Low power digital type analog-to-digital converter

Ng, Richard Wee Tar

2011

Ng, R. W. T. (2011). Low power digital type analog-to-digital converter. Master's thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/47580>

<https://doi.org/10.32657/10356/47580>

Low Power Digital Type Analog-to-Digital Converter

Ng Wee Tar Richard

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Master of Engineering

2011

Acknowledgements

This thesis has only come to completion with the help and support of several individuals. First I must thank my supervisor, Prof. Siek Liter for his invaluable advice and guidance throughout the course of this research work. I thank Christoph Braun for sharing his patent. I gratefully acknowledge Infineon Technologies Asia Pacific for sponsoring my research studies in Nanyang Technology University.

Finally, I would not have accomplished anything without the support from my family. Importantly, I have to thank my wife who has been at my side for the entire course of work, offering valuable advice for my presentations and research paper writings.

Table of Contents

Acknowledgements	i
Table of Contents	ii
Summary	v
List of Figures	vii
List of Tables	xi
1 Introduction.....	1
1.1 Motivation	2
1.2 Contributions	3
1.3 Thesis Outline.....	5
2 Full Digital Analog-to-Digital Converters	6
2.1 Full Digital ADCs Architectures	6
2.1.1 Voltage Line to Delay Conversion.....	7
2.1.2 Statistical Sampling	8
2.1.3 Stochastic Logic Sampling	11
2.1.4 Linear pulse modulation with constant impulse amplitude	13
2.2 Analysis of Full Digital ADCs Architectures	16
2.3 Summary and Objective.....	21
3 Overview and Modeling of D-ADC.....	23
3.1 Analog Front End	23
3.2 Digital Filtering	26
3.3 Matlab Modeling of D-ADC	27
3.3.1 System Modeling of D-ADC	27
3.3.2 Modeling of Analog Front End.....	31
3.3.3 Modeling of Digital Processing Unit	35

3.4	Conclusion.....	36
4	Power Reduction Proposal for Digital Unit in D-ADC.....	37
4.1	Decimation and Interpolation.....	38
4.1.1	The Decimation Process	38
4.1.2	The Interpolation Process	41
4.1.3	Frequency Translation by Decimation	44
4.1.4	Polyphase Decomposition	45
4.2	A Simplified Approach to Baseband Recovery	49
4.3	A Simplified Approach to Baseband Recovery in D-ADC	52
4.3.1	Determining the carrier frequency in D-ADC	52
4.3.2	System Modeling with the Simplified Approach.....	53
4.3.3	Possible Savings in Gate Count and Power.....	55
4.4	Digital Band-pass Filter Reduction	57
4.4.1	No Interpolation	57
4.4.2	Efficient Implementation Using Polyphase Decomposition	58
4.4.3	Coefficients Silicon Footprint Reduction.....	60
4.4.4	Possible Savings in Gate Count and Power.....	62
4.5	Conclusion.....	62
5	Possible Enhancements for Analog Unit in D-ADC.....	63
5.1	Effects of Sampling Rate Reduction.....	63
5.2	Differential Scheme	64
5.2.1	Differential with Dithering Scheme	66
5.2.2	Trade-Offs	67
5.3	Conclusion.....	68
6	Realization of Hardware with VHDL	69
6.1	Realization of Hardware with VHDL.....	70
6.1.1	Analog Front End Unit	72
6.1.2	Averaging Unit.....	73
6.1.3	Band-pass Filter Unit	74
6.1.4	Gate Count and Power Estimation	75
6.2	Conclusion.....	77

7	Experimental Results.....	78
7.1	Test Setup.....	78
7.1.1	Equipment Setup.....	79
7.2	Experimental Results	81
7.2.1	Sampling Signal Amplitude and its Quantized Step Voltage.....	82
7.2.2	Idle Tone Analysis	82
7.2.3	Dynamic Testing	83
7.2.4	DNL and INL Measurements	86
7.2.5	Discussion on Experimental Results	87
7.2.6	Discussion on Differential Scheme with Dithering for AFE.....	89
8	Conclusions and Recommendations	91
8.1	Recommendations for Future Research	91
8.1.1	Silicon Footprint Reduction for Analog Front End	91
8.1.2	Power Reduction for Analog Front End.....	93
	Author's Publications.....	94
	Bibliography	95
	Appendix A.....	97
A.1	VHDL of Analog Front End Unit Zero-Crossing Detection.....	97
A.2	VHDL of Averaging Unit	98
A.3	VHDL of Band-pass Filtering Calculation	101
A.4	Captured waveform of Zero-Crossing Detection	102

Summary

As complementary metal–oxide–semiconductor (CMOS) technology rapidly progresses into deep-sub micron scales, analog type Analog-to-Digital Converters (ADCs) face mounting challenges from reduced signal-to-noise ratio (SNR), lower intrinsic gain in CMOS devices, increased device leakage, larger transistor mismatches, and lack of high quality passives such as inductors and capacitors in lower transistor geometries. Digital functions instead have benefited from this technology scaling into smaller scales. Due to increased challenges of analog integration into these smaller scales, there is a drive to push these analog functions into the digital domain. Various digital type ADCs have been proposed, but these ADCs faced challenges from having large silicon area, and high power consumptions as compared to their analog counterparts.

In this thesis, Christoph Braun’s approach to a digital type ADC is studied and improved in the areas of power consumption and bandwidth. Christoph’s approach is adopted because of its silicon proven measurement with a good non-linearity of $\pm 0.024\%$ Full Scale (FS) performance result. The approach to reduce power consumption in this digital type ADC is by digital gate count reduction, and lowering of the operating frequency of these digital gates in the digital processing unit. Proposal for possible enhancements in the analog front end (AFE) is also done. The reduction in operating frequency of the AFE circuitries will pave the way for higher bandwidth in this ADC. The reduction in operating frequency will allow a higher operating frequency to be reused in the AFE when a higher bandwidth is needed.

The enhanced digital type ADC is modeled in Matlab showing a dynamic SNR performance of up to -90 dB with a high sampling rate of greater than 1 GHz in the AFE. The digital part of the enhanced digital type ADC is implemented in VHSIC hardware description language (VHDL), and verified to be correct functionally with a digital test bench. The digital part of the design is synthesized with a 0.18 μm technology node, and the estimated gate count of a single channel ADC including the AFE is approximately 0.3573 mm^2 . The silicon area is 30% less than

the original design which has an estimated single channel active area of 0.525 mm^2 published in its paper. The estimated power consumption is 1.8631 mW for a single channel operation. This estimates to be 50% less than the original design which consumes about 3.6 mW .

The enhanced D-ADC being an almost full digital design is implemented and measured on a Xilinx Virtex 4 ML402 Field Programmable Gate Array (FPGA) platform, with the AFE part represented by discrete passive resistors and capacitors mounted on a veroboard. The converted data has a resolution of 16 digital bits. With an AFE sampling rate of 300 MHz , a maximum dynamic SNR of -62 dB is obtained with analog input amplitude of 2.68 V peak-to-peak. The acceptable input bandwidth is $0 - 20 \text{ kHz}$. The high sampling rate is needed as to correctly detect the zero-crossings in the AFE. Analog input voltage can range from $0.00 - 3.35 \text{ V}$ maximum depending on the voltage type used on the FPGA device. The measured differential non-linearity is between $+1.04 \text{ LSB}$ and -0.88 LSB . The measured integral non-linearity is between $+5.72 \text{ LSB}$ and -0.45 LSB . Effective number of bits (ENOB) is measured up to 10.0 bits. Figure of Merit (FOM) is estimated to be 0.455 pJ/level .

List of Figures

Figure 2.1: Voltage-line delay conversion.	7
Figure 2.2: Delay chain architecture.	8
Figure 2.3: Statistical sampling architecture.	9
Figure 2.4: Probability Density Function shift concept.	10
Figure 2.5: Stochastic ADC design.	12
Figure 2.6: Overall DAD architecture.	13
Figure 2.7: DAD front end sampling architecture.	14
Figure 2.8: 1 st order frequency spectrum of the time location of the dirac pulses.	14
Figure 2.9: DAD digital signal processing basics.	15
Figure 2.10: Delay Vs the supply voltage of a DE.	18
Figure 2.11: Linearity result implementation from first and second approach.	18
Figure 3.1: Overall D-ADC architecture.	23
Figure 3.2: 1 st order frequency spectrum of the time location of the dirac pulses.	25
Figure 3.3: High speed counters to approximate the location of the impulses with respect to the carrier sine wave.	26
Figure 3.4: D-ADC digital signal processing basics.	27
Figure 3.5: Matlab system modeling of D-ADC.	29
Figure 3.6: Frequency spectrum of sampled analog input freq with low sampling zero crossing detection freq.	30
Figure 3.7: Frequency spectrum of sampled analog input freq with high sampling zero crossing detection freq.	31
Figure 3.8: D-ADC Analog Front End.	32
Figure 3.9: Matlab model of D-ADC Analog Front End.	33

Figure 3.10: (a) Noise-shaped digital sine ROM data (b) Analog sine carrier wave (c) Zero-crossings from the modulated sine wave.	34
Figure 3.11: Frequency spectrum of sampled analog input freq with high sampling zero crossing detection freq.....	36
Figure 4.1: The decimation process with a filter preceding the decimator.	38
Figure 4.2: Decimation process for $D=2$. (a) Input sequence, $x(n)$ (b) Output sequence for $y(m)$	39
Figure 4.3: (a) Original signal spectrum, $X(e^{j\omega})$, and (b) Resultant down-sampled spectrum, $Y(e^{j\omega})$, if $-\pi/D < X(e^{j\omega}) < \pi/D$, and (c) Result down-sampled spectrum, $Y(e^{j\omega})$, if $-\pi/D > X(e^{j\omega}) > \pi/D$	40
Figure 4.4: The interpolation process with a filter following the expander.	41
Figure 4.5: Interpolation process for $L=2$. (a) Input sequence, $x(n)$, and (b) Output sequence for $y(m)$, and (c) Interpolated zero-valued samples, $u(n)$	42
Figure 4.6: (a) Original signal spectrum, $X(e^{j\omega})$, and (b) Resultant up-sampled spectrum, $Y(e^{j\omega})$, and (c) Output spectrum of the anti-image filter, $U(e^{j\omega})$	43
Figure 4.7: The frequency translation by decimation process.....	44
Figure 4.8: Frequency translation by decimation for $m=2$	45
Figure 4.9: M-decimation filter implemented on (a) Direct form, (b) Polyphase decomposition, and (c) First cascaded equivalence.	47
Figure 4.10: L-interpolation filter implemented on (a) Direct form, (b) Polyphase decomposition, and (c) First cascaded equivalence.	48
Figure 4.11: Original signal and resultant down-sampled spectrum.	49
Figure 4.12: Original signal and resultant down-sampled spectrum	50
Figure 4.13: Analog input signal recovered by decimation	51
Figure 4.14: Time domain operation of multiplication-less.....	51
Figure 4.15: System model for D-ADC with multiplier-less removal of carrier frequency and recovery of the signal components.	54
Figure 4.16: Frequency spectrum extracted by multiplication-less removal of carrier frequency.	55
Figure 4.17: Gate count reduction illustration.....	56

Figure 4.18: Direct decimation to targeted output sample frequency without interpolation.....	58
Figure 4.19: Block diagram of designed poly-phase digital band-pass filter.....	59
Figure 4.20: Frequency response of the polyphase band-pass filter.....	60
Figure 4.21: Point interpolation for a typical FIR filter response.....	61
Figure 4.22: Frequency response of the 200-point interpolated filter.....	61
Figure 5.1: Frequency spectrum using sampling rate of ~4 MHz.	64
Figure 5.2: Noise injection model for analog front end stage.	65
Figure 5.3: Differential scheme for analog front end.....	65
Figure 5.4: Frequency spectrum using differential scheme.....	65
Figure 5.5: Frequency spectrum from differential and dither scheme.	66
Figure 5.6: System model of D-ADC with differential and dither scheme.....	67
Figure 6.1: VHDL DUT simulation environment.....	71
Figure 6.2: Sampled data output from two differential single channel operations.	71
Figure 6.3: Frequency spectrum of differential sampled data output.	72
Figure 6.4: Differential implementation scheme process.....	73
Figure 6.5: High speed counters to approximate the location of the impulses with respect to the carrier sine wave.....	74
Figure 6.6: Normalization of the count.	74
Figure 6.7: Block diagram of the band-pass filtering unit	75
Figure 7.1: Block diagram of the design setup on FPGA and veroboard.	80
Figure 7.2: Equipment Setup.	80
Figure 7.3: (a) Photo of the board setup (b) Photo of the electrical components on the veroboard	81
Figure 7.4: (a) 2.5 V – idle tone deviation chart (b) 3.3 V – idle tone deviation chart (c) 5.0 V – idle tone deviation chart.....	83
Figure 7.5: Frequency spectrum for dynamic testing for all 3 voltage level outputs.	84
Figure 7.6: SNR performance of various input amplitude for single tone test.....	85
Figure 7.7: SNR performance for different sampling frequency for single tone test	85
Figure 7.8: Differential non-linearity measurement.	87
Figure 7.9: Integral non-linearity measurement.	87
Figure 7.10: ENOB across various input frequencies.....	88

Figure 7.11: Differential Scheme with dithering setup on FPGA	90
---	----

List of Tables

Table 2-1: Summary of Performance for Voltage-line Delay [4].....	8
Table 2-2: Table of Performance for Statistical Sampling [5]	11
Table 2-3: Table of Performance for Stochastic Sampling [6].....	12
Table 2-4: Table of Performance for Linear Pulse Modulation [10]	16
Table 2-5: Table of Summary for FD-ADCs analysis	21
Table 3-1: Digital Sequence to extract analog input signal.....	26
Table 3-2: Table of values used in the Matlab model.....	28
Table 3-3: Table of values used in the Matlab model.....	35
Table 4-1: Table of values used in the Matlab model.....	53
Table 4-2: Gate count comparison between the 2 approaches	57
Table 4-3: Approximate power savings from the simplified approach.....	57
Table 4-4: Table of specifications for digital FIR band-pass filter.....	59
Table 6-1: 0.18 um technology node information	76
Table 6-2: Table for gate count estimation for a single channel	76
Table 6-3: Table for digital power estimation for a single channel	76
Table 6-4: Table for analog power estimation for a single channel	77
Table 6-5: Table for active area and power comparison	77
Table 7-1: Table for measured signal amplitude and the related quantized step.....	82
Table 7-2: Table of performance for enhanced D-ADC	89
Table 8-1: Table for analog power estimation for a single channel	93

Chapter One

1 Introduction

As complementary metal–oxide–semiconductor (CMOS) technology rapidly progresses into deep-sub micron scales, the average digital design space for digital designers has increased exponentially and cheaply, following Moore’s Law. The technologies of digital integrated circuits have advanced tremendously in speed and size over the past two decades. This has lead to possibilities of highly sophisticated digital signal processing (DSP) systems. These systems operate on a wide variety of digital data from fields of speech, communication, audio, video, medical, power and so on.

And one of the main successes to the use of such sophisticated DSP systems is the advancement in the designs of analog-to-digital converters (ADCs). These converters have played an important role in digitalizing our physical world which is analog in nature. Due to the need to digitalize large number signal types, much ADCs architecture has evolved [1]. In most applications, both analog and digital functions co-exist on the same integrated circuit (IC) chip. It has been a preferred or mandatory requirement that the analog functions are implemented in the same digital CMOS technology so as to avoid additional processing steps, and thus avoiding addition cost of production to the IC chip.

However, the analog integration has gained progress on a much slower scale than its digital counterpart. While digital functions have benefited from technology scaling into deep-sub micron scales, this has been counter effective on analog functions. Analog functions are affected by reduced signal-to-noise ratio (SNR), lower intrinsic gain in CMOS devices, increased device leakage, larger transistor mismatches, and lower quality passives such as inductors and capacitors in lower transistor geometries [2]. Following an ADC survey done in 1999 [3], it has been shown that ADC performance has only improved by 2 bits over a decade for the same sampling speed.

Due to increased challenges of analog integration in deep-sub micron scales, there is a drive to push more and more analog functions into the digital domain. This includes the ADC [4,5,6,8,9,10,11].

1.1 Motivation

An ADC is often a fundamental building block in IC applications as it interfaces the physical analog world to the IC chip that man uses for digital computation, having a full digital ADC allows for fast adaption to new CMOS technologies or integration with other digital CMOS functions, which could be done easily with automation tools. As new CMOS technologies emerge, porting these digital based designs will require minimum effort and time.

The main advantages of having a full digital ADC are therefore summarized as below:

- **Low Cost on Area**

As digital CMOS technology progresses with shrinking transistor geometry, there are large amount of digital area available at low cost.

- **One-time design effort for all future CMOS technologies**

Digital designs are described extensively on register transfer level by VHSIC hardware description language (VHDL) or Verilog, which are technology independent descriptions. The digital functions can be described fully by these hardware languages and verified on digital simulators before they are translated to gates and wires using design automated tools.

With Field Programmable Gate Arrays (FPGA) gaining popularity in pre-silicon system verification, the function of the full digital ADCs can also be synthesized and tested on FPGA before the actual silicon gets fabricated. FPGA caters for functional checks of such type of ADCs even before silicon is available.

The lowering of supply voltages for newer CMOS technologies will also have limited impact on the digital based ADCs if no custom cells are used. The migration of the fully digital ADC to newer CMOS technologies can be zero effort on re-design. This will allow for one-time design effort for all future CMOS technologies. This results in a low cost and reuse-able one-time development for a full digital ADC.

- **Technology implementation automation and fast integration**

With the ease of usage and availability of design automated tools, digital design implementation has become almost fully automated with high reliability and guaranteed first-time right silicon.

This in turn results in lower cost in the design and development process and avoids expensive re-spins due to silicon variation issues.

As ADCs are widely used in most IC applications, being the interface between the physical world and silicon, a full digital ADC will allow automated and fast integration with other digital functions on the same silicon.

Furthermore, simulations are done purely on digital simulators which are time and computing efficient, and allow for long real world run time. Free digital simulators are also widely available. This avoids the need for expensive design automated tool licenses.

- **Resistant to environmental noise and process variation**

In many applications, it is preferred that the entire circuit is realized in the digital domain because digital circuitries are more resistant to environmental noise and process variation.

Thus the performance of the design does not vary too much from simulation and the real silicon.

However, there are many challenges to achieve a fully digital ADC. In general, some form of analog elements is still necessary to be in a full digital ADC in order to receive signal from the physical world, which is analog in nature. This is inevitable. One of the main challenge is to keep the amount of analog elements low, and the other is that the performance of the analog elements that are used have to be unaffected by silicon deviation or low power supplies.

1.2 Contributions

In this thesis, Christoph Braun's approach [10] to Full Digital ADC design is being selected for study and improvement in areas of power consumptions and bandwidth. Christoph's approach is adopted because of its silicon proven measurement with a good non-linearity of +/- 0.024% Full Scale (FS) performance result which is much better than other surveyed digital type ADC designs. The improvement in power consumption is necessary in order to make the

architecture attractive for use on silicon in various platform applications such as mobile platforms where power consumption has to be low. The improvement in bandwidth will allow this digital type ADC to handle the large number of different types of signals in various platform applications.

Power consumption in digital circuitries is mainly governed by the amount of digital circuitries that is presented, and also the operating frequency of these circuitries. In this thesis, the approach to reduce power consumption in this digital type ADC is by digital gate count reduction, and lowering of the operating frequency of these digital gates. The digital arithmetic computation involved in the digital band-pass filtering and the removal of the carrier frequency are proposed for digital gate count reduction. The circuitries in analog front end and the digital band-pass filter are proposed for reduction in operating frequency. The reduction in operating frequency of the analog front end circuitries will pave the way for higher bandwidth in the ADC, as the reduction in operating frequency will allow a higher operating frequency to be reused in the AFE when a higher bandwidth is desired.

In summary, the thesis contributes by making the following proposals to the implementation of the digital type ADC.

- A multiplierless approach is proposed for the removal of the carrier frequency in the ADC and the recovery of the analog input frequency components. In this approach, the band-pass type frequency spectrum from the analog front end is down-converted to baseband spectrum with the use of frequency translation by decimation. By creating a relationship between the down-converted carrier frequency, the intermediated sampled frequency and the targeted output sampled rate of the converted data, a multiplierless removal of the carrier frequency can be done. In addition, the analog input frequency components can be recovered without the need for any low-pass filtering circuitries. This help to save silicon area and power.
- The original ADC design implements a band-pass filter with over 2 million coefficients. This thesis proposes to reduce the band-pass filter complexity with the following 3 approaches. Interpolation of the data delivered by the analog front end is not allowed in order to keep the operating frequency of the filter to the minimum, thus avoiding the need for large amount of coefficients. The polyphase architecture is then used to efficiently implement the filter avoiding unnecessary arithmetic computation. Coefficient silicon

space is reduced by using interpolation to calculate the missing required coefficients. This means that only some part of the coefficients is implemented on the silicon. With these 3 approaches, the digital gate count and the operating frequency of the band-pass filter can be kept low.

- A differential dithering scheme is proposed to reduce the noise in the analog front end circuitries when the sampling rate of the zero-crossings is reduced.

Research findings reported in this thesis have been published or are being submitted for consideration for publication in the following papers:

- R.W.T. Ng and L. Siek, “A Simplified Approach for Baseband Recovery in SDR Architectures”, presented and published at *IEEE Asia Pacific Conference on Circuits and Systems*, pp. 1059 – 1062, December 2010.
- R.W.T. Ng and L. Siek, “Low Power Digital Type ADC”, accepted at *13th International Symposium on Integrated Circuits*, December 2011.

1.3 Thesis Outline

Chapter 1 gives an introduction to strengths of digital circuitries and the potent benefits of a full digital type ADC. The contribution of the thesis is also given in this chapter.

In chapter 2, a literature research is done on the various proposed architectures to a full digital type ADC. This chapter discusses about the available digital type ADCs and their performances. An analysis of these ADCs in terms of their ability to withstand varying supply voltages and silicon variation is provided.

In chapter 3, the workings of Christoph Braun’s approach to a digital type ADC is studied and presented. The design is represented with Matlab models.

In chapter 4 and 5, methods to reduce power consumption in Christoph Braun’s ADC is proposed in details, and the enhancements are proven with Matlab simulations.

In chapter 6 and 7, the enhanced ADC implemented with VHDL and verified with a digital simulator is presented. The estimated gate count and power is shown. The experimental setup for the enhanced ADC verification with the use of a FPGA platform is also explained and shown.

Chapter 8 concludes this research work with recommendations for future work.

Chapter Two

2 Full Digital Analog-to-Digital Converters

Full digital (FD) analog-to-digital converters (ADCs) convert analog signal to digital data using only digital circuitries such as digital registers, AND, and OR gates. However, it is inevitable to use some of analog elements or customized digital logic cells to process the analog signal delivered from the physical world. This is true for all the FD ADCs proposed by all the researchers. Overall, the motivation to adopt full digital architecture for ADCs should be that this will make ADCs tolerant to changes to linearity and bandwidth due to decreased supply voltages and silicon variation. Decreased supply voltages and higher silicon variation are two phenomena of downscaled CMOS technologies.

This chapter discusses about the available FD ADCs proposals and their performances. This chapter introduces the several FD ADC architectures proposed by different researchers and their evaluated results. An analysis of these FD ADCs in terms of their ability to withstand varying supply voltages and silicon variation is provided. Power consideration is also included in the analysis.

2.1 Full Digital ADCs Architectures

A total of 4 full digital ADC architectures are discussed in this section. First, the voltage line to delay conversion architecture is introduced, followed by statistical sampling, stochastic logic sampling and linear pulse modulation with constant impulse amplitude respectively. Performance, in terms of gate count requirement of each architecture, is also tabled in each subsection.

2.1.1 Voltage Line to Delay Conversion

The first approach is based on a voltage line to delay conversion as shown in Fig 2.1.

Figure 2.1: Voltage-line delay conversion.

This design approach is originally proposed by T. Watanabe [4], which exploits the relationship between electric current and delay in digital CMOS buffer that is made up of two CMOS inverters forming a delay element (DE). The bigger the voltage and hence its current, the smaller will be the delay in the DE, thus decreasing the time a pulse will take to propagate from its input to output. These digital CMOS buffers that are modified with current control are chained together to form a delay chain with feedback. The bigger the input current, the smaller is the time taken for a pulse to travel from one end of the DE based delay chain to another and feedback to its input again. Fig 2.2 shows the delay chain architecture. The output from this chain is used to clock digital counters whereby the values recorded will be a representation of the voltage level of the analog input signal in a pre-determined look-up table. In this way, the analog signal to digital data conversion is done completely with digital elements with customized digital CMOS buffer.

The design is realized in $0.65\ \mu\text{m}$ CMOS process with a silicon active area of $0.29\ \text{mm}^2$ [4]. It gives a non-linearity of $\pm 1\%$ in Full Scale (FS) of 200 mV. Extending the scale to 3.0 V gives a non-linearity of $\pm 5\%$. The table of performance is summarized in Table 2-1.

Figure 2.2: Delay chain architecture.

Table 2-1: Summary of Performance for Voltage-line Delay [4]

Evaluated Parameters	Results
Silicon Evaluated	Yes
Presence of Analog Elements	No, but required customized digital buffers
Non-linearity	+/- 1 % Full Scale; +/- 5 % Full Scale
Voltage Input range	0 - 200 mV; 0 - 3.00 V
Temperature Drift	Large
Gate Count	2000 gates
Power consumption	< 1 mW
Further data processing needed	Yes

2.1.2 Statistical Sampling

The second approach is based on statistical sampling in which a randomly generated voltage reference (Gaussian distributed) is compared with the analog input signal [5]. This will result in a stream bit with a certain Probability Density Function (PDF). The comparator used could be a low cost 1-bit analog comparator. The multi-level randomly generated voltage reference can be achieved with an analog reference built with a resistor-capacitor (RC) charge-discharge circuit, and sampled with random instant using linear feedback shift register (LFSR) timed circuit. This

comparison will result in a shift in the Probability Density Function (PDF) of a Gaussian distribution, which will translate to a certain voltage level in a pre-determined look-up table. Overall architecture is shown in Fig 2.3 (extracted from figure 3 of [5]). The PDF shift concept is shown in Fig 2.4.

Figure 2.3: Statistical sampling architecture.

Figure 2.4: Probability Density Function shift concept.

Fig 2.4(a) shows the model of a comparator in which the output of the comparator (v_{out}) switches to the digital high voltage if the analog input signal (v_{in}) is of a higher voltage than the random voltage reference (ref_in), and to a digital low if the opposite occurs. Fig 2.4(b) shows the convolution of the PDF of the analog input signal ($p\{v_{in}\}$) with the PDF of the random voltage reference ($p\{ref_in\}$). This results in a shift of the PDF of the random voltage reference by a certain value, V . By taking a certain number of samples, one can calculate the mean of V by doing an average. This will translate to the voltage level of the analog input signal in a pre-determined look-up table.

The design is evaluated on a FPGA prototype board with 4 LM311 comparators, 4 random analog references created by low-passing signals coming from signal generator outputs [5]. The non-linearity result is not published, but it is mentioned that linearity occurs only in a voltage range of 400 mV, and is highly dependent on noise distribution and could widely vary. The table of performance is summarized in Table 2-2.

Table 2-2: Table of Performance for Statistical Sampling [5]

Evaluated Parameters	Results
Silicon Evaluated	No. FPGA only
Presence of Analog Elements	Yes; Need for 1-bit comparators and analog reference
Non-linearity	Data not published
Voltage Input range	0 - 400mV
Temperature Drift	Possibly large
Gate Count	3000 gates (excluding comparators)
Power consumption	Data not published
Further data processing needed	Yes

2.1.3 Stochastic Logic Sampling

The principle of the stochastic ADC is first proposed by J.G Ortega [6] based on stochastic logic theory proposed by B.R Gaine [7]. Realization of such ADCs on programmable logic device (PLD) or FPGA systems are shown to be viable [8]. The stochastic ADC design replaces its digital-to-analog (DAC) digitally in a successive approximation type ADC. The DAC is replaced by a stochastic based pulse generator whose pulse sequences depend on a digital input from the successive approximation registers. The mean value of these pulse sequences is obtained by doing a low-pass filtering of these pulses. This leads to a full digital ADC with only a simple analog type low-pass filter and a single-bit comparator required externally. The constructed stochastic ADC is shown in Fig 2.5.

The design is evaluated on a Programmable Logic Device (PLD) board with 1 LM393 comparator, a resistor and capacitor acting as a low-pass filter [8]. The non-linearity result is not neither published nor mentioned. It is observable that the voltage input range is large at more

than 3.0 V. This might improve the SNR of the design. One major observation is that the conversion time required is large achieving only 1.62 KSamples per second. The table of performance is summarized in Table 2-3.

Figure 2.5: Stochastic ADC design.

Table 2-3: Table of Performance for Stochastic Sampling [6]

Evaluated Parameters	Results
Silicon Evaluated	No. PLD only
Presence of Analog Elements	Yes. Need for 1-bit comparators and passive resistor-capacitor filter
Non-linearity	Data not available
Voltage Input range	0 - 3.0 V
Temperature Drift	Possibly Large due to resistor-capacitor filter
Gate Count	Data not published
Power consumption	Data not published
Further data processing needed	No

2.1.4 Linear pulse modulation with constant impulse amplitude

The third approach is the digital ADC (DAD) architecture [10] originally proposed by Christoph Braun. The overall DAD architecture is shown in Fig 2.6.

Figure 2.6: Overall DAD architecture.

The design is based on a linear pulse modulation with constant impulse amplitude. The analog input signal is sampled with a sine wave carrier which is generated by a digital ROM. The crossing of these two sine waves leads to the generation of dirac pulses at specific time locations, which in the idealized case will have a frequency modulated function that gives a relationship between the input frequency and the carrier frequency. The overall architecture consists of mostly digital CMOS gates and simple RC network. The DAD analog front end sampling architecture is shown in Fig 2.7. The 1st order frequency spectrum of the time location of the dirac pulses is shown in Fig 2.8. The 1st order of the frequency spectrum of the dirac pulses is typically a band-pass spectrum with a bandwidth of 2 times the highest input signal that is being sampled. f_c denotes the carrier frequency, while f_a denotes is the input analog frequency.

Figure 2.7: DAD front end sampling architecture.

Figure 2.8: 1st order frequency spectrum of the time location of the dirac pulses.

The time stamps that locate the dirac pulses are recorded by high speed digital counters. These values are then passed on to a digital band-pass FIR filter for filtering, followed by baseband down-conversion through decimation, convolution and digital low-pass filtering in order to recover the original analog input frequency. The quantized data processing sequence is illustrated in Fig 2.9. It involves extracting the 1st order band-pass type frequency spectrum using a digital band-pass filter, followed by a baseband down-conversion of the extracted spectrum down by digital mixing with the carrier frequency. A digital low-pass filter is used to remove 2nd images formed by the digital mixing. This results in the input analog frequency component being correctly extracted.

The design is realized in $0.18\ \mu\text{m}$ CMOS process with a silicon active area of $1.05\ \text{mm}^2$ [10]. It gives a non-linearity of less than $\pm 0.024\%$ in full scale of $1.8\ \text{V}$. Input bandwidth is allowed at $0 - 20\ \text{kHz}$. The table of performance is summarized in Table 2-4.

