The Vault

https://prism.ucalgary.ca

Open Theses and Dissertations

2013-06-03

A Time-Based 5GS/s CMOS Analog-to-Digital Converter

Macpherson, Andrew Robert

Macpherson, A. R. (2013). A Time-Based 5GS/s CMOS Analog-to-Digital Converter (Doctoral thesis, University of Calgary, Calgary, Canada). Retrieved from https://prism.ucalgary.ca. doi:10.11575/PRISM/25056 http://hdl.handle.net/11023/750

Downloaded from PRISM Repository, University of Calgary

UNIVERSITY OF CALGARY

A Time-Based 5GS/s CMOS Analog-to-Digital Converter

by

Andrew Robert Macpherson

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

CALGARY, ALBERTA MAY, 2013

© Andrew Robert Macpherson 2013

Abstract

In deep-submicron CMOS technology, the switching speed of digital circuits has become extremely fast. At the same time, analog design has become increasingly difficult due to very low supply voltage levels. This makes it advantageous to represent signals in the timedomain as the delay between two digital pulse edges, rather than the conventional voltage domain. This work applies the idea of time-based processing to high-speed analog-to-digital converters (ADCs).

The proposed time-based ADC consists of two stages. The first is the voltage-to-time converter (VTC), which uses a modified current-starved inverter architecture. The VTC accepts an analog voltage input and produces a series of pulses in which the delay of each pulse is proportional to the input at the time the pulse was created. The second stage is the time-to-digital converter (TDC). The TDC measures the delay on each VTC pulse and converts it to a digital logic value. The VTC and TDC can be physically separated with the VTC output transmitted to the TDC input over coaxial cables. An on-chip digital programming system in the TDC allows the entire ADC to be calibrated, and an automatic calibration scheme is presented.

Two prototypes were fabricated, a 3-bit 2.5GS/s ADC in 90nm CMOS and a 4-bit 5GS/s ADC in 65nm CMOS. The 65nm circuit achieves an effective resolution bandwidth of 2.1GHz and consumes 34.6mW of power. The figure of merit is 1.0pJ/conversion. This is the fastest time-based ADC published to date.

Acknowledgements

To begin with, I wish to thank my supervisor Dr. Jim Haslett. His advice and guidance has been instrumental in successfully performing my research and completing my thesis. I would also like to express my gratitude to Dr. Leo Belostotski for all of his helpful advice. Dr. Abdel Yousif's ideas shaped the direction of my research and I am grateful for his guidance during his time with the RFIC research group. I am also thankful to Dr. Yongsheng Xu for using his knowledge and experience to help me whenever I needed it.

I would like to thank my fellow graduate students in the Department of Electrical and Computer Engineering for their camaraderie and support over the years. In particular I thank my collaborator Dr. Ken Townsend, from whom I learned a great deal about integrated circuit design during our work together.

I gratefully acknowledge the financial support provided by the Natural Sciences and Engineering Research Council of Canada and Alberta Innovates Technology Futures, as well as the subsidized fabrication services provided by the Canadian Microsystems Corporation.

In the preparation of this thesis I took advantage of several excellent free software tools, including MiKTeX, Texmaker, JabRef, Inkscape and Xcircuit. I wish to thank the creators of these programs for making them available.

A special thank you goes out to my family for being a source of warmth and stability not only during my graduate studies but throughout my life. Finally, I want to thank Christie for providing much-needed support and encouragement during the writing process.

Table of Contents

owledgements of Contents	iii iv	
of Contents	iv	
f T-hlan		
List of Tables vi		
of Figures v	iii	
of Abbreviations	xi	
troductionTime-Based ProcessingConventional ADC Architectures for Sampling Rates Above 1GS/s1.2.1Flash1.2.2Pipeline1.2.3Successive Approximation Register1.2.4Time-InterleavingTime-Based ADCs1.3.1Integrating ADCs1.3.2Voltage-Controlled Oscillator Based ADCs1.3.3Voltage-to-Time Converter Based ADCsTime-to-Digital ConvertersMotivation: The Square Kilometre ArrayThesis Organization	$ \begin{array}{r} 1 \\ 2 \\ 3 \\ 3 \\ 6 \\ 8 \\ 9 \\ 11 \\ 11 \\ 12 \\ 12 \\ 14 \\ 17 \\ 17 \\ 17 \\ 17 \\ 17 \\ 17 \\ 17 \\ 17 \\ 17 \\ 17 \\ 17 \\ 17 \\ 17 \\ 17 \\ 17 \\ 12 \\ 12 \\ 14 \\ 17 \\ 11 \\ 12 \\ 12 \\ 12 \\ 12 \\ 12 \\ 12 \\ 12 \\ 14 \\ 17 \\ 17 \\ 17 \\ 17 \\ 17 \\ 17 \\ 17 \\ 17 \\ 17 \\ 17 \\ 17 \\ 17 \\ 17 \\ 17 \\ 17 \\ 17 \\ 17 \\ 11 \\ 11 \\ 11 \\ 12 \\ 11 \\ 12 \\ 12 \\ 12 \\ 12 \\ 17 \\ 17 \\ 17 \\ 11 \\ 1$	
nearity AnalysisImage: Derivation of ENOB2.1.1ENOB of Non-Quantized Systems2.1.1ENOB of Non-Quantized SystemsMeasuring ENOB2.2.1The Histogram Method2.2.2The FFT Method2.2.3Comparison Between Histogram and FFT Methods2.3.1Analytic Linearity Calculation2.3.2Using Series ExpansionsDifferential Inputs2.4.1Odd and Even Functions	 19 22 23 24 24 27 27 30 31 32 	
c nti 1 2 	of Abbreviations ntroduction .1 Time-Based Processing . .2 Conventional ADC Architectures for Sampling Rates Above 1GS/s .1.2.1 Flash . .1.2.2 Pipeline .1.2.3 Successive Approximation Register .1.2.4 Time-Interleaving . .3 Time-Based ADCs . .1.3.1 Integrating ADCs . .1.3.2 Voltage-Controlled Oscillator Based ADCs . .1.3.3 Voltage-to-Time Converter Based ADCs . .1.3.4 Time-to-Digital Converters . .5 Motivation: The Square Kilometre Array . .6 Thesis Organization . .7.1 ENOB of Non-Quantized Systems . .2.1 The Histogram Method . .2.2.2 The FFT Method . .2.3.3 Comparison Between Histogram and FFT Methods . .3.4 Nalytic Calculation . .3.5 Linearity Calculation . .3.6 Differential Inputs . .3.7 Line Stress Stresses . .3.8 Linearity Calculation . .3.9 Linearity Calculation . .3.1 Analytic Calculation . .3.2 Linearity Calculation . .3.3 Linearity Calculation . .3.4 Differential Inputs . .3.2 Using Series Expansions . .4.1 Odd and Even Functions . .4.2 Di	

3	\mathbf{VT}	C Analysis - Theory	36
	3.1	Explanation of Operation	36
		3.1.1 Detailed sequence of operation	37
		3.1.2 Derivation of VTC delay	39
		3.1.3 Extraction of Level 1 model parameters from BSIM4 simulations	40
		3.1.4 Comparison with BSIM4 models	42
	3.2	Single-ended VTC Optimization	44
		3.2.1 Maximizing Single-Ended Linearity	44
		3.2.2 Range and Absolute Delay - Single-Ended VTC	47
		3.2.3 Design Procedure - Single-ended VTC	48
		3.2.4 Design Example - Single-Ended VTC	50
	3.3	Differential VTC Optimization	52
		3.3.1 Maximizing Differential Linearity	52
		3.3.2 Bange and Absolute Delay - Differential VTC	54
		3.3.3 Design Procedure - Differential VTC	56
		3.3.4 Design Example - Differential VTC	57
	3.4	Jitter Analysis	58
	0.1	3 4 1 CMOS Inverter	59
		3.4.2 Current-Starved Inverter	61
		343 VTC	64
			01
4	Firs	st Generation VTC and TDC in 90nm CMOS	66
	4.1	Time-Interleaved VTC	67
		4.1.1 Measured Results	70
	4.2	3-bit Parallel-Branch TDC	74
	4.3	2.5GS/s Time-Based ADC	76
5	\mathbf{Pse}	eudo-Differential VTC in 65nm CMOS	79
	5.1	VTC Half-Cell	79
		5.1.1 Dutv-Cycle Adjustment Circuit	79
		5.1.2 VTC Core	80
	5.2	Simulated Results	81
		5.2.1 Output Driver	85
	5.3	VTC Calibration	86
		5.3.1 System Description	86
		5.3.2 Test Results	89
	5.4	Measured Results	91
6	4-bi	It Vernier Delay Line TDC in 65nm CMOS	100
	0.1		100
	0.2	De de deixe Outente	102
	0.3	Re-clocking Outputs	103
	0.4		105
		0.4.1 Delay Tuning	106
		0.4.2 Simulated Results	108

		6.4.3 Complete Differential Delay Blocks	110
	TDC Programming	111	
		6.5.1 Serial-to-Parallel (S2P) and DAC Blocks	112
		6.5.2 Tuning Resolution	114
	6.6	Output Decoding	115
		6.6.1 On-Chip Implementation	118
	6.7	Layout and Simulated Power Consumption	119
	6.8	TDC Calibration Algorithm	121
		6.8.1 Possible Inputs and Expected Histograms	123
		6.8.2 Algorithm Details	125
	6.9	Measured Results	127
_	0 F		100
7	65n	m ADC Measurements	132
	7.1	Automatic Calibration	133
	7.2	DC Input Characteristics	134
	7.3	Wideband Input Characteristics	136
		7.3.1 Analysis of 5GS/s Wideband Performance	138
	7.4	Figures of Merit	141
8	Con	clusions and Future Work	144
	8.1	Future Work	145
		8.1.1 Self-Contained, PVT-Independent Circuits	145
		8.1.2 Integration of VTC with SKA Receiver Chain	146
Bi	bliog	rraphy	147

Bibliography

List of Tables

3.1	Single VTC Design example (a) Extracted/estimated values from simulator,	
	(b) specifications, and (c) resulting design.	50
3.2	Single VTC Design results using (a) Level 1 models, (b) BSIM4 models, and	
	(c) BSIM4 models after tweaking voltages for correct range at peak ENOB .	51
3.3	Differential VTC Design example (a) Extracted/estimated values from simu-	
	lator, (b) specifications, and (c) resulting design	57
3.4	Differential VTC Design results using (a) Level 1 models, (b) BSIM4 models,	
	and (c) BSIM4 models after tweaking voltages for correct range at peak ENOB	57
3.5	Model parameters used for all calculations in this section	61
3.6	Simulated and calculated values for inverter jitter and delay	61
41	Summary of 90nm ADC measured results	78
T . I		10
5.1	Measured VTC bandwidth at several clock frequencies using two different	
	bandwidth definitions	95
5.2	Measured power levels for for 5GS/s VTC output components $\ldots \ldots \ldots$	98
5.3	Summary of 65nm VTC Measured Performance	99
6.1	Minimum Setup and Hold Times for TDC Re-clocking Circuit	105
6.2	Summary of 65nm TDC Measured Performance	131
7.1	Measured ADC performance at different sampling frequencies	138
7.2	Summary of 65nm ADC Measured Performance	143
	v	

List of Figures and Illustrations

1.1	A 3-bit Flash ADC	4
1.2	Conceptual Diagram of an N-bit folded flash ADC with a folding factor of 2	5
1.3	Pipeline ADC	6
1.4	Single pipeline ADC stage	$\overline{7}$
1.5	A basic successive approximation ADC	9
1.6	A time-interleaved ADC	10
1.7	An integrating ADC	12
1.8	A time-based ADC consisting of a VTC and TDC	13
1.9	An N-bit delay line TDC with resolution t_{δ}	15
1.10	An N-bit Vernier delay line TDC with resolution t_{δ}	15
1.11	Artist's rendering of dish antennas for the Square Kilometre Array	17
2.1	ENOB derivation: (a) Quantized staircase and ideal output (b) Quantization	
	error	20
2.2	Error in calculated SINAD formula for quantization noise	22
2.3	Simulated effect of noise on FFT and histogram ENOB calculation methods	25
2.4	Effect of input amplitude on FFT and histogram ENOB measurement methods	26
2.5	ENOB of example functions for $-1 \le x \le 1$	29
2.6	Comparison of single-ended and differential ENOB for $y = \frac{A}{x-k}$	35
3.1	VTC schematic with parasitic capacitor	36
3.2	Circuit schematic for extracting Level 1 parameters	40
3.3	Level 1 parameters extracted from BSIM4 simulations for different drain currents	41
3.4	Simulated VTC drain current for different input levels	42
3.5	Simulated VTC delays using Level 1 and BSIM4 models	43
3.6	Simulated VTC waveforms using Level 1 and BSIM4 models	43
3.7	Comparison of single-ended ENOB calculated analytically, by numerical sim-	
	ulation of analytic delay model, and CAD simulation with Level 1 transistor	
	models	45
3.8	VTC ENOB curves for single-ended VTC design example	51
3.9	Optimum ENOB achievable with differential and single-ended VTCs	54
3.10	Comparison of differential ENOB calculated analytically, by numerical simu-	
	lation of analytic delay model, and CAD simulation with Level 1 transistor	
0 1 1	MTC ENOD comments for differential VTC design community	00 E0
び.11 2 1 0	VIC ENOB curves for differential VIC design example	58
5.12	VTC	59
3.13	Small-signal noise models for jitter calculation; a)Starved-inverter and b)VTC	
	(simplified)	61
3.14	Comparison of derived current-starved inverter jitter and delay models with	
	BSIM4 simulations	63
3.15	Comparison of derived VTC jitter model with BSIM4 simulation	65

4.1	90nm VTC and TDC chip photograph	66		
4.2	90nm VTC top-level block diagram			
4.3	90nm VTC channel 1 details. Channel 2 is identical but with its own clock,			
	reset and output signals.	68		
4.4	90nm VTC track-and-hold circuit with charge injection cancellation	68		
4.5	90nm VTC core schematic	69		
4.6	Illustration of starved inverter output			
4.7	Measured 90nm VTC delay with DC input. Output delay is relative to the			
	output clock.	71		
4.8	Measured 90nm VTC delay curves with different V_{cap} tuning bias levels. Input			
	voltage is in addition to the constant DC bias	72		
4.9	Measured 90nm VTC waveforms captured directly from oscilloscope	72		
4.10	0 Measured 90nm 5GS/s VTC wideband linearity using SDR-based ENOB			
4.11	Measured 90nm 5GS/s VTC wideband output range with constant input am-			
	plitude	74		
4.12	90nm 3-bit parallel-branch TDC block diagram	75		
4.13	Measured 90nm ADC DNL and INL for DC input	76		
4.14	Measured 90nm ADC wideband linearity based on histogram testing	77		
L 1	Full 65mm VTC half call achematic	00		
0.1 ธ.ว	Simulated VTC subjut delay upper input valtage for typical (TT) clay (SC)	00		
0.2	simulated VIC output delay versus input voltage for typical (III), slow (SS)	01		
59	Monte Carle analyzis for VTC delay range variation	01		
0.0 5 4	Simulated VTC ENOD upravise input frequency at 5CC (a using a) new output	02		
0.4	deta, and b)an ideal 4 bit TDC applied to the data	09		
55	Simulated VTC output dolou range versus input frequency at 5CS/a	00 04		
5.5 5.6	Output driver schematic with pad and bondwire model	04 85		
5.0	Simulated rise and fall times for output driver with 1.2V supply and 5CHz	00		
0.1	clock input	86		
58	VTC Calibration System Block Diagram	87		
5.0	Measured DAC output for correct up and down counting with a 100S/s cali-	01		
0.5	bration clock	90		
510	Measured DAC output showing incorrect counting performance	90		
5 11	DAC output and up/dn signal during VTC Calibration The calibration op-	00		
0.11	erates correctly for the first 60 clock cycles after which errors occur	91		
5.12	VTC circuit board with filter networks	92		
5.13	VTC differential output delay with DC inputs at 5GS/s	93		
5.14	Measured VTC ENOB peaks at 5GS/s	94		
5.15	VTC power consumption versus clock frequency with 1 0V and 1 2V supply	01		
0.10	voltage	95		
5.16	Measured VTC wideband linearity and gain at three different sampling fre-	50		
0.10	quencies	96		
5.17	Measured frequency spectrum for 5GS/s VTC output with 500MHz input			
-	signal	97		
5.18	Measured Differential VTC Jitter	98		

6.1	Number of delays required for flash and vernier delay line TDCs	101	
6.2	3-bit TDC Thresholds	101	
6.3	4-bit 65nm TDC Core Schematic		
6.4	Simulated TDC Waveforms		
6.5	Eye Diagram of TDC Re-sampling Clock and Data	105	
6.6	Single Delay Block Schematic	106	
6.7	Simulated delay block (a) absolute delay and (b) differential delay	109	
6.8	Simulated delay block pulse widths	109	
6.9	Differential delay blocks for generating (a) t_{δ} (b) $-7t_{\delta}$	110	
6.10	Monte Carlo analysis for (a) t_{δ} (b) -7 t_{δ} delay blocks	111	
6.11	TDC Programming System	112	
6.12	TDC DACs (a) PMOS-type, (b) NMOS-type, and (c) bias block	113	
6.13	Simulated t_{δ} delay block with DAC control: (a) full sweep and (b) step resolution	115	
6.14	Simulated $7t_{\delta}$ delay block with DAC control: (a) full sweep and (b) step		
	resolution	116	
6.15	Average output values for fat-tree and minimal-logic thermometer decoders		
	with single bit errors	118	
6.16	Schematics for (a) AND gate and (b) OR gate (not used on chip)	119	
6.17	Final Minimum-Logic Decoder Design	120	
6.18	TDC Layout	121	
6.19	Breakdown of TDC power consumption (simulated data)	122	
6.20	TDC Automatic Calibration Algorithm	124	
6.21	Simulated TDC Calibration Performance	127	
6.22	TDC circuit board with filter networks	128	
6.23	TDC measurements: (a) Delay range for a single delay block, and (b) transfer		
	curves with a single delay set to various values	129	
6.24	Measured TDC ENOB using histograms with simple 2-pt tuning method	129	
6.25	Measured TDC transfer curve using histogram data	130	
6.26	TDC Power Consumption versus clock frequency	131	
71	65nm VTC and TDC chip photo	132	
72	Two-board test setup for ADC measurements	133	
7.3	Measured ADC ENOB during automatic calibration process	134	
7.4	Measured ADC output for DC differential input)	135	
7.5	Measured ADC DNL and INL for DC input	136	
7.6	Measured ADC wideband linearity for 1GS/s. 2.5GS/s. 5GS/s and 6GS/s		
	sampling frequencies	137	
7.7	Measured performance of 5GS/s ADC with amplitude optimized at each fre-		
	quency, as opposed to using a constant amplitude for all frequencies	139	
7.8	Measured ADC output frequency spectrum with 500MHz and 900MHz inputs	140	
7.9	Measured ADC output harmonics for 500MHz and 900MHz input signals	140	
7.10	Figure of merit for the 65nm ADC operated at different sampling frequencies		
	using both the ISSCC and ITRS FOM definitions	142	
7.11	Figure of merit for published ADCs with time-based ADCs highlighted	143	

List of Abbreviations

ADC	Analog-to-digital converter
BSIM4	Berkeley Short-channel IGFET Model version 4
CMOS	Complementary metal oxide semiconductor
DAC	Digital-to-analog converter
DLL	Delay-locked loop
DNL	Differential non-linearity
ENOB	Effective number of bits
ERBW	Effective resolution bandwidth
FFT	Fast Fourier transform
FOM	Figure of merit
GPIB	General purpose interface bus
IGFET	Insulated-gate field-effect transistor
INL	Integral non-linearity
ISSCC	International Solid-State Circuits Conference
LSB	Least significant bit
MDAC	Multiplying digital-to-analog converter
MSB	Most significant bit
PPM	Pulse-position modulation
PSD	Power spectral density
PVT	Process, voltage and temperature
PWM	Pulse-width modulation
RMS	Root mean square
S2P	Serial to parallel
SAR	Successive approximation register
SDR	Signal to distortion ratio
SFDR	Spurious-Free Dynamic Range
SINAD	Signal to noise and distortion ratio
SNR	Signal to noise ratio
TDC	Time-to-digital converter
THD	Total harmonic distortion
USB	Universal serial bus
VCO	Voltage-controlled oscillator
VDL	Vernier delay line
VTC	Voltage-to-time converter
ZCBC	Zero crossing based comparators

Chapter 1

Introduction

We live in an analog world. With the introduction of powerful digital processors in the mid-20th century, the need arose to bridge the gap between continuous analog information and discrete digital information so that digital systems could process inputs from the real world. This is the function of analog-to-digital converters (ADCs).

As the information revolution has progressed, there has been a continuous demand for faster ADCs to provide more detailed representations of analog signals. In 1987, the first ADC was created with the ability to sample a signal at 1GS/s, or 10⁹ times per second [1]. From this time through the 1990's, gigasample-rate ADCs were designed mainly for use in oscilloscopes. They used bipolar transistor technology and typically used at least 3 watts of power [2–6]. While this was acceptable for oscilloscopes, new applications began to emerge which required fast ADCs with lower power consumption. The reasons for saving power include battery life in mobile applications as well as cost-savings and heat management in large-scale operations.

Near the turn of the 21st century, the first CMOS (complementary metal-oxide semicondutor) gigasample-rate ADCs began to appear [7–9]. The performance of CMOS has always lagged behind bipolar technology, which is why CMOS gigasample-rate ADCs did not appear for more than a decade after their bipolar counterparts. Once CMOS technology was capable of reaching the desired sampling rates, it soon began to dominate the ADC industry due to its improved power efficiency and ease of integration with digital circuitry. Since then, the race has been on to produce ADCs that consume less and less power without sacrificing speed or accuracy. A wide variety of architectures have been invented to further this goal. One such class of architectures is time-based processing.

1.1 Time-Based Processing

In electronic circuits and systems, information can be represented in various ways. For analog signals, the most common way is using voltage, in which the level of a voltage signal directly maps to the analog information being represented. Another common representation is current. Signals can easily be converted between voltage and current, using a transistor for example, or even passively using resistors. Voltages and currents can represent information that is not only continuous-valued (i.e. analog), but also continuous-time. In contrast, information can also be represented digitally, as a series of bits (which themselves can be represented by voltage or current). Digital representations must be discrete-time as well as having discrete values. It is also possible for a signal to be continuous-valued but discrete time. For example, the output of a sample-and-hold circuit changes only once per clock period, but can take any value within the range of the circuit. Another discrete-time but continuous-valued representation is time-based processing, in which the time delay of a pulse can be mapped to analog information [10].

In time-based processing, information is represented in terms of time delays, for example the time between the rising edges of two pulse waveforms. This form of processing has been gaining popularity due to a fundamental observation about deep-submicron CMOS technology (i.e. with feature sizes below 250nm). This observation has been expressed succinctly by Staszewski [11]:

> "In a deep-submicron CMOS process, time-domain resolution of a digital signal edge transition is superior to voltage resolution of analog signals."

There are two reasons for this. First, as CMOS gate-lengths shrink the switching speed of digital circuits (e.g. inverters) increases proportionally due to the decreased gate capacitance of the devices. Second, as gate lengths have been scaled down the supply voltage has shrunk proportionally in order to prevent dielectric breakdown. This results in less headroom for

voltage signals, meaning the signal amplitude tends to be lower and the signal-to-noise ratio (SNR) decreases.

1.2 Conventional ADC Architectures for Sampling Rates Above 1GS/s

A number of ADC architectures have been reported to achieve sampling rates greater than 1GS/s. Some of these, notably sigma-delta ADCs, use the principle of oversampling and are only suitable for input signals with frequencies far below the sampling rate [12]. This section will describe those architectures that can be considered Nyquist ADCs - that is, they have an input bandwidth that is close to the Nyquist rate (one-half of the sampling rate).

1.2.1 Flash

The earliest gigasample-rate ADCs used the flash architecture [13, 14]. Flash ADCs make use of parallel processing and a simple design to push the speed limits for a given fabrication technology. In CMOS, single-channel flash ADCs have been reported with sampling rates as high as 7.5GS/s [15].

A diagram of a 3-bit flash ADC is shown in Fig. 1.1. A resistive reference ladder generates 8 equally spaced reference voltages across the input span of the ADC. Each reference voltage is fed to its own comparator, along with the analog input signal V_{in} . Once every clock period the comparators make a decision, outputting a '1' (high voltage) or '0' (low voltage) signal depending on whether V_{in} is higher or lower than that particular reference voltage. The result is a 7-bit thermometer code carried by signal lines T0-T6. It is called a thermometer code because starting from the bottom the code contains '1's up to the input signal level, followed by '0's up to the top. As the input signal moves higher or lower the transition point from '1's to '0's will move up and down, similar to the way a thermometer responds to temperature changes. A digital thermometer decoder block converts the 7-bit thermometer code into a standard 3-bit binary code, which is the final output of the ADC.



Figure 1.1: A 3-bit Flash ADC



Figure 1.2: Conceptual Diagram of an N-bit folded flash ADC with a folding factor of 2

The drawback of flash ADCs is that for every additional bit, the ADC becomes twice as large and consumes twice as much power. For any number of bits N, 2^N comparators are required. The resistive ladder also consumes static power. For these reasons, flash ADCs are typically larger and more power hungry than other architectures. Various innovations have been reported to reduce this power and area penalty. For instance, in [16] the resistive ladder is removed completely. This ADC uses "dynamic comparators" which have built-in offsets controlled using banks of capacitors connected to each comparator. The drawback is that the offsets must be calibrated, but the 4-bit ADC is reported to operate at 1.25GS/s with a power consumption of only 2.5mW.

Another technique used to reduce the power and area requirements for flash ADCs is folding [17, 18]. The folding circuit is located in front of the standard flash architecture, as shown in Fig. 1.2. This diagram shows a folded flash ADC with a folding factor of 2, meaning that the folding stage adds an additional bit to the standard flash ADC. In this example, the input signal V_{in} can range from $-V_{amp}$ to V_{amp} , centred about 0. The comparator checks whether the input is greater than or less than 0. If $V_{in}>0$, the input is passed to the flash ADC as is. However if $V_{in}<0$ then V_{amp} is added to it. This way the input to the flash ADC is always greater than 0. The final binary output is then composed of the most significant bit (MSB) provided by the folding stage and the remaining bits provided by a flash ADC.

Using this scheme, an extra bit can be added to the flash ADC without doubling the



Figure 1.3: Pipeline ADC

number of comparators. Instead only one additional comparator is needed, plus some analog circuitry to perform the voltage addition. It is possible to use a folding factor greater than 2 as well. For instance, with a folding factor of 4 the folding stage will provide the two most significant bits, and the signal will be shifted by one of four possible levels (including 0) in order to fit within one-quarter of the original input span. A folding factor of 9 was used in [17]. The most power-efficient currently published flash ADC, based on the ISSCC figure-of-merit (see section 7.4) uses a folding factor of 2 [18].

1.2.2 Pipeline

The pipeline ADC can be thought of as an extension of the folding idea discussed in the previous section. Pipeline ADCs consist of multiple cascaded stages, each providing a small number of additional bits (typically 1-3), followed by a final flash ADC as shown in Fig. 1.3.

Pipeline ADCs get their name from the fact that on each clock cycle, a new input enters the first stage of the ADC while the existing inputs move one stage forward along the chain. This means that at a given time there are many different inputs being processed inside the ADC. Since the ADC does not have to wait to finish one input sample before starting the next, pipeline ADCs can achieve fairly high throughputs. However this also means that there is a latency between the time an input arrives at the ADC input and the time the



Figure 1.4: Single pipeline ADC stage

corresponding digital output is produced.

Each of the stages consists of a small flash ADC (which can be as simple as a single comparator for a 1-bit stage), along with a sample and hold block, a digital-to-analog converter (DAC), a subtractor and an amplifier, as shown in Fig. 1.4. After the flash ADC digitizes the signal, the digital value is converted back to analog using the DAC and subtracted from the original sampled input value. This results in a residue signal which (ignoring DAC inaccuracies) is the quantization error for the stage. The residue is then multiplied by a constant factor before being sent to the next stage. For instance, in a one-bit stage the maximum quantization error would be half of the input amplitude, so the residue would be multiplied by 2. In CMOS pipeline ADCs the sample and hold, DAC, subtractor and amplifier are typically all combined into a switched-capacitor circuit known as a multiplying DAC (MDAC).

As a result of the delays as an input signal moves through each stage of the ADC, a time alignment block is required to store the digital stage outputs and assemble them correctly once the conversion is complete. In order to reduce the accuracy requirements on the comparators, most pipeline ADCs are designed with extra bits in each stage [19,20]. For instance, rather than using 1-bit stages, 1.5 bit stages are used (so the flash ADC consists of 2 comparators). This allows for some overlap between the stages. This way if one stage makes an error for an input signal close to its threshold, the multiplied residue will still be within the input range of the next stage. Once the outputs of all stages are available, digital logic is then applied to correct the error. A 1.5 bit stage still only contributes a single bit to the final, corrected value.

While pipeline ADCs cannot quite compete with flash ADCs in terms of sample rates (due to the time taken for the analog components to operate), they offer increased resolution, typically at least 8 bits. Currently the fastest published pipeline ADC runs at 3GS/s [21].

1.2.3 Successive Approximation Register

Successive Approximation Register (SAR) ADCs have proven to be the most energy efficient ADC architecture. Based on the ISSCC figure-of-merit, the most efficient published ADCs are all SAR designs [22–25], with a maximum sampling frequency of 1MS/s. Even when considering only ADCs operating above 1GS/s, the current record holder uses a SAR architecture [26].

The basic idea behind SAR ADCs is as follows, referring to Fig. 1.5. The input signal V_{in} is sampled and provided to one input of a comparator for the duration of the conversion process. This process proceeds in an algorithmic fashion. First, the logic block sets the contents of the register to be a '1' followed by '0's, representing one-half of the full-scale value. The register's value is converted to an analog signal by the DAC. Modern power-efficient SAR ADCs use switched-capacitor-based charge redistribution DACs [12]. The comparator output determines whether the DAC output is higher or lower than the sampled input signal, completing the first step. Next, the digital logic uses the comparison result to set the register's value to the midpoint of the current possible space. For example, if the result of the first step was that the comparator indicated that the sampled input was greater than the DAC output, the signal must lie between 50 and 100% of the full scale value. In the second step the digital logic would set the register to 75% of the full scale value (two '1's followed by '0's for the remaining bits). In this way the ADC produces closer and closer approximations to the sampled input value. After N steps (one per bit), the conversion is



Figure 1.5: A basic successive approximation ADC

finished and the value held in the digital register is the ADC output for the current sample. This process then repeats for each new sample.

From this description, it is clear why SAR ADCs typically operate at low clock rates - each conversion requires multiple steps, and unlike pipeline ADCs, a new conversion cannot be started until the current one is completed. However, clever modifications to the basic design have allowed SAR-based designs to operate at published speeds as high as 1.25GS/s [27]. These innovations include the use of N comparators in parallel [28] and asynchronous rather than clocked operation [26, 27, 29]. While SAR ADCs have been able to break through the 1GS/s barrier, the prospect of multi-gigasample operation in the near future appears unlikely.

1.2.4 Time-Interleaving

The fastest ADCs use time-interleaving to achieve high sampling rates. A time-interleaved ADC consists of M separate ADCs, each running at $\frac{1}{M}$ of the full sampling rate (f_s), as shown in Fig. 1.6. These sub-ADCs can use any architecture including pipeline [30], flash [31] or SAR [31]. A front-end sample and hold samples the input at f_s and distributes the samples to the sub-ADCs. A timing generation circuit produces M clock signals with frequency $\frac{f_s}{M}$, each



Figure 1.6: A time-interleaved ADC

one phase-shifted from the next by $\frac{T_s}{M}$, where $T_s = \frac{1}{f_s}$. These slower clocks are distributed to the sub-ADCs so that each sub-ADC is responsible for converting one input sample every M periods of the full speed clock. The sub-ADCs typically have their own sample and hold circuits running at $\frac{f_s}{M}$. A multiplexer combines the outputs of the sub-ADCs into a single digital output running at the full sampling rate.

The advantage of time-interleaving is that it relaxes the speed requirements on the actual ADCs. As long as the front-end sample and hold is accurate, the sub-ADCs can operate below the f_s . The output multiplexer also operates at f_s but since the signals have already been digitized at this point this is not usually a problem. However, time-interleaved ADCs suffer from gain and offset mismatches between the ADCs. Often calibration is necessary to overcome these errors and this can increase the system complexity, particular for high levels of interleaving [30].

Time-interleaved ADCs typically do not achieve the same levels of energy efficiency as single-channel ADCs [32] due to the additional overhead of clock generation and routing, output multiplexing and calibration. However, time-interleaving enables the use of sample rates far exceeding anything possible in a single channel architecture. The fastest published time-interleaved ADC using CMOS technology runs at 40GS/s [33].

1.3 Time-Based ADCs

There are several ADC architectures that can be considered time-based; that is, they make use of time delays during the conversion process. This section will describe these architectures.

1.3.1 Integrating ADCs

Integrating ADCs generate a voltage ramp with slope proportional to the input, as shown in Fig. 1.7. A comparator is used to determine when the ramp reaches a fixed reference voltage (V_{ref}) . A timer counts the number of clock cycles between the start and end point of the ramp, and generates an output voltage based on this time delay.

One drawback to this scheme is that the ramp slope is highly sensitive to the resistor and capacitor values. A modification of this idea for improved accuracy is to first make the circuit ramp upwards proportional to V_{in} for a fixed amount of time, then ramp downwards proportional to V_{ref} until the ramp reaches zero volts. This is known as a dual-slope integrating ADC [12]. The time taken for the downward ramp ends up being proportional to the ratio of V_{in}/V_{ref} , independent of the analog component values.

For modern integrated circuits, analog op-amps are often too slow and power hungry to be practical. Instead, these circuits use switched-capacitor circuits and current sources to generate the ramps, and zero-crossing-based comparators (ZCBCs) to detect when the voltages on two signals are equal. These circuits are significantly faster and more power efficient than op-amp based integrating ADCs [34, 35].

While integrating ADCs can achieve high accuracy, it comes at the expense of speed.



Figure 1.7: An integrating ADC

Since a timer is used to measure the delay, the timing clock must be much faster than the sampling rate (a minimum of 2^{N} times faster). This makes integrating ADCs impractical for gigasample operation.

1.3.2 Voltage-Controlled Oscillator Based ADCs

ADCs based on voltage-controlled oscillators (VCOs) [36] use an input voltage to modulate the frequency of a VCO, then measure the frequency using a variety of methods. These methods include counting the number of VCO cycles in a set time period [37], sampling the VCO output with a fixed-frequency clock [38] or using a time-to-digital converter (TDC) [39]. As VCOs tend to be non-linear, the linearity must be corrected by the use of post-conversion look-up-tables [40] or VCO linearization techniques [41].

Like integrating architectures, VCO-based ADCs are only suited for low-sampling-rate applications. This is because the oscillator frequency must be significantly higher than the sampling rate, which is not practical in CMOS for sampling rates of 1GS/s or above.

1.3.3 Voltage-to-Time Converter Based ADCs

A voltage-to-time converter (VTC) is a circuit that takes in an analog voltage signal and produces a series of pulses where the delay on each pulse is proportional to the input at the time the pulse was generated. The signal consisting of delayed pulses can be described as



Figure 1.8: A time-based ADC consisting of a VTC and TDC

pulse-width modulated (PWM) or pulse-position modulated (PPM) depending on whether one or both pulse edges are delayed. A PWM signal has only one edge delayed so that a pulse gets wider or narrower depending on the input value. In a PPM signal the pulses maintain a constant width since both edges are delayed, but the entire pulse is shifted in time proportional to the input value. It can be said that a VTC takes information from the voltage domain, where the value of the signal is proportional to the voltage amplitude, and converts it to the time domain, where the value of the signal is proportional to the time delay on a digital pulse edge.

To complete the ADC, the VTC output is fed into a TDC, which measures the delay between pulse edges on two signals and converts the value to a digital representation. As a fully digital circuit, the TDC is able to leverage the switching-speed of deep-submicron CMOS technology. The VTC-based ADC is shown in Fig. 1.8.

High-speed VTC circuits in CMOS are based on the idea of a current-starved inverter. In this design, a standard CMOS inverter has one or more additional transistors added to limit the current available for switching. These current-starving devices are controlled by the analog input to allow more or less current depending on the input value, which modifies the output delay of the inverter.

VTCs tend to exhibit two problems: non-linearity and susceptibility to process, voltage and temperature (PVT) variations. Linearization of a VTC has been reported using multiple input starving transistors in parallel [42], although manual bias tuning is required. The ADC in [43] uses non-linear VTCs to convert both the input and a set of reference voltages generated by a resistive ladder. Time-based comparators are then used to digitize the signal in a flash architecture. In [44] the VTC and TDC are not separate blocks; instead VTCs are distributed throughout the TDC. A novel clock generation scheme is used to automatically adjust to PVT variations.

This thesis will present a new VTC-based ADC.

1.4 Time-to-Digital Converters

TDCs are used to measure the time delay between two pulse edges and produce a digital value. These two edges are typically referred to as a "start event" and a "stop event". Historically TDCs have been used for various applications, including particle detection for nuclear science [45], range finders based on pulsed lasers [46], and measurement devices such as oscilloscopes [47]. More recently TDCs have become more popular for integrated circuit applications including digital phase-locked loops [48] and ADCs.

There are various TDC architectures to choose from depending on the length of delay being measured and the time resolution needed. For relatively coarse resolutions, a TDC can be implemented using a simple timer, counting the number of clock cycles between the start and stop events. The time resolution is then limited to a single clock period. If high resolution is needed but the conversion time can be long, often a two-step TDC will be implemented where a timer is used for a coarse measurement, and the remaining residue (less than one clock period) will be measured with a high resolution TDC. A time-amplifier [49] can also be used to measure a short delay with high resolution. It works by creating a delay proportional to the original short delay but many times longer, so that it can be measured with a more coarse technique. However, this limits the throughput of the converter.

For high-speed ADC applications it is necessary to have a TDC with high resolution as well as high throughput. The required TDC throughput is equal to the ADC sampling rate,



Figure 1.9: An N-bit delay line TDC with resolution t_{δ}



Figure 1.10: An N-bit Vernier delay line TDC with resolution t_{δ}

while the required resolution (t_{δ}) is equal to

$$t_{\delta} = \frac{t_{max}}{2^N} \tag{1.1}$$

where t_{max} is the maximum delay produced by the VTC in an N-bit ADC. The maximum delay must be less than the length of a clock period, so for multi-gigasample ADCs the required resolution can be below 10ps.

An architecture capable of operating under these conditions is the delay-line TDC [50]. As shown in Fig 1.9, this architecture sends the start pulse through a chain of $2^N - 1$ delay buffers. The start pulse will make its way through the chain over time, limited by the delay of each buffer. The output of each buffer is tapped and sent to a flip-flop clocked with the stop pulse. When the stop event occurs the flip-flops will sample the buffer outputs. The output of a flip-flop will be '0' if the buffered start pulse occurred before the stop pulse, or '1' if the buffered start pulse occurred after the stop pulse. These flip-flop outputs are in fact a thermometer-encoded value of the delay being measured. The outputs can then be sent to a thermometer decoder to produce standard binary outputs.

The resolution of the delay-line TDC is limited to the delay of one buffer, which in CMOS is normally implemented as two inverters. Even in a 65nm process the delay of 2 inverters is greater than 10ps. A modification to allow for higher resolution uses a Vernier delay-line (VDL) [51]. This design is similar to the simple delay-line TDC except that both the start and stop pulses are sent through a chain of buffers. The buffers are designed so that each buffer in the upper chain has a slightly higher delay than each buffer in the lower chain by t_{δ} . The delay between the start and stop pulses will decrease by this difference after each buffer, where both lines are tapped and sent to flip-flops. The result is again a thermometer-coded signal with resolution t_{δ} . The advantage of the VDL is that t_{δ} can be set to less than one buffer delay. The disadvantage is that it requires double the number of buffers.

An alternative to the delay-line TDCs is a parallel or "flash" TDC, inspired by the flash ADC architecture [52,53]. Rather than using a chain of delay buffers, this architecture uses $2^N - 1$ parallel delay buffers, each leading to a flip-flop. The first buffer delays the signal by t_{δ} , the second by $2t_{\delta}$, and so on. The result is a thermometer-encoded output just as with the delay-line TDCs. A diagram of a flash TDC can be found in section 4.2. The disadvantage of a flash TDC is that many more delay cells are needed, resulting in increased chip area and power consumption. However, independent tuning for each threshold can be achieved, which is impossible for delay-line TDCs.

High-speed TDCs have been reported with resolutions as low as 1ps operating in the 100-200MHz range [53, 54].



Figure 1.11: Artist's rendering of dish antennas for the Square Kilometre Array (Credit:SKA Organization/TDP/DRAO/Swinburne Astronomy Productions) http://www.skatelescope.org

1.5 Motivation: The Square Kilometre Array

Of particular interest for this work is the Square Kilometre Array (SKA) project, an international collaboration to build the most sensitive radio telescope in the world [55]. There is a need for high speed, low resolution ADCs as part of the millions of receivers that will be part of the system. An advantage of the time-based ADC for this application is that the VTC and TDC can be physically separated. For example, each antenna could have a VTC in the antenna feed, possibly even integrated with other components in the analog signal chain such as low-noise amplifiers [56]. The outputs of the VTCs could then be transmitted via optical fibre to the base of the antenna, where a TDC would digitize the signal. This would minimize the noisy digital-switching circuits in the vicinity of the antenna feed, and would also reduce the power that must be routed up the antenna.

1.6 Thesis Organization

This thesis consists of eight chapters, including this introduction. Chapter 2 introduces the linearity metrics used to evaluate the VTC, TDC and ADC. It describes and compares two measurement techniques, the FFT method and the histogram method. Mathematical techniques are then developed to analyze linearity analytically for arbitrary functions.

Chapter 3 presents the VTC circuit and describes its operation in detail. A model is developed to predict the VTC delay. The analytical techniques from the previous chapter are then used to analyze the linearity of the VTC circuit and optimize it for either singleended or differential operation. Finally, jitter models are developed to predict the jitter of a CMOS inverter, a current-starved inverter and the full VTC circuit.

Chapter 4 briefly presents the design and measurement of a first generation prototype of a 3-bit time-based ADC based on a VTC and TDC, fabricated in 90nm CMOS.

Chapter 5 gives a detailed description of the finalized VTC circuit, fabricated in 65nm CMOS. Simulated and measured results are shown. This chapter also describes an automatic calibration system for the VTC based on a delay-locked loop, which was implemented on the chip.

The 65nm TDC circuit is presented in Chapter 6. The reasoning behind the choice to use a VDL architecture is explained, along with the design and simulated results of each subcircuit. An on-chip serial delay tuning system is explained, as well as its use in an algorithm for automatically calibrating the TDC. The layout is shown, followed by measured results.

Chapter 7 presents measured results for the full 4-bit ADC using the 65nm VTC and TDC.

Finally, Chapter 8 summarizes the contributions of this thesis to the field and suggests possible directions for future work.

Chapter 2

Linearity Analysis

Linearity is an important property of any converter, including ADCs, VTCs and TDCs. There are various metrics for quantifying linearity, including integral non-linearity (INL), differential non-linearity (DNL), total harmonic distortion (THD), signal to noise and distortion ratio (SINAD), and effective number of bits (ENOB). Of these, ENOB is the most commonly used for data converters, due to its intuitive nature. For example, if one is told that a 4-bit ADC has an ENOB of 3.2 bits, this is much easier to grasp than being told that the SINAD is 21dB (even though the two metrics are exactly equivalent). This chapter will explore several concepts relating to linearity and ENOB, beginning with the basic definition and derivation.

2.1 Derivation of ENOB

ENOB is commonly defined using the formula

$$ENOB = \frac{SINAD - 1.76}{6.02}.$$
 (2.1)

This formula is based on the quantization noise of an ideal converter, first calculated in [57]. To calculate quantization noise, first consider Fig. 2.1a. This plot shows the staircaselike output versus input curve of a quantizer, also known as the transfer curve. The height of each step is q, representing one least significant bit (LSB). The non-quantized (ideal) curve is also shown as a dashed line. The difference between the two curves is the quantization error, e(x), plotted in Fig. 2.1b. The error is a sawtooth function ranging from $-\frac{1}{2}q$ to $\frac{1}{2}q$ for each step.

To calculate the quantization error, we need to find the mean-square value of e(x). We can simplify the calculation by only considering a single step, say from x = 0 to $x = x_0$.



Figure 2.1: ENOB derivation: (a) Quantized staircase and ideal output (b) Quantization error

Over this range, the error can be described by the function

$$e(x) = \frac{q}{x_0}x - \frac{q}{2}, \qquad 0 \le x \le x_0.$$
 (2.2)

We can then find the root-mean-square (RMS) value of e(x) as follows:

$$\overline{e(x)^2} = \frac{1}{x_0} \int_0^{x_0} \left(\frac{q}{x_0}x - \frac{q}{2}\right)^2 dx$$
(2.3)

$$= \frac{1}{x_0} \int_0^{x_0} \left(\frac{q^2}{x_0^2} x^2 - \frac{q^2}{x_0} x + \frac{q^2}{4} \right) \mathrm{dx}$$
(2.4)

$$= \frac{1}{x_0} \left[\frac{1}{3} \frac{q^2}{x_0^2} x_0^3 - \frac{1}{2} \frac{q^2}{x_0} x_0^2 + \frac{q^2}{4} x_0 \right]$$
(2.5)

$$=\frac{1}{3}q^2 - \frac{1}{2}q^2 + \frac{1}{4}q^2 \tag{2.6}$$

$$=\frac{1}{12}q^2$$
 (2.7)

or an RMS value of

$$\sqrt{\overline{e(x)^2}} = \frac{1}{\sqrt{12}}q.$$
 (2.8)

The quantization error acts as noise in the output of a converter. In the frequency domain this noise is evenly spread between 0 and $\frac{1}{2}f_s$, half the sampling frequency. We can now calculate the SINAD of the output, assuming the signal is a sinusoid covering the full span of the converter. The span of an N-bit converter is $2^N q$, so the input signal is

$$s(t) = \left(\frac{1}{2}2^N q\right) \sin(2\pi f_0 t) \tag{2.9}$$

where f_0 is the signal frequency. To find the RMS value of the signal, we simply divide the amplitude by the square root of 2, so we have

$$\sqrt{\overline{s(t)^2}} = \frac{1}{2\sqrt{2}} 2^N q.$$
 (2.10)

Now the SINAD can be calculated as

$$SINAD = 20 \log \left(\frac{\sqrt{\overline{s(t)^2}}}{\sqrt{\overline{e(x)^2}}} \right)$$
(2.11)

$$= 20 \log \left(\frac{\frac{1}{2\sqrt{2}} 2^N q}{\frac{1}{\sqrt{12}} q} \right) \tag{2.12}$$

$$= 20 \log\left(\frac{\sqrt{6}}{2}2^N\right) \tag{2.13}$$

$$= 20 \log\left(\frac{\sqrt{6}}{2}\right) + 20 \log\left(2^N\right) \tag{2.14}$$

$$= 1.76 + N \bullet 20 \log(2) \tag{2.15}$$

$$= 1.76 + 6.02N. \tag{2.16}$$

So far we have simply calculated the effect of quantization noise in an ideal converter. The concept of ENOB is an abstraction used to characterize real converters. A real converter will have a particular SINAD which may include the results of thermal noise and jitter, nonlinearity, sampling error, and any other source of noise and distortion. The idea of ENOB is to relate this SINAD value to an ideal converter with only quantization noise. The standard formula of equation 2.1 is obtained by simply solving equation 2.16 for N.

It should be noted that this formula is a slight approximation, since the quantization noise was derived for a linear signal. The true quantization noise for a sinusoidal signal is



Figure 2.2: Error in calculated SINAD formula for quantization noise

much more complex to analyze [58]. The inaccuracy is generally considered insignificant, but it becomes worse for converters with a small number of bits. Since this thesis deals with 3and 4-bit converters, it is important to examine the accuracy of equation 2.16. A simulation was performed by quantizing a sinusoidal input made up of 2^{20} samples. The simulated SINAD was compared to the calculated value for converters between 2 and 12 bits, with the results shown in Fig. 2.2. The upper two plots show the SINAD error expressed in dB and as a percentage of the true SINAD value (still in dB). The bottom plot shows the error in bits. For a 3-bit converter, this error amounts to 0.05 bits. This is not a significant amount for the work presented here, so the standard formula will be used throughout.

2.1.1 ENOB of Non-Quantized Systems

Normally ENOB is used to describe the linearity of a quantized converter. However, in certain cases it can also be useful to use ENOB to evaluate non-quantized circuit blocks as well. For instance, in a receiver chain it is useful to ensure that the circuit blocks prior to the ADC, such as amplifiers, filters and mixers, have an equivalent ENOB at least as high as that of the ADC. Similarly, in a transmitter the blocks following a DAC should have an equivalent ENOB greater than or equal to that of the DAC. Whether the system is quantized or not, ENOB is really just another way of expressing the SINAD.

2.2 Measuring ENOB

There are two common techniques for measuring ENOB: the histogram method and the fast Fourier transform (FFT) method.

2.2.1 The Histogram Method

The histogram method [59] involves applying an input covering the full span of the converter and collecting samples. The input frequency should be such that a repeating pattern will not be formed with the sampling frequency over the duration of the test. Every sample can be saved, or every Mth sample (decimation), or random samples. As the test proceeds, a count is kept of how many times each output code is produced.

The analysis of the results depends on the type of input signal used. The simplest case is a linear ramp signal. In this case, an ideal converter produces a uniform histogram. The DNL for each output code can be found by dividing the actual count by the ideal count and subtracting 1. The INL for a particular code is the cumulative sum of the DNLs up to that code.

For high-speed testing, it is not practical to generate linear ramp signals. Instead, sinusoidal signals are used. This complicates the analysis of the histogram test, since the expected histogram is no longer linear. Following [59], the sinusoidal histogram is analyzed as follows. First, a cumulate histogram CH is built from the regular histogram H:

$$CH(i) = \sum_{k=1}^{i} H(k)$$
 (2.17)

The transition levels of the converter can then be calculated (normalized to the range of -1 to 1) using the formula

$$V(i) = -\cos\left(\frac{\pi C H(i)}{C H(2^N)}\right)$$
(2.18)

for i between 1 and $2^N - 1$ (the number of transitions). The DNL and INL can then be found directly from these transition levels.

ENOB can be calculated from the histogram (or transition levels for sinusoidal inputs) by directly computing the quantization error of each output step, using the method described in [60].

2.2.2 The FFT Method

In the FFT method [61], a data record of a specified length is recorded from the output of the converter. The record can include all samples, or every M^{th} sample (decimation). For best results, the test must be arranged so that an exact integer number of input cycles occurs during the test period - this is known as coherent sampling. In other words, the following should be true:

$$f_0 = \frac{n}{c} f_s \tag{2.19}$$

where f_0 is the signal frequency, f_s is the sampling frequency, n is the total number of samples in the record, and c is the integer number of cycles in the record. Any integer can be used for c in order to test different input frequencies.

Once the record has been taken, an FFT is performed on the data to obtain the frequency response. The SINAD can then be calculated directly from the FFT data, and the ENOB is calculated using the standard formula.

2.2.3 Comparison Between Histogram and FFT Methods

There are significant differences between the two FFT methods discussed. The main differences can be categorized as hardware requirements, noise sensitivity and amplitude sensitiv-


Figure 2.3: Simulated effect of noise on FFT and histogram ENOB calculation methods ity.

In terms of hardware requirements, the main difference between the methods is that the FFT method requires the samples to be either a complete record of adjacent samples, or decimated by a constant amount. In contrast, the histogram method works even if the samples are random and in no particular order. This allows histogram tests to be performed with more basic equipment, without requiring a large storage capacity or precise timing.

The two tests respond very differently in the presence of random noise. The histogram test measures only the non-linearity of the converter and is largely immune to noise since noise will tend to average out between the different histogram bins during the test. The FFT test, however, is a measure of both noise and non-linearity. Fig. 2.3 shows the stark contrast between the methods with increasing amounts of noise for an ideal 4-bit converter. For example, input noise with an RMS amplitude of 5% of the input span degrades the FFT ENOB by 1.5 bits, while the histogram ENOB degrades by less than 0.05 bits.

Amplitude sensitivity is the other main difference between the methods. Ideally, an



Figure 2.4: Effect of input amplitude on FFT and histogram ENOB measurement methods

ENOB test will use an input signal with an amplitude exactly matching the full scale input span of the converter. In practice however, sometimes it may be difficult to match the input span precisely. Fig. 2.4 shows the effect that input amplitude variation has on each method. The figure uses real measured data from the 65nm ADC, as well as simulated data for an ideal 4-bit converter with simulated noise added. It can be observed that the histogram method is very sensitive to amplitude variation, with steep drop-offs on either side of the ideal value. The reason for this is that an incorrect input amplitude skews the histogram away from the ideal counts in each bin. The FFT has a more gentle roll-off for incorrect input amplitude. The reason for the roll-off in the FFT ENOB for lower input amplitudes is a decrease in the relative power of the signal to the fixed quantization noise, while the reason for the ENOB decrease for higher amplitudes is non-linear behaviour due to clipping.

