Energy Efficient SAR ADC with Resolution Enhancement for Sensor Signals

by

Harneet Singh Khurana
B.S., University of Maryland (2007)
S.M., Massachusetts Institute of Technology (2010)

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

February 2021

Author .....
Department of Electrical Engineering and Computer Science
January 7, 2021

Certified by .....
Anantha P. Chandrakasan
Vannevar Bush Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Certified by .....
Hae-Seung Lee
ATSP Professor of Electrical Engineering
Thesis Supervisor

Accepted by.....
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Energy Efficient SAR ADC with Resolution Enhancement for Sensor Signals

by

Harneet Singh Khurana

B.S., University of Maryland (2007)
S.M., Massachusetts Institute of Technology (2010)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Abstract

Many signals from sensors are low activity signals that spend most of its time around middle of the full scale with occasional large activity. A/D conversion of such signals using a conventional ADC with a constant resolution and a full-scale search space consumes unnecessary amounts of time and energy. SAR ADC architecture using a comparator and Capacitor-DAC has been the choice for this application space due to minimal analog components and low static power consumption while providing moderate speed and resolution that is adequate for sensor signals. DAC and comparator power reduction has been the focus of attention as the logic automatically benefits from digital centric process scaling. This work develops an energy efficient 10B/12B SAR ADC for such sensor signals using a new algorithm to save energy and time and use the savings for resolution enhancement.

Thesis Supervisor: Anantha P. Chandrakasan
Title: Vannevar Bush Professor of Electrical Engineering and Computer Science

Thesis Supervisor: Hae-Seung Lee
Title: ATSP Professor of Electrical Engineering

Acknowledgements

Firstly, I would like to thank my advisors, Prof. Anantha P. Chandrakasan and Prof. Hae-Seung Lee, for their infinite support and guidance throughout this journey. I regard their wisdom and admire them as successful experts in the field of Electrical Engineering. They will always be the source of inspiration to me for the rest of my life. Thank you both so much.

I am extremely fortunate to be part of the two research groups and everyone at MIT that I got to know, talk, and learn from during my time here. I have lots of memories with everyone I came across and will cherish them forever going forward.

I would like to thank my thesis committee member Prof. Ruonan Han for his valuable feedback.

I am grateful to be part of MIT and everything that makes it such a wonderful place. I am fortunate to have come here and know of the endless possibilities here and outside going forward.

Thanks to my dad for teaching me to keep working hard. Thanks to my mom for teaching me to keep learning. Her memories have encouraged and showed me the way all along.

I am very thankful to my wife and kids for their lots of love, patience and their time in hours, days, and years that they gave to me and for making my dream as their own. Love you lots.

Contents

1 Introduction 14

1.1 Capacitive DAC energy 19

1.2 Comparator energy 21

1.3 Digital energy with scaling process 26

1.4 Thesis organization 27

2 Background 29

2.1 Conventional SAR (MSB-first) 30

2.2 LSB-first 32

3 Recode-then-LSB-first (RLSB-first) algorithm 37

3.1 Split-DAC 39

3.2 RLSB-first algorithm 41

3.3 Energy calculation 46

3.4 Comparison 49

3.5 RLSB-first benefits 59

3.6 System architecture 59

4 Resolution enhancement 63

4.1 Steps for resolution enhancement 63

4.2 Computation of the extra 2B 66

4.3 Accuracy benefits of two-step 67

5 Noise measurement 72

5.1 Comparator noise and output average relations 72
5.2 Steps for noise measurement 74
6 Split-DAC calibration 79
6.1 Error voltage definition and relations 80
6.2 Full-bit residual voltage acquisition steps 84
6.3 Half-bit residual voltage acquisition steps 86
6.4 Calibration loop and error voltage calculation 87
7 Design details 91
7.1 Recoding algorithm 91
7.2 Decode module 92
7.3 Encode module 97
7.4 Split-DAC layout 103
7.5 Comparator 108
7.6 Bootstrapping circuit 113
7.7 Voltage and device choices 114
8 Measurements 117
9 Conclusions 125
10 Bibliography 129

List of Figures

1-1. Sensor system block diagram 14

1-2. Analog front end (AFE) block diagram 15

1-3. Small signal model for latch 22

2-1. Conventional SAR A/D converter 30

2-2. LSB-first block diagram 32

2-3. LSB-first algorithm 33

2-4. Example comparison of bit-cycles and DAC switching energy for LSB-first and conventional MSB-first A/D conversion 34

2-5. Example comparison of bit-cycles and DAC switching energy for LSB-first and conventional MSB-first A/D conversion 35

3-1. RLSB-first block diagram 38

3-2. Conventional DAC (top) and split-DAC (bottom) 39

3-3. Switch configurations for a bit in a Conventional DAC (top) and split-DAC (bottom) 40

3-4. RLSB-first algorithm 41

3-5. Encoding rules shown with an example 44

3-6. Decoding rules shown with an example 45

3-7. Worst energy (left) and worst bit-cycles (right) vs. code change 51

3-8. Average energy vs. input signal amplitude at 0.49Fs (left) and 0.10Fs (right) 54

3-9. Average energy vs. input signal amplitude at 0.49Fs (left) and 0.10Fs (right) 54
3-10. Energy and bit-cycles for ECG signal 55
3-11. Energy and bit-cycles for bearing vibration signal 56
3-12. Energy and bit-cycles for EMG signal 57
3-13. Energy and bit-cycles for EMG signal 58
3-14. System architecture block diagram 60
3-15. Phase diagram and module schedule conditioned on bit-cycles used during 10B RLSB-first 61
4-1. Initial state and preparation (STEP 0) after 10B RLSB-first conversion ends 64
4-2. Coarse estimate (STEP 1) and fine estimate (STEP 2) of resolution enhancement 65
4-3. Comparison of accuracy versus rms noise of two step and one step 70
5-1. Iterations to measure Y_N using negative reference 74
5-2. Iterations to measure Y_P using positive reference 76
6-1. Pre-charge and charge distribution steps for one iteration of full bit calibration 84
6-2. Pre-charge and charge distribution steps for one iteration of half bit calibration 86
7-1. Decode module logic 94
7-2. Dependencies of decode module 96
7-3. Decode example 96

7-4. Decode modules connected together 96
7-5. Encoding module 99
7-6. Dependencies of Encode module 101
7-7. Encode example 101
7-8. Encode modules connected together 102
7-9. Split-DAC unit capacitor layout 103
7-10. Split-DAC layout of single side. Replicated for the differential pair ... 107
7-11. Comparator 108
7-12. Simulation of dynamic comparator 109
7-13. Bootstrapping circuit 113
7-14. Simulation with starting code 716 and input sample of 10mV. Resultant output code is 522 114
8-1. Test setup 117
8-2. DNL and INL without calibration 118
8-3. DNL and INL with calibration and before offset correction 119
8-4. DNL and INL with calibration and after offset correction 120
8-5. FFT for 10B quantization at 1.9KHz and 100Hz sinusoidal input 121
8-6. FFT for relative comparison of FFT with and without resolution enhancement. Notice noise floor moves down with resolution enhancement 122
8-7. Die picture 123
9-1. Energy (pJ) versus SNDR (dB) for ADCs 127

List of Tables

1-1. Example of bio-signals with their voltage and frequency range 16

1-2. Example of AFEs 17

1-3. Example of SAR A/D energy saving techniques 18

1-4. Dennard constant field scaling factors 26

3-1. Summary of recoding rules 43

3-2. LSB-first example conversion 50

3-3. RLSB-first example conversion 50

7-1. Truth table for E[n] and T[n] 92

7-2. Truth table for P[n] and D[n] 94

7-3. Truth table for C[n] 98

7-4. Truth table for R[n] 98

7-5. Truth table for S[n] 98

7-6. Truth table for E[n] 100

8-1. Comparison with other works 123

Chapter 1

Introduction

With the advent of microelectronics and the accompanying small form factor, low power sensor signal acquisition systems have become ubiquitous. These sensor systems find use in industrial, medical, and consumer settings. Sensor systems for patient health monitoring could be deployed externally or be implanted.

Sensor systems consists of a transducer (sensor), analog-front-end (AFE), digital signal processor (DSP), and communication blocks in the signal chain as shown in Figure 1-1. These are powered through a battery or an energy harvester and their power is

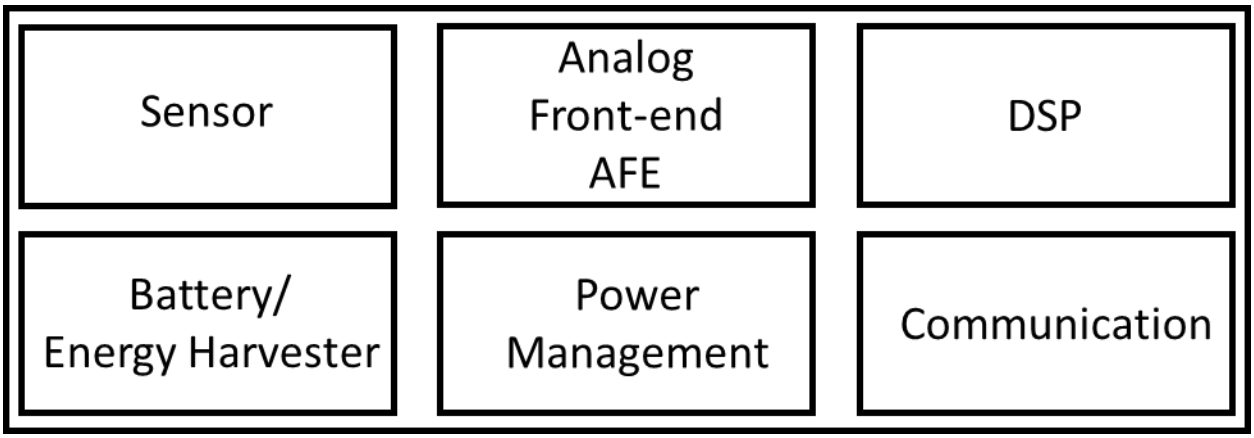


Figure 1-1. Sensor system block diagram.

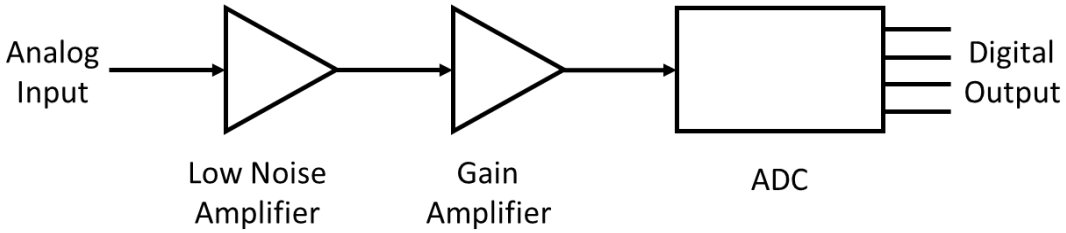


Figure 1-2. Analog front-end (AFE) block diagram.

managed through a power management block. Transducers are sensitive to nonelectrical physical or chemical quantities by converting one form of energy into electrical. Once the signal is in electrical domain it is conditioned through circuits in the analog-front-end (AFE) block through amplification and band-limiting before converting it from analog to digital domain through analog-to-digital (A/D) converter. After A/D conversion the digital information is usually processed through data compression and filtered in the DSP block before it is output through a radio in the communication block.

Due to small sensor signal amplitudes, analog-front-ends (AFE) are a critical part of any sensor system as it determines the quality of signal acquired. The amplitude and frequency range of common signals like ECG, EEG, and EMG are listed in Table 1-1 [6]. AFE comprises of a low noise amplifier (LNA) and gain amplifier, followed by an A/D converter as shown in Figure 1-2. LNA is designed to have low input referred noise below the noise level at the input for negligible degradation of the input signal to noise ratio (SNR) while providing a modest gain. The gain amplifier is designed to provide most of the gain. The noise contributed by this amplifier is made insignificant when referred to the

Table 1-1. Example of bio-signals with their voltage and frequency range [6].

Signal Name	**Voltage range (mV)**	**Frequency range (Hz)**
ECG (Electrocardiogram)	0.1-3.0	0.1-200
EEG (Electroencephalography)	0.01-1	0.2-100
EMG (Electromyography)	0.01-100	50-2000

input due to large gain. An ADC samples the analog signal and converts it into digital signal to enable digital storage and extraction of useful information. An ADC should introduce minimal circuit noise to avoid degradation of SNR at its input. Examples of AFE include [4], [5], [6], [7], [8] and listed in Table 1-2. A low-voltage and ultra-low power sensor interface for electromyogram (EMG) [4] consists of an amplifier and SAR ADC powered by a 0.3V supply that consumes only 3.8nW. An implantable 64nW ECG-monitoring mixed-signal SoC for arrhythmia diagnosis [5] is a syringe-implantable ECG recording device that is less susceptible to 60Hz noise sources. A fully integrated and programmable biomedical sensor interface chip in [6] [7] is powered with a 1V and consumes 450nW. A very low-power CMOS mixed signal IC for implantable pacemaker in [8] runs on a single battery charge for 10 to 12 years.

Sensor systems typically have low power budget in the range of sub-micro-watts range as they run from a small form battery or harvested energy [2], [3], and have to stay in service for a long time. To get an estimate available energy and power, assume a sensor uses a $1\mathrm{mm}^3$ size Lithium ion battery which has specific energy density of about 200Whr/Kg and volumetric energy density of about 300Whr/L. This would give us about

Table 1-2. Example of AFE systems.

System	Specifications	Power
EMG sensor interface, [4]	0.3V, 40dB gain, 8B A/D 20-425Hz bandwidth	3.8nW
ECG monitoring system, [5]	0.6V, 51-96dB gain, 8B A/D 250Hz bandwidth, includes DSP	64nW
Sensor interface, [6][7]	1V, 45.6-60dB gain, 12B A/D 292Hz bandwidth	450nW
Pacemaker, [8]	8B A/D 1KS/s, 12B A/D 100S/s 12yrs battery lifetime	8uW

1J of energy to use. For 30 days of uninterrupted operation it means we have about 425nW to power our sensor node. Frequent battery change is either not an option due to associated complications such as in the case of bio-implants or infeasible due to a large number of such sensors deployed in an industrial monitoring network. A continuously-on radio in a sensor system can dominate the power of the whole sensor system. The radio power is significantly reduced through duty cycled operation by minimization of the data to be transmitted [1]. Data reduction can be accomplished through feature extraction in the analog or digital domain. It makes amplifier and ADC dominate power expenditure in such applications [1]. This work focusses on contributing to lowering energy in the ADC part of the front-end for sensor applications.

Many signals from sensors are low activity signals that spend most of its time around middle of the full scale with occasional large activity. A/D conversion of such signals using a conventional ADC with a constant resolution and a full-scale search space consumes unnecessary amounts of time and energy. Capacitive DAC based SAR ADC has been the choice for low power and medium resolution A/D conversion. Its main

Table 1-3. Example of SAR A/D energy saving techniques.

System	Specifications	Technique
Resolution-reconfigurable SAR ADC, [18]	5-10B DAC, 0.4-1V 206nW at 10B	Resolution reconfigurable DAC, voltage scaling
Adaptive resolution ADC for neural sensor, [19]	3-8B DAC, 100KS/s 390nW at 8B	Resolution reconfigurable DAC
SAR ADC with bypass window, [20]	10B, 200KS/s 1uW	Threshold setting using reference, comparator
SAR ADC using LSB-first, [16] [17]	10B, 4KS/s 8.7-31nW	Data-driven, LSB-first algorithm

components include capacitive digital-to-analog converter (DAC), comparator, and logic. DAC and comparator power reduction has been the focus of attention as the logic automatically benefits from digital-centric process scaling as discussed below. Previous solutions to reduce power include resolution reconfigurable DACs [18], [19], bypassing bit-cycles when signal falls in a range [20], and signal-activity-based power saving algorithm [16], [17] and listed in Table 1-3. The solution in [18] scales power with resolution and bandwidth requirements of an application but it doesn't adapt to changes in signal activity. In [19] the resolution is changed through a periodic calibration routine that sets the DAC resolution according to data content of the signal. However, conversion steps are identical for each sample and do not benefit from small changes in consecutive samples. The solution in [20] saves energy when signal falls in a preset signal range and requires a separate reference and comparators to set threshold adding analog complexity. The solution in [16], [17] scales energy of ADC with signal activity on an average. It is a prediction-based A/D converter that starts its search from a prediction code and finishes A/D conversion in a fewer steps. However certain input sample

sequences degrade performance and energy savings and is discussed in detail in the following chapter.

This thesis focusses on contributing to the design of low power prediction-based SAR ADC for sensor signals that scales energy consumption of analog components with signal activity by leveraging the low power logical operations in scaling digital processes. Since low energy is of prime concern in sensor signal applications, an estimate of the least energy numbers in key ADC components is an important factor. In prediction-based SAR ADC, ideally prediction should always be correct, and lead to least energy spent. The least energy should be only that which is spent to confirm that the prediction made is correct. For sensor signals, that are close to zero amplitude most of the time, a prediction of corresponding mid-code is most likely and could be used for least DAC energy calculation. Also, at-least a couple of comparator decisions are needed to confirm that this correct prediction is bounded within an LSB. These energy numbers are calculated below to get an idea of ideal low energy numbers involved.

1.1 Capacitive DAC energy

Prediction-based SAR A/D converters [16], [17], [37] start by first loading a prediction code and then searching around for the current sample code. Capacitive DAC energy in such SAR A/D converters should ideally be only that needed to confirm a correct prediction made in the digital domain. Minimum DAC energy needed should only be that needed to initialize this correct prediction code and no more spend in switching DAC and

searching around. Since the application is for sensor signals that spend most of the time around mid-rail, a prediction code of mid-rail can be used for estimation of this lowest possible energy. Let us calculate energy required to initialize a mid-code prediction on a differential capacitive DAC with a capacitance of C_{dac} on each side of a differential DAC. To load this value, half of the capacitance on each DAC flips to effectively charge two capacitances of size $C_{dac}/2$ connected in series. The series combination is equivalent to $C_{dac}/4$. For simplicity assume reference and supply voltage are the same and equal to V. The energy used is

$$E_{dac} = 2\frac{C_{dac}}{4}V^2 = \frac{C_{dac}}{2}V^2$$

Let us use a differential sinusoidal input signal of amplitude V as a test signal and sample it onto a differential capacitive DAC. If the mismatch requirements for the required SNR are met in the chosen technology or calibration is employed, the total DAC capacitance 2C is dictated by thermal sampling noise. The sampled mean square noise on each side of the differential DAC is kT/C and the SNR is

$$SNR = \frac{\left(\frac{V}{\sqrt{2}}\right)^2}{2\frac{kT}{C_{dac}}}$$

Manipulating and substituting C_{dac} in equation for E_{dac}, we get

$$E_{dac} = 2kT.SNR$$

For a 12B resolution, the calculated lowest DAC switching energy is approximately 0.21pJ.

1.2 Comparator energy

Dynamic comparator energy is estimated for an N bit SAR ADC as in [11] adapted to our case of two comparisons needed to confirm that the prediction code is correct. Simplifying assumptions are made as in [11] so that the estimate is independent of comparator circuit topology. A dynamic comparator is modelled with a latching output and a capacitive load of C_{latch} at each output. For a correct prediction, after the DAC is initialized with the code, the input to the comparator V_i is a residual voltage that can be modelled as uniformly distributed over an LSB size interval. Dynamic comparator has two phases called reset and amplification. In the reset phase the output nodes are charged up to the supply voltage and in the amplification phase positive feedback is used to latch output starting from an initial input dependent differential voltage. The total energy is therefore

$$E_{comp} = E_{latch} + E_{reset}$$

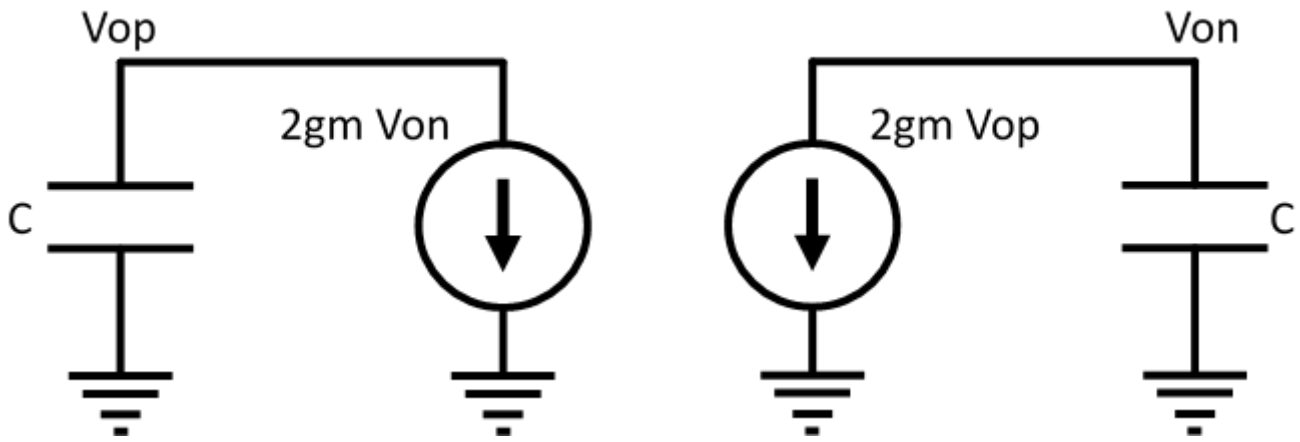


Figure 1-3. Small signal model of the latch.

At the beginning of the reset phase, it is assumed that one of output capacitor is fully charged to the supply voltage V from previous decision. Therefore, the supply is used to charge only one output capacitor and the corresponding reset energy is

$$E_{reset} = C_{latch}V^2 \quad (1)$$

During the amplification phase the energy supplied during the latching can be estimated with peak current I_{max} and regenerative time t as

$$E_{latch} \leq I_{max}Vt \quad (2)$$

The feedback gain is maximum when V_{gs} is around V/2. At this bias maximum current I_{max} is sourced from the supply, through the transistors to ground. Considering low power, low voltage operation and transistors operating in subthreshold we get

$$I_{max} = g_m \frac{nkT}{q}$$

In the above equation, n is the sum of oxide capacitance and depletion capacitance divided by the oxide capacitance of a mosfet. The small signal model of the circuit during the amplification phase can be described with the following coupled differential equations at the output nodes V_{on} and V_{op} as in [11], [42] and shown in Figure 1-3

$$\frac{dV_{op}}{dt} + \frac{2g_m}{C_{latch}} V_{on} = 0$$

$$\frac{dV_{on}}{dt} + \frac{2g_m}{C_{latch}} V_{op} = 0$$

Let us make a simplifying approximation of initial voltage at t=0 on latch output to be the same as the voltage at the comparator input V_i. Solving as in [42] for the time it takes for regeneration from this initial V_i to the supply voltage V we get

$$V_o = V_{op} - V_{on} = V_i e^{\frac{2g_m}{C_{latch}} t}$$

$$t = \frac{C_{latch}}{2g_m} \ln\left[\frac{V}{V_i}\right]$$

Substituting this result for t in (2) we get

$$E_{latch} = \frac{nkT}{2q} C_{latch} V \ln\left[\frac{V}{V_i}\right]$$

Averaging over uniformly distributed inputs V_i over an LSB range results in

$$E_{latch} = \frac{nkT}{2q} C_{latch}\, 2^{N-1} \int_0^{\frac{V}{2^{N-1}}} \ln\left[\frac{V}{V_i}\right] dV_i$$

$$E_{latch} = \frac{nkT}{2q} C_{latch} V\{(N-1).\ln 2 \; + 1\} \quad (3)$$

Input referred comparator noise has been derived to be scaled kT/C in [10]. Although less accurate it is approximated as in [11] to be a sum of two times kT/C_{latch} noise from each of the latch outputs. With this the SNR is

$$SNR = \frac{\frac{1}{2}V^2}{2\frac{kT}{C_{latch}}}$$

$$C_{latch} V^2 = 4kT.SNR \quad (4)$$

Using (4) in (1) and (3) results in

$$E_{latch} = \frac{nkT}{qV} 2kT.SNR.\{(N-1).\ln 2 \; + 1\} \quad (5)$$

$$E_{reset} = 4kT.SNR \quad (6)$$

The result for E_{latch} in (5) was obtained using a bias of $V/2$ as the latch common mode before positive feedback starts. However, the output voltage needs to discharge from reset value of V to the common mode. This initial discharge from V to V/2 can be approximated to have the same energy efficiency and use the same result in (5) for initial zero current just after reset to peak supply current of I_{max} just before positive feedback takes over [11]. Multiplying (5) by 2 to account for this initial discharge and adding it to (6) gives the net resultant total average energy for each comparator cycle

$$E_{comp} = kT.SNR.\left[4 + 4\frac{nkT}{qV}\{(N-1).\ln 2\ + 1\}\right]$$

Evaluating for V equal to 0.5V and approximating n to 1, for a 12B and 10B SNR, the calculated energy for two comparisons is approximately 1.18pJ and 0.07pJ respectively. The total ideal analog energy for a 12B conversion should be approximately

$$E_{analog} = E_{dac} + E_{comp} = 0.21pJ + 1.18pJ = \sim 1.4pJ$$

Table 1-4. Dennard constant field scaling factors [13].

Device or Circuit Parameter	**Scaling Factor**
Device dimension, t_{ox}, L, W	$1/\kappa$
Voltage, V	$1/\kappa$
Current, I	$1/\kappa$
Threshold Voltage, V_T	$1/\kappa$
Capacitance, C, $\epsilon A/t_{ox}$	$1/\kappa$
Delay time/circuit, VC/I	$1/\kappa$
Power dissipation/circuit, VI	$1/\kappa^2$
Power density, VI/A	1
Energy/operation, CV^2	$1/\kappa^3$
Energy Delay Product	$1/\kappa^4$

1.3 Digital energy with scaling process

The number of components on a single integrated circuit has increased over years in line with the prediction in [12] bringing down the cost per component. Along with the cost, the process scaling has brought down energy and power consumption per digital operation. Moore's law [12] is made possible with constant field Dennard scaling [13] where all features and voltages are scaled down by the same factor, κ. The scaling factors for important variables are listed in Table 1-2. Constant electric fields are important to avoid

silicon breakdown and electron tunneling. Scaling voltages and capacitances brings about benefits of reduced energy-delay-product (EDP). In deep-submicron CMOS process classical scaling predictions become increasingly less accurate and practical scaling methods are needed [14]. With process scaling from 65nm to 7nm there is a 17X improvement in energy per digital operation [14] while EDP scales down by over 100X. A/D architectures could benefit from designs that use application specific algorithms and using small components with large mismatch by using digital correction for low energy analog operation.

