2011-01-01

An Ultra Low Power Subthreshold Delta Sigma Digital To Analog Converter

Ricardo Baca
University of Texas at El Paso, ricardo.baca@gmail.com

AN ULTRA LOW POWER SUBTHRESHOLD DELTA SIGMA DIGITAL TO ANALOG CONVERTER

RICARDO BACA BAYLON

Department of Electrical and Computer Engineering

APPROVED:

Eric MacDonald, Ph.D., Chair

John Moya, Ph.D.

Ryan B. Wicker, Ph.D.

Patricia D. Witherspoon, Ph.D.
Dean of the Graduate School

Dedication

To my family and friends

AN ULTRA LOW POWER SUBTHRESHOLD DELTA SIGMA DIGITAL TO ANALOG CONVERTER

by

RICARDO BACA BAYLON, BSEE

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Department of Electrical and Computer Engineering

THE UNIVERSITY OF TEXAS AT EL PASO

May 2011

Acknowledgements

I would like to thank my advisor Dr. Eric MacDonald for making this thesis possible and for all his support and guidance throughout the entire course of this research I would also like to thank my other committee members – Dr. Moya and Dr. Wicker for being on my defense committee and giving me valuable suggestions and comments. Additionally I like to thank to Dr. Ameet Chavan, Praveen Palakurthi and Matt Markulik for the advices and support during this research.

Abstract

In the last four decades the incorporation of analog, digital and mixed-signal integrated circuits (IC) using Complementary Metal Oxide Semiconductor (CMOS) technology has been dramatically increasing and is improving almost every aspect of modern life. Furthermore, the continuous advances in CMOS Very Large System Integration (VLSI) technology allow higher integration densities and the increasing number of subsystems in a single chip results in higher speed execution and lower power consumption. Therefore, the increasing digitalization in electronic circuits across a wide range of battery-operated applications where reliability and power dissipation are critical requires Digital to Analog Converters (DACs) with higher resolution and lower power consumption.

The present work analyzes the operation and response of ΔΣ DAC at subthreshold voltage levels. At subthreshold regime, the voltage supply is far below traditional voltage levels. Additionally, subthreshold operations have the advantage over other techniques to achieve the optimum operating point per operation and thereby reduce the overall power dissipation. The ΔΣ DAC was designed in TSMC 0.25 μm CMOS technology. The design entry was done with Virtuoso Schematic Editor and the simulations at block-level and chip-level were performed with Virtuoso Spectre Simulator, which is able to interpret and simulate SPICE netlist. Furthermore, the physical design layout was made at the symbolic level with Virtuoso Layout Suit; post-layout simulations were performed utilizing the extracted netlist of the design.

The DAC was simulated with a changing digital input generated from eight ideal pulse sources, which gives the maximum power consumption case. The input frequencies and supply voltage were changing in a range from 3 KHz to 293MHz and 150mv to 2.5v respectively. The main results of the DAC are the chip area 180μm x 450μm and power consumption is 4.49 μW at 350mv. The design will be fabricated in May 2011 and experimental validation of the design will be performed by the end of the year.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 MOTIVATION

In the last four decades the incorporation of analog, digital and mixed-signal integrated circuits (IC) using Complementary Metal Oxide Semiconductor (CMOS) technology has been growing and is applied to almost any application in modern life. Research advances in CMOS technology are focused on how to create faster and smaller devices, in order to satisfy the primary concerns of high performance (usually operates in the range of hundreds of megahertz up to several gigahertz) and low power (i.e. battery operated devices, energy harvesting, etc.). Modern advances in CMOS technology allow higher integration densities and the increasing number of subsystems in a single chip produces higher speed execution and lower power consumption. However, these achievements are not for free and present several limitations.

Significant research has concentrated on those limitations concerned with lowering power consumption. Different techniques have been applied to achieve ultra-low power devices involving low clock frequencies (in the range of Kilohertz), producing a tradeoff between performance and power dissipation. Furthermore, is not surprising that the parameter mA/MHz has become highly important for battery operated devices like mobile electronics and implanted bio-medical devices in which reliability and power dissipation are critical.

Signal processing is the basis for interaction with the digital domain and natural signals of the analog world. In order to perform the Digital Signal Processing (DSP) is necessary to have data conversion, which is achieved by the implementation of sampling circuits: Analog to Digital (A/D) convertors and Digital to Analog (D/A) convertors. Previous research focuses on the analog part of the design, which is where most of the power is consumed, but the digital realm of data conversion needs further attention in order to maximize the low power design of a DAC, even if is only a small percent of the total power.

The increasing numbers of digital applications are demanding an ultra-low power capability, which requires special attention, new design methods and techniques. An alternative to decrease power consumption is to develop CMOS devices that operate in a low subthreshold voltage level. The subthreshold operations have the advantage over other techniques to achieve the optimum operating point per operation and thereby reduce the overall power dissipation.

1.2 ORGANIZATION OF THE THESIS

Research concentrates on the analysis and performance of 8 bit Sigma-Delta DAC, which operates exclusively in the regime of subthreshold voltage levels while providing superthreshold analog voltage levels through an optimized level shifter and a RC low pass filter. The organization of the thesis is as follows.

Chapter 2 gives exposure of the subthreshold operations levels continued by the performance metrics and a basic overview of the different architectures of data converters.

Chapter 3 explains the methodology, circuit design, and technology considerations of the Delta-sigma DAC.

Chapter 4 discusses the results obtained after the analysis.

Chapter 5 provides the conclusions and future work.

Appendix shows the Verilog Code, Hspice netlist and ocean scripts utilized in the design analysis.

Chapter 2: Background

Subthreshold is currently the focus of substantial research in which low energy per operation is the driving advantage. Simultaneously, CMOS technology continues to scale down in both dimensions as well as voltage levels. The scaling naturally leads further advantage to subthreshold operation and also requires significant effort in understanding the implications to design of both analog and digital circuits.

2.1 SUBTHRESHOLD LEVELS

In digital CMOS technology transistors usually operate either in the ON state (saturation, active mode, strong inversion) or in the OFF state (subthreshold, weak inversion). However, subthreshold means either OFF state or almost-ON state (still in subthreshold regime but with weak inversion) [1]. Operation curves at subthreshold and normal voltage levels for both NMOS, and PMOS are illustrated in the Figure 1.1[1]. As the transistor moves away from the traditional ON state and approaches to subthreshold levels, the voltage gate-source (V_{gs}) goes below the Voltage of threshold (V_{th}) and the dominant mechanism of current becomes limited by diffusion current instead of drifting current, at subthreshold regime, the MOSFET behavior can be compared with a bipolar transistor, were the current is exponentially dependent on the V_{gs}, and approximately independent of the Voltage drain-source (V_{ds}).

2.1.2 Power consumption and voltage scaling

In CMOS technology two main power dissipation sources are present: static power, which results in power dissipation through resistive paths from the power source to the ground connection (primarily through leakages), and dynamic power, which results from switching capacitive loads between different voltage levels. The dynamic power dissipation in a CMOS circuit [2] is expressed as:

$$P_{dynamic} = \alpha C {V_{dd}}^2 f \qquad (2.1).$$

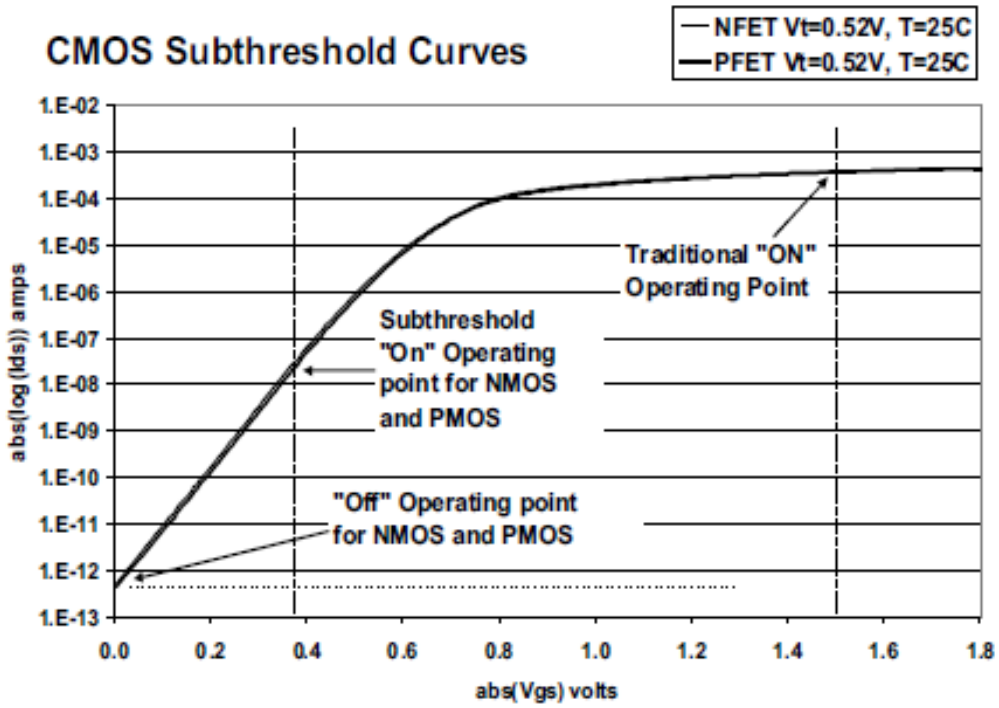

Figure 1.1: CMOS Subthreshold Curves.

Where V_{dd} is the power supply, C is the total capacitance of the circuit, and α is the activity ratio, which means the number of times the node switches within each clock cycle. Equation 2.1 shows that, if V_{dd} is scaled down at subthreshold levels the dynamic power will be affected with a largest impact since the V_{dd} has a quadratic correlation with the power.

Voltage scaling, which affects the operation of the circuit, incrementing the delays as the power supply approaches to V_{th}, which is due to the threshold voltage and the capacitance are constant as shown in [2][3]. Analyzing a quadratic model of the transistors leads to:

$$t_d = K \frac{C V_{dd}}{(V_{dd} - V_{th})^2} \qquad (2.2).$$

Where K is the technology related factor, C is the capacitance, and t_d is the time delay. To achieve the performance requirements, shall be necessary to reduce the value of the threshold voltage, in order to establish the optimal operation point by providing sufficient drive current. During the design process, the leakage currents and the logical delays are two of the top challenges to overcome.

However, lowering threshold voltage has a negative impact on static or idle mode power, as given by the following Equation [4]:

$$I_{ds} = \frac{W}{L} I_s \times e^{\left(\frac{V_{gs}-V_{th}}{aV_T}\right)} \quad (2.3).$$

Where, V_{tn} is the voltage of threshold and V_T is the thermal voltage, which is temperature dependent. Equation 2.3 shows that leakage current has an exponential relationship to V_{th} and the relationship to temperature (T) [5]. Subthreshold circuits operate with a supply voltage that is less than the threshold voltage of the transistor – far below traditional levels and consequently the transistor operates essentially on leakage, hence, leakage current is used as the operational current, which results in a significant limitation on the maximum performance of the subthreshold circuits.

2.1.3 Optimal operation points for energy per operations

In order to reduce the power dissipation in the CMOS circuits design, the source voltage and the threshold voltage have been scaled down at far below traditional levels, resulting in a tradeoff between performance and power consumption; in order to measure the efficiency of circuits operating in subthreshold levels, the usage of design metrics are necessary and provide a measure of goodness when comparing different designs that carry out the same logical function and can also be used to optimize a design to achieve a minimum value for a given design metric or a weighted set of metrics [6].

The Energy Delay Product (EDP), which is essentially the energy per operation, is given by [6]:

$$Energy \times Delay = (Energy_{dynamics} + Energy_{leak}) \times Delay \quad (2.4)$$

The EDP needs to be plotted with different set of values for V_{dd} and V_{th}, in order to give the optimal operation point for a minimum EDP [5].

2.2 DATA CONVERTERS

Data converters are used in electronic circuits to interface between the analog and the digital domain. Conceptually a data converter performs a transformation of signals: from discrete time and quantized–amplitude to continuous-time and continuous-amplitude and vice-versa [7]. The research presented in this thesis is focused on digital to analog conversions; the Figure 2.1 illustrates a block diagram of an ideal DAC.

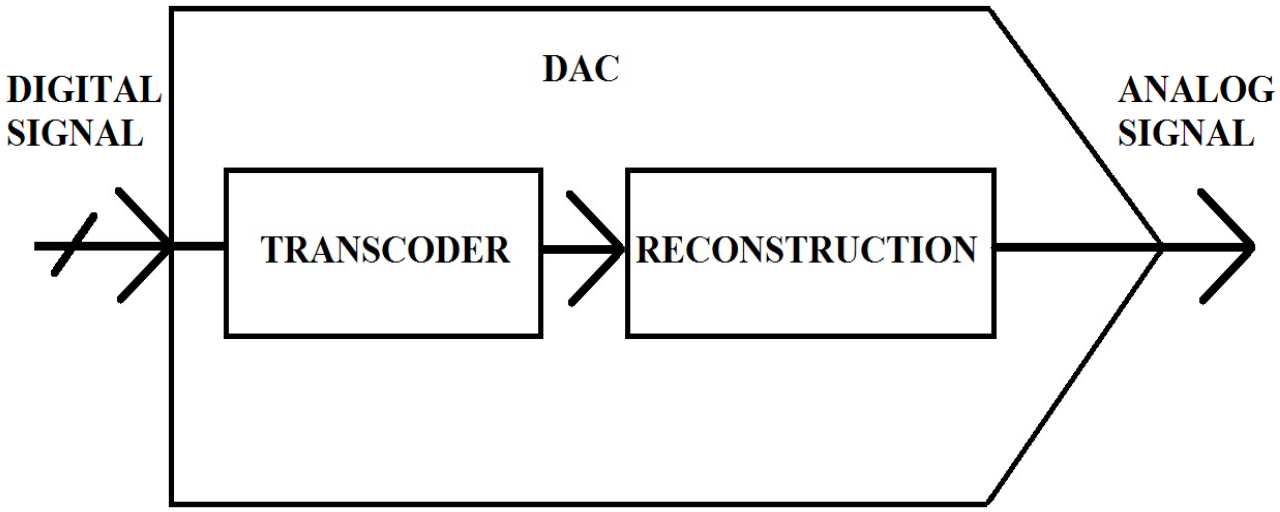

Figure 2.1: Ideal DAC Block Diagram.

The ideal DAC is divided in two operation blocks, the transcode block and reconstruction block. The purpose of the transcoder block is to take a digital signal of n bits and transform them into an equivalent analog signal. The reconstruction block removes the high frequency domain components of the sampled data, given a clear analog output. The output of a DAC can be a voltage or a current; for the present research a voltage output is preferred to represent a portion of a reference signal or digital input. In the ideal voltage DACs the output is given by [8]:

$$v_{out} = \alpha * V_{ref} \tag{2.5}$$

Where α is a proportionality factor of V_{ref}, which can be obtain by [9]:

$$\alpha = [b_0 2^{-1} + b_1 2^{-2} + \cdots + b_{n-2} 2^{-n-1} + b_{n-1} 2^{-n}] \quad (2.6)$$

Where n is the number of bits of the DAC, b_0 is the Most Significant Bit (MSB) and b_{n-1} is the Less Significant Bit (LSB).

2.3 PERFORMANCE METRICS OF DATA CONVERTERS

The performance metrics of data converters describes the deviation of the actual function from the ideal function. Additionally are classified in three broad categories: General parameters, the static response and the dynamic response.

2.3.1 General Parameters

The first parameter is the resolution, which is specified by the number of bits (n) of the digital input. In a DAC, resolution describes the smallest standard incremental change in the output voltage [10]. The value of a single LSB changes can be defined as

$$LSB = \frac{V_{ref}}{2^n} \quad (2.7)$$

According to Equation (2.6) an infinite number of bits would be necessary to have an α equal to one. Hence, difference between the analog out and V_{ref} needs to be measured. This relationship is called the Full Scale Range (FSR) and is given by

$$FSR = V_{ref} - LSB \quad (2.8)$$

The previous concept leads to quantizing error or quantization noise, which is the maximum deviation from a straight-line transfer function. However, separation is never greater than 1LSB and can be designed to be ± ½ LSB. The quantization noise can be expressed as

$$Q_{noise} = \frac{LSB}{\sqrt{12}} rms \tag{2.9}$$

2.3.2 Static Parameters

The static response describes the deviation of the actual from the ideal response when the input is held at a fixed value. The mismatch between equal components, which differ from the nominal value, will be the most common source of statics errors. Static parameters are shown in Figure 2.2[8].

2.3.2.1 Offset Error

The offset error, also known as zero error, is the difference between the ideal analog output and the actual analog output of a DAC, with digital input value of zero or

$$E_{offset} = \frac{\tilde{v}_{out}}{LSB}\Big|_{b_{n-1}} \tag{2.10}$$

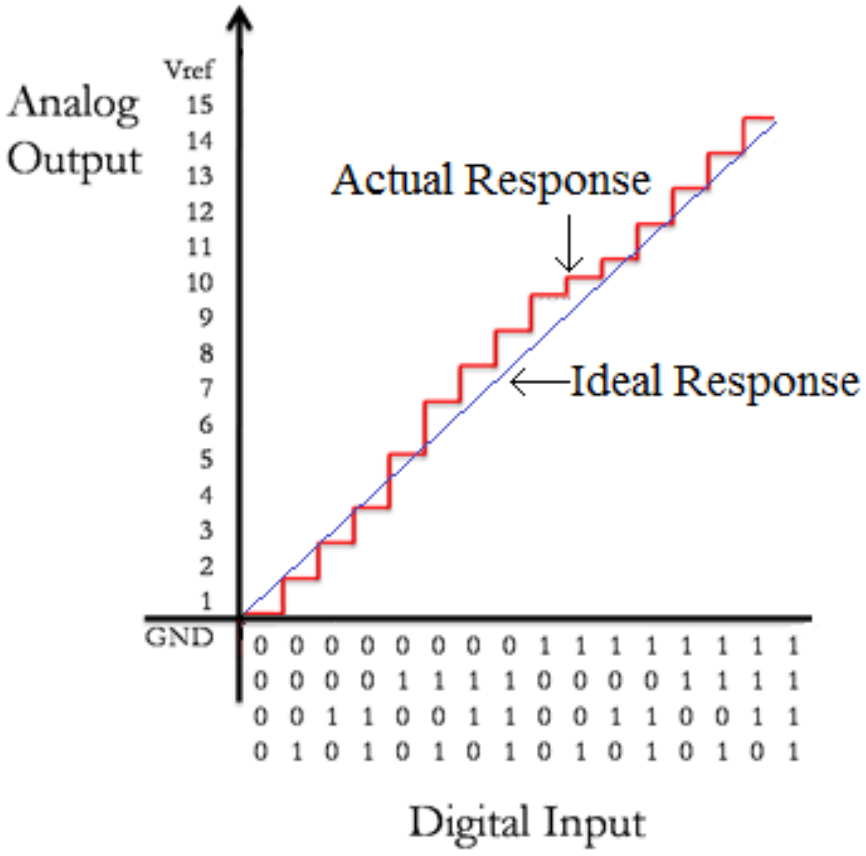

Figure 2.2: D/A Static Parameters.

2.3.2.2 Full Scale Error

The full scale error is similar to offset error, however is evaluated with digital input value of MSB such as

$$E_{full-scale} = \frac{\tilde{v}_{out}}{LSB}\Big|_{b_0} - 2^n - 1 \qquad (2.11)$$

2.3.2.3 Gain Error

Gain error measures the error on the slope of a straight line interpolating the input-output characteristic of the converter. In order to measure gain error is necessary the combination of Equation (2.10) and (2.11), which leads to

$$E_{gain} = \frac{\tilde{v}_{out}}{LSB}\Big|_{b_0} - \frac{\tilde{v}_{out}}{LSB}\Big|_{b_{n-1}} - 2^n - 1 \quad (2.12)$$

Or

$$E_{gain} = E_{full-scale} - E_{offet} \quad (2.13)$$

2.3.2.4 Differential nonlinearity error

Differential nonlinearity indicates the maximum deviation between the ideal and the measured output responses for successive DAC codes. Therefore, ideal DAC response would have an analog output values exactly the step size of one LSB, and can be defined as

$$DNL_k = \tilde{v}_{out}|_{k+1} - \tilde{v}_{out}|_k - LSB \quad (2.14)$$

Where *k* is the index of the digital input applied.

