

Graduate Theses, Dissertations, and Problem Reports

2020

Rail-to-Rail Operational in Low-Power Reconfigurable Analog Circuitry

Jared Dale Baker West Virginia University, jdbaker@mix.wvu.edu

Follow this and additional works at: https://researchrepository.wvu.edu/etd

Part of the Electrical and Electronics Commons, and the Engineering Education Commons

Recommended Citation

Baker, Jared Dale, "Rail-to-Rail Operational in Low-Power Reconfigurable Analog Circuitry" (2020). *Graduate Theses, Dissertations, and Problem Reports.* 7975. https://researchrepository.wvu.edu/etd/7975

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.

Rail-to-Rail Operation in Low-Power Reconfigurable Analog Circuitry

Jared D. Baker

Thesis submitted to the Benjamin M. Statler College of Engineering and Mineral Resources at West Virginia University in partial fulfillment of the requirements for the degree of

> Master of Science in Electrical Engineering

David W. Graham, Ph.D., Chair Jeremy Dawson, Ph.D. Daryl Reynolds, Ph.D.

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia 2020

Keywords: Rail-to-Rail Operation, Field Programmable Analog Array, Asynchronous Conversions, Extrema Sampling, Operational Amplifier

Copyright 2020 Jared D. Baker

Abstract

Rail-to-Rail Operation in Low-Power Reconfigurable Analog Circuitry

Jared D. Baker

Analog signal processing (ASP) can be used to decrease energy consumption by several orders of magnitude over completely digital applications. Low-power field programmable analog arrays (FPAA) have been previously used by analog designers to decrease energy consumption. Combining ASP with an FPAA, energy consumption of these systems can be further reduced. For ASP to be most functional, it must achieve rail-to-rail operation to maintain a high dynamic range. This work strives to further reduce power consumption in reconfigurable analog circuitry by presenting a novel data converter that utilizes ASP and rail-to-rail operation. Rail-to-Rail operation is achieved in the data converter with the use of an operational amplifier presented in this work. This efficient yet elementary data converter has been fabricated in a 0.5μ m standard CMOS process. Additionally, this work looks deeper into the challenges of students working remotely, how MATLAB can be used to create circuit design tools, and how these developmental tools can be used by circuit design students.

Acknowledgments

I would first like to thank Dr. David W. Graham for all of his guidance and help while completing the work in this thesis, as well as giving me the opportunity to work underneath of him for undergraduate and graduate research. I would also like to thank Steven Andryzcik for the countless questions that he answered and helping with anything that I needed. I would also like the thank all my other lab mates for help me along the war regarding research, classes, and general questions about life as a graduate student.

I also need to thank my family and friends who gave countless support and were always willing to lend a listening ear when I needed to talk about my research. A special thanks goes out to my mother, Julie and my father, Randy. Without there unconditional support over the years none of this work would have been possible.

Contents

A	Abstract			
A	cknov	vledgments	iii	
Li	st of	Figures	vii	
Li	st of	Tables	ix	
1	Intr 1.1	oduction Outline	$f 1 \ 3$	
2	Low 2.1 2.2 2.3 2.4 2.5	Power Reconfigurable Analog SystemsIntroduction to Reconfigurable CircuitryImportance of Reconfigurable Low-Power Analog CircuitryReconfigurable Analog/Mixed-signal Platform2.3.1Architecture of the RAMP2.3.2RAMP 1.1Development BoardImprovements to Future RAMPsChapter Summary	4 6 7 8 11 12 12	
3	Asy: 3.1 3.2 3.3	nchronous Extrema Sampling Analog-to-Digital Converter Synchronous Vs Asynchronous Conversions	13 14 14 16 17 17 18 19 20 20 21 22 22	
	3.4	Simulating Floating Gate Transistors Via LTspice	23 24 24	

	~ ~	3.4.2 Successive Approximation ADC	25
	3.5	Chapter Summary	26
4	Rai	l-to-Rail Operational Amplifier Design	28
	4.1	Rail-to-Rail Input Stage	28
		4.1.1 Complementary Input Pairs	29
		4.1.2 Dual N-Channel Input Pairs	29
		4.1.3 Dual P-Channel Input Pairs	30
	4.2	Rail-to-Rail Output Stage	30
	4.3	Design of Rail-to-Rail Operational Amplifier	31
		4.3.1 Input Stage Design	31
		4.3.2 Transconductance Compensation	34
		4.3.3 Input Stage Overview	36
		4.3.4 Output Stage	37
	4.4	Simulation Results	38
		4.4.1 DC Parameters	38
		4.4.2 Frequency and Transient Response	39
		4.4.3 Comparison With Other Amplifiers	39
	4.5	Use of Designed Amplifier in Extrema Sampling Circuit	40
	4.6	Chapter Summary	41
5	Circ	cuit Design using Free Software	43
-	5.1	Shortcomings of LTspice	44
	-	5.1.1 Parametric Analysis	44
		5.1.2 Data Visualization	44
		5.1.3 Random Number Generation	45
	5.2	Application Development for Student Use	45
	5.3	Operational Amplifier Simulation Aid (Zip Folder)	46
		5.3.1 Using the Operational Amplifier Simulation Aid	47
	5.4	Monte Carlo Simulations (Custom Interface)	49
		5.4.1 Creating a Cutsom Application Via MATLAB	49
		5.4.2 Monte Carlo Sumulation Application	50^{-0}
		5.4.3 Using Monte Carlo Simulation Application	52
	5.5	Summary and Future Work	52
6	Sun	nmary and Conclusions	54
D,	foro		56
10	eierei	inces	50
Α	App	bendix	60
	A.1	Code from Operational Amplifier Simulation Aid	60
		A.1.1 Main Script	60
		A.1.2 Data Formater Function	62
		A.1.3 Edit Netlist: Change DC Sweep	65
	A.2	Example Text File for Custom Application	65

 \mathbf{V}

CONTENTS

Approval Page

vi

i

List of Figures

2.1	(a) Signal Chain of an always-on digital sensor. (b) Signal Chain of an always- on sensing system that Incorporated the RAMPs analog signal processing.	6
2.2	Architechture of the BAMP 1 1 IC	8
2.3	(a) Standard P-channel MOSFET. (b) A FG Transistor made from a P-	Ŭ
	Channel MOSFET	10
2.4	IV curve of floating gate transistor	10
2.5	Ramp 1.1 Development Board	11
3.1	Sampling at the Nyquist rate the signal is sampled 20 times. While sampling	1 -
2.0	at local extrema the same signal is only sampled 9 times [1]	15
3.2	(a) Bezier curve reconstruction between two adjacent extrema. (b) Continuous	10
? ?	Disgram of AES ADC. The Apples Signal Procession contains the extreme	10
ა.ა	sampling system while the Asymphronous ADC contains the SA ADC based	
	upon the SA ADC presented in ^[1]	17
3.4	SkH1 and SkH2 are clocked at opposite times. This allows the outputs of	11
0.4	the systems to alternate which contains a higher value	17
35	The improvements from the Bail-to-Bail amplifier designed in chapter 4 are	11
0.0	easily seen compared to the current amplifier being used. Adding this amplifier	
	ultimately allows the AES-ADC to be a fully Rail-to-Rail system.	18
3.6	Shows the discontinuous non-linearity's that are experienced by a complemen-	-
	tary stage rail-to-rail comparator.	19
3.7	Floating gate implementation of N-stage of the comparator used in the SA-ADC.	20
3.8	A block Diagram of the Extrema Sampling System.	20
3.9	10-bit Successive Approximation Register [1]	22
3.10	10-bit parallel charging DAC. C is chosen by the designer, lower capacitance	
	values produce quicker switching speeds and uses significantly less die space	
	compare to larger values of C	23
3.11	Block Diagram of SA-ADC.	23
3.12	Extrema detection of a 100Hz sine wave using the system in Fig. 3.8. Each	
	black pulse denotes that a local extrema has been detected \ldots	24
3.13	Actual Vs Ideal Transfer Function with no FG compensation	25
3.14	Actual Vs Ideal Transfer Function with FG compensation	26
3.15	Integral Non-Linearity test of the AES-ADC	27

4.1	A simple Rail-to-Rail input stage using complementary Input Pairs	29	
4.2	Dual n-channel input pair used for rail-to-rail input stage	30	
4.3	Dual p-channel input pair rail-to-rail op amp.	31	
4.4	Rail-to-Rail input pair when $V_{sup} >= V_{sup,min}$	32	
4.5	Rail-to-Rail input pair when $V_{sup} \ll V_{sup,min}$	32	
4.6	Transconductance issue with simple complementary input stage	33	
4.7	Transconductance issue with simple complementary input stage	34	
4.8	Transconductace compensation circuit		
4.9	4.9 Locations on where the compensation circuit can be attached to either add c		
	subtract current during the transition region	36	
4.10	Compares the three different styles of compensation discussed in this chapter	36	
4.11	Input stage of rail-to-rail amplifier ignoring biasing circuitry	37	
4.12	Class AB output stage	38	
4.13	Full Schematic of designed operational amplifier. Excludes biasing circuitry.		
4.14	Total transconductance of the operational amplifier across the input common		
	mode range	39	
4.15	Magnitude and Phase response of the operational amplifier	40	
4.16	Transient response of 1kHz sine wave	41	
4.17	Displyas the voltage range of the fabricated system and the fabricated system		
	with the use of this amplifier.	42	
5.1	Example of DC simulation parameter's.	48	
5.2	Example of Normalized Transconductance output.	48	
5.3	Example of Transient Analysis.	49	
5.4	Example of Bode Plot Output from AC analysis.	50	
5.5	User Interface when Using Monte Carlo Simulations with results plotted from		
	adjusting threshold values of a digital inverter.	51	

List of Tables

2.1	Components in the first 10 stages of the RAMP 1.1	9
2.2	RAMP 1.1. Unique Stage Circuits	9
4.1	DC Parameters of Operational Amplifier	38
4.2	Comparison of Operational Amplifier	40

Chapter 1

Introduction

The introduction of reconfigurable circuitry revolutionized the circuit design flow. Traditionally circuit design uses the following design procedure: design, fabricate, test, and use in application. Reconfigurable circuitry has greatly reduced the design time for circuit designers. Instead of redesigning another circuit only to make a small change, designers can now reprogram an already fabricated system for their application. In 1984, Altera released its first version of a programmable array logic (PAL) circuit. The system that Altera released could be programmed and erased multiple times in a similar manor to EEPROM. Since then, vast improvements have been made to reconfigurable circuits. The most well-known reconfigurable circuit on the market today is the Field Programmable Gate Array (FPGA).

The analog counter part to the FPGA is the field programmable analog array (FPAA). Reconfigurability in analog circuits is vital due to complex system design. Analog design often requires precise biasing circuitry to ensure proper system operation. If an application specific circuit (ASIC) is designed but has an improper bias, the entire system will need to be redesigned and fabricated again. Reconfigurable systems contain multiple biases and elementary components that can be interchanged as needed.

Former researchers at West Virginia University have developed the Reconfigurable Analog/Mixed-Signal Platform (RAMP) [2]. The RAMP consists of a low-power FPAA and subsequent digital circuitry. This system allows circuit designers to reap the benefits of a low-power analog system, while having a digital interface for re-programming the circuit. Further operation of the RAMP itself is discussed in Chapter 2. The second generation of the RAMP (RAMP 1.1) covers a vast variety of applications, containing but not limited to: programmable current sources, analog filleting, and clock generation. The RAMP 1.1 is currently limited to laboratory settings. To bring the RAMP 1.1 into the field, a portable platform must be designed. In chapter 2 a portable platform for the RAMP 1.1 is presented. With the use of this platform the low-power operation that the RAMP 1.1 provides can be utilized in real-world applications.

