NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF MATERIALS SCIENCE

CAROLINA SIMÃO VIEGAS

BSc in Micro and Nanotechnologies Engineering

LOW NOISE TIME TO DIGITAL CONVERTERS AS PHASE DETECTORS FOR ALL DIGITAL PLLS

MASTER IN MICRO AND NANOTECHNOLOGIES ENGINEERING

NOVA University Lisbon
September, 2022

NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF MATERIALS SCIENCE

LOW NOISE TIME TO DIGITAL CONVERTERS AS PHASE DETECTORS FOR ALL DIGITAL PLLS

CAROLINA SIMÃO VIEGAS

BSc in Micro and Nanotechnologies Engineering

Adviser: Luís Augusto Bica Gomes de Oliveira Ph.D.
Associate Professor, NOVA University Lisbon

Co-adviser: Michael Figueiredo Ph.D.
Specialist Engineer, AuraSemi

Examination Committee:

Chair: Pedro Miguel Cândido Barquinha Ph.D.
Associate Professor, NOVA University Lisbon

Rapporteurs: João Carlos Ferreira de Almeida Casaleiro Ph.D.
Assistant Professor, ISEL

Luís Augusto Bica Gomes de Oliveira Ph.D.
Associate Professor, NOVA University Lisbon

MASTER IN MICRO AND NANOTECHNOLOGIES ENGINEERING

NOVA University Lisbon
September, 2022

Low Noise Time to Digital Converters as Phase Detectors for All Digital PLLs

Acknowledgements

Finishing this chapter of my journey I would like to express my gratitude to all the people who supported me through this process. I would like to thank my family, my friends and my advisor for their guidance and patience.

More importantly I would like to thank my parents and my sister for their continuous support, for their patience, their guidance and encouragement, without which I couldn't have done it.

"The best way to predict the future is to create it. "

Peter Drucker

Abstract

Nowadays PLLs are used in in almost every electronic circuit, because phase correction and detection are very important in a circuit. For this phase detection TDCs are used.

This work proposes and demonstrates a Low noise Time to Digital Converter (TDC). This Time to Digital converter will be used as a phase detector in an all Digital PLL, with a 100 MHz frequency.

The proposed topology employs CMOS inverters, and Set and Reset Flip Flops, due to their simplicity, to achieve a 4 bit circuit. The performance of the circuit was studied by evaluation fundamental parameters like RMS jitter, linearity, resolution and range. To further test the circuit a mismatch and noise analysis was performed, by testing the circuit with the PVT corners and Monte Carlo variations.

The proposed TDC is simulated, using UMC 130 nm CMOS technology, achieves a RMS jitter of 22.9 fs, a INL and DNL error of 0.13 and 0.11 LSB respectively and a resolution of 15.3 ps. The TDC also has a power consumption of 1.11 mW and a area of 0.143 mm^2.

Keywords: Time-to-Digital Converter (TDC), Phase Locked Loop (PLL), All Digital PLL (ADPLL), Phase detector, Inverter, Delay line, Set and Reset Flip-Flop

Resumo

Atualmente as PLLs são utilizadas em quase todos os circuitos eletrónicos, porque a correção e a deteção de fase são muito importantes num circuito. Para esta deteção de fase são utilizados CTDs.

Este trabalho propõe e demonstra um conversor de tempo para digital (CTD) de baixo ruído. Este conversor de tempo para digital será utilizado como detetor de fase num PLL completamente Digital, com frequência de 100 MHz.

A topologia proposta emprega inversores CMOS e Flip Flops Set e Reset, devido à sua simplicidade, para obter um circuito de 4 bits. O desempenho do circuito foi estudado pela avaliação de parâmetros fundamentais como jitter RMS, linearidade, resolução e alcance. Para testar ainda mais o circuito foi realizada uma análise de incompatibilidade e ruído, testando o circuito com os cantos PVT e variações de Monte Carlo.

O CTD proposto é simulado, usando tecnologia UMC 130 nm CMOS, atinge um jitter RMS de 22,9 fs, um erro INL e DNL de 0,13 e 0,11 LSB respetivamente e uma resolução de 15,3 ps. O CTD tem também um consumo de energia de 1,11 mW e uma área de 0.143 mm^2.

Palavras-chave: Conversor de Tempo para Digital, Phase Locked Loop, PLL Completamente Digital, Detetor de Fase, Inversor, Linha de atraso, Flip-Flop Set e Reset

Contents

List of Figures

List of Tables

List of Listings

Acronyms

ADC Analog-to-Digital Converter 7, 8, 19
ADE Analog Design Environment 53
ADPLL All Digital Phase Locked Loop 4, 7, 35

CMOS Complementary metal–oxide–semiconductor 2, 6, 9, 12, 13, 25, 35, 53

DCO Digitally Controlled Oscillator 4
DNL Differential Non Linearity 29, 30, 32, 34, 35, 48, 49

ENOB Effective number of bits 30, 34, 35, 49

FOM Figure-of-Merit 30, 34, 49

HEP high-energy physics 1

INL Integral Non Linearity 29, 30, 32, 34, 35, 48, 49

LDO Low-dropout regulator 21, 24, 32, 36
LSB Least Significant Bit 29, 32, 34, 35

NMOS Negative Channel Metal Oxide Semiconductor 6, 9, 10, 14, 27

op-amp Operational Amplifier 8

PLL Phase Locked Loop 1, 2, 3, 4, 14
PMOS Positive Channel Metal Oxide Semiconductor 6, 9, 10, 14, 27, 28
PNOISE Periodic Noise 27, 50, 51, 52
PSS Periodic Steady-State 27, 50, 52

PVT Process Voltage and Temperature 48

SNR Signal-to-Noise Ