A

Thesis Report

On

DIGITAL CALIBRATION OF 1.5 BIT PER STAGE PIPELINED ADC

Submitted towards the fulfilment of requirement for the award of degree of

Master of Engineering

In

Electronics and Communication Engineering

Submitted by:

Jupinder Kaur

Roll No: 801461012

Under the Guidance of:

Dr. Alpana Agarwal

Associate Professor

ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

(Established under the section 3 of UGC Act, 1956)

PATIALA – 147004 (PUNJAB)

Certificate

Certified that the thesis entitled **"Digital Calibration of 1.5 Bit Per Stage Pipelined ADC"** being submitted by **Ms. Jupinder Kaur** to the **Department of Electronics and Communication Engineering, Thapar University, Patiala** in the fulfilment of the requirements for the award of the degree of **"Master of Engineering"** is a record of bona fide research work carried out by her. She has worked under my guidance and supervision and fulfilled the requirements for the submission of this thesis which has reached the requisite standard. The matter presented in this thesis does not incorporate any material previously published or written by any other person except where due reference is made in the text.

The results contained in this thesis have not been submitted in part or full to any other institute or university for the award of degree or diploma.

(Alpana Agarwal)
Associate Professor

Declaration

I hereby declare that the thesis report entitled **"Digital Calibration of 1.5 Bit Per Stage Pipelined ADC"** is an authentic record of my study carried out as requirement for the award of degree of ME (Electronics and Communication Engineering) at Thapar University, Patiala, under the supervision of **Dr. Alpana Agarwal, Associate Professor**, Electronics and Communication Engineering Department, during master's degree.

Date: 15-07-2016

(Jupinder Kaur)
Roll No: 801461012

It is certified that the above statement made by the student is correct to the best of my knowledge and belief.

Date: 15-07-2016

(Alpana Agarwal)
Associate Professor
ECED

Countersigned by:

Head, ECED
Thapar University, Patiala
Date:

Dean of Academic affairs
Thapar University, Patiala
Date:

Acknowledgement

I would like to express my gratitude to **Dr. Alpana Agarwal, Associate Professor,** Electronics and Communication Engineering Department, Thapar University, Patiala for her patient guidance and support throughout the work. Also, valuable inputs provided by Mr. Anil Singh at every step have been of great help.

I am also thankful to **Dr. Sanjay Sharma, Professor and Head, Dr. Amit Kumar Kohli, P.G. Coordinator** and entire faculty and staff of Electronics and Communication Engineering Department for their encouragement and cooperation.

The financial support provided by Department of Electronics & Information Technology (DeitY), GOI through SMDP-VLSI C2SD project is gratefully acknowledged.

Jupinder Kaur

ME-ECE

801461012

Abstract

An analog to digital converter (ADC) is a mixed signal electronic device that converts the amplitude of input physical quantity *i.e.* either voltage or current into a digital representation. This digital representation is in the form of a stream of bits which is easy to store, can be processed using digital signal processing techniques, and can be converted back to analog after processing.

Out of various types of ADCs, pipelined ADC is used where high sampling rate and medium to high resolution is required. Pipelined ADC performs analog to digital conversion stage by stage, resolving some bits in each stage. All stages in a pipelined ADC are identical except the last one which is a low resolution flash ADC. Pipelined ADC becomes imperfect due to imperfections in analog circuit components like capacitor mismatch, op-amp finite open loop gain, finite unity gain bandwidth *etc*.

This work focuses on correction of gain error caused due to finite open loop gain of op-amp. The technique suggested in this work is a foreground digital calibration technique based on least mean squares (LMS) algorithm. Simulation results prove that after calibration of 12-bit, 1.5 bit per stage pipelined ADC, differential non linearity (DNL) improves by 30%. Integral non linearity (INL) reduces from values 60/-60 LSB to +0.77/-0.77 LSB. Also, signal to noise and distortion ratio (SNDR) and spurious free dynamic range (SFDR) improve significantly from 35.9193 dB and 36.7348 dB to 75.3619 dB and 82.2884 dB respectively after calibration.

Table of Contents

1. INTRODUCTION 1

1.1 ARCHITECTURE OF PIPELINED ADC 2
1.2 DATA LATENCY IN PIPELINED ADC 5
1.3 COMMON SOURCES OF ERRORS IN PIPELINED ADC 6
1.4 PIPELINED ADC PERFORMANCE METRICS 7

1.4.1 Static Performance Metrics 8

Quantization error 8
Offset error 9
Gain error 11
Non-Linearity 11
Differential non-linearity or differential linearity error 12
Integral non-linearity or integral linearity error 12
Total unadjusted error 13

1.4.2 Dynamic Performance Metrics 13

Signal to noise ratio (SNR) 13
Total harmonic distortion (THD) 13
Spurious free dynamic range (SFDR) 14
Signal to noise and distortion ratio (SNDR or SINAD) 14
Effective number of bits (ENOB) 14

1.5 CALIBRATION METHODS FOR PIPELINED ADC 14

Calibration in analog domain 15
Calibration in digital domain 15
Foreground calibration 15
Background calibration 15

2. LITERATURE REVIEW 17

3. PROBLEM STATEMENT 25

3.1 PROBLEM STATEMENT 25
3.2 PROBLEM EXPLANATION 25

4. FOREGROUND CALIBRATION METHOD 28

4.1 METHOD AND MATLAB IMPLEMENTATION 28
4.2 VERILOG IMPLEMENTATION 32

5. SIMULATION RESULTS 39

5.1 MATLAB SIMULATION RESULTS 39
5.2 VERILOG SIMULATION RESULTS 42

6. CONCLUSION AND FUTURE SCOPE 45

REFERENCES 47

PUBLICATION 50

List of Acronyms

ADC	Analog to Digital Converter
DAC	Digital to Analog Converter
LSB	Least Significant Bit
UGB	Unity Gain Bandwidth
DNL	Differential Non Linearity
INL	Integral Non Linearity
DC	Direct Current
FFT	Fast Fourier Transform
SNR	Signal to Noise Ratio
THD	Total Harmonic Distortion
SFDR	Spurious Free Dynamic Range
SNDR	Signal to Noise and Distortion Ratio
SINAD	SIgnal to Noise And Distortion ratio
ENOB	Effective Number Of Bits
LMS	Least Mean Squares
DBGE	Decision Boundary Gap Estimation
CMOS	Complementary Metal Oxide Semiconductor
MDAC	Multiplying Digital to Analog Converter
MATLAB	MATrix LABoratory
FSM	Finite State Machine
FPGA	Field Programmable Gate Array
LUT	Look Up Table
FF	Flip Flop
IOB	Input/Output Buffer
BUFG	Global Buffer

List of Figures

Figure 1.1: Comparison of various ADCs in terms of resolution (y-axis) and sample rate (x-axis) [31].....1
Figure 1.2: 1.5 bit per stage, N-bit pipelined ADC.....2
Figure 1.3: Architecture of a 1.5 bit pipelined stage.....3
Figure 1.4: Input (x-axis) vs residue (y-axis) plot for first pipelined stage [32] 4
Figure 1.5: Redundancy removal in 4-bit, 1.5 bit per stage pipelined ADC 5
Figure 1.6: Data Latency in pipelined ADC [33] 5
Figure 1.7: Fully differential op-amp based 1.5 bit pipelined stage [30] 6
Figure 1.8: Transfer function of an ideal 3-bit ADC [34].....8
Figure 1.9: Quantization error in ideal 3-bit ADC [34] 9
Figure 1.10: Quantization error in 3-bit ADC with added half LSB offset to input [34] 9
Figure 1.11: Zero scale offset error in ADC transfer function [34].....10
Figure 1.12: Full scale offset error in ADC transfer function [34].....10
Figure 1.13: Gain offset error in ADC transfer function [34].....11
Figure 1.14: Differential non-linearity error and missing code [34] 12
Figure 1.15: Integral non-linearity error [34] 12
Figure 1.16: Residue plot of first pipelined stage with real op-amp of open loop gain 40 V/V 26
Figure 1.17: Residue plot of second pipelined stage with real op-amp of open loop gain 40 V/V in first and second stages 26
Figure 1.18: Transfer function of a 12-bit, 1.5 bit per stage pipelined ADC with missing codes due to gain error in all pipelined stages.....27
Figure 1.19. i^{th} stage calibration of N=12-bit pipelined ADC using two extra stages for calibration (N_{cali}=15).....29
Figure 1.20. Convergence of weight (w_1) of first pipelined stage for different values of step size (μ).....31
Figure 1.21. Flow chart representing the proposed LMS based calibration algorithm.....31
Figure 1.22: Inputs and outputs of calibration process, and memory interface.....32
Figure 1.23: Calibration controller FSM 33
Figure 1.24: FSM for Vbei or Vtoti calculation 35
Figure 1.25: FSM for LMS equation for weight update 36
Figure 1.26: 21-bit floating point representation 37

Figure 1.27: Residue plot of first pipelined stage of ideal 12-bit, 1.5-bit per stage pipelined ADC.....39
Figure 1.28: Residue plot of second pipelined stage of ideal 12-bit, 1.5-bit per stage pipelined ADC.....39
Figure 1.29: Residue plot of first pipelined stage with op-amp open loop gain of 40 V/V 40
Figure 1.30: Residue plot of second pipelined stage with op-amp open loop gain of 40 V/V in both, first and second stages.....40
Figure 1.31: DNL and INL of 12-bit, 1.5 bit per stage pipelined ADC, with gain error in all pipelined stages 41
Figure 1.32: DNL and INL of 12-bit, 1.5 bit per stage pipelined ADC, with gain error in all pipelined stages 41
Figure 1.33. Frequency response of uncalibrated ADC.....41
Figure 1.34. Frequency response of calibrated ADC.....42
Figure 1.35. Verilog simulation waveforms 42
Figure 1.36: Sign bit, exponent bits, mantissa bits of weight obtained after calibration...43
Figure 1.37: Input clock signal 44

List of Tables

Table 1.1: Decision of output bits (B_1B_0), d_k and V_{dac} in single pipelined stage.....3
Table 4.1: Different states of the main calibration controller FSM and their functionality 34
Table 4.2: Different states of the FSM for V_{bei} calculation and their functionality.....35
Table 4.3: Different states of the LMS weight update FSM and their functionality 36
Table 5.1: Device utilization of the calibration logic for FPGA SP605 evaluation kit 44
Table 6.1: Comparison of this work to prior works on 12-bit pipelined ADC calibration 45

Chapter 1

INTRODUCTION

An analog to digital converter is a mixed signal electronic device that converts the amplitude of input physical quantity *i.e.* either voltage or current into a digital representation. This digital representation is in the form of a stream of bits which is easy to store, can be processed using digital signal processing techniques, and can be converted back to analog after processing.

Various types of ADCs are brought into use depending on requirement of resolution and speed. Figure 1.1 demonstrates resolution and speed of different types of ADCs. In applications that require good resolution as well as high speed, pipelined ADCs are used.

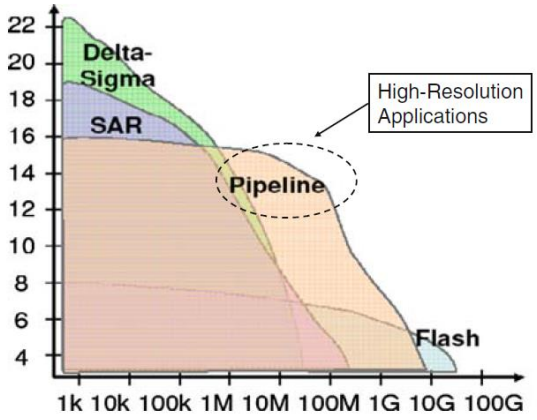

Figure 1.1: Comparison of various ADCs in terms of resolution (y-axis) and sample rate (x-axis) [31]

Ideally, an ADC should be capable of representing the amplitude of physical quantity in digital form accurately. Practically, a good and accurate ADC should perform the conversion task with only a negligible error. Accuracy of an ADC is limited by a number of factors that cause errors explained in further sections. These errors in a pipelined ADC can be removed in analog domain by modifying the design of analog components used in ADC. But it makes the circuit complex, consumes more die area and increases power consumption. Another way is to correct ADC non idealities in digital domain. Digital methods require additional circuitry external to ADC. But such methods consume lesser power and die area than analog calibration methods do.

1.1 Architecture of Pipelined ADC

Pipelining means assigning a certain task to a number of sub systems that can operate concurrently. Thus in a pipelined ADC, there are a number of cascaded stages operating in parallel on different samples of input. While one stage works on one sample, its preceding stage works on next sample. So at any instant of time after some latency, all the pipelined stages in an *N*-bit pipelined ADC would work on *N* different samples concurrently. This makes the ADC more time efficient allowing greater throughput as compared to non-pipelined architectures.

A sample value taken at a particular instance is fed to first stage of pipelined ADC. Each pipelined stage provides a certain number of significant bits along with one overlapping bit. In addition, a residue voltage is obtained as output from each stage. Residue obtained from first stage is fed as input to the second stage. Similarly residue from second stage is given as input to third stage, and so on. The overlapping bit is added to the first output bit of next stage to get a final stream of bits as ADC output. In this work 1.5 bit per stage pipelined ADC is used in which each stage provides one significant bit and one overlapping bit (represented as .5 bit), along with residue voltage.

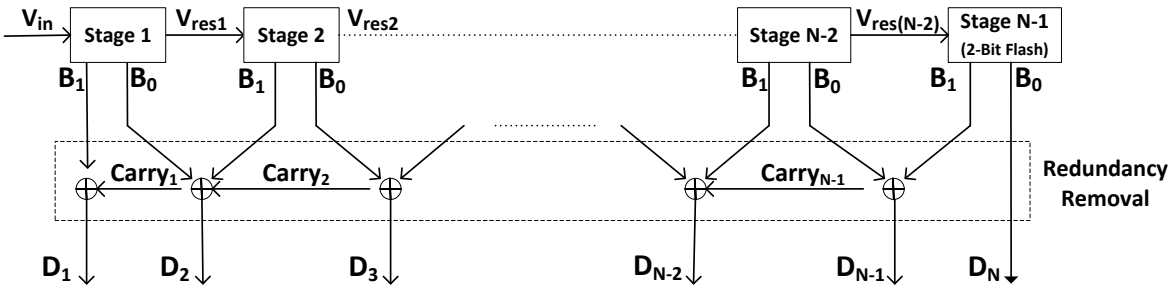

Figure 1.2: 1.5 bit per stage, *N*-bit pipelined ADC

Figure 1.2 shows a 1.5 bit per stage, *N*-Bit pipelined ADC. It consists of *N-2* pipelined stages, and a 2-bit flash ADC as last stage. Architecture of single pipelined stage is shown in figure 1.3. It can be noted that a 1.5 bit pipelined stage includes a sample and hold circuit, a 2-bit sub-ADC which provides two bits as stage output, a 2-bit sub-DAC that reconverts two output bits into analog, a subtractor that subtracts sub-DAC output from current stage input, and a controlled gain amplifier.

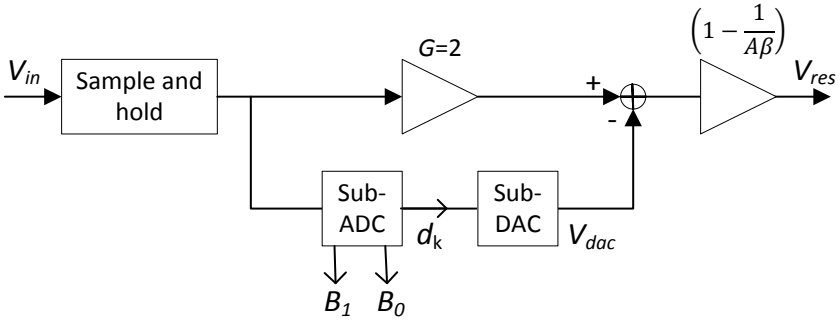

Figure 1.3: Architecture of a 1.5 bit pipelined stage

The sub-ADC in pipelined stage provides output bits (say B_1B_0) as 00/01/10, and decision values (d_k) -1/0/1 according to the range in which input sample value lies. Sub-DAC in each stage produces analog equivalent V_{dac} of bits given by Sub-ADC. V_{dac} can take a value among $-V_{ref}/0/V_{ref,}$, where V_{ref} represents ADC reference voltage. Apart from two bits, each pipelined stage provides residue voltage at output according to equation (1.1)

$$V_{res} = (G \times V_{in}) - V_{dac} \tag{1.1}$$

where V_{in} represents input voltage to a pipelined stage and G is the gain of input amplifier (refer figure 1.3). Gain G is typically chosen to be 2^B where B is number of useful bits per stage. Hence in this case $G = 2^1 = 2$ ideally. Table 1.1 explains how bits B_1B_0 and sub-DAC output V_{dac} are decided.