While the RAMP 1.1 is an ideal system for low-power circuitry, the energy consumption can still be improved. Currently, all versions of the RAMP do not include an on-chip analogto-digital converter (ADC). With this a separate micro-control unit (MCU) is utilized for the ADC. The addition of an on-chip ADC would decrease energy consumption, ultimately increasing the battery life of the system. Chapter 3 presents an Asynchronous Extrema Sampling ADC (AES-ADC) that could be implemented on future RAMPs.

For any analog system to be effective it must be able to handle a variety of signals, ideally this would include any signal within the rails of the circuit. The RAMP 1.1 is not a fully rail-to-rail system. With this comes a limitation on the signals that can be input or output of the system. Currently pre-processing is required for signals coming into the RAMP 1.1, ultimately causing a less than ideal dynamic range. To further improve the RAMP 1.1, railto-rail operation needs to be achieved. The most immediate issue is buffering signals to and from the RAMP. Chapter 4 presents the design of a rail-to-rail amplifier to be implemented on future RAMPs to improve system versatility.

The covid-19 pandemic has caused circuit designers and students to transform their work environments. With working from home, designers and students may not have access to the expensive software and/or lab equipment to run their test. With this many are seeking out free simulation software such as LTspice, Pspice, OrCAD, etc. While free software allows for many different types of simulations, it is not ideal. With this chapter 5 presents how being a circuit design student has changed with working remotely; how third-party software such as MATLAB, can be used to help bridge the gaps between free and expensive software; how applications can be used to aid other students and the future of developmental tools.

The main objective of this work is to decrease energy consumption of the future RAMP devices while increasing system versatility. These goals have been realized with the design of a mobile platform for the RAMP 1.1, a novel analog-to-digital converter and a rail-to-rail amplifier. With these new additions to future RAMP systems, they will present decreased energy consumption allowing them to operate outside of the laboratory for longer periods of time. The secondary objective was to address the issues of students working remotely and how the covid-19 pandemic could change circuit design for future students.

1.1 Outline

The remainder of this work is organized as follows. Chapter 2 will provide the background of reconfigurable circuitry, the benefits of low power analog circuitry, the RAMP, a portable design platform for in field applications, and how the RAMP can be improved. Chapter 3 will discuss the design of an AES-ADC, the use of floating gate transistors to improve performance post-fabrication, and how the AES-ADC can be improved. Chapter 4 will discuss a rail-to-rail operational amplifier and the design process of the circuit. Chapter 5 will discuss a new design process and tool development to aid students while working remotely. Finally, chapter 6 will discuss a summary of this work.

Chapter 2

Low Power Reconfigurable Analog Systems

Low power reconfigurable analog systems have been shown to be useful in many applications in modern technologies. However, the emergence of analog design has been hindered due to the vast knowledge required to design analog circuits. The field programmable analog array (FPAA) has allowed many designers to easily design in the analog domain. To understand how these systems can be improved, we must first understand how these systems operate. Therefore, this chapter presents an introduction to reconfigurable circuitry, why low-power analog circuitry is important, a reconfigurable analog/mixed signal platform (RAMP), a portable in-field system for the RAMP, and how future RAMPs can be improved.

2.1 Introduction to Reconfigurable Circuitry

In 1984, Altera released its first version of a programmable array logic (PAL) circuit [3]. The system that Altera released could be programmed and erased multiple times in a similar manor to EEPROM. Since then vast improvements have been made to reconfigurable circuits. The most well-known reconfigurable circuit on the market today is the Field Programmable Gate Array (FPGA). A FPGA is and integrated circuit that was designed to be configured by the consumer for their specific application. The FPGA typically contains an array of programmable logic blocks, a reconfigurable connection matrix, and logic blocks. Elementary digital components such as logic gates, flip flops, memory elements, etc. are at the disposal of the consumer with these devices. While FPGA's are convenient in the aspect that they

can easily be reprogramed they fall short in several categories compared to the application specific integrated circuit (ASIC). ASIC's are designed for a singular purpose allowing them to be more efficient, smaller in size, and increased performance over FPGA's [4].

The FPGA has an analog counterpart, this is known as the Field Programmable Analog Array (FPAA). The term FPAA was first used in 1991 by Lee and Gulak [5], which was later stapled as the term for these devices. FPAAs are created with a similar structure goal in mind as the FPGA [6]. The FPAA should be configurable for the user's immediate and future needs. While the FPAA has obvious draw backs over an FPGA such as linearity, signal-to-noise ratios, bandwidths, etc. FPAAs can significantly reduce the power consumptions in some systems and even produce better results than ASIC designs [7]. This is due to the amount of post-fabrication optimization that can be completed with reconfigurable circuitry.

Both FPGAs and FPAAs contain elementary building blocks, and a connection matrix that allow the user to reconfigure the system. FPAAs contain computational analog blocks (CABs) and computational logic blocks and digital circuitry for routing [8]. CABs can contain basic building blocks such as operational amplifiers, filters, multipliers etc. Computational Logic Blocks (CLBs) are the same in FPGAs and FPAAs, they contain elementary reconfigurable logic gates. A FPAA will typically contain an array of CABs and CLBs. Each of these CABs and CLBs are interconnected via a switch matrix. These connections are completely reconfigurable and set by the user for their specific application.

FPAAs are typically programed via a digital interface using shift registers and multiplexers [2, 5, 9]. It is important to note that each FPAA is designed with special functions in mind. This ultimately limits the number of specialized CABs included. This is critical in controlling the number of input/outputs (I/O) that are needed to program the FPAA. The switch matrix can be broken down into two types: crossbar and multistage. Crossbar interconnection provides full interconnection capability between any connected elements [5]. This method provides a shorter delay on the data transfer but takes a large amount of area on the silicon [5]. On the contraries if die size is a main design factor the multistage method can be used, this comes with the downfall of longer delays between data transfer.

2.2 Importance of Reconfigurable Low-Power Analog Circuitry

Reconfigurability allows the designer to make changes to circuit design post fabrication. This is ideal in analog circuit design as it often require precise biasing to ensure expected operation. Several chips have been designed based upon the use of floating gate transistors (FGs) [2, 10]. FGs are used as programmable current sources than can adjust biasing conditions on circuit components. This allows for circuit optimization post fabrication [11]. Analog processing (ASP) can be utilized in reconfigurable circuitry [6]. Combining reconfigurability with ASP the energy consumption can be reduced by several orders of magnitude compared to an all-digital solution [2] This keeps the digital circuitry in a low-power state until relevant data is extracted.

Speech detection is an ideal appliciton for an FPAA [12]. Let us think of a voice activated system such as the Amazon Alexa, Google home, or other voice activated device. These devices use always on listening that allows them to determine when a user has spoken the wake-up word. With this approach the device is always turned on at full capability and is drawing the full amount of current needed for the system. This inherently is wasting power when nobody is talking, or a noise is made that is not a human voice. Analog signal processing is used to determine when the system should power up or stay in a low-power state. Fig. 2.1 shows a sample signal chain for one of these devices compared to the signal chain of the RAMP for a wake-up detection application.



Figure 2.1: (a) Signal Chain of an always-on digital sensor. (b) Signal Chain of an always-on sensing system that Incorporated the RAMPs analog signal processing.

Using the RAMP as a front-end analog signal processor can save over 1000 times power compared to a standard digital filter [13]. The RAMP is also able to perform signal analysis before the data is sent to the analog-digital converter. This works by selective sampling the signals that are coming into the RAMP. In [13] they demonstrate that a 12kHz signal can be selectively sampled down to 2kHz. This is necessary due to the often-low frequencies that MCU ADCs can sample at.

Using an FPAA over an ASIC give the designer the capability to optimize the circuit post fabrication [6]. Bias voltages/currents can be adjusted, components can be optimized, and even new components can be added to the system. One example that has been tested was for an acoustic vehicle detector [7]. It was presented that an ASIC for the application had a battery life of 2.4 years while the system developed on the RAMP had a battery life of 7.5 years.

2.3 Reconfigurable Analog/Mixed-signal Platform

Former researchers at West Virginia University have developed, fabricated, and tested this platform. RAMPs provide the much-needed flexibility for ultra-low-power pre-processing hardware [2]. This allows for quick reconfigurability that can easily be completed outside of a laboratory setting. The RAMP is crucial in saving power, this system can provide multiple orders of magnitude of power savings compared to completely digital systems [2, 13]. This is done by using always on sensing with the low power analog components of the system. When relevant data is detected, it is then converted into the digital domain for processing. This conversion process consumes large amounts of power and is therefore benefited from only being powered on when necessary.

The RAMP is a mixed-signal chip, it leverages both analog and digital circuitry to perform its tasks. The inclusion of digital circuitry alongside analog circuitry controls how the analog circuits interact with one another [2]. The digital circuits do not need to be incorporated with the analog circuits. If the users wished they can synthesize completely digital circuits on the RAMP. This system allows the RAMP to approach a one-size fits all solution for the analog needs of sensor nodes [2]. Other works have been investigated with mixed-signal design within FPAAs [2, 14, 15]. The Ramp was designed specifically with low-power wireless sensor applications in mind, it is fully self-contained with non-linear building blocks that can make decisions itself [2].

2.3.1 Architecture of the RAMP

The RAMP itself contains a mixed-signal FPAA, analog and digital circuits, and various control circuits. The RAMP and its contents were designed from the ground up for low power consumption [2]. The mixed signal FPAA contains CABs, CLBs, and a crossbar switch matrix. The switch matrix is designed with 4-way switches that allow for any CAB/CLB to be interconnected with another CAB/CLB. These switches are programmed via a serialperipheral interface (SPI) that allows the user to connect the components in their desired configuration. In total, 20,380 switches are included in the FPAA [2]. The CABs and CLBs are arranged into a stage/channel configuration [2]. The first version of the RAMP contains a total of 80 computational blocks that are in a 10 stage 8 channel configuration. All 8 channels are identical to help aid in parallel processing. The second version of the RAMP has an additional stage added deemed the "unique stage". As the term hints, each channel in the stage is a unique circuit. The signal chain flow is shown below in Fig 2.2. To allow for diverse operations the RAMP has multiple types of building blocks containing circuits and basic circuit components. Table 2.1 shows the computational elements that are in the 10 identical stages of the RAMP. Table 2.2 shows the circuits that are in the unique stage of the RAMP.



Figure 2.2: Architechture of the RAMP 1.1 IC.

The RAMPs CAB blocks can be broken down into 5 major subsections [2]. These subsets

8 BPFs	56 OTAs	8 Inverters	16 Envelope Detectors
8 LPFs	8 Multipliers	32 Comparators	48 Current Sources/Sinks
56 Caps	9 Op-Amps	8 Bump Circuits	16 Pulse Generators
8 PNPs	16 Resistors	8 Time-to-Voltage	16 Asymmetric Integrators
$16 \mathrm{S/Hs}$	$144 \ \mathrm{FETs}$	32 JK Flip Flops	16 6-input 2-output LUTs

Table 2.1: Components in the first 10 stages of the RAMP 1.1.

Channel	Circuit
0	Temerature Compensation
1	Neuron
2	BiQuad Filter
3	AGU Mux Offset
4	Band Gap Temperature Compensation
5	Wheatstone Bridge
6	Ring Oscillator
7	Reid Harrison Neural Amplifier

Table 2.2: RAMP 1.1. Unique Stage Circuits

are spectral analysis, transconductors, transistors, sensor interfacing, and mixed signal. The CLBs are essentially small FPGAs to be used within the FPAA. These digital circuits are used for control signals of the analog circuitry. A summary of the digital elements is shown in table 2.1. Many analog circuits require specific biasing values that cannot be set with the digital circuitry. These values are biased with floating gate transistors (FG).