Based on this comparison, it is clear that the FFT method is superior for evaluating a converter, since it includes the effects of random noise while not being unnecessarily degraded by small variations in the test signal amplitude. However, there is an application for which the histogram method is ideal: an automatic tuning system. For this application, insensitivity to noise is an advantage since random variations will only increase the time needed for a tuning loop to converge. Furthermore, the steep slope of the ENOB versus input amplitude curve will help the tuning loop converge quickly to the correct input span. An automatic calibration system for the TDC using the histogram method is detailed in section 6.8.

2.3 Analytic Linearity Calculation

The linearity of an analog block can be found by applying a sinusoidal input, calculating the SINAD of the output, and converting this value to an ENOB. The same process is used whether we are measuring circuits in the lab, performing numerical simulations or using analytic models for hand calculations. We will consider analytic models for this analysis.

2.3.1 Analytic Calculations

Analytic models have an advantage over numerical calculation because they can be done by hand and offer physical insight into a circuit. Calculating SINAD (and thus ENOB) analytically is usually complicated by the necessity of finding each harmonic. However, we can start with the trivial example of an ideal amplifier. The amplifier's output y is related to its input x by the transfer function y = Ax, where A is the gain. We also define the valid range for x to be from -1 to 1, matching the output range of the standard sine function. To begin the process, we set the input to be sinusoidal:

$$x = \sin(t). \tag{2.20}$$

The output of the amplifier becomes

$$y = A\sin(t). \tag{2.21}$$

Since the output does not contain any harmonics other than the fundamental, the SINAD can be calculated. Obviously the SINAD (and thus the ENOB) is infinite for a perfectly linear circuit, since there are no harmonics other than the fundamental.

For a more interesting analysis, consider a transfer function of the form

$$y_1(x) = A[x-k]^2. (2.22)$$

This could be, for example, the large-signal transfer function of a common-source amplifier using the CMOS Level 1 model. Applying the input $x = \sin(t)$, the output becomes

$$y_1(t) = A[\sin(t) - k]^2.$$
 (2.23)

By using a sinusoid, we have again set the range of x to be from -1 to 1. The harmonic components must be separated, so we can simplify the above to

$$y_1(t) = A[\sin^2(t) - 2k\sin(t) + k^2]$$
(2.24)

The $\sin^2(t)$ term is still a problem - we need an expression containing only linear sinusoidal terms. We can use the trigonometric identity

$$\sin^2(t) = \frac{1}{2} [1 - \cos(2t)] \tag{2.25}$$

to transform the equation into a useable form:

$$y_1(t) = A\left[\frac{1}{2}(1 - \cos(2t)) - 2k\sin(t) + k^2\right]$$
(2.26)

$$y_1(t) = A\left(k^2 + \frac{1}{2}\right) - 2Ak\sin(t) - \frac{A}{2}\cos(2t)$$
(2.27)

The expression is now separated into its harmonic components. The DC component $A\left(k^2 + \frac{1}{2}\right)$ is not needed for the linearity analysis and can be ignored. We are left with the fundamental, having amplitude -2Ak, and the second harmonic having amplitude $-\frac{A}{2}$. The SINAD, in linear units, is

$$SINAD = \frac{(-2Ak)^2}{(-\frac{A}{2})^2}$$
(2.28)

$$= 16k^2$$
 (2.29)



Figure 2.5: ENOB of example functions for $-1 \le x \le 1$

Thus the linearity is increased by increasing k. We can use this result to gain physical insight into a circuit. The SINAD can be converted to ENOB using the standard equation. The result is plotted in Fig. 2.5. We can express the transfer function of a CMOS common-source amplifier in the standard form of equation 2.22 as follows:

$$y_1(x) = A[x-k]^2 = -R_{out} \frac{\mu_n C_{ox}}{2} \frac{W}{L} [V_{amp}x + V_{bias} - V_T]^2$$
(2.30)

$$= -R_{out}V_{amp}^2 \frac{\mu_n C_{ox}}{2} \frac{W}{L} \left[x - \left(\frac{-V_{bias} + V_T}{V_{amp}}\right) \right]^2$$
(2.31)

Here V_{amp} is the input amplitude and V_{bias} is the DC bias voltage for the input. Comparing this expression to equation 2.22, it is clear that

$$k = \left(\frac{-V_{bias} + V_T}{V_{amp}}\right) \tag{2.32}$$

and therefore we can say

$$SINAD = 16 \left(\frac{-V_{bias} + V_T}{V_{amp}}\right)^2.$$
(2.33)

Since we know V_{bias} must be a positive value greater than V_T , we can maximize the amplifier's linearity by making the DC overdrive voltage $V_{bias} - V_T$ large and keeping the input amplitude V_{in} small.

The algebraic technique used for this example will work for any polynomial transfer function. For any positive integer exponent of sin(t), trigonometric identities can be used to transform the term into a linear combination of sine or cosine terms for different harmonics. For example, the first few identities are

$$\sin^2(t) = \frac{1}{2} [1 - \cos(2t)] \tag{2.34}$$

$$\sin^3(t) = \frac{1}{4} \left[3\sin(t) - \sin(3t) \right]$$
(2.35)

$$\sin^4(t) = \frac{1}{8} [-4\cos(2t) + \cos(4t) + 3]$$
(2.36)

$$\sin^5(t) = \frac{1}{10} [10\sin(t) - 5\sin(3t) + \sin(5t)].$$
(2.37)

It is interesting to note that odd powers of sin(t) include only odd harmonics (including the fundamental) while even powers of sin(t) include only even harmonics (which do not include the fundamental) and DC components. This fact will be important in the analysis of differential input in section 2.4.

Unfortunately, many important functions cannot be broken down into their harmonic components by simple trigonometric identities and algebra. These functions require more advanced techniques for analysis.

2.3.2 Using Series Expansions

Infinite series expansions can be used to approximate the harmonics of non-polynomial functions. Volterra series and Taylor series techniques are useful for expressing a function as a polynomial approximation [62], which can then be analysed as in the previous section.

As an example, consider the expression

$$y_2(x) = \frac{A}{x-k} = \frac{A}{\sin(t)-k}, \qquad |k| > 1.$$
 (2.38)

The constraint on k is needed to avoid y becoming infinite. After making the substitution $x = \sin(t)$, it's clear that this function cannot be algebraically decomposed into a linear

combination of sinusoids. However, we can express the function as a Taylor series:

$$y_2(x) = -\frac{A}{k} - \frac{A}{k^2}x - \frac{A}{k^3}x^2 - \frac{A}{k^4}x^3 - \frac{A}{k^5}x^4 - \dots$$
(2.39)

The terms can be labelled T_0 for the DC term, T_1 for the *x* coefficient, T_2 for the x^2 coefficient, and so on. Making the substitution $x = \sin(t)$ and using the trigonometric identities developed in the previous section, we can now approximate the function as a linear combination of sinusoids. Limiting the analysis to the first 3 harmonics, we have

$$y_2(t) = T_0 + T_1 \sin(t) + T_2 \sin^2(t) + T_3 \sin^3(t)$$
(2.40)

$$= T_0 + T_1 \sin(t) + T_2 \left(\frac{1}{2} [1 - \cos(2t)]\right) + T_3 \left(\frac{1}{4} [3\sin(t) - \sin(3t)]\right)$$
(2.41)

$$= \left[T_0 + \frac{1}{2}T_2\right] + \left[T_1 + \frac{3}{4}T_3\right]\sin(t) - \left[\frac{1}{2}T_2\right]\cos(2t) - \left[\frac{1}{4}T_3\right]\sin(3t).$$
(2.42)

Using this expression, we can now calculate the SINAD:

$$SINAD = \frac{\left[T_1 + \frac{3}{4}T_3\right]^2}{\left[\frac{1}{2}T_2\right]^2 + \left[\frac{1}{4}T_3\right]^2}$$
(2.43)

Making the substitutions $T_1 = -\frac{A}{k^2}$, $T_2 = -\frac{A}{k^3}$, $T_3 = -\frac{A}{k^3}$ and simplifying gives us

$$SINAD = \frac{(4k^2 + 3)^2}{4k^2 + 1}.$$
 (2.44)

The SINAD increases without bound as the absolute value of k increases beyond 1. So we can maximize the linearity of the function $y = \frac{A}{x-k}$ by making the absolute value of k as large as possible. The ENOB is shown in Fig. 2.5.

2.4 Differential Inputs

Using differential inputs for RF signals has advantages including noise rejection and doubling the output signal without increasing headroom requirements [63]. In the case of a VTC the headroom advantage applies not to voltage but to time - the VTC output can have double the delay range without modifying the timing constraints. Another important advantage of differential operation is improved linearity. In this section, the linearity advantage will be analyzed using the techniques developed previously.

2.4.1 Odd and Even Functions

The concept of odd and even functions is useful in understanding the effect of differential inputs on linearity. We will consider functions normalized to have an input range of $-1 \le x \le 1$, and therefore centered around x = 0. A function exhibiting symmetry about the line x = 0 is classified as an even function. Formally, even functions obey the relation $f_e(-x) = f_e(x)$. A function exhibiting inverse symmetry about x = 0 is classified as an odd function, obeying the equation $f_o(-x) = -f_o(x)$.

Mathematically, any function f(x) can be expressed as the sum of an even and an odd function. So if we establish the effect of differential inputs on both even and odd functions, any function can then be analyzed by breaking it into even and odd components. An important property of even and odd functions relates to the harmonics of these functions when given sinusoidal inputs. Even functions have only even harmonics, while odd functions have only odd harmonics.

First, consider the general form of a function with differential input:

$$f_d = f(x) - f(-x)$$
(2.45)

For an even function, the differential form will be

$$f_d(x) = f_e(x) - f_e(-x)$$
(2.46)

$$= f_e(x) - f_e(x)$$
 (2.47)

$$=0 \tag{2.48}$$

An odd function's differential form will be

$$f_d(x) = f_o(x) - f_o(-x)$$
(2.49)

$$= f_o(x) + f_o(x)$$
 (2.50)

$$=2f_o(x) \tag{2.51}$$

Therefore the effect of differential input on any function is that the even portion of the function is eliminated while the odd portion is doubled. In terms of harmonics, the output of a differential function will contain only odd harmonics. Since even harmonics are pure distortion, differential operation improves linearity.

The examples from the previous sections will now be analyzed with differential inputs.

2.4.2 Differential Examples

The first non-trivial example analyzed previously was the function $y_1 = A[x - k]^2$. The differential form of this function is

$$y_{d1}(x) = A[x-k]^2 - A[(-x)-k]^2$$
(2.52)

$$= -4Akx. \tag{2.53}$$

The differential form is in fact a linear equation and thus has an infinite SINAD and ENOB. This is expected since the single-ended form contained distortion in the form of a single second harmonic component (equation 2.27) and it has just been established that differential operation eliminates even harmonics. Furthermore, the coefficient of the linear term has doubled, from -2Ak in the single-ended expression to -4Ak in the differential version.

The second single-ended example analyzed was $y = \frac{A}{x-k}$ with the constraint |k| > 1. Used differentially, the equation becomes

$$y_{d2}(x) = \frac{A}{x-k} - \frac{A}{(-x)-k}$$
(2.54)

$$=\frac{2Ax}{x^2 - k^2}.$$
 (2.55)

The Taylor series representation of this function is

$$y_{d2}(x) = -\frac{2A}{k^2}x - \frac{2A}{k^4}x^3 - \frac{2A}{k^6}x^5 - \dots$$
(2.56)

Comparing this to the Taylor expansion of the single ended function (equation 2.39), it is apparent that the odd harmonics have doubled in magnitude while the even harmonics have disappeared, as expected.

Since there are only odd harmonics, we will analyze the linearity for the first 5 harmonics. Once again denoting the terms as T_1 for the linear term, T_3 for the x^3 term and T_5 for the x^5 term, and letting $x = \sin(t)$, the function can be expressed as

$$y_{d2}(t) = T_1 \sin(t) + T_3 \sin^3(t) + T_5 \sin^5(t)$$

$$= T_1 \sin(t) + T_3 \left(\frac{1}{4} [3\sin(t) - \sin(3t)] \right) + T_5 \left(\frac{1}{10} [10\sin(t) - 5\sin(3t) + \sin(5t)] \right)$$
(2.58)

$$= \left[T_1 + \frac{3}{4}T_3 + T_5\right]\sin(t) + \left[-\frac{1}{4}T_3 - \frac{1}{2}T_5\right]\sin(3t) + \left[\frac{1}{10}T_5\right]\sin(5t)$$
(2.59)

so the SINAD is

$$SINAD = \frac{\left[T_1 + \frac{3}{4}T_3 + T_5\right]^2}{\left[-\frac{1}{4}T_3 - \frac{1}{2}T_5\right]^2 + \left[\frac{1}{10}T_5\right]^2}$$
(2.60)

$$=\frac{(4k^4+3k^2+4)^2}{k^4+4k^2+4.16}\tag{2.61}$$

As with the single-ended SINAD, (equation 2.44), the differential SINAD increases without bound as the absolute value of k increases, so the linearity is maximized by making k as large as possible. However, plotting the linearity of both the single-ended and differential functions (Fig. 2.6) shows that the differential linearity increases dramatically faster than the single-ended linearity.

It can also be noted that if T_5 is assumed to be small and discarded, the differential SINAD simplifies to $(4k^2 + 3)^2$. This is larger than the single-ended SINAD by a factor of $4k^2 + 1$, which explains the exponential improvement in Fig. 2.6.

Not all functions will be improved by differential inputs however. As a counter-example, take the odd function $y_3(x) = Ax^3$. Applying the input $x = \sin(t)$ and using the trig



Figure 2.6: Comparison of single-ended and differential ENOB for $y = \frac{A}{x-k}$

identities mentioned previously, this function can be expressed as

$$y_3(t) = \frac{3A}{4}\sin(t) - \frac{A}{4}\sin(3t).$$
(2.62)

The SINAD is

$$SINAD = \frac{\left(\frac{3A}{4}\right)^2}{\left(\frac{A}{4}\right)^2} \tag{2.63}$$

$$= 9$$
 (2.64)

The differential form is $y_{3d}(x) = Ax^3 - A(-x)^3 = 2Ax^3$. Since the SINAD above is independent of A, multiplying the equation by 2 will not affect the ENOB. This makes sense since only odd harmonics were present in the single-ended function, so using differential inputs doubled the output but did not otherwise effect the harmonics. However, if the more general function $y = A(x - k)^3$ had been used instead, differential operation would still improve the linearity for non-zero k because this is not an odd function.

Chapter 3

VTC Analysis - Theory



Figure 3.1: VTC schematic with parasitic capacitor

This chapter will present an analysis of the VTC including an explanation of the physical operation of the circuit, a derivation of the output delay using hand-analysis, optimization of the output linearity, a design procedure for both single-ended and differential VTC variations and a jitter analysis. A schematic diagram of the VTC under analysis is shown in Fig. 3.1. This design is used for the fabricated 65nm VTC presented in Chapter 5.

3.1 Explanation of Operation

The operation of the VTC can be explained by considering idealized transistor models with the addition of parasitic capacitor C_{out} , which is mainly composed of the drain capacitances of M1 and M2 and the gate capacitances of M5 and M6. It is assumed that all other node capacitances are significantly less than C_{out} (when designing the VTC the device sizes should be chosen accordingly to ensure this assumption is true).

In the figure, M1-M4 make up a voltage-starved inverter, while M5-M6 form a standard CMOS inverter used to sharpen the edges of the signal $V_{out}(t)$. The gate input to M3 is the input signal to the VTC, V_{in} . In this analysis, it will be assumed that V_{in} can be considered constant over a single VTC conversion cycle. Simulations show that this assumption is highly accurate even for AC input signals at the Nyquist frequency of the converter. The reason for this is that the VTC is only sensitive to the input for a short time during the switching process, so it effectively samples the input. The gate input to M4, V_{const} , is a DC bias voltage used to tune the gain and linearity of the VTC.

Since the M1-M2 inverter has starving devices between M2 and ground but not between M1 and V_{DD} , falling edges of Clk will be passed through to V_{out} with minimal delay. However, rising edges of Clk will be slowed down by the starved inverter, depending on the value of V_{in} . The delay on this edge, and how it varies with V_{in} , is what we are interested in analyzing.

A basic summary of the VTC operation is as follows: When a rising edge occurs on Clk(t), $V_{out}(t)$ begins to ramp downwards from V_{DD} at a rate dependent on V_{in} . When this ramping signal reaches the threshold of the M5-M6 inverter, a rising edge is triggered on the inverter output. This operation will now be analyzed in detail. In order to develop an intuitive model for hand calculation, the analysis uses Level 1 transistor models with $\lambda = 0$. All device lengths and threshold voltages are assumed equal.

3.1.1 Detailed sequence of operation

0. Initial Conditions

The starting point for the conversion is a steady state with Clk(t) and $V_A(t)$ equal to 0 and $V_{out}(t)$ equal to V_{DD} . In the starved inverter, devices M1, M3 and M4 are in deep triode while M2 is in cut-off. In the second inverter, M5 is in cut-off while M6 is in deep triode.

1. Rising edge on clock input

When Clk(t) goes high, M1 enters cut-off. M2 turns on, entering the saturation region. Charge stored on C_{out} flows through M2. M3 and M4 are still in deep triode so conduct very little current. The result is a rapid voltage increase of $V_A(t)$.

2. M3 and M4 enter saturation

Since $V_A(t)$ is increasing rapidly, it can be assumed that M3 and M4 enter saturation mode simultaneously even if $V_{in} \neq V_{const}$. Once M3 and M4 are saturated, they limit the current flowing through M2 to a constant amount, I_{max} . This causes V_{out} to linearly ramp down at a constant rate. The fixed current can be calculated as

$$I_{max} = I_3 + I_4 (3.1)$$

$$= \frac{1}{2}KP_3[V_{in} - V_T]^2 + \frac{1}{2}KP_4[V_{const} - V_T]^2$$
(3.2)

where

$$KP_3 = \mu C_{ox} \frac{W_3}{L}$$
 and $KP_4 = \mu C_{ox} \frac{W_4}{L}$ (3.3)

and the ramp rate is

$$R_{ramp} = \frac{C_{out}}{I_{max}}.$$
(3.4)

The constant current through M2 sets the gate-source voltage of M2 to a constant value. Since the gate is held constant at V_{DD} , the voltage of $V_A(t)$ is fixed at

$$V_{A,max} = V_{DD} - V_{GS2} \tag{3.5}$$

$$= V_{DD} - \left(\sqrt{\frac{W_3(V_{in} - V_T)^2 + W_4(V_{const} - V_T)^2}{W_2}} + V_T\right).$$
(3.6)

3. M2 enters triode

This occurs when $V_{DS2} \leq V_{GS2} - V_T$, which will correspond to $V_{out}(t)$ dropping

by V_T . After this point, V_{GS2} will no longer be constant, causing V_A to decrease. However, the current through M2 will still be I_{max} , so the linear ramp on V_{out} is maintained. It is expected that $V_{out}(t)$ will reach the threshold of the M5-M6 inverter during this step, triggering a rising edge on the inverter output. $V_{out}(t)$ will no longer affect the VTC output after this switching point.

4. M3 and M4 enter triode

Once either M3 or M4 enters triode, the current through M2 will begin to decrease, and the ramp on V_{out} will no longer be linear. This occurs when $V_A(t)$ reaches the greater of $V_{in} - V_T$ and $V_{const} - V_T$. From this equation, it can be concluded that V_{in} and V_{const} should each be no higher than $\frac{1}{2}V_{DD}$ + V_T to ensure that M3 and M4 remain saturated until after the M5-M6 inverter is triggered.

5. System returns to steady state

 $V_{out}(t)$ and $V_A(t)$ continue to ramp down until they reach 0.

3.1.2 Derivation of VTC delay

Using the analysis of the previous section, the VTC delay can be derived. This will be the delay from the rising clock edge of step 1 to the point where V_{out} reaches the threshold level for the M5-M6 inverter, which can be estimated as $\frac{1}{2}V_{DD}$. Ignoring the time delay of step 1, which should be small, the VTC delay will simply be the time it takes for V_{out} to ramp linearly from V_{DD} to $\frac{1}{2}V_{DD}$. The resulting expression is (using equations 3.2 and 3.4)

$$delay = R_{ramp}(V_{DD} - \frac{1}{2}V_{DD}) \tag{3.7}$$

$$=\frac{\frac{1}{2}V_{DD}C_{out}}{\frac{1}{2}KP_3(V_{in}-V_T)^2+\frac{1}{2}KP_4(V_{const}-V_T)^2}$$
(3.8)

$$= \frac{V_{DD}C_{out}}{KP_3(V_{in} - V_T)^2 + KP_4(V_{const} - V_T)^2}.$$
(3.9)

=



Figure 3.2: Circuit schematic for extracting Level 1 parameters

3.1.3 Extraction of Level 1 model parameters from BSIM4 simulations

The parameters V_T , μC_{ox} and C_{out} were extracted from BSIM4 simulations for use in the Level 1 model for hand analysis. BSIM4 models for 65nm CMOS transistors were provided by the foundry. C_{out} is simply the total DC capacitance at the V_{out} node. However, the transistor parameters V_T and μC_{ox} were not as simple to determine. The BSIM4 model is highly complex with more than a hundred different parameters, none of which can be mapped directly to V_T and μC_{ox} . In order to approximate these parameters, a simulation was performed in which a diode-connected transistor was connected to a DC current source, as shown in Fig. 3.2. The current was swept and the gate-source voltage (V_{GS}) was recorded at each point. For any two points, the Level 1 transistor current equations can be written to create a solvable system of two equations with two unknowns, as follows:

$$I_1 = \frac{1}{2}\mu C_{ox} \frac{W}{L} (V_{GS1} - V_T)^2$$
(3.10)

$$I_2 = \frac{1}{2}\mu C_{ox} \frac{W}{L} (V_{GS2} - V_T)^2.$$
(3.11)



Figure 3.3: Level 1 parameters extracted from BSIM4 simulations for different drain currents

This system of equations can be solved for the desired parameters with the following result:

$$V_T = \frac{V_{GS1} - V_{GS2}\sqrt{\frac{I_1}{I_2}}}{1 - \sqrt{\frac{I_1}{I_2}}}$$
(3.12)

$$\mu C_{ox} = \frac{2L}{W} \left(\frac{\sqrt{I_1} - \sqrt{I_2}}{V_{GS1} - V_{GS2}} \right)^2.$$
(3.13)

Using this technique on closely-spaced data points, the parameters were estimated over a range of currents. For example, to estimate the parameters at 10 μ A, data points for 9.99 and 10.01 μ A were used. The resulting curves are plotted in Fig. 3.3. It can be seen that the estimated parameters depend on the drain current, particularly for μC_{ox} .

To choose one particular value for each parameter, the drain current of the VTC starving transistor, shown in Fig. 3.4, was examined. The peak drain current varies widely depending on the VTC input voltage, so this is not terribly helpful. The best choice will necessarily be a compromise between the different drain current levels. In order to select the best values, the drain current was treated as a fitting parameter. Values of V_T and μC_{ox} for different



Figure 3.4: Simulated VTC drain current for different input levels

drain currents were plugged into the derived delay expression (equation 3.9) as the input voltage was swept. The results were then compared against the delay curve predicted by BSIM4 simulation and the closest curve was selected. The values chosen were $V_T = 0.24$ V and $\mu C_{ox} = 95 \mu/V^2$, corresponding to a drain current of 12 μ A.

3.1.4 Comparison with BSIM4 models

Simulations were performed to compare the simplified hand analysis model with the highly advanced BSIM4 models from a TSMC 65nm process. Identical device sizes were used, and Level 1 model parameters were extracted from BSIM4 simulations as described in the previous section. The design parameters are those given later in this chapter in Table 3.1. Fig 3.5 shows the resulting delay plots. The delay predicted by equation 3.9 closely matches simulation using both the Level 1 and BSIM4 models. Fig. 3.6 shows simulated waveforms for the same circuit using Level 1 and BSIM4 models.



Figure 3.5: Simulated VTC delays using Level 1 and BSIM4 models



Figure 3.6: Simulated VTC waveforms using Level 1 and BSIM4 models

3.2 Single-ended VTC Optimization

3.2.1 Maximizing Single-Ended Linearity

The series expansion technique can be used to analyze the VTC, and to optimize its design for maximum linearity. The expression for VTC delay (developed in section 3.1.2) is

$$delay = \frac{V_{DD}C_{out}}{KP_3(V_{in} - V_T)^2 + KP_4(V_{const} - V_T)^2}.$$
(3.14)

In order to use an input of $x = \sin(t)$, we can express the input as

$$V_{in} = V_B + V_{amp}x \tag{3.15}$$

where V_B is the DC bias voltage and V_{amp} is the amplitude of the sinusoidal input signal. For ease of analysis, we will put the equation into a standardized form

$$delay(x) = \frac{A}{(x+k)^2 + B^2}.$$
(3.16)

Comparing equations 3.14 and 3.16 it is apparent that

$$A = \frac{V_{DD}C_{out}}{KP_3(V_{amp})^2}$$
(3.17)

$$k = \frac{V_B - V_T}{V_{amp}} \tag{3.18}$$

$$B = \sqrt{\frac{KP_4}{KP_3}} \frac{(V_{const} - V_T)}{V_{amp}}.$$
(3.19)

The analysis begins by expressing the delay as a Taylor series up to the third harmonic:

$$delay(x) = \left[\frac{A}{k^2 + B^2}\right] + \left[\frac{-2Ak}{(k^2 + B^2)^2}\right]x + \left[\frac{-A(B^2 - 3k^2)}{(k^2 + B^2)^3}\right]x^2 + \left[\frac{4Ak(B^2 - k^2)}{(k^2 + B^2)^4}\right]x^3 \quad (3.20)$$

Labelling the terms as T_0 , T_1 , T_2 and T_3 , the form of the SINAD is identical to equation 2.43. Substituting the values in and simplifying, the expression becomes

$$SINAD = \frac{\left[T_1 + \frac{3}{4}T_3\right]^2}{\left[\frac{1}{2}T_2\right]^2 + \left[\frac{1}{4}T_3\right]^2}$$
(3.21)

$$=\frac{4k^2\left(2\,k^4+4\,k^2B^2+2\,B^4+3\,k^2-3\,B^2\right)^2}{9\,k^8+12\,k^6B^2-2\,k^4B^4-4\,k^2B^6+B^8+4\,k^6-8\,k^4B^2+4\,k^2B^4}.$$
(3.22)



Figure 3.7: Comparison of single-ended ENOB calculated analytically, by numerical simulation of analytic delay model, and CAD simulation with Level 1 transistor models

This expression is difficult to analyze. However, looking at T_2 , it is clear that setting $B^2 = 3k^2$ (or $B = \sqrt{3}k$) will eliminate the second harmonic. It can be shown that $B = \sqrt{3}k$ is, in fact, a local maxima for the SINAD expression. Therefore, this relationship between B and k results in optimal linearity. Substituting the circuit values back in gives the design equation:

$$B = \sqrt{3}k\tag{3.23}$$

$$\sqrt{\frac{KP_4}{KP_3}}(V_{const} - V_T) = \sqrt{3}(V_B - V_T)$$
(3.24)

$$KP_4(V_{const} - V_T)^2 = 3KP_3(V_B - V_T)^2.$$
(3.25)

In other words, the saturation current drawn by M4 biased with V_{const} should be triple that of M3 biased with V_B . The SINAD at the optimum point can then be simplified to

$$SINAD_{opt} = (16k^2 - 3)^2 \tag{3.26}$$

$$= \left(16\left(\frac{V_B - V_T}{V_{amp}}\right)^2 - 3\right)^2. \tag{3.27}$$

The existence of a linearity peak can be observed in Fig. 3.7. This plot shows the result of a fixed design being operated with a range of values for the bias voltage V_{const} . The curve labelled "Numerical" was produced by a numerical simulation in which the delay expression (equation 3.9) was applied to a sinusoidal input, an FFT was taken of the result and the SINAD and ENOB were calculated. The curve labelled "Simulated" is based on CAD simulation of the VTC with Level 1 transistor models, while the "Analytical" curve uses the SINAD equation (equation 3.22) directly. The parameters used for the curves are listed in table 3.1.