1.4 Thesis organization

Chapter 2 gives background by describing the conventional SAR search algorithm and a modified SAR search called LSB-first and its benefits. It is followed by chapter 3 which describes the new recoding algorithm called Recode-then-LSB-first (RLSB-first) and its benefits in low power A/D SAR conversion. It is followed by the proposed 10B/12B ADC architecture that uses this new algorithm for the first 10B and do 2B resolution enhancement in saved conversion time. Chapter 4 describes the technique of resolution enhancement. It is followed by chapter 5 that talks about comparator noise measurement that is needed for the resolution enhancement and chapter 6 that talks about the calibration of DAC mismatches in background. The design choices made are presented in chapter 7 followed by the measurement results in chapter 8 and conclusion in chapter 9.

Chapter 2

Background

SAR ADC architecture has been the choice for low power and medium resolution applications due to minimal analog components and low static power consumption. This chapter first describes the conventional method of SAR A/D conversion that starts from the MSB side of the DAC and evaluates the MSB first before moving on to evaluating the lower bits. It is followed by a modified way of doing SAR conversion called LSB-first where the SAR conversion begins from the LSB side instead. LSB-first has the benefit of reducing DAC switching energy and number of bit-cycles used per conversion on an average for low activity signals defined as signals with small code changes per sample over its dynamic range [16]. However, certain small code transitions still require large switching energy and long bit-cycles. This is shown later with examples that compare LSB-first and conventional SAR search.

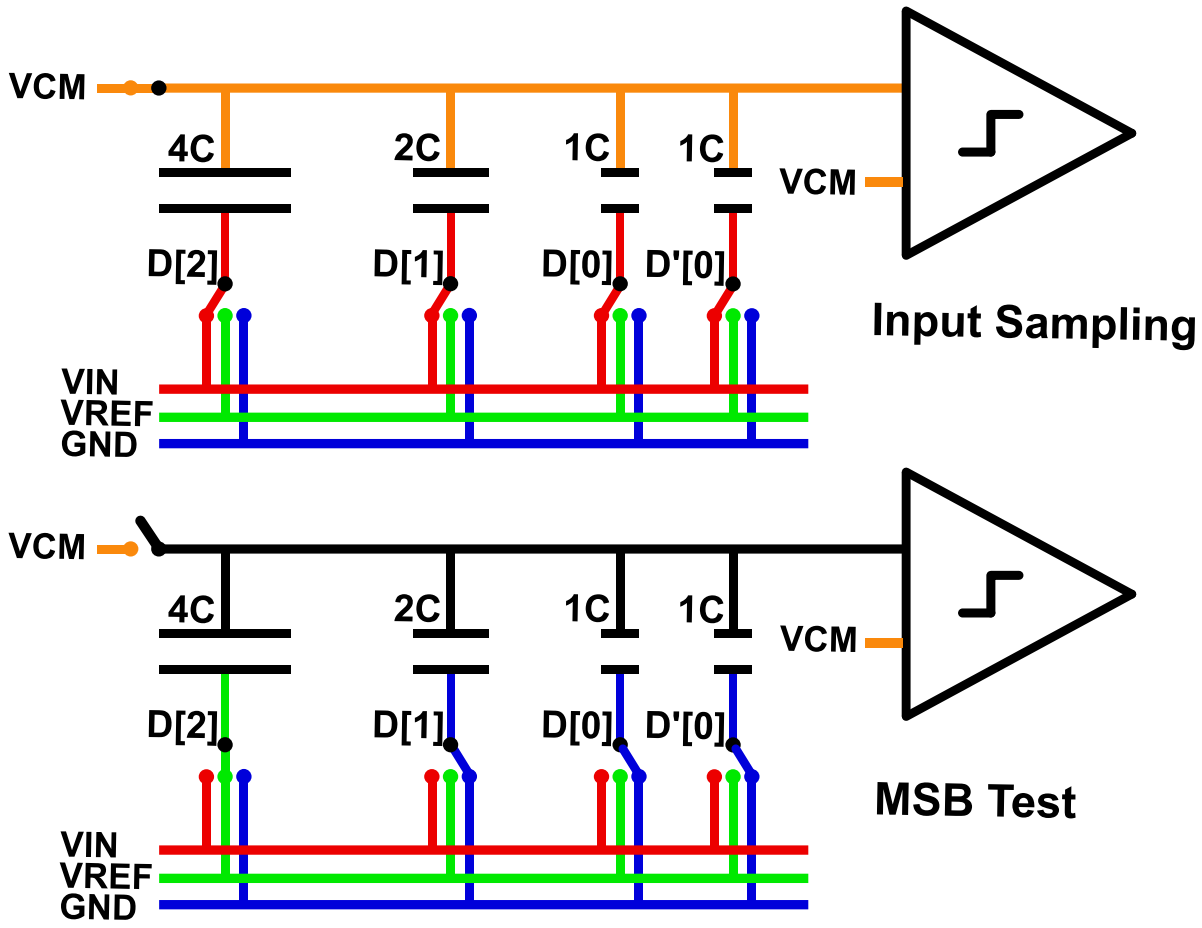


Figure 2-1. Conventional SAR A/D converter.

2.1 Conventional SAR

Conventional SAR A/D converter uses a binary search algorithm to convert an input sample into a best matching digital code [15], [36]. The most common approach is to use

a capacitive DAC to sample the input and generate binary weighted fractions of a reference voltage VREF during successive approximation.

1 bit/cycle SAR converters use N bit-cycles for a N Bit resolution and an additional cycle to sample the input. In the first step the input gets sampled as shown in the top part of Figure 2-1. The top plate of the DAC is held at a common mode voltage, VCM, and the bottom plate of all capacitors are connected to the input, VIN, being sampled. This step charges the DAC with an effective VCM-VIN voltage. After the input is sampled, the DAC is disconnected from the input source and the A/D conversion begins. As shown in bottom part of Figure 2-1, the conversion begins by connecting the bottom plate of the MSB to VREF. This effectively adds VREF/2 to the voltage seen on the top plate of the DAC. The difference between the sampled input and VREF/2 is compared by strobing the comparator. If the sampled input is larger than VREF/2, a register corresponding to MSB is set to 1. If the sampled input is smaller than VREF/2, MSB is toggled back to GND and a 0 is registered. The conversion moves and next checks the remaining residue by adding VREF/4 using the next smaller capacitance of the DAC. This continues on until all bits from MSB to LSB are evaluated and digital code is stored in their corresponding register.

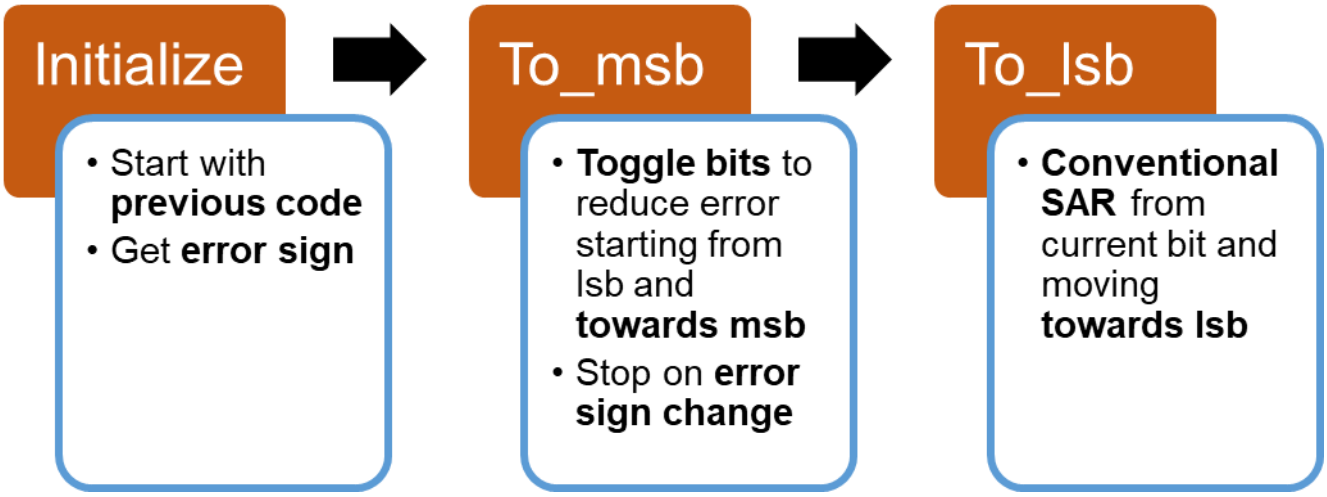


Figure 2-2. LSB-first block diagram.

2.2 LSB-first

LSB-first algorithm [17] performs a modified SAR search by starting from the LSB side of the DAC rather than the MSB side in conventional technique. LSB-first is suitable for low-activity signals as it reduces the DAC switching energy and the number of bit-cycles required per conversion. An overview of the algorithm is shown in Figure 2-2 with details of the algorithm shown in Figure 2-3 and discussed below.

LSB-first algorithm has three phases, namely Initialization, To_lsb, and To_msb. Assume a 10B A/D conversion using this algorithm on a binary weighted DAC. In the Initialization phase the input is first sampled and then the DAC bits, D[9:0], are set to the estimate of previous sample bits, DPREV[9:0]. The unused unit size capacitor in a binary

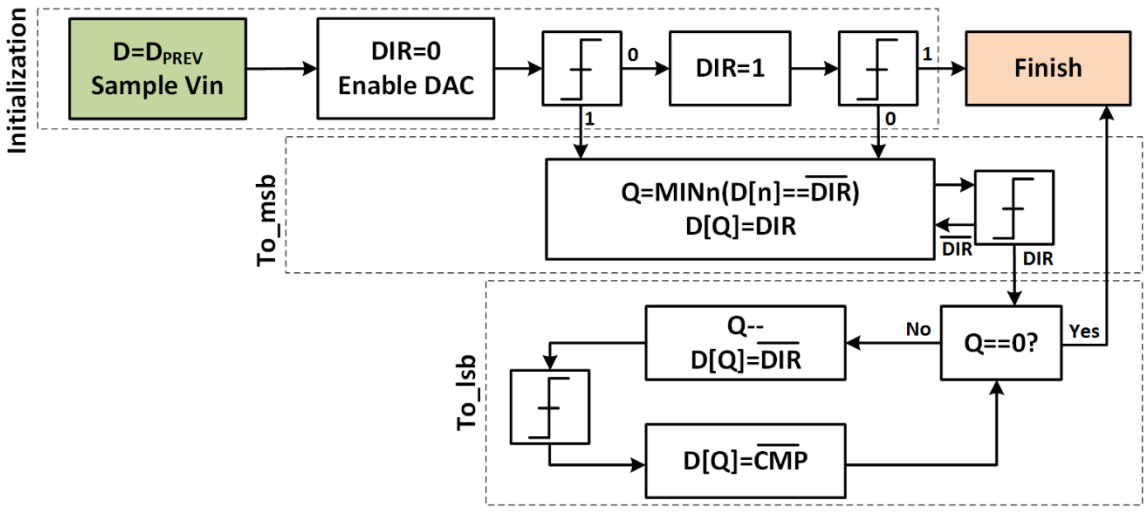


Figure 2-3. LSB-first algorithm.

weighted DAC, usually called dummy capacitor, is used in this algorithm and referred to as DIR for direction. DIR is initially set to 0 and a comparison is made to determine the error and hence the direction in which to make code changes. An error of 0 means that the previous code is less than the current sample and 1 means the previous code is more than the current sample. DIR is updated to 1 if the error is 0 and another comparison is made to determine the error. An error of 1 would mean the current sample code is the same as the previous DPREV and the conversion ends. In all other cases, the next phase, To_msb, is activated.

In the To_msb phase, the algorithm toggles one bit at a time in order to decrease the difference between the estimate and the input. The bit index, Q, chosen to toggle is the least significant bit that is set to not-DIR. Each time D[Q] toggles, comparison is made and the error is checked. The switch in the error value implies an overshoot due to the

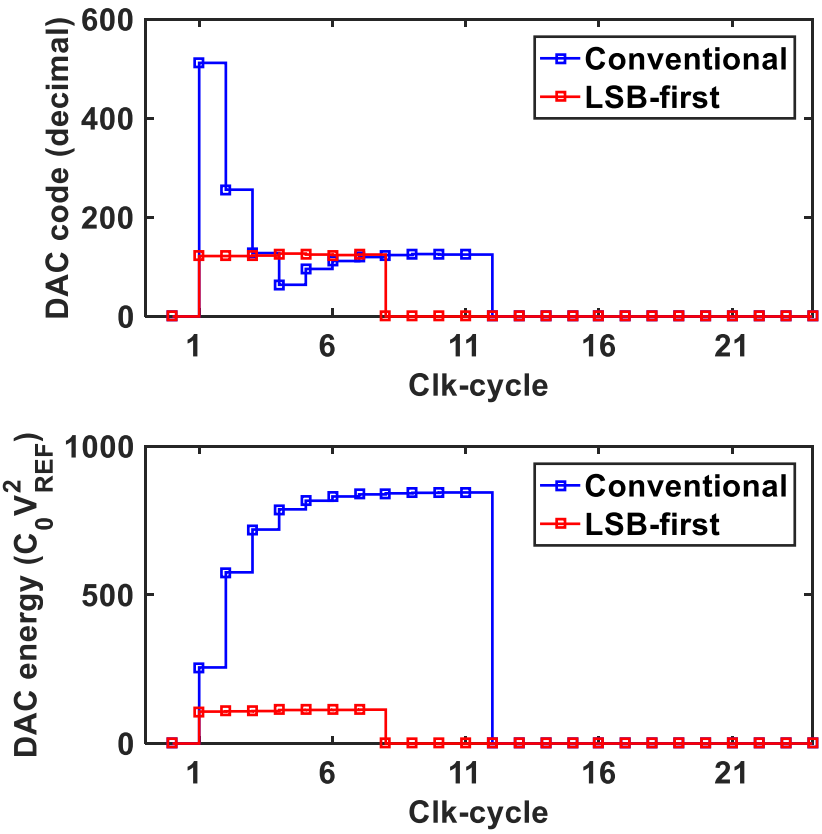


Figure 2-4. Example comparison of bit-cycles and DAC switching energy for LSB-first and Conventional MSB-first A/D conversion.

over correction of the difference. On overshoot the algorithm proceeds to the next phase, To_lsb. In the To_lsb phase, the algorithm performs a conventional SAR search and evaluates the remaining bits, D[Q-1:0].

LSB-first search saves bit-cycles and energy on an average for low activity signals. For example, as shown in Figure 2-4, if the current sample code is 18 and previous sample code was 17 in a 10B ADC, while a conventional MSB-first SAR search would have toggled the MSB bit and required ten bit-cycles and ~853 energy units for this

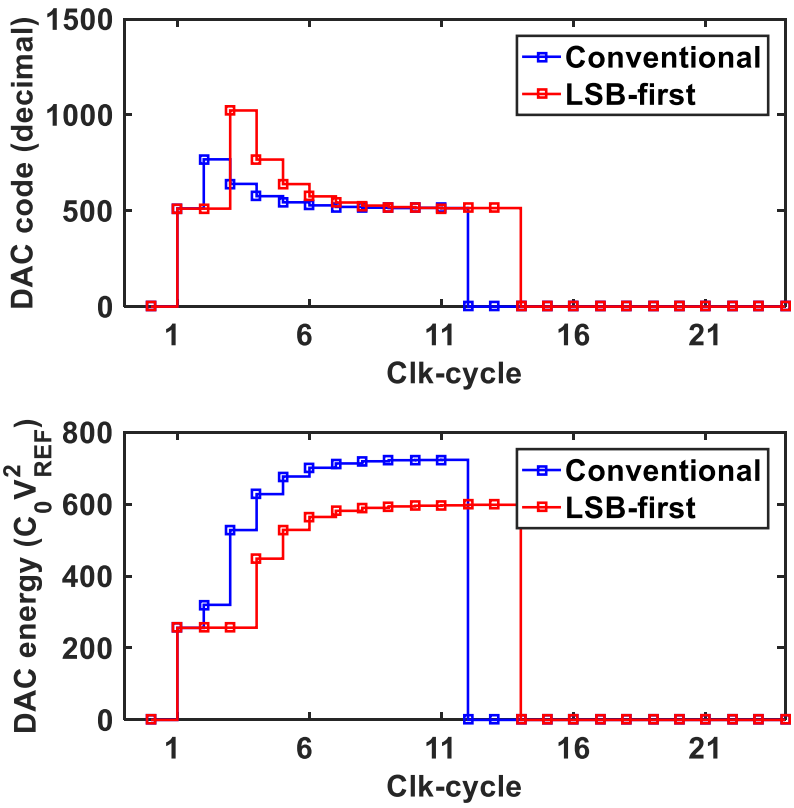


Figure 2-5. Example comparison of bit-cycles and DAC switching energy for LSB-first and Conventional MSB-first A/D conversion.

conversion, LSB-first algorithm does it without toggling the MSB bit and requires only four bit-cycles and ~19 energy units for this full conversion.

However, certain small code transitions still require large switching energy and long bit-cycles. For example, as shown in Figure 2-5 if the current sample code is 514 and previous sample code was 511, LSB-first requires twelve bit-cycles and ~598 units of energy for the conversion even when the current sample code is only three LSB codes more than the previous sample code.

Chapter 3

Recode-then-LSB-first (RLSB-first) algorithm

Many signals from sensors are low activity signals that spend most of its time around middle of the full scale with occasional large activity. A/D conversion of such signals using a conventional ADC with a constant resolution and a full-scale search space consumes unnecessary amounts of time and energy. LSB-first ADC in [17] is suitable for low-activity signals. It reduces the DAC switching energy and the number of bit-cycles required per conversion. However, certain small code transitions still require large switching energy and long bit-cycles. This chapter discusses the new algorithm called Recode-then-LSB-first (RLSB-first) to do A/D conversion of such sensor signals. RLSB-first leads to more energy efficient switching and faster convergence for sensor signal applications for all cases of small changes in sensor signal.

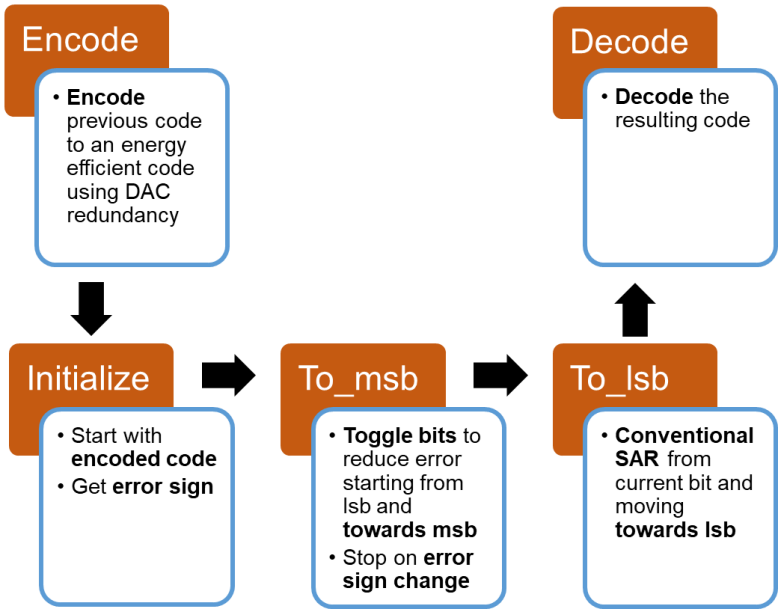


Figure 3-1. RLSB-first block diagram.

In the proposed RLSB-first algorithm [24], there are five phases namely Encode, Initialize, To_msb, To_lsb, and Decode as shown in Figure 3-1. The three phases, Initialize, To_msb, and To_lsb, are similar to the three phases in the LSB-first algorithm. Where this algorithm differs is in the Encode phase where the previous code, DPREV, is first systematically recoded to a new starting previous code before starting LSB-first. This leads to significant savings in energy and reduction in the number of bit-cycles per conversion. After the conversion completes, the result is decoded in the Decode block. We first describe the DAC architecture from a previous work from a new perspective which will be used in the proposed algorithm. It will be followed by the details of the algorithm.

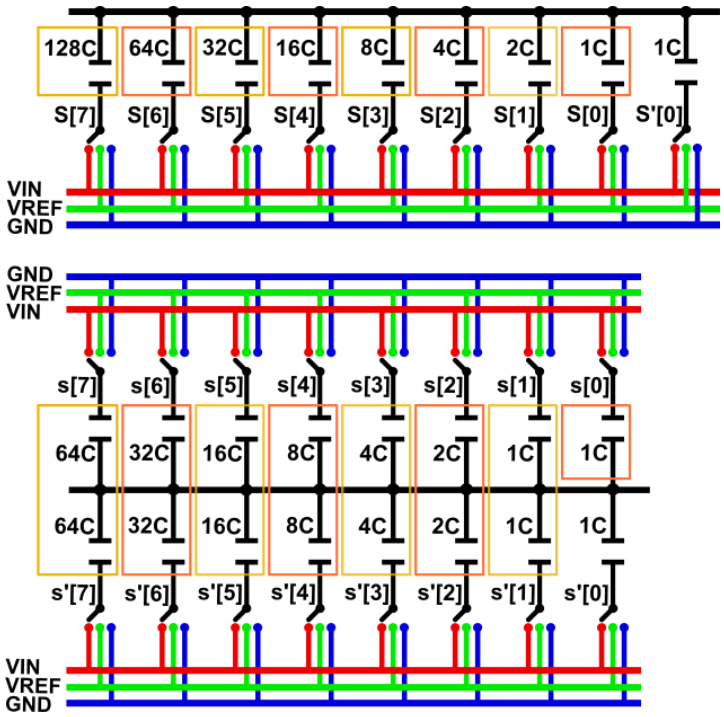


Figure 3-2. Conventional DAC (top) and split-DAC (bottom).

3.1 Split-DAC

A split-DAC architecture is presented in [22], [23] to enable an energy efficient DAC switching for a conventional MSB-first SAR search. In a split-DAC, each bit capacitor above LSB is split into two halves and each half is independently switched as shown in Figure 3-2. This enables three switching states for each bit above LSB. The two states, as shown in Figure 3-3, are the same as in a conventional non-split DAC, namely 1 and 0 where the two halves of a bit are connected to VREF and GND respectively. The additional third state ½ in a split-DAC is a state with one half of a bit connected to VREF and second half of a bit connected to GND. In [22] the split-DAC is used to generate an

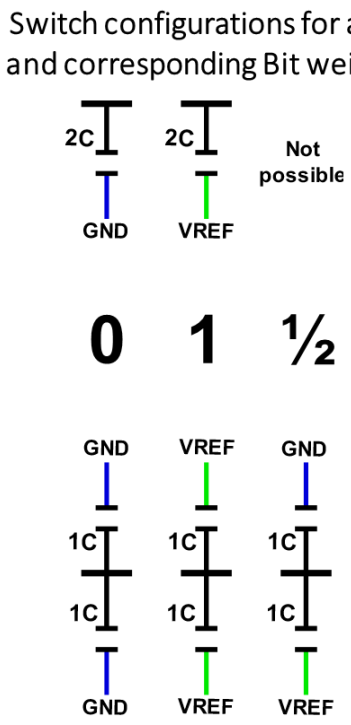


Figure 3-3. Switch configurations for a bit in a Conventional DAC (top) and split-DAC (bottom).

initial DAC voltage level of VREF/2 by setting the LSB to 1 and each bit above to ½. The code equivalence between (10..00) and (½½..½1) to set a voltage level of VREF/2 on a binary weighted split-DAC can be generalized to rules for other voltage levels as discussed later. It is interesting to observe that the split-DAC structure has a large code redundancy for other voltage levels that are not explored in [22]. In the proposed RLSB-first we encode previous sample code, DPREV, using the three possible states (0, 1, and ½) for each bit and then initialize the split-DAC to the encoded three state DPREV before doing LSB-first algorithm SAR search. Encoding DPREV using the method below saves energy and bit-cycles.

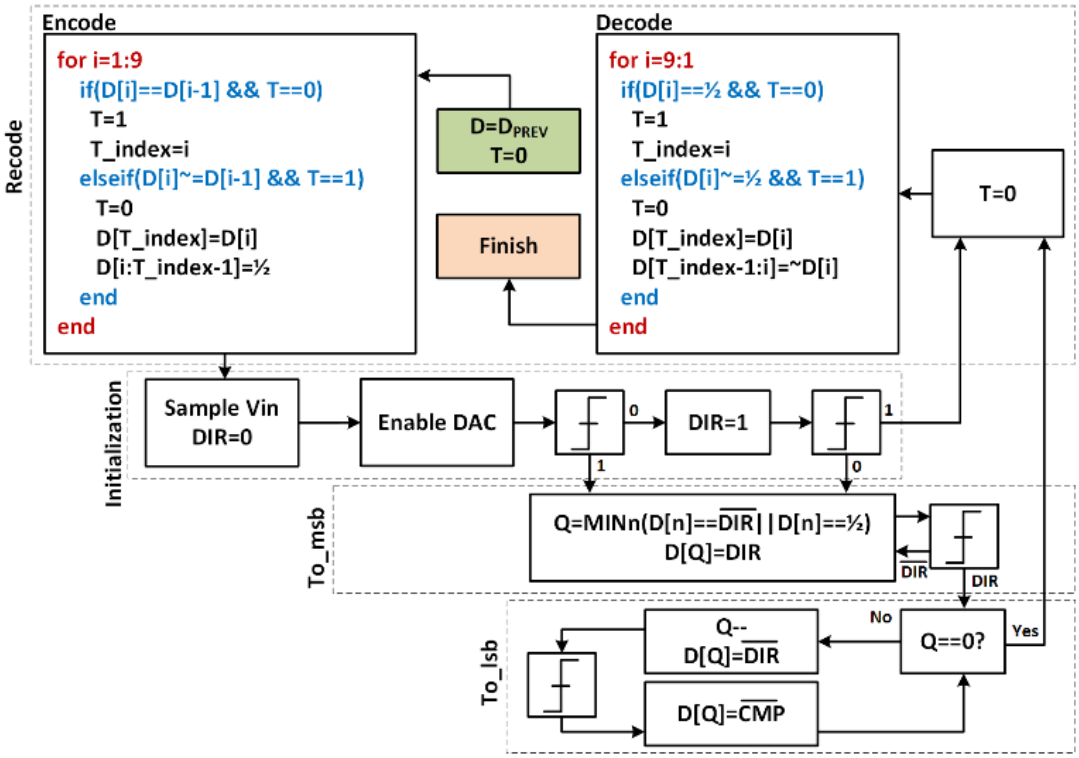


Figure 3-4. RLSB-first algorithm.

3.2 RLSB-first algorithm

In the Recode phase of RLSB-first [24] there are two main blocks, Encode and Decode as shown in Figure 3-4. The Encode block takes a two-state (0, 1) code, DPREV, and converts it into a three-state code (0, 1, ½). The three-state code is then passed onto the Initialization block. The rules for encoding are as follows. In a two-state code, DPREV, we observe that patterns such as (..01111..) or (..10000..) anywhere in the code prevent small code increments or decrements without toggling its MSB bit. Toggling of the MSB

bit for a small code change is energy inefficient. The goal is to recode the two-state code, DPREV, to a much more energy efficient code by selecting one of the many codes that represent the same voltage level as the DPREV by using all three possible states (0, 1, ½) of a split-DAC architecture. In a split-DAC, a bit initially set to a ½ can be toggled to a 0 or a 1. Also patterns such as (..01010..) permit small code changes in either direction without toggling MSB bits. Therefore, if we recode DPREV using a combination of ½ and unequal adjacent bits before the LSB-first algorithm search, it will enable energy efficient small code change without the DAC energy wasted in toggling MSB bits.

There are two key rules for patterns in codes that generate equivalent DAC voltage levels in a binary weighted split-DAC. Rule X1 in Table 3-1 indicates that a bit is equivalent to half of an adjacent MSB-side bit. Using Rule X1 we can replace (01) in a code with (½0). If we have (011) then applying Rule X1 once would give us (½01) and applying Rule X1 again on (½01) will give us (½½0). This gives us our Rule X2 which replaces (011) with (½½0). In general, 0 followed by n 1s can be replaced by n ½s followed by 0 by repeatedly using Rule A. This gives us our most general Rule X.

Table 3-1. Summary of recoding rules.

Code using 0 or 1	**Code using 0, 1, or ½**	
…01…	…½0…	Rule X1
…10…	…½1…	Rule Y1
…011…	…½½0…	Rule X2
…100…	…½½1…	Rule Y2
…0[n x 1]…	…[n x ½]0…	Rule X
…1[n x 0]…	…[n x ½]1…	Rule Y

Similarly, Rule Y1 indicates that a bit is equivalent to the sum of a half of it and a full adjacent LSB-side bit. Using Rule Y1 we can replace (10) in a code with (½1). If we have (100) then applying Rule Y1 once would give us (½10) and applying Rule Y1 again on (½10) will give us (½½1). This gives us our Rule Y2 which replaces (100) with (½½1). In general, 1 followed by n 0s can be replaced by n ½s followed by 1 by repeatedly using Rule Y1. This gives us our most general Rule Y.

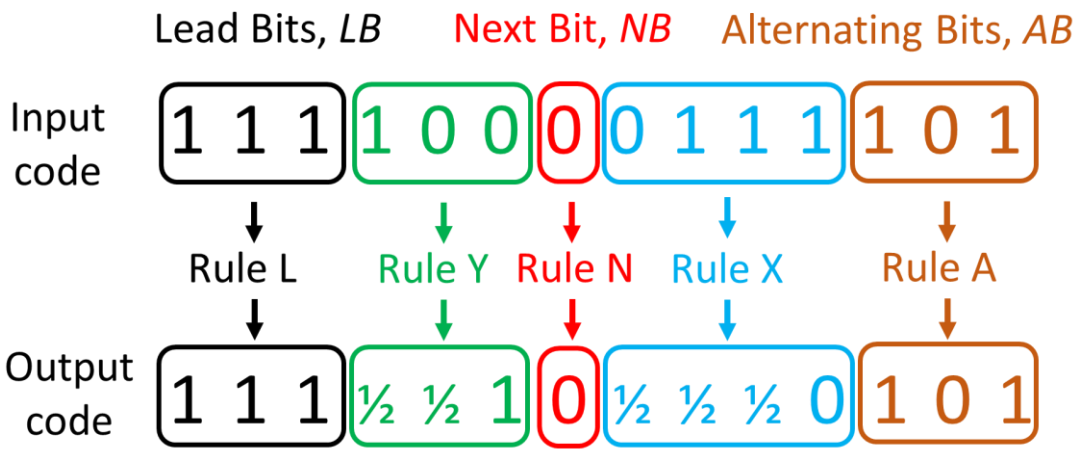


Figure 3-5. Encoding rules shown with an example.

The encoding steps are best illustrated with an example in Figure 3-5. To encode a binary number, we start from the LSB end and invoke the Rule X and Y only when we encounter two adjacent 1s or 0s. The Alternating Bits, AB, are output as it is in the encoded code as it would at most cause a 1B jump to change code in either direction when doing A/D conversion from the LSB side. Next set of bits, 0111, has repeating 1s which are recoded using Rule X. Following that the bit labelled Next Bit, NB, is unchanged as it has a ½ to its right in the encoded output. An adjacent ½ can be toggled in any direction and so unchanged NB would at most cause a 1B jump to change code in either direction during conversion. Following NB, we have another set of repeating 0s in 100 which is recoded using Rule Y. After that there are repeating 1s labelled Lead Bits, LB, leading up-to the MSB. These bits are not recoded and are left unchanged in the encode output. The reason to leave them unchanged is because all 1s, or all 0s cannot be encoded using Rule X or Rule Y which requires an inverted bit, 0 or 1 respectively, as the

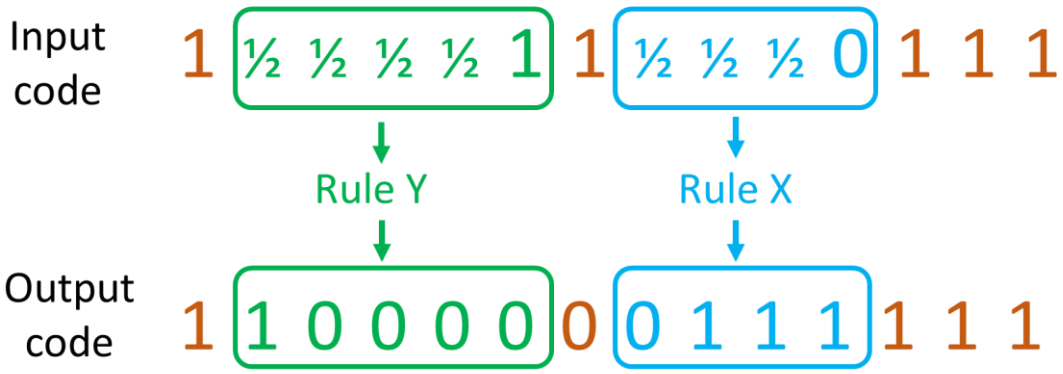


Figure 3-6. Decoding rules shown with an example.

left most bit. Another reason is that there is no need to encode LB because all 1s are already arranged perfectly to do a code search below the code value it represents and there are no codes to search above all 1s. Similarly, all 0s in the Lead Bits are arranged perfectly to do a code search above the code value it represents and there are no codes to search below all 0s.

After encoding in the Recode phase, the algorithm proceeds with the next three phases as in LSB-first algorithm. After the three phases, the three state (0, 1, ½) output is returned back to the Recode phase and processed by the Decode block into a two-state code (0, 1) after the conversion finishes. The Decode block is the opposite operation of the Encode block and the details of the steps are shown in Figure 3-4. An example of decoding is shown in Figure 3-6. In this example Rule Y and Rule X are used to convert

½s followed by a 1 or 0 to a code that uses only 1s and 0s. All other 1s and 0s in the input code are kept as it is into the output code.

To compare and calculate energy for switching steps using a split-DAC for LSB-first and RLSB-first, we need to be able to calculate switching energy for intermediate arbitrary steps for split-DAC switching from initial to final split-DAC codes of Dinit[9:0] and Dfinal[9:0] respectively. The derivation is adapted from [22] and presented below.

3.3 Energy Calculation

Assume a 10B binary weighted Split-DAC switching from initial code of D_{init}[9:0] to the final code D_{final}[9:0]. The capacitance W[i] of the i-th bit is given by

$$W[i] = C_0 \quad \text{for } i = -1 \tag{1}$$

$$W[i] = 2^{i-1}C_0 \quad \text{for } 9 \geq i \geq 0 \tag{2}$$

In (1), index i=-1 is for the dummy capacitor in the DAC and C_0 is a unit capacitance. The bits of a DAC can be set according to

$$D[i] = 0 \text{ or } 1 \quad \text{for } i = 0, -1 \tag{3}$$

$$D[i] = 0, 1/2, \text{or } 1 \quad \text{for} \quad 9 \geq i \geq 1 \tag{4}$$

At places dummy bit D[-1] will be called DIR. The total DAC capacitance is given by (5) and the DAC capacitance connected to the reference voltage V_{REF} is given by

$$C_{TOT} = 2^{10}C_0 \tag{5}$$

$$C_{TOP} = \sum_{i} D[i]W[i] \tag{6}$$

Let V_{IN} be the sampled input voltage. The voltage level at the DAC top plate is given by (7) and the change in voltage level at the DAC top plate is given by (8)

$$V_X = \frac{V_{REF}}{2} - V_{IN} + \frac{C_{TOP}V_{REF}}{C_{TOT}} \tag{7}$$

$$V_{X,del} = V_{X,final} - V_{X,init} = \frac{(C_{TOP,final} - C_{TOP,init})V_{REF}}{C_{TOT}} \tag{8}$$

Energy required for DAC transition can be understood as having two parts, E_{SW} and E_{TOP}, as in [22]. The E_{SW} is the energy required to switch the DAC from the initial configuration to the final configuration without changing the voltage level on the DAC top plate. The E_{TOP} is the energy required for charging capacitors that are connected to V_{REF} in the final configuration due to change in the voltage level on the DAC top plate. E_{TOP} is given by

$$E_{TOP} = -C_{TOP,final}V_{X,del}V_{REF} \tag{9}$$

E_{SW} is calculated by first figuring out the increase in capacitance that gets connected to V_{REF} in going from the initial to final code at each bit location using (10), (11), (12), and (13)

$$D_{DEL}[i] = D_{final}[i] - D_{init}[i] \quad (10)$$

$$D_{SW}[i] = D_{DEL}[i] \text{ for } D_{DEL}[i] > 0 \quad (11)$$

$$D_{SW}[i] = 0 \text{ for } D_{DEL}[i] \leq 0 \quad (12)$$

$$C_{SW} = \sum_{i} D_{SW}[i]W[i] \quad (13)$$

The switching energy is then calculated

$$E_{SW} = C_{SW}V_{REF}V_{REF}$$

The total energy required for DAC transitioning from initial to final configuration is the given by

$$E_{TOT} = E_{SW} + E_{TOP}$$

3.4 Comparison

The energy calculation formulas for arbitrary transitions on split-DAC enable us to compare energy consumption changes while using LSB-first and RLSB-first. Let us assume a previous sample DPREV 511 and new current sample being converted DNEW 514 in a 10B A/D conversion. In Table 3-2 LSB-first takes 12 bit-cycles and ~598 units of DAC switching energy to do a full conversion. Whereas in Table 3-3 for the same DPREV and DNEW, recoding in RLSB-first eliminates MSB bit toggling and completes a full conversion using 6 bit-cycles and only 261 units of DAC switching energy. Note that there is no extra analog energy needed by RLSB-first algorithm to initialize the split-DAC with the recoded previous code. This is because recoding happens in the digital domain and initializing DAC with a previous code in any configuration cost the same analog energy.

Table 3-2. LSB-first example conversion.

D_{PREV}[9:0]= 01111 11111 **(**511 in decimal)					
D_{NEW}[9:0]= 10000 00010 (514 in decimal)					
Bit-cycle, DIR	**Dinit[9:0]**	**Dfinal[9:0]**	**E_{TOP}**	**E_{SW}**	**E_{TOT}**
1, DIR=0	00000 00000	01111 11111	-255.0	511.0	256.0
2, DIR=1	01111 11111	01111 11111	-0.5	0.5	256.5
3, DIR=1	01111 11111	11111 11111	-512.0	512.0	256.5
4, DIR=1	11111 11111	10111 11111	192.0	0.0	448.5
5, DIR=1	10111 11111	10011 11111	80.0	0.0	528.5
6, DIR=1	10011 11111	10001 11111	36.0	0.0	564.5
7, DIR=1	10001 11111	10000 11111	17.0	0.0	581.5
8, DIR=1	10000 11111	10000 01111	8.3	0.0	589.7
9, DIR=1	10000 01111	10000 00111	4.1	0.0	593.8
10, DIR=1	10000 00111	10000 00011	2.0	0.0	595.8
11, DIR=1	10000 00011	10000 00001	1.0	0.0	596.8
12, DIR=1	10000 00001	10000 00010	-0.5	2.0	598.3

Table 3-3. RLSB-first example conversion.

D_{PREV}[9:0]= 01111 11111 **(**511 in decimal)					
D_{NEW}[9:0]= 10000 00010 (514 in decimal)					
Encoding	01111 11111	½½½½½ ½½½01			
Bit-cycle, DIR	**Dinit[9:0]**	**Dfinal[9:0]**	**E_{TOP}**	**E_{SW}**	**E_{TOT}**
1, DIR=0	00000 00000	½½½½½ ½½½01	-255.0	511.0	256.0
2, DIR=1	½½½½½ ½½½01	½½½½½ ½½½01	-0.5	1.0	256.5
3, DIR=1	½½½½½ ½½½01	½½½½½ ½½½11	-1.0	2.0	257.5
4, DIR=1	½½½½½ ½½½11	½½½½½ ½½111	-1.0	2.0	258.5
5, DIR=1	½½½½½ ½½111	½½½½½ ½½101	1.0	0.0	259.5
6, DIR=1	½½½½½ ½½101	½½½½½ ½½110	-0.5	2.0	261.0
Decoding	½½½½½ ½½110	10000 00010			

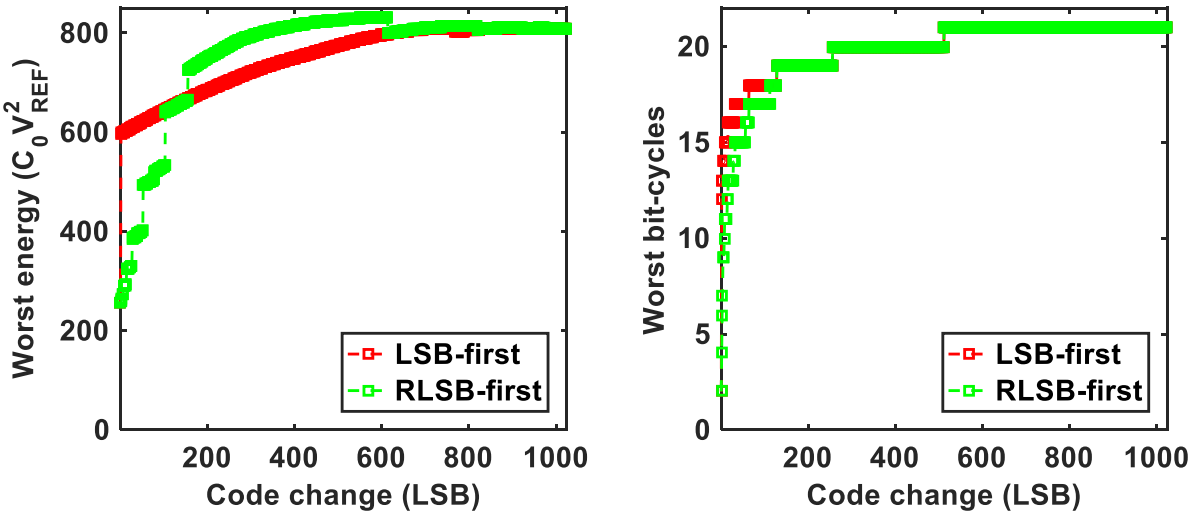


Figure 3-7. Worst energy (left) and worst bit-cycles (right) vs. code change.

For a small code change, the worst-case energy for the RLSB-first algorithm is ~2.5X lower than the LSB-first algorithm as shown in Figure 3-7 (left). For a large code change, the worst-case energy for the RLSB-first matches the LSB-first algorithm. For further energy benefits, RLSB-first could be modified to do a split-DAC SAR search as in [22], [23], instead of a conventional SAR search in the To_lsb step of the algorithm. However, since large code changes are infrequent in sensor signal A/D conversion, this implementation does a conventional SAR search for simplicity. For a small code change, the worst-case number of bit-cycles in the proposed RLSB-first is ~3X lower than LSB-first as shown in Figure 3-7 (right). For a large code change, the worst number of bit-cycles for RLSB-first matches LSB-first.

The average energy with a sinusoidal input centered at MSB code transition point for 0.49Fs Hz and 0.10Fs Hz versus varying amplitude is lower than LSB-first as shown in Figure 3-8. The average energy approaches 250 units of $C_0V_{REF}^2$ for small amplitudes at both low and high frequencies. Equivalently in terms of total 10B DAC capacitance of C_{dac} ($1024C_o$) on one side of a differential DAC, the average energy approaches $\sim C_{dac}V_{REF}^2/4$. The total differential DAC energy is $\sim C_{dac}V_{REF}^2/2$ which is the desired least energy as discussed in section 1.1 of Chapter 1. The average bit-cycles with a sinusoidal input centered at MSB code transition point for 0.49Fs Hz, and 0.10Fs Hz versus varying amplitude are much better than LSB-first for low amplitude as shown in Figure 3-9. The average bit-cycles approach ~2 for small amplitudes which is a desired result as discussed in section 1.2 of Chapter 1.

RLSB-first algorithm was also simulated with real data sourced from various databanks scaled to a 10B resolution. Figure 3-10 shows results with raw ECG input signal sourced from the database ecgiddb at Physionet [38,41]. Average RLSB-first energy and bit-cycles are about 13% and 4.3% lower compared with LSB-first respectively. Figure 3-11 shows results for bearing vibration input signal sourced from [39]. RLSB-first energy and bit-cycles are about 29% and 14.7% lower compared with LSB-first respectively. Figure 3-12 shows results for EMG input signal from patient with neuropathy condition from database emgdb at Physionet [38,40]. RLSB-first energy and bit-cycles are about 9% and 11% lower compared with LSB-first respectively. Figure 3-13 shows results for EMG input signal from patient with myopathy condition from

database emgdb at Physionet [38,40]. RLSB-first energy and bit-cycles are about 14.7% and 9.7% lower compared with LSB-first respectively.

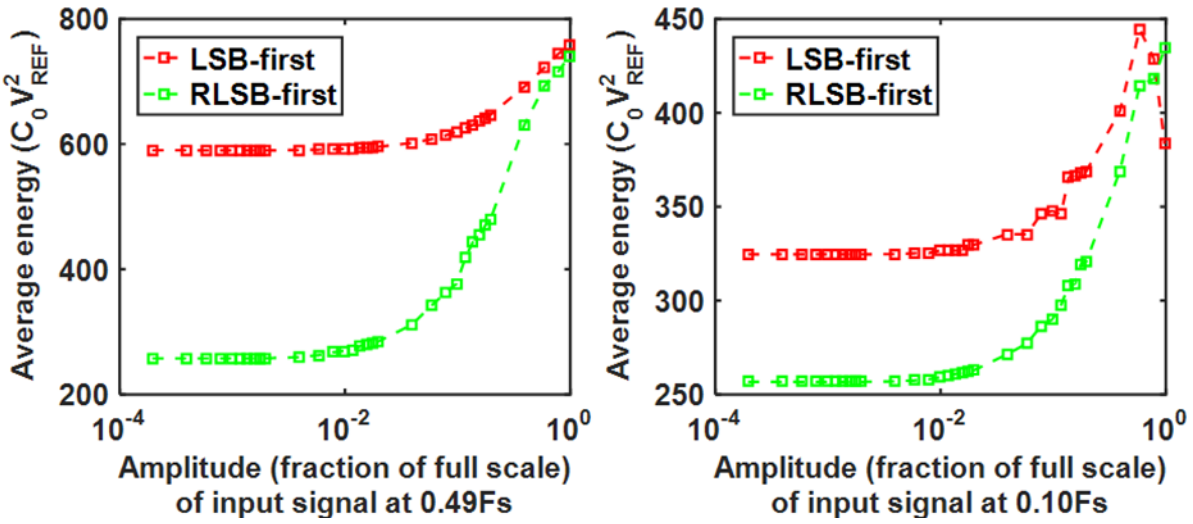


Figure 3-8. Average energy vs. input signal amplitude at 0.49Fs (left) and 0.10Fs (right).

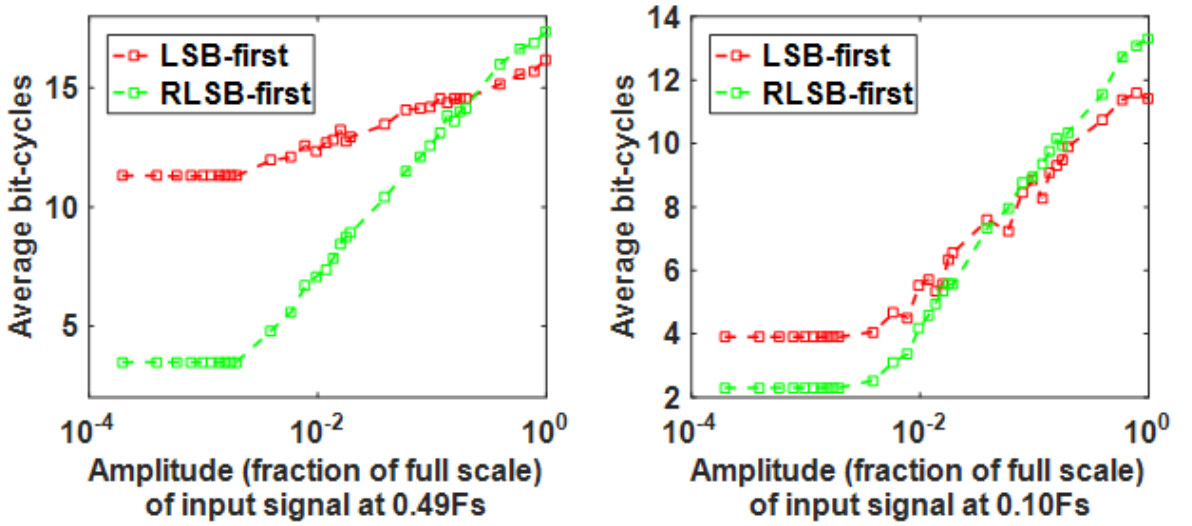


Figure 3-9. Average bit-cycles vs. input signal amplitude at 0.49Fs (left) and 0.10Fs (right).

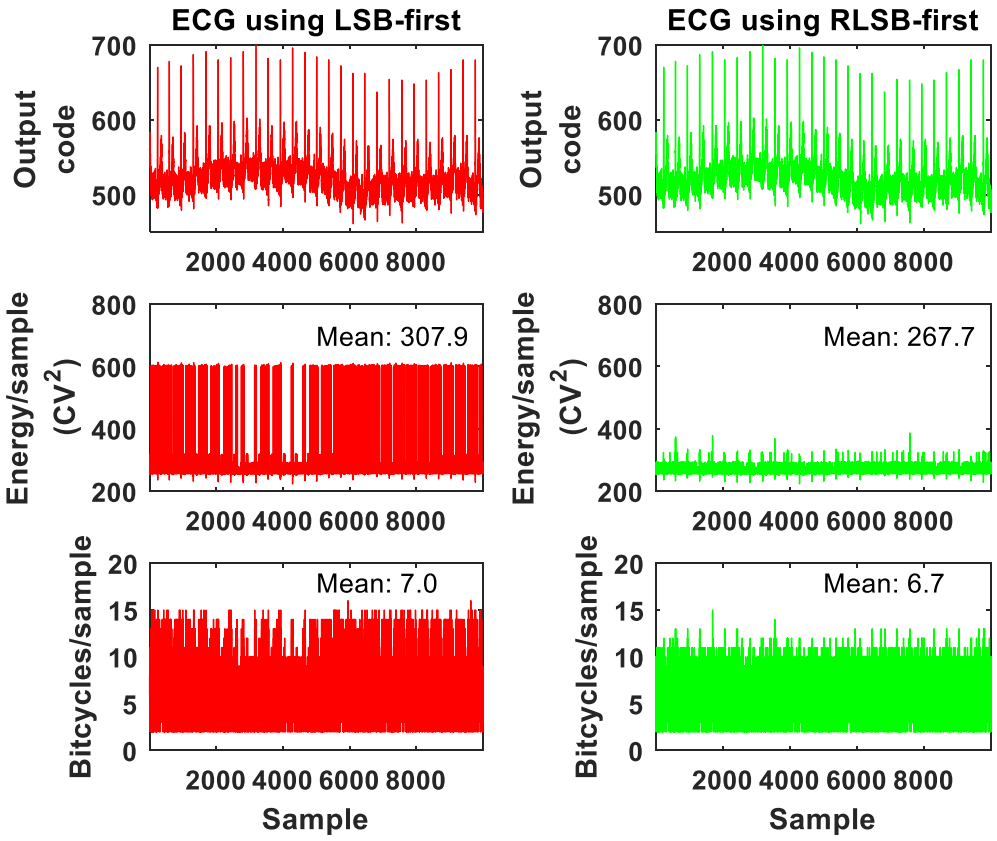


Figure 3-10. Energy and bit-cycles for ECG signal sampled at 500Hz using LSB-first (left) and RLSB-first (right). Data sourced from database ecgiddb at Physionet [38,41]. Mean RLSB-first energy savings compared with LSB-first is 13%. Mean RLSB-first bit-cycle savings compared with LSB-first is 4.3%. Mean code change is 6.4.