Figure 2.9: DAD digital signal processing basics.

Table 2-4: Table of Performance for Linear Pulse Modulation [10]

Evaluated Parameters	Results
Silicon Evaluated	Yes
Presence of Analog Elements	Yes. Passive resistors and capacitors required
Non-linearity	+/- 0.024% Full Scale
Voltage Input range	0 - 1.8 V. Limited by amplitude of sine carrier
Temperature Drift	Possibly Small
Gate Count	> 100 000 gates
Power consumption	< 7.2 mW
Further data processing needed	No

2.2 Analysis of Full Digital ADCs Architectures

The four approaches to full digital (FD) ADCs are analyzed here in terms of their ability to withstand

- Low supply voltages,
- How linearity is affected by low supply voltage,
- Bandwidth performance improvement, and
- Power considerations

In general, it is due to the presence of analog elements that affects an ADC design's ability to withstand these three factors over different CMOS technologies. All the four approaches required some form of analog element to process the incoming analog input signal. The first uses a delay element; the second uses an analog reference, and a comparator; the third uses an analog type low-pass filter and a comparator; while the fourth uses resistors and capacitors. The use of analog elements is inevitable as the incoming signal is analog in nature and some form of analog conditioning element is required.

The presence of analog elements in ADCs affects how dynamic the supply voltages can range. The silicon characteristics of these analog elements do not change proportional with supply voltages at times.

The delay element (DE) in the first approach consists of modified digital CMOS inverters. The supply voltage of these inverters is the analog input signal. In order for an inverter to work correctly, a supply voltage of two times the voltage threshold (V_t) of the transistor used is required. This in turn will limit the lower limit of the input signal range of the analog signal. The second approach requires an analog random generator reference and a comparator, which is also susceptible to supply voltage changes with respect to its performance. The third one requires a good choice of resistor and capacitor value in order to achieve accurate voltage reference for the comparator. This selection of value might not be easy in order to accommodate varying supply voltages. The fourth one constructs a sine wave carrier that is generated digitally, smoothed with resistors and capacitors. One will expect supply voltage to affect the sine wave carrier generated due to presence of passive analog elements, and possibly mis-matched passive resistors. However, the originator mentioned that this issue is solved by generating the sine wave carrier with third order noise shaping [12]. The analog input range is also reduced as one move to lower supply CMOS technologies, as the amplitude of the sine wave carrier will decrease with lower supply. However, this phenomenon will be true for all ADC designs. As only the fourth design approach's performance is not truly influenced by low supply voltages, it is one of the few suitable for porting to newer CMOS technologies with low supply voltages.

The presence of analog elements in ADCs affects the linearity of ADCs because the analog elements' properties change with different CMOS technology process and inaccuracies in the manufacturing process.

In general, the delay in a digital CMOS inverter is not linear with respect to its supply voltage in different CMOS technologies. A typical delay relation with the supply voltage of a DE [11] is shown in Fig 2.10 (extracted from Fig 3 of [11]). One could observe from the plot that the delay of a delay element changes non-linearly with the supply voltage to it.

Figure 2.10: Delay Vs the supply voltage of a DE.

Due to a non-linear relation with delay and supply voltage, the design approach of the ADC will have limited linearity over a large input analog signal range. However, there has been a new proposed architecture for the DE in order to improve the linearity [11]. Digital compensation is also possible. The second approach fared not too better in linearity than the first approach. The implementation linearity result of both the first and second approach [5] is shown in Fig 2.11 (extracted from Fig 5 of [5]). The delay line (DL) ADC plot shows the linearity of the first approach over a certain voltage input range with respect to its measured output voltage, while the statistical sampling (SS) ADC shows the linearity of the second approach over the same range. It is mentioned that the linearity of the SS-ADC is highly dependent on noise distribution and could widely vary.

Figure 2.11: Linearity result implementation from first and second approach.

The third approach has its linearity affected by the choices of the resistor and capacitor for the analog type low-pass filter. And there is no systemic way to obtain such optimal choices.

The fourth approach is not affected in linearity by the precision of its analog components. The important factor is to get an accurate sine wave carrier in its frequency to be super-imposed onto the analog input signal. This is achieved by a noise-shaped digitally generated sine wave which allows a large tolerance of imprecision in the resistors and capacitors. An accurate extraction of a single tone of 4 kHz signal by the approach is published by the originator [10]. The THD+SNR performance of this ADC design is 73 dB.

Bandwidth of the FD-ADCs is of concern also with these four FD-ADCs approach. In the first approach, the limitation here is the need to have the analog input signal stable at a point in time for the measurement of the delay to be done. If one can do a very fast and accurate measurement of the input signal, high bandwidth will be achievable. The second approach generally requires a large number of samples in order to be reasonably accurate in measuring the average. These might required many times the sampling rate of a normal ADC. However, this large amount of sample collection could be done with parallelism. This will be limited by the technology process switching speed or silicon area available. The third approach is limited by the conversion time of the analog type low-pass filter. If a large value of RC is used, it will impact the conversion time greatly. The fourth approach requires a sine wave carrier frequency which is three times the frequency of the analog input signal. One will expect that the generation of the sine wave carrier will require an operation clock which is a few hundred times of the carrier frequency. On top of this, a high frequency sampling clock is required to determine the time stamps of the signal crossings. This will be limited by the technology process switching speed.

In general, FD-ADCs are associated with high gate count and potentially high operating frequency, thus power consumption of the FD-ADCs is also of a concern with these four FD-ADCs approach. The first approach uses a small number of digital circuits which includes a frequency counter, latch and encoder, in addition to the delay elements. This setup is measured to consume less than 1 mW of power in its conversion operations. However, a practical usage for this approach as a digital sensor requires additional of a correction digital circuit [4]. This could potentially increase the power consumption in real practical usage depending on the resolution and voltage input range as sampling frequency could need to be increased, or additional correction circuits have to be added. The second approach requires a randomly generated

reference, a single bit comparator in this analog setup, and digital circuits for averaging and correction. The resolution is dependent on the number of parallel analog comparators used, and the depth of the averaging. Thus a need of higher resolution and large voltage input range could potential increase the need for more parallel comparators or larger depth in averaging. The third approach using stochastic logic sampling seems to be consistent in its gate count usage, which result in predictable power consumption in its applications. The only need is to improve its sampling or operating frequency in order to lower its long conversion time. The fourth approach is also consistent in its gate count usage. However, there is a need for a high sampling frequency in detecting the dirac pulses in its analog setup. There is also a need for quite some amount of digital circuits to perform the baseband down-conversion of its frequency spectrum. In an audio application, the power consumption used is 7.2 mW [10].

2.3 Summary and Objective

In this chapter, 4 different full digital ADC are analyzed with respect to their ability to withstand low supply, and how linearity will be affected by low supply and their potential to improve bandwidth. Power consumption is also considered in the analysis. A table of summary for the FD-ADCs analysis is given in Table 2-5.

Table 2-5: Table of Summary for FD-ADCs analysis

Evaluated Parameters	Voltage Line	Statistical	Stochastic	Linear Pulse
Silicon Evaluated	Yes	No.	No.	Yes
Presence of Analog Elements	No.	Yes	Yes.	Yes.
Non-linearity	+/- 1 % Full Scale; +/- 5 % Full Scale	Data not published	Data not available	+/- 0.024% Full Scale
Voltage Input range	0 - 200 mV; 0 - 3.00 V	0 - 400mV	0 - 3.0 V	0 - 1.8 V.
Temperature Drift	Large	Possibly large	Possibly	Possibly Small
Gate Count	2000 gates	3000 gates (excluding comparators)	Data not published	> 100 000 gates
Power consumption	< 1 mW	Data not published	Data not published	< 7.2 mW
Further data processing needed	Yes	Yes	No	No

In general, all the 4 approaches to FD-ADCs use simple or minimum number of analog components as their analog setup. The first 3 approaches have already inherent challenges to improve their linearity in a single CMOS technology. Voltage Delay-line ADC has a non-linearity of $\pm 1\%$ FS, while statistical and stochastic ADC has a non-linearity highly dependent on noise distribution which is not easy to predict. However, linear pulse modulation ADC has emerged as a good potential for further studies and improvement as it has a good non-linearity of $\pm 0.024\%$ FS performance result over the other 3 ADC designs.

With its relatively good non-linearity performance over the other 3 ADC designs, linear pulse modulation ADC comes with an expense of high gate count, and power consumption. In order that this ADC design could be more desired, the gate count and power consumption should be reduced further. It will also be advantageous to further improve the bandwidth of the design.

Thus the objective of this research work is to further improve the linear pulse modulation ADC design in terms of:

- Power consumption
- Bandwidth

This digital ADC design will be referred as D-ADC throughout the rest of the thesis.

Chapter Three

3 Overview and Modeling of D-ADC

In this chapter, the workings of the D-ADC is studied and presented. The analog part of the D-ADC is being referred as the analog front end part, while the digital part consists of digital filtering. This chapter first gives an introduction of the analog front end fundamentals, followed by a discussion about the digital filtering part of the D-ADC design. The D-ADC is then analyzed from a top-down approach in a system-level Matlab model, after which the analog front end and the digital filtering is modeled separately. The Matlab models are used to study the workings of the D-ADC, and to explore methods to improve power consumption and bandwidth. To begin with, the overall D-ADC architecture is re-illustrated in Fig 3.1 below.

Figure 3.1: Overall D-ADC architecture.

3.1 Analog Front End

The D-ADC analog front end (AFE) operating fundamentals are first analyzed using equation 3.1. It involves superimposing the analog input signal with a self-generated sine wave, and the crossing of these two signals leads to the generation of dirac pulses at specific time

locations. This time location holds the key information of the amplitude and frequency of the analog input signal.

$$S_t(t) = S_a(t) + S_m(t) \quad (3.1)$$

where:

$S_m(t)$ = injected carrier freq,

$S_a(t)$ = analog input signal,

$S_t(t)$ = resultant signal of superposition of the injected and analog input signal.

And magnitude of $S_a(t)$ has to be less than $S_m(t)$. Let $S_m(t) = A.\cos(2\pi f_0 t)$ and $S_a(t) = B.\cos(2\pi f_1 t)$ which is typically representations of analog sine signals. In order to simplify equation 3.1, $A = 1$, and $B = 1$ is used. From equation 3.1, one will have:

$$S_t(t) = \cos(2\pi f_1 t) + \cos(2\pi f_0 t) \quad (3.2)$$

The time location of the crossings is given as:

$$\cos(2\pi f_1 t) + \cos(2\pi f_0 t) = 0 \quad (3.3)$$

Solving equation 3.3, one will have:

$$t = \frac{n}{f_0 + f_1} \text{ or } t = \frac{n}{f_0 - f_1} \quad \text{where } -\infty < n < +\infty \quad (3.4)$$

The dirac pulses will then be given in the time domain as:

$$g(t) = \sum_{-\infty}^{+\infty} \delta\left(t - \frac{n}{f_0 + f_1}\right) + \sum_{-\infty}^{+\infty} \delta\left(t - \frac{n}{f_0 - f_1}\right) \quad (3.5)$$

Doing a fourier transformation of equation (3.5), one obtains equation (3.6):

$$g(f) = (f_0 + f_1) \cdot \sum_{-\infty}^{+\infty} \delta(t - n(f_0 + f_1)) + (f_0 - f_1) \cdot \sum_{-\infty}^{+\infty} \delta(t - n(f_0 - f_1)) \quad (3.6)$$

The 1st order frequency response spectrum plot where $n=1$ is given in Fig 2.8. The figure is repeated here in Fig 3.2.

Figure 3.2: 1st order frequency spectrum of the time location of the dirac pulses.

The detection of time location of the impulses in present scheme is to use high speed counters to approximate the location of the impulses with respect to the carrier sine wave. This is illustrated in Fig 3.3. The counter values collected is then forwarded and processed by digital filters to extract the frequency and amplitude of the analog input signal.

Figure 3.3: High speed counters to approximate the location of the impulses with respect to the carrier sine wave.

3.2 Digital Filtering

From equation 3.6, it is known that the frequency spectrum of the dirac impulses generated from the AFE is a series of band-pass components. By performing the digital filtering sequence order as in Table 3-1, the analog input signal inclusive of its magnitude and frequency can be extracted. The below sequence is previously illustrated in Fig 2.9. The figure is repeated here in Fig 3.4.

Table 3-1: Digital Sequence to extract analog input signal

<i>Step Sequence</i>	<i>To-do</i>
1	Digital Band-pass filtering to extract the 1 st order of the band-pass component
2	Down-sample to reach the lower sampling rate
3	Digital mix to remove carrier freq
4	Digital Low-pass filtering to remove 2 nd order frequency components
5	Down-sample to reach the desired sampling rate

Figure 3.4: D-ADC digital signal processing basics.

3.3 Matlab Modeling of D-ADC

Based on the basics outlined in Section 3.1 and 3.2, the D-ADC design concept is simulated in Matlab to prove its viability. In this section, the system, analog front end and digital filtering sequence modeling of D-ADC in Matlab is presented.

3.3.1 System Modeling of D-ADC

In this sub-section, the system modeling of D-ADC is conducted with Matlab. The full D-ADC is represented with Matlab components. The injected carrier frequency, $S_m(t)$ is set to be $1.\cos(2\pi \times 66.15 \times 10^3 \times t)$, while the analog input signal, $S_a(t)$ is set to be $0.9.\cos(2\pi \times X \times 10^3 \times t)$. $S_m(t)$ is set to be of a frequency of 66.15 kHz as it has to be 3 times greater than the frequency of the maximum input analog signal frequency [10]. Parameter, X is first chosen to be at a low 500 Hz to test the lower frequency range of the design, followed by a high 20 kHz to test the higher frequency range. The values chosen for parameter, X is suited for

audio applications. The sampling rate of the zero-crossings is set at $10.8486 \times 10^6 \text{ Hz}$. This sampling rate is chosen as it is a multiple of 44.1 kHz , which is the targeted output sampling rate. The zero-crossing detection block outputs a dirac pulse whenever a crossing is detected. This sampling rate is much lower than the required 1 GHz rate. This is due to the memory limits of the Matlab Simulink tool that the design could not be sampled at such a high rate. This limit is overcome by using the Matlab scripts, avoiding the use of the Simulink tool. The workings of the Matlab script are demonstrated in Section 3.3.3. The digital band-pass filter pass-band edge is set at $66.15 \times 10^3 \text{ Hz} \pm 20 \times 10^3 \text{ Hz}$, and its stop-band edge at $0 - 37 \times 10^3 \text{ Hz}$; $97 \times 10^3 \text{ Hz} - 10.8486 \times 10^6 \text{ Hz}$. Sampling rate of the band-pass filter is set to be the same as the zero-crossing sampling rate at $10.8486 \times 10^6 \text{ Hz}$. This is necessary as the dirac pulses from the zero-crossing detection block is processed directly by the band-pass filter. Last, but not least a digital low filter with a pass-band edge of $0 - 22 \times 10^3 \text{ Hz}$, and stop-band edge of $30 \times 10^3 \text{ Hz} - 264.6 \times 10^3 \text{ Hz}$ is used to remove the 2nd frequency components formed in the digital mixing stage where the carrier frequency is removed. The parameters chosen in the Matlab modeling are summarized in Table 3-2. The coefficients for the digital filters are generated using the Filter Design and Analysis tool (FDATool) in Matlab.