Condition	**Bits (B_1B_0)**	**d_k**	**V_{dac}**
$V_{in} < \frac{-V_{ref}}{4}$	00	-1	$-V_{ref}$
$\frac{-V_{ref}}{4} \leq V_{in} < \frac{V_{ref}}{4}$	01	0	0
$V_{in} \geq \frac{V_{ref}}{4}$	10	1	$+V_{ref}$

Table 1.1: Decision of output bits (B_1B_0), d_k and V_{dac} in single pipelined stage

To understand residue calculation, let us rewrite equation (1.1) for first stage residue,

$$V_{res1} = (G \times V_{in}) - V_{dac}$$
$$\Rightarrow V_{res1} = (G \times V_{in}) - \left(d_1 \times V_{ref}\right) \tag{1.2}$$

Residue from second stage is calculated as,

$$V_{res2} = G \times \Big((G \times V_{in}) - \big(d_1 \times V_{ref}\big)\Big) - \big(d_2 \times V_{ref}\big) \tag{1.3}$$

Similarly, residue from third stage is given as,

$$V_{res3} = G \times \Big(G \times \Big((G \times V_{in}) - \big(d_1 \times V_{ref}\big)\Big) - \big(d_2 \times V_{ref}\big)\Big) - \big(d_3 \times V_{ref}\big)$$

$$=> V_{res3} = G^3.(V_{in}) - [G^2.d_1 + G^1.d_2 + G^0.d_3]\big(V_{ref}\big) \tag{1.4}$$

Divide both sides by $\frac{G^3}{V_{ref}}$,

$$=> \frac{V_{res3}}{G^3} = V_{in} - \frac{[G^2.d_1 + G^1.d_2 + G^0.d_3].V_{ref}}{G^3}$$

$$=> \frac{V_{res3}}{G^3} = V_{in} - \frac{[G^2.d_1 + G^1.d_2 + d_3].V_{ref}}{G^3}$$

$$=> \frac{V_{in}}{V_{ref}} = \frac{V_{res3}}{V_{ref}.G^3} + \frac{d_1}{G} + \frac{d_2}{G^2} + \frac{d_3}{G^3} \tag{1.5}$$

It can be generalized for *N* number of stages as follows,

$$\frac{V_{in}}{V_{ref}} = \frac{V_{resN}}{V_{ref}.G^N} + \sum_{k=1}^{N} \frac{d_k}{G^k} \tag{1.6}$$

Here, term $\sum_{k=1}^{N} \frac{d_k}{G^k}$ represents output code of ADC. And the term $\frac{V_{resN}}{V_{ref}.G^N}$ represents quantization error.

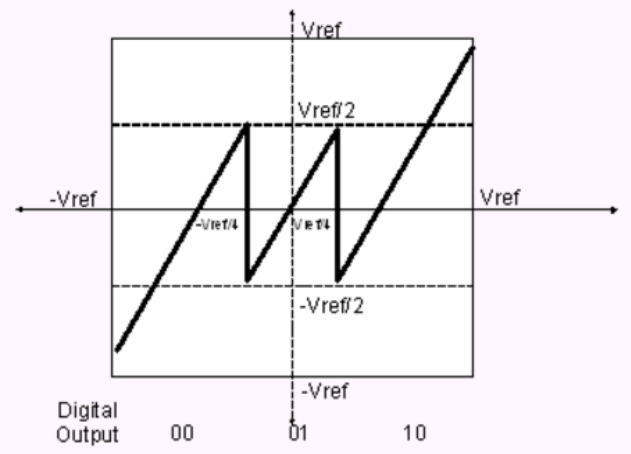

Figure 1.4: Input (x-axis) *vs* residue (y-axis) plot for first pipelined stage [32]

Figure 1.4 shows transfer function of one pipelined stage. Since maximum amplitude of residue voltage calculated is always less than maximum amplitude of input signal, V_{ref} of subsequent stage should be reduced so that accuracy is retained. But it would make the pipelined stages non-identical to each other. However, by amplifying the

residue, we can eliminate the need to scale down V_{ref} in further stages. The disadvantage is that the use of amplifying op-amp results in reduction of speed. Also, op-amp finite open loop gain results in non-idealities in ADC behaviour.

After all the stages have finished processing a sample, output bits from each stage are operated upon for redundancy removal. Last bit from last stage (2-bit flash ADC) is taken directly as LSB of total ADC output. Rest all bits are added with carry to remove redundancy as shown in figure 1.2. Final ADC output consists of N number of bits D_1 to D_N representing digital equivalent of input sample value. For example, consider a 4-bit, 1.5 bit per stage pipelined ADC. Let the output codes from the three stages of pipelined ADC be 10, 01, 10. Figure 1.5 represents addition of bits to obtain final 4-bit ADC output.

```
  1   0
      0   1
          1   0
-------------------
  1   1   0   0
-------------------
```

Figure 1.5: Redundancy removal in 4-bit, 1.5 bit per stage pipelined ADC

1.2 Data Latency in pipelined ADC

Before sending the output bits from each stage to redundancy removal, each sample must propagate through all the pipelined stages.

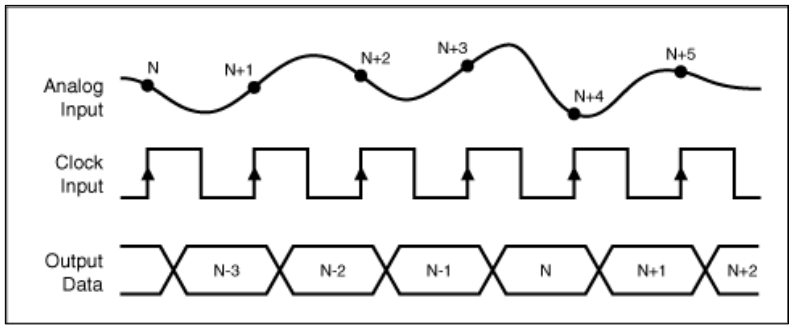

Figure 1.6: Data Latency in pipelined ADC [33]

For example in a 4-bit, 3-stage pipelined ADC, each sample should propagate through all the three stages. Refer figure 1.6. If one stage completes its work in one clock cycle, then it should take three clock cycles (one clock cycle each stage) to completely convert amplitude of N^{th} sample into its digital equivalent. Pipelined ADC is used in applications where some latency is tolerable. However, due to pipelining of analog to digital conversion process, after some latency each clock cycle provides digital equivalent of a sample. Thus, pipelined ADC offers fast speed, and greater throughput than non-pipelined stages.

1.3 Common Sources of Errors in Pipelined ADC

Errors in pipelined ADCs may be caused by noisy components, inadequate supply bypassing, clock jitter, and input to output coupling. Digital circuits introduce a lot of noise on power lines. If power source employed for analog components in a mixed signal device is the same as that employed for digital components, this noise can couple into analog device components through their supply pins [34]. Also, because of output capacitance, some noise may be introduced. In addition, noisy amplifiers, high value resisters and input signal conditioning circuitry, if present, adds to noise.

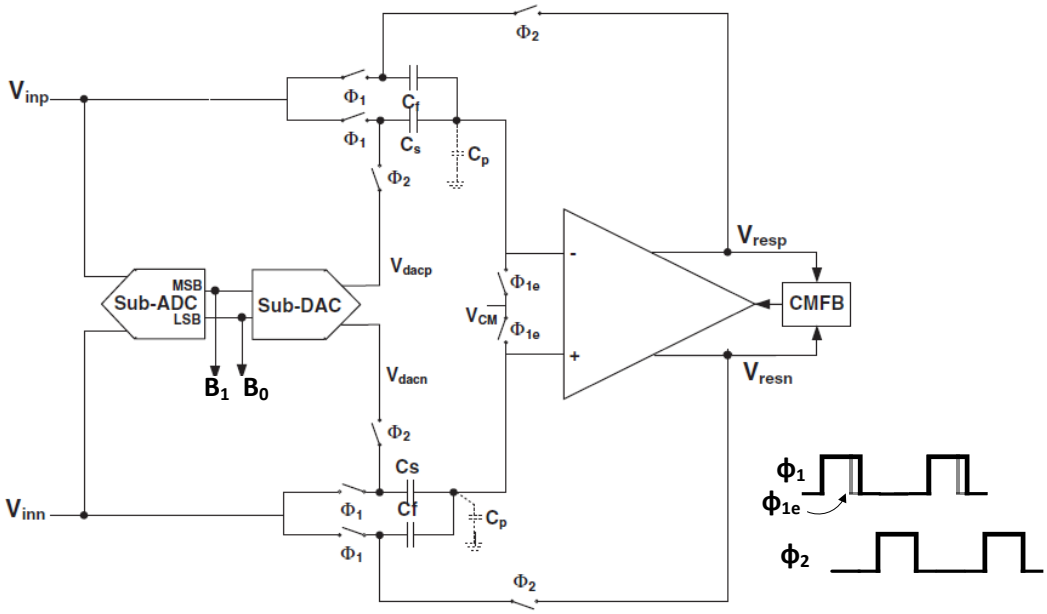

Figure 1.7: Fully differential op-amp based 1.5 bit pipelined stage [30]

A clock signal may practically have cycle to cycle variation in duty cycle. This is called jitter *i.e.* time variation of zero crossing of the signal. It causes uncertainty in sampling time, which in turn, reduces dynamic performance. Causes of clock jitter include imperfect clock source, poor grounding, and coupling of energy into clock line from signal sources.

Non-idealities in pipelined ADC due to any of these factors result in erroneous calculation of residue voltage from a pipelined stage. Figure 1.7 shows a fully differential op-amp based 1.5 bit pipelined stage. Taking different errors into consideration, equation for residue calculation can be written as [30],

$$V_{resi} \approx \left(1 - \frac{1}{A\beta}\right)(1 - e^{-t.UGB})[(G + \Delta)V_{ini} - (1 + \Delta)V_{daci}] \tag{1.7}$$

Here, A is open loop gain of op-amp, β is feedback factor given as $C_f/(C_s + C_f + C_p)$, C_f is feedback capacitance, C_s is sampling capacitance, C_p is parasitic capacitance, *UGB* is unity gain bandwidth, Δ represents capacitor mismatch error given as $(C_s - C_f)/C_f$. V_{ini} and V_{resi} represent input voltage and output residue voltage for i^{th} pipelined stage with $V_{in} = V_{inp} - V_{inn}$ and $V_{res} = V_{resp} - V_{resn}$. Φ_1 and Φ_2 are non-overlapping clocks. Φ_1 is clock for sampling phase and Φ_2 is for amplification phase. Φ_{1e} is the early falling edge version of clock Φ_1. Ideally, unity gain bandwidth should be large so that $(1 - e^{-t.UGB}) \rightarrow 1$. Also, capacitors C_s and C_f should be perfectly matched so that $\Delta = 0$. If C_s and C_f are not perfectly matched, it causes deviation of stage gain from ideal value. Parasitic capacitance C_p also adds to gain error. Ideally, open loop gain of op-amp (A) should be large so that $1/A\beta \xrightarrow{yields} 0$ and hence, $(1 - 1/A\beta) \xrightarrow{yields} 1$. Due to finite open loop gain, stage gain becomes non-ideal. It is evident that there are many factors that contribute to non-ideal behaviour of a pipelined stage. Issues are mainly caused by inter stage gain error, gain stage offset, sub-DAC error, capacitor mismatch, switch charge injection mismatch. There are many methods proposed by researchers to calibrate pipelined ADC for one or more of these errors.

1.4 Pipelined ADC Performance Metrics

Figure 1.8 shows the transfer characteristics of an ideal 3-bit ADC with high level reference voltage V_{ref} = 8 V. Practically, on conversion from analog to digital a certain level of inaccuracy is introduced and the input-output characteristic stair-case is not so

perfect. First, converting continuous time signal to discrete time signal causes loss of information at the times lying between two consecutive sample instants. Second, the samples chosen also do not get converted to digital equivalent accurately. In order to improve accuracy, resolution needs to be improved by using more number of bits and hence smaller step size. To get smaller steps using same number of bits, we can lower the reference voltage. But small reference voltage means that it can process only small range of input voltages *i.e.* lower input dynamic range. This can also cause a small signal to be lost in noise thus reducing signal to noise ratio.

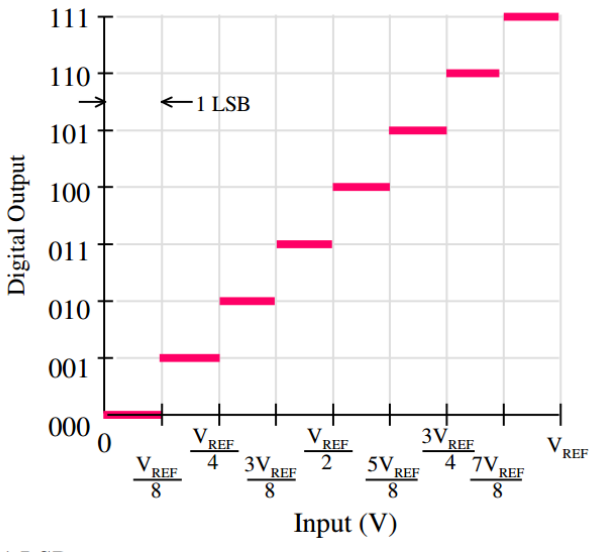

Figure 1.8: Transfer function of an ideal 3-bit ADC [34]

It is important to analyze the performance of an ADC before use. It can be done using some performance metrics. Performance metrics for any type of ADC including pipelined ADC can be categorized into two groups *viz.* static and dynamic parameters.

1.4.1 Static Performance Metrics

Static metrics are measured with respect to low frequency signals, ramp signals or even DC signals. These parameters include gain, offset, differential non-linearity (DNL), integral non-linearity (INL). Static metrics can be determined from ADC input-output characteristics as follows:

Quantization error: Referring to characteristics in figure 1.8, ADC input of 0 V produces output code of 0 0 0. As voltage increases towards $V_{ref}/8$, output code still

remains 0 0 0. Thus, as input approaches towards $V_{ref}/8$, error increases as the input is no longer 0 V. Again, at $V_{ref}/8$ V, code changes to 0 0 1 *i.e.* output accurately represents input and hence error is reduced to zero. Plot for quantization error is shown in figure 1.9.

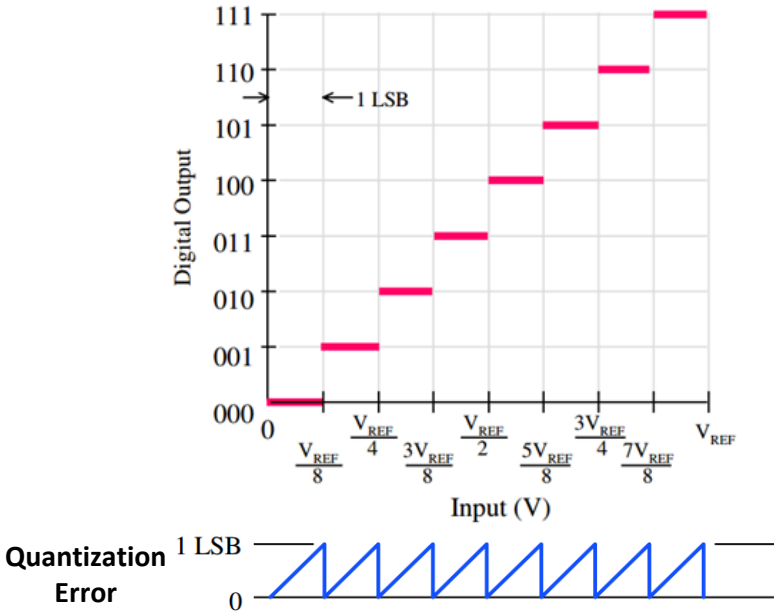

Figure 1.9: Quantization error in ideal 3-bit ADC [34]

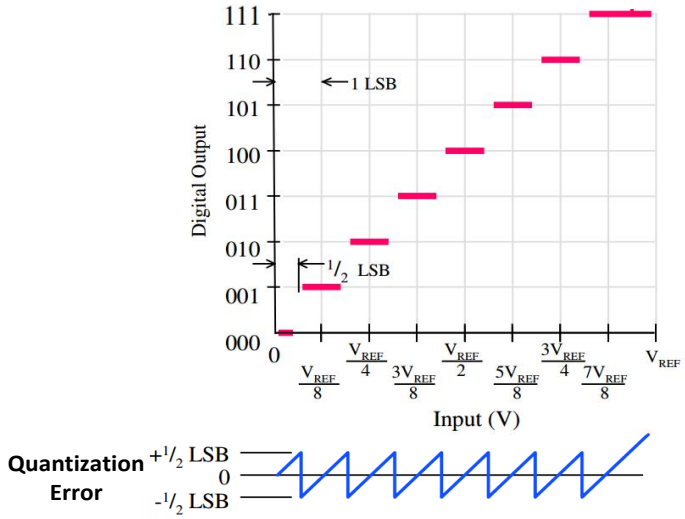

Figure 1.10: Quantization error in 3-bit ADC with added half LSB offset to input [34]

Magnitude of error ranges from 0 to 1 LSB, and this range is called quantization uncertainty. Maximum quantization uncertainty is called quantization error. Cause of quantization error is finite resolution of ADC. An *n*-bit ADC can resolve the input into

2^n discrete levels only. Each output code represents a range of input values called 'quanta' (q) [34].

By adding half LSB offset to the ADC input, the output digital level would change half LSB before it would have changed without offset. Thus, quantization error ranges from -0.5 LSB to +0.5 LSB instead of 0 to 1 LSB as shown in figure 1.10.

Offset error: Ideally, an input voltage of quanta/2 *i.e.* q/2 should cause transition from zero to next digital code. Difference between this ideal input voltage and practical input voltage where first output digital code transition takes place is known as zero scale offset error, shown in figure 1.11.

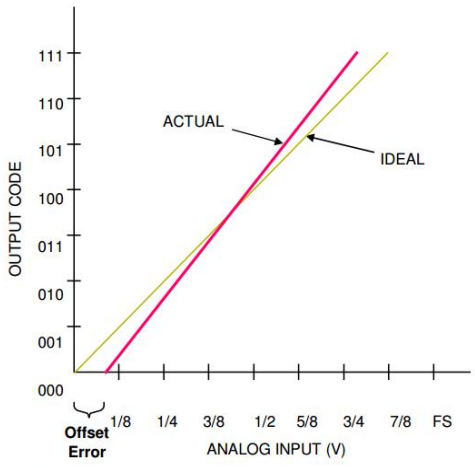

Figure 1.11: Zero scale offset error in ADC transfer function [34]

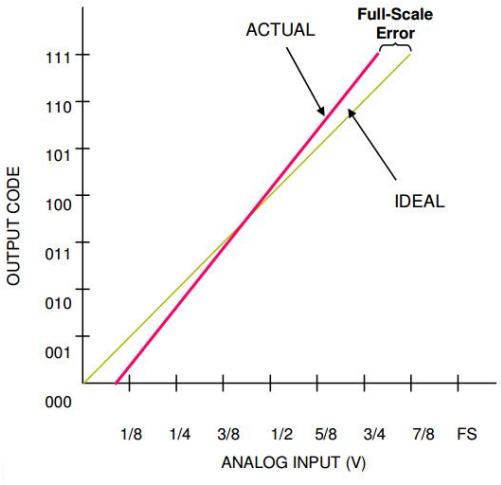

Figure 1.12: Full scale offset error in ADC transfer function [34]

If first transition input voltage is higher than the ideal, offset error is said to be positive. If first transition input voltage is lower than the ideal, offset error is said to be negative. Error in real full scale output transition point from the ideal value is called full scale offset error, shown in figure 1.12. Full scale offset error is caused partly due to offset error and partly due to error in slope of input-output characteristic. Since offset error is constant, it can easily be eliminated.

Offset error is measured in percent of full scale voltage (Volts) or in terms of LSBs.

Gain error: Full scale gain error is defined as difference between ideal and practical slopes of transfer function.

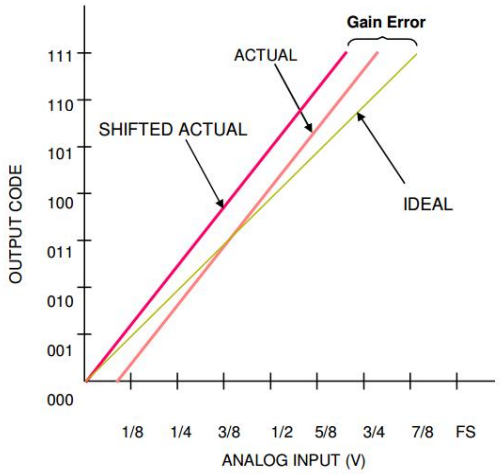

Figure 1.13: Gain offset error in ADC transfer function [34]

Thus, if actual transfer function is shifted in a way such that zero scale offset error gets nullified, then the difference between actual and ideal last transitions in ADC transfer stair case is called gain error, as demonstrated in figure 1.13.

$$gain\ error\ =\ full\ scale\ error - offset\ error$$

It is measured in terms of LSBs or as percentage of ideal full scale voltage.

Non-Linearity: There are the following two kinds of non-linearity errors measured to quantify the effect of errors in ADCs.

(i) Differential non-linearity or differential linearity error

(ii) Integral non-linearity or integral linearity error

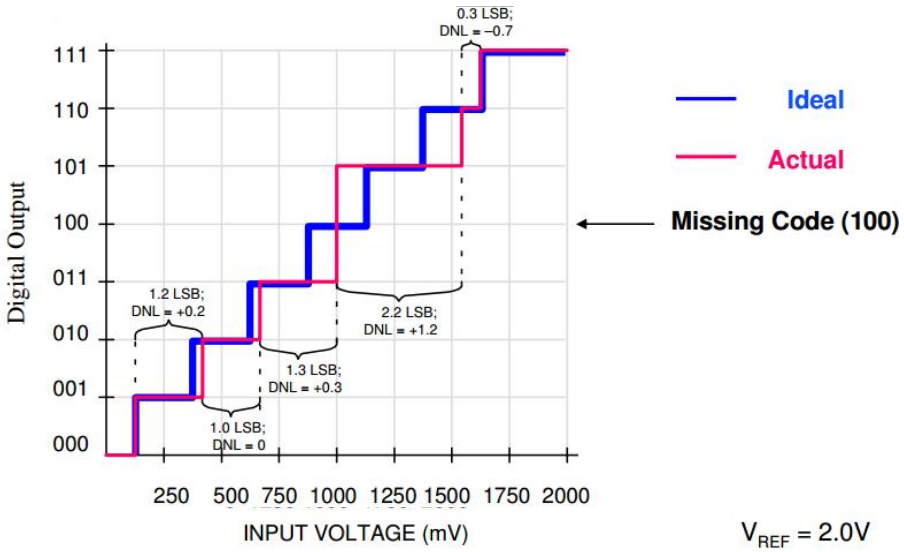

Figure 1.14: Differential non-linearity error and missing code [34]

<u>Differential non-linearity or differential linearity error</u>: Differential non-linearity describes the error in step size. In ideal converter, output digital code transitions occur exactly after each LSB. Thus, step size of transfer characteristic stair case shown in figure 1.14 should be 1 LSB. The difference between ideal step size and maximum practical step size occurring in transfer function is called differential non-linearity. Referring figure 1.14, digital output level does not change even if input changes from 1000 mV to above 1500 mV. Code 1 0 0 never appears at output. This is called missing code.

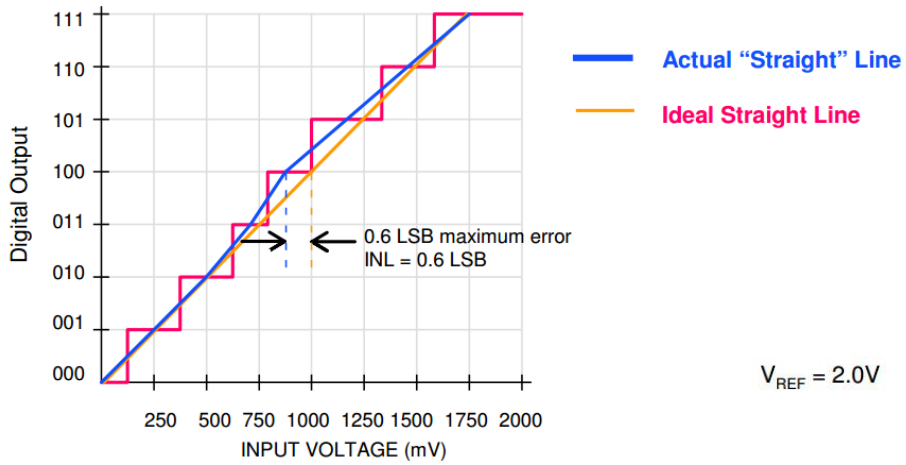

Figure 1.15: Integral non-linearity error [34]

<u>Integral non-linearity or integral linearity error</u>: Integral non-linearity describes the bow in transfer function *i.e.* describes deviation from ideal linear transfer curve

as shown in figure 1.15. Integral linearity error is the measure of straightness of transfer function. Quantization, offset and gain errors are not included in INL. The size of differential non-linearity determines the integral linearity of the ADC.

Total unadjusted error: It is a comprehensive specification that includes linearity errors, gain error, offset errors. It is worst case deviation from ideal device performance [34].

1.4.2 Dynamic Performance Metrics

Analysis of dynamic metrics of ADC is based on frequency response of ADC. Therefore, dynamic parameters are determined using high frequency signals. A high frequency signal is fed at input of the ADC which provides quantized version of the same signal after reconversion of ADC digital output. Frequency spectrum of this reconstructed high-frequency signal obtained by taking Fast Fourier Transform (FFT) is analysed to determine dynamic parameters of the ADC under test. These dynamic parameters are explained below.

Signal to noise ratio (SNR)

Ratio of signal power to noise power is called SNR. For an ADC of resolution *N*, maximum SNR (dB) that can be achieved is given as,

$$SNR_{max}(dB) = 6.02 \times N + 1.76$$

Apart from linearity errors, other factors mentioned in section 1.1.4 like clock jitter, thermal noise also contribute to degradation of SNR. Thus, SNR is the ratio of signal power to the sum of different noise powers. Since jitter increases due to increase in input signal frequency, and signal power decreases with decrease in input signal amplitude, thus SNR can be said to be input signal dependent. A high SNR indicates good ADC performance.

Total harmonic distortion (THD)

Spectrum of ADC output obtained from a non-ideal ADC contains harmonics at the frequencies which are multiples of signal frequency. Total harmonic distortion is defined as ratio of sum of power of all the harmonics to the signal power.

$$THD = \frac{\sum_{h=2}^{m} P(h)}{P(s)}$$

where *m* is the total number of harmonics present in spectrum, *P(h)* represents power of h^{th} harmonic, *P(s)* represents signal power or power of fundamental harmonic.

Spurious free dynamic range (SFDR)

All the tones occurring at integer multiples of input frequency are called harmonics. Frequencies at non-integer multiples of input frequencies, where unwanted tones occur are called spurious frequencies. SFDR is the ratio of signal power to any spectral component except the DC component. The spectral component in denominator can be a tone occurring at harmonic frequency or a tone occurring at spurious frequency.

$$SFDR = \frac{P(s)}{\max(P(spectrum))}$$

Here *P(s)* represents signal power, *P(spectrum)* represents power of other spectral components occurring at integer or non-integer multiples of signal frequency.

Signal to noise and distortion ratio (SNDR or SINAD)

SNDR is a complete dynamic metric that covers effect of noise as well as distortion. SNDR is defined as ratio of signal power to sum of powers of all the harmonics (as in numerator of THD) and sum of power of noise components (as in denominator of SNR).

$$SNDR = \frac{P(s)}{\sum_{h=2}^{m} P(h) + P(n)}$$

Here *P(s)* represents signal power, *P(h)* represents power of h^{th} harmonic, *P(n)* represents noise power which further is sum of jitter, DNL, thermal noise powers.

Effective number of bits (ENOB)

Due to non-idealities, there can be difference in ideal resolution of ADC and real resolution of ADC. According to the SNDR obtained for ADC under test, real resolution of ADC can be determined using below formula. This is called effective number of bits.

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

1.5 Calibration Methods for Pipelined ADC

Calibration means to improve accuracy and other performance parameters of an ADC in only a tolerable time and power consumption. Commonly achieved resolution with pipelined ADCs is within the range of 10-16 effective number of bits. But due to errors

discussed in previous section, effective number of bits reduces intolerably. High effective resolution can be achieved using extra calibration logic attached to the ADC. Calibration techniques may be employed in analog domain or in digital domain.

Calibration in analog domain: Analog calibration requires modification in design of analog components used in ADC in order to achieve desired parameters. Many research works are based on modification of op-amp design to achieve open loop gain and unity gain bandwidth as high as possible. However, these parameters are improved at the cost of increased power consumption and circuit complexity. Improving a particular parameter of an analog component results in some other parameter to be put to stake.

Calibration in digital domain: Digital calibration does not require modification in design of analog components. In such techniques, ADC response is obtained for a particular input stimulus. ADC response is then compared to the expected response (ideal response) so as to generate error signal. This error signal is then used for radix extraction digitally.

Apart from above categories, calibration methods may be categorized into foreground and background methods.

Foreground calibration: In foreground calibration technique, ADC should be idle while calibration is carried out. A pre-known test signal is injected at input to the ADC. Error is then measured by comparing the real output code with ideally expected output. As during calibration, the device is idle, foreground calibration cannot track device and environmental variations. However, this technique is generally faster than background calibration techniques.

Background calibration: In background calibration techniques, circuit operation continues while calibration is carried out. In such methods, normal ADC input serves as calibration input when calibration has to take place. This method has an advantage over foreground calibration method as it does not have to wait for the circuit to be idle to carry out calibration. However, background calibration methods are generally slower than foreground methods due to additional delay of a slow but ideal reference

ADC used to generate expected (ideal) ADC output. Background calibration proves to be better when it comes to tracking device and environmental variations.

Sometimes it is necessary to use a foreground calibration technique at start-up of the device. In such a case, foreground method can be combined with background calibration method to get more accurate output from ADC.

Chapter 2

LITERATURE REVIEW

This section highlights the literature review done on pipelined ADCs and techniques for their calibration.

Andrew N. Karanicolas *et al.* [1] described a digital self-calibration technique based on radix lesser than 2 applied to a 1.5 bit per stage pipelined ADC. This technique corrects errors due to capacitor mismatch, finite open loop gain of op-amp and circuit non linearity. The technique described requires only addition, subtraction, and small data storage.

Un-Ku Moon and Bang-Sup Song [2] explained the skip and fill algorithm developed to digitally self-calibrate the pipelined ADC in background. This technique uses the concept of skipping conversion cycles randomly and then filling the data later by nonlinear interpolation. The missing data is filled before the digital values are provided. In order to measure errors resulting from capacitor mismatch, op-amp dc gain, offset, and switch feed through in real time, the calibration test signal is injected in place of the input signal making use of split-reference injection technique. The nonlinear interpolation technique used was derived for an arbitrary $(2n-1)^{th}$ order polynomial. Missing signal within two third of Nyquist bandwidth is restored with 16-bit accuracy using interpolation.

Boris Murmann and Bernhard E. Boser [3] described a digital calibration technique that allows replacement of precision amplifiers by simple power-efficient open-loop stages. In first stage of 12-bit 75 MS/sec, more than 60% residue amplifier power is saved as compared to the conventional implementation.

Jipeng Li and Un-Ku Moon [4] presented a digital calibration method for correction of linearity errors in 1.5 bit-per-stage pipelined ADC resulting from capacitor mismatch and finite op-amp gain. The method proposed requires only a minimal addition of analog hardware. In the proposed scheme, equivalent radices for each stage are extracted by applying a correlation based algorithm using pseudorandom noise sequence called dither.

It is not required to reduce the input dynamic range in order to inject pseudorandom calibration signal. The extracted equivalent radix is used to re-calculate the digital output of each stage. Radix extraction is done in background in order to allow uninterrupted working of ADC. Simulation results showed that the signal to noise and distortion ratio (SNDR) before calibration was 69 dB. After the proposed background calibration was applied, the SNDR improved to 102 dB.