A floating gate transistor is made of a traditional metal-oxide semiconductor field effect transistor (MOSFET) with two additional capacitors connected to the gate of the device. This creates an electrically floating node on the gate of the device. Traditional MOSFETs are 4 terminal devices with the gate, drain, source, and bulk/well. FGs add an additional terminal often denoted as the tunneling junction or Vtun. A comparison between a standard MOSFET and a FG are shown in Fig. 2.3. FGs are used in the ramp as analog non-volatile memory that can be changed using two quantum phenomena. These are Fowler Nordheim tunneling and hot-electron injection. Using these phenomena, we can adjust the charge on the gate of the device ultimately creating a programmable current source.

Fowler Nordheim tunneling is used to decrease the number of electrons on the gate of the device ultimately decreasing the current through the channel. This is done by applying a large voltage to the tunneling junction while maintaining a constant source to drain voltage.



Figure 2.3: (a) Standard P-channel MOSFET. (b) A FG Transistor made from a P-Channel MOSFET.

This creates a large electric field allowing electrons to tunnel through the oxide of the MOS cap [16]. Hot-electron injection is used to increase the number of electrons on the gate ultimately increasing the current through the device. This is done by creating a sufficiently large source to drain voltage. This allows electrons to impact ionize with holes and increase in energy level. Some electrons obtain a high enough energy level to travel through the gate oxide onto the gate of the device. Fig. 2.4 shows the effect of these phenomena on the IV curve of an FG.



Figure 2.4: IV curve of floating gate transistor

2.3.2 RAMP 1.1 Development Board

I have designed a development board for the RAMP 1.1, shown in Fig. 2.5. This system was designed for the use of the RAMP in a non-laboratory setting. The board includes the RAMP IC, a boost converter for programming the on-chip non-volatile memory, two 2.5V regulators for supply rails to the subsequent analog and digital circuitry, a 1.25V reference, a current reference with a resistor bank for temperature compensation, an operational amplifier, and level shifters.



Figure 2.5: Ramp 1.1 Development Board

The circuit board itself includes headers for a daughter board for an external CPU, onchip wheat stone bridge, connections for external sensors, and to measure signals passing through the system. This design allows for any MCU board such as an Arduino or Particle Xenon to be connected to our RAMP development board. The external MCU board controls the reprogramming the RAMP and enabling sensors. The use of an external MCU board allows users to reprogram the RAMP in a design environment of their choosing. The development board can be powered by a micro USB cable or a battery pack that is kept below the development board. With the ramp powered down and an Arduino Nano in sleep mode the development board has a static current draw 20 μA .

Compared to the RAMP 1.0 development board presented in [2], the RAMP 1.1 developmental board uses a different microprocessor due to the PanStamp no longer in production. While this led the design to allow the user to choose which MCU they would like to develop with this can inherently lead to a higher power consumption due to the user's choice of MCU. Ultimately this allows the user to choose the best MCU for their specific application.

2.4 Improvements to Future RAMPs

One of the main goals of this research is to aid in the improvement of future RAMPS. One major improvement that can be made is to add an on-chip analog-to-digital converter. This would allow the ramp to keep the power-hungry digital circuitry in a low-power mode for longer periods of time. This would also allow the specifications for a desired CPU to be lowered and possibly open the doors to even lower power options in the future. Chapter 3 will present an Asynchronous Extrema Sampling ADC that will be placed on future versions of the RAMP.

Another goal of this work is to increase the versatility to future RAMPs. This could be done be allowing the RAMP to handle all signals that fall within the supply rails. The RAMP currently lacks at buffering signals with rail-to-rail capability. In chapter 4 a rail-torail operational amplifier has been designed. This amplifier was designed to improve future RAMPs ability to buffer signals and provide rail-to-rail operation where applicable.

2.5 Chapter Summary

Low-power reconfigurable analog circuits are expressed inside of an FPAA. These devices are easier to program than typical analog circuitry due to an FPGA like structure that allows for the use of analog building blocks called CABs. These CABs can be connecting in any configuration that the designer desires via a switch matrix. This allows for an easier and shorter design process for those using analog circuitry. Some najor advantages to reconfigurable analog circuitry include the reduction in power compared to an all-digital system, and reconfigurability post fabrication for optimization.

In this chapter a portable platform has been presented for the RAMP 1.1. This allows for the RAMP to be used in the field for real-world applications. The RAMP development board was tested with an Arduino Nano and produced a quiescent current draw of 20μ A. The design of this system allows the user to choose their own MCU for development and can be optimized for different applications. The remaining goals of this work are to improve the performance and versatility of the RAMP. The forthcoming chapter will present an AES-ADC that will help increase the performance of the RAMP.

Chapter 3

Asynchronous Extrema Sampling Analog-to-Digital Converter

In the previous chapter we discussed a Reconfigurable Analog/Mixed-Signal Platform known as the RAMP. The RAMP 1.1 can be used in the field with the design of the portable platform presented in chapter 2. While this was a big step in the correct direction, the RAMP 1.1 can still be improved upon. An interface between the outside world in the RAMP still does not exist on-chip. An on-chip analog-to-digital converter (ADC) would allow the RAMP to interface with the outside world without powering the MCU. This would decrease the energy usage of the system by keeping the power-hungry digital circuitry in a low-power state for longer periods of time. A low-power ADC would be ideal to minimize energy consumption when converting signals. This can be done via asynchronous sampling, specifically sampling at local extrema (i.e. maxima or minima). Extrema sampling allows for the number of samples taken to be reduced by at least 50% ultimately consuming less energy due to the reduced number of converted samples.

In this chapter, an asynchronous extrema sampling analog-to-digital converter (AES-ADC) will be presented along with how it aids in the power savings of an overall system. This design is based upon the successive approximation analog-to-digital converter (SA-ADC) presented in [1, 17]. The design itself uses a double tail comparator [18], a successive-approximation register, and a 10-bit digital-to-analog converter (DAC). The AES-ADC has been fabricated in a 0.5μ m process, due to the working remotely the circuit itself was not able to be tested for this thesis. The simulation results were produced via LTspice for proof of concept.

3.1 Synchronous Vs Asynchronous Conversions

Synchronous operation is when all parts of the system are clocked at the same time based upon an externally provided clock signal. Most ADCs utilize synchronous conversions. When using synchronous conversions, the sampling rate must meet or exceed the Nyquist-Shannon sampling theorem [19]. This theorem states that to accurately reconstruct a signal it must be sample at minimum of twice the highest frequency component of the signal. For example, let us say that we are sampling a signal that has a base function of a 1kHz sine wave, but has a "bursty" section that has a frequency of 1MHz. We must sample this entire signal at a sampling frequency of 2MHz to reconstruct the signal accurately. While this may not seem like a large problem this ultimately causes many more conversion cycles for the ADC. This causes a large amount of data to be taken and an exponential increase in power consumption of the system. While synchronous sampling does include some major drawbacks to low power systems these can be easily fixed using an asynchronous system.

Asynchronous operation is when components in the system operate on different clock cycles, these clock cycles can be internally of externally provided. Asynchronous signal processing is the ideal approach for processing bursty signals [20]. The AES-ADC utilizes asynchronous operation giving it ideal properties for low-power applications. The AES-ADC genreates a "wake up" pulse to start the conversion process. To generate the wake up signal each system must have its own event detection circuitry. In the case of the AES-ADC this is done by detecting local extrema in the input signal. Using approximation theory along with event detection the signal can be reconstructed with far less sampling enabling the system to increase its energy savings [21]. This will be further discussed in the next section.

3.2 Asynchronous Signal Reconstruction

For this system we are reconstructing a signal that has only been sampled at local extrema (i.e. local maxima and minima). With this method of sampling we can reduce the overall sampling of the signal. Figure 3.1 shows the difference in samples taken using extrema sampling versus sampling at the Nyquist rate. This method of sampling has at minimum 2x reduced sampling points compared to using the Nyquist rate. Figure 3.1 shows a reduction in samples, but this method shows drastic reduction in sampling when signals are bursty or have long periods of no activity.

Jared D. Baker Chapter 3. Asynchronous Extrema Sampling Analog-to-Digital Converter 15



Figure 3.1: Sampling at the Nyquist rate the signal is sampled 20 times. While sampling at local extrema the same signal is only sampled 9 times [1].

The original signal can be reconstructed from the extrema samples using methods for complex lines in computer graphics called Bézier curves [22]. Bézier curves are cubic polynomials that can reconstruct a smooth curve between two extrema points. The cubic formula has four parameters: Two end points (P_0 at x=0 and P_3 at x = 1) and two concavity points (P_1 and P_2) [1]. The Bézier polynomial is given by equation 3.1:

$$B(t) = (1 - x^3)P_0 + 3(1 - x)^2 x P_1 + e(1 - x)x^2 P_2 + x^3 P_3$$
(3.1)

When this formula is used, we can successfully reconstruct the signal for N extrema using N-1 segments. For each segment, equation 1 must be applied to the voltages (V) and the time (T) of the samples individually [1]. Figure 3.2a shows how equation 1 is used to reconstruct the signal between two adjacent extrema. In Equation 1, P_0 and P_3 are represented by the voltage-time pair of each extrema, given by $P_0=(\text{Vmax},\text{Tmax})$ and $P_3=(\text{Vmax},\text{Tmax})$. Concavity points P_1 and P_2 define the smooth transitions between P_0 and P_3 . Equation 1 can be modified into equation 3.2 and equation 3.3 to create a voltage and time vector that can then be used to reconstruct the signal. Figure 3.2b shows the bézier curves to reconstruct our sample signal

$$V(x) = (1-x)^3 V_{max} + 3(1-x)^2 x V_{max} + 3(1-x)x^2 V_{min} + x^3 V_{min}$$
(3.2)

$$T(x) = (1-x)^{3} V T_{max} + \frac{3}{2} (1-x)^{2} x (T_{max} - T_{min}) + \frac{3}{2} (1-x) x^{2} (T_{max} - T_{min}) + x^{3} T_{min}$$
(3.3)



Figure 3.2: (a) Bézier curve reconstruction between two adjacent extrema. (b) Continuous piecewise reconstruction using bézier curves using local extrema [1].

3.3 AES-ADC Architecture

The AES-ADC consist of basic building blocks for analog circuit designers. The circuit itself is broken down into two categories. The first, is an extrema detection system that acts as an analog signal processor. This extrema detection system uses two sample and hold systems, a double tailed comparator, and a memory buffer for event detection. The second, is the SA-ADC that uses a double tailed comparator, a successive approximation register and a 10-bit DAC. The full system can be seen in Fig 3.3. This system has been fabricated in a 0.5μ m process for proof of concept but could easily be synthesized on a FPAA such as

the RAMP. Ideally this would be a pre-built system that was internally connected to allow for ease of use for the designer. Let us now take a deeper look into each component of the AES-ADC and the operation of each subsystem.



Figure 3.3: Diagram of AES-ADC: The Analog Signal Processor contains the extrema sampling system while the Asynchronous ADC contains the SA-ADC based upon the SA-ADC presented in[1].

3.3.1 Extrema Detection

Sample and Hold System

The sample and hold systems are one of two major components to allow for detection of local extrema. The system itself contains 4 operational amplifiers, 4 switches, and 2 capacitors which makes up two sample and hold systems. These systems share an input signal but are clocked at opposite times. This allows for system 1 to be sampling while system 2 is holding and vice versa. The theory behind this is that the outputs of each system will be alternating on which is greater than the other. The schematic of the sample and holds can be seen in Fig 3.4.



Figure 3.4: S&H1 and S&H2 are clocked at opposite times. This allows the outputs of the systems to alternate which contains a higher value.