Table 3-2: Table of values used in the Matlab model

<i>Parameters</i>	<i>Values</i>
$S_m(t)$	$1.\cos(2\pi \times 66.15 \times 10^3 \times t)$
$S_a(t)$	$0.9.\cos(2\pi \times X \times 10^3 \times t)$ where value of X can be $0 < X < 20 \times 10^3 \text{ Hz}$
Crossing sample rate	$10.8486 \times 10^6 \text{ Hz}$
Digital band-pass FIR filter specifications	Pass-Band: $66.15 \times 10^3 \text{ Hz} \pm 20 \times 10^3 \text{ Hz}$ Stop-Band: $0 - 37 \times 10^3 \text{ Hz}$; $97 \times 10^3 \text{ Hz} - 10.8486 \times 10^6 \text{ Hz}$ Sampling rate: $10.8486 \times 10^6 \text{ Hz}$
Digital low-pass FIR filter specifications	Pass-Band: $0 - 22 \times 10^3 \text{ Hz}$ Stop-Band: $30 \times 10^3 \text{ Hz} - 264.6 \times 10^3 \text{ Hz}$ Sampling rate: $264.6 \times 10^3 \text{ Hz}$
Output sample frequency at audio application range	$44.1 \times 10^3 \text{ Hz}$

Fig 3.5 illustrates the Matlab building blocks for the whole D-ADC design. It consists of a total of 4 stages. The 1st stage is called the analog front end processing stage that superimposes the carrier frequency, $S_m(t)$ onto the incoming analog input signal, $S_a(t)$. The zero crossings of the resultant signal are sampled and the dirac pulses generated are passed onto the 2nd stage which holds the digital FIR band-pass filter. After band-pass filtering, the digital data is passed on to the 3rd stage where the carrier frequency is removed. The 4th stage involves low-pass filtering and down-sampling to remove the unwanted 2nd order frequency components and to obtain the final quantized data.

Figure 3.5: Matlab system modeling of D-ADC.

The extracted 500 Hz and $20 \times 10^3\text{ Hz}$ analog input frequency is shown in Fig 3.6(a) and Fig 3.6(b) respectively. Due to the low sampling rate used for zero crossing detection in Matlab Simulink tool, the quantization noise floor is simulated to be approximate -50 dB . However, by representing Stage 2 to Stage 4 in Matlab script, the full Matlab model at a high zero-crossing sampling rate of $1.08486 \times 10^9\text{ Hz}$ can be simulated. The coefficients for the high sampling rate

digital filters are also generated using FDATool in Matlab. The quantization noise floor is simulated to be approximate -90 dB. An $15 \times 10^3 \text{ Hz}$ analog input signal is used as the test signal. The frequency extracted by high zero-crossing sampling rate at $1.08486 \times 10^9 \text{ Hz}$ is shown in Fig 3.7. This proves that by having a high sampling rate of greater than 1 GHz , a high resolution converted data is obtainable by the D-ADC.

Figure 3.6: Frequency spectrum of sampled analog input freq with low sampling zero crossing detection freq.

Figure 3.7: Frequency spectrum of sampled analog input freq with high sampling zero crossing detection freq.

3.3.2 Modeling of Analog Front End

The injected sine wave carrier is generated by a digital Read-Only Memory (ROM), and superimposed on the analog input signal illustrated in Fig 3.8. The resultant signal is smoothed by resistors and capacitors. The dirac pulse locations are represented by the rising and falling edges of the generated signal from the CMOS buffers.

Due to the use of the analog passive elements, in this case the resistors, the digital sine contents are noise-shaped. This will enable a good tolerance of the mismatch of resistors on silicon [10][12]. The generation of such noise-shaped digital sine contents is not modeled in Matlab. An attempt to simulate the zero-crossing detection by the AFE in Matlab is done and the block diagram used in Matlab is shown in Fig 3.9. The simulated signals are plotted on Fig 3.10. Fig 3.10(a) shows the plot of the noise-shaped digital sine ROM data that is generated out of the digital sine ROM. This data is forwarded to a set of smoothing capacitors which are represented by zeros and poles function to generate the analog sine carrier wave. The analog sine carrier wave is shown in Fig 3.10(b). The analog sine carrier wave is then superimposed on the analog

input signal and a modulated sine wave is generated. The zero-crossing of this modulated wave generates rising and falling edges which represents the dirac pulses. The zero-crossings from the modulated sine wave are illustrated in Fig 3.10(c).

Figure 3.8: D-ADC Analog Front End.

Figure 3.9: Matlab model of D-ADC Analog Front End.

(a)

(b)

(c)

Figure 3.10: (a) Noise-shaped digital sine ROM data (b) Analog sine carrier wave (c) Zero-crossings from the modulated sine wave.

3.3.3 Modeling of Digital Processing Unit

The time location of the dirac pulses generated from the zero-crossing detection block is to be collected and filtered according to the sequence stated in Section 3.2 with the specifications listed below. However, due to Matlab Simulink limitations, high sampling rate for the detection of the zero-crossings is not allowed. Instead the AFE model is replaced by stage 1 illustrated in Fig 3.9 and the zero-crossings are collected and filtered accordingly. The digital filtering sequence is done using a Matlab script. A high sampling rate of greater than 1.0 GHz is used for digital band-pass filter as to demonstrate that a low noise floor can be reached if a high sampling frequency for detecting the impulses is used. The parameters chosen in the Matlab modeling are summarized in Table 3-3. These parameters are similar to the ones used in section 3.3.1, except that the sampling rate of the zero-crossings is set at $1.0848600 \times 10^9\text{ Hz}$, instead of $10.8486 \times 10^6\text{ Hz}$. And the sampling rate of the digital band-pass filter needs to be matched with that of the zero-crossings, which is $1.0848600 \times 10^9\text{ Hz}$. The coefficients for the digital filters are generated using FDATool in Matlab. The frequency spectrum showing the extracted $15 \times 10^3\text{ Hz}$ analog input frequency is shown in Fig 3.7. This figure is repeated in Fig 3.11.

Table 3-3: Table of values used in the Matlab model

<i>Parameters</i>	<i>Values</i>
$S_m(t)$	$1.\cos(2\pi \times 66.15 \times 10^3 \times t)$
$S_a(t)$	$0.9.\cos(2\pi \times X \times 10^3 \times t)$ where value of X can be $0 < X < 20 \times 10^3\text{ Hz}$
Crossing sample rate	$1.0848600 \times 10^9\text{ Hz}$
Digital band-pass FIR filter specifications	Pass-Band: $66.15 \times 10^3\text{ Hz} \pm 20 \times 10^3\text{ Hz}$ Stop-Band: $0 - 37 \times 10^3\text{ Hz}$; $97 \times 10^3\text{ Hz} - 1.08486 \times 10^9\text{ Hz}$ Sampling rate: $1.08486 \times 10^9\text{ Hz}$
Digital low-pass FIR filter specifications	Pass-Band: $0 - 22 \times 10^3\text{ Hz}$ Stop-Band: $30 \times 10^3\text{ Hz} - 264.6 \times 10^3\text{ Hz}$ Sampling rate: $264.6 \times 10^3\text{ Hz}$
Output sample frequency at audio application range	$44.1 \times 10^3\text{ Hz}$

Figure 3.11: Frequency spectrum of sampled analog input freq with high sampling zero crossing detection freq.

3.4 Conclusion

In this chapter, the AFE and the digital processing unit of the D-ADC is modeled in Matlab and simulated. Due to the limitation of Matlab Simulink, high sampling rate for detection of the zero-crossings in the AFE is not allowed, and to work around this limitation, the digital processing unit is represented in Matlab scripts. The simulated results showed that for a sampling frequency greater than 1 GHz in the AFE, the achieved quantization noise floor is around -90 dB.

Chapter Four

4 Power Reduction Proposal for Digital Unit in D-ADC

D-ADC has a good non-linearity performance of $\pm 0.024\%$ at full scale of 1.8 V. However, this good performance comes at an expense of high gate count of more than 100,000 gates, not including the analog front end components, and having a power consumption of 7.2 mW for a differential channel ADC operation [10]. The lowest published power consumption of a similar ADC implemented on the same CMOS technology process of 0.18 μm is 90 μW [13]. Thus one of the objectives of the thesis is to propose methods to reduce the power consumption in the D-ADC in order to achieve a low-power digital type ADC. As most of the D-ADC comprises of digital logic, the optimal solution to achieve a low-power digital type ADC is by reducing the amount of digital logic used by design, and also reduce the toggling rate of the digital logic.

In this chapter, power reduction for D-ADC by digital logic and operating frequency reduction will be presented. This chapter first review fundamentals of digital signal processing methods such as interpolation, decimation and frequency translation effect of decimation. This is necessary as these fundamentals are extensively used in the D-ADC digital logic processing unit. Second, the polyphase decomposition technique is discussed and explained. Following the review of the fundamentals of the digital signal processing methods, a multiplier-less baseband down-conversion with removal of the carrier frequency is presented. In this scheme, the carrier frequency is removed by digital mixing with a simplified sequence of 0, 1, 0, -1 which is a sine representation at quarter of its sample rate. And further allowing a decimation factor of 2 to occur, the signal components will alias onto one another, avoiding the need for low-pass filtering to remove any 2nd order frequency components formed in the digital mixing sequence. This reduces the amount of digital logic needed in the D-ADC. Next, the polyphase

decomposition technique is proposed to be adopted in the D-ADC in order to reduce the operating frequency of the digital band-pass filter in the D-ADC, and perform a more efficient filtering computation. Lastly, point interpolation for the band-pass filter coefficients is proposed to reduce the coefficient area footprint on silicon. This is necessary as it helps to reduce the silicon area by a large factor.

The proposed enhancements are proved viable and presented with Matlab simulations.

4.1 Decimation and Interpolation

This section introduces the interpolation and decimation process and the translational effect of decimation which is used in the proposed digital logic processing unit of the D-ADC.

4.1.1 The Decimation Process

Decimation is the process of down-sampling a signal to decrease its sampling rate. A filter, $H(z)$ normally precedes the decimation process, conducted by the decimator to avoid aliasing in the decimation process. This filter is called as a decimation filter. This setup is illustrated in Fig 4.1.

Figure 4.1: The decimation process with a filter preceding the decimator.

In the time domain, the decimator takes in $x(n)$ input, and outputs one sample out of every D number of inputs. D is an integer and is defined as the decimation or down-sampling factor of the decimator. Therefore, the relationship between the output sequence, $y(m)$ and the input, $x(n)$ is given as follows:

$$y(m) = x(Dm) \quad (4.1)$$

An example of the decimation process, where $D=2$ is shown in Fig 4.2. For an input sequence of $x(n)$ shown in Fig 4.2(a), the output sequence $y(m)$ is shown in Fig 4.2(b). The sampling rate of $y(m)$ is D times lower than $x(n)$ after the decimation process.

Figure 4.2: Decimation process for $D=2$. (a) Input sequence, $x(n)$ (b) Output sequence for $y(m)$.

Denoting the z -transform of $x(n)$ as $X(Z)$, the z -transform of $y(m)$ as $Y(Z)$, the z -transform of the output of the decimator, $Y(Z)$ is given by:

$$Y(Z) = \frac{1}{D} \sum_{k=0}^{D-1} X\left(Z^{\frac{1}{D}} e^{-j\frac{2k\pi}{D}}\right) \quad (4.2)$$

Substituting Z by $e^{j\omega}$ in equation 4.2, the Fourier transform of the decimator output is obtained as:

$$Y(e^{j\omega}) = \frac{1}{D} \sum_{k=0}^{D-1} X\left(e^{j\frac{\omega - j2k\pi}{D}}\right) \quad (4.3)$$

From equation 4.3, it can be observed that $Y(e^{j\omega})$ has the original $X(e^{j\omega})$ spectrum with a sampling rate reduced by factor D . The replicas are closer to each other. The magnitude of the original $X(e^{j\omega})$ spectrum is reduced by a factor of D . These effects are illustrated in Fig 4.3.

Figure 4.3: (a) Original signal spectrum, $X(e^{j\omega})$, and (b) Resultant down-sampled spectrum, $Y(e^{j\omega})$, if $-\pi/D < X(e^{j\omega}) < \pi/D$, and (c) Result down-sampled spectrum, $Y(e^{j\omega})$, if $-\pi/D > X(e^{j\omega}) > \pi/D$.

Fig 4.3(a) shows the original spectrum $X(e^{j\omega})$, and Fig 4.3(b) shows the resultant spectrum of $X(e^{j\omega})$ after the decimation process by a factor of D . In this illustration, frequency spectrum of $X(e^{j\omega})$ is bandlimited to π/D , and no aliasing occurs after decimation. If $X(e^{j\omega})$ has a bandwidth

that is greater than π/D , overlapping between the original spectrum and the replica will occur, therefore producing the resultant spectrum in Fig 4.3(c). Thus in order that no aliasing occurs, the decimator is normally preceded by a filter, $H(z)$ to ensure that the spectrum, $X(e^{j\omega})$ is bandlimited to π/D .

4.1.2 The Interpolation Process

In contrast to the decimation process, interpolation is the process of up-sampling a signal to increase its sampling rate. An anti-image filter, $H(z)$ normally follows after the expander to remove the images in the interpolation process. This filter is called the interpolation filter. This setup is illustrated in Fig 4.4.

Figure 4.4: The interpolation process with a filter following the expander.

In the time domain, the expander takes in $x(n)$ input, and inserts $(L-1)$ number of zeros in-between each input samples. L is an integer and is defined as the interpolation factor or up-sampling factor of the expander. Therefore, the relationship between the output sequence, $y(m)$ and the input, $x(n)$ is given as follows:

$$y(m) = \begin{cases} x(\frac{m}{L}), & m = kL, k \text{ is an integer} \\ 0, & \text{otherwise} \end{cases} \quad (4.4)$$

An example of the interpolation process, where $L=2$ is shown in Fig 4.5. For an input sequence of $x(n)$ shown in Fig 4.5(a), the output sequence, $y(m)$ is shown in Fig 4.5(b). The sampling rate of $y(m)$ is L times higher than $x(n)$ after the interpolation process. The anti-image filter, $H(z)$ also acts to interpolate the zero-valued samples to produce the sequence, $u(m)$. The sequence, $u(m)$ is shown in Fig 4.5(c).

Figure 4.5: Interpolation process for $L=2$. (a) Input sequence, $x(n)$, and (b) Output sequence for $y(m)$, and (c) Interpolated zero-valued samples, $u(n)$.

Denoting the z -transform of $x(n)$ as $X(Z)$, the z -transform of $y(m)$ as $Y(Z)$, the z -transform of the output of the expander, $Y(Z)$ is given by:

$$Y(Z) = X(Z^L) \quad (4.5)$$

Substituting Z by $e^{j\omega}$ in equation 4.5, the Fourier transform of the expander output is obtained as:

$$Y(e^{j\omega}) = X(e^{j\omega L}) \quad (4.6)$$

From equation 4.6, it can be observed that it can be observed that $Y(e^{j\omega})$ has the original spectrum, $X(e^{j\omega})$ with a sampling rate increased by a factor L . The replicas are shifted closer to each other by factor L . These effects are illustrated in Fig 4.6. Fig 4.6(a) shows the original spectrum, $X(e^{j\omega})$, and Fig 4.6(b) shows the resultant spectrum of $X(e^{j\omega})$, after the expander output

with a factor $L=2$. It can be seen that the bandwidth of the original spectrum, $X(e^{j\omega})$ is reduced by a factor of $L=2$. And replicas occur at π/L . The anti-image filter, $H(z)$ can be used to remove the unwanted replicas. Fig 4.6(c) shows the output spectrum of the anti-image filter, $H(z)$ if the filter is a low-pass filter with a cut-off frequency of π/L .

Figure 4.6: (a) Original signal spectrum, $X(e^{j\omega})$, and (b) Resultant up-sampled spectrum, $Y(e^{j\omega})$, and (c) Output spectrum of the anti-image filter, $U(e^{j\omega})$.

4.1.3 Frequency Translation by Decimation

Decimating band-pass signals has inherent frequency translation property. A band-pass filter, $H(z)$ normally precedes decimation to ensure that the incoming signal is band-pass in nature. This setup is illustrated in Fig 4.7.

As long as the signals are band-pass in nature, and $BW < f_s/2D$. BW is given as the bandwidth of the signals, f_s as the sampling rate of the signals, and D as the decimation factor. This will ensure that the final sample rate is high enough for the signal of interest band and confined the frequencies in the region: $m\pi/D < \omega < (m+1)\pi/D$. M is given as an integer greater than zero. An example of the frequency translation by decimation is shown in Fig 4.8. The first part of Fig 4.8 illustrates an example of $m=2$ such that $XBp(e^{j\omega})$ is bandlimited to $2\pi/D < \omega < 3\pi/D$. The second part of Fig 4.8 shows that the band-pass spectrum is translated to the baseband region of $0 < \omega < \pi/D$ by decimation. With this phenomenon, one can translate a band-pass signal down to base-band without need for multiplication resources.

Figure 4.7: The frequency translation by decimation process with a filter preceding the decimator.

Figure 4.8: Frequency translation by decimation for $m=2$.

4.1.4 Polyphase Decomposition

Decimation and interpolation processes involving FIR (Finite Impulse Response) filters have computational complexity which can be reduced with the help of the polyphase decomposition [14] technique. In this section, the polyphase decomposition technique is reviewed and analyzed with equations.

Denoting the z-transform of a FIR-type filter, $h(n)$ as $H(z)$:

$$H(z) = \sum_{n=-\infty}^{+\infty} h(n)z^{-n} \quad (4.7)$$

The N-channel poly-phase decomposition of $H(z)$ is given by:

$$H(z) = \sum_{l=0}^{N-1} z^{-l} E_l(z^N) \quad (4.8)$$

Where the sub-phase filters are given by:

$$E_l(z) = \sum_{n=-\infty}^{+\infty} e_l(n)z^{-n}; l = 0, 1, \dots, N-1 \quad (4.9)$$

And with the l -th sub-phase filter as:

$$e_l(n) = h(Nn + l) \quad (4.10)$$

Therefore for a decimation filter, $H(z)$ with a decimation factor of M , it can be decomposed into its M number of polyphase components. Fig 4.9(a) shows the original decimation filter, $H(z)$ structure. Fig 4.9(b) shows the decomposition of $H(z)$ into M polyphase components. Using first cascaded equivalence, the structure shown in Fig 4.9(b) can be implemented as Fig 4.9(c). The sampling rate at the input of each polyphase component is reduced by a factor of M with respect to the sampling rate of input sequence, $x(n)$. This reduces the computational rate of the filter structure by a factor of M .

Similarly for an interpolation filter, $H(z)$ with an interpolation factor of L , it can be decomposed into its L number of polyphase components. Fig 4.10(a) shows the original interpolation filter, $H(z)$ structure. Fig 4.10(b) shows the decomposition of $H(z)$ into L number of polyphase components. Using first cascaded equivalence, the structure shown in Fig 4.10(b) can be implemented as Fig 4.10(c). The sampling rate at the input of each polyphase component is equivalent to the input sequence, $x(n)$. This reduces the computational rate of the filter structure by a factor of L , as compared to the original structure.

Figure 4.9: M -decimation filter implemented on (a) Direct form, (b) Polyphase decomposition, and (c) First cascaded equivalence.

Figure 4.10: L-interpolation filter implemented on (a) Direct form, (b) Polyphase decomposition, and (c) First cascaded equivalence.

4.2 A Simplified Approach to Baseband Recovery

This section reviews a method for a band-pass signal spectrum down-conversion to baseband without the use of any multiplication resources. From Section 4.1.3, a band-pass signal band confined in the region of $m\pi/D < \omega < (m+1)\pi/D$ can be down-converted to baseband of $0 < \omega < \pi/D$ purely by decimation. With this, if a band-pass spectrum contains a carrier whose frequency is in the middle of the region of $m\pi/D < \omega < (m+1)\pi/D$, and letting $m=2$ as an example, the resultant spectrum by decimation is illustrated in Fig 4.11.

Figure 4.11: Original signal and resultant down-sampled spectrum.

In order to remove the carrier frequency from the signal spectrum under conventional approach, convolution of the quantized signal data with the carrier signal using multipliers in digital mixers is needed. However in this case, the sampling frequency is 4 times the frequency of the carrier sine signal which is always given as repetitions of 0, 1, 0, -1. Thus no multipliers are needed to digital mix the carrier sine signal with the quantized signal data. Only an inversion

of the Most Significant Bit (MSB) is needed. This results in the signal spectrum given in Fig 4.12. The aliased image is also shown as the 2nd order frequency components. After aliasing, the resultant spectrum will consist of signal f_a and signal $2f_c - f_a$ in the baseband. By applying a decimation of 2 of the spectrum, all the signals are aliased into one another, giving only a signal f_a in the base-band. This is illustrated in Fig 4.13.

The two step process of performing a baseband down-conversion of a band-pass spectrum involving convolution of the quantized signal data with the carrier signal, followed by decimation by 2, a multiplication-less process can be used. This is only possible by creating a relationship between the down-converted carrier frequency and the sampling frequency of the down-converted data. This sampling frequency of the down-converted carrier frequency has to be a quarter of the down-converted data. In the time domain, the required data to be calculated is purely multiplication of $(-1)^n$. In digital logic format, this will mean only an inversion of the MSB of the digital data. Thus the removal of the carrier frequency from the signal spectrum, and the recovery of the input analog frequency components are done without the use of multiplication resources. The time domain operation of the whole process is illustrated in Fig 4.14.

Figure 4.12: Original signal and resultant down-sampled spectrum

Figure 4.13: Analog input signal recovered by decimation

Figure 4.14: Time domain operation of multiplication-less removal of carrier frequency

4.3 A Simplified Approach to Baseband Recovery in D-ADC

In the D-ADC, extraction of the input analog frequency components involved performing a baseband recovery of a band-pass signal spectrum. Thus by applying the simplified approach to baseband recovery concept from Section 4.2, the need for multiplication and digital low pass filter resources for the removal of the carrier sine frequency and baseband signal recovery can be avoided by choosing an appropriate carrier sine frequency for the D-ADC.

4.3.1 Determining the carrier frequency in D-ADC

A systemic method to realize the multiplication-less extraction of the input analog frequency components is listed in the steps below. This involves using the correct sine carrier frequency for the D-ADC.

- (1) State the required output sampling rate. Let the output sampling rate be S_{output} .
- (2) Calculate the base-band carrier frequency. Let the base-band carrier frequency be $S_{output}/2$.
- (3) Determine the requirements of the carrier frequency. Let the carrier frequency be F_c . For example, F_c might need to be greater than 5 kHz. This will result in a requirement of $F_c > 5 \text{ kHz}$.
- (4) Calculate the carrier frequency to be used. Therefore, $F_c = S_{output}/2 + n.(S_{output})$; and $F_c > 5 \text{ kHz}$. And n can be any integer: n=1,2,3 and so on.

Using the steps outlined above, one can choose an appropriate carrier sine frequency for the D-ADC in order to avoid the need for multiplication and digital low pass filter. This calculation is illustrated below. Applying the above steps:

- (1) Let $S_{output}=44.1 \text{ kHz}$, as output sampling rate for audio application is 44.1 kHz.
- (2) The base-band carrier frequency is determined to be $S_{output}/2 = 44.1 \text{ kHz}/2 = 22.05 \text{ kHz}$.
- (3) F_c has to be 3 times the analog input bandwidth of 20 kHz. Thus $F_c > 60 \text{ kHz}$.

- (4) Using $F_c = \frac{S_{output}}{2} + n.(S_{output})$, Let $n=1$, and $F_c = 66.15 \text{ kHz}$ and the condition of $F_c > 20 \text{ kHz}$ is satisfied.

Therefore from the above systemic way of determining the frequency of the carrier, one has avoided the need to use multiplication and digital low pass filtering resources to remove the carrier frequency.

4.3.2 System Modeling with the Simplified Approach

To prove the viability of the baseband recovery for the D-ADC using the simplified approach presented in Section 4.2, the D-ADC is re-modeled in Matlab as shown in Fig 4.15. Parameters being used for the model are detailed in Table 4-1.

Table 4-1: Table of values used in the Matlab model

Parameters	Values
$S_m(t)$	$1.\cos(2\pi \times 66.15 \times 10^3 \times t)$
$S_a(t)$	$0.9.\cos(2\pi \times 15 \times 10^3 \times t) + 0.1.\cos(2\pi \times 1 \times 10^3 \times t)$
Crossing sample rate	$10.8486 \times 10^6 \text{ Hz}$
Digital band-pass FIR filter specifications	Pass-Band: $66.15 \times 10^3 \text{ Hz} \pm 20 \times 10^3 \text{ Hz}$ Stop-Band: $0 - 37 \times 10^3 \text{ Hz}; 97 \times 10^3 \text{ Hz} - 10.8486 \times 10^6 \text{ Hz}$ Sampling rate: $10.8486 \times 10^6 \text{ Hz}$
Output sample frequency at audio application range	$44.1 \times 10^3 \text{ Hz}$

In the Matlab model, stage 3 is replaced by a down-sampling stage which down-samples the data down to the target output sampling rate followed by multiplication of the quantized data with square pulses consisting of $(-1)^n$. From this model, it could be seen that the only complex digital logic is the digital band-pass filter. The signal spectrum is plotted in Fig 4.16, and it shows the correct extracted signal frequency and amplitude. This has proven that the multiplication-less implementation for the baseband recovery of the signal components in the D-ADC is viable.

Figure 4.15: System model for D-ADC with multiplier-less removal of carrier frequency and recovery of the signal components.

Figure 4.16: Frequency spectrum extracted by multiplication-less removal of carrier frequency.

4.3.3 Possible Savings in Gate Count and Power

The possible savings by digital logic reduction is illustrated in Fig. 4.17. The use of the simplified approach to baseband recovery in D-ADC has avoided the use of a multiplier in stage 3 and removes the need of a digital low-pass filter in stage 4.

The digital band-pass filter has coefficient bit width of 22-bit in order to reach a -90 dB quantization noise floor. Thus the use of the multiplier in stage 3 would have to be a 22-by-22 bit width signed multiplier, introducing a gate count use of an estimated 2600 NAND gates, synthesized with a 0.18 μm technology node, using DesignWare libraries from Synopsys. The use of the digital low-pass filter in stage 4 introduces a need for approximately 80 coefficients with bit width of 18 bits. This translates to the use of 180 bytes of Random Access Memory (RAM) or Read-Only Memory (ROM) space, and a buffer RAM of similar size. Additionally, a 44-by-18 bit width signed multiplier and an adder running at 80 times the sampling speed are required to complete the filter calculations. Table 4-2 gives a comparison of gate count usage using the original and the simplified approach. From Table 4-2, we can estimate the gate count

savings from the simplified approach with respect to the original is approximately 6550, and memory space requirement is reduced by 360 bytes.

Table 4-3 gives the approximate power savings with the simplified approach due to the reduction in gate count. The switching power consumption of a digital gate is taken as approximately $70 \text{ nW} / \text{gate} / \text{MHz}$ and ROM (of 512 kbit) is taken as approximately $0.4 \text{ mW} / \text{MHz}$ in $0.18 \text{ } \mu\text{m}$ technology node. The potential savings in power is approximately 1.048 mW .

Figure 4.17: Gate count reduction illustration

Table 4-2: Gate count comparison between the 2 approaches

Approach	Multipliers (A)	Adders (B)	Digital inversion (C)	Memory space	Total digital gate count (A+B+C)
Original	1 x 22-by-22 bit; 1 x 44-by-18 bit	1 x 36 bit	-	2 x 180 bytes	6650
Simplified	-	-	1 x 22-bit	-	100

Table 4-3: Approximate power savings from the simplified approach

Components	Digital gate count	Operating Frequency (MHz)	Power consumption (mW)
22-by-22 bit multiplier	2600	0.2646	0.0482
44-by-18 bit multiplier	3800	3.528	0.938
36 bit adder	250	3.528	0.0617
360 bytes ROM	-	0.0441	0.0001
Digital inversion	10	0.0441	0.0000309
Total Savings			1.048

4.4 Digital Band-pass Filter Reduction

The present D-ADC scheme uses a digital FIR band-pass filter with over 2 million coefficients, as a combination of interpolation and decimation factors are used to obtain the targeted output sampling rate of 44.1 *kHz*. In this section, the proposal to reduce the power consumed by the band-pass filter is to (1) avoid interpolation in order to avoid costly coefficient implementation, and (2) the poly-phase filtering concept is introduced to efficiently reduce the operation frequency of the digital band-pass filter to an optimum of 44.1 *kHz*.

4.4.1 No Interpolation

Interpolation increases the sampling rate of the input data. And the FIR filter order increases proportionally with the sampling rate. In order to avoid using large number of coefficients for the

FIR band-pass filter in the D-ADC, interpolation should be avoided. As the sampling rate of the dirac pulses is required to be greater than 1 GHz in order to achieve a 16-bit resolution, thus an appropriate sampling rate would be 1.084860 GHz . This sampling rate is 24600 times of the targeted output sample rate of 44.1 kHz . This will mean the required decimation rate is 24600. The requirement is illustrated below in Fig 4.18.

Figure 4.18: Direct decimation to targeted output sample frequency without interpolation.

4.4.2 Efficient Implementation Using Polyphase Decomposition

In order to ensure that digital computation rate is optimal, the computational rate should be kept at a value not exceeding twice the highest frequency component contained in the sampled domain. In this case, the optimal computation rate is 44.1 kHz . The optimal computational rate of the digital FIR band-pass filter in combination with decimation function can be achieved by using polyphase decomposition technique presented in Section 4.1.4. By applying the polyphase decomposition design technique, the specifications for the polyphase digital band-pass FIR filter is presented in Table 4-4. The block diagram of the designed polyphase digital band-pass filter is illustrated in Fig 4.19. The frequency response of the digital polyphase band-pass filter is illustrated in Fig 4.20. The interface operating frequency between $u(n)$ and the polyphase digital band-pass filter can be reduced to a minimal by using time location of the dirac pulses instead of channeling the pulses directly into the polyphase filter.

Table 4-4: Table of specifications for digital FIR band-pass filter

Parameters	Values
Pass-Band	$66.15 \times 10^3 \text{ Hz} \pm 20 \times 10^3 \text{ Hz}$
Stop-Band	$0 - 37 \times 10^3 \text{ Hz}; 97 \times 10^3 \text{ Hz} - 1.08486 \times 10^9 \text{ Hz}$
Pass-Band Attenuation	0.05 (Not critical to used application)
Stop-Band Attenuation	0.00001 (To reach -100 dB noise floor)
Sampling rate	$1.08486 \times 10^9 \text{ Hz} / 24600 = 44.1 \text{ kHz}$
Number of coefficients	885600
Fix-point width	22-Bit
Number of poly-phase components	24600
Number of taps per poly-phase component	$885600 / 24600 = 36$

Figure 4.19: Block diagram of designed poly-phase digital band-pass filter.

Figure 4.20: Frequency response of the polyphase band-pass filter.

4.4.3 Coefficients Silicon Footprint Reduction

The number of coefficients required to implement the digital FIR band-pass filter is calculated to be 885,600, and the optimal fix-point width is 22-bit in order to reach a -90dB quantization noise floor. This takes up 2.5 Mbytes of RAM or ROM space if this band-pass filter is to be implemented on a silicon chip. Thus there is a need to further reduce the number of coefficients in order to reduce the silicon footprint taken up by the FIR band-pass filter. The proposal to achieve a lower silicon area is to interpolate the points of the coefficients of the filter from a smaller set of points. Since the FIR-type filter coefficients is symmetrical in its centre, only half of the coefficients need to be in the silicon. By using point interpolation illustrated in Fig 4.21, the filter coefficients footprint can be further reduced. The black dots shown in the figure represents the coefficients recorded and that is used to calculate the actual filter response of the filter. Figure 4.22 shows the frequency response of the 200-point interpolated band-pass filter. With a 200-point interpolated filter, the silicon footprint of the coefficients is reduced to memory size of 6 kbytes. The stop-band attenuation of the filter achievable is -90 dB.

