

UC Berkeley

UC Berkeley Electronic Theses and Dissertations

Title

Design Techniques for Ultra-High-Speed Time-Interleaved Analog-to-Digital Converters

Permalink

<https://escholarship.org/uc/item/2ft3480b>

Author

Duan, Yida

Publication Date

2015

Peer reviewed|Thesis/dissertation

Design Techniques for Ultra-High-Speed Time-Interleaved Analog-to-Digital Converters

(ADCs)

By

Yida Duan

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 Elad Alon, Chair

Professor Ali M. Niknejad

Professor Paul K. Wright

Spring 2015

Design Techniques for Ultra-High-Speed Time-Interleaved Analog-to-Digital Converters
(ADCs)

Copyright 2015
by
Yida Duan

Abstract

Design Techniques for Ultra High Speed Analog-to-Digital Converters

by

Yida Duan

Doctor of Philosophy in Engineering – Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Elad Alon, Chair

Analog-to-Digital Converters (ADCs) serve as the interfaces between the analog natural world and the binary world of computer data. Due to this essential role, ADC circuits have been well studied over 40 years, and many problems associated with them have already been solved. However in recent years, a new species of ADCs has appeared, and since then attracted lots of attention. These are ultra-high-speed (often greater than 40GS/s) time-interleaved ADCs of low or medium resolution (around 6 to 8 bit) built in CMOS processes. Although such ADCs can be used in high-speed electronic measurement equipment and radar systems, the recent driving force behind them is next generation 100Gbps/400Gbps fiber optical transceivers. These transceivers take advantage of ultra-high-speed ADCs and digital-signal-processors (DSPs) to enable ultra-high data-rate communications in long-haul networks (city-to-city, transcontinental, and transoceanic fiber links), metro networks (fibers that connect enterprises in metropolitan areas), and data centers (fiber links within data center infrastructures). At such high sampling rate, massively time-interleaved successive-approximation ADC (SAR ADC) architecture has emerged as the dominant solution due to its excellent power efficiency. Several recent works has demonstrated success in achieving high sampling rate. However, the sampling network has become the bottleneck that limits the input bandwidth in these ADCs. It is apparent that conventional switch-based track-and-hold (T&H) circuit cannot satisfy the >20GHz bandwidth requirement. In addition, it is unclear what the optimal interleaving configuration is. Each state-of-the-art design adopts a different interleaving configuration – from straightforward conventional 1-rank interleaving to 2-rank hierarchical sampling or even 3 ranks. How to partition interleaving factors among different ranks has not yet been investigated. Furthermore, asynchronous SAR sub-ADCs are often used in these designs to push the sampling rate even further. The well-known sparkle-code issues caused by comparator meta-stability in asynchronous SARs can significantly increase the Bit-Error-Rate (BER) of the transceivers unless power hungry error correction coding are implemented in the system. Although many works in the literature attempted to deal with the meta-stability in asynchronous SARs, the effectiveness of these approaches have not been fully demonstrated. In this thesis, I will first propose a new cascode-based T&H circuits to improve the ADC bandwidth beyond the limit of conventional switch-based T&H circuits. Then, a system design and optimization methodology of hierarchical time-interleaved sampling network is presented in the context of cascode T&H. To deal with sparkle-code issue in asynchronous SAR sub-ADCs, a new back-end meta-stability correction technique is employed. An extensive statistical analysis is provided to verify the correction algorithm can greatly reduce sparkle-code error-rates. To further demonstrate the effectiveness of the proposed circuits and techniques, two prototype ADCs have

been implemented. The first 7b 12.5GS/s hierarchically time-interleaved ADC in 65nm CMOS process demonstrates 29.4dB SNDR and >25GHz bandwidth. The later 6b 46GS/s ADC in 28nm CMOS employs asynchronous SAR sub-ADC design with back-end meta-stability correction. The measurement results show it achieves sparkle-code error free operation over $1e10$ samples in addition to achieving >23GHz bandwidth and 25.2dB SNDR. The power consumption is 381mW from 1.05V/1.6V supplies, and the FOM is 0.56pJ/conversion-step.

To my mother

To Wen

Contents

Contents	ii
List of Figures	iv
List of Tables	vi
1. Introduction	1
1.1 Thesis organization	3
2. High-speed sampling	4
2.1 Track-and-hold circuits	4
2.1.1 Switch-based T&H circuits	5
2.1.2 Cascode-based T&H circuits	7
2.1.3 Mitigation of non-idealities in T&H circuits	9
2.2 Hierarchical time-interleaving	16
2.2.1 Optimization of hierarchical sampling network	19
2.3 Implementation and measurement results of the 7b 12.8GS/s ADC	22
3. High-speed power-efficient sub-ADC	25
3.1 SAR sub-ADC	26
3.1.1 Synchronous and asynchronous SAR	27
3.1.2 Comparator meta-stability and sparkle-codes	31
3.2 Back-end meta-stability correction	34
3.2.1 Meta-stability correction by detect-then-stop method	35
3.2.2 Back-end meta-stability correction	37
3.2.3 Statistical analysis of sparkle-code error-rate	40
4. 6b 40GS/s hierarchically time-interleaved asynchronous SAR ADC	46
4.1 System overview	46
4.2 Sampling circuits	48
4.3 Clock generation and distribution	49
4.3.1 Frequency dividers	49
4.3.2 Phase interpolator and duty-cycle correction circuits	51
4.3.3 Variable delay-line	52
4.4 Sub-ADC implementation	53
4.5 Measurement results and discussion	57
5. Conclusion	64
5.1 Summary	64
5.2 Future work	65

Bibliography

List of Figures

1.1	Applications for ultra-high-speed ADCs	1
1.2	Cumulative distribution function (CDF) of conversion error of a common ADC	2
2.1	(a) Schematic and (b) small signal model of conventional T&H circuit	5
2.2	The speed-power trade-off curves for (a) switch-based T&H circuit and (b) cascode-based T&H circuit	6
2.3	(a) Schematic and (b) small-signal model of cascode-based T&H circuit	7
2.4	Schematic of a 2x time-interleaved differential cascode T&H	8
2.5	Schematic and functional block diagram of cascode samplers with (a) triode PMOS load, and (b) saturation NMOS load	10
2.6	Input of a differential pair vs. (a) normalized trans-conductance and (b) HD_3 for different over-drive voltages	10
2.7	normalized input frequency vs. HD_3 for a 400mV peak-to-peak input sinewave and different overdrive voltages (V_{ov})	12
2.8	Illustration of charge injection and clock-feedthrough in cascode-based T&H with (a) triode PMOS load and (b) saturation NMOS load	13
2.9	illustration of signal-feedthrough issue in (a) cascode T&H and (b) cascode T&H with feedthrough cancellation	14
2.10	Layout of cascode NMOS (a) with large C_{ds} and (b) with minimum C_{ds}	14
2.11	(a) Schematic of cascode T&H with replica MOM C_{ds} feedthrough cancellation (b) layout of replica MOM C_{ds}	15
2.12	Conventional time-interleaved ADC	16
2.13	Block diagram of a 3 rank hierarchically time-interleaved ADC	17
2.14	Timing diagram of a 3 rank hierarchically time-interleaved ADC	18
2.15	Block diagram of a general N-rank hierarchically time-interleaved ADC	19
2.16	Block diagram of the 7b 12.8GS/s ADC in 65nm CMOS	23
2.17	the input frequency vs. (a) normalized output amplitude, and (b) SNDR of the 7b 12.8GS/s ADC	24
3.1	Illustration of the conversion algorithm of 6-bit SAR ADC	26
3.2	(a) schematic and (b) timing diagram of a 6-bit synchronous SAR ADC	27
3.3	(a) Schematic of a typical Strongarm comparator, and (b) its output waveforms	28
3.4	(a) Schematic and (b) timing diagram of classic asynchronous SAR ADC	30
3.5	SAR ADC waveforms for a scenario that results in a sparkle-code	32
3.6	ADC input vs. normalized total regeneration time for a 6-bit asynchronous SAR ADC with 400mV input swing and 1.05V supply	34
3.7	Estimated sparkle-code error-rate vs. normalized additional conversion time for a 6-bit asynchronous SAR ADC with 400mV input swing and 1.05V supply	34
3.8	Waveform of a classic asynchronous SAR ADC in the scenario that (a) a meta-stability event does not cause a sparkle-code and (b) a meta-stability event caused a sparkle-code	35
3.9	Schematic of asynchronous SAR ADC with detect-then-stop sparkle-code correction method	36

3.10	Waveform of detect-then-stop meta-stability correction circuit in the scenario of (a) the meta-stability is correctly detected, and (b) the detection circuit goes meta-stable	36
3.11	Schematic asynchronous SAR ADC with back-end meta-stability correction circuit	37
3.12	Waveform of the back-end meta-stability correction circuit when the first correction bit ($L_m < 5 >$) goes meta-stable	38
3.13	Waveform of (a) a false-positive and (b) false-negative detection error	40
3.14	The PDF of comparator input for all 6 bit-cycles	44
3.15	(a) PDF and (b) CDF of the ADC residue error	44
3.16	Additional conversion time vs. the sparkle-code error-rate for asynchronous SAR ADC with and without back-end correction	45
4.1	(a) block diagram and (b) timing diagram of the ADC chip	46
4.2	The implementation of Rk-1 cascode T&H circuit	48
4.3	Schematic of front-end frequency divider (FD_1)	49
4.4	Schematic of the first CMOS frequency divider (FD_2)	50
4.5	Schematic of the front-end phase interpolator (PI_1) and duty-cycle correction circuits (DCC)	51
4.6	Schematic of variable-delay-line	52
4.7	Complete schematic of 6-bit asynchronous SAR sub-ADC	53
4.8	Comparator schematic	54
4.9	Schematic of (a) SAR logic cell (LC) and (b) meta-stability cell (MC) used in the SAR ADC	55
4.10	die photo	57
4.11	Test setup	58
4.12	ADC output spectrum using 3600-pt FFT for (a) a 5.5GHz input tone, and (b) a 23.5GHz input tone	59
4.13	Input frequency vs. normalized ADC output amplitude	60
4.14	ADC input frequency vs. SNDR, SFDR and SNR	60
4.15	Examples of captured sparkle-codes	61
4.16	Detailed power breakdown of the ADC chip	62

List of Tables

I. Comparison table

63

Acknowledgements

Spending a decade at UC Berkeley is a long time, even for people like me who refuse to graduate. It is exactly a third of my life now. Many things happened in this decade: besides gaining fifty pounds then losing fifty pounds then gaining back fifty pounds and the numerous sleepless nights stressed out for tape-outs, I generally had an enjoyable decade. I enjoyed my Berkeley decade not because of the tasty local food or the interesting local people or the magnificent protests, but because of the great colleagues, awesome friends, and the tremendously knowledgeable and kind mentors without whom I will not be who I am right now. Here are the people to whom I owe my sincere gratitude.

Needlessly to say, I would first and foremost thank my advisor and friend, Prof. Elad Alon. He is not only vastly knowledgeable, but also extremely kind and considerate. Regardless how busy he was, he always found time for in-depth technical discussions with me and my fellow students. He has supported me throughout my ph.D. career, despite many of my progress-less years. Under his guidance, I have become an analog/mixed-signal circuit designer. I owe my career to him.

Second, I would sincerely thank my long time mentor and friend, Prof. Ali M. Niknejad. I have learned so much through his lectures and office hours. I enjoyed so much having technical discussions with him. He has given me great guidance in my ph.D. career. He is not just a vastly knowledgeable professor, a great educator, and a kindly humble person, but also a tremendous soccer player. I still remember his powerful headers vividly. I would also like to thank him for organizing the amazing “Soccer and Circuits” retreat, during which I had so much fun.

In addition, I would like to thank Prof. Vladimir Stejanovic for his guidance and support during my last year at UC Berkeley. He is such a nice and easy going person with so many crazy and interesting ideas. I wish I could have more time to work with him. I would also like to thank Prof. Bernhard E. Boser, who first brought me into the world of the data converters and taught me so much about them.

Then, I would also like to thank Prof. Paul K. Wright for being so kind to be a member of my dissertation committee. I would like to thank Prof. Seth Sanders and Prof. Liwei Lin, for being members of my qualification exam committee and suffered 2 hours of my exhausting presentation.

Furthermore, I would also like to thank my friends and colleagues, to whom I wasn't always nice to: Jiashu Chen, Lingkai Kong, Yue Lu, Paul Liu, Jaeduk Han, Dusan Stepanovic, Jun-Chau Chien, Nai-Chung Kuo, Chintan Thakkar, Kwangmo Jung, John Crossley, Hanh-Phuc Le, Steven Callendar, Ping-Chen Huang, Wenting Zhou, Shinwon Kang, Jaehwa Kwak, Lu Ye, Jung-Dong Park, Siva Thyagarajan, Chen Sun, Krishna Settaluri, Sen Lin, Eric Chang, Nathan Narevsky, Brian Zimmer, Kosta Trotskovsky, Wei-hung Chen, Debopriyo Chowdhury, Amin Arbabian, Ehsan Adabi, Zhiming Deng, Chinwuba Ezekwe ... I would also like to thank BWRC staffs, especially Fred Burghardt, Tom Boot, Sarah Jordan, and Olivia Nolan for their support in the office and lab.

At last, I would like to thank many great mentors from industrial and academia whom I was fortunate to get to know. I would like to thank Dr. Qiong Wu, and Kevin Mahooti of NXP

semiconductor, Dr. Xiaoye (Sean) Wang, Gene Lin, and Dr. Li Lin of Marvell semiconductor, Dr. Ken Chang of Xilinx inc., Prof. Boris Murmann of Stanford University, Prof. Un-Ku Moon of Oregon State University, Dr. Robert Neff and Dr. Ken Nishmura of Keysight, Dr. Stephane Le Tual, and Dr. Andreia Cathelin of ST Microelectronics, Dr. James Gorecki and Lawrence Tsai of Inphi Corporation.

Chapter 1

Introduction

An Analog-to-Digital Converter (ADC) is a basic circuit block that typically converts a continuous-time analog input signal into samples of quantized binary data that can be processed by digital processors. Throughout history, ADCs played important roles in a wide range of electronic systems. Recently, there are particularly high demands for ADCs of extremely high sampling rate, usually $>40\text{GS/s}$. This is driven by applications such as high-speed electronic measurement systems, radars, and most recently, the next-generation of optical transceivers. Electronic measurement systems such as real-time oscilloscopes use high-speed ADCs to acquire input analog waveform and accurately present it to the user with high fidelity. Modern radar systems employ ADC/DSP for reliable detection. In order to satisfy exponentially growing demands for data consumption, 100Gbps/400Gbps fiber optics transceivers employ high-order modulation schemes to push the data-rate beyond the bandwidth of existing fibers and optical components [1], [2], [3]. For example, DP-QPSK (Dual-Polarization Quadrature-Phase-Shift-Keying) has been deployed recently in most existing long-haul/metro fiber links, with 16-QAM expected to follow in a few years, fiber links within mega data-centers are currently moving toward 4-PAM solutions – pulse amplitude modulation with 4 levels. These modulation schemes rely on ADC/DSP for demodulation, channel equalization, and clock-data recovery. In all these application, using CMOS technologies is very beneficial not only due to their advantage in cost and power, but also because many DSP functions can be readily integrated along with the ADC to process its raw data before sending the useful information off-chip. Medium resolution between 6b and 8b is usually sufficient for these applications.

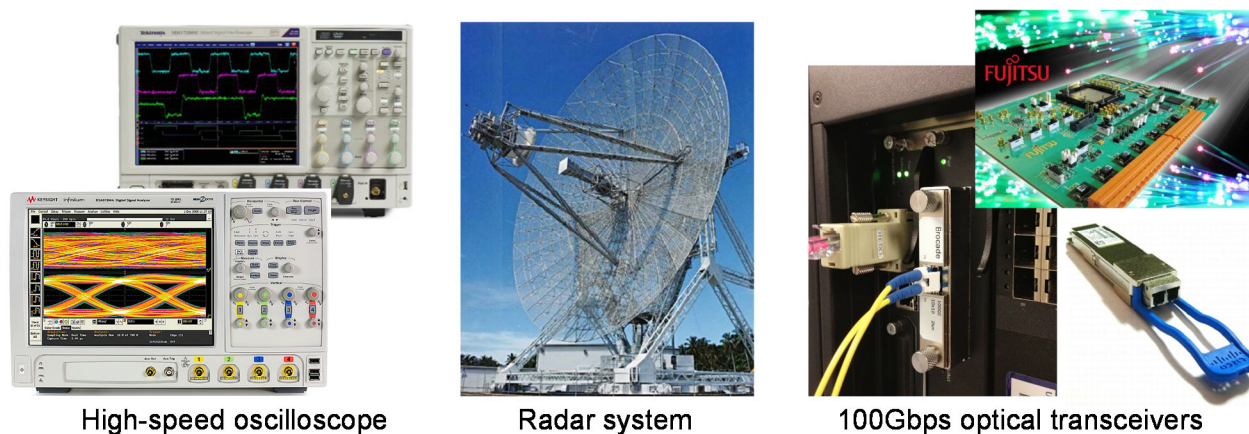


Figure 1.1 Applications for ultra-high-speed ADCs

Previous works [1], [2], [3], [4], have demonstrated that massively time-interleaved Successive-Approximation-Registers (SAR) ADCs are able to achieve $>40\text{GS/s}$ in CMOS with reasonable Figure-Of-Merit (FOM). $>40\text{GS/s}$ sampling rate has been achieved with 16 or more SAR sub-ADC channels. However, their bandwidths are below 20GHz – much lower than the Nyquist-rate. This bandwidth limitation is due to several issues associated with sampling. First, the input of a traditional time-interleaved ADC is directly connected to a large number of parallel switches [1] [5] [6]. The ADC input capacitance caused by the parasitic capacitors of all these switches can be quite large, and the low-pass filter formed by this ADC input capacitance with the 25Ω equivalent input resistance – the 50Ω termination resistor in parallel with the 50Ω cable/transmission line characteristic impedance – may have a low cut-off frequency. To make things worse, since all the sampling switches track a continuous changing analog input, the sampling clocks that drive these switches must satisfy stringent jitter requirement to avoid SNDR attenuation at high frequencies. Unless extra power is spent to distribute these many sampling clocks with low jitter, the Effective-Resolution-Bandwidth (ERBW) of the ADC can be even lower.

In addition, the input signal must pass through the sampling switch before charging the sampling capacitor in conventional track-and-hold circuits. The sampling capacitor sees the on resistance of the switch in series with the output impedance of the circuits in front, which may be the output impedances of the ADC driver [1] [6], or in some cases, another sampling switch [4]. The series resistance penalty of these switches can significantly lower the track-and-hold bandwidth. Once the sampling is performed, the discrete-time interleaved samples must be quantized by sub-ADCs, which introduce more errors to the system. Other than quantization noise, thermal/flicker noise, large amplitude error may occur as a result of meta-stability of the decision circuit inside the sub-ADCs. These rarely occurring meta-stability-induced errors – sparkle codes – have a non-Gaussian distribution profile, and may have much higher error-rate than the Gaussian distributed thermal/flicker noise at large amplitude (Figure 1.2). Electronic measurement systems and radars require extremely low error-rate (beyond $1e-15$) to avoid any false measurement results or detection [7]. As a result, excessive power may be spent to just to keep the sparkle-code error rate below the acceptable levels.

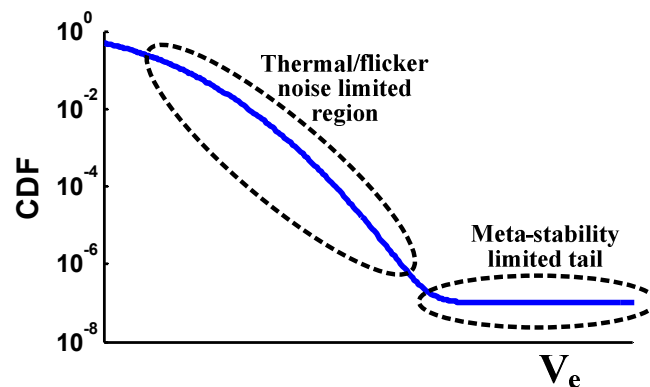


Figure 1.2 Cumulative distribution function (CDF) of conversion error of a common ADC

This thesis investigates the BW limitation and sparkle-code error-rate issue in ultra-high-speed time-interleaved ADCs. Novel sampling architectures such as hierarchical time-interleaving that enables high sampling rate with good power efficiency is also studied. Circuit techniques such as cascode T&H circuits are proposed to alleviate the BW limitation. An optimization method for hierarchical sampling network with cascode T&H circuit is presented. A new back-end meta-stability correction circuit is developed to reduce sparkle-code error rate for asynchronous SAR sub-ADCs. A 6b 46GS/s prototype ADC is fabricated in 28nm Fully-Depleted Silicon-on-Insulator (FDSOI) process to demonstrate the results.

1.1 Thesis Organization

Despite the numerous architecture and implementations, all ADCs perform 2 basic functions: sampling and quantization. Thus, the discussion in this thesis is divided into 2 parts: Chapter 2 starts the discussion from sampling, especially focusing on high speed sampling. After a brief review of sampling errors, track-and-hold bandwidth, and time-interleaving, the limitations of conventional switch-based samplers are introduced. Then, a new cascode-based sampling circuit is proposed. The concept of time-interleaving is also reviewed, and the technique of hierarchical time-interleaved sampling is studied. Later in the chapter, a general optimization method for hierarchically time-interleaved sampler networks using cascode sampler is presented. At the end of the chapter, the implementation and measurement results of the 7b 12.8GS/s ADC test-chip is presented to demonstrate the effectiveness of these techniques. Chapter 3 focuses on the other aspect of ADCs: quantization. After a brief discussion of different quantizer (sub-ADC) architectures, the 2 popular types of SAR architecture (synchronous and asynchronous) are introduced. Meta-stability caused sparkle-code is extensively analyzed in the context of SAR ADC, and a new back-end meta-stability correction circuit to reduce sparkle-code error rate is propose. A statistical analysis and associated measurements are provided to demonstrate the effectiveness of the proposed correction technique. After the theoretical discussions and analysis, chapter 4 presents the detailed implementation and measurement results of the prototype 6b 46GS/s hierarchically time-interleaved ADC. Finally, chapter 4 concludes this thesis and provides a brief discussion of future works.

Chapter 2

High Speed Sampling

Sampling is the process that converts a continuous-time signal into discrete-time samples. The output samples of an ideal sampler exactly equals its input at the respective sampling instances. In reality however, a practical sampler usually suffers from finite bandwidth – i.e., the signal is low-pass filtered before the samples are taken. This can significantly attenuate signal amplitude at high frequencies. Random jitter in the sampling clock can move sampling instances away from its ideal position, which corrupts the samples for fast changing inputs. The sampler can also give rise to nonlinear distortion depending on its implementation.

In ultra-high-speed ADCs, sampler design is extremely challenging. As the sampling time (T_s) is reduce to approach the rise/fall time of digital gates, all practical sampler implementations fail. This can only be resolved by time-interleaving, which takes advantage of parallelism: many lower speed samplers take turns to sample the input, and together achieve high aggregate throughput. Even if the sampling rate of the samplers can be brought down significantly by time-interleaving, broad bandwidth must be maintained in each individual sampler. The following section starts the sampler study from the designs of ultra-high bandwidth sampling circuits.

2.1 Track-and-hold circuit

Electronic samplers are implemented by track-and-hold circuits. In CMOS processes, the most common way to build track-and-hold circuit is the single transistor switch, as shown in Figure 2.1a. Like digital latch circuits, track-and-hold circuits have two phases defined by high and low state of the sampling clock: a transparent phase (track time) and an opaque phase (hold time). A unity gain buffer usually precedes the track-and-hold switch. Although it requires static power consumption, this buffer is necessary in most ADC systems because it serves several important purposes: first, it reduces the kick-back effect to ADC input from the sampling clock. The rising or falling edge of the sampling causes a glitch at the input side of the sampling switch due to capacitive coupling by the parasitic gate to source/drain capacitance of the transistor. In fiber optic transceivers, this glitch may be reflected back-and-forth between the trans-impedance amplifier (TIA, usually reside in another chip in front of the ADC chip) and the ADC resulting in long-term post-cursor inter-symbol-interference (ISI). The workload of DSP must be increased to cancel this post-cursor, thus power consumption must be increased. Even without direct capacitive feed-through from the sampling clock, the time-variant input capacitance of the sampling switch itself causes time-variant reflection coefficient at the ADC input. In addition to kick-back reduction, the buffer isolates the sampling capacitor from the ADC input, and reduces the input capacitances

of the ADC. Unlike low-speed counter parts, the input capacitances of ultra-high-speed ADCs must be very small in order not to limit the bandwidth. As mentioned earlier, the 25Ω equivalent input impedance and the ADC input capacitance form a low pass filter, and the cut-off frequency of the filter must be higher than the bandwidth to avoid high frequency attenuation. Furthermore, embedded ADCs are often preceded by other circuits such as continuous-time linear equalizer (CTLE), variable-gain amplifier (VGA), etc. A large ADC input capacitance imposes stringent requirement on the driving capabilities of its preceding circuits, and causes significant increase in power consumption. In section 2.1.1, we start the discussion from analyzing issues of conventional switch-based track-and-hold circuit.

2.1.1 Switch-based track-and-hold

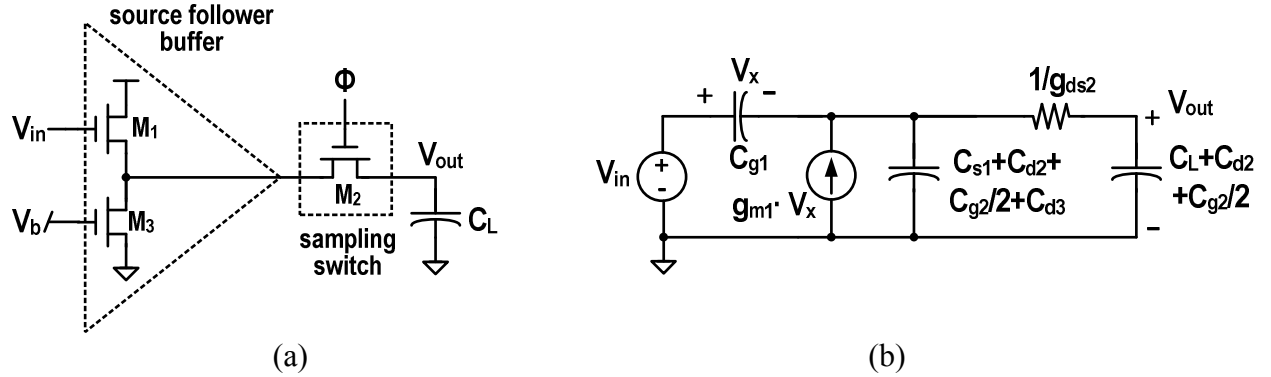


Figure 2.1 (a) Schematic and (b) small signal model of conventional T&H circuit

A conventional track-and-hold circuit (Figure 2.1) consists of a source follower buffer combined with a series sampling switch. As mentioned earlier, the front-end buffer is necessary to reduce kick-back and input capacitance. The final load capacitance C_L is thus driven by the sum of the output resistance of the source follower and the switch resistance. Even if bootstrapped [8] [9], the on-resistance of the sampling switch can be still a significant fraction of the total impedance seen by C_L for broadband designs. This series configuration of resistors makes the conventional sampling circuit very power-inefficient in high speed designs. In order to rigorously highlight this issue, we can use a small signal model (Figure 2.1b) to analyze the power-speed trade off.

To provide some quantitative insights, we will assume all NMOS's has the same f_T ¹ and the f_T of all the PMOS transistors is half that of the NMOS transistors, and that the ratio between $C_{d,s}$ and C_g is 1 for all transistors. We will further assume (as is the case in most CMOS technologies nodes from 65nm down to 20nm) that the maximum triode g_{ds} of a transistor is roughly twice the maximum saturation g_m . With all of these assumptions combined, if f_T is the unity current-gain frequency of all the NMOS transistors, then $g_{ds3}/C_{g3} = 2 \cdot 2\pi f_T$. Finally, g_{ds3} is equal to g_{m1} for unity DC gain. Using first-order moment matching, the dominant pole of the conventional T&H circuit can be estimated as:

¹ Although it is not strictly optimal for the current sources device (M_3) as f_T can be scaled larger with smaller size, the headroom limitation in low supply modern process usually limits the degree of its downsizing.

$$P_1 \cong \frac{1}{\frac{C_{g1} + C_{s1} + C_{d3} + C_{d2} + C_{s2} + C_{g2} + C_L}{g_{m1}} + \frac{C_L + C_{d2} + C_{g2}/2}{g_{ds2}}} \quad (2.1)$$

Utilizing the assumptions as stated earlier, (2.1) becomes:

$$P_1 = \frac{2\pi f_T}{\frac{15}{4} + 3\beta + \left(\frac{1}{2\beta} + 1\right) \frac{2\pi f_T C_L}{g_{m1}}} \quad (2.2)$$

Where $\beta = W_2/W_1$ is the ratio of the widths of M_2 and M_1 . The dominant pole achieves its maximum value when $\beta = 1/\sqrt{3g_{m1}/\pi f_T C_L}$, and this optimal P_1 is:

$$\frac{P_1}{2\pi f_T} = \frac{1}{\frac{15}{4} + \frac{2\pi f_T C_L}{g_{m1}} + 2\sqrt{\frac{3}{2}} \cdot \frac{2\pi f_T C_L}{g_{m1}}} \quad (2.3)$$

Equation (2.3) relates the normalized bandwidth of the T&H, $P_1/2\pi f_T$, to the normalized trans-conductance of the source follower buffer, $g_{m1}/2\pi f_T C_L$. Since the trans-conductance is directly proportional to the static current consumption, equation (2.3) represents the speed-power trade-off of the T&H circuit. This trade-off curve is plotted in Figure 2.2. For low frequency designs ($P_1 \ll 2\pi f_T$), the T&H power consumption scales linearly with bandwidth. As the design bandwidth approaches a significant fraction of f_T ($P_1 \sim \frac{1}{10} f_T$), the trade-off quickly bends upwards, as increasingly more power is required for every small increment in bandwidth. This is due to capacitive self-loading of the source follower buffer and sampling switch, and the switch resistance penalty makes this trade-off extremely power inefficient. As the ADC bandwidth requirement approaches >20GHz, the switch-based sampler quickly becomes unviable.

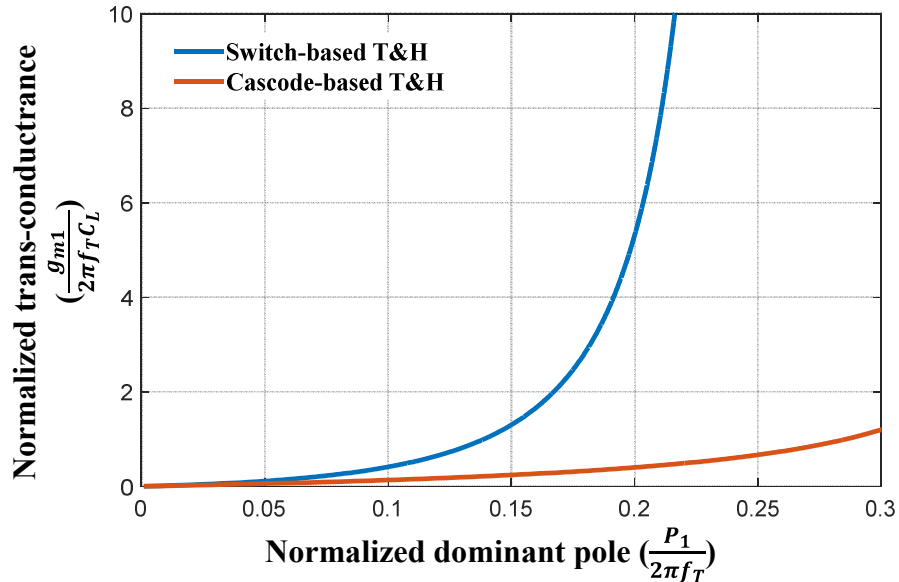


Figure 2.2 The speed-power trade-off curves for (a) switch-based T&H circuit and (b) cascode-based T&H circuit

Although power inefficient, switch-based T&H circuit is widely used in medium/low speed applications. This is due to its excellent linearity performance. The source follower buffer can easily achieve >50dB spurious-free-dynamic-range (SFDR) over the entire bandwidth due to its inherent internal feed-back. Although charge injection, non-zero fall time, and voltage-dependent switch resistances, can cause sampling errors, the error voltages these effects create are for the most part linearly dependent on input voltage. Thus, they only cause a gain error to the first order, which can be easily fixed by gain calibration in the DSP. The slight higher-order dependence of these effects on input voltage is usually not significant enough to degrade SNDR for low to medium resolution ADCs.

