ON-CHIP SIGNAL GENERATION AND RESPONSE

WAVEFORM EXTRACTION FOR ANALOG

BUILT-IN SELF-TEST

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Engineering

By

BRIAN S. POLING
B.S., Wright State University, 2004

2007
Wright State University
WRIGHT STATE UNIVERSITY
SCHOOL OF GRADUATE STUDIES

September 14th, 2007

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY
SUPERVISION BY Brian S. Poling ENTITLED On-Chip Signal Generation and
Response Waveform Extraction for Analog Built-In Self-Test BE ACCEPTED IN
PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
Master of Science in Engineering.

__________________________
Henry Chen, Ph.D.
Thesis Director

__________________________
Fred Garber, Ph.D.
Department Chair

Committee on
Final Examination

__________________________
Henry Chen, Ph.D.

__________________________
Raymond Siferd, Ph.D.

__________________________
Marty Emmert, Ph.D.

__________________________
Joseph F. Thomas, Jr., Ph.D.
Dean, School of Graduate Studies
ABSTRACT


Built-In Self-Test (BIST) is a method of designing and creating an electronic chip or an electronic system that can self test for correct functionality and ensure no manufacturing defects. The reason for analog BIST is the testing of analog parts of analog and mixed-signal ICs is a costly process that traditionally requires the use of expensive high-end automatic test equipment. Due to the nature of the testing and length of the testing process, an efficient analog BIST scheme is in high demand for the ever increasing complexity of analog and mixed-signal circuits. This thesis presents a BIST scheme for generation and response waveform extraction that allows the detection of a faulty circuit design. Along with the detection, an approach to test high speed analog and mixed-signal circuits with test signals upwards of 1GHz is presented. A practical application is to test analog or mixed-signal IC that has a wide bandwidth ADC in its front-end. The BIST scheme includes a method to store the test signal and generate it for the circuit to be tested along with a way to extract the response test signal from multiple test points and allow fault detection. Along with this research, a stepping stone is implemented for analog modeling using MATLAB for accuracy and speed of circuit simulations. The problems associated with the BIST scheme and analog modeling is discussed, along with recommendations.
# TABLE OF CONTENTS

Chapter 1: Introduction ........................................................................................................... 1
1.1 Analog Built-In Self-Test Background ............................................................................ 1
1.2 Analog Circuit Modeling with MATLAB Background ......................................................... 3

Chapter 2: Analog Built-In Self-Test ....................................................................................... 6
2.1 BIST Scheme Goal ............................................................................................................ 6
2.2 BIST Scheme .................................................................................................................... 7
  2.2.1 Digital Data for On-Chip ROM .................................................................................. 7
    2.2.1.1 Ideal ADC in MATLAB .................................................................................... 7
    2.2.1.2 Ideal ADC in VerilogAMS .............................................................................. 10
  2.2.2 8-Bit Digital to Analog Converter ............................................................................. 12
    2.2.2.1 8-Bit DAC Problems ...................................................................................... 14
    2.2.2.2 Pre-Amp Design .............................................................................................. 15
      2.2.2.2.1 Active Load Amp ...................................................................................... 15
      2.2.2.2.2 Simulation Results of Pre-Amp Design ....................................................... 19
      2.2.2.2.3 Problems with Modified Analog Buffer Design ........................................... 20
  2.2.2.3 New Buffer Design ............................................................................................... 20
    2.2.2.3.1 New Analog Buffer Design ........................................................................ 20
    2.2.2.3.2 Bias Circuit Design ..................................................................................... 22
    2.2.2.3.3 Simulation Results of New Buffer Design .................................................... 23
    2.2.2.3.4 Problems with New Buffer Design ............................................................... 27
  2.2.2.4 Final Buffer Design ............................................................................................... 27
    2.2.2.4.1 Final Buffer Amplifier ................................................................................ 28
    2.2.2.4.2 Simulation Results of New Buffer Amplifier .............................................. 29
    2.2.2.4.3 Problem and Design Solution to Final Buffer Amplifier ............................ 32
    2.2.2.4.4 Simulation Results of Final Buffer Design .................................................. 33
  2.2.3 Analog MUX .............................................................................................................. 36
    2.2.3.1 Analog MUX Design ....................................................................................... 37
      2.2.3.1.1 Inverter Design ......................................................................................... 37
      2.2.3.1.2 Analog Switch ......................................................................................... 39
      2.2.3.1.3 Decoder .................................................................................................. 43
      2.2.3.1.4 Final Design of Analog MUX .................................................................. 46
  2.2.4 Analog Response Extractor ....................................................................................... 48
  2.2.5 Digital Response Analyzer ....................................................................................... 50

2.3 BIST and 8-Bit DAC Simulation Results ........................................................................... 50

Chapter 3: Analog Modeling with MATLAB ........................................................................... 60
3.1 BSIM3 Modeling .............................................................................................................. 60
  3.1.1 1-V Model ................................................................................................................. 60
    3.1.1.1 BSIM3 Parameters ......................................................................................... 61
    3.1.1.2 Effective Channel Length and Width .............................................................. 63
    3.1.1.3 Quantity Calculation ..................................................................................... 63
    3.1.1.4 Effective Bulk-Source Voltage ...................................................................... 64
    3.1.1.5 Depletion Thickness of the Gate-Induced Region ........................................ 65
    3.1.1.6 Threshold Voltage Calculation ........................................................................ 65
LIST OF FIGURES

Figure 1: Test Chip Demonstrating Analog BIST Scheme ........................................... 2
Figure 2: Ideal ADC in MATLAB at 125MHz ............................................................... 9
Figure 3: Ideal ADC in MATLAB at 1GHz ................................................................. 9
Figure 4: Ideal ADC in Verilog-AMS at 125MHz ...................................................... 11
Figure 5: Output of DAC at 125MHz ............................................................... 11
Figure 6: 8-Bit DAC ...................................................................................... 13
Figure 7: Frequency Response of Original Analog Buffer ........................................... 14
Figure 8: Schematic of Original Analog Buffer .......................................................... 16
Figure 9: Schematic of Modified Analog Buffer with Pre-Amp ................................ 16
Figure 10: Active Load Pre-Amp ........................................................................... 17
Figure 11: Frequency Response of Active Load Pre-Amp ........................................... 18
Figure 12: Transient Response at 1GHz ................................................................... 18
Figure 13: Frequency Response Modified Buffer with Pre-Amp ................................ 19
Figure 14: Schematic of Amplifier Design ............................................................... 21
Figure 15: Schematic of Bias Circuit ........................................................................ 22
Figure 16: Bias Circuit Simulation ........................................................................... 23
Figure 17: Frequency Response of New Amplifier Design ......................................... 24
Figure 18: DC Response of New Amplifier Design ................................................... 25
Figure 19: 300MHz Transient Response of New Amplifier Design ........................... 25
Figure 20: Power Consumption of New Amplifier Design at 300MHz ..................... 26
Figure 21: 1GHz Transient Response of New Amplifier Design ............................... 26
Figure 22: Power Consumption of New Amplifier Design at 1GHz ........................... 27
Figure 23: Final Buffer Amplifier Configuration ...................................................... 28
Figure 24: DC Response of Final Buffer Amplifier .................................................. 30
Figure 25: Schematic of Final Buffer Design ............................................................ 30
Figure 26: Frequency Response of Final Buffer Amplifier ....................................... 31
Figure 27: Transient Response of Final Buffer Amplifier at 125MHz ....................... 31
Figure 28: Transient Response of Final Buffer Amplifier at 1GHz ........................... 32
Figure 29: Transient Response of Final Buffer Design at 125MHz .......................... 34
Figure 30: Transient Response of Final Buffer Design at 1GHz ............................... 35
Figure 31: Frequency Response of Final Buffer Design ........................................... 35
Figure 32: Power Dissipation of Final Buffer Design .............................................. 38
Figure 33: Schematic of Inverter ............................................................................ 38
Figure 34: Functional Simulation Results of Inverter ................................................ 38
Figure 35: Schematic of CMOS Switch ................................................................. 40
Figure 36: Wn vs. Vin MATLAB Plot ....................................................................... 41
Figure 37: Simulation of CMOS Switch at 300MHz .................................................. 41
Figure 38: Simulation of CMOS Switch at 800MHz .................................................. 42
Figure 39: Simulation of CMOS Switch at 1GHz ...................................................... 43
Figure 40: Schematic of 2 to 4 Decoder .................................................................. 44
Figure 41: Schematic of AND Gate ........................................................................ 45
LIST OF TABLES

Table 1: Original Buffer Design vs. New Modified Buffer Design ........................................ 34
Table 2: Logic Flow of 2 to 4 Decoder .............................................................................. 43
Table 3: Logic Inputs and Output Control of Analog MUX ............................................. 46
Table 4: BSIM3 Parameters ............................................................................................. 61
Table 5: BSIM3 Quantity Calculations ............................................................................. 64
Table 6: UC Berkley Parameters ..................................................................................... 69
Table 7: Lambda and Kn Test Points .............................................................................. 74
Table 8: UC Berkley Simulation vs. Cadence Simulation ................................................. 76
Acknowledgements

I would like to begin by thanking my thesis advisor Professor Henry Chen, for his encouragement and understanding throughout the research process and his technically knowledge on the subject matter. I would also like to thank Professors Ray Siferd and Marty Emmert for taking the time out of their busy schedules to be a part of my thesis defense committee.

I would like to thank my fellow students who helped me along the way in both technical matter and helping in setting up the circuit simulators. Without their help, I wouldn’t be as proficient on the programs as I am now, and I would probable still be stuck on that “dang disc quota error!”

I would like to thank AFRL for sponsoring this research project and for giving me a chance to apply what I've learned into design situations.

I would like to thank my fellow co-workers for being patient and encouraging me and allowing myself to work on some of the research during my spare time at work. Also my thanks’ goes to Jennifer for doing all the scan work on all the figures, and final copy of this thesis.

Finally, I would like thank my family, my wife Stephanie, and my son Scott who showed me to continue non-stop to achieve the goals I was reaching for.

Brian Poling

WSU, September 2007
1.0 Introduction

1.1 Analog Built-In Self-Test Background

Testing the analog parts of mixed-signal designs has been a costly process. Traditionally, this requires high-end expensive automatic test equipment (ATE), and the test time is usually long. The total testing cost can be even higher than the manufacturing cost of the chip itself. In addition, integrating more analog circuits makes the diagnosis of the chip more difficult by increasing the complexity of the circuit. Allowing the increase of the number of off-chip observation nodes may help, but results in higher packaging cost and greater interference to the analog circuits being tested. Consequently, there is a strong demand for a low-cost and efficient analog built-in self-test (BIST) scheme.

BIST is a method of designing and creating an electrical chip, printed circuit board (PCB), or electrical system that can self test for correct functionality and ensures no manufacturing defects. BIST allows for testing of circuits at the circuit under test’s (CUT) operating speeds. This guarantees a more reliable test then using ATE’s, which generally run at slower clock rates then the CUT’s clock rate. The high-end ATE’s can operate at 1 GHz, but the cost of the machine and the test doesn’t allow for its practical use. The use of a BIST scheme to test highly sophisticated analog circuits decreases the cost of manufacturing of those circuits. By not absorbing the cost of purchasing, maintaining, depreciation, and the cost of accessories for the expensive ATE’s allows for the development cost of the circuit to be considerably less. Not only can the BIST
scheme eliminate the use of ATE’s, but Analog Mixed Signal (AMS) testing requires function generators which add a fixed cost to the overall testing procedure. These costs can range from $500,000 to $1,000,000 [1], thus making it impractical to spend that kind of money for one specific test on one specific circuit.

The use of a BIST scheme allows for testing of multiple circuits and is upgradeable for newer technology if designed correctly. This allows for one BIST design to be used on one PCB that might have multiple, complex circuits. With the addition of a few components, the internal nodes can be observed during the testing sequence. This allows for greater understanding of the functionality of the analog circuit in production.

The BIST scheme presented in this thesis uses the combination of an on-chip ROM, Shift Registers, 8-bit DAC, Analog MUX, an Analog Response Extractor (ARE), and a digital Response Analyzer (RA). Figure 1 displays the block diagram of the presented BIST scheme.

![Test Chip Diagram](image)

Figure 1: Test Chip Demonstrating Analog BIST Scheme
The on-chip ROM will store the digitized test signal at the specific frequencies requested. The ROM will then feed the test signal into the bank of shift registers, which will then load the data at the correct clock rate into the 8-bit DAC. The DAC will then output the analog test signal into the CUT. The analog MUX is connected to internal nodes, or observation points, of the CUT that by using control logic can select the desired point of interest. For analog response analysis, the ARE consisting of a 1-bit $\Sigma\Delta$ modulator converts the signal to a 1-bit digitized stream that is outputted to the Digital RA.

The object of research presented with this design is two fold: one, to implement a BIST scheme that can generate upwards of a 1GHz test signal for analog CUT's and two, to implement test signals at multiple frequencies.

1.2 Analog Circuit Modeling with MATLAB Background

Current analog circuit modeling using software is done by various schematic capture programs one of which is used primarily at universities is Cadence. This program, along with the countless other schematic simulation programs, utilize model equations of the transistor to accurately predict the way the transistors will behave when excited by the electrons passing through. Electronic device classes at universities teach how to model transistors by using Level – 1 equations. This is recognized by the faithful transistor equation as seen in chapter 3 of this research thesis.

By using model equations, output characteristics can be found and combined together to generate currents and voltages that can be transferred among the different components used in the simulated design. From time factors along with the voltages and
currents calculated, DC characteristics, AC response waveforms, and transient analysis can be generated for the complete circuit analysis.

Problems are seen when modeling circuits using the various simulation programs. By simulating a simple analog amplifier with few components, Cadence can take just a few seconds to generate results. Simulating complex circuits with hundreds of transistors and Cadence will only take a few hours to simulate. Trying to simulate Very Large Scale Integrated circuits (VLSI) might take days or even weeks. To complicate the problem, when one change is made to one transistor, the results could take another week. More problems with Cadence, and the various other simulation programs, is that if the same change needs to be made to multiple transistors, one has to make that change to each and every transistor individually. If the design being tested is a Pentium 4 microprocessor, classified as a VLSI design, it could take months to make the same change to millions of transistors, and then the simulation of the circuit would need to be completed on top of that. Thus a way to accurately model analog circuits with an efficient simulation time and efficient schematic design management is needed.

