

Time-interleaved $\Delta\Sigma$ -DAC for Broadband Wireless Applications

by

Jennifer Pham

A thesis submitted in conformity with the requirements
for the degree of Master of Applied Science
Graduate Department of Electrical and Computer Engineering
University of Toronto

©Copyright by Jennifer Pham, 2007

Time-interleaved $\Delta\Sigma$ -DAC for Broadband Wireless Applications

Jennifer Pham

Master of Applied Science, 2007

Graduate Department of Electrical and Computer Engineering
University of Toronto

Abstract

The analysis and design of a time-interleaved delta-sigma digital-to-analog converter (TIM $\Delta\Sigma$ -DAC) is presented. The digital front-end of the TIM $\Delta\Sigma$ -DAC comprises a 95th-order time-interleaved-by-8 FIR interpolation filter and a 3rd-order time-interleaved-by-8 $\Delta\Sigma$ modulator. The time-interleaved architecture uses parallelism to support a low OSR of 8, which results in a large effective bandwidth for broadband applications. The 4-bit output of the $\Delta\Sigma$ modulator is converted into analog using 16 current-steering cells with continuous current calibration. The chip was fabricated in 90nm CMOS. It was designed to operate at 4GS/s with a bandwidth of 250MHz. The analog back-end was tested with modulated data from a simulation of the digital front-end. It was measured at 2.66GS/s and achieved a bandwidth of 166MHz, an SNR of 46dB and an SFDR of 56dB. At 2GS/s, the prototype consumed 102mW from a 1V supply.

Acknowledgments

Throughout the course of my thesis work, I have encountered numerous obstacles, at which there was always someone coming along with ingenuity, inspiration, and encouragement. There are so many people I would like to thank.

First of all, I am truly grateful to my supervisor, Tony Chan Carusone, who has given me continuous support and insight throughout this work. He gave me the freedom of research and motivated me to explore the field where I was a complete stranger. He has been like a friend who is always there to help and to listen.

I would also like to thank my colleagues for lending a hand whenever I got caught in the midst of confusion. Without their support, I would have much trouble completing this work. In particular, I would like to express my gratitude to Kentaro Yamamoto and Joseph Aziz who assisted me in the CAD design and experimental testing; Tyler Brandon from University of Alberta who patiently fixed countless DRC problems and guided me through the maze of 90nm CMOS Place & Route; Ahmad Darabiha, Ian Kuon, and Zdravko Lukic who supported me at different phases of the digital design flow; Keith Tang who provided me with custom RF pads; Marcus van Ierssel, Oleksiy Tyshchenko, and Cintia Man who helped me with the PCB design and digital test setup; and Jaro Pristupa who saved me in many CAD tool panics. I also would like to thank my peers in BA5000 for the endless laughters and priceless memories.

To my family, who never stopped encouraging and believing.

To Darren, who never stopped loving and caring.

Table of Contents

List of Figures	xiv
List of Tables	xv
List of Acronyms	xvii
1 Introduction	1
1.1 Motivations	2
1.1.1 Oversampled $\Delta\Sigma$ vs. Nyquist-rate DAC	3
1.1.2 Lowpass vs. Bandpass $\Delta\Sigma$ -DAC	4
1.1.3 Time-interleaved vs. Conventional $\Delta\Sigma$ -DAC	4
1.1.4 Summary	8
1.2 State-of-the-Art	9
1.3 Thesis Outline	10
2 Theoretical Background	13
2.1 System Architecture for $\Delta\Sigma$ -DAC	13
2.2 $\Delta\Sigma$ Modulator Architectures	15
2.2.1 Error-Feedback $\Delta\Sigma$ Modulator Architecture	16
2.2.2 Error-Feedback $\Delta\Sigma$ Modulator Stability Analysis	16
2.3 Time-interleaved $\Delta\Sigma$ Modulator	20
2.3.1 Polyphase Decomposition	20
2.3.2 Block Digital Filtering	21
2.3.3 Time-interleaved Error-Feedback $\Delta\Sigma$ Modulator	23
2.3.4 Practical Considerations	25

2.4	Time-interleaved Interpolation Filter	26
2.5	Summary	27
3	Time-interleaved $\Delta\Sigma$-DAC Design	29
3.1	Architecture Overview	29
3.2	Time-interleaved Interpolation Filter	30
3.3	Time-interleaved $\Delta\Sigma$ Modulator	34
3.3.1	DSM Architecture	34
3.3.2	NTF Optimization	35
3.3.3	Time-interleaved DSM	37
3.3.4	TIM-DSM Performance	40
3.4	Digital-to-Analog Converter Model	43
3.5	Analog Reconstruction Filter	45
3.6	Summary	45
4	Time-interleaved $\Delta\Sigma$-DAC Implementation	47
4.1	Digital Baseband Front-End	48
4.1.1	Hardware Optimization	48
4.1.2	Accuracy Optimization	49
4.1.3	Speed Optimization	51
4.1.4	Time-interleaved Interpolation Filter	53
4.1.5	Time-interleaved $\Delta\Sigma$ Modulator	56
4.1.6	Digital Integrated Circuits Design Flow	58
4.1.7	Digital Front-end Simulation Results	60
4.2	High-Speed Digital Interface	63
4.2.1	Multiplexer	63
4.2.2	Binary-to-Thermometer Converter and Switch Drivers	67
4.2.3	High-Speed Digital Interface Simulation Results	68
4.3	High Speed Analog Back-End	70
4.3.1	Current Calibration Circuitry	70
4.3.2	Current-Steering Digital-to-Analog Converter	73

4.3.3	Analog Back-end Simulation Results	80
4.4	TIM $\Delta\Sigma$ -DAC Integration	83
5	Time-interleaved $\Delta\Sigma$-DAC Performance	87
5.1	PCB Design and Test Setup	87
5.2	Digital Design Issues and Solutions	91
5.3	High Speed Analog Measurements	92
5.3.1	Initial Verifications	92
5.3.2	Accuracy Measurements	94
5.3.3	Linearity Measurements	97
5.3.4	Power Consumption	99
5.3.5	Performance Summary	100
6	Conclusions	101
6.1	Future Work	103
A	Conventional $\Delta\Sigma$ Modulator	105
B	TIM $\Delta\Sigma$-DAC Matlab Results	107
B.1	Analog Reconstruction Filter	107
B.2	TIM-IF-DSM Output Spectrum with DAC Mismatches	111
C	TIM $\Delta\Sigma$-DAC Implementation	113
C.1	TIM-IF Coefficients	113
C.2	TIM-IF Sum Trees	114
C.3	TIM-IF and TIM-DSM Timing Synthesis	116
C.4	Binary-to-Thermometer Converter and Switch Drivers	117
C.5	Current Calibration Principles	118
	References	120

List of Figures

1.1	Block diagram of a 60GHz radio	2
1.2	Parallel $\Delta\Sigma$ modulator based on Frequency Division Multiplexing (FDM) . .	5
1.3	Parallel $\Delta\Sigma$ modulator based on Code Division Multiplexing (CDM)	6
1.4	Parallel $\Delta\Sigma$ modulator based on Time Division Multiplexing (TDM)	7
1.5	TIM $\Delta\Sigma$ modulator based on digital block filtering	7
2.1	$\Delta\Sigma$ -DAC block diagram	14
2.2	Spectrum at each internal node in $\Delta\Sigma$ -DAC [1]	14
2.3	Linear model of a single-bit error-feedback $\Delta\Sigma$ modulator	16
2.4	Single-bit error-feedback $\Delta\Sigma$ modulator with digital limiter	17
2.5	Bit-wise analysis of error-feedback $\Delta\Sigma$ modulator	18
2.6	Stable error-feedback $\Delta\Sigma$ modulator	19
2.7	(a) Scalar transfer function, (b) Time-interleaved-by-M version [2]	22
2.8	Linear model of time-interleaved error-feedback $\Delta\Sigma$ modulator	24
2.9	First-order time-interleaved-by-two $\Delta\Sigma$ modulator	24
2.10	Conventional interpolation filter	26
2.11	Time-interleaved interpolation filter and time-interleaved $\Delta\Sigma$ modulator . .	27
3.1	Time-interleaved-by-8 $\Delta\Sigma$ -DAC block diagram	30
3.2	A 95 th -order FIR interpolation filter with and without coefficient quantization	32
3.3	A 95 th -order FIR interpolation filter block diagram	34
3.4	$\Delta\Sigma$ modulator noise transfer function optimization	37
3.5	Conventional 3 rd -order error-feedback $\Delta\Sigma$ modulator architecture	37
3.6	Time-interleaved-by-8 3 rd -order error feedback $\Delta\Sigma$ modulator	39

3.7	TIM-IF-DSM versus conventional DSM response (Matlab simulations)	40
3.8	TIM-IF-DSM performance for a single tone at $0.25f_B$ for non-optimized, optimized and quantized optimized NTF (Matlab simulations)	41
3.9	TIM-IF-DSM output spectrum for Matlab simulations with 0dBFS input amplitude at different frequencies a) $0.13f_B$ b) $0.25f_B$ c) $0.50f_B$ d) $0.93f_B$. . .	42
3.10	TIM-IF-DSM response (Matlab simulations)	43
3.11	Time-interleaved-by-8 $\Delta\Sigma$ -DAC architecture	46
4.1	a) Conventional $\Delta\Sigma$ -DAC b) Time-interleaved-by-8 $\Delta\Sigma$ -DAC	47
4.2	Error reduction rounding scheme	50
4.3	Example of an 8-bit CSA with 1-1-1-2-3 staging	52
4.4	TIM-IF Physical Implementation	54
4.5	TIM-IF sum tree for path 2 and 8	55
4.6	TIM-DSM sum tree	57
4.7	Digital design flow	59
4.8	TIM-IF-DSM output spectrum for VHDL behavioural simulations with 0dBFS input amplitude at different frequencies: a) $0.13f_B$ b) $0.25f_B$ c) $0.50f_B$ d) $0.93f_B$	61
4.9	TIM-IF-DSM VHDL Behavioural vs. Matlab Response	62
4.10	An 8-to-1 ring multiplexer	64
4.11	Timing diagram for an 8-to-1 ring multiplexer	65
4.12	An 8-to-1 ring multiplexer transient response (TT corner) a) 4GHz b) 2GHz	66
4.13	High-speed digital interface theoretical response	69
4.14	High-speed digital interface Cadence transient response (TT corner)	69
4.15	Current calibration implementation	71
4.16	Continuous current calibration system for 4-bit DAC	72
4.17	a) Bias current mirror b) Dummy calibration cell schematic	73
4.18	Current-steering cell with self-calibration circuitry	74
4.19	a) Output swing b) Output noise model c) Simplified output noise model . .	75
4.20	Current-steering DAC output load options	77
4.21	Active vs. Passive load output	78

4.22	DNL offset Monte Carlo analysis	80
4.23	TIM $\Delta\Sigma$ -DAC performance with and without current calibration (for active load, typical corner with transistor mismatch)	81
4.24	TIM $\Delta\Sigma$ -DAC's SNR/SNDR vs. Input amplitude for a single-tone input at $0.25f_B$ (TT corner)	82
4.25	TIM $\Delta\Sigma$ -DAC's SNR/SNDR vs. Input frequency for a single-tone amplitude of 0dBFS (TT corner)	82
4.26	a) Divide-by-8 clock divider b) I/O Driver	83
4.27	TIM $\Delta\Sigma$ -DAC floor planning	85
4.28	TIM $\Delta\Sigma$ -DAC final layout	85
5.1	Die photos of the TIM $\Delta\Sigma$ -DAC chip fabricated in 90nm CMOS	88
5.2	TIM $\Delta\Sigma$ -DAC prototype a) Packaging and b) Testboard	88
5.3	Analog back-end test flow	89
5.4	Full test setup for Agilent 93K SOC or Agilent ParBert platform	90
5.5	Experimental setup for analog back-end	90
5.6	Current-steering DAC stair case transient response	92
5.7	Output spectrum with calibration feed-through	93
5.8	Clock Divider Transient Response	94
5.9	CS-DAC transient response for a single-tone, 0dBFS input amplitude (top - single ended outputs; bottom - differential output)	94
5.10	Noise shape and inband spectra for a single-tone, 0dBFS input amplitude at $0.13f_B$ and $0.29f_B$	95
5.11	CS-DAC accuracy performance with single-tone input and passive load	96
5.12	Two-tone spectrum and SFDR measurements	97
5.13	Multi-tone Test	98
A.1	Linear model of first-order $\Delta\Sigma$ modulator	105
B.1	A 7 th -order elliptic analog filter response	108

B.2	TIM $\Delta\Sigma$ -DAC output spectrum with analog LPF for Matlab simulations with 0dBFS input amplitude at different input frequencies a) $0.13f_B$ b) $0.25f_B$ c) $0.50f_B$ d) $0.93f_B$	109
B.3	TIM $\Delta\Sigma$ -DAC response with an ideal vs. analog filter	110
B.4	TIM-IF-DSM output spectrum with thermometer DAC element mismatches	111
C.1	TIM-IF sum tree for path 3 and 7	114
C.2	TIM-IF sum tree for path 4 and 6	115
C.3	TIM-IF sum tree for path 5	115
C.4	Binary-to-thermometer schematic	117
C.5	Switch driver schematic	118
C.6	Calibration principle a) Calibration b) Operation	118

List of Tables

1.1	$\Delta\Sigma$ -DAC Design Specifications	8
3.1	Interpolation Filter Characteristics	32
4.1	CSA Staging Optimization	53
4.2	Binary-to-thermometer conversion and gate logic	67
4.3	Analog Back-end Transistor Properties	79
4.4	TIM $\Delta\Sigma$ -DAC Simulated Power Consumption	81
5.1	TIM $\Delta\Sigma$ -DAC Power Consumption	99
5.2	TIM $\Delta\Sigma$ -DAC Performance Summary	100
6.1	TIM $\Delta\Sigma$ -DAC Performance Comparisons	102
B.1	Analog Low-pass Filter Characteristics	108
C.1	A 95 th -order Time-interleaved-by-8 Interpolation Filter Coefficients	113
C.2	TIM-IF Synthesized Performance	116
C.3	TIM-DSM Synthesized Performance	116

List of Acronyms

ADC	Analog-to-Digital Converter
B2T	Binary-to-Thermometer
CDM	Code Division Multiplexing
CIFB	Cascaded Integrators with distributed Feedback
CIFF	Cascaded Integrators with Feed Forward coupling
CLA	Carry Look-ahead Adder
CRFB	Cascaded Resonators with distributed Feedback
CRFF	Cascaded Resonators with Feed Forward coupling
CS	Current-Steering
CSA	Carry Select Adder
CSD	Canonic Sign Digit
CS-DAC	Current-Steering Digital-to-Analog Converter
DAC	Digital-to-Analog Converter
dBFS	Decibel with respect to Full-Scale
DFF	D-flipflop
DFT	Discrete Fourier Transform
DR	Dynamic Range
DRC	Design Rules Check
DSM	Delta-Sigma Modulator
DSP	Digital Signal Processing
DWA	Data-Weighted Averaging
EFB	Error-Feedback
EFB-DSM	Error-Feedback Delta-Sigma Modulator
ENOB	Effective Number of Bits
FCC	Federal Communications Commission
FDM	Frequency Division Multiplexing
FIR	Finite Impulse Response

FO2	Fanout-of-2
FVT	Filter Visualization Tool
IF	Interpolation Filter
IIR	Infinite Impulse Response
ILA	Individual Level Averaging
IM2	Second-order Intermodulation
IM3	Third-order Intermodulation
IST	Information Society Technologies
LPF	Low-pass Filter
LSB	Least Significant Bit
LTI	Linear Time Invariant
LVS	Layout Versus Schematic
MASH	Multi-Stage Noise Shaping
MSB	Most Significant Bit
MTPR	Multi-tone Power Ratio
NTF	Noise Transfer Function
OBG	Out-of-band Gain
OFB	Output-Feedback
OFDM	Orthogonal Frequency Division Multiplexing
OSR	Oversampling Ratio
ParBert	Parallel Bit-Error-Rate
PCB	Printed Circuit Board
PLL	Phase Locked Loop
PVT	Process, Voltage, and Temperature
RCA	Ripple Carry Adder
RTL	Register Transfer Level
SFDR	Spurious-Free Dynamic Range
SISO	Single-Input Single-Output
SNDR	Signal to Noise plus Distortion Ratio
SNR	Signal to Noise Ratio

STF	Signal Transfer Function
STI	Shallow Trench Isolation
TDM	Time Division Multiplexing
TG	Transmission Gate
THD	Total Harmonic Distortion
TIM-DSM	Time-interleaved Delta Sigma Modulator
TIM-IF	Time-interleaved Interpolation Filter
UWB	Ultra-Wide Band
WPAN	Wireless Personal Area Networks

Introduction

In recent years, there exists a great competition for ultra-high data rate wireless communication systems to meet the emergence of broadband multimedia applications. The demand for wideband wireless personal area networking (WPAN) or wireless local area network (WLAN), and point-to-point or point-to-multipoint data links continuously pushes the capacity of wireless networks. Currently, the transfer capacity exceeds what can be accommodated in the widely used unlicensed bands (2.4GHz and 5.8GHz) for WLAN systems. An alternative solution is to resort to higher bands where bandwidth is abundant. Namely, in 2001, the Federal Communications Commission (FCC) set aside a continuous unlicensed block of 7GHz of spectrum between 57 and 64 GHz for short-range indoor WPAN/WLAN applications. A year later, the FCC opened up a licensed block 7.5GHz of spectrum between 3.1 GHz to 10.6 GHz for short-range indoor ultra wideband (UWB) applications [3].

As the complexity of a wireless communication system grows rapidly, research on of analog-to-digital (ADC) and digital-to-analog converters (DAC) have become completely independent of each other. While much of the research attention has been focused on the design of oversampled delta-sigma ADCs ($\Delta\Sigma$ -ADCs), their counterparts, delta-sigma DACs ($\Delta\Sigma$ -DACs), have been lagging behind. In reality, $\Delta\Sigma$ -DACs are as equally important and their implementations are often as challenging as those of $\Delta\Sigma$ -ADCs.

1.1. Motivations

subsectionApplications There has been continuous research by the European IST (Information Society Technologies) on system integration for 60GHz radios. They had proposed a hybrid dual-frequency system called BROADWAY which is based on an integration of HIPERLAN/2 (an existing 802.11a WLAN at 5GHz) and HIPERSPOT (an innovative fully ad-hoc extension at 60GHz). As a result, the intermediate frequency is purposely taken at 5GHz as shown in figure 1.1. This integration will result in wider acceptance and lower cost for both systems while providing a new solution for dense urban deployments [4].

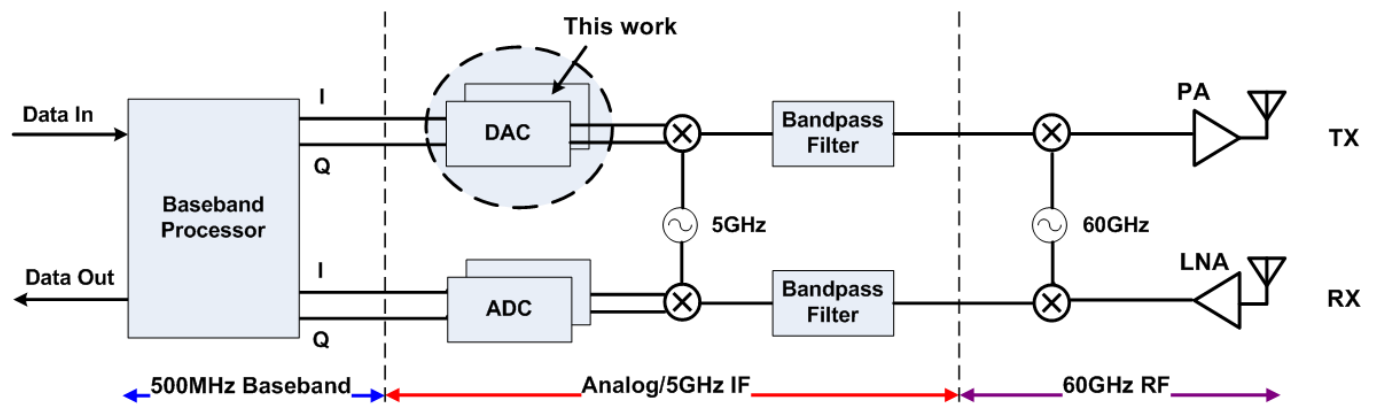


Figure 1.1: Block diagram of a 60GHz radio

While the final standards for 60GHz WPAN/WLAN and UWB are still under debate, both systems are likely to employ multi-band OFDM (orthogonal frequency division multiplexing) schemes capable of transmitting data in the 500Mb/s range [4, 5, 6, 7]. The DAC accuracy required for these systems ranges anywhere from 4-5 bits for an UWB transmitter [6, 7] to 8-10 bits for a 60GHz transmitter [8]. Since the 60GHz transmitter demands more rigorous requirements, this work will focus on designing a DAC for this application which will certainly suffice for an UWB design.

1.1.1. Oversampled $\Delta\Sigma$ vs. Nyquist-rate DAC

In the past, Nyquist-rate DACs were a popular choice for high speed, wideband data conversion while oversampled $\Delta\Sigma$ -DACs were favourable for high resolution and high linearity.

In Nyquist-rate DACs, the design of a subsequent analog reconstruction filter can be quite challenging due to its sharp cut-off, high attenuation requirements. Furthermore, they require a large analog circuitry, which is susceptible to potential analog circuit mismatches particularly in deep sub-micron processes (i.e.: 90nm, 65nm, etc). As CMOS is scaling, the matching factor degrades rapidly and becomes highly dependent on physical geometry and layout. Also, the severe short channel effects not only increase leakage current, but also intensify electrical fluctuations such as the deviation of threshold voltage [9, 10]. Consequently, future design trends lean toward a simple and small analog circuitry.

In oversampled $\Delta\Sigma$ -DACs, more design emphasis is placed on the digital front-end, thus relaxing the requirements on the analog back-end. For example, with the integration of a digital interpolation filter, the requirements on the analog reconstruction filter can be reduced. Also, since a $\Delta\Sigma$ modulator (DSM) modulates a multi-bit stream to only a few bits, this cuts down the number of analog unit elements significantly.

However, to date, most $\Delta\Sigma$ -DACs are for high linearity, high resolution, and narrow bandwidth applications such as digital audio. For 60GHz/UWB radio where wideband is the highest priority, the $\Delta\Sigma$ -DACs face a major challenge due to their narrow effective bandwidth as a result of large oversampling ratios (OSR). While some published Nyquist DACs can certainly meet the bandwidth requirement (e.g: [11, 12, 13, 14]), there has been relatively little research on high-speed, wideband $\Delta\Sigma$ -DACs.

Thus, this motivates the research of a high-speed, wideband $\Delta\Sigma$ -DAC to meet the demands of broadband wireless applications, and to accommodate the analog design challenges in deep sub-micron processes.

1.1.2. Lowpass vs. Bandpass $\Delta\Sigma$ -DAC

In a transmitter design, there are two design choices for the DAC: bandpass $\Delta\Sigma$ or lowpass $\Delta\Sigma$. For a bandpass design, the in-phase (I) and quadrature (Q) data components from the digital baseband are upconverted to an intermediate frequency before feeding into a quadrature bandpass $\Delta\Sigma$ -DAC. To date, $\Delta\Sigma$ -DACs employing bandpass modulation are still uncommon (eg: [15, 16, 17]), let alone quadrature $\Delta\Sigma$ -DACs. Research into the utility of quadrature $\Delta\Sigma$ -DACs indicates some promise [15, 18]. Nevertheless, at this point, the research is demonstrating feasibility rather than demonstrating significant advantages.

For a lowpass design, a quadrature lowpass $\Delta\Sigma$ -DAC is equivalent to a pair of lowpass modulators operating independently on the quadrature data components [1]. Thus, each I or Q data component can operate on its own lowpass $\Delta\Sigma$ -DAC before upconverting to an intermediate frequency, as depicted in figure 1.1. Certainly, quadrature mismatches between the $\Delta\Sigma$ -DACs of the two channels will be a source of degradation. One method to combat this error is through the use of quadrature mismatch shaping [19]. However, this topic is beyond the scope of this work. In general, compared to the bandpass option, the lowpass design shows a much lower level of complexity and has been proved to be feasible. Hence, this motivates the design of a lowpass $\Delta\Sigma$ -DAC.

1.1.3. Time-interleaved vs. Conventional $\Delta\Sigma$ -DAC

One way to achieve high bandwidth in a $\Delta\Sigma$ -DAC is to reduce the OSR. Some approaches to effectively reduce the OSR while maintaining the resolution are to either increase the order of the $\Delta\Sigma$ modulator (DSM), or to increase the number of converter bits, or a combination of both. Unfortunately, increasing the DSM order can lead to loop instability hence reducing the resolution and input dynamic range. An alternative to this problem is to employ a multistage noise shaping (MASH) $\Delta\Sigma$ loop. The main disadvantage of this scheme is that one must deal with analog mismatches when combining the stages' outputs together. On the other hand, increasing the number of converter bits can lead to high non-linearity due to component mismatches in the subsequent DAC. Often, additional circuitry is required to correct for these errors such as calibration, mismatch shaping, or digital correction. The DAC

complexity and its error-correction circuitry grows exponentially by a factor of 2^k for a k -bit DAC. Typically, k is chosen to be between 4-6 bits for practical implementations [1]. Some sophisticated designs for $\Delta\Sigma$ -DACs with over 6 bits such as dual-truncation or segmentation can also yield high SNR performance results but do not alleviate the non-linearity errors.

Another way to increase the bandwidth of a $\Delta\Sigma$ -DAC is to parallelize the DSM into multiple channels operating at lower speeds then combine their outputs. Parallel DSM can be categorized into three groups based on different multiplexing schemes: frequency division multiplexing (FDM), code division multiplexing (CDM), and time division multiplexing (TDM).

Figure 1.2 shows the block diagram of a parallel system based on a FDM scheme. This system contains a bank of parallel bandpass DSMs which have different band reject noise transfer functions and operate on different frequency sub-bands [20]. A digital bandpass filter attenuates the out-of-band noise in each channel and allows for recombination of the frequency-decomposed input signal [21]. This system has a high level of design and hardware complexity due to the requirement of many bandpass DSMs and bandpass filters, each with different center frequencies.

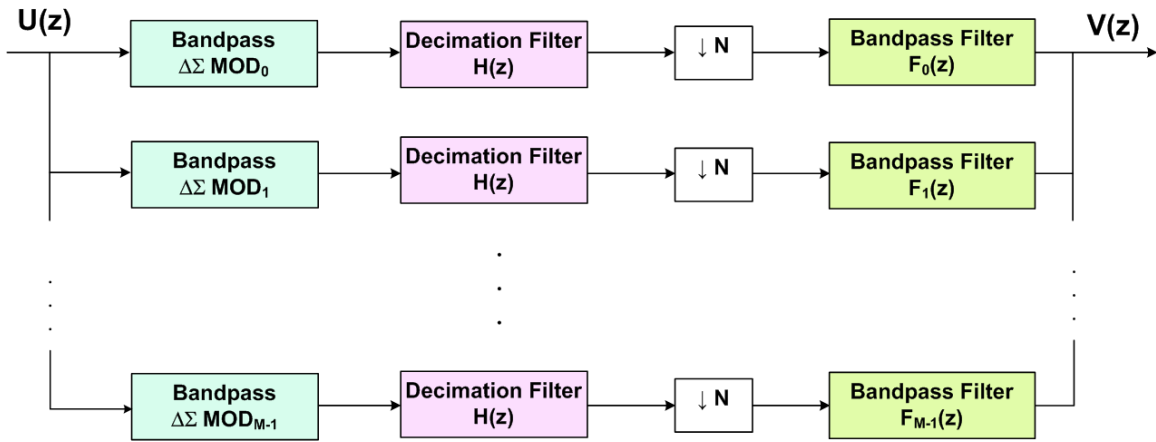


Figure 1.2: Parallel $\Delta\Sigma$ modulator based on Frequency Division Multiplexing (FDM)

Figure 1.3 shows the block diagram of a parallel system based on a CDM scheme, which is also known as a $\Pi - \Delta\Sigma$ modulator. In [22], a *Hadamard* transformation is used to decompose the input into multiple spread spectrum channels by modulating it with different

Hadamard sequences. Each of these channels is then modulated by a DSM, filtered by a decimation filter, and demodulated by a delayed version of the same Hadamard sequences before adding the channels together. There is still very little research on this parallel system for a DAC application.

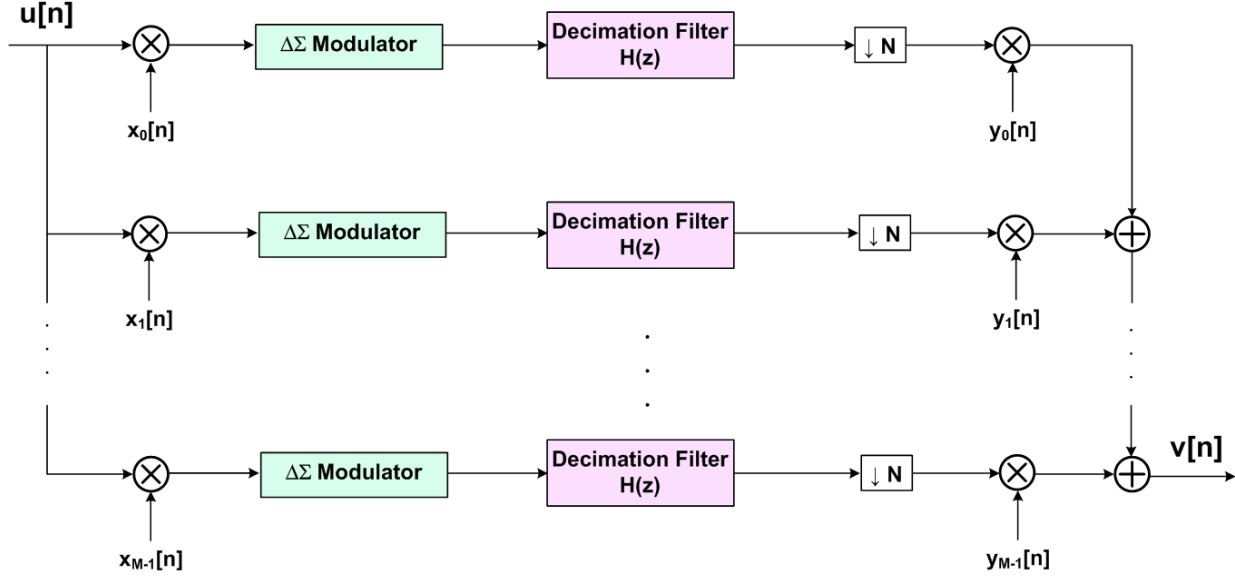


Figure 1.3: Parallel $\Delta\Sigma$ modulator based on Code Division Multiplexing (CDM)

Lastly, figure 1.4 shows the block diagram of a parallel system based on a TDM scheme. Among the three different structures, this is the simplest form of parallelism. Here, the input is demultiplexed to M channels in which each operates at $(1/M)$ of the input sampling frequency. The channels are then recombined through unit delays and a multiplexer. However, this brute-force approach often results in a small SNR improvement. Specifically, there is only a 3dB-improvement in SNR for each doubling of the number of parallel modulators regardless of their order [2].

Alternatively, a novel approach to modify the TDM scheme proposed by Khoini-Poorfard *et al.* has significantly improved the SNR while meeting the low OSR requirement. In [2, 23], a block digital filtering technique was used to successfully transform a conventional DSM into a time-interleaved DSM with interconnecting channels. By having M interconnected channels running in parallel as shown in figure 1.5, the total effective sampling rate becomes M times the sampling rate of each channel. The improvement in SNR is $6(n + 1/2)dB$ for

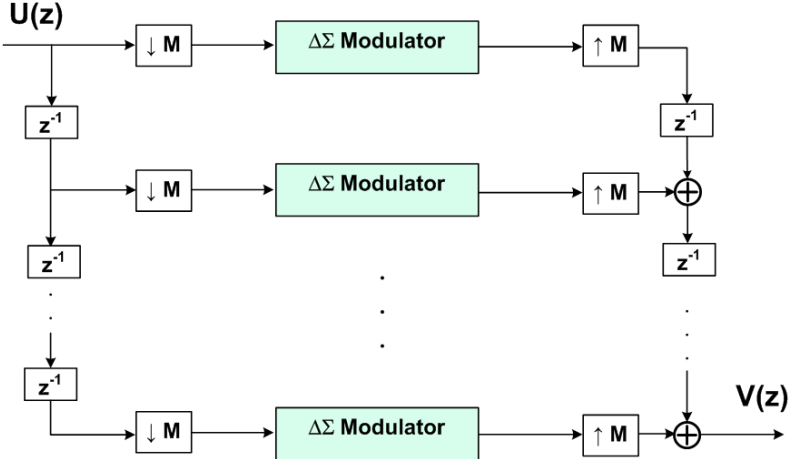


Figure 1.4: Parallel $\Delta\Sigma$ modulator based on Time Division Multiplexing (TDM)

each doubling of the number of n^{th} order modulators [2]. Furthermore, the preceding interpolation filter can be time-interleaved based on a polyphase decomposition without an increase in hardware complexity. Hence, this motivates the choice of time-interleaved DSM architecture for this work. Further details on the time-interleaved $\Delta\Sigma$ -DAC (TIM $\Delta\Sigma$ -DAC) will be discussed later in chapter 2. To distinguish it from the conventional TDM scheme, this modified TDM scheme will be referred as TIM from this point onward.

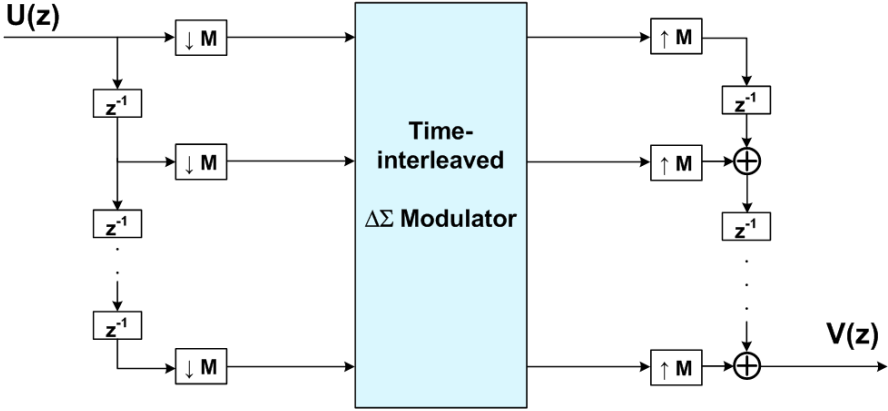


Figure 1.5: TIM $\Delta\Sigma$ modulator based on digital block filtering

1.1.4. Summary

In summary, the motivations for this work are:

- To push $\Delta\Sigma$ -DAC designs to higher speeds to meet the demands of broadband wireless applications and to accommodate the design challenges of deep-sub micron processes.
- To employ a lowpass $\Delta\Sigma$ -DAC design due to its reasonable level of complexity and feasibility.
- To employ a time-interleaved $\Delta\Sigma$ -DAC design to achieve broadband and high resolution performance.

Table 1.1 summarizes the design targets of this work. Due to the speed and system integration requirements, the technology is chosen to be STMicroelectronics 90nm CMOS, 1V supply process.

Table 1.1: $\Delta\Sigma$ -DAC Design Specifications

Design Parameter	Value	Units
Data Rate (f_N)	500	MS/s
Data Bandwidth (f_B)	250	MHz
Oversampling Ratio (OSR)	8	-
Sampling Rate (f_S)	4	GS/s
Resolution ($ENOB$)	9	bits
Power	100 - 120	mW
Modulator Architecture	Lowpass, time-interleaved $\Delta\Sigma$	
Process	ST 90nm CMOS 7M2T, 1V supply	
Applications	UWB or 60GHz WPAN/WLAN	

1.2. State-of-the-Art

As mentioned earlier, some published Nyquist DACs using CMOS technology can meet the required specifications shown in table 1.1. For example, a recently published Nyquist DAC in [11] has a measured resolution of at least 8 bits up to a bandwidth of 193MHz at a sampling rate of 800MS/s while consuming 49mW. In [12] and [14], the bandwidth is up to 250MHz for a sampling rate of 500MS/s and a resolution of 12 and 10 bits while consuming 216mW and 125mW, respectively. Another impressive design in 0.35 μ m CMOS [13] goes beyond the required specifications with a bandwidth of 500MHz for a sampling rate of 1Gs/s and 10-bit resolution while consuming 110mW. For a conventional (i.e.: without parallelism) lowpass $\Delta\Sigma$ -DAC fabricated in CMOS technology, the most relevant work in 0.5 μ m could only achieve 5MHz bandwidth [24].

The idea of TDM parallelism has been introduced for many years but is still uncommon in DAC applications. For instance, the DAC in [18] employs a heterodyne technique to commutate the output of the polyphase interpolation filter into multiple parallel paths. Each path has its own DSM to perform the modulation which is then time-interleaved in the digital domain before feeding into the single DAC. While time-interleaving, the parallel spectra are aligned such that the signal-band experiences coherent gain while the noise-band experiences destructive cancellation. Although only simulation results were presented in [18], it promises a solution for a wideband $\Delta\Sigma$ -DAC, as well as the possibility of a quadrature parallel $\Delta\Sigma$ -DAC.

A design fabricated in 0.13 μ m CMOS utilizing TDM parallelism is presented in [25]. Here, the digital input stream is modulated into a 6-bit digital output through a 3rd-order DSM being oversampled by a factor of 6. There are two parallel channels with each operating on a separate DAC in a time-interleaving manner. This design achieves a signal bandwidth of 29MHz at an oversampling rate of 350MHz and a resolution of almost 12 bits while consuming 62mW. Although this design has an impressive resolution, its signal bandwidth is insufficient for broadband applications.

Similar to the case of conventional $\Delta\Sigma$ modulation, much research effort has been focused on TIM $\Delta\Sigma$ -ADCs while TIM $\Delta\Sigma$ -DACs have been largely overlooked. In [26], a 2nd-order

time-interleaved-by-2 (TIM2) DSM with a single-bit DAC is implemented on an FPGA chip just to show the significant SNR improvement (i.e.: 15dB) over that of a conventional $\Delta\Sigma$ -DAC. In [27], the simulation results for a 2nd-order MASH structure with TIM4 and a 6-bit DAC are presented. Although the four internal channels of each DSM are interconnected, each channel has its own DAC and the recombination is done in the analog domain. This design is prone to two sources of mismatch errors which come from the parallel DACs and from the multistage nature of a MASH system. The simulation results based on a 0.18 μ m CMOS process show a resolution of 12 bits and a signal bandwidth of 40MHz (for an effective sampling rate of 640MS/s).

Although an interpolation filter (IF) has not yet been mentioned here, it is a required block preceding the DSM in an oversampled $\Delta\Sigma$ -DAC. Its purpose is to increase the input sampling frequency (f_N) by an OSR factor and to suppress all unwanted replicas of the signal between baseband and $OSR \cdot f_N$. Aside from a brief mention of a time-interleaved simple zero-order hold interpolator in [26], the design of an IF is omitted in all $\Delta\Sigma$ -DACs described in [24], [25] and [27]. In these designs, the input is interpolated and filtered externally before feeding into the DSM. Unlike the previous works, this thesis integrates a complete design of a time-interleaved interpolation filter (TIM-IF).

Up to date, there is still no fabricated design reported for a TIM $\Delta\Sigma$ -DAC with integrated TIM-IF. This gives an even higher level of motivation to carry out this work.

1.3. Thesis Outline

This thesis focuses on the design and implementation of a time-interleaved $\Delta\Sigma$ -DAC for broadband wireless applications. The flow of the thesis is as follows.

Chapter 2 introduces an overview of $\Delta\Sigma$ -DAC architectures and the theoretical background of $\Delta\Sigma$ modulation for both conventional and time-interleaved structures. The idea of block digital filtering is introduced as a method to transform a conventional IF and DSM into a time-interleaved IF and DSM.

Chapter 3 discusses the system-level design of a $\Delta\Sigma$ -DAC using Matlab based on the previous theoretical background. Specific architectural details on each sub-block are discussed,

namely, the time-interleaved interpolation filter, the time-interleaved $\Delta\Sigma$ modulator, the multi-bit DAC, and the analog reconstruction filter. These sub-blocks are the essential components to form a complete TIM $\Delta\Sigma$ -DAC. System-level simulation results are also presented.

The circuit and physical implementation of this TIM $\Delta\Sigma$ -DAC are described in chapter 4. This design is implemented using STMicroelectronics 90nm CMOS process using Cadence, Synopsis, VHDL, and First Encounter. The integration of this TIM $\Delta\Sigma$ -DAC is divided into 3 parts: a digital baseband front-end, a high-speed digital interface, and a high-speed analog back-end. Circuit-level simulation results are also presented.

Chapter 5 presents the experimental results and some testing issues of the fabricated TIM $\Delta\Sigma$ -DAC. The accuracy and linearity performance of TIM $\Delta\Sigma$ -DAC are measured.

Finally, chapter 6 summarizes the current work and gives suggestions for future work.