2.3.2.5 Integral nonlinearity error

Integral nonlinearity error (INL) is the maximum deviation of the input/output characteristic from straight line passed through its end points and can be calculated as

$$INL_k = \tilde{v}_{out}|_k - v_{out}|_k \quad (2.15)$$

2.3.2.6 Monoticity

Monoticity can be defined as: every change in the digital input ought to result in a change in the analog output in the same direction and is given by

$$|INL_k| \leq \frac{1}{2} LSB \text{ for all } k \tag{2.16}$$

2.3.3 Dynamic Parameters

The dynamic response describes the deviation of the actual analog output from the ideal response with time varying inputs. Dynamic parameters are illustrate in Figure 2.3[8]

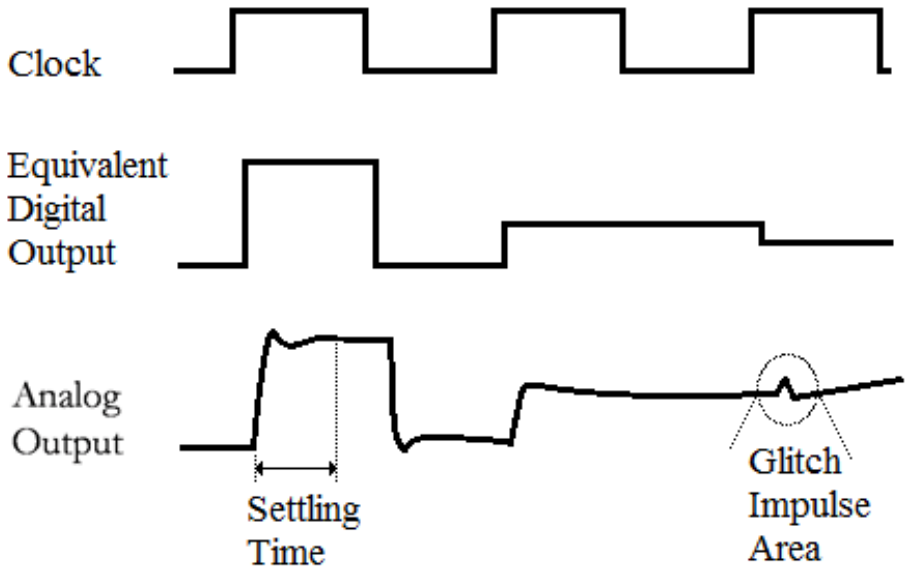

Figure 2.3: D/A Dynamic Parameters.

2.3.3.1 Settling Time

Settling time is the amount of time required for the output to experience full-scale transitions and settle within a $\frac{1}{2}$ LSB of the voltage of reference. The maximum settling time occurs at half scale when the MSB changes. In order to calculate the maximum sample rate of a D/A converter from the settling time will be required to use the following Equation

$$sampling\ rate\ (max) = \frac{1}{settling\ time} \tag{2.17}$$

2.3.3.2 Glitch Energy

The glitch energy also refers to glitch energy area, which is the maximum area under any extraneous glitch that appears at the output after the input code changes and is measured at half scale when DAC input switches around the MSB and may switches change state (i.e., from 01…11 to 10…00) [7] in units of volt-second.

2.3.3.3 Latency

Latency is the delay from the time the digital input changes plus the settling time of analog output.

2.3.3.4 Signal-to- Noise Ratio

The Signal-to-noise ratio (SNR) is the ratio of the root-mean-square (rms) value of the output signal to the rms values of the quantization noise. The SNR is expressed as

$$SNR = \frac{v_{out}rms}{Q_{noise}rms} \tag{2.18}$$

For the reconstruction of sine waves the maximum SNR is defined in decibels (dB) as

$$SNR_{max}dB = 6.02n + 1.76 \tag{2.19}$$

The inverse relationship is applied to oversampling converters to convert an SNR into an effective number of bits.

But if the reconstructed signal is a triangle wave then

$$SNR_{max}dB = 6.02n \tag{2.20}$$

2.3.3.5 Signal-to- Noise-and-Distortion Ratio

The Signal-to-Noise-and-Distortion Ratio (SNDR) is the ratio of the value of the output signal to the power of the quantization noise including AC harmonics. Since static and harmonic limitations cause a non-linear response, the SNDR depend on the frequency of the input signal and decrease proportionally with the input amplitude.

2.3.3.6 Total Harmonic Distortion

The Total Harmonic Distortion (THD) is calculated by taking the ratio between the root mean square of the signal and the root mean square of harmonics. Additionally, unless otherwise specified the harmonic distortion accounts from the second through tenth harmonics [7].

2.3.3.7 Dynamic Range

The Dynamic Range (DR) is the value of the input signal at which the SNR or the SNDR are equal to 0 dB. The DR can be calculated by

$$DR\ dB = 6.02n \tag{2.21}$$

2.3.3.8 Effective Number of Bits

The effective number of bits (ENOB) measures the signal-to-noise and distortion ratio using bits. A full scale sinusoidal reconstruction ENOB is determined by

$$ENOB = \frac{SNDR\ dB - 1.76}{6.02} \tag{2.22}$$

2.3.3.9 Spurious Free Dynamic Range

The Spurious Free Dynamic Range (SFDR) is a measure in dB of the difference in power between the output signal and the largest spur present in the frequency spectrum.

2.4 DATA CONVERTERS ARCHITECTURES

Commonly used DAC architectures are reviewed, in order to identify the most suitable DAC for a specific application. The current research mainly concentrates on exploiting subthreshold operation of digital circuits to low power in DAC; as mentioned in the beginning of this chapter, the circuits that operate at subthreshold levels consume less power. Hence, most of the circuitry involved in the converter architecture should be in the digital domain.

DACs are classified in two main categories based on sample frequency (f_s): *Nyquist-rate* and *oversampling* converters. In the *nyquist-rate* the sampling frequency is slightly higher than twice the analog signal bandwidth to allow accurate reproduction of the original data. On the other hand, in the *oversampling* converter, the signal is sampled at many times the Nyquist-rate and subsequent digital filtering is utilized to remove the noise outside the desired signal bandwidth [8].

In Nyquist-rate converters, the linearity and accuracy is determined by matching accuracy of the analog components (resistors, capacitor, and current sources). Practical conditions restrict the matching accuracy to about two percent; hence, the ENOB is equal to about 12 bits for these converters [11]. However, Oversampling data converters makes extensive use of digital processing, based on the fact as the technology scale down, VLSI is better suited for providing fast digital circuits than for providing precise analog circuits [12]. Since the sampling rates are much higher than the nyquist-rate and generate each output utilizing preceding inputs, the oversampling converters are able to achieve over 20 ENOB of resolution at high conversion speed rate.

Furthermore, DACs are categorized as either being serial or parallel. Serial DACs convert the analog output one bit at a time and therefore require a conversion time for each bit (nT), where n is the number of bits and T is the time of conversion of 1-bit whereas Parallel DACs convert all bits at the same time, hence the conversion time is T. Additionally, DACs are further classified by the electrical components and how are connected to the scaling network in order to generate v_{out}: voltage scaling, current steering, charge distribution or delta-sigma in the case of oversampling converters.

2.4.1 Voltage Scaling

One of the simplest voltage DACs is the Kelvin divider. The Kelvin divider is composed of an equal value resistive string connected across the positive and negative references. The output voltage is generated by switches, which are controlled by digital inputs. Figure 2.4[7] illustrates a Kelvin divider DAC of 2 bits.

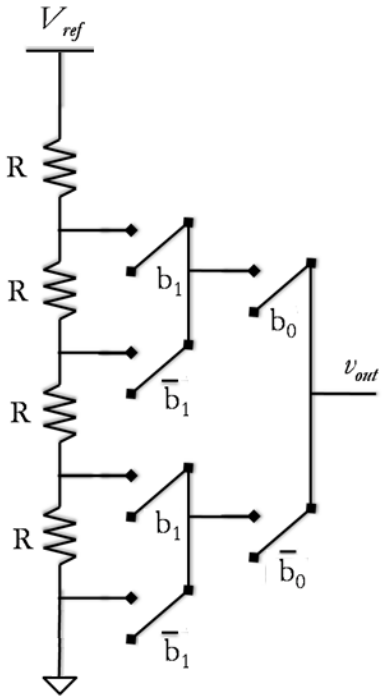

Figure 2.4: Kelvin Divider, Scaling Voltage DAC.

However, limitations of the resistive divider are high power consumption due to the resistive string, besides the number of resistors increases exponentially with the number of bits (n) of the DAC. Therefore, implementing the *R-2R* configuration, shown in Figure 2.5[7] constrain the number of resistors. The output voltage of an *R-2R* is the successive division of V_{ref} by two, and is given by

$$\boldsymbol{V_{Out} = \frac{V_{ref}}{2} b_{n-1} + \frac{V_{ref}}{4} b_{n-2} + \cdots + \frac{V_{ref}}{2^{n-1}} b_1 + \frac{V_{ref}}{2^n} b_0} \qquad (2.23)$$

2.4.2 Current Steering

The *R-2R* configuration can also generate a current as an output. In order to set the *R-2R* configuration in current mode, the switches either direct the current to the output or discharge them to ground, as shown in Figure 2.6[7]. Furthermore, if the current output is connected to an op-amp on the inverting configuration an analog voltage is obtained.

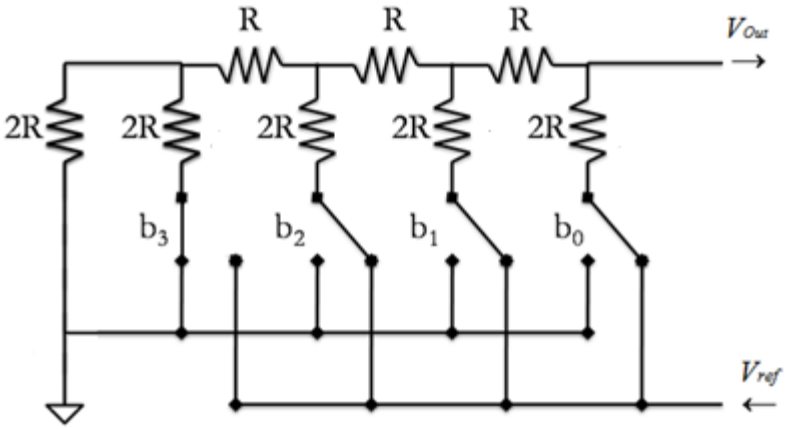

Figure 2.5: Voltage R-2R 4-bit DAC.

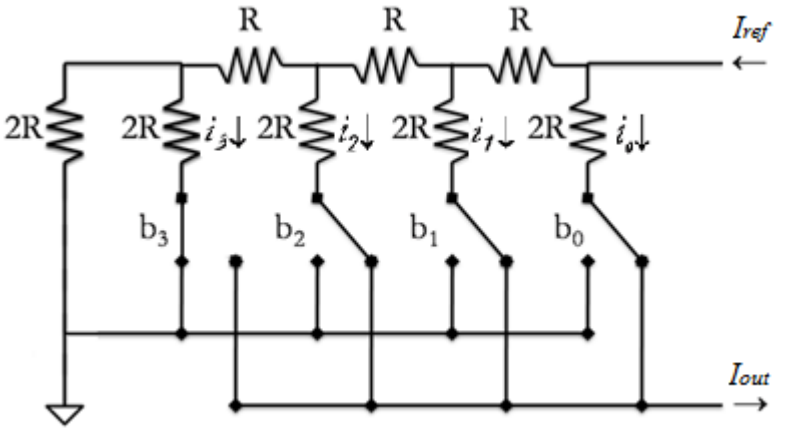

Figure 2.6: Current R-2R 4-bit DAC.

However, the layout of the resistors consumes relatively more silicon area if the value of unity resistance is large and the used layer has a low specific resistance [7]. If the silicon area is a constraint,

the *R-2R* architecture can be implemented by replacing the resistors with *MOS* transistors; the single resistance (*R*) is made by one *MOS* transistor and the double resistance (*2R*) is made by two *MOS* transistors connected in series. However, the usage of MOS transistors decreases the accuracy due the body effect.

Another implementation of a current steering converter is the binary weighted currents shown in Figure 2.7[7]. In binary weighted architecture the I_{ref} is divided in a set of 2^n uniform segment currents. In order to achieve the uniform segment currents, each transistor is doubled compared to the neighbor element. The switching function is done with a differential pair called current steering cell. Depending on the resolution and application of the converter, the switching cell may be sized uniformly of binary weighted.

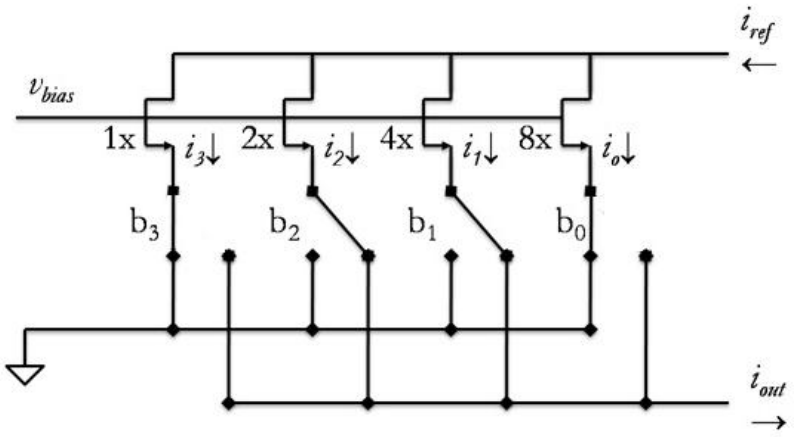

Figure 2.7: Current Steering Binary Weighted DAC.

2.4.3 Charge Redistribution

The Charge DAC is designed with binary weighted capacitors. The charge of the DAC is stored on a binary weighted array of capacitors; hence the output voltage is generated depending on the charge distributed at each capacitor. The charge distribution is controlled by a switch; the controlling switches are typically designed with NMOS pass transistors with complementary input signal. Additionally,

charge DACs are a popular architecture for low power design because the static consumption is zero. Figure 2.8[7] illustrate a four bit implementation.

In charge redistribution DAC the non-linearity is given by a number of factors: capacitor mismatch, capacitor non-linearity and parasitic junction capacitance for any MOS switch connected at the output node.

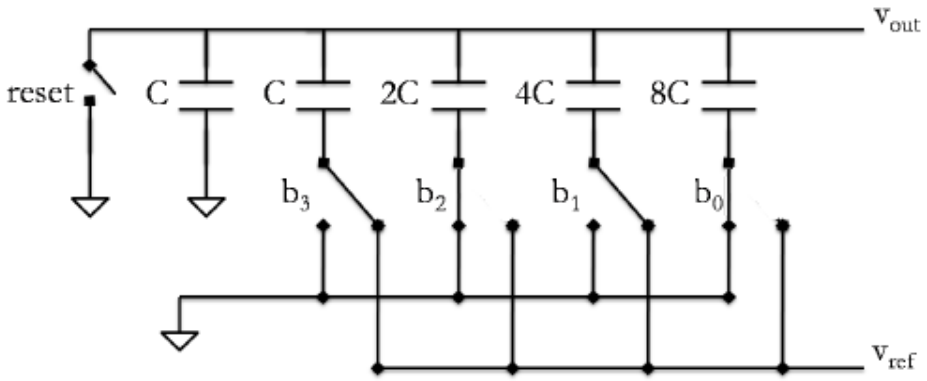

Figure 2.8: 4-Bits Charge Redistribution DAC.

2.4.4 Delta-Sigma DAC

The Delta-Sigma DAC block diagram is shown in Figure 2.9. The DAC consists of two main blocks: the Delta-Sigma (ΔΣ) Modulator and the Low Pass Filter (LPF). The ΔΣ Modulator is the core of the converter; the output of the ΔΣ Modulator is a stream of pulses of equal width, where the average density of the stream pulses corresponds to the digital input value. In order to obtain a precise analog output the bitstream is passed through the LPF. Hence, the complexity and size of the LPF depends on the frequency of the bitstream [13].

The operation of the digital ΔΣ modulator can be described as follows. The *Σ adder* functions as an integrator, which accumulates the input at a rate proportional to the magnitude of the input. When Σ becomes a negative number (i.e. when the MSB of the Σ register equals one) the Δ error signal is

subtracted from Σ such that the accumulated value is reduced to a smaller positive value and the integration is continued until the overflow takes place again. Therefore, the MSB of Σ register is the bitstream and the rate at which the MSB becomes one is directly proportional to the DAC input [13]. A Digital ΔΣ modulator is illustrated in Figure 2.10.

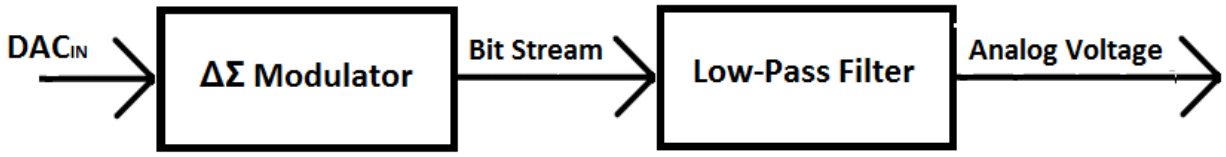

Figure 2.9: Block Diagram ΔΣ DAC.

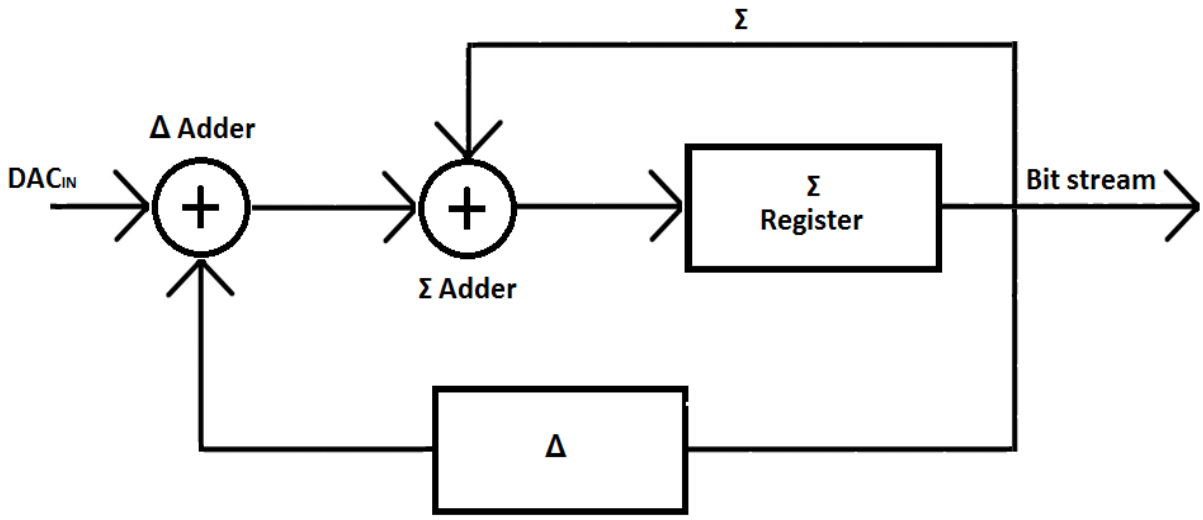

Figure 2.10: Block Diagram Digital Modulator.

2.5 DAC PERFORMANCE

Selecting a DAC for a specific application depends on the requirements of the application. For a Subthreshold operating DAC, most of the circuitry that integrates the converter needs to be in the digital domain. Therefore, due the high density of digital elements, the most suitable DAC for operations at subthreshold levels is the Digital ΔΣ DAC, as shown in Table 2.1.[14][15]

Table 2.1: DACs Parameters.

Parameters	ΔΣ DAC Digital	Voltage Scaling	Charge Redistribution	Current Steering
Linearity	High	High	High	Low
Speed	low	Medium	High	High
Power	Low	Low	Low	High
Noise	Very Low	Low	High	Medium
Digital Elements Density	High	Low	Low	Low

2.6 PREVIOUS WORK

Present-day, the interaction between the digital domain and analog world has been increasing constantly. Furthermore, the high demand of battery operated or energy constrained devices in mobile applications, bio-medical applications and energy harvesting devices have escalated the need for low power DACs. Improvements needed on DACs design are low power dissipation, while simultaneously maintaining reliable performance of the device.

Many papers have been published in the design of ΔΣ ADCs and ΔΣ DACs, but the papers published on ΔΣ ADCs outnumber those on DAC's by a factor of 10 or more. The reason lies in the historical circumstances of development [16]. Research and development has been focused on ADCs, thus there are still plenty of improvements opportunities on DAC design as well are equally important

and therefore have awakened an interest in ASIC group. The following is brief review of different ΔΣ DACs found in the literature.

A low power 98-dB ΔΣ Mutibit DAC architecture is presented in [17]. The relative high power consumption of (28mW) is due to the use of standard 3.3 voltage supply 0.35µm CMOS technology. However, the performance of ASIC (Application Specific Integrated Circuit) was achieved with a tradeoff of power consumption and incrementing the order of the reconstruction filter. A novel low power ΔΣ DAC architecture is proposed in [18]. The DAC is implemented in a 0.35µm CMOS technology, consuming 2.6mW from a 0.8 Voltage supply and achieving an 88 dB dynamic range; the design consist in two main stages: the digital stage and analog part. However, most of the power is consumed by the analog stage, as shown in Table 2.2 [18]. Other research [19], proposed a 0.5 voltage supply ΔΣ one-bit DAC in a 0.18µm CMOS technology without implementing any internal voltage boosting or low-threshold devices. In order to achieve the 300µW of power consumption and SNDR of 74dB, the architecture implemented the use of only low power design techniques.

Table 2.2: Power Consumption.