2.1.2 Cascode-based track-and-hold

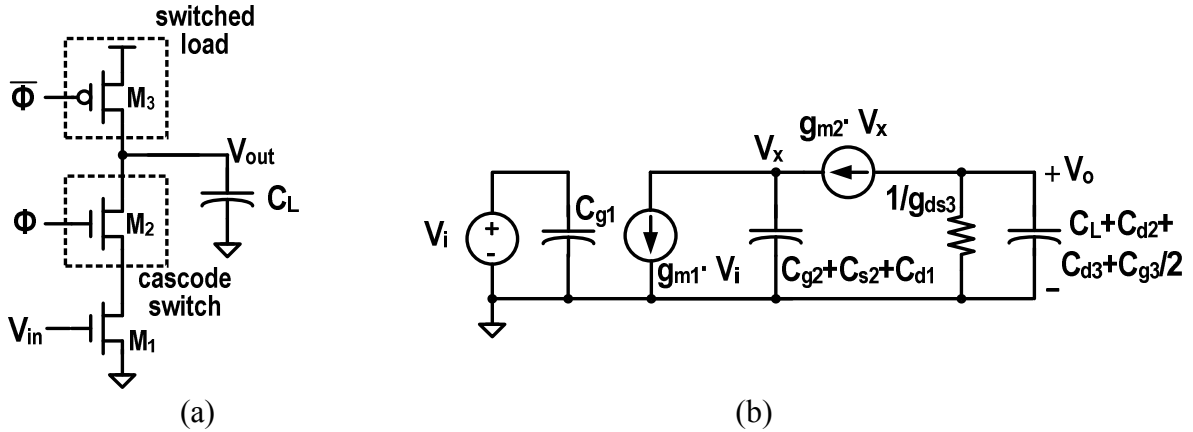


Figure 2.3 (a) Schematic and (b) small-signal model of cascode-based T&H circuit

In order to mitigate the penalty caused by the series resistance of the sampling switch and hence improve the tradeoff between sampling speed and power consumption, we propose a cascode T&H circuit that merges the sampling operation into the buffer itself [10] [11]. A single-ended version of the proposed cascode T&H schematic and its equivalent small-signal model are shown in Figure 2.3. During the track phase when Φ is high, $M_{1,2,3}$ form a cascode common-source amplifier, with in this case, the PMOS M_3 acting as a triode load resistor. It worth mentioning that the load device does not necessarily have to be PMOS, and as discussed later, a NMOS load device might be preferred depending on the design requirements. M_1 and M_3 are sized to provide a DC gain of ~ 1 . During the hold phase when Φ is low, both M_2 and M_3 are cut-off and the output voltage is held on C_L . The key advantage of this design is that as long as the cascode device (M_2) operates in saturation and has sufficiently high f_T relative to the operating rate, the dominant pole of the circuit is set only by the output node resistance and capacitance. In other words, in contrast to the traditional sampling circuit, the addition of the sampling switch does not directly affect the settling time.

Similar to the analysis of switch-based T&H, we can use small signal models (Figure 2.3b) to estimate the dominant pole for the cascode T&H circuit:

$$P_1 = -\frac{g_{ds3}}{C_L + C_{d2} + C_{d3} + C_{g3}/2} \quad (2.4)$$

Using the earlier assumptions for switch-based T&H, (2.4) can be rewritten as:

$$\frac{P_1}{2\pi f_T} = \frac{1}{\frac{2\pi f_T C_L}{g_{m1}} + \frac{5}{2}} \quad (2.5)$$

With equation (2.5) in hand, we can plot the trade-off between g_m and the bandwidth for cascode-based T&H along with conventional T&H on the Figure 2.2. Notice that the advantage of the cascode sampler is most apparent when the circuit bandwidth approaches a significant fraction of f_T (but remains well below f_T so that the source node of the cascode is still relatively fast). Specifically, for $P_1 = \frac{1}{8} f_T$ – which is $\sim 30\text{GHz}$ in a typical 28nm process – the conventional sampler requires more than four times higher g_m (and hence power) than the proposed cascode sampler².

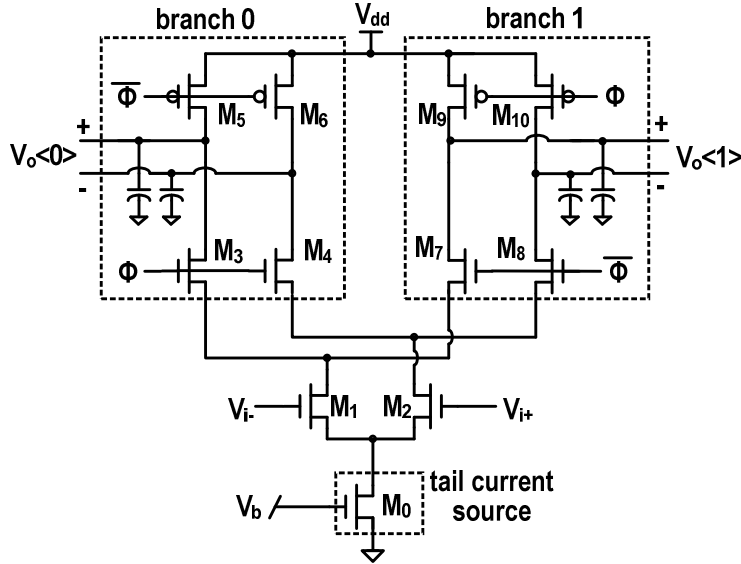


Figure 2.4 Schematic of a 2x time-interleaved differential cascode T&H

As shown in Figure 2.4, a differential cascode-based T&H circuit also includes a tail current source device (M_0) to reject input common mode variation. The example has 2 differential current branches (branch 0 and branch 1) to track the input signal in both high and low clock phases. For example, when Φ is low, $V_o\langle 0 \rangle$ is held on the sampling capacitors, and branch 0 is disabled. During this inactive period of branch 0, the current of the differential pair is steered to branch 1 to enable it to track the input at $V_o\langle 0 \rangle$. This way, 1-to-2 way de-multiplexing function can be implemented, and hence 2X higher sampling speed can be achieved without increasing static power consumption. In addition, utilizing the complementary clock phases can also keep the input capacitance of the T&H constant for both high and low clock phases. In general, 1-to-N de-multiplexor (or N-way time-interleaved sampler) requires N cascode branches and N phases of non-overlapping clocks with $1/N$ duty-cycle. As the interleaving factor N grows large to achieve higher aggregate sampling rate, the metal routing to reach each sub-ADC also linearly increases

² This analysis is based on schematic simulation. The parasitic in post-layout usually makes the advantages of cascode-based T&H circuit even greater.

as the pitch of each sub-ADC is fixed by the technology and its load capacitance. Even if top metal layer is used for routing, it may still cause significant parasitic capacitance – in some cases it can be even larger than the sampling capacitors. The buffer of the conventional switch-based T&H circuit must drive this parasitic capacitor as well as the sampling capacitor, thus making it even more power inefficient. In contrast, this parasitic capacitor is naturally mitigated in the cascode-based T&H circuit, because the metal routing is connected to the low impedance source nodes of the cascode devices ($M_{3,4,7,8}$) and results in the 2nd pole at:

$$P_2 = \frac{g_m}{N \cdot C_d + C_g + C_p} \quad (2.6)$$

Where g_m , C_g , and C_d are the trans-conductance, drain junction, and gate capacitances of the cascode devices; C_p is the parasitic capacitances of the wire routing. Using the assumption stated as the beginning of this section, equation (2.6) becomes:

$$P_2 = \frac{f_T}{N + \frac{C_w}{g_m}} \quad (2.7)$$

To guarantee sufficient settling time or tracking bandwidth, the second pole must be larger than the first pole, $P_1 \gg P_2$. Given a technology, this would require g_m to be sufficiently high, and N to be not too large. When designing a cascode-based T&H circuit, we suggest start from equation 2.5 to calculate g_m for a given load capacitor. Assuming the casocde device has the same size of the differential pair, equation 2.7 can be used to obtain maximum interleaving factor (or demultiplexing factor) per T&H circuit.

2.1.3 Mitigation of non-idealities in T&H circuits

Other than tracking bandwidth and settling time, practical switch-based T&H circuits are also affected by many non-ideal buffer and switch behaviors, such as nonlinear distortion, charge injection, clock and hold-time signal feedthrough. Similarly, the cascode-based T&H circuit also faces their issues. In some cases, the device non-idealities can cause more severe problems in cascode T&H than the switch T&H. In this section, we will analyze these non-ideal behaviors in the context of cascode-based T&H circuit and discuss methods to mitigate these issues.

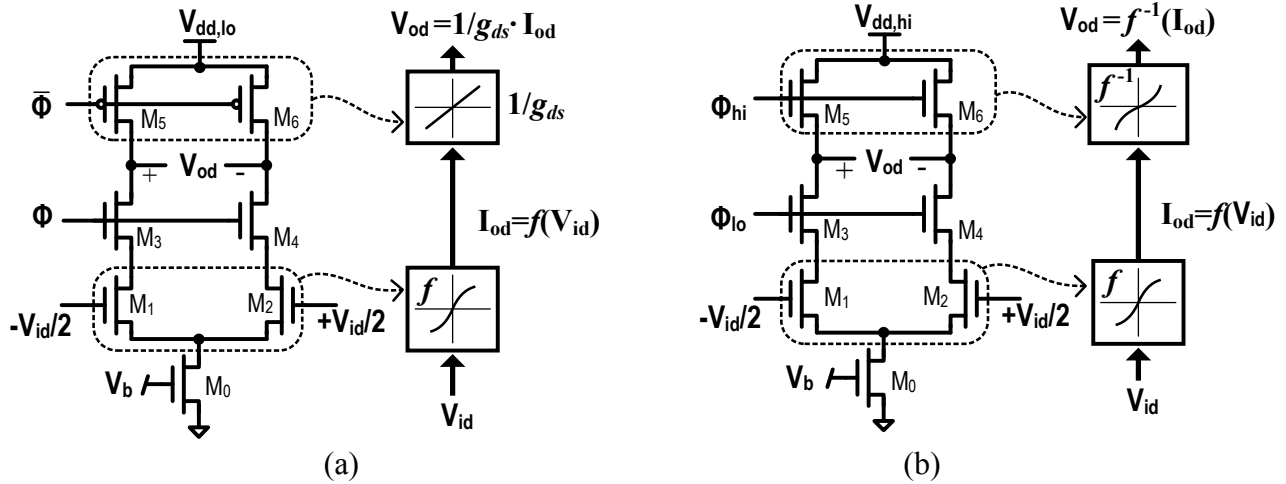


Figure 2.5 Schematic and functional block diagram of cascode samplers with (a) triode PMOS load, and (b) saturation NMOS load

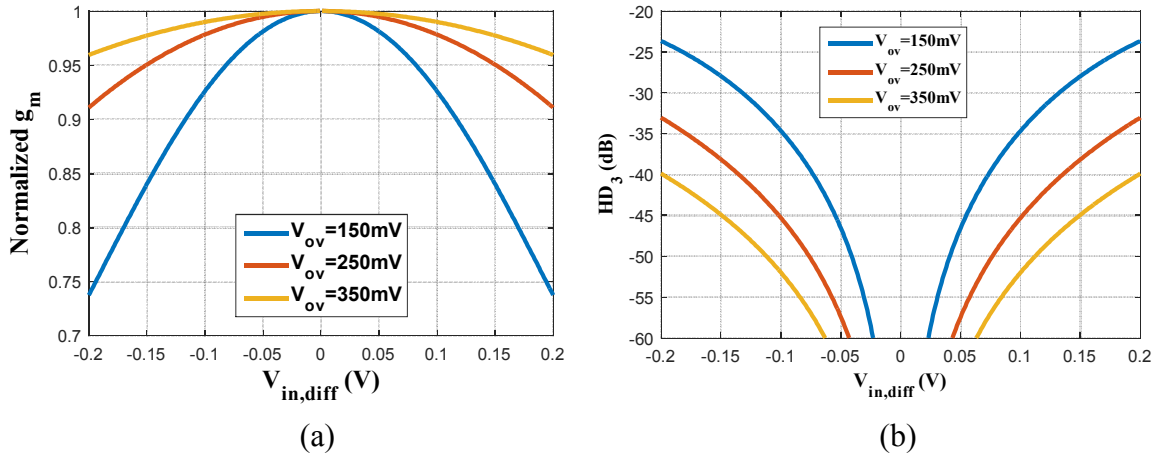


Figure 2.6 Input of a differential pair vs. (a) normalized trans-conductance and (b) HD_3 for different over-drive voltages

First, the speed advantage of the cascode sampling structure does not come without expense. In particular, while in the conventional switch-based design linearity is significantly improved by the internal feedback of the source follower circuit, the cascode T&H with PMOS load illustrated in Figure 2.5a is an open loop structure, and thus suffers from distortion due to inherent nonlinearity of MOSFET devices. Since g_{ds} of the triode PMOS loads $M_{3,4}$ is as a fairly constant conductance, the dominant source of non-linearity is g_m of the differential pair $M_{1,2}$. Thus, one can simply examine the transfer characteristics of a differential pair to predict the distortion of the cascode T&H circuit. As shown in Figure 2.6a, the variation in large-signal G_m with large differential input amplitude gives rise to third-order distortion (HD_3). Assuming only HD_3 is present – which is a good approximation for differential circuit with moderate signal swing – the large-signal G_m can be modeled as a function of input voltage:

$$G_m = G_{m0} - \frac{\Delta G_m}{V_{sw}^2} \cdot V_{in}^2 \quad (2.8)$$

Where V_{sw} is the peak differential input swing, G_{m0} is the trans-conductance at $V_{in} = 0$ – the same as small-signal g_m , and ΔG_m is the change in trans-conductance as the input increases from 0 to the peak swing. For a sinewave input with amplitude $V_{in,diff}$, $V_{in} = V_{in,diff} \sin(\omega t)$, the large signal output current is:

$$I_{od} = G_{m0} \cdot V_{in,diff} \sin(\omega t) - \Delta G_m \cdot V_{in,diff}^2 \sin^3(\omega t) \quad (2.9)$$

Thus, the differential output voltage is:

$$V_{od} \approx \frac{1}{g_{ds}} [G_{m0} \cdot V_{in,diff} \sin(\omega t) - \Delta G_m \cdot V_{in,diff}^2 \sin^3(\omega t)] \quad (2.10)$$

Using Equation 2.10, the HD₃ caused by input-dependent large-signal G_m can be approximated as:

$$HD_3 \approx 20 \cdot \log_{10} \left(\frac{\Delta G_m}{4G_{m0}} \right) \quad (2.11)$$

According to Equation 11, the HD₃ of the cascode T&H is directly related to the percentage deviation of the large-signal trans-conductance at the amplitude from the small-signal g_m . The normalized trans-conductance of a differential pair ($G_m(V_{in,diff})/G_{m0}$) vs. differential input amplitude ($V_{in,diff}$) curves for different overdrive voltages for a 28nm technology are shown in Figure 2.6a. As the input swing becomes smaller, the deviation of its full-swing G_m from its peak value G_{m0} is reduced. Thus, a straightforward way to improve HD₃ is to reduce input swing at the expense of reduced signal-to-noise ratio (SNR). With a given input amplitude, one also has the option to reduce the HD₃ by increasing the overdrive voltage, V_{ov} . However, increasing V_{ov} has the side effects of reducing power efficiency and output headroom. To design a cascode sampler, one has to carefully choose $V_{in,diff}$ and V_{ov} to satisfy the HD₃ requirement. In a typical 28nm process, with a V_{ov} of 350mV for the input transistors and a moderate $V_{in,diff}$ of ~200mV (peak-to-peak voltage swing of 400mV), the HD₃ is well below -40dBc, which is sufficient for a 6-bit design. An attractive feature of using triode PMOS load is that the voltage drop across the load can be quite small. Therefore, to keep a stack of only three transistors ($M_{0,1,3}/M_{0,2,4}$) in saturation, core supply voltage (~1V) can be used to reduce power.

In addition to using brute force to reduce HD₃, a more sophisticated way is to replace the switched triode PMOS load in the cascode T&H with switched saturation NMOS load, as shown in Figure 2.5b. As long as $M_{5,6}$ is matched to $M_{1,2}$, the nonlinear output current produced by the large-signal G_m of $M_{1,2}$ is inverted by the same large-signal G_m of $M_{5,6}$. In reality, although g_{ds} -modulation, device mismatch, and body-effect limits the achievable HD₃ in this circuit, simulation has shown it has at least 8dB better linearity than triode PMOS load. Additional benefit of using cascode T&H with saturation NMOS loads are better supply rejection ratio and less clocking power consumption due to single-phase clock. To keep all transistors in saturation and reduce the effect of g_{ds} -modulation over the entire output swing, a high supply voltage of 1.6V is required. In addition, the gate voltage V_g of the NMOS load and the cascode switch must be level shifted above 1V to keep the cascode devices in saturation; chapter 3 will discuss details on how to level-shift V_g 's to appropriate levels. When the cascode T&H with NMOS load is used as the front-end T&H circuit, one must pay special attention to frequency-dependent effects of HD₃. As shown in

Figure 2.7, although nonlinearity inversion by the NMOS loads greatly improve HD_3 at very low frequency, the HD_3 may quickly degrades as frequency increases. This dependence causes a bowl-shaped HD_3 curve, and the worst-case HD_3 is slightly lower than $1/3 \cdot BW$. This can be intuitively understood as frequency increases, the impedance of the load capacitor becomes smaller, and more output current is shunted into the load capacitor. The part of output current shunted into the load capacitor does not get inverted by the matched NMOS load, and thus raising HD_3 . As the frequency increases beyond $1/3 \cdot BW$, the HD_3 is higher than the T&H BW and gets filtered out by the inherent low-pass filter. Since this distortion term has memory effect, analyzing it mathematically requires to invoke Volterra series expansion, therefore can be cumbersome. To study the effect of frequency dependent linearity reduction for the cascode T&H with NMOS load, large signal transient simulations are used. As shown in Figure 2.7, with 200mV peak amplitude and 350mV overdrive voltage, the cascode T&H with NMOS load achieves $>-40\text{dB}$ HD_3 over the entire bandwidth. At this point, it worth pointing out that the rank-2 and rank-3 sample-and-hold circuits in a hierarchically time-interleaved sampling network track the output of front-end (rank-1) T&H – a constant voltage during the entire tracking phase, which will be discussed in the next section. As long as the output voltage of these circuits settles, the HD_3 does not degrade from this. Therefore, cascode T&H circuit with NMOS load can be used to achieve its full linearity potential as analog de-multiplexors after the signal is sampled.

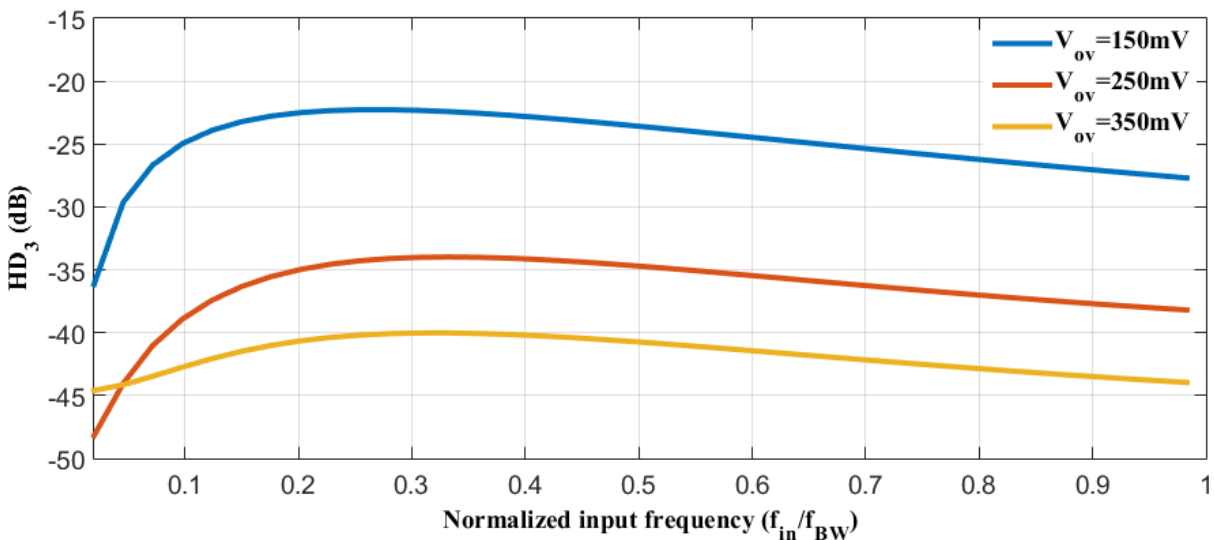


Figure 2.7 Normalized input frequency vs. HD_3 for a 400mV peak-to-peak input sinewave and different overdrive voltages (V_{ov})

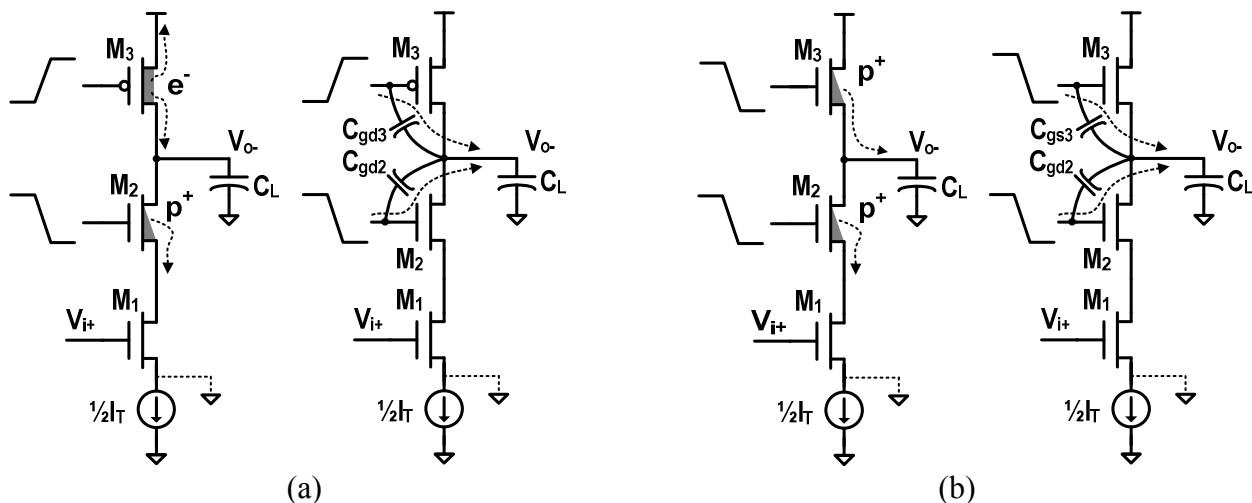


Figure 2.8 Illustration of charge injection and clock-feedthrough in cascode-based T&H with (a) triode PMOS load and (b) saturation NMOS load

In addition to linearity issues, just like conventional switch-based T&H, the “top plate sampling” used by the cascode T&H is prone to signal-dependent noise, and we need to analyze issues such as charge injection and clock/signal feed-through and make sure their effects do not degrade the SNDR. In the case of conventional sampling circuit, inversion charge in the sampling switch can flow to the sampling node when the switch is opened, causing signal-dependent voltage error [12]. Fortunately, such problem does not exist in cascode T&H with triode PMOS load to the first order. Since the cascode devices $M_{3,4}$ are in saturation during track time (Figure 2.8a), their channels are “pinched off” at the drain nodes (output nodes). Most of inversion charges will inject into the source node because they are distributed close to the source node [13]. Therefore, the inversion charges in $M_{3,4}$ do not affect the sampled output voltage when they are turned off. The only remaining potential source of charge injection error in the cascode sampler circuit is from the triode PMOS loads. Although some of the signal-dependent inversion charge in these devices will transfer to the output when they are turned off, this effect does not necessarily degrade SNDR. Specifically, the linearly dependent inversion charge merely causes a gain error; it is only the nonlinearly dependent inversion charge that gives rise to distortion. Fortunately, as verified by SPICE simulations, this nonlinearly dependent portion of the inversion charge in the PMOS loads is not significant enough to be a concern for medium resolution designs.

On the other hand, the problem of clock feed-through in cascode T&H with PMOS load is similar to conventional T&H. The single-end output voltage change caused by clock feed-through is: $\Delta V_o = (-V_{t,n}C_{gd,n} + V_{t,p}C_{gd,p})/C_L$. This is merely a common-mode shift and do not cause SNDR degradation if the threshold and C_{gd} of transistors are constant. To make things better, the PMOS load and NMOS cascode devices are drive by opposite clock phases, and the feedthrough effects from these devices partially cancel each other. The resulting common-mode shift is usually too small to cause any problem for the circuits it drives. In reality, the slight signal-dependence in threshold voltage and C_{gd} can slightly degrade SFDR, but this dependence is so small that it does not cause any linearity problem for medium resolution ADCs. This is verified with SPICE simulation. Similarly, the cascode devices $M_{3,4}$ in cascode T&H with saturation NMOS load do not inject charges to the output (Figure 2.8b). However, nearly all channel charges in the NMOS load flows to the output at the falling edge of the clock. Although this charge injection might not

cause significant distortion, the resulting T&H gain attenuation is more severe than the triode PMOS load. The differential voltage error caused charge injection of the NMOS loads is approximately $\Delta V_o = -C_{gs}/C_L \cdot V_{in}$. For very high speed designs, C_L can be as small as C_{gs} of the load devices, so charge injection may cause quite a lot signal attenuation. One way to alleviate this issue is to add dummy switches at the output to suck out the injected charges, but doing so adds more capacitance at the output load and reduce the bandwidth. A simple but effective approach is to increase the gain during track time by slightly reducing the width of load NMOS to compensate the attenuation caused by charge injection. To design the cascode T&H with NMOS load, its T&H gain must be verified by transient simulation.

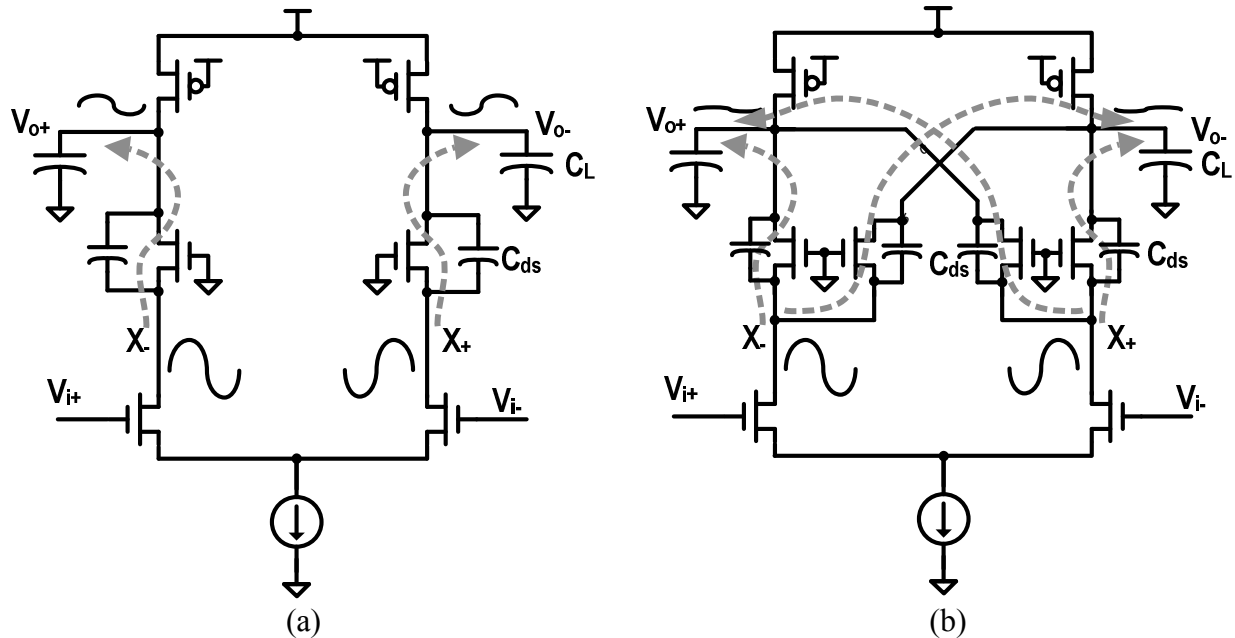


Figure 2.9 Illustration of signal-feedthrough issue in cascode T&H (a) without and (b) with feedthrough cancellation

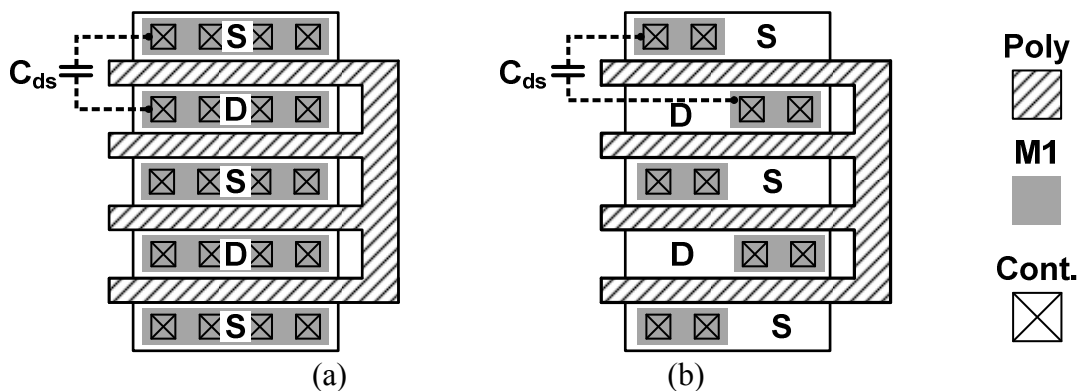


Figure 2.10 Layout of cascode NMOS (a) with large C_{ds} and (b) with minimum C_{ds}

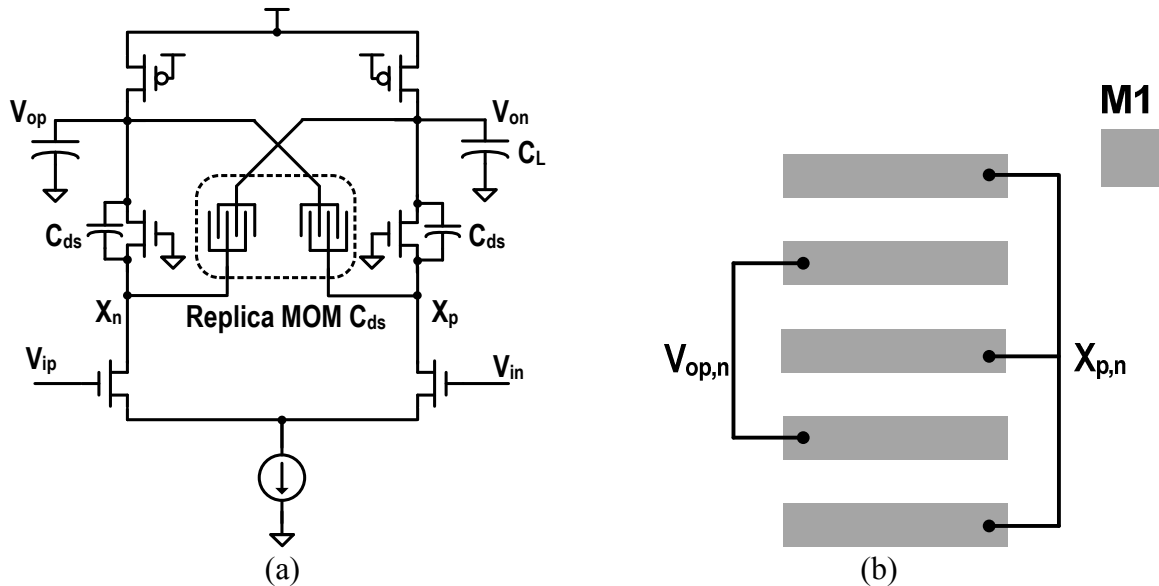


Figure 2.11 (a) Schematic of cascode T&H with replica MOM C_{ds} feedthrough cancellation (b) the layout of the replica MOM C_{ds}

Another problem in T&H circuit is the hold-time signal feed-through caused by capacitive coupling from the source nodes of $M_{3,4}$ ($X_{+,-}$) and the output nodes ($V_{o+,-}$) through C_{ds} when the sampler is in hold mode (Figure 2.9a). The magnitude of this error is proportional to $\frac{C_{ds}}{C_L}$. This is a well-known issue of the time-interleaved ADCs with top-plate sampling structure. A common solution is to cancel the feed-through by adding dummy transistors that cross couple $X_{+,-}$ and $V_{o+,-}$, as shown in Figure 2.9b [14]. However, the drain/source junction capacitance added by the dummy transistors in this approach can reduce the speed of the sampler. To mitigate the effect of signal feed-through without sacrificing speed, we introduce 2 feed-through reduction methods that take advantage of advanced layout. First, we can directly reduce C_{ds} by appropriately laying out the devices [11], [15]. In particular, instead of the typical layout shown in Figure 10a, one can minimize the overlap between the source/drain contacting regions as shown in Figure 10b. The only downside of this layout strategy is increased contact resistance to the source/drain, but in many designs/processes, the resulting effect on the bandwidth of the buffer is negligible. Post-layout simulations indicate that this layout technique achieves a more than 10X reduction of C_{ds} . If increasing contact resistance is not tolerable or the design rule does not allow Metal-1 drain/source stripes to have unequal length, one can also use fringe capacitor made by replica Metal-1 stripes instead a dummy transistor to cancel feed-through (Figure 2.11b). This replica C_{ds} mimics the actual transistor C_{ds} without adding parasitic drain/source to bulk/gate capacitance. Although the replica Metal-1 capacitor may not exactly match C_{ds} of the transistor due to the fringing field from Metal-1 to polysilicon gate, the amount of reduction by this approximate C_{ds} is usually more than enough for the 6-bit design.

To demonstrate effectiveness of cascode samplers, a 7b 12.8GS/s ADC with PMOS loadtest chip is fabri

2.2 Hierarchical time-interleaving

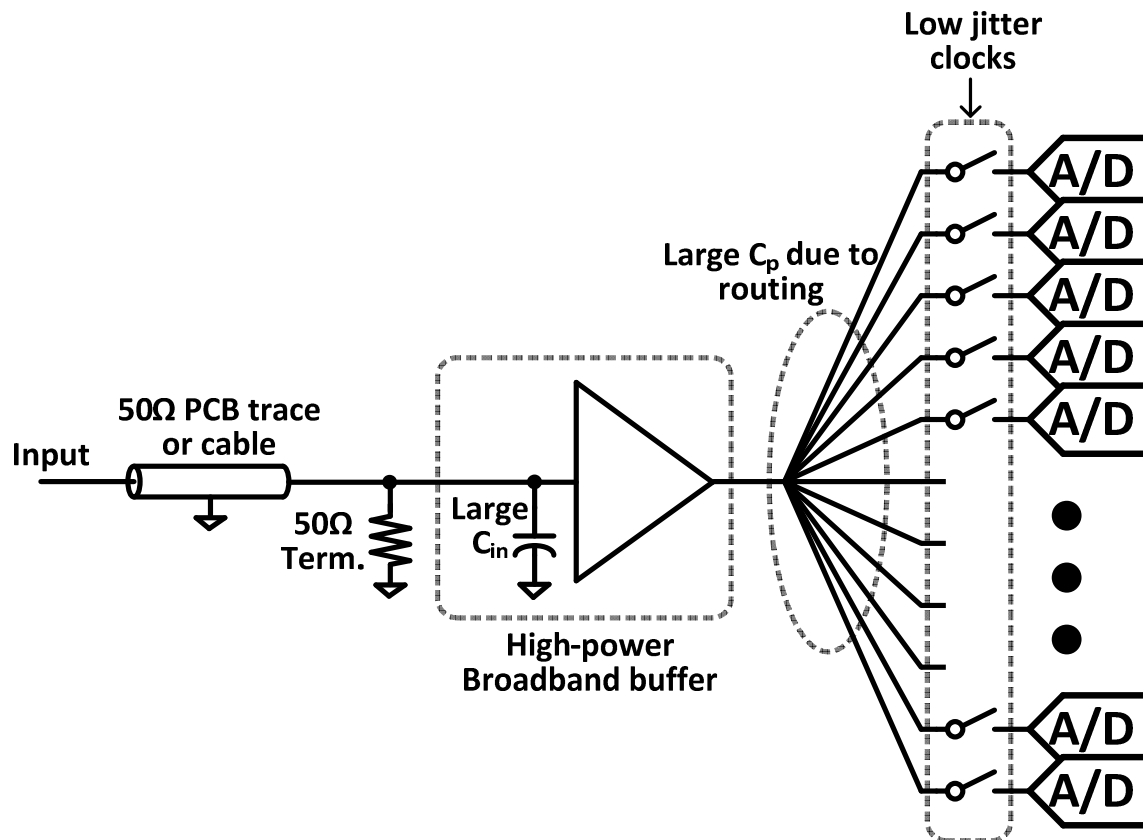


Figure 2.12 Conventional time-interleaved ADC

As mentioned earlier and highlighted in Figure 2.12, the conventional time-interleaved ADC consists of a broadband input buffer that drives many parallel sampling switches followed by sub-ADCs. 3 issues make this structure infeasible for ultra-high-speed designs. First, since all the sampling switches directly sample the continuous changing analog output of the buffer, jitter in all the sampling clocks may translate into sampled voltage error, therefore degrading SNR at high frequencies. For example, to limit jitter induced noise below the quantization noise level for a 6-bit 50GS/s ADC at Nyquist, the sampling clock jitter must be below 81fs. To meet such stringent jitter requirement, many stages of large CMOS gates (often >10μm in width) must be used to distribute the sampling clocks in order to reduce added jitter and to keep sharp sampling edges for better supply noise rejection. As a result, excessive amount of power is spent just to distribute these many sampling clocks across a long distance to reach the sub-ADCs. Second, the cluster of routing at the output node of the buffer adds a large parasitic capacitor in addition to the parasitic capacitors of the large number sampling switches. In many cases, the total parasitic capacitance at the output of the buffer can be orders of magnitude larger than the input capacitor of the sub-ADCs. Therefore, in order to maintain the ADC bandwidth, lots of power in the buffer is wasted to drive these “useless” parasitic capacitors. At last, the large-size broadband input buffer may have a large input capacitor, and the low-pass filter formed by the input capacitor of the buffer and the equivalent input resistance adds a 2nd pole in addition to the pole caused by the track-and-hold

circuits, which further reduces the ADC bandwidth. In a standalone ADC module, the input resistance is limited by the termination resistance and characteristic impedance of the transmission line of the PCB trace or the cable, which is usually 250 Ω (500 Ω in parallel with 500 Ω). For a Nyquist rate 50GS/s ADC, if the T&H circuit (buffer with the sampling switches) is designed to have a pole at 30GHz, the total ADC input capacitor including the input capacitor of the buffer and ESD capacitor (if any) must be < 83fF to maintain the overall ADC bandwidth above 25GHz. This is an extremely stringent requirement. In the case of embedded ADCs, the ADC is preceded by driver circuits such as TIA and CTLE. In this case, the input pole is formed by the input capacitor of the buffer and the output impedance of the driver circuits. A large buffer input capacitance imposes extremely stringent requirement on the driver circuits, and can make the overall system solution infeasible.

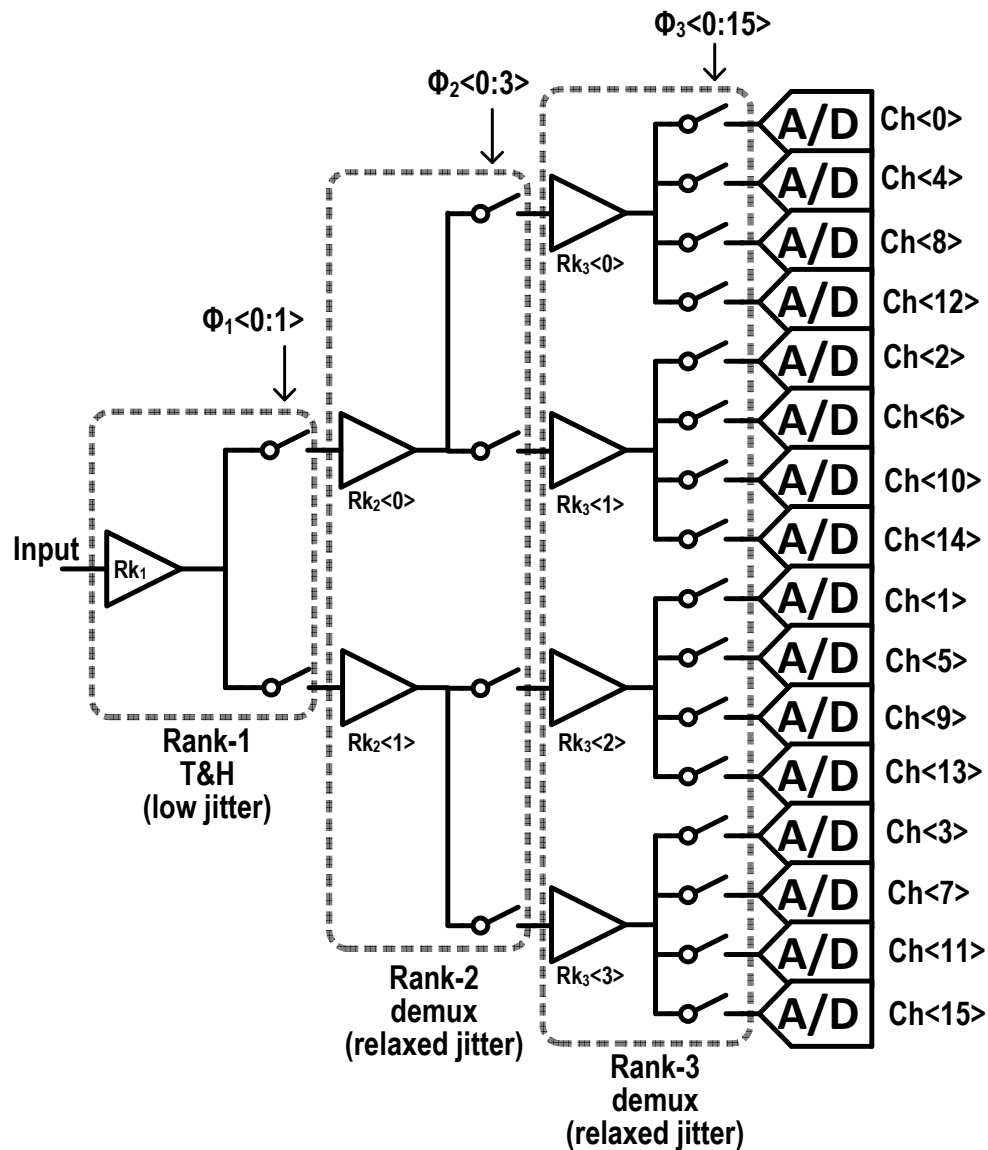


Figure 2.13 Block diagram of a 3 rank hierarchically time-interleaved ADC

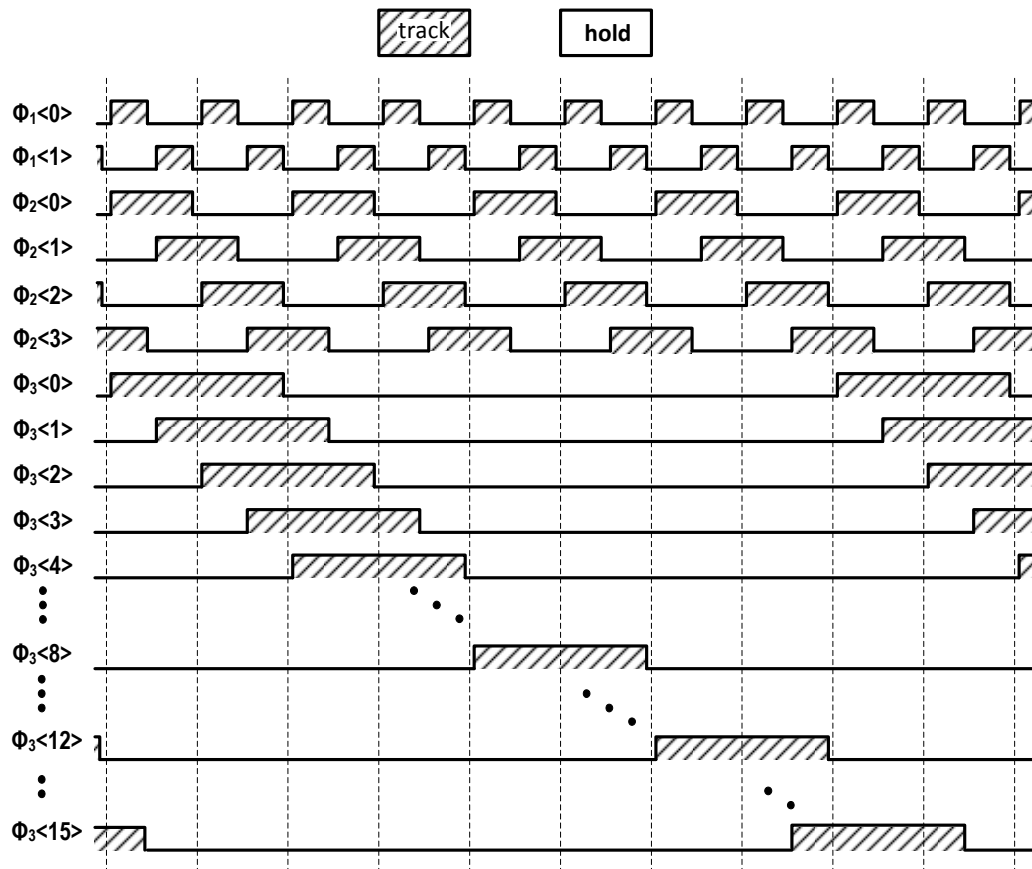


Figure 2.14 Timing diagram of a 3 rank hierarchically time-interleaved ADC

An elegant way to alleviate all these issues is hierarchical time-interleaving [4], [14], [11], [15], [10], [16], [17], [18], [19]. An example design adopting a hierarchical time-interleaving approach is shown in Figure 2.13. Figure 2.14 illustrates its timing diagram. The example has 3-ranks of samplers and a total of 16 sub-ADC channels. The rank-1 T&H circuit (Rk-1) is 2-way time-interleaved in order to reduce the number of required low-jitter clocks that needs be generated and distributed. Once the continuously changing input voltage is sampled and held by rank-1 T&H, the output of this circuit is a constant voltage during the entire hold time. Thus, any perturbation of sampling clock edges at the Rank-2 sampler ($\Phi_2\langle 0:3 \rangle$) do not directly translate into voltage errors as long as it is within this hold window, allowing the jitter requirements for the Rank-2 and subsequent ranks of samplers to be greatly relaxed. As a result, the only jitter-critical clock in the entire sampler system is the 2 sampling clocks of the front-end sampler, Φ_1 . An additional benefit of hierarchical sampling is the greatly reduced signal routing at the output of the front-end buffer; since it can limit the input bandwidth of the entire ADC, the bandwidth of this buffer is critical. As opposed to conventional time-interleaved ADCs where the input buffer must fan-out to all sub-ADCs (in this example, 16 sub-ADCs), the front-end sampler drives only the next rank (Rk-2) of samplers/de-multiplexors (in this case, a single Rank-2 demux.), thus substantially reducing the parasitic capacitance at the output of the front-end sampler. Furthermore, by utilizing a small interleaving factor at the front-end, the errors caused by sampling time mismatch between subsequent sub-channels are almost entirely removed (as long as the intermediate stages settle), eliminating the amount of sampling clocks that require precise timing calibration [18]. As shown

in Figure 2.14, the duty-cycle of sampling clocks for rank-2 and rank-3 ($\Phi_2\langle 0:3 \rangle$ and $\Phi_3\langle 0:15 \rangle$) are set such that only 1 interleaved switch following the inter-rank buffer is closed at any given time. For example, the sampling clocks associated with sampler $Rk_3\langle 0 \rangle - \Phi_3\langle 0 \rangle$, $\Phi_3\langle 4 \rangle$, $\Phi_3\langle 8 \rangle$, and $\Phi_3\langle 12 \rangle$ – are non-overlapping clocks with 25% duty-cycle. This 25% duty-cycle clocking scheme for Φ_3 is the most efficient because each Rk_3 buffer “sees” only 1 sub-ADC at a time instead of 2 sub-ADCs if 50% duty-cycle clock is used. Thus, the driving capability (g_m) of Rk_3 buffer in the case of 25% duty-cycle clock can be made 2X smaller than the case of 50% duty-cycle clock. Note the duration of track time becomes longer as the analog signal propagates from Rk_1 to subsequent ranks. Therefore, the settling requirement for later ranks can be greatly relaxed to save power. The next section will provide a detailed analysis on how to size the T&H at each rank according to the settling time requirement for minimum power.

2.2.1 Optimization of hierarchical sampling network

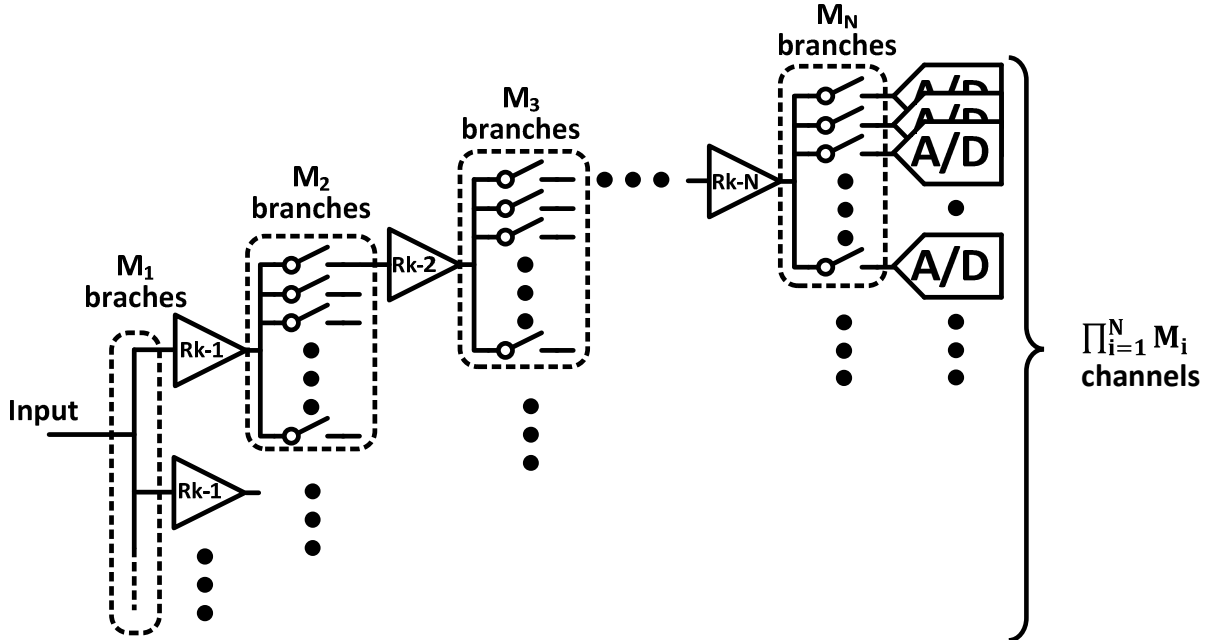


Figure 2.15 Block diagram of a general N-rank hierarchically time-interleaved ADC

In this section, we develop a general method to optimize the sizes of the T&H circuits and the sampling capacitors for a cascode sampler network with a fixed sampling hierarchy (i.e. fixed number of ranks and branching factors at each rank). Assuming the sampling network has N ranks and each sampler at $Rk-i$ fans out to m_i branches (Figure 2.15) and the non-overlapping clock scheme, the available settling time at each rank, T_i , can be calculated as a function of sampling rate of the overall ADC, f_s , and m_i 's:

$$T_i = \begin{cases} \frac{1}{m_1 m_2 f_s} & (i \leq 2) \\ \left(\prod_{k=2}^{i-1} m_k - \prod_{k=1}^{i-2} m_k \right) \frac{1}{2f_s} & (i > 2) \end{cases} \quad (2.12)$$

As shown in Section II, the dominant pole of the cascode sampler is its output pole, so if the DC gain is one and the settling error is set to be ϵ , the settling constraint at Rank- i is:

$$\frac{T_i}{\ln(\varepsilon)} = \frac{C_{total,i}}{g_{m,i}} \quad (2.13)$$

where $g_{m,i}$ and $C_{total,i}$ are the trans-conductance and total load capacitance of Rank- i , respectively. In addition to the settling time constraint, the front-end sampler also has to meet the overall ADC bandwidth requirement:

$$\frac{1}{2\pi f_B} = \frac{C_{total,1}}{g_{m,1}} \quad (2.14)$$

Where f_B is the ADC bandwidth. Combining (1), (2), and (3) results in:

$$\frac{C_{total,i}}{g_{m,i}} = \tau_i' \quad (2.15)$$

where:

$$\tau_i' = \begin{cases} \min\left(\frac{1}{\ln(\varepsilon)} \cdot \frac{1}{2m_1 m_2 f_s}, \frac{1}{2\pi f_B}\right) & (i = 1) \\ \frac{1}{\ln(\varepsilon)} \cdot \frac{1}{2m_1 m_2 f_s} & (i = 2) \\ \left(\prod_2^{i-1} m_k - \prod_1^{i-2} m_k\right) \cdot \frac{1}{\ln(\varepsilon)} \cdot \frac{1}{2f_s} & (i > 2) \end{cases} \quad (2.16)$$

The total input referred noise must be less than the ADC noise budget to avoid SNDR degradation. The sampled noise power at the output of Rank- i is:

$$\overline{V_{n,i}^2} = \frac{NF \cdot kT}{C_{total,i}} \quad (2.17)$$

Substituting (4) in to (6),

$$\overline{V_{n,i}^2} = \frac{NF \cdot kT}{\tau_i'} \cdot \frac{1}{g_{m,i}} \quad (1.18)$$

where NF is the effective noise factor of the cascode sampler. The total input referred noise of the sampler network can be written as a dot product, and the sampler noise constraint becomes:

$$\overline{V_{n,in}^2} = NF \cdot kT \left[\frac{1}{\tau_1'} \quad \dots \quad \frac{1}{\tau_N'} \right] \cdot \begin{bmatrix} \frac{1}{g_{m,1}} \\ \vdots \\ \frac{1}{g_{m,N}} \end{bmatrix} \leq N_B \quad (1.19)$$

where N_B is the budget for thermal and flicker noise. Since sampling capacitors are added to reduce the thermal noise, an additional constraint must be imposed on these capacitors. $C_{total,i}$ is

the sum of the output capacitance of Rank- i , the added sampling capacitance ($C_{L,i}$), and the input capacitance of Rank- $i+1$:

$$C_{total,i} = \frac{\gamma \cdot g_{m,i}}{f_T} + C_{L,i} + \frac{g_{m,i+1}}{f_T} \quad (1.20)$$

Where γ is the ratio between the output capacitance and the input capacitance³. Substituting (9) into (4) results in:

$$C_{L,i} = \left(\tau_i' - \frac{\gamma}{f_T} \right) g_{m,i} - \frac{1}{f_T} g_{m,i+1} \quad (1.21)$$

With the help of matrix formulation, the dependence of $C_{L,i}$ on $g_{m,i}$ for all the ranks in the sampler network described in (10) can be written in one equation:

$$\begin{bmatrix} C_{L,1} \\ C_{L,2} \\ \vdots \\ C_{L,N-1} \\ C_{L,N} \end{bmatrix} = \begin{bmatrix} \tau_1'' & -1/f_T & \dots & 0 \\ 0 & \tau_2'' & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \tau_{N-1}'' & -1/f_T \\ & & 0 & \tau_N'' \end{bmatrix} \cdot \begin{bmatrix} g_{m,1} \\ g_{m,2} \\ \vdots \\ g_{m,N-1} \\ g_{m,N} \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ C_{SAR} \end{bmatrix} \quad (1.22)$$

where $\tau_i'' = \tau_i' - \frac{\gamma}{f_T}$. The inequality in (10) describes the fact that the additional sampling capacitances must be greater than zero, except for the sampling capacitor of the last rank which needs to be larger than the input capacitance of the SAR ADC. Finally, the total power of the sampler network is proportional to the sum of the g_m 's of all the samplers in the network:

$$Power \propto \left[m_1 \quad \dots \quad \prod_{k=1}^{N-2} m_k \quad \prod_{k=1}^{N-1} m_k \right] \cdot \begin{bmatrix} g_{m,1} \\ \vdots \\ g_{m,N-1} \\ g_{m,N} \end{bmatrix} \quad (1.23)$$

With equations (8), (11), and (12), the overall optimization can now be formulated as:

$$\text{Minimize:} \quad [1 \quad \dots \quad \prod_{k=1}^{N-2} m_k \quad \prod_{k=1}^{N-1} m_k] \cdot \overline{g_m}$$

$$\text{Subject to:} \quad 1. \quad NF \cdot kT \left[\frac{1}{\tau_1'} \quad \dots \quad \frac{1}{\tau_N'} \right] \cdot \text{inv_pos}(\overline{g_m}) \leq N_B^4$$

$$2. \quad \begin{bmatrix} \tau_1'' & -1/f_T & \dots & 0 \\ 0 & \tau_1'' & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \tau_{N-1}'' & -1/f_T \\ & & 0 & \tau_N'' \end{bmatrix} \cdot \overline{g_m} \geq \begin{bmatrix} 0 \\ \vdots \\ C_{SAR} \end{bmatrix}$$

³ The C_g of a transistor can have a slight dependence on overdrive voltage, C_g is usually not directly related V_{ov} . Therefore, γ also have some dependence on V_{ov} . In this analysis, a constant V_{ov} is assumed.

⁴ $\text{inv_pos}(x)$ is defines as $1/x$, where $x > 0$.

where $\overline{g_m} = \begin{bmatrix} g_{m,1} \\ \vdots \\ g_{m,N} \end{bmatrix} > 0$ are the variables to be optimized. Since the cost function and the constraints are convex functions of $\overline{g_m}$, a convex optimization algorithm can be used [20].

When designing high speed hierarchically time-interleaved ADC with a power constraint, I recommend to start from sub-ADC design. The speed of SAR sub-ADC determines the minimum overall interleaving factor of the sampler. Once the input capacitances and maximum sampling rate of sub-ADC are obtained, different sampling hierarchies can be explored with the power optimization method developed in this section to find the configuration(s) that yields the lowest sampler network power consumption. Note that the power consumption of clock generation and distribution network is not taken into account in this optimization method. Therefore, the designer must make intelligent decision to pick the most feasible sampling network configuration with consideration of clocking power in mind. I recommend to go through the designs of clock distribution network for several feasible sampler hierarchies and find the solution with lowest overall power consumption.

2.3 Implementation and measurement results of the 7b 12.8GS/s ADC

To demonstrate the effectiveness of cascode T&H circuits and hierarchical time-interleaving, a 7b 12.8GS/s ADC is implemented in a 65nm CMOS process [10], [11] (Figure 2.16). The ADC consists of 3 ranks of sampler with a single front-end (Rk-1) T&H circuits to reduce ADC input capacitance and jitter-critical sampling clocks. A single Rk-2 samplers/demultiplexers demultiplexes the full-rate samples into 4 time-interleaved samples at 3.2GS/s. 4 Rk-3 samplers/demultiplexers further bring down the sample rate to 400MS/s – the rate of sub-ADCs. Finally, the 32 SAR sub-ADCs converts the analog samples into digital codes. Note that only 1 out of the 37 clocks (Φ_1) is jitter critical and requires power hungry drivers. Thus, small size CMOS gates are used to distribute the rest of the clocks. The phase-interpolators (PI) and frequency dividers (FD) are used to generate all the required sampling clocks. The details of their implementation will be discussed in chapter 4. The cascode T&H circuits with PMOS loads are used to implement the T&H circuits. Figure 2.17a shows ADC output amplitude versus the input frequency, and Figure 2.17b shows the measured SNDR. Despite the relatively low f_T of the process, the ADC achieved 25GHz effective resolution bandwidth (ERBW) thanks to the casode sampling circuits. To the best of the author's knowledge, it achieves the highest ERBW among published CMOS ADCs. Thanks to hierarchically time-interleaved architecture, the power optimized ADC achieves the effective number of bits (ENOB) of 4.2 at 25GHz while consuming only 168mW. The figure-of-merit (FOM) is 0.79pJ/conversion-step.

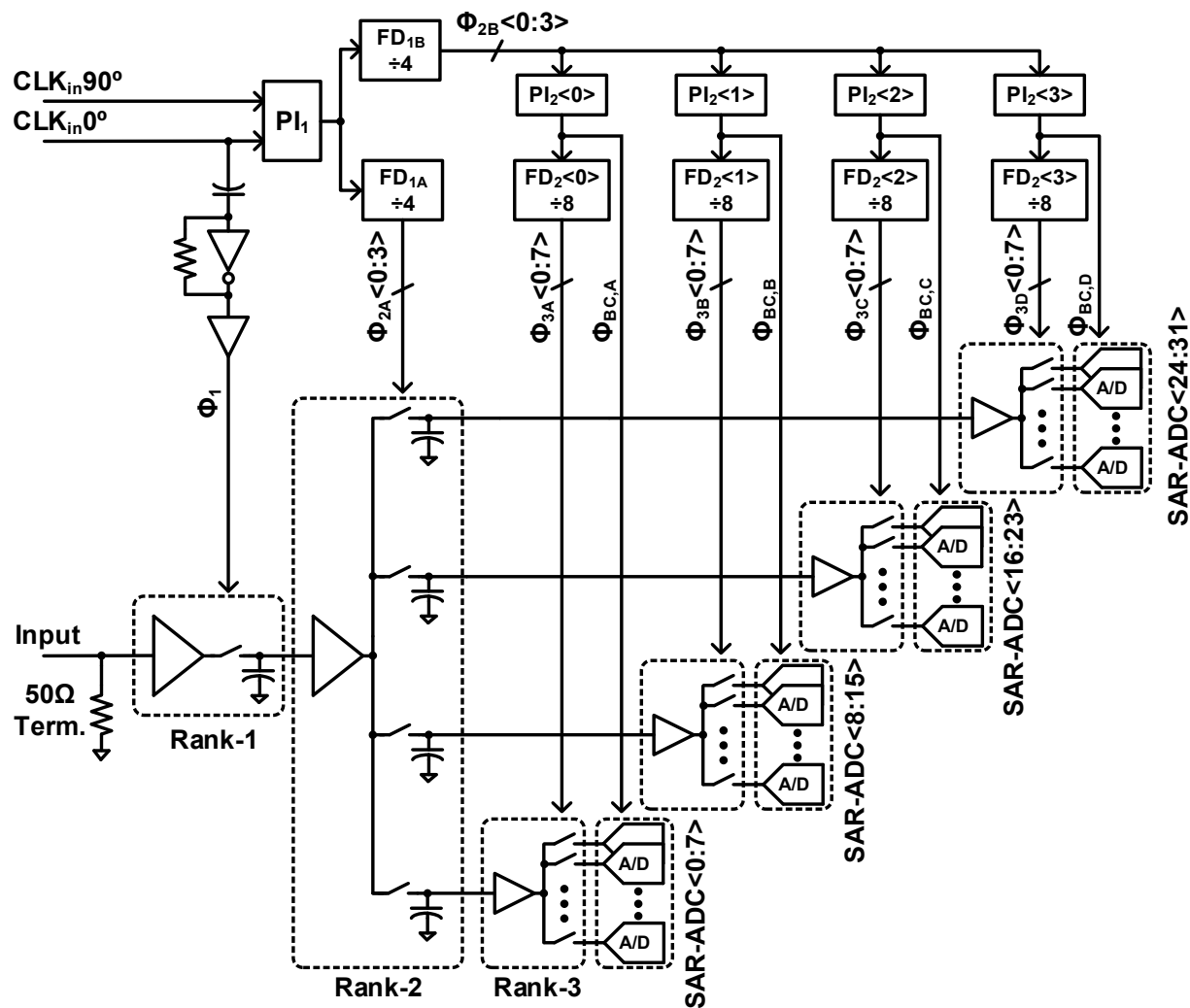


Figure 2.16 Block diagram of the 7b 12.8GS/s ADC in 65nm CMOS

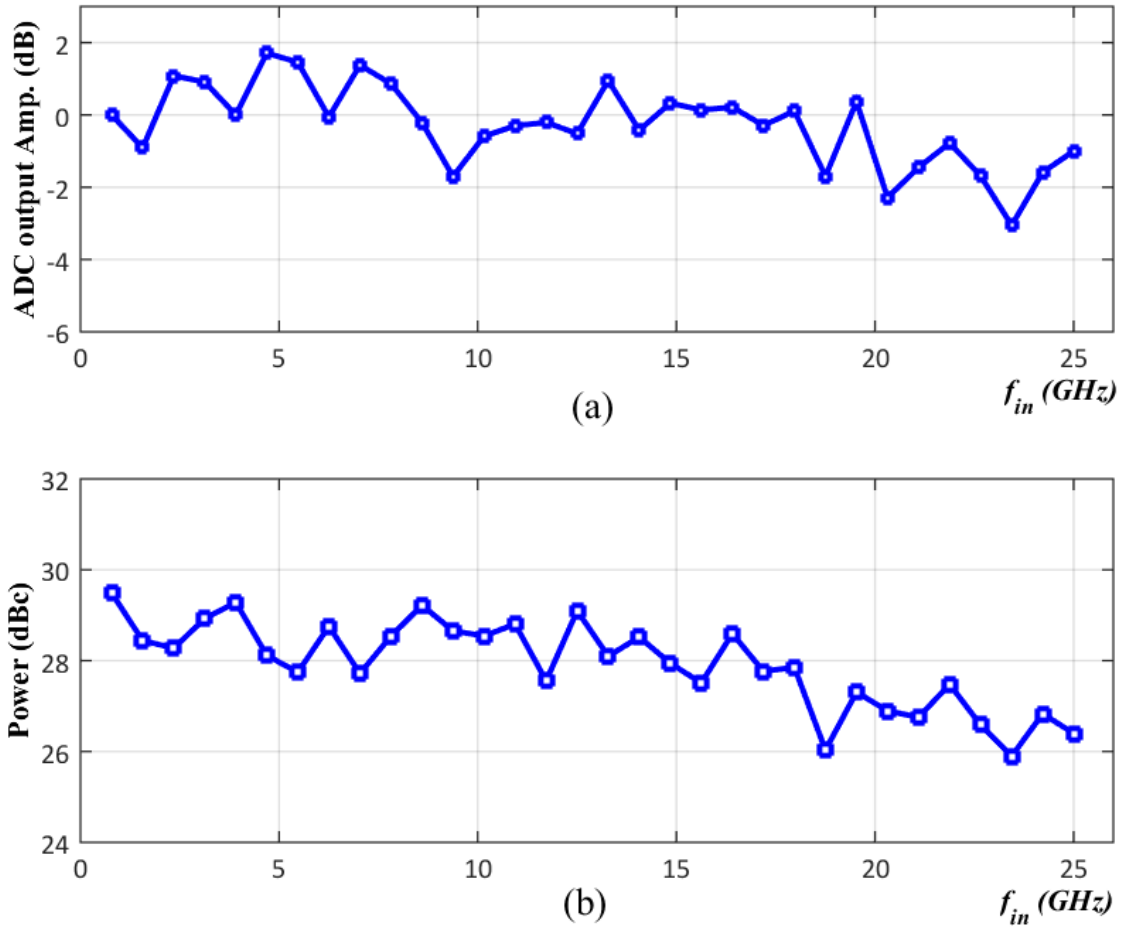


Figure 2.17 the input frequency vs. (a) normalized output amplitude, and (b) SNDR of the 7b 12.8GS/s ADC

Chapter 3

High-speed power-efficient sub-ADC

As mentioned earlier, the sub-ADC design is extremely important because it determines the overall interleaving factor of the sampler and have a large impact on overall ADC power. In ADC designs with a high degree of interleaving, the power consumption, the area, and the input capacitance of sub-ADCs must be carefully considered. The importance of sub-ADC power is perhaps self-evident, but sub-ADC area can be equally important since a large sub-ADC implies longer wiring to route the inputs and clocks. These long wires can lead to significant parasitic loading and hence substantially increased sampler/clock distribution power. Similarly, the input capacitance of sub-ADCs are the loads for last rank de-multiplexors. Thus, large sub-ADC input capacitance can raise the power consumption of the entire sampling network.