Theoretical Background

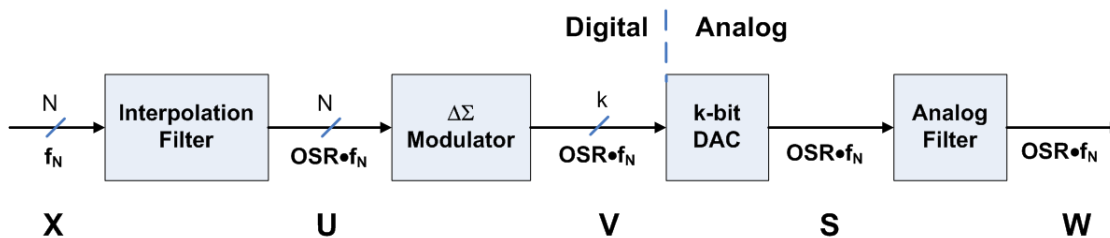
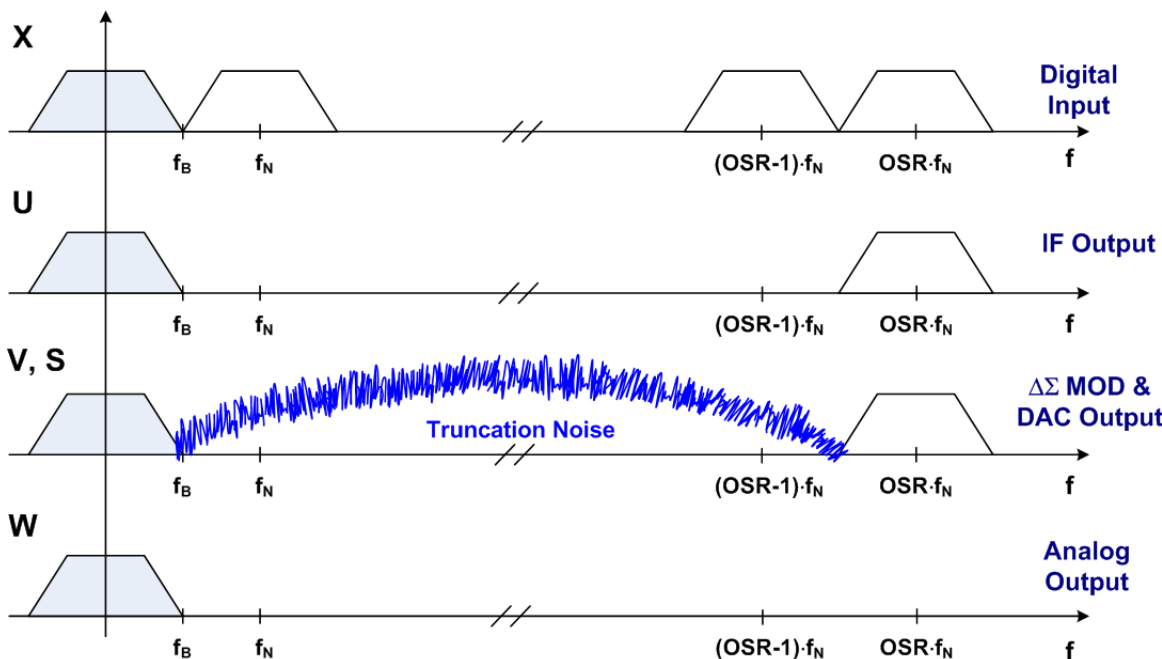
In this chapter, the theoretical background of $\Delta\Sigma$ modulation is introduced. Both conventional and time-interleaved $\Delta\Sigma$ architectures are presented with an emphasis on digital-to-analog converter applications. A block digital filtering technique is also presented here as a method to transform a conventional $\Delta\Sigma$ modulator (DSM) into a time-interleaved $\Delta\Sigma$ modulator (TIM-DSM).

2.1. System Architecture for $\Delta\Sigma$ -DAC

Whereas in $\Delta\Sigma$ -ADCs, the *quantization error* is spectrally shaped, the noise being shaped in $\Delta\Sigma$ -DACs is instead the *truncation error* from the finite word-length of its digital circuitry. Compared with Nyquist-rate DACs, more design emphasis is placed on the digital front-end in $\Delta\Sigma$ -DACs, which allows the use of a relatively robust and simple analog back-end.

Figure 2.1 illustrates the basic architecture of a $\Delta\Sigma$ -DAC. The digital front-end contains the interpolation filter (IF) and the DSM, while the analog back-end contains the multi-bit DAC and the analog reconstruction filter. Figure 2.2 shows the signal spectrum at each internal node of the $\Delta\Sigma$ -DAC. The input signal, \mathbf{x} , is a N-bit digital stream sampled at the Nyquist rate f_N .

The IF serves two purposes: to raise the input frequency (f_N) by an oversampling ratio

Figure 2.1: $\Delta\Sigma$ -DAC block diagramFigure 2.2: Spectrum at each internal node in $\Delta\Sigma$ -DAC [1]

(OSR) and to suppress all unwanted replicas of the signal between baseband and sampling frequency (i.e. $f_S = OSR \cdot f_N$), which arise due to the upsampling. The out-of-band attenuation of the IF improves the dynamic range of the DSM since larger signals can be accommodated. In addition, it reduces the attenuation requirements on the analog filter since only out-of-band truncation noise needs to be suppressed. Finally, the amount of intermodulated out-of-band noise that can fold back into the signal band is reduced, thus relaxing the analog filter linearity requirements.

The DSM truncates the word-length of its signal to k bits where $k < N$. The modulator output contains the input signal, as well as the filtered truncation noise caused by the reduced

word-length. Similar to an analog DSM, an ideal 1-bit modulator would yield an inherently linear DAC. However, it may cause loop instability, as well as make the analog filter's design a challenging task due to the high-frequency content of the high slew-rate output. In contrast, multi-bit modulators improve both loop stability and noise shaping capability by allowing a higher order noise transfer function. Also, they contain less out-of-band noise and lower slew-rate requirements which significantly reduce the complexity of the analog filter. However, additional circuitry is required to correct for the nonlinearity of a multi-bit DAC. Overall, the performance and design benefits outweigh the additional hardware, thus favouring the multi-bit structure in most $\Delta\Sigma$ -DAC, eg: [24, 28, 29, 30, 31].

Ideally, the DAC should produce an analog signal at its output without any distortion. Thus, its output spectrum should be identical to that at the output of the DSM. Finally, the analog reconstruction low-pass-filter should suppress most of the out-of-band noise, leaving only the signal spectrum within the band of interest.

2.2. $\Delta\Sigma$ Modulator Architectures

There are many different DSM architectures for a $\Delta\Sigma$ -DAC. All of the configurations (i.e.: MASH, CIFB, CIFF, CRFB, CRFF, etc) available for $\Delta\Sigma$ -ADCs are also valid for $\Delta\Sigma$ -DACs. Certainly, the components in these configurations are now digital instead of analog; for example, the integrators are replaced by accumulators which use digital adders and multipliers instead of op amps and switched capacitors. Since all signals in the modulator are digital to begin with, there is no need for internal data conversion as in the case of ADCs. The signal processing in the loop can be highly accurate and it is unnecessary to account for any analog imperfections when predicting the loop behaviour. As a consequence, this allows the use of a highly efficient error-feedback (EFB) structure which is impractical in ADCs design. General background details on conventional $\Delta\Sigma$ modulator can be found in Appendix A, while the details on EFB $\Delta\Sigma$ modulator are discussed here.

2.2.1. Error-Feedback $\Delta\Sigma$ Modulator Architecture

The architecture for an Error-Feedback $\Delta\Sigma$ Modulator (EFB-DSM) is shown in figure 2.3. While this architecture is highly efficient for DACs, it is never used in ADCs which are overly sensitive to the imperfections of the analog loop filter $H(z)$ and the analog subtraction needed to generate the quantization error $E(z)$ [1]. Unlike an analog DSM, the quantizer (Q) is now replaced by a truncator (T) in the digital DSM.

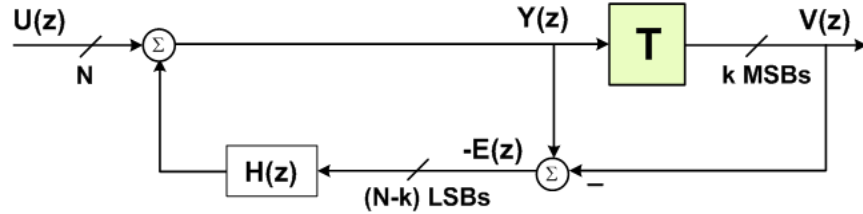


Figure 2.3: Linear model of a single-bit error-feedback $\Delta\Sigma$ modulator

In the EFB structure, instead of feeding back the MSBs of the output $V(z)$, the discarded LSBs (i.e.: the truncation error $E(z)$), are fed back to the input. The digital loop filter, $H(z)$, is now located in the feedback path rather than in the forward path as in the conventional DSM. Linear analysis shows that the transfer function for an EFB-DSM is:

$$V(z) = U(z) + [1 - H(z)]E(z) \quad (2.1)$$

in which $STF(z) = 1$ and $NTF(z) = 1 - H(z)$.

For a first-order modulator, since $NTF(z) = (1 - z^{-1})$ from equation A.1, this results in $H(z) = z^{-1}$, or simply a single digital delay. For a n^{th} -order EFB-DSM, the system can be designed by solving for $H(z)$ from the $NTF(z)$ in equation A.3:

$$\begin{aligned} NTF(z) &= (1 - z^{-1})^n = 1 - H(z) \\ \Rightarrow H(z) &= 1 - NTF(z) = 1 - (1 - z^{-1})^n \end{aligned} \quad (2.2)$$

2.2.2. Error-Feedback $\Delta\Sigma$ Modulator Stability Analysis

The design of a digital DSM encounters similar problems to those of an analog DSM plus some different ones. Instead of dealing with element-matching errors and op amp non-idealities,

the dominant errors in a digital DSM are due to coefficient truncation and round-off errors of the digital operations [1]. Similar to an analog DSM, these can affect the modulator's noise shaping capability. Further details on this topic will be discussed later in chapter 4.

Higher order (i.e.: 3rd-order and higher) EFB-DSM is often chosen to achieve higher in-band noise shaping which directly corresponds to higher ENOB. However, a high order EFB-DSM is prone to suffer from instability when the input to the truncator (i.e: $Y(z)$ in figure 2.3) grows beyond the operating range of the digital number representation.

For signed or unsigned arithmetic, an overflow causes $Y(z)$ to saturate to its largest possible value. However, for 2's complement, overflows cause $Y(z)$ to wrap around, implying the output $V(z)$ suddenly decreases with increasing $Y(z)$. While saturation is usually acceptable, wrap-around causes large errors and must be prevented [1]. Since 2's complement arithmetic operations are generally advantageous, it is critical to resolve the overflow wrap-round problem. By adding a digital limiter before the truncator [32] as shown in figure 2.4, $Y(z)$ will saturate before an overflow can occur.

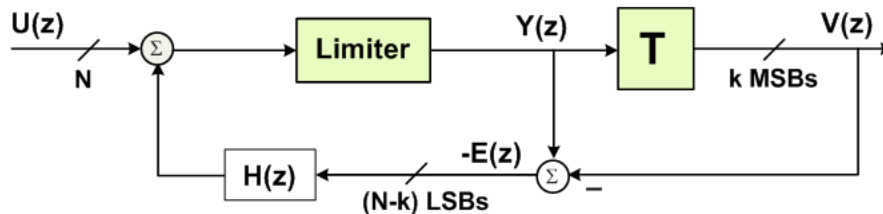


Figure 2.4: Single-bit error-feedback $\Delta\Sigma$ modulator with digital limiter

In addition to the external limiter, certain conditions must be imposed on the truncator input to improve the modulator's robustness and stability. Much research has been focused on improving DSM stability, yet there is not a solid theoretical explanation to predict this behaviour for high-order DSMs. A conservative empirical rule from *Lee's criterion*, which only applies for single-bit modulators, requires the NTF's out-of-band gain (OBG) to be less than 1.5 (i.e.: $\max|NTF(e^{jw})| < 1.5$) [33]. For a multi-bit modulator, a stability condition proposed by Richard Schreier [1] determines how many input truncation levels are needed to keep the DSM stable. The condition states that for any input less than half of the quantizer input range, A , the modulator is guaranteed not to experience overloading (i.e.:

$\max|u(n)| < \mathbf{A}/\mathbf{2} + \mathbf{2}$). While these conditions ensure stability of the DSMs, they dramatically reduce the input dynamic range (DR) and thus, limit the achievable performance of higher order modulators.

A bit-wise stability analysis on EFB-DSM (figure 2.5) in [33] allows higher dynamic range, as well as higher out-of-band gain. In the EFB architecture, since $H(z)$ is an FIR transfer function, there is no need for an accumulator, unlike conventional output-feedback (OFB) topologies. Hence, the word-length at all internal nodes can be predicted without complex numerical analysis.

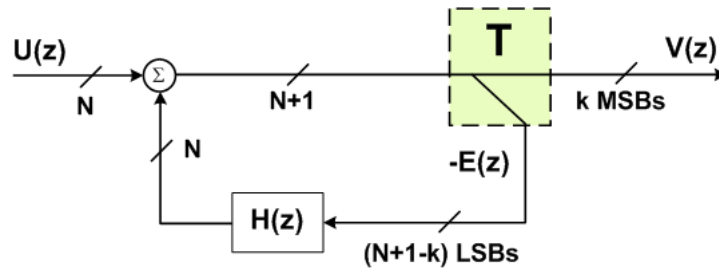


Figure 2.5: Bit-wise analysis of error-feedback $\Delta\Sigma$ modulator

Let $U(z)$ be a digital input stream of word-length N and T be a k -bit truncator. The input summer adds at most 1 bit to give $(N + 1)$ bits to the truncator. Here, the truncation is done by simply splitting k MSB bits to $V(z)$ and feeding back $(N + 1 - k)$ LSB bits to the loop filter $H(z)$. Also, assume that the number of additional bits due to $H(z)$ (i.e.: $n_{H(z)}$) is the same as its order. This is a reasonable assumption since the number of taps $H(z)$ is the same as its order. Hence, in order to keep all internal signals bounded and as long as $H(z)$ is an FIR filter, the number of bits at the output of $H(z)$ can only be at most N and thus:

$$\begin{aligned} (N + 1 - k) + n_{H(z)} &= N \\ \Rightarrow n_{H(z)} &= k - 1 \end{aligned} \quad (2.3)$$

Equation 2.3 implies that in order to have all internal signals bounded, *the order of $H(z)$ must be 1 less than the number of truncating bits*. In other words, the stability criterion is:

An error-feedback modulator with an k -bit truncator and a loop filter of order $(k-1)$ is stable. [33]

The simulation results of an EFB system in [33] based on the above criterion show a superior performance in both stability and signal-to-noise ratio over a conventional OFB system. The k -bit EFB system can tolerate a full-scale out-of-band gain (OBG) of 2^{k-1} while the equivalent OFB system can only tolerate OBG up to approximately 3.5.

The combination of both a limiter and a stability-criterion-based EFB modulator design shown in figure 2.6 ensures that the modulator is stable and robust.

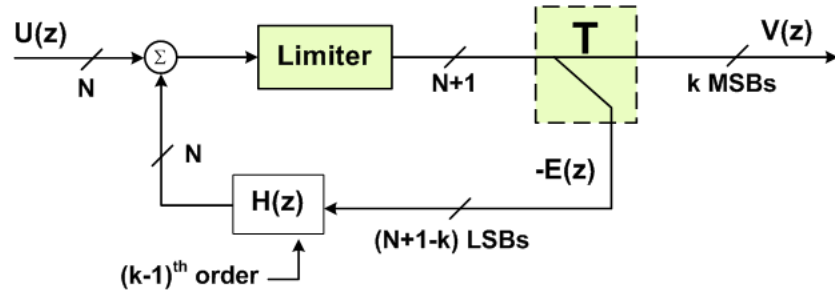


Figure 2.6: Stable error-feedback $\Delta\Sigma$ modulator

2.3. Time-interleaved $\Delta\Sigma$ Modulator

Unlike the TDM structure in figure 1.4, a parallel $\Delta\Sigma$ structure proposed in [2] uses M mutually cross-coupled DSMs, in which each operates in parallel at the same clock rate. This results in a total effective sampling rate of M times the rate of each modulator. The main concept is to use a polyphase decomposition and block digital filtering to transform the single-input single-output (SISO) transfer function to an equivalent $M \times M$ matrix form. Based on this transfer function matrix, along with a commutator at both the front and back ends, an equivalent time-interleaved-by- M architecture can be derived.

2.3.1. Polyphase Decomposition

Polyphase decomposition is very popular in multirate DSP applications such as decimation and interpolation filters, and Discrete Fourier Transform (DFT) filter banks. A detailed polyphase decomposition technique was described in [34] and is summarized in brief here.

Let $H(z) = \sum_{n=-\infty}^{\infty} h(n)z^{-n}$ represent the transfer function of a digital filter which can be rewritten in the form:

$$\begin{aligned} H(z) = & [.. + h(-2)z^2 + h(0) + h(2)z^{-2} + ..] \\ & + z^{-1}[.. + h(-1)z^2 + h(1) + h(3)z^{-2} + ..] \end{aligned} \quad (2.4)$$

Essentially, equation 2.4 groups the impulse-response coefficients $h(n)$ into even samples, $e_0(n) = h(2n)$ and odd samples, $e_1(n) = h(2n + 1)$. If the z -transforms of e_0 and e_1 are $E_0(z)$ and $E_1(z)$, respectively then:

$$E_0(z) = \sum_{n=-\infty}^{\infty} h(2n)z^{-n} \text{ and } E_1(z) = \sum_{n=-\infty}^{\infty} h(2n + 1)z^{-n}$$

Thus, $H(z)$ can be re-expressed as:

$$H(z) = E_0(z^2) + z^{-1}E_1(z^2) \quad (2.5)$$

The quantities $E_0(z)$ and $E_1(z)$ are the polyphase components of $H(z)$ and the representation in 2.5 is called the *two-component* polyphase decomposition of $H(z)$. This decomposition

is valid for the case when $H(z)$ is either a FIR or IIR filter. Also, it is possible to extend $H(z)$ to an M -component polyphase decomposition in the form:

$$H(z) = \sum_{k=0}^{M-1} z^{-k} E_k(z^M) \quad (2.6)$$

where the polyphase components $E_k(z)$ is defined as

$$E_k(z) = \sum_{n=-\infty}^{\infty} h(nM + k) z^{-n}, \quad 0 \leq k \leq M - 1 \quad (2.7)$$

Basically, the impulse-response coefficients $h(n)$ have been divided into M groups and $E_k(z)$ are simply the M -fold decimated sequences of $H(z)$. Here, $H(z)$ is called a *Type 1 polyphase decomposition* and $E_k(z)$ are called a *Type 1 polyphase components*. Type 2 polyphase decomposition is similar to Type 1, except that the components are renumbered:

$$H(z) = \sum_{k=0}^{M-1} z^{-(M-1-k)} R_k(z^M) \quad \text{where} \quad R_k(z) = E_{M-1-k}(z) \quad (2.8)$$

Type 1 and Type 2 decompositions are well suited for the design of decimation and interpolation filters, respectively. Their implementations can be found in full detail in [34].

2.3.2. Block Digital Filtering

A block digital filter is a multirate system where parallelism is used to reduce the processing speed of each element [23]. Consider an SISO linear time-invariant (LTI) system with transfer function $H(z)$ as shown in figure 2.7(a). According to [2], the digital blocked versions of length M for the input $u(n)$ and output $v(n)$ are:

$$\bar{u}(n) = [u_{M-1}(n), u_{M-2}(n), \dots, u_0(n)]^T \quad (2.9)$$

$$\bar{v}(n) = [v_{M-1}(n), v_{M-2}(n), \dots, v_0(n)]^T \quad (2.10)$$

where $u_k(n) = u(nM + k)$ and $v_k(n) = v(nM + k)$ for $(0 \leq k \leq M - 1)$. Equations 2.9 and 2.10 closely resemble the polyphase decomposition form in (2.7). In fact, their components are M -fold decimated versions of $u(n)$ and $v(n)$.

Hence, the z -transform of the two vector-sequences are related by an $M \times M$ transfer matrix $\bar{H}(z)$, i.e.:

$$\bar{V}(z) = \bar{H}(z) \bar{U}(z) \quad (2.11)$$

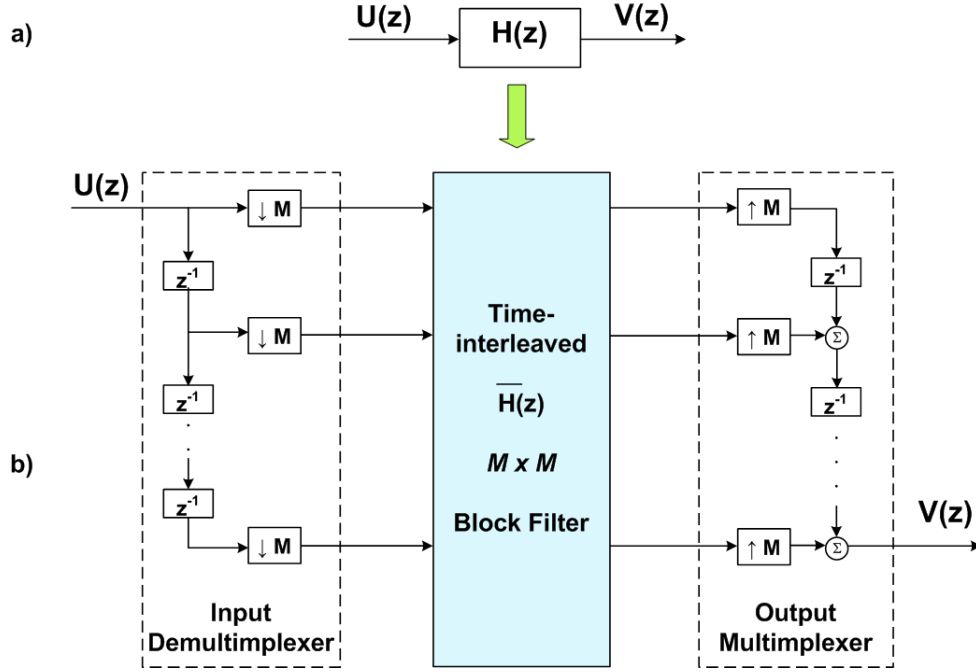


Figure 2.7: (a) Scalar transfer function, (b) Time-interleaved-by-M version [2]

The $M \times M$ matrix $\bar{H}(z)$ is a blocked version of $H(z)$ and its implementation is called *block digital filtering*. Figure 2.7(b) depicts a time-interleaved-by-M version of a scalar system.

From [2], the general structure of $\bar{H}(z)$ is:

$$\bar{H}(z) = \begin{bmatrix} E_0(z) & E_1(z) & E_2(z) & \dots & E_{M-1}(z) \\ z^{-1}E_{M-1}(z) & E_0(z) & E_1(z) & \dots & E_{M-2}(z) \\ z^{-1}E_{M-2}(z) & z^{-1}E_{M-1}(z) & E_0(z) & \dots & E_{M-3}(z) \\ \vdots & \vdots & \vdots & & \vdots \\ z^{-1}E_1(z) & z^{-1}E_2(z) & z^{-1}E_3(z) & \dots & E_0(z) \end{bmatrix} \quad (2.12)$$

$$= \begin{bmatrix} \bar{H}_{11} & \bar{H}_{12} & \bar{H}_{13} & \dots & \bar{H}_{1M} \\ \bar{H}_{21} & \bar{H}_{22} & \bar{H}_{23} & \dots & \bar{H}_{2M} \\ \bar{H}_{31} & \bar{H}_{32} & \bar{H}_{33} & \dots & \bar{H}_{3M} \\ \vdots & \vdots & \vdots & & \vdots \\ \bar{H}_{M1} & \bar{H}_{M2} & \bar{H}_{M3} & \dots & \bar{H}_{MM} \end{bmatrix}$$

The elements in the first row of $\overline{H}(z)$ matrix are indeed the *Type 1 polyphase components* of $H(z)$ as defined in (2.6). Each element \overline{H}_{ij} corresponds to the contribution of the j^{th} input to the i^{th} output. For example, \overline{H}_{12} would correspond to the contribution of the input of path 2 to the output of path 1.

In the matrix $\overline{H}(z)$, each row is a circularly shifted version of the row above it except for the elements below the diagonal entries which also contain a delay. This type of matrix is called *pseudo-circulant*, which is a necessary condition for the block digital filter $\overline{H}(z)$ to represent a SISO linear time-invariant transfer function $H(z)$ [23].

2.3.3. Time-interleaved Error-Feedback $\Delta\Sigma$ Modulator

A detailed realization of a time-interleaved $\Delta\Sigma$ Modulator (TIM-DSM) was proposed in [2, 23]. Another TIM-DSM structure was proposed in [21, 35] that is hardware efficient for more complex structures like CIFB and CIFF. In this work, the chosen $\Delta\Sigma$ modulator architecture is error-feedback (EFB). Since this structure contains no integrators or accumulators as in an OFB structure, the method described in [21, 35] does not result in any improvement over that in [2, 23]. Hence, the time-interleaved (TIM) realization method described in [2, 23] is used in this work. The time-interleaved version of an EFB structure based on block digital filtering is shown in figure 2.8. Unless mentioned otherwise, the term *TIM-DSM* specifically refers to the time-interleaved implementation of an EFB-DSM.

To illustrate the realization of a TIM-DSM, a first-order, single-bit, time-interleaved-by-2 (i.e.: $M=2$) modulator is used as an example. From (2.2), the loop filter for a first-order EFB modulator is $H(z) = z^{-1}$. Using the two-component polyphase decomposition of $H(z)$ from (2.5), it was found that $E_0(z) = 0$ and $E_1(z) = 1$. Thus, by substituting $E_0(z)$ and $E_1(z)$ into (2.12), the matrix blocked version of $H(z)$ is:

$$\overline{H}(z) = \begin{bmatrix} 0 & 1 \\ z^{-1} & 0 \end{bmatrix} \quad (2.13)$$

Since \overline{H}_{ij} corresponds to the contribution of the j^{th} input to the i^{th} output, the architecture of a TIM-DSM can be realized as depicted in figure 2.9. Compared to the equivalent OFB modulator in [2], the EFB structure has a lower level of circuit complexity and requires

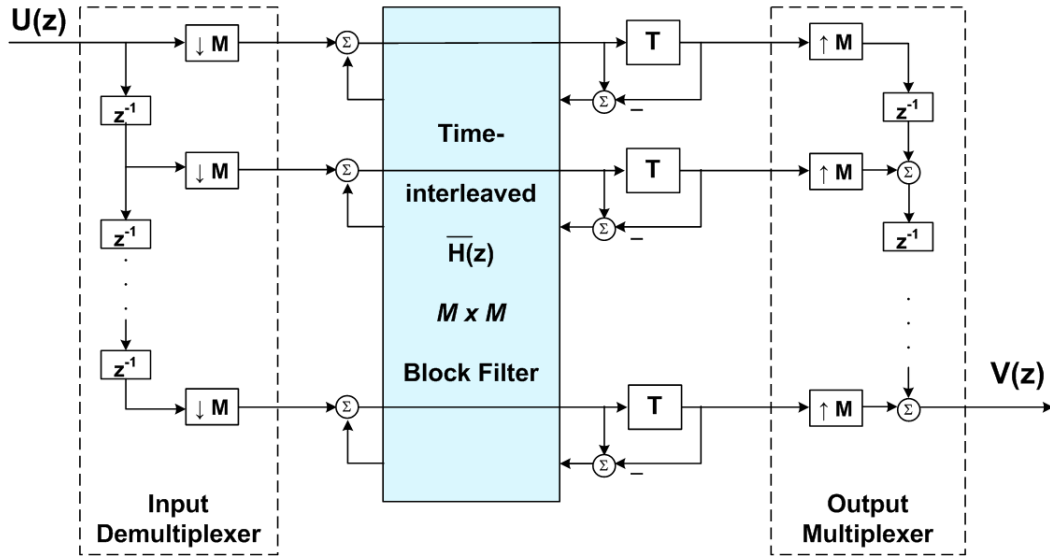


Figure 2.8: Linear model of time-interleaved error-feedback $\Delta\Sigma$ modulator

much less hardware. These savings would become even more significant for a higher order modulator and for a higher time-interleaving factor, M .

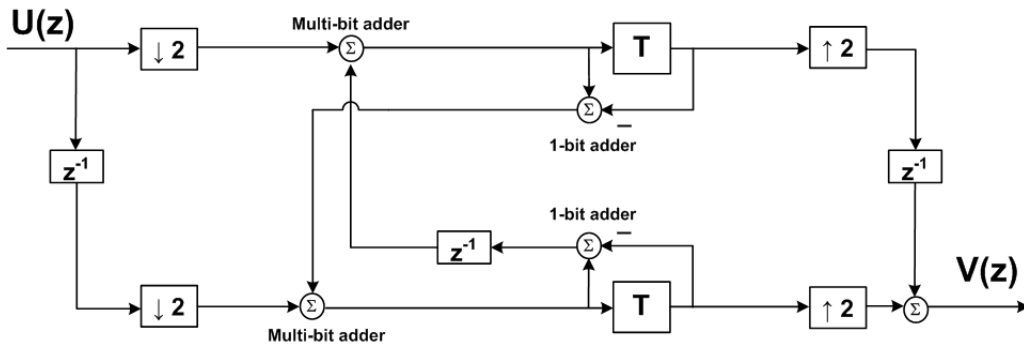


Figure 2.9: First-order time-interleaved-by-two $\Delta\Sigma$ modulator

In summary, the steps to realize the architecture of any TIM-DSM are as follows:

1. Determine $H(z)$ from equation (2.2)
2. Perform M -component polyphase decomposition of $H(z)$ using equation (2.6) and (2.7)
3. Substitute $E_k(z)$ into equation (2.12) to find the $M \times M$ matrix $\overline{H}(z)$
4. Use the relations \overline{H}_{ij} to realize the feedback filters in a TIM architecture

2.3.4. Practical Considerations

A main issue to be considered in the design of a TIM-DSM is the critical path. In the previous example, the critical path consists of two multi-bit adders and two one-bit adders, whereas in the non-TIM case, the critical path contains half the number of components. Clearly, if the critical path slows down the TIM-DSM by a factor of 2, this would totally defeat the purpose of time-interleaving in the first place. However, clever design techniques can eliminate this problem to a great extent.

One approach to reduce the critical path was proposed in [36]; it uses a vector quantizer to form two parallel matrix transformations that move intensive computations outside the feedback loop. This technique has a high level of circuit complexity and requires a large amount of hardware. Also, the number of unique comparisons required in the decision circuitry grows exponentially with the time-interleaving factor which practically limits the level of parallelism. Thus, for a design that requires a large number of parallel paths, this technique may not result in a significant critical path delay improvement.

Another approach is to pipeline the adders such that each adder operates as soon as it receives the LSB. The timing analysis in [2] shows that when using ripple-carry adders, the delay in this example is about 12% higher than that of a non-TIM case. The critical timing is expected to improve for even faster adder architectures (e.g.: carry-save, carry-select, or carry-lookahead adders) as long as they are realized in a pipelined fashion.

In this work, both pipelined carry-save and carry-select adders are suitable with comparable critical paths. To add multiple numbers together, a carry-select adder tree results in a shorter tree depth than that of a carry-save adder. Also, at the last stage of carry-save adder tree, a ripple-carry adder is required, which can result in a long critical path for a large word length (> 10 bits). A hybrid solution is to use carry-save adders throughout the sum tree, except for the last stage where carry-select adder is used. For simplicity, only pipelined carry-select adders are used in this work.

2.4. Time-interleaved Interpolation Filter

As mentioned earlier, an interpolation filter (IF) is required before the DSM. It consists of an OSR-fold interpolator and a lowpass filter with a band edge of (π/OSR) as shown in figure 2.10. The interpolator increases the input sampling frequency by a factor of OSR by inserting $(OSR - 1)$ zeros between adjacent samples of $X(z)$. The subsequent filter $G(z)$ eliminates all unwanted images of the input signal.

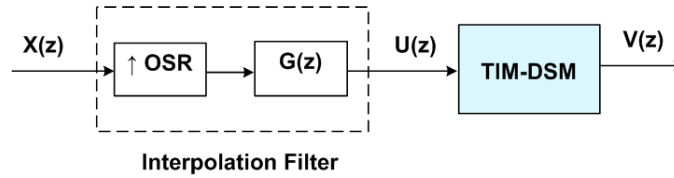
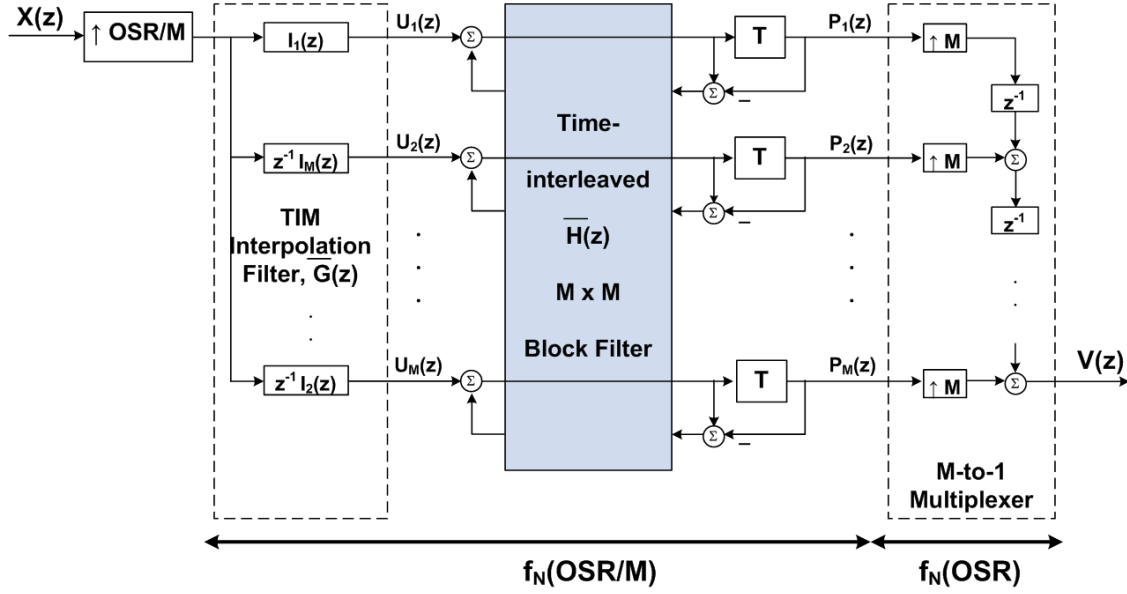


Figure 2.10: Conventional interpolation filter

For a conventional IF followed by a TIM-DSM, the input is first upsampled by the interpolator then downsampled by the DSM's input demultiplexer. By applying the same digital block filtering technique on $G(z)$, the IF can also be transformed into a time-interleaved IF (TIM-IF). If the time-interleaving factor for the IF is same as that of the DSM (i.e.: M), no multiplexer/demultiplexer is needed between them. Furthermore, if M is chosen to be the same as OSR, the upsampler preceding the IF is also eliminated. Figure 2.11 shows the TIM-IF integrated together with the TIM-DSM. Here, all sub-blocks operate at a sampling rate of $f_N \cdot (\frac{OSR}{M})$, except the output multiplexer which operates at a sampling rate of f_S (i.e.: $f_N \cdot OSR$).

Note that the upsampler preceding $G(z)$ in a conventional IF ensures that only one of every M inputs is nonzero. This simplifies the $\overline{G}(z)$ of a TIM-IF from an $M \times M$ matrix down to an $M \times 1$ matrix. In other words, this reduces the required number of elements of $\overline{G}(z)$ from M^2 down to M . Thus, only the first column of $\overline{G}(z)$ is implemented, resulting in a TIM-IF having the same complexity as a conventional IF but operating at $(1/M)^{th}$ the rate [2]. It should also be noted that unlike the case of a TIM-DSM where the internal paths are interconnected, the paths of TIM-IF are independent of each other. Using the same steps to

Figure 2.11: Time-interleaved interpolation filter and time-interleaved $\Delta\Sigma$ modulator

realize a TIM structure from the previous section, an $M \times 1$ matrix for a TIM-IF is given by:

$$\bar{G}(z) = \begin{bmatrix} I_0(z) \\ z^{-1}I_{M-1}(z) \\ z^{-1}I_{M-2}(z) \\ \vdots \\ z^{-1}I_1(z) \end{bmatrix} \quad (2.14)$$

where the polyphase components $I_k(z)$ is defined as:

$$I_k(z) = \sum_{n=-\infty}^{\infty} g(nM + k)z^{-n}, \quad 0 \leq k \leq M - 1 \quad (2.15)$$

2.5. Summary

In general, this chapter gave a brief overview of $\Delta\Sigma$ -DAC architectures. Particularly, the TIM $\Delta\Sigma$ -DAC is of great interest since it combines the well-known benefits of a $\Delta\Sigma$ modulator and the potential for wider bandwidth of a parallel structure. The IF can also be time-interleaved to simplify the overall integrated TIM $\Delta\Sigma$ -DAC design while resulting in no additional hardware complexity.

Time-interleaved $\Delta\Sigma$ -DAC Design

This chapter discusses the architectural design of a time-interleaved (TIM) $\Delta\Sigma$ -DAC. The digital front-end of a TIM $\Delta\Sigma$ -DAC contains a time-interleaved interpolation filter (TIM-IF) and a time-interleaved $\Delta\Sigma$ modulator (TIM-DSM). The analog back-end of a TIM $\Delta\Sigma$ -DAC contains a DAC and an analog reconstruction filter. Specific details of these sub-blocks are discussed in the order of which they appear in the system.

3.1. Architecture Overview

As mentioned earlier, the motivation of this work is to push $\Delta\Sigma$ -DAC to higher speeds and to accommodate the design challenges of deep sub-micron processes. The core of this design is based on a time-interleaving architecture to meet the high data rate, wide bandwidth requirements of 60GHz or UWB applications.

From table 1.1, since the design targets for an ENOB around 9 bits, this corresponds to an accuracy (SNR) and linearity (SFDR) performance of approximately 56 dB. Note that SNR is the ratio of the fundamental signal power to the inband noise power, but it does not account for harmonic distortion. The parameter that accounts for both noise and distortion is called the SNDR (Signal to Noise plus Distortion Ratio), which is often less than that of SNR. The design targets in table 1.1 are conservative but the top-level design aims for even

higher performance to give extra margins.

From chapter 2, if the time-interleaving factor M is same as the OSR then there is no need for an input interpolator. Hence, both M and the OSR are chosen to be 8, which is reasonable in terms of hardware complexity and digital circuit speed as will be shown later in chapter 4. Figure 3.1 shows the block diagram of a time-interleaved-by-8 (TIM8) $\Delta\Sigma$ -DAC. Here, the digital front-end operates at $f_N \cdot (OSR/M) = f_N$ while the analog back-end operates at $f_N \cdot (OSR) = f_S$, which correspond to 500MS/s and 4GS/s, respectively.

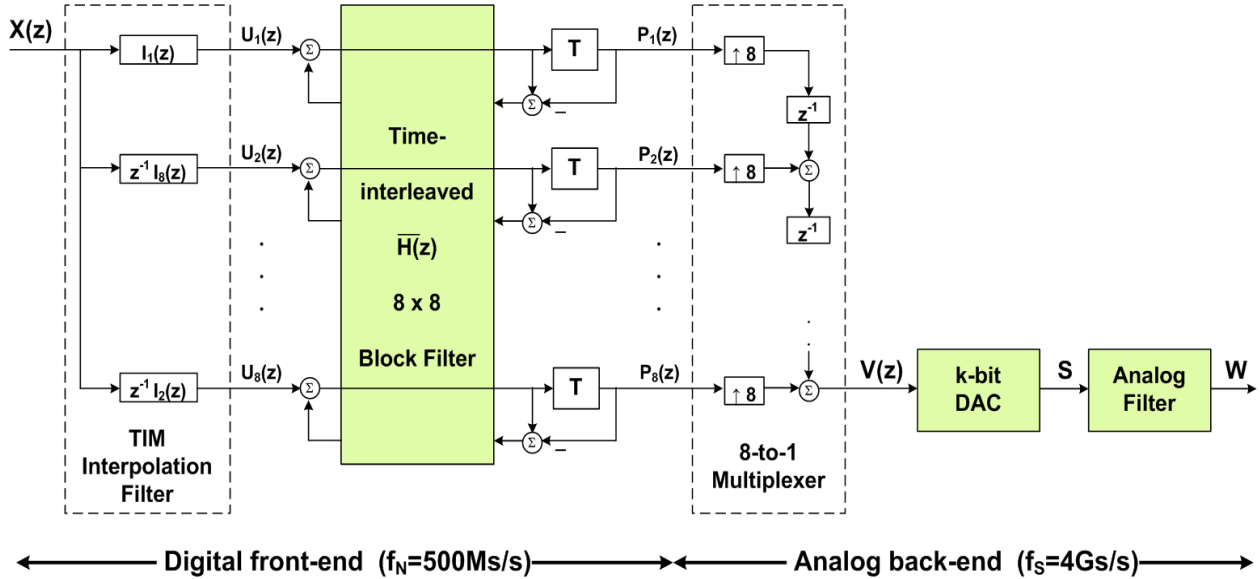


Figure 3.1: Time-interleaved-by-8 $\Delta\Sigma$ -DAC block diagram

3.2. Time-interleaved Interpolation Filter

Based on a built-in digital filter function and Filter Visualization Tool in Matlab, a multirate filter was designed. From the specifications in table 1.1, the interpolation filter, $G(z)$, is required to have an interpolation factor of 8 and a bandwidth of 250MHz.

In a practical design, an ideal or “brick wall” filter is not implementable since its impulse response is infinite and non-causal. To create a finite-duration impulse response, this filter is truncated by applying a window. By retaining the central section of the impulse response, a linear phase finite impulse response (FIR) filter can be obtained. There are different

types of windowing (e.g.: Kaiser, Blackman-Harris, Hamming, Gaussian, etc) which have different trade-offs depending on the design. A long polyphase FIR length gives a high cutoff frequency (i.e.: wide bandwidth) and high attenuation but also has a high implementation complexity. In this application, Kaiser windowing (with $\alpha=0.5$ by default) gives the optimal trade-offs in terms of bandwidth, attenuation and complexity.

To design an FIR interpolation filter (IF), Kaiser windowing requires a polyphase length (pl) and a stopband attenuation (α_s , in dB). The IF cutoff becomes sharper with higher pl to a point when pl is large enough such that increasing it further only results in a small improvement. Note that only when $pl \rightarrow \infty$ does the IF become ideal. In addition, large pl results in an impractical implementation due to the large number of coefficients (i.e.: filter polyphase terms or filter taps). Through simulations, pl is chosen to be 96, which corresponds to a 95th order FIR filter. On the other hand, increasing α_s gives higher out-of-band attenuation hence reducing the analog filter's attenuation requirement. However, this significantly reduces the IF roll-off rate. Since large out-of-band truncation noise is added by the DSM after the IF, having a large α_s does not give a significant benefit. Thus, $\alpha_s = 40dB$ is found to be sufficient for this design.

