amplification does not limit the linearity of the sub-ADC. Figure 5.8 shows the simulation results for the linearity of the ring amplifier.

Residue amplification gain: The feedback capacitance in the closed-loop amplifier is designed to be adjustable in the generator using CDACs as feedback capacitors in the loop. With the various gain settings, the effective second-stage resolution is adjusted based on the attainable frequency range of the VCO. The ring amplifier exhibits different stability at different gain settings. The worst-case stability is checked to ensure functionality with proper biasing.

Channel clock skew calibration: A single-stage floating transistor CDAC with 11 bits is used to correct the inter-channel skews. By using a combination of binary (7-bit) and thermometer code (4-bit), a 4 fs LSB is achieved. Although the tuning curve is not completely monotonic, fine tuning steps guarantee the suppression of spurs, which has a higher priority than the monotonicity. The skew calibration covers a range of delay of approximately 3 ps, which is sufficient based on the simulation of the clock distribution network. The curve showing the simulated delay against the DAC code is plotted in Figure 5.10 (a). As equation 4.36 shows, a 10 fs step size is already small enough. And the variation of the DAC LSB delay is shown in Figure 5.10 (b). The variation, which is the combination of the on-

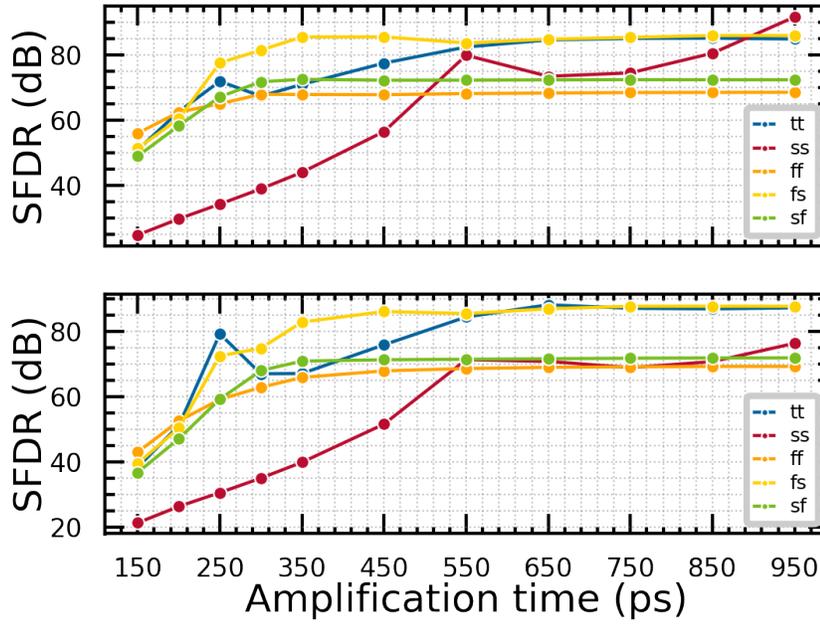


Figure 5.9: Simulated results of the ring amplifier linearity across different corners.

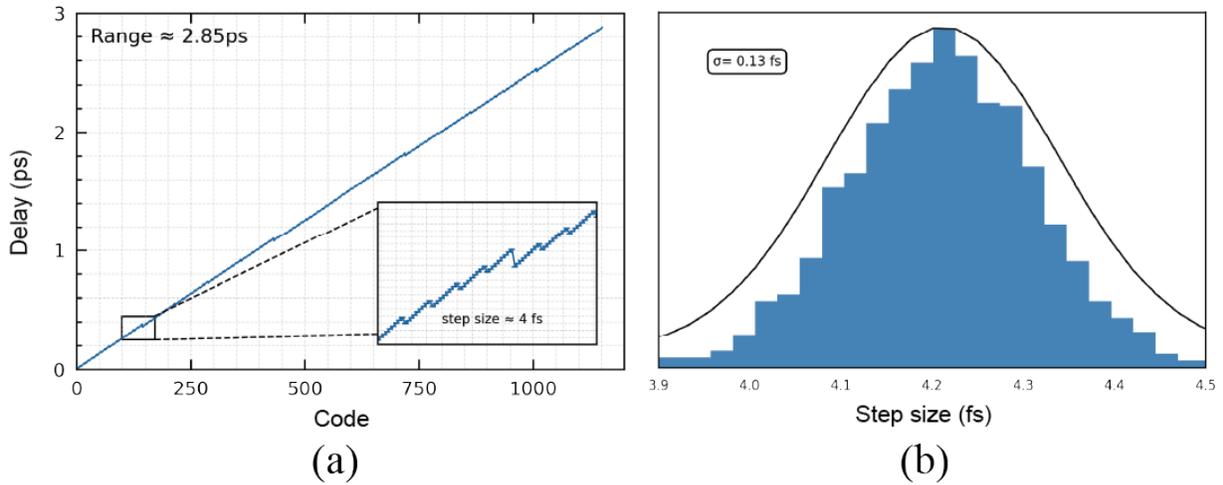


Figure 5.10: (a) Simulated tuning steps and range of the skew correction circuit. (b) Monte Carlo simulation of the LSB delay spread.

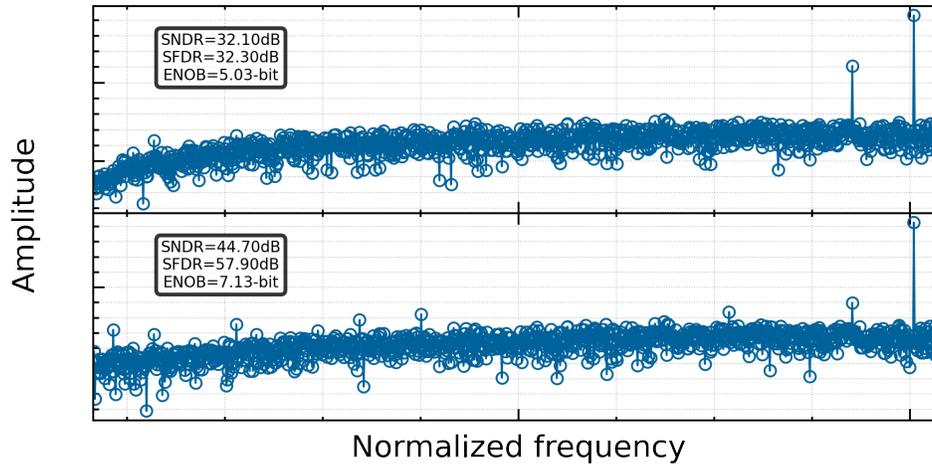


Figure 5.11: Simulation of VCO-based ADC, before (top) and after (bottom) calibration.

resistance variation of the driving stage and the variation in the equivalent LSB capacitance, is sufficiently small to guarantee the tuning resolution.

Channel offset calibration: In the sub-ADC design, the comparator and the ring amplifier are connected to the top plate of CDACs. Therefore, the equivalent channel offset can be combined and calibrated together. The channel offset calibration is performed by adjusting the bias voltage of an offset calibration pair within the comparator. Incorporating the offset calibration function is implemented as an option in each comparator generator. A differential RDAC instance is generated for each channel, providing sufficient range and resolution for the offset calibration. The RDAC's outputs are sufficiently decoupled to the ground to avoid extra noise.

Bandwidth calibration: An auxiliary CDAC, generated using the same generator as the CDAC in the SAR ADC, is placed inside each channel to provide bandwidth calibration knob among the interleaved channels.

Moreover, each channel can be turned off individually for testing purposes. The scan interface also provides testing functionality for the data capture memory, which will be discussed in the next subsection. The non-linear voltage-to-frequency curve of VCO cores are corrected off-chip. In the simulation, post-processing is applied to the raw data. Figure 5.11 shows the effect of a 32-segment piece-wise-linear calibration.

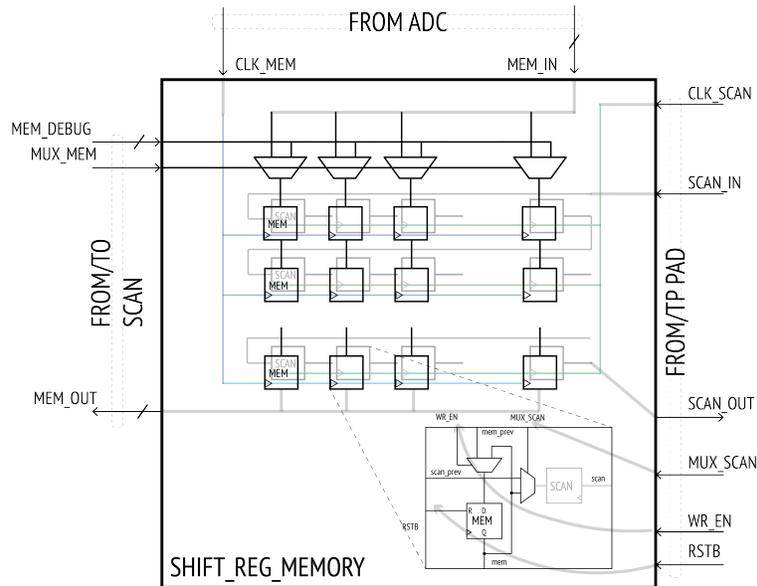


Figure 5.12: Diagram of the data capture memory.

5.3.6 Chip-Top Integration

This section describes the integration of the chip top-level. Besides the analog circuits that are fully generated from the proposed generator framework, the passive front end for both the signal and clock is manually routed from the bump to the design. A scan interface sets configuration bits and provides debugging features for the chip. A custom data capture memory is integrated into both prototypes to capture the raw data and send it off-chip for further analysis.

5.3.6.1 Scan Interface

Due to the limited number of signal bumps, the scan interface is shared by multiple designs on the same die. This necessitates a different address for each design to facilitate communication between the scan chain and the specific sub-chip. With a unique address, each sub-chip can communicate with the scan chain by first receiving address bits and then acknowledging the read/write operation once the received address bits match one of the pre-existing addresses associated with a specific design. The low-speed scan interface supports read and write operations, allowing for setting the configuration bits and the reading of state bits from the chip.

5.3.6.2 Data Capture Memory

To effectively test the prototype, transmitting the raw digital data generated by the ADC off-chip is essential for subsequent analysis. Directly transferring data to off-chip components via a parallel bus at gigahertz-per-second speeds would significantly increase costs in terms of power, area and design complexity. The decimation techniques have been employed in some ADC prototype evaluations, which, while reducing speed requirements, still necessitate a significant number of signal bumps. This approach is impractical within the constraints of the given bump map. As a result, a streamlined data capture memory has been employed to acquire the raw data and transmit it through a low-speed serial interface. Figure 5.12 illustrates the schematic of the data capture memory. Two discrete sets of registers are utilized for high-speed data capturing and low-speed readout, respectively. The lower right corner of the figure shows the diagram of a unit memory cell. When the `WR_EN` signal is asserted, the memory takes the parallel input data stream at the channel clock speed. A retimer is placed before the memory to realign the data to synchronize the digital signal with the high-speed memory clock and ensure proper timing. The data remains locked within the unit cell when the `WR_EN` signal is deactivated. Once the `MUX_SCAN` signal is activated, the data is transferred from the parallel register set to the scan register set, preparing it for sequential off-chip readout. Additionally, various debug functions have been incorporated to improve the design's testability. First, a `SCAN_IN` signal is available at the chip interface, allowing for the execution of a scan chain test to verify the functionality of the scan register set. Random data bits can be streamed to the chip with an immediate readout during testing, this function can be used to evaluate the low-speed read interface. Second, a `MUX_MEM` signal along with `MEM_DEBUG` signals, is connected to the scan chain interface. When the `MUX_MEM` signal is asserted, the memory receives patterns from the `MEM_DEBUG`, which is defined through the scan interface. This enables the verification of the parallel data capture function. Lastly, the memory output is also linked to the scan interface. Although the memory outputs are not used during normal operation, and overflowed data is discarded, connecting the final stage of the parallel register to the scan interface offers an alternative method for examining the parallel operation of the memory. In both prototypes, a 512 depth is used.

Figure 5.13 and 5.14 show the two prototypes' chip micrographs. The ADC core, scan interface, and data capture memory are outlined in both images. Both prototypes are implemented as a sub-chip within a $4\text{mm} \times 4\text{mm}$ die. A custom ball grid array (BGA) package has been employed to evaluate both prototypes. Figure 5.15 (a) shows the top and bottom sides of the custom package design, which includes three dies intended for various testing objectives of 4 different sub-chips. Die 2 is specifically allocated for evaluating prototypes of the proposed ADC generator. Both versions share the same bump map due to limited available BGA packages. However, some bumps are repurposed for different signals due to design modifications between the two prototypes. Figure 5.15 (b) and (c) show the bump maps for two prototypes. The first prototype is implemented using Intel 22FFL process, and the second prototype uses the Intel 16 process. Due to the similarity of both prototypes, details of the sub-ADC channel design are only shown in Figure 5.14. Each hybrid sub-ADC

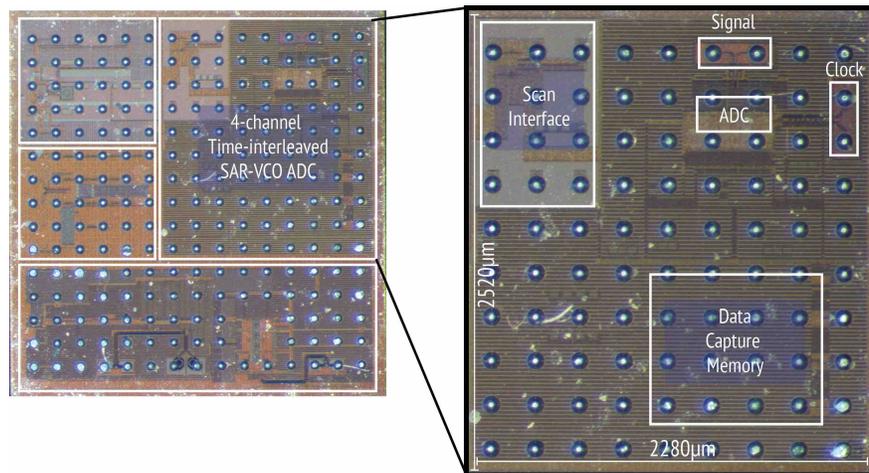


Figure 5.13: Chip micrograph of the first TI SAR-VCO prototype.

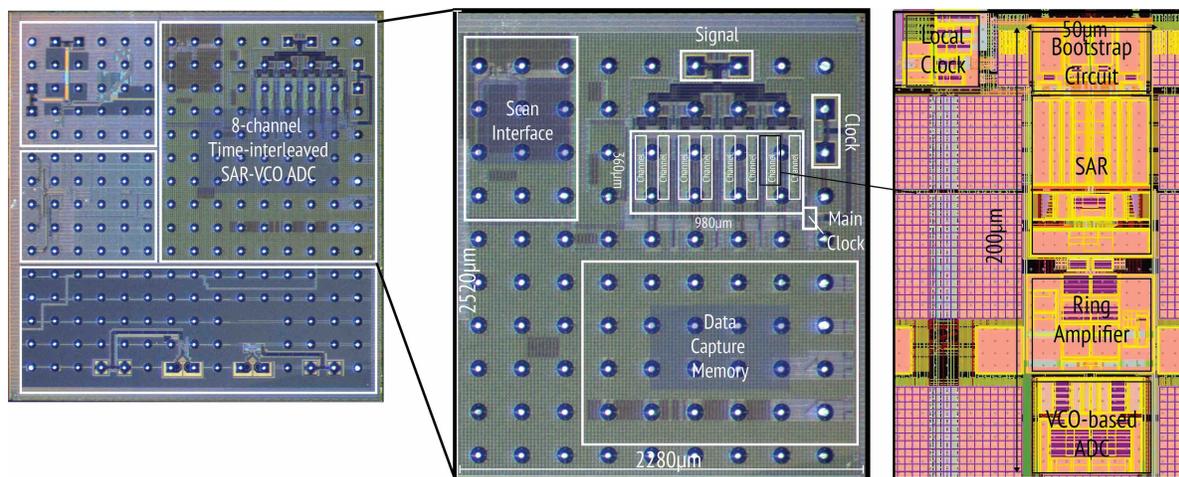


Figure 5.14: Chip micrograph of the second TI SAR-VCO prototype and sub-ADC layout details.

is implemented symmetrically in the layout with optimizations at the critical nodes to maximize the speed and reduce parasitics. The bootstrapped signal generator is placed at the top, followed by the SAR, ring amplifier, and then VCO-based ADC. The empty region around the ADC is filled with decoupling capacitors made from MOS, MOM, and MIM capacitors. These capacitors provide sufficient decoupling for critical voltages, especially for reference voltages.