A solution to speeding up the simulation time and allowing for an efficient schematic design management is to model analog circuits in MATLAB. By creating text files of the circuit, and then letting MATLAB compute all the model equations, code can be written to allow MATLAB to efficiently process the code. Since text files are being used to design the circuit, making changes to the required transistors increases in speed since all that is needed is to do a "search" for the required transistors and "cut and paste" the specified values that is being changed.
This research was initiated as a way to help speed up both simulation and design time of analog circuits. Presented in this thesis is a stepping-stone to completing the desired goal of analog circuit modeling using MATLAB.
2.0 Analog Built-In Self-Test

2.1 BIST Scheme Goal

There are numerous Built-In Self-Test (BIST) schemes developed for on-chip system testing, yet there is a common problem with each one. That common problem is the ability to perform high speed testing of analog circuits with testing frequencies at 1GHz. The design goal of this research is to develop and construct a BIST scheme that can generate test signals up to 1GHz for high speed testing of an analog to digital converters (ADC’s) and general receiver systems and to perform circuit fault detection. Along with this goal is the ability to do multiple tone frequency generation and allow for the BIST scheme to probe multiple points on the circuit under test (CUT). The method of constructing the BIST scheme consists of multiple components to allow for each performance goal of the circuit being developed in this thesis. Figure 1 (see chapter 1) outlines the block diagram of the BIST scheme and shows the location of each component that needs to be developed to successfully implement the proposed BIST scheme goals.

An on-chip ROM is used to store the digital bit stream of the test signal, an 8-bit bank of Shift Registers will feed the digital signal into the BIST at the correct timing, and an 8-bit digital to analog converter (DAC) will convert the stored signal to an analog test signal. Once the analog test signal is fed into the CUT, a high speed Analog Multiplexer (MUX) is used to perform the ability to probe multiple test points of the CUT. This allows the ability to perform a true self-test on the CUT. The output of the
Analog MUX is then digitized using a 1-bit ΣΔ Modulator as the Analog Response Extractor (ARE). This will convert the tested output signal to a 1-bit digital stream that can then be compared to the stored digital stream of the correct operating circuit. This action is performed in the Digital Response Analyzer (RA). A flag can be generated when the circuit is not working correctly and with the Analog MUX testing specific test points on the CUT, some simple control logic can help perform diagnosis of the CUT.

2.2 BIST Scheme

2.2.1 Digital Data for On-Chip ROM

To begin the self-test, a test signal is generated for an input into the CUT. To generate the test signal for this BIST scheme, an 8-bit DAC and the digital representation of the test signal need to be generated on chip. To generate the digital representation of the analog test signal, an ideal 8-bit ADC can be used. This ideal 8-bit ADC will take in the desired analog test signal and convert it to a digital signal that can then be stored in an on-chip ROM. Once called upon, the stored digital signal will then be supplied into the input of the DAC using an 8-bit shift register to generate the analog test signal for the CUT. Various approaches were investigated to generate this test signal with two ways becoming the mainstream that can be used. Creating an ideal 8-bit ADC in MATLAB and creating an ideal 8-bit ADC in Verilog-AMS. With Verilog-AMS, you can create and use modules that describe the high-level behavior and structure of analog, digital, and mixed-signal components and systems.

2.2.1.1 Ideal ADC in MATLAB

From previous work undertaken [2], an ideal 8-bit ADC was created in MATLAB for this particular 8-bit DAC being used in the BIST scheme. Figure 2 shows the digital
signal created by this 8-bit ADC at 125MHz. This digital input generates an accurate analog signal of 125MHz from the 8-bit DAC. Thus this ideal ADC in MATLAB can be used to generate a digital test signal at low frequencies.

The problem associated with this ideal ADC is that in the code, as the analog test signal's frequency goes up, the number of samples goes down. Using a sampled signal of 125MHz, the number of samples taken per signal period is 10. Using a sampled signal of 250MHz, the number of samples taken per signal period is 5. With this decrease in sampling, the highest frequency that can be sampled is 500MHz, which has 2.5 samples per period. The 500MHz signal cannot be used due to the inability to input half a sample into the 8-bit DAC. From this, the maximum frequency that the ideal ADC can generate for an input is 416.67MHz, which has 3 samples. The next test frequency can then be 312.5MHz, which has 4 samples, and then a 250MHz signal down to 125MHz. This trade off is not evident in the 125MHz sampled analog signal. Figure 3 shows an analog sampled signal of 1GHz, which has approximately 1.25 samples per period. From Shannon's theorem, this cannot work to accurately recreate the 1GHz sampled signal since more than 2 samples are needed per period to reconstruct the analog sampled signal (and the problem with loading a half sample into the DAC). This problem does not appear avoidable using this code. Due to this problem, an ideal ADC was required to be implemented to sample analog signals up to 1GHz.
Figure 2: Ideal ADC in MATLAB at 125MHz

Figure 3: Ideal ADC in MATLAB at 1GHz
The first approach to remedy this problem was to create an ideal 8-bit ADC in MATLAB, just constructed differently from the previous code. In appendix A, the MATLAB code of this attempt is supplied. The ideal ADC can sample signals at 1GHz, but more refinement is necessary to achieve an accurate digital sample. Thus this version of an ideal 8-bit ADC does not work correctly with the 8-bit DAC being used in the BIST

2.2.1.2 Ideal ADC in Verilog-AMS

A second approach to create an ideal 8-bit ADC to digitize the analog test signal was to create the ADC in Verilog-AMS. This approach allows for ease of testing when combined with the BIST circuit in Cadence to test in an Analog Mixed Signal (AMS) environment. In appendix B, the code from the first developed ideal 8-bit ADC can be seen that it is an extension of the ideal 4-bit ADC [2]. After testing of this ADC, the DAC did not produce desirable results. Modifications were implemented and a new ADC was created and tested. Figure 4 displays the digital signal created from an analog test signal of 125MHz. Figure 5 shows the output of the 8-bit DAC, before the buffer, to show that the Verilog-AMS ideal 8-bit ADC can generate an accurate 125MHz test signal.
Figure 4: Ideal ADC in VerilogAMS at 125MHz

Figure 5: Output of DAC at 125MHz
One problem that can be seen is the problem with getting the digital sampled signal from the output of the ideal 8-bit ADC into the stored memory of an on-chip ROM. Simply downloading the digital signal is not an option into the ROM chip using the Verilog-AMS model. Yet one could by hand, create a text file from the output of the ADC that could be downloaded into the ROM. Another solution to this problem is to also set up a test circuit within the AMS environment involving the ideal ADC and a ROM chip. A simulation can be run with the ADC connected to the ROM, and from that simulation, the ROM can be seen with the digital data stored within it.

2.2.2 8-Bit Digital to Analog Converter

The 8-bit DAC used in the BIST scheme was developed under research by [2]. This 8-bit DAC is a current steering architecture using a hybrid approach consisting of two sections of current sources, one a course array and the other a fine array. The course array current steering sources have 16 times the current drive then the current steering sources in the fine array [2]. The 8-bit DAC utilizes a clock speed of 1.25GHz to perform at 1.25 Giga Samples per Second (GSPS). A Binary-to-Thermometer decoder is used on the input of the DAC to decode the digital signal in conjunction with the current steering source array. This decoder uses And-Or-Invert (AOI) logic architecture [2]. Each current steering cell consists of transistors along with digital logic; the output of the current steering cells is then fed into an analog buffer, which then filters and adjusts the signal. The analog buffer consist of a low pass filter and an folded cascode amplifier along with DC offset correction to center the output signal to utilize the full output dynamic range (DR) [2]. Figure 6 shows the schematic design of the 8-bit DAC along with the analog buffer on the output.
2.2.2.1 8-Bit DAC Problems

For this BIST design, test signals for the CUT were needed to be upwards of 1GHz. The 8-bit DAC currently constructed did not allow for an output signal to be in excess of 1GHz and still have the output dynamic range (DR) required. The requirement of the CUT, being a 4-bit Flash ADC, is to have an output DR equal to 240mV causing an output load of 300fF on the BIST centered on a DC offset of 200mV. This output DR was required throughout the entire bandwidth (BW) of the signals that would be inputted into the ADC. The analog buffer was determined to be the culprit of the limited BW and the need to design a new buffer was evident.

Figure 7 shows the frequency response of the buffer used in the 8-bit DAC [2]. From the plot, it can be seen that the two conditions can not be meet and that just changing the DC offset correction circuit will not allow for the correct output DR throughout the BW of the buffer. This proved to be the greatest problem to generate test signals in excess of 1GHz and showed the need for a new buffer design.

Figure 7: Frequency Response of Original Analog Buffer
2.2.2.2 Pre-Amp Design

The first design solution investigated to successfully generate a 1GHz output test signal with good output DR was to design and integrate a pre-amp in front of the original analog buffer design. This approach was used to help ease the capacitive load on the current steering cells from the original analog buffer. By utilizing a pre-amp, an amplifier could be design with low input capacitance and high BW and high gain. The amplifier could then be designed to drive large loads and have a DC offset on the output at the correct bias voltage for the analog buffer.

Figure 8 shows the schematic of the original analog buffer, the DC offset correction is performed by using blocking capacitors on the input and output, then using a resistor divider circuit to bias at the correct DC offset. The low pass filter helps clean up the signal and the negative feedback allows for stability [2]. Figure 9 shows the modified analog buffer with the pre-amp integrated into the design. The pre-amp is placed before the input in the folded cascode amp, but after the output of the low pass filter. By changing the values of the resistor divider circuit to the correct bias voltage for the pre-amp, the pre-amp itself can then be designed to output the analog signal at the correct DC bias for the folded cascode amplifier.

2.2.2.2.1 Active Load Amp

Various amplifiers were investigated with the active load amp chosen as the pre-amp. The reasoning behind this is that an active load amp has low gain and high BW along with low input capacitance. The active load amp would be able to drive the large capacitive load caused by the folded cascode amplifier.
Figure 8: Schematic of Original Analog Buffer

Figure 9: Schematic of Modified Analog Buffer with Pre-Amp
The design of the active load amp was constricted to have a BW of at least 1GHz and as low gain as possible. The reason for the low gain was so that the insertion of the pre-amp would not cause the analog signal to exceed the input DR of the folded cascode amplifier of the original buffer. With these constraints and setting the length to 200nm, the width of the PMOS load transistor was determine to be 62.57um, and the width of the NMOS transistor was determine to be 64um. From simulations, the DC bias voltage at the input was found to be 0.521V to achieve an output DC offset of 0.6V, which is the required DC bias voltage of the folded cascode amplifier. The active load amplifier utilizes a VDD of 1.2V and VSS of 0.0V. By using the design constraints of an output DC offset of 0.6V, VDD of 1.2V, and VSS of 0.0V, this allowed for easier implementation into the original analog buffer design with minimal design changes.

Figure 10 shows the completed design schematic of the pre-amp.
Figure 11: Frequency Response of Active Load Pre-Amp

Figure 12: Transient Response at 1GHz
Figure 11 shows the frequency response of the active load amp along with the gain and figure 12 shows the transient response of an analog test signal at 1GHz. This shows that the active load amplifier can handle 1GHz signals.

2.2.2.2 Simulation Results of Pre-Amp Design

Figure 7 shows the frequency response of the original analog buffer design which had a 0dB BW of 463MHz with a 0dB frequency of 472MHz. The gain at the highest peak of the frequency response is 8.99dB with a settling time of approximately 60ns. Inserting the pre-amp into the analog buffer design as shown in figure 9 yielded vast improvements to the BW of the analog buffer. The modified buffer with pre-amp has a 0dB BW of 1.062GHz with a 0dB frequency of 1.066GHz. The gain at the highest peak of the frequency response is 13dB with a settling time of approximately 150ns. From this gain, the output DR of the modified buffer increased approximately to 600mV from the original buffer output DR of 400mV [2]. Figure 13 shows the frequency response of the modified buffer with the active load amplifier as the pre-amp.

![Graph showing frequency response](image)

Figure 13: Frequency Response Modified Buffer with Pre-Amp
2.2.2.3 Problems with Modified Analog Buffer Design

From figure 13, it can be seen what the main problem is with the modified analog buffer. The frequency response shows that the modified buffer does not have a constant gain across the BW. From this problem, this means that the output DR of the modified buffer will not be constant and will change throughout the BW and that in actuality; the BW is not what it appears to be. The BW for the flash ADC from this buffer is much smaller since the $f_{3db}$ frequency would need to be taken from the highest peak. From this, the BW is approximately 400MHz and with the same approach of measuring $f_{3db}$ utilized on the original buffer design as well, this showed a lower BW than what was initially measured. The 4-bit flash ADC and various other CUT’s cannot deal with the multi-changing of the analog test signal being inputted in. Thus another buffer was needed that could provide a constant gain across the BW (being the $-3$dB BW) and still have a BW of 1GHz.

2.2.2.3 New Buffer Design

A new approach to the buffer problem was determined and a new buffer was required to be designed. By investigating various amplifier ideals, using differential input, high BW amplifiers with inductors, the amplifier design from [4] was chosen. This amplifier proved that it could have unity gain and high BW and showed that the gain was constant across the BW. This would allow the output DR of the buffer to be constant along the frequency range of test signals used.

2.2.2.3.1 New Analog Buffer Design

Figure 14 shows the finished schematic design of the new analog buffer amplifier [4]. The approach used to design the buffer was to start off with the differential input
stage and design the stage for unity gain. Once load currents were established, each stage
could then be set equal to the previous stage load currents and widths could then be
determine for each stage. The output stage would therefore be designed separately and use
design constraints set for the BW of 1GHz and a load capacitive of 1pF (generic value).
By over estimating the capacitive load, the actual BW being designed would be greater
than 1GHz. Once all the transistor widths were determined, the coupling capacitor and
resistor were calculated. By setting the capacitor lower by a factor of 0.22 of the load
capacitance, the resistor was then determined by:

$$f_{3dB} < \left( \frac{1}{RC} \right)$$

(1)

Letting $R = 3.3\,\text{K}\Omega$, this ensured that the zero introduced from the resistor would
be greater than 1GHz, thus the BW of the new buffer can be 1GHz.

![Schematic of Amplifier Design](image)

Figure 14: Schematic of Amplifier Design
2.2.2.3.2 Bias Circuit Design

The next component that needed to be designed for the new analog buffer was the bias circuit that supplied the two bias voltages. Figure 15 shows the completed schematic of the bias circuit. Bias voltage 1 was required to be 827.5mV and bias voltage 2 was required to be 759.5mV [4]. By adjusting the widths of the transistors, the final values simulated for bias voltage 1 was equal to 827.492mV and bias voltage 2 equal to 759.532mV. Figure 16 shows the Cadence simulation of the bias voltages.

Figure 15: Schematic of Bias Circuit
2.2.2.3.3 Simulation Results of New Buffer Design

From [4] the BW of the original amplifier design was 722.14MHz and the power dissipation was 1.5mW at slightly above unity gain. The new amplifier design showed a significant increase in the BW and a slight increase in the power dissipation. Figure 17 shows the frequency response and figure 18 shows the DC response of the new amplifier design. The new amplifier design showed an increase in BW up to 947MHz at a load of 1pF. This increase in BW only raised the power dissipation to 2.36mW throughout the frequency range of the BW. From figure 18, the output DR that the amplifier can handle is greater than 300mV, which allows for the required output DR of 240mV. The most important improvement is the constant gain over the entire BW region. This constant gain allows for the output DR to remain the same throughout the frequency test range. Figure 19 shows the transient response at 300MHz and figure 20 shows the power
consumption at 300MHz. Figure 21 shows the transient response at 1GHz and figure 22 shows the power consumption at 1GHz. Even though the BW is not quite 1GHz, simulations were performed at 1GHz to see the test results. The signal did not degrade much as initially expected. So theoretically, a 1GHz signal could be outputted from the new amplifier design.