As of the fabrication of the AES-ADC, the sample and holds were a major bottle neck in the system, specifically the operational amplifiers used. This was due to the operational amplifier used not capable of handling signals ranging from rail-to-rail. The current fabricated design was only able to handle signals from 0V - 2.1V. While this was a limiting factor, the rail-rail operational amplifier presented in Chapter 4 has been used for the simulations of this circuit. Using this amplifier the system was able to achieve full rail-to-rail capabilities. Fig. 3.5 shows the results of the previous system and the system with rail-to-rail capabilities.



Figure 3.5: The improvements from the Rail-to-Rail amplifier designed in chapter 4 are easily seen compared to the current amplifier being used. Adding this amplifier ultimately allows the AES-ADC to be a fully Rail-to-Rail system.

Double Tailed Comparator

The double tail comparator is based on the design presented in [1, 18]. The design used in [1] added complementary comparators allowing the system to have rail-to-rail operation. Depending on the input signal to the comparator either the P-stage or the N-stage is selected. This ensures that an accurate comparison can be made across the entire input range. While accurate comparisons can be achieved from rail-to-rail the comparators were discontinuous and non-linear with one another. This would cause issues when we are dealing with signals near or around mid-rail that are switching between the two different stages of the comparators. Fig. 3.6 shows the non-linearity between the two stages.

The comparator includes four floating gates (FGs). These FGs are used post fabrication to trim the offset off the comparator between the two stages[23, 24]. This is done by shifting the threshold value of the input transistors, allowing the user to tune the circuit properly. This will ultimately provide a linear relationship across the system with proper programming. Fig. 3.7 shows the FG implementation of the N-stage of the comparator.



AES-ADC Output for both nFET and pFET stages

Figure 3.6: Shows the discontinuous non-linearity's that are experienced by a complementary stage rail-to-rail comparator.

Extrema Detection System

The extrema sampling system starts by inputting the analog voltage value to both sample and hold systems. As these systems alternate the output values from sample and hold system 1 (SH1) and sample and hold system 2 (SH2) alternate on which is higher. These outputs are connected to the inputs of the double tail comparator. Specifically, SH1 is connected to the non-inverting terminal and SH2 is connected to the inverting terminal. The alternating



Figure 3.7: Floating gate implementation of N-stage of the comparator used in the SA-ADC.

of SH1 and SH2 create an alternating signal on the output of the comparator. This continues to happen until the outputs of SH1 and SH2 are within the resolution of the comparator. At this point the comparator will not be able to detect which is higher producing several logic "low" outputs in series. I.e. a local extremum has been detected. A memory buffer connected to the output of the double tailed comparator detects these extrema and generates a pulse signal. This pulse signal is ultimately used to trigger the SA-ADC and convert this point of the input wave into the digital domain. Fig. 3.8 shows the full extrema sampling system.



Figure 3.8: A block Diagram of the Extrema Sampling System.

3.3.2 Successive Approximation ADC

Double Tail Comparator

The comparator used in the SA-ADC is the same presented in [1]. It is not to be mistaken for the same comparator used in the extrema detection. The SA-ADC requires its

own comparator is used. This operates in the same principle as the other comparator but differs in the selection of which stage of the comparator. Instead of the comparator using the input signal to decide which stage to use it is based upon the most significant bit (MSB) of the successive approximation register (SAR).

The MSB of the SAR determines if the signal is above or below the mid-rail voltage. If the MSB is a logic high, then the N-stage of the comparator is selected. If the MSB is a logic low, then the P-stage of the comparator is selected. Selecting the appropriate stage based upon the signal is what allows for accurate comparisons across the entire rail-to-rail range.

Successive Approximation Register

The successive approximation register (SAR) presented in [1] was used as the basis for this design. The SAR presented only operates as a 9-bit system. With the addition of two flip flops a 10-bit SAR was realized. Fig. 3.9 shows the SAR. This SAR is the control circuit that allows for the binary search algorithm that is required by a SA-ADC [1]. The SAR contains a shift register that allows for the binary search to be completed. This portion of the SAR passes down a logic high upon the end of each clock cycle. The closed in area in figure 3.9 shows the shift register. At the end of the shift register two additional flip flops are needed to generate a load signal and the Done signal. The load signal loads in the digital code word of the conversion into a memory buffer that allows it to be used in other systems. The Done signal allows for the system to be reset and wait until the system is triggered again.

The second row of flip flops shown in Fig. 3.9 is the SAR itself. When a flip-flop in the shift register is high the corresponding flip-flop in the SAR also goes high. This codeword is then applied to the DAC, converted, and then fed back to the inverting terminal of the comparator. Before the output of the DAC or comparator is updated in the next clock cycle, the previous bit is changed to reflect the output of the comparator[1]. This essentially allows the system to decide if it made the correct guess or if it was wrong and needs to update the value. This continues to happen until the signal passes through the shift register and produces the reset signal for the SAR. This event signifies that the analog-to-digital conversion has been completed.

Jared D. Baker Chapter 3. Asynchronous Extrema Sampling Analog-to-Digital Converter 22



Figure 3.9: 10-bit Successive Approximation Register [1].

10-bit Digital-to-Analog Converter

The SA-ADC utilizes a DAC in a feedback loop to convert the signals from the analog domain to the digital domain. The DAC that is utilized is a parallel charge DAC, shown in Fig. 3.10. This style of DAC is known for use in low power applications and does not require an extra sample and hold system ultimately saving die space. This DAC works by charging parallel capacitors that are binary weighted to be converted. The output of a parallel capacitive charging DAC can be calculated by the equation 3.4, where Q_n is the MSB.

$$V_{out} = V_{dd} \left(\frac{Q_n}{2} + \frac{Q_{n-1}}{2^2} + \dots + \frac{Q_1}{2^{n-1}} + \frac{Q_0}{2^n}\right)$$
(3.4)

Successive Approximation ADC

The SA-ADC uses a binary search algorithm to convert electrical signals from the analog domain into the digital domain. This system utilizes the three major building blocks mention above. Figure 3.11 shows a block diagram of the SA-ADC. The principle of this system is to use the input voltage and the output voltage of the 10-bit DAC as the non-inverting and inverting terminals of the comparator respectfully. The comparator uses these two values and outputs a logic high or logic low depending on the inputs. The SAR then outputs the current bit string into the DAC and the corresponding voltage value is then changed on



Figure 3.10: 10-bit parallel charging DAC. C is chosen by the designer, lower capacitance values produce quicker switching speeds and uses significantly less die space compare to larger values of C.

the inverting terminal of the comparator. This cycle repeats until the conversion has been completed.



Figure 3.11: Block Diagram of SA-ADC.

Simulating Floating Gate Transistors Via LTspice

Due to Covid-19 the simulations for this thesis were completed remotely. With this access to cadence and the normal simulation software typically used for floating gates was not able to be used. This is typically done with the hot-electron injection and Fowler-Nordheim tunneling. In cadence a custom FG model has been designed for use. In LTspice no such model has been utilized. The main purpose of injection and tunneling is to adjust the threshold value of the transistor. Increasing the threshold of the device is the equivalent of injecting the transistor while decreasing the threshold of the device is equivalent to tunneling

the device. In LTspice this can be done by creating a secondary model where the threshold of the device is a variable input. This was done by using the following spice directive: .model FGmodel AKO: ami06p (VTH0 = -x), where x is the desired threshold value.

3.4 Simulation Results

3.4.1 Extrema Detection System

The extreme detection system shown in Fig. 3.8 has been tested using a 100Hz sine wave as the input signal. The system generates a pulse when a local extrema has been detected. This system worked ideally as all local extrema were detected and a pulse signal was generate with each extrema. Fig. 3.12. This system has been tested for the audiable frequency range of 20Hz - 20kHz.



Figure 3.12: Extrema detection of a 100Hz sine wave using the system in Fig. 3.8. Each black pulse denotes that a local extrema has been detected

3.4.2 Successive Approximation ADC

Working with an asynchronous system we cannot complete all the standard test for an ADC. The tests that were focused on help prove the linearity of the ADC. This consisted on reporting the gain error, offset error, differential non-linearity (DNL) and integral non-linearity (INL). These were all tested by simulating a finely stepped ramp signal through the ADC. With the limitations of working from home a resolution of 100μ V was used for this test. First the gain error and offset error was measured from the uncompensated data. The AES-ADC shows a gain error of 0.3LSB or approximately 1mV. The offset error of the ADC differs for the P stage of the system and the N stage of the system. These offsets were -1.9mV and 1.5mV respectfully. The FG transistors that are attached to the input stage can be used to trim off offset error. Figure 3.13 shows the ideal transfer function versus the transfer function of the AES-ADC. In this graph it is very apparent that we have an offset between the two stages of our comparator.



Figure 3.13: Actual Vs Ideal Transfer Function with no FG compensation

Using the attached FGs on the input pairs the offset error can be minimized. The offset error can be changed by applying equal charge on each FG of a single stage of the comparator. Increasing the charge stored on the FG will shift the threshold of the input transistor itself.

The threshold value can be adjusted to higher or lower values using hot-electron injection and Fowler-Nordheim tunneling respectfully. This ultimately allows the offset error to be optimized post fabrication. Figure 3.14 shows the transfer function of the AES-ADC after compensation with the FGs. With optimized offset correction, the gain error of the ADC was 0.3 LSB, total offset error was less than 0.1 LSB, DNL was measured at +- 0.25LSB and the INL was measured at less than +- 0.4LSB. Figure 3.15 shows the INL test results.



Figure 3.14: Actual Vs Ideal Transfer Function with FG compensation

3.5 Chapter Summary

In this chapter we discussed the difference between asynchronous and synchronous sampling, how extrema can be sampled and reconstructed accurately, an asynchronous extrema sampling analog-to-digital converter, and how to simulate FG's in LTspice. Extrema sampling enables the AES-ADC to conserve power by sampling at far lower than the Nyquist rate. The AES-ADC provides a linear 10-bit ADC with proper programming of the floating gate transistors attached to the comparator. This system was fabricated in a 0.5μ m process but was not able to be physically tested due to the covid-19 outbreak. This system was
Jared D. Baker Chapter 3. Asynchronous Extrema Sampling Analog-to-Digital Converter 27



Figure 3.15: Integral Non-Linearity test of the AES-ADC.

simulated via LTspice to obtain results for proof of concept. This system would ideally be placed on the future versions of the RAMP to aid in energy savings.

Chapter 4

Rail-to-Rail Operational Amplifier Design

In the previous chapters we discussed two main problem areas where rail-to-rail operation was needed but not currently available. The first was on our RAMP chip. The RAMP cannot currently buffer out signals that range from rail-to-rail. The second was in the asynchronous extrema sampling ADC. The sample and hold systems are not able to handle input signals near the upper supply rail. This is due to the use of a typical p-channel input pair amplifier. In this chapter we will discuss the design of a fully rail-to-rail operational amplifier to aid in the foretold problem areas. An in-depth look will be taken at the input stage of the amplifier as a simple class-AB output stage was used. Lastly, this operational amplifier was placed in our extrema sampling circuit and the results are shown.

4.1 Rail-to-Rail Input Stage

Rail-to-Rail input stages have been an area of interest for researchers for many years. Today this is a very important parameter as the supply voltages are shrinking. With lower supplies voltages that are approaching the hundreds of millivolts [25], designers can not give up any of the already small input range. Over the years three main styles of rail-to-rail input stages have emerged. These consist of using complementary input pairs, dual n-channel input pairs and dual p-channel input pairs. Each style of input stage has advantages and drawbacks. First let us take a brief look at each style of input stage.

