

Low-Power High-Performance SAR ADC with Redundancy and Digital Background Calibration

by

Albert Hsu Ting Chang

B.S., Electrical Engineering and Computer Science,
University of California, Berkeley (2007)

S.M., Electrical Engineering and Computer Science,
Massachusetts Institute of Technology (2009)

Submitted to the

Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2013

© Massachusetts Institute of Technology 2013. All rights reserved.

Author
Department of Electrical Engineering and Computer Science
May 22, 2013

Certified by
Duane S. Boning
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Certified by
Hae-Seung Lee
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by
Leslie A. Kolodziejki
Chairman, Department Committee on Graduate Theses

Low-Power High-Performance SAR ADC with Redundancy and Digital Background Calibration

by

Albert Hsu Ting Chang

Submitted to the Department of Electrical Engineering and Computer Science
on May 22, 2013, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

As technology scales, the improved speed and energy efficiency make the successive-approximation-register (SAR) architecture an attractive alternative for applications that require high-speed and high-accuracy analog-to-digital converters (ADCs). In SAR ADCs, the key linearity and speed limiting factors are capacitor mismatch and incomplete digital-to-analog converter (DAC)/reference voltage settling. In this thesis, a sub-radix-2 SAR ADC is presented with several new contributions. The main contributions include investigation of using digital error correction (redundancy) in SAR ADCs for dynamic error correction and speed improvement, development of two new calibration algorithms to digitally correct for manufacturing mismatches, design of new architecture to incorporate redundancy within the architecture itself while achieving 94% better energy efficiency compared to conventional switching algorithm, development of a new capacitor DAC structure to improve the SNR by four times with improved matching, joint design of the analog and digital circuits to create an asynchronous platform in order to reach the targeted performance, and analysis of key circuit blocks to enable the design to meet noise, power and timing requirements.

The design is fabricated in standard 1P9M 65nm CMOS technology with 1.2V supply. The active die area is 0.083mm² with full rail-to-rail input swing of 2.4V_{P-P}. A 67.4dB SNDR, 78.1dB SFDR, +1.0/-0.9 LSB₁₂ INL and +0.5/-0.7 LSB₁₂ DNL are achieved at 50MS/s at Nyquist rate. The total power consumption, including the estimated calibration and reference power, is 2.1mW, corresponding to 21.9fJ/conv.-step *FoM*. This ADC achieves the best *FoM* of any ADCs with greater than 10b ENOB and 10MS/s sampling rate.

Thesis Supervisor: Duane S. Boning

Title: Professor of Electrical Engineering and Computer Science

Thesis Supervisor: Hae-Seung Lee

Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

I am ever so grateful that after six years I am able to complete my PhD degree here at MIT. I have been blessed to be surrounded by family, friends, professors, office-mates and colleagues who have provided cheers and support throughout my PhD. I am indebted to all of them. Without their constant support, this would have never been possible.

I would like to thank my research advisors, Professor Duane Boning and Professor Hae-Seung (Harry) Lee. Both of them have provided me not only technical and research guidance, but also practical advices in my career. I was given a lot of freedom to explore research areas that excited me, and they have always been respectful of the decisions I have made. I will always remember that during our meetings, they would challenge me to push the boundaries of my research further when I strongly believed it wasn't possible. It was only later that I found out that this process helped me become more critical in thinking about a research problem, and only through this process could innovation occur. I could not have asked for better advisors and I only hope that throughout my career, I will be able to pass along a fraction of the wisdom and learning that they had taught me over the years. I would also like to thank my thesis committee member, Professor Anantha Chandrakasan. Even though the meeting time was limited, he always pointed me towards the right direction and helped me put my research into perspective.

I would like to thank Professor Dejan Markovic for giving me the opportunity to participate in his research project at the Berkeley Wireless Research Center (BWRC). He guided me through my first research project and taught me how research is conducted at the professional level. I would also like to thank Dr. Simone Gambini and Professor Jan Rabaey at the BWRC. Even though I was only an undergraduate, they trained me to conduct research at the graduate level and inspired me to look for my own solutions rather than just implementing a solution that other graduate students had developed.

I would like to thank Professor Terry Orlando and Dr. Ann Orlando for being

great mentors of mine. They have been a source of inspiration, support and love when I was off-campus and away from work. I had the privilege to work with them when I was the chair of the Ashdown House Executive Committee (AHEC). They taught me how to become a better decision maker and a socially responsible leader. When I was struggling with personal issues, they provided me with needed comforts and care. I was invited to their apartment multiple times and they always made me feel welcomed even though, sometimes, I felt that I was merely repeating myself. Thank you for your kindness and patience.

I would like to thank my mentors from church: Patrick Lin, Mike Chen, Susan Chen, Lan-Hui Chou, Cherilyn Hu, Chun-Yi Hu, Erika Cheng and Shin-Jong Chung. Patrick, thank you for always being there for me. It is difficult to express all my gratitude to you in words. You have helped me through the toughest times I have experienced in my life. No matter what other people thought of me, you always knew my true intentions and always believed in me. Your continuous encouragement really helped transform my life and matured me spiritually. Thank you for being patient and kind with all my long phone calls. Mike and Susan, thank you for being so loving and supportive. Even with your busy schedule, you would squeeze out an entire night to listen and to share your personal experience with me. You were always able to cheer me up during the bad days and shared my laughs during the good ones. I only hope that I can capture and learn a fraction of the life wisdom that you've taught me. Lan-Hui, thank you for accompanying me through my difficult times. Your frankness helped me see myself from a different perspective and your sharing always touched my heart. Thank you for all the great food that you cooked for me. Cherilyn and Chun-Yi, thank you for treating me as your own son. From your teaching, I can sense that you genuinely cared about me, my well-being and my personal development. I am very fortunate to have you as someone to look up to. Erika and Shin-Jong, thank you for spending time to share and to pray with me.

There are many friends at MIT who I want to thank for their love and support: Samuel Chang, Tsung-Yu Kao, Wen-Hsuan Lee, Kay Hsi, Tsung-Hsiang Chang, Chia-ying Lee, JianKang Wang, Dawson Hwang, Wei-Shan Chiang, Keng-Yen Chiang,

William Leight, Amanda Zangari and Dheera Venkatraman. Samuel, I am blessed to have you as my friend and thank you for being there for me when I needed you the most. You walked into my life when the rest of the world seemed to walk out. You sacrificed a lot of your own time to listen or just to keep me company during difficult times. I hope in the coming future, we will continue to develop our cherished friendship. Kao Tsung, thank you for being like a big brother to me. You are the most knowledgeable source I have regarding research or life in general. I really enjoyed many of our late-night discussion on various aspects of life. Samuel, Kao Tsung, Wen-Hsuan, Kay, Tsung-Hsiang and Chia-Ying, I will always remember the potlucks that we used to have at Ashdown. JianKang, thank you for the kind words and encouragement when I was applying for work. Dawson, Wei-Shan and Keng-Yen, thank you for all the sharing at lunch or dinner. William and Amanda, thank you for being awesome and considerate roommates for the past three years. Dheera, thank you for sharing many of your interesting stories and being supportive when we were on AHEC together.

I would like to thank my friends from church: Shu-i Hsiung, Jonathan Leu, Duane Chang, Peter Wang, Chia-Yu Chen, Jane Wang, Jennifer Cheng, Ying Tang, Reby Lin and Hsun-An Yu. Shu-i, thank you for the love, the encouragement and the unwavering support you have shown me in the past. You truly touched and transformed my life. I am blessed that I was able to spend some of the happiest moments of my life with you. You taught me how to truly love someone as described in Corinthians 13:4-8. I cannot thank you enough. Jonathan, Duane, Peter and Chia-Yu, thank you for being a great source of support and wonderful brothers in church. Jane, Jennifer, Ying, Reby and Hsun-An, thank you for offering care, support and kind words when I needed them.

I am grateful to all the past and present members of both the Boning and HSLee research groups, with whom I've shared many interesting conversations and who have helped me in many capacities. I would like to acknowledge John Lee, Li Yu, Joy Johnson, Wei Fan, Sunghyuk Lee, Ayman Shabra, Daniel Kumar, Payam Lajevardi, Marianan Markova, Xi Yang, Do Yeon Yoon, Miguel Perez, Sabino Pietrangelo, Jack

Chu, Philip Godoy, Sung-Won Chung and Hyun Boo for their contributions to my technical understanding and/or my well-being over the past six years. John, Li, Joy and Wei, thank you for being such excellent office-mates, with whom I can share not just technical problems but also life stories. I will definitely miss our weekly happy hour together. In addition, I would like to thank Grace Lindsay, Mira Whiting and Carolyn Collins for all their administrative help and Debb Hodges-Pabon for making my life at MTL so much better and more enjoyable.

Finally, I would like to thank my family, especially my parents and my brother: Ben Chang, I-Chen Cheng and Arthur Chang. You have continued to show unconditional love and support throughout my life, no less so than during my PhD. You made a lot of sacrifices for me and Arthur to come to the States to get a better education. I know how proud you are of me and I cannot thank you enough for all the unconditional love you have always provided and continue to provide. Your steadfast confidence and belief in me have been the biggest inspiration for me to move forward. Thank you! Arthur, thank you for being a very supportive and a wonderful brother. I know I can always go to you for motivation and comforts. Uncle Rong, thank you for helping me with various crucial decisions I had to make in life, including job offers and a decision to come to MIT. Yea-Lin and Pao-Shen, thank you for being amazing grandparents who have warmly welcomed me whenever I go back home.

The authors would like to acknowledge the Cooperative Agreement between the Masdar Institute of Science and Technology (Masdar Institute) and the Massachusetts Institute of Technology (MIT) for funding and Taiwan Semiconductor Manufacturing Company (TSMC) for chip fabrication.

Contents

1	Introduction	25
1.1	Challenges of Technology Scaling	26
1.2	ADC Architecture Overview	28
1.3	Trend of Analog-to-Digital Converters	30
1.4	Thesis Contributions	38
2	Approaches and Challenges in Traditional SAR ADCs	41
2.1	Search Algorithms for Nyquist-rate ADCs	42
2.1.1	Flash Algorithm	43
2.1.2	Binary Successive Approximation Algorithm	45
2.1.3	Pipeline Algorithm	46
2.1.4	Summary of the Search Algorithms	48
2.2	The SAR Architecture	49
2.3	Static Error Sources in SAR ADCs	53
2.3.1	Capacitor Mismatches	55
2.3.2	Offset Errors	56
2.4	Dynamic Error Sources in SAR ADCs	58
3	Redundancy in SAR ADCs	61
3.1	Redundancy Overview	62
3.1.1	Error Tolerance Windows for Redundancy	65
3.2	Digital Calibratability	67
3.2.1	Condition of Digital Calibratability	67

3.2.2	Amount of Redundancy	68
3.2.3	Radix and the Number of Steps	69
3.3	Redundancy and its Speed Benefit	73
3.3.1	Prior Work	74
3.3.2	Behavioral Models	74
3.3.3	Effectiveness of Redundancy	77
4	Digital Background Calibration of SAR ADCs	85
4.1	Missing Codes in Code Density Histogram	88
4.2	Calibration Algorithm I	92
4.3	Calibration Algorithm II	98
4.3.1	Choice of Calibration Signal	99
4.3.2	Calibration using a Sine Wave	101
4.4	Calibration Algorithm III	107
4.4.1	Integer Step Sizes Extraction	107
4.4.2	Fractional Step Sizes Extraction	111
4.4.3	Unknown Input Statistics	116
4.4.4	Calibration Examples	117
4.4.5	Comparisons of the Calibration Algorithms	118
5	Design and Implementation of a SAR ADC with Redundancy	123
5.1	Architecture	124
5.1.1	Energy Consumption in Switching Scheme	125
5.1.2	Main-Sub-DAC Array	141
5.1.3	Redundancy Implementation	146
5.1.4	The Overall Architecture	149
5.2	Key Circuit Building Blocks	152
5.2.1	Latch Comparator	153
5.2.2	Sampling Circuit	159
5.2.3	Pulse Generator	165
5.2.4	Capacitive DAC array	168

5.2.5	Kickback Noise	172
5.3	Summary	174
6	Packaging, Test Setup and Measurement Results	177
6.1	Packaging	177
6.2	Test Setup	183
6.3	Measurement Results	184
7	Conclusion and Future Work	191
7.1	Conclusion	191
7.2	Future Work	193

List of Figures

1-1	A plot of the resolution versus the input sampling frequency for recent published analog-to-digital converters in ISSCC and VLSI (data adopted from [1]).	28
1-2	The Schreier Figure of Merit (FoM_3) versus CMOS process nodes from $1\mu\text{m}$ to 28nm of state-of-the-art ADCs published at ISSCC and VLSI Symposium (data adopted from [1]). Even though technology scaling does not directly benefit analog integrated circuits, steady improvement in conversion energy efficiency, FoM_3 , is still shown. This trend is the result of using more digital friendly architectures in recent designs.	32
1-3	The conversion energy efficiency FoM_3 from year 1997-2012 (data adopted from [1]). This trend emphasizes the importance of energy efficiency in recent designs.	33
1-4	Walden's FoM versus sampling frequency of state-of-the-art ADCs published at ISSCC and VLSI Symposium (data adopted from [1]). .	34
1-5	Walden's FoM versus resolutions of state-of-the-art ADCs published at ISSCC and VLSI Symposium (data adopted from [1]).	35
1-6	Schreier's FoM versus sampling frequency of state-of-the-art ADCs published at ISSCC and VLSI Symposium (data adopted from [1]). The plot shows the architecture front, the technology front, and the FoM_3 corner. These terminologies are introduced by Schreier and Temes in [2].	36

1-7	Energy per Nyquist sample versus SNDR of state-of-the-art ADCs published at ISSCC and VLSI Symposium (data adopted from [1]). The plot indicates the FoM at $100fJ/\text{conv.-step}$ and $10fJ/\text{conv.-step}$ with black dotted lines, and the red line denotes the “architecture front” of high-accuracy ADCs that are noise limited.	37
2-1	An example of 5-bit quantization using “brute-force” direct search. It is done by directly comparing the analog input with all $2^N - 1$ decision levels.	43
2-2	Basic architecture of a flash converter. The resistor ladder generates the needed decision levels and the following comparators generate thermometer output codes that represent the limit in which the input is greater than one of the decision levels. The decoder then converts the thermometer output codes into binary-weighted output bits. The number of the required comparators scales exponentially with the number of bits.	44
2-3	An example of 5-bit quantization using a binary search algorithm. Instead of using just one clock cycle per conversion, it requires five clock cycles to complete the conversion process and to realize the final 5-bit output.	45
2-4	The transfer function of each multiply-by-two pipeline stage for the second type of binary search algorithm.	46
2-5	The block diagram for a traditional pipeline ADC. By cascading N 1-bit stages together, the ADC is able to produce N -bit resolution outputs.	47
2-6	An example of a 5-bit quantization using the second type of binary search.	48
2-7	Basic block diagram of a SAR ADC. It includes a S&H, a DAC, a comparator and a SAR control logic block.	49
2-8	Schematics of the charge redistribution SAR implementation.	50

2-9	Switching scheme of a conventional SAR ADC.	52
2-10	An example ADC transfer function for SAR ADCs with/without capacitor mismatches.	56
2-11	Effective number of bits (ENOB) versus normalized capacitor mismatch σ_{C_u}/C_u in a 12-bit binary weighted SAR ADC. The plot shows that 1% unit capacitor mismatch can sometimes lead to 1b loss in ENOB.	57
2-12	Schematic of a SAR ADC with offset errors.	58
3-1	Binary search algorithm without redundancy. The search step sizes in this example are binary weighted with values equal to 8, 4, 2 and 1.	63
3-2	Comparison of using a traditional binary search algorithm (4-bit 4-step) and a sub-binary search algorithm (4-bit 6-step). Black decision levels indicate that in each step, transitions to the nearest decision levels above and below the current step level are possible.	63
3-3	Digital error correction using redundancy in SAR ADCs. We've seen that even though the digital output bits are different in all three cases, they all represent the same D_{out}	64
3-4	Highlighted error tolerance windows (ϵ_t) for a sub-binary search SAR ADC. The error tolerance windows are as follows: $\epsilon_t(5) = \pm 3$, $\epsilon_t(4) = \pm 1$, $\epsilon_t(3) = 0$, $\epsilon_t(2) = 0$ and $\epsilon_t(1) = 0$	66
3-5	Transfer functions for SAR designs with step sizes that are binary, sub-radix-2 and super-radix-2 weighted.	67
3-6	Effective number of bits (N) versus number of steps (M) for different radices (α). Converters with smaller α require additional conversion steps to achieve the same effective resolution, but they have more built-in redundancy against dynamic and static conversion errors.	70
3-7	The maximum radix α and the minimum number of conversion steps M versus the standard deviation of the unit capacitor, in order to achieve digital calibratability in a 12-bit ADC.	72

3-8	Behavioral model of a SAR ADC. The critical delay path is divided into three components: the latch delay (T_C), the logic delay (T_L) and the DAC settling delay (T_D).	75
3-9	Behavioral model when the i^{th} capacitor in the DAC is being charged or discharged.	75
3-10	The effectiveness of redundancy in SAR ADCs when the delay through the DAC array (T_D) dominates.	78
3-11	The effectiveness of redundancy in SAR ADCs when the delay through the latch (T_C) dominates the other delay components.	79
3-12	The number of metastability events when the delay through the latch (T_C) dominates the other delay components.	80
3-13	Effectiveness of redundancy in SAR ADCs (SPICE). The results show that the fastest clock period that a non-redundant SAR ADC can run is $500ps$ while the fastest clock period that a redundant SAR ADC can run is $320ps$.	81
3-14	Effectiveness of redundancy in SAR ADCs using behavioral model simulation.	81
3-15	Effectiveness of redundancy in SAR ADCs using behavioral model simulation. This figure is a one-dimensional slice taken at $\tau = 40ps$ from Figure 3-14.	82
3-16	Speed improvement from adopting redundancy. The left plot shows per-cycle speed improvement and the right plot shows the overall speed improvement of the ADC.	83
3-17	Effectiveness of redundancy in SAR ADCs with an added pre-amplifier in front of the latch comparator in SPICE simulation.	83
4-1	Normalized code density histogram with missing codes: 3, 4, 6, 7, 8, 9, 11 and 12. The histogram is generated with a linear input ramp over the full scale. The missing codes are the result of redundancy.	90

4-2	Normalized code density histogram with fractional capacitor values. The histogram is generated with a linear input ramp over the full scale.	91
4-3	Tree representation of a sub-binary search. The diagram shows how a decision level is reached from a previous decision level.	93
4-4	Tree representation of a sub-binary search with marked decision levels.	94
4-5	A sub-binary search tree with highlighted regions R_C , indicating the input range corresponds to code C .	96
4-6	Spectrum data before and after calibration scheme I. The effective number of bits (ENOB) improves from 8.18b to 11.35b.	97
4-7	Statistics of a sinusoid signal. The sinusoid signal is assumed to have amplitude A and offset voltage V_0 .	100
4-8	Using the “bounded regions” to extract the actual step sizes. The bounded regions are highlighted in black. This calibration scheme uses the statistics of the input signals rather than relying on the exact knowledge of the input signals as in the case of the first calibration algorithm.	104
4-9	Spectrum data before and after using the statistical calibration algo- rithm. ENOB improves from 8.18b to 11.35b and SFDR improves from 60.71dB to 87.01dB.	105
4-10	Calculating D_{out} when all the step sizes are integer multiples of each other. In this case, $D_{out} = F_{out}$.	108
4-11	Calculating D_{out} when step sizes are fractional. Since not all the code bins have 1 LSB bin width, there is static nonlinearity in this ADC.	112
4-12	Static nonlinearity before and after using the third calibration algo- rithm. The DNL improves from +2.53/-1.0 LSB_{12} to +0.56/-0.56 LSB_{12} ; the INL improves from +7.8/-7.9 to +0.6/-0.6 LSB_{12} .	119
4-13	Spectrum data before and after using the third calibration algorithm. The ENOB improves from 8.6b to 11.6b; and the SFDR improves from 59.5dB to 92.0dB.	120

5-1	Energy consumption when charges change on capacitor C_A	125
5-2	Conventional SAR switching algorithm, showing energy consumption related to capacitor switching transitions.	127
5-3	The top-plate waveform when using the conventional switching algorithm. The input is assumed to have magnitude equal to 0.9 with $V_{IN+} = 0.95$, $V_{IN-} = 0.05$ and $V_{REF} = 1.0$. The final output bit sequence is 111100.	127
5-4	Split-capacitor switching algorithm, showing reduced energy consumption compared to Figure 5-2.	129
5-5	Energy-saving switching algorithm, showing reduced energy consumption compared to Figures 5-2 and 5-4.	131
5-6	The top-plate waveform when using the energy-saving switching algorithm. The input is assumed to have magnitude equal to 0.9 with $V_{IN+} = 0.95$, $V_{IN-} = 0.05$ and $V_{REF} = 1.0$. The final output bit sequence is 111100. Comparing the top-plate waveform in a conventional algorithm and in a energy-saving algorithm, they are differed in that in a conventional switching algorithm, the top-plate voltage begins with V_{CM} , but in the energy-switching algorithm, the top-plate voltage begins with V_{REF}	132
5-7	Monotonic switching algorithm, showing reduced energy consumption compared to Figures 5-2, 5-4 and 5-5.	134
5-8	The top-plate waveform when using the monotonic switching algorithm. The input is assumed to have magnitude equal to 0.9 with $V_{IN+} = 0.95$, $V_{IN-} = 0.05$ and $V_{REF} = 1.0$. The final output bit sequence is 111100. Rather than converging towards V_{CM} at the end of the conversion progress, the top plate voltages of the upper/lower DACs both converge to ground.	135
5-9	Merged capacitor switching algorithm, showing reduced energy consumption compared to Figures 5-2, 5-4, 5-5 and 5-7.	136

5-10	The top-plate waveform when using the MCS algorithm. The input is assumed to have magnitude equal to 0.9 with $V_{IN+} = 0.95$, $V_{IN-} = 0.05$ and $V_{REF} = 1.0$. The final output bit sequence is 111100. This switching scheme requires an additional reference voltage V_{CM} compared to previous switching algorithm.	137
5-11	Inverted merged capacitor switching (IMCS) algorithm, achieving the same energy efficiency as the MCS algorithm. It inverts the first charging sequences such that the conversion accuracy is not affected by the parasitic capacitance on the top plates of the DAC.	138
5-12	The top-plate waveform when using the IMCS algorithm. The input is assumed to have magnitude equal to 0.9 with $V_{IN+} = 0.95$, $V_{IN-} = 0.05$ and $V_{REF} = 1.0$. The final output bit sequence is 111100. This switching algorithm achieves the same energy efficiency as the MCS algorithm, but the accuracy of the IMCS algorithm is not sensitive to parasitic capacitances on the top plates of the DAC.	139
5-13	Configuration to consider the effect of parasitic capacitance on IMCS algorithm.	139
5-14	Comparing energy consumption of different switching algorithms (conventional, split-cap, energy saving, monotonic and MCS/IMCS.) . . .	140
5-15	Comparison of different switching schemes in terms of various figures of merit. The IMCS algorithm is able to achieve the best figure of merit across the board.	141
5-16	An 8-bit example of using the main-sub-dac array architecture. Using the main-sub-dac array architecture the total capacitance can be reduced from $128C$ to $24C$	142
5-17	A more general representation of the main-sub-dac array architecture. The LSB DAC has a total of L -bit resolution and the MSB DAC has a total of M -bit resolution. The bridge capacitor C_B is a fractional value.	142
5-18	New main-sub-dac array architecture. This new architecture resolves the matching and over-range problem together.	145

5-19	Redundancy implementation using conventional switching algorithm. It is done by directly sizing the capacitors proportional to the desired searching step sizes.	147
5-20	Redundancy implementation using IMCS algorithm. It can be done by directly sizing the capacitors proportional to the desired searching step sizes, while still maintaining a symmetric search window size. . .	148
5-21	Comparison of error tolerance windows (ϵ_t) between two redundancy implementations. Implementing redundancy using the IMCS algorithm allows symmetric search window size and symmetric tolerance to dynamic settling errors.	148
5-22	The overall architecture incorporating previous new architectural techniques. The ADC generates 16 raw output bits with four redundant decisions, making it a 12-bit effective resolution.	150
5-23	Timing waveform of the asynchronous SAR ADC using the inverted merged capacitor switching (IMCS) algorithm.	152
5-24	The design of StrongARM latch comparator. It consumes no static power during standby period and only dynamic current is present during regeneration.	153
5-25	Large signal transient response of the latch comparator. After clock signal goes high, the differential outputs begin to discharge together before one output starts moving to V_{DD} and the other output continues to discharge towards ground.	155
5-26	Input referred noise, power consumption and speed as a function of $\rho = \frac{W_1/L_1}{W_{clk}/L_{clk}}$	157
5-27	Product of noise power, power consumption and delay as a function of $\rho = \frac{W_1/L_1}{W_{clk}/L_{clk}}$. It shows that it is possible to optimize such product by properly ratioing the size of input pairs and the transistor M_{clk}	157
5-28	Simulation setup to extract the noise variance. The simulation is done in Cadence <i>SpectreRF</i> using transient noise analysis.	158

5-29	Bottom plate sampling circuit to help improve linearity. “1” represents at time 1 and “1d” represents a delay after time 1.	160
5-30	Difference in charge injection versus different input value.	161
5-31	Bootstrapped sampling switches. This circuit allows the gate voltage to track the source voltage to maintain a constant V_{GS} , regardless of what the input voltage is.	162
5-32	Simulation of the bootstrapped sampling circuit. The gate voltage is able to track the input voltage.	162
5-33	Comparison between the switch resistances. Even though the bootstrapped switch is not perfectly constant, its resistance is much flatter compared to switches made out of NMOS, PMOS or transmission gates.	163
5-34	Asynchronous pulse generator. A Schmitt trigger is added to avoid voltage spikes in dynamic operation and to improve the robustness against noise.	166
5-35	Timing diagram for the asynchronous pulse generation. By tuning the node VTUNE, the pulse width can be increased (or decreased) to slow down (or speed up) the asynchronous operation.	166
5-36	The pulse width in the fast and slow modes of operation. The slow mode is designed for debugging purposes.	168
5-37	A simple noise model for sampling circuits.	169
5-38	Reduction in ENOB due to thermal noise. Here, the thermal noise is in the unit of LSBs.	170
5-39	Capacitor layout for our DAC. Even though the capacitors are not binary weighted, common-centroid layout practice is still employed here to minimize mismatch.	171
5-40	Kickback noise generation. Unequal charges are injected onto the input nodes if the impedances looking back are different.	172
5-41	Array of bootstrapped switches to reduce the effect of kickback noise. All the switches share a common clock multiplier circuit.	173

6-1	Die micrograph of the fabricated chip in TSMC 65nm technology. . .	178
6-2	Separation between the analog and digital supplies can help improve isolation to reduce noise coupling. Our design uses approach (c) above.	180
6-3	Die bonding diagram, following the design principle described in Section 6.1.	182
6-4	ADC evaluation test setup. The setup includes a DC power supply, a signal generator, a clock generator, a logic analyzer, a FPGA and our PCB board.	185
6-5	Measured DNL and INL for 12-bit resolution with 1.2 supply at 50MS/s with 24.7MHz input sine wave.	187
6-6	Measured spectrum data for 12-bit resolution with 1.2V supply at 50MS/s with 24.7MHz input sine wave.	188
6-7	Measured SNDR and SFDR at different input frequencies and the summary of measurement result.	189
6-8	Comparison with the state-of-the-art (data adopted from [1]).	189

List of Tables

4.1	Relationship between the output bit combinations, decision level progressions, the location of missing codes and their corresponding D_{out} 's for integer step sizes. This shows why missing codes can occur in the output code density map when there is redundancy.	90
4.2	Relationship between the output bit combinations, decision level transitions, the location of missing codes and their corresponding D_{out} 's for fractional step sizes. It shows that some code bins can have different sizes.	92
4.3	Estimation of step sizes using the first calibration algorithm. Without adding circuit noise, this extraction procedure is able to extract the actual step sizes with high accuracy; with the addition of circuit noise, this extraction procedure begins to lose accuracy.	98
4.4	Estimation of step sizes using the statistical calibration algorithm. The accuracy of this extraction procedure is not affected by circuit noise. Due to the statistical nature of this calibration scheme, the extraction precision can be increased by collecting more samples.	105
4.5	Estimation of step sizes using the third calibration algorithm. The difference between the actual and the estimated step sizes are small, with largest difference equal to 0.15. This shows the effectiveness of the calibration algorithm even in the presence of circuit noise.	119
5.1	The minimum number of time constants needed for a first-order RC circuit to settle within half a LSB for an n -bit ADC.	164

6.1 Designed versus extracted capacitor values. Some capacitors show large discrepancy between the designed and the extracted values. These results confirm that calibration is necessary to achieve high resolution in a SAR ADC design. 184

Chapter 1

Introduction

Modern electronic systems store and process information in the digital domain. For these systems to interface with real-world analog signals, conversions between the analog and digital signals are required. As a result, one of the keys to the success of these systems has been the advance in analog-to-digital converters (abbreviated A/D or ADCs).

Pure analog circuits can do substantial signal processing in a low-cost and well-established fashion. For example, analog circuits are more than sufficient for simple processing functions such as filtering and amplification. With the complexity of advanced electronic systems, implementing them with pure analog solutions becomes too costly or even unfeasible. Digital signal processing (DSP) offers crucial extensions to these required functionalities as DSP provides perfect storage capability, unlimited signal-to-noise ratio and options to carry out complex algorithms to enable new features with the DSP's unprecedented computation power. To take advantage of such capabilities, analog signals have to be converted to digital signals in the early stage of the processing chain, making the analog-to-digital converter a critical design block. In many cases, the performance of today's digital system is defined by the quality and speed of the data converters.

Continuous and aggressive scaling of complementary metal-oxide-semiconductor (CMOS) technologies has dramatically increased the speed, power efficiency and integration of electronic systems. Moore's law continues to predict the scaling and levels

of integration fairly well and the rate of scaling even outperforms the prediction in recent years [10]. This scaling improvement in system performance has driven the need for improvement in its corresponding data converters. The trend is to continue development in high-performance data converters while simultaneously reducing ADC power consumption. Another trend is to shift the A-to-D conversions “upstream” to allow more signal processing to be done in the digital domain in order to take full advantage of digital scaling and to eliminate unwanted interferers and noise.

The number of ADC applications is also expanding. The application is as diverse as industrial process controls, communication infrastructure, automotive controllers, audio/video functions and medical devices among numerous others. Moving the data conversion upstream in these applications generally requires much higher sampling rate and resolution. For high performance applications, such as wireless communication devices, software radio, and millimeter-wave imaging systems, among others, moving the ADC upstream would require resolutions of 12 bits or higher and sampling rates of a few tens of megahertz (MHz), and the requirements are steadily headed toward a few hundreds of MHz or even in the gigahertz (GHz) range. Moreover, the increased popularity in portable/battery-powered electronics demands better energy efficiency in data converter design. This creates a number of challenges to achieve high performance, high resolution and low power in the same design, especially in the deeply scaled CMOS technologies.

1.1 Challenges of Technology Scaling

Technology scaling benefits digital integrated circuits in terms of improved integration and unity gain frequency f_T , but scaling does not necessarily benefit analog circuits, such as operational amplifiers, in the same way. As feature size shrinks, voltage headroom (V_{DD}), intrinsic gain of transistors ($g_m r_o$) and gate oxide thickness (t_{OX}) all decrease with scaling. Despite the benefits in speed, these factors make designing analog circuit extremely difficult.

Scaling lowers the supply voltage and reduces the available signal for critical analog

blocks, but unfortunately, it does not lower the noise floor and the rate of threshold voltage scaling is not proportional to that of the supply voltage in order to prevent excessive increases in “off-leakage” current. These further aggravate the difficulties caused by supply voltage shrinkage. For example, when the supply voltage reduces by half from 1.8V to 0.9V, the SNR automatically decreases by 6dB. To maintain the same SNR, the noise power needs to be four times smaller. To achieve such noise level, the capacitor size has to be increased by four times since noise power is proportional to kT/C . If the system is designed to have certain bandwidth of gm/C , the increase in capacitance needs to be accommodated by the increase in transconductance gm . This will result in two¹ times the power consumption to maintain the same bandwidth as in the original design.

Due to limited voltage headroom, it also becomes increasingly impractical to use cascoding techniques to improve the DC gain of operational amplifiers. To increase the DC gain, designers have resorted to using boosted cascoding or multi-stage designs [13, 14]. Even though these techniques can provide enough DC gain, they can introduce multiple poles at low frequencies, making it challenging to design a closed-loop stable system.

Device variation is another important factor. Variation manifests itself particularly in the variation of threshold voltage, given in Equations 1.1 and 1.2.

$$2\phi_B = 2 \frac{kT}{q} \ln \frac{N_a}{n_i} \quad (1.1)$$

$$V_T = V_{fb} + 2\phi_B + \frac{\sqrt{qN_a 2\varepsilon_s}}{C_{ox}} (\sqrt{2\phi_B + V_{sb}} - \sqrt{2\phi_B}) \quad (1.2)$$

The equations show that threshold voltage depends on the doping concentration (N_a), the flat-band voltage (V_{fb}) and oxide thickness (t_{ox}). The doping concentration especially suffers from random dopant fluctuation arising from the ion implantation and thermal annealing steps. This makes it difficult to develop techniques to mitigate the variation in threshold voltage, and this variation in threshold voltage makes matching difficult for deeply-scaled devices. Device variation leads to random offsets

¹Total capacitance is increased by four times but V_{DD} is reduced by two times.

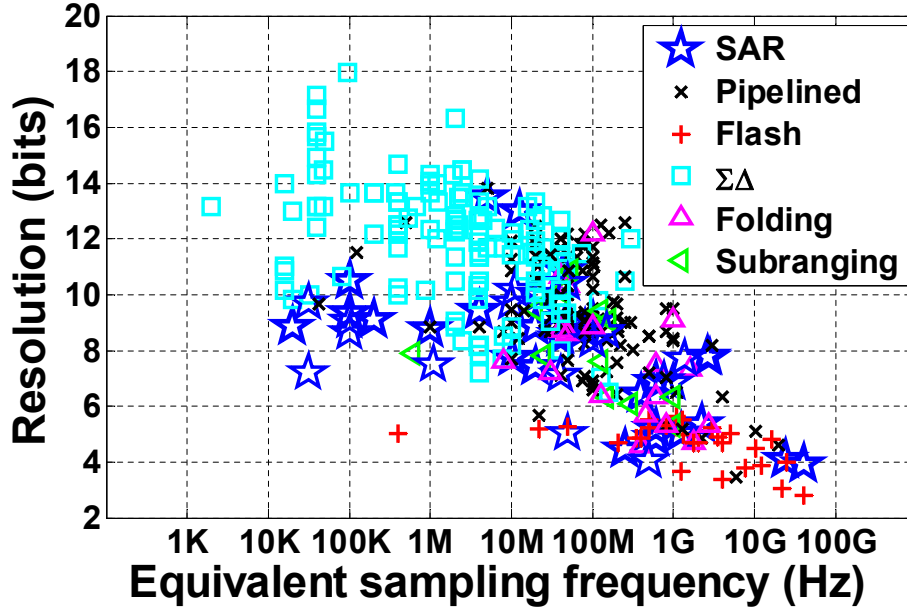


Figure 1-1: A plot of the resolution versus the input sampling frequency for recent published analog-to-digital converters in ISSCC and VLSI (data adopted from [1]).

in analog circuits, which can limit the achievable performance. Other short channel effects, such as drain-induced barrier lowering (DIBL), gate current leakage, velocity saturation, and parasitic source/drain resistance also raise concerns for analog-heavy design. Our goal is to take the benefits of digital scaling and design around these analog limitations.

1.2 ADC Architecture Overview

Figure 1-1 shows the resolution and sampling frequency for all ADCs published in key technical conferences in this field (ISSCC and VLSI) between 1997 and 2012 [1]. The plot shows the trend that increasing sampling frequency goes with decreasing resolution. Of the classical architectures, $\Sigma\Delta$ converters dominate the high resolution and low sampling frequency region, flash and folding ADCs have the highest sampling frequency but with the lowest resolution, successive-approximation-register (SAR) converters are used for low-to-medium speed and medium-to-high resolution applications, and pipelined converters are used for applications that require medium-to-high speed and resolution.

The flash topology, along with its folding and interpolating variants, has been the choice for high-speed and low-resolution applications. It is able to achieve the highest throughput, but it suffers from a number of drawbacks due to its high level parallelism. Since the number of comparators grows exponentially with the resolution, these ADCs require excessive power and area for resolutions above 8 bits. The large number of comparators also gives rise to other problems such as large input loading and kickback noise. Large input loading limits the speed of the ADCs, and kickback noise can affect the accuracy of references or the analog input. The ensuing difficulty motivates the use of other ADC architectures.

Sigma-delta converters are traditionally used for high resolution, low bandwidth digital audio applications. Bandwidth is typically in the kilohertz range and resolution can be as high as 18 bits. Recently, work has demonstrated converters with improved speed at a few megahertz samples per second [11, 12]. Sigma-delta converters trade off speed for resolution, and sample the input many times faster than the Nyquist rate in order to perform noise shaping. Because the internal circuits have to run at speed much faster than the sampling rate, the power consumption can be significantly higher compared to Nyquist rate ADCs. The design of digital decimation filters can also be challenging.

Pipelined ADCs are traditionally used for medium-to-high speed and resolution applications. One advantage of pipelined ADCs is that the hardware requirement scales linearly with the number of bits. By adding another pipelined stage, we can potentially increase the resolution of the overall pipelined ADC by the resolution of that extra stage. The parallelism enables high throughput at the cost of extra power consumption and latency. For example, a six-stage pipelined ADC would have a latency of at least six clock cycles between the analog input and the digital output. At the heart of the pipelined operations, it relies on the operational amplifier to multiply the residue from the previous stage to the next stage. The op-amp must be designed to have high gain/bandwidth to achieve the desired performance. In deeply scaled CMOS technologies, however, as discussed in Section 1.1, it would be difficult to achieve such gain with limited power supply while being closed-loop stable.

Recent work has demonstrated using an open-loop comparator or a zero-crossing-based detector in place of an op-amp to mitigate problems due to scaling [15–18]. In this thesis, though, we explore other architectures to overcome these difficulties.

Capacitor array successive-approximation-register (SAR) ADCs were introduced in 1975 by McCreary *et al.* [4] and have been extensively used for medium-speed applications. A conventional SAR ADC includes a digital-to-analog converter (DAC) driving a comparator. The comparator output is then processed by the digital control logic, which in turn feeds back a control signal to the DAC. This feedback logic is performing a binary search to find the correct digital output bits to minimize the difference between the voltage on the DAC and the analog input. The DAC is typically composed of binary-weighted capacitors, which also serve as the input sampling capacitor. A sub-DAC can be used to avoid large capacitor values and enable high resolution implementations. The architecture has very high energy efficiency since other than the comparator, the remaining blocks only consume dynamic power. One drawback of the SAR architecture is that it takes multiple clock cycles, usually the same as the number of bits, to generate an output. This has made it difficult for the SAR architecture to run at sampling rate more than 5MHz in the past. Digital scaling helps improve the speed of CMOS technologies, now making SAR a viable option for higher speed applications. Moreover, scaling issues affecting other architectures are not present in SAR ADCs because of its high digital composition.

1.3 Trend of Analog-to-Digital Converters

A widely adopted Figures of Merit (FoM), also called Walden’s Figures of Merit [3], that incorporates resolution, speed and power consumption in order to provide a platform for energy efficiency comparison is shown below:

$$FoM_1 = \frac{P}{2f_{sig} \cdot 2^{ENOB}} \quad (1.3)$$

$$ENOB = \frac{SNDR - 1.76}{6.02} \quad (1.4)$$

where P is the total power consumption, $ENOB$ is the effective number of bits, defined in Equation 1.4, and f_{sig} is the input frequency of the signal. $SNDR$ is the signal-to-noise-distortion ratio in dB measured with a sinusoidal input. This FoM is intended to provide a measure of how much energy is required to perform a conversion step, expressed in *picojoules* (pJ) per conversion step. The development of this FoM is mostly based on empirical data after surveying a large number of ADCs in academic publications or commercial ADCs. This metric is created under the assumption that power tends to scale linearly with the input frequency and SNDR. It allows designers to compare energy efficiency across ADCs operating under different conditions. However, this metric suffers from an important limitation. In a higher accuracy ADC that is 10 bits or more, the resolution is mostly limited by thermal noise, which is in the form of $\sqrt{kT/C}$. In order to increase the resolution by 1 bit (or SNR by 6dB), C has to quadruple. If the operating frequency is kept constant, the power consumption has to be increased by a factor of four for an improvement of a factor of two in resolution. This implies that improving the resolution by 1 bit automatically worsens the FoM by a factor of two.

To address these limitation due to thermal noise, a modified FoM is proposed by [8] as shown in Equation 1.5,

$$FoM_2 = \frac{P}{2f_{sig} \cdot SNTR^2} \quad (1.5)$$

where $SNTR$ is the signal-to-thermal-noise ratio. In the absence of distortion and quantization noise, $SNTR = 2^{ENOB}$. Since the sampled thermal noise of an ADC is in the form of $\sqrt{kT/C}$, the square of signal-to-thermal-noise-ratio ($SNTR^2$) is thus proportional to C . In other words, at a fixed sampling frequency, the increase in power requirement is the same as the increase in $SNTR^2$, making the overall FoM_2 constant. This makes FoM_2 more suitable for comparing high-accuracy ADCs that are thermal-noise limited.

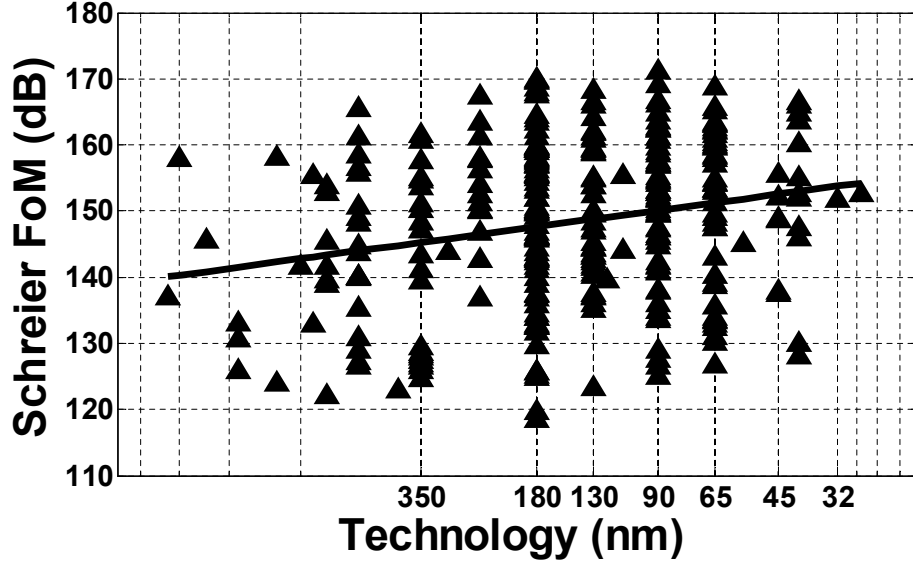


Figure 1-2: The Schreier Figure of Merit (FoM_3) versus CMOS process nodes from $1\mu\text{m}$ to 28nm of state-of-the-art ADCs published at ISSCC and VLSI Symposium (data adopted from [1]). Even though technology scaling does not directly benefit analog integrated circuits, steady improvement in conversion energy efficiency, FoM_3 , is still shown. This trend is the result of using more digital friendly architectures in recent designs.

Another variant of a figure of merit is named the Schreier FoM , listed below as FoM_3 [2]. It inverts the previous FoM_2 and expresses the term in dB, increasing in values for higher performance ADCs at the same frequency.

$$FoM_3 = SNDR_{dB} + 10 \cdot \log \left(\frac{2f_{sig}}{P} \right) \quad (1.6)$$

Figure 1-2 and Figure 1-3 plot Schreier's FoM (FoM_3) versus technology and years for the state-of-the-art ADCs published at ISSCC and VLSI Symposium between 1997 and 2012 [1]. It shows a general increasing trend; on average, FoM_3 increases by 1.3dB per year. This is surprising given that scaling makes designing analog circuits more challenging as described in Section 1.1. This improvement in FoM_3 could partially be attributed to the use and invention of more digital friendly ADC architectures.

Figure 1-4 and Figure 1-5 show the values of (1.3) plotted against sampling frequency and resolution, respectively. The best converters can achieve figure of merit

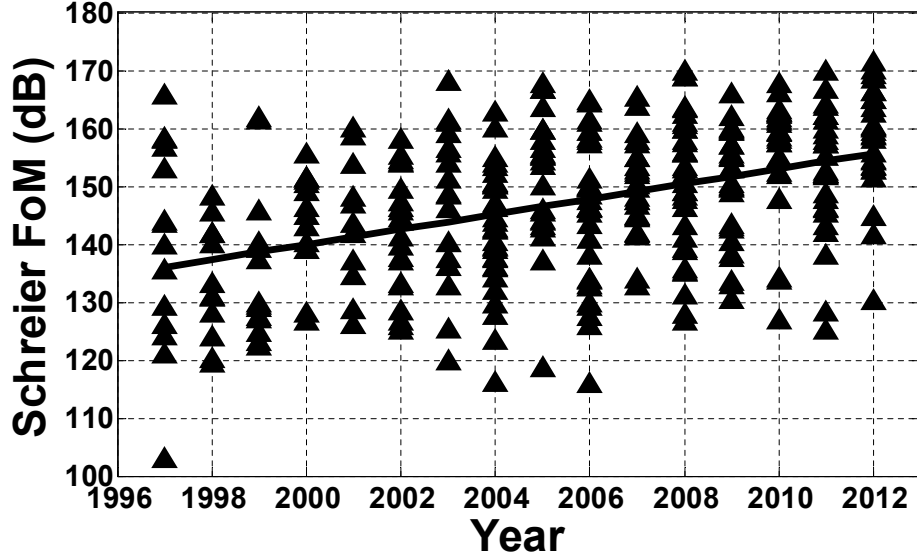


Figure 1-3: The conversion energy efficiency FoM_3 from year 1997-2012 (data adopted from [1]). This trend emphasizes the importance of energy efficiency in recent designs.

at tens of femtojoule per conversion step; however, these ADCs tend to be at resolution lower than 10 bits and sampling rate less than a few mega-samples per second. In terms of energy efficiency, Figure 1-4 shows that the SAR architecture dominates the figure of merit over all other architectures for all sampling frequencies between 10KS/s and 1GS/s. When sampling frequency increases, it becomes more difficult to achieve the same energy efficiency as for designs with lower frequencies. These so-called “high-speed” ADCs rely more heavily on the speed capability of the underlying transistors. To run at faster speed, extra power needs to be burnt to hit the performance targets.

In terms of resolution, Figure 1-5 shows another interesting trend. Converters with ENOBs between 6 and 10 bits are able to achieve the best Walden’s FoM (FoM_1). This window is the “sweet spot” for achieving energy efficient designs, and we refer to this as the Energy Efficiency Window (EEW). The EEW is especially suited for designs that are used for the battery-sensitive portable devices. For resolution lower than 6 bits, the design is typically targeted for very high-speed ADCs, where, as explained before, it is difficult to improve energy efficiency due to technology limitations. For resolution more than 10 ENOBs, thermal noise degrades FoM_1 . Because

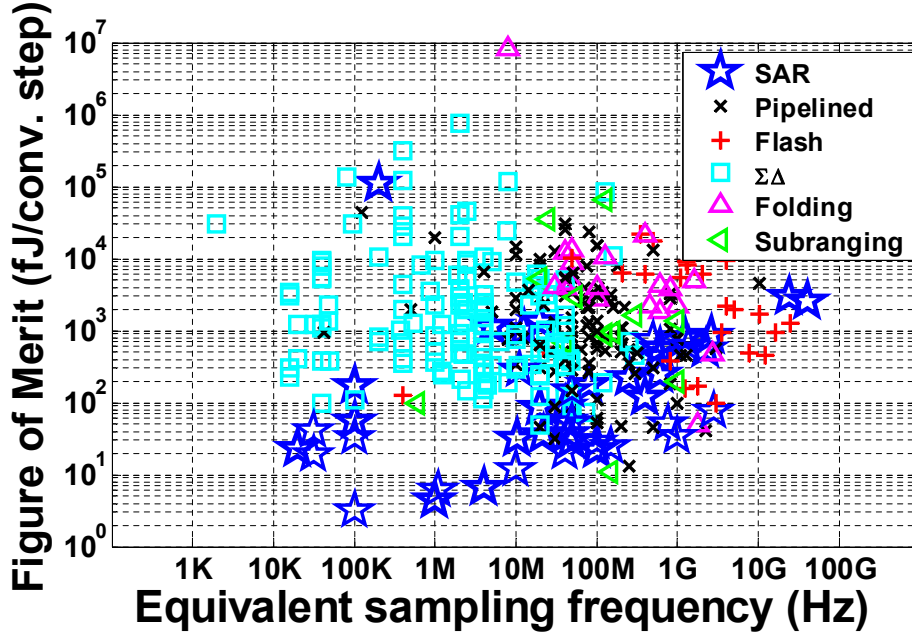


Figure 1-4: Walden’s FoM versus sampling frequency of state-of-the-art ADCs published at ISSCC and VLSI Symposium (data adopted from [1]).

the designs are noise-limited, various oversampling techniques are needed to lower the effective thermal noise within band. These techniques are typically hard to implement in an energy-efficient fashion due to its high oversampling ratio. As shown in Figure 1-5, within the EEW, the SAR architecture again dominates the energy efficiency over other architectures. One of the goals in this thesis is to expand the width of the EEW to allow energy efficient design in higher resolutions that have more than 10 ENOBs.

Because the goal of this thesis is to advance higher resolution ADC design, looking at the energy efficiency using another metrics will give us additional perspective. We re-plot Figure 1-4 in terms of FoM_3 in Figure 1-6, which is a more desirable figure of merit for comparing higher accuracy converters. For frequency less than roughly 30MHz, the top performance ADCs are clustered below what is called the “architecture front” by Schreier and Temes [2]. These ADCs have low input bandwidth but rather high resolution. The performance is typically noise-limited and FoM_3 is limited by the energy efficiency of the architecture, not by process technology. The diagonal line in Figure 1-6 is called the “technology front.” Converters clustered near

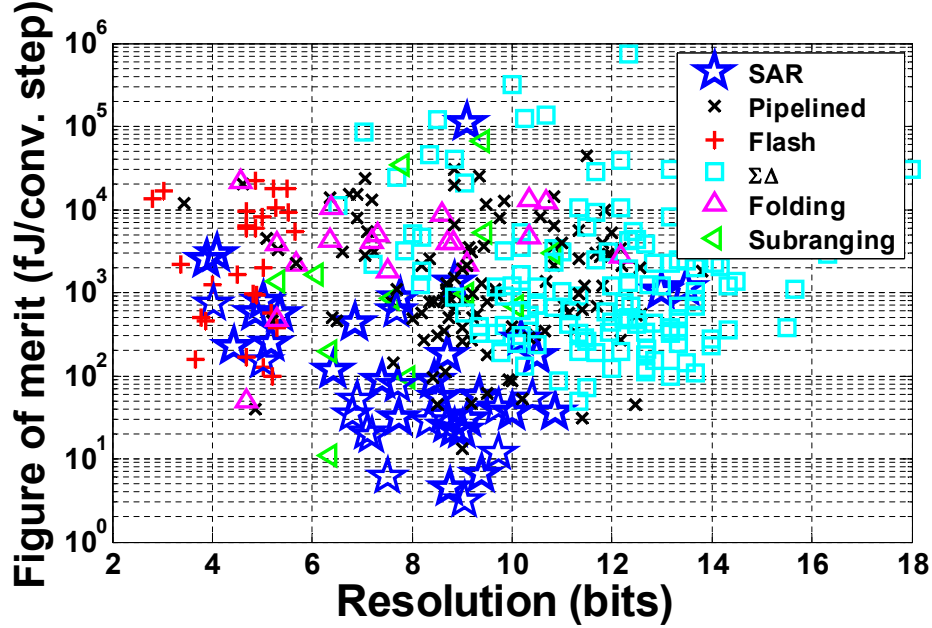


Figure 1-5: Walden’s FoM versus resolutions of state-of-the-art ADCs published at ISSCC and VLSI Symposium (data adopted from [1]).

this line typically have higher speed and moderate resolution. These ADCs rely on the speed capability of process technology and frequently, a less energy efficient architecture is used to enable higher-speed operations. An FoM_3 corner, marked on the plot, represents the intersection between the “architecture front” and the “technology front.” The work presented in this thesis helps push the “FoM corner” further to the right by using various new architectural techniques and energy efficient switching.

Figure 1-7 plots conversion energy versus SNDR. Both Walden’s and Schreier’s figures of merit (FoM and FoM_3) are drawn on the same plot. We can see that most ADCs are operating on the left of the red line ($FoM_3 = 170\text{dB}$), which represents the “architecture front” described previously. In regards to FoM_1 , most recent ADCs are moving towards achieving FoM close to a few tens of femtojoules per conversion step. The ADC designed in this thesis pushes the boundary further by showing an ADC with better SNDR and lower conversion energy. To meet this competitive energy efficiency at ENOB more than 10 bits, we choose to explore the SAR architectures because of its high energy efficiency, small feature size and good digital compatibility. Being free of precision analog circuitry (besides the comparator), SAR ADCs scale

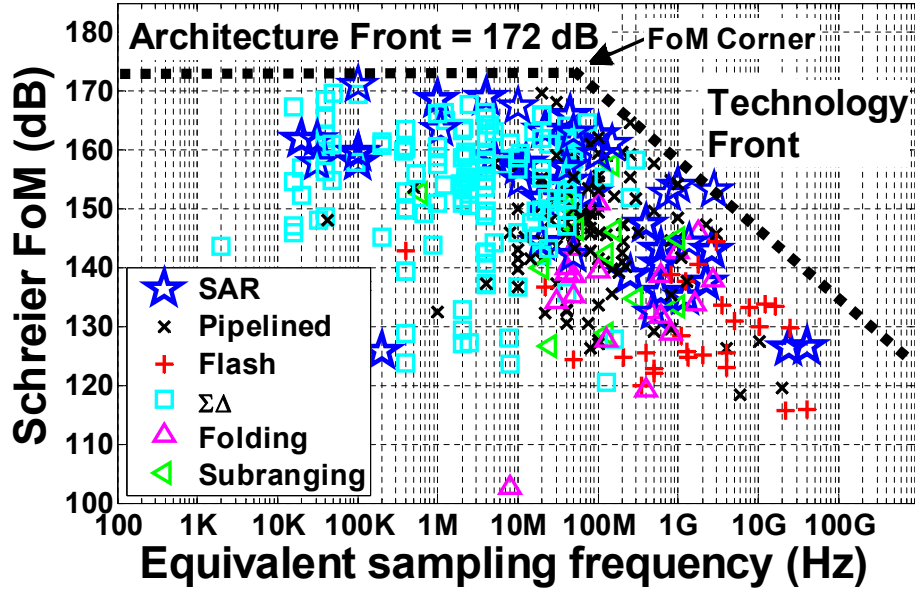


Figure 1-6: Schreier's FoM versus sampling frequency of state-of-the-art ADCs published at ISSCC and VLSI Symposium (data adopted from [1]). The plot shows the architecture front, the technology front, and the FoM_3 corner. These terminologies are introduced by Schreier and Temes in [2].

very well with technology as they are less affected by the degraded intrinsic gain and shrunk voltage headroom than opamp based architectures such as pipeline ADCs. They can take better advantage of the speed and energy efficiency in deeply scaled CMOS processes.

Despite the architectural advantage in its energy efficiency, there are still several pending limitations within the SAR architecture that need to be solved to push the energy efficiency below $FoM = 50fJ/\text{conv.-step}$ and performance beyond 10MS/s at resolution more than 10b ENOB. The key linearity and speed limiting factors are capacitor mismatches and incomplete DAC/reference voltage settling. Unfortunately, both of these errors do not scale as technology advances and can significantly limit the design from achieving the targeted performance. New precision techniques are crucial for the SAR architecture to cross such barriers.

Previous precision techniques include trimming and calibration. Post-fabrication laser trimming is often needed to achieve higher resolution. For example, the AD574 by Analog Devices used laser trimmed thin-film resistors to achieve the desired ac-

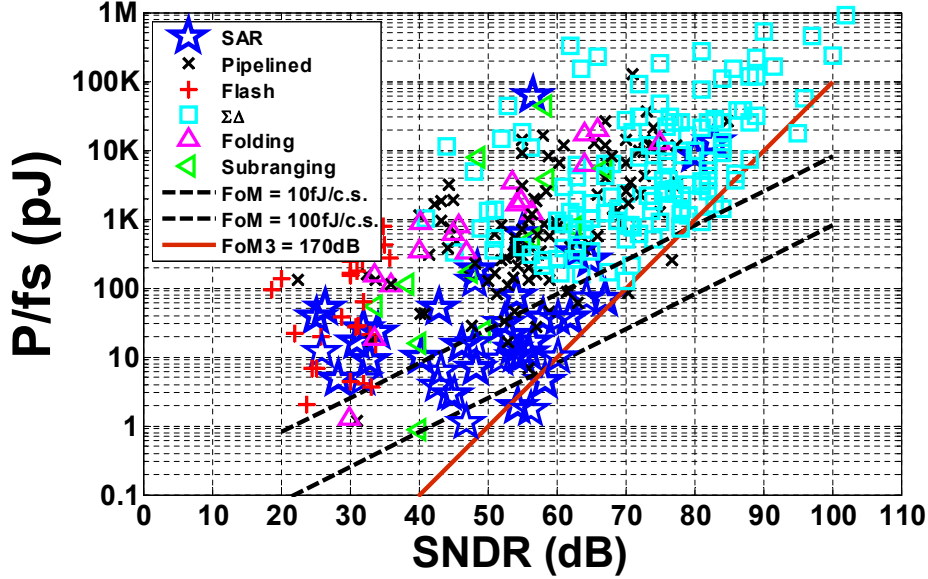


Figure 1-7: Energy per Nyquist sample versus SNDR of state-of-the-art ADCs published at ISSCC and VLSI Symposium (data adopted from [1]). The plot indicates the FoM at $100fJ/conv\text{-step}$ and $10fJ/conv\text{-step}$ with black dotted lines, and the red line denotes the “architecture front” of high-accuracy ADCs that are noise limited.

accuracy and linearity. The process adds additional cost and complexity to the manufacturing process. Since the trimming process is done during manufacturing, any subsequent drifts in the trimmed parameters cannot be corrected. For example, stress during packaging, temperature changes, aging, etc., may all change the trimmed parameters and re-trimming is typically not an option after the chip is shipped to the customers.

Pre-fabrication techniques have also been developed to mitigate this problem [9]. Techniques such as common-centroid layout, dummy device insertion and large device sizing are employed. These techniques help improve matching to a certain extent, but are insufficient to achieve our targeted precision level. Another category of precision techniques uses digital post-processing to correct for analog issues digitally. Since analog-to-digital converters are typically followed by a digital signal processor, this makes these digital calibration circuits easy to incorporate into the overall system.

There are two types of digital calibration: foreground calibration and background calibration. Foreground calibration relies on having prior-knowledge of the input calibration signal. Depending on the difference between the analog input and the

observed digital output signals, the conversion errors can be measured and corrected accordingly. One problem associated with foreground calibration is that in order to apply the stimulus at the input, it has to interrupt the normal A-D conversion operations. On the other hand, background calibration is transparent to the normal ADC operations. It analyzes the characteristics of the input and output relationship, and based on its specific system architecture, the calibration engine can optimize for the correct parameters. Typically, background calibration requires additional hardware or test input signals to fully explore the system and create enough observability of the errors.

1.4 Thesis Contributions

This work focuses on the design of high-precision, high-speed and energy efficient SAR ADCs, with particular emphasis on doing such design in deeply scaled CMOS processes. Our goal is to create an ADC that runs faster than 10 MS/s at more than 10b ENOB, while achieving FoM lower than 50fJ/conv.-step. The main contributions are investigation of using digital error correction (redundancy) in SAR ADCs for dynamic error correction and speed improvement, invention of two new calibration algorithms to digitally correct for manufacturing mismatches, design of new architecture to incorporate redundancy within the architecture itself while achieving 90% better energy efficiency compared to conventional switching algorithm [39], development of a new capacitor DAC structure to improve the SNR by four times with improved matching, joint design of the analog and digital circuits to create an asynchronous platform in order to reach the targeted speed and resolution, analysis of key circuit blocks to enable the design to meet noise, power and timing requirements, and the design and implementation of the entire SAR ADC in a standard CMOS digital process.

In Chapter 2, we review the operation of traditional successive-approximation-register ADCs. The architectural advantages of SAR ADCs and the common error sources that limit the SAR performance are discussed. The error sources are broken

down into static and dynamic error sources in this chapter, where each individual error source is analyzed in terms of their contribution to the overall ADC non-linearity. Chapter 3 presents the redundancy algorithm in the SAR architecture. Implementations using sub- and super-radix-2 are contrasted in terms of their digital calibratability. We derive the needed radix to accommodate for different levels of mismatches in the manufacturing processes. The effectiveness of using redundancy to improve the operational speed is also examined. All crucial building blocks of SAR ADCs in the critical path are taken into consideration while analyzing the potential benefits. Chapter 4 introduces two new background calibration algorithms designed specifically for our architecture to digitally correct for analog mismatches. Both algorithms are compared with previous calibration schemes in terms of their effectiveness and added complexity. Their effectiveness in correcting digital errors is demonstrated. A new architecture that incorporates redundancy, a new switching algorithm and a new split-capacitor array are presented in Chapter 5. Key design considerations and analysis of noise, matching, timing, clock distribution and bandwidth are shown. Chapter 6 presents the test setup and the measurement results. Both static and dynamic results are shown and compared with existing ADCs in the literature. Chapter 7 concludes our research findings and suggests potential future work.

Chapter 2

Approaches and Challenges in Traditional SAR ADCs

In the previous chapter, we described the importance of an ADC design to help push the boundary of modern electronic systems to a new level. Three different figures of merit are introduced to analyze the trend in recent converter design. Using the published data from ISSCC and VLSI, the trend shows that designs are moving towards having ADCs that run at faster speed and with higher resolution, but at the same time, achieving unprecedented levels of energy efficiency. Our goal is to design an ADC that resides in the region where it runs faster than 10MS/s and with SNDR better than 60dB. Together, our goal is to achieve a FoM that is better than $50fJ/conv\text{-step}$. Based on the analysis of different ADC architectures, the successive-approximation-register (SAR) ADC is chosen due to its energy-efficient switching and digital scalability with technology.

In this chapter, we dive further into the operation of a traditional SAR ADC. We first introduce the conventional search algorithm for Nyquist-rate ADCs. Knowing the search algorithm can help us better understand the deficiencies existing in different architectures. The second part of this chapter discusses the implementation and operation of a SAR ADC. The architectural requirements of the individual circuit blocks are also analyzed. Due to the non-idealities of these circuit components, errors will occur during the conversion process; these errors limit the achievable speed and

accuracy of a SAR design. The third part of this chapter focuses on analyzing these errors. Errors are broken down into static and dynamic parts based on the sources of these errors.

2.1 Search Algorithms for Nyquist-rate ADCs

Analog-to-digital converters convert continuous analog signals into discrete digital outputs. This process can be broken down into two parts. The first part is sampling and holding the analog input signal, and the second part is quantizing the sampled analog signal into digital bits. Sampling is a simpler process compared to quantization and therefore, there is less research and design variation in sampling circuits. On the other hand, the design of quantization methods and circuits is a much richer topic. As already hinted in the previous chapter, various architectural solutions cover different combinations of bandwidth, resolution and energy efficiency.

An ideal N -bit quantizer divides the full input range into 2^N distinct output decision levels. For each analog input, the goal of the quantizer is to search for the decision level that is nearest to the analog input. In the ideal case, the difference between the analog input and the quantized digital output should be less than half of the LSB, corresponding to $V_{FULL}/2^{N+1}$, where V_{FULL} is the full signal swing of the ADC. Fundamentally, depending on the required signal bandwidth, some architectures require one clock cycle to complete the search process, while others may require more than one clock cycles. For modest signal bandwidth, the allocated time for each conversion is long enough to allow the search algorithm to use multiple clock cycles to complete one conversion; for high signal bandwidth, on the contrary, the conversion process is required to be done as soon as possible to maximize the speed of the overall system, and therefore, only one or limited number of clock cycles is permitted for each conversion.

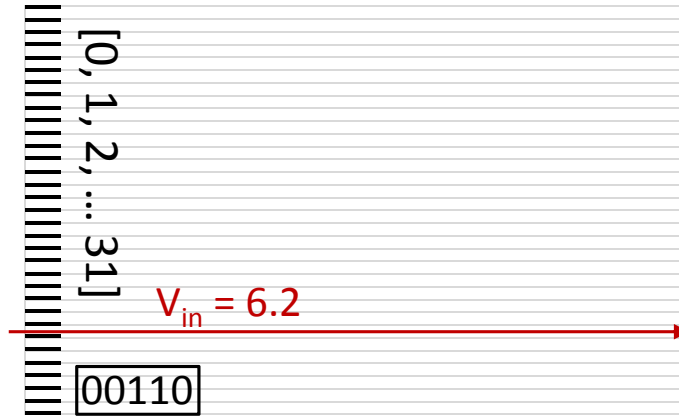


Figure 2-1: An example of 5-bit quantization using “brute-force” direct search. It is done by directly comparing the analog input with all $2^N - 1$ decision levels.

2.1.1 Flash Algorithm

Flash ADCs find the digital output codes in a “brute force” fashion by directly comparing the analog input with all $2^N - 1$ possible decision levels at once. If the analog input falls between the i^{th} and $(i + 1)^{\text{th}}$ decision levels, the input is mapped to the i^{th} digital output. By convention, the largest analog input is mapped to $11 \cdots 1$ and the smallest analog input is mapped to $00 \cdots 0$. Figure 2-1 shows an example of a 5-bit ADC with full signal range from 0 to 32, quantizing an analog input of 6.2. As 6.2 is greater than the 00110 decision level, but smaller than the 00111 decision level, it is converted to the binary code 00110.

A typical flash implementation is shown in Figure 2-2. In flash ADCs, the resistor ladder on the left generates the decision levels, the comparators generate the limits in which the input is greater than one of the decision levels and produce thermometer output codes, and finally, the decoder decodes the thermometer codes and converts them into binary output bits. As demonstrated by this example, the hardware requirement grows exponentially (as 2^N) with the number of bits, N . As a result, even though it is a time-efficient architecture, it can become too expensive for applications that required higher resolutions.

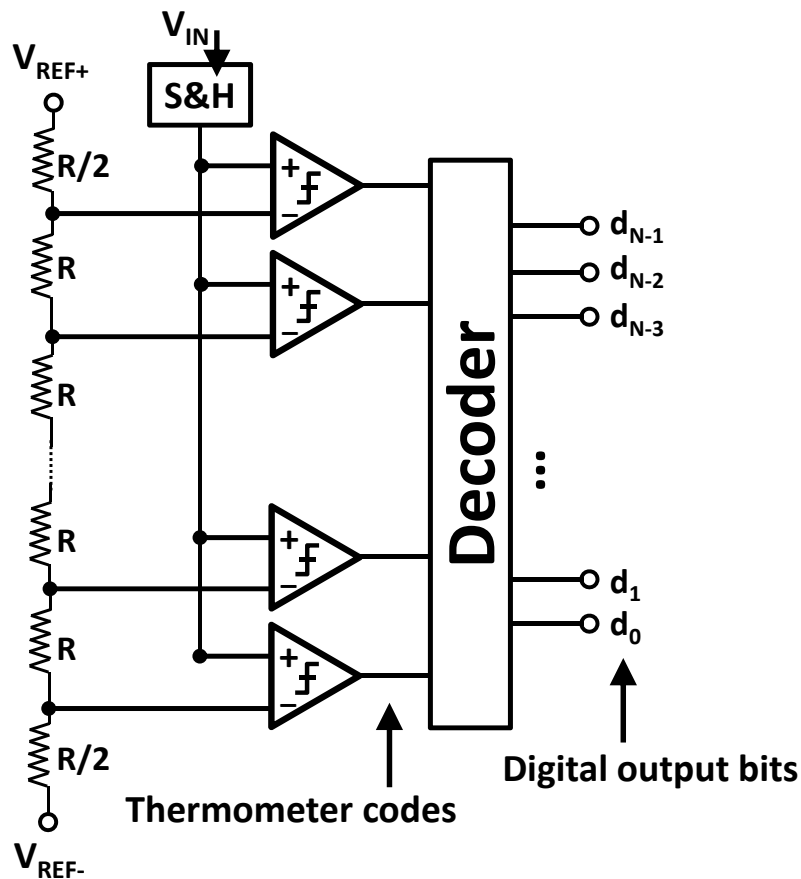


Figure 2-2: Basic architecture of a flash converter. The resistor ladder generates the needed decision levels and the following comparators generate thermometer output codes that represent the limit in which the input is greater than one of the decision levels. The decoder then converts the thermometer output codes into binary-weighted output bits. The number of the required comparators scales exponentially with the number of bits.

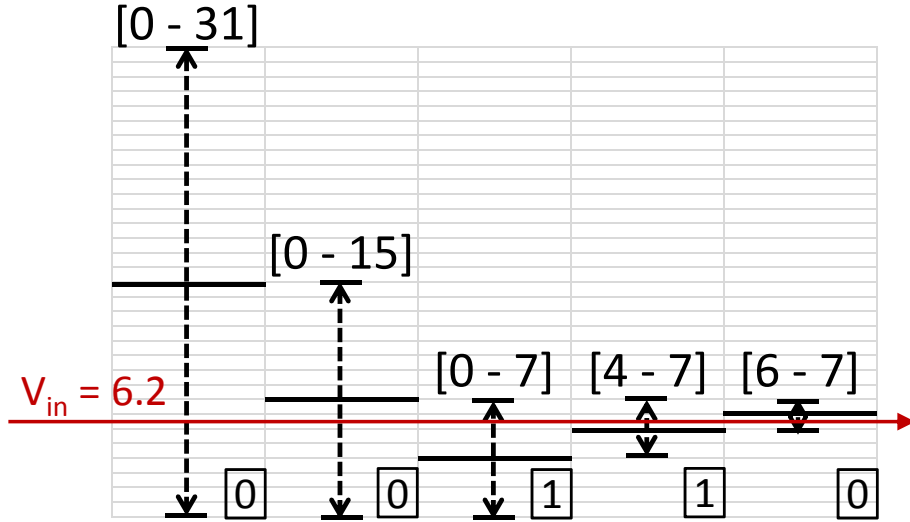


Figure 2-3: An example of 5-bit quantization using a binary search algorithm. Instead of using just one clock cycle per conversion, it requires five clock cycles to complete the conversion process and to realize the final 5-bit output.

2.1.2 Binary Successive Approximation Algorithm

Binary search resolves the output one bit at a time. It generates the first bit by comparing the input to the mid-full-scale-level of the current search range. Depending on the comparison outcome, it eliminates half of the search range and continues the same process until the entire conversion is completed. Instead of using one clock cycle per conversion, it requires N clock cycles and thus, N comparisons to complete a conversion. While binary search takes more clock cycles to complete a conversion than flash, it significantly relaxes the hardware complexity because all N comparisons share and use the same set of hardware. In this respect, binary successive approximation search is the exact opposite of the flash search algorithm: binary successive approximation search is hardware efficient but time consuming, while flash search is time efficient but hardware intensive.

Figure 2-3 shows an example of a 5-bit quantization of input 6.2 using binary successive approximation search. The solid black lines represent the mid decision level of the current search range and the solid red line indicates the location of the input level. In the beginning of the process, the search range is from 0 to 31. During the first comparison, V_{IN} (equal to 6.2) is compared with the mid-full-scale level of

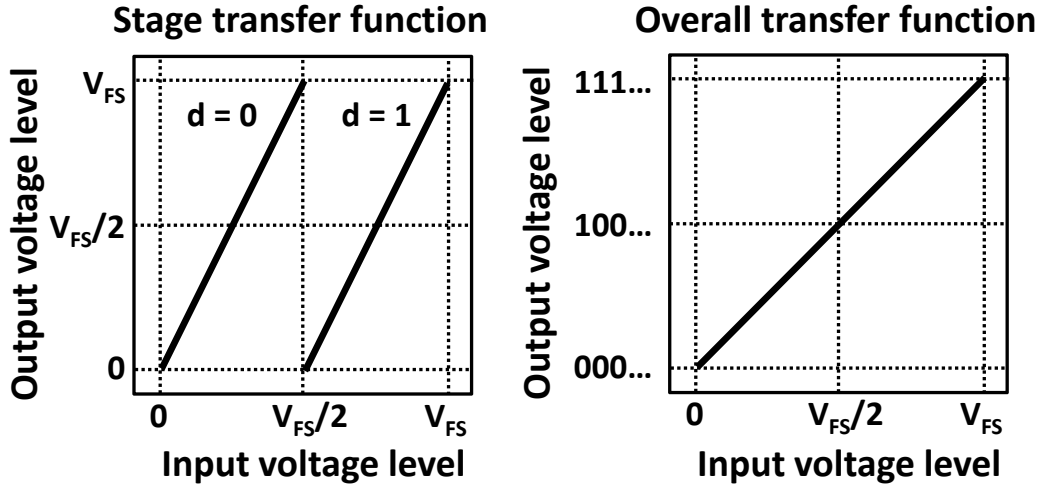


Figure 2-4: The transfer function of each multiply-by-two pipeline stage for the second type of binary search algorithm.

the initial search range. Since 6.2 is less than 16, the ADC outputs a '0' and the search range becomes the lower half of the previous search range. The search process continues for a total of five clock cycles to produce the final binary output equal to 00110. The last search reduces the range of uncertainty to one *LSB*, resulting in quantization error within $\pm 0.5LSB$.

Binary conversion is quite sensitive to errors made during the conversion process. In a typical binary implementation, none of the search ranges overlap. This implies that once a search range is dropped from the search process, it can never be re-entered, so if an error is made, there is no decision path that recovers or returns to the correct search range and thus the digital output can never be corrected. As a result, to produce correct digital outputs, it is important that each conversion step is accurate and correct, which is difficult to accomplish in practice. Traditional SAR ADCs use the binary search algorithm; however, we will see in a later chapter that digital error correction (or redundancy) can be used to greatly alleviate this problem.

2.1.3 Pipeline Algorithm

Binary search can be done in another way to speed up the overall conversion process, using what is typically called the pipeline architecture. The transfer function of each

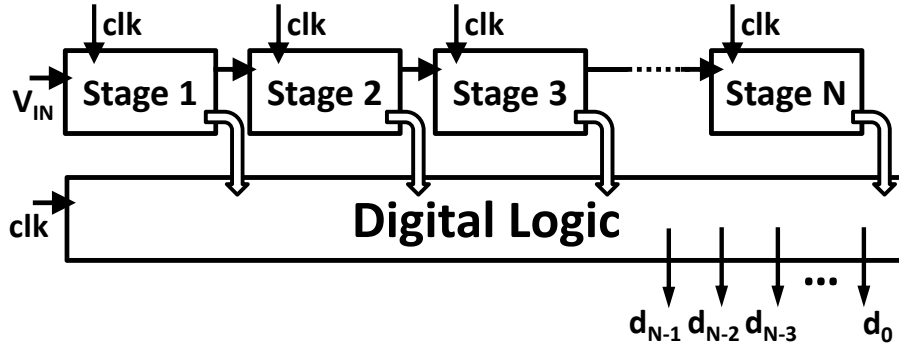


Figure 2-5: The block diagram for a traditional pipeline ADC. By cascading N 1-bit stages together, the ADC is able to produce N -bit resolution outputs.

multiply-by-two stage is drawn in Figure 2-4. Each stage determines whether the input is greater or less than the mid decision level and generates an output bit, d . The stage then produces a residue signal by subtracting 0 (when $d = 0$) or $V_{FS}/2$ (when $d = 1$) from the input. Finally, the residue is multiplied by two to restore the full signal swing for the next pipeline stage. This process essentially involves removing what is known and amplifying the leftover quantization error, which is unknown, for the next stage. The signal range stays constant and the input of each stage is always compared to the same decision level. This significantly relaxes the design complexity and reference voltage generation. By cascading N stages together, as shown in Figure 2-5, we can achieve a total of N -bit resolution.

Even though all stages are performing essentially the same function, the later stages can be designed with much relaxed noise and matching specification because the input signal has been amplified by 2^x times, where x is the total number of preceding stages assuming each stage has a multiplication factor of 2. In other words, due to signal amplification, we achieve noise suppression as a side benefit. To illustrate the operation in an example, Figure 2-6 demonstrates a 5-bit quantization process using the pipeline operation.

In the first stage, since V_{IN} is less than 16, the ADC outputs a '0' and subtracts 0 from the input. The remaining residue, 6.2, is multiplied by two to generate the input level, 12.4, for the next stage. In the second stage, since V_{IN} is still less than 16, the same process of residue amplification repeats and we obtain 24.8 as the input

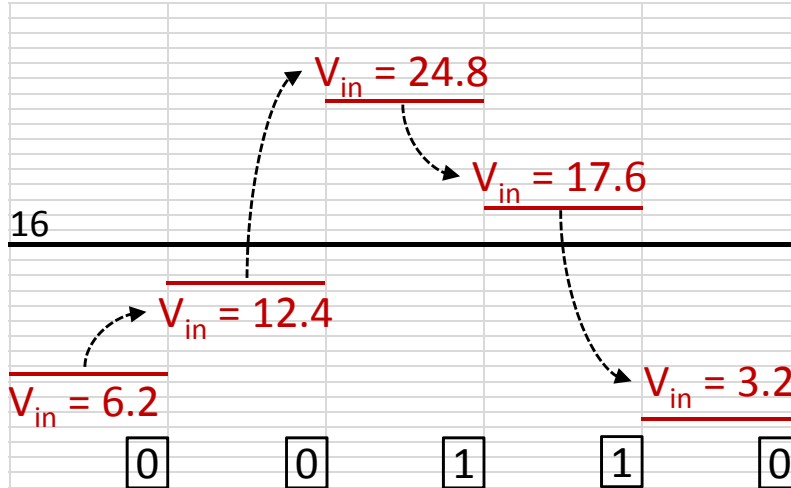


Figure 2-6: An example of a 5-bit quantization using the second type of binary search.

to the third stage. In the third stage, we now have $V_{IN} = 24.8$, which is greater than 16. Therefore, 16 is subtracted from the input first and the residue is again amplified by 2 times. The same conversion process continues until it reaches the last bit and the ADC successfully converts 6.2 to the final bit-sequence of 00110. In general, the pipeline architecture can have more than 1-bit of resolution per stage and each stage does not need to have the same number of bits. The total resolution of the ADC is the sum of bits at individual stages.

2.1.4 Summary of the Search Algorithms

We have described three basic search algorithms for Nyquist-rate ADCs in this section. In real implementations, these algorithms are realized by analog circuits, such as amplifiers, comparators, filters and references along with digital control logic. Each algorithm requires various degrees of accuracy from each component, and depending on its composition, it has its own merits and disadvantages. As discussed in Chapter 1, the SAR architecture implementing the binary search algorithm has gained great popularity in recent years. The architecture is relatively simple to design and scales well with technology due to its high digital composition. Scaling benefits SAR in terms of its switching speed and energy efficiency. At the same time, a SAR design usually has much smaller feature size compared to ADC designs with the same reso-

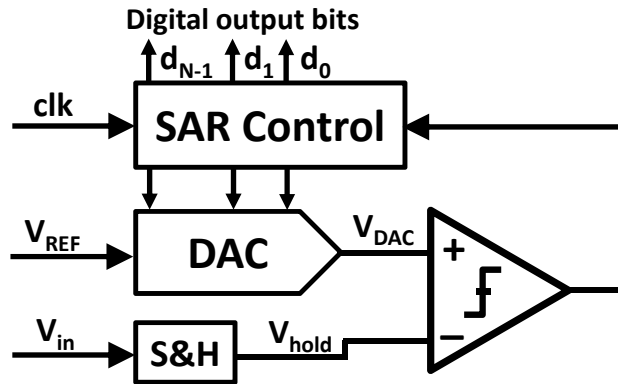


Figure 2-7: Basic block diagram of a SAR ADC. It includes a S&H, a DAC, a comparator and a SAR control logic block.

lution but implemented with flash or pipeline architectures. The flash architecture is losing its advantage in speed because of technology scaling and improvement in time-interleaving techniques. Designs now do not have to sacrifice power consumption, resolution and hardware complexity for speed. The pipeline architecture requires precision amplifiers at the core of its operations; technology scaling makes it more challenging to design a high gain and bandwidth amplifier that is closed-loop stable.

2.2 The SAR Architecture

The SAR architecture performs the A-to-D conversions over multiple clock cycles by using the information of the previous determined bit to assist in finding the next significant bit. Figure 2-7 depicts a typical block diagram of a SAR ADC. It involves four basic building blocks: sample and hold (S&H), DAC, comparator and SAR control. The S&H samples one instance of the continuous analog input signal during the first clock period and holds the value for the remaining conversion process. The comparator resolves each bit by comparing V_{hold} with V_{DAC} . The SAR control reconfigures and updates the DAC according to the output bits of the comparator.

An effective implementation of the DAC is the so-called charge redistribution or capacitor array scheme [4, 6]. It merges the sample/hold function together with the capacitive DAC to perform subtractions in the charge domain using capacitors. At the end of the conversion process, the charge is properly re-distributed such that the

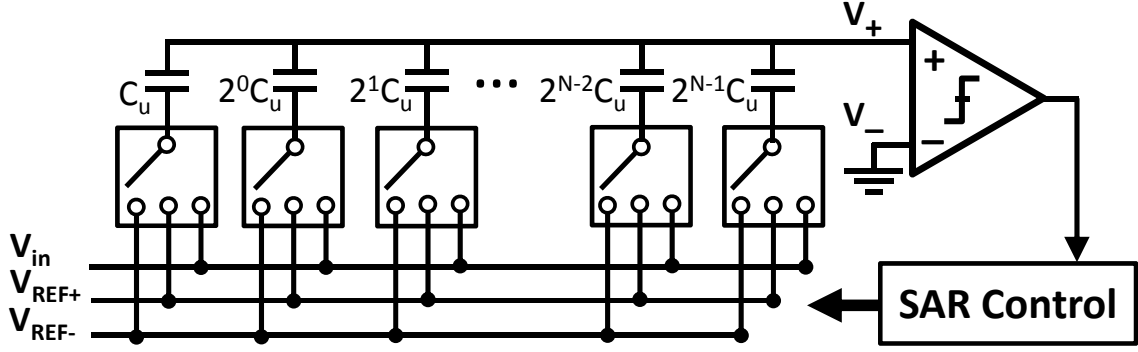


Figure 2-8: Schematics of the charge redistribution SAR implementation.

top plate voltage on the DAC is the same as the voltage on the other input of the comparator, which in this case is zero as depicted in Figure 2-8. The SAR consists of an N -bit binary-weighted capacitive DAC, a comparator and a SAR control logic block. Each capacitor within the DAC can be re-configured to connect to either the input or the pulse/minus reference voltages. The total capacitance sums up to C_{Tot} , where

$$C_{Tot} = \sum_{i=0}^{N-1} 2^i \cdot C_u + C_u = 2^N \cdot C_u. \quad (2.1)$$

During the sample and hold phase, the DAC array samples the input signal by connecting the bottom plates of the array to the input and the top plate of the array to ground (Figure 2-9(a)). The total charge stored in the array is

$$Q_{Tot} = (0 - V_{IN}) \cdot C_{Tot} = -V_{IN} \cdot C_{Tot}. \quad (2.2)$$

After the sampling phase, we enter the conversion phase. During the first step, we connect the most-significant-bit (MSB) capacitor to V_{REF+} and the remaining capacitors to V_{REF-} as shown in Figure 2-9(b). For simplicity, in our example, we assume $V_{REF+} = V_{REF}$ and $V_{REF-} = 0$. Using the superposition principle, the voltage on the top plate of the array, V_+ , becomes

$$V_+ = -V_{IN} + \frac{2^{N-1} \cdot C_u}{C_{Tot}} \cdot V_{REF} = -V_{IN} + \frac{1}{2} \cdot V_{REF}. \quad (2.3)$$

The first term represents the contribution of input sampling and the second term represents the contribution from the MSB capacitor. By comparing V_+ directly to ground, we can determine the first output bit d_{N-1} and set the configuration for the next bit calculation. If $d_{N-1} = 1$, $2^{N-1}C_u$ stays connected with V_{REF} ; if $d_{N-1} = 0$, $2^{N-1}C_u$ is switched to ground for the remaining cycles. In both cases, $2^{N-2}C_u$ is switched to V_{REF} . The two different configurations can be shown in Figure 2-9(c) and Figure 2-9(d), respectively. The top plate voltages of the two configurations become Equations 2.4 and 2.5. The process of comparing and reconfiguring continues until we reach the last bit.

$$V_+ = -V_{IN} + \frac{(2^{N-1} + 2^{N-2})C_u}{C_{Tot}} \cdot V_{REF} = -V_{IN} + \frac{3}{4} \cdot V_{REF} \quad (2.4)$$

$$V_+ = -V_{IN} + \frac{2^{N-2}C_u}{C_{Tot}} \cdot V_{REF} = -V_{IN} + \frac{1}{4} \cdot V_{REF} \quad (2.5)$$

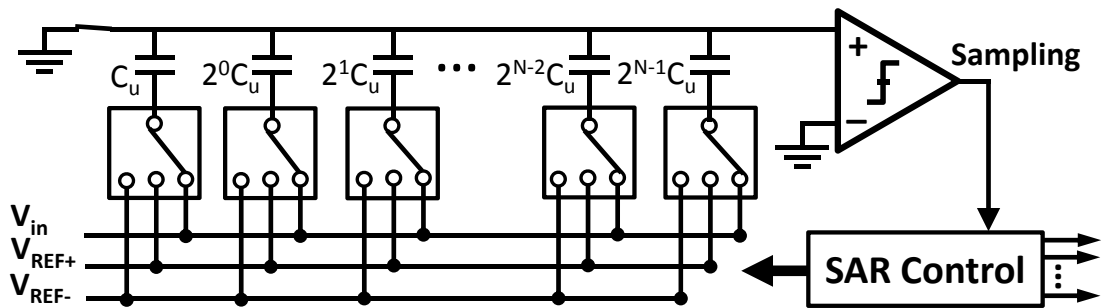
At the end of the conversion, the ADC converts the input into binary-weighted bit sequences, $[d_{N-1}, d_{N-1}, \dots, d_0]$, and the final voltage on V_+ is

$$V_+ = -V_{IN} + \sum_{i=0}^{N-1} 2d_i \cdot \frac{2^i C_u}{C_{Tot}} \cdot V_{REF} - \frac{C_u}{C_{Tot}} \cdot V_{REF}. \quad (2.6)$$

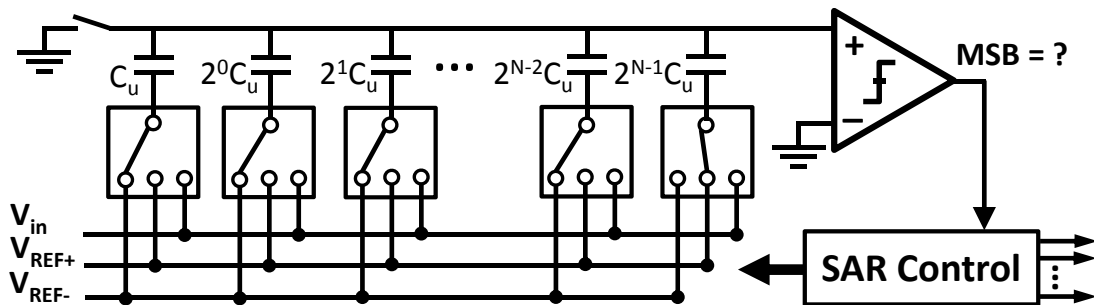
This voltage represents the quantization error of the entire conversion process. Note that both the top and bottom plates of the DAC can have parasitic capacitances contributed from non-ideal layout/wiring, channel capacitances of MOS switches and gate capacitance of comparators. The parasitic capacitances on the bottom plate are driven by low impedance reference supplies, V_{REF+} and V_{REF-} . Typically, these do not affect the conversion process as long as the reference voltages are completely settled. The parasitic capacitance on the top plate, on the other hand, attenuates the amplitude of sampled input. The attenuation factor can be calculated as

$$\beta = \frac{C_{Tot}}{C_{Tot} + C_P} \quad (2.7)$$

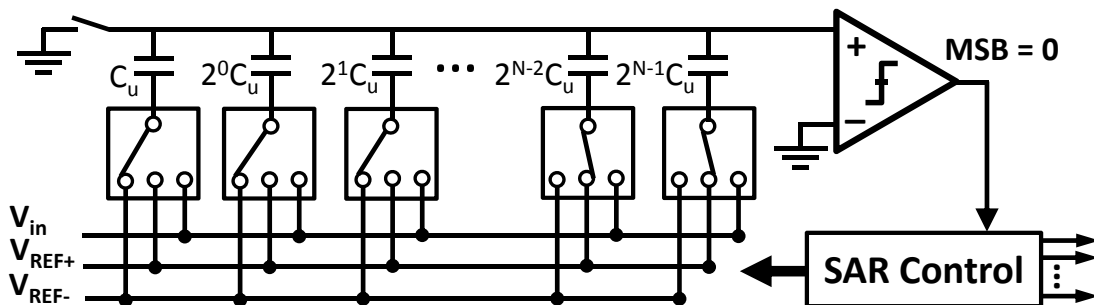
where C_P is the total parasitic capacitance on the top plate. This attenuation reduces



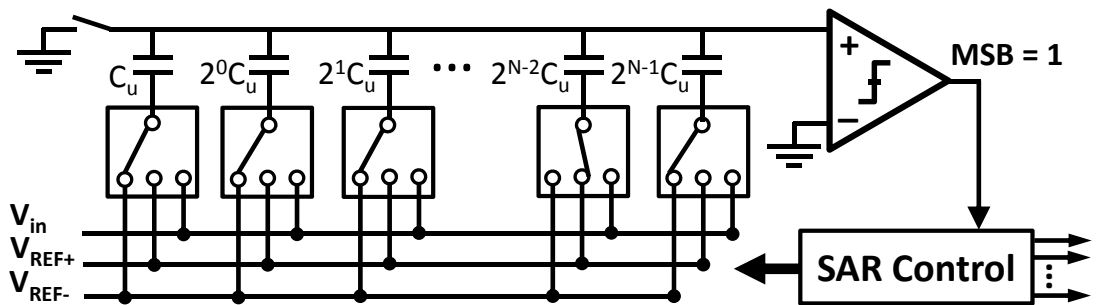
(a) Sample and hold phase.



(b) Conversion phase, step 1.



(c) Conversion phase, step 2a.



(d) Conversion phase, step 2b.

Figure 2-9: Switching scheme of a conventional SAR ADC.

the effective signal power, but does not change the polarity of the comparison result, which is the only relevant information for determining the correct output bits. The bottom-plate sampling essentially enables this feature. In the sampling phase, the top plate is pre-charged to ground before the node becomes floating and remains floating until the end of the conversion phase. During the conversion, the voltage on the top plate moves but returns to a voltage that is near zero at the end of the process. As a result, the total charge on C_P is the same at the beginning and at the end of the process and therefore, from the perspective of charge, capacitor C_P does not cause any charge error. Therefore, it does not affect the overall accuracy of the conversion process.

In summary, the advantages of using a charge redistribution scheme in a SAR ADC is that it is energy efficient and only has dynamic but no DC power consumption, if no pre-amplifier is used in the comparator design. The ADC is robust against circuit non-idealities, such as parasitic capacitances. The architecture is less limited by technology and supply voltage scaling compared to other architectures; instead, it has the potential to take full advantage of improved energy efficiency and speed in deeply-scaled CMOS due to its high digital composition. If implemented correctly, a SAR ADC typically supports full rail-to-rail input range, which can be advantageous for high-resolution designs. Lastly, since it shares the sampling capacitors with the configurable DAC, SAR ADCs can save significant areas and result in small chip area.

2.3 Static Error Sources in SAR ADCs

Even with all the architectural benefits discussed in the previous section, the converter resolution is contingent on the matching of analog components. For example, mismatches in the capacitive DAC can lead to incorrect charge distribution during the conversion phase; mismatches in transistors can lead to offset errors in the comparator. To fully characterize and evaluate the performance of an ADC, in addition to using dynamic metrics, such as ENOB, SNDR, SFDR, etc., discussed in Chapter 1, static metrics are also important to look at. The most common static metrics

are differential nonlinearity (DNL), integral nonlinearity (INL), offset error and gain error.

The offset error quantifies the amount by which the actual transfer function is linearly shifted from the ideal transfer function. The gain error quantifies the slope deviation of the transfer function from the actual staircase slope. Both gain and offset errors do not introduce harmonics and nonlinearity and therefore, they typically are given less design attention; however, in some applications, such as in test and measurement, they are important error sources and need to be removed or calibrated out.

DNL is defined as the deviation of the actual step width from the ideal value of 1 LSB. For an ideal N -bit ADC, the output is divided uniformly into 2^N equal analog steps, each with size 1 LSB, and therefore, DNL is equal to 0 LSB for all steps. When the specified DNL is within ± 1 LSB, monotonicity is guaranteed with no missing codes. Monotonicity implies that digital output increases or remains the same for increasing analog input, and therefore, there are no sign changes in the transfer function. The DNL is usually characterized after the static gain error has been removed and defined as follows,

$$DNL_i = [(V_{in}(i+1) - V_{in}(i))/V_{LSB-IDEAL} - 1] \quad (2.8)$$

where $V_{in}(i)$ is the analog transition voltage for the i th digital output code and $V_{LSB-IDEAL}$ is the ideal spacing between two adjacent digital codes.

The step-size error quantized by DNL can lead the transfer function to move away from the ideal straight line. This deviation of the actual transfer function from an ideal straight line transfer function is quantized by the INL. Since it is the total deviation from the ideal straight line transfer function, INL for a specific code can be obtained by summing all the previous DNL's up to that code as given by Equation 2.9.

$$INL_i = \sum_{x=0}^i DNL_x \quad (2.9)$$

2.3.1 Capacitor Mismatches

Good capacitor matching is the key for high accuracy ADCs. Matching is controlled and influenced by manufacturing processes and physical design. The variation sources can be broken down into random statistical fluctuation and systematic mismatches. Random mismatches include fluctuations in device dimensions, wire sizing, doping, oxide thickness and other effects that change component values. These type of mismatches cannot be completely eliminated. Typically, the best solution is to increase the overall dimension to improve matching; this approach works in some cases, for example when a constant (small) deviation can be reduced by averaging over a larger size area. Systematic mismatches are the result of temperature gradients, diffusion interactions, mechanical stresses, biases in the processing steps, and a host of other causes. Even though some of these mismatch sources can be combated through careful design and layout, it is still difficult to attain more than 10 bits of resolution.

When capacitors within the DAC are perfectly matched in a SAR ADC, the input/output transfer function resembles a straight curve, shown as a dotted line in Figure 2-10. This implies that all the steps have equal size and they are evenly spaced over the full range to create a linear mapping between the inputs and the outputs. This 12-bit example is free of any DNL and INL errors.

On the other hand, when mismatch errors are present, the transfer function deviates from the straight line and the decision levels are no longer uniformly spaced. As shown by the solid blue curve in Figure 2-10, misalignments occur in both the vertical and horizontal directions. Misalignment in the vertical direction creates missing codes, which implies that certain digital output codes do not occur at the outputs and the DNL exceeds -1 . Misalignment in the horizontal direction creates missing levels, which implies that multiple analog inputs map to the same digital outputs and some part of the original analog information is lost. Typically, missing codes are digitally correctable and missing levels are not. As a result, ADCs should be designed to avoid missing levels. More details on digital calibration will be discussed in Chapter 3 and 4. Figure 2-11 shows the plot of ENOB versus the standard deviation

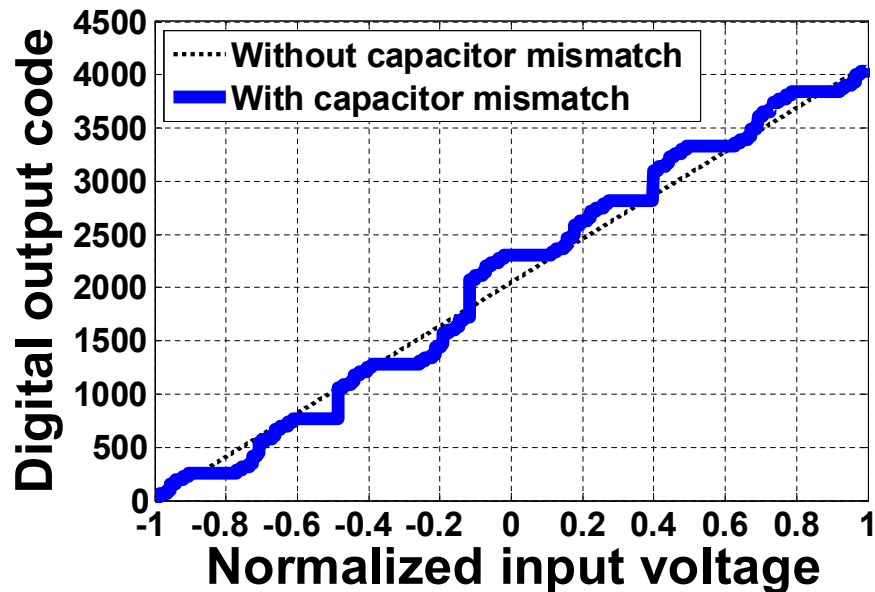


Figure 2-10: An example ADC transfer function for SAR ADCs with/without capacitor mismatches.

of the unit capacitor, simulated using behavioral models. Using Pelgrom’s mismatch model, the standard deviation of the larger capacitors is scaled up by a factor that is proportional to the square root of its normalized area to the unit capacitor [5]. The standard deviation is swept from 0 to 0.1, each time with 40 runs. We see a strong correlation between the achievable ENOB and the variance in capacitors. Even at 1% standard deviation in C_u , the ENOB can be degraded by more than 1 bit without taking into consideration thermal noise or other non-idealities in the design. Therefore, for high-resolution design, it is important to control and calibrate for the mismatches in capacitors.

2.3.2 Offset Errors

The offset error in a SAR ADC only causes a linear shift in the transfer function, but does not introduce linearity problems since the error is signal-independent. There are two sources of offset. The first source is a result of charge injection from the sampling switches. When the switch turns off at the sampling instance, the charge stored in the gate-to-channel capacitors is injected onto the top plate of the DAC. Since bottom-plate sampling is typically employed during sampling, the amount of charge injected

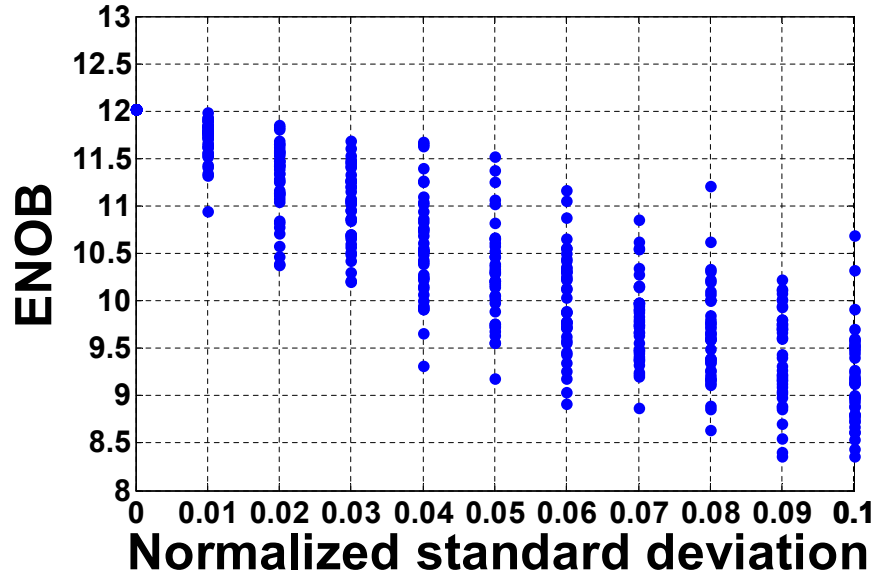


Figure 2-11: Effective number of bits (ENOB) versus normalized capacitor mismatch σ_{C_u}/C_u in a 12-bit binary weighted SAR ADC. The plot shows that 1% unit capacitor mismatch can sometimes lead to 1b loss in ENOB.

onto the plate is mostly constant and independent of the input signal, at least to the first-order estimation. The second source of offset errors in a SAR ADC is the offset of the comparator. The offset in the comparator is also signal-independent for two reasons. First, unlike in some other architectures (for example, the flash ADC), only one comparator is used repeatedly during the conversion phase. Therefore, only the offset of that comparator affects the operation. Second, for different input voltages, the top plate always returns to zero at the end of the conversion phase. This implies that the input common mode voltage of the comparator at the end of the conversion is the same regardless of the input signal, and thus, the offset voltage is always the same.

The residue at the end of the conversion is given by Equation 2.10. It shows that the additional terms introduced by offset voltages do not depend on the input voltage, V_{in} . Even though it does not introduce nonlinearity, offset voltage can become an important factor in measurement application or ADCs intended for time-interleaving purposes. Offset cancellation techniques may be used for these applications, but this comes at a cost of increased design complexity and degraded energy efficiency and

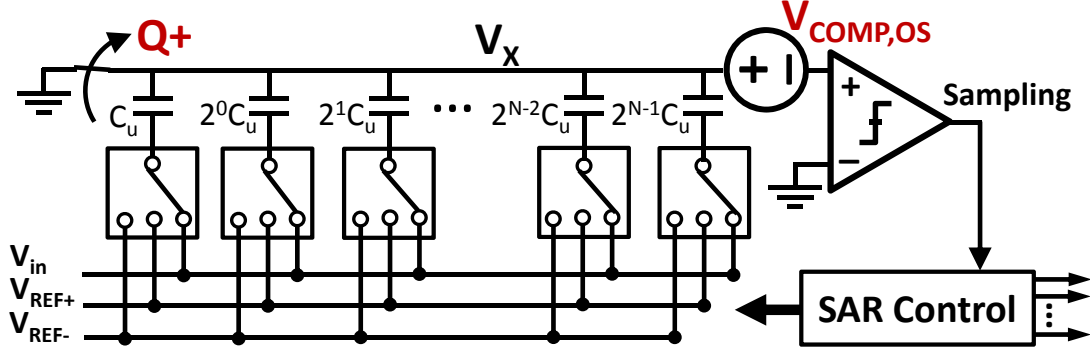


Figure 2-12: Schematic of a SAR ADC with offset errors.

speed.

$$V_+ = -V_{IN} + \sum_{i=0}^{N-1} 2d_i \cdot \frac{2^i C_u}{C_{Tot}} \cdot V_{REF} - \frac{C_u}{C_{Tot}} \cdot V_{REF} - V_{COMP,OS} - \frac{Q_+}{C_{Tot}} \quad (2.10)$$

2.4 Dynamic Error Sources in SAR ADCs

When analyzing the static error sources, we assume that during the SAR operations, each conversion is given enough time for V_+ to completely settle within the necessary resolution. In reality, conversion errors can occur when the comparator makes its decision before V_+ settles adequately. Because it is using a binary search and each analog input always maps to one distinct digital output code, errors made during the conversion process cannot be recovered at the end of the search process. As a result, to ensure correct operation, it is essential that each comparison is made correctly during the conversion process. The RC settling of the DAC sets the minimal time that needs to be allocated for each conversion step and therefore also sets the maximum operation speed of the ADC. Equation 2.12 gives the required time for an N -bit ADC to settle within $0.5 \times LSB$, where $\tau = R_{Tot} \cdot C_{Tot}$, R_{Tot} is the total resistance of the switches and C_{Tot} is the total capacitance of the DAC. To improve speed of operation,

small R_{Tot} should be used in the design.

$$V_{REF} \times e^{-t/\tau} \geq \frac{V_{REF}}{2^{N+1}} \quad (2.11)$$

$$t \geq \tau \times \ln(2^{N+1}) \quad (2.12)$$

Other factors in a practical design can also affect the conversion accuracy and further lengthen the settling time requirement. In our previous analysis, we assume that all voltage sources, V_{REF+} , V_{REF-} and ground, are ideal with zero impedance. However, in real implementation, the references have non-zero finite impedance. In addition, if the references are off-chip, they have to go through the bond wires and IO-pads, which can introduce both parasitic capacitance and inductance. When the DAC is switching, capacitors can be connected or disconnected from the references for charging or discharging, and this could introduce voltage spikes/ringing due to the parasitic inductance (L_0) and parasitic capacitance (C_0). The ringing frequency is approximately equal to the resonance frequency of the LC tank, $1/(2\pi \times \sqrt{L_0 C_0})$. Depending on how large is the real part of the impedance (or the Q of the resonance circuit), the ringing will decay at different rates.

This is especially problematic for the first few MSB transitions due to their potential large voltage and current swings. These problems are difficult to model because it is highly dependent on packages. On-chip input and reference buffers provide isolation between the on-chip circuit and the external world, but due to the increasing bandwidth requirement of modern ADC design, the approaches add significant design complexity and power consumption. Our goal in the next few chapters is to describe the techniques we developed in this research in order to combat these challenges.

Chapter 3

Redundancy in SAR ADCs

In Chapter 2, we discussed the operation of a traditional binary weighted SAR ADC. Even though it has many architectural advantages, such as its unparalleled energy efficiency, small chip size, amenability to digital scaling, rail-to-rail input swing, 100% capacitance utilization for input sampling, and ease of implementation, its resolution and speed are still limited by a few key design challenges that need to be resolved. The main linearity and performance limiting factors are capacitor mismatches and incomplete reference voltage settling due to high switching activities.

In this chapter, we introduce and analyze the redundancy algorithm in SAR ADCs and see how it can help mitigate these limitations discussed previously. We begin the chapter by giving a conceptual overview of SAR redundancy and discuss its benefits in terms of speed and achievable resolution over the traditional binary search algorithm. We can see that having redundant bits provides the extra leverage during the search process so that conversion errors in the earlier steps can be corrected later. In the second part of this chapter, we establish that redundancy can provide the necessary digital calibratability to calibrate out the mismatches in the capacitor array. The expected random mismatches within the capacitors determine the amount of redundancy that is necessary to cover this variation. We analyze and build the relationship between the two parameters. In the last part of this chapter, we analyze under what conditions redundancy can help improve sampling rate in a SAR ADC. Although, redundancy has been used in the past to improve sampling rate, we will show in this

section that redundancy is only beneficial in certain cases. It does not always help improve sampling rates.

3.1 Redundancy Overview

In Chapter 2, we've described the binary search algorithm and demonstrated the quantization process for a particular input level ($V_{IN} = 6.2$). We previously mentioned that in a binary search process, no conversion errors can be tolerated. This is because for every analog input value, there is a unique corresponding digital output code. Once a decision error is made, the ADC cannot recover and produce the correct output codes due to its one-to-one mapping property. This becomes even clearer from Figure 3-1. The plot highlights the decision levels, search range, and search sequence for a 4-bit binary-weighted SAR ADC. The x -axis indicates the sequences of binary search and the y -axis shows the full search range. The plot shows that none of the ranges within the same search cycle overlaps, meaning that once a range is eliminated during the searching process, the range is dropped from the search procedure and it will never be reconsidered again. This confirms our previous conclusion that errors made during the conversion process cannot be corrected in a binary search.

The search presented in Figure 3-1 has no error tolerance capability, but does suggest that if the search ranges within the same cycle do overlap, the already dropped search range can potentially be recovered and produce the correct digital output. For overlapped search ranges, a less than radix-2 (sub-binary) search is needed. Essentially, a sub-binary search takes more than N steps to convert an analog input into a N -bit digital output. Even though this search algorithm is less efficient in terms of the number of steps required to reach a certain resolution, it provides the necessary tolerances to boost the robustness of the overall operation.

Figure 3-2 compares the two search algorithms. Here, $s(i)$'s represent the step sizes during the search process. In an N -bit binary weighted algorithm, $s(i)$'s are binary weighted with values 2^{N-i} , where i is between 0 and $N - 1$. On the other hand, a SAR ADC with redundancy requires M steps to realize N -bit digital output,

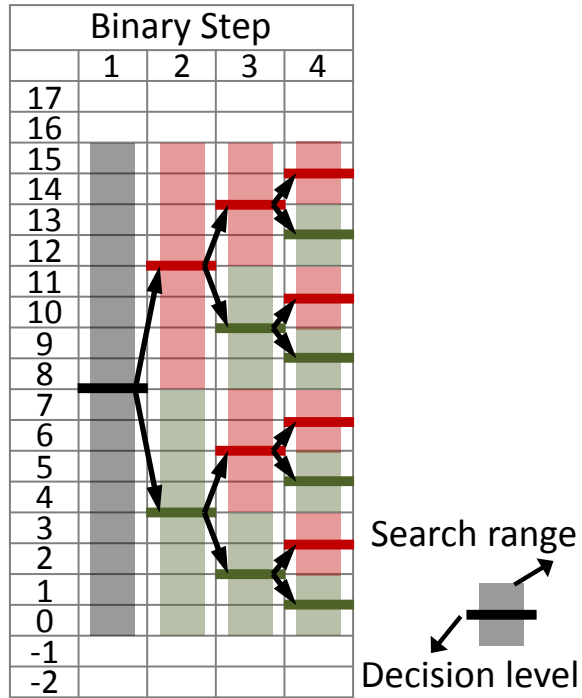


Figure 3-1: Binary search algorithm without redundancy. The search step sizes in this example are binary weighted with values equal to 8, 4, 2 and 1.

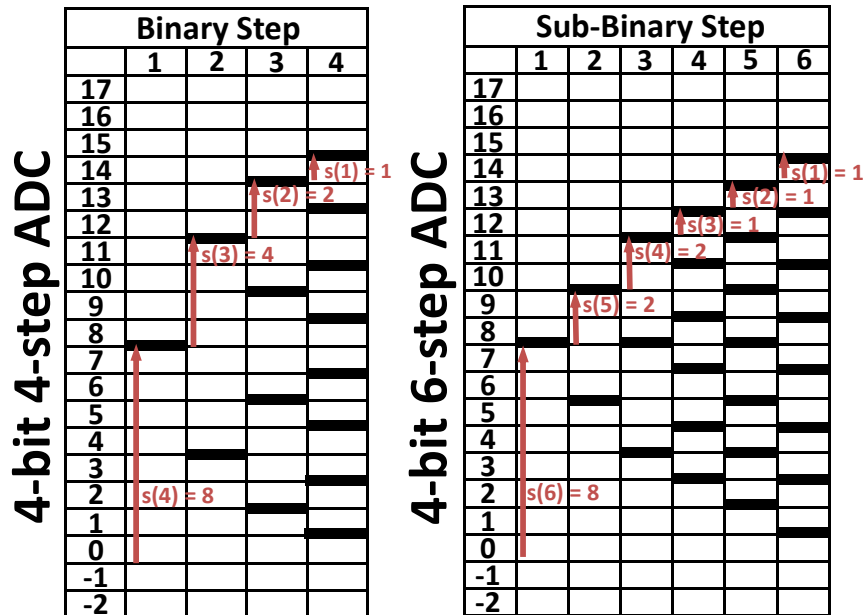


Figure 3-2: Comparison of using a traditional binary search algorithm (4-bit 4-step) and a sub-binary search algorithm (4-bit 6-step). Black decision levels indicate that in each step, transitions to the nearest decision levels above and below the current step level are possible.

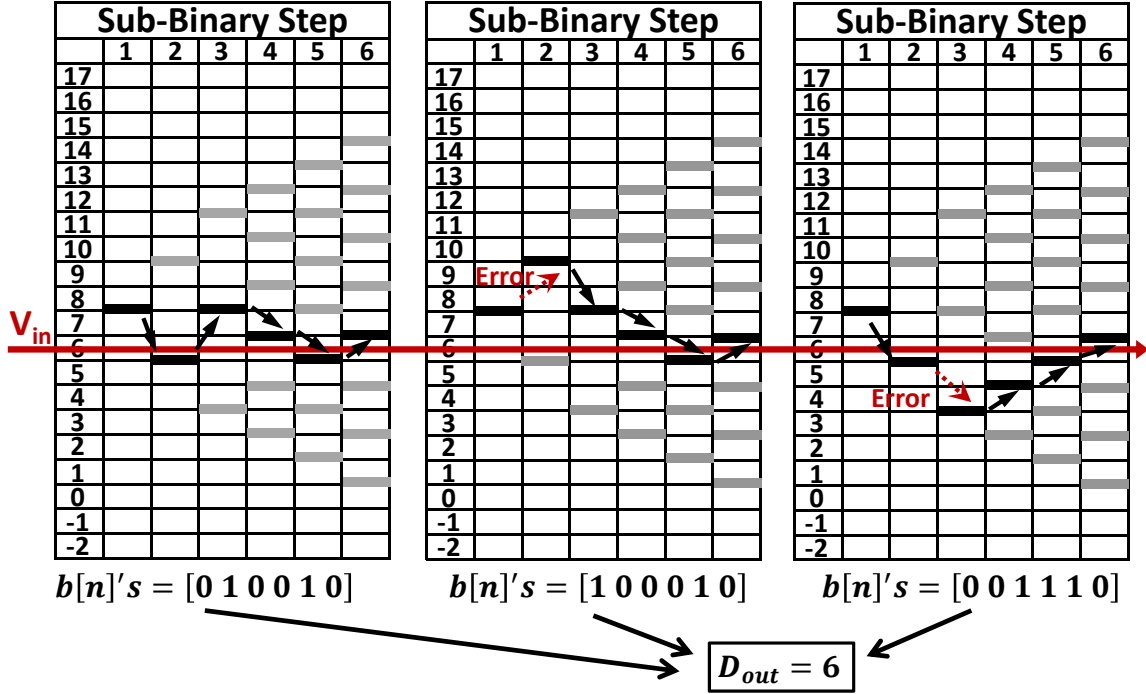


Figure 3-3: Digital error correction using redundancy in SAR ADCs. We've seen that even though the digital output bits are different in all three cases, they all represent the same D_{out} .

where $M > N$. As an example, in Figure 3-2, the binary case only requires four steps with binary weighted $s = [8, 4, 2, 1]$, while the sub-binary case requires six steps to achieve the same resolution with $s = [8, 2, 2, 1, 1, 1]$. s adds up to 15 in both cases, implying that the two algorithms have identical search range. The final digital output for an N -bit M -step ADC can be calculated using Equation 3.1.

$$D_{out} = s(M) + \sum_{i=1}^{M-1} [2 \cdot b[i] - 1] \times s(i) + [b[0] - 1] \quad (3.1)$$

where D_{out} is the final digital output expressed in decimals, $b[n]$ is the i^{th} digital output bit, N is the effective resolution and M is the total number of steps. In our example, the extra two steps added to the original binary search provide tolerance to bit decision errors; Figure 3-3 gives an example demonstrating this error resilience.

The left-most plot in Figure 3-3 shows an ideal example where all decisions are made correctly during the conversion process; the middle plot shows an example

where a decision error is made in the first step during the conversion process; finally, the right-most plot shows an example in which a decision error is made in the second step during the conversion process. For $V_{IN} = 6.2$, each of these gives different digital output bit sequences: [010010], [100010] and [100010], respectively. Their digital outputs can be calculated according to Equation 3.1 and they all result in the same $D_{out}(= 6)$ as shown in Equation 3.2, 3.3 and 3.4. This demonstrates that redundancy has the capability to digitally correct for at least some bit decision errors.

$$\begin{aligned} [010010] \mapsto & 8 + (2 \cdot 0 - 1) \times 2 + (2 \cdot 1 - 1) \times 2 + (2 \cdot 0 - 1) \times 1 \\ & + (2 \cdot 0 - 1) \times 1 + (2 \cdot 1 - 1) \times 1 + (0 - 1) = 6 \end{aligned} \quad (3.2)$$

$$\begin{aligned} [100010] \mapsto & 8 + (2 \cdot 1 - 1) \times 2 + (2 \cdot 0 - 1) \times 2 + (2 \cdot 0 - 1) \times 1 \\ & + (2 \cdot 0 - 1) \times 1 + (2 \cdot 1 - 1) \times 1 + (0 - 1) = 6 \end{aligned} \quad (3.3)$$

$$\begin{aligned} [001110] \mapsto & 8 + (2 \cdot 0 - 1) \times 2 + (2 \cdot 0 - 1) \times 2 + (2 \cdot 1 - 1) \times 1 \\ & + (2 \cdot 1 - 1) \times 1 + (2 \cdot 1 - 1) \times 1 + (0 - 1) = 6 \end{aligned} \quad (3.4)$$

3.1.1 Error Tolerance Windows for Redundancy

Even though redundancy gives the SAR algorithm additional tolerance to decision errors, it does not provide unlimited amount of error tolerance. If the decision errors are too large during the conversion process, even with redundancy, the errors still cannot be recovered and incorrect digital outputs will be generated. As a result, for each conversion step, a range of recoverable analog voltage can be highlighted around the decision level. This implies that during the transition, if an analog voltage falls within this range and error is made, the ADC can recover from the errors if no mistakes are made in the rest of the conversion process. We denote this error tolerance window as ϵ_t . For the n^{th} output bit, $\epsilon_t(n)$ can be calculated according to Equation 3.5.

$$\epsilon_t(n) = \sum_{i=1}^{n-1} s(i) - s(n-1) \quad (3.5)$$

As an example, Figure 3-4 shows a redundant SAR ADC with $s = [8, 2, 2, 2, 1]$.

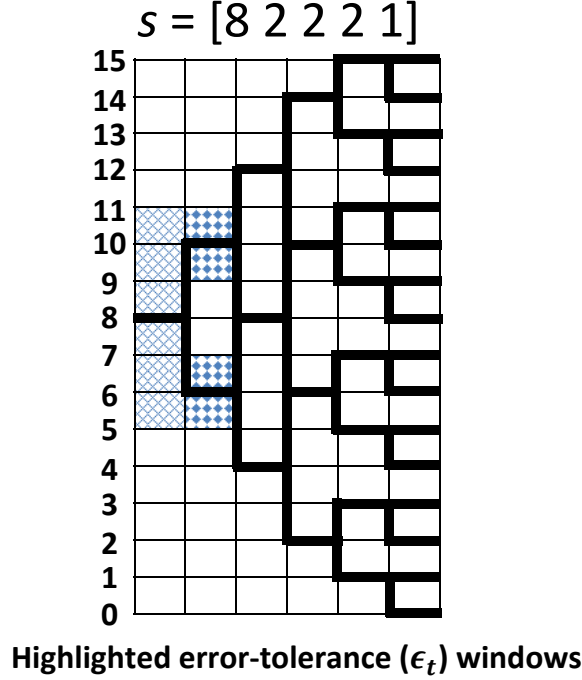


Figure 3-4: Highlighted error tolerance windows (ϵ_t) for a sub-binary search SAR ADC. The error tolerance windows are as follows: $\epsilon_t(5) = \pm 3$, $\epsilon_t(4) = \pm 1$, $\epsilon_t(3) = 0$, $\epsilon_t(2) = 0$ and $\epsilon_t(1) = 0$.

For the 5th output bit, the error tolerance window is given by Equation 3.6.

$$\epsilon_t(5) = s(3) + s(2) + s(1) - s(4) = 2 + 1 + 1 - 2 = 3 \quad (3.6)$$

This quantity implies that for the 5th output bit (or the 1st conversion step), an error can be tolerated as long as the input voltage sits between 5 ($= 8 - 3$) and 11 ($= 8 + 3$). The formula can be intuitively understood as follows. For the n^{th} output bit, once a decision is made, the next decision level will either move up or down by the step size of $s(n - 1)$. If this decision is erroneous, then the sum of the follow-on step sizes, $s(n - 2), s(n - 3), \dots, s(1)$, must be large enough and exceed the value of the current step size to counteract this mistake. The exceeded amount is the tolerance window for that decision level.

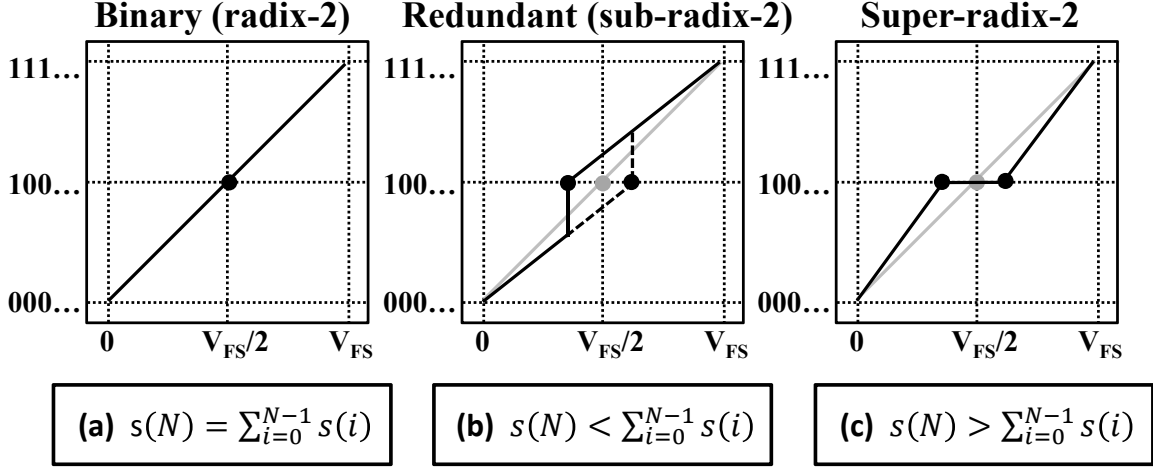


Figure 3-5: Transfer functions for SAR designs with step sizes that are binary, sub-radix-2 and super-radix-2 weighted.

3.2 Digital Calibratability

In the last section, we discussed how redundancy can help resolve dynamic conversion error. In the section, we explore the condition of digital calibratability in the presence of static mismatches in capacitors. These mismatches lead to mismatches in the searching steps, $s(n)$.

3.2.1 Condition of Digital Calibratability

Three transfer functions are shown in Figure 3-5. The leftmost plot shows the ideal transfer function in which the analog input is linearly mapped to digital output code. For example, an zero input is converted to digital output code $000 \dots 0$, the maximum input (V_{FS}) is converted to digital output code $111 \dots 1$ and $V_{FS}/2$ is converted to $100 \dots 0$ or $011 \dots 1$. The other two plots in the same figure show specific variations from the ideal transfer function. Figure 3-5 (b) has the MSB step size smaller than its nominal value and Figure 3-5 (c) has the MSB step size larger than its nominal value. As defined previously, one is referred as the sub-radix-2 search and the other one is referred as the super-radix-2 search. In a super-radix-2 search, a horizontal misalignment (missing level) appears in the transfer function. This shows that multiple analog inputs are mapped to the same digital output code, in this

case, the analog information is lost and the errors cannot be corrected digitally. In contrast, in a sub-radix-2 search, vertical misalignments (missing codes) appear in the transfer function. In this case, there are missing digital output codes; one analog input could potentially be mapped to more than one digital output codes while some of the digital output codes never show up during normal operations. This error is digitally correctable since the analog information is not lost in this case. For example, by linearly shifting the upper-segment of the curve down to align it with the lower-segment of the curve in Figure 3-5 (b), the large vertical jump in the transfer function is removed and the errors due to mismatches are digitally corrected. The large vertical jump is embodied in the redundant search algorithm. By designing $s(N)$ intentionally larger than the sum of the remaining $s(n)$, we can create digitally correctable codes.

This idea can be extended into every search step in the sub-binary search to build redundancy into all decision levels

$$\sum_{i=0}^{n-1} s(i) - s(n) \geq 0, \quad (3.7)$$

where $i = 1, 2, \dots, N$. As long as all decision levels satisfy Inequality 3.7, there are no missing levels and all static errors are digitally correctable. We see that the analysis on static error calibration here leads to the same conclusion as the one we derived for dynamic error correction in the previous section, and redundancy is an effective method to mitigate both problems at once.

3.2.2 Amount of Redundancy

From the previous discussion, we understand that when step sizes satisfy Inequality 3.7, we have redundancy built into the search algorithm. One simple way to achieve this inequality is by choosing a fixed radix α that is less than 2. Since the step size in a real implementation is proportional to the size of the capacitors in the DAC and the capacitors experience random manufacturing variation, it is expected to see variation in the search step as well. Even though the design is originally built to satisfy Equation 3.7, the added variation can break this relationship and create

missing levels that are not digitally correctable. In this section, we discuss and establish a relationship that determines the amount of redundancy needed to guarantee Inequality 3.7 with respect to different amounts of DAC capacitance variation.

When the ADC is designed with a fixed radix, α , we have the following relationship

$$\alpha = s(i)/s(i - 1) \leq 2 \quad (3.8)$$

where $i = M - 1, M - 2, \dots, 1$. Since the radix is less than 2 in this case, the required number of conversion steps, M , to complete a conversion is more than the resolution N . The effective number of bits, N , can be calculated using Equation 3.9,

$$N \leq \log_2 \frac{s_{Tot}}{s(0)} = \log_2 \frac{\alpha^M + \alpha - 2}{\alpha - 1} \quad (3.9)$$

where s_{Tot} is the sum of all the step sizes, N is the effective number of bits and M is the total number of conversion steps. Figure 3-6 shows that converters with smaller radix, α , require more steps to achieve the same resolution as the converter with larger radix; however, converters with smaller radix has built-in redundancy and is more resilient against both dynamic and static conversion errors.

3.2.3 Radix and the Number of Steps

In order to incorporate redundancy to improve robustness in dynamic operation and to provide the capability to digitally calibrate for static random mismatches, Inequality 3.7 must be satisfied at all times even with the presence of variation. Due to manufacturing variation, when using capacitors with small dimensions, random variation is unavoidable. Since the step sizes ($s(M), s(M - 1), \dots, s(0)$) are proportional to the capacitor sizes (C_M, C_{M-1}, \dots, C_0), Equation 3.7 can be re-written as follows

$$\sum_{i=0}^{n-1} C_i - C_n \geq 0 \quad (3.10)$$

where $C_i = \alpha^i \times C_0$ is the desired (or designed) relationship between the capacitances

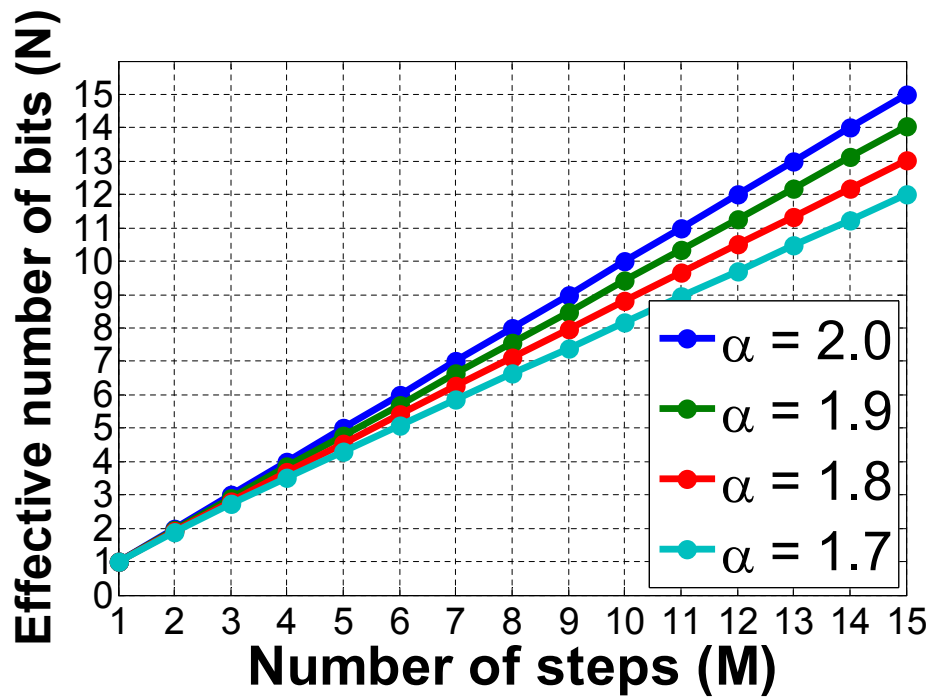


Figure 3-6: Effective number of bits (N) versus number of steps (M) for different radices (α). Converters with smaller α require additional conversion steps to achieve the same effective resolution, but they have more built-in redundancy against dynamic and static conversion errors.

in the DAC. Manufacturing variation in C_i 's can break this relationship. Our goal is to find the appropriate radix number and the number of steps such that Inequality 3.10 is satisfied with high probability, even in the face of variation.

Let us assume that the mismatches in the capacitors are independent and the unit capacitor observes a Gaussian distribution in capacitance with the mean equal to $C_{0,mean}$ and the variance equal to $\sigma_{C_0}^2 \times C_{0,mean}^2$. Then the mean and variance of the i^{th} capacitor are $\alpha^i \times C_{0,mean}$ and $\alpha^{2i} \times \sigma_{C_0}^2 \times C_{0,mean}^2$, respectively, assuming each C_i is composed of α^i of these unit capacitors. The left side of Equation 3.10 is a sum of independent Gaussian variables and observes an overall normal distribution with mean and variance specified below:

$$\mu = \frac{(2 - \alpha)(\alpha^n - 1)}{\alpha - 1} \times C_{0,mean} \quad (3.11)$$

$$\sigma^2 = \sigma_{C_0}^2 \times \frac{\alpha^{2(n+1)} + \alpha^2 - 2}{\alpha^2 - 1} \times C_{0,mean}^2 \quad (3.12)$$

For Equation 3.10 to be true with a probability of 99.865% (+3 σ of the normal cumulative distribution function), the following inequality needs to be true,

$$\mu - 3\sigma \geq 0 \iff \frac{(2 - \alpha)(\alpha^n - 1)}{\alpha - 1} - 3 \times \sqrt{\sigma_{C_0}^2 \times \frac{\alpha^{2(n+1)} + \alpha^2 - 2}{\alpha^2 - 1}} \geq 0. \quad (3.13)$$

By looking at the equation, we see that once the variance of the capacitor ($\sigma_{C_0}^2$) is known, an appropriate radix α can be chosen so that Inequality 3.13 is satisfied for every capacitor in the DAC. The variance $\sigma_{C_0}^2$ can typically be found in the design manual provided by individual foundries. In theory, all n from $n = M$ to $n = 1$ have to satisfy this inequality; however, here, we only check whether this equation is held for the first few MSB capacitors because they are the main contributors to DNL and INL errors. In this case, we make another assumption that $\alpha^n \gg 1$. This is an appropriate assumption because typically, redundancy is only needed for high resolution ADCs, in which designs are very sensitive to capacitor mismatches. For lower resolution ADCs, manufacturing matching may already be sufficient. With this

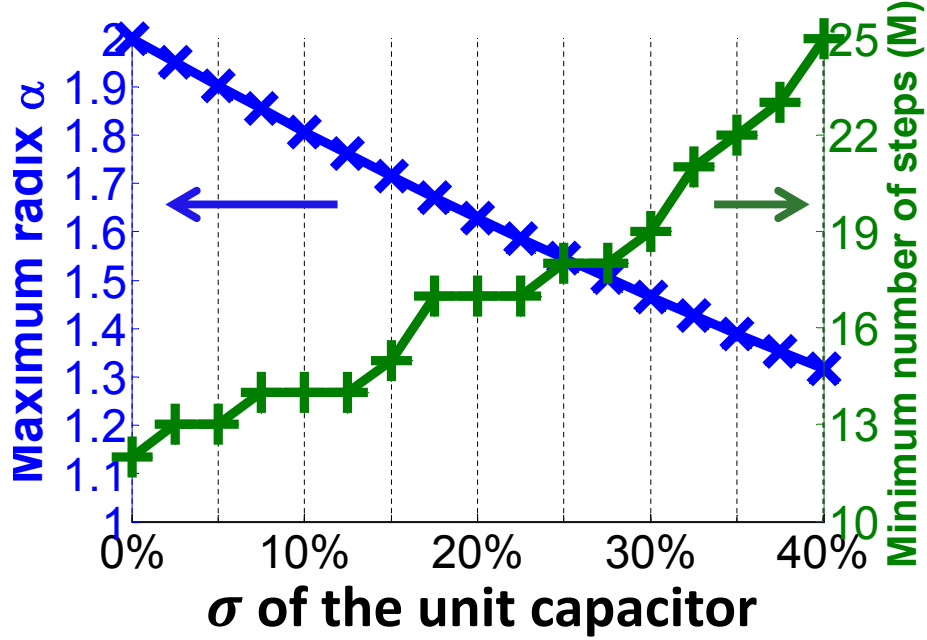


Figure 3-7: The maximum radix α and the minimum number of conversion steps M versus the standard deviation of the unit capacitor, in order to achieve digital calibratability in a 12-bit ADC.

additional assumption, Inequality 3.13 can be simplified to

$$\frac{2 - \alpha}{\alpha - 1} - 3.0 \times \sigma_{C0} \frac{\alpha}{\sqrt{\alpha^2 - 1}} \geq 0 \quad (3.14)$$

Using Inequality 3.14, for a given σ_{C0} we can calculate the maximum α we need to build a capacitive DAC with guaranteed digital calibratability. Now we have σ_{C0} from the design manual, targeted resolution N and the calculated α ; the only remaining parameter we need is the number of conversion steps M . This should be chosen such that Equation 3.9 is satisfied with high probability to prevent yield loss. Using a similar approach as before, we can calculate the mean and the variance of the terms on the left hand side of Equation 3.9 and subsequently, find the condition when this equation is true with probability 99.865%. This leads to Inequality 3.15.

$$\frac{\alpha^M + \alpha - 2}{\alpha - 1} - 2^N - 3 \times \sigma \times \sqrt{\frac{\alpha^{2N} + \alpha^2 - 2}{\alpha^2 - 1}} \geq 0 \quad (3.15)$$

These inequalities agree with our intuition that larger capacitor mismatches re-

quire smaller radix (α) and a larger number of conversion steps. Figure 3-7 shows a plot of maximum radix and the minimum number of conversion steps needed for a given amount of capacitor mismatches in a 12-bit ADC. From the example, we see that when σ is 0%, $\alpha = 2.0$ and $M = 12$; this corresponds to the classic non-redundant binary search ADC case. On the other hand, when σ is 40%, $\alpha = 1.186$ and $M = 37$. This shows that it takes three times the number of conversion steps when $\sigma = 40\%$ compared to when $\sigma = 0\%$ in order to maintain the same digital calibratability.

3.3 Redundancy and its Speed Benefit

In the past, there has been a common belief that even though redundancy requires a larger number of steps to complete the conversion process compared to binary search, redundancy could improve the overall speed because the first few MSBs do not have to be completely settled within 0.5LSB, as errors can be corrected by later steps. As a result, it is believed that each step can take significantly less time compared to the binary case, and in aggregate, the total conversion time can be reduced even though more steps are required. In this section, we will challenge this belief by analyzing the effectiveness of redundancy in relationship to DAC settling time, latch delay, digital logic delay and sampling rate in SAR ADCs. Behavioral models of SAR ADCs are developed which run at a speed that is four orders of magnitude faster than simulations done in FastSPICE, to predict ADC time progression and to quickly identify the maximum sampling rates that can be used in both redundant and non-redundant cases. The result obtained from behavioral-model simulation shows good matching with the result from SPICE simulation. Using these, we are able to show that redundancy does not always improve sampling rate; instead, the maximum sampling rate depends on the relative magnitudes of different ADC delay components. More importantly, this analysis provides guidance in ADC design with redundancy such that overall performance can be improved.

3.3.1 Prior Work

Kuttner in [48] uses a sub-binary scaling of the DAC capacitors to introduce redundancy in each bit decision cycle; a reconfigurable capacitor array is built to vary the amount of redundancy in SAR ADCs. Kuttner shows improvement in converter accuracy as the redundancy increases from 0% to $\pm 6.4\%$ and $\pm 12.7\%$ for a fixed sampling rate and a fixed clocking frequency, leading to the conclusion that redundancy increases sampling rate. For a fixed clocking frequency, a redundant ADC requires extra cycles to complete the conversions compared to an ADC without redundancy; as a result, it would be an unfair comparison if a fixed sampling rate is assumed for both cases. Here, we extend and improve the previous analysis by taking into account the effect of extra clock cycles in redundant SAR ADCs by comparing maximum sampling rates.

Ogawa *et al.* in [19] show that redundancy can be used to increase sampling rate, by analyzing the relationship between the achievable sampling rate and different redundancy patterns; however, the analysis is specific to the case where the conversion time is dominated by capacitive DAC settling. Other factors such as the settling time in the latch preamplifier or the latch delay can impact the effectiveness of redundancy in increasing sampling rate. In this section, these issues are addressed by analyzing the role of redundancy in a more general scenario.

3.3.2 Behavioral Models

A behavioral model of a SAR ADC is depicted in Figure 3-8. The model comprises three main delay components: latch (T_C), logic (T_L) and DAC settling (T_D) delays. The latch delay refers to the delay between the enable signal and the output of the latch. T_C , in this case, only includes the delay through the regenerative latch but not the delay through the latch pre-amplifier (if one is placed in front of the latch). The logic delay refers to the delay through the control logic block, and the DAC delay refers to the delay through capacitive DAC settling. The delay through the latch pre-amplifier can be lumped into T_D because both delay components contribute to

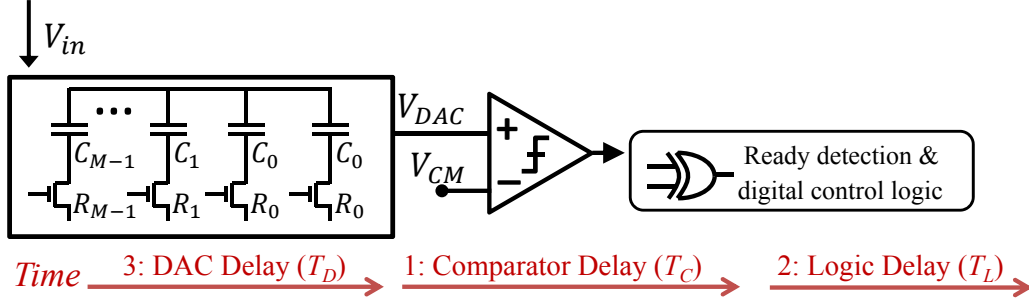


Figure 3-8: Behavioral model of a SAR ADC. The critical delay path is divided into three components: the latch delay (T_C), the logic delay (T_L) and the DAC settling delay (T_D).

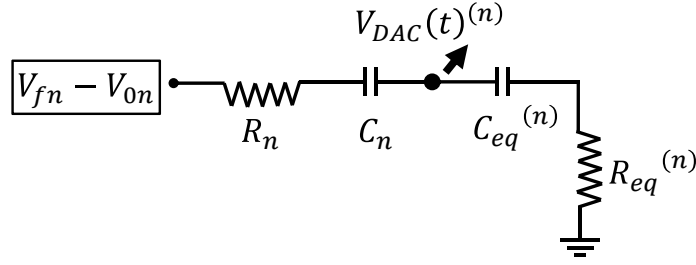


Figure 3-9: Behavioral model when the i^{th} capacitor in the DAC is being charged or discharged.

incomplete settling on the inputs of the latch.

Figure 3-9 provides a representative scenario of a DAC charging condition for the n^{th} capacitor in the DAC array. C_n and R_n represent the capacitance and the series resistance of the MOS switch, respectively, associated with the n^{th} capacitor. $C_{eq}^{(n)}$ and $R_{eq}^{(n)}$ represent the capacitance and the series resistance, respectively, looking into the capacitive DAC, excluding C_n and R_n . $V_{DAC}(t)^{(n)}$ is the voltage contribution from the n^{th} capacitor on the DAC at time t , V_{fn} (and V_{0n}) represent the final (and initial) voltage on the bottom plate of the n^{th} capacitor. Equation 3.16 and 3.17 model the T_D delay as a first-order RC circuit, where t_{0n} represents the time when the n^{th} capacitor begins charging (or discharging). The DAC output voltage at time t can be calculated using Equation 3.18.

$$V_{DAC}(t)^{(n)} = \frac{R_{eq}^{(n)}}{R_n + R_{eq}^{(n)}} \times \left[1 + \left(1 - e^{-\frac{t-t_{0n}}{\tau}} \right) \frac{C_n R_n - C_{eq}^{(n)} R_{eq}^{(n)}}{R_{eq}^{(n)} (C_n + C_{eq}^{(n)})} \right] (V_{fn} - V_{0n}) \quad (3.16)$$

$$\tau = \frac{C_n + C_{eq}^{(n)}}{C_n C_{eq}^{(n)} (R_n + R_{eq}^{(n)})} \quad (3.17)$$

$$V_{DAC}(t) = V_{DAC}(0) + \sum_{n=1}^N V_{DAC}(t)^{(n)} \quad (3.18)$$

Equation 3.16 can be simplified into Equation 3.19 to approximate the DAC settling behavior for conservative estimation of settling time. Equations 3.16 and 3.19 share the same RC time constant, τ , but differ in their initial conditions. The initial voltage in the simplified model is 0, while the initial voltage in the more complex model is $\frac{R_{eq}^{(n)}}{R_n + R_{eq}^{(n)}} \times (V_{fn} - V_{0n})$. We see that the predicted time-to-settle is less using the complex model, compared to using the simpler model. In principle, the settling transient can be completely eliminated if resistance and capacitance are sized inversely proportional to each other such that $C_n R_n = C_{eq}^{(n)} R_{eq}^{(n)}$ for all n 's. In reality, it is difficult to achieve such perfect matching because the on-resistance of MOSFET switches changes with the operating condition of the circuit. Therefore, Equation 3.16 typically produces results that are too optimistic and Equation 3.19 is a better approximate model for a real DAC.

$$V_{DAC}(t)^n = \left[\left(1 - e^{-\frac{t-t_{0n}}{\tau}} \right) \right] \times \frac{C_n}{C_n + C_{eq}^n} \times (V_{fn} - V_{0n}) \quad (3.19)$$

The latch delay, T_C , is modeled by Equation 3.20, where ω_0 is the bandwidth of the latch and V_{in} is the input voltage of the latch when the enable signal goes high. When V_{in} of the latch is too small, the latch can go into metastability and is not able to resolve the output to logic levels; in this case, we force the output to logic 0 after time $0.75 \times T_S$. The delay through the control logic block, T_L , is modeled as a constant, T_{L0} .

$$T_C = \begin{cases} \omega_0 \times \ln \frac{V_{ref}}{2 \times V_{in}} & T_C < 0.75 \times T_S \\ 0.75 \times T_S & \text{otherwise} \end{cases} \quad (3.20)$$

3.3.3 Effectiveness of Redundancy

The analysis to evaluate the effectiveness of redundancy in SAR ADCs is done for two different cases: one case assumes that the DAC settling time (T_D) dominates the total allowed settling period (T_S), and the other case assumes that the latch delay (T_C) takes up the majority of the total allowed settling period. The ADC decision errors introduced by large T_D are due to incomplete settling on the capacitive DAC array: the latch output would have been different if the capacitive DAC had settled completely. The ADC decision errors introduced by large T_C are due to latch metastability, where small input differences cannot be resolved by the latch within $0.75 \times T_S$. In this case, the latch output is forced to logic 0, resulting in a wrong decision made by the latch.

In the following examples, we focus on the comparison between a 10-bit non-redundant SAR ADC and a 10-bit 15-step redundant SAR ADC with search steps $s = [512, 106, 150, 95, 59, 38, 24, 15, 9, 6, 4, 2, 1, 1, 1]$ (\approx radix-1.6). The simulation uses a differential SAR ADC with unit capacitor $C_u = 10fF$, $V_{ref} = 1V$ and $V_{CM} = 0.5V$.

Case 1: T_D Dominates

In the first analysis, the DAC settling delay T_D is the dominant portion of the total allowed settling period (T_S), T_C is set to a constant value at $50ps$ and T_L is set to $0ps$ in order to focus on the impact of T_D . The analysis, shown in Figure 3-10, is done in two dimensions, sweeping the allowed period (T_S) to find the maximum sampling rate and the time constant, τ , of the DAC. Each rectangle represents the integrated INL errors over the full digital range of an ADC configuration. For an N -bit ADC, we test it by putting in 2^N analog inputs that correspond to the 2^N distinct digital outputs in the ideal case. We then subtract the actual digital output code from the ideal digital output code for each input, and sum the absolute of the differences to get the integrated INL errors. A functional ADC in this case is defined as an ADC that does not make any integrated INL errors. These functional operating conditions are highlighted in dark blue (with 0 integrated INL error) in all the figures presented

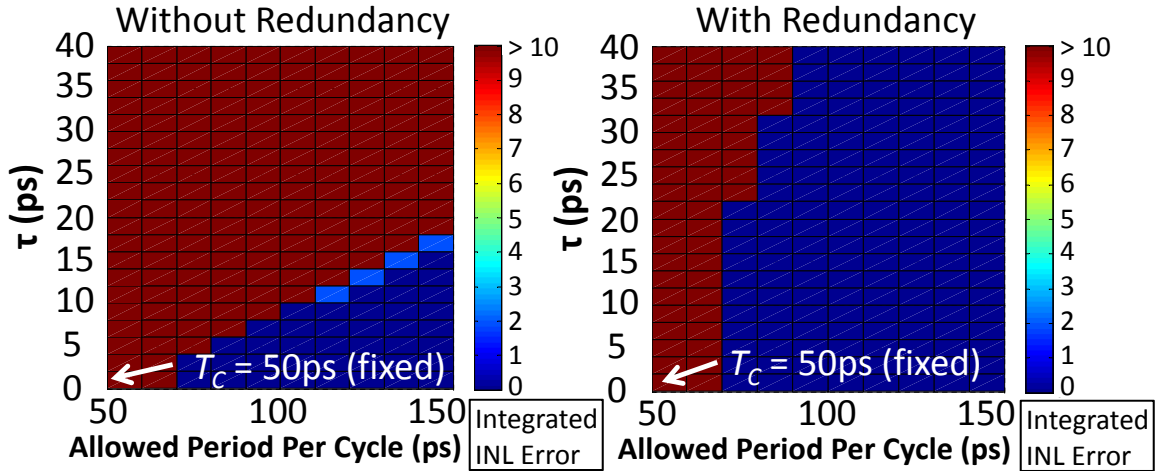


Figure 3-10: The effectiveness of redundancy in SAR ADCs when the delay through the DAC array (T_D) dominates.

this section.

The results in Figure 3-10 shows that redundancy can be used to reduce problems due to incomplete settling of the capacitive DAC, especially when τ is large. At small τ , the non-redundant SAR ADC can run at the same clock speed as the SAR ADC with redundancy, resulting in a faster sampling rate because it takes fewer clock cycles to complete the conversion. As τ increases, the non-redundant SAR ADC begins to fail at a higher clock speed when the allowed sampling period decreases compared to the one with redundancy. The difference between the minimum allowed period of the two cases continues to widen as τ increases, showing that redundancy can help improve sampling rate when T_D is the main delay contributor. For example, when $\tau = 15ps$ and $T_C = 50ps$, the minimum conversion time improves from $1400ps$ (10 cycles with $140ps/cycle$) in the non-redundant SAR ADC to $900ps$ (15 cycles with $60ps/cycle$) in the redundant one.

Case 2: T_C Dominates

In the second analysis, the latch delay is assumed to dominate T_D and T_L , so more careful modeling of this delay component is needed. In this case, T_C is set according to Equation 3.20 and T_L is set to 0. The impact of T_D is assessed by sweeping only the smaller values of τ , in order to study the relationship between the effectiveness

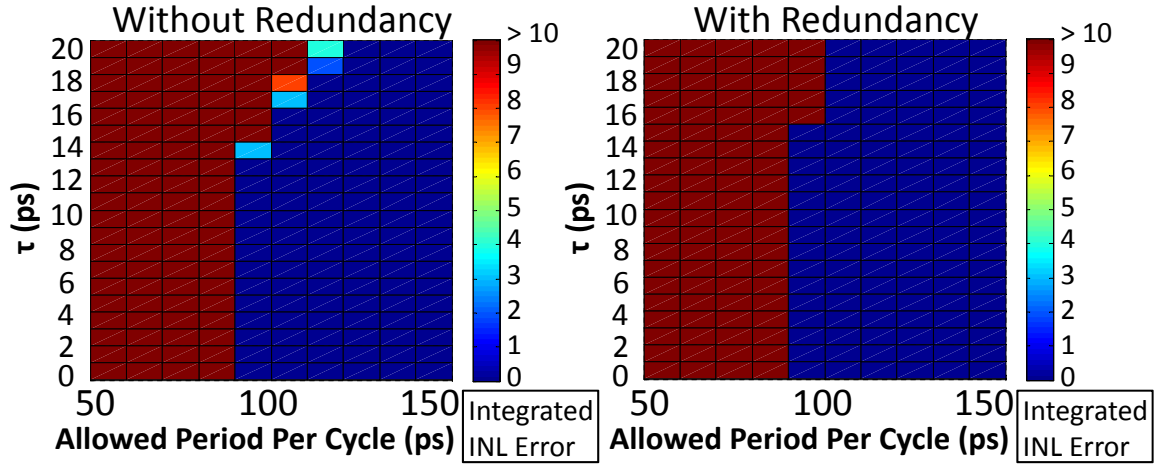


Figure 3-11: The effectiveness of redundancy in SAR ADCs when the delay through the latch (T_C) dominates the other delay components.

of redundancy and T_C .

Figure 3-11 shows that redundancy is not helpful in terms of reducing the problems due to the latch delay. In general, redundancy provides the needed additional steps for the outputs of the DAC to gradually approach zero. When there is latch metastability, the latch output is forced to logic 0, regardless of its input voltage. Even with the added redundancy, the output of the capacitive DAC is not driven towards zero because of the erroneous latch outputs. Figure 3-12 shows the number of runs (out of 1024) in which a metastability event happens. The strong correlation between Figure 3-11 and 3-12 shows that whenever a metastability event occurs, both the redundant and non-redundant cases produce erroneous digital values. Thus, correct operation of the ADC requires operation away from regions where latch metastability can occur, and thus redundancy provides little benefit.

Case Study with SPICE Simulation

To verify the behavioral models, 10-bit SAR ADCs with and without redundancy are designed in 65nm bulk CMOS technology and in behavioral models. The same s , C_u , V_{ref} and V_{CM} as in the previous simulations are adopted here. Bandwidth, ω_0 , of latch is designed to be 100GHz; average DAC time constant, τ , is 40ps; and delay through the digital control block, T_L , is estimated to be 225ps from SPICE

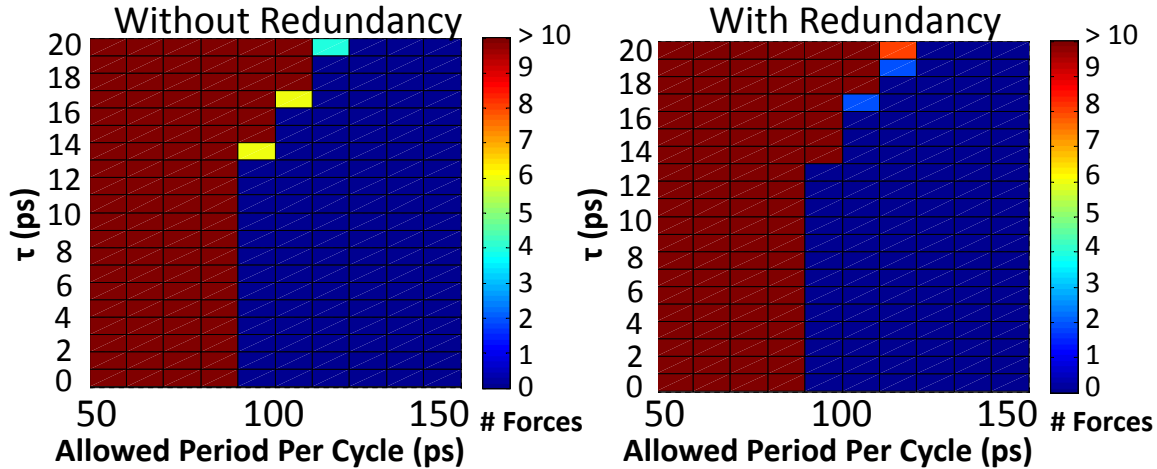


Figure 3-12: The number of metastability events when the delay through the latch (T_C) dominates the other delay components.

simulation. The architecture of the SAR ADC follows that described in Chapter 4.

Figure 3-13 shows the SPICE simulation results of SAR ADCs with and without built-in redundancy. The y-axis shows the total integrated INL errors and the x-axis shows the allowed or given period per cycle. The fastest clock period that an ADC can run is defined as the clock frequency when no integrated INL errors are made. As an example, when the allowed period per cycle is $400ps$, the non-redundant ADC has a total of 58 errors (non-functional), and the redundant SAR ADC makes no errors (functional) in Figure 3-13. Using this, we find that the fastest period that a non-redundant SAR ADC can run is $500ps$, and the fastest clock period that a redundant SAR ADC can run is $320ps$, with total conversion time of $5ns$ (10 clock cycles and $500ps/cycle$) and $4.8ns$ (15 clock cycles and $320ps/cycle$), respectively. Thus, for this particular example, the redundant ADC can run marginally faster. Figure 3-14 shows the same simulation done in the behavioral model. Since it takes significantly less time to simulate using the behavioral model, a two-dimensional plot, that allows us to sweep different time constants (τ), can be generated. We then take a horizontal slice at $\tau = 40ps$, which is the designed RC time constant used in our SPICE simulation, from Figure 3-14 and the result is plotted in Figure 3-15. The results in Figure 3-13 and 3-15 indicate good matching between the behavioral and SPICE simulations. This verifies that our behavioral model can accurately capture the important settling

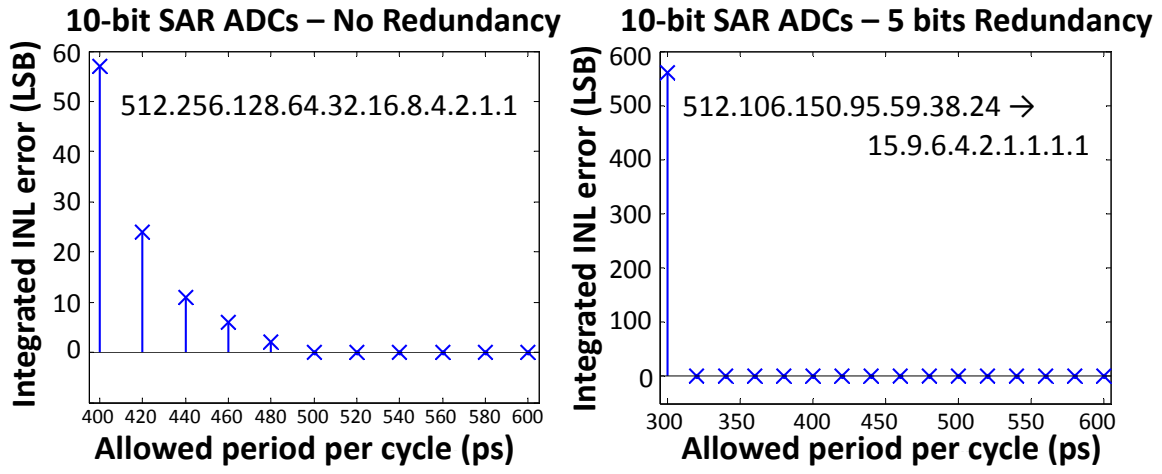


Figure 3-13: Effectiveness of redundancy in SAR ADCs (SPICE). The results show that the fastest clock period that a non-redundant SAR ADC can run is 500ps while the fastest clock period that a redundant SAR ADC can run is 320ps .

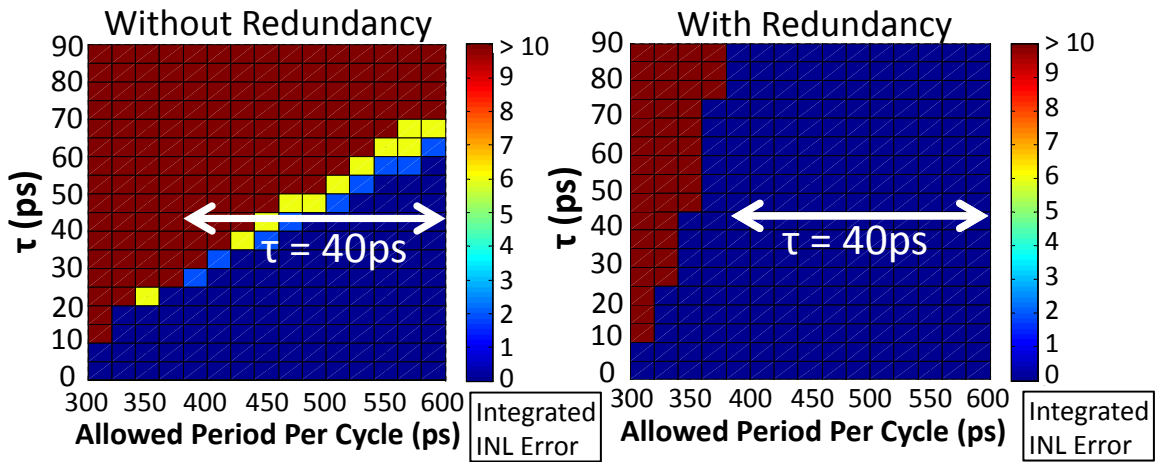


Figure 3-14: Effectiveness of redundancy in SAR ADCs using behavioral model simulation.

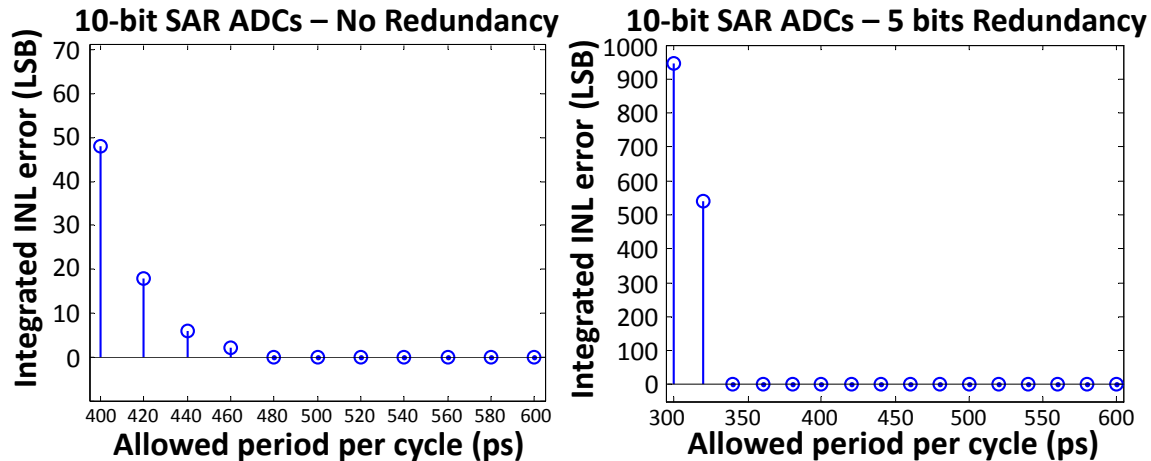


Figure 3-15: Effectiveness of redundancy in SAR ADCs using behavioral model simulation. This figure is a one-dimensional slice taken at $\tau = 40ps$ from Figure 3-14.

behavior.

The effectiveness of redundancy in terms of per-cycle and total ADC speed improvement is shown in the left and right plots of Figure 3-16, respectively. The left plot indicates that regardless of the values of the time constant (τ), a redundant design is always able to run at a faster per-cycle rate compared to a non-redundant design. The right plot indicates that even though a redundant design is always able to run at a faster per-cycle rate, it requires more cycles to complete the conversion process and therefore, only when the time constant is large enough, a redundant design has advantages over a non-redundant design in terms of the overall sampling rate. In this particular case, the time constant (τ) has to be greater than $50ps$ for the system to see improvement in overall sampling rate. The behavioral model developed in this section runs at 10,000 times faster compared to FastSPICE simulation. It provides a quick approach at the design stage, before doing lengthy simulations in SPICE, to accurately determine whether there is speed loss or speed improvement from using redundancy.

Additional SPICE simulation is done with a pre-amplifier added in front of the latch-based comparator, similar to the architecture presented in [48, 49]. A pre-amplifier is frequently placed in front of a latch comparator in order to provide buffers and reduce the effect of kickback noise. The additional settling delay introduced

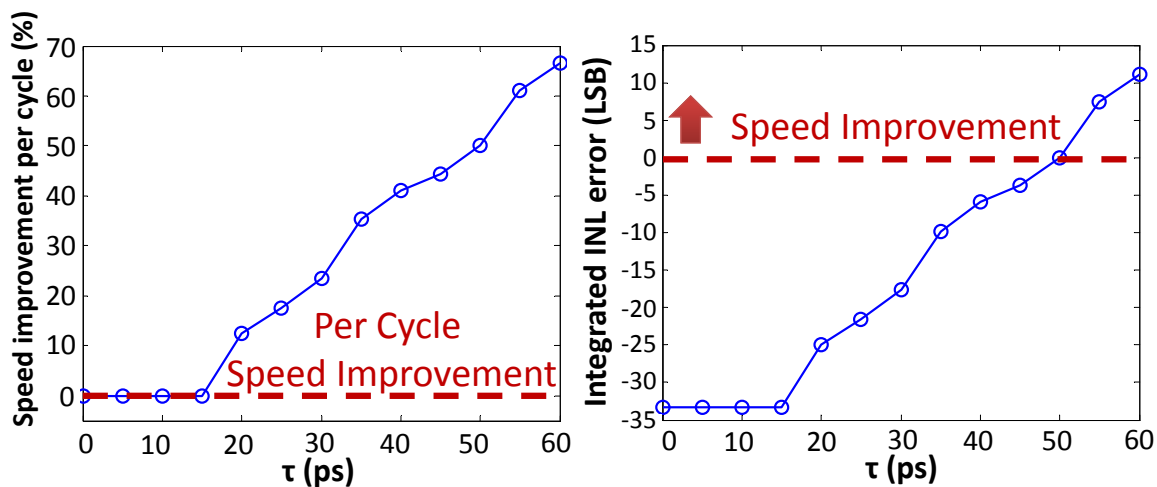


Figure 3-16: Speed improvement from adopting redundancy. The left plot shows per-cycle speed improvement and the right plot shows the overall speed improvement of the ADC.

