

AN ABSTRACT OF THE THESIS OF

Jerry Leung for the degree of Master of Science in Electrical and Computer Engineering presented on December 8, 2014.

Title: Data Driven Optimization in SAR ADC

Abstract approved: _____

Un-Ku Moon

Recent publications show that successive approximation register (SAR) analog to digital converters (ADC) are capable of achieving high efficiency over other ADC topologies. Furthermore, techniques have been adopted to process signals with low activity periods, such as biomedical and industrial sensors. Prior work used least-significant bit first quantization (LSBFQ) to conserve capacitor switching energy and comparator decisions (bitcycles).

This work improves on the published least significant bit (LSB) first successive approximation ADC by restructuring its algorithm for further energy efficient switching, lowering its bitcycle range, and extending its range of applications. For target applications, these proposed solutions will outperform the bit-skipping LSBFQ and the merged capacitor switching (MCS) SAR, the most energy-efficient traditional most significant bit (MSB) first SAR.

©Copyright by Jerry Leung
December 8, 2014
All Rights Reserved

Data Driven Optimization in SAR ADC

by

Jerry Leung

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented December 8, 2014
Commencement June 2015

Master of Science thesis of Jerry Leung presented on December 8, 2014.

APPROVED:

Major Professor, representing Electrical and Computer Engineering

Director of the School of Electrical Engineering and Computer Science

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Jerry Leung, Author

ACKNOWLEDGEMENTS

After being in Corvallis throughout my undergraduate and graduate education, it truly was eye opening to realize how much I have learned in the past six years. When I first started my education at Oregon State University, I did not even have any basic understandings of electrical circuits. More importantly, I lacked the capability to motivate and drive my passion to realize my potential. My teachers, mentors, friends and family paved the path and guided me along to being who I am today. With this page of acknowledgements, I want to note some of the things that made this past quarter of my life memorable and valuable. By the way, I just want to point out that this is not an exhaustive list because I am trying to finish writing this part so I can submit to grad school.

I would like to acknowledge Dr. Un-Ku Moon for his great personality and guidance that were always in my best interest. His welcoming attitude made grad school very enjoyable. He even got us a pull-up bar to keep up with our physical appearance during the hard times. Thanksgiving dinners and 4th of July BBQs at Dr. Moon's along with all the times we had group outings were awesome! The annual Shasta trips were definitely the best.

While I'm at it, I would like to mention all of Dr. Moon's research group members who I have had a chance to interact with (Brandilyn Coker, Farshad Farahbakhshian, Yue (Simon) Hu, Spencer Leuenberger, Jason Muhlestein, Hyuk (Tim) Sun, Allen Waters, Yang Xu, Praveen Venkatachala, Manideep Gande, Hari Venkatram, Taehwan Oh, and Jon Guerber during my undergraduate studies).

These fellows made my day and provided a friendly environment whenever I came in to the office. I became great friends with many of them through discussions of circuits, course work, countless lunches at restaurants on Monroe, tailgating during football games, intramural sports, etc. Sometimes I wish to stay longer just to hang out more.

My parents and brothers are wonderful and provided me the stepping stones for success. From helping with my laundry whenever I come home to giving me life advice during family dinner. I continue to grow whenever I am around them.

To all my childhood friends who I have known for around two thirds of my life, you guys are cool and I hope that our friendship will continue to last.

Lastly, I want to note some of the fun moments that occurred. Thanks to Dr. Karti during his frequent visits to the graduate office and pretends to use the microwave just to say hello. Bomb's away for its low prices (\$2 for a pint!). My research pals who took care of me when I tailgated too hard at the football game. Dr. Moon's famous quote, "In the end, nothing matters!" The butter coffee that Farshad recommended for the research group to try. Tim and his bananas in the office. Our research group's fantasy football league. Pumpkin carving at Spencer's. Country western swing dancing. Playing Cards Against Humanity with professors during the shasta trip.

TABLE OF CONTENTS

	<u>Page</u>
1 Introduction	1
2 Background	3
2.1 Successive Approximation Register Analog to Digital Converters . . .	4
2.2 Bit-Skipping Least Significant Bit First SAR ADC	9
2.2.1 Drawbacks of Bit-Skipping LSBFQ	16
3 Bit-Repeating Least Significant Bit First SAR ADC	19
3.1 Algorithm Improvements	19
3.2 Implementation Details	19
3.3 Results	22
4 Selectable Starting Bit SAR ADC	26
4.1 Algorithm Improvements	26
4.2 Implementation Details	27
4.3 Results	32
5 Adaptive Starting Bit SAR Quantizer	39
5.1 Implementation Details	39
5.2 Results	43
5.3 Future Direction	46
6 Conclusion	49
Bibliography	50

LIST OF FIGURES

Figure	Page
2.1 Energy Consumption by SNDR with previously published architectures up to 2012. [3]	5
2.2 Conventional SAR structure. [9]	6
2.3 Conventional 4-bit SAR operation. [9]	7
2.4 Conventional SAR Operation for N-bits	8
2.5 DAC architecture for a 10-bit conventional SAR ADC	8
2.6 Target Applications for LSBFQ	10
2.7 DAC architecture for a 10-bit LSBFQ ADC	10
2.8 Bit-Skipping LSBFQ algorithm	11
2.9 Example of Bit-Skipping LSBFQ	12
2.10 Bit-skipping LSBFQ Operation	12
2.11 Bit-skipping LSBFQ operation: best case scenario	13
2.12 Bit-Skipping LSBFQ measured mean bitcycles/sample and mean energy consumption at $f_s = 10kHz$ and $V_{DD} = 0.6V$ as a function of mean output code change per sample, for sine wave and ECG signals [12]	14
2.13 ADC response to ECG test input signal with $f_s = 10kHz$ and $V_{DD} = 0.5V$ [12]	15
2.14 Examples of bit-skipping LSBFQ, for (a) good case and (b) bad case.	17
2.15 Number of bitcycles required as a function of ΔD_{out} in a 10-bit quantizer with input frequency of $10^{-3}f_s$, for bit-skipping LSBFQ. .	18
3.1 Bit-Repeating LSBFQ algorithm	20
3.2 Bit-repeating LSBFQ Operation	21
3.3 Example of Bit-Repeating LSBFQ	22

LIST OF FIGURES (Continued)

<u>Figure</u>	<u>Page</u>
3.4	Number of bitcycles required as a function of ΔD_{out} in a 10-bit quantizer with input frequency of $10^{-3}f_s$, for bit-repeating LSBFQ. 23
3.5	Capacitor switching energy as a function of input signal amplitude, for (a) $0.10f_s$ and (b) $0.49f_s$. The proposed algorithm is compared to MCS [5], and bit-skipping LSB-first [4] methods. 24
3.6	Successive-approximation bitcycles as a function of input signal amplitude, for (a) $0.10f_s$ and (b) $0.49f_s$. The proposed algorithm is compared to conventional [2], MCS [5], and bit-skipping LSB-first [4] methods. 25
4.1	Target Applications for SSB SAR ADC 26
4.2	Proposed Selectable Starting Bit (SSB) algorithm. 28
4.3	Examples of proposed SSB algorithm with the SSB set to the third bit in a 5-bit quantizer. (a) example with the toMSB phase and (b) example skipping the toMSB phase. 29
4.4	Selectable Starting Bit ADC Operation: skipping the toMSB phase 30
4.5	Selectable Starting Bit ADC Operation 31
4.6	Mean number of bitcycles required by proposed algorithm as a function of frequency swept to Nyquist with various bits set as the SSB, for (a) 10% of fullscale input and (b) fullscale input. 33
4.7	Range in number of bitcycles required by proposed algorithm as a function of frequency swept to Nyquist with various bits set as the SSB, for (a) 10% of fullscale input and (b) fullscale input. 34
4.8	Capacitor switching energy required by proposed algorithm as a function of frequency swept to Nyquist with various bits set as the SSB, for (a) 10% of fullscale input and (b) fullscale input. 35
4.9	Comparisons between the SSBQ to MCS [5] and bit-skipping LSBFQ [4] for mean number of bitcycles. Comparisons are swept to Nyquist with (a) 10% fullscale input and (b) fullscale input. 37

LIST OF FIGURES (Continued)

<u>Figure</u>		<u>Page</u>
4.10	Comparisons between the SSBQ to MCS [5] and bit-skipping LSBFQ [4] for capacitor switching energy. Comparisons are swept to Nyquist with (a) 10% fullscale input and (b) fullscale input.	38
5.1	Adaptive Starting Bit ADC: feedback to adapt to the correct SSB range	40
5.2	Adaptive Starting Bit ADC Operation: increasing the SSB range .	41
5.3	Adaptive Starting Bit ADC Operation: decreasing the SSB range .	42
5.4	Comparisons between ASB and SSB ADC with various bits set using full-scale input for (a) mean bitcycle and (b) switching energy. . . .	44
5.5	Bitcycle comparisons between ASB ADC, bit-skipping LSBFQ, and MCS using full-scale input for (a) mean and (b) max bitcycles. . . .	45
5.6	Switching energy comparison between ASB ADC, bit-skipping LSBFQ, and MCS using full-scale input.	46

Chapter 1: Introduction

Successive approximation register (SAR) analog to digital converter (ADC) is utilized in many modern day applications for its simple structure and efficient power dissipation compared to other converter types. To further optimize on power, techniques have been adopted to process signals with low activity periods, such as biomedical and industrial sensors [13], [14].

Prior work used least-significant bit first quantization (LSBFQ) to conserve switching energy and comparator bitcycles [12]. Referred as Bit-Skipping LSBFQ, the structure uses the previous output code as the initial estimate, and incremental steps starting from the least significant bit (LSB) are made to ensure minimal switching activity in the feedback digital-to-analog converter (DAC). While this approach requires less switching energy and fewer comparator decisions (bitcycles) compared to conventional methods, they are limited to low signal activities and is subject to a large bitcycle range in the quantizer.

An improvement in the algorithm is proposed which reduces the large bitcycle range in the quantizer. Referred as Bit-Repeating LSBFQ, the new switching algorithm ensures that initial DAC code changes are always initially small and slowly increase in magnitude, with no disproportionately large steps. The result is that the number of comparator bitcycles and DAC switching energy is more correlated to the signal activity. This further improves the use of LSBFQ in low-

activity input signals.

A novel selectable starting bit (SSB) SAR is proposed which starts quantization with neither the MSB nor the LSB, but an *intermediate* bit chosen for target applications. It is shown that the proposed algorithm reduces bitcycle range in the quantizer compared to LSBFQ, and provides design flexibility for various activity signals. Furthermore, the proposed solution encompasses LSBFQ since it is a specific case of the proposed architecture.

An adaptive starting bit (ASB) SAR is further proposed which introduces a digital feedback in the SSB SAR to calibrate for the appropriate starting bit given the input. With the ability to adapt to the best starting bit with a given input, bitcycle savings is maximized.

For target applications, these proposed solutions will save bitcycles in an A/D conversion, as well as switching energy, over the bit-skipping LSBFQ and the merged capacitor switching (MCS) SAR, the most energy-efficient traditional most significant bit (MSB) first SAR.

Chapter 2: Background

There are four common ADC topologies: Flash, Pipeline, Successive Approximation Register (SAR), and Delta Sigma [8]. The most fundamental ADC is the flash. The flash ADC is composed of a string of comparators set at different reference voltages. The input voltage is compared with a ladder of reference voltages to determine which step of the voltage ladder the input is within. Since the quantization is performed in a single stage, flash ADCs can accommodate a large bandwidth. However, for an N -bit flash ADC, it requires 2^N comparators. Therefore, although flash is the fundamental of all the ADCs, the hardware increases exponentially for higher number of bits. The other ADCs use the basic flash converter but form ways to minimize the required hardware by implementing various feedback techniques [6].

The pipeline ADC cascades multiple stages of the flash converter. During each stage of the ADC, a small number of bits is determined and the analog input is subtracted from the digital flash output such that only the residue remains. This analog residue is then multiplied by the stage's flash resolution with an amplifier such that the next stage can quantize from the full-scale range. This operation continues as the input is quantized through the pipeline until all the bits are determined. Doing so reduces the number of comparators required since it now scales linearly ($2 \times N$ number of comparators in a 1-bit per stage case) instead

of 2^N with a flash ADC. However, this architecture requires power demanding amplifiers.

In a delta sigma ADC, an integrating feedback is built around a flash or other low resolution quantizer [11]. By oversampling the system such that the analog input does not change much from one conversion to the next, the digital code from the output of the quantizer is fed back and subtracted from the analog input. As a result, quantization noise remains while hardly any signal from the analog input remains. When integrated, this effectively shifts the quantization noise to a higher frequency while containing the input to a low frequency. Referred as noise shaping, the noise in the high frequency is then filtered out to achieve very high accuracy. Although this architecture provides high performance in terms of accuracy, it is limited to low input bandwidths due to its necessity to oversample the input signal.

The SAR ADC is so named because the input signal is quantized by using steps of successive approximations (the register part come from how early ADCs were built). It is the converter of choice for high efficiency for mid-range conversions. The SAR will be the foundation of this thesis. As such, the SAR ADC will be explained with details in its own section below.

2.1 Successive Approximation Register Analog to Digital Converters

As device sizes and supply voltages reduce in sub-micron technologies, digital circuits are able to reap the power reduction and speed benefits. However, analog

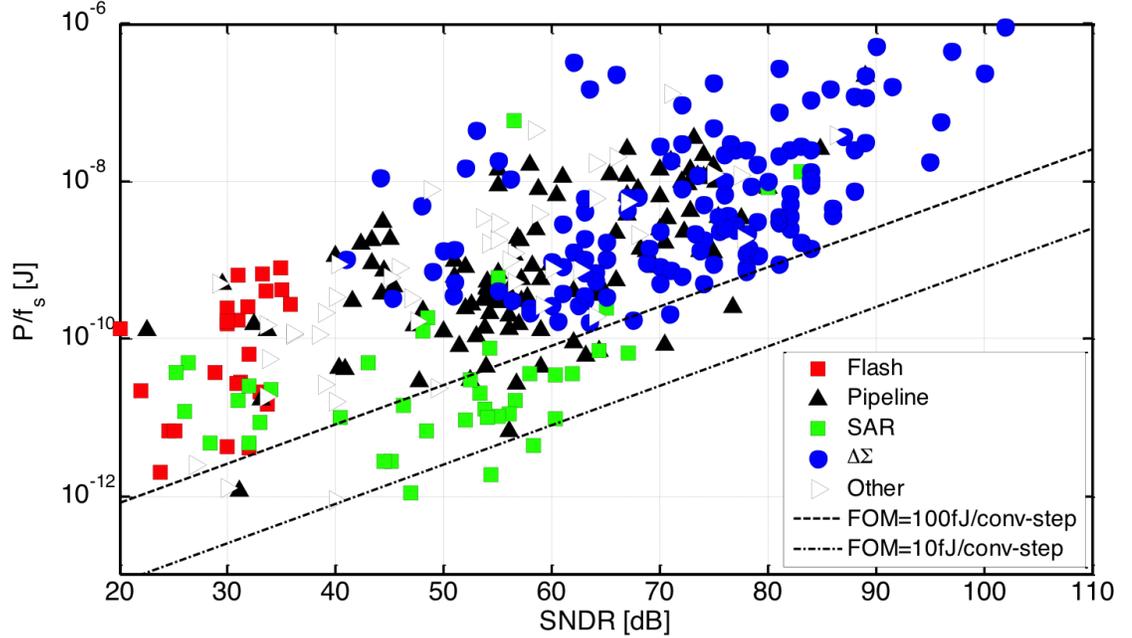


Figure 2.1: Energy Consumption by SNDR with previously published architectures up to 2012. [3]

circuits become harder to design due to the reduced voltage range. As such, analog interfaces such as ADCs in sub-micron technologies require strenuous efforts to maintain its specifications.