Figure 4.21: Point interpolation for a typical FIR filter response

Figure 4.22: Frequency response of the 200-point interpolated filter.

4.4.4 Possible Savings in Gate Count and Power

The original D-ADC uses 32 kbytes of ROM space to hold its 2,000,000 coefficients using point interpolation to certain extent.

With the proposed approach using (1) no interpolation in the digital band-pass filter and (2) coefficient silicon footprint reduction: no interpolation reduces the number of digital band-pass filter coefficients from 2,000,000 to 885,600. With this, memory footprint is reduced by half. In addition, the 200-point interpolation further reduces the memory footprint by 200 times. The resultant ROM space required is approximately 6 kbytes. However, this approach in turn requires addition of a ROM space to hold the value of the incremental steps that is associated with each point of the filter coefficient. The incremental step holder needs 3.3 kbytes of ROM space. Additionally, a 12-by-8 bit width multiplier and an adder is required. Thus the resultant ROM space required is only 9.3 kbytes, and an additional 750 gates is needed for the digital calculator. This approach gives a savings of 22.7 kbytes of ROM space. A snippet of the digital calculator is given in the Appendix, section A.3.

With the proposed approach of using efficient implementation with polyphase decomposition, the digital band-pass filter would run at the output sampling rate of 44.1 kHz, reduced down from the expected 264.6 kHz in the original design. The expected power saving is reduced by a factor of 6 in the digital band-pass filter.

4.5 Conclusion

In this chapter, the D-ADC is proposed to have the following enhancements to its digital logic unit: (1) Simplified approach to baseband recovery, (2) Polyphase decomposition technique for the digital band-pass filter, and (3) Point interpolation for the band-pass filter coefficients. Enhancement (1) and (3) helps to reduce the amount of silicon area used to implement the D-ADC, and (2) helps to reduce the operating frequency of the digital logic unit in the D-ADC. This further reduces the power consumption required for the D-ADC operation.

Chapter Five

5 Possible Enhancements for Analog Unit in D-ADC

In order to attain a low quantization noise floor of approximate -90 dB for a high resolution 16-bit ADC operation, a high sampling rate of 1 GHz is required in the analog front end. This is undesirable as a ring oscillator or a phase-locked loop (PLL) is necessary for such high frequency generation. In addition, having such high sampling rate, it would be difficult to increase the analog front end signal bandwidth, as a higher signal bandwidth will require an even higher sampling rate.

In this chapter, an analysis of the analog front end sampling rate reduction is presented. And the differential with dithering scheme is proposed to help to reduce the analog front end sampling rate without impacting the performance of the D-ADC.

The proposal is proved viable and presented with Matlab simulations.

5.1 Effects of Sampling Rate Reduction

In this section, the effects of the reduction of sampling rate are studied and presented. With the system models established in Section 3.3, it can be observed that with the sampling frequency reduction from 1.08486×10^9 Hz to 10.8486×10^6 Hz, the quantization noise floor raises from -90 dB to approximate -60 dB. Further reduction of the sampling frequency down to 4.2336×10^6 Hz raises the quantization noise floor greater than -50 dB. This is illustrated in Fig 5.1. With this understanding, 2 methods are proposed to reduce the noise floor while keeping the sampling frequency low. Two main schemes are (1) Differential Scheme for Analog Front End, (2) Dithering Scheme in addition to differential scheme.

Figure 5.1: Frequency spectrum using sampling rate of ~4 MHz.

5.2 Differential Scheme

With a reduction in sampling frequency, it is observable that random noise or error gets injected into the analog front end stage, and the block model of such stage is hypothesized and illustrated in Fig 5.2. In order to reduce the amount of noise injected into the system through the analog front end stage, a differential scheme is proposed. The differential scheme is applied in such a way that it helps to subtract away the error that is injected into system at the analog front end stage. The block diagram of such a scheme is illustrated in Fig 5.3. The resultant signal spectrum from a sampling rate of $4.2336 \times 10^6 \text{ Hz}$ is re-analyzed in such a differential scheme system. This signal spectrum is shown in Fig 5.4. From the analysis, it seems that the differential scheme does not provide much improvement in reduction of the quantization noise floor. The noise floor value is still at an approximate -50 dB. Only common mode noise is rejected.

Figure 5.2: Noise injection model for analog front end stage.

Figure 5.3: Differential scheme for analog front end.

Figure 5.4: Frequency spectrum using differential scheme.

5.2.1 Differential with Dithering Scheme

The differential scheme did not noticeably reduce the noise of the system. The hypothesis of the failure is attributed to that the noise injected into the system could be uncorrelated in a certain way, and the noise could not be removed effectively by the differential scheme. Thus an additional dithering scheme is to be introduced in order to correlate the analog input signal before it enters the system. This will ensure that the quantization stage introduces error which is completely correlated. The analog front end Matlab model with a differential and dithering scheme is illustrated in Fig 5.6. The resultant frequency spectrum is plotted in Fig 5.5. From the plot, one could see an improvement in lowering the noise floor back to approximate -60 dB, giving the same performance result when using a sampling rate of $10.8486 \times 10^6 \text{ Hz}$.

Figure 5.5: Frequency spectrum from differential and dither scheme.

Figure 5.6: System model of D-ADC with differential and dither scheme.

5.2.2 Trade-Offs

The fundamental trade-off for introducing the differential with dithering scheme is that the analog front end area has to be doubled. This involves introducing two sets of passive low-pass filters and high frequency counters to track the zero-crossings that occur. In the digital processing unit of the D-ADC, additional gate area is needed to hold or multiplex differential

data into the computational resources, and computational frequency may have to be doubled. In this way, the power savings might not actually be significant. The usefulness of this scheme comes when the sampling rate of the analog front end zero-crossings is limited by the technology node, and bandwidth needs to be increased further. Power saving is not considered with this scheme.

5.3 Conclusion

In this chapter, the D-ADC is proposed to have the following enhancement to its analog front end unit: Differential with dithering scheme. This enhancement helps to reduce the sampling rate required by the analog front end, and paves the way for a higher bandwidth D-ADC.

Chapter Six

6 Realization of Hardware with VHDL

The D-ADC is improved in power consumption by reducing the digital logic gates, and digital logic frequency operation. This is achieved by introducing (1) multiplication-less removal of the carrier frequency and recovery of the signal components, (2) use of polyphase architecture to implement the digital band-pass filter, and (3) use of point interpolation to reduce the silicon footprint of the filter coefficients. These three features allow for optimal use of computational resources resulting in reduced digital logic gate count, and digital logic frequency operation. This in turn reduces the power consumption of the design. The sampling rate for the dirac pulses is also targeted for operating frequency reduction. This is done by introducing a differential subtractive dithering scheme for the analog front end. This scheme potentially allow for the reduction of the sampling rate without deteriorating the ADC's performance. This might not reduce the power consumption as the analog front end component size will increase. Reduction of the sampling rate for the dirac pulses also paths the way for higher bandwidth of the D-ADC.

With the enhanced features, the objectives of the project for delivering a lower power, and a higher bandwidth D-ADC are met. This chapter presents the realization of the hardware of the enhanced features of the D-ADC on hardware by the author with the use of the VHDL language which allows the design to be described into hardware. The chapter first presents the block architecture of the coded VHDL design, and the Design-Under-Test (DUT) simulation environment is discussed and explained. Second, the key components of the design are discussed. Lastly, the gate count and power estimation is presented.

6.1 Realization of Hardware with VHDL

The D-ADC design is designed and simulated in a full VHDL environment. The analog front end of the D-ADC is also modeled digitally in VHDL, and the whole design is simulated with a full digital simulator, Modelsim XE. Fig 6.1 provides an overview of the VHDL DUT simulation environment. Fig 6.2 shows the sampled data output from two differential single channel operations. It can be observed that the sampled data represents a sinusoidal output correctly, and each channel is differential with respect to each other. Fig 6.3 shows the differential sampled data output frequency spectrum. This is done by subtracting the channel output from the other and do perform an averaging of the resultant data. Some distortions are observed, and the potential root cause is due to the use of a digital single tone wave. This is verified to be a non-critical issue, as the correctness of the hardware will be re-verified on Field Programmable Gate Arrays (FPGA).

The VHDL analog front end model consists of a single tone digital sine wave generator which subtracts or adds itself from the digital sine carrier generated outside the analog front end. A noise-shaped digital sine carrier is not used in the test setup. Instead a pure sine wave is used as the carrier, as noise shaping effect cannot be tested in a digital simulated environment. The time location of the dirac pulses are recorded by high speed counters which are then forwarded to the digital part of the D-ADC for averaging and dithering before it is processed by the band-pass filter which in this case is a polyphase type coefficient look-up table. The resultant processed data is a 16-bit signed digital data which can be written out in a text file whose frequency spectrum is analyzed by Matlab. A single tone frequency spectrum is the expected test result. This setup results in a low cost development environment for verification. The whole setup is realized on low cost Personal Computer (PC) using a free digital simulator, Modelsim XE.

In the next section, the key components of the D-ADC VHDL hardware are described in details. The gate count and power estimation of a single channel is also given.

Figure 6.1: VHDL DUT simulation environment.

Figure 6.2: Sampled data output from two differential single channel operations.

Figure 6.3: Frequency spectrum of differential sampled data output.

6.1.1 Analog Front End Unit

The AFE unit of the D-ADC involves superimposing the analog input signal with the D-ADC sine carrier, and the crossing of these two signals leads to the generation of dirac pulses at specific time locations. In the DUT test environment, the analog input signal is represented by a pre-generated single tone digital sine wave which is 16-bit signed data in nature. The carrier is represented by a pre-generated single tone digital sine wave which is 7-bit signed data in nature. This replaces the original noise-shaped sine wave. The zero-crossings are generated by subtracting or adding the two signed data together, and comparing the resultant data with the zero threshold. A digital logic level of ‘1’ is generated if the resultant is greater than zero, and a digital logic of ‘0’ is generated if the resultant is less than zero. The rising and falling edges of the generated signal represents the location of the zero-crossings. The differential implementation scheme process is illustrated in Fig. 6.4. A snippet of the VHDL code is given in Appendix A.1.

Figure 6.4: Differential implementation scheme process.

6.1.2 Averaging Unit

High frequency counters are used to track the location of the zero-crossings with respect to D-ADC sine carrier. This tracking process is previously illustrated in Fig 3.3, and is repeated in Fig 6.5. In each sweep of the D-ADC sine carrier, two zero-crossings are recorded in counter A and Counter B respectively. These count values carries the information of time location of the zero-crossings with respect to the D-ADC sine carrier period, and can be forwarded directly to the polyphase band-pass filter for filtering. However, such a design structure is inflexible to a change in the sampling rate of the zero-crossings when needed. A change in the sampling rate of the zero-crossings will require a complete re-design of the band-pass filter.

In order to improve the above situation, a counter C and a digital averaging logic unit is introduced. Counter C is a reference counter that tracks the time period of the full sine carrier, and typically carries the information on the sampling frequency used to sample the zero-crossings. These count values are forwarded to a digital averaging unit that performs a normalization of the count to a base count of 16,400. This base count value is a constant which is the ratio of the original sampling rate of the zero-crossings to the D-ADC sine carrier frequency.

In this case, the mathematical expression is given by $1.08486 \times 10^9 / 66150 = 16,400$. Fig 6.6 illustrates this process. The use of the averaging unit allows a flexible change of the sampling

rate of the zero-crossings, without the need to re-design the whole band-pass filter. A snippet of the VHDL code is given in Appendix A.2.

Figure 6.5: High speed counters to approximate the location of the impulses with respect to the carrier sine wave.

$$\frac{\text{Counter A}}{\text{Counter C}} \times 16,400 = \text{first time location in single sweep}$$

$$\frac{\text{Counter B}}{\text{Counter C}} \times 16,400 = \text{second time location in single sweep}$$

Figure 6.6: Normalization of the count.

6.1.3 Band-pass Filter Unit

The band-pass filter unit performs the band-pass filtering of the data from the averaging process. It consists of a polyphase coefficient lookup table which is 9.3 kbytes in size. This is implemented with Read-Only-Memory (ROM). A buffer unit holds the incoming data for

filtering. This buffer is implemented in Static Random Access Memory (SRAM) and is 200 bytes in size. Fig 6.7 illustrates the block diagram of the implemented band-pass filtering unit. A snippet of the VHDL code is given in Appendix A.3.

Figure 6.7: Block diagram of the band-pass filtering unit

6.1.4 Gate Count and Power Estimation

In order to do a power estimation of the enhanced D-ADC design, gate count estimation is required. The silicon area and power consumption of the components in 0.18 μm technology node is given in Table 6-1. The digital part of the enhanced design is synthesized with a 0.18 μm technology node and the silicon area estimation for a single channel is given in the Table 6-2. By doubling the single channel to achieve a differential ADC, the active silicon area comes up to approximate $0.3573 \times 2 = 0.7146 \text{ mm}^2$. The original design has an active area of 1.05 mm^2 in 0.18 μm technology node [10]. Thus the silicon area of the enhanced design is estimated to be 30% less than the original design. The main savings in silicon area comes from the ROM area.

Digital power estimation for a single channel is given in Table 6-3. Analog Power estimation for a single channel is given in Table 6-4. From the given tabled data, the total estimated power for a single channel operation is approximately 3.31 mW . This power estimation is a pessimistic approximate, as the power consumed by the high speed counters is taken at the worst possible scenario where all the digital bits are assumed to be incrementing throughout the whole data conversion operation, which is not true. If the high speed counters are assumed to be

incrementing only at half the operation process, the total estimated power is 1.8631 *mW*. The original design consumes 3.6 *mW* with a single channel design [10]. This estimates to be 50% less than the original design. The main power savings comes from the gate count reduction by the simplified approach with baseband recovery, and the low operating frequency of the digital processing unit, which operates at only twice the frequency of the input bandwidth.

Table 6-1: 0.18 um technology node information

<i>Parameters</i>	<i>Area (mm²)</i>	<i>Power consumption</i>
Digital gate	8 x 10 ⁻⁶	70 nW / gate / MHz
ROM (512 kbits)	0.44	0.4 mW / MHz
SRAM (32 kbits)	0.40	0.08 mW / MHz

Table 6-2: Table for gate count estimation for a single channel

<i>Parameters</i>	<i>Size</i>	<i>Area (mm²)</i>
ROM	64 kbits	0.055
SRAM	1.6 kbits	0.02
Digital Part	16343 Gates	0.1323
Analog Part	Single channel	0.15
	<i>Total</i>	<i>0.3573</i>

Table 6-3: Table for digital power estimation for a single channel

<i>Parameters</i>	<i>Switching Power (mW/MHz)</i>	<i>Freq (MHz)</i>	<i>Power (mW)</i>
ROM (64 kbits)	0.05375	0.0441	0.00237
SRAM (1.6 kbits)	0.004	0.0441	0.0001764
Digital Part	1.1475	0.0441	0.0506
		<i>Total</i>	<i>0.0531</i>

Table 6-4: Table for analog power estimation for a single channel

<i>Parameters</i>	<i>Switching Power (mW/MHz)</i>	<i>Freq (MHz)</i>	<i>Power (mW)</i>
Ring Oscillator	N.A	1000	0.085
Digital Counters	N.A	1000	3.0
RC network	N.A	N.A	0.225
<i>Total</i>			<i>3.31</i>

6.2 Conclusion

The enhanced D-ADC design is implemented in hardware using VHDL by the author. The design is verified using a full digital simulation environment. The resultant gate count savings from the enhanced design is approximately 30%, and the estimated power consumption is 50% less than the original design. Table 6-5 summaries the comparison between the original design and enhanced design in terms of active area and power.