Supply	Power Consumption ΔΣ Modulator	Total Power
0.7 v	0.098 mW	2.6 mW
0.8 v	0.128 mW	2.6 mW
0.9 v	0.162 mW	2.6 mW
1.0 v	0.200 mW	2.6 mW
1.1 v	0.242 mW	2.6 mW
1.2 v	0.288 mW	2.6 mW
1.5 v	0.495 mW	2.6 mW
1.8 v	0.954 mW	2.6 mW

Chapter 3: Proposed 8-bit Subthreshold Delta Sigma DAC

The proposed converter consists of 3 main blocks: ΔΣ digital modulator, level shifter and reconstruction low pass filter. The DAC takes an 8-bits input in order to obtain a corresponding analog output; however, the voltage of the inputs as well as voltage supply for the ΔΣ digital modulator operates at subthreshold levels in order to conserve energy. The level shifter is used to amplify the bitstream voltage of the ΔΣ Modulator to superthreshold levels and a subsequent low pass filter is used to acquire analog output by averaging the digital output signal. The proposed block diagram is shown in Figure 3.1.

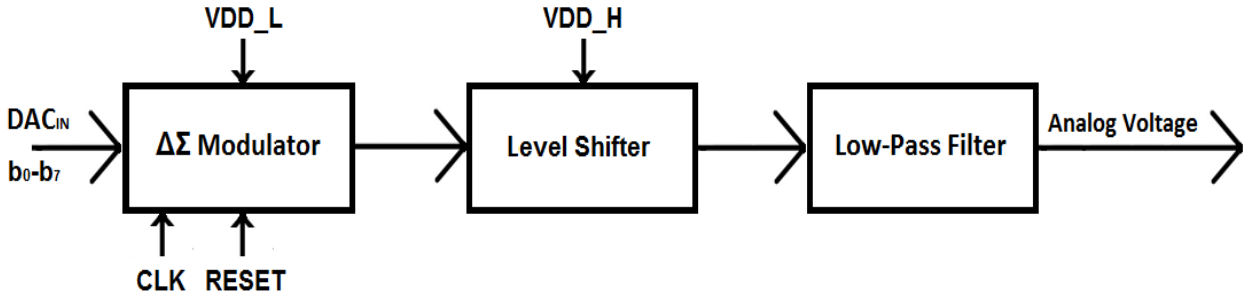

Figure 3.1: Block Diagram of Proposed ΔΣ DAC.

3.1 DIGITAL DELTA SIGMA MODULATOR

The digital ΔΣ modulator is the core of the DAC. There exists various techniques in order to implement a digital ΔΣ modulator (i.e. software only, hardware only or a combination of both); the software-only implementation presents an inexpensive digital ΔΣ Modulator, but includes some penalties such as low speed and loss of accuracy. However, a hardware-only implementation is the best possible method in terms of speed, performance and accuracy [13] but at the expense of requiring a circuit to fabricate. The Digital ΔΣ Modulator is designed in Verilog, which contains an algorithm that executes the fundamental operations of the modulator. Such operations are the delta adder (Δ) and the sigma adder or sigma integrator (Σ), all of which is required in order to obtain the output bitstream the average of which (DC value) provides the analog output. The ΔΣ Modulator is illustrated in Figure 3.2 [13].

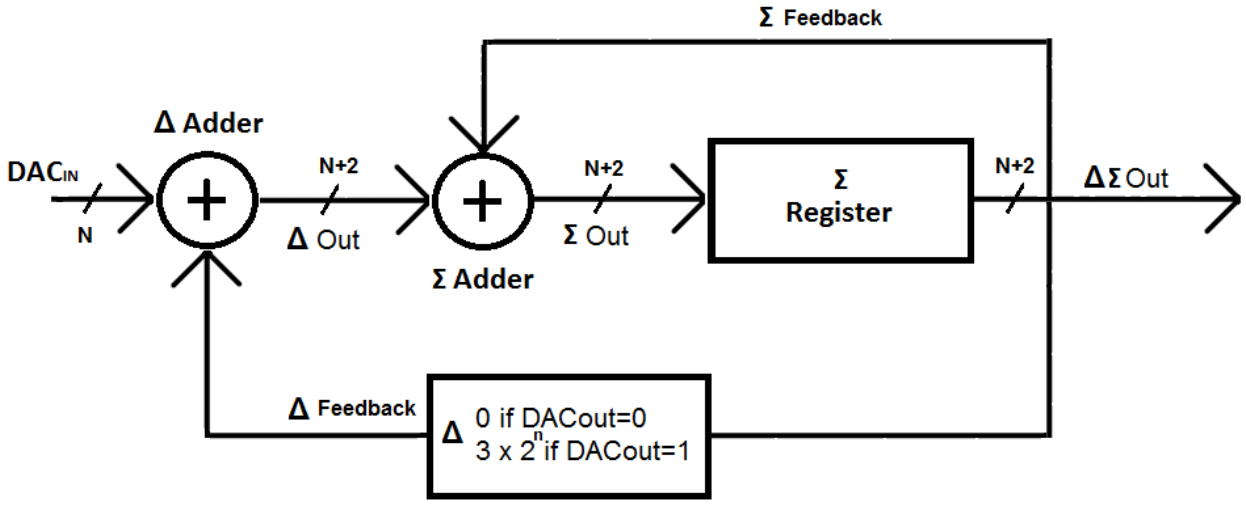

Figure 3.2: Block Diagram of Digital The ΔΣ Modulator.

According to the ΔΣ Modulator algorithm for an N bit input, the input should be signed. However, the input bit number is unsigned. Therefore, an extra two bits are added to obtain the input bit number as a signed number. As shown in Figure 3.2, the function of the Δ adder is to obtain the difference between the DAC input and ΔΣ output; therefore, the Δ feedback to Δ adder is related to the ΔΣ output, which is either 1 or 0. The Δ feedback is an N + 2 bit number with all in 0 if the ΔΣ output is 0; the Δ feedback is the 1's complement of the highest N bit number, sign extended to N + 2 bit if the ΔΣ output is 1[13]. Simultaneously the Σ adder operates as an integrator, which accumulates the input at a rate proportional to the magnitude of the input. The Σ adder sums the output of the Δ adder (Δout) and the current N + 2 bit number content in the Σ register, then the output of the Σ adder (Σout) is stored in Σ register; the ΔΣ output is the MSB of the N + 2 bit number stored in the Σ register.

The operation of an 8-bit ΔΣ Modulator can be explained with a 3-bit example. Table 3.1[13] illustrates the bitstream output steps of a 3-bit ΔΣ Modulator with 011 value at the input. The process is explained as follows: Initializing the Σ register with 10000 and as the MSB of the Σ register is the ΔΣ Out, hence ΔΣ Out is 1. Therefore, Δ feedback results in value of 11000, otherwise results in all 0's; in order to obtain the Δout 11011, the Δ adder computes the addition between the DAC input 00011 and the Δ feedback 11000. In the Σ adder the value of Σout 01011 is computed by the sum of the Σ register

10000 and the value of Δout 11011. The output of the Σ adder, which is Σout 01011, will be stored in the Σ register, the MSB will give the next value for the ΔΣ Out, which is 0. Since the output of the ΔΣ Out is 0, Δ feedback results in value of 00000. Therefore, the ΔOut 00011 of the Δ adder is obtained by the sum of DAC input 00011 and Δ feedback 00000. Σ adder sums the values between ΔOut 00011 and Σ register 01011 in order to obtain the value of Σout 01110, which will be stored in the Σ register providing the next value of the ΔΣ Out, which is 0. Furthermore, the ΔΣ Modulator operations are continued in time until the density of 1's is directly proportional to the input digital value, then the operations start over. Hence, for a 3-bit input DAC with an input of 011, the density of 1's to frame time is 3/8, as shown in Table 3.1.

Table 3.1: 3bit ΔΣ Modulator Bitstream Output Steps.

	TIME							
	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7
Δ	11000	00000	00000	11000	00000	00000	11000	00000
Δ_{Out}	11011	00011	00011	11011	00011	00011	11011	00011
Σ	10000	01011	01110	10001	01100	01111	10010	01101
Σ_{Out}	01011	01110	10001	01100	01111	10010	01101	10000
$\Delta\Sigma_{Out}$	1	0	0	1	0	0	1	0

The Relationship between of an 8-bit DAC input values, and the density of pulses of the bitstream is explained as follows: As the DAC input increases; the duty cycle and the frequency of the output bitstream also increase. However, the output bitstream reaches the maximum frequency when the DAC input is at half scale or 1000 0000 for the presented 8 bit ΔΣ Modulator. Beyond the half scale DAC input value; the output frequency begins to decrease as the duty cycle of the output bitstream continues to increase. Table 3.2 [20] illustrates the corresponding duty cycle and frequency for different input values for an 8-bit DAC.

Table 3.2: Bitstream Frequency and Duty Cycle for an 8-bit ΔΣ Modulator.

DAC Digital Input	Modulator Output Frequency	DAC Output Duty Cycle
1	Clk / 256	1/256
2	Clk / 128	1/128
4	Clk / 64	1/64
8	Clk / 32	1/32
16	Clk / 16	1/16
32	Clk / 8	1/8
64	Clk / 4	1/4
128	Clk / 2	1/2
192	Clk / 4	3/4
224	Clk / 8	7/8
240	Clk / 16	15/16
248	Clk / 32	31/32
252	Clk / 64	63/64
254	Clk / 128	127/128
255	Clk / 256	255/256

A behavioral analysis of the ΔΣ Modulator Verilog code is obtained by NC Verilog simulator, which stimulates the Verilog code with the test bench in order to obtain the dump file. Additionally, the test bench code contains the setup values, initial conditions, clock frequency, input frequency and the precise time which are used to stimulate the ΔΣ digital modulator Verilog code. The dump obtained from the NC Verilog simulator, is used to plot the results in Simvision. Figure 3.3 illustrate the changing inputs (blue colored waveforms) and data output (red colored waveform), which gives the pulse density bitstream. However, the proposed ΔΣ Modulator operates at variable voltage levels from subthreshold to superthreshold levels and performance needs to be analyzed at a CMOS gate-level. The Cadence Register Transfer Level (RTL) compiler (RC) provides an automated synthesis tool. The RC tool compiles the Verilog code and the library components (combinational and sequential circuit cells) available in the technology library in order to obtain a gate level netlist, which is generated according to the constraints defined by the design. Virginia Tech University's TSMC 0.25 μm Standard cells are used in designing the proposed ΔΣ Modulator. Using the Graphical User Interface (GUI) from RC compiler a gate level schematic of the ΔΣ Modulator is presented, as shown in Figure 3.4.

In addition to the gate-level net list report, the RC compiler also generates text reports (i.e. area and timing): an area report lists the related area of each logic module in the design; timing report describes the critical path of the design. The critical path is defined as the slowest logical path between any two registers in the design and thereby limiting the operation frequency of the design. The critical path for the ΔΣ Modulator, obtained from the RC synthesis tool, is from register zero (REG0) to register eight (REG8) passing through ten logical gates, which are one AND logical gate then passing through eight Full Adders and one OR logical gate. Figure 3.4 illustrates the critical path (red colored path) of ΔΣ Modulator. Additionally, the synthesized netlist of the ΔΣ Modulator needs to be implemented in to layout. Cadence Encounter provides a place-and-route tool, which utilizes the synthesized netlist file of the ΔΣ Modulator, the technology file for vtvt tsmc 0.25μm, and the Library Exchange Format (LEF) file, which contains information for the layout such as layers, vias, placements types and macro cells definitions, to generate an equivalent layout view of the ΔΣ Modulator. Furthermore, the synthesized

netlist and equivalent layout view of the ΔΣ Modulator were imported to Cadence Schematic viewer and Cadence Layout viewer in order to verify if the Schematic and the layout have properly generated through Design Rule Check (DRC) and Layout Vs Schematic check (LVS). Figure 3.5 illustrates the imported layout to Cadence.

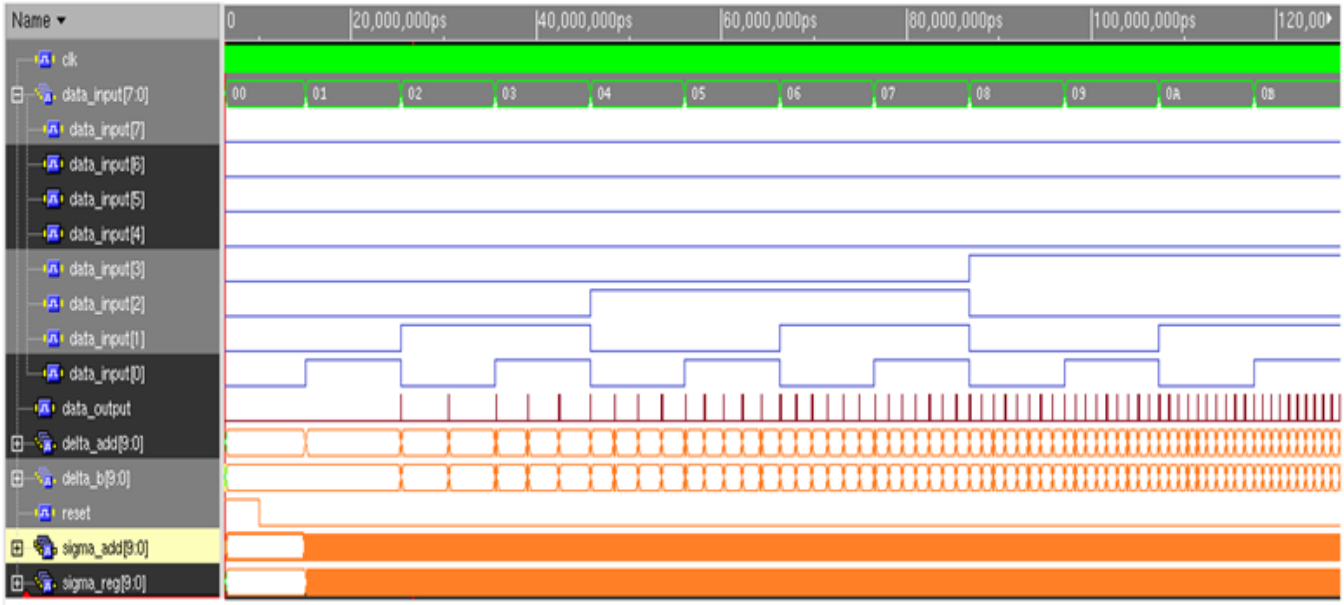

Figure 3.3: ΔΣ Digital Modulator Verilog Code Waveforms with Inputs in Blue and Output in Red

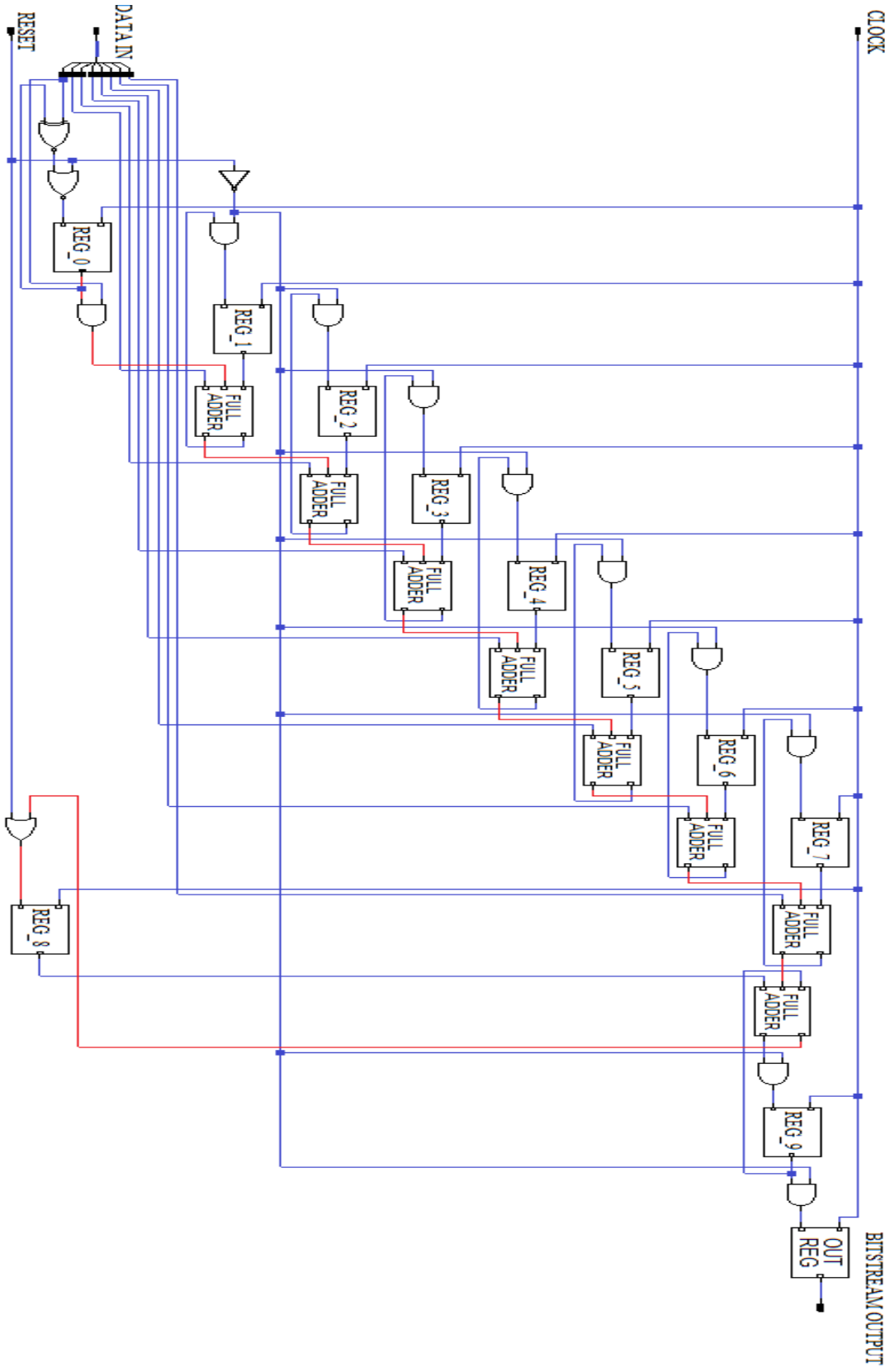

Figure 3.4: Proposed Digital ΔΣ Modulator Schematic with Critical Path in Red

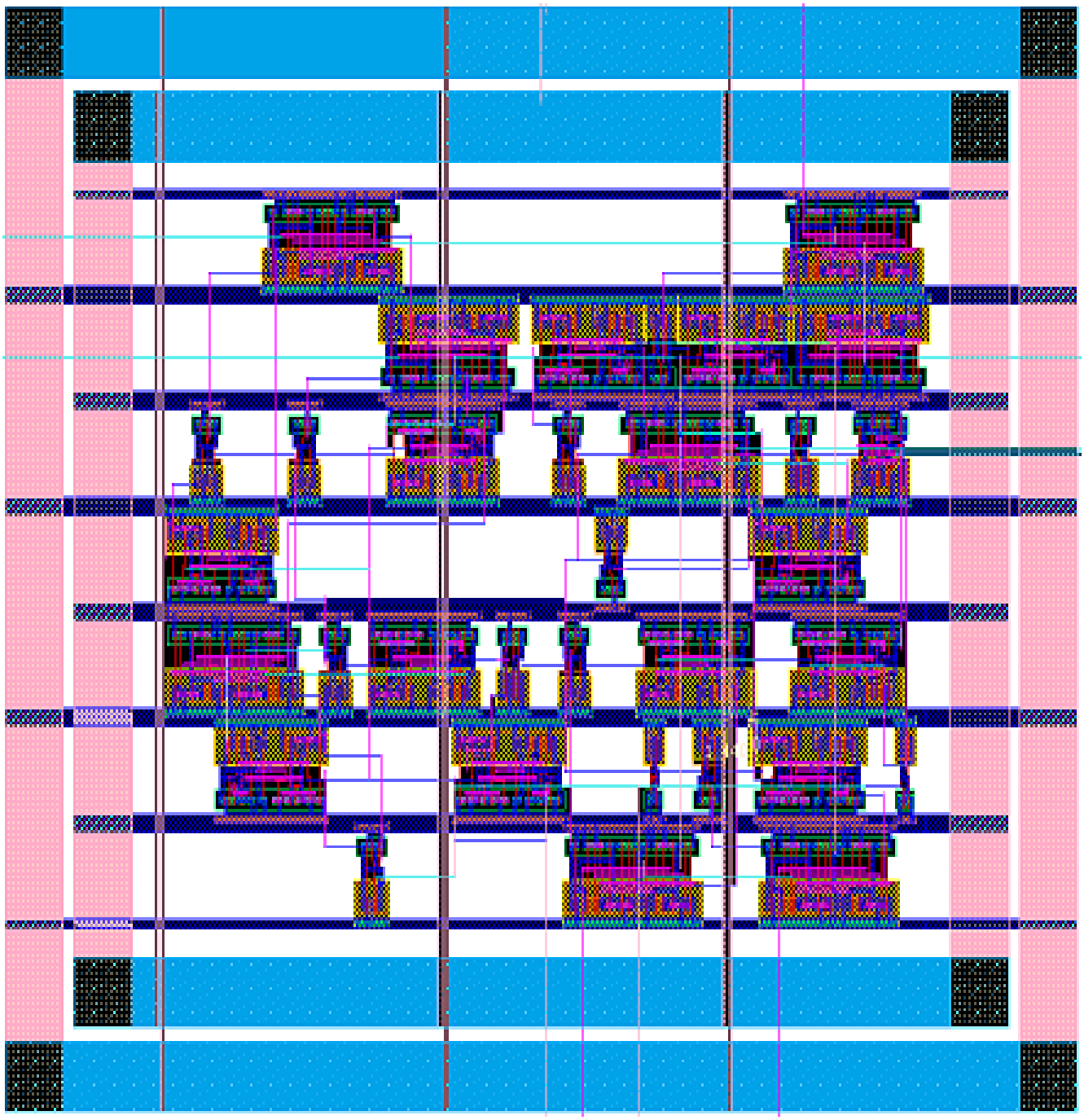

Figure 3.5: Proposed Digital ΔΣ Modulator Layout

3.1 LEVEL SHIFTERS

As the voltage operation regime for the proposed ΔΣ Modulator is at subthreshold levels, the output needs to be incremented to superthreshold levels. Consequently, two Level Shifters (LS) were placed to interface low voltage circuits with high voltage circuits. The traditional current mirror LS design is well suited for a wide voltage range conversion with regards to performance and presents an improved performance with a stable current driving capability [21]. Therefore, the most suitable LS for the ΔΣ DAC design is the current mirror LS shown in Figure 3.6. However, traditional current mirror LS present high power consumption because a leakage path forms through either PFET in the current mirror or one of the NFET pull down devices, which is always on in static sense [21]. In order to overcome the high power consumption problem of traditional level shifter, the ASIC Team of the University of Texas at El Paso (UTEP) proposed in [22] an improved specified subthreshold current mirror LS, which is still pending for publication. Both LS are used in the design in order to analyze the different driving capability at a varying voltage from subthreshold to superthreshold voltage levels.

