

An Analog/Mixed Signal FFT Processor for Ultra-Wideband OFDM Wireless Transceivers

Mark Lehne

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Electrical Engineering

Sanjay Raman, Chair

Jeffrey H. Reed

Steven W. Ellingson

Joseph G. Tront

Cameron Patterson

William H. Woodall

August 28, 2008

Blacksburg, Virginia

Keywords: OFDM, UWB, MB-OFDM, FFT Processor, Analog, Mixed Signal,
WiMedia, IC

Copyright 2008, Mark Lehne

An Analog/Mixed Signal FFT Processor for Ultra-Wideband OFDM

Wireless Transceivers

Mark Lehne

ABSTRACT

As Orthogonal Frequency Division Multiplexing (OFDM) becomes more prevalent in new leading-edge data rate systems processing spectral bandwidths beyond 1 GHz, the required operating speed of the baseband signal processing, specifically the Analog-to-Digital Converter (ADC) and Fast Fourier Transform (FFT) processor, presents significant circuit design challenges and consumes considerable power. Additionally, since Ultra-WideBand (UWB) systems operate in an increasingly crowded wireless environment at low power levels, the ability to tolerate large blocking signals is critical. The goals of this work are to reduce the disproportionately high power consumption found in UWB OFDM receivers while increasing the receiver linearity to better handle blockers.

To achieve these goals, an alternate receiver architecture utilizing a new FFT processor is proposed. The new architecture reduces the volume of information passed through the ADC by moving the FFT processor from the digital signal processing (DSP) domain to the discrete time signal processing domain. Doing so offers a reduction in the required ADC bit resolution and increases the overall dynamic range of the UWB OFDM receiver.

To explore design trade-offs for the new discrete time (DT) FFT processor, system simulations based on behavioral models of the key functions required for the processor are presented. A new behavioral model of the linear transconductor is introduced to better capture non-idealities and mismatches. The non-idealities of the linear transconductor, the largest contributor of distortion in the processor, are individually varied to determine their sensitivity upon the overall dynamic range of the DT FFT processor. Using these behavioral models, the proposed architecture is validated and guidelines for the circuit design of individual signal processing functions are presented. These results indicate that the DT FFT does not require a high degree of linearity from the linear transconductors or other signal processing functions used in its design.

Based on the results of the system simulations, a prototype 8-point DT FFT processor is designed in 130 nm CMOS. The circuit design and layout of each of the circuit functions; serial-to-parallel converter, FFT signal flow graph, and clock generation circuitry is presented. Subsequently, measured results from the first proof-of-concept IC are presented. The measured results show that the architecture performs the FFT required for OFDM demodulation with increased linearity, dynamic range and blocker handling capability while simultaneously reducing overall receiver power consumption. The results demonstrate a dynamic range of 49 dB versus 36 dB for the equivalent all-digital signal processing approach. This improvement in dynamic range increases receiver performance by allowing detection of weak sub-channels attenuated by multipath. The measurements also demonstrate that the processor rejects large narrow-band blockers, while maintaining greater than 40 dB of dynamic range. The processor enables a 10x reduction in power consumption compared to the equivalent all digital processor, as it consumes only 25 mW and reduces the required ADC bit depth by four bits, enabling application in hand-held devices.

Following the success of the first proof-of-concept IC, a second prototype is designed to incorporate additional functionality and further demonstrate the concept. The second proof-of-concept contains an improved version of the serial-to-parallel converter and clock generation circuitry with the additional function of an equalizer and parallel-to-serial converter.

Based on the success of system level behavioral simulations, and improved power consumption and dynamic range measurements from the proof-of-concept IC, this work represents a contribution in the architectural development and circuit design of UWB OFDM receivers. Furthermore, because this work demonstrates the feasibility of discrete time signal processing techniques at 1 GSps, it serves as a foundation that can be used for reducing power consumption and improving performance in a variety of future RF/mixed-signal systems.

Acknowledgments

First and foremost, I would like to thank God, through whom all things are possible.

I would like to thank my committee chair and faculty advisor, Sanjay Raman Ph.D. for his guidance, support, and tireless help. I would like to thank my committee members, Jeffrey H. Reed Ph.D, Steven W. Ellingson Ph.D, Joseph G. Tront Ph.D, Cameron Patterson Ph.D, and William H. Woodall Ph.D, for their time, advice, and good discussions.

I am especially thankful to my wife, Rebecca, for her daily support, motivation and inspiration and to my family for their patience while I pursued my dream.

I would like to thank Doug Juanarena and Andrew Duggleby, Ph.D for their encouragement throughout my years in Blacksburg, and to Ken Boehlke of Focus Enhancements Semiconductor Group for his discussions and perspective.

I am grateful to the Bradley Department of Electrical and Computer Engineering and the Institute for Critical Technologies and Science (IC-TAS) for their financial support.

It has been a pleasure working with the members of Virginia Tech Wireless Microsystems Lab, Jun Zhao, Gustina Collins, Krishna Vummidi, Rich Sivetik, Ibrahim Chamas, Swaminathan Muthukrishnan, Joe Wood, Nikhil Kakkar, and Marcus Oliver. I am thankful for the conversations and entertainment through the countless hours in the lab together .

Contents

1	Introduction	1
1.1	An Introduction to OFDM Systems	2
1.1.1	The Indoor Wireless Channel	4
1.1.2	OFDM Symbol Generation	7
1.1.3	Cyclic prefix and windowing	13
1.1.4	WiMedia MB-OFDM for UWB	18
1.2	Architectural challenges in UWB OFDM transceivers	21
1.2.1	Performance Metrics for Wireless Receivers	23
1.2.2	UWB OFDM Receiver Front-Ends	29
1.2.3	Analog-to-Digital Converters for Ultra-Wideband Receivers	32
1.2.4	State-of-the-Art Digital FFT Processors for UWB OFDM	35
1.3	UWB baseband processing using discrete-time Analog Signal Process- ing	37
1.4	Proposed OFDM Architecture	39
1.5	Dissertation Organization	39
1.5.1	Objective of Dissertation	39

1.5.2	Outline of Dissertation	40
2	Discrete Time FFT Processor Architecture	42
2.1	A Discrete Time Signal Processing Compatible FFT Topology	42
2.1.1	The Fast Fourier Transform	43
2.2	The Proposed Discrete Time Analog FFT Processor	46
2.2.1	Discrete Time Butterfly Structure	47
2.2.2	Serial-to-Parallel Function	52
2.2.3	Clock Generation	54
2.2.4	The Discrete Time Sub-Channel Equalizer	55
2.2.5	Parallel-to-Serial Converter	56
2.3	Summary	57
3	System simulations of the DT FFT Processor	58
3.1	Discrete Time Signal Processing	58
3.1.1	Multipliers for use in Discrete Time Signal Processing	59
3.1.2	Adders for use in Discrete Time Signal Processing	63
3.1.3	Discrete Time Memory	64
3.2	Behavioral Models	66
3.3	System Simulation Results	73
3.3.1	Optimizing the G_{m_0} value	74
3.3.2	Voltage Gain through the Multiplier and Adder	74
3.3.3	a-to- V_{max} ratio	76
3.3.4	A_r ratio	78
3.3.5	A_r variation	78

3.3.6	G_m offset	79
3.3.7	V_{in} offset	79
3.3.8	Jitter	80
3.3.9	Comparison with All Digital Processing	82
3.3.10	Blockers	84
3.3.11	Ptolemy System Simulations	86
3.3.12	Power Consumption Savings	87
3.4	Summary	88
4	Circuit Design and Layout	89
4.1	Multiply and Add Function	89
4.1.1	Multiplier	89
4.1.2	Analog Adder	94
4.2	Sample-and-Holds	99
4.3	Clock Generation Circuitry	106
4.3.1	“Power-PC” D-flip-flop	108
4.4	IC Peripheral Circuit Designs	110
4.4.1	Driver Amplifiers	114
4.5	IC Layout	120
4.6	Summary	126
5	Measurement Results	127
5.1	Test Setup	128
5.2	Characterization of Instrumentation Amplifiers, Instrumentation Multiplexer and Driver Amplifiers	135
5.3	Characterization of the Serial-to-Parallel Converter Test IC	137

5.4	Characterization of the DT FFT Processor IC	139
5.5	Summary	144
6	An Improved DT FFT Processor Design	146
6.1	Equalizer	146
6.2	Parallel-to-Serial Conversion Function	149
6.2.1	Buffer SHA	150
6.2.2	Combining Sample-and-Hold circuit	150
6.3	Clocking Circuitry	152
6.3.1	Differential Sense Amplifier D-flip-flop	156
6.3.2	Differential AND, Inverters	159
6.4	IC Peripheral Circuit Designs	160
6.5	IC Layout	160
6.6	Summary	163
7	Conclusions and Future Work	167
7.1	Conclusions	167
7.2	Future Work	171
A	Verilog-AMS listings and SPICE Netlists	172
	Bibliography	180

List of Figures

1.1	A hypothetical receiver based on a bank of ideal filters that allow frequency division multiplexing of simultaneously received parallel narrowband channels.	3
1.2	Frequency Division Multiplexed system requiring guard bands between each channel (a), the OFDM approach (b) is more spectrally efficient.	4
1.3	An example indoor power delay profile showing the rms delay spread.	5
1.4	The frequency response of the example delay profile from Figure 1.3 .	6
1.5	Block diagram of the OFDM symbol creation process	7
1.6	The constellation plot of the QPSK symbol given by $x_k = 1 e^{j\angle 90^\circ}$. .	9
1.7	(a) Time domain plot of a single OFDM symbol consisting of a QPSK symbol $x_k = 1 e^{j\angle 90^\circ}$ mapped to a sub-carrier of normalized frequency 3. (b) Frequency spectra of the OFDM symbol.	10
1.8	The constellation plot of the symbol given by $x_k = 0.5 e^{j\angle -45^\circ}$	11
1.9	(a) Time domain plot of a single OFDM symbol consisting of a symbol $x_k = 0.5 e^{j\angle -45^\circ}$ mapped to a sub-carrier of normalized frequency -1. (b) Frequency spectra of the OFDM symbol.	11
1.10	(a) Time domain plot of a single OFDM symbol consisting of the symbols $x_k = 1 e^{j\angle 90^\circ}$ and $x_k = 0.5 e^{j\angle -45^\circ}$ mapped to sub-carriers of frequency 3 and -1 respectively. (b) Frequency spectra of the OFDM symbol.	12

1.11 (a) Time domain plot of a discrete sampled OFDM symbol consisting of the symbol $x_k = 1 e^{j\angle 90^\circ}$ mapped to a sub-carrier of frequency 3. (b) Frequency spectra of the OFDM symbol.	13
1.12 Example symbol separated into three individual example sub-carriers 3, 6 and 12 in (a-c), and summed in (d). The effects of channel delay spread profile only degrade the leading part of the symbol which is located in the guard interval.	15
1.13 An example of the addition of cyclic prefix and windowing of a single OFDM symbol. (a) shows the 64-point output of the IFFT. (b) The lead and tail portions are copied to the head and tail of the longer symbol. (c) Finally the symbol is filtered with a Hanning window. The final symbol is made up of 112 discrete time samples: 16 samples for the header window, 16 samples for the cyclic prefix, 64 samples contain the data payload, and 16 samples for tail windowing.	17
1.14 The frequency band plan for the WiMedia MB-OFDM standard [1] .	18
1.15 Block diagram of a direct conversion OFDM transceiver. (a) Transmitter data path, (b) Receiver data path	22
1.16 The receiver RF front-end, baseband, analog-to-digital conversion and DSP are represented by different signaling domains: continuous-time versus discrete-time and variable signal amplitude versus fixed signal amplitude. Although OFDM receivers are typically quadrature, only one baseband path is shown for simplicity.	23
1.17 Front-end spurious free dynamic range is calculated from the input referred third-order intercept point and the input noise power.	24
1.18 (a) The shape of the input amplitude versus SNDR plot for a typical circuit. (b) The three principal contributors, noise, distortion and clipping, that affect the shape of the typical input amplitude versus SNDR plot.	26

1.19	The non-linear harmonics and intermodulation harmonics resulting from a two tone test are shown for continuous time frequency spectrum in (a) and the discrete time frequency spectrum in (b). In the discrete time case, sub-sampling of higher frequency spurs causes them to ‘fold’ around the F_s point, into the lower frequency band.	28
1.20	(a) The link budget of a receiver front-end and ADC shows the difference between the dynamic range of the 6-bit ADC and 10-bit ADC. (b) For the case of an in-band blocker, the dynamic range of the 6-bit ADC is insufficient and the weaker sub-channels are lost.	31
1.21	Moore’s law shows microprocessor performance growth doubling every 1.5 years. Meanwhile, flash ADC performance is doubling only every 5.7 years.	35
1.22	Parallelism is used to achieve the 409.6MSps data rate required of digital FFT processors for WiMedia MB-OFDM.	36
1.23	The block diagram of the baseband signal processing portion for a (a) traditional OFDM receiver and (b) the proposed modified OFDM receiver. Three different signaling domains separate the circuit functions.	40
2.1	The signal flow lattice representation of an 8-point FFT.	45
2.2	The signal flow diagram of the butterfly structure	46
2.3	The FFT lattice shown in an discrete time signal processing compatible form	47
2.4	Block diagram of the proposed Discrete Time FFT processor	48
2.5	FFT butterfly circuit with hardwired coefficients constructed from transconductance amplifiers and current adders.	49
2.6	FFT butterfly circuit with tunable coefficients constructed from transconductance amplifiers and current adders.	51
2.7	The z-domain representation of the serial to parallel function.	52
2.8	Open loop Sample and Hold	53

2.9	(a) The serial-to-parallel function realized with sample-and-hold amplifiers. (b) The clock timing diagram used.	54
2.10	Signal flow diagram of one channel of the complex equalizer	55
2.11	(a) The parallel to serial function realized with sample-and-hold amplifiers. (b) The clock timing diagram used.	56
3.1	The typical schematic of a discrete time signal processing based FIR filter.	59
3.2	The differential pair multiplying DAC architecture. The current sources can either be binary weighted for a binary scaled DAC or equally sized for a segmented DAC.	60
3.3	A multiplying DAC based on the Gilbert cell	61
3.4	The pseudo differential multiplying DAC architecture	62
3.5	The linear degenerated differential pair	63
3.6	The input coupled linear degenerated differential pair	63
3.7	The cross-coupled current steering transconductor	64
3.8	A cascode transresistive current adder	65
3.9	Open loop Sample and Hold	65
3.10	The curves used in the behavioral model of the G_m cell coefficient multiplier. (a) The voltage-in current-out curve defined by equation (3.1) (b) The voltage-in transconductance-out curve formed by the derivative of equation (3.1)	68
3.11	The setup used to simulate the discrete-time FFT processor.	73
3.12	Varying the transconductance of the multipliers affects the useable input voltage range when operating current is held constant.	75
3.13	Simulating the DT FFT processor with different G_m values shows that lower values allow a larger dynamic range.	75

3.14	The combined gain of the multiplier and adder combination affects the dynamic range of the system.	76
3.15	Varying the a-to- V_{max} ratio of the G_m cell behavioral model determines the quasi-linear range of the transconductance curve useful for multiplication (inset). The SNDR curves show that the a-to- V_{max} ratio does not have a strong effect on dynamic range for values above 50%. . . .	77
3.16	Amplitude ripple, A_r models the non-ideality found in the quasi-linear region of the G_m cell's transconductance curve (inset). The SNDR curves show that high levels of amplitude ripple lower peak SNDR but do not degrade the dynamic range.	79
3.17	Monte-Carlo simulation of the discrete-time FFT processor with several values of standard deviation in (a) G_m offset and (b) voltage offset applied to the G_m cell behavioral model	81
3.18	Simulation results of the discrete-time FFT processor with clock jitter applied to the clock divider input.	82
3.19	The simulation setup used to simulate the all digital comparison FFT processor.	82
3.20	Simulation results of the discrete-time FFT processor (solid) compared to simulation results of the all-digital FFT processor with varying levels of input ADC quantization (dashed). The discrete-time FFT processor exceeds the dynamic range of the all-digital FFT processor with 9-bit resolution.	83
3.21	Simulation results of the discrete-time FFT processor dynamic range (solid) versus narrow band blocker magnitude demonstrates that the processor is able to perform demodulation in the presence of large narrow-band blockers. For comparison, the blocker performance of the 6-bit all digital system is shown (dashed).	85
3.22	The system simulation setup used in Ptolemy based simulations. . . .	86
3.23	The EVM sweep across input signal magnitude shows that the DT FFT Processor performs better than an ideal digital system of 8-bits. . . .	87

4.1	A portion of the butterfly structure used in the transistor level design of the coefficient multiply and add.	90
4.2	The common source differential pair is one of the simplest forms of the CMOS transconductor	91
4.3	The ideal transconductor has a voltage-to-current transfer function (a) and a voltage-to-transconductance transfer function (b) with a wide flat region near the center, V_{in} . In contrast, the typical source coupled differential pair is also shown.	92
4.4	The linear transconductor used in the construction of the FFT butterfly structure.	93
4.5	Simulated transconductance of the variable G_m cell is adjusted through bias C_k	95
4.6	The adder circuit used in the construction of the FFT butterfly structure provides independant common-mode resistance and differential mode resistance.	95
4.7	Simulated Adder circuit transresistance tuning as a function of Pbias	96
4.8	(a) Simulated voltage-in, voltage-out transfer function of the half butterfly structure. (b) shows the derivative of (a), which is the voltage gain of the half butterfly structure.	98
4.9	Simulated frequency response of the half butterfly structure with typical loading.	98
4.10	The serial-to-parallel conversion function implemented by two banks of sample-and-hold amplifiers.	99
4.11	The PFET based sample-and-hold with source follower amplifier. . .	100
4.12	Simulated drain-source resistance versus device width of a PFET switch with $L_g = 120nm$ and 4 fingers. The left axis shows gate-to-bulk capacitance.	102
4.13	Simulated open switch frequency response of the sample-and-hold amplifier.	102

4.14	Simulation results of an $800mV_{pk-pk}$ 80MHz sine-wave passing through the track-and-hold with 1GHz clock.	104
4.15	The NFET switch based sample-and-hold with source following amplifier.	105
4.16	Simulated drain-source resistance versus device width of a NFET switch with $L_g = 120nm$ and 4 fingers. The left axis shows channel capacitance.	106
4.17	The ten phase clock divider constructed from D-flip-flops and NAND gates.	107
4.18	The NAND circuit used in the 10 phase clock generator. Outputs are scaled to drive SHAs.	109
4.19	The simulation results of the NAND gate.	110
4.20	The “PowerPC” D-FlipFlop design used in the 10 phase clock generation.	111
4.21	Simulation results of the ten-phase clock divider showing clock phases 2 and 3	112
4.22	Noise Filter and Diode Latch-up protection circuit for voltage biased pads.	113
4.23	Noise Filter and Diode Latch-up protection circuit for current biased pads.	113
4.24	On chip 50-Ohm termination reduces RF coupling to substrate. . . .	114
4.25	The instrumentation mux and driver amplifier consists of the input level shift amplifier, impedance buffer amplifier, output mux, and 50Ω driver amplifier.	115
4.26	The instrumentation level shift amplifier.	116
4.27	The transimpedance feedback amplifier extends amplifier bandwidth.	117
4.28	The low input capacitance buffer amplifier.	118
4.29	The 50Ω output impedance driver amplifier.	119
4.30	The layout of the DT FFT processor with the DT FFT processor core, instrumentation interface circuits and driver amplifiers.	121

4.31	The layout of the DT FFT processor core consisting of clock divider, PFET switch SHA bank, NFET switch SHA bank, and four columns of multiply and adder circuits.	121
4.32	The wirebonding diagram shows how the IC is connected to the package with the shortest bondwires used for the sensitive RF input and output paths.	122
4.33	The layout of the ten phase clock divider. The D-flip-flops are placed close together to minimize interconnect delay whereas the NAND gates are spaced loosely to aid in the full custom layout process.	123
4.34	The layout of the D-flip-flop is made compact to maximize switching speeds.	124
4.35	The layout of the pseudo-differential sample-and-hold amplifier consists of two single ended sample-and-hold amplifiers placed as mirror images about the horizontal axis of symmetry.	124
4.36	The layout of the butterfly structure consists of G_m cells, adders and a current mirror.	125
4.37	The layout of a pair of G_m cells. Common centroid and interleaving techniques are applied to minimize mismatch.	126
5.1	The die photograph of the DT FFT processor prototype with pins and key sections labeled.	128
5.2	The signal generation and measurement setup used for the Discrete-Time FFT processor.	129
5.3	The physical measurement setup used to measure the Discrete-Time FFT Processor.	130
5.4	The printed circuit board with the test IC, bias DACs and voltage regulators.	134
5.5	Through test IC S-parameters (a) S21 single ended, (b) S22 from 10 MHz to 500 MHz, (c) S11 input match, (d) S22 output match	136

5.6	The measurement result of a 10 MHz, 600 mV_{pk} triangle wave applied to the through test IC.	137
5.7	The down-sampled OFDM symbol stream measured at the output of the serial-to-parallel converter.	138
5.8	(a) A 1GSps OFDM input signal as applied to the input of the OFDM processor. (b) Four of the eight parallel demodulated outputs.	140
5.9	Measurement results after being captured on the oscilloscope and re-combined in MATLAB for a single demodulated output channel from the FFT processor.	141
5.10	Measurement results after symbol timing recover in MATLAB of a single demodulated output channel displayed in XY format.	142
5.11	Measurement results for the Discrete-Time FFT processor show a peak SNDR of 36dB and a Dynamic Range of 49dB.	142
5.12	Measurement results for the Discrete-Time FFT processor dynamic range after rejecting sinusoidal blocker of varied input magnitude. . .	143
6.1	The equalizer circuit scales real and imaginary inputs to correct for sub-channel magnitude and phase error.	148
6.2	The adder circuit used in the equalizer cell is similar to Figure 4.6 but with M1,M2 sized for higher resistance and higher gain.	148
6.3	The Parallel-to-Serial conversion function consists of a bank of impedance buffer SHAs, a bank of switches and a single summing capacitors. For simplicity, the differential I and Q lines are represented by a single line in the signal flow diagram.	149
6.4	The low input capacitance SHA used in the parallel-to-serial converter.	150
6.5	The combining sample-and-hold circuit operates by sequentially turning on one switch at a time to sample the parallel input data onto C_{hold}	152
6.6	The clock generation circuit used in the first prototype of the DT FFT processor.	154

6.7	The clock generation circuit used in the second prototype IC creates 10 clock phases and utilizes inverter drivers individually scaled to drive the circuit functions within the DT FFT Processor.	155
6.8	The clock generating diagram for the second prototype IC including the synchronization input.	156
6.9	The sense amplifier D-flip-flop is constructed from two circuits, a pulse generator and a slave latch.	157
6.10	The circuit diagram of the sense amplifier D-flip-flop. The sense amplifier pulse generating circuit (a) and the set-reset slave latch (b) . .	158
6.11	The differential AND gate used in the clock generation circuitry. . . .	159
6.12	The 50 Ω output impedance driver amplifier.	161
6.13	The layout of the improved DT FFT processor with the DT FFT processor core, instrumentation interface circuits and driver amplifiers. .	164
6.14	The layout of the improved DT FFT processor core consisting of clock generation circuit, serial-to-parallel convert, three columns of multiply and add circuits, equalizer and parallel-to-serial converter.	164
6.15	The layout of the clock generation circuit.	165
6.16	The layout of the sense amplifier D-flip-flop.	165
6.17	The layout of a single channel of the equalizer.	166
6.18	The layout of the buffer SHA.	166
A.1	Verilog-AMS code of the G_m cell coefficient multiplier behavioral model	173
A.2	Verilog-AMS code of the Sample-and-Hold Amplifier behavioral model	174
A.3	Verilog-AMS code of the adder	174
A.4	Verilog-AMS code of the Serial-to-Parallel Function	175
A.5	cont. Verilog-AMS code of the Serial-to-Parallel Function	176
A.6	Verilog-AMS code of the Parallel-to-Serial Function	177

A.7	SPICE netlist of the Butterfly Structure for P1N1	178
A.8	SPICE netlist of the AMS FFT Processor	179

List of Tables

1.1	Multiband OFDM System Parameters	19
1.2	Performance of WiMedia MB-OFDM Receiver Front Ends.	29
1.3	High Speed Analog to Digital Converters suitable for UWB OFDM.	34
2.1	The quadrature differential wiring of the PS block	49
2.2	The Timing Requirements for the Serial-to-Parallel Function	53
2.3	The Timing Requirements for the Parallel-to-Serial Function	57
3.1	Summary of Model Parameters used in Jitter and Blocker Simulations	80
3.2	Summary of Design Goals based on System Simulations of the discrete-time FFT Processor	88
4.1	Summary of Simulation Results for the PFET Switch SHA design	101
4.2	Summary of Simulation Results for the NFET Switch SHA design	105
4.3	The capacitive load presented to the different clock outputs	108
4.4	The timing results of the NAND simulation.	108
5.1	The specifications of the Tek AWG7102 Arbitrary Waveform Generator	131
5.2	The specifications of the Tek TDS694C Oscilloscope	132
5.3	The specifications of the AD5308 bias generation DAC	133
5.4	Summary of Measurement Results	144

6.1	Simulation Results for the buffer SHA design	151
6.2	Simulation Results of clock load capacitance for the Combining Sample- and-Hold circuit	151
6.3	The capacitive load presented to the each clock output from the clock generation circuit	153
6.4	The timing results of the Sense Amplifier D-flip-flop simulation. . . .	159

Chapter 1

Introduction

Since the advent of wireless digital communications, there has been tremendous growth in the demand for wireless information transfer between various multimedia and computing devices. In recent years, the transmission requirements have become sufficiently large to require transceivers that operate over significantly wider radio channels.

In 2002, the United States Federal Communications Commission (FCC) responded to these demands with the approval of several new allocations of radio frequency spectrum for use with Ultra-WideBand (UWB) radios, primarily in the 3.1-10.6 GHz range. The FCC defines UWB transmissions as those having a bandwidth greater than 25% of the center radio frequency or greater than 500 MHz [2]. Because data rate is proportional to bandwidth, UWB enables a significant increase in wireless data capacity compared to narrowband systems using equivalent transmitter powers. Following the opening of the new UWB spectrum, the IEEE 802.15.3a standard, which later evolved into the WiMedia standard, was developed to address indoor wireless networks operating over the 3.1 to 10.6 GHz range [3].

As with previous indoor wireless local area networking standards, IEEE 802.11.3g at 2.4 GHz, and IEEE 802.11.3a at 5 GHz, the WiMedia standard utilizes Orthogonal Frequency Division Multiplexing (OFDM). OFDM is a digital data modulation technique developed specifically to overcome the physical limitations of the indoor wireless channel for high-data rate systems. The maximum data rate of the previous IEEE 802.11.3a/g standards is 54Mbits/sec, while the maximum data rate of the proposed

WiMedia standard is 384 Mbits/sec; future indoor wireless standards aspire to data rates in excess of 1 Gbit/sec.

Although Gbit/sec data rates are theoretically possible, there are significant challenges to realizing these data rates in low-cost, low-power, silicon Complementary Metal Oxide Semiconductor (CMOS) technology using conventional signal processing techniques. The objective of this dissertation is to explore new approaches to perform high-speed signal processing for OFDM modulation at UWB frequencies that will enable future low power CMOS implementations of indoor wireless digital communications systems.

1.1 An Introduction to OFDM Systems

The maximum amount of data that can be transferred through a wireless communications channel is defined by the Shannon capacity limit which defines the theoretical maximum capacity C in (bits/sec) as:

$$C = B \log_2 (1 + SNR) \quad (1.1)$$

where B is the bandwidth of the channel and SNR is the *signal-to-noise ratio*. The signal-to-noise ratio (SNR) is the signal power at the receiver divided by the noise at the receiver. Thus, as new communication standards attempt to increase the data rate of a system, they can either increase the bandwidth of the system or the SNR . Since wirelessly transmitted signals lose signal power with distance, there are two fundamental means of increasing the SNR : one is to increase the transmitter power, the other is to decrease the operating distance. In digital wireless communications, symbols are used to represent one or more data bits; the higher the expected receiver SNR , the more bits that can be included in a symbol. If the expected SNR at a receiver is increased, more data bits can be included in each symbol, increasing the overall data rate.

Since the FCC sets a limit on transmitter power, and consumer application requirements demand maximum transmission distance, the expected receiver SNR is typically limited. However, given the large available bandwidth of the new 3.1-10.6 GHz

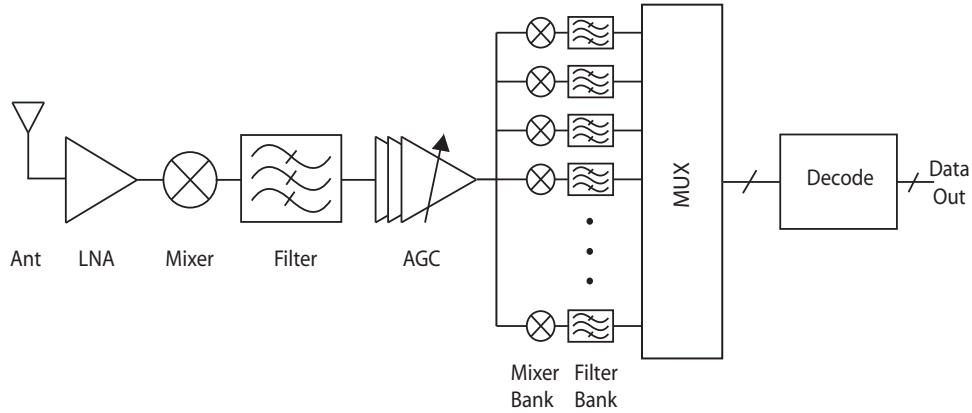


Figure 1.1: A hypothetical receiver based on a bank of ideal filters that allow frequency division multiplexing of simultaneously received parallel narrowband channels.

UWB band, systems that can effectively increase operating bandwidths have the opportunity to significantly increase data rates.

However, a physical limitation known as multi-path inhibits wireless systems from easily increasing operating bandwidths to more than a few hundred megahertz. Multi-path and the properties of the wireless air channel are discussed in greater detail below. However, first consider a basic method of increasing data rate and operating bandwidth through parallelism.

If N parallel low bandwidth digital transceivers were used to transmit data in separate parts of a large bandwidth, the cumulative data rate could be large. However, using N antennas, amplifiers, filters, etc., runs counter to the goal of a low-power, small form-factor consumer device for high data rate communication system.

Instead, consider the hypothetical Frequency Division Multiplexing (FDM) receiver as shown in Figure 1.1 which requires a parallel bank of mixers and filters. This hypothetical receiver uses frequency division over a large number of narrowband channels to achieve an overall high system data rate [4]. Each narrowband channel supports a low data rate and uses a narrowband filter to isolate the data from other channels. When these channels are multiplexed together, a faster overall data rate is achieved.

The problem with this viewpoint is that it is not efficient with the use of frequency spectrum. In practice, filters have finite roll off (Q), and therefore, guard bands are needed to avoid interference between adjacent channels [Figure 1.2(a)]. Alternatively,

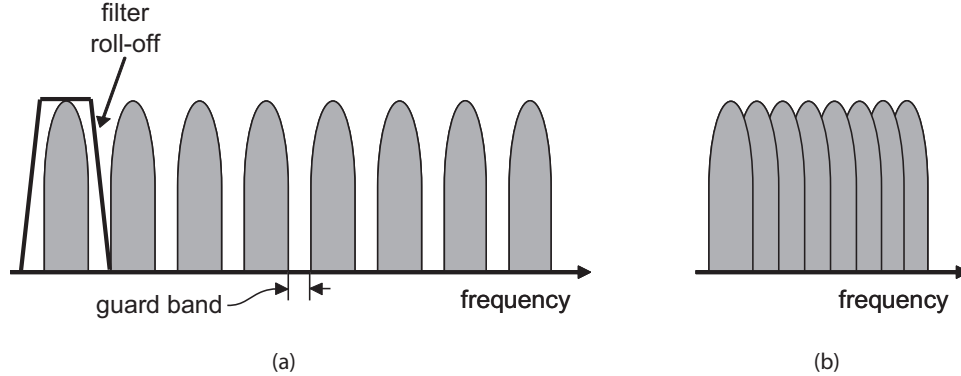


Figure 1.2: Frequency Division Multiplexed system requiring guard bands between each channel (a), the OFDM approach (b) is more spectrally efficient.

if each channel could be made orthogonal by another means, guard bands and high-Q physical filters would not be needed and the system could be implemented monolithically. In OFDM systems, the orthogonal nature of the Fourier transform is used to separate the sub-channels, resulting in no wasted spectrum for filter guard bands [Figure 1.2(b)]. This allows for higher data rates and efficient spectrum usage.

1.1.1 The Indoor Wireless Channel

The indoor wireless channel is uniquely different from many common terrestrial radio propagation channels. Antennas are often physically small and omnidirectional due to required form factors and the multi-gigahertz frequency range of operation [5].

Because of the short wavelength of signals in the UWB band, signal paths exist between the transmitter and receiver resulting from reflections off the walls, floor, ceiling, furniture, and even people in the surrounding environment [6]. The distance along each of these paths is different, causing delayed signals to arrive at the receiver at different times and combine at different magnitudes and phases. This is known as multi-path. The distribution of arrival times of these different paths is called the delay profile, and can be used to describe the wireless environment for a given space. Although the delay profile is a continuous function, due to the edges and the rough surfaces of the reflectors in a typical indoor environment, it is frequently shown as a collection of discrete impulses that each represent a particular propagation path [7, 8].

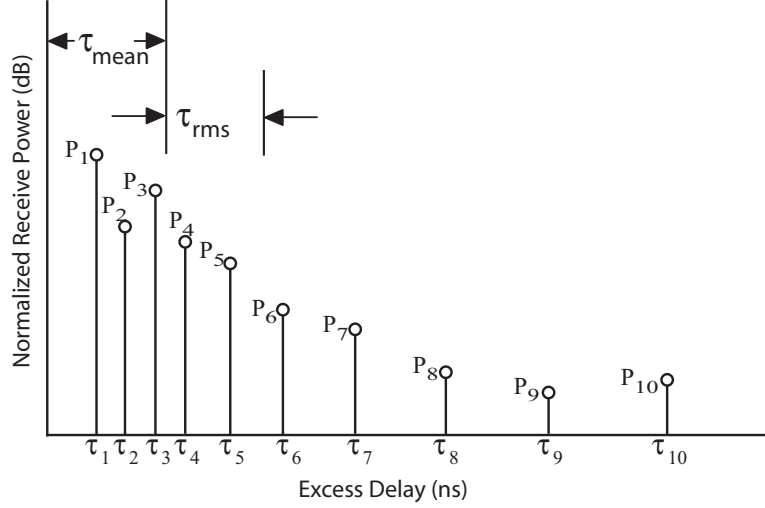


Figure 1.3: An example indoor power delay profile showing the rms delay spread.

Figure 1.3 shows an example of a typical indoor delay profile.

When comparing different delay profiles, the measure of rms delay spread, τ_{rms} , is often used. τ_{rms} is the standard deviation of the delay profile, and is given by:

$$\tau_{rms} = \sqrt{\frac{\sum_k P_k \tau_k^2}{\sum_k P_k} - \left(\frac{\sum_k P_k \tau_k}{\sum_k P_k} \right)^2} \quad (1.2)$$

where P_k is the linear power of the k th path, and τ is the arrival time of the k th path [9].

When translated into the frequency domain, the delay profile represents a frequency response with sharp nulls. These nulls are known as frequency-selective fades, i.e. frequencies at which very little energy will be propagated. For many indoor channels, the frequency response is assumed to be time invariant, or changing so slowly that its effects are negligible during the transmission time of a single data packet. The coherence bandwidth B_c is also a typical parameter used to describe a wireless air-channel and is inversely proportional to the delay spread [9]:

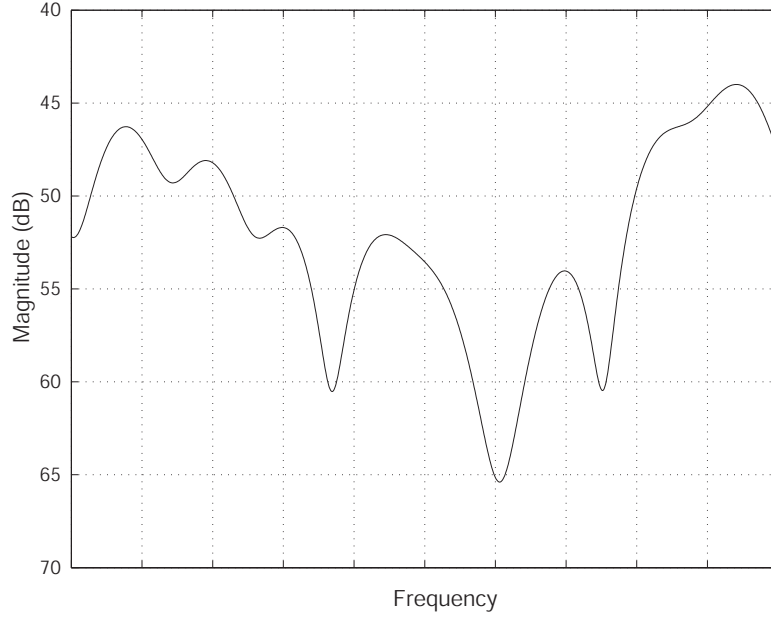


Figure 1.4: The frequency response of the example delay profile from Figure 1.3

$$B_c \approx \frac{1}{5\tau_{rms}} \quad (1.3)$$

Because the coherence bandwidth is only approximately defined, it is more precise to discuss rms delay spread. However, it is sometimes constructive to use the coherence bandwidth for illustration [9]. If the bandwidth of a wireless signal is less than the coherence bandwidth of the channel, it is said to experience “flat-fading”. Flat fading is desirable because the received signal does not experience frequency selective fading, making it easier to receive signals. When the bandwidth of the wireless signal exceeds the coherence bandwidth, there is a high probability of frequency selective fades affecting a portion of the signal bandwidth, causing some frequencies to be significantly attenuated. Figure 1.4 shows an example of a frequency selective fading. The example frequency response is the Fourier transform of the delay profile shown previously in Figure 1.3. The receiver must correct the attenuated portions of the frequency spectrum that have experienced fading, a process which can require intensive signal processing, known as equalization.

For modeling UWB indoor channels between 3.1 and 10.6 GHz, researchers have

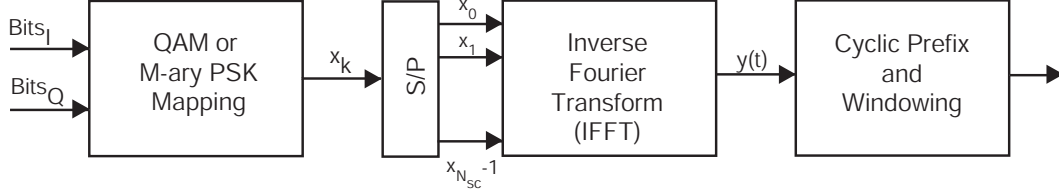


Figure 1.5: Block diagram of the OFDM symbol creation process

suggested a typical τ_{rms} value of $5ns$ and a maximum value of $25ns$ be used [6–8,10,11]. Recall that the time it takes an electromagnetic wave to travel one meter in free space is approximately 3.3 nanoseconds; this value is $\frac{2}{3}$ the reported rms delay spread value of $5ns$. Thus, the typical indoor environment will have multiple propagation paths which differ in length by approximately 1.5 meters. Meanwhile, the maximum reported value of detectable delay paths of $25ns$ corresponds to a maximum path length of approximately 7.5 meters. It is assumed that longer reflection paths are largely attenuated [1].

In order to design a system that is robust in the presence of frequency selective fading channels, it is beneficial to select a low enough symbol rate R_{symb} , such that the symbol period τ_{symb} is greater than τ_{rms} , or in other words, one that has a much higher probability of only experiencing flat fades. Yet to achieve high data rates, it is necessary to use the fastest possible symbol rate which may require $\tau_{symb} < \tau_{rms}$. In the next section it will be shown how OFDM maintains $\tau_{symb} > \tau_{rms}$ while simultaneously increasing the effective symbol rate.

1.1.2 OFDM Symbol Generation

The generation of an OFDM symbol is a multi-step process that consists of mapping data bits to symbols at a high input symbol rate and then using the inverse Fourier transform to map the high input symbol rate to a single low symbol rate OFDM output with long symbol times. Figure 1.5 illustrates this process.

In the first step, bits are mapped to M-ary quadrature amplitude modulation (QAM) or phase shift keying (PSK) [12]. This gives each symbol x_k a magnitude, $|x_k|$, and an angle, $\angle x_k$. After the symbols are mapped, a total of N_{sc} symbols (the subscript

sc refers to sub-carriers, which will be discussed below) are simultaneously passed to the inverse Fourier transform. This is often performed as a serial-to-parallel (S/P) function, storing the serial symbols x_k until all N_{sc} symbols are collected.

The inverse Fourier transform is defined as:

$$x(t) = \int_{-\infty}^{\infty} X(f) \exp(j2\pi ft) df \quad (1.4)$$

where $X(f)$ is the input frequency domain waveform, and $x(t)$ is the output time domain waveform. Using the inverse Fourier transform to map input symbols, the k th parallel input symbol, given by, $|x_k| \exp(j\angle x_k)$ is mapped to the k th sub-carrier f_{sc} :

$$f_{sc} = \frac{k}{T_{sOFDM}} \quad (1.5)$$

where T_{sOFDM} is the symbol time for an OFDM symbol.

The sub-carriers are represented by impulse (Dirac delta) functions in the frequency domain. If the sub-carriers are orthogonal then they all exist at unique frequencies. In the time domain the sub-carriers are represented by a complex exponential $\exp(j2\pi f_{sc}t)$ with a magnitude of unity. Thus the integral of Equation 1.4 can be reduced to a summation as given by equation (1.6):

$$y(t) = \sum_{k=0}^{N_{sc}-1} |x_k| \exp(j\angle x_k) \exp\left(\frac{j2\pi kt}{T_{sOFDM}}\right) * \text{rect}\left(\frac{t}{T_{sOFDM}}\right) \quad (1.6)$$

where k is the sub-carrier position. ‘rect’ is the rectangular function which is convolved with the complex exponential sub-carriers to bound the time to a length of T_{sOFDM} . Although Equation 1.6 is discrete in the frequency domain, it is continuous in the time domain. Equation 1.6 defines sub-carriers with only integer values of k . This ensures that the orthogonal nature of the sub-carriers is preserved in the time domain. Using integer values of k also means that number of periods over the symbol time T_{sOFDM} is an integer. If a sub-carrier with a fractional value of k were permitted, then the convolution of the ‘rect’ function would cause energy from the sub-carrier

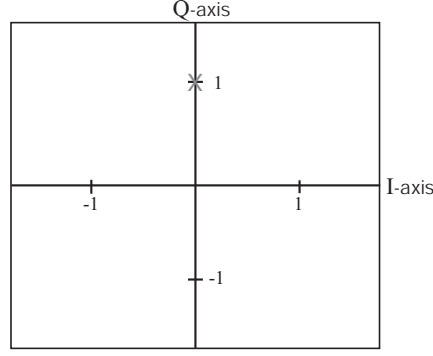


Figure 1.6: The constellation plot of the QPSK symbol given by $x_k = |1| e^{j\angle 90^\circ}$.

to contribute to other sub-carriers.

For illustration purposes, it is helpful to consider the case of a single input symbol x_k being mapped to the k th sub-carrier with all other input symbols being zero. In this case, the output $y(t)$ is given by:

$$y(t) = |x_k| \exp \left(\frac{j2\pi kt}{T_{sOFDM}} + j\angle x_k \right) * \text{rect} \left(\frac{t}{T_{sOFDM}} \right) \quad (1.7)$$

The Fourier transform of $y(t)$ is calculated to be:

$$Y(f) = T_{sOFDM} |x_k| \exp(\angle x_k) \cdot \text{sinc}(\pi T_{sOFDM} (f - f_{sc})) \quad (1.8)$$

where the ‘sinc’ is the well known function, $\sin(x)/x$. As can be seen, the frequency spectrum $Y(f)$ is that of a sinc function centered at k , with lobes at multiples of $1/T_{sOFDM}$ and with the phase and magnitude of the input symbol x_k represented at the center frequency of the main lobe.

As an example consider the case of $x_k = |1| e^{j\angle 90^\circ}$ which represents a simple QPSK symbol as shown in the constellation diagram in Figure 1.6. In this discussion, the frequency is normalized by setting the symbol time to $T_{sOFDM} = 1$. Consider this symbol mapped to the third sub-carrier, $f_{sc} = 3$.

$$y(t) = 1 \cdot \exp(j2\pi(3)t + 90^\circ) * \text{rect} \left(\frac{t}{1} \right) \quad (1.9)$$

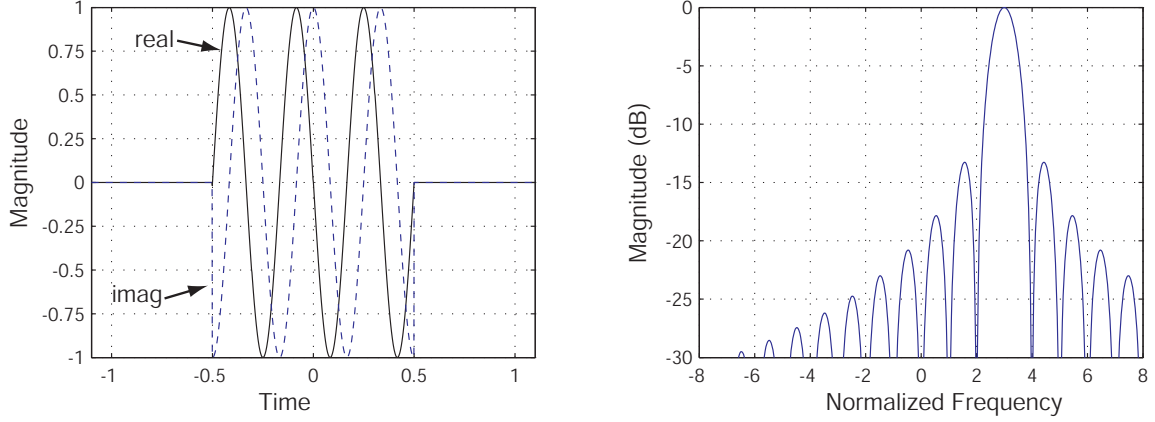


Figure 1.7: (a) Time domain plot of a single OFDM symbol consisting of a QPSK symbol $x_k = |1|e^{j\angle 90^\circ}$ mapped to a sub-carrier of normalized frequency 3. (b) Frequency spectra of the OFDM symbol.

The corresponding frequency spectra is:

$$Y(f) = 1 \cdot \exp(j90^\circ) \cdot \text{sinc}(\pi(f - 3)) \quad (1.10)$$

$y(t)$ and $Y(f)$ for this example are shown in Figure 1.7. Note that the complex sinusoid in 1.7(a) is limited to one time period $T_{s_{OFDM}} = 1$ and has three cycles. Also note that the phase is $+90^\circ$ at time zero. In 1.7(b) the sinc function results in side-lobes at non-integer frequencies; however, at the integer frequencies defined by $k/T_{s_{OFDM}}$ the magnitude is zero. This is significant because it demonstrates that energy from this symbol will not interfere with sub-carriers at other integer frequencies, a key feature of OFDM processing.

Now, consider the case of the symbol $x_k = |0.5|e^{j\angle -45^\circ}$, shown in the constellation plot in Figure 1.8, mapped to the sub-carrier at normalized frequency -1 ($f_{sc} = -1$). Here $y(t)$ is represented by:

$$y(t) = 0.5 \cdot \exp(j2\pi(-1)t - 45^\circ) * \text{rect}\left(\frac{t}{1}\right) \quad (1.11)$$

and the corresponding frequency spectra is:

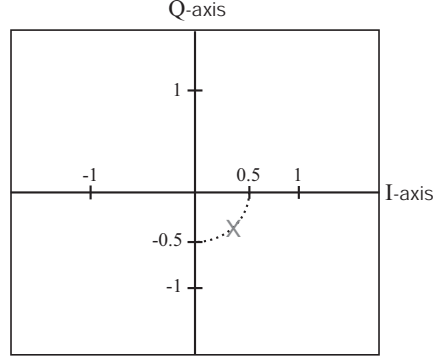


Figure 1.8: The constellation plot of the symbol given by $x_k = |0.5|e^{j\angle -45^\circ}$.

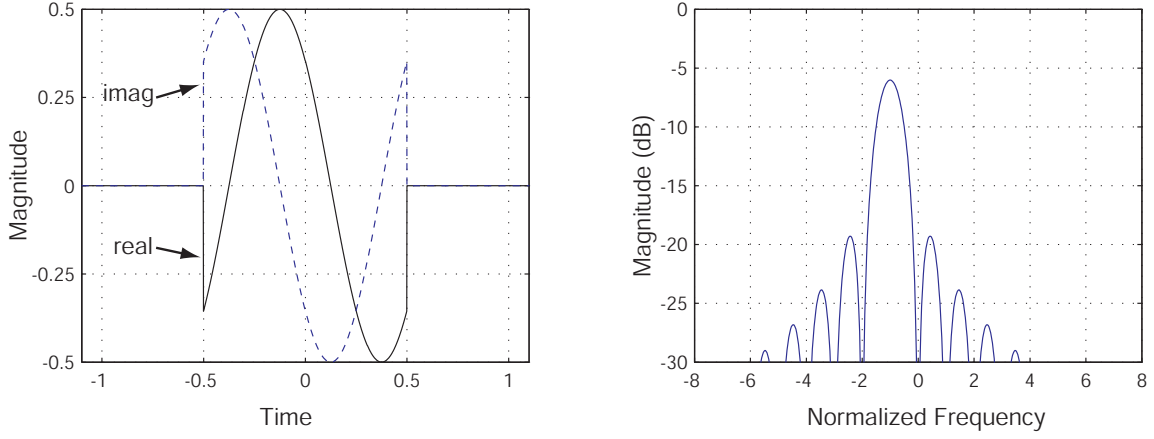


Figure 1.9: (a) Time domain plot of a single OFDM symbol consisting of a symbol $x_k = |0.5|e^{j\angle -45^\circ}$ mapped to a sub-carrier of normalized frequency -1. (b) Frequency spectra of the OFDM symbol.

$$Y(f) = 0.5 \cdot \exp(-j45^\circ) \cdot \text{sinc}(\pi(f + 1)) \quad (1.12)$$

For this case, $y(t)$ and $Y(f)$ are shown in Figure 1.9. Note that the sub-carrier has one complete cycle and fits into the symbol time $T_{sOFDM} = 1$. In the frequency spectra, the magnitude of the primary lobe of the sinc function is 6dB below unity, corresponding to $|x_k| = 0.5$.

In the example shown in Figure 1.10, the two symbols previously discussed $x_k = |1|e^{j\angle 90^\circ}$ and $x_k = |0.5|e^{j\angle -45^\circ}$, are simultaneously mapped to the sub-carriers,

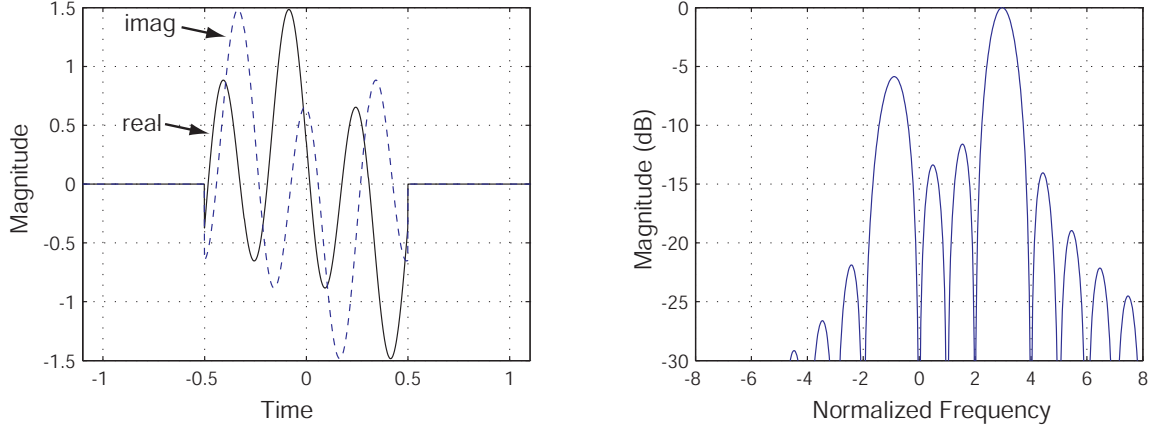


Figure 1.10: (a) Time domain plot of a single OFDM symbol consisting of the symbols $x_k = |1|e^{j\angle 90^\circ}$ and $x_k = |0.5|e^{j\angle -45^\circ}$ mapped to sub-carriers of frequency 3 and -1 respectively. (b) Frequency spectra of the OFDM symbol.

$f_{sc} = 3$ and $f_{sc} = -1$, respectively. Because the two sub-carriers are orthogonal, they add without creating interference at integer frequencies. In the time domain [Figure 1.10(a)] the sinusoids add both constructively and destructively over time, while creating a waveform that is still cyclic over the time $T_{sOFDM} = 1$. In the frequency domain [Figure 1.10(b)] it is easy to see the magnitude and frequency of the two OFDM encoded symbols.

The three previous examples all utilized a continuous time representation for visualization purposes; however OFDM systems typically operate in the discrete-time sampled domain. For the discrete-time case, Equation 1.6 can be simplified for time samples n over the symbol time $T_{sOFDM} = N_{sc}$ to be:

$$y[n] = \sum_{k=0}^{N_{sc}-1} |x_k| \exp(j\angle x_k) \exp\left(\frac{j2\pi kn}{N_{sc}}\right) \quad (1.13)$$

The rect function is not needed in the discrete-time representation of the inverse Fourier Transform as time, index n , is limited to N_{sc} samples.

Consider a discrete-time example similar to the first example of $x_k = |1|e^{j\angle 90^\circ}$ and $f_{sc} = 3$ (Figure 1.7), but with $y[n]$ discrete-time sampled with $N_{sc} = 8$ samples in the period of time, $T_{sOFDM} = 1$. The discrete-time OFDM symbol is defined by two

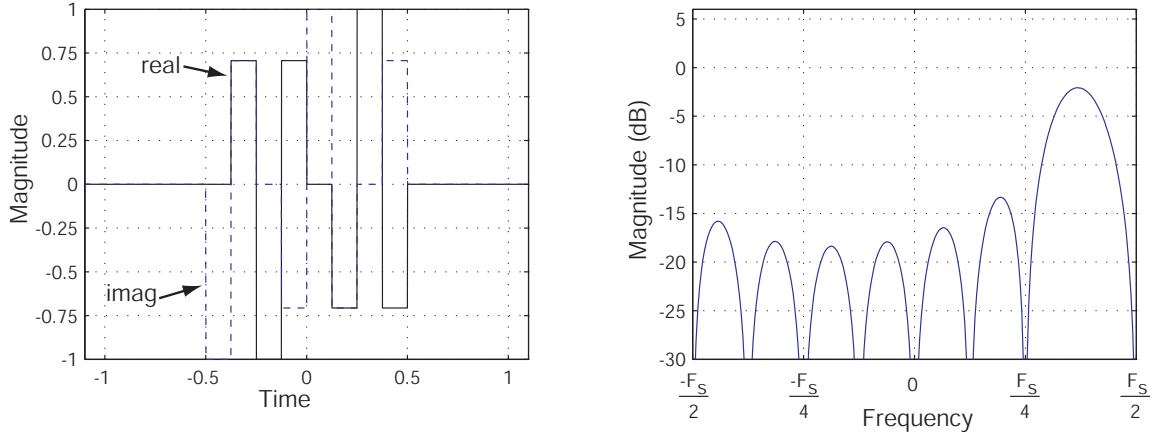


Figure 1.11: (a) Time domain plot of a discrete sampled OFDM symbol consisting of the symbol $x_k = |1| e^{j\angle 90^\circ}$ mapped to a sub-carrier of frequency 3. (b) Frequency spectra of the OFDM symbol.

time constants: the sample time, T_{samp} and the symbol time, T_{sOFDM} . Figure 1.11(a) shows the time domain plot, and Figure 1.11(b) shows the frequency domain plot in terms of Nyquist frequency, F_s , where $F_s = 1/T_{\text{samp}}$.

Having described the basics of OFDM symbol generation in this section, the next section discusses additional features of the OFDM modulation approach, specifically the cyclic prefix and windowing.

1.1.3 Cyclic prefix and windowing

Although an OFDM symbol is primarily based on the Fourier transform, the addition of a cyclic prefix is required for acceptable wireless transmission. As discussed above, the Fourier transform ensures orthogonality between sub-carriers and separates the individual sub-channels in the frequency domain. Since the sub-channels are narrow compared to the coherence bandwidth, they are robust against frequency selective fades. However there is still the issue of the transient response of the delay spread profile interacting with the leading edge of each periodic OFDM symbol.

Mathematically the effect of transmission through the wireless channel is equivalent to convolving the delay spread profile with the transmitted signal. The time domain response of this effect at the receiver is a transient period of distortion that settles and

is followed by the remnant of the periodic symbol, possibly altered in magnitude and phase. Because the individual sub-carriers in the OFDM symbol are independent, superposition applies. Therefore, the effect of multi-path delays on the full OFDM symbol is equivalent to applying the delay spread individually to each sub-carrier and then summing [13].

Consider the example shown in Figure 1.12. The three steady state sinusoids, labeled as the “payload” in Figure 1.12(a-c), represent three orthogonal sub-carriers used to construct an OFDM symbol. When the delay spread is introduced, the signals are distorted for an initial transient period. Figure 1.12(d) shows the result of summing the three subcarriers. It is noted that, although altered in phase and magnitude, the symbol remaining after the initial transient period is still periodic.

Thus, if the OFDM symbol is constructed such that the initial transient period is actually a non data-bearing “guard interval”, then the data bearing portion of the symbol will experience no transient distortion. This is significant as it demonstrates that orthogonality is maintained between the sub-channels even after they experience multi-path distortion. When the guard interval is discarded in the receiver, the remaining symbol is free from transient distortion.

The signal placed in the guard interval, known as the *cyclic prefix*, is a redundant ($\sim 25\%$) portion of the inverse Fourier transformed symbol. The length of the prefix is chosen to exceed the rms delay spread, τ_{rms} . The cyclic prefix is typically taken from the tail end of the inverse Fourier transformed symbol. Since two periodic signals placed sequentially are together periodic, the OFDM symbol formed from the concatenation of the cyclic prefix and the inverse Fourier transformed symbol is also periodic. This ensures that, at the receiver after the cyclic prefix is discarded, the remaining portion of the symbol, also known as the payload, is free from delay spread distortion and the orthogonal properties of the sub-carriers are retained. The data bearing portions of the signal that experience gain and phase rotation behave as if they had only experienced flat fading, which can easily be corrected for in an equalizer.

Windowing can also be employed, in addition to the cyclic prefix, in systems that require increased orthogonality between the sub-channels. As was seen in Equations 1.7 - 1.8, the result of limiting the periodic symbol in time with the *rect* function

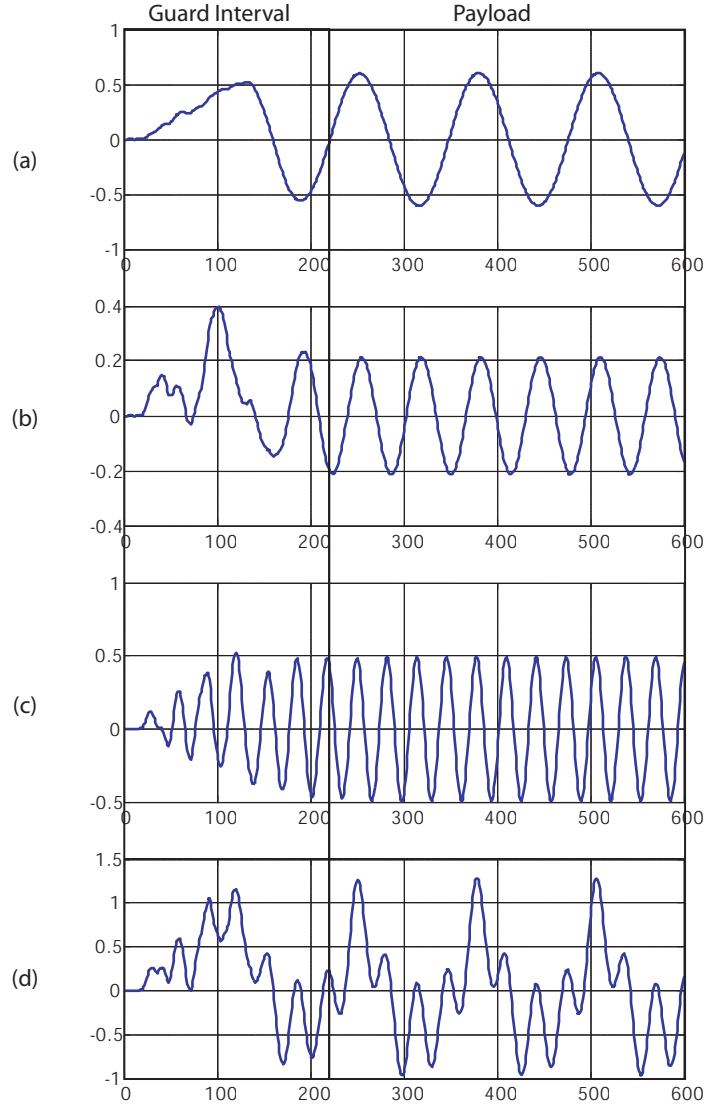


Figure 1.12: Example symbol separated into three individual example sub-carriers 3, 6 and 12 in (a-c), and summed in (d). The effects of channel delay spread profile only degrade the leading part of the symbol which is located in the guard interval.

causes a *sinc* function in the frequency domain to occur centered at the sub-carrier. In signal processing theory, the *rect* function would be called a “brick-wall filter” [14]. The drawback to the brick-wall filter is that the first side-lobe is only 13 dB below the magnitude of the main lobe. The use of other windowing filters, such as the Hamming, Hanning or Blackman, are known to increase the attenuation of the side-lobes. When one of these filters is applied to an OFDM symbol, side-lobes are further suppressed.

To add a windowing filter, additional portions of the symbol are copied from the data bearing payload and are added to the head and tail of the symbol, increasing its length. The symbol is then filtered with the chosen filter function before transmission by the windowing function. The additional filtering smooths the time domain transition between one symbol and the next. In the frequency domain, the windowing decreases the sub-channel sidelobes, further reducing the potential for inter-subcarrier interference.

The example in Figure 1.13 shows a complete OFDM symbol based on a 64-point inverse Fourier transform with cyclic prefix, header and tail windows. This symbol is 112 discrete time samples in length and long enough to clearly observe that the cyclic prefix function and windowing effects. The 64 sample data payload resulting from a 64-point inverse Fourier transform can be seen at time samples 33-96. The header window, at time samples 1-16, and the cyclic prefix, at time samples 17-32 in (b), can be seen to be copies of the data payload samples at time samples 65-96 in (a). The tail window, at samples 97-112 in (b) can be seen to be a replica of data payload samples 33-48 in (a). The entire OFDM symbol has also been passed through a Hanning window which has filtered the header and tail portions of the symbol. In total, this example OFDM symbol is comprised of 112 discrete-time samples, of which, 64 represent the actual data.

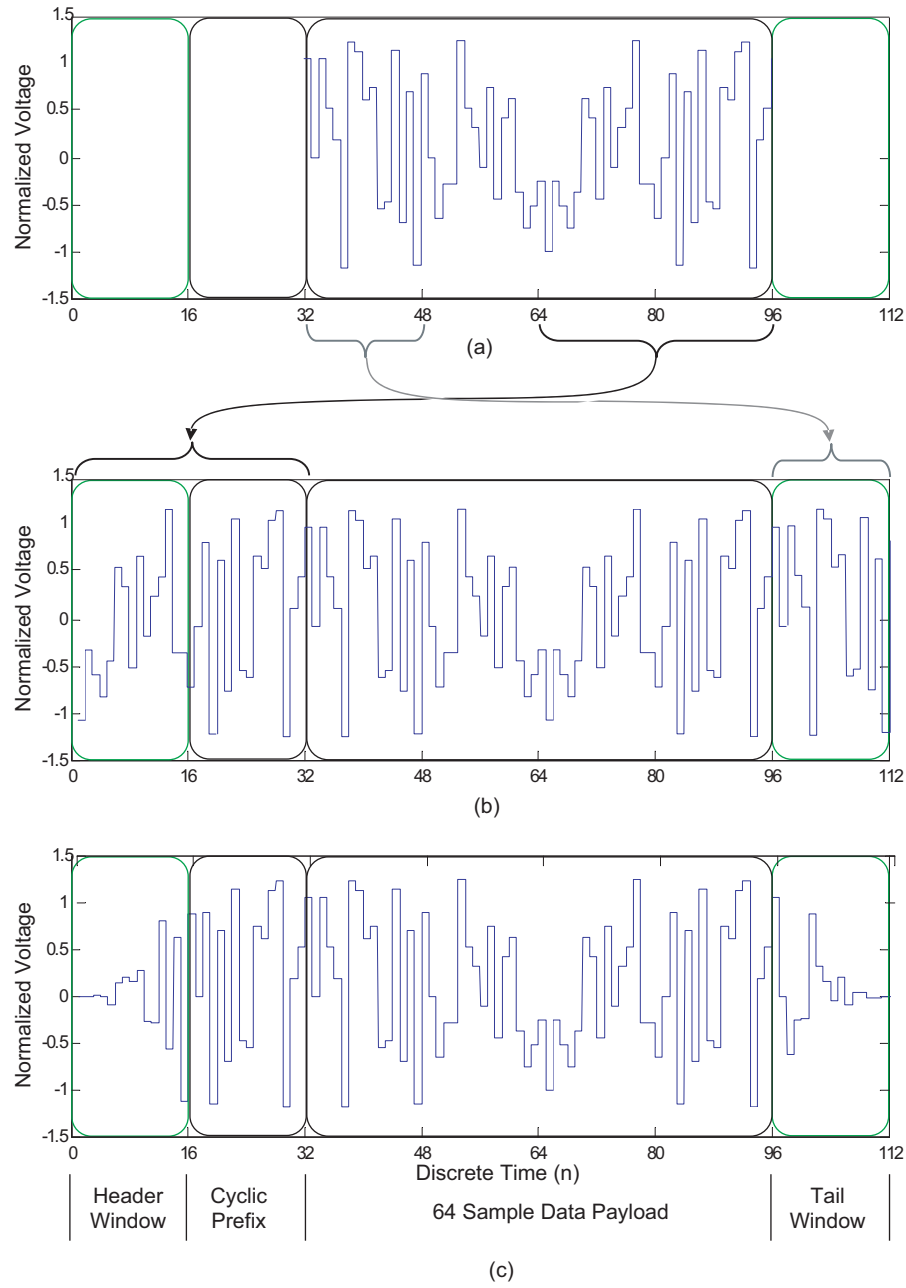


Figure 1.13: An example of the addition of cyclic prefix and windowing of a single OFDM symbol. (a) shows the 64-point output of the IFFT. (b) The lead and tail portions are copied to the head and tail of the longer symbol. (c) Finally the symbol is filtered with a Hanning window. The final symbol is made up of 112 discrete time samples: 16 samples for the header window, 16 samples for the cyclic prefix, 64 samples contain the data payload, and 16 samples for tail windowing.

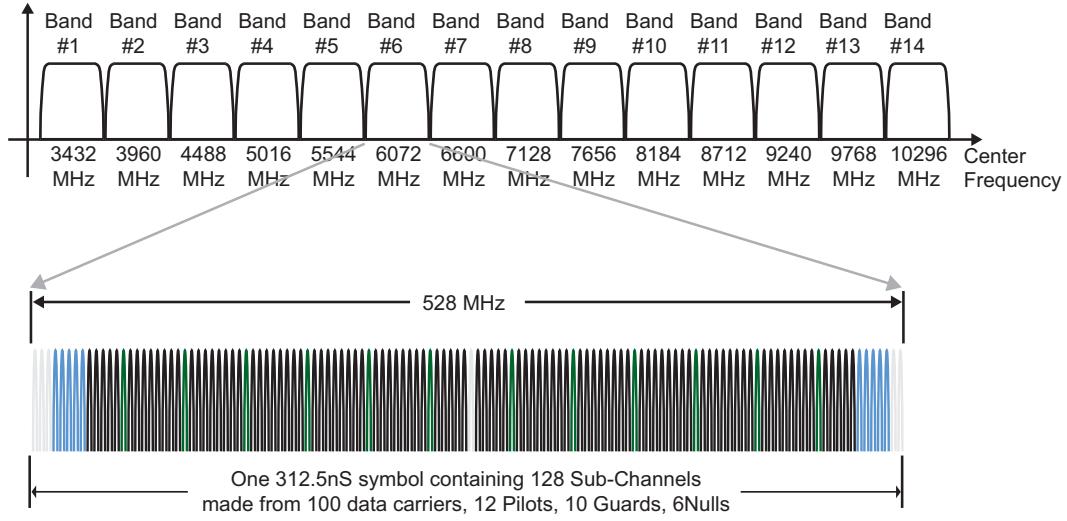


Figure 1.14: The frequency band plan for the WiMedia MB-OFDM standard [1] .

1.1.4 WiMedia MB-OFDM for UWB

The WiMedia MB-OFDM UWB specification (formerly the proposed IEEE 802.15.3a standard) is targeted for data rates up to 480 Mbps at indoor distances less than 10 meters [1]. The WiMedia MB-OFDM frequency plan divides the 3.1-10.6 GHz spectrum into fourteen 528 MHz bands. Each of the 528 MHz bands is made up of 128 sub-channels of 4.125 MHz each. The frequency domain mapping of the sub-channels can be seen in Figure 1.14.

The 528 MHz bandwidth was chosen to allow for the maximum compatibility with different countries' spectral masks, while still meeting the FCC definition of UWB. Another advantage of the proposed 14 band scheme, is that it allows time division band hopping making room for more simultaneous users. Band hopping also allows for avoidance bands with strong interferers. However, when three or less bands are available, time hopping becomes less useful and can represent a significant loss in throughput. Currently, in the United States all 14 bands are available for UWB use; comparatively, in Europe bands 1-3 and 7-10 are permitted, in Japan bands 2-3 and 9-13, and in Korea bands 1-3 and 9-13. The lower bands, 1-3, are the most desirable since the transmission loss is lower, allowing for greater transmission distances. Bands 4-5 are not typically used to avoid potentially strong blockers from Wireless LAN 802.11.a and UNII transmitters.

Table 1.1: Multiband OFDM System Parameters

RF Channel Bandwidth	528 MHz
Complex Baseband Channel Bandwidth	264 MHz
FFT Size	128
Sub-Channel Bandwidth	4.125 MHz
Number of Data sub-channels	100
Total Symbol Period	312.5ns
Windowing	0ns
Guard Interval	9.5ns
Cyclic Prefix	60.6ns
Data Payload Time	242.4ns
bits encoded per sub-channel	2
max error correction coding rate	3/4
Max Data Rate	480Mbps

When selecting the FFT size, WiMedia system designers initially estimated that the FFT processor would comprise $\sim 25\%$ of the transceiver's baseband digital complexity and sought to optimize the system to minimize FFT processor size and therefore power consumption [1]. FFT sizes of 64-points and 128-points were both extensively simulated by the 802.15.3a committee through expected multi-path environments and ultimately the 128 point FFT was determined to perform slightly better than the 64-point FFT [10].

The timing of the OFDM symbol is shown in Table 1.1. The period occupied symbols is $312.5ns$. This consists of a $9.5ns$ null-time between symbols to avoid inter-symbol-interference (ISI), a $60.6ns$ cyclic prefix and a $242.4ns$ data payload. The cyclic prefix of $60.6ns$ allows for delay spreads in excess of the $5ns$ expected from the channel model discussed in Section 1.1.1 [6–8, 10, 11]. The $242.4ns$ payload of the OFDM symbol consists of 128 samples that make up the inverse fast Fourier transform (IFFT). In the frequency domain, the 128 time samples correspond to the 128 frequency sub-channels. Only 100 frequency sub-channels carry data and the remaining consist of 12 pilot sub-channels, 10 guard sub-channels, and 6 sub-channels. The pilot sub-channels are used to aid in equalization of the receiver and are placed among the 100 data sub-channels. The guard sub-channels contain psuedo-random data that can be distorted by filter roll-off and discarded in the DSP portion of the receiver. This allows for finite Q, low-order channel select and DAC filters that can inadvertently

distort the edge sub-channels near their cutoff frequency. At the extreme band edges of the 528 MHz, five sub-channels are nulled to improve the shape of the transmitted spectral mask and improve adjacent channel power rejection. A single sub-channel at the center of each band is nulled to allow for AC coupling to avoid DC offsets if a direct-conversion receiver is used.

There is an efficiency impact that arises in the frequency domain when non-data bearing sub-channels are used, and a similar efficiency impact in the time-domain when the cyclic prefix and windowing samples are used. The cost of the frequency domain pilot sub-channels, guard sub-channels and null sub-channels is a data throughput efficiency of 78.1%, i.e. only 78.1% of the total frequency band is being used for data. The total cost of the time domain guard interval and cyclic prefix is a data throughput efficiency of 77.5%, i.e. only 77.5% of the total symbol time is used for data transmission. The cumulative effect of these inefficiencies impacts the final data rate realized. In addition, there is a data efficiency loss due to the error correction coding used in the DSP portion of the radio. The achievable data rate through the physical portion of the WiMedia MB-OFDM transceiver can be calculated from:

$$Data\ Rate\ (bps) = \frac{1}{symbol\ period} \cdot \#data\ carriers \cdot \frac{bits}{sub\ channel} \cdot coding\ rate \quad (1.14)$$

where the symbol period accounts for the time domain efficiency, the data carriers account for the frequency domain efficiency, the coding rate accounts for the error correction encoding and the bits per sub-channel accounts for the spectral efficiency of the input symbol used, i.e. QAM or M-ary PSK.

Since WiMedia MB-OFDM uses a 312.5ns symbol period, with 100 data carriers each carrying 2-bits information, and an error correction coding rate of $\frac{3}{4}$, the maximum system data rate using Equation 1.14 is calculated to be 480 Mbps. Since 160 samples are passed in the 312.5ns time, the sample rate is 528 MS/s.

Several other lower data rate options are also included in the WiMedia MB-OFDM specification that increase coding redundancy and increase transmission distance. With nominal indoor multi-path models the system is expected to achieve 480 Mbps at 4 meters and 110 Mbps at 10 meters [15]. Regardless of data rate, the FFT remains

128-point, and the sample rate remains 528 MS/s.

The primary limitation in transmission distance of the WiMedia MB-OFDM system arises from the FCC restriction that UWB devices transmit with a power less than -41dBm/MHz . This translates to a maximum average transmitted power of -10.3 dBm and a maximum average expected receiver power of -40.3 dBm . The expected minimum receiver signal power is -80.5 dBm at 100 Mbps and -73.2 dBm at 480 Mbps. The difference between the maximum power of -40.3 dBm and the minimum power of -80.5 dBm is only 40 dB which represents a shift in emphasis for receiver design, as architectures no longer need to provide the large dynamic ranges (e.g. $> 80\text{dB}$) typically required for narrowband wireless communications systems covering much longer transmission distances.

1.2 Architectural challenges in UWB OFDM transceivers

Figure 1.15 shows the block diagram of a typical OFDM transceiver. The data transmission process, Figure 1.15(a), begins with baseband data from the media access controller (MAC) being formatted in the forward error correction (FEC) encoder to ensure the lowest possible error rate. This process includes removing long streams of continuous zeros or ones, interleaving to counter burst errors, and forward error coding to add parity or redundancy to the data in order to be more robust against transmission errors.

The error corrected data bits can then be mapped to the either M-ary phase shift keying (PSK) or higher order quadrature amplitude modulation (QAM) constellations depending on the required signal to noise ratio (SNR) at the receiver. WLAN 802.11a systems use QAM constellations, and require a high receiver SNR. Since WiMedia MB-OFDM is oriented toward wide bandwidth at low SNR, it can employ a digital modulation that does not require as high an SNR such as QPSK.

The phase and/or amplitude modulated symbols are converted from a serial data stream into parallel streams (S/P) that are then mapped to frequency sub-carriers by the IFFT processor. From the parallel outputs of the IFFT processor, the cyclic

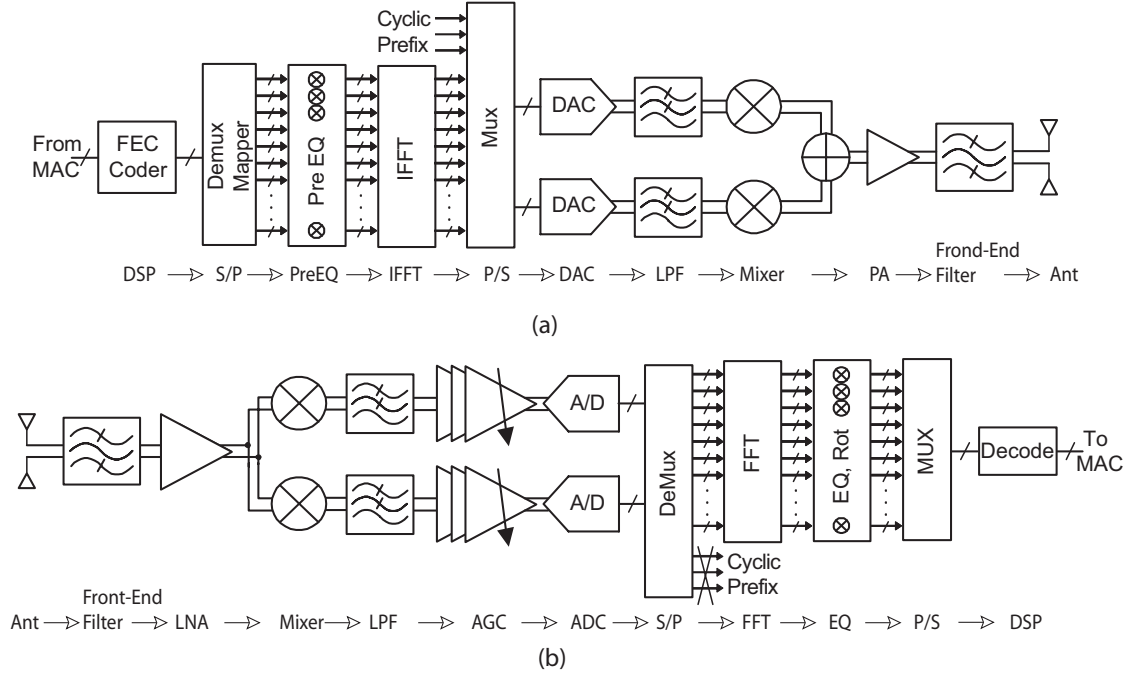


Figure 1.15: Block diagram of a direct conversion OFDM transceiver. (a) Transmitter data path, (b) Receiver data path

prefix is added in the multiplexer. This forms a serial “mini-packet” referred to as a single ‘OFDM symbol’.

Finally, the OFDM symbols are passed through quadrature (I/Q) digital-to-analog converters (DACs) and up-converted in the RF transmitter to the desired band frequency. The DAC is typically clocked at a higher rate than the data, providing over-sampling with rates between 600 MHz and 1024 MHz. It should be noted that the carrier frequency (LO) generation for UWB OFDM transmitters is an area of active research, but is beyond the scope of this work.

Once transmitted to the air channel, the OFDM sub-channels are distorted and attenuated by propagation loss. The RF receiver, as seen in Figure 1.15(b), receives the symbols, down-converts them in quadrature to baseband, and passes them through the channel filters and IF automatic gain control (AGC) amplifiers.

At this point, the received baseband signal containing the OFDM symbols, and any interference not removed by the filters, is passed through the analog-to-digital converter. The serial-to-parallel block converts the quadrature I and Q data to parallel

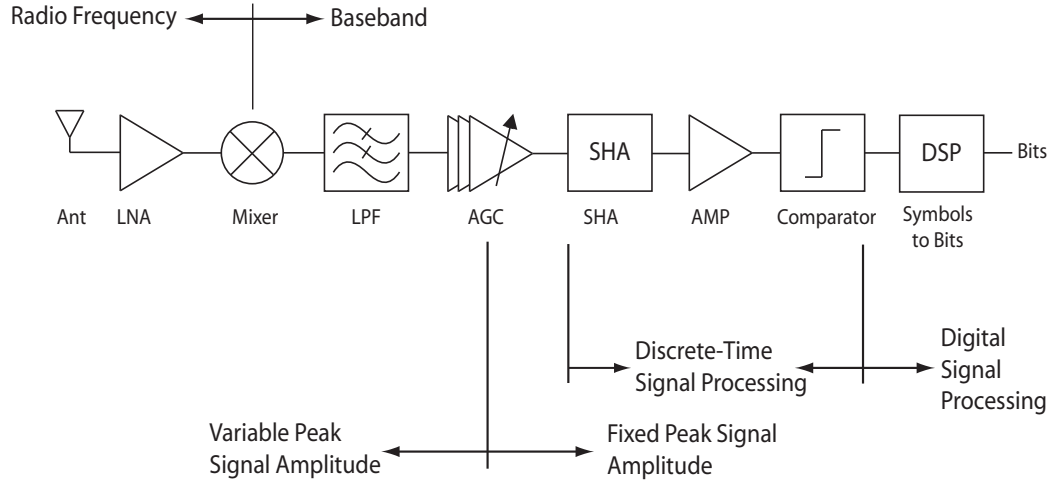


Figure 1.16: The receiver RF front-end, baseband, analog-to-digital conversion and DSP are represented by different signaling domains: continuous-time versus discrete-time and variable signal amplitude versus fixed signal amplitude. Although OFDM receivers are typically quadrature, only one baseband path is shown for simplicity.

complex samples and removes the cyclic prefix. The FFT block demodulates the sub-carriers, resulting in received QPSK symbols. Because each sub-carrier is distorted independently during transmission, the sub-channels each have a phase rotation and attenuation that must be corrected for in the equalizer. The equalization process involves multiplying each sub-channel by a gain and phase correction derived from measurement of the pilot sub-carriers. The equalized symbols are finally decoded in the error correction and decoder block and passed to the receiver MAC.

1.2.1 Performance Metrics for Wireless Receivers

To further analyze the OFDM receiver, the receiver can be sub-divided in terms of signal processing function. Figure 1.16 shows the stages of a simplified version of the direct conversion receiver from Figure 1.15(b). The analog-to-digital converter has been expanded into its basic components: sample-and-holds (SHAs), amplifiers, and comparators. In order to analyze system design trade-offs, it is important to understand the differences between the signaling domains, the functions of the receiver stages, and the definitions of their performance specifications.

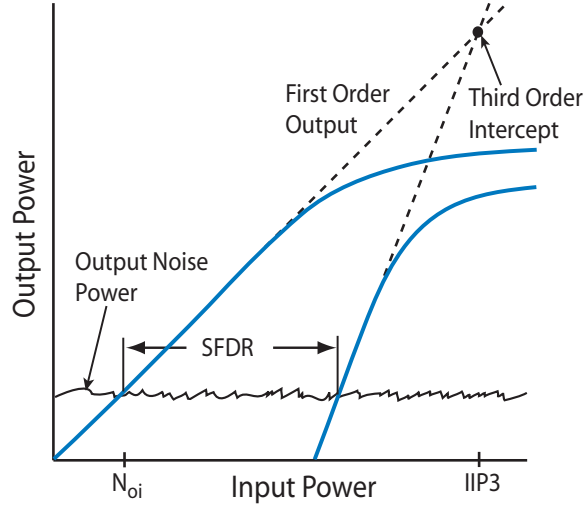


Figure 1.17: Front-end spurious free dynamic range is calculated from the input referred third-order intercept point and the input noise power.

The RF circuitry consists of low-noise amplifiers and mixers. These amplify and down convert the RF signals received at the antenna. If the magnitude of the receiver signal is small, large blocking signals can saturate the LNA causing corruption of the small received signal. Thus, the specifications for optimal receiver front-ends focus on simultaneously minimizing noise figure and maximizing the input third order intermodulation intercept point (IIP3). The spurious free dynamic range (SFDR) can be expressed for the receiver front-end stages by Equation 1.15 [16].

$$SFDR_{RF} = \frac{2}{3} (IIP3 - N_{oi}) \quad (1.15)$$

where N_{oi} is the input noise power. This equation is based on the assumption that the largest spurs in the system will arise from third order intermodulation and that the non-linearity can be expressed in terms of a 3rd order power series. Figure 1.17 shows how the SFDR is represented graphically on the output power versus input power curves for the fundamental and third order intermodulation distortion of an amplifier.

After the target receive signal has been amplified and mixed to baseband in the RF front-end, a low pass filter, also known as the channel select filter, removes unwanted

interferer signals. The filter is followed by an automatic gain control (AGC) amplifier that amplifies the received signal so that the peak signal magnitude is set to full scale input of the baseband processing circuitry.

The AGC ideally acts as a signaling boundary between signals with an unknown peak amplitude and signals with a fixed peak amplitude. Once the receive signal's peak amplitude becomes fixed, signal to noise ratio (SNR) is subsequently used to describe the effects of noise on the signal. Since the signal power is large, usually just a few decibels below the compression point, distortion consists of many harmonic and intermodulation products. To represent these effects, the total distortion, or error, contributed by a stage is given by:

$$\text{Distortion Power} = (V_{out}(t) - V_{in}(t))^2 \quad (1.16)$$

where V_{in} and V_{out} are the input and output voltages of the stage. One method of specifying distortion when digitally modulated signals are employed is the Error Vector Magnitude (EVM), which is the RMS average of the distortion power:

$$EVM = \sqrt{\frac{1}{\tau} \int_t^{t+\tau} (V_{out}(t) - V_{in}(t))^2 dt} \quad (1.17)$$

EVN is specified for a particular digital modulation scheme. The distortion power and the noise power can also be combined to define the Signal to Noise and Distortion Ratio (SNDR) [17].

$$SNDR = 10 \log_{10} \left(\frac{\text{Signal Power}}{\text{Noise} + \text{Distortion Power}} \right) \quad (1.18)$$

This definition of SNDR holds for both sinusoidal signals and digitally modulated input signals because the type of input signal is not defined. It is common to plot the SNDR as a function of input power as seen in Figure 1.18(a). Figure 1.18(b)

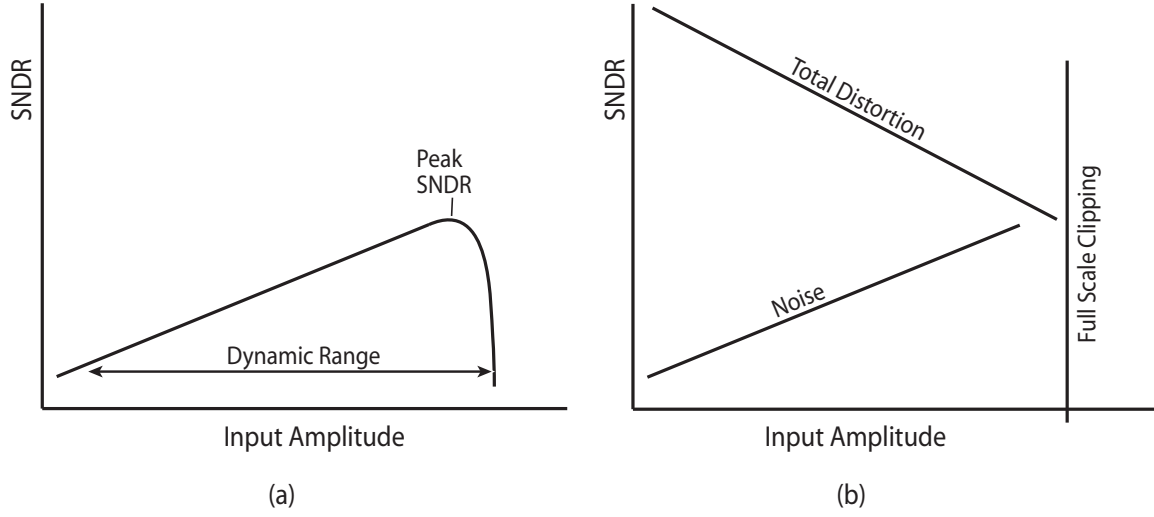


Figure 1.18: (a) The shape of the input amplitude versus SNDR plot for a typical circuit. (b) The three principal contributors, noise, distortion and clipping, that affect the shape of the typical input amplitude versus SNDR plot.

shows how the SNDR can be separated into the contributions from three factors. On a log scale, the SNDR increases linearly due to the contribution to noise, and decreases linearly due to the contribution from distortion. At the full scale signal value, clipping occurs and the SNDR decreases rapidly. The input magnitude that produces the peak SNDR is the best input signal level at which to operate the circuit. Thus, baseband circuits using sinusoidal inputs are typically designed to operate one decibel below the full scale value or 1 dBFS . Baseband circuits using signals with a large peak-to-average level are typically designed to operate backed off from the full scale value.

Another way to represent SNDR is with the effective number of bits (ENOB) [18]:

$$ENOB = \frac{(SNDR - 1.76)}{6.02} \quad (1.19)$$

This represents SNDR in terms of the number of bits required to achieve the same SNDR from an ideal ADC.

The *dynamic range* of a modulated signal can be calculated using the SNDR curve shown in Figure 1.18(a). It is defined as the ratio between the maximum detectable signal power and the minimum detectable signal power [19].

$$DR_{mod} = 10 \log_{10} \left(\frac{\textit{Maximum Detectable Signal Power}}{\textit{Minimum Detectable Signal Power}} \right) \quad (1.20)$$

The SNR required for a signal to be detectable varies based on the type of digital modulation used and typically ranges between 0 and 10 dB. This causes the DR are dependent upon the specified type of modulation.

The sample and hold amplifier (SHA) in Figure 1.16 acts as a boundary between two signaling domains. Prior to the SHA signals are continuous in time, after the SHA they are represented by discrete samples in time. Discrete-time signal processing is advantageous compared to continuous-time signal processing since techniques utilizing memory and pipelining are possible. This allows precise analysis of the behavior of the signal over time and offers the potential to perform z-domain filtering.

One drawback of discrete-time signal processing is that intermodulation distortion and harmonic terms are aliased or folded back into the discrete time frequency spectrum [20], as shown in Figure 1.19. Aliasing occurs for signals whose frequency exceeds half the Nyquist frequency, appearing to have a frequency within the sampled bandwidth. The effect, as seen in Figure 1.19, is that higher frequencies appear ‘folded’ back into the sampling frequency domain [14]. Because of this folding, close-in intermodulation terms are difficult to distinguish from high order intermodulation terms. Therefore, instead of using third-order intermodulation distortion to calculate SFDR, in discrete time baseband signal processing, the entire spurious response above the noise floor is considered using:

$$SFDR_{DT} = 10 \log_{10} \left(\frac{\textit{Signal Power}}{\textit{Largest Spurious Power}} \right) \quad (1.21)$$

SFDR also captures clock coupling, LO leakage and spurs from other sources that may couple into a circuit. Thus, SFDR is useful to quantify the worst case spur in a circuit. In flash ADCs, a rule of thumb is that SFDR is approximately 10 dB less than SNDR [19].

While a portion of the analog-to-digital converter is in the discrete-time domain, it’s

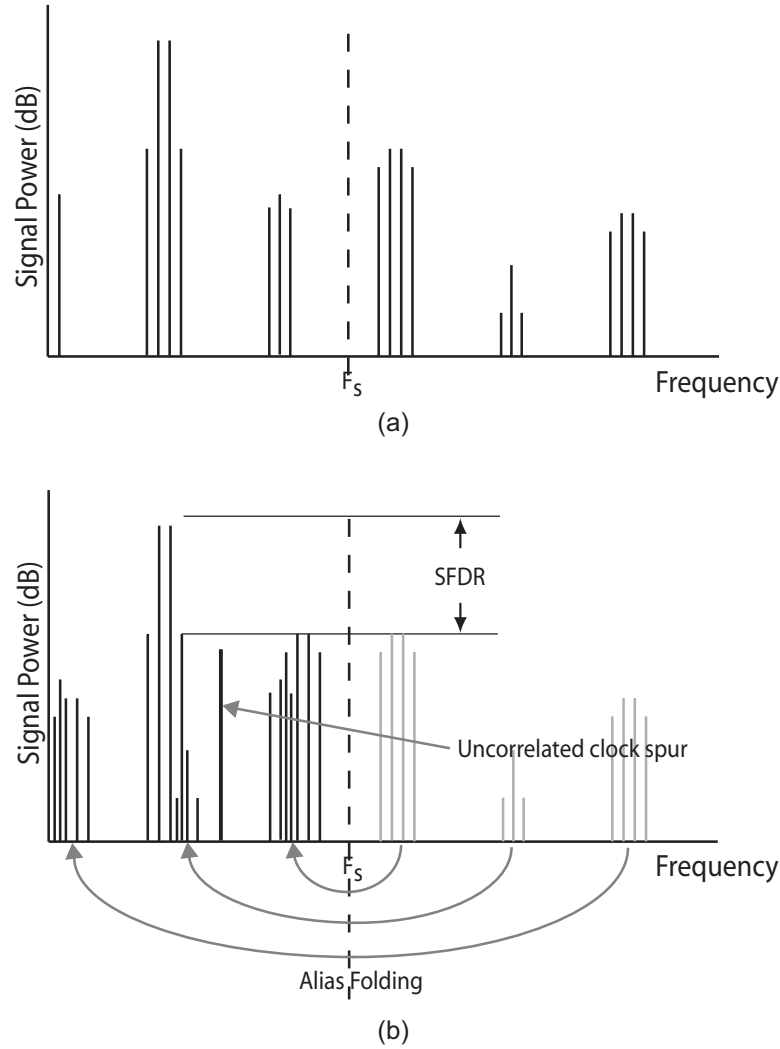


Figure 1.19: The non-linear harmonics and intermodulation harmonics resulting from a two tone test are shown for continuous time frequency spectrum in (a) and the discrete time frequency spectrum in (b). In the discrete time case, sub-sampling of higher frequency spurs causes them to ‘fold’ around the F_s point, into the lower frequency band.

output and subsequent signal processing are in the digital signal processing domain. This is shown as the last stage in 1.16. In the DSP domain, the real valued voltages from the discrete-time signal processing domain are quantized to bits. Signal processing is carried out through digital logic operation and the only noise contribution is from quantization. Thermal noise and non-linear distortion are no longer contributed to the signal.

Table 1.2: Performance of WiMedia MB-OFDM Receiver Front Ends.

Reference	Process Technology	IIP3 in High Gain Mode	NF	SFDR	Receiver Power Consump.
Ismail 2005 [21]	0.18 μ m SiGe BiCMOS	-19.5dBm	3.3dB	40.9dB	237mW
Chen 2006 [22]	0.18 μ m CMOS	-10.3dBm	5.8dB	45.4dB	81mW
Valdes 2006 [23]	0.25 μ m SiGe BiCMOS	-14dBm	6dB	42.8dB	285mW
Tanaka 2006 [24]	90nm CMOS	-16.5dBm	6.3dB	40.9dB	224mW
Shi 2005 [25]	0.25 μ m SiGe BiCMOS	-12.7dBm	6dB	43.7dB	84mW
Razavi 2005 [26]	0.13 μ m CMOS	-16.5dBm	6.5dB	40.8dB	105mW

Obviously, digital signal processing is the optimal domain for many types of signal processing; however, when high speed, high linearity, and low power consumption are critical, each of the other signal processing domains shown in Figure 1.16 has its merits. Dynamic range is one of the most important metrics in many stages to the receiver. In the next section, the dynamic range performance of several UWB OFDM RF front-ends will be examined.

1.2.2 UWB OFDM Receiver Front-Ends

A number of UWB OFDM receiver front-end designs can be found in the recent literature [21–26]. These works all use the direct conversion architecture to down convert the received RF signal to baseband. The total power consumption of the published RF front-ends varies between 81 mWatts and 285 mW, but for different levels of functionality. For example, in [25] only an LNA and direct-conversion mixer is reported with 84 mWatts of power consumption, whereas [24] includes the LNA, mixer, filters, AGC, VCO and dividers with 224 mWatts of power consumption reported.

Most of the receivers demonstrate a noise figure of approximately 6 dB which exceeds the WiMedia requirement of 4 dB. The only exception is [21], which reports a 3.3 dB noise figure. The noise level presented to the ADC, referred to the receiver input is given by [27]:

$$Noise\ Floor = -174dBm/Hz + 10\log BW + \frac{NF_{FE}}{A_{Filt}} + NF_{Filt} \quad (1.22)$$

where NF_{FE} is the receiver front-end noise figure, NF_{Filt} is the noise figure of the external band-select filter which is typically around $2dB$, and A_{Filt} is the gain of the external band-select filter (A_{Filt} of $-2dB$ corresponds to $2dB$ of NF). Since this is typically a lossy passive filter, the gain will be less than one. For MB-OFDM systems, the bandwidth is assumed to be 600 MHz due to low pass filter being required to exceed 512 MHz . Thus, for a 6 dB noise figure, the *Noise Floor* is $-78.2dBm$.

After calculating the Noise Floor, the dynamic range can be calculated from Equation 1.15. Using this equation, the SFDR values of the receiver front ends given in Table 1.2 are calculated. The typical value is between 40 dB and 45 dB which corresponds an SNDR of 50 and 55 dB or an ENOB between 8.1 and 8.9 -bits. Since the receiver presents an available dynamic range equivalent to approximately 9 -bits, the ADC should exceed this value so as not to add any further degradation to the received signal.

Given that current ultra-wide band transceivers process bandwidths of 500 MHz , the probability of encountering at least one narrow band blocker is quite high. In order to be robust against such interferers the largest practical system dynamic range should be used. Receiver front-ends share the system selectivity between three stages: the external pre-LNA band-select filter, an on chip baseband channel select filter, and the FFT processor. However, the front-end band select and baseband channel select filters only remove out-of-band blockers, leaving in-band blockers to be handled by the FFT processor. This means that the ADC must have sufficient dynamic range to linearly pass in-band blockers to the DSP-based FFT for removal in the digital domain. Because strong in-band blockers will saturate the automatic gain control loop (AGC) causing it to lower the receiver's gain to its lowest level, weak desired signals will not be sufficiently amplified, and will remain below the noise floor and be undetected.

Consider the link budget example shown in Figure 1.20(a). A link budget is a tool frequently used to allocate receiver front-end noise figure, P_{1dB} and dynamic range to various gain stages within the receiver. P_{1dB} can be approximately related to the IIP3 presented in Table 1.2 by the relationship $P_{1dB} = IIP3 - 9.6$ [16]. Here,

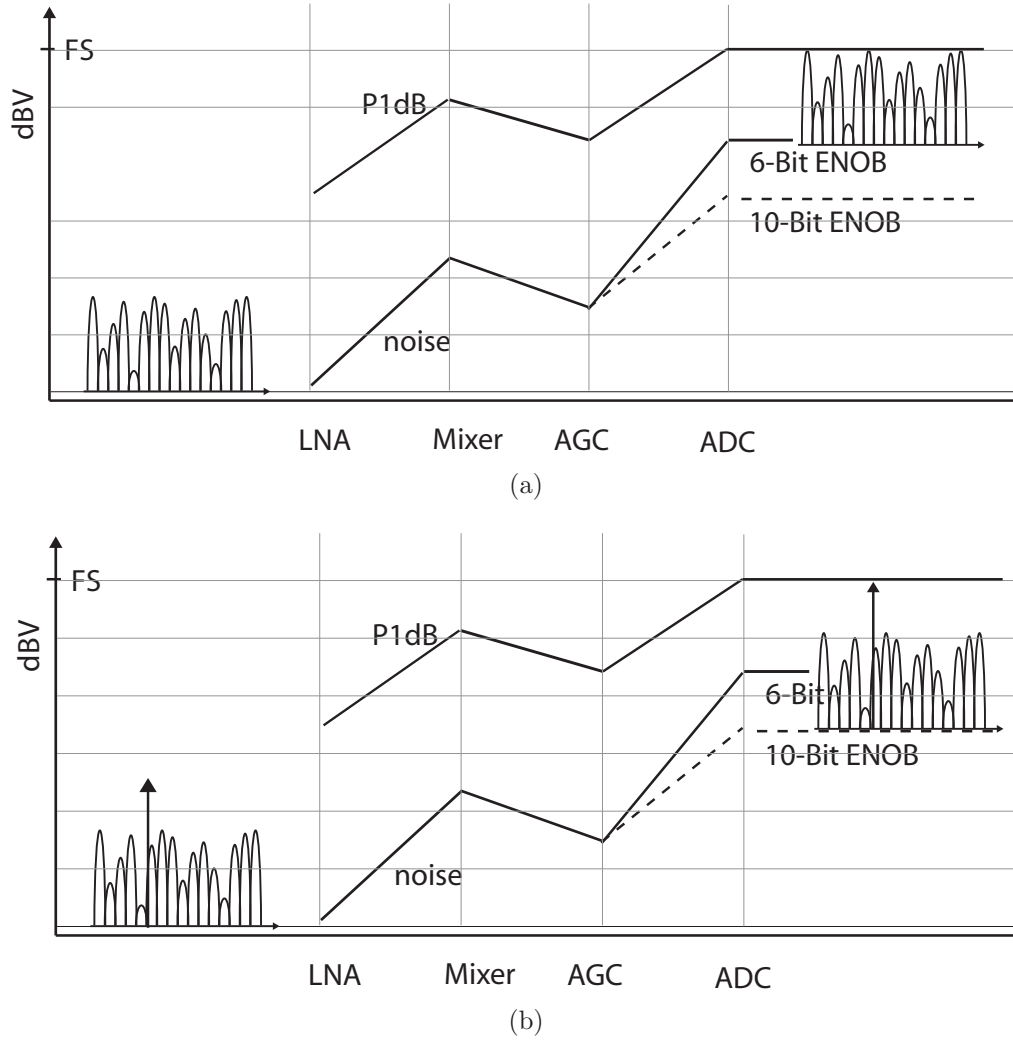


Figure 1.20: (a) The link budget of a receiver front-end and ADC shows the difference between the dynamic range of the 6-bit ADC and 10-bit ADC. (b) For the case of an in-band blocker, the dynamic range of the 6-bit ADC is insufficient and the weaker sub-channels are lost.

the link budget shows a received OFDM signal that is amplified by the AGC to set the strongest carriers to the full scale level of the ADC. The link budget carries 10-bits of dynamic range through the receiver LNA, mixer and AGC stages but the dynamic range is reduced at the 6-bit ADC. Nonetheless, all of the sub-channels are recoverable.

In the next example, shown in Figure 1.20(b), a strong blocker is included with the received OFDM signal. In this case the AGC can not fully amplify the OFDM signal

because it sets the blocker to the full scale level of the ADC. For the case of the 6-bit ADC, many of the OFDM sub-carriers are lost because they are below the quantization noise level of the 6-bit ADC input. However, if a 10-bit ADC is used, the OFDM signal could be fully recovered because there is sufficient dynamic range for the blocker and the OFDM signal. Only if sufficient dynamic range exists through the entire receiver chain, can the blocker be removed by digital filtering in the Fourier Transform.

1.2.3 Analog-to-Digital Converters for Ultra-Wideband Receivers

From the preceding discussion, the need for analog-to-digital converters for Ultra-WideBand receivers to provide adequate dynamic range is apparent. In addition, these ADCs should have a wide sampling bandwidth of at least 300 MHz, high speed with sample rates of at least 600 MSps and low power consumption. In this section, the state-of-the-art ADCs applicable to UWB systems are presented.

ADC Metrics

There are a number of important metrics used to compare ADC architectures for use in a wireless receiver. The most often quoted metric is the number of output digital bits, for example, a 6-bit ADC or an 8-bit ADC. The ENOB will be somewhat lower than the number of designed output digital bits, although for good designs the ENOB is 0.5 bits less than the number of output digital bits [18]. Another significant metric is sampling rate, F_s , the rate at which the ADC samples the continuous time input. In practice, the performance of the ADC deteriorates near one half of the Nyquist frequency. In wireless systems, the ADC sampling rate is typically specified to be at least 20% greater than the twice the required analog bandwidth [27, 28]. Therefore, for a WiMedia MB-OFDM signal with 528 MHz of RF bandwidth and 264 MHz of baseband bandwidth, the minimum sampling rate required would be approximately 600 MHz.

Understanding the target number of output digital bits and sample rate for an ADC allows architectural choices to be made. For high speed ADCs, the performance met-

rics, SNDR, ENOB and SFDR discussed in Section 1.2.1, are important. Typically, the SFDR is 8-12dB higher than SNDR. A Figure of Merit (FOM) frequently used in reporting ADC performance is specified as:

$$FOM = \frac{P_{diss}}{2^{SNDR_{bits}} F_s} \quad (1.23)$$

where P_{diss} is the dissipated power consumed by the ADC, $SNDR_{bits}$ is the signal to noise and distortion ratio in units of bits, and F_s is the sampling frequency. The units used to express the FOM are typically *picoJoules/conversion step*. Unfortunately, the frequency of the input tone used to measure ENOB, SNDR, SFDR and FOM is not specified, and therefore, reported results may be presented at a frequency “sweet-spot”. Thus, it is more useful to plot ENOB, SNDR and SFDR across a sweep of input frequencies which gives a more complete indication of performance, and many authors do include this.

Other common metrics for ADCs are Differential Non-Linearity (DNL) and Integral Non-Linearity (INL). Ideally, when digitization occurs, all of the quantization steps are of equal size. However this is not the case in practice because of circuit mismatches. DNL is the measure of the maximum difference between any two consecutive quantization steps. INL is the integral of DNL over many samples and represents the total deviation from the analog input to an ideal linear coded output value [19]. These quantities indicate the degree to which the digital codes represent the actual voltage for slow moving or static inputs. Values for both DNL and INL should be less than $\frac{1}{2}$ of a Least Significant Bit (LSB).

To better understand the state-of-the-art in analog-to-digital converters for UWB OFDM, it is helpful to review the literature for ADCs with dynamic ranges between 6-bits and 10-bits of resolution and sample rates greater than 600 MHz.

The ADC bottleneck

Table 1.3 compares the performance of high-speed ADCs found in the literature with sample rates exceeding 600 MSps, the minimum rate needed to digitize a baseband WiMedia MB-OFDM signal. All of the ADCs reviewed are in CMOS technology

Table 1.3: High Speed Analog to Digital Converters suitable for UWB OFDM.

Reference	Process CMOS	Type	Sample Rate (MSPS)	Power (mW)	ENOB	FOM
Gupta 2006 [29]	0.13 μm	Folded, Interp.	1000	250	8.35	0.8
Taft 2004 [30]	0.18 μm	Folded, Interp.	1600	774	7.20	3.3
Sander 2005 [31]	0.13 μm	Flash	1200	130	5.02	3.3
Shen 2007 [32]	0.18 μm	Pipeline Interp.	800	105	5.02	4.0
XiChen 2001 [33]	0.13 μm	Flash, Interp.	2000	310	4.77	5.7
Sholtens 2002 [34]	0.18 μm	Flash	1600	328	5.00	6.4
Choi 2001 [35]	0.35 μm	Flash	1300	545	5.19	11.5
Yu 2001 [36]	0.25 μm	Flash, Interp.	900	450	5.19	13.7
Utten 2003 [37]	0.25 μm	Flash	1300	600	4.86	15.9
Paulus 2004 [38]	0.13 μm	Flash	4000	990	3.69	19.1

(SiGe and Bipolar based converters are fast, but typically not a good technology choice for higher-bit depths due to their large area, power consumption and cost). Typical power levels are several hundred milliwatts with ENOBs of approximately 5-bits. A noted exception, [29], has an ENOB of 8.4-bits, but has many large spurs which effectively reduces the usable dynamic range to approximately 6-bits. This cause of the spurs is the interpolating architecture which is known to limit dynamic range [19]. These results show that typical power levels of 300 mWatts and less than six effective number of bits can be expected from today's leading CMOS ADCs capable of UWB application.

In [39–41] the state of high speed analog-to-digital converter technology is reviewed and historical trends are analyzed. In [41] it is shown that the product of ENOB and sample rate doubles every 5.7 years for flash ADCs. In [40] it is shown that the product of ENOB and sample rate doubles every 5.3 years for all ADC architectures. In both cases, this performance growth rate suggests that wide dynamic range ADCs of 10 bits of digital output or greater for WiMedia MB-OFDM will not be available for more than a decade. In the meantime, digital processing capability continues to grow

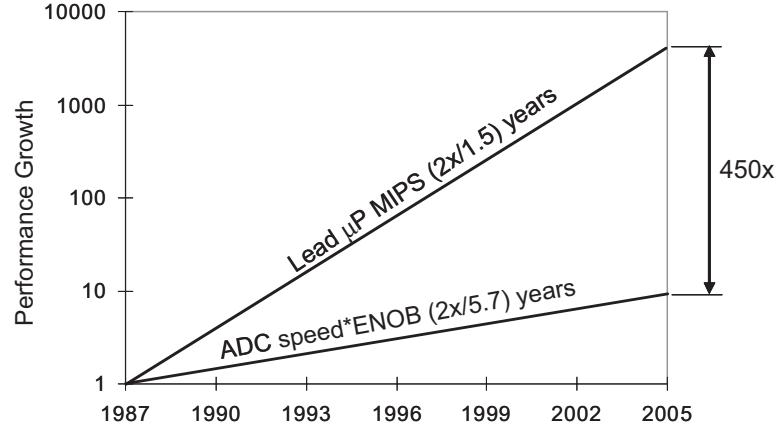


Figure 1.21: Moore’s law shows microprocessor performance growth doubling every 1.5 years. Meanwhile, flash ADC performance is doubling only every 5.7 years.

following Moore’s law, which asserts that digital logic doubles in speed power metric every 1.5 years. Figure 1.21 illustrates the resulting divergence between Moore’s law and anticipated ADC performance.

Given the outlook for growth in high speed, wide dynamic range ADCs, it is necessary to consider alternate architectures for OFDM that allow the FFT to filter narrow-band blockers without the need for wide dynamic range ADCs. In the next section, examples of WiMedia MB-OFDM capable FFT processors are reviewed.

1.2.4 State-of-the-Art Digital FFT Processors for UWB OFDM

The Fast Fourier Transform (FFT) is the critical signal processing function in OFDM system. The FFT provides digital filtering between sub-channels and is the source of orthogonality between them. As discussed in Section 1.1.2, the individual OFDM sub-channels contain PSK or QAM modulated data mapped to sinusoidal sub-carriers. Thus, it is essential that the Fourier transform processor independently resolve each sub-carrier and its magnitude and phase with minimal distortion.

There are many digital signal processing architectures that mathematically implement the Fast Fourier Transform [42–48], each with specific features designed for

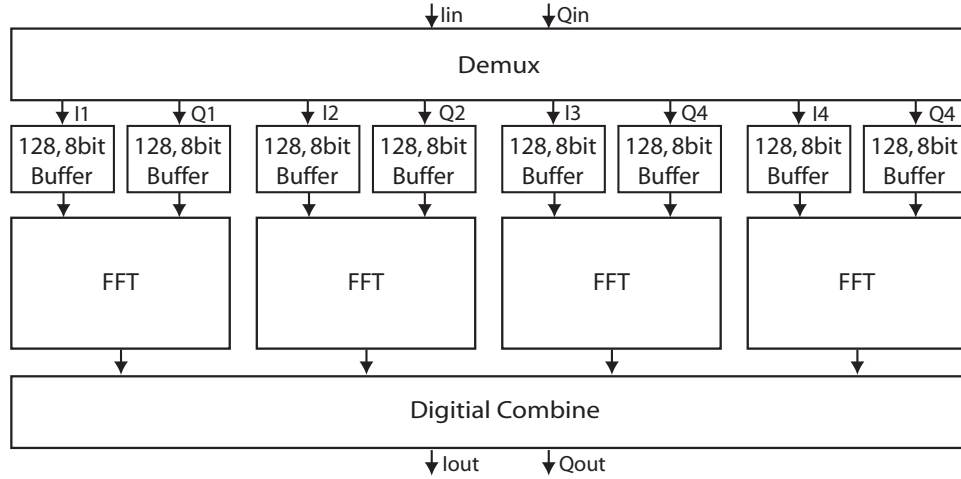


Figure 1.22: Parallelism is used to achieve the 409.6MSps data rate required of digital FFT processors for WiMedia MB-OFDM.

the target signal processing application and the technology used to implement the hardware. In the previous generation of OFDM receivers for 802.11a wireless local area network (WLAN) standard, the 17 MHz baseband bandwidth and 64-point FFT requirement permitted low power solutions operating at less than 40 MSps. Digital implementation was straightforward because the digital logic could be clocked at a much higher rate than the required sample rate. Several FFT processor architectures have been proposed and/or built based on the presumption of logic rates exceeding the symbol rate [45, 47, 49, 50].

As an example, [45] presents a WLAN FFT processor that operates on a 20 MHz clock while consuming only 41 mWattsof power. The approach breaks the 64-point FFT into smaller matrix blocks that can be processed by the two 8-point FFT processors, and an 8-point parallel phase rotation block that shifts the phase of set of symbols before returning them to memory. Each of the 8-point FFT processors contains twelve 16-bit complex constant multipliers and seven complex multipliers. 56 16-bit registers, hold intermediate results from the two 8-point processors, and 22 clock cycles are used to fully process a single 64-point FFT. The only drawback to the design, is the large amount of chip area required, 6.8 mm², primarily due to the large number of digital multipliers used.

In contrast to WLAN, WiMedia MB-OFDM requires a much higher symbol rate

of 409.6 MSps for the FFT processor which heightens the need for some sort of parallelism. [42–44, 46, 48] present FFT processors that achieve their high sample rate through time-interleaving multiple processors. Figure 1.22 shows an example signal flow diagram used to reduce the data with eight buffers and four parallel FFT processors. The large amount of digital hardware consumes power and additional chip real-estate. [44] uses two parallel data paths and [42, 43, 46, 48] use four parallel data paths to achieve the WiMedia MB-OFDM data rate. Power consumption levels range from 77 mWattsto 575 mWattsand gate count varies by architecture. Although the architectures reviewed are each unique, they all basically use an approach that calculates a small portion of the FFT, and stores the result in a memory buffer, working the overall FFT problem in parts. The best performance is displayed by [48] which functions up to clock rates of 250 MHz to achieve an FFT processing rate of 1 GSps through four times interleaving. The processor uses 3 mm² of die area and 175 mWattsof power. Although parallelism is effective in achieving the target data rate, the inefficient use of die area and high power consumption drive the development of an improved architecture OFDM systems requiring even higher data rates. In the next section, discrete-time analog signal processing is suggested to be an area and power efficient signal processing technique for multiplication intensive applications.

1.3 UWB baseband processing using discrete-time Analog Signal Processing

Discrete time signal processing has been used in the past in improve efficiency in a wide variety of multiplication intensive applications. One of these areas is in the implementation of Finite Impulse Response (FIR) filters. Typical applications of ASP FIR filters are equalizers, correlators and filters. In the past ten years, extensive research was done in applying analog filters to computer hard drive magnetic read channels as adaptive equalizers that implement a partial response maximum likelihood (PRML) algorithm [51–56]. The discrete-time ASP filters typically run at 100-200 MSps and are between 7th and 15th order filters. For PRML detection speeds above 200 MHz, continuous filters are used rather than discrete-time FIR filters [51, 54].

In [57] an analysis of the trade offs involving analog versus digital FIR filters for

CDMA correlators is presented. The authors note that analog correlators are typically applied in high speed, low precision applications, whereas, digital correlators are typically applied in high precision, high complexity applications where the speed requirements are lower. The work shows that as signal processing rate increases relative to transistor f_t , analog processing becomes more power efficient than digital signal processing. The results also show that although the power consumption of the analog correlators is considerably less than a digital implementation for low filter order, the power consumption grows at a quadratic rate with filter order for the analog implementation but grows more linearly for the digital implementation. Thus based on technology, filter order, speed and precision requirements, there will be a clear advantage between the one of the two implementation methods. The authors suggest that for each application, simulations should be performed to compare the two approaches to determine whether an analog or digital implementation is appropriate. The paper shows that for lower filter orders, less than 100, analog implementations are more power efficient than digital implementations.

In [58–60] analog FIR filters are used for channel select filtering in sub-sampling receivers. In [59] an eight tap FIR filter that operates at 230 MSps and consumes 37 mWatts is presented. The filter is able to tolerate large interferers in the stopband with little distortion. In [58], a sub-sampling receiver is presented that includes a 23-tap analog FIR filter which consumes 47 mW.

In [61,62] analog FIR filters are used as equalizers to correct the frequency response of copper backplanes for wireline digital communications. [62] presents a 6th order adaptive FIR equalizer that operates at 1 GSps and consumes 45 mW. [61] presents a 4th order adaptive FIR equalizer that operates at 2.5 GSps and consumes 95 mW. In each of these examples the power efficiency realized by using ASP is significantly higher than the pure DSP equivalent.

Another area of active research in high speed analog signal processing is the development of Viterbi, turbo and Low Density Parity Check (LDPC) decoders [63–66]. Decoders differ from FIR filters in that a more complex signal flow graph is used to represent the probabilities of different received symbol combinations. In [66], a 4 state, 115 Mbps Viterbi decoder is presented with a power consumption is 14.9 mW, which is approximately 1/3 the power consumption of the equivalent digitally implemented decoder. In [65], a 13 Mbps analog Turbo decoder is presented that consumes

185 mWattsof power. In [63] an LDPC decoder is presented that runs at 6 Mbps and consumes 5 mW. These works indicate that even complex signal flow graphs can be implemented in ASP.

1.4 Proposed OFDM Architecture

After reviewing the radio requirements of the UWB OFDM transceiver and the constraints placed on the ADC, it is clear that alternate receiver architectures must be explored that improves OFDM receiver/ADC performance in terms of both power consumption and dynamic range. Since analog signal processing is an attractive alternative to more power consumptive digital signal processing in other areas of communications, it is explored for UWB OFDM here.

In this dissertation, an alternate OFDM receiver architecture, in which the FFT processor is transferred from the digital domain and placed in front of the ADC, is proposed. Figure 1.23 shows the elements of the traditional OFDM receiver (a) and the proposed alternative (b). Relocating the FFT processor will allow a reduction in the total information conversion burden on the ADC, which will allow a lower bit-depth ADC to be used. Since power is known to double for each bit represented by the ADC, this can have a significant impact on overall power consumption. Placing the FFT processor ahead of the ADC will allow in-band blockers to be removed before conversion, and thus they will only impact the dynamic range of the FFT pre-processor rather than the dynamic range of the ADC. Thus, the proposed approach has the potential to significantly improve the dynamic range of OFDM receivers.

1.5 Dissertation Organization

1.5.1 Objective of Dissertation

The principal objective of this dissertation is to re-examine the baseband circuit architecture of the UWB OFDM receivers in search of a more power efficient architecture. Previous success in using analog signal processing techniques to augment power constrained digital signal processing suggests that an analog solution may prove ben-

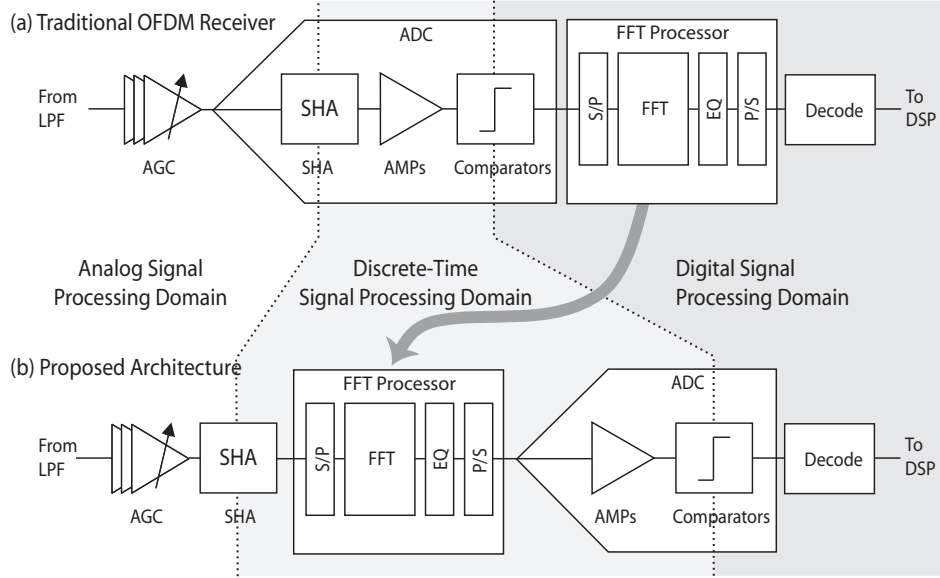


Figure 1.23: The block diagram of the baseband signal processing portion for a (a) traditional OFDM receiver and (b) the proposed modified OFDM receiver. Three different signaling domains separate the circuit functions.

eficial [42] - [66]. In addition, this research re-examines the dynamic range constraints placed on the current generation of UWB OFDM receiver by conventional analog-to-digital converters and considers alternatives. A new architecture that addresses both of these deficiencies is introduced and a prototype CMOS integrated circuit design is implemented to validate the functional legitimacy of the solution.

1.5.2 Outline of Dissertation

This chapter has presented OFDM and its utility for achieving high data rates in the UWB indoor wireless channel. After introducing how OFDM symbols are generated, the WiMedia MB-OFDM standard was described. The functional blocks of the typical transceiver were shown in Section 1.2 and a review of published UWB radio front-ends followed. Analog-to-digital converters capable of digitizing an WiMedia MB-OFDM signal were reviewed and issues of low dynamic range and high power consumption were identified. Examples of discrete-time analog signal processing were shown to be a potential alternative to digital systems for multiplication, intensive high-speed, low power implementations. Finally, a new alternative OFDM receiver architecture was proposed to improve the overall dynamic range and lower the power consumption of

ultra wide-band OFDM receivers.

In Chapter 2, an analog VLSI compatible form of the FFT is derived. The signal flow diagrams for the proposed discrete-time Analog FFT processor are also presented. In Chapter 3, system simulations are used to explore the capabilities of the proposed architecture and further quantify specifications for the transistor-level circuit design. Chapter 4 covers the CMOS circuit design of a first prototype analog FFT processor and shows CAD simulation results of its estimated performance. Layout issues are also presented. Chapter 5 shows the measurement results from the first prototype IC. Chapter 6 presents the circuit design of an improved version of the prototype IC. Chapter 7 discusses the results and presents future research building on this work.

Chapter 2

Discrete Time FFT Processor Architecture

In the first chapter, OFDM modulation was reviewed as a promising technology for ultra-wideband indoor wireless data transmission. The increased baseband signal processing demands for UWB OFDM receivers were discussed and the bottleneck in terms of power consumption and dynamic range for ADC performance was highlighted. To alleviate this bottleneck, an alternate OFDM receiver architecture, shown in Figure 1.23, was proposed, in which the Fast Fourier Transform processor is placed ahead of the ADC. In this chapter, the new discrete time FFT processor is described. The signal flow diagram of the discrete time FFT is first covered, followed by the individual functions required to implement processor.

2.1 A Discrete Time Signal Processing Compatible FFT Topology

In order to develop a Discrete Time FFT solution that has performance advantages over its digital counterpart, the signal processing must be optimized for a discrete time implementation. The principal circuit functions used in discrete time signal processing are multiplications, additions and sample-and-holds (memory). In discrete time signal processing, multipliers consume considerably less power and die real-estate

than those implemented in digital signal processing (DSP), a difference that will be exploited here. On the other hand, discrete time signal processing circuits can contribute noise and intermodulation distortion. Assuming unity gain for all circuit stages in a discrete time system, the total noise is the sum of noise contributions from circuits in the signal flow path. In addition to noise and non-linear distortion, if the 3-dB bandwidth of each discrete time signal processing circuit is insufficient, low pass filtering of the input signal will occur, reducing the leading edge of each non-repeated discrete time symbol and causing inter symbol interference. The effects of finite 3-dB bandwidth from cascaded processing stages are cumulative and can be approximated with the following equation [67]:

$$BW_{casc} = BW \sqrt[m]{2^{\frac{1}{N}} - 1} \quad (2.1)$$

where BW_{casc} is the cumulative bandwidth of N identical circuit stages with 3-dB bandwidth BW and m is the order of the equivalent low-pass filter. The value of $m = 2$ is for a first order low pass response and $m = 4$ is for second order low pass response. Although, the equation is pessimistic as it assumes that the equivalent filter poles are identical and therefor worst case, it is useful for generalized estimates. As an example, consider that three identical cascaded stages modeled by a first-order filter response have a combined bandwidth of 50% relative to the individual bandwidth of each stage. Seven identical cascaded stages modeled by a first-order response have a combined bandwidth of 33% relative to the individual bandwidth of each stage. Since increased bias current is typically required to increase bandwidth in analog circuits, bandwidth is directly related to power consumption. Thus, a discrete time FFT implementation that minimizes the number of cascaded amplifier paths may be the optimal solution.

2.1.1 The Fast Fourier Transform

There are several signal flow graphs that can represent the Discrete Fourier Transform (DFT) equation given in section 1.1.2 and repeated here for reference:

$$y_n = \sum_{k=0}^{N_{sc}-1} |x_k| \exp(j\angle x_k) \exp\left(\frac{j2\pi k_{sc}n}{N_{sc}}\right) \quad (2.2)$$

where $|x_k| \exp(j\angle x_k)$ represents the k th input sample, k_{sc} is the subcarrier position, n is the sample index, and N_{sc} is the number of samples.

The DFT signal flow graph (SFG) that is best suited to a discrete time implementation is the decimation-in-time FFT algorithm since it minimizes the number of cascaded stages and has symmetry between all signal paths. Figure 2.1 shows an example of a modified 8-point decimation-in-time FFT SFG. The decimation-in-time FFT algorithm is chosen because it orders the multiplication functions, placing the real valued multiplications first and the complex valued multiplications later in the SFG. Assuming that complex valued multiplications are more difficult to accurately perform, and therefore potentially introduce more distortion than real valued multiplication placing them later in the SFG reduces the potential for introducing signal errors. As can be seen in the figure, the SFG fully represents the Discrete Fourier Transform of Equation 2.2 for $N_{FFT} = 8$, and consists of the twelve cross-coupled signal flow pairs. Higher order FFTs can be similarly realized with the decimation-in-time algorithm and have N_{FFT} parallel inputs and $\log_2(N_{FFT})$ cascaded stages [14].

In Figure 2.1, inputs $x_0 - x_7$, are passed through multiplications (represented by triangles) and summations (represented by Σ) to the outputs $y_0 - y_7$. The multiplications are of the form $Y = c \cdot X$, where the coefficient c is a complex number with magnitude of 1 and angle $\frac{2\pi k}{N_{FFT}}$. Equation 2.3 can be used to calculate the coefficient values:

$$c_k = \exp \frac{j2\pi k}{N_{FFT}} \quad (2.3)$$

where N_{FFT} is the order of the FFT. As a discrete time sample passes through the decimation in time FFT SFG, it is added to a sample originating from the another of the parallel inputs. By the time a sample reaches the output, it contains contributions from samples originating from every input.

The SFG of the decimation-in-time FFT can be implemented in discrete time signal processing technology, by breaking the signal flow graph into a number of repeatable, simplified signal flow structures. This minimizes the number of unique circuits that

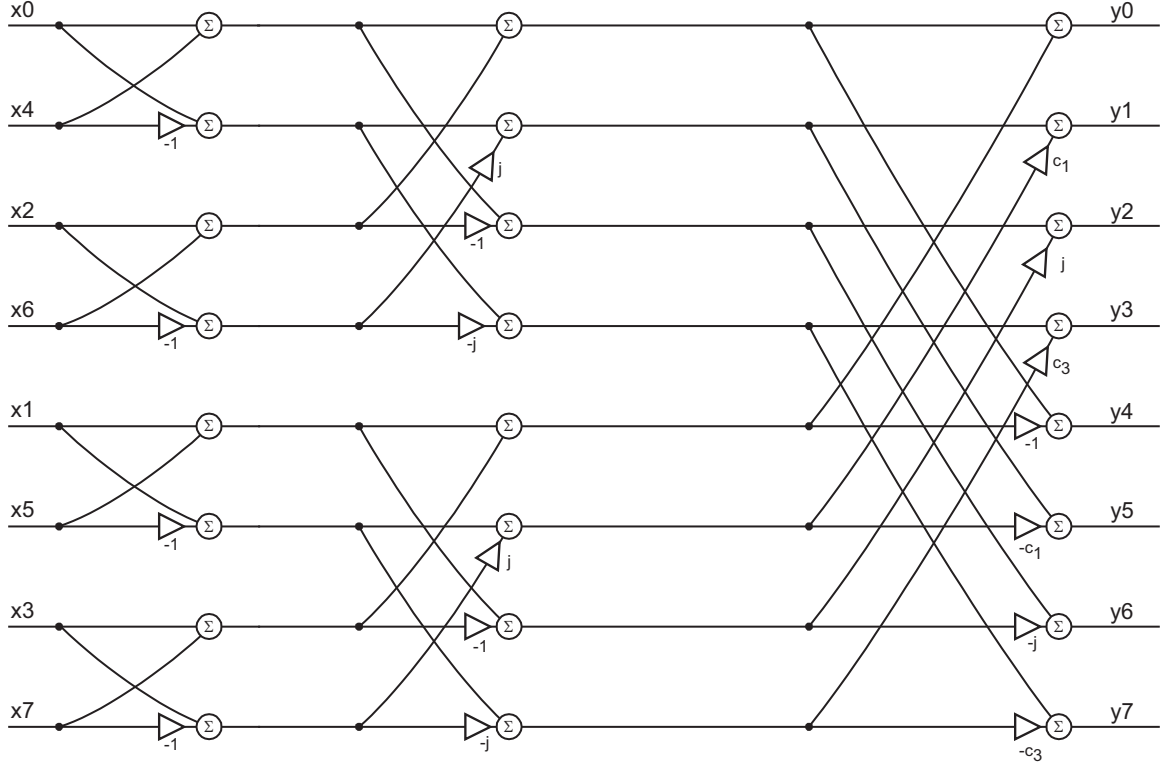


Figure 2.1: The signal flow lattice representation of an 8-point FFT.

need to be designed, reducing custom circuit layout time and possibly allowing an automated auto-routing layout to be used, and also, results in improved symmetry in the final design. Another goal is to allow each basic signal structure to be programmable, which allows for dynamic correction or adjustment of the final design post fabrication. On the other hand, because of the challenge of meeting the high target data rates of MB OFDM, simplifications that degrade operating speed must be weighed carefully.

A repeated operation in the decimation-in-time FFT SFG is multiplication by a coefficient and then summation with another signal. To simplify the SFG, the signal pairs can be grouped into a smaller two-input, two-output signal flow graphs known as a butterfly structure. Figure 2.2 shows an example of the butterfly structure in which, the X_B input is copied on to two paths, one multiplied by the coefficient c_k and the other $-c_k$ and each are then added to copies of X_A to create Y_A and Y_B respectively.

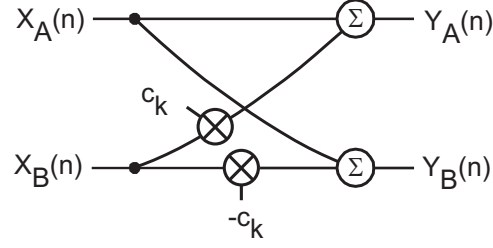


Figure 2.2: The signal flow diagram of the butterfly structure

Utilizing the butterfly structure as the basic unit cell, the signal flow graph in Figure 2.1 can be redrawn as shown in Figure 2.3. With this simple two-input two-output butterfly structure, any order N_{FFT} of discrete time FFT can be implemented. The number of butterfly structures required can be calculated by:

$$N_{\text{Butterflies}} = \frac{N_{FFT}}{2} \log_2 N_{FFT} \quad (2.4)$$

In the next section, discrete time signal processing methods will be applied to the implementation of the decimation-in-time FFT SFG to create the discrete time FFT processor suitable for demodulating OFDM signals.

2.2 The Proposed Discrete Time Analog FFT Processor

In this section, the complete discrete time FFT processor is presented based on multipliers, adders and memory elements compatible with discrete time signal processing.

Figure 2.4 shows the block diagram of the proposed discrete time FFT processor. The diagram consists of four primary functions: a serial-to-parallel converter that translates an OFDM input signal made up of discrete time samples into a bank of parallel samples; a discrete time butterfly structure based FFT signal flow graph which performs the FFT on the parallel input samples; a bank of equalizers that correct for attenuated sub-channels due to multipath; and finally a parallel-to-serial function that converts the FFT-processed parallel symbols back into a serial data

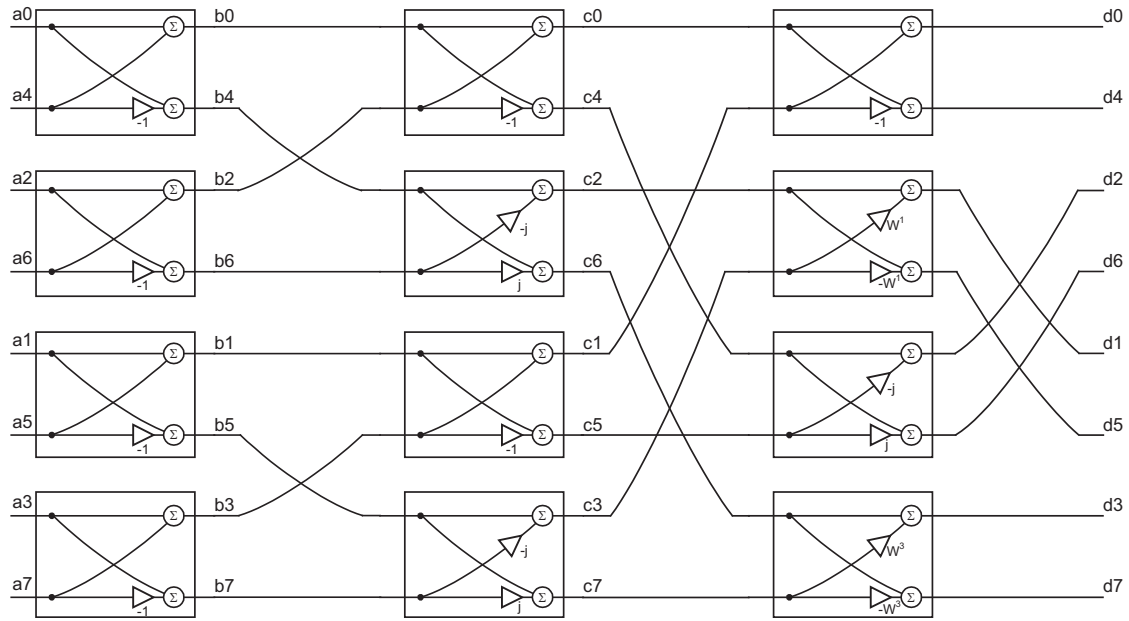


Figure 2.3: The FFT lattice shown in an discrete time signal processing compatible form stream.

The discrete time implementation of the FFT butterfly structure is first covered in the following subsection.

2.2.1 Discrete Time Butterfly Structure

The butterfly structure was introduced in section 2.1.1 and an SFG was presented for a discrete time FFT implementation. The butterfly structure is a two signal input, two signal output structure with the principal operation being to take two complex discrete time input signals, multiply one by a complex coefficient c_k , and then add the result to the other signal. A second output repeats this operation, but with the negative complex coefficient $-c_k$. Depending on c_k , the butterfly structure can be required to perform either complex multiplication or only real valued multiplication. Since the real valued butterfly structure is simpler, it will be described first.

Figure 2.5 shows the signal flow diagram of the real coefficient butterfly structure. Typical RF direct conversion receivers require the use of differential quadrature signal-

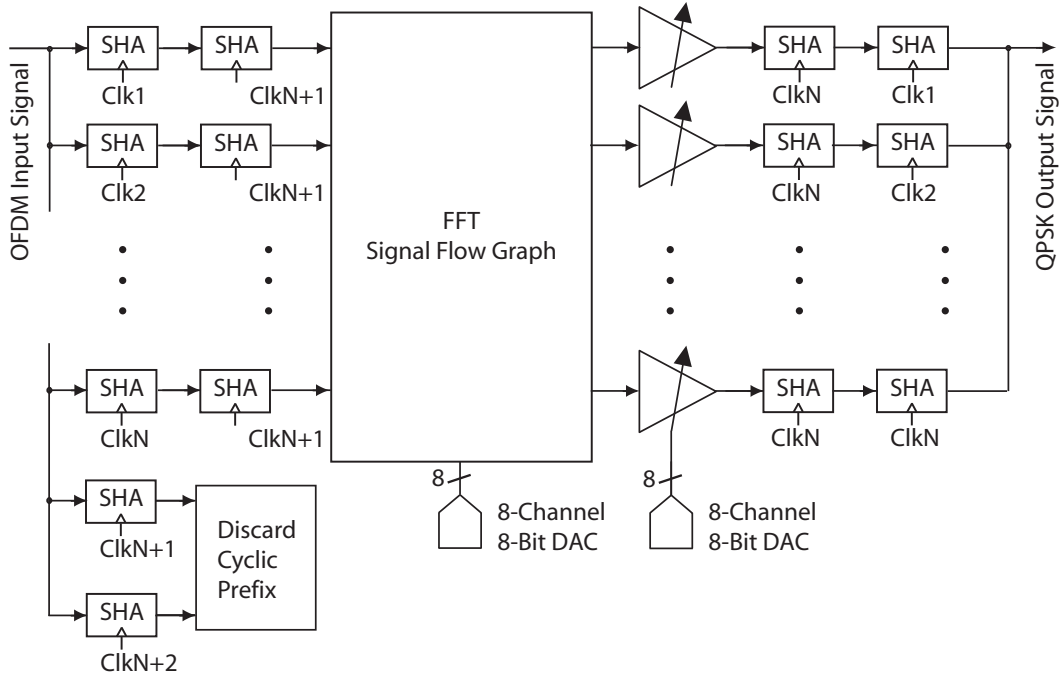


Figure 2.4: Block diagram of the proposed Discrete Time FFT processor

ing, thus the signal flow lines in these diagrams actually represent four physical wires in a hardware implementation. The subscripts I_p and I_n represent the differential I signal, or the real portion of the sample, whereas Q_p and Q_n represent the differential Q signal, or the complex portion of the sample. Within the butterfly structure, the voltage input signals X_A and X_B are converted by variable gain transconductors, or G_m cells, into current signals. Multiplication occurs in the G_m cell when the input signal is scaled in magnitude. The multiplication coefficient c_k controls the variable transconductance of the G_m amplifier. Signal addition is implemented in a transresistive addition circuit that converts the differential current signal back to a differential voltage output with a resistance value of R_{add} . To achieve a voltage gain of 1 from X to Y, the value of G_m can be calculated as:

$$G_m = \frac{|c_k|}{R_{add}} \quad (2.5)$$

For the real coefficient multiplier, c_k always has the magnitude of 1, but the phase can be 0° , 90° , 180° or 270° . When using differential transconductors, a 180° phase shift is easily achieved by swapping the differential output wires in the circuit. 0° , 90° , 180°

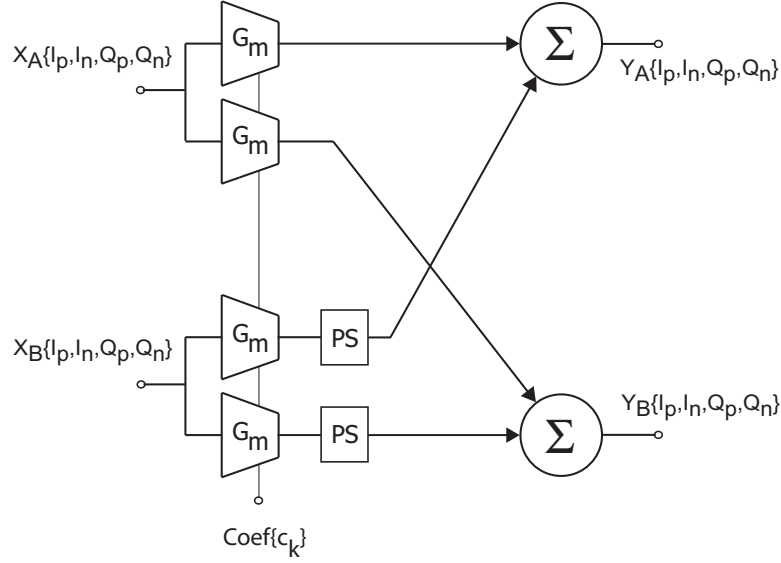


Figure 2.5: FFT butterfly circuit with hardwired coefficients constructed from transconductance amplifiers and current adders.

Table 2.1: The quadrature differential wiring of the PS block

	0°	90°	180°	270°
I_p	I_p	Q_n	I_n	Q_p
I_n	I_n	Q_p	I_p	Q_n
Q_p	Q_p	I_p	Q_n	I_n
Q_n	Q_n	I_n	Q_p	I_p

or 270° can be implemented using the I and Q differential outputs and routing the outputs through a hardwired phase shift block labeled PS. Table 2.1 shows how the four input wires can be re-routed to effect rotation of multiples of 90°. In layout, the PS block for each G_m cell is individually programmed for the multiplication quadrant required by its parent butterfly circuit. This allows the circuit design to be simplified so that fewer butterfly circuits are required. Using this technique, the real-coefficient butterfly circuit can effect the coefficient multiplication values of +1, -1, +j or -j.

For coefficient multiplication angles other than multiples of 90°, the complex coefficient butterfly structure, shown in Figure 2.6 is used. An extra pair of G_m cells and an extra current path for the Q_p , Q_n wires are required to allow for phase rotations of the input signal X_B . The transconductance values of the lower pairs of G_m cells following X_B are G_{m_c} and G_{m_s} . Equation 2.6 is used to determine the transconductance

values G_{m_c} and G_{m_s} .

$$\begin{aligned} G_{m_c} &= G_{m_0} \cos\left(\frac{2\pi k}{N_{FFT}}\right) \\ G_{m_s} &= G_{m_0} \sin\left(\frac{2\pi k}{N_{FFT}}\right) \end{aligned} \quad (2.6)$$

These are set by coefficient bias voltages C_c and C_s , which can be controlled by a bias DAC. An adequate number of bits is required to reduce quantization limits on the of the coefficient. For the case of $N_{FFT} = 8$, G_{m_c} , $G_{m_s} = 70.7\% G_{m_0}$.

Based on the radix-2 decimation-in-time algorithm and the multiplier implementation described, the total number of G_m cells and adder cells can be calculated to be:

$$N_{G_m} = 4N_{FFT} + 16 \sum_{k=2}^{\log_2 N_{FFT}} \frac{N_{FFT}}{2^k} + 12 \sum_{k=3}^{\log_2 N_{FFT}} \frac{N_{FFT}}{4} \quad (2.7)$$

$$N_{Adds} = 2N_{FFT} \log_2 N_{FFT} \quad (2.8)$$

For example, for an eight point FFT processor, 104 G_m cells and 48 add cells would be required; for a sixteen point FFT processor, 272 G_m cells and 128 add cells would be required. Thus the described method is best suited for low order FFTs. However, higher order FFTs can be implemented with an in-place computational method based on the re-use of a low order FFT core as described in [20].

For the circuit design of the G_m cells and adder circuits it is helpful to know the load drive requirements of the butterfly structure. The output of each butterfly structure must drive two G_m cells in a subsequent butterfly structure. Thus, the total capacitive load includes the input capacitance of these G_m cells in addition to the interconnect wiring capacitance,

$$C_{in} = 2C_{G_m} + 2C_{wire} \quad (2.9)$$

The drive requirements are also affected by the maximum signal voltage swing, V_{swing} , and the time it takes for the signal to transition between discrete time samples, T_r .

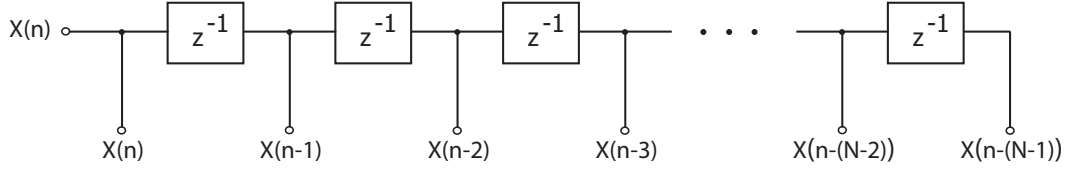


Figure 2.7: The z-domain representation of the serial to parallel function.

the two basic butterfly structures shown in Figures 2.5 and 2.6.

2.2.2 Serial-to-Parallel Function

The other important discrete time signal processing function required by the proposed FFT processor is the serial-to-parallel function. The input to the discrete time FFT processor is a serial stream of complex discrete time samples that must be collected and stored in memory to be simultaneously (parallel) processed by the parallel discrete time FFT SFG; i.e., in order to fully implement Equation 2.2, N_{FFT} samples are required. Figure 2.7 shows the z-domain signal flow diagram of the required serial-to-parallel function.

Sample-and-hold amplifiers (SHAs) can be used for implementing z^{-1} memory delay elements. SHAs acquire and track an input signal during one clock phase and hold the signal during the next clock phase.

A serial-to-parallel function with two sets of SHA banks is shown in Figure 2.9. The purpose of the first bank of SHAs is to sample the serial inputs and store the results for a short time period $T_s N_{FFT}$. The second bank of SHAs extends the hold time to $T_s N_{FFT}$ reducing the sample rate for ensuing stages. Table 2.2 summarizes the timing requirements of the serial-to-parallel bank conversion block where $T_s = 1ns$ for a $1GS/s$ sample rate. The maximum acquire and track time for the first bank of SHAs, $T_s/2$, is set by the period during which the discrete input symbol is valid. After sampling the input signal, each of the SHAs in the first bank must hold the symbol for a full OFDM symbol length, $T_s N_{FFT}$. Because the OFDM data contains a cyclic prefix that is discarded in the receiver, the symbol time occupied by the cyclic prefix can be utilized to ease the timing constraint on the second bank of SHAs. During the cyclic prefix, when no samples are being acquired by the first SHA bank,

Table 2.2: The Timing Requirements for the Serial-to-Parallel Function

	Spec	Proposed 1 GSps, $N_{FFT} = 8$
SHA bank1 Max acquire time	$T_s/2$	500 pS
SHA bank1 Min hold time	$T_s N_{FFT}$	8 nS
SHA bank2 Max acquire time	T_s	1 nS
SHA bank2 Min hold time	$T_s N_{FFT}$	8 nS

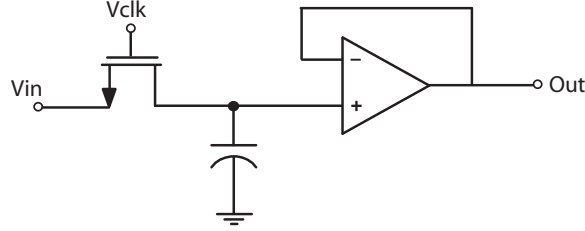


Figure 2.8: Open loop Sample and Hold

the second SHA bank simultaneously samples the output of the first bank. Thus the second bank of SHAs has a maximum acquire and track time of T_s , and a hold time of $T_s N_{FFT}$.

An open-loop buffer amplifier, as seen in Figure 2.8, is used at the output of each sample-and-hold to protect the charge stored on the memory capacitor. The open loop amplifier of the second SHA bank is followed by the input stages of the butterfly structures. The slew rate of this buffer can also be calculated using (2.11). The transition time between each discrete time samples, T_r is used for the value of ΔT . For example, for a 1 GSps system with a 400 mV voltage swing, ΔV , a 25 fF load capacitance, C , and a transition time of 200pS, the minimum slewing current is 50 μ A.

It is possible in an alternative form of the FFT signal flow graph to insert an additional bank of SHAs between each column of butterfly structures. This would further reduce the bandwidth requirement for the buffer amplifier and butterfly structures due to the cascaded bandwidth effect described in 2.1. The cost would be additional power consumption and die area for the additional hardware. Since the $N_{FFT} = 8$ case

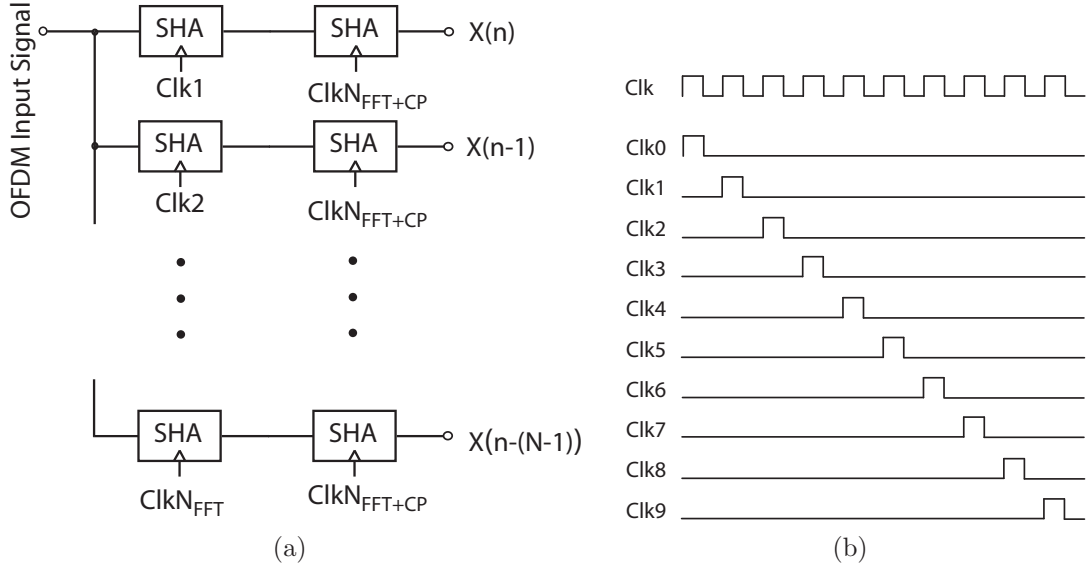


Figure 2.9: (a) The serial-to-parallel function realized with sample-and-hold amplifiers. (b) The clock timing diagram used.

only cascades 3 butterfly structures, this is not necessary. For a large N_{FFT} with significantly more cascaded stages, it may be beneficial to reduce the effect of cascaded bandwidth with additional SHA banks inserted with the FFT SFG.

2.2.3 Clock Generation

The serial-to-parallel function requires N_{FFT+CP} sequential clock signals to function. Figure 2.9(b) shows the required clock diagram for the example $N_{FFT} = 8$ case. Each of the clock signals Clk0 through Clk N_{FFT+CP} have short logic high pulse to drive the associated SHA into signal tracking mode and a long logic low pulse to hold the sampled signal over the time of a full OFDM symbol period, N_{FFT+CP}/Fs .

As long as there is a high speed SHA before the OFDM processor, the physical layout of the clocks is not critical, because the skew between phases is much less than 1 sample time. Otherwise, the clocks should all be synchronized with the master clock, Clk, avoiding any variation in path length that could cause time interleaved sampling errors.

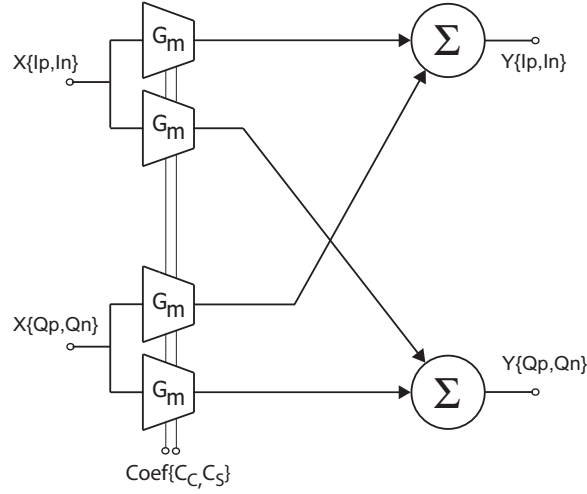


Figure 2.10: Signal flow diagram of one channel of the complex equalizer

2.2.4 The Discrete Time Sub-Channel Equalizer

Although the sub-channel equalizer is not a required function for the discrete-time FFT processor, it is necessary for an OFDM receiver implementation. The purpose of the sub-channel equalizer is to add gain to weak sub-channels that were attenuated by multi-path, and correct for phase shifts between the sub-channels.

The simplest implementation approach for this work is to reuse the G_m multiplier and transresistive adder from the real coefficient butterfly structure. Figure 2.10 shows an equalizer block diagram. A bank of these equalizers, N_{FFT} in total, one for each parallel output from the FFT signal flow graph implementation, is required. The equalizer approach uses Equation 2.12

$$\begin{aligned} G_{m_c} &= G_{m_k} \cos(\theta_k) \\ G_{m_s} &= G_{m_k} \sin(\theta_k) \end{aligned} \tag{2.12}$$

where G_{m_k} is the equalizing gain of the k th subchannel and θ_k is the equalizing phase of the k th subchannel. These values are determined in the digital signal processing portion of the receiver and fed back to the discrete time domain equalizer. Based on the applied values of G_{m_c} and G_{m_s} the input signal can be corrected in gain and

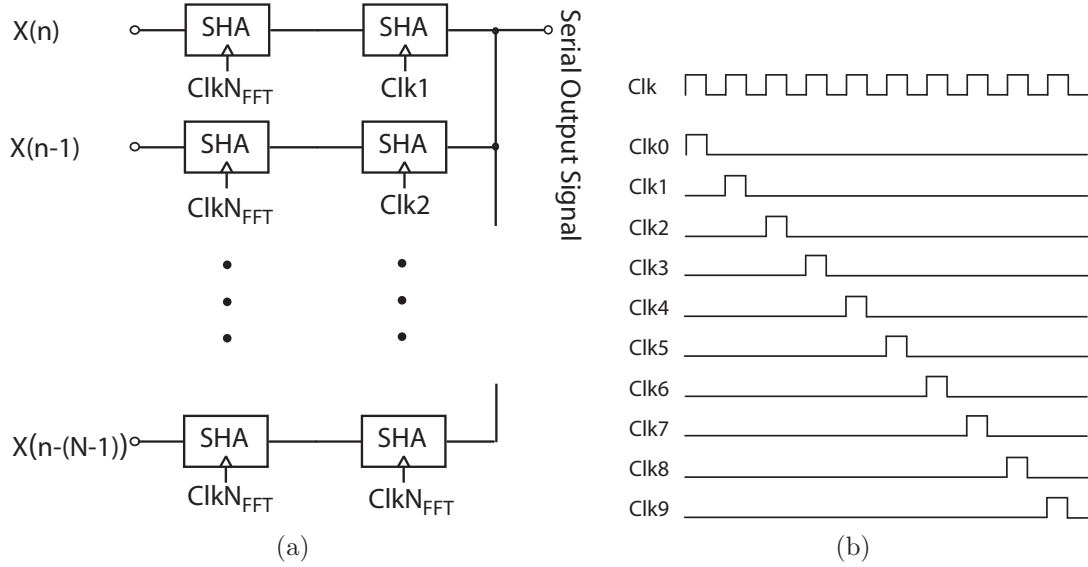


Figure 2.11: (a) The parallel to serial function realized with sample-and-hold amplifiers.
(b) The clock timing diagram used.

phase.

2.2.5 Parallel-to-Serial Converter

The final section of the discrete time FFT processor is the parallel-to-serial converter (Figure 2.11). This function reverses the process of the serial-to-parallel converter and uses the same clocks given in section 2.2.3. Once again, two SHA banks are used; the first is used to extend the time each symbol is held and the second is used to convert the parallel samples to serial samples at a higher rate. If the first bank of SHAs were not included, the butterfly structures in the FFT signal flow graph would have to run at the full sample rate F_s rather than at the reduced rate of $\frac{F_s}{N_{\text{FFT}}}$. The timing requirements for the parallel-to-serial function are given in Table 2.3. In this case, the first SHA bank runs slower at with an acquire time of T_s and the second SHA bank is faster with an acquire time of $T_s/2$. Because of the similarities to the serial-to-parallel function, the parallel-to-serial function can be implemented with similar SHA circuit designs.

Table 2.3: The Timing Requirements for the Parallel-to-Serial Function

	Spec	Proposed 1 GSps, $N_{FFT} = 8$
SHA bank1 Max acquire time	T_s	1 nS
SHA bank1 Min hold time	$T_s N_{FFT}$	8 nS
SHA bank2 Max acquire time	$T_s/2$	500 pS
SHA bank2 Min hold time	$T_s N_{FFT}$	8 nS

2.3 Summary

In this chapter, the architecture of an FFT processor compatible with discrete time signal processing was presented. The processor consists of several key functions: (1) the serial-to-parallel function; (2) the FFT signal flow graph; (3) the sub-channel equalizer; and (4) the parallel-to-serial function. Design considerations for each of these function were discussed. In the next chapter, behavioral system simulations of these functions are employed to explore the capabilities of the proposed architecture and to further refine the specifications for the transistor-level circuit design.

Chapter 3

System simulations of the DT FFT Processor

In this chapter, system simulations of the proposed DT FFT processor are presented based on behavioral models of the key circuits. The behavioral models are utilized to help construct the proposed DT FFT processor architecture and evaluate early design assumptions, as well as further define the performance requirements for the transistor level circuit design. With this approach, an optimal assignment of resources in the circuit functions of the processor can be performed.

Before introducing the behavioral models of the system, typical circuits used in discrete time signal processing applications are reviewed in the next section.

3.1 Discrete Time Signal Processing

Although there have been no discrete time signal processing based FFT processors in the literature to date, one discrete time signal processing application with some similarity to an FFT is the discrete time FIR filter [52, 55, 61, 62]. When used in receivers, discrete time FIR filters have many of the same requirements as the DT FFT processor – they must have wide dynamic range, add little distortion to the signal, and have coefficients that can be adjusted according to operating conditions. Figure 3.1 shows the schematic of the typical discrete time FIR filter, which uses

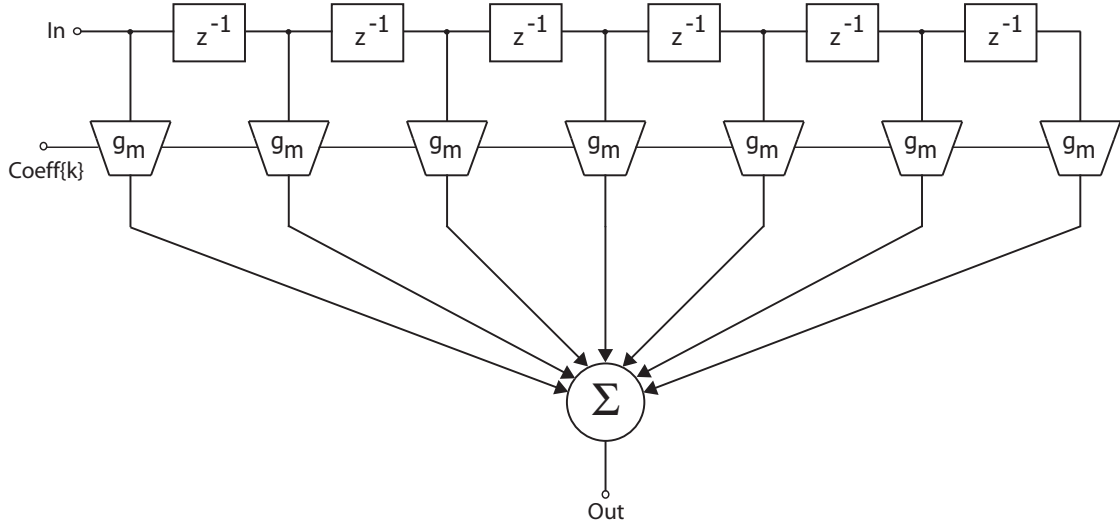


Figure 3.1: The typical schematic of a discrete time signal processing based FIR filter.

a mix of technologies, both discrete time and analog, to implement the filter. The discrete time memories (or delays) are represented by z^{-1} and add a single sample delay between their input and output. The coefficient multiplications operate on the samples, acting as analog variable gain amplifiers, scaling magnitude of the samples by the programmed coefficient value k_0-k_n . The variable gain amplifiers typically have a voltage input and current output and are thus transconductive amplifiers, making them compatible with a current domain addition function. The addition function uses a linear transresistance to sum multiple input currents and produce a voltage output.

3.1.1 Multipliers for use in Discrete Time Signal Processing

In discrete time signal processing, analog multipliers are used for their power efficient operation and compact size. Several types of analog multipliers are available: four quadrant multipliers, variable gain amplifiers and multiplying DACs. In FIR filters and in the proposed DT FFT processor, only coefficient multipliers are required so that the complexity of four quadrant multipliers is not warranted. Both variable gain amplifiers and multiplying DACs can be implemented as linear transconductors. The primary difference is in their method of control. The variable linear transcon-

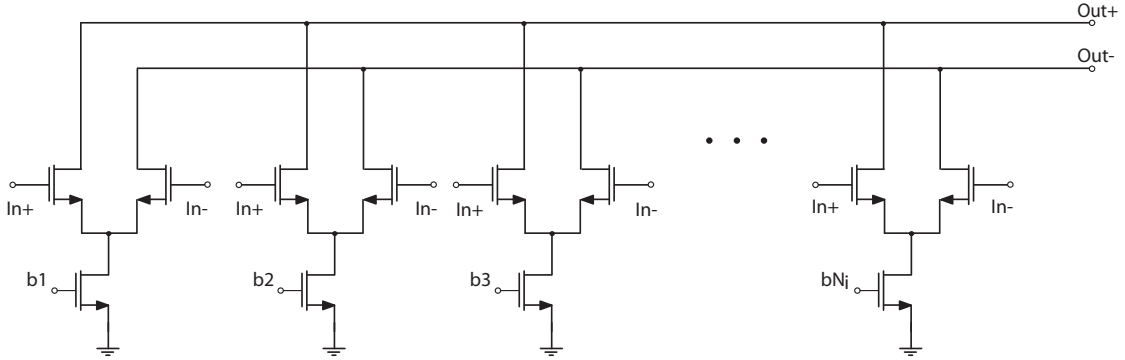


Figure 3.2: The differential pair multiplying DAC architecture. The current sources can either be binary weighted for a binary scaled DAC or equally sized for a segmented DAC.

ductor is controlled by adjusting a bias voltage or current; whereas a multiplying transconductive DAC is controlled by switching on or off some number of repeated unit transconductors. Ultimately, both are similar, assuming that the bias voltage or bias current of the variable transconductor is generated by a bias DAC. The primary difference is the location where the digital logic asserts change on the analog circuit.

One of the simplest analog multipliers is the differential multiplying DAC shown in Figure 3.2. In this circuit, differential pairs are used as the unit cells. The analog input signal is applied to the gates of the differential pairs and the tail current sources are switched on and off by digital logic. These unit cells are repeated with either binary scaling of the differential pairs or equal sized segmented differential pairs. Segmenting is a technique where the unit cells are all sized equally and interleaved in the layout, to ensure good matching. By laying out 2^N equal size unit cells, randomizing their locations, and assigning binary weighting to the number of unit cells each logic bit controls, a more linear DAC can be realized [68].

In [52], a 170 MHz discrete time FIR filter is implemented using a 6-bit differential pair multiplying DAC. The multiplying DAC uses 16 segmented differential pairs for the 4 most significant bits (MSBs) and binary weighted differential pairs for the 2 least significant bits (LSBs) resulting in a total of 18 differential pairs. A 3.3Volt $1.2\mu\text{m}$ CMOS process was used. Given the 3.3Volt headroom, the multiplying DAC is implemented with one difference from the circuit shown in Figure 3.2; it uses cascode current sources instead of single transistor current sources.

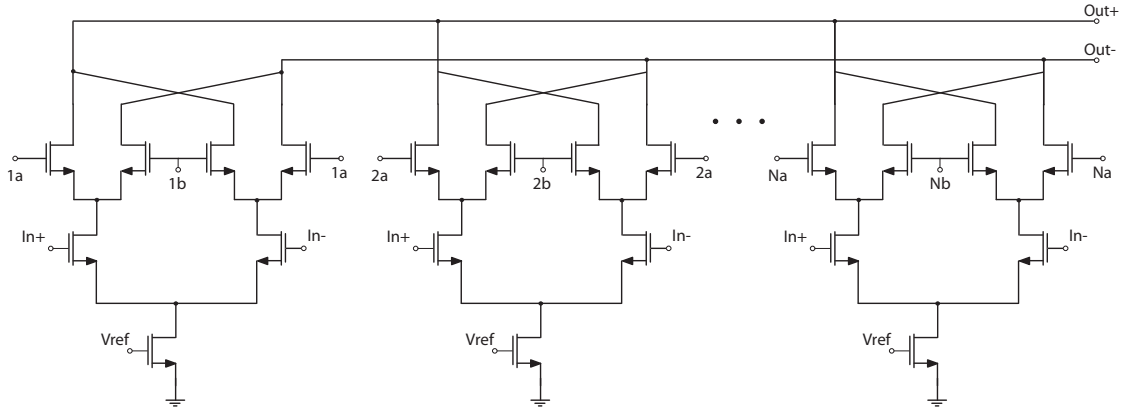


Figure 3.3: A multiplying DAC based on the Gilbert cell

In [55], a 6-bit transconductive multiplying DAC based on Gilbert multipliers (Figure 3.3), is used to implement a discrete time FIR based adaptive equalizer in $0.5\mu\text{m}$ CMOS. Six binary weighted Gilbert multiplier unit cells are used. The Gilbert multiplier “RF” inputs are fed by the input signal while the “LO” inputs are fed by the DAC. Binary weighting of the tail currents is used. The Gilbert multiplier allows for a G_m function that is linear over a wide range of input voltages. The drawback to this circuit is that three vertically stacked transistors consume considerable voltage headroom.

In [61], where speeds above 2.5 GSps are required, a simple pseudo differential transconductance multiplying DAC is used to implement a discrete-time FIR as shown in Figure 3.4. The circuit is essentially the same as the multiplying DAC based on the Gilbert cell, but with the tail current source removed. In high speed amplifiers, when the tail current source is removed the circuit is called, “pseudo-differential”. This approach is faster and requires less voltage headroom than the classic differential pair; however, this comes at the cost of linearity [69]. The results in [61] show a 400 mV linear range using a 2.5Volt supply in $0.25\mu\text{m}$ CMOS.

Besides the multiplying DAC, the other primary method of implementing coefficient multiplication is through variable gain transconductance amplifiers [65]. In this case, the coefficient is programmed by a bias current or bias voltage supplied by a low speed bias DAC. The approach of using variable gain transconductive amplifiers has the advantage that when coefficient values in the discrete time signal processing application

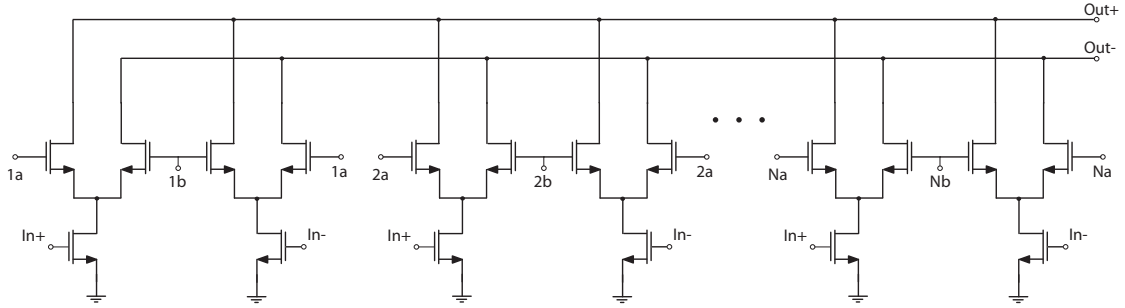


Figure 3.4: The pseudo differential multiplying DAC architecture

are repeated, only one DAC is needed and the bias coefficient can be replicated. Although variety of classic analog multipliers exist, [70–76] they require excessive voltage headroom and use too numerous transistors to be implemented in large quantity in modern CMOS processes. For modern discrete time signal processing applications, where many multipliers are required, more simple, low voltage compact layout circuits are required.

One of the simplest linear transconductors is the degenerated differential pair, shown in Figure 3.5. This circuit extends the linearity of the classic differential pair through the use of differential degeneration transistors, M3, M4 operating in the linear resistive region. In this circuit, common-mode current is supplied to M1, M2 by the current sources; however, differential mode current flows through M3, M4. The transconductance of this circuit is varied by changing the voltage bias on the gates of M3, M4 which changes their drain-to-source resistance [68].

A similar linear transconductor is the “input coupled linear degenerated differential pair”, shown in Figure 3.6. This circuit connects the gates of the differential degeneration transistors M3, M4, to the inputs, slightly extending the linear region of operation [77]. The current mirrors are varied to change the transconductance of this circuit over the same range as the linear degenerated differential pair of Figure 3.5.

Another linear transconductor is shown in Figure 3.7. This circuit uses cross-coupled current steering to increase the linearity of the transconductance region. M1–M4 operate in the linear region as the input transistors, varying the degeneration of the differential current steering NFETs M5–M8. The variable transconductance of the circuit is controlled by the differential bias voltage applied between M3, M4 and

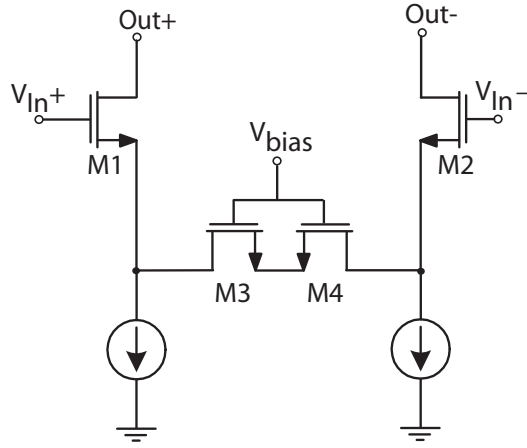


Figure 3.5: The linear degenerated differential pair

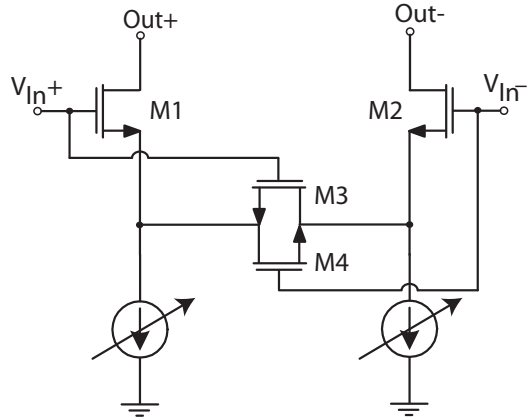


Figure 3.6: The input coupled linear degenerated differential pair

M5,M6 [77].

All of the multiplier topologies reviewed have in common a current mode output signal that can be fed into a transresistive adder circuit.

3.1.2 Adders for use in Discrete Time Signal Processing

The purpose of adder is to sum multiple current signals and convert the sum to an output voltage. The simplest way of doing this is through the use of passive lumped element resistors, as done in [78]. However, this method lacks the flexibility of setting

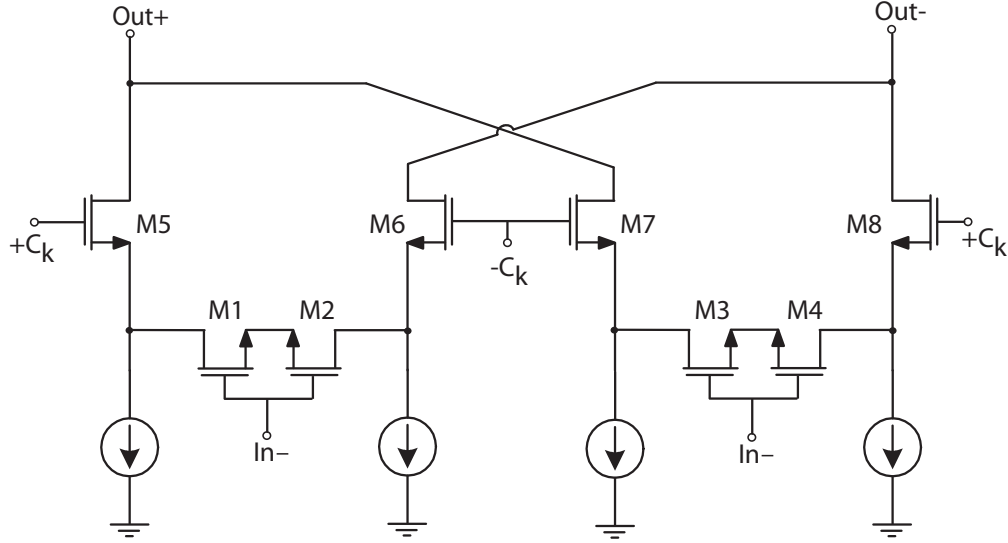


Figure 3.7: The cross-coupled current steering transconductor

the common-mode bias level independent of the resistance, which limits the topology to a single bias level and single operating speed. An alternate method that allows more flexibility in adjusting the differential mode resistance is shown in Figure 3.8. In this topology, the differential resistance is set by M1 and M2, whereas the common mode output voltage is adjusted by V_{ref} . The additional use of cascode NFETs allows the impedance to be linear over a wider range of current inputs. This approach is used in [52] for a discrete time FIR filter implementation.

3.1.3 Discrete Time Memory

In discrete time signal processing, memory is a key function that delays a sample so that it can be processed with another later sample in time. In switched capacitor circuits, memory has traditionally been implemented by storing a sample as a charge on capacitor, and then isolating the capacitor by opening switches on both sides of the capacitor until the signal is to be passed on [79]. The drawback to switched capacitor circuits is that they require two phases of non-overlapping clocks to allow closing of one set of switches before opening another set. At speeds above 100 MHz, non-overlapping clocks are difficult to realize, and sample-and-hold amplifier (SHA) circuits are used instead. SHAs can operate at higher speeds because they only

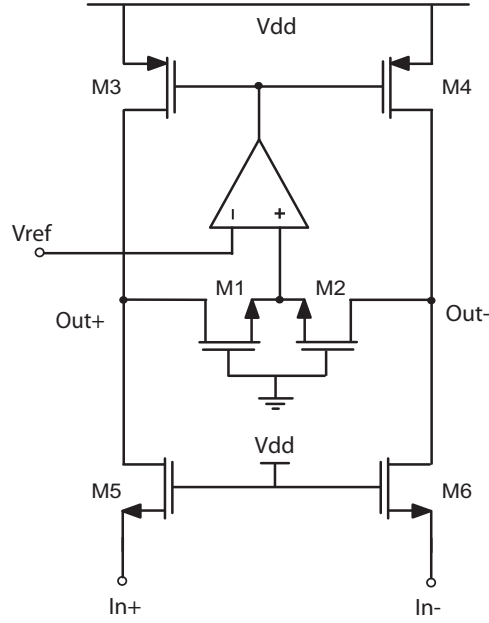


Figure 3.8: A cascode transresistive current adder

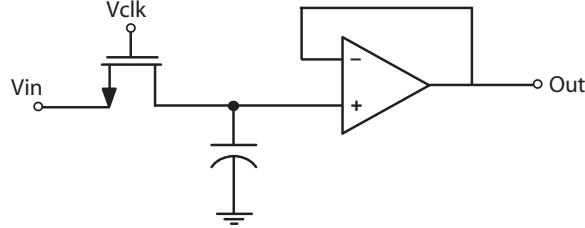


Figure 3.9: Open loop Sample and Hold

require a two phase clock which transitions the SHA between tracking mode and charge storage mode.