Figure 3.2 shows different responses of the IF with and without quantization of the 96 coefficients. In simulations, the IF coefficients are first obtained from the “windowed” impulse response with full-precision. In an actual implementation, these coefficients are rounded-off due to the fixed-length multipliers. For a large digital system, multipliers occupy large area and slow down the operating speed. However, if the coefficients are quantized using canonic sign digit (CSD) representation where they are represented as sums or differences of power-of-2, only adders and subtractors are required. This eliminates the need for digital multipliers and ultimately results in an effective and robust digital implementation as long as the discrepancies are reasonably acceptable. In this work, the quantization algorithm allocates one and two CSD terms for coefficients with magnitude $< |0.1|$ and $> |0.1|$, respectively. In depth details on the multiplierless IF hardware implementation will be discussed in chapter 4.

In these responses, there is little discrepancy between the ideal and quantized IF in the passband. Outside the passband, especially after $2 \cdot f_B$, the attenuation of the quantized

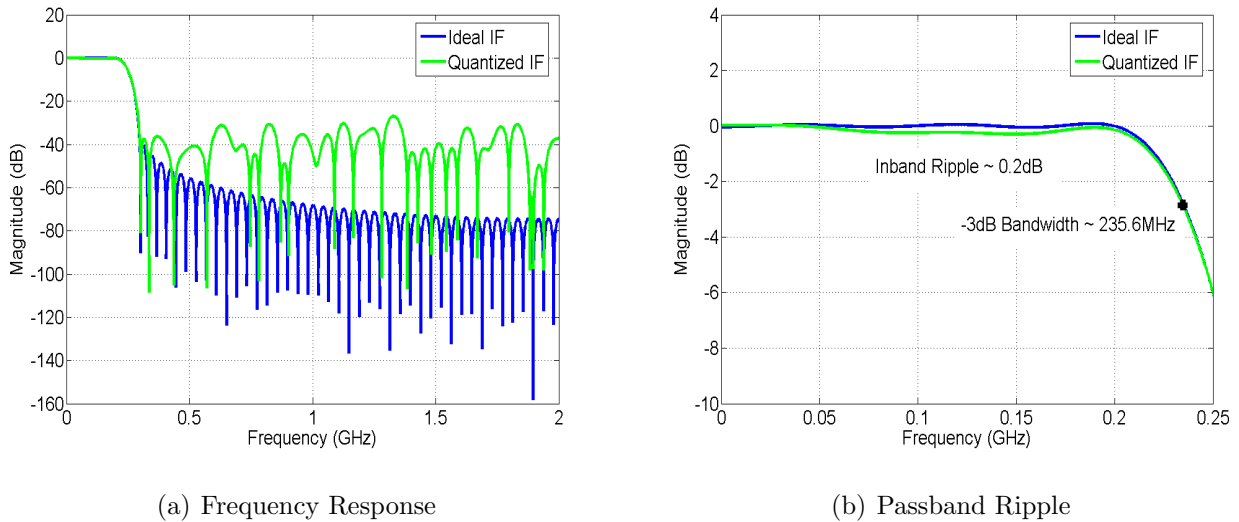


Figure 3.2: A 95th-order FIR interpolation filter with and without coefficient quantization

IF degrades significantly. However, this is acceptable since within the critical band (f_B to $2 \cdot f_B$), the attenuation is still around 40dB as intended. After this band, the truncation noise becomes dominant hence a reduction in stopband attenuation does not cause too much damage. Nevertheless, the quantized IF attenuates all images by at least 28dB over the entire stopband which still helps relaxing the analog filter's attenuation requirement.

From figure 3.2(b), the -3dB bandwidth is around 235MHz. The passband ripple is approximately 0.2dB, which is quite acceptable. Table 3.1 summarizes the IF design method as well as its performance.

Table 3.1: Interpolation Filter Characteristics

	Parameter	Description
Design	Window	Kaiser
	Polyphase length (pl)	96
	Filter order (l)	95
	Stopband attenuation (α_s)	40 dB
Performance (Quantized IF)	-3dB Bandwidth (BW_{-3dB})	235 MHz
	Passband ripple	0.2 dB
	Stopband attenuation	≥ 28 dB

Thus, the IF is a 95th order FIR filter, $G(z)$, of the following form:

$$G(z) = \sum_{k=0}^{l=95} z^{-k} g(k) = g(0) + g(1)z^{-1} + g(2)z^{-2} + \dots + g(n)z^{-l} \quad (3.1)$$

where $g(k)$ are the filter coefficients.

From section 2.3, the 8-component polyphase decomposition of $G(z)$ is expressed as:

$$G(z) = \sum_{k=0}^7 z^{-k} I_k(z^8) = I_0(z^8) + z^{-1}I_1(z^8) + \dots + z^{-7}I_7(z^8) \quad (3.2)$$

where the polyphase components $I_k(z)$ are defined as:

$$I_k(z) = \sum_{i=0}^{\frac{l+1}{8}-1} g(8i+k)z^{-(8i)}, \quad 0 \leq k \leq 7 \quad (3.3)$$

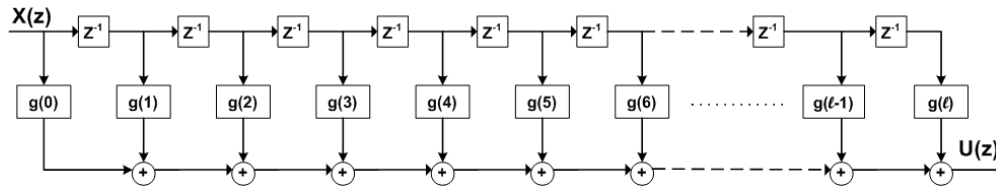
That is:

$$\begin{aligned} I_0(z) &= g(0) + g(8)z^{-8} + g(16)z^{-16} + \dots + g(88)z^{-88} \\ I_1(z) &= g(1) + g(9)z^{-8} + g(17)z^{-16} + \dots + g(89)z^{-88} \\ &\quad \vdots \\ I_7(z) &= g(7) + g(15)z^{-8} + g(23)z^{-16} + \dots + g(95)z^{-88} \end{aligned}$$

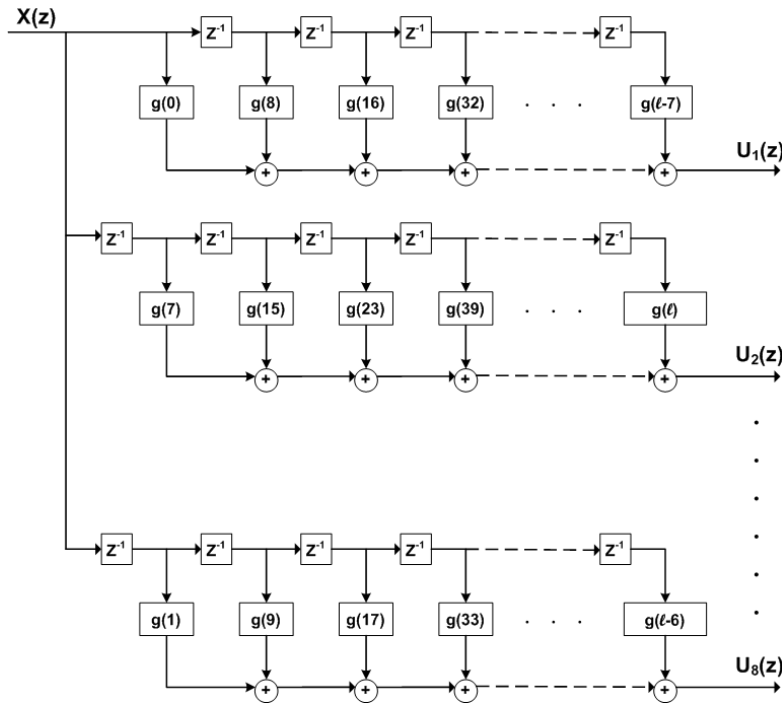
Based on $I_k(z)$ and equation 2.14 from section 2.4, $\overline{G}(z)$ of a TIM-IF is given by:

$$\overline{G}(z) = \begin{bmatrix} I_0(z) \\ z^{-1}I_7(z) \\ z^{-1}I_6(z) \\ \vdots \\ z^{-1}I_1(z) \end{bmatrix} \quad (3.4)$$

Figure 3.2 shows the realization of a TIM-IF based on the conventional IF for this work. Notice that aside from the first path (U_1), all subsequent paths ($U_2 - U_8$) are in reverse order of the polyphase components ($I_7(z) - I_1(z)$).



(a) Conventional IF



(b) Time-interleaved-by-8 IF

Figure 3.3: A 95th-order FIR interpolation filter block diagram

3.3. Time-interleaved $\Delta\Sigma$ Modulator

3.3.1. DSM Architecture

As mentioned in section 2.2.1, the main parameters that control the SNR performance are: the OSR, modulator order (m) and number of truncator bits (k). Based on these parameters, the maximum SNR can be estimated according to the following calculations.

Let the input signal be a sinusoidal wave. Its full-swing amplitude, A , is defined as

$2^k(\Delta/2)$ where Δ is the unit quantization step size (or 1 LSB - least significant bit) and k is the number of truncator bits. Hence, the signal power, P_s , is given by ([37], Ch.14):

$$P_s = \frac{A^2}{2} = \left(\frac{2^k \Delta}{2\sqrt{2}} \right)^2 = \frac{2^{2k} \Delta^2}{8} \quad (3.5)$$

The noise power, P_e , for an m^{th} -order DSM is given by:

$$P_e = \left(\frac{\Delta^2}{12} \right) \left(\frac{\pi^{2m}}{2m+1} \right) \left(\frac{1}{OSR^{2m+1}} \right) \quad (3.6)$$

Thus, the maximum SNR (in dB) is given by:

$$SNR_{max} = 10 \log \left(\frac{P_s}{P_e} \right) = 10 \log \left[\frac{3(2m+1)2^{2k-1}}{\pi^{2m}} (OSR)^{2m+1} \right] \quad (3.7)$$

According to table 1.1, the OSR is 8. This leaves only m and k to be determined. From the stability analysis of an EFB-DSM in section 2.2, the modulator order should be at least 1 less than the number of truncator bits (i.e.: $m \leq k - 1$). Based on equation 3.7, choosing $m=3$ and $k=4$ results in an SNR of 68dB which allows sufficient design margin beyond the target of 56dB.

3.3.2. NTF Optimization

For a 3^{rd} -order DSM, the conventional noise transfer function (NTF) is:

$$NTF(z)_{conv} = (1 - z^{-1})^3 \quad (3.8)$$

in which, all zeros and poles are located at $z=1$ and $z=0$, respectively.

According to ([1], Ch. 4), significant improvement in SNR can be obtained by optimizing the NTF zero locations. By spreading the zeros along the z -domain unit circle, the total inband noise power can be reduced. The optimal NTF zeros can be found by equating the partial derivatives of the noise power to zero. The mathematical derivations are not discussed here and the optimization is done using a built-in function in Richard Schreier's Delta-Sigma Toolbox [38].

Although, moving the poles closer to the zeros reduces the out-of-band (OBG) gain results in improved stability, this was not done here. As discussed in section 2.2, using stability

criterion in [33], the stability of an EFB system can be maintained while tolerating much higher OBG than an OFB system. Thus, there was no need for NTF pole optimization.

Figure 3.4(a) shows the pole-zero plot of the optimized NTF for a 3^{rd} -order DSM. Optimizing the NTF zeros results in a notch at DC and another one at $\sqrt{\frac{3}{5}} \cdot f_B$ or $\frac{\sqrt{3/5} \cdot f_s}{2 \cdot OSR}$ ([1], Ch. 4). This improves the SNR by 8dB compared to the case where all zeros are at DC. Similar to the TIM-IF, the NTF coefficients must be quantized using CSD representation for digital realization. Since there are only 3 taps for a 3^{rd} -order DSM, large discrepancies between quantized and non-quantized coefficients degrade the SNR performance. Hence, the quantized NTF coefficients should be close to the optimized NTF value by utilizing more CSD terms. Here, they are represented by 3 CSD terms as given below.

Thus, the optimized NTF becomes:

$$NTF_{opt}(z) = (1 - z^{-1})(1 - 1.908z^{-1} + z^{-2}) \quad (3.9)$$

and the quantized optimized NTF is:

$$NTF_{quan}(z) = (1 - z^{-1})(1 - 1.875z^{-1} + z^{-2}) \quad (3.10)$$

For this NTF(z), the feedback loop filter, $H(z)$, is:

$$H(z) = 1 - NTF_{quan}(z) = az^{-1} - az^{-2} + z^{-3} \quad (3.11)$$

where $a = 2.875 = 2^1 + 2^0 - 2^{-3}$.

Figure 3.4(b) overlays the response of all NTF versions. The quantization results in a slight degradation of inband noise shaping and a shift in notch location closer to the band edge. However, these have a small impact on the SNR performance which will be quantified in a later section.

Figure 3.5 shows the Matlab model of a 3^{rd} -order $\Delta\Sigma$ -DAC with conventional IF and conventional DSM. Here, the 10-bit quantizer at the input generates a 10-bit digital stream while the 4-bit quantizer near the DAC represents the 4-bit truncator with digital limiter.

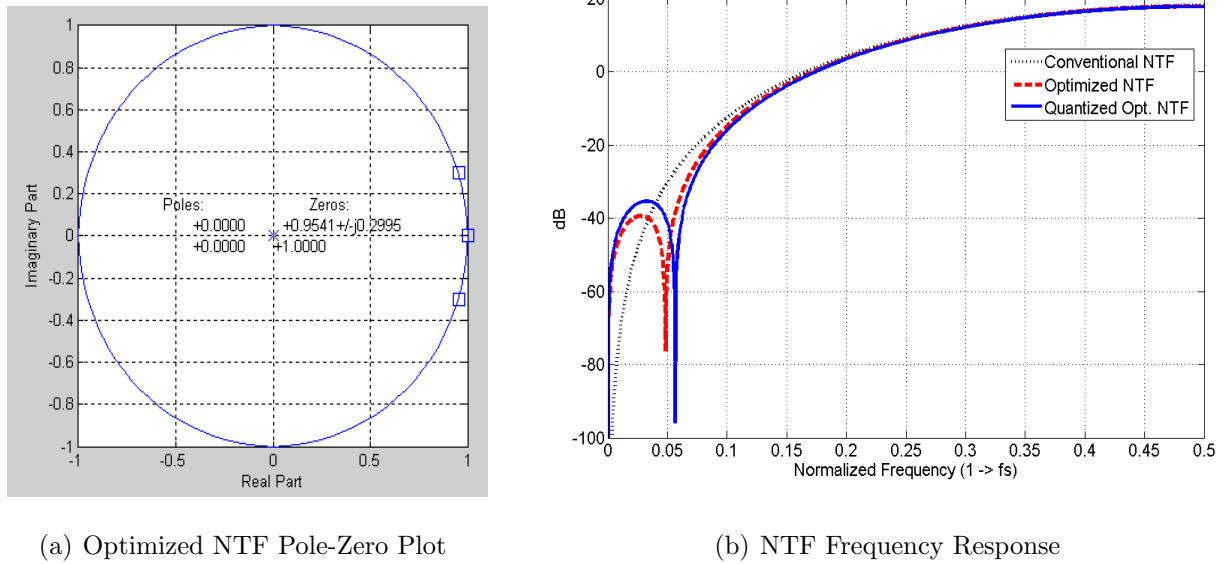


Figure 3.4: $\Delta\Sigma$ modulator noise transfer function optimization

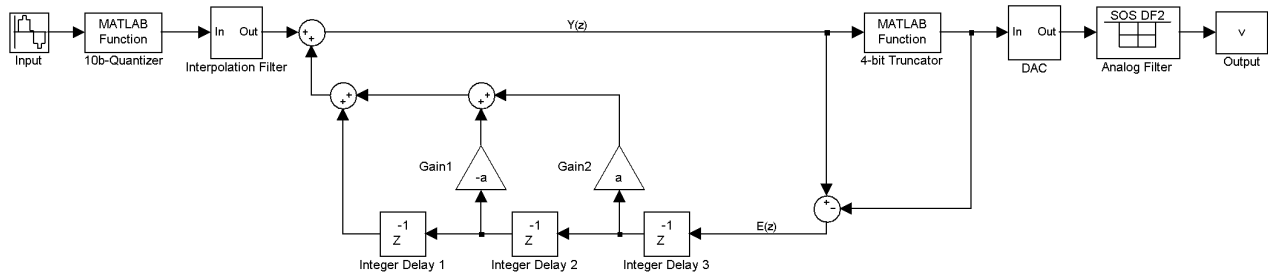


Figure 3.5: Conventional 3rd-order error-feedback $\Delta\Sigma$ modulator architecture

3.3.3. Time-interleaved DSM

Using the steps from section 2.3, a conventional DSM can be transformed into a TIM-DSM. Similar to the TIM-IF, the 8-component polyphase decomposition of $H(z)$ from 3.11 is:

$$H(z) = \sum_{k=0}^7 z^{-k} E_k(z^8) = E_0(z^8) + z^{-1} E_1(z^8) + \dots + z^{-7} E_7(z^8) \quad (3.12)$$

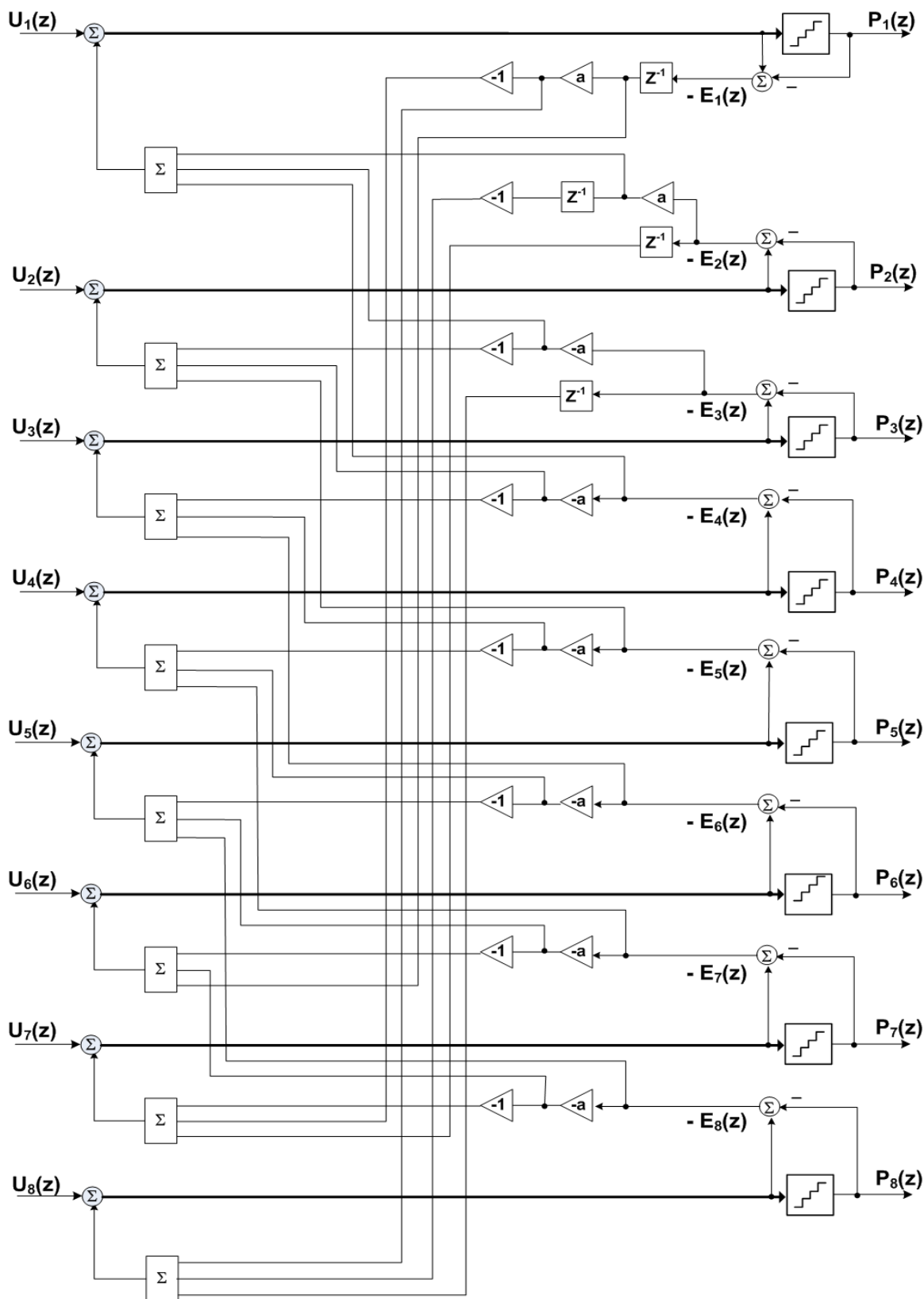
where the polyphase components $E_k(z)$ are defined as:

$$\begin{aligned}
 E_0(z) &= 0 & E_4(z) &= 0 \\
 E_1(z) &= a & E_5(z) &= 0 \\
 E_2(z) &= -a & E_6(z) &= 0 \\
 E_3(z) &= 1 & E_7(z) &= 0
 \end{aligned}
 \tag{3.13}$$

Next, substitute the above polyphase components $E_k(z)$ into equation (2.12) to get:

$$\begin{aligned}
 \overline{H}(z) &= \begin{bmatrix} E_0(z) & E_1(z) & E_2(z) & \dots & E_7(z) \\ z^{-1}E_7(z) & E_0(z) & E_1(z) & \dots & E_6(z) \\ z^{-1}E_6(z) & z^{-1}E_7(z) & E_0(z) & \dots & E_5(z) \\ \vdots & \vdots & \vdots & & \vdots \\ z^{-1}E_1(z) & z^{-1}E_2(z) & z^{-1}E_3(z) & \dots & E_0(z) \end{bmatrix} \\
 &= \begin{bmatrix} 0 & a & -a & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & a & -a & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & a & -a & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & a & -a & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & a & -a & 1 \\ z^{-1} & 0 & 0 & 0 & 0 & 0 & a & -a \\ -az^{-1} & z^{-1} & 0 & 0 & 0 & 0 & 0 & a \\ az^{-1} & -az^{-1} & z^{-1} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}
 \end{aligned}
 \tag{3.14}$$

Lastly, using the relation \overline{H}_{ij} which corresponds to the contribution of the j^{th} input to the i^{th} output, the architecture of a TIM-DSM for this work can be realized as depicted in figure 3.6.

Figure 3.6: Time-interleaved-by-8 3^{rd} -order error feedback $\Delta\Sigma$ modulator

3.3.4. TIM-DSM Performance

This section presents the architectural simulation results for the digital front-end which contains both TIM-IF and TIM-DSM. While the TIM-IF corresponds to a 95th-order FIR interpolation filter, the TIM-DSM corresponds to a 3rd-order, 4-bit $\Delta\Sigma$ modulator. The results are obtained at the output of the 8-to-1 multiplexer (i.e.: $V(z)$ in figure 3.1) and assumed to be filtered by a “brick wall” lowpass filter. The coefficients of both TIM-IF and TIM-DSM are quantized using CSD representation as discussed earlier. For simplicity, the term “TIM-IF-DSM” refers to the integration of the TIM-IF and TIM-DSM, excluding the DAC. Recall that since the OSR equals 8, this corresponds to $f_B = \frac{1}{16}f_S = 250MHz$.

Figure 3.7 shows the response of a time-interleaved DSM (in figure 3.6) versus a conventional DSM (in figure 3.5). Figure 3.7(a) shows the SNR versus input amplitude for a tone at $0.25f_B$. Figure 3.7(b) shows the SNR versus input frequency (normalized to f_B) at 0dBFS amplitude. These figures show identical responses, implying that the time-interleaved system is indeed equivalent to the conventional one.

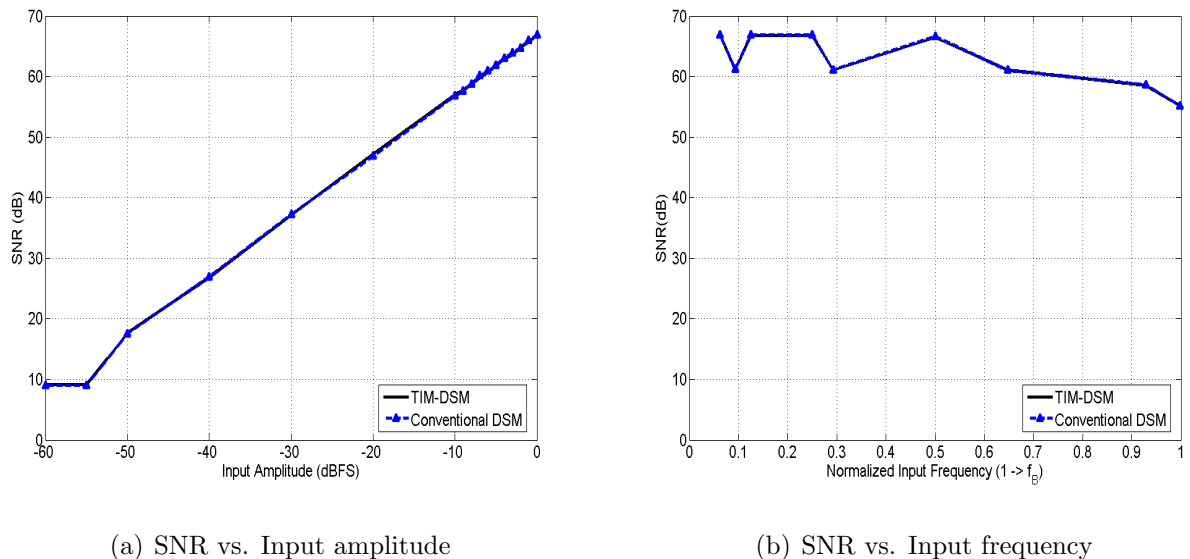


Figure 3.7: TIM-IF-DSM versus conventional DSM response (Matlab simulations)

Figure 3.8 shows the TIM-IF-DSM output SNR and SNDR versus input amplitude for a single tone at $0.25f_B$ for non-optimized (NTF), optimized (NTF_{opt}) and quantized

(NTF_{quan}) optimized NTF. In these simulations, the input is quantized to 10 bits, but internal computations are performed with full precision even when the TIM-IF and TIM-DSM coefficients are quantized. As discussed earlier, this figure shows an SNR improvement of 8dB between NTF and NTF_{opt} . Compared to the NTF_{opt} , the NTF_{quan} shows a 2dB in SNR degradation but less than 1dB in SNDR degradation. This implies the NTF_{quan} is an acceptable design.

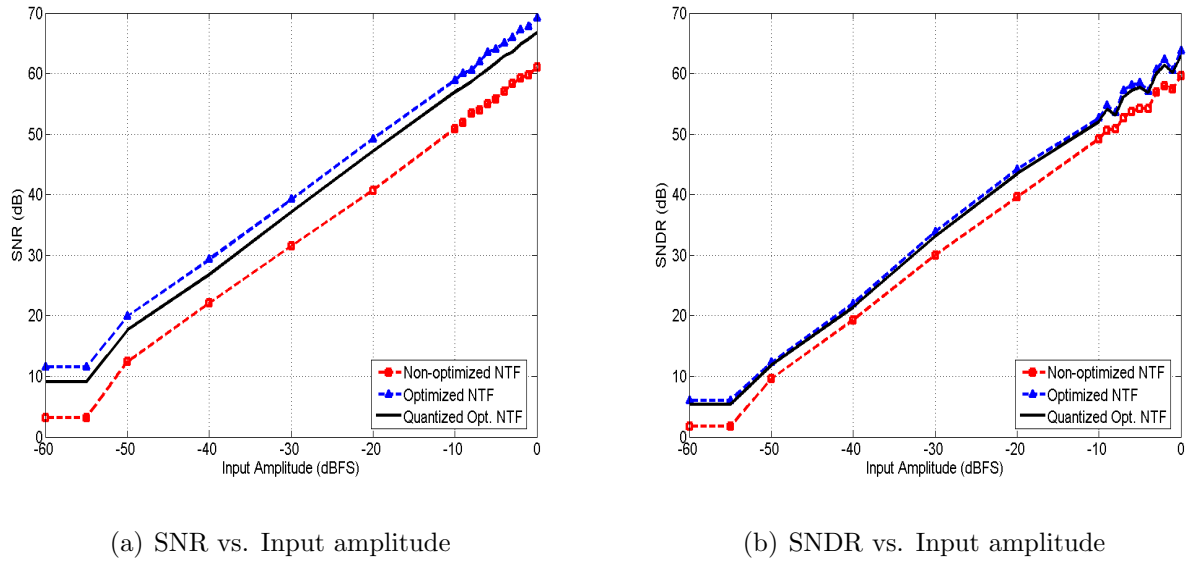


Figure 3.8: TIM-IF-DSM performance for a single tone at $0.25f_B$ for non-optimized, optimized and quantized optimized NTF (Matlab simulations)

Figure 3.9 shows the TIM-IF-DSM output spectrum for different input frequencies, ranging from $0.13f_B$ to $0.93f_B$. For input frequencies below $0.33f_B$, the odd harmonics, caused by truncation error, show up as inband tones while above this frequency, the odd harmonics are out of band. Although the harmonics are less of a concern for high-frequency inputs, the output amplitude is attenuated due to the band limitation of both practical digital IF and analog filter. In general, while the low-frequency degradation is dominated by the inband harmonics, the high-frequency degradation is dominated by the IF filter bandwidth.

Figure 3.10(a) shows the TIM-IF-DSM performance versus input amplitude for an input tone at $0.25f_B$. The SNDR degrades by approximately 4dB with respect to that of SNR strictly due to inband harmonics. On the other hand, figure 3.10(b) shows the TIM-IF-DSM

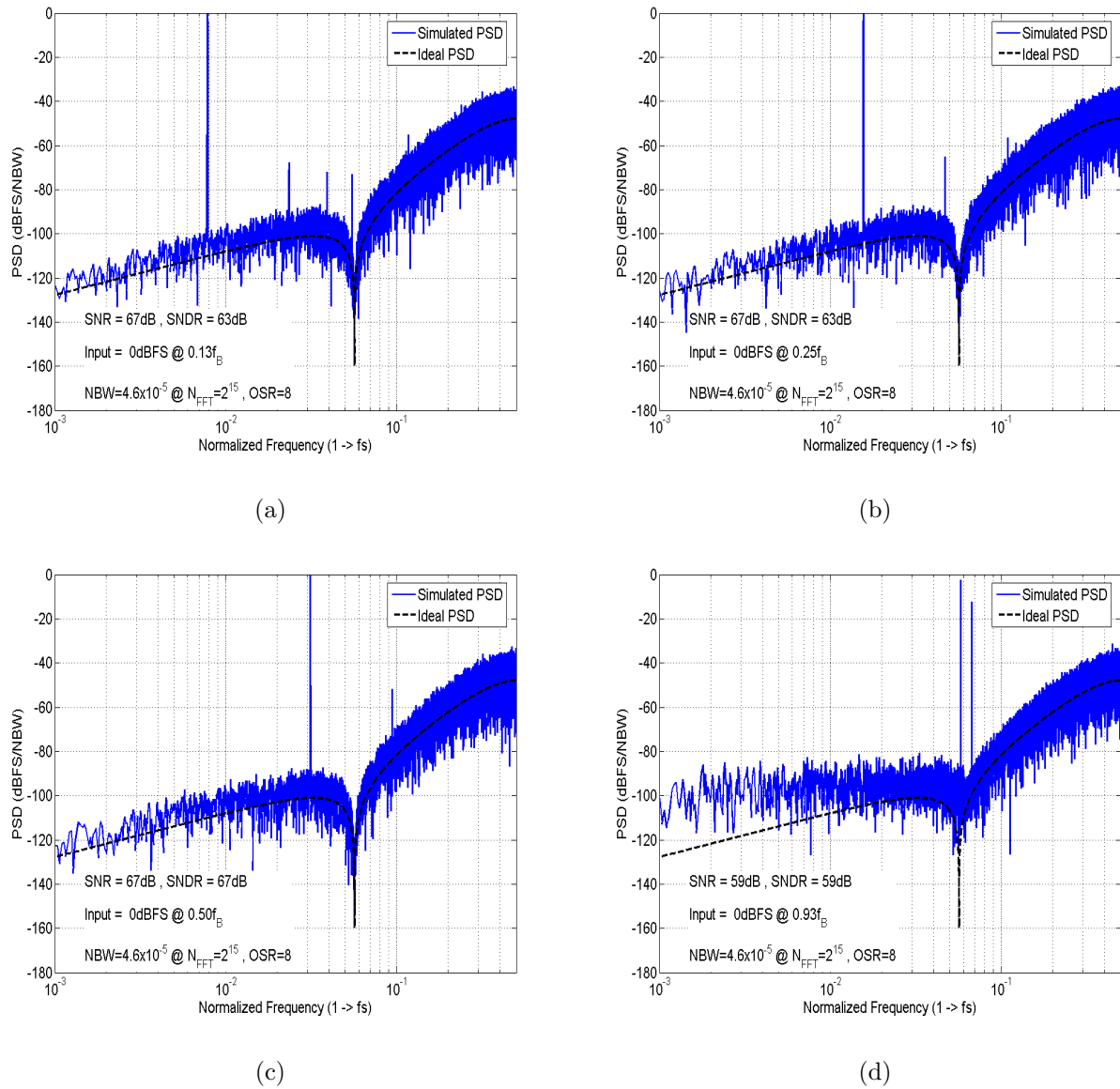
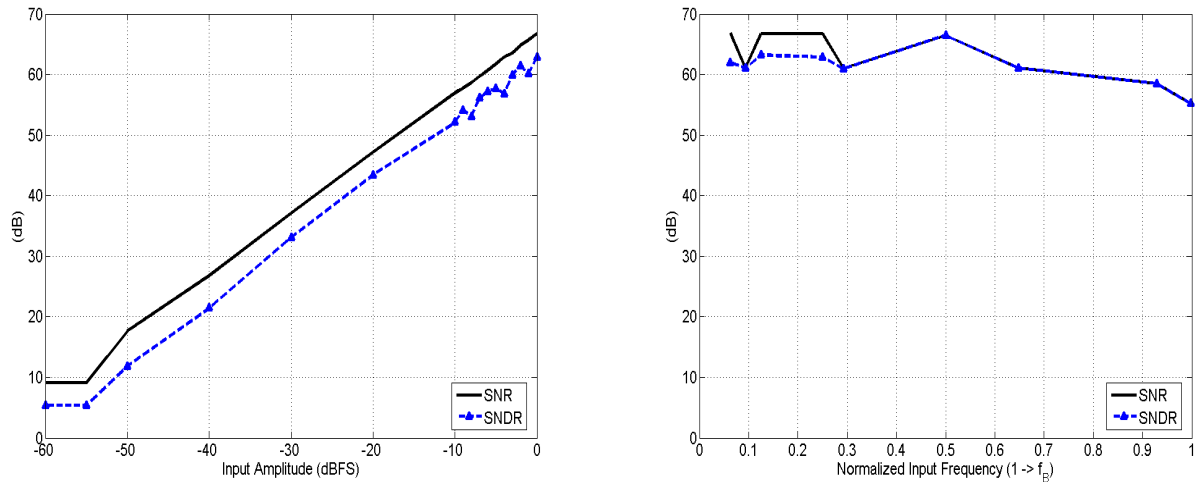


Figure 3.9: TIM-IF-DSM output spectrum for Matlab simulations with 0dBFS input amplitude at different frequencies a) $0.13f_B$ b) $0.25f_B$ c) $0.50f_B$ d) $0.93f_B$



(a) SNR and SNDR vs. Input amplitude

(b) SNR and SNDR vs. Input frequency

Figure 3.10: TIM-IF-DSM response (Matlab simulations)

performance versus input frequency for an input amplitude of 0dBFS. It shows that the SNDR degradation compared to SNR is only prominent for input frequencies below $0.33f_B$ (where the 3^{rd} harmonic falls inband). For higher frequencies, the SNDR is identical to SNR which remains around 60dB up to $0.8f_B$; after which, it starts to degrade due to the dominance of the TIM-IF's frequency response (in figure 3.2(b)). Also, figure 3.10(b) shows a performance of at least 8.8 bits for the entire input frequency band, which is quite acceptable for the targeted applications of this work.

3.4. Digital-to-Analog Converter Model

The multiplexed output of the TIM-IF-DSM is fed into a DAC for digital-to-analog conversion. For a high-speed application, a current-steering DAC (CS-DAC) is a popular choice where each unit cell switches a current to either output or ground. The switches are controlled by thermometer codes generated by passing the TIM-IF-DSM output through a binary-to-thermometer (B2T) converter. A thermometer-based CS-DAC has many advantages over its binary counterpart, such as low non-linearity errors, guaranteed monotonicity, and low glitching noise ([37], Ch.12).

A thermometer-based CS-DAC consists of $2^k - 1$ unit cells. Non-linearities, which arise due to mismatches between unit cells, generate inband harmonics and increase the noise floor due to the folding of high-frequency truncation noise into the signal band ([1], Ch.6). Consequently, SNR, SNDR and ENOB are all degraded. Figure B.4 in appendix B shows these degradations for different DAC element mismatches (i.e.: 1% – 4%).

An effective strategy to eliminate spurious harmonics and lower the noise floor is to use *mismatch error shaping*. There are many techniques to achieve this but the most common ones are: DWA (data-weighted averaging), ILA (individual level averaging), vector-based mismatch shaping, butterfly shuffler, and tree-structure element selection. Theoretical analyses and system architectures for some of these techniques are well presented in [1, 39].

Among these multi-bit DAC linearization techniques, DWA and current calibration are the most common ones. There are several different DWA schemes, all of which basically rely on a high OSR to rotate the DAC unit elements so that their average long-term usage is the same. However, the effectiveness of DWA degrades dramatically at low OSRs. Furthermore, while most DWA schemes are conceptually simple, their implementations are quite complex, especially for a high number (> 8) of DAC levels [24]. Also, in this work, the large DWA circuitry would have to operate at 4GHz, which may not be feasible and consume a large amount of power.

On the other hand, current calibration linearizes the multi-bit DAC by dynamically matching its unit current elements. This technique has been used to realize very well-matched current sources (e.g.: up to 16-bit accuracy in [40]). In addition, the calibration circuitry is more straightforward and suitable for high-speed operation. Thus, current calibration is chosen to alleviate the degradations caused by multi-bit DAC element mismatches.

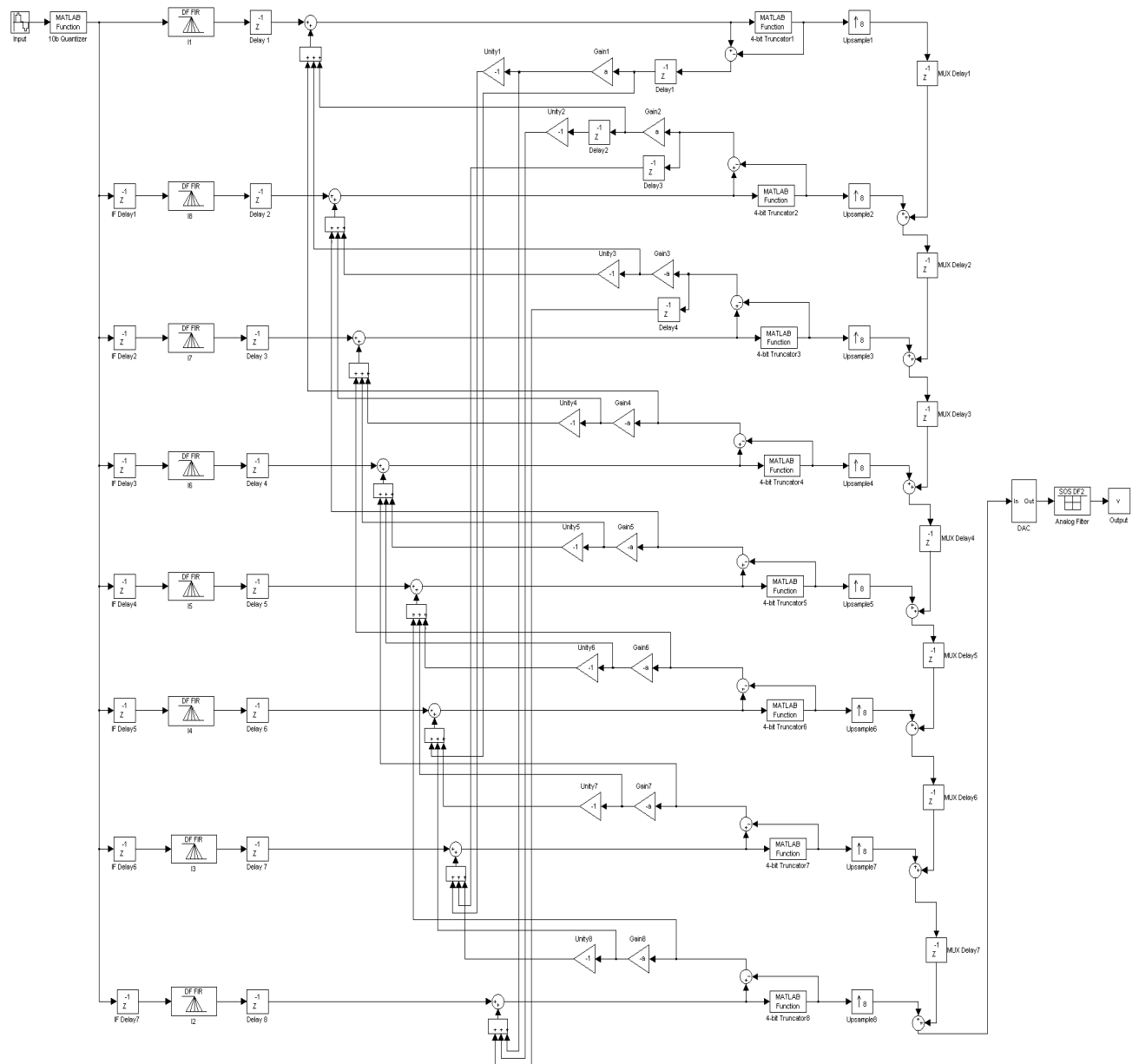
Unlike most linearization techniques which can be modeled accurately in Matlab, it is quite challenging to model and quantify a current calibration system. Hence, for simplicity, the DAC model this architectural-level is assumed to be ideal and mismatch-free. The circuit-level performance of this multi-bit CS-DAC with current calibration will be discussed in chapter 4.