4.1.1 Complementary Input Pairs

Jared D. Baker

A complementary input stage consists of two input pairs. One is made from n-channel devices and the other consist of p-channel devices. Many of these designs are modeled after the design in [26, 27, 28]. A simple complementary input stage can be seen in Fig 4.1.



Figure 4.1: A simple Rail-to-Rail input stage using complementary Input Pairs.

The operating principle behind this structure is based upon that one input pair can operate near a supply rail. For this style of circuit there are three main operating regions. Region 1 contains voltages that are near the lower rail of the circuit. In this region only the p-channel devices are active as the n-channel devices are still in the cut-off region. Region 2 consists of voltages that range from V_{th} of the n-channel devices to V_{DD} - V_{th} of the p-channel devices. In this region both input pairs are active. Region 3 is near the top rail and only the n-channel devices are active. While this allows for rail-to-rail operation there is a major flaw. This is that in region 2 the transconductance of the circuit is doubled. This makes optimal frequency compensation impossible as the circuit would become unstable in certain scenarios.

4.1.2 Dual N-Channel Input Pairs

Dual n-channel input stages have been presented in [29, 30]. Fig. 4.2 shows a dual n-channel input stage that was presented in [30]. This style of design allows for less strict matching that is required by complementary input pairs. To achieve rail-to-rail operation the use of a level shifter is needed to place the input common mode voltage seen by one input pair at a higher value [29]. This design still has issues near mid rail where both input

pairs are on and a compensation network is still necessary. While this design choice does solve some of the issues caused by complementary input pairs it does require a large amount of overhead circuitry comparably.



Figure 4.2: Dual n-channel input pair used for rail-to-rail input stage

4.1.3 Dual P-Channel Input Pairs

The same technique can be applied to p-channel devices as the n-channel devices as presented in [31]. This design uses a sensing circuit to detect which input pair the current should be passed through. This ensures that a constant transconductance is held over the entire input common mode range. The full design is shown in Fig 4.3.

While this amplifier design has overcome the challenges of using complementary input transistors, a complex circuit is needed. The design presented in [31] uses a voltage reference, a comparator, level shifters and feedback circuitry just for the input stage of the amplifier. This ultimately leads to the question of how much added complexity are we willing to use for added benefits of the circuit.

4.2 Rail-to-Rail Output Stage

The output stage of this design is a standard class-AB output stage. This design allows the output stage to achieve rail-to-rail operation that is needed for a large signal-to-noise



Figure 4.3: Dual p-channel input pair rail-to-rail op amp..

ratio. The design of the output stage is noncritical compared to that of the input stage. For this reason, there will be a very short discussion on the output stage.

4.3 Design of Rail-to-Rail Operational Amplifier

The operational amplifier design was based from the typical two stage amplifier. The first stage was created using a complementary input pair giving the amplifier the ability to operate from rail-to-rail. The first stage consists of a folded cascode structure previously used in [26, 27]. The input stage also consists of a transconductance compensation network. The second stage of the amplifier consist of a class-AB output stage that is capable of rail-to-rail output voltages. These two stages together allow for a fully functional rail-to-rail amplifier. This amplifier was implemented in a 0.5 μm process and simulated in LTspice.

4.3.1 Input Stage Design

The first stage of this design is based upon a folded cascode topology. The focus of this design is the input pair themselves. The input pairs are made of n-channel and p-channel devices. Using a complementary input stage, we can achieve a rail to rail operation. Fig. 4.4 shows a simple rail-to-rail input stage using complementary input pairs. This design allows the n-channel devices to reach the positive supply rail, while the p-channel devices reach the negative supply rail. With this design, the supply voltage must be greater than the minimum supply voltage or a full rail-to-rail operation will not be achievable [26]. Figures

4.4 and 4.5 show scenarios where the supply voltage is met and is not met respectively. The minimum supply voltage can be calculated by equation 4.1:

$$V_{Sup,min} = V_{gsp} + V_{gsn} + 2 * V_{dsat} \tag{4.1}$$



Figure 4.4: Rail-to-Rail input pair when $V_{sup} >= V_{sup,min}$



Figure 4.5: Rail-to-Rail input pair when $V_{sup} \ll V_{sup,min}$

This design has three modes of operation that are presented in [26]. Region 1 is when the input common mode voltages are from $V_{ss} - V_{th,n}$. During this region, the p-channel devices are on while the n-channel are in cut off. Region 2 is defined when the input common mode voltages are from $V_{th,n} - (V_{dd} - |V_{th,p}|)$. During these times both input pairs are active. Region 3 is from $(V_{dd}-|V_{th,p}|) - V_{dd}$. In the final region only the n-channel devices are active while the p-channel devices are in cut off. While this design is very simple and easy to follow, there is a major design flaw. While the circuit is biased in region 2 the total transconductance

Jared D. Baker

through the input stage is essentially doubled, shown in Fig 4.6. The transconductance can be calculated for each part of the input common mode range given by equation 4.2 [26].

$$gm = \sqrt{kI_{ref}}, \text{ where } k = \mu_p C_{ox} \left(\frac{W}{L}\right)_p = \mu_n C_{ox} \left(\frac{W}{L}\right)_n$$

$$(4.2)$$

With proper sizing of the input devices we can ensure that the n-channel and p-channel devices have the exact same transconductance.



Figure 4.6: Transconductance issue with simple complementary input stage.

One of the major goals in this design is to have a constant transconductance. A simple sensing circuit can be added to reduce the varying transconductance significantly. This sensing circuit is shown in Fig 4.7, transistors M5-M7 make the sensing circuit. A bias voltage is placed on the gate of M5, making the device act as a common mode sensor. The common mode voltage of M1 and M2 is compared to the bias voltage placed on the gate of M5. If the common mode voltage of M1 and M2 is less than the bias voltage, all current passes through the p-channel input pair. When the common mode voltage of M1 and M2 is greater than the bias voltage, the current passes through transistor M5 and mirrored into the n-channel input pair through M6 and M7. While this method does help decrease the varying transconductance, there is still a large amount of variation.

Comparing this to the simple input stage shown in Fig 4.4, we see a reduction from 100% variation to 35% variation. Similar results were seen with other current switching



Figure 4.7: Transconductance issue with simple complementary input stage.

techniques as presented in [32]. While this is significant other current switching techniques are more efficient in limiting the variation of transconductance. To help reduce the varying transconductance even further a novel compensation network has been designed.

4.3.2 Transconductance Compensation

To further reduce the variation of transconductance across the input common mode range, a novel compensation network has been designed. This compensation network adds 9 total transistors to the design of the input stage. The compensation network is shown in Fig. 4.8. The circuit works on the same principle as the main input pairs. While the common mode voltage of M1 and M2 is lower than the bias voltage applied to M7 the current passes through M3 and is mirrored into M4. But at this point the transistor M5 is in cut-off, so no current flows through the branch of M4 and M5. Similarly, when all the current is passing through M7 it is then mirrored through M6 and into M5, but M4 is in cutoff and no current flows through the branch. The only scenario where the input stage is affected is when current is mirrored into M4 and M5 simultaneously.

This design allows us to add or subtract current during the transition region between the n-channel and p-channel devices. In this specific case we need to subtract current in this region. Figure 4.9 shows how we can connect this compensation network to add or subtract current during our transition time. If we wish to add current during the transition



Figure 4.8: Transconductace compensation circuit

region the compensation network can be connected to node "Add". This will pull extra current through the n-channel devices while in transition to being turned on. In this case the compensation network will be connected to node "Sub", this will draw some current away from the n-channel pair in the transition region. With the addition of this network, we can reduce the transconductance variation by an additional 15% making our total variation 20%. While this is typical for current switching techniques [33], we would like to achieve a lower variation to make it comparable to other methods.

To further lower the variation of the transconductance, a second compensation circuit was added. This compensation network is the same as the first in all aspects except one. The bias voltage on the second network is slightly larger than that of the first. This allows for a further minimization of variance in the transconductance in the transition region. Figure 4.10 displays the transconductance curves of the three networks. With the addition of the second compensation network the variance in transconductance was reduced to 6.3%. This



Figure 4.9: Locations on where the compensation circuit can be attached to either add or subtract current during the transition region.

puts it in the same category as several other amplifier designs [29, 30, 31, 33].



Figure 4.10: Compares the three different styles of compensation discussed in this chapter

4.3.3 Input Stage Overview

The overall input stage is largely based from a folded cascode design presented in [26]. This allows for a fully rail-to-rail input stage. The folded cascode design in [26] suffered from a large amount of variation in transconductance across the input common mode range. In Fig. 4.7, a current switch was added to allow for an overlapping transition region of the input devices. While this reduced the transconductance variation by 65% over a simple design it was still not ideal. A novel transconductance compensation branch was then designed. With the use of two of these compensation networks the variation in transconductance was reduced to 6.3% over the input common mode range. The full input stage schematic can be seen in Fig. 4.11.



Figure 4.11: Input stage of rail-to-rail amplifier ignoring biasing circuitry.

4.3.4 Output Stage

The output stage for this design uses a class-AB architecture to achieve rail-to-rail operation. This is very important when considering the SNR of the amplifier. If the output stage is not capable of a rail-to-rail output swing the SNR will be negatively affected. The output stage is based on an elementary push-pull common source amplifier. The output stage itself is shown in Fig. 4.12. It should be noted that miller compensation along with a lead resistor is also included in the output stage. This is shown in the full schematic in Fig. 4.13

With the combination of the input and output stage a rail-to-rail operational amplifier has been realized. The outputs of the cascode branch are used to bias the common source amplifier in the output stage. The compensation technique was chosen following the technique described in [34], and then adjusted to achieve a slightly higher phase margin. The full schematic excluding bias circuitry is shown in Fig 4.13.



Figure 4.12: Class AB output stage.



Figure 4.13: Full Schematic of designed operational amplifier. Excludes biasing circuitry.

4.4 Simulation Results

4.4.1 DC Parameters

One of the major focuses of this design is to have a constant transconductance. This allows for other amplifier characteristics to also be held constant [29]. Fig. 4.14 shows a plot of transconductance versus the input common mode voltage of the circuit. It can be seen here that we have 6.3% deviation. This is greatly reduced compared to older techniques [26] and comparable to newer techniques as well [29]. Table 4.1 shows all other DC parameters that were measured in this circuit.

Parameter	Measurement	
Input Common Mode Range	V_{ss} - V_{dd}	
Gain (dB)	90.1	
Power Consumption (mW)	0.68	

Table 4.1: DC Parameters of Operational Amplifier



Figure 4.14: Total transconductance of the operational amplifier across the input common mode range.

4.4.2 Frequency and Transient Response

The two major AC parameters for operational amplifiers are unity gain band width, and phase margin. These factors are important in the stability of the circuit. A phase margin of at least 60° was sought out to ensure stability during the expected operating conditions. The phase margin of the circuit was measured at 61° with a unity gain frequency of 5MHz. Fig. 4.15 gives the magnitude and frequency response of this circuit.

The transient response in the amplifier is critical, due to it being used as a voltage buffer. The total harmonic distortion of the output was calculated when inputting a 1kHz 3.3V p-p sine wave into the system. The measured total harmonic distortion of this circuit is 0.087%. Fig 4.16 shows the input and output signals from this test.

4.4.3 Comparison With Other Amplifiers

The amplifiers simulation results are compared to other amplifier designs over time. This table represents how this amplifier stands to others that have been previously designed. The results show that comparable results were achieved with this simple design while including



Figure 4.15: Magnitude and Phase response of the operational amplifier.

a relatively low overhead compared to other designs. Table 4.2 shows the results from this amplifier and others that were previously designed.