![Graph](image)

Figure 17: Frequency Response of New Amplifier Design
Figure 18: DC Response of New Amplifier Design

Figure 19: 300MHz Transient Response of New Amplifier Design
Figure 20: Power Consumption of New Amplifier Design at 300MHz

Figure 21: 1GHz Transient Response of New Amplifier Design
Figure 22: Power Consumption of New Amplifier Design at 1GHz

2.2.2.3.4 Problems with New Buffer Design

As with the pre-amp design, problems arise from the new buffer design. This problem consists of not having a great enough output dynamic range DR on the output signal. Though the amplifier itself can handle an output signal with an output DR just greater than 300mV, the actual output DR was so small that it was not necessary to include the value in this thesis! The problem is from the fact that the amplifier is at just greater than unity gain, and the output of the DAC, but before the input to the analog buffer, is at a low voltage of amplitude 5mV (see figure 5). Thus the analog buffer acted like a true buffer and did not amplify the output of the DAC. Without the gain, the output DR of 240mV could not be reached. Yet a new approach was needed to satisfy the design constraints of a BW of 1GHz driving a 300fF capacitive load and an output DR of 240mV.

2.2.2.4 Final Buffer Design

The final buffer design comes from the idea of combining the previous two buffer designs together. By using the new buffer design and the active load pre-amp, the theory is that the benefits of both amplifiers can be used to achieve the design goal of the analog
buffer. The low input capacitance and high BW of the active load pre-amp and the constant gain throughout the BW of the new buffer amplifier.

2.2.2.4.1 Final Buffer Amplifier

The approach taken to design the final amplifier design was to ensure an amplifier that would have a BW of at least 1GHz and be able to achieve a high gain. With the low transistor widths on the input of the new buffer from the previous section, the current steering arrays from the DAC would not need to drive a large load. With the low input capacitance, the new pre-amp design from the previous sections could then be connected to the output of the new buffer design and then chained together with a second pre-amp design to provide the required gain to achieve the correct output DR. Figure 23 shows the amplifier configuration with the new buffer design as the first block (see figure 14) and the second and third block as the pre-amp design (see figure 10). As a side note, though it's called a pre-amp, the two active load amplifiers are used on the output of the amplifier design. The term pre-amp comes from the first attempt to redesign the buffer (see previous sections).

Figure 23: Final Buffer Amplifier Configuration
2.2.2.4.2 Simulation Results of New Buffer Amplifier

Simulations results of the amplifier showed that for optimal gain and input DR, a DC offset of 700mV would be required. Figure 24 shows the DC output waveform. From the DC output, it can be see that the output DR of the amplifier is approximately 500mV and the input DR is approximately 150mV. This shows that the new configuration will handle an output signal of 240mV, provided that there is enough gain to adjust the low input signal. To achieve the DC input offset voltage of 700mV and achieve the DC offset voltage of 200mV, the approach taken in [2] was used by placing DC blocking capacitors on the input and output of the buffer design and then using a resistor divide network to perform the DC offset correction. DC blocking capacitors where set at values of 5pF for the input and 4pF for the output [2], though recent simulations have shown that these values can be reduced. By reducing the blocking capacitors, the overall frequency response will increase but the output DR will decrease slightly. Figure 25 shows the completed schematic of the final buffer design with the DC blocking capacitors and the previous required bias circuit that is used in the DAC.

Figure 26 shows the AC frequency response of the final buffer amplifier configuration. From the figure, the gain can be seen as 7.72dB with an f_{0dB} BW of 2.2474GHz, a significant increase from the previous two design attempts. The most important increase is the f_{3dB} BW, which increased to 1.192GHz. This increase shows the new amplifier configuration can handle 1GHz test signals. Figure 27 shows the transient response of the amplifier at 125MHz and figure 28 shows the transient response at 1GHz.
Figure 24: DC Response of Final Buffer Amplifier

Figure 25: Schematic of Final Buffer Design
Figure 26: Frequency Response of Final Buffer Amplifier

Figure 27: Transient Response of Final Buffer Amplifier at 125MHz
Figure 28: Transient Response of Final Buffer Amplifier at 1GHz

2.2.2.4.3 Problem and Design Solution to Final Buffer Amplifier

From figures 27 and 28, the output DR is still not at the required 240mV. This problem occurs due to the gain of the amplifier not being great enough to overcome the low output signal of the current steering arrays in the DAC. From [2], the output signal of the DAC is a current signal that is converted to a voltage for the buffer by a resistor tied to the output of the current steering arrays. This resistor can be seen in figure 25 connected to the input label “Iout”. By adjusting the value of the resistor and utilizing the input DR of the final buffer design, the output signal of the DAC can be increased to allow the final buffer design to perform the DC offset correction and amplitude
adjustment so an output signal will have the required 240mV. To calculate the value of
this resistor, first the required output signal of the DAC would need to be determined.
This output signal is the input signal of the final buffer design. From simulations, a
signal of amplitude 74mV would be required on the output of the current steering arrays.
The next step to calculate the resistor value was to setup the math equation to find the
proportional value set equal to the relationship between the original resistor value and
original voltage value. This is performed by using ohm’s law and setting the current of
each equation equal to each other.

\[ R' = \left( \frac{(\text{New Amplitude}) \times (\text{Original Resistance})}{\text{Original Amplitude}} \right) \]  

(2)

From the previous sections, the new amplitude is equal to 74mV, the original
resistance is equal to 62Ω, and the original amplitude is 5mV. This calculated to a new
resistor of 917.6Ω, but due to parasitic capacitance within the circuit, the final value
chosen was raised to an even 1KΩ.

2.2.2.4.4 Simulation Results of Final Buffer Design

Figure 29 shows the transient response at 125MHz with an input signal of 74mV
amplitude and figure 30 shows the transient response at 1GHz. This shows the final
buffer achieving the required output DR of 240mV. Figure 31 shows the AC response of
the final buffer design including the DC blocking capacitors and figure 32 shows the
power dissipation of the final buffer design with the DC blocking capacitors and bias
circuits. From these plots, the improvements over the original buffer design can be seen
and the two required conditions are then met. Table 1 shows the comparisons between
the original buffer design [2] and the new modified design.
Table 1: Original Buffer Design vs. New Modified Buffer Design

<table>
<thead>
<tr>
<th></th>
<th>Original Buffer</th>
<th>New Buffer</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_{V_{LF}}$</td>
<td>6.8dB</td>
<td>6.35dB</td>
</tr>
<tr>
<td>-3dB BW</td>
<td>300MHz</td>
<td>923.5MHz</td>
</tr>
<tr>
<td>0dB BW</td>
<td>463MHz</td>
<td>1.5177GHz</td>
</tr>
<tr>
<td>Power Dissipation</td>
<td>27.2mW*</td>
<td>18.875mW</td>
</tr>
</tbody>
</table>

Figure 29: Transient Response of Final Buffer Design at 125MHz
Figure 30: Transient Response of Final Buffer Design at 1GHz

Figure 31: Frequency Response of Final Buffer Design
Figure 32: Power Dissipation of Final Buffer Design

From table 1, the improvements can be seen that the -3dB BW was greatly increased. The -3dB BW measurement is taken from the original buffer AC frequency response (see figure 7) and the final buffer design AC frequency response. By locating the maximum gain on the curve, then finding the frequency on both sides of the gain at -3dB, the difference between the two frequencies is the -3dB BW. This measurement is critical to the overall performance of the DAC as this measurement shows the BW the DAC can still output the test signal at the correct output DR. The note of interest in the table is the power dissipation. The power dissipation for the original buffer design is of the cascode amplifier design without its bias circuits and the DC blocking capacitors. The power dissipation of the final buffer design is of the completed buffer. As a side note, the DC blocking capacitors add problems to the overall frequency response and there is a need to remove them from the design. This will be presented in the recommendation section.
2.2.3 Analog MUX

To have the ability to probe multiple points for testing purposes, a high speed analog MUX was required to be developed that could handle analog signals in excess of 1GHz, and be able to drive an average size capacitive load. To begin the design, the constrains where set as having a need for a BW in excess of 1GHz and a output capacitive load at 1.5pF. The 1.5pF load was chosen to help ensure that if a large load was connected to the output of the MUX, such as the ΣΔmodulator, that the analog MUX could still perform at high speeds with a BW of 1GHz. If the analog MUX could not handle a 1GHz input signal, then the BIST scheme itself would not operate correctly.

2.2.3.1 Analog MUX Design

With the design constraints set at a BW of 1GHz and capacitive load at 1.5pF, along with VDD equal to 1.2V and VSS equal to 0.0V, the approach to designing the MUX was a straightforward approach. First, an inverter was needed and then an analog switch, a decoder was required, and then combining everything together to implement the analog MUX. For this effort, a 4 to 1 analog MUX was designed though the circuit could easily be improved to higher orders of a MUX.

2.2.3.1.1 Inverter Design

The inverter design was a straightforward design that used the general rule of thumb of making the PMOS widths 3 times the size of the NMOS widths. Since the inverter would not drive large capacitive loads, the minimum lengths and small widths could be used to keep power consumption and chip area down. Figure 33 shows the completed schematic of the inverter, with a width of 1um for the NMOS and a width of 3um for the PMOS transistors. Figure 34 shows the simulation results that confirm the
inverter operates correctly, since the inverter does not perform any operations on the input test signal, there was no need to design the inverter for 1GHz.

Figure 33: Schematic of Inverter

Figure 34: Functional Simulation Results of Inverter
2.2.3.1.2 Analog Switch

To design the analog switch, the main design constraint of a BW of 1GHz would need to be taken into account. The main focal point of the analog MUX operation and performance is primarily on the analog switch. Investigating an NMOS switch and PMOS switch, the final design chosen was the CMOS switch. This would combine the best of both worlds from the NMOS and PMOS switches and minimize the trade offs associated with the two styles of switches.

Figure 35 shows the completed schematic of the CMOS switch. To reduce the effects of charge injection and capacitive feed through, the widths of both the PMOS and NMOS transistors where set equal to one another. To begin the design, both transistors were modeled as resistors, with the PMOS being $R_{on1}$ and the NMOS being $R_{on2}$. This model would represent what the switch would look like during operation when both transistors were turned on. With both modeled resistance in parallel, an $R_{eq}$ could be determine and then the RC time constant could be calculated to ensure the BW of 1GHz:

$$R_{eq} \times C_L = \left(\frac{1}{f_0}\right)$$  \hspace{1cm} (3)

Setting $C_L$ equal to 1.5pF and $f_0$ equal to 1GHz from the previously stated design constraints calculated that $R_{eq}$ was equal to 667Ω. Since the width of the NMOS transistor and the width of the PMOS were set equal and in parallel, then $R_{on1}$ and $R_{on2}$ would be twice as much as $R_{eq}$. Thus the on resistance of each transistor was 1.333KΩ.

39
Figure 35: Schematic of CMOS Switch

From this information, using the equations:

\[ R_{on} = \frac{1}{\beta \times (V_{GS} - V_m)} \]  
\[ R_{ON} = \frac{1}{K \times \left(\frac{W}{L}\right) \times (V_{DD} - V_{in})} \]  

Equation 5 is just the previous equation expanded out. From figure 35, it can be seen that \( V_{gs} \) of either transistor is equal to \( V_{DD} - V_{in} \). From this equation, setting it equal to \( W \) (width of transistor) and letting \( V_{in} \) equal to a range from 0V to \( V_{DD} - V_{in} \), a plot of the width vs. input voltage can be obtained. Figure 36 shows the plot of the width vs. input voltage performed in MATLAB of the NMOS transistor. This plot shows the values of the width that can be used throughout a range of input values. The plot also
shows the input DR of the CMOS switch. From the plot, picking transistor width values within the input range (the flat part of the curve) and then setting the PMOS transistor width equal to the value of the NMOS width determines the design of the CMOS switch. Since these values are approximate values due to low level modeling equations and parameters (see chapter 3 of this thesis for information on analog modeling), simulations were performed to finalize the design of the CMOS switch. Figure 37 shows the simulation results of the CMOS switch at 300MHz. Figure 38 shows the simulation results at 800MHz and figure 39 shows the simulation results at 1GHz. The CMOS switch performed as expected, showing it could handle an input frequency range up to 1GHz.
Figure 37: Simulation of CMOS Switch at 300MHz

Figure 38: Simulation of CMOS Switch at 800MHz
Figure 39: Simulation of CMOS Switch at 1GHz

2.2.3.1.3 Decoder

For this research, a 4 to 1 analog MUX was designed which then requires a 2 to 4 decoder circuit to control the analog switches. Table 2 shows the logic outputs of the 2 to 4 decoder circuit.

Table 2: Logic Flow of 2 to 4 Decoder

<table>
<thead>
<tr>
<th>( S_1 )</th>
<th>( S_0 )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Figure 40 shows the completed schematic of the 2 to 4 decoder. Along with using the previous designed inverter, a 2-input AND gate was needed to be designed. The approach was taken from using the minimal length transistors and using the same approach from the inverter to determine the widths. Figure 41 shows the completed schematic of the AND logic gate. Figure 42 shows the simulation results of the functionality of the AND gate with "C" being the output and figure 43 shows the simulation results of the 2 to 4 decoder. The simulations show that both designs work as expected.

Figure 40: Schematic of 2 to 4 Decoder
Figure 41: Schematic of AND Gate

Figure 42: Simulation Results of AND Gate
2.2.3.1.4 Final Design of Analog MUX

Table 3 shows the logic table of the analog MUX and figure 44 shows the schematic of the final design of the analog MUX. Figure 45 shows the simulation results using input analog test signals of 1 GHz at different voltage levels and different offset voltages. The simulations show that the analog MUX works as expected and can be incorporated into the final BIST scheme design to allow for probing of multiple points.

Table 3: Logic Inputs and Output Control of Analog MUX

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>S₁</td>
<td>S₂</td>
<td>Out</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>In₀</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>In₁</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>In₂</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>In₃</td>
</tr>
</tbody>
</table>
Figure 44: Schematic of Analog MUX

Figure 45: Simulation Results of Analog MUX
2.2.4 Analog Response Extractor

The Analog Response Extractor (ARE) consists of a first order \(\Sigma\Delta\) modulator to convert the output signal of the probed point to a 1-bit digitized signal. This signal can then be compared to a working model signal to give detection of a fault within the CUT.

The first order \(\Sigma\Delta\) modulator is provided by [2] and implemented in Verilog-AMS code. This is mostly due to time constraints as the original goal was to design and build a first order \(\Sigma\Delta\) modulator in the Cadence schematic capture environment. The modulator consists of three blocks which include an analog signal generator for clock generation, comparator for clock generation, and the first order \(\Sigma\Delta\) modulator [2]. The code is basic code for all components and is not included in the appendix. It can be found in the reference [2].