3.5. Analog Reconstruction Filter

The last block of a TIM $\Delta\Sigma$ -DAC is an analog low-pass filter (LPF). The purpose of this filter is to suppress the out-of-band truncation noise, leaving only the signal spectrum within the band of interest. The analog filter's implementation is out of the scope of this work and will not be integrated in the final fabrication. However, one possible design is discussed in Appendix B as an example.

3.6. Summary

In summary, this chapter presents the design details of each sub-block in a TIM $\Delta\Sigma$ -DAC. Figure 3.11 shows a complete Matlab model of this DAC. Note that the parallel paths must be multiplexed in reverse order to generate a correct output. Architectural-level simulation results are presented together with design trade offs and decisions. All sub-blocks, except for the analog filter (TIM-IF, TIM-DSM, MUX, and CS-DAC) will be integrated in STMicroelectronics 90nm CMOS process.

Figure 3.11: Time-interleaved-by-8 $\Delta\Sigma$ -DAC architecture

Time-interleaved $\Delta\Sigma$ -DAC Implementation

This chapter discusses the physical implementation of a TIM $\Delta\Sigma$ -DAC fabricated using STMicroelectronics 90nm CMOS process. It consists of 3 parts as depicted in figure 4.1(b): a digital baseband front-end, a high-speed digital interface, and a high-speed analog back-end. Unlike the conventional $\Delta\Sigma$ -DAC in figure 4.1(a) which operates entirely at $f_S = OSR \cdot f_N$, only the interface and analog section of the TIM $\Delta\Sigma$ -DAC operate at this speed, while the main digital portion operates at f_N .

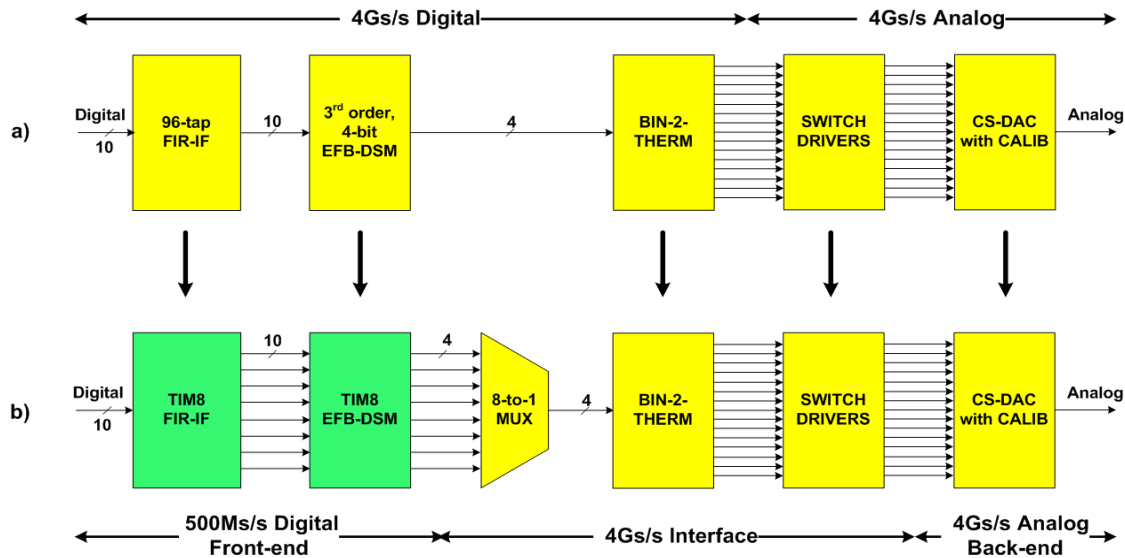


Figure 4.1: a) Conventional $\Delta\Sigma$ -DAC b) Time-interleaved-by-8 $\Delta\Sigma$ -DAC

4.1. Digital Baseband Front-End

The digital baseband front-end is the largest block of the TIM $\Delta\Sigma$ -DAC. It consists of two sub-blocks, a TIM-by-8 IF and a TIM-by-8 DSM, that operate at the same speed as the input sampling frequency (i.e.: 500MS/s). Before discussing the implementation of these sub-blocks, three different optimization techniques are presented for hardware, accuracy, and speed. They reduce the hardware complexity, finite word-length inaccuracies, and propagation delay of this design, respectively.

4.1.1. Hardware Optimization

Multiplierless Implementation

In a practical implementation of a fixed-point digital filter, its coefficients must first be quantized using a power-of-2 representation (e.g: 2's complement) with a fixed word length. The filter requires one multiplier for each of its coefficients. For a high order filter (i.e.: 95th-order TIM-IF), this results in a huge amount of hardware and power consumption, and a long critical path which limits the operating speed. A general purpose multiplier assumes all bits in both the multiplier and multiplicand may change during operation. However, since the coefficients' binary representations are known prior to implementation, multiplication by a constant is equivalent to a combination of binary shifts and additions of only the active bits.

For example, $A \times B$ where $B = 0.1110_2$ can be implemented using only 3 shifters and 2 adders, instead of a 4-bit multiplier, as: $(A \gg 1) + (A \gg 2) + (A \gg 3)$ (where \gg denotes a right shift).

Implementing each filter coefficient using this technique reduces the amount of hardware significantly, thus resulting in lower power consumption and higher operating speed.

Canonic Sign Digit Representation

The multiplierless technique makes hardware complexity proportional to the number of non-zero bits (i.e.: logic 1's) in the filter coefficients. For a further optimization, a canonic sign

digit (CSD) representation can be used where the constant coefficients are represented using the fewest possible number of non-zero bits. It is a signed power-of-2 representation, in which each bit is in the set $\{0, 1, \bar{1}\}$ (where $\bar{1} = -1$) [41]. Here, the coefficients are represented as sums or differences of the fewest possible power-of-2 terms.

For the above example, by converting $B = 0.1110_2$ to $B = 1.00\bar{1}0_2$, $A \times B$ can be implemented using only 1 shifter and 1 adder as: $A - (A \gg 3)$.

Compared to a binary representation, CSD results in further hardware reduction due to a fewer number of shifters and adders required.

4.1.2. Accuracy Optimization

The accuracy of a digital filters is limited by the finite word length arithmetic operations. Three sources of error due to the finite word length are [42]:

1. the quantization of the input signal,
2. the quantization of the filter coefficients, and
3. the accumulation of roundoff errors during arithmetic operations.

Since the input to the TIM $\Delta\Sigma$ -DAC already has a fixed word-length (10 bits), the input quantization error is not applicable in this work. The remaining two sources of errors will be considered in this section.

Optimized CSD Representation of the Filter Coefficients

In this work, one of the major challenges is the physical implementation of the 95th-order IF. Due to its high order, coefficient quantization can have a significant effect on its stopband attenuation. Fortunately, since the IF in this work is integrated with a “noise-shaping” DSM, some degradation in the IF’s stopband attenuation can be tolerated. The out-of-band noise will be dominated by a large amount of shaped truncation noise introduced after the TIM-DSM. Hence, the IF implementation is acceptable as long as it preserves the passband response while providing a reasonable amount of attenuation in the stopband, as shown previously in figure 3.2.

To reduce the coefficients' quantization errors while maintaining a practical implementation, a rule of thumb proposed in [43] is used as the basis to optimize their CSD representations:

- *One nonzero digit in the CSD code is typically required for each 20dB of stopband attenuation in the filter specifications.*

Recall from the IF design in chapter 3, the filter's stopband attenuation was 40dB. Thus, two nonzero CSD digits are generally used to represent each filter coefficient.

Roundoff Errors Reduction Scheme

Roundoff errors are inevitable in fixed-length digital operations. There has been much research to reduce these deterministic errors. In [44], an adaptive carry generation circuitry, based on an exhaustive simulation or statistical analysis, is used to approximate the roundoff errors being compensated. Inspired by this idea of carry compensation, the rounding scheme in this work uses both an exact and an approximate carry as shown in figure 4.2.

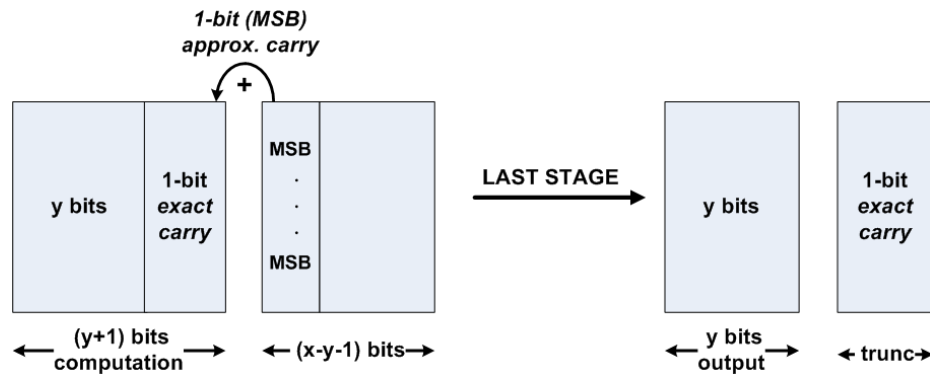


Figure 4.2: Error reduction rounding scheme

Specifically, to obtain a y -bit output from the sum of x -bit inputs (where $x > y$), all computations are done using $(y+1)$ bits, where the extra bit represents the *exact carry*. To account for the truncated $(x-y-1)$ bits, the MSB of this portion is added to the $(y+1)$ -bit sum; this MSB represents the *approximate carry*. Finally, the $(y+1)$ -bit sum is truncated to y -bit output at the last stage. In the example below, three 8-bit numbers are to be added then truncated to form a 4-bit sum.

Example:

Decimal		Binary
+ 79	⇒	0100 : 1111
+ 55		0011 : 0111
+ 39		0010 : 0111
<hr style="width: 100%; border: 0.5px solid black;"/>		
173		

$$173 \times 2^{-4}$$

$$\cong 10.8$$

1. Truncation with No Rounding	2. Truncation with Approximate Rounding	3. Truncation with Exact & Approx. Rounding
$\begin{array}{r} 0100 \\ + 0011 \\ \hline 0010 \\ \hline 1001 \end{array}$ <p>$1001_2 = 9_{10}$</p> <p> error = 1.8</p>	$\begin{array}{r} 0100 + 1 \\ + 0011 + 0 \\ \hline 0010 + 0 \\ \hline 1010 \end{array}$ <p>$1010_2 = 10_{10}$</p> <p> error = 0.8</p>	$\begin{array}{r} 0100 \mathbf{1 + 1} \\ + 0011 \mathbf{0 + 1} \\ \hline 0010 \mathbf{0 + 1} \\ \hline 1011 \mathbf{0} \\ \Rightarrow 1011 \end{array}$ <div style="border: 1px solid black; padding: 2px; display: inline-block; margin: 5px;"> $1011_2 = 11_{10}$ </div> <p> error = 0.2</p>

The correct result is approximately 10.8 in a decimal representation, where the 4-bit truncation is included by multiplying by 2^{-4} . The first truncation method, which does not include any rounding, results in a largest error ($\Delta=1.8$). The second truncation method, which includes a 1-bit approximate carry, results in a nominal error ($\Delta=0.8$). Lastly, the proposed truncation method, which includes a 1-bit exact and 1-bit approximate carry, results in a smallest error ($\Delta=0.2$).

Note that with more exact-carry bits, even higher accuracy can be achieved. However, this would degrade the speed and increase the area for a small improvement in accuracy. For this design, a 1-bit exact carry and 1-bit approximate carry give a good trade-off between these design considerations.

4.1.3. Speed Optimization

Parallel Adder Architecture

As mentioned in chapter 2, pipelined or parallel adders are required in this design to minimize the critical path delay. Many different adder architectures are possible; the best choice depends on the specific design. For example, a ripple carry adder (RCA) has the smallest

area and lowest power but also slowest speed. On the other hand, a carry look-ahead adder (CLA) has the fastest speed but its power consumption is relatively high. A carry select adder (CSA) is a compromise between the high-speed operation of the CLAs and the low-power consumption of the RCAs ([45], Ch. 7). Thus, the CSA architecture is used in this work.

Figure 4.3 shows the architecture of a CSA, which consists of two full adders (FAs) for each bit's addition: one FA assumes the carry-in (C_{in}) is '1' while the other assumes the C_{in} is '0'. The FAs are grouped into "stages", each of which is a RCA. At each stage, the C_{in} is obtained from the previous stage, except for the first stage where C_{in} is an input. This C_{in} selects one of the two sums, and one of the two carries, through simple 2-to-1 multiplexers.

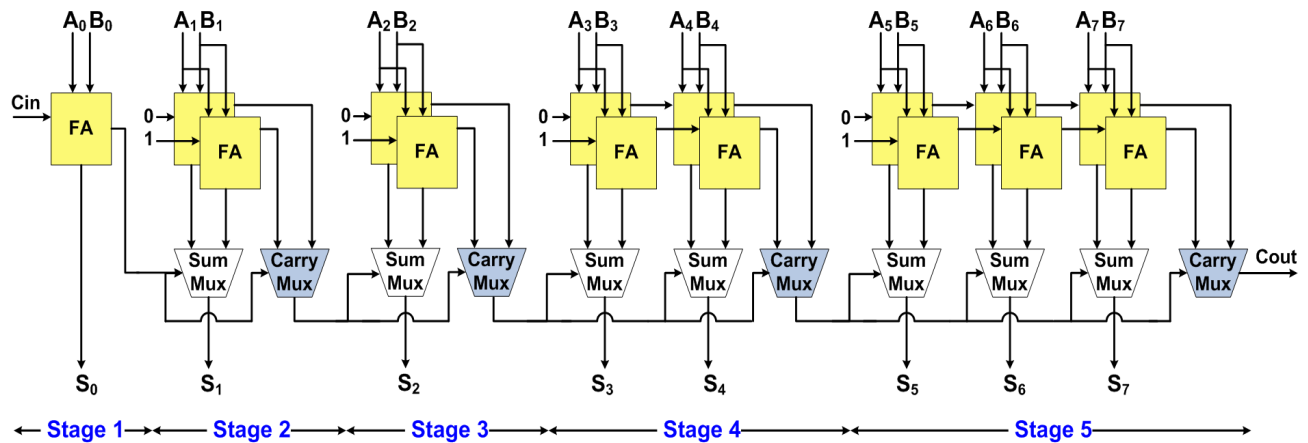


Figure 4.3: Example of an 8-bit CSA with 1-1-1-2-3 staging

CSA Staging Optimization

The critical path, and hence the maximum operating speed of the CSA depends to a great extent on the number of bits allocated to each stage. For example, a staging of (4-4-4-4-4-4-4) for a 32-bit adder does not result in the maximum speed due to the multiplexing delay of the carry path([45], Ch. 7). The optimal CSA staging depends on the specific technology and adder word-length. Table 4.1 shows the CSA staging that results in the shortest critical path for different CSA lengths. It also shows the estimated and synthesized delay using 90nm CMOS standard-Vt digital libraries (CORE90GPSVT and CORX90GPSVT). A

combination of RCAs for low-bit adders (4-5 bits) and CSAs for medium to high-bit adders can be used for further speed optimization. However, for simplicity, CSAs are used for all bit adders. The CSA delay can be estimated as:

$$t_{CSA} = (\# \text{ of stages} - 1) \times t_{MUX} + (\max \# \text{ of FAs per stage}) \times t_{FA} \quad (4.1)$$

For example, for a 8-bit CSA, which has 1-1-1-2-3 staging, $t_{CSA} = 4(t_{MUX}) + 3(t_{FA})$.

Table 4.1: CSA Staging Optimization

CSA Bits	Staging	# of Stages	Estimated t_{CSA}	Synthesized t_{CSA} (ns)
4	1-1-2	3	$2(t_{MUX}) + 2(t_{FA})$	0.20
5	1-1-1-2	4	$3(t_{MUX}) + 2(t_{FA})$	0.22
6	1-1-1-1-2	5	$4(t_{MUX}) + 2(t_{FA})$	0.26
7	1-1-1-2-2	5	$4(t_{MUX}) + 2(t_{FA})$	0.29
8	1-1-1-2-3	5	$4(t_{MUX}) + 3(t_{FA})$	0.30
9	1-1-1-2-2-2	6	$5(t_{MUX}) + 2(t_{FA})$	0.28
10	1-1-1-2-2-3	6	$5(t_{MUX}) + 3(t_{FA})$	0.30
11	1-1-1-2-2-2-2	7	$6(t_{MUX}) + 2(t_{FA})$	0.31
12	1-1-1-2-2-2-3	7	$6(t_{MUX}) + 3(t_{FA})$	0.35

4.1.4. Time-interleaved Interpolation Filter

This section presents the physical implementation of the TIM-IF which was designed in section 3.2. For a 95th-order IF with a time-interleaving factor of 8, this corresponds to 8 parallel paths with 12 coefficients/path. For simplicity, the coefficient notations in each *local* path are referred to as $g'(n)$, instead of the *global* $g(n)$ as shown in figure 3.3. Table C.1 in appendix C lists the coefficient values for all TIM-IF paths in both original and quantized CSD representation.

Figure 4.4 shows the physical implementation of the TIM-IF. Here, the D-flipflop (DFF) represents a delay while the “sum tree” represents the summation of all coefficients for each path. Recall from chapter 3 that all TIM-IF paths, except path 1 (U_1), are in reverse order;

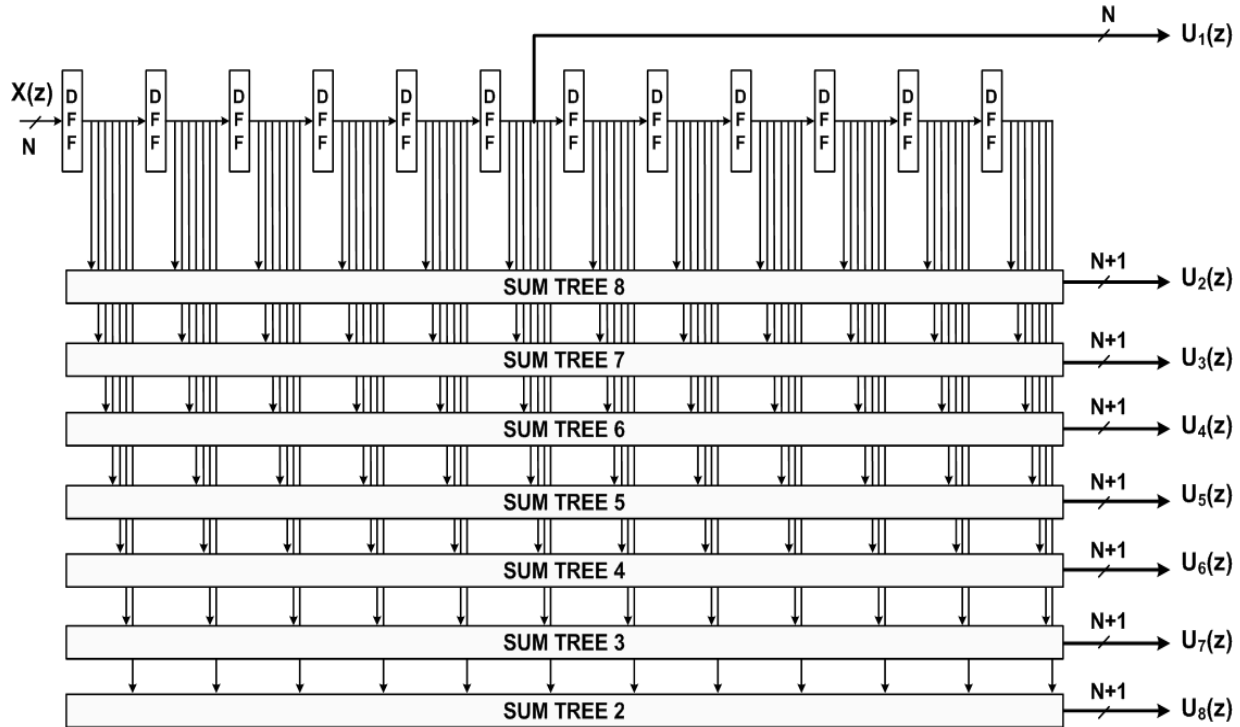


Figure 4.4: TIM-IF Physical Implementation

for example, sum tree 8 actually belongs to path 2 and so on. For an N -bit input, the word-length at the output of the TIM-IF should be $(N+1)$ bits to account for digital arithmetic overflow. This overflow bit is also shared with the TIM-DSM. Thus, it is not necessary to provide another overflow bit for the TIM-DSM which never overflow through the use of a digital limiter. In this work, the input and output are 10 and 11 bits, respectively.

For each path, there are 12 coefficients with each being represented by 1 to 3 CSD terms. Thus, there are many CSD terms and summing operations involved for each path. This makes the fixed word-length output (11 bits) a challenging task. To meet the fixed word-length requirement and also to minimize the delay, a custom “sum tree” is created for each TIM-IF path. Notice that the coefficients for paths 2 & 8 are the same but in reverse order, thus the same sum tree design can be used. The same applies to paths 3 & 7 and paths 4 & 6. Figure 4.5 shows an example of a sum tree for path 2 & 8; the sum trees for the remaining paths can be found in appendix C.

All sum trees use the CSA described in section 4.1.3. Binary sign extension is used

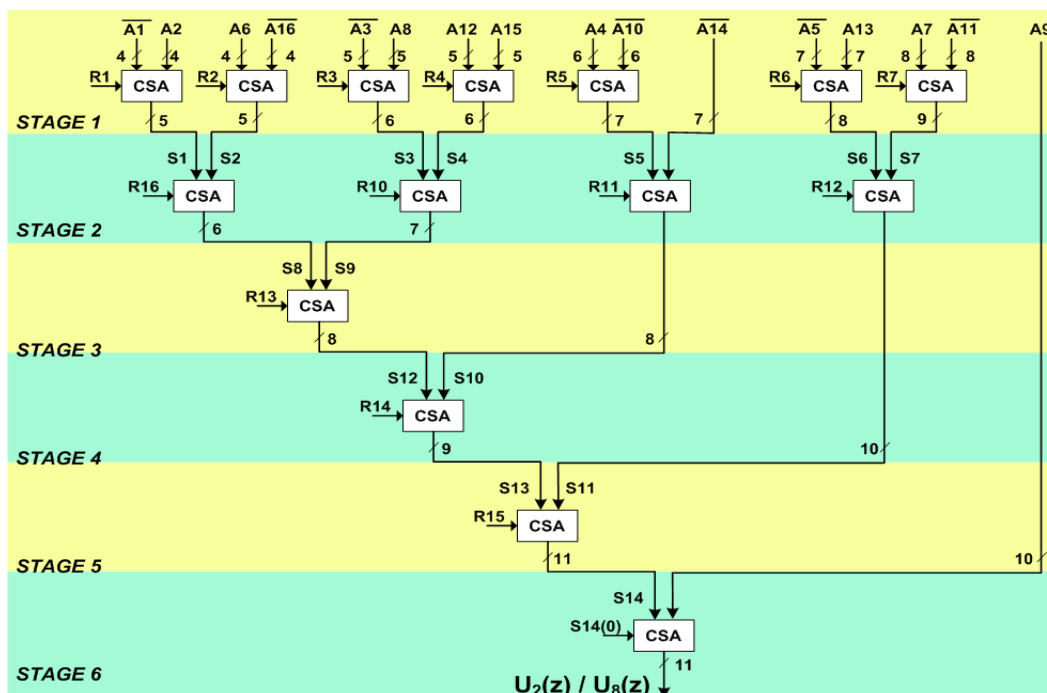


Figure 4.5: TIM-IF sum tree for path 2 and 8

whenever needed to ensure that both inputs to each CSA have the same word-length. The shortest word-length terms are summed up first, then the longest terms last. Also, the proposed rounding scheme is applied to each sum tree: the approximate carry-ins (e.g.: R1, R2, etc) are fed into all available CSAs, while the exact carry-ins are part of the CSD terms from the beginning. All intermediate sums are computed with one extra bit, except the last summation where the final output is rounded off to the desired length. Overall, the sum tree ensures that a carry is accounted at every CSA and maintained a final output at a fixed word-length of 11 bits.

Synthesized timing simulation results for this TIM-IF can be found in Appendix C.

4.1.5. Time-interleaved $\Delta\Sigma$ Modulator

Unlike the 95th-order TIM-IF, the 3rd-order TIM-DSM contains much fewer CSD terms. A great advantage in implementing the TIM-DSM is that although the inputs to the summers (see figure 3.6) are different, all 8 paths have identical arithmetic operations. Thus, they all have the same sum tree structure.

Recall that the feedback loop filter, $H(z)$, is:

$$\begin{aligned} H(z) &= az^{-1} - az^{-2} + z^{-3} \\ &= 2.875z^{-1} - 2.875z^{-2} + z^{-3} \\ &= (2^1 + 2^0 - 2^{-3})z^{-1} - (2^1 + 2^0 - 2^{-3})z^{-2} + z^{-3} \end{aligned}$$

Based on $H(z)$ and figure 3.6, the output of each TIM-DSM path is:

$$P_x(z) = U_x(z) + (2^1 + 2^0 - 2^{-3})E_{x+1}(z) - (2^1 + 2^0 - 2^{-3})E_{x+2}(z) + E_{x+3}(z) \quad (4.2)$$

where $E_x(z)$ is the truncation error from the x^{th} path of the TIM-DSM, and $U_x(z)$ is the output of the x^{th} path of the TIM-IF.

The sum tree for a TIM-DSM path is shown in figure 4.6. The same summation and rounding scheme used in the TIM-IF are used for the TIM-DSM to maintain an 11-bit word-length output. Recall from chapter 2 that a digital limiter was integrated with the DSM. This ensures that the modulator will operate with a fixed word-length of 11 bits and saturate to the largest digital value in case of an overflow. Through simulations, an overflow only occurs when the input amplitude to the TIM-IF-DSM (TIM-IF + TIM-DSM) is close to the full-scale value, namely $> -0.5dBFS$. In these cases, even though the TIM-DSM saturates, the full system simulation still indicated good performance. Therefore, it is not necessary to assign another overflow bit to the TIM-DSM. In fact, having an overflow bit for the TIM-DSM would deteriorate performance since this 12th bit would usually be '0' and hence does not contain any information. Thus, after bit truncation, only 3 out of 4 bits actually contain meaningful data, resulting in a loss of output amplitude.

Synthesized timing results for this TIM-DSM can be found in Appendix C. Since the digital front-end contains both the TIM-IF and TIM-DSM (i.e.: TIM-IF-DSM), unless mentioned otherwise, the behavioural simulations and physical design will contain both blocks.

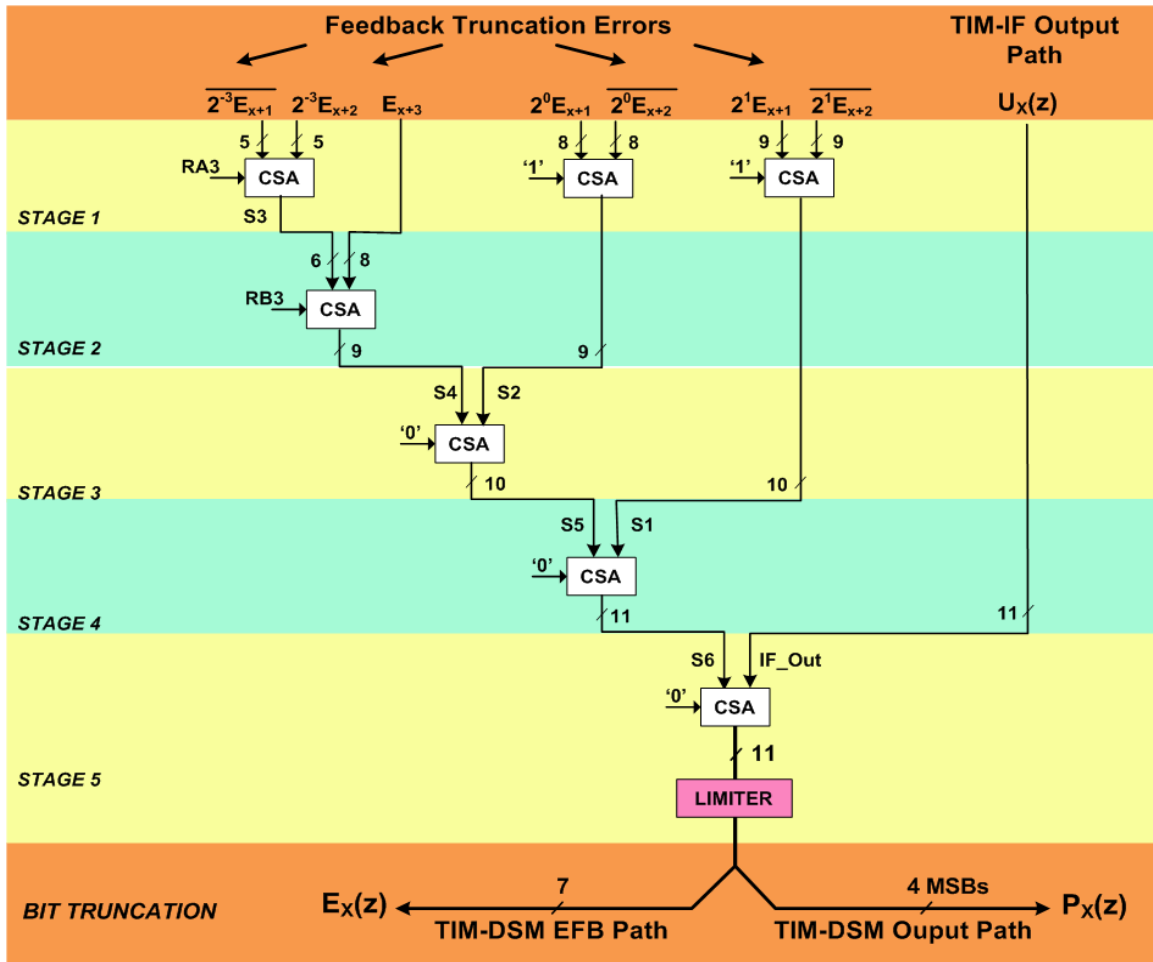


Figure 4.6: TIM-DSM sum tree

4.1.6. Digital Integrated Circuits Design Flow

The digital implementation of the TIM-IF-DSM front-end consists of two design phases, soft design and physical design, as depicted in figure 4.7. The soft design is done using Matlab, VHDL, and Synopsis; whereas the physical design is done using SOC First Encounter.

The soft design phase requires several iterations before the design is finalized. Initially, the architectural-level is translated into the register-transfer-level (RTL) using VHDL. If the VHDL behavioral simulations do not meet the required specification, the architecture needs to be modified or re-designed. Similarly, if the design fails to meet the required speed after synthesis, the timing constraints need to be modified. In a linear system, the RTL behavioural simulations can be verified using a self-checking test bench or a scan chain test. However, due to the non-linearity nature of $\Delta\Sigma$ modulation, the RTL behavioural simulations was verified by performing FFT function to obtain their spectra and SNR performance.

In the second phase, the synthesized design is imported into SOC First Encounter for physical placement and routing. The initialization step involves floor planning and power planning. The floor planning specifies the chip's dimensions and its core utilization. The higher the core utilization, the smaller the area; however, this makes signal routing a difficult task. In this design, a core utilization of 50% is used for 90nm CMOS. The power planning step specifies the appropriate VDD/GND rings and grids for uniform power distribution.

Placement is one of the critical steps in physical design. It includes I/Os, standard cell, and clock tree placement. Placement is done based on timing-driven criteria which require multiple iterations. Even for the 90nm CMOS process, it is non-trivial to achieve a 500MHz clock rate in this highly dense design through the use of digital standard cells. The timing margins (shown in Appendix C) are tight even with exhaustive optimizations. The estimated clock skew was about 16ps after placement, which is acceptable for a 500MHz clock.

Auto routing involves power and signal routing. Power routing distributes power to all standard cells. Signal routing ensures all physical geometry rules (e.g.: metal width, spacing, density, etc) are met while minimizing the propagation delay to meet the timing constraints. Lastly, design verifications including DRC and LVS are done before the digital front-end layout is integrated with the custom layout high-speed blocks in Cadence.

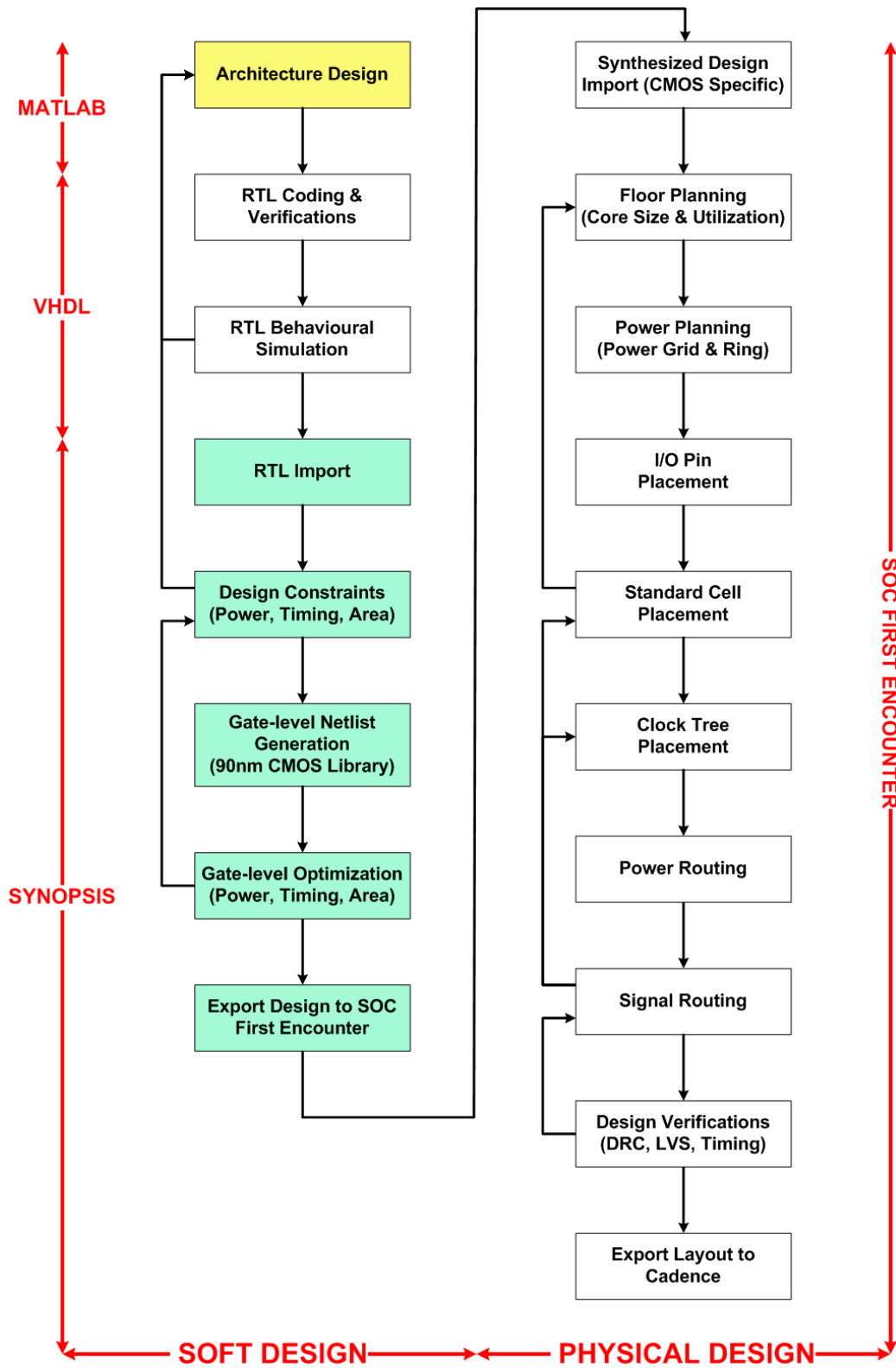


Figure 4.7: Digital design flow

4.1.7. Digital Front-end Simulation Results

This section presents the behavioural simulation results for the digital TIM-IF-DSM front-end. The behavioural results are first obtained from a VHDL simulator then multiplexed and post-processed in Matlab under “brick wall” filtering assumption.

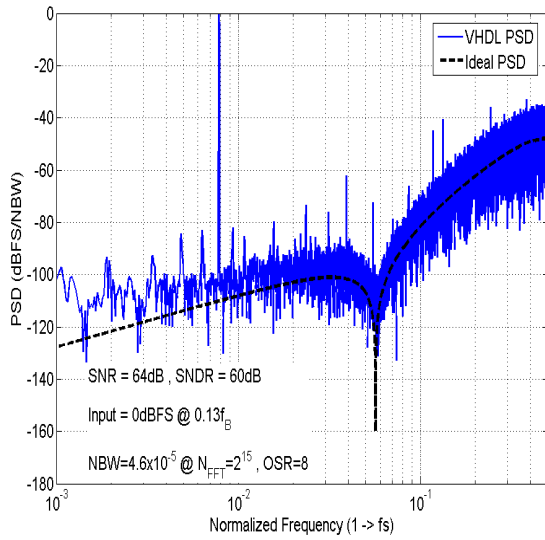
Figure 4.8 shows the TIM-IF-DSM VHDL behavioural output spectrum for 0dBFS input amplitude at different frequencies. For comparison, an equivalent figure, which shows the same output spectrum simulated in Matlab with floating-point precision, is figure 3.9.

Figure 4.9(a) and 4.9(b) show the SNR and SNDR versus input amplitude, respectively, for the TIM-IF-DSM simulated in VHDL (fixed-point) and Matlab (floating-point) with a single tone at $0.25f_B$. The VHDL behavioural SNR degrades about 3dB on average as compared to that of the Matlab simulation. On the other hand, the VHDL behavioural SNDR fluctuates by about ± 2 dB.

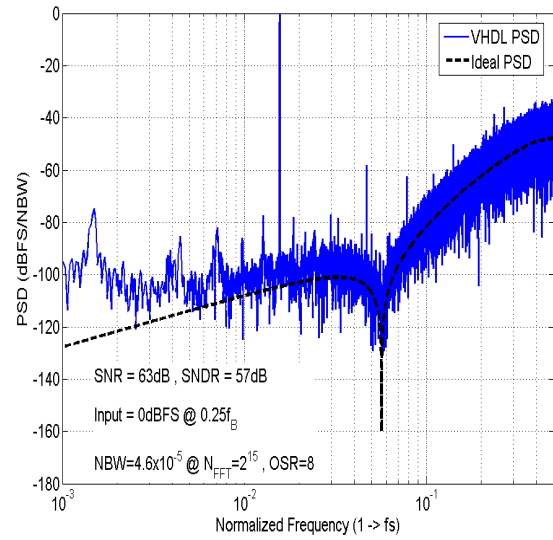
Figure 4.9(c) and 4.9(d) show the SNR and SNDR versus input frequency, respectively, for TIM-IF-DSM simulated VHDL and Matlab with a 0dBFS input amplitude. Similar to the previous analysis, there is an average of 3dB degradation in VHDL results as compared to those in Matlab simulations.

An input sampling frequency of 500MS/s corresponds to a clock period (t_{CLK}) of 2ns. Tables C.2 and C.3 in appendix C show the worst-case timing margins from Synopsis for the TIM-IF and TIM-DSM, respectively. These tables show the synthesized timing for each sum tree alone, then for a full path which contains a sum tree plus D-flipflop (DFF) and buffers. The synthesized power consumption, which obtained from the digital standard cell library, is around 51mW based on a 1V supply.

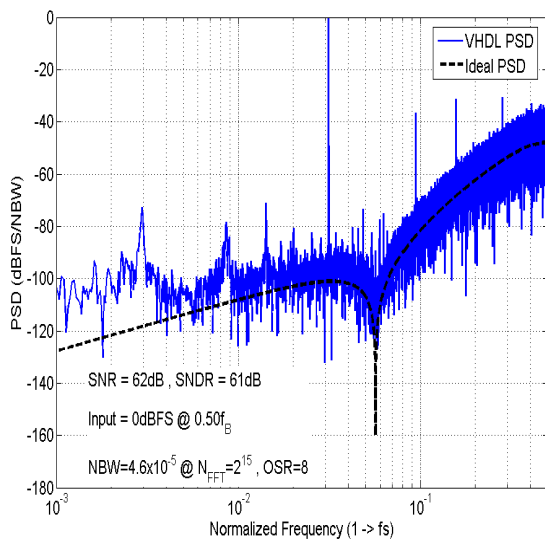
The positive timing margins imply that both the TIM-IF and TIM-DSM can operate at 500MS/s. Since Synopsis obtains the worst-case timing data from the 90nm CMOS standard cell library and it also accounts for the wiring interconnection, having positive timing margins indicates that the physical design should function properly.



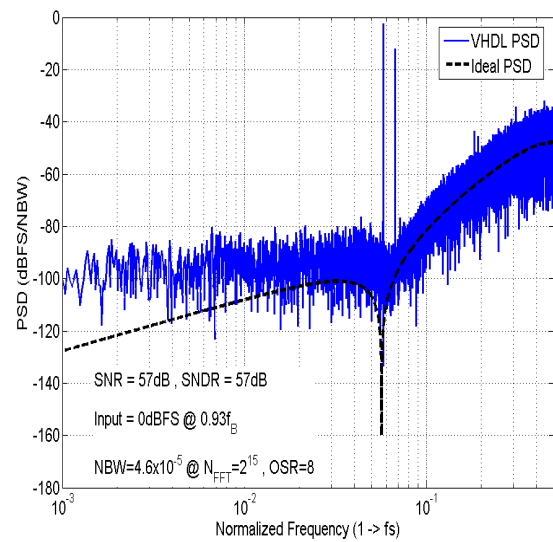
(a)



(b)

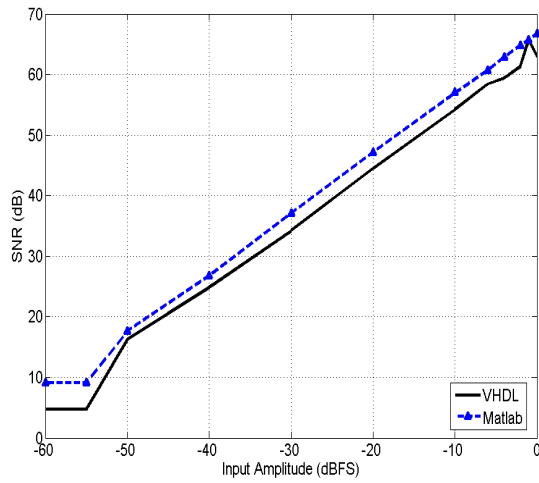


(c)

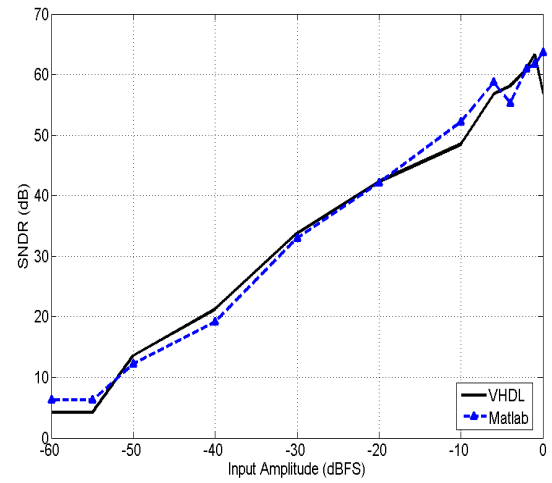


(d)

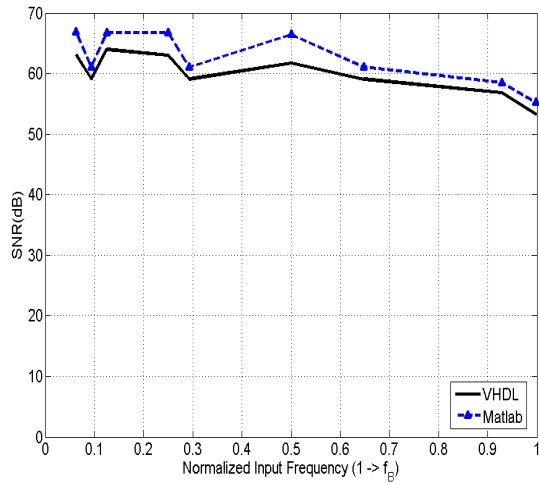
Figure 4.8: TIM-IF-DSM output spectrum for VHDL behavioural simulations with 0dBFS input amplitude at different frequencies: a) $0.13f_B$ b) $0.25f_B$ c) $0.50f_B$ d) $0.93f_B$



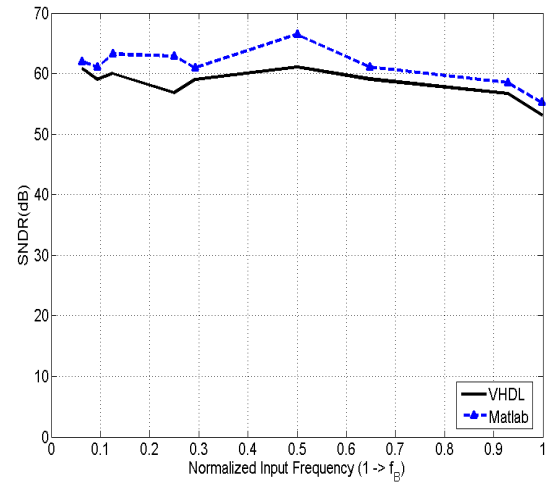
(a) SNR vs. Input amplitude



(b) SNDR vs. Input amplitude



(c) SNR vs. Input frequency



(d) SNDR vs. Input frequency

Figure 4.9: TIM-IF-DSM VHDL Behavioural vs. Matlab Response

4.2. High-Speed Digital Interface

The high-speed digital interface bridges the gap between the digital front-end and the analog back-end. It consists of three sub-blocks which all operate at the oversampling rate of 4GS/s: an 8-to-1 multiplexer, a binary-to-thermometer converter, and the switch drivers. From this point onward, all design and simulations will be done at the transistor level using Cadence.

4.2.1. Multiplexer

Compared to a conventional $\Delta\Sigma$ -DAC, the only additional block in a time-interleaved $\Delta\Sigma$ -DAC is the multiplexer, as shown in figure 4.1. The purpose of the multiplexer is to serialize the parallel time-interleaved paths down to a single path. The multiplexing factor is identical to that of the time-interleaving factor, namely a factor of 8. Traditionally, an 8-to-1 multiplexer that achieves 4GS/s data rate would require three different clock rates: 500MHz, 1GHz, and 2GHz. This work proposes an 8-to-1 “ring” multiplexer which only requires a single clock rate of 4GHz.

The 8-to-1 ring multiplexer consists of three parts, as depicted in figure 4.10: a ring shift register, a switch shift register, and a data multiplexer. The ring shift register consists of 8 cascaded DFFs clocked at 4GHz (CLK_a). It also consists of 2 transmission gates to set the ring’s initial state to a known value (of logic 1) at power-up. This ring creates a pulse signal, S_0 , which has a period of 2ns and a pulse width same as CLK_a period, namely 250ps. S_0 has two purposes: it is used as the 500MHz clock pulse (CLK_{sw}) for the DFFs in the data multiplexer, and also used to generate 8 switch signals ($S_1 - S_8$) in the switch shift register.

Instead of taking $S_1 - S_8$ directly from the ring shift register, a separate series of 8 DFFs (i.e.: switch shift register) is needed because ($S_1 - S_8$) are not consecutively shifted (by 1 clock cycle) versions of S_0 , but rather its delayed versions. These signals activate the switches of the data paths in reverse as shown figure 4.10.

Figure 4.11 shows the timing diagram for an 8-to-1 ring multiplexer. Before S_0 is used as CLK_{sw} , it has to go through a clock tree, which consists of 5 stages of branching fanout-of-2 buffers to drive 32 DFFs (since there are 8 parallel paths and 4 bits/path). Thus, CLK_{sw} is delayed by $t_{clk.tree}$ from S_0 .

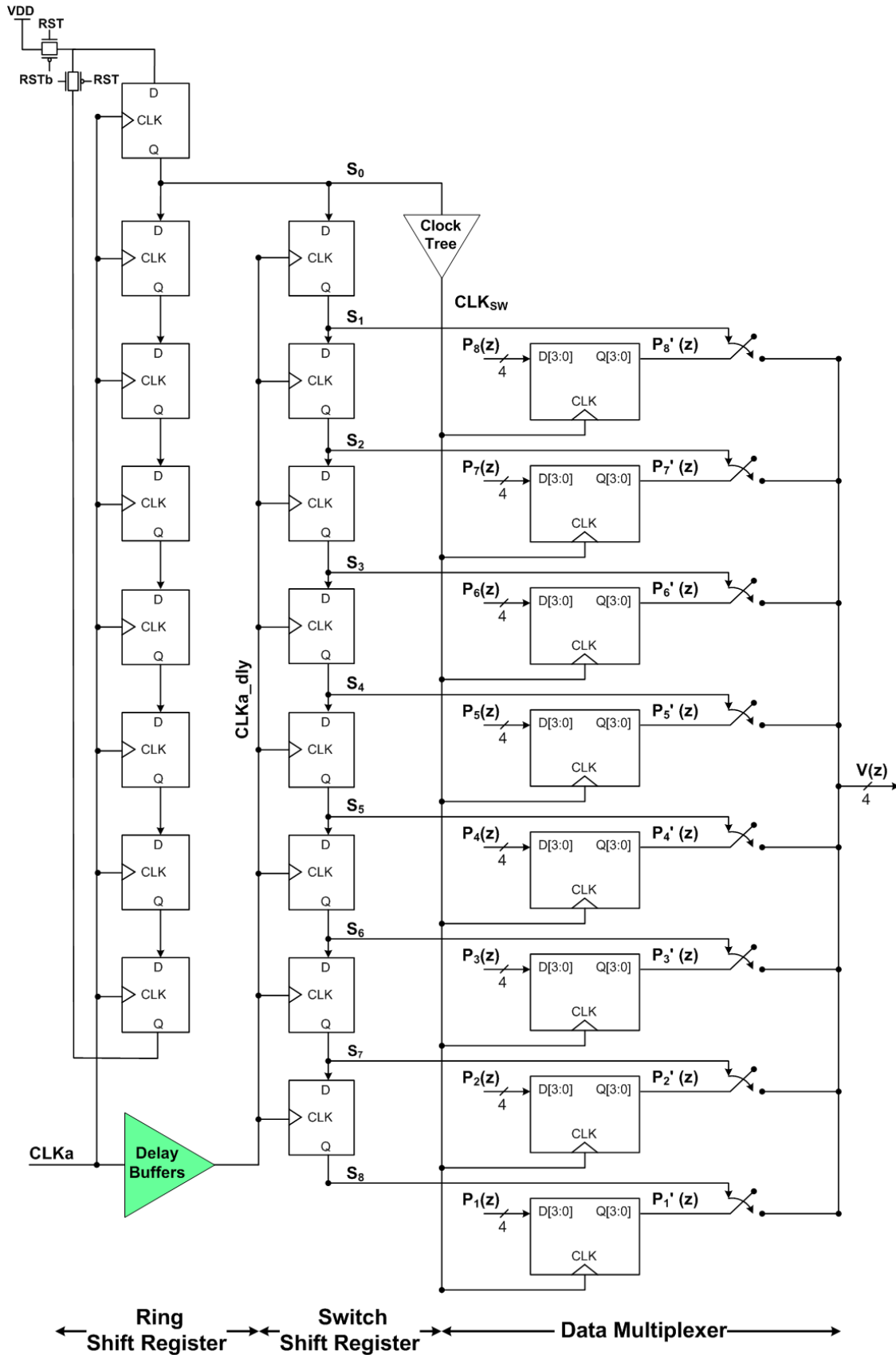


Figure 4.10: An 8-to-1 ring multiplexer

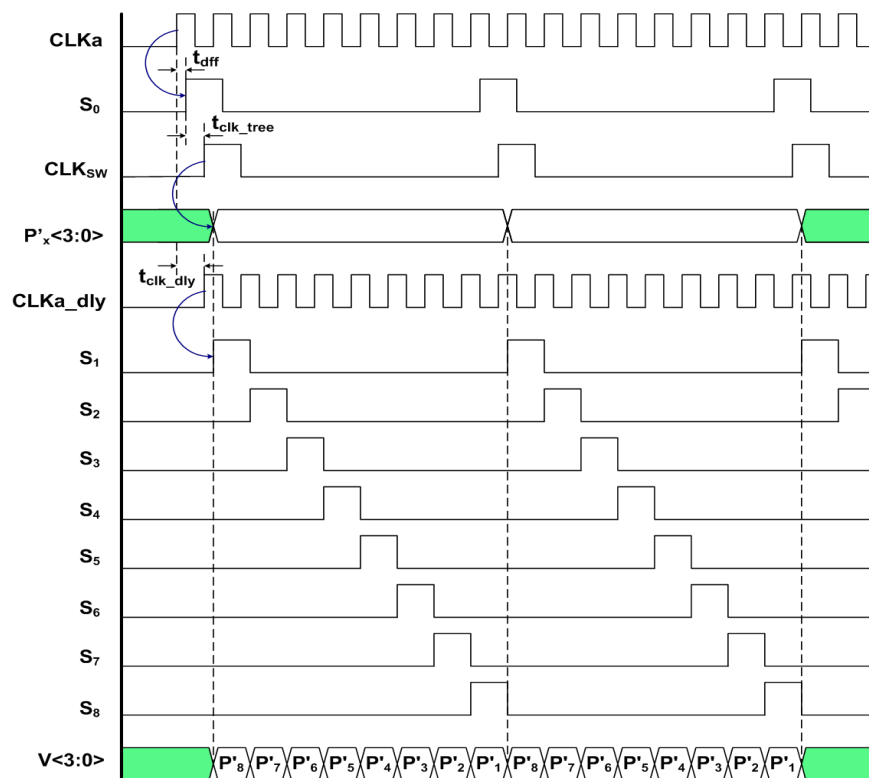


Figure 4.11: Timing diagram for an 8-to-1 ring multiplexer

To meet the correct timing, the switch signals need to be aligned with the data paths so they can output valid data. Since the same DFF is used everywhere, if the switch shift register's clock is aligned with the data multiplexer's clock (CLK_{sw}), then the switch signals are guaranteed to line up with the data paths.

The switch shift register's DFFs must be clocked at the same rate as CLK_a . However, this clock is required to be delayed (CLK_a_dly) such that it will be edge-aligned with CLK_{sw} . According to figure 4.11, a simple solution is to delay CLK_a approximately by one DFF plus the clock tree's propagation delay as: $t_{clk_dly} = t_{dff} + t_{clk_tree}$.

A more accurate but complicated solution is to use a PLL to align the phases of CLK_a_dly and CLK_{sw} . For this design, it is not necessary to use a PLL for exact alignment since a skew of 20ps can be tolerated as long as all 8 switch pulses are within one data period. To ensure proper timing alignment, the output multiplexed data will be re-timed by another DFF at the switch driver before entering the analog back-end.

Figure 4.12(a) and 4.12(b) show the Cadence simulation results (in TT corner, 27°C) for the 8-to-1 ring multiplexer (MUX8) operating at 4GHz and 2GHz clock, respectively. Here, the two clocks (CLK_{sw} and CLK_{dly}) are aligned within 15ps and all eight switch pulses (S_1-S_8) are contained within one data period. Figure 4.12(b) shows that even for a frequency lower than the one being designed for, the MUX8 still operates properly. Simulations over different process corners and temperatures (i.e.: SS, 105°C and FF, -40°C) also showed the MUX8's functionality.

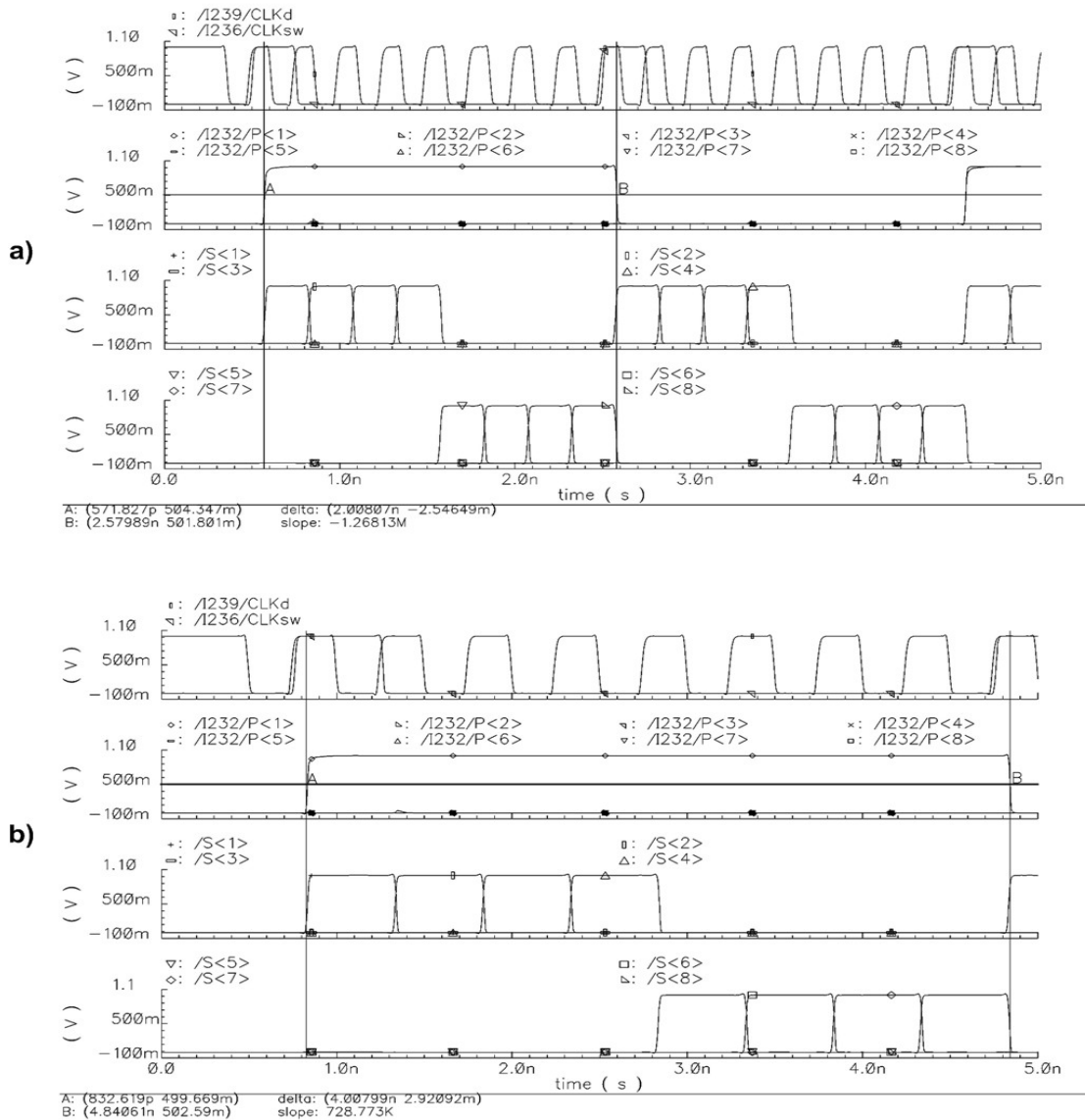


Figure 4.12: An 8-to-1 ring multiplexer transient response (TT corner) a) 4GHz b) 2GHz

4.2.2. Binary-to-Thermometer Converter and Switch Drivers

A binary-to-thermometer (B2T) converter has been a standard digital block for designs that operate in the kHz to MHz range. However, the B2T converter in this work is required to operate at 4GHz, hence eliminating the usage of digital standard cell libraries. To meet the high-speed timing, its propagation delay is required to be within 1/2 of the CLKa period (i.e.: 125ps) . This level of performance can only be achieved with custom CMOS logic and high-speed layout.

Table 4.2 shows the 4-bit B2T conversion and gate logic. Since the TIM-IF-DSM's outputs are based on 2's complement while the DAC requires thermometer code inputs, it is required to convert the 4-bit data from 2's complement to unsigned binary representation, and then to a 15-bit thermometer code.

Table 4.2: Binary-to-thermometer conversion and gate logic

2's COMP. V<3:0>				UNSIGNED A<3:0>				THERMOMETER T<15:1>															
V ₃	V ₂	V ₁	V ₀	A ₃	A ₂	A ₁	A ₀	T ₁₅	T ₁₄	T ₁₃	T ₁₂	T ₁₁	T ₁₀	T ₉	T ₈	T ₇	T ₆	T ₅	T ₄	T ₃	T ₂	T ₁	
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
1	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
1	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
1	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
1	1	1	1	0	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	1	1	0	0	1	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
0	0	1	0	1	0	1	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
0	0	1	1	1	0	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
0	1	0	0	1	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	0	1	1	1	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Binary to Thermometer CMOS Logic	$T_1 = (A_3 + A_2) + (A_1 + A_0)$				$T_9 = (A_3A_2) + A_3(A_1 + A_0)$																		
	$T_2 = (A_3 + A_2) + (A_1)$				$T_{10} = (A_3A_2) + (A_3A_1)$																		
	$T_3 = (A_3 + A_2) + (A_1A_0)$				$T_{11} = (A_3A_2) + A_3(A_1A_0)$																		
	$T_4 = (A_3 + A_2)$				$T_{12} = (A_3A_2)$																		
	$T_5 = (A_3) + A_2(A_1 + A_0)$				$T_{13} = (A_3A_2)(A_1 + A_0)$																		
	$T_6 = (A_3) + (A_2A_1)$				$T_{14} = (A_3A_2)A_1$																		
	$T_7 = (A_3) + A_2(A_1A_0)$				$T_{15} = (A_3A_2)(A_1A_0)$																		
	$T_8 = (A_3)$																						

The decimal range for a 4-bit binary word is [-8,7] and [0,15] in 2's complement and unsigned binary, respectively. Thus, the conversion from 2's complement to unsigned binary would only require an addition of 8. This is accomplished by inverting the most significant bit

while the remaining bits ($V < 2 : 0 >$ and $A < 2 : 0 >$) are identical in both representations. The B2T converter's area is reduced through gate re-use and its codes propagation delays are matched as much as possible.

The thermometer codes are used to switch the DAC's current-steering cells. Before entering the analog back-end, these codes need to be re-timed and driven by the switch drivers. The details on switch drivers are discussed in appendix C.

4.2.3. High-Speed Digital Interface Simulation Results

This section presents the transient simulation results (in TT corner, $27^\circ C$) for the high-speed digital interface which contains the 8-to-1 ring multiplexer, binary-to-thermometer converter, and switch drivers. As an example, figures 4.13 and 4.14 present the expected theoretical and Cadence simulation results, respectively. The TIM-IF-DSM outputs are first multiplexed by MUX8, converted to thermometer codes by the B2T converter, then driven to the analog back-end by the switch drivers. Both figures agree implying that this interface works properly at 4GHz.

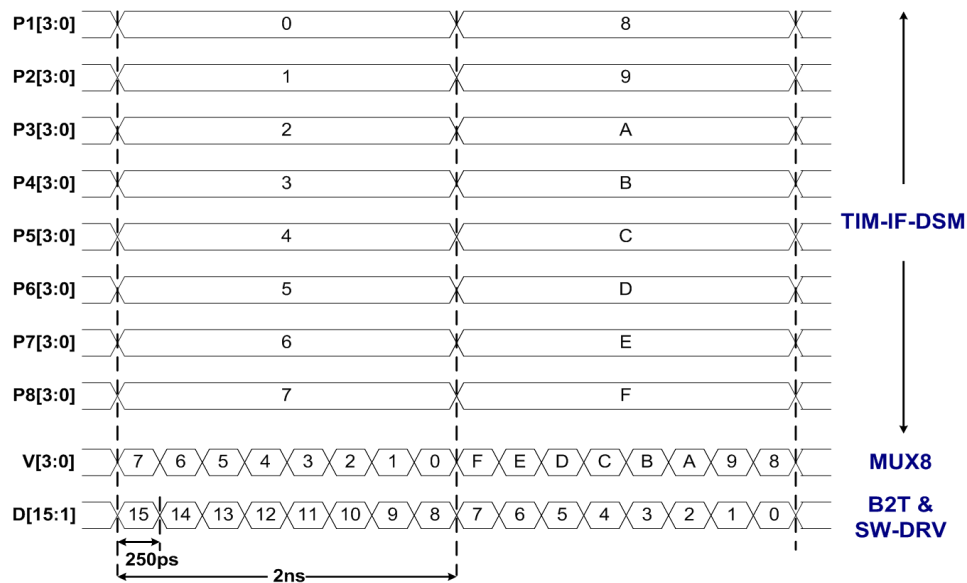


Figure 4.13: High-speed digital interface theoretical response

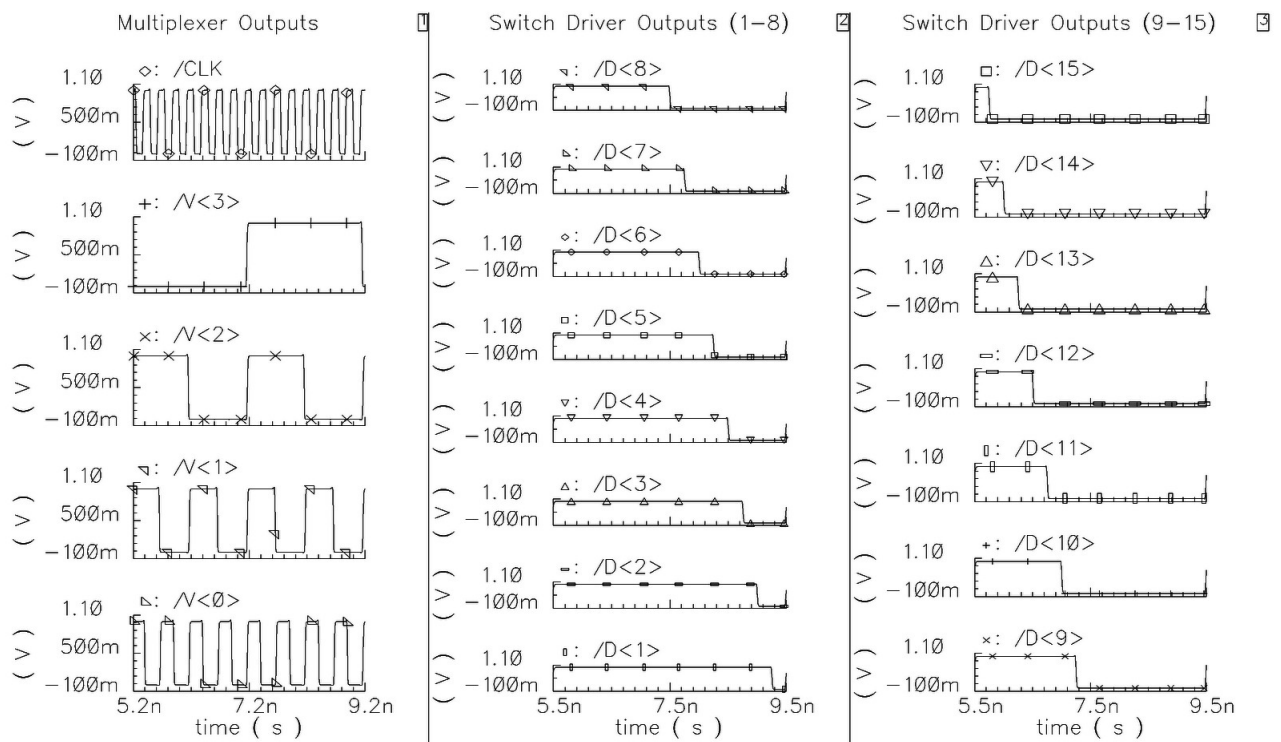


Figure 4.14: High-speed digital interface Cadence transient response (TT corner)

4.3. High Speed Analog Back-End

The analog back-end, which operates at 4GS/s, consists of two sub-blocks: a current calibration circuit and a current-steering DAC. The current calibration technique is used to achieve the required resolution and current-steering DAC is used to achieve high-speed operation.

4.3.1. Current Calibration Circuitry

As mentioned in chapter 3, a current calibration technique is used to linearize the multi-bit DAC by dynamically matching its unit current elements. Some calibration techniques require a calibration period during which the DAC is temporarily disabled.

In this work, a self-calibration is used which does not require a calibration period, thus allowing the DAC to operate continuously. The details of current calibration principles are discussed in appendix C. A calibration period T_c of 160ns (i.e.: 10ns/cell calibration time for 16 current cells) is sufficient for this design.

Practical Considerations

A major challenge in current calibration is matching the output current I_{out} between the cells. The main mismatches occur at the calibration switches and the MOS transconductance. The switch mismatches are due to their sizes, which are required to be small to keep I_{leak} minimal. Thus, a mismatch in the charge-injection for each cell is expected [40]. To reduce this effect, two additional transmission-gate (TG) switches (T_2 and T_3) are added to the main switch (T_1) to cancel the charge-injection occurring at the gate of M_1 , as depicted in figure 4.15.

To minimize the effect of mismatches between copies of M_1 , the transconductance g_m can be made small, thus reducing the drain current's sensitivity to V_{gs} variations. To achieve this task, a secondary current source, I_2 , is added in parallel with M_1 to sink about 90% of I_{ref} [37]. Since M_1 only sinks the remaining 10% of I_{ref} , its g_m can be relatively small.

To achieve a small g_m , the W/L aspect ratio should be made as small as possible. Also, having a large W and an especially large L transistor increase C_{gs} and improves the matching of the current cells. Therefore, charge-injection and leakage current effects are reduced in accordance with equation C.2. However, there is a limitation on how small the W/L aspect

returns to the original state and the next cell is calibrated. The dummy cell's I_{out} is now available for the next cell.

Figure 4.16 shows an example when cell 1 is under calibration for a 4-bit DAC. This technique ensures that there are always 15 equal currents available at the output terminal, hence allowing the DAC to operate uninterrupted.

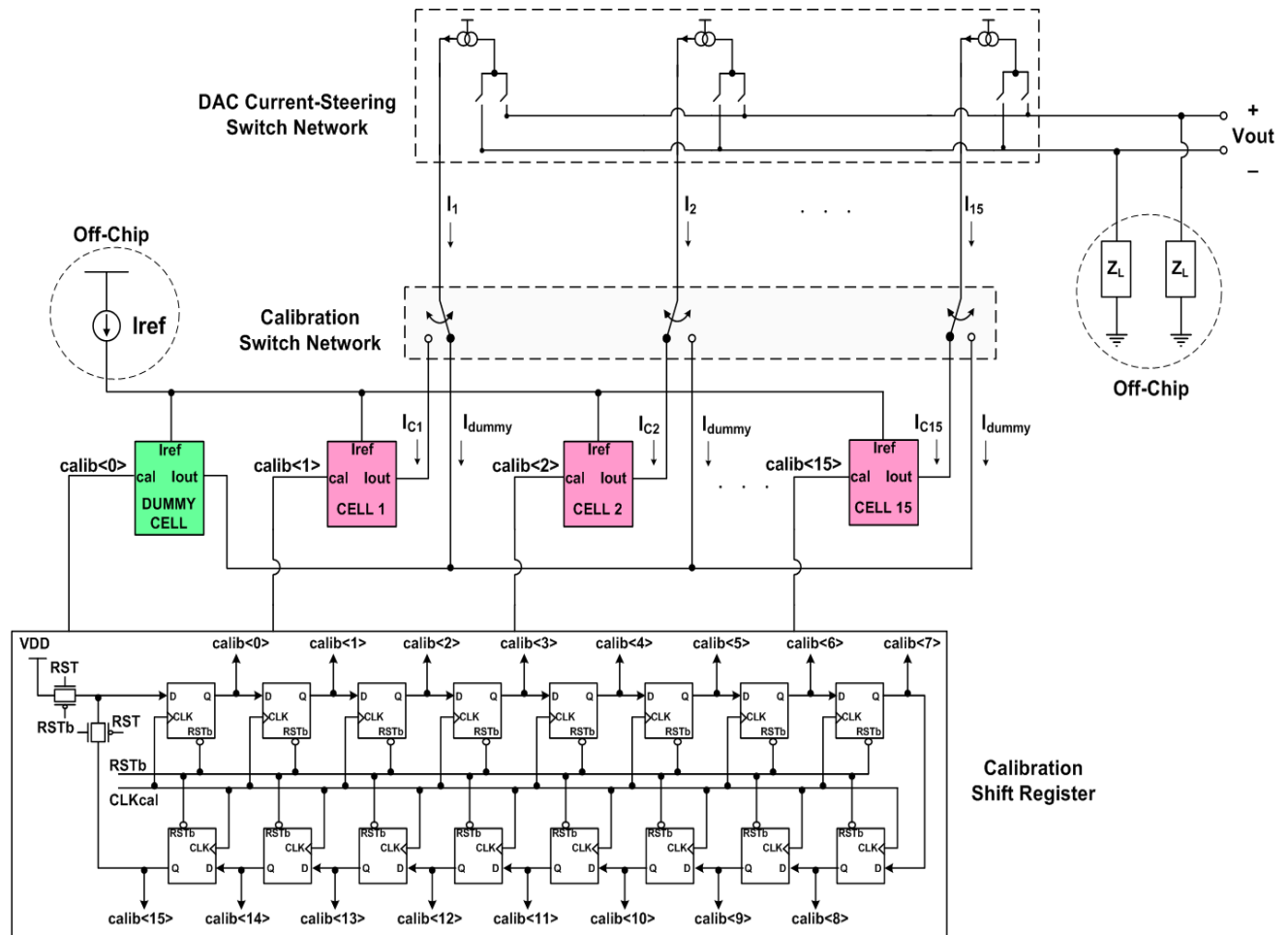


Figure 4.16: Continuous current calibration system for 4-bit DAC

The calibration circuitry, which only consists of a charge-storage MOS transistor and switches, requires no external components. Thus, it can be integrated together with the DAC current-steering cells. The calibration simulation results will be shown together with the current-steering DAC in the next section.

4.3.2. Current-Steering Digital-to-Analog Converter

A current-steering DAC (CS-DAC) is a common choice for high-speed data conversion where each unit current cell switches a current to either output or ground. For a differential CS-DAC configuration, which has a high immunity to common mode noise, the current-steering cell switches the current to either output or its complement.

Current-Steering Cell Configuration

Figure 4.17(a) shows the bias current mirror circuitry, which replicates an off-chip current source to generate I_{ref} and a current array, $I_c < 15 : 0 >$, to bias the secondary calibration source I_2 . Here, simple current mirrors are used instead of cascode current mirrors due to the headroom limitation of a 1V analog supply voltage ($VDDa$). Figure 4.17(b) shows the dummy calibration cell schematic which supplies I_{dummy} to whichever current-steering cell being calibrated.

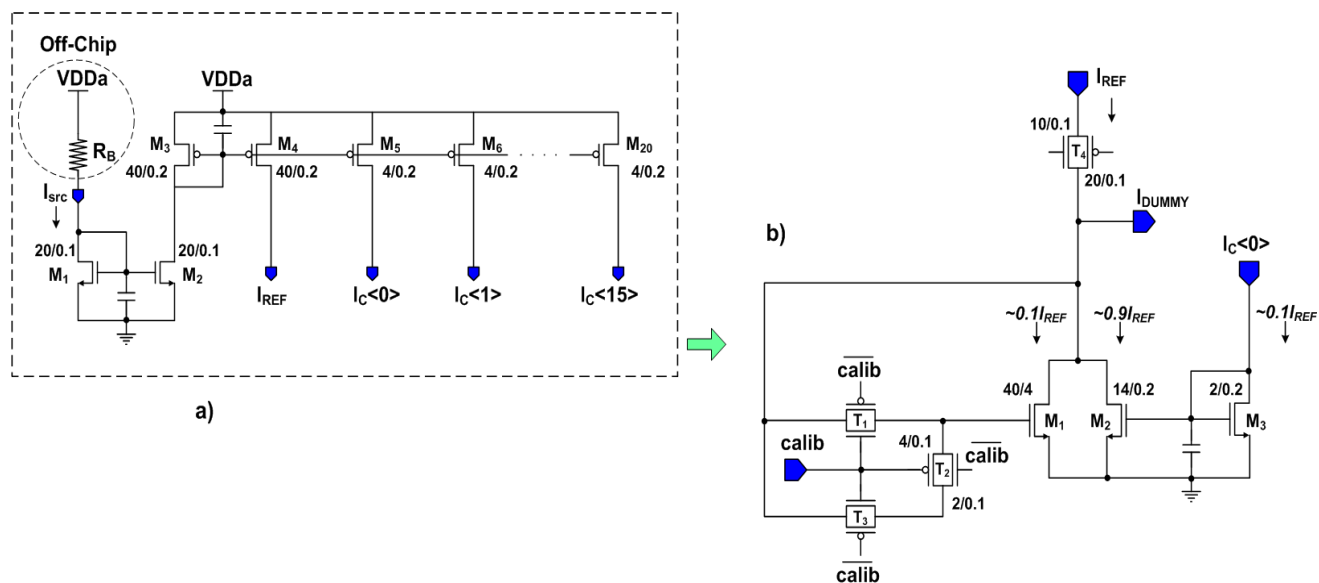


Figure 4.17: a) Bias current mirror b) Dummy calibration cell schematic

Figure 4.18 shows the current-steering cell with self-calibration circuitry. Here, the TG switches, ($T_1 - T_3$) and the MOS transistors ($M_1 - M_3$) belong to the *calibration cell*; whereas the other three TG switches ($T_4 - T_6$) belong to the *calibration switch network*. The remaining

MOS transistors, $M_4 - M_7$, belong to the current-steering cell, in which, M_6 and M_7 are the current-steering (CS) switches.

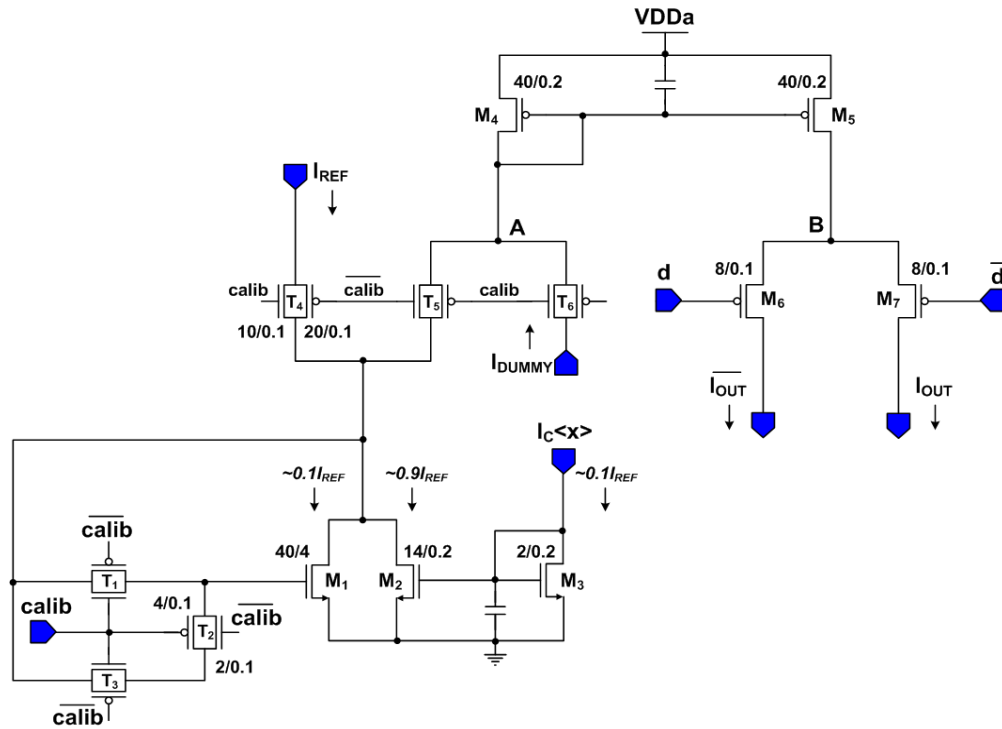


Figure 4.18: Current-steering cell with self-calibration circuitry

In a conventional configuration, the current-steering switches are connected directly to the current source. Specifically, M_6 and M_7 would be NMOS transistors that are connected to node A and the output loads are connected between their open-drain outputs and $VDDa$. In such case, there is no need for the M_4 and M_5 current mirror. However, in 90nm CMOS technology where $VDDa$ is only 1V, the limited headroom does not allow the stacking of M_1 , T_5 , M_6 , and the output load. Therefore, a simple current mirror consisting of M_4 and M_5 is introduced to fold the current over to a new branch that has fewer transistors stacked up, thus allowing higher output swing. Although a cascode current mirror would give a better output resistance and supply noise rejection, again the problem of limited headroom and reduced output swing arises. An alternative to compensate for a simple current mirror is to use longer gate-length transistors; however, this results in a high glitching noise. Overall, the current-steering cell in figure 4.18 provides a compromise.

Output Swing and Noise Estimation

Once the current-steering schematic is chosen, the next task is to determine the appropriate output swing, such that it not only ensures the current mirror's functionality but also meets the SNR requirement. For an analog supply voltage of 1V, there is a little available headroom to start with.

To maintain the current mirror's functionality, namely keeping M_5 in saturation, the drain-source voltage of M_5 requires at least 300mV (i.e.: $V_{ds5} \approx 300\text{mV}$). Since M_6 and M_7 operate as full-swing switches, there is about 100mV drop across each of them. This leaves at most 600mV per side for the output swing as shown in figure 4.19(a).

To meet the required SNR of 56dB, the output swing has to be large enough to sufficiently overcome the output noise. Assuming that the main noise source is dominated by thermal noise and neglecting flicker ($1/f$) noise at low frequency, the output noise can be modelled as illustrated in figure 4.19(b).

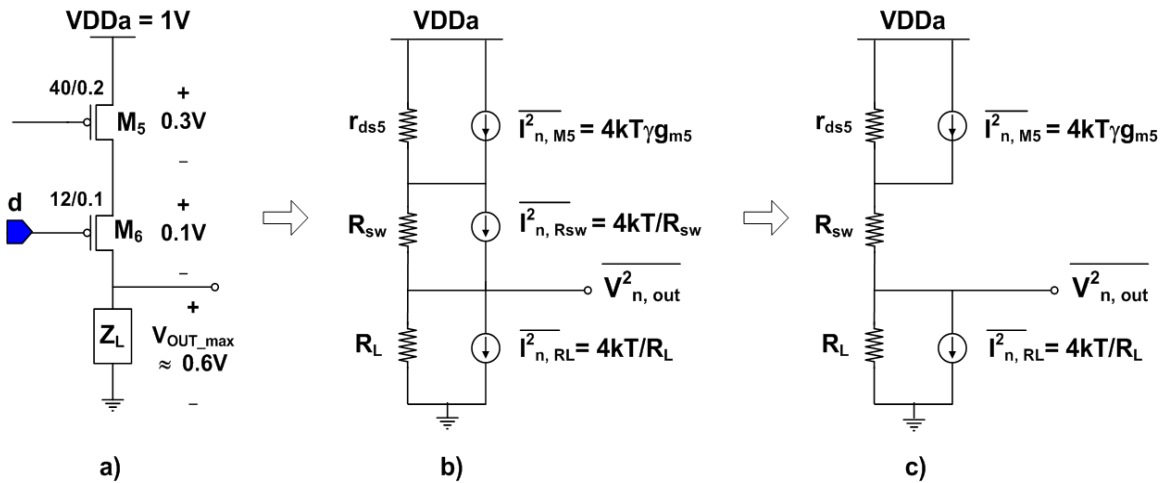


Figure 4.19: a) Output swing b) Output noise model c) Simplified output noise model

Here, the current-steering switch is represented by its ON resistance, R_{sw} , and the off-chip load is represented by a passive load resistance, R_L . The thermal noise of a long-channel MOS operating in saturation can be represented as a current source, $\overline{I_{n, M5}^2}$, connected between its drain and source terminals. The thermal noise of a resistor is a current source, $\overline{I_{n, R}^2}$, connected

in parallel [46]. The simple representations for these noise sources are:

$$\overline{I_{n,M5}^2} = 4kT\gamma g_m \quad \text{and} \quad \overline{I_{n,R}^2} = 4kT/R \quad (4.3)$$

where γ has a value of 2/3 for long-channel transistors but higher for deep sub-micron transistors. Its exact value varies depending on the CMOS process and is still under research. For example, in [47], γ has a value around 1.6 and 1.8 for PMOS and NMOS, respectively.