Specification	[26]	[29]	[35]	This Work
Supply Voltage	3	2.2	1	3.3
Process (μm)	1	0.5	0.35	0.5
gm Variation (%)	17	6	3.6	6.3
Power Consumption (mW)	0.54	0.6	0.015	0.68
Gain (dB)	85	111	90	90.1
Phase Margin (°)	66	59	-	61
GBW (MHz)	2.6	15	0.278	5
Capacitive Load (pF)	-	10	$55 \parallel 1M\Omega$	$30 \mid\mid 1M\Omega$

Table 4.2: Comparison of Operational Amplifier

4.5 Use of Designed Amplifier in Extrema Sampling Circuit

One of the main motivations behind this design was for use in the extrema sampling circuit in the AES-ADC. The current circuit presented in chapter 3 used a standard p-



Figure 4.16: Transient response of 1kHz sine wave

channel input pair amplifier that was unable to operate near the top rail. This ultimately forced us to attenuate our signals before passing them into the extrema detection system. The same system has been constructed but has been replaced with the amplifier presented in this chapter. The improved system allows a larger dynamic range, ultimately allowing for more accurate data conversions by the system. Observing Fig. 4.17 we can see the improvements made by replacing the previous amplifier with this rail-to-rail amplifier.

4.6 Chapter Summary

Rail-to-Rail operation for amplifiers is a desired characteristic, but comes with several problems. One of the major problems discussed is having a constant transconductance across all input common mode voltages. This has been achieved with a novel compensation network shown in Fig. 4.8. Using this style of network, we were able to achieve a transconductance variation of 6.3% across the ICMR. This is comparable to other amplifiers that are considered "constant transconductance". Lastly, we looked at the use of this amplifier inside of an extrema sampling circuit that was proposed in chapter 3. The extrema sampling system was able to achieve rail-to-rail operation with the use of this amplifier.



Figure 4.17: Displyas the voltage range of the fabricated system and the fabricated system with the use of this amplifier.

Chapter 5

Circuit Design using Free Software

With the Covid-19 outbreak, I was forced to complete the finishing work for this thesis at home. Not having access to my normal laboratory has changed the outcome of this work, but was the ultimate driving force behind the creation of several pieces of software. Working in the traditional laboratory setting typically involves working around very expensive software and/or hardware. While there is not much that can be done about hardware, other software can be used. At WVU we have access to cadence in our circuit design lab, this is essentially the Rolls-Royce of circuit design software. There are many additional features that are taken for granite until they are no longer available. While most students cannot afford a license to cadence, there are free spice simulators. For the remainder of this chapter everything will be about LTspice as that was my software of choice.

LTspice is a free circuit simulation software offered by Linear Technologies. It is a wellknown spice simulator that allows for all major types of simulations and graphical results. LTspice does offer the ability to get all necessary information from a circuit, but it is by no means convenient. LTspice is completely free to use, but there are many short coming's in the program compared to Cadence. While this is expected from a free tool, software can be developed in a third-party program (i.e. MATLAB, Visual Studios, etc.) to help bridge the gap between LTspice and Cadence. There are existing functions to port data from LTspice into MATLAB [36], there are no functions to examine specific circuits or simulations. All software has been developed in MATLAB as most students are able to access this program for free through their university.

In this chapter we will further look into the shortcomings of using free software, how to use third-party software to overcome these shortcomings, how these free tools can help students while working safely from home, a simulation aid for an operational amplifier, a simulation aid for Monte Carlo simulations, and the future of this work.

5.1 Shortcomings of LTspice

LTspice is widely used across the electronic design industry. Many choose to use LTspice as a simulation tool as it has a semi-user-friendly interface, is free, and allows the use of third-party models. Combining all these aspects LTspice can be used to simulate your exact devices that will be fabricated. LTspice does offer all that is needed to simulate complex circuits but lacks in making quick changes and determining the effects of the changes on the circuit quickly. Some of the lacking areas that will be discussed in this chapter include parametric analysis, data visualization, and random number generation. We will also discuss the benefit of applications for student use and the two types of MATLAB applications as well as examples of each.

5.1.1 Parametric Analysis

As of the writing of this thesis, LTspice does not include a parametric analysis function that other circuit design software includes. Parametric analysis is the study of the influence of different geometric and/or physical parameters on the outcome of the circuit. This analysis is very useful, particularly to analog design engineers. Adjusting the ratio of the width and length of any device in a circuit can greatly impact the results. In the operational amplifier simulation aid, a parametric analysis function was created and used. This allowed the user to complete parametric analysis on their circuits while still using LTspice. The function uses the existing LTspice netlist and rewrites it for every iteration that the user would like to test. While this is not necessary to complete the parametric style analysis it greatly reduces the amount of time taken to achieve the same results.

5.1.2 Data Visualization

LTspice has an interactive graphing window that users can select nodes of interest to either plot voltages or currents. While this window is very convenient for simulations with less than 5 runs, it is very difficult to handle large amounts of data. When running tens or hundreds of iterations of a simulation the graphing window becomes so cluttered that you cannot see which data corresponds to which simulation. LTspice has combated this by allowing the user to choose which runs to plot on the graph. While this cleans up the graph space, it still requires the user to first find this option in LTspice, and then sort through the list of all the data that was taken only to choose a few simulations to be shown. To combat this a graphing function has been created in MATLAB. After porting the simulation data into MATLAB, it can easily be post-processed and visualized.

When designing any circuit, it is important to understand which region of operation your device is operating in. Expensive software such as Cadence offers an in-application tool that displays the region of operation on the schematic after simulation, along with other common values such as V_{ds} and V_{gs} of the device. LTspice does not offer anything of this nature. LTspice requires the user to find these values by calculating the difference and manually determining the region of operation. MATLAB has been used to simulate large circuits and determine the region of operation. The data can easily be parsed through to determine the region of operation.

5.1.3 Random Number Generation

There are many scenarios when a user may need to generate a random variable to input to a circuit. Currently LTspice has four functions that allow for creation of a "random variable", these are rand, random, white, and flat. While they claim to be random numbers, they truly are pseudo-random numbers. The seed inside the function needs to be changed to create a different value. While this could manually be adjusted to get a new random number set for every simulation, it cannot be automated inside of LTspice. To complete any scenario that requires a truly random number you must use a third-party tool. For example, MATLAB has a function named rand, this function generates a random number from run to run. Using MATLAB, a random variable can be generated and used for simulation in LTspice.

5.2 Application Development for Student Use

There are two major ways that students could develop applications in MATLAB. The first, is the easier but messier solution, creating a zip folder with MATLAB script files that run in a specific order to complete a task. The second, is to create an application in MATLAB. This does take more work but ultimately produces a custom interface for the application that is significantly easier to use compared to several MATLAB scripts. The creation of a custom interface is much easier for a user to pick up and use, but offers no adjustability to the user. If an issue arises, they will need to contact the developer to report the issue. Later in this chapter, we will discuss two applications where one was created with a zip folder and the other with a custom interface.

The creation of an application has the potential to be very useful to students taking circuit design classes or completing research. The creation of these applications allows students to play with different styles of circuit simulation or expand their knowledge on different circuits without having to fully derive them mathematically. Applications that have been designed could be placed in a repository. This would allow students to go back and download other applications made from past students. Students would then spend more time working on the assignment at hand instead of coding a nice interface to work with. It would not be realistic for a repository of everything needed but this would allow students to see that any custom application could be created and uploaded to a central repository for future use.

MATLAB allows for development of applications to be standalone or web-based apps. Standalone applications are rather big in size (>1GB) depending on how sophisticated the program is. But a standalone application does not require the internet to run so students who did not have access at home would still be able to further their work. For students who have access to internet, they would be able to use the web-based app. These typically are significantly smaller than the standalone applications. Once the application is opened it can be used like other applications on their computer. Ideally each program downloaded would be required to have a help or read me text file to explain the application, how it is intended to be used and an example problem. This would help ensure that the student understood how to use the application itself.

5.3 Operational Amplifier Simulation Aid (Zip Folder)

After using an op-amp simulation tutorial as practice, I quickly realized that the design of a more complicated circuit would be a challenge solely using LTspice. This led me to create an application to aid in the simulation process. This application contains 27 different functions and 1 main script. This style of application is more of an "open hood" model as the user is easily able to change and create functions to use but lacks a pleasant appearance. Most of the functions that have been created are general purpose, meaning that they could be used on any type of circuit. One example of this is the region of operations function. This function uses the voltages at the drain, gate, and source of each device. This then outputs an array of numbers that correspond to the region of operation for that device. Other general functions are used to adjust several netlist parameters when simulating a circuit.

Jared D. Baker

The simulation tool contains several advantages over the traditional simulation process of just using LTspice. The major advantage is that the results are calculated and then easily displayed on the screen. If a function is benefited by a graph it can be enabled from the inputs of the function itself. This also allows us to easily change multiple transistor width or lengths by entering the needed values into an array-based format. This can be done for single or multiple devices at a time. Lastly, this method allows for parametric analysis of single devices in a design.

5.3.1 Using the Operational Amplifier Simulation Aid

This simulation aid requires the use of LTspice IV and MATLAB. All results shown in this section are those produced from the op-amp designed in chapter 4. First the user will place the design into LTspice and place the operational amplifier in an open-loop configuration. If this is done, the user will be able to complete all DC simulations via this schematic. Next, the user will be required to put in several inputs into MATLAB. These inputs consist of 6 file paths and 3 labels that were placed on both input terminals and the output terminal of the device. After these have been entered the DC simulation can be completed.

The simulation utilized the system command in MATLAB that allows us to run LTspice via the command line. It first simulates the open-loop circuit and then the closed-loop circuit. When this is complete all DC parameters are calculated. Lastly, the simulator brings a dialog box on the screen containing the DC parameters of the operational amplifier. Fig. 5.1 and Fig. 5.2 display the outputs of the DC simulation.



Figure 5.1: Example of DC simulation parameter's.



Figure 5.2: Example of Normalized Transconductance output.

After completion of the DC analysis, the user may run a transient analysis on their design. The utilizes the buffer configuration that was previously configured in the DC simulation. The transient analysis function takes inputs of a sine wave and allows the user to plot the input and output of the system. This function also calculates a rough estimate of the total harmonic distortion that is present in the system. This calculation only uses the first 5 harmonics in the system as the third and fourth harmonics are the two dominant harmonics. Figure 5.3 displays an example of the result when inputting a 1kHz sine wave into this

function.



Figure 5.3: Example of Transient Analysis.

To complete the AC analysis, we must edit a few items in LTspice. We first must place our offset voltage source in the design, followed by changing the simulation type to AC. Lastly, we will have to change the input voltage source to have an AC amplitude of 1 to allow us to easily view our transfer function. The simulation will be run in LTspice and then the graphical data needs to be manually exported into MATLAB. There is currently no way to automatically export the data from LTspice into MATLAB. Once the data is in MATLAB the user can then run the AC analysis portion of the simulation aid. This portion calculates the phase margin along with displaying a bode plot of the information shown in figure 5.4.

5.4 Monte Carlo Simulations (Custom Interface)

5.4.1 Creating a Cutsom Application Via MATLAB

The Monte Carlo simulation application was developed using MATLAB's App Designer. App Designer allows for the creation of a custom interface designed by the user. Each



Figure 5.4: Example of Bode Plot Output from AC analysis.

application designed would have its own interface, if applicable an application can contain multiple tabs for different types of simulations. App Designer allows for the insertion of push buttons, text boxes, numerical boxes, drop down list, switches, etc. This allows users to create any custom interface by physically placing components in a window compared to creating a button with code. Each addition to the custom interface can then be given a call back function for when it is selected. Programming the callback functions is identical to creating a script in MATLAB. The code to execute the created functions is automatically created by App Designer.