The common sub-ADC candidates are flash, pipeline, single/dual slope, and SAR ADCs. Although flash architecture has the potential to achieve high speed [21] [22] [5] [23], it requires a large number of comparators for medium resolution ADCs – 63 comparators for a 6-bit sub-ADC – resulting in large area and input capacitance. Conventional pipeline ADCs can have a much smaller input capacitance, but due to the numerous analog residue amplifiers needed in the design, both their area and power efficiency are poor [6]. Although novel techniques such as ring-amplifier based pipeline architectures [24] [25] [26] can achieve extremely high figure-of-merit (FOM) and small area, they suffer from limitations of large dead-zone induced noise and low sampling speed. As a result, pipeline architectures are not suited for sub-ADC designs in ultra-high-speed ADCs with medium resolution. Another architecture that has gained lots of attention in recent years is the single/dual slope ADC. Based on the linear search algorithm, it converts the signal voltage into a time delay using a voltage to time converter, and then counts the number of pulses from a reference clock cycles that passed within the delay. Although single/dual slope ADCs [27] can achieve very good power and area efficiency at low sampling rate, it quickly run out of steam for medium to high speed designs. For example, a 6-bit 1GS/s single/dual slope ADC requires a 64GHz reference clock, which is infeasible to implement with even 28nm CMOS logic gates. Unlike single/dual slope ADCs, SAR ADC is based on binary search algorithm, and thus is able to run much faster [28]. Although SAR ADC is usually slower than pipeline or flash, it does not have any analog components. Therefore, SAR can achieve high power efficiency. In addition, the input capacitance and area of a SAR are limited by the size of capacitive digital-to-analog converter (CDAC), and can be made extremely small thanks to modern CMOS process with fine pitch and tall metal stack [29]. Due to these reasons, SAR ADC is an ideal candidate for sub-ADC designs in a massively time-interleaved ADC system.

3.1 SAR sub-ADC

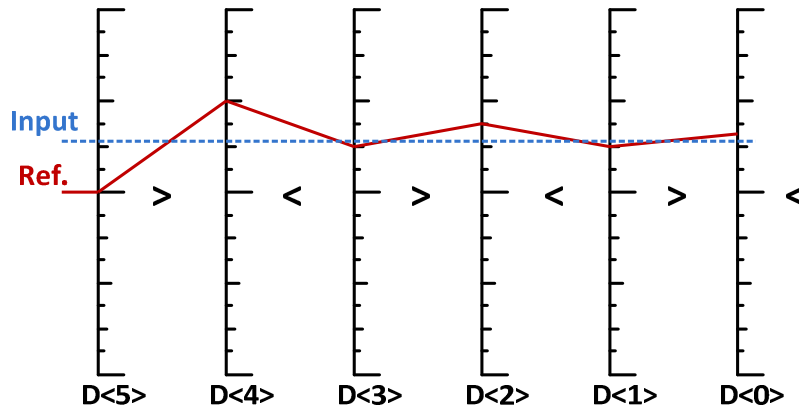


Figure 3.1 Illustration of the conversion algorithm for a 6-bit SAR ADC

As mentioned earlier, SAR ADCs use the binary search procedure. Take a 6-bit SAR ADC for example (Figure 3.1), the most-significant-bit (MSB) – D<5> in this case – is first obtained by comparing the input voltage to the half-scale voltage, $\frac{1}{2}V_{fs}$. Depending on the result of MSB, the reference level is moved up or down by a quarter of full-scale, $\pm\frac{1}{4}V_{fs}$. Then, the next bit, D<4>, is decided by comparing the input voltage to the new reference voltage. Based on the decision of D<4>, the reference level is moved by $\pm\frac{1}{8}V_{fs}$. This process is repeated for 4 more times with a smaller change in the reference level each time to complete conversion. Note that 6 comparison cycles – also known as bit-cycles – are needed for 6-bit SAR ADC, and after the initial setting to half-scale, the reference level is moved 5 times. The difference between input level and reference level is reduced by a factor of 2 after each cycle, and the residue difference at the end of SAR conversion represents the quantization error.

3.1.1 Synchronous and asynchronous SAR

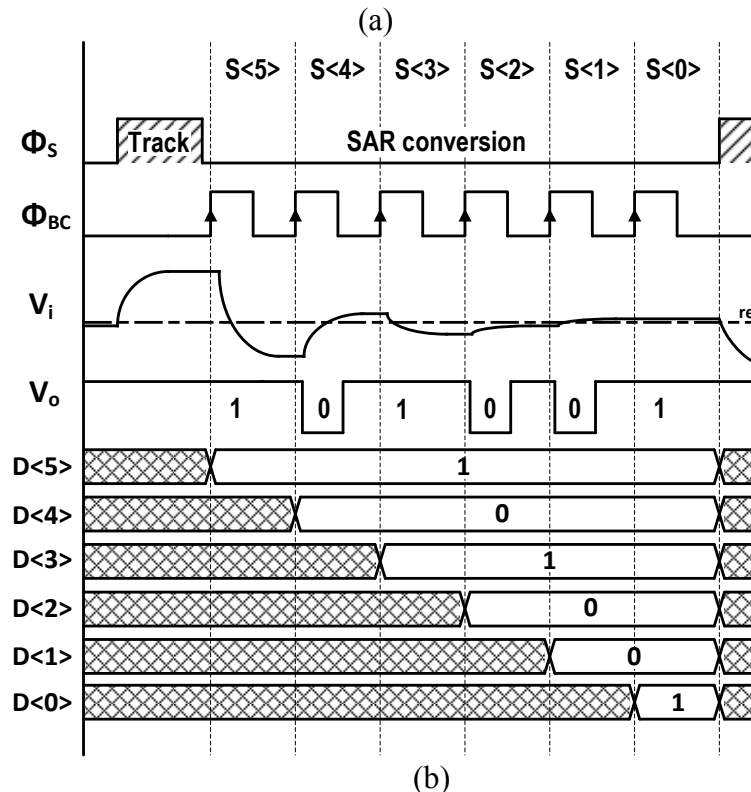
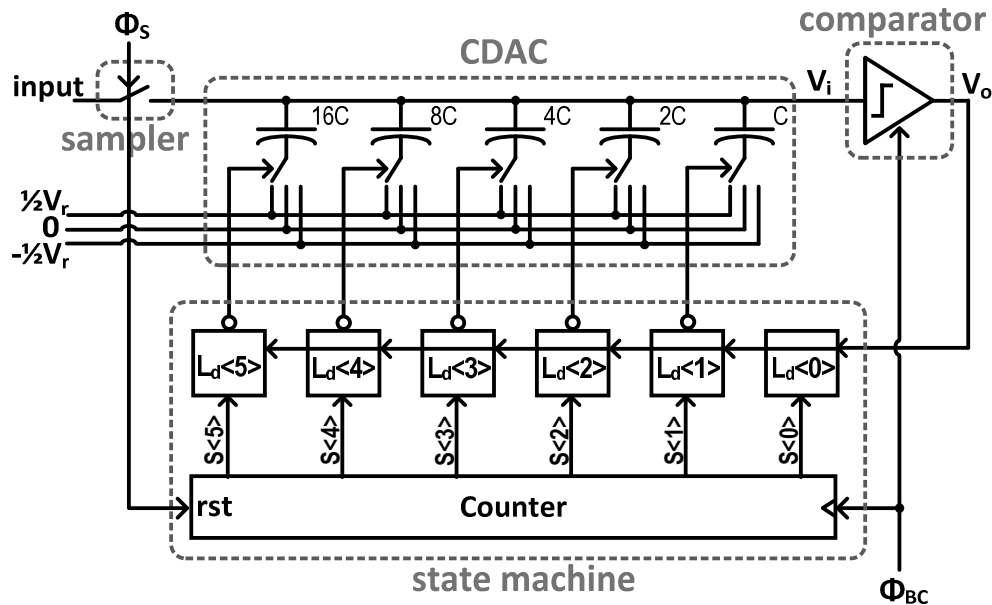


Figure 3.2 (a) schematic and (b) time diagram of a 6-bit synchronous SAR ADC

The implementation of a 6-bit synchronous SAR ADC (Figure 3.2a) consists of 4 basic building blocks, a sampler, a capacitive digital-to-analog converter (CDAC), a comparator, and a state machine. Since the sampler in this case is the cascode T&H circuit of the last rank which has been thoroughly studied in Chapter 2, it will not be discussed here. As mentioned earlier, metal-to-

metal fringe capacitors in advanced CMOS processes have high capacitance density and extremely well-controlled matching – usually lower than 2% variation for 1fF unit size capacitors. Thus, small size capacitor digital-to-analog converter (CDAC) is almost always used to save power. The SAR ADC in Figure 3.2a uses a 5-bit binary weighted CDAC to move reference levels across V_{fs} . Since the MSB decision is the sign bit, it can be decided without changing the state of the CDAC. Thus, 5-bit CDAC – instead of 6-bit CDAC – is sufficient for this 6-bit SAR ADC design. As shown in Figure 3.2b, The conversion time is evenly divided into 6 bit-cycles by a digital counter triggered by rising edge of a high frequency “bit-cycling” clock Φ_{BC} . The 6 bit-cycles are represented by the 6 states of the counter, $S\langle 5:0 \rangle$. Only 1 out of 6 $S\langle 5:0 \rangle$ can be high at a time. From the start of the SAR conversion, $S\langle 5:0 \rangle$ are pulsed sequentially from $S\langle 5 \rangle$ to $S\langle 0 \rangle$. The asserted counter output, $S\langle i \rangle$, represents the active bit-cycle at the time.

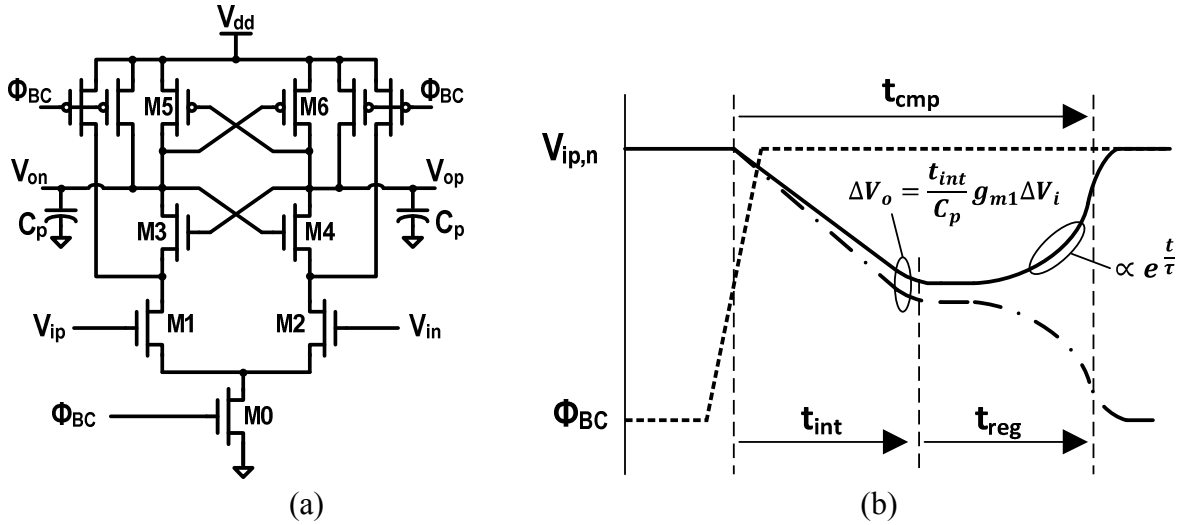


Figure 3.3 (a) Schematic of a typical Strongarm comparator, and (b) its output waveforms

At the beginning of a bit-cycle, the comparator is first triggered by the rising edge of Φ_{BC} . During the entire bit-cycle, the corresponding $S\langle i \rangle$ is also pulsed high to create a transparent window for the data latch, $L_d\langle i \rangle$, while the rest of latches remain opaque. After a certain amount of comparator delay (t_{cmp}) – which depends on the comparator input voltage – the decision bits are available at the input of $L_d\langle i \rangle$. After D-to-Q delay of $L_d\langle i \rangle$ ($t_{d \rightarrow q}$) and the propagation delay of a few additional logic gates (t_p), the switches that drive the corresponding capacitor cell in the CDAC are closed to move the reference voltage. Before the comparison of the next bit-cycle starts, the output voltage of the CDAC – equivalently, input voltage of the comparator – must fully settle. Any settling error in CDAC can cause a comparison errors and result in increased quantization error. Thus, the settling error must be controlled to be within a fraction of a least-significant-bit (LSB) of the ADC. For example, a 6-bit ADC requires the CDAC percentage settling error to be $< 1.5\%$, which requires minimum settling time of the CDAC (t_{CDAC}) to be $> 4.15\tau$, where τ is the time constant of the CDAC. To sum up, the duration of the bit-cycle must satisfy the following requirement to avoid SNDR degradation:

$$T_{BC} = \frac{T_{conv}}{N} > t_{cmp} + t_{d \rightarrow q} + t_p + t_{CDAC} \quad (3.1)$$

Where T_{conv} is the conversion time, and N is the number of bits. The right-hand-side of Equation 3.1 is sometimes referred as loop delay of SAR because it represents the time it takes for the information to travel from the input of the comparator through the state-machine and CDAC back to the input of the comparator. Clearly, Equation 3.1 sets the maximum speed of the SAR ADC. $t_{d \rightarrow q}$ and t_p are digital delay of logic gates, and are usually set by the technology and logic style of choice. t_{CDAC} depends on the size of CDAC. As mentioned earlier, since the CDAC is quite small for medium resolution SAR ADCs using advanced CMOS process, t_{CDAC} is usually not a significant factor. The remaining t_{cmp} – the comparator delay – is usually the dominant term in Equation 3.1. For a common comparator circuit such as Strongarm latch in Figure 3.3a, t_{cmp} consists of a linear integration time, t_{int} , followed by a latch regeneration time t_{reg} :

$$t_{cmp} = t_{int} + t_{reg} \quad (3.2)$$

During reset phase ($\phi_{BC} = 0$), the outputs of the comparator, $V_{op,on}$, is set to V_{dd} . At the beginning of the comparison, the differential pair, $M_{1,2}$, acts as input-controlled current-sources that discharges the output nodes. Both V_{op} and V_{on} linearly decreases as a result. At the same time, the difference between the 2 output voltages, ΔV_o , accumulates at the rate that's proportional to the differential input voltage, ΔV_i . It is worth mentioning that t_{int} cannot be set too short in practical designs as the input referred noise of the comparator is inversely proportional to t_{int} . For embedded ultra-high-speed ADCs, the swing at the input of SAR sub-ADCs is limited to few hundred millivolts by CLTE, VGA, and T&H circuits. Even for medium resolution ADCs, the LSB size can as small as a few millivolts, imposing stringent noise requirement on the comparator. For example, a 6-bit ADC with 400mV input swing requires the sub-ADC root-mean-square (RMS) noise to be $\ll 1.8\text{mV}$ to void SNDR degradation. This is the particular reason why t_{cmp} is often the dominant delay term in Equation 3.1. Once the V_{op} fall to the point that the latch devices M_{3-6} are turned on, the positive feedback loop of the cross-coupled devices (M_{3-6}) takes over the action to start regenerate ΔV_o . The regeneration delay (t_{reg}) can be approximated as:

$$t_{reg} = \tau \ln \left(\frac{V_{DD}}{A_{int} |V_i|} \right) \quad (3.3)$$

Where V_{DD} is the supply voltage, $A_{int} = \frac{t_{int}}{C_p} g_m$ is the integration gain, and τ is the regeneration time constant of the cross couple devices M_{3-6} – usually inversely related to f_T of a given process. Note t_{reg} increases with decreasing input voltage of the comparator, ΔV_i , which can be different from cycle to cycle (see Figure 3.2b). Since duration of the bit-cycles is fixed by the period of Φ_{BC} , T_{BC} must be kept greater than the loop delay for the worst-case comparator input voltage to avoid SNR degradation, which is usually smaller than 1LSB. Assuming the sensitivity (minimum resolvable voltage within its allowed delay) of the comparator is below 0.5LSB, we can combine 3.1, 3.2, and 3.3 to write the conversion time as for a synchronous SAR ADC as:

$$\begin{aligned} T_{conv} &= (N - 1)(t_{d \rightarrow q} + t_p + t_{CDAC}) + N t_{int} + N \tau \cdot \max \left[\ln \left(\frac{V_{DD}}{A_{int} |V_i|} \right) \right] \\ &= (N - 1)(t_{d \rightarrow q} + t_p + t_{CDAC}) + N t_{int} + N(N + 1) \tau \cdot \ln \left(\frac{2V_{DD}}{A_{int} V_{sw}} \right) \end{aligned} \quad (3.4)$$

Where N is the number of bits, and V_{sw} is the input swing of the ADC. This is an inefficient allocation of the conversion time, because the worst-case loop delay scenario can occur in only 2 bit-cycles at most. The extra time to allow comparator to resolve in the rest of bit-cycles is wasted.

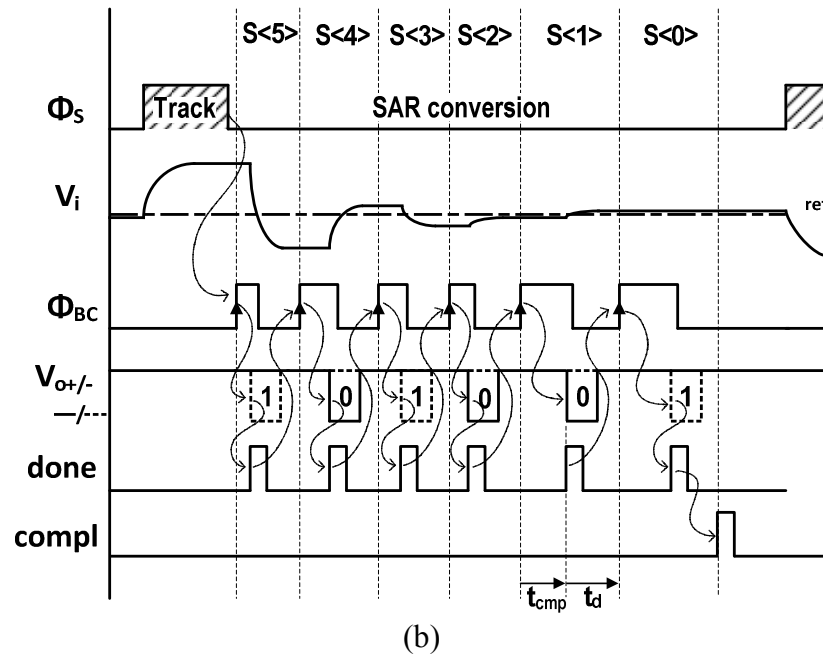
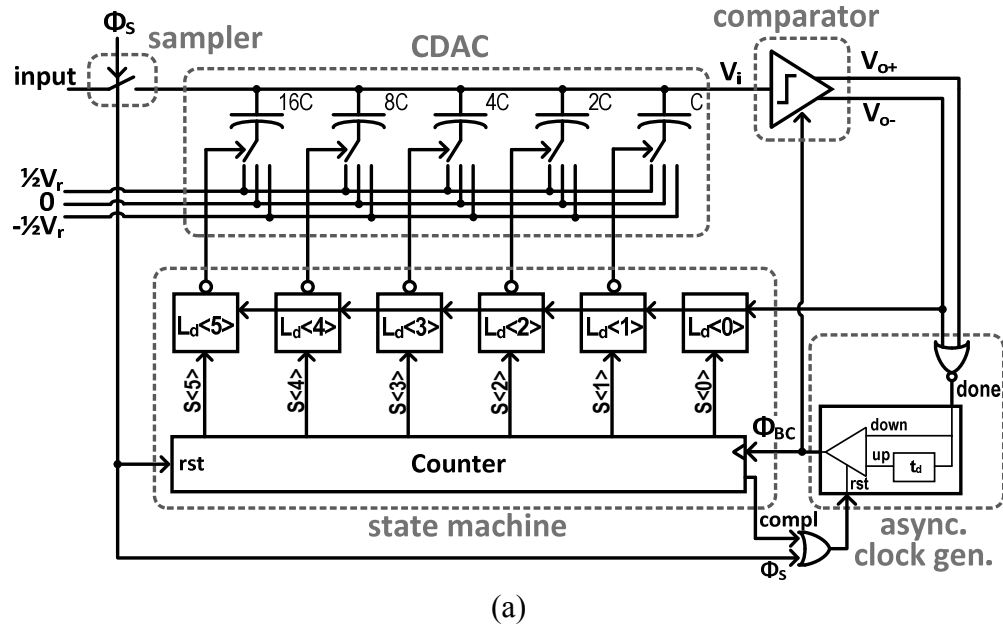


Figure 3.4 (a) Schematic and (b) timing diagram of classic asynchronous SAR ADC

An elegant way to efficiently utilize conversion time and improve the speed of SAR ADC is to use asynchronous design [30]. In a classic asynchronous SAR ADC shown in Figure 3.4a, the bit-cycle duration is not fixed by a high frequency clock (Φ_{BC}), but instead, varies depending on the input voltage of the comparator (V_i) for the current bit-cycle. If the comparison finishes early, the asynchronous clock generator will move on to the next cycle as soon as the CDAC settles

instead of waiting for an external clock edge (Figure 3.4b). This is accomplished by detecting when the comparator has made a decision. After the rising edge of Φ_{BC} , both V_{o+} and V_{o-} of the Strongarm comparator are observed. If either of the outputs falls, a done signal is raised to signal the comparison is finished. This in turn resets Φ_{BC} . After a fixed delay (t_d) to allow bit propagation and CDAC settling, Φ_{BC} is raised again for the next bit-cycle. Note that the self-generated Φ_{BC} has variable period and duty-cycle. The bit-cycle duration for small V_i can be considerably longer than the bit-cycle for large V_i . This asynchronous timing scheme can be faster than synchronous approach because the extra comparison time for small V_i is effectively “borrowed” from the bit-cycles with large V_i . In fact, it can be shown that asynchronous SAR ADC can save half of total comparison time comparing to the synchronous counterpart [30]. After all 6 bit-cycles, the counter raises *compl* to signal completion of the conversion and to keep Φ_{BC} low until the next sample conversion.

Besides its speed advantage, an additional benefit of asynchronous SAR ADCs is that they do not require the external high-frequency Φ_{BC} , and thus saves the power needed to generate and distribute Φ_{BC} . This is extremely important for massively time-interleaved ADCs. As mentioned in Chapter 2, such ADCs can be quite large, so distributing a high frequency clock across the long distance can be power hungry. Furthermore, since each sub-ADC have a slight phase offset, a number of delay-lock-loops (DLL) might be required to generate the synchronous bit-cycling clocks for each sub-ADCs. This further increases the total ADC power and design complexity.

3.1.2 Comparator meta-stability and sparkle-code

One drawback of asynchronous SAR ADC is the well-known comparator meta-stability issue.⁵ As mentioned earlier, meta-stability describes the event that the input to the comparator is so small that the comparison time is extremely long. Since the self-generated Φ_{BC} in asynchronous SAR ADCs always waits for the comparison to be done before moving to the next bit-cycle, a comparator meta-stability event can cause Φ_{BC} to wait forever. As a result, the SAR may not complete all the bit-cycles within the conversion time. This can sometime causes sparkle-codes – rarely occurring large errors that do not follow Gaussian distribution profile. Figure 3.5 illustrates such a scenario. In this example, the MSB decision enters meta-stable state: The input voltage is so small that the bit-cycle lasts extremely long time. After the MSB decision ($S\langle 5 \rangle$) resolves, there is enough time left for only 2 more bit-cycles, $S\langle 4 \rangle$, $S\langle 3 \rangle$. At the end of conversion, the 3 MSB latches, $L_d\langle 5:3 \rangle$, contain the correct decision bits, and the 3 LSB latches, $L_d\langle 2:0 \rangle$, are still in the initial states. The latches of the LSB segment that left in the initial states can be interpreted as having a value of $b'100=d'4$.⁶ The conversion result is binary number of $b'100100=d'36$. Comparing to the error-free 6-bit ADC output with a mid-rail input if $d'32$, the error amplitude is as large as 4-LSB. Another way to intuitively understand meta-stability caused error is to examine residue voltage at the end of conversion. The large V_{res} left at the end of conversion in Figure 3.5

⁵ Meta-stability issue is not unique with asynchronous SAR ADCs. Most types of ADC suffer from meta-stability. This thesis focuses on the meta-stability issue of asynchronous SAR ADCs.

⁶ Note that if $L_d\langle 2:0 \rangle$ is implemented as digital latch and is reset low, the their initial value should be $b'000$, instead of $b'100$. However, each L_d shown in Figure 3.4a actually have 3-states to controls 3 switches, implemented with 2 digital latches that stores both V_{o+} and V_{o-} of the comparator, L_{dp} and L_{dn} (chapter 4). The initial values of both L_{dp} and L_{dn} is used to interpret the CDAC codes.

represents a large conversion error, whereas the residue voltage in ideal SAR ADC is always less than 1LSB.

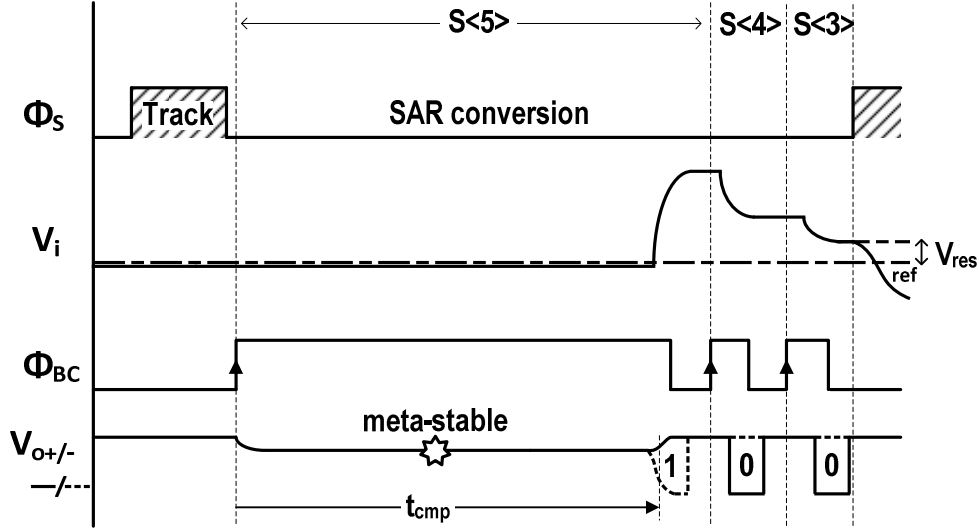


Figure 3.5 SAR ADC waveforms when the meta-stability event in MSB results in a sparkle-code

Since the meta-stability induced sparkle-code is not caused by stationary additive Gaussian noise, it does not follow a Gaussian distribution profile. As mentioned earlier, some applications require extremely low sparkle-code error-rate, therefore, the asynchronous SAR ADC designer must understand the trade-offs to reduce sparkle-code probability. Since the mechanism to produce the sparkle-code of a particular amplitude is a complex nonlinear function, it is extremely difficult to calculate the exact distribution profile for the sparkle-code errors. Fortunately, we know that all sparkle-codes in asynchronous SAR ADCs are caused by insufficient conversion time, so we can estimate the total sparkle-code error-rate by calculating the probability that the SAR cannot complete conversion in a given time. To calculate this probability, we begin from calculating total time required for this 6-bit SAR ADC to complete the conversion. Combining Equation 3.1, 3.2 and 3.3, we can write down the duration of n^{th} bit-cycle as:

$$T_{BC}(n) = t_{d \rightarrow q} + t_p + t_{CDAC} + t_{int} + \tau \ln \left(\frac{V_{DD}}{A_{int} |V_i(n)|} \right) \quad (3.5)$$

Where the D-to-Q delay of the latch, $t_{d \rightarrow q}$, delay of digital gates, t_p , settling time of the CDAC, t_{CDAC} , and the integration time of the comparator, t_{int} , are fixed by design and usually do not depend on input voltage of the comparator. Therefore, the total time required to complete all bit-cycles for an N-bit asynchronous SAR ADC is:

$$T_{conv} = 5(t_{d \rightarrow q} + t_p + t_{CDAC}) + 6t_{int} + T_{reg}(V_{i,ADC}) \quad (3.6)$$

where $V_{i,ADC}$ is ADC input voltage, and $T_{reg}(V_{i,ADC})$ is the total regeneration time:

$$T_{reg}(V_{i,ADC}) = \tau \sum_{n=0}^6 \ln \left(\frac{V_{DD}}{A_{int}|V_i(n)|} \right) \quad (3.7)$$

Where τ is the regeneration time constant. The comparator input voltage in Equation 3.7, $V_i(n)$, can be calculated using the following recursive equation:

$$\begin{cases} V_i(N) = V_{i,adc} & (n = 6) \\ V_i(n) = V_i(n + 1) - \frac{V_r}{2^{N-n-1}} \cdot \text{sign}(V_i(n + 1)) & (n < 6) \end{cases} \quad (3.8)$$

Where V_r is the CDAC reference voltage and equals the ADC full swing. Using Equation 3.6, 3.7 and 3.8 and with the help of Matlab, we can calculate the required total regeneration time, T_{reg} , as a function of ADC input. Note T_{reg} is the direct representation of the speed of the SAR, because the total conversion time of the SAR (T_{conv}) can be easily obtained from T_{reg} by adding a constant term, $5(t_{d \rightarrow q} + t_p + t_{CDAC}) + 6t_{int}$ in this case. Figure 3.6 shows the input voltage vs. the normalized total regeneration time, $\frac{T_{reg}}{\tau}$, for a 6-bit asynchronous SAR ADC with 0.4V swing and 1.05V supply (V_{dd}). The curve has numerous local minima and singularities. In fact, the total regeneration time is infinite at all 63 CDAC thresholds, and the local minima occur around half way between any 2 adjacent thresholds. This is because the comparator delay approaches to infinity logarithmically as its input goes to 0 (Equation 3.7), and if the ADC input equals any SAR thresholds, there is always a bit-cycle when the comparator input is exactly zero. If we magically make sparkle-code disappear, $T_{reg,max}$ can be set to the worst-case local minima – at around $V_{in} = 44.5mV$ – without penalizing the signal-to-noise ratio (SNR). However, this result in horrendous sparkle-code errors in practice, because the most part of the curve is above the maximum allowable regeneration time ($T_{reg,max}$). One way to reduce the sparkle-code error-rate is to slow down the speed of the SAR ADC to allow more regeneration time. As shown in Figure 2b, if $T_{reg,max}$ is increased from the worst-case local minima by $\Delta T_{reg,max}$, the segment of the curve falls above $T_{reg,max}$ is reduced in width – from ΔV_{in1} to ΔV_{in2} . Since the probability for uniformly distributed V_{in} to fall within ΔV_{in} is directly proportional to ΔV_{in} , the chances of the SAR running out of conversion time is thus reduced. Let's define this additional conversion time to reduce sparkle-code error-rate ΔT_{add} . To more accurately estimate the sparkle-code error-rate for a given $T_{reg,max}$, we need to calculate sum of the ΔV_{in} 's around all the 63 thresholds and sum them up. Figure 3.7 shows the calculated sparkle-code error-rate as a function of normalized additional conversion time (ΔT_{add}). Not surprisingly, the sparkle-code error rate falls exponentially as the allowed regeneration time increases – equivalently, slower speed. This exponential trade-off between speed and sparkle-code error-rate is not efficient enough for the systems with extremely stringent error-rate requirement. For instance, to achieve sparkle-code error-rate of $1e-15$ – required in most wireline/optical communication systems, we need to budget $> 25\tau$ of additional time for the SAR ADC, which causes a significant speed penalty. Unfortunately, the meta-stability is a fundamental issue associated with the positive-feedback circuit used in the comparators. Therefore, meta-stability caused sparkle-codes cannot be completely eliminated. Despite of its persistence, it is possible to reduce the sparkle-code error-rate using clever techniques. The next section will focus on these techniques to “correct” the sparkle-code errors.