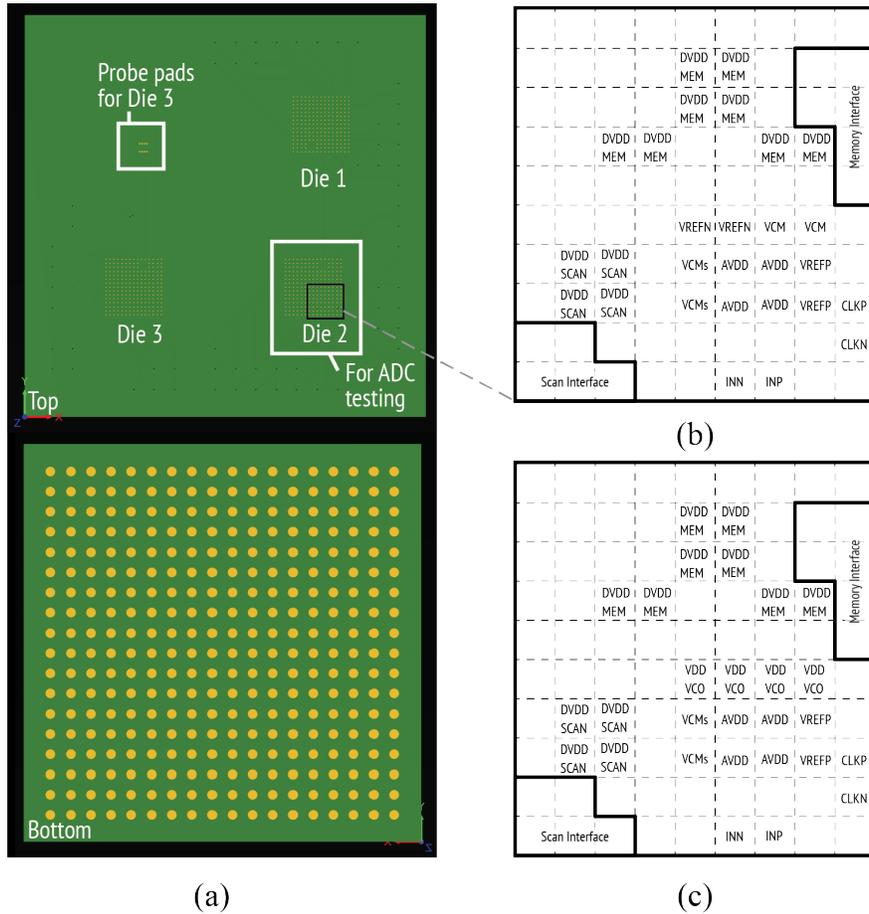


Figure 5.15: (a) Layouts of the custom BGA package design and bump maps of (b) the first and (c) the second prototypes.

5.4 Measurement

This section presents the measurement of the time-interleaved SAR-VCO ADC prototypes. Subsection 5.4.1 describes the design of the evaluation boards and the measurement setup. And the measurement results and performance summary of the prototype are presented.

5.4.1 Measurement Setup

Figure 5.16 illustrates the diagram of board designs for prototype evaluation. Due to constraints in package availability, two separate boards have been designed to enhance board yield, simplify the bring-up process, and facilitate the testing of multiple chips. On the left side of the figure, the chip board incorporates the BGA package, which includes three dies for different measurement purposes. The chip board also has SMA connectors for high-speed

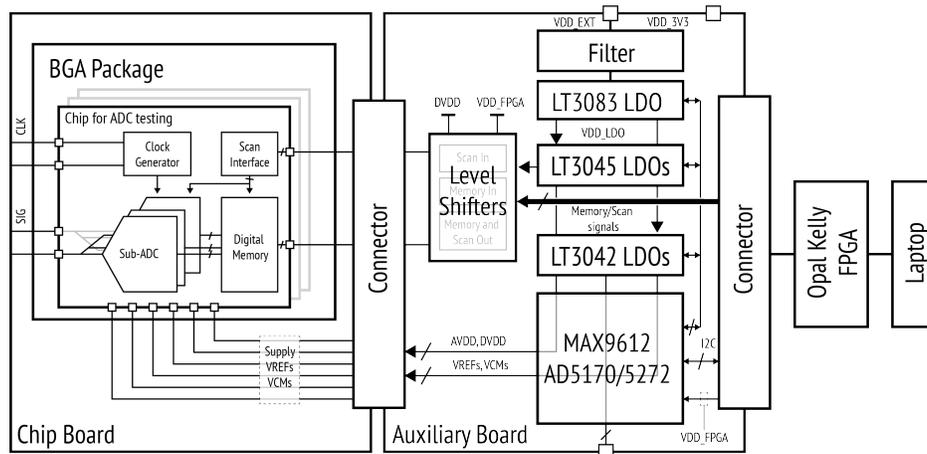


Figure 5.16: Diagram of the board designs including a chip board for BGA package and an auxiliary board for the supply regulation and low-speed digital signal communication.

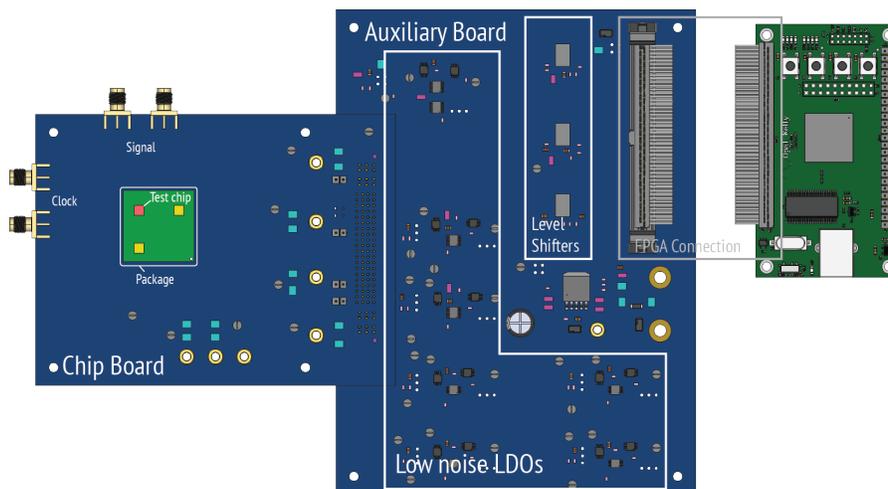


Figure 5.17: Top view of the 3D model of evaluation boards connecting to an FPGA.

signal and clock inputs as well as a connector for low-speed signals and DC supplies that are connected to the auxiliary board. Clock and signal traces on the chip board are ground shielded and designed to provide a $100\ \Omega$ differential impedance. The separate auxiliary board on the right side of Figure 5.16 includes off-chip low dropout regulators (LDOs) that are responsible for generating DC supplies for different domains and reference voltages. The output voltages of LDOs are programmed via the potentiometers (AD5170 and AD5272),

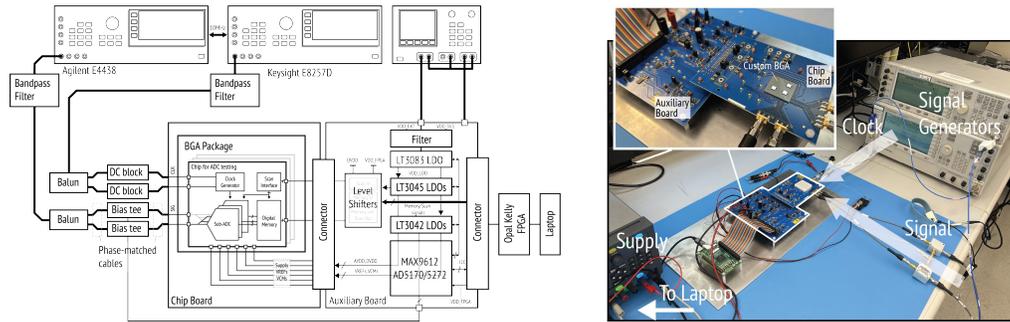


Figure 5.18: Diagram and photograph of the measurement setup.

which are connected to the SET port of the LDOs and communicate through the I2C interface. Each LDO is equipped with a MAX9612 current sensor, which is connected to the ILIM port of the LDOs and used for monitoring power consumption. The low-noise regulators LT3042 and LT3045 are used for generating reference and DC power supplies, respectively. A two-step supply regulation is used, the LT3083 is chosen to supply the regulators with low noise, and a low-pass filter with a ferrite bead is inserted between the external supply and the input of LT3083 to further improve the supply quality.

For the scan interface and data capture memory, low-speed digital signals are accessed using FPGA, which establishes a connection to a laptop and communicates with measurement code using a Python interface. The signals in the 3.3 V supply domain from the FPGA pass through level shifters and transition into the chip's digital supply domain (DVDD). All of the level shifters are powered by the 3.3 V FPGA supply and the DVDD supply is generated from LDOs on the auxiliary board. Due to the repurpose of BGA bumps in the second prototype, the chip board can be configured to allow both prototypes to share the same board design. Figure 5.17 shows the 3D model of the board design, including the FPGA on the right side, providing a more detailed view of the evaluation board. The chip in red on the upper left corner of the BGA package is used for ADC testing, while the other two are for testing other designs on the same chip. Various backup connectors are included on both boards for debugging purposes. The completed measurement setup diagram and the photo are shown in Figure 5.18. Both the clock and signal for the prototype chip are generated from low phase noise signal generators. Measuring the performance of the ADC using spectrum analysis requires the input signal frequency to be coprime with the clock frequency. The clock and signal generator are locked in frequency using a 10 MHz reference port. Both signals are converted to differential signals using broadband baluns (BAL0006, BAL0010) with low amplitude and phase mismatch. The band-pass filters are used to further improve the signal's spectral purity of both the signal and the clock. Finally, the signal and clock are AC-coupled to the test chip by a bias-tee (ZFBT-4R2G) and DC-block (BLK-89-S), respectively. Phase-matching cables are used to ensure the quality of differential signals.

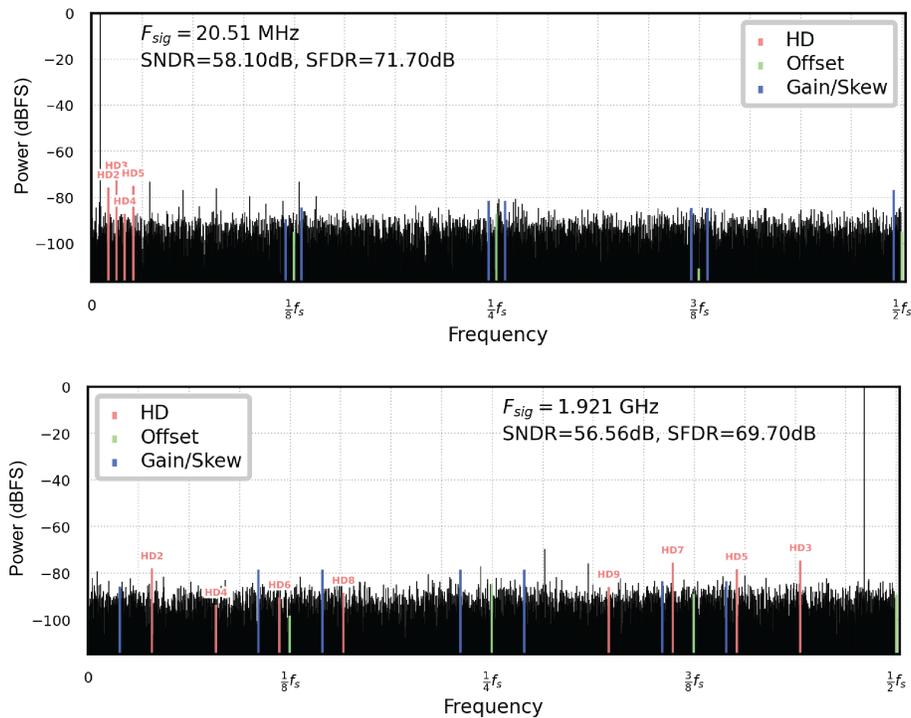


Figure 5.19: Measured output spectrum after calibration, sampling at 4 GS/s for low frequency (top) and Nyquist frequency (bottom) input signals.

5.5 Measurement Results

The ADC is foreground calibrated using both off-chip correction and tuning the on-chip calibration circuit. Each channel in the test chip is first calibrated by feeding a full-scale low-frequency sinusoidal wave. The first stage SAR ADC's functionality is ensured and the CDAC weights are approximated. Next, a small amplitude sinusoidal wave is used for the calibration of the residue amplifier and the VCO-based ADC. Then the chip is loaded with the resulting comparator offset, ring amplifier biasing, and stability control transistor settings through the scan interface. The CDAC weights, channel gain, and offset are logged and corrected through post-processing. Feeding low and high-frequency signals helps identify interleaving spurs that originate from different mechanisms. The bandwidth and timing skews are corrected on-chip as well. First, the timing skew is calibrated using the on-chip floating transistor DACs. Since the bandwidth calibration also manifests itself at the same location as gain and skew mismatch. The objective of the bandwidth calibration is set to bring the skew settings at three different frequencies close to each other. In this way, the effect of bandwidth mismatch is largely canceled. Figure 5.20 show the spectrum before and after the calibrations of time-interleaving errors.

Figure 5.19 shows the measured output spectrum after calibration at a sampling speed of

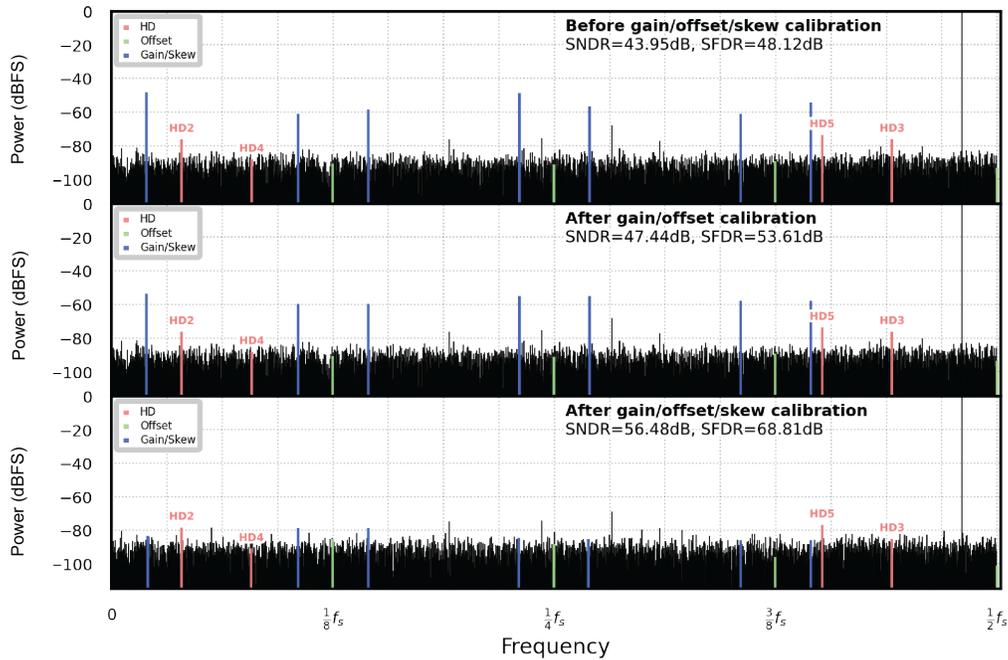


Figure 5.20: Measured spectrum before and after time-interleaved error calibrations.

4 GS/s and an input signal of 20 MHz. The third-order harmonics dominate the nonlinearity. The bottom plot in Figure 5.19 shows the spectrum near the Nyquist frequency. All remaining sub-ADC/interleaving-related spurs are suppressed due to the calibration circuit. Figure 5.21 shows the SNDR and SFDR as a function of the input frequency for the 4 GS/s sampling rate for three different samples. The peak SFDR is about 72 dB and drops to around 60 dB at a 4 GHz input. The SNDR at low frequencies is approximately 59 dB. However, it drops to 56 dB at the Nyquist frequency and further drops to 50 dB when the input signal is at 4 GHz. Both the SNDR and SFDR curves stay relatively flat up to 3.5 GHz, and after that, both start dropping significantly. When the input frequency is increased towards the Nyquist frequency and beyond, several factors contribute to the degradation of the SNDR. These factors include frequency-dependent residual interleaving errors, harmonic distortion, the increasing effect of signal and clock jitter, and the loss of signal gain. Figure 5.22 shows the SNDR/SFDR plotted against input amplitude and frequency for both the interleaved ADC and the single ADC channel. Figure 5.23 shows the SNDR and SFDR as a function of the input amplitude. The measurements were taken using a low-frequency input signal and a Nyquist-frequency input signal, with a sampling rate of 4 GHz. The dynamic range of the SAR+VCO and SAR-only modes are measured and the result is presented in 5.23.