Although there are numerous circuits that implement the sample and hold function, for high speed circuit designs, open-loop topologies are typically preferable. Thus, no analog feedback loops are used, reducing the number of parasitic elements and reducing the concern of parasitic poles compromising the phase margin of the feedback loop. Figure 3.9 shows the typical circuit for an open loop SHA, consisting of a single switch and a unity gain buffer amplifier, similar to the topology used in [52]. For this circuit only a single clock is required and the output tracks the input when the switch

is closed. Thus, this circuit is sometimes also referred to as a track-and-hold circuit.

At speeds above 1 GHz, discrete time analog FIR filters in the literature use continuous time delays circuits rather than SHAs. Both [61] and [62] use an analog circuit delay. Although the advantage of this choice is higher speed operation, the disadvantage, is the limitation in available operating speeds.

There are similarities between the discrete time FIR filters discussed in this section and the proposed discrete time FFT, with two main exceptions. First, the FIR consists of many parallel signal paths that combine together in a single adder. On the other hand, the FFT is multiple input-multiple output, and requires interactions between the many parallel signal paths that interact through the multiple signal additions. Because of the large number of adders, the basic circuit used to implement the addition needs to be more simple and power efficient than those used in analog FIR filters where only a single adder is required. The second difference is that coefficient multipliers in the FIR do not typically repeat coefficient values more than twice [80]; however, in the FFT, a smaller number of distinct coefficients is required, but they are repeated many times. This means that instead of using a single multiplication DAC for each coefficient multiplier instance, it would be more efficient to use multiple variable gain transconductance amplifiers (G_m cells) controlled by a single DAC per coefficient value.

Having reviewed some of the typical circuit topologies used in discrete time signal processing applications, behavioral models for key blocks of the DT FFT processor are developed in the next section.

3.2 Behavioral Models

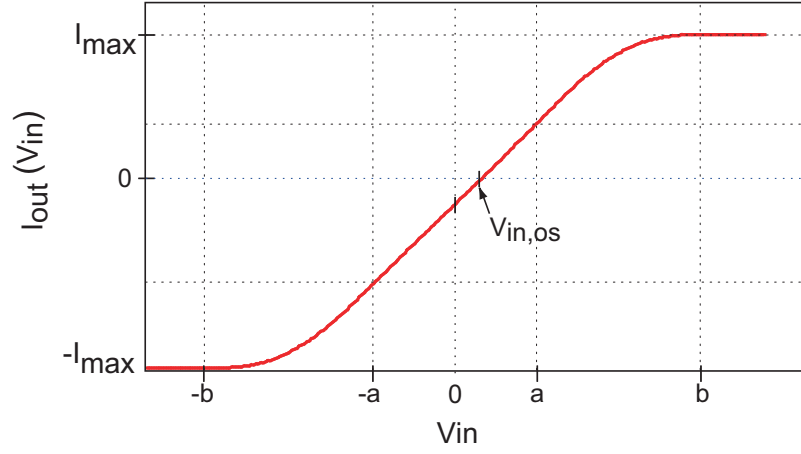
The behavioral modeling presented in this chapter was developed to achieve a compromise between a “back of the envelope” level system analysis and a detailed transistor level simulation. Although detailed transistor level models are more accurate, for the DT FFT processor with thousands of transistors, some simulations such as Monte Carlo are prohibitively time consuming. The other benefit of using behavioral models rather than transistor level models is that general system performance can be evaluated without the time consuming aspects of transistor level design such as

biasing and transistor sizing.

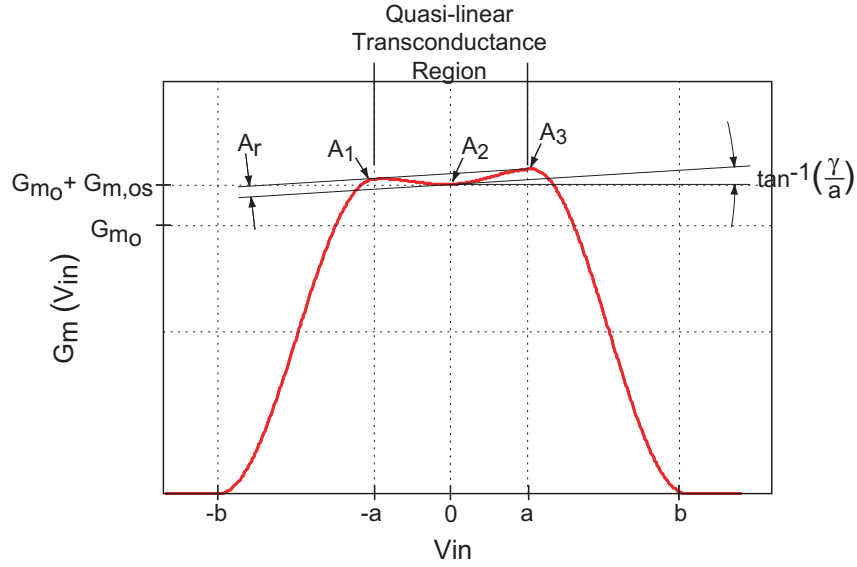
The software tool used to aid in the behavioral model simulations is Verilog-AMS operating within the Agilent ADS [81] and Cadence Virtuoso [82] simulation environments. Verilog-AMS is a SPICE oriented language that is similar in construct to the dominant digital circuit design language, Verilog-VHDL, but with functionality aimed toward modeling analog and mixed signal circuits. Although MATLAB and MATLAB/Simulink [83] are currently the most popular behavioral modeling tools, they are limited in that the user must be aware of design non-idealities and manually control the time simulation engine to capture them. The drawbacks of this approach include long simulations times, the potential to miss unmodeled glitches and other fast events not predicted ahead of time, and not taking advantage of progress made in SPICE transient solvers and harmonic balance engines. However, by describing models in Verilog-AMS which utilizes the SPICE engine to analyze model behavior, it is easy to perform DC, harmonic balance, and transient simulations in a tool that is efficient and tailored to solve the complex circuits, capturing subtle interactions and fast events. One of the primary advantages of the SPICE transient time domain solver is that it does not use a constant time-step, but instead moves forward, and occasionally backwards, in time, placing time steps at the exact moment when events occur. This is particularly useful in simulating jitter or timing glitches caused by transistor mismatch. For post simulation analysis such as FFTs that require evenly sampled data, the simulator's post-processor allows the data to be resampled. Verilog-AMS is currently integrated into all of the major IC development tools from Cadence, Mentor Graphics, Synopsys, and Agilent.

The primary behavioral models required are the SHA, multiplier G_m cell, and adder.

The behavioral model of the coefficient multiplier is based on the established transfer function for several different linear transconductors [16, 67]. A behavioral model needs to be computationally efficient and should be based on parameters that are intuitive to both the system designer and the circuit designer. The coefficient multiplier behavioral model contains parameters derived from two transfer functions: I_{out} versus V_{in} (Figure 3.10a), and G_m versus V_{in} (Figure 3.10b). From the I_{out} versus V_{in} function, the model inputs are I_{max} , the DC bias current, and $V_{in,os}$, the input offset voltage. From the G_m versus V_{in} function, the model inputs are: G_{m0} , the small signal transconductance value; a , which defines the extent of the quasi-linear



(a) I_{out} vs V_{in}



(b) G_m vs V_{in}

Figure 3.10: The curves used in the behavioral model of the G_m cell coefficient multiplier.
(a) The voltage-in current-out curve defined by equation (3.1) (b) The voltage-in transconductance-out curve formed by the derivative of equation (3.1)

portion of the transconductance curve; A_r , the ripple magnitude between A_1 and A_2 ; $G_{m,os}$, the deviation in the magnitude of transconductance at zero input; and γ , which defines the slope of the quasi-linear portion of the transconductance curve. To simplify intermediate calculations in the model, the parameters b , A_1 , A_2 and A_3 are also shown. For the system designer, the inputs I_{max} , G_{m0} are critical to making initial calculations about performance. In the design phase, when G_m circuits are

$$I_{diff} = \begin{cases} -\frac{A_1 b}{2} - \frac{A_2 a}{2}, & V_{in} \geq -b \\ \frac{A_1 (a-b)}{2\pi} \sin\left(\pi \frac{V_{in}+a}{a-b}\right) + \frac{A_1 V_{in} - a A_2}{2}, & -b < V_{in} \leq -a \\ G_{m0} \left[-1^N \frac{a A_r}{2N\pi} \sin\left(\frac{\pi N V_{in}}{a}\right) + \left(1 + G_{m,os} + \frac{A_r}{2}\right) V_{in} + \frac{\gamma}{2a} V_{in}^2 \right], & -a < V_{in} \leq a \\ \frac{A_3 (b-a)}{2\pi} \sin\left(\pi \frac{V_{in}-a}{b-a}\right) + \frac{A_3 V_{in} + a A_2}{2}, & a < V_{in} \leq b \\ \frac{A_3 b}{2} + \frac{A_2 a}{2}, & V_{in} < b \end{cases} \quad (3.1)$$

evaluated, the model inputs a and A_r are needed to used to compare the linearity of prospective architectures. During the circuit design phase, the parameters of $V_{in,os}$, $G_{m,os}$, and γ can be used in to evaluate trade-offs in transistor sizing and matching.

The equation used to implement the behavioral model is given by (3.1). The piecewise implementation represents the five regions seen in Figure 3.10(a): the flat outer regions, $|V_{in}| > |b|$; the transitional regions between $|a|$ and $|b|$; and the quasi-linear region, $|V_{in}| < |a|$. The quasi-linear region is important because this region is useful for implementing mathematical multiplication. Ideally, the quasi-linear region is represented by a linear slope equal to the multiplication value; however, when the model nonideality inputs A_r and γ are included, the linear slope becomes quasi-linear. The three magnitudes, A_1 , A_2 , and A_3 given in (3.2) define the magnitude of the G_m versus V_{in} curve at the points $-a$, 0 , and $+a$ respectively.

$$\begin{aligned} A_1 &= G_{m0} (1 + A_r + G_{m,os} - \gamma) \\ A_2 &= G_{m0} (1 + G_{m,os}) \\ A_3 &= G_{m0} (1 + A_r + G_{m,os} + \gamma) \end{aligned} \quad (3.2)$$

Equation (3.3) defines the extent of the transitional region which ranges from $|a|$ to $|b|$.

$$b = \frac{2I_{max} - A_2a}{A_3} \quad (3.3)$$

The three central regions of the model are constructed from cosine functions in the G_m versus V_{in} curve. The center cosine function between $\pm a$ has any number of ripples N , a magnitude of $A_r/2$ and an offset of $G_{m,os}$. The outer cosine functions between $\pm a$ and $\pm b$, are used to describe the roll-off of G_m . When the piecewise transconductance curve containing the three cosine functions is mathematically integrated, the five piecewise components of (3.1) are created.

The Verilog-AMS code of the G_m multiplier is shown in Appendix A, Figure A.1. This is a fully differential implementation of equations (3.1)-(3.3). The parameters of $G_{m,os}$, $V_{in,os}$ and γ are controlled by the SPICE Monte-Carlo engine and passed to each G_m multiplier instantiation within the FFT signal flow graph. Similarly the thermal noise, approximately $4kT(2/3G_m)$ for long channel devices, is also controlled by the SPICE engine for each G_m cell instantiation and applied to the voltage input. The other parameters of the behavioral model are passed as globals from the netlist. The implementation of the model is continuous in its first and second derivatives when $\gamma = 0$, and continuous in its first derivative when there is slope to the quasi-linear region. This allows the SPICE engine to transition across the operating regions without difficulty, making the model fast and accurate for large signal simulations.

Besides the behavioral model of the G_m cell-based multipliers, several other models are needed to implement the FFT processor. The behavioral model of the SHA includes voltage limits and an offset to model the voltage shift typically found in source follower amplifiers topologies [84]. Figure A.2 in Appendix A shows the Verilog-AMS code of the SHA.

The behavioral model for the adders are simply implemented as ideal resistors with the parameter of resistance passed to the Verilog-AMS code. An ideal model was used here based on the assumption that the primary sources of error in the system are from the multipliers. Appendix A, Figure A.3 contains a listing of the Verilog-AMS code used.

The serial-to-parallel function is constructed from 16 instantiations of the SHA Verilog-AMS function, and additional code to create the sequential clocks described in the

previous chapter. Appendix A Figure A.4 shows the listing of this code. The parallel-to-serial function uses the same sequential clock generating code, but directly maps the parallel output to a single serial output in code without instantiating the SHA behavioral models. This simplification was chosen to speed up simulations. The listing for the parallel-to-serial function is shown in Appendix A Figure A.6.

Using the models of the G_m cell based multipliers, adder, serial-to-parallel and parallel-to-serial function, the core functions of the FFT processor are described in Verilog-AMS. The higher level functions, constructed from these blocks are described in SPICE netlists. This eases the inclusion of transient noise sources and Monte Carlo statistical parameters that must be controlled by the SPICE engine and cannot easily be passed through to the Verilog-AMS level.

Appendix A, Figure A.7 shows the netlist of one of the butterfly structures. Eight transient voltage noise sources are included to add the input referred noise generated by multiplier circuit. The instantiation of the G_m cells includes the Gaussian statistical control for the variation of the parameters within the G_m cell.

Appendix A, Figure A.8 contains the netlist of the complete discrete time FFT processor described by behavioral models. Simulations are run with the SPICE transient time simulation engine of Agilent ADS. Figure 3.11 shows the setup used in the simulations. MATLAB is used to construct an OFDM signal that appears as it would at the physical input of the DT FFT processor; this signal is applied as a stimulus to the transient simulation and is read into a transient voltage generator where it is applied to the input of the Verilog-AMS representation of the DT FFT processor. The SPICE transient simulator simulates over the length of time of the input data, and the output is read back into MATLAB.

In the post-processing section, the MATLAB input is passed through a parallel ideal FFT and the SPICE output of the behavioral model is compared to the output of the ideal FFT. The vector difference between the two signals is subtracted to form an error signal. The error signal represents both the noise and distortion contributed by the DT FFT processor.

One of the first performance decisions required in a top-down design of an Discrete-Time FFT processor is the power budget. Knowing the power allocated to the FFT signal flow graph and the order of the FFT, equation (2.7) can be used to determine

the number of G_m cells, which in-turn is used in Equation (3.4) to determine the current I_{max} available to each G_m cell.

$$I_{max} = \frac{\text{FFTPower}}{V_{dd}N_{G_m}} \quad (3.4)$$

When ultimate operating speed, relatively low resolution, and low power consumption are required, open-loop G_m cells should be used. In open-loop G_m cells, the bias current can be fully applied to the output signal swing, making the cells more power efficient. Because closed-loop feedback is not applied, only one pole exists per G_m cell, allowing the circuit to be fast. The maximum voltage swing in and out of G_m cells V_{max} is limited by the supply voltage and the threshold voltage levels of the n and p-type devices, which ultimately limits V_{max} to about $V_{dd}/2$ in deep sub-micron CMOS processes [85]. V_{max} should be maximized to increase signal-to-noise ratio but must also be compared against other design goals. Additionally, it is important that the signal magnitude at the input and output of each butterfly circuit remain approximately equal to maintain signal levels at the optimal SNR. Since each butterfly circuit adds two current signals together, the voltage gain of any single path should be 1/2. Then, the transresistance of the adder can be set to $\frac{1}{2} G_{m_0}$. Using the values of I_{max} , V_{max} and A_v the small signal transconductance level G_{m_o} can be calculated using equation (3.5).

$$G_{m_o} = \frac{2A_v I_{max}}{V_{max}} \quad (3.5)$$

Assuming fixed bias current, I_{max} , (3.5) indicates that minimizing G_{m_0} will maximize V_{max} and thus maximize the processor dynamic range. Thus G_m circuits with a large linear voltage swing should be selected rather than circuits with large G_{m_o} . On the other hand, if I_{max} is not limited, then increasing either I_{max} or V_{max} will increase dynamic range.

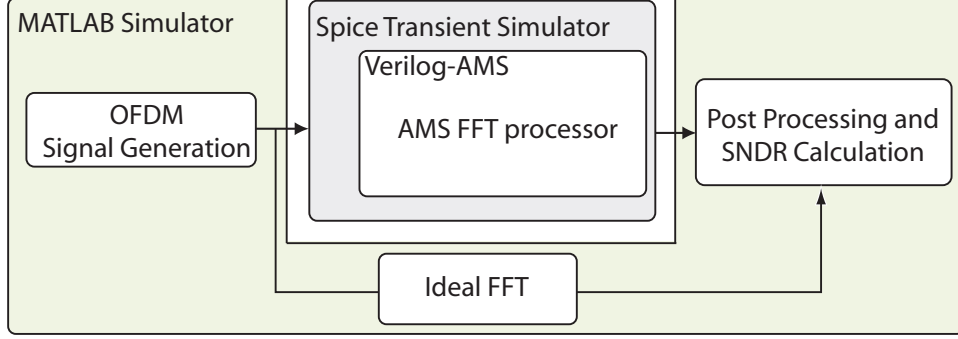


Figure 3.11: The setup used to simulate the discrete-time FFT processor.

3.3 System Simulation Results

System simulations can be pursued to better evaluate the non-idealities included in the G_m model and to verify the feasibility of the proposed approach. In order to understand the architectural trade-offs in the design, the input magnitude of the OFDM input signal is swept and applied to the Verilog-AMS description of the system as shown in Figure 3.11. The SNDR is calculated by creating a error signal from the output signal against the results of a perfect FFT performed on the input signal. The SNDR is then the rms magnitude of the input signal relative to the rms magnitude of the error signal:

$$SNDR = 20\log_{10} \frac{\sqrt{\frac{1}{N} \sum_{k=1}^N V_{ideal}(k)^2}}{\sqrt{\frac{1}{N} \sum_{k=1}^N (V_{out}(k) - V_{ideal}(k))^2}} \quad (3.6)$$

This allows the SNDR of the FFT processor to be evaluated at both weak and strong signal levels providing a clearer picture of how sub-channels attenuated by multi-path perform when passed through the FFT processor. From the SNDR versus input magnitude curves, two metrics are of interest: the peak SNDR and the dynamic range. The peak SNDR occurs at a large input magnitude where the distortion and noise contributed by the DT FFT processor are minimal. However, because the input signal is located in the receiver before the equalization, there are both strong sub-channels and weak sub-channels contained within the same signal. Therefore, having a wide range of input magnitudes with sufficient SNDR to pass all sub-channels is

more important than having a large peak SNDR at just one input magnitude. Thus, the dynamic range of the FFT processor is the significant metric. Dynamic range is defined here as the range of input values for which the SNDR exceeds the minimum SNR detection threshold for OFDM, 7 dB. Above this value, OFDM signals can be detected with a bit error rate of less than 1×10^{-5} [1].

3.3.1 Optimizing the G_{m_0} value

System simulations can be performed to verify the value of transconductance G_{m_0} that provides the optimal dynamic range. In this simulation, it is assumed that the bias current was fixed at I_{max} and had a value of $40\mu A$, the ratio of a -to- V_{max} was initially assumed to be $\frac{1}{3}$ (to be further investigated in the next section), and $G_{m,os}$ and γ were set to zero. Figure 3.12 shows the three associated transconductance curves used in the simulation.

Using the three resulting transconductance curves, the SNDR of the DT FFT processor was simulated while sweeping input signal magnitudes. Figure 3.13 shows the results. The smallest value of transconductance, $50\mu A/V$, had the widest voltage swing and tolerated the largest signals; however for weak input signals, it contributed the most noise. The largest value of transconductance, $200\mu A/V$, contributed less noise but also had less headroom to tolerate large signals. When comparing the dynamic range values, the small values of transconductance with the largest voltage swings are best. Thus, for the DT FFT processor, designing the transconductors to operate over a maximum voltage swing is a good design goal. The value of $G_m = 100\mu A/V$ is selected as the smallest feasible choice for the DT FFT processor.

3.3.2 Voltage Gain through the Multiplier and Adder

The voltage gain of each butterfly structure was determined by the transconductive gain of the coefficient multiplier, the transresistive gain of the adder, and the number of current branches that feed into the adder. In different parts of the FFT lattice there are either two or three sets of transconductors combining current signals into each adder. Although the transresistance of the adder can be set to the inverse of G_{m_0} to achieve a gain of unity, this is not necessarily the ideal case. This simulation varied

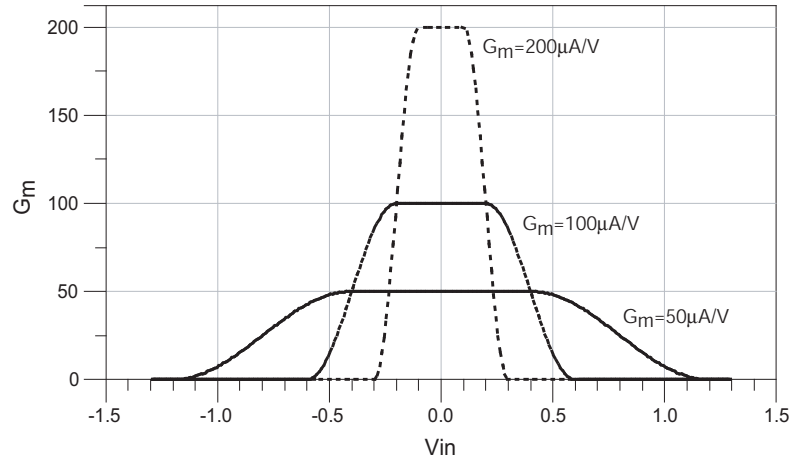


Figure 3.12: Varying the transconductance of the multipliers affects the useable input voltage range when operating current is held constant.

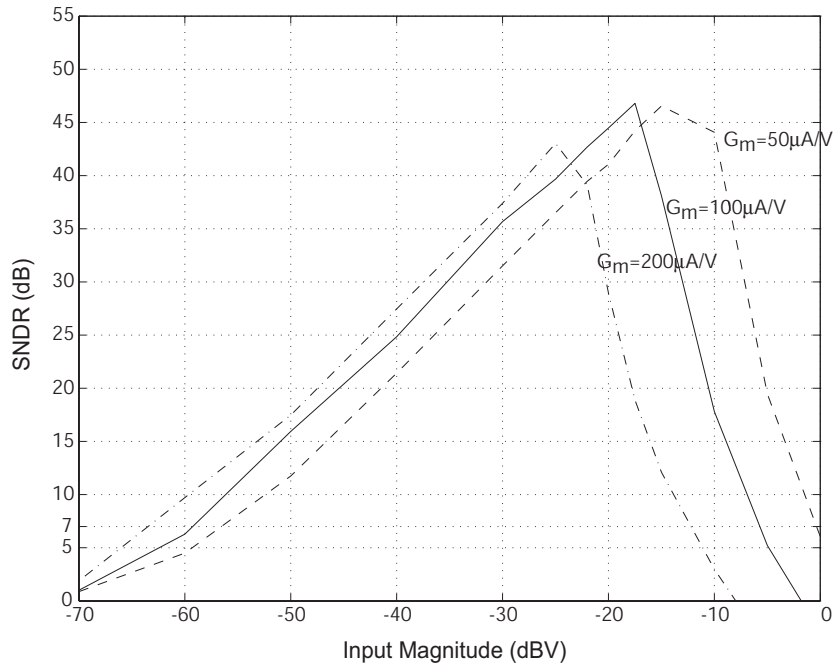


Figure 3.13: Simulating the DT FFT processor with different G_m values shows that lower values allow a larger dynamic range.

the resistance of the adder circuit to determine which effective gain gave the best results. Again, it is assumed that the bias current is fixed at $40\mu\text{A}$, $G_{m0} = 100\mu\text{A}/V$, the ratio of a-to- V_{max} is $1/3$, and $G_{m,os}$ and γ are zero. Figure 3.14 shows these results. A gain of $1/2$ achieved the widest dynamic range and the greatest peak SNDR.

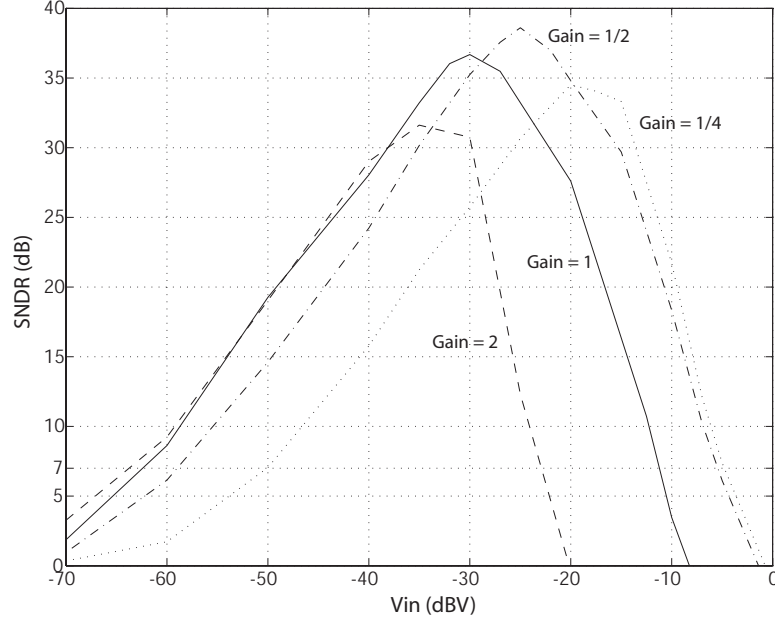


Figure 3.14: The combined gain of the multiplier and adder combination affects the dynamic range of the system.

3.3.3 a-to- V_{max} ratio

Assuming the designer has determined bias current I_{max} and the voltage swing V_{max} for the G_m cells, the range of voltage inputs with quasi-linear transconductance can be determined. For the behavioral model given, the range of the quasi-linear region is set by the a-to- V_{max} ratio. Figure 3.15 shows the SNDR values for the processor versus OFDM input signal magnitude. The inset of figure 3.15 shows the corresponding transconductance curves for the a-to- V_{max} ratio values of 100%, 50%, 25% and 0%. The value of 100% represents the ideal transconductor for which the voltage input range is quasi-linear, whereas the value of 0% represents the transconductance curve of the typical differential pair for which there is no quasi-linear region.

The three cases should be expected to behave the same for small input signals where

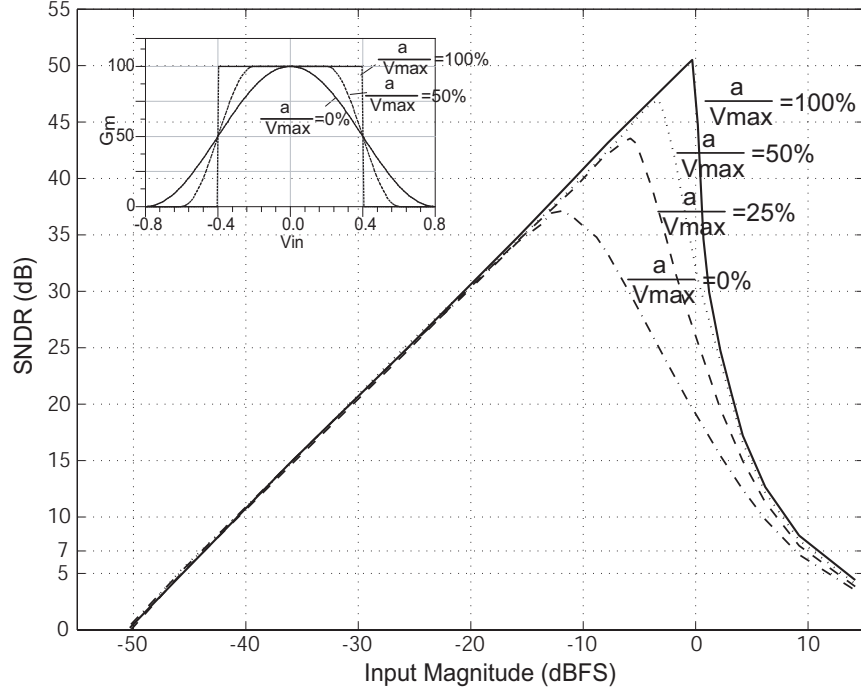


Figure 3.15: Varying the a -to- V_{max} ratio of the G_m cell behavioral model determines the quasi-linear range of the transconductance curve useful for multiplication (inset). The SNDR curves show that the a -to- V_{max} ratio does not have a strong effect on dynamic range for values above 50%.

large signal effects are negligible. The difference occurs with large input signal magnitude. As expected, the ideal case of 100% has the highest peak SNDR of 51 dB; however, the a -to- V_{max} value of 50% has only a slight reduction to a 47dB peak SNDR, whereas the a -to- V_{max} value of 0% has a peak of 37 dB. The dynamic range for the 100% and 50% results differs by less than 0.5 dB and the dynamic range of the 100% and 0% results differ by only 2 dB. This indicates that a wide quasi-linear region is not required of the G_m cell. This also means that complex feedback linearization circuits are not needed, but instead less complex circuit topologies can be used. The value of a -to- V_{max} of 50% was selected for the rest of the simulations as it is both straight forward value to design for and it retains similar dynamic range to the ideal case.

3.3.4 A_r ratio

The level of ripple A_r in the quasi-linear region is also an important nonideality of the G_m cell. Circuit design efforts to maintain a constant transconductance over a wide range of input voltages will not be perfect. The metric of A_r in the behavioral model (4)-(6) accounts for the fact that it is nearly impossible to make this region perfectly flat. For small signal amplifiers with a known third Input Intercept Point (IIP3), A_r can be calculated from:

$$A_r = \left(\frac{4a}{\pi} \frac{1}{10^{IIP3/20}} \right)^2 \quad (3.7)$$

System simulations of the FFT processor were used to determine the acceptable level of A_r . The SNDR curves in Figure 3.16 show the simulation results of the FFT processor for SNDR versus OFDM input signal magnitude. The inset in Figure 3.16 shows the G_m curve for the corresponding values A_r of 0%, 10% and 20%. Each of the five results were similar for small input signals. In the range of SNDR from -16 dBFS to 6 dBFS, the SNDR tended to decrease with larger amounts of amplitude ripple. For very large signals between 6 dBFS and 15 dBFS, where most clipping occurs, the SNDR results were essentially unchanged. For all practical purposes, any value of amplitude ripple less than 20% had negligible impact on the dynamic range of the FFT processor. As with the previous simulations, the amplitude ripple simulation showed that the FFT processor does not require stringent design specifications on the transconductors used as multipliers.

3.3.5 A_r variation

The matching requirements between the coefficient multipliers and adders within the system were analyzed with Monte Carlo simulations the parameters, A_r , $V_{in,os}$ and $G_{m,os}$. In these simulations, circuit noise was turned off in order to clearly understand the minute effects of mismatch and offset voltage.

Besides determining the amplitude ratio, A_r , the standard deviation of variations in A_r between G_m cells was also simulated. The results showed that values of σ_{A_r} less than A_r had little effect on dynamic range. Thus targeting values of σ_{A_r} less than or

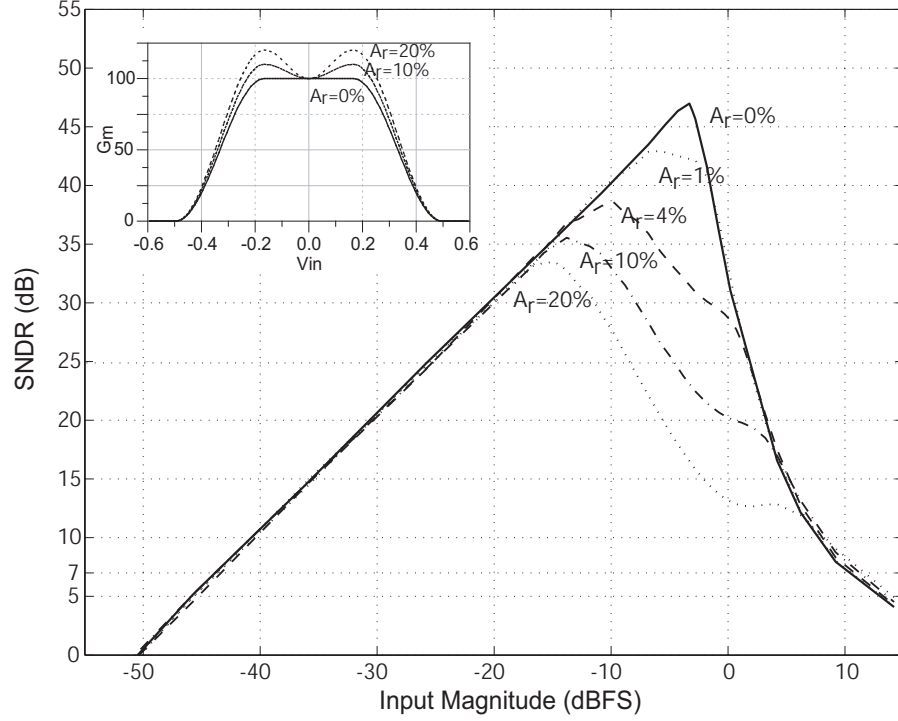


Figure 3.16: Amplitude ripple, A_r models the non-ideality found in the quasi-linear region of the G_m cell's transconductance curve (inset). The SNDR curves show that high levels of amplitude ripple lower peak SNDR but do not degrade the dynamic range.

equal to A_r is desirable.

3.3.6 G_m offset

The offset in transconductance, $G_{m,os}$, was varied in each G_m cell by the Monte Carlo engine. Figure 3.17(a) shows the SNDR versus input signal magnitude curves. From these simulation results, it can be seen that a matching of $G_{m,os}$ better than 10% maintains adequate SNDR, but it has no effect on dynamic range.

3.3.7 V_{in} offset

The matching of the standard deviation of input voltage offset, $V_{in,os}$, for the G_m cells was also varied by the Monte Carlo engine. Figure 3.17(b) shows the SNDR versus

Table 3.1: Summary of Model Parameters used in Jitter and Blocker Simulations

Model Parameter	Value
I_{max}	$40\mu\text{A}$
G_{m0}	$100\mu\text{A/V}$
V_{max}	$400\text{mV}_{pk,diff}$
a-to- V_{max}	50%
A_r	10%
σ_{A_r}	10%
$\sigma_{G_{m,os}}$	2%
$\sigma_{V_{in,os}}$	0.5mV

input signal magnitude curves. SNDR performance for weak input signal magnitudes behaves in a similar manner to thermal noise, with SNDR linearly increasing as input signal magnitude increases. The results show that the $V_{in,os}$ needed to be less than 0.5mV to equal the effect of thermal noise created in each transconductance stage. In deep sub-micron CMOS process, where precise transistor matching is difficult, affecting $V_{in,os}$, care must be taken to ensure that $\sigma_{V_{in,os}}$ is less than thermal noise and does not limit the FFT processor's dynamic range.

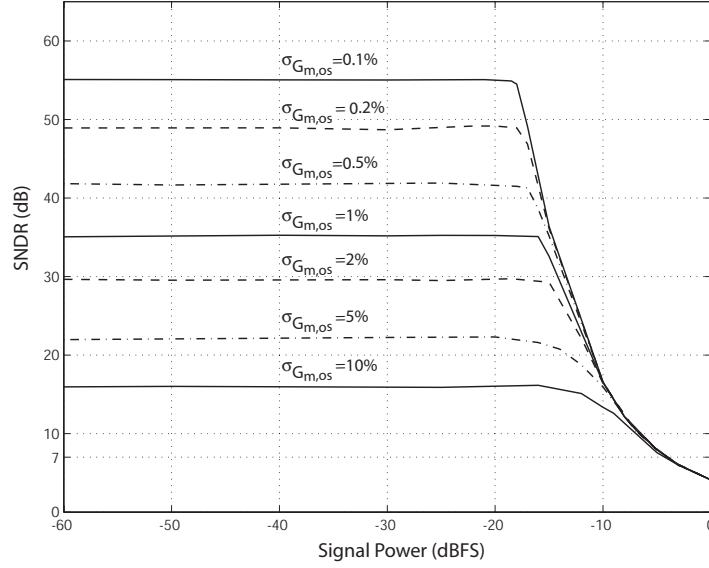
3.3.8 Jitter

It is also important to verify that the FFT processor can operate with realistic levels of clock jitter. For these and subsequent simulations, the G_m cell model values in Table I were used. Figure 3.18 shows the SNDR simulation results with jitter applied to the system clock. The input signal magnitude was $V_{in} = -6$ dBFS. For comparison, the theoretical SNDR curve for a single ideal sample and hold amplifier clocked with jitter is given by equation (3.8) and is also shown as a dashed line in Figure 3.18.

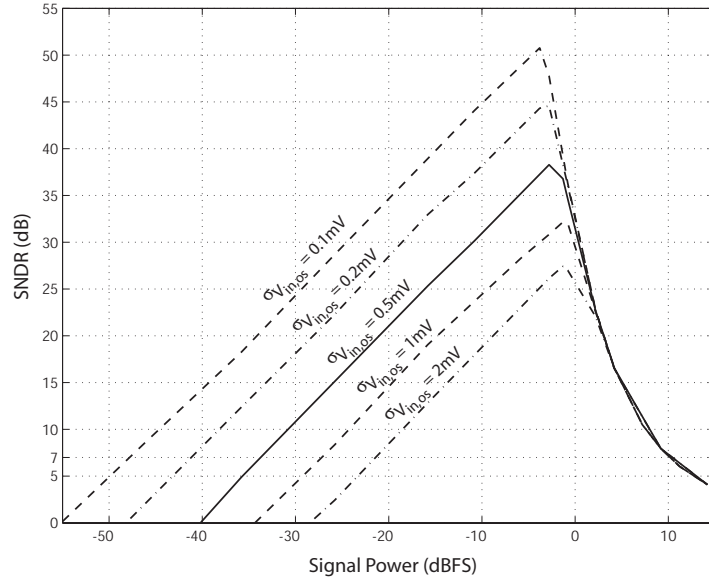
$$SNDR = 10 \log_{10} \frac{1}{(2\pi f_{sinusoid} \sigma_t)^2} \quad (3.8)$$

where σ_t is the standard deviation in jitter of the clock signal.

The FFT processor simulation results track the theoretical curve for levels above



(a) A Monte-Carlo simulation of G_m offset



(b) A Monte-Carlo simulation of offset voltage

Figure 3.17: Monte-Carlo simulation of the discrete-time FFT processor with several values of standard deviation in (a) G_m offset and (b) voltage offset applied to the G_m cell behavioral model

15 pS. Below 8 pS, the FFT processor shows no degradation in performance. This demonstrates that the DT FFT processor will operate well for typical jitter levels in the clocks of UWB ADCs.

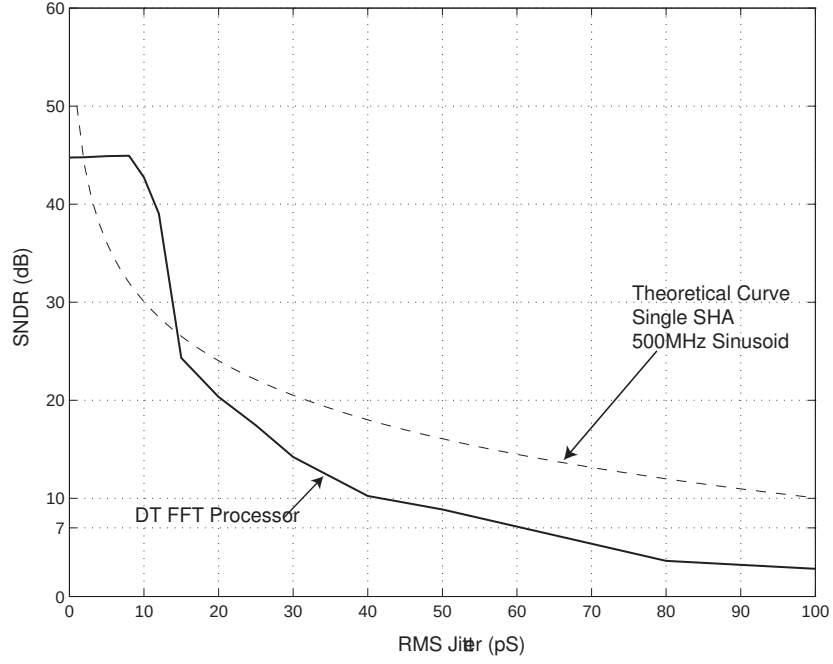


Figure 3.18: Simulation results of the discrete-time FFT processor with clock jitter applied to the clock divider input.

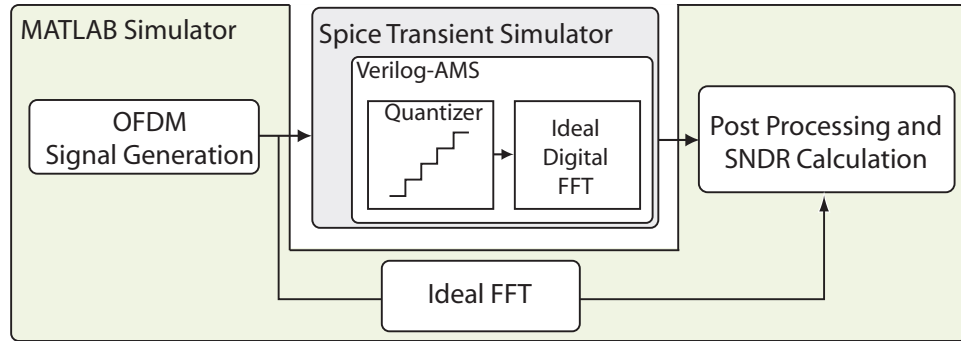


Figure 3.19: The simulation setup used to simulate the all digital comparison FFT processor.

3.3.9 Comparison with All Digital Processing

Using the system simulations, comparisons between the performance of the DT FFT processor and the traditional digital OFDM architecture can be made. A traditional digital system was constructed from an ADC represented by an ideal quantizer and an infinite precision, ideal FFT processor. The simulation setup used to test the digital

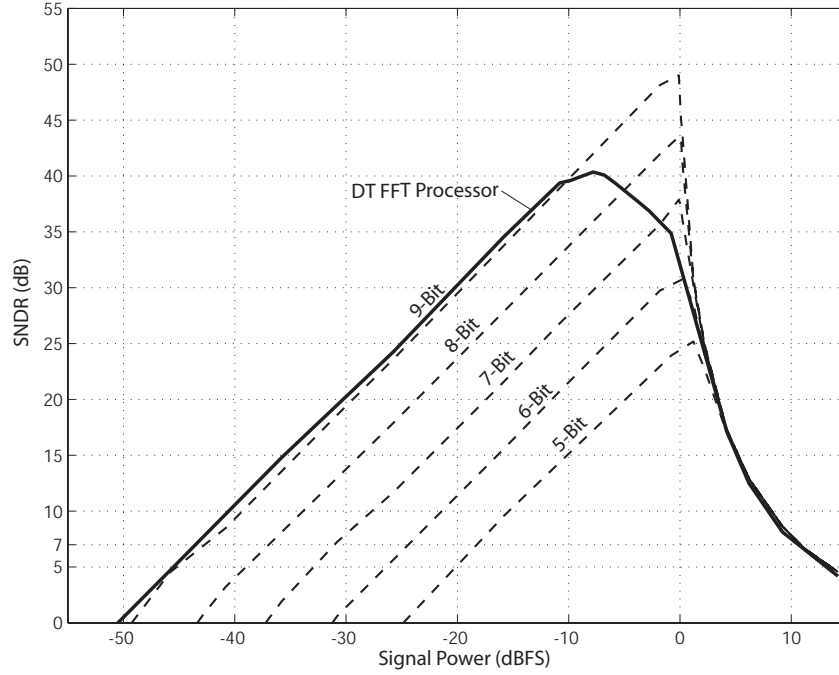


Figure 3.20: Simulation results of the discrete-time FFT processor (solid) compared to simulation results of the all-digital FFT processor with varying levels of input ADC quantization (dashed). The discrete-time FFT processor exceeds the dynamic range of the all-digital FFT processor with 9-bit resolution.

processor with OFDM demodulation is shown in Figure 3.19. The only non-ideality included in the digital system is the quantization noise of the ADC. Although other distortion contributors exist in the traditional digital system, they vary by architecture making the analysis overly complex. By using the best case scenario digital model, the performance advantages of the DT FFT processor are only heightened.

Figure 3.20 shows the results for 5-bit, 6-bit, 7-bit, 8-bit and 9-bit quantized ADCs as dashed traces. The SNDR grows linearly with signal input power up to the point of full-scale input. The performance of the DT FFT processor was plotted on the same figure with a solid trace. It shows that the dynamic range of the DT FFT processor exceeds that of an ideal 9-bit all digital ADC and FFT processor.

3.3.10 Blockers

Since the FFT processor must be able to tolerate narrow-band blockers that lie in the receive bandwidth of the OFDM receiver, blocking tones are injected with the OFDM input signal to explore the processor's blocker performance. When a large blocker is present, it is assumed that the system automatic gain control will amplify it to the back off point of the ADC, typically between -11 dBFS and -6 dBFS, and the remaining sub-channels will remain weak signals. In this case, the FFT processor needs to retain sufficient dynamic range to demodulate the weak signals while rejecting the AGC limited blocker. In other cases, additional blockers may exist at weaker levels but still should not compromise the performance of the FFT processor. To perform this simulation, the SNDR versus OFDM input signal magnitude test was performed with a range of injected blocker tone magnitudes. The model non-idealities were set to the values given in table 3.1. Figure 3.21 shows the results for the DT-FFT processor with a solid trace. For comparison purposes, the same blocker tone was also simulated in the all digital system having 6-bit quantization noise; the results of this simulation are shown with the dashed trace.

The results show that the FFT processor effectively tolerates blockers with little performance degradation. Blockers between -60 dBFS and -27 dBFS, are removed leaving the DT FFT processor 54 dB of available dynamic range with which to detect weak sub-channels. This is an 18 dB improvement over simulation results for the ideal 6-bit all digital system. For a large -6 dBFS blocker, the dynamic range is 6 dB greater than that of a 6-bit all digital system. Thus, the proposed DT FFT processor architecture tolerates narrow-band blockers and improves receiver selectivity over the traditional all-digital approach.

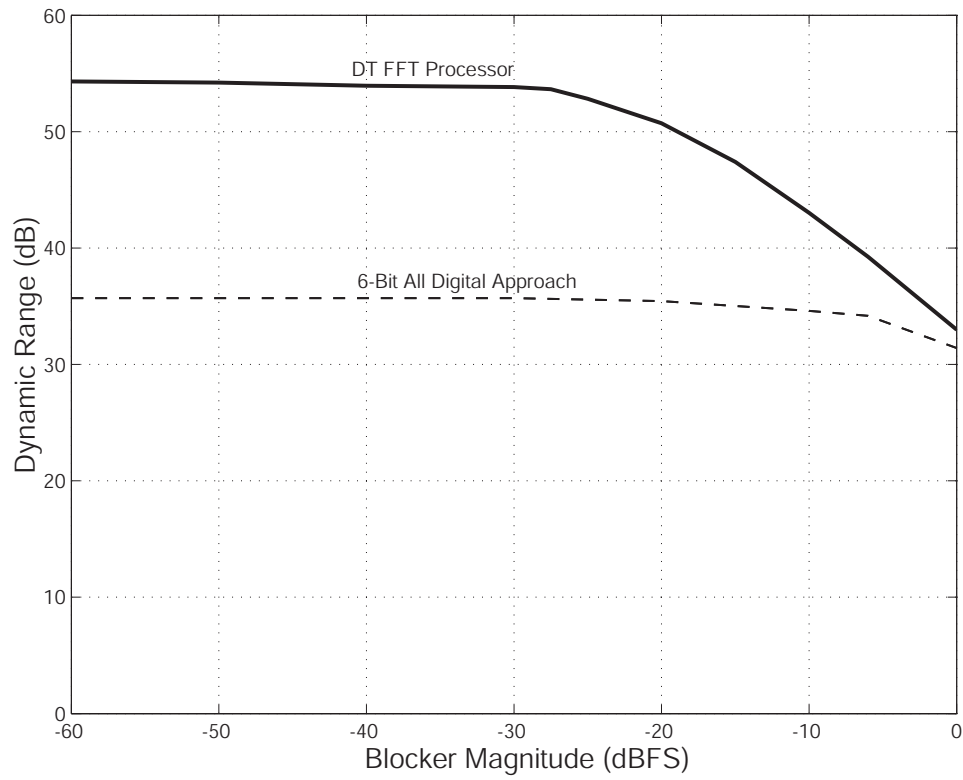


Figure 3.21: Simulation results of the discrete-time FFT processor dynamic range (solid) versus narrow band blocker magnitude demonstrates that the processor is able to perform demodulation in the presence of large narrow-band blockers. For comparison, the blocker performance of the 6-bit all digital system is shown (dashed).

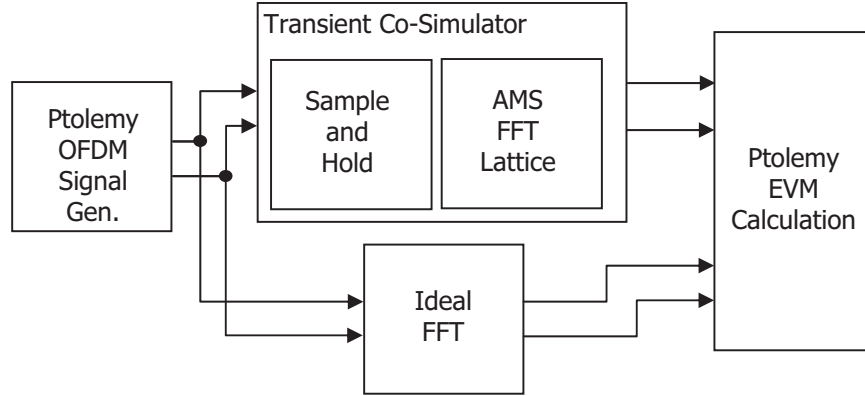


Figure 3.22: The system simulation setup used in Ptolemy based simulations.

3.3.11 Ptolemy System Simulations

Additional simulations were performed with the test setup shown in Figure 3.22, using Agilent Ptolemy in place of MATLAB as the high level simulator. These system level simulations also include a quantization limited ADC and ideal DSP based FFT processor. The Ptolemy representation of the ADC is that of an ideal quantizer of either 6-bit and 8-bit resolution.

The dynamic range of the DT FFT processor was simulated and the results shown in Figure 3.23. In this simulation, the OFDM input signal applied to the DT FFT was varied in magnitude, and the resulting EVM was measured and averaged on a sub-channel basis. When the input signal is weak the thermal noise of the converter degrades the EVM. When the input signal is strong, clipping occurs and the signal is distorted. For comparison, the performance of the 6-bit and 8-bit all digital ADC and FFT systems are shown. As can be seen, the proposed system performs better than the 8-bit digital system. All systems simulated were based on 1.2 Volt power supply. However, in the system based on the all digital FFT, input signals above 17 dBV were completely distorted and unrecoverable; whereas, in the discrete-time FFT, large signals were distorted in a more gradual manner. These results also confirm the Blocker simulations discussed in the previous section.

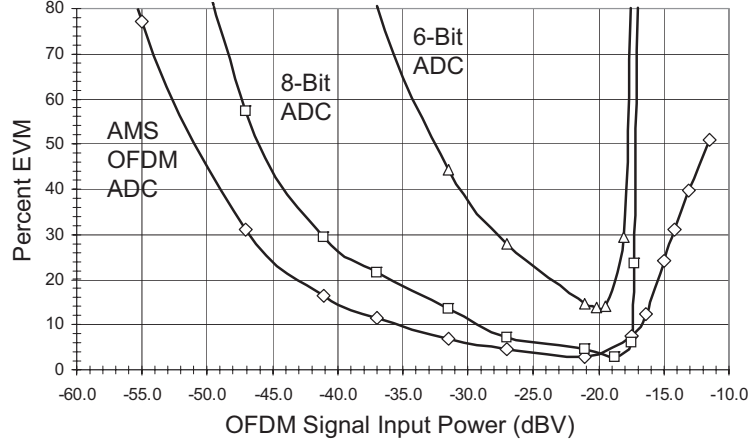


Figure 3.23: The EVM sweep across input signal magnitude shows that the DT FFT Processor performs better than an ideal digital system of 8-bits.

3.3.12 Power Consumption Savings

The results of the system simulations in these sections demonstrate that the DT FFT processor has improved linearity over the traditional all digital approach. The results also show that the DT FFT processor does not require a linear transconductor with a highly optimized linear response, but can be implemented with a less than ideal coefficient multiplier. Table 3.2 summarizes the results. The multiplier can be implemented with any level of a-to- V_{max} ratio, up to 10% ripple with σ_{A_r} of 10%, and $\sigma_{G_{m,os}}$ and $\sigma_{V_{in,os}}$ less than 1% and 0.5mV respectively.

There are also significant power consumption savings demonstrated by the proposed architecture. The relocation of the signal processing functions of the FFT from the digital signal processing domain to the discrete-time domain typically result in a 75% power reduction [57, 63]. The all-digital FFT processors in [42, 48] that consume 175 mWatts and 450 mWatts, respectively, at 1 GSps could be improved to better than 40 mWatts by shifting to discrete-time domain. Meanwhile, in going from a 6-bit flash ADC for the leading edge UWB data rates to a 2-bit flash ADC following the DT FFT processor, better than a factor of ten savings in ADC power consumption can be achieved [37]. Combined, these power savings are significant and suggest that the DT FFT processor will help enable future leading data rate OFDM receivers to be used in mobile hand-held applications.

Table 3.2: Summary of Design Goals based on System Simulations of the discrete-time FFT Processor

Model Parameter	Value
V_{max}	maximize
I_{max}	$\leq 40 \mu\text{A}$
f_c	$\geq 700 \text{ MHz}$
a-to- V_{max}	$\geq 0\%$
A_r	$\leq 10 \%$
σ_{A_r}	$\leq 10 \%$
$\sigma_{G_{m,os}}$	$\leq 10\%$
$\sigma_{V_{in,os}}$	$\leq 0.5\text{mV}$
σ_{A_r}	$\leq 10\%$
jitter	$\leq 10 \text{ pSec}$

3.4 Summary

In this chapter, multiplier circuits were reviewed and behavioral models were introduced to describe the proposed discrete time FFT processor in simulation. System simulations were performed to explore the circuit design requirements of the G_m multiplier and adders used within the FFT signal flow graph.

The results of the system simulations in these sections demonstrate that the DT FFT processor has improved linearity over the traditional all digital approach. The results also show that the DT FFT processor does not require a linear transconductor with a highly optimized linear response, but can be implemented with a less than ideal coefficient multiplier. The multiplier can be implemented with any level of a-to- V_{max} ratio, up to 10% ripple with σ_{A_r} of 10%, and $\sigma_{G_{m,os}}$ and $\sigma_{V_{in,os}}$ less than 1% and 0.5mV respectively. A discussion of the benefits of the proposed architecture also were shown to allow a factor of ten reduction in system power consumption.

In the next chapter, detailed transistor-level circuit design of the multiplier, adders, and sample-and-hold amplifiers is presented based on the system-level studies in this chapter.

Chapter 4

Circuit Design and Layout

In the previous chapter, system level simulations of the DT FFT processor were performed using developed behavioral circuit models. The simulations provided insight into the requirements of the analog signal processing functions and the circuits required to implement them. In this chapter circuit design and analysis of those functions is presented and the circuits are optimized to best meet the requirements of the proposed DT FFT processor.

4.1 Multiply and Add Function

From the butterfly structure derived in the previous chapter, a half butterfly structure is shown in Figure 4.1, consisting of two coefficient multipliers and an adder. In this section, the multiplier and adder circuit components will be described and the half butterfly structure will be used in simulations to demonstrate the feasibility of using these building blocks for implementing the FFT signal flow graph (SFG).

4.1.1 Multiplier

One of the most important functions in the Discrete Time FFT processor is the coefficient multiplier, $Y = c_k \cdot X$. The non-idealities of this function have the most significant impact on the distortion contributed to signals within the DT FFT pro-

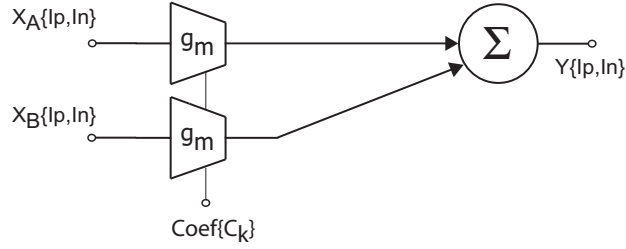


Figure 4.1: A portion of the butterfly structure used in the transistor level design of the coefficient multiply and add.

cessor. The coefficient multiplier bounds the limits of processor linearity and is also a primary contributor to the total system power consumption. As described in Section 2.2.1, the coefficient multiplier is implemented as a variable linear transconductor that transforms an input voltage signal into a current signal. When the current signal outputs of several variable transconductors are linearly added in the transresistive adder, the sum of the current signals are converted to a voltage signal.

One of the simplest transconductor circuit topologies that can implement $Y = c_k \cdot X$ is the source coupled (SC) differential pair, shown in Figure 4.2. In the source coupled differential pair, current provided by the current mirror, is steered by transistors M1 and M2 to the outputs in direct proportion to the differential voltage V_{in} between the gates of M1 and M2. Coefficient multiplication occurs when the tail current I_{mirr} is made variable, and proportional to the coefficient value, c_k .

Although classically illustrated by a single current mirror at the source of M1 and M2, in modern CMOS process, where matching is a concern, two symmetrical current mirrors are employed and physically located adjacent to M1 and M2. The diagram in Figure 4.2 helps to illustrate the actual structure of the differential pair.