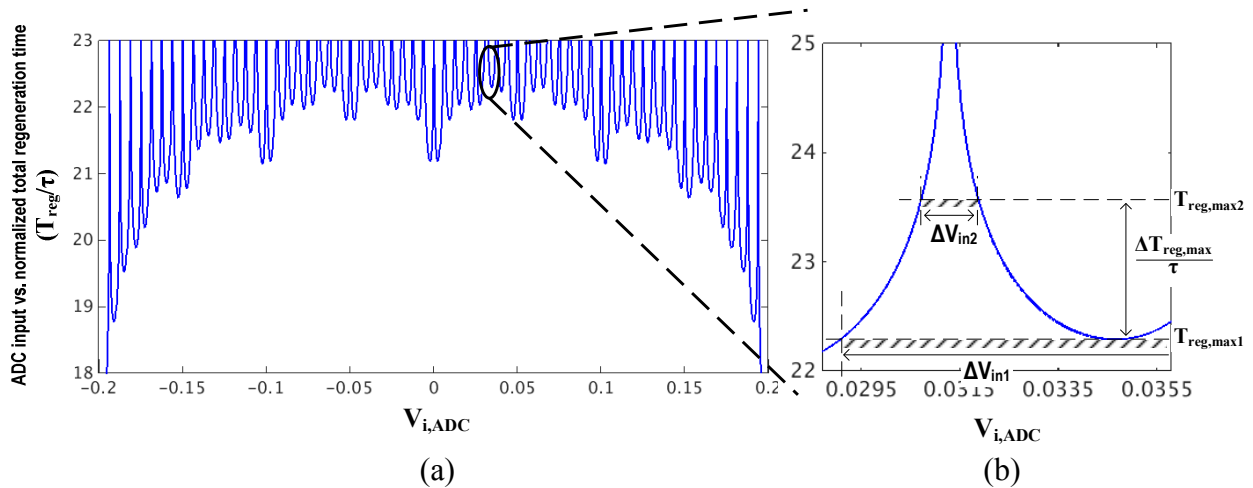


Figure 3.6 ADC input vs. normalized total regeneration time for a 6-bit asynchronous SAR ADC with 400mV input swing and 1V supply.

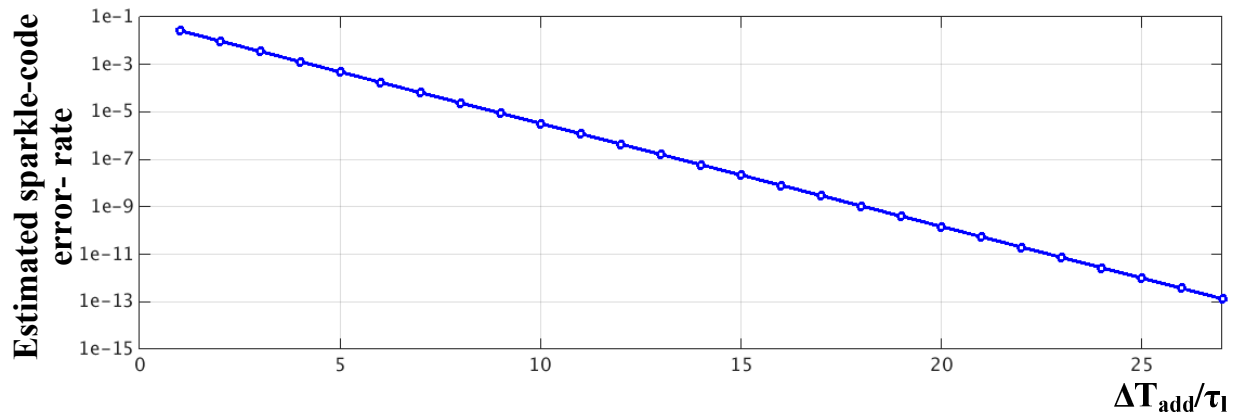


Figure 3.7 Estimated sparkle-code error-rate vs. normalized additional conversion time for a 6-bit asynchronous SAR ADC with 400mV input swing and 1V supply

3.2 Back-end meta-stability correction

As mentioned earlier, the sparkle-code error-rate in classical asynchronous SAR ADC falls exponentially with more additional conversion time, and may incur significant speed penalty in low error-rate systems. Several recent works [31] [32] attempted to solve this problem. The most popular approach in these works is to detect the meta-stability events, and then immediately stop the conversion if an event is detected – detect-then-stop approach. Unfortunately, the effectiveness of this approach to significantly reduce the sparkle-code error-rate has not been demonstrated, either in theory or experiment. In this section we will investigate why the detect-then-step method might not be able to significantly reduce sparkle-code error-rate (Section 3.2.1). Then, after the introduction of the new back-end meta-stability correction method in Section 3.2.2, an extensive statistical analysis is presented in Section 3.2.3 to demonstrate effectiveness of the proposed correction technique.

3.2.1 Meta-stability correction by detect-then-stop method

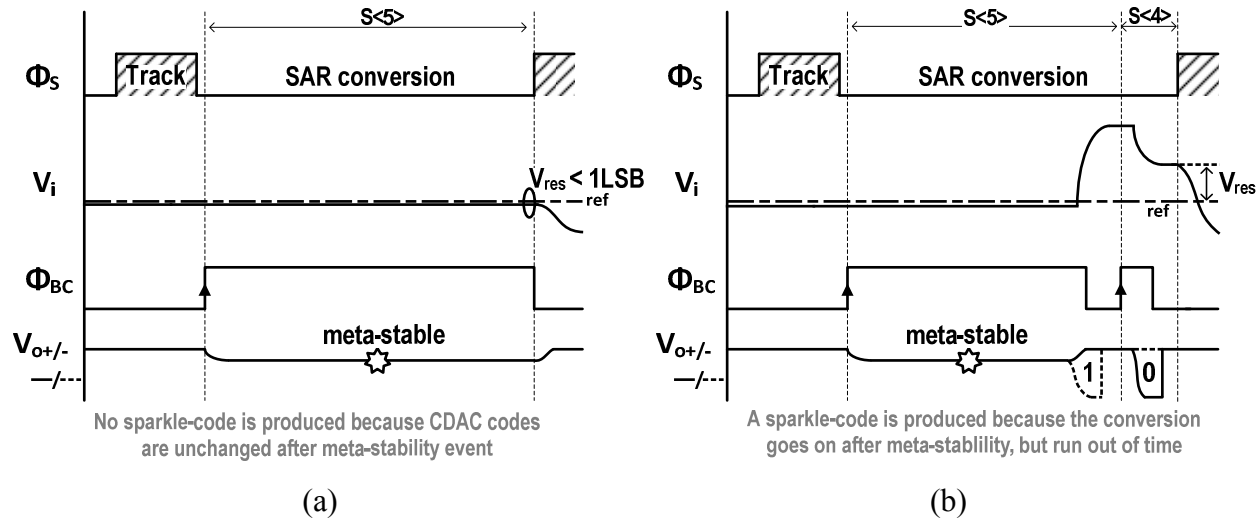


Figure 3.8 Waveform of a classic asynchronous SAR ADC in the scenario that (a) a meta-stability event does not cause a sparkle-code and (b) a meta-stability event caused a sparkle-code

Before going into the details of the detect-then-stop correction method, it is worthwhile to re-examine more closely the meta-stability event in asynchronous SAR ADCs and understand how exactly a sparkle-code is produced by such an event. Consider the scenario illustrated in Figure 3.8a, a meta-stability event occurs at MSB, resulting in such a long comparator delay that even the MSB itself does not get resolved. Interestingly, this scenario does not produce a sparkle-code. Since the residue voltage (V_{res}) left on the CDAC is approximately 0 at the end of the conversion, the initial CDAC codes can be interpreted to give a good approximation of the ADC input, and subsequent comparison is no longer necessary. This example implies that the CDAC codes during any comparator meta-stable event can actually produce the correct ADC output. Another way to understand it is to realize CDAC voltage during the meta-stability event is always extremely small. If this is the case, how does a meta-stable event produce a sparkle-code? In order to get a sparkle-code, we need to consider the scenario in Figure 3.8b: although a meta-stable event occurs in MSB, it is still resolved before the end conversion. There are even enough time left for 1 more bit-cycle ($S<4>$). As a result, a large unconverted residue voltage is left at the end of conversion, and a sparkle-code is produced. From this example, we can conclude that a sparkle-code is produced if the comparator delay cause by meta-stability is long enough so that not all subsequent bit-cycles can finish, but it mustn't be too long to stall the current and subsequent bit-cycles.

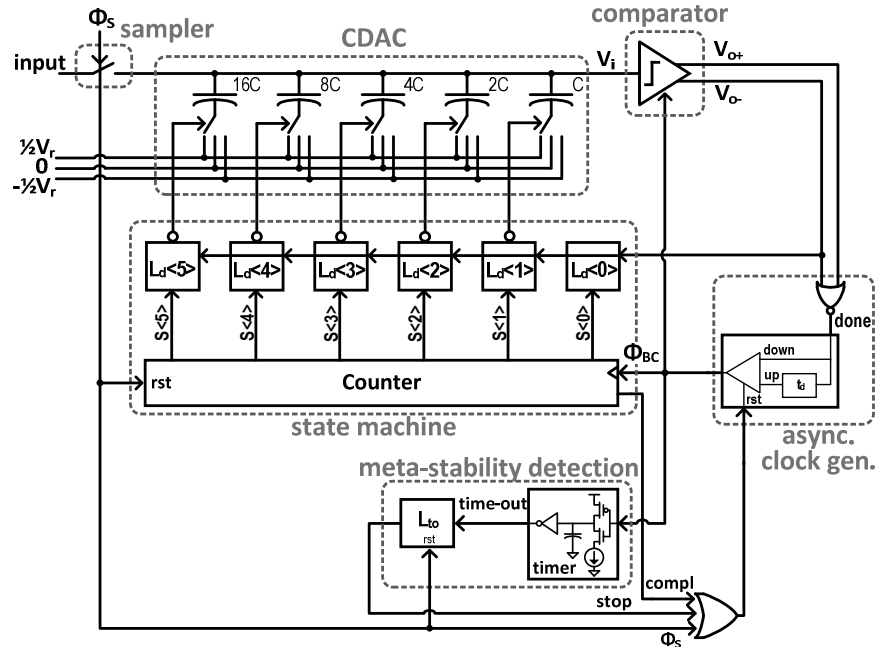


Figure 3.9 Schematic of asynchronous SAR ADC with detect-then-stop sparkle-code correction method

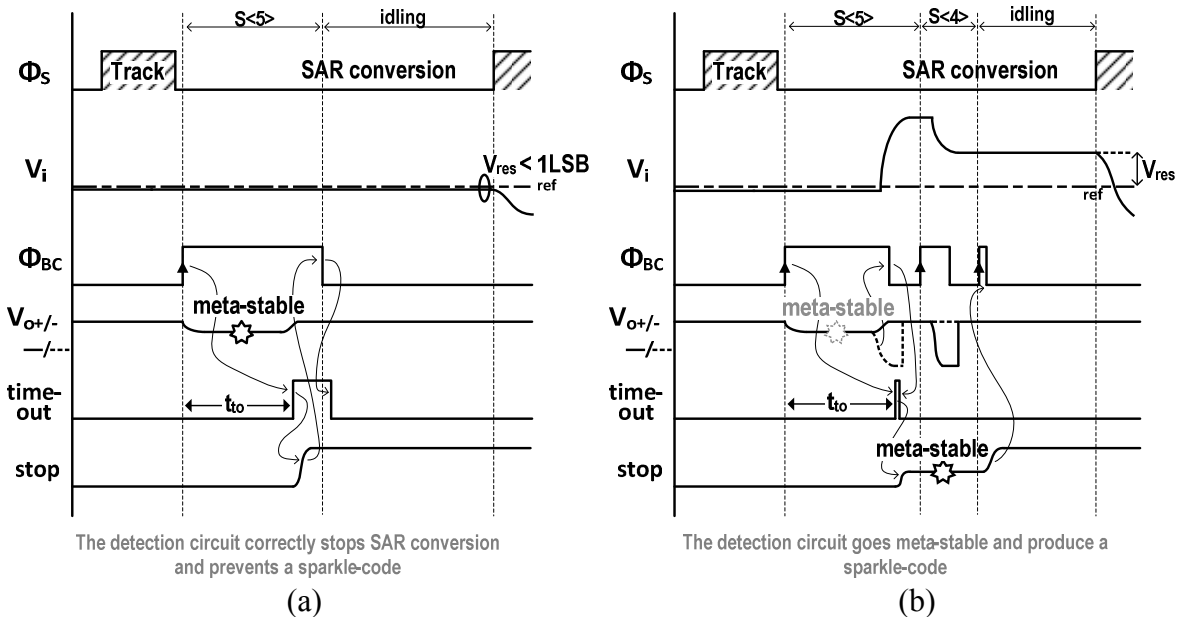


Figure 3.10 Waveform of detect-then-stop meta-stability correction circuit in the scenario of (a) the meta-stability is correctly detected, and (b) the detection circuit goes meta-stable

Based on the previous observation, one natural way to prevent the meta-stability event from corrupting the already perfect CDAC codes – to avoid the scenario in Figure 3.10b – is to force stop the SAR conversion in the middle of a meta-stability event. This is exactly the detect-then-stop meta-stability correction method. Its circuit implementation is shown in Figure 3.9. To detect the comparator meta-stability, a detection path that consists of timer and a pulse latch called “time-out latch” (L_{to}) is added. As shown in Figure 3.10a, the timer pulls its output (time-out) high a

fixed amount of time (t_{to}) after the rising edge of Φ_{BC} , and resets time-out immediately after the falling edge of Φ_{BC} . t_{to} is ideally set to equal the maximum allowable comparator delay by a current limited inverter. As a result, a high pulse would appear in the time-out node if a meta-stability event occurs ($t_{cmp} > t_{to}$), and the width of the pulse is roughly equal to $t_{cmp} - t_{to}$. Otherwise the time-out node remains low. L_{to} is a one-shot circuit that latches to V_{dd} upon “seeing” a time-out pulse. The state of L_{to} is used to stop the SAR conversion of the current and subsequent bit-cycles. This approach works fine with an ideal meta-stability detection circuit. However, it falls apart in practical design, because the pulse latch L_{to} is a decision circuit implemented with positive-feedback, the thus susceptible to meta-stability errors. Consider the scenario illustrated in Figure 3.10b, the comparator is at the verge of being meta-stable – for example, t_{cmp} is just slightly greater than t_{to} . The resulting narrow pulse at time-out might trigger a meta-stability event in L_{to} , causing it to latch to V_{dd} at a much later time. This in turn stops the SAR conversion after the comparator meta-stability has passed, resulting in a sparkle-code similar to classic asynchronous SAR ADCs (Figure 3.10b). Therefore, the detect-then-stop approach does not fundamentally correct meta-stability caused errors; it only moves the meta-stability problem from the comparator circuit to the meta-stability detection circuit.

Upon closer look, we can find the reason that the detector meta-stability cause sparkle-code is because the detector decision is used immediately after a detection. When a meta-stable event occurs in the comparator, the detector must resolve its decision to stop the asynchronous clock generator in extremely short amount of time – before the comparator resolves its own meta-stability. In order to prevent the meta-stability from happening in the detection circuit, the detection time must be relaxed. Thus, the high-speed feed-back path from the output of the detector to reset the clock generator in Figure 3.9 must be avoided. In the next section, we will propose a correction method that fix this detector meta-stability issue by using the detector detection at a much later.

3.2.2 Back-end meta-stability correction

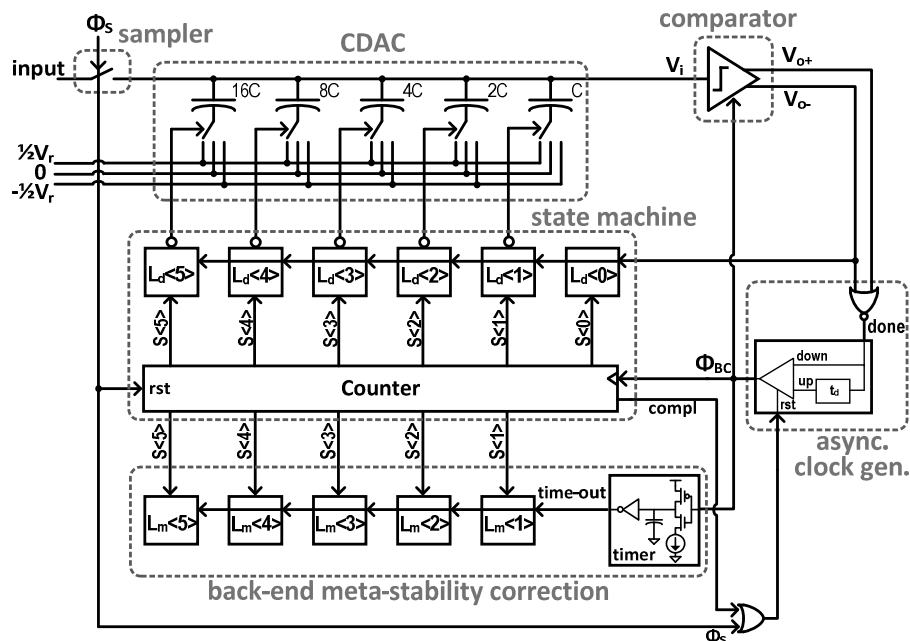


Figure 3.11 Schematic asynchronous SAR ADC with back-end meta-stability correction circuit

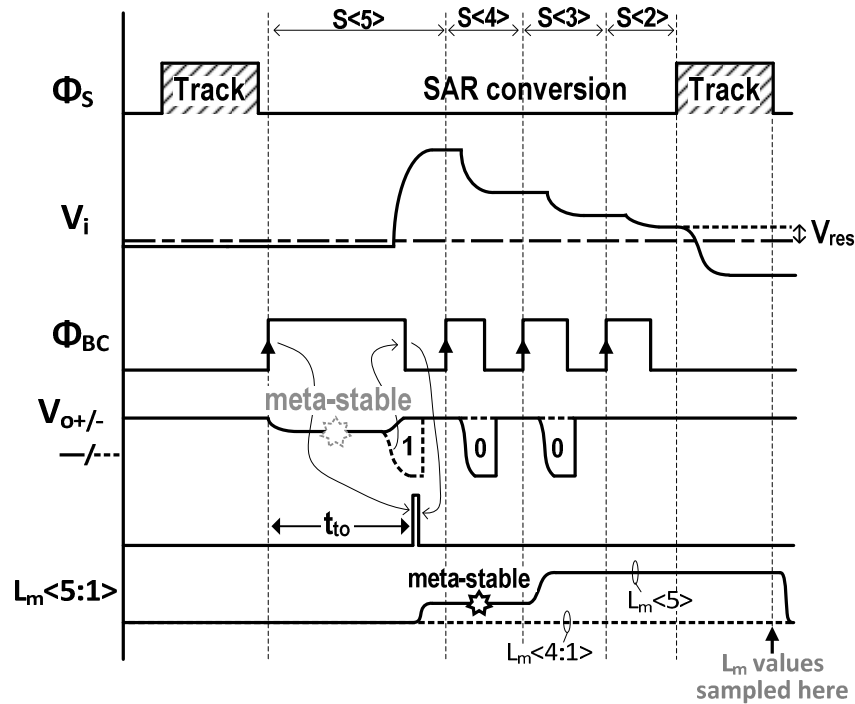


Figure 3.12 Waveform of the back-end meta-stability correction circuit when the first correction bit ($L_m\langle 5 \rangle$) goes meta-stable

As highlighted earlier, the key issue that results in the failure of the detect-then-stop approach is its high-speed feedback path to stop the clock generator. In order to properly deal with comparator meta-stability, a back-end meta-stability correction method is proposed to break this feedback path and store detection results for later correction in the digital back-end (Figure 3.12). The proposed circuit uses the same timer to generate the time-out pulse(s) to signal the occurrence(s) of comparator meta-stability event(s). Instead of immediately stopping the SAR conversion, the information carried by the pulse(s) are stored in 5 meta-stability latches ($L_m\langle 5:1 \rangle$) that corresponds to 5 MSB states. Each L_m is controlled by a state signal S exactly the same as the data latches (L_d); they flag the corresponding bit-cycle in which the meta-stable event has occurred. Since the information stored in $L_m\langle 5:1 \rangle$ are not used until the start of the conversion of the next sample – more than a T_{track} later, there are in practice sufficient time for any meta-stable events to resolve. For example, assuming the comparator MSB decision is marginally meta-stable – the comparator delay is slightly greater than t_{to} (Figure 3.12). The narrow pulse generated by the timer causes $L_m\langle 5 \rangle$ to enter a meta-stable state. However, $L_m\langle 5 \rangle$ has long been resolved at the time its value is taken (indicated by the black arrow in Figure 3.12). Thus, the comparator meta-stable event is successfully detected. What happened after MSB does not affect the corrected ADC output, because $L_m\langle 5 \rangle = 1$ tells the digital backend to ignore values in $L_d\langle 5:1 \rangle$. In fact, the probability that any L_m is still meta-stable by the time it arrives at digital back-end correction logic is negligibly small – even for systems requiring $< 1e-15$ error-rates, because the meta-stability flag bits (L_m) along with data bits (L_d) are usually re-sampled by a cascade of flip-flop stages before feeding into DSP or sending off chip.

To correct for meta-stability, we only need to reconstruct the CDAC code prior to the meta-stable event. With the knowledge of which bit-cycle is meta-stable, it can be done simply by

replacing the post meta-stable bit CDAC segment by a binary string of 100... In this 6-bit example, the corrected bits can be formulated as:

$$D_c = (L_d \& A_m) || B_m \quad (3.9)$$

Where $\&$ is the bit-wise and operator, and $||$ is the bit-wise or. The 6 bit wide meta-stability correction terms A_m and B_m can be related to L_m as:

$$A_{m,i} = \begin{cases} \overline{L_{m,i+1}} || L_{m,i} & (0 \leq i < 5) \\ \overline{L_{m,5}} & (i = 6) \end{cases} \quad (3.10)$$

and

$$B_m = A_m \gg 1 + 1'b1 \quad (3.11)$$

Where $||$ represents or operator; \gg is right-hand bit-wise shift operator with zero-padding, and $1'b1$ is the 1 bit wide binary 1. The first correction term A_m zeros out the LSB segment of CDAC codes post the meta-stability event, and the second term B_m is the b'100... replacement string to be added.

All the analysis thus far shows the meta-stability event in the back-end correction circuit can be practically avoided because there are always sufficient time for L_m 's to resolve. However, the back-end meta-stability correction method cannot completely eliminate sparkle-codes, because the detector can make still errors. In fact, the inherent random jitter produced the timer circuit as well as the meta-stability latches can randomly modulate the time-out pulse width, causing both false-positive and false negative detection errors. In the case of a false-positive detection error (Figure 3.13a), the comparator does not enter meta-stable state in the i^{th} bit-cycle ($S<i>$). However, the instantaneous jitter in the timer circuit causes the rising edge of the time-out to arrive too early – earlier than the falling edge of Φ_{BC} . As a result, an erroneous time-out pulse appears and causes a flag bit of 1 to be registered in the $L_m<i>$. In this case, a fake meta-stability event is invented, causing an error in the corrected ADC output. Fortunately, as will be investigated in the next section, this false-positive error has a close-to-Gaussian distribution profile. Therefore, it has the same effect as comparator noise and does not cause sparkle-codes. On the other hand, a comparator meta-stable event is missed by the detection circuit in the case of a false-negative error (Figure 3.13b). This is because the instantaneous timer jitter causes the rising edge of the time-out signal to arrive too late – after the meta-stability is resolved. As a result, a would-be time-out pulse disappears, and no flag bit is raised. In this case, the detection circuit simply fails to catch the meta-stability event, and a sparkle-code might be produced due to insufficient conversion time as if there is not meta-stability detection at all. Despite of still having sparkle-codes, the rate of this false-negative detection error fortunately falls much faster than the exponential decay shape in the case of classic asynchronous SAR. Therefore, the back-end meta-stability correction can significantly reduce the conversion speed penalty. In next section, we will back up all these claims with extensive statistical analysis.

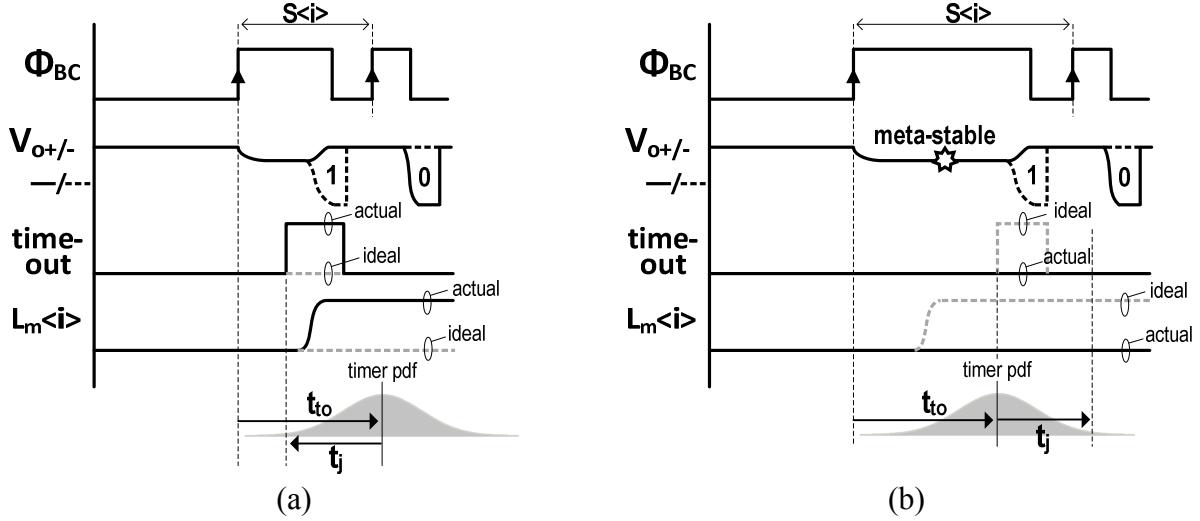


Figure 3.13 Waveforms of (a) a false-positive and (b) a false-negative detection error

3.2.3 Statistical analysis of sparkle-code error-rate

In this section, we will calculate the probability distribution profile of the false-positive errors to show its shape is similar to a Gaussian function, thus does not cause sparkle-code. To demonstrate the back-end meta-stability correction circuit can significantly reduce sparkle-codes, the total probability of false-negative error is calculated and compared against classic asynchronous SAR ADC. Before starting the analysis, we need to make several assumptions: first, we assume the input voltage is uniformly distributed. This is usually a good assumption for most communication systems. Second, the only noise sources in the system is the comparator noise ($V_{n,cmp}$) and timer jitter; all other noise source are ignored. Since the sampling KT/C noise and CDAC nonlinearity have little effect on the sparkle-codes, they are thus ignored for simplicity. The CDAC noise affects the SAR ADC exactly the same ways as the comparator noise, and their contribution is usually much smaller. Thus, they are ignored as well in this calculation.

Like classic asynchronous SAR ADCs, additional conversion time (T_{add}) can be added to allow more time to resolve the meta-stability event. To start the analysis, recall that the false-negative error occurs when the meta-stable event causes the SAR ADC to run out of conversion time (even with added additional time margin, T_{add}) and the detector failed to catch the meta-stability event because of non-zero jitter. Using Equation 3.3 and 3.4, a false-negative error must satisfy the following conditions:

$$\left\{ t_{int} + \tau \ln \left(\frac{V_{DD}}{A_{int} |V_{i,n} + V_N|} \right) < t_{to} + t_j \right. \quad (3.12)$$

$$\left. t_{int} + \tau \ln \left(\frac{V_{DD}}{A_{int} |V_{i,n} + V_N|} \right) > t_{to} + T_{add} \right. \quad (3.13)$$

Where τ is the comparator regeneration time constant; A_{int} is the integration gain; V_N is the instantaneous comparator noise; t_j is overall instantaneous jitter including the contribution of timer and meta-stability latch; t_{to} is the predefined time-out delay explained in the previous

section; $V_{i,n}$ is comparator input voltage at n^{th} bit-cycle in which $1 \leq n \leq 6$. $n = 6$ is assumed to be the MSB cycle, and $n = 1$ is the LSB cycle. The left hand side of both Equation 3.12 and 3.13 represents the comparator delay: $t_{cmp} = t_{int} + \tau_l \ln \left(\frac{V_{DD}}{A_{int}|V_{i,n} + V_N|} \right)$. The first condition (Equation 3.12) describes the scenario that the instantaneous jitter delayed the rising edge of the time-out signal so long (longer than t_{cmp} in meta-stability state) that the meta-stability flag bit is not raised. The second condition (Equation 3.13) suggests that $V_{i,n}$ is so small such that t_{cmp} is longer than the maximum allowable resolution time (t_{to}) with additional conversion time added (T_{add}). We can convert Equation 3.12 and 3.13 into the following inequality in voltage domain:

$$V_M e^{-\frac{t_j}{\tau_l}} < |V_{i,n} + V_N| \leq V_M e^{-\frac{T_{add}}{\tau_l}} \quad (3.14)$$

Where the constant $V_M = \frac{V_{DD}}{A_{int}} \cdot e^{-\frac{t_{to} - t_{int}}{\tau_l}}$ represents a predefined voltage range within which comparator meta-stability event occurs. We will refer to it as meta-stability range in the rest of the section. Because of the timer jitter randomly modulating V_M by the factor of e^{-t_j/τ_l} , the meta-stability detector may or may not catch the meta-stability event. If the meta-stability is not caught by the detector and the SAR runs out of conversion time (with T_{add} added), a sparkle-code would occur. Note $V_{i,n}$, V_N , and t_j are independent random variables, and V_N and t_j have Gaussian distribution profiles. The probability of this false-negative error at n^{th} bit-cycle can be written as the following volume integral:

$$P_{FN,n} = \iiint_{C_{FN}} f_{V_{i,n}}(V_{i,n}) \cdot \phi\left(\frac{V_N}{\sigma_N}\right) \phi\left(\frac{t_j}{\sigma_j}\right) \cdot dV_{i,n} dV_N dt_j \quad (3.15)$$

Where C_{FN} is the false-negative condition defined by 3.14; $\phi(x) = \frac{2}{\sqrt{\pi}} e^{-\frac{x^2}{2}}$ is the normalized Gaussian function; σ_N and σ_j are variance of the comparator noise and jitter; $f_{V_{i,n}}(V_{i,n})$ is the probability density functions (PDF) of the comparator input voltage at n^{th} bit-cycle. Since false-negative error at any bit-cycle may result in sparkle-codes, Equation 3.15 needs to be evaluated for all the 5 bit-cycles with the correction bits (<5:1>) to obtain the overall sparkle-code probability. The total sparkle-code error probability of the SAR is:

$$P_{spa} = \sum_{n=1}^5 P_{FN,n} \quad (3.16)$$

Every term in Equation 3.15 and 3.16 are known except $f_{V_{i,n}}(V_{i,n})$. To find $f_{V_{i,n}}(V_{i,n})$ for every bit-cycle, we need first examine the following 3 cases:

$$\text{Case 1: } V_M e^{-\frac{T_{add}}{\tau}} < |V_{i,n} + V_N| < V_M e^{-\frac{t_j}{\tau}} \quad (3.17)$$

$$\text{Case 2: } V_{i,n} + V_N > V_M e^{-\frac{t_j}{\tau}} \quad (3.18)$$

$$\text{Case 3: } V_{i,n} + V_N < -V_M e^{-\frac{t_j}{\tau}} \quad (3.19)$$

In Case 1, the comparator is not meta-stable because its input is outside the range that the SAR runs out of conversion time. However, a meta-stable flag bit is nevertheless raised because the input falls inside the meta-stable range widened by the jitter ($V_M e^{-t_j/\tau}$). This constitute a false-positive detection error. Recall that in this scenario, the algorithm of the digital back-end would ignore the current and the rest of bit-cycle decisions as explained in the previous section. Therefore, we do not have to analyze the followings bit-cycles in Case 1. The effective ADC residue error is simply $V_{i,n}$. The PDF of this false-positive residue error at n^{th} cycle can be written as:

$$f_{FP,n}(V_e) = f_{V_{i,n}}(V_e) \cdot \iint_{C_{FP}} \phi\left(\frac{V_N}{\sigma_N}\right) \phi\left(\frac{t_j}{\sigma_j}\right) \cdot dV_N dt_j \quad (3.20)$$

Where C_{FP} is the false-positive condition defined by Equation 3.17, V_e is the residue error amplitude. In Case 2 and 3, the comparator is not meta-stable, the current bit decision is taken into account in the digital back-end. Since the SAR logic will change the CDAC value and move to next bit-cycle, we need to look at the comparator input of the next bit-cycle ($n - 1$):

$$V_{i,n-1} = \begin{cases} V_{i,n} - \frac{V_r}{2^{7-n}} & \text{if case 2} \\ V_{i,n} + \frac{V_r}{2^{7-n}} & \text{if case 3} \end{cases} \quad (3.21)$$

Where V_r is the CDAC differential reference voltage, which is also equal to the ADC full swing voltage. To find the PDF of comparator input for the cycle ($n - 1$), we can first calculate the PDF of $V_{i,n}$ under the condition of both Case 2 and 3 separately. From Equation 3.16, the PDF of $V_{i,n-1}$ is simply the sum of the 2 PDF's shifted by the amount of $\pm \frac{V_r}{2^{7-n}}$:

$$\begin{aligned} f_{V_{i,n-1}}(V_{i,n-1}) &= f_{V_{i,n}}\left(V_{i,n-1} + \frac{V_r}{2^{7-n}}\right) \cdot \iint_{C_+} \phi\left(\frac{V_N}{\sigma_N}\right) \phi\left(\frac{t_j}{\sigma_j}\right) \cdot dV_N dt_j + f_{V_{i,n}}\left(V_{i,n-1} - \frac{V_r}{2^{7-n}}\right) \\ &\quad \cdot \iint_{C_-} \phi\left(\frac{V_N}{\sigma_N}\right) \phi\left(\frac{t_j}{\sigma_j}\right) \cdot dV_N dt_j \quad (3.22) \end{aligned}$$

Where C_+ and C_- are the conditions defined by Equation 3.18 and 3.19.