For the completeness of measurement, the static properties of a single ADC sample are measured. The measurements of INL and DNL of the VCO-based ADC are different

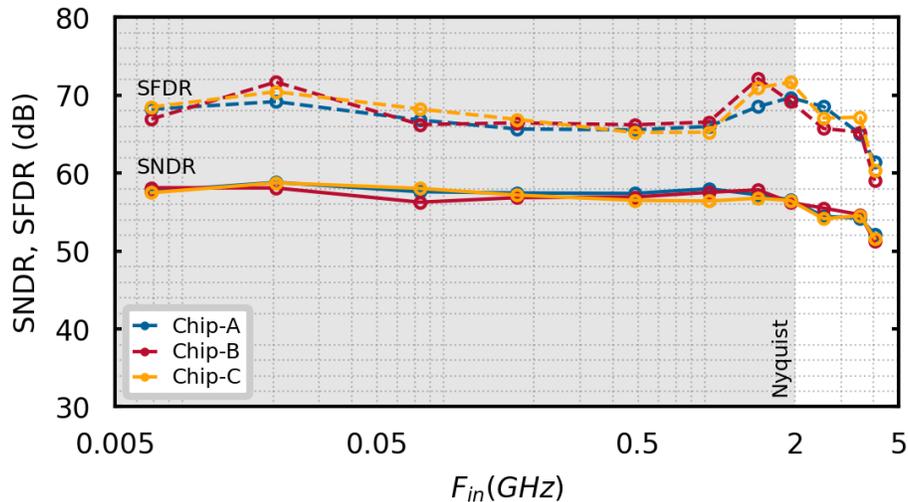


Figure 5.21: Measured SFDR and SNDR versus input frequency sampling at 4 GS/s for three ADC samples.

from those of traditional voltage-domain ADCs. These measurements can be obtained by analyzing the code density at each code level. As described in section 4.4.1, the mismatches between delay cells can increase the level of quantization noise. The DNL of the VCO-based ADC can be defined as

$$DNL[n] = \frac{t_{d,n}}{t_{avg}} - 1, \quad (5.1)$$

where $t_{avg} = \frac{1}{N} \sum_{i=1}^N t_i$. The INL of the RO is defined as

$$INL[n] = \sum_{i=1}^n DNL[i]. \quad (5.2)$$

The INL and DNL of each oscillator are measured statistically. First, a large amount of unprocessed raw data is collected from each channel. Since the design is limited by thermal noise, the input at each VCO when the differential signal is shorted together can be approximated as a random signal. With perfect matching between stages, the frequency of appearance for each phase should be very close in the collected data. And the mismatches are assumed to be the only reason causing a difference in the number of times each phase appears. The DNL of the oscillator is approximated using this method, and the INL is calculated. Figure 5.24 shows the measured INL and DNL of the ring oscillators in each channel. The total power of the measured sample is around 124.6 mW. Figure 5.26 shows the power breakdown when the ADC is sampling at 4 GS/s. The ADC channels collectively account for approximately 43%. The clocking circuit consumes approximately 32% of the power, while the calibration and supporting circuits take the remaining portion. Figures 5.25

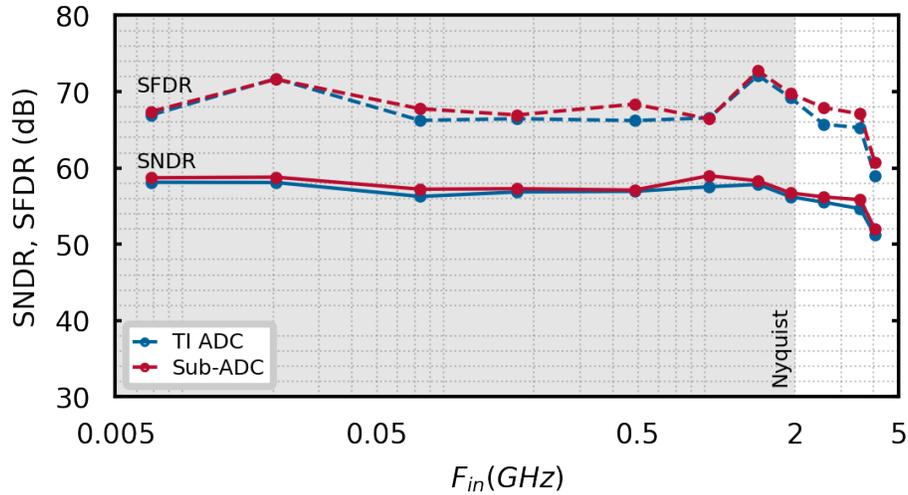


Figure 5.22: Measured SFDR and SNDR versus input frequency for a single channel (samples at 500 MS/s) and the time-interleaved channels (samples at 4 GS/s).

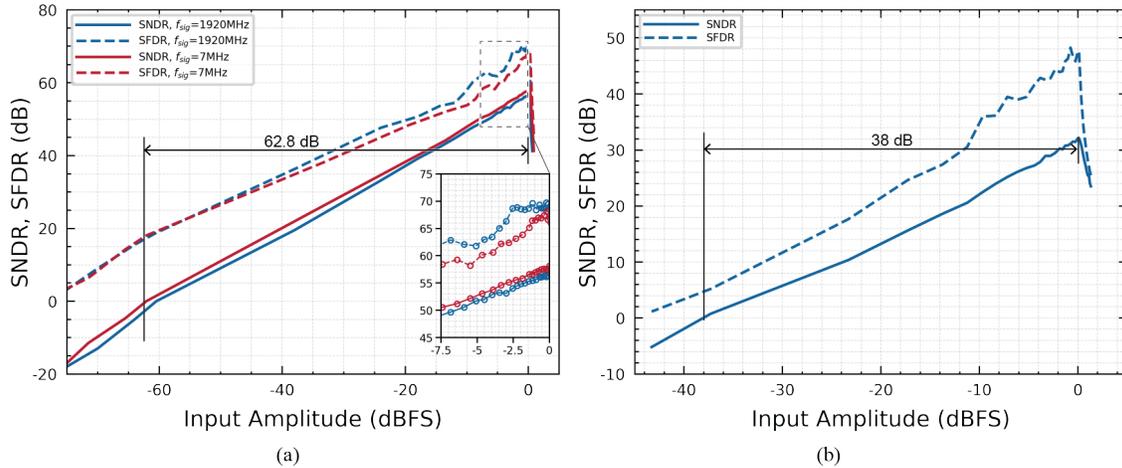


Figure 5.23: Measured SNDR and SFDR as a function of the input amplitude for (a) the SAR+VCO ADC and (b) the SAR ADC only.

show the figure-of-merit (FOM) for the prototype chip (indicated by the red star) and other similar works (indicated by the red circles). The generator prototypes achieve a Walden FOM (FOM_W) of 60.5 fJ/conv.-step and a Schreier FOM (FOM_S) of 158.4 dB. The performance summary of the prototype is shown in Figure 5.27, along with a comparison to the state-of-the-art works. [3, 22, 53, 92, 197, 198, 199, 196, 195, 200, 201, 202]

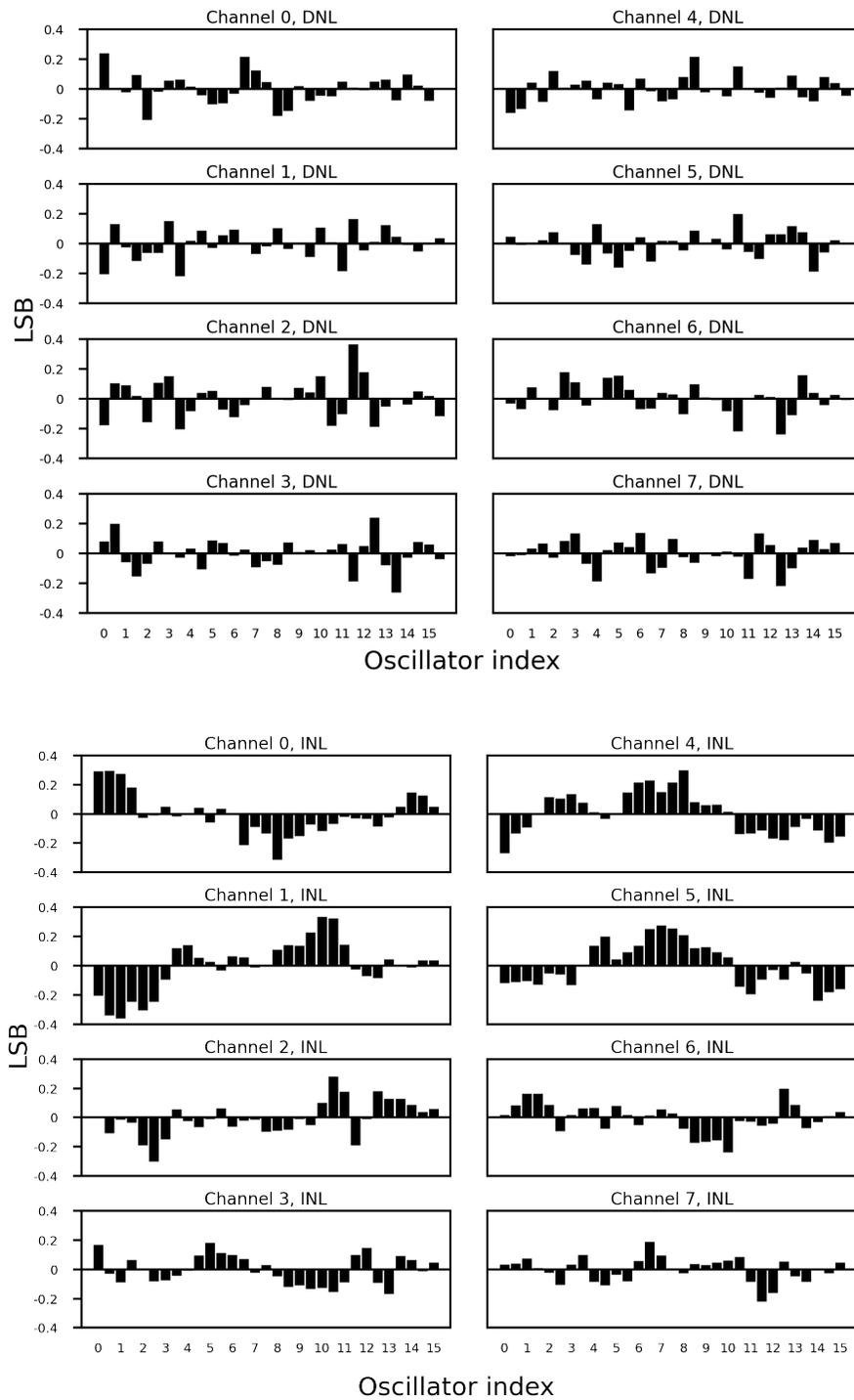


Figure 5.24: Measured DNL and INL of ring oscillators in 8 channels.

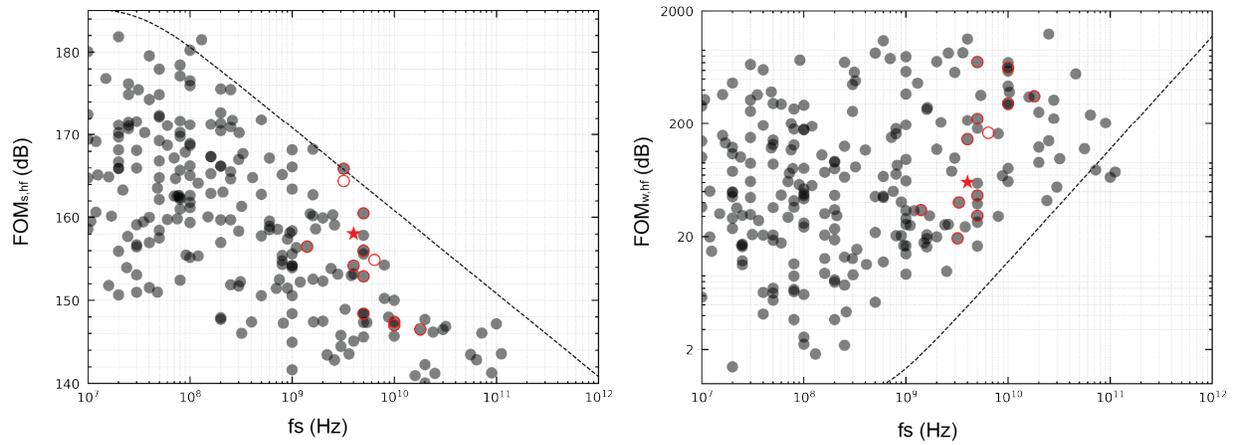


Figure 5.25: The figure of merit of this work compared to other similar designs.

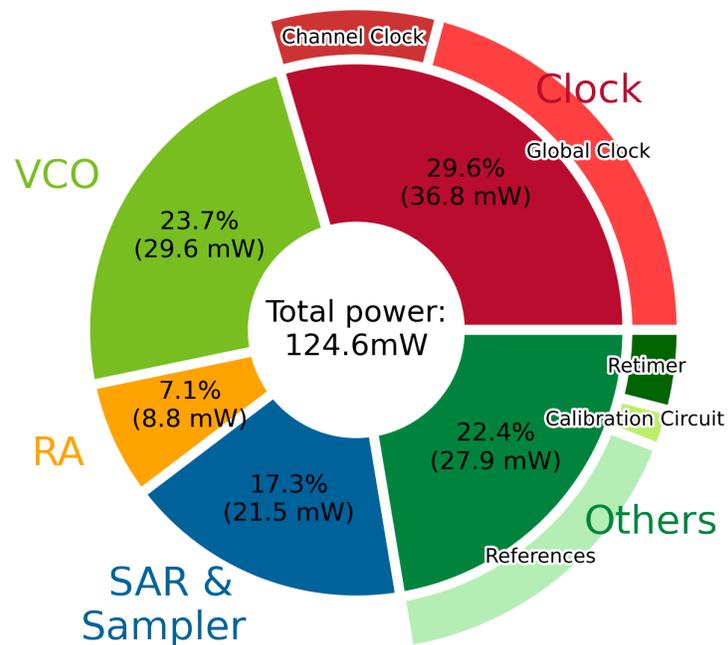


Figure 5.26: Power breakdown.

	This Work	Ricci VLSI '23	Moon VLSI '22	Chang VLSI '22	Ali ISSCC '20	Ramkaj ISSCC '19	Hershberg ISSCC '19	Baert ISSCC '19	Nam JSSCC '18	Vaz VLSI '18	Devarajan ISSCC '17	Ali ISSCC '16	Wu ISSCC '16
Technology	16nm	28nm	5nm	28nm SOI	16nm	28nm	16nm	28nm	65nm	16nm	28nm	16nm	28nm
Resolution (bit)	12	11	12	10	12-14	12	13	8	12	13	12	14	13
Supply (V)	0.9	1.0	0.85	1.0	1.0/1.8	1.0	0.85/1.8	1.0/0.85	1.1/1.2/2.5	0.9/1.8	1.0/2.0	0.9/1.8/2.5	0.8/1.8
Architecture	TI SAR-VCO	TI SAR	TI Pipe-SAR	TI SAR	TI Pipe	TI Pipe-SAR	TI Pipe	TI VCO	TI SAR	TI Pipe-SAR	TI Pipe	TI Pipe	TI Pipe
Interleaving Factor	8x	8x	16x	8x	2/4/8x	8x	4x	8x	8/16/32x	8x	8x	2x	4x
Sample Rate (GS/s)	4.0	2.0	10.0	1.0	18.0/10.0	5.0	3.2	5.0	1.6/3.2/6.4	5.0	10.0	5.0	4.0
SNDR @Low freq (dB)	58.8	57.2		52		62.4	62.9		67.3 @6.4GS/s	60	59	63	60
SFDR @Low freq (dB)	69.2	62.2				75.2	80.3		82.5 @6.4GS/s	75	69	80	75
SNDR @High freq (dB)	56.5	57.3	48.0	48.0	52/48/55/53 @ 8/4GHz	58.5	61.7	45.2	65.3/64.9/ 58.4	57.0	55.0	58.0	56.0
SFDR @High freq (dB)	68.2	69.9	56.0	59.5	56/54/65/65 @ 8/4GHz	65.4	73.3	57.1	72.1/71.5/ 68.2	61.9	64.0	70.0	68.0
Power (mW)	124.6	58.6	625.0	9.9	1300/850	158.6	61.3	22.7	40.7/120.6/ 225.0	641.0	2900.0	2300.0	300.0
FOM _{Walden} (f1/conv.-step)	60.5	48.9	305.0	33.0	222/351.9/ 163.3/194.1	46.1	19.3	30.5	16.9/39.1/ 165.5	221.6	631.2	708.7	145.5
FOM _{Schreier} (dB)	158.4	159.6	147.0	157.0	150.4/146.4/ 153.0/151.5	160.5	165.9	155.6	169.2/164.4/ 54.9	152.9	147.4	148.4	154.2

Figure 5.27: Performance summary and comparison with state-of-the-art ADC.