Luckily, SAR ADCs blocks are highly digital, therefore they have become popular in the past decade due to their ability to achieve high energy efficiency by leveraging devices sizes and reduced supplies in sub-micron process technologies. This trend is shown in Fig. 2.1 with SAR ADCs achieving very low energy consumption. The block diagram for the SAR ADC is shown in Fig. 2.2.

At the start of each conversion, the analog input is sampled through the track and hold block. Meanwhile, the voltage reference V_{ref} is set to half of the full-scale

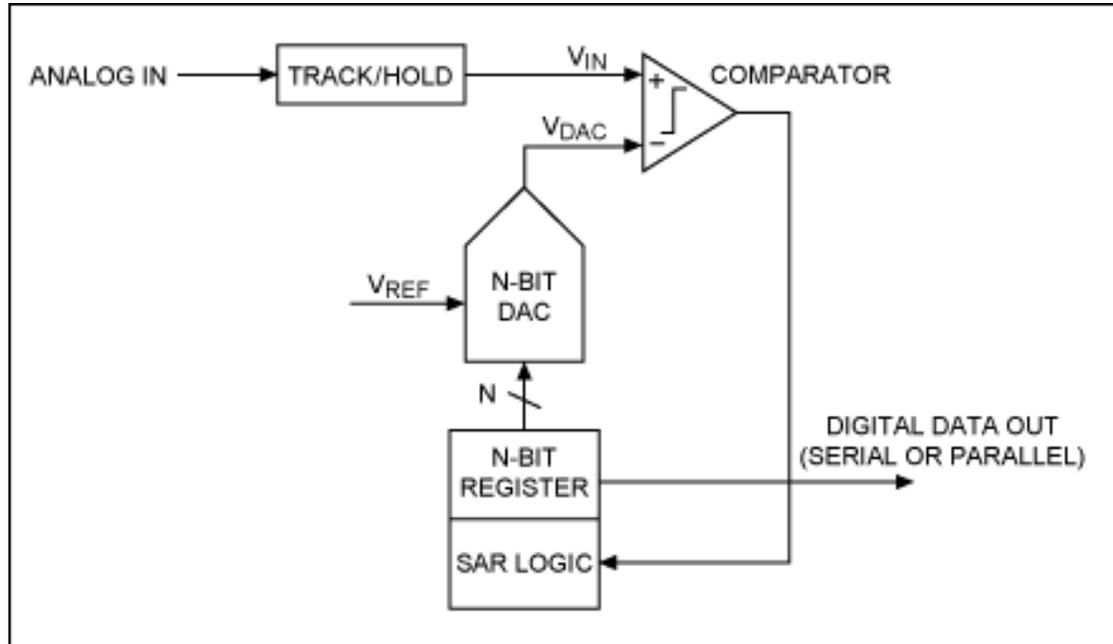


Figure 2.2: Conventional SAR structure. [9]

voltage range. Once setup is complete, the comparator will compare the sample input voltage with the reference voltage. If the comparator outputs high, it means the sampled input voltage is higher than the reference, and the most significant bit (MSB) will be set to 1. Vice versa, if the comparator outputs low, MSB will be set to 0. When the output is determined, information is sent to the SAR logic where it will store the bit in a register, and configure the N-bit Digital-to-Analog Converter (DAC) with the MSB value. Once it is configured, V_{ref} is now shifted to reflect the MSB voltage and a comparison is made again for the next bit. This continues until all bits are determined. For a N-bit traditional SAR ADC, N number of comparator decisions (bitcycles) are required.

A visual representation of the SAR operation is shown in Fig. 2.3. However,

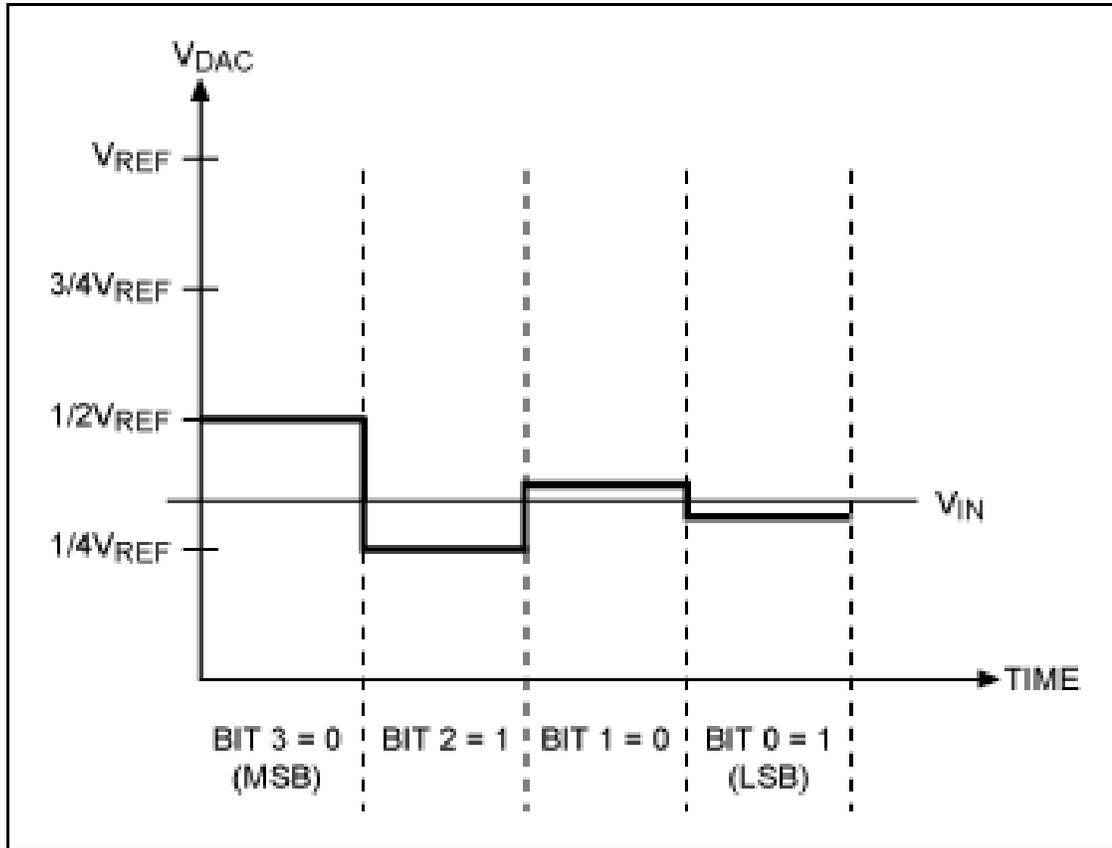


Figure 2.3: Conventional 4-bit SAR operation. [9]

another way to visualize this operation is to consider the range of possible voltage values shrinking during each bitcycle until the sampled input is within one LSB voltage step. This visualization is shown in 2.4, and similar conceptual understanding will be applied to explain the improvements/changes made to the SAR algorithm in the following chapters.

A big component of power consumption in the ADC is switching the capacitor array in the DAC. Starting with the LSB with the unit size capacitor, the capacitor size is doubled for each bit in the ADC. Therefore, significant amount of charge

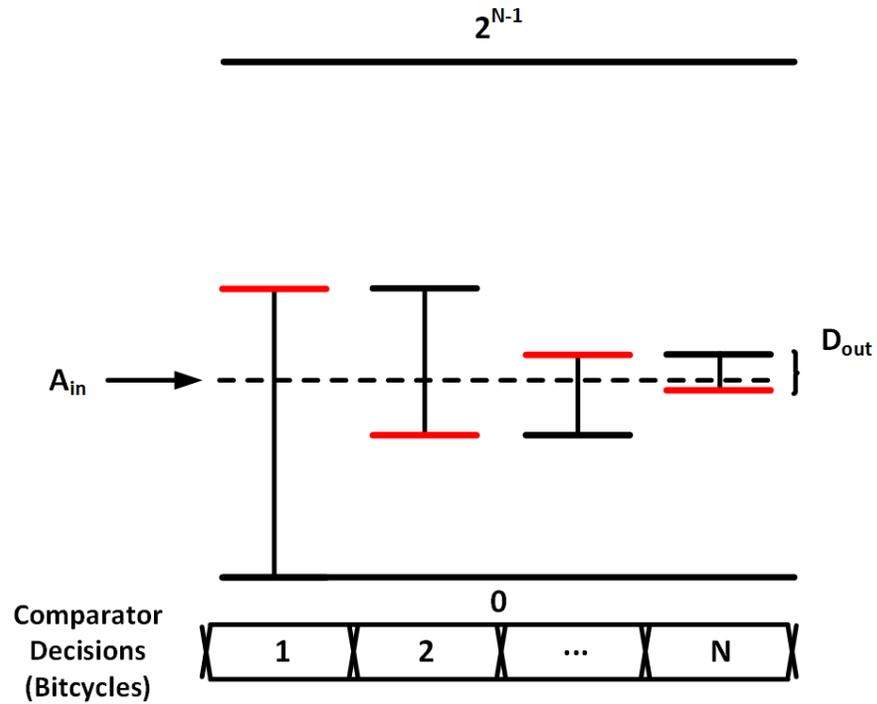


Figure 2.4: Conventional SAR Operation for N-bits

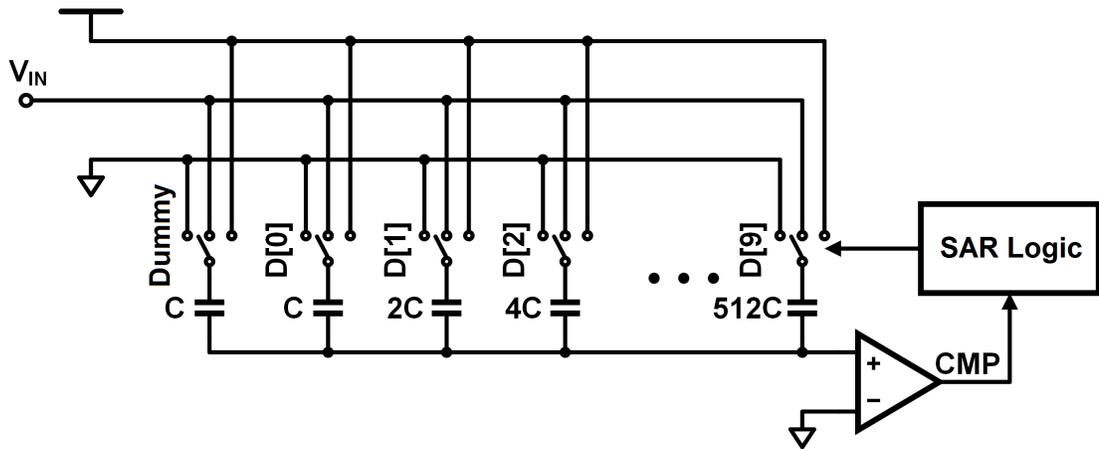


Figure 2.5: DAC architecture for a 10-bit conventional SAR ADC

will be required to switch the MSBs. In SAR ADCs, energy is often examined where energy (Joules) is the charge used times the supply voltage or the power multiplied by some time window. The following equation can be used to calculate the power required for the capacitive system:

$$P_{VDD} = I \times V_{VDD} = \frac{Q}{T} V_{VDD} = \frac{\alpha C_T V}{T} V_{VDD} = \alpha C_T V_{DD}^2 f \quad (2.1)$$

Where α is the capacitance activity factor (the percentage of the total capacitance is switched in each cycle) and f is the switching frequency of the capacitance.

Traditional MSB First SAR ADCs are great for high activity signals near Nyquist rate when MSBs are switched consistently. However, MSBs do not change frequently for slow moving signals such as battery monitoring regulator circuits or ECG signals with long periods of low activity. A following ADC denoted by Bit-skipping Least Significant Bit First SAR ADC is introduced in [4] to take advantage of this characteristic in low activity signals for power savings and bitcycle reductions.

2.2 Bit-Skipping Least Significant Bit First SAR ADC

The goal of Bit-skipping Least Significant Bit First Quantizer (LSBFQ) is to achieve low energy per conversion when the input does not vary too much sample to sample. Fig. 2.6 is a representation to illustrate the benefits of LSBFQ over conventional SAR.

The SAR structure using LSBFQ is shown in Fig. 2.7. Note that an N -bit

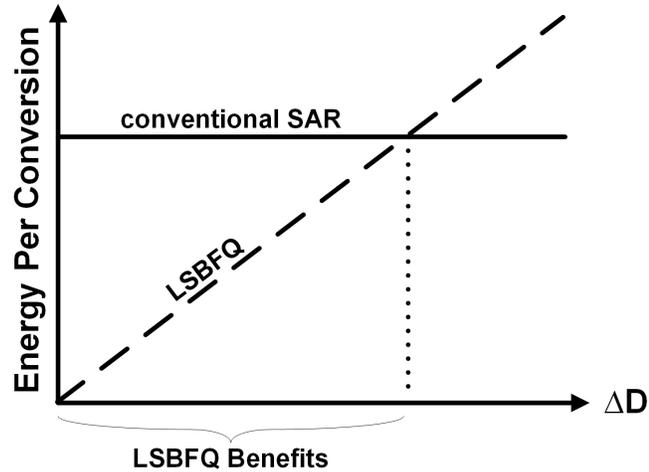


Figure 2.6: Target Applications for LSBFQ

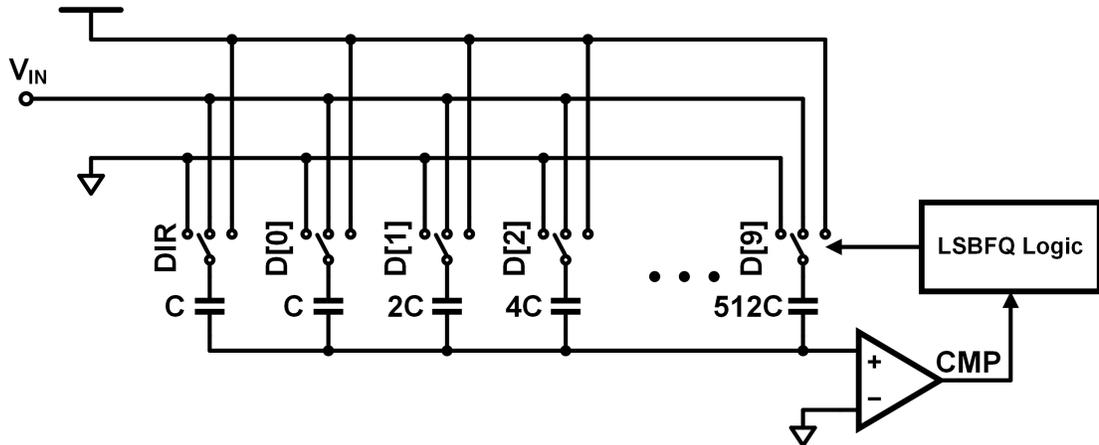


Figure 2.7: DAC architecture for a 10-bit LSBFQ ADC

LSBFQ requires $N + 1$ DAC capacitors, to include an extra unit-sized capacitor controlled by signal DIR . Otherwise the structure strongly resembles a conventional SAR, with the exception of the DAC control logic.