Using superposition and assuming that as long as, $r_{ds5} \gg (R_{sw} + R_L)$:

$$\overline{V_{n,out(r_{ds5})}^2} = \left(\frac{r_{ds5}R_L}{r_{ds5} + R_{sw} + R_L} \right)^2 \overline{I_{n,M5}^2} \approx \overline{I_{n,M5}^2} R_L^2 \quad (V_{rms}^2/Hz) \quad (4.4)$$

$$\overline{V_{n,out(R_{sw})}^2} = \left(\frac{R_{sw}R_L}{r_{ds5} + R_{sw} + R_L} \right)^2 \overline{I_{n,R_{sw}}^2} \approx 0 \quad (V_{rms}^2/Hz) \quad (4.5)$$

$$\overline{V_{n,out(R_L)}^2} = (R_L || (r_{ds5} + R_{sw}))^2 \overline{I_{n,R_L}^2} \approx \overline{I_{n,R_L}^2} R_L^2 \quad (V_{rms}^2/Hz) \quad (4.6)$$

Thus, $\overline{V_{n,out}^2}$ can be simplified as depicted in figure 4.19(c) and:

$$\overline{V_{n,out}^2} \approx (\overline{I_{n,M5}^2} + \overline{I_{n,R_L}^2}) R_L^2 = 4kTR_L(\gamma g_m R_L + 1) \quad (V_{rms}^2/Hz) \quad (4.7)$$

Equation 4.7 suggests a small value for R_L to minimize the output thermal noise, but this would also decrease the output swing. In this work, the current-steering cells are designed for $R_L=50\Omega$ to ease the impedance-matching with the 50Ω test equipment.

Based on the value $R_L=50\Omega$ and simulations under nominal conditions, an output swing around 500mV per side or 1V differential would sufficiently yield an SNR of 56dB.

Output Load Configurations

Since the CS-DAC outputs differential currents, the loads at the open-drained outputs can be either passive or active, as depicted in figure 4.20. Both contain a resistor which converts differential currents into a differential voltage:

$$V_{od} = V_{out+} - V_{out-} = (I_{out+} - I_{out-})R_L \quad (4.8)$$

For a passive load in figure 4.20(a), the resistors are connected between the CS-DAC output and ground. The advantage of a passive load is that there is high bandwidth due to its simple open-loop configuration. However, the downside is that the output swing is

limited by the available voltage headroom (i.e.: 600mV max) to keep the current source (M_5) in saturation. In addition, the CS-DAC's output resistance per side, R_{out} , varies slightly depending on the number of active current cells in use. R_{out} can be approximated as:

$$R_{out} \approx R_L \parallel \frac{r_{o5} + R_{sw}}{N_{cs}} \quad (4.9)$$

where r_{o5} is the output resistance of M_5 , R_{sw} is the on resistance of switch M_6 , and N_{cs} is the number of active current cells in use. The variations in R_{out} directly correspond to the variations in the output LSB step size, which is a highly undesirable effect that degrades the CS-DAC's linearity performance.

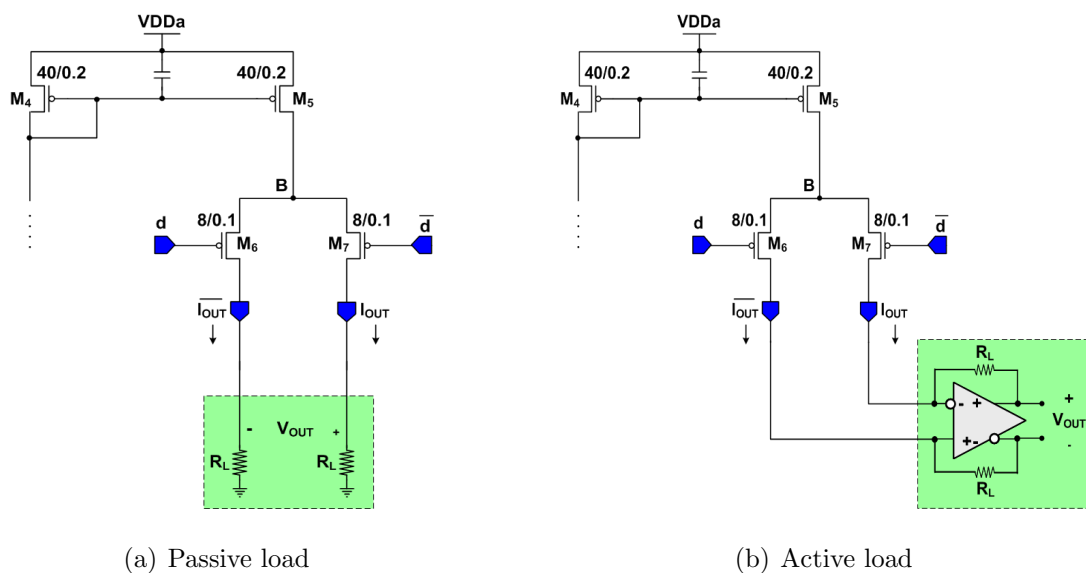


Figure 4.20: Current-steering DAC output load options

For an active load in figure 4.20(b), the resistor is connected in feedback through a differential opamp. Since the opamp's input impedance is much higher than R_L , all current will flow into R_L . The active load offers higher swing than that of the passive load since the output currents are now connected to the opamp's inputs which act like AC virtual ground. Also, since R_{out} looking into V_{out} is constant, the active load does not suffer V_{out} variations as in the case of a passive load. The disadvantages of an active load are limited bandwidth due to the close-loop (feedback) configuration and non-idealities from the opamp's design (e.g.: finite gain, offset, bandwidth, etc). An opamp gain of 60dB is sufficient for this design.

Figure 4.21 compares the output stair case for active versus passive load. It shows the increase in V_{out} variations for passive load, as more current is switched to either side. Therefore, an active load is more favourable in this design.

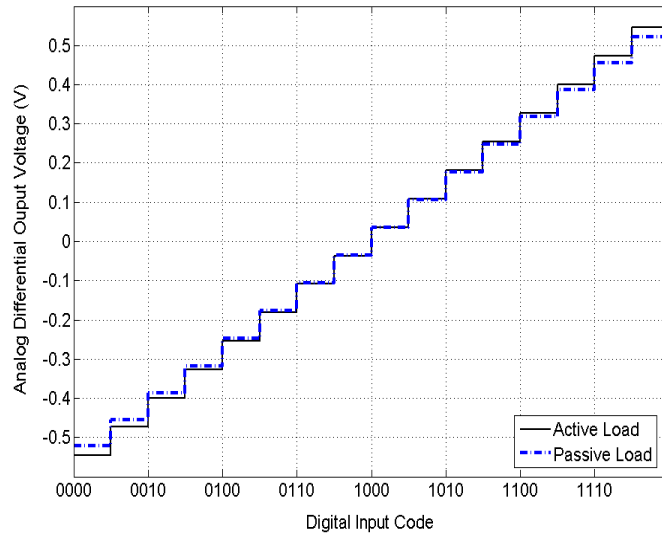


Figure 4.21: Active vs. Passive load output (TT corner)

Deep Sub-micron Design Challenges

This design is fabricated using STMicroelectronics 90nm CMOS with 7 metal layers and 1V supply for the entire chip. While a low power supply lowers the power consumption of a large digital circuitry, it makes the analog design a challenging task. For instance, this supply does not allow the stacking of multiple transistors due to the limited available headroom to maintain the transistor's operating region.

Another major issue of deep sub-micron technology is the high leakage currents which include subthreshold leakage, gate oxide tunneling leakage, junction leakage, hot-carrier injection leakage, gate-induced drain leakage, and punch-through leakage currents [48]. As a result, although the design dissipates minimum dynamic power during switching, its static leakage power begins to catch up. For instance, the simulated total power consumption for the digital front end is around 51mW, of which 23mW (45%) is leakage power. There is a great deal of ongoing research to replace the SiO_2 gate dielectric with a high-k dielec-

tric material to combat increasing leakage currents and to sustain the scaling of CMOS technology.

Lastly, since STMicroelectronics 90nm CMOS was a rather new technology, its model parameters were still not accurate or well defined. For example, the gate resistance was not modelled, hence a gate resistor was added to the transistor with a value according to [49]:

$$R_g = \frac{R_{gsq}}{3} \frac{W_f}{N_f l_G} + \frac{R_{cont}}{N_{cont} N_f} \quad (4.10)$$

where N_f and W_f are the number of fingers and finger width in μm , R_{gsq} and R_{cont} are gate resistance/square and gate contact resistance, and N_{cont} and l_G are the number of gate contacts and gate length in μm , respectively. Table 4.3 summarizes the CS-DAC design including transistor sizes, layout, and drain current.

Table 4.3: Analog Back-end Transistor Properties

CS-DAC	Transistor (NMOS/PMOS)	Size (W/L)	Layout ($N_f \times W_f \times L$)	Current (mA)
Bias Current	M_1	20/0.1	$10 \times 2\mu\text{m} \times 0.1\mu\text{m}$	0.67
	M_2	20/0.1	$10 \times 2\mu\text{m} \times 0.1\mu\text{m}$	0.78
	M_3	40/0.2	$20 \times 2\mu\text{m} \times 0.2\mu\text{m}$	0.78
	M_4	40/0.2	$20 \times 2\mu\text{m} \times 0.2\mu\text{m}$	0.78
	$M_5 - M_{20}$	4/0.2	$2 \times 2\mu\text{m} \times 0.2\mu\text{m}$	0.08
Dummy/ Regular Cell	M_1	40/4	$10 \times 4\mu\text{m} \times 4\mu\text{m}$	0.09
	M_2	14/0.2	$7 \times 2\mu\text{m} \times 0.2\mu\text{m}$	0.69
	M_3	2/0.2	$1 \times 2\mu\text{m} \times 0.2\mu\text{m}$	0.08
	M_4	40/0.2	$10 \times 4\mu\text{m} \times 0.2\mu\text{m}$	0.78
	M_5	40/0.2	$10 \times 4\mu\text{m} \times 0.2\mu\text{m}$	0.72
	$M_6 - M_7$	8/0.1	$2 \times 4\mu\text{m} \times 0.1\mu\text{m}$	0.72
	$T_1 - T_3(W_p/L_p)$	4/0.1	$1 \times 4\mu\text{m} \times 0.1\mu\text{m}$	0
	$T_1 - T_3(W_n/L_n)$	2/0.1	$1 \times 2\mu\text{m} \times 0.1\mu\text{m}$	0
	$T_4 - T_6(W_p/L_p)$	20/0.1	$5 \times 4\mu\text{m} \times 0.1\mu\text{m}$	0.78
	$T_4 - T_6(W_n/L_n)$	10/0.1	$5 \times 2\mu\text{m} \times 0.1\mu\text{m}$	0.78

4.3.3. Analog Back-end Simulation Results

Figures 4.22(a) and 4.22(b) show the DNL offsets Monte Carlo analysis for 15 current cells without and with current calibration, respectively. Without current calibration, the offset range is almost twice as much as the case with current calibration; this corresponds to an SNR improvement of almost 6dB.

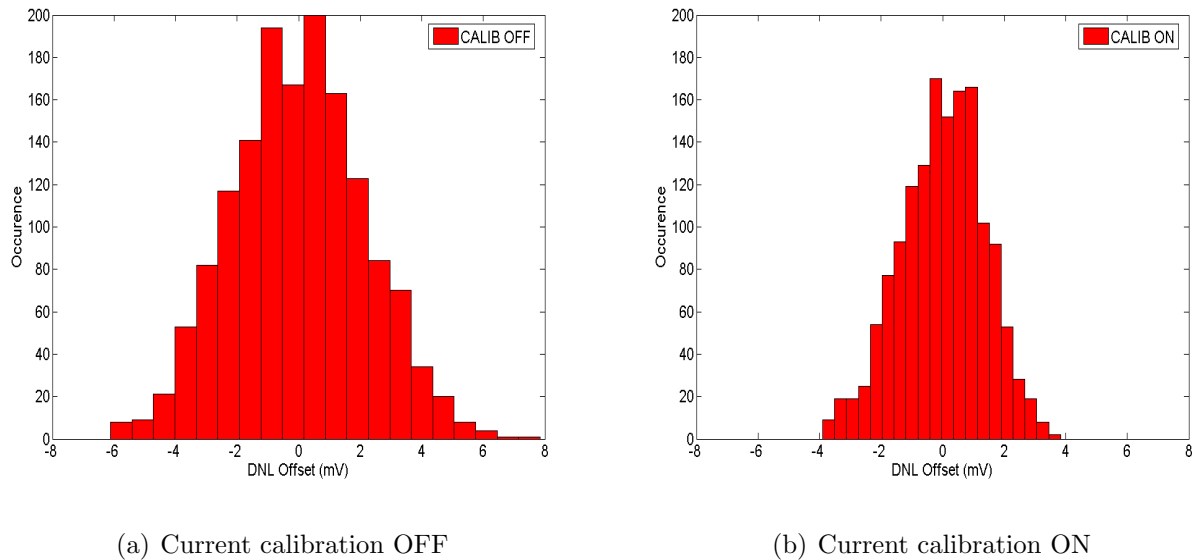
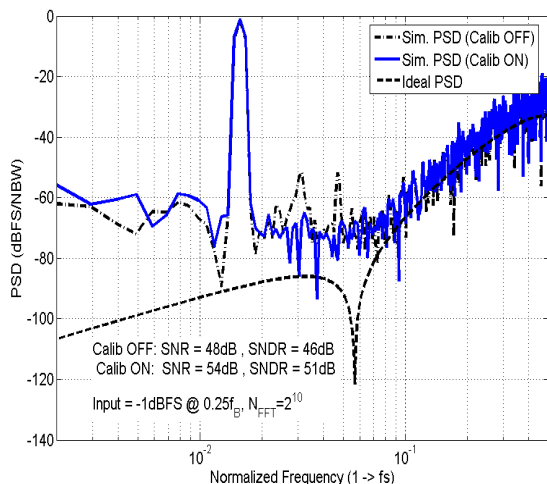
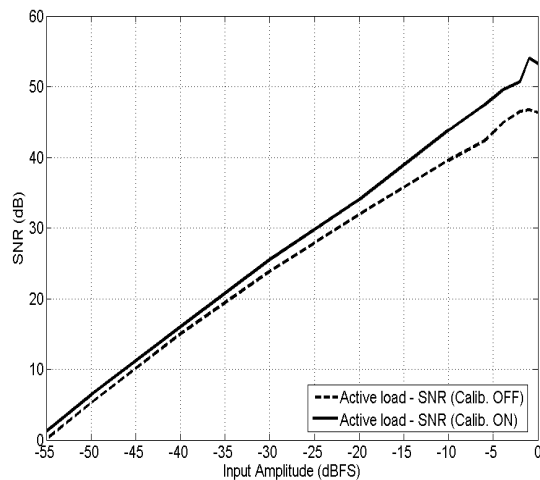


Figure 4.22: DNL offset Monte Carlo analysis (TT corner)

Figure 4.23(a) depicts an example of a TIM $\Delta\Sigma$ -DAC output spectrum with and without current calibration. With calibration on, the inband harmonics are reduced, resulting in higher linearity. Figure 4.23(b) depicts the SNR performance versus input amplitude with and without current calibration. It shows that with calibration on, there is an average SNR improvement of 1dB and 5dB for input amplitude below and above -10dBFS, respectively. Unless mentioned otherwise, all subsequent simulation results will have the current calibration on.

Figures 4.24(a) and 4.24(b) depict the TIM $\Delta\Sigma$ -DAC's accuracy performance versus input amplitude without and with transistor mismatch, respectively. Similarly, figures 4.25(a) and 4.25(b) depict the TIM $\Delta\Sigma$ -DAC's accuracy performance versus input frequency without and with transistor mismatch, respectively. Without transistor mismatch, the peak SNRs

(a) Output spectrum at $0.25f_B$ 

(b) SNR vs. Input amplitude

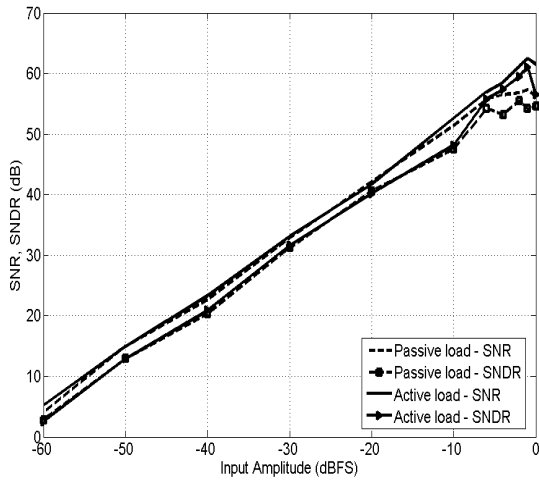
Figure 4.23: TIM $\Delta\Sigma$ -DAC performance with and without current calibration (for active load, TT corner with transistor mismatch)

are 57dB (9.2 bits) and 62dB (10 bits) for passive and active load, respectively. However, with transistor mismatch, the peak SNRs degrade to 50dB (8 bits) and 54dB (8.7 bits), and the dynamic ranges are 52dB and 56dB for passive and active load, respectively.

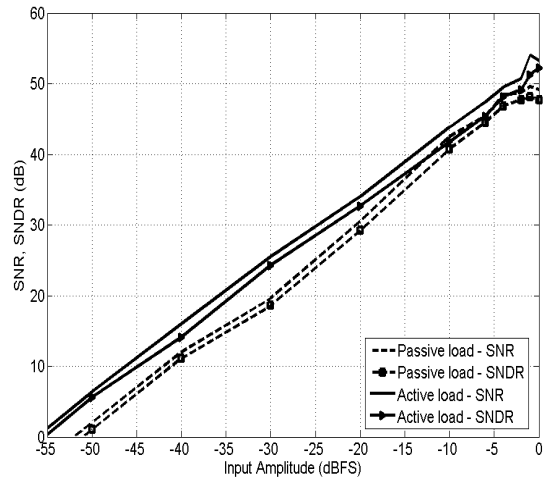
Lastly, table 4.4 shows the simulated power consumption of the TIM $\Delta\Sigma$ -DAC at 1V supply. The digital front-end consumes the most power since it contains the most computations and largest hardware partition.

Table 4.4: TIM $\Delta\Sigma$ -DAC Simulated Power Consumption

Circuit Block	Simulated Power		
	mW	%	$f_{sampling}$
Digital Front-end	51	43	500 MS/s
High-speed Interface	38	32	4 GS/s
Analog Back-end	24	20	4 GS/s
I/Os	7	6	4 GS/s
Total (mW) @ 1V Supply	120		

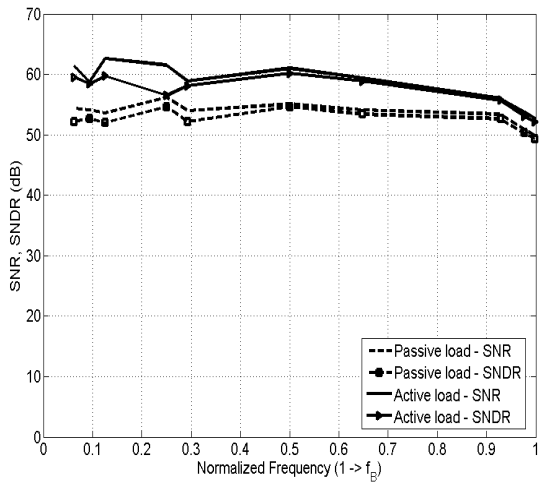


(a) Without transistor mismatch

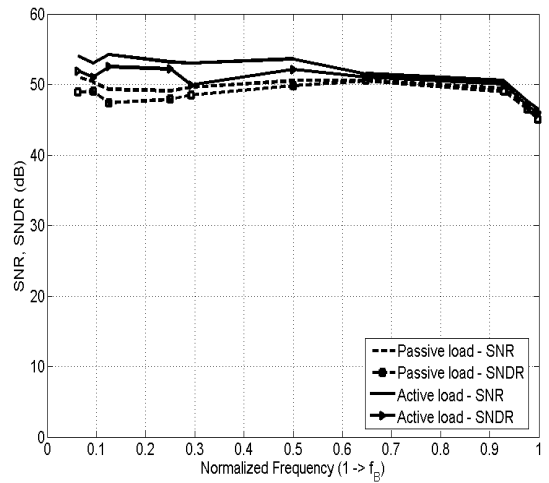


(b) With transistor mismatch

Figure 4.24: TIM $\Delta\Sigma$ -DAC's SNR/SNDR vs. Input amplitude for a single-tone input at $0.25f_B$ (TT corner)



(a) Without transistor mismatch



(b) With transistor mismatch

Figure 4.25: TIM $\Delta\Sigma$ -DAC's SNR/SNDR vs. Input frequency for a single-tone amplitude of 0dBFS (TT corner)

4.4. TIM $\Delta\Sigma$ -DAC Integration

This section presents the TIM $\Delta\Sigma$ -DAC full chip integration. Before the integration, there are two separate sub-blocks that have not been discussed, they are the clock divider and I/O drivers.

The clock divider (figure 4.26(a)) slows down the analog clock (4GHz) by a factor of 8 to generate a digital clock (500MHz). The effects of jitter on the analog and digital clocks have not been fully investigated. For the digital domain, it is assumed that clock jitter does not cause major problems since sufficient timing margin is available at 500MHz. For the analog domain, careful layout and floor planning are needed to minimized clock skew and jitter.

The I/O driver in figure 4.26(b) is designed for flexibility during testing. It accommodates 4 bits of high-speed input or output data. When used as an output, it drives the TIM-IF-DSM's 4-bit multiplexed digital output off chip. When used as an input, it routes the external 4-bit digital data directly to the analog back-end. Therefore, these I/O drivers allow the designer to debug the digital and analog sections separately.

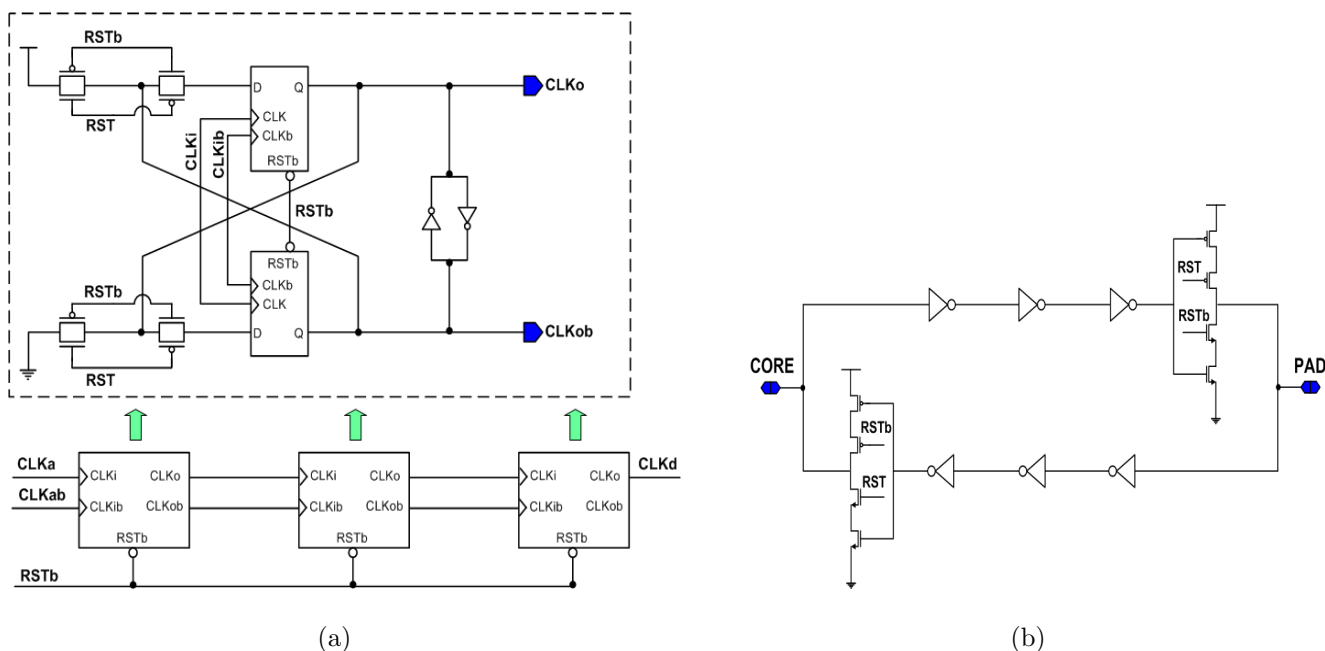


Figure 4.26: a) Divide-by-8 clock divider b) I/O Driver

The chip is designed to have several power supplies for better power management and

testing flexibility. Specifically, the supply for the baseband digital, high-speed interface, and high-speed analog are VDDd, VDDhs, and VDDa, respectively. The I/O drivers also has its own power supply, VDDio. Thus, if necessary, each section of the chip can operate on a different power supply to improve performance. In this work, the entire chip operates on a 1V supply. Also, having multiple power supplies give the flexibility to test each section of the chip separately, while powering down the rest of the chip.

Figures 4.27 and 4.28 show the floor planning and final layout of the TIM $\Delta\Sigma$ -DAC. The chip occupies about $1.52\text{mm} \times 1.52\text{mm}$ of silicon area in 90nm CMOS technology. The layout was pad-limited. The pad frame contains all analog/RF pads without ESD protection to achieve high speed operation. The core area fits within 1.06mm^2 , of which 0.34mm^2 contains the digital standard cells for TIM8-IF and TIM8-DSM.

Both the high-speed digital interface and the analog back-end require a custom layout optimized for high-speed and low mismatch operations. To accommodate a high-speed layout, multi-finger transistors with double-gate connections are used to minimize gate resistance. In addition, high metal layers (M4-M7) are used for high-speed signal routing to minimize substrate parasitic capacitance. To reduce mismatches between current cells, they are routed as close as possible and contained at least 2 dummy gates on each side. Furthermore, a common centroid or finger inter-digitation layout is used to improve matching.

Lastly, each subcircuit (e.g.: current mirror) is surrounded with a ring of substrate contacts to reduce substrate resistance and crosstalk. Multiple N/P-well rings surround each digital or analog section, as shown in figure 4.28, to provide as much isolation as possible.

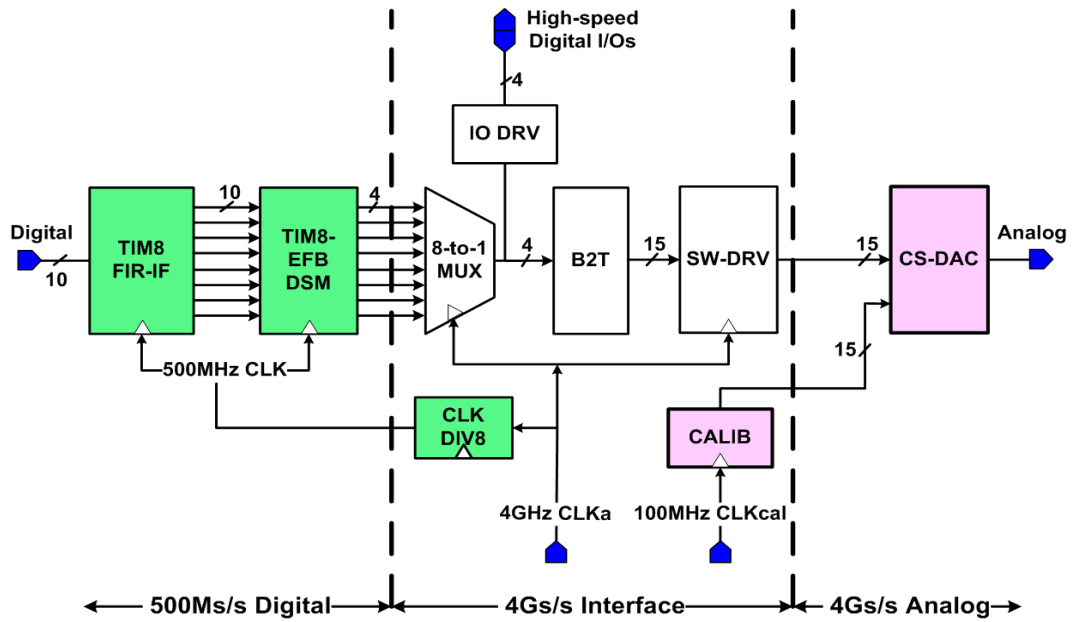


Figure 4.27: TIM $\Delta\Sigma$ -DAC floor planning

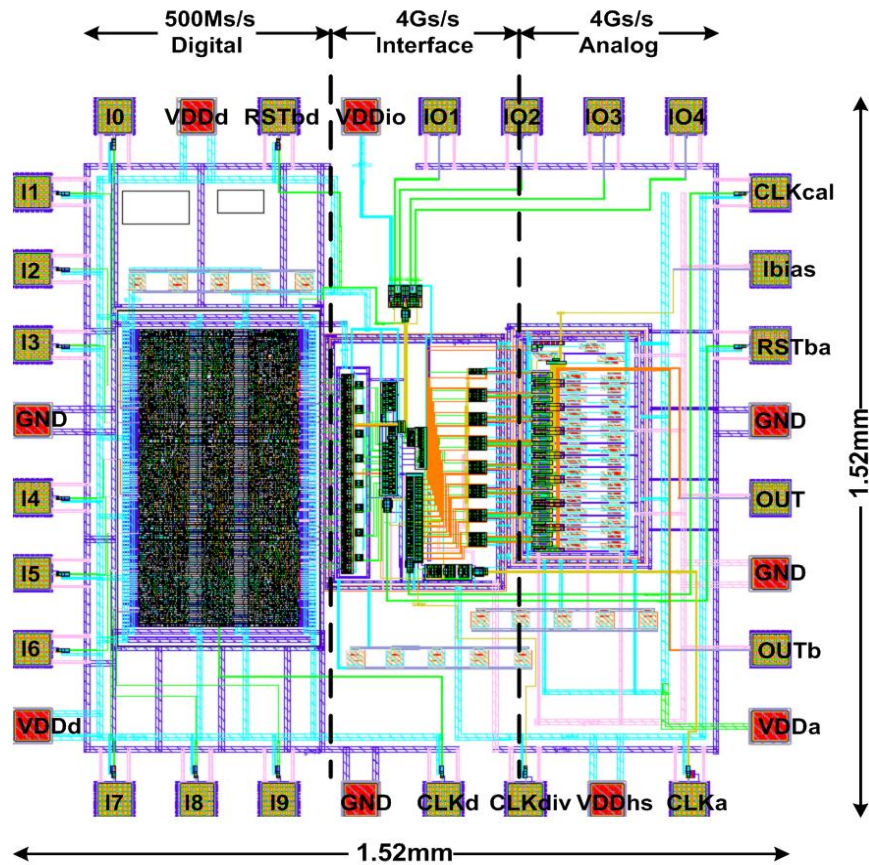


Figure 4.28: TIM $\Delta\Sigma$ -DAC final layout

Time-interleaved $\Delta\Sigma$ -DAC Performance

This chapter presents the experimental results of the fabricated TIM $\Delta\Sigma$ -DAC in STMicroelectronics 90nm CMOS. Specifically, the accuracy and linearity performance of TIM $\Delta\Sigma$ -DAC are measured. The experimental setup and testing issues are also discussed here.

5.1. PCB Design and Test Setup

Figure 5.1(a) shows a die photo of the TIM $\Delta\Sigma$ -DAC chip. The digital front-end (TIM-IF-DSM) was designed using 90nm CMOS standard cell libraries (CORE90GPSVT and CORX90GPSVT). The high-speed digital interface and analog back-end were designed with custom layout as shown in figure 5.1(b).

In order to test the functionalities of the TIM $\Delta\Sigma$ -DAC, the chip must be packaged then integrated onto a PCB. Since this chip operates at a relatively high speed, it is important to select a package and bonding material which are capable of high-speed operation. In this work, the package is a 32-pin ceramic FlatPack (FP32) that uses gold bond wires and supports an operating frequency up to 7GHz. Figures 5.2(a) and 5.2(b) show the packaged chip and its integration with the PCB, respectively.

The PCB supports several testing configurations. Firstly, it permits testing with either a passive or active load. Switches steer the output currents to either a grounded 50 Ω resistor

or a resistor in feedback around a differential opamp. Due to the low voltage supply (1V) and broad bandwidth (250MHz) of the TIM $\Delta\Sigma$ -DAC, it is hard to find a commercial differential opamp which will not limit the performance of the device under test (DUT). For example, a differential opamp from Texas Instruments (THS4508) has sufficient gain and bandwidth; however, its minimum common-mode output voltage is still higher than 1V. Since the DUT's outputs are open-drain PMOS devices, having a drain voltage higher than its supply can damage the entire chip via the forward-biased diode in the N-well. Thus, a passive load of 50Ω is used for all measurements.

Secondly, to allow higher testing flexibility, the PCB is designed to support testing with either an Agilent 93000 SOC tester or an Agilent 81250 parallel bit-error-rate (ParBert) tester as the input source. The full test setup is depicted in figure 5.4. Both the 93K SOC and ParBert testers have the ability to test the entire chip, the digital front-end or the analog back-end alone. For full-chip testing, the tester will send a 10-bit digital pattern (generated from Matlab) to the TIM $\Delta\Sigma$ -DAC. A 2-way 180° power combiner is used to convert the differential outputs into a singled-ended analog output. Lastly, a spectrum analyzer is used to analyze the analog spectrum and to capture data for the Matlab post-processing. Due to some design issues in the digital front-end, which will be discussed in the next section, the full chip was not tested. Thus, only test results from analog back-end and its interface (i.e.: B2T converter & switch driver) are reported.

To test the analog back-end alone, a VHDL simulation is used to generate the 4-bit digital output of the TIM-IF-DSM. This data is then multiplexed in Matlab and transferred to the ParBert, which in turn sends a 4-bit data pattern to the chip's I/Os. Figures 5.3 and 5.5 show the analog test flow and its experimental setup, respectively.

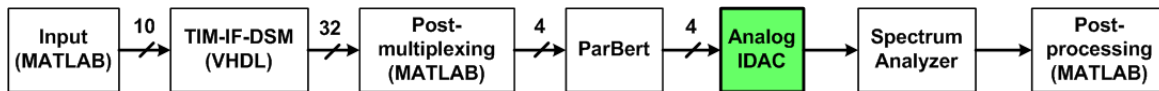


Figure 5.3: Analog back-end test flow

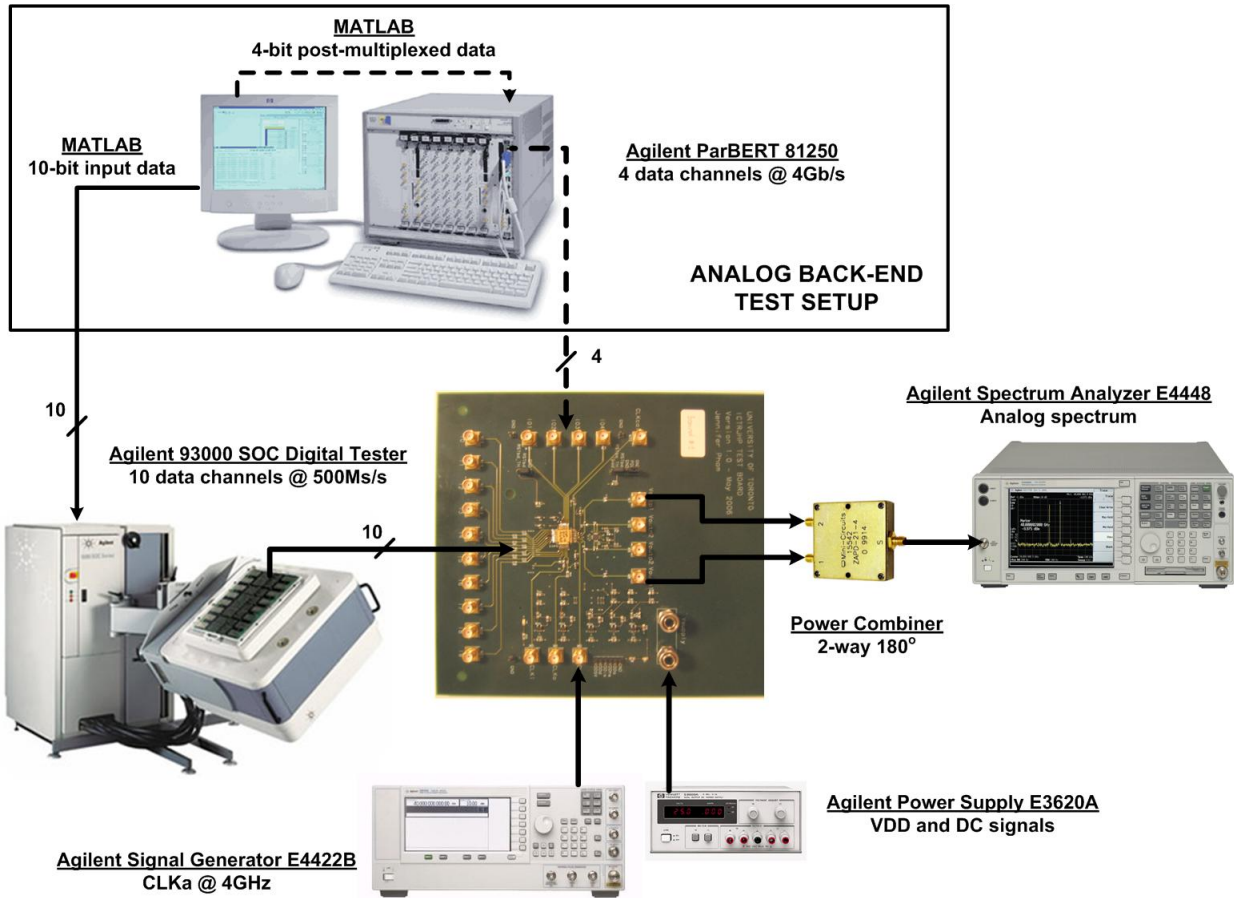


Figure 5.4: Full test setup for Agilent 93K SOC or Agilent ParBert platform

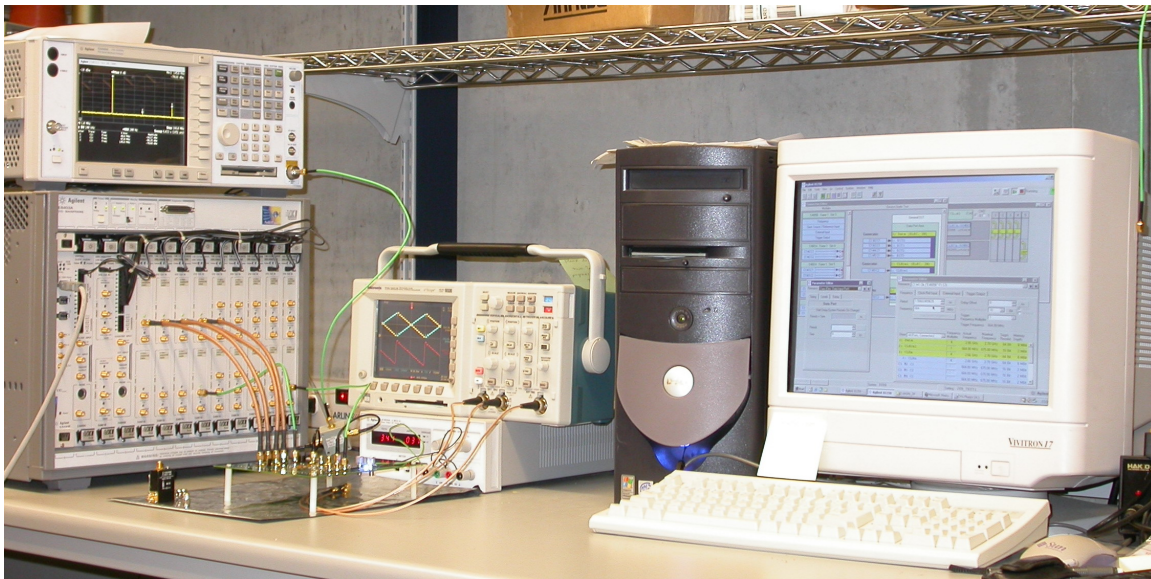


Figure 5.5: Experimental setup for analog back-end

5.2. Digital Design Issues and Solutions

Initial testing of the digital front-end of the fabricated chip revealed four errors in the digital design. Firstly, the time-interleaved paths were multiplexed in the *forward order* instead of *reverse order* (as shown in figure 4.11). Since the digital front-end and analog back-end were designed using different CAD tools, there was not a full transistor-level schematic for LVS purposes. The LVS was done separately for each section before final integration; the full-chip was verified manually, thus resulting in human errors. Also, the “path-reversal” detail was not well-defined at the transistor-level and was not discovered until later. This explains why the “path-reversal” was emphasized throughout chapters 3 and 4.

Secondly, the 2’s complement numbers were not converted into unsigned numbers before thermometer code conversion (as discussed in section 4.2.2). While the TIM-IF-DSM’s outputs were 2’s complement, the B2T converter design, which was done in different CAD tool, used only unsigned test patterns. A simple solution is to add an extra inverter at the MSB as proposed in section 4.2.2. This error could be avoided if the full chip was designed using the same CAD tool, thus eliminating manual inspections during full chip integration.