Table 6-5: Table for active area and power comparison

<i>Design</i>	<i>Active Area(mm²)</i>	<i>Power (mW)</i>
Original D-ADC	0.525	3.6
Enhanced D-ADC	~0.353	~1.9162

Chapter Seven

7 Experimental Results

Field Programmable Gate Arrays (FPGA) are gaining popularity as an alternative to custom integrated circuits and these FPGAs offered a programmable platform where both combinatorial and sequential logic can be implemented. High-end FPGAs can support multi-million gate designs, and can reach as high as 1 *GHz* of clock operation. With such capabilities, FPGAs are often sourced as a platform as pre-silicon verification for digital based designs. The enhanced D-ADC being an almost full digital design is implemented and tested on a Xilinx Virtex 4 ML402 FPGA platform [16], with the analog front end part represented by discrete passive resistors and capacitors mounted on a veroboard, or also known as stripboard. This allows a pre-silicon evaluation of the D-ADC design on real hardware. In this chapter, the test setup is described in section 7.1, and the remaining sections discuss about the results obtained from the test setup.

7.1 Test Setup

A single channel is realized in this test setup. The digital portion of the enhanced D-ADC is synthesized and implemented with the Xilinx ML402 FPGA platform, and the analog portion is represented on a veroboard using passive resistors and capacitors. The block diagram of the design setup is illustrated in Fig 7.1. The digital sine wave filter is achieved using a 2nd order low-pass resistor-capacitor (RC) filter. The smoothed digital sine wave is directed into the AD8056 amplifier which acts as a high-input impedance buffer using a voltage follower configuration. This prevents the input pad of the FPGA from distorting the smoothed digital sine wave as it draws current from the setup. The input pad of the FPGA also acts as a zero-crossing detector which detects the zero-crossing point at the mid-point voltage level. Glitches occur at the zero-crossing point due to random noise on the smoothed digital sine wave, thus there is a need to implement a 1st order low-pass RC glitch filter to remove these glitches. This setup in

turn averaged out the random noise on the smoothed digital sine wave. The sampling rate of the zero-crossings used is 300 *MHz*. This is the maximum frequency that the clock routing resources can be supported by the Virtex 4 FPGA device family. The output sampling rate of the converted data is 44.1 *kHz*. The converted data is dumped into a FPGA FIFO memory which is 8192 in depth, sufficient for performing a Fast Fourier Transform (FFT) to analyze the amplitude and frequencies present in the converted data. These contents are extractable by using Joint Test Action Group (JTAG) protocol using a Personal Computer (PC).

7.1.1 Equipment Setup

The equipment setup surrounding the FPGA design is shown in Fig 7.2. A photo of the setup is taken and shown in Fig 7.3. Fig 7.3(a) shows the complete board test setup to do a dynamic test of the D-ADC. Fig 7.3(b) shows the electric components that are mounted on the veroboard. The FPGA platform used is Xilinx Virtex 4 ML402 evaluation platform. The clock generator used for the D-ADC design is from an onboard 100 *MHz* crystal oscillator. The D-ADC connects to the external world through the onboard connectors by routing the needed signals to these connectors. The Agilent signal generator is used to generate a single-ended sine signal that is low-passed by a resistor-capacitor (RC) filter to remove random noise before it is superimposed onto the noise-shaped sine signal that is generated by the D-ADC design. This signal acts as the external analog signal to be sampled by the D-ADC. The Xilinx JTAG module is used to communicate with the internal FPGA memory which helps to extract the 16-bit signed converted data onto the Personal Computer (PC) using a Xilinx JTAG cable. This converted data is then evaluated by Matlab. The working of the zero-crossing detection waveform is captured on the oscilloscope and is given in Appendix A.4.

Figure 7.1: Block diagram of the design setup on FPGA and veroboard.

Figure 7.2: Equipment Setup.

(a)

(b)

Figure 7.3: (a) Photo of the board setup (b) Photo of the electrical components on the veroboard

7.2 Experimental Results

As the sampling rate of the zero-crossings is reduced to 300 MHz due to the limitations of the technology node of the FPGA, the expected Signal-to-Noise Ratio (SNR) is an approximate -73 dB from Matlab simulations. This simulation results includes the potential noise effects due to the averaging unit. The amplitude of the D-ADC digitally generated sine signal is dependent on the output logic '1' voltage of the digital pads. This will also determine the maximum input analog signal amplitude that can be sampled. The biased point of input signal could vary about at the centre voltage depending on the output logic '1' voltage of the digital pads.

7.2.1 Sampling Signal Amplitude and its Quantized Step Voltage

The digital sampling sine signal is generated using 3 types of voltage leveled outputs which is done by shorting the relevant bank voltage pins to the required voltage level. The chosen voltage level is 2.5 V, 3.3 V and 5.0 V. Voltage level 2.5 V and 3.3 V is readily available on the ML402 FPGA platform and selected using a jumper. Voltage level 5.0 V is given externally whenever required to the voltage bank of the FPGA device.

The resultant amplitude of the generated sine signal is measured and its related single quantized step level is summarized in Table 7-1. The quantized step is calculated based on a 16-bit resolution converted data.

Table 7-1: Table for measured signal amplitude and the related quantized step

<i>Voltage at Output(V)</i>	<i>Signal Amplitude (V)</i>	<i>Quantized Step (mV)</i>
2.5	1.675	0.0256
3.3	2.243	0.0342
5.0	3.350	0.0511

7.2.2 Idle Tone Analysis

The idle tone analysis is done by feeding the digitally generated sine wave signal back to the D-ADC design through the AD8056 amplifier in the voltage follower configuration mode without superimposing any external signal. The measured converted data from the D-ADC gives the amount of noise in the overall hardware setup. This directly affects the dynamic SNR performance of the D-ADC design. 8192 samples of converted data are collected for each voltage level type of 2.5 V, 3.3 V and 5.0 V. The deviation of the sampled data in terms of Least Significant Bits (LSBs) are analyzed and plotted in Fig 7.4. From Fig 7.4(c), it is observed that 5.0 V voltage level output gives the least LSB error of 5.7, while 2.5 V level output gives the most LSB error of 6.5.

Figure 7.4: (a) 2.5 V – idle tone deviation chart (b) 3.3 V – idle tone deviation chart (c) 5.0 V – idle tone deviation chart.

7.2.3 Dynamic Testing

A single tone sine wave with a frequency of 6.559 kHz is used to do a dynamic test on the D-ADC design. The sine wave amplitude is a factor of 0.8 with respect of the digitally generated sampling sine wave. The test is conducted for each of the voltage level type of 2.5 V, 3.3 V and 5.0 V. 8192 samples of converted data are collected for each voltage level type, and the data are analyzed and plotted in Fig 7.5. Up to 20 spurious distortions are removed in the analysis. The single tone sine wave frequency is correctly converted in all the 3 voltage level types. Fig 7.5(a) shows the SNR performance for 2.5 V voltage output level. The achieved dynamic SNR is at an

approximate -58 dB. Fig 7.5 (b) shows the dynamic SNR performance for 3.3 V voltage output level. The achieved SNR is similar to 2.5 V voltage type. Fig 7.5(c) shows the SNR performance for 5.0 V voltage output level. The achieved dynamic SNR is an approximate -62 dB.

Figure 7.5: Frequency spectrum for dynamic testing for all 3 voltage level outputs.

The single tone test is repeated with increasing input amplitude with the voltage level type of 3.3 V, and the resultant SNR for various input amplitude is shown in Fig. 7.6. As expected, the SNR is best with increasing input amplitude towards full-scale. Input amplitude of full-scale is not possible, as no zero-crossings could be detected, and the internal counters might overflow giving wrong conversion results.

Figure 7.6: SNR performance of various input amplitude for single tone test

The single tone test is repeated with decreasing sampling frequency with the voltage level type of 3.3 V, and the resultant SNR for different sampling frequency is shown in Fig. 7.7. As expected, the SNR is best with higher sampling frequency, as the time locations of the zero-crossings can be captured correctly with higher sampling frequency. This trend agrees with the findings from Matlab in Section 5.1.

Figure 7.7: SNR performance for different sampling frequency for single tone test

7.2.4 DNL and INL Measurements

The differential non-linearity (DNL) and integral non-linearity (INL) measurement is done by collecting the samples at each digital output at the voltage level type of 3.3 V. A total of 8192 samples is collected for each digital output, and performed for a limited range of digital outputs from -22300 to -21000. The analysis of the data is done using a histogram test for linear ramp input concept [18].

Based on this test, the DNL is defined as:

$$DNL(n) = \frac{h(n)_{actual}}{h(n)_{theoretical}} - 1 \quad (7.1)$$

where $h(n)_{actual}$ is the exact number of occurrences of each digital code, and $h(n)_{theoretical}$ is the expected number of occurrences of each digital code.

The INL is defined as:

$$INL(n) = \sum_{i=1}^n DNL_i \quad (7.2)$$

where the integration of DNL yields the INL data.

The DNL measurement is shown in Fig. 7.8, and is between +1.04 LSB and -0.88 LSB. The INL measurement is shown in Fig 7.9, and records a maximum +5.72 LSB and -0.45 LSB. Since DNL is more than +1 LSB at certain points, a missing code potentially occurs at that point. The measured INL records a large measurement of +5.72 LSB based on 16-bit of converted data resolution and this potentially reflects that the effective number of bits (ENOB) of the FPGA prototype is approximately 10 bits or so.

Figure 7.8: Differential non-linearity measurement.

Figure 7.9: Integral non-linearity measurement.

7.2.5 Discussion on Experimental Results

The test setup is able to verify that the enhanced D-ADC design works conceptually and realizable on hardware. From the simulation, the expected dynamic SNR of D-ADC with a 300 *MHz* sampling rate of the zero-crossings is approximate -73 dB. However, from the test setup, only a maximum dynamic SNR of - 62dB is obtained. This potentially can be explained by the

presence of noise on zero-crossing detector. The simulation uses an ideal clean sine wave signal. From the idle tone analysis, the estimated noise on the zero-crossing detector varies from 10 mV to 15 mV. To overcome the noise on the overall hardware setup, the magnitude of the digital sine wave has to be increased. As proven in the measured results, the 5.0 V voltage level type gives the best result with respect to 2.5 V and 3.3 V, reaching a SNR of -62 dB in the single tone dynamic test. The SNR can potentially be much higher if the analog front end is implemented fully in silicon, as the noise level on the zero-crossing detector should be much lower.

With a single tone dynamic test, the effective number of bits (ENOB) is defined as [17]:

$$ENOB = \frac{SNR - 1.76}{6.02} \quad (7.3)$$

Therefore the highest ENOB is achieved with a voltage output level of 5.0 V of 10.0 bits. The ENOB is evaluated across different input analog frequencies for voltage level type of 3.3 V, and this is shown in Fig. 7.10.

Figure 7.10: ENOB across various input frequencies

Using the estimated power consumption for the single channel from section 6.1.4, the Figure of Merit (FOM) is defined as [13]:

$$FOM = \frac{P}{2 \times f_B \times 2^{(DR-1.76)/6.02}} \quad (7.4)$$

Where P , f_B and DR denote the power dissipation, signal bandwidth and the dynamic range respectively. The enhanced D-ADC design achieves an estimated 0.455 pJ/level. Table 7-2 summarizes the performance of the D-ADC evaluated on the FPGA platform.

Table 7-2: Table of performance for enhanced D-ADC

<i>Targeted technology</i>	Digital 0.18 μm (Tested on Xilinx Virtex 4 XCVSX35 Device)
<i>Estimated active area</i>	0.3573 mm^2
<i>Dynamic range</i>	-62 dB (Voltage level 5 V for analog front end on FPGA platform)
<i>Estimated Power</i>	1.8631 mW
<i>Figure of Merit</i>	0.455 $pJ/level$

7.2.6 Discussion on Differential Scheme with Dithering for AFE

The experimental results shown in Chapter 7 are all collected based on a single channel setup on the FPGA platform, and is conclusive that the enhanced D-ADC design works conceptually and realizable on hardware.

The setup with differential scheme with dithering for AFE is attempted on the FPGA platform and illustrated in Fig 7.11. However, with such a setup it is found that the converted data from each of the two channels has noise elements that are not equivalent to each other, and worsen the final converted data instead. This could relate to the difference in the noise level that runs in each of the AFE setup for each channel. Thus the FPGA setup is not effective in testing this scheme effectively and conclusively. This is left as a topic for future work.

This also in turn prompts us to take care for silicon implementation that the differential setup for AFE has to match equivalently, or else difference in noise level in the circuit might not improve the D-ADC performance.

Figure 7.11: Differential Scheme with dithering setup on FPGA

Chapter Eight

8 Conclusions and Recommendations

In this thesis, a literature research on the various full digital ADCs architectures is done. Christoph Braun's approach to a full digital ADC is adopted for study and improvement in areas of power consumption and bandwidth. The approach to reduce power consumption in the digital unit of the D-ADC is to reduce digital gate count and the operating frequency of the digital gates. The reduction in operating frequency in the analog front end (AFE) is proposed to be done by using a differential subtractive scheme. The outcome of these improvements results in a possible 30% reduction in silicon area, and a 50% reduction in power consumption. The enhanced D-ADC is implemented in hardware using VHDL, and a single channel configuration is tested on a Field Programmable Gate Arrays (FPGA) platform. The FPGA platform has proven the enhanced D-ADC works in principle, and achieved a -62 dB Signal-to-Noise Ratio (SNR) on itself. Effective number of bits on this platform is 10.0 bits.

8.1 Recommendations for Future Research

The power consumption of the enhanced D-ADC is still not ideal. This thesis focuses on enhancing the digital unit of the D-ADC, with few recommendations for the enhancement of the AFE. The AFE enhancement is worthy of future research. A few possible recommendations for future research are discussed in this section.

8.1.1 Silicon Footprint Reduction for Analog Front End

The AFE takes up approximately 0.15 mm^2 of silicon area on a 0.18 um technology node. The main area is taken up by the implementation of the bulky passive resistors and capacitor network. The D-ADC uses a resistor network to generate a sine wave digitally. This network consists of 7 resistors placed in parallel to one other, and terminated with a small capacitor. In

order that the current drawn in these resistors are in range of mA , the resistors have to be in $k\Omega$ magnitude. The D-ADC also uses the resistor and capacitor in series to form a low-pass filter, also known as resistor-capacitor (RC) filter. The cutoff frequency of a RC filter is given as:

$$f_{cutoff} = \frac{1}{2 \times \pi \times R \times C} Hz \quad (8.1)$$

Where f_{cutoff} , R and C denote the cutoff frequency of the filter, resistor and capacitor values respectively.

In the D-ADC AFE design, the maximum frequency that it has to pass through is the analog input signal that is to be converted. This maximum frequency is the bandwidth of the D-ADC design, and is designed to be 20 kHz . This in turns gives a required f_{cutoff} of 20 kHz . To achieve this, the RC value is given by:

$$R \times C = \frac{1}{2 \times \pi \times f_{cutoff}} = 7.96 \times 10^{-6} \quad (8.2)$$

For a typical 0.18 μm technology node, resistivity is an approximate 270 - 350 Ω per *square* using polysilicon material, while capacitor is represented in 0.7 fF per μm^2 . To reduce power consumption in range of mA , the resistor values has to be in order of $k\Omega$ magnitude, this will turn set to the capacitor value in the order of pF in magnitude. For a 1st order RC filter, with R = 100 $k\Omega$ and C = 3 pF , the approximate silicon footprint is 4400 μm^2 , factoring placement in the layout. Using a 2nd order RC filter, the silicon footprint increases by a factor of 2, giving an area size of 8800 μm^2 . If the resistor network for the sine wave generation is considered, this adds another approximate 4400 μm^2 again. Thus the total area taken by these bulky resistors is an approximate 13200 μm^2 .