Two conclusions can be made from Fig. 3.7. First, comparing the "Simulated" and "Numerical" curves, it can be concluded that the derived delay expression (equation 3.9) closely predicts the location of the ENOB peak, although the absolute ENOB value is somewhat lower. This is due to the simplifications made in reducing the behaviour of the VTC circuit to a single equation. Second, comparing the "Analytic" and "Numerical" curves, it can be concluded that the direct SINAD formula (equation 3.22) closely approximates the full FFT-based SINAD calculation. The small discrepancy is due to the higher-order harmonics being discarded.

Equation 3.27 further suggests that the linearity will be improved by making V_{amp} small and using a large overdrive voltage $V_B - V_T$. However these parameters will also affect the range of the VTC. How should these values be chosen? The answer is presented in the next section.

3.2.2 Range and Absolute Delay - Single-Ended VTC

Other than linearity, the main considerations for the VTC design are the total range and the maximum absolute delay. The total range (t_{range}) is defined as the difference between the output delay for maximum input and for minimum input. This range should be as large as possible to relax the resolution requirements on the TDC, but it is limited by the clock period and pulse width being used. Since the maximum input is $V_B + V_{amp}$ and the minimum input is $V_B - V_{amp}$, the range can be expressed analytically as

$$t_{range} = \frac{V_{DD}C_{out}}{KP_3(V_B - V_{amp} - V_T)^2 + KP_4(V_{const} - V_T)^2} - \frac{V_{DD}C_{out}}{KP_3(V_B + V_{amp} - V_T)^2 + KP_4(V_{const} - V_T)^2}.$$
(3.28)

For optimal linearity, we can apply the condition of equation 3.25 and simplify to obtain

$$t_{range} = \frac{V_{DD}C_{out}}{KP_3(V_B - V_{amp} - V_T)^2 + 3KP_3(V_B - V_T)^2} - \frac{V_{DD}C_{out}}{KP_3(V_B + V_{amp} - V_T)^2 + 3KP_3(V_B - V_T)^2}$$
(3.29)

$$=\frac{4V_{DD}C_{out}(V_B-V_T)V_{amp}}{KP_3[16(V_B-V_T)^4+4(V_B-V_T)^2(V_{amp})^2+(V_{amp})^4]}$$
(3.30)

It is given that $(V_B - V_T) > V_{amp}$ (in order for transistor M3 to be above cut-off at all times), so we will assume that the $(V_B - V_T)^4$ term dominates the denominator, and simplify to

$$t_{range} = \frac{V_{DD}C_{out}V_{amp}}{4KP_3(V_B - V_T)^3}.$$
(3.31)

The maximum absolute delay (t_{max}) is the other important consideration. This is the VTC delay for the smallest possible input. Since only the falling edge is delayed, when the VTC delay increases it shrinks the pulse width of the VTC output. It must therefore be limited to a safe level for robust VTC operation. For ease of analysis, we will define the nominal absolute delay (t_{abs}) as being that produced by the common-mode output, $V_{in} = V_B$.

$$t_{abs} = \frac{V_{DD}C_{out}}{KP_3(V_B - V_T)^2 + KP_4(V_{const} - V_T)^2}.$$
(3.32)

The maximum delay, assuming the VTC is relatively linear, is then

$$t_{max} = t_{abs} + \frac{1}{2} t_{range}.$$
 (3.33)

Applying the optimum linearity condition (equation 3.25), the nominal absolute delay can be simplified to

$$t_{abs} = \frac{V_{DD}C_{out}}{4KP_3(V_B - V_T)^2}.$$
(3.34)

We can now use t_{abs} to simplify the range (equation 3.31) to

$$t_{range} = \frac{V_{amp} t_{abs}}{(V_B - V_T)}.$$
(3.35)

Equation 3.35 can be used to set the ratio of the input amplitude V_{amp} to the common-mode overdrive voltage $V_B - V_T$ in order to achieve a desired absolute delay and range.

3.2.3 Design Procedure - Single-ended VTC

The procedure to design an optimal single-ended VTC will be described in this section. The first step is to determine the fixed parameters and design specifications. First, the technology-dependent parameters μC_{ox} and V_T can be estimated from the simulator using standard techniques [64]. Using these values as a starting point, it is then recommended to use μC_{ox} as a fitting parameter to achieve the best match between the hand analysis equations and BSIM4 simulations.

The simulator can also be used to estimate C_{out} , which is highly dependent on the input capacitance of the inverter following the VTC. This inverter can be the same size as VTC devices M1 and M2, or it can be made larger. Making the inverter larger will increase C_{out} , which will increase the current needed to achieve a given slope on the output node during VTC ramping operation. This larger current will require larger widths of M3 and M4, which will improve their matching and noise performance at the expense of increased power consumption. V_{amp} should be chosen based on design specifications. The desired range and maximum absolute delay can be chosen based on clock period (T_{clk}) . For 5GS/s operation in a 65nm process, a range of $\frac{1}{8}T_{clk}$ is quite conservative, while a range of $\frac{1}{4}T_{clk}$ is fairly aggressive and is likely to result in a less robust design. The maximum absolute delay (t_{max}) is determined based on the minimum acceptable pulse width. If the clock uses a standard 50% duty cycle, the minimum pulse width will be $\frac{1}{2}T_{clk} - t_{max}$. However, the clock duty cycle can be adjusted prior to the VTC in order to allow more leeway for the absolute delay. In this case a t_{max} of up to $\frac{1}{2}T_{clk}$ may be achievable.

The unknown parameters are V_B , V_{const} , KP_3 and KP_4 . These can be chosen based on the optimization equations derived previously. First, V_B is set to achieve the desired range and maximum absolute delay, using equations 3.33 and 3.35. The result is

$$V_B = V_T + \frac{V_{amp}(t_{max} - \frac{1}{2}t_{range})}{t_{range}}$$
(3.36)

$$= V_T + V_{amp} \left(\frac{t_{max}}{t_{range}} - \frac{1}{2} \right).$$
(3.37)

Next, KP_3 can be determined using equation 3.30:

$$KP_{3} = \frac{4V_{DD}C_{out}(V_{B} - V_{T})V_{amp}}{t_{range}[16(V_{B} - V_{T})^{4} + 4(V_{B} - V_{T})^{2}(V_{amp})^{2} + (V_{amp})^{4}]}.$$
(3.38)

The width of M3 is determined from KP_3 . This leaves KP_4 and V_{const} to be determined from equation 3.25. There is an extra degree of freedom available - as long as equation 3.25 is satisfied, the choice of KP_4 and V_{const} will not affect the linearity, range or delay of the VTC. It is suggested that the width of M4 be kept fairly small to improve noise performance and reduce parasitic capacitance on the node connected to the drain of M3 and M4, although not so small that V_{const} exceeds the limit of $\frac{1}{2}V_{DD} + V_T$ as established in section 3.1.1. In the examples below the width of M4 will be fixed at 1µm. In any case, once the width is selected we can find V_{const} as follows:

$$V_{const} = V_T + \sqrt{\frac{3KP_3}{KP_4}} (V_B - V_T).$$
(3.39)

V_T	0.240V				V_B	$365 \mathrm{mV}$
μC_{ox}	$95\frac{\mu A}{V^2}$	Vamp	50mV		W_3	2.33µm
V_{DD}	1V	t_{max}	75ps		W_4	1.0µm
Cout	15fF	trange	25ps		V_{const}	$570 \mathrm{mV}$
(a)		(b)		(c)		

Table 3.1: Single VTC Design example (a) Extracted/estimated values from simulator, (b) specifications, and (c) resulting design.

This completes the single-ended VTC design. Values should then be fine-tuned in the simulator with full BSIM4 models. In particular, the value of V_{const} should be swept in order to make sure the VTC is in fact biased for peak linearity.

3.2.4 Design Example - Single-Ended VTC

To better illustrate the design procedure, an example will be detailed in this section. The necessary values were extracted or estimated from the simulator using BSIM4 models for a commercial 65nm CMOS process, as shown in Table 3.1a. The chosen design specifications are listed in Table 3.1b. Plugging the values into the equations gives values for W_3, W_4, V_B and V_{const} as tabulated in Table 3.1c.

To test the design, it was first simulated using only Level 1 transistor models with $\lambda = 0$. The simulation results, given in Table 3.2a, match the design targets very closely, although the ENOB is slightly reduced. The reduction in ENOB is due to the simplifications made in deriving the VTC SINAD. The design was also simulated over a range of different V_{const} values. The resulting curve in Fig. 3.8 shows that the design for Level 1 models is biased for peak ENOB.

Next, the same simulation was performed with full BSIM4 models. The results, listed in Table 3.2b agree closely with the design targets. However, when V_{const} is swept, the ENOB curve in Fig. 3.8 shows that the BSIM4 design is not biased at peak ENOB with the calculated voltages. This is due to imperfect estimation of V_T and μC_{ox} , as well as the limitations of the level 1 model for deep-submicron transistors. The solution is to tweak V_B



Table 3.2: Single VTC Design results using (a) Level 1 models, (b) BSIM4 models, and (c) BSIM4 models after tweaking voltages for correct range at peak ENOB



Figure 3.8: VTC ENOB curves for single-ended VTC design example

and V_{const} iteratively until the VTC is biased at peak ENOB with the target output range. In this case, the tweaked design required increasing V_B to 420mV and reducing V_{const} to 585mV (a difference of 55mV and 15mV respectively). The final results are given in Table 3.2c. The range is correct and the maximum delay is 14ps below the target specification. The ENOB curve of the tweaked design in Fig. 3.8 is very similar to the Level 1 curve, and shows that the design is now biased for peak ENOB.

3.3 Differential VTC Optimization

3.3.1 Maximizing Differential Linearity

The same procedure will now be followed to optimize the VTC for differential operation. With the same values of A, k and B (equations 3.17-3.19), the differential delay can be expressed as

$$delay_{diff} = \frac{A}{(x+k)^2 + B^2} - \frac{A}{(-x+k)^2 + B^2}$$
(3.40)

$$= -\frac{4Akx}{x^4 + 2(B^2 - k^2)x^2 + (k^2 + B^2)^2}.$$
(3.41)

The Taylor expansion of this expression up the fifth harmonic is

$$delay_{diff} = -\frac{4Ak}{\left(k^2 + B^2\right)^2}x + \frac{8Ak\left(B^2 - k^2\right)}{\left(k^2 + B^2\right)^4}x^3 - \frac{4Ak\left(3\,k^4 - 10\,k^2B^2 + 3\,B^4\right)}{\left(k^2 + B^2\right)^6}x^5 + \dots \tag{3.42}$$

The even harmonics have been eliminated as expected. Denoting the terms as T1, T3 and T5 as usual, the SINAD has the same form as equation 2.60:

$$SINAD_{diff} = \frac{\left[T_1 + \frac{3}{4}T_3 + T_5\right]^2}{\left[-\frac{1}{4}T_3 - \frac{1}{2}T_5\right]^2 + \left[\frac{1}{10}T_5\right]^2}.$$
(3.43)

When the terms are substituted in and expanded, the SINAD expression is too long to fit on the page, and an exact solution has not been found to optimize it. Instead, we will look at the SINAD using only the first and third harmonics:

$$SINAD_{diff3} = \frac{\left[T_1 + \frac{3}{4}T_3\right]^2}{\left[-\frac{1}{4}T_3\right]^2}$$
(3.44)

$$= \left[\frac{2(B^2 + k^2)^2 - 3(B^2 - k^2)^2}{B^2 - k^2}\right]^2.$$
(3.45)

This more manageable expression can clearly be maximized by setting $B^2 = k^2$. Substituting the values of B and k back in, our differential optimization condition is

$$B^2 = k^2 \tag{3.46}$$

$$KP_3(V_B - V_T)^2 = KP_4(V_{const} - V_T)^2.$$
 (3.47)

Note that this is a different B: k ratio than that derived for single-ended operation (equation 3.23).

This optimization makes the third harmonic go to zero, causing SINAD_{diff3} to become infinite. The Taylor expansion is simplified considerably when $B^2 = k^2$:

$$delay_{diff,opt} = -\frac{A}{k^3}x + \frac{A}{4k^7}x^5 + \dots$$
 (3.48)

This allows the optimum SINAD to be calculated:

$$SINAD_{diff,opt} = \frac{[T_1 + T_5]^2}{\left[-\frac{1}{2}T_5\right]^2 + \left[\frac{1}{10}T_5\right]^2}$$
(3.49)

$$=\frac{50}{13}(4k^4-1)^2\tag{3.50}$$

$$=\frac{50}{13}\left(4\left(\frac{V_B-V_T}{V_{amp}}\right)^4-1\right)^2.$$
 (3.51)

Just as it was in the single-ended analysis, the optimal linearity can be further improved by increasing the ratio of the overdrive voltage $V_B - V_T$ to the signal amplitude V_{amp} . However, for ratios above 1.5 the differential VTC can achieve higher linearity than the single-ended VTC, as shown in Fig. 3.9.

Fig. 3.10 shows the differential ENOB as V_{const} is swept in order to verify the existence of an optimal bias point. The "Numerical" curve was produced by a numerical simulation of equation 3.41, applying the equation to a sinusoidal input and performing an FFT on the result in order to calculate SINAD and ENOB. The "Analytical" curve uses the derived SINAD expression (equation 3.45) directly. The result of a Level 1 CAD simulation is also shown for comparison. The parameters used for the curves are listed in table 3.3.

Compared to the single-ended plot in Fig. 3.7, the differential curves in Fig. 3.10 show substantially higher ENOB levels due to harmonic cancellation, as discussed in section 2.4. Similar to the single-ended results, Fig. 3.10 shows that the derived delay expression (equation 3.41) is useful in predicting the optimum linearity point, although the predicted peak linearity is slightly high due to simplifications in the model. Once again, ENOBs calculated



Figure 3.9: Optimum ENOB achievable with differential and single-ended VTCs

using the direct SINAD expression (equation 3.45) closely approximate the predicted ENOBs based on full numerical simulation of the delay expression (equation 3.41) despite ignoring higher-order harmonics.

3.3.2 Range and Absolute Delay - Differential VTC

As with the single-ended VTC, we can optimize the design based on specifications for total range and maximum absolute delay. For the differential VTC, the total delay range is double that of the single-ended VTC.

$$t_{range} = \frac{2V_{DD}C_{out}}{KP_3(V_B - V_{amp} - V_T)^2 + KP_4(V_{const} - V_T)^2} - \frac{2V_{DD}C_{out}}{KP_3(V_B + V_{amp} - V_T)^2 + KP_4(V_{const} - V_T)^2}.$$
(3.52)



Figure 3.10: Comparison of differential ENOB calculated analytically, by numerical simulation of analytic delay model, and CAD simulation with Level 1 transistor models

Applying the differential optimization condition (equation 3.47) yields

$$t_{range} = \frac{2V_{DD}C_{out}}{KP_3(V_B - V_{amp} - V_T)^2 + KP_3(V_B - V_T)^2} - \frac{2V_{DD}C_{out}}{KP_3(V_B + V_{amp} - V_T)^2 + KP_3(V_B - V_T)^2}$$
(3.53)

$$=\frac{8V_{DD}C_{out}V_{amp}\left(V_B-V_T\right)}{KP_3\left[V_{amp}^{4}+4(V_B-V_T)^4\right]}.$$
(3.54)

Once again we will assume that the $4(V_B - V_T)^4$ term dominates the denominator and simplify to

$$t_{range} = \frac{2V_{DD}C_{out}V_{amp}}{KP_3 (V_B - V_T)^3}.$$
 (3.55)

To find the maximum absolute single-ended delay, we must consider one half of the differential VTC on its own. The nominal absolute delay (produced when $V_{in} = V_B$) is as follows, simplified using the optimization condition:

$$t_{abs} = \frac{V_{DD}C_{out}}{KP_3(V_B - V_T)^2 + KP_4(V_{const} - V_T)^2}$$
(3.56)

$$=\frac{V_{DD}C_{out}}{2KP_3(V_B - V_T)^2}.$$
(3.57)

We can combine equations 3.55 and 3.57 to express the range as

$$t_{range} = \frac{4V_{amp}t_{abs}}{V_B - V_T}.$$
(3.58)

To find the maximum delay, we will assume the half-VTC is relatively linear on its own. Since the range of the half-VTC is half the total VTC range, we can express the maximum absolute single-ended delay as

$$t_{max} = t_{abs} + \frac{1}{4} t_{range}.$$
(3.59)

3.3.3 Design Procedure - Differential VTC

To begin the design of a differential VTC, the needed design parameters are determined using the same considerations as the single-ended procedure (section 3.2.3). The procedure then follows that of the single-ended design but using the equations for the differential VTC.

The input bias voltage V_B is found using equations 3.58 and 3.59:

$$V_B = V_T + \frac{4V_{amp}(t_{max} - \frac{1}{4}t_{range})}{t_{range}}$$
(3.60)

$$= V_T + 4V_{amp} \left(\frac{t_{max}}{t_{range}} - \frac{1}{4}\right).$$
(3.61)

The next step is to find KP_3 using the full t_{range} expression (equation 3.54):

$$KP_3 = \frac{8V_{DD}C_{out} V_{amp} (V_B - V_T)}{t_{range} [V_{amp}^4 + 4(V_B - V_T)^4]}.$$
(3.62)

The width of M3 can then be determined from KP_3 . As in the single-ended design, there is flexibility in the choice of KP_4 and V_{const} . An optimized design requires that the parameters meet the condition in equation 3.47. If KP_4 is chosen first (for example, to set the width of M3 to 1µm), then V_{const} is found using

$$V_{const} = V_T + \sqrt{\frac{KP_3}{KP_4}}(V_B - V_T).$$
 (3.63)

Again, it should be ensured that V_{const} does not exceed the limit of $\frac{1}{2}V_{DD} + V_T$ as established in section 3.1.1. With all the parameters found, the final step is to simulate the

V_T	0.240V					V_B	490mV
μC_{ox}	$95\frac{\mu A}{V^2}$	V	amp	50mV		W_3	1.21µm
V_{DD}	1V	t	max	75ps		W_4	1.0µm
C_{out}	15fF	t_r	range	50ps		V_{const}	$515 \mathrm{mV}$
(a)			(b)		(c)		

Table 3.3: Differential VTC Design example (a) Extracted/estimated values from simulator, (b) specifications, and (c) resulting design.

$\begin{array}{c c} t_{range} & 43ps \end{array}$		(b)		(c)		
+	/0ng		+	70pg	+	50pg
t_{max}	$71 \mathrm{ps}$		t_{max}	83ps	t_{max}	64ps
ENOB	9.1		ENOB	8.2	ENOB	11.0

Table 3.4: Differential VTC Design results using (a) Level 1 models, (b) BSIM4 models, and (c) BSIM4 models after tweaking voltages for correct range at peak ENOB

design using BSIM4 models to fine tune the voltages to achieve the desired t_{range} with peak ENOB.

3.3.4 Design Example - Differential VTC

An example of a differential VTC design will now be shown. This design will use the same constraints and parameters as the single-ended VTC design example in section 3.2.4 except that the desired range (t_{range}) will be doubled to 50ps, so that each half of the differential VTC will have the same range as the single-ended VTC design (25ps). All specifications, as well as the resulting design, are given in Table. 3.3.

Table 3.4 lists the simulated results for the design. Using Level 1 models, the design is very close to the specification, despite being slightly off of the peak as shown in Fig. 3.11. Switching to BSIM4 models results in slightly off-target delay values as well as missing the peak. Adjusting V_B to 525mV and V_{const} to 540mV (a difference of 35mV and -25mV respectively) results in a delay of 50ps at peak linearity, with a maximum delay 11ps below the specification.



Figure 3.11: VTC ENOB curves for differential VTC design example

3.4 Jitter Analysis

In a time-based system, noise affects the signal by causing random variations in the time of an edge transition, also known as jitter. The uncertainty is caused by noise in the voltage or current affecting the time it takes for the output of a switching circuit, such as an inverter or VTC, to reach the threshold level of a subsequent switching circuit.

In the work being presented, there are three potential sources of noise: thermal noise from the devices, noise on the supply lines and noise from the analog input voltages.

Noise on the supply lines is primarily caused by multiple switching circuits operating on shared supply lines. Since CMOS inverter-based switching circuits draw appreciable current only when switching, there can be large fluctuation in the current drawn over the course of a clock cycle, which can cause the supply voltage to fluctuate. The effect can be reduced by distributing coupling capacitors throughout the chip layout to strongly couple the supply and ground rails, as was done in the 65nm time-based ADC described in Chapters 5 and 6.



Figure 3.12: Schematics for jitter analysis: a) Inverter, b) Current-starved inverter, and c) VTC.

The sensitive VTC is also protected from the switching noise of the much larger TDC by being on physically separate substrates with separate supply and ground connections.

This section presents an analysis of VTC jitter due to thermal noise generated by the circuit as well as noise on the input voltages. First, the accepted jitter analysis for a CMOS inverter will be reviewed. The technique will then be extended to analyze a current-starved inverter before finally presenting the full VTC jitter analysis.

3.4.1 CMOS Inverter

A thorough analysis of jitter in the CMOS inverter has been published by Abidi [65]. The analytic technique used is to calculate the power spectral density (PSD) of the current flowing into capacitor C_{out} in Fig. 3.12a. After a sharp transition from 0 to V_{DD} on Clk, M1 enters cut-off and does not contribute to the noise. M2 enters saturation, and produces a current PSD of

$$S_{i_n} = 4kT\gamma g_{m2}.\tag{3.64}$$

The parameter g_{m2} is the small-signal transconductance of M2 and γ is a process-specific parameter which is ideally 2/3 for long-channel transistors.

This current PSD is integrated as it charges the capacitor C_{out} over t_d , the time it takes for the inverter output V_{out} to reach the threshold of the next inverter. The result of this integration is the mean-squared voltage noise on the output $\langle v_n^2 \rangle$ calculated in [65] as

$$\langle v_n^2 \rangle = \frac{S_{i_n}}{2C_{out}^2} t_d. \tag{3.65}$$

For a noisy linear ramp signal, voltage noise and jitter are directly related by the slope of the ramp. The mean-squared voltage noise can be converted to mean-squared jitter (σ^2) by multiplying by the square of one over the slew-rate:

$$\sigma^2 = \langle v_n^2 \rangle \left(\frac{C_{out}}{I}\right)^2 \tag{3.66}$$

$$\sigma^2 = \frac{4kT\gamma g_m t_d}{2I^2} \tag{3.67}$$

where I is the DC drain-current of M2 which charges the capacitor. The RMS jitter in time units is simply the square root of σ^2 .

For hand-analysis, we can extend the result of [65] by using the CMOS level one model to calculate the DC current. The transition time t_d can be estimated as the time for the output voltage V_{out} to ramp from V_{DD} to $\frac{1}{2}V_{DD}$ as used previously in this chapter. Thus we have the following relations:

$$I = \frac{1}{2}\mu C_{ox} \frac{W_2}{L_2} (V_{DD} - V_T)^2$$
(3.68)

$$g_{m2} = \frac{2I}{V_{DD} - V_T}$$
(3.69)

$$t_d = \frac{C_{out} V_{DD}}{2I}.$$
(3.70)

Combining equations 3.67, 3.68 and 3.70 yields a complete hand-analysis jitter model for the CMOS inverter:

$$\sigma^{2} = \frac{8kT\gamma C_{out}V_{DD}}{(\mu C_{ox}\frac{W_{2}}{L_{2}})^{2}(V_{DD} - V_{T})^{5}}.$$
(3.71)

This model was compared against BSIM4 simulations in CMOS 65nm technology using the parameters in Table 3.5 (as estimated in section 3.2.4) and using $\gamma = 2/3$. Table 3.6
V_T	0.240V
μC_{ox}	$95\frac{\mu A}{V^2}$
V_{DD}	1V
C_{out}	15fF

Simulated Delay4.2psCalculated Delay4.1psSimulated RMS Jitter2.4fsCalculated RMS Jitter5.7fs

Table 3.5: Model parameters used for all calculations in this section





Figure 3.13: Small-signal noise models for jitter calculation; a)Starved-inverter and b)VTC (simplified)

presents the results of the comparison. In this case the calculated jitter is slightly more than double the simulated value. This is most likely due to the BSIM4 model exhibiting velocity saturation when the drain current through M2 is large, which makes the quadratic dependency of current on input voltage modelled by the equations inaccurate. However, when the starving device is added the current will be substantially reduced and the equations become much more accurate, as will be shown in the following sections.