In addition, two other design errors, which would not cause faulty results but would limit the TIM $\Delta\Sigma$ -DAC’s performance, appeared in the fabricated chip:

1. The “roundoff error reduction scheme” (section 4.1.2): The fabricated TIM-IF utilized the “truncation with no rounding” technique which caused large roundoff errors and limited the SNR of entire TIM $\Delta\Sigma$ -DAC.
2. The “unnecessary overflow bit” (section 4.1.5): The fabricated TIM-DSM was over-designed to have a final sum of 12 bits because a digital limiter was not yet introduced. Since the 12th bit was mostly ‘0’, this resulted in a loss of output amplitude.

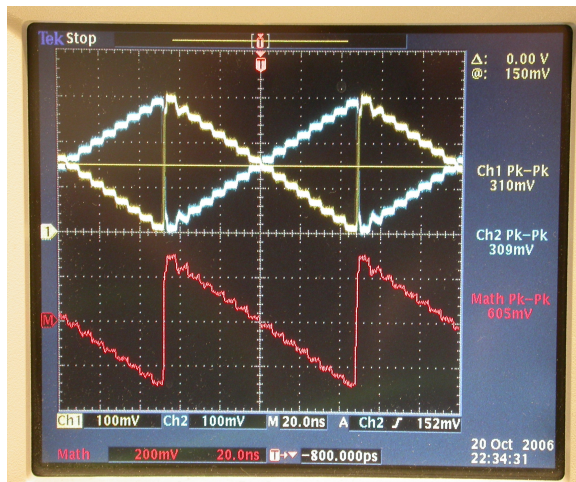
Thus, the digital front-end measurements were omitted since its outputs will not contain meaningful data. However, the simulations in chapter 4 and the analog back-end’s experimental results in this chapter employ a digital design with all of these errors corrected. Here, the corrected VHDL digital front-end results are imported into Cadence for a full-chip mixed-signal simulation; this method detects any system integration or design errors.

5.3. High Speed Analog Measurements

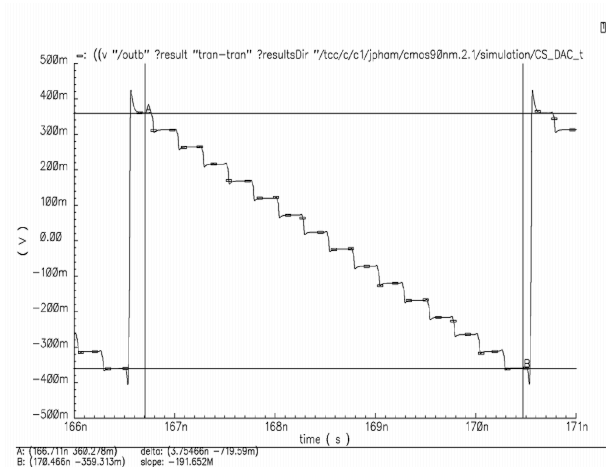
This section presents the experimental results for the analog back-end (i.e.: CS-DAC) with 4-bit post-multiplexed, $\Delta\Sigma$ -modulated inputs. Although the analog test setup is intended for 4GS/s data rate, the available ParBert can only support a data rate up to 2.66GS/s. Thus, unless mentioned otherwise, all measurements were taken at a sampling rate, f_S , of 2.66GS/s; this corresponds to an analog bandwidth, f_B , of 166MHz.

5.3.1. Initial Verifications

A few verifications were carried out before measuring the CS-DAC's dynamic range performance. Firstly, it was important to verify that all current cells are operating by sweeping through all possible digital codes. Figure 5.6(a) shows a stair case transient response of the CS-DAC for digital inputs ranging from 1111 to 0000. The 16 different voltage levels in this figure show that all current cells are functional. The differential output swing (V_{out}) is around 600mV, which corresponds to an average step size of 40mV. While the simulated differential V_{out} is 1V with typical device models, it did not account for post layout parasitics and PVT variations. Simulations at the slow process corner and 105°C resulted in only 720mV peak-to-peak output swing as depicted in figure 5.6(b).



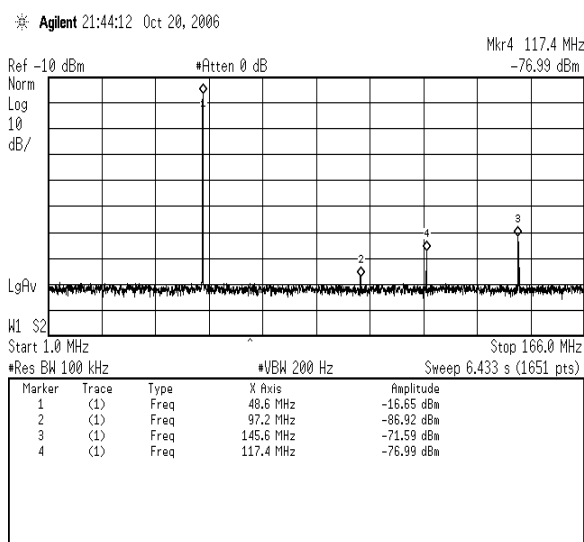
(a) Measured stair case transient



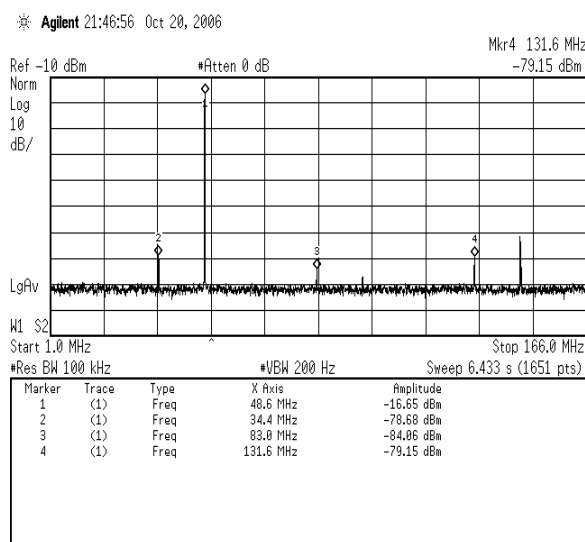
(b) Simulated stair case transient (SS, 105°C)

Figure 5.6: Current-steering DAC stair case transient response

Secondly, the current calibration circuitry was verified. For a passive load, aside from an increase of around 100mV in V_{out} , there was no improvement in accuracy or linearity regardless of calibration being on or off. Instead, having the calibration on introduced some calibration feed-through which mixed with the fundamental signal, and generated inband tones. For example, figures 5.7(a) and 5.7(b) show the calibration feed-through tones for an input at $0.29f_B$ and CLK_{calib} at $0.4f_B$ and $0.2f_B$, respectively. In figure 5.7(a), the second-order intermodulation product (IM2), $f_{signal} + f_{calib}$, shows up at $0.7f_B$ (marker 4). In figure 5.7(b), the IM2 shows up at $0.49f_B$ (marker 3), as well as its harmonics at marker 2 and 4. Thus, the calibration circuitry is switched off for all subsequent measurements.



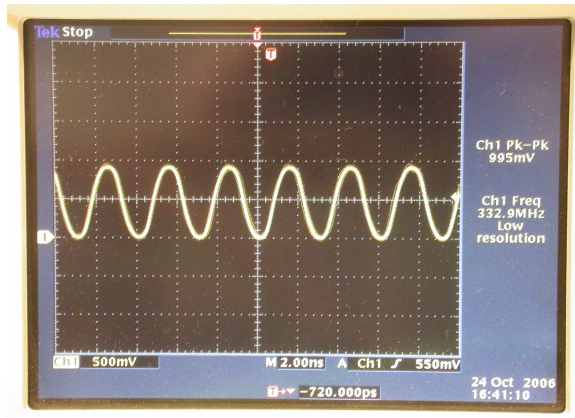
(a) Calibration feed-through for $0.29f_B$ input and $CLK_{calib} = 0.4f_B$



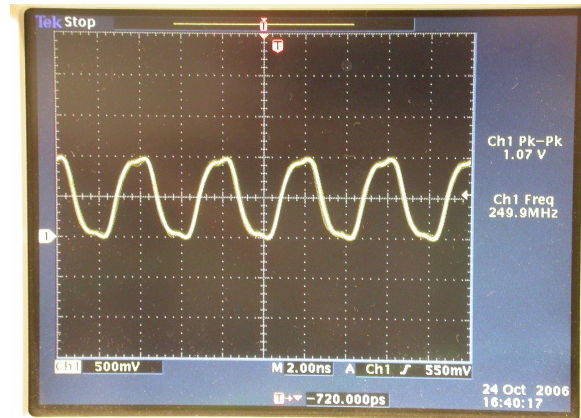
(b) Calibration feed-through for $0.29f_B$ input and $CLK_{calib} = 0.2f_B$

Figure 5.7: Output spectrum with calibration feed-through

Lastly, the clock divide-by-8 circuitry was verified even though it was intended to generate a clock for the digital front-end. Figures 5.8(a) and 5.8(b) show two examples of the clock divider operating with 2.66GHz and 2.0GHz clock inputs, respectively. The divided clocks are 332.9MHz and 249.9MHz, implying the clock divider operates correctly.



(a) Divided-by-8 clock for 2.66GHz input



(b) Divided-by-8 clock for 2GHz input

Figure 5.8: Clock Divider Transient Response

5.3.2. Accuracy Measurements

Figures 5.9(a) and 5.9(b) depict the CS-DAC transient response for a delta-sigma modulated single-tone 0dBFS input at $0.13f_B$ and $0.29f_B$, respectively. Figures 5.10(a)-5.10(d) depict their noise shaped and inband spectra, as well as the inband harmonics.

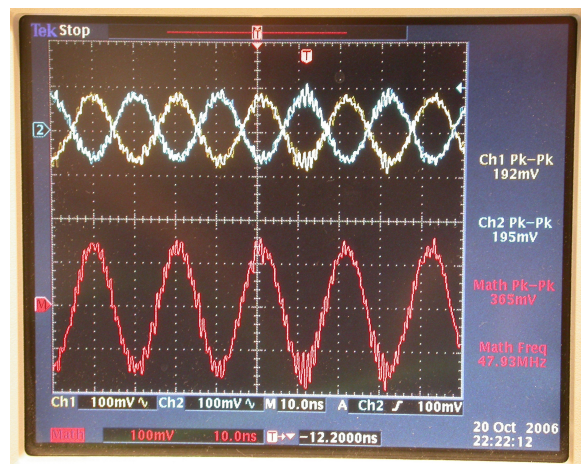
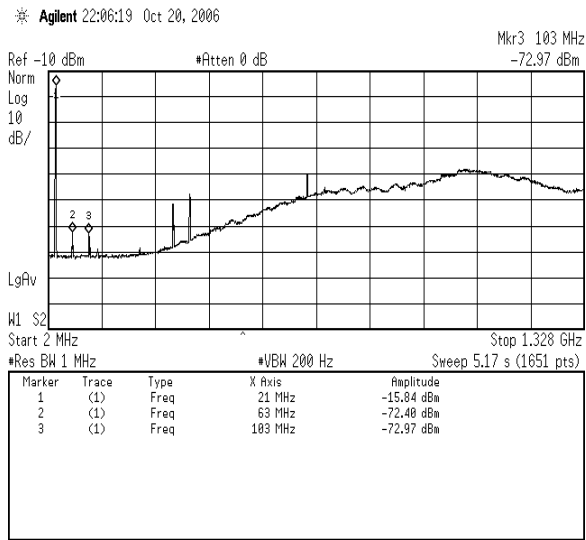
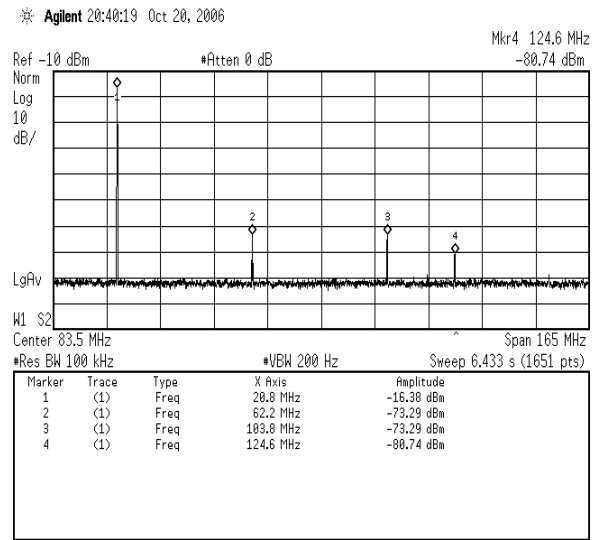
(a) Transient response for $0.13f_B$ input(b) Transient response for $0.29f_B$ input

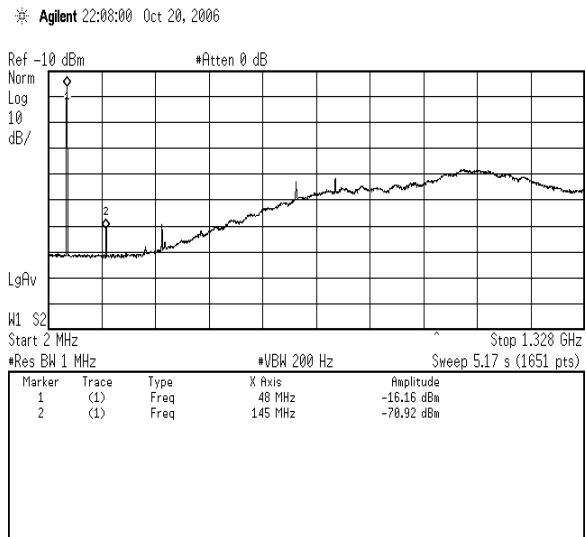
Figure 5.9: CS-DAC transient response for a single-tone, 0dBFS input amplitude (top - single ended outputs; bottom - differential output)



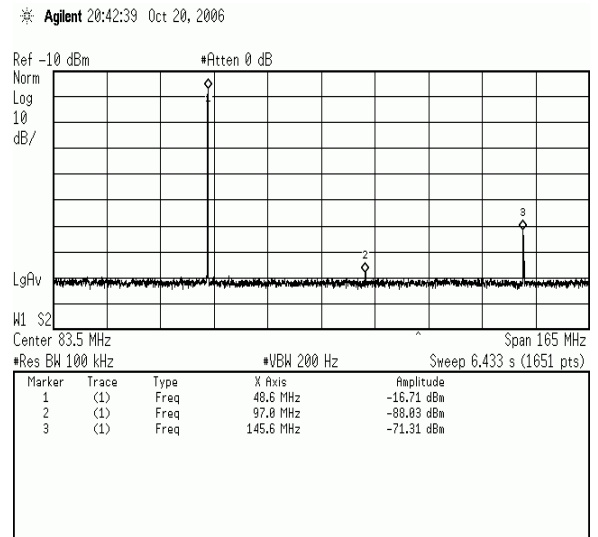
(a) Noise shaped spectrum for $0.13f_B$ input



(b) Inband spectrum for $0.13f_B$ input



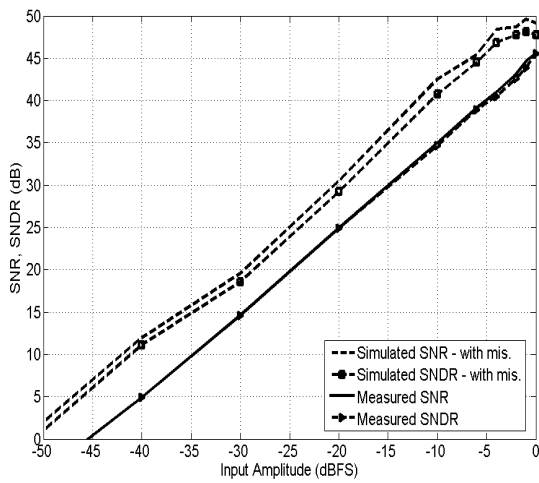
(c) Noise shaped spectrum for $0.29f_B$ input



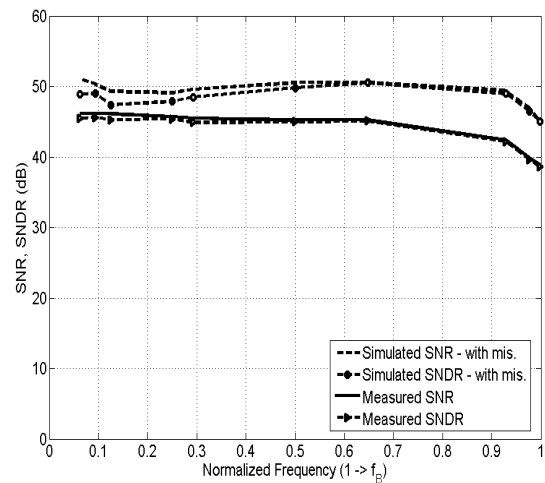
(d) Inband spectrum for $0.29f_B$ input

Figure 5.10: Noise shape and inband spectra for a single-tone, 0dBFS input amplitude at $0.13f_B$ and $0.29f_B$

Figure 5.11 shows the measured and simulated CS-DAC accuracy performance. The measured SNR and SNDR are almost identical since the dominant noise source is the noise floor rather than the harmonic distortions. For the measurements versus amplitude, the input is a single tone at $0.25f_B$; for the measurements versus frequency, the input amplitude is 0dBFS. Figure 5.11(a) shows a peak measured SNR/SNDR of 46dB, which corresponds to an accuracy of 7.3 bits. The dynamic range is also around 46dB. Figure 5.11(b) shows a measured accuracy of at least 44dB (7 bits) up to $0.8f_B$, and 38dB (6 bits) for the entire bandwidth. Compared to the simulated results (passive load with transmistor mismatch), there is an average discrepancy of 5dB due to unaccounted parasitics and PVT variations.



(a) SNR and SNDR vs. Input amplitude

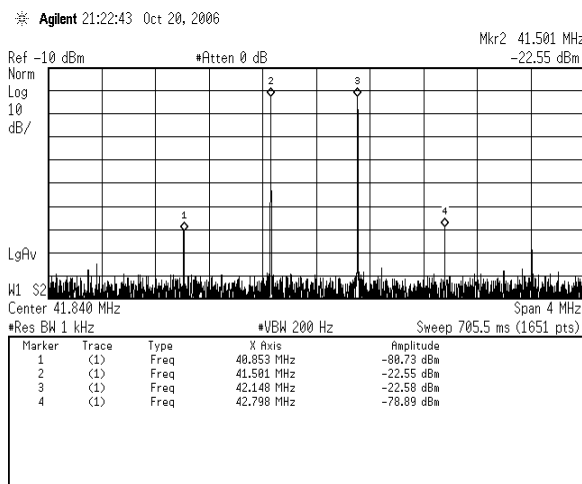


(b) SNR and SNDR vs. Input frequency

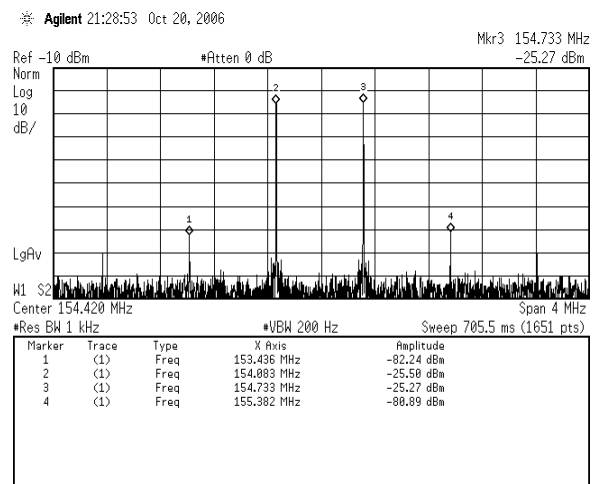
Figure 5.11: CS-DAC accuracy performance with single-tone input and passive load

5.3.3. Linearity Measurements

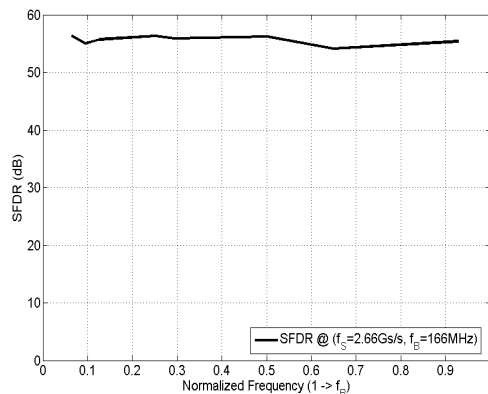
A two-tone test is used to measure the CS-DAC linearity (SFDR) performance; in which, the input contains two tones at f_1 and f_2 . The third-order intermodulation (IM3) products are located at $2f_1 - f_2$ and $2f_2 - f_1$. The SFDR is measured as the amplitude difference between the output tones at f_1 and f_2 and their IM3s. In this test, the two input tones have an amplitude of -6dBFS and are separated by $0.004f_B$ or 0.665MHz. Figures 5.12(a) and 5.12(b) show the two-tone test spectra for input tones near $0.25f_B$ and $0.93f_B$; their SFDR measurements are 56.3dB and 55.4dB, respectively. Figure 5.12(c) shows the SFDR around 55.4dB up to $0.93f_B$ input frequency; this corresponds to a linearity of 8.9 bits.



(a) Two-tone spectrum near $0.25f_B$



(b) Two-tone spectrum near $0.93f_B$

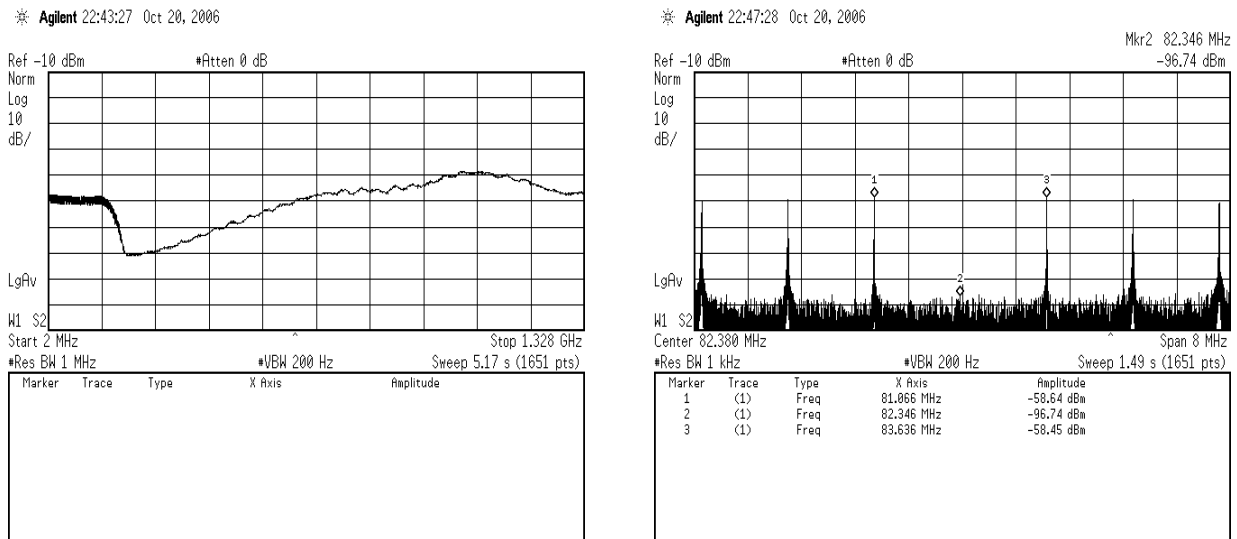


(c) SFDR vs. Input frequency

Figure 5.12: Two-tone spectrum and SFDR measurements

Another linearity measurement is the “missing-tone” test, in which the inband spectrum contains multiple equally-spaced tones except leaving the middle one empty. The intermodulation products of these tones will be concentrated at the empty bin, causing the “missing-tone” to appear. The amplitude difference between the input signal tones and the “missing-tone” is called the *Multi-tone Power Ratio* (MTPR), which reflects the system linearity. This test is particularly relevant for systems employing OFDM since the transmitted spectrum consists of many sub-channels at equally-spaced frequencies.

This experiment uses 128 tones (sub-channels) based on an UWB standard from [6], in which the 64th tone is left empty. For a bandwidth of 166MHz, this corresponds to a sub-channel spacing of 1.3MHz. Figures 5.13(a) and 5.13(b) show the multi-tone noise shaped spectrum and the MTPR measurement, respectively. The measured MTPR is 38dB.



(a) Multi-tone noise shaped spectrum

(b) Multi-tone power ratio measurement

Figure 5.13: Multi-tone Test

5.3.4. Power Consumption

Table 5.1 shows the simulated and measured power consumption for the TIM $\Delta\Sigma$ -DAC at 1V supply. In practice, the digital front-end consumed much more power than that predicted in synthesis, even at half the operating speed. This discrepancy indicates that the power estimation of the digital CAD tools are over optimistic and inaccurate. On the other hand, the analog back-end consumed less power than in simulation; this translates to the loss of output swing V_{out} discussed earlier. This could be the result of inaccuracies in the 90nm CMOS corner models; in particular, the ST 90nm CMOS version 1.0 design kit was still not well defined when the chip was designed.

Table 5.1: TIM $\Delta\Sigma$ -DAC Power Consumption

Circuit Block	Simulated Power			Measured Power		
	mW	%	$f_{sampling}$	mW	%	$f_{sampling}$
Digital Front-end	51	43	500 MS/s	75	70	250 MS/s
High-speed Interface	38	32	4 GS/s	14	13	2.66 GS/s
Analog Back-end	24	20	4 GS/s	15	14	2.66 GS/s
I/Os	7	6	4 GS/s	3	3	2.66 GS/s
Power Distribution						
Total (mW) @ 1V Supply	120			107		

The measured power consumption is 107mW; in which, 32mW is due to the analog prototype sampled at 2.66GS/s and 75mW is due to the digital front-end sampled at 250MS/s. The digital front-end was tested at 250MS/s instead of 333MS/s due to the speed limitation of the Agilent 93K SOC tester. The measured power consumption is 102mW when the digital front-end was sampled at 250MS/s while the rest of the chip was sampled at 2GS/s. Overall, the TIM $\Delta\Sigma$ -DAC power distribution shows that the digital front-end consumes the most power since it contains a large amount of digital circuitry and computation volume.

5.3.5. Performance Summary

Table 5.2 summarizes and compares the TIM $\Delta\Sigma$ -DAC simulated versus measured performance. In both cases, the digital front-end's VHDL behavioural simulation results are used as inputs to the analog back-end. The simulated results are for the active or passive load in typical corner *with* and *without* transistor mismatch (i.e.: *Mis* and *Typ*). The measured results are for the passive load only.

Table 5.2: TIM $\Delta\Sigma$ -DAC Performance Summary

Parameter	Simulated				Measured	Units
	Active load		Passive load		Passive load	
	Typ.	Mis.	Typ.	Mis.		
Peak SNR	62	54	57	50	46	dB
Peak SNDR	60	52	55	48	46	dB
Dynamic Range	63	56	58	52	46	dB
Peak SFDR	-	-	-	-	56	dB
MTPR (128 tones)	-	-	-	-	38	dB
Bandwidth (f_B)	250				166	MHz
Sampling Rate (f_S)	4				2.66	GS/s
Oversampling Ratio (OSR)	8				8	-
Supply Voltage	1				1	V
Power	120				107	mW
Area	1.52mm \times 1.52mm					
Process Technology	STMicroelectronics 90nm CMOS, 7M2T					

Conclusions

In conclusions, this thesis presents the analysis and design of a time-interleaved delta-sigma digital-to-analog converter (TIM $\Delta\Sigma$ -DAC). The digital front-end of the TIM $\Delta\Sigma$ -DAC comprises a 95th-order time-interleaved-by-8 FIR interpolation filter (TIM-IF) and a 3rd-order, 4-bit, time-interleaved-by-8 $\Delta\Sigma$ modulator (TIM-DSM). The analog back-end of the TIM $\Delta\Sigma$ -DAC comprises a 4-bit current-steering DAC with continuous current calibration. The high-speed digital interface between these two domains comprises of an 8-to-1 ring multiplexer, a binary-to-thermometer converter, and 15 switch drivers.

The time-interleaved architecture uses parallelism based on block digital filtering to support a low OSR of 8; this results in a large effective bandwidth for broadband applications. The TIM-DSM utilizes an error-feedback architecture with optimized NTF zero to improve SNR performance. The digital front-end (TIM-IF-DSM) implementation uses CSD representation with rounding scheme for minimum round-off errors, and parallel CSA adders with optimized staging for minimum propagation delays.

The eight parallel outputs of the TIM-IF-DSM is serialized into a single 4-bit stream through an 8-to-1 ring multiplexer. These bits are converted into thermometer codes then into analog signal using 15 current-steering cells. An additional dummy current-steering cell is used to allow continuous current calibration. The differential analog outputs are open-drain which gives the flexibility of having either a passive or an active output load.

The TIM $\Delta\Sigma$ -DAC was designed to operate at 4GS/s with a bandwidth of 250MHz.

The simulation results show a peak SNR of 62dB and 57dB for active and passive load with no transistor mismatch, respectively; the peak SNRs are 54dB and 50dB, with transistor mismatch.

The chip was fabricated in STMicroelectronics 90nm CMOS. The analog back-end was tested with modulated data from VHDL simulation of the digital front-end. It was measured at 2.66GS/s and achieved a bandwidth of 166MHz, an SNR of 46dB and an SFDR of 56dB. At 2GS/s, the prototype consumed 102mW from a 1V supply.

Table 6.1 briefly compares the performance this work with the prior state-of-the-art which utilizes parallelism in $\Delta\Sigma$ modulation (either time-division multiplexing, TDM, or time-interleaving, TIM).

Table 6.1: TIM $\Delta\Sigma$ -DAC Performance Comparisons

Ref	$\Delta\Sigma$ -DAC Architecture	f_s (MHz)	f_B (MHz)	SNR (dB)	SFDR (dB)	Power (mW)	Process/VDD/ Test Results
[25] Clara	TDM2, OFB OSR = 6, 3 rd -order DSM, 6-bit DAC	350	29.16	73.4	76	62	0.13 μ m CMOS / 1.5V / Measured
[26] Khoini- Poorfard	TIM2, EFB OSR = 8, 2 nd -order DSM, 1-bit DAC	352.8	22.05	-	-	-	FPGA Simulated
[27] Choi	TIM4, MASH OSR = 6, 2 nd -order DSM, 6-bit DAC	640	40	73	87	-	0.18 μ m CMOS / 1.8V / Simulated
This Work	TIM8, EFB OSR = 8, 3 rd -order DSM, 4-bit DAC	2660	166	46	56	107	90nm CMOS / 1.0V / Simulated & Measured

6.1. Future Work

Further future work can be done for this design. First of all, the digital front-end design issues discussed in chapter 5 must be corrected before re-fabrication. The challenge of digital round-off errors would still exist, however its effect on the overall performance can be minimized through clever and efficient rounding schemes. There is still ongoing research to minimize the accumulation of round-off errors during digital arithmetic operations.

Secondly, the current-calibration circuitry needs further design to improve the TIM $\Delta\Sigma$ -DAC's linearity. Also, the current-steering DAC's output resistance can be improved to reduce its sensitivity to the number of active current cells under a passive load. However, the issues of deep sub-micron CMOS and low power supply may also limit this design choice.

Thirdly, since this design requires a large amount of hardware integration, extensive post-layout simulations, together with PVT variations and transistor mismatch, will give a better estimate of the actual performance. Furthermore, a full-chip transistor-level simulation, which comprises of a place & route digital front-end and a custom-layout analog back-end, would definitely yield a higher level of design confidence.

Lastly, the idea of parallelism has been widely used in $\Delta\Sigma$ -ADC, yet it is almost forgotten in $\Delta\Sigma$ -DAC. A time-interleaved $\Delta\Sigma$ -DAC is quite promising for future broadband applications which demand high bandwidth and high data rate. The potential of TIM $\Delta\Sigma$ -DAC is certainly an area of research yet to be fully explored.

Appendix A: Conventional $\Delta\Sigma$ Modulator

The function of a digital $\Delta\Sigma$ modulator (DSM) is to reduce the word-length of the input signal to a few bits without affecting its in-band spectrum. Since the reduction in word-length introduces a large truncation error, the modulator must push this added noise outside the band of interest, hence the term “noise shaping”.

The conventional first-order single-bit DSM is shown in figure A.1. It contains three main components: the digital loop filter $H(z)$ (i.e.: Σ), the bit truncator T , and the feedback delay & subtractor (i.e.: Δ). Although this system is highly non-linear, a simple linear model in the z -domain can be used to analyze its operation. Since the main noise component is generated by the truncator T , its linear model is represented by an additive noise source, $E(z)$.

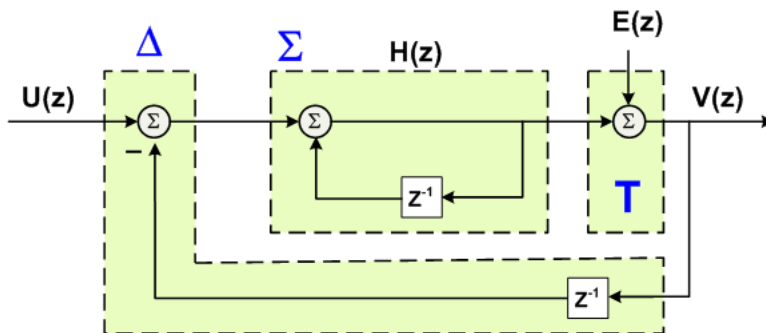


Figure A.1: Linear model of first-order $\Delta\Sigma$ modulator

From figure A.1, the input and output of a first-order DSM can be related as follows:

$$V(z) = U(z) + (1 - z^{-1})E(z) \tag{A.1}$$

Equation A.1 can be written in the general form,

$$V(z) = STF(z)U(z) + NTF(z)E(z) \tag{A.2}$$

where the signal transfer function, $STF(z) = 1$ and the noise transfer function, $NTF(z) = (1 - z^{-1})$. Here, the signal is the exact replica of the input while the truncation noise is shaped by a high-pass response (which suppresses the noise near DC and amplifies the out-of-band noise). For a n^{th} -order lowpass DSM, the system transfer function is:

$$V(z) = U(z) + (1 - z^{-1})^n E(z) \quad (\text{A.3})$$

in which $NTF(z) = (1 - z^{-1})^n$.

If the input signal is a full-scale sine wave with peak amplitude A and the truncation error is assumed to be uniformly distributed, the signal to noise ratio (SNR) for 1st-order DSM can be approximated as [1]:

$$SNR = \frac{9A^2(OSR)^3}{2\pi^2} \quad (\text{A.4})$$

In equation A.4, the OSR is the oversampling ratio which defines how fast the system is oversampled with respect to the Nyquist-rate. It is the ratio between the system sampling frequency, f_S , and twice the signal bandwidth, f_B (i.e.: the Nyquist sampling frequency).

$$OSR = \frac{f_S}{2f_B} \quad (\text{A.5})$$

The resolution of a data converter is often specified by its effective number of bits (ENOB) which is related to the output SNR (in dB) with a sine-wave input by the following equation:

$$SNR = 6.02ENOB + 1.76 \quad (\text{A.6})$$

$$\Rightarrow ENOB = \frac{SNR - 1.76}{6.02} \quad (\text{A.7})$$

In a $\Delta\Sigma$ -DAC, the SNR can be controlled by three main parameters: the OSR, the order of $H(z)$, and the number of truncator bits. Increasing any of these parameters will increase the SNR which directly translates to an improvement in ENOB. However, there are always trade-offs between resolution, speed, power consumption, and design complexity.

Appendix B: TIM $\Delta\Sigma$ -DAC Matlab Results

B.1. Analog Reconstruction Filter

From figure 3.2, since the -3dB bandwidth is limited to 235MHz by the digital IF, the pass-band requirement for this analog LPF should also be around this frequency. Furthermore, from figure 3.9, the out-of-band truncation noise should be attenuated by at least 50dB so that the final spectrum is at the same level as the noise floor (around -100dBFS).

According to ([37], Ch. 14), the order of this analog filter should be at least one order higher than that of the $\Delta\Sigma$ modulator (i.e.: ≥ 4). If the analog filter has the same order as that of the DSM, the slope of the rising truncation noise matches up with the filter's falling attenuation, resulting in a constant spectral density all the way up to $f_S/2$. In addition, this filter must be able to strongly attenuate the high-frequency truncation noise concentrated around $f_S/2$.

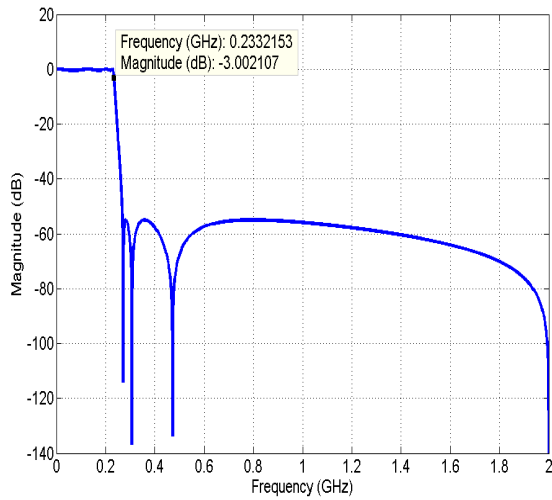
Based on these requirements, an analog LPF can be designed using Matlab. Elliptic filter is chosen due to its high attenuation rate and low ripple passband response. Also, an odd-order elliptic filter has an advantage over that of even-order due to its deep notch at $f_S/2$ which is desirable in this design. The elliptic filter design details are summarized in table B.1 and its responses are shown in figure B.1.

Figure B.2 shows the TIM-IF-DSM output spectrum together with the analog LPF at 0dBFS input amplitude and different input frequencies, ranging from $0.13f_B$ to $0.93f_B$ where $f_B = \frac{1}{16}f_S = 250\text{MHz}$.

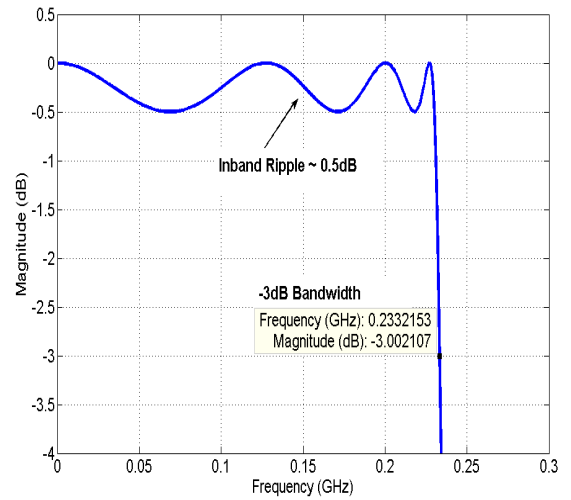
Figure B.3 shows the TIM-IF-DSM response for the system with an ideal "brick-wall" filter and for the system with an analog LPF. Compared to the ideal filter, the analog LPF results in about 2.3dB and 1.2dB degradation in SNR and SNDR, respectively, as depicted

Table B.1: Analog Low-pass Filter Characteristics

	Parameter	Description
Design	Filter Type	Elliptic
	Filter Order	7
	Passband Frequency	230MHz
	Stopband Frequency	240MHz
	Passband Ripple	0.5dB
	Stopband Attenuation	55 dB
Performance	-3dB Bandwidth	235 MHz
	Passband ripple	0.2 dB
	Stopband attenuation	≥ 55 dB

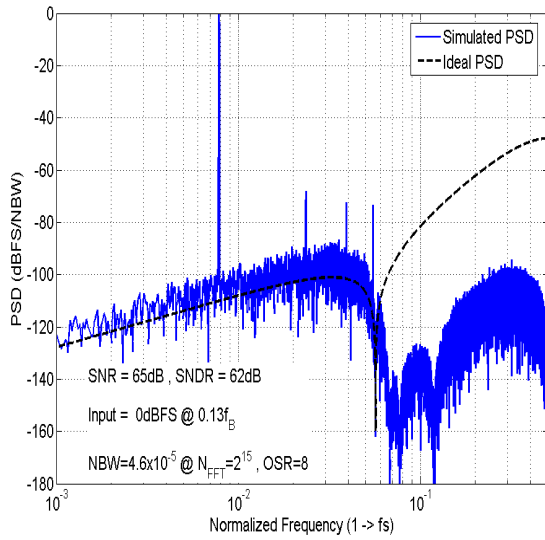


(a) Frequency Response

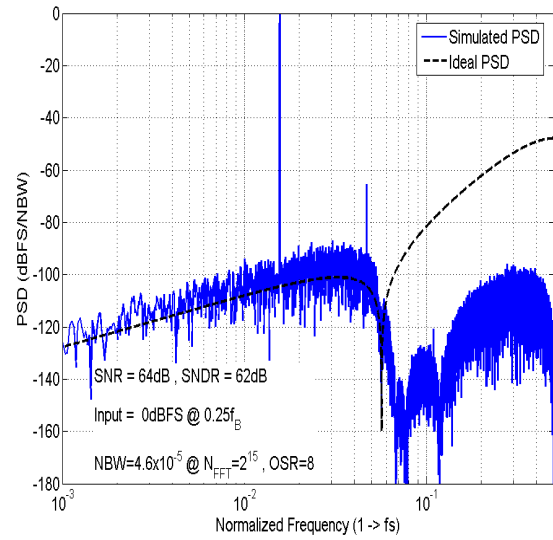


(b) Passband Ripple

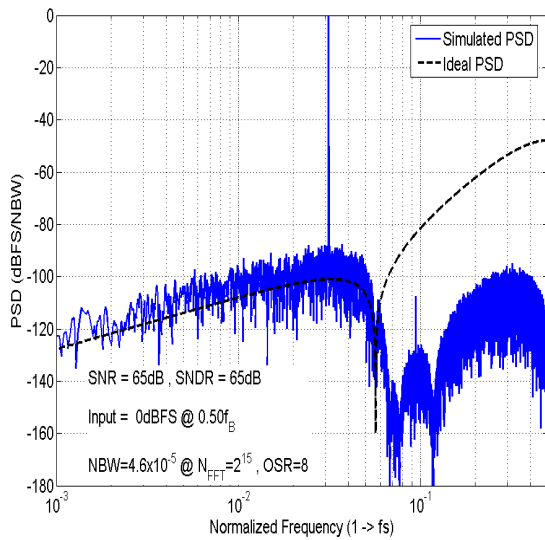
Figure B.1: A 7th-order elliptic analog filter response



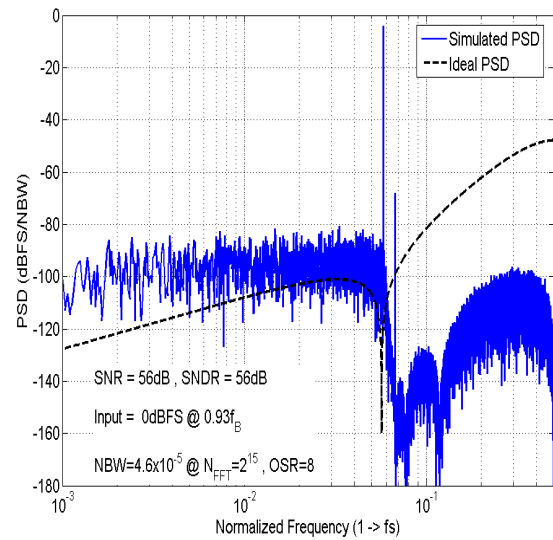
(a)



(b)



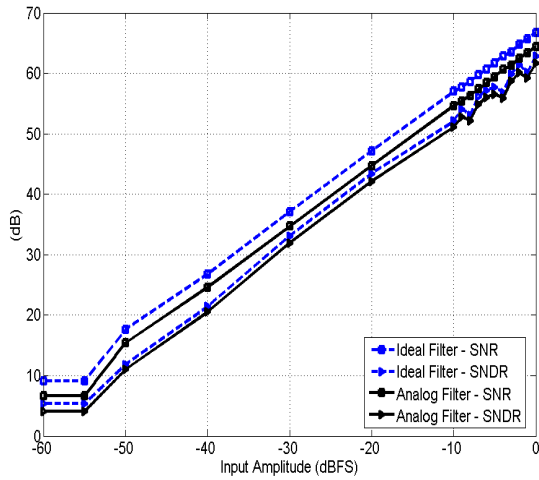
(c)



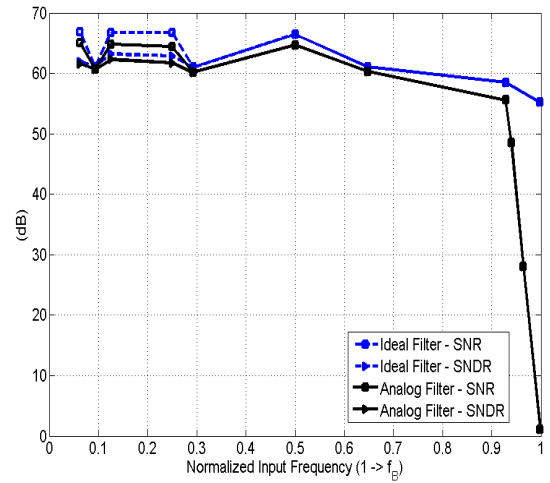
(d)

Figure B.2: TIM $\Delta\Sigma$ -DAC output spectrum with analog LPF for Matlab simulations with 0dBFS input amplitude at different input frequencies a) $0.13f_B$ b) $0.25f_B$ c) $0.50f_B$ d) $0.93f_B$

in figure B.3(a). This degradation is quite acceptable since the full TIM $\Delta\Sigma$ -DAC, including the analog filter, still yields about 9 bits accuracy up to $0.93f_B$, as depicted in figure B.3(b).