Chapter 6

Conclusion

This research employs a generator-based methodology to investigate the design of an energy-efficient, scalable, and high-speed ADC. The achievement of process portability and automated circuit generation has been realized by using the BAG3 framework. By combining process-agnostic generator scripts with process-specific primitives, the proposed ADC can be quickly ported to various processes. Different ADC architectures are explored in both the voltage domain and the time domain. Hybrid and time-interleaved techniques are utilized to accommodate various target specifications. The concepts of high-speed ADC design using a generator-based methodology are demonstrated on various prototype chips. The latest prototype implements an 8-way interleaved SAR-VCO ADC operating at a 4 GHz sampling rate with a resolution greater than 9 ENOB.

6.1 Key Accomplishments

The main achievements of this research are as follows:

- The development of the proposed ADC generator involves the collaborative development and maintenance of a generator framework. Different features that facilitate the automation of mixed-signal circuit design are incorporated into the generator framework. As the generator has undergone testing in multiple processes, libraries of process primitives have been established and maintained.
- We have explored architectural alternatives for high-speed ADCs and techniques that enhance conversion and improve sampling linearity. The focus of our circuit design exploration has been on architectures and topologies that can be easily adapted to scaled processes, enabling the development of generators that can be used across different processes.
- We have developed a comprehensive library of components for time-interleaved hybrid ADC generators. The peak performance of the generated instances is aimed at high-speed, high-resolution ADCs, which are ideally suited for direct RF communication

systems. Our optimized building blocks also cater to diverse use cases with lower resolutions or sampling rates. The generator can be easily expanded by incorporating high-level scripts and assembling the necessary building blocks or supporting circuits for various applications.

- As a demonstration of the proposed architecture and design methodology, we have implemented several prototype chips, including integrating analog generators into a complete system-on-chip and creating prototypes for performance tests of generated instances. The most recent generator prototypes demonstrate performance on par with state-of-the-art manually designed circuits.

6.2 Future Work

The work in this dissertation reviews the architectural benefits and challenges associated with different ADC circuit topologies and outlines the trajectory of ADC architectures. It presents a framework for understanding the generator-based design methodology, particularly with regard to ADC design. An automated process-portable ADC generator framework is built to generate various designs. A framework of quick ADC implementations and architecture evaluations is built. Based on these, several directions can be pursued to further augment this work:

- The prototype chips implement foreground calibration, a complex and time-intensive process due to limited access to intermediate information. In addition to circuit design, on-chip calibration can be integrated with the generated instances, necessitating the inclusion of a more advanced debugging interface.
- Given the time constraints, this thesis focuses on developing generator prototypes that implement one specific architecture time-interleaved SAR-VCO architecture. Nevertheless, the current generator framework can expedite further exploration of other ADC architectures of interest. Other time-domain topologies can be integrated with existing blocks. Rapid implementation of alternative architectures is achievable by utilizing existing building blocks. Possible implementations using the existing generators include a standalone VCO-based ADC with a bootstrapped sampler, pipelined SAR ADC, or pipelined ADC employing a ring amplifier-based switch capacitor circuit.
- While the proposed ADC generator primarily focuses on analog and mixed-signal design, the design process still heavily relies on the designer's interaction with generators and design tools. By integrating the design and measurement APIs offered in the BAG framework with a design algorithm, it is possible to create an automated solution for selecting generator circuit parameters that fulfill specified target specifications. This approach enables the automation of closed-loop ADC design. Furthermore, a machine learning-based circuit design that considers layout effects can be developed to achieve a comprehensive end-to-end design.

Bibliography

- [1] J. Mitola. “The Software Radio Architecture”. In: *IEEE Communications Magazine* 33.5 (May 1995), pp. 26–38.
- [2] Asad A. Abidi. “The Path to the Software-Defined Radio Receiver”. In: *IEEE Journal of Solid-State Circuits* 42.5 (May 2007), pp. 954–966.
- [3] Siddharth Devarajan et al. “A 12-b 10-GS/s Interleaved Pipeline ADC in 28-Nm CMOS Technology”. In: *IEEE Journal of Solid-State Circuits* 52.12 (Dec. 2017), pp. 3204–3218.
- [4] Athanasios T. Ramkaj et al. “A 5-GS/s 158.6-mW 9.4-ENOB Passive-Sampling Time-Interleaved Three-Stage Pipelined-SAR ADC With Analog-Digital Corrections in 28-Nm CMOS”. In: *IEEE Journal of Solid-State Circuits* (2020), pp. 1–12.
- [5] Shiva Kiran et al. “Digital Equalization With ADC-Based Receivers: Two Important Roles Played by Digital Signal Processing in Designing Analog-to-Digital-Converter-Based Wireline Communication Receivers”. In: *IEEE Microwave Magazine* 20.5 (May 2019), pp. 62–79.
- [6] Samuel Palermo et al. “CMOS ADC-based Receivers for High-Speed Electrical and Optical Links”. In: *IEEE Communications Magazine* 54.10 (Oct. 2016), pp. 168–175.
- [7] Ahmad Khairi et al. “A 1.41-pJ/b 224-Gb/s PAM4 6-Bit ADC-Based SerDes Receiver With Hybrid AFE Capable of Supporting Long Reach Channels”. In: *IEEE Journal of Solid-State Circuits* 58.1 (Jan. 2023), pp. 8–18.
- [8] Yoel Krupnik et al. “112-Gb/s PAM4 ADC-Based SERDES Receiver With Resonant AFE for Long-Reach Channels”. In: *IEEE Journal of Solid-State Circuits* 55.4 (Apr. 2020), pp. 1077–1085.
- [9] Pui-In Mak and Rui P Martins. “High-/Mixed-Voltage RF and Analog CMOS Circuits Come of Age”. In: *IEEE Circuits and Systems Magazine* 10.4 (2010), pp. 27–39.
- [10] John Ferguson. *Assessing the True Cost of Process Node Transitions [online]*. URL: <https://www.techdesignforums.com/practice/technique/assessing-the-true-cost-of-node-transitions/>.
- [11] Eric Chang et al. “BAG2: A Process-Portable Framework for Generator-Based AMS Circuit Design”. In: *2018 IEEE Custom Integrated Circuits Conference (CICC)*. San Diego, CA: IEEE, Apr. 2018.

- [12] Erik Sall and Mark Vesterbacka. “Thermometer-to-Binary Decoders for Flash Analog-to-Digital Converters”. In: *2007 18th European Conference on Circuit Theory and Design*. Sevilla, Spain: IEEE, Aug. 2007, pp. 240–243.
- [13] Jong-In Kim et al. “A 6-b 4.1-GS/s Flash ADC With Time-Domain Latch Interpolation in 90-Nm CMOS”. In: *IEEE Journal of Solid-State Circuits* 48.6 (June 2013), pp. 1429–1441.
- [14] Yun-Shiang Shu. “A 6b 3GS/s 11mW Fully Dynamic Flash ADC in 40nm CMOS with Reduced Number of Comparators”. In: *2012 Symposium on VLSI Circuits (VLSIC)*. Honolulu, HI, USA: IEEE, June 2012, pp. 26–27.
- [15] Robert C. Taft et al. “A 1.8 V 1.0 GS/s 10b Self-Calibrating Unified-Folding-Interpolating ADC With 9.1 ENOB at Nyquist Frequency”. In: *IEEE Journal of Solid-State Circuits* 44.12 (Dec. 2009), pp. 3294–3304.
- [16] Shuo-Wei Michael Chen and Robert W. Brodersen. “A 6-Bit 600-MS/s 5.3-mW Asynchronous ADC in 0.13- μ m CMOS”. In: *IEEE Journal of Solid-State Circuits* 41.12 (Dec. 2006), pp. 2669–2680.
- [17] Lukas Kull et al. “A 3.1 mW 8b 1.2 GS/s Single-Channel Asynchronous SAR ADC With Alternate Comparators for Enhanced Speed in 32 Nm Digital SOI CMOS”. In: *IEEE Journal of Solid-State Circuits* 48.12 (Dec. 2013), pp. 3049–3058.
- [18] Long Chen et al. “A 0.95-mW 6-b 700-MS/s Single-Channel Loop-Unrolled SAR ADC in 40-Nm CMOS”. In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 64.3 (Mar. 2017), pp. 244–248.
- [19] S.H. Lewis. “Optimizing the Stage Resolution in Pipelined, Multistage, Analog-to-Digital Converters for Video-Rate Applications”. In: *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 39.8 (Aug. 1992), pp. 516–523.
- [20] D.W. Cline and P.R. Gray. “A Power Optimized 13-b 5 Msamples/s Pipelined Analog-to-Digital Converter in 1.2 μ m CMOS”. In: *IEEE Journal of Solid-State Circuits* 31.3 (Mar. 1996), pp. 294–303.
- [21] B. Murmann and B.E. Boser. “A 12-Bit 75-MS/s Pipelined ADC Using Open-Loop Residue Amplification”. In: *IEEE Journal of Solid-State Circuits* 38.12 (Dec. 2003), pp. 2040–2050.
- [22] Ahmed M. A. Ali et al. “16.1 A 12b 18GS/s RF Sampling ADC with an Integrated Wideband Track-and-Hold Amplifier and Background Calibration”. In: *2020 IEEE International Solid-State Circuits Conference - (ISSCC)*. San Francisco, CA, USA: IEEE, Feb. 2020, pp. 250–252.
- [23] Boris Murmann. “The Successive Approximation Register ADC: A Versatile Building Block for Ultra-Low- Power to Ultra-High-Speed Applications”. In: *IEEE Communications Magazine* 54.4 (Apr. 2016), pp. 78–83.

- [24] Chun-Cheng Liu et al. “A 10b 100MS/s 1.13mW SAR ADC with Binary-Scaled Error Compensation”. In: *2010 IEEE International Solid-State Circuits Conference - (ISSCC)*. San Francisco, CA, USA: IEEE, Feb. 2010, pp. 386–387.
- [25] Chun-Cheng Liu. “27.4 A 0.35mW 12b 100MS/s SAR-assisted Digital Slope ADC in 28nm CMOS”. In: *2016 IEEE International Solid-State Circuits Conference (ISSCC)*. Jan. 2016, pp. 462–463.
- [26] Lukas Kull et al. “22.1 A 90GS/s 8b 667mW 64x Interleaved SAR ADC in 32nm Digital SOI CMOS”. In: *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*. San Francisco, CA, USA: IEEE, Feb. 2014, pp. 378–379.
- [27] Cristiano Niclass et al. “A 128×128 Single-Photon Image Sensor With Column-Level 10-Bit Time-to-Digital Converter Array”. In: *IEEE Journal of Solid-State Circuits* 43.12 (Dec. 2008), pp. 2977–2989.
- [28] Yahya M. Tousi and Ehsan Afshari. “A Miniature 2 mW 4 Bit 1.2 GS/s Delay-Line-Based ADC in 65 Nm CMOS”. In: *IEEE Journal of Solid-State Circuits* 46.10 (Oct. 2011), pp. 2312–2325.
- [29] KwangSeok Kim, WonSik Yu, and SeongHwan Cho. “A 9 Bit, 1.12 Ps Resolution 2.5 b/Stage Pipelined Time-to-Digital Converter in 65 Nm CMOS Using Time-Register”. In: *IEEE Journal of Solid-State Circuits* 49.4 (Apr. 2014), pp. 1007–1016.
- [30] Andrew R. Macpherson, James W. Haslett, and Leonid Belostotski. “A 5GS/s 4-Bit Time-Based Single-Channel CMOS ADC for Radio Astronomy”. In: *Proceedings of the IEEE 2013 Custom Integrated Circuits Conference*. San Jose, CA, USA: IEEE, Sept. 2013.
- [31] H. Pekau, A. Yousif, and J.W. Haslett. “A CMOS Integrated Linear Voltage-to-Pulse-Delay-Time Converter for Time Based Analog-to-Digital Converters”. In: *2006 IEEE International Symposium on Circuits and Systems*. Island of Kos, Greece: IEEE, 2006, pp. 2373–2376.
- [32] M. Wagih Ismail and Hassan Mostafa. “A New Design Methodology for Voltage-to-Time Converters (VTCs) Circuits Suitable for Time-based Analog-to-Digital Converters (T-ADC)”. In: *2014 27th IEEE International System-on-Chip Conference (SOCC)*. Las Vegas, NV, USA: IEEE, Sept. 2014, pp. 103–108.
- [33] Hassan Mostafa and Yehea I. Ismail. “Highly-Linear Voltage-to-Time Converter (VTC) Circuit for Time-Based Analog-to-Digital Converters (T-ADCs)”. In: *2013 IEEE 20th International Conference on Electronics, Circuits, and Systems (ICECS)*. Abu Dhabi, United Arab Emirates: IEEE, Dec. 2013, pp. 149–152.
- [34] Masaya Miyahara et al. “22.6 A 2.2GS/s 7b 27.4mW Time-Based Folding-Flash ADC with Resistively Averaged Voltage-to-Time Amplifiers”. In: *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*. San Francisco, CA, USA: IEEE, Feb. 2014, pp. 388–389.

- [35] Shuang Zhu et al. “A Skew-Free 10 GS/s 6 Bit CMOS ADC With Compact Time-Domain Signal Folding and Inherent DEM”. In: *IEEE Journal of Solid-State Circuits* 51.8 (Aug. 2016), pp. 1785–1796.
- [36] Y. Arai, T. Matsumura, and K.-i. Endo. “A CMOS Four-channel*1K Time Memory LSI with 1-Ns/b Resolution”. In: *IEEE Journal of Solid-State Circuits* 27.3 (Mar. 1992), pp. 359–364.
- [37] Elvi Räsänen Ruotsalainen, Timo Rahkonen, and Juha Kostamovaara. “A High Resolution Time-to-Digital Converter Based on Time-to-Voltage Interpolation”. In: ().
- [38] Stephan Henzler et al. “A Local Passive Time Interpolation Concept for Variation-Tolerant High-Resolution Time-to-Digital Conversion”. In: *IEEE Journal of Solid-State Circuits* 43.7 (July 2008), pp. 1666–1676.
- [39] P. Andreani et al. “Multihit Multichannel Time-to-Digital Converter with $\pm 1\%$ Differential Nonlinearity and near Optimal Time Resolution”. In: *IEEE Journal of Solid-State Circuits* 33.4 (Apr. 1998), pp. 650–656.
- [40] R.B. Staszewski et al. “1.3 V 20 Ps Time-to-Digital Converter for Frequency Synthesis in 90-Nm CMOS”. In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 53.3 (Mar. 2006), pp. 220–224.
- [41] Antti Mantyniemi, Timo Rahkonen, and Juha Kostamovaara. “A CMOS Time-to-Digital Converter (TDC) Based On a Cyclic Time Domain Successive Approximation Interpolation Method”. In: *IEEE Journal of Solid-State Circuits* 44.11 (Nov. 2009), pp. 3067–3078.
- [42] J.-P. Jansson, A. Mantyniemi, and J. Kostamovaara. “A CMOS Time-to-Digital Converter With Better Than 10 Ps Single-Shot Precision”. In: *IEEE Journal of Solid-State Circuits* 41.6 (June 2006), pp. 1286–1296.
- [43] Belal M. Helal et al. “A Low Jitter 1.6 GHz Multiplying DLL Utilizing a Scrambling Time-to-Digital Converter and Digital Correlation”. In: *2007 IEEE Symposium on VLSI Circuits*. Kyoto, Japan: IEEE, June 2007, pp. 166–167.
- [44] Robert Baron. “The Vernier Time-Measuring Technique”. In: *Proceedings of the IRE* 45.1 (1957), pp. 21–30.
- [45] Tae-Kwang Jang et al. “A Highly-Digital VCO-Based Analog-to-Digital Converter Using Phase Interpolator and Digital Calibration”. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 20.8 (Aug. 2012), pp. 1368–1372.
- [46] Yongkuo Ma et al. “A 4.39ps, 1.5GS/s Time-to-Digital Converter with $4\times$ Phase Interpolation Technique and a 2-D Quantization Array”. In: *2021 IEEE Asian Solid-State Circuits Conference (A-SSCC)*. Busan, Korea, Republic of: IEEE, Nov. 2021, pp. 1–3.