Now, we have derived the complete set of equations to calculate the sparkle-codes: We can calculate the PDF of the comparator input from the previous bit-cycle using Equation 3.18, 3.19, 3.22. For each of the 6 bit-cycles, we can calculate probability of the false-negative errors using Equation 3.15, and the PDF of the false positive errors can be calculated using Equation 3.20. The sparkle-code is just the sum of the probability of false-negative errors of all the first 5 MSB cycles

(the LSB does not need a detection bit), and the overall ADC conversion error (excluding sparkle-code) is just the sum of the false-positive PDFs of all 5MSB cycles plus the PDF of the unconverted residue voltage left at the end of LSB. The set of equations to calculate sparkle-code error-rate and ADC conversion error PDF is summarized below:

1. At start, we have: $f_{i,6}(V_{i,6}) = \begin{cases} \frac{1}{V_r}; & \text{if } -\frac{V_r}{2} < V_{i,5} < \frac{V_r}{2} \\ 0; & \text{otherwise} \end{cases}$

2. To move to cycle $(n - 1)$:

$$f_{V_{i,n-1}}(V_{i,n-1}) = f_{V_{i,n}}\left(V_{i,n-1} + \frac{V_r}{2^{7-n}}\right) \cdot \iint_{C_+} \phi\left(\frac{V_N}{\sigma_N}\right)\phi\left(\frac{t_j}{\sigma_j}\right)dV_N dt_j + f_{V_{i,n}}\left(V_{i,n-1} - \frac{V_r}{2^{7-n}}\right) \cdot \iint_{C_-} \phi\left(\frac{V_N}{\sigma_N}\right)\phi\left(\frac{t_j}{\sigma_j}\right)dV_N dt_j$$

$$\text{Where } 0 < n < 5, \text{ and } \begin{cases} C_+: V_{i,n} + V_N > V_M e^{-\frac{t_j}{\tau}} \\ C_-: V_{i,n} + V_N < -V_M e^{-\frac{t_j}{\tau}} \end{cases}$$

3. The false-negative probability at n^{th} cycle is:

$$P_{FN,n} = \iiint_{C_{FN}} f_{V_{i,n}}(V_{i,n})\phi\left(\frac{V_N}{\sigma_N}\right)\phi\left(\frac{t_j}{\sigma_j}\right)dV_{i,n}dV_N dt_j$$

$$\text{Where } 2 \leq n \leq 6, \text{ and } C_{FN}: V_M e^{-\frac{t_j}{\tau}} < |V_{i,n} + V_N| \leq V_M e^{-\frac{T_{add}}{\tau}}$$

4. The false-positive error PDF is:

$$f_{FP,n}(V_e) = f_{V_{i,n}}(V_e) \cdot \iint_{C_{FP}} \phi\left(\frac{V_N}{\sigma_N}\right)\phi\left(\frac{t_j}{\sigma_j}\right)dV_N dt_j$$

$$\text{Where } C_{FP}: V_M e^{-\frac{T_{add}}{\tau}} < |V_{i,n} + V_N| \leq V_M e^{-\frac{t_j}{\tau}}$$

5. The PDF of the ADC residue error is:

$$f_{res}(V_e) = \sum_{n=2}^6 f_{FP,n}(V_e) + f_{V_{i,0}}(V_e)$$

6. The ADC sparkle-code probability is:

$$P_{spa} = \sum_{n=1}^5 P_{FN,n}$$

Unfortunately, $f_{res}(V_e)$ and P_{spa} cannot be solved in closed form. However, the calculations can be easily scripted in any common numerical analysis tools such as Matlab. Assuming $V_r = 0.4V$, $V_{dd} = 1.05V$, $A_{int} = 2.3$, $\sigma_n = 1.2mV$, $\sigma_j/\tau = 0.2$, $T_{add}/\tau_l = 2$,⁷ the PDF of the comparator input voltage for each bit-cycle is shown in Figure 3.14. Note that the ADC input PDF ($f_{vi,6}$) has a uniform distribution, thus a flat line. As expected, the binary search algorithm reduces the distribution range of the comparator input for each succeeding bit-cycle is reduced by factor of $\frac{1}{2}$. As a result, the comparator input voltage becomes more concentrated around 0. Figure 15a shows the PDF of final residue error of the ADC including false-positive detection errors, $f_{res}(V_e)$. Apparently, the shape of $f_{res}(V_e)$ is very close to a Gaussian function. Its cumulative distribution function (CDF) further confirms it (Figure 15b). A complimentary error function (Q-function) with 1.8mV variance is plotted for reference. The CDF of V_e is bounded by the Q-function even for the BER as low as $1e-15$. It confirms the false-positive detection error as well as comparator thermal noise does not cause any sparkle-code.

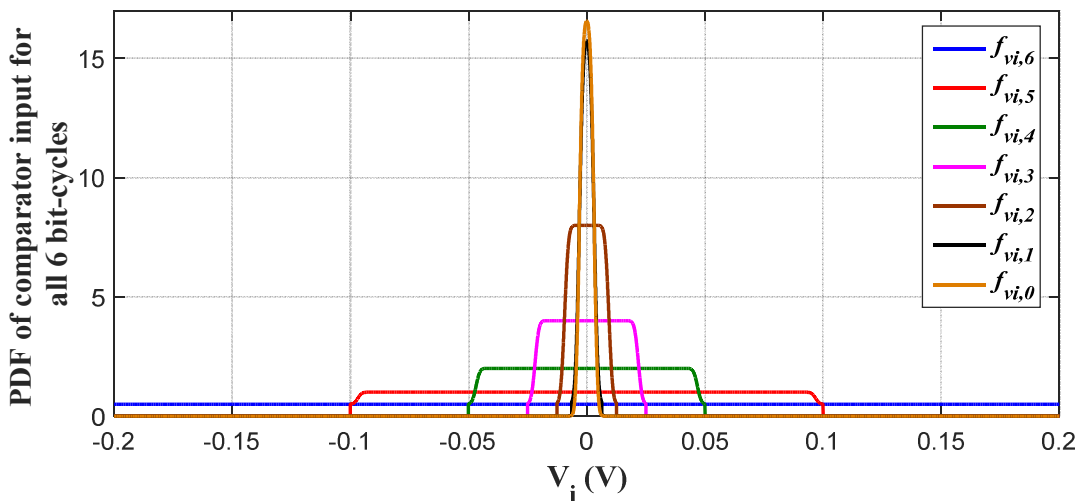


Figure 3.14 The PDF of comparator input for all 6 bit-cycles

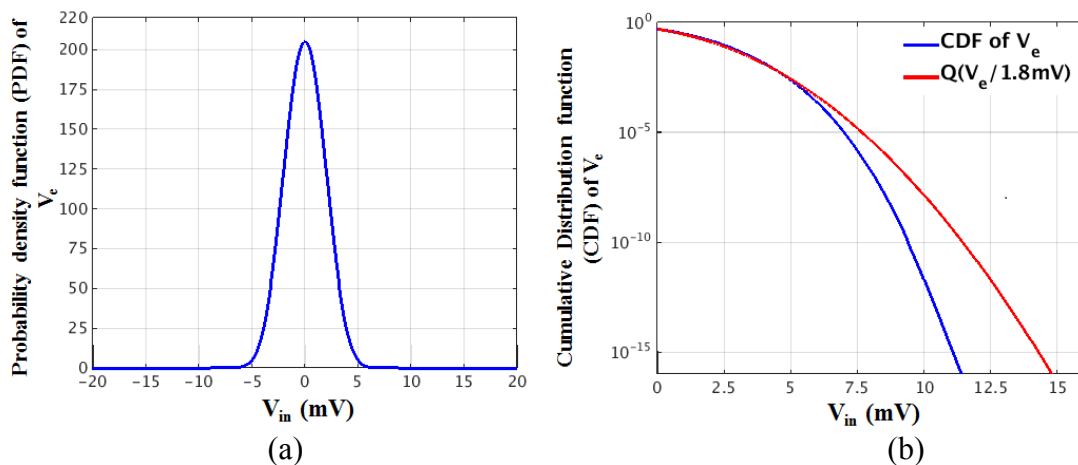


Figure 3.15 (a) PDF and (b) CDF of the ADC residue error

⁷ The parameters used here are extracted from the 6-bit 46GS/s ADC design. Note σ_j is intentionally overestimated using $\sigma_j/\tau = 0.2$ to leave some margin. The actual σ_j is mainly caused by thermal/flick noise of the discharge current source inside the current-limit inverter in Figure 3.11.

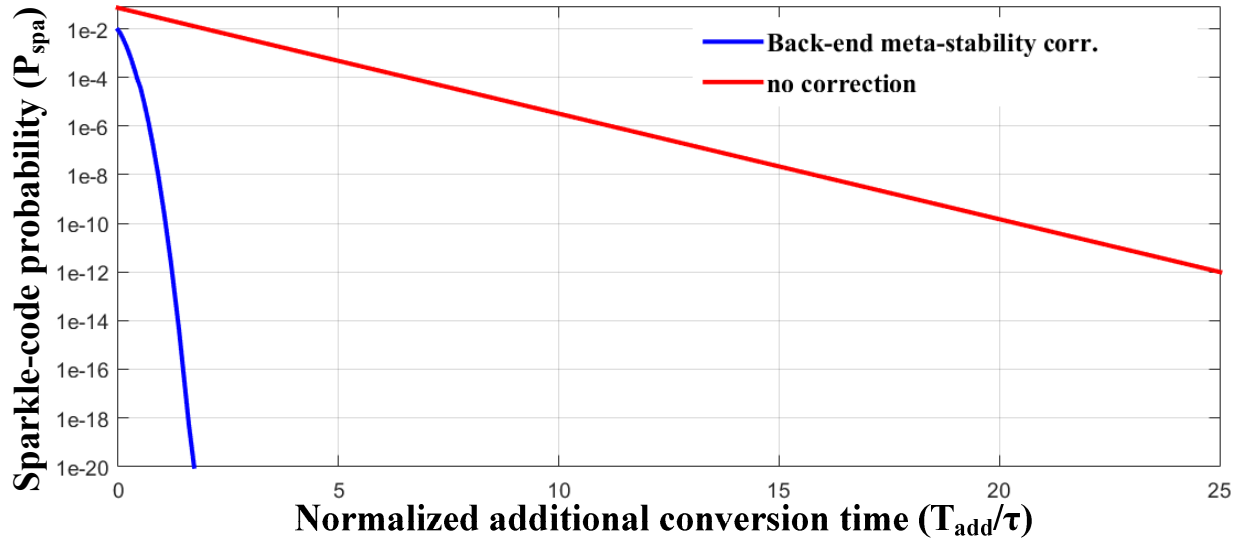


Figure 3.16 Additional conversion time vs. the sparkle-code error-rate for asynchronous SAR ADC with and without back-end meta-stability correction.

The sparkle-code probability after back-end meta-stability correction is plotted in Figure 3.16 as a function of normalized additional conversion time. The sparkle-code probability for a classic asynchronous SAR ADC without correction is also shown for reference. With correction, the sparkle-code rate falls like a waterfall curve. Correction is able to reduce the sparkle-code error-rate to $<10^{-15}$ with roughly only 2τ additional time, while the speed penalty for the same error-rate level without correction is greater than 25τ . This waterfall shape of corrected sparkle-code error-rate is not so surprising, if we realize the cause for the sparkle-codes after the correction is jitter. Intuitively, since jitter has a Gaussian distribution profile, the CDF of any jitter-caused false negative error should roughly follow the Q-function. With this argument, jitter-caused sparkle-code should fall down as the Q-function as the timing is more and more relaxed. With the back-end meta-stability correction method, the mechanism to produce sparkle-codes is changed from meta-stability to jitter, and thus its speed penalty is reduced.

Chapter 4

6b 46GS/s hierarchically time-interleaved asynchronous SAR ADC

Using the circuits and ideas introduced earlier, a 6b 46GS/s 72-channel hierarchically time-interleaved asynchronous SAR ADC prototype is fabricated in 28nm FDSOI process. In this chapter, the circuit level implementation will be discussed, and the measurement results will be presented. Section 4.1 gives a block-level overview of the entire ADC chip, including sampling network configuration, clock generation and distribution, and organization of sub-ADCs. Section 4.2 discusses the detailed implementation of cascode T&H circuits. Section 4.3 focuses on the circuits used for the clock generation and distribution. The implementation of the asynchronous SAR sub-ADC with back-end meta-stability correction is discussed in Section 4.4. At last, the measurement results of the ADC chip is presented in Section 4.5.

4.1 System overview

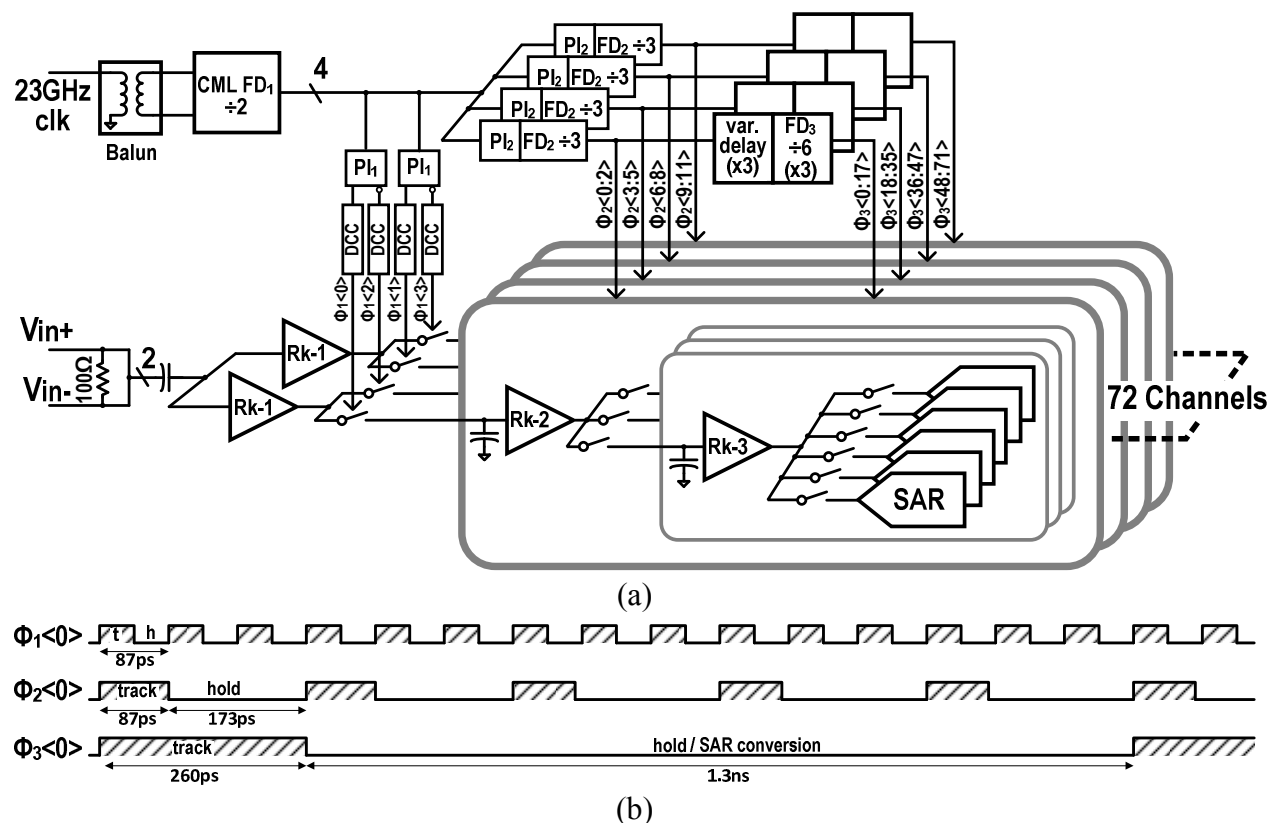


Figure 4.1 (a) block diagram and (b) the timing diagram of the ADC chip

As shown in Figure 4.1a, the broadband differential analog inputs are terminated on-chip with 100Ohm differential resistor and AC-coupled to the front-end samplers (Rk-1). The entire sampling network of the ADC consists of 3 ranks of cascode-based T&H circuits. This 3-rank 72-way sampling network is designed using method developed in Section 2.2.1. Each of the 2 Rk-1 T&H circuits functions as 1-to-2 analog multiplexors driven by 4 11.5GHz CMOS clocks in quadrature phase ($\Phi_1\langle 0:3 \rangle$). Together, they sample the continuous-time input signal into quadrature time-interleaved samples at 11.5GS/s. As shown in Figure 4.1b, the 50% duty-cycle Φ_1 provides Rk-1 87ps track time and 87ps of hold time. The reason to use the 2 2-way time-interleaved front-end T&H circuits is because the 11.5GHz quadrature clock with 50% duty-cycle can be readily generated with simple frequency divider circuit. Following Rk-1, the 4 11.5GS/s samples are further de-multiplexed by 4 Rk-2 T&H circuits into 12 time-interleaved samples at the rate of 3.83GS/s. To accomplish this, each of the 4 Rk-2 T&H functions as 1-to-3 de-multiplexor controlled by 3 3.8GHz non-overlapping clocks with 33% duty-cycle – $\Phi_2\langle 0:2 \rangle$, $\Phi_2\langle 3:5 \rangle$, $\Phi_2\langle 6:8 \rangle$, or $\Phi_2\langle 9:11 \rangle$. Each Rk-2 T&H has 87ps of track time and 173ps of hold time. Then, each of the 12 samples is de-multiplexed one last time by 12 Rk-3 T&H circuits. Each Rk-3 functions as 1-to-6 de-multiplexor controlled by 6 638MHz clocks with equally spaced phases (Φ_3). Finally, after 3 ranks of de-multiplexing, the 72 time-interleaved samples are converted by 72 6-bit asynchronous SAR sub-ADC's. The 72 Φ_3 's are also used as reset clocks for the SAR ADC's. The conversion time for each SAR ADC is 1.3ns. As highlighted in chapter 2, although a total of 88 clocks are used in the sampling network, only the 4 11.5GHz front-end clock are jitter-critical and require large-size power hungry drivers. The jitter and edge-rate requirements of the rest of the 84 clocks can be greatly relaxed so that small CMOS inverters are used to distribute them.

To generate the 88 clocks for the entire sampling network from a single-ended 23GHz external clock input, a clock generation and distribution network is implemented on chip. The 23GHz single-ended clock input is converted into differential clocks by a 23GHz transformer-based on-chip balun. The input impedance of the balun together with its load is designed to be 50Ohm at its resonance. The 23GHz differential clock drives a $\div 2$ current-mode-logic (CML) frequency divider. The $\div 2$ CML divider outputs the quadrature phased 11.5GHz clock ($\Phi_1\langle 0:3 \rangle$) needed for Rk-1 T&H circuits. In order to calibrate out any sampling time skew among the 4 quadrature clocks caused by mismatch, 2 differential current-mode phase interpolators (PI_1) followed by 4 duty-cycle correction circuits (DCC) are used. PI_1 adjusts the skew between 0° and 90° clocks, and between 180° and 270° clocks. The DCC fine tunes out any skews between the complementary phases, 0° and 180° clocks, 90° and 270° clocks. Another function of the DCC circuits is to convert the CML clocks into full-swing CMOS clocks. As mentioned earlier, since the 4 11.5GHz clocks are jitter-critical, large sized CMOS drivers with fewest possible CMOS gates are used. The 12 $\Phi_2\langle 0:11 \rangle$ required for Rk-2 T&H circuits are derived from $\Phi_1\langle 0:3 \rangle$ using 4 $\div 3$ CMOS frequency dividers. Each of these dividers is designed to output 3 equal spaced clock phases at 3.8GHz. To coarsely align $\Phi_2\langle 0:11 \rangle$ to $\Phi_1\langle 0:3 \rangle$ such that the falling edge of Φ_2 is slightly before the rising edge of Φ_1 , 4 phase interpolators (PI_2) are inserted in front of the dividers. Similarly, the 72 clocks for Rk-3 ($\Phi_3\langle 0:71 \rangle$) are derived from $\Phi_2\langle 0:11 \rangle$ using 12 $\div 6$ frequency dividers. To align the rising edges of $\Phi_3\langle 0:71 \rangle$ to falling edges of $\Phi_2\langle 0:11 \rangle$, 12 CMOS variable-delay-lines are used instead of phase interpolators to save area.

The 72 6-bit asynchronous SAR sub-ADCs operate at 638MHz, the frequency of Φ_3 . Each SAR has 1.3ns to complete all 6 bit-cycles. The outputs of the SAR are temporarily stored on-chip using a memory of 72 samples wide and 80 samples deep. After the memory is filled, the data are sent off-chip at low speed for processing. The entire ADC core has 2 supply domains: all cascode T&H circuits are connected to 1.6V supply, and the rest of the core including clock generation and distribution circuits, sub-ADCs, and reference DACs are connected to 1.05V supply.

4.2 Sampling circuits

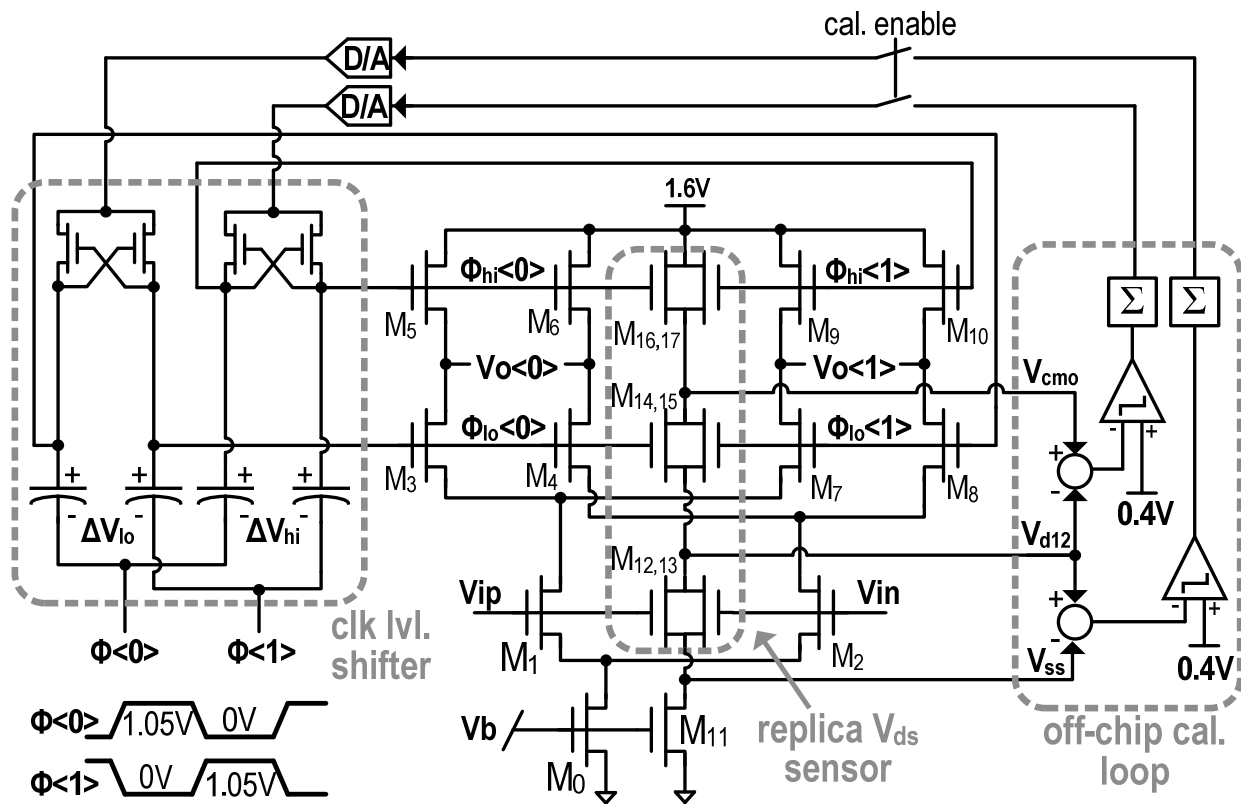


Figure 4.2 The implementation of Rk-1 cascode T&H circuit.

Figure 4.2 shows the implementation of the Rk-1 cascode T&H circuit. The sampler core (M_{0-10}) is a 2-way time-interleaved cascode-based sampler with saturation NMOS load: M_{0-2} is the differential input pair with tail current source; $M_{3,4}$ and $M_{7,8}$ are 2 cascode device pairs operating on complementary phases; $M_{5,6}$ and $M_{9,10}$ are the saturation NMOS loads. The supply voltage is 1.6V to provide enough headroom for a stack of 4 NMOS. As mentioned earlier, the clocks that drive the cascode devices ($M_{3,4,7,8}$) and the NMOS loads ($M_{5,6,9,10}$) must be level shifted to the appropriate voltages – above 1.05V – to keep them in saturation. The clock level-shifter is implemented using a NMOS pump circuit [8] [33]. The basic operation of this pump circuit is based on putting the desired ΔV onto a capacitor during the low clock phase, and then use the charged capacitor as a floating battery when clock is high. 2 pump circuits are used: 1 for the

clocks that drive the cascode devices ($M_{3,4,7,8}$), and the other boosts the gate voltages of NMOS loads ($M_{5,6,9,10}$). The amount of ΔV shifted by the pumps can be adjusted by 2 on-chip voltage DACs. A replica sensor circuit (M_{11-17}) is used to measure V_{ds} of all the transistor of the sampler core. The V_{ds} of all the active devices in the sampler core (M_{0-6} or $M_{0-2,7-10}$) are designed to be 0.4V to reduce the effect of channel-length modulation, and an off-chip feed-back loop is activated at start-up to force all the V_{ds} to be the correct value regardless of process and temperature variation. The cascode T&H circuits for Rk-2 and Rk-3 are implemented in a similar way.

4.3 Clock generation and distribution

This section describes circuit implementations for clock generation and distribution network. Section 4.3.1 discuss frequency divider circuits used in the ADC chip. The front-end CML based divider (FD_1) and the CMOS based divider FD_2 are discussed in detail. Since the implementation of the low frequency divider FD_3 is similar to FD_2 , it will not be discussed here. Section 4.3.2 focuses on phase interpolator and duty-cycle correction circuits. Only one of the two phase interpolators, the front-end phase interpolator PI_1 is discussed. PI_2 is implemented in a similar way, and thus skipped. At last, section 4.3.3 discusses the variable delay line implementation.

4.3.1 Frequency divider circuits

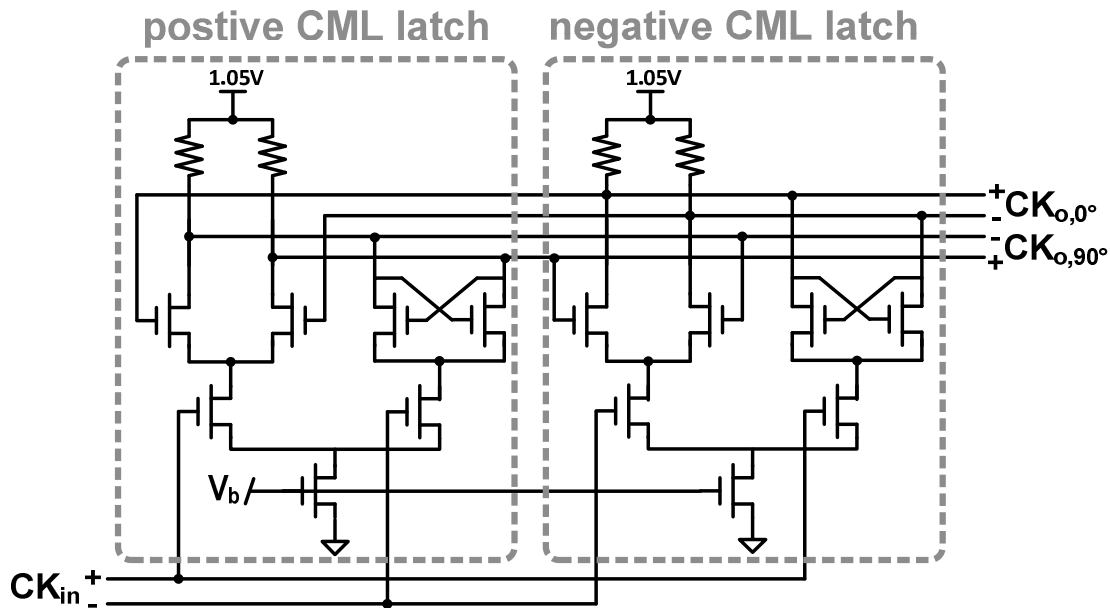


Figure 4.3 Schematic of front-end frequency divider (FD_1)

As shown in Figure 4.3, the front-end $\div 2$ frequency divider (FD_1) is a classic CML divider with 2 back-to-back connected CML latches. The differential clock inputs to the divider (CK_{in}) are at 23GHz. The nominal peak to peak swing is 600mV. The 2 latches are transparent on opposite clock phases. The output nodes of both CML latches are tapped out to produce 11.5GHz quadrature clock outputs. Since the device mismatch as well as the mismatch of the metal wire

routing can cause phase imbalance among the quadrature clocks, the outputs from FD_1 cannot be directly used to sample the ADC input. 2 phase interpolators and 4 duty-cycle-correction circuits take the outputs of from FD_1 and calibrate out any phase imbalance.

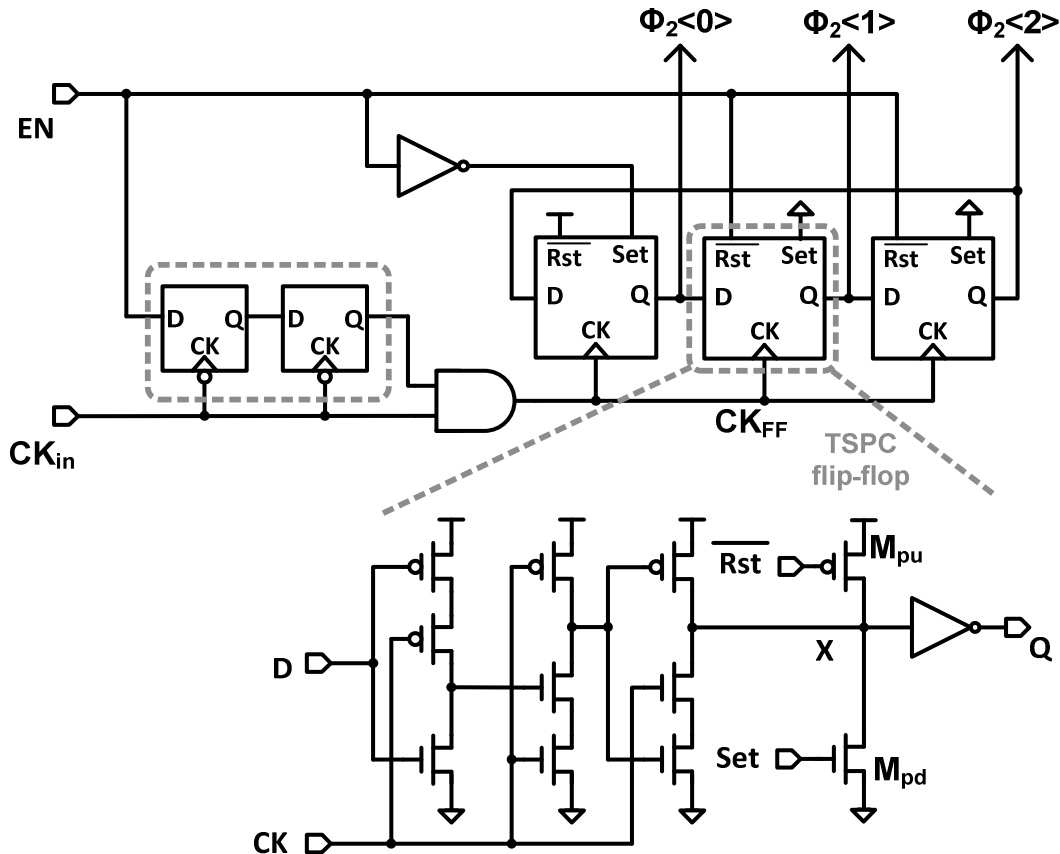


Figure 4.4 Schematic of the first CMOS frequency divider (FD_2)

Following FD_1 , the $\div 3$ CMOS frequency divider (FD_2) is implemented with 3 true single-phase clocking (TSPC) flip-flops connected in a loop (Figure 4.4). The clock input to the divider is at 11.5GHz with full V_{dd} swing. Since the 3 3.8GHz outputs of FD_2 are used as sampling clocks for each Rk-2 T&H circuit, they must be non-overlapping with 33% duty-cycle. To accomplish this, the initial state of 1 TSPC flip-flop is set to high, while the other 2 are set to low. During normal operation, the high-low-low pattern rotates in the flip-flop loop. Therefore, only one of the 3 flip-flop outputs is high at any given time. To initialize the TSPC flip-flops, its storage node is either connected to V_{dd} or ground by the pull-up or pull-down transistors ($M_{pu,pd}$) before divider is enabled – when CK_{FF} is low and X is floating. An enable signal initializes the flip-flops by turning on $M_{pu,pd}$. To guarantee CK_{FF} is low before the divider starts to fire, CK_{FF} is gated by the enable signal. Finally, since the enable comes from the configuration bits of the ADC bit and is not necessarily synchronized to the divider clock (CK_{in}), a cascade of 2 flip-flops are inserted to synchronize the enable to the rise edge of CK_{in} . Since the last frequency divider FD_3 is similar to FD_2 except connecting 6 flip-flops in a loop instead of 3 to implement $\div 6$ function, it will not be discussed here.