The size of these bulky resistors is also limiting the design to be used in low bandwidth applications such as temperature sensors, where the bandwidth required is less than 100 Hz . This is because the required RC value will be much larger, and the silicon area increases.

Thus there is a need to find an improved method of realizing the bulky resistors and capacitors in the D-ADC AFE. The research work to replace these components by other

alternatives such as transistors will help to further reduce the silicon footprint of the D-ADC design.

8.1.2 Power Reduction for Analog Front End

The analog front end takes up more than 90% of the estimated power consumption in the enhanced D-ADC. Thus if any power consumption improvement is to be done, it should be in the analog part of the D-ADC. This proves to be next key approach to a low power digital type ADC design. The analog power estimation is shown in Table 6-4 and is repeated in Table 8-1.

Table 8-1: Table for analog power estimation for a single channel

<i>Parameters</i>	<i>Switching Power (mW/MHz)</i>	<i>Freq (MHz)</i>	<i>Power (mW)</i>
Ring Oscillator	N.A	1000	0.085
Digital Counters	N.A	1000	3.0
RC network	N.A	N.A	0.225
<i>Total</i>			<i>3.31</i>

From Table 8-1, 2 components of the AFE design consume the most power. They are the high frequency counters and the RC network. The counters consume high power due to its high frequency switching. The RC network draws a worst case of 0.126 mA of current through its 7 resistors in parallel considering a supply voltage of 1.8 V. The RC network power consumption could potentially be solved by solutions to Section 8.1.1. There is not yet an immediate solution to the high frequency switching for the counters. The proposed frequency reduction in this thesis increases the silicon area by a factor of 2 using the differential subtractive scheme. This would potentially not reduce the power consumption in the AFE.

Thus there is a need for further research work to find an improved method to avoid the high frequency counters.

Author's Publications

Research findings reported in this thesis have been published or are being submitted for consideration for publication in the following papers:

- R.W.T. Ng and L. Siek, “A Simplified Approach for Baseband Recovery in SDR Architectures”, *IEEE Asia Pacific Conference on Circuits and Systems*, pp. 1059 – 1062, December 2010.
- R.W.T. Ng and L. Siek, “Low Power Digital Type ADC”, accepted at *13th International Symposium on Integrated Circuits*, December 2011.

Bibliography

- [1] H. Lee and C.Sodini, "Analog-to-Digital Converters: Digitizing the AnalogWorld," *Proceedings of IEEE*, vol. 96, no. 2, February 2008.
- [2] Y. Chiu, B. Nikoli, and P. R. Gray, "Scaling of Analog-to-Digital Converters into Ultra-Deep-Submicron CMOS," *Proceedings of IEEE on Custom Integrated Circuits Conference*, pp. 375–382, September 2005.
- [3] R.H. Walden, "Analog-to-Digital Converter Survey and Analysis". *IEEE Journal on Selected Areas in Communications*. vol.17, no. 4, pp. 539 –550. April 1999.
- [4] T. Watanabe, T. Mizuno, and Y. Makino, "An All-Digital Analog-to- Digital Converter With 12- μ V/LSB Using Moving-Average Filtering," *IEEE Journal of Solid-state Circuits*, vol. 38, pp. 120–125, January 2003.
- [5] J. Mainardi, A. Adao, L. Carro and A. Susin, "A Comparison of Totally Digital ADCS for SOCs," *IEEE International Symposium on Circuits and Systems*, vol. 1, pp. 641-644, 2004.
- [6] J. Ortega and C.L. Janer, "Analog to Digital and Digital to Analog Conversion based on Stochastic Logic," *IEEE International Conference on Industrial Electronics, Control and Instrumentation*, vol.2, pp. 995–999. 1995.
- [7] B. Gaine. "Stochastic Computing System" *Advances In Information Systems Science* 2, pp37–172, 1969.
- [8] S. Toral and Quero, "Stochastic A/D sigma-delta converter on FPGA," *IEEE 42nd Midwest Symposium on Circuits and Systems*, vol.1, pp. 35–38, 1999.
- [9] Fu-yuan Wang, Yong-liang Li, Jiang-tao Xi and Chicharo, J.F., "Implementation of a Quasi-digital ADC on PLD," *IEEE Conference on Solid-State and Integrated Circuit Technology*, pp. 1791-1793, October 2006.
- [10] C. Braun and B.Engl, "A 1.8V digital A/D Converter in 0.18 μ m CMOS," *Symposium On VLSI Circuits Digest of Technical Papers*, pp. 144-145. 2002.
- [11] H. Farkhani, M. Meymandi-Nejad and M. Sachdev, "A Fully Digital ADC Using A New Delay Element With Enhanced Linearity," *IEEE International Symposium on Circuits and Systems*, pp. 2406–2409, May 2008.

- [12] US Patent No: 6300892 B2, "Linearized Multi-bit Digital/Analog Converter and its use in a Multi-bit Delta-Sigma Analog/Digital Converter".
- [13] S.Pavan, N. Krishnapura, R. Pandarinathan and P.Sankar, "A 90 μ W 15-bit $\Delta\Sigma$ ADC for Digital Audio," European Solid State Circuit Conference, pp. 198–201 ,2007
- [14] M.G. Bellanger, G. Bonnerot and M. Coudreuse, "Digital filtering by polyphase network: Application to sample rate alteration and filter banks," *IEEE Transaction on Acoustic Speech Signal Process*, vol. 24, pp. 109–114, April 1976.
- [15] F. Gerfers, M. Ortmanns and Y. Manoli, "A 1.5-V 12-bit power-efficient continuous-time third-order $\Delta\Sigma$ modulator," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 8, pp. 1343–1352, 2003.
- [16] Xilinx Evaluation platform user guide – ug080.pdf
- [17] D. A. Johns and K. Martin, *Analog Integrated Circuit Design*. Toronto: John Wiley
- [18] Analog Devices, *The Data Conversion Handbook*. Newnes, 2005.

Appendix A

A.1 VHDL of Analog Front End Unit Zero-Crossing

Detection

The VHDL given contains a main process where the pre-generated digital sine tone wave is read from a text file. And this digital sine wave is then superimposed onto the D-ADC carrier wave following the process in a differential way.

-- vhdl code begins

```
rfile : process
  file readfile    : text is in "sinus.dat"; -- read in pre-generated digital single tone wave
  variable vecline  : line;
  variable data_base : string (3 downto 1) := "BIN";
  variable var_ulv   : std_logic_vector (15 downto 0);
  variable add_int   : natural           := 0;
  variable zb        : natural           := 0;
begin
  while not endfile(readfile) loop
    readline (readfile, vecline);
    case data_base is
      when "BIN" => read (vecline, var_ulv);
      when others =>
    end case;
    memory(add_int) <= var_ulv;
    add_int      := add_int + 1;
  end loop;

  for k in 1 to 3999 loop
    for ki in 1 to 512 loop
      wait for clkring; -- determines the frequency of the digital sine tone
      wait for clkring;
      if (phase_tm = 0) then
        tsinus <= signed(memory(zb));
        zb    := zb + 1;
        if (zb = 10240) then
          zb := 0;
        end if;
      else
        tsinus <= (others => '0');
      end if;
    end loop;
  end loop;
```

```

        end loop;
    end loop;
end process rfile;
---DAC + Comparator function channel R-----
tzdacina    <= signed(dacin);
tzdacin(15 downto 9) <= tzdacina;
tzdacin(16)  <= tzdacina(6);
tzdacin (8 downto 0) <= "000000000";
zsin        <= tzdacin - tsinus;
softkomp    <= '1' when zsin >= 0 else '0';
---DAC + Comparator function channel L-----
tzdacial    <= signed(dacinl);
tzdacinl(15 downto 9) <= tzdacial;
tzdacinl(16) <= tzdacial(6);
tzdacinl (8 downto 0) <= "000000000";
zsinl       <= tzdacinl + tsinus;
softkompl   <= '1' when zsinl >= 0 else '0';

-- vhdl code ends
-----

```

A.2 VHDL of Averaging Unit

The division unit is an open source code downloaded from www.opencores.org and adapted to D-ADC requirements. It implements a non-restoring unsigned divider in verilog. This module is then instantiated for use in D-ADC, and the dividend, divisor parameters are loaded at a fixed period to do the averaging calculation. These are shown in the code below.

```

//-----
/-- division unit
module div_uu(clk, ena, z, d, q, s);

parameter z_width = 32;
parameter d_width = 16;

// inputs & outputs
input clk;           // system clock
input ena;           // clock enable

input [z_width-1:0] z; // dividend
input [d_width-1:0] d; // divisor
output [d_width-1:0] q; // quotient
output [d_width-1:0] s; // remainder
reg [d_width-1:0] q;
reg [d_width-1:0] s;

// functions

function [z_width:0] gen_s;
    input [z_width:0] si;
    input [z_width:0] di;
begin

```

```

    if(si[z_width])
        gen_s = {si[z_width-1:0], 1'b0} + di;
    else
        gen_s = {si[z_width-1:0], 1'b0} - di;
    end
endfunction

function [d_width-1:0] gen_q;
    input [d_width-1:0] qi;
    input [z_width:0] si;
    begin
        gen_q = {qi[d_width-2:0], ~si[z_width]};
    end
endfunction

function [d_width-1:0] assign_s;
    input [z_width:0] si;
    input [z_width:0] di;
    reg [z_width:0] tmp;
begin
    if(si[z_width])
        tmp = si + di;
    else
        tmp = si;

    assign_s = tmp[z_width-1:z_width-d_width];
end
endfunction

// variables

reg [d_width-1:0] q_pipe [d_width-1:0];
reg [z_width:0] s_pipe [d_width:0];
reg [z_width:0] d_pipe [d_width:0];

reg [d_width:0] div0_pipe, ovf_pipe;

// perform parameter checks

// synopsys translate_off
initial
begin
    if(d_width !== z_width / 2)
        $display("div.v parameter error (d_width != z_width/2).");
    end
// synopsys translate_on

integer n0, n1, n2, n3;

// generate divisor (d) pipe
always @(d)
    d_pipe[0] <= {1'b0, d, {(z_width-d_width){1'b0}} };

always @(posedge clk)
    if(ena)
        for(n0=1; n0 <= d_width; n0=n0+1)

```



```

    d_pipe[n0] <= #1 d_pipe[n0-1];

// generate internal remainder pipe
always @(z)
    s_pipe[0] <= z;

always @(posedge clk)
    if(ena)
        for(n1=1; n1 <= d_width; n1=n1+1)
            s_pipe[n1] <= #1 gen_s(s_pipe[n1-1], d_pipe[n1-1]);

// generate quotient pipe
always @(posedge clk)
    q_pipe[0] <= #1 0;

always @(posedge clk)
    if(ena)
        for(n2=1; n2 < d_width; n2=n2+1)
            q_pipe[n2] <= #1 gen_q(q_pipe[n2-1], s_pipe[n2]);

// flags (divide_by_zero, overflow)
always @(z or d)
begin
    ovf_pipe[0] <= !(z[z_width-1:d_width] < d);
    div0_pipe[0] <= ~|d;
end

always @(posedge clk)
    if(ena)
        for(n3=1; n3 <= d_width; n3=n3+1)
        begin
            ovf_pipe[n3] <= #1 ovf_pipe[n3-1];
            div0_pipe[n3] <= #1 div0_pipe[n3-1];
        end

// assign outputs
always @(posedge clk)
    if(ena)
        ovf <= #1 ovf_pipe[d_width];

always @(posedge clk)
    if(ena)
        div0 <= #1 div0_pipe[d_width];

always @(posedge clk)
    if(ena)
        q <= #1 gen_q(q_pipe[d_width-1], s_pipe[d_width]);

always @(posedge clk)
    if(ena)
        s <= #1 assign_s(s_pipe[d_width], d_pipe[d_width]);
endmodule

/-- division unit end
/-----

```

```

-----
-- vhdl instantiating divider and its use.

-- right divider -> crossing / zref
divider_r_0_inst : div_uu
port map (
  clk      => clk_i,
  ena      => ena_divider_r,
  z        => div_mul_v_r,
  d        => div_ref_s,
  q        => div_ans_r,
  s        => div_rm_r
);

-- at fixed period, the dividend is normalized with the help of the divider unit.

if (dadc_cnt_s = 254) then
  div_mul_v_r <= conv_unsigned(zabin_s(zab_count_s),16) * var; -- 16bit x 16bit
  div_ref_s <= zrefin_s; -- reference count
  ena_divider_r <= '1';
end if;

-- vhdl instantiating divider and its use.
-- end.
-----

```

A.3 VHDL of Band-pass Filtering Calculation

The VHDL given shows the main equations used for extracting the coefficients and the associated interpolated data from the ROM in order to generate the correct coefficient for the band-pass filtering.

```

-----
-- vhdl code begins

state_cal_p : process (clk_i, areset_n_i)
  variable coeff_data : integer range -(2**23) to (2**23)-1 ; -- 24 bit signed
  variable intp_steps : integer range -(2**11) to (2**11)-1 ; -- 12 bit signed
  variable intp_result : signed(20 downto 0); -- 20 bit signed

begin -- process
  if (areset_n_i = '0') then
    bandp_pt(0) <= 0;
    bandp_pt(1) <= 0;
    bandp_pt(2) <= 0;
  elsif (clk_i'event and clk_i = '1') then
    if state = "01" then
      case row_count is
        when 0 =>
          coeff_data := rom_1_c(conv_integer(unsigned(tap_ram_s_0(rd_count))) + (offs_rd_count*41));
          intp_steps := intp_rom_1_c(conv_integer(unsigned(tap_ram_s_0(rd_count))) + (offs_rd_count*41));
          intp_result := conv_signed(intp_steps, 12) * unsigned(tap_iram_s_0(rd_count));
          bandp_pt(0) <= coeff_data + conv_integer(signed(intp_result));
        when 1 =>

```

```

coeff_data := rom_2_c(conv_integer(unsigned(tap_ram_s_1(rd_count))) + (offs_rd_count*41));
intp_steps := intp_rom_2_c(conv_integer(unsigned(tap_ram_s_1(rd_count))) + (offs_rd_count*41));
intp_result := conv_signed(intp_steps, 12) * unsigned(tap_iram_s_1(rd_count));
bandp_pt(1) <= coeff_data + conv_integer(signed(intp_result));
when 2 =>
    coeff_data := rom_3_c(conv_integer(unsigned(tap_ram_s_2(rd_count))) + (offs_rd_count*41));
    intp_steps := intp_rom_3_c(conv_integer(unsigned(tap_ram_s_2(rd_count))) + (offs_rd_count*41));
    intp_result := conv_signed(intp_steps, 12) * unsigned(tap_iram_s_2(rd_count));
    bandp_pt(2) <= coeff_data + conv_integer(signed(intp_result));
when others =>
    end case;
end if;
end if;
end process;

-- vhdl code ends

```

A.4 Captured waveform of Zero-Crossing Detection

The input pad of the FPGA which acts as a zero-crossing detector is routed out of the FPGA and probed with the use of the oscilloscope to verify that the detection is working in general. This probed waveform is shown in the Fig. A.1. The sine wave is the smoothed digital sine wave generated by the D-ADC, and the square wave is the output of the input pad of the FPGA used for zero-crossing detection. Fig. A.2 shows the superimposed input analog sine onto the smoothed digital sine wave, and its corresponding detected zero-crossings represented by the square wave.

Figure A.1: Probe of zero-crossing detector output with respect to the input sine wave

Figure A.2: Probe of zero-crossing detector output with respect to the modulated sine wave