Figure 46 shows the connection of the modulator components along with where the input to the modulator is. Figure 47 shows the simulation results of the \(\Sigma\Delta\) modulator at 125MHz and figure 48 shows the results at 1GHz including using a higher sampling frequency. From these simulations, it can be seen that the analog input signal is digitized into a 1-bit digital bit stream. This digital signal can then be sent to the Digital Response Analyzer for comparison and verification of the CUT.

Figure 46: Schematic View of First Order \(\Sigma\Delta\) Modulator
Figure 47: Simulation Results of First Order $\Sigma \Delta$ Modulator at 125MHz

Figure 48: Simulation Results of First Order $\Sigma \Delta$ Modulator at 1GHz
2.2.5 Digital Response Analyzer

Coupled with a Digital Response Analyzer (RA), the BIST scheme is completed for on-chip fault and circuit detection. The time constraints proved to not allow for a fully developed RA, but the overall design is basic enough. By storing the correctly working digital signal in an on-chip ROM or look up table (LUT), simple digital logic circuits can be designed to then compare that working stored digital signal with the output of the $\Sigma\Delta$ modulator. Flags can be set up to trigger events using the digital control logic when the two digital signals are not the same. The comparison circuit would even compare the results bit by bit with proper timing to ensure that there is a detection of a fault within the CUT.

2.3 BIST Simulation Results

Stated before, the goal of the BIST scheme being presented was to generate test signals upwards of 1GHz for a CUT and to be able to perform fault detection on the CUT. The test setup for the process utilizes the 8-bit ideal ADC in Verilog-AMS, an analog signal generator, and a clock circuit for the ideal ADC along with an 8-bit shift register when necessary, all in Verilog-AMS. These blocks combine to form the input into the 8-bit DAC. Figure 49 shows the block setup of the BIST scheme test setup. Some points of interest is that the analog MUX is not included in the test setup and due to the timing of the DAC, the 8-bit shift register is not needed for an analog signal of 125MHz. This is due to the timing of the ideal ADC matching the clock of the DAC, so there is no need for the bank of shift registers for the test setup at 125MHz. The shift registers are used for other test frequencies and would be used for the BIST scheme, since the on-chip ROM would load the data into the shift registers and then the shift registers...
could input the data into the DAC at the correct rate of 1.25GHz. The analog MUX was not used since the CUT is in fact the 8-bit DAC at this time. Though once implemented, the BIST scheme would utilize the analog MUX to achieve the ability to probe multiple points of the CUT. Thus the BIST test setup was designed to show proof of concept of the ability to do fault detection.

Figure 50 shows the output of the BIST test setup. The signal “vout” is the output of the ΣΔ modulator and “net17” is the output of the final analog buffer design. The signal “net067” is the output of the DAC after the DC offset correction, but before the input into the final analog buffer design. From figure 50, “vout” shows the digitized signal of “net17”, which is at a frequency of 125MHz. The voltage level of 1V represents a logic 1 and voltage level of -1V represents logic 0.

Figure 49: BIST Test Setup
Figure 50: BIST and DAC Simulation Results at 125MHz

Figure 51 shows the output of the BIST test setup when a fault condition has occurred in the CUT (which is the 8-bit DAC). By considering transistor stuck-on fault on one NMOS transistor from its drain to its source, the fault condition was created. The signal “vout” represents the output of the EΔ modulator and the signal “net17” is the output of the completed DAC (which includes the final buffer design). The output of the DAC shows that there is still some oscillation going on at a frequency of 125MHz. Yet it is clear that the output is not a good sinusoidal signal with a frequency of 125MHz.

Comparing the output of the ΣΔ modulator to the output of the ΣΔ modulator in figure 50 shows that the digitized data created generates a different set of bits then the correctly working DAC. From this difference, the fault is detected and the circuit is then determined not to be working correctly.
Figure 51: BIST Fault Simulation Results at 125MHz

Figure 52: BIST and DAC Simulation Results at 250MHz
Figure 53: BIST Fault Simulation Results at 250MHz

Figure 52 and figure 53 show the same test setup as the previous test, but using a test signal of 250MHz for both the correct working output and the fault output. From the two figures, it can still be seen that the digitized data generated, "vout" is different from figure 52 to figure 53. This shows the fault detection is achieved at 250MHz.

Using the simulator Virtuoso Aptivia Specification-Driven Environment (Vsde) in Cadence yielded a Spur Free Dynamic Range (SFDR) for test signals of 125MHz at 19.68dB and 250MHz at 17.43dB, based on the 256 point FFT output spectrum. Figure 54 and figure 55 show the spectrum plots generated from the Vsde simulator.
generated from the Vsde simulator. It is believe that the plotted output is the correct SFDR, reasons are discussed in the recommendation section (see chapter 4).

Figure 54: Spectrum of 125MHz

Figure 55: Spectrum of 250MHz
Figure 56: DAC Simulation Results at 416.67MHz

Figure 57: Spectrum of 416.67MHz
Figure 56 shows the DAC output at 416.67MHz labeled as "VT("/OUT")". Due to timing issues that could not be corrected at this time, the BIST test setup could not be tested at this specific frequency. From figure 56, it can be seen that the DAC can produce a 416.67MHz output waveform with good DR. From the Vssd simulation, the test signal of 416.67MHz generated a SFDR of 11.14dB, figure 57 shows the spectrum output plot to show the difference in data generated. It is believe that a longer simulation run would be required to accurately produce the SFDR of this test signal. Due to disc space, a simulation could not be performed.

Due to Verilog-AMS code problems, fault signals could not be generated for a test frequency of 800MHz and 1GHz at this present time. This is due to using a wrong time scale that limited the maximum sampling frequency that could be used for the ideal ADC at 5GHz. For the ideal ADC used in the BIST test setup, a sampling frequency 10 times greater than the desired output test signal was required to produce a correct digital representation. Simulations were run with the sampling frequency capped at 5GHz for a test signal of 1GHz, but the DAC output was not a clean analog signal. Upon correcting the time scale problem defined in Verilog-AMS, not enough time was available to run the BIST fault test setup. Also due to timing errors loading the inputs, an 800MHz and a 1GHz test signal could not be demonstrated at this time. Recommendations are including in chapter 4 for correcting this problem.

From this problem, the SFDR could not be determined from the simulations on the 8-bit DAC. For analysis and comparison purposes, the Vsde simulator was used on the final design of the analog buffer with test signals of 800MHz and 1GHz. The test signal of 1GHz produced a SFDR of 11.98dB and the test signal of 800MHz produced a
SFDR of 14.36dB. Though these numbers show the performance of the final design of the analog buffer, it is believed that comparable numbers could be generated on the modified 8-bit DAC as well with slight changes in the simulation environment. Recommendations are presented that should allow for a more accurate SFDR calculation. Figures 58 and 30 show the output of the final design analog buffer with a load capacitance of 300fF and an input that is comparable to what would be the input to this buffer from within the DAC at test frequencies of 800MHz and 1GHz. Figure 60 shows the output of the 8-bit DAC with a test signal of the combined 125MHz and 250MHz. This shows the modified 8-bit DAC can generate two-tone signals.

![Graph](image)

Figure 58: Transient Response of Final Buffer Design at 800MHz
Figure 59: DAC Simulation Results at 125MHz and 250MHz
3.0 Analog Modeling with MATLAB

3.1 BSIM3 Modeling

Two models were researched during the process to determine the accuracy of them compared to the Cadence simulation models. The first model investigated was the BSIM3 level -11 model. This model consists of multiply equations to represent the physical properties of the silicon material used to construct an Enhancement MOSFET Transistor. The common trend between the BSIM3 model and other third generation models is the use of smoothing functions [5]. The smoothing function allows for a continuous transition from one operating region to the next. This allows for separate mathematical equations to be derived that describe each operating region of the transistor, and then the smoothing function connects the different operating regions together. As will be discussed below, there are many factors to consider when modeling a MOSFET transistor. When combined together, all the contributing models will allow for the DC Characteristics, AC Response, and Transient Analysis to be simulated in a computer environment. Discussed below is a brief background and reference of device modeling along with the description of the various factors used. See Appendices D through W for equations and MATLAB code and to cross-reference the specific section with the specific quantity calculated.

3.1.1 I-V Model

The I-V (Current-Voltage) Model starts out by defining different parameters that will be used in the calculations of the different drain currents and voltages of the
MOSFET Transistor. Relatively few input conditions are needed by the user, but the obvious ones of Gate Voltage \( V_G \), Drain Voltage \( V_D \), Bulk Voltage \( V_B \), Source Voltage \( V_S \), and the Device Temperature \( T_{\text{device}} \) are needed to be inputted for the equations to model the transistor. When binning the model together, the inputted data required comes from collecting the voltages from previous models in the overall circuit being simulated. For testing purposes, the values used are: \( V_G = 0.6 \) V, \( V_D = 1.2 \) V, \( V_B = 0.0 \) V, \( T_{\text{device}} = 25^\circ \text{C} \), and \( V_S = 0.0 \) V.

3.1.1.1 BSIM3 Parameters

Table 4 list the parameters used to calculate all the equations for the BSIM3 model along with a description of each parameter [4] and [5]. Values are not listed due to confidentiality agreements.

<table>
<thead>
<tr>
<th>BSIM3 Parameter (Units)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOFF (V)</td>
<td>Offset Voltage</td>
</tr>
<tr>
<td>VSAT (m/s)</td>
<td>Carrier Saturation Velocity</td>
</tr>
<tr>
<td>VTH0 (V)</td>
<td>Threshold Voltage of Device</td>
</tr>
<tr>
<td>A0 (no units)</td>
<td>Short Channel Bulk Charge Coefficient</td>
</tr>
<tr>
<td>A1 (V(^{-1}))</td>
<td>Saturation Voltage Fitting Parameter 1</td>
</tr>
<tr>
<td>A2 (no units)</td>
<td>Saturation Voltage Fitting Parameter 2</td>
</tr>
<tr>
<td>AGS (V(^{-1}))</td>
<td>Gate Bias Effect on Abulk</td>
</tr>
<tr>
<td>B0 (m)</td>
<td>Narrow Channel Bulk Charge Coefficient</td>
</tr>
<tr>
<td>B1 (m)</td>
<td>Offset to B0</td>
</tr>
<tr>
<td>CDSCB (F/(V(^{2})m(^3)))</td>
<td>Substrate bias dependency of CDSC</td>
</tr>
<tr>
<td>CDSC (F/(m(^2)))</td>
<td>Drain Source-Channel Coupling Capacitance</td>
</tr>
<tr>
<td>CDSCD (F/(V(^{2})m(^3)))</td>
<td>Drain Bias Dependency of CDSC</td>
</tr>
<tr>
<td>CIT (F/(m(^3)))</td>
<td>Interface Trap Capacitance</td>
</tr>
<tr>
<td>DELTA (V)</td>
<td>Vdsx Smoothing Parameter</td>
</tr>
<tr>
<td>deltal (no units)</td>
<td>Constant</td>
</tr>
<tr>
<td>DROUT (no units)</td>
<td>DIBL effect on Rout Coefficient</td>
</tr>
<tr>
<td>DVT1 (no units)</td>
<td>Short Channel Effects Coefficient 2</td>
</tr>
<tr>
<td>DVT1W (m(^{-1}))</td>
<td>Width Effect on Short Channel Effect Coefficient 2</td>
</tr>
<tr>
<td>DVT2 (V(^{-1}))</td>
<td>Substrate Bias Dependence of Short Channel Effects</td>
</tr>
<tr>
<td>DVT2W (V(^{-1}))</td>
<td>Substrate Bias Influence of Width on Short Channel Effects</td>
</tr>
</tbody>
</table>

61
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVT0</td>
<td>Short Channel Effect Coefficient 1</td>
</tr>
<tr>
<td>DVT0W</td>
<td>Width Effect on Short Channel Effect Coefficient 1</td>
</tr>
<tr>
<td>DSUB</td>
<td>Substrate Bias Effect on DIBL</td>
</tr>
<tr>
<td>DWG</td>
<td>Gate Bias Dependence of Effective Channel Width</td>
</tr>
<tr>
<td>DWB</td>
<td>Substrate Bias Dependence of Effective Channel Width</td>
</tr>
<tr>
<td>ETA0</td>
<td>Subthreshold DIBL Coefficient</td>
</tr>
<tr>
<td>ETAB</td>
<td>Substrate Bias Effect on Subthreshold DIBL</td>
</tr>
<tr>
<td>imax</td>
<td>Max Current Transistor can Conduct</td>
</tr>
<tr>
<td>imelt</td>
<td>Current at which Transistor will Physically Melt</td>
</tr>
<tr>
<td>K1</td>
<td>Body Effect on Threshold Voltage (First Order Term)</td>
</tr>
<tr>
<td>K2</td>
<td>Body Effect on Threshold Voltage (Second Order Term)</td>
</tr>
<tr>
<td>K3</td>
<td>Narrow Width Coefficient</td>
</tr>
<tr>
<td>K3b</td>
<td>Substrate Bias Dependence of K3</td>
</tr>
<tr>
<td>KETA</td>
<td>Substrate Bias Effect on the Bulk Charge</td>
</tr>
<tr>
<td>LNT</td>
<td>Source/Drain underdiffusion of gate</td>
</tr>
<tr>
<td>LL</td>
<td>Coefficient for Length Dependence of Length</td>
</tr>
<tr>
<td>LLN</td>
<td>Power Coefficient for Length Dependence of Length</td>
</tr>
<tr>
<td>LMIN</td>
<td>Length of transistor</td>
</tr>
<tr>
<td>LW</td>
<td>Coefficient of width dependence for length offset</td>
</tr>
<tr>
<td>LWL</td>
<td>Coefficient of length and width cross term for length offset</td>
</tr>
<tr>
<td>LWN</td>
<td>Power of width dependence of length offset</td>
</tr>
<tr>
<td>NLX</td>
<td>Nonuniform Lateral Doping Parameter</td>
</tr>
<tr>
<td>NCH</td>
<td>Channel Doping Concentration</td>
</tr>
<tr>
<td>NDS</td>
<td>Constant</td>
</tr>
<tr>
<td>NFACTOR</td>
<td>Subthreshold slope coefficient</td>
</tr>
<tr>
<td>NGATE</td>
<td>Gate Doping Concentration</td>
</tr>
<tr>
<td>PCLM</td>
<td>Channel Length modulation parameter</td>
</tr>
<tr>
<td>PDIIBC1</td>
<td>Output Conductance DIBL Parameter 1</td>
</tr>
<tr>
<td>PDIIBC2</td>
<td>Output Conductance DIBL Parameter 2</td>
</tr>
<tr>
<td>PDIIBC</td>
<td>Substrate bias dependency of DIBL</td>
</tr>
<tr>
<td>PRWB</td>
<td>Substrate bias dependency of the series resistance</td>
</tr>
<tr>
<td>PRWG</td>
<td>Bias Dependency of the series resistance</td>
</tr>
<tr>
<td>PSCBE1</td>
<td>SCBE parameter 1</td>
</tr>
<tr>
<td>PSCBE2</td>
<td>SCBE parameter 2</td>
</tr>
<tr>
<td>PVAG</td>
<td>Gate bias dependency of the earlier voltage</td>
</tr>
<tr>
<td>RDSW</td>
<td>Series Resistance/unit length</td>
</tr>
<tr>
<td>TINOM</td>
<td>Default temperature of device</td>
</tr>
<tr>
<td>TOX</td>
<td>Oxide thickness</td>
</tr>
<tr>
<td>U0</td>
<td>Low field mobility</td>
</tr>
<tr>
<td>UA</td>
<td>Gate field induced mobility reduction parameter (First Order)</td>
</tr>
<tr>
<td>UB</td>
<td>Gate field induced mobility reduction parameter (Second Order)</td>
</tr>
<tr>
<td>UC</td>
<td>Bias Dependence of Mobility Reduction</td>
</tr>
<tr>
<td>W0</td>
<td>Narrow width effect W offset</td>
</tr>
<tr>
<td>WINT</td>
<td>Isolation reduction of channel width</td>
</tr>
<tr>
<td>WL</td>
<td>Coefficient of length dependence for width offset</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>WLN</td>
<td>Power of length dependence of width offset</td>
</tr>
<tr>
<td>WMIN</td>
<td>Min. width of transistor</td>
</tr>
<tr>
<td>WR</td>
<td>Channel effect of the series resistance</td>
</tr>
<tr>
<td>WW</td>
<td>Coefficient of width dependence for width offset</td>
</tr>
<tr>
<td>WWLN</td>
<td>Coefficient of length and width cross term for width offset</td>
</tr>
<tr>
<td>WWN</td>
<td>Power of width dependence of width offset</td>
</tr>
<tr>
<td>XJ</td>
<td>Junction depth</td>
</tr>
<tr>
<td>xp/xpart</td>
<td>Charge partition flag</td>
</tr>
<tr>
<td>UA1</td>
<td>Temperature coefficient for UA</td>
</tr>
<tr>
<td>UB1</td>
<td>Temperature coefficient for UB</td>
</tr>
<tr>
<td>UTE</td>
<td>Temperature coefficient for U0</td>
</tr>
<tr>
<td>PRT</td>
<td>Temperature coefficient of RDSW</td>
</tr>
<tr>
<td>AT</td>
<td>Temperature coefficient for carrier saturation velocity</td>
</tr>
</tbody>
</table>