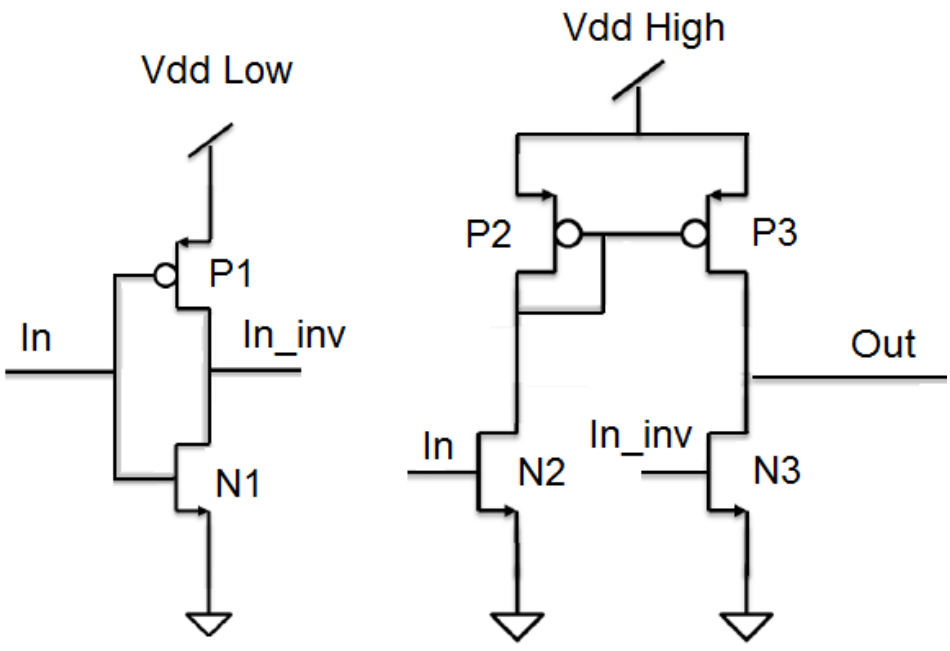

Figure 3.6: Traditional Constant Current Mirror Level Shifter

The basic operation for both Current Mirror level shifters is explained as follows: At the beginning of operation, consider the input "In" of the low source supply inverter as a 0, hence the output In_inv is 1. As gate voltage of N2 is lower than gate voltage of N3, N2 is turned off and consequently P3 and P2, hence the output voltage is 0. As the input of the low source supply inverter changes to 1, N3 is turned off and N2 is on and the voltage obtained at the output is 1 at high voltage. Nevertheless, P2 is connected as a diode typology, the current, which passes through P2, sets the value for V_{gs} for P2 and P3 and consequently if P2 and P3 are identical in properties, the current of P3 is equal to the current of P2. Furthermore, In order to operate at subthreshold levels the drive strength between the PMOS and NMOS needs to be similar. The traditional level shifter was designed by weakening the pull-up PMOS transistor drive strength or increasing the drive strength of the pull-down NMOS. Table 3.3 illustrates the values of the transistors for the traditional current mirror level shifter.

Table 3.3: Traditional Current Mirror Level Shifter transistors values.

Transitor	W	L	Multiplier
P1	1.98 µm	240 nm	1
P2	360 nm	240 nm	1
P3	360 nm	240 nm	1
N1	1.02 µm	240 nm	1
N2	4.98 µm	240 nm	1
N3	4.98 µm	240 nm	1

The layout for the traditional current mirror LS is found in Figure 3.7. Additionally, Figure 3.8 illustrates the layout for the improved current mirror LS proposed in [22]. Both layouts were designed in Cadence Virtuoso tool.

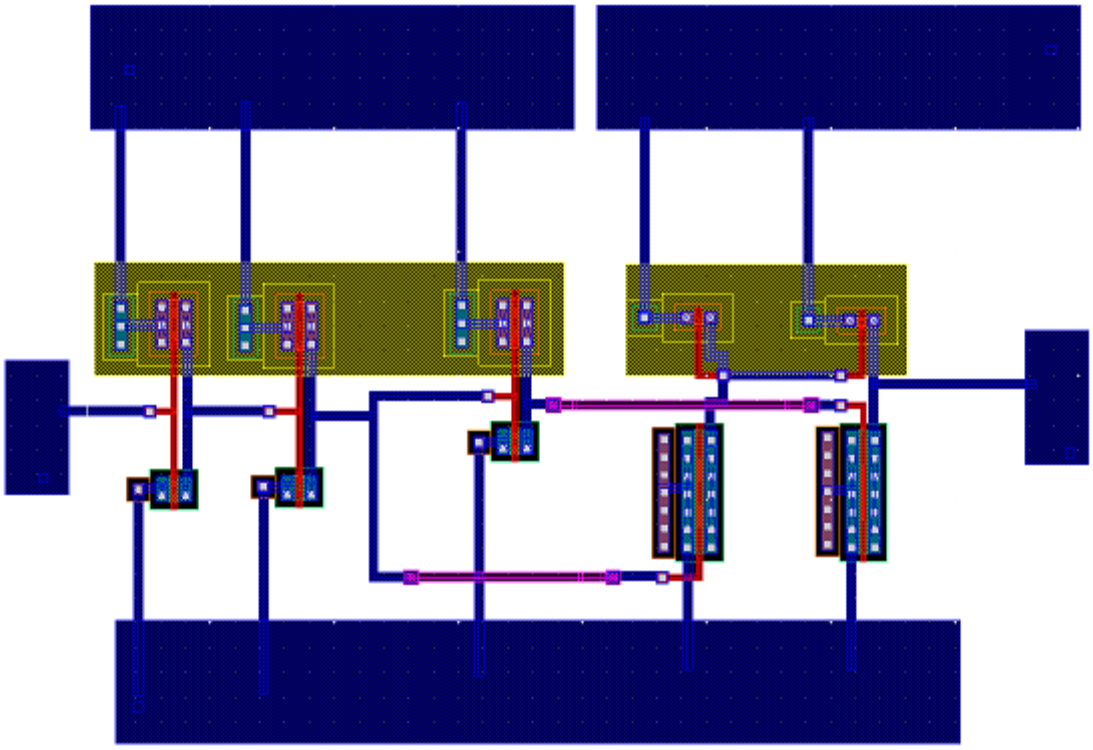

Figure 3.7: Traditional Constant Current Mirror Level Shifter Layout

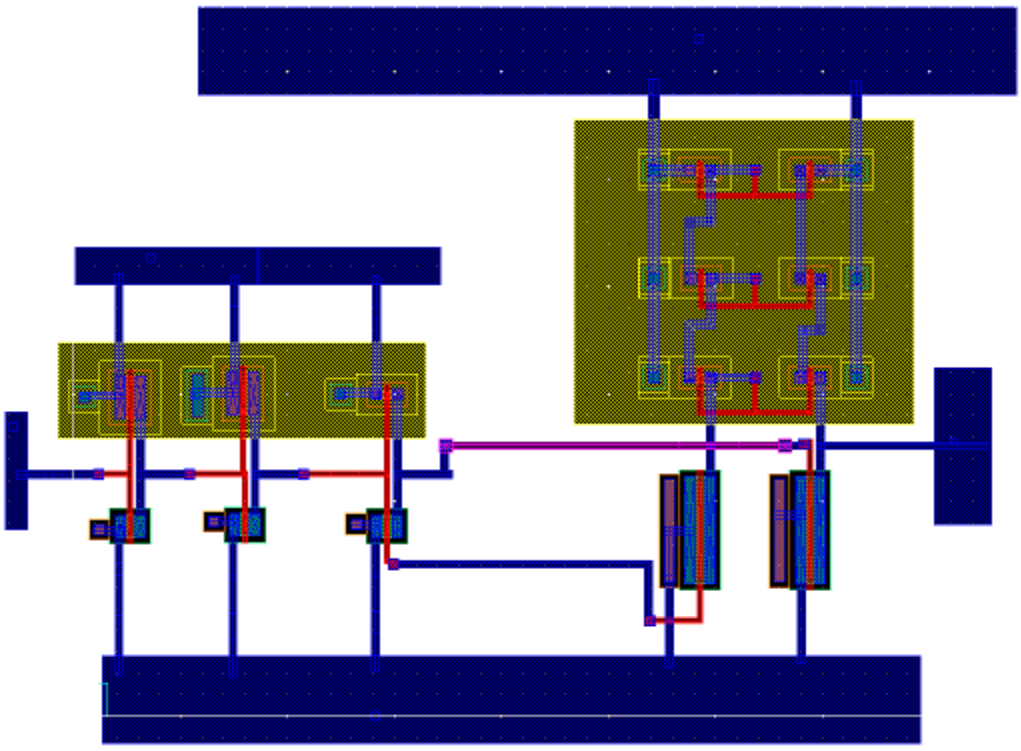

Figure 3.8: Proposed Constant Current Mirror Level Shifter Layout

3.2 OUTPUT BUFFER

As the output current on both levels shifters is not sufficient to drive the large output load. Buffers were designed and implemented from the level shifters output in order to boost the current capability of the output signal of the proposed ΔΣ DAC. Buffers are shown in Figure 3.9. Additionally, each next buffer is four times larger than previous one. Buffers sizing is illustrated in Table 3.4.

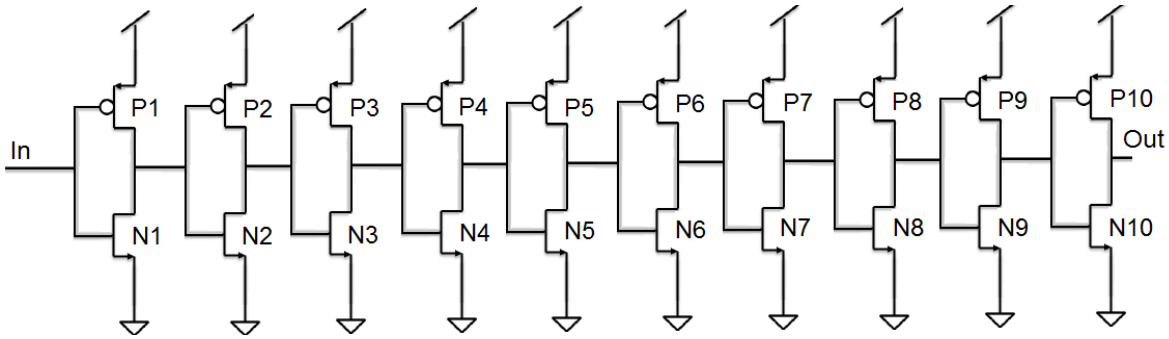

Figure 3.9: Output buffers

Table 3.4: Presented Boosting Output Buffer Sizing

Transistor	W	L	Multiplier
P1 & P2	1 µm	240 nm	1
P3 & P4	4 µm	240 nm	1
P5 & P6	4 µm	240 nm	4
P7 & P8	4 µm	240 nm	16
P9 & P10	4 µm	240 nm	64
N1 & N2	4 µm	240 nm	1
N3 & N4	4 µm	240 nm	1
N5 & N6	4 µm	240 nm	2
N7 & N8	4 µm	240 nm	8
N9 & N10	4 µm	240 nm	32

The layout for the buffer is found in Figure 3.10. Additionally, the use of multipliers was implemented in order to obtain a lean design.

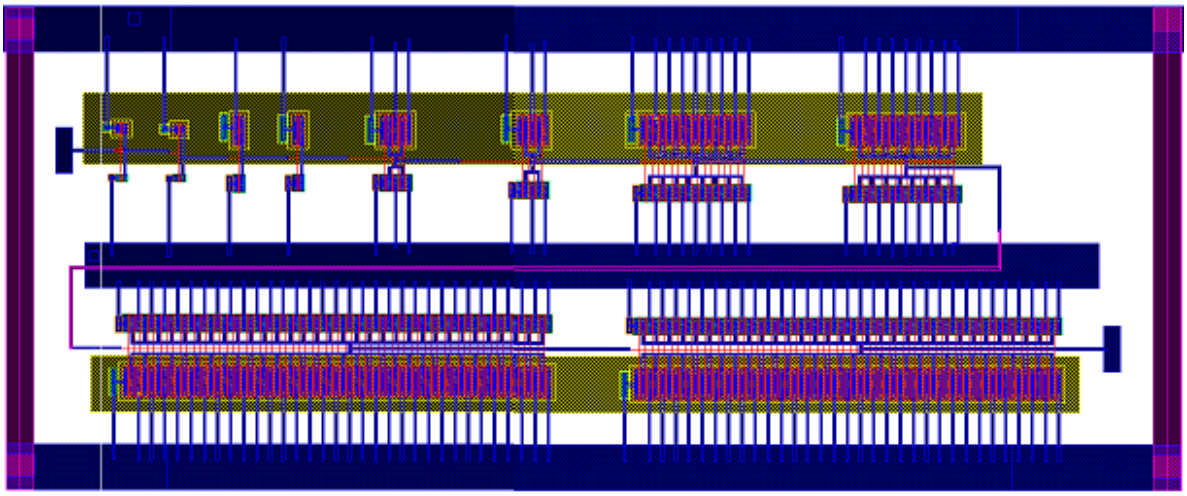

Figure 3.10: Output Buffers Layout

The layout of the entire proposed ΔΣ DAC is shown in Figure 3.11 occupying an area of 180µm x 450µm.

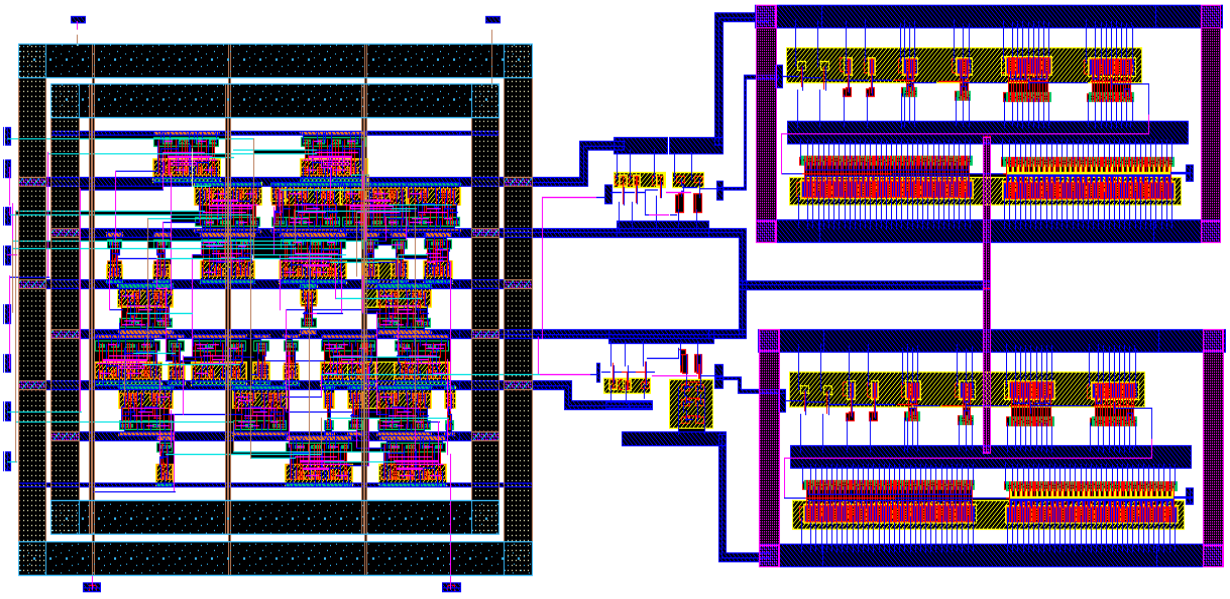

Figure 3.11: Proposed ΔΣ DAC Layout

3.3 LOW PASS FILTER

The pulse density bitstream output of the proposed ΔΣ Modulator is directly proportionate to ΔΣ DAC input value. However, to generate an analog output corresponding to the digital input, the output bitstream is passed through an analog resistive capacitive Low Pass Filter (LPF). The output voltage for the LPF is given by:

$$|Vout| = |Vin| \frac{1}{\sqrt{1+(\omega RC)^2}} \tag{3.1}$$

As the bitstream frequency changes, the value of the resistor remains constant; however, the capacitive reactance of the capacitor varies, which at low frequencies the capacitive reactance increases over the value of the resistor, hence an increment in the voltage across the capacitor occurs. However, at high frequencies the capacitive reactance decreases below the value of the resistor; therefore, the voltage across the capacitor decreases.

Consequently, the LPF attenuates high frequencies and passes low frequencies with no attenuation. In frequency responsive circuits, the cutoff frequency (f_c) determines the boundary at which frequency the circuits begin to attenuate the signal, as illustrated in Figure 3.12. [23]. The f_c is determined by:

$$f_c = \frac{1}{2\pi RC} \tag{3.2}$$

Additionally, at f_c break point, the output signal is attenuated to 70.7% of the input value or -3dB of gain voltage allowed to pass through LPF. After the f_c break point, the magnitude decreases obtaining a slope of -20dB per decade, as shown in Figure 3.12. Magnitude is given by:

$$dB = -20\log\frac{Vout}{Vin} \tag{3.3}$$

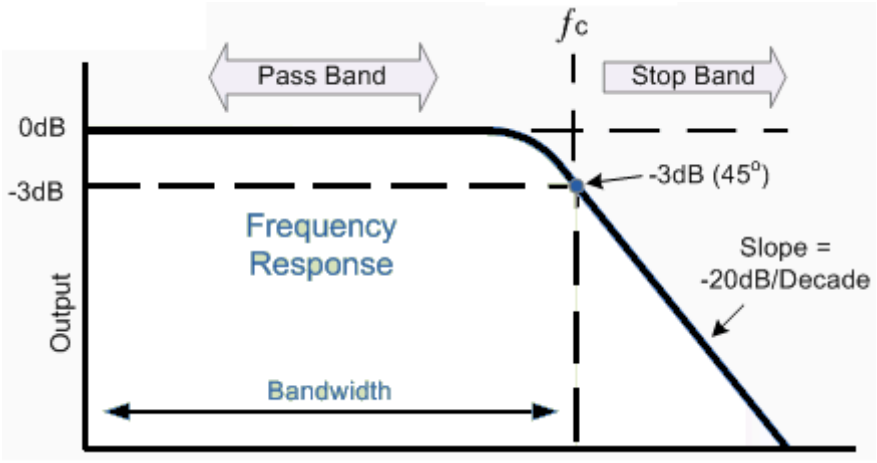

Figure 3.12: Low Pass Filter Bode Plot.

The required attenuation (or Maximum ripple allowed) for the designed LPF has to be equal to or less than the corresponding value of 1 LSB and has to be achieved at the lowest possible frequency. Additionally, to keep the design of LPF reasonable is assumed that DAC input is always above 5% of the total input [13], which for an 8-bit DAC the 5% input value is 0000 1100. In order to obtain the required attenuation at the resulting lowest possible frequency the RC time constant of the filter is obtained by the combination of Equations. (3.1) and (3.3) and is given by:

$$RC = \frac{1}{2\pi f_{5\%}} \sqrt{\frac{2^{n*2}}{Vref^2} - 1} \tag{3.4}$$

Where $f_{5\%}$ is the bitstream output frequency at 5% of the total input, n is the number of DAC bits and V_{ref} is the reference voltage at the output of the level shifter.

In addition, the settling time of the ΔΣ DAC depends on the resolution, which is 8 bits, and the RC value obtained from Equation (3.4). The charging capacitor formula is defined by:

$$V_c = V_{ref}\left(1 - e^{\frac{-t}{RC}}\right) \tag{3.5}$$

As the smallest voltage change that can be made in the DAC's output is equal to 1 LSB and is defined in Equation (2.7). Furthermore, combining both Equations (3.5) and (2.7) leads to the minimal time needed to settle within the required accuracy [24], which can be obtained by:

$$V_c = V_{ref}\left(1 - e^{-\frac{t}{RC}}\right) = V_{ref} - V_{LSB} = V_{ref} - \frac{V_{ref}}{2^n} = V_{ref}\left(1 - \frac{1}{2^n}\right) \tag{3.5}$$

Which leads to

$$e^{-\frac{t}{RC}} = \frac{1}{2^n} \tag{3.6}$$

Or

$$t = RC \ln 2^n \tag{3.7}$$

Where t is the settling time, RC is time constant obtained from Equation (3.4) and n is the number of bits.