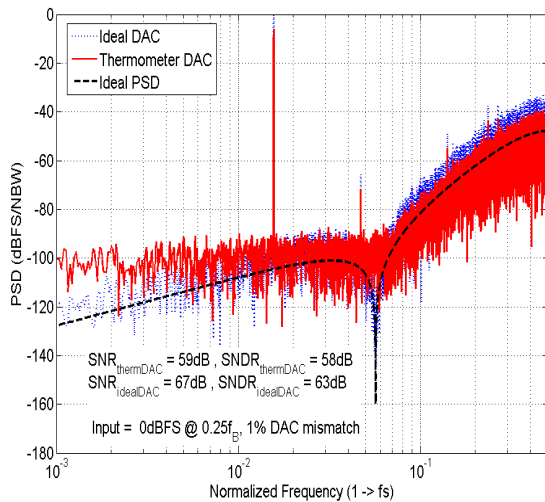
(a) SNR and SNDR vs. Input amplitude



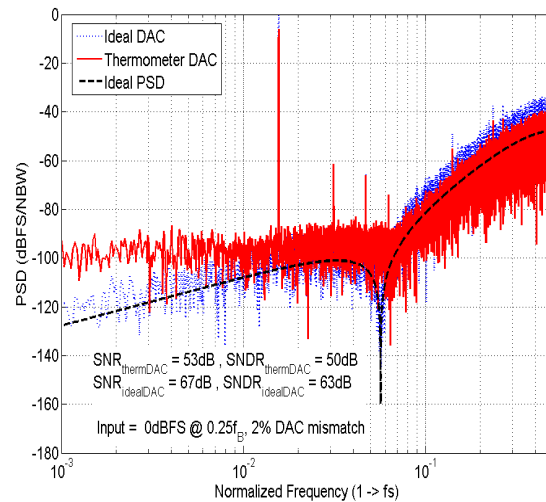
(b) SNR and SNDR vs. Input frequency

Figure B.3: TIM $\Delta\Sigma$ -DAC response with an ideal vs. analog filter

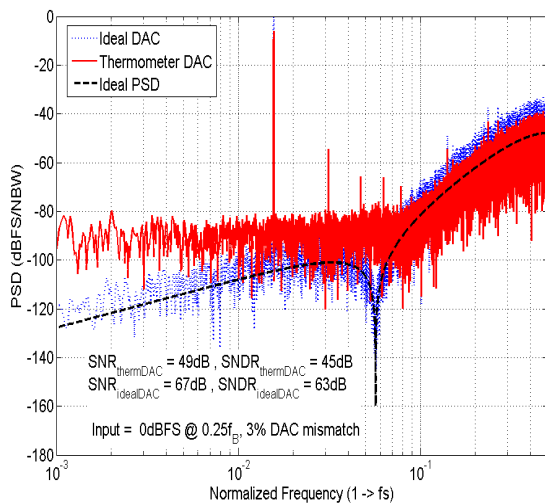
B.2. TIM-IF-DSM Output Spectrum with DAC Mismatches



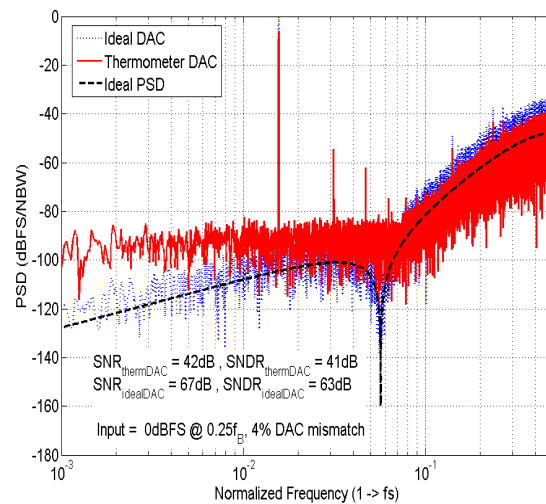
(a) 1% DAC element mismatches



(b) 2% DAC element mismatches



(c) 3% DAC element mismatches



(d) 4% DAC element mismatches

Figure B.4: TIM-IF-DSM output spectrum with thermometer DAC element mismatches

Appendix C: TIM $\Delta\Sigma$ -DAC Implementation

C.1. TIM-IF Coefficients

Table C.1: A 95th-order Time-interleaved-by-8 Interpolation Filter Coefficients

	TIM-IF Path 1				TIM-IF Path 2				TIM-IF Path 3				TIM-IF Path 4			
	Original	Quantized		% error	Original	Quantized		% error	Original	Quantized		% error	Original	Quantized		% error
	Value	Value	CSD		Value	Value	CSD		Value	Value	CSD		Value	Value	CSD	
g'(0)	0	0	-	0	-0.003	-0.004	-2^{-8}	18	-0.007	-0.008	-2^{-7}	11	-0.010	-0.008	-2^{-7}	24
g'(1)	0	0	-	0	0.008	0.008	2^{-7}	4	0.017	0.016	2^{-6}	5	0.024	0.031	2^{-5}	33
g'(2)	0	0	-	0	-0.016	-0.016	-2^{-6}	4	-0.032	-0.031	-2^{-5}	3	-0.046	-0.031	-2^{-5}	32
g'(3)	0	0	-	0	0.030	0.031	2^{-5}	5	0.060	0.063	2^{-4}	5	0.084	0.063	2^{-4}	26
g'(4)	0	0	-	0	-0.056	-0.055	$-2^{-4} + 2^{-7}$	3	-0.114	-0.117	$-2^{-3} + 2^{-7}$	3	-0.163	-0.125	-2^{-3}	23
g'(5)	0	0	-	0	0.135	0.133	$2^{-3} + 2^{-7}$	2	0.293	0.281	$2^{-2} + 2^{-5}$	4	0.463	0.469	$2^{-1} - 2^{-5}$	1
g'(6)	1	1	2^0	0	0.974	0.969	$2^0 - 2^{-5}$	1	0.898	0.875	$2^{-1} + 2^{-2} + 2^{-3}$	3	0.780	0.750	$2^{-1} + 2^{-2}$	4
g'(7)	0	0	-	0	-0.103	-0.109	$-2^{-3} + 2^{-6}$	6	-0.169	-0.156	$-2^{-3} - 2^{-5}$	8	-0.198	-0.188	$-2^{-3} - 2^{-4}$	5
g'(8)	0	0	-	0	0.048	0.063	2^{-4}	31	0.081	0.063	2^{-4}	23	0.098	0.125	2^{-3}	28
g'(9)	0	0	-	0	-0.026	-0.031	-2^{-5}	22	-0.044	-0.031	-2^{-5}	29	-0.053	-0.063	-2^{-4}	17
g'(10)	0	0	-	0	0.014	0.016	2^{-6}	13	0.024	0.031	2^{-5}	33	0.028	0.031	2^{-5}	11
g'(11)	0	0	-	0	-0.007	-0.008	-2^{-7}	16	-0.011	-0.008	-2^{-7}	30	-0.013	-0.016	-2^{-6}	20
	TIM-IF Path 5				TIM-IF Path 6				TIM-IF Path 7				TIM-IF Path 8			
	Original	Quantized		% error	Original	Quantized		% error	Original	Quantized		% error	Original	Quantized		% error
	Value	Value	CSD		Value	Value	CSD		Value	Value	CSD		Value	Value	CSD	
g'(0)	-0.013	0	-	100	-0.013	-0.016	-2^{-6}	20	-0.011	-0.008	-2^{-7}	30	-0.007	-0.008	-2^{-7}	16
g'(1)	0.028	0	-	100	0.028	0.031	2^{-5}	11	0.024	0.031	2^{-5}	33	0.014	0.016	2^{-6}	13
g'(2)	-0.054	-0.063	-2^{-4}	17	-0.053	-0.063	-2^{-4}	17	-0.044	-0.031	-2^{-5}	29	-0.026	-0.031	-2^{-5}	22
g'(3)	0.098	0.125	2^{-3}	27	0.098	0.125	2^{-3}	28	0.081	0.063	2^{-4}	23	0.048	0.063	2^{-4}	31
g'(4)	-0.194	-0.188	$-2^{-3} - 2^{-4}$	3	-0.198	-0.188	$-2^{-3} - 2^{-4}$	5	-0.169	-0.156	$-2^{-3} - 2^{-5}$	8	-0.103	-0.109	$-2^{-3} + 2^{-6}$	6
g'(5)	0.630	0.625	$2^{-1} - 2^{-3}$	1	0.780	0.750	$2^{-1} + 2^{-2}$	4	0.898	0.875	$2^{-1} + 2^{-2} + 2^{-3}$	3	0.974	0.969	$2^0 - 2^{-5}$	1
g'(6)	0.630	0.625	$2^{-1} - 2^{-3}$	1	0.463	0.469	$2^{-1} - 2^{-5}$	1	0.293	0.281	$2^{-2} + 2^{-5}$	4	0.135	0.133	$2^{-3} + 2^{-7}$	2
g'(7)	-0.194	-0.188	$-2^{-3} - 2^{-4}$	3	-0.163	-0.125	-2^{-3}	23	-0.114	-0.117	$-2^{-3} + 2^{-7}$	3	-0.056	-0.055	$-2^{-4} + 2^{-7}$	3
g'(8)	0.098	0.125	2^{-3}	27	0.084	0.063	2^{-4}	26	0.060	0.063	2^{-4}	5	0.030	0.031	2^{-5}	5
g'(9)	-0.054	-0.063	-2^{-4}	17	-0.046	-0.031	-2^{-5}	32	-0.032	-0.031	-2^{-5}	3	-0.016	-0.016	-2^{-6}	4
g'(10)	0.028	0	-	100	0.024	0.031	2^{-5}	33	0.017	0.016	2^{-6}	5	0.008	0.008	2^{-7}	4
g'(11)	-0.013	0	-	100	-0.010	-0.008	-2^{-7}	24	-0.007	-0.008	-2^{-7}	11	-0.003	-0.004	-2^{-8}	18

C.2. TIM-IF Sum Trees

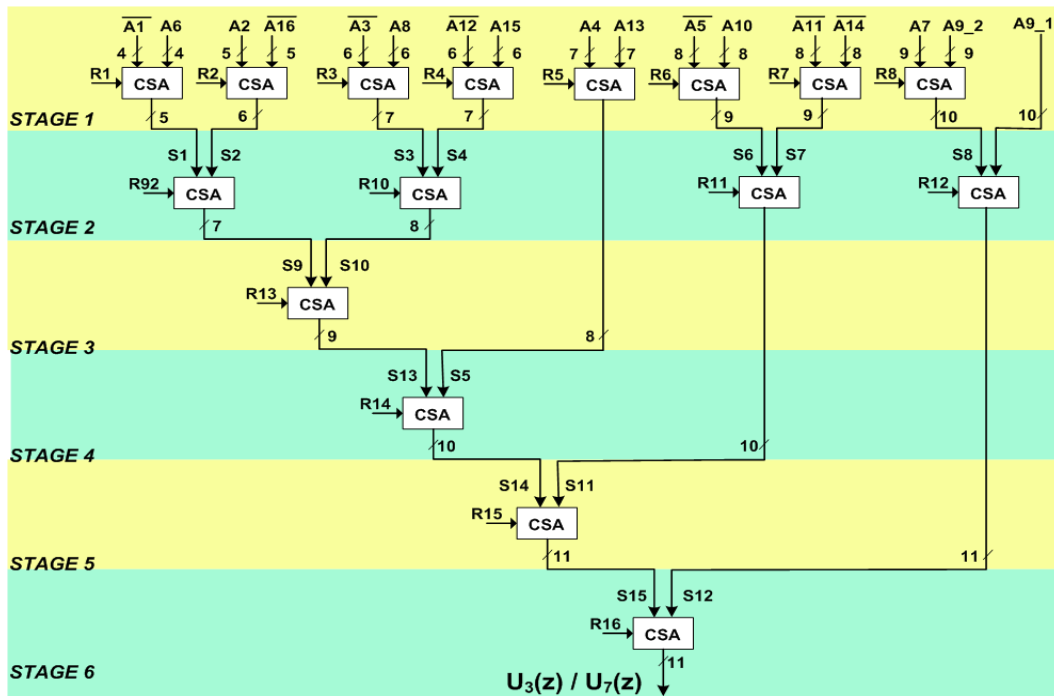


Figure C.1: TIM-IF sum tree for path 3 and 7

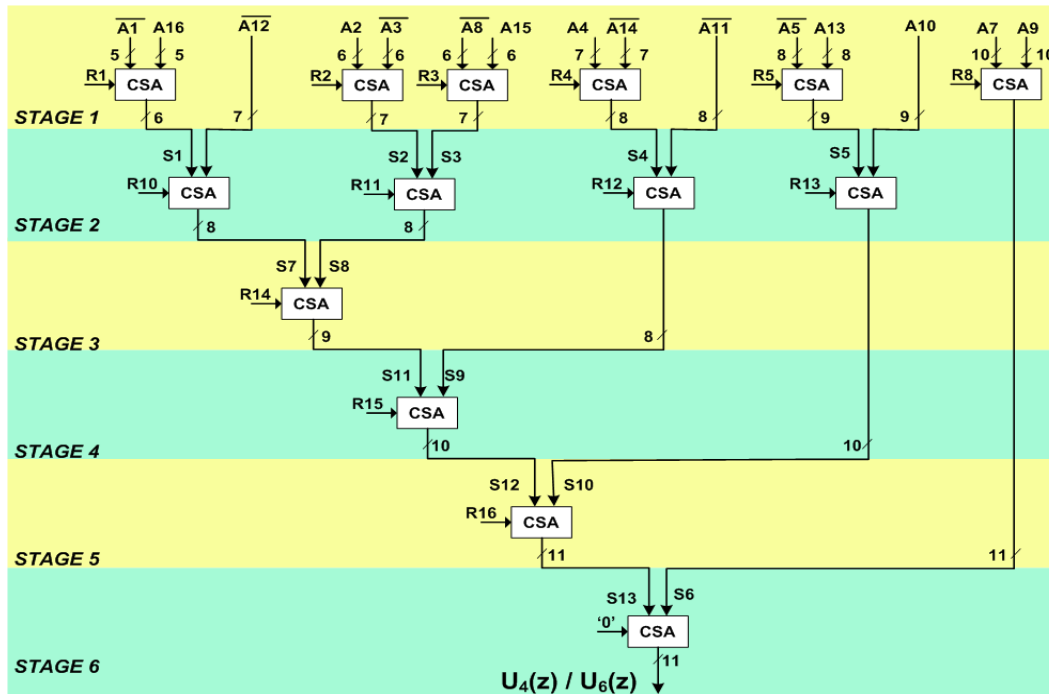


Figure C.2: TIM-IF sum tree for path 4 and 6

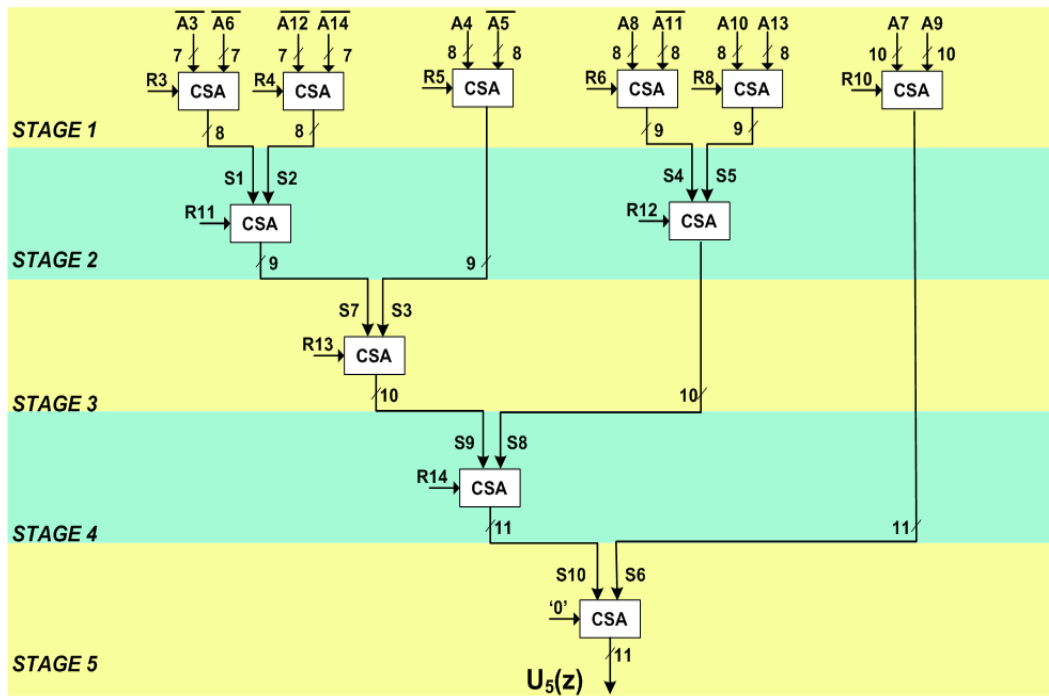


Figure C.3: TIM-IF sum tree for path 5

C.3. TIM-IF and TIM-DSM Timing Synthesis

Table C.2: TIM-IF Synthesized Performance

Description	Propagation Delay (ns)	Timing Margin (ns)
Sum tree 2 or 8	1.66	0.34
Sum tree 3 or 7	1.69	0.31
Sum tree 4 or 6	1.63	0.37
Sum tree 5	1.54	0.46
Path 1	0.44	1.56
Path 2 or 8	1.91	0.09
Path 3 or 7	1.91	0.09
Path 4 or 6	1.91	0.09
Path 5	1.89	0.11

Table C.3: TIM-DSM Synthesized Performance

Description	Propagation Delay (ns)	Timing Margin (ns)
Sum Tree Only	1.61	0.39
Path 1	1.87	0.13
Path 2	1.74	0.26
Path 3	1.74	0.26
Path 4	1.75	0.25
Path 5	1.74	0.26
Path 6	1.74	0.26
Path 7	1.74	0.26
Path 8	1.73	0.27

C.4. Binary-to-Thermometer Converter and Switch Drivers

Figure C.4 depicts the binary-to-thermometer converter schematic with gate re-use and signed-to-unsigned number conversion (by adding an extra inverter for bit $V < 3 >$).

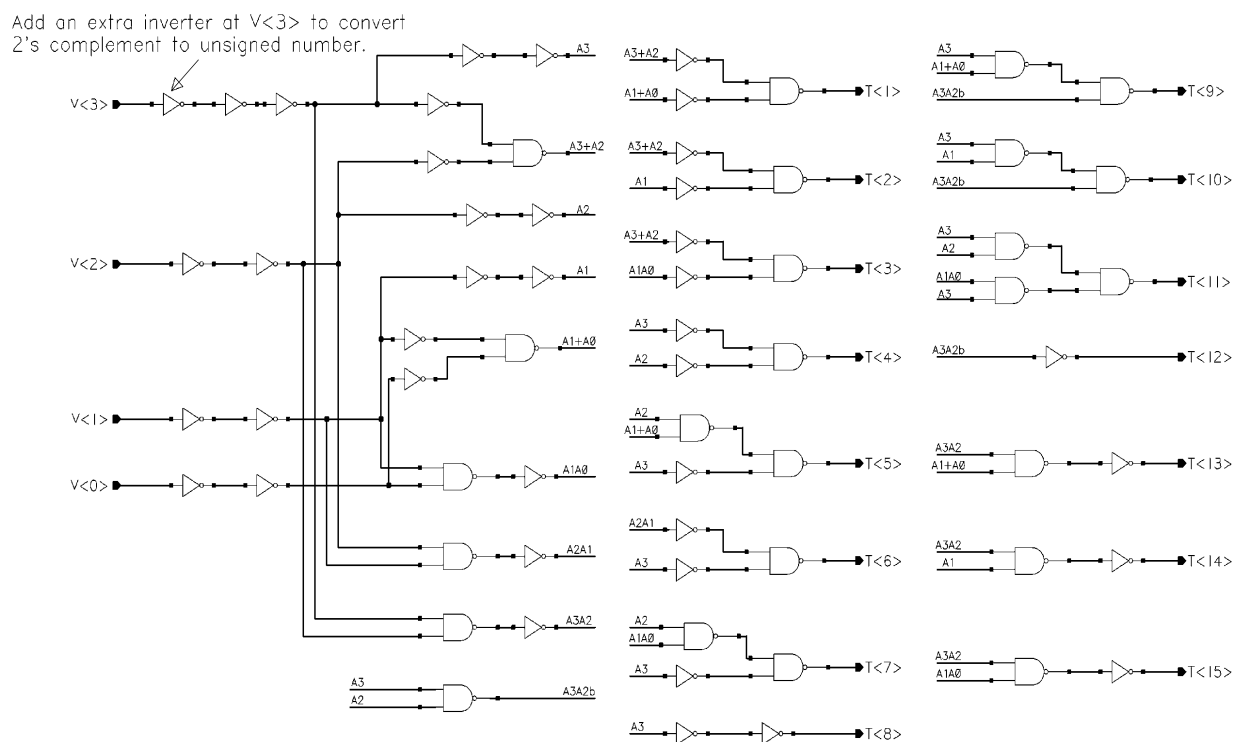


Figure C.4: Binary-to-thermometer schematic

Figure C.5 depicts the schematic of a switch driver and a DFF. The DFF's purpose is to sample/re-time the thermometer codes at 4Gs/s to ensure their proper timing alignment. The latch between output data path (Do) and its complement (\overline{Do}) is used to align their edge intersections to half-swings. Lastly, the additional transmission gate on \overline{Do} path is used for propagation delay matching.

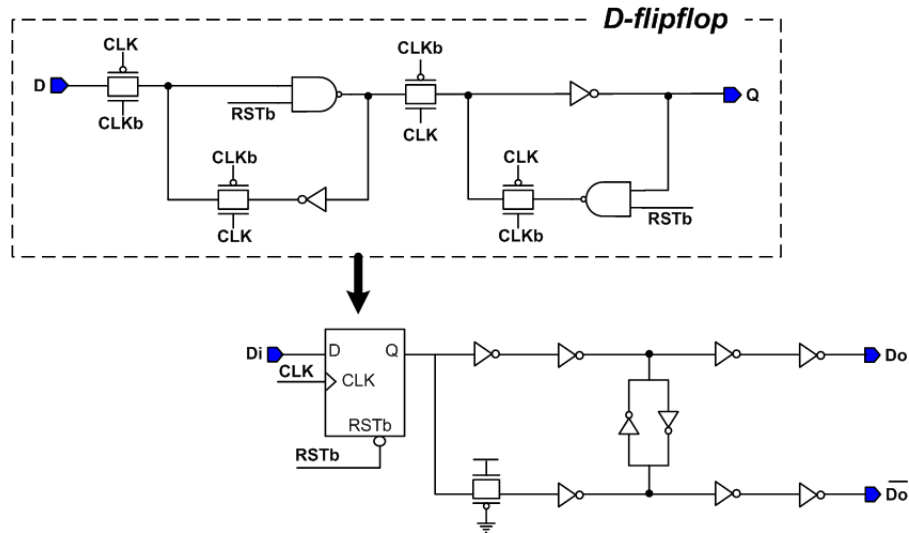


Figure C.5: Switch driver schematic

C.5. Current Calibration Principles

The calibration technique works based on charge storage on the gate-source capacitance (C_{gs}) of CMOS transistors. It uses the same reference current (I_{ref}) to calibrate all current cells. The current value of each cell does not need to be the same as I_{ref} but needs to accurately match the other cells [37]. Figure C.6 shows the calibration principle for one of the current cells [40].

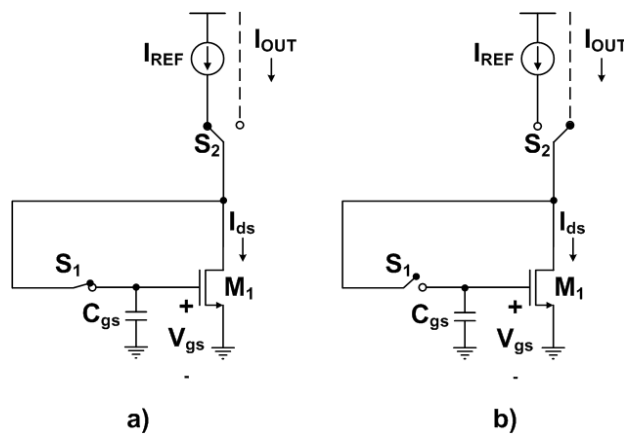


Figure C.6: Calibration principle a) Calibration b) Operation

The switches S_1 and S_2 are in the states depicted in of figures C.6(a) and C.6(b) for the calibration and operation phases, respectively. During calibration, S_1 puts the MOS transistor M_1 into saturation due to its diode connection while S_2 allows I_{ref} to flow into M_1 . This forces the gate-source voltage (V_{gs}) and the charge on the parasitic capacitance C_{gs} of M_1 to whatever value required so that its drain current, I_{ds} , equals I_{ref} . During the operation phase, although S_1 is opened, V_{gs} is theoretically unchanged since the charge on C_{gs} is preserved. This allows S_2 to source approximately the same current, I_{ref} , from the output.

In a practical implementation, S_1 and S_2 are made of MOS transistors. Whenever S_1 switches off, its channel charge is partly dumped on to the gate of M_1 (called “charge-injection”), causing the charge on C_{gs} to decrease by the same amount. This results in a sudden decrease of V_{gs} . In addition, another effect causes V_{gs} to decrease. Although S_1 is off, the reverse-biased diode between its source and substrate is still present, causing V_{gs} to decrease gradually due to leakage current [40].

The reduction in V_{gs} , due to charge-injection (Δq) and leakage current (I_{leak}), causes I_{ds} to decrease as a function of time according to the following calculations [40]:

$$I_{ds}(t) = I_{ref} - g_m \frac{\Delta q}{C_{gs}} - g_m \frac{I_{leak} t}{C_{gs}} \quad (\text{C.1})$$

where $C_{gs} = \frac{2}{3} W L C_{ox}$ and $g_m = \sqrt{2\mu C_{ox} \frac{W}{L} I_{ds}}$.

Thus, equation C.1 can be rewritten as:

$$I_{ds}(t) = I_{ref} - \frac{g_m}{C_{gs}} (\Delta q + I_{leak} t) = I_{ref} - \frac{3}{2L} \sqrt{\frac{2\mu}{C_{ox}} \cdot \frac{I_{ds}}{WL}} (\Delta q + I_{leak} t) \quad (\text{C.2})$$

Equation C.2 indicates that after a certain time T_c , the cell needs to be re-calibrated to maintain its output current with a specified accuracy.

References

- [1] Richard Schreier and Gabor C. Temes, *Understanding Delta-Sigma Data Converters*. Hoboken, New Jersey, USA: John Wiley & Sons, Inc, 2005.
- [2] R. Khoini-Poorfard, *Analysis methods and time-interleaved architectures for oversampling modulators*. PhD thesis, University of Toronto, Edward S. Rogers Sr. Department of Electrical and Computer Engineering, 1995.
- [3] Danijela Cabric, Mike S.W. Chen, David A. Sobel, Jing Yang and Robert W. Brodersen, “Future wireless systems: UWB, 60GHz, and Cognitive radios,” in *IEEE Custom Integrated Circuits Conference, CICC*, pp. 793–796, September 2005.
- [4] M. de Courville, S. Zeisberg, M. Muck, and J. Schonhner, “BroadWay - the Way to Broadband access at 60GHz,” in *International Conference on Telecommunication*, June 2002.
- [5] D. Dardari and V. Tralli, “High-speed indoor wireless communications at 60 GHz with coded OFDM,” *IEEE Transactions on Communications*, vol. 47, no. 11, pp. 1709–1721, November 1999.
- [6] B. Razavi, T. Aytur, C. Lam, F. Yang, K. Li, R. Yan, and H. Kang, “A UWB CMOS transceiver,” *IEEE Journal of Solid-State Circuits*, vol. 40, no. 12, pp. 2555–2562, December 2005.
- [7] J. Balakrishnan, A. Batra, and A. Dabak, “A multi-band OFDM system for UWB communication,” in *IEEE Conference on Ultra Wideband Systems and Technologies*, pp. 354–358, 2003.

- [8] P. Smulders, “60 GHz radio: prospects and future directions,” in *Proceedings of IEEE 10th Symposium on Communications and Vehicular Technology*, pp. 1–8, November 2003.
- [9] T. C. Chen, “Where CMOS is going: trendy hype vs. real technology,” in *IEEE International Solid-State Circuits Conference, ISSCC*, pp. 1–18, February 2006.
- [10] Fu-Liang Yang, Jiunn-Ren Hwang, and Yiming Li, “Electrical characteristic fluctuations in sub-45nm CMOS devices,” in *IEEE Custom Intergrated Circuits Conference, CICC*, vol. 1, pp. 691–694, 2006.
- [11] Jing Cao, Haiqing Lin, Yihai Xiang, Chungpao Kao, and Ken Dyer, “A 10-bit 1GSample/s DAC in 90nm CMOS for embedded applications,” in *IEEE Custom Intergrated Circuits Conference, CICC*, vol. 1, pp. 165–168, 2006.
- [12] K. Doris, J. Briaire, D. Leenaerts, M. Vertregt, and A. van Roermund, “A 12b 500MS/s DAC with >70dB SFDR up to 120MHz in 0.18 μ m CMOS,” in *IEEE International Solid-State Circuits Conference, ISSCC*, pp. 116–117, 588, February 2005.
- [13] Anne Van den Bosch, Marc A. F. Borremans, Michel S. J. Steyaert, and Willy Sansen, “A 10-bit 1-GSample/s Nyquist current-steering CMOS D/A converter,” *IEEE Journal of Solid-State Circuits*, vol. 36, no. 3, pp. 315–324, March 2001.
- [14] Chi-Hung Lin and Klass Bult, “A 10-b, 500-Msample/s CMOS DAC in 0.6 mm^2 ,” *IEEE Journal of Solid-State Circuits*, vol. 33, no. 12, pp. 1948–1958, December 1998.
- [15] David B. Barkin, Andrew C.Y. Lin, David K. Su, and Bruce A. Wooley, “A CMOS oversampling bandpass cascaded D/A Converter with digital FIR and current-mode semi-digital filtering,” *IEEE Journal of Solid-State Circuits*, vol. 39, no. 4, pp. 585–593, April 2004.
- [16] Todd S. Kaplan, Joseph F. Jensen, Charles H. Fields, and M. Frank Chang, “A 2-Gs/s 3-bit $\Delta\Sigma$ -Modulated DAC with tunable bandpass mismatch shaping,” *IEEE Journal of Solid-State Circuits*, vol. 40, no. 3, pp. 603–610, March 2005.

- [17] Susan Luschas, Richard Schreier, and Hae-Seung Lee, "Radio frequency digital-to-analog converter," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 9, pp. 1462–1467, September 2004.
- [18] Fred Harris, and Pranesh Sinha, "On synthesizing high speed sigma-delta DACs by combining the outputs of multiple low speed sigma-delta DACs," in *IEEE Conference on Signals, Systems and Computers*, vol. 2, pp. 1050–1054, November 2002.
- [19] Richard Schreier, "Quadrature mismatch-shaping," in *IEEE International Symposium on Circuits and Systems, ISCAS*, pp. 675–678, May 2002.
- [20] R. Cormier Jr, T. Sculley, and R. Bamberger, "Combining sub-band decomposition and sigma-delta modulation for wide-band A/D conversion," in *IEEE International Symposium on Circuits and Systems, ISCAS*, pp. 357–360, June 1994.
- [21] Mucahit Kozak, and Izzet Kale, "Novel topologies for time-interleaved Delta-Sigma modulators," *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 47, no. 7, pp. 639–654, July 2000.
- [22] Ian Galton, and Henrik T. Jensen, "Oversampling parallel Delta-Sigma modulator A/D conversion," *IEEE Transactions on Circuits and Systems - II: Analog and Digital Signal Processing*, vol. 43, no. 12, pp. 801–810, December 1996.
- [23] Ramin Khoini-Poorfard, Lysander B. Lim, and David A. Johns, "Time-interleaved oversampling A/D converters: Theory and Practice," *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 44, no. 8, pp. 634–645, August 1997.
- [24] Katayoun Falakshahi, Chih-Kong Ken Yang, and Bruce A. Wooley, "A 14-bit, 10-Msamples/s D/A converter using multibit $\Delta\Sigma$ modulation," *IEEE Journal of Solid-State Circuits*, vol. 34, no. 5, pp. 607–615, May 1999.
- [25] Martin Clara, Wolfgang Klatzer, Andreas Wiesbauer, and Dietmar Straeussnigg, "A 350MHz low-OSR $\Delta\Sigma$ current-steering DAC with active termination in 0.13 μm

- CMOS,” in *IEEE International Solid-State Circuits Conference, ISSCC*, pp. 118–119, 588, 2005.
- [26] Ramin Khoini-Poorfard, and David A. Johns, “Mismatch effects in time-interleaved oversampling converters,” in *IEEE International Symposium on Circuits and Systems, ISCAS*, pp. 429–432, 1994.
- [27] Yunyoung Choi, and Franco Maloberti, “Design of oversampling current steering DAC with 640Mhz equivalent clock frequency,” in *IEEE International Symposium on Circuits and Systems, ISCAS*, vol. 1, pp. 109–112, May 2002.
- [28] Tao Shui, R. Schreier, and F. Hudson, “Mismatch shaping for a current-mode multibit Delta-Sigma DAC,” *IEEE Journal of Solid-State Circuits*, vol. 34, no. 3, pp. 331–338, March 1999.
- [29] I. Fujimori, A. Nogi, and T. Sugimoto, “A multibit Delta-Sigma audio DAC with 120-dB dynamic range,” *IEEE Journal of Solid-State Circuits*, vol. 35, no. 8, pp. 1066–1073, August 2000.
- [30] M. Annovazzi, V. Colonna, G. Gandolfi, F. Stefani, and A. Baschiroto, “A low-power 98-dB multibit audio DAC in a standard 3.3-V 0.35- μm CMOS technology,” *IEEE Journal of Solid-State Circuits*, vol. 37, no. 7, pp. 825–834, July 2002.
- [31] T. Hamasaki, Y. Shinohara, H. Terasawa, K. Ochiai, M. Hiraoka, and H. Kanayama, “A 3-V, 22-mW multibit current-mode $\Sigma\Delta$ DAC with 100dB dynamic range,” *IEEE Journal of Solid-State Circuits*, vol. 31, no. 12, pp. 1888–1894, December 1996.
- [32] P. Naus, E. Dijkmans, E. Stikvoort, A. McKnight, D. Holland, and W. Brandinal, “A CMOS stereo 16-bit D/A converter for digital audio,” *IEEE Journal of Solid-State Circuits*, vol. 22, no. 3, pp. 390–395, June 1987.
- [33] Peter Kiss, Jesus Arias, Dandan Li, and Vito Boccuzzi, “Stable high-order Delta-Sigma digital-to-analog converters,” *IEEE Transactions on Circuits and Systems-I: Regular Papers*, vol. 51, no. 1, pp. 200–205, January 2004.

- [34] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Eaglewood Cliffs, New Jersey, USA: P T R Prentice Hall, Inc , 1993.
- [35] Mucahit Kozak, Mustafa Karaman, and Izzet Kale, “Efficient architectures for time-interleaved oversampling Delta-Sigma converters,” *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 47, no. 8, pp. 802–810, August 2000.
- [36] D. P. Scholnik, “A parallel digital architecture for delta-sigma modulation,” in *IEEE International Midwest Symposium on Circuits and Systems*, vol. 1, pp. 352–355, August 2002.
- [37] David A. Johns and Ken Martin, *Analog Integrated Circuit Design*. Toronto, Canada: John Wiley & Sons, Inc, 1997.
- [38] R. Schreier, “Delta-Sigma Toolbox Version 6.0,” 2003.
- [39] Jared Welz, and Ian Galton, “Necessary and sufficient conditions for mismatch shaping in a general class of multibit DACs,” *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 49, no. 12, pp. 748–759, December 2002.
- [40] D. Wouter J. Groeneveld, Hans J. Schouwenaars, Henk A. H. Termeer, and Cornelis A. A. Bastiaansen, “A self-calibration technique for monolithic high-resolution D/A converters,” *IEEE Journal of Solid-State Circuits*, vol. 24, no. 6, pp. 1517–1522, December 1989.
- [41] D.A. Parker and K.K. Parhi, “Area-efficient parallel FIR digital filter implementations,” in *IEEE International Conference on Application-Specific Systems, Architectures and Processors*, pp. 93–111, August 1996.
- [42] Bede Liu, “Effect of finite word length on the accuracy of digital filters - A Review,” *IEEE Transactions on Circuits Theory*, vol. 18, no. 6, pp. 670–677, November 1971.
- [43] Henry Samuelli, “An improved search algorithm for the design of multiplierless FIR filters with powers-of-two coefficients,” *IEEE Transactions on Circuits and Systems*, vol. 36, no. 7, pp. 1044–1047, July 1989.

- [44] Kyung-Ju Cho, Kwang-Chul Lee, Jin-Guyn Chung, and Keshab K. Parhi, “Design of low-error fixed-width modified Booth multiplier,” *IEEE Transactions on Very Large Scale Integration Systems*, vol. 12, no. 5, pp. 522–531, May 2004.
- [45] Abdellatif Bellaouar, and Mohamed I. Elmasry, *Low-Power Digital VLSI Design: Circuits and Systems*. Norwell, Massachusetts, USA: Kluwer Academic Publishers, 2000.
- [46] Behzad Razavi, *Design of Analog CMOS Integrated Circuits*. New York, NY, USA: McGraw-Hill Companies, Inc, 2001.
- [47] Tae-young Oh, Christoph Jungemann, and Robert W. Dutton, “Hydrodynamic simulation of RF noise in deep-submicron MOSFETs,” in *International Conference on Simulation of Semiconductor Processes and Devices, SISPAD*, pp. 87–90, September 2003.
- [48] K.Roy, S.Mukhopadhyay, and H.Mahmoodi-Meimand, “Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits,” in *Proceedings of the IEEE*, vol. 91, pp. 305–327, February 2003.
- [49] Timothy O. Dickson, Rudy Beerkens, and Sorin P. Voinigescu, “A 2.5-V, 40-Gb/s decision circuit using SiGe BiCMOS logic,” in *Proceedings of the IEEE*, pp. 206 – 209, June 2004.