### 3.1.1.2 Effective Channel Length and Width

The effective channel length and width for the I-V calculations is not the length and width specified by the user. When an integrated circuit is “floor planned” at the graphic design level, the channel length and width are generally defined by two types of shapes [5]. There’s a drawn width and drawn length used in the layout of the transistor. This drawn dimension is what the designer in the circuit layout and simulation uses. Implemented on the silicon wafer is the printed measurement of the length and width [5]. Formulating a relationship between the printed measurement and the drawn measurement allows for the calculation of the effective length and width. This measurement is what is fabricated on the wafer and is the correct length and width used in the calculations for simulation purposes (see Appendix D).

### 3.1.1.3 Quantity Calculation

Before calculating the current and voltages of the transistor being model, specific quantities need to be determined by calculations. Table 5 lists the quantities calculated before the current and voltage equations can be used.
For the below table, a note must be taken to see that the temperature is converted to Kelvin from Celsius. This ensures proper units and realistic values when computing the effects of temperature on the device (see Appendix E).

Table 5: BSIM3 Quantity Calculations

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( q )</td>
<td>Charge of Electron</td>
</tr>
<tr>
<td>( k )</td>
<td>Temperature Constant</td>
</tr>
<tr>
<td>( \varepsilon_{OX} )</td>
<td>Permittivity of Silicon Dioxide</td>
</tr>
<tr>
<td>( \varepsilon_{S} )</td>
<td>Silicon Permittivity</td>
</tr>
<tr>
<td>( T ) (K)</td>
<td>Temperature of device in Kelvin</td>
</tr>
<tr>
<td>( E_g _T ) (eV)</td>
<td>Silicon Band Gap Energy Value</td>
</tr>
<tr>
<td>( n_i ) (q<em>E_g _T</em>NOM)</td>
<td>Intrinsic Carrier Concentration</td>
</tr>
<tr>
<td>( \text{two}_\phi _f ) (V)</td>
<td>Strong Inversion Surface Potential</td>
</tr>
<tr>
<td>( \text{CoxPrime/Cox} )</td>
<td>Oxide Capacitance per Area</td>
</tr>
<tr>
<td>( \text{VFB} ) (V)</td>
<td>Flatband Voltage</td>
</tr>
<tr>
<td>( V_{bi} ) (V)</td>
<td>Built-In Potential of Source/Drain-Bulk Junction</td>
</tr>
</tbody>
</table>

3.1.1.4 Effective Bulk-Source Voltage

The Effective Bulk-Source voltage is given as a way to prevent \( V_{BS} \) from going below a specified most negative value [6]. This specified most negative value is determined by specific conditions using the parameters \( K_1 \), \( K_2 \) and \( \text{two}_{\phi} \_f \). If \( V_{BS\_eff} \) were to become more negative than allowed, than unrealistic values and numerical problems would occur in following calculations.

Another check that is needed is to check and see if \( V_{BS\_eff} \) is greater than zero. This implies that the transistor is forward bias, which is a condition not normally encountered in MOSFETS [6]. If \( V_{BS\_eff} \) is greater then zero, then future equations need to be modified with a new calculation (see Appendix F and the special condition).
3.1.1.5 Depletion Thickness of the Gate-Induced Region

The Depletion Thickness (X\text{dep}) is used to account for the short channel effects in the calculation of the Threshold voltage (V_T) when there is a drain bias. Two equations are used to determine X\text{dep}, the first equation accounts for when V_{BS\_eff} is not equal to zero and subtracts the value of V_{BS\_eff} in the calculation of X\text{dep}. The second equation accounts for when V_{BS\_eff} is equal to zero. Though the equation looks like V_{BS\_eff} is simply removed, the equation is needed in future calculations, and the original equation account for V_{BS\_eff} cannot be used since V_{BS\_eff} will not be zero (see Appendix G).

3.1.1.6 Threshold Voltage Calculation

The Threshold voltage is calculated using the tried and true level-1 equation, which accounts for the Body Effect and by compensating for Narrow Channel Effects, Short Channel Effects and Lateral Nonuniform Doping [5].

In MOSFETS, the gate-induced depletion region will spread outside of the width-defined channel, which increases the depletion charge in narrow devices [5]. This increase causes a significant effect on the Threshold voltage. By adding a Narrow Channel Effects term (deltaVT\_narrow\_width) to the threshold voltage, this increase in the depletion charge can be accounted for. The term is removed for a non-narrow device by the fact the equation is inversely proportional to the Effective Width. As the width gets larger, thus less narrow, then the deltaVT\_narrow\_width term gets smaller and eventually goes to zero (see Appendix H).

Short Channel Effects are accounted for by subtracting a Short Channel Effects term from the Threshold voltage [5]. This Short Channel Effects term is broken up into subset terms of a Charge Sharing component and a Drain-Induced Barrier Lowering
(DIBL) component. The Charge Sharing is farther split into a similar charging sharing component and a small size component to help account for the Short Channel Effects more accurately (see Appendix H).

The Lateral Nonuniform Doping is accounted for by adding the effect caused by impurities in the doping process of the silicon material. As the channel length decreases of the device, the impurities, which cause some more heavily doped regions, cause the threshold voltage to increase [5].

3.1.1.7 Smoothing Function Effective Gate-Source Voltage

The Effective Gate-Source Voltage ($V_{GS_{eff}}$) accounts for the Poly-Depletion Effects that are caused by not being able to heavily dope the polysilicon gate all the way down to its interface with the gate oxide [5]. After calculating the $V_{GS}$, a smoothing function is introduced to even out the transition between the subthreshold region and the inversion regions. This transition occurs when $V_T$ is subtracted from $V_{GS_{eff}}$, i.e. when the transistor is turned on (see Appendix I).

3.1.1.8 Effective Mobility

To account for the reduction of mobility by the vertical field, three models are used to calculate the effect [5]. Two of these models are based on a more physically realistic evaluation of the mobility (see Appendix J). The third model is based on an average electric field description, such that the carriers in the inversion layer, regardless of their depth below the oxide-silicon interface, are influenced by an effective electric field [5]. Only one is used in the I-V model calculation, but that one is chosen based on the parameter MOBMOD.
3.1.2 Drain Current Model

The Drain Current Model is connected to the I-V model, but for this research separating it allows for a more specific representation of showing the drain current calculations and the effects of various quantities on the drain current.

3.1.2.1 Drain to Source Resistance

No transistor is perfect; if they were then the transistor would not dissipate any energy in the form of heat when turned on. By accounting for the drain to source resistance ($R_{DS}$), the correct current can be calculated. This resistance comes from the silicon-doped material having a resistivity of a specific ohm-cm, which the electrons travel through when the channel is opened up (i.e. the transistor turns on, see Appendix K).

3.1.2.2 Drain to Source Voltage

The drain to source voltage calculated in the saturation region is to be used in calculating the current in the saturation region ($V_{ds_{sat}}$). This voltage takes in effect the device characteristics in the saturation region along with taking in consideration the effects of $R_{DS}$ (see Appendix L).

By linking $V_{DS}$ and $V_{ds_{sat}}$ using the current model, a new factor is calculated to allow for a smooth transition between the regions of device operation. By substituting the Effective Drain to Source Voltage ($V_{ds_{eff}}$) into the drain current equation, a single equation to calculate $V_{DS}$ can be used resulting in a single equation to calculate the drain current with no discontinuities (see Appendix M).
3.1.2.3 Drain to Source Current

The Early Saturation Voltage ($V_{a, sat}$) (see Appendix R) must first be calculated, then after determining the effects of Channel Length Modulation ($V_{a, clm}$), Drain Induced Barrier Lowering ($V_{a, clm, dibl}$), and Substrate Current Induced Body Effect ($V_{a, scobe}$) (see Appendix N, O, and P) the factors can be combined together along with $V_{ds, sat}$ to calculate the Drain to Source current ($I_{DS}$). Each calculated quantity is added and subtracted from the drain current when the early voltage is equal to infinity (reference appendix Q for the correct setup).

3.1.3 Capacitance Model

The Capacitance Model (C-V) was not researched fully due to time constraints. Because of this, it is encouraged to reference either [5] or [6] or the many other Analog modeling books for information regarding the Capacitance-Voltage model.

3.2 UC Berkley Models

The UC Berkley Model is similar to the BSIM3 model with exceptions related to the level of depth to the equations that is used. The UC Berkley model is level -3, which by inspection, is not as accurate as the level - 11 BSIM3 (though both are developed by UC Berkley).

3.2.1 UC Berkley Parameters

The UC Berkley model requires less parameter’s to be determined, but these parameters are harder to obtain and most must be extracted from the technology being used. Table 6 list the parameters needed for the UC Berkley model [7].
Table 6: UC Berkley Parameters

<table>
<thead>
<tr>
<th>UC Berkley Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>1 = nmos, -1 = pmos</td>
</tr>
<tr>
<td>vt0</td>
<td>Threshold voltage of device</td>
</tr>
<tr>
<td>gamma</td>
<td>threshold factor</td>
</tr>
<tr>
<td>beta</td>
<td>Kn / 2, where Kn = un * Cox</td>
</tr>
<tr>
<td>lambda</td>
<td>Channel Length Modulation</td>
</tr>
<tr>
<td>vdsat</td>
<td>Saturation drain voltage</td>
</tr>
<tr>
<td>ld</td>
<td>Transistor length</td>
</tr>
<tr>
<td>wd</td>
<td>Transistor width</td>
</tr>
</tbody>
</table>

3.2.2 Threshold Voltage Calculation

Once the correct parameters are inputted in and the initial MATLAB file is ran, the correct threshold voltage must be computed. The threshold voltage can change based on temperature, bulk to source voltage, and with various other factors [8].

3.2.3 NMOS Computation

The next step is to compute the technology used to obtain the data needed to plot the drain current and gate to source voltages.

The amount of research done into the UC Berkley model is not comparable to the research performed for the BSIM3 model. This is due to the UC Berkley model not being as accurate (see results below) and the main focus of the analog modeling project towards the BSIM3 modeling. The UC Berkley model was researched a little due to the complexity of the BSIM3 model and mathematical problems encountered (see problems below).

3.3 Simulation Results

3.3.1 Cadence Simulation Results

Simulations results were obtained and compared to validate the BSIM3 model and UC Berkley model. The goal of the models themselves was to allow for constructing of
an analog circuit in MATLAB and allow MATLAB to perform the heavy mathematical computations in a hope to reduce the time to complete the simulations. Thus the analog model needed to be as accurate to its Cadence counterpart. Figure 60 shows the test simulation results in Cadence using a NMOS transistor of width 1 micrometer (μm) and length 0.13 micrometer (μm). The technology used is the IBM 130 nanometer (nm) process.

Figure 60 shows that at a drain to source voltage ($V_{DS}$) equal to 0.6 volts (V), the drain current ($I_D$) is equal to 68.27 microamps (mA) and at a $V_{DS}$ equal to 1.0 V, the $I_D$ is equal to 78.86 mA. The control setup used a gate to source voltage ($V_{GS}$) equal to 0.6 V with a device temperature of 25°C Celsius (C). These values became the test values to compare the BSIM3 and UC Berkeley model with.

![NMOS Transistor in Cadence](image-url)
3.3.2 UC Berkley Simulation Results

The UC Berkley model used required two parameters that were not specified in the parameter list for the IBM 130 nm technology. These two parameters were $K_n$ and $\Lambda$. It can be shown and proven in undergraduate classes that $\Lambda$ and $K_n$ depend on the width and length of the transistor. With that, various data points were needed to be determined to calculate $\Lambda$ and $K_n$ for various widths (length was kept constant for testing purposes). To achieve this, Cadence was used to test the model transistor at various widths with a length of 0.13 um. Figures 61, 62, 63, and 64 show various simulations plotting $I_D$ vs. $V_{DS}$ of different transistor widths and lengths.