3.4.2 Current-Starved Inverter

A similar analysis can be performed on the current-starved inverter shown in Fig. 3.12b. This analysis will be similar but not identical to that of [66]. Fig. 3.13a shows the small-signal noise model for the circuit after a rising edge on Clk. Assuming M3's output resistance is high, g_{ds3} can be removed. It can then be seen that i_{n2} , the noise current due to M2, will not reach C_{out} . The noise current entering the capacitor is therefore due only to M3. The two noise current PSD sources contributed by M3 are the device noise $4kT\gamma g_{m3}$ and the current due to voltage noise on the input $v_{in,n}^2 g_{m3}^2$, where $v_{in,n}$ is the voltage noise on the gate of M3. These two sources can be analyzed independently as they are uncorrelated.

Considering the device noise first, the situation parallels that of the CMOS inverter, except that device M3 is responsible for both the DC current charging C_{out} and the noise current. In addition, the DC overdrive voltage of M3 is $V_{in} - V_T$ rather than $V_{DD} - V_T$. The mean-squared jitter due to device noise is therefore

$$\sigma^{2} = \frac{8kT\gamma C_{out}V_{DD}}{(\mu C_{ox}\frac{W_{3}}{L_{3}})^{2}(V_{in} - V_{T})^{5}}.$$
(3.72)

Fig. 3.14 compares the jitter predicted by equation 3.72 to that obtained from a BSIM4 simulation for a range of input voltage levels. The delay predicted by equation 3.70 is also compared to the simulated value. The values agree very closely for both jitter and delay.

Next, the contribution of the input noise voltage on the gate of M3 will be analyzed. Since the noise current PSD is $v_{in,n}^2 g_{m3}^2$, the mean-squared voltage noise on C_{out} is

$$\langle v_n^2 \rangle = \frac{S_{i_n}}{2C_{out}^2} t_d \tag{3.73}$$

$$=\frac{v_{in,n}^2 g_{m3}^2}{2C_{out}^2} t_d \tag{3.74}$$

and the mean-squared jitter is

$$\sigma^2 = \langle v_n^2 \rangle \left(\frac{C_{out}}{I}\right)^2 \tag{3.75}$$

$$=\frac{4kTRC_{out}V_{DD}}{(\mu C_{ox}\frac{W_2}{L_2})(V_{DD}-V_T)^3}.$$
(3.76)

The voltage noise on the input will likely be the thermal noise of the 500hm matching resistor (fabricated on-chip for the VTC). Using realistic values for equations 3.76 and 3.72, we can compare the relative contributions of each noise source to the total output jitter. Two extremes were tested. Using a large W/L ratio of 100 and a high overdrive voltage of 500mV, the RMS jitter due to device noise is 3.4 times that due to the 500hm resistor. Using a small W/L ratio of 5 and a low overdrive voltage of 50mV, the ratio increases to 150. Therefore it is concluded that device noise dominates the output jitter for realistic



Figure 3.14: Comparison of derived current-starved inverter jitter and delay models with BSIM4 simulations

design parameters, and that the input voltage noise contribution need not be considered in the following section.

3.4.3 VTC

Finally we will analyze the jitter of the full VTC as shown in Fig. 3.12c. The simplified small-signal model is shown in Fig. 3.13b. The noise current PSD is composed of contributions from both M3 and M4:

$$S_{i_n} = 4kT\gamma g_{m3} + 4kT\gamma g_{m4} \tag{3.77}$$

$$= 4kT\gamma[KP_{3}(V_{in} - V_{T}) + KP_{4}(V_{const} - V_{T})].$$
(3.78)

As before, we can use S_{i_n} to find the mean-squared voltage noise on the output:

$$\langle v_n^2 \rangle = \frac{S_{i_n}}{2C_{out}^2} t_d. \tag{3.79}$$

The mean-squared jitter can be calculated using the standard approximation for t_d (equation 3.70):

$$\sigma^2 = \langle v_n^2 \rangle \left(\frac{C_{out}}{I}\right)^2 \tag{3.80}$$

$$=\frac{S_{i_n}}{2C_{out}^2}t_d\left(\frac{C_{out}}{I}\right)^2\tag{3.81}$$

$$=\frac{4kT\gamma[KP_3(V_{in}-V_T)+KP_4(V_{const}-V_T)]}{2C_{out}^2}\left(\frac{C_{out}V_{DD}}{2I}\right)\left(\frac{C_{out}}{I}\right)^2\tag{3.82}$$

$$=\frac{kT\gamma C_{out}V_{DD}[KP_{3}(V_{in}-V_{T})+KP_{4}(V_{const}-V_{T})]}{I^{3}}.$$
(3.83)

Finally, the DC current provided by M3 and M4 is substituted in to produce the final meansquared jitter expression:

$$\sigma^{2} = \frac{kT\gamma C_{out}V_{DD}[KP_{3}(V_{in} - V_{T}) + KP_{4}(V_{const} - V_{T})]}{\left[\frac{1}{2}KP_{3}(V_{in} - V_{T})^{2} + \frac{1}{2}KP_{4}(V_{const} - V_{T})^{2}\right]^{3}}$$
(3.84)

$$=\frac{8kT\gamma C_{out}V_{DD}[KP_3(V_{in}-V_T)+KP_4(V_{const}-V_T)]}{[KP_3(V_{in}-V_T)^2+KP_4(V_{const}-V_T)^2]^3}.$$
(3.85)

The jitter predicted by equation 3.85 is compared against BSIM4 simulation data in Fig. 3.15. Once again, excellent agreement is achieved.



Figure 3.15: Comparison of derived VTC jitter model with BSIM4 simulation

Chapter 4

First Generation VTC and TDC in 90nm CMOS

This chapter describes a time-based ADC consisting of a VTC and TDC designed in 90nm CMOS. This was the first working circuit to be fabricated as a part of this project, and the work was a stepping stone to the more mature 65nm design, on which the remaining chapters will focus. The design was a collaboration between the author, who was primarily responsible for the VTC, and fellow graduate student Ken Townsend, whose main focus was the TDC. This work led to conference papers detailing the VTC [67], TDC [68] and combined ADC [69], and was also a part of Ken Townsend's PhD thesis [66].

The ADC was designed to be operated at 5GS/s. The VTC and TDC were tested individually at 5GS/s and both performed well, as will be detailed in this chapter. However, when combined into an ADC at 5GS/s the performance was quite poor. For this reason, the circuits were tested as an ADC at 2.5GS/s. A photograph of the chip, fabricated in an STMicroelectronics 90nm process, is shown in Fig. 4.1.



Figure 4.1: 90nm VTC and TDC chip photograph



Figure 4.2: 90nm VTC top-level block diagram

4.1 Time-Interleaved VTC

The 90nm VTC was designed to achieve 5GS/s by the time-interleaving of two channels, as shown in Fig. 4.2. An on-chip clock generator takes a 50% duty cycle 5GS/s clock as input and generates two 25% duty cycle 2.5GS/s clocks that are 180° out of phase. The lower duty cycle allows each VTC channel much more time to complete each conversion before the next cycle begins. The maximum conversion time is the time between the falling edge of a pulse and the rising edge of the next pulse, so this time-interleaving system increases the maximum conversion time by a factor of 3 (from 0.5 to 1.5 periods of the full-speed clock).

The single-ended analog input V_{in} is distributed to both channels, and each channel produces a pulse train with the delay on each pulse proportional to the value of V_{in} at the time of conversion. OR gates are then used to combine the time-interleaved channel outputs into a single output, VTC_{out} . Each channel also produces an output clock signal with no input-dependent delay. These output clock signals travel through an equivalent delay path to the channel outputs so that each pulse in CLK_{out} is synchronized to its delayed copy in VTC_{out} . This system minimizes the effect of systematic channel offsets due to slight unavoidable differences in layout path lengths between the channels. It also provides immunity to jitter on the input clock.



Figure 4.3: 90nm VTC channel 1 details. Channel 2 is identical but with its own clock, reset and output signals.



Figure 4.4: 90nm VTC track-and-hold circuit with charge injection cancellation

Fig. 4.3 shows a detailed view of channel 1. The main output path contains two VTC cores in series. Each core delays the rising and falling edges of the input clock by up to 25ps, so that the final output is delayed by up to 50ps. The clock path also contains two VTC cores, the difference being that the AC input signal is not fed to these cores, only the DC bias voltage.

Since the second core in the VTC output path performs its conversion slightly after the first, a track-and-hold circuit is included to ensure that both cores are acting on the same input voltage. The track-and-hold circuit is very simple, consisting of an NMOS pass gate



Figure 4.5: 90nm VTC core schematic

and a hold capacitor. An additional NMOS capacitor is included to offset charge injection. A schematic of the track-and-hold is shown in Fig.4.4.

The transistor-level view of a single VTC core is presented in Fig. 4.5. Between the input (CLK) and the output (Out), the signal travels through 6 inverters. The second and fifth inverters are special current-starved inverters, designed to produce a linear output delay depending on the value of V_{in} . Both starved inverters act on the falling edge of their respective outputs. Due to the inversions in the chain, this means that in terms of the final output signal, the second inverter delays the falling edge and the fifth inverter delays the rising edge. Since only the rising edge delay will be measured by the TDC, this edge is more important. The falling edge is delayed only to maintain a constant pulse width.

The starved inverter consisting of M1-M6 will now be explained in detail. The circuit is slightly different from the 65nm version, which was analyzed in detail in chapter 3. M1 and M2 make up a standard CMOS inverter, and M3 acts to limit the current during the falling edge transition. M4 is connected as a CMOS capacitor, and its value can be changed using the V_{cap} DC bias signal. When a rising edge occurs on the gate of M2, charge is rapidly transferred from the inverter output (i.e. the drain of M2) through the channel of M2 onto the gate of the CMOS capacitor M4. This causes a rapid voltage drop that can be adjusted



Figure 4.6: Illustration of starved inverter output

using V_{cap} . The starved inverter then begins its ramping phase, with the slope of the ramp controlled by V_{in} . When the ramping signal reaches the threshold of the next inverter in the chain, that inverter switches. The process is illustrated in Fig. 4.6. The net effect of all of this is that the V_{cap} signal can be used to adjust the gain of the VTC; that is the amount of delay range produced for a given input range.

The starved inverter also includes small switches M5 and M6 to reset the nodes between conversions. The reset signal for M6 is generated on-chip along with the various clocks. Resetting these nodes after each clock cycle minimizes memory effects.

4.1.1 Measured Results

The 90nm VTC was mounted on a PCB board and the board traces were connected to the IC pads using gold bondwires. The circuit was first tested with DC inputs and a 5GS/s clock signal. Fig. 4.7 shows the results over an extended input range. The output delay is the time between a rising edge on the VTC output and a rising edge on the output clock signal. So a delay of 50ps means that the clock output leads the VTC output by 50ps, while a delay of -50ps indicates that the VTC output leads the clock output by 50ps. The highlighted area shows the intended 100mV input span with 50ps output range. Outside of this range the curve is highly non-linear with delay saturation for very low and very high voltages.

For this test, the DC bias voltages were tuned manually to produce a 50ps delay with



Figure 4.7: Measured 90nm VTC delay with DC input. Output delay is relative to the output clock.

100mV input span. To illustrate the effect of the tuning voltage V_{cap} , Fig. 4.8 shows the output delay curves for various V_{cap} voltages. The input voltage in the plot is in addition to the constant DC bias voltage, which was re-tuned for each curve. It was found that in order to produce a delay of 50ps, it was necessary to bias V_{cap} slightly below the ground level, at -0.1V. This was unexpected as simulations showed that the VTC could be tuned for all corners using a V_{cap} between 0.1 and 0.3V.

The VTC was also tested with AC inputs. The AC input was added to the DC bias voltage using a bias tee. Fig. 4.9 shows sample waveforms measured using a high-speed digital sampling oscilloscope and transferred to a PC through a GPIB interface.

Evaluating the VTC linearity using AC inputs presented a problem with the equipment available. The digital sampling oscilloscope being used could only process regularly repeating signals. This meant that to view the VTC output, the input frequency had to be an integral fraction of the clock frequency. It also means that waveforms being produced should be



Figure 4.8: Measured 90nm VTC delay curves with different V_{cap} tuning bias levels. Input voltage is in addition to the constant DC bias.



Figure 4.9: Measured 90nm VTC waveforms captured directly from oscilloscope



Figure 4.10: Measured 90nm 5GS/s VTC wideband linearity using SDR-based ENOB

considered averaged, since what appears to be a single rising edge was actually composed of individual samples taken across a large number of clock periods. This means that the oscilloscope was not able to measure timing jitter coming out of the VTC. So rather than the standard SINAD used to calculate ENOB, we could measure only the signal-to-distortion ratio (SDR). This is still highly useful for quantifying the VTC linearity, but it should be kept in mind that the true SINAD-based ENOB would be expected to be somewhat worse due to jitter.

The SDR-based ENOB was measured over a wide bandwidth, with and without the trackand-hold circuit. The measured results are presented in Fig.4.10 along with the corresponding simulated results. The simulated data shows the importance of the track-and-hold circuit for input frequencies above 1.5GHz. However, the fabricated VTC was found to not work beyond 300MHz input with the track-and-hold enabled. The reason is believed to be charge injection errors. With the track-and-hold circuit disabled, the VTC maintains greater than 3 effective bits up to 2GHz. The VTC output amplitude drops sharply above 1GHz, as



Figure 4.11: Measured 90nm 5GS/s VTC wideband output range with constant input amplitude $% \mathcal{A} = \mathcal{A} = \mathcal{A} + \mathcal{A}$

the measured data in Fig.4.11 shows. A working track-and-hold circuit would prevent this drop-off.

4.2 3-bit Parallel-Branch TDC

As the 90nm TDC was designed by another PhD student [66], it will not be covered in detail here. However, the architecture will be described in order to be compared with that of the 65nm TDC (Chapter 6).

A block diagram of the parallel-branch TDC is shown in Fig. 4.12. The inputs to the TDC are the CLK_{out} and VTC_{out} signals from the VTC. The variable-delay VTC_{out} signal is routed in parallel directly to the data input of 7 flip-flops. The constant-delay CLK_{out} signal is routed to the clock input of each flip-flop, but only after passing through a variable-delay block (t₁-t₇). Each variable delay block has a different delay, with t₁-t₇ increasing in increments of the minimum time-resolution, t_{δ}. The delays are individually programmable



Figure 4.12: 90nm 3-bit parallel-branch TDC block diagram

via serial input in order to tune the TDC to correct for delay variations related to process, voltage and temperature. For the 3-bit TDC with 50ps delay range (t_{max}) , the minimum resolution is

$$t_{\delta} = \frac{t_{max}}{2^N} = \frac{50ps}{2^3} = 6.25ps.$$
 (4.1)

So delay t_2 would be 6.25ps higher than t_1 and so on. The flip-flops act as phase detectors, producing an output of '0' if a rising edge occurs on the clock input before one occurs on the data input, or a '1' if the rising edge occurs first on the data input. In this way a 7-bit thermometer code output is generated. It can be noted that this circuit directly parallels the flash ADC architecture, but rather than comparing an input voltage level to a set of reference voltages it compares an input time delay to a set of reference delays.

To complete the conversion, a copy of the CLK_{out} signal delayed by t_{sync} is used as the clock input to a second set of flip-flops in order to synchronize the thermometer outputs



Figure 4.13: Measured 90nm ADC DNL and INL for DC input

(T1-T7). These outputs are then passed to a thermometer decoder (not shown) to produce a standard 3-bit binary output. Complete details on this TDC can be found in the literature [66, 68, 69].

4.3 2.5GS/s Time-Based ADC

The intended use for the 90nm VTC and TDC circuits was to connect them together as a time-based ADC. This was accomplished with each circuit on its own chip and PCB board, using coaxial cables to connect the VTC outputs to the TDC inputs. Although both circuits were separately functional at 5GS/s, it was found that when operated together as an ADC the circuits were unacceptably noisy. For this reason, the time-based ADC was tested at 2.5GS/s where the performance was much better.

The VTC was hand-tuned to produce a 50ps delay range with a 140mV input span at 2.5GS/s. The TDC tuning system was then used to fine-tune the ADC linearity. A DC



Figure 4.14: Measured 90nm ADC wideband linearity based on histogram testing

test was performed by sweeping the input voltage in 1mV increments across the full input span. Fig. 4.13 shows the measured DNL and INL for the ADC. The independent TDC delay tuning capability made it possible to achieve very good linearity, with a maximum DNL and INL of 0.02 and 0.04 LSB respectively.

The time-based ADC was tested with AC inputs by using a bias tee to combine a 100mV peak-peak sinusoidal signal with the DC input bias voltage. The 3 TDC outputs were connected to a digital-sampling oscilloscope, which was used to record single samples of the high-speed outputs. The ENOB was estimated at each frequency using a histogram test, as described in section 2.2.1. 6300 samples were taken at each frequency in order to provide 99% confidence that the estimated DNL is within 0.1 LSB of the true value, according to the formula developed in [59]. The results of the wideband sweep are plotted in Fig. 4.14. The ADC achieves an ENOB of up to 2.9 at low frequencies and an ENOB of 2.1 at the effective resolution bandwidth (ERBW) of 1300MHz. The ERBW is defined as the input frequency

Process	90 nm
Area	0.04 mm^2
Sampling Rate	$2500 \mathrm{MS/s}$
Power Dissipation	13 mW
Resolution	3 bits
ENOB (peak)	2.9 bits
ENOB @ ERBW	2.1 bits
ERBW	1300 MHz
FOM	1.1 pJ/conversion

Table 4.1: Summary of 90nm ADC measured results

range over which an ENOB of 2.5 ± 0.5 bits is achieved.

The time-based ADC draws a combined total of 12.1mW at 2.5GS/s, including the VTC output buffers but not the TDC output buffers. The calculated figure-of-merit for the ADC is 1.1pJ/conversion, using the formula

$$FOM = \frac{P}{2^{ENOB} * f_s}$$
(4.2)

where P is the power consumption, f_s is the sampling frequency and the ENOB is taken at the ERBW. Additional discussion of FOM is provided in section 7.4 These results are summarized in Table 4.1.

Chapter 5

Pseudo-Differential VTC in 65nm CMOS

This chapter will discuss the 65nm VTC which was fabricated in a general-purpose CMOS process by TSMC. The VTC was designed to operate at 5GS/s and provide a linear delay range of 50ps between differential outputs. The differential input is composed of a DC bias voltage coupled to each of the two RF differential signals through bias tees (off-chip). The range of the VTC is tunable to correct for process, voltage and temperature variations, and the tunability can also be used to operate at clock frequencies other than 5GS/s.

5.1 VTC Half-Cell

The core VTC is a pseudo-differential circuit composed of two half-cells. The schematic of the half-cell is shown in Fig. 5.1. Each half cell is fed by the same clock and bias voltages with complementary RF inputs. Examining the schematic, it can be seen that it is composed of 8 CMOS inverters, two of which have additional starving devices between the inverter and ground. A bias tee is used (off-chip) to couple the AC input signal with a DC bias voltage.

5.1.1 Duty-Cycle Adjustment Circuit

The first starved inverter, composed of M3-M5, adds a fixed delay to the falling edge of the output signal. This device is used to adjust the duty cycle of the clock to allow greater conversion time. The starving device M5 is biased with the full supply voltage, and is sized to delay the falling edge by 25ps with respect to the rising edge. For a 5GS/s clock signal, this results in a pulse width of 125ps with 75ps between pulses.

The additional pulse width allows additional time for the VTC to complete each conversion cycle. Since the VTC delays only the rising edge of the output, the pulse width of the



Figure 5.1: Full 65nm VTC half-cell schematic

output is variable. The design of the VTC is such that the average pulse width of the final output was 100ps. The total range of pulse widths was 87.5ps to 112.5ps.

5.1.2 VTC Core

The core VTC functionality is provided by the starved inverter composed of M10-M14. With the exception of M14, this is the VTC circuit that was analyzed in detail in Chapter 3. M14 is a small additional NMOS starving device with its gate connected to the output of the M15-M16 inverter that comes after the VTC inverter. This device is included only to ensure that the node at the source of M11 is fully discharged every cycle, even at the slow process corner. M14 does not play any role in the delay process because it remains off until the M15-M16 inverter has already switched.

It can be noted that the duty-cycle adjustment circuit is the second inverter in the chain and the VTC core is the fifth. Since there is an odd number of inverters in between them, they act to delay opposite edges of the final output.



Figure 5.2: Simulated VTC output delay versus input voltage for typical (TT), slow (SS) and fast (FF) process corners

5.2 Simulated Results

The VTC was designed to accept differential input voltages ranging from -100mV to 100mV and produce differential output delays between -25ps and 25ps. Fig. 5.2 shows the results of a BSIM4 simulation. The delay curve for the typical (TT) corner meets the specification exactly, and achieves a linearity of 6.1 effective bits. Running the simulation at the fast (FF) and slow (SS) corners causes the delay range to vary somewhat from the ideal 50ps. The delay range at SS is 65ps while the delay range at FF is 40ps. After adjustment, the VTC achieved 50ps delay range with an ENOB of 6.1 bits at all process corners. A Monte Carlo analysis was performed to predict the statistical distribution of delay ranges. The result, shown in Fig 5.3, is that the delay range falls between 45 and 55ps 83% of the time. Regardless of the corner, the range can be corrected by adjusting the V_{const} bias voltage value. All subsequent simulations were performed at the TT corner.



Figure 5.3: Monte Carlo analysis for VTC delay range variation

The wideband performance of the VTC was simulated using AC inputs of varying frequencies. Fig. 5.4a shows the simulation results. It should be noted that the VTC output is not quantized, so there is no theoretical limit to the ENOB in this test. Since the VTC is intended for use with a 4-bit TDC, the ENOB was recalculated after performing an ideal 4-bit time-to-digital conversion. Unlike the raw ENOB plot, this data takes into account variations in the output delay range that will cause increased clipping or quantization noise at the ADC output, as well as limiting the maximum linearity to 4 effective bits. The result, plotted in Fig. 5.4b, is the best performance that can be achieved using the VTC as part of an ADC. The ENOB is greater than 3.8 up to the Nyquist frequency of 2.5GHz and remains above 3.5 up to 7GHz.

The output delay range for full-scale inputs is also plotted against frequency in Fig. 5.5. The drop-off at high frequencies is due to the non-instantaneous sampling of the VTC. There exists a short "window" of time over which the input transistor is sensitive to the input



Figure 5.4: Simulated VTC ENOB versus input frequency at 5GS/s using a)raw output data, and b)an ideal 4-bit TDC applied to the data



Figure 5.5: Simulated VTC output delay range versus input frequency at 5GS/s

voltage. For lower frequencies, this window is small enough to be effectively instantaneous, since the input signal does not change appreciably during the sensitivity window. At input frequencies beyond 3GHz, however, the signal begins to vary enough during the window to affect the output, with the VTC effectively averaging the input during the sensitivity window. Since the VTC is intended for use in a Nyquist ADC running at 5GS/s, the maximum expected input frequency is 2.5GHz.

There are several possible definitions for the bandwidth of the VTC. Applying a standard 3dB bandwidth test to the output delay range data, the 3dB bandwidth is found to be 11GHz. Perhaps a more useful definition is the effective resolution bandwidth (ERBW) used in ADCs. This is the bandwidth over which the ENOB remains within 0.5 bits of the low frequency ENOB value (or equivalently, within 3dB of the low frequency SINAD value). For the raw VTC ENOB data of Fig. 5.4a, the result would be a bandwidth of 2.3GHz. However, this value is needlessly limited by the high ENOB of 6.1 bits at low frequencies, well beyond



Figure 5.6: Output driver schematic with pad and bondwire model

the 4-bit ENOB of the intended application. The best bandwidth metric is thus provided by the 4-bit quantized data of Fig. 5.4b, which yields a bandwidth of 7.1GHz. This is the true limit on the bandwidth that could be achieved by the VTC as part of a 4-bit ADC.

5.2.1 Output Driver

The output driver schematic used for the VTC is shown in Fig. 5.6. The figure includes a model for a 90x90 μ m pad, based on previous measurements performed within the research group, as well as an inductor to model a 1mm bondwire. The output is terminated with a 50 Ω load that models the input resistance of an oscilloscope or the TDC input. The circuit must be able to drive 5GHz pulse signals with sharp rise and fall times and sufficient amplitude to drive the TDC.

The widths of transistors M11 and M12 were chosen to provide an "on" resistance of 50Ω for each. This results in the output signal swinging between ground and $\frac{1}{2}V_{DD}$. M11 uses 16 fingers of 1.75µm each while M12 uses 4 fingers of 0.88µm each. The remaining transistors (M1-M10) were sized in order to maintain consistent rise and fall times throughout the chain while presenting a low input capacitance at the circuit input. Fig 5.7 shows the simulated 10/90 rise and fall times of the circuit for a 5GHz clock input with V_{DD} set to 1.2V. The times are between 15-20ps for all corners.



Figure 5.7: Simulated rise and fall times for output driver with 1.2V supply and 5GHz clock input

5.3 VTC Calibration

The VTC was designed to allow for two possible approaches to calibration:

Option 1: Bias the VTC at a set level determined from simulation and allow the output delay to vary with PVT variations. Include the VTC in the overall calibration loop for the ADC, using the TDC's tuning capabilities. Simulations show that biasing the VTC for a delay of 50ps at the typical process corner can result in delays of 40-65ps at the fastest and slowest process corners.

Option 2: Use a delay-locked loop (DLL) to calibrate the VTC delay precisely.

The DLL calibration system was implemented on-chip. Details and measured results will be presented in this section.

5.3.1 System Description

The VTC delay-locked loop calibration system is shown in Fig. 5.8. The VTC blocks used in the loop are the same circuits used when not calibrating. A multiplexer is used to switch



Figure 5.8: VTC Calibration System Block Diagram

the clock input of VTC_N between clk_P (the same as VTC_P , for normal operation) and clk_N (shifted by 25ps with respect to clk_P , for calibrating). These clocks must be provided from off-chip. It is expected that a mature realization of the VTC would include internally generated clocks using a phase-locked-loop, which will allow the 25ps offset to be set precisely as one-eighth of the clock period. For calibration, the analog VTC inputs Vin_P and Vin_N must be set to DC levels corresponding to the maximum and minimum values, respectively, that will be used in normal operation. So

$$Vin_{P} = V_{bias} + V_{amp}$$

$$(5.1)$$

$$Vin_{N} = V_{bias} - V_{amp}$$

$$(5.2)$$

where V_{bias} and V_{amp} are the DC bias and AC amplitude of the VTC input used in normal operation. The idea is that when the VTCs are correctly calibrated, they will provide a 25ps offset in the opposite direction of the original clock offset, resulting in aligned VTC outputs. The total differential delay will then be 50ps (+25ps with Vin_P high and Vin_N low, and -25ps with Vin_P low and Vin_N high).