Yun Chiu *et al.* [5] presented an adaptive digital technique to calibrate pipelined ADCs. Digital post-processing is used to correct non linearity. With the help of a slow but accurate ADC, adaptive finite impulse response filter is used to remove the effect of errors like capacitor mismatch, finite op-amp gain, op-amp offset error, and sampling-switch-induced offset, if they are signal independent. Slow ADC samples the input signal at a lower sample rate. Difference between corresponding samples of two ADCs is used to correct fast ADC digital output using an adaptive Least-Mean-Square algorithm working in background.

John P. Keane *et al.* [6] designed a background self-calibration technique to correct both linear and nonlinear errors in the inter-stage amplifiers of pipelined ADCs. Stage redundancy in pipelined ADCs is exploited in this scheme in order to measure gain errors which are then corrected using digital processing. Simulation of proposed method was done using 12-bit ADC considering non-ideal inter-stage residue amplifiers. It was proved that tracking and convergence rates became ten times faster after using the described calibration technique. SNDR of 72 dB and SFDR of 112 dB were achieved.

Carl R. Grace *et al.* [7] presented a calibration algorithm to adaptively correct gain error, closed-loop gain variation, and slew-rate limiting. Along with every stage of the pipelined ADC, the input sample and hold is also calibrated. In suggested scheme, first two stages undergo independent correction for constant as well as signal dependent gain errors. Stages 3-6 undergo independent correction for constant gain errors. Rest of the stages undergo a common correction process for constant gain error. Calibration algorithm is carried out by an embedded custom microprocessor. The DAC is used for calibration of the analog to digital converter, and the analog to digital converter is used to calibrate the digital to analog converter. Test results proved that maximum DNL achieved was -0.09 LSB, and maximum INL achieved was - 0.24 LSB. Also, maximum signal to noise and

distortion ratio and spurious free dynamic range achieved were 71.0 and 79.6 dB respectively, with a 40 MHz sinusoidal input.

Cheongyuen Tsang *et al.* [8] explained the Least Mean Square (LMS) algorithm using a 100MS/sec pipelined ADC. In this scheme, the high bandwidth ADC is digitally calibrated by a slow sigma-delta ADC using LMS algorithm. It corrects the linear as well as nonlinear memory-less residue gain errors of pipelined stages. The described method when used with a 411 kHz sinusoidal input, improved the peak SNDR from 28 dB to 59 dB and the spurious-free dynamic range (SFDR) from 29 dB to 68 dB.

Imran Ahmed and David A. Johns [9] presented a technique to calibrate the multi-bit first stage of an 11-bit pipelined ADC using dual-ADC based approach. The scheme corrects both DAC and gain errors in first stage. Simulation results showed an improvement in peak INL of 45 MS/s ADC from 6.4 LSB to 1.1 LSB after calibration. SNDR/SFDR was improved from 46.9 dB/48.9 dB to 60.1 dB/70 dB after calibration.

Lane Brooks and Hae-Seung Lee [10] described a background calibration method called Decision Boundary Gap Estimation (DBGE) that monitors the size of code gaps that result at decision boundaries of each stage. Code gaps result from effects like capacitor mismatch, finite op-amp gain, finite current source output impedance, comparator offset and charge injection. No additional hardware is required to implement this calibration scheme. The calibration is performed using input signal and it is required that input signal exercises the codes in the vicinity of the decision boundaries of each stage. If it does not exercise such codes, then lack of calibration is less critical because the nonlinearities will not appear in the output signal. Simulation results indicated that DBGE is highly accurate, robust, and adaptive even in the presence of parameter drift and circuit noise.

Alma Delic-Ibukic and Donald M. Hummels [11] described a background calibration technique for 14-bit pipeline ADC with 1-bit/stage pipeline architecture. Static errors in pipeline ADCs were discussed and corrected by the implemented digital calibration algorithm using two extra stages for calibration only. Verilog implementations of pipelined stage and error correction logic were simulated using Verilog-XL simulator.

Sun Kexu and He Lenian [12] designed a technique for correction of nonlinearity caused by finite op-amp gain and capacitor mismatch in multiplying analog to digital converters. The calibration technique proposed is a combination of foreground and background calibration schemes. An example of 14 bit 100 Msample/sec pipelined ADC is taken to describe the technique. The calibration parameter is extracted immediately at the start up using foreground calibration and then it is continuously updated at background in order to adapt to device and environment changes. Another technique of parallel background calibration is proposed with signal shifted correlation. This technique allows injection of large dithering when the stage input signal is in range $-Vref$ to $+Vref$. The proposed method reduces the background calibration tracking cycle.

Sudipta Sarkar *et al.* [13] presented a digital background calibration technique to treat capacitor mismatch, gain error, and nonlinearity in a pipelined ADC based on split-ADC architecture. Behavioural simulation results demonstrated that SNDR and SFDR performance of a 15-bit split-pipelined ADC were improved from 42 dB and 50 dB to 88 dB and 102 dB on average, respectively.

John A. McNeill *et al.* [14] presented split ADC architecture in which die area is split into two halves. Each half converts the same analog input to digital. Final result is obtained by averaging the results from two independent halves. The difference of the outputs from two halves can be used as information required by background calibration algorithm.

Meysam Mohammadi *et al.* [15] presented a background calibration technique using slow but high accuracy ADC. The technique takes care of errors due to nonlinear and finite inter-stage gain. LMS algorithm is used to extract error correction coefficients. It does not cause much design complexity.

J. Yuan *et al.* [16] introduced digital calibration technique using interpolation-based methodology. The presented digital calibration technique improved the ADC differential nonlinearity and integral nonlinearity from 1.47 LSB and 7.85 LSB to 0.2 LSB and 0.27 LSB for 12-bit resolution.

H. Mafi and A. Sodagar [17] suggested a method to continuously measure residue amplifier nonlinearity error and digitally compensate it. This scheme is beneficial in

decreasing power consumption and increasing sampling rate in pipelined ADC. A fifth-order polynomial is used to eradicate conversion errors. In this method, a single pseudorandom sequence is used for error correction. Simulation results proved that signal-to-noise-and-distortion-ratio (SNDR) was improved from 40 to 66 dB and spurious-free-dynamic-range (SFDR) was increased from 48 to 80 dB.

D. Meganathan *et al.* [18] proposed an approach for low-power 10-bit, 100 MS/s pipelined analog-to-digital converter (ADC). A modified two stage high gain operational trans-conductance amplifier having wide bandwidth is used in track-and-hold amplifier and multiplying digital to analog converter sections in order to reduce power consumption and thermal noise. The signal swing of the analog functional blocks is allowed to exceed the supply voltage (1.8 V). It increases the dynamic range of the circuit. Dynamic power dissipation and kickback noise of comparator is reduced using charge sharing comparator. Bottom plate sampling technique and bootstrap technique is employed to reduce nonlinearity error that causes signal-to-noise-distortion ratio of 58.72 at 2 MHz and 57.57 dB at Nyquist frequency. Maximum differential nonlinearity (DNL) of +0.6167/−0.351 LSB and maximum integral nonlinearity (INL) of +0.4271/−0.4712 LSB was achieved. Dynamic range of the ADC came out to be 58.72 dB for full-scale input signal.

Anil Singh and Alpana Agarwal [19] presented a digital background calibration scheme for charge pump based pipelined ADC. A 10-bit 100 MS/s pipelined ADC is designed using TSMC 0.18 µm CMOS technology operating on 1.8 V power supply. An op-amp-less power efficient charge pump based technique is used to achieve desired stage voltage gain of 2. Inter stage gain error is calibrated using digital background calibration method. ADC achieved an SNDR of 66.78 dB and SFDR of 79.3 dB after calibration. DNL improved to +0.6/–0.4 LSB. INL improved to within ±0.5 LSB.

A. Larsson and S. Sonkusale [20] suggested a background non-invasive true calibration technique to correct pipelined ADC non-idealities. The digital calibration scheme discussed in this work uses a low-speed, low-power, high resolution Sigma-Delta ADC to estimate digital error correction parameters using adaptive LMS algorithm. The technique corrects static errors within a single framework - finite amplifier gain, capacitor mismatch, voltage reference errors and amplifier non-linearity.

S. Roy *et al.* [21] discussed a foreground calibration technique for pipelined ADC. The work emphasizes on correction of error due to capacitor mismatch. Capacitor mismatch contributes majorly to ADC nonlinearity. The proposed technique makes use of an arithmetic unit to calculate radix from ADC output for calibration. To verify the calibration algorithm, pipeline ADCs of distinct resolutions were designed and simulated in 0.18 μm CMOS process.

Z. Liren *et al.* [22] presented a digital background calibration technique for a 12-bit 100 MS/s CMOS pipelined analog-to-digital converter (ADC). Keeping the architecture of multiplying digital-to-analog converter (MDAC) unchanged, a calibration signal of large magnitude is injected into it. It takes 2.8 s to calibrate the 12-bit prototype ADC with sampling rate of 100 MS/s and a peak spurious-free dynamic range of 85 dB and a peak signal-to-noise plus distortion ratio of 66 dB was achieved with input signal of frequency 2 MHz. Integral nonlinearity improved from 1.9 to 0.6 LSB after calibration. The chip was fabricated in a 0.18 μm CMOS process that consumes 205 mW at 1.8 V and occupies an active area of 2.3×1.6 mm^2.

M. Abdelhamid *et al.* [23] introduced a 12-bit pipeline Analog to Digital Converter (ADC) using 1.2V and 0.13μm CMOS technology. Embedded sample and hold technique is used in order to avoid the dedicated power hungry sample and hold circuit. Op-amp's finite open loop gain and non-linearity are taken care of using low gain op-amps, and a foreground digital calibration scheme. The ADC consumes 65 mW of power and reaches a maximum SNDR of 68.5 dB with an SFDR up to 80 dB.

Xiaoyue Wang *et al.* [24] described a background calibration method for a 12-bit, 20-Msamples/s pipelined analog-to-digital converter (ADC), using an algorithmic ADC. The algorithmic ADC is calibrated using foreground calibration scheme. Capacitor mismatch error and finite op-amp gain error are corrected in both - the pipelined ADC and the algorithmic ADC. With a 58-kHz sinusoidal input, a peak signal-to-noise-and-distortion ratio (SNDR) of 70.8 dB, a peak spurious-free dynamic range (SFDR) of 93.3 dB, a total harmonic distortion (THD) of 92.9 dB, and a peak integral nonlinearity (INL) of 0.47 least significant bit (LSB) was achieved.

I. Ahmed *et al.* [25] presented a low-power pipelined ADC that used capacitive charge pumps and source-followers. A digital calibration method was also discussed to avoid power-hungry op-amps and achieve better linearity in a pipelined ADC. The differential charge pump technique achieves 10-bit linearity without an explicit common-mode-feedback circuit. According to measured results SNDR and SFDR of the ADC approached 58.2 dB (9.4 ENOB), and 66 dB respectively. The ADC consumed 3.9 mW power for active circuitry and 6 mW for clocking and digital circuits.

Li Zhang [26] reviewed the history of calibration techniques. The paper explains typical calibration approaches *i.e.* analog and digital calibration techniques. A detailed theory is presented on foreground and background calibration techniques. The paper suggests that because of circuit complexity and power penalty, designers pay more attention to digital calibration methods. It also highlights that although foreground calibration has some disadvantages compared to background calibration, it can be applied in some special systems that allow intermittent operation.

Esmaeil Fatemi-Behbahani *et al.* [27] presented a digital background calibration scheme for redundancy based multi bit pipelined ADC. Error voltage is calculated first and then residue of each stage is calculated as sum of the ideal and error signal. In calibration process, non-linear effect in digital output from backend ADC is eliminated first. Then the effect of this nonlinearity is eliminated from total ADC output by digital using digital adaptive filter. Simulation results have shown that INL and DNL of 14-bit radix-4 pipelined ADC improve to 0.45 LSB/0.41 LSB after calibration. SNDR and SFDR increase from 30 dB/36 dB to 83 dB/90 dB respectively.

Bing-Nan Fang and Jieh-Tsorng Wu [28] fabricated a 10-bit pipelined ADC using 65 nm CMOS technology. Switching op-amps with short turn on time were used to reduce consumption of power. Digital background calibration scheme was used to correct gain error. A bias scheme was used in order to maintain the settling behaviour of the op-amp against temperature variations. Simulation results proved that DNL and INL improve to 0.52/ 0.4 LSB and 0.99/ 1.65 LSB respectively. SNDR and SFDR improve to 55.4 dB and 67.2 dB respectively.

Li-Han Hung and Tai-Cheng Lee [29] proposed a digital background calibration technique for correction of gain error in pipelined ADC. The error is estimated and the concept of split ADC is used as a base for an adaptive error correction scheme. Capacitor mismatch and gain error can be combined and corrected with the suggested calibration scheme for 1 or 1.5 bit pipelined stage. Simulation results proved that SNDR improved from 56.4 to 73.8 dB. Calibration logic was tested with a 12-bit pipelined ADC with gain accuracy of 8 bits and capacitor mismatch of 0.125%. The calibration method reaches convergence in about 200 000 iterations.

Chapter 3

PROBLEM STATEMENT

3.1 Problem Statement

Gain error occurs due to finite open loop gain of op-amp and causes missing codes in ADC transfer function *i.e.* non-linearity is introduced. Extent of non-linearity is measured in terms of DNL and INL which increase with increase in gain error. This work aims on the problem of gain error correction in a 1.5 bit per stage pipelined ADC, using a digital calibration technique.

3.2 Problem Explanation

Recall equation (1.7) obtained from figure 1.15. As mention earlier in section 1.3, ideally unity gain bandwidth should be large such that $(1 - e^{-t.UGB}) \rightarrow 1$. Thus equation (1.7) becomes,

$$V_{resi} \approx \left(1 - \frac{1}{A\beta}\right)(1 - e^{-t.UGB})[(G + \Delta)V_{ini} - (1 + \Delta)V_{daci}]$$

$$= \left(1 - \tfrac{1}{A\beta}\right)[(G + \Delta)V_{ini} - (1 + \Delta)V_{daci}] \quad (1.8)$$

Now, if capacitors C_s and C_f are perfectly matched then,

$$\frac{(C_s - C_f)}{C_f} = 0$$

$$=> \quad \Delta = 0$$

Equation (1.8) then reduces to,

$$V_{resi} \approx \left(1 - \tfrac{1}{A\beta}\right)[G.V_{ini} - V_{daci}] \quad (1.9)$$

Ratio $C_f/(C_s + C_f) = 0.5$ because of matched capacitors C_s and C_f. However, due to parasitic capacitance C_p, β deviates from ideal value. Due to finite open loop gain, closed loop gain of op-amp is reduced by a factor of $\frac{1}{A\beta}$. Factor $\left(1 - \frac{1}{A\beta}\right)$ represents gain error. In this work, unity gain bandwidth is assumed to be infinitely large. Also, capacitors C_s and C_f are assumed to be perfectly matched.

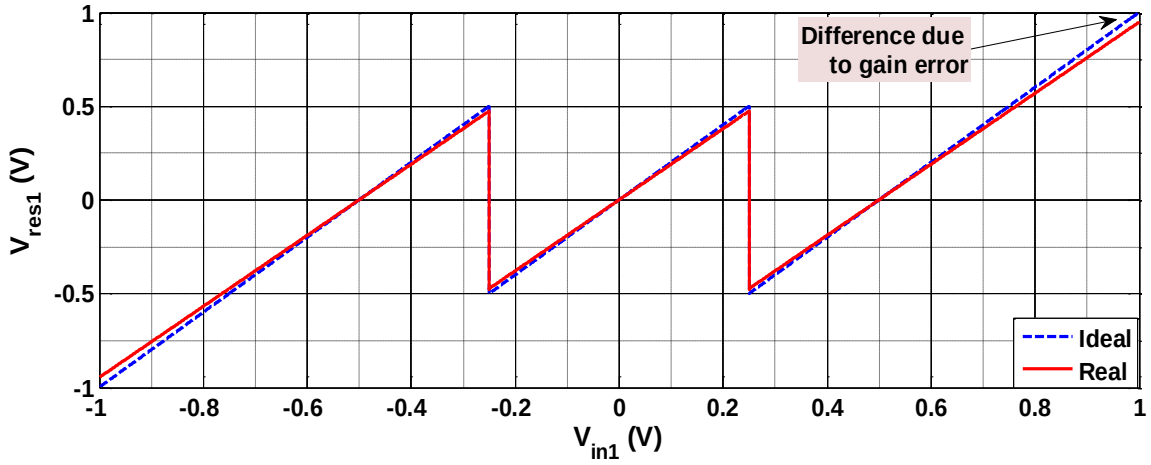

Figure 1.16: Residue plot of first pipelined stage with real op-amp of open loop gain 40 V/V

Figure 1.16 shows residue of first stage with real op-amp gain of 40 V/V. Figure 1.17 shows effect of gain error on second stage residue, when real op-amp open loop gain in both, first and second stages is 40 V/V.