Figure 61: $W_n = 2\mu m, L_n = 0.13\mu m$
Figure 62: Wn = 10um, Ln = 0.13um

Figure 63: Wn = 1um, Ln = 0.3um
Figure 64: Wn = 1um, Ln = 0.5um

Simulations were performed with Wn kept constant and Ln set equal to different lengths to show the relationship between Ln and the slope of the I_D vs. V_DS plot in the saturation region. The slope relates to lambda, which helps show the relationship between Ln and Lambda. These simulations were not used in the calculation of Lambda and Kn. From figures 60, 61, 62, test data was gathered at two points in the saturation region. To ensure the test points were in the saturation region, calculating the transition point was performed from the basic equation of:

$$V_{DS} \geq V_{GS} - V_{t0}$$  \hspace{1cm} (6)$$
Table 7: Lambda and Kn Test Points

<table>
<thead>
<tr>
<th>$V_{DS}$</th>
<th>$W_n = 1.0\mu m$</th>
<th>$W_n = 2.0\mu m$</th>
<th>$W_n = 10\mu m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6V</td>
<td>$I_D = 68.27\mu A$</td>
<td>$I_D = 134.20\mu A$</td>
<td>$I_D = 661.48\mu A$</td>
</tr>
<tr>
<td>1.0V</td>
<td>$I_D = 78.86\mu A$</td>
<td>$I_D = 150.51\mu A$</td>
<td>$I_D = 765.06\mu A$</td>
</tr>
</tbody>
</table>

Table 7 shows the test points determined from the simulation results. These values were then plugged into the basic equation for NMOS transistor:

$$I_D = \left(\frac{1}{2}\right) \times K_n \times \left(\frac{W}{L}\right) \times \left[V_{GS} - V_t\right]^2 \times \left[1 + \lambda \times V_{DS}\right]$$  \hspace{1cm} (7)

Plugging in both sets of values for each length, thus forming two equations with two unknowns, and then solving for $K_n$ and Lambda allowed for the test data points of $K_n$ and Lambda. Once determined, these points could be plotted and then have a linear fitted curve calculated to determine the different values of $K_n$ and Lambda needed for different values of $W_n$.

Figure 65 and 66 show the relationship of Lambda and $K_n$ to the $W_n$ of the transistor. The plots show that Lambda and $K_n$ decrease, as $W_n$ gets larger. This data was substituted into the UC Berkley model along with the rest of the parameters and simulations where able to be performed. Below, figure 67 shows the results at a width of 1 um and length of 130 nm
Figure 65: Kn vs. Wn

Figure 66: Lambda vs. Wn
Figure 67: NMOS Transistor using UC Berkley Model

Table 8: UC Berkley Simulation vs. Cadence Simulation

<table>
<thead>
<tr>
<th>V_{DS}</th>
<th>UC Berkley</th>
<th>Cadence</th>
<th>% error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6V</td>
<td>69.85 uA</td>
<td>68.27 uA</td>
<td>2.51 %</td>
</tr>
<tr>
<td>1.0V</td>
<td>80.85 uA</td>
<td>78.86 uA</td>
<td>2.52 %</td>
</tr>
</tbody>
</table>

Table 8 displays the correspondence between the UC Berkley simulation results and Cadence simulation results. The percent error for both measurements taken is over 2%, which is not accurate enough to achieve desirable results. Using a 2% error for just one transition is not too terrible, but using a 2% error for a microprocessor would cause unrealistic results. The results could not be used for designing the chip and thus making

76
the model pointless. This brings the need for a more accurate analog modeling technique, thus using the BSIM3 level-11 modeling is the next step in the process.

3.3.3 BSIM3 Simulation Results

Timing constraints were evident in the research involved with the BSIM3 model. These timing constraints were caused by another research project going on as well (BIST research from chapter 2). Due to these timing issues, the research is not complete on the BSIM3 model, but a stepping-stone was achieved to help future progress.

The BSIM3 model required numerous calculations and formulas to achieve a more accurate model than the UC Berkeley. With the BSIM3 model, various parameters were created to allow for user-defined parameters. These parameters allow the user to adjust the output of the equation to meet the accuracy needed. At this present time, no good simulation data was obtained for BSIM3 model.

3.4 Analog Modeling with MATLAB Problems and Solutions

Mentioned above, there was no good simulation data for the BSIM3 model obtained during the research. The most notable reason is due to time constraints and the need to focus on other research projects. This doesn't mean that the BSIM3 model doesn't work though. Appendix D through Appendix W obtains the MATLAB code that can be used as a stepping-stone to create the analog model.

One such problem is the deep mathematical troubleshooting and Engineering Physics that is required for the analog model equations. The need to be able to understand the transistor at the submicron level was a must, but the need to be able to apply the mathematical concepts (relies heavily on differential equations) proved to be the most cumbersome. One constant problem was that the DC current model would
always give 0 amps (A). This could be due to the user-defined parameters not matching up to provide realistic values.

Another problem that caused the DC current to be 0 A is that early on, the BSIM3 model was generating unrealistic threshold voltage values. The threshold voltage ($V_t$) equations would generate very large values, thus acting like the transistor would never turn on. By adjusting the deltaVT_charg_sharing term in the $V_t$ equation and adjusting K3 and K3B parameter (user-defined), $V_t$ became more realistic and comparable to the $V_{th}$.

The most notable problem that was encountered while troubleshooting was the fact that BSIM3 relies on user defined parameters to help achieve desirable results. The problem is that this requires time to converge on values for each parameter, and stated above, time was a constraint on this research project.

Progress to be achieved on the research is to calculate realistic values for the effective voltages. The effective voltages help smooth out the output waveform while transitioning to each operating region. Once again, user defined parameters will help resolve this problem, but due to time constraints and the deep math theory involve, this could not be achieved during the course of the research.

The final problem area that could not be addressed due to time constraints was the issue of binning together the transistor model. Binning is the process of combining together the transistors and various components to allow for a more complex circuit. The results would be the ability to design a circuit in MATLAB using the analog model generated. This issue alone is a complex issue that would need to be resolved with time.
On a final note, there is a team within MATLAB that is working on creating an analog model environment. At this present time, it is unknown what the progress is.
4.0 Recommendations and Conclusion

4.1 Recommendations

4.1.1 BIST Scheme Recommendations

The performance of the BIST hardware can be improved by noticing the simple little things that prohibit the overall bandwidth BW and performance. The major problem associated with the BW of the output test signal can be contributed to the DC block capacitors used in the 8-bit DAC Buffer design. These capacitors allow for the correct DC offset voltage to be utilized for proper biasing of the input transistors of the buffer. Yet both DC block capacitors restrict the overall BW of the frequency response (see figure 31). By finding a new way to perform the DC offset correction, the BW can be increased to exceed 1GHz. One way to perform this correction is to design another output stage buffer that will use the DC offset from the buffer design in the 8-bit DAC and then output a DC offset of 200mV used for the Flash ADC CUT. This can be done with an active load amp, due to the amplifier benefits of high BW and low gain. From the simulations of the final buffer design, the output DC offset of the final buffer design without the offset correction is 313mV. This DC offset is not too low of a voltage and could easily be attained using an output buffer that could then be design to output a DC offset of 200mV. The only draw back will be redesigning the initial analog buffer from this thesis to ensure the 240mV output DR.

Another issue is the output DR of the buffer from the 8-bit DAC. The output DR was needed to be 240mV throughout the entire BW of the buffer. Due to the low input

80
signal of the buffer from the DAC, the buffer is actually an amplifier with minimal gain. Due to this gain, a high BW is relatively hard to achieve due the inverse relationship of gain to BW of any amplifier. Increase in gain causes a decrease in BW. The problem can be remedy some by changing the resister on the output of the DAC, but before the input of the buffer. The resistor on the output of the DAC and before the input of the buffer converts the current output signal of the current steering arrays to a voltage for the buffer [2]. The resistor was reduced in value to help address the large capacitive loads from the buffer on the current steering arrays [2]. By doing this, the input DR of the buffer was decreased in the process. Thus the low level signal inputted into the buffer (see figure 5). The final buffer design from the modified 8-bit DAC allows for a greater input DR (150mV). From this the solution of increasing the resistor value was utilized and the correct output DR was obtained. Yet the buffer has a BW of just greater than 900MHz, by adjusting the resistor even more and modifying the buffer even more, a 1GHz BW should be attainable.

The most time consuming problem that was ran into was the limitations of the simulator used to perform the analog and AMS simulations on the 8-bit DAC. Numerous times were encountered that the simulator would fail due to an error unrelated to the actual circuit, and with each simulation taking about one day to complete, this caused the most damaged to the time involved in the research.

The other problem associated with the simulator is seen when calculating the SFDR of the 8-bit DAC. There appears to be differences in the calculated SFDR from the data generated and the plotted data that allows for visual interpretation of the SFDR. From [2], a test signal of 312.5MHz produced a SFDR of 25.28dB and using a test signal
of 416.67MHz produced a SFDR of 19.88dB. This is considerably higher than the
calculated SFDR of comparable frequencies of the modified DAC used in the BIST test
setup. The reason for this is believed to be due to the settle time of the two different 8-bit
DAC's. The original DAC required less time to converge on an output then the modified
DAC. From this, the Cadence Virtuoso specification driven environment (VSDE)
simulator takes in account the entire time frame that the simulation is ran for. Since the
modified DAC requires more time to converge on an output signal, this extra time shows
more noise and unstable signals till the convergence that then gets calculated in the
overall SFDR from the VSDE simulator. To remedy this problem, a longer simulation is
required to perform the test, but due to disc quota errors, this could not be performed.
Using a higher order FFT would help as well to generate true numbers. Removing the
DC blocking capacitor is believed to help with this performance specification as well.

The final problem encountered during the course of this research is the timing
issues in the BIST test setup. The BIST test setup uses an ideal ADC to generate the
digitized inputs for the 8-bit DAC. The ideal ADC is required to have a sampling
frequency of 10 times the value of the analog test signal desired. From this, the data from
the ideal ADC will output data at a rate 10 times faster than the desired analog signal. If
the test signal is not a 125MHz signal, the DAC will not receive the data at the correct
rate of 1.25GHz from the ideal ADC. An 8-bit shift register was implemented to remedy
this problem for a 250MHz test signal, but further refinement is necessary for higher
frequencies. The planned solution is to implement the signal generator in Verilog-AMS
and manually input the digitized data into the signal generator from the ideal ADC. At
this present time, this solution could not be implemented. Once the BIST scheme is

82
implemented with an on-chip ROM, timing issues will not be a factor due to setting the output timing of the ROM chip at 1.25GHz and setting up the 8-bit shift register to load the data into the DAC at 1.25GHz.

4.1.2 Analog Modeling with MATLAB Recommendations

As stated in chapter 3, the most notable problem with the BSIM3 analog model is the user-defined parameters. Though with these parameters, the model can be simulated with pinpoint accuracy. Yet with these parameters, iteration after iteration would be required per parameter to converge on a value for the specific user defined parameter. The problem requires time to solve and many mathematical formulas to help converge the required value. The easiest solution to this problem is to just pick a value and cross your fingers, but this proves to hinder the convergence of the actual quantities that are calculated in the MATLAB code. These quantities (reference chapter 3) are the values used in the mathematical equations to model the physical behavior of the transistor (i.e. threshold value). From this, unrealistic values will be calculated and then this would lead to a model that outputs 0A of current. Narrowing down the formula’s that use the user defined parameters, then setting restrictions on the upper and lower bounds based on the realistic values of the transistor can help to pin point the user defined parameter values. This approach basically forces the user-defined parameters to converge on a specific value; the problem is that by “guessing” what the quantity is in the equation, this leads to inaccurate results. Yet this approach could decrease the number if iterations as stated above that would be required for each user-defined parameter.

The most important problem with the analog model is the required time to complete the project and the overwhelming work that would be required. Thus stated in
chapter 3, there is a team within MATLAB that is developing a similar project. Once completed, some minor attachments could be made to allow for the goals of this project to be completed (most notable the ability to have an efficient design environment).

4.2 Conclusion

In conclusion, a BIST scheme is presented that showcases the ability to perform fault detection of the CUT. The new buffer circuit allows for high frequency test signals from the DAC along with utilizing a 1.2V VDD instead of both 1.2V and 1.4V. Further improvements can be made to remove the 1.4V VDD dependence completely by some simple simulations and design modifications to allow the bias circuit for the DAC to output the correct voltage using the 1.2V VDD. By doing this, power consumption will be reduced even more. A new analog buffer was designed that allows for greater BW and less power dissipation to allow for upwards of 900MHz analog test signal. This new buffer allows for the output signal to be a constant value across the entire BW with only a −3dB BW just greater than 900MHz. Proof of concept was shown that fault detection can be seen using a first order EΔ modulator to digitize the output test point of the CUT. This difference then allows for bit-wise comparison with a correctly working digital bit stream to allow for the fault detection. Used in conjunction with the multiplexer allows for some diagnosis and ease of trouble-shooting. Simulations show that test signals can be generated in excess of 416.67MHz using the mixed-signal environment and that theoretically test signals can exceed 900MHz. Limitations within the simulation environment prohibited further testing of high frequency signals.

An approach was discussed to perform analog modeling with MATLAB along with the problems and possible solutions. Code is including that utilizes the BSIM3
analog model and that can be used to help further any research on analog modeling with MATLAB.

The goal overall is that this research helps future research to improve and achieve even greater results for the ingenuity and enhancement of our field of study. Though the world is moving towards digital, it will always be analog in nature.
Appendix A

MATLAB Ideal 8-Bit ADC

% test of an ideal ADC
clc;
clear;

% frequency of digitized signal
freq = 1e9;

% sample frequency
freq_s = 18e9;

% length of sampling
time_end = 5e-9;

% length of sampling
theta = (1e-13):(1e-13):time_end;
k = length(theta);

% input signal
f = sin(2*pi*(freq)*theta) + 1;

% clock
clk = square(2*pi*(freq_s)*theta);

% Ideal ADC
vref = 1;

for i = (1:8)
    b(i) = 0;
end

d_half_range = vref / 2;
d_vin = 0;
j = 1;

counter = 0;
for t = (2:k)
    if (clk(t) == 1) & (clk(t-1) == -1)
        sample(j) = t;
        j = j + 1;
        counter = counter + 1;
    end
end
d_vin = f(i);
for i = (1:8)
    if d_vin > d_half_range
        b(i) = 1;
        d_vin = d_vin - d_half_range;
    else
        b(i) = 0;
        d_vin = d_vin * 2;
    end
end

%reverse order of bits
    bit7(counter) = b(1);
    bit6(counter) = b(2);
    bit5(counter) = b(3);
    bit4(counter) = b(4);
    bit3(counter) = b(5);
    bit2(counter) = b(6);
    bit1(counter) = b(7);
    bit0(counter) = b(8);
end

%plot input signal and clock
plot(theta,f,'g', theta,clk,'b')
Appendix B

VerilogAMS Ideal 8-Bit ADC Version 1

//Verilog-AMS HDL for "test_signal", "ideal_ADC", "verilogams"

`include "constants.vams"
`include "disciplines.vams"
timescale 1ns/100ps

module ideal_ADC (analogSignal, clk, bit7, bit6, bit5, bit4, bit3, bit2, bit1, bit0);

    output bit7, bit6, bit5, bit4, bit3, bit2, bit1, bit0;
    input analogSignal, clk;
    electrical analogSignal;
    logic clk;
    logic bit7, bit6, bit5, bit4, bit3, bit2, bit1, bit0;

    parameter real vref = 1.0;

    real d_vin;
    real d_half_range;
    reg[7:0] b;
    integer i;

    assign bit7 = b[7];
    assign bit6 = b[6];
    assign bit5 = b[5];
    assign bit4 = b[4];
    assign bit3 = b[3];
    assign bit2 = b[2];
    assign bit1 = b[1];
    assign bit0 = b[0];

    initial begin
        b = 0;
        d_half_range = vref / 2;
        d_vin = 0;
    end

    always @(posedge clk) begin
        d_vin = V(analogSignal);
        for (i=7, i>=0, i=i-1) begin
            if (d_vin > d_half_range) begin
                b[i] = 1;
            end
        end
    end

endmodule
d_vin = d_vin - d_half_range;
end else b[i] = 0;
d_vin = d_vin * 2;
e
end
endmodule
Appendix C