The transconductance of the differential transconductor is defined as the first derivative of the differential output current, I_{diff} with respect to the differential input voltage:

$$G_m = \frac{dI_{diff}}{dV_{in}} \quad (4.1)$$

For example, consider the plot of I_{out} versus V_{in} for the source coupled differential

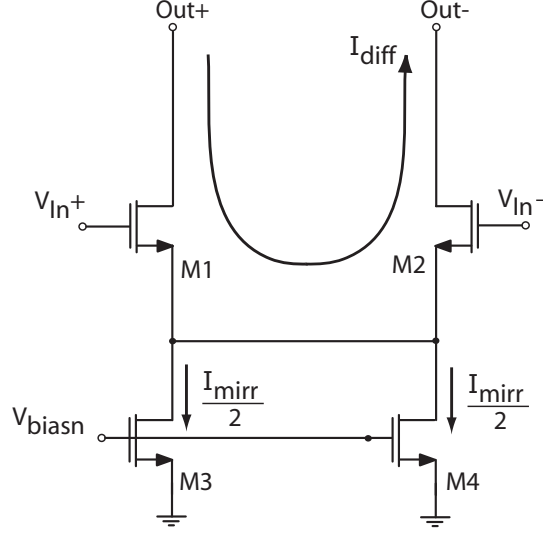


Figure 4.2: The common source differential pair is one of the simplest forms of the CMOS transconductor

pair (Figure 4.3(a)). The upper and lower bound on signal swing are defined by the bias current used in the circuit, I_{mirr} . The derivative of the differential current given by I_{out} versus V_{in} , is G_m versus V_{in} , as defined by Equation 4.1, and shown in Figure 4.3(b). Ideally, the transconductance would remain constant over a wider range of input voltages. When used as a multiplier, this allows a wide range of input voltages to be multiplied by the same transconductance value G_{m0} . As can be seen in the Figure, the source coupled differential pair does not perform as an ideal linear transconductor. The range of input voltage for which the transconductance is close to G_{m0} is limited to a narrow range of values. If the source coupled differential pair were used in the DT FFT processor, the dynamic range would be significantly limited compared to the ideal case.

In Section 3.2 a behavioral model was introduced and system simulations were performed to determine the circuit requirements of the DT FFT processor, including the coefficient multiplier. $V_{in,os}$ matching was found to be the most important parameter to optimize, followed by the a-to- V_{max} ratio, A_r and $G_{m,os}$. In addition to these requirements, wide-bandwidth, ($f_c \geq 700$ MHz), and a wide linear voltage swing V_{max} are desired.

There are many circuit topologies found in the literature that implement four quad-

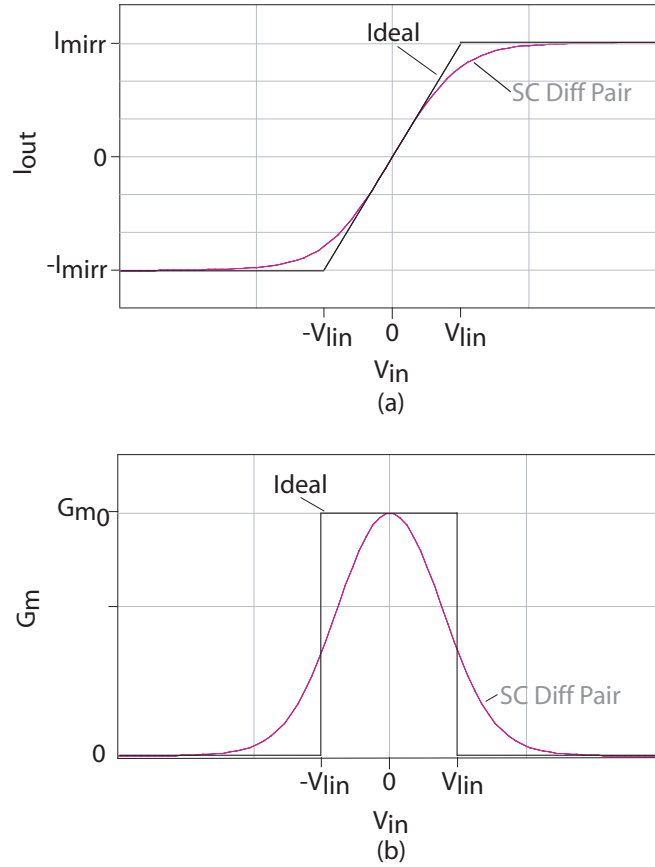


Figure 4.3: The ideal transconductor has a voltage-to-current transfer function (a) and a voltage-to-transconductance transfer function (b) with a wide flat region near the center, V_{in} . In contrast, the typical source coupled differential pair is also shown.

rant multiplication, $Y = X_1 \cdot X_2$, and coefficient multiplication, $Y = c_k \cdot X$ [70–76, 78]. A number of these topologies rely on circuit feedback loops to linearize the response; however, due to the speed requirements in this work, open-loop topologies with a single pole are better suited for bandwidths exceeding 500 MHz [77, 78]. To maximize the output voltage swing, the number of transistors stacked vertically should also be kept to a minimum. Given a 1.2Volt supply, a linear multiplication range of 400mV_{pk-pk} is feasible [54, 61, 78].

The circuit shown in Figure 4.4 is a linear transconductor that offers a wide constant transconductance region approaching the ideal case [77]. To understand the operation of this circuit, it is initially helpful to envision the input transistors M1 and M2 as

similar multipliers in cascade. The nominal transconductance G_{m0} is primarily set by the linear mode resistance of M1 and M2. The tuning range of 0 to 150 $\mu A/V$ is achieved by setting M1,M2 to a width of 4 μm with 4 fingers. Ideally M3-M6 would provide a very large transconductance, requiring them to be large, so that only the degeneration resistance from M1 and M2 would define the circuit's effective transconductance; however, they also ideally should have zero input capacitance so that the circuit would be fast. As a compromise, they are sized at a width of 2 μm and 4 fingers. The mirror transistors M7-M10 are designed for a maximum output resistance at their drains and to minimize mismatch between the mirror currents. Transistors with gate lengths of 0.24 μm , twice the minimum length for the technology are selected to improve output resistance. The width of 2.5 μm with 2 fingers is selected to minimize mismatch.

Figure 4.5 shows the G_m versus V_{in} of the circuit in Figure 4.4 for several values of c_k . The transconductance is reasonably linear over a differential input range of 250 mV. However, when the input voltage becomes too large, M1 and M2 no longer operate in the linear resistive region, but move into saturation. For the simulation shown in Figure 4.5 this effect occurs at V_{in} of ± 200 mV. When either M1 or M2 saturate, no additional current can be steered by the associated differential pair, and the differential output current remains constant as the input voltage further increases. This results in a nearly linear roll-off of the transconductance with increasing V_{in} .

The coefficient c_k smoothly controls the transconductance over a wide range of values. Thus, the circuit demonstrates suitable characteristics for a programmable coefficient multiplier.

4.1.2 Analog Adder

The adder acts as a transresistor, summing the input currents and converting them into an output voltage at the desired common-mode level. The simplest way to do this is with passive lumped element resistors [78]. However, this approach does not have the flexibility of setting the common-mode bias level independent of the resistance, so this topology limits the bias current and operating speed to one value. A method that separates the common-mode and differential mode resistance allows more flexibility in adjusting the differential resistance level independent of the bias current [52].

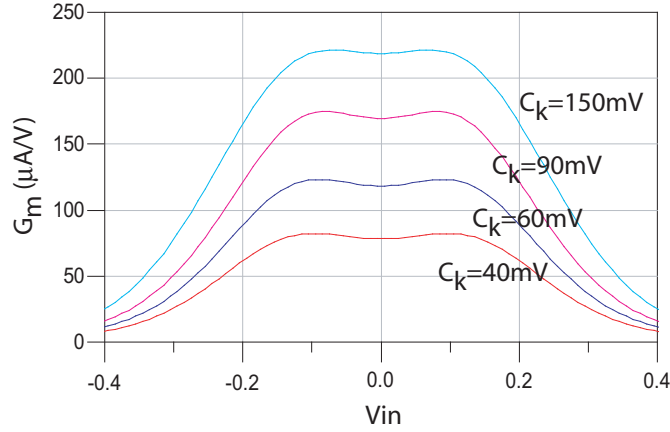


Figure 4.5: Simulated transconductance of the variable G_m cell is adjusted through bias C_k .

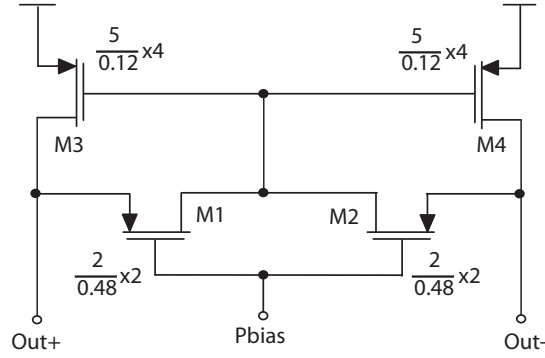


Figure 4.6: The adder circuit used in the construction of the FFT butterfly structure provides independent common-mode resistance and differential mode resistance.

The adder circuit shown in Figure 4.6 has the features of small size, adjustable differential resistance, and common-mode feedback to stabilize the common mode bias in the G_m multipliers. The inputs to the circuit are differential currents from the output of the multiplier circuit of Figure 4.4. Several current sources can be combined when connected to these common nodes. One adder circuit can sum currents from several coefficient multipliers. The output of the circuit is the differential voltage arising between nodes $Out+$ and $Out-$. Thus, the input and output ports are physically the same, but the signaling domain changes from current to voltage.

For proper circuit operation, transistors M1 and M2 operate in the linear resistive

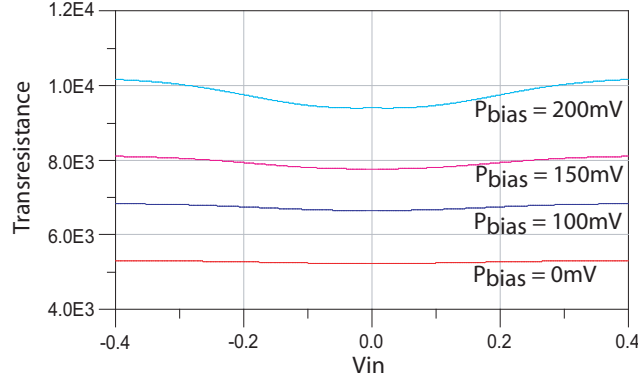


Figure 4.7: Simulated Adder circuit transresistance tuning as a function of P_{bias}

region while M3 and M4 operate in saturation as current sources. The linear resistors M1 and M2 can be tuned by adjusting P_{bias} . The center tap between the two resistors provides a point to sense the common mode voltage between the outputs. This value is fed back into the current source transistors M3 and M4. Common-mode feedback enables the circuit to tolerate variations in the bias current from the G_m cells without affecting the common-mode output voltage, and allows the circuit to maintain an optimal wide voltage swing.

The adder is simulated using the test circuit previously shown in Figure 4.1. Figure 4.7 shows the simulation results of the transresistance versus V_{in} for several values of P_{bias} . The transresistance value can be varied from $5\text{ k}\Omega$ to $10\text{ k}\Omega$. This allows a voltage gain of $\frac{1}{2}$ as required by the system simulations in the previous chapter. Ideally this value should remain independent of input voltage magnitude; however, for large current swings, the circuit no longer functions properly. When the differential input current is large, the transistors M1 and M2 saturate, and the differential resistance of the adder circuit also increases resulting in non-linearity. This can be seen in the figure for the case of $P_{bias} = 200\text{mV}$. At higher values of P_{bias} beyond 200mV the effect grows worse. Fortunately, the circuit is not required to provide transresistances above $10\text{ k}\Omega$, so the circuit makes an effective linear adder for this application.

When used in the prototype DT FFT processor, the ability to adjust the transresistance post-fabrication is a useful feature. However, it does require an additional off-chip pin. If the cost of an extra pin is not justified, P_{bias} can be simply connected

to ground.

Having described the multiplier and adder circuits, further simulations are performed with the two coefficient multipliers and an adder circuit as shown in Figure 4.1. The test circuit is further loaded with two more identical half butterfly structure circuits, the typical load existing in the DT FFT signal flow graph.

The voltage-in, voltage-out transfer function of the butterfly test circuit is shown in Figure 4.8(a). The maximum voltage swing of the circuit is $\pm 450\text{mV}$, but due to the $10\text{ k}\Omega$ upper limit of the adder, the full voltage swing is not realized. Instead the voltage swing for the unity voltage gain case is $\pm 300\text{mV}$. This slightly reduces the dynamic range of this circuit. Redesigning to adjust the size of the adder circuit differential resistance would allow the full range.

Figure 4.8(b) shows the plot of voltage gain versus input voltage. The voltage gain is roughly flat over a 300mV input range. The amplitude ripple in the quasi-linear region is larger than in the transconductance plot of Figure 4.5. This is due to the combination of the non-linearity in the adder and the non-linearity of the multipliers.

The frequency response of the test circuit is shown in Figure 4.9. The bandwidth of the circuit is seen to be 700 MHz , which is lower than the target value of 1000 MHz . A trade-off was made in the coefficient multiplier between increasing the size of the input transistors to increase the voltage swing or decreasing their size to increase the circuit bandwidth. Alternatively, the bias current could be increased to increase the bandwidth, at the expense of power consumption.

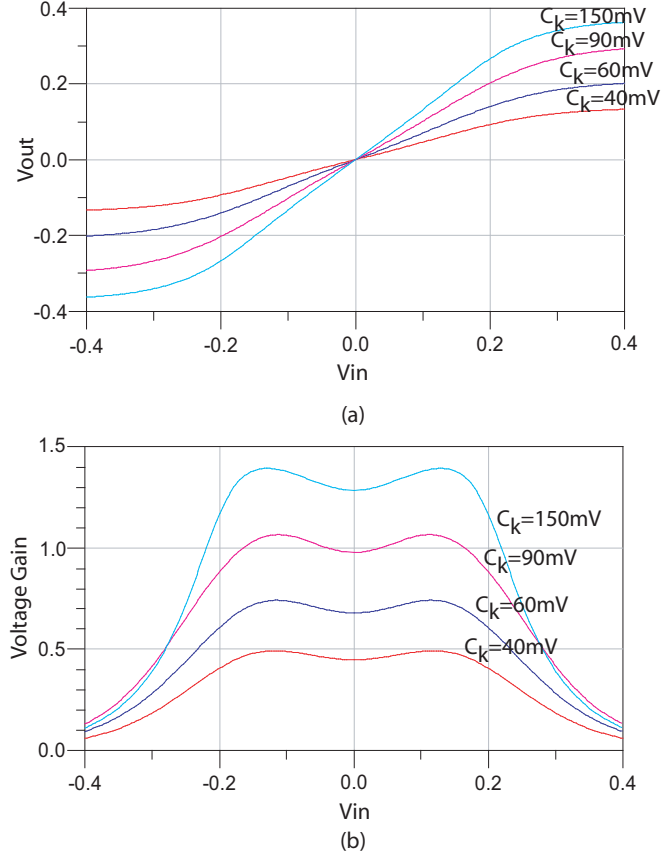


Figure 4.8: (a) Simulated voltage-in, voltage-out transfer function of the half butterfly structure. (b) shows the derivative of (a), which is the voltage gain of the half butterfly structure.

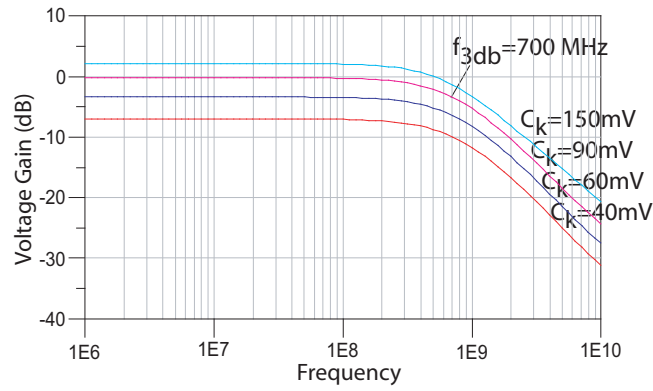


Figure 4.9: Simulated frequency response of the half butterfly structure with typical loading.

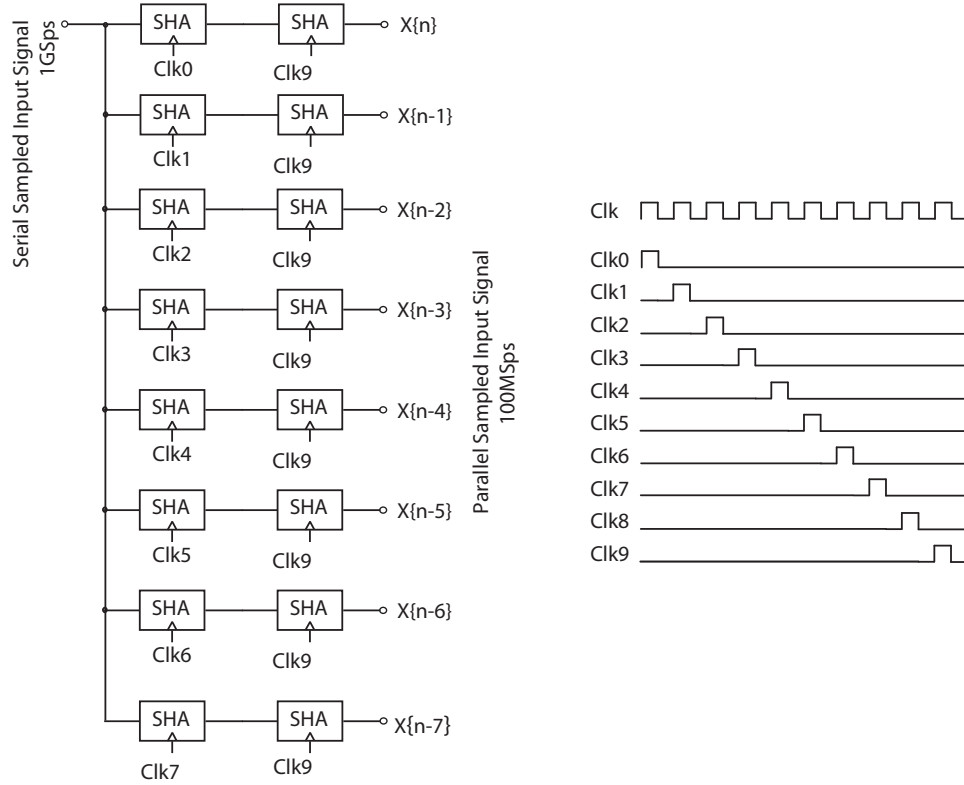


Figure 4.10: The serial-to-parallel conversion function implemented by two banks of sample-and-hold amplifiers.

4.2 Sample-and-Holds

The serial-to-parallel conversion function required for the DT FFT processor can be constructed from two parallel banks of sample-and-hold amplifiers. Figure 4.10 shows the block diagram to be implemented with the circuits presented in this section. The first bank is clocked by one of eight clock phases from the clock generation circuit. Although circuit paths are shown as single ended in Figure 4.10 for clarity, the SHAs require differential clock signals and use the notation $\text{ClkP}\{k\}$ and $\text{ClkN}\{k\}$ in the circuit diagrams to indicate the complementary phases. The Clk9 phase is used to clock the second bank of SHAs. The clock generation circuitry will be described in Section 4.3.

High speed sample-and-hold amplifiers typically consist of three primary elements: sampling switch; hold capacitor; and unity-gain buffer amplifier [86–88]. Figure 4.11

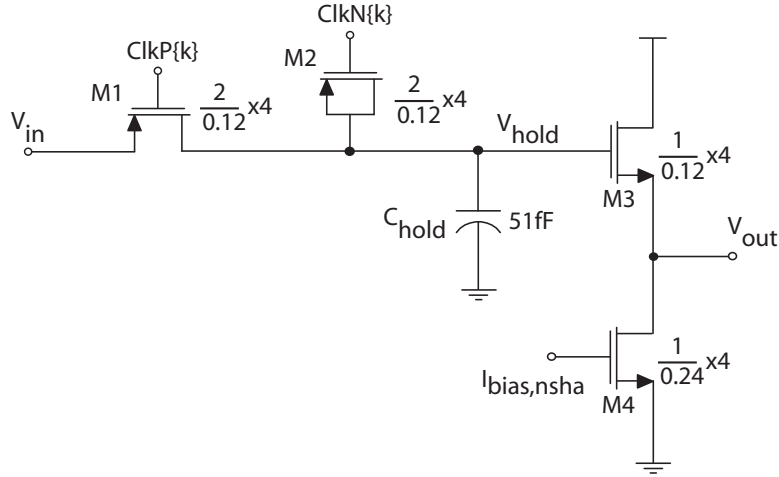


Figure 4.11: The PFET based sample-and-hold with source follower amplifier.

shows one half of a pseudo-differential sample-and-hold amplifier based on a PFET switch. The SHA operates in two modes controlled by the sampling clock. In the tracking mode, the switch M1 is on and the hold capacitor charges to the input voltage level and then tracks it. In the hold mode, the switch M1 is off and the voltage on the capacitor, C_{hold} , remains fixed. In both states, the source follower buffer amplifier, consisting of M3,M4, tracks the voltage on the hold capacitor. Because the NFET M3 is a MOS device, no direct current path exists from V_{hold} to V_{out} , and the amplifier operates with large current gain. This allows the buffer amplifier to supply current to load devices without draining charge off of the hold capacitor C_{hold} .

The circuit design requirements of the buffer amplifier are determined by the required input and output voltage swing, the required bandwidth and the load requirements. In this case, the buffer amplifier must drive a load capacitance of 52 fF with good linearity over the system voltage swing of 500 mV_{pk-pk} and a rise-time better than 0.5 nsec. The value of 52 pF was chosen for the hold capacitor because it is the minimum size to hold the voltage without drooping due to charge leakage through the gate of the buffer amplifier. One drawback to 120 nm CMOS technology is the gate leakage currents due to tunneling. For M3 at the shown size, a tunneling current of 3 nA is expected from simulations. Using the slew rate Equation 2.11, the required slew current is $52\text{ }\mu\text{A}$. Adding a design margin of 20% requires a minimum bias current of $60\text{ }\mu\text{A}$ in the buffer amplifier. Scaling M3 to have a width of $1\text{ }\mu\text{m}$ and 4 fingers

Table 4.1: Summary of Simulation Results for the PFET Switch SHA design

Bias Current	49 μ A
Bandwidth	1.85 GHz
Output Drive Impedance*	3850 Ohms
Positive Clock Load	6.3 fF
Negative Clock Load	6.3 fF
Input Impedance*	90 fF
Total Power Consumption	117 μ W
Required Clock Power	18 μ W

*single ended

meets the bandwidth and voltage swing requirements. The buffer amplifier output impedance is 3.85 $k\Omega$, single ended.

With the value of the hold capacitor selected, the switch can be sized so that its on resistance does not negatively impact circuit performance. The drain to source resistance of the switch R_{sw} , the output resistance of the previous stage, R_{out} and the hold capacitor form a first order low-pass filter, with cutoff frequency given by:

$$f_{c,input} = \frac{1}{2\pi C_{hold}(R_{out} + R_{sw})} \quad (4.2)$$

For optimal input voltage tracking, the corner frequency of this RC filter should exceed the fifth harmonic of 500 MHz (half the 1 GHz sample rate or 2.5 GHz). With a 52 fF hold capacitor this requires that the switch resistance be less than 1.22 $k\Omega$. Figure 4.12 shows the simulation results of a series transistor with a gate length of 120 nm, 4 fingers versus width W . The results emphasize the fact that the series resistance decreases with increased gate width while the gate capacitance increases with increased width. In the design shown in Figure 4.11, a value of 2 μm was selected. This gives a total bandwidth for the SHA of 1.85 GHz as seen in Figure 4.13.

When the PFET switch M1 turns off (moves into a high impedance state), the positive charge trapped under the gate must flow out into the SHA junctions. Assuming that the charge flow splits evenly between the source and drain, the charge flowing into

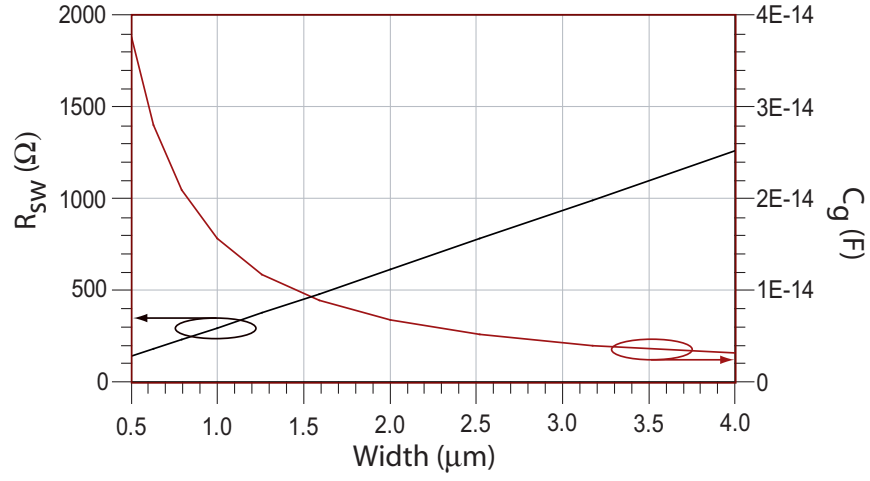


Figure 4.12: Simulated drain-source resistance versus device width of a PFET switch with $L_g = 120nm$ and 4 fingers. The left axis shows gate-to-bulk capacitance.

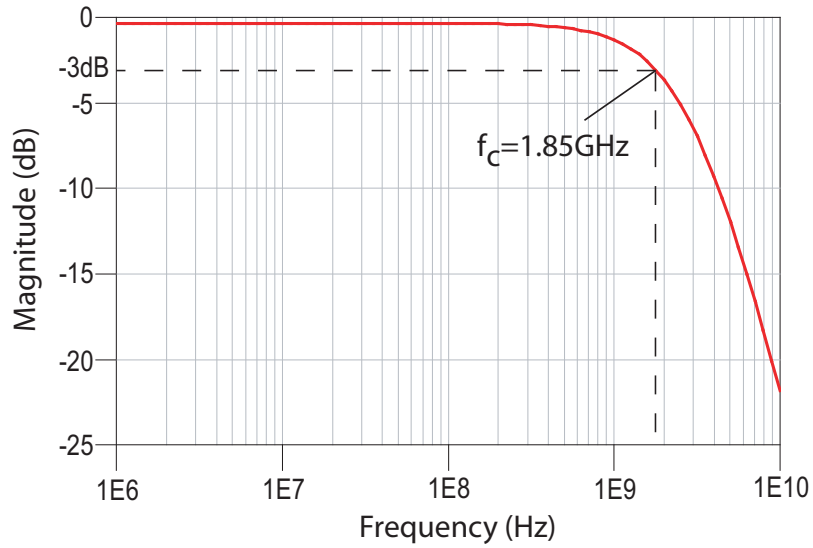


Figure 4.13: Simulated open switch frequency response of the sample-and-hold amplifier.

the low impedance SHA input dissipates. However, the trapped charge on the hold capacitor side of the switch sees a high impedance and cannot dissipate. As the switch closes, this trapped charge is stored on the hold capacitor creating a voltage error known as a *charge pedestal* [79]. For a PFET switch this voltage offset can be

calculated using Equation 4.3.

$$\Delta V_{hold} = \frac{C_{ox}WL(V_{tp} - V_{in})}{2C_{hold}} \quad (4.3)$$

where C_{ox} is the oxide capacitance, V_{tp} is the p-channel threshold voltage, and V_{in} is the input voltage. ΔV_{hold} is linearly related to the input voltage V_{in} which causes a gain error, but it is also related to V_{tp} which varies non-linearly with V_{in} . This non-linearity can result in distortion in the SHA and must be avoided.

The simplest method of reducing the charge pedestal ΔV_{hold} is to minimize the switch size. However, as already seen, there is a limit to this minimization due to the switch resistance. Another means of reducing the charge pedestal is to add a shorted dummy transistor to the node with C_{hold} , that is clocked with an opposite phase to the switch [79]. Although this does not create a path for the trapped charge to dissipate it does create trapped charge of opposite polarity on the hold capacitor. In theory, if the two phases are exactly opposite, the charge should cancel out. The dummy transistor should be half the size of the switch transistor. In practice, the problem with canceling charge pulses is the assumption that exactly half the charge flows out the drain and the other half flows out the source. Because the charge trapped under the gate sees additional dissipation paths through the substrate, it is difficult to accurately simulate what percentage will be trapped. Alternatively, if the impedance of the circuit driving the switch is high, then all of the charge will remain on the hold node. In this case it is better to size the dummy transistor to be the same as the switch transistor.

Although the above methods may not completely eliminate the charge pedestal, the use of differential signaling helps to further reduce the effects of the error. In this work, a pseudo-differential approach is used in the form of two identical SHAs with the input signal applied differentially. For small signals, the inputs are nearly identical and the charge pedestal in ΔV_{hold} is small. However, at large signal swings, the differential cancellation is less effective. Figure 4.14 shows a time-domain simulation of an 80 MHz sinusoid of 800 mV_{pk-pk} being clocked through the SHA at 1 GHz. The output of the SHA should ideally overlay the input during the track phase, and be flat (constant) during the hold phase. The ΔV_{hold} can be seen most noticeably where the signal is largest. For example at 21.8nS, in the figure, the offset is 40 mV.

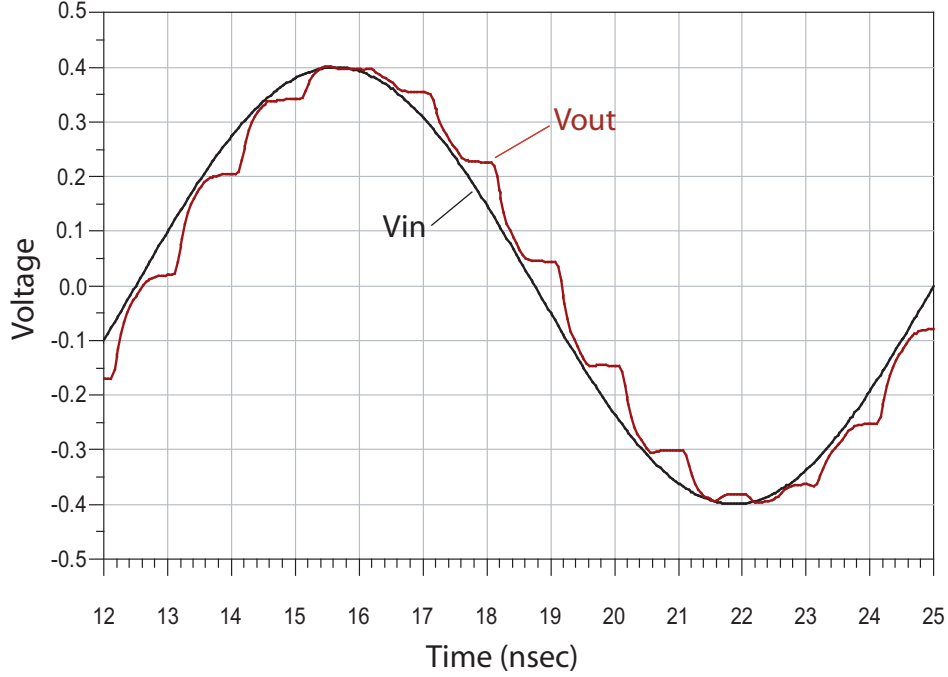


Figure 4.14: Simulation results of an $800mV_{pk-pk}$ 80MHz sine-wave passing through the track-and-hold with 1GHz clock.

The simulation results of the PFET switch based SHA are given in Table 4.1. The input and output load impedances are given for a single-ended half of the pseudo-differential SHA. However, the total power consumption of $117 \mu W$ is given for the full pseudo-differential SHA.

In addition to the PFET switch based sample-and-hold amplifiers, a second bank of SHAs are required to follow the first bank for the serial-to-parallel conversion function as shown in Figure 4.10. Because of the voltage level down-shifting of the NFET source follower used in the first stage, the second bank of SHAs is implemented with complementary PFET transistors to shift the output signal back up to the original level. Thus, a set of NFET-switched SHAs are also required.

For the NFET SHA design, shown in Figure 4.15, the constraints on the hold capacitor are somewhat relaxed due to the fact that tunneling currents looking into the PFET unity gain amplifier are simulated to be less than 0.1 nA. This allows the hold capacitor to be reduced in size, which in turn allows for a wider bandwidth SHA. The hold capacitor is set to 26 fF, the minimum reliable size recommended by the layout

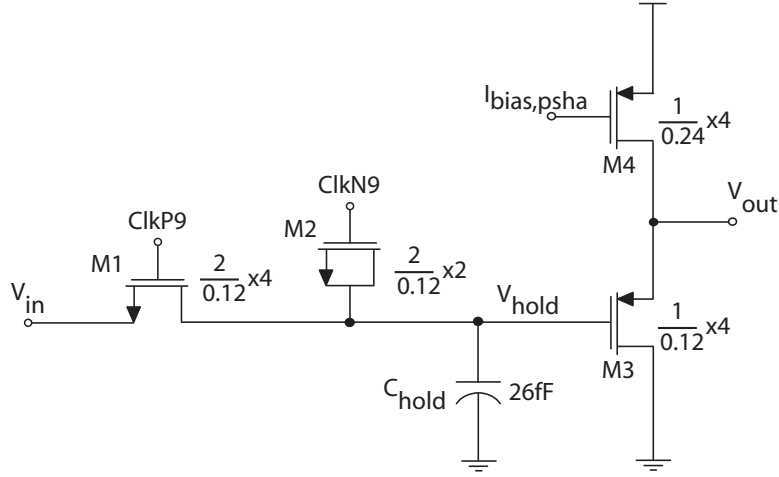


Figure 4.15: The NFET switch based sample-and-hold with source following amplifier.

Table 4.2: Summary of Simulation Results for the NFET Switch SHA design

Bias Current	24 μ A
Bandwidth	2.40 GHz
Output Drive Impedance*	3200 Ohms
Positive Clock Load	10.86 fF
Negative Clock Load	5.42 fF
Input Impedance*	45 fF
Total Power Consumption	57.6 μ W
Required Clock Power	20.1 μ W
*single ended	

guidelines. The buffer amplifier in the PFET unity gain amplifier sees a reduced load capacitance in the stages that follow it, so the bias current is accordingly reduced to 24 μ A.

Figure 4.16 shows the switch impedance versus width for a four finger, 120 nm NFET transistor. The left axis shows the channel capacitance. The width of 2 μ m was selected as a good compromise between switch resistance R_{sw} and channel capacitance. For the case of an NFET switch, the charge trapped in the channel when the transistor shuts off has a negative polarity. Thus, a negative charge pedestal will occur on the V_{hold} node. The equation used to evaluate the charge pedestal for an NFET

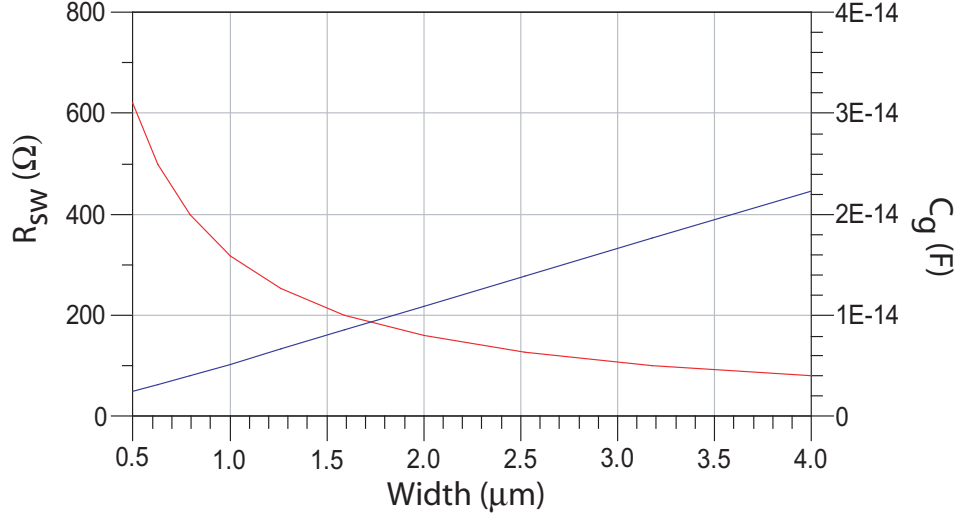


Figure 4.16: Simulated drain-source resistance versus device width of a NFET switch with $L_g = 120nm$ and 4 fingers. The left axis shows channel capacitance.

switch is given by Equation 4.4 [79]:

$$\Delta V_{hold} = -\frac{C_{ox}WL(V_{DD} - V_{tp} - V_{in})}{2C_{hold}} \quad (4.4)$$

where V_{tp} is the p-channel threshold voltage.

The simulation results of the NFET switch based SHA are given in Table 4.2. Again, the input and output load impedances are given for a single-ended half of the pseudo-differential SHA. The total power consumption of $57\mu W$ is for the pseudo-differential SHA. The resulting bandwidth is larger than required for this design. Although the bias current cannot be reduced because of slewing requirements, the NFET switch could be made smaller, which would reduce the amount of clock power consumed in the stage.

4.3 Clock Generation Circuitry

In the previous section the serial-to-parallel circuitry was presented. In this section the ten-phase clock divider that generates the clocks for the serial-to-parallel function

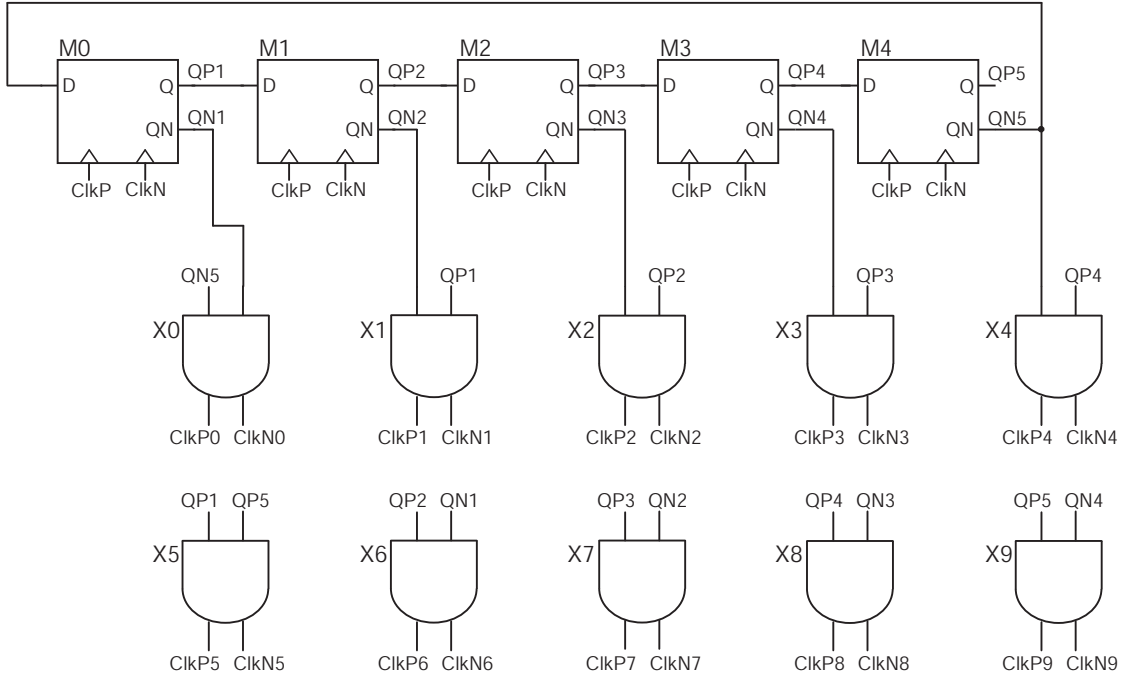


Figure 4.17: The ten phase clock divider constructed from D-flip-flops and NAND gates.

is described [59]. Figure 4.17 shows how the ten phase clock signals can be generated using D-flip-flops and NAND gates. The circuit shown, uses five D-flip-flops each with complementary phase outputs. It is possible to generate ten phase clock signals using either five or ten D-flip-flops cascaded sequentially. When 10 D-flip-flops are used the phases are available directly at the Q outputs of the D-flip-flops. When 5 D-flip-flops are used, AND gates are required to generate the phases from the complementary outputs Q and QN. The latter approach is more straightforward and also allows more flexibility by independently assigning the drive strength to the NAND gates. The D-flip-flops are driven by the full rate clock from off-chip into their clock port. This is important in high speed digital logic design as it avoids the application of the clock directly to the D-flip-flop input [89].

In Tables 4.1 and 4.2, the loading capacitances of the SHAs were given. The total capacitive load for each clock phase can be calculated using this information and the number of clock inputs given in the circuit diagram in Figure 4.10. Using the load capacitance, the required drive strength of the NAND gates can be optimally sized for the target load. Table 4.3 shows the total load presented to each clock phase.

Table 4.3: The capacitive load presented to the different clock outputs

Clock Name	Load Capacitance
ClkP{0-7}	25.3fF
ClkN{0-7}	25.3fF
ClkP8	unused
ClkN8	unused
ClkP9	347.5fF
ClkN9	173.8fF

Table 4.4: The timing results of the NAND simulation.

	In(ClkP,Vdd)	In(ClkP,Vdd)	In(ClkP,ClkP)	In(ClkP,ClkP)
t_{pLH}	23ps	75ps	14ps	82ps
t_{pHL}	52ps	52ps	58ps	39ps
t_r	37ps	45ps	20ps	44ps
t_f	85ps	30ps	84ps	26ps

Clk9 requires a significantly larger output driver than the other gates.

The NAND gate circuit is shown in Figure 4.18. The PFETs are scaled to be three times the width of NFETs. The sizes are adjusted to maintain a worst case rise-time or fall-time less than 100ps, 10% of the 1ns symbol time. Figure 4.19 shows the simulation results of the NAND circuit driving a typical load. Table 4.4 summarizes the results.

4.3.1 “Power-PC” D-flip-flop

The D-flip-flop is based on what is commonly referred to as the PowerPC flip-flop architecture, and is known to be a fast architecture with low power consumption [90] [91]. Figure 4.20 shows the transistor sizing chosen for this design in 120 nm CMOS. The design consists of minimum sized transmission gate, TG2, minimum sized feedback transistors, M3–M6 and M9–M12, and a minimum sized second inverter. The PFETs M1 and those in TG1 were swept over several sizes to optimize gate speed. The third, fourth and fifth inverters are progressively scaled to be able to

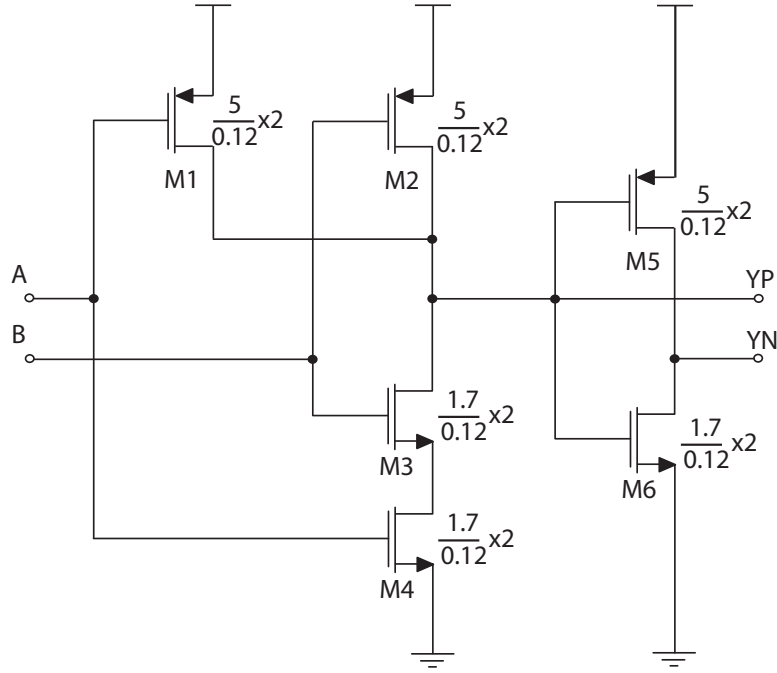


Figure 4.18: The NAND circuit used in the 10 phase clock generator. Outputs are scaled to drive SHAs.

meet the drive requirements. In this case the positive output Q is required to drive another flip-flop and two NAND gates, and the negative output must drive two NAND gates.

The advantages of the PowerPC flip-flop architecture are power efficiency and speed; the disadvantage is that it is not a differential architecture. Being single ended, the output QN must pass through one additional inverter after QP, incurring an additional propagation delay. A small time offset between the QP and QN outputs results and causes positive and negative current to flow through the power supply at the different transition times. With a differential approach, the supply current would theoretically cancel at the power supply.

Figure 4.21 shows the simulation results of the complete ten phase divider circuit for the second and third clock phases. The simulation includes loads on each clock as given in Table 4.3. The circuit is clocked at 1 GHz. The rise time is measured to be 50 ps and the fall time 35 ps. These results exceed the design goal of keeping the rise and fall times less than 100 ps.

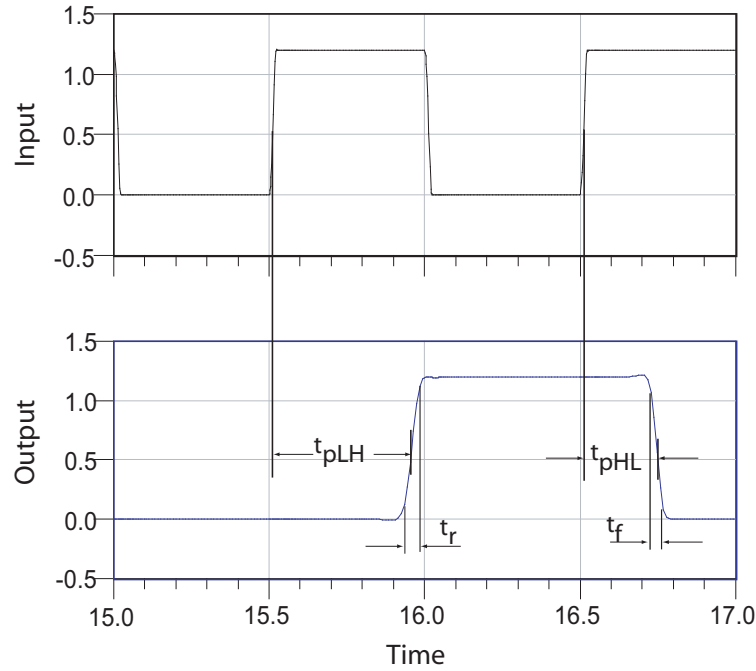


Figure 4.19: The simulation results of the NAND gate.

4.4 IC Peripheral Circuit Designs

In addition to the primary circuit functions included on the prototype IC, several peripheral circuits are required to fully implement a testable chip. The primary functions of these circuits are to: buffer inputs while filtering out noise; amplify output signals to drive off-chip loads; and distribute supply and bias voltages around the chip.

In any mixed signal IC that contains both strong digital signals as well as noise sensitive analog circuits, noise coupling is an issue. In this prototype a bond-wired package is the target test platform due to the large number of test pins. In bringing a strong clock signal into the IC through bond-wires, there is a high probability of mutual inductive coupling occurring through the bond-wires. This typically occurs in the frequency range of 100 MHz to 2 GHz [92]. Historically RF circuits have operated above this frequency range while digital circuits have operated below it, making the problem negligible. However, with the advent of ultra-wideband signal processing requirements, the baseband frequency has become high enough to warrant additional

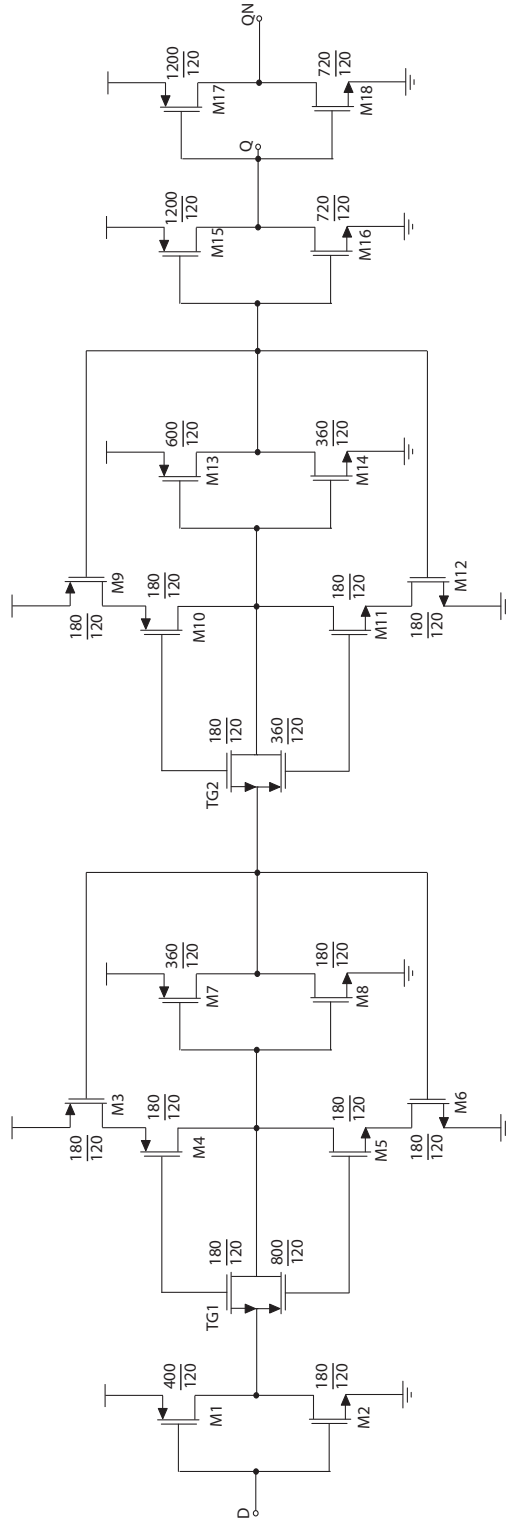


Figure 4.20: The “PowerPC” D-FlipFlop design used in the 10 phase clock generation.

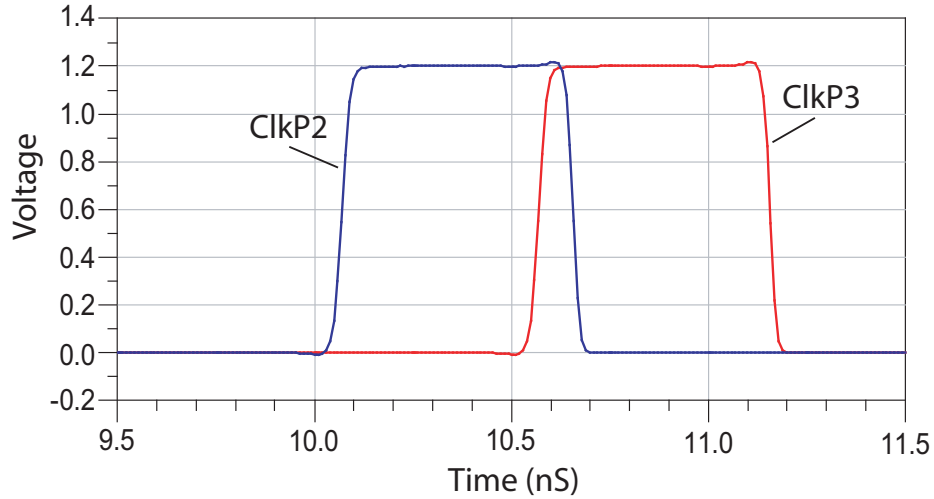


Figure 4.21: Simulation results of the ten-phase clock divider showing clock phases 2 and 3 precautions.

Printed circuit boards (PCBs) typically employ filtering through the use of surface mount (SMT) series resistors, ferrite beads and bypass capacitors; however, because of the the finite size of SMT packages, they cannot effectively filter above ≈ 500 MHz [93]. In addition, noise that couples into the IC through the bond-wires cannot be removed by external PCB filters. Thus IC pins that do not need to pass signals in these bands should employ on-chip filtering to remove any unwanted bond-wire coupled signal.

Voltage biased pins, which do not carry much current (typically less than $10 \mu\text{A}$) allow low frequency low-pass-filtering that eliminates much of the coupled noise . Usually a low impedance off-chip supply or DAC generates the bias voltages with the target sink being a high impedance on-chip node. By including a large series resistor and a shunt capacitor to act as a low pass filter, bond-wire resonances are damped and high frequency energy eliminated. Figure 4.22 shows the voltage biased filter used on all voltage bias pins in the prototype. It has a low pass corner frequency of 1.5 MHz.

It is also possible to add a bias filter to current biased pins. However, in this case, the currents are larger and the series resistor must be reduced in size to avoid a significant voltage drop. The current biased pins in this design are scaled to handle

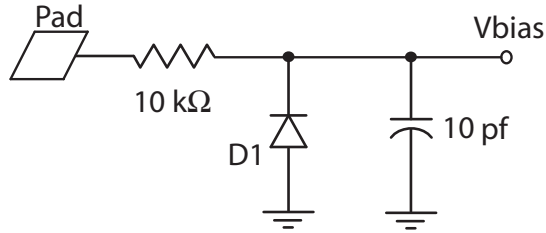


Figure 4.22: Noise Filter and Diode Latch-up protection circuit for voltage biased pads.

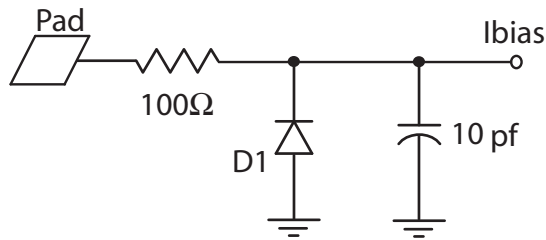


Figure 4.23: Noise Filter and Diode Latch-up protection circuit for current biased pads.

values between $100\ \mu\text{A}$ and $1\ \text{mA}$. Thus a $100\ \Omega$ series resistor has little effect on the overall function. Figure 4.23 shows the current bias filter used in this design. The corner frequency is $159\ \text{MHz}$ which is still low enough to provide some attenuation to any bond-wire coupled resonances at higher frequencies.

In addition to filtering the bias pins, electromagnetic radiation from high speed digital signals can be reduced by including on-chip terminations [92]. Figure 4.24 shows the method employed to terminate high-speed analog and digital input signals on the prototype chip. Including $50\ \Omega$ resistors on the die helps to provide clean sample edges with minimal overshoot to the input clock and discrete-time signals. The two on-chip resistors reference an external ground, rather than IC ground so that common mode energy, and mismatch between the resistors, does not create a strong substrate path for the high-speed signal.

Four power supply domains are used in the IC to separate the noisy digital sections from the analog sections. The four domains are, digital, which contains the clock generation circuitry, mixed signal, which contain all of the sample-and-holds, and multiplexers, analog, which contains the functions of the FFT SFG, and instrumentation, which contains the instrumentation amplifiers, a driver amplifiers. All of the

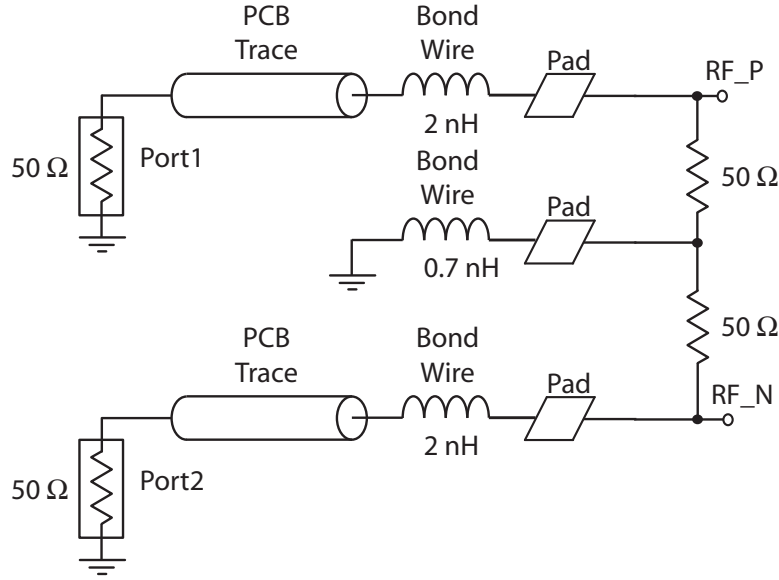


Figure 4.24: On chip 50-Ohm termination reduces RF coupling to substrate.

supplies operate at 1.2Volts with the exception of the instrumentation supply which operates at 3.3Volts.

4.4.1 Driver Amplifiers

The output of the DT FFT processor core is only designed to drive loads up to a few tens of femto-farads. To interface and test the processor with 50 Ω test equipment, buffer amplifiers must be inserted between the processor and the pad edges. The buffer amplifier circuitry is not considered in the power consumption of the proof-of-concept DT FFT processor core, so more power consumptive circuits based on available 3.3V transistors are used. Separate power supply, ground and bias pins isolate this portion of the IC from the core. The goal of this interface circuitry is to provide current gain while maintaining unity voltage gain and contributing minimal noise or distortion. A second goal is to achieve a sufficient bandwidth in the instrumentation circuit so as not to filter the output signals from the core. For this a design goal of $f_c = 1$ GHz was chosen.

In this design, the buffer amplifier is broken up into three sections, an impedance buffer amplifier with level shift, an analog switch, and a 50 Ω driver amplifier. Each of

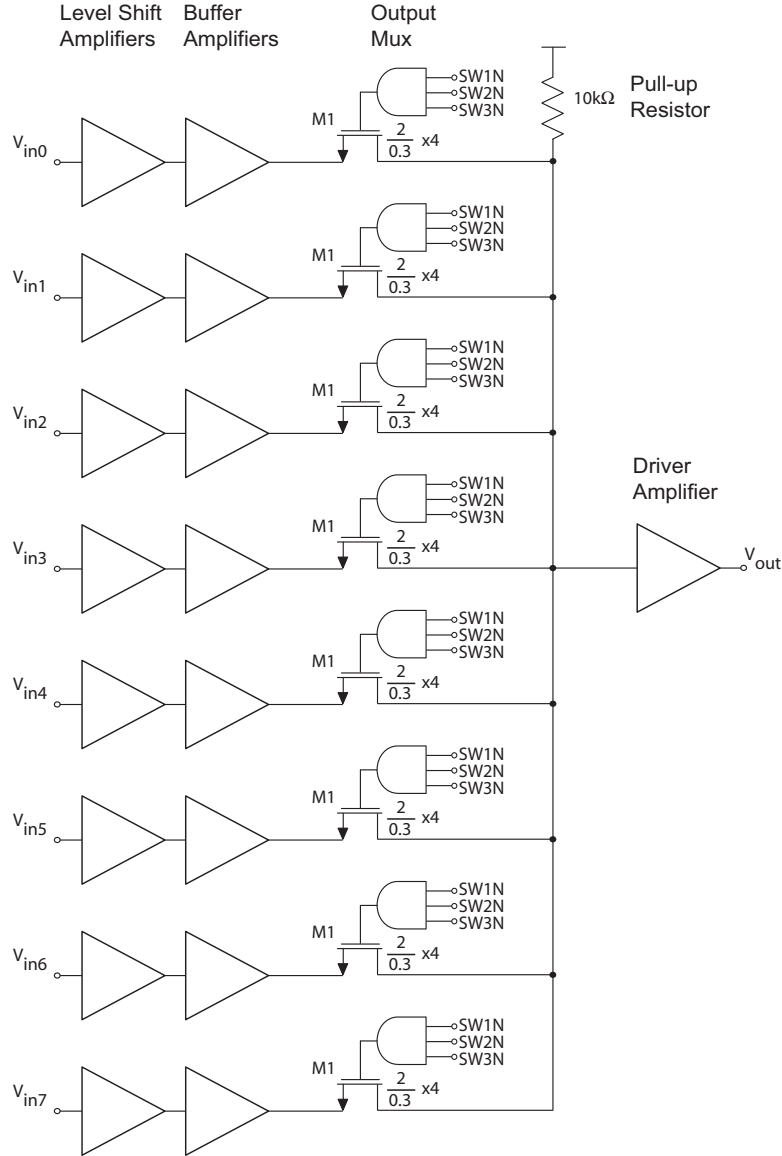


Figure 4.25: The instrumentation mux and driver amplifier consists of the input level shift amplifier, impedance buffer amplifier, output mux, and 50Ω driver amplifier.

the eight parallel outputs of the DT FFT processor core are connected to an individual buffer amplifier, followed by an analog switch that routes the selected output to the driver amplifier. Figure 4.25 illustrates the approach.

The first amplifier stage is a level shift stage. Since a primary goal of the instrumentation amplifiers is to preserve the signal from the DT FFT processor core without

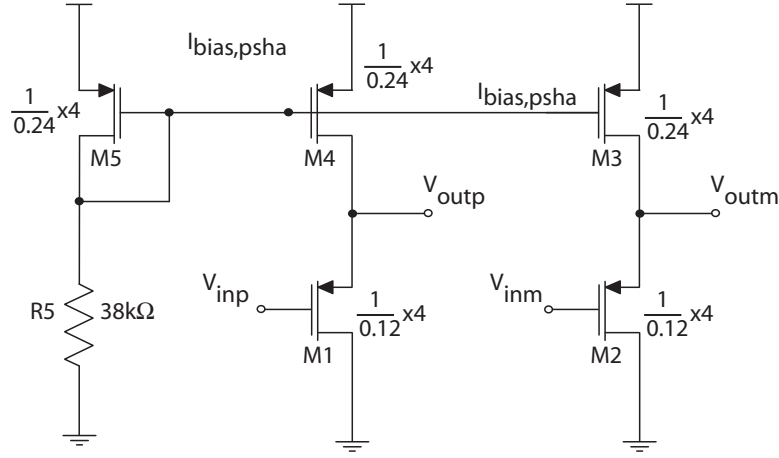


Figure 4.26: The instrumentation level shift amplifier.

adding any additional distortion, it is important to move the signal voltage range into an optimal range for use with the 3.3V circuits. After the buffer amplifiers were designed, simulations showed that linearity suffered at low common-mode input voltage levels below 400mV but worked well between 400mV and 2.2V. Thus a PFET source follower with a 500mV shift moves core common-mode output voltages that are as low as 0V up to 500mV, within the optimal range of the buffers.

Figure 4.26 shows the circuit diagram of the PFET level shift buffer amplifier. It is a pseudo-differential source follower circuit. Bias current of $70\mu\text{A}$ is provided by a fixed resistor R5 connected to the current mirror formed by M3, M4 and M5. The input capacitance is 10fF per side.

The next stage following the level shift buffer is a wideband amplifier. This amplifier has an input capacitance of 10fF and output drive capability of 200fF. To achieve the wide bandwidth, a transimpedance feedback amplifier approach is used. Wideband driver amplifiers typically use either inductive peaking or feedback to achieve their wide bandwidth. This helps to reduce the large inter-amplifier parasitic capacitance that reduces bandwidth in drivers that must have a large, fast output stage. Designs in [94–96] use inductive peaking of the interstage amplifiers, whereas; [97] uses resistive feedback and [98] uses transconductive feedback. The latter method, transconductive feedback, is employed here as shown in Figure 4.27.

Figure 4.28 shows the transistor-level realization of the transconductive feedback am-

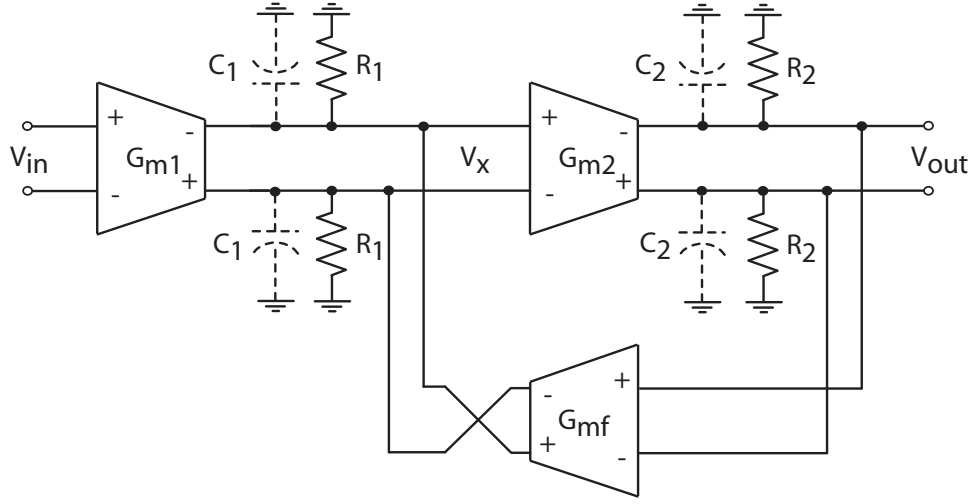


Figure 4.27: The transimpedance feedback amplifier extends amplifier bandwidth.

plifier. Physical resistors are included at each interstage node to establish the output operating impedance and maintain a wide-bandwidth.

The output of the first wideband amplifier is routed to the 50Ω driver amplifier by the analog switches. The switches are controlled by three digital bits routed from pads and driven by the off-chip microcontroller. These 3 bits feed an AND gate which drives the NFET switch that routes the analog signal. A single $10k\Omega$ pull-up resistor on the right side of the switch establishes a finite input impedance for the 50Ω driver amplifier and pulls current through the closed switch. The output of the other channels see an open switch (high impedance) and negligible current flows from them to the pull-up resistor.

In the final stage, the driver amplifier, applies current gain to the signal to drive 50Ω . A transimpedance feedback amplifier similar in architecture to the buffer amplifier is employed. The primary difference is that the resistances are lower and the bias currents are higher. Simulation results show that this stage can drive an external 50Ω load with $600mV_{pk-pk}$ swing.

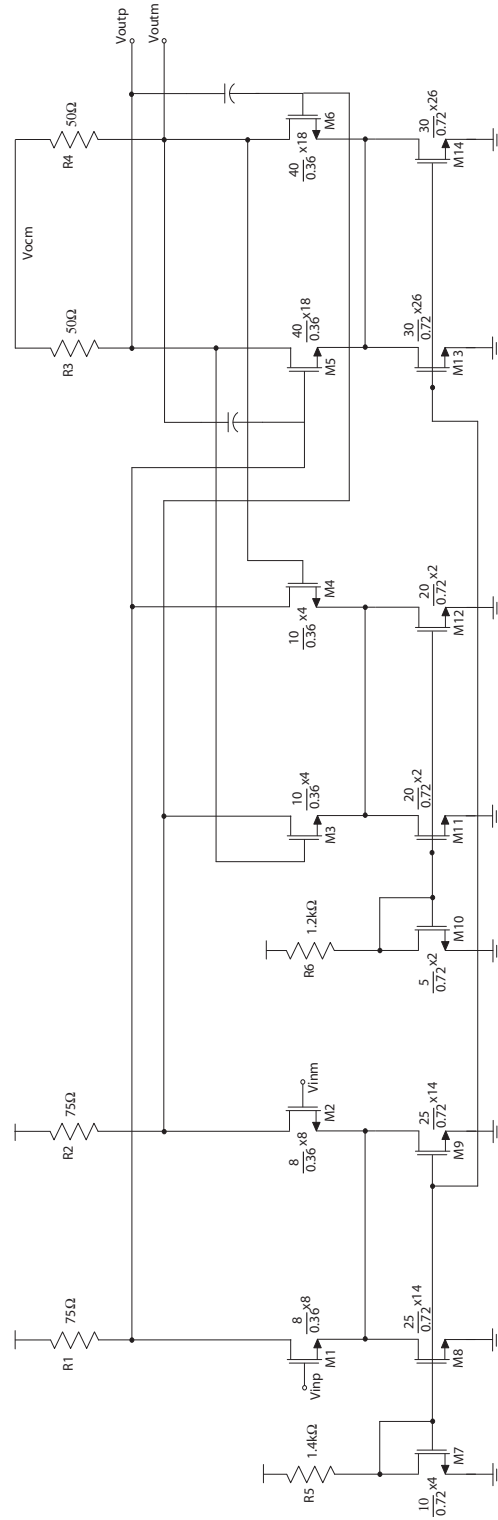


Figure 4.29: The 50 Ω output impedance driver amplifier.

4.5 IC Layout

The layout of the circuit design is a critical part of the mixed signal design process. Traditionally analog circuits have used a full-custom layout approach in which transistors and traces are individually placed and routed by hand to optimize the performance of the circuit. Alternatively, in purely digital circuits, automated algorithms make place and route decisions. As a high speed mixed signal IC, this design is too complex for full custom layout, and yet it requires the care of full custom layout to meet the speed requirements. The compromise selected is to use a full-custom layout approach for the individual circuits such as the sample-and-hold and butterfly structure, but then to place the larger blocks and the interconnect between them on a loose grid. The loose grid reduces the density of the layout but allows for the layout and interconnect to be added quickly.

Figure 4.30 shows the layout of the full IC, including the DT FFT processor, pads, terminations, interface circuits and driver amplifiers. The compact DT FFT processor core is located near the left center of the die. The remaining portions of the die are more spread out because the area of the die is limited by the number of pads. Figure 4.31 focuses on the layout of the DT FFT processor core. The serial input signal are brought in on the left and fed to the input of the NFET switch sample and hold bank. In this stage, serial data samples are converted to parallel. In the following stages, the parallel signal progresses from left to right across the layout. Thus each section is constructed of vertical columns of repeated blocks.

Because the signal and clock inputs are the high speed signals with a large input power, these are the most important to isolate and avoid coupling to other parts of the IC. Thus, these inputs are allocated the shortest interconnect bondwires, as shown in Figure 4.32, the wirebonding diagram of the DT FFT processor. For this reason they are placed on the left edge, where the bondwires will be short and perpendicular to the bondwires at the top and bottom of the IC which carry more sensitive bias signals. The signal output of the IC is on the center right edge of the chip, also in order to minimize bondwire length. The center of the package is a large conductive pad used for ground. A total of nine very short bondwires, called *down-bonds* are used to connect the IC ground to the package ground. The remaining pads are used for bias pins, digital control lines and power supplies.

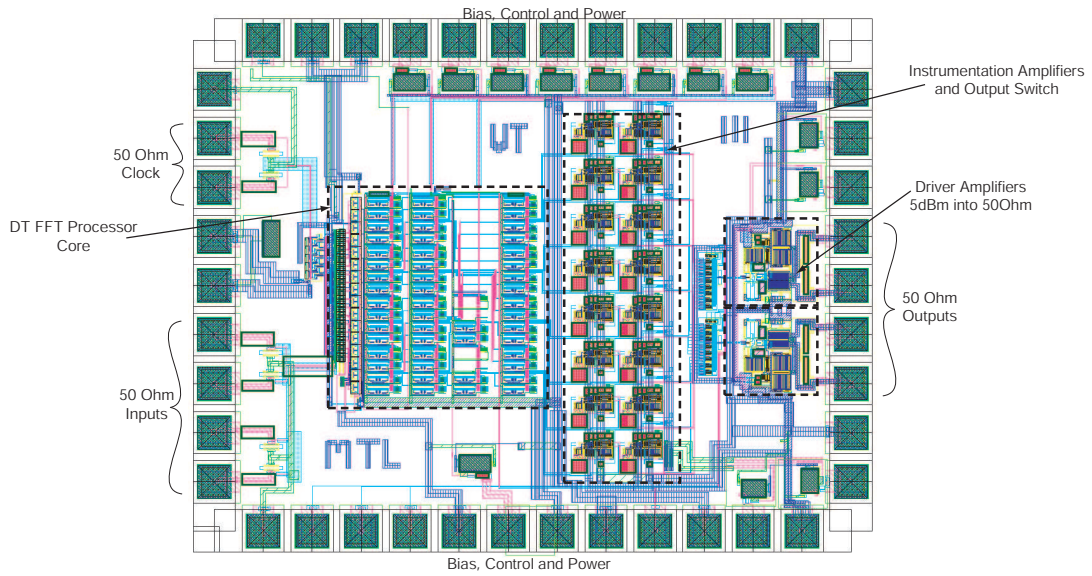


Figure 4.30: The layout of the DT FFT processor with the DT FFT processor core, instrumentation interface circuits and driver amplifiers.

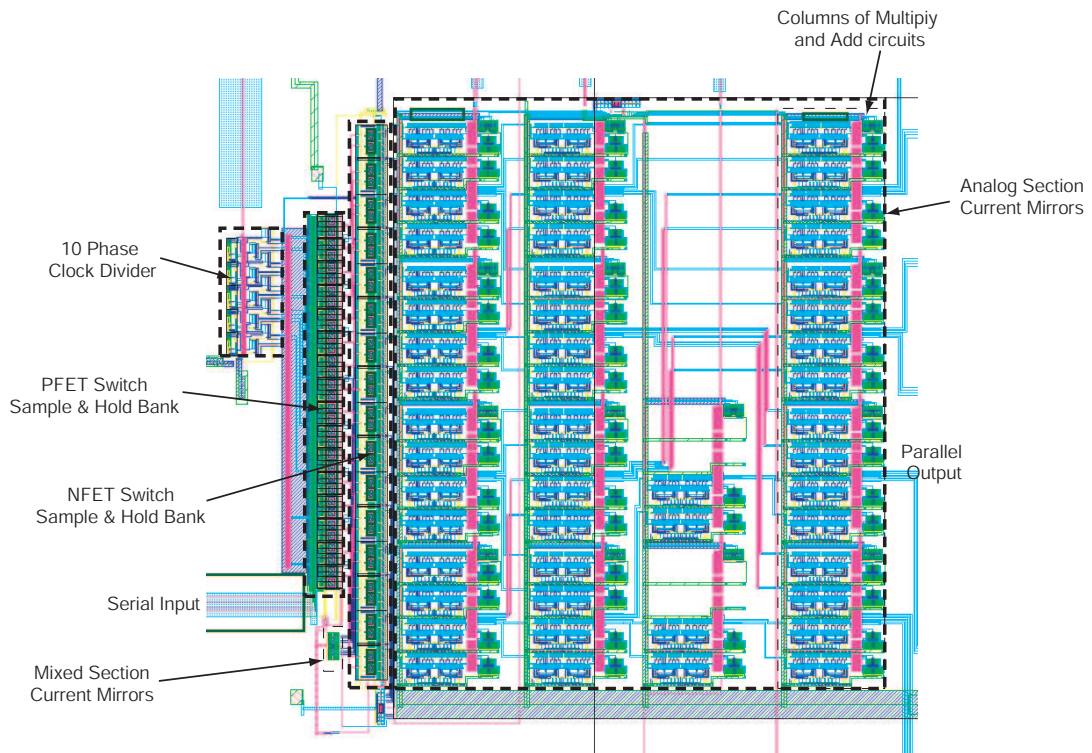


Figure 4.31: The layout of the DT FFT processor core consisting of clock divider, PFET switch SHA bank, NFET switch SHA bank, and four columns of multiply and adder circuits.

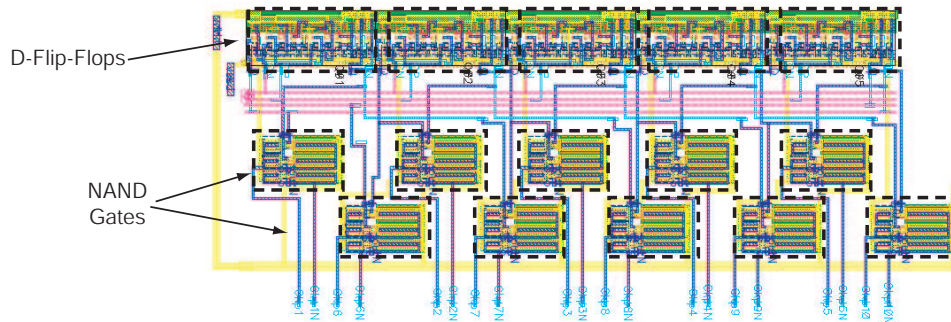


Figure 4.33: The layout of the ten phase clock divider. The D-flip-flops are placed close together to minimize interconnect delay whereas the NAND gates are spaced loosely to aid in the full custom layout process.

The ten phase clock divider is the first circuit function on the left. This can be seen in Figure 4.33 in more detail. The five D-flip-flops are positioned close together to minimize mismatch parasitics that would skew timing. The rest of the clock generator layout looks very similar to the circuit schematic of Figure 4.17 in orientation. The NAND gates are placed with extra space between them to make the trace names clear and to help avoid making wiring errors. Although LVS finds wiring errors, it is time consuming to rely on LVS when performing interconnections at the block level with thousands of transistors. Thus for full custom layout of blocks that are relatively insensitive to long interconnects, it is better to add working space between the outputs.

The layout of the D-flip-flop is shown in Figure 4.34. All of the transistors are packed as close together as possible since as a purely digital circuit, crosstalk between transistors is not as much an issue compared to minimizing interconnect capacitance.

The layout of the PFET switch based sample-and-hold amplifier is shown in Figure 4.35. As a pseudo-differential circuit, two distinct single ended sample-and-hold amplifiers are placed as mirror images of one another around the horizontal axis of symmetry. From left to right are the switch transistors, the hold capacitor and the unity gain buffer amplifier. The capacitor adds physical separation between the mixed-signal portion of the circuit and the analog domain. Vertical rows of alternating polarity substrate contacts are placed here that run the full height of the DT FFT processor core. These serve to isolate substrate switched charge from the sensitive

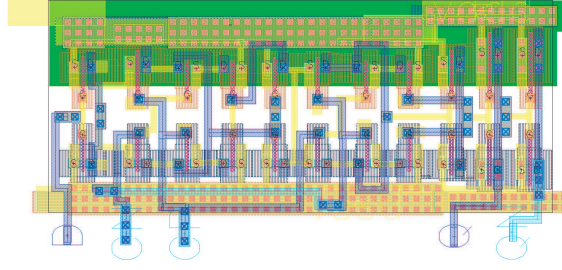


Figure 4.34: The layout of the D-flip-flop is made compact to maximize switching speeds.

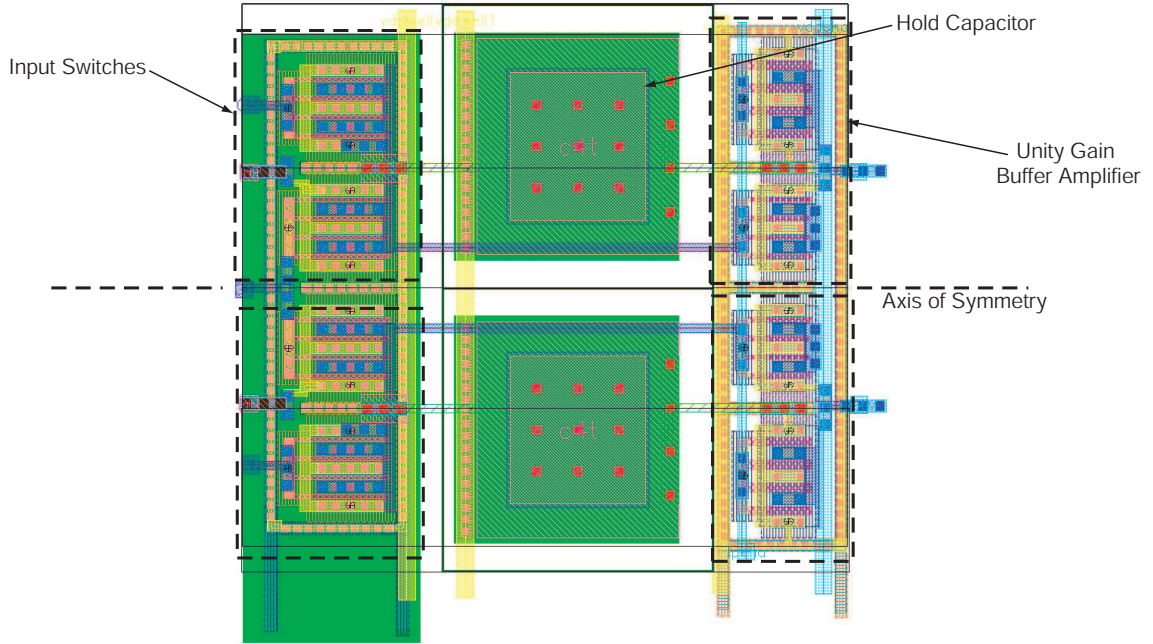


Figure 4.35: The layout of the pseudo-differential sample-and-hold amplifier consists of two single ended sample-and-hold amplifiers placed as mirror images about the horizontal axis of symmetry.

analog portion.

The layout of the complete butterfly structure is shown in Figure 4.36. Eight transconductors and four adders are included. The eight transconductors can be seen on the left, labeled $G_m0 - G_m7$ and the adders can be seen on the right labeled R1 – R4. Also included in this cell is a current mirror that is used to replicate the current bias locally between the eight transconductors. Ground impedance is an issue in this

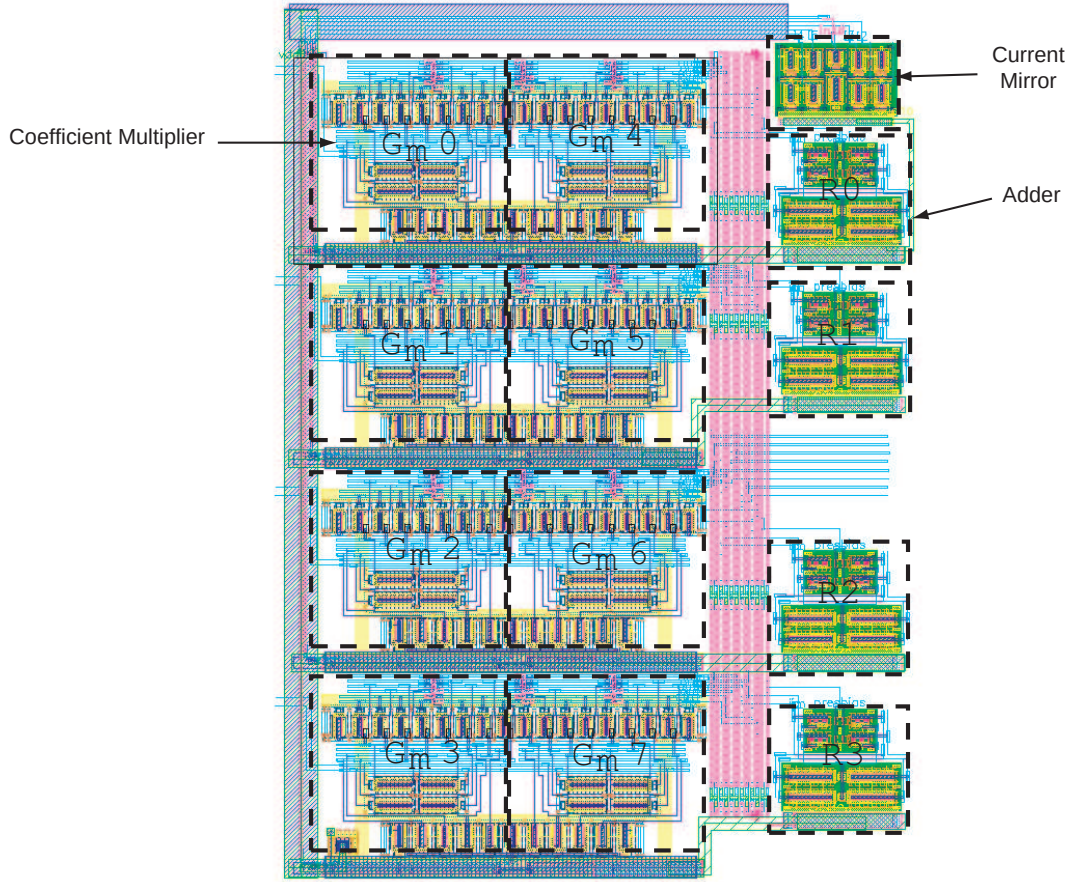


Figure 4.36: The layout of the butterfly structure consists of G_m cells, adders and a current mirror.

circuit, so large, multi-layer ground traces are used to minimize loss.

The layout of a pair of linear transconductors is shown in Figure 4.37. The rows of current steering switches, linear degeneration transistors and current references can be seen in the figure. The linear degeneration transistors are arranged in a common-centroid pattern to minimize mismatches. The current references and the current steering switches use an alternating pattern to separate the transistors pairs across the row in an attempt to minimize mismatch. The rows of input and output traces are attached in a bus-like manner to minimize area.

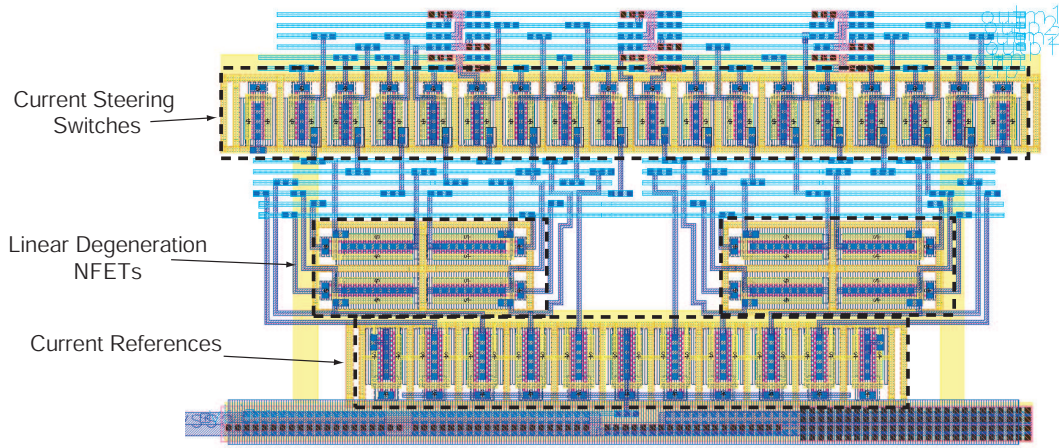


Figure 4.37: The layout of a pair of G_m cells. Common centroid and interleaving techniques are applied to minimize mismatch.

4.6 Summary

The final section of this chapter has presented the layout of the first prototype DT FFT processor IC. The first part of this chapter focused on the circuit design of the critical signal processing stages, the multiplier, adder and sample-and-hold amplifiers. Next, this chapter presented the circuit design of additional circuitry and the peripheral circuits that allow the circuit to be tested. The final section of this chapter presented the layout of the entire IC and the individual cells. In the following chapter, measurement results from the fabricated chip will be presented.

Chapter 5

Measurement Results

In this chapter, the measurement results from the initial DT FFT processor prototype are presented. These measurements of the proof of concept IC validate the functionality for the DT FFT approach.

The test chip was designed and fabricated in the Jazz CA13 $0.13\mu\text{m}$ CMOS process, which has a single poly and six metal layers. Figure 5.1, shows a photograph of the fabricated die with key sections labeled. The processor core is contained within the $450\mu\text{m} \times 450\mu\text{m}$ block labeled “DT FFT Core”. The interface circuitry, including: 50Ω input buffers, bias filters and instrumentation multiplexer and driver amplifiers are also shown. Ultimately the IC is pad limited and thus there are various areas of metal fill around the bias filters and driver amplifiers; this can be seen as the gold grid pattern outside the functional blocks.

After the fabricated die were received from the foundry, the parts were wire-bonded to an MLF5x5 28-pin open-face package at RF Micro Devices, Greensboro, N.C., and the packaged die were soldered to a printed circuit board (PCB) for testing. Multiple copies of each of the three variants were tested. The description of the test setup follows.

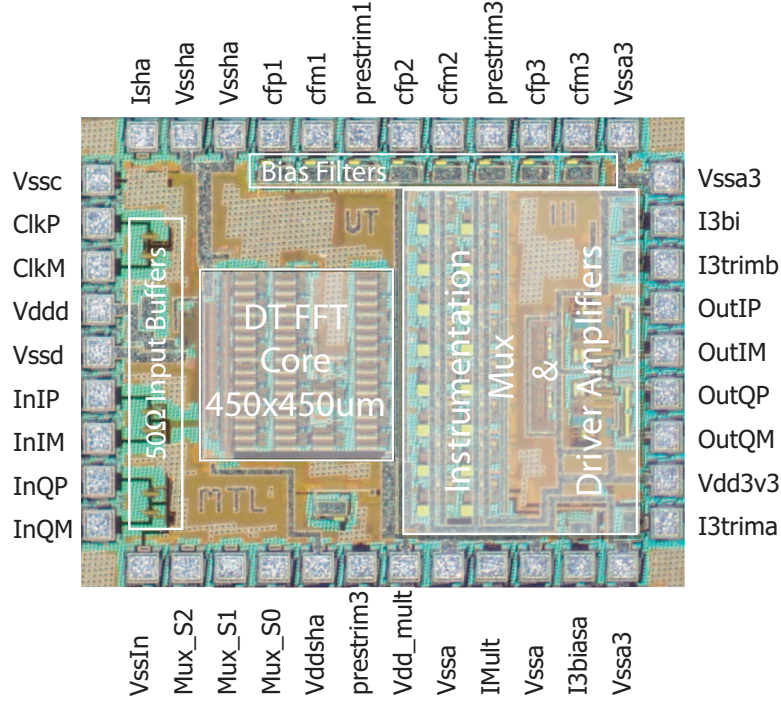


Figure 5.1: The die photograph of the DT FFT processor prototype with pins and key sections labeled.

5.1 Test Setup

In order to test the DT FFT processor at its target operating speed of 1 GSps, careful planning of the test setup was required. The goal is to apply full data-rate signals to the processor that emulate the expected conditions seen in a typical OFDM receiver. For the purposes of measuring distortion contributed by the processor, the block diagram shown in Figure 5.2 was used. Here, stimulus signals are created in MATLAB and applied both to the physical measurement setup and an ideal FFT within MATLAB. After passing through the physical measurement setup, frame and symbol timing recovery are performed. These measured results can then be compared against the ideal case. Equation 5.1 is used to calculate the Signal to Noise and Distortion Ratio (SNDR) based on this approach.

$$SNDR = 20 \log_{10} \frac{\sqrt{\frac{1}{N} \sum_{k=1}^N V_{ideal}(k)^2}}{\sqrt{\frac{1}{N} \sum_{k=1}^N (V_{measured}(k) - V_{ideal}(k))^2}} \quad (5.1)$$

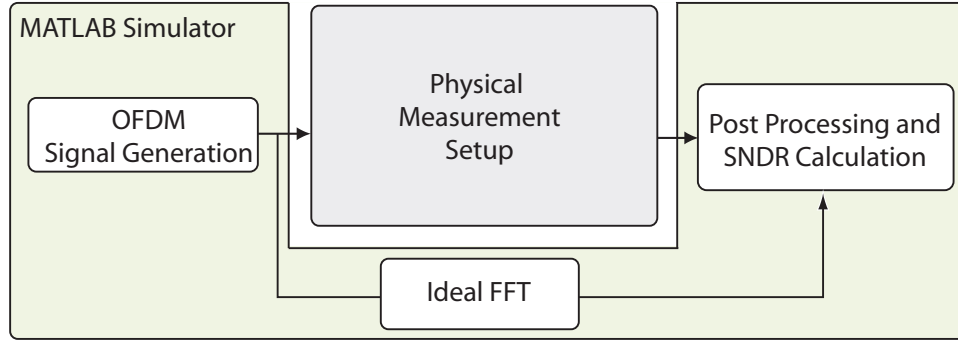


Figure 5.2: The signal generation and measurement setup used for the Discrete-Time FFT processor.

When the input magnitude of the sub-channels are swept versus SNDR, the measure of dynamic range can be found. Dynamic range is defined as the range of input magnitudes for which the SNDR is greater than 7 dB, a value which ensures a bit error rate of less than 1×10^{-5} for OFDM [1].

Figure 5.3 shows the physical measurement setup used for the measurements. Differential I and Q signals were generated in the Tektronix AWG7102 arbitrary waveform generator (AWG). It was assumed that the signal had already been magnitude adjusted to the full scale input of the DT FFT processor by the automatic gain control, and that it had been sampled by the front end sample-and-hold into discrete sampled data. The arbitrary waveform generator (AWG) used needed to have 4 output channels and to be capable of emulating a 1 GSps sample-and-hold. A differential clock signal was also derived from the AWG. To provide the target data rate of 1 GSps to the DT FFT processor, the AWG was oversampled and clocked at 2 GSps to provide crisp edged sample to the processor inputs that emulate those that would be found inside the receiver at the interface to the discrete-time signaling domain.

Table 5.1 shows the relevant specifications of the Tektronix AWG7102, one of the fastest AWGs currently available. As a 10 GSps generator, it has more than enough speed to adequately test the DT FFT processor. It also has more than sufficient memory to output the long OFDM symbol streams generated in MATLAB. With dual differential 50Ω outputs, the generator is easily matched to pass high frequency signals to the 50Ω differential inputs of the IC. The drawbacks to this generator is that it outputs 8-bits of resolution instead of the target 10-bits and its spurious

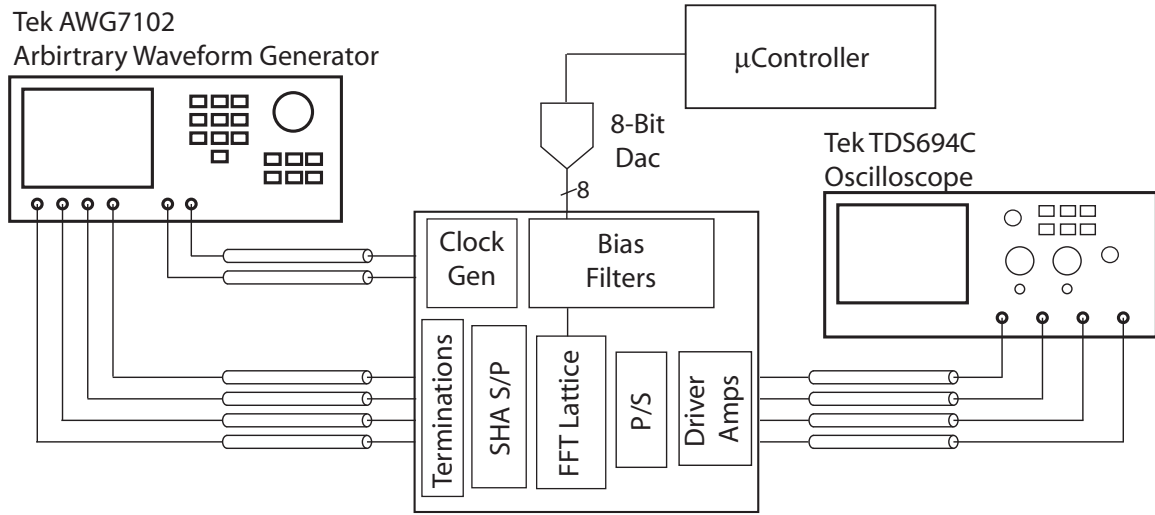


Figure 5.3: The physical measurement setup used to measure the Discrete-Time FFT Processor.

free dynamic range (SFDR) is only 45 dB instead of the ideal value of 70 dB for a distortion free test setup. This required additional care to ensure that the AWG was not contributing distortion to the measurement results. The output amplitude can be varied between 400mV_{pk-pk} to 1V_{pk-pk} which exceeds the DT FFT processor which was designed for 400mV_{pk-pk} . This allows the full 8-bits of resolution to be applied to the signal range of the processor without wasting resolution on gain adjustment. Although the generator is capable of applying a DC offset of up to 500mV to output signals, this is not enough to supply the required bias of 700mV-900mV needed at the inputs of the IC. Thus external bias tees were required between the generator and the IC.

The Tektronix AWG7102 was also used to generate the master clock for the prototype IC. Although it would be more flexible to use an independent master clock, the AWG7102 does not have the capability to accept an external trigger. Thus, in order to maintain good timing correlation between the master clock and the AWG output, was is important to use the AWG7102 to generate all clock signals. The clock inputs on the DT FFT processor are differential shunt 50Ω terminations, as shown in Figure 4.24, followed by clock generation circuitry. The output of the AWG7102 clock generator is a differential 0V to 1.2V square wave into 50Ω s. This conveniently matches the clocking requirements of the prototype IC.

Table 5.1: The specifications of the Tek AWG7102 Arbitrary Waveform Generator

Description	Specification
Channels	2 Differential
Sample Rate	10MSps to 10GSps
Waveform Length	2 x 32MSamples
Resolution	8-bits
SFDR	45dB
Output Impedance	50 Ω
Amplitude	1V _{pk-pk} maximum
DC Offset	$\pm 0.5V$
Trigger	Output Only
Internal clock phase noise	-90dBc at 100kHz

The prototype DT FFT output measurement requirements are less stringent than the input requirements due to decimation in time created by the internal serial-to-parallel operation. The expected speed reduction is a factor of 10. At an input symbol rate of 1 GSps, this translates to 100 MSps at the output. The test equipment is an order of magnitude better than the device under test (DUT) and contributes minimal distortion. Since the DT FFT processor shows a simulated bandwidth of 700 MHz, a 7 GHz bandwidth for the oscilloscope is quite sufficient.

Table 5.2 shows the specifications of the Tektronix TDS694C Oscilloscope used in the test setup. This digital sampling scope captures signals in the analog domain and then allows the digital post-processing in MATLAB. The sample rate of 10 GSps allows the DT FFT processor outputs to be oversampled and then symbol synchronized. Four 50 Ω oscilloscope channels allow the four outputs of the prototype IC to be connected without the need for baluns. The differential channels are converted to single-ended in MATLAB. The 3 GHz real-time bandwidth does degrade the rise and fall times of the output of the processor, but not significantly. The vertical resolution of 8-bits is more than sufficient because the OFDM demodulated output of the DT FFT processor is a QPSK or BPSK signal.

To control the output multiplexer, set the bias DAC levels and control the DACs that set the coefficients used in the FFT signal flow graph, a micro-controller was connected to the two DACs. Three eight-bit registers within the micro-controller contained the

Table 5.2: The specifications of the Tek TDS694C Oscilloscope

Total Channels	4
Input Impedance	50 Ω s
Real-time Bandwidth	3 GHz
Sample Rate	10 GSps
Maximum Record Length	30 kSamples
Vertical Resolution	8 Bits
Vertical Sensitivity	1 mV/div
Time Accuracy	15 ps

multiplication coefficients of the FFT signal flow graph. Eight additional registers were used to set bias voltages and currents within the test chip and allow flexibility in the setting of operating conditions. The instrumentation output multiplexer was also controlled by the micro-controller.

The bias DACs used were two Analog Devices AD5308, an 8-bit 8-channel DAC. These were left off-chip in the prototype, because their location is not critical, and including them on-chip would increase the implementation risk. The specifications of the DACs are not critical, and many alternate off the shelf parts would work.

As a bias generation DAC, the AD5308 is ideal. It incorporates eight distinct resistive ladder converters and a low output bandwidth of 200 kHz. It also operates from a 1.4Volt supply. These specifications are similar to what would be included in an on-chip version in later revisions of the DT FFT processor. The low bandwidth of the bias DAC also reduces the likelihood of high frequency spurs. The DAC was connected to the IC via copper traces on the PCB and located less than one half inch away from the processor. To limit the noise contribution to the IC, a low pass RC filter was included on each PCB bias line, consisting of a series 2 k Ω resistor and a shunt 0.1 μ F capacitor. These create an 800 Hz corner frequency and a kT/C noise voltage of 200nV_{rms}, which is insignificant [79].

The digital codes for the DAC were generated by an Microchip PIC18F252 micro-controller and sent via serial-peripheral interface (SPI) bus to the DAC. The bit-rate of the SPI bus is 200 kHz, a rate determined to be slow enough not to couple into the prototype IC.

Table 5.3: The specifications of the AD5308 bias generation DAC

Resolution	8 Bits
Relative Accuracy	± 0.15 LSB
Differential Nonlinearity	± 0.02 LSB
Offset Error	± 60 mV
Gain Error	± 0.30 % of FSR
DC Output Impedance	0.5Ω s
Supply Voltage	1.4 Volts
Real-time Bandwidth	200 kHz
Digital Interface	SPI

The PIC micro-controller uses an ANSI C code interface with 16, 8-bit variables that store the values of the of the registers for the bias voltages and FFT coefficients. The micro-controller operates at 3 Volts and has 3 Volt I/O logic. The output switch in the instrumentation multiplexer of the prototype is implemented using thick oxide CMOS transistors that are compatible with 3 volt logic. Unfortunately, the AD5308s are not 3 Volt logic compatible and require a logic level translation chip (National 74HC04) to step down the 3 volt logic to 1.4 Volts. A reason for selecting the PIC micro-controller is that it uses ANSI C code, which allows parameterized equations to be included in the code. Based on tuning relationships determined in simulation, first or second order polynomials are coded into the micro-controller that allow the test operator to tune the behavior to find the optimal behavior in the minimal amount of time. Without this correlation of the coefficients to simulated performance, tuning 9 independent variables would have been excessively time consuming.

Of the four power supply domains required by the IC, two are supplied by voltage regulators on the PCB and two are supplied by external laboratory power supplies. To limit complexity in the test setup and the number of power supplies needed, the digital domain 1.2 Volt net from the IC is tied to a fixed 1.2 Volt linear regulator on the PCB. For the same reason, the driver amplifier is connected to a 3.3 Volt fixed regulator. In order to facilitate small adjustments and to explore circuit behavior, the 1.2V mixed signal supply for the serial-to-parallel converter and the 1.2V analog supply for the FFT lattice are brought out separately to external power supplies.

Figure 5.4 shows a photograph of the printed circuit board containing the test chip,

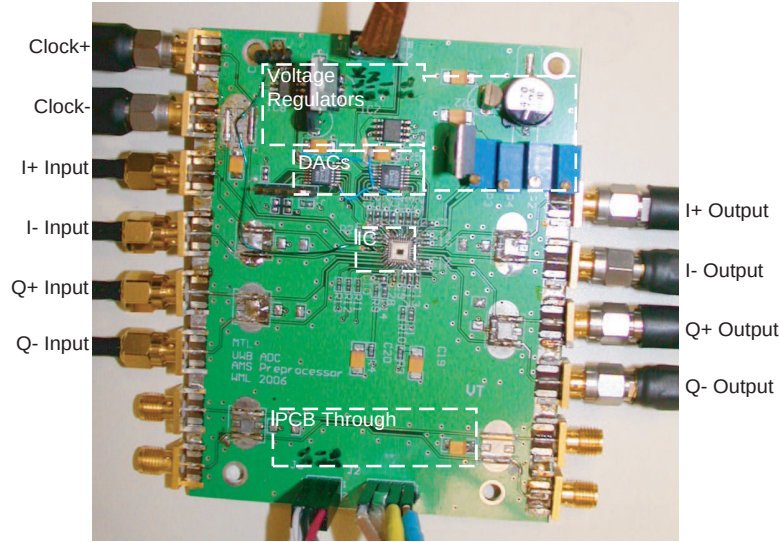


Figure 5.4: The printed circuit board with the test IC, bias DACs and voltage regulators.

micro-controller, two 8-channel, 8-bit DACs, voltage regulators and decoupling capacitors.

The final critical piece of the test setup is the RF cabling. The test setup goal was to maintain better than 18° matching between the four RF input cables and the two clock input cables. This value ensures that the I and Q symbols are well aligned and that the clock is being phased at the correct sample point in each symbol. 18° at 1 GSps is less than 0.2 inch in Teflon cable. Because this is difficult to achieve with off-the-shelf matched coaxial cables, adjustable in-line coaxial delay elements were also used.

In the next section, the test setup is used to evaluate the instrumentation amplifiers, multiplexer and driver amplifiers on the prototype IC.

5.2 Characterization of Instrumentation Amplifiers, Instrumentation Multiplexer and Driver Amplifiers

Before conducting the measurement of the full DT FFT processor, it was necessary to calibrate out the losses and distortion due to the test setup and instrumentation circuitry. This was done here using a “Through Test IC” calibration chip that was laid out and fabricated in addition to the primary proof-of-concept chip with the DT FFT processor. The “Through Test IC” is an IC with only the input buffers, instrumentation multiplexer and driver amplifiers. The layout of the “Through Test IC” is identical to the DT FFT processor IC of Figure 4.30, with the exception that the region labeled *DT FFT Processor Core* is replaced with four horizontal traces connecting the input to the output. This provides a way to test the effects of the bond-wire and packaging, as well as to evaluate the behavior of the multiplexer, instrumentation amplifiers, instrumentation multiplexer and driver amplifier. The measurement results of this chip will be covered in this section. A second test IC, the “Serial-to-Parallel Test IC” includes clock generation circuitry, serial-to-parallel block and instrumentation circuitry, and will be covered in the next section.

First, the S-parameters of the “Through Test IC”, were measured using the Agilent PNA E8364B Network Analyzer. The network analyzer was calibrated with a 50 Ω calibration kit, over a sweep range of 10 MHz to 5 GHz using a continuous wave (CW) tone with a power of -30 dBm. After this, the S-parameters of the test chip for each of the four differential I and Q paths were measured. The results of these measurements are shown in 5.5(a)–(d). In Figure 5.5(a), the S21 plot shows the input to output voltage transfer function. At low frequencies the value of the loss measures -7.9 dB. Since a differential driver amp is used, a single ended measurement is expected to be -6 dB. Thus the actual loss through the channel is -1.9 dB. This does not include the cable losses since they were removed by the network analyzer calibration. The measured bandwidth of the output driver amplifiers is 287 MHz.

Figure 5.5(c) shows the input match in terms of S11. Since the return loss for the input is better than 12 dB for the entire range of interest, the non-ideal effects of the PCB trace, bond-wire and package are not significant. Figure 5.5(b), (d) show the

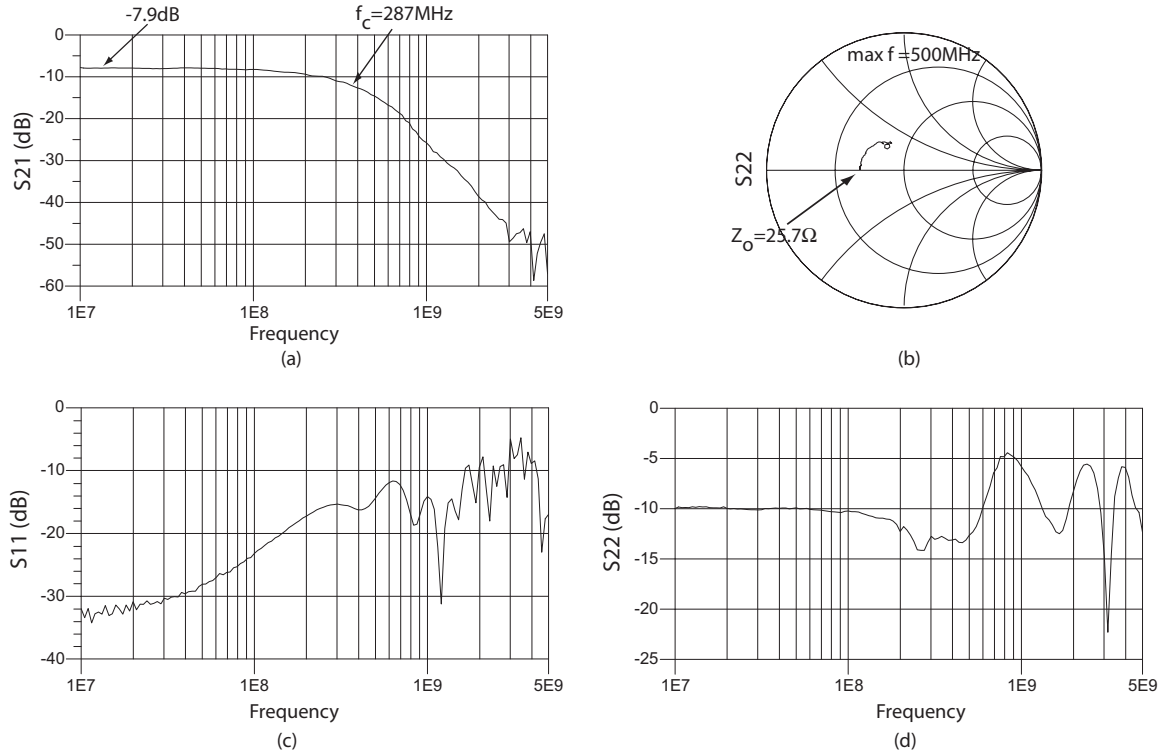


Figure 5.5: Through test IC S-parameters (a) S21 single ended, (b) S22 from 10 MHz to 500 MHz, (c) S11 input match, (d) S22 output match

output match S22. The output match has an issue in that it is nominally 25Ω instead of 50Ω . The reason for this is that the original test plan called for a bias tee with a shunt inductor to pass the DC bias. However, initial measurement results found that this shunt inductor made for a poor S22 output match. To fix this, a bias tee with a shunt 50Ω resistor was used. The lower impedance required the driver amplifier to slew twice the current it was designed to for, moving it out of the desired operating range. This also explains the lower than expected bandwidth in the driver amplifier frequency response.

Having measured the S-parameters of the Through Test IC, the next step was to test the clock generation circuit and serial-to-parallel converter.

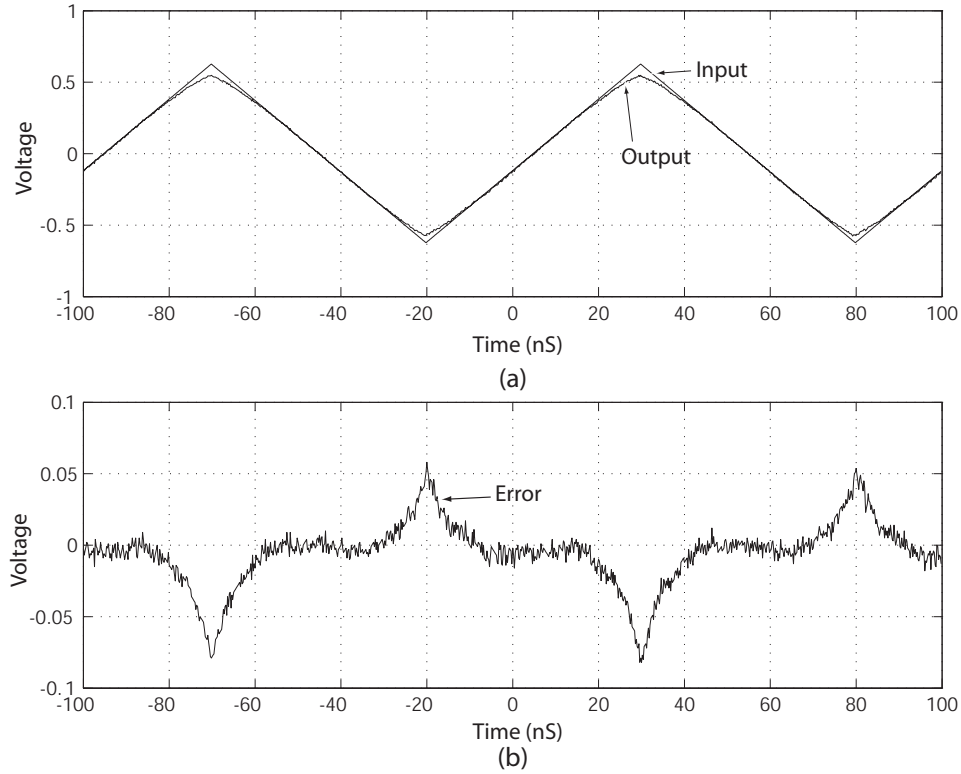


Figure 5.6: The measurement result of a 10 MHz, 600 mV_{pk} triangle wave applied to the through test IC.

5.3 Characterization of the Serial-to-Parallel Converter Test IC

The second test chip variant includes the clock generation circuitry and the serial-to-parallel converter, in addition to the instrumentation circuitry of the “Through Test IC”. The layout of the “Serial-to-Parallel Test IC” is also identical to that of Figure 4.30, except that in this layout the features labeled, *Columns of Multiply and Add Circuits* from Figure 4.31 are replaced by 32 horizontal traces connecting the input and output. Thus the features labeled *10 Phase Clock Divider*, *PFET Switch Sample & Hold Bank*, *NFET Switch Sample & Hold Bank* in Figure 4.31 are included on this test IC in addition to the features of the “Through Test IC”. This test IC allows verification of the functionality of the serial-to-parallel function of the DT FFT Processor.

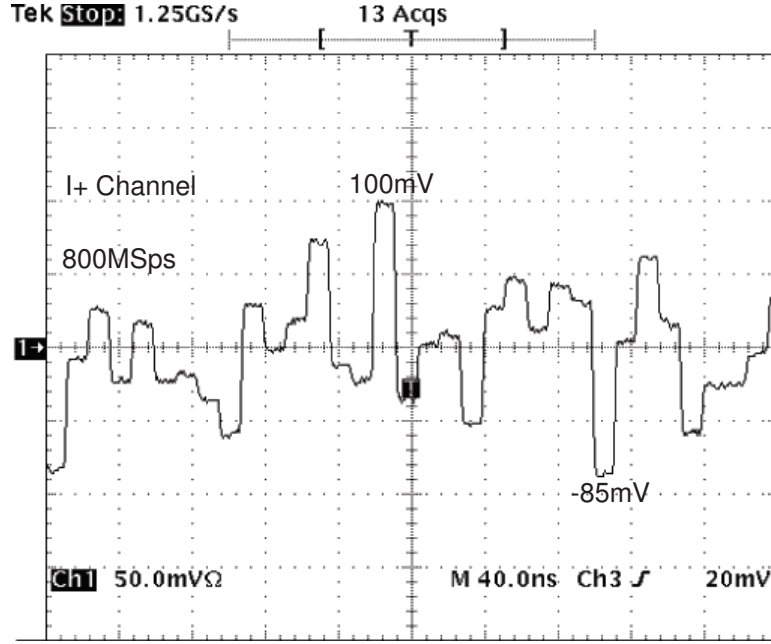


Figure 5.7: The down-sampled OFDM symbol stream measured at the output of the serial-to-parallel converter.

The first test of the serial-to-parallel function was to apply a low frequency triangle wave stimulus signal. The test setup shown previously in Figure 5.3 was again used for this measurement. A 10 MHz, 600 mV_{pk} triangle wave was generated in MATLAB and applied through the AWG to the device under test (DUT). The measured response was captured by the four single ended channels of the oscilloscope and returned to MATLAB where the differential signals were re-combined. The input and output signals are shown in Figure 5.6(a); the error voltage between them is shown in Figure 5.6(b). The amplitude is compressed near the maximum and minimum values of the triangle wave which indicates that an input range of 720mV_{pk-pk} is the maximum linear range of the driver amplifier.

To verify the decimation function of the serial-to-parallel conversion, a second test was performed with OFDM data. An OFDM symbol stream was applied to the DUT input and each of the decimated sub-channels was checked. This verified that the cyclic prefix was being removed and the OFDM symbols properly converted to parallel symbol streams. Figure 5.7 shows a screen shot from the I+ channel of a decimated 800 MSps OFDM input symbol stream. The observed symbol rate is

80 MSps with the input symbol rate of 800 MSps which is the expected factor of decimation in time. Each symbol has clean edges and does not show signs of a charge pedestal.

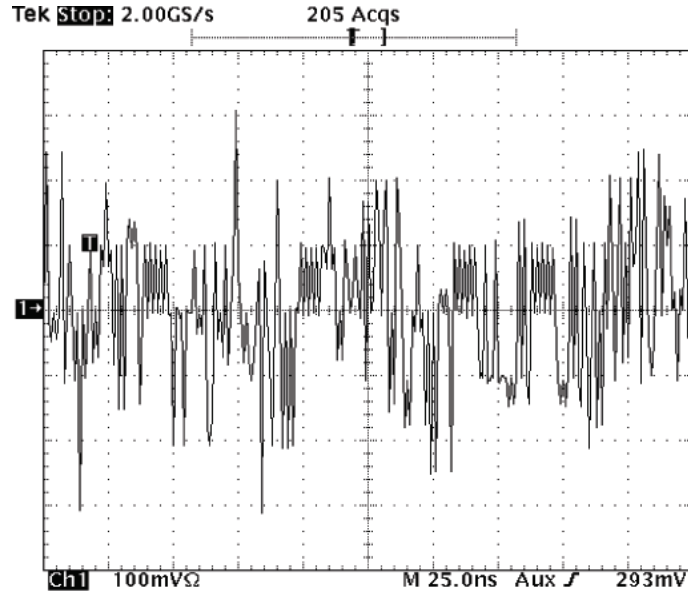
5.4 Characterization of the DT FFT Processor IC

Having verified the correct functionality of the instrumentation amplifiers, and the serial-to-parallel function, the IC variant containing the full DT FFT processor core was tested. To do this, the test setup of Figure 5.3 was again used and an I and Q differential OFDM stimulus signal applied to the processor. Figure 5.8(a) shows the I+ channel of the stimulus signal. The stimulus signal is a 1 GSps BPSK modulated OFDM signal with eight mapped sub-channels containing pseudo-random data. BPSK modulation was chosen over QPSK modulation because it is easier to visually interpret when viewed on the oscilloscope.