- [47] Akinori Matsumoto et al. “A Design Method and Developments of a Low-Power and High-Resolution Multiphase Generation System”. In: *IEEE Journal of Solid-State Circuits* 43.4 (Apr. 2008), pp. 831–843.
- [48] Jorg Daniels, Wim Dehaene, and Michiel Steyaert. “All-Digital Differential VCO-based A/D Conversion”. In: *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*. Paris, France: IEEE, May 2010, pp. 1085–1088.
- [49] P.M. Levine and G.W. Roberts. “A High-Resolution Flash Time-to-Digital Converter and Calibration Scheme”. In: *2004 International Conference on Test*. Charlotte, NC, USA: IEEE, 2004, pp. 1148–1157.
- [50] Hayun Chung, Minji Hyun, and Jungwon Kim. “A 360-Fs-Time-Resolution 7-Bit Stochastic Time-to-Digital Converter With Linearity Calibration Using Dual Time Offset Arbiters in 65-Nm CMOS”. In: *IEEE Journal of Solid-State Circuits* 56.3 (Mar. 2021), pp. 940–949.
- [51] M. Hovin et al. “Delta-Sigma Modulators Using Frequency-Modulated Intermediate Values”. In: *IEEE Journal of Solid-State Circuits* 32.1 (Jan. 1997), pp. 13–22.
- [52] A. Iwata et al. “The Architecture of Delta Sigma Analog-to-Digital Converters Using a Voltage-Controlled Oscillator as a Multibit Quantizer”. In: *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 46.7 (July 1999), pp. 941–945.
- [53] Maarten Baert and Wim Dehaene. “20.1 A 5GS/s 7.2 ENOB Time-Interleaved VCO-Based ADC Achieving 30.5fJ/Conv-Step”. In: *2019 IEEE International Solid-State Circuits Conference - (ISSCC)*. San Francisco, CA, USA: IEEE, Feb. 2019, pp. 328–330.
- [54] E. Gutierrez, P. Rombouts, and L. Hernandez. “Why and How VCO-based ADCs Can Improve Instrumentation Applications”. In: *2018 25th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*. Bordeaux: IEEE, Dec. 2018, pp. 101–104.
- [55] Jaewook Kim et al. “Analysis and Design of Voltage-Controlled Oscillator Based Analog-to-Digital Converter”. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 57.1 (Jan. 2010), pp. 18–30.
- [56] Mohsen Hassanpourghadi, Praveen Kumar Sharma, and Mike Shuo-Wei Chen. “A 6-b, 800-MS/s, 3.62-mW Nyquist Rate AC-Coupled VCO-Based ADC in 65-Nm CMOS”. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 64.6 (June 2017), pp. 1354–1367.
- [57] Jorg Daniels et al. “A 0.02mm² 65nm CMOS 30MHz BW All-Digital Differential VCO-based ADC with 64dB SNDR”. In: *2010 Symposium on VLSI Circuits*. Honolulu, HI, USA: IEEE, June 2010, pp. 155–156.

- [58] Matthew Z. Straayer and Michael H. Perrott. “A Multi-Path Gated Ring Oscillator TDC With First-Order Noise Shaping”. In: *IEEE Journal of Solid-State Circuits* 44.4 (Apr. 2009), pp. 1089–1098.
- [59] Long Pham and John McNeill. “Improved Lookup-Table-Based Algorithm for Background Linearization of VCO-based ADCs”. In: *2015 11th Conference on Ph.D. Research in Microelectronics and Electronics (PRIME)*. Glasgow, United Kingdom: IEEE, June 2015, pp. 196–199.
- [60] Gerry Taylor and Ian Galton. “A Mostly-Digital Variable-Rate Continuous-Time Delta-Sigma Modulator ADC”. In: *IEEE Journal of Solid-State Circuits* 45.12 (Dec. 2010), pp. 2634–2646.
- [61] Sachin Rao et al. “A 4.1mW, 12-Bit ENOB, 5MHz BW, VCO-based ADC with on-Chip Deterministic Digital Background Calibration in 90nm CMOS”. In: (), p. 2.
- [62] Sachin Rao et al. “A Deterministic Digital Background Calibration Technique for VCO-Based ADCs”. In: *IEEE Journal of Solid-State Circuits* 49.4 (Apr. 2014), pp. 950–960.
- [63] Gerry Taylor and Ian Galton. “A Reconfigurable Mostly-Digital Delta-Sigma ADC With a Worst-Case FOM of 160 dB”. In: *IEEE Journal of Solid-State Circuits* 48.4 (Apr. 2013), pp. 983–995.
- [64] Sachin Rao et al. “A 71dB SFDR Open Loop VCO-based ADC Using 2-Level PWM Modulation”. In: (), p. 2.
- [65] Peng Gao et al. “Design of an Intrinsically-Linear Double-VCO-based ADC with 2nd-Order Noise Shaping”. In: *2012 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. Dresden: IEEE, Mar. 2012, pp. 1215–1220.
- [66] Georges Gielen, Luis Hernandez-Corporales, and Pieter Rombouts. *Time-Encoding VCO-ADCs for Integrated Systems-on-Chip: Principles, Architectures and Circuits*. Cham: Springer International Publishing, 2022.
- [67] Eric Gutierrez. “Oversampled Analog-To-Digital Converter Architectures Based On Pulse Frequency Modulation”. PhD thesis.
- [68] Eric Gutierrez et al. “A Pulse Frequency Modulation VCO-ADC in 40 Nm”. In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 66.1 (Jan. 2019), pp. 51–55.
- [69] Luis Hernandez and Eric Gutierrez. “Analytical Evaluation of VCO-ADC Quantization Noise Spectrum Using Pulse Frequency Modulation”. In: *IEEE Signal Processing Letters* 22.2 (Feb. 2015), pp. 249–253.
- [70] Xinpeng Xing and Georges G. E. Gielen. “A 42 fJ/Step-FoM Two-Step VCO-Based Delta-Sigma ADC in 40 Nm CMOS”. In: *IEEE Journal of Solid-State Circuits* 50.3 (Mar. 2015), pp. 714–723.

- [71] Min Park and Michael H. Perrott. “A VCO-based Analog-to-Digital Converter with Second-Order Sigma-Delta Noise Shaping”. In: *2009 IEEE International Symposium on Circuits and Systems*. Taipei, Taiwan: IEEE, May 2009, pp. 3130–3133.
- [72] M. Park and M. Perrott. “A 0.13 μm CMOS 78dB SNDR 87mW 20MHz BW CT $\Delta\Sigma$ ADC with VCO-based Integrator and Quantizer”. In: *2009 IEEE International Solid-State Circuits Conference - Digest of Technical Papers*. San Francisco, CA: IEEE, Feb. 2009, 170–171, 171a.
- [73] Karthikeyan Reddy et al. “A 16-mW 78-dB SNDR 10-MHz BW CT $\Delta\Sigma$ ADC Using Residue-Cancelling VCO-Based Quantizer”. In: *IEEE Journal of Solid-State Circuits* 47.12 (Dec. 2012), pp. 2916–2927.
- [74] Matthew Z. Straayer and Michael H. Perrott. “A 12-Bit, 10-MHz Bandwidth, Continuous-Time $\Delta\Sigma$ ADC With a 5-Bit, 950-MS/s VCO-Based Quantizer”. In: *IEEE Journal of Solid-State Circuits* 43.4 (Apr. 2008), pp. 805–814.
- [75] Fernando Cardes et al. “0.04-Mm² 103-dB-A Dynamic Range Second-Order VCO-Based Audio $\Delta\Sigma$ ADC in 0.13-Mm CMOS”. In: *IEEE Journal of Solid-State Circuits* 53.6 (June 2018), pp. 1731–1742.
- [76] Amir Babaie Fishani and Pieter Rombouts. “A Mostly Digital VCO-Based CT-SDM With Third-Order Noise Shaping”. In: *IEEE Journal of Solid-State Circuits* 52.8 (Aug. 2017), pp. 2141–2153.
- [77] Akshay Jayaraj et al. “Highly Digital Second-Order $\Delta\Sigma$ VCO ADC”. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 66.7 (July 2019), pp. 2415–2425.
- [78] Yi Zhong et al. “A Second-Order Purely VCO-Based CT $\Delta\Sigma$ ADC Using a Modified DPLL Structure in 40-Nm CMOS”. In: *IEEE Journal of Solid-State Circuits* 55.2 (Feb. 2020), pp. 356–368.
- [79] Kareem Ragab and Nan Sun. “A 12-b ENOB 2.5-MHz BW VCO-Based 0-1 MASH ADC With Direct Digital Background Calibration”. In: *IEEE Journal of Solid-State Circuits* 52.2 (Feb. 2017), pp. 433–447.
- [80] Abhishek Ghosh and Sudhakar Pamarti. “Linearization Through Dithering: A 50 MHz Bandwidth, 10-b ENOB, 8.2 mW VCO-Based ADC”. In: *IEEE Journal of Solid-State Circuits* 50.9 (Sept. 2015), pp. 2012–2024.
- [81] Hamidreza Maghami et al. “A Highly Linear OTA-Less 1-1 MASH VCO-Based $\Delta\Sigma$ ADC With an Efficient Phase Quantization Noise Extraction Technique”. In: *IEEE Journal of Solid-State Circuits* 55.3 (Mar. 2020), pp. 706–718.
- [82] Hyunchul Yoon et al. “A 65-dB-SNDR Pipelined SAR ADC Using PVT-Robust Capacitively Degenerated Dynamic Amplifier”. In: *IEEE Journal of Solid-State Circuits* (2023).

- [83] Xiaofeng Guo et al. “A 13b 600-675MS/s Tri-State Pipelined-SAR ADC With Inverter-Based Open-Loop Residue Amplifier”. In: *IEEE Journal of Solid-State Circuits* (2022).
- [84] Minglei Zhang, Qiyuan Liu, and Xiaohua Fan. “Gain-boosted Dynamic Amplifier for pipelined-SAR ADCs”. In: *Electronics Letters* 53.11 (May 2017), pp. 708–709.
- [85] Jorge Lagos et al. “A 10.1-ENOB, 6.2-fJ/Conv.-Step, 500-MS/s, Ringamp-Based Pipelined-SAR ADC With Background Calibration and Dynamic Reference Regulation in 16-Nm CMOS”. In: *IEEE Journal of Solid-State Circuits* 57.4 (Apr. 2022), pp. 1112–1124.
- [86] Y. Lyu, A. Ramkaj, and F. Tavernier. “High-gain and Power-efficient Dynamic Amplifier for Pipelined SAR ADCs”. In: *Electronics Letters* 53.23 (Nov. 2017), pp. 1510–1512.
- [87] Hai Huang et al. “A Non-Interleaved 12-b 330-MS/s Pipelined-SAR ADC With PVT-Stabilized Dynamic Amplifier Achieving Sub-1-dB SNDR Variation”. In: *IEEE Journal of Solid-State Circuits* 52.12 (Dec. 2017), pp. 3235–3247.
- [88] Yongzhen Chen et al. “A 625MS/s, 12-Bit, SAR Assisted Pipeline ADC with Effective Gain Analysis for Inter-stage Ringamps”. In: *ESSCIRC 2019 - IEEE 45th European Solid State Circuits Conference (ESSCIRC)*. Cracow, Poland: IEEE, Sept. 2019, pp. 197–200.
- [89] Wenning Jiang et al. “A Temperature-Stabilized Single-Channel 1-GS/s 60-dB SNDR SAR-Assisted Pipelined ADC With Dynamic Gm-R-Based Amplifier”. In: *IEEE Journal of Solid-State Circuits* 55.2 (Feb. 2020), pp. 322–332.
- [90] Yong Lim and Michael P. Flynn. “A Calibration-Free 2.3 mW 73.2 dB SNDR 15b 100 MS/s Four-Stage Fully Differential Ring Amplifier Based SAR-assisted Pipeline ADC”. In: *2017 Symposium on VLSI Circuits*. Kyoto, Japan: IEEE, June 2017, pp. C98–C99.
- [91] Yong Lim and Michael P. Flynn. “A 1 mW 71.5 dB SNDR 50 MS/s 13 Bit Fully Differential Ring Amplifier Based SAR-Assisted Pipeline ADC”. In: *IEEE Journal of Solid-State Circuits* 50.12 (Dec. 2015), pp. 2901–2911.
- [92] Athanasios Ramkaj et al. “3.3 A 5GS/s 158.6mW 12b Passive-Sampling 8 \times -Interleaved Hybrid ADC with 9.4 ENOB and 160.5dB FoMS in 28nm CMOS”. In: *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*. San Francisco, CA, USA: IEEE, Feb. 2019, pp. 62–64.
- [93] W. Yang et al. “A 3-V 340-mW 14-b 75-Msample/s CMOS ADC with 85-dB SFDR at Nyquist Input”. In: *IEEE Journal of Solid-State Circuits* 36.12 (Dec. 2001), pp. 1931–1936.
- [94] “A 2.02–5.16 fJ/Conversion Step 10 Bit Hybrid Coarse-Fine SAR ADC With Time-Domain Quantizer in 90 Nm CMOS”. In: *IEEE Journal of Solid-State Circuits* 51.2 (Feb. 2016), pp. 357–364.

- [95] Yi-Long Yu et al. “A Two-Step ADC With Statistical Calibration”. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 67.8 (Aug. 2020), pp. 2588–2601.
- [96] Minglei Zhang et al. “A 0.6-V 13-Bit 20-MS/s Two-Step TDC-Assisted SAR ADC With PVT Tracking and Speed-Enhanced Techniques”. In: *IEEE Journal of Solid-State Circuits* 54.12 (Dec. 2019), pp. 3396–3409.
- [97] Haoyi Zhao and Fa Foster Dai. “A 12-Bit 260-MS/s Pipelined-SAR ADC With Ring-TDC-Based Fine Quantizer for Automatic Cross-Domain Scale Alignment”. In: *IEEE Journal of Solid-State Circuits* (2023), pp. 1–14.
- [98] Amy Whitcombe et al. “A VTC/TDC-Assisted $4\times$ Interleaved 3.8 GS/s 7b 6.0 mW SAR ADC With 13 GHz ERBW”. In: *IEEE Journal of Solid-State Circuits* 58.4 (Apr. 2023), pp. 972–982.
- [99] Arindam Sanyal and Nan Sun. “A 18.5-fJ/Step VCO-based 0–1 MASH $\Delta\Sigma$ ADC with Digital Background Calibration”. In: *2016 IEEE Symposium on VLSI Circuits (VLSI-Circuits)*. Honolulu, HI, USA: IEEE, June 2016, pp. 1–2.
- [100] Arindam Sanyal et al. “A Hybrid SAR-VCO Sigma-Delta ADC with First-Order Noise Shaping”. In: *Proceedings of the IEEE 2014 Custom Integrated Circuits Conference*. San Jose, CA, USA: IEEE, Sept. 2014.
- [101] A. K. Gupta, K. Nagaraj, and T. R. Viswanathan. “A Two-Stage ADC Architecture With VCO-Based Second Stage”. In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 58.11 (Nov. 2011), pp. 734–738.
- [102] Young-Gyu Yoon et al. “A Time-Based Bandpass ADC Using Time-Interleaved Voltage-Controlled Oscillators”. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 55.11 (Dec. 2008), pp. 3571–3581.
- [103] Boris Murmann. *ADC Performance Survey 1997-2023 [Online]*. URL: Available: %20https://github.com/bmurmann/ADC-survey.
- [104] J. Crossley et al. “BAG: A Designer-Oriented Integrated Framework for the Development of AMS Circuit Generators”. In: *2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. San Jose, CA: IEEE, Nov. 2013, pp. 74–81.
- [105] Juergen Scheible and Jens Lienig. “Automation of Analog IC Layout: Challenges and Solutions”. In: *Proceedings of the 2015 Symposium on International Symposium on Physical Design*. Monterey California USA: ACM, Mar. 2015, pp. 33–40.
- [106] Michael Eick and Helmut E. Graeb. “MARS: Matching-Driven Analog Sizing”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 31.8 (Aug. 2012), pp. 1145–1158.
- [107] Lihong Zhang et al. “Parasitic-Aware Optimization and Retargeting of Analog Layouts: A Symbolic-Template Approach”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 27.5 (May 2008), pp. 791–802.