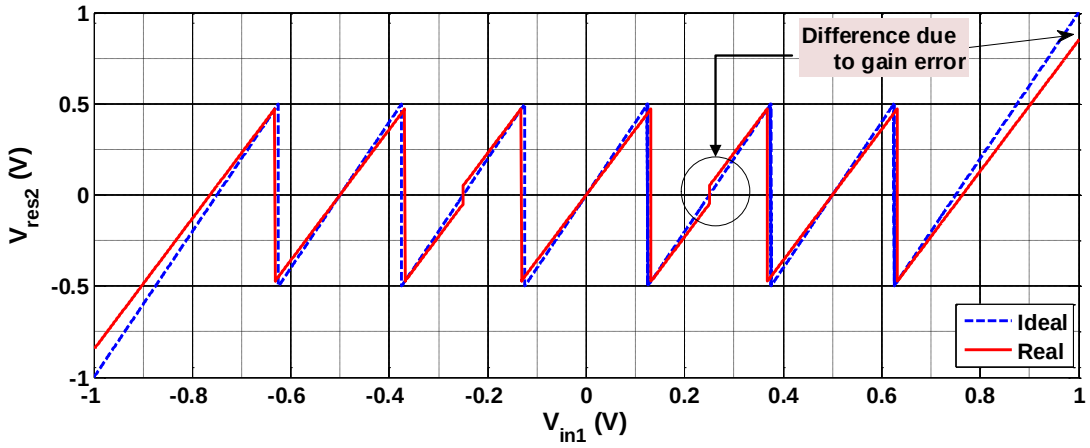

Figure 1.17: Residue plot of second pipelined stage with real op-amp of open loop gain 40 V/V in first and second stages

Due to erroneous residue calculation (equation (1.9)), missing codes occur in ADC output as shown in figure 1.18. To correct this, various analog and digital error correction techniques are used, as mentioned in chapter 2. This work focuses on the problem of gain error correction in a 12-bit, 1.5 bit per stage pipelined ADC using a digital calibration technique. A novel fast foreground method for calibration is developed as a solution, and is discussed in next section.

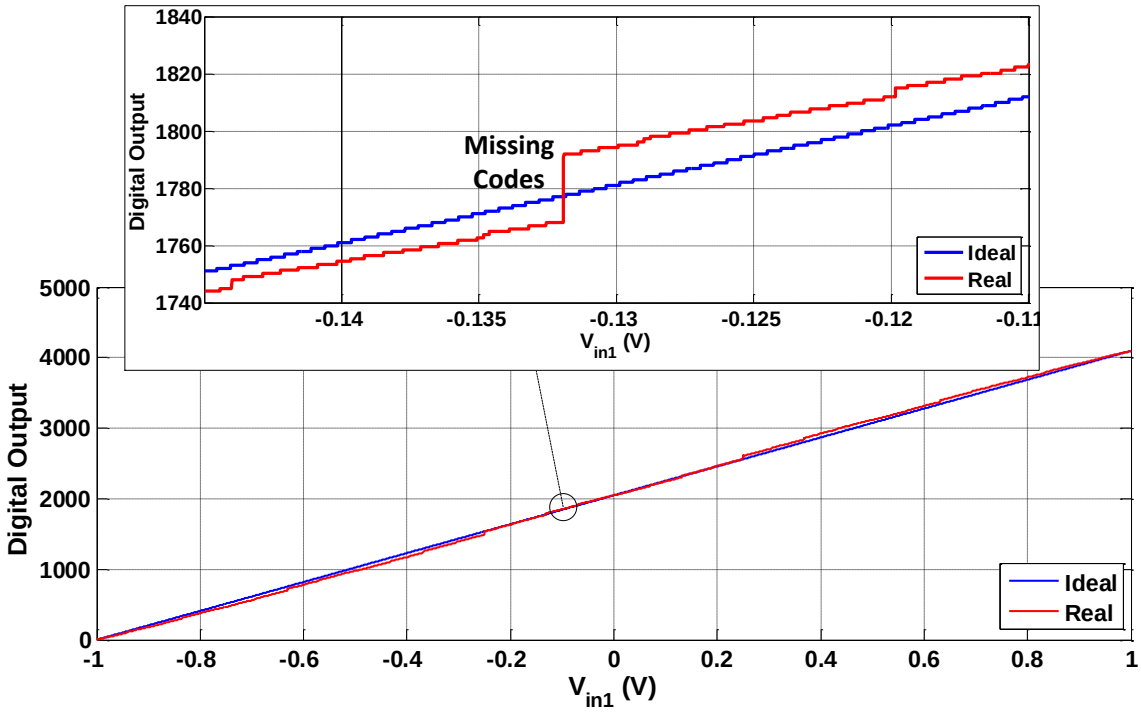

Figure 1.18: Transfer function of a 12-bit, 1.5 bit per stage pipelined ADC with missing codes due to gain error in all pipelined stages

Chapter 4

FOREGROUND CALIBRATION METHOD

4.1 Method and MATLAB Implementation

The proposed method is built over the background calibration scheme discussed in [19]. Due to error, stage gain *G* reduces from ideal value of 2, as discussed in chapter 3. To keep linearity of the pipelined ADC intact even with erroneous gain, an extra pipelined stage is used. As an example, an ideal 12 bit pipelined ADC should have 10 pipelined stages, and one 2-bit flash ADC as last stage. It has digital levels from 0 to $(2^{12} - 1)$ *i.e.* 0 to 4095. Suppose that due to error, gain reduces from 2 to ~1.90. Now to get levels from 0 to 4095, we need more number of bits. So, with gain error, an extra pipelined stage must be added to maintain linearity. It means that an ADC with 12-bit resolution with gain error must produce $N = 13$ bits at output. Hence there should be 11 pipelined stages and one 2-bit flash ADC in the end.

The last stage (2-bit flash ADC) is assumed to be ideal. Two extra pipelined stages, also assumed to be ideal, are cascaded before the 2-bit flash ADC only for calibration purpose. These three stages – two extra pipelined stages for calibration and 2-bit flash stage, together form an ideal back-end ADC. Thus, for calibration, total 15-bit (14 stage) ADC is used, out of which stages 12 and 13 are added only during calibration process, as shown in figure 1.19. For convenience, let us represent calibration ADC as N_{cali}-bit ADC. Stages 1 to 11 need to be calibrated one by one starting with stage 11. Once a stage is calibrated, it is considered to be a part of ideal back-end ADC for next iteration. Thus, while calibrating i^{th} stage, stages from *(i+1)* to last stage form ideal back-end ADC or more precisely, calibrated back-end ADC.

In this scheme, the input voltage for calibration (V_{cali}) is always given as high level ADC reference voltage ($V_{cali} = V_{refH} = +V_{ref}$) to the stage to be calibrated (say i^{th} stage). The advantage of doing so is that it eliminates the need of a slower but ideal reference ADC to generate ideal voltage value (V_{ideal}), as used in [19], [20]. It reduces hardware and improves speed. In order to implement calibration algorithm, floating point representation of V_{refH} is stored in memory as a constant. This stored value is used as V_{ideal}. Another advantage of using V_{refH} as calibration input is that maximum error is

found at V_{refH}. Calibrating the ADC against maximum error brings it closer to ideal, as compared to calibrating against lesser error.

Figure 1.19 represents calibration of i^{th} stage in N_{cali}-bit ADC. Calibration input *(V_{cali})* is given to i^{th} stage, which generates two output bits and residue voltage (*V_{resi})* according to equation (1.9). Here, $V_{ini} = V_{cali} = V_{refH}$.

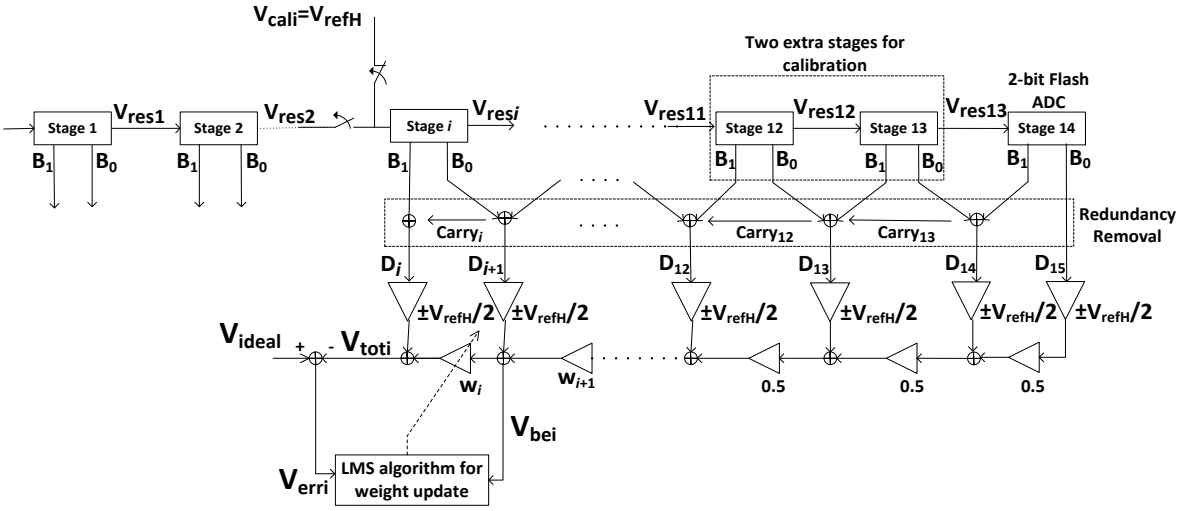

Figure 1.19. i^{th} stage calibration of N=12-bit pipelined ADC using two extra stages for calibration (N_{cali}=15)

V_{dac} can take value according to table 1.1. Residue voltage is fed as input to next stage *(i+1)*, which generates residue voltage to be used as input to stage *(i+2)*, and so on. After each stage (from stage *i* to stage 14) has generated two bits, all these bits are sent to redundancy removal block, which provides (N_{cali}-*i+1)*-bit representation of V_{cali} at output. Bits from $D_{(i+1)}$ to D_{15} are used to generate back end voltage *(V_{bei})*, according to equation (1.10).

$$V_{bei} = \left\{\left(D_{15} - \frac{1}{2}\right) \times V_{refH} \times (w_{14}.w_{13}.w_{12} \cdots w_{i-1})\right\} + \left\{\left(D_{14} - \frac{1}{2}\right) \times V_{refH} \times (w_{13}.w_{12} \cdots w_{i-1})\right\} + \left\{\left(D_{13} - \frac{1}{2}\right) \times V_{refH} \times (w_{12} \cdots w_{i-1})\right\} + \cdots + \left\{\left(D_{i+1} - \frac{1}{2}\right) \times V_{refH}\right\} \quad (1.10)$$

where, w_i represents weight to be determined and is equal to $1/G$. So ideally $w_i = 0.5$. Since two extra pipelined stages attached for calibration and 2-bit flash ADC stage are considered to be ideal, therefore,

$$w_{14} = w_{13} = w_{12} = 0.5.$$

It can be noted that,

$$\left(D_i - \frac{1}{2}\right) \times V_{refH} = \begin{cases} +\dfrac{V_{refH}}{2}, & D_i = 1 \\ -\dfrac{V_{refH}}{2}, & D_i = 0 \end{cases}$$

Therefore this term is represented as $\pm V_{refH}/2$ in figure 1.19 and figure 1.21. V_{toti} represents analog equivalent of V_{cali} with i^{th} stage uncalibrated, and ideal/calibrated back end ADC. V_{toti} is calculated according to equation (1.11).

$$V_{toti} = \{V_{bei} \times w_i\} + \left\{\left(D_i - \tfrac{1}{2}\right) \times V_{refH}\right\} \tag{1.11}$$

This value is then subtracted from ideal analog equivalent of calibration input (V_{ideal}) to obtain error (V_{erri}) according to equation (1.12). As mentioned before, V_{ideal} is same as V_{refH}.

$$V_{erri} = V_{ideal} - V_{toti} = V_{refH} - V_{toti} \tag{1.12}$$

Values V_{bei} and V_{erri} are passed to calibration logic comprising of Least Mean Sqaures (LMS) adaptive Algorithm, in order to calculate new value of weight w_i, according to equation (1.13).

$$w_{i(new)} = w_{i(old)} + \mu . V_{erri} . V_{bei} \tag{1.13}$$

Here, μ is the step size. It acts as a factor to adjust convergence speed of the algorithm *i.e.* how fast can appropriate value of w_i be reached that makes $V_{erri} < LSB/4$. Small value of μ implies slow convergence but more accuracy. Large value of μ implies fast convergence but lesser accuracy. However, in this scenario, V_{erri} is a small value, typically closer to some multiple of ADC LSB ($LSB = (V_{ref} - (-V_{ref}))/(2^{12})$). Therefore, even if μ is set to unity, it will still update w_i with a sufficiently small value as desired. Also, it reduces number of multiplications required and hence makes the algorithm faster. Figure 1.20 shows convergence of weight update algorithm for first stage calibration (i=1) with different values of μ.

With new value of w_i, equations (1.11) to (1.13) are repeated unless V_{erri} lesser than LSB/4 is obtained. Once correct value of w_i is obtained, i is decremented and corresponding stage is calibrated next. This goes on until all the stages get calibrated. Weight w_1 is considered to be the final new weight *i.e.* the non-ideal gain of each pipelined stage, since all stages are supposed to be identical. The algorithm is presented in form of a flow chart in figure 1.21. Every time calibration has to take place, ADC has to be given calibration input voltage V_{refH}.

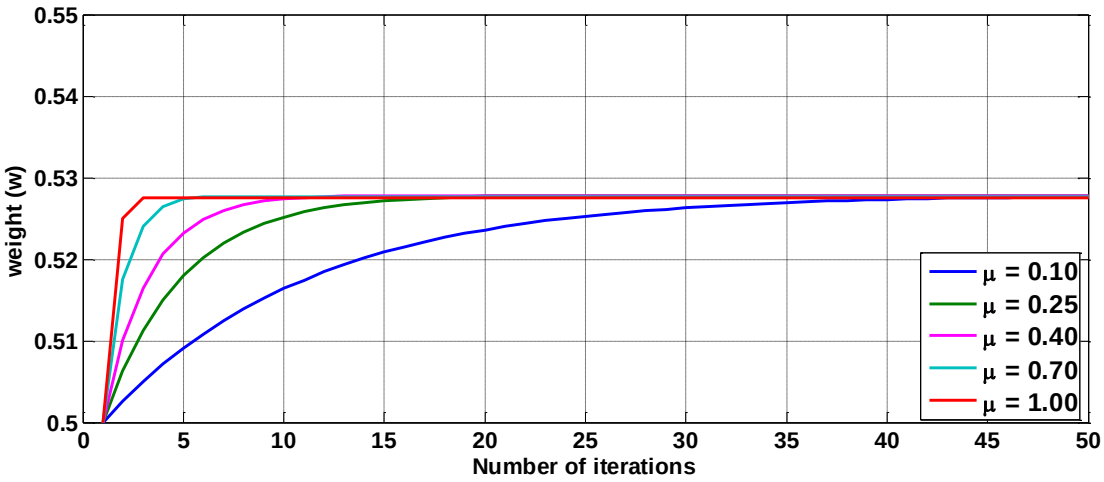

Figure 1.20. Convergence of weight (w_1) of first pipelined stage for different values of step size (μ)