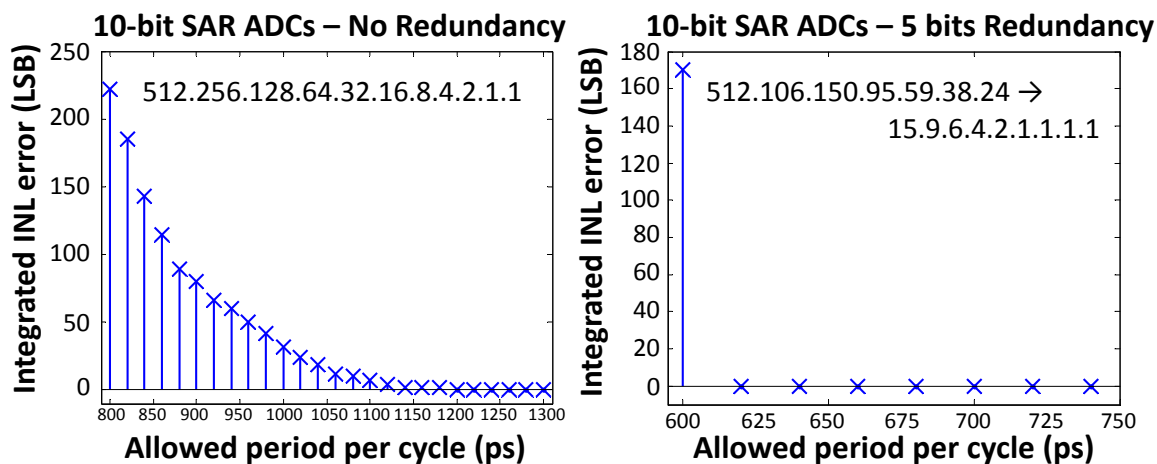


Figure 3-17: Effectiveness of redundancy in SAR ADCs with an added pre-amplifier in front of the latch comparator in SPICE simulation.

by the pre-amplifier is added to the T_D delay components discussed previously, and therefore, this effectively increases the total settling time τ . The increase in τ increases the effectiveness of redundancy in improving the overall sampling rate. As shown in Figure 3-17, the minimum per cycle period in a non-redundant and a redundant SAR ADC is $1200ps$ and $620ps$, respectively. This corresponds to the minimum total conversion time of $12ns(= 1200ps \times 10)$ and $9.3ns(= 620ps \times 15)$, respectively. In these ADC configurations, redundancy significantly increase maximum sampling rate.

In summary, depending on the relative magnitudes of different delay components, it is shown that redundancy is not always useful in increasing the overall sampling rate of SAR ADCs. When the latch delay dominates, redundancy is generally not helpful because it does not reduce latch metastability. When DAC settling time dominates, redundancy becomes increasingly useful as the DAC settling time becomes larger. Even though redundancy may not always be able to improve sampling rates, it may still be useful to improve tolerance to bit decision errors and to provide redundant information for digital calibration as will be discussed in Chapter 3.

Chapter 4

Digital Background Calibration of SAR ADCs

In Chapter 3, we introduced the redundancy algorithm in successive approximation register analog-to-digital converters. We are able to show that, if implemented correctly, redundancy can provide tolerance to both dynamic and static error sources during the conversion process. In terms of dynamic error sources, redundancy provides room for making early decision mistakes such that these errors can be corrected in the later conversion steps. We are also able to show that depending on the specific design parameters, redundancy has the potential to improve the overall sampling rates. In terms of static error sources, we analyzed the requirements on redundancy to guarantee digital calibratability in the presence of mismatches. We provided a simple relationship between the maximum radix number, the minimum total number of conversion steps and the expected manufacturing random variance of capacitors. This relationship can help us quickly identify the design parameters to be used to ensure good linearity.

In this chapter, we will take another step towards designing higher resolution SAR converters by providing two new digital background calibration schemes that can utilize the redundant information to digitally remove the nonlinearity. In the absence of trimming or calibration, SAR design usually suffers from static nonlinearities which prevent the resolution from going above 8-10 bits [22]. These nonlinearities motivate

active research in developing new calibration techniques to achieve designs with higher accuracy. For example, to achieve exact multiplication by a factor of two regardless of capacitor mismatch error in a pipelined ADC, Li *et al.* in [23] came up with a ratio-independent algorithmic technique, and Song *et al.* in [24] proposed a capacitor error averaging technique. These techniques [23–28] remove static nonlinearity using analog components in the signal path. Even though they are effective ways to remove static nonlinearities in the design, these techniques typically come at the expense of degraded conversion speed and added circuit noise. The circuit noise based *FoM* degradation is roughly 12X and 9X in [23] and [24], respectively.

On the other hand, digital calibration techniques, which can realize the benefit of technology scaling in terms of energy efficiency and speed, have also been developed. They can be classified into two groups: foreground calibration and background calibration. Foreground calibration is done during a calibration phase at startup, measuring nonlinearity by driving the inputs with specific calibration signals to extract the mismatch information. For example, Lee *et al.* in [21] developed a self-calibrated capacitor array in a SAR ADC, exploiting a binary weighted capacitor array. During calibration, the ratio errors of the capacitors are measured sequentially from the MSB capacitor to the LSB capacitor. The mismatch data is stored in a RAM. During the normal operation, the mismatch data is used to correct matching errors of the capacitor array. Other calibration schemes use statistically-based methods to extract nonlinearities based on histogram measurements or code density [29, 30]. Because these calibration schemes require collection of measurement data at the beginning of the operation, they interrupt the normal operation of the ADC. To minimize the effect, it is typical to run these calibration schemes during manufacturing or at startup, meaning that they cannot track parameter drifts.

In contrast, digital background calibration runs transparently in the background so it does not interrupt the normal conversion process. A common approach is to inject a known calibration signal, δ_a , onto the signal path [31–35]. Assuming the corresponding digitized output for the injected calibration signal δ_a is δ_d , with an ideal linear transfer function, a constant shift of δ_d in its output, independent of

the input signals, is expected. In other words, when δ_d is subtracted from the final digitized output, there should be no correlation between the injected signal and the output signal. The calibration engine is, therefore, designed to null such correlation by adjusting the calibration parameters. Using this approach, because the signal path must accommodate the addition of the calibration signal, the signal range and the over-range protection is reduced in the design, in which the headroom may be already limited. Moreover, the effectiveness of calibration also depends on the matching between δ_a and δ_d .

Rather than tampering with the input signal path, another approach uses the input signal itself to estimate the static errors without using a calibration signal [36–38]. Adaptive equalization techniques, prevalent in the digital communication community, are used to resolve nonlinearity problems for pipelined and SAR ADCs in [36,37] and [38], respectively. These techniques typically require an accurate reference ADC to estimate and correct the errors. Even though the reference ADC may run at a slower speed compared with the core ADCs, the added complexity associated with the implementation translates into either higher power consumption or reduction in conversion speed. In summary, the previous techniques are to different degrees either hardware or algorithmically expensive.

As a result, our goal in this chapter is to develop new calibration algorithms that have the following characteristics. First, digital background calibration is preferred so that the calibration process does not interrupt the normal ADC operation. Second, the calibration algorithm uses the input signal itself as stimulus rather than requiring an accurate external calibration signal. Third, the calibration approach uses the original ADC core without adding an additional reference or a calibration channel in the design. Finally, the calibration needs to be built with simple digital hardware. Our calibration schemes are similar to the statistically-based methods that use code density measurements to estimate the capacitor mismatches within the SAR design. The first calibration method that we introduce requires the knowledge of the value of the input signal. The second calibration algorithm requires knowing the statistics of the input waveform, while the third calibration method does not require any knowl-

edge of the input signals as long as the input has a smooth probability distribution function.

The first section of this chapter explains why redundancy in SAR ADCs leads to missing codes in the output code histogram. This helps give us a deeper understanding of redundancy, which leads to the formation of our later calibration algorithms. We then introduce a preliminary calibration algorithm, which requires accurate knowledge of the analog input signal. This approach is impractical in a real implementation but gives us better intuition of the search operation and how to design a calibration scheme without needing knowledge of the analog input. The next part of the chapter discusses the second calibration algorithm we propose, in which the only necessary information we need for the calibration is the statistics of the input. For example, the algorithm has to know whether the input is a sinusoid or an input ramp. The last algorithm we propose improves further upon previous calibration schemes and does not require any knowledge of the signal or the statistics of the input signals; rather, the signals are only required to have smooth statistics. All of these new calibration algorithms have been tested in simulation to verify their effectiveness.

4.1 Missing Codes in Code Density Histogram

As discussed in Section 3.2, the condition that allows digital calibratability of static mismatch errors is $\sum_{i=0}^{n-1} s(i) - s(n) \geq 0$ for all i 's between 1 and M . This condition guarantees that the transfer function only includes missing codes with no missing levels. With no missing levels, sufficient analog information is kept so digital calibration is possible to recover the lost resolution.

For an N -bit M -step redundant SAR ADC, the histogram or output code density is created by counting the number of times each of the 2^M raw-bit combinations has occurred. Because of redundancy ($M > N$), some bins in the histogram or output code density can be zero. The zero code bin represents a missing code. A constant offset error is represented by a shift in the code density map. The number of counts in each code bin is defined as the width of the bin. In the linear ramp case, the

width of the bin is proportional to the size of the analog input range that maps into this bin. In an ideal binary weighted ADC ($N = M$), a full-scale input ramp will generate a code density with equal number of codes within each bin, which means that the width of each bin is equal to $V_{FS}/2^N = 1$ LSB. In an ADC with redundancy ($N < M$), some zero code bins are expected. If all the step sizes are integer multiples and no dynamic errors occur during the search process, with full-scale input ramp, the number of non-zero code bins is exactly 2^N and the number of zero code bins is $2^M - 2^N$. Each bin has an equal number of occurrences. We will see this in the following example.

Figure 4-1 shows an example of missing codes in a normalized output code density plot. The example is plotted for a 3-bit 4-step redundant SAR ADC with $s = [4, 1, 1, 1]$. The equation to calculate D_{out} for a N -bit M -step ADC is repeated here from Chapter 3 as Equation 4.1. For example, when the raw output bin is “0101”, its D_{out} is calculated as $2^2 - 1 + 1 - 1 = 3$. To assure the symmetry of the search algorithm in our design, the search always starts at the mid-level of the full input range, meaning that $s(M)$ is equal to 2^{N-1} in Equation 4.1. This symmetry allows equal tolerance windows during the “up” and “down” transitions of the searching process so that the search is not more sensitive to errors in one half of the input range compared to the other half.

$$D_{out} = s(M) + \sum_{i=1}^{M-1} [2 \cdot b[i] - 1] \times s(i) + [b[0] - 1] \quad (4.1)$$

In this example, code bins 3, 4, 6, 7, 8, 9, 11 and 12 are empty. Table 4.1 gives the relationship between the raw output bits, the progression of decision levels, the digitized output and an indication that shows whether the bin is empty or not. The empty code bin is the result of contradicting logic statements during the transition of decision levels. For example, the output code bin “3” corresponds to raw bits of “0011”, in which it has decision level progressions: (1) $V_{in} < 4$, (2) $V_{in} < 3$, (3) $V_{in} > 2$, and (4) $V_{in} > 3$. The second logic statement and the fourth logic statement clearly contradict one another since the same input cannot be greater and less than 3

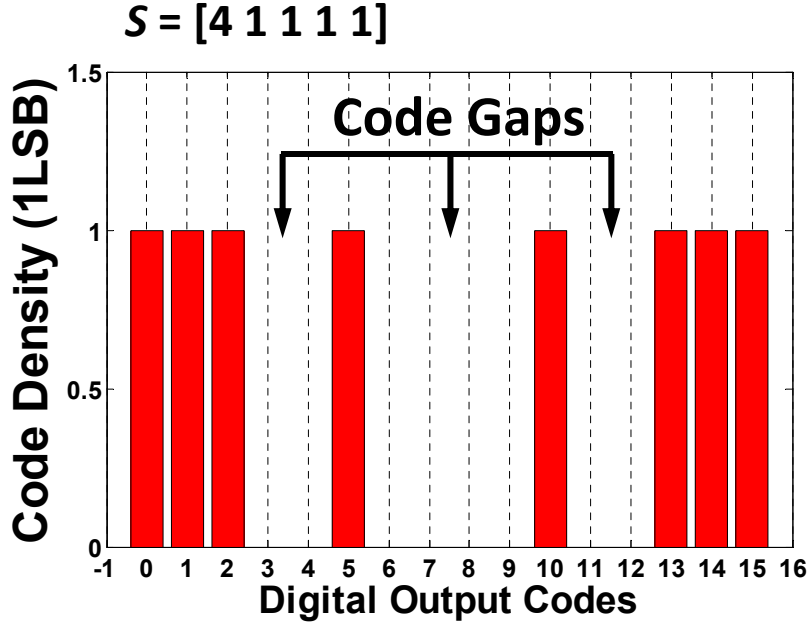


Figure 4-1: Normalized code density histogram with missing codes: 3, 4, 6, 7, 8, 9, 11 and 12. The histogram is generated with a linear input ramp over the full scale. The missing codes are the result of redundancy.

Raw bits	Transition of decision levels	Missing codes?	D_{out}
1111	$V_{in} > 4 \mapsto V_{in} > 5 \mapsto V_{in} > 6 \mapsto V_{in} > 7$		7
1110	$V_{in} > 4 \mapsto V_{in} > 5 \mapsto V_{in} > 6 \mapsto V_{in} < 7$		6
1101	$V_{in} > 4 \mapsto V_{in} > 5 \mapsto V_{in} < 6 \mapsto V_{in} > 5$		5
1100	$V_{in} > 4 \mapsto V_{in} > 5 \mapsto V_{in} < 6 \mapsto V_{in} < 5$	Y	
1011	$V_{in} > 4 \mapsto V_{in} < 5 \mapsto V_{in} > 4 \mapsto V_{in} > 5$	Y	
1010	$V_{in} > 4 \mapsto V_{in} < 5 \mapsto V_{in} > 4 \mapsto V_{in} < 5$		4
1001	$V_{in} > 4 \mapsto V_{in} < 5 \mapsto V_{in} < 4 \mapsto V_{in} > 3$	Y	
1000	$V_{in} > 4 \mapsto V_{in} < 5 \mapsto V_{in} < 4 \mapsto V_{in} < 3$	Y	
0111	$V_{in} < 4 \mapsto V_{in} > 3 \mapsto V_{in} > 4 \mapsto V_{in} > 5$	Y	
0110	$V_{in} < 4 \mapsto V_{in} > 3 \mapsto V_{in} > 4 \mapsto V_{in} < 5$	Y	
0101	$V_{in} < 4 \mapsto V_{in} > 3 \mapsto V_{in} < 4 \mapsto V_{in} > 3$		3
0100	$V_{in} < 4 \mapsto V_{in} > 3 \mapsto V_{in} < 4 \mapsto V_{in} < 3$	Y	
0011	$V_{in} < 4 \mapsto V_{in} < 3 \mapsto V_{in} > 2 \mapsto V_{in} > 3$	Y	
0010	$V_{in} < 4 \mapsto V_{in} < 3 \mapsto V_{in} > 2 \mapsto V_{in} < 3$		2
0001	$V_{in} < 4 \mapsto V_{in} < 3 \mapsto V_{in} < 2 \mapsto V_{in} > 1$		1
0000	$V_{in} < 4 \mapsto V_{in} < 3 \mapsto V_{in} < 2 \mapsto V_{in} < 1$		0

Table 4.1: Relationship between the output bit combinations, decision level progressions, the location of missing codes and their corresponding D_{out} 's for integer step sizes. This shows why missing codes can occur in the output code density map when there is redundancy.

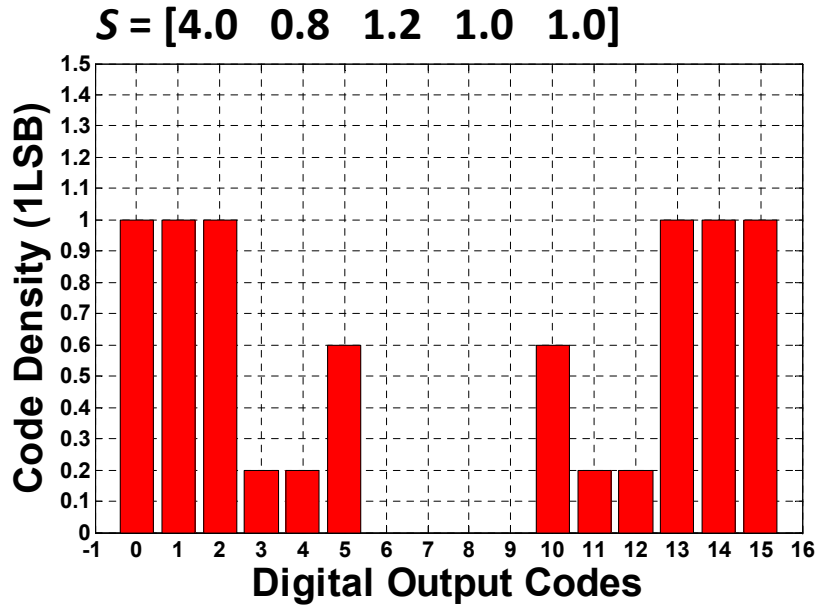


Figure 4-2: Normalized code density histogram with fractional capacitor values. The histogram is generated with a linear input ramp over the full scale.

at the same time. As a result, no input can fall into this output bin. By going through all possible bit combinations and their corresponding decision-level transitions, we can identify bins with missing codes.

Another way to think about why missing codes occur in this case is that for an ADC with 3-bit effective resolution, the ADC can only differentiate $2^3 = 8$ distinct input levels. Therefore, even though there is 16 possible raw output bit-combinations with 16 possible output bins, only 8 bins can be filled. This observation is contingent upon having integer multiple step sizes, s , along with the assumption that the ADC has no circuit noise. For example, Figure 4-2 shows the code density plot with fractional step sizes, $s = [4, 0.8, 1.2, 1, 1]$. In the previous case with integer step sizes, the code bin “0011” was empty, but the same bin is now filled with normalized density of 0.2. Again, following the same procedure as before, we can write down the four logic statements: (1) $V_{in} < 4$, (2) $V_{in} < 3.2$, (3) $V_{in} > 2$ and (4) $V_{in} > 3$. The second and the fourth statements no longer lead to contradiction. However, the range of analog inputs that fall into this bin (or the width of this code bin) is small and can be calculated as $3.2 - 3 = 0.2$.

Raw bits	Transition of decision levels	MC?	D_{out}
1111	$V_{in} > 4 \mapsto V_{in} > 4.8 \mapsto V_{in} > 6.0 \mapsto V_{in} > 7.0$		7
1110	$V_{in} > 4 \mapsto V_{in} > 4.8 \mapsto V_{in} > 6.0 \mapsto V_{in} < 7.0$		6
1101	$V_{in} > 4 \mapsto V_{in} > 4.8 \mapsto V_{in} < 6.0 \mapsto V_{in} > 5.0$		5
1100	$V_{in} > 4 \mapsto V_{in} > 4.8 \mapsto V_{in} < 6.0 \mapsto V_{in} < 5.0$		4
1011	$V_{in} > 4 \mapsto V_{in} < 4.8 \mapsto V_{in} > 3.6 \mapsto V_{in} > 4.6$		4
1010	$V_{in} > 4 \mapsto V_{in} < 4.8 \mapsto V_{in} > 3.6 \mapsto V_{in} < 4.6$		4
1001	$V_{in} > 4 \mapsto V_{in} < 4.8 \mapsto V_{in} < 3.6 \mapsto V_{in} > 2.6$	Y	
1000	$V_{in} > 4 \mapsto V_{in} < 4.8 \mapsto V_{in} < 3.6 \mapsto V_{in} < 2.6$	Y	
0111	$V_{in} < 4 \mapsto V_{in} > 3.2 \mapsto V_{in} > 4.4 \mapsto V_{in} > 5.4$	Y	
0110	$V_{in} < 4 \mapsto V_{in} > 3.2 \mapsto V_{in} > 4.4 \mapsto V_{in} < 5.4$	Y	
0101	$V_{in} < 4 \mapsto V_{in} > 3.2 \mapsto V_{in} < 4.4 \mapsto V_{in} > 3.4$		3
0100	$V_{in} < 4 \mapsto V_{in} > 3.2 \mapsto V_{in} < 4.4 \mapsto V_{in} < 3.4$		3
0011	$V_{in} < 4 \mapsto V_{in} < 3.2 \mapsto V_{in} > 2.0 \mapsto V_{in} > 3.0$		3
0010	$V_{in} < 4 \mapsto V_{in} < 3.2 \mapsto V_{in} > 2.0 \mapsto V_{in} < 3.0$		2
0001	$V_{in} < 4 \mapsto V_{in} < 3.2 \mapsto V_{in} < 2.0 \mapsto V_{in} > 1.0$		1
0000	$V_{in} < 4 \mapsto V_{in} < 3.2 \mapsto V_{in} < 2.0 \mapsto V_{in} < 1.0$		0

Table 4.2: Relationship between the output bit combinations, decision level transitions, the location of missing codes and their corresponding D_{out} 's for fractional step sizes. It shows that some code bins can have different sizes.

4.2 Calibration Algorithm I

The main goal of the calibration algorithm in a SAR ADC is to determine the actual step sizes during the search. Since the final D_{out} is calculated based on the values of S as given in Equation 4.1, the effective resolution of the ADC directly depends on how accurately we know these s parameters. In a SAR ADC, the step sizes, s , are designed by ratioing the capacitors within the DAC. Due to manufacturing variation, it is difficult to achieve matching beyond 10 bit resolution between these capacitors. As a result, calibration is necessary for high resolution design. In this section, we introduce a simple method to calibrate for capacitor mismatches. This first method is not practical in a real implementation, but it provides details that are helpful to motivate the design of our subsequent calibration algorithms.

Figure 4-3 plots a search tree that shows the progression of decision levels graphically. The ADC discussed here is an N -bit M -step ADC with step sizes $S(M)$, $S(M-1)$, ..., $S(1)$, and output bits $b[M-1]$, $b[M-2]$, ..., $b[0]$. The u and d super-

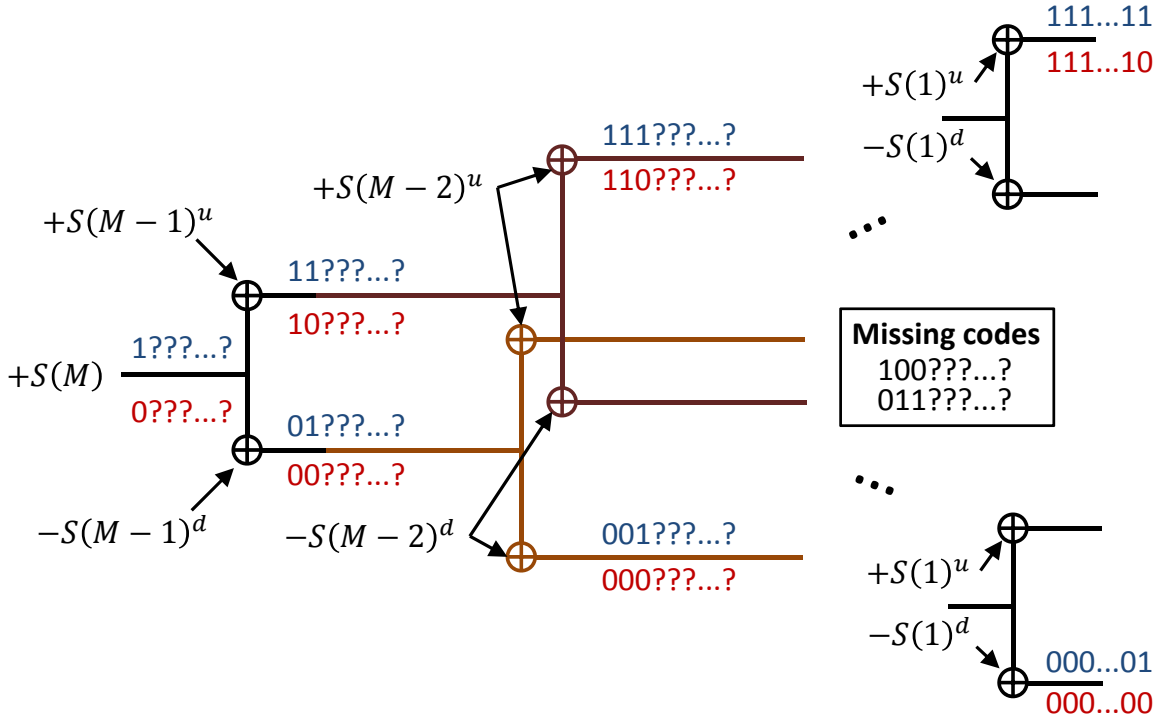


Figure 4-3: Tree representation of a sub-binary search. The diagram shows how a decision level is reached from a previous decision level.

script in the figure distinguish whether it is an “up” or a “down” transitions. In the nominal case, they are the same value. The search begins by comparing V_{in} to the first decision level, $S(M)$. If V_{in} is greater than $S(M)$, then $b[M-1] = 1$; if V_{in} is less than $S(M)$, then $b[M-1] = 0$. After the first step, the search continues by comparing the input to the second decision level. Depending on whether the value of $b[M-1]$ is 0 or 1, the next decision level is either $S(M) + S(M-1)^u$ or $S(M) - S(M-1)^d$, respectively. The next decision level is reached by adding $S(M-2)^u$ or subtracting $S(M-2)^d$ from the previous decision level. This process continues until the searching process is complete.

In this specific example, we have the following condition,

$$S(M) < S(M-2) + S(M-1) \iff S(M-2) > S(M) - S(M-1) \quad (4.2)$$

as shown in Figure 4-4. This condition verifies that redundancy is built within the search algorithm as it satisfies the redundancy criterion described by Equation 3.7 in

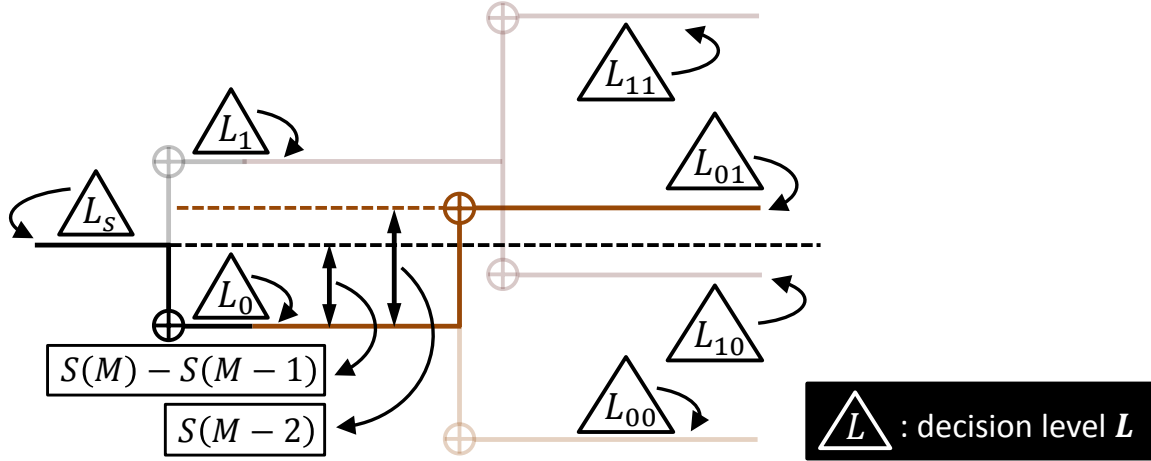


Figure 4-4: Tree representation of a sub-binary search with marked decision levels.

Chapter 3. The effect of redundancy can be viewed graphically in Figure 4-4, where the highlighted path ($L_S \rightarrow L_0 \rightarrow L_{01}$) shows one particular progression of decision levels. We can see that the third decision level L_{01} is higher than the first decision level L_S . In the nominal case in which no decision errors are made, the decision level L_{01} is redundant because the first comparison of the highlighted path already determines that V_{in} is less than L_S . Since L_{01} is larger in magnitude compared to L_S , the decision level L_{01} will always output a “0” in the ideal case. In other words, this shows that the search algorithm has inherent built-in redundancy that it will re-search the region that has already been searched before. However, if no conversion errors are made, all codes begin with “011” are missing.

Figure 4-5 highlights the input range along with its corresponding output bits for the first three transitions of the SAR algorithm. In a binary search case, all the decision levels act as boundaries for one or more input regions, R . In the redundant case, however, this is not always true. For example, in Figure 4-5, decision level L_S acts as boundary for $R_0, R_1, R_{01}, R_{10}, R_{010}, R_{101}$, but decision levels L_{01} and L_{10} ; are buried within the input ranges R_{010} and R_{101} and do not act as boundaries for any of the input ranges during the entire search process. The reason why L_{01} and L_{10} do not act as boundaries is the same as the reason why there are redundancy and missing codes. As explained previously, since decision level L_{01} is above decision level L_S , and an earlier decision already determines that V_{in} in this search path is less than

L_S , there is no output codes that begin with “011”. At the same time, even though V_{in} can be less than decision level L_{01} , the range of inputs that fall into this region is not between L_{01} and L_0 , but between L_S and L_0 . This not only means L_{01} will not be the boundary of any input ranges, it also means that from the data we collected, it cannot be used to extract the value of L_{01} since it is buried within the input range. In this example, we call decision levels L_{01} and L_{10} “un-extractable decision levels (UEDL)” and decision levels L_{00} and L_{11} “extractable decision levels (EDL).” We also define the regions that are bounded by the decision level of the current search step and its immediate preceding search step the “bounded region (BR),” and regions outside are defined as the “unbounded region (UBR).” For example, R_{001} is a bounded region because it bounded by the decision level of the current search step L_{00} and of its immediate preceding search step L_0 ; on the other hand, R_{010} is an unbounded region because it is not bounded by the decision level of the current search step. An unbounded region can, however, be bounded by any previous decision levels in the search path, but the exact decision levels it is bounded by are unknown only until the actual step sizes are extracted. As a result, due to this uncertainty, they cannot be used to extract the decision levels.

As this example suggests, decision levels with all ones or zeros ($L_{11\dots1}$ or $L_{00\dots0}$) are always extractable decision levels. This is because to get to these decision levels, the SAR algorithm always moves in one direction after the first decision level. This guarantees that the next decision level always falls in a region that has not been searched before. For instance, decision progression $L_S \mapsto L_1 \mapsto L_{11}$ only moves upwards to a region without any previous decision levels. This helps prevent overlapping with previous decision levels and allow these decision levels to always be the boundary of an input range. On the other hand, for decision level with both 1’s and 0’s, meaning that the decision path moves in both directions during the search, without knowing the exact S values, we cannot determine for certain whether it will be an un-extractable decision level or an extractable decision level.

If we want to extract the actual value of $S(M)$, the easiest and most intuitive way is to sweep the input voltage slowly from 0 to full scale, at a resolution that

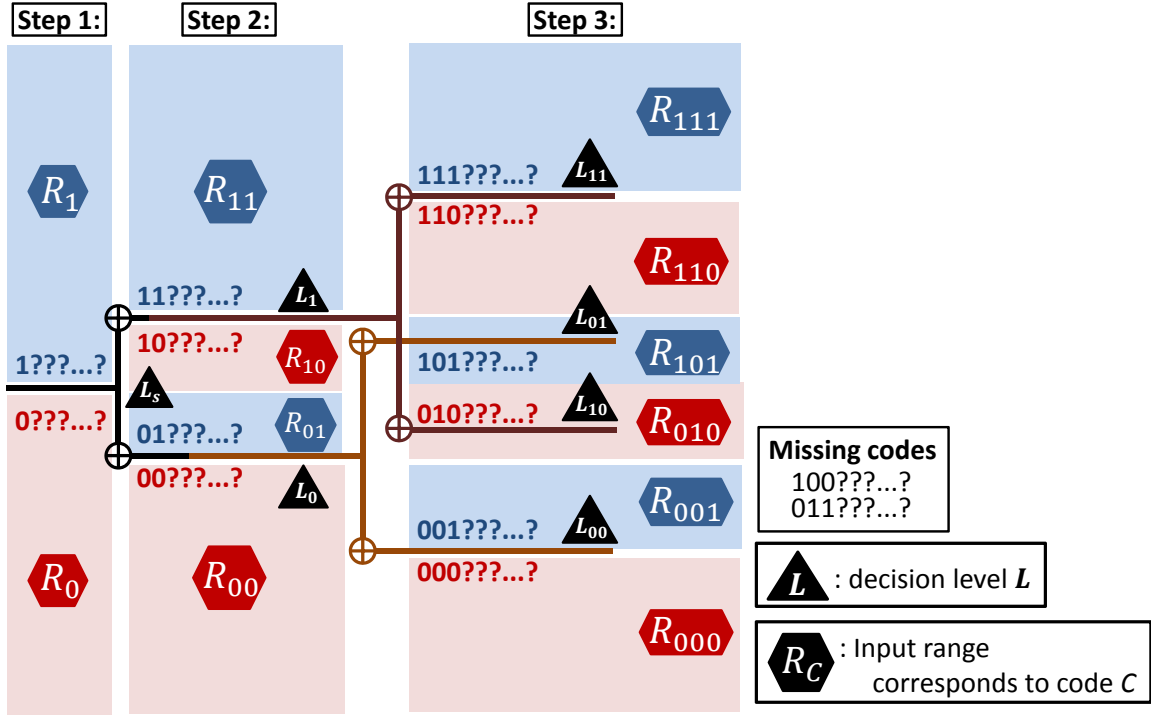


Figure 4-5: A sub-binary search tree with highlighted regions R_C , indicating the input range corresponds to code C .

is a fraction of the LSB. The goal is to find the voltage level at which $b[M-1]$ switches from 0 to 1 and this voltage level corresponds to L_S or $S(M)$. To extract $S(M-1)$, we follow the same procedure as before. In this case, we have two options. We can either sweep the input voltage to find when the bit pattern changes from $00??? \dots ?$ to $01??? \dots ?$ or when the bit pattern changes from $10??? \dots ?$ to $11??? \dots ?$. These correspond to L_0 and L_1 , respectively. Since we know $L_0 = S(M) - S(M-1)$ and $L_1 = S(M) + S(M-1)$, once $S(M)$ is known from the first extraction, both equations can be used to calculate the value of $S(M-1)$. To extract $S(M-2)$, the same procedure is followed, but we can no longer use all decision levels. As shown in Figure 4-5, sweeping the input voltage can only identify where L_{00} and L_{11} are located since L_{10} and L_{01} are both buried within the input range, and consequently, they do not act as decision boundaries and cannot be used to extract $S(M-2)$. In other words, in order to extract the S parameters, only extractable decision levels are used. Continuing with the same procedure, all S can be extracted.

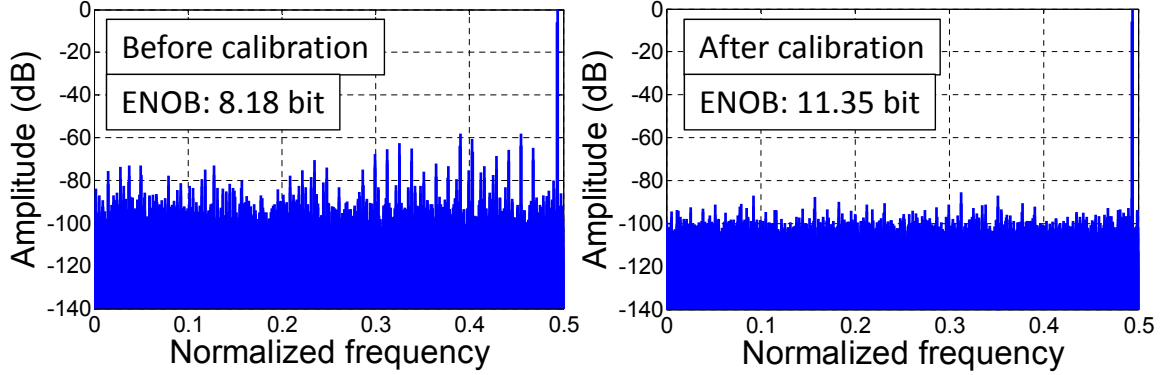


Figure 4-6: Spectrum data before and after calibration scheme I. The effective number of bits (ENOB) improves from 8.18b to 11.35b.

Table 4.3 demonstrates the application of this calibration algorithm on a simulated 12-bit 16-step ADC. “ S_{design} ,” “Actual S ,” “Ext S ,” and “Ext $S w n$ ” represent the designed value, the actual value, the extracted value and the extract value in the presence of noise of the step sizes, respectively. The actual step sizes are generated assuming the unit step has additive random normal variation with standard deviation of 0.05. The extraction simulation is done by sweeping the input linearly in the increment of 0.01 LSB. We see that this algorithm does a good job in extracting the actual step sizes as shown in Table 4.3. Figure 4-6 shows the spectrum data before and after the calibration. The ENOB increases from 8.18b to 11.35b and the harmonic distortion is significantly reduced with the spurious-free dynamic range (SFDR) improved from 58.25dB to 89.2dB. In this first simulation, however, we assume there is no circuit noise. Circuit noise can blur the transition voltage because noise allows the same input near the decision boundary to be mapped to different digital output codes. Since the algorithm uses the minimum and maximum voltage around the bit transition to estimate the step sizes, it is sensitive to any variation in the voltage around the decision levels. As a result, this blurring can degrade the effectiveness of this algorithm; we can see that the extracted step sizes in the presence of noise are less accurate compared to the case without noise, shown in the same table. In this simulation, the noise is assumed to have standard deviation of 0.5 LSB.

Even though this method provides an effective and accurate way to extract the true

M	S_{design}	Actual S^u	Ext S^u	Ext S^u w n	Actual S^d	Ext S^d	Ext S^d w n
16	448	447.77	447.77	448.44	449.93	449.96	450.53
15	608	613.27	613.28	614.21	610.15	610.18	611.03
14	384	383.82	383.83	384.29	379.42	379.43	380.31
13	240	236.79	236.80	237.13	245.78	245.79	245.89
12	144	144.35	144.35	144.50	143.47	143.48	143.68
11	96	94.17	94.18	94.14	96.90	96.90	97.13
10	48	48.46	48.46	48.73	44.94	44.94	44.75
9	32	32.87	32.87	32.62	30.92	30.92	31.06
8	16	16.14	16.14	15.93	15.55	15.55	15.47
7	16	14.62	14.62	14.07	14.52	14.52	14.22
6	5	4.93	4.93	5.08	5.22	5.22	4.86
5	4	3.99	3.99	3.69	4.02	4.02	3.95
4	2	2.02	2.02	1.91	2.02	2.02	1.72
3	2	1.94	1.94	1.23	2.06	2.06	1.30
2	1	1.06	1.06	0.58	0.94	0.94	0.62
1	1	1.00	1.00	0.44	1.01	1.01	0.44

Table 4.3: Estimation of step sizes using the first calibration algorithm. Without adding circuit noise, this extraction procedure is able to extract the actual step sizes with high accuracy; with the addition of circuit noise, this extraction procedure begins to lose accuracy.

stepping sizes, S , it has a few limitations. First, the accuracy of S is a direct function of the accuracy of the input voltage. To extract S for a high-resolution ADC, an external instrument is needed to generate an accurate input signal for this calibration approach. This is not a practical solution since all chips have to go through this post-fabrication procedure. The calibration cannot track parameter drift over time, but which also increases the overall cost of the design. Also, the algorithm depends on finding the minimum and maximum of a range to extract the extractable decision levels. The presence of circuit noise blurs such boundaries and the accuracy of the extraction decreases.

4.3 Calibration Algorithm II

From the previous section, we see that using the absolute magnitude of the input signal as stimulus for calibration puts a stringent requirement on the resolution of the

input and at the same time, the calibration procedure is very sensitive to circuit noise. Rather than relying on the absolute magnitude, in this section we introduce another calibration algorithm that uses signal statistics to calibrate for mismatches. Because it is based on the statistics of code counts, the effect of circuit noise is averaged out when more data is collected. This method is especially suited for high resolution converters (≥ 12 bits), where circuit noise is the major limiting factor. Compared with the first algorithm in which each code is collected once, this algorithm allows more accurate and consistent extraction.

Similar to what was done before, we estimate the location of decision level L in order to extract the actual step sizes S . Rather than sweeping the input voltage to find L , we collect enough counts over the full input range and create a code density map. The exact code count requirement is based on factors such as the input waveform, the percent confidence and the resolution of the ADC. The code density along with the knowledge of the probability distribution of the input waveform can then be used to estimate the location of the decision levels. Using Figure 4-5 as an example, to estimate the decision levels, the ADC is stimulated using an input signal with known probability distribution function. The amplitude of the input waveform has to be greater than the full scale to exercise all codes, but it is not necessary to know the exact amplitude. To extract L_S , the total code counts in R_0 and R_1 are attained from the collected data. Based on their relative counts and the knowledge of input probability distribution function, L_S can be calculated. To calculate L_0 , the code density in region R_{01} between L_S and L_0 is used; and likewise, to calculate L_{00} , the code density in region R_{10} between L_0 and L_{00} is used. Again, as in the case of the first calibration algorithm, the unbounded regions (UBR) cannot be used to extract the decision levels, because they are bounded by decision levels that are unknown until the actual step sizes are extracted.

4.3.1 Choice of Calibration Signal

The natural choice of the input waveform would be a ramp or a uniformly distributed random input. Both inputs will generate an approximately equal number of samples

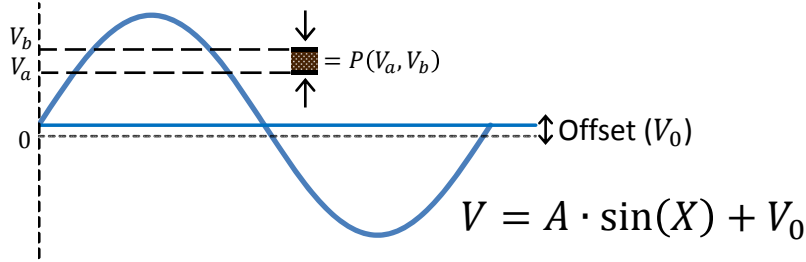


Figure 4-7: Statistics of a sinusoid signal. The sinusoid signal is assumed to have amplitude A and offset voltage V_0 .

within each bin except for the first and last bin since all codes outside the input range will accumulate in these bins. The fundamental limitation of using a ramp is that it is difficult to generate a ramp with high linearity. A few percentage changes in the slope of the ramp would directly translate into the same percentage change in INL after calibration. This error can skew the extraction result of the decision levels and consequently affect the accuracy of the extracted step sizes. A uniformly distributed random input is expected to have equal likelihood of the voltage over the full scale input range. One approach to generate such a signal is by using a pseudorandom digital sequence generator. The digital sequence then goes through an ideal low-pass filter to generate a “random” analog voltage. The drawback of such an approach is that the digital input amplitude must stay constant and the filter must be ideal to avoid introducing any distortion into the input signal. Another difficulty associated with using a ramp is that it is difficult to quantify its distortion levels. On the other hand, a sinusoid input can be generated with very low distortion and the distortion level can easily be quantified by taking the FFT of the waveform. Commercial filters (such as ones made by TTE, Inc.) have total harmonic distortion of roughly 96dB, meaning that a very low distortion sine wave signal can be obtained. It is difficult to generate a ramp with comparable distortion level.

Figure 4-7 shows a sine wave with amplitude A and offset voltage V_0 . The probability density function of a sine wave is defined as the relative likelihood for the sine wave $A \cdot \sin(X) + V_0$ to take on a given value over one period where $0 \leq X \leq 2\pi$.

The function can be described by Equation 4.3.

$$P(V) = \frac{1}{\pi\sqrt{A^2 - (V - V_0)^2}} \quad (4.3)$$

Integrating this function with respect to voltage can give us the probability distribution function of a sine wave, as shown in Equation 4.4.

$$P_{a,b} = P(V_a, V_b) = \frac{1}{\pi} \left\{ \sin^{-1} \left[\frac{V_b - V_0}{A} \right] - \sin^{-1} \left[\frac{V_a - V_0}{A} \right] \right\} \quad (4.4)$$

where $P_{a,b}$ represents the probability that a voltage falls in the range between V_a and V_b . The sine wave must be sampled at random. To sample at random means to sample at a rate that is not harmonically related to the input frequency; otherwise, the same voltage is going to be sampled repetitively over time resulting in a code density that has many empty codes bins. To prevent improper sampling, the samples are collected with a sampling frequency that is not harmonically related to the input frequency. The following analysis builds upon a previous analysis done in [20].

4.3.2 Calibration using a Sine Wave

In this section, we discuss how to extract the decision levels by looking at the code density generated using a sine wave input. The sampled frequency and the input frequency are non-harmonically related.

Calibration Step I

The first step of the calibration algorithm is to extract the offset voltage, which is the same as the first decision level L_S given in Figure 4-5. This is done by counting the number of samples in regions R_0 and R_1 and dividing the number by the total number of counts. We denote the total counts in region R_0 and R_1 by N_0 and N_1 , respectively, and the total number of counts by N_{tot} .

Going back to Figure 4-7, the probability that the ADC output is positive (p_p) is the probability that the sampled voltage is between V_0 and A . Substituting $V_b = A$

and $V_a = V_0$ in Equation 4.4, the probability of being positive (p_p) can be calculated as follows,

$$p_p = \frac{1}{\pi} \left\{ \sin^{-1} \left[\frac{A}{A} \right] - \sin^{-1} \left[\frac{V_0}{A} \right] \right\} \quad (4.5)$$

Since the ADC output is either positive or negative, the probability that a sampled voltage is negative (p_n) can be calculated from p_p as follows,

$$p_n = 1 - p_p \quad (4.6)$$

Solving Equation 4.5 and 4.6 together, a closed-form solution of V_0 can be obtained as follows,

$$V_0 = A \frac{\pi}{2} \sin(p_p - p_n) \quad (4.7)$$

p_p can be estimated from the sampled data using N_1/N_{tot} and p_n can be estimated from the sampled data using N_0/N_{tot} . The estimated V_0 , \hat{V}_0 , is calculated by replacing p_p and p_n with these estimated values.

$$\hat{V}_0 = A \frac{\pi}{2} \sin\left(\frac{N_1 - N_0}{N_{tot}}\right) \quad (4.8)$$

Note that the solution \hat{V}_0 is expressed as a function of the unknown amplitude A . This will not be a problem as we will see later that all extracted step sizes are expressed as a function of A . Since we know the sum of all steps must be equal to 2^N where N is the total number of bits, A can easily be calculated. The value of \hat{V}_0 corresponds to the first decision level L_S and $S(M) = L_S$.

Calibration Step II

The next step of the calibration algorithm is to extract the remaining decision levels from the collected data. Taking the cosine of both sides of Equation 4.4 can lead to a solution of V_b in terms of V_a . The result yields:

$$V_b = V_0 + (V_a - V_0) \cos(\pi \cdot P_{a,b}) + (-1)^s \cdot \sin(\pi \cdot P_{a,b}) \sqrt{A^2 - (V_a - V_0)^2} \quad (4.9)$$

$$s = \begin{cases} 0 & \text{if } V_b \geq V_a \\ 1 & \text{if } V_b < V_a \end{cases}$$

This equation gives a way to calculate V_b once we know V_a and the code counts between V_a and V_b . In this case, since we already have an estimate of V_0 from the previous calibration step, to estimate the next decision level L_1 (or L_0), we can substitute \hat{V}_0 for V_a and count the total number of codes between V_0 and L_1 (or L_0) to estimate these decision levels.

As an example, to estimate L_1 (or L_0) in Figure 4-5, we count the number of codes that fall in region R_{10} (or R_{01}) and denote this quantity as N_{10} (or N_{01}). An estimate of L_1 (or L_0) can then be calculated as follows:

$$\hat{L}_1 = \hat{V}_0 + (\hat{V}_0 - \hat{V}_0) \cos\left(\pi \cdot \frac{N_{10}}{N_{tot}}\right) + \sin\left(\pi \cdot \frac{N_{10}}{N_{tot}}\right) \sqrt{A^2 - (\hat{V}_0 - \hat{V}_0)^2} \quad (4.10)$$

$$\hat{L}_0 = \hat{V}_0 + (\hat{V}_0 - \hat{V}_0) \cos\left(\pi \cdot \frac{N_{01}}{N_{tot}}\right) - \sin\left(\pi \cdot \frac{N_{01}}{N_{tot}}\right) \sqrt{A^2 - (\hat{V}_0 - \hat{V}_0)^2} \quad (4.11)$$

Subsequently, $S(M-1)$ can be calculated:

$$S(M-1)^u = +(\hat{L}_1 - \hat{L}_S) \quad (4.12)$$

$$S(M-1)^d = -(\hat{L}_0 - \hat{L}_S) \quad (4.13)$$

Similarly, to extract the next step size, $S(M-2)$, we follow the same procedure by counting the number of codes that fall in the regions R_{110} and R_{001} and denote these quantities as N_{110} and N_{001} , respectively. Estimates of L_{11} , L_{00} and $S(M-2)$ can be calculated as follows:

$$\hat{L}_{11} = \hat{V}_0 + (\hat{L}_1 - \hat{V}_0) \cos\left(\pi \cdot \frac{N_{110}}{N_{tot}}\right) + \sin\left(\pi \cdot \frac{N_{110}}{N_{tot}}\right) \sqrt{A^2 - (\hat{L}_1 - \hat{V}_0)^2} \quad (4.14)$$

$$\hat{L}_{00} = \hat{V}_0 + (\hat{L}_1 - \hat{V}_0) \cos\left(\pi \cdot \frac{N_{001}}{N_{tot}}\right) - \sin\left(\pi \cdot \frac{N_{001}}{N_{tot}}\right) \sqrt{A^2 - (\hat{L}_1 - \hat{V}_0)^2} \quad (4.15)$$

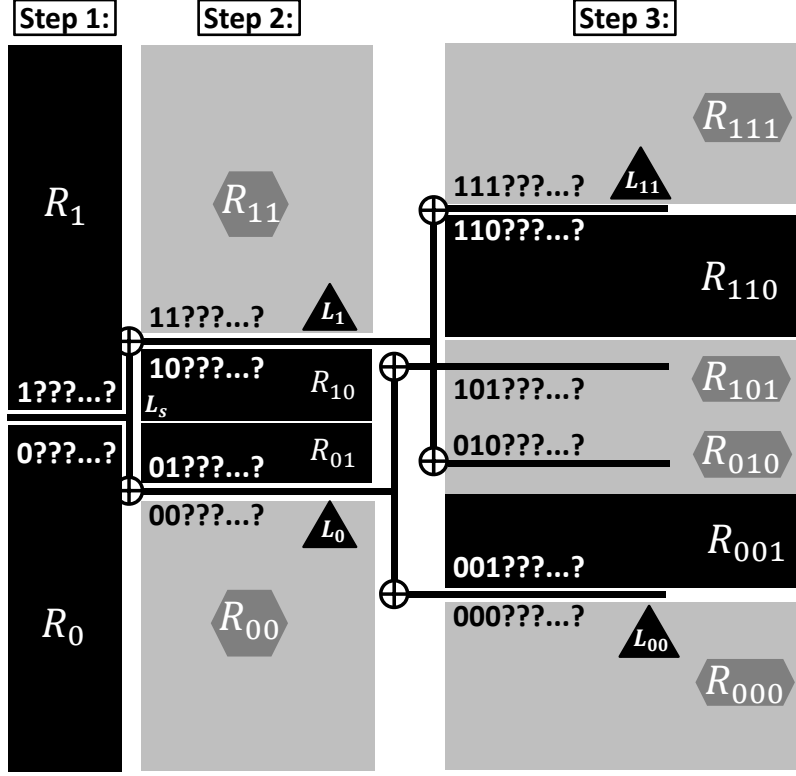


Figure 4-8: Using the “bounded regions” to extract the actual step sizes. The bounded regions are highlighted in black. This calibration scheme uses the statistics of the input signals rather than relying on the exact knowledge of the input signals as in the case of the first calibration algorithm.

$$S(M - 2)^u = +(\hat{L}_{11} - \hat{L}_1) \quad (4.16)$$

$$S(M - 2)^d = -(\hat{L}_{00} - \hat{L}_0) \quad (4.17)$$

Note that in this case, again, we do not use the code counts in the “unbounded regions” R_{101} and R_{010} . The code counts in this example correspond to the step size $S(M - 1)$, rather than $S(M - 2)$. In a more general case, however, we do not know which step size is calculated when using the code counts in the “unbounded region” until the actual step sizes are extracted. This procedure can be done recursively by counting the number of codes in the “bounded regions” of each step to extract the corresponding step sizes as shown in Figure 4-8.

Table 4.4 shows the extraction results of using the second calibration algorithm for our simulated 12-bit 16-step ADC. Again, the actual step sizes are generated

M	S_{design}	Actual S^u	Ext S^u	Ext $S^u w n$	Actual S^d	Ext S^d	Ext $S^d w n$
16	448	451.41	451.39	451.30	439.09	439.10	439.05
15	608	607.10	607.15	607.14	614.39	614.41	614.41
14	384	382.97	382.96	383.07	382.63	382.61	382.62
13	240	239.50	239.48	239.48	236.92	236.93	236.99
12	144	145.14	145.16	145.12	146.28	146.26	146.28
11	96	95.17	95.16	95.15	98.15	98.18	98.15
10	48	48.01	48.00	48.04	49.31	49.30	49.23
9	32	32.29	32.30	32.27	32.81	32.82	32.87
8	16	15.03	15.03	15.01	16.13	16.12	16.18
7	16	15.68	15.69	15.69	16.98	16.97	16.91
6	5	4.92	4.90	4.89	4.76	4.78	4.79
5	4	4.06	4.07	4.05	3.80	3.80	3.78
4	2	1.87	1.87	1.88	1.95	1.93	1.98
3	2	2.00	2.00	1.99	1.85	1.85	1.84
2	1	0.92	0.91	0.91	1.04	1.05	1.03
1	1	0.96	0.95	0.98	0.91	0.91	0.92

Table 4.4: Estimation of step sizes using the statistical calibration algorithm. The accuracy of this extraction procedure is not affected by circuit noise. Due to the statistical nature of this calibration scheme, the extraction precision can be increased by collecting more samples.

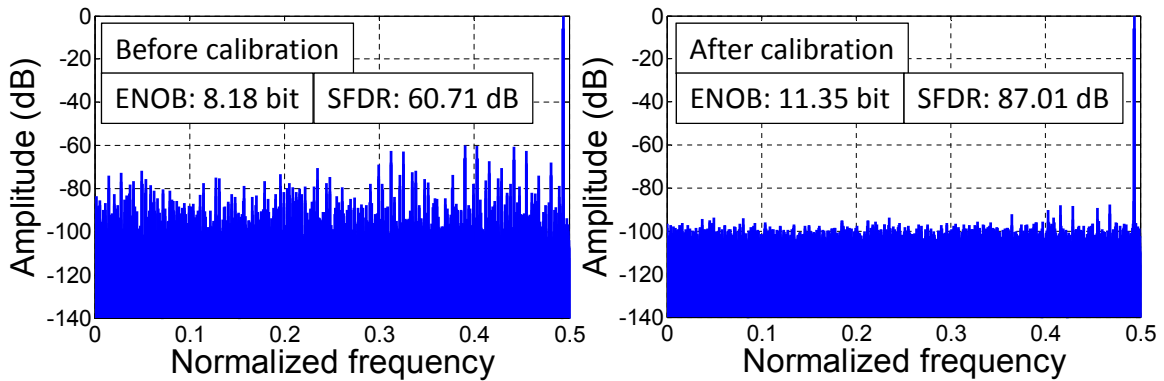


Figure 4-9: Spectrum data before and after using the statistical calibration algorithm. ENOB improves from 8.18b to 11.35b and SFDR improves from 60.71dB to 87.01dB.

assuming the unit step size has additive random normal variation with standard deviation of 0.05. The simulation is done by sampling a 24.7MHz sinusoid waveform at 50MS/s, with a total of 2^{20} samples. The sinusoid has amplitude $A = 1.1 \times V_{ref}$ and offset voltage $V_0 = 20$ LSB. The results show that the extraction procedure is able to extract the actual step size with high accuracy. Another simulation is done by assuming there is additive random circuit noise that has a standard deviation equal to 0.5 LSB. Because of the statistical nature of the extraction procedure, the effect of circuit noise is averaged out when a large amount of data is collected. Therefore, unlike in the case of the first calibration algorithm, the accuracy of the extraction result is not affected as shown in the same table. The spectrum data before and after using this calibration algorithm is plotted in Figure 4-9. We see that the ENOB improves from 8.18b to 11.35b, and the SFDR improves from 60.71dB to 87.01dB. All harmonic distortions are significantly reduced.