- [108] Po-Hsuan Wei and Boris Murmann. “Analog and Mixed-Signal Layout Automation Using Digital Place-and-Route Tools”. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 29.11 (Nov. 2021), pp. 1838–1849.
- [109] Francois Stas, Gueric de Streel, and David Bol. “Sizing and Layout Integrated Optimizer for 28nm Analog Circuits Using Digital PnR Tools”. In: *2016 14th IEEE International New Circuits and Systems Conference (NEWCAS)*. Vancouver, BC, Canada: IEEE, June 2016.
- [110] Allen Waters and Un-Ku Moon. “A Fully Automated Verilog-to-Layout Synthesized ADC Demonstrating 56dB-SNDR with 2MHz-BW”. In: *2015 IEEE Asian Solid-State Circuits Conference (A-SSCC)*. Xia’men, China: IEEE, Nov. 2015.
- [111] Ming Ding et al. “A Hybrid Design Automation Tool for SAR ADCs in IoT”. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 26.12 (Dec. 2018), pp. 2853–2862.
- [112] Suyoung Bang et al. “25.1 A Fully Synthesizable Distributed and Scalable All-Digital LDO in 10nm CMOS”. In: *2020 IEEE International Solid-State Circuits Conference - (ISSCC)*. San Francisco, CA, USA: IEEE, Feb. 2020, pp. 380–382.
- [113] Tonmoy Dhar et al. *ALIGN: A System for Automating Analog Layout*. Aug. 2020. arXiv: 2008.10682 [cs].
- [114] Keren Zhu et al. “Tutorial and Perspectives on MAGICAL: A Silicon-Proven Open-Source Analog IC Layout System”. In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 70.2 (Feb. 2023), pp. 715–720.
- [115] Xu Jingnan, J. Vital, and N. Horta. “A SKILLTM-Based Library for Retargetable Embedded Analog Cores”. In: *Proceedings Design, Automation and Test in Europe. Conference and Exhibition 2001*. Munich, Germany: IEEE Comput. Soc, 2001, pp. 768–769.
- [116] Nuttorn Jangkrajarn et al. “IPRAIL—Intellectual Property Reuse-Based Analog IC Layout Automation”. In: *Integration* 36.4 (Nov. 2003), pp. 237–262.
- [117] Ricardo Martins, Nuno Lourenco, and Nuno Horta. “LAYGEN II—Automatic Layout Generation of Analog Integrated Circuits”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 32.11 (Nov. 2013), pp. 1641–1654.
- [118] Keertana Settaluri et al. “Automated Design of Analog Circuits Using Reinforcement Learning”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 41.9 (Sept. 2022), pp. 2794–2807.
- [119] Kouros Hakhmaneshi et al. “BagNet: Berkeley Analog Generator with Layout Optimizer Boosted with Deep Neural Networks”. In: *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. Westminster, CO, USA: IEEE, Nov. 2019.

- [120] A.M. Abo and P.R. Gray. “A 1.5-V, 10-Bit, 14.3-MS/s CMOS Pipeline Analog-to-Digital Converter”. In: *IEEE Journal of Solid-State Circuits* 34.5 (May 1999), pp. 599–606.
- [121] D. Aksin, M. Al-Shyoukh, and F. Maloberti. “Switch Bootstrapping for Precise Sampling Beyond Supply Voltage”. In: *IEEE Journal of Solid-State Circuits* 41.8 (Aug. 2006), pp. 1938–1943.
- [122] Eric Swindlehurst et al. “An 8-Bit 10-GHz 21-mW Time-Interleaved SAR ADC With Grouped DAC Capacitors and Dual-Path Bootstrapped Switch”. In: *IEEE Solid-State Circuits Letters* 2.9 (Sept. 2019), pp. 83–86.
- [123] Athanasios T. Ramkaj et al. “A 1.25-GS/s 7-b SAR ADC With 36.4-dB SNDR at 5 GHz Using Switch-Bootstrapping, USPC DAC and Triple-Tail Comparator in 28-Nm CMOS”. In: *IEEE Journal of Solid-State Circuits* 53.7 (July 2018), pp. 1889–1901.
- [124] Kostas Doris et al. “A 480 mW 2.6 GS/s 10b Time-Interleaved ADC With 48.5 dB SNDR up to Nyquist in 65 Nm CMOS”. In: *IEEE Journal of Solid-State Circuits* 46.12 (Dec. 2011), pp. 2821–2833.
- [125] Tao Jiang et al. “Single-Channel, 1.25-GS/s, 6-Bit, Loop-Unrolled Asynchronous SAR-ADC in 40nm-CMOS”. In: *IEEE Custom Integrated Circuits Conference 2010*. San Jose, CA, USA: IEEE, Sept. 2010.
- [126] Athanasios T. Ramkaj, Michiel S. J. Steyaert, and Filip Tavernier. “A 13.5-Gb/s 5-mV-Sensitivity 26.8-Ps-CLK–OUT Delay Triple-Latch Feedforward Dynamic Comparator in 28-Nm CMOS”. In: *ESSCIRC 2019 - IEEE 45th European Solid State Circuits Conference (ESSCIRC)*. Cracow, Poland: IEEE, Sept. 2019, pp. 167–170.
- [127] Daniel Schinkel et al. “A Double-Tail Latch-Type Voltage Sense Amplifier with 18ps Setup + Hold Time”. In: *2007 IEEE International Solid-State Circuits Conference. Digest of Technical Papers*. San Francisco, CA: IEEE, Feb. 2007, pp. 314–605.
- [128] Michiel van Elzakker et al. “A 10-Bit Charge-Redistribution ADC Consuming 1.9 μ W at 1 MS/s”. In: *IEEE Journal of Solid-State Circuits* 45.5 (May 2010), pp. 1007–1015.
- [129] T. Kobayashi et al. “A Current-Controlled Latch Sense Amplifier and a Static Power-Saving Input Buffer for Low-Power Architecture”. In: *IEEE Journal of Solid-State Circuits* 28.4 (Apr. 1993), pp. 523–527.
- [130] Masaya Miyahara et al. “A Low-Noise Self-Calibrating Dynamic Comparator for High-Speed ADCs”. In: *2008 IEEE Asian Solid-State Circuits Conference*. Fukuoka, Japan: IEEE, Nov. 2008, pp. 269–272.
- [131] Harijot Singh Bindra et al. “A 1.2-V Dynamic Bias Latch-Type Comparator in 65-Nm CMOS With 0.4-mV Input Noise”. In: *IEEE Journal of Solid-State Circuits* 53.7 (July 2018), pp. 1902–1912.

- [132] Pi-Feng Chiu, Brian Zimmer, and Borivoje Nikolic. “A Double-Tail Sense Amplifier for Low-Voltage SRAM in 28nm Technology”. In: *2016 IEEE Asian Solid-State Circuits Conference (A-SSCC)*. Toyama, Japan: IEEE, Nov. 2016, pp. 181–184.
- [133] Aikaterini Papadopoulou, Vladimir Milovanovic, and Borivoje Nikolic. “A Low-Voltage Low-Offset Dual Strong-Arm Latch Comparator”. In: *2017 IEEE Asian Solid-State Circuits Conference (A-SSCC)*. Seoul: IEEE, Nov. 2017, pp. 281–284.
- [134] Ata Khorami and Mohammad Sharifkhani. “A Low-Power High-Speed Comparator for Precise Applications”. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 26.10 (Oct. 2018), pp. 2038–2049.
- [135] Hao Xu and Asad A. Abidi. “Analysis and Design of Regenerative Comparators for Low Offset and Noise”. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 66.8 (Aug. 2019), pp. 2817–2830.
- [136] Xin Xin et al. “Ultra-low Power Comparator with Dynamic Offset Cancellation for SAR ADC”. In: *Electronics Letters* 53.24 (Nov. 2017), pp. 1572–1574.
- [137] S.H. Lewis et al. “A 10-b 20-Msample/s Analog-to-Digital Converter”. In: *IEEE Journal of Solid-State Circuits* 27.3 (Mar. 1992), pp. 351–358.
- [138] R. Vitek et al. “A 0.015mm² 63fJ/Conversion-Step 10-Bit 220MS/s SAR ADC with 1.5b/Step Redundancy and Digital Metastability Correction”. In: *Proceedings of the IEEE 2012 Custom Integrated Circuits Conference*. San Jose, CA, USA: IEEE, Sept. 2012.
- [139] Draxelmayr. “A Self Calibration Technique for Redundant A/d Converters Providing 16b Accuracy”. In: *1988 IEEE International Solid-State Circuits Conference, 1988 ISSCC. Digest of Technical Papers*. San Francisco, CA: IEEE, 1988, p. 204.
- [140] F. Kuttner. “A 1.2V 10b 20MSample/s Non-Binary Successive Approximation ADC in 0.13 μ m CMOS”. In: *2002 IEEE International Solid-State Circuits Conference. Digest of Technical Papers (Cat. No.02CH37315)*. Vol. 1. San Francisco, CA, USA: IEEE, 2002, pp. 176–177.
- [141] Z. Boyacigiller, B. Weir, and P. Bradshaw. “An Error-Correcting 14b/20Msample/s CMOS A/D Converter”. In: *1981 IEEE International Solid-State Circuits Conference. Digest of Technical Papers*. New York, NY, USA: IEEE, 1981, pp. 62–63.
- [142] I. Daubechies et al. “Beta Expansions: A New Approach to Digitally Corrected A/D Conversion”. In: *2002 IEEE International Symposium on Circuits and Systems. Proceedings (Cat. No.02CH37353)*. Phoenix-Scottsdale, AZ, USA: IEEE, 2002, pp. II–II.
- [143] Vito Giannini et al. “An 820 μ W 9b 40MS/s Noise-Tolerant Dynamic-SAR ADC in 90nm Digital CMOS”. In: *2008 IEEE International Solid-State Circuits Conference - Digest of Technical Papers*. San Francisco, CA, USA: IEEE, Feb. 2008, pp. 238–610.

- [144] Wenbo Liu and Yun Chiu. “An Equalization-Based Adaptive Digital Background Calibration Technique for Successive Approximation Analog-to-Digital Converters”. In: *2007 7th International Conference on ASIC*. Guilin, China: IEEE, Oct. 2007, pp. 289–292.
- [145] Brian P. Ginsburg and Anantha P. Chandrakasan. “500-MS/s 5-Bit ADC in 65-Nm CMOS With Split Capacitor Array DAC”. In: *IEEE Journal of Solid-State Circuits* 42.4 (Apr. 2007), pp. 739–747.
- [146] Chun-Cheng Liu et al. “A 10-Bit 50-MS/s SAR ADC With a Monotonic Capacitor Switching Procedure”. In: *IEEE Journal of Solid-State Circuits* 45.4 (Apr. 2010), pp. 731–740.
- [147] B.P. Ginsburg and A.P. Chandrakasan. “An Energy-Efficient Charge Recycling Approach for a SAR Converter With Capacitive DAC”. In: *2005 IEEE International Symposium on Circuits and Systems*. Kobe, Japan: IEEE, 2005, pp. 184–187.
- [148] Yan Zhu et al. “A 10-Bit 100-MS/s Reference-Free SAR ADC in 90 Nm CMOS”. In: *IEEE Journal of Solid-State Circuits* 45.6 (June 2010), pp. 1111–1121.
- [149] C. Yuan and Y. Lam. “Low-Energy and Area-Efficient Tri-Level Switching Scheme for SAR ADC”. In: *Electronics Letters* 48.9 (2012), p. 482.
- [150] N. Sun and A. Sanyal. “SAR ADC Architecture with 98% Reduction in Switching Energy over Conventional Scheme”. In: *Electronics Letters* 49.4 (Feb. 2013), pp. 248–250.
- [151] Xiaoli Song, Yu Xiao, and Zhangming Zhu. “VCM-based Monotonic Capacitor Switching Scheme for SAR ADC”. In: *Electronics Letters* 49.5 (Feb. 2013), pp. 327–329.
- [152] Shubin Liu, Yi Shen, and Zhangming Zhu. “A 12-Bit 10 MS/s SAR ADC With High Linearity and Energy-Efficient Switching”. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 63.10 (Oct. 2016), pp. 1616–1627.
- [153] Hung-Yen Tai et al. “11.2 A 0.85fJ/Conversion-Step 10b 200kS/s Subranging SAR ADC in 40nm CMOS”. In: *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*. San Francisco, CA, USA: IEEE, Feb. 2014, pp. 196–197.
- [154] Pieter Harpe et al. “A 12fJ/Conversion-Step 8bit 10MS/s Asynchronous SAR ADC for Low Energy Radios”. In: *2010 Proceedings of ESSCIRC*. Sevilla, Spain: IEEE, Sept. 2010, pp. 214–217.
- [155] Akira Shikata et al. “A 0.5 V 1.1 MS/Sec 6.3 fJ/Conversion-Step SAR-ADC With Tri-Level Comparator in 40 Nm CMOS”. In: *IEEE Journal of Solid-State Circuits* 47.4 (Apr. 2012), pp. 1022–1030.
- [156] A.A. Abidi. “Phase Noise and Jitter in CMOS Ring Oscillators”. In: *IEEE Journal of Solid-State Circuits* 41.8 (Aug. 2006), pp. 1803–1816.

- [157] Chengxin Liu and J.A. McNeill. “Jitter in Oscillators with $1/f$ Noise Sources”. In: *2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No.04CH37512)*. Vancouver, BC, Canada: IEEE, 2004, pp. I-773-6.
- [158] Clemenz L Portmann and Teresa H Y Meng. “Power-Efficient Etastability Error Reduction in CMOS Flash A/D Converters”. In: 31.8 (1996).
- [159] David Rennie et al. “Performance, Metastability and Soft-Error Robustness Tradeoffs for Flip-Flops in 40nm CMOS”. In: *2011 IEEE Custom Integrated Circuits Conference (CICC)*. San Jose, CA, USA: IEEE, Sept. 2011, pp. 1–4.
- [160] Abhishek Mukherjee et al. “A 1-GS/s 20 MHz-BW Capacitive-Input Continuous-Time $\Delta\Sigma$ ADC Using a Novel Parasitic Pole-Mitigated Fully Differential VCO”. In: *IEEE Solid-State Circuits Letters* 2.1 (Jan. 2019).
- [161] Chih-Chan Tu, Yu-Kai Wang, and Tsung-Hsien Lin. “A $0.06\text{mm}^2 \pm 50\text{mV}$ Range -82dB THD Chopper VCO-based Sensor Readout Circuit in 40nm CMOS”. In: *2017 Symposium on VLSI Circuits*. Kyoto, Japan: IEEE, June 2017, pp. C84–C85.
- [162] Matthew Z. Straayer and Michael H. Perrott. “An Efficient High-Resolution 11-Bit Noise-Shaping Multipath Gated Ring Oscillator TDC”. In: *2008 IEEE Symposium on VLSI Circuits*. Honolulu, HI, USA: IEEE, June 2008, pp. 82–83.
- [163] Jaewook Kim and Seonghwan Cho. “A Time-Based Analog-to-Digital Converter Using a Multi-Phase Voltage-Controlled Oscillator”. In: *2006 IEEE International Symposium on Circuits and Systems*. Island of Kos, Greece: IEEE, 2006, pp. 3934–3937.
- [164] Maarten Baert and Wim Dehaene. “A 5-GS/s 7.2-ENOB Time-Interleaved VCO-Based ADC Achieving 30.5 fJ/Cs”. In: *IEEE Journal of Solid-State Circuits* (2020), pp. 1577–1587.
- [165] Yikun Chang et al. “An 80-Gb/s 44-mW Wireline PAM4 Transmitter”. In: *IEEE Journal of Solid-State Circuits* 53.8 (Aug. 2018), pp. 2214–2226.
- [166] B. Nikolic et al. “Improved Sense-Amplifier-Based Flip-Flop: Design and Measurements”. In: *IEEE Journal of Solid-State Circuits* 35.6 (June 2000), pp. 876–884.
- [167] Sedigheh Hashemi and Behzad Razavi. “A 7.1 mW 1 GS/s ADC With 48 dB SNDR at Nyquist Rate”. In: *IEEE Journal of Solid-State Circuits* 49.8 (Aug. 2014), pp. 1739–1750.
- [168] Vladimir Milovanovic and Horst Zimmermann. “On Fully Differential and Complementary Single-Stage Self-Biased CMOS Differential Amplifiers”. In: *Eurocon 2013*. Zagreb, Croatia: IEEE, July 2013, pp. 1955–1963.
- [169] Bob Verbruggen, Masao Iriguchi, and Jan Cranineckx. “A 1.7 mW 11b 250 MS/s 2-Times Interleaved Fully Dynamic Pipelined SAR ADC in 40 Nm Digital CMOS”. In: *IEEE Journal of Solid-State Circuits* 47.12 (Dec. 2012), pp. 2880–2887.

- [170] Yifan Lyu and Filip Tavernier. “A 4-GS/s 39.9-dB SNDR 11.7-mW Hybrid Voltage-Time Two-Step ADC With Feedforward Ring Oscillator-Based TDCs”. In: *IEEE Journal of Solid-State Circuits* 55.7 (July 2020), pp. 1807–1818.
- [171] James Lin, Masaya Miyahara, and Akira Matsuzawa. “A 15.5 dB, Wide Signal Swing, Dynamic Amplifier Using a Common-Mode Voltage Detection Technique”. In: *2011 IEEE International Symposium of Circuits and Systems (ISCAS)*. Rio de Janeiro, Brazil: IEEE, May 2011, pp. 21–24.
- [172] Shiuh-Hua Wood Chiang, Hyuk Sun, and Behzad Razavi. “A 10-Bit 800-MHz 19-mW CMOS ADC”. In: *IEEE Journal of Solid-State Circuits* 49.4 (Apr. 2014), pp. 935–949.
- [173] Badr Malki et al. “A Complementary Dynamic Residue Amplifier for a 67 dB SNDR 1.36 mW 170 MS/s Pipelined SAR ADC”. In: *ESSCIRC 2014 - 40th European Solid State Circuits Conference (ESSCIRC)*. Venice Lido, Italy: IEEE, Sept. 2014, pp. 215–218.
- [174] Chun-Cheng Liu and Mu-Chen Huang. “28.1 A 0.46mW 5MHz-BW 79.7dB-SNDR Noise-Shaping SAR ADC with Dynamic-Amplifier-Based FIR-IIR Filter”. In: *2017 IEEE International Solid-State Circuits Conference (ISSCC)*. San Francisco, CA, USA: IEEE, Feb. 2017, pp. 466–467.
- [175] Frank van der Goes et al. “A 1.5 mW 68 dB SNDR 80 Ms/s $2\times$ Interleaved Pipelined SAR ADC in 28 Nm CMOS”. In: *IEEE Journal of Solid-State Circuits* 49.12 (Dec. 2014), pp. 2835–2845.
- [176] Yuanming Zhu et al. “A 1.5GS/s 8b Pipelined-SAR ADC with Output Level Shifting Settling Technique in 14nm CMOS”. In: *2020 IEEE Custom Integrated Circuits Conference (CICC)*. Boston, MA, USA: IEEE, Mar. 2020.
- [177] Zihao Zheng et al. “16.3 A Single-Channel 5.5mW 3.3GS/s 6b Fully Dynamic Pipelined ADC with Post-Amplification Residue Generation”. In: *2020 IEEE International Solid- State Circuits Conference - (ISSCC)*. San Francisco, CA, USA: IEEE, Feb. 2020, pp. 254–256.
- [178] Minglei Zhang et al. “A 0.8–1.2 V 10–50 MS/s 13-Bit Subranging Pipelined-SAR ADC Using a Temperature-Insensitive Time-Based Amplifier”. In: *IEEE Journal of Solid-State Circuits* 52.11 (Nov. 2017), pp. 2991–3005.
- [179] Xiyuan Tang et al. “27.4 A 0.4-to-40MS/s 75.7dB-SNDR Fully Dynamic Event-Driven Pipelined ADC with 3-Stage Cascoded Floating Inverter Amplifier”. In: *2021 IEEE International Solid- State Circuits Conference (ISSCC)*. San Francisco, CA, USA: IEEE, Feb. 2021, pp. 376–378.
- [180] Xiyuan Tang et al. “An Energy-Efficient Comparator With Dynamic Floating Inverter Amplifier”. In: *IEEE Journal of Solid-State Circuits* 55.4 (Apr. 2020), pp. 1011–1022.
- [181] Md Shakil Akter, Kofi A. A. Makinwa, and Klaas Bult. “A Capacitively Degenerated 100-dB Linear 20–150 MS/s Dynamic Amplifier”. In: *IEEE Journal of Solid-State Circuits* 53.4 (Apr. 2018), pp. 1115–1126.

- [182] Longheng Luo et al. “A Capacitively-Degenerated High-Linearity Dynamic Amplifier Using a Real-Time Gain Detection Technique”. In: *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*. Sapporo, Japan: IEEE, May 2019.
- [183] Yigi Kwon et al. “A 348- μ W 68.8-dB SNDR 20-MS/s Pipelined SAR ADC With a Closed-Loop Two-Stage Dynamic Amplifier”. In: *IEEE Solid-State Circuits Letters 4* (2021), pp. 166–169.
- [184] Jingchao Lan et al. “A Novel Ring Amplifier with Low Common-Mode Voltage Variation and Noise Reduction Using Floating Power Technique”. In: *2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*. Lansing, MI, USA: IEEE, Aug. 2021, pp. 949–953.
- [185] Jingchao Lan et al. “A Single-Channel 1.25-GS/s 11-Bit Pipelined ADC with Robust Floating-Powered Ring Amplifier and First-Order Gain Error Calibration”. In: *2022 IEEE 65th International Midwest Symposium on Circuits and Systems (MWSCAS)*. Fukuoka, Japan: IEEE, Aug. 2022.
- [186] Benjamin Hershberg et al. “Ring Amplifiers for Switched Capacitor Circuits”. In: *IEEE Journal of Solid-State Circuits* 47.12 (Dec. 2012), pp. 2928–2942.
- [187] Benjamin Hershberg et al. “A 61.5dB SNDR Pipelined ADC Using Simple Highly-Scalable Ring Amplifiers”. In: *2012 Symposium on VLSI Circuits (VLSIC)*. Honolulu, HI, USA: IEEE, June 2012, pp. 32–33.
- [188] Jorge Lagos et al. “A Single-Channel, 600-MS/s, 12-b, Ringamp-Based Pipelined ADC in 28-Nm CMOS”. In: *IEEE Journal of Solid-State Circuits* 54.2 (Feb. 2019), pp. 403–416.
- [189] Jorge Lagos et al. “A 1-GS/s, 12-b, Single-Channel Pipelined ADC With Dead-Zone-Degenerated Ring Amplifiers”. In: *IEEE Journal of Solid-State Circuits* 54.3 (Mar. 2019), pp. 646–658.
- [190] Yongzhen Chen et al. “A 800 MS/s, 12-Bit, Ringamp-Based SAR Assisted Pipeline ADC with Gain Error Cancellation”. In: *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*. Florence: IEEE, 2018.
- [191] Benjamin Hershberg et al. “A 4-GS/s 10-ENOB 75-mW Ringamp ADC in 16-Nm CMOS With Background Monitoring of Distortion”. In: *IEEE Journal of Solid-State Circuits* 56.8 (Aug. 2021), pp. 2360–2374.
- [192] Benjamin Hershberg et al. “A 1-MS/s to 1-GS/s Ringamp-Based Pipelined ADC With Fully Dynamic Reference Regulation and Stochastic Scope-on-Chip Background Monitoring in 16 Nm”. In: *IEEE Journal of Solid-State Circuits* (2021), pp. 1227–1240.
- [193] Y.-C. Jenq. “Digital Spectra of Nonuniformly Sampled Signals: A Robust Sampling Time Offset Estimation Algorithm for Ultra High-Speed Waveform Digitizers Using Interleaving”. In: *IEEE Transactions on Instrumentation and Measurement* 39.1 (Feb./1990), pp. 71–75.

- [194] Keertana Settaluri. “Practical Solutions to Accelerating ASIC Design Development Using Machine Learning”. PhD thesis.
- [195] Jae-Won Nam et al. “A 12-Bit 1.6, 3.2, and 6.4 GS/s 4-b/Cycle Time-Interleaved SAR ADC With Dual Reference Shifting and Interpolation”. In: *IEEE Journal of Solid-State Circuits* 53.6 (June 2018), pp. 1765–1779.
- [196] Benjamin Hershberg et al. “3.1 A 3.2GS/s 10 ENOB 61mW Ringamp ADC in 16nm with Background Monitoring of Distortion”. In: *2019 IEEE International Solid-State Circuits Conference - (ISSCC)*. San Francisco, CA, USA: IEEE, Feb. 2019, pp. 58–60.
- [197] Kyoung-Jun Moon et al. “A 12-Bit 10GS/s 16-Channel Time-Interleaved ADC with a Digital Processing Timing-Skew Background Calibration in 5nm FinFET”. In: *2022 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*. Honolulu, HI, USA: IEEE, June 2022, pp. 172–173.
- [198] Dong-Jin Chang and Seung-Tak Ryu. “A Relative-Prime Rotation Based Fully On-Chip Background Skew Calibration for Time-Interleaved ADCs”. In: *2022 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*. Honolulu, HI, USA: IEEE, June 2022, pp. 174–175.
- [199] Ahmed M.A. Ali et al. “A 14-Bit 2.5GS/s and 5GS/s RF Sampling ADC with Background Calibration and Dither”. In: *2016 IEEE Symposium on VLSI Circuits (VLSI Circuits)*. Honolulu, HI, USA: IEEE, June 2016, pp. 1–2.
- [200] Bruno Vaz et al. “16.1 A 13b 4GS/s Digitally Assisted Dynamic 3-Stage Asynchronous Pipelined-SAR ADC”. In: *2017 IEEE International Solid-State Circuits Conference (ISSCC)*. San Francisco, CA, USA: IEEE, Feb. 2017, pp. 276–277.
- [201] Jiangfeng Wu et al. “27.6 A 4GS/s 13b Pipelined ADC with Capacitor and Amplifier Sharing in 16nm CMOS”. In: *2016 IEEE International Solid-State Circuits Conference (ISSCC)*. San Francisco, CA, USA: IEEE, Jan. 2016, pp. 466–467.
- [202] L Ricci et al. “A 2GS/s 11b 8x Interleaved ADC with 9.2 ENOB and 69.9dB SFDR in 28nm CMOS”. In: *2023 Symposium on VLSI Circuits (VLSIC)*. IEEE, 2023.
- [203] Jaeduk Han et al. “A Generated 7GS/s 8b Time-Interleaved SAR ADC with 38.2dB SNDR at Nyquist in 16nm CMOS FinFET”. In: *2019 IEEE Custom Integrated Circuits Conference (CICC)*. Austin, TX, USA: IEEE, Apr. 2019.

Appendix A

Layout Generation Engines in BAG2 and BAG3

While the schematic generation APIs are mostly identical, there are several layout generation engines in the BAG2 and BAG3 frameworks. Although implemented differently, all layout generation engines rely on the fact that the floorplan of circuits often has a large portion of invariant characteristics, such as the relative location of transistors and routing tracks. Moreover, the complex design rules are hidden under the extra layer of the grid that is specific to different layout generation engines. Layout primitives are developed for different processes to ensure DRC correctness. Different types of engines are explained in this appendix.

A.1 LAYGO

LAYGO stands for LAYout with Gridded Objects. The LAYGO engine adopts an approach similar to LEGO blocks to create a portable layout generator [203]. The layout methodology of LAYGO is shown in Figure A.1. It handles complex design rules using hand-made primitives and predefined routing grids for all possible combinations of layers, spaces, and widths. The individual components, such as transistors with different widths and numbers of fingers, can be compared to LEGO blocks, while the routing grid functions as the LEGO bump. Imagine what would happen if the size of LEGO blocks scales. As long as they are still assembled according to the smaller Lego bumps, the same result can be expected despite the changes in the dimensions. Similarly, in advanced technology nodes, DRC rules become increasingly complex. However, if the predefined unit blocks can capture different rules, the resulting layout will still comply with design rules. Generally speaking, various types of hand-crafted cells capture the most intricate front-end design rules. The metal patterns are wired up following a predefined routing grid with specific spacing, width, and via types. This approach is similar to the design of a digital circuit that uses standard cells. The difference is that various types of devices can be chosen as long as the corresponding unit block templates are implemented.

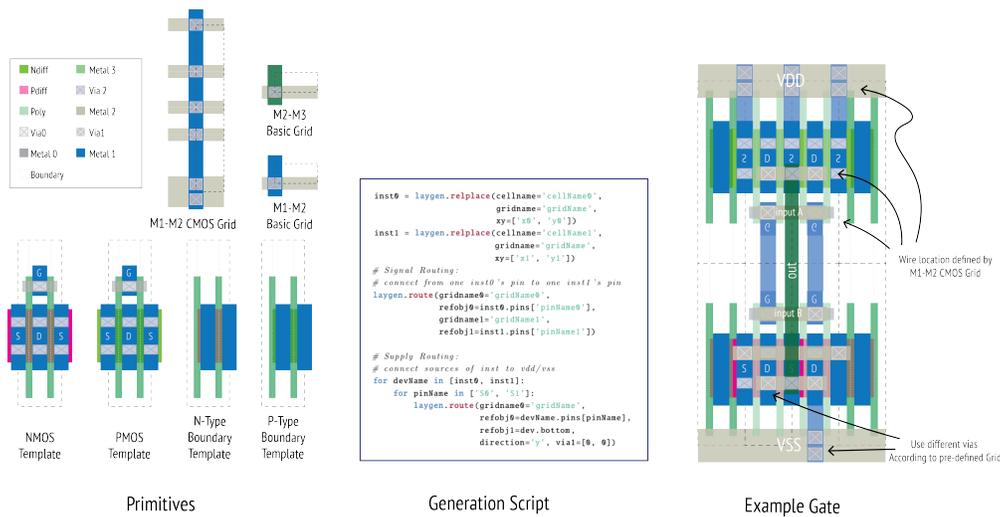


Figure A.1: Illustration of the LAYGO layout generation flow.

In Figure A.1, the bottom left side shows NMOS and PMOS templates with two fingers and a predefined width. Inside these templates, one unit transistor is implemented with a bounding box, which conveniently makes it easy to place in an array. The pins' names, G, D, and S stand for gate, drain, and source, respectively. They are defined in coordinates that are compatible with the corresponding routing grid, which is the M1-M2 CMOS grid in this case. Other types of transistor templates are available to handle various situations. For example, a one-finger transistor with gate connections on the left or right side is used to implement minimum logic gates. Transistor templates are arranged in an array when implementing a layout generator in Python scripts. It is guaranteed that there will be no DRC issue in the middle. As for the boundary rules on two sides should be handled by N-Type and P-Type boundary templates. For example, it left enough space on the sides to ensure that it would not conflict with other primitives. Similar to transistor templates, various boundary cells can handle different scenarios. Besides transistors, other passive devices such as capacitors, diodes, and resistors are also supported in LAYGO.

Some example codes are listed below to demonstrate the LAYGO APIs:

```
#Placement: (x0, y0), (x1, y1) are the origin points where instances are placed
inst0 = laygen.replace(cellname='cellName0', gridname='gridName', xy=[x0, y0])
inst1 = laygen.replace(cellname='cellName1', gridname='gridName', xy=[x1, y1])
#Connection: connect from one inst0's pin to one inst1's pin
laygen.route(gridname0='gridName0', refobj0=inst0.pins['pinName0'],
            gridname1='gridName1', refobj1=inst1.pins['pinName1'])
# Export pin
laygen.pin(name='pinName', gridname='gridName', refobj=inst0)
```

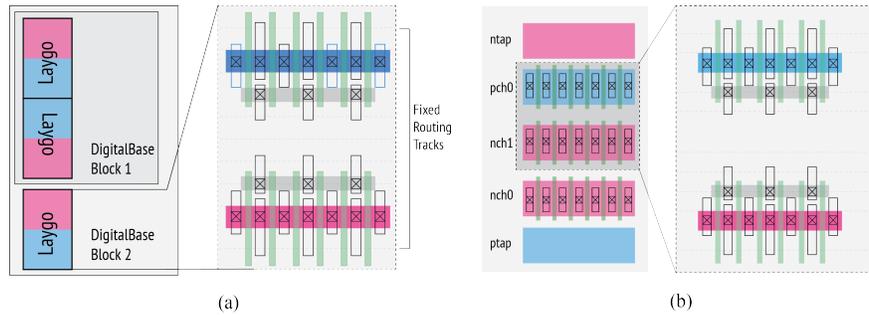


Figure A.2: Illustrations of (a) `DigitalBase` and (b) `AnalogBase` in the framework.