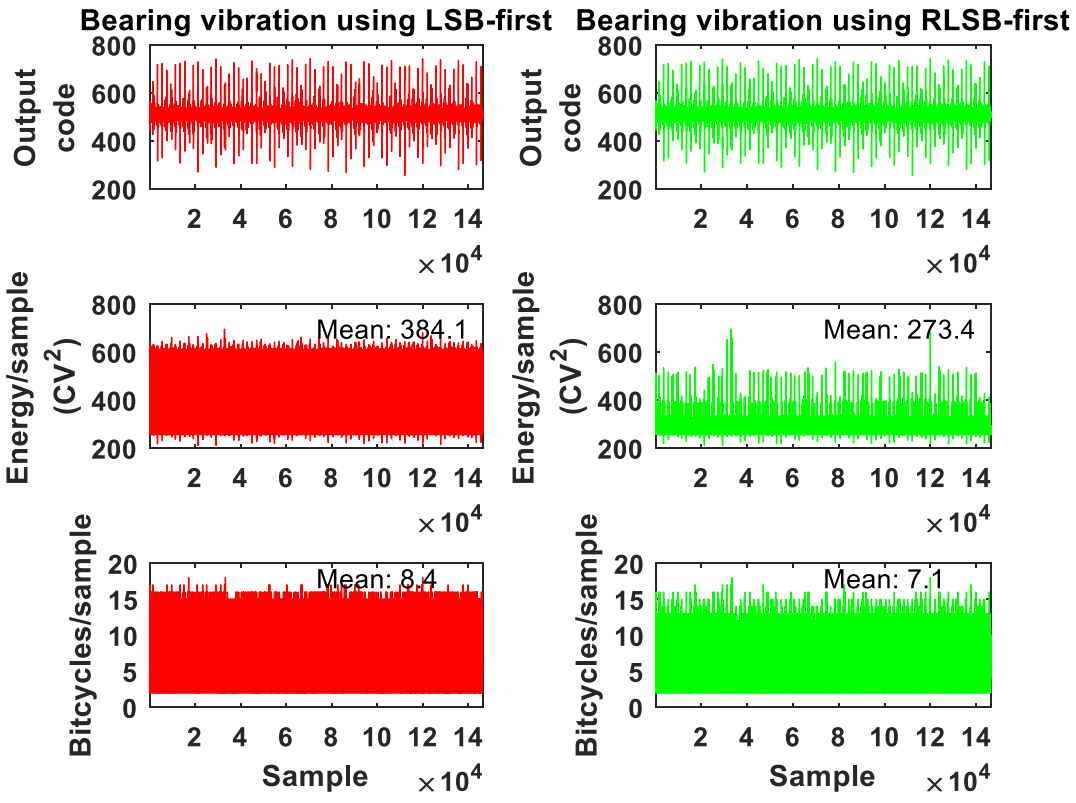


Figure 3-11. Energy and bit-cycles for bearing vibration signal sampled at 48KHz using LSB-first (left) and RLSB-first (right). Data sourced from [39] under “seven inner race fault condition”. Mean RLSB-first energy savings compared with LSB-first is 28.8%. Mean RLSB-first bit-cycle savings compared with LSB-first is 14.7%. Mean code change is 8.5.

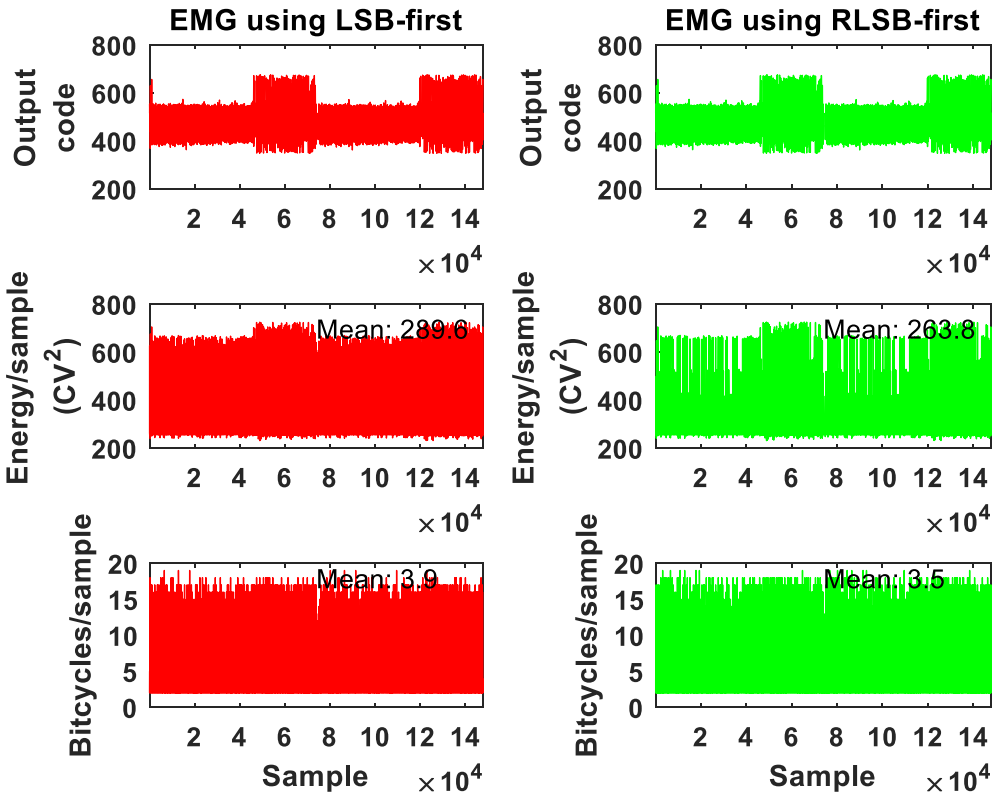


Figure 3-12. Energy and bit-cycles for EMG signal sampled at 4KHz using LSB-first (left) and RLSB-first (right). Data sourced from database emgdb/neuropathym at Physionet [38,40]. Mean RLSB-first energy savings compared with LSB-first is 9%. Mean RLSB-first bit-cycle savings compared with LSB-first is 11%. Mean code change is 3.9.

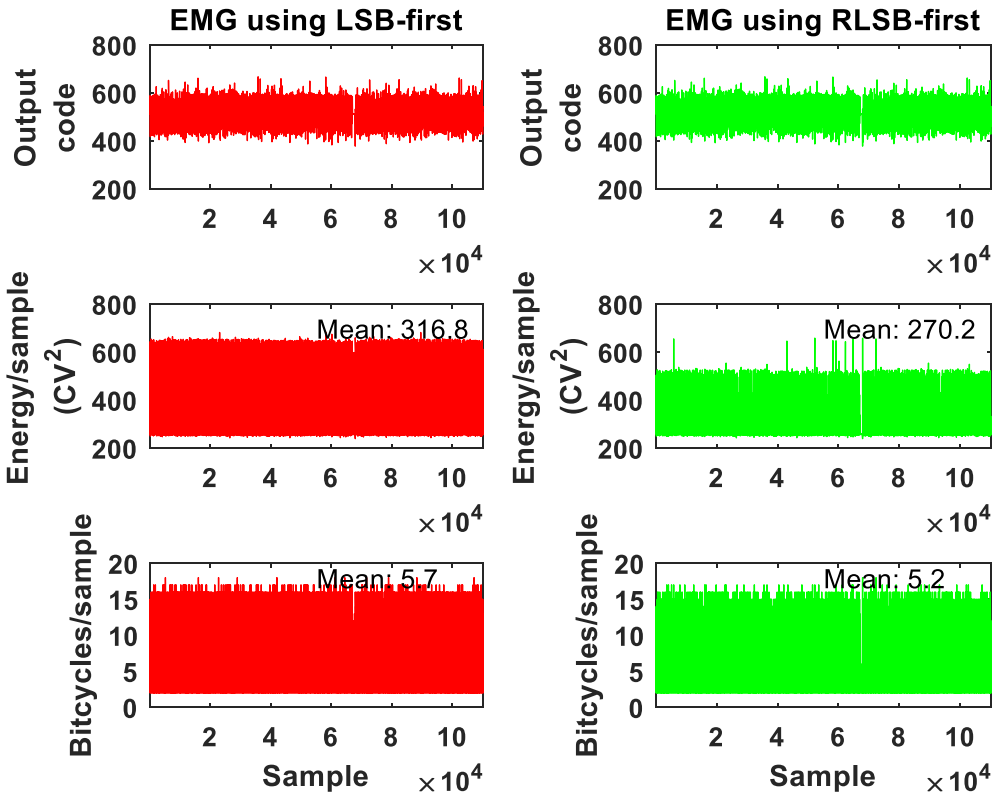


Figure 3-13. Energy and bit-cycles for EMG signal sampled at 4KHz using LSB-first (left) and RLSB-first (right). Data sourced from database emgdb/myopathym at Physionet [38,40]. Mean RLSB-first energy savings compared with LSB-first is 14.7%. Mean RLSB-first bit-cycle savings compared with LSB-first is 9.7%. Mean code change is 7.1.

3.5 RLSB-first benefits

The proposed RLSB-first algorithm achieves low switching energy and bit-cycles for small code changes over previous sample both in the worst case and also on an average. The algorithm achieves this by recoding previous code before initializing the split-DAC and performing LSB-first algorithm. Recoding previous code in the digital domain avoids toggling MSB bits on the DAC thereby saving both energy and bit-cycles per conversion. The simulation results for RLSB-first show improvements of upto 3X in worst case energy and has the low average switching energy for small code change over previous sample in comparison to LSB-first.

3.6 System architecture

This section provides an overview of the complete SAR ADC system architecture designed in this work and as shown in Figure 3-10. The goal is to do an energy efficient 10B SAR A/D conversion by using RLSB-first to take advantage of the small changing amplitude of the sensor signals. The saved bit-cycles will be used to enhance resolution during small input amplitudes, correct for capacitive mismatches in DAC, and measure comparator noise to support resolution enhancement. The analog blocks consist of a comparator and a differential 10B split-DAC coupled to a 7B calibration DAC. The digital

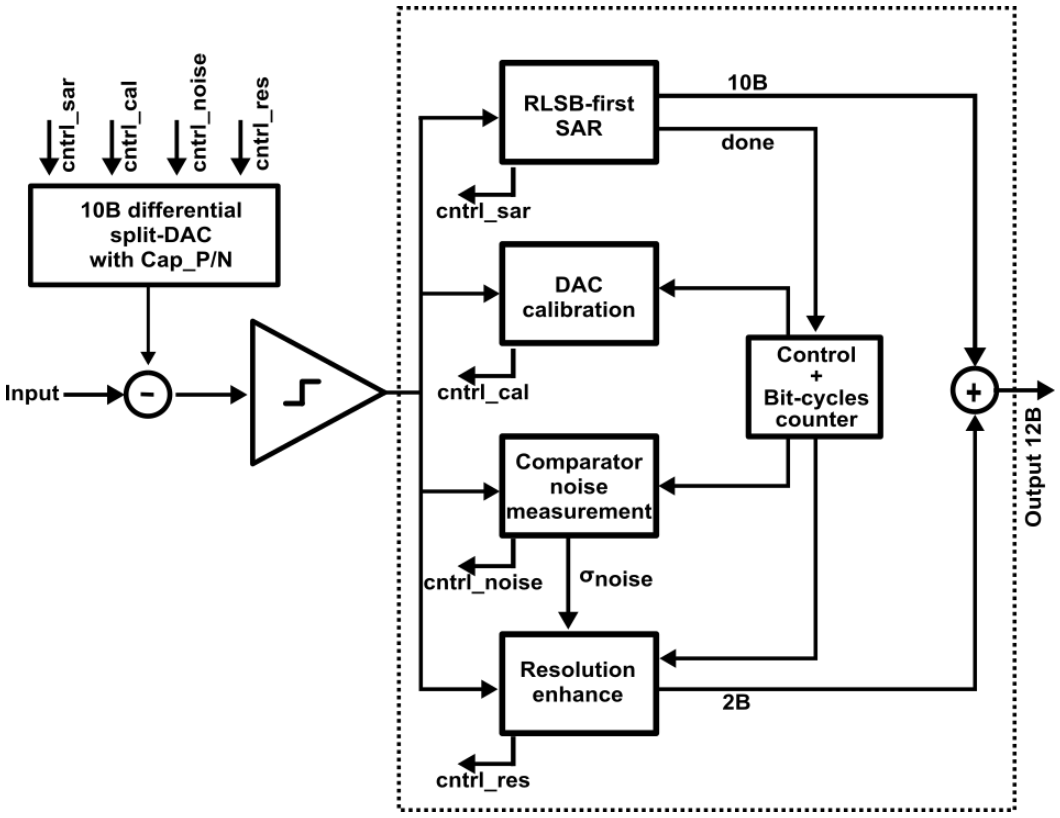
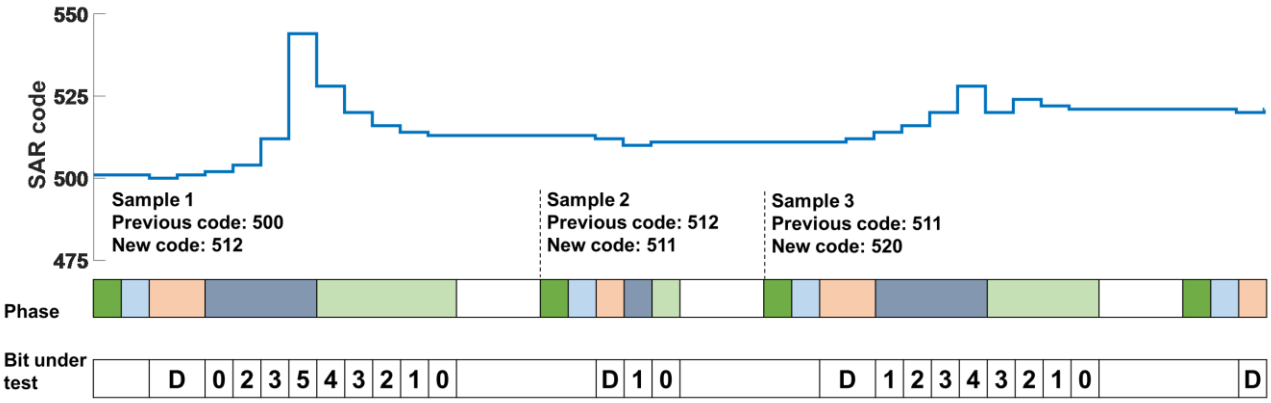


Figure 3-14. System architecture block diagram.

block consists of the sub-blocks namely RLSB-first SAR, DAC Calibration, Comparator Noise Measurement, and Resolution Enhancement.

The ADC uses RLSB-first to do a 10B SAR conversion. If the current signal sample is close to the previous input sample then the algorithm completes the conversion in a few bit-cycles. This saves energy in both DAC and comparator and also unused bit-cycles. The saved bit-cycles are used for additional benefits by scheduling other modules as per Table 3-4. If the10B conversion uses seven or fewer cycles, the saved bit-cycles are used for an additional 2B resolution enhancement. The resolution enhancement is



Color	Phase Name
	Recode
	Sample
	Dir
	ToMSB
	ToLSB

Modules to Schedule in a Conversion Cycle	If RLSB-first SAR Bit-cycles
RLSB-first SAR (10B)	Always scheduled
Resolution Enhance (2B)	<8
Noise measure (if enabled)	8 to 15
DAC calibration (if enabled)	8 to 15

Figure 3-15. Phase diagram (above) and module schedule (bottom) conditioned on bit-cycles used during 10B RLSB-first.

based on a modification of the technique in [25]. If the 10B conversion uses between eight and fifteen bit-cycles, some bit-cycles are used for comparator noise measurement and DAC mismatch calibration if the modules are enabled. The comparator noise measurement is needed for resolution enhancement. Measuring noise in background tracks temperature and process variations. DAC mismatch calibration makes it possible to use small size, large mismatch unit capacitors.

Chapter 4

Resolution enhancement

Resolution enhancement based on noise in multiple comparators was previously reported [25]. This technique uses the noise present in multiple comparators to make a residue estimation, using the inverse error function to linearize and map comparator outputs to final output estimation. In this work, instead of using multiple comparators, a single comparator is strobed multiple times exploiting the spare time created by the RLSB-first algorithm. Issues related to unequal offset voltages of multiple comparators are also avoided. In addition, a coarse step is added to improve the accuracy of resolution enhancement.

4.1 Steps for resolution enhancement

RLSB-first algorithm concludes the 10B conversion early when signal amplitudes are small. The spare time after conversion can be translated into increasing dynamic range through resolution enhancement. After the RLSB-first algorithm finishes early, the spare

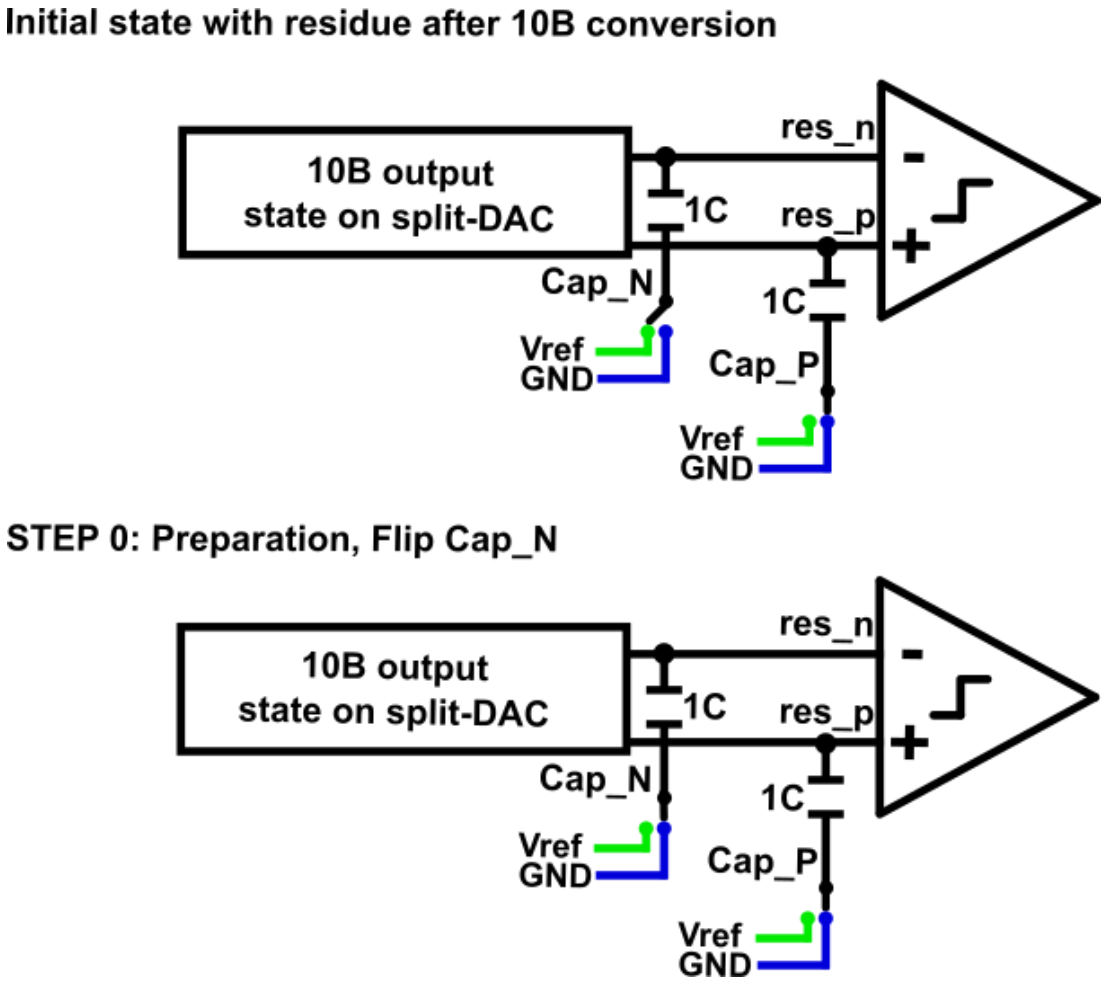


Figure 4-1. Initial state and preparation (STEP 0) after 10B RLSB-first conversion ends.

bit cycles are used to enhance resolution by 2B by strobing comparator 2N times in two steps. Two extra unit capacitors, Cap_P and Cap_N, are used in this implementation on P and N side of the differential DAC respectively. During the first 10B SAR conversion, Cap_P and Cap_N stay connected to GND and Vref, respectively. After the 10B conversion ends, the initial state with residue is as shown in Figure 4-1. Before starting the 2B resolution enhancement, Cap_N is toggled to GND in the preparation STEP 0 as

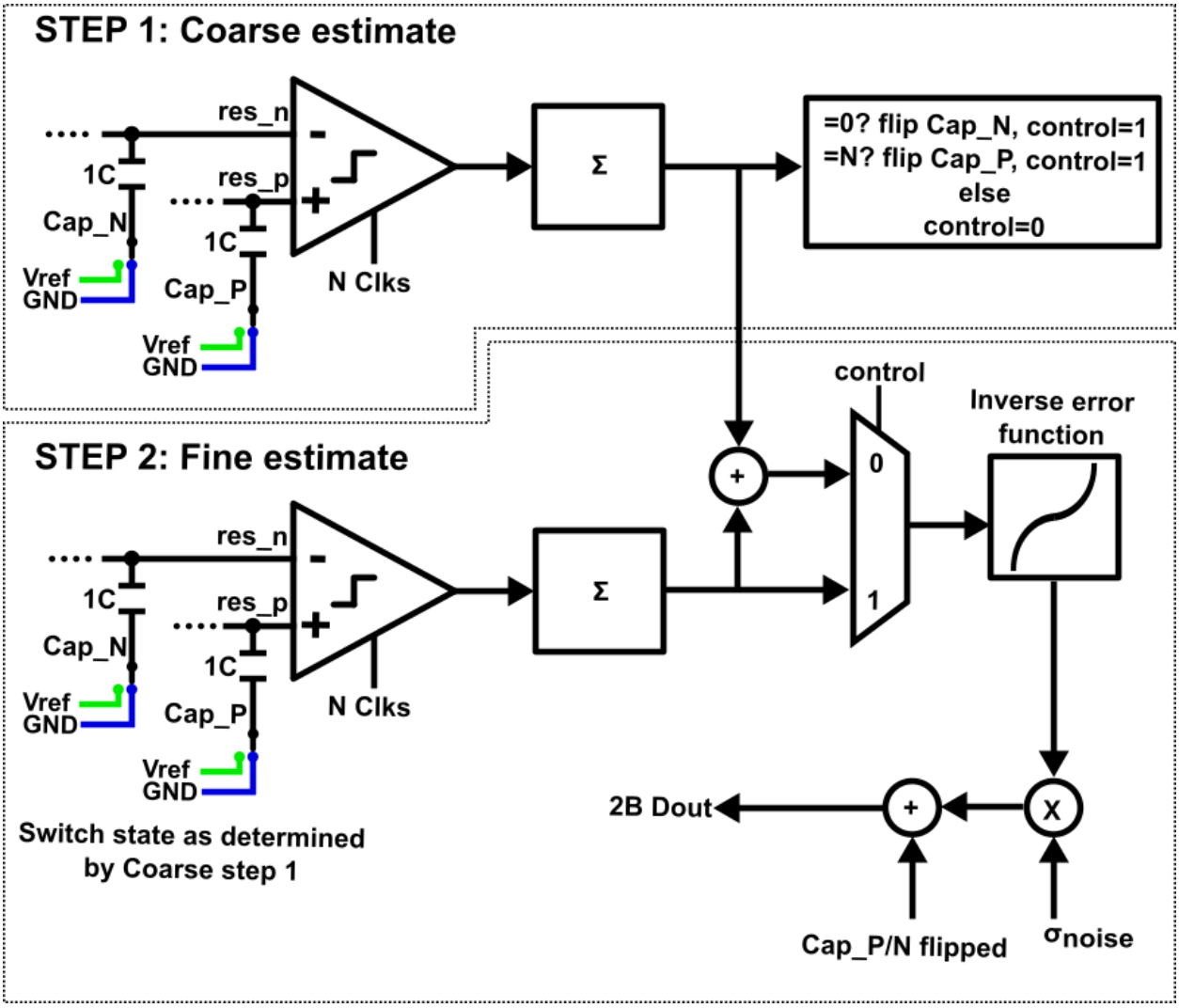


Figure 4-2. Coarse estimate (STEP 1) and fine estimate (STEP 2) of resolution enhancement.

shown in Figure 4-1. This toggle adds a 10B ½ LSB to the split-DAC. After this reference addition, the comparator is repeatedly strobed N times to make a coarse estimate in STEP 1 as shown in Figure 4-2. If comparator decisions are all 1 or all 0s, the residue is likely to be close to the positive or negative edge of the residue range respectively. In case of all 1s, Cap_P is toggled to Vref and if all 0s, Cap_N is toggled to Vref, effectively adding

or subtracting an additional ½ LSB. This brings the residue within the higher accuracy range of the stochastic estimator whose accuracy falls off as the residue moves farther away from zero [25]. This is followed by fine estimation step as shown in STEP 2. The comparator is repeatedly strobed N times and the average of which goes through an inverse error function estimated during the comparator noise measurement. If no reference addition or subtraction is made in the coarse step, the average is taken over all spare cycles as illustrated with setting control signal to 0 in Figure 4-2. The result is truncated to 2B to produce the additional resolution.

4.2 Computation of the extra 2B

When the coarse estimate doesn't result in all 0s or all 1s in the STEP 1 in Figure 4-2, that is the result doesn't saturate, the average M is taken over all 2N comparisons in both coarse and fine steps. In this case, the equation in [25] with an added 10B ½ LSB is used to estimate and output the 2B after STEP 2

$$\mathrm{D_{OUT}} = \mathrm{f(M)} = \sqrt{2}\sigma_{\mathrm{noise}}\mathrm{erf}^{-1}(2\mathrm{M}-1) + \frac{1}{2}\mathrm{LSB}$$

The σ_{noise} is the measured comparator noise in units of 10B LSB and steps to measure it are detailed in the next chapter. When the coarse estimate saturates in STEP 1 and Cap_N or Cap_P are flipped in response, the average M is taken over only N comparisons

made in the fine estimate STEP 2. In this case, the following equations are used to estimate 2B after STEP 2

$$\mathrm{D_{OUT}} = \mathrm{f(M)} = \sqrt{2}\sigma_{\mathrm{noise}}\mathrm{erf}^{-1}(2\mathrm{M}-1) + \frac{1}{2}\mathrm{LSB} + \frac{1}{2}\mathrm{LSB}$$

(if Cap_N is flipped in STEP 1)

$$\mathrm{D_{OUT}} = \mathrm{f(M)} = \sqrt{2}\sigma_{\mathrm{noise}}\mathrm{erf}^{-1}(2\mathrm{M}-1) + \frac{1}{2}\mathrm{LSB} - \frac{1}{2}\mathrm{LSB}$$

(if Cap_P is flipped in STEP 1)

4.3 Accuracy benefits of two-step

The addition of the coarse step improves the accuracy for a given number of comparator cycles 2N and noise σ_{noise}. This also makes the ADC more tolerant to errors due to incomplete DAC settling and comparator noise by increasing the residue estimation range. The following derivation is an adaptation of [25] to our case with an additional coarse step. The probability of obtaining a positive comparator decision is given by

$$\mathrm{p} = \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{\mathrm{v_{in}}}{\sqrt{2}\sigma_{\mathrm{noise}}}\right)\right]$$

The vin is the comparator input residue during resolution enhancement. The rms output error σ_{DOUT} due to stochastic estimation when comparator makes 2N comparisons is given as in [25]

$$\sigma_{DOUT} = \frac{\sigma_{noise}}{\sqrt{2N}} S$$

where S is

$$S = \sqrt{2\pi} \frac{\sqrt{p(1-p)}}{e^{-\frac{v_{in}^2}{2\sigma_{noise}^2}}}$$

For a uniformly distributed input residue and $-v_{range} < v_{in} < v_{range}$ the average rms error in the output is given by

$$\overline{\sigma_{DOUT}} = \sqrt{\frac{1}{2v_{range}} \int_{-v_{range}}^{v_{range}} \frac{\sigma_{noise}^2}{2N} S^2 dv_{in}}$$

However, if we strobe comparator 2N times in two steps, N times in coarse step and N times in fine step, the accuracy improves for the same 2N number of comparisons. This is shown as follows.