To begin the calibration procedure, cal_enable is set high and a low-speed clock is used for cal_clk (both from off-chip). The flip-flop following the VTCs is identical to those used in the TDC. This flip-flop acts as a phase detector, outputting a '0' if VTC_{outP} goes high first or a '1' if VTC_{outN} goes high first. The output is low-pass filtered by a simple RC network. The filtered signal is used as an up/down signal for a 6-bit binary counter. The counter output feeds a 6-bit resistive DAC. The DAC generates V_{const} , which is the tuning signal for the VTCs. The loop is designed to force V_{const} to the value corresponding to the correct VTC delay.

The blocks can now be described in more detail. The RC network consists of a $30k\Omega$ resistor and a CMOS capacitor designed to give a 3dB bandwidth of approximately 1MHz. This restricts cal_clk to the kHz range, meaning that each calibration clock cycle will include more than 10^6 VTC output pulses. The filter output is buffered with four CMOS inverters to provide a very high gain and force the up/down signal to logic level '0' or '1'.

The 6-bit binary counter was designed using standard digital blocks. It includes an automatic reset triggered by the cal_enable signal switching from low to high. The output resets to '111011' or 59 out of a maximum value of 63. The reason for this is to ensure that V_{const} does not start too low, which under certain conditions (a fast VTC clock rate or a slow process corner) could cause the VTC output to stop switching and put the loop into a "stuck" state. The reason the counter does not reset to the maximum level is to prevent the output from wrapping around to the minimum value if the first few bits erroneously indicate an "up" condition. The counter output does not change when cal_enable is set to '0', so that there is no noise added to V_{const} during normal operation.

The 6-bit DAC uses a chain of 63 resistors, each with value 390Ω , for a static power consumption of 41μ W. A hierarchical network of PMOS pass-gates selects the tap location along the chain based on the 6-bit binary input value. The top and bottom of the resistive chain are connected directly to pads for maximum flexibility. The intended bias points are 1V for the top and 0.5V for the bottom, in order to more than cover the expected range of V_{const} values with a step size under 10mV.

5.3.2 Test Results

First, the counter and DAC operation were tested on their own by setting the VTC input so that they would never converge. Fig. 5.9 shows the resulting DAC outputs using a 100S/s calibration clock. The plot shows the expect linear ramping behaviour with some glitches around the bit transitions. The reset behaviour can also be seen, with the first ramp starting from slightly below the maximum level in each plot. The absolute voltages shown are likely inaccurate due to scaling issues in the oscilloscope used.

However, in some cases the correct counting behaviour was not observed, producing data like that shown in Fig. 5.10. Conditions that bring about the problematic behaviour include certain VTC clock frequencies and certain input delay settings between the clocks. To investigate this problem, a buffered version of the up/dn signal from the low-pass filter was probed. It was found that this signal is often 0.5V, indicating a metastable condition. It appears that the buffer designed to force the low-pass filter output to the rails fails to do this under certain conditions. Two different chips were tested and exhibited the same behaviour.

To test the actual calibration performance, two 5GS/s input clocks were set to have an offset of 25ps using a phase shifter and checked with a digital sampling oscilloscope. These clocks were connected to the VTC, along with the appropriate inputs as described in the previous section. Fig. 5.11 shows the DAC output during calibration as well as the up/dn signal. Focusing on the V_{const} signal, the calibration loop appears to behave correctly at first, converging to a constant value. The up/dn does show metastable behaviour during this period. After roughly 60 cycles of the calibration clock a large glitch occurs on V_{const} .

It was found that this pattern occurs consistently, so as a workaround a calibration routine was established which stops after 40 calibration clock cycles. Repeating this process 10 times resulted in an identical output of 815mV each time. The VTC output was then tested with calibration disabled and the delay range was found to be 49ps. The conclusion is



Figure 5.9: Measured DAC output for correct up and down counting with a 100S/s calibration clock



Figure 5.10: Measured DAC output showing incorrect counting performance



Figure 5.11: DAC output and up/dn signal during VTC Calibration. The calibration operates correctly for the first 60 clock cycles after which errors occur.

that the procedure is valid but the circuits need to be refined to avoid metastable operation. Subsequent measurements do not use the automated calibration process but instead simply use a DC source to provide V_{const} .

5.4 Measured Results

The VTC was fabricated in a TSMC 65nm process. The chip photo can be seen in Fig. 7.1 in Chapter 7. The total active area is 40 x 20 μ m for the VTC core and 150 x 75 μ m including the calibration system. The IC was mounted on a custom-designed PCB board built with Rogers 5880 substrate, shown in Fig. 5.12. Connections from PCB traces to IC pads were made via wire-bonding. The board contains filter networks for DC inputs, which include the large electrolytic capacitors seen in the photo.

Using DC differential inputs and a 5GS/s clock, output delay data for a wide input range was captured and is plotted in Fig. 5.13 along with the simulated delay values. The VTC tuning voltage V_{const} was increased to 800mV from the simulated value of 700mV in order



Figure 5.12: VTC circuit board with filter networks

to give the VTC the correct gain. It can be seen that over the designed input range of ± 100 mV, the VTC output appears linear and ranges from -25ps to 25ps, giving an output range of 50ps. For very large positive or negative voltages, the output delay saturates. When this occurs, the current-starving transistor in one half of the pseudo-differential VTC is in cut-off, while in the other half the current-starving transistor has a high overdrive voltage but the current is limited by the inverter NMOS device (M11 in Fig. 5.1).

Quantitative linearity measurement was performed by applying a sinusoidal differential input and capturing a record of sequential output delays of the VTC using a high-speed realtime oscilloscope. The data was then analyzed using an FFT to calculate the SINAD and ENOB, using the standard ADC analysis techniques. Also like an ADC, the input frequencies were chosen to ensure coherent sampling as described in section 2.2.2. For these tests, a sample length of 64 data points was used. Unlike an ADC output that consists of simple bits, the VTC output consists of delayed pulses. It was necessary to use the oscilloscope



Figure 5.13: VTC differential output delay with DC inputs at 5GS/s

to determine the instant at which the rising edge of each pulse crosses the midway point between V_{DD} and ground. An automated solution was used to perform this processing, but it added a significant delay to the data capture time. The chosen data record length of 64 points represents a compromise between data capture time and measurement accuracy.

Before testing the wideband VTC performance, it was desirable to test the theory presented in chapter 3. One of the key results of that analysis is the existence of optimum linearity peaks when the bias voltage V_{const} is swept. These peaks were confirmed to occur in the measured results from the chip, as shown in Fig. 5.14. This plot shows the linearity (in ENOB) of the VTC output as V_{const} is swept. The measured peaks are less pronounced than the theoretical curve due to noise in the real VTC, which tends to limit high-ENOB values from being measured. However, the peak occurs at approximately the same V_{const} value in both curves.

The power consumption of the VTC at various clock frequencies is plotted in Fig. 5.15.



Figure 5.14: Measured VTC ENOB peaks at 5GS/s

With a 1.0V supply, the VTC is operational up to 5.5GS/s. Increasing the supply voltage to 1.2V enables operation at clock frequencies of up to 7.5GS/s. At clock frequencies where the VTC works at both supply voltage levels, operating with the higher 1.2V supply results in roughly 1.6 times the power consumption as with the 1.0V supply. The measured power consumption is roughly double that predicted from simulation.

Next the wideband VTC test results are presented. Tests were performed at three different sampling frequencies: 1GS/s, 5GS/s and 7.5GS/s. The ENOB and delay range curves are shown in Fig. 5.16. As discussed in section 5.2, the VTC bandwidth can be defined using either the ENOB or the delay range. The bandwidths at the three sampling rates are listed in table 5.1. This table gives the 3dB bandwidth of the output delay range as well as the bandwidth over which the ENOB remains greater than 3.5 bits. The most important conclusion from this data is that the VTC has adequate bandwidth to be used in a 4-bit ADC at the design frequency of 5GS/s.



Figure 5.15: VTC power consumption versus clock frequency with 1.0V and 1.2V supply voltage

Fclk	3dB Bandwidth	Bandwidth with ENOB>3.5
1GS/s	1.6GHz	1.6GHz
5GS/s	3.5GHz	4.1GHz
$7.5 \mathrm{GS/s}$	2.7GHz	>7GHz

Table 5.1: Measured VTC bandwidth at several clock frequencies using two different bandwidth definitions



Figure 5.16: Measured VTC wideband linearity and gain at three different sampling frequencies


Figure 5.17: Measured frequency spectrum for 5GS/s VTC output with 500MHz input signal

In order to identify sources of non-linearity, the measured data for the 500MHz input frequency will be examined in detail. Fig. 5.17 shows the frequency spectrum of the output data, with the signal and several harmonics identified. Table 5.2 lists the exact power levels of each harmonic, as well as the total power of the random noise at all non-harmonic frequencies. The data shows that the largest source of non-linearity is the second harmonic frequency. Since the VTC is pseudo-differential, even harmonics should theoretically cancel out (see chapter 2). It can therefore be concluded that the two VTC half-cells are not perfectly matched, resulting in even harmonics not being fully cancelled. To produce a more linear VTC, it would be recommended to focus on matching the half-cells. The non-harmonic noise degrades the SINAD more than any harmonic distortion component other than the second harmonic. All harmonics beyond the fourth are below -30dB, making them insignificant.

The final measured VTC data is the differential random jitter, shown in Fig. 5.18. Random jitter is defined as the standard deviation in a Gaussian distribution of samples, and is

Signal	18.0dB
2^{nd} Harmonic	-11.4dB
3^{rd} Harmonic	-16.1dB
4 th Harmonic	-19.1dB
Non-Harmonic Noise	-15.3dB

Table 5.2: Measured power levels for for 5GS/s VTC output components



Figure 5.18: Measured Differential VTC Jitter

equivalent to the RMS value used in theoretical calculations and simulations. The measured jitter is approximately 0.5ps RMS regardless of clock frequency. This is more than double the simulated jitter. The simulated data does not include noise sources such as phase noise on the clock and supply noise.

The 65nm VTC measured performance is summarized in Table 5.3.

Process	65 nm
Area (core)	0.0008 mm^2
Area (w/ calibration)	0.0035 mm^2
Sampling Rate (designed)	5 GHz
Sampling Rate (maximum)	7.5 GHz
Input Range	200 mV peak-peak differential
Output Delay Range	-25 ps to +25 ps
Power Dissipation $(@5GS/s)$	4.0 mW
Input Bandwidth (ENOB>3.5)	4.1 GHz
ENOB ₀	4.4 bits
Output Jitter	0.51 ps RMS differential

Table 5.3: Summary of 65nm VTC Measured Performance

Chapter 6

4-bit Vernier Delay Line TDC in 65nm CMOS

6.1 Choice of Architecture

The 65nm 5GS/s 4-bit ADC required a new TDC design. The addition of an extra bit increases the required number of timing decisions from 7 to 15. For a flash design, each flip-flop making a timing decision requires an independent delay path. On the other hand, a VDL design uses a single delay path for all flip-flops. Reducing the necessary delays results in savings in chip area and power consumption.

Both VDL and Flash TDCs are described in section 1.4. An analysis of the savings of a VDL over a flash TDC can be performed theoretically using a reference delay of 1 unit. This corresponds to the resolution of the TDC, i.e. $\frac{t_{span}}{2^N}$ where t_{span} is the input delay span and N is the number of bits. For the analysis, we will consider a differential TDC input of the type produced by the VTC in Chapter 5. This VTC is designed so that an input of one half of the full span results in the positive and negative output pulses being aligned in time. A maximum input is represented by the positive pulse leading the negative pulse by a fixed amount $\frac{1}{2}t_{span}$, and a minimum input is represented by the positive pulse lagging the negative pulse by the same $\frac{1}{2}t_{span}$. This is in fact the optimum input encoding scheme minimizing delay in a flash TDC, as well as being simple and practical to realize physically in a differential VTC.

Figure 6.1 shows the result of the analysis. To understand how the numbers are arrived at, consider a 3-bit TDC. This TDC has 8 possible output values, labelled 0-7. 7 comparisons are needed to decide which output value corresponds to the given input. The thresholds for these comparisons are labelled T_1 through T_7 in Fig. 6.2, where Δt_{in} is the time difference between the rising edges of the positive and negative TDC inputs. For a flash TDC, T_4 can



Figure 6.1: Number of delays required for flash and vernier delay line TDCs

Figure 6.2: 3-bit TDC Thresholds

be checked with no additional delays. T_3 requires one delay of the positive input relative to the negative, while T_5 requires one delay of the negative input relative to the positive. T_2 and T_6 each require two delays (in opposite directions), while T_1 and T_7 require 3 delays each. Thus the total number of delay units needed for the 3-bit flash TDC is calculated as 2(1+2+3) = 12.

For the VDL there is a single delay path for all comparisons. The first step is to delay the positive input 3 units relative to the negative in order to check T_1 . Then each subsequent comparison requires one additional delay of the positive input relative to the negative. The total delays needed is therefore 3 + 6 = 9. Thus the advantage of using VDL over flash for a 3-bit TDC is 12/9 or 1.33. This ratio doubles for each bit added: 2.67 for 4 bits, 5.33 for 5 bits, and so on. For this reason, a VDL architecture was chosen for the 4-bit 65nm TDC.



Figure 6.3: 4-bit 65nm TDC Core Schematic

6.2 TDC Core

Fig. 6.3 shows the design of the 4-bit VDL-based TDC. The inputs In_P and In_N are timebased pulse trains that represent sampled information as the difference between the rising edges of the two signals (Δt_{in}). The TDC is designed to operate at 5GS/s with a maximum input delay range Δt_{max} of 50ps. The necessary timing resolution is

$$t_{\delta} = \frac{t_{max}}{2^N} = \frac{50\text{ps}}{2^4} = 3.125\text{ps.}$$
 (6.1)

The TDC consists of 15 stages, each consisting of a tunable delay and a flip-flop. The delays are tuned using control signals VP_i and VN_i, as will be discussed in later sections. The first delay is designed to delay the negative input relative to the positive by $7t_{\delta}$ or 21.875ps. All subsequent delays shift the inputs in the opposite direction, delaying the positive input relative to the negative input by t_{δ} . After each delay, a flip-flop makes a decision on which input's rising edge occurs first, and stores the output (e.g. Out₁, Out₂, etc.). The flip-flops use a sense-amplifier design from [70] (also used in [66]). The flip-flop outputs make up the 15-bit thermometer code representation of the TDC output.

Fig. 6.4 shows simulated TDC waveforms for a single input pulse. For this example, signals In_P and In_N arrive at the TDC input exactly aligned. After passing through the first delay stage, a delay of $-7t_{\delta}$ (or -21.875ps) is introduced between the signals A_1 and B_1 . Each subsequent delay stage (2 through 15) adds a positive t_{δ} (or 3.125ps) to the delay between



Figure 6.4: Simulated TDC Waveforms

the two signals. In the figure, only the outputs of odd-numbered delay stages are shown in order to avoid clutter.

It can be seen that the delay between the two signals sweeps through the range of $-7t_{\delta}$ to $+7t_{\delta}$ as the signals travel through the VDL. Also shown in Fig. 6.4 are the outputs of the flip-flops, labelled Out₁, Out₃ and so on. The flip-flop output will be '0' when its clock signal (B_i) arrives before its input signal (B_i), or in other words when $\Delta t_i > 0$. The output will be '1' when the data arrives before the clock, or when $\Delta t_i < 0$. In this way, the thermometer code output is built up. In the example shown, the output represents a value of 8 (Out₈, not shown, has a final value of 1).

6.3 Re-clocking Outputs

Before they can be decoded, the delays between the flip-flops must be accounted for. Since the flip-flops have unaligned signals as their clocks, the flip-flop outputs (Out_1 , Out_2 etc.) are not aligned. In order to synchronize the outputs they must be re-clocked using a single clock signal. It is important that the re-clocking signal is aligned correctly with respect to the outputs in order to avoid violating setup time or hold time requirements in the re-clocking flip-flops. The delays in the VDL are unpredictable due to process variations, so if the clock coming from off-chip were used directly its phase would likely not be aligned with the data being re-clocked. Instead, the clock is taken from the end of the VDL, so that the re-sampling clock delay will be subject to the same process variations as the data. This re-sampling signal is labelled clk_{rsmp} in Fig. 6.3. Since the rising edges of the VDL signals are modulated by the VTC, they are not uniformly spaced and are therefore unsuitable as re-clocking signal. To correct this issue, an inverting buffer is used for the re-clocking signal so that the unmodulated falling edges of the VDL signals become the rising edges of the re-clocking signal. This ensures that the rising edges of the re-clocking signal are uniformly spaced. The falling edges become non-uniformly spaced, but this does not affect circuit operation as the flip-flops are only sensitive to rising edges.

From simulations, the time delay between the first and last flip-flop outputs ranges from 200ps (FF corner) to 299ps (SS corner). Since this exceeds the 5GHz clock period of 200ps, directly re-clocking these outputs would result in errors. Instead, buffers are added to each flip-flop output to delay the outputs before re-clocking. The number of buffers after each flip-flop output is inversely proportional to the flip-flop's position in the VDL, so 15 buffers are added to flip-flop 1, 14 buffers are added to flip-flop 2, and so on down to a single buffer after flip-flop 15. This results in all outputs reaching the re-clocking circuit at approximately the same time. The buffers are simple CMOS inverter pairs. Since the timing decisions have already been made, the flip-flop outputs are not susceptible to jitter. This allows the buffers to be made using minimal-size transistors, so the buffers consume very little chip area and power.

Some misalignment still remains between the buffered outputs because the delay of the

Process Corner	Min. Hold Time (ps)	Min. Setup Time (ps)
SS	29	104
TT	28	110
\mathbf{FF}	30	113

Table 6.1: Minimum Setup and Hold Times for TDC Re-clocking Circuit



Figure 6.5: Eye Diagram of TDC Re-sampling Clock and Data

buffers is not identical to the delay of the VDL delay cells. The minimum setup time (t_S) and hold time (t_H) for the re-clocking circuit was evaluated in the simulator. The results are tabulated in Table 6.1. It can be seen that the phase relationship between the re-clocking signal and the outputs remains relatively constant over process. Fig. 6.5 is an eye diagram showing t_S and t_H from a simulation at the TT corner.

6.4 Delay Blocks

The variable delays needed for the VDL are produced by the delay block shown in Fig. 6.6. The core of the delay block is formed by M4 and M5 in a standard CMOS inverter configuration, but with voltage-controlled current-starving devices M3 and M6 limiting the maximum current through the inverter. Changing the voltages V_{gP} and V_{gN} adjusts the delay of the rising and falling edges, respectively, of the inverter. Devices M1 and M2 are comparatively small (one quarter width) devices whose purpose is to ensure that a minimum amount of



Figure 6.6: Single Delay Block Schematic

current is able to flow even if M3 and M6 enter cut-off mode. This ensures that an output will be produced regardless of the values of V_{gP} and V_{gN} , which is critical for automatic calibration (section 6.8).

Devices M7 and M8 form a standard CMOS inverter that sharpens the edge transitions and negates the inversion of the first portion of the delay block. This way, the output rising edges continue to correspond to the input rising edges, and likewise for the falling edges.

6.4.1 Delay Tuning

As mentioned above, the DC tuning voltages V_{gP} and V_{gN} can be modified to change the delay of the circuit. It is desirable to find a relationship between these voltages so that they can be treated as a single tuning parameter that causes both the rising and falling edges to be delayed by the same amount. To find this relationship, first consider the design of a standard CMOS inverter. From [71] a simple approximation for the propagation time for the rising edge (t_{pLH}) and falling edge (t_{pHL}) are

$$t_{pLH} = \frac{1}{2} \frac{V_{DD} C_L}{I_{DP}}$$
(6.2)

$$t_{pHL} = \frac{1}{2} \frac{V_{DD} C_L}{I_{DN}}$$
(6.3)

where V_{DD} is the supply voltage, C_L is the output load capacitance, and I_{DP} and I_{DN} are the drain currents of the PMOS and NMOS transistors respectively during switching. The propagation time is defined as the time for the output to rise or fall to 50% of the full scale voltage when excited by a voltage step (positive or negative). Since both transistors are charging or discharging the same load capacitance, to equalize the rising and falling edge propagation times we must make the NMOS and PMOS drain currents equal. Using simple Level 1 models for transistors in saturation, the currents are

$$I_{DN} = \frac{\mu_n C_{ox}}{2} \frac{W_N}{L} (V_{DD} - V_{TN})^2$$
 (NMOS) (6.4)

$$I_{DP} = \frac{\mu_p C_{ox}}{2} \frac{W_P}{L} (V_{DD} - |V_{TP}|)^2$$
(PMOS) (6.5)

where μ_n and μ_p are the electron and hole mobilities respectively, C_{ox} is oxide capacitance per unit area, W_N and W_P are the NMOS and PMOS gate widths, L is the gate length, and V_{TN} and V_{TP} are the absolute NMOS and PMOS threshold voltages. V_{TN} and V_{TP} are similar, so all that is left is to adjust the device widths so that the ratio $\frac{W_P}{W_N}$ is equal to $\frac{\mu_n}{\mu_p}$. This results in equal propagation delays for the rising and falling edges. Simulations show that the correct ratio for the 65nm process is 2.6.

For the voltage-controlled current-starving devices (M3 and M6 in Fig. 6.6), the current produced is

$$I_{DN} = \frac{\mu_n C_{ox}}{2} \frac{W_N}{L} (V_{gN} - V_{TN})^2$$
 (NMOS) (6.6)

$$I_{DP} = \frac{\mu_p C_{ox}}{2} \frac{W_P}{L} (V_{DD} - V_{gP} - |V_{TP}|)^2$$
(PMOS). (6.7)

Once again the ratio $\frac{W_P}{W_N}$ is set equal to $\frac{\mu_n}{\mu_p}$ in order to provide the same current drive from each device. The gate inputs V_{gP} and V_{gN} must then be adjusted together according to the

equation

$$V_{gN} = V_{DD} - V_{gP}.$$
 (6.8)

This relationship ensures that the rising and falling edges are delayed by roughly equal amounts, keeping the output pulse width from growing or shrinking relative to the input pulse width.

6.4.2 Simulated Results

Fig. 6.7a shows the delay produced by the block in Fig. 6.6 as V_{gN} and V_{gP} are swept. This is the absolute delay of the block; that is, the time between the input rising edge crossing the 50% threshold and the output rising edge crossing the same threshold. Process corners have a major impact on this delay, with an achievable range of 13-27ps at FF compared to 20-40ps at SS.

For the VDL, the more important delay characteristic is the differential delay shown in Fig. 6.7b. This is the delay between the rising edge of the delay block of Fig. 6.6 with a fixed delay block (the same circuit but with V_{gN} and V_{gP} hard-wired to VDD and VSS, respectively). In this case process variations affect only the maximum achievable delay (14ps at FF versus 20ps at SS) since a delay of 0 can always be achieved. The target delay for normal operation is t_{δ} or 3.125ps as mentioned above. However the additional tuning range enables the TDC to be used at lower data rates with increased timing resolution. Tuning resolution does decrease for higher delays however, due to the steeper slope of the delay curve for $V_{gN} < 0.6$.

Fig. 6.8 shows that the $V_{gP}=V_{DD}-V_{gN}$ relationship is effective in keeping the pulse width within ± 5 ps of the nominal 100ps over the entire tuning range for all corners. When the circuit is biased for 3.125ps differential delay the pulse widths range from 98.7ps at SF to 102.2ps at FS.



Figure 6.7: Simulated delay block (a) absolute delay and (b) differential delay



Figure 6.8: Simulated delay block pulse widths



Figure 6.9: Differential delay blocks for generating (a) t_{δ} (b) -7 t_{δ}

6.4.3 Complete Differential Delay Blocks

As described previously, the single t_{δ} delays are generated using a differential circuit with one path having a controllable variable delay and the other having a fixed delay. Fig. 6.9a shows this circuit. The lower path has its starving devices biased for maximum current, resulting in the minimum delay possible for the circuit. The upper path introduces an additional delay, controlled by the V_{gN} and V_{gP} inputs.

For the delay of $-7t_{\delta}$, the configuration of Fig. 6.9b is used. The same individual delay elements are used (that of Fig. 6.6), but with 4 elements in series. In this case, the top path exhibits minimum delay while the bottom path is delayed using V_{gN} and V_{gP}, producing a delay in the opposite direction of the t_{δ} circuit. Using 4 elements to generate 7 times the delay means the circuits must be biased further up the delay curve. The tuning resolution of this block is therefore lower due to both the factor of 4 and the increased slope as delay increases (see Fig. 6.7b). This will be discussed further in section 6.5.2. However, using 4 elements rather than 7 decreases the power consumption, noise generation and layout area for the circuit. Since $-7t_{\delta}$ (-21.875ps) is within the tuning range of the absolute delay for all corners (Fig. 6.7a), it would be possible to save much more area by using a single delay element in the bottom path and no element at all in the top path. However, this was considered too risky as any process variation beyond what the simulator predicts could make it impossible to reach the desired delay.



Figure 6.10: Monte Carlo analysis for (a) t_{δ} (b) $-7t_{\delta}$ delay blocks

The expected statistical distribution of delays was analyzed using a Monte Carlo simulation. In this test, the delay blocks were biased to give the correct differential delay at the TT process corner. The results are shown in Fig. 6.10 for both the t_{δ} (3.1ps) delay block and the $-7t_{\delta}$ (-21.9ps) delay block. All of these variations can be corrected by changing the tuning voltages of the blocks.

6.5 TDC Programming

The TDC's adjustable delays are controlled digitally and programmed through a serial connection. The variable delays are needed to set the delay range to a fixed amount regardless of process variations during chip fabrication, as well as variable temperature and supply voltage during operation. When used in an ADC, the delays can also be adjusted to tune the input/output transfer characteristic to be maximally linear, compensating for any nonlinearity in the VTC. If the TDC is being used at lower frequencies the delays can be increased significantly (since the period is longer, the maximum delay can be increased beyond the designed value of 50ps). This reduces the relative impact of jitter at the TDC input. A block diagram of the TDC programming system is shown in Fig. 6.11.



Figure 6.11: TDC Programming System

6.5.1 Serial-to-Parallel (S2P) and DAC Blocks

Serial to parallel conversion is accomplished using 240 flip-flops in series. The 240 bit code consists of 15 pairs of 8-bit values. Each value goes to a DAC. The first of each pair goes to a PMOS DAC, while the second goes to an NMOS DAC. The DACs convert the digital values to voltages, labelled VP_1 and VN_1 , VP_2 and VP_2 , and so on. Each pair of voltages goes to a TDC delay block, allowing for independent control of the rising and falling edge delays produced by the block.