A.2 XBase in BAG2

The `AnalogBase` and `DigitalBase` are abstract classes used for drawing layouts of analog and digital circuits, respectively, in the `XBase` engine. Both are based on the `TemplateBase` introduced in Chapter 3. The primitive generation codes define abstract methods that encapsulate all design rules specific to the supported floorplan. The `AnalogBase` draws multiples of N- and P-type devices' rows, as shown in Figure A.2 (b). Layouts with only one type of device are also possible. However, mixing N- and P-type devices alternatively is not supported and can only be implemented in the `TemplateBase`. This can lead to inconvenience in generator development in some cases. The `DigitalBase` aims to implement a floorplan similar to that of the `LAYGO` engine. The conceptual diagram is shown in Figure A.2 (a). Compared to the `AnalogBase`, only the first available routing layers are included in the primitives, while the `AnalogBase` takes three layers. Alternate N- and P-type devices are possible, making them suitable for compact digital circuit design. Another difference between the `AnalogBase` and the `DigitalBase` is the fixed lower layer available track and track widths and spaces in `DigitalBase`. Some example codes of the `AnalogBase` are shown below. It implements the `draw_layout()` method and calls `draw_base()` to correctly assemble primitives.

```
def draw_layout():
    ...
    tr_manager = TrackManager(grid=self.grid, tr_widths=tr_widths, tr_spaces=tr_spaces)
    self.draw_base(...)
    # Draw transistors
    n_conn = self.draw_mos_conn(mos_type='nch', row_idx=0, col_idx=col0, fg=seg)
    p_conn = self.draw_mos_conn(mos_type='pch', row_idx=1, col_idx=col1, fg=seg)
    # Connections
    # Get track information
    g_tid = self.get_wire_id('pch', 1, 'g', wire_name='in')
    # Connect gates
    self.connect_to_tracks([n_conn['g'], p_conn['g']], g_tid)
    # VDD/VSS
    self.connect_to_substrate('ptap', [n_conn['d']])
```

```

self.connect_to_substrate('ntap', [p_conn['d']])
# fill dummies
tr_w = tr_manager.get_width(hm_layer, 'sup')
vss_warrs, vdd_warrs = self.fill_dummy(vdd_width=tr_w, vss_width=tr_w)

```

In the DigitalBase, the floorplan is set up by `setup_floorplan()`, some example codes are shown below

```

def draw_layout():
...
vss_tid, vdd_tid = self.setup_floorplan(config, row_layout_info, max(fg_p, fg_n))
# Add blocks
p_conn = self.add_laygo_mos(1, 0, seg_p, w=wp, gate_loc='d', stack=stack)
n_conn = self.add_laygo_mos(0, 0, seg_n, w=wn, gate_loc='d', stack=stack)
pin = p_conn['g']
nin = n_conn['g']
# Compute overall block size and fill spaces
self.fill_space()
# connect input
tid = TrackID(hm_layer, in_tid, width=tr_w_in)
in_warr = self.connect_to_tracks([pin, nin], tid, min_len_mode=min_len_mode_inv_in)

```

Some differences exist between the LAYGO and the XBase. First, the layout primitives are all manually implemented, which provides a straightforward way of quick process porting. In contrast, XBase uses abstract base classes to support programmable primitives, which requires a thorough understanding of the design rules and the code construction to implement primitives with similar programmability. Also, although both LAYGO and XBase use a custom layout grid for their track system, different from the predefined grid template in LAYGO, XBase uses `TrackManager` to organize routings. All the routings are assigned different categories with varying widths and spaces, and the generators avoid using hard-coded numbers.

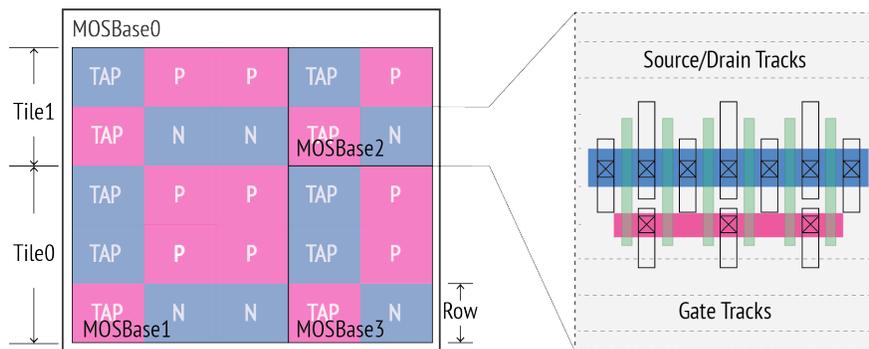


Figure A.3: Illustration of the MOSBase in the BAG3 framework.

A.3 XBase in BAG3

In the BAG3 framework, only the `XBase` engine is used, and the `AnalogBase` and `DigitalBase` converge to the `MOSBase`, which can implement floorplans for both analog and digital circuits. `MOSBase` allows tiling similar blocks together, similar to `DigitalBase`. Moreover, it allows adding substrate connections in the same row as transistors. The transistors' contacts are also usually at the first available routing layer. However, different from the `DigitalBase`, `MOSBase` can specify any combination of tap and transistor rows, and the track arrangements are handled by the `TrackManager`. A conceptual diagram of the `MOSBase` is shown in Figure A.3. The layout floorplan of the `MOSBase` is established by the `draw_base()` method. The Appendix shows two examples of the `MOSBase` code. B.

Appendix B

Generator Examples

This Appendix shows two example codes of an inverter generator and a comparator generator. Both generators are implemented in the BAG3 framework. The proposed ADC generator includes a large number of generators; these two blocks are representative to show the generator implementations in a simple logic and a complex hierarchical generator.

B.1 An Inverter Generator Example



Figure B.1: Example inverter generator in BAG3.

Figure B.1 shows the code for the inverter inside the `draw_layout()` method. The code is divided into multiple parts, marked from 1 to 4, with the generated instances at each step. First, N- and P-type transistors are added at row `ridx_n` and `ridx_p`, respectively. In step 2, the horizontal track index at the output of N- and P-type transistors is derived and used to connect the drains of the transistors. Next, the vertical connection between two transistors is made by first calculating the `vm_tidx` and setting up `vm_tid`. The horizontal outputs are connected to the vertical track by `self.connect_to_tracks`. Finally, the sources of transistors are connected to supply tracks, as shown in the last generated instance.

B.2 Comparator Generator Examples

The comparator generators are introduced in 4.3.1; an example of a self-timed double-tail comparator generator is shown here to illustrate the construction of a relatively complicated generator.

```
class SingleEndTxGroup(MOSBase):

    def __init__(self, temp_db: TemplateDB, params: Param, **kwargs: Any) -> None:
        MOSBase.__init__(self, temp_db, params, **kwargs)

    @classmethod
    def get_params_info(cls) -> Dict[str, str]:
        return dict(...)

    @classmethod
    def get_default_param_values(cls) -> Dict[str, Any]:
        return dict(...)

    def draw_layout(self):
        place_info = MOSBasePlaceInfo.make_place_info(self.grid, self.params['pinfo'])
        self.draw_base(place_info)

        ridx_dict_temp: ImmutableSortedDict[str, int] = self.params['ridx_dict']
        seg_dict: ImmutableSortedDict[str, int] = self.params['seg_dict']
        w_dict: ImmutableSortedDict[str, int] = self.params['w_dict']
        conn_pair_list: ImmutableList[Tuple[str, str]] = self.params['conn_pair_list']
        g_on_s_list: ImmutableList[Tuple[str, str]] = self.params['g_on_s_list']

        # ===== Placement =====
        for k in tx_name_list:
            _col = ndum + ofst_dict.get(k, 0)
            m_dict[k] = self.add_mos(ridx_dict[k], _col, seg_dict[k], w=w_dict[k], g_on_s=k)

        self.set_mos_size(num_cols=max_seg + (max_seg & 1))

        for inst0, inst1 in conn_pair_list:
            inst0_conn = getattr(m_dict[inst0[0]], inst0[1])
            inst1_conn = getattr(m_dict[inst1[0]], inst1[1])
            self.connect_wires([inst0_conn, inst1_conn])

        for k, v in m_dict.items():
            self.add_pin(k + 'd', v.d, hide=True)
            self.add_pin(k + 'g', v.g, hide=True)
            self.add_pin(k + 's', v.s, hide=True)
```

Read parameters

Place Transistor Group

Connect Shared S/D

Export TX group pins

Figure B.2: Generator code of the single-ended transistor group used in the comparator generators.

First, a class called `SingleEndTxGroup` is constructed to represent a group of transistors. In a simple comparator implementation, differential pairs can be placed on two sides of the layout. However, a design that requires higher matching accuracy necessitates a common-centroid arrangement. Both layout strategies can be implemented by partitioning transistors into groups and arranging the groups of transistors directly. In Figure B.2, the `SingleEndTxGroup` class reads in parameters and puts transistors that belong to the same group in a column. Internal connections are made depending on the parameter `conn_pair_list`. The group's ports interface with other portions of the layout and are exported as hidden pins, which can be easily accessed in the higher-level layout generator.

```

class PreAmpMatch(SingleEndTxGroup):
    ...
    inherits from SingleEndTxGroup

    def draw_layout(self):
        place_info = MOSBasePlaceInfo.make_place_info(self.grid, self.params['pinfo'])
        self.draw_base(place_info)

        ngroups = self.params['ngroups']
        ridx_dict: ImmutableSortedDict[str, int] = self.params['ridx_dict']
        seg_dict: ImmutableSortedDict[str, int] = self.params['seg_dict']
        w_dict: ImmutableSortedDict[str, int] = self.params['w_dict']

        flip_np = self.params['flip_np']

        group_params = dict(
            pinfo=self.params['pinfo'],
            seg_dict=seg_dict_group,
            w_dict=w_dict_group,
            ridx_dict=ridx_dict_group,
            conn_pair_list=conn_pair_list,
            g_on_s_list=['in'] if not has_cas_dev else [],
            ndum=0,
        )

        group_temp: MOSBase = self.new_template(params=group_params, temp_cls=SingleEndTxGroup)

        # ===== Placement =====
        m_tail_l = self.add_mos(ridx_dict['tail'], tail_ncol, w=w_dict['tail'],
                               seg=seg_dict['tail'] // 2, g_on_s=True)
        m_tail_r = self.add_mos(ridx_dict['tail'], ncol_tot - tail_ncol, w=w_dict['tail'],
                               seg=seg_dict['tail'] // 2, flip_lr=True, g_on_s=True)

        tx_groups = []
        for idx in range(ngroups // 2):
            cur_col += group_temp.num_cols
            tx_groups.append(self.add_tile(group_temp, 0, cur_col, flip_lr=True))
            cur_col += min_sep
        for idx in range(ngroups // 2, ngroups):
            tx_groups.append(self.add_tile(group_temp, 0, cur_col))
            cur_col += group_temp.num_cols + min_sep

        # ===== Routings =====
        ...
        # ===== Add pins =====
        self.add_pin('inn', inn)
        ...

        self.sch_params = dict(
            ...
        )

```

Setup
SingleEndTxGroup
parameters

Place
shared
transistors

Place
transistor
groups

> class DynLatchMatch(SingleEndTxGroup): -- Similar implementation

Figure B.3: Generator code for the preamplifier.

```

class DoubleTailSelfTimeWrap(MOSBase, TemplateBaseZL):
    ...
    def draw_layout(self):
        pinfo = MOSBasePlaceInfo.make_place_info(self.grid, self.params['pinfo'])
        self.draw_base(pinfo)
        ...
        # ===== Make templates =====
        preamp_params = dict(pinfo=self.get_tile_pinfo(1), fill_dummy=False, has_cas_dev=False,
                             ridx_dict=ridx_dict['preamp'], ngroups=ngroups_dict['preamp'],
                             seg_dict=seg_dict['preamp'], w_dict=w_dict['preamp'], flip_np=flip_np)
        dyn_latch_params = dict(pinfo=self.get_tile_pinfo(3), fill_dummy=False, flip_np=self.params['flip_np'],
                                split_ck=True, ridx_dict=ridx_dict['dyn_latch'], ngroups=ngroups_dict['dyn_latch'],
                                seg_dict=seg_dict['dyn_latch'], w_dict=w_dict['dyn_latch'])
        preamp_master = self.new_template(PreAmpMatch, params=preamp_params)
        dyn_latch_master = self.new_template(DynLatchMatch, params=dyn_latch_params)
        ...
        # Floorplanning
        ...
        preamp_ncol = ...
        dyn_latch_ncol = ...
        # ===== placement =====
        # Place the pre-amplifier and the dynamic latch
        preamp = self.add_tile(preamp_master, 1, (tot_ncol - preamp_ncol) // 2)
        dyn_latch = self.add_tile(dyn_latch_master, 3, (tot_ncol - dyn_latch_ncol) // 2)
        ...
        # Add substrate connection
        tap0_conn = self.add_substrate_contact(0, 0, seg=tot_ncol, tile_idx=0, port_mode=SubPortMode.BOTH)
        tap1_conn = self.add_substrate_contact(0, 0, seg=tot_ncol, tile_idx=2, port_mode=SubPortMode.BOTH)
        tap2_conn = self.add_substrate_contact(0, 0, seg=tot_ncol, tile_idx=4, port_mode=SubPortMode.BOTH)
        ...
        # Fill in dummy structures
        fill_and_collect(self, 1, preamp_p_dum, preamp_n_dum)
        fill_and_collect(self, 3, dyn_latch_p_dum, dyn_latch_n_dum)
        ...
        # ===== Routings =====
        ...
        # ===== Add pins =====
        ...
        self.sch_params = dict(
            preamp=preamp_master.sch_params,
            dyn_latch=dyn_latch_master.sch_params,
        )

```

Figure B.4: Self-timed double-tail comparator generator.

Figure B.3 shows how the preamplifier in a double-tail transistor is constructed from the transistor groups. In this `PreAmpMatch` generator, the current tail transistor is placed in the bottom row of the layout. The tail transistor is partitioned into two parts to facilitate a differential layout. The input and load transistors are placed within transistor groups. First, the `group_params` are constructed from the input parameters. The transistor group template is constructed using the `self.new_template()` and it is placed using the `self.add_tile`. The dynamic latch class `DynLatchMatch` in the second stage of the double-tail comparator is constructed similarly. Lastly, the `DoubleTailSelfTimeWrap` class, shown in Figure B.4, takes two layout generators and assembles them into a full comparator. The templates are constructed first. Then, the templates can be used to perform floorplanning, as all the information associated with each layout class is now accessible. Then, both templates are placed, and substrate connections are added. The unused space is filled for both DRC and matching purposes. Two generated examples with one and four transistor groups are shown separately in Figures B.5 and B.6.

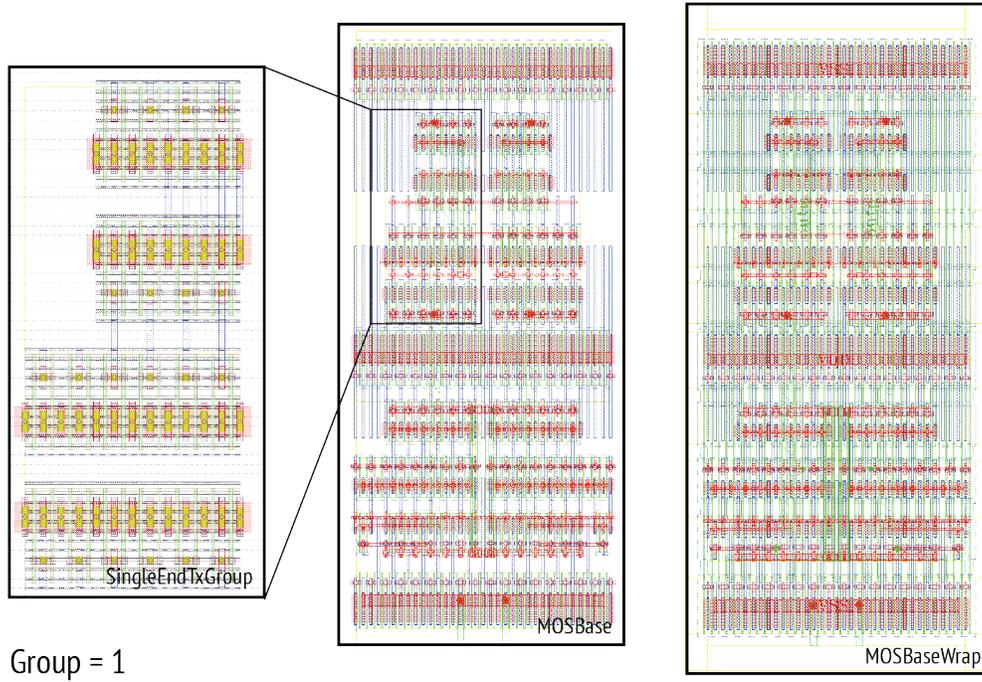


Figure B.5: Generated instance of a simple differential comparator in the `cds-ff-mpt` process.

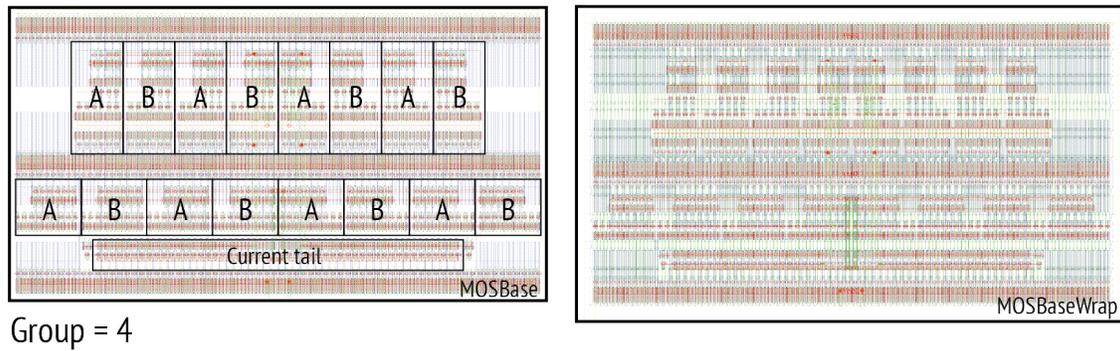


Figure B.6: Generated instance of a comparator with multiple groups in the `cds-ff-mpt` process.