This calibration algorithm has several benefits over the previous calibration algorithm. First, the calibration process can be done much faster because it is no longer necessary to sweep the input in small and high-resolution steps. The resolution of the extracted step size is a function of the resolution of the input in the previous approach, but in the statistical approach here, the resolution can be increased arbitrarily by collecting more samples. Second, the accuracy of the extraction is not affected by circuit noise, since noise is averaged out when more data is collected. Although this calibration scheme is more practical compared to the first calibration algorithm, it still requires the knowledge of the probability distribution function of the input signal. Moreover, as discussed before, even though it is easier to generate a sine wave with higher linearity and low distortion, an external instrument is still needed. In the next section, we will introduce another calibration algorithm that does not require any prior knowledge of the input signal while still able to extract the step sizes accurately.

4.4 Calibration Algorithm III

In this section, we introduce a new calibration algorithm that does not require the knowledge of the input signal. The only requirement of this calibration scheme is that the input waveform needs to have a smooth probability distribution function, meaning that there are no abrupt changes in the code densities. To understand this algorithm, for simplicity, we will assume that the input is a ramp with equal probability of generating any analog voltages over the full input range. We will generalize the algorithm in the later parts of this section to allow any input signals with unknown probability distribution function.

4.4.1 Integer Step Sizes Extraction

Equation 4.1 describes the relationship such that an N -bit digitized output (D_{out}) can be calculated from the actual step sizes (S 's) and M digital output bits. Since D_{out} has to be an integer, when we used this equation in the past, we assume that all the step sizes are integer multiples of each other. In real implementation, however, $S(i)$'s are typically not integer multiples of each other due to manufacturing variation. As a result, D_{out} can be a non-integer if using Equation 4.1. To avoid this problem, we rewrite the previous relationship to generate F_{out} rather than D_{out} , given by Equation 4.18. The new parameter F_{out} now represents the final decision level during the search process, which could be a non-integer due to non-integer step sizes. D_{out} is now obtained by taking the floor operation on the new parameter F_{out} as shown in Equations 4.18 and 4.19. In this section, we focus on integer $S(i)$'s, with $D_{out} = F_{out}$.

$$F_{out} = S(M) + \sum_{i=1}^{M-1} [2 \cdot b[i] - 1] \times S(i) + [b[0] - 1] \times S(0) \quad (4.18)$$

$$D_{out} = \text{floor}(F_{out}) \quad (4.19)$$

Figure 4-10 plots the code density for a 3-bit 4-step ADC with step sizes, $S = [4, 1, 1, 1, 1]$. For each code bin, the plot shows the range of analog inputs that fall into this bin, the digital output bit sequences, and the procedure of calculating its

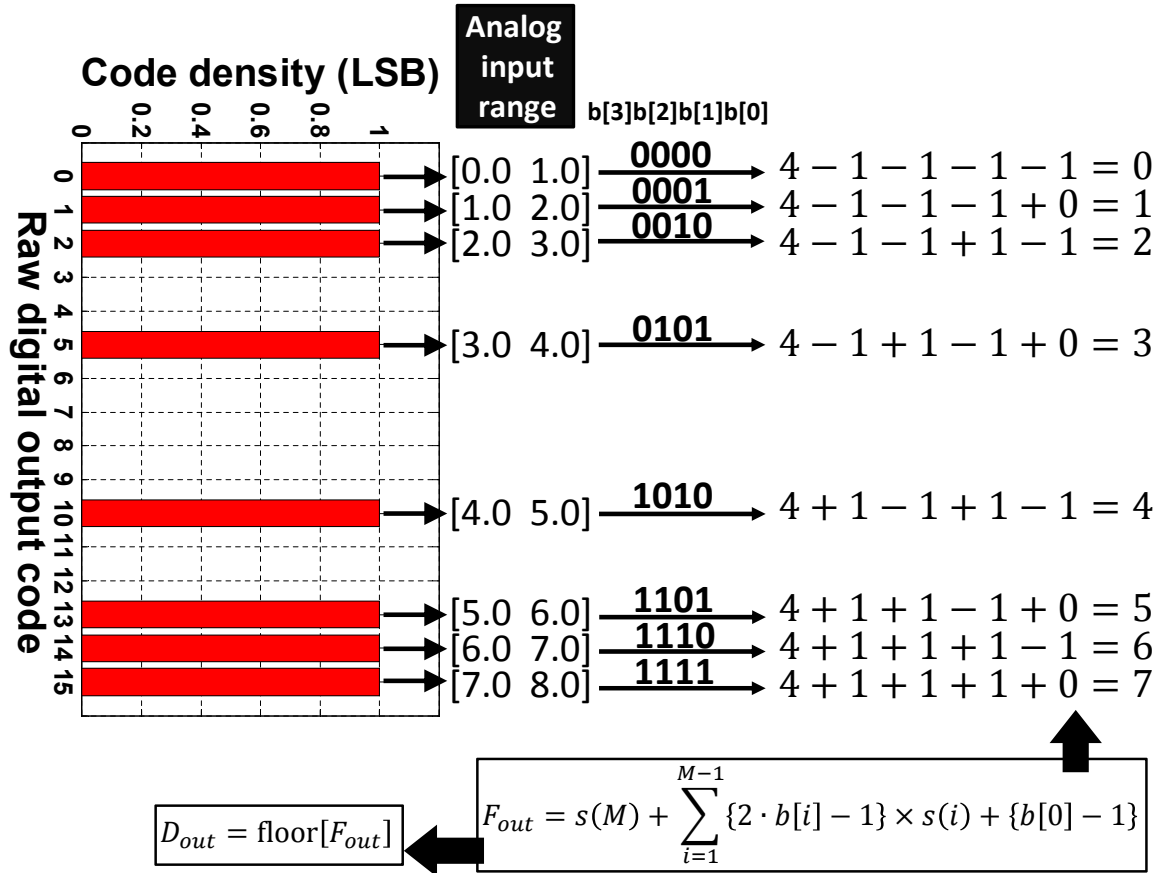


Figure 4-10: Calculating D_{out} when all the step sizes are integer multiples of each other. In this case, $D_{out} = F_{out}$.

corresponding D_{out} . The code density has eight non-empty code bins and each has a normalized bin size of 1 LSB. Since the input is a ramp with uniform probability distribution function, the normalized bin size is equal to the width of the digital output code. In this case, when all code bins have width of 1 LSB, the ADC has perfect static linearity.

The code density plot is generated assuming that no dynamic errors are made during the conversion process. This implies that even with the presence of redundancy, when one output bit sequence is larger than another output bit sequence, its corresponding analog input is also larger than the analog input of this other sequence. We can think of the output bit sequences as the record of the comparison results during the search process. While comparing two different output bit sequences, X and Y , from the MSB to the LSB bits, we assume that the first bit they differ in is bit i and

$X[z] = 1$ and $Y[z] = 0$. This means that at this bit comparison, the search process determines that the analog input voltage (X_A) corresponding to X is larger than a decision level and the analog input voltage (Y_A) corresponding to Y is smaller than that same decision level. If no conversion errors are made during the search process, this means that $X_A > Y_A$.

This monotonicity allows us to determine the input range of each code bin by accumulating the bin width from the LSB bin to the MSB bin. For example, we know that the smallest analog input, 0, must fall into bin 0 because it has the smallest bit sequence “0000.” Since 0 is the smallest analog input, it must also be the lower bound of the analog input range for bin 0. To calculate the upper bound of the input range, we add its bin width (= 1 LSB), to the lower bound of the input range. This leads to the analog input range that is equal to $[0 \ 1]$. Next, we know the analog input ranges must be continuous without gaps since all analog inputs are mapped to some digital outputs. This means that the adjacent input ranges must have equal upper and lower bounds. As a result, the lower bound of bin “0001” must be the same as the upper bound of bin “0000.” To obtain the upper bound of the input range for this bin, we follow the same procedure by adding its bin width to the lower bound and obtain the analog input range for bin 1 equal to $[1 \ 2]$. This can be done successively to find all the input ranges. Note that D_{out} in this example is the same as the lower bound of the analog input range, which can be obtained by accumulating the bin width from LSB to the MSB bin, with the first D_{out} being equal to 0.

To extract the actual step sizes, similar to the previous calibration algorithm, the ADC has to sample enough input points to generate a code density map that is statistically significant. Since the bin size is proportional to the width of the digital output code, the code density is first being normalized by making the sum of all bin widths equal to 2^N , and in this case, equal to 8. For each code bin, an equation can then be written according to Equation 4.18. Their corresponding F_{out} ’s are obtained by accumulating the bin width from the LSB bin to the MSB bin. The resulting

equations are listed in 4.20.

$$\left\{ \begin{array}{l} S(4) - S(3) - S(2) - S(1) - S(0) \cdot 1 = F_{0000} = 0 \\ S(4) - S(3) - S(2) - S(1) + S(0) \cdot 0 = F_{0001} = 1 \\ S(4) - S(3) - S(2) + S(1) - S(0) \cdot 1 = F_{0010} = 2 \\ S(4) - S(3) + S(2) - S(1) + S(0) \cdot 0 = F_{0101} = 3 \\ S(4) + S(3) - S(2) + S(1) - S(0) \cdot 1 = F_{1010} = 4 \\ S(4) + S(3) + S(2) - S(1) + S(0) \cdot 0 = F_{1101} = 5 \\ S(4) + S(3) + S(2) + S(1) - S(0) \cdot 1 = F_{1110} = 6 \\ S(4) + S(3) + S(2) + S(1) + S(0) \cdot 0 = F_{1111} = 7 \end{array} \right. \quad (4.20)$$

These equations are solved by first subtracting the i^{th} equation from the $(i^{th} + 1)$ equation. The results are re-arranged in a matrix form given in Equation 4.21. The equation is then solved by the ordinary least square solution, which leads to a closed-form expression for the estimated value of the step sizes, given in Equation 4.22.

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & -1 \\ 0 & 2 & -2 & 1 \\ 2 & -2 & 2 & -1 \\ 0 & 2 & -2 & 1 \\ 0 & 0 & 2 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} S(3) \\ S(2) \\ S(1) \\ S(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad \longleftrightarrow \quad X \times S = Y \quad (4.21)$$

$$\begin{bmatrix} S(3) \\ S(2) \\ S(1) \\ S(0) \end{bmatrix} = (X^T X)^{-1} X^T Y = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad (4.22)$$

Note that the value of $S(4)$ cannot be extracted by solving the above linear equa-

tions. To extract the value of $S(4)$, a similar approach that compares the total counts of the positive outputs versus negative outputs can be used here as in the case of the second calibration algorithm. However, since the value of $S(4)$ only introduces a constant offset, it does not affect linearity of the converter. The exact value of $S(4)$ is typically not as important compared to the values of other step sizes. Overall, we see that the calibration procedure introduced here is able to extract the actual step sizes accurately in this simple example.

One concern may be that at higher resolution, the size of the matrix X can grow considerably larger and require a much longer computation time in order to solve. In this case, however, the matrix X is a simple matrix with beneficial structure. First, it is filled with only a few distinct elements, ± 1 , ± 2 . Second, the matrix is sparse, meaning that it is populated primarily with zeros. Techniques such as Markowitz reordering can be used to minimize fill-in and a follow-up iterative or direct method can be used to solve this sparse matrix very efficiently. In addition, the sparse matrix also has benefits in terms of storage. To store a typical two-dimensional N by M matrix, a total of $N \times M$ memory elements are needed. In a sparse matrix, substantial memory reduction can be realized by only storing the non-zero elements in the array. Depending on the exact distribution of the non-zero entries, different data structures can be adapted to yield significant savings in memory.

4.4.2 Fractional Step Sizes Extraction

Figure 4-11 plots the code density for a 3-bit 4-step ADC with non-integer step sizes $S = [4, 0.9, 1.1, 1, 1]$. Similar to what was done in the integer case, for each bin, the range of analog inputs, the output bit sequences and the F_{out} are marked on the same figure. From this code density plot, unlike in the integer case, not all of the code bins have bin width equal to 1 LSB. This implies that there is static nonlinearity in this ADC. Again, the code density is generated assuming there are no dynamic errors during the conversion process.

As alluded to in the previous discussion, when the step sizes are not integers, the calculated final decision level (F_{out}) can also become non-integer. This is evident

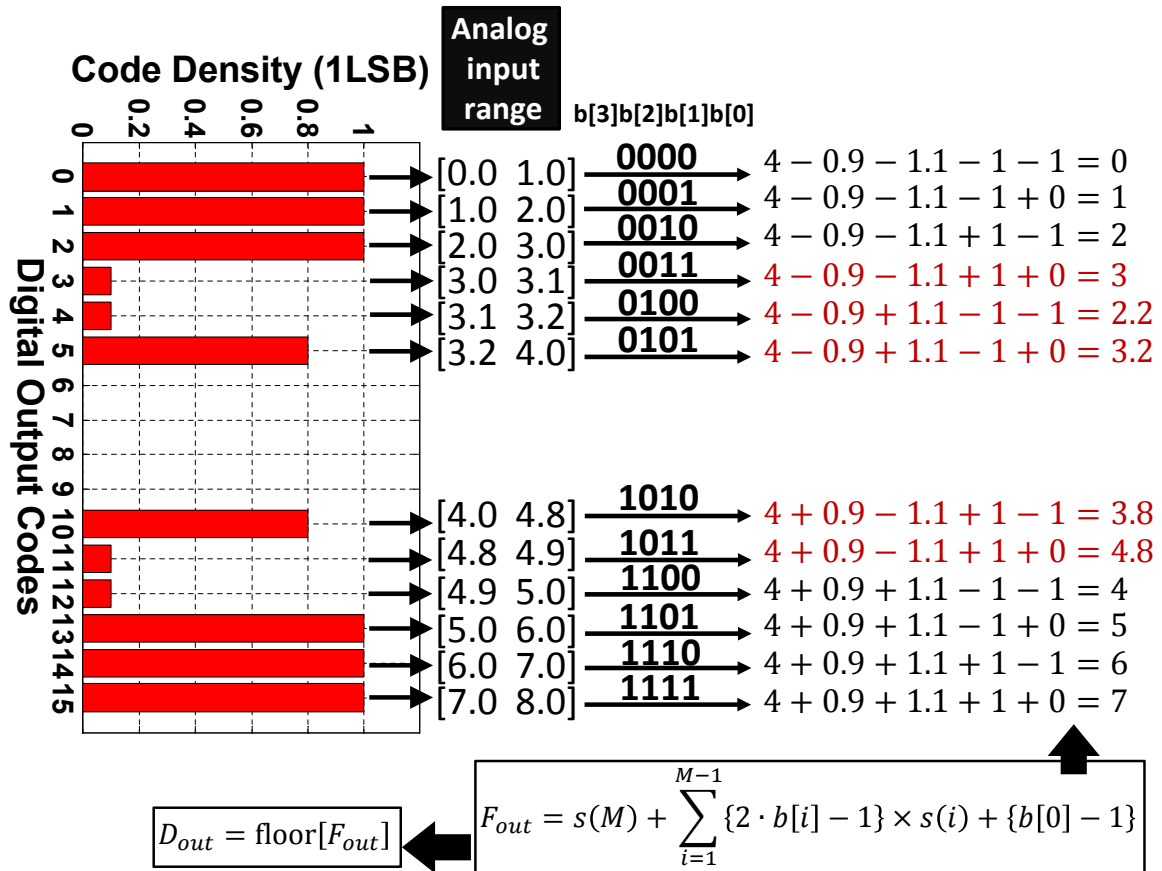


Figure 4-11: Calculating D_{out} when step sizes are fractional. Since not all the code bins have 1 LSB bin width, there is static nonlinearity in this ADC.

from Figure 4-11, where F_{0100} , F_{0101} , F_{1010} and F_{1011} are all fractional values. Another observation, which is different from the integer case, is that F_{out} of a bin is not always the same as the lower bound of its analog input range. Taking bin “0100” as an example, the lower bound of its input range is 3.1, but F_{0100} is calculated to be 2.2. This can be understood by observing the progressions of the decision levels during the search process. For bin “0100” the progressions are $4 \mapsto 3.1 \mapsto 4.2 \mapsto 3.2 \mapsto 2.2$. The four comparisons imply that the input of this bin is less than 4, greater than 3.1, less than 4.2 and less than 3.2. Since the last comparison states that the input is less than 3.2, Equation 4.19 calculates F_{0100} to be equal to the last decision level minus 1. This is because this formula does not know the bin width beforehand and therefore, it has to make an assumption on the bin width when calculating F_{out} ; it is only natural to assume a bin has a nominal width of 1 LSB for this purpose.

$$\left\{ \begin{array}{l}
 S(4) - S(3) - S(2) - S(1) - S(0) \cdot 1 = F_{0000} = 0 \\
 S(4) - S(3) - S(2) - S(1) + S(0) \cdot 0 = F_{0001} = 1 \\
 S(4) - S(3) - S(2) + S(1) - S(0) \cdot 1 = F_{0010} = 2 \\
 S(4) - S(3) - S(2) + S(1) + S(0) \cdot 0 = F_{0011} = 3 \\
 S(4) - S(3) + S(2) - S(1) - S(0) \cdot 1 = F_{0100} = 3.1 \\
 S(4) - S(3) + S(2) - S(1) + S(0) \cdot 0 = F_{0101} = 3.2 \\
 S(4) + S(3) - S(2) + S(1) - S(0) \cdot 1 = F_{1010} = 4 \\
 S(4) + S(3) - S(2) + S(1) + S(0) \cdot 0 = F_{1011} = 4.8 \\
 S(4) + S(3) + S(2) - S(1) - S(0) \cdot 1 = F_{1100} = 4.9 \\
 S(4) + S(3) + S(2) - S(1) + S(0) \cdot 0 = F_{1101} = 5 \\
 S(4) + S(3) + S(2) + S(1) - S(0) \cdot 1 = F_{1110} = 6 \\
 S(4) + S(3) + S(2) + S(1) + S(0) \cdot 0 = F_{1111} = 7
 \end{array} \right. \quad (4.23)$$

We follow the same extraction procedure as in the integer case and generate an equation for each code bin. The right hand side of the equation (F_{out}) is still ob-

tained by accumulating the bin width from the LSB to the MSB bins, even though we know in this case, we cannot estimate the actual value of F_{0100} from the lower bound of the input range as described by the F_{0100} example previously. The list of equations is shown in 4.23. By subtracting the neighboring equations, we can obtain a similar matrix as before. The actual step sizes can then be solved using the solution for linear least square as shown in Equation 4.24. As expected, the solution $[1.12, 1.22, 0.83, 0.67]$ does not match the actual step sizes $[0.9, 1.1, 1, 1]$ since the estimation of F_{out} using the lower bound of a input range is not accurate if the step sizes are fractional.

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & -1 \\ 0 & 0 & 0 & 1 \\ 0 & 2 & -2 & -1 \\ 0 & 0 & 0 & 1 \\ 2 & -2 & 2 & -1 \\ 0 & 0 & 0 & 1 \\ 0 & 2 & -2 & -1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} S(3) \\ S(2) \\ S(1) \\ S(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0.1 \\ 0.1 \\ 0.8 \\ 0.8 \\ 0.1 \\ 0.1 \\ 1 \\ 1 \end{bmatrix} \quad \mapsto \quad \begin{bmatrix} S(3) \\ S(2) \\ S(1) \\ S(0) \end{bmatrix} = \begin{bmatrix} 1.12 \\ 1.22 \\ 0.83 \\ 0.67 \end{bmatrix} \tag{4.24}$$

This problem can be fixed with a simple observation. Comparing the F_{out} estimated using the lower bound of the input range and the actual F_{out} calculated using Equation 4.23, we can see that the bins with the smaller bin width have larger discrepancy compared to the bins with larger bin width. For example, F_{out} 's are calculated incorrectly for bins "0100," "1010" and "1100". Among these bins, the largest bin ("1010") with width 0.8 LSB estimates F_{out} to be 4 LSB from the cumulative histogram while the actual F_{out} is equal to 3.8 LSB, making an error of 0.2 LSB. The smaller bin ("0100") with width 0.1 LSB estimates F_{out} to be 3.1 from the cumulative histogram while the actual F_{out} is equal to 2.2 LSB, making an error of 0.9 LSB.

This result can be understood intuitively. Equation 4.18 assumes that all the step sizes are integer multiples of each other and all the bins have ideal width of 1 LSB. If the bin has a width of Δ , which is different from 1, the estimated F_{out} can be $1 - \Delta$ outside its desired analog input range. The parameter “ $1 - \Delta$ ”, therefore, also represents the minimum error we make while estimating F_{out} using the cumulative histogram. When the step sizes are designed correctly to sufficiently cover the manufacturing variation, $\Delta \leq 1$. In the ideal case when $\Delta = 1$, $\Delta - 1 = 0$ and the estimated F_{out} experiences little error; in the case when Δ is small and $1 - \Delta$ is large, the estimated F_{out} tends to experience larger error.

The goal here is to find a better estimate of F_{out} , rather than always using the lower bound of an input range. Define the lower bound of an input range, R_L , and the upper bound of an input range, R_H . From the previous discussion on the progression of decision levels, when the last decision bit is a “0,” the final decision level (F_{out}) is larger than the input signal, meaning that F_{out} should be closer to (or equal to) R_H of the input range; when the last decision bit is a “1,” on the other hand, the final decision level (F_{out}) is closer to (or equal to) R_L of the input range. As a result, when $b[0] = 0$, it is more accurate to use R_H to estimate F_{out} ; when $b[0] = 1$, it is more accurate to use R_L to estimate F_{out} . The final expression is given in Equation 4.25.

$$F_{out} = \begin{cases} R_L & \text{if } b[0] = 1 \\ R_H - 1 & \text{if } b[0] = 0 \end{cases} \quad (4.25)$$

The new approach here makes two modifications to the previous approach. First, since the bins with smaller width are more likely to create a larger error while estimating F_{out} from the code density, we do not use any equations associated with the bins that have bin width smaller than a pre-determined threshold, β . Second, for the rest of the bins, we estimate F_{out} according to Equation 4.25. Note that when a bin has size equal to 1 LSB, the two expressions in Equation 4.25 lead to the same result.

$$\left\{ \begin{array}{l}
S(4) - S(3) - S(2) - S(1) - S(0) \cdot 1 = F_{0000} = 0 \\
S(4) - S(3) - S(2) - S(1) + S(0) \cdot 0 = F_{0001} = 1 \\
S(4) - S(3) - S(2) + S(1) - S(0) \cdot 1 = F_{0010} = 2 \\
S(4) - S(3) + S(2) - S(1) + S(0) \cdot 0 = F_{0101} = 3.2 \\
S(4) + S(3) - S(2) + S(1) - S(0) \cdot 1 = F_{1010} = 3.8 \\
S(4) + S(3) + S(2) - S(1) + S(0) \cdot 0 = F_{1101} = 5 \\
S(4) + S(3) + S(2) + S(1) - S(0) \cdot 1 = F_{1110} = 6 \\
S(4) + S(3) + S(2) + S(1) + S(0) \cdot 0 = F_{1111} = 7
\end{array} \right. \quad (4.26)$$

As an example, we can set $\beta = 0.6$ and re-write Equation 4.23 according to the two new rules. The resulting list of equations is given in 4.26. Following the same procedure by subtracting the i^{th} equation from the $(i + 1)^{th}$ equation, we can again put the list of equations into matrix form, and the matrix can be solved by linear least square solution. In this case, we are able to obtain the correct step sizes as shown in Equation 4.27.

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & -1 \\ 0 & 2 & -2 & 1 \\ 2 & -2 & 2 & -1 \\ 0 & 2 & -2 & 1 \\ 0 & 0 & 2 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} S(3) \\ S(2) \\ S(1) \\ S(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1.2 \\ 0.6 \\ 1.2 \\ 1 \\ 1 \end{bmatrix} \quad \mapsto \quad \begin{bmatrix} S(3) \\ S(2) \\ S(1) \\ S(0) \end{bmatrix} = \begin{bmatrix} 0.9 \\ 1.1 \\ 1 \\ 1 \end{bmatrix} \quad (4.27)$$

4.4.3 Unknown Input Statistics

As discussed in the earlier part of this section, this calibration algorithm is not limited to only inputs with uniform distribution. For input signals with non-uniform

probability density, the histogram is locally normalized over a small input range for which the probability density of input remains reasonably constant.

When the input has non-uniform probability distribution, the number of code counts that represent the same bin width can be quite different. For example, in an area with high probability density (P_0), a 1 LSB bin may correspond to code count of 1000, but in another area with lower probability density ($P_0/2$), a code bin representing a 1 LSB bin width may only have code count of 500. Thus, to normalize locally in each region of the code, our approach is to find the code count that represents a reference bin width with size 1 LSB, and divide the neighboring bins by the counts of this reference bin. We call the resulting bin width after normalization, the true bin width.

If the step sizes are designed correctly with enough redundancy to accommodate for mismatches, all bins should have width $\Delta \leq 1$ LSB; otherwise, the ADC would experience static nonlinearity. In other words, while looking at any input ranges, the bins with the highest code count typically represent the true bin width of 1 LSB since that is the largest that Δ can go. As a result, before applying the calibration algorithm to a code density, the algorithm goes through each region of the code and finds the bins with the highest number of counts. These counts are then averaged together to provide an estimate for the counts of a 1 LSB bin. The neighboring bins are then normalized by this estimated number.

4.4.4 Calibration Examples

The new calibration algorithm is simulated under many different conditions. Shown here is a 12-bit 16-step ADC. The unit step size is assumed to have random variation with a standard deviation of 20%. The variance of the larger step sizes are scaled up proportionally. The circuit is simulated with random circuit noise that has standard deviation of 0.5 LSB. The algorithm is tested with a wide range of inputs including a sine wave, ramp, uniformly random and other inputs with smooth probability distribution function. With no assumption or prior knowledge of the input signal, the local normalization technique is able to have consistent performance over a variety range of

input distributions as long as enough counts are collected over the full scale. We do observe one limitation in our approach; an input with zero probability at particular codes is problematic, since the algorithm has no way to distinguish whether the zero comes from the input or from the ADC itself. Applications of such zero probability input characteristics would not be a good candidate for this calibration algorithm.

For this particular example, the input is simulated with a sine wave that runs at 24.7MHz with 50MS/s sampling rate. The amplitude of the sine wave is 1.1 times the full scale. A total of 2^{20} samples are collected during simulation. Table 4.5 shows good matching of the actual step sizes and the extracted step sizes even in the presence of noise. The maximum absolute error in extracting the step sizes is 0.15 LSB_{12} . Figure 4-12 shows the static performance of the ADC before and after calibration. The DNL improves from $+2.53/ - 1.00 \text{LSB}_{12}$ to $\pm 0.56 \text{LSB}_{12}$ and the INL improves from $+7.8/ - 7.9 \text{LSB}_{12}$ to $\pm 0.6 \text{LSB}_{12}$. Figure 4-13 shows the dynamic performance of the ADC before and after calibration. The spurious-free dynamic range goes from 59.5dB to 92dB, the SNR goes from 55dB to 71.6dB, the THD goes from -58.9dB to -87.2dB and ENOB improves from 8.6b to 11.6b. We do not achieve perfect ENOB because of the added circuit noise, which reflects the condition of real operation.

4.4.5 Comparisons of the Calibration Algorithms

In this chapter, we have introduced three new background calibration algorithms. All three algorithms do not require significant extra hardware on-chip to perform the calibration. The first algorithm requires knowing the exact input waveform, and the resolution of the extraction depends on how accurately the input signal is known. This algorithm uses the minimum and maximum voltages around the decision boundaries to estimate the step sizes. Even though it is able to extract the actual step sizes correctly when provided with a high precision input signal, this scheme is sensitive to the presence of circuit noise since noise can blur the boundary significantly.

The second calibration algorithm improves upon the previous algorithm by using the statistics of the input signal rather than the absolute magnitude. Even though this calibration algorithm does not require the knowledge of the input signal to a

M	S_{design}	Actual S	Extracted S with $N_{std} = 0.5$	Differences
15	1820	1815.94	1816.09	-0.15
14	1050	1045.54	1045.57	-0.03
13	560	560.69	560.58	0.11
12	280	284.05	284.00	0.05
11	140	139.41	139.38	0.03
10	105	106.99	107.10	-0.11
9	70	71.71	71.69	0.02
8	35	34.92	35.04	-0.12
7	16	17.36	17.42	-0.06
6	8	8.26	8.29	-0.03
5	6	4.89	4.92	-0.03
4	2	2.03	2.03	-0.00
3	2	1.96	1.83	0.13
2	1	1.18	1.05	0.13
1	1	1.06	1.00	0.07

Table 4.5: Estimation of step sizes using the third calibration algorithm. The difference between the actual and the estimated step sizes are small, with largest difference equal to 0.15. This shows the effectiveness of the calibration algorithm even in the presence of circuit noise.

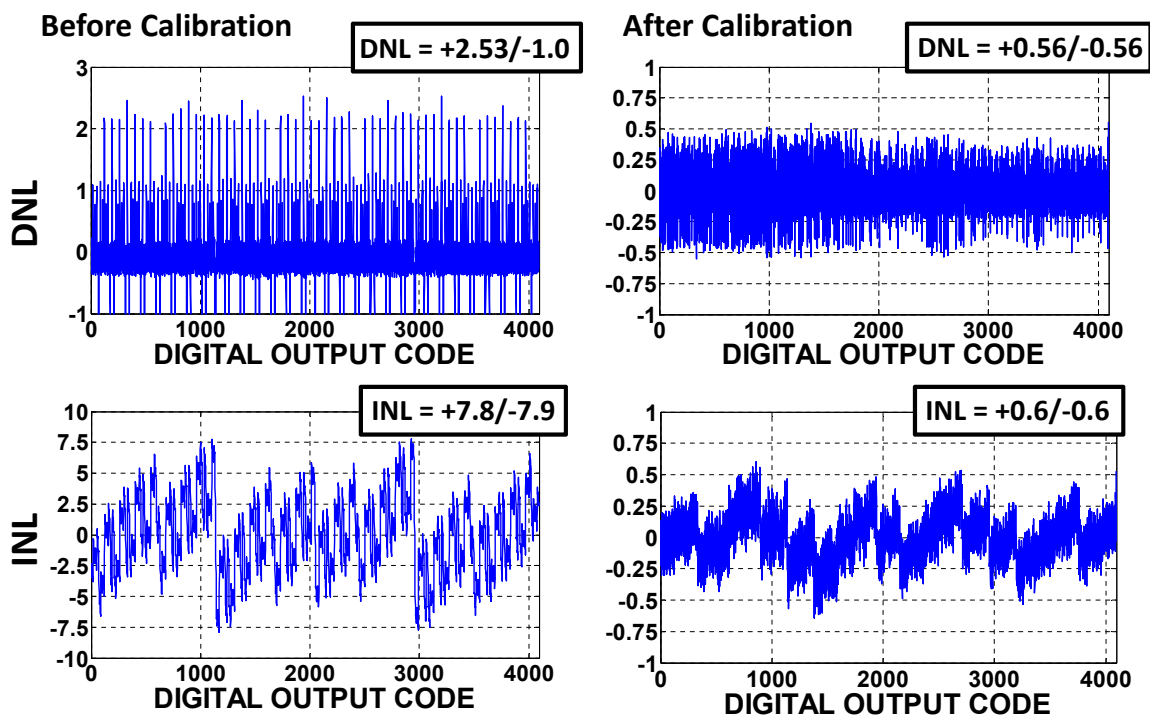
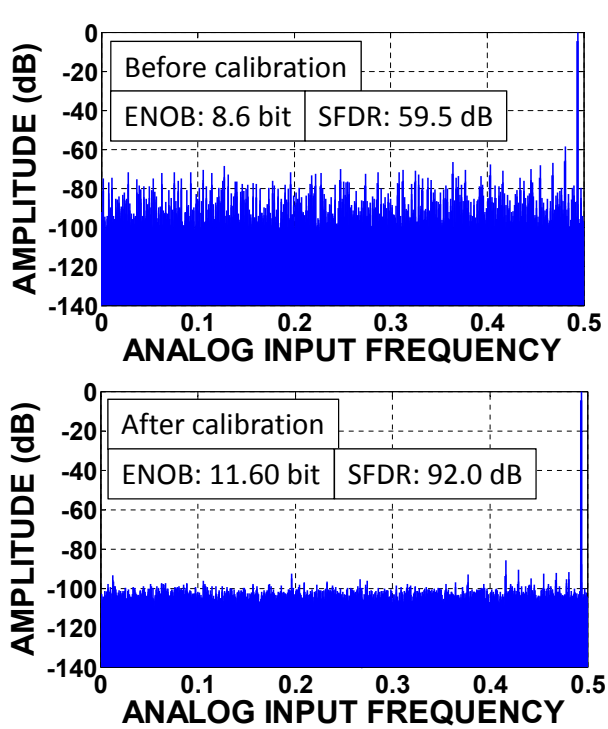


Figure 4-12: Static nonlinearity before and after using the third calibration algorithm. The DNL improves from $+2.53/-1.0$ LSB_{12} to $+0.56/-0.56$ LSB_{12} ; the INL improves from $+7.8/-7.9$ to $+0.6/-0.6$ LSB_{12} .



	Before Calibration	After Calibration
2 nd	-95.1	-92.4
3 rd	-59.5	-92.4
4 th	-83.7	-99.0
5 th	-67.6	-92.0
6 th	-79.6	-103.6
7 th	-69.0	-93.7
8 th	-82.6	-97.0
9 th	-84.2	-95.1
THD	-58.9	-87.2
SNR	55.0	71.6
SFDR	59.5	92.0
SNDR	53.3	71.5
ENOB	8.6	11.6

Figure 4-13: Spectrum data before and after using the third calibration algorithm. The ENOB improves from 8.6b to 11.6b; and the SFDR improves from 59.5dB to 92.0dB.

high precision, it requires the knowledge of the exact statistics of the input waveform. A sine wave input was used as an example for calibration. Since this calibration algorithm is statistically-based, circuit noise can be averaged out when many samples are collected during simulation. The result shows good extraction precision in the presence of noise.

The third calibration algorithm further improves upon the previous two algorithms. It also uses statistics for calibration as in the case of the second calibration algorithm; however, it does not require the knowledge of the exact statistics of the waveform. The input waveform only needs to have a smooth probability distribution with no zero probability of any codes. A local normalization technique is developed to take in a code density map and normalize it such that the code density resembles the code density of a ramp. Because it is also a statistically based algorithm, circuit noise again can be averaged out as more data is collected. The calibration is tested with different input signals without prior assumption on the exact probability distribution. The results show improvements in both static and dynamic performance. We use the third calibration algorithm on real silicon data in Chapter 6.

Chapter 5

Design and Implementation of a SAR ADC with Redundancy

In Chapter 4, we introduced three new calibration algorithms that are able to utilize the redundancy information to digitally correct and remove variation in the step sizes. The effectiveness of the calibration scheme was explained in theory first, followed by a case study. The case study validates the methodology in simulation, while also comparing the calibration schemes in terms of their requirements on input waveforms, accuracy of the extraction procedure, and tolerance to circuit noise.

The first algorithm requires knowing the input signal to a high precision in order to extract the step sizes to the same precision. Even though this algorithm is able to extract the step sizes accurately with good knowledge of the input signals, it is proven to be impractical due to the precision requirement on the input signal and its sensitivity to thermal noise. The second algorithm uses a statistically-based approach for extraction. The extraction does not rely on knowing the input signals to a high precision, but rather, the precision and noise tolerance can be improved by sampling more times. However, this algorithm still requires knowing the exact statistics of the input waveform. The third calibration algorithm is able to improve upon the deficiencies of the previous two algorithms by not requiring the knowledge of the input signal or the input statistics. Any input signal with smooth probability distribution functions can be used as stimuli. The third calibration scheme is able to achieve the

same extraction precision in the presence of thermal noise without knowing the exact input values or distribution.

Thus far, our discussion has focused on the benefits of incorporating redundancy in the SAR architecture and how to use this redundant information for digital calibration. All the analysis is based on the assumption that redundancy described previously can be implemented in hardware. In this chapter, we the real implementation of a redundant SAR ADC is described. In the first part, we will focus on the implementation at the architectural level with discussion of several new contributions. First, we develop a new DAC switching scheme that is able to achieve higher energy efficiency while removing some of the shortcomings in previous switching schemes. Second, we improve the original design of the split-capacitor architecture and eliminate the over-range problems while achieving better matching properties. Third, we introduce a new way to incorporate redundancy into the SAR architecture without increasing the design complexity and area overhead. This simple solution is able to maintain symmetric error-tolerance windows as well.

The next part of the chapter discusses the design at the circuit level. We begin by deriving the high-level requirements on various circuit blocks. We specifically focus on the noise, mismatch, bandwidth and timing requirements in order to achieve the target speed, resolution, and asynchronous operation. We then dive further into analyzing the design of each individual circuit component, including the design of sampling circuit, comparators, pulse generators, timing circuits, switches, clock generation, and the capacitive DAC. We conclude the chapter by combining the circuit blocks and analyzing how all these blocks work together.

5.1 Architecture

The design of our new SAR architecture is divided into several parts. The first part of this section discusses the evaluation of the energy consumption in different switching algorithms and compares them to our new switching scheme in terms of different figures of merit. We are able to show that the new switching algorithm retains

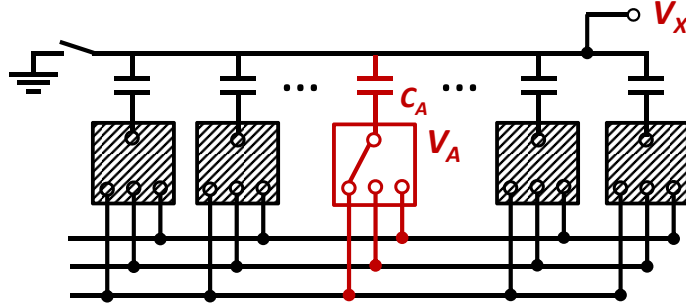


Figure 5-1: Energy consumption when charges change on capacitor C_A .

all the benefits of previous switching schemes [39–43], but is able to achieve better energy efficiency. The second part reviews the conventional split-capacitor array architecture and provides new solutions to previous limitations. These limitations include matching and over-range problems. The third part of this section shows how redundancy can be implemented in the SAR architecture with minimal additional hardware complexity.

5.1.1 Energy Consumption in Switching Scheme

Conventional Switching Algorithm

The DAC in a SAR ADC serves two purposes: it samples the input voltage, and it generates error residues between the input and the current digital estimate. Figure 5-2 shows the conventional SAR switching algorithm for a 3-bit ADC in a fully-differential implementation; Figure 5-3 shows the top-plate waveform for a 6-bit ADC using the conventional switching algorithm when input is equal to 0.9 with $V_{IN+} = 0.95$, $V_{IN-} = 0.05$ and $V_{REF} = 1.0$. Even though this switching algorithm is able to produce the correct logic operations by moving the charges between the capacitors according to the value of the input signal, the scheme does not move the charges around efficiently, wasting energy during operation. Figure 5-1 explains a sampled DAC array that shows how much total energy is consumed when charges are moved between capacitors.

Assume at time 0^+ , the bottom plate of C_A is switched from $V_{A,ini}$ to $V_{A,final}$. V_X is initially at $V_X(0)$ at time 0^- . When the bottom plate of the C_A capacitor switches,

the voltage on V_X settles completely by time $t = T$. The total energy drawn from voltage source V_A can be calculated using Equation 5.1.

$$E = \int_{0^+}^T i_A(t) \times V_A dt = V_A \int_{0^+}^T i_A(t) dt \quad (5.1)$$

Since $i_{REF}(t) = -dQ_{C_A}/dt$, Equation 5.1 can be simplified to

$$\begin{aligned} E &= -V_A \int_{0^+}^T \frac{dQ_{C_A}}{dt} dt = -V_A \int_{Q_{C_A}(0^+)}^T dQ_{C_A} \\ &= -V_A \times C_A \times (Q_{C_A}(T) - Q_{C_A}(0^+)) \\ &= -V_A \times C_A \times ((V_X(T) - V_{A,final}) - (V_X(0) - V_{A,init})) \end{aligned} \quad (5.2)$$

Following the same procedure, we can calculate the energy consumption of each transition in Figure 5-2. During the first phase, the differential inputs are sampled onto the upper and lower arrays of the DAC, respectively. After sampling, the input is disconnected from the DAC. The MSB capacitor of the DAC is charged to V_{REF} and the remaining capacitors are charged to ground for the top array. For the bottom array, the opposite is done. The total energy consumption for this operation is $4CV_{REF}^2$.¹ The first bit decision is generated by comparing the voltage on the *plus* and *minus* nodes of the comparator. Depending on whether the bit is a “0” or a “1,” the switching scheme either takes the “up” or “down” transitions, respectively. If it takes the “up” transition, the total energy consumption is CV_{REF}^2 ; if it takes the “down” transition, the total energy consumption is $5CV_{REF}^2$.

Observing the first two transitions, we see a few potential areas where energy efficiency can be improved. For this simple example, the first transition compares the magnitude of V_{IN+} and V_{IN-} and generates the sign bit of the input signal. There are a total of four potential transition paths that the SAR algorithm can take depending on the values of the input signal, as shown in Figure 5-2. Taking the upper most path

¹To be more rigorous in this case, V_{REF}^2 should be replaced by $V_{REF} \times V_{DD}$. This is because due to noise reasons, V_{REF} is generated from a linear regulator using V_{DD} in most systems. Therefore, the power drawn from V_{DD} must be considered, not power drawn from V_{REF} and V_{REF}^2 should be replaced by $V_{REF} \times V_{DD}$. In this thesis, for simplicity, we just use V_{REF}^2 .

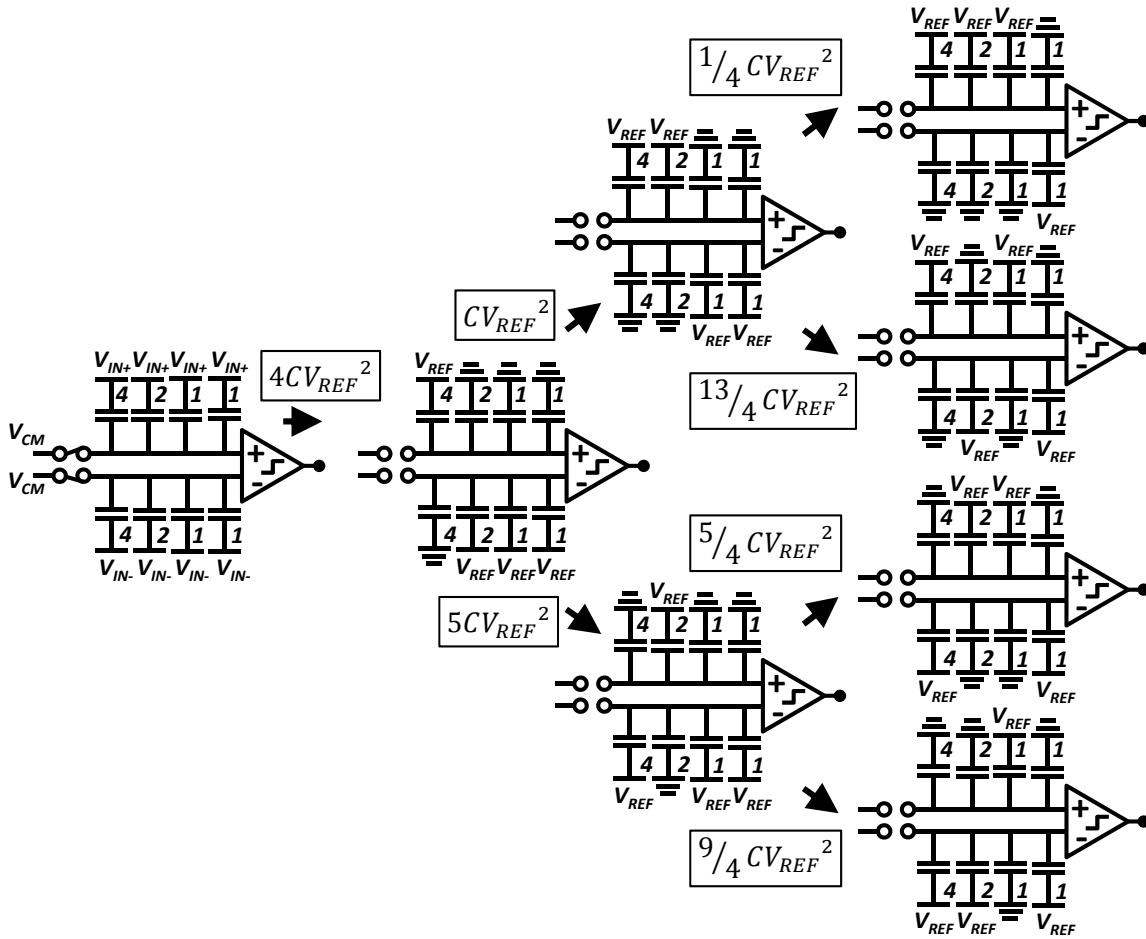


Figure 5-2: Conventional SAR switching algorithm, showing energy consumption related to capacitor switching transitions.

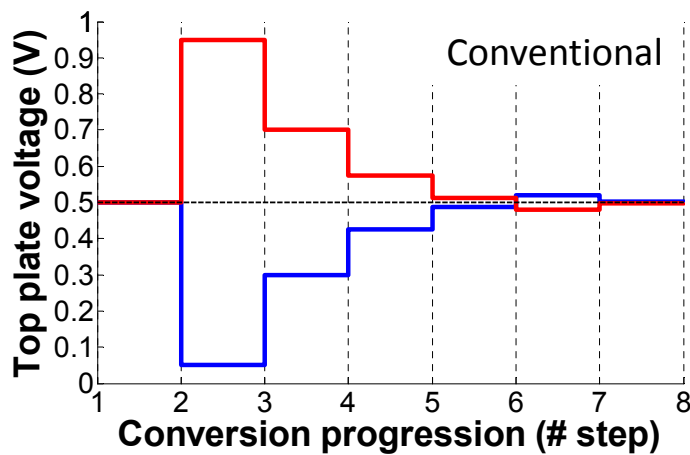


Figure 5-3: The top-plate waveform when using the conventional switching algorithm. The input is assumed to have magnitude equal to 0.9 with $V_{IN+} = 0.95$, $V_{IN-} = 0.05$ and $V_{REF} = 1.0$. The final output bit sequence is 111100.

as an example, the first step makes up more than 75% of the total energy consumption to just generate the sign bit. Intuitively, the sign bit can be generated by comparing V_{IN+} and V_{IN-} directly after sampling without consuming any energy. It implies that simpler methods can be developed to replace the present first step to avoid this energy loss.

We also observe that the energy consumption in “up” and “down” transitions is greatly imbalanced during the second phase. The “down” transition requires $5CV_{REF}^2$, which is five times more energy compared to the “up” transition. This phenomenon can be understood intuitively. The discussion from now on will only refer to the upper capacitive DAC, since the bottom half is just the complementary opposite of the upper half. During an “up” transition, only capacitor 2 is charged to generate the next digital estimate; during a “down” transition, however, not only capacitor 2 is charged, the previously charged capacitor 4 also is discharged back to ground. The inefficiency exists in that the charge in capacitor 4 is not recycled and used to charge capacitor 2. In other words, during this transition, the previously stored energy in capacitor 4 is lost and new energy is drawn from the supply to charge capacitor 2. The average switching energy of an n -bit conventional switching algorithm can be derived as follows:

$$E_{conv} = \sum_{i=1}^n 2^{n+1-2i}(2^i - 1)CV_{REF}^2 \quad (5.3)$$

Methods have been investigated to find ways to reuse the previously stored energy to charge the later capacitors. Several attempts in the past [40–43] have successfully improved energy efficiency compared to the conventional switching algorithm. In the following sections, we will first review a few representative works that show how energy efficiency has improved. It will be followed by a new energy switching algorithm and its comparison with prior art.

Split-Capacitor Switching Algorithm

Ginsburg *et al.* in [40] proposed a switching scheme to solve the imbalanced energy consumption in the “up” and “down” transitions. The scheme modifies the conven-

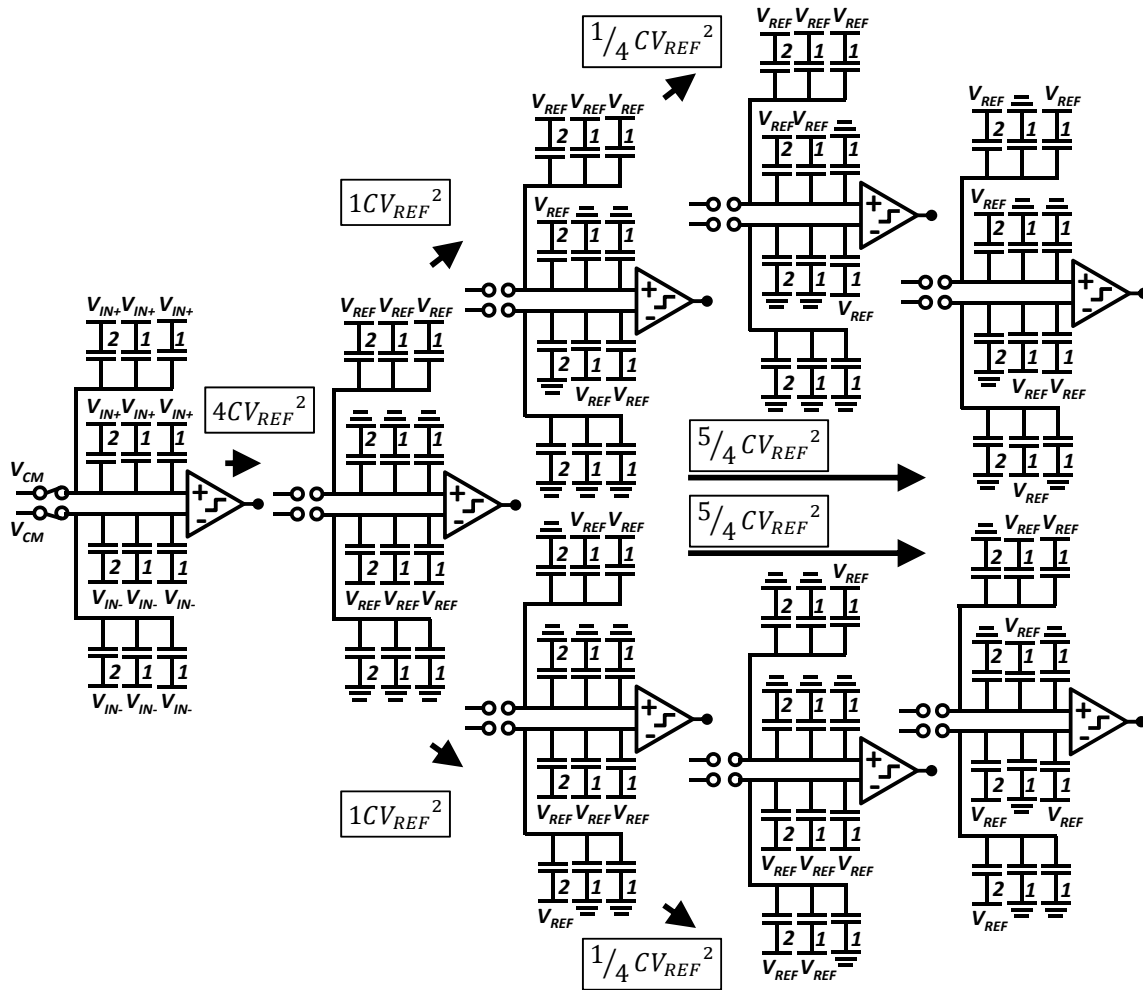


Figure 5-4: Split-capacitor switching algorithm, showing reduced energy consumption compared to Figure 5-2.

tional algorithm by splitting the *MSB* capacitor into a binary weighted sub-capacitor array. As shown in Figure 5-4, the original capacitor 4 is split into binary weighted sub-capacitors, [2, 1, 1]. During the first bit cycle, the *MSB* capacitor array is charged to V_{REF} , while the remaining capacitors are connected to ground. During the next bit cycle, if it is an “up” transition, a capacitor of size 2 is charged to V_{REF} , similar to what is done in the conventional case. If it is a “down” transition, instead of discharging capacitor 4 and charging capacitor 2, a capacitor of size 2 is available for discharging. In the conventional implementation, this capacitor is not available.

For any subsequent “up” transition, a capacitor in the original array is connected to V_{REF} and for any subsequent “down” transition, a capacitor in the *MSB* array is connected to ground. This can avoid any previously charged capacitor from discharging during the “down” transition as in the conventional design. Even though this approach requires twice as many switches and more complex switching algorithm, the average switching energy is reduced by 38% compared to previous implementation. The average switching energy of an n -bit split-capacitor switching algorithm can be derived as follows

$$E_{split-cap} = 2^{n-1} + \sum_{i=2}^n 2^{n+1-2i}(2^{i-1} - 1) \quad (5.4)$$

Splitting the *MSB* capacitor into a binary weighted sub-capacitor array does not change the voltage transitions on the top plates so the top-plate waveform in this case is the same as the one given in Figure 5-3.