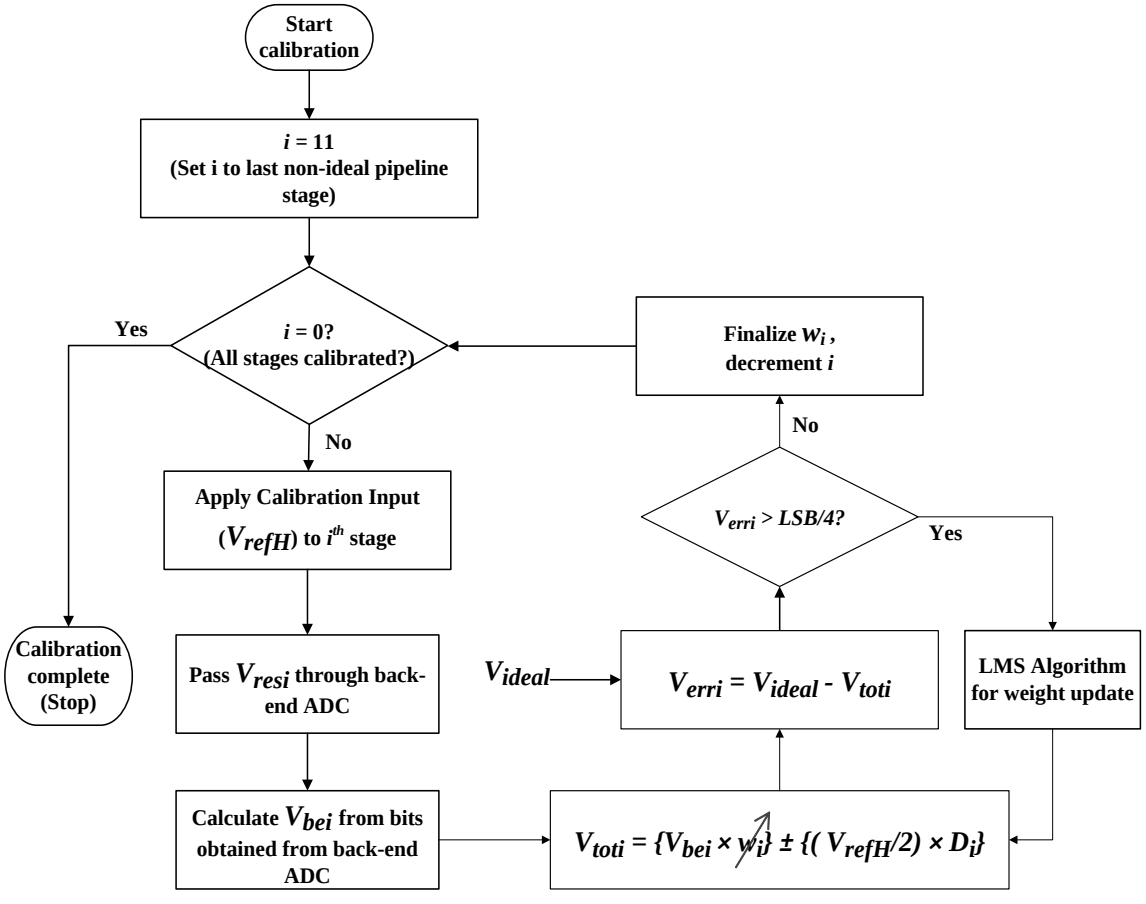

Figure 1.21. Flow chart representing the proposed LMS based calibration algorithm

No modification in ADC internal circuit is required for this calibration scheme. Entire calibration algorithm is implemented digitally using ADC output bits only.

However, it requires the ADC to be idle to perform calibration. Therefore, this scheme is a foreground digital calibration scheme.

4.2 Verilog Implementation

The calibration logic is implemented in Verilog as a finite state machine (FSM). Input to the main module is a clock signal of 50 Hz, and a reset signal. The output is new weight (reciprocal of stage gain) after calibration, and a flag indicating the availability of correct weight at output *i.e.* marks the completion of calibration. As discussed before, value of V_{refH} is stored in memory which is used as V_{ideal} and also in calculation of V_{bei}. Output bits obtained from pipelined ADC for calibration input are stored in memory as well. In case of 12-bit pipelined ADC, when i=11, 5 bits are obtained; for i=10, 6 bits are obtained, and so on. Thus, a total of $5 + 6 + 7 + \cdots + 15 = 110$ bits are obtained for 12-bit ADC. Weights w_1 to w_{15} are also stored in memory, which are read as well as updated during calibration process.

Figure 1.22 shows the block diagram representing interaction of the calibration process with memory.

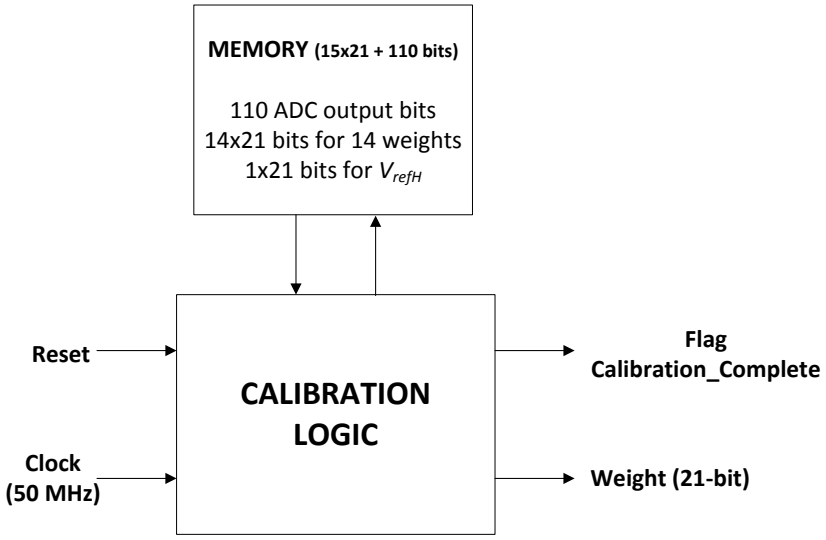

Figure 1.22: Inputs and outputs of calibration process, and memory interface

Before triggering calibration, it is important to reset the calibration circuit so that all the retained values are flushed out. Calibration logic can then be triggered by sending a low value to the reset signal. The whole calibration process is controlled by a main controller FSM shown in figure 1.23 where the text in ovals represents *state*

name/output1/output2. *Output1* corresponds to the calibration process output flag (*Calibration_Complete* as in figure 1.22) that marks completion of calibration process. *Output2* corresponds to the weight obtained after calibration.

For i^{th} pipelined stage calibration (i takes values from 11 to 1 as shown in flow chart shown in figure 1.21), ADC output bits obtained for calibration input from stage i to stage 14 are read from memory one by one. After i^{th} stage is calibrated, i is decremented and corresponding state is calibrated next. This goes on until first pipelined stage is calibrated. Weight (w_1) obtained after calibration of pipelined stage 1 is the correct weight (reciprocal of stage gain) sought for. Table 4.1 explains the functionality implemented in each state of the main calibration controller FSM in detail.

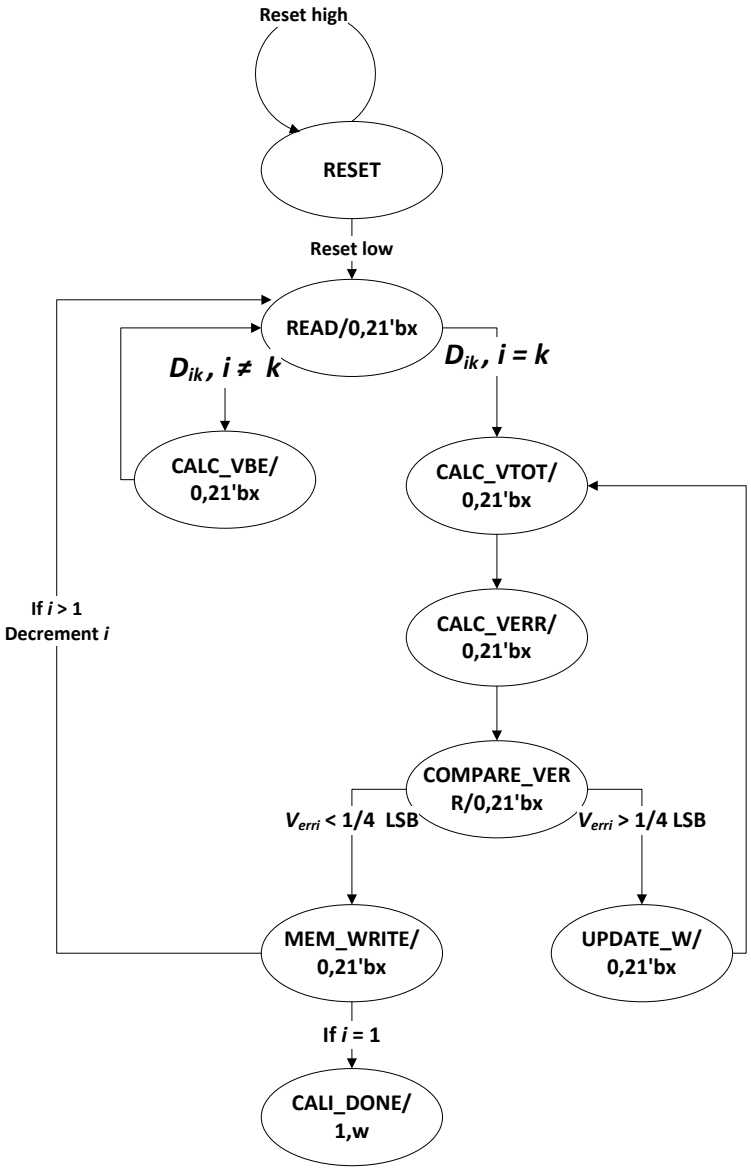

Figure 1.23: Calibration controller FSM

State	Functionality
State 0 (RESET)	FSM is idle in this state and previously retained values are flushed out.
State 1 (READ_BIT)	A bit is read from the memory for i^{th} stage calibration, starting from bit D_{15}. Next time this state is reached bit D_{14} is read, then next time bit D_{13} is read and so on, until bit D_i is read in the end for i^{th} stage calibration. Let us represent the bit read as D_{ik}, where i represents stage under calibration, and k represents the bit read.
State 2 (CALC_VBE)	This state updates the value of V_{bei} according to equation $$V_{bei} = (V_{bei} \times w_k) + (D_{ik} \times \pm V_{refH}/2)$$ When this state is reached for the very first time, value of V_{bei} is taken as 0.
State 3 (CALC_VTOT)	V_{bei} calculated in state 2 is used to calculate V_{toti}. This state is reached when bit read in state 1 is bit D_{ik} with i=k *e.g.* for i=11th stage calibration, when k=11.
State 4 (CALC_VERR)	In this state, V_{erri} is calculated according to equation (1.12)
State 5 (COMPARE_VERR)	V_{erri} is compared against the tolerable error value of ¼ LSB. If $V_{erri} > ¼\ LSB$ then in next state, weight has to be updated.
State 6 (UPDATE_W)	Weight w_i is updated according to equation (1.13). After updating the weight states 3 to 5 are repeated unless $V_{erri} <$ ¼ LSB is yielded in state 5.
State 7 (MEM_WRITE)	Value of w_i obtained from state 6 is written to memory. Also, if i >1 then i is decremented so that in next iteration, bits for corresponding pipelined stage are read. However, if i=1, next state is CALI_DONE, that marks end of calibration process.
State 8 (CALI_DONE)	Weight w_1 is provided as calibration logic output, and flag to mark the completion of calibration is set high.

Table 4.1: Different states of the main calibration controller FSM and their functionality

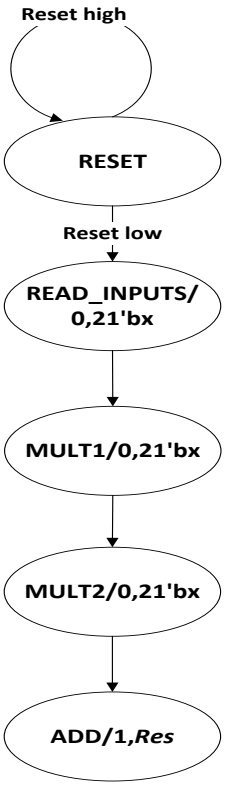

Figure 1.24: FSM for Vbei or Vtoti calculation

State	Functionality
State 0 (RESET)	It is the state when FSM is idle. All the previously retained values are flushed out.
State 1 (READ_INPUTS)	Value the bit (D_{ik}, i = stage under calibration, k = number bit read) received from the calibration controller FSM is checked. If $D_{ik} = 0$ then multiplication factor too be used in next state is set to $-V_{refH}/2$. If $D_{ik} = 1$ then multiplication factor is set to $+V_{refH}/2$.
State 2 (MULT1)	Bit it D_{ik} is multiplied to the multiplication factor obtained from state 1 to obtain intermediate result Res_1. $$Res_1 = \left(D_{ik} \times \pm V_{refH}/2\right)$$
State 3 (MULT2)	V_{bei} obtained from calibration controller FSM is multiplied to weight w_k to obtain intermediate result Res_2. $$Res_2 = (V_{bei} \times w_k)$$
State 4 (ADD)	Final result *Res* is calculated (it can be V_{toti} or updated value of V_{bei}) is calculated by adding Res_1 and $Res_2 i.e\ Res = Res_1 + Res_2$

Table 4.2: Different states of the FSM for V_{bei} calculation and their functionality

The different states of main FSM are also further implemented as FSMs. Figure 1.24 shows FSM for the state where V_{bei} is calculated. The same FSM is used for calculation of V_{toti} as the sequence of operations is same in both the cases. However, when to use this FSM for calculation of V_{bei} and when to use it for calculation of V_{toti} is decided by the calibration controller FSM as shown in figure 1.23. Table 4.2 enlists all the states involved in FSM created for calculation of V_{bei} (or V_{toti}), along with their respective functionalities.

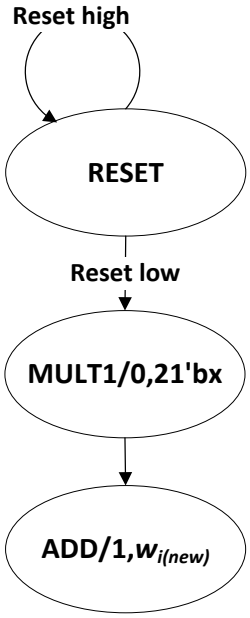

Figure 1.25: FSM for LMS equation for weight update

State	Functionality
State 0 (RESET)	It is the state when FSM is idle. All the previously retained values are flushed out.
State 1 (MULT1)	In this state, V_{bei} and V_{erri} obtained from calibration controller FSM are multiplied to obtain intermediate result, say *WUpdateFactor*. $$WUpdateFactor = V_{bei} \times V_{erri}$$
State 2 (ADD)	In this state, factor *WUpdateFactor* is added to initial value of weight w_i to obtain the updated weight. $$w_{i(new)} = w_{i(old)} + WUpdateFactor$$

Table 4.3: Different states of the LMS weight update FSM and their functionality

Figure 1.25 shows implementation of LMS update equation, $w_{i(new)} = w_{i(old)} + V_{erri}.V_{bei}$ in the form of FSM. Note that this equation is obtained from equation (1.13). As mentioned before, value of μ is taken as unity in this work. Hence μ can be eliminated to avail the benefit of reduced number of multiplications. Table 4.3 enlists different states involved in LMS weight update FSM and their respective functionality.

It is important to note that all the numbers are represented in floating point representation. It has been observed that 21-bit representation proves to be appropriate to maintain the precision of results of arithmetic operations involved. With any number of bits lesser than 21, the precision is lost so notably that DNL and INL errors show less improvement after calibration. The 21-bit floating point representation used in this work is shown in figure 1.26. There is one bit for sign representation, six bits for exponent and fourteen bits for mantissa. One bit with value '1' is implied and is not stored explicitly.

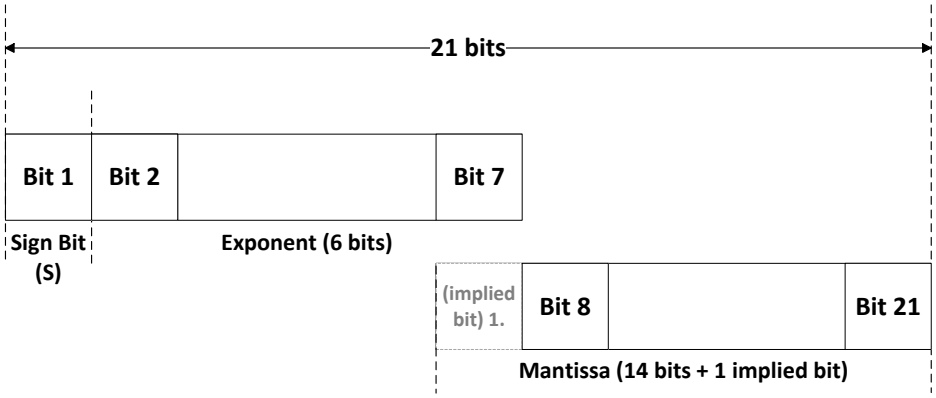

Figure 1.26: 21-bit floating point representation

To convert the 21-bit floating point binary number to decimal (say $(Number)_{10}$), the following formula is used,

$$(Number)_{10} = (-1)^S \times (Mantissa) \times 2^{Exponent-bias}$$

where *S* represents sign bit, *Mantissa* represents 15 bit mantissa (1 implied bit and 14 explicitly stored mantissa bits) and *Exponent* represents 6 bit exponent value. *bias* is the offset which is added in binary floating point form in order to accommodate negative exponents. Thus for 6 bit exponent, *bias* value is 31 (011111 in binary) which makes exponent values from -30 to 31 to be accommodated in 6 bit binary representation.