VerilogAMS Ideal 8-Bit ADC Version 2

//Verilog-AMS HDL for "testlib", "adc_8_bit", "verilogams"

`include "constants.vams"
`include "disciplines.vams"
`timescale 1ns/100ps

module adc_8_bit (analogSignal, clk, bit7, bit6, bit5, bit4, bit3, bit2, bit1, bit0);

    output bit7, bit6, bit5, bit4, bit3, bit2, bit1, bit0;
    input analogSignal, clk;
    electrical analogSignal;
    logic clk;
    electrical bit7, bit6, bit5, bit4, bit3, bit2, bit1, bit0;

    parameter real vref = 0.4;
    parameter real vdd = 1.2;

    real d_vin;
    real d_half_range;
    // reg[7:0] b;
    real b[7:0];
    integer i;

    //assign bit7 = b[7];
    //assign bit6 = b[6];
    //assign bit5 = b[5];
    //assign bit4 = b[4];
    //assign bit3 = b[3];
    //assign bit2 = b[2];
    //assign bit1 = b[1];
    //assign bit0 = b[0];

    initial begin
        //b = 0;
        d_half_range = vref / 2;
        d_vin = 0;
    end

    always @(posedge clk) begin
        d_vin = V(analogSignal);
        for (i=7; i>=0; i=i-1) begin
            
        end

    end
if (d_vin > d_half_range) begin
    b[i] = vdd;
    d_vin = d_vin - d_half_range;
end else b[i] = 0;
    d_vin = d_vin * 2;
end

analog begin
    V(b7) <+ b[7];
    V(b6) <+ b[6];
    V(b5) <+ b[5];
    V(b4) <+ b[4];
    V(b3) <+ b[3];
    V(b2) <+ b[2];
    V(b1) <+ b[1];
    V(b0) <+ b[0];
end
endmodule
Appendix D

MATLAB Effective Channel Length and Width

%-----------------------------------------------

%Effective Channel Length and Width for I-V

deltaL = (LL/(L^LLN)) + (LW/(W^LWN)) + (WL/(L^LLN)*(W^LWN));
deltaW = (WL/(L^WLN)) + (WW/(W^WWN)) + (WWL/(L^WLN)*(W^WWN));
%C.208
% deltaW_bias_dependency = DWG*Vgsteff + DWB*[sqrt(phiS... % - Vbsceff) - sqrt(phiS)];
deltaW_bias_dependency = 0.0;

L_eff = L - 2*LINT - 2*deltaL; %equation 11.34 pg. 389
W_eff = W - 2*WINT - 2*deltaW - 2*deltaW_bias_dependency; %equation 11.35 pg. 389
WprimeEff = W - 2*WINT - 2*deltaW; %C.206

%check if W_eff is negative and replace
if W_eff < 0.02e-6
    W_eff = (2e-8)*((4e-8)-W_eff) / ((6e-8)-2*W_eff);
else
    W_eff = W_eff;
end

%-----------------------------------------------

%Effective Channel Length and Width for C-V

deltaL_CV = (LLC/(L^LLN)) + (LWC/(W^LWN)) + (LWLC/(L^LLN)*(W^LWN));
deltaW_CV = (WLC/(L^WLN)) + (WWC/(W^WWN)) + (WWLC/(L^WLN)*(W^WWN));

L_eff_CV = L - 2*DLC - 2*deltaL_CV;
W_eff_CV = W - 2*DWC - 2*deltaW_CV;

%-----------------------------------------------
Appendix E

MATLAB Quantity Calculation

%--------------------------------------------------

%Quantity calculation
q = 1.60212e-19; %charge of electron
k = 1.3806226e-23; %temperature constant
epsilonOX = (8.85e-12) * 3.9018;
epsilonoS = (8.85e-12) * 11.7055;

T = T_device + 273.15;

%Silicon band gap energy value (eV)
Eg_T = 1.160 - (((7.02e-4)*(T^2))/(T + 1108));

%Intrinsic carrier concentration (q*Eg_T*TNOM)
n_i = (1.45e10)*((T/300.15)^(3/2))*((exp(21.5565981 - ((q/... (2*k*(TNOM+273.15)))*Eg_T)));

%Strong Inversion Surface potential (V)
%it's different for CAPMOD = 3 (see pg. 166, MOSFET Models)
two_phi_f = 2*((k*(TNOM + 273.15))/q)*((log(NCH/(n_i)));

%Oxide capacitance per area
%Note: Cox value looks better than CoxPrime
CoxPrime = epsilonOX / TOX;
Cox = Weff.CV * Leff.CV * CoxPrime;

%Flatband voltage (default is -1V)
VFB = deltaNP*VTH0 - two_phi_f - K1*(sqrt(two_phi_f));

%VTH0*(Vbs) - two_phi_f - K1*(sqrt(two_phi_f));

%--------------------------------------------------

%Built-in potential of source/drain-bulk junction
V_bi = ((k*(TNOM + 273.15))/q)*((log((NCH*NDS)/(n_i^2)));

if V_bi < 0.01
  V_bi = 0.01;
else
  V_bi = V_bi;
end
$V_{bi} - \text{two}_{\phi_i f}$; \%must be positive for practical devices

%-----------------------------------------------
Appendix F

MATLAB Effective Bulk-Source Voltage

%Reference Liu pg. 257

%This is to prevent the device's Vbs from going below a specified most negative value (Liu pg. 275 and 257)

Vx = 0.9 * (two_phi_f - ((K1^2)/(4*(K2^2))));

if Vx > (-3)
    v_bc = (-3);
elseif Vx < (-30)
    v_bc = (-30);
else
    v_bc = Vx;
end

%delta_symbol1 = 0.001;
delta_symbol1 = 0.02;

%commented sections for test purposes
%shows as Vbs is more negative, Vbs = Vbc
%as Vbs is in practical range (Vbc < Vbs < 0), Vbs = Vbs_eff

% Vbs = -15:0.1:0;
% test_bound = length(Vbs);
% (test_i)

% for test_i = 1:test_bound
    Vbs_eff = v_bc + 0.5*(Vbs - v_bc - delta_symbol1 + ...
        sqrt(((Vbs - v_bc - delta_symbol1)^2) - (4*delta_symbol1*v_bc)));
% end

% plot (Vbs,Vbs_eff)
% xlabel('Vbs')
% ylabel('Vbs_eff')
%Special condition (see Liu pg. 476)
if Vbs_eff > 0
    replace_condition = ((sqrt(two_phi_f))^3)/(two_phi_f + (Vbs_eff/2));
else
    replace_condition = (sqrt(two_phi_f - Vbs_eff));
end
Appendix G

MATLAB Depletion Thickness

%---------------------------------------------------------------
%Both Xdep need to be positive

%Depletion thickness in bulk
Xdep = sqrt((2*epsilonS*(two_phi_f - Vbs_eff))/(q*NCH));

%Depletion thickness in bulk at zero Vbs_eff
Xdep0 = sqrt((2*epsilonS*two_phi_f)/(q*NCH));

%---------------------------------------------------------------
Appendix H

MATLAB Threshold Voltage

%----------------------------------------------------------------------------------------

% Threshold voltage
% Need to double check deltaVT Charg_sharing
%----------------------------------------------------------------------------------------

% Also called L'd, DIBL
Lt0 = sqrt((epsilonS*Xdep0)/CoxPrime);
%----------------------------------------------------------------------------------------

% Characteristic Length, also called L't, CS
if (DVT2 * Vbs_eff) >= (-0.5)
    Lt = (sqrt((epsilonS*Xdep)/CoxPrime))*(1 + DVT2*Vbs_eff);
else
    Lt = (sqrt((epsilonS*Xdep)/CoxPrime))*((1 + 3*DVT2*Vbs_eff)/...
        (3 + 8*DVT2*Vbs_eff));
end
%----------------------------------------------------------------------------------------

% Characteristic Length for the added width, also called L't, CS, W
if (DVT2W * Vbs_eff) >= (-0.5)
    Ltw = (sqrt((epsilonS*Xdep)/CoxPrime))*(1 + DVT2W*Vbs_eff);
else
    Ltw = (sqrt((epsilonS*Xdep)/CoxPrime))*((1 + 3*DVT2W*Vbs_eff)/...
        (3 + 8*DVT2W*Vbs_eff));
end
%----------------------------------------------------------------------------------------

% Does not depend on channel Length
% Causes VT to increase implies should be positive
deltaVT_body_effect = K1*((TOX/TOXM)*((replace_condition) - ...
    (sqrt(two_phi_f)))- K2*(TOX/TOXM)*Vbs_eff;
%----------------------------------------------------------------------------------------

% Causes VT to decrease
% Should be positive (is subtracted in final VT equation)
% As L'eff increases, deltaVT Charg_sharing decreases
% Approaches 0 for long channel device because charge sharring is more
% prominent in short channel devices
%DVT0 scales the value based on the technology used
%deltaVT_charge_sharing (-Vbs) > deltaVT_charge_sharing (Vbs=0)
deltaVT_charge_sharing = DVT0*(((exp((-DVT1*(Leff))/(2*Lt))) + (2*(exp((-DVT1*(-Leff))/Ltw)))*V_bi - two_phi_f);

%---------------------------------------------------------------
%Should always be positive so that Vds large is implies a smaller VT at
%Vds = 0
%Should be 0 for long channel device and increase for short channel
%deltaVT_DiBL(-Vbs) > deltaVT_DiBL(Vbs = 0)
%DSUB is used to control the amount of deltaVT_DiBL as a function of Leff
%ETA0 is a proportional constant to scale the DIBL value for the
%threshold correction in the absence of Vbs
%ETAB is used to correct for the Vbs dependence

deltaVT_DiBL_tmp = (ETA0 + ETAB*Vbs_eff)
if deltaVT_DiBL_tmp >= (10^-4)
    deltaVT_DiBL = (exp(-DSUB*(Leff/(2*Lt0))) + 2*exp(-DSUB*(Leff/Lt0))))... - deltaVT_DiBL_tmp*Vds
else
    deltaVT_DiBL = (exp(-DSUB*(Leff/(2*Lt0))) + 2*exp(-DSUB*(Leff/Lt0))))... - Vds*((2*(10^-4)) - deltaVT_DiBL_tmp)/(3 - (2*(10^-4))... - deltaVT_DiBL_tmp);
end

%---------------------------------------------------------------
%Also labeled as V_LND
%As Leff decrease, VT increase, this implies that deltaVT_reverse_ %short_channel should be positive and also increases as Leff decreases

deltaVT_reverse_short_channel = K1*(TOX/TOXM)*((sqrt(1 + (NLX/(Leff))))... - 1)*(sqrt(two_phi_f));

%---------------------------------------------------------------
%Also labeled as V_NCE
%Two math equations for V_NCE: pg. 399, Foyt
% 3*pi*(TOX/Weff)*two_phi_f
% (pi*q*NSUB*(Xdep^2))/(2*CoxPrime*Weff)
%Can change K3 and K3B and W0 to optimize simulation
deltaVT_narrow_width = (K3 + K3b*Vbs_eff)*(TOX/(Weff + W0))*two_phi_f;

%---------------------------------------------------------------

%Note: formula has deltaVT_small_size as "-" instead of "+"
deltaVT_small_size = DVT0W*(exp((-DVT1W*((Weff*Leff)/(2*Ltw)))) + ... (2*(exp((-DVT1W*((Weff*Leff)/Ltw)))))*V_bi - two_phi_f);
%---------------------------------------------------------------------
% Removed deltaVT_charg_sharing and replaced "+" with "-" in front of
% deltaVT_small_size. This produces good looking VT values
% Foyt, pg. 398, possible two equations for deltaVT_charg_sharing.
% - deltaVT_charg_sharing

% Original formula:
% VT = VTH0 + deltaNP*(deltaVT_body_effect - deltaVT_charg_sharing ...
% - deltaVT_DIBL + deltaVT_reverse_short_channel ...
% + deltaVT_narrow_width + deltaVT_small_size)

% Revised formula with ". deltaVT_charg_sharing removed" and
% "deltaVT_small_size subtracted from VTH0
VT = VTH0 + deltaNP*(deltaVT_body_effect - deltaVT_DIBL ...
  + deltaVT_reverse_short_channel + deltaVT_narrow_width ...
  - deltaVT_small_size)

% Used for testing purpose
% for y = (1:bound)
% VT(y) = VTH0;
% end

%---------------------------------------------------------------------
Appendix I

MATLAB Effective Gate-To-Source Voltage

%-------------------------------------------------------------------------

%Effective gate-to-source votage (Vgs_eff)
%accounts for Poly-Depletion Effects

delta_symbol2 = 0.05;

Vpoly = ((q*epsilonS*NGATE*(CoxPrime^2)*(10^6))/2)*...
(((sqrt(1 + ((2*(Vgs - VFB - two_phi_f))/...
(q*epsilonS*NGATE*(CoxPrime^2)*(10^6)))) - 1)^2);

% Vpoly = ((q*epsilonS*NGATE*(CoxPrime^2)))*...  
%  (((sqrt(1 + ((2*(Vgs - VFB - two_phi_f))/...
%  (q*epsilonS*NGATE*(CoxPrime^2)))) - 1));

% Vpoly_eff = 1.12 - (0.5*(1.12 - Vpoly - delta_symbol2 + ...
%  (sqrt(((1.12 - Vpoly - delta_symbol2)^2) + (4*delta_symbol2*1.12))));

% Vgs_eff = Vgs - Vpoly_eff;
Vgs_eff = VFB + two_phi_f + Vpoly;

%-------------------------------------------------------------------------
Appendix J

MATLAB Effective Channel Mobility

%---------------------------------------------------------

%channel mobility (u_eff)

if MOBMOD == 1
    for g = (1:bound)
        u_tmp(g) = (UA + UC*Vbs_eff)*((Vgst_eff + 2*VT(g))/TOX) + ... 
                   UB*(((Vgst_eff + 2*VT(g))/TOX)^2);
    end
elseif MOBMOD == 2
    for g = (1:bound)
        u_tmp(g) = (UA + UC*Vbs_eff)*((Vgst_eff/TOX) + UB*((Vgst_eff/TOX)^2);
    end
elseif MOBMOD == 3
    for g = (1:bound)
        u_tmp(g) = (UA*((Vgst_eff + 2*VT(g))/TOX) + UB*((Vgst_eff + 2*VT(g))/TOX)^2)) * 
                   (1 + UC*Vbs_eff);
    end
else
    errorMessage_001 = 'MOBMOD can not be greater than 3 and less then 1'
end

if u_tmp >= (-0.8)
    u_denominator = 1 + u_tmp;
else
    u_denominator = (0.6 + u_tmp)/(7 + 10*u_tmp);
end

for k = (1:bound)
    u_eff(k) = U0 / u_denominator(k);
end

%---------------------------------------------------------
Appendix K

MATLAB Drain to Source Resistance

%------------------------------------------------------

% Rds

if RDSW < 0
    Rds = 0;
else
    Rds_tmp = PRWG*Vgst_eff + PRWB*(replace_condition - sqrt(two_phi_f));
    if Rds_tmp >= (-0.9)
        Rds = (RDSW/((10^6)*WprimeEff^WR))* (1 + Rds_tmp);
    else
        Rds = (RDSW/(((10^6)*WprimeEff^WR)) * ...
            (1 + (0.8*Rds_tmp)/(17 + 20*Rds_tmp)))
    end
end