As illustrated in Fig. 2.8, Bit-skipping LSB-first successive approximation algorithm [4] takes place in three phases: Initialization, toMSB, and toLSB. Let D be

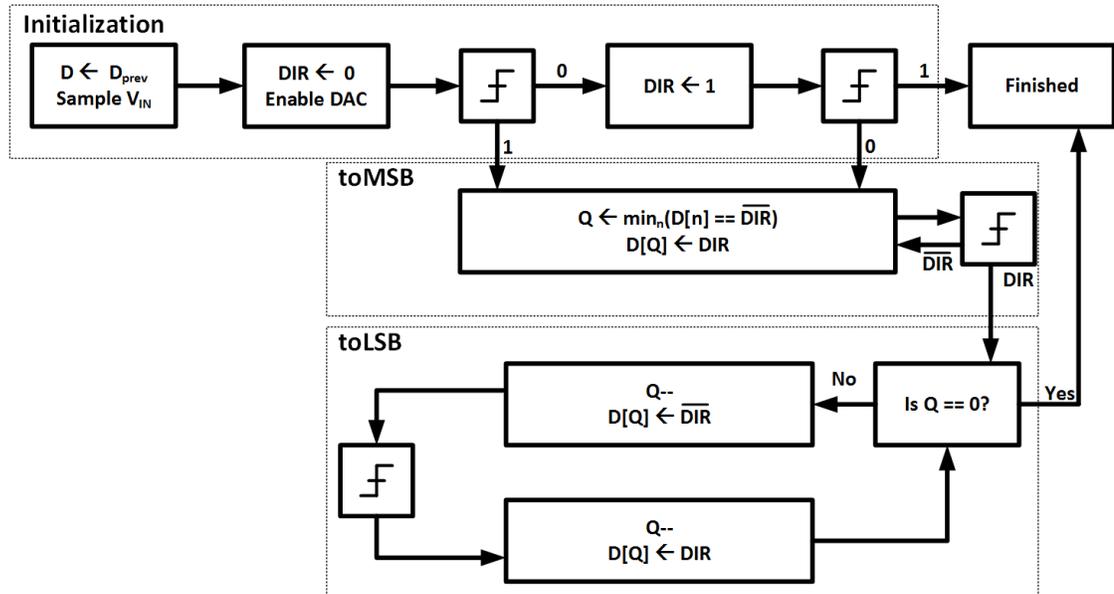


Figure 2.8: Bit-Skipping LSBFQ algorithm

the current DAC code in the SAR, and let $D_{out}[n]$ be the final output code of the n -th A/D conversion. The algorithm begins with the previous output code as the initial estimate, $D = D_{out}[n - 1]$. After the first comparator decision is made, the direction DIR is determined to indicate whether the DAC output must increase or decrease, indicated with $DIR = 1$ and $DIR = 0$ respectively. After direction is determined, there is a check of one LSB voltage step towards the direction denoted by DIR . This allows the quantizer to catch signals close to the initial guess and finish quantization withing two bitcycles. Fig. 2.11 is included to show this case. If $D_{out}[n]$ is beyond the range of this check, the algorithm enters the toMSB phase.

During the toMSB phase in the bit-skipping LSBFQ [4], the algorithm finds the least significant bit that is set opposite from DIR and toggles it. This in turn causes a step in the DAC output towards the desired direction. The operation

D _{out} [n-1] = 11, D _{out} [n] = 18, DIR = 1			
Bitcycle	D[4:0]	CMP	ΔD
1	01011	0	---
2	01011	0	---
3	01 1 11	0	+4
4	1 1111	1	+16
5	1 0 111	1	-8
6	10 0 11	1	-4
7	100 0 1	0	-2
8	1001 0	1	+1

Figure 2.9: Example of Bit-Skipping LSBFQ

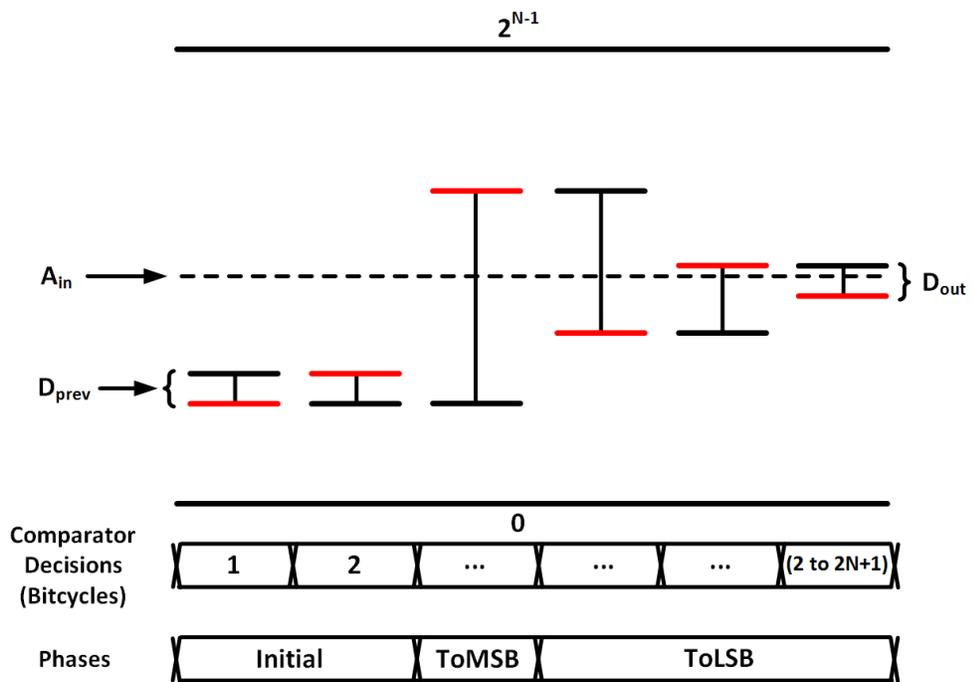


Figure 2.10: Bit-skipping LSBFQ Operation

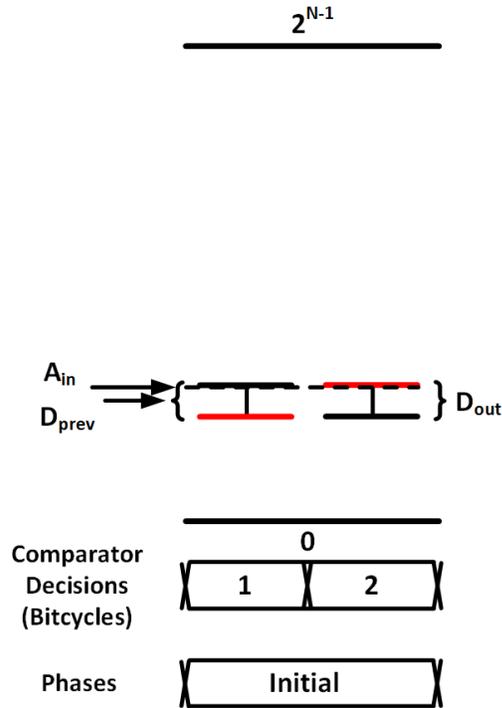


Figure 2.11: Bit-skipping LSBFQ operation: best case scenario

continues until the comparator detects the DAC output overshooting the correct value. Note that this implementation will skip over least significant bits that are already set to DIR , and toggles more significant bits with correspondingly larger voltage steps.

Once the comparator detects an overshoot, the toLSB phase conducts a conventional SAR conversion beginning with bit $Q-1$ and work back towards the LSB. For this phase it is important that all the bits in the range 0 to $Q-1$ are initially set to the same value (either 1-filled or 0-filled), as guaranteed by the toMSB phase.

A numerical example of this implementation is shown in Fig. 2.9 and a visual example of this implementation is shown in Fig. 2.10. Fig. 2.11 shows the

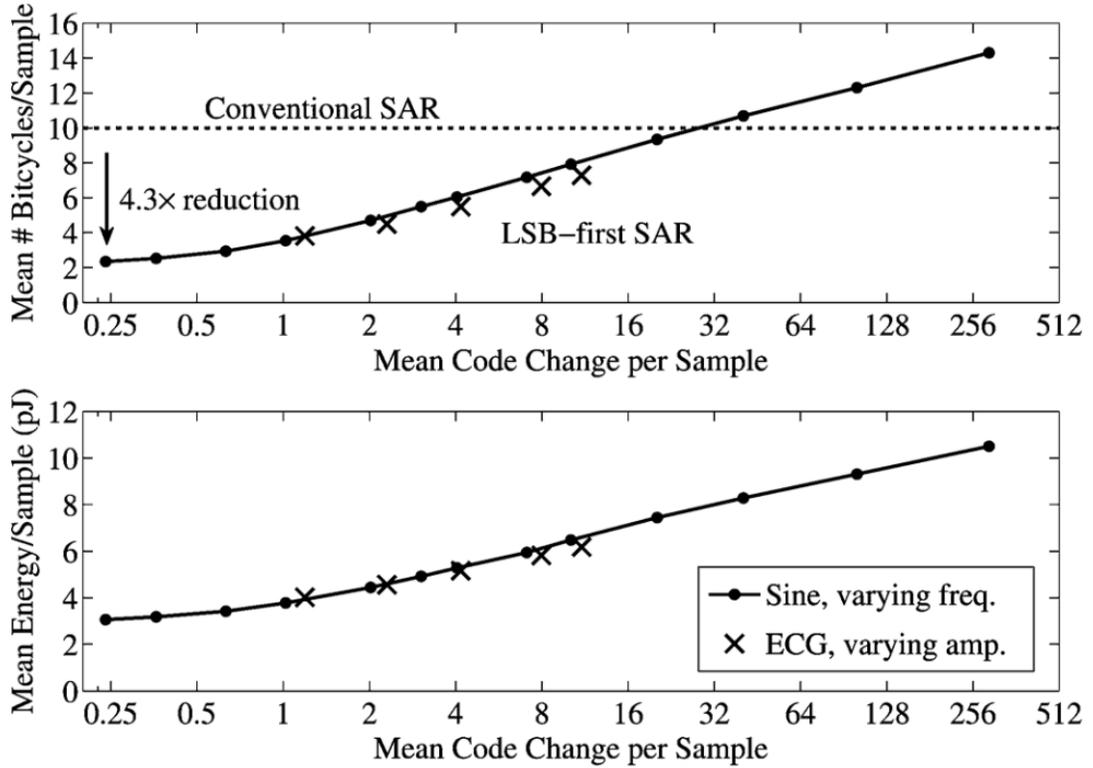


Figure 2.12: Bit-Skipping LSBFQ measured mean bitcycles/sample and mean energy consumption at $f_s = 10kHz$ and $V_{DD} = 0.6V$ as a function of mean output code change per sample, for sine wave and ECG signals [12]

best case scenario for the LSBFQ when quantization can be completed within the initialization phase.

The number of bitcycles required for the bit-skipping LSBFQ ranges from 2 to $2N+1$ for N bits. In the best case scenario, the conversion is completed in the initialization phase when the input voltage is within 2 LSB voltage steps of the initial guess. The worst case scenario occurs when the guess is off by $2^N - 1$ LSB voltage steps and every single bit has to be toggled during the toMSB phase.

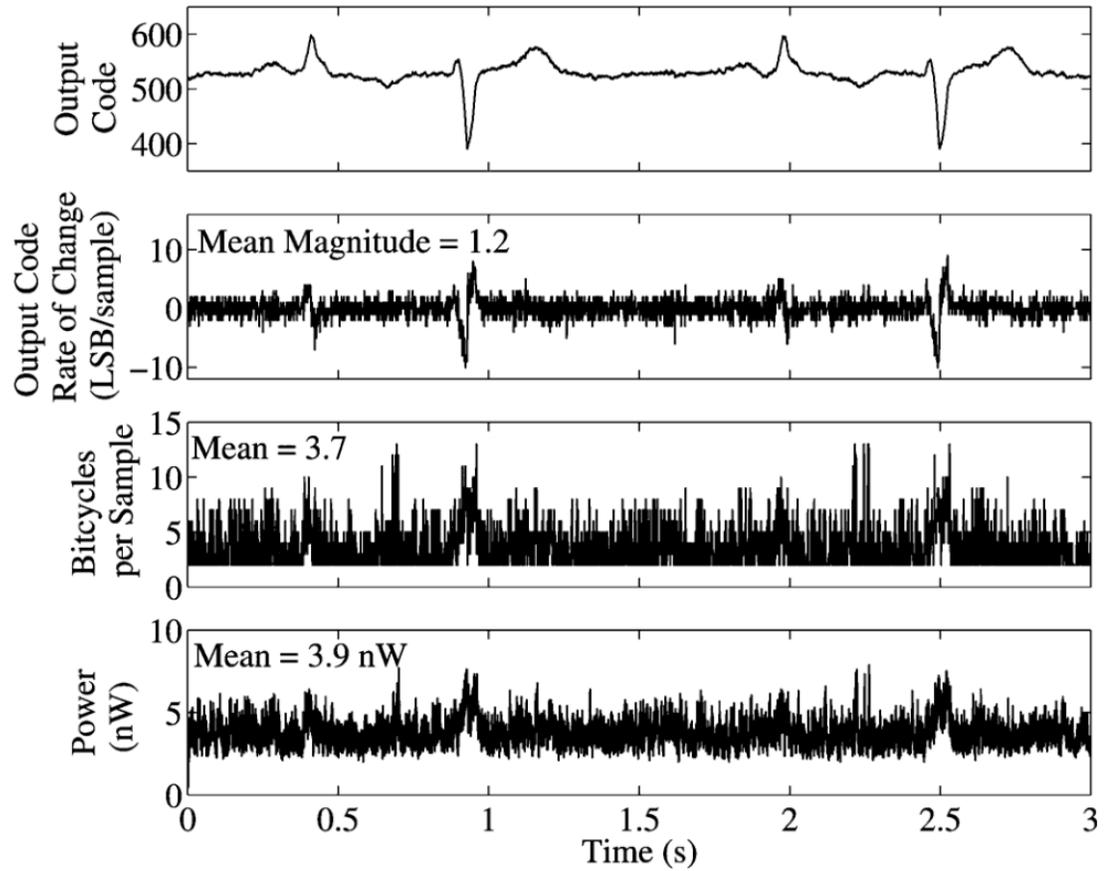


Figure 2.13: ADC response to ECG test input signal with $f_s = 10kHz$ and $V_{DD} = 0.5V$ [12]

Fig. 2.12 and Fig.2.13 are published results from a test chip showing its data-dependent energy reduction. The graphs shows the correlation between input activity and energy reduction.

2.2.1 Drawbacks of Bit-Skipping LSBFQ

Since it is intended for low-activity signals, the LSBFQ should require bitcycles and switching energy proportional to the initial code error ΔD_n . Figure 2.14a shows an example of the algorithm from [4], in which the small code change allows a 5-bit conversion is performed in 5 bitcycles. However, Figure 2.14b shows an example of an equally small code change that requires 7 bitcycles. This is because the least significant bit initially set to \overline{DIR} is $D[4]$, the toMSB phase skips over the lower four bits and causes a very large jump in the DAC output code.

On average, the bit-skipping LSBFQ saves bitcycles and switching energy for small code changes compared to a conventional SAR. However, there are still cases in which the algorithm consumes unnecessarily large energy and bitcycles for even small code steps.

Note that for different initial (i.e. previous) codes, the number of bitcycles can vary significantly even for the same code step. Fig. 2.15 shows the bitcycles required for a 10-bit LSBFQ with a full-scale input at $10^{-3}f_s$, where f_s is the sample rate. This input frequency is slow enough that the output code changes by three *LSB* or less each conversion. However, due to bit-skipping, a code change of one *LSB* can take anywhere between 2 and 10 bitcycles. The cases that require 10 bitcycles also consume much larger switching energy than the 2 bitcycle case.