4.3.2 Phase interpolator and duty-cycle correction circuits

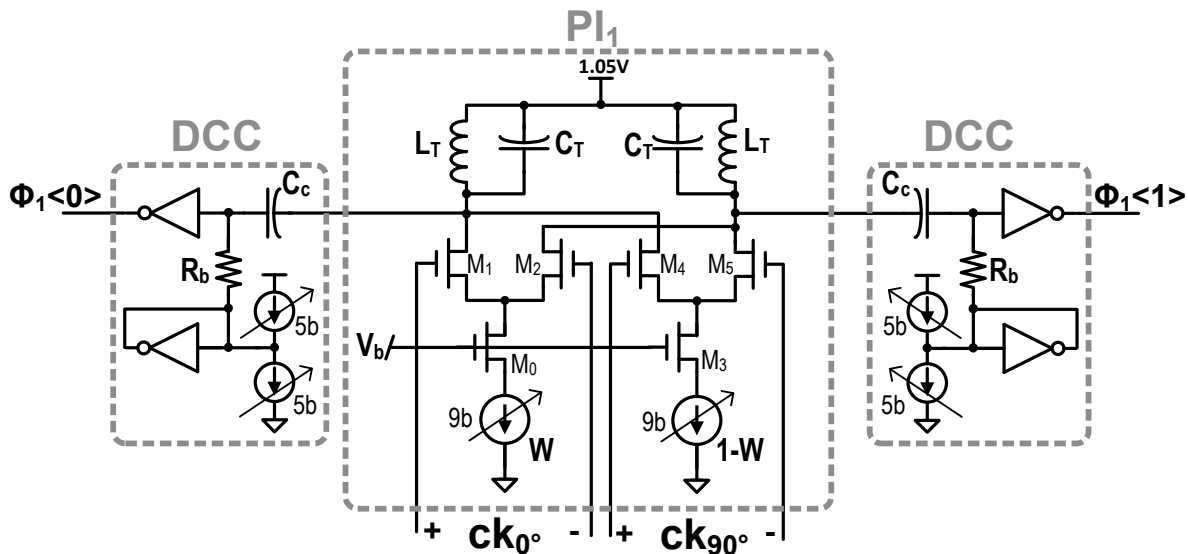


Figure 4.5 Schematic of the front-end phase interpolator (PI₁) and duty-cycle correction circuits (DCC)

PI₁ is used to calibrate out any delay mismatch between the 0° and 90° clocks produced by FD₁. As shown in Figure 4.5, the front-end phase interpolator (PI₁) is based on mixing the currents of the 0° and 90° clock inputs (CK_{0°,90°}) with different weights [34] [35]. The 2 23GHz input clocks drive 2 differential pairs (M_{1,2} and M_{4,5}) with 2 separate tail current DACs. The output currents of the differential pairs are summed together and then fed into a LC tank load (L_T, C_T). The resonant frequency of the tank is designed to be 11.5GHz. The use of LC tank as load instead of resistor load is three fold: first, since the load capacitors and parasitic capacitors of PI₁ is absorbed into C_T, the impedance of tank at resonance can be much higher than the resistor load. In the case of resistance load, no matter how large the resistance value is used, the maximum impedance at 11.5GHz is limited by the load capacitance and wiring capacitance, which can be significantly lower than $Q_T L_T \omega$. Secondly, the band-pass shape of the LC tank impedance filters out the low frequency noise resulting in lower jitter contribution. Finally, the LC tank reduces the distortion tones – especially the even order harmonics – produced by switching the differential pairs (M_{1,2} and M_{4,5}), thus improves the its outputs duty-cycle. With LC tank as the load, the tuning range of the following duty-cycle correction stage can be greatly relaxed, which enables a single-stage DCC design. The control codes of the current DACs are set such that the total current of the 2 DACs are always equal to the maximum current of one current DAC. The cascode transistors (M_{0,3}) are added to reduce parasitic capacitances at the common source nodes of the differential pairs. It can be shown that the input-output delay of PI₁ can be approximated as:

$$t_d = W \cdot \frac{T}{4} + t_0 \quad (4.1)$$

Where $W = \frac{D}{2^B}$ is the weighting factor; B and D are the number of bits and the control codes of the current DAC; T is the period of the clock inputs; t_0 is a constant term representing the intrinsic

delay of the PI. In this design, 9-bit current DACs are used to reduce the nominal step size to 0.18° . After the calibration, the residue timing-skew caused spurious tone are designed to be well below the noise floor of the 5-ENOB ADC. Note that the tuning range of PI_1 is designed to be 90° or $T/4$ – instead of 360° or T – to relax the resolution of current DACs.

The duty-cycle correction circuit (DCC) following PI_1 is used to calibrate out any phase imbalance between $\Phi_2\langle 0 \rangle$ and $\Phi_2\langle 1 \rangle$. It also converts the 11.5GHz sinusoid input into a square wave with full- V_{dd} swing. The correction circuit core is simply an AC coupled inverter. The duty-cycle is adjusted by changing the DC bias of the inverter input. As mentioned earlier, since the sinusoid input to the DCC already has close to 50% duty-cycle, the DCC only needs to have a small tuning range. To limit the DCC tuning range and to relax the adjustment resolution, a replica self-biased inverter is used to generate the nominal input DC bias, and bias adjustment is done by tuning the current injected or taken away from the input node of replica inverter. 2 5-bit current DACs are used to move the DC bias up and down by $\sim 50\text{mV}$, causing the duty-cycle to vary by $\pm 10\%$. The nominal step size is $\sim 1\%$.

4.3.3 Variable delay line

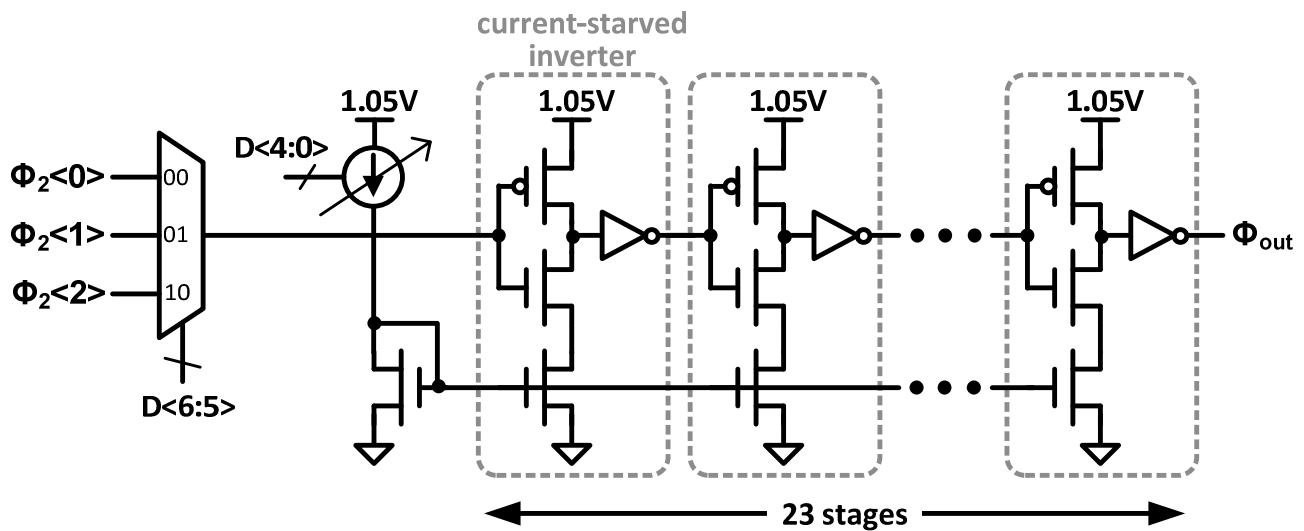


Figure 4.6 Schematic of variable-delay-line

As shown in Figure 4.6, the variable-delay-line used to align Φ_3 's to rising edges of Φ_2 's are implemented with 23 stages of current-starved inverters. A CMOS inverter is inserted between adjacent stages to keep sharp rising and falling edges. The fine delay adjustment is done by changing the current limit of the current-starved inverters using a 5-bit current DAC. The coarse adjustment is done by multiplexing the 3 different phases of Φ_2 . Together, the fine and coarse adjustment guarantee the minimum tuning range of 260ps (a period of Φ_2).

4.4 Sub-ADC implementation

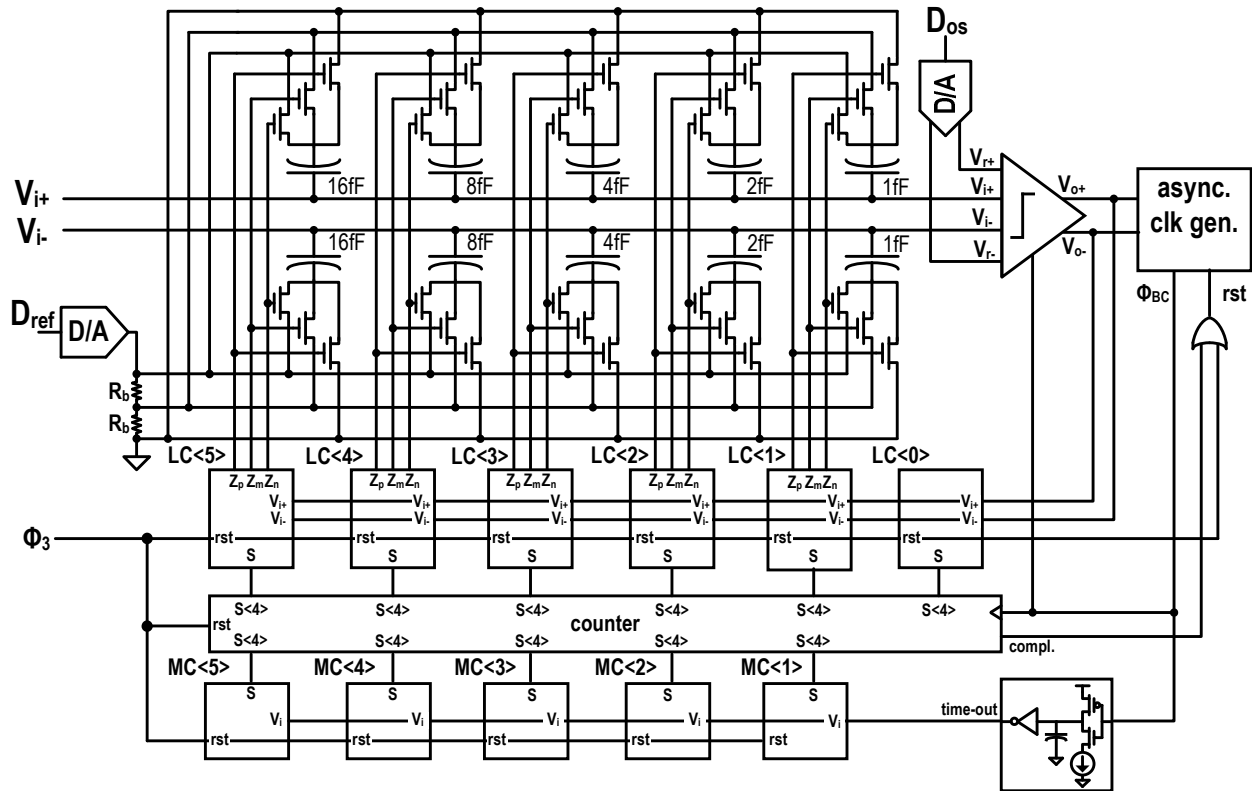


Figure 4.7 Complete schematic of 6-bit asynchronous SAR sub-ADC

Figure 4.7 shows the complete schematic of the 6-bit asynchronous SAR sub-ADC. As highlighted earlier, a 5-bit differential CDAC with small unit-size capacitor ($\sim 1\text{fF}$) is used to save power and area. Note the top-plate CDAC sampling switch is not shown in the figure, because the sampling switch for CDAC in this case is the Rk-3 cascode T&H circuits discussed in section 4.2. The nominal reference voltage (V_{ref}) is 200mV – for 400mV differential V_{pp} , so the bottom-plate switches for the CDAC employ only NMOS-type transistors. 3 reference switches are used: the middle switch controlled by Z_m is connected to $V_{\text{ref}}/2$, establishing the common-mode voltage for the bottom-plates; If closed, the V_{refp} and V_{refn} switches (controlled by $Z_{p,n}$) are connected to $V_{\text{ref}}/\text{ground}$, adding/subtracting the CDAC differential output voltage by $V_{\text{ref}}/2^{6-n}$, where $1 < n < 5$ is the corresponding bit-cycles. The reference voltage of each SAR sub-ADC can be adjusted by a separate voltage DAC to calibrate out any inter-channel gain mismatch. A 2nd voltage DAC adjusts the comparator offset to calibrate inter-channel offset mismatch. The comparator outputs are connected to 6 latch cells (LC). Each LC contains latch circuits to store the decision bits and combination logic circuits to drive the CDAC switches. The meta-stability correction path consists of a timer circuit that drives 5 meta-stability cells (MC) to correct the 5 MSB decisions ($D<5:1>$). Each MC contains a pulse latch circuit to store the correction bits. Since the bit-cycle timing has been discussed in detail in chapter 3, the rest of the section will only focus on the circuit implementation of each building block.

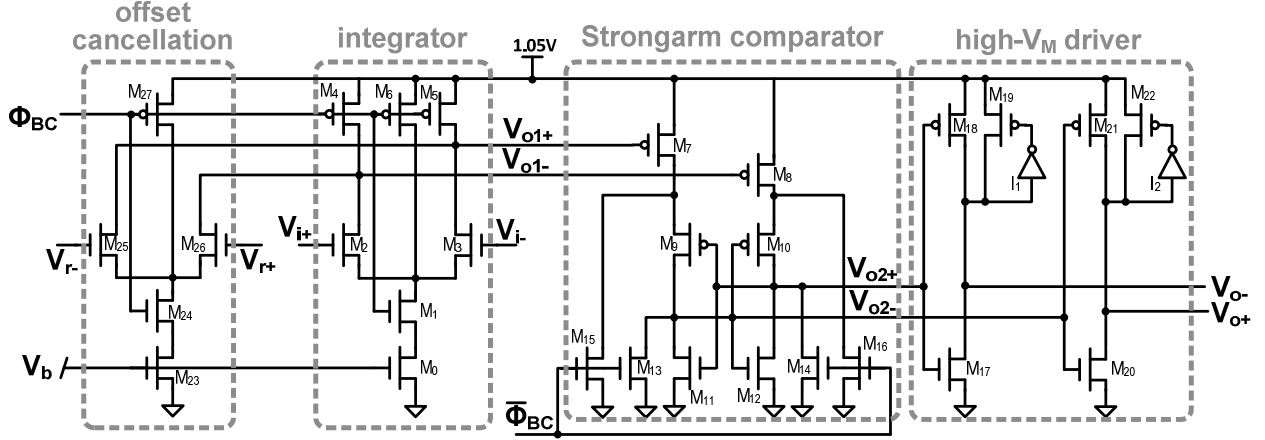


Figure 4.8 Comparator schematic

The comparator circuits shown in Figure 4.8 is based on [36]. It has 2 stages: an NMOS integrator stage front-end followed by a PMOS Strongarm latch without the tail switch. The reason for this 2 stage design instead of a simple 1 stage Strongarm latch is to reduce input referred noise of the comparator. The integrator is designed to provide a voltage gain >2.3 , the noise contribution from the Strongarm comparator is thus reduced by the factor of the integrator gain. The integrator core (M_{0-5}) consists of a differential pair ($M_{2,3}$) with switched tail current source ($M_{0,1}$). During the rest phase ($\Phi_{BC} = 0$), the integrator output nodes ($V_{o1+/-}$) are pulled to V_{dd} by the PMOS switches $M_{4,5}$. At the same time, the tail current source is cut-off by shutting down the NMOS switch M_1 . Note that the common-source node of the differential pair is also pulled to V_{dd} by the PMOS switch M_6 to eliminate comparator hysteresis and to keep the comparator input capacitances independent of V_{in} . If this common-source node is not properly reset, both hysteresis and input-dependent capacitance effect can reduce SNDR of the sub-ADC. During evaluation phase ($\Phi_{BC} = V_{dd}$), the average of V_{o1+} and V_{o1-} is discharged at the rate of $\frac{I_{tail}}{2C_{p1}}$, where C_{p1} is the total parasitic capacitance at $V_{o+,-}$ and I_{tail} is the I_{ds} of M_0 in saturation. At the same time, the differential output voltage is built up at the rate of $\frac{g_{m2,3}V_i}{C_{p1}}$. When the average of V_{o1+} and V_{o1-} is around a V_{th} below the V_{dd} , the PMOS input pair of the Strongarm latch ($M_{7,8}$) is turned on, starting to charge the V_{o2+} and V_{o2-} . Therefore, the effective integration time for the 1st stage integrator is:

$$t_{int} = \frac{2V_{th}}{I_{tail}} C_{p1} \quad (4.2)$$

The effective integration gain can be approximated as:

$$A_{int} = \frac{g_{m1,2}}{C_{p1}} t_{int} = \frac{2V_{th}g_{m1,2}}{I_{tail}} = \frac{2V_{th}}{V_{ov1,2}} \quad (4.3)$$

The overdrive voltage of $M_{1,2}$, $V_{ov1,2}$, can be adjusted in design phase to obtain the desired integration gain by tuning the V_b knob.

The PMOS Strongarm latch in Figure 4.8 requires a complementary bit-cycling clock $\bar{\Phi}_{BC}$. During reset phase ($\bar{\Phi}_{BC} = 1$), the integrator reset switches pull the input to the Strongarm to V_{dd} and cut off its PMOS differential pair ($M_{7,8}$), therefore, a PMOS tail switch is not required. Note this comparator design is very tolerant of the clock skews between $\bar{\Phi}_{BC}$ and Φ_{BC} . In fact, the falling edge of $\bar{\Phi}_{BC}$ can arrive much earlier than the rising edge of Φ_{BC} , because Strongarm latch can wait for the integrator. Since $M_{7,8}$ remain in cut-off before the integration of the 1st stage is finished, without a charge path to V_{dd} , V_{o2+} and V_{o2-} are kept in their initial state (0V). On the other hand, as long as the falling edge of $\bar{\Phi}_{BC}$ arrives before the integration is done – t_{int} after the rising edge of Φ_{BC} , the comparison result would not be affected. In this design, the falling edge of $\bar{\Phi}_{BC}$ is set to arrive one inverter delay earlier than Φ_{BC} .

Following the Strongarm latch, a pair of inverter-based drivers (M_{17-20} and $I_{1,2}$) are used to buffer the Strongarm outputs from the latch and meta-stability cells. The 2 inverter cores consist of $M_{17,18}$ and $M_{20,21}$. The purpose of the small auxiliary inverters ($I_{1,2}$) and additional pull-up PMOS's ($M_{19,20}$) is to shift the low-to-high input transition points (V_M) of the inverters above $V_{dd}/2$. This is because when a meta-stability even occurs, both V_{o2+} and V_{o2-} can remain slightly below $V_{dd}/2$ for a long time before regeneration. Pulling the inverter transition voltage higher can avoid glitches at the driver outputs.

Finally, to provide offset tuning capability for inter-channel offset calibration, an auxiliary integrator (M_{23-27}) is added in parallel to the main integrator. The input of the auxiliary integrator is connected to on-chip generated reference voltage controlled by voltage DACs. To save power and to reduce the impact on the comparator speed, the auxiliary integrator is sized as 1/4 of the main integrator.

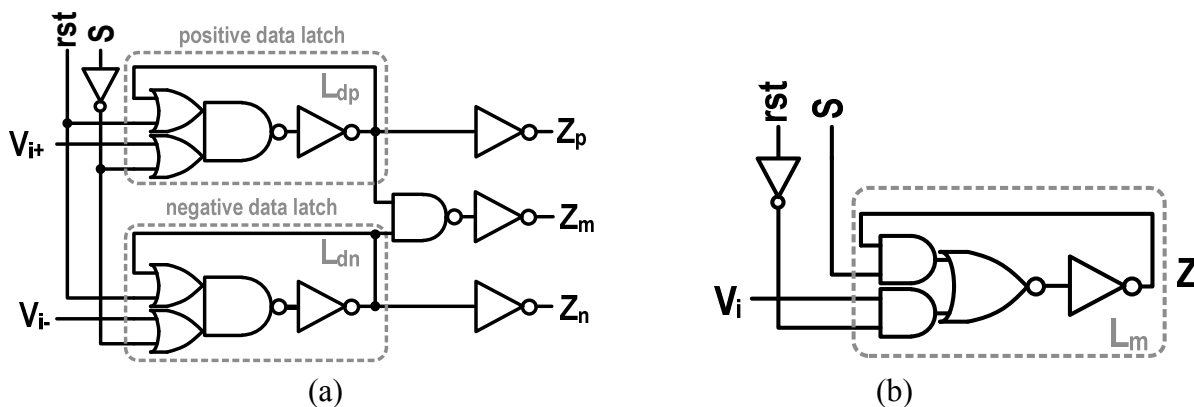


Figure 4.9 Schematic of (a) SAR logic cell (LC) and (b) meta-stability cell (MC) used in the SAR ADC

The core of SAR logic are made of 6 logic cells (LC). Each LC (shown Figure 4.9a) stores the comparator decision bit and produces the control signals for the 3 CDAC switches (Z_p , Z_m , and Z_n). The latch is formed by a complex Or-And-Invert (OAI) gate and an inverter with self feedback. Despite the symbol of the OAI gate looks complex, it can be readily implement with 1 CMOS stage. Note 2 latches are used, one for the positive comparator output (L_{dp}) and the other for negative comparator output (L_{dn}). Although only 1 latch is necessary to store the decision bit, using 2 latches can simplify the combinational logic of the logic cell and greatly relieve their fan-out load, thus reduces the logic delay. The operation of these data latches is as following: during reset time of the SAR ADC (when $\Phi_3 = V_{dd}$), both rst is pulled high, and S is pulled to 0, initializing both L_{dp} and L_{dn} to V_{dd} . This in turn resets Z_p and Z_n to 0, shutting off the V_{refp} and V_{refn} switches. The middle switch controlled by Z_m is closed. After the SAR conversion starts, rst falls low while S remains low. Since V_{i+} , V_{i-} are gated by the OR gate, they are not visible to the $L_{dp/dn}$. Therefore, $L_{dp/dn}$ holds their initial value at V_{dd} . When current bit-cycle starts, S is pulled high by the counter, and L_{dp} and L_{dn} become transparent. However, since the comparator outputs are reset high, $L_{dp/dn}$ will remain high until the decision is made and one of V_{i+} , V_{i-} falls. In other words, $L_{dp/dn}$ can wait for the comparator decision without latching to the incorrect value. This feature significantly relaxes the bit-cycle timing. The rising edge of S can even arrive before comparator fires so that the critical signal path is always from $V_{i+/-}$ to Z_m (4 CMOS gates), not delayed by S . After the decision, the asynchronous clock generator resets the comparator, raises $V_{i+/-}$ to V_{dd} again, while S remains high. However, if a 0 decision is stored in $L_{dp/dn}$, the feedback path of the latch together with the final AND gate keeps its value low regardless of S and $V_{i+/-}$. Therefore, as long as the high window of S covers the 0 decision pulse of the comparator, this latch would not cause any timing violation.

Another way to understand of the functionality of L_{dp}/L_{dn} is to realize they are nothing by 1-shot pulse latches. After reset high, $L_{dp/dn}$ will latch to 0 if it senses a 0-pulse at the input; otherwise, it remains high. Since the comparator outputs are a series of 0-pulses in the SAR ADC, control signal S is added to “pick out” the pulse at the correct bit-cycle. The only requirement for $L_{dp,dn}$ to work properly is to guarantee the decision pulses of comparator is wider than the minimum detectable pulse width of the latch, which is about 2 gate delays.

The time-out pulses the carries that meta-stability information has the opposite sign of data pulses. The time-out signal has a low reset value, and they are pulsed high when a meta-stability event occurs. Therefore, an And-Or-Invert (AOI) gate followed by an inverter is used to implement 1-pulse latch used in the meta-stability cell (MC). As mentioned earlier, the value of MC is not used immediately, but rather in the digital back-end after a series of resampling flip-flops.

4.5 Measurement results

The ADC is fabricated in ST Microelectronics' 28nm FDSOI CMOS process. Figure 4.10 shows the die photo. Thanks to compact SAR sub-ADC design, the size of the ADC core is only 200um X 700um. Most die area is occupied by the 63kb data memory used to store the ADC output bits. To bring broadband analog input signal and 23GHz clock signal on chip, RF probes are used. The differential input is probed from the left side of the die by a ground-signal-signal-ground (GSSG) probe, and the single-ended clock is probed from the right side by a ground-signal-ground (GSG) probe. To leave some clearance for wire-bonding of the supplies and low speed signals at the top/bottom edge of the die, the probe pads must be placed close to the left/right edge. 2 coplanar transmission lines (T-line) are used to route input and clock to the center of the ADC core. A 100Ohm differential GSSG transmission line is used to route the input, and the 50Ohm GSG transmission line is used to route the clock. The ADC core is laid out in a "star" configuration to minimize the routing length to the sub-ADCs: both the input and the clock starts at the center of the ADC core and "radiates" outward up and down, left and right to reach sub-ADCs. The 72 sub-ADCs are organized into 4 18X ADC banks, and placed at the 4 quadrants.

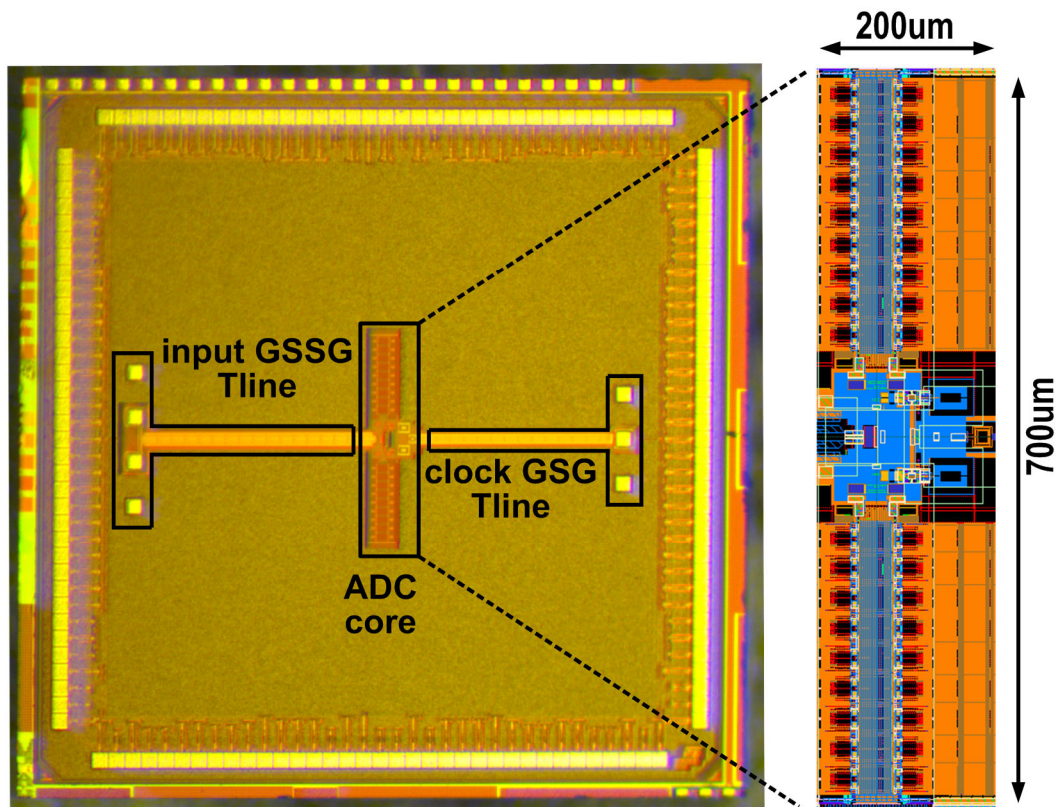


Figure 4.10 die photo

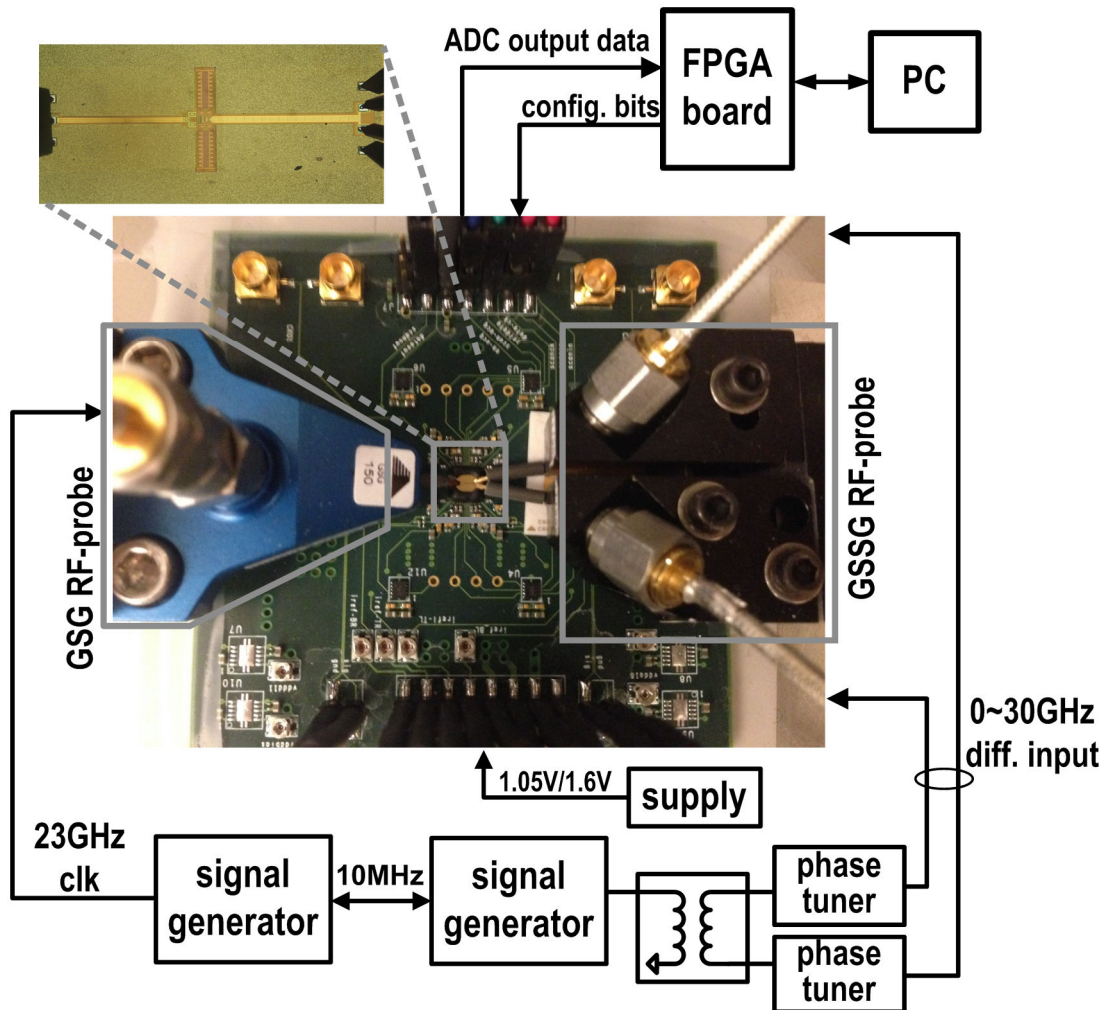


Figure 4.11 Testing setup

In order to test the prototype, the ADC chip is directly attached to a custom designed FR4 PCB board (Figure 4.11). 2 Agilent E8257D PSG signal generators are used to provide the 23GHz clock and ADC input for single-tone test. After the on-chip memory is filled with ADC output bits, the data are first read into a FPGA board with a low speed serial interface, then streamed into PC to perform Fast-Fourier-Transform (FFT) and calculate ENOB. The FPGA board also serves as the interface to configure the ADC chip – to set the control codes of the on-chip DACs and PIs at foreground. Due to limited on-chip memory depth, 3600-pts FFT is taken with rectangular window to plot the output spectrum and calculate ENOB. This requires the input frequency to be locked to the clock in the following relation:

$$f_{in} = \frac{k}{3600} \cdot f_{clk} \quad (4.4)$$

Where k and 3600 do not have any common denominator except 1. To satisfy Equation 4.3 in testing, the clock generator and input signal generator are lock in frequency with 10MHz reference

ports, and the f_{clk} is set to be 23.004GHz, so that $f_{in} = k \cdot 6.5MHz$ is a terminating decimal that can be exactly set on the signal generator. A broadband balun is used to generate fully differential input test-tones from the single-ended signal generator. To tune out any phase imbalance, a pair of 0-50GHz tunable phase shifters with phase-matched cables are used, and the input phase balance is verified with an Agilent DCA 86100 wideband sampling oscilloscope before every single-tone test. The amplitude loss of the balun, phase shifters, as well as the cables is calibrated out by adjusting the output power of the signal generator. 400mV peak-to-peak differential input are used for single-tone tests. Unfortunately, due to unexpected process variations that are not counted during the design phase, the gain of the cascode of 3 ranks T&H circuit is approximately 0.78, which results in a digital peak-to-peak swing of 49-LSB (instead of 63-LSB of full digital swing). This error has caused the measured ENOB to be lower than the 5-bit design target.