The sum of repeated comparator outputs in the coarse estimate follows a binomial distribution where N is the number of comparator outputs and p is the probability of obtaining a positive comparator decision in each comparator output

$$B(N, p)$$

We define the following probabilities which can be easily calculated from the binomial distribution

$p_{coarse,all\ 1s}$ is probability of getting all 1s in coarse estimate.

$p_{coarse,all\ 0s}$ is probability of getting all 0s in coarse estimate.

$p_{coarse,mid}$ is probability of getting neither all 1s or 0s in coarse estimate.

The proposed two-step resolution enhancement has the following average rms error in the output when we plug in the probabilities of the coarse step outcomes and shift v_{in} in two terms corresponding to the cases when Cap_P/N are flipped in the coarse step and the residue changes

$$\overline{\sigma_{DOUT}} = \sqrt{\frac{1}{2v_{range}} \int_{-v_{range}}^{v_{range}} \left[\begin{array}{c} p_{coarse,all\ 1s} \frac{\sigma_{noise}^2}{N} S(v_{in} - 0.5)^2 \\ + \\ p_{coarse,mid} \frac{\sigma_{noise}^2}{2N} S(v_{in})^2 \\ + \\ p_{coarse,all\ 0s} \frac{\sigma_{noise}^2}{N} S(v_{in} + 0.5)^2 \end{array} \right] dv_{in}}$$

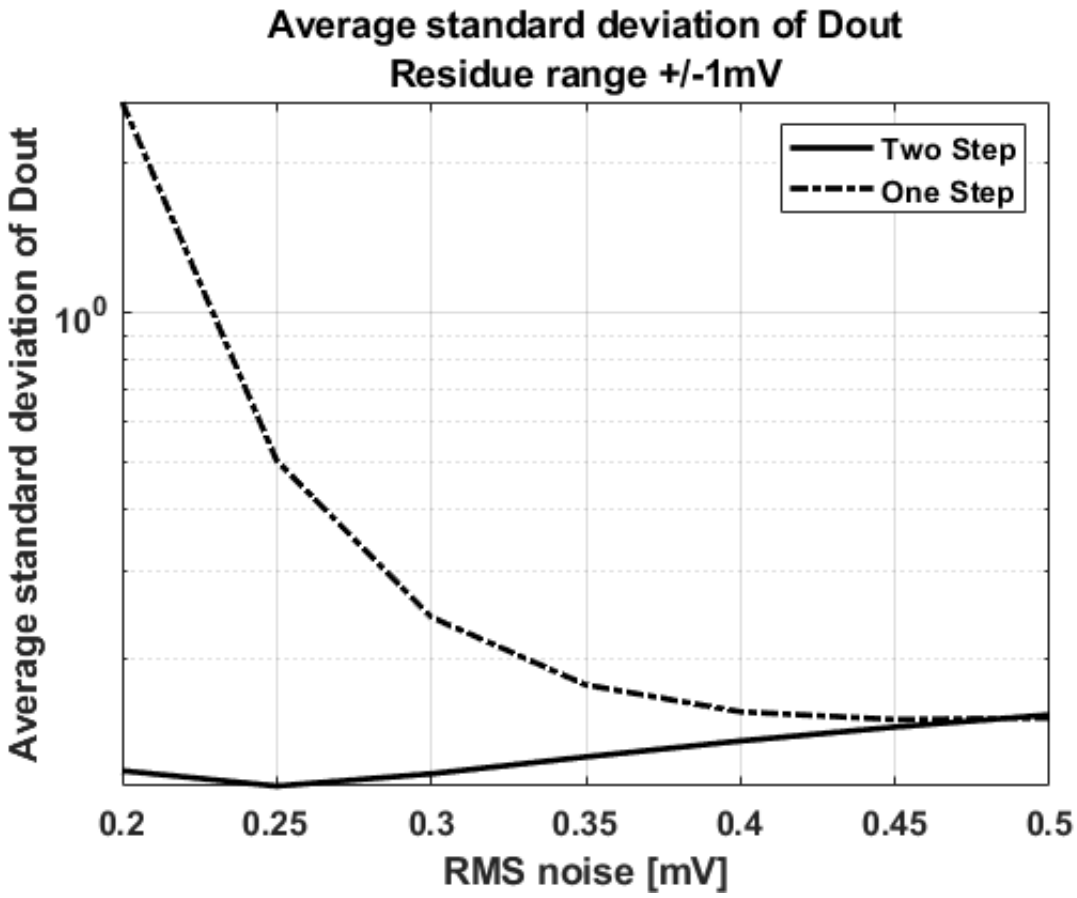


Figure 4-3. Comparison of accuracy versus rms noise of two step and one step.

The plot of the above expression for 2N equal to 32 comparisons and input residue in the range of $-1\mathrm{mV} < \mathrm{v_{in}} < 1\mathrm{mV}$ is shown in Figure 4-3. The proposed two-step achieves better accuracy than one-step for the same number of total comparator comparisons.

Chapter 5

Noise Measurement

The comparator noise is measured to correctly map the comparator output average to the residue at the comparator input for 2B of extra resolution. It is used as a multiplier as discussed in the previous chapter on resolution enhancment. Comparator noise is measured in units of a 10B LSB by averaging comparator outputs with a known reference applied at its input. In the following sections the noise mesurement relations are given followed by the steps taken to measure noise in background during normal A/D conversion.

5.1 Comparator noise and output average relations

Comparator noise σ_{noise} is measured by using the following relation between comparator output average M, comparator noise, and estimated output $\mathrm{D}_{\mathrm{out}}$ as in [25]

$$D_{OUT} = f(M) = \sqrt{2}\sigma_{noise}\text{erf}^{-1}(2M - 1)$$

If the above equation is used with a known input reference of 10B ½ LSB in the presence of an implicit comparator offset and the average of the comparator output MP is related through

$$0.5\text{LSB} + \text{offset} = \sqrt{2}\sigma_{noise}\,\text{erf}^{-1}(2M_P - 1)$$

If the same relation is used with an input reference of a 10B -½ LSB, the average MN is related througth

$$-0.5\text{LSB} + \text{offset} = \sqrt{2}\sigma_{noise}\,\text{erf}^{-1}(2M_N - 1)$$

Subtracting the above two equations, the comparator offset gets cancelled and we get the desired relation

$$1\text{LSB} = \sqrt{2}\sigma_{noise}\{\text{erf}^{-1}(2M_P - 1) - \text{erf}^{-1}(2M_N - 1)\}$$

This equation can be rearranged to measure noise in units of LSBs as

$$\sqrt{2}\sigma_{noise} = \frac{1}{\{\text{erf}^{-1}(2M_P - 1) - \text{erf}^{-1}(2M_N - 1)\}}$$

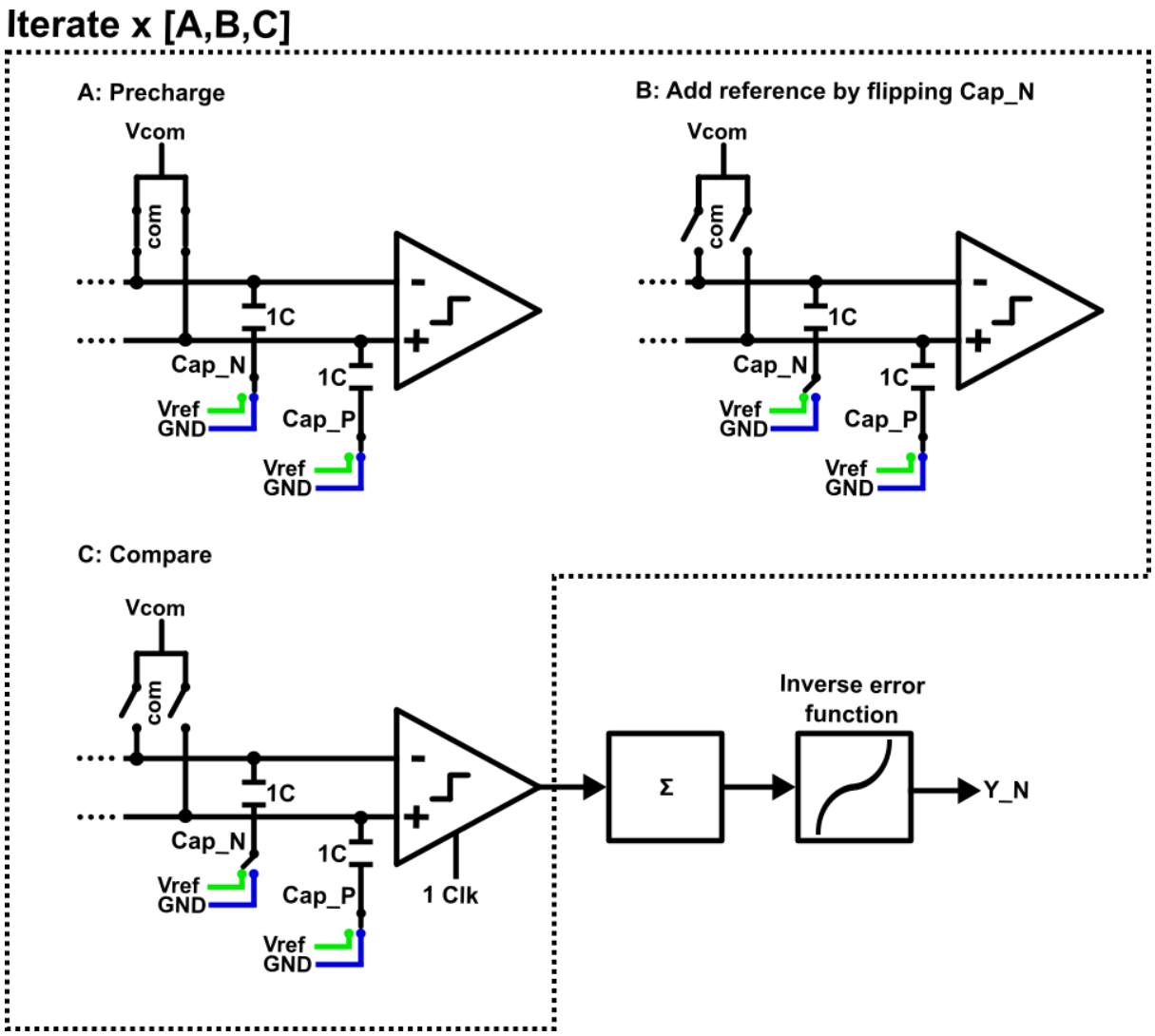


Figure 5-1 Iterations to measure Y_N using negative reference.

5.2 Steps for noise measurement

The implementation details of the steps taken to measure the averages MN and MP and subsequenty using them to measure noise are as shown in Figure 5-1 and Figure 5-2. The noise measurement begins by first precharging the DAC as in step A of Figure 5-1.

The bottom plates of Cap_N and Cap_P are connected to GND and top plate of DAC is connected to Vcom. After that Cap_N is flipped on the N side of the DAC to effectively subtract a 10B ½ LSB reference as in step B of Figure 5-1. Thereafter a comparison is made in step C and comparator output is stored. Steps A,B, and C are repeated once each time the 10-bit A/D conversion ends early and presents an opportunity to do the steps in background. The steps are iterated many times and comparator output results are accumulated. The accumulated resultant average is passed through an inverse error function implemented as a look up table and the output is stored as Y_N

$$\mathrm{Y\_N} = \mathrm{erf}^{-1}(2\mathrm{M_P} - 1)$$

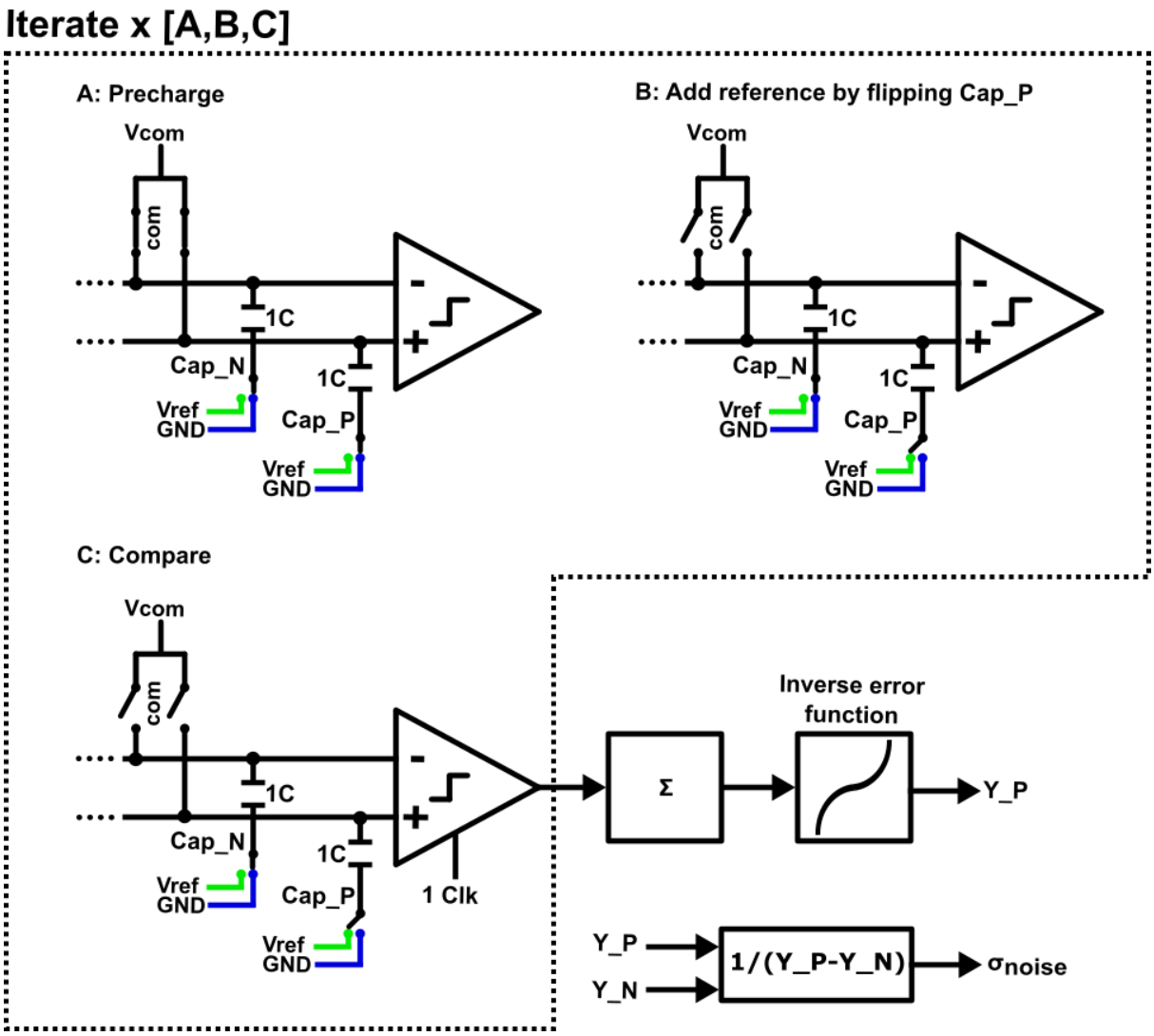


Figure 5-2. Iterations to measure Y_P using positive reference.

Steps A,B, and C are repeated again but this time by flipping Cap_P on the P side of the DAC to effectively add a 10B ½ LSB weighted reference as in step B of Figure 5-2. Thereafter a comparison is made and output is stored as in step C. Step A,B, and C with this positive reference are repeated once each time the 10B A/D conversion ends early and presents an opportunity. The comparator outputs results are accumlated and

the resultant average is passed through an inverse error function look up table and the output is stored as Y_P

$$\mathrm{Y\_P} = \mathrm{erf}^{-1}(2\mathrm{M_M} - 1)$$

Next Y_P and Y_N are subtracted and the result is reciprocated using another stored lookup table to arrive at the desired result

$$\sqrt{2}\sigma_{\mathrm{noise}} = \frac{1\mathrm{LSB}}{\mathrm{Y_P} - \mathrm{Y_N}}$$

The resultant noise multiplier is registered and used to scale during resolution enhancement. Since noise is measured in background it can continuously track changes in temperature or process variations.

Chapter 6

Split-DAC calibration

This work uses the self-calibration technique in [26] to calibrate full-bit and half-bit mismatches for the top 5 MSBs of the split-DAC utilizing spare bit-cycles created by RLSB-first algorithm. This technique requires no gold-standard reference to calibrate the capactive mismatches of the DAC. In the first section general variables for voltages and capacitances are defined in terms of the nominal values and capacitance mismatch errors. In the next section the relations in [26] are adapted to the split-DAC topology used in this work. It is followed by a section each on the sequence of steps taken for full-bit and half-bit residual voltages measurements. The last section describes the calibration loop for many such measurements and final calculation of the error voltages from the residual voltages.

6.1 Error voltage definition and relations

Each capacitor in the 10B split-DAC is assumed to be off by a factor of $(1+\epsilon)$ from its ideal value

$$C_{n,a} = 2^{n-1}C(1+\epsilon_{n,a}), \qquad n = 1,2,3,\ldots..9 \tag{1}$$

$$C_{n,b} = 2^{n-1}C(1+\epsilon_{n,b}), \qquad n = 1,2,3,\ldots..9 \tag{2}$$

$$C_{0,a} = C(1+\epsilon_{0,a}) \tag{3}$$

$$C_{0,b} = C(1+\epsilon_{0,b}) \tag{4}$$

Define the total capacitance as

$$C_{total} = \sum_{0}^{9}(C_{i,a} + C_{i,b})$$

The unit capacitance is defined as

$$C = \frac{C_{total}}{2^N} = \frac{C}{2^N}\left[(1+\epsilon_{0,a}) + (1+\epsilon_{0,b}) + \sum_{1}^{9}\left(2^{n-1}(1+\epsilon_{n,a}) + 2^{n-1}(1+\epsilon_{n,b})\right)\right]$$

$$C = \frac{C}{2^N}\left[1 + 1 + \sum_{1}^{9}(2^{n-1} + 2^{n-1}) + \epsilon_{0,a} + \epsilon_{0,b} + \sum_{1}^{9} 2^{n-1}(\epsilon_{n,a} + \epsilon_{n,b})\right]$$

$$C = C + \frac{C}{2^N}\left[\epsilon_{0,a} + \epsilon_{0,b} + \sum_{1}^{9} 2^{n-1}(\epsilon_{n,a} + \epsilon_{n,b})\right]$$

This gives us the following relation as in [26] applied to our case of split-DAC

$$\epsilon_{0,a} + \epsilon_{0,b} + \sum_{1}^{9} 2^{n-1}(\epsilon_{n,a} + \epsilon_{n,b}) = 0 \quad (5)$$

The output voltage in terms of digital code and capacitor value is

$$V_o = \frac{V_{ref}}{C_{total}} \sum_{0}^{9} C_i D_i$$

$$V_o = \frac{V_{ref}}{2^N\left[C + \frac{C}{2^N}\left[\epsilon_{0,a} + \epsilon_{0,b} + \sum_1^9 2^{n-1}(\epsilon_{n,a} + \epsilon_{n,b})\right]\right]} C\left[(1 + \epsilon_{0,a})D_{0,a} + (1 + \epsilon_{0,b})D_{0,b} + \sum_{1}^{9}\left(2^{n-1}(1 + \epsilon_{n,a})D_{i,a} + 2^{n-1}(1 + \epsilon_{n,b})D_{i,b}\right)\right]$$

Using (5) it becomes

$$V_o = \frac{V_{ref}}{2^N}\left[(1+\epsilon_{0,a})D_{0,a} + (1+\epsilon_{0,b})D_{0,b} + \sum_{1}^{9}\left(2^{n-1}(1+\epsilon_{n,a})D_{i,a} + 2^{n-1}(1+\epsilon_{n,b})D_{i,b}\right)\right]$$

Setting all epsilons to zero gives us the ideal output voltage

$$V_{o,ideal} = \frac{V_{ref}}{2^N}\left[D_{0,a} + D_{0,b} + \sum_{1}^{9}\left(2^{n-1}D_{i,a} + 2^{n-1}D_{i,b}\right)\right]$$

The difference between ideal and actual voltages, the error voltage is

$$V_{error} = V_o - V_{o,ideal}$$

$$V_{error} = \frac{V_{ref}}{2^N}\left[\epsilon_{0,a}D_{0,a} + \epsilon_{0,b}D_{0,b} + \sum_{1}^{9}\left(2^{n-1}\epsilon_{n,a}D_{i,a} + 2^{n-1}\epsilon_{n,b}D_{i,b}\right)\right] \quad (6)$$

Define the error voltage due to the nth capacitor mismatch as

$$V_{\epsilon,n,a} = \frac{V_{ref}}{2^N}2^{n-1}\epsilon_{n,a}, \qquad n = 1,2,3,\ldots.9$$

$$V_{\epsilon,n,b} = \frac{V_{ref}}{2^N}2^{n-1}\epsilon_{n,b}, \qquad n = 1,2,3,\ldots.9$$

$$V_{\epsilon,0,a} = \frac{V_{ref}}{2^N}\epsilon_{0,a}$$

$$V_{\epsilon,0,b} = \frac{V_{ref}}{2^N} \epsilon_{0,b}$$

The total error votlage in equation (6) becomes

$$V_{error} = \frac{V_{ref}}{2^N} \sum_{0}^{9} \left(V_{\epsilon,n,a} D_{n,a} + V_{\epsilon,n,b} D_{n,b} \right)$$

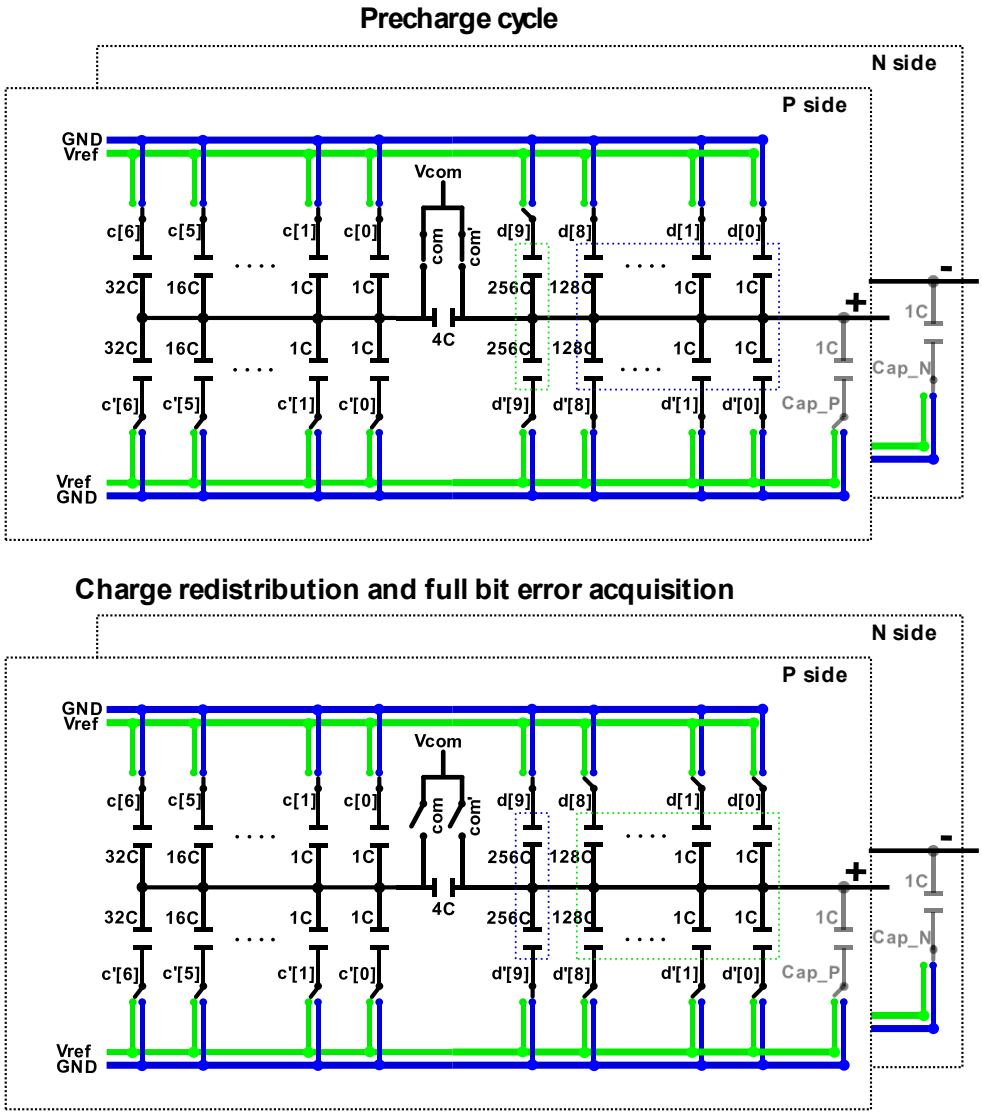


Figure 6-1. Pre-charge and charge distribution steps for one iteration of full bit calibration.

6.2 Full-bit residual voltage acquisition steps

The calibration begins by acquiring full MSB mismatch between MSB and all other capacitors in terms of residual charge. In Figure 6-1, the 10B split-DAC to the right of the VCOM switch is the main DAC being caliberated. The 7B DAC to the left of the VCOM

switch is the DAC used for calibration. In the first precharge step, Vcom-Vref is sampled on full MSB and Vcom is sampled on all other capacitors as show in Figure 6-1. The sampled charge in the capacitor array becomes

$$Q_{precharge} = (V_{COM} - V_{ref})(C_{n,a} + C_{n,b}) + V_{COM}\sum_{0}^{n-1}(C_{i,a} + C_{i,b})$$

The switch configuration is reversed, as show in Figure 6-1, leading to charge redistribution and mismatch error acquisition in terms of residual charge as

$$Q_{x,n} = V_{ref}\left[C_{n,a} + C_{n,b} - \sum_{0}^{n-1}(C_{i,a} + C_{i,b})\right] \quad (7)$$

A successive approximation search is carried out using the calibration-DAC to measure this residual charge and finally calculate the error voltages. This is further discussed in a separate section later.