Chapter 4: Results

The presented ΔΣ DAC consists of three main blocks: ΔΣ digital modulator, level shifter and reconstruction low pass filter, which were designed with different design tools. In order to guarantee the proper operation, circuit simulations were done at transistor level using SPICE netlist models. Additionally, design entry was done with Virtuoso Schematic Editor and the simulations at block-level and chip-level were performed with Virtuoso Spectre Simulator, which is able to interpret and simulate SPICE netlist. Furthermore, the physical design layout was made at the symbolic level with Virtuoso Layout Suit, which provides Diva, a parasitic extraction tool. The Diva tool was used to obtain the extracted netlist, which includes parasitic capacitors and resistors present in the design. In order to achieve more realistic results, post-layout simulations were performed. In addition, Virtuoso Spectre Simulator provides an OCEAN script, which was used to speed up the analysis without the necessity of a graphical user interface. In the present chapter, the simulated results for the 8-bit ΔΣ DAC were presented.

4.1 Simulation Results

The maximum operation frequency of the ΔΣ Digital Modulator as well as the maximum sampling frequency of the proposed ΔΣ DAC are important parameters to be determined. The maximum operation frequency is obtained by measuring the total delay of the critical path of the ΔΣ Digital Modulator at different voltages, as shown in Table 4.1

As mentioned in chapter three, the maximum sampling frequency of the proposed ΔΣ DAC is determined by the settling time of the Low Pass Filter (LPF). Therefore, Equations (3.4) and (3.7) were used to calculate the RC time constant and subsequently the settling time. Table 4.2 illustrates, the settling time and the values of off-chip resistors and capacitors used to obtain the RC time constant of the LPF.

Table 4.1: Delay and Maximum Operation Frequency of the ΔΣ Digital Modulator

Voltage	Critical Path Delay	Maximum Operation Frequency
150 mv	331.862 us	3013.30 Hz
250 mv	59.224 us	16.88 KHz
350 mv	9.707 us	103.01 KHz
500 mv	729.60 ns	1.37 MHz
700 mv	60.543 ns	16.51 MHz
1.2 v	9.203 ns	108.65 MHz
2.5 v	3.41 ns	293.23 MHz

Table 4.2: Maximum Sampling Frequency of the ΔΣ DAC and RC Values

Voltage	Maximum Sampling Frequency	RC Value	Resistor	Capacitor
150 mv	1.58 Hz	0.11408	110 MΩ	1 nf
250 mv	8.43 Hz	0.02138	22 MΩ	1 nf
350 mv	52.71 Hz	0.00342	3.3 MΩ	1 nf
500 mv	526.94 Hz	0.000342	330 KΩ	1 nf
700 mv	8.43 KHz	21.3813 u	22 KΩ	1 nf
1.2 v	56.91 KHz	3.1687 u	3.3 KΩ	1 nf
2.5 v	154.38 KHz	1.1680 u	1.2 KΩ	1 nf

In order to stimulate the circuit, the setup of the testbench was designed with a changing digital input generated from eight ideal pulse sources, which gives the maximum power consumption case. The maximum power consumption case is obtained by changing the data input from hexadecimal values of 00 to 80 to FF to 80 to 00. A series of input buffers were used after the ideal pulse sources to give more realistic rise and fall times. As demonstrated in Table 4.2, the maximum sampling frequency at

subthreshold voltage is low, in the order of Hertz. Therefore, to perform the analysis and obtain the power delay product for the ΔΣ Digital Modulator and also for the two level shifters (LS), a series of HSPICE netlists were designed; in each netlist the operating frequency of the ΔΣ Digital Modulator was increased for different voltage supplies. Table 4.3 illustrates the delay measurements of the ΔΣ Digital Modulator, the traditional LS and the proposed LS. Table 4.4 demonstrates the power measurement of the ΔΣ Digital Modulator, the traditional LS and the proposed LS.

The maximum operation frequency of the ΔΣ DAC is constrained by the critical path delay of the ΔΣ Modulator. However, other limitation exists in the ΔΣ DAC operation, which comes from the delay of the LS. The delay of the traditional LS does not have a big impact on the ΔΣ DAC but consumes more power. On the other hand, the proposed LS was designed to operate at low voltage/low frequency and hence consume low power. However, the proposed LS operating at high frequency adds more delay to the output, which limits the functionality of the ΔΣ DAC. As the operation frequency of the ΔΣ Modulator increases, the cascode current mirror architecture of the proposed LS is not able to drive all the output-signal swing possible before transistors enter the triode region. Figures 4.3, 4.4 and 4.5 demonstrate the decreasing output-signal swing capability as the frequency increase. Level shifters input are shown in blue, the output of the traditional and proposed level shifters are shown in pink and red respectively.

Table 4.3: Delay Measurements of the ΔΣ Digital Modulator, the Traditional and Proposed LS

Voltage	Operating Frequency	Sampling Frequency	Modulator Delay	Traditional LS Delay	Proposed LS Delay
150 mv	3 kHz	1.58 Hz	1.04 ms	18.506 µs	32.987 µs
250 mv	16 kHz	8.43 Hz	138.32 µs	3.68 µs	4.2712 µs
350 mv	100 kHz	52.71 Hz	21.5 µs	551.82 µs	604.78 ns
500 mv	1 Mhz	526.94 Hz	2.14 µs	41.89 ns	45.266 ns
700 mv	16 Mhz	8.43 KHz	135.94 ns	4.6365 ns	7.1218 ns
1.2 v	100 Mhz	56.91 KHz	28.865 ns	1.183 ns	N/A[1]
2.5 v	293 Mhz	154.38 KHz	7.545 ns	850.52 ps	N/A

[1]Not Applicable

Table 4.4: Power Measurements of the ΔΣ Digital Modulator, the Traditional and Proposed LS

Voltage	Maximum Sampling Frequency	Modulator Power	Traditional LS Power	LS Traditional Buffer Power	Total Traditional LS Power	Proposed LS Power	LS Proposed Buffer Power	Total Proposed LS Power
150 mv	1.58 Hz	425 pW	14.7 nW	7.98 µW	8.00 µW	9.39 nW	6.39 µW	6.40 µW
250 mv	8.43 Hz	989 pW	97.5 nW	3.94 µW	4.04 µW	55.9 nW	4.43 µW	4.23 µW
350 mv	52.71 Hz	4.30 nW	695 nW	4.71 µW	5.41 µW	342 nW	3.88 µW	4.49 µW
500 mv	526.94 Hz	69.9 nW	10.1 µW	8.60 µW	18.8 µW	3.60 µW	8.57 µW	12.2 µW
700 mv	8.43 KHz	2.27 µW	100 µW	250 µW	352 µW	4.80 µW	456 µW	463 µW
1.2 v	56.91KHz	58.8 µW	223 µW	1.07 mW	1.36 mW	N/A[1]	N/A	N/A
2.5 v	154.38KHz	2.71 mW	190 µW	2.23 mW	5.13 mW	N/A	N/A	N/A

[1]Not Applicable

The power consumption is illustrated in a logarithmic plot in Figure 4.1

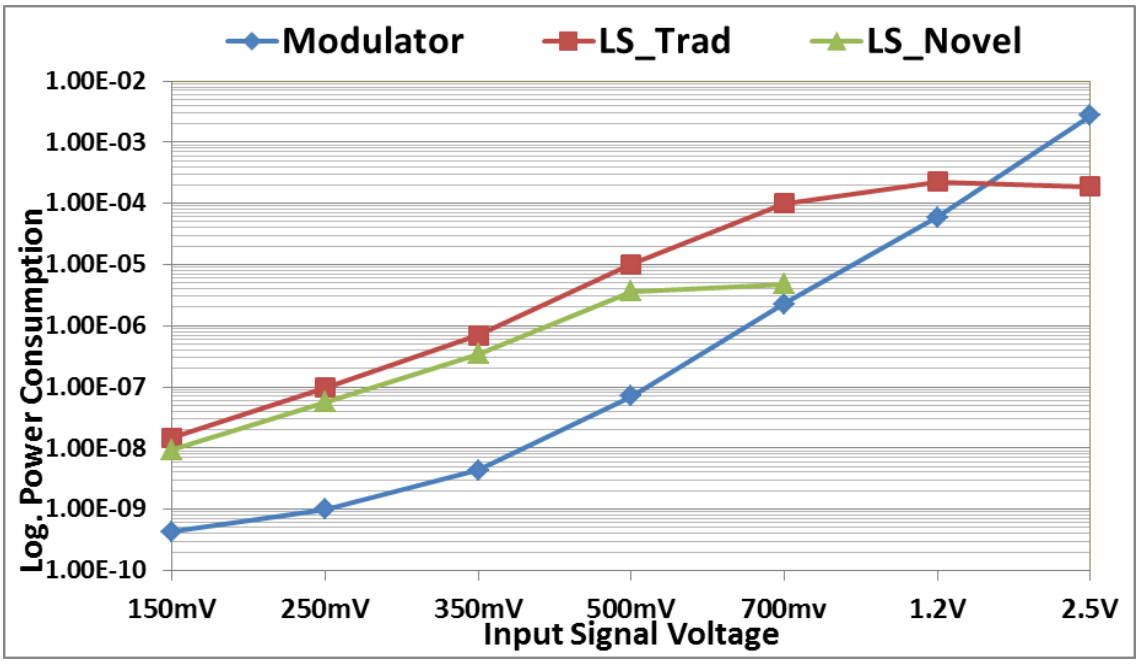

Figure 4.1: Power Consumption

The power delay product is determined by multiplying the power and the delay of the ΔΣ Digital Modulator and the level shifters. Figure 4.2 illustrates the Power Delay Product plot.

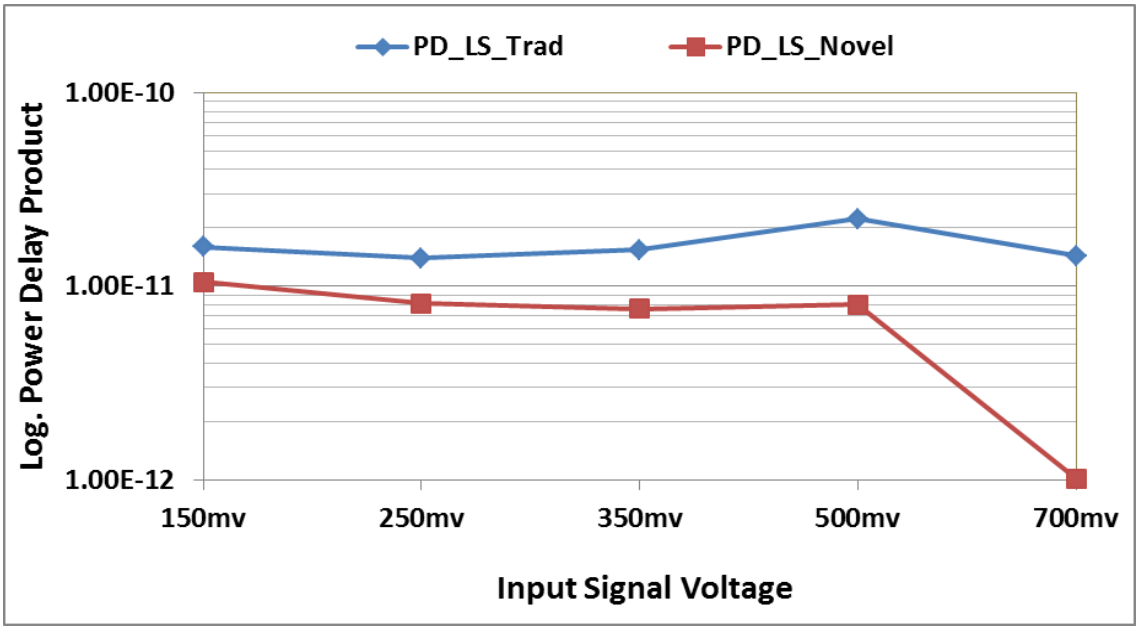

Figure 4.2: Power Delay Product

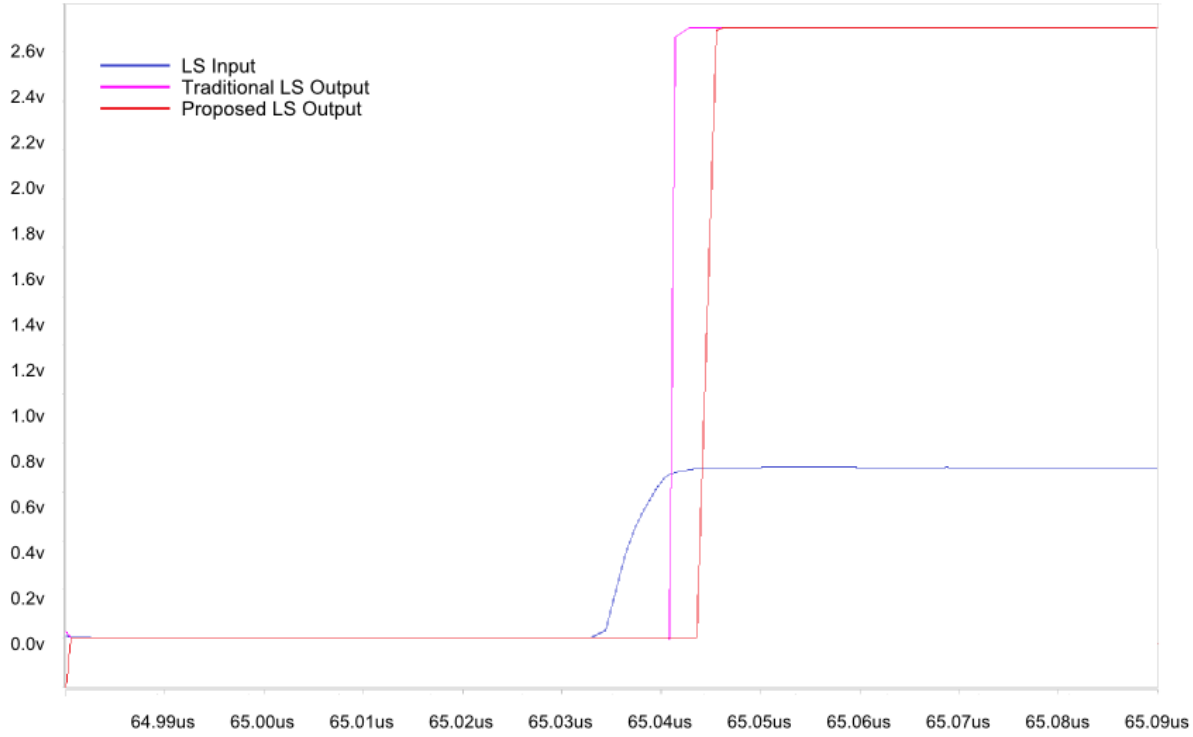

Figure 4.3: Level Shifters Signal-Output Swing at 700mv with an Operating Frequency of 16 MHz.

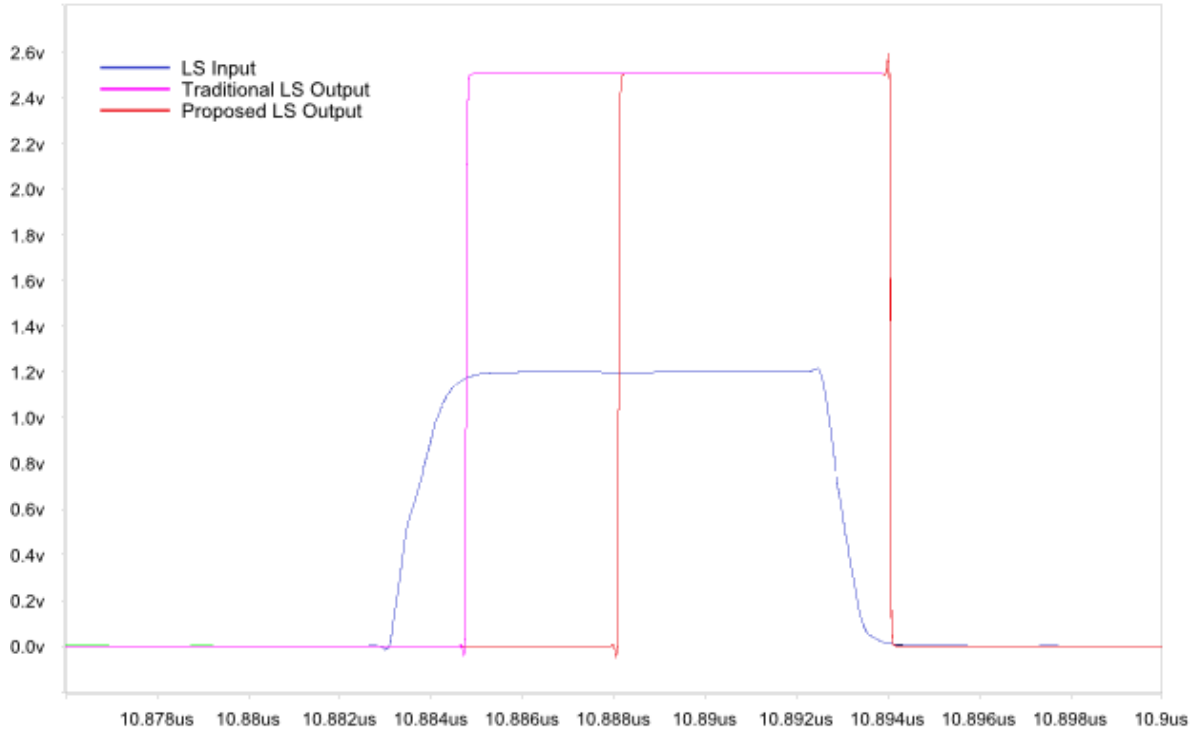

Figure 4.4: Level Shifters Signal-Output Swing at 1.2v with an Operating Frequency of 100 MHz

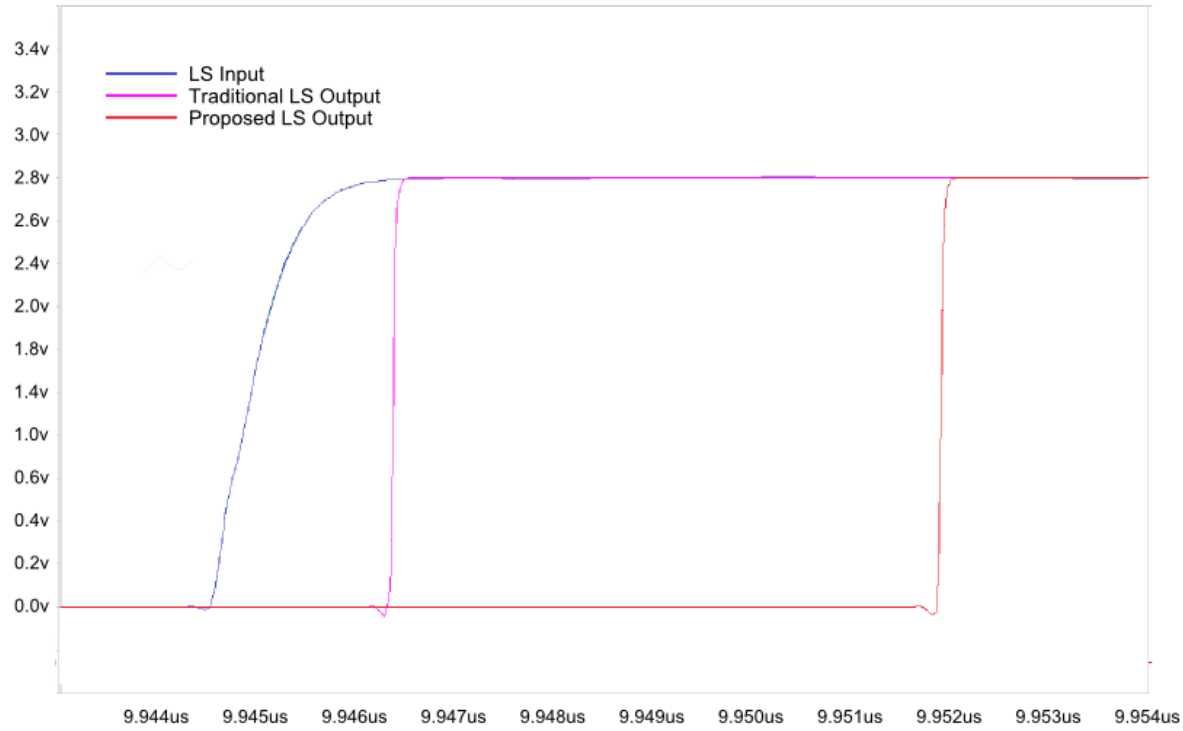

Figure 4.5: Level Shifters Signal-Output Swing at 2.5v with an Operating Frequency of 293 MHz.