For arithmetic operations, 21-bit floating point comparator, adder and multiplier blocks are required. In this work, these blocks are taken from Xilinx IP Core DS-335. The

design units from Xilinx IP Core are available as black boxes. However, flexibility is provided to choose the number of exponent and mantissa bits in floating point representation of operands and result.

Chapter 5

SIMULATION RESULTS

5.1 MATLAB Simulation Results

A 12-bit, 1.5-bit per stage pipelined ADC was modelled in MATLAB, with input signal sampled at the rate of 100 MSamples/sec. It has 11 pipelined stages, and a 2-bit flash ADC as last stage. All the pipelined stages are made identical. This modelled ADC behaves ideally unless some error is deliberately induced to it. Figure 1.27 shows residue plot of first stage with respect to ADC input, which is taken as a ramp signal in range $-V_{ref}$ to $+V_{ref}$, sampled at 100 MSamples/sec. Figure 1.28 shows the residue plot of second stage with respect to ADC input.

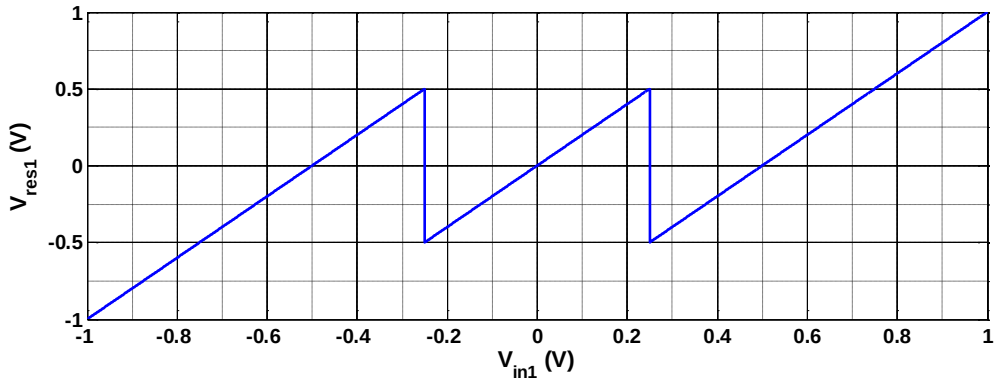

Figure 1.27: Residue plot of first pipelined stage of ideal 12-bit, 1.5-bit per stage pipelined ADC

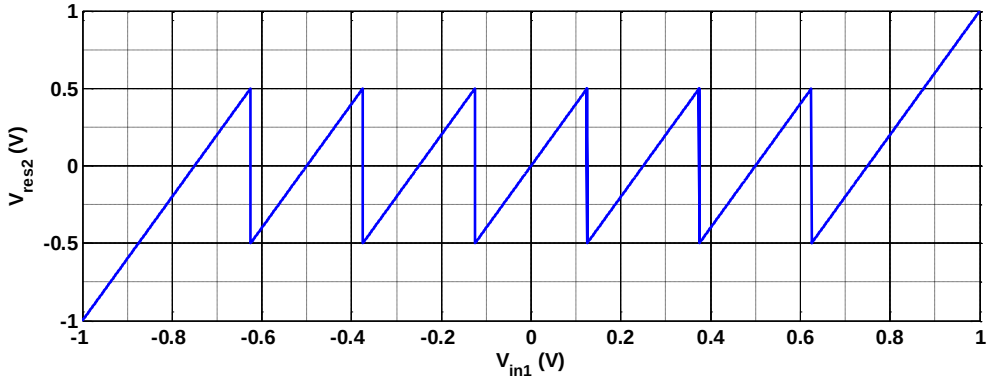

Figure 1.28: Residue plot of second pipelined stage of ideal 12-bit, 1.5-bit per stage pipelined ADC

In order to simulate a practical ADC, gain error is introduced in ADC. To do so, stage gain of ~1.89 (with open loop gain of op-amp taken as 40 V/V) is used in each stage instead of ideal value 2. Figure 1.29 shows residue plot of first stage of practical 12-bit, 1.5 bit per stage pipelined ADC, when open loop gain of op-amp is taken as 40 V/V. Figure 1.30 shows residue plot of second stage of practical 12-bit, 1.5 bit per stage pipelined ADC, when open loop gain of op-amp is taken as 40 V/V in both, first and second stages.

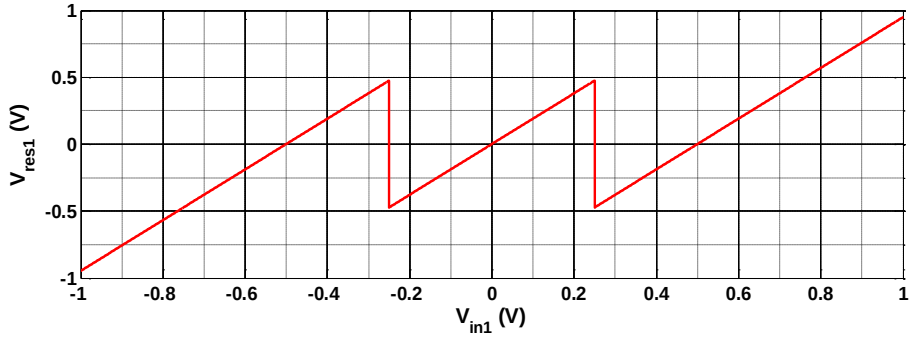

Figure 1.29: Residue plot of first pipelined stage with op-amp open loop gain of 40 V/V

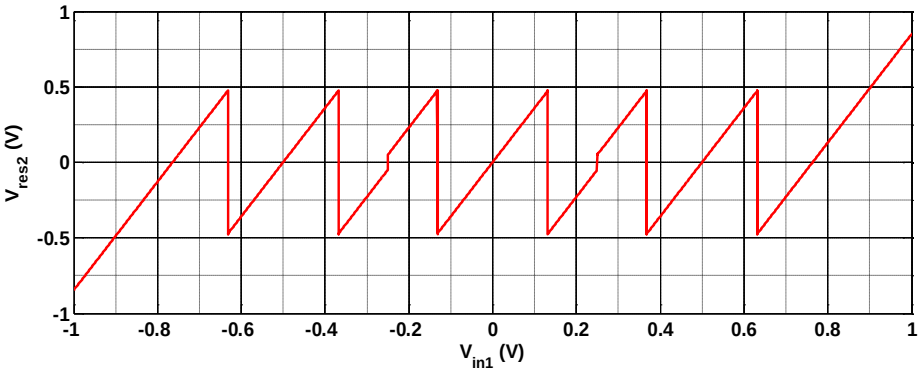

Figure 1.30: Residue plot of second pipelined stage with op-amp open loop gain of 40 V/V in both, first and second stages

Figure 1.31 shows differential non-linearity (DNL) and integral non-linearity (INL) of practical ADC with gain error. DNL lies in 0.6296/-1 LSB and INL lies in the range of -60/60 LSB. To correct this gain error, the foreground calibration algorithm discussed in chapter 4 is also modelled in MATLAB. After calibration, the DNL and INL values of +0.7/-0.63 LSB and +0.77/-0.77 LSB are achieved respectively. Figure 1.32 shows DNL and INL after calibration. DNL within ±1 LSB post calibration indicate that there are no missing codes.

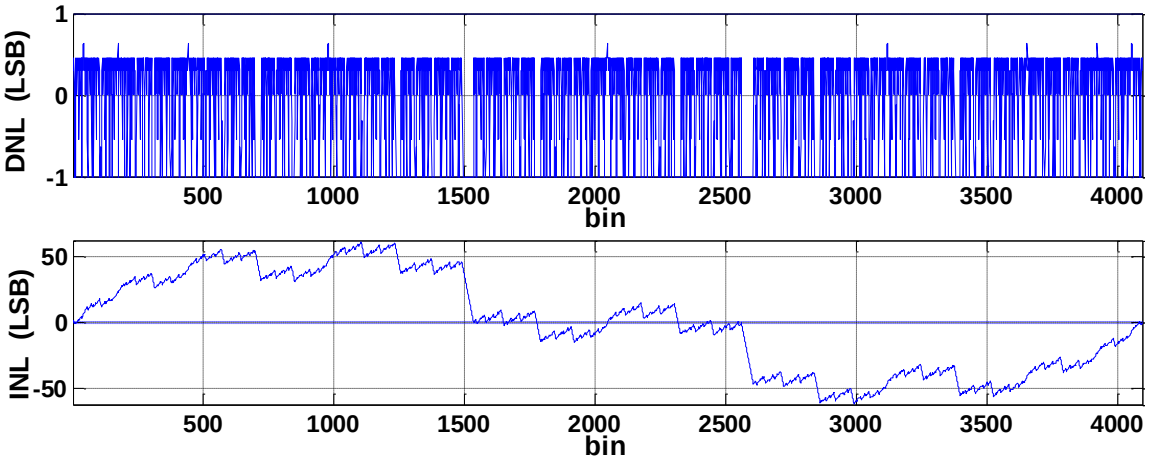

Figure 1.31: DNL and INL of 12-bit, 1.5 bit per stage pipelined ADC, with gain error in all pipelined stages

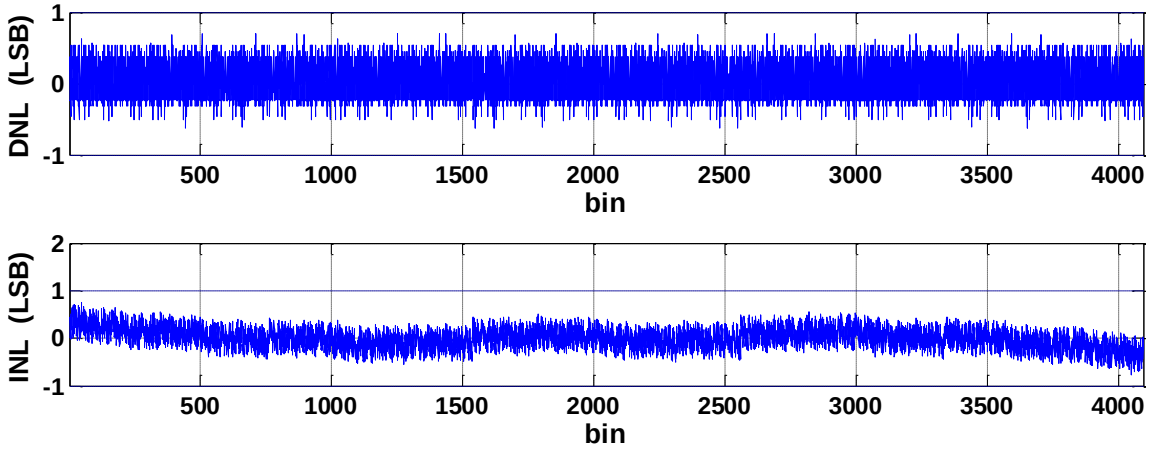

Figure 1.32: DNL and INL of 12-bit, 1.5 bit per stage pipelined ADC, with gain error in all pipelined stages

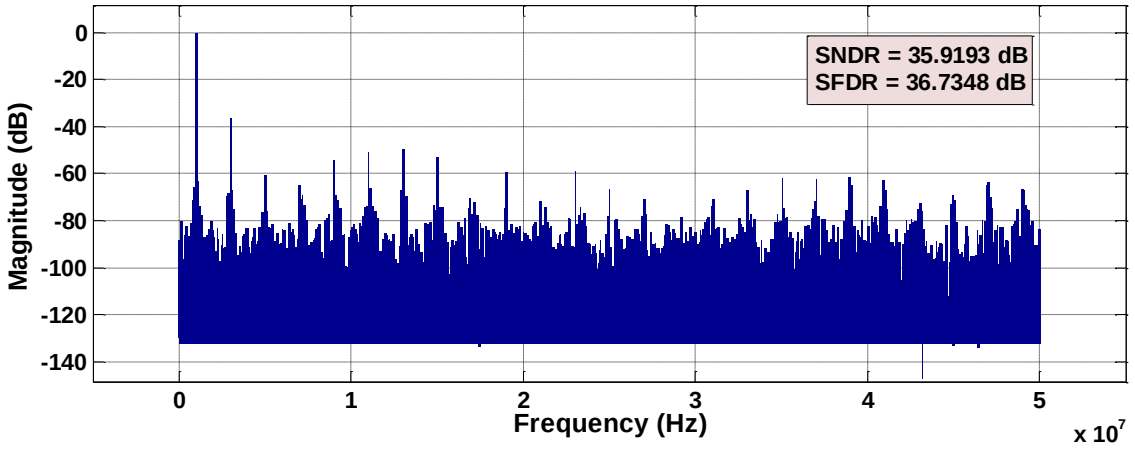

Figure 1.33. Frequency response of uncalibrated ADC

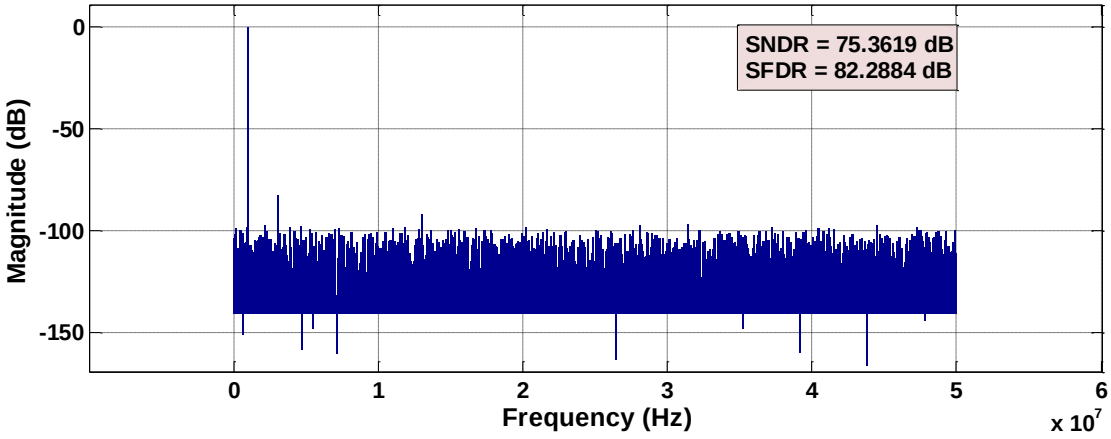

Figure 1.34. Frequency response of calibrated ADC

Figure 1.33 and figure 1.34 show frequency response of the pipelined ADC under test, before and after calibration respectively. It is evident that SNDR and SFDR improve significantly from 35.9193 dB and 36.7348 dB to 75.3619 dB and 82.2884 dB respectively after calibration. Effective number of bits (ENOB) after calibration comes out to be 12.22.

5.2 Verilog Simulation Results

Calibration logic is also implemented in Verilog as discussed in detail in previous chapter. Xilinx ISE Design Suite 14.6 is used for Verilog programming. Xilinx ISE Simulator (ISim) is used for simulation.

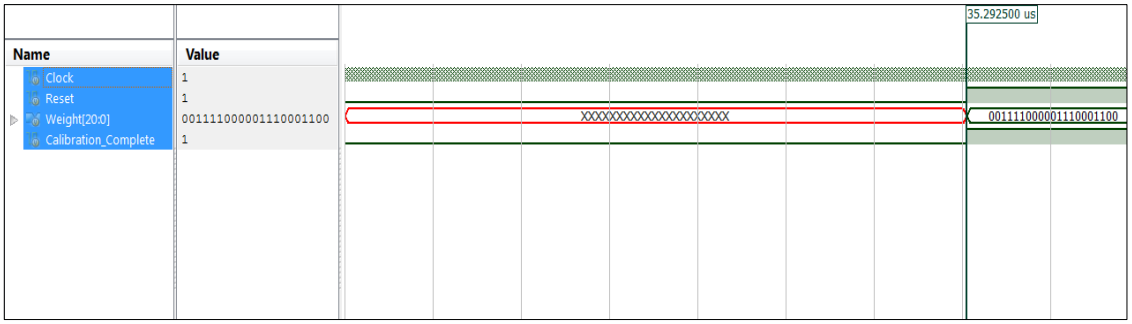

Figure 1.35. Verilog simulation waveforms

Figure 1.35 shows the waveforms representing input and output signals of calibration logic. Signals ‘Clock’ and ‘Reset’ are the inputs; flag ‘Calibration_Complete’ and 21-bit wide signal ‘Weight’ are the outputs (Refer figure 1.22 for detail). It can be

noted from figure 1.35 that calibration logic completes its task of determining radix in approximately 35.3 µsec. The output signal 'Weight' has 21-bit binary value '001111000001110001100'. This is 21-bit floating point representation of weight (reciprocal of stage gain) obtained after calibration. As shown in figure 1.36, first bit represents sign bit, next 6 bits represent biased exponent, and next 14 bits form a part of 15-bit mantissa.

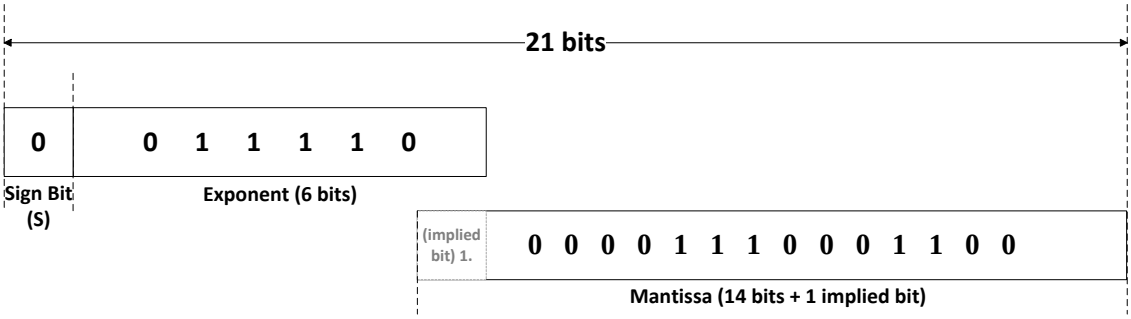

Figure 1.36: Sign bit, exponent bits, mantissa bits of weight obtained after calibration

Thus the result can be interpreted as follows,

$$Weight_{decimal} = (-1)^{S} \times (Mantissa) \times 2^{Exponent-bias}$$

where $Weight_{decimal}$ represents decimal equivalent of 21-bit floating point binary representation and *bias* for 6-bit exponent is 31 (011111 in binary).

$$\begin{aligned} => Weight_{decimal} &= (-1)^{0} \times (1.00001110001100) \times 2^{011110-011111} \\ &= 1 \times (1.055419921875) \times 2^{30-31} \\ &= 1.055419921875 \times 2^{-1} \\ &= 0.52770996 \end{aligned}$$

This weight ($Weight_{decimal}$) is the reciprocal of stage gain.

$$stage\ gain = \frac{1}{Weight_{decimal}} = \frac{1}{0.52770996} \cong 1.89498$$

Figure 1.37 shows the input clock signal clearly. Clock has period of 20 ns *i.e.* frequency of $1/(20 \times 10^{-9})$ = 50 MHz.

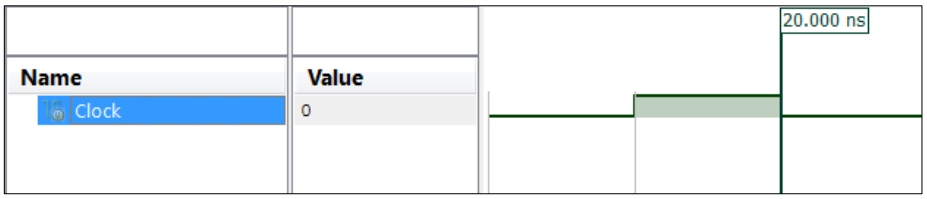

Figure 1.37: Input clock signal

The logic implemented is also tested on Spartan-6 FPGA SP605 evaluation kit. Table 5.1 enlists utilization in numbers and percentage of various components on the evaluation kit. The table is generated by Xilinx ISE as a part of design summary. It can be observed that the calibration logic uses very less percentage of the available resources.

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	417	54576	~ 0%
Number of Slice LUTs	2163	27288	~ 7%
Number of fully used LUT-FF pairs	109	2471	~ 4%
Number of bonded IOBs	24	296	~ 8%
Number of BUFG/BUFGCTRLs	2	16	~ 12%

Table 5.1: Device utilization of the calibration logic for FPGA SP605 evaluation kit

Chapter 6

CONCLUSION AND FUTURE SCOPE

The foreground digital calibration technique for pipelined ADC proposed in this work proves to improve the static and dynamic performance metrics hugely as discussed in previous chapter and is time efficient also. Table 6.1 compares this work to different prior works done on calibration of 12-bit pipelined ADC with 100 MSamples/sec input sampling rate.

Parameters		**[10]**	**[14]**	**[12]**	**[13]**	**[11]**	**This work**
Resolution (bits)		12	12	12	12	12	12
Sampling Rate (MS/s)		20	80	100	20	200	100
Calibration Method		background	Foreground and background	background	-	foreground	foreground
DNL (LSB)	Before calibration	-0.60/+0.60	1	-	1.47	-	+0.61/-1
	After calibration	+0.41/-0.41	0.09	+0.1/-0.08	0.27	-	+0.7/-0.63
INL (LSB)	Before calibration	+4.21/-4.21	20.5	1.9	7.85	-	+60.30/-60.25
	After calibration	+0.47/-0.47	0.24	+0.4/-0.6	0.2	-	+0.77/-0.77
SNDR (dB)	Before calibration	58.2	50.1	-	41.3	30	35.92
	After calibration	70.8	72.6	66	72.5	68.5	75.36
SFDR (dB)	Before calibration	59.4	52.3	66	52.5	32.4	36.73
	After calibration	93.3	84.5	85	84.4	80	82.29
ENOB (bits)		11.5	11.8	-	11.8	11.13	12.22

Table 6.1: Comparison of this work to prior works on 12-bit pipelined ADC calibration

It can be noted that the foreground calibration method discussed in this work achieves greatest SNDR and ENOB as compared to other works that focus on calibration of a 12-bit pipelined ADC.

Since there is no requirement of ideal reference ADC, the proposed method does not suffer from delay of slow reference ADC. Complete calibration algorithm works external to the ADC, without any modification required in internal architecture of pipelined ADC. Moreover, the method shows very fast convergence for unity step size, as evident from figure 1.20. INL improves largely from +60.30/-60.25 LSB to +0.77/-0.77 LSB. DNL shows a change from +0.61/-1 LSB to +0.7/-0.63 LSB. It can be observed that maximum DNL before calibration was 1 LSB. After calibration, maximum DNL achieved is 0.7. Therefore 30% improvement occurs in DNL. SNDR and SFDR values after calibration are 75.36 dB and 82.29 dB respectively.

The logic implemented in Verilog can be made to work faster by developing fast adder and multiplier to replace Xilinx IP Core DS-335 floating point arithmetic blocks. Moreover, efforts can be put to develop a digital algorithm that does not require extra pipelined stages to be inserted during calibration process. It will significantly reduce hardware and power consumption of calibration logic.

REFERENCES

[1] A. N. Karanicolas, H. S. Lee, and K. L. Bacrania, “A 15-b 1-Msample/s Digitally Self-Calibrated Pipeline ADC,” *IEEE J. Solid-State Circuits*, vol. 28, no. 12, pp. 1207–1215, 1993.

[2] Un-Ku Moon and Bang-Sup Song, "Background digital calibration techniques for pipelined ADCs", *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 44, no. 2, pp. 102-109, 1997.

[3] B. Murmann and B. E. Boser, “A 12-bit 75-MS/s Pipelined ADC Using Open-Loop Residue Amplification,” in *IEEE Journal of Solid-State Circuits*, 2003, vol. 38, no. 12, pp. 2040–2050.

[4] J. Li and U. K. Moon, “Background calibration techniques for multistage pipelined ADCs with digital redundancy,” *IEEE Trans. Circuits Syst. II Analog Digit. Signal Process.*, vol. 50, no. 9, pp. 531–538, 2003.

[5] Y. Chiu, C. W. Tsang, B. Nikolic, and P. R. Gray, “Least mean square adaptive digital background calibration of pipelined analog-to-digital converters,” *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 51, no. 1, pp. 38–46, 2004.

[6] J. Keane, P. Hurst and S. Lewis, "Background interstage gain calibration technique for pipelined ADCs", *IEEE Trans. Circuits Syst. I*, vol. 52, no. 1, pp. 32-43, 2005.

[7] C. R. Grace, P. J. Hurst, and S. H. Lewis, “A 12-bit 80-MSample/s pipelined ADC with bootstrapped digital calibration,” *IEEE J. Solid-State Circuits*, vol. 40, no. 5, pp. 1038–1046, 2005.

[8] C. Tsang, Y. Chiu, J. Vanderhaegen, S. Hoyos, C. Chen, R. Brodersen, and B. Nikolic, “Background ADC calibration in digital domain,” in *Proceedings of the Custom Integrated Circuits Conference*, 2008, pp. 301–304.

[9] I. Ahmed and D. Johns, "An 11-Bit 45 MS/s Pipelined ADC With Rapid Calibration of DAC Errors in a Multibit Pipeline Stage", *IEEE J. Solid-State Circuits*, vol. 43, no. 7, pp. 1626-1637, 2008.

[10] L. Brooks and H. S. Lee, “Background calibration of pipelined ADCs via decision boundary gap estimation,” *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 55, no. 10, pp. 2969–2979, 2008.

[11] A. Delić-Ibukić and D. M. Hummels, “Continuous digital calibration of pipeline A/D converters,” *IEEE Trans. Instrum. Meas.*, vol. 55, no. 4, pp. 1175–1185, 2006.

[12] K. Sun and L. He, "A fast combination calibration of foreground and background for pipelined ADCs", *Journal of Semiconductors*, vol. 33, no. 6, p. 065007, 2012.

[13] S. Sarkar, Y. Zhou, B. Elies and Y. Chiu, "PN-Assisted Deterministic Digital Background Calibration of Multistage Split-Pipelined ADC", *IEEE Trans. Circuits Syst. I*, vol. 62, no. 3, pp. 654-661, 2015.

[14] J. A. McNeill, M. C. W. Coln, D. R. Brown, and B. J. Larivee, "Digital background-calibration algorithm for 'Split ADC' architecture," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 56, no. 2, pp. 294–306, 2009.

[15] M. Mohammadi, A. Alipour, E. Aghdam and K. Sadeghipour, "A new technique for background calibration of pipelined ADCs", *2013 21st Iranian Conference on Electrical Engineering (ICEE)*, 2013.

[16] J. Yuan, S. W. Fung, K. Y. Chan, and R. Xu, "An interpolation-based calibration architecture for pipeline ADC with nonlinear error," *IEEE Trans. Instrum. Meas.*, vol. 61, no. 1, pp. 17–25, 2012.

[17] H. Mafi and A. Sodagar, "A background calibration in pipelined ADCs", *AEU - International Journal of Electronics and Communications*, vol. 67, no. 8, pp. 729-732, 2013.

[18] D. Meganathan, A. Sukumaran, M. Dinesh Babu, S. Moorthi and R. Deepalakshmi, "A systematic design approach for low-power 10-bit 100MS/s pipelined ADC", *Microelectronics Journal*, vol. 40, no. 10, pp. 1417-1435, 2009.

[19] A. Singh and A. Agarwal, "Digital background calibration of charge pump based pipelined ADC", *International Journal of Electronics*, pp. 1-13, 2016.

[20] A. Larsson and S. Sonkusale, "A background calibration scheme for pipefined ADCs including non-linear operational amplifier gain and reference error correction", *IEEE International SOC Conference, 2004. Proceedings.*

[21] S. Roy, H. Basak and S. Banerjee, "Foreground calibration technique of a pipeline ADC using capacitor ratio of Multiplying Digital-to-Analog Converter (MDAC)", *Microelectronics Journal*, vol. 44, no. 12, pp. 1336-1347, 2013.

[22] Z. Liren, L. Lei, Y. Fan, X. Jun and R. Junyan, "A 12-bit 100 MS/s pipelined ADC with digital background calibration", *Journal of Semiconductors*, vol. 30, no. 11, p. 115007, 2009.

[23] M. Abdelhamid, K. Megawer, F. Hussien and M. Aboudina, "A 12-bit, 200MS/s digitally calibrated pipeline ADC with Embedded Sample and Hold", *MELECON 2014 - 2014 17th IEEE Mediterranean Electrotechnical Conference*, 2014.

[24] Xiaoyue Wang, P. Hurst and S. Lewis, "A 12-bit 20-Msample/s pipelined analog-to-digital converter with nested digital background calibration", *IEEE J. Solid-State Circuits*, vol. 39, no. 11, pp. 1799-1808, 2004.

[25] I. Ahmed, J. Mulder and D. Johns, "A Low-Power Capacitive Charge Pump Based Pipelined ADC", *IEEE J. Solid-State Circuits*, vol. 45, no. 5, pp. 1016-1027, 2010.

[26] L. Zhang, "The Calibration Technique for Pipelined ADC", *2008 International Conference on MultiMedia and Information Technology*, 2008.

[27] E. Fatemi-Behbahani, E. Farshidi and K. Ansari-Asl, "A new digital background calibration for redundant radix-4 pipelined ADCs by modeling of adaptive filter for linear and nonlinear errors", *Measurement*, vol. 83, pp. 123-134, 2016.

[28] B. N. Fang and J. T. Wu, "A 10-bit 300-MS/s pipelined ADC with digital calibration and digital bias generation," *IEEE J. Solid-State Circuits*, vol. 48, no. 3, pp. 670–683, 2013.

[29] L. H. Hung and T. C. Lee, "A split-based digital background calibration technique in pipelined ADCs," *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 56, no. 11, pp. 855–859, 2009.

[30] A. Singh and A. Agarwal, "Power and Area Efficient Pipelined ADC Stage in Digital CMOS Technology", *IETE Technical Review*, pp. 1-9, 2016.

[31] M. Steyaert, A. Roermund and H. Casier, *Analog circuit design*. Dordrecht: Springer, 2010.

[32] "Imran Ahmed's website", *Iadc.ca*, 2016. [Online]. Available: http://www.iadc.ca/ADC_architectures_tutorial.htm#1.6. [Accessed: 03- Jul- 2016].

[33] "Understanding Pipelined ADCs - Tutorial - Maxim", *Maximintegrated.com*, 2016. [Online]. Available: https://www.maximintegrated.com/en/app-notes/index.mvp/id/1023. [Accessed: 03- Jul- 2016].

[34] N. Gray, "ABCs of ADCs Analog-to-Digital Converter Basics," *Natl. Semicond. Signal Sound Inf.*, no. June, p. 64, 2006.

PUBLICATION

Jupinder Kaur, Alpana Agarwal, "Fast foreground digital calibration method for pipelined ADC", *International Journal of Electronics and Communication* (Impact factor: 0.786), *Elsevier,* 2016. **(Communicated)**

801461012_Thesis.docx

ORIGINALITY REPORT

16%	%	16%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Meganathan, D.. "A systematic design approach for low-power 10-bit 100MS/s pipelined ADC", Microelectronics Journal, 200910 Publication	1%
2	Singh, Anil, and Alpana Agarwal. "Digital Background Calibration of Charge Pump Based Pipelined ADC", International Journal of Electronics, 2016. Publication	1%
3	Zhou Liren. "A 12-bit 100 MS/s pipelined ADC with digital background calibration", Journal of Semiconductors, 11/2009 Publication	1%
4	Lane Brooks. "Background Calibration of Pipelined ADCs Via Decision Boundary Gap Estimation", IEEE Transactions on Circuits and Systems I Regular Papers, 11/2008 Publication	1%
5	Jin-Fu Lin. "A Reduced Code Linearity Test Method for Pipelined A/D Converters", 2008 17th Asian Test Symposium, 11/2008 Publication	1%