D _{out} [n-1] = 8, D _{out} [n] = 10, DIR = 1			
Bitcycle	D[4:0]	CMP	ΔD
1	01000	0	---
2	01000	0	---
3	0100 1	0	+1
4	010 1 1	1	+2
5	0101 0	0	-1

} Initialization
 } toMSB
 } toLSB

(a)

D _{out} [n-1] = 15, D _{out} [n] = 17, DIR = 1			
Bitcycle	D[4:0]	CMP	ΔD
1	01111	0	---
2	01111	0	---
3	1 1111	1	+16
4	1 0 111	1	-8
5	10 0 11	1	-4
6	100 0 1	1	-2
7	1000 0	0	-1
7.1	1000 1	---	+1

} Initialization
 } toMSB
 } toLSB

(b)

Figure 2.14: Examples of bit-skipping LSBFQ, for (a) good case and (b) bad case.

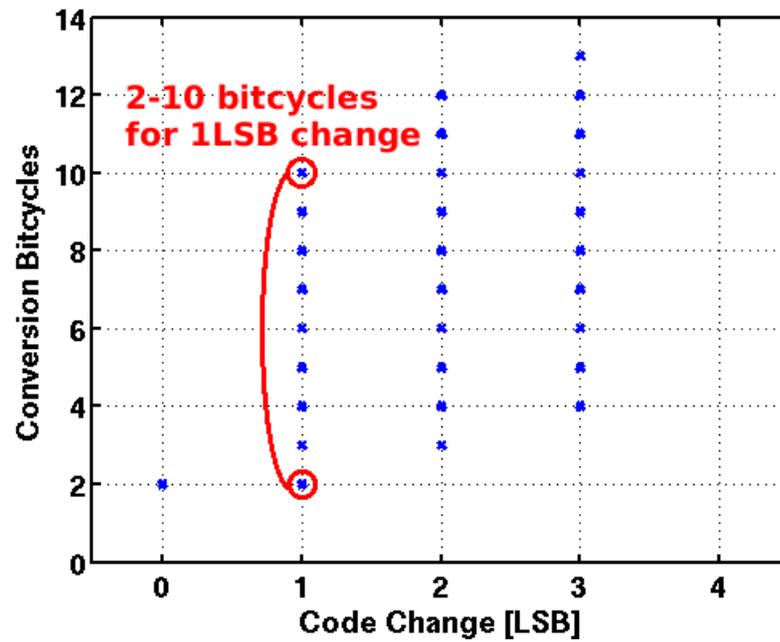


Figure 2.15: Number of bitcycles required as a function of ΔD_{out} in a 10-bit quantizer with input frequency of $10^{-3}f_s$, for bit-skipping LSBFQ.

Chapter 3: Bit-Repeating Least Significant Bit First SAR ADC

Although the bit-skipping LSBFQ saves bitcycles and switching energy on average, there are still instances when bitcycles and energy are disproportional and require much more for a small code change. For a 10-bit quantizer, a code change of one *LSB* in bit-skipping LSBFQ can take anywhere between 2 and 10 bitcycles.

3.1 Algorithm Improvements

The bit-repeating least significant bit first quantizer (LSBFQ) shows an improved solution that would avoid large bit-skips, such that the bitcycles and switching energy are *always* proportional to the code change ΔD_n rather than only being proportional on average.

3.2 Implementation Details

The proposed LSBFQ algorithm is illustrated in Fig. 4.2 and a visual representation is shown in Fig. 3.2. The modification is in the toMSB phase, which now begins with bit $D[0]$ regardless of whether it is initially set to *DIR* or not. In the case that the current bit under test, $D[Q]$, is already set to *DIR*, the new algorithm looks ahead to find the next bit that is set to \overline{DIR} . Define the index of this bit as X . The algorithm now sets $D[X] = DIR$ but also toggles bits $D[Q$ to $X-1]$

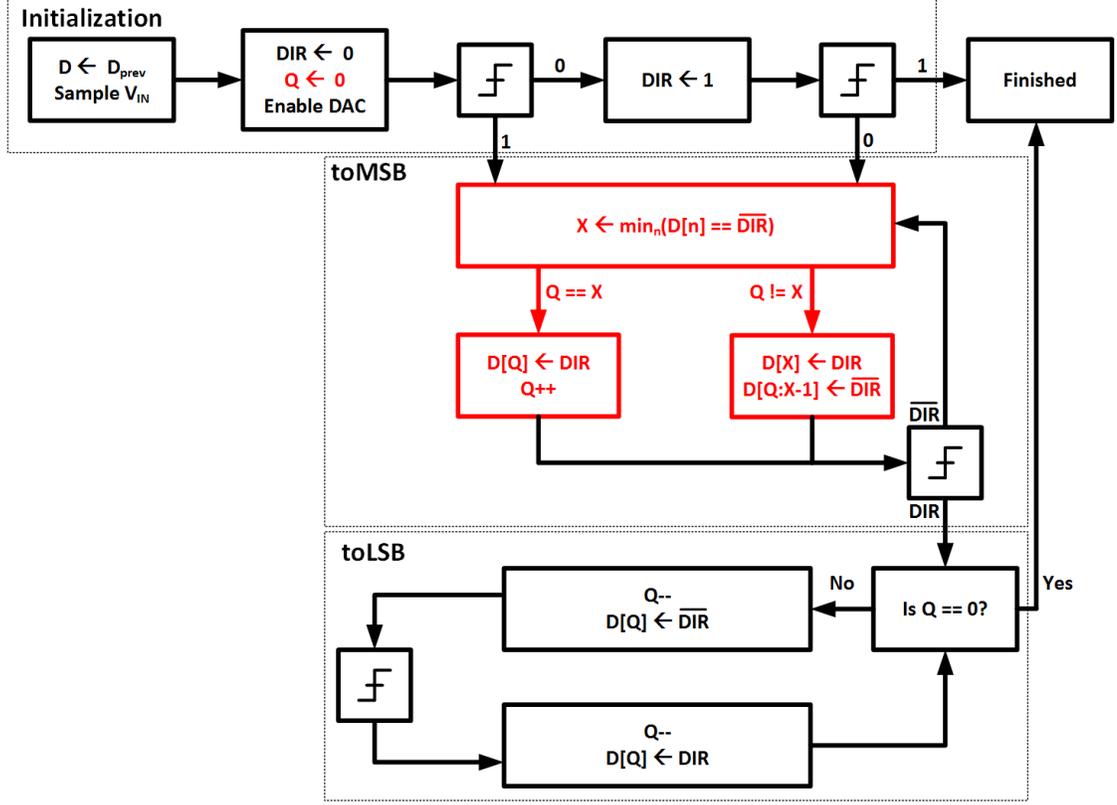


Figure 3.1: Bit-Repeating LSBFQ algorithm

back to $\overline{\text{DIR}}$. This means that the DAC output code step is the same as if only bit $D[Q]$ toggled. If all the capacitor switching occurs at the same time, the switching capacitance is equal to C_Q (the capacitance of bit $D[Q]$) [10].

Since the toLSB phase resembles conventional SAR switching and relies on all bits $D[0$ to $Q-1]$ being set in the same direction, the algorithm cannot leave $D[Q] = \overline{\text{DIR}}$. When the look-ahead action is performed as described above, the index of the bit under test (Q) does not increment after the comparison. Instead Q remains pointing to the same bit and toggles it again, repeating the same output

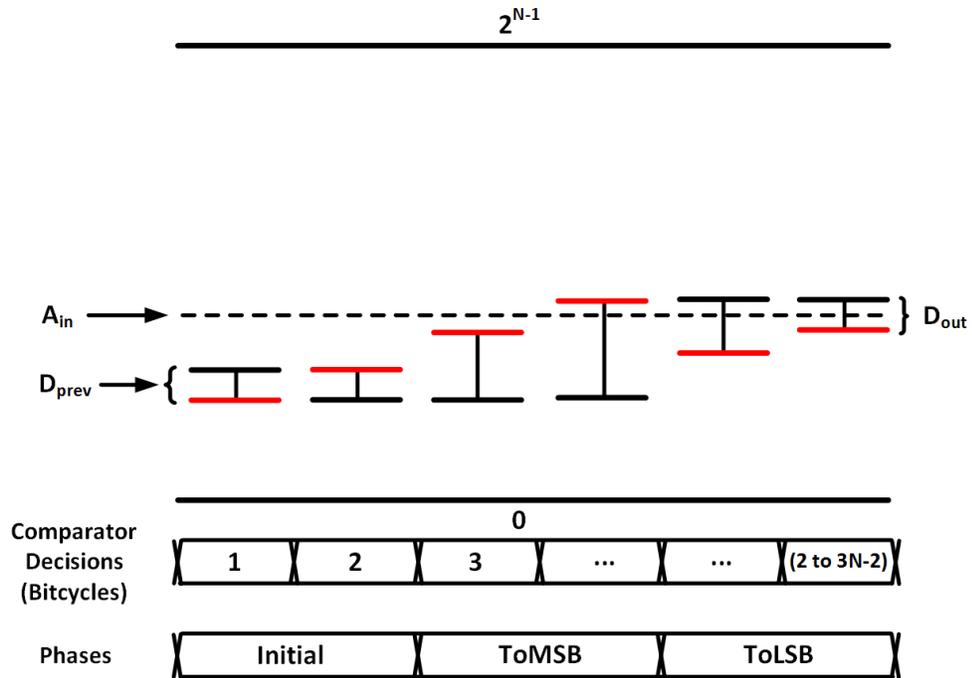


Figure 3.2: Bit-repeating LSBFQ Operation

code increment. As opposed to the bit-skipping that occurs in [4], this results in bit-repeating. For small code changes this ensures that the number of bitcycles is proportional to ΔD_n and is not skewed by bit-repeating (see Figure 3.4). It is true that in some cases when ΔD_n is large, this will require more bitcycles. But this is acceptable because large-activity signals do not fall within the target applications for LSBFQ.

Bit-repeating LSBFQ [1] prevents the bit skipping that occurs during the toMSB phase in the previous algorithm. This keeps the step size small and ensures maximum energy savings in low signal activities. An example of the same code conversion as Fig. 2.9 is shown in Fig. 3.3. Energy savings is shown in the ΔD

D _{out} [n-1] = 11, D _{out} [n] = 18, DIR = 1			
Bitcycle	D[4:0]	CMP	ΔD
1	01011	0	---
2	01011	0	---
3	01 100	0	+1
4	0110 1	0	+1
5	011 1 1	0	+2
6	100 11	1	+4
7	100 0 1	0	-2
8	1001 0	1	+1

} Initialization
 } toMSB
 } toLSB

Figure 3.3: Example of Bit-Repeating LSBFQ

column illustrating the switching activity in the DAC.

3.3 Results

Fig. 3.4 shows the bitcycles required for a 10-bit LSBFQ with a full-scale input at $10^{-3}f_s$, where f_s is the sample rate. The bit-repeating LSBFQ shows a lot smaller of a range with low activity signals.

Simulations were performed to compare the bitcycles and energy consumption of the proposed algorithm to the previously published bit-skipping LSBFQ method. A comparison to MCS quantization [7] is also included for reference.

Figure 3.5 illustrates the capacitor switching energy for sinusoids of varying amplitude, for both low-frequency and Nyquist rate inputs. Both LSB-first methods outperform MCS for low activity inputs (either low amplitude or low frequency). Since MCS is known to be 87.5% more efficient than conventional SAR

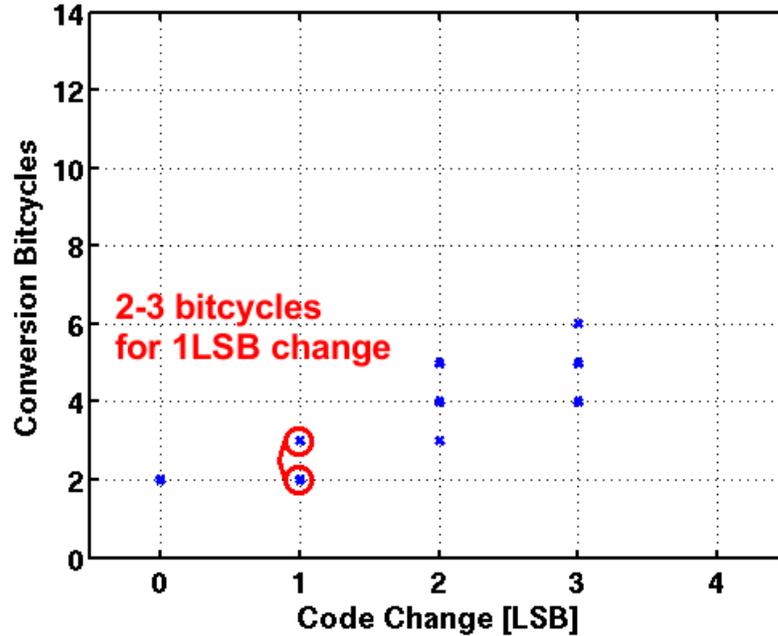
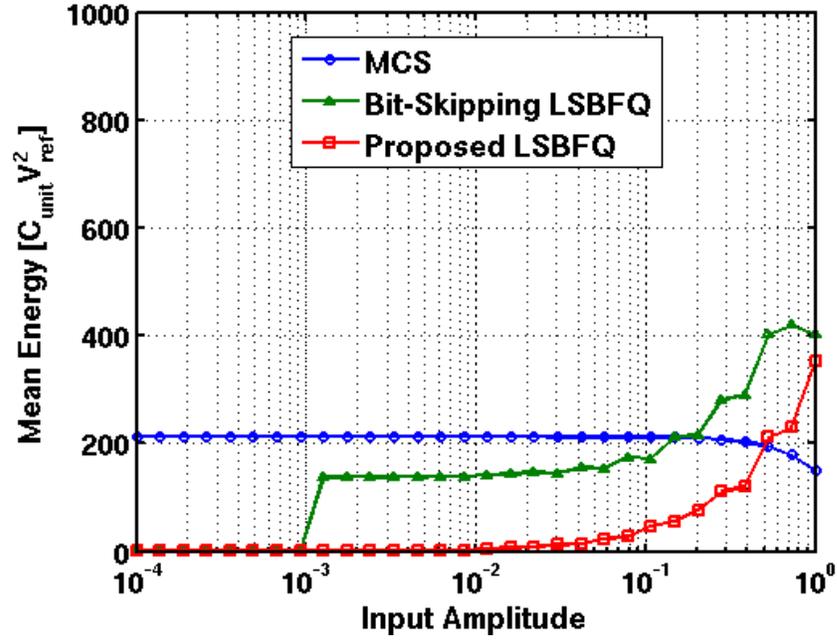


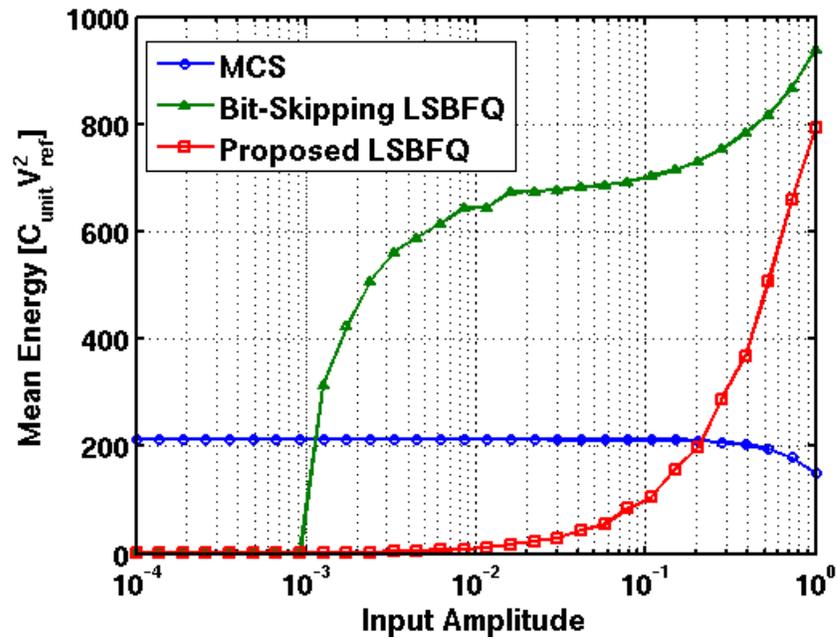
Figure 3.4: Number of bitcycles required as a function of ΔD_{out} in a 10-bit quantizer with input frequency of $10^{-3}f_s$, for bit-repeating LSBFQ.

switching [7], the comparison between LSBFQ and conventional switching would be even more favorable. Furthermore, the proposed bit-repeating algorithm uses less switching energy than the bit-skipping method for all cases.

Figure 3.6 shows that for low activity signals, bit-repeating saves bitcycles as well. There is a crossover point at which bit-repeating begins to require more bitcycles than bit-skipping, but these occur at high signal activity, for which LSB-first quantization would not be used in the first place.

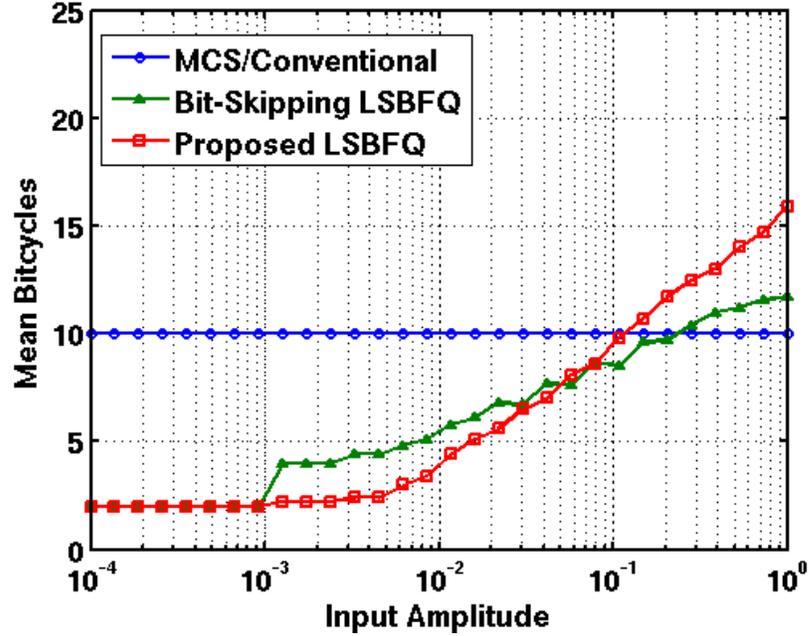


(a)

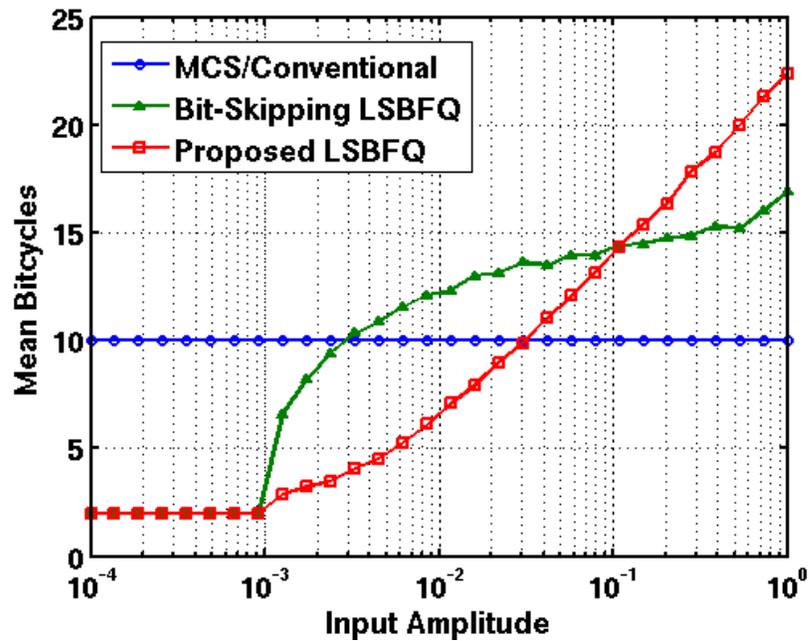


(b)

Figure 3.5: Capacitor switching energy as a function of input signal amplitude, for (a) $0.10f_s$ and (b) $0.49f_s$. The proposed algorithm is compared to MCS [5], and bit-skipping LSB-first [4] methods.



(a)



(b)

Figure 3.6: Successive-approximation bitcycles as a function of input signal amplitude, for (a) $0.10f_s$ and (b) $0.49f_s$. The proposed algorithm is compared to conventional [2], MCS [5], and bit-skipping LSB-first [4] methods.

Chapter 4: Selectable Starting Bit SAR ADC

With small code changes, the bit-skipping LSBFQ saves switching energy and bitcycles when compared to a conventional SAR on average [4]. However, there are cases in which the algorithm consumes a large amount of energy and bitcycles even with small code changes [12]. This was rectified in the bit-repeating LSBFQ [1] to ensure large code skips are avoided. Nonetheless, the application range of both these algorithms is limited to very low signal activity.

4.1 Algorithm Improvements

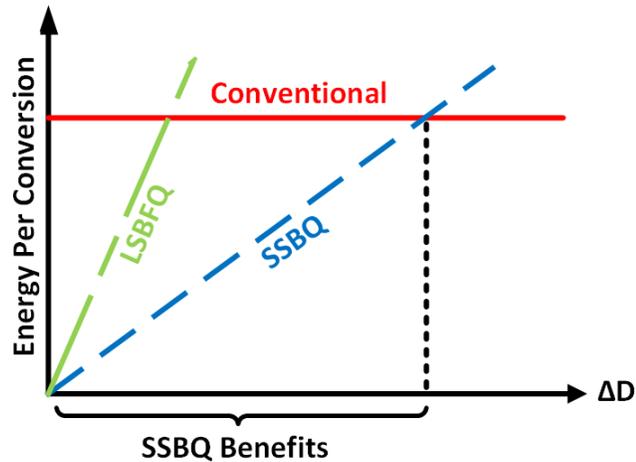


Figure 4.1: Target Applications for SSB SAR ADC

The selectable starting bit SAR (SSB) ADC improves upon previous works by

increasing design flexibility based on signal activity. The ADC modifies the SAR algorithm such that quantization is not limited to starting with the most-significant bit (MSB) as in traditional SARs or least-significant bit (LSB) as in LSBFQ, but any bit in between. This proposed method opens up design for maximum reductions in energy and comparator bitcycles for target applications. Fig. 4.1 shows how SSB SAR can extend the target applications for high activity signals. Furthermore, the proposed solution encompasses the bit-repeating LSBFQ since it is a specific case of the proposed solution.

Before diving into implementations, the motivation for this ADC must first be explained. For optimal operation with the minimum number of bitcycles, it is crucial that the algorithm takes as little bitcycles as possible in the toMSB phase. The toMSB phase exits when the correct starting bit is chosen such that the sampled input voltage is within the voltage range. This is shown back in LSBFQ in Fig. 2.10 and 3.2. Therefore to reduced the amount of bitcycles required, the starting voltage range should corresponds to signal activity. The SSB SAR takes advantage of this and is able to start with a variety of voltage ranges chosen by designer.

4.2 Implementation Details

The SAR structure for the selectable starting bit quantizer (SSBQ) is the same as the LSBFQ and strongly resembles that of a conventional SAR, with the exception of DAC control logic and one extra LSB capacitor [4] controlling the signal *DIR*.

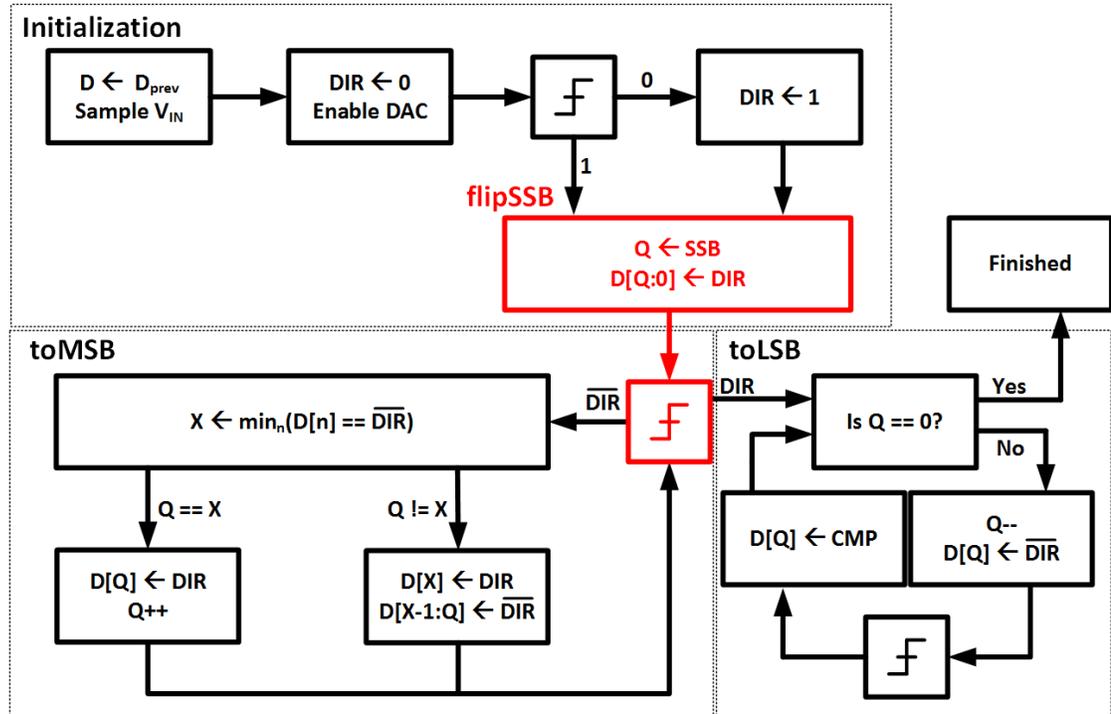


Figure 4.2: Proposed Selectable Starting Bit (SSB) algorithm.

The proposed SSB algorithm is illustrated in Fig. 4.2 and example conversions are shown in Fig. 4.3. The algorithm begins in the initialization phase with the previous output code as the initial estimate. **After the comparator resolves and DIR is determined, the designated starting bits down to the LSB $D[SSB : 0]$ are set to DIR .** Another comparison is made at the start of the toMSB phase. If the comparator output toggles, it means the final desired output code of the A/D conversion is in between the initial guess and the bits that flipped during the initialization phase. When this is occurs, the toMSB phase can be skipped and the algorithm goes directly to the toLSB phase and performs a conventional SAR operation from the starting bit back down to LSB. A numerical

D _{out} [n-1] = 11, D _{out} [n] = 18, DIR = 1				
Bitcycle	D[4:0]	CMP	ΔD	
1	01011	0	---	} Initialization
2	01111	0	+4	
3	10011	1	+4	} toMSB
4	10001	0	-2	
5	10010	1	1	} toLSB

(a)

D _{out} [n-1] = 14, D _{out} [n] = 11, DIR = 0				
Bitcycle	D[4:0]	CMP	ΔD	
1	01110	1	---	} Initialization
2	01000	0	-6	
3	01100	1	+4	} toLSB
4	01010	0	-2	
5	01011	0	+1	

(b)

Figure 4.3: Examples of proposed SSB algorithm with the SSB set to the third bit in a 5-bit quantizer. (a) example with the toMSB phase and (b) example skipping the toMSB phase.

example of these cases is shown in Fig. 4.3b and a visual example is shown in Fig. 4.4. However, when the comparator output does not toggle, the algorithm goes into the toMSB phase and the bit under test, $D[Q]$, heads from the starting bit towards the MSB.

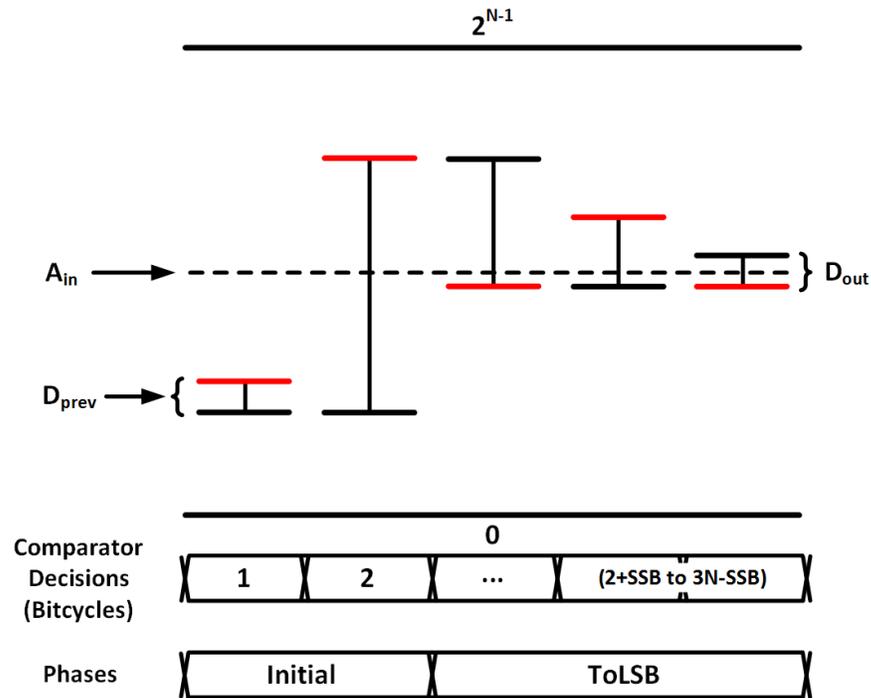


Figure 4.4: Selectable Starting Bit ADC Operation: skipping the toMSB phase

Taking into account DAC switching energy and bitcycle range, the toMSB phase implementation is similar to the bit-repeating LSBFQ [1]. In the bit-repeating LSBFQ, the algorithm looks ahead and ensures no large skips occur by switching the bit ahead $D[X]$ to DIR , while at the same time toggling the least significant bits back to \overline{DIR} . However, to minimize bitcycle range, the toMSB phase in the proposed SSB algorithm leaves the starting bits down to the LSB

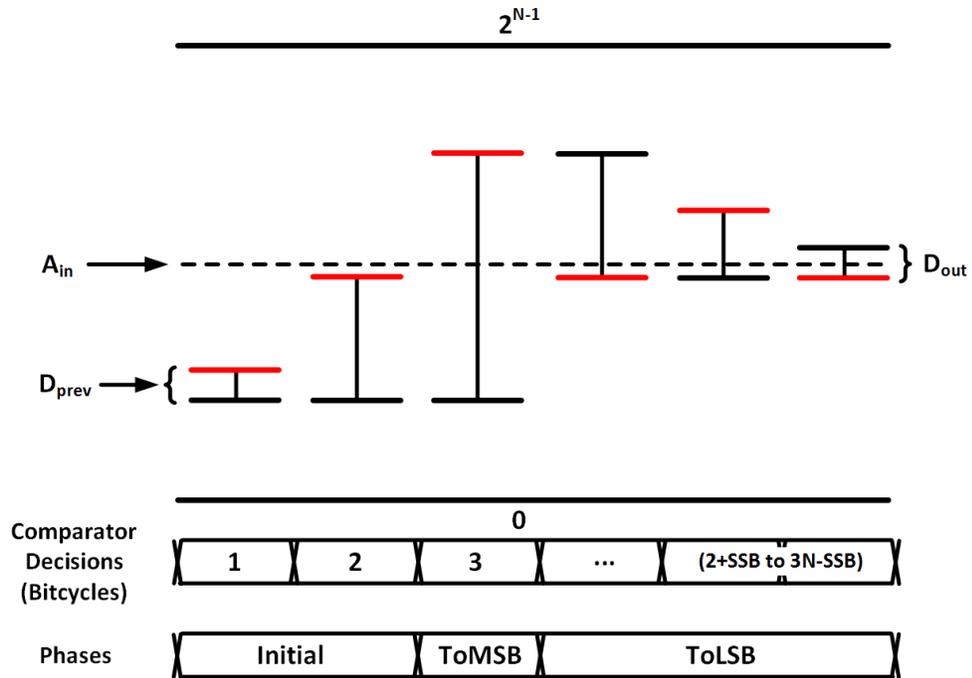


Figure 4.5: Selectable Starting Bit ADC Operation

alone and only toggles the bits from $D[SSB]$ to the bit ahead $D[X]$ back to \overline{DIR} . A numerical example of this operation is illustrated in Fig. 4.3a and the visual example is shown in Fig. 4.5.

The toMSB phase stops when the comparator output toggles. This is followed by the toLSB phase where it performs a conventional SAR operation from the prior bit under test $D[Q - 1]$ back to the LSB.

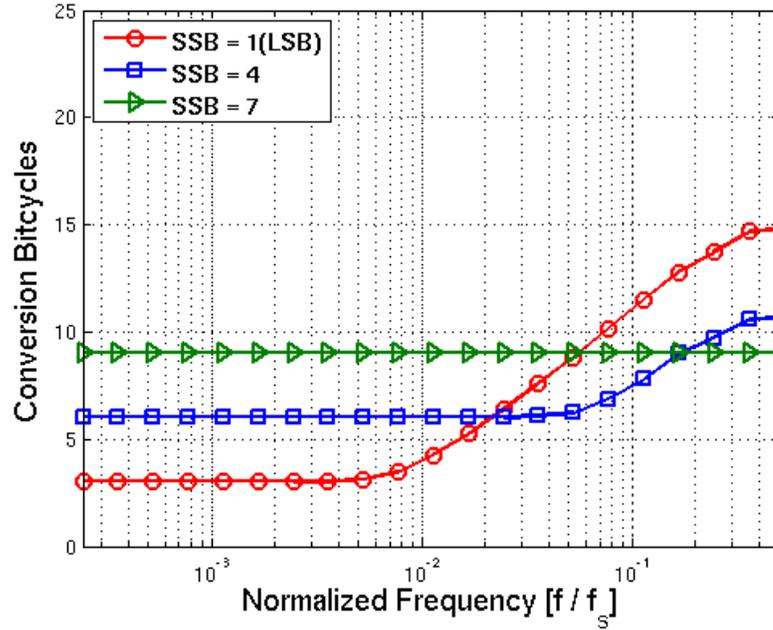
A toMSB flag is implemented to identify whether or not the algorithm entered the toMSB phase. This is used to optimize the algorithm and avoid unnecessary redundancy. This flag is noted as it will be used as a critical signal path in chapter 5.

Designers should configure the starting bit based on signal activity. When activity is low, sample voltages are close to each other, therefore starting bit should be close to LSBs to reflect a low voltage range. This will avoid having too large of a range to start with and will help save bitcycles during the toLSB phase. However with higher signal activity, the starting bit should be set close to the MSBs to avoid spending too many bitcycles in the toMSB phase. In an ideal situation, the starting bit should be set just high enough towards the MSB such that that the voltage range is just large enough to cover incoming sampled signal.

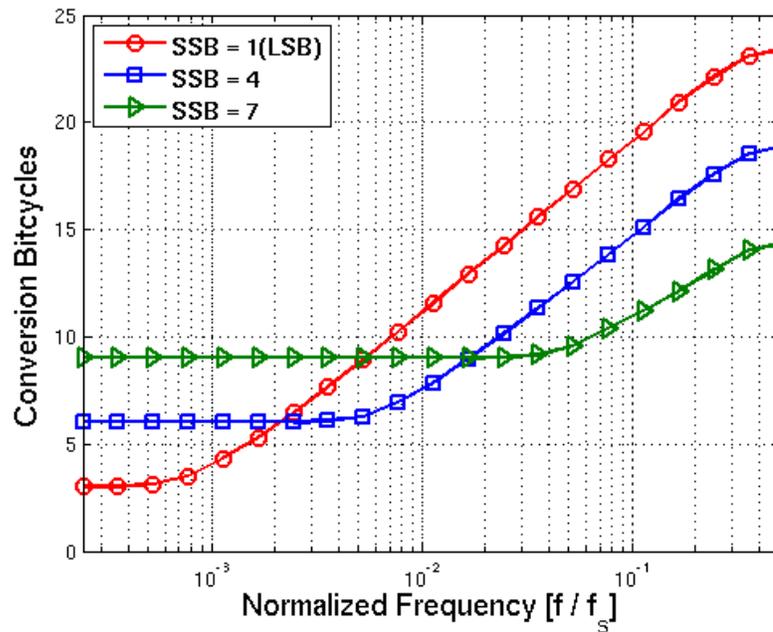
4.3 Results

There are two variables that determine signal activity: frequency and input amplitude with respect to full-scale. Since the intended purpose for the SSBQ is to show its increased bandwidth over LSBFQ to accommodate high activity signals, a frequency sweep is performed while holding amplitude constant.

Simulations with various bits designated as SSB were performed with the proposed algorithm. Let $SSB = 1$ signify toggling the LSB during initialization phase. Different SSBs were swept in a 10-bit quantizer to model the trend that occurs between bitcycles and switching energy. Fig. 4.6 shows the mean bitcycles it takes the A/D to perform a conversion, for both low amplitude and full-scale input up to Nyquist. As the SSB is set closer to the MSB, the overall mean bitcycles increases at low frequencies. However, there is an optimal region when the algorithm outperforms both the conventional and LSBFQ SAR. This is further shown in Fig.

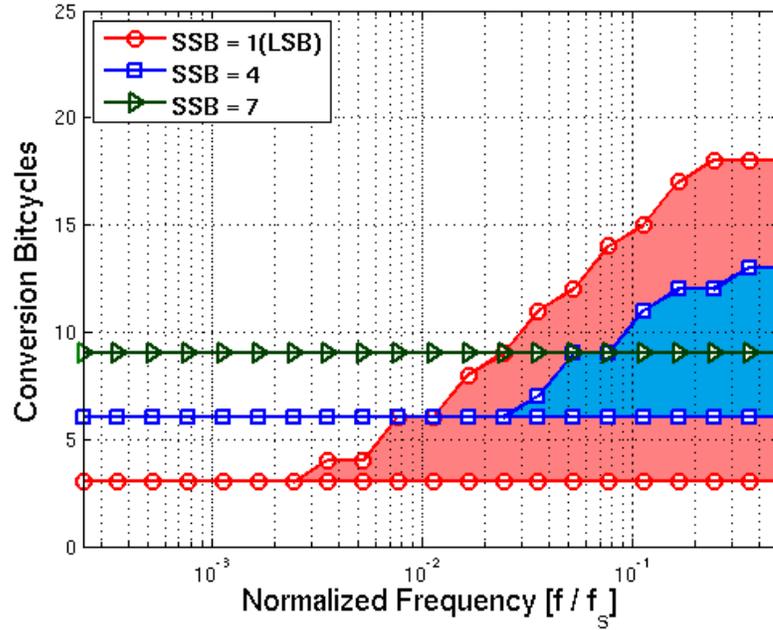


(a)

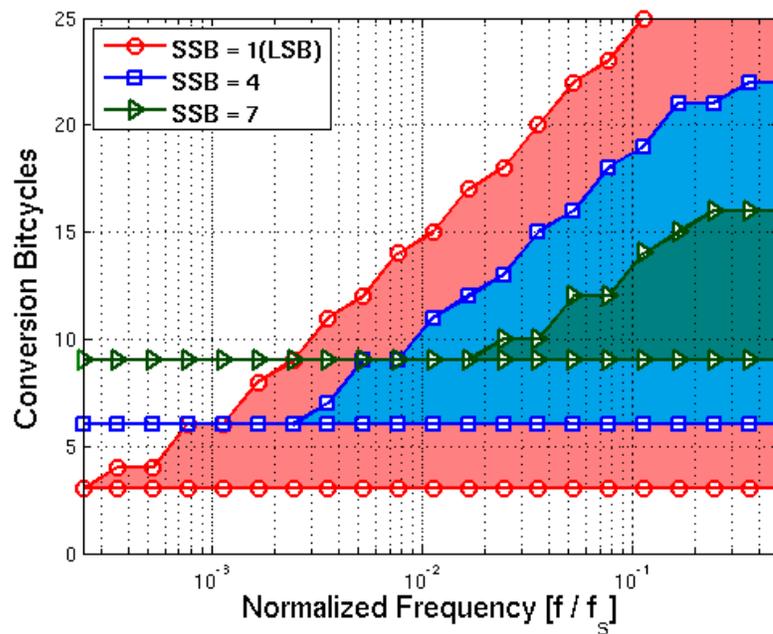


(b)

Figure 4.6: Mean number of bitcycles required by proposed algorithm as a function of frequency swept to Nyquist with various bits set as the SSB, for (a) 10% of fullscale input and (b) fullscale input.

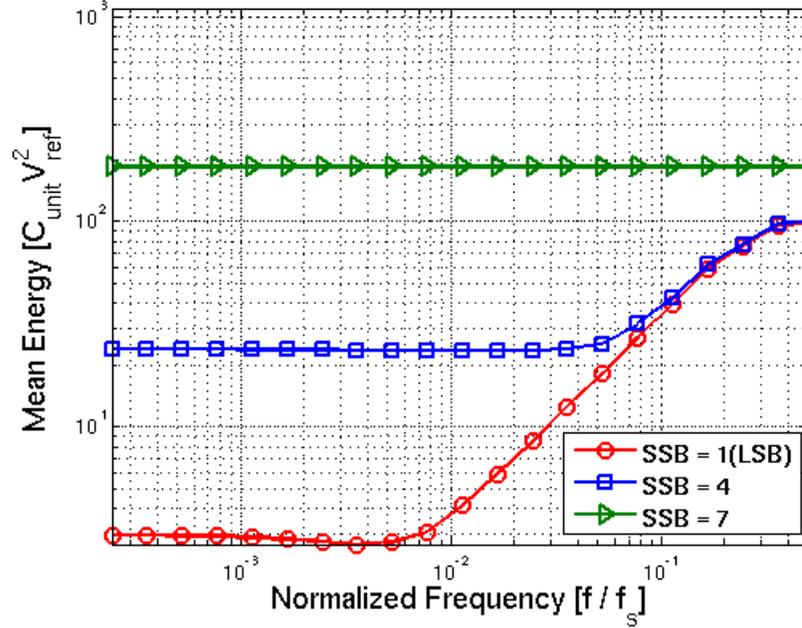


(a)

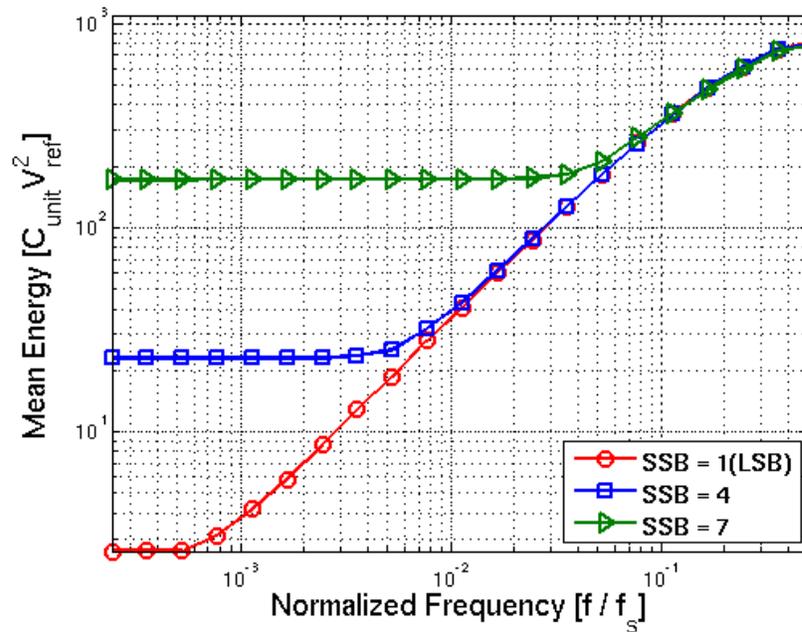


(b)

Figure 4.7: Range in number of bitcycles required by proposed algorithm as a function of frequency swept to Nyquist with various bits set as the SSB, for (a) 10% of fullscale input and (b) fullscale input.



(a)



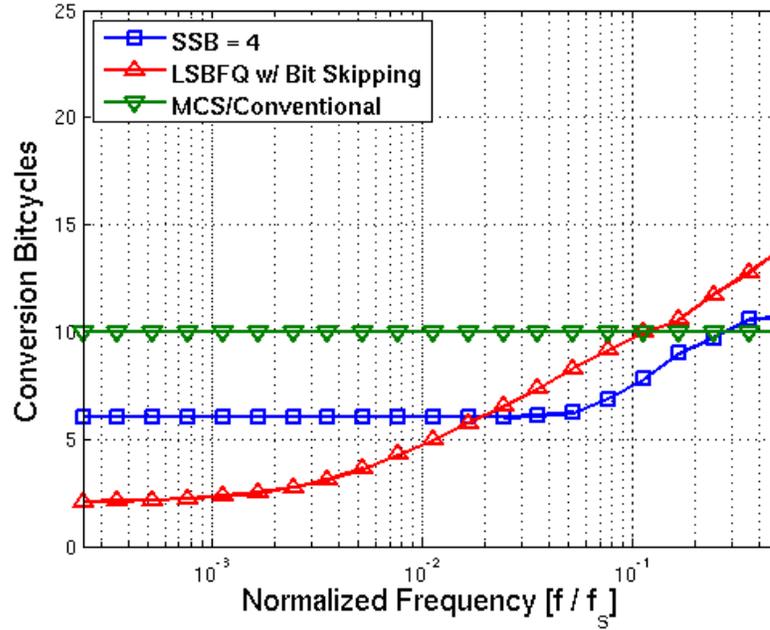
(b)

Figure 4.8: Capacitor switching energy required by proposed algorithm as a function of frequency swept to Nyquist with various bits set as the SSB, for (a) 10% of fullscale input and (b) fullscale input.