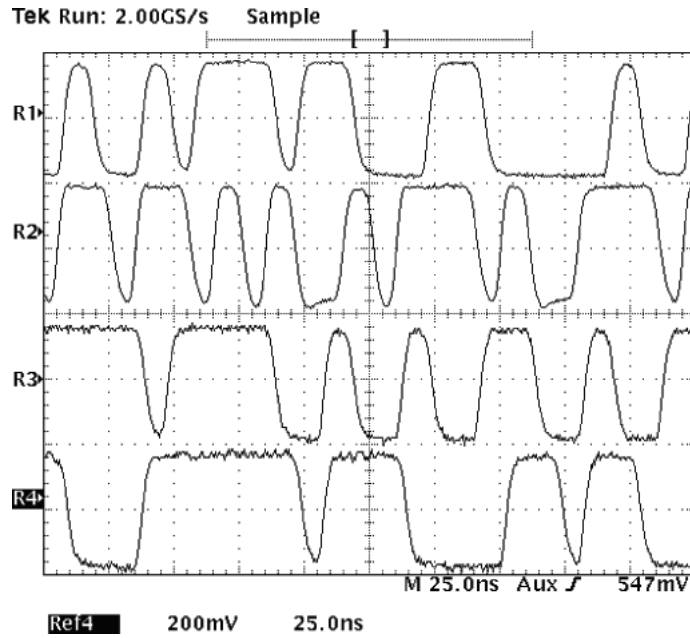
The oscilloscope screen capture of Figure 5.8(b) shows four of the eight sub-channels demodulated to BPSK by the DT FFT processor and measured at the I+ output. The rise time of the output symbols is measured to be 4.9 nS. This figure shows that the Discrete-Time FFT processor is able to successfully demodulate OFDM symbols at a rate of 1 GSps.

Figure 5.9 shows one of the measured BPSK sub-channels after recombining the differential signals in post-processing. The post processing in MATLAB includes differential re-combining, and symbol timing recovery. Figure 5.10 shows the constellation diagram of this measured signal in XY format. A 1.9° phase rotation can be seen in the output, due to the imperfect matching of the output cable lengths.

Next, the distortion contributed by the processor was measured by sweeping the input signal magnitude of the OFDM sub-channels and measuring the SNDR using equation 5.1. In this test, a long OFDM signal is generated in MATLAB and uploaded to the AWG. The processor demodulates the signal and the results are captured by the oscilloscope and ported back to MATLAB. After symbol timing recovery in MATLAB, the error vector between in the input and output OFDM signal is calculated and the RMS magnitude of the error vector is used to form the SNDR. By varying the magnitude of the signal applied by the AWG, the SNDR performance of the processor



(a) input



(b) output

Figure 5.8: (a) A 1GSps OFDM input signal as applied to the input of the OFDM processor.
(b) Four of the eight parallel demodulated outputs.

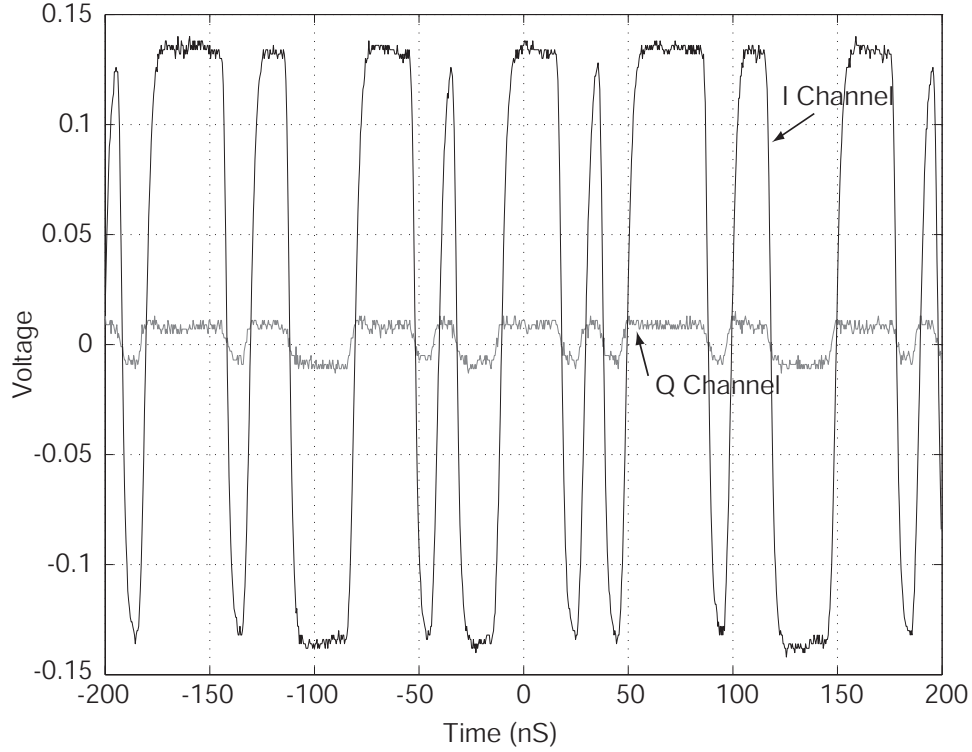


Figure 5.9: Measurement results after being captured on the oscilloscope and recombined in MATLAB for a single demodulated output channel from the FFT processor.

at several power levels can be determined. These measured values are shown in Figure 5.11, along with a data fit curve, (heavy line). The left portion of the curve, typically dominated by thermal noise and $V_{in,os}$ has 5 dB lower SNDR than predicted in simulation. This likely indicates that $V_{in,os}$ was set too small in the simulations. The dynamic range, measured between the 7 dB points on the SNDR curve, is 49 dB and the peak SNDR is 36 dB. The measured dynamic range is also 5 dB lower than the simulated value. This indicates that the system simulations were effective at predicting large signal performance. Comparing the measured results in Figure 3.20 with the quantization limited digital FFT processor of Chapter 3, the measured dynamic range of the DT FFT processor is equivalent to 8.4-bits.

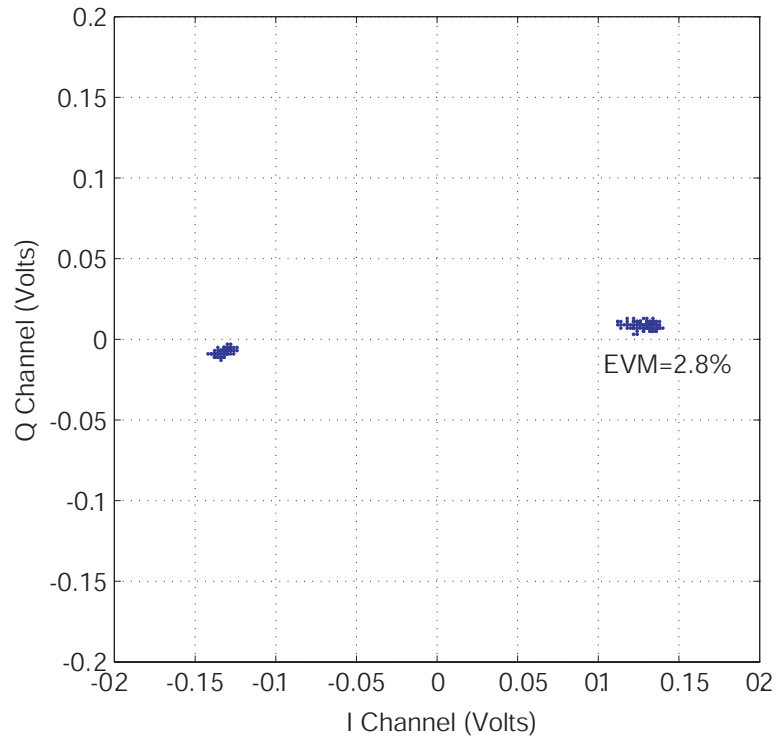


Figure 5.10: Measurement results after symbol timing recover in MATLAB of a single de-modulated output channel displayed in XY format.

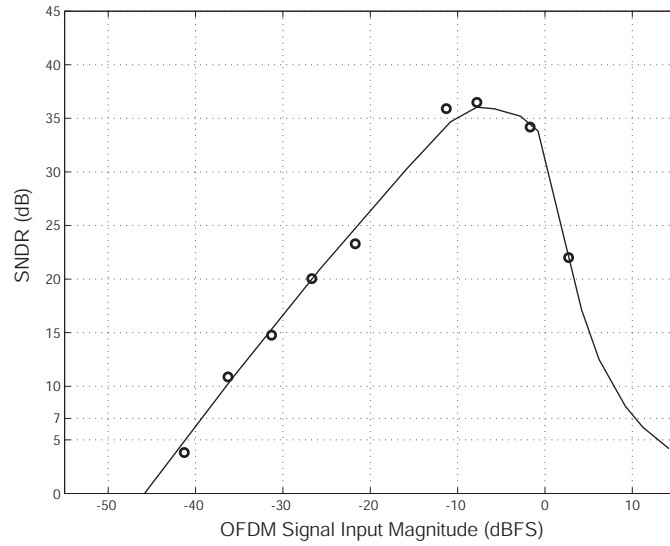


Figure 5.11: Measurement results for the Discrete-Time FFT processor show a peak SNDR of 36dB and a Dynamic Range of 49dB.

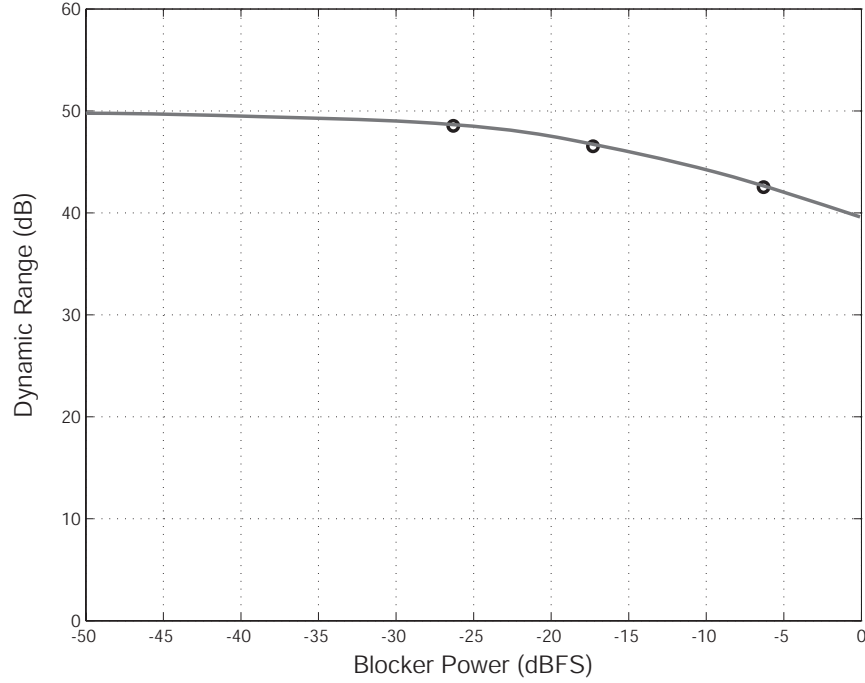


Figure 5.12: Measurement results for the Discrete-Time FFT processor dynamic range after rejecting sinusoidal blocker of varied input magnitude.

Figure 5.12 shows the measurement results when a 345 MHz blocker tone is added to the 1 GSps input signal at magnitudes of -26 dBFS, -17 dBFS and -6 dBFS. The full scale value of $300\text{mV}_{pk,diff}$ is used in calculations. In each blocker measurement, the corrupted sub-channel was attenuated and the remaining sub-channels used to calculate SNDR. The curve fit of the data points shown in Figure 5.12 shows that the DT FFT processor can effectively remove large blockers of any magnitude less than full scale while maintaining good dynamic range.

The processor consumes a total power of 25 mWatts from a 1.2Volt supply. 3.6 mWatts was consumed in the serial-to-parallel converter, 2.4 mWatts in the clock divider and 19 mWatts in the FFT signal flow graph. There are a total of 104 multipliers in the processor, resulting in $180\mu\text{Watts}$ per multiplier. This is a factor of four larger than the design value of $40\mu\text{Watts}$. This discrepancy is due to a mistake in the layout, in which the adder current mirrors were sized four times too small, and the current in the multiplier was forced to be adjusted four times larger for the measurements to compensate. Without this error, the processor would have consumed

Table 5.4: Summary of Measurement Results

Process	CMOS 0.13 μ m
Supply Voltage	1.2V
Size	0.203mm ²
Total Power Consumption	25mW @ 1GSps
SHA Power Consumption	56 μ W @ 1GSps
Serial-to-Parallel Converter Power Consumption	3.6mW @ 1GSps
Analog Multiplier Power Consumption	1.8mW @ 1GSps
Clock Divider Power Consumption	2.4mW @ 1GSps
Input Range	400mV _{<i>pk-pk,diff</i>}
Output Range	400mV _{<i>pk-pk,diff</i>}
Interleaver Ratio	1:8
EVM	2.8% @ 1GSps
Peak SNDR	36dB
Dynamic Range	49dB
ENOB	8.4bits

only 11 mWatts. Regardless, the measured power of 25 mWatts is a significant improvement over the best reported power consumption for an all digital 1 GSps FFT processor (175 mWatts) [48].

Table 5.4 presents a summary of the measurement results. The discrete-time FFT processor achieves the design goals of high operating speed, power efficiency and high linearity. The full scale signal range is 400mV_{*pk-pk,diff*} both in and out of the processor. The area occupied by the processor is 0.2mm². The error vector magnitude for BPSK was 2.8%. With dynamic range of 49 dB and an ENOB of 8.4, the processor is a good candidate for future leading data-rate UWB OFDM systems requiring 1 GSps or more.

5.5 Summary

This chapter has presented an experimental demonstration of a discrete-time FFT processor as a proof-of-concept for an improved architectural approach to OFDM receivers. The architecture performs the FFT required for OFDM demodulation in the

discrete-time domain, reducing the overall receiver power consumption while increasing linearity and blocker handling capability. The measured results demonstrate that the discrete-time FFT processor has a dynamic range of 49 dB, versus 36 dB with an all digital approach. This improvement in dynamic range increases receiver performance by allowing detection of weak sub-channels attenuated by multi-path. The measurements also demonstrate that the processor rejects large narrow-band blockers, while maintaining greater than 40 dB of dynamic range. The processor enables a 10x reduction in receiver power consumption as it reduces the required ADC bit depth by four bits and consumes only 25 mWatts, enabling application in hand-held devices.

In the next chapter, the design and layout of a second generation of the DT FFT processor is presented, incorporating improvements based upon the results found from the proof-of-concept IC presented in this chapter.

Chapter 6

An Improved DT FFT Processor Design

In the previous chapter, measurement results from the first proof-of-concept DT FFT processor IC were presented. Building on the potential of that IC, a second proof-of-concept IC was designed and laid out with additional functionality. Furthermore, the new design includes improvements to some of the circuits from the first design.

The two new functions added to the DT FFT implementation are the Equalizer and Parallel-to-Serial function. Integrated in the IC, they allow for a more complete implementation of the DT FFT processor. These functions are discussed in the first two sections of this chapter.

In addition to the new functions, the clock generating function and the output driver have been improved for this IC and are discussed in the third section and fourth sections. Finally the layout of the new IC is presented at the end of the chapter.

6.1 Equalizer

Inclusion of the Equalizer in the improved IC design allows the DT FFT processor to apply gain and phase corrections to weak sub-channels attenuated by multi-path, and to apply attenuation to sub-channels corrupted by strong blocker signals. This allows for a more complete implementation of the DT FFT processor. Equalization

can either be applied at the parallel outputs of the FFT SFG, or serially after the FFT SFG output has passed through a parallel-to-serial converter. The former approach leads to a single high speed equalizer and the latter approach leads to multiple parallel equalizers with relaxed constraints. In this work, multiple parallel equalizers were chosen due to the relaxed constraints.

As described in Section 2.2.4 and shown again in Figure 6.1, the equalizer can be implemented with a pair of variable gain linear transconductors and an adder circuit to implement the equation:

$$\begin{aligned} Y_I &= R_{add} (G_{m_c} X_I - G_{m_s} X_Q) \\ Y_Q &= R_{add} (G_{m_s} X_I + G_{m_c} X_Q) \end{aligned} \quad (6.1)$$

where R_{add} is the differential impedance of the adder circuit and G_{m_c} , G_{m_s} are defined to be:

$$\begin{aligned} G_{m_c} &= G_{m_k} \cos(\theta_k) \\ G_{m_s} &= G_{m_k} \sin(\theta_k) \end{aligned} \quad (6.2)$$

where θ is the correction phase and G_{m_k} is the correction magnitude of a given sub-channel. In a wireless transceiver implementation, these values would come from the digital signal processing portion of the receiver after the data conversion.

The equalization circuit is similar to the circuit of the real coefficient butterfly structure from Figure 2.5. In this design some of the circuits are re-used, which serves to reduce the complexity and lower the risk of implementation error in the improved DT FFT IC. The linear transconductor is implemented in the equalizer exactly as that in the FFT SFG.

On the other hand, the adder circuit is different from that implemented in the FFT SFG and is implemented as shown in Figure 6.2. This circuit is similar to that in Figure 4.6 but with M1,M2 sized to allow for a larger gain. The impedance R_{add} is defined as the differential impedance between *Out+* and *Out-* and is primarily set by

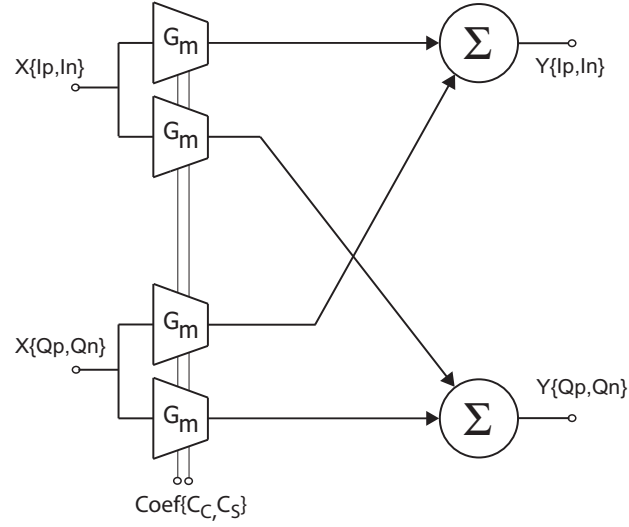


Figure 6.1: The equalizer circuit scales real and imaginary inputs to correct for sub-channel magnitude and phase error.

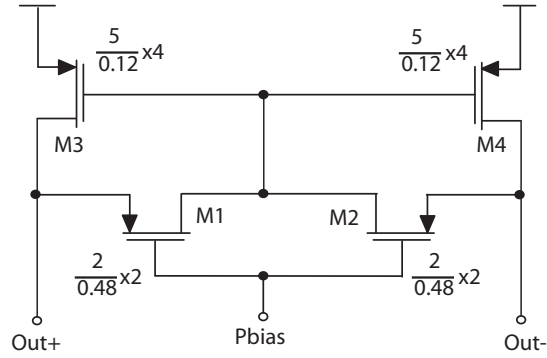


Figure 6.2: The adder circuit used in the equalizer cell is similar to Figure 4.6 but with M1,M2 sized for higher resistance and higher gain.

M1,M2. M1,M2 are sized at $W = 2 \mu m$ and $L = 2 \mu m$ to have a gain that varies from -10dB to +20dB for the equalizer. To attenuate blockers, the linear transconductor is switched off, ideally leading to infinite attenuation of the given sub-channel.

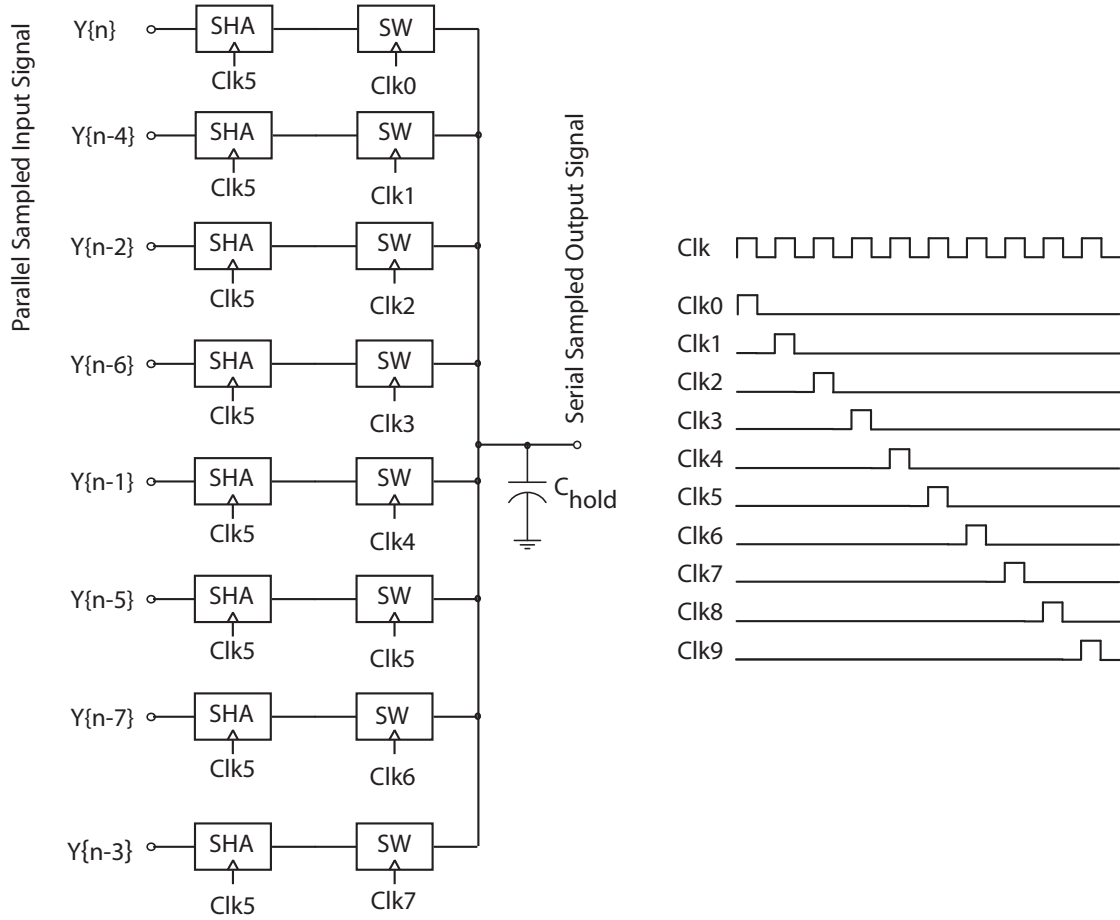


Figure 6.3: The Parallel-to-Serial conversion function consists of a bank of impedance buffer SHAs, a bank of switches and a single summing capacitors. For simplicity, the differential I and Q lines are represented by a single line in the signal flow diagram.

6.2 Parallel-to-Serial Conversion Function

The purpose of the parallel-to-serial conversion function is to recombine the parallel data samples from the output of the equalizer and FFT SFG into a serial stream of data samples. Figure 6.3 shows the principal components of this circuit: a bank of buffer SHAs, a bank of switches and a single set of summing capacitors. As with signal flow diagrams previously presented, the circuits are pseudo-differential I and Q, resulting in four identical circuits for each single ended circuit shown.

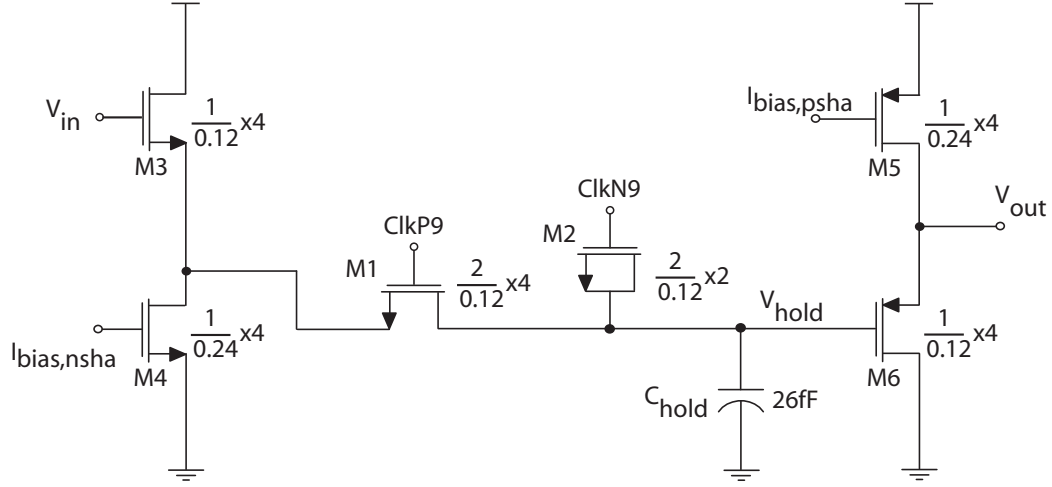


Figure 6.4: The low input capacitance SHA used in the parallel-to-serial converter.

6.2.1 Buffer SHA

Figure 6.4 shows a single buffer SHA. This circuit lowers the input capacitance of the SHA with an NFET source follower that provides a negative DC level shift and drives a NFET switch. The SHA uses the NFET switch M1 to sample the input signal onto C_{hold} . M2 is sized to be half the area of M1 for charge pedestal canceling. The PFET source follower, M5–M6 amplifies the charge on C_{hold} and drives the circuits following the buffer SHA, which are the switch and capacitive summing circuits.

Simulations of the circuit in Figure 6.4 are summarized in Table 6.1 and show that the circuit drives the load of the ensuing summing stage with a bandwidth of 0.89 GHz while drawing 110 μ W. The input impedance is only 1.4 fF and will have a minimal loading effect on the preceding equalizer circuit.

6.2.2 Combining Sample-and-Hold circuit

The combining sample-and-hold circuit, shown in Figure 6.5 consists of eight parallel sampling switches that drive a single capacitor. The clock signals supplied from the clock generation circuit ensure that only one switch at a time is ever closed. As the clock signals sequence from $ClkP0$ to $ClkP7$ each of the parallel inputs is sampled serially onto the capacitor C_{hold} . Half of the value for C_{hold} , 26 fF, is implemented

Table 6.1: Simulation Results for the buffer SHA design

Bias Current	24 μ A
Bandwidth	0.89 GHz
Output Drive Impedance*	3200 Ohms
Positive Clock Load	10.86 fF
Negative Clock Load	5.42 fF
Input Impedance*	1.4 fF
Total Power Consumption	110 μ W
Required Clock Power	20.1 μ W

*single ended

Table 6.2: Simulation Results of clock load capacitance for the Combining Sample-and-Hold circuit

Positive Clock Load per clock phase	6.3 fF
Negative Clock Load per clock phase	3.9 fF

with a physical capacitor; the other half is contributed by the layout capacitance of the eight parallel paths feeding C_{hold} . The dummy transistors connected to the negative clocks are physically located next to the switches and are used to cancel the charge pedestal from the switch closing. The simulation results of the capacitive load from the switches presented to the clocking circuitry is shown in Table 6.2. The output of this circuit is fed into the low input capacitance instrumentation amplifier.

The total capacitive load presented to each clock can be calculated using the capacitive load values presented in Chapter 4, Tables 4.1 and 4.2, in addition to that from Tables 6.1 and 6.2. These values are given is given in Table 6.3. Based on these values the required drive strength for each of the clock drivers can be determined. In the next section, the clock generation circuitry for the second prototype IC is described.

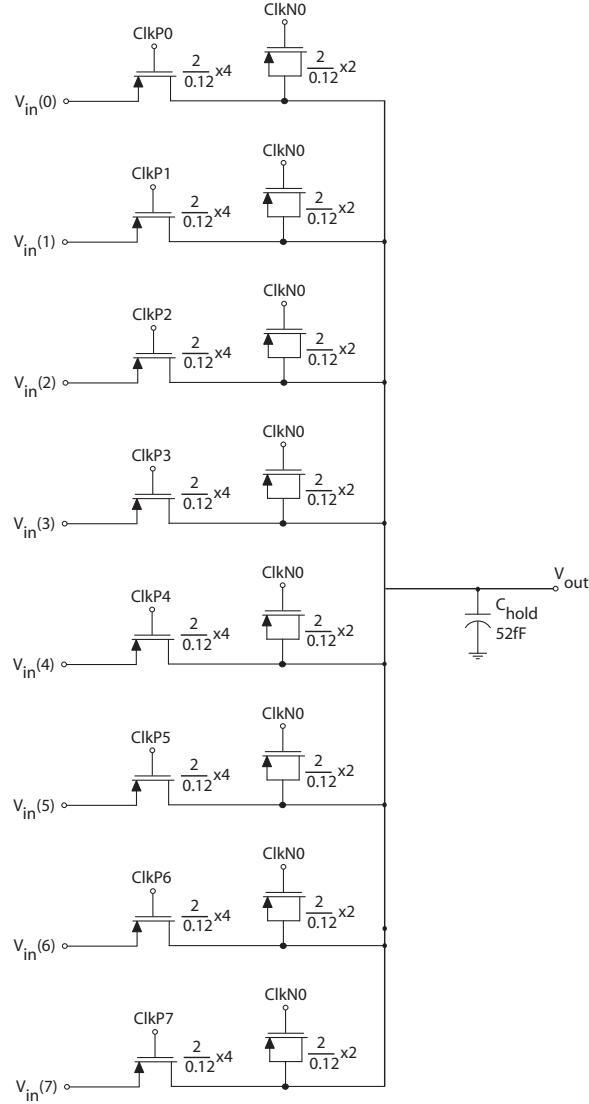


Figure 6.5: The combining sample-and-hold circuit operates by sequentially turning on one switch at a time to sample the parallel input data onto C_{hold} .

6.3 Clocking Circuitry

Based on the measurements from the first DT FFT processor prototype described in Chapters 4-5, it was clear that the functional block limiting operation to below 1 GSps was the clocking circuitry. First the processor functioned with excellent dynamic range at sample rates from 100 MSps to 1 GSps, but at 1.05 GSps the SNDR dropped quickly to zero. Secondly, above 1 GSps, the digital voltage supply

Table 6.3: The capacitive load presented to the each clock output from the clock generation circuit

Clock Name	Load Capacitance
ClkP{0-3}	12.6fF
ClkN{0-3}	8.2fF
ClkP4	6.3fF
ClkN4	4.1fF
ClkP5	93.2fF
ClkN5	47.5fF
ClkP{6-8}	12.6fF
ClkN{6-8}	8.2fF
ClkP9	99.5fF
ClkN9	51.6fF

pin, *DVDD*, became sensitive to supply level. At 1.00 GSps a drop of 50 mV from 1.2 Volts to 1.15 Volts caused the processor to cease functions whereas at 100 MSps this same voltage drop had no effect. Considering different possible sources of these failures, the multipliers and adders within the FFT SFG could be eliminated since they are expected to gradually degrade in performance as operating rates increase beyond their 3 dB frequency. The serial-to-parallel function was eliminated because the voltage supply pin to the serial-to-parallel function, *VDDSHA* did not any show sensitivity to voltage level. Thus the logical conclusion is that the high speed digital logic, i.e. the clock generation circuitry was the performance portion of the design above 1 GSps.

After further analysis and simulation of the clock generation circuit given in Chapter 4 and repeated here as Figure 6.6, it was noted that a timing asymmetry exists in the original scheme. The AND gates X1–X4 and X6–X9 are driven by one D-flip-flop *Q* output and one *QN* output; but the AND gate X0 is driven by two D-flip-flop *QN* outputs and the AND gate X5 is driven by two *Q* outputs. Although the “PowerPC” D-flip-flop in Figure 4.20 is a widely used circuit topology, the outputs *Q* and *QN* are not symmetrical. Instead, as mentioned in Chapter 4, there is an inverter delay time between the *Q* output and the *QN* output. When operated above 1 GSps the inverter delay becomes significant enough to cause the inputs QN5 and QN1 to never overlap and thus the AND gate X0 does not fire.

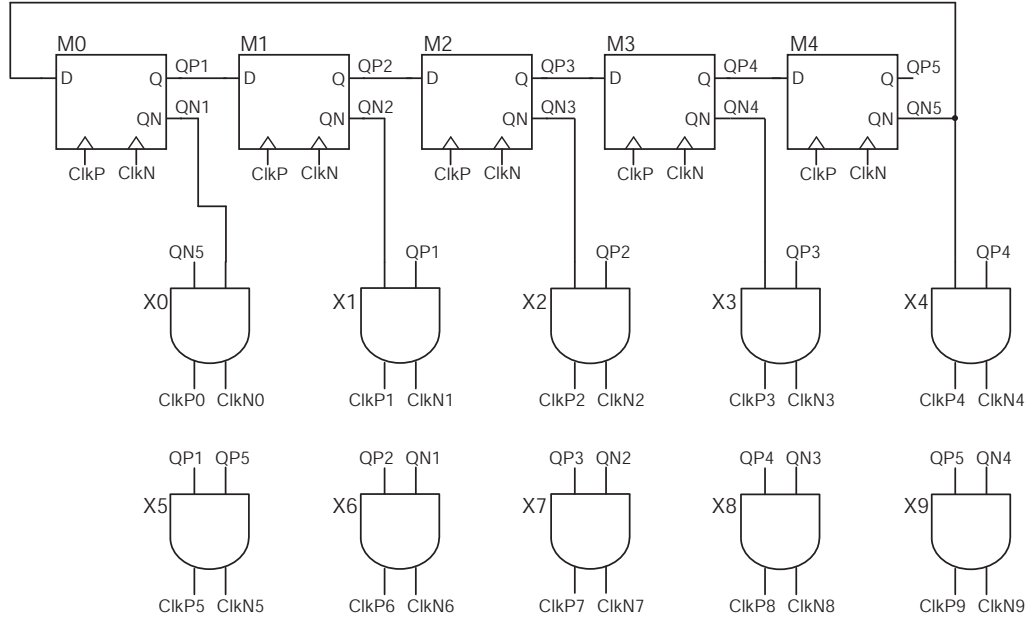


Figure 6.6: The clock generation circuit used in the first prototype of the DT FFT processor.

For the second prototype IC, the architecture shown in Figure 6.7 is used. The new clock generation circuit attempts to be more symmetrical than the previous one by using ten D-flip-flops instead of five so that each clock is derived from a single D-flip-flop and never combines the output of two. Differential AND gates are used to synchronize the D-flip-flop outputs with the main clock, $ClkP$ and $ClkN$.

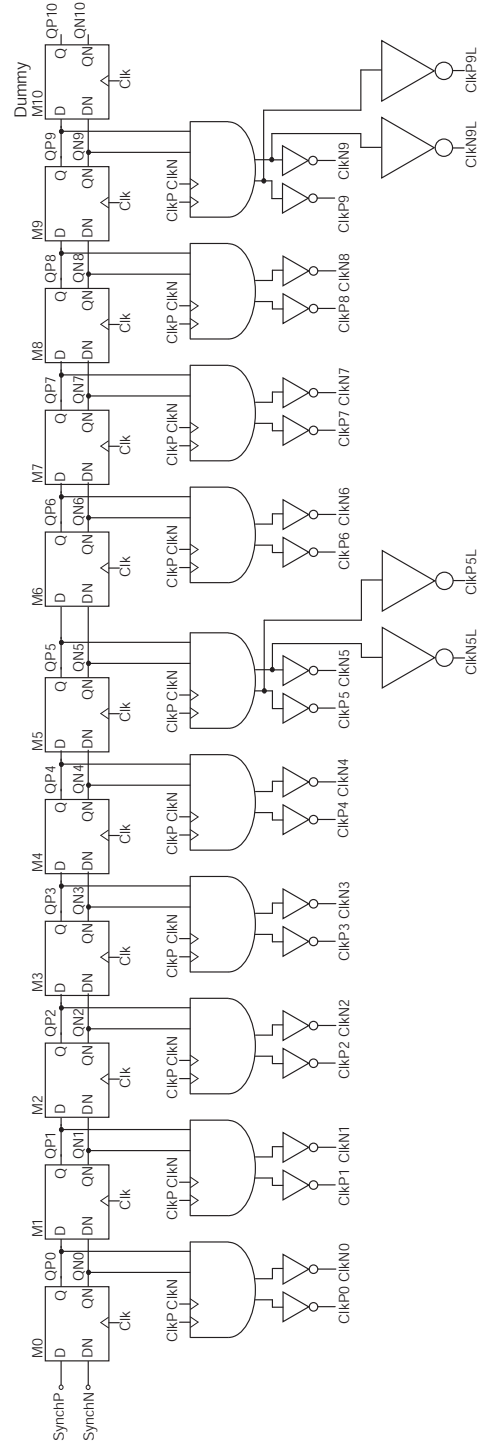


Figure 6.7: The clock generation circuit used in the second prototype IC creates 10 clock phases and utilizes inverter drivers individually scaled to drive the circuit functions within the DT FFT Processor.

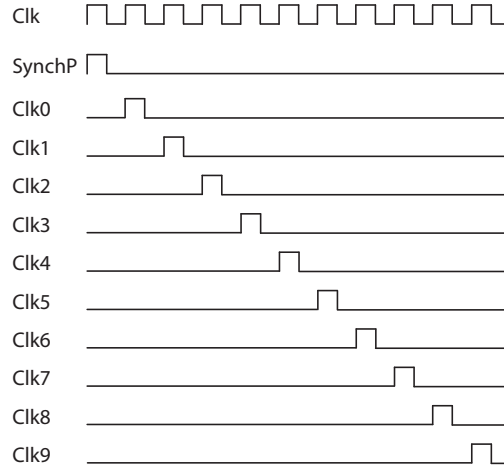


Figure 6.8: The clock generating diagram for the second prototype IC including the synchronization input.

The new clock generation scheme requires an additional input for the circuit and the periphery of the IC. The additional input port is for a differential logic synchronization signal, called *SynchP* and *SynchN*. This signal is used to indicate the beginning of an OFDM packet, and causes the clocking sequence to begin with *Clk0*. Figure 6.8 shows the clocking diagram. Following *Clk0*, a pulse propagates down the D-flip-flop chain, making *Clk1*, *Clk2*, etc. to *Clk10*. The pulse ends at the dummy D-flip-flop instead of recirculating to the beginning as done in the previous clock generation circuit. The addition of the *SynchP* and *SynchN* signal also allows more flexibility in testing since the time between OFDM symbols is controlled off-chip.

6.3.1 Differential Sense Amplifier D-flip-flop

To further improve the symmetry of the clock generation circuitry, an alternate D-flipflop circuit topology is used. Instead of the “PowerPC” topology, described in Chapter 4 (based on transmission gates and a single ended Master-Slave latch pair) the differential “Sense Amplifier” D-flip-flop topology is used for the symmetrical timing of its differential output signals. The operating speed of both topologies is similar [90,91].

The sense amplifier D-flip-flop topology relies on a pulse generator latch followed by a slave latch as shown in Figure 6.9. The pulse generator is sensitive to the input on

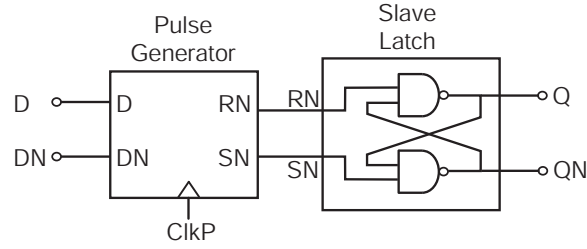


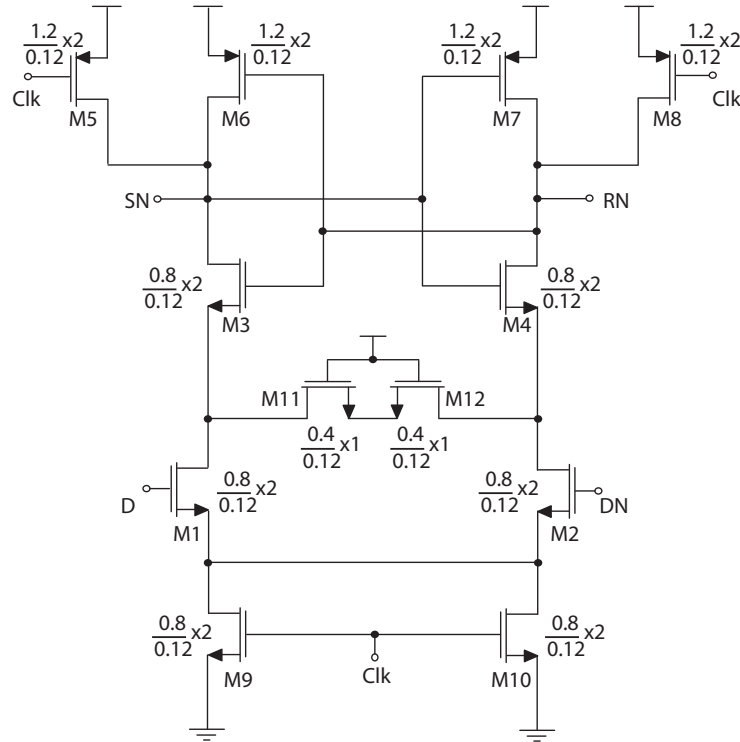
Figure 6.9: The sense amplifier D-flip-flop is constructed from two circuits, a pulse generator and a slave latch.

the rising edge of the clock. Although the sense amplifier topology is fully differential for logic input and output, it uses a single ended clock. The circuit diagram of the sense amplifier D-flip-flop is shown in Figure 6.10.

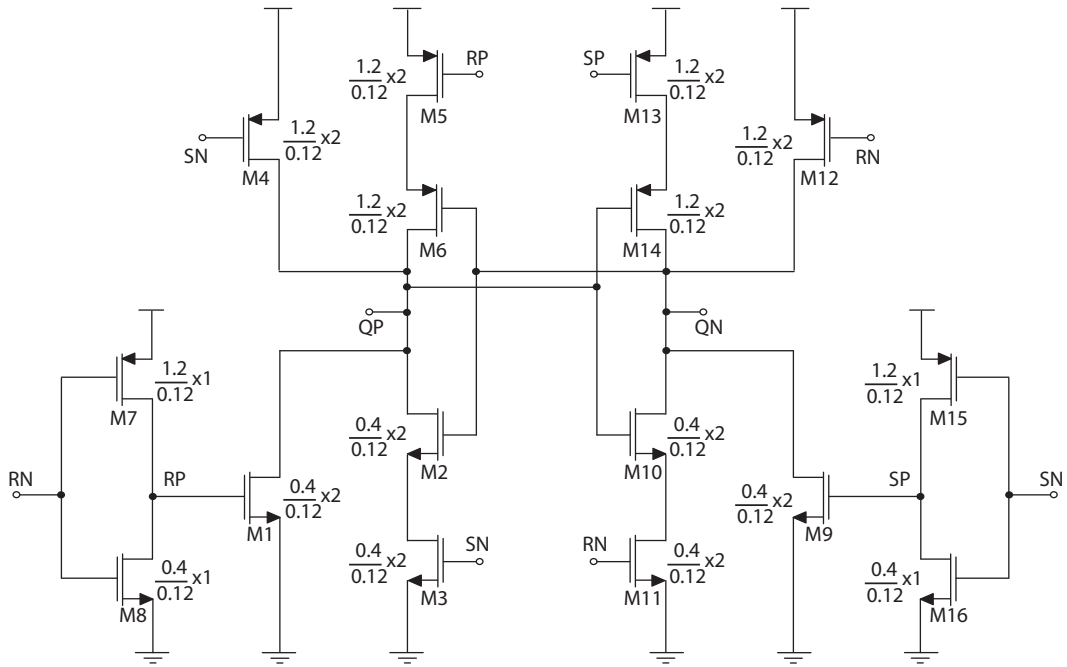
As shown in Figure 6.10(a), the sense amplifier pulse generator is fundamentally a differential pair formed by M1 and M2, with current source M9, M10, and a positive feedback load formed by the cross-coupled inverters M3, M6 and M4, M7 [99]. When the clock is low, the sense amplifier pre-charges all of the internal voltage nodes. Switches M5, M8 are closed, pulling the outputs high and pre-charging the inputs of the slave latch to V_{dd} . M3 and M4 are also closed on low clock and pre-charge the drains of the differential pair, M1, M2 to $V_{dd} - V_{th}$. The differential pair is off because the current source M9, M10 is off.

When the clock transitions high, the differential pair is turned on through the current source M9, M10. Simultaneously, M5 and M8 turn off, initially isolating the differential pair from V_{dd} and connecting the output of the differential pair to SN and RN . The differential pair senses the differential voltage between D and DN and pulls charge off the node SN or RN through the path M3,M1,M9 or M4,M2,M10 respectively. After the initial pulse is generated by the sense amplifier, the positive feedback in the cross coupled inverter pair, M3,M6 and M4,M7 hold the output value fixed.

The slave set-reset latch functions in a similar manner to a pair cross-coupled NAND gates as shown in Figure 6.10(b). The inputs SN and QN are NANDed in M2 and M3 to drive the QP output. The inputs RN and QP are NANDed in M10, M11 to drive the QN output. The additional NFETs M1,M9 are included to add symmetry to the drive strength of the outputs [99]. The complementary PFET logic is implemented



(a) D-flip-flop Sense Amplifier Pulse Generating Circuit



(b) D-flip-flop Set-Reset Slave Latch

Figure 6.10: The circuit diagram of the sense amplifier D-flip-flop. The sense amplifier pulse generating circuit (a) and the set-reset slave latch (b)

Table 6.4: The timing results of the Sense Amplifier D-flip-flop simulation.

Transmission Times	Output QP	Output QN
t_{pLH}	335 ps	354 ps
t_{pHL}	108 ps	81 ps
t_r	73ps	67ps
t_f	38ps	35ps

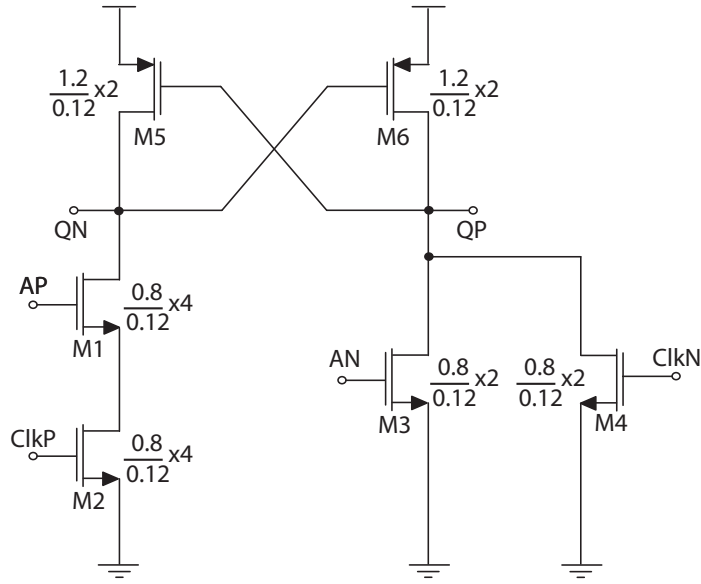


Figure 6.11: The differential AND gate used in the clock generation circuitry.

in M4–M6, M12–M14.

The sense-amplifier D-flip-flop is simulated and the timing results shown in Table 6.4. The symmetry between the differential outputs QP and QN is better than 25 ps.

6.3.2 Differential AND, Inverters

In addition to the sense amplifier D-flip-flop, a differential AND gate is also required for the new clock generation circuit. Figure 6.11 shows the circuit. The logic uses complementary NFET functions for the AP , $ClkP$ and AN , $ClkN$ inputs and uses cross-coupled PFET loads to improve operating speed.

Following the AND gates, inverters are used as driver circuits. The individual drivers are scaled in size according to the expected load capacitance from Table 6.3 presented to each of the clock signals $Clk0$ - $Clk9$.

Using the inverters, differential AND gates and sense amplifier D-flip-flops described in this section, the clock generation circuit given in Figure 6.7 is implemented in the improved DT FFT processor IC design.

6.4 IC Peripheral Circuit Designs

In the test setup of the first proof-of-concept IC, the initially planned method to interface differential output signals from the IC to single ended $50\ \Omega$ test equipment was to use external off-chip baluns. The supply voltage for the driver amplifier was intended to be passed through the center-tap of the balun. This approach allowed the sharing of pads between the supply voltage and output signals. After initial testing of the circuit board and baluns with a network analyzer, it was clear that the baluns were being de-tuned by interaction between the bias-tee and the balun, causing ≈ 3 dB of ripple in the S21 response. Broadband baluns operating to 1 GHz bandwidth are known to be sensitive to parasitic grounding inductance [100]. To alleviate this problem, shunt resistive bias tees were used, lowering the output impedance seen by the driver to $25\ \Omega$ and reducing the intended operating bandwidth.

To avoid this sensitivity, in the improved proof-of-concept IC, the driver amplifier was modified as shown in Figure 6.12 to include an additional output pad, V_{ocm} for the driver supply voltage. The addition of V_{ocm} allows the supply voltage of 3.3 V to be applied to through the resistors R3, R4 which were previously used as differential-mode resistors. The resistors R3, R4 also serve to set the output impedance of the driver amplifier at $50\ \Omega$.

6.5 IC Layout

Figure 6.13 shows the layout of the improved DT FFT IC, designed in the same CMOS process as the first proof-of-concept IC, Jazz $0.13\ \mu\text{m}$. This allowed re-use of

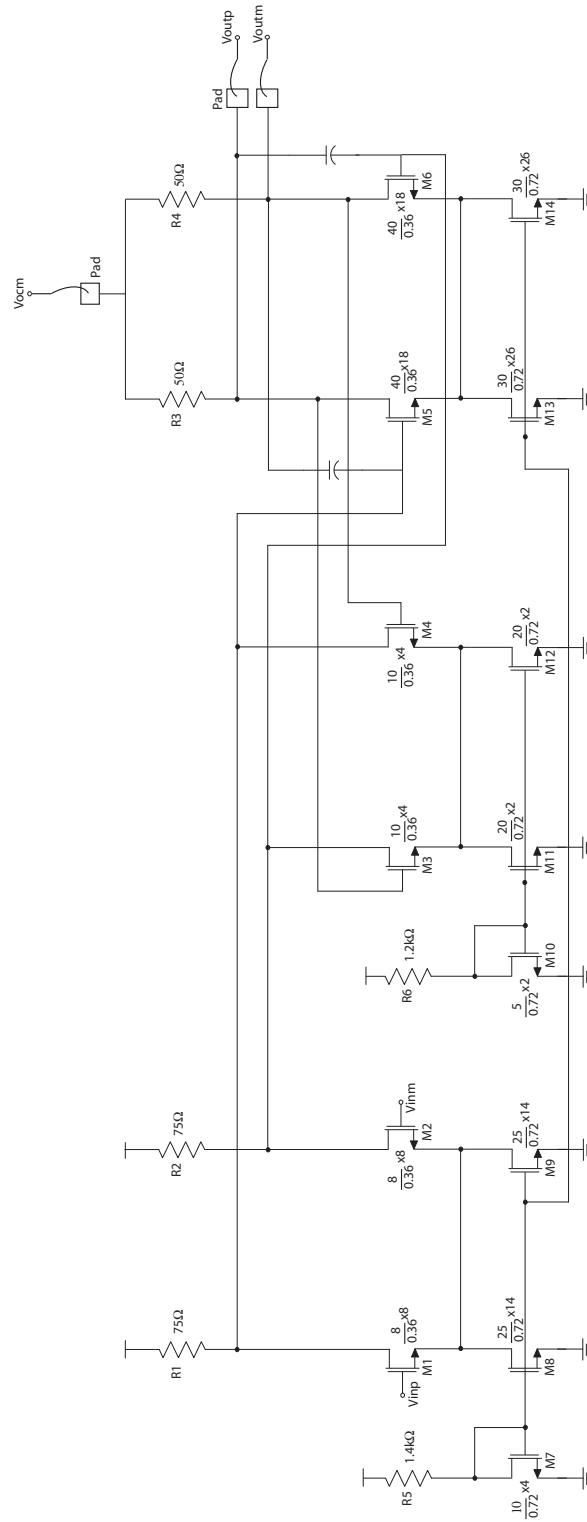


Figure 6.12: The 50 Ω output impedance driver amplifier.

some of the interconnect networks between bias, control and power; however since new pads were added for clock synchronization, driver amplifier V_{dd} and equalizer control, changes were unavoidable. As can be seen in the figure, the layout is still pad constrained. In addition to the new pads, the instrumentation amplifiers, were considerably reduced in size, only needing to buffer the serial output of the FFT core instead of a parallel output. The dimensions of this layout are $1500\text{ }\mu\text{m}$ by $1400\text{ }\mu\text{m}$, or an area of (2.1 mm^2) .

The DT FFT core itself, shown in Figure 6.14 is $540\text{ }\mu\text{m}$ by $520\text{ }\mu\text{m}$, (0.28 mm^2) . In comparison, the area of the first proof-of-concept DT FFT core was 0.20 mm^2 .

The signal flow in the DT FFT core starts in the lower left corner as four serial inputs. These are converted to parallel data in the serial-to-parallel converter and passed into the FFT SFG in the middle of the figure. The new clock generation circuitry is located to the left of the serial-to-parallel converter in the figure, and separated by an isolation guard ring and separate power supply domain. The equalizer can be seen to the right of the FFT SFG, followed by the parallel-to-serial converter. The four serial outputs are on the right.

The layout of the new clock generation circuitry can be seen in Figure 6.15. The figure is rotated clockwise 90° from the figure of the processor core. The 10 sense amplifier D-flip-flops can be seen along the top edge of the figure. The differential AND logic and driver inverters are at the lower edge of the circuit. The ten phased clock signals are routed on a bus that is shielded from the substrate to avoid clock noise coupling. Metal layer 2 is used for the shield and metal layers 3 and 4 are used for the clock routing.

The layout of the sense amplifier D-flip-flop is shown in Figure 6.16. The upper half contains the set-reset slave latch and the lower half contains the sense amplifier. Symmetry about the vertical axis is attempted in the sense amplifier layout, although there is some compromise to accommodate signal routing.

The layout of a single channel of the equalizer is shown in Figure 6.17. The signal flow is from left to right with the linear transconductors on the left followed by the adder circuit on the right.

The layout of the buffer SHA is shown in Figure 6.18. The low input capacitance buffer is on the left, so as to be placed close to the equalizer output. Some physical

separation between the input buffer and remaining portion of the SHA allows room for routing clocking signal. Symmetry about the horizontal axis is intended to improve matching between the differential I and Q paths.

6.6 Summary

This chapter presented the design and layout of an improved DT FFT processor IC. The new circuit functions of the equalizer and parallel-to-serial function were presented. Design enhancements to the clock generation circuit were also reviewed and the use of an improved sense amplifier D-flip-flop was described. Finally the layout of the complete IC was presented.

In the next chapter, conclusions from the entire work on the DT FFT processor will be presented and future work described.

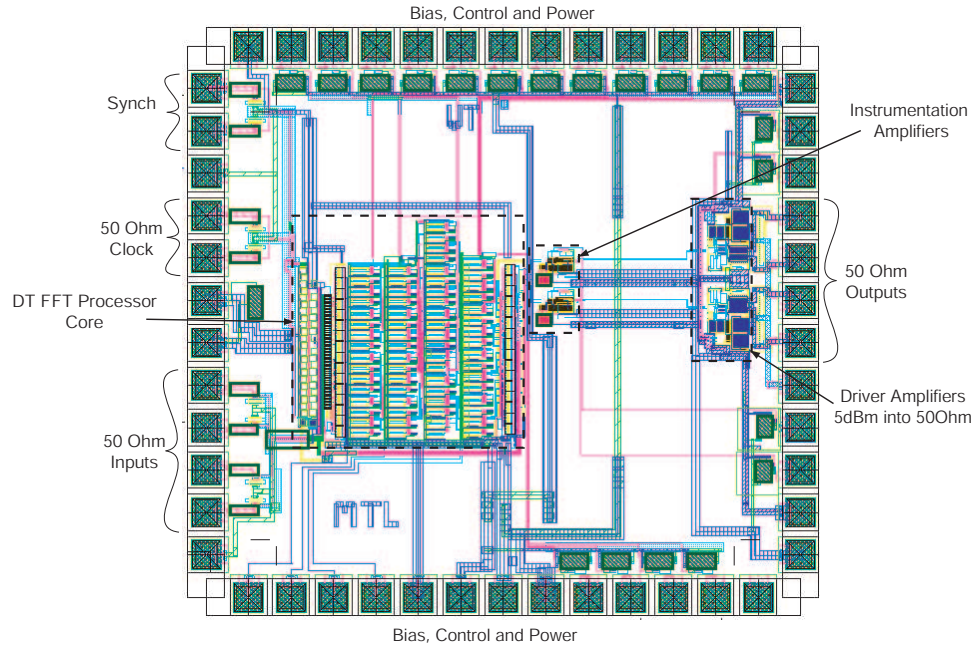


Figure 6.13: The layout of the improved DT FFT processor with the DT FFT processor core, instrumentation interface circuits and driver amplifiers.