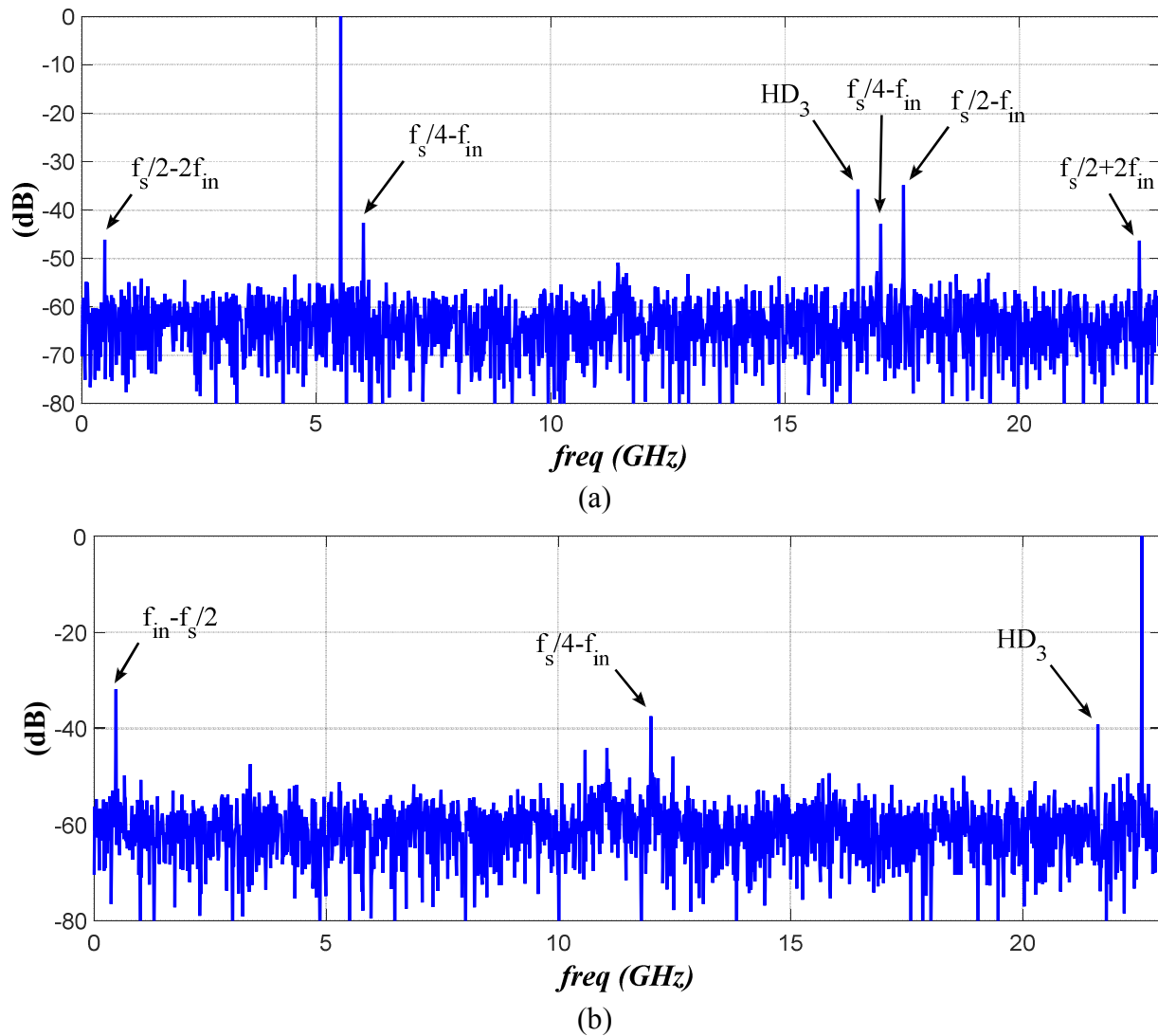


Figure 4.12 ADC output spectrum using 3600-pt FFT for (a) a 5.5GHz input tone, and (b) a 23.5GHz input tone

As mentioned before, inter-stage gain, offset mismatch as well as the front-end sampling time skew is calibrated in foreground with the help of a pilot tone at $\sim 5.5\text{GHz}$. Figure 4.12a shows the ADC spectrum for a 5.5GHz input tone after calibration. Thanks to the cascode T&H with NMOS load, the third order distortion is -36dB , below the noise floor of the 5-ENOB design. The intermodulation products between f_{in} and $f_s/2$ and $f_s/4$ (at $f_s/4 \pm f_{in}$ and $f_s/2 \pm f_{in}$) are caused by residue sampling time skew of the front-end T&H circuit. The SFDR is limited to -35dBc by the intermodulation spur at $f_s/4 - f_{in}$. The remaining visible spurs at $f_s/2 \pm 2f_{in}$ are caused by weak second-order distortion (HD_3) present in one of the Rk-1 T&H circuits. They are well below -45dBc , thus have negligible effect on SNDR. For a 23.5GHz input tone (Figure 4.12b), HD_3 is improved to -39dBc . This is because as the signal frequency becomes higher, its third harmonic distortion can increase beyond with bandwidth of the front-end T&H circuit, and thus get filtered out. On the other hand, the effect of sampling time skew becomes more apparent at high input frequencies, and the SFDR is limited to -32dBc by the intermodulation tone at $f_{in} - f_s/2$.

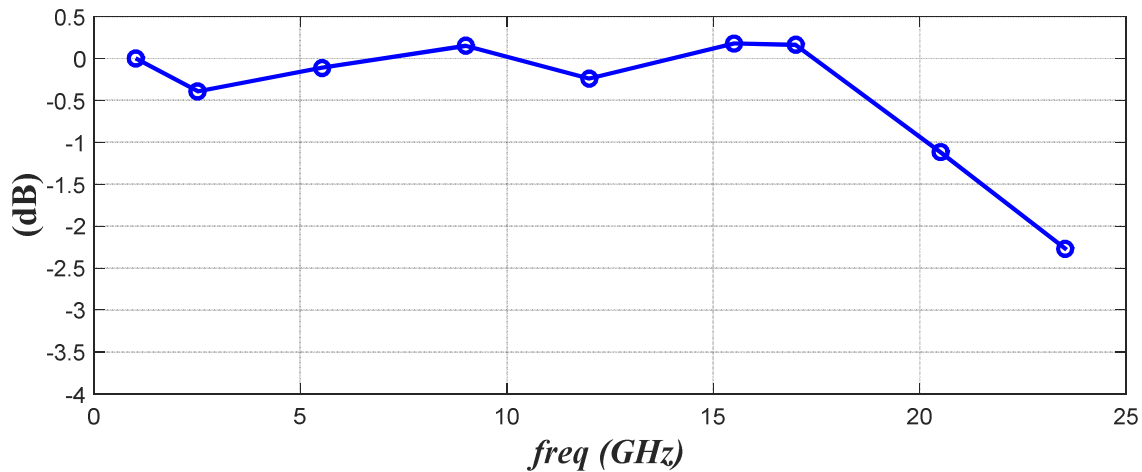


Figure 4.13 Input frequency vs. normalized ADC output amplitude

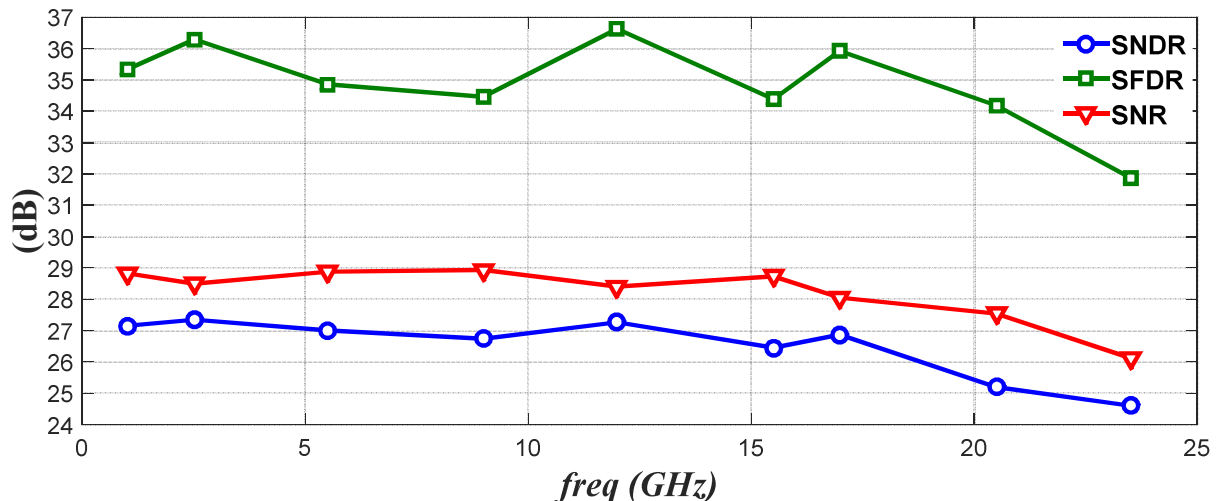


Figure 4.14 ADC input frequency vs. SNDR, SFDR and SNR

As shown in Figure 4.13, the normalized ADC output amplitude remains flat for the entire 23GHz band. The in-band ripple is within $\pm 0.5\text{dB}$. The attenuation at 23.5GHz is -2.3dB . The -3dB bandwidth is beyond the Nyquist rate. Figure 4.14 shows the input frequency vs. the SNDR, SFDR, and SNR of the ADC. The SNDR is 27dB at low frequency resulting in 4.2-ENOB, and at 23.5GHz, the SNDR is 25.2dB resulting in 3.9-ENOB. Note the SNDR curve follows the shape of SNR curve, indicating the dominant error source of the ADC is the random noise either caused by thermal/flicker noise or clock jitter. However, the distortion and sampling time skew have non-negligible effect on ADC performance as the SNDR is lower than SNR by $\sim 2\text{dB}$.

To verify the effectiveness of the back-end meta-stability correction circuits and to measure sparkle-code error rate, a sinusoid with very small amplitude at low frequency is applied to the ADC input such that the difference between successive samples is within 1-LSB. The sparkle-code is defined as the event if the difference between successive samples is $\geq 5\text{LSB}$. Since the SNR is about 29dB (Figure 4.14) for 49-LSB peak-to-peak swing, the noise variance is about 0.61-LSB. The probability for Gaussian distribution random noise to cause a 5LSB error is $Q(5/0.61) \sim 1e-30$. Therefore, any 5LSB errors are very likely caused by comparator meta-stability. Both corrected and uncorrected ADC outputs are sent out for comparison. $\sim 5e-8$ sparkle-code error-rate is observed without correction. With back-end meta-stability correction, no sparkle-code is observed for over $1e10$ samples collected. Some examples of sparkle-rate event is shown in Figure 4.15. As shown, the correction circuit successfully corrected the sparkle-code in all the occurrences.

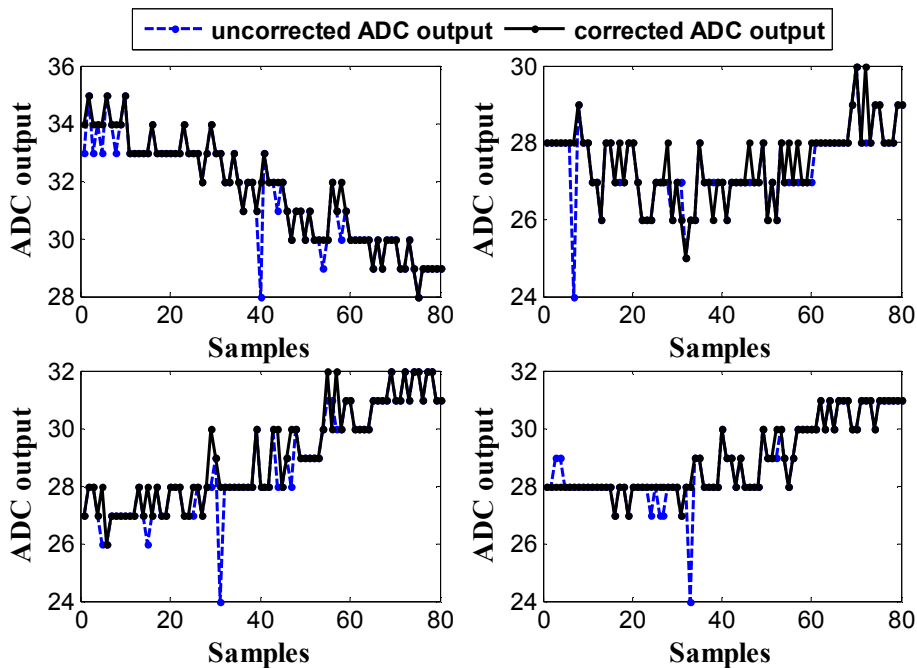


Figure 4.15 Examples of captured sparkle-codes

Figure 4.16 shows the power breakdown of the ADC chip. The power consumption of the entire ADC chip is 381mW. The 3 ranks of T&H circuit (Rk-1, Rk-2, and Rk-3) consumes a total of 132.17mW. The 72 sub-ADCs consumes a total of 160.9mW, and the remaining 87.93mW is consumed by the clock generation and distribution circuits. The ADC achieves a Figure-Of-Merit (FOM) of 0.45pJ/conversion-step at low frequency and 0.56pJ/conversion-step at 23.5GHz. Table I compares this ADC with the state-of-art >46GS/s ADCs. This work has achieved good FOM without sacrificing speed or SNDR. It achieved more 4X better FOM than [1] while have the same ENOB. [3] has similar FOM, but more than 2X lower BW. Although [4] achieved much higher sampling rate and lower FOM, the design heavily rely on high- f_T and extremely low logic delay of the partially-depleted silicon-on-insulator (PDSOI) process. To achieve the desired sampling rate, [4] also adopted higher than nominal supply voltage for the CMOS gates (1.0V/1.1V compared to 0.9V nominal supply). In addition, thanks to the proposed cascode T&H circuits, the ADC achieved the highest input bandwidth among all the state-of-arts. The back-end meta-stability correction circuit successfully keep the ADC sparkle-code free for more than $1e10$ samples collected. To the best of the author's knowledge, this is the only published work that demonstrates $<1e-10$ sparkle-code error rate at this speed.

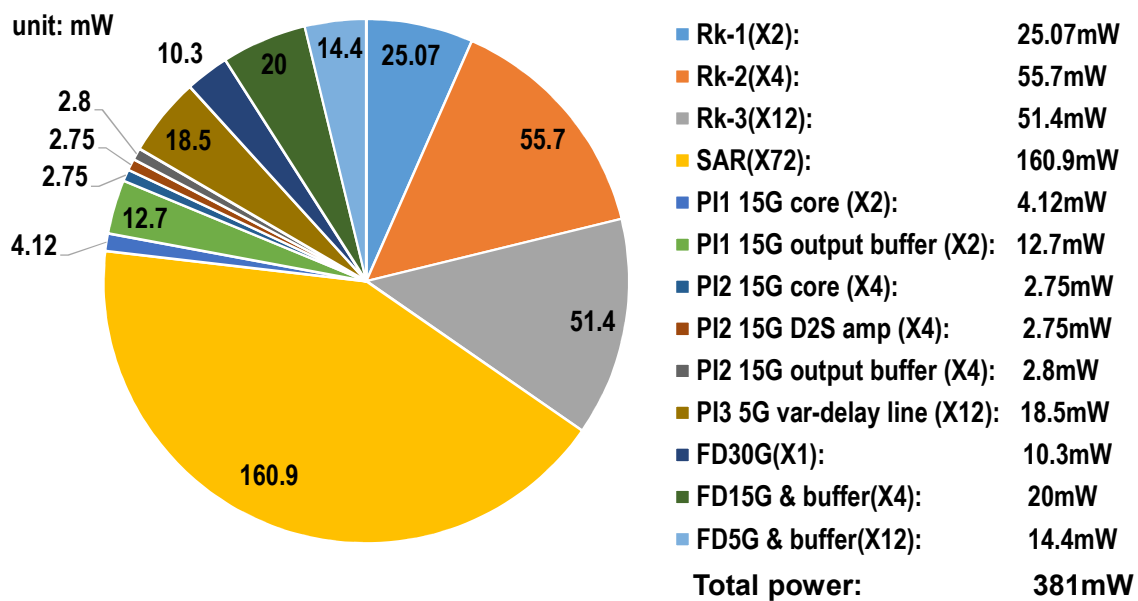


Figure 4.16 Detail power breakdown of the ADC chip

Table I Comparison table

		[1]	[2]	[3]	[4]	This work
Technology		40nm bulk CMOS	65nm Bulk CMOS	32nm PD-SOI CMOS	32nm PD-SOI CMOS	28nm FD-SOI CMOS
Sampling rate		40GS/s	56GS/s	68GS/s	70GS/s**	46GS/s
BW		14/18*GHz	16GHz	10GHz	<20GHz	>23GHz
SNDR	@ low freq.	-----	-----	-----	37.7dB	27dB
	@ Nyquist (or BW)	25.2dB	-----	36.7dB	34.2dB	25.2dB
Diff. input swing		1.2V	-----	-----	0.7V	0.4V
Supply voltage		1V/1.2V	-----	-----	1V/1.1V***	1.05V/1.6V
Power		1.5W	1.2W	2.1W	355mW	381mW
FOM @ Nyquist (pJ/conv.-step)		2.5	-----	0.55	0.12	0.56
Sparkle-code error-rate		-----	-----	-----	-----	<1e-10

* calibration performed at each test frequency

** numbers taken for lowest FOM

*** the nominal supply is 0.9V in this process

Chapter 5

Conclusion

5.1 Thesis summary

The recent explosive growth on 100/400Gbps fiber optics links has created high demand for embedded ultra-high-speed ADCs. These embedded ADCs are the key components to enable communications with high degree of modulation, and thus push the bit-rate beyond the bandwidth of the optical components. Although recent advances in ADCs at this speed has demonstrated massively time-interleaved SAR ADC is able to achieve $>40\text{GS/s}$, their input bandwidth is limited to much less 20GHz . This is because the sampling capacitor is driven by the series of sampling switch and output impedance of the driver circuits – switch resistance penalty. To alleviate this issue, a cascode-based T&H circuit is proposed in this thesis, in which the active cascode transistor of a common source amplifier replaces the switch in conventional T&H circuit. The Cascode-based T&H circuit has shown $>4\text{X}$ improvement in the bandwidth, but with the trade-off on linearity. To improve its linearity, a saturation NMOS load is employed to invert the nonlinear trans-conductance of the differential pair.

Although hierarchically time-interleaved sampling network has gained popularity in recent high-speed ADC designs, this architecture has not been systematically studied. This thesis presents a thorough investigation of the pros and cons for such structure. The design of a general hierarchically time-interleaved sampler network is systematically studied, and a power optimization method in the context of cascode-based T&H circuit is provided.

Despite the higher conversion speed and lower power consumption of asynchronous SAR sub-ADCs, the fear of sparkle-code errors have prevented their wide-spread use in some applications. This work thus provides a detailed statistical analysis on the error-rate of the meta-stability caused sparkle-codes in classic asynchronous SAR ADCs. It demonstrates the sparkle-code error-rate decreases exponentially with conversion time. Upon careful investigation, the attempts to correct meta-stability by “detect-then-stop” in recent works are shown to be ineffective to significantly reduce sparkle-codes error-rate. To properly address the issue, a back-end meta-stability correction method is proposed. The extensive statistical analysis shows the sparkle-code error-rate after correction falls with conversion time following the shape of a complimentary error function, thus significantly reduces the sparkle-code error-rate.

Finally, to demonstrate the proposed circuits and techniques, a 6-bit 46GS/s 3-rank 72X hierarchically time-interleaved asynchronous SAR ADC is fabricated in 28nm FDSOI process. The cascode-based T&H circuit enabled the prototype to achieve >23GHz input BW, the highest BW among >40GS/s state-of-art monolithic ADCs. The design uses the optimization method developed in this thesis to achieve total power consumption of 381mW. The FOM is 0.56pJ/conversion-step. The Back-end meta-stability correction technique successfully kept the ADC sparkle-code free for more than $1e10$ samples collected.

5.2 Future work

In the near future, continuing strong growth of data-rate in the fiber links is expected. The advancement of internet-of-things and cloud-computing are expected to generate even more data traffic, putting high demands on ADCs with even faster speed, higher bandwidth, and higher resolution. Since the f_T of the advanced technology nodes has not been improving with scaling, it becomes more and more difficult to push the input bandwidth using time-interleaving alone. On the other hand, improving the resolution (or ENOB) of the ADC can enable higher degree of modulation, allowing more bits to be communicated within the same bandwidth. Therefore, an interesting future direction is to improve ADC ENOB either by clever analog design or by more complex digital calibration. To push the BW beyond the limit of time-interleaved ADCs, 2 interesting approaches seem to hold great promise. One is to precede the time-interleaved ADC with a frequency interleaved front-end. The frequency-interleaving takes advantages of distributed amplifier and broadband RF mixer designs to break signal band into 2 or more smaller chunks and bring them to baseband, thus alleviating the bandwidth requirement of subsequent ADCs [37] [38]. Another revolutionary approach is to take advantage of low noise phase-lock-laser and high quality factor photonic components to sample the analog input optically [39]. It can theoretically reduce sampling jitter and push the input bandwidth beyond the limit of electrical ADCs. The fields of frequency-interleaved sampling and photonic sampling are still young, and needless to say, a lot of researches need to be done to advance these fields.

Bibliography

- [1] Y. Greshishchev, J. Aguirre, M. Besson, R. Gibbins, D. Falt, P. Flemke, N. Ben-Hamida, D. Pollex, P. Schvan and S.-C. Wang, "A 40GS/s 6b ADC in 65nm CMOS," in *International Solid-State Circuits Conference (ISSCC)*, 2010.
- [2] I. Dedic, "56Gs/s ADC : Enabling 100GbE," in *Optical Fiber Communication (OFC)*, 2010.
- [3] Semtech, "Semtech Announces Ultra-High Speed ADC and DAC for Advanced Communication Systems," 14 March 2014. [Online]. Available: <http://www.semtech.com/Press-Releases/2014/Semtech-Announces-Ultra-High-Speed-ADC-and-DAC-for-Advanced-Communication-Systems.html>.
- [4] L. Kull, T. Toifl, M. Schmatz, P. A. Francese, C. menolfi, M. Braendli, M. Kossel, T. Morf, T. M. Andersen and Y. Leblebici, "A 90GS/s 8b 667mW 64× Interleaved SAR ADC in 32nm Digital SOI CMOS," in *International Solid-State Circuits Conference (ISSCC)*, 2014.
- [5] M. El-Chammas and B. Murmann, "A 12-GS/s 81-mW 5-bit Time-Interleaved Flash ADC With Background Timing Skew Calibration," *Journal of Solid State Circuits (JSSC)*, vol. 46, no. 4, pp. 838-847, 2011.
- [6] K. Poulton, R. Neff, B. Setterberg, B. Wuppermann, T. Kopley, R. Jewett, J. Pernillo, C. Tan and A. Montijo, "A 20 GS/s 8 b ADC with a 1 MB memory in 0.18 μm CMOS," in *International Solid-State Circuits Conference (ISSCC)*, 2003.
- [7] B. Setterberg, K. Poulton, S. Ray, D. Huber, V. Abramzon, G. Steinbach, J. Keane, B. Wuppermann, M. Clayson, M. Martin, R. Pasha, E. Peeters, A. Jacobs, F. Demarsin, A. Al-Adnani and P. Brandt, "A 14b 2.5GS/s 8-way-interleaved pipelined ADC with background calibration and digital dynamic linearity correction," in *International Solid-State Circuit Conference (ISSCC)*, 2013.
- [8] J. Gorecki, "Dynamic input sampling switch for CDACS". US Patent 5084634 A, 24 October 1990.
- [9] A. Abo and P. Gray, "A 1.5-V, 10-bit, 14.3-MS/s CMOS pipeline analog-to-digital converter," *Journal of Solid State Circuits*, vol. 34, no. 5, pp. 599-606, 1999.
- [10] Y. Duan and E. Alon, "A 12.8GS/s time-interleaved SAR ADC with 25GHz 3dB ERBW and 4.6b ENOB," in *Custom Integrated Circuits Conference (CICC)*, 2013.
- [11] Y. Duan and E. Alon, "A 12.8 GS/s Time-Interleaved ADC With 25 GHz Effective Resolution Bandwidth and 4.6 ENOB," *Journal of Solid State Circuits*, vol. 49, no. 8, pp. 1725-1738, 2014.
- [12] G. Wegmann, E. Vittoz and F. Rahali, "Charge injection in analog MOS switches," *Journal of Solid State Circuits*, vol. 22, no. 6, pp. 1091-1097, 1987.

- [13] L. Dai and R. Harjani, "CMOS switched-op-amp-based sample-and-hold circuit," *Journal of Solid State Circuits*, vol. 35, no. 1, pp. 109-113, 2000.
- [14] K. Doris, E. Janssen, C. Nani, A. Zanicopoulos and G. van der Weide, "A 480 mW 2.6 GS/s 10b Time-Interleaved ADC With 48.5 dB SNDR up to Nyquist in 65 nm CMOS," *Journal of Solid State Circuits*, vol. 46, no. 12, pp. 2821-2833, 2011.
- [15] M. El-Chammas, X. Li, S. Kimura, J. Coulon, J. Hu, D. Smith, P. Landman and M. Weaver, "15.8 90dB-SFDR 14b 500MS/S BiCMOS switched-current pipelined ADC," in *International Solid-State Circuits Conference (ISSCC)*, 2015.
- [16] E. Janssen, K. Doris, A. Zanicopoulos, A. Murrioni, G. van der Weide, Y. Lin, L. Alvado, F. Darthenay and Y. Fregeais, "An 11b 3.6GS/s time-interleaved SAR ADC in 65nm CMOS," in *International Solid State Circuits Conference (ISSCC)*, 2013.
- [17] M. El-Chammas, X. Li, S. Kimura, K. Maclean, J. Hu, M. Weaver, M. Gindlesperger, S. Kaylor, R. Payne, C. Sestok and W. Bright, "A 12 Bit 1.6 GS/s BiCMOS 2x2 Hierarchical Time-Interleaved Pipeline ADC," *Journal of Solid State Circuits*, vol. 49, no. 9, pp. 1876-1885, 2014.
- [18] S. Gupta, M. A. Inerfield and W. J., "A 1-GS/s 11-bit ADC With 55-dB SNDR, 250-mW Power Realized by a High Bandwidth Scalable Time-Interleaved Architecture," *Journal of Solid State Circuits*, vol. 41, no. 12, pp. 2650-2657, 2006.
- [19] S. Le Tual, P. Singh, C. Curis and P. Dautriche, "A 20GHz-BW 6b 10GS/s 32mW time-interleaved SAR ADC with Master T&H in 28nm UTBB FDSOI technology," in *International Solid-State Circuits Conference (ISSCC)*, 2014.
- [20] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.
- [21] J. Pernillo and F. M.P., "A 1.5-GS/s Flash ADC With 57.7-dB SFDR and 6.4-Bit ENOB in 90 nm Digital CMOS," *Transactions on Circuits and Systems II (TCAS-II)*, vol. 58, no. 12, pp. 837-841, 2011.
- [22] J. Pernillo and M. Flynn, "A 9b 2GS/s 45mW 2X-interleaved ADC," in *European Solid-State Circuits Conference (ESSCIRC)*, 2013.
- [23] V.-C. Chen and L. Pileggi, "A 69.5mW 20GS/s 6b time-interleaved ADC with embedded time-to-digital calibration in 32nm CMOS SOI," in *International Solid-State Circuit Conference (ISSCC)*, 2014.
- [24] B. Hershberg and U.-K. Moon, "Ring Amplifiers for Switched Capacitor Circuits," *Journal of Solid State Circuits*, vol. 47, no. 12, pp. 2928-2942, 2012.
- [25] B. Hershberg and U.-K. Moon, "A 75.9dB-SNDR 2.96mW 29fJ/conv-step ringamp-only pipelined ADC," in *Symposium on VLSI Circuits (VLSIC)*, 2013.
- [26] Y. Lim and M. Flynn, "11.5 A 100MS/s 10.5b 2.46mW comparator-less pipeline ADC using self-biased ring amplifiers," in *International Solid-State Circuits Conference (ISSCC)*, 2013.

- [27] V. Abramzon, "Analog-to-digital converters for high-speed links," Thesis. Stanford University.
- [28] L. Kull, T. Toifl, M. Schmatz, P. Francese, C. Menolfi, M. Brandli, M. Kossel, T. Morf, T. Andersen and Y. Leblebici, "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," *Journal of Solid State Circuits Conference*, vol. 48, no. 12, pp. 3049-3058, 2013.
- [29] V. Tripathi and B. Murmann, "Mismatch Characterization of Small Metal Fringe Capacitors," *Transactions on Circuits and Systems I (TCAS-I)*, vol. 61, no. 8, pp. 2236-2242, 2014.
- [30] S.-W. Chen and R. Brodersen, "A 6-bit 600-MS/s 5.3-mW Asynchronous ADC in 0.13-um CMOS," *Journal of Solid State Circuits (JSSC)*, vol. 41, no. 12, pp. 2669-2680, 2006.
- [31] S.-H. Cho, C.-K. Lee, S.-G. Lee and S.-T. Ryu, "A Two-Channel Asynchronous SAR ADC With Metastable-Then-Set Algorithm," *Transactions on Very Large Scale Integration (TVLSI)*, vol. 20, no. 4, pp. 765-769, 2011.
- [32] J.-W. Nam, D. Chiong and M.-W. Chen, "A 95-MS/s 11-bit 1.36-mW asynchronous SAR ADC with embedded passive gain in 65nm CMOS," in *Custom Integrated Circuits Conference (CICC)*, 2013.
- [33] T. Cho and G. P.R., "A 10 b, 20 Msample/s, 35 mW pipeline A/D converter," *Journal of Solid State Circuits*, vol. 30, no. 12, pp. 166-172, 1995.
- [34] S. Sidiropoulos and M. Horowitz, "High Performance Inter-Chip Signalling," Thesis. Stanford University, 1998.
- [35] C. Thakkar, L. Kong, K. Jung, A. Frappe and E. Alon, "A 10 Gb/s 45 mW Adaptive 60 GHz Baseband in 65 nm CMOS," *Journal of Solid State Circuits*, vol. 47, no. 4, pp. 952-968, 2012.
- [36] M. van Elzakker, E. van Tuijl, P. Geraedts, D. Schinkel, E. Klumperink and B. Nauta, "A 1.9 μ W 4.4fJ/Conversion-step 10b 1MS/s Charge-Redistribution ADC," in *International Solid State Conference (ISSCC)*, 2008.
- [37] Y. Greshishchev, "Embedded CMOS ADCs for Optical Communications," *Forum presentation. International Solid-State Circuits Conference*, 2015.
- [38] S. Callender, "Wideband Signal Acquisition via Frequency-Interleaved Sampling," Thesis. UC Berkeley, 2015.
- [39] A. Khilo, S. Spector, M. Grein, A. Nejadmalayeri and et. al., "Photonic ADC: overcoming the bottleneck of electronic jitter," *Optics Express*, vol. 20, no. 4, pp. 4454-4469, 2012.