Energy Saving Switching Algorithm

Chang *et al.* in [41] proposed an energy-saving switching algorithm to further reduce the average energy consumption. The algorithm is shown in Figure 5-5 and the top-plate waveform is shown in Figure 5-6. It modifies the conventional algorithm such that rather than consuming $4CV_{REF}^2$ during the first bit decision cycle, this algorithm consumes no power during this cycle. It is achieved by setting the initial voltage on the top plate of the DAC to V_{REF} instead of V_{CM} as in a conventional switching. With

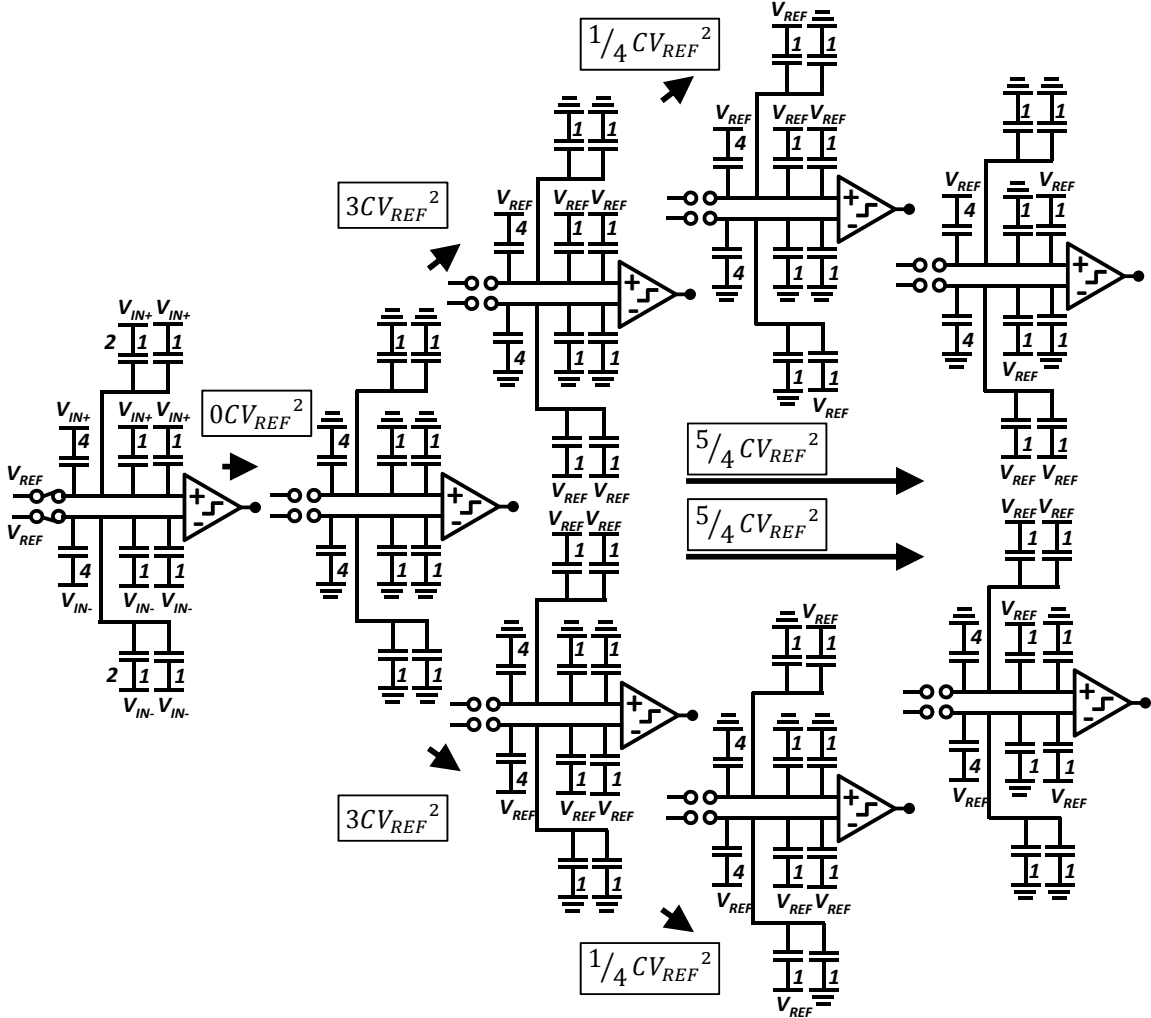


Figure 5-5: Energy-saving switching algorithm, showing reduced energy consumption compared to Figures 5-2 and 5-4.

this change, during the first transition, all the bottom-plate voltages are discharged to ground, which consumes no energy. Moreover, this switching scheme merges the previous split-capacitor technique [40] into its switching scheme. One difference is that in this case, rather than splitting the *MSB* capacitor, it splits the *MSB* – 1 capacitor into a binary weighted sub-capacitor array to reduce the energy consumption during the “down” transition of each switching phase. As a result, it consumes 57% less energy compared to the conventional implementation. The average switching energy of an *n*-bit energy saving algorithm can be derived as follows

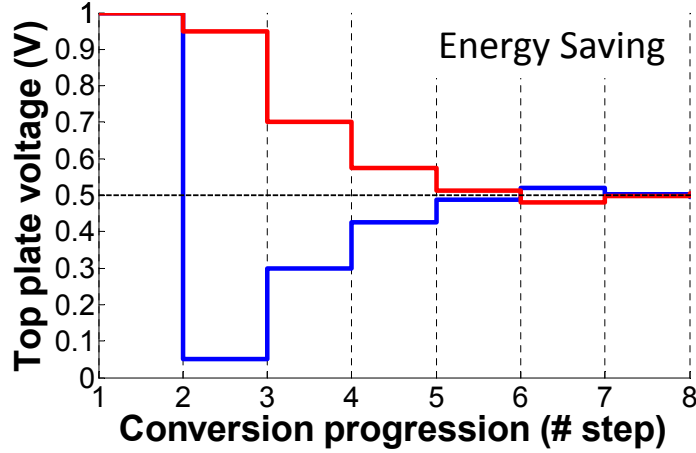


Figure 5-6: The top-plate waveform when using the energy-saving switching algorithm. The input is assumed to have magnitude equal to 0.9 with $V_{IN+} = 0.95$, $V_{IN-} = 0.05$ and $V_{REF} = 1.0$. The final output bit sequence is 111100. Comparing the top-plate waveform in a conventional algorithm and in a energy-saving algorithm, they are differed in that in a conventional switching algorithm, the top-plate voltage begins with V_{CM} , but in the energy-switching algorithm, the top-plate voltage begins with V_{REF} .

$$E_{energy-saving} = 3 \cdot 2^{n-3} + \sum_{i=3}^n (2^{i-1} - 1) CV_{REF}^2 \quad (5.5)$$

Note that in the previous two switching algorithms, the voltages on the top plates of the DAC begin and return to the same voltage V_{CM} , such that the charges across these parasitic capacitors are the same at both time instances. From the perspective of the switching scheme, these parasitic capacitors are thus transparent to the DAC operation. In this implementation, however, the top plates begin at voltage V_{REF} , and return to V_{CM} at the end of the conversion process as shown in Figure 5-6; as a result, the parasitic capacitances on the top plates affect the conversion accuracy. If the parasitic capacitors stay constant and the capacitances do not vary with the voltage across the capacitors, it would only introduce a gain error. On the other hand, if the parasitic capacitors behave like a varactor and the capacitances vary with the voltage across the capacitors, the effect becomes signal-dependent, which introduces harmonic distortion at the output spectrum.

Monotonic Switching Algorithm

Liu *et al.* in [42] proposed a monotonic switching algorithm as shown in Figure 5-7. The switching sequence of this approach only has discharging with no charging operation. For an n -bit ADC, instead of requiring a total of 2^n capacitors, it only requires 2^{n-1} , which is half of the previous requirement. This is because the monotonic switching algorithm does not need the *MSB* capacitor. This switching algorithm is able to tackle both potential areas that could improve energy efficiency. First, the sign bit is generated by directly comparing V_{IN+} and V_{IN-} without consuming any energy. Second, it does not include any operation that requires charging up a capacitor and discharging the same capacitor in the later phase. As a result, it consumes 81% less energy compared to the conventional implementation without splitting or adding additional switches. The average switching energy of an n -bit monotonic switching algorithm can be derived as follows:

$$E_{monotonic} = \sum_{i=1}^{n-1} 2^{n-2-i} C V_{REF}^2 \quad (5.6)$$

This algorithm also has a similar problem as in the energy-saving switching scheme, in that the top plates do not begin and return to the same voltage; thus the accuracy of this scheme would be affected by the parasitic capacitances of the top plates. Another drawback of this scheme is that the common-mode voltage of the DAC decreases from V_{CM} towards ground during the conversion as shown in Figure 5-8. This changes the common-mode of the input of the comparator and changes the comparator offset during the conversion, which can degrade the achievable linearity of the ADC.

Merged Capacitor Switching Algorithm

Hariprasath *et al.* in [43] proposed a merged capacitor switching (MCS) based SAR ADC as shown in Figure 5-9. During the first bit cycle, the bottom plate of the capacitive DAC is connected to V_{CM} while the top plate is connected to the input signal, V_{IN} . The first sign bit can be determined immediately after sampling without

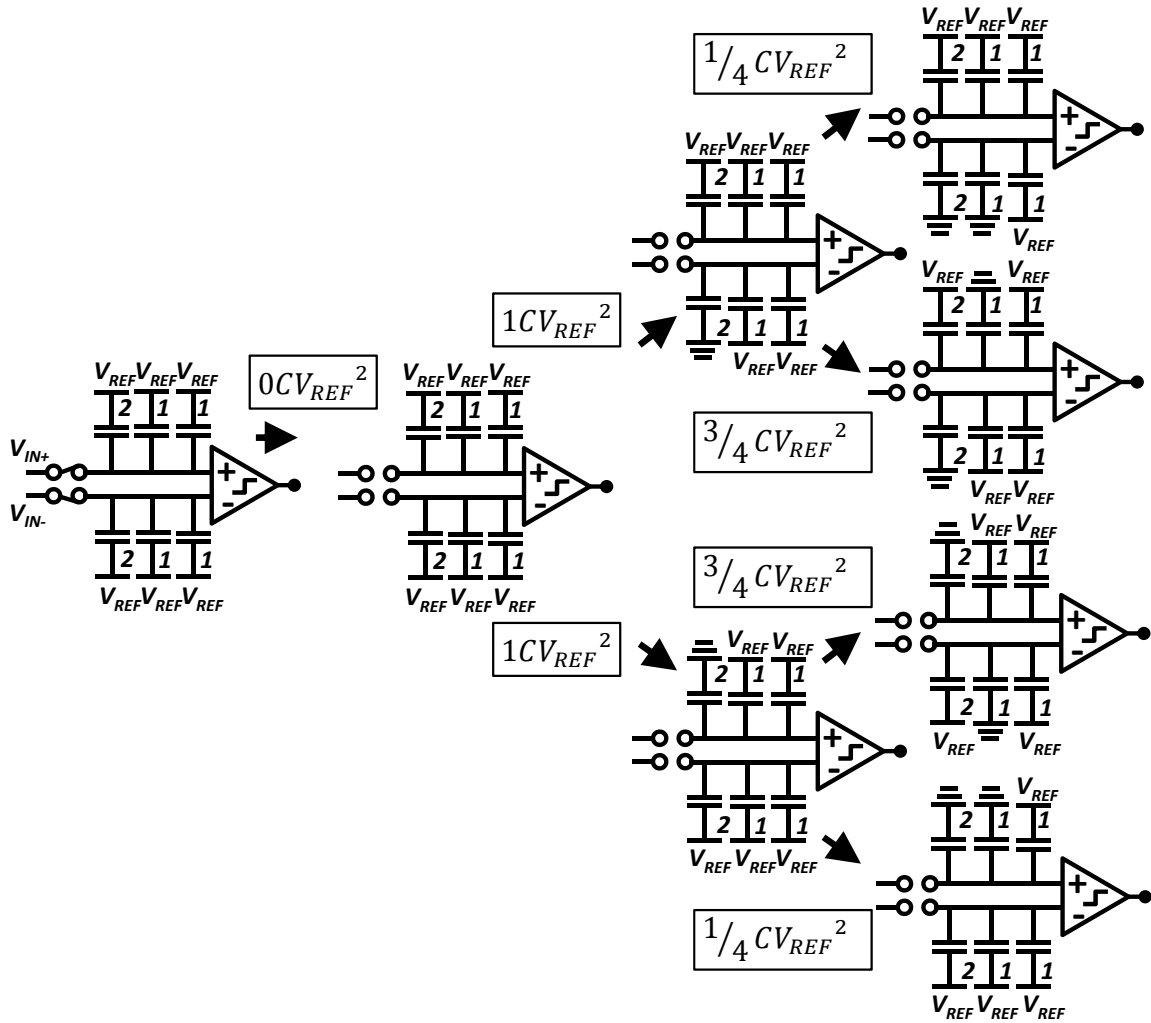


Figure 5-7: Monotonic switching algorithm, showing reduced energy consumption compared to Figures 5-2, 5-4 and 5-5.

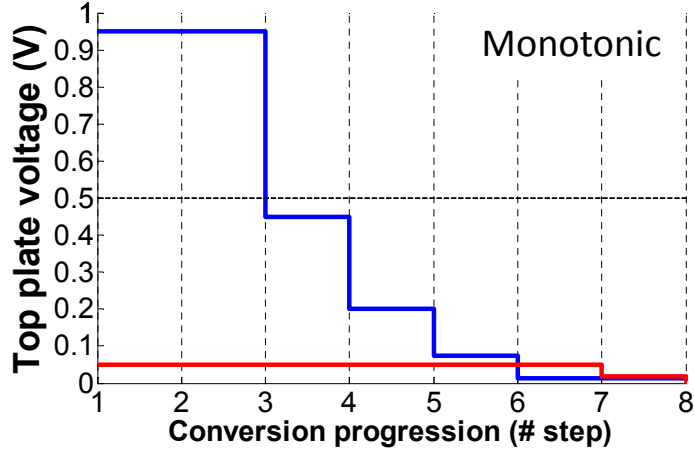


Figure 5-8: The top-plate waveform when using the monotonic switching algorithm. The input is assumed to have magnitude equal to 0.9 with $V_{IN+} = 0.95$, $V_{IN-} = 0.05$ and $V_{REF} = 1.0$. The final output bit sequence is 111100. Rather than converging towards V_{CM} at the end of the conversion progress, the top plate voltages of the upper/lower DACs both converge to ground.

consuming any energy from the capacitor array. Depending on the bit value of the first comparison, the next bit cycle either charges capacitor 2 from V_{CM} to V_{REF} or discharges capacitor 2 from V_{CM} to ground.

If we compare this switching algorithm with the monotonic switching scheme, there are a few similarities and differences. First, both switching schemes have intrinsically one more bit resolution than other switching schemes, such that it only requires half of the total capacitance. Second, the monotonic switching scheme moves the voltage by a full V_{REF} on either the top or bottom half of the capacitive DAC; on the other hand, the MCS algorithm moves the voltage by $V_{REF} - V_{CM} = (1/2)V_{CM}$ on both the top and bottom half of the DAC. Since the energy consumption is proportional to CV^2 , the energy consumption in the monotonic case is proportional to CV_{REF}^2 , and the energy consumption in the MCS case is proportional to $2 \times C(V_{REF}/2)^2 = (1/2)CV_{REF}^2$. We can see that the MCS algorithm has even higher energy efficiency compared to the monotonic switching algorithm. Third, the common-mode voltage of the DAC in the MCS algorithm does not change as in the case of the monotonic switching algorithm. This simplifies the design of the comparator and improves the linearity of the ADC. Fourth, it has a similar drawback as in

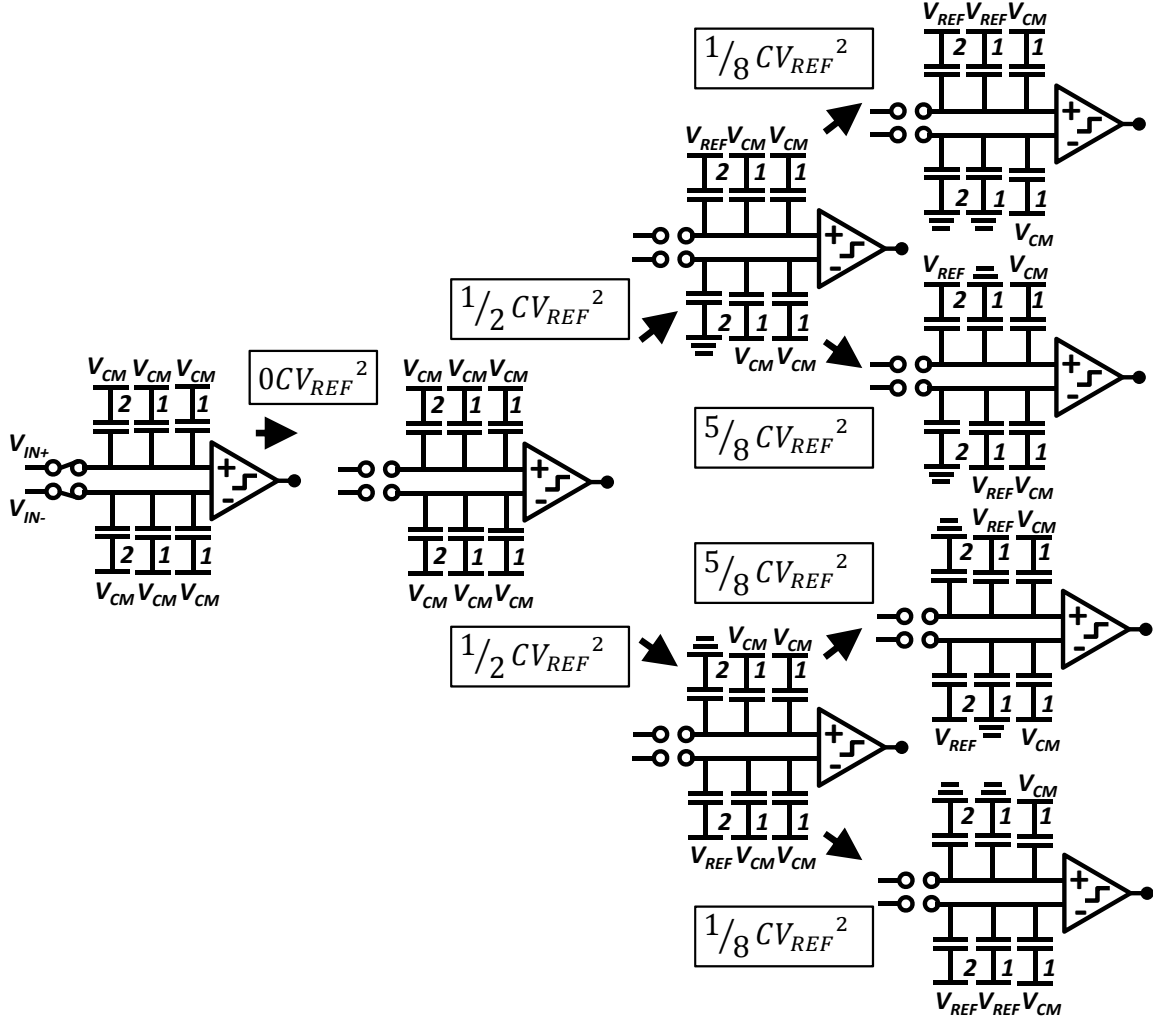


Figure 5-9: Merged capacitor switching algorithm, showing reduced energy consumption compared to Figures 5-2, 5-4, 5-5 and 5-7.

some of the previous algorithms in that the voltage on the top plate does not begin and return to the same voltage during the conversion process. As a result, it is sensitive to the parasitic capacitors on that node. The MCS algorithm consumes 94% less energy compared to the conventional implementation. The average switching energy of an n -bit merged capacitor switching algorithm can be derived as follows:

$$E_{MCS} = \sum_{i=1}^{n-1} 2^{n-3-2i} \times (2^i - 1) CV_{REF}^2 \quad (5.7)$$

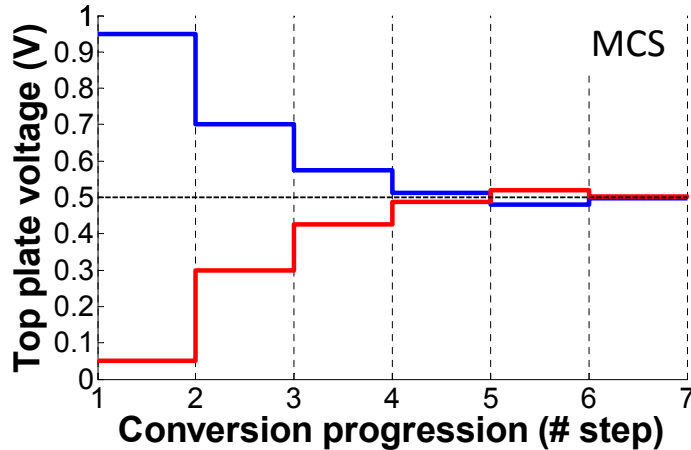


Figure 5-10: The top-plate waveform when using the MCS algorithm. The input is assumed to have magnitude equal to 0.9 with $V_{IN+} = 0.95$, $V_{IN-} = 0.05$ and $V_{REF} = 1.0$. The final output bit sequence is 111100. This switching scheme requires an additional reference voltage V_{CM} compared to previous switching algorithm.

Inverted Merged Capacitor Switching Algorithm

In this section, we propose an inverted merged capacitor switching (IMCS) algorithm as shown in Figure 5-11. The key idea is to invert the sampling by sampling V_{CM} on the top plates and the inputs on the bottom plates of the DAC. Both sources are then disconnected from the DAC and V_{CM} is applied to the bottom plates of the DAC for the first comparison. The rest of the steps remain the same as in the previous MCS algorithm. The top-plate waveform is shown in Figure 5-12. Without costing additional energy, the IMCS ensures that the voltages on the top plates of the DAC begin and end at the same voltage, V_{CM} , thereby eliminating its sensitivity to parasitic capacitances on that node as well as signal dependence of charge injection. The average energy consumption of the IMCS algorithm is the same as the average energy consumption of the MCS algorithm. Figure 5-13 picks one of the four final configurations from Figure 5-11 to show how the IMCS algorithm removes the effect of the parasitic capacitances C_P . The voltages on node V_+ and V_- , assuming they have both settled completely, are calculated for this final configuration. We can see from Equation 5.8 that the effect of the parasitic capacitance is canceled out and does not affect the accuracy of conversion.

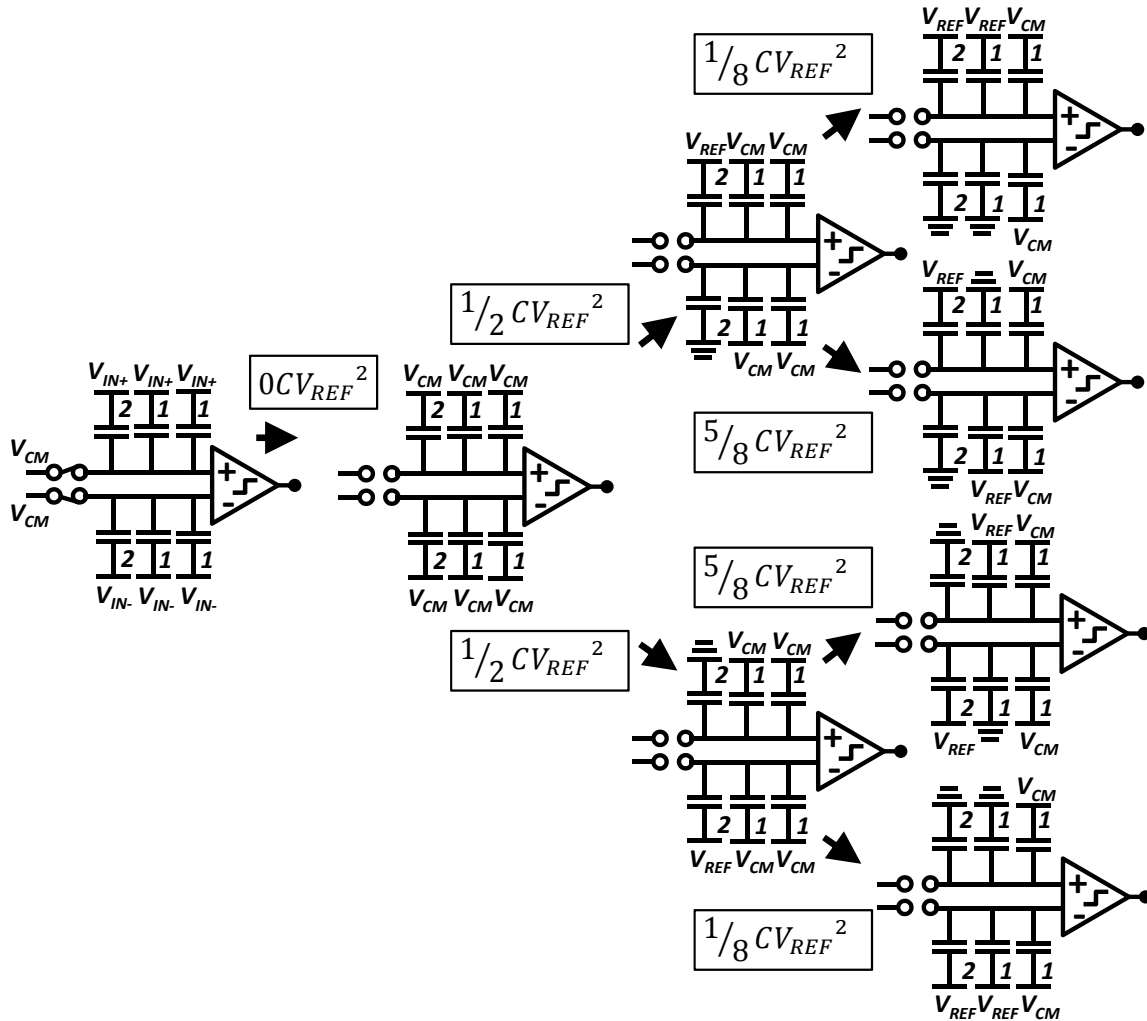


Figure 5-11: Inverted merged capacitor switching (IMCS) algorithm, achieving the same energy efficiency as the MCS algorithm. It inverts the first charging sequences such that the conversion accuracy is not affected by the parasitic capacitance on the top plates of the DAC.

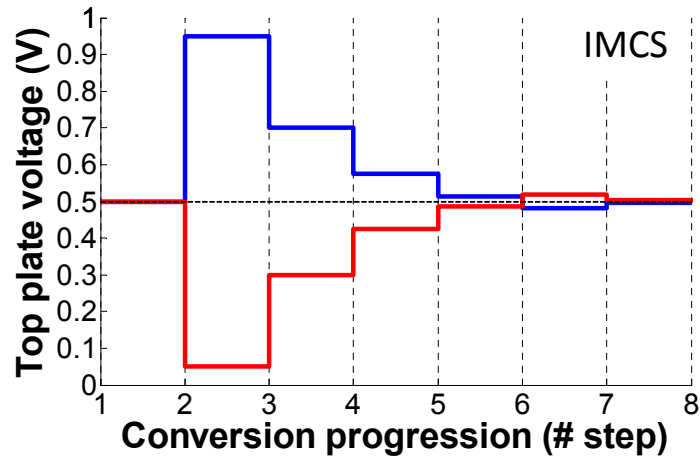


Figure 5-12: The top-plate waveform when using the IMCS algorithm. The input is assumed to have magnitude equal to 0.9 with $V_{IN+} = 0.95$, $V_{IN-} = 0.05$ and $V_{REF} = 1.0$. The final output bit sequence is 111100. This switching algorithm achieves the same energy efficiency as the MCS algorithm, but the accuracy of the IMCS algorithm is not sensitive to parasitic capacitances on the top plates of the DAC.

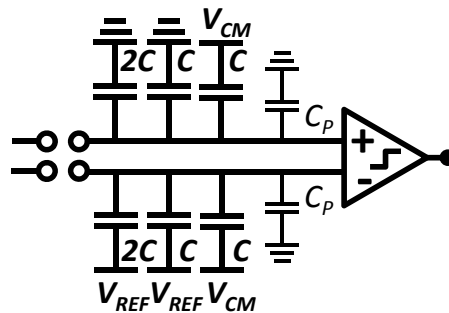


Figure 5-13: Configuration to consider the effect of parasitic capacitance on IMCS algorithm.

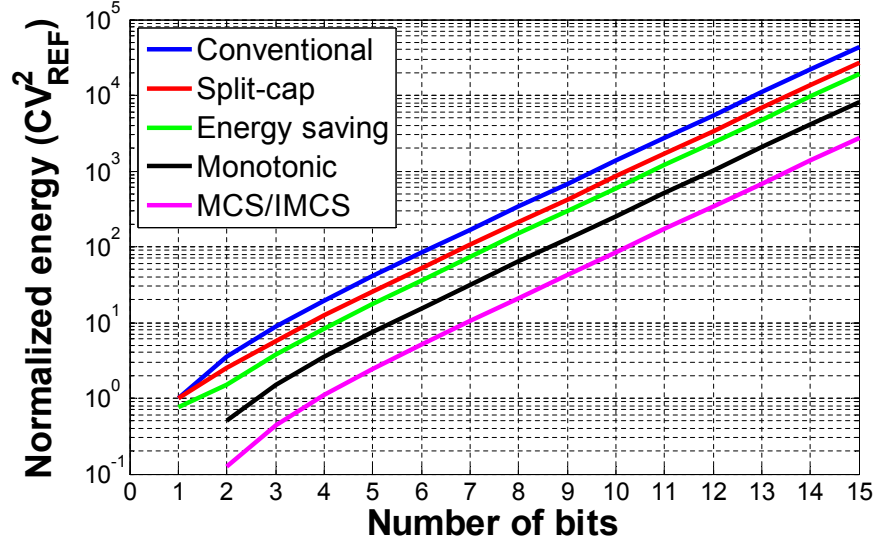


Figure 5-14: Comparing energy consumption of different switching algorithms (conventional, split-cap, energy saving, monotonic and MCS/IMCS.)

$$\begin{aligned}
 V_+ &= \left(1 + \frac{4C}{4C + C_P}\right)V_{CM} - \frac{4C}{4C + C_P}V_{IN+} - \frac{3C}{4C + C_P}V_{CM} \\
 V_- &= \left(1 + \frac{4C}{4C + C_P}\right)V_{CM} - \frac{4C}{4C + C_P}V_{IN-} - \frac{3C}{4C + C_P}V_{CM} + \frac{3C}{4C + C_P}V_{REF} \\
 V_+ - V_- &\geq 0 \quad \longleftrightarrow \quad V_{IN+} - V_{IN-} \geq \frac{3}{4}V_{REF} \quad (5.8)
 \end{aligned}$$

Summary and Comparison of the Switching Algorithms

Figure 5-14 compares the average energy consumption of the five switching schemes versus different number of bits. The MCS and IMCS switching schemes are able to achieve the highest energy efficiency. Figure 5-15 shows the common figures of merit that are used to evaluate switching algorithms. Across the board, in terms of the total number of switches, whether the switching scheme allowing rail-to-rail input swing, total capacitance, common mode remaining fixed during the conversion process, insensitivity to parasitic capacitance, and energy consumption, the IMCS algorithm is able to resolve previous limitations and achieve the best overall figures of merit.

	Number of Switches	Rail-to-Rail Input	Total Capacitors	Common Mode	Sensitive to Parasitic	Energy
Conventional	1 X	yes	1 X	constant	no	100%
Split Capacitor	2 X	yes	1 X	constant	no	62%
Energy Saving	2 X	no	1 X	constant	yes	43%
Monotonic	1 X	yes	(1/2) X	varying	yes	18%
MCS	1 X	yes	(1/2) X	constant	yes	6%
IMCS	1X	yes	(1/2) X	constant	no	6%

Figure 5-15: Comparison of different switching schemes in terms of various figures of merit. The IMCS algorithm is able to achieve the best figure of merit across the board.

5.1.2 Main-Sub-DAC Array

As the resolution increases, the size of the MSB capacitor increases exponentially and at certain resolution, it may become impractical to implement the large capacitor on-chip. For example, if the unit capacitor is 50fF , the MSB capacitor of a 16-bit ADC would need to be $2^{16-2} \cdot 50\text{fF} = 820\text{pF}$, if implemented using the IMCS switching algorithm. If the capacitance has density of $1.0\text{fF}/\mu\text{m}^2$, the area of this one MSB capacitor is roughly 900 by 900 μm^2 . This size is impractical and too costly to implement on chip, especially in sub-micron technologies. Moreover, with such large input capacitance, it becomes increasingly difficult to design an input buffer to drive such load with high bandwidth, good linearity and low noise.

One method of reducing the size of the capacitors is by introducing a bridge capacitor and splitting the DAC into two parts, main and sub DACs. An 8-bit example is shown in Figure 5-16, in which the total weight of the sub-DAC is equal to the weight of the LSB capacitor in the main DAC. The sum of the total capacitance before and after applying the main-sub-dac array architecture is $128C$ and $40C$, respectively. This architecture is able to reduce the total size of the capacitor by more than three times in this example. For the more general case where there are a total of L bits

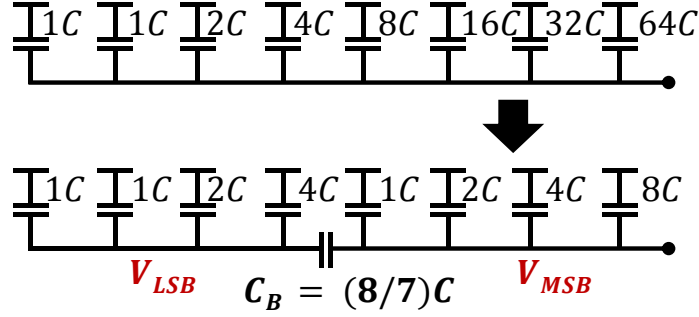


Figure 5-16: An 8-bit example of using the main-sub-dac array architecture. Using the main-sub-dac array architecture the total capacitance can be reduced from $128C$ to $24C$.

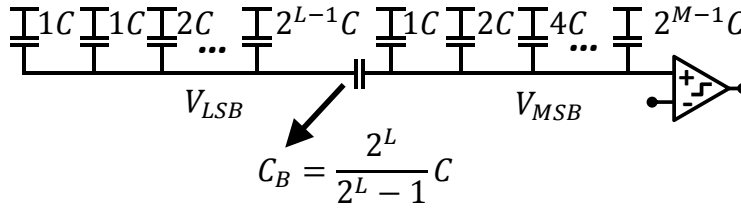


Figure 5-17: A more general representation of the main-sub-dac array architecture. The LSB DAC has a total of L -bit resolution and the MSB DAC has a total of M -bit resolution. The bridge capacitor C_B is a fractional value.

in the sub-DAC, the bridge capacitor C_B can be calculated as follows:

$$C_B // 2^L \cdot C = C$$

$$C_B = \frac{2^L}{2^L - 1} C \quad (5.9)$$

Equation 5.9 suggests that the bridge capacitor has to have a fractional value for the equivalent weight of the LSB DAC to be 1 with respect to the MSB capacitors. This imposes a limitation on the matching properties between the bridge capacitor and the remaining capacitors on the DAC.

One problem associated with the main-sub-dac array architecture is that the voltage on the sub DAC output (V_{LSB} node), in Figure 5-17, can go beyond or below the rails during the conversion process. This can happen because the comparator is connected to the V_{MSB} nodes, and therefore, the digital control block only looks at the voltage on the V_{MSB} nodes and tries to configure the DAC such that this node

converges to the common mode voltage at the end of the conversion process. In other words, the voltage on the V_{LSB} is not relevant to the operations of the conversion process and potentially can go to an undefined value.

The ADC experiences the worst over-range problem for an input voltage that generates all 0's MSB bits and all 1's LSB bits or vice versa. This worst case can be understood as follows. At the beginning of the conversion process, the voltages on the V_{MSB} and V_{LSB} nodes begin at the same value. In this specific case, when all the MSB bits are 0's, the initial voltages on V_{MSB} (and V_{LSB}) are close to V_{REF+} . After all the MSB bits are switched to V_{REF-} , the voltage on V_{MSB} is moved down by an amount that drives the voltage on V_{MSB} below the common mode voltage V_{CM} . We know this because the remaining bits are all 1's.

The voltage on V_{LSB} , though, is not moved down by the same amount as the voltage on V_{MSB} when the MSB bits are switched to V_{REF-} . Since there is an attenuation by the bridge capacitor C_B , when the MSB bits are switched, the voltage on V_{LSB} barely moves. In other words, V_{LSB} is still near its initial value, which is close to V_{REF+} . Now since the voltage on V_{MSB} is below V_{CM} , all the LSB bits have to be switched to 1's to drive the V_{MSB} node back near V_{CM} . The already near-the-rail voltage node, V_{LSB} , will move further up and beyond the rail. The first-order approximation of this effect can be described by Equation 5.10.

$$\begin{aligned}
 V_{LSB,worst} = 2 \cdot V_{CM} - V_{IN} + & \overbrace{\frac{C_B/(C_B + C_{LT})}{C_{MT} + \frac{C_B \times C_{LT}}{C_B + C_{LT}}} \times C_{MT} \times (V_{REF-} - V_{CM})}^{\text{Contribution to voltage on } V_{LSB} \text{ from the MSB caps}} \\
 & + \underbrace{\frac{1}{C_{LT} + \frac{C_B \times C_{MT}}{C_B + C_{MT}}} \times C_{LT} \times (V_{REF+} - V_{CM})}_{\text{Contribution to voltage on } V_{LSB} \text{ from the LSB caps}} \quad (5.10)
 \end{aligned}$$

where C_{LT} is the sum of the total capacitance on the LSB DAC, and C_{MT} is the sum of the total capacitance on the MSB DAC. The worst case configuration for the final voltage on V_{LSB} occurs when all MSB bits are 0's and all LSB bits are 1's. If we make the approximation that $C_{MT} \gg C_B$ and $C_{LT} \gg C_B$, the above equation can

be simplified as follows:

$$\begin{aligned} V_{LSB, worst} &\approx 2 \cdot V_{CM} - V_{IN} + \frac{1}{C_{LT}} \times (V_{REF-} - V_{CM}) + (V_{REF+} - V_{CM}) \\ &\approx V_{CM} - V_{IN} + V_{REF+} \end{aligned} \quad (5.11)$$

In this particular case, since all the MSB bits are 0's and all the LSB bits are 1's, the input voltage is close to V_{REF-} .²

If $V_{IN} = V_{REF-}$ and we want $V_{LSB, worst}$ to stay within the rail and less than V_{DD} , then the following inequality must be satisfied, assuming $V_{CM} = V_{DD}/2$:

$$(V_{REF+} - V_{REF-}) \leq V_{DD}/2 \quad (5.12)$$

From Inequality 5.12, we see that in order for V_{LSB} to stay within the rails, the effective input signal range has to be reduced by two times or equivalently, the signal power has to be reduced by four times. To maintain the same SNR and the same speed, four times the total capacitance and power are needed.

To resolve the matching issues, Agnes *et al.* in [45] replace the fractional bridge capacitor by a unit capacitor and remove one of the size 1 capacitors from the LSB array. Even though the total weight of the LSB DAC is the same as the lowest bit in the MSB array, this approach introduces a 1 LSB gain error. Chen *et al.* in [44] picks a value for the bridge capacitor that is slightly larger than the calculated size in Equation 5.9. A tunable capacitor is added on the LSB side to adjust the total weight of the sub DAC and calibrate out the mismatches resulting from the fractional capacitor value. To prevent the over-range problem, the approaches in [44, 46] reduce the input signal swing.

Figure 5-18 shows a new split-capacitor array architecture that is able to solve both problems at once. An intentional grounding capacitor, C_X , is added on the LSB side of the array. Using the same principle, we can calculate the new bridge capacitor

²This is especially true for high-resolution ADCs.

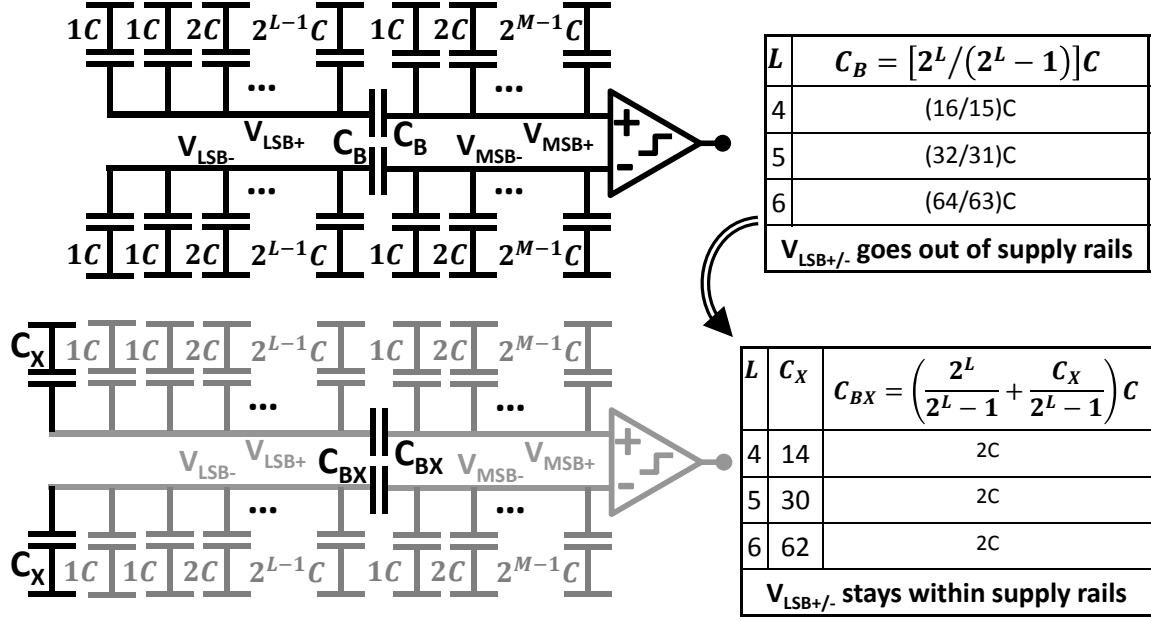


Figure 5-18: New main-sub-dac array architecture. This new architecture resolves the matching and over-range problem together.

value as follows:

$$C_{BX} // (2^L C + C_X) = C$$

$$C_{BX} = \left(\frac{2^L}{2^L - 1} + \frac{C_X}{2^L - 1} \right) C \quad (5.13)$$

If the value of the intentional grounding capacitor is properly picked, the value of the new bridge capacitor (C_{BX}) can be an integer, as shown in the example of Figure 5-18 with $L = 4, 5$ and 6 . The size of the capacitor C_X is roughly equal to the total size of the capacitor in the LSB array. In other words, when the LSB bits are switching, the voltage jumps are now attenuated by a factor equal to $\frac{C_X}{C_{LT} + C_X} \approx \frac{1}{2}$. Thus, the grounding capacitor also helps prevent the V_{LSB} node from going beyond or below the rails. This approach not only improves the linearity because of better matching, but it also allows rail-to-rail signal range, which can significantly improve signal-to-noise ratio (SNR). This is especially important in advanced CMOS technologies in which the supply rail is limited. In a real design, before adding the grounding capacitor C_X , there will be existing parasitic capacitance at the output of the sub-DAC already. An

intentional grounding capacitor is added such that the total capacitance on the node, including the parasitic capacitance, is equal to the desired capacitor size calculated according to Equation 5.13.

5.1.3 Redundancy Implementation

Figure 3-2 shows the map of the desired searching scheme for the redundant SAR ADC. We have discussed the benefits of redundancy and its corresponding calibration algorithm in the previous chapters and we assumed that this search pattern can be realized in real implementation. In this section, we will discuss the realization of this redundant search pattern in our SAR design with minimal added digital complexity and power consumption.

Kuttner and Hesener *et al.* in [48, 49] introduce redundancy by making a DAC array with only unit capacitors. All of the capacitors in the array are individually controllable using two thermometer decoders. The redundancy is not built into the DAC, but rather is calculated in the digital part of the converter with an arithmetical unit. Compared to the original SAR architecture, it requires one decoder unit for each individual capacitor, row and column thermometer decoders, an arithmetical unit and complex digital control. Even though this implementation provides the flexibility to program the amount of redundancy even after fabrication, the added complexity and power consumption can be the main bottleneck of this approach.

Another technique, by Liu *et al.* in [35], bypasses such complexity and implements the redundancy algorithm directly by sizing the capacitors with a sub-binary ratio. This technique allows the design to incorporate redundancy directly without the previous complexity, but the search steps become asymmetric, thus the tolerance to errors becomes asymmetric. As an example, in Figure 5-19, we show the first two bit decision cycles of a redundant SAR ADC if the redundancy is implemented directly with the conventional switching algorithm. During the first bit cycle, the sign bit is determined. Depending on the value of this bit comparison, the switching algorithm either takes an “up” or a “down” transition. According to the decision level progressions in Figure 3-2, the search steps should move up or down by the same

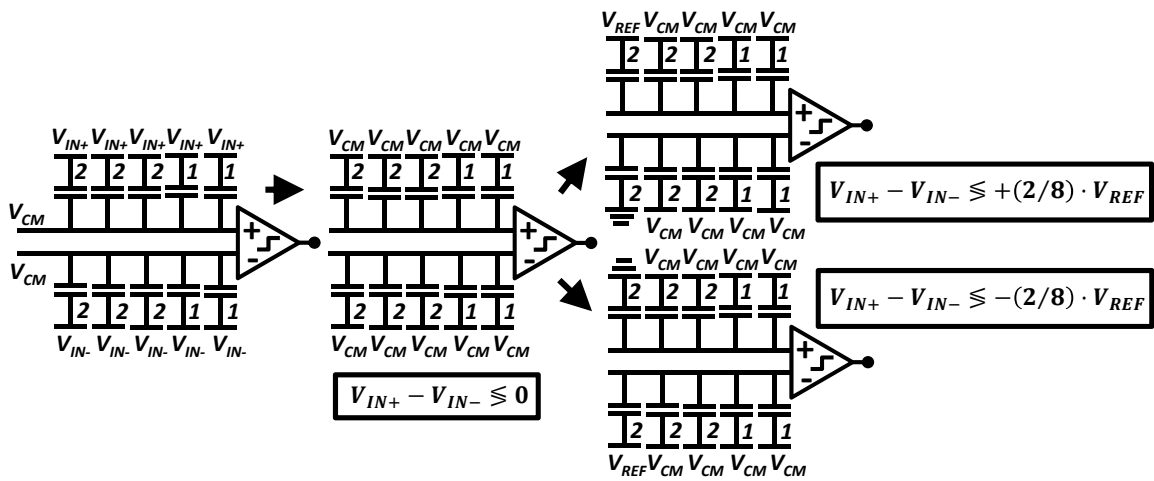


Figure 5-20: Redundancy implementation using IMCS algorithm. It can be done by directly sizing the capacitors proportional to the desired searching step sizes, while still maintaining a symmetric search window size.

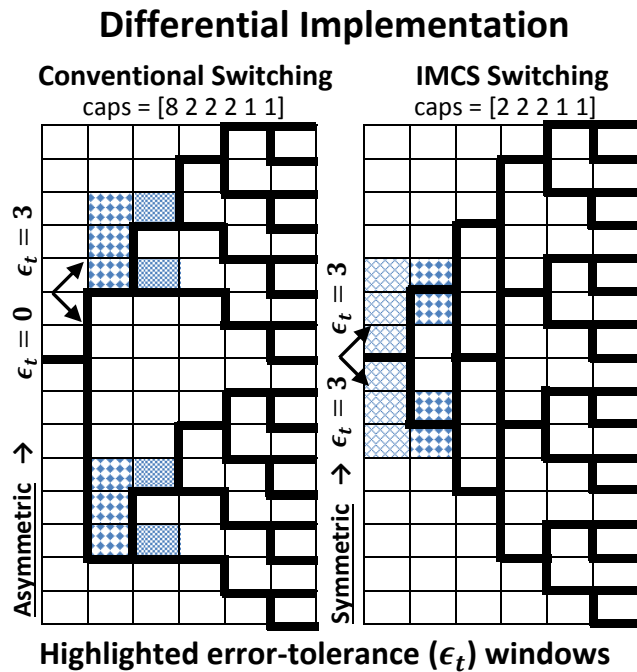


Figure 5-21: Comparison of error tolerance windows (ϵ_t) between two redundancy implementations. Implementing redundancy using the IMCS algorithm allows symmetric search window size and symmetric tolerance to dynamic settling errors.

In preview, the new prototype is able to implement redundancy directly into the DAC by just sizing the capacitors proportional to the desired stepping sizes during the search algorithm. At the same time, it is able to maintain the symmetry during the search process, which allows equal tolerance to settling errors regardless of whether it is a “up” or a “down” transition. Even though this implementation does not have the flexibility to program the amount of redundancy after fabrication as the case in [48], it avoids the extra power, complex digital circuitry and delay introduced by the added arithmetical unit and decoders. Moreover, the IMCS switching algorithm is 94% more energy efficient compared to the conventional switching algorithm used in [35]. A better figure of merit is expected when combining all of the benefits of this redundancy implementation.

5.1.4 The Overall Architecture

The new architecture is able to combine all the new techniques discussed previously in this section. First, we are able to incorporate the inverted merged capacitor switching (IMCS) algorithm to achieve the highest switching efficiency and eliminate the linearity limiting factor of the previous MCS algorithm. Second, by introducing an intentional grounding capacitor on the sub capacitive DAC, both the matching and the over-range problems are eliminated. This not only helps improve linearity, but also allows full rail-to-rail input range that can significantly increase the achievable SNR. Third, redundancy is incorporated directly into the capacitive DAC using the IMCS algorithm without introducing significant power consumption or digital complexity. Combining all these new techniques, we will use the statistically-based digital background calibration algorithm developed in Chapter 4 to further improve the linearity.

Figure 5-22 shows the overall architecture and the key building blocks of the prototype. This ADC has effective resolution of 12 bits with four redundant decisions, making a total of 16 decisions. An intentional grounding capacitor of size 15 is added to the LSB DAC such that an integer value of bridge capacitor can be used and the voltage V_{LSB} does not go beyond or below the rails. Redundancy is built directly into

the DAC as the ratio between the capacitors is less than binary. The ADC is built with a radix that is roughly equal to 1.6 and the resulting sub-binary search step sizes are equal to [875, 420, 280, 175, 105, 70, 52.5, 35, 17.5, 8, 4, 3, 1, 1, 0.5, 0.5]. The sum of these steps is equal to 2048, which is the required sum for a 12-bit ADC using the IMCS algorithm. The building blocks include the DAC, the sampling circuit, the ready signal generator, the pulse generator, output registers, digital control logic and bootstrapped switches.

In a conventional implementation of the SAR algorithm, the sampling phase and each conversion period is driven by an external clock. For a 10-bit SAR ADC with sampling rate of F_S , an external clock that runs at $(10+1) \times F_S$ is needed. To generate and distribute the clock at such high speed would likely consume significant power. From the perspective of speed, in a synchronous design, every clock cycle must provide the worst-case comparison time, which includes the maximum DAC settling time and maximum comparator delay to resolve the minimum resolvable input level [51]. It is also more difficult to generate a low-jitter clock at such a high speed. The prototype, therefore, employs asynchronous operation.

Figure 5-23 shows the important waveforms during the conversion process when using the new IMCS switching algorithm. The conversion begins by sampling the input signal and the common mode voltage on the bottom and top plates of the capacitive DAC, respectively. During the sampling phase, both Φ_{CM} and Φ_{VIN} are high. After enough time for the input signal to settle onto the DAC, Φ_{VIN} goes low first before Φ_{CM} goes low to allow “bottom plate sampling.” This avoids the signal-dependent charge injection effect to first order³. After the sampling phase, both Φ_{CM} and Φ_{VIN} go low and Φ_{CM2} goes high to charge all the bottom plates of the DAC to V_{CM} . This specific step is what differentiates the IMCS algorithm from the regular MCS algorithm.

At the end of this phase, Φ_{CM2} goes low. The falling edge of Φ_{CM2} triggers the latch clock to go high and the comparator compares the voltages at the outputs of

³This signal dependent effect will be analyzed when the sampling circuit is discussed in Section 5.2.2.

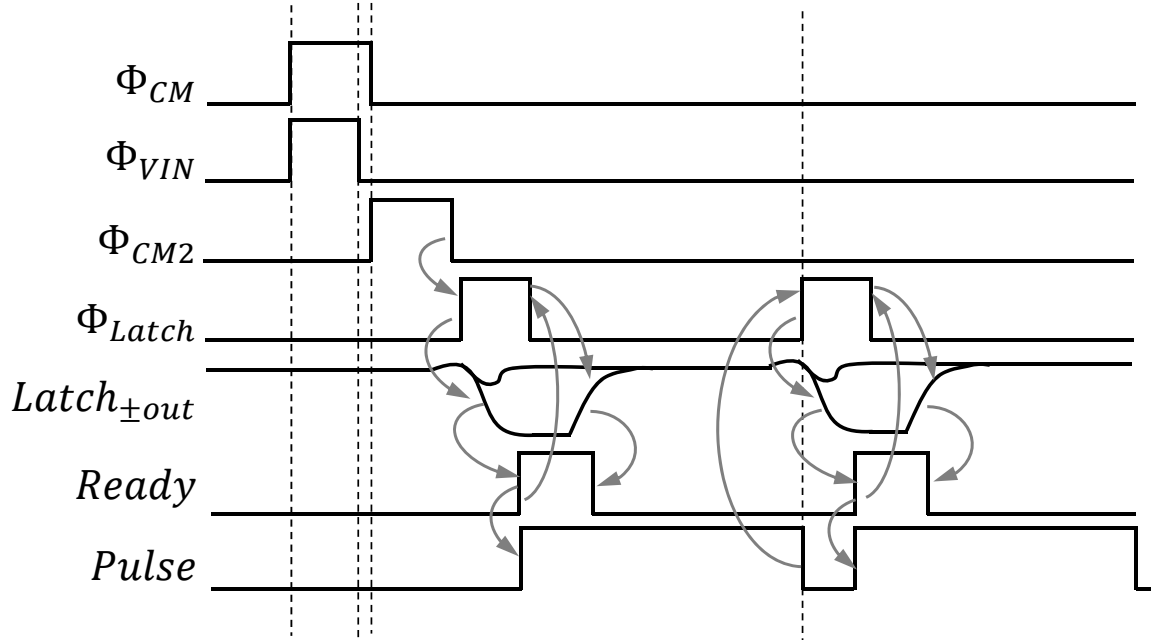


Figure 5-23: Timing waveform of the asynchronous SAR ADC using the inverted merged capacitor switching (IMCS) algorithm.

the differential DAC. Depending on the magnitude of the input differential signal, the latch outputs will begin to diverge at different rates. The “ready generator” block detects the time when the two outputs of the latch have a large enough difference and generates a “ready” signal. This “ready” signal does three things. First, it latches the output onto a control register and subsequently, this control register will re-configure the DAC to generate the next digital estimate of the analog input signal. Second, it will reset the comparator and both outputs of the comparator will go back high. Third, it generates a pulse with controllable pulse width. This pulse width represents an estimate of the total time needed for the DAC outputs to settle. The falling edge of this pulse triggers the latch clock to go high and the same procedure repeats until all bit comparisons are done.

5.2 Key Circuit Building Blocks

In this section, we will discuss in detail the key building blocks of the SAR ADC. Performance matrices are analyzed and related to transistor parameters. Simulations

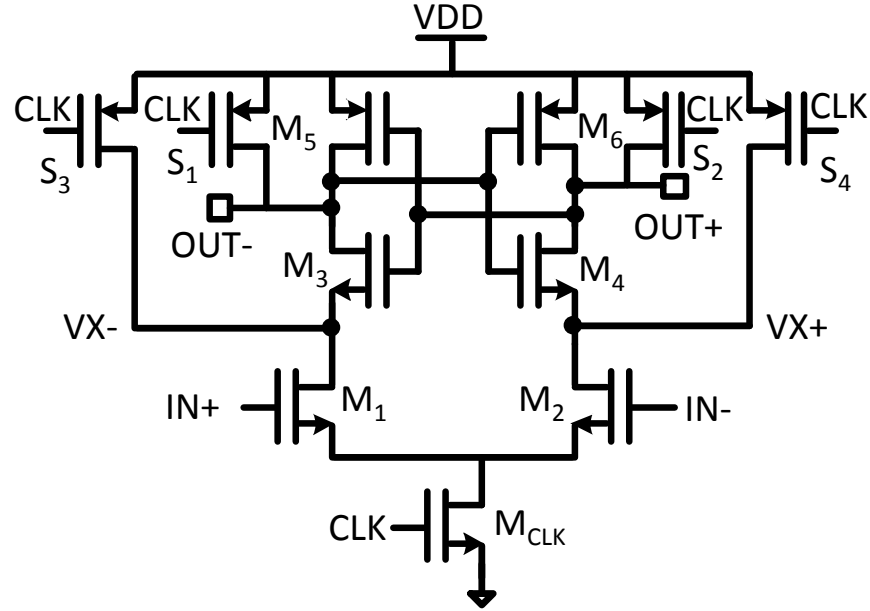


Figure 5-24: The design of StrongARM latch comparator. It consumes no static power during standby period and only dynamic current is present during regeneration.

are done to verify the design.

5.2.1 Latch Comparator

One of the key building blocks of the SAR architecture is the comparator. To ensure ultra-low-power operation, a StrongARM latch comparator has proven to be very suitable for these applications as it consumes no static current during the standby period and only dynamic current is present during regeneration [53]. Moreover, a StrongARM latch comparator maximizes the speed of operation by using positive feedback to regenerate its outputs to logic levels. Because of its high speed and power efficiency, a StrongARM latch comparator is chosen for the ADC prototype.

Figure 5-24 shows the design of the StrongARM latch comparator. The design uses no pre-amplifications, and therefore, it avoids using any static bias current. Besides the operational speed and power consumption, another key factor in designing a latch comparator is its thermal noise. The dynamic latch noise becomes especially important in the absence of linear analog components. If not designed properly, the comparator thermal noise can degrade the ENOB [55]. Since the biasing conditions

are continuously changing during the regeneration, traditional small signal analysis that linearizes the parameters around one biasing condition does not produce an accurate estimate of noise. The noise analysis must be done in the large signal domain, in which the circuit is analyzed under varying biasing conditions.

The operation can be divided into three different phases based on the transitions of transistors from one operating region to another. Assumptions are made such that the transitions between phases are instantaneous, thus the circuit can be analyzed separately in each region. Noise is analyzed in the time domain using stochastic differential equations (SDE), following the approach in [54]. We will use the convention $X_{i,j}$ to denote the parameter X_i in phase j .

Figure 5-25 shows the transient simulation of our latch comparator. During the reset phase, the voltages on nodes $VX+$, $VX-$, $OUT+$ and $OUT-$ are all reset to V_{DD} . When the clock goes high, transistor M_{CLK} begins conducting, turning transistor M_{1-2} on to discharge nodes $VX+$ and $VX-$. When nodes $VX+$ and $VX-$ are sufficiently discharged, nodes $OUT+$ and $OUT-$ begin to go down at slightly different rates due to the differential input. The rates depend on the magnitude of the input differential signal. When the outputs reach roughly $V_{CM} (= V_{DD}/2)$, the two outputs diverge and one moves to V_{DD} while the other continues to move towards V_{SS} .

The operation can be divided into three different phases. Phase 1 is defined as the time when only transistors M_{CLK} and M_{1-2} are on. During this phase, transistor M_{CLK} is in the linear region while transistors M_{1-2} are in the saturation region. When the voltage on VX reach $V_{DD} - V_{T3-4}$, thus turning on transistors M_{3-4} , the transient reaches Phase 2. This is defined as the time interval when transistors M_{1-4} are all in the saturation region. When node V_X finally discharges below $V_{CM} - V_{T1}$, transistors M_{1-2} move out of the saturation region and enter the linear region. This concludes Phase 2 and the transient enters Phase 3. During Phase 3, only the cross-coupled inverters are active and the input differential voltage has negligible effect on the output. With the three phases defined, in each phase, the noise can be separately characterized using the linearized small signal parameters in that region. The detailed

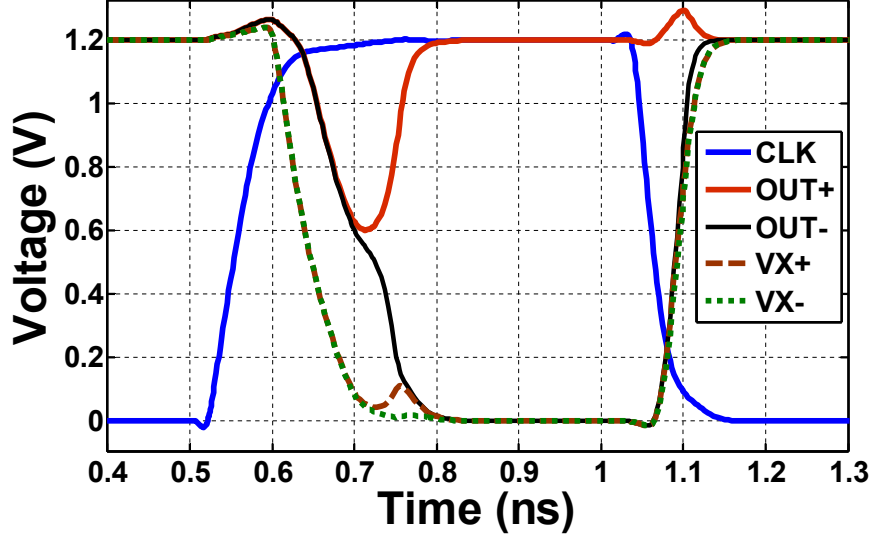


Figure 5-25: Large signal transient response of the latch comparator. After clock signal goes high, the differential outputs begin to discharge together before one output starts moving to V_{DD} and the other output continues to discharge towards ground.

derivation can be found in [54] and the results are presented here:

$$\begin{aligned}
\sigma_{M_1}^2 &= \frac{2kT\gamma}{C_X\mathcal{F}} \\
\sigma_{S_1}^2 &= \frac{kT}{2C_O\mathcal{F}^2} + \frac{kT}{2C_X\mathcal{F}^2\mathcal{H}} + \frac{kTC_O}{8C_X^2\mathcal{F}^2\mathcal{H}^2} \\
\sigma_{M_{3-5}}^2 &= \frac{kT\gamma}{2C_X\mathcal{F}^2\mathcal{H}} + \frac{kT\gamma C_O}{8C_X^2\mathcal{F}^2\mathcal{H}^2} \\
\sigma_{S_3}^2 &= \frac{kT}{2C_X\mathcal{F}^2}
\end{aligned} \tag{5.14}$$

$$\begin{aligned}
\mathcal{F} &= \frac{V_{T3}}{V_{ov1,1}} \\
\mathcal{H} &= \frac{V_{DD} - V_{CM}}{V_{ov3,2}} \frac{I_{D3,2}}{I_{D1,2} - I_{D3,2}}
\end{aligned} \tag{5.15}$$

where C_X is the total capacitance on the V_X node, C_O is the total capacitance on the OUT node, I_D is the average current, V_{ov} is the over-drive voltage, V_T is the threshold voltage and γ is the noise factor (typically equal to 2/3 in CMOS).

From the noise expressions in Equation 5.14, we see that they have the typical kT/C form, with the addition of a few noise factors. The equations suggest that

besides the typical way of adding more capacitance to lower the total integrated noise, increasing parameters \mathcal{F} and \mathcal{H} also have the effect of reducing noise. Comparing the two strategies, the latter strategy focuses on decreasing the thermal noise using the most influential factors. As a result, this strategy may be more energy efficient compared to adding more capacitance directly.

Increasing the two parameters \mathcal{F} and \mathcal{H} helps decrease the integrated thermal noise, based on the following intuition. \mathcal{F} is the threshold voltage of M_3 by the over-drive voltage of M_1 during Phase 1. Larger V_{T3} and smaller $V_{ov1,1}$ both imply that the transient will spend more time in phase 1. As discussed previously, the transient only enters phase 2 when the V_X voltage is discharged below $V_{DD} - V_{T3}$. Larger V_{T3} means that the amount of charge that needs to be discharged is more compared to smaller V_{T3} . On the other hand, $V_{ov1,1}$ is proportional to the rate of discharging during Phase 1. Smaller $V_{ov1,1}$ implies that the discharging rate is lower than larger $V_{ov1,1}$. As a result, more charge to be discharged and lower discharging rate both lead to more time spent in Phase 1.

During Phase 1, all transistors are in saturation regions. Similar to the linear amplification case, the input differential voltage is multiplied by the transconductance of the input pairs and the resulting differential current is integrated onto the output nodes. More time in Phase 1 means that the input has longer time to integrate onto V_X , which consequently implies that the relative signal-to-noise ratio is higher compared to the case when the latch comparator spends less time in Phase 1. Similar logic can be applied to parameter \mathcal{H} to understand why larger \mathcal{H} can also help lower the thermal noise.

To increase the parameter \mathcal{F} , a larger input pair with low over-drive voltage should be used. This can be done by increasing the W/L of the input pairs, decreasing the discharging current in phase 1 by reducing the W/L of M_{CLK} , and decreasing the input common mode voltage V_{CM} .

Figure 5-26 shows a plot of input referred noise, power consumption versus the (W/L) ratio, ρ , between transistor M_{1-2} and transistor M_{CLK} . The comparator is designed in 65nm CMOS technology with supply voltage (V_{DD}) of 1.2V and input

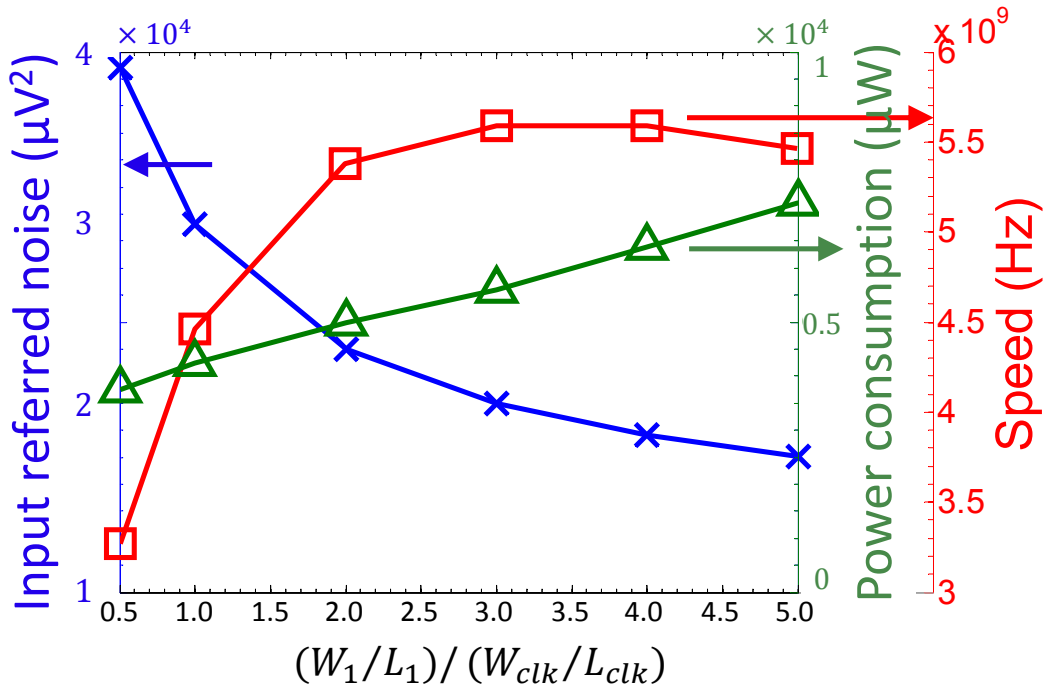


Figure 5-26: Input referred noise, power consumption and speed as a function of $\rho = \frac{W_1/L_1}{W_{clk}/L_{clk}}$.

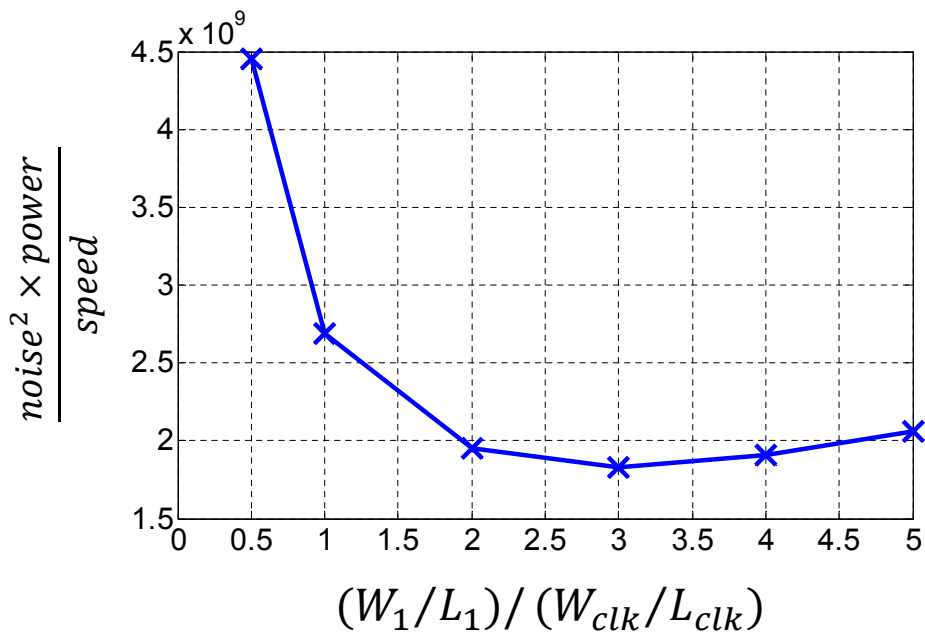


Figure 5-27: Product of noise power, power consumption and delay as a function of $\rho = \frac{W_1/L_1}{W_{clk}/L_{clk}}$. It shows that it is possible to optimize such product by properly ratioing the size of input pairs and the transistor M_{clk} .

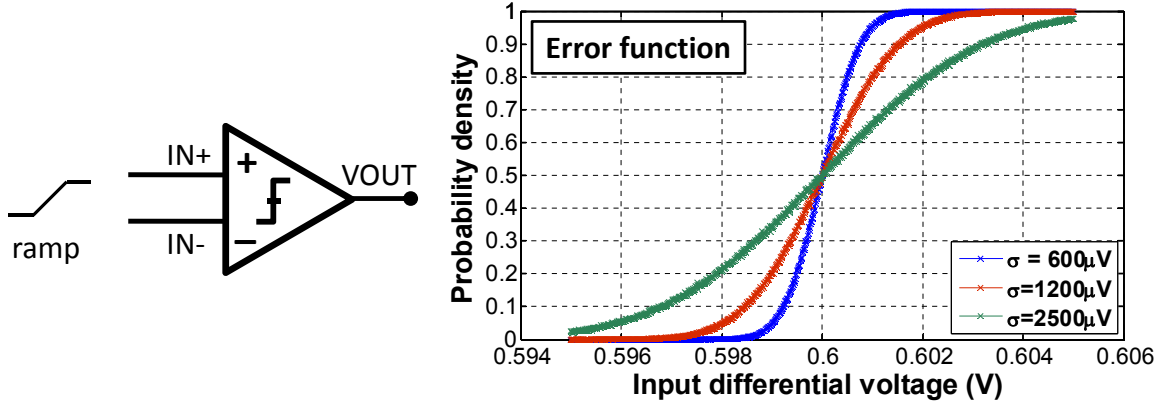


Figure 5-28: Simulation setup to extract the noise variance. The simulation is done in Cadence *SpectreRF* using transient noise analysis.

common mode voltage (V_{CM}) of $600mV$. The power consumption is simulated under the assumption that the comparator is clocked at 1 GHz. The thermal noise is estimated using the transient-noise simulation in *SpectreRF*. Figure 5-28 shows the simulation setup. The input signal is a ramp with each input value denoted as x_i , and its corresponding output value denoted as y_i . The input to the latch comparator is assumed to have input common-mode voltage equal to $600mV$. In an ideal comparator with no thermal noise, for all input x_i below $600mV$, its output y_i is a 0, and for all input greater than $600mV$, its output is a 1. For a comparator with Gaussian noise, the output y_i is the error function, which is a cumulative probability density function of a normal distribution. From this plot, the noise variance can be extracted.

From Figure 5-26, we observe that increasing ρ can help reduce the total input referred noise, but the noise reduction has diminishing return, as evidenced by the decreasing slope while ρ increases. In other words, initially, increasing ρ has a large effect on noise reduction, but gradually, it becomes less effective. It can also be observed that using this method to reduce thermal noise is more energy efficient compared to just adding more capacitance. For example, in the traditional approach, increasing the capacitance by four times, thus consuming four times more power, can help reduce the noise RMS voltage by two times at constant speed. In this case, however, when ρ increase from 0.5 to 1, the noise RMS voltage is reduced by 16% while the power consumption only increases by 11.5%. Since the noise power and the

comparator power shows an opposite and non-linear trend, we can optimize the design from the product of the two, dividing by operating speed as shown in 5-27. From the figure, it can be shown that when $\rho = 3$, the comparator is able to achieve the optimal figure of merit in terms of noise power, speed and total power consumption.

5.2.2 Sampling Circuit

The main criteria for designing a sampling circuit are input bandwidth, distortion, input voltage swing and sampling noise. Our prototype, it does not have a dedicated front-end sample-and-hold circuit; rather, the input is sampled directly onto the capacitive DAC of the SAR ADC. The switches are bootstrapped to reduce the on-resistance variation of the switches. This helps maintain a more constant resistance even when the value of the input signal changes to improve the achievable dynamic linearity. Bottom-plate sampling is used to reduce signal dependent charge injection and consequently, lower the potential distortion introduced at the output. Figure 5-29 shows the demonstration of bottom-plate sampling. The switch connected to the top plate is open first before opening the bottom-plate switch. Since the voltage connected to the top plate switch is always V_{CM} , independent of the input signal, the charge injection is supposed to be constant. However, in real implementation, even with bottom plate sampling, the top plate charge injection is still not constant. This is because even though the bottom-plate switches has constant V_{GS} , the V_{SB} voltage still varies with the input voltage.

The on-resistance of the switch is given by Equation 5.16.

$$R_{on} = \frac{1}{\mu_n C_{ox} \frac{W}{L} (V_{GS} - V_T)} \quad (5.16)$$

where V_T is given by Equation 5.17

$$V_T = V_{T0} + \gamma \left(\sqrt{|2\Phi_F + V_{SB}|} - \sqrt{|2\Phi_F|} \right) \quad (5.17)$$

V_{T0} is the threshold voltage when $V_{SB} = 0$, γ is the body effect coefficient, Φ_F is

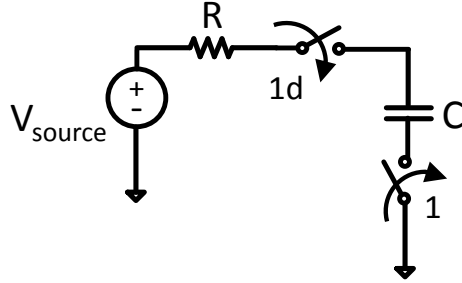


Figure 5-29: Bottom plate sampling circuit to help improve linearity. “1” represents at time 1 and “1d” represents a delay after time 1.

the surface potential and V_{SB} is the source to body voltage. As shown by the two equations, the on-resistance is a function of the threshold voltage and the threshold voltage is a function of the source-to-body voltage V_{SB} . When the input voltage varies, it modulates the on-resistance of the switches even with the bootstrapped configuration. Therefore, given different input voltages, the on-resistance of the bottom plate switch is different and this causes the injected charge from the bottom plate switch to vary with the input voltages at the sampling instant. Figure 5-30 shows the simulation result of this effect. The input range is swept from 0V to 1.2V and the resulting charge injection varies almost linearly from $-8.32mV$ to $-8.29mV$. The difference between the maximum and the minimum charge injection is roughly $30\mu V$. Since this effect displays a linear behavior, it simply scales the input signal by a factor $\frac{1.2V}{1.2V+30\mu V}$ and does not affect SNDR. Even if the entire effect is nonlinear, $30\mu V$ is approximately 1/10 of the LSB_{12} at the 12 bit level.

We can re-examine the R_{on} expression given in Equation 5.16. To design a resistor with low impedance to increase the sampling bandwidth, μ_n , V_T and C_{ox} are process dependent parameters and cannot easily be changed. The channel length L is typically chosen to be the minimum value to minimize the on-resistance. The only remaining parameters are W and V_{GS} . Increasing W can help reduce the on-resistance of the switch, but it will also increase parasitic capacitance. This can increase charge injection. Another drawback of using large W is that it can increase the amount of charge injection when the switch turns off. Another common way to lower the on-resistance of a switch is by using bootstrapped switch [52]. This technique has

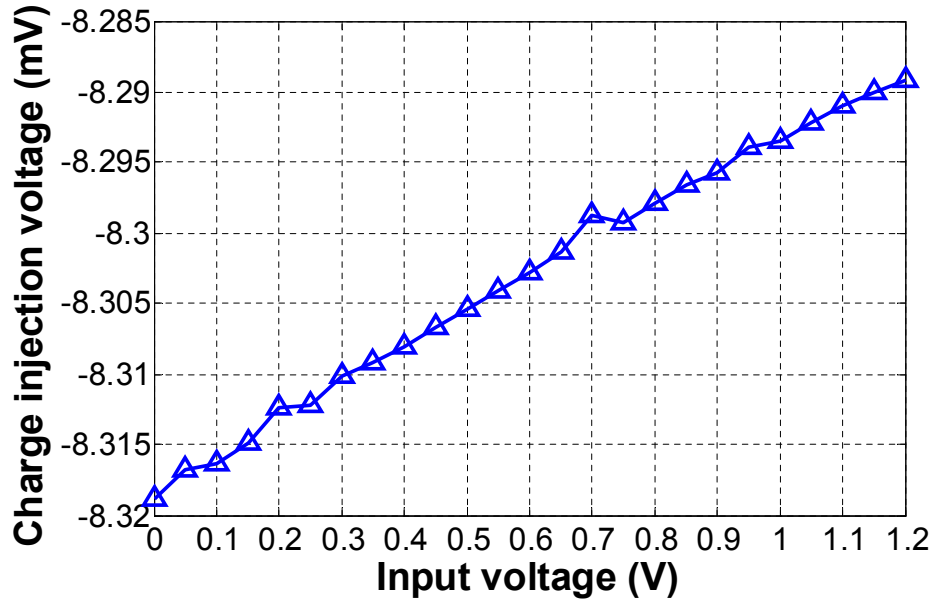


Figure 5-30: Difference in charge injection versus different input value.

become very popular in scaled CMOS processes to achieve low resistance. It keeps a constant voltage between the gate and source terminals of the switch. The circuit implementation is shown in Figure 5-31.

The bootstrapped switch is designed to keep a constant gate-to-source voltage of the switch while achieving low impedance. This is done in such a way that the gate oxide would not exceed V_{DD} to avoid stressing the device and ensure device reliability. The actual switch is not drawn in Figure 5-31. The gate of the switch is connected to “VGATE” and the source of switch is connected to “VIN.” In the “off” state, the gate of the switch is connected to ground and the device is in cutoff. In the “on” state, a constant voltage of roughly V_{DD} is always across V_{GS} , regardless of what the input voltage is. Although the absolute voltage can go beyond V_{DD} , no terminal-to-terminal voltage exceeds V_{DD} during operations.

Figure 5-32 shows the simulation result of the bootstrapped switches. The clock runs at 100 MS/s and the input is a 10 MHz sinusoid wave. When the clock is low, transistors M9 and M10 discharge “VGATE” to ground and transistors M4 and M3 charge the voltage across the capacitor C3 to V_{DD} . This will serve as a battery across the gate and source terminal during the “on” state. Transistors M7 and M11 isolate

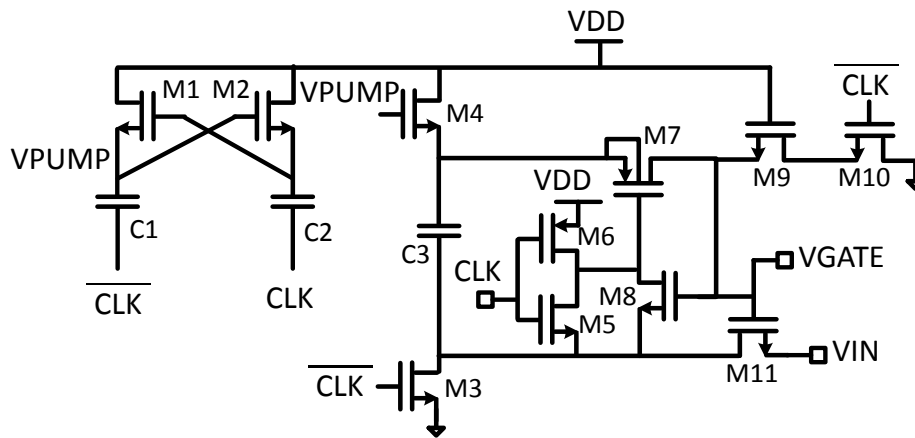


Figure 5-31: Bootstrapped sampling switches. This circuit allows the gate voltage to track the source voltage to maintain a constant V_{GS} , regardless of what the input voltage is.

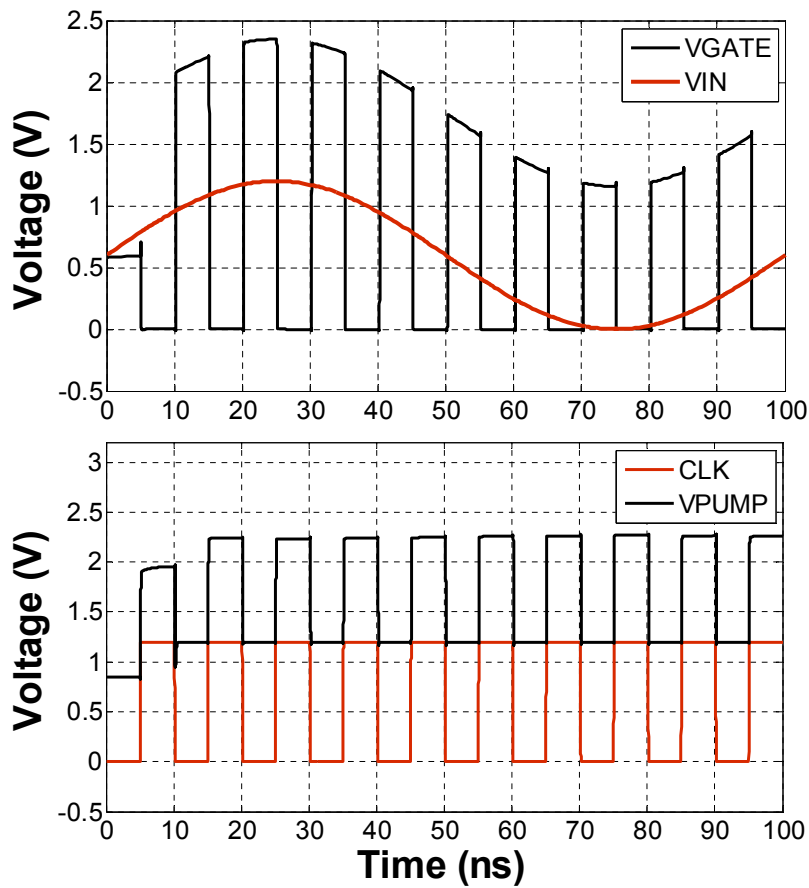


Figure 5-32: Simulation of the bootstrapped sampling circuit. The gate voltage is able to track the input voltage.

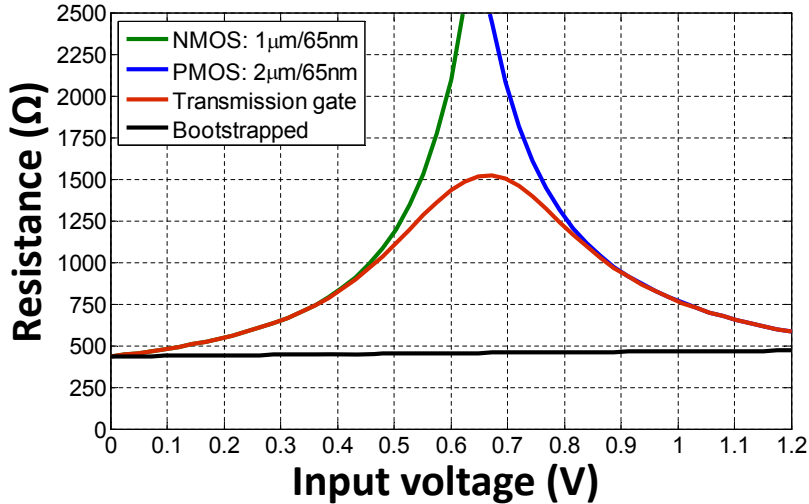


Figure 5-33: Comparison between the switch resistances. Even though the bootstrapped switch is not perfectly constant, its resistance is much flatter compared to switches made out of NMOS, PMOS or transmission gates.

the capacitor C3 from the input and the switch during charging. When the clock goes high, transistors M4 and M3 turn off and transistors M7 and M11 connect the capacitor C3 across the gate and source terminal of the switch. This allow the gate to track the input voltage with constant V_{GS} of V_{DD} . Transistors M1 and M2 together with capacitors C1 and C2 form a clock multiplier that enable M4 to unidirectionally charge C3 during the “off” state. The lower plot in Figure 5-32 shows successfully bootstrapping the VPUMP voltage to roughly $2 \times V_{DD}$.

A plot of on-resistance for four different types of switches versus input voltage is shown in Figure 5-33. The switch is on and a voltage is connected to both the drain and the source terminals in this simulation. The types include an NMOS transistor with $W = 1\mu\text{m}$ and $L = 65\text{nm}$, a PMOS transistor with $W = 1\mu\text{m}$ and $L = 65\text{nm}$, a transmission gate with the previous two switches in parallel and a bootstrapped switch. As shown in the plot, the NMOS bootstrapped switch has almost constant resistance over the input range compared to the other three cases. However, the resistance still experiences a slight increase when increasing V_{IN} from 0 to 1.2V. First, this is due to the increased back-gate effect. Since the body terminal of the switch is tied to ground, not to the source terminal, as the source-to-body potential increases, the threshold voltage of the device also increases, which results in higher on-resistance. If

n	X
8	≥ 6.24
10	≥ 7.62
12	≥ 9.01
14	≥ 10.40
16	≥ 11.78

Table 5.1: The minimum number of time constants needed for a first-order RC circuit to settle within half a LSB for an n -bit ADC.

a triple-well process is available, the switch should have its source and body terminals tied together to minimize such effect. Second, the varying on-resistance is also due to charge sharing between capacitor C3 and parasitic capacitances in the signal path between C3 and the switch. These parasitic capacitances are non-linear capacitances, which change their values when different input voltages are applied through the bootstrapped switch. Therefore, it also introduces additional input-dependent non-linear effect.

The bandwidth of the sampling circuit is designed to be sufficient to track and sample the input signal onto the DAC. For a simple first-order RC sampling circuit, a sinusoid input signal (given in Equation 5.18) will generate an output, given in Equation 5.19.

$$V_{in}(t) = A \cdot \cos(\omega t + \phi) \quad (5.18)$$

$$V_{out}(t) = -\frac{A \cdot \cos(\phi - \theta)}{\sqrt{1 + \omega^2 \tau^2}} e^{-\frac{t}{\tau}} + \frac{A \cdot \cos(\omega t + \phi - \theta)}{\sqrt{1 + \omega^2 \tau^2}} \quad (5.19)$$

where $\tau = R \times C$ and $\theta = \arctan(\omega \tau)$. The first term in Equation 5.19 represents the error due to the exponential settling of the initial transient response. In order to minimize this error, more settling time must be given to the transient. The number of time constant cycles (X) needed for the sampling circuit to settle within $\frac{1}{2}$ LSB in a n -bit ADC is as follows:

$$X \geq \ln(2 \times 2^n) \quad (5.20)$$

Table 5.1 shows a few examples of such calculation. For a 12-bit ADC, it requires at least 10 time constants to guarantee half LSB₁₂ settling.

The second term in Equation 5.19 represents the magnitude attenuation and phase shift in the steady state form. This error depends on the RC time constant and cannot be reduced by extending the settling period. In the prototype, we set the RC time constant such that the maximum magnitude attenuation is less than $\frac{1}{4}$ LSB.

5.2.3 Pulse Generator

Figure 5-23 in Section 5.1.4 shows the timing waveform of the asynchronous operation, where we include “READY” and “PULSE” signals in the timing waveform. The “READY” signal represents that the comparison result is ready to be latched onto the register, and the width of the “PULSE” represents the total time that is allocated for DAC voltage settling. As discussed in the previous section, redundancy can be used to improve sampling frequency if designed correctly. As a result, the pulse width used here is programmable to accommodate different sampling rates and to test the limit of the speed operation.

The circuit implementation to generate the two signals is shown in Figure 5-34. The circuit is designed to minimize the timing of the critical path, which includes the delay through the comparator, the digital control logic and the DAC settling time. First, observing that the voltage on the output nodes of the comparator (OUT+ and OUT-) are reset to V_{DD} at the end of each conversion cycle, this voltage drop in either node would indicate the start of the latching process. The circuit on the left of Figure 5-34 implements the ready detection function. Rather than using the conventional XOR gate to detect when the output of the comparator is ready, we use dynamic logic here. This can significantly improve the power efficiency and the speed of the operation. The “READY” node, in this case, is pre-charged to V_{SS} before the start of each comparison. The OUT+ and OUT- go down together initially to almost $V_{DD}/2$ before one moves back to V_{DD} and the other one goes down to V_{SS} . As a result, both PMOS transistors M1 and M2 are turned on to charge the “READY” node in the beginning. This can significantly speed up the “READY” signal generation. The timing diagram of the key circuit nodes is given in Figure 5-35.

The “READY” signal activates the pulse generation circuit to generate a pulse.

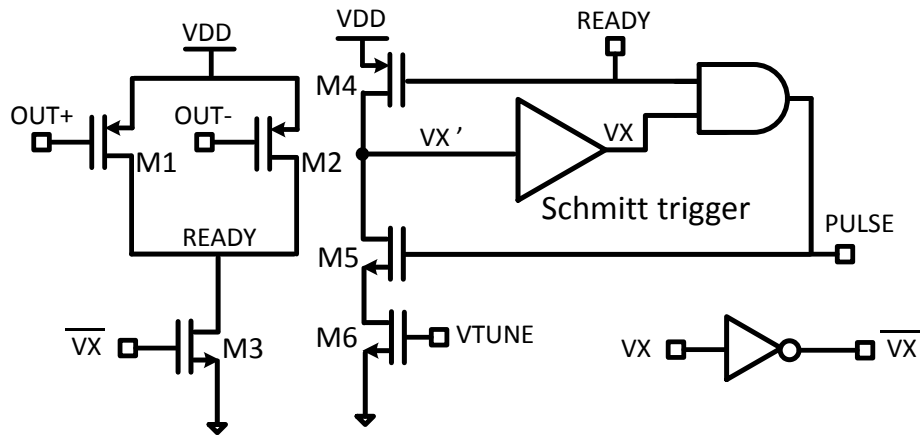


Figure 5-34: Asynchronous pulse generator. A Schmitt trigger is added to avoid voltage spikes in dynamic operation and to improve the robustness against noise.

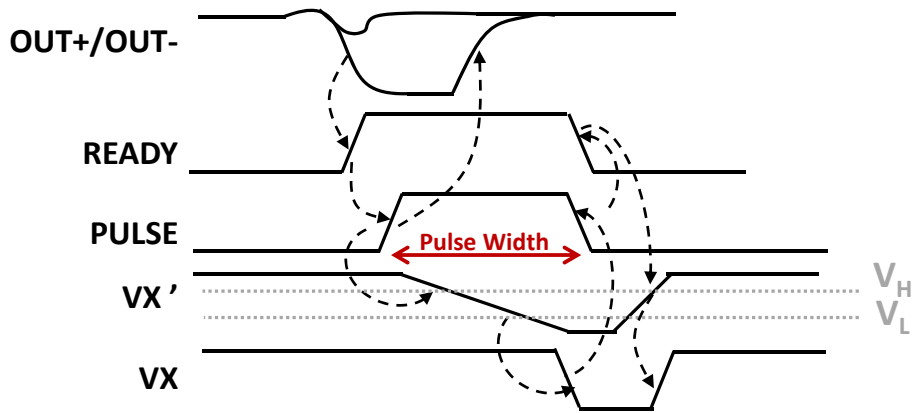


Figure 5-35: Timing diagram for the asynchronous pulse generation. By tuning the node VTUNE, the pulse width can be increased (or decreased) to slow down (or speed up) the asynchronous operation.

This pulse creates a signal to reset the comparator, and at the same time, it also starts the charge redistribution process on the DAC. The width of this pulse is designed to allow sufficient time for the DAC voltage to settle within $\frac{1}{2}$ LSB. The circuit for generating such pulse is presented on the right of Figure 5-34. When the ready signal goes high, it turns off transistor M4 and turns on transistor M5. The node VX' begins to discharge from V_{DD} toward V_{SS} through transistors M5 and M6. The rate of discharging is governed by the gate voltage (VTUNE) of the transistor M6. The VTUNE voltage is made tunable in this case to allow programmable pulse width. In the original design of pulse generator in [50], no buffer is inserted between the node VX' and digital logic gates. Since dynamic logic is more sensitive to coupling noise, it may introduce a voltage spike on the VX' node that causes a false pulse. In our implementation, we introduce a Schmitt trigger between the two nodes. A Schmitt trigger has different transition thresholds for inputs moving in opposite directions. As we can see from Figure 5-35, the voltage on VX' must move below V_L for the buffer to trigger and pull down VX to V_{SS} . This adds additional noise immunity and improves the overall reliability of the circuit.

By the time node VX goes low, the comparator is already reset for a sufficient amount of time to allow nodes OUT+ and OUT- to return back to V_{DD} , thus turning off transistors M1 and M2. Therefore, when node VX goes low, it can discharge the ready signal back to V_{SS} through transistor M3 without having to compete with the pull-up transistors (M1 and M2) and avoids any short-circuit current. Meanwhile, the pulse signal also gets pulled down to V_{SS} , signaling the conclusion of the DAC settling period. The pulled-down PULSE will turn off transistor M5 to stop the discharging on node VX'. The pulled-down ready signal will enable transistor M4 to start charging VX' back to V_{DD} . The falling edge of the pulse signal serves as the stroke signal for the latch comparator. The comparator will begin the next comparison and the same procedure repeats.

In real implementation, we put two pulse generators in parallel and the ADC can be configured to activate either pulse generator. This allows the design to have two modes of operations: a fast mode and a slow mode. The fast-mode pulse generator

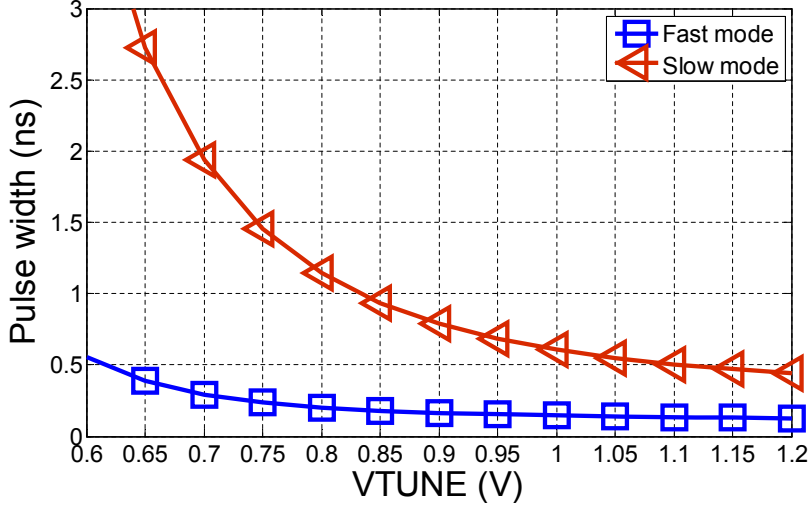


Figure 5-36: The pulse width in the fast and slow modes of operation. The slow mode is designed for debugging purposes.

can produce any pulse width between 100ps and 400ps and the slow-mode pulse generator can produce any pulse width between 400ps and 2ns. Using two pulse generators together, our design can generate any pulse width between 100ps and 2ns. The VTUNE voltage is only tuned between 650mV and 1.2V, since in this input range, transistor M6 operates in the linear region and the pulse width changes slowly with the gate voltage.

5.2.4 Capacitive DAC array

In Chapter 4, we introduced a calibration algorithm that is able to calibrate out the capacitor mismatch due to manufacturing variation. Therefore, the size of the unit capacitor in our design is limited by thermal noise. Figure 5-37 shows the simplified noise model for the RC sampling network. The resistor generates a thermal noise that can be expressed according to Equation 5.21. The thermal noise is approximately white, implying that its power spectral density is constant over the entire frequency spectrum.

$$S_{n,R}(f) = v_{n,R}^2(f) = 4kTR \quad (5.21)$$

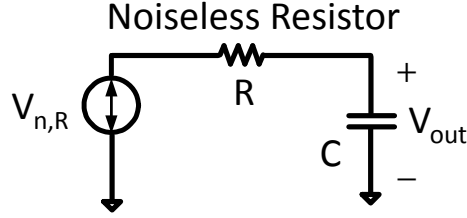


Figure 5-37: A simple noise model for sampling circuits.

where k is the Boltzmann constant (1.38×10^{-23} J/K) and T is the temperature expressed in Kelvin. The total mean-square noise is the integral of $S_{n,R}(f)$ over the entire frequency range. If the noise is not filtered, the integration is infinity. In real implementation, however, the circuit cannot be designed to have infinite bandwidth, and therefore, the spectral noise is always filtered, resulting in finite total integral noise. The input noise signal in Figure 5-37 is filtered by the low-pass RC circuit. The transfer function between the input and output is given in Equation 5.22.

$$\frac{v_{out}}{v_{n,R}}(s) = \frac{1}{RCs + 1} \quad (5.22)$$

The output noise spectral density can be obtained by multiplying the input spectral density by the power transfer function. The power transfer function is the square magnitude of the transfer function given in Equation 5.22, and the resulting output noise spectral density is given in Equation 5.23.

$$S_{out}(f) = 4kTR \frac{1}{(2\pi fRC)^2 + 1} \quad (5.23)$$

The total mean-square noise can be obtained by integrating the output noise spectral density over the entire frequency range as given in Equation 5.24.

$$S_{out} = \int_0^{\infty} 4kTR \frac{1}{(2\pi fRC)^2 + 1} df = \frac{kT}{C} \quad (5.24)$$

Even though the noise is generated by the resistor, from Equation 5.24, we see that the sampling noise here is only a function of the capacitor. This is because the spectral noise density is proportional to the value of the resistor, but the bandwidth of the

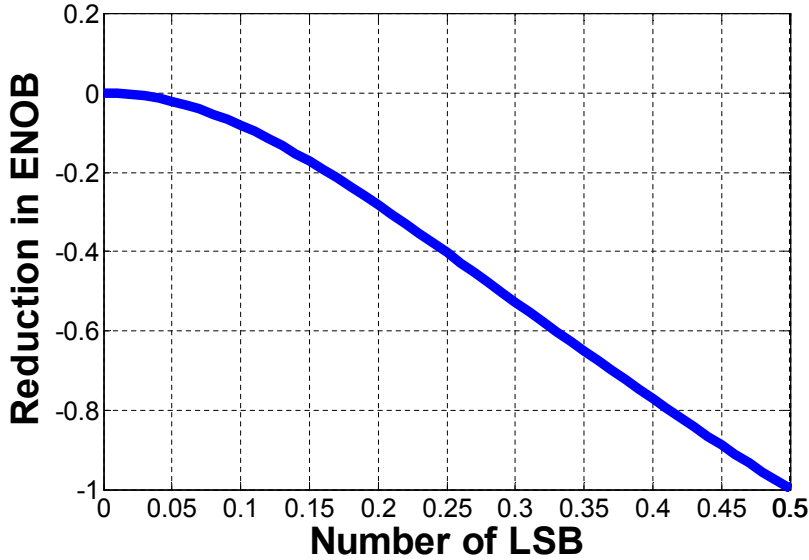


Figure 5-38: Reduction in ENOB due to thermal noise. Here, the thermal noise is in the unit of LSBs.

sampling network is inversely proportional the value of the resistor. When multiplying the two factors together, the effect of the resistor cancels out. Even though the total noise in this case is only a function of the capacitor, it is still important to know that the power spectral density is a function of the resistor in the case when noise is further filtered by subsequent circuits.

Figure 5-38 shows a plot of the reduction in effective number of bits versus the amount of thermal noise. The amount of thermal noise is expressed in relation to the size of the LSB signal. As shown in the plot, when the thermal noise is $0.5 \times \text{LSB}$, the ENOB is reduced by 1 bit. In our design, we pick the total noise to be roughly equal to $0.15 \times \text{LSB}_{12}$.

The capacitive DAC is designed using metal-oxide-metal (MOM) capacitors. Even though the MOM capacitors have more parasitic capacitance and worse matching compared to metal-insulator-metal (MIM) capacitors, they are compatible with any standard digital process without any specialized process options. On the other hand, MIM capacitors require a more specialized process, which are not offered in all digital technologies. This may cause the SAR architecture to lose one of its main advantages, its digital compatibility. The layout of the capacitive array presented in Figure 5-22

1	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D
2	D	50M	50M	24M	D	D	D	D	D	D	24M	50M	50M	D	
3	D	50M	50M	24M	3M	D	D	D	D	D	24M	50M	50M	D	
4	D	50M	50M	24M	3M	D	D	D	D	D	4M	24M	50M	50M	D
5	D	50M	50M	24M	4M	16M	10M	D	10M	16M	4M	24M	50M	50M	D
6	D	50M	50M	24M	4M	16M	10M	2M	10M	16M	3M	24M	50M	50M	D
7	D	50M	50M	24M	16M	16M	10M	2M	10M	16M	16M	24M	50M	50M	D
8	D	50M	50M	24M	16M	16M	10M	1M	10M	16M	16M	24M	50M	50M	D
9	D	50M	50M	24M	16M	16M	10M	B	10M	16M	16M	24M	50M	50M	D
10	D	50M	50M	24M	#	8L	4L	B	4L	8L	#	24M	50M	50M	D
11	D	50M	50M	24M	#	8L	4L	3L	4L	8L	#	24M	50M	50M	D
12	D	50M	50M	24M	#	8L	1L	3L	1L	8L	#	24M	50M	50M	D
13	D	50M	50M	24M	#	8L	0.5L	3L	0.5L	8L	#	24M	50M	50M	D
14	D	50M	D	D	#	#	#	#	#	#	#	D	D	50M	D
15	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D
16	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

MSB DAC
LSB DAC
Bridge cap
Dummy cap on LSB
Dummy cap for layout

Figure 5-39: Capacitor layout for our DAC. Even though the capacitors are not binary weighted, common-centroid layout practice is still employed here to minimize mismatch.

is shown in Figure 5-39. The cells in yellow represent the MSB part of the capacitive DAC, the cells in dark blue represent the LSB part of the capacitive DAC, the cells in green are the bridge capacitor, the cells in orange form the dummy capacitor C_X placed on the LSB DAC, and the cells in light blue form the dummy capacitors to ensure better layout matching.

Common-centroid layout practice is also used here. The unit capacitor is sized at $11.5fF$ with roughly $1.6pF$ of total capacitance. Even though the calibration algorithm can be used to remove the mismatch due to manufacturing variation, it is still a good practice to try to minimize variation as much as possible. As discussed in Chapter 3, although calibration can remove mismatch, larger mismatch between capacitors requires more redundancy. Since the amount of redundancy is pre-configured in the design and cannot be changed after the chip is fabricated, if too much variation occurs after fabrication exceeding the capability of the redundancy, calibration may

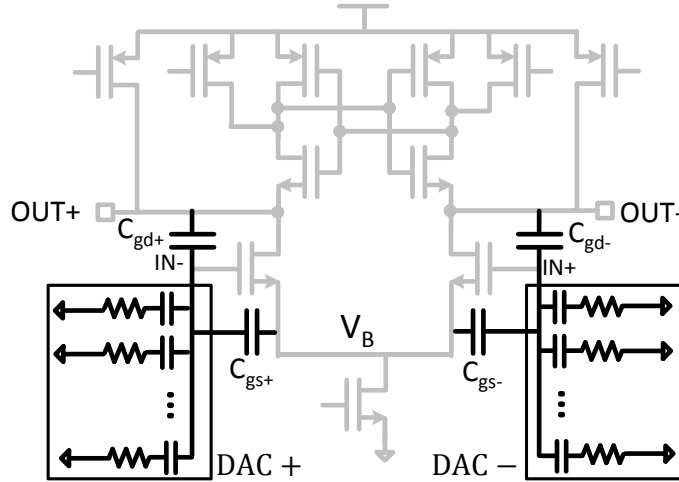


Figure 5-40: Kickback noise generation. Unequal charges are injected onto the input nodes if the impedances looking back are different.

not able to completely correct the mismatch.

5.2.5 Kickback Noise

Figure 5-40 shows the latch comparator used in our prototype. During the reset phase, $OUT+$ and $OUT-$ both get pulled up to V_{DD} . When the clock goes high, M_{CLK} pulls V_B towards ground and begins discharging $OUT+$ and $OUT-$. This rapid change in voltage on V_B and the output nodes will capacitively couple charges onto $IN+$ and $IN-$ through device capacitance C_{gs} and C_{gd} of the input pairs. The disturbance of the input nodes due to the rapid change in voltage on the internal nodes of the comparator is called the kickback noise.

If the on-resistances of the switches connected to the DAC capacitors are mismatched, the total impedance looking into the $IN+$ and $IN-$ nodes would be different. As a result, when V_B suddenly drops in voltage, different amount of charge is injected onto the gates of the input pairs. If the sign of the injected charge is different from the sign of the sampled input, when the injected charges are unequal, it can flip the polarity of the comparator output and produce erroneous results. Moreover, the mismatch in kickback noise is signal-dependent since the impedances looking into the gate terminal of the input pairs depend on the configuration of the DAC. During

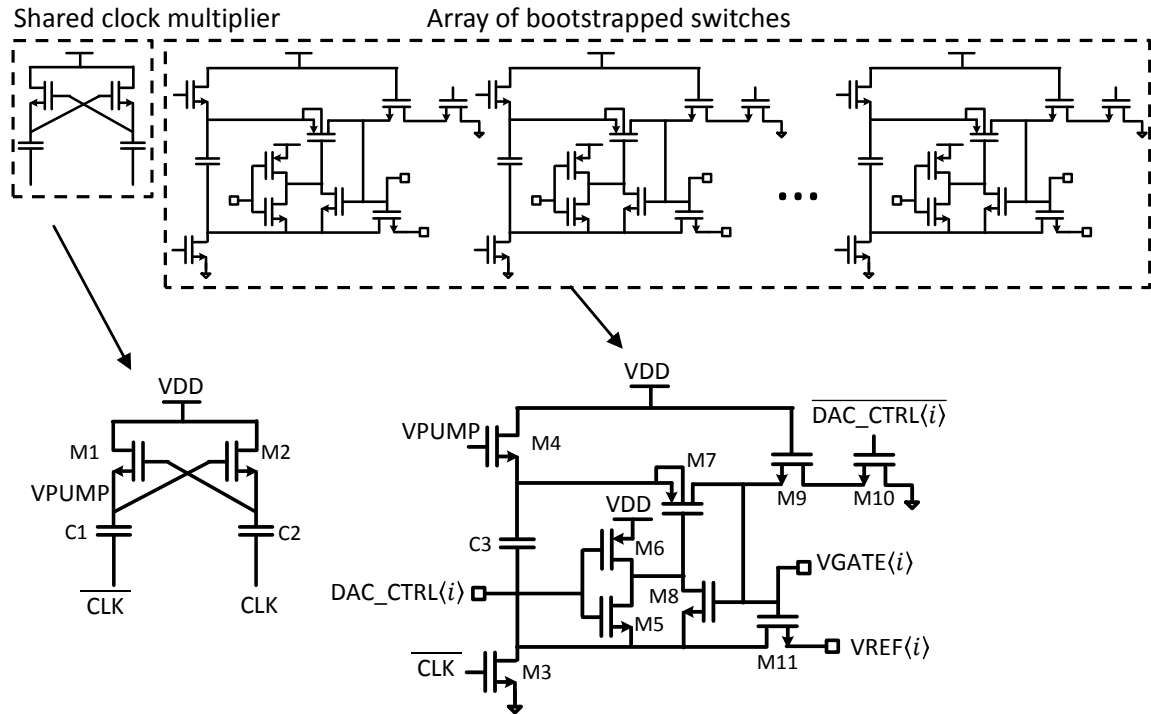


Figure 5-41: Array of bootstrapped switches to reduce the effect of kickback noise. All the switches share a common clock multiplier circuit.

the conversion process, the DAC configures itself to generate the closest digital estimate of the current input signal. As a result, the kickback noise can be highly signal dependent, which can lead to a high level of harmonic distortion at the output.

If the on-resistance of the switches connecting the reference voltages to the bottom plates of the capacitors are similar enough, the kickback noise affects the input pairs equally, which consequently do not introduce any harmonic distortion. In the inverted merged capacitor switching (IMCS) algorithm, there are three different types of reference voltages, V_{CM} , V_{REF+} and V_{REF-} , that can connect to the bottom plates of the DAC as shown in Figure 5-11. At the beginning of the conversion, all of the bottom plates are connected to V_{CM} . Depending on the comparison result, the capacitors in the same position of the differential DAC (DAC+ and DAC- in Figure 5-40) would switch to opposite references, one to V_{REF+} and the other one to V_{REF-} . Therefore, in order to match the impedance looking into the two terminals, we have to match the on-resistance of the switches connected to V_{REF+} and V_{REF-} . Traditionally, an NMOS switch is used to charge the bottom plate to V_{REF-} and a PMOS switch is

used to charge the bottom plate to V_{REF+} ; however, it is difficult to match the on-resistance of NMOS and PMOS switches, especially in the presence of process and temperature variations.