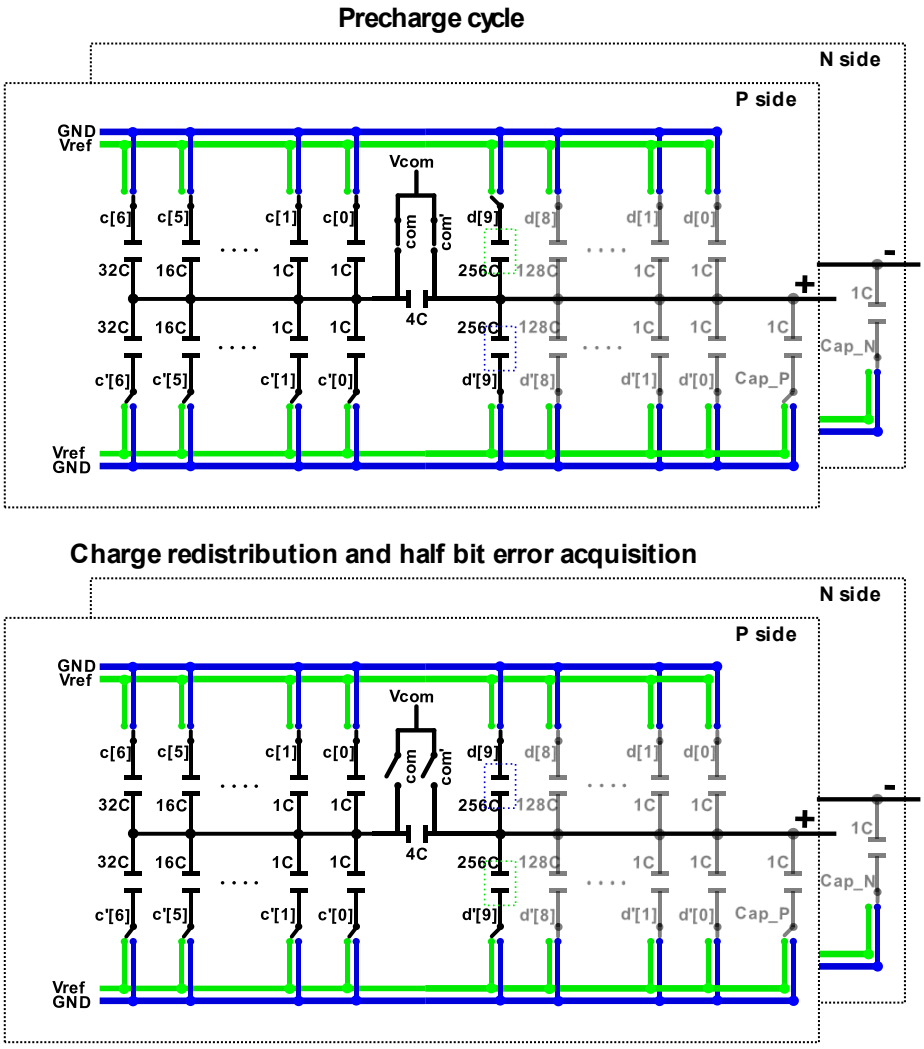


Figure 6-2. Pre-charge and charge distribution steps for one iteration of half bit calibration.

6.3 Half-bit residual voltage acquisition steps

Using a similar technique used for full-bit mismatch measurement, mismatch between two halves of the MSB in the split-DAC is measured in terms of residual charge. In the first precharge step Vcom-Vref is sampled on one half of the MSB and while Vcom is

sampled on all other capacitors as shown in Figure 6-2. The sampled charge in the capacitor array becomes

$$Q_{precharge} = (V_{COM} - V_{ref})C_{n,a} + V_{COM}C_{n,b} + V_{COM}\sum_{0}^{n-1}(C_{i,a} + C_{i,b})$$

The switch configuration between the two halves of the MSB is reversed, as shown in Figure 5-2, leading to charge redistribution and half-bit mismatch error acquisition in terms of residual charge as

$$Q_{x,n,a} = V_{ref}[C_{n,a} - C_{n,b}] \quad (8)$$

A successive approximation search is carried out using the calibration-DAC to measure this residual charge and finally calculate the error voltages. This is further discussed in a separate section to follow.

6.4 Calibration loop and error voltage calculation

Full-bit and half-bit residual voltages are acquired and measured for the top five bits of the main-DAC. Residual voltages acquired through the above technique of pre-charge and charge re-distribution are measured through successive approximation search using the 7B calibration-DAC. The calibration-DAC is coupled to the main-DAC through a bridge-capacitor of four units capacitor size as shown in Figure 6-1 and 6-2. To do a full SAR search in background while the ADC is operational, only one bit on calibration-DAC

is evaluated each time an opportunity is presented when the ADC 10B conversion ends early. Incomplete SAR search calibration bits are saved in between ADC conversion cycles. When next opportunity to resume calibration is presented, the incomplete SAR search bits are reloaded after charge re-distribution step and the next bit in the search is evaluated and so on until the search is completed in the background over many A/D conversion cycles. Many such complete SAR searches are done and averaged for each of the top five full-bit and half-bit residual voltages. The average residual voltages are used to calculate the error voltages through the following relations.

Substituting (1-5) in (7), and simplifying we get

$$Q_{x,n} = CV_{ref}\left[2^{n-1}\varepsilon_{n,a} + 2^{n-1}\varepsilon_{n,b} - \sum_{i=n}^{i=9}\left(2^{i-1}\varepsilon_{i,a} + 2^{i-1}\varepsilon_{i,b}\right)\right]$$

$$\frac{Q_{x,n}}{2^N C} = \frac{V_{ref}}{2^N}\left[2^{n-1}\varepsilon_{n,a} + 2^{n-1}\varepsilon_{n,b} - \sum_{i=n}^{i=9}\left(2^{i-1}\varepsilon_{i,a} + 2^{i-1}\varepsilon_{i,b}\right)\right]$$

$$V_{x,n} = \left(V_{\varepsilon,n,a} + V_{\varepsilon,n,b}\right) - \sum_{i=n}^{i=9}\left(V_{\varepsilon,i,a} + V_{\varepsilon,i,b}\right)$$

$$\frac{V_{x,n}}{2} = \left(V_{\varepsilon,n,a} + V_{\varepsilon,n,b}\right) - \frac{1}{2}\sum_{i=n+1}^{i=9}\left(V_{\varepsilon,i,a} + V_{\varepsilon,i,b}\right)$$

$$\left(V_{\varepsilon,n,a} + V_{\varepsilon,n,b}\right) = \frac{1}{2}\left[V_{x,n} - \sum_{i=n+1}^{i=9}\left(V_{\varepsilon,i,a} + V_{\varepsilon,i,b}\right)\right] \quad (9)$$

Substituting (1-4) in (8), and simplifying we get

$$Q_{x,n,a} = CV_{ref}\left[2^{n-1}\varepsilon_{n,a} - 2^{n-1}\varepsilon_{n,b}\right]$$

$$\frac{Q_{x,n,a}}{2^N C} = \frac{V_{ref}}{2^N}\left[2^{n-1}\varepsilon_{n,a} - 2^{n-1}\varepsilon_{n,b}\right]$$

$$\left(V_{\varepsilon,n,a} - V_{\varepsilon,n,b}\right) = V_{x,n,a} \qquad (10)$$

Equations (9) and (10) can be solved to get the general relation between residual voltages and error voltages

$$V_{\varepsilon,n,a} = \frac{V_{x,n,a}}{2} + \frac{V_{x,n}}{4} - \frac{\sum_{i=n+1}^{i=9}\left(V_{\varepsilon,i,a} + V_{\varepsilon,i,b}\right)}{4}$$

$$V_{\varepsilon,n,b} = -\frac{V_{x,n,a}}{2} + \frac{V_{x,n}}{4} - \frac{\sum_{i=n+1}^{i=9}\left(V_{\varepsilon,i,a} + V_{\varepsilon,i,b}\right)}{4}$$

The codes corresponding to the error voltages are stored for use during A/D conversion when the calibration is enabled. The error voltage codes corresponding to the bits set to 1 on the main-DAC are added and applied to the calibration-DAC to correct for capacitive mismatches in the main-DAC.

Chapter 7

Design details

This chapter discusses the technique and choices made in the design of important components of the A/D converter. First, the two blocks used for recoding, Encode and Decode, are discussed in detail. It is followed by a discussion on split-DAC unit capacitor size selection and its layout. In the end is a discussion on comparator and bootstrap switch circuit.

7.1 Recoding algorithm

The recoding in the RSLB-first algorithm consist of encoding and decoding that is done by the two blocks, Encode and Decode, respectively. After a 10B conversion ends, the ADC output code is in three possible weights, 0,1, and ½. This is first decoded into a code in two weights, 0 and 1, using the Decode block. Before the start of next ADC conversion cycle, the output of the Decode block is encoded in terms of 0, 1, and ½ using the Encode

Table 7-1. Truth table for E[n] and T[n].

A[n]	B[n]	E[n]	T[n]
0	0	0	0
1	1	1	0
0	1	½	1

block. Since the rules for recoding, as discussed in a separate chapter on RLSB-first, are based on identification and replacement of

repeating adjacent bit values in the input to the two blocks, the approach taken is to design a repeating module, one for each bit location, with the module output dependent on code values in adjacent input bits. In other word, each bit output of the Encode and Decode module is a function of code in adjacent input bits. The Decode block is discussed first followed by the Encode block.

7.2 Decode module

The input code at each bit location, n, in a Decode block are two input signals A[n], B[n] with three code possibilities of 00, 11, or 01 corresponding to the bit weights of 0,1, and ½. It will be useful to define two new signals E[n] and T[n] at each bit location to understand the decode block easily. T[n] is XOR of A[n] and B[n]. E[n] is only defined for convenience and uses the three weights 0, 1, and ½ so we don't have to use two signals

A[n] and B[n] each time referring to input bit value. The truth table for E[n] and T[n] is shown in Table 7-1.

For the Decode module for each bit, let us understand the main functions that the connected adjacent modules should perform to implement the recoding rules of RLSB-first. If the current bit input to the module is a 1 or 0 followed by ½s in adjacent higher bits, then this bit value is passed on to all higher modules with consecutive ½s until next 1 or 0 input is encountered. The bit being passed on is output at the last location with a ½ while an inversion of it is output at all other locations it is passed. This becomes clear with an example decoding of 00½½½1 which after decoding should be 001000. In 00½½½1, one is followed by three ½s to the left and it is passed on until a 0 input is encountered to its left. It is output at the last location it is passed and inversion of it, that is 0, is output at all other location that it is passed on.

A single decode module is shown in Figure 7-1. Here P[n] is the passed-on bit and takes the value of the bit being decoded if E[n] is 1 or 0, that is if A[n] and B[n] are both equal 1 or both equal 0. If E[n] is ½, that is A[n] and B[n] are 0 and 1 respectively, T[n] takes the value of 1 which in turn sets P[n] equal to P[n-1] passed on to it from a module to the right on the LSB side. The decoded output bit D[n] is set to P[n] if the next bit to be decoded to the left on the MSB side is 0 or 1, that is T[n+1] is 0. Otherwise the decoded bit D[n] is set to ~P[n] if the next bit to be decoded to the left is a ½. These logical relations between T, P, and D are clearly tabulated in Table 7-2.

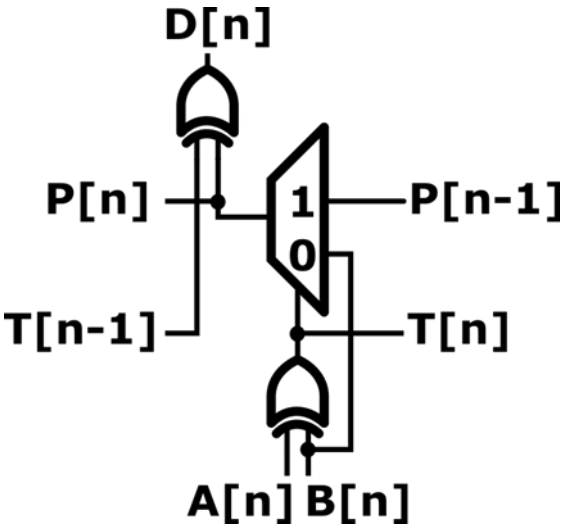


Figure 7-1. Decode module logic.

Table 7-2. Truth table for P[n] and D[n].

T[n+1]	T[n]	P[n]	D[n]
0	0	E[n]	E[n]
1	0	E[n]	~E[n]
0	1	P[n-1]	P[n-1]
1	1	P[n-1]	~P[n-1]

The Decode module dependencies of output signals on input signals are shown in Figure 7-2 for the fifth module location as an example. The blue squares are module inputs and green squares are module outputs. The boundary input values to modules at bit location 0 and 9 are also given to satisfy the RLSB-first recoding rules. A complete Decode example with filled in table values is as shown in Figure 7-3. Decode modules connected together are shown in Figure 7-4.

n		9	8	7	6	5	4	3	2	1	0	
Input, E												
T	0											
P												0
Output, D												

Figure 7-2. Dependencies of decode module.

n		9	8	7	6	5	4	3	2	1	0	
Input, E		0	½	½	0	1	½	½	1	0	1	
T	0	0	1	1	0	0	1	1	0	0	0	
P		0	0	0	0	1	1	1	1	0	1	0
Output, D		0	0	1	1	1	1	0	0	0	1	

Figure 7-3. Decode example.

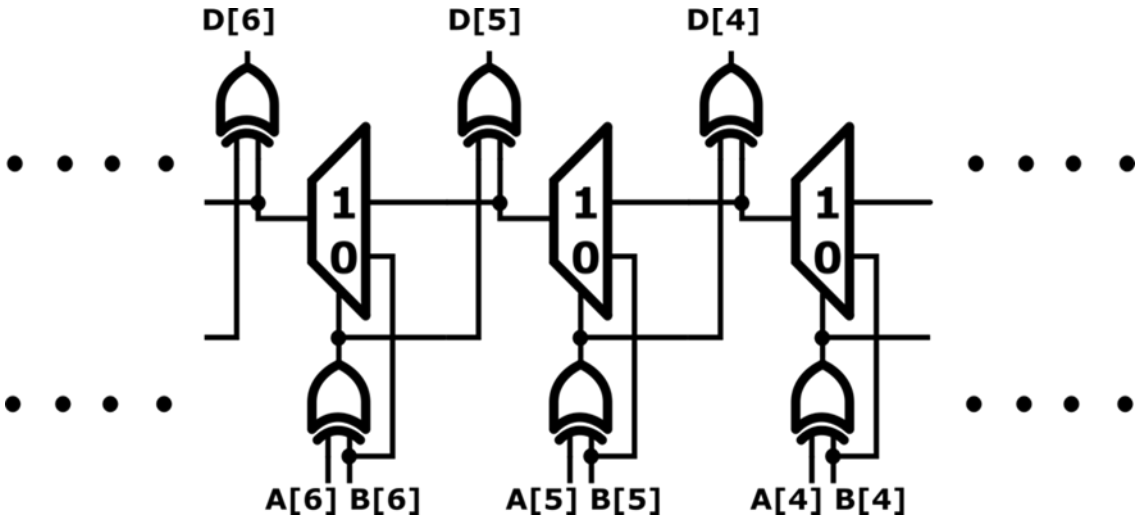


Figure 7-4. Decode modules connected together.

7.3 Encode module

The input code at each bit location n to an Encode block is the output of the decode block D[n] in terms of the two weights, 0 or 1. The output code at each bit location of the Encode block is one of the three possible codes 0,1, or ½. It will be useful to define three new signals R[n], S[n], and C[n] at each bit location to understand the encode block easily.

R[n] is 1 if the input D[n] is not the same as the output E[n] at bit location n. In other words, R[n] is 1 whenever output E[n] is generated by recoding input D[n] either by inverting D[n] or outputting a ½, which are only two possibilities to recode D[n]. Else, R[n] is 0 if E[n] is simply set to D[n], that is not recoded and output as it is. The truth table for R is shown in Table 7-4.

S[n] is 1 when D[n] and D[n-1] are same, that is input at bit location n is the same as input at adjacent bit location n-1 to the right. Otherwise S[n] is 0. The truth table for S is shown in Table 7-5.

C[n] sets the upper bit location beyond which no encoding is performed and the output bit E[n] is simply set to the input D[n] beyond that bit location. The truth table for C is shown in Table 7-3.

Table 7-3. Truth table for C[n].

C[n+1]	S[n]	C[n]
0	0	1
0	1	0
1	X	1

Table 7-4. Truth table for R[n].

D[n]	E[n]	R[n]
0	0	0
0	1	1
0	½	1
1	0	1
1	1	0
1	½	1

Table 7-5. Truth table for S[n].

D[n]	D[n-1]	S[n]
0	0	1
0	1	0
1	0	0
1	1	1

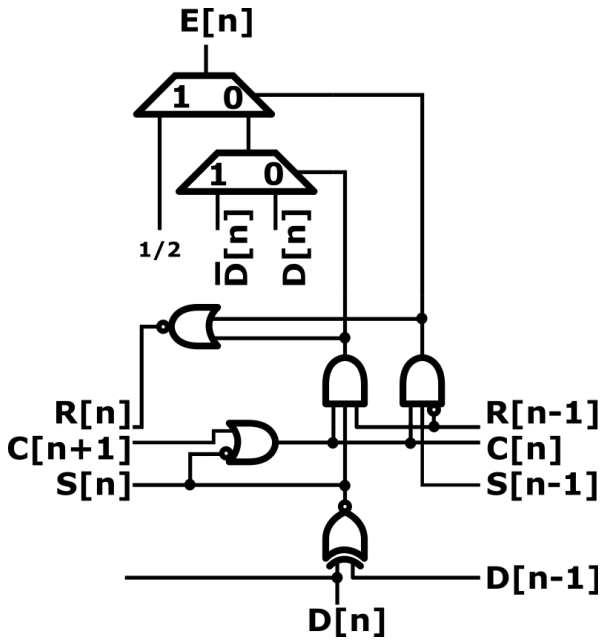


Figure 7-5. Encoding module.

A single encode module is shown in Figure 7-5. To understand the Encode module, let us find the key dependencies of the output E[n] on the derived signals C, R, and S that satisfy the recoding rules of RLSB-first. If the adjacent bit on the LSB side gets recoded (i.e. R[n-1] is 1) then the Encoding bit output for the current location E[n] depends on S[n]. In this case E[n] is the same as D[n] if S[n] is 0 and ~D[n] if S[n] is 1. If the adjacent bit on the LSB side does not get recoded (i.e. R[n-1] is 0) then the Encoding bit output for the current location E[n] depends on S[n-1]. In this case E[n] is set to D[n] if S[n-1] is 0 and ½ if S[n-1] is 1. These rules are overridden if C[n] is 0 in which case no recoding takes place and E[n] is set to D[n] for that bit location. These are tabulated in Table 7-6.

Table 7-6. Truth table for E[n].

C[n]	R[n-1]	S[n]	S[n-1]	E[n]
0	x	x	x	D[n]
1	0	1	x	~D[n]
1	0	0	x	D[n]
1	1	x	1	½
1	1	x	0	D[n]

The Encode module dependencies of output signals on input signals are shown in Figure 7-6 for the fifth module location as an example. The blue squares are module inputs and green squares are module outputs. The boundary input values to modules at bit location 0 and 9 are also given to satisfy the RLSB-first recoding rules. A complete Encode example with filled in table values is as shown in Figure 7-7. Encode modules connected together are shown in Figure 7-8.

n		9	8	7	6	5	4	3	2	1	0	
Input, D												~D[0]
C	0											
S												0
R												1
Output, E												

Figure 7-6. Dependencies of Encode module.

n		9	8	7	6	5	4	3	2	1	0	
Input, D		0	0	1	1	1	1	0	0	0	1	~D[0]
C	0	0	1	1	1	1	1	1	1	1	1	
S		1	0	1	1	1	0	1	1	0	0	0
R		0	1	1	1	0	1	1	1	0	0	1
Output, E		0	½	½	0	1	½	½	1	0	1	

Figure 7-7. Encode example.

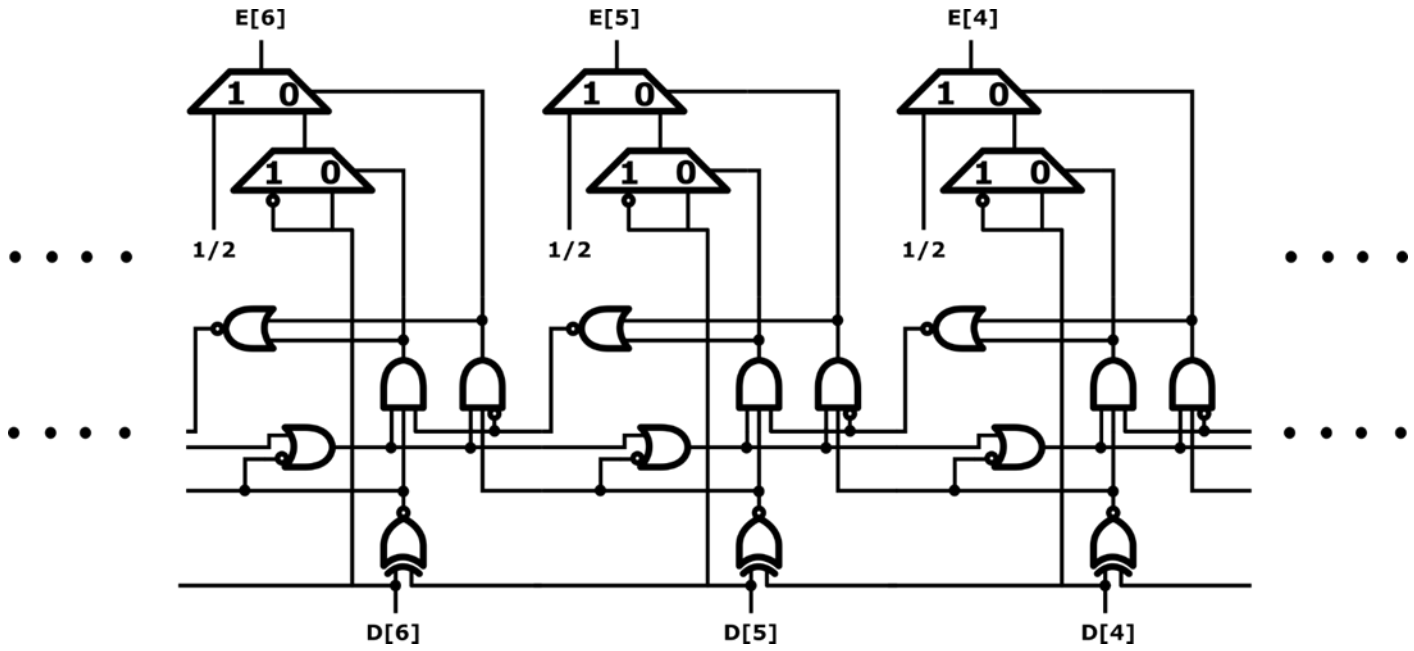


Figure 7-8. Encode modules connected together.

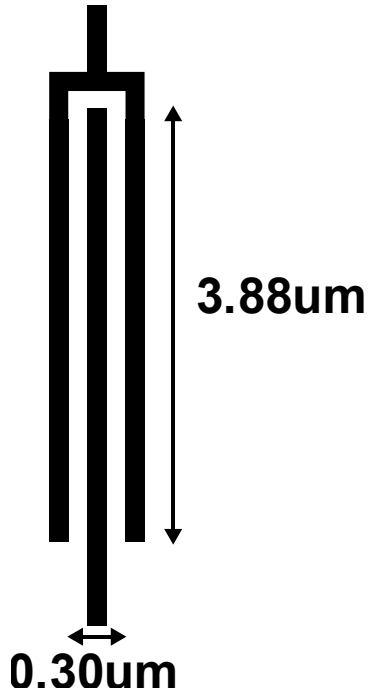


Figure 7-9. Split-DAC unit capacitor layout.

7.4 Split-DAC layout

The split-DAC uses a custom-designed unit capacitor as in [27]. The structure is implemented with metal layers 5,6, and 7 and the top view is as shown in Figure 7-9. The parasitic capacitance from the two nodes to GND is minimized by using higher layers. Capacitor matching requirements are set as in [27]. Let σ_u be standard deviation and μ_u be the nominal value of the unit capacitor. The mismatch of unit capacitor is

$$\frac{\sigma_u}{\mu_u}$$

For the worst-case transition when all unit capacitors in the differential DAC toggle, the worst-case standard deviation σ_{worst_dnl} in terms of unit capacitor σ_u would be

$$\sigma^2_{worst\_dnl} = 2\ 2^B {\sigma_u}^2$$

$$\sigma_{worst\_dnl} = \sqrt{2\ 2^B}\sigma_u$$

For three standard deviations of accuracy better than the differential LSB of $2\mu_u$, the constraint that must be met is

$$3\sigma_{worst\_dnl} < \frac{1}{2} 2\mu_u$$

Simplifying we get the following result for maximum unit capacitor mismatch

$$3\sqrt{2\ 2^B}\ \sigma_u < \mu_u$$

$$\frac{\sigma_u}{\mu_\mathrm{u}} < \frac{1}{3\sqrt{2\ 2^B}}$$

$$\frac{\sigma_u}{\mu_\mathrm{u}} < \frac{1}{3\sqrt{2\ 2^B}} 100\%$$

For a 10B DAC with 10B accuracy the mismatch is 0.74%. For a 10B DAC with 12B accuracy the mismatch is ¼ of the 10B accuracy result and equals 0.18%.

The standard MOM capacitor model available in the technology was simulated to get an estimate of mismatch in the technology. The capacitance per unit area of a parallel place capacitor is equal to dielectric constant divided by spacing. The parallel plate spacing in the designed fringe capacitor was kept the same as in standard capacitors. Since mismatch is inversely proportional to square root of the area [21] we get the following relations

$$\frac{\left(\frac{\sigma}{\mu}\right)_{design}}{\left(\frac{\sigma}{\mu}\right)_{standard}} = \sqrt{\frac{A_{standard}}{A_{design}}} = \sqrt{\frac{C_{standard}}{C_{design}}}$$

Maximum sampled thermal noise that can be tolerated for a 12B resolution A/D conversion gives us the minimum DAC capacitance size. By setting the sampled thermal noise equal to the 12B rms quantization noise we get

$$2\frac{kT}{C} = \frac{LSB^2}{12} = \frac{\left(\frac{1}{2^{12}}\right)^2}{12}$$

Using the above expression C is found to be 1.7pF. The minimum size for the unit capacitor in a 10B size DAC with total capacitance of 1.7pF is 1.6fF.