%------------------------------------------------------
Appendix L

MATLAB Drain to Source Voltage

%-------------------------------------------------------

%Drain-source saturation voltage (Vds_sat)

%ensures A2 makes sense
if A2 < 0.01
  A2 = 0.01;
elseif A2 > 1
  A2 = 1;
  A1 = 0;
else
  A2 = A2;
  A1 = A1;
end

%ensure 0 < lambda < 1
delta_symbol3 = 0.001;
if A1 > 0
  lambda_tmp = (1 - A2) - (0.5*(1 - A2 - A1*Vgst_eff - delta_symbol3 + ...
         sqrt(((1 - A2 - A1*Vgst_eff - delta_symbol3)^2) + ...
         4*delta_symbol3*(1 - A2)))
  lambda = A2 + lambda_tmp;
% lambda = A2 + A1*Vgst_eff;
else
  A1_prime = (-1)*A1;
  lambda_tmp = A2 - (0.5*(A2 - A1_prime*Vgst_eff - delta_symbol3 + ...
         sqrt(((A2 - A1_prime*Vgst_eff - delta_symbol3)^2) + ...
         4*delta_symbol3*A2)))
  lambda = A2 - lambda_tmp;
end

for l = (1:bound)
  Esat(l) = (2*VSAT) / u_eff(l);
end

if Rds == 0
  for y = (1:bound)
    Vds_sat(y) = (Esat*Leff*(Vgst_eff + ((2*k*T)/q)))/...
               (Abulk*Esat*Leff + (Vgst_eff + ((2*k*T)/q)));
  end
else
end

104
a = (Abulk^2)*Weff*VSAT*CoxPrime*Rds + ((1/lambda) - 1)*Abulk;
for m = (1:bound)
    b(m) = -(Vgst_eff + ((2*k*T)/q))*((2/lambda) - 1) + Abulk*(Esat(m))*Leff...
    + 3*Abulk*(Vgst_eff + ((2*k*T)/q))*Weff*VSAT*CoxPrime*Rds;
end
for p = (1:bound)
    c(p) = (Vgst_eff + ((2*k*T)/q))*(Esat(p))*Leff + 2*((Vgst_eff + ((2*k*T)/q))^2)... 
    *Weff*VSAT*CoxPrime*Rds;
end
for r = (1:bound)
    Vds_sat(r) = ((-b(r)) - sqrt(((b(r))^2) - (4*a*c(r))))/(2*a);
end
end

%-----------------------------------------------

105
Appendix M

MATLAB Effective Drain to Source Voltage

%-----------------------------------------------------

%Effective Vds (Vds_eff)

for h = (1:bound)
    Vds_eff(h) = Vds_sat(h) - (0.5*(Vds_sat(h) - Vds(h) - DELTA + ... 
                               sqrt(((Vds_sat(h) - Vds(h) - DELTA)^2) + 4*DELTA*Vds_sat(h))));
end

%-----------------------------------------------------
Appendix N

MATLAB Channel Length Modulation Effects

%-------------------------------------------------------

%Early voltage due to channel length modulation (Va_clm)

Lltl = sqrt((epsilonS/epsilonOX)*TOX*XJ);

if PCLM <= 0
    errorMessage002 = 'PCLM cannot be less than or equal to 0'
else
    PCLM = PCLM;
end

if (Vds - Vds_eff) > (10^-(-10))
    Va_clm = ((Abulk*Esat*L_eff + Vgst_eff)/(PCLM*Abulk*Esat*Litl))*...
              (Vds - Vds_eff);
else
    Va_clm = 5.834617425e14;
end

%-------------------------------------------------------
Appendix O

MATLAB Drain Induced Barrier Lowering Effects

%-------------------------------------------------------------

%Early voltage due to drain induced barrier lowering (V_a_clm_dibl)

theta_Rout = PDIBLC1*((exp(-(DROUT*Leff)/(2*Ll))) + (2*(exp(-(DROUT*Leff)...
/Ll)))) + PDIBLC2;

if (PDIBLCB*Vbs_eff) >= (-0.9)
    V_a_dibl_replace_condition = 1/(1 + PDIBLCB*Vbs_eff);
else
    V_a_dibl_replace_condition = (17 + 20*PDIBLCB*Vbs_eff)/...
        (0.8 + PDIBLCB*Vbs_eff);
end

if theta_Rout >= 0
    V_a_dibl = (((Vgst_eff + ((2*k*T)/q))/(theta_Rout))*...
        V_a_dibl_replace_condition)*(1 - ((Abulk*Vds_sat)/...
        (Abulk*Vds_sat + Vgst_eff + ((2*k*T)/q))));
else
    V_a_dibl = 5.834617425e14;
end

for w = (1:bound)
    V_a_clm_dibl(w) = (1 + ((PVAG*Vgst_eff)/(Esat(w)*Leff))*)((1/V_a_clm(w)) + ...
        (1/V_a_dibl))(-1));
end

%-------------------------------------------------------------
Appendix P

MATLAB Substrate Current Induced Body Effect

%%%--------------------------------------------------------

%%Early voltage due to substrate current induced body effect (Va_scbe)

if PSCBE2 > 0
    for i = (1:bound)
        Va_scbe(i) = (Leff/PSCBE2)*(exp((PSCBE1*Litl)/(Vds(i) - Vds_eff(i))));
    end
else
    for i = (1:bound)
        Va_scbe(i) = 5.834617425e14;
    end
end

%%%--------------------------------------------------------
Appendix Q

MATLAB Drain to Source Current Model

%---------------------------------------------------------------

%Drain-source current when early voltage is infinity (lds0)

for x = (1:bound)
    lds0(x) = (Weff*u_eff(x)*CoxPrime*Vgst_eff)/(Leff*(1 + (Vds_eff(x)/(Esat(x)*L eff))))*... 
                (1 - ((Abulk*Vds_eff(x))/(2*(Vgst_eff + ((2*k*T)/q)))))*Vds_eff(x);
end

%---------------------------------------------------------------

%Drain-source current (lds) and small-signal conductance (gds) for j = (1:bound)

lds(j) = (lds0(j))/(1 + ((Rds*lds0(j))/Vds_eff(j)))*(1 + ((Vds(j) - Vds_eff(j))/Va(j)))*... 
                (1 + ((Vds(j) - Vds_eff(j))/Va_scbe(j)));
end

%---------------------------------------------------------------
Appendix R

MATLAB Early Voltage at the Saturation Voltage

%---------------------------------------------------------------

%Early voltage at the saturation voltage (Va_sat)

for t = (1:bound)
    Va_sat(t) = (Esat(t)*Leff + Vds_sat(t) + 2*Rds*VSAT*CoxPrime*Weff*Vgst_eff/... 
        ((2/lambda) - 1 + Rds*VSAT*CoxPrime*Weff*Abulk))*... 
        (1 - ((Abulk*Vds_sat(t))/(2*(Vgst_eff + ((2*k*T)/q)))));
end

%---------------------------------------------------------------
Appendix S

MATLAB Subthreshold Ideality Slope Factor

%-------------------------------------------------------------

%Subthreshold ideality slope factor (n)

CdePPrime = epsilonS / Xdep;

n_tmp = NFACTOR*(CdePPrime/CoxPrime) + (CIT/CoxPrime) + ...
((CDSC + (CDSCD*Vds) + (CDSCB*Vbs_eff))/CoxPrime)*...
((exp(-DVT1*(Leff/(2*Lt)))) + (2*exp(-DVT1*(Leff/Lt))));

if n_tmp >= (-0.5)
    n = 1 + n_tmp;
else
    n = (1 + 3*n_tmp)/(3 + 8*n_tmp);
end

% n = mean(n1);

%-------------------------------------------------------------
Appendix T

MATLAB Early Voltage

%A---------------------------------------

%AEarly voltage (Va)

Va = Va_sat + Va_clm_dibl;

%A---------------------------------------
Appendix U

MATLAB Bulk Charge Factor

%-------------------------------------------------------------------------

%Bulk charge factor (Abulk)

Abulk_tmp = (1 + (K1/(2*replace_condition))*(TOX/TOXM)*((((A0*Left)/...
(Left + 2*sqrt(XJ*Xdep)))*(1 - AGS*Vgst_eff*...
((Left/(Left + 2*sqrt(XJ*Xdep)))/2)) + (B0/(Weff + B1))));

%ensure (1/(1 + KETA*Vbs_eff)) doesn't fall below -0.9
%which is unphysical
Abulk_condition = (1/(1 + KETA*Vbs_eff));
if Abulk_condition >= (-0.9)
    Abulk_condition = Abulk_condition;
else
    Abulk_condition = (17 + 20*KETA*Vbs_eff)/(0.8 + KETA*Vbs_eff);
end

%ensures Abulk has a real physical meaningful value
if Abulk_tmp >= (0.1)
    Abulk = Abulk_tmp * Abulk_condition;
else
    Abulk = ((0.2 - Abulk_tmp)/(3 - 20*Abulk_tmp)) * Abulk_condition;
end

%-------------------------------------------------------------------------
Appendix V

MATLAB Effective Gate to Source Voltage

%-----------------------------------------------

%Smothing function (Vgst_eff)
%smooths out subthreshold and inversion regions (Vgs_eff - VT)

CdepPrime0 = epsilonS / Xdep0;

Vgst_eff = (2*((n*k*T)/q)*log(1 + exp((Vgs_eff - VT)/(2*((n*k*T)/q)))))/...
(1 + 2*n*(CoxPrime/CdepPrime0)*(exp(-(Vgs_eff - VT - 2*VOFF)/...
((2*n*k*T)/q)));

%-----------------------------------------------
Appendix W

MATLAB NMOS Transistor Model

%Main nmos script to model nmos in bsim3v3
%using IBM 0.13u tech.
%Only need to run this script to calculate model

clear;
clc;

%-----------------------------------------------------------

%Design specs
Vgs = 0.6;
Vdd = 1.2
Vss = 0
W = 1e-6;
L = 0.13e-6;
Vbs = 0.0;
MOBMOD = 2; %to choose Mobility u_eff, pick value 1, 2, or 3
% CLoad = 500e-15;
% Rin = 50;
T_device = 25; %operating temperate in Celcius
deltaNP = 1; %1 for nmos, (-1) for pmos

%-----------------------------------------------------------

%sweep Vds
Vds = 0.001:0.1:(Vdd - Vss);
bound = length(Vds);

%-----------------------------------------------------------

run nmos_4_parameters
run nmos_4_effective_W_L
run nmos_4_quantity_calculation
run nmos_4_vbs_eff
run nmos_4_Xdep
run nmos_4_VT
run nmos_4_factor_n
run nmos_4_Vgs_eff
run nmos_4_Vgst_eff
run nmos_4_Abulk
run nmos_4_u_eff
run nmos_4_Rds
run nmos_4_Vds_sat
run nmos_4_Vds_eff
run nmos_4_Va_sat
run nmos_4_Va_clm
run nmos_4_Va_clm_dibl
run nmos_4_Va
run nmos_4_Va_scbe
run nmos_4_Ids

%----------------------------------------------------------

%Plot Drain current vs. Drain to Source voltage of nmos transistor

format short eng;
plot(Vds,Ids)
xlabel('Vds')
ylabel('Ids')

%----------------------------------------------------------
Appendix X

VerilogAMS 8-Bit Shift Register

//Verilog-AMS HDL for “teslib”, “shift_regs” “verilogams”

`include “constants.vams”
`include “disciplines.vams”
`timescale 1ns/100ps

Module shift_regs (in7, in6, in5, in4, in3, in2, in1, clk, bit7, bit6, bit5, bit4, bit3, bit2, bit1, bit0);

    output bit7, bit6, bit5, bit4, bit3, bit2, bit1, bit0;
    input clk, in7, in6, in5, in4, in3, in2, in1, in0;
    electrical in7, in6, in5, in4, in3, in2, in1, in0
    logic clk;
    electrical bit7, bit6, bit5, bit4, bit3, bit2, bit1, bit0;

real a[7:0];  //first register
real b[7:0];
real c[7:0];
real d[7:0];
real e[7:0];
real f[7:0];
real g[7:0];
real h[7:0];  //last register

always @(posedge clk) begin

    h[7] = g[7];
    g[7] = f[7];
    f[7] = e[7];
    e[7] = d[7];
    d[7] = c[7];
    c[7] = b[7];
    b[7] = a[7];
    a[7] = V(in7);

    h[6] = g[6];
    g[6] = f[6];
    f[6] = e[6];
    e[6] = d[6];
    d[6] = c[6];
    c[6] = b[6];
b[6] = a[6];
a[6] = V(in6);

h[5] = g[5];
g[5] = f[5];
f[5] = e[5];
e[5] = d[5];
d[5] = c[5];
c[5] = b[5];
b[5] = a[5];
a[5] = V(in5);

h[4] = g[4];
g[4] = f[4];
f[4] = e[4];
e[4] = d[4];
d[4] = c[4];
c[4] = b[4];
b[4] = a[4];
a[4] = V(in4);

h[3] = g[3];
g[3] = f[3];
f[3] = e[3];
e[3] = d[3];
d[3] = c[3];
c[3] = b[3];
b[3] = a[3];
a[3] = V(in3);

h[2] = g[2];
g[2] = f[2];
f[2] = e[2];
e[2] = d[2];
d[2] = c[2];
c[2] = b[2];
b[2] = a[2];
a[2] = V(in2);

h[1] = g[1];
g[1] = f[1];
f[1] = e[1];
e[1] = d[1];
d[1] = c[1];
c[1] = b[1];
b[1] = a[1];
a[1] = V(in1);

h[0] = g[0];
g[0] = f[0];
f[0] = e[0];
e[0] = d[0];
d[0] = c[0];
c[0] = b[0];
b[0] = a[0];
a[0] = V(in0);

end

analog begin

V(bit7) <+ h[7]    //MSB
V(bit6) <+ h[6]
V(bit5) <+ h[5]
V(bit4) <+ h[4]
V(bit3) <+ h[3]
V(bit2) <+ h[2]
V(bit1) <+ h[1]
V(bit0) <+ h[0]    //LSB

end
endmodule
References