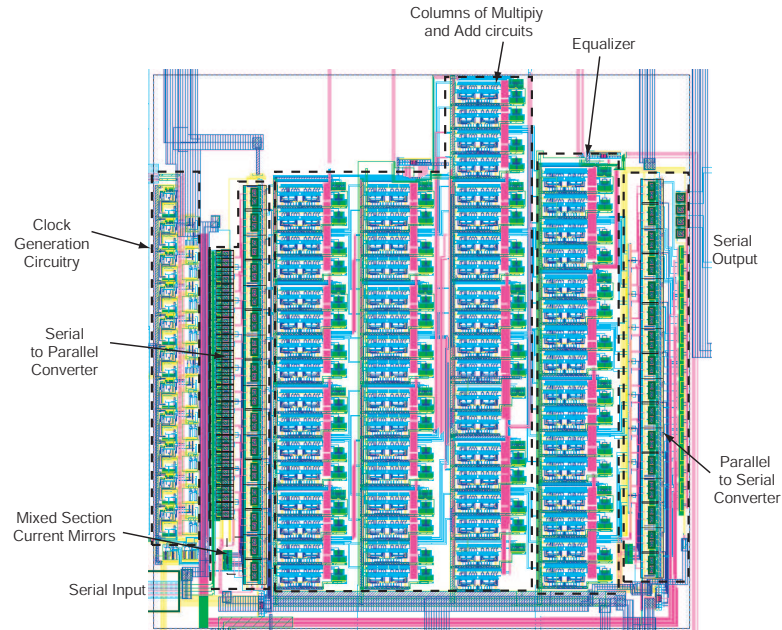


Figure 6.14: The layout of the improved DT FFT processor core consisting of clock generation circuit, serial-to-parallel convert, three columns of multiply and add circuits, equalizer and parallel-to-serial converter.

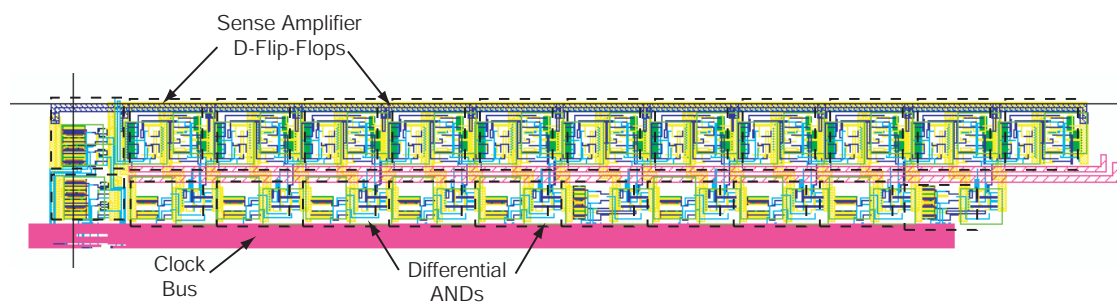


Figure 6.15: The layout of the clock generation circuit.

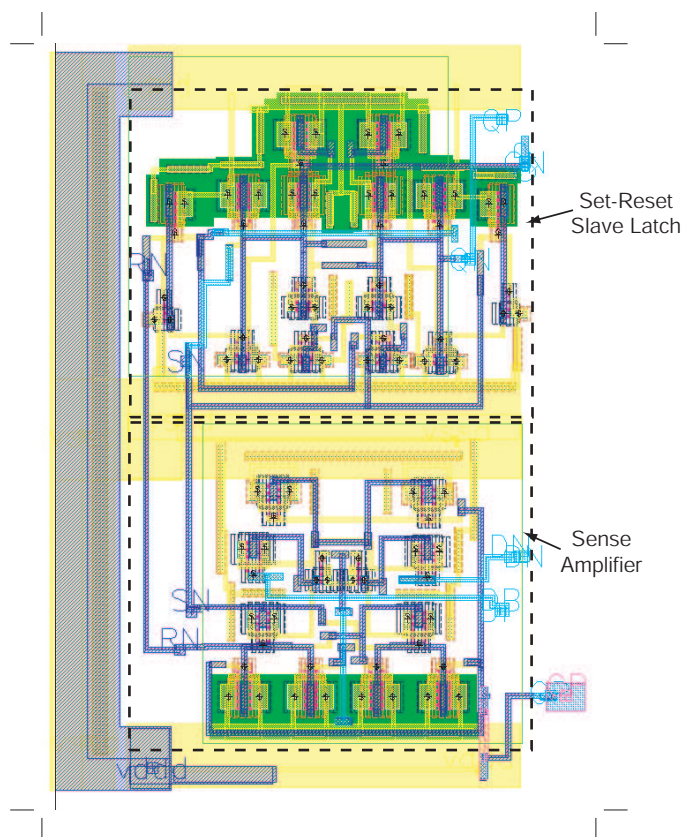


Figure 6.16: The layout of the sense amplifier D-flip-flop.

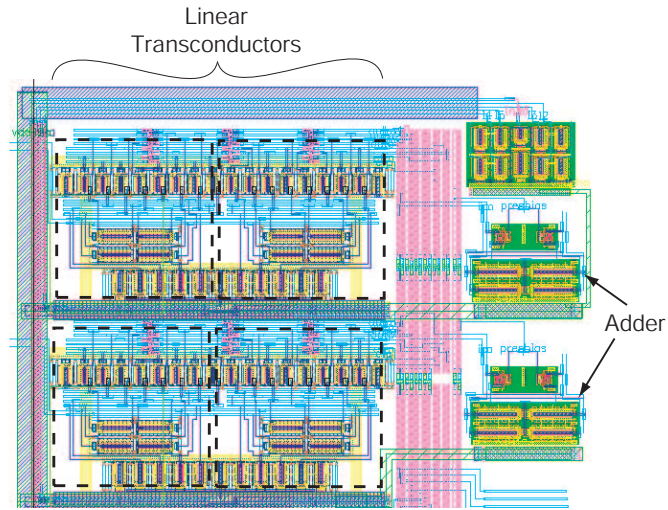


Figure 6.17: The layout of a single channel of the equalizer.

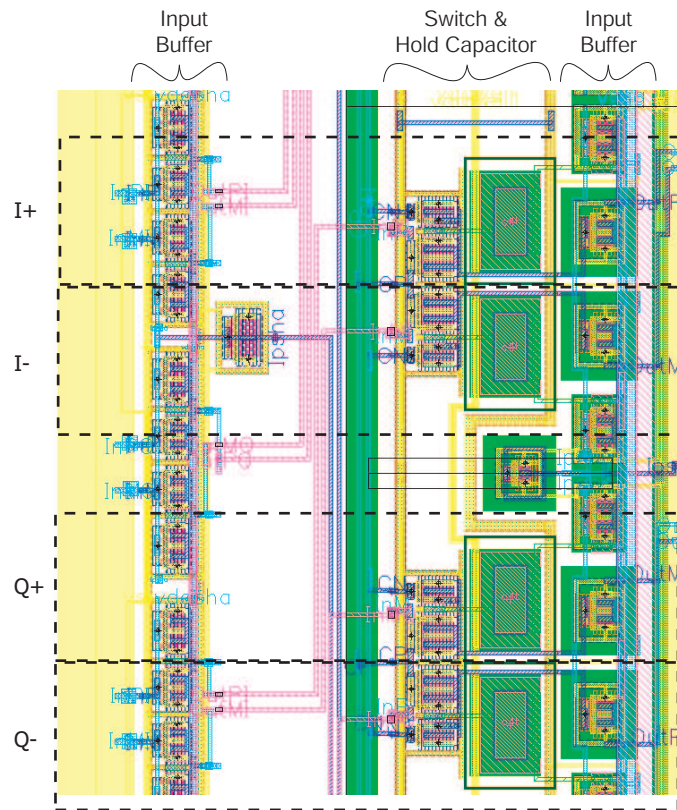


Figure 6.18: The layout of the buffer SHA.

Chapter 7

Conclusions and Future Work

This dissertation has presented a new Discrete Time FFT Processor architecture, system simulations, circuit design of a proof-of-concept IC and measurement results, and the design of an improved second generation DT FFT Processor IC. In this chapter, conclusions are drawn and future work is presented.

7.1 Conclusions

This work was motivated by the goal of reducing the disproportionately high power consumption typically found in UWB OFDM receivers due to the need for high-speed high-bit-depth data conversion. However, it was observed that the total volume of information (complex modulated data) does not need to be passed through the ADC, since the information is quickly reduced through demodulation in the FFT processor immediately following the ADC in conventional architectures. Therefore, a new baseband architecture for UWB OFDM receivers was proposed that moves the FFT processor from the digital signal processing domain to the discrete-time analog signal processing domain. Doing so offers a reduction in the required ADC bit resolution because the high dynamic range OFDM modulation is reduced to digital QAM or Mary-PSK modulation, with lower linearity requirements after the FFT processor. For UWB capable ADCs, reducing the bit depth requirement, reduces power consumption by a factor of 2 for each bit of reduction in resolution. The proposed architecture represents the first discrete time FFT processor for OFDM

demonstrated to date in the literature.

The proposed DT FFT architecture is primarily based on a discrete-time serial-to-parallel converter, an analog-multiplication-based FFT signal flow graph, and a discrete-time analog equalizer. In a discrete time signal processing approach, the continuous-in-time, continuous-in-magnitude signal at the input of the baseband portion of the typical receiver is sampled to realize a discrete-in-time, continuous-in-magnitude signal. Discrete time signal processing results in individual samples of the signal that can be stored in memory, to be compared or processed against later time samples. This signaling domain enables straightforward serial-to-parallel conversion in which N_{FFT} (the number of points in the FFT) time samples are stored and then passed through the FFT signal flow graph in parallel. The continuous-in-magnitude aspect of discrete time signal processing allows coefficient multiplications to be performed using analog variable gain amplifiers, implemented in this work using linear transconductors. By performing the multiplication intensive portion of the signal processing (the FFT) with much more power efficient linear transconductors, significant power savings are achievable since a large number of power intensive digital multipliers are not required.

In the first part of this work, the proposed architecture was validated through extensive system level simulations. Behavioral models of the primary signal processing blocks were introduced to enable these simulations, including a new behavioral model for linear transconductors which accurately incorporates various circuit non-idealities. The simulation results show that the proposed DT FFT processor has a dynamic range equivalent to a 9-bit all digital FFT processor at a 7x reduced power consumption. Blocker simulations show that the processor can reject blockers up to the full scale magnitude of the processor, and retain dynamic range. The non-idealities of the linear transconductor were individually varied in Monte Carlo simulations to determine their impact upon the overall dynamic range of the DT FFT processor. These results indicated that the DT FFT can tolerate up to 10% amplitude ripple and 10% offset in the transconductance of the multipliers, indicating that the DT FFT processor can be built with low linearity transconductors.

In the second part of this work, a prototype 0.13 μm CMOS integrated circuit implementation of the 8-point FFT was designed and fabricated based on the above architecture. The first prototype IC contained the two most critical circuit functions

to the approach: the serial-to-parallel converter and the discrete time FFT signal flow graph. The measured results demonstrate a dynamic range of 49 dB compared to 36 dB with the equivalent all digital approach. The measurements also demonstrate that the processor rejects large full scale narrow-band blockers, while maintaining greater than 40 dB of dynamic range. At the same time the processor enables a 10x reduction in power consumption compared to the equivalent all-digital approach, consuming only 25 mWatts, and reducing the required ADC bit depth by four bits, greatly reducing ADC power requirements by a factor of 16.

Based on the potential demonstrated by the first prototype DT FFT processor, a second iteration IC was designed to incorporate additional functionality required for UWB OFDM system applications. The design includes improved versions of the serial-to-parallel converter and the FFT signal flow graph, and the added functions of equalizer and parallel-to-serial converter. The second prototype allows simultaneous readout of all of the decoded sub-channels from the DT FFT processor, offering the potential for application as a complex baseband filter.

A summary of the contributions of this work are as follows:

1. A novel DT FFT processing architecture that improves UWB OFDM receivers by reducing power consumption and increasing dynamic range compared to the equivalent all digital approach. This includes the novel idea of applying discrete-time analog signal processing techniques to perform the FFT function for OFDM demodulation.
2. System simulation approaches for the new DT FFT processor that demonstrate its ability to improve performance in the UWB OFDM receiver. System level simulations quantify performance and help designers to make circuit specification trade-offs that optimize performance for their specific needs. Behavioral Models of the transconductor were introduced that improve the system level simulation speed. The system level simulations also lay the ground work for bounding performance and could be expanded to higher order FFT processors.
3. The design, layout and test of a proof-of-concept prototype IC in 0.13 μm CMOS that demonstrates performance enhancements. The proof-of-concept DT FFT processors has a dynamic range of 49 dB and consumes 25 mWatts.
4. The design and layout a second proof-of-concept prototype IC in 0.13 μm CMOS. The second proof-of-concept IC will expand the potential range of applications for

the DT FFT processor, by allowing modulations schemes other than OFDM to be tested.

5. Development of measurement setups to test the prototype DT FFT IC. Scripts were developed to link an arbitrary wave form generator and digitizing oscilloscope together with MATLAB for the calculation of EVM and SNDR. This capability to apply modulated waveforms to test circuits and calculate the output EVM and SNDR is beneficial to future research.

In addition the PIC micro-controller program and bias DACs used to generate the voltage biases for the prototype can be leveraged for future reconfigurable mixed-signal components.

Several peer reviewed publications and two patents were generated from this work:

- Mark Lehne, Sanjay Raman, “A Discrete-Time FFT Processor for Ultra Wide-band OFDM Wireless Transceivers, Part I: Architecture and Behavioral Modeling,” *submitted to IEEE Transactions on Circuits and Systems I*.
- Mark Lehne, Sanjay Raman, “A Discrete-Time FFT Processor for Ultra Wide-band OFDM Wireless Transceivers, Part II: Circuit Implementation and Measurement Results,” *submitted to IEEE Transactions on Circuits and Systems I*.
- Mark Lehne, Sanjay Raman, “A Prototype Analog/Mixed-Signal Fast Fourier Transform Processor IC for OFDM Receivers,” *2008 IEEE Radio and Wireless Symposium*, January 20, 2008, pp. 803-806.
- Mark Lehne, Sanjay Raman, “An Analog-Mixed-Signal Fourier Transform Pre-processor For Enhanced Dynamic Range In Broadband OFDM Receivers,” *IEEE 2006 Wireless and Microwave Tech. Conf.*, December 4, 2006.
- Mark Lehne, Sanjay Raman, “An Analog/Mixed-Signal FFT Processor for Wideband OFDM Systems,” *2006 IEEE Sarnoff Symposium on Advances in Wired and Wireless Communication*, March 27, 2006.
- Mark Lehne and Sanjay Raman, “An Analog/Mixed Signal Orthogonal Frequency Division Multiplexing Analog-to-Digital Converter Architecture” US Patent #60/784,468.

- Mark Lehne and Sanjay Raman, “An Analog Fourier Transform Channelizer and OFDM Receiver US Patent #11/698,816.

In addition to the results achieved in this dissertation, the potential for DT FFT processors raises new research questions and future work.

7.2 Future Work

Based on the experience gained from the 8-point discrete time FFT proof-of-concept IC demonstrated in this dissertation, several potential areas for future work were identified.

1. The second proof-of-concept IC remains to be fabricated. Measurement and analysis of the results from this IC should be performed.
2. The DT FFT processor could be expanded to handle larger sized FFTs. An N_{FFT} of 64, or 128 would increase the potential for adoption into current OFDM standards. The 8-point core, with equalizer and additional bank of adders is sufficient to process larger sized FFTs when the parallel input is broken down into sub-groups of 8-points. Some of the DSP algorithms used for higher order FFTs that are based on an 8-point FFT core for [42, 45] could be adapted to a discrete time signal processing implementation.
3. The DT FFT processor can be implemented as a discrete time filter for generalized use in receiver front-ends to null blocking signals. The second proof-of-concept IC includes the capability to test this function. Further analysis of spurious signals arising from the clocked nature of the DT FFT processor and their effect on the ultimate attenuation of nulled tones should be investigated.
4. The existing algorithms for DSP based equalizers could be analyzed to determine if a more efficient algorithm exists for use with the discrete time equalizer implementation. This algorithm would likely be mixed mode, making equalization decisions in DSP and driving sub-channel gain and phase correction values back to the discrete time analog equalizer bank. This investigation could lead to interesting approaches with advantages compared to the established OFDM equalization algorithms.

Appendix A

Verilog-AMS listings and SPICE Netlists

This appendix contains the Verilog-AMS listings of the behavioral models used in the DT FFT processor as well as the SPICE Netlists of the block level circuits.

Figure A.1: Verilog-AMS code of the G_m cell coefficient multiplier behavioral model

```

`include "disciplines.vams"
`include "constants.vams"
`define N 1
`define sign -1

module gm_diff_limits_new(inp, inn, outn, outp);
    input inp, inn;
    output outp, outn;
    voltage inp, inn;
    electrical outp, outn;

    parameter real gmdiff = 100e-6;
    parameter real a = 1;
    parameter real Ibias=40e-6;
    parameter real Ar = 0.0;
    parameter real gmoft = 0;
    parameter real vinoft = 0;
    parameter real gamma = 0;

    real vin, y, b, A1, A2, A3;

    analog begin
        vin = V(inp, inn) + vinoft;

        A1 = (1+Ar+gmoft-gamma);
        A2 = (1+gmoft);
        A3 = (1+Ar+gmoft+gamma);
        b = (2*Ibias/gmdiff - a)/(1+Ar+gamma);

        // Middle section -a < x < a
        y = 'sign*a*0.5*Ar/('N*'M_PI)*sin('M_PI/a*'N*vin)
            + (1+gmoft+0.5*Ar)*vin + gamma/(2*a)*vin*vin;

        // Right section a < x < b
        if (vin > a)
            y = 0.5*(A3*(b-a)/'M_PI*sin('M_PI*(vin-a)/(b-a))
                + A3*vin+a*A2);
        // Left section -b < x < -a
        if (vin < -a)
            y = 0.5*(A1*(a-b)/'M_PI*sin('M_PI*(vin+a)/(a-b))
                + A1*vin-a*A2);
        // Right zero x < b
        if (vin > b)
            y = A3*b/2+A2*a/2;
        // Left zero -b < x
        if (vin < -b)
            y = -A1*b/2-A2*a/2;

        I(outp) <+ -0.5*gmdiff*y;
        I(outn) <+ 0.5*gmdiff*y;

    end
endmodule

```

Figure A.2: Verilog-AMS code of the Sample-and-Hold Amplifier behavioral model

```

'include "disciplines.vams"
'include "constants.vams"

module SHA1(in,clk,out);
    input in,clk;
    output out;
    voltage in,out,clk;
    //electrical in;

    parameter real max_out = 1.2;
    parameter real min_out = 0.3;
    parameter real Vgain   = 1.0;
    parameter real offset  = 0.4;

    real vout;

    analog begin
        @(cross(V(clk)-0.5,+1)) begin
            vout = Vgain*V(in) + offset;
            if (vout > max_out)
                vout = max_out;
            if (vout < min_out)
                vout = min_out;
            end

            V(out) <+ transition(vout,0,1e-10);
        end
    endmodule

```

Figure A.3: Verilog-AMS code of the adder

```

'include "disciplines.vams"
'include "constants.vams"
'define Vdd 1.8

module adder_ideal_va(inp,inn);
    inout inp,inn;
    electrical inp,inn;

    parameter real Rload = 10e3;

    real Idiff;

    analog begin
        Idiff = I(inp) - I(inn); //sign inversion
        V(inp) <+ 'Vdd - 0.5*Rload*Idiff;
        V(inn) <+ 'Vdd + 0.5*Rload*Idiff;

        end
    endmodule

```

Figure A.4: Verilog-AMS code of the Serial-to-Parallel Function

```

'include "disciplines.vams"
'include "constants.vams"
'define Nlen 10

module S2P_one2ten(in,clkN,out0,out1,out2,out3,out4,out5,out6,out7,out8,out9);
    input in,clkN;
    output out0,out1,out2,out3,out4,out5,out6,out7,out8,out9;
    electrical in;
    voltage clkN;
    voltage out0,out1,out2,out3,out4,out5,out6,out7,out8,out9;
    voltage mid0,mid1,mid2,mid3,mid4,mid5,mid6,mid7; //,mid8,mid9
    voltage clk0,clk1,clk2,clk3,clk4,clk5,clk6,clk7,clk8,clk9,clk9b;

    parameter real max_out = 1.2;
    parameter real min_out = 0.3;
    parameter real Vgain = 1.0;

    real vout,clk_pulse;
    real cp0,cp1,cp2,cp3,cp4,cp5,cp6,cp7,cp8,cp9;
    integer count,clks,clksN,ckA;

//Module Connections
    SHA1 #(.max_out(max_out),.min_out(min_out),.Vgain(Vgain)) shaN0( in,clk0,mid0);
    SHA1 #(.max_out(max_out),.min_out(min_out),.Vgain(Vgain)) shaP0(mid0,clk9,out0);
    SHA1 #(.max_out(max_out),.min_out(min_out),.Vgain(Vgain)) shaN1( in,clk1,mid1);
    SHA1 #(.max_out(max_out),.min_out(min_out),.Vgain(Vgain)) shaP1(mid1,clk9,out1);
    SHA1 #(.max_out(max_out),.min_out(min_out),.Vgain(Vgain)) shaN2( in,clk2,mid2);
    SHA1 #(.max_out(max_out),.min_out(min_out),.Vgain(Vgain)) shaP2(mid2,clk9,out2);
    SHA1 #(.max_out(max_out),.min_out(min_out),.Vgain(Vgain)) shaN3( in,clk3,mid3);
    SHA1 #(.max_out(max_out),.min_out(min_out),.Vgain(Vgain)) shaP3(mid3,clk9,out3);
    SHA1 #(.max_out(max_out),.min_out(min_out),.Vgain(Vgain)) shaN4( in,clk4,mid4);
    SHA1 #(.max_out(max_out),.min_out(min_out),.Vgain(Vgain)) shaP4(mid4,clk9,out4);
    SHA1 #(.max_out(max_out),.min_out(min_out),.Vgain(Vgain)) shaN5( in,clk5,mid5);
    SHA1 #(.max_out(max_out),.min_out(min_out),.Vgain(Vgain)) shaP5(mid5,clk9,out5);
    SHA1 #(.max_out(max_out),.min_out(min_out),.Vgain(Vgain)) shaN6( in,clk6,mid6);
    SHA1 #(.max_out(max_out),.min_out(min_out),.Vgain(Vgain)) shaP6(mid6,clk9,out6);
    SHA1 #(.max_out(max_out),.min_out(min_out),.Vgain(Vgain)) shaN7( in,clk7,mid7);
    SHA1 #(.max_out(max_out),.min_out(min_out),.Vgain(Vgain)) shaP7(mid7,clk9,out7);
    SHA1 #(.max_out(max_out),.min_out(min_out),.Vgain(Vgain)) shaN8( in,clk8,mid8);
    SHA1 #(.max_out(max_out),.min_out(min_out),.Vgain(Vgain)) shaP8(mid8,clk9,out8);
    SHA1 #(.max_out(max_out),.min_out(min_out),.Vgain(Vgain)) shaN9( in,clk9,mid9);
    SHA1 #(.max_out(max_out),.min_out(min_out),.Vgain(Vgain)) shaP9(mid9,clk9b,out9);

//Analog Behavior
    analog begin
        @(initial_step) begin
            count = -1;
            clks = 0;
            clksN = 1;
            ckA = max_out-min_out;
        end

        @(cross(V(clkN)-0.6,+1)) begin
            count = count + 1;
            clks = 1;
            if (count > ('Nlen-1))
                count = 0;
    end

```

Figure A.5: cont. Verilog-AMS code of the Serial-to-Parallel Function

```

cp0 = (count == 0) ? 1 : 0;
cp1 = (count == 1) ? 1 : 0;
cp2 = (count == 2) ? 1 : 0;
cp3 = (count == 3) ? 1 : 0;
cp4 = (count == 4) ? 1 : 0;
cp5 = (count == 5) ? 1 : 0;
cp6 = (count == 6) ? 1 : 0;
cp7 = (count == 7) ? 1 : 0;
cp8 = (count == 8) ? 1 : 0;
cp9 = (count == 9) ? 1 : 0;
clksN = 0;
end

@(cross(V(clkN)-0.6,-1)) begin
    clks = 0;
    clksN = 1;           //An inverse clock to clks
end

//make only half clock period wide.
V(clk0) <+ cp0*clks;
V(clk1) <+ cp1*clks;
V(clk2) <+ cp2*clks;
V(clk3) <+ cp3*clks;
V(clk4) <+ cp4*clks;
V(clk5) <+ cp5*clks;
V(clk6) <+ cp6*clks;
V(clk7) <+ cp7*clks;
V(clk8) <+ cp8*clks;
V(clk9) <+ cp9*clks;
V(clk9b) <+ cp9*clksN;
end
endmodule

```

Figure A.6: Verilog-AMS code of the Parallel-to-Serial Function

```

'include "disciplines.vams"
'include "constants.vams"

module P2S_ten2one(in0,in1,in2,in3,in4,in5,in6,in7,in8,in9,clk,out);
    input    clk;
    input    in0,in1,in2,in3,in4,in5,in6,in7,in8,in9;
    output   out;
    voltage in0,in1,in2,in3,in4,in5,in6,in7,in8,in9;
    voltage clk;
    voltage out;

    parameter real max_out = 1.2;
    parameter real min_out = 0.3;
    parameter real Vgain = 1.0;

    real vout;
    integer count;

    analog begin
        @(initial_step) begin
            count = -1;
        end

        @(cross(V(clk)-0.6,+1)) begin
            count = count + 1;
            if (count > 9)
                count = 0;

            if (count == 0)
                vout = V(in0);
            if (count == 1)
                vout = V(in1);
            if (count == 2)
                vout = V(in2);
            if (count == 3)
                vout = V(in3);
            if (count == 4)
                vout = V(in4);
            if (count == 5)
                vout = V(in5);
            if (count == 6)
                vout = V(in6);
            if (count == 7)
                vout = V(in7);
            if (count == 8)
                vout = V(in8);
            if (count == 9)
                vout = V(in9);
        end

        V(out) <+ transition(vout,0,1e-10);
    end
endmodule

```

Figure A.7: SPICE netlist of the Butterfly Structure for P1N1

```
Options ResourceUsage=yes UseNutmegFormat=no TopDesignName="C:\users\default\
FFTA_Veriloga_prj\networks\test_MATLAB_FFTprv5"

vn=1.0507e-8

;Noise Source Connections
V_Source:Vn1      inA1 inA1n V_Noise=vn SaveCurrent=1
V_Source:Vn2      inA3 inA3n V_Noise=vn SaveCurrent=1
V_Source:Vn3      inB1 inB1n V_Noise=vn SaveCurrent=1
V_Source:Vn4      inB3 inB3n V_Noise=vn SaveCurrent=1
V_Source:Vn5      inA1 inC1n V_Noise=vn SaveCurrent=1
V_Source:Vn6      inA3 inC3n V_Noise=vn SaveCurrent=1
V_Source:Vn7      inB1 inD1n V_Noise=vn SaveCurrent=1
V_Source:Vn8      inB3 inD3n V_Noise=vn SaveCurrent=1

;Module Connections
gm_diff_limits_new:Gm1  inA1 inA3 outA1 outA3      gmdiff=Gmdiff a=a Ibias=Ibias Ar=Ar
    stat{ gauss +/- Ar_std % } gmoft=gmoft stat{ gauss +/- gm_std } vnoft=vnoft stat{
        gauss +/- voff_std } gamma=gamma nostat{ gauss +/- slope_std }
gm_diff_limits_new:Gm2  inA2 inA4 outA2 outA4      gmdiff=Gmdiff a=a Ibias=Ibias Ar=Ar
    stat{ gauss +/- Ar_std % } gmoft=gmoft stat{ gauss +/- gm_std } vnoft=vnoft stat{
        gauss +/- voff_std } gamma=gamma nostat{ gauss +/- slope_std }
gm_diff_limits_new:Gm3  inA1 inA3 outB1 outB3      gmdiff=Gmdiff a=a Ibias=Ibias Ar=Ar
    stat{ gauss +/- Ar_std % } gmoft=gmoft stat{ gauss +/- gm_std } vnoft=vnoft stat{
        gauss +/- voff_std } gamma=gamma nostat{ gauss +/- slope_std }
gm_diff_limits_new:Gm4  inA2 inA4 outB2 outB4      gmdiff=Gmdiff a=a Ibias=Ibias Ar=Ar
    stat{ gauss +/- Ar_std % } gmoft=gmoft stat{ gauss +/- gm_std } vnoft=vnoft stat{
        gauss +/- voff_std } gamma=gamma nostat{ gauss +/- slope_std }

;This is for the +1-j and -1+j rotate
gm_diff_limits_new:Gm5  inB1 inB3 outA1 outA3      gmdiff=GmdiffC a=a Ibias=Ibias Ar=Ar
    stat{ gauss +/- Ar_std % } gmoft=gmoft stat{ gauss +/- gm_std } vnoft=vnoft stat{
        gauss +/- voff_std } gamma=gamma nostat{ gauss +/- slope_std }
gm_diff_limits_new:Gm6  inB2 inB4 outA1 outA3      gmdiff=GmdiffS a=a Ibias=Ibias Ar=Ar
    stat{ gauss +/- Ar_std % } gmoft=gmoft stat{ gauss +/- gm_std } vnoft=vnoft stat{
        gauss +/- voff_std } gamma=gamma nostat{ gauss +/- slope_std }
gm_diff_limits_new:Gm7  inB1 inB3 outA4 outA2      gmdiff=GmdiffS a=a Ibias=Ibias Ar=Ar
    stat{ gauss +/- Ar_std % } gmoft=gmoft stat{ gauss +/- gm_std } vnoft=vnoft stat{
        gauss +/- voff_std } gamma=gamma nostat{ gauss +/- slope_std }
gm_diff_limits_new:Gm8  inB2 inB4 outA2 outA4      gmdiff=GmdiffC a=a Ibias=Ibias Ar=Ar
    stat{ gauss +/- Ar_std % } gmoft=gmoft stat{ gauss +/- gm_std } vnoft=vnoft stat{
        gauss +/- voff_std } gamma=gamma nostat{ gauss +/- slope_std }

gm_diff_limits_new:Gm9  inB1 inB3 outB3 outB1      gmdiff=GmdiffC a=a Ibias=Ibias Ar=Ar
    stat{ gauss +/- Ar_std % } gmoft=gmoft stat{ gauss +/- gm_std } vnoft=vnoft stat
    { gauss +/- voff_std } gamma=gamma nostat{ gauss +/- slope_std }
gm_diff_limits_new:Gm10 inB2 inB4 outB3 outB1      gmdiff=GmdiffS a=a Ibias=Ibias Ar=Ar
    stat{ gauss +/- Ar_std % } gmoft=gmoft stat{ gauss +/- gm_std } vnoft=vnoft stat
    { gauss +/- voff_std } gamma=gamma nostat{ gauss +/- slope_std }
gm_diff_limits_new:Gm11 inB1 inB3 outB2 outB4      gmdiff=GmdiffS a=a Ibias=Ibias Ar=Ar
    stat{ gauss +/- Ar_std % } gmoft=gmoft stat{ gauss +/- gm_std } vnoft=vnoft stat
    { gauss +/- voff_std } gamma=gamma nostat{ gauss +/- slope_std }
gm_diff_limits_new:Gm12 inB2 inB4 outB4 outB2      gmdiff=GmdiffC a=a Ibias=Ibias Ar=Ar
    stat{ gauss +/- Ar_std % } gmoft=gmoft stat{ gauss +/- gm_std } vnoft=vnoft stat
    { gauss +/- voff_std } gamma=gamma nostat{ gauss +/- slope_std }

adder_ideal_va:adder1  outA1 outA3      Rload=Rload
adder_ideal_va:adder2  outA2 outA4      Rload=Rload
adder_ideal_va:adder3  outB1 outB3      Rload=Rload
adder_ideal_va:adder4  outB2 outB4      Rload=Rload
```

Figure A.8: SPICE netlist of the AMS FFT Processor

```
Options ResourceUsage=yes UseNutmegFormat=no TopDesignName="C:\users\default\
FFTA_Veriloga_prj\networks\test_MATLAB_FFTprv4"

define Butterfly_N1_GMcell_netNS( inA1 inA2 inA3 inA4 inB1 inB2 inB3 inB4 outA1 outA2
    outA3 outA4 outB1 outB2 outB3 outB4 )
parameters Ibias=10e-6 Gmdiff=10e-6 Rload=100e3
#ifndef inc_C__users_default_FFTA_Veriloga_prj_networks_Butterfly_N1_GMcell_netNS
#define inc_C__users_default_FFTA_Veriloga_prj_networks_Butterfly_N1_GMcell_netNS
    inc_C__users_default_FFTA_Veriloga_prj_networks_Butterfly_N1_GMcell_netNS
#include "C:\users\default\FFTA_Veriloga_prj\networks\Butterfly_N1_GMcell_netNS.net"
#endif
end Butterfly_N1_GMcell_netNS

define Butterfly_PJNJ_GMcell_netNS( inA1 inA3 inA2 inA4 inB1 inB3 inB2 inB4 outA1 outA3
    outA2 outA4 outB1 outB3 outB2 outB4 )
parameters Ibias=10e-6 Gmdiff=10e-6 Rload=100e3
#ifndef inc_C__users_default_FFTA_Veriloga_prj_networks_Butterfly_PJNJ_GMcell_netNS
#define inc_C__users_default_FFTA_Veriloga_prj_networks_Butterfly_PJNJ_GMcell_netNS
    inc_C__users_default_FFTA_Veriloga_prj_networks_Butterfly_PJNJ_GMcell_netNS
#include "C:\users\default\FFTA_Veriloga_prj\networks\Butterfly_PJNJ_GMcell_netNS.net"
#endif
end Butterfly_PJNJ_GMcell_netNS

define Butterfly_P1N1_GMcell_netNS( inA1 inA3 inA2 inA4 inB1 inB3 inB2 inB4 outA1 outA3
    outA2 outA4 outB1 outB3 outB2 outB4 )
parameters Ibias=10e-6 Gmdiff=10e-6 GmdiffC=10e-6 GmdiffS=10e-6 Rload=100e3
#ifndef inc_C__users_default_FFTA_Veriloga_prj_networks_Butterfly_P1N1_GMcell_netNS
#define inc_C__users_default_FFTA_Veriloga_prj_networks_Butterfly_P1N1_GMcell_netNS
    inc_C__users_default_FFTA_Veriloga_prj_networks_Butterfly_P1N1_GMcell_netNS
#include "C:\users\default\FFTA_Veriloga_prj\networks\Butterfly_P1N1_GMcell_netNS.net"
#endif
end Butterfly_P1N1_GMcell_netNS

define Butterfly_P3N3_GMcell_netNS( inA1 inA3 inA2 inA4 inB1 inB3 inB2 inB4 outA1 outA3
    outA2 outA4 outB1 outB3 outB2 outB4 )
parameters Ibias=10e-6 Gmdiff=10e-6 GmdiffC=10e-6 GmdiffS=10e-6 Rload=100e3
#ifndef inc_C__users_default_FFTA_Veriloga_prj_networks_Butterfly_P3N3_GMcell_netNS
#define inc_C__users_default_FFTA_Veriloga_prj_networks_Butterfly_P3N3_GMcell_netNS
    inc_C__users_default_FFTA_Veriloga_prj_networks_Butterfly_P3N3_GMcell_netNS
#include "C:\users\default\FFTA_Veriloga_prj\networks\Butterfly_P3N3_GMcell_netNS.net"
#endif
end Butterfly_P3N3_GMcell_netNS

S2P_one2ten:S2P_1 in1 clk mida_p1s0 mida_p1s1 mida_p1s2 mida_p1s3 mida_p1s4
    mida_p1s5 mida_p1s6 mida_p1s7 mida_p1s8 mida_p1s9 max_out=1.2 min_out=-1.2 Vgain
    =1.0
S2P_one2ten:S2P_3 in3 clk mida_p3s0 mida_p3s1 mida_p3s2 mida_p3s3 mida_p3s4
    mida_p3s5 mida_p3s6 mida_p3s7 mida_p3s8 mida_p3s9 max_out=1.2 min_out=-1.2 Vgain
    =1.0
S2P_one2ten:S2P_2 in2 clk mida_p2s0 mida_p2s1 mida_p2s2 mida_p2s3 mida_p2s4
    mida_p2s5 mida_p2s6 mida_p2s7 mida_p2s8 mida_p2s9 max_out=1.2 min_out=-1.2 Vgain
    =1.0
S2P_one2ten:S2P_4 in4 clk mida_p4s0 mida_p4s1 mida_p4s2 mida_p4s3 mida_p4s4
    mida_p4s5 mida_p4s6 mida_p4s7 mida_p4s8 mida_p4s9 max_out=1.2 min_out=-1.2 Vgain
    =1.0
```

Bibliography

- [1] A. Batra, J. Balakrishnan, G. R. Aiello, J. R. Foerster, and A. Dabak, “Design of a Multiband OFDM System for Realistic UWB Channel Environments,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 52, no. 9, pp. 2123–2138, 2004.
- [2] F. C. Commission, “First Report and Order in the Frequency (MHz) Matter of Revision of Part 15 of the Commission’s Rules Regarding Spectral Nulling with Dummy Tones 1=16 Ultra-Wide Band Transmission Systems,” *ET Docket*, vol. 02-48, 2002.
- [3] *Multiband OFDM Physical Layer Specification*,. Release 1.0, Jan. 2005 [Online]. Available: <http://www.multibandofdm.org>.
- [4] R. Prasad, *Universal Wireless Personal Communications*. Boston: Artech House, 1998.
- [5] T. Yang, T. Yang, W. A. Davis, and W. L. Stutzman, “Small, Planar, Ultra-Wideband Antennas with Top-Loading,” in *2005 IEEE Antennas and Propagation Society International Symposium*, W. A. Davis, Ed., vol. 2A, 2005, pp. 479–482 vol. 2A.
- [6] D. Hibbard, “The Impact of Signal Bandwidth on Indoor Wireless Systems in Dense Multipath Environments,” Ph.D. dissertation, Virginia Polytechnic Institute and State University, 2004.
- [7] A. Saleh and R. Valenzuela, “A Statistical Model for Indoor Multipath Propagation,” *IEEE Journal on Selected Areas in Communications*, vol. 5, no. 2, pp. 128–137, 1987.

- [8] C. Anderson, "Design and Implementation of an Ultrabroadband Millimeter-Wavelength Vector Sliding Correlator Channel Sounder and In-Building Multipath Measurements at 2.5 and 60 GHz," Ph.D. dissertation, Virginia Polytechnic Institute and State University, 2002.
- [9] T. S. Rappaport, *Wireless Communications: Principles and Practice*. Upper Saddle River: Prentice Hall, 1996.
- [10] J. Foerster and Q. Li, "Channel Modeling Sub-Committee Report Final," *IEEE P802.15-02/490r1-SG3a*, November 2002.
- [11] J. B. Andersen, T. S. Rappaport, and S. Yoshida, "Propagation Measurements and Models for Wireless Communications Channels," *IEEE Communications Magazine*, vol. 33, no. 1, pp. 42–49, 1995.
- [12] J. G. Proakis, *Digital Communications*. New York: McGraw-Hill, 2001.
- [13] R. Prasad, *OFDM for Wireless Communications Systems*. Boston: Artech House, 2004.
- [14] A. V. Oppenheim, A. S. Willsky, and I. T. Young, *Signals and Systems*, 1st ed. Prentice Hall, 1994.
- [15] J. Balakrishnan, A. Batra, and A. Dabak, "A Multi-Band OFDM System for UWB Communication," *IEEE Conference on Ultra Wideband Systems and Technologies*, pp. 354–358, 2003.
- [16] T. H. Lee, *The Design of CMOS Radio-Frequency Integrated Circuits*. Cambridge University Press, 2000.
- [17] M. Gustavsson, J. J. Wikner, and N. N. Tan, *CMOS Data Converters for Communications*. Boston: Kluwer Academic Publishers, 2000.
- [18] R. Baker, Jacob, *CMOS Mixed-Signal Circuit Design*. Wiley-IEEE Press, 2002.
- [19] M. Gustavsson, J. Wikner, and N. Tan, *CMOS Data Converters for Communications*. Boston: Kluwer Academic, 2000.
- [20] A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing*, 2nd ed. Prentice Hall, 1999.

- [21] A. Ismail and A. A. Abidi, "A 3.1- to 8.2-GHz Zero-IF Receiver and Direct Frequency Synthesizer in 0.18- μm SiGe BiCMOS for Mode-2 MB-OFDM UWB Communication," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 12, pp. 2573–82, 2005.
- [22] Y.-H. Chen, C.-W. Wang, C.-F. Lee, T.-Y. Yang, C.-F. Liao, G.-K. Ma, and S.-I. Liu, "A 0.18 μm CMOS Receiver for 3.1 to 10.6GHz MB-OFDM UWB Communication Systems," *IEEE Radio Frequency Integrated Circuits Symposium*, p. 4 pp., 2006.
- [23] A. Valdes-Garcia, C. Mishra, F. Bahmani, J. Silva-Martinez, and E. Sanchez-Sinencio, "An 11-band 3.4 to 10.3 GHz MB-OFDM UWB Receiver in 0.25 μm SiGe BiCMOS," *Symposium on VLSI Circuits*, p. 2 pp., 2006.
- [24] A. Tanaka, H. Okada, H. Kodama, and H. Ishikawa, "A 1.1v 3.1-to-9.5GHz MB-OFDM UWB Transceiver in 90nm CMOS," *IEEE International Solid-State Circuits Conference*, p. 10 pp., 2006.
- [25] B. Shi and M. Y. W. Chia, "A 3.1-10.6 GHz RF Front-End for Multiband UWB Wireless Receivers," *IEEE Radio Frequency Integrated Circuits (RFIC) Symposium*, pp. 343–6, 2005.
- [26] B. Razavi, T. Aytur, Y. Fei-Ran, Y. Ran-Hong, K. Han-Chang, H. Cheng-Chung, and L. Chao-Cheng, "A 0.13 μm CMOS UWB Transceiver," *IEEE International Solid-State Circuits Conference*, vol. Vol. 1, pp. 216–594, 2005.
- [27] B. Brannon and C. Cloninger, "Redefining the Role of ADCs in Wireless," *Applied Microwave and Wireless*, vol. 13, no. 3, pp. 94–96, 2001.
- [28] B. Brannon, "Wide-Dynamic-Range A/D Converters Pave the Way for Wide-band Digital-Radio Receivers," *EDN*, vol. 41, no. 23, pp. 187–92, 1996.
- [29] S. K. Gupta, M. A. Inerfield, and J. Wang, "A 1-GS/s 11-bit ADC With 55-dB SNDR, 250-mW Power Realized by a High Bandwidth Scalable Time-Interleaved Architecture," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 12, pp. 2650–2657, 2006.
- [30] R. C. Taft, C. A. Menkus, M. R. 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.
- [31] C. Sandner, M. Clara, A. Santner, T. Hartig, and F. Kuttner, “A 6-bit 1.2-GS/s Low-Power Flash-ADC in 0.13- μm Digital CMOS,” *IEEE Journal of Solid-State Circuits*, vol. 40, no. 7, pp. 1499–1505, 2005.
- [32] D. L. Shen and T. C. Lee, “A 6-bit 800-MS/s Pipelined A/D Converter With Open-Loop Amplifiers,” *IEEE Journal of Solid-State Circuits*, vol. 42, no. 2, pp. 258–268, 2007.
- [33] X. Jiang and M. C. F. Chang, “A 1-GHz Signal Bandwidth 6-bit CMOS ADC With Power-Efficient Averaging,” *IEEE Journal of Solid-State Circuits*, vol. 40, no. 2, pp. 532–535, 2005.
- [34] P. C. S. Scholtens and M. Vertregt, “A 6-b 1.6-Gsample/s Flash ADC in 0.18- μm CMOS Using Averaging Termination,” *IEEE Journal of Solid-State Circuits*, vol. 37, no. 12, pp. 1599–1609, 2002.
- [35] M. Choi and A. A. Abidi, “A 6-b 1.3-Gsample/s A/D Converter in 0.35- μm CMOS,” *IEEE Journal of Solid-State Circuits*, vol. 36, no. 12, pp. 1847–1858, 2001.
- [36] B. Yu and J. Black, W. C., “A 900 MS/s 6b Interleaved CMOS Flash ADC,” *IEEE Conference on Custom Integrated Circuits*, pp. 149–152, 2001.
- [37] K. Uyttenhove and M. S. J. Steyaert, “A 1.8-V 6-bit 1.3-GHz Flash ADC in 0.25- μm CMOS,” *IEEE Journal of Solid-State Circuits*, vol. 38, no. 7, pp. 1115–1122, 2003.
- [38] C. Paulus, H. M. Bluthgen, M. Low, E. Sicheneder, N. Bruls, A. Courtois, M. Tiebout, and R. Thewes, “A 4GS/s 6b Flash ADC in 0.13 μm CMOS,” *Symposium on VLSI Circuits*, pp. 420–423, 2004.
- [39] B. Le, T. W. Rondeau, J. H. Reed, and C. W. Bostian, “Analog-to-Digital Converters,” *IEEE Signal Processing Magazine*, vol. 22, no. 6, pp. 69–77, 2005.
- [40] R. H. Walden, “Analog-to-Digital Converter Survey and Analysis,” *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 4, pp. 539–550, 1999.

- [41] B. Murmann and E. Boser, Bernhard, *Digitally Assisted Pipeline ADCs: Theory and Implementation*. Springer, 2004.
- [42] G. Zhong, F. Xu, and J. Willson, A. N., "A Power-Scalable Reconfigurable FFT/IFFT IC Based on a Multi-Processor Ring," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 2, pp. 483–495, 2006.
- [43] H.-Y. Liu, C.-C. Lin, Y.-W. Lin, C.-C. Chung, K.-L. Lin, W.-C. Chang, L.-H. Chen, H.-C. Chang, and C.-Y. Lee, "A 480Mb/s LDPC-COFDM-Based UWB Baseband Transceiver," *IEEE International Solid-State Circuits Conference*, pp. 444–609 Vol. 1, 2005.
- [44] J. Lee, H. Lee, S.-i. Cho, and S.-S. Choi, "A High-Speed, Low-Complexity Radix-2⁴ FFT Processor for MB-OFDM UWB Systems," *IEEE International Symposium on Circuits and Systems*, p. 4, 2006.
- [45] K. Maharatna, E. Grass, and U. Jagdhold, "A 64-point Fourier Transform Chip for High-Speed Wireless LAN Application Using OFDM," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 3, pp. 484–493, 2004.
- [46] R. S. Sherratt, O. Cadenas, N. Goswami, and S. Makino, "An Efficient Low Power FFT Implementation for Multiband Full-Rate Ultra-Wideband (UWB) Receivers," *Proceedings of the Ninth International Symposium on Consumer Electronics*, pp. 209–214, 2005.
- [47] Y. Jung, H. Yoon, and J. Kim, "New Efficient FFT Algorithm and Pipeline Implementation Results for OFDM/DMT Applications," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 1, pp. 14–20, 2003.
- [48] Y.-W. Lin, H.-Y. Liu, and C.-Y. Lee, "A 1-GS/s FFT/IFFT Processor for UWB Applications," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 8, pp. 1726–1735, 2005.
- [49] J. Thomson, B. Baas, E. M. Cooper, J. M. Gilbert, G. Hsieh, P. Husted, A. Lokanathan, J. S. Kuskin, D. McCracken, B. McFarland, T. H. Meng, D. Nakahira, S. Ng, M. Rattehalli, J. L. Smith, R. Subramanian, L. Thon, W. Yi-Hsiu, R. Yu, and Z. Xiaoru, "An Integrated 802.11a Baseband and MAC

- Processor,” *IEEE International Solid-State Circuits Conference*, vol. 1, pp. 126–451 vol.1, 2002.
- [50] D. L. G. Yeo, Z. Wang, B. Zhao, and Y. He, “Low Power Implementation of FFT/IFFT Processor for IEEE 802.11a Wireless LAN Transceiver,” *The 8th International Conference on Communication Systems*, vol. 1, pp. 250–254 vol.1, 2002.
- [51] D. Sun, A. Xotta, and A. A. Abidi, “A 1 GHz CMOS Analog Front-End for a Generalized PRML Read Channel,” *IEEE Journal of Solid-State Circuits*, vol. 40, no. 11, pp. 2275–2285, 2005.
- [52] X. Wang and R. R. S. Spencer, “A Low Power 170 MHz Discrete-Time Analog FIR Filter,” *Custom Integrated Circuits Conference*, pp. 13–16, 1997.
- [53] G. T. Uehara and P. R. Gray, “Parallelism in Analog and Digital PRML Magnetic Disk Read Channel Equalizers,” *IEEE Transactions on Magnetics*, vol. 31, no. 2, pp. 1174–1179, 1995.
- [54] B. E. Bloodworth, P. P. Siniscalchi, G. A. De Veirman, A. Jezdic, R. Pierson, and R. Sundararaman, “A 450-Mb/s Analog Front End for PRML Read Channels,” *IEEE Journal of Solid-State Circuits*, vol. 34, no. 11, pp. 1661–1675, 1999.
- [55] M. Q. Le, P. J. Hurst, and J. P. H. Keane, “An Adaptive Analog Noise-Predictive Decision-Feedback Equalizer,” *IEEE Journal of Solid-State Circuits*, vol. 37, no. 2, pp. 105–113, 2002.
- [56] K. Parsi, R. P. Burns, A. Chaiken, M. J. Chambers, W. R. Forni, D. Harnishfeger, S. Kaylor, M. J. Pennell, J. O. Perez, N. Rao, M. Rohrbaugh, M. Ross, and G. L. Stuhlmiller, “A PRML Read/Write Channel IC Using Analog Signal Processing for 200 Mb/s HDD,” *IEEE Journal of Solid-State Circuits*, vol. 31, no. 11, pp. 1817–1830, 1996.
- [57] M. D. Hahm, E. G. Friedman, and E. L. H. Titlebaum, “A Comparison of Analog and Digital Circuit Implementations of Low Power Matched Filters for Use in Portable Wireless Communication Terminals,” *IEEE Transactions on*

- Circuits and Systems II: Analog and Digital Signal Processing*, vol. 44, no. 6, pp. 498–506, 1997.
- [58] D. Jakonis, K. Folkesson, J. Dbrowski, P. Eriksson, and C. Svensson, “A 2.4-GHz RF Sampling Receiver Front-End in 0.18- μm CMOS,” *IEEE Journal of Solid-State Circuits*, vol. 40, no. 6, pp. 1265–1277, 2005.
 - [59] S. Lindfors, A. Parssinen, and K. A. I. Halonen, “A 3-V 230-MHz CMOS Decimation Subampler,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 50, no. 3, pp. 105–117, 2003.
 - [60] A. A. Abidi, “The Path to the Software-Defined Radio Receiver,” *IEEE Journal of Solid-State Circuits*, vol. 42, no. 5, pp. 954–966, 2007.
 - [61] X. Lin, J. Liu, H. Lee, and H. L. Liu, “A 2.5- to 3.5-Gb/s Adaptive FIR Equalizer With Continuous-Time Wide-Bandwidth Delay Line in 0.25 μm CMOS,” *IEEE Journal of Solid-State Circuits*, vol. 41, no. 8, pp. 1908–1918, 2006.
 - [62] X. Lin, S. Saw, and J. Liu, “A CMOS 0.25- μm Continuous-Time FIR Filter With 125 ps Per Tap Delay as a Fractionally Spaced Receiver Equalizer for 1-Gb/s Data Transmission,” *IEEE Journal of Solid-State Circuits*, vol. 40, no. 3, pp. 593–602, 2005.
 - [63] S. Hemati, A. H. Banihashemi, and C. Plett, “A 0.18 μm CMOS Analog Min-Sum Iterative Decoder for a (32,8) Low-Density Parity-Check (LDPC) Code,” *IEEE Journal of Solid-State Circuits*, vol. 41, no. 11, pp. 2531–2540, 2006.
 - [64] D. Vogrig, A. Gerosa, A. Neviani, A. G. Amat, G. Montorsi, and S. Benedetto, “A 0.35- μm CMOS Analog Turbo Decoder for the 40-bit Rate 1/3 UMTS Channel Code,” *IEEE Journal of Solid-State Circuits*, vol. 40, no. 3, pp. 753–762, 2005.
 - [65] V. C. Gaudet and P. G. Gulak, “A 13.3-Mb/s 0.35- μm CMOS Analog Turbo Decoder IC With a Configurable Interleaver,” *IEEE Journal of Solid-State Circuits*, vol. 38, no. 11, pp. 2010–2015, 2003.
 - [66] A. Demosthenous and J. Taylor, “A 100-Mb/s 2.8-V CMOS Current-Mode Analog Viterbi Decoder,” *IEEE Journal of Solid-State Circuits*, vol. 37, no. 7, pp. 904–910, 2002.

- [67] M. Ismail and T. Fiez, *Analog VLSI Signal Information Processing*. McGraw-Hill, Inc., 1994.
- [68] B. Razavi, *Principles of Data Conversion System Design*. IEEE Press, 1995.
- [69] —, *Design of Analog CMOS Integrated Circuits*. McGraw-Hill, 2000.
- [70] B. Gilbert, “A high-performance monolithic multiplier using active feedback,” *IEEE Journal of Solid-State Circuits*, vol. 9, no. 6, pp. 364–373, 1974.
- [71] J. N. Babanezhad and G. C. Temes, “A 20-V four-quadrant CMOS analog multiplier,” *IEEE Journal of Solid-State Circuits*, vol. 20, no. 6, pp. 1158–1168, 1985.
- [72] K. Bult and H. Wallinga, “A CMOS four-quadrant analog multiplier,” *IEEE Journal of Solid-State Circuits*, vol. 21, no. 3, pp. 430–435, 1986.
- [73] Z. Wang, “A CMOS four-quadrant analog multiplier with single-ended voltage output and improved temperature performance,” *IEEE Journal of Solid-State Circuits*, vol. 26, no. 9, pp. 1293–1301, 1991.
- [74] S.-I. Liu and Y.-S. Hwang, “CMOS four-quadrant multiplier using bias feedback techniques,” *IEEE Journal of Solid-State Circuits*, vol. 29, no. 6, pp. 750–752, 1994.
- [75] H. R. Mehrvarz and C. Y. Kwok, “A novel multi-input floating-gate MOS four-quadrant analog multiplier,” *IEEE Journal of Solid-State Circuits*, vol. 31, no. 8, pp. 1123–1131, 1996.
- [76] G. Han and E. Sanchez-Sinencio, “CMOS transconductance multipliers: a tutorial,” *IEEE Transactions on Circuits and Systems II: Analog Digital Signal Process.*, vol. 45, no. 12, pp. 1550–1563, 1998.
- [77] T.-C. Lee and B. Razavi, “A 125-MHz CMOS mixed-signal equalizer for gigabit ethernet on copper wire,” in *IEEE Conference on Custom Integrated Circuits, 2001.*, B. Razavi, Ed., 2001, pp. 131–134.
- [78] J. Abbott, C. Plett, and J. W. M. Rogers, “A 1.2V CMOS multiplier for 10 Gbit/s equalization,” in *Proceedings of the 31st European Solid-State Circuits Conference, 2005.*, C. Plett, Ed., 2005, pp. 379–382.

- [79] D. Johns and K. Martin, *Analog Integrated Circuit Design*. Oxford University Press, 1999.
- [80] J. G. Proakis and D. K. Manolakis, *Digital Signal Processing: Principles, Algorithms and Applications*. Prentice Hall, 1995.
- [81] *Agilent Advanced Design System*. [Online]. Available: <http://eesof.tm.agilent.com>.
- [82] *Cadence Virtuoso Multi-mode Simulation Environment*. [Online]. Available: <http://www.cadence.com>.
- [83] *Mathworks MATLAB*. [Online]. Available: <http://www.mathworks.com>.
- [84] A. Baschiroto, "A low-voltage sample-and-hold circuit in standard CMOS technology operating at 40 Ms/s," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 48, no. 4, pp. 394–399, 2001.
- [85] R. Baker, Jacob, *CMOS Circuit Design, Layout, and Simulation*. Wiley-IEEE Press, 2004.
- [86] S. Limotyrakis, S. D. Kulchyski, D. K. Su, and B. A. Wooley, "A 150-MS/s 8-b 71-mW CMOS time-interleaved ADC," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 5, pp. 1057–1067, 2005.
- [87] S. M. Louwsma, E. J. M. van Tuijl, M. Vertregt, S. P. C. S., and B. A. Nauta, "A 1.6GS/s, 16 Times Interleaved Track and Hold with 7.6 ENOB in 0.12 μ m CMOS," in *Proceedings of the 30th European Solid-State Circuits Conference, 2004*, E. J. M. van Tuijl, Ed., 2004, pp. 343–346.
- [88] D. Jakonis and C. Svensson, "A 1 GHz linearized CMOS track-and-hold circuit," in *IEEE International Symposium Circuits and Systems, 2002*, C. Svensson, Ed., vol. 5, 2002, pp. V–577–V–580 vol.5.
- [89] K. Martin, *Digital Integrated Circuit Design*. John Wiley and Sons, Inc, 1997.
- [90] M. Hansson and A. Alvandpour, "Power-Performance Analysis of Sinusoidally Clocked Flip-Flops," *23rd NORCHIP Conference, 2005.*, pp. 153–156, 2005.

- [91] D. Markovic, B. Nikolic, and R. W. Brodersen, "Analysis and Design of Low-Energy Flip-Flops," *International Symposium on Low Power Electronics and Design 2001.*, pp. 52–55, 2001.
- [92] W. J. Dally and J. W. Poulton, *Digital Systems Engineering.* Cambridge University Press, 1998.
- [93] H. Johnson, *High Speed Digital Design: A Handbook of Black Magic.* Prentice Hall, 1993.
- [94] Y.-H. Oh and S.-G. Lee, "An Inductance Enhancement Technique and its Application to a Shunt-Peaked 2.5 Gb/s Transimpedance Amplifier Design," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 51, no. 11, pp. 624–628, 2004.
- [95] S. S. Mohan, M. D. M. Hershenson, S. P. Boyd, and T. H. Lee, "Bandwidth extension in CMOS with optimized on-chip inductors," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 3, pp. 346–355, 2000.
- [96] S. M. R. Hasan, "Design of a low-power 3.5-GHz broad-band CMOS transimpedance amplifier for optical transceivers," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, no. 6, pp. 1061–1072, 2005.
- [97] B. Analui and A. Hajimiri, "Bandwidth Enhancement for Transimpedance Amplifiers," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 8, pp. 1263–1270, 2004.
- [98] C. D. Holdenried, M. W. Lynch, and J. W. Haslett, "Modified CMOS cherry-hooper amplifiers with source follower feedback in 0.35 μm technology," in *Proceedings of the 29th European Solid-State Circuits Conference, 2003. ESSCIRC '03.*, 2003, pp. 553–556.
- [99] 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, 0018-9200.
- [100] J. Sevic, *Transmission Line Transformers.* Noble, 1996.