The mismatch limited unit capacitor size for a 12B accuracy is 7.8fF. The unit capacitor size of 2.2fF was chosen which is closer to the thermal constraint rather than

satisfy the mismatch constraint so as to minimize the DAC capacitance and lower switching energy. This size gives us around 10B accurate DAC with the remaining accuracy to be met with mismatch calibration for a total of 12B target accuracy.

The layout of the binary weighted 10B main DAC and 7B calibration DAC was done using centroid scheme and capacitor placement is as shown in Figure 7-10. The coupling capacitance that connects the calibration DAC to the main DAC is implemented using four unit-capacitors. A and B labels are used for the two halves of each Bit in this split-DAC architecture.

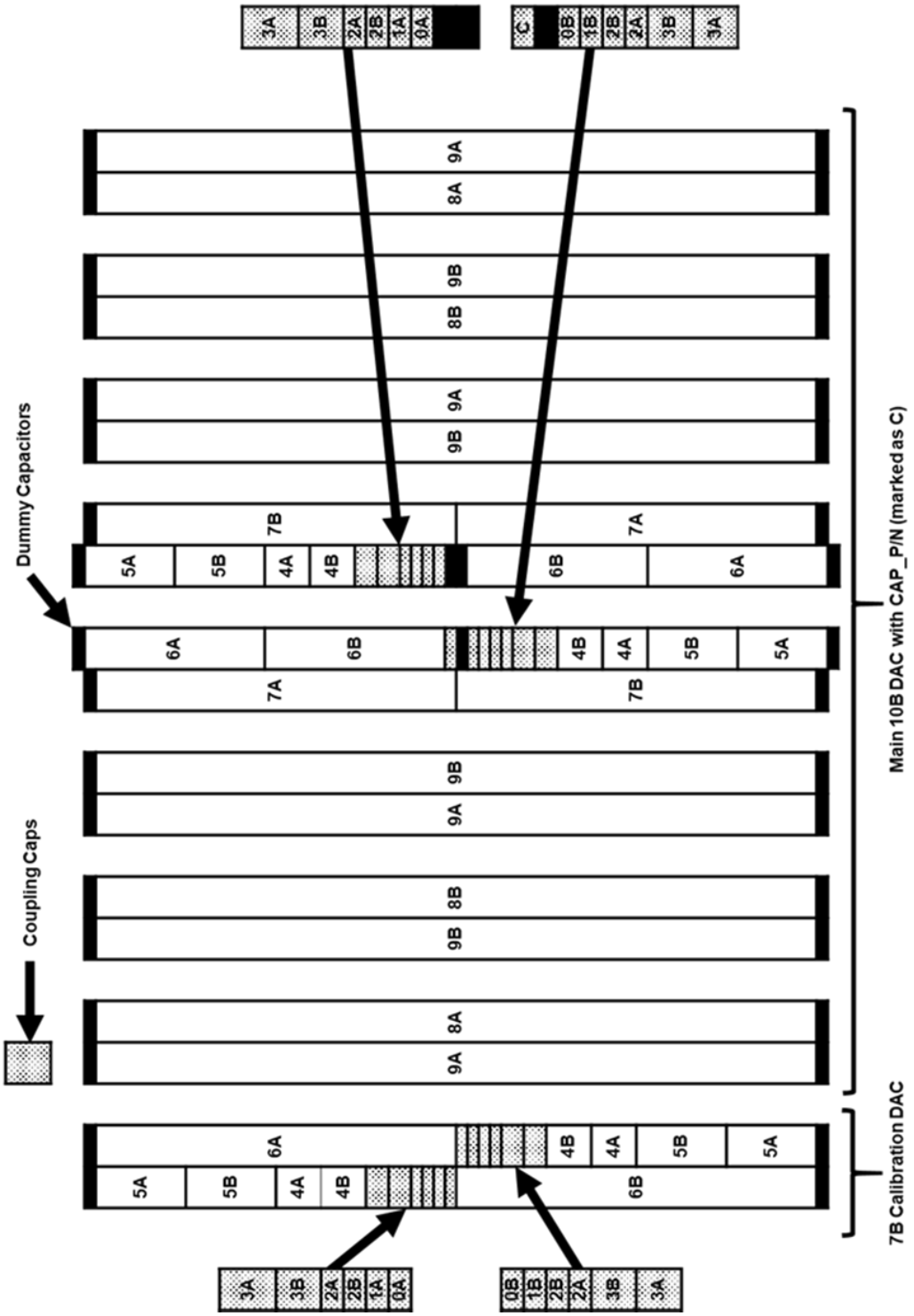


Figure 7-10. Split-DAC layout of single side. Replicated for the differential pair.

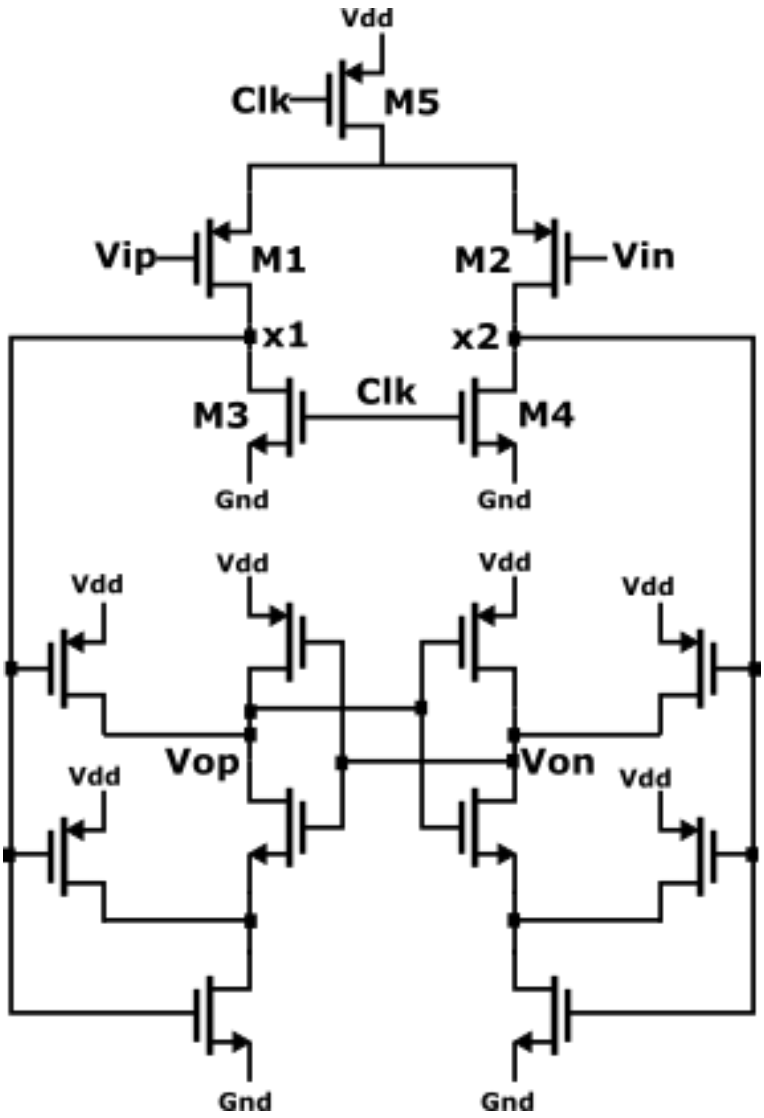


Figure 7-11. Comparator.

7.5 Comparator

The comparator used in this design is Miyahara's as in [28] and shown in Figure 7-11. The comparator comprises two stages. In the reset phase, the comparator nodes x1 and

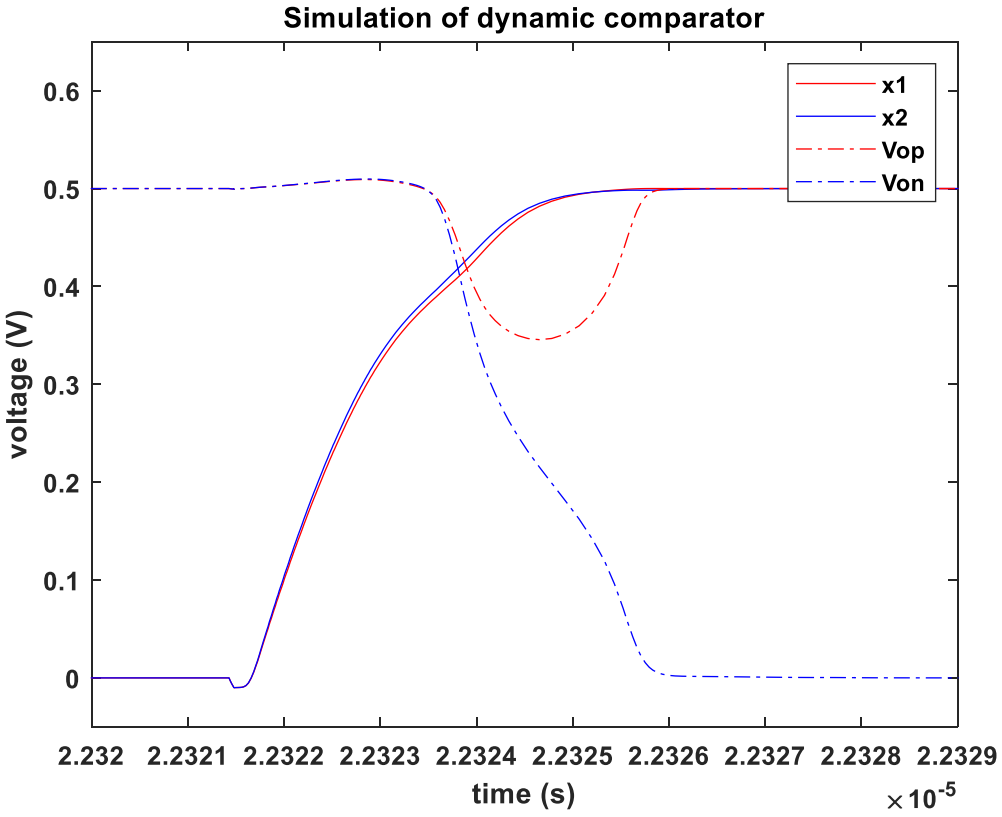


Figure 7-12. Simulation of dynamic comparator.

x2 are discharged to GND by M3 and M4. At the CLK edge M5 turns on and M3 and M4 turn off. In this stage a differential input voltage on top of a common-mode bias at the comparator input generates an input dependent differential current on top of a common-mode current that integrates on capacitances at nodes x1 and x2. The differential input voltage is amplified at nodes x1 and x2 and the common-mode voltage increases. The second stage is a regenerative latch and is self-timed and so the comparator needs only

one edge of the clock to operate. The second stage turns on with the rising common mode voltage on x1 and x2. The simulation of this comparator is shown in Figure 7-11.

Input referred noise of this regenerative comparator can be reduced by maximizing the dynamic gain in the first stage. Using the stochastic model in [10] for the first stage of the comparator with input pair transconductance as g_m, we get an equivalent noise resistance of

$$R_n = \frac{1}{\gamma g_m}$$

where γ is noise factor typically equal to 2/3. The variance of the noise voltage at nodes x1 and x2 in Figure 7-11 is a function of time as

$$E[v^2(t)] = 2\frac{2kT}{C^2 R_n}t + 2\frac{kT}{C}$$

$$E[v^2(t)] = \frac{4kT\gamma}{C^2} g_m t + \frac{2kT}{C}$$

The second term is the sampled noise at time t=0 on an effective capacitance of C at each of the two nodes, x1 and x2, in Figure 7-11. Labelling the dynamic gain of this stage with G we get

$$G = \frac{g_m t}{C}$$

The time t in the above expression is the time for the x1 and x2 nodes to charge up and turn on the second stage. It is approximately equal to

$$t = \frac{C}{I} V_t$$

where V_t is the turn-on threshold voltage for the second stage. The expression for the dynamic gain becomes

$$G = \frac{g_m}{I} V_t$$

where g_m/I is the transconductance efficiency. In above-threshold it is equal to

$$\frac{g_m}{I} = \frac{2}{V_{gs} - V_t}$$

and in sub-threshold it is equal to

$$\frac{g_m}{I} = \frac{q}{nkT}$$

Input referred noise variance due to first stage of comparator becomes

$$\sigma^2_{input} = \frac{E[v^2(t)]}{G^2}$$

$$\sigma^2_{input} = \frac{2kT}{C}\left(\frac{2\gamma}{G} + \frac{1}{G^2}\right)$$

Since dynamic comparator gain is proportional to transconductance efficiency as in equation, to minimize input referred noise, G can be maximized by operating the input transistor pair M1 and M2 in weak inversion. Since t is inversely proportional to the common mode current I, it slows down the first stage of this comparator. However due to fast positive feedback regeneration in second stage, this delay is not a problem in our slow sampling rate application and overall comparator time.

The comparator was designed for input referred noise around 10B quantization noise, $10B\ LSB/\sqrt{12}$ equal to 300uVrms. The comparator noise was measured by doing multiple transient simulation for different comparator input voltage settings and plotting the average of positive comparator outputs versus input voltage.

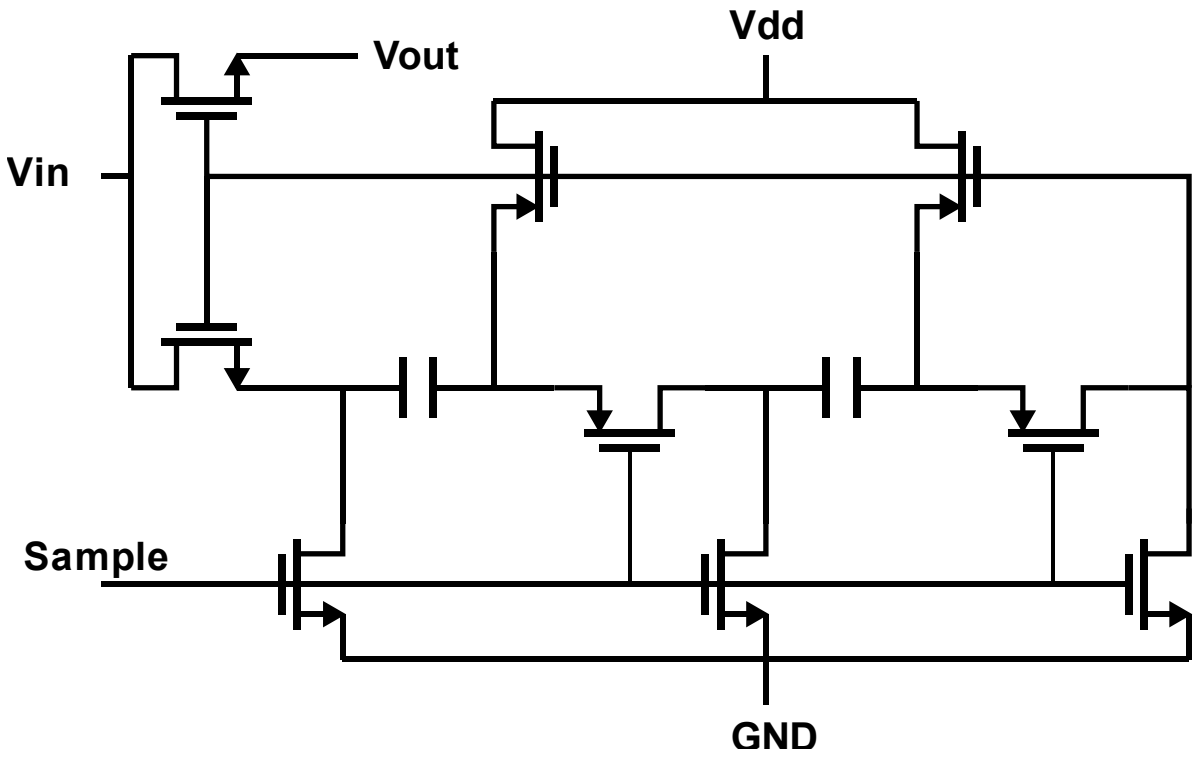


Figure 7-13. Bootstrapping circuit.

7.6 Bootstrapping circuit

The sampling switch causes input dependent turn on resistance contributing to linearity errors. To mitigate this effect the input switches are bootstrapped using the circuit in [29] and shown in Figure 7-12. This removes the input dependent error while sampling by turning on the sampling switch with the same overdrive voltage for all values of input signal.

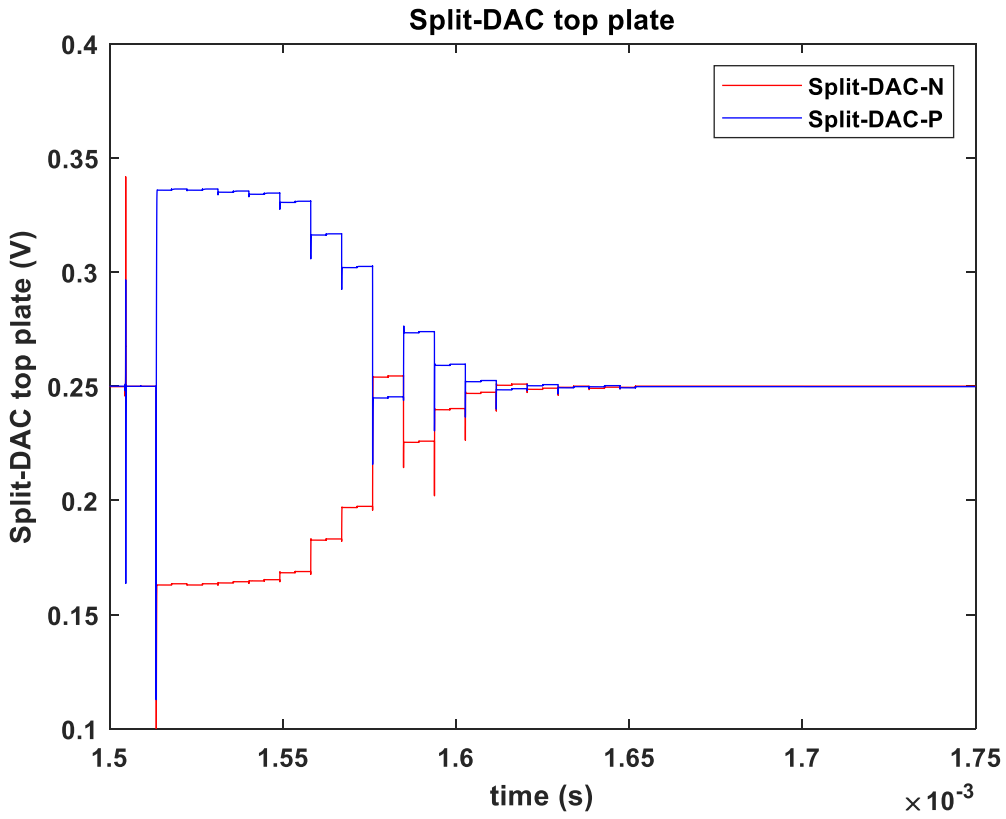


Figure 7-14. Simulation with starting code 716 and input sample of 10mV. Resultant output code is 522.

7.7 Voltage and device choices

For low power operation voltage supply of 0.5V was chosen. The common mode input to the comparator is set at half of the supply at 0.25V. This sets the input transistor operation near subthreshold for maximum transconductance efficiency for low input referred noise and maximum resolution during LSB evaluation.

The digital logic was coded in Verilog and synthesized and placed using GENUS and INNOVUS tools by Cadence. High threshold (High-Vt) devices in the standard library were used during synthesis to reduce leakage currents. The impact on speed could be tolerated due to low sampling frequency of 4KHz. Figure 7-12 shows simulation for voltage signal at the input of the comparator on the top plates of the split-DAC for conversion with previous starting code of 716 and differential input sample of 10mV that results in an output code of 522.

Chapter 8

Measurements

The test setup is as shown in Figure 8-1. It includes a custom designed PCB with a socket for the test chip and a port to connect with Opal Kelly FPGA. The PCB has digital

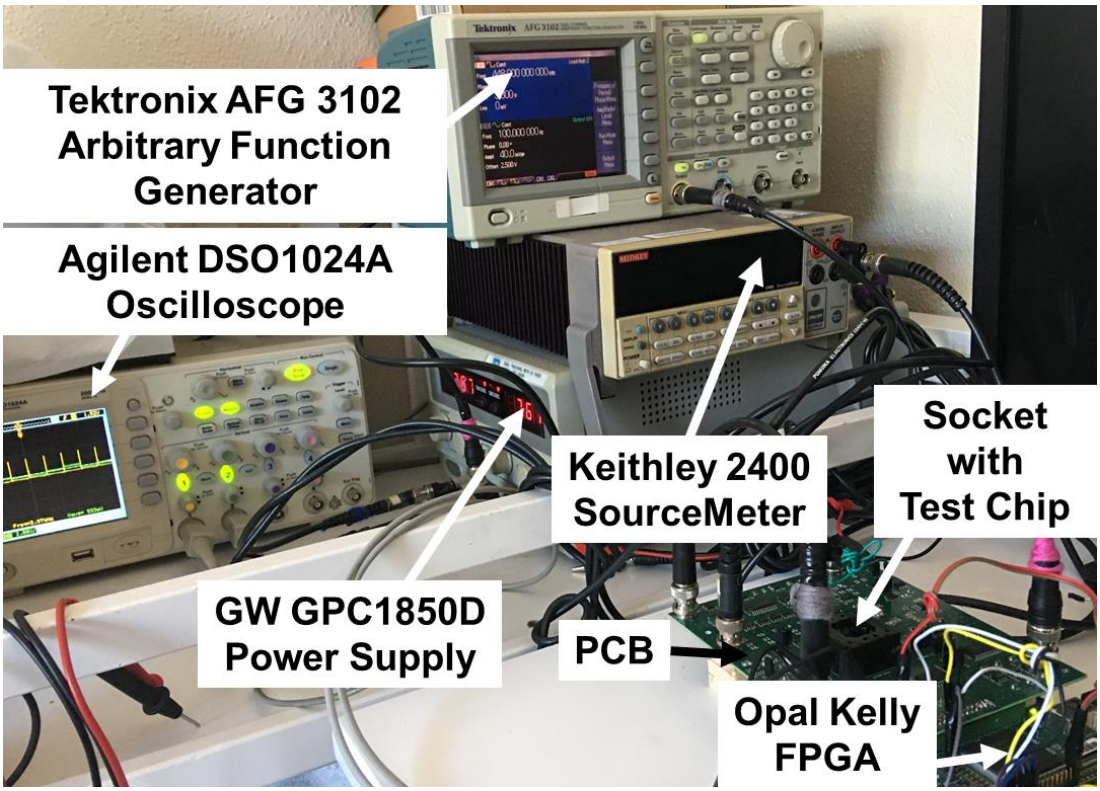


Figure 8-1. Test setup.

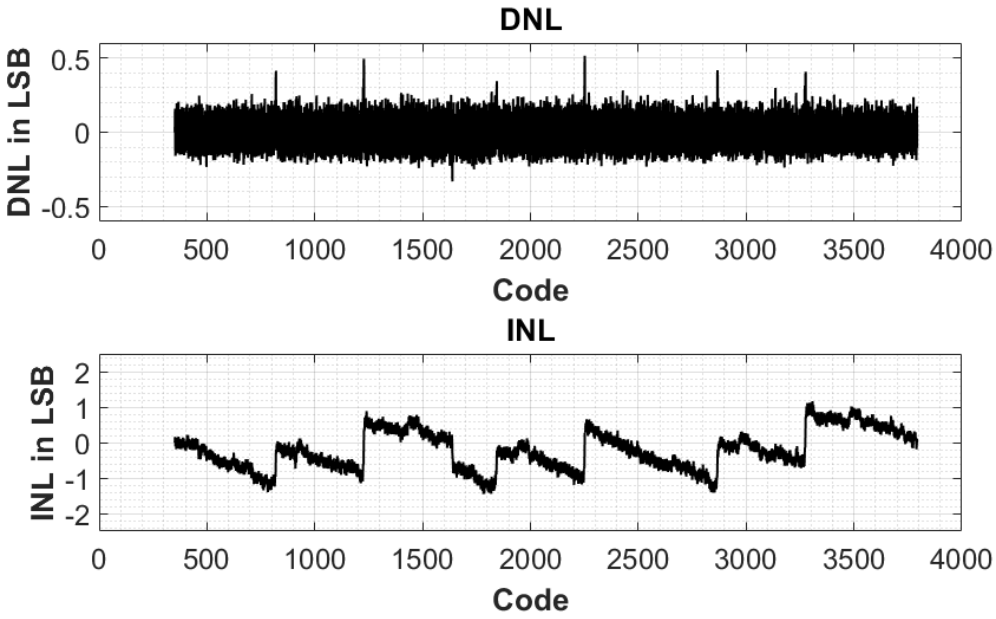


Figure 8-2. DNL and INL without calibration.

potentiometers to accurately set the supply voltages. The chip on the PCB communicates through FPGA to PC for data acquisition. The sinusoidal test input signal was generated using Tektronix Arbitrary Function Generator.