The PMOS DAC is shown in Fig. 6.12a. It is a current-steering DAC consisting of 8 branches, each controlled by a switch transistor (e.g. M0). When a branch is switched on, it sends current through resistor R, producing a voltage at the output V_{out} . The branches are binary weighted with branch 0 producing the smallest current and each subsequent branch producing double the current of the branch before it. The currents can be adjusted using V_{REF} , a bias voltage from off-chip. The NMOS DAC (shown in Fig. 6.12b) works the same way but uses bias voltage V_{REFP} which is generated on-chip from V_{REF} in the DAC bias block (Fig. 6.12c).

The binary weighted DAC architecture can be designed with low power consumption and a small layout footprint. However, it is susceptible to device mismatch due to process and





(b)



Figure 6.12: TDC DACs (a) PMOS-type, (b) NMOS-type, and (c) bias block

temperature variation. These variations can result in unequal step sizes and non-monotonic behaviour. To improve matching, devices use gate lengths considerably larger than minimum. The gate widths were tweaked using simulations with extracted parasitics to make the step sizes as equal as possible.

The PMOS DAC produces outputs of 0-351 mV at the slow corner (SS) and 5-883mV at the fast corner (FF). This makes it well suited to biasing the PMOS-type starving device in a current-starved inverter as used in the TDC. The NMOS DAC produces outputs of 649-1000mV at SS and 81-994mV at FF, making it well-suited to biasing the NMOS-type starving device in a current-starved inverter.

6.5.2 Tuning Resolution

The DAC resolution of 8 bits was chosen in order to reach a target step resolution of 0.1ps in the variable delay blocks. Simulations with extracted parasitics were performed with an NMOS and a PMOS DAC driving a differential delay block. The resulting delay curves are plotted in Fig. 6.13a. At TT and FF the maximum tunable delay is 16ps, while at SS it is limited to 5.4ps. The primary reason for the reduced delay range at SS is the DAC's more limited output voltage range at this corner. However there is still ample margin for producing the desired 3.125ps delay at all corners.

This data can be examined more closely to evaluate the tuning resolution around the target delay at different corners. To do this, 10 consecutive samples surrounding the target delay at each corner were analyzed. The average, maximum and minimum delay steps for these samples are presented in Fig. 6.13b. At the TT and FF corners the minimum step is negative, indicating slightly non-monotonic DAC behaviour. This is not a significant problem for tuning the TDC; in fact, it is much more important that there are no large jumps which could make it impossible to tune 0.1ps accuracy. The average step sizes were computed after discarding any negative steps (there was only one negative step in each of the TT and SS data sets). The average resolution is 0.03ps at SS, 0.06ps at TT and 0.10ps at FF.



Figure 6.13: Simulated t_{δ} delay block with DAC control: (a) full sweep and (b) step resolution

The $-7t_{\delta}$ differential delay block was simulated with DACs in the same fashion. The delay curves are plotted in Fig. 6.14a. At TT and FF the delay can be tuned from 0 to -67ps and -65ps respectively. At SS the maximum delay is -25ps, just enough for the nominal delay of 21.875ps. Fig. 6.14b shows the tuning resolution for this circuit. As expected, the tuning steps are larger than those of the t_{δ} circuit, with maximum absolute delay steps of 0.34ps, 0.55ps and 0.75ps at SS, TT and FF. Such large steps are not ideal, however since it will only affect a single output code the impact on overall converter linearity will be minimal.

6.6 Output Decoding

After the VDL output has been re-clocked, the final step in the TDC is the thermometer decoder. This block converts the 15-bit thermometer-coded output of the VDL (Out₁ through Out_{15}) to a 4-bit binary output (B₀ through B₃). Normally it can be expected that the thermometer code will consist of ones below the input level and zeros above it. However due to noise, jitter or metastability, real systems sometimes exhibit what are known as bubble errors. These are single bit errors resulting in an extra transition from one to zero and back to one, resembling a bubble when visualizing the output as an actual thermometer. In addition to correctly decoding ideal thermometer-coded outputs, it is desirable for the



Figure 6.14: Simulated $7t_{\delta}$ delay block with DAC control: (a) full sweep and (b) step resolution

thermometer decoder to be as resilient as possible to bubble errors.

Various architectures can be used for the decoder. At lower throughput rates, read-onlymemory (ROM) based decoders are commonly used [72, 73]. These have the advantage of being highly resilient against bubble errors, as a large number of possible thermometer codes can be handled via a lookup table. However the ROMs are highly complex, consuming chip area and drawing significant power. Furthermore, complex ROMs are unlikely to be fast enough for 5GS/s operation.

The fat tree architecture [74] can also be used. This decoder consists of two stages. The first stage detects transitions between neighbouring pairs of bits, producing a one-out-of-N code. The second stage uses trees of OR gates to produce the binary outputs. Fat tree decoders are faster than ROMs. They also offer uniform loading of the thermometer bits and a maximum fanout of 2. The delay path from the inputs to each output bit is equal. The disadvantage is that it has no resiliency against bubble errors.

Another option is to use Karnaugh maps to directly map the thermometer-coded inputs to the binary outputs. This approach has the advantage of minimizing the amount of logic required, with comparable speed to the fat tree architecture. This can be referred to as a minimal-logic decoder (MLD). The minimized sum-of-products expression for each of the 4 output bits are

$$B3 = T8 \tag{6.9}$$

$$B2 = T12 + T4 \bullet \overline{T8} \tag{6.10}$$

$$B1 = T14 + T10 \bullet \overline{T12} + T6 \bullet \overline{T8} + T2 \bullet \overline{T4}$$

$$(6.11)$$

$$B0 = T15 + T13 \bullet \overline{T14} + T11 \bullet \overline{T12} + T9 \bullet \overline{T10} + T7 \bullet \overline{T8} + T5 \bullet \overline{T6} + T3 \bullet \overline{T4} + T1 \bullet \overline{T2}$$
(6.12)

where B3 is the MSB and T1-T15 are the thermometer-coded bits. Several observations can be made about this logic. First, it has a maximum fanout of 3 (for $\overline{T8}$). It requires a total of 11 AND gates and 11 OR gates (assuming only 2-input gates are used), for a total of 22. In comparison, the fat tree requires 14 AND gates and 26 OR gates, for a total of 40. However, in the true MLD approach there are no logic gates between the inputs and B3, while there are 4 gates between the inputs and B0. This will result in a significant timing mismatch at 5GS/s. As a result, dummy logic gates must be inserted so that each output has 4 gates between the input and output. This more practical MLD requires 15 AND gates and 17 OR gates, for a total of 32.

Comparing the MLD and fat tree decoder, the MLD has a small advantage in number of logic gates. The fat tree has more uniform loading and a lower maximum fanout. Both approaches require 4 gates between input and output so their speeds will be comparable. So far neither decoder seems to have a significant advantage overall. The remaining factor is resiliency to bubble errors. In order to compare the two approaches a simulation was performed. Errors were introduced into thermometer codes by flipping one bit in the code. This was repeated for all 15 bits of each of the 16 possible correct codes, for a total of 240 iterations. Each erroneous code was converted using both fat tree logic and MLD logic to produce output binary codes. The output codes were then compared to the input codes prior to errors being introduced to evaluate how the two thermometer decoders handle erroneous codes.



Figure 6.15: Average output values for fat-tree and minimal-logic thermometer decoders with single bit errors

Fig. 6.15 shows the average value of the output codes produced for each ideal input code. The best possible result would be a direct 1-to-1 mapping, as shown by the "Ideal" line on the graph. It's clear that the MLD curve tracks the ideal line much more closely than the fat tree curve. For a quantitative analysis, the standard deviations (calculated using mean-squared error) are 4.5 bits for the fat tree decoder and 2.4 bits for the MLD.

6.6.1 On-Chip Implementation

The MLD was implemented using CMOS logic, as shown in Fig. 6.16. However, the OR gate performed poorly at 5GS/s. The reason for this was traced to the uppermost PMOS transistor. Due to the lower mobility of holes as compared to electrons, PMOS device M1 lacks the drive current to quickly pull the central node up to V_{DD} through the resistance of M2. Increasing the width of the device increased the drive current but also increased the parasitic capacitance at the gate and drain of M1, resulting in no net increase in switching speed. The AND gate does not suffer from this difficulty because its PMOS transistors are connected in parallel and thus each transistor can switch quickly enough on its own without the resistance of the other being a problem. The NMOS transistors connected in series have



Figure 6.16: Schematics for (a) AND gate and (b) OR gate (not used on chip)

enough drive current to switch the output at an acceptable rate.

Because of this issue, the decoder was designed using only AND logic. The conversion from OR gates to AND gates was done using boolean algebra so that the overall logic is identical. The decoder design using ANDs and NANDs (which consist of an AND gate followed by an inverter) is shown in Fig. 6.17. No additional logic is needed for the inverted thermometer outputs since the flip-flops used in the TDC core produce differential outputs. It can be seen that $\overline{18}$ has a fanout of 3, $\overline{14}$ and $\overline{112}$ each have fan-outs of 2, and every other thermometer output drives a single logic gate input. At the transistor level, each logic gate input connects to one NMOS and one PMOS device. Where dummy logic gates are used to add delay, a logical '1' (V_{DD} at circuit level) on one output of an AND gate allows the gate's other input to pass directly to the output.

6.7 Layout and Simulated Power Consumption

The completed layout for the TDC is shown in Fig. 6.18. The total active chip area is 280 x 290 μ m. The output drivers, which re-used the design of the 65nm VTC drivers, are not shown.

The simulated power consumption of the full TDC running at 5GS/s, excluding output buffers, is 25mW at the typical corner with a 1V supply. This decreases slightly to 24mW



Figure 6.17: Final Minimum-Logic Decoder Design



Figure 6.18: TDC Layout

at the SS corner and increases to 26mW at the FF corner. Fig. 6.19 shows a breakdown of the power consumption by block. The power consumption is dominated by the main tunable delay path, the flip-flops making decisions, and the re-sampling that takes place prior to the thermometer decoder.

6.8 TDC Calibration Algorithm

An automatic calibration algorithm has been developed for the TDC. While the delay tuning circuits themselves were fabricated on-chip, control logic to run the tuning algorithm was not. It would be possible to integrate this logic on-chip using a finite state machine. Instead, MATLAB code running on a PC was used to calibrate the chip using the algorithm. The PC uses a GPIB interface to connect to an oscilloscope in order to read the TDC outputs, and a USB interface to a serial controller to program the DACs.

The basis of the calibration technique is to apply a periodic input of time-varying pulses to the TDC and recording histograms. After each histogram is completed, the measured histogram is compared against the ideal histogram that should be produced by an ideal TDC with the specified input. Based on this comparison, the digital tuning values for the



Figure 6.19: Breakdown of TDC power consumption (simulated data)

DACs that control the TDC delays are adjusted. This cycle is repeated multiple times until the measured histogram is evaluated to be sufficiently close to the ideal histogram.

As it is difficult to generate time-based PWM signals with sufficient resolution at gigasample/second rates, the input would normally come from a VTC which has a voltage-based input applied to it. This is actually a major advantage when using the VTC/TDC combination as an ADC. The reason is that non-linearity, gain error and offset error in the VTC can be calibrated out by the TDC calibration. This makes it practical to save complexity and power by not calibrating the VTC at all. As long as the PVT variation is within the limits of the TDC tuning system, the TDC calibration algorithm will be able to calibrate the VTC-TDC system correctly.

6.8.1 Possible Inputs and Expected Histograms

There are several choices for the periodic input used in the algorithm. The simplest choice is a ramp signal. For a ramp input exciting the full span of the TDC, the ideal histogram is uniform. A low-speed ramp generator could even be integrated on the chip for completely self-contained operation. For an N-bit TDC, the distribution of a histogram with S total samples will be

$$H(i) = \frac{S}{2^N}, \qquad i = 1, ..., 2^N.$$
(6.13)

For high speed inputs, the most practical input waveform is sinusoidal. Any other waveform will have frequency content higher than the fundamental frequency, which for GHz signals will be distorted by low-pass filtering effects in the system. A sinusoidal signal with an amplitude exercising the full span of the TDC produces a histogram that can be computed as follows, based on [60]. First, defining the scale of the input as being between 0 and 1, the bit transitions are located at

$$b_i = \frac{i}{2^N}, \qquad i = 1, ..., 2^N.$$
 (6.14)

The cumulative histogram of the ideal quantized sinusoid is then

$$CH(i) = \frac{N}{\pi} \cos^{-1}(\frac{1}{2} - b_i), \qquad i = 1, ..., 2^N.$$
 (6.15)

Finally, the histogram itself is calculated simply as

$$H(1) = CH(1) (6.16)$$

$$H(i) = CH(i) - CH(i-1), \qquad i = 2, ..., 2^{N}.$$
(6.17)

In certain applications, it is possible to use this system to perform continuous background calibration. The requirements to do so are that the input signal in the application can be expected to have a statistically predictable histogram which exercises the full span of the converter. In general, however, normal operation must be paused while the calibration is performed (this includes the SKA radio astronomy project).



Figure 6.20: TDC Automatic Calibration Algorithm

6.8.2 Algorithm Details

A flowchart representation of the algorithm is shown in Fig. 6.20. The first step in the algorithm is to record a number of samples $(N_{samples})$ to produce a histogram. The input frequency must be such that the sampling clock does not form a repeating pattern with the input waveform during the recording process. The choice of $N_{samples}$ is a trade-off, with smaller numbers favouring decreased hardware complexity and quicker completion time, but larger numbers increasing the statistical accuracy of the histogram in the presence of noise. The formula given in [59] as

$$N_{samples} \ge \frac{Z_{\alpha/2}^2 \pi 2^{n-1}}{\beta^2} \tag{6.18}$$

can be used to calculate the minimum number of samples needed to achieve a confidence level of $100(1-\alpha)$ in estimating the DNL to within β for an n-bit converter. Z is the standard normal distribution. For example, in order to achieve 95% confidence of a DNL within 0.1 LSB of the true value for a 4-bit converter, a minimum of 9655 samples are needed.

As each sample is taken, one of the counters $N_0, N_1, ..., N_{15}$ is incremented depending on the output code. After all the samples have been taken these counters contain the full histogram. The ENOB is then calculated from the histogram using the method described in section 2.2.1. If the ENOB is greater than the desired value (ENOB_{min}) the calibration is finished. If not, the algorithm proceeds to tune the TDC.

The tuning values are adjusted based on discrepancies between the measured histogram counts $(N_0 - N_{15})$ and the calculated ideal values for the given input signal $(H_0 - H_{15})$. The term Δ is the desired accuracy of the tuning. For instance, if $\Delta = 0.1$ then the algorithm will attempt to shift all histogram bins to within 10% of their ideal values. A smaller Δ may allow higher ENOB values to be reached, but also increases the likelihood of instability in the calibration loop.

The tuning steps requires an understanding of the TDC tuning system, as described in section 6.4. There are a total of 15 tuning values that can be adjusted $(DAC_0 \text{ through})$

 DAC_{14}). DAC_0 determines the position of the first code transition (between a '0' and a '1') while ($DAC_1 - DAC_{14}$ determine the width of codes '1' through '14'. This leaves code '15' without any direct adjustment. The width of this bin is determined by the cumulative effect of all other tuning values. Thus the main challenge in developing a tuning algorithm is to determine what effect the measured histogram value for code '15' (N_{15}) should have on the tuning values. The chosen solution is to shift all other target ranges slightly up or down depending on whether N_{15} needs to be decreased or increased. So instead of the target range being $\pm \Delta$ around the ideal value, the new target range will be from Δ_p above the ideal value to Δ_n below it. Changing the values of Δ_p and Δ_n slides the target range up or down around the ideal value. The result of this will be a slight shift in the size of all other bins. If the other bins are made slightly smaller, for example, the result will be more samples falling into bin 15, so N_{15} will increase. So the first tuning step is to define the values of Δ_p and Δ_n based on how close N_{15} is to the ideal value H_{15} , and in which direction, in order to set the target range.

The second tuning step adjusts DAC_0 based on N_0 , the number of '0' codes in the histogram. If N_0 is below the target range, DAC_0 is decreased to shift the first transition upwards. If N_0 is above the target range DAC_0 is increased, and if N_0 is within the target range DAC_0 is not changed.

The last tuning step is to adjust each of $DAC_1 - DAC_{14}$ based on $N_1 - N_{14}$ respectively. This works the same way as adjusting DAC_0 except that the adjustment is made in the opposite direction, due to the architectural differences between the initial delay block and the remaining delay blocks (see section 6.4). With the tuning complete, the algorithm returns to the start to record a new histogram with the updated settings.

Fig. 6.21 shows a simulation of the algorithm for a 4-bit TDC. In this case, the "ENOB $> \text{ENOB}_{min}$ " step is omitted so that the algorithm continues to run even after achieving a good ENOB. The simulation uses a sinusoidal input signal as well as some random noise.



Figure 6.21: Simulated TDC Calibration Performance

The simulation starts with random tuning values, resulting in a low ENOB. As the TDC is tuned, the ENOB converges to remain above 3.97 for the final 100 cycles.

6.9 Measured Results

The TDC was fabricated in a TSMC 65nm general purpose process. The chip photo can be seen in Fig. 7.1 in Chapter 7. After fabrication, the IC was mounted on a custom-designed PCB board made with Rogers 5880 substrate, as shown in Fig. 6.22. Connections from IC pads to PCB traces were made via manual wire-bonding with gold wire. As with the VTC board, the TDC board includes filtering for the DC inputs, using surface mount components as well as large electrolytic capacitors. The TDC was confirmed to be operational, as was the serial delay programming system.

To measure the delay tuning characteristics, a test was performed in which one delay was swept over the full range while the others were held constant. The serial programming system was used to program the delays at each step. Using a digital pattern generator to provide the inputs, the input delay was swept to find the delay range between output code



Figure 6.22: TDC circuit board with filter networks

transitions. In this case, the bin corresponding to output code 13 was tested. The result of the sweep is shown in Fig. 6.23a. The tuning range could be adjusted between 0 and 26ps, with an average increase of 0.15ps per tuning step. Fig. 6.23b shows the TDC transfer curves for 3 particular values of delay 13. It can be seen that increasing the size of bin 13 pushes higher bins forward while leaving lower bins unchanged.

The TDC was first tested using a coarse 2-point tuning method, in which all of the 3.125ps delay blocks were adjusted together to provide the correct delay range, without necessarily optimizing the linearity. This was done over a range of sampling rates up to 10GS/s. As Fig. 6.24 shows, the TDC is fully operational up to 9GS/s, with an ENOB (calculated via histograms) of at least 3 bits. It should be stressed that these ENOB values can almost certainly be improved by fine tuning the delays individually, such as with the automated tuning method described in section 6.8. However, it is interesting to note that at 1GS/s the linearity of the converter is essentially perfect, with an ENOB of 3.94 bits. At 2.5GS/s the ENOB is 3.68 bits, good enough to be used without further tuning. At 3GS/s and beyond



Figure 6.23: TDC measurements: (a) Delay range for a single delay block, and (b) transfer curves with a single delay set to various values



Figure 6.24: Measured TDC ENOB using histograms with simple 2-pt tuning method



Figure 6.25: Measured TDC transfer curve using histogram data

the ENOB drops below 3.5 bits. These results indicate that for lower sampling rates (and corresponding longer delay times) the matching between delay blocks is adequate, but at sampling rates above 2.5GS/s fine tuning is needed to compensate for mismatch between the blocks.

Fig. 6.25 shows the transfer curve of the TDC measured with a 1GS/s sampling rate with a delay range of 50ps. Automatic calibration was used to tune the TDC. The plot is made up of histogram data, with darker points representing more hits at a given location and lighter points representing fewer hits. This plotting style allows the full curve to be seen, including noise. It can be seen that increased noise appears at code transitions where multiple bits are switching at the same time, particularly the 7/8 transition (all 4 bits switching) and the 3/4 and 11/12 transitions (3 bits switching). The other noise-related effect that can be seen in the figure is the significant overlap between each code. This is the result of jitter bumping the output up or down to the next adjacent code.

The measured power consumption of the TDC is plotted in Fig. 6.26. The measurement



Figure 6.26: TDC Power Consumption versus clock frequency

closely agrees with the simulated values.

The performance of the TDC is summarized in Table 6.2.

Process	65 nm
Area	0.08 mm^2
Clock Frequency (designed)	5 GHz
Clock Frequency (maximum)	9 GHz
Input Delay Range	-25 ps to +25 ps
Output Resolution	4 bits
Power ($@$ 5GS/s)	24 mW

Table 6.2: Summary of 65nm TDC Measured Performance

Chapter 7

65nm ADC Measurements

The 65nm VTC and TDC were fabricated on a single chip. A photograph is shown in Fig. 7.1. The chip measures 1.5mm x 0.7mm. The active areas used for the VTC and TDC are 40x20µm and 280x290µm respectively. Due to pad sharing the VTC and TDC cannot both be connected on one chip.

As discussed previously, the VTC and TDC can be mounted on separate circuit boards and connected with coaxial cables to form a complete ADC. Due to a design error with the TDC input buffers, the TDC input is inverted. This error makes the VTC and TDC incompatible since the VTC adds delay to the rising edge of pulses, but the TDC measures the error on the falling edge. To overcome this problem, passive inverting transformers were obtained from Picosecond Pulse Labs (model #5100). These devices perform an extra inversion on the VTC output so that the falling pulse edges carry the signal information going into the TDC. The fix works, although it is possible that the transformers introduce distortion into the signal. The transformers have an insertion loss of 1.5dB, meaning that the signal swing of the VTC output is compressed, which can be expected to degrade the signal-to-noise ratio (SNR).



Figure 7.1: 65nm VTC and TDC chip photo


Figure 7.2: Two-board test setup for ADC measurements

The setup used for testing the ADC is shown in Fig. 7.2. The differential input is connected to the VTC board through bias tees, biased with the common-mode voltage V_{bias} . The VTC calibration functionality is disabled for simplicity and because it is not needed. The VTC tuning voltage V_{const} is set to a level known to provide roughly correct output range and is not changed, since the TDC calibration system will adapt to the signal level it receives. The two VTC outputs pass through the inverting transformers, and bias tees are used to set the DC level to $\frac{1}{2}V_{\text{DD}}$. The TDC uses serial clock and data connections for programming, which are controlled by MATLAB software to implement the automatic calibration functionality. V_{ref} is the bias voltage for the TDC DACs. Power and ground connections are not shown. Separate power supplies are used for the VTC, VTC output drivers, TDC and TDC output drivers.

7.1 Automatic Calibration

The ADC was calibrated using the TDC calibration algorithm (section 6.8) with a 100mV peak-peak differential sinusoidal input prior to all tests. The results of the calibration process can be seen in action in Fig. 7.3. This plot shows the ADC ENOB, calculated using both



Figure 7.3: Measured ADC ENOB during automatic calibration process

the histogram method and the FFT method, after each cycle of the calibration process. The TDC delays are all initially set to minimum values, resulting in nonsensical ENOB values of less than one. As the process progresses, the calibration algorithm adjusts the delays to improve the ENOB to upwards of 3.5 effective bits.

7.2 DC Input Characteristics

To test the ADC with DC inputs, the AC input source was disabled and the bias voltages for the positive and negative VTC inputs were controlled separately in differential fashion. In other words, if one input is increased by 1mV above the common-mode bias voltage, the other will be decreased by the same amount below the common-mode voltage. Sweeping the differential input voltage in 1mV increments produced the staircase plot shown in Fig. 7.4. The plot appears correct with the exception of a small number of glitches. These are produced



Figure 7.4: Measured ADC output for DC differential input)

near bit transitions where multiple ADC output bits are switching. When one bit switches slightly before the others, the result is an erroneous output. The worst case for this is the transition between the '7' (0111) and '8' (1000) output codes, where all four bits are switching. In the figure, glitches can also be observed between the '11' (1011) and '12' (1100) outputs and between the '13' (1101) and '14' (1110) outputs.

The output codes greater than '8' exhibit more glitches than codes '7' and below. This is most likely the result of jitter accumulating in the Vernier Delay Line (VDL) for the TDC. For lower codes, the delayed pulses only pass through the first few delay cells in the VDL before reaching their comparator. For higher codes, the pulse signals must pass through an increased number of delay cells, each of which adds a small amount of jitter to the signal, before arriving at the correct comparator. For instance the '1' output code only passes through a single delay cell, while the '15' output code passes through 15 delay cells before reaching its comparator. Full details on the VDL architecture can be found in section 6.2.



Figure 7.5: Measured ADC DNL and INL for DC input

Using the same histogram data, the DC ENOB is calculated as 3.76 effective bits. The DNL and INL are plotted in Fig. 7.5. The maximum DNL and INL are 0.34 and 0.38 LSB respectively.

7.3 Wideband Input Characteristics

The ADC was tested with wideband inputs by sending AC differential sinusoidal voltage signals to the VTC inputs. Standard ADC test procedures were followed [75] including the use of coherent sampling to ensure good FFT results. Results are shown in Fig. 7.6 for 1GS/s, 2.5GS/s, 5GS/s and 6GS/s sampling frequencies. The supply voltage used is 1V except for the 6GS/s test, which used an increased supply voltage of 1.2V. The system was calibrated using the TDC auto-calibration system prior to each test.

A summary of the ADC performance at each sampling frequency is given in table 7.1. The effective resolution bandwidth (ERBW) is calculated at each sampling rate as the range



Figure 7.6: Measured ADC wideband linearity for 1GS/s, 2.5GS/s, 5GS/s and 6GS/s sampling frequencies

Rate (GS/s)	$ENOB_0$	ERBW (MHz)	Power (mW)
1	3.7	3000	7
2.5	3.4	3500	17
5	3.2	2100	35
6	3.3	1500	52

Table 7.1: Measured ADC performance at different sampling frequencies

over which the ENOB is within 0.5 bits of the low frequency value.

7.3.1 Analysis of 5GS/s Wideband Performance

Examining the 5GS/s ENOB plot in detail, it can be seen that there are dips at certain input frequencies within the ERBW, most noticeable around 300MHz and 900MHz. The cause of these dips will be investigated in this section.

Referring back to Chapter 5, the VTC output amplitude is known to vary with frequency. Incorrect VTC output amplitude will increase noise after TDC conversion - clipping noise if the amplitude is too large or quantization noise if it's too small. An experiment was performed to determine to what extent this effect limits ADC performance. With the sampling rate set to 5GS/s, the wideband input frequency sweep was repeated. However this time, the amplitude of the VTC input signal was varied at each input frequency in order to produce the highest possible ENOB. The result of the experiment is plotted in Fig. 7.7. This experiment demonstrates that some dips, such as the one around 300MHz, are simply caused by incorrect VTC output amplitude. These dips could be corrected by including a pre-distortion filter prior to the VTC input. However, the most serious dip within the ERBW, around 900MHz, is not helped at all by amplitude correction.