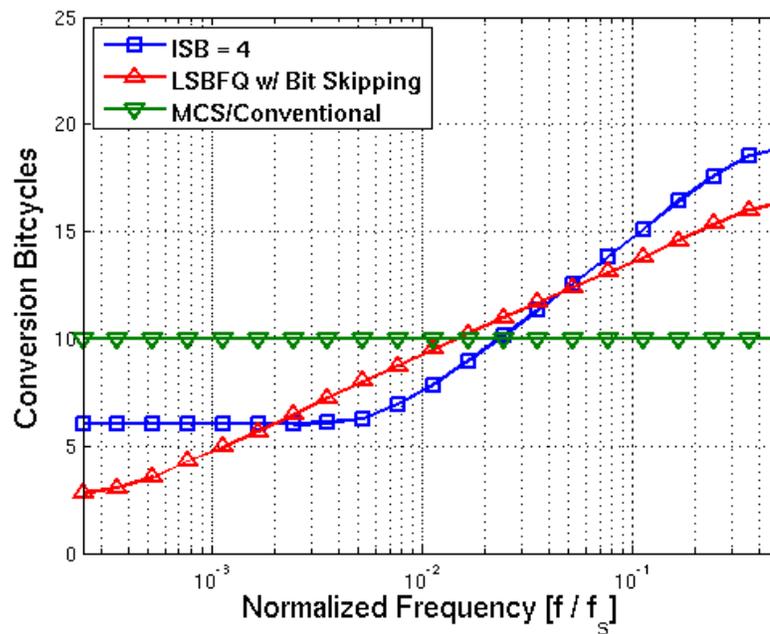
4.9a, where a direct comparison is made between the conventional/MCS and bit-skipping LSBFQ SAR ADC at 10% of full-scale input to reflect low signal activity. The figure illustrates the optimal region starts at around $0.02fs$ when SSB is set to 4. An important note is that the optimal region location can be shifted for target applications by changing the SSB index.

Alongside the mean, it is also important to show the minimum and maximum bitcycles required for the A/D to perform a conversion. Ultimately, the quantizer must account for the maximum number of bitcycles required when determining its conversion speed. Fig. 4.7 shows the minimum and maximum number of bitcycles required for a given SSB with the shaded area showing the range. As signal activity increases, the maximum bitcycles required in order to perform a conversion are higher. However, by changing the designated SSB closer to MSB, the range can be reduced and pushed out to higher frequencies.

Fig. 4.8 illustrates the capacitor switching energy for sinusoid of varying frequencies, for both low amplitude and full-scale input. As the SSB is set closer to the MSB, switching energy increases. A direct comparison is made in Fig. 4.10 between MCS and bit-skipping LSBFQ method at low activity inputs. Again, since MCS is known to be 87.5% more efficient than the conventional SAR switching [7], comparing the conventional switching would be even more favorable for the SSBQ. Using 10% of full-scale input to reflect low signal activity, there is a crossover point at around $0.01fs$ in which the SSBQ outperforms both the LSBFQ and the MCS SAR when SSB is set to 4. Again, this crossover point can be shifted by changing the SSB index.

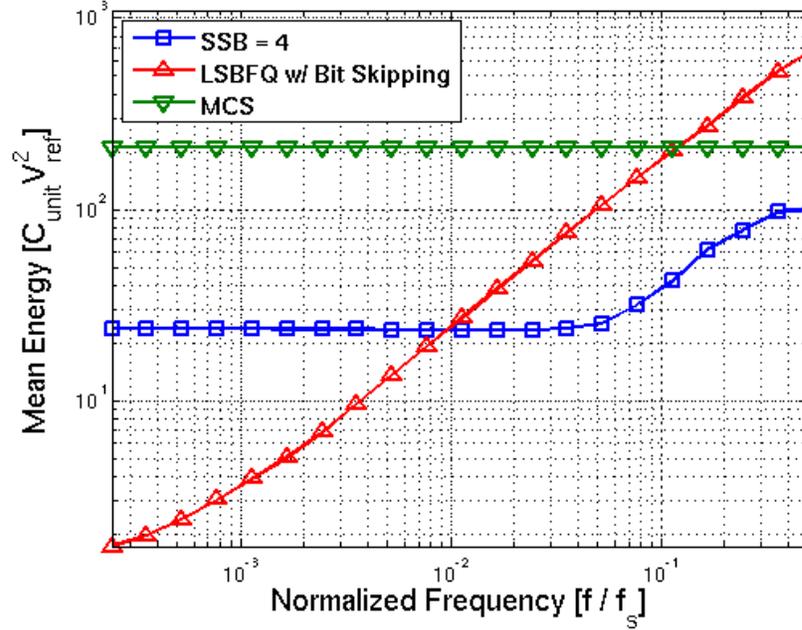


(a)

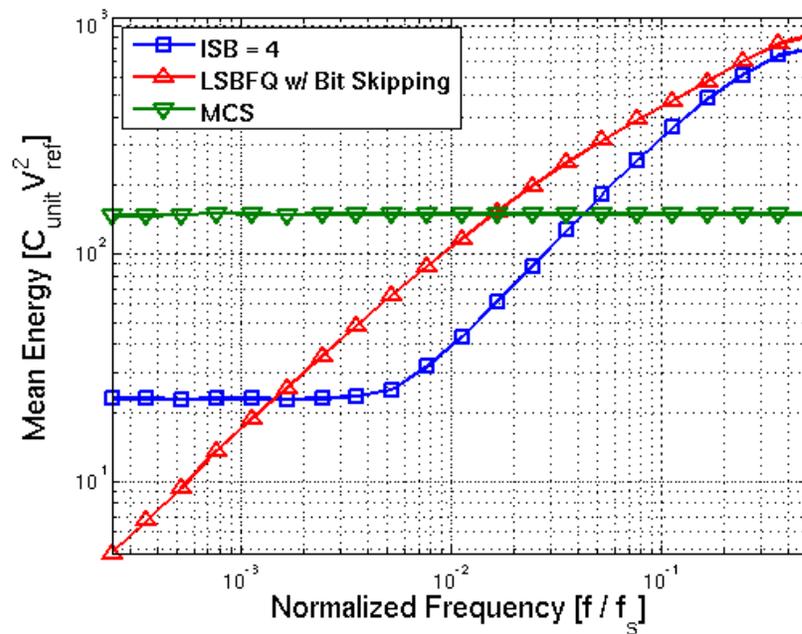


(b)

Figure 4.9: Comparisons between the SSBQ to MCS [5] and bit-skipping LSBFQ [4] for mean number of bitcycles. Comparisons are swept to Nyquist with (a) 10% fullscale input and (b) fullscale input.



(a)



(b)

Figure 4.10: Comparisons between the SSBQ to MCS [5] and bit-skipping LSBFQ [4] for capacitor switching energy. Comparisons are swept to Nyquist with (a) 10% fullscale input and (b) fullscale input.

Chapter 5: Adaptive Starting Bit SAR Quantizer

Least Significant Bit First Quantizers offer maximum energy savings but are limited to very low bandwidth. Selectable Starting Bit SAR ADC offers designers the flexibility to design for energy savings vs effective bandwidth. However, this must be chosen during the design phase and thus its maximum benefits are limited to signals with known characteristics. Is there a way for the quantizer to adapt to incoming signals such that the maximum benefits of SSB SAR can always be leveraged?

The answer to the rhetorical question is yes. An Adaptive Starting Bit SAR ADC implements a digital feedback to the SSB SAR such that the starting bit is calibrated based on signal activity.

5.1 Implementation Details

The structure for the ASB SAR is the same as SSB SAR with the addition of the feedback digital logic.

Using the pre-existing toMSB flag register already implemented in the SSB SAR, the feedback is implemented by keeping track of whether or not the algorithm enters the toMSB phase. During each conversion, if it is necessary to enter the toMSB phase, a flag is triggered. The ASB SAR utilizes this flag by implementing

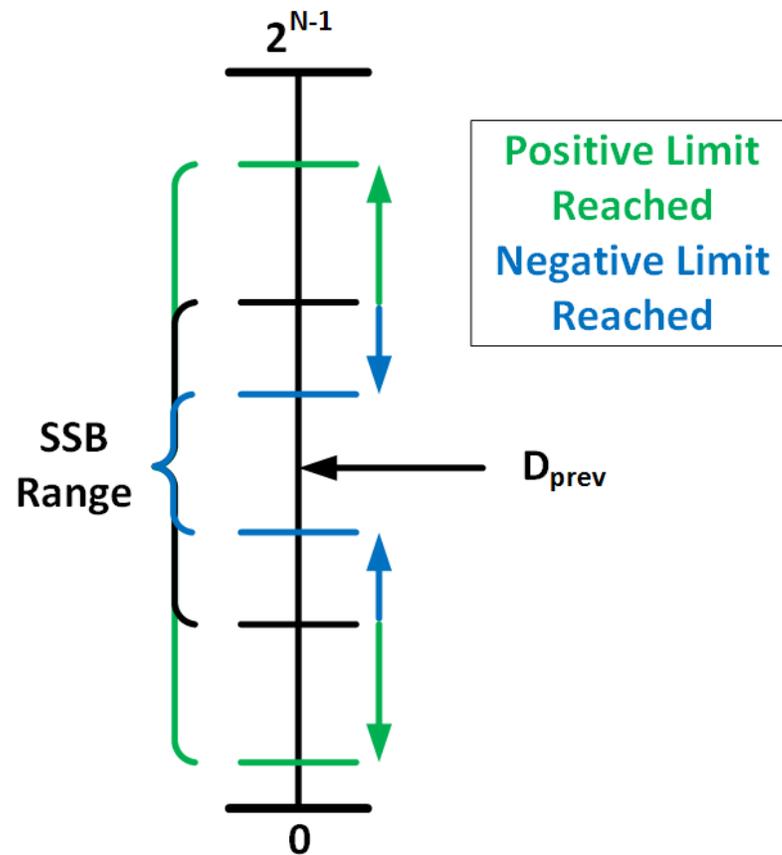


Figure 5.1: Adaptive Starting Bit ADC: feedback to adapt to the correct SSB range

a counter to keep track of the number of times the quantizer enters the toMSB phase. Used as digital feedback information for the ASB SAR, when the limit is reached in the counter, the starting bit in the ADC will be shifted appropriately to maximize its benefits based on input signal activity.

Fig. 5.1 is an illustration showing how the starting voltage range increases or decreases based on feedback from the counter. Every time the ADC enters

the toMSB phase, the counter increments. When the upper limit of the counter is reached, the starting bit will shift towards the MSBs and the voltage range is increased for future conversions. Vice versa, when the ADC skips the toMSB phase, the counter decrements. When the lower limit of the counter is reached, the starting bit will shift towards the LSBs and the voltage range is decreased for future conversions.

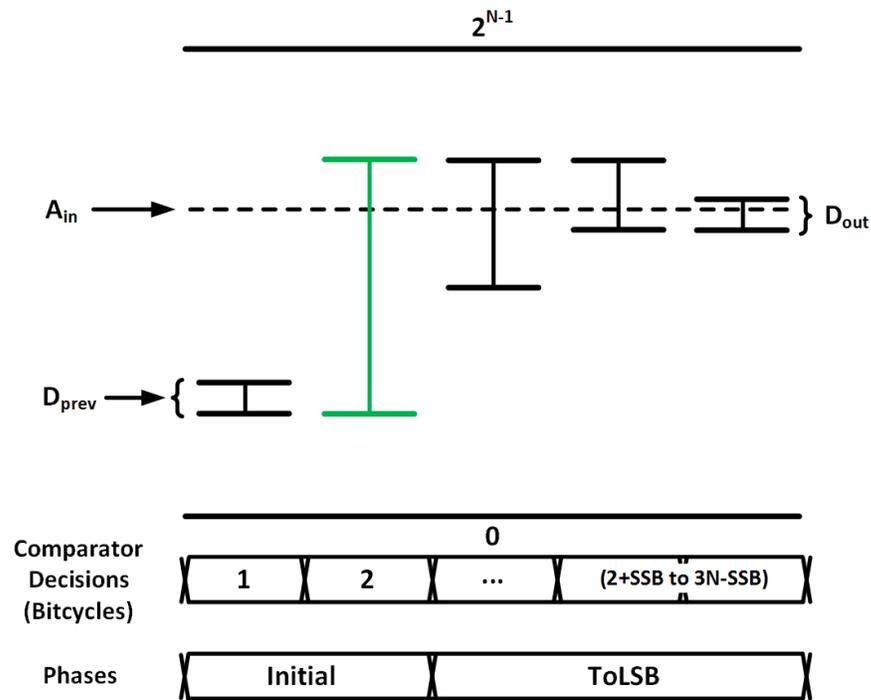


Figure 5.2: Adaptive Starting Bit ADC Operation: increasing the SSB range

Fig. 5.2 and 5.3 shows example conversions of the ASB SAR ADC. The examples show how the ADC will cater towards the input signal to minimize the number of bitcycles required by shifting the starting bit to find the optimal voltage range for the conversion.

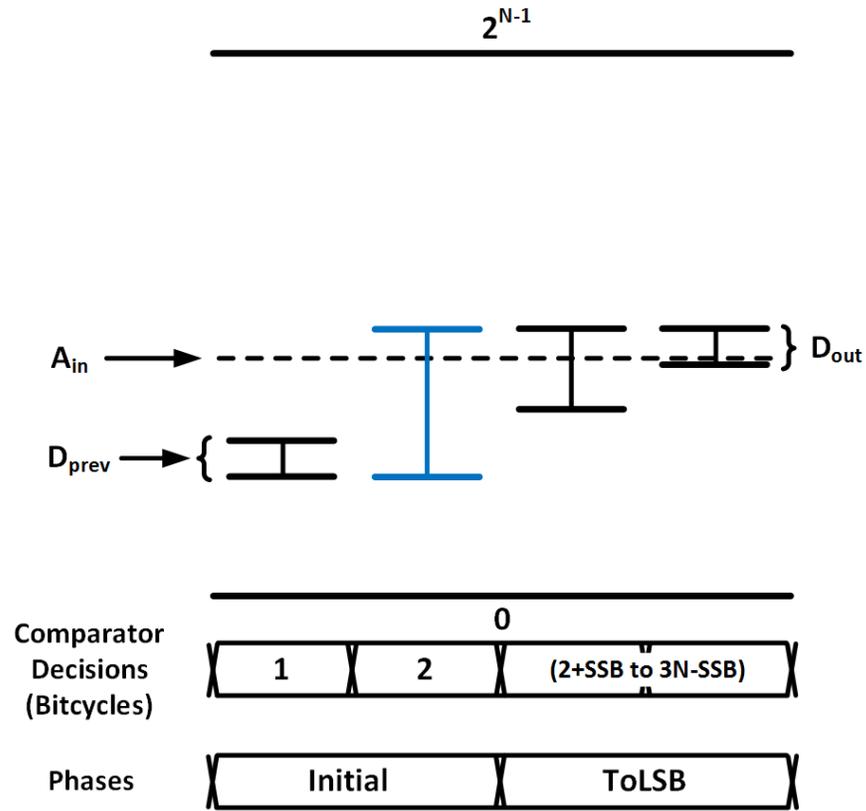


Figure 5.3: Adaptive Starting Bit ADC Operation: decreasing the SSB range

Counter limit is a design parameter in the ASB SAR. Having a low counter limit allows faster feedback to the system such that the starting voltage range can be adjusted frequently. However, counter limit should be no less than ± 2 to avoid instances when the MSBs are switched even though code changes are small, such as $0111 \rightarrow 1000$ in this 4-bit example.