5.4.2 Monte Carlo Sumulation Application

Monte Carlo simulations uses uncertainty of device fabrication to project the possible performance of the fabricated circuit. This style of simulation is very useful when trying to obtain real world results as circuits are rarely fabricated at the designed values. LTspice does currently provide a Monte Carlo function, but it is not truly random. This is flawed due to the pseudo-random number generation that is produced via LTspice.

To combat this issue, an application has been developed to allow designers to model

Chapter 5. Circuit Design using Free Software

Jared D. Baker

transistors or circuits while using Monte Carlo simulations. This application allows for the user to complete truly random simulations without knowing much about how a Monte Carlo simulation works. This is completed by first generating a random variable in MATLAB according to the distribution the user has selected. The existing netlist is then updated with a new value for the selected parameter and simulated via LTspice. After the simulation, the data is brought back into MATALB, sorted and saved into an array. This is then repeated for the desired number of iterations specified by the user. The application can be seen in Fig. 5.5 All data produced by the application can be saved into .mat files for further processing in MATLAB.



Figure 5.5: User Interface when Using Monte Carlo Simulations with results plotted from adjusting threshold values of a digital inverter.

Functions from the operational amplifier application were repurposed for the use in this application. The repurposed functions edited existing netlist, parsing data from LTspice into

MATLAB and formatting data once inside MATLAB. Ideally functions that user create can be repurposed and used by other students in their work.

5.4.3 Using Monte Carlo Simulation Application

Similarly, to the operational amplifier simulation aid we first must start with an LTspice netlist, let us consider a digital inverter for this example. After this we can open the Monte Carlo Application shown in Fig. 5.something. Once in the application the user should first click the help button for a detailed explanation on how the application works. The help file will walk the user step by step through the application. The help text file is shown in the appendix of this work. After successfully entering all the proper information the plot button can be pressed. An example output is shown in Fig. 5.5.

5.5 Summary and Future Work

The work presented in this chapter just scratches the surface of two possible applications that could be developed for students to use. This serves as a building block for other to continue upon this idea to increase the usability of free software. This work proves that circuit design in free software can be aided from custom applications. Future students could take this idea and expand upon its possibilities. With a dedicated graduate student this work could be expanded upon for several different applications ranging from class projects, class laboratories, or even personal research. The continuation of developmental tools will hopefully increase student's awareness to different types of circuits and circuit simulations.

In this chapter we started by discussing the change that the covid-19 outbreak has brought upon us and how it has affected circuit design students. The drastic change from a laboratory environment to working at home has brought new challenges in terms of software available. LTspice was chosen as a building software for applications to use while working remotely as it is a free tool. Although LTspice is free there are downfalls with the software. Thirdparty applications can be developed in MATLAB to help eliminate some of the nuances that LTspice contains such as no parametric analysis, graphing problems, and no true statistical analysis.

Applications for operational amplifier simulations and Monte Carlo simulations have been developed and tested. These applications allow users to complete simulations without using the LTspice interface. All processing can be completed in MATLAB or in the custom interface provided. Ultimately these applications would benefit students after they have been created. Students could use them for class projects, labs, or even to just further their knowledge of different types of circuits.

Chapter 6

Summary and Conclusions

This work has presented a portable platform for the RAMP 1.1, a low-power data converter by implementing asynchronous extrema sampling, and a rail-to-rail amplifier to be used as an elementary building block for future circuits. The AES-ADC described in chapter 3, was a great proof of concept for a low power data converter to be implemented on future RAMPs. The fabrication of this system on a custom IC will allow for the transfer onto the RAMP without worry of the systems performance. The inclusion of this system on future RAMPs would allow to keep power-hungry digital system in a low-power state for longer periods of time, ultimately decreasing the total power consumption of the system.

In chapter 4, a rail-to-rail operational amplifier was designed. The purpose for this design was driven to help improve the RAMP to buffer signals with rail-to-rail capabilities. The design presented has achieved a stable rail-to-rail amplifier, that can easily be placed onto future RAMP chips in the future. The amplifier was also used to improve the performance of the sample and hold systems inside the AES-ADC. With the inclusion of this amplifier, the AES-ADC was increased its dynamic range by 27% over the system that was fabricated. This amplifier still needs to be fabricated and tested before placed upon future RAMPs. After testing, the use of this amplifier on a RAMP would increase the overall versatility of the system and alleviate the need for off-chip voltage buffers.

The covid-19 pandemic has forced circuit designers to adapt to a new design environment. For some this means transitioning from a lab with top quality software and precision equipment, while others are completing class labs at home. While transitioning from Cadence into LTspice many issues were discovered when simulating circuits. Applications to simulate single transistors, simple circuits, and complex circuits have been developed via MATLAB to aid in the simulation process while working from home. These applications can be used by future students with minimal knowledge of coding to simulate their circuits, allowing them to continue their work safely from home. In the future more applications could be designed and stored in a repository for students to use while away from the laboratory.

References

- B. Kelly, A. DiLello, and D. Graham, "Reconfigurable analog preprocessing for efficien asynchronous analog-to-digital conversion," *Low Power Electronics and Applications*, vol. 9, no. 3, Aug 2019.
- [2] B. Rumberg, D. Graham, S. Clites, B. Kelly, M. Navidi, A. Dilello, and V. Kulathumani, "RAMP: Accelerating wireless sensor hardware design with a reconfigurable analog/mixed-signal platform," in *Proceedings of the ACM/IEEE Conference on Information Processing in Sensor Networks*, Apr 2015, pp. 47–58.
- [3] L. Brock J., 'Introduction to Logic Circuits & Logic Design with Verilog. New York, NY: Springer International, 2017.
- [4] I. Kuon and J. Rose, "Measuring the gap between fpgas and asics," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 2, pp. 203–215, Feb 2007.
- [5] E. K. Lee and P. G. Gulak, "A cmos field programmable analog array," Solid State Physics, vol. 26, no. 12, pp. 1860–1867, Dec 1991.
- [6] C. M. Twigg, P. Hasler, and D. V. Anderson, "Large-scale fpaa devices for signal processing applications," in 2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07, vol. 2, 2007, pp. II-69–II-72.
- [7] S. Bhattacharyya, S. Andryzick, and D. Graham, "An acoustic vehicle detector and classifier using a reconfigurable analog/mixed-signal platform," *Journal of Low Power Electronics and Applications*, vol. 10, no. 1, p. 22, Feb 2020.
- [8] J. Hasler, "Large-scale field-programmable analog arrays," *Proceedings of the IEEE*, vol. 108, no. 8, pp. 1283–1302, 2020.
- [9] A. Basu, C. M. Twigg, S. Brink, P. Hasler, C. Petre, S. Ramakrishnan, S. Koziol, and C. Schlottmann, "Rasp 2.8: A new generation of floating-gate based field programmable analog array," in 2008 IEEE Custom Integrated Circuits Conference, 2008, pp. 213–216.
- [10] A. Basu, S. Brink, C. Schlottmann, S. Ramakrishnan, C. Petre, S. Koziol, F. Baskaya, C. M. Twigg, and P. Hasler, "A floating-gate-based field-programmable analog array," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 9, pp. 1781–1794, Sept. 2010.

- [11] V. Srinivasan, G. J. Serrano, J. Gray, and P. Hasler, "A precision cmos amplifier using floating-gate transistors for offset cancellation," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 2, pp. 280–291, 2007.
- [12] S. Shah and J. Hasler, "Low power speech detector on a fpaa," in 2017 IEEE International Symposium on Circuits and Systems (ISCAS), 2017, pp. 1–4.
- [13] B. Rumberg, D. Graham, V. Kulathumani, and R. Fernandez, "Hibernets: Energyefficient sensor networks using analog signal processing," *IEEE Journal of Emerging* and Selected Topics in Circuits and Systems, vol. 1, no. 3, pp. 321–334, Sept. 2011.
- [14] P. Lajevardi, A. P. Chandrakasan, and H.-S. Leee, "Zero-crossing detector based reconfigurable analog system," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 11, pp. 2478 – 2487, Nov 2011.
- [15] E. Vittoz, "Future of analog in the vlsi environment," in *IEEE International Symposium* on Circuits and Systems, May 1990.
- [16] M. Lenzlinger and E. Snow, "Fowler-nordheim tunneling into thermally grown sio2," *IEEE Transactions on Electron Devices*, vol. 15, no. 9, p. 686, Sept. 1968.
- [17] B. Kelly and D.W.Graham, "An asynchrounous add with reconfigurable analog preprocessing," in *IEEE International Symposium on Quality Electronic Design*, 2016, pp. 1062–1065.
- [18] D. Schinkel, E. Mensink, E. Klumperink, E. van Tuijl, and B. Nauta, "A double-tail latch-type voltage sense amplifier with 18ps setup+hold time," in 2007 IEEE International Solid-State Circuits Conference. Digest of Technical Papers, 2007, pp. 314–605.
- [19] A. J. Jerri, "The shannon sampling theorem—its various extensions and applications: A tutorial review," *Proceedings of the IEEE*, vol. 65, no. 11, pp. 1565–1596, 1977.
- [20] A. Can, E. Sejdic, and L. Chaparro, "Asynchronous sampling and reconstruction of sparse signals," in 2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO), 2012, pp. 854–858.
- [21] 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.
- [22] G. Farin, Curves and Surfaces for CAGD A Practical Guide. San Francisco, CA: Morgan Kaufmann Publishers, 2002.
- [23] F. Adil, G. Serrano, and P. Hasler, "Offset removal using floating-gate circuits for mixed-signal systems," in *Southwest Symposium on Mixed-Signal Design*, 2003., 2003, pp. 190–195.
- [24] S. Brink, J. Hasler, and R. Wunderlich, "Adaptive floating-gate circuit enabled largescale fpaa," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 11, pp. 2307–2315, 2014.

- [25] K. Woo and B. Yang, "A 0.25-v rail-to-rail three-stage ota with an enhanced dc gain," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 7, pp. 1179– 1183, 2020.
- [26] R. G. H. E. Ron Hogervorst, John P. Tero and J. H. Huijsing, "A compact powerefficient 3 v cmos rail-to-rail input/output operational amplifier for vlsi cell libraries," *IEEE Journal of Solid-State Circuits*, vol. 29, no. 12, pp. 1505–1513, Dec 1994.
- [27] P. d. J. J. F. R. W. R. Hogervorst, R.J. Wiegerink and J. Huijsing, "Cmos low-voltage operational amplifiers with constant-g/sub m/ rail-rail input stage," in *IEEE International Symposium on Circuits and Systems*, May 1992, pp. 2876–2879.
- [28] S. H. E. Minsheng Wang, Terry L. Mayhugh Jr. and E. Sanchez-Sinencio, "Constant-gm rail-to-rail cmos op-amp input stage with overlapped transisition region," *IEEE Journal* of Solid-State Circuits, vol. 34, no. 2, pp. 148–156, Feb. 1999.
- [29] R. L. Beal, "A low voltage rail-to-rail operational amplifier with constant operation and improved process robustness," Master's thesis, Iowa State University, 2009.
- [30] G. T. J.M. Carrillo, J.F. Duque-Carrillo and J. Ausin, "Constant-gm constant-slewrate high bandwidth low-voltage rail-to-rail cmos input stage for vlsi cell libraries," in *International Symposium on Circuits and Systems*, May 2003, pp. 165–168.
- [31] R. K. Weixun Yan and H. Zimmermann, "A low-voltage low-power fulley differential railto-rail input/output opamp in 65-nm cmos," in *International Symposium on Circuits* and Systems, May 2008, pp. 2274–2277.
- [32] B. Lee and T. Higman, "Extremely simple constant-gm technique for low voltage rail-torail amplifier input stage," in 2011 18th IEEE International Conference on Electronics, Circuits, and Systems, 2011, pp. 314–317.
- [33] T. S. Shouli Yan, Jungyu Hu and E. Sanchez-Sinencio, "Constant-gm techniques for railto-rail cmos amplifier input stages: a comparative study," in *International Symposium* on Circuits and Systems, May 2005, pp. 2571–2574.
- [34] D. Johns and K. Martin, Analog Integrated Circuit Design. Hoboken, NJ: John Wiley and Sons, Inc, 1997.
- [35] Y. Y.-S. Deng Liang, Deng Hong-Hui and Z. Guo-Min, "1-v rail-to-rail constant transconductance operation amplifier," in *International Conference on Electronic Mea*surement and Instruments, Aug 2009, pp. 2–798 – 2–801.
- [36] J. Baker, "Matlab examples at cmosedu.com," http://www.CMOS.edu, July 2020.
- [37] P. R. Kinget, "Scaling analog circuits into deep nanoscale cmos: Obstacles and ways to overcome them," in 2015 IEEE Custom Integrated Circuits Conference (CICC), 2015, pp. 1–8.

- [38] T. Insturments, "Msp403f15x, msp430f16x, msp430f161x mixed signal controller," Texas Instruments: Dallas TX, USA, 2002.
- [39] B. Rumberg and D. Graham, "A low-power magnitude detector for analysis of transientrich signals," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 3, pp. 676–685, March 2012.
- [40] A. Golberger, L. Amarall, L. Glass, J. Hausdorff, P. Ivanoc, R. Mark, J. Mietus, G. Moody, C. Peng, and H. Stanley, "Components of a new research resource for complex physiological signals," *PhysioBank, PhysioToolkit, and PhysioNet*, vol. 101, no. 23, pp. 215–220, 2000.

Appendix A

Appendix

A.1 Code from Operational Amplifier Simulation Aid

A.1.1 Main Script

```
%% User Inputs Needed:
tic();
LTspice_exe_dir = 'Filepath_of_LTspice_Executable';
LTspice_asc_dir = 'Filepath_of_LTspice_.asc_file_for_simulation';
LTspice_raw_dir = 'Filepath_of_LTspice_.raw_file_for_simulation';
OGNetlist_dir = 'Filepath_of_original_LTspice.net_file';
NewNetlist_dir = 'Filepath_of_new_.net_file_created_by_simulation_aid';
NewNetlist_raw_dir = 'Filepath_of_new_.raw_file_created_by_simulation_aid'
PositiveTerminal = 'Node_of_positive_terminal_of_device';
OutputLabel = 'Node_of_output_terminal_of_device_';
Negativeterminal = 'Nod_of_negative_terminal_of_device';
%% Call the System command to generate Netlist
SystemCommand = char(strcat('"', LTspice_exe_dir, '"', " -b -run ", '"', ...
    LTspice_asc_dir , '"'));
system (SystemCommand)
%% Output voltage Range, Input Offset, and Open-Loop Gain
[signame, vRaw, ~] = readLtsRaw(LTspice_raw_dir);
[measurements, StepArray] = OlgDataFormater(signame, vRaw, OGNetlist_dir, ...
```

Jared D. Baker

Appendix

PositiveTerminal , OutputLabel);

```
OutPut_Range = [min(measurements(:,2)) max(measurements(:,2))];
inputoffset = InputOffset(measurements, StepArray,0);
[OLG, OLGdB] = OpenLoopGain(measurements, StepArray);
Transconductance2(signame,vRaw,OGNetlist_dir)
```

```
%% Generate NewNetlist and Call the system command to simulate %the NewNetlist
```

```
RewriteNetlist_Buffer(OGNetlist_dir, NewNetlist_dir, Negativeterminal,...
OutputLabel)
```

```
SystemCommand = char(strcat('"', LTspice_exe_dir, '"', " -b -run ",...
'"', NewNetlist_dir, '"'));
```

```
system(SystemCommand)
```

%% ICMR and Region of Operations

```
[signame, vRaw, ~] = readLtsRaw(NewNetlist_raw_dir);
```

```
[measurements2,TranType,StepArray2] = BufferDataFormater(signame,vRaw,...
NewNetlist_dir,PositiveTerminal,OutputLabel);
```

```
[LJCMR, UJCMR] = ICMR(measurements2, StepArray2, 1);
```

```
RegionArray = OperationRegion(measurements2, StepArray2, TranType);
```

```
%% Results: Text box to display results from DC simulation
string1 = strcat('ICMR:', num2str(LICMR)," ", 'V'," ", 'to', ...
" ", num2str(UICMR)," ", 'V');
```

```
string2 = strcat('OutPut_Range:'," ",num2str(OutPut_Range(1))," ",'V',...
",'to'," ",num2str(OutPut_Range(2))," ",'V');
```

```
string3 = strcat('Open_Loop_Gain:'," ", num2str(OLG)," ", ...
```

```
OR', ", num2str(OLGdB), 'dB');
```

```
string4 = strcat('Input_Offset_Voltage:'," ",...
```

```
char(arrayfun(@char,inputoffset,'uniform',0))," ",'V');
msgbox({string1,string2,string3,string4},'Results')
time = toc();
```

Jared D. Baker

Appendix

%% AC simulation
[F P G] = Parser_TransferPhaseMargin('Schematic.txt');
phasemargin = PhaseMargin(F,P,G);
TransferFunction(F,P,G)

%% Voltage Buffer Test

Transient_Analysis_Sine(NewNetlist_dir, 5,0,20e-6,3.3,1e3,["Vp","Out"])

A.1.2 Data Formater Function

```
function [measurements, StepArray] = OlgDataFormater(signame,vRaw,...
Netlist,InPin,OutPin)
```

%% Converts the character array given from readLtsRaw into a string array, % it then ignores the irrelevant lines. It create a dummy line at the top % and whatever variable is swept is added on the second line. These need % to be removed so when parsing through the data the variable "STRING" % lines up properly with the data in vRaw.

```
for c = 1: length (signame (:, 1))
```

```
string(c,1) = convert Chars To Strings(signame(c,:));
```

end

 \mathbf{end}

%% This section does the following: Opens the netlist, reads the file line % by line. Creates an array for the transistor types, creates an array on % which nodes are connected to each termincal of the device. I will then
63

```
\% use this to load an array of the proper format to fit the functions that
% were already created.
fid = fopen(Netlist, 'rt');
Node = strings(2,2); temp= strings(1,3);
line = fgetl(fid);
count = 1;
while ischar(line)
    line = fgetl(fid);
    if line \tilde{} = -1
        if contains (line, 'l=') && contains (line, 'w=')
             spaces = isspace(line);
             space_location = find(spaces == 1);
             for c = 1: length (space_location)
                 if c == length(space_location)
                     Node(count, c) = line(space_location(c)+1:end);
                 else
                     Node(count, c) = line(space_location(c)+ ...
                          1: space_location (c+1)-1;
                 end
            end
        end
        if contains (line, '.dc')
             space = isspace(line);
             space_location = find(space == 1);
             for c = 2: length (space_location)
                 if c == length(space_location)
                     temp(1, c-1) = line(space_location(c)+1:end);
                 else
                     temp(1, c-1) = line(space_location(c) + \dots
                          1: space_location (c+1)-1;
                 end
            end
```

Appendix

```
StepArray = [str2double(temp(1,1)): str2double(temp(1,3)): \dots
                  \operatorname{str2double}(\operatorname{temp}(1,2))];
        end
    end
    count = count + 1;
end
fid = fclose(fid);
% ensure that all labels are lowercase
for c = 1: length (Node (:, 1))
   for x = 1: length (Node (1, :))
      Node(c, x) = lower(Node(c, x));
   end
end
% appends input value to the end of the measurements array
for x = 1: length (STRING)
    if contains (STRING(x), lower(InPin))
         measurements (:, 1) = vRaw(x, :);
         break
    end
end
% appends the output value to the end of the measurements array
for x = 1: length (STRING)
    if contains(STRING(x), lower(OutPin))
         measurements (:, \mathbf{end}+1) = vRaw(x, :);
         break
    end
end
if exist ('StepArray') ~= 1
disp('Error:_Make_sure_LTspice_schematic_is_in_open-loop_configuration!')
    keyboard
end
```

Jared D. Baker

Appendix

end

A.1.3 Edit Netlist: Change DC Sweep

```
function EditDcSweep(Netlist, NewNetlist, StartValue, StopValue, StepSize)
fid = fopen(Netlist, 'rt');
fidNew = fopen(NewNetlist, 'wt');
line = fgetl(fid);
while ischar(line)
    line = fgetl(fid);
    if line \tilde{} = -1
         if contains (line, '.dc')
            spaces = isspace(line);
            space_location = find(spaces == 1);
            NewLine = strcat (line (1: space_location (2))," ",...
                num2str(StartValue)," ",num2str(StopValue),...
                " ", num2str(StepSize));
            fprintf(fidNew, '%s\n', NewLine);
         else
             fprintf(fidNew, '%s\n', line);
        end
    end
end
fid = fclose(fid);
fid = fclose(fidNew);
end
```

A.2 Example Text File for Custom Application

The file will teach you how to use the Monte Carlo Simulation Via LTspice V1.1 Application. This application was created to allow the user to complete Monte Carlo style simulations with minimal knowledge of how this simulation itself works. The user must complete the following steps in the correct order to properly use this application.

1. Open up Monte Carlo Simulation Via LTspice.

Jared D. Baker

Appendix

2. Click the upper browse button in the upper right corner, beside the LTspice text box. Find the scad3.exe file on your PC and open the file.

3. Click the middle browse button in the upper right corner, beside the Model text box. Find your transistor model file on your PC and open the file.

4. Click the lower browse button in the upper right corner, beside the Netlist text box. Find your netlist file on your PC and open the file.

5. Determine which type of distribution you would like to use for your Monte Carlo Simulation Flat: Generates numbers between -1*tolerance ; x ; 1*tolerance Normal: Generates random numbers from a normal distribution Worst Case: Generates 3 runs, 1st at Vt+Vt*tolerance, 2nd at Vt, and 3rd at Vt-Vt*tolerance

6. Select the type of device that you are completing the Monte Carlo simulation

7. Do not worry about the Simulation Level parameters. This will fill in automatically after the plot button is pushed.

8. Select a Model Parameter that you would like to sweep from the drop down list.

9. Set the number of iterations that you would like the Monte Carlo simulation to run For reference it takes about 5 seconds to run 10 simulations. Keep this in mind if running a large number of iterations. If you choose Worst Case iterations will default to 3. If you select flat or normal it will default to 10.

10. Set the tolerance level between 0 - 100.

11. Click the Plot button and wait for simulation to be completed.

12. If you wish to save the data select File -> Save and follow the dialog box to save your file. The .mat file can then be loaded into a MATLAB script for further processing.

As of Version 1.1 This simulation only completes sweeps of NMOS and PMOS devices as well as input and output voltage of circuits. The FGMOS device does not currently have a working model. This is planned to be updated for a future release.

When creating a 3rd party model for this application please use the following format: It is critical that only one space exists between the numerical and text values for the function to parse your model correctly. Shown below are a few example lines of a third party model.

.MODEL NMOS NMOS (LEVEL = xx +VERSION = xxx TNOM = xxx TOX = xxx ...

Rail-to-Rail Operation in Low-Power Reconfigurable Analog Circuitry

Jared D. Baker

Thesis submitted to the Benjamin M. Statler College of Engineering and Mineral Resources at West Virginia University in partial fulfillment of the requirements for the degree of

> Master of Science in Electrical Engineering

Lane Department of Computer Science and Electrical Engineering

APPROVAL OF THE EXAMINING COMMITTEE

Jeremy Dawson, Ph.D.

Daryl Reynolds, Ph.D.

David W. Graham, Ph.D., Chair

Date