The functionality of the ΔΣ DAC was verified by post-layout simulations. Additionally, post-layout simulations were performed using OCEAN scripts. The setup of the testbench was designed with a changing digital input generated from eight ideal pulse sources, which change from FF to C0 to 80 to 40 to 00. Table 4.5 shows the voltage and operating frequency of the ΔΣ Modulator, sampling frequency of the ΔΣ DAC and off-chip resistor and capacitor values for the LPF.

Table 4.5: Testbech Values for Functionality Test

Voltage	Operating Frequency	Sampling Frequency	Resistor value	Capacitor Value
350mv	100 KHz	50 Hz	3.3 MΩ	1 nf
2.5v	100 MHz	50 KHz	330 Ω	1 nf

Simulations with the extracted netlist of amplified output bitstream from the ΔΣ Modulator are shown in Figure 4.6 and 4.7. Additionally, Figure 4.7 illustrates the decreased output-signal swing capability of the proposed LS at higher frequencies. Output of the traditional and proposed level shifters are shown in blue and red respectively.

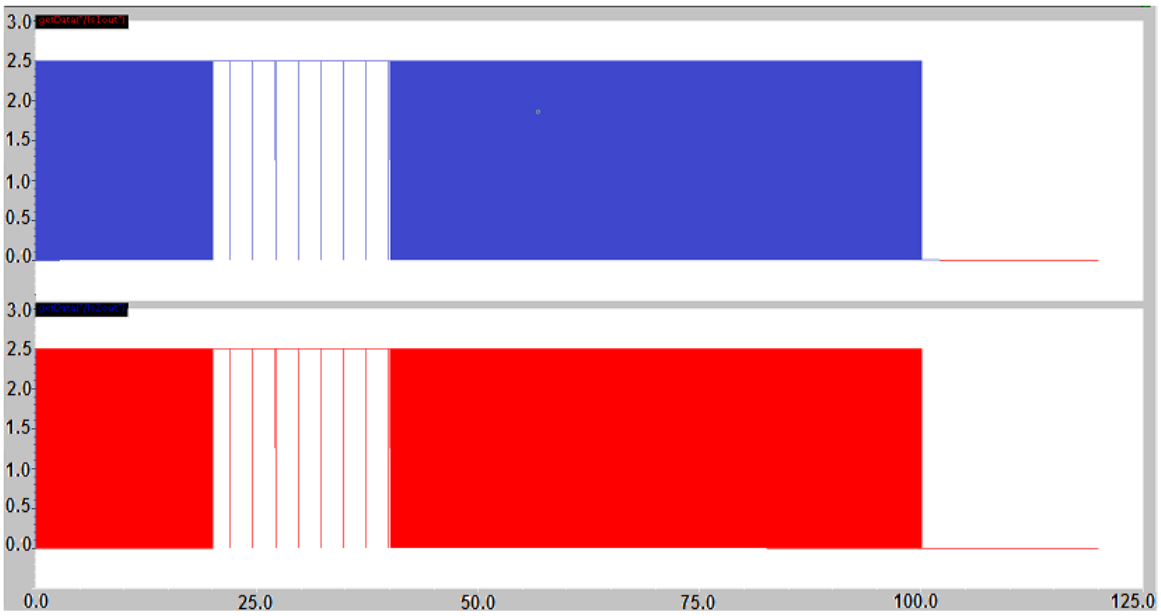

Figure 4.6: Post Layout Level Shifters Outputs at 350mv Operating at Frequency of 100 Khz

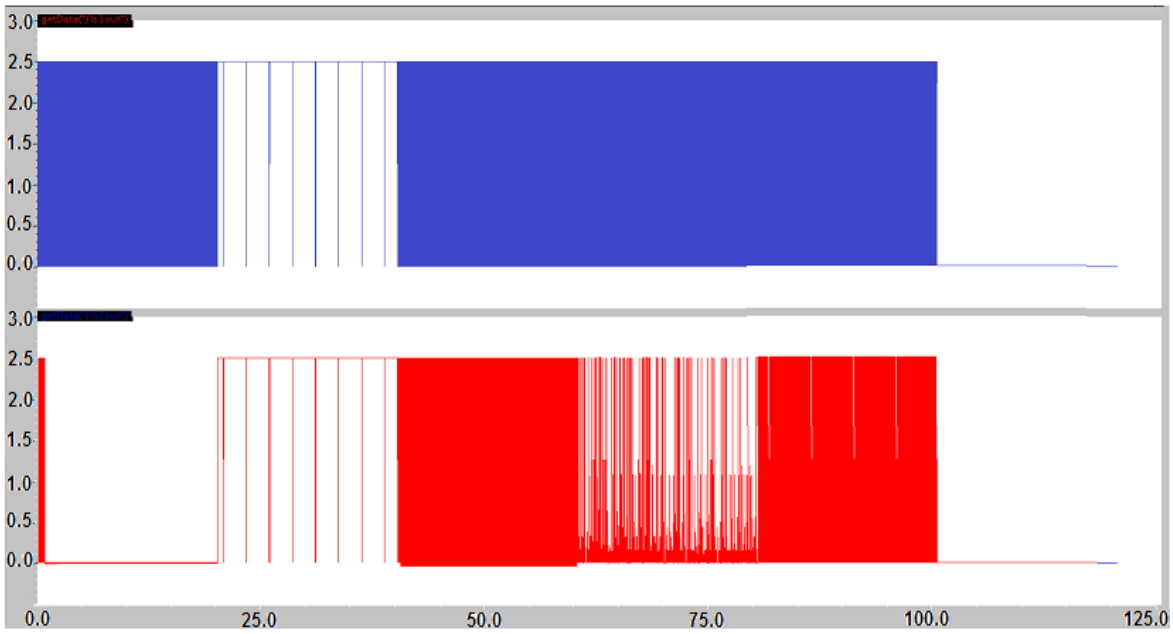

Figure 4.7: Post Layout Level Shifters Outputs at 2.5v with decreasing output signal swing of proposed LS an Operating Frequency of 10 Mhz

The analog outputs of the LPFs obtained from post-layout simulations are shown in figures 4.8 and 4.9. The outputs of both LPF in figure 4.8 are overlapped and represent an equal plot. At high frequency, the LPF output connected to the traditional LS presents an accurate plot but the LPF output connected to the proposed LS presents a degraded plot, as illustrated in Figure 4.9.

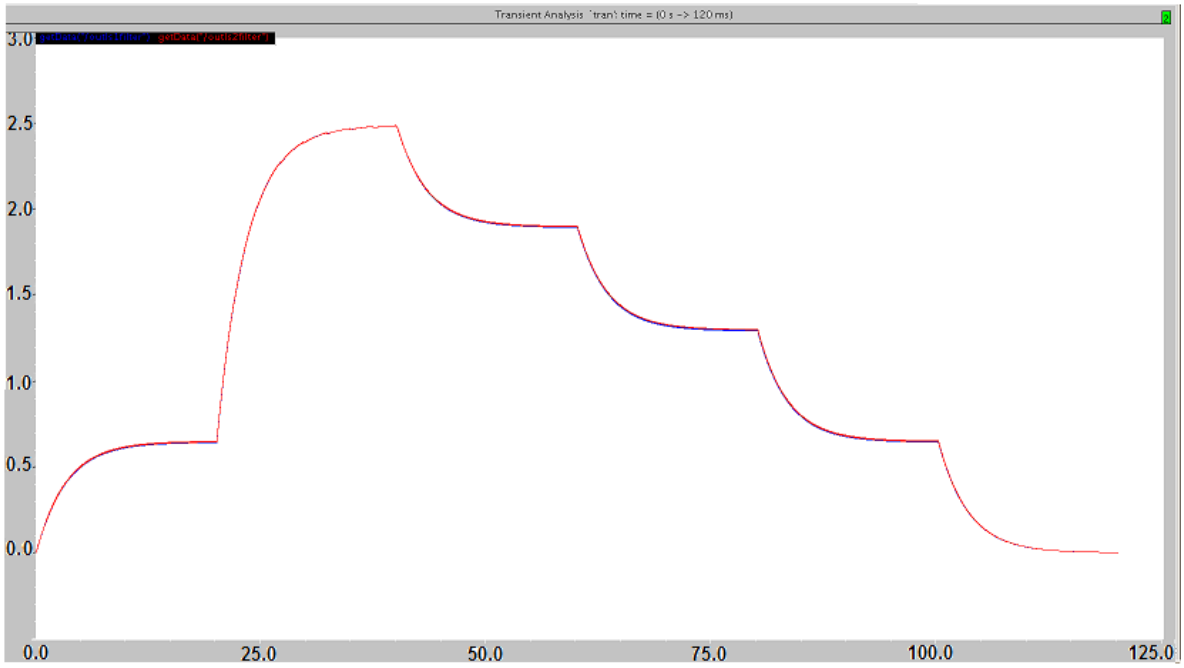

Figure 4.8: Post Layout LPF output simulations at 350mv with an Operating Frequency 100 Khz

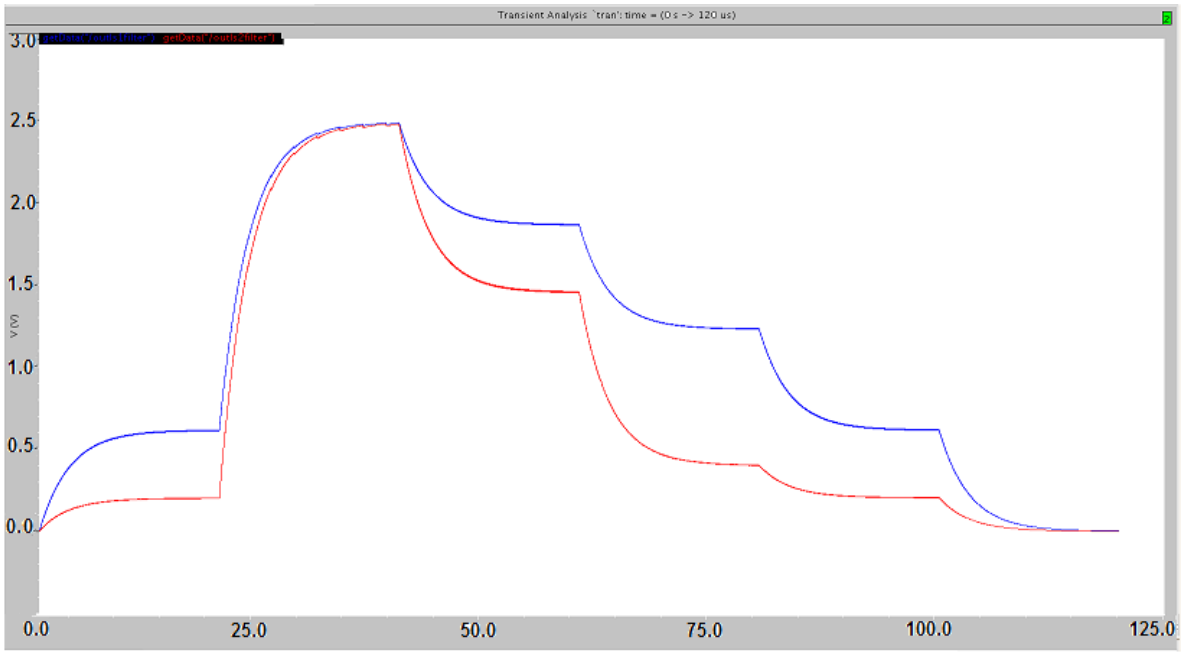

Figure 4.9: Post Layout LPF output simulations at 2.5v with an Operating Frequency 10 Mhz

Chapter 5: Conclusion and Future Work

5.1 CONCLUSION

A low power 8 bit ΔΣ DAC was designed with standard TSMC 0.25µm CMOS technology. The proposed converter can operate at different supply voltages, which can range from 150 mv to 2.5v. Additionally, the ΔΣ DAC can be configured for different operating frequencies via the off chip voltage source and resistor and capacitor of the low pass filter. The main results of the DAC are the chip area 180µm x 450µm and power consumption is 4.49 µW at 350mv, which is compared with previous work as shown in Table 5.1.

Table 5.1: Power Comparison to Previous Work

	Year	Supply Voltage	Technology	Power	Sampling Frequency	Area
M.Annovazzi[17]	2002	3.3 v	0.35µm CMOS	28 mW	20 KHz	2.9 mm^2
K. Lee [18]	2009	.7 v to 1.8 v	0.35µm CMOS	2.6 mW	24 KHz	3.52 mm^2
K.-P. Pun[19]	2007	.5 v	0.18µm CMOS	300 µW	25 KHz	0.6 mm^2
ΔΣ DAC	2011	.15 v	0.25µm CMOS	6.40 µW	1.58 Hz	.081 mm^2
ΔΣ DAC	2011	.25 v	0.25µm CMOS	4.23 µW	8.43 Hz	.081 mm^2
ΔΣ DAC	2011	.35 v	0.25µm CMOS	4.49 µW	52.71 Hz	.081 mm^2
ΔΣ DAC	2011	.5 v	0.25µm CMOS	12.2 µW	526.94 Hz	.081 mm^2
ΔΣ DAC	2011	.7 v	0.25µm CMOS	352 µW	8.43 KHz	.081 mm^2
ΔΣ DAC	2011	1.2 v	0.25µm CMOS	1.36 mW	56.91 KHz	.081 mm^2
ΔΣ DAC	2011	2.5 v	0.25µm CMOS	5.13 mW	154.38 KHz	.081 mm^2

5.2 FUTURE WORK

The immediate future work needed is the full characterization of the ΔΣ DAC with a fabricated IC. As the frequencies at subthreshold levels are in the order of Hertz, the total time for a full scale simulation passing for all the possible values will be in the order of seconds, and with the fact that the Spectre simulation tool and HSPICE were designed to perform circuit analysis in the order of pico seconds to some hundred milliseconds, the real time for a full range simulations will take more than a couple of days. In addition, an optimization can be done by improving the ΔΣ DAC to operate at higher frequencies with subthreshold voltage levels, as well, reduce the power consumption by upgrading the buffers, which holds the major fraction in the proposed ΔΣ DAC.

References

[1] A. Chavan, E. MacDonald, y B. Graniello, “Optimized Charge Pumps for Subthreshold Operation”, 2009.

[2] R. Gonzalez, B.M. Gordon, y M.A. Horowitz, “Supply and threshold voltage scaling for low power CMOS”, *Solid-State Circuits, IEEE Journal of*, vol. 32, 1997, pág. 1210–1216.

[3] Amitava Chatterjee,, “An Investigation of the Impact of Technology Scaling on Power Wasted asShort-Ciruit Current in Low Voltage Staitic CMOS Circuits”, Ago. 1996.

[4] M. Horowitz, T. Indermaur, y R. Gonzalez, “Low-power digital design”, *Low Power Electronics, 1994. Digest of Technical Papers., IEEE Symposium*, 1994, pág. 8–11.

[5] A. Wang, A.P. Chandrakasan, y S.V. Kosonocky, “Optimal supply and threshold scaling for subthreshold CMOS circuits”, *isvlsi*, 2002, pág. 0007.

[6] D. Sengupta y R. Saleh, “Power-delay metrics revisited for 90nm cmos technology”, 2005.

[7] Maloberti, Franco, *Data Converters*, Pavia University, Italia: Springer, 2007.

[8] B. Razavi, *Principles of Data Conversion System Design*, Wiley-IEEE Press, 2004.

[9] D. Johns, Ken Martin, *Analog Integrated Circuit Design*, Wiley, 1996.

[10] J. Lenk, *Simplified Design of Data Converters (EDN Series for Design Engineers)*, Newnes, Ap.

[11] G.C. Temes, with Richard Schreier, *Understanding Delta-Sigma Data Converters*, Wiley-IEEE Press, 2004.

[12] James C. Candy, *Oversampling Delta-Sigma Data Converters: Theory, Design, and Simulation*, Wiley-IEEE Press, 1991.

[13] H. Cheung, y S. Raj,, “Implementation of 12-bit delta-sigma DACwith MSC12xx controller”, *Analog Applications Journal*, 1Q.

[14] Frank Ohnhäuser,, “Digital-to-Analog Converter (DAC)Converter Types”.

[15] D.A. Mercer, “Digital to Analog Converter Design”, 2008.

[16] S.R. Norsworthy y R. Schreier, Gabor C. Temes, *Delta-Sigma Data Converters: Theory, Design, and Simulation*, Wiley-IEEE Press, 1996.

[17] M. Annovazzi, V. Colonna, G. Gandolfi, F. Stefani, y A. Baschirotto, “A low-power 98-dB multibit audio DAC in a standard 3.3-V 0.35-μm CMOS technology”, *Solid-State Circuits, IEEE Journal of*, vol. 37, 2002, pág. 825–834.

[18] K. Lee, Q. Meng, T. Sugimoto, K. Hamashita, K. Takasuka, S. Takeuchi, U.-K. Moon, y G.C. Temes, “A 0.8 V, 2.6 mW, 88 dB Dual-Channel Audio Delta-Sigma D/A Converter With Headphone Driver”, *IEEE Journal of Solid-State Circuits*, vol. 44, 2009, págs. 916-927.

[19] K.-P. Pun, S. Chatterjee, y P.R. Kinget, “A 0.5-V 74-dB SNDR 25-kHz Continuous-Time Delta-Sigma Modulator With a Return-to-Open DAC”, *IEEE Journal of Solid-State Circuits*, vol. 42, 2007, págs. 496-507.

[20] “Delta-Sigma ADC”, Oct. 2009.

[21] A. Chavan y E. MacDonald,, “Ultra Low Voltage Level Shifters to Interface Sub andSuper Threshold Reconfigurable Logic Cells”, *IEEEAC paper#1345*, Dic. 2007.

[22] E. MacDonald, with A. Chavan,, Miguel, “Level Converters Optimized to Interface Subthreshold and Superthreshold Ciruits”, *Pendent IEEE publication*.

[23] “Low Pass Filter”, *Electronics-Tutorial.ws*.

[24] F.B. Boschker, “Design of a 12bit 500Ms/s standalone charge redistribution Digital-to-Analog Converter”, 2008.

Appendix A Verilog

Verilog is a Hardware Description Language (HDL) developed to model electronics systems and used to describe behavioral model of digital circuits.

```
////////////////////////////////////////////////ΔΣ Modulator  Verilog Code//////////////////////////////////////////////////
`timescale 100 ps / 10 ps
module deltasigma8b(data_output, data_input, clk, reset);
output data_output;
input [7:0] data_input;
input clk;
input reset;
reg data_output;
wire       [9:0] delta_add;
wire       [9:0] sigma_add;
reg        [9:0] sigma_reg;
wire       [9:0] delta_b;
assign  delta_b = {sigma_reg[9], sigma_reg[9], 8'b0} ;
assign  delta_add = data_input + delta_b;
assign  sigma_add = delta_add + sigma_reg;
always @(posedge clk)
  if(reset) data_output <=  1'b0;
  else      data_output <=  sigma_reg[9];
always @(posedge clk)
  if(reset) sigma_reg <=  9'b1_0000_0000;
  else      sigma_reg <=  sigma_add;

endmodule
```

```
//////////////////////Testbench used to stimulate the Verilog Code of the ΔΣ Modulator//////////////////
timescale 1ns / 1ps
module dstb();
reg            clk;
reg           reset;
reg   [11:0]   data_input;
////////////////////////////////////////////////
// clock stimilus
////////////////////////////////////////////////
initial              clk = 0;
initial forever #20   clk = ~clk;
reg  [7:0]       sample_counter;
always @(posedge clk)
     if(reset)  sample_counter = 0;
     else       sample_counter = sample_counter + 1;
wire            sample_clk = sample_counter[7];
////////////////////////////////////////////////
// Reset and completion logic
////////////////////////////////////////////////
initial
 begin
              reset        <= 1;
              data_input        <= 0;
#10000          reset        <= 0;
              $display("resetN de-asserted");
              @(posedge sample_clk) data_input <= 1;
```

```
@(posedge sample_clk) data_input <= 2;
@(posedge sample_clk) data_input <= 3;
@(posedge sample_clk) data_input <= 4;
@(posedge sample_clk) data_input <= 5;
@(posedge sample_clk) data_input <= 6;
@(posedge sample_clk) data_input <= 7;
@(posedge sample_clk) data_input <= 8;
@(posedge sample_clk) data_input <= 9;
@(posedge sample_clk) data_input <= 10;
@(posedge sample_clk) data_input <= 11;
@(posedge sample_clk) data_input <= 12;
@(posedge sample_clk) data_input <= 13;
@(posedge sample_clk) data_input <= 14;
@(posedge sample_clk) data_input <= 15;
@(posedge sample_clk) data_input <= 16;
@(posedge sample_clk) data_input <= 17;
@(posedge sample_clk) data_input <= 18;
@(posedge sample_clk) data_input <= 19;
@(posedge sample_clk) data_input <= 20;
@(posedge sample_clk) data_input <= 21;
@(posedge sample_clk) data_input <= 22;
@(posedge sample_clk) data_input <= 23;
@(posedge sample_clk) data_input <= 24;
@(posedge sample_clk) data_input <= 25;
@(posedge sample_clk) data_input <= 26;
@(posedge sample_clk) data_input <= 27;
@(posedge sample_clk) data_input <= 28;
```

```
@(posedge sample_clk) data_input <= 29;
@(posedge sample_clk) data_input <= 30;
@(posedge sample_clk) data_input <= 31;
@(posedge sample_clk) data_input <= 32;
@(posedge sample_clk) data_input <= 33;
@(posedge sample_clk) data_input <= 34;
@(posedge sample_clk) data_input <= 35;
@(posedge sample_clk) data_input <= 36;
@(posedge sample_clk) data_input <= 37;
@(posedge sample_clk) data_input <= 38;
@(posedge sample_clk) data_input <= 39;
@(posedge sample_clk) data_input <= 40;
@(posedge sample_clk) data_input <= 41;
@(posedge sample_clk) data_input <= 42;
@(posedge sample_clk) data_input <= 43;
@(posedge sample_clk) data_input <= 44;
@(posedge sample_clk) data_input <= 45;
@(posedge sample_clk) data_input <= 46;
@(posedge sample_clk) data_input <= 47;
@(posedge sample_clk) data_input <= 48;
@(posedge sample_clk) data_input <= 49;
@(posedge sample_clk) data_input <= 50;
@(posedge sample_clk) data_input <= 51;
@(posedge sample_clk) data_input <= 52;
@(posedge sample_clk) data_input <= 53;
@(posedge sample_clk) data_input <= 54;
@(posedge sample_clk) data_input <= 55;
```

```
@(posedge sample_clk) data_input <= 56;
@(posedge sample_clk) data_input <= 57;
@(posedge sample_clk) data_input <= 58;
@(posedge sample_clk) data_input <= 59;
@(posedge sample_clk) data_input <= 60;
@(posedge sample_clk) data_input <= 61;
@(posedge sample_clk) data_input <= 62;
@(posedge sample_clk) data_input <= 63;
@(posedge sample_clk) data_input <= 64;
@(posedge sample_clk) data_input <= 65;
@(posedge sample_clk) data_input <= 66;
@(posedge sample_clk) data_input <= 67;
@(posedge sample_clk) data_input <= 68;
@(posedge sample_clk) data_input <= 69;
@(posedge sample_clk) data_input <= 70;
@(posedge sample_clk) data_input <= 71;
@(posedge sample_clk) data_input <= 72;
@(posedge sample_clk) data_input <= 73;
@(posedge sample_clk) data_input <= 74;
@(posedge sample_clk) data_input <= 75;
@(posedge sample_clk) data_input <= 76;
@(posedge sample_clk) data_input <= 77;
@(posedge sample_clk) data_input <= 78;
@(posedge sample_clk) data_input <= 79;
@(posedge sample_clk) data_input <= 80;
@(posedge sample_clk) data_input <= 81;
@(posedge sample_clk) data_input <= 82;
```

```
        @(posedge sample_clk) data_input <= 83;
        @(posedge sample_clk) data_input <= 84;
        @(posedge sample_clk) data_input <= 85;
        @(posedge sample_clk) data_input <= 86;
        @(posedge sample_clk) data_input <= 87;
        @(posedge sample_clk) data_input <= 88;
        @(posedge sample_clk) data_input <= 89;
        @(posedge sample_clk) data_input <= 90;
        @(posedge sample_clk) data_input <= 91;
        @(posedge sample_clk) data_input <= 92;
        @(posedge sample_clk) data_input <= 93;
        @(posedge sample_clk) data_input <= 94;
        @(posedge sample_clk) data_input <= 95;
        @(posedge sample_clk) data_input <= 96;
        @(posedge sample_clk) data_input <= 97;
        @(posedge sample_clk) data_input <= 98;
        @(posedge sample_clk) data_input <= 99;
        @(posedge sample_clk) data_input <= 100;
#10000000000    $finish;
end
deltasigma deltasigma
        (.clk           ( clk               ),
         .reset         ( reset             ),
         .data_input    ( data_input        ),
         .data_output   ( ouptut            ));
```

```
////////////////////////////////////////////////
// Waveform Generation
////////////////////////////////////////////////
initial  $dumpfile("verilog.vcd");
initial  $dumpvars();
endmodule
```