The technique in [35] was used to measure INL and DNL. With the calibration turned off the DNL is measured to be under +/- 0.5 LSB and INL is measured to be around +/-1 LSB at a 12B accuracy as shown in Figure 8-2. However, with calibration turned on the DNL/INL degrades as shown in Figure 8-3. The problem was suspected to be with the residual offset that was not cancelled during residual voltage measurements. To confirm this, the stored error voltages were read out after DAC calibration looped through

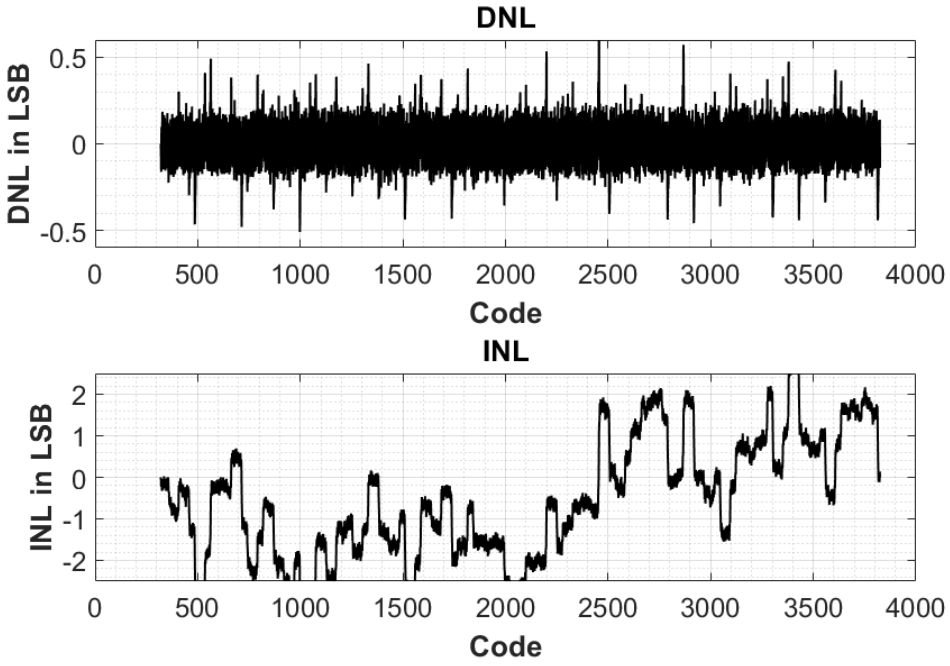


Figure 8-3. DNL and INL with calibration and before offset correction.

full calibration routine. Using the relations between the measured residual voltages and error voltages from the previous chapter on split-DAC calibration

$$V_{\varepsilon,n,a} = \frac{V_{x,n,a}}{2} + \frac{V_{x,n}}{4} - \frac{\sum_{i=n+1}^{i=9}\left(V_{\varepsilon,i,a} + V_{\varepsilon,i,b}\right)}{4}$$

$$V_{\varepsilon,n,b} = -\frac{V_{x,n,a}}{2} + \frac{V_{x,n}}{4} - \frac{\sum_{i=n+1}^{i=9}\left(V_{\varepsilon,i,a} + V_{\varepsilon,i,b}\right)}{4}$$

and assuming that the comparator offset is a constant OFF_{meas} during each residual voltage measurement, the following relations are derived for the errors $OFF_{n,a}$ and $OFF_{n,b}$ in the measured error voltages

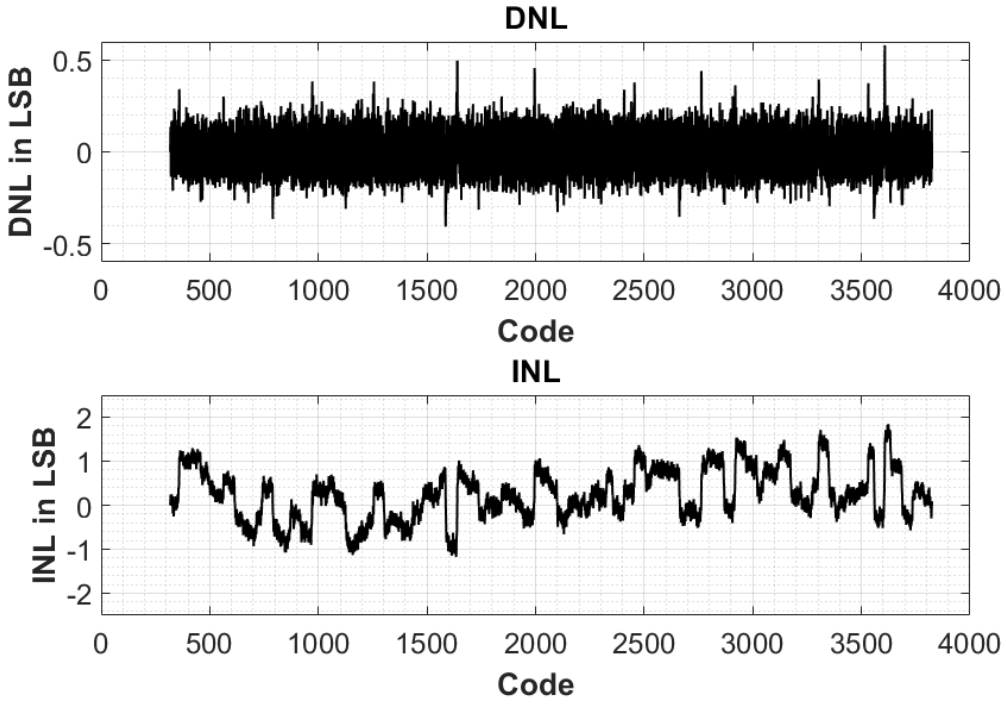


Figure 8-4. DNL and INL with calibration and after offset correction.

$$\mathrm{OFF_{n,a}} = \frac{3}{4}\mathrm{OFF_{meas}} - \frac{1}{4}\sum_{i=n+1}^{i=N}(\mathrm{OFF_{i,a}} + \mathrm{OFF_{i,b}});$$

$$\mathrm{OFF_{n,b}} = -\frac{3}{4}\mathrm{OFF_{meas}} - \frac{1}{4}\sum_{i=n+1}^{i=N}(\mathrm{OFF_{i,a}} + \mathrm{OFF_{i,b}});$$

The comparator offset $\mathrm{OFF_{meas}}$ was measured and used to calculate the correction errors $\mathrm{OFF_{n,a}}$ and $\mathrm{OFF_{n,b}}$ by substituting them in the above equations. These error offsets were subtracted out from the error voltages and the new error voltages were reloaded

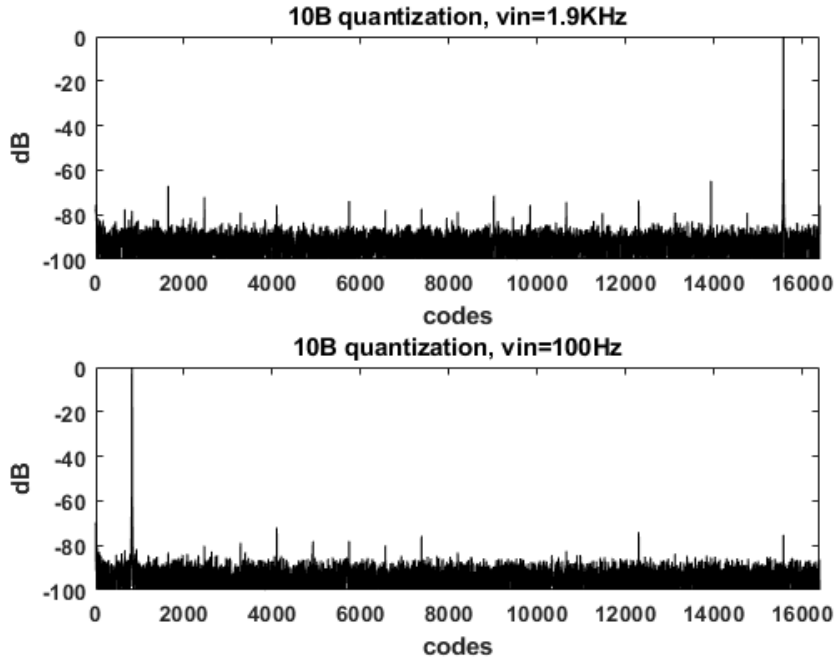


Figure 8-5. FFT for 10B quantization at 1.9KHz and 100Hz sinusoidal input.

back on the chip. After a few iterations of trying different values nearby comparator offset, the resultant DNL/INL improved as shown in Figure 8-4.

The ENOB is 9.0 at 10B when sampling a 100Hz tone at 4KHz and ENOB is 8.9 at 10B when sampling at 1.9KHz tone at 4KHz as shown in Figure 8-5. The effectiveness of the 2B residue enhancement is demonstrated with a very small differential input (8 mVpp). As shown in Figure 8-6 the noise floor is reduced by 5.4dB with the resolution enhancement. Walden FOM [32] [33] for 10B quantizer is 15.3fJ/conv-step at 1.9KHz input and 14.3fJ/conv-step at 100Hz. Schreier FOM for 10B quantizer is 163dB. It must

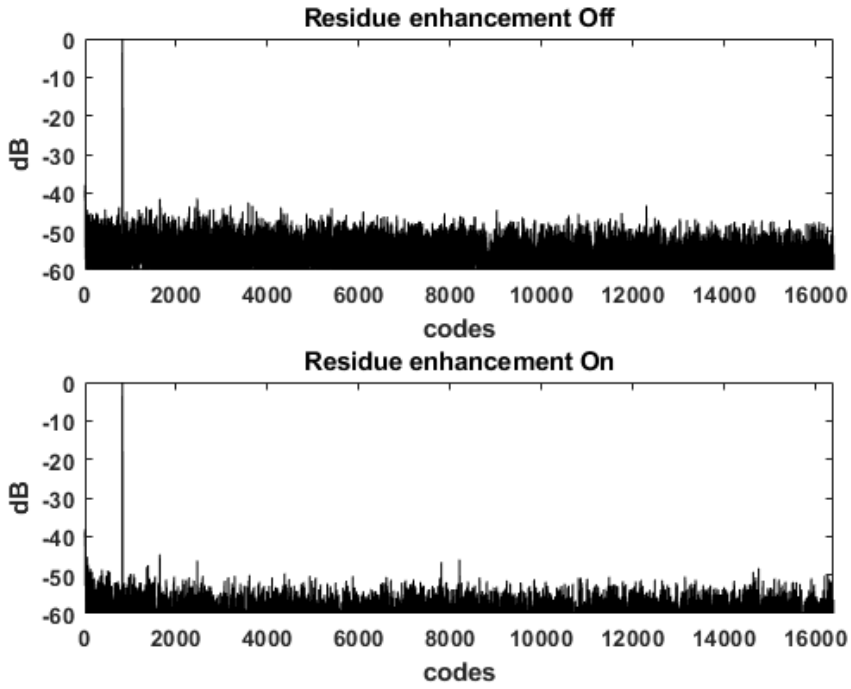


Figure 8-6. FFT for relative comparison of FFT with and without resolution enhancement. Notice noise floor moves down with resolution enhancement.

be noted that the resolution enhancement is triggered only for small input amplitudes, thus the traditional SNDR/SFDR and FOM are not appropriate measures for this mode.

The chip measures 520umx90um and was fabricated in 65nm process. The die photograph is shown in Figure 8-7. Table 8-1 compares this work to [17], [25], [20], and [31]. This work has FOM of 15.3fJ/con-step which is comparable to the works listed. The FOM is better than [31] and four times higher sampling rate with a similar ~9 ENOB. The DNL/INL are better compared with [25].

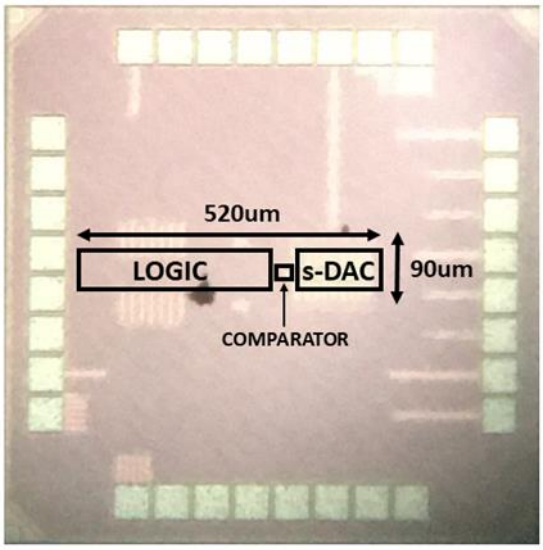


Figure 8-7. Die picture.

Table 8-1. Comparison with other works.

	[17] 10B	[25] 12B	[20] 10B	[31] 10B	This Work 10B	This Work 12B
Technology (nm)	180	40	180	130	65	
Input (Vpp, Diff)	1.0	1.6	1.13	2	1.0	
Area (mm^2)	0.120	0.056	0.082	0.191	0.046	
Vdd (V)	0.5	1.1	0.6	1 & 0.4	0.5	
SNDR (dB)	59.2	>60	57.6	56.7	55.14	-
SFDR (dB)	75	<70	66.09	67.6	64.86	-
FoM (fJ/con-step)	2.9-17	<15	8.03	94.5	15.3	-
INL	0.20	-0.8/1.8	-0.4/0.4	-0.4/0.4	-0.4/0.2	-1.0/1.0
DNL	0.08	-1.1/1.0	-0.2/0.2	-0.6/0.5	-0.2/0.2	-0.5/0.5
Power	8.7-31nW	420uW	1.04uW	53nW	29nW	49nW
Sample rate	4 KS/s	35 MS/s	200 KS/s	1 KS/s	4 KS/s	4 KS/s
ENOB (bit)	9.55 ERBW 1.3KHz	>9.67 ERBW 80MHz	9.34	9.1	8.9	9.9 small amp

Chapter 9

Conclusions

An energy-efficient SAR A/D converter for sensor signals was designed and tested in this thesis work. It uses the new algorithm called Recode-then-LSB-first (RLSB-first) that enables energy and bit-cycle savings for all cases of small code change over previous input sample. The saved bit-cycles are used for resolution enhancement, background calibration and comparator noise measurement when the input signal has small amplitude. Background calibration and comparator noise measurement are used for calibration of DAC mismatches and resolution enhancement respectively.

The proposed RLSB-first algorithm enables low DAC switching energy and bit-cycles for small code changes over previous sample both in the worst case and also on an average [24]. The algorithm achieves this by recoding previous code before initializing the DAC and performing LSB-first algorithm. Recoding previous code in the digital domain avoids toggling MSB bits on the DAC thereby saving both energy and bit-cycles per conversion. The simulation results for RLSB-first algorithm with a sinusoidal-input showed improvements of up-to 3X in worst case energy and has the low average switching energy

for small code change over previous sample in comparison to LSB-first. Simulations with various real sensor signals showed energy and bit-cycle savings up to ~29% and ~15% respectively. The saved cycles were used for resolution enhancement through repeatedly strobing the comparator in two-steps rather than in one-step in previous work. Two-step resolution enhancement results in better accuracy for the same number of comparisons. The comparator noise is measured in background to maintain accuracy of the resolution enhancement by tracking temperature changes or with process variations.

The RLSB-first algorithm and the ADC are designed to enable low power analog operation with choices for key components made as per fundamental limiting constraints of thermal noise rather than mismatches in current technology. Thermal noise, not mismatch limited, capacitive DAC and noisy comparator were selected for low power analog operation while using the benefits of the algorithm for extra resolution and correction in the digital domain. In deep-submicron CMOS process, the energy benefits of this algorithmically assisted technique will become more visible. With process scaling from 65nm to 7nm there is a 17X improvement in energy per digital operation [14] while energy-delay-product scales down by over 100X. In this test chip, about 80% of power consumption is digital. At 12B, the ideal analog energy per conversion calculated in section 1.1 and 1.2 for prediction-based SAR A/D converters is ~1.4pJ. At 12B, the analog energy per conversion for this test chip is ~2.4pJ. The scaling would make digital power insignificant and bring A/D system closer to the fundamental thermal limits.

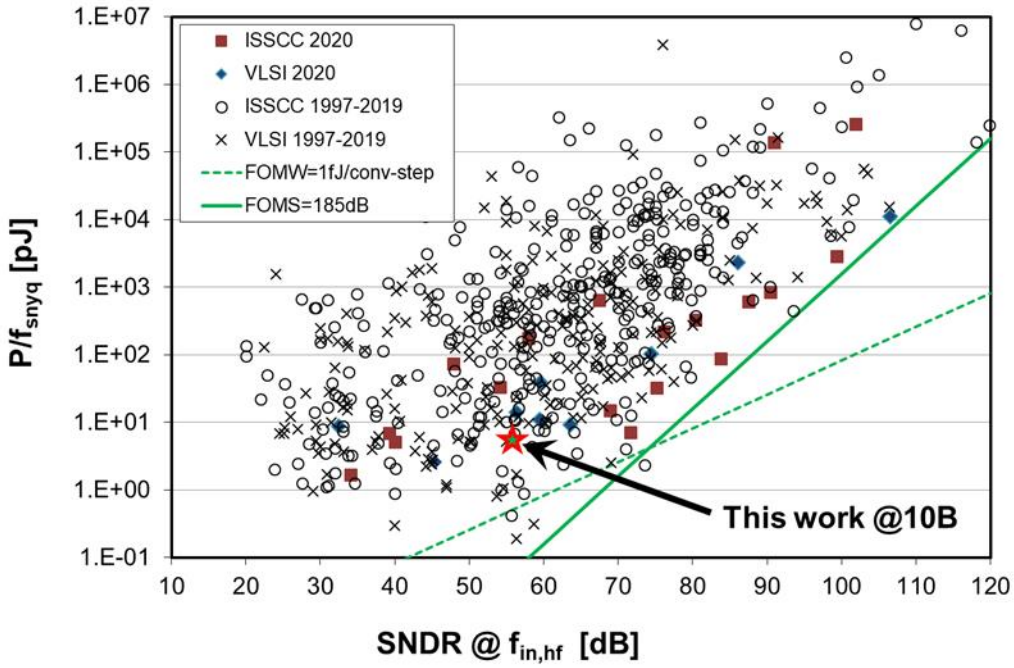


Figure 9-1. Energy (pJ) versus SNDR (dB) for ADCs [34].

The test chip has a Walden FOM of ~15fJ/conv-step and Schreier FOM is ~163dB which is comparable to other work in this field and expected to get better on more advance nodes. The plot of energy versus SNDR [34] in Figure 9-1 compares this work with others. Leveraging the small signal activity algorithmically makes this design suitable for sensor signal A/D conversion while benefitting from future digital process scaling.

Some of the future works could be in the direction of incorporating a prediction making algorithm in the digital domain. Multiple sensory input data acquisition systems could exploit spatial correlation across channels along with the temporal information to

make a prediction. For example, imaging sensor arrays could use correlation in nearby pixels while digitizing a frame on a row basis.

Bibliography

[1] M. Yip, “Ultra-low-power Circuits and Systems for Wearable and Implantable Medical Devices,” Ph.D. thesis, Massachusetts Institute of Technology, 2013.

[2] J. Olivo, S. Carrara and G. De Micheli, "Energy Harvesting and Remote Powering for Implantable Biosensors," in IEEE Sensors Journal, vol. 11, no. 7, pp. 1573-1586, July 2011.

[3] P. D. Mitcheson, "Energy harvesting for human wearable and implantable bio-sensors," 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology, Buenos Aires, 2010, pp. 3432-3436.

[4] S. Orguc, H. S. Khurana, H. Lee and A. P. Chandrakasan, "0.3 V ultra-low power sensor interface for EMG," ESSCIRC 2017 - 43rd IEEE European Solid State Circuits Conference, Leuven, 2017, pp. 219-222.

[5] D. Jeon et al., "24.3 An implantable 64nW ECG-monitoring mixed-signal SoC for arrhythmia diagnosis," 2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), San Francisco, CA, 2014, pp. 416-417.

[6] X. Zou, X. Xu, L. Yao and Y. Lian, "A 1-V 450-nW Fully Integrated Programmable Biomedical Sensor Interface Chip," in IEEE Journal of Solid-State Circuits, vol. 44, no. 4, pp. 1067-1077, April 2009.

[7] Xiaoyuan Xu, Xiaodan Zou, Libin Yao and Yong Lian, "A 1-V 450-nW fully integrated biomedical sensor interface system," 2008 IEEE Symposium on VLSI Circuits, Honolulu, HI, 2008, pp. 78-79.

[8] L. S. Y. Wong, S. Hossain, A. Ta, J. Edvinsson, D. H. Rivas and H. Naas, "A very low-power CMOS mixed-signal IC for implantable pacemaker applications," in IEEE Journal of Solid-State Circuits, vol. 39, no. 12, pp. 2446-2456, Dec. 2004.

[9] B. Murmann, "Energy limits in A/D converters," 2013 IEEE Faible Tension Faible Consommation, Paris, 2013, pp. 1-4.

[10] P. Nuzzo, F. De Bernardinis, P. Terreni and G. Van der Plas, "Noise Analysis of Regenerative Comparators for Reconfigurable ADC Architectures," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 55, no. 6, pp. 1441-1454, July 2008.

[11] J. Fredenburg and M. P. Flynn, "ADC trends and impact on SAR ADC architecture and analysis," 2015 IEEE Custom Integrated Circuits Conference (CICC), San Jose, CA, 2015, pp. 1-8.

[12] G. E. Moore, "Cramming more components onto integrated circuits, Reprinted from Electronics, volume 38, number 8, April 19, 1965, pp.114 ff.," in IEEE Solid-State Circuits Society Newsletter, vol. 11, no. 3, pp. 33-35, Sept. 2006.

[13] R. H. Dennard, F. H. Gaensslen, Hwa-Nien Yu, V. L. Rideout, E. Bassous and A. R. Leblanc, "Design Of Ion-implanted MOSFET's with Very Small Physical Dimensions," in Proceedings of the IEEE, vol. 87, no. 4, pp. 668-678, April 1999.

[14] Stillmaker, A. & Baas, B. Scaling equations for the accurate prediction of CMOS device performance from 180 nm to 7 nm. Integr. VLSI J. 58, 74–81 (2017).

[15] D. Johns and K. Martin, "Analog Integrated Circuit Design," Wiley, 1997.

[16] F. M. Yaul and A. P. Chandrakasan, "A 10 bit SAR ADC With Data-Dependent Energy Reduction Using LSB-First Successive Approximation," in IEEE Journal of Solid-State Circuits, vol. 49, no. 12, pp. 2825-2834, Dec. 2014.

[17] F. M. Yaul and A. P. Chandrakasan, "11.3 A 10b 0.6nW SAR ADC with data-dependent energy savings using LSB-first successive approximation," 2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), San Francisco, CA, 2014.

[18] M. Yip and A. P. Chandrakasan, "A Resolution-Reconfigurable 5-to-10-Bit 0.4-to-1 V Power Scalable SAR ADC for Sensor Applications," in IEEE Journal of Solid-State Circuits, vol. 48, no. 6, pp. 1453-1464, June 2013.

[19] S. O'Driscoll, K. V. Shenoy and T. H. Meng, "Adaptive Resolution ADC Array for an Implantable Neural Sensor," in IEEE Transactions on Biomedical Circuits and Systems, vol. 5, no. 2, pp. 120-130, April 2011.

[20] G. Huang, S. Chang, C. Liu and Y. Lin, "A 1-μW 10-bit 200-kS/s SAR ADC With a Bypass Window for Biomedical Applications," in IEEE Journal of Solid-State Circuits, vol. 47, no. 11, pp. 2783-2795, Nov. 2012.

[21] A. Hastings, The Art of Analog Layout. Upper Saddle River, NJ: Prentice Hall, 2001.

[22] B. P. Ginsburg and A. P. Chandrakasan, "An energy-efficient charge recycling approach for a SAR converter with capacitive DAC," 2005 IEEE International Symposium on Circuits and Systems, Kobe, 2005, pp. 184-187 Vol. 1.

[23] B. P. Ginsburg and A. P. Chandrakasan, "500-MS/s 5-bit ADC in 65-nm CMOS With Split Capacitor Array DAC," in IEEE Journal of Solid-State Circuits, vol. 42, no. 4, pp. 739-747, April 2007.

[24] H. S. Khurana, A. P. Chandrakasan and H. Lee, "Recode then LSB-first SAR ADC for Reducing Energy and Bit-cycles," 2018 IEEE International Symposium on Circuits and Systems (ISCAS), Florence, 2018, pp. 1-5.

[25] B. Verbruggen, J. Tsouhlarakis, T. Yamamoto, M. Iriguchi, E. Martens and J. Craninckx, "A 60 dB SNDR 35 MS/s SAR ADC With Comparator-Noise-Based Stochastic Residue Estimation," in IEEE Journal of Solid-State Circuits, vol. 50, no. 9, pp. 2002-2011, Sept. 2015.

[26] Hae-Seung Lee and D. Hodges, "Self-calibration technique for A/D converters," in IEEE Transactions on Circuits and Systems, vol. 30, no. 3, pp. 188-190, March 1983.

[27] P. J. A. Harpe et al., "A 26 μW 8 bit 10 MS/s Asynchronous SAR ADC for Low Energy Radios," in IEEE Journal of Solid-State Circuits, vol. 46, no. 7, pp. 1585-1595, July 2011.

[28] Masaya Miyahara, Yusuke Asada, Daehwa Paik and Akira Matsuzawa, "A low-noise self-calibrating dynamic comparator for high-speed ADCs," 2008 IEEE Asian Solid-State Circuits Conference, Fukuoka, 2008, pp. 269-272.

[29] S. Chang, K. Al-Ashmouny and E. Yoon, "A 0.5V 20fJ/conversion-step rail-to-rail SAR ADC with programmable time-delayed control units for low-power biomedical application," 2011 Proceedings of the ESSCIRC (ESSCIRC), Helsinki, 2011, pp. 339-342.

[30] F. Maloberti, “Data Converters,” Springer, 2007.

[31] D. Zhang, A. Bhide and A. Alvandpour, "A 53-nW 9.1-ENOB 1-kS/s SAR ADC in 0.13-μm CMOS for Medical Implant Devices," in IEEE Journal of Solid-State Circuits, vol. 47, no. 7, pp. 1585-1593, July 2012.

[32] R. H. Walden, "Analog-to-digital converter technology comparison," Proceedings of 1994 IEEE GaAs IC Symposium, Phildelphia, PA, USA, 1994, pp. 217-219.

[33] R. H. Walden, "Analog-to-digital converter survey and analysis," in IEEE Journal on Selected Areas in Communications, vol. 17, no. 4, pp. 539-550, April 1999.

[34] B. Murmann, “ADC Performance Survey 1997-2020.” [Online]. Available: http://www.stanford.edu/~murmann/adcsurvey.html.

[35] J. Doernberg, H. -. Lee and D. A. Hodges, "Full-speed testing of A/D converters," in IEEE Journal of Solid-State Circuits, vol. 19, no. 6, pp. 820-827, Dec. 1984.

[36] B. Razavi, “Design of Analog CMOS Integrated Circuits,” McGraw-Hill, 2001.

[37] J. Van Rethy, M. De Smedt, M. Verhelst and G. Gielen, "Predictive sensing in analog-to-digital converters for biomedical applications," International Symposium on Signals, Circuits and Systems ISSCS2013.

[38] Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark RG, Mietus JE, Moody GB, Peng C-K, Stanley HE. PhysioBank, PhysioToolkit, and PhysioNet:

Components of a New Research Resource for Complex Physiologic Signals. Circulation 101(23):e215-e220. Circulation Electronic Pages:

http://circ.ahajournals.org/cgi/content/full/101/23/e215]; 2000 (June 13).

[39] “Fault Data Sets.” Society For Machinery Failure Prevention Technology, 27 Feb. 2020, www.mfpt.org/fault-data-sets/.

[40] Examples of Electromyograms, archive.physionet.org/physiobank/database/emgdb/.

[41] Lugovaya T.S. Biometric human identification based on electrocardiogram. [Master's thesis] Faculty of Computing Technologies and Informatics, Electrotechnical University "LETI", Saint-Petersburg, Russian Federation; June 2005.

[42] B. Razavi, Principles of Data Conversion System Design. Wiley-IEEE Press, 1994.