To avoid this issue, two design techniques are adopted here. First, the switches are sized large enough to have low impedance. Second, only NMOS switches are used even for switches charging to V_{REF+} . All of the NMOS switches are bootstrapped in this case to lower the on-resistance and to improve matching between the switches charging to opposite reference voltages. This ensures that the coupling noise to the two input terminals is approximately equal even under process and temperature variation. Figure 5-41 shows the schematics of the array bootstrapped switches. The switches are built using low-threshold-voltage (LVT) devices to further reduce their on-resistance. The basic structure of the array bootstrapped switches is the same as the bootstrapped switch used in the sampling circuit. One difference is that there is only one clock multiplier circuit and it is shared among all the switches. During the reset stage, the capacitor $C3$ of all the switches is pre-charged to V_{DD} . After pre-charging, the transistors M3/M4 are turned off. Transistors M10/M11 do not turn on immediately after transistors M3/M4 turn off as in the sampling circuit. Rather, the i^{th} switch only turns on when the i^{th} bit decision is ready. All transistors inside the bootstrapped circuit are scaled proportional to the size of the capacitor that it is designed to charge or discharge.

5.3 Summary

In this chapter, we discussed a real implementation of a redundant SAR ADC. With many newly-developed architectural techniques, this design allows improved energy efficiency, less digital complexity, smaller die area and input load, higher signal-to-noise ratio (SNR), easier implementation and better dynamic noise/error tolerance than previous approaches. In the second part of this chapter, key circuit building blocks are presented, along with a discussion of the main design challenges and limitations. Various circuit techniques are used here to resolve and improve upon existing

solutions. The system is simulated with post-layout extracted model in SpectreRF. The output is sent to Matlab for further analysis of noise and linearity. The operations and the accuracy of the ADC are verified.

Chapter 6

Packaging, Test Setup and Measurement Results

In this chapter, we focus on the silicon implementation of the design. The chapter is divided into three sections. The first section discusses the floor planning and the placement of the IO pins; these are important issues because the location of the pins can affect the amount of signal feedthrough and noise coupling, which subsequently reduces the achievable accuracy. The second section focuses on the setup of the testing environment. It includes discussion of the testing equipment, the making of the printed circuit board (PCB), and the testing flow that allows different blocks to communicate and enables the collection of final measurement results. Finally, in the last part of this chapter, we discuss the measurement results of our fabricated chip. The design has two versions, the first version includes more configurability than the second version; however, because of this extra configurability, it consumes more power compared to the second revision. Figure 6-1 shows the die photo of the second fabricated chip in TSMC 65nm technology.

6.1 Packaging

The mixed-signal nature of our SAR ADC design contains both analog and digital circuit. Combining both analog and digital components onto a single system frequently

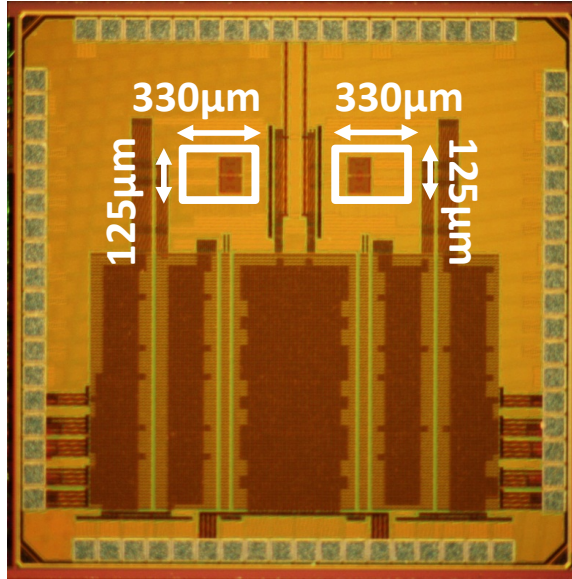


Figure 6-1: Die micrograph of the fabricated chip in TSMC 65nm technology.

leads to noise issues. Therefore, it is important to consider the potential noise sources at the design stage to prevent degradation in accuracy and difficulties during chip debugging. To capture all the noise sources, simulation must include as much of the existing parasitic capacitance and inductance effects as possible; however, this can significantly increase the simulation time and complexity, and it becomes harder to complete such simulations as the design gets larger. Moreover, it is difficult to capture all of the possible noise effects, especially those related to off-chip components. In this section, we introduce a few guidelines that we followed to determine the pin placements, power/ground decoupling capacitors, interconnect coupling and the design of our printed circuit board (PCB) in order to create a robust system against noise.

At high frequencies, the impedance associated with the interconnection and bond wires can significantly affect the stability of the internal power and ground. For the digital section where ringing can be less of an effect, the simplest solution is to add more parallel power/ground pins to reduce the total impedance. To reduce noise in the analog section, a common practice is to separate the digital and analog power supplies to improve the isolation between the noisy digital switching and quieter analog switching. Independent power/ground interconnect should be considered for

analog cells that have large current transients.

The configuration in Figure 6-2(a) is very sensitive to noise coupling and is poor for stability in power supplies. This configuration provides no isolation between the analog and digital cells, which means that any transient current that occurs in either analog or digital cells will directly couple onto the other cell. A better design for noise isolation is shown in Figure 6-2(b), where an internal resistors are added to help reduce the noise coupling between the two blocks. This method is typically adopted when there is limited number of output pins available for power supplies. The best configuration to provide maximum noise isolation is shown in Figure 6-2(c). It uses separate pads for analog and digital power supplies and grounds. There exists no direct path for noise coupling in this case. It is important to note that even using the separate pins, noise coupling can still occur through the substrate, parasitic capacitors, and mutual inductance. Therefore, the noisy circuit should be placed as far from the quiet analog blocks as possible.

To further reduce the power supply noise, decoupling capacitors between the power and ground nodes should be used. Off-chip capacitors have a maximum frequency at which they behave as a capacitor, because capacitors also have a series inductance. At certain frequencies, the impedance begins to look more inductive than capacitive. Large capacitors have lower self-resonance frequencies. As a result, to maintain capacitive characteristic even at high frequencies, it is important to put capacitors with different values in parallel. To achieve better results, off-chip capacitors should be placed as close to the packaged chip as possible.

At high frequencies, off-chip capacitors are not able to provide as much filtering due to the bond-wire, interconnect and package inductance. Typically, bond wires have between 1-2nH inductance. On-chip decoupling capacitors are needed to reduce this high-frequency noise. Empty areas on the die can be filled with decoupling capacitors. The capacitors are built by putting layers of MOM capacitors (using all metal layers) and MOSCAP in parallel. Instead of using the core devices, the MOSCAP is built with I/O transistors to reduce the leakage current. This can provide significant improvement in power supply noise without increasing the size of the die.

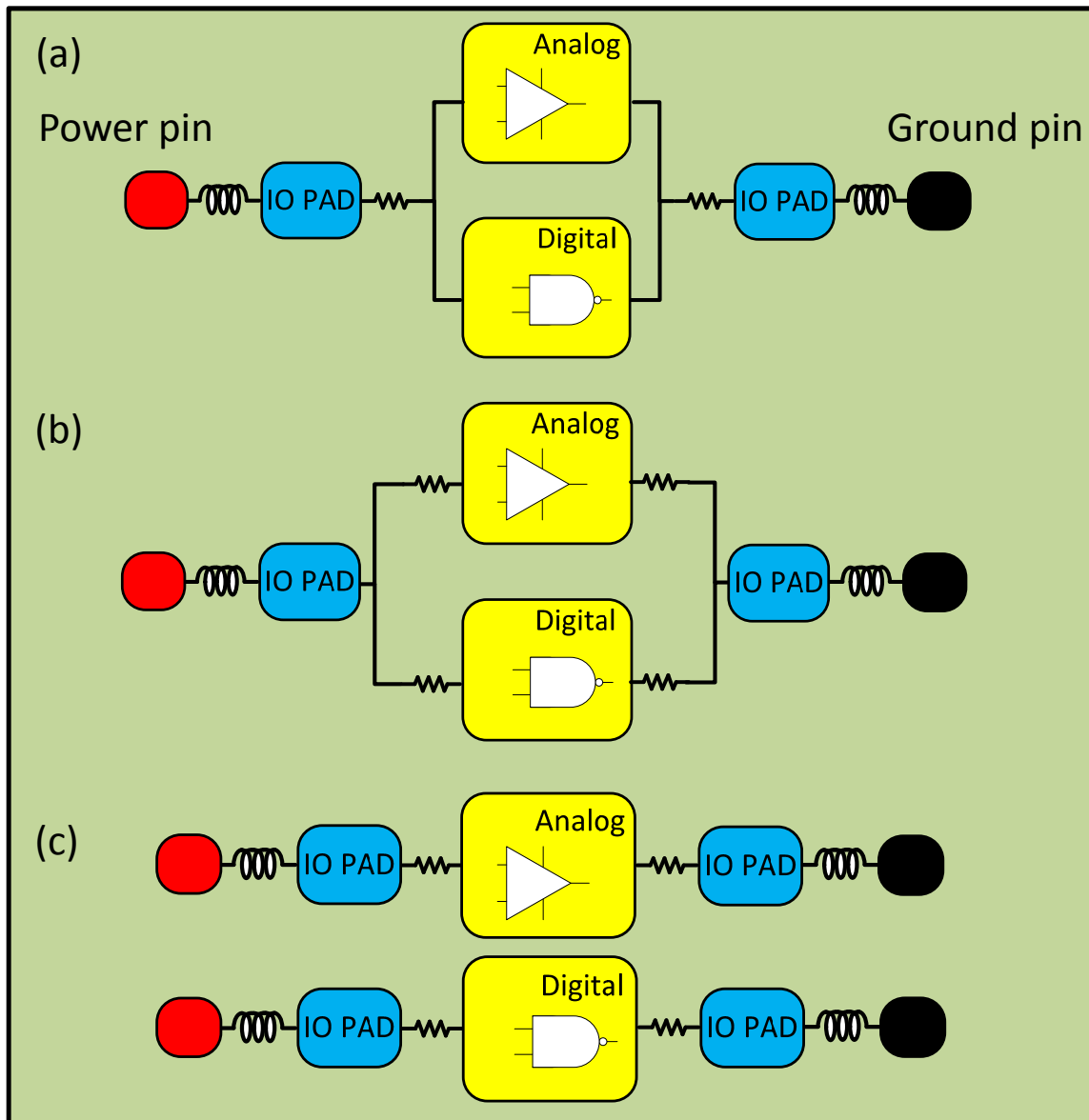


Figure 6-2: Separation between the analog and digital supplies can help improve isolation to reduce noise coupling. Our design uses approach (c) above.

Combining the techniques introduced previously help provide significant filtering that enables lower impedance and cleaner supplies for the analog circuits on the die. Long analog signal routing is avoided to prevent potential noise coupling. Digital and analog parts of the circuit are kept as far away from each other as possible. For the cases where analog and digital wiring cannot be separated, shielding is used to reduce noise coupling. Shielding is done by routing the sensitive analog signal in between two quiet ground wires.

The same theory can be applied to I/O pin placement. Analog output pins should be separated from the digital output pins, preferably in the opposite side of the chip. Neighboring analog output pins are also isolated by quiet pins. Typically, the quiet pins include the ground pin and the static digital pins that are used for programming or enabling the chip. The bonding diagram and the location of the I/O pad placement are shown in Figure 6-3. The die has a total of 72 output pins and the QFN package has a total of 56 output pins. The package has less pin counts compared to the die since many pins are ground pins, which are bonded directly to the round paddle. The design follows the general guideline introduced previously. As shown in the diagram, all of the sensitive analog signals are placed on the top side of the die, the noisy digital output pins along with digital power supply (VDD and VDDIO) are placed on the left and right side of the die, and finally, the quiet analog power supply (AVDD) and the reference voltages (VREF+, VREF- and VCM) are placed on the bottom side of the die. Sensitive pins are shielded by ground pins. For example, the clock signal (clk) and the differential input signal (VP and VN) are separated by analog ground pins, the reference voltage (VCM), or the analog power supply (AVDD). As many power and ground pads are used as are available in order to lower the on-resistance and the inductance of the bond wire. The empty areas are filled with decoupling capacitors. A total of $1.8nF$ decoupling capacitors are placed between VDD/VSS, AVDD/VSS and VREF+/VREF- on-chip.

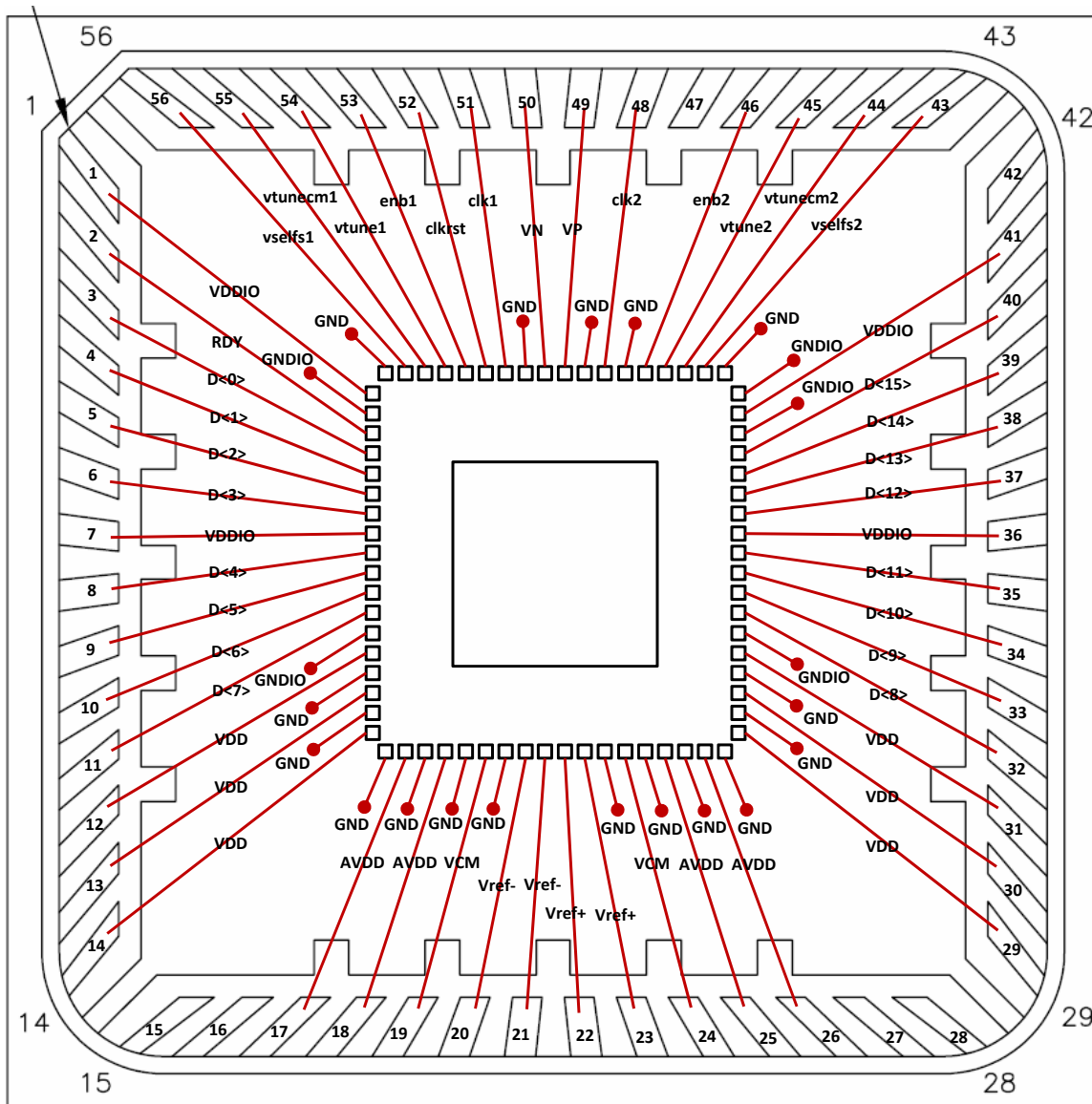


Figure 6-3: Die bonding diagram, following the design principle described in Section 6.1.

6.2 Test Setup

The ADC test setup is an important part of the design in order to obtain accurate measurement results. After the design comes back, the first step is to choose a suitable package for the die. To minimize the parasitic inductance of the package, QFN packages were used. Other options, such as chip-on-board (COB), can be used to further reduce the bond wire inductance, but this option greatly increases the complexity of post processing. The second step is to design a printed circuit board (PCB) that can be used as an interface between the dies and external test equipment. Linear regulators with low dropout voltage (LT3021) are used on the PCB to supply clean power and reference voltages. Separate linear regulators are used for different supplies and references. As a result, there are a total of six regulators on the PCB in order to generate VDD, AVDD, VDDIO, VREF+, VREF- and VCM. The separation of analog power supply, digital power supply and reference voltage ensures low noise coupling between them. The tuning voltage (VTUNE) that controls the pulse width of the conversion period is generated by an external DAC.

The ADC has different speed configurations that can be programmed externally through the XEM6010-LX45 FPGA board made by Opal Kelly. The same FPGA is also used to program the DAC to create the correct tuning voltage (VTUNE). A single ended sine wave signal is generated using an Agilent 8644B signal generator. A band-pass filter is used at the output of the signal generator to improve the purity of the input signal. This input signal is then transformed into differential signal using two transformers made by mini-circuit (ADT1-6T+). The use of two transformer helps reduce the second order distortion caused by the transformers. The CMOS clock is generated by a Synthesized Clock generator (CG 635) made by Stanford Research Systems. Two such clock generators, outputting clocks with the same frequency, are phase-locked together with a phase offset. The two waveforms are then regenerated on-chip and combined using an AND gate. This generates a clock waveform with non-50% duty cycle. The phase difference between the two clock signals is controlled externally, and this circuit generates a clock with variable duty cycles. The duty

M	$C_{designed}$	$C_{extracted}$
15	1750	1733.21
14	840	838.73
13	560	561.04
12	350	354.03
11	210	211.90
10	140	144.55
9	105	106.60
8	70	70.17
7	35	35.62
6	16	17.75
5	8	8.89
4	6	6.73
3	2	2.26
2	2	2.26
1	1	1.19
1	1	1.02

Table 6.1: Designed versus extracted capacitor values. Some capacitors show large discrepancy between the designed and the extracted values. These results confirm that calibration is necessary to achieve high resolution in a SAR ADC design.

cycle here represents the amount of time given for sampling. This variable duty cycle allows us to explore the ADC performance by changing the time allocated for the sampling and the conversion phases. The digital output data is collected by a logic analyzer (TLA715) made by Tektronix. The probes of the logic analyzer adds very small capacitive load to the output drivers of the die. A block diagram that shows the testing flow is given in Figure 6-4.

6.3 Measurement Results

The prototype ADC is fabricated in standard TSMC 1P9M 65nm low-power CMOS technology with 1.2V supply voltage. Two identical channels are time-interleaved. The active area of each channel is roughly $0.0412mm^2$ ($330\mu m \times 125\mu m$) with the total active area of $0.083mm^2$. The implementation allows full input swing of $2.4V_{P-P}$. The DAC is implemented with standard MOM capacitors from metal 3 to metal 5 with a total capacitance of $1.6pF$. Several chips are measured and all measurements

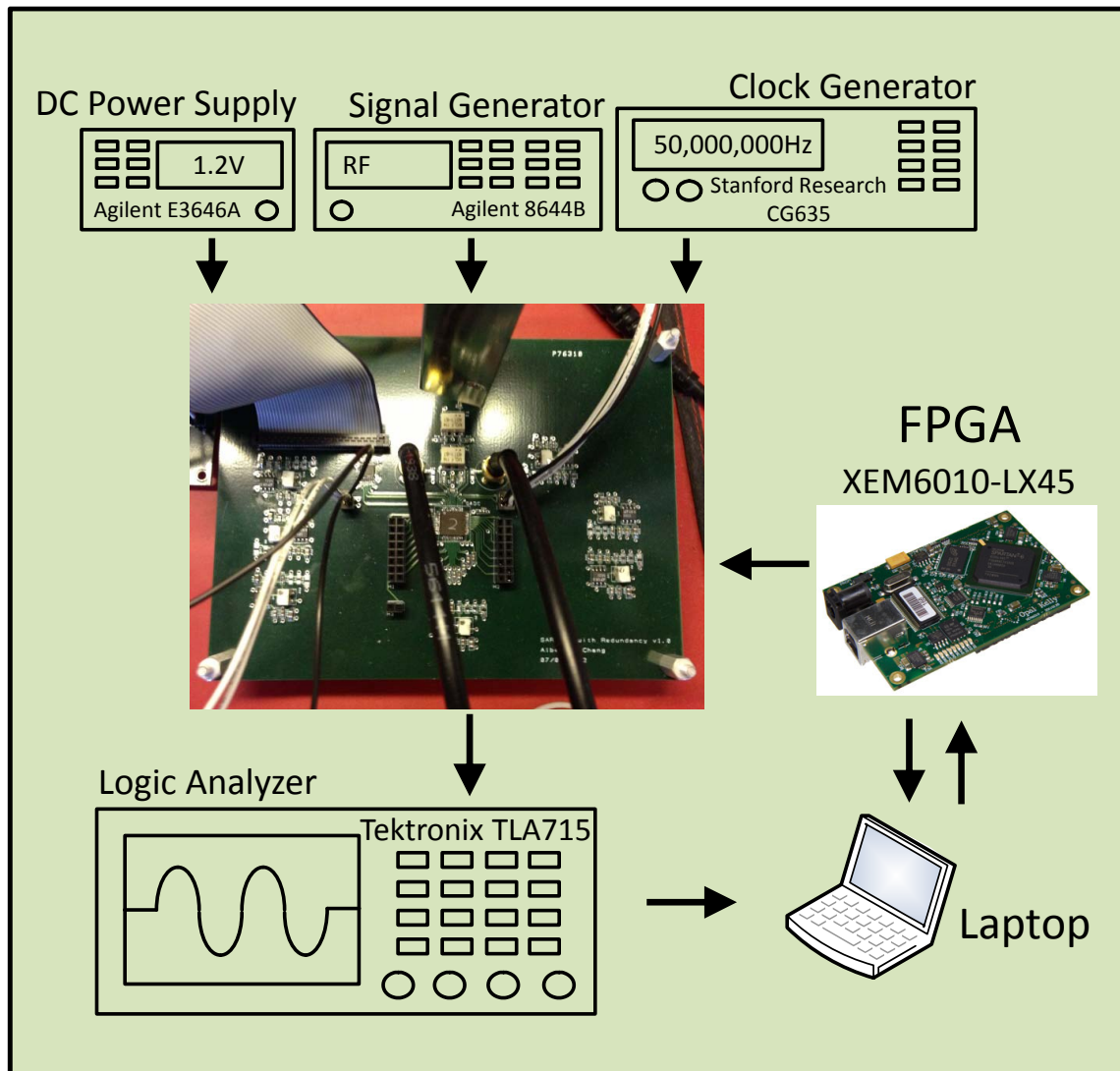


Figure 6-4: ADC evaluation test setup. The setup includes a DC power supply, a signal generator, a clock generator, a logic analyzer, a FPGA and our PCB board.

are performed at room temperature.

While the ADC operates at 50MS/s, a 24.7MHs full-scale sine wave input is used to test the static and dynamic performance. Figure 6-5 shows the measured DNL and INL before and after the calibration. Before calibration, the maximum DNL and INL errors are $+1.3/ - 1.1\text{LSB}_{12}$ and $+14.3/ - 14.0\text{LSB}_{12}$, respectively. The linearity is mainly limited by the capacitor mismatch. The calibration algorithm, introduced in Chapter 4, is applied here on the collected data to extract the actual capacitor sizes. A sine wave input is used here as the calibration stimuli and 1 million data points are collected to ensure that the information is statistically significant. The designed and extracted capacitor values are shown in Table 6.1. After calibration, the maximum DNL and INL errors improve to $+0.5/ - 0.7\text{LSB}_{12}$ and $+1.0/ - 0.9\text{LSB}_{12}$, respectively.

Figure 6-6 shows the measured dynamic performance of the SAR ADC, calculated based on 8192-point FFT. Before calibration, the ADC has 51.4dB of SNDR, 51.9dB of SFDR and achieves 8.2b ENOB. After calibration, the ADC has 67.4dB of SNDR, 78.1dB of SFDR and achieves 10.9b ENOB. The calibration engine is built in simulation and the estimated power is roughly $68\mu W$ running at 50Hz. The total power consumption including the estimated calibration and reference power is $2.1mW$, corresponding to $21.9fJ/\text{conv.-step}$ of FoM . The SNDR/SFDR versus input frequency at 50MS/s before and after calibration is shown in Figure 6-7. The same figure also shows the performance summary of the ADC. Figure 6-8 shows the comparison with the state-of-the-art as published in ISSCC and VLSI between 2009 and 2012. This ADC achieves the best FoM of any ADC that has resolution greater than 10b ENOB and speed over 10MS/s.

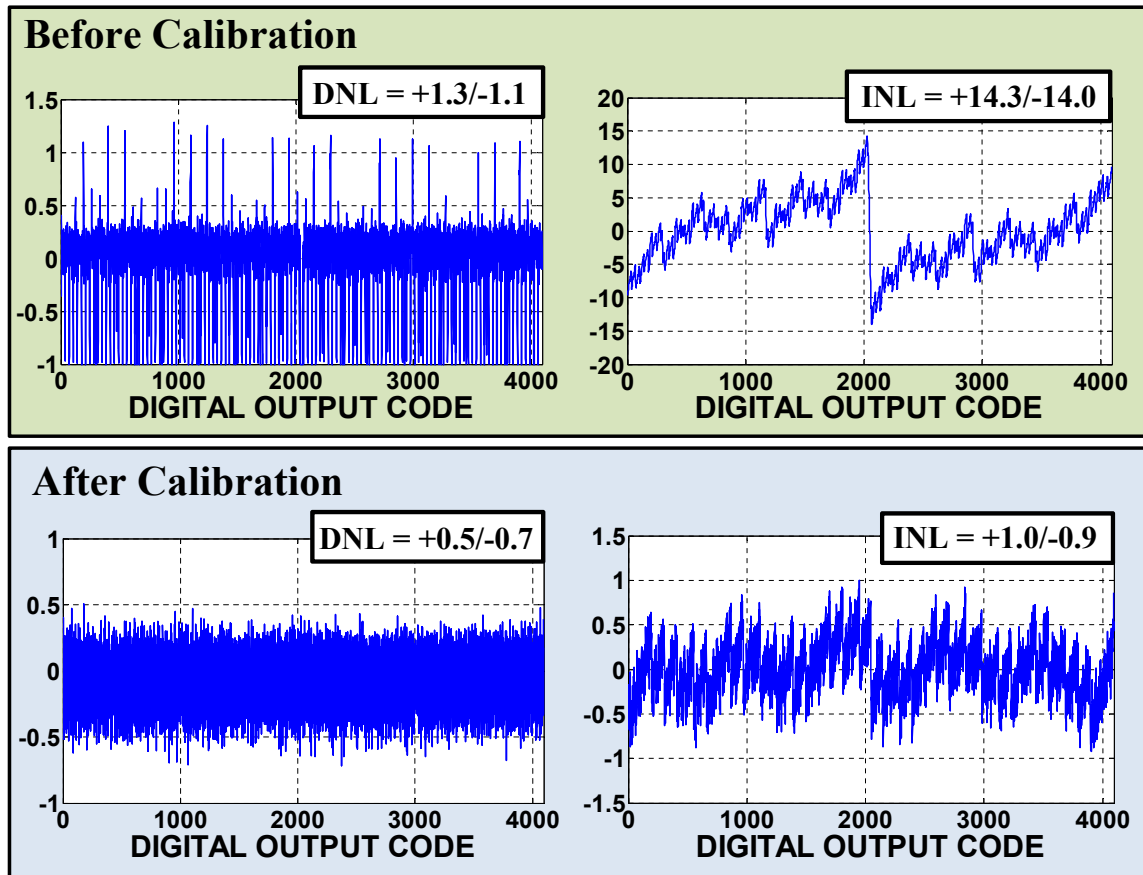


Figure 6-5: Measured DNL and INL for 12-bit resolution with 1.2 supply at 50MS/s with 24.7MHz input sine wave.

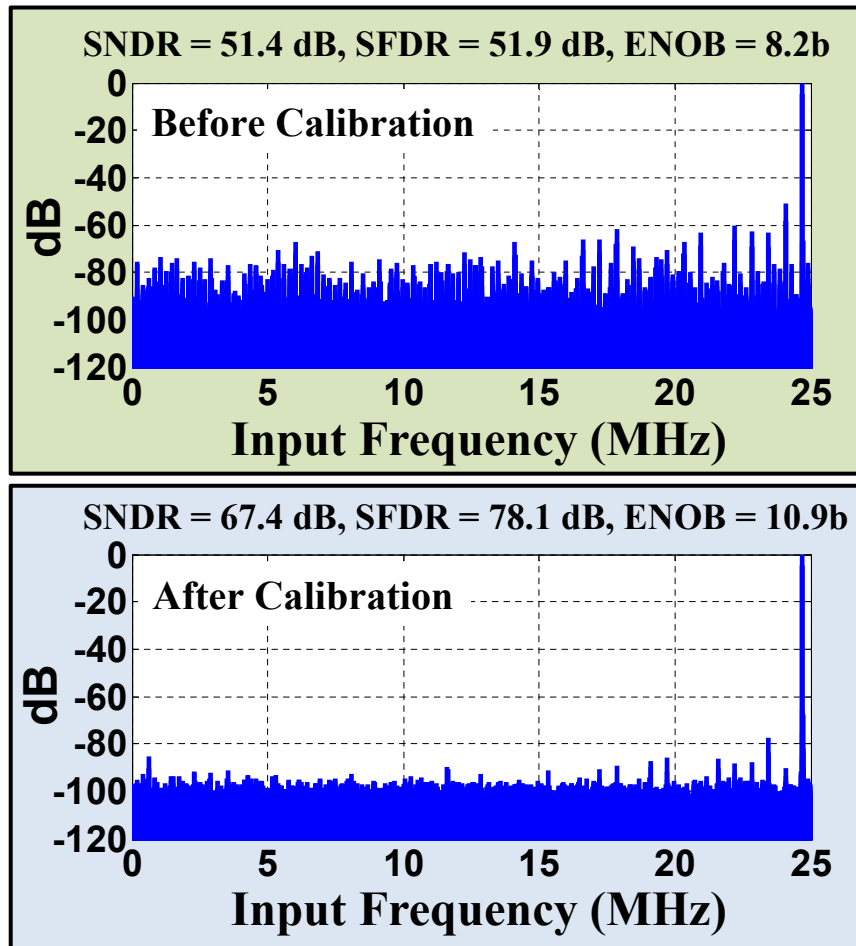


Figure 6-6: Measured spectrum data for 12-bit resolution with 1.2V supply at 50MS/s with 24.7MHz input sine wave.

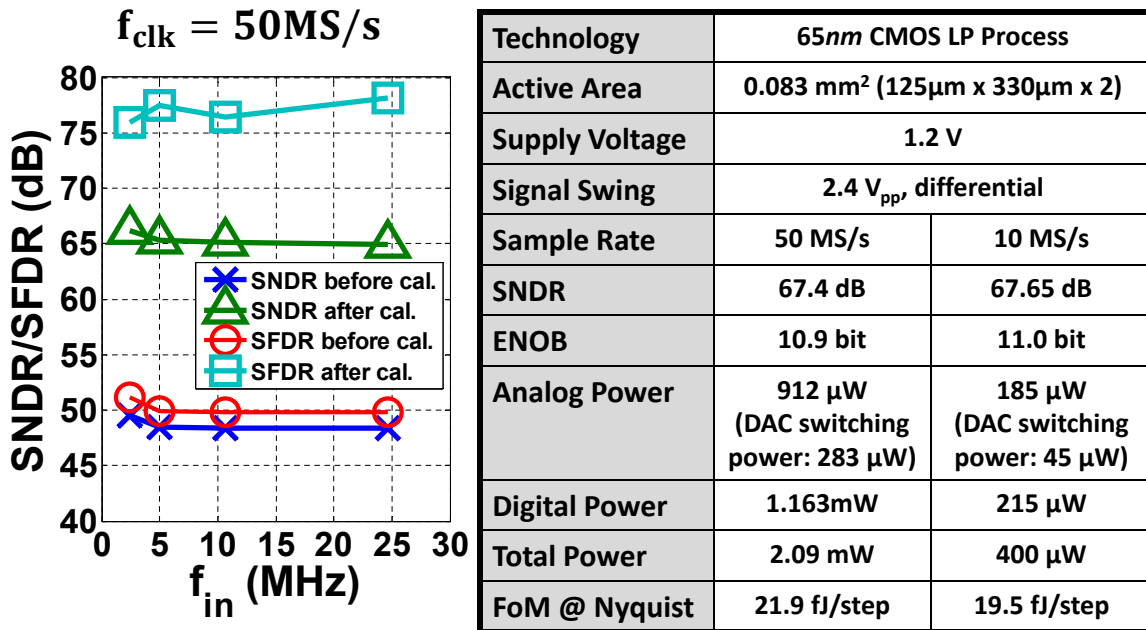


Figure 6-7: Measured SNDR and SFDR at different input frequencies and the summary of measurement result.

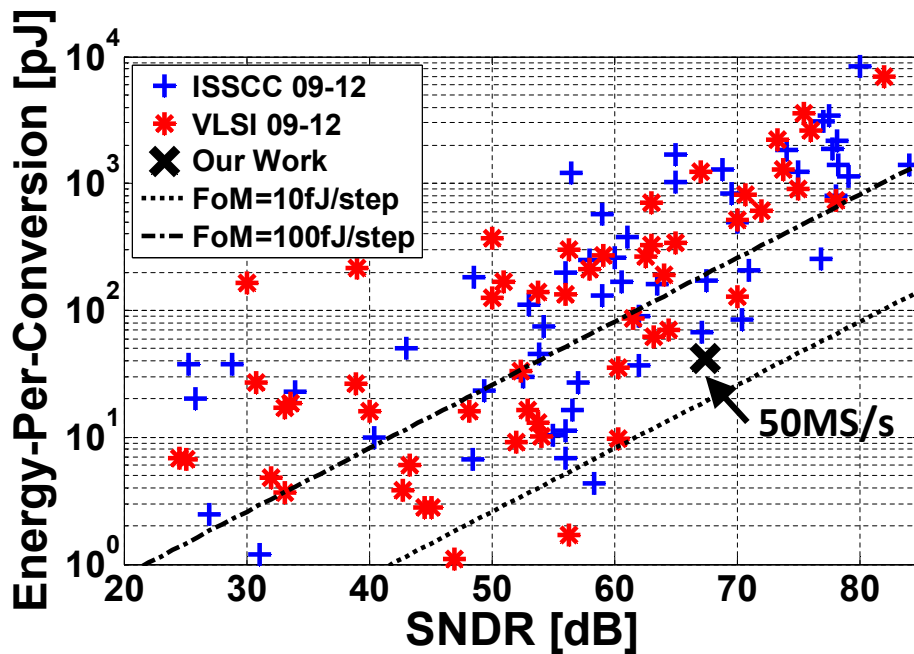


Figure 6-8: Comparison with the state-of-the-art (data adopted from [1]).

Chapter 7

Conclusion and Future Work

7.1 Conclusion

Continuous and aggressive scaling of CMOS technology has dramatically increased the energy efficiency, speed and the amount of integration of electronic systems. The improvement in system performance has driven the need for corresponding improvements in data converter performance, which serves as the interface between the analog front-end and digital back-end circuitry. The trend is to shift the analog-to-digital conversions “upstream” so that more processing can be done in the digital domain in order to perform more complex algorithms and to conserve energy. This motivates the development of higher-speed and higher-precision data converters while simultaneously the ADC has to consume less power. The choice of ADC architecture also tends to follow a similar trend: architectures that are composed of more digital circuitry are becoming preferable compared to more analog-based counterparts. Because of its high digital composition, in this thesis, we focus on the design of a high-precision, high-speed and energy efficient ADC using the successive-approximation-register (SAR) architecture in deeply scaled CMOS technology.

The conventional architectural implementation of a SAR ADC and its switching algorithm are reviewed and discussed. Even though the SAR architecture is very energy efficient and has high compatibility with deeply scaled digital technologies, the accuracy and speed of the conversion process is still limited by capacitor mismatch

and incomplete DAC settling due to its high switching activities. Redundancy has been introduced to resolve both issues. We are able to show that redundancy (or sub-radix-2) search is able to provide the additional information needed for digital calibration. Conditions for digital calibratability are also discussed and derived. In general, missing codes in the input-output transfer function can be digitally corrected while missing levels cannot. Smaller radix and more conversion steps are needed to tolerate larger expected mismatch. Moreover, using redundancy, the DAC and comparator pre-amplifier settling errors made in the earlier conversion steps can be corrected in the later step. This means that if designed correctly, even though redundancy requires more conversion steps, each step takes much less time and the overall conversion speed can be improved.

Three new calibration algorithms are presented here. They are designed specifically to use the redundant information provided by our redundancy algorithm to extract the actual step sizes during the searching process. The first algorithm requires knowing the exact input signal value to the same precision as the precision we need to know in the step sizes. Even though it is an impractical solution in real implementation, this algorithm provides great intuition to understand the latter two calibration algorithms. Rather than using the input signal value, the second algorithm uses the statistics of the input signal to perform the calibration. Even though the exact input shape is not needed in this case, the calibration algorithm requires knowing the statistics of the input signal. The last algorithm further improves upon the previous two algorithms. It does not require any knowledge of the input value or the input statistics; the only requirement is that the signal has to have a smooth and non-zero probability distribution function. The effectiveness of these calibration algorithms are verified in simulation. Compared to other calibration algorithms introduced in the literature, the last calibration algorithm introduced here does not require injection of a known calibration signal, any redundant channel or a reference converter to calibrate against. This calibration algorithm is either less hardware or less algorithmically expensive.

The physical implementation of the SAR ADC is discussed. A new DAC switching

scheme is developed to achieve the highest energy efficiency while eliminating some of the shortcomings in previous work. The switching algorithm is combined with the revised main-sub-dac array architecture. This revised architecture is able to improve matching and remove the over-range problem to achieve better signal-to-noise performance. Finally, the design is able to incorporate redundancy into the SAR architecture without increasing the design complexity and area significantly. The performance requirements and implementations of various key circuit building blocks, including the latch comparator, sampling circuit, pulse generator and DAC design/layout, are described. These blocks are optimized for energy efficiency and speed.

In the last part of this thesis, we summarize a few techniques to reduce noise coupling between analog and digital circuits when designing the pad ring and the printed circuit board. The testing setup is also presented to show how to obtain correct and accurate measurement results. Two channels are time-interleaved. Our design generates 16 raw output bits for a 12-bit effective resolution. The calibration is done off chip. The estimated calibration power is included in the total power consumption. The prototype ADC is fabricated in standard 1P9M 65nm LP CMOS with 1.2V supply. The active die area is 0.083mm². The implementation allows full input swing ($2.4V_{p-p}$) because of the intentional dummy capacitor. The DAC is implemented with standard MOM cap with a total capacitance of 1.60pF. A 67.4dB SNDR, 78.1dB SFDR, +1.0/-0.9LSB₁₂ INL and +0.5/-0.7LSB₁₂ DNL are achieved at 50MS/s with 24.7MHz input frequency. Total power consumption including the estimated calibration and reference power is 2.1mW for 21.9fJ/conv.-step *FoM*.

7.2 Future Work

Although this prototype SAR ADC is able to achieve the best *FoM* for any ADC reported to date that has higher than 10MS/s sampling rate with more than 10b ENOB, there are still many opportunities for improvement. The main goal is to push the SAR architecture to achieve higher resolution with higher sampling rates,

while still being able to achieve low power consumption. Using the same calibration technique and design principles, we are able to demonstrate, in simulation, that the calibration algorithm can calibrate for ADCs with resolution between 12b and 16b. As a result, it is theoretically possible to apply the same principles introduced in this thesis to obtain an ADC with much higher ENOB.

The serial operation of a SAR ADC limits the achievable sampling rates compared to other architectures (such as pipelined and flash ADCs) that use more parallel architectures. To further improve the sampling rate of a SAR architecture, there are two options. One option is to combine the SAR architecture with another architecture to create a hybrid design. The combinations of different ADC architectures could help leverage their strengths together to create a design with higher performance and better energy efficiency. Another option is to time-interleave an array of SAR ADCs to take advantage of parallelism to achieve higher sampling rate. Such designs usually experience a high level of spurious tones due to timing, offset and gain mismatch between different channels. Simple and automatic solutions need to be researched and developed to resolve these issues.

A latch comparator is an important building block, especially for design in deeply scaled CMOS technology with limited supply headroom. The goal of many such designs is to replace the traditional analog building blocks, such as an operational amplifier, with a digital comparator. Since the latch comparator has internal positive feedback and cannot be linearized easily like an operational amplifier, it is difficult to develop a closed-form solution to ease the design and simulation process. Therefore, it is important to be able to come up with a new design approach to quickly identify various performance parameters, such as bandwidth and noise, of a latch comparator. Moreover, the large kickback noise is still an important issue for the comparator. Adding a preamplifier can slow down the comparator speed significantly and it also requires an additional analog biasing circuit. One example of such new comparator design is proposed by Miyahara *et al.* in [56]. This new comparator design is able to minimize kickback noise without requiring a preamplifier. It also consumes no DC power. More efforts are still needed to explore new topologies for latch comparators.

Bibliography

- [1] B. Murmann, “ADC Performance Survey 1997-2013,” <http://www.stanford.edu/~murmman/adcsurvey.html>.
- [2] R. Schreier and G. C. Temes, *Understanding Delta-Sigma Data Converters*. New York: John Wiley & Sons, 1 ed., 2005.
- [3] R. Walden, “Analog-to-Digital Converter Survey and Analysis,” *IEEE Journal on Selected Areas in Communications*, vol. 17, pp. 539–550, Apr. 1999.
- [4] R. E. Suarez, P. Gray, and D. Hodges, “All-MOS Charge-Redistribution Analog-to-Digital Conversion Techniques. II,” *IEEE Journal of Solid-State Circuits*, vol. 10, no. 6, pp. 379–385, 1975.
- [5] M. Pelgrom, A. C. J. Duinmaijer, and A. Welbers, “Matching Properties of MOS Transistors,” *IEEE Journal of Solid-State Circuits*, vol. 24, no. 5, pp. 1433–1439, 1989.
- [6] J. McCreary and P. Gray, “All-MOS Charge Redistribution Analog-to-Digital Conversion Techniques. I,” *IEEE Journal of Solid-State Circuits*, vol. 10, no. 6, pp. 371–379, 1975.
- [7] B. Jonsson, “A Survey of A/D-Converter Performance Evolution,” in *2010 17th IEEE International Conference on Electronics, Circuits, and Systems (ICECS)*, pp. 766–769, Dec. 2010.
- [8] H.-S. Lee and C. Sodini, “Analog-to-Digital Converters: Digitizing the Analog World,” *Proceedings of the IEEE*, vol. 96, pp. 323–334, Feb. 2008.
- [9] A. Hastings, *The Art of Analog Layout*. Upper Saddle River, New Jersey: Pearson Education, Inc., 2 ed., 2005.
- [10] S. Chou, “Integration and Innovation in the Nanoelectronics Era,” in *IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 36–41, 2005.
- [11] V. Srinivasan, V. Wang, P. Satarzadeh, B. Haroun, and M. Corsi, “A 20mW 61dB SNDR (60MHz BW) 1b 3rd-Order Continuous-Time Delta-Sigma Modulator Clocked at 6GHz in 45nm CMOS,” in *IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 158–160, 2012.

- [12] P. Shettigar and S. Pavan, "A 15mW 3.6GS/s CT- $\Sigma\Delta$ ADC with 36MHz Bandwidth and 83dB DR in 90nm CMOS," in *IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 156–158, 2012.
- [13] B. Murmann, P. Nikaeen, D. Connelly, and R. Dutton, "Impact of Scaling on Analog Performance and Associated Modeling Needs," *IEEE Transactions on Electron Devices*, vol. 53, no. 9, pp. 2160–2167, 2006.
- [14] S. Thompson, P. Packan, and M. Bohr, "MOS Scaling: Transistor Challenges for the 21st Century," *Intel Technology Journal Q398*, pp. 1–19, 1998.
- [15] J. Fiorenza, T. Sepke, P. Holloway, C. Sodini, and H.-S. Lee, "Comparator-Based Switched-Capacitor Circuits for Scaled CMOS Technologies," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 12, pp. 2658–2668, 2006.
- [16] L. Brooks and H.-S. Lee, "A Zero-Crossing-Based 8-bit 200 MS/s Pipelined ADC," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 12, pp. 2677–2687, 2007.
- [17] S.-K. Shin, Y.-S. You, S.-H. Lee, K.-H. Moon, J.-W. Kim, L. Brooks, and H.-S. Lee, "A Fully-Differential Zero-Crossing-Based 1.2V 10b 26MS/s Pipelined ADC in 65nm CMOS," in *IEEE Symposium on VLSI Circuits*, pp. 218–219, 2008.
- [18] L. Brooks and H.-S. Lee, "A 12b, 50 MS/s, Fully Differential Zero-Crossing Based Pipelined ADC," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 12, pp. 3329–3343, 2009.
- [19] T. Ogawa, H. Kobayashi, M. Hotta, Y. Takahashi, H. San, and N. Takai, "SAR ADC Algorithm with Redundancy," in *IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, pp. 268–271, Nov. 2008.
- [20] J. Doernberg, H.-S. Lee, and D. Hodges, "Full-Speed Testing of A/D Converters," *IEEE Journal of Solid-State Circuits*, vol. 19, pp. 820–827, Dec. 1984.
- [21] H.-S. Lee, D. Hodges, and P. Gray, "A Self-Calibrating 15 Bit CMOS A/D Converter," *IEEE Journal of Solid-State Circuits*, vol. 19, pp. 813–819, Dec. 1984.
- [22] L. Brooks and H.-S. Lee, "Background Calibration of Pipelined ADCs Via Decision Boundary Gap Estimation," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 55, pp. 2969–2979, Nov. 2008.
- [23] P. Li, M. Chin, P. Gray, and R. Castello, "A Ratio-Independent Algorithmic Analog-to-Digital Conversion Technique," *IEEE Journal of Solid-State Circuits*, vol. 19, pp. 828–836, Dec. 1984.
- [24] B.-S. Song, M. Tompsett, and K. Lakshmikummar, "A 12-Bit 1-MSample/s Capacitor Error-Averaging Pipelined A/D Converter," *IEEE Journal of Solid-State Circuits*, vol. 23, pp. 1324–1333, Dec. 1988.

- [25] C.-C. Shih and P. Gray, "Reference Refreshing Cyclic Analog-to-Digital and Digital-to-Analog Converters," *IEEE Journal of Solid-State Circuits*, vol. 21, pp. 544–554, Aug. 1986.
- [26] S. Sutarja and P. Gray, "A Pipelined 13-Bit 250-KS/s 5-V Analog-to-Digital Converter," *IEEE Journal of Solid-State Circuits*, vol. 23, pp. 1316–1323, Dec. 1988.
- [27] S.-Y. Chin and C.-Y. Wu, "A CMOS Ratio-Independent and Gain-Insensitive Algorithmic Analog-to-Digital Converter," *IEEE Journal of Solid-State Circuits*, vol. 31, pp. 1201–1207, Aug. 1996.
- [28] Y. Chiu, "Inherently Linear Capacitor Error-Averaging Techniques for Pipelined A/D Conversion," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 47, pp. 229–232, Mar. 2000.
- [29] L. Jin, D. Chen, and R. Geiger, "A Digital Self-Calibration Algorithm for ADCs Based on Histogram Test Using Low-Linearity Input Signals," in *IEEE International Symposium on Circuits and Systems, 2005. ISCAS 2005*, pp. 1378–1381, May 2005.
- [30] J. Elbornsson and J. E. Eklund, "Histogram Based Correction of Matching Errors in Subranged ADC," in *Proceedings of the 27th European Solid-State Circuits Conference, 2001. ESSCIRC 2001*, pp. 555–558, Sept. 2001.
- [31] E. Siragusa and I. Galton, "Gain Error Correction Technique for Pipelined Analogue-to-Digital Converters," *Electronics Letters*, vol. 36, pp. 617–618, Mar. 2000.
- [32] J. Bjornsen, O. Moldsvor, T. Saether, and T. Ytterdal, "A 220mW 14b 40MSPS Gain Calibrated Pipelined ADC," in *Proceedings of the 31st European Solid-State Circuits Conference, 2005. ESSCIRC 2005*, pp. 165–168, Sept. 2005.
- [33] B. Murmann and B. Boser, "A 12-bit 75-MS/s Pipelined ADC using Open-Loop Residue Amplification," *IEEE Journal of Solid-State Circuits*, vol. 38, pp. 2040–2050, Dec. 2003.
- [34] J. Li and U.-K. Moon, "Background Calibration Techniques for Multistage Pipelined ADCs with Digital Redundancy," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 50, pp. 531–538, Sept. 2003.
- [35] W. Liu, P. Huang, and Y. Chiu, "A 12b 22.5/45MS/s 3.0mW 0.059mm² CMOS SAR ADC Achieving over 90dB SFDR," in *IEEE International on Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 380–381, Feb. 2010.
- [36] S. Sonkusale, J. Van der Spiegel, and K. Nagaraj, "True Background Calibration Technique for Pipelined ADC," *Electronics Letters*, vol. 36, pp. 786–788, Apr. 2000.

- [37] Y. Chiu, C. Tsang, B. Nikolic, and P. Gray, "Least Mean Square Adaptive Digital Background Calibration of Pipelined Analog-to-Digital Converters," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 51, pp. 38–46, Jan. 2004.
- [38] W. Liu, Y. Chang, S.-K. Hsien, B.-W. Chen, Y.-P. Lee, W.-T. Chen, T.-Y. Yang, G.-K. Ma, and Y. Chiu, "A 600MS/s 30mW 0.13 μ m CMOS ADC Array Achieving over 60dB SFDR with Adaptive Digital Equalization," in *IEEE International Conference on Solid-State Circuits (ISSCC)*, pp. 82–83, 2009.
- [39] R. Suarez, P. Gray, and D. Hodges, "An All-MOS Charge-Redistribution A/D Conversion Technique," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, vol. XVII, pp. 194–195, Feb. 1974.
- [40] B. Ginsburg and A. Chandrakasan, "An Energy-Efficient Charge Recycling Approach for a SAR Converter with Capacitive DAC," in *IEEE International Symposium on Circuits and Systems*, vol. 1, pp. 184–187, May 2005.
- [41] Y.-K. Chang, C.-S. Wang, and C.-K. Wang, "A 8-bit 500-KS/s Low Power SAR ADC for Bio-Medical Applications," in *IEEE Asian Solid-State Circuits Conference (ASSCC)*, pp. 228–231, Nov. 2007.
- [42] C.-C. Liu, S.-J. Chang, G.-Y. Huang, and Y.-Z. Lin, "A 0.92mW 10-bit 50-MS/s SAR ADC in 0.13 μ m CMOS process," in *Symposium on VLSI Circuits*, pp. 236–237, June 2009.
- [43] V. Hariprasath, J. Guerber, S.-H. Lee, and U.-K. Moon, "Merged Capacitor Switching Based SAR ADC with Highest Switching Energy-Efficiency," *Electronics Letters*, vol. 46, pp. 620–621, April 2010.
- [44] Y. Chen, X. Zhu, H. Tamura, M. Kibune, Y. Tomita, T. Hamada, M. Yoshioka, K. Ishikawa, T. Takayama, J. Ogawa, S. Tsukamoto, and T. Kuroda, "Split Capacitor DAC Mismatch Calibration in Successive Approximation ADC," in *IEEE Custom Integrated Circuits Conference*, pp. 279–282, Sept. 2009.
- [45] A. Agnes, E. Bonizzoni, P. Malcovati, and F. Maloberti, "A 9.4-ENOB 1V 3.8 μ W 100kS/s SAR ADC with Time-Domain Comparator," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 246–247, Feb. 2008.
- [46] M. Yoshioka, K. Ishikawa, T. Takayama, and S. Tsukamoto, "A 10b 50MS/s 820 μ W SAR ADC with On-Chip Digital Calibration," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 384–385, Feb. 2010.
- [47] V. Giannini, P. Nuzzo, V. Chironi, A. Baschiroto, G. Van der Plas, and J. Craninckx, "An 820 μ W 9b 40MS/s Noise-Tolerant Dynamic-SAR ADC in 90nm Digital CMOS," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 238–239, Feb. 2008.

- [48] F. Kuttner, "A 1.2V 10b 20MSample/s Non-Binary Successive Approximation ADC in 0.13 μ m CMOS," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, vol. 1, pp. 176–177, 2002.
- [49] M. Hesener, T. Eichler, A. Hanneberg, D. Herbison, F. Kuttner, and H. Wenske, "A 14b 40MS/s Redundant SAR ADC with 480MHz Clock in 0.13 μ m CMOS," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 248–249, Feb. 2007.
- [50] J. Yang, T. Naing, and B. Brodersen, "A 1-GS/s 6-bit 6.7-mW ADC in 65-nm CMOS," in *IEEE Custom Integrated Circuits Conference (CICC)*, pp. 287–290, Sept. 2009.
- [51] S.-W. Chen and R. Brodersen, "A 6b 600MS/s 5.3mW Asynchronous ADC in 0.13 μ m CMOS," in *IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 2350–2352, 2006.
- [52] A. Abo and P. Gray, "A 1.5-V, 10-bit, 14.3-MS/s CMOS Pipeline Analog-to-Digital Converter," *IEEE Journal of Solid-State Circuits*, vol. 34, pp. 599–606, May 1999.
- [53] J. Montanaro, R. Witek, K. Anne, A. Black, E. Cooper, D. Dobberpuhl, P. Donahue, J. Eno, W. Hoepfner, D. Kruckemyer, T. Lee, P. Lin, L. Madden, D. Murray, M. Pearce, S. Santhanam, K. Snyder, R. Stehpany, and S. Thierauf, "A 160-MHz, 32-b, 0.5-W CMOS RISC Microprocessor," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 11, pp. 1703–1714, 1996.
- [54] P. Nuzzo, F. De Bernardinis, P. Terreni, and G. Van der Plas, "Noise Analysis of Regenerative Comparators for Reconfigurable ADC Architectures," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 55, no. 6, pp. 1441–1454, 2008.
- [55] G. V. der Plas and J. Craninckx, "A 65fJ/Conversion-Step, 0-50MS/s 0-0.7 mW 9b Charge-Sharing SAR ADC in 90nm Digital CMOS," in *IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 246–247, 2007.
- [56] M. Miyahara, Y. Asada, D. Paik, and A. Matsuzawa, "A Low-Noise Self-Calibrating Dynamic Comparator for High-Speed ADCs," in *IEEE Asian Solid-State Circuits Conference 2008*, pp. 269–272, 2008.