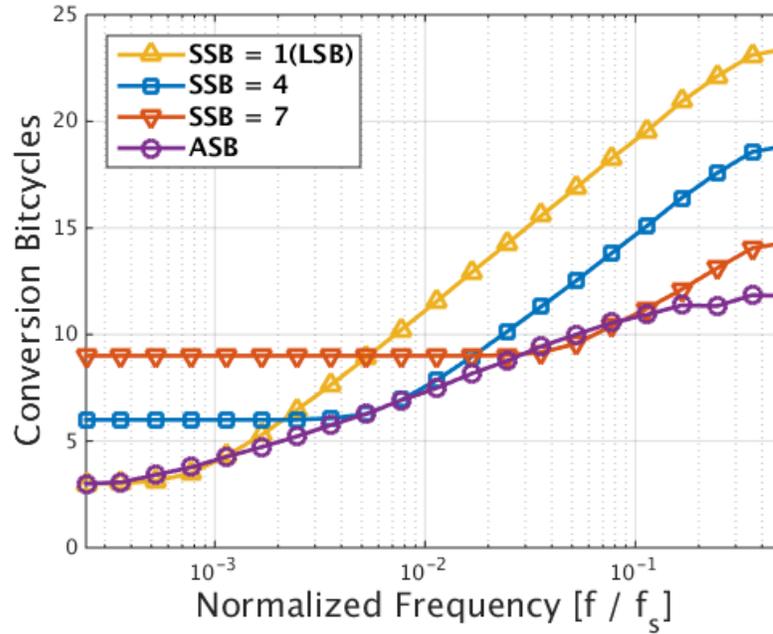
When a high counter limit is set, the feedback to the system will occur more slowly and the starting bit will shift less often. This should be chosen in systems where incoming signals are noisy and undesirable voltage spikes can cause MSBs to

toggle. Having a high counter limit serves as a buffer to avoid shifting the starting bit when these voltage spikes occur.

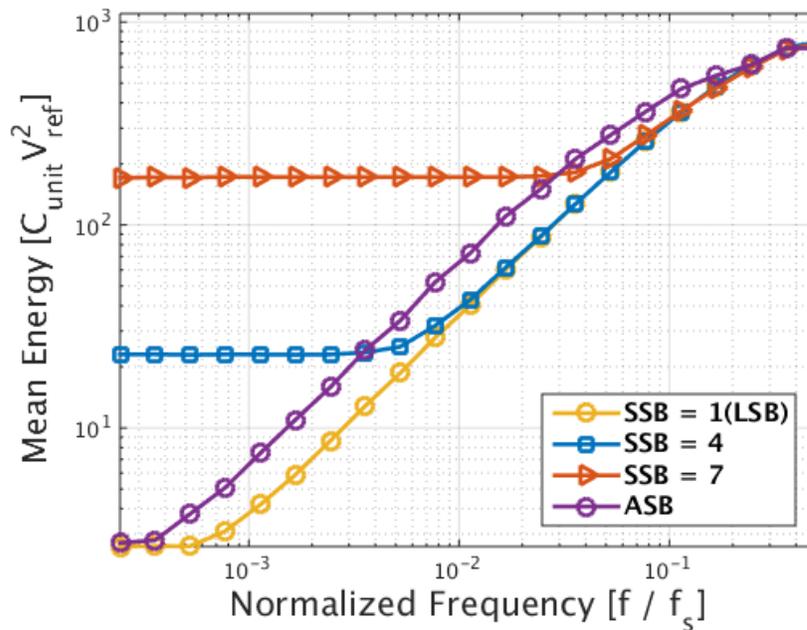
5.2 Results

The following results show how the ASB ADC compares to the previous architectures. Fig. 5.4a and 5.4b shows the comparison between the ASB and SSB selected at various bits. For Fig. 5.4a, the ASB leverages the peak performance for all the various starting bits in the SSB. As for Fig. 5.4b, the energy curve for the ASB follows the $SSB = 1(\text{LSB})$ curve, with the exception slightly higher energy consumption. This is due to the fact that the current implementation of ASB will adapt to the appropriate starting bit, but will never settle to the best starting bit. The starting bit for the ASB will always toggle between the best starting bit and the bit that is adjacent to it. Since the ASB theoretically is on the best starting bit half the time, and the bit next to the best starting bit the other half of the time, the ASB will require slightly more energy compared to when the SSB when the starting bit is set to the LSB.

Fig. 5.5a, 5.5b, and 5.6 shows the comparisons between the ASB, bit-skipping LSBFQ, and MCS for mean number of bitcycles, max number of bitcycles, and switching energy respectively. As shown, the ASB outperforms the bit-skipping LSBFQ in all three comparisons and extended the crossover point with MCS.

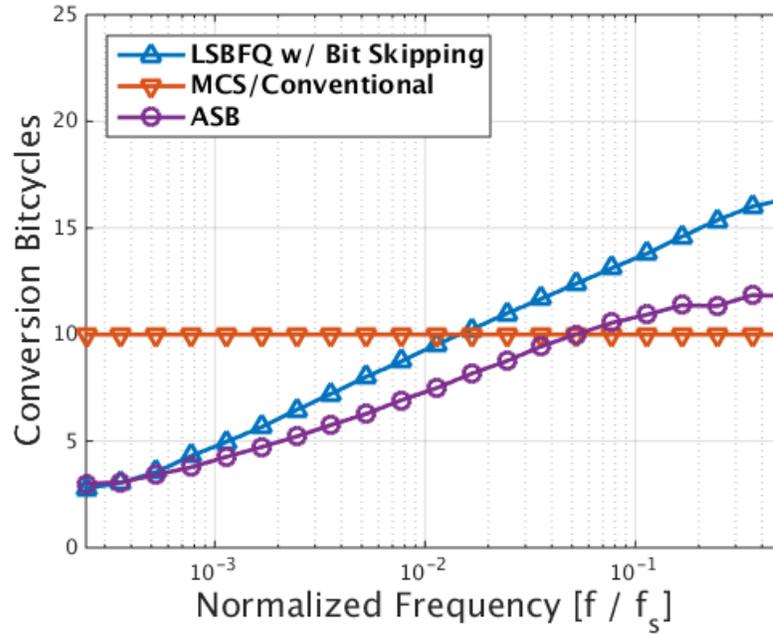


(a)

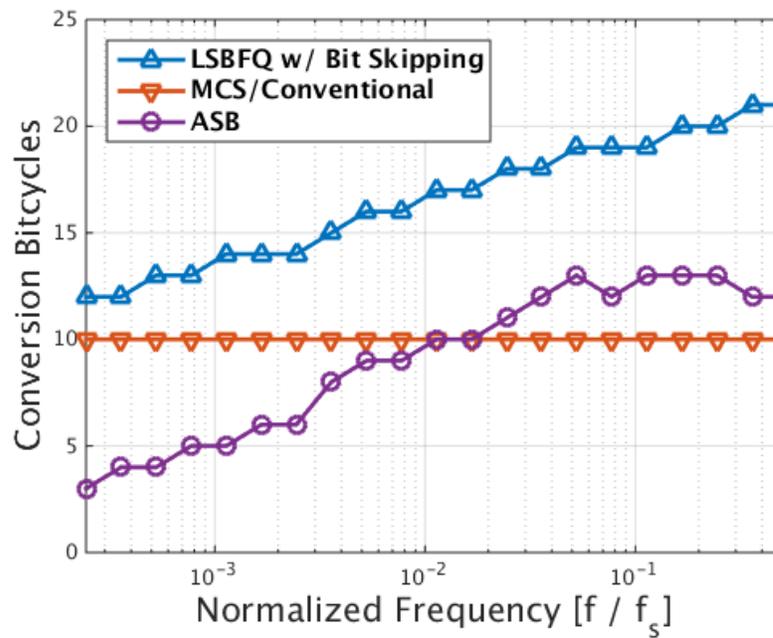


(b)

Figure 5.4: Comparisons between ASB and SSB ADC with various bits set using full-scale input for (a) mean bitcycle and (b) switching energy.



(a)



(b)

Figure 5.5: Bitcycle comparisons between ASB ADC, bit-skipping LSBFQ, and MCS using full-scale input for (a) mean and (b) max bitcycles.

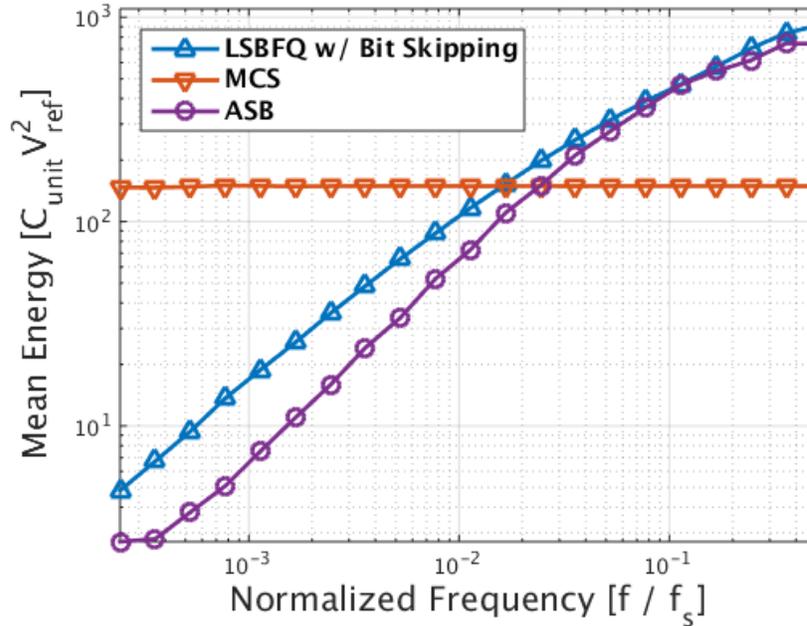


Figure 5.6: Switching energy comparison between ASB ADC, bit-skipping LSBFQ, and MCS using full-scale input.

5.3 Future Direction

With the results of the ASB SAR outperforming the bit-skipping LSBFQ in all cases, there are numerous future directions that can further add benefits to the ASB SAR.

One obvious improvement to the ASB is to implement a stable region in the feedback such that the ASB will not toggle between the best starting bit and the bit adjacent to it. This will lower the switching energy required to match the energy curve of when the starting bit is set to the LSB in SSB, the best case scenario for switching energy consumption. Possible implementation to achieve this can be

to use a weighted averaging system. With a weighted averaging implementation, determining the starting bit will not only be dependent on a number of prior samples, but a numerical averaged value can be stored from all previous samples. This will provide a more desirable starting bit for signals with consistent activity. However, for signals such as ECG where there are short bursts of high activity followed by long periods low activity, this will have less of an advantage. Another method is to add a check to the system to signal whether or not toggling the starting bit in the initial phase is the correct decision or not. If the starting bit toggles back after the initial phase, it means the starting bit can be shifted towards the LSB phase, and the counter should decrement. On the other hand, if the starting bit does not toggle back, it means the starting bit is correct and the counter should do nothing. Keeping the implementation of incrementing the counter when the ADC enters the toMSB phase, the system now has a way to shift the starting bit towards the MSB when activity is high, shift towards the LSB when activity is low, and maintain the current starting bit when the starting bit is correct.

Another possible improvement is to implement a switch such that if signal activity is high and the benefits of ASB SAR cannot be leveraged, the logic will switch to the conventional MSB SAR. This will provide maximum energy savings with all range of signal activity.

Another direction to take is to use the quantizer in a delta sigma modulator. Since delta sigma modulators require oversampling such that the effective bandwidth is lowered,

$$EffectiveBandwidth = \frac{SamplingFrequency}{2 \times OSR} \quad (5.1)$$

there can be potential to use the ASB SAR in a delta sigma modulator due to its high performance with low power consumption in low bandwidth signals.

Chapter 6: Conclusion

LSBFQ methods are effective solutions for processing low activity signals. However, applications are bandwidth limited and lack the ability to optimize with various range of signal activity. The proposed SSB SAR allows designers to choose between bandwidth and energy savings based on the designated SSB. Furthermore, this method encompasses the LSBFQ with bit-repeating, as setting the SSB to the LSB would produce essentially the same result. When the SSB is set closer to the MSB, bitcycle range is reduced and an optimal region occurs. This region with the proposed method outperforms both LSBFQ and MCS SAR in terms of average bitcycle and energy consumption. However, the SSB SAR ADC must be previously set, therefore applications are only useful when characteristics of the incoming signal is known. The ASB SAR ADC will adapt to any incoming signal such that the best starting bit will be used. Simulation results have proven that the ASB will reduce even more bitcycles and switching energy compared to the bit-skipping LSBFQ, and extended the crossover point for which it outperforms MCS. Considering that MCS is the most energy-efficient of the traditional MSB-first SAR, the SSB algorithm is the most promising solution for moderate to low activity applications.

Bibliography

- [1] A. Waters, J. Leung, U. Moon. LSB-First SAR ADC with Bit-Repeating for Reduced Energy Consumption. In *Proc. IEEE Int. Conf. on Electronics, Circuits and Syst.*, 2014.
- [2] B. Ginsburg and A. Chandrakasan. An Energy-Efficient Charge Recycling Approach for a SAR Converter With Capacitive DAC. In *Proc. IEEE Int. Symp. on Circuits and Syst.*, pages 184–187, 2005.
- [3] B. Murmann. "ADC Performance Survey 1997-2014," [Online]. Available: <http://web.stanford.edu/~murmam/adcsurvey.html>.
- [4] F. Yaul and A. Chandrakasan. A 10b 0.6nW SAR ADC with Data-Dependent Energy Savings Using LSB-First Successive Approximation. In *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, pages 198–199, 2014.
- [5] H. Venkatram, J. Guerber, S.-H. Lee, and U. Moon. Merged capacitor switching based SAR ADC with highest switching energy-efficiency. *IET Electron. Lett.*, 46(4):620–621, April 2010.
- [6] J. Guerber. Time and Statistical Information Utilization in High Efficiency Sub-Micron CMOS Successive Approximation Analog to Digital Converters, 2012.
- [7] J. Guerber, H. Venkatram, T. Oh, and U. Moon. Enhanced SAR ADC Energy Efficiency from the Early Reset Merged Capacitor Switching Algorithm. In *Proc. IEEE Int. Symp. on Circuits and Syst.*, pages 2361–2364, 2012.
- [8] D. Johns and K. Martin. *Analog Integrated Circuit Design*. Wiley and Sons, Toronto, Canada, 1997.
- [9] Maxim Integrated. "Understanding SAR ADCs," [Online]. Available: <http://www.maximintegrated.com/en/app-notes/index.mvp/id/1080>.
- [10] J.L. McCreary and P.R. Gray. All-mos charge redistribution analog-to-digital conversion techniques. i. *Solid-State Circuits, IEEE Journal of*, 10(6):371–379, Dec 1975.

- [11] R. Schreier and G. Temes. *Understanding Delta-Sigma Data Converters*. IEEE Press, 2005.
- [12] F.M. Yaul and A.P. Chandrakasan. A 10-bit sar adc with data-dependent energy reduction using lsb-first successive approximation. *Solid-State Circuits, IEEE Journal of*, Early Access, Dec 2014.
- [13] R.F. Yazicioglu, Sunyoung Kim, T. Torfs, Hyejung Kim, and C. Van Hoof. A $30\ \mu\text{w}$ analog signal processor asic for portable biopotential signal monitoring. *Solid-State Circuits, IEEE Journal of*, 46(1):209–223, Jan 2011.
- [14] 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. *Solid-State Circuits, IEEE Journal of*, 48(6):1453–1464, June 2013.