In order to investigate the performance with 900MHz input, the ADC data for this frequency was analyzed and compared to the data for an input of 500MHz. The ADC achieves a good ENOB of 3.5 at 500MHz, but at 900MHz it dips to 2.7. First, the FFT plots for both input frequencies are presented in Fig. 7.8. The obvious difference between the two is that the 900MHz plot has multiple spurs jutting out of the noise floor. These are caused



Figure 7.7: Measured performance of 5GS/s ADC with amplitude optimized at each frequency, as opposed to using a constant amplitude for all frequencies

by harmonics of the input signal. Possible sources of harmonic distortion are quantization and clipping of a sinusoidal input and a non-linear ADC transfer function. Using the same FFT data, the first 20 harmonics are plotted for each input frequency in Fig. 7.9. Harmonic 1 is the signal itself. It can be seen that harmonics 2-5 in the 900MHz plot are significantly larger than those in the 500MHz plot. With these harmonics removed from the 900MHz data, mathematical analysis shows that the resulting ENOB would be 3.7, close to ideal for a 4-bit ADC. The reason for these harmonics is not well understood but may be the result of ringing on the supply rails at that particular input frequency. Since the VTC on its own exhibits no such problems, the ringing most likely occurs in the TDC where it could cause issues with the delay generation circuits and the flip-flops used for timing detection.



Figure 7.8: Measured ADC output frequency spectrum with 500MHz and 900MHz inputs



Figure 7.9: Measured ADC output harmonics for 500MHz and 900MHz input signals

7.4 Figures of Merit

A wide variety of figures of merit (FOMs) for ADCs have been proposed [76]. Of these, the most common and influential is known as the ISSCC (or Walden) FOM [32, 77]. The definition is

$$FOM_{ISSCC} = \frac{P}{2^{ENOB} * f_s}$$

where P is the power consumption and f_s is the sampling frequency. The equation results in a value for the energy per bit conversion, in which lower values indicate better efficiency. This definition itself is not completely sufficient, as there is disagreement about which frequency the ENOB should be measured at. Walden's original ADC survey specified the use of low-frequency ENOB, although it included only converters which have an ERBW of at least one-quarter of the sampling frequency. Today the most influential ADC survey¹ is maintained by Murmann [32]. Murmann also uses the ISSCC FOM, but specifies that ENOB values should be taken near $\frac{1}{2}f_s$, or in the case of bandwidth-limited ADCs at the "highest reasonable/useable" input frequency. In this thesis Murmann's definition of the ISSCC FOM will be used, with the added specification that the "highest reasonable/useable" input frequency is the ERBW. It should be noted that for bandwidth-limited ADCs, Walden's definition will yield FOM values that appear better by a factor of 2^{0.5} or 1.4 compared to Murmann's.

The other common definition is the ITRS FOM, as published in the International Technology Roadmap for Semiconductors [78]. This FOM uses the following slightly different definition:

$$FOM_{ITRS} = \frac{P}{2^{ENOB_0} * \min[f_s, 2 * ERBW]}$$

Here $ENOB_0$ is unambiguously the low frequency ENOB value. For ADCs with full Nyquist input bandwidth, the ITRS FOM uses the sampling frequency just like the ISSCC FOM.

¹As well as being the most widely discussed survey in this author's experience, Murmann's website is the top result when searching either google.com or bing.com for the phrase "ADC survey" as of April 2013.



Figure 7.10: Figure of merit for the 65nm ADC operated at different sampling frequencies using both the ISSCC and ITRS FOM definitions

However, for converters with less than Nyquist bandwidth, double the ERBW is used instead. The performance of ADCs with very high sampling rates (including the work in this thesis) typically drop off below the Nyquist frequency. This makes the ITRS FOM useful for comparing these fast ADCs, since ERBW is taken into account. Fig. 7.10 shows both figures of merit for the 65nm ADC operated at various sampling frequencies.

Using Murmann's data, the ISSCC FOM of the 5GS/s 65nm ADC is plotted along with all other ADCs from the ISSCC and VLSI conferences in Fig. 7.11. Other relevant time-based ADCs are also included. Other than our previous 2.5GS/s ADC in 90nm CMOS [69], the next closest time-based ADC is [79] which achieves a better FOM but at a lower sampling frequency (1.2GS/s).

The performance of the 65nm ADC is summarized in Table 7.2.



Figure 7.11: Figure of merit for published ADCs with time-based ADCs highlighted

Process	65 nm	
Chip area (VTC)	0.0008 mm^2	
Chip area (TDC)	$0.0 \ 8 \text{mm}^2$	
Input Range	200 mV peak-peak differential	
Output Resolution	4 bits	
Sampling rate	5 GS/s	
ERBW	2100 MHz	
SINAD (Peak/@ERBW)	22.9/18.4 dB	
ENOB (Peak/@ERBW)	3.5/2.8 bits	
SFDR (Peak/@ERBW)	34.0/22.3 dB	
Max DNL/INL @DC	0.34/0.38 LSB	
Power dissipation	34.6 mW	
FOM	1.0 pJ/conversion	

Table 7.2: Summary of 65nm ADC Measured Performance

Chapter 8

Conclusions and Future Work

This work has made several important contributions, including:

- The first published time-based Nyquist ADC operating at over 1GS/s [67].
- The fastest time-based ADC currently reported (5GS/s).
- The use of physically separated VTC and TDC chips to form a spatiallydistributed ADC.
- Linearity analysis for a starved-inverter VTC with a closed-form SINAD expression.
- VTC jitter analysis with a simple equation giving close results to full BSIM4 simulations.
- A functional TDC automatic calibration system.

The main product of this work is a 4-bit, 5GS/s ADC fabricated in 65nm CMOS technology. The time-based ADC uses a VTC and TDC on separate boards connected by coaxial cables. With DC inputs, the ADC achieves a maximum DNL and INL of 0.38 and 0.34 LSB respectively and an ENOB of 3.8. At the maximum input frequency of 2100MHz, the DNL and INL are 0.91 and 0.95 LSB and the ENOB is 2.8. The combined power consumption of the VTC and TDC, not including output buffers, is 34.6mA. This results in an ISSCC figure-of-merit of 1.0 pJ/conversion.

8.1 Future Work

The use of time-based ADCs is likely to continue to gain popularity as designers seek to exploit the advantages of deep-submicron CMOS technology. Some suggestions will be offered for future research stemming from the presented work.

8.1.1 Self-Contained, PVT-Independent Circuits

One drawback of starved-inverter circuits is that they suffer from delay variations based on process, voltage and temperature and they require finely tuned bias voltages to correct for this. A major improvement of this work would be to produce self-contained chips that self-generate all necessary bias voltages.

For the VTC, one suggestion is to design bias-generation circuits that automatically adjust to correct for PVT variations. A delay-locked-loop-based solution is a possibility if precise delay control is required, although the complexity of this solution takes away from the simplicity of the VTC and may generate too much noise to allow the VTC to be integrated with other analog blocks. A more elegant solution would involve a reference generator that tracks PVT appropriately without requiring a closed loop. This solution is less likely to guarantee perfect behaviour in all conditions, but the performance may still be acceptable for the application, particular if the TDC is able to calibrate itself specifically to the levels coming from the VTC.

For the TDC, automatic calibration has already been demonstrated with the use of a PC interface. It would be possible to make the calibration process self-contained, however. The proposed system needs to store 4096 4-bit samples. Decimation of the output would reduce the speed requirements for the storage elements. Once all samples have been captured, a digital state machine running at low speed implements the algorithm in Fig. 6.20. The main challenges foreseen are the interface of the decimation system with the digital storage elements as well as the sheer complexity of a state machine with so much storage.

8.1.2 Integration of VTC with SKA Receiver Chain

As mentioned in section 1.5, the ADC was designed with the Square Kilometre Array (SKA) project in mind. The ability to place the VTC and TDC in separate locations is an advantage, since the TDC can be located away from the antenna feed where noise generation and power consumption are less of an issue. For the VTC, integrating the circuit directly with the analog receiver chain would be highly cost-effective. A suggested future project is to produce a chip containing all amplification and filtering blocks needed for an SKA antenna feed along with a VTC. The time-based VTC output would then be sent over fibre-optic cables to the base of the antenna to be digitized by the TDC.

Bibliography

- K. Poulton, J. Corcoran, and T. Hornak, "A 1-GHz 6-bit ADC system," *IEEE Journal of Solid-State Circuits*, vol. 22, no. 6, pp. 962–970, 1987.
- [2] R. Hagelauer, F. Oehler, G. Rohmer, J. Sauerer, and D. Seitzer, "A gigasample/second 5-b ADC with on-chip track and hold based on an industrial 1 mu;m GaAs MESFET E/D process," *IEEE Journal of Solid-State Circuits*, vol. 27, no. 10, pp. 1313–1320, 1992.
- [3] K. Poulton, K. Knudsen, J. Corcoran, K.-C. Wang, R. Nubling, R. Pierson, M.-C. Chang, P. Asbeck, and R. T. Huang, "A 6-bit, 4 GSa/s ADC fabricated in a GaAs HBT process," in *Gallium Arsenide Integrated Circuit (GaAs IC) Symposium*, pp. 240–243, 1994.
- [4] K. Nary, R. Nubling, S. Beccue, W. Colleran, J. Penney, and K.-C. Wang, "An 8-bit, 2 gigasample per second analog to digital converter," in *Gallium Arsenide Integrated Circuit (GaAs IC) Symposium*, pp. 303–306, 1995.
- [5] C. Baringer, J. Jensen, L. Burns, and B. Walden, "3-bit, 8 GSPS flash ADC," in International Conference on Indium Phosphide and Related Materials, pp. 64–67, 1996.
- [6] K. Poulton, K. Knudsen, J. Kerley, J. Kang, J. Tani, E. Cornish, and M. VanGrouw, "An 8-GSa/s 8-bit ADC System," in *Symposium on VLSI Circuits*, 1997, pp. 23–24, 1997.
- [7] W. Ellersick, C.-K. Yang, M. Horowitz, and W. Dally, "GAD: A 12-GS/s CMOS 4bit A/D converter for an equalized multi-level link," in *Symposium on VLSI Circuits*, pp. 49–52, 1999.

- [8] K. Uyttenhove, A. Marques, and M. Steyaert, "A 6-bit 1 GHz acquisition speed CMOS flash ADC with digital error correction," in *Custom Integrated Circuits Conference*, pp. 249–252, 2000.
- [9] M. Choi and A. Abidi, "A 6 b 1.3 GSample/s A/D converter in 0.35 mu;m CMOS," in International Solid-State Circuits Conference, pp. 126–127, 438, 2001.
- [10] K. Mhaidat and O. H. . S. U. O. S. of Science & Engineering. Dept. of Biomedical Engineering, *Representations and Circuits for Time Based Computation*. PhD thesis, Oregon Health & Science University, OGI School of Science & Engineering., 2006.
- [11] R. B. Staszewski *et al.*, "All-digital TX frequency synthesizer and discrete-time receiver for Bluetooth radio in 130-nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 12, pp. 2278–2291, 2004.
- [12] R. Baker, CMOS: Circuit Design, Layout, and Simulation. John Wiley & Sons, 2nd ed., 2005.
- [13] V. Garuts, E. Traa, Y.-S. Yu, and T. Yamaguchi, "A dual 4-bit, 1.5 Gs/s analog to digital converter," in *Bipolar Circuits and Technology Meeting*, pp. 141–144, 1988.
- [14] D. Daniel and B. Bosch, "A silicon bipolar 4-bit 1-Gsample/s full Nyquist A/D converter," *IEEE Journal of Solid-State Circuits*, vol. 23, no. 3, pp. 742–749, 1988.
- [15] H. Chung et al., "A 7.5-GS/s 3.8-ENOB 52-mW flash ADC with clock duty cycle control in 65nm CMOS," in Symposium on VLSI Circuits, (Kyoto, Japan), pp. 268–269, 2009.
- [16] G. Van der Plas, S. Decoutere, and S. Donnay, "A 0.16pJ/Conversion-Step 2.5mW 1.25GS/s 4b ADC in a 90nm Digital CMOS Process," in *IEEE International Solid-State Circuits Conference*, pp. 2310–, 2006.

- [17] R. Taft, C. Menkus, M. Tursi, O. Hidri, and V. Pons, "A 1.8-V 1.6-GSample/s 8-b self-calibrating folding ADC with 7.26 ENOB at Nyquist frequency," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 12, pp. 2107–2115, 2004.
- B. Verbruggen, J. Craninckx, M. Kuijk, P. Wambacq, and G. Van der Plas, "A 2.2mW 5b 1.75GS/s Folding Flash ADC in 90nm Digital CMOS," in *IEEE International Solid-State Circuits Conference*, pp. 252–611, 2008.
- [19] S. Tanifuji et al., "High sampling rate 1 GS/s current mode pipeline ADC in 90 nm Si-CMOS process," in IEEE MTT-S International Microwave Workshop Series on Intelligent Radio for Future Personal Terminals (IMWS-IRFPT), pp. 1–4, 2011.
- [20] T. Sundstrom, C. SVENSSON, and A. Alvandpour, "A 2.4 GS/s, Single-Channel, 31.3 dB SNDR at Nyquist, Pipeline ADC in 65 nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 7, pp. 1575–1584, 2011.
- [21] C.-Y. Chen and J. Wu, "A 12b 3GS/s pipeline ADC with 500mW and 0.4 mm2 in 40nm digital CMOS," in Symposium on VLSI Circuits, pp. 120–121, 2011.
- M. Van Elzakker, E. Van Tuijl, P. Geraedts, D. Schinkel, E. Klumperink, and B. Nauta,
 "A 1.9uW 4.4fJ/Conversion-step 10b 1MS/s Charge-Redistribution ADC," in *IEEE International Solid-State Circuits Conference*, pp. 244–610, 2008.
- [23] H.-Y. Tai, H.-W. Chen, and H.-S. Chen, "A 3.2fJ/c.-s. 0.35V 10b 100KS/s SAR ADC in 90nm CMOS," in *Symposium on VLSI Circuits*, pp. 92–93, 2012.
- [24] P. Harpe, E. Cantatore, and A. van Roermund, "2.2/2.7fJ/conversion-step 10/12b
 40kS/s SAR ADC," in *IEEE International Solid State Circuits Conference*, 2013.
- [25] C.-Y. Liou and C.-C. Hsieh, "A 2.4-to-5.2fJ/conversion-step 10b 0.5-to-4MS/s SAR ADC with Charge-Average Switching DAC in 90nm CMOS," in *IEEE International Solid State Circuits Conference*, 2013.

- [26] L. Kull et al., "A 3.1mW 8b 1.2GS/s Single-Channel Asynchronous SAR ADC with Alternate Comparators for Enhanced Speed in 32nm Digital SOI CMOS," in International Solid-State Circuits Conference, 2013.
- [27] T. Jiang, W. Liu, C. Zhong, C. Zhong, and P. Chiang, "A Single-Channel, 1.25-GS/s, 6bit, 6.08-mW Asynchronous Successive-Approximation ADC With Improved Feedback Delay in 40-nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 10, pp. 2444– 2453, 2012.
- [28] Z. Cao, S. Yan, and Y. Li, "A 32mW 1.25GS/s 6b 2b/step SAR ADC in 0.13um CMOS," in International Solid-State Circuits Conference, pp. 542–634, 2008.
- [29] J. Yang, T. Naing, and R. Brodersen, "A 1 GS/s 6 Bit 6.7 mW Successive Approximation ADC Using Asynchronous Processing," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 8, pp. 1469–1478, 2010.
- [30] K. Poulton et al., "A 20 GS/s 8 b ADC with a 1 MB memory in 0.18 mu;m CMOS," in IEEE International Solid-State Circuits Conference, pp. 318–496 vol.1, 2003.
- [31] C.-C. Huang, C.-Y. Wang, and J.-T. Wu, "A CMOS 6-Bit 16-GS/s time-interleaved ADC with digital background calibration," in *IEEE Symposium on VLSI Circuits*, pp. 159–160, 2010.
- [32] B. Murmann, "ADC Performance Survey 1997-2013." [Online], April 2013. Available: http://www.stanford.edu/ murmann/adcsurvey.html.
- [33] Y. Greshishchev, J. Aguirre, M. Besson, R. Gibbins, C. Falt, P. Flemke, N. Ben-Hamida,
 D. Pollex, P. Schvan, and S.-C. Wang, "A 40GS/s 6b ADC in 65nm CMOS," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, pp. 390–391, 2010.

- [34] J. Fiorenza, T. Sepke, P. Holloway, C. Sodini, and H.-S. Lee, "Comparator-Based Switched-Capacitor Circuits for Scaled CMOS Technologies," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 12, pp. 2658–2668, 2006.
- [35] L. Brooks and H.-S. Lee, "A Zero-Crossing-Based 8b 200MS/s Pipelined ADC," in *IEEE International Solid-State Circuits Conference*, pp. 460–615, 2007.
- [36] H. C. Hor and L. Siek, "Review on VCO based ADC in modern deep submicron CMOS technology," in *IEEE International Symposium on Radio-Frequency Integration Technology*, pp. 86–88, 2012.
- [37] T. Watanabe, T. Mizuno, and Y. Makino, "An all-digital analog-to-digital converter with 12-μV/LSB using moving-average filtering," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 1, pp. 120–125, 2003.
- [38] S. Rao, B. Young, A. Elshazly, W. Yin, N. Sasidhar, and P. Hanumolu, "A 71dB SFDR open loop VCO-based ADC using 2-level PWM modulation," in *Symposium on VLSI Circuits (VLSIC)*, pp. 270–271, 2011.
- [39] H. C. Hor and L. Siek, "K-locked-loop and its application in time mode ADC," in International Symposium on Integrated Circuits, pp. 101–104, IEEE, 2009.
- [40] T. Watanabe and T. Terasawa, "An all-digital A/D converter TAD with 4-shift-clock construction for sensor interface in 0.65- x03BC;m CMOS," in *Proceedings of the ESS-CIRC*, pp. 178–181, 2010.
- [41] A. Tritschler, "A Continuous Time Analog-to-Digital Converter With 90μW and 1.8 μV/LSB Based on Differential Ring Oscillator Structures," in *IEEE International Sym*posium on Circuits and Systems, pp. 1229–1232, IEEE, 2007.
- [42] H. Pekau, A. Yousif, and J. Haslett, "A CMOS integrated linear voltage-to-pulse-delaytime converter for time based analog-to-digital converters," in *IEEE International Sym-*

posium on Circuits and Systems, pp. 4 pp.-2376, 2006.

- [43] Y.-J. Min, A. Abdullah, H.-K. Kim, and S.-W. Kim, "A 5-bit 500-MS/s time-domain flash ADC in 0.18um CMOS," in *International Symposium on Integrated Circuits*, pp. 336–339, 2011.
- [44] Y. Tousi and E. Afshari, "A Miniature 2 mW 4 bit 1.2 GS/s Delay-Line-Based ADC in 65 nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 10, pp. 2312–2325, 2011.
- [45] Y. Arai and T. Baba, "A CMOS time to digital converter VLSI for high-energy physics," in Symposium on VLSI Circuits, pp. 121–122, 1988.
- [46] I. Nissinen, A. Mantyniemi, and J. Kostamovaara, "A CMOS time-to-digital converter based on a ring oscillator for a laser radar," in *European Solid-State Circuits Conference*, pp. 469–472, 2003.
- [47] K. Park and J. Park, "Time-to-digital converter of very high pulse stretching ratio for digital storage oscilloscopes," *Review of Scientific Instruments*, vol. 70, no. 2, pp. 1568– 1574, 1999.
- [48] R. Staszewski, D. Leipold, C.-M. Hung, and P. Balsara, "TDC-based frequency synthesizer for wireless applications," in *IEEE Radio Frequency Integrated Circuits Symposium*, pp. 215–218, 2004.
- [49] M. Lee and A. Abidi, "A 9 b, 1.25 ps Resolution Coarse-Fine Time-to-Digital Converter in 90 nm CMOS that Amplifies a Time Residue," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 4, pp. 769–777, 2008.
- [50] C. Gray, W. Liu, W. Van Noije, J. Hughes, T.A., and R. Cavin, "A sampling technique and its CMOS implementation with 1 Gb/s bandwidth and 25 ps resolution," *IEEE Journal of Solid-State Circuits*, vol. 29, no. 3, pp. 340–349, 1994.

- [51] P. Dudek, S. Szczepanski, and J. Hatfield, "A high-resolution CMOS time-to-digital converter utilizing a Vernier delay line," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 2, pp. 240–247, 2000.
- [52] J. Borremans, K. Vengattaramane, V. Giannini, B. Debaillie, W. Van Thillo, and J. Craninckx, "A 86 MHz-12 GHz Digital-Intensive PLL for Software-Defined Radios, Using a 6 fJ/Step TDC in 40 nm Digital CMOS," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 10, pp. 2116–2129, 2010.
- [53] T. Yamaguchi, S. Komatsu, M. Abbas, K. Asada, N. N. Mai-Khanh, and J. Tandon, "A CMOS flash TDC with 0.84-1.3 ps resolution using standard cells," in *IEEE Radio Frequency Integrated Circuits Symposium*, pp. 527–530, 2012.
- [54] Y.-H. Seo, J.-S. Kim, H.-J. Park, and J.-Y. Sim, "A 1.25 ps Resolution 8b Cyclic TDC in 0.13 m CMOS," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 3, pp. 736–743, 2012.
- [55] P. Dewdney, P. Hall, R. Schilizzi, and T. Lazio, "The Square Kilometre Array," Proceedings of the IEEE, vol. 97, no. 8, pp. 1482–1496, 2009.
- [56] D. Navaratne, "Wide-band Low-Noise CMOS Amplification Stage for a Square Kilometre Array Receiver," Master's thesis, University of Calgary, 2011.
- [57] W. R. Bennett, "Spectra of Quantized Signals," *Bell Systems Technical Journal*, vol. 27, pp. 446–472, July 1948.
- [58] W. Kester, "MT-001: Taking the Mystery out of the Infamous Formula, "SNR=6.02N + 1.76dB," and Why You Should Care." [Online], October 2012. Analog Devices, REV.
 0, 10-03-2005, Available:http://www.analog.com/static/imported-files/tutorials/MT-001.pdf.

- [59] J. Doernberg, H.-S. Lee, and D. A. Hodges, "Full-speed testing of A/D converters," *IEEE Journal of Solid-State Circuits*, vol. 19, no. 6, pp. 820–827, 1984.
- [60] M. F. Wagdy and S. S. Awad, "Determining ADC effective number of bits via histogram testing," vol. 40, no. 4, pp. 770–772, 1991.
- [61] "IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters," 2011.
- [62] P. Wambacq and W. M. Sansen, Distortion Analysis of Analog Integrated Circuits. Norwell, MA, USA: Kluwer Academic Publishers, 1998.
- [63] B. Razavi, Design of Analog CMOS Integrated Circuits. McGraw-Hill series in electrical and computer engineering, McGraw-Hill, 2001.
- [64] O. Cobanoglu, "Estimating Hand Calculation Parameters (K, Cox, Vth)." [Online], November 2012. Available:http://personalpages.to.infn.it/~cobanogl/lowlevelstuff /tutparext/.
- [65] A. Abidi, "Phase Noise and Jitter in CMOS Ring Oscillators," IEEE Journal of Solid-State Circuits, vol. 41, pp. 1803–1816, Aug. 2006.
- [66] K. Townsend, Towards an Interference-Mitigating Transmitted-Reference UWB Receiver. PhD thesis, University of Calgary, 2010.
- [67] A. R. Macpherson, K. A. Townsend, and J. W. Haslett, "A 5GS/s voltage-to-time converter in 90nm CMOS," in *Proc. European Microwave Integrated Circuits Conference*, pp. 254–257, 2009.
- [68] K. A. Townsend, A. R. Macpherson, and J. W. Haslett, "A fine-resolution Time-to-Digital Converter for a 5GS/S ADC," in *Proc. IEEE Int Circuits and Systems (ISCAS)* Symposium, pp. 3024–3027, 2010.

- [69] A. R. Macpherson, K. A. Townsend, and J. W. Haslett, "A 2.5GS/s 3-bit time-based ADC in 90nm CMOS," in Proc. IEEE Int Circuits and Systems (ISCAS) Symp, pp. 9– 12, 2011.
- [70] B. Nikolic, V. G. Oklobdzija, V. Stojanovic, W. Jia, J. K.-S. Chiu, and M. Ming-Tak Leung, "Improved sense-amplifier-based flip-flop: design and measurements," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 6, pp. 876–884, 2000.
- [71] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital integrated circuits- A design perspective*. Prentice Hall, 2ed ed., 2004.
- [72] S. Padoan, A. Boni, C. Morandi, and F. Venturi, "A novel coding scheme for the ROM of parallel ADCs, featuring reduced conversion noise in the case of single bubbles in the thermometer code," in *IEEE International Conference on Electronics, Circuits and Systems*, vol. 2, pp. 271–274 vol.2, 1998.
- [73] K. Uyttenhove and M. Steyaert, "A 1.8-V 6-bit 1.3-GHz flash ADC in 0.25-um CMOS," IEEE Journal of Solid-State Circuits, vol. 38, pp. 1115 – 1122, july 2003.
- [74] D. Lee, J. Yoo, K. Choi, and J. Ghaznavi, "Fat tree encoder design for ultra-high speed flash A/D converters," in *The 2002 45th Midwest Symposium on Circuits and Systems*, vol. 2, pp. II–87 – II–90 vol.2, aug. 2002.
- [75] T. Linnenbrink, S. Tilden, and M. Miller, "ADC testing with IEEE Std 1241-2000," in IEEE Instrumentation and Measurement Technology Conference, vol. 3, pp. 1986–1991, 2001.
- [76] B. E. Jonsson, "Using Figures-of-Merit to Evaluate Measured A/D Converter Performance," in 2011 International Workshop on ADC Modelling, Testing and Data Converter Analysis and Design and IEEE 2011 ADC Forum, (Orvieto, Italy), pp. 1–6, June 2011.

- [77] R. Walden, "Analog-to-digital converter survey and analysis," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 4, pp. 539–550, 1999.
- [78] "System Drivers, International Technology Roadmap for Semiconductors, 2011 Edition." [Online]. Available: http://www.itrs.net/Links/2011ITRS/2011Chapters/2011SysDrivers.pdf.
- [79] Y. Tousi and E. Afshari, "A Miniature 2 mW 4 bit 1.2 GS/s Delay-Line-Based ADC in 65 nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 46, pp. 2312 –2325, Oct. 2011.