Appendix B HSpice

HSPICE (Hailey Simulation Program with Integrated Circuit Emphasis) is used in industry for accurate circuit simulation and offers foundry-certified MOS device models with state-of-the-art simulation and analysis algorithms.

```
**********************************************************************
* Hspice netlist to calculate the power and delay of the ΔΣ DAC
* Top Cell Name: deltasigma8b
* View Name:     schematic
**********************************************************************
*.BIPOLAR
*.RESI = 2000
*.RESVAL
*.CAPVAL
*.DIOPERI
*.DIOAREA
*.EQUATION
*.SCALE METER
*.MEGA
*.PARAM
*.GLOBAL Gnd
+        VddMod
*.PIN Gnd
*+    VddMod
```

```
********************************************************************************
* Library Name: ricardo_deltasigma1
* View Name:    schematic
********************************************************************************
* Sources Modulator150mv
********************************************************************************
vddM   VddMod   0 dc vddpam
vddls1 VddLS1H  0 dc vddsup
vddls2 VddLS2H  0 dc vddsup

vddbuls1 Vddbufls1 0 dc vddsup
vddbuls2 Vddbufls2 0 dc vddsup

vdd4   VddLS1L  0 dc vddpam
vdd5   VddLS2L  0 dc vddpam
v0  Gnd 0 dc 0

vclk clk 0 PULSE(0 150m 166.667u 16.666u 16.666u 166.667u 333.333u)
v1 data_input<0> 0 PULSE(0 150m 721.81m 16.666u 16.666u 720m 3.6)
v2 data_input<1> 0 PULSE(0 150m 721.81m 16.666u 16.666u 720m 3.6)
v3 data_input<2> 0 PULSE(0 150m 721.81m 16.666u 16.666u 720m 3.6)
v4 data_input<3> 0 PULSE(0 150m 721.81m 16.666u 16.666u 720m 3.6)
v5 data_input<4> 0 PULSE(0 150m 721.81m 16.666u 16.666u 720m 3.6)
v6 data_input<5> 0 PULSE(0 150m 721.81m 16.666u 16.666u 720m 3.6)
v7 data_input<6> 0 PULSE(0 150m 721.81m 16.666u 16.666u 720m 3.6)
v8 data_input<7> 0 PULSE(0 150m 1.81m 16.666u 16.666u 2.16 4.32)
vreset reset 0 PWL(0 150m 666.666u 150m 666.67u 0)
```

```
********************************************************************************
* Library Name: ricardo_deltasigma1
* View Name:    schematic
********************************************************************************
*.PININFO
CC2 data_outputrc Gnd 50f
XI5 net073 data_input<0> net094 data_input<1> net092 data_input<2> net090
+ data_input<3> net068 data_input<4> net047 data_input<5> net081 data_input<6>
+ net062 data_input<7> net050 clk net086 reset VddMod Gnd  buffers8b
XI0 clk data_input<7> data_input<6> data_input<5> data_input<4> data_input<3>
+ data_input<2> data_input<1> data_input<0> data_outputrc reset VddMod Gnd
+ deltasigma8b_good
********************************************************************************
*OPTIONS dsdac_V150
********************************************************************************
.option post=1
.op all
.TRAN  12.6m tm uic

.param vddpam=150m
.param vddsup=2.5
.param tm=3.024

.nodeset v(n_4)=0
.nodeset v(n_8)=0
.nodeset v(n_12)=0
.nodeset v(n_15)=0
```

```
.nodeset v(n_19)=0
.nodeset v(n_23)=0
.nodeset v(n_26)=0
.nodeset v(n_29)=0
.nodeset v(n_32)=vddpam
.nodeset v(n_33)=0
.nodeset v(n_2)=0
*.nodeset v(L_out)=0
*.nodeset v(out)=0
*.IC v(l_out)=0
.IC v(data_outputrc)=0

.print TRAN v(data_input<7>) v(data_input<0>) v(data_outputrc) v(reset) v(clk) i(VddMod)
.plot TRAN v(data_input<7>) v(data_input<0>) v(data_outputrc)  v(reset) v(clk) i(VddMod)
.print P(VddMod)

.measure TRAN iiddM    integral  i(vddM)    FROM=0 TO=tm
.measure TRAN iiddLS1  integral  i(vddls1)  FROM=0 TO=tm
.measure TRAN iiddLS2  integral  i(vddls2)  FROM=0 TO=tm
.measure TRAN iiddbufls1  integral  i(vddbuls1)  FROM=0 TO=tm
.measure TRAN iiddbufls2  integral  i(vddbuls2)  FROM=0 TO=tm
.measure TRAN iiddbufls1low  integral  i(vdd4)  FROM=0 TO=tm
.measure TRAN iiddbufls2low  integral  i(vdd5)  FROM=0 TO=tm

.measure TRAN energyM   param='iiddM*vddpam'
.measure TRAN energyLS1 param='iiddLS1*vddsup'
.measure TRAN energyLS2 param='iiddLS2*vddsup'
```

```
.measure TRAN energybufls1 param='iiddbufls1*vddsup'
.measure TRAN energybufls2 param='iiddbufls2*vddsup'
.measure TRAN energybufls1low param='iiddbufls1low*vddsup'
.measure TRAN energybufls2low param='iiddbufls2low*vddsup'

.measure TRAN PWRM param='energyM/tm'
.measure TRAN PWRLS1 param='energyLS1/tm'
.measure TRAN PWRLS2 param='energyLS2/tm'
.measure TRAN PWRBUFLS1 param='energybufls1/tm'
.measure TRAN PWRBUFLS2 param='energybufls2/tm'
.measure TRAN PWRBUFLS1low param='energybufls1low/tm'
.measure TRAN PWRBUFLS2low param='energybufls2low/tm'
********************************************************************************
.Include tsmc25.mod
********************************************************************************
* Library Name: ricardo_deltasigma1
* Cell Name:    inv_1
* View Name:    schematic
********************************************************************************
.SUBCKT inv_1 ip op VddMod Gnd
*.PININFO ip:I op:O
MP0 op ip VddMod VddMod PM W=3.36u L=240.00n m=1
MN0 op ip Gnd Gnd NM W=1.68u L=240.00n m=1
.ENDS
```

```
********************************************************************************
* Library Name: ricardo_deltasigma1
* Cell Name:    buffers8b
* View Name:    schematic
********************************************************************************
.SUBCKT buffers8b b0 b0o b1 b1o b2 b2o b3 b3o b4 b4o b5 b5o b6 b6o b7 b7o clk
+ clko reset reseto VddMod Gnd
*.PININFO b0:I b1:I b2:I b3:I b4:I b5:I b6:I b7:I clk:I reset:I b0o:O b1o:O
*.PININFO b2o:O b3o:O b4o:O b5o:O b6o:O b7o:O clko:O reseto:O
XI19 net25 clko VddMod Gnd  inv_1
XI18 net27 reseto VddMod Gnd  inv_1
XI17 clk net25 VddMod Gnd  inv_1
XI16 reset net27 VddMod Gnd  inv_1
XI15 net37 b4o VddMod Gnd  inv_1
XI14 net39 b5o VddMod Gnd  inv_1
XI13 net41 b6o VddMod Gnd  inv_1
XI12 net43 b7o VddMod Gnd  inv_1
XI11 b4 net37 VddMod Gnd  inv_1
XI10 b5 net39 VddMod Gnd  inv_1
XI9 b6 net41 VddMod Gnd  inv_1
XI8 b7 net43 VddMod Gnd  inv_1
XI7 net47 b3o VddMod Gnd  inv_1
XI6 b3 net47 VddMod Gnd  inv_1
XI5 b2 net49 VddMod Gnd  inv_1
XI4 net49 b2o VddMod Gnd  inv_1
XI3 net55 b1o VddMod Gnd  inv_1
XI2 b1 net55 VddMod Gnd  inv_1
```

```
XI26 net59 b0o VddMod Gnd  inv_1
XI25 b0 net59 VddMod Gnd  inv_1
.ENDS
********************************************************************************
* Library Name: ricardo_deltasigma1
* Cell Name:    inv_4
* View Name:    schematic
********************************************************************************
.SUBCKT inv_4 ip op VddMod Gnd
*.PININFO ip:I op:O
MP0 op ip VddMod VddMod PM W=3.36u L=240.00n m=1
MP1 VddMod ip op VddMod PM W=3.36u L=240.00n m=1
MN1 Gnd ip op Gnd NM W=1.68u L=240.00n m=1
MN0 op ip Gnd Gnd NM W=1.68u L=240.00n m=1
.ENDS
********************************************************************************
* Library Name: ricardo_deltasigma1
* Cell Name:    dp_1
* View Name:    schematic
********************************************************************************
.SUBCKT dp_1 ck ip q VddMod Gnd
*.PININFO ck:I ip:I q:O
MN0 net44 ip Gnd Gnd NM W=840.0n L=240.0n m=1
MN1 net38 ck Gnd Gnd NM W=840.0n L=240.0n m=1
MN2 net35 net38 net44 Gnd NM W=840.0n L=240.0n m=1
MN3 net32 ck net35 Gnd NM W=840.0n L=240.0n m=1
MN4 net32 net78 Gnd Gnd NM W=840.0n L=240.0n m=1
```

```
MN5 net78 net35 Gnd Gnd NM W=840.0n L=240.0n m=1
MN6 net20 net78 Gnd Gnd NM W=840.0n L=240.0n m=1
MN7 net63 ck net20 Gnd NM W=840.0n L=240.0n m=1
MN8 net12 net38 net63 Gnd NM W=840.0n L=240.0n m=1
MN9 Gnd q net12 Gnd NM W=840.0n L=240.0n m=1
MN10 q net63 Gnd Gnd NM W=840.0n L=240.0n m=1
MP0 net88 ip VddMod VddMod PM W=1.68u L=240.0n m=1
MP1 net38 ck VddMod VddMod PM W=1.68u L=240.0n m=1
MP2 net35 ck net88 VddMod PM W=1.68u L=240.0n m=1
MP3 net80 net38 net35 VddMod PM W=1.68u L=240.0n m=1
MP4 VddMod net78 net80 VddMod PM W=1.68u L=240.0n m=1
MP5 net78 net35 VddMod VddMod PM W=1.68u L=240.0n m=1
MP6 net68 net78 VddMod VddMod PM W=1.68u L=240.0n m=1
MP7 net63 net38 net68 VddMod PM W=1.68u L=240.0n m=1
MP8 net60 ck net63 VddMod PM W=1.68u L=240.0n m=1
MP9 VddMod q net60 VddMod PM W=1.68u L=240.0n m=1
MP10 q net63 VddMod VddMod PM W=1.68u L=240.0n m=1
.ENDS
********************************************************************************
* Library Name: ricardo_deltasigma1
* Cell Name:    and2_2
* View Name:    schematic
********************************************************************************
.SUBCKT and2_2 ip1 ip2 op VddMod Gnd
*.PININFO ip1:I ip2:I op:O
MN2 op net32 Gnd Gnd NM W=1.68u L=240.0n m=1
MN1 net32 ip1 net27 Gnd NM W=1.68u L=240.0n m=1
```

```
MN0 net27 ip2 Gnd Gnd NM W=1.68u L=240.0n m=1
MP1 net32 ip1 VddMod VddMod PM W=3.36u L=240.0n m=1
MP2 op net32 VddMod VddMod PM W=3.36u L=240.0n m=1
MP0 net32 ip2 VddMod VddMod PM W=3.36u L=240.0n m=1
.ENDS
********************************************************************************
* Library Name: ricardo_deltasigma1
* Cell Name:    nor2_2
* View Name:    schematic
********************************************************************************
.SUBCKT nor2_2 ip1 ip2 op VddMod Gnd
*.PININFO ip1:I ip2:I op:O
MP0 net35 ip1 VddMod VddMod PM W=3.36u L=240.0n m=1
MP1 op ip2 net35 VddMod PM W=3.36u L=240.0n m=1
MN0 op ip2 Gnd Gnd NM W=1.68u L=240.0n m=1
MN1 op ip1 Gnd Gnd NM W=1.68u L=240.0n m=1
.ENDS
********************************************************************************
* Library Name: ricardo_deltasigma1
* Cell Name:    xnor2_2
* View Name:    schematic
********************************************************************************
.SUBCKT xnor2_2 ip1 ip2 op VddMod Gnd
*.PININFO ip1:I ip2:I op:O
MP0 VddMod ip1 net096 VddMod PM W=1.68u L=240.0n m=1
MP18 VddMod ip1 net069 VddMod PM W=1.68u L=240.0n m=1
MP14 op net072 VddMod VddMod PM W=3.36u L=240.0n m=1
```

```
MP17 net071 ip2 VddMod VddMod PM W=1.68u L=240.0n m=1
MP16 net072 ip2 net084 VddMod PM W=1.68u L=240.0n m=1
MP19 net069 net071 net072 VddMod PM W=1.68u L=240.0n m=1
MP15 net084 net096 VddMod VddMod PM W=1.68u L=240.0n m=1
MN19 Gnd net096 net0108 Gnd NM W=840.0n L=240.0n m=1
MN13 op net072 Gnd Gnd NM W=1.68u L=240.0n m=1
MN18 net0108 net071 net072 Gnd NM W=840.0n L=240.0n m=1
MN20 net0116 ip1 Gnd Gnd NM W=840.0n L=240.0n m=1
MN17 net072 ip2 net0116 Gnd NM W=840.0n L=240.0n m=1
MN16 net071 ip2 Gnd Gnd NM W=840.0n L=240.0n m=1
MN21 Gnd ip1 net096 Gnd NM W=840.0n L=240.0n m=1
.ENDS
********************************************************************************
* Library Name: ricardo_deltasigma1
* Cell Name:    and2_1
* View Name:    schemati
********************************************************************************
.SUBCKT and2_1 ip1 ip2 op VddMod Gnd
*.PININFO ip1:I ip2:I op:O
MP0 net12 ip2 VddMod VddMod PM W=1.68u L=240.0n m=1
MP2 op net12 VddMod VddMod PM W=1.68u L=240.0n m=1
MP1 net12 ip1 VddMod VddMod PM W=1.68u L=240.0n m=1
MN0 net063 ip2 Gnd Gnd NM W=840.0n L=240.0n m=1
MN1 net12 ip1 net063 Gnd NM W=840.0n L=240.0n m=1
MN2 op net12 Gnd Gnd NM W=840.0n L=240.0n m=1
.ENDS
********************************************************************************
```

```
* Library Name: ricardo_deltasigma1
* Cell Name:    fulladder
* View Name:    schematic
********************************************************************************
.SUBCKT fulladder a b ci co s VddMod Gnd
*.PININFO a:I b:I ci:I co:O s:O
MP5 co net76 VddMod VddMod PM W=1.68u L=240.0n m=1
MP2 net76 b net6 VddMod PM W=1.68u L=240.0n m=1
MP3 net14 ci net76 VddMod PM W=1.68u L=240.0n m=1
MP4 s net90 VddMod VddMod PM W=1.68u L=240.0n m=1
MP6 net50 net76 net90 VddMod PM W=1.68u L=240.0n m=1
MP7 net90 ci net42 VddMod PM W=1.68u L=240.0n m=1
MP8 net50 a VddMod VddMod PM W=1.68u L=240.0n m=1
MP9 VddMod b net50 VddMod PM W=1.68u L=240.0n m=1
MP10 net50 ci VddMod VddMod PM W=1.68u L=240.0n m=1
MP11 net46 a net50 VddMod PM W=1.68u L=240.0n m=1
MP12 net42 b net46 VddMod PM W=1.68u L=240.0n m=1
MP15 net14 a VddMod VddMod PM W=1.68u L=240.0n m=1
MP16 VddMod b net14 VddMod PM W=1.68u L=240.0n m=1
MP17 net6 a net14 VddMod PM W=1.68u L=240.0n m=1
MN0 s net90 Gnd Gnd NM W=840.0n L=240.0n m=1
MN5 net90 ci net98 Gnd NM W=840.0n L=240.0n m=1
MN4 co net76 Gnd Gnd NM W=840.0n L=240.0n m=1
MN2 net73 ci net76 Gnd NM W=840.0n L=240.0n m=1
MN3 net76 b net66 Gnd NM W=840.0n L=240.0n m=1
MN6 net106 net76 net90 Gnd NM W=840.0n L=240.0n m=1
MN7 net101 a Gnd Gnd NM W=840.0n L=240.0n m=1
```

```
MN9 Gnd a net106 Gnd NM W=840.0n L=240.0n m=1
MN10 net106 b Gnd Gnd NM W=840.0n L=240.0n m=1
MN11 Gnd b net73 Gnd NM W=840.0n L=240.0n m=
MN12 net66 a Gnd Gnd NM W=840.0n L=240.0n m=1
MN13 net73 a Gnd Gnd NM W=840.0n L=240.0n m=1
MN14 Gnd ci net106 Gnd NM W=840.0n L=240.0n m=1
MN15 net98 b net101 Gnd NM W=840.0n L=240.0n m=1
.ENDS
********************************************************************************
* Library Name: ricardo_deltasigma1
* Cell Name:    or2_1
* View Name:    schematic
********************************************************************************
.SUBCKT or2_1 ip1 ip2 op VddMod Gnd
*.PININFO ip1:I ip2:I op:O
MP2 op net46 VddMod VddMod PM W=1.68u L=240.0n m=1
MP1 net29 ip1 VddMod VddMod PM W=1.68u L=240.0n m=1
MP0 net46 ip2 net29 VddMod PM W=1.68u L=240.0n m=1
MN2 op net46 Gnd Gnd NM W=840.0n L=240.0n m=1
MN0 net46 ip1 Gnd Gnd NM W=840.0n L=240.0n m=1
MN1 net46 ip2 Gnd Gnd NM W=840.0n L=240.0n m=1
.ENDS
```

```
********************************************************************************
* Library Name: ricardo_deltasigma1
* Cell Name:    deltasigma8b_good
* View Name:    schematic
********************************************************************************
.SUBCKT deltasigma8b_good clk data_input<7> data_input<6> data_input<5>
+ data_input<4> data_input<3> data_input<2> data_input<1> data_input<0>
+ data_output reset VddMod Gnd
*.PININFO clk:I data_input<7>:I data_input<6>:I data_input<5>:I
*.PININFO data_input<4>:I data_input<3>:I data_input<2>:I data_input<1>:I
*.PININFO data_input<0>:I reset:I data_output:O
Xg1355 reset n_11 VddMod Gnd  inv_4
Xsigma_reg_reg[1] clk n_8 sigma_reg[1] VddMod Gnd  dp_1
Xsigma_reg_reg[2] clk n_12 sigma_reg[2] VddMod Gnd  dp_1
Xsigma_reg_reg[8] clk n_32 sigma_reg[8] VddMod Gnd  dp_1
Xsigma_reg_reg[9] clk n_33 sigma_reg[9] VddMod Gnd  dp_1
Xsigma_reg_reg[3] clk n_15 sigma_reg[3] VddMod Gnd  dp_1
Xsigma_reg_reg[4] clk n_19 sigma_reg[4] VddMod Gnd  dp_1
Xsigma_reg_reg[5] clk n_23 sigma_reg[5] VddMod Gnd  dp_1
Xsigma_reg_reg[6] clk n_26 sigma_reg[6] VddMod Gnd  dp_1
Xsigma_reg_reg[7] clk n_29 sigma_reg[7] VddMod Gnd  dp_1
Xsigma_reg_reg[0] clk n_4 sigma_reg[0] VddMod Gnd  dp_1
Xdata_output_reg clk n_2 data_output VddMod Gnd  dp_1
Xg1352 data_input<0> sigma_reg[0] n_1 VddMod Gnd  and2_2
Xg1351 n_11 sigma_reg[9] n_2 VddMod Gnd  and2_2
Xg1348 reset n_3 n_4 VddMod Gnd  nor2_2
Xg1350 data_input<0> sigma_reg[0] n_3 VddMod Gnd  xnor2_2
```

```
Xg1345 n_11 n_6 n_8 VddMod Gnd  and2_1
Xg1323 n_11 n_30 n_33 VddMod Gnd  and2_1
Xg1342 n_11 n_10 n_12 VddMod Gnd  and2_1
Xg1339 n_11 n_14 n_15 VddMod Gnd  and2_1
Xg1336 n_11 n_17 n_19 VddMod Gnd  and2_1
Xg1333 n_11 n_21 n_23 VddMod Gnd  and2_1
Xg1330 n_11 n_25 n_26 VddMod Gnd  and2_1
Xg1327 n_11 n_28 n_29 VddMod Gnd  and2_1
Xg1343 sigma_reg[2] n_5 data_input<2> n_9 n_10 VddMod Gnd  fulladder
Xg1346 sigma_reg[1] n_1 data_input<1> n_5 n_6 VddMod Gnd  fulladder
Xg1325 sigma_reg[9] n_27 sigma_reg[8] n_30 n_31 VddMod Gnd  fulladder
Xg1337 sigma_reg[4] n_13 data_input<4> n_16 n_17 VddMod Gnd  fulladder
Xg1340 sigma_reg[3] n_9 data_input<3> n_13 n_14 VddMod Gnd  fulladder
Xg1331 sigma_reg[6] n_20 data_input<6> n_24 n_25 VddMod Gnd  fulladder
Xg1334 sigma_reg[5] n_16 data_input<5> n_20 n_21 VddMod Gnd  fulladder
Xg1328 sigma_reg[7] n_24 data_input<7> n_27 n_28 VddMod Gnd  fulladder
Xg1324 n_31 reset n_32 VddMod Gnd  or2_1
.ENDS
********************************************************************************
*LEVEL SHIFTER NOVEL
**********subth inverter*****************************************
MN3     sub1_b_n    data_outputrc   0        0        NM w=0.5u L=240N   m=1
MP3     sub1_b_n    data_outputrc   VddLS2L   VddLS2L   PM w=1.0u L=240N   m=1
***************pull down nfets************************************************
MN4     A2              data_outputrc     0       0       NM w=5u   L=240N  m=1
MN5     LS2_out         sub1_b_n          0       0       NM w=5u   L=240N  m=1
**********************pmos current mirror*************************************
```

```
MP4    A          A          VddLS2H   VddLS2H  PM w=360N L=240N  m=1
MP5    Z          A          VddLS2H   VddLS2H  PM w=360N L=240N  m=1
MP6    A1         A1         A         VddLS2H  PM w=360N L=240N  m=1
MP7    Z1         A1         Z         VddLS2H  PM w=360N L=240N  m=1
MP8    A2         A2         A1        VddLS2H  PM w=360N L=240N  m=1
MP9    LS2_out    A2         Z1        VddLS2H  PM w=360N L=240N  m=1


*********************************************************************************
*BUFFER SHIFTER NOVEL
*********************************Inverter1*********************************
MN310    x1    LS2_out    0        0          NM w=0.5u L=240N   m=1
MP310    x1    LS2_out    Vddbufls2  Vddbufls2    PM w=1.0u L=240N   m=1
MN311    x2    x1         0        0          NM w=0.5u L=240N   m=1
MP311    x2    x1         Vddbufls2  Vddbufls2    PM w=1.0u L=240N   m=1
*********************************Inverter2*********************************
MN312    x3    x2         0        0          NM w=2.0u L=240N   m=1
MP312    x3    x2         Vddbufls2  Vddbufls2    PM w=4.0u L=240N   m=1
MN313    x4    x3         0        0          NM w=2.0u L=240N   m=1
MP313    x4    x3         Vddbufls2  Vddbufls2    PM w=4.0u L=240N   m=1
*********************************Inverter3*********************************
MN314    x5    x4         0        0          NM w=2.0u L=240N   m=4
MP314    x5    x4         Vddbufls2  Vddbufls2    PM w=4.0u L=240N   m=4
MN315    x6    x5         0        0          NM w=2.0u L=240N   m=4
MP315    x6    x5         Vddbufls2  Vddbufls2    PM w=4.0u L=240N   m=4
*********************************Inverter4*********************************
MN316    x7    x6         0        0          NM w=2.0u L=240N   m=16
```

```
MP316    x7    x6        Vddbufls2  Vddbufls2    PM w=4.0u L=240N   m=16
MN317    x8    x7        0        0            NM w=2.0u L=240N   m=16
MP317    x8    x7        Vddbufls2  Vddbufls2    PM w=4.0u L=240N   m=16
**********************************Inverter5**********************************
MN318    x9    x8        0        0            NM w=2.0u L=240N   m=64
MP318    x9    x8        Vddbufls2  Vddbufls2    PM w=4.0u L=240N   m=64
MN319    outbls2  x9      0        0            NM w=2.0u L=240N   m=64
MP319    outbls2  x9      Vddbufls2  Vddbufls2    PM w=4.0u L=240N   m=64
*****************************************************************************
*LEVEL SHIFTER TRADITIONAL
************************subth inverter*************************************
MN31    sub2_b_n    data_outputrc    0        0        NM w=0.5u L=240N   m=1
MP31    sub2_b_n    data_outputrc    VddLS1L    VddLS1L   PM w=1.0u L=240N   m=1
***************pull down nfets*************************************************
MN41    Ab            data_outputrc    0       0      NM w=5u   L=240N  m=1
MN51    LS1_out         sub2_b_n        0       0      NM w=5u   L=240N  m=1
***********************pmos current mirror***************************************
MP41    Ab          Ab          VddLS1H   VddLS1H  PM w=360N L=240N  m=1
MP51    LS1_out       Ab          VddLS1H   VddLS1H  PM w=360N L=240N  m=1
*****************************************************************************
*BUFFER SHIFTER traditional
**********************************Inverter1**********************************
MN320    xy1    LS1_out     0        0            NM w=0.5u L=240N   m=1
MP320    xy1    LS1_out     Vddbufls1  Vddbufls1    PM w=1.0u L=240N   m=1
MN321    xy2    xy1         0        0            NM w=0.5u L=240N   m=1
MP321    xy2    xy1         Vddbufls1  Vddbufls1    PM w=1.0u L=240N   m=1
**********************************Inverter2**********************************
```

```
MN322    xy3    xy2       0       0          NM w=2.0u L=240N   m=1
MP322    xy3    xy2       Vddbufls1  Vddbufls1    PM w=4.0u L=240N   m=1
MN323    xy4    xy3       0       0          NM w=2.0u L=240N   m=1
MP323    xy4    xy3       Vddbufls1  Vddbufls1    PM w=4.0u L=240N   m=1
**********************************Inverter3**********************************
MN324    xy5    xy4       0       0          NM w=2.0u L=240N   m=4
MP324    xy5    xy4       Vddbufls1  Vddbufls1    PM w=4.0u L=240N   m=4
MN325    xy6    xy5       0       0          NM w=2.0u L=240N   m=4
MP325    xy6    xy5       Vddbufls1  Vddbufls1    PM w=4.0u L=240N   m=4
**********************************Inverter4**********************************
MN326    xy7    xy6       0       0          NM w=2.0u L=240N   m=16
MP326    xy7    xy6       Vddbufls1  Vddbufls1    PM w=4.0u L=240N   m=16
MN327    xy8    xy7       0       0          NM w=2.0u L=240N   m=16
MP327    xy8    xy7       Vddbufls1  Vddbufls1    PM w=4.0u L=240N   m=16
**********************************Inverter5**********************************
MN328    xy9    xy8       0       0          NM w=2.0u L=240N   m=64
MP328    xy9    xy8       Vddbufls1  Vddbufls1    PM w=4.0u L=240N   m=64
MN329    outbls1  xy9       0       0          NM w=2.0u L=240N   m=64
MP329    outbls1  xy9       Vddbufls1  Vddbufls1    PM w=4.0u L=240N   m=64
*****************************************************************************

.END

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
**********************************************************************
*Voltage levels, input stimulus for analysis at 250mv
*SOURCES 250mv
**********************************************************************
v0  Gnd 0 dc 0
vclk clk 0 PULSE(0 250m 31.25u 3.125u 3.125u 31.25u 62.5u)
v1 data_input<0> 0 PULSE(0 250m 134.271m 3.125u 3.125u 134m 670m)
v2 data_input<1> 0 PULSE(0 250m 134.271m 3.125u 3.125u 134m 670m)
v3 data_input<2> 0 PULSE(0 250m 134.271m 3.125u 3.125u 134m 670m)
v4 data_input<3> 0 PULSE(0 250m 134.271m 3.125u 3.125u 134m 670m)
v5 data_input<4> 0 PULSE(0 250m 134.271m 3.125u 3.125u 134m 670m)
v6 data_input<5> 0 PULSE(0 250m 134.271m 3.125u 3.125u 134m 670m)
v7 data_input<6> 0 PULSE(0 250m 134.271m 3.125u 3.125u 134m 670m)
v8 data_input<7> 0 PULSE(0 250m 271u 3.125u 3.125u 402m 804m)
vreset reset 0 PWL(0 250m 124.9u 250m 150u 0)
**********************************************************************
*time parameters for analysis at 250mv
*OPTIONS 250mv
**********************************************************************
.option post=1
.op all
.TRAN  2.35m tm uic
.param vddpam=250m
.param vddsup=2.5
.param tm=562.8m
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
**********************************************************************
*Voltage levels, input stimulus for analysis at 350mv
*SOURCES 350mv
**********************************************************************
v0  Gnd 0 dc 0
vclk clk 0 PULSE(0 350m 5u .5u .5u 5u 10u)
v1 data_input<0> 0 PULSE(0 350m 22.044m .5u .5u 22m 110m)
v2 data_input<1> 0 PULSE(0 350m 22.044m .5u .5u 22m 110m)
v3 data_input<2> 0 PULSE(0 350m 22.044m .5u .5u 22m 110m)
v4 data_input<3> 0 PULSE(0 350m 22.044m .5u .5u 22m 110m)
v5 data_input<4> 0 PULSE(0 350m 22.044m .5u .5u 22m 110m)
v6 data_input<5> 0 PULSE(0 350m 22.044m .5u .5u 22m 110m)
v7 data_input<6> 0 PULSE(0 350m 22.044m .5u .5u 22m 110m)
v8 data_input<7> 0 PULSE(0 350m 44u .5u .5u 66m 132m)
vreset reset 0 PWL(0 350m 19.9u 150m 20u 0)
**********************************************************************
*time parameters for analysis at 350mv
*OPTIONS 350mv
**********************************************************************
.option post=1
.op all
.TRAN  385u tm uic
.param vddpam=350m
.param vddsup=2.5
.param tm=92.4m
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
**********************************************************************
*Voltage levels, input stimulus for analysis at 350mv
*SOURCES 500mv
**********************************************************************
v0  Gnd 0 dc 0
vclk clk 0 PULSE(0 500m 500n 50n 50n 500n 1u)
v1 data_input<0> 0 PULSE(0 500m 2.2044m 50n 50n 2.2m 11m)
v2 data_input<1> 0 PULSE(0 500m 2.2044m 50n 50n 2.2m 11m)
v3 data_input<2> 0 PULSE(0 500m 2.2044m 50n 50n 2.2m 11m)
v4 data_input<3> 0 PULSE(0 500m 2.2044m 50n 50n 2.2m 11m)
v5 data_input<4> 0 PULSE(0 500m 2.2044m 50n 50n 2.2m 11m)
v6 data_input<5> 0 PULSE(0 500m 2.2044m 50n 50n 2.2m 11m)
v7 data_input<6> 0 PULSE(0 500m 2.2044m 50n 50n 2.2m 11m)
v8 data_input<7> 0 PULSE(0 500m 4.4u 50n 50n 6.6m 13.2m)
vreset reset 0 PWL(0 500m 2.099u 500m 2.1u 0)
**********************************************************************
*time parameters for analysis at 500mv
*OPTIONS 500mv
**********************************************************************
.option post=1
.op all
.TRAN  1n tm uic
.param vddpam=500m
.param vddsup=2.5
.param tm=9.24m
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
**********************************************************************
```

```
*Voltage levels, input stimulus for analysis at 350mv
*SOURCES 700mv
*********************************************************************************
v0  Gnd 0 dc 0
vclk clk 0 PULSE(0 700m 31.25n 3.125n 3.125n 31.25n 62.5n)
v1 data_input<0> 0 PULSE(0 700m 140.274u 3.125n 3.125n 140u 700u)
v2 data_input<1> 0 PULSE(0 700m 140.274u 3.125n 3.125n 140u 700u)
v3 data_input<2> 0 PULSE(0 700m 140.274u 3.125n 3.125n 140u 700u)
v4 data_input<3> 0 PULSE(0 700m 140.274u 3.125n 3.125n 140u 700u)
v5 data_input<4> 0 PULSE(0 700m 140.274u 3.125n 3.125n 140u 700u)
v6 data_input<5> 0 PULSE(0 700m 140.274u 3.125n 3.125n 140u 700u)
v7 data_input<6> 0 PULSE(0 700m 140.274u 3.125n 3.125n 140u 700u)
v8 data_input<7> 0 PULSE(0 700m 274n 3.125n 3.125n 420u 840u)
vreset reset 0 PWL(0 700m 126.49n 700m 126.5n 0)
*********************************************************************************
*time parameters for analysis at 700mv
*OPTIONS 700mv
*********************************************************************************
.option post=1
.op all
.TRAN 50u tm uic
.param vddpam=700m
.param vddsup=2.5
.param tm=588u
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
**************************************************************************
*Voltage levels, input stimulus for analysis at 350mv
*SOURCES 1200mv
**************************************************************************
v0  Gnd 0 dc 0
vclk clk 0 PULSE(0 1.2 4.6296n 462.9p 462.9p 4.6296n 9.25925n)
v1 data_input<0> 0 PULSE(0 1.2 20.0041u 462.9p 462.9p 20u 100u)
v2 data_input<1> 0 PULSE(0 1.2 20.0041u 462.9p 462.9p 20u 100u)
v3 data_input<2> 0 PULSE(0 1.2 20.0041u 462.9p 462.9p 20u 100u)
v4 data_input<3> 0 PULSE(0 1.2 20.0041u 462.9p 462.9p 20u 100u)
v5 data_input<4> 0 PULSE(0 1.2 20.0041u 462.9p 462.9p 20u 100u)
v6 data_input<5> 0 PULSE(0 1.2 20.0041u 462.9p 462.9p 20u 100u)
v7 data_input<6> 0 PULSE(0 1.2 20.0041u 462.9p 462.9p 20u 100u)
v8 data_input<7> 0 PULSE(0 1.2 41n 462.9p 462.9p 60u 120u)
vreset reset 0 PWL(0 1.2 18.519n 1.2 18.52n 0)
**************************************************************************
*time parameters for analysis at 1200mv
*OPTIONS 1200mv
**************************************************************************
.option post=1
.op all
.TRAN  .1u tm uic
.param vddpam=1.2
.param vddsup=2.5
.param tm=65u
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
**************************************************************************
*Voltage levels, input stimulus for analysis at 350mv
*SOURCES 2500mv
**************************************************************************
v0  Gnd 0 dc 0
vclk clk 0 PULSE(0 2.5 1.70648n 170.64p 170.64p 1.70648n 3.412969n)
v1 data_input<0> 0 PULSE(0 2.5 7.3152u 170.64p 170.64p 7.3u 36.5u)
v2 data_input<1> 0 PULSE(0 2.5 7.3152u 170.64p 170.64p 7.3u 36.5u)
v3 data_input<2> 0 PULSE(0 2.5 7.3152u 170.64p 170.64p 7.3u 36.5u)
v4 data_input<3> 0 PULSE(0 2.5 7.3152u 170.64p 170.64p 7.3u 36.5u)
v5 data_input<4> 0 PULSE(0 2.5 7.3152u 170.64p 170.64p 7.3u 36.5u)
v6 data_input<5> 0 PULSE(0 2.5 7.3152u 170.64p 170.64p 7.3u 36.5u)
v7 data_input<6> 0 PULSE(0 2.5 7.3152u 170.64p 170.64p 7.3u 36.5u)
v8 data_input<7> 0 PULSE(0 2.5 15.2n 170.64p 170.64p 21.9u 43.8u)
vreset reset 0 PWL(0 2.5 7.69n 2.5 7.70n 0)
**************************************************************************
*time parameters for analysis at 2500mv
*OPTIONS 2500mv
**************************************************************************
.option post=1
.op all
.TRAN  .129u tm uic
.param vddpam=2.5
.param vddsup=2.5
.param tm=31u
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

Appendix C Ocean

Open Command Environment for Analysis (OCEAN) is a text based process that can be run from a UNIX shell or form Virtuoso's Command Interpreter Window (CIW). OCEAN uses Cadence language to configure the design environment in order to create scripts to automate circuit verification also be useful to run simulations from a non-graphic, remote terminal.

```
//////////////////////////////////////////OCEAN Script to Plot the Output Spectrum//////////////////////////////////////
simulator( 'spectre )
design(
"/users/eesunz/faculty/cdsemac/cadence/simulation/final_dm_sim/spectre/schematic/netlist/netlist")
resultsDir( "/users/eesunz/faculty/cdsemac/cadence/simulation/final_dm_sim/spectre/schematic" )
modelFile(
   '("/export/cadence/linux/tools.lnx86/dfII/local/ncsu-cdk-
1.6.0.beta/models/hspice/public/tsmc25dN.m" "")
   '("/export/cadence/linux/tools.lnx86/dfII/local/ncsu-cdk-1.6.0.beta/models/hspice/public/tsmc25dP.m"
"")
)
analysis('tran ?stop "120m"  ?errpreset "liberal"  ?method "gear2"
                  ?relref "sigglobal"  ?compression "yes"  ?strobeperiod "5u"  )
converge( 'ic "/ls1out" "0" )
converge( 'ic "/ls2out" "0" )
temp( 27 )
run()
selectResult( 'tran )
plot(getData("/outls2filter") getData("/outls1filter") getData("/ls2out") getData("/ls1out")
getData("/reset") getData("/clk") getData("/data_input<7>") getData("/data_input<6>")
```

getData("/data_input<5>") getData("/data_input<4>") getData("/data_input<3>") getData("/data_input<2>") getData("/data_input<1>") getData("/data_input<0>"))

<OPTIONAL: If Appendix not used, delete the entire page including the following page break>

Vita

Ricardo Baca Baylon was born on June 6, 1980 in Chihuahua, Mexico. The second born son of Ricardo Baca Villanueva and Bibiana Baylon Montes, he completed his schooling from Colegio de Bachilleres Numero Tres, Chihuahua. He then entered Instituto Tecnologico de Chihuahua, Chihuahua, Mexico to pursue Bachelor degree in Electronics Engineering. He graduated with a Bachelor's degree in October 2003. Ricardo worked in Delphi Mexico Technical Center for five years as Design Engineer during this period he joined The University of Texas at El Paso, United States as a graduate student to pursue his Master of Science in Electrical Engineering. His area of research has been VLSI with focus on - Digital Circuit Design.

Permanent address: 419 Eulalio Gutierrez Street
Chihuahua, Chihuahua, 31230
Mexico.

This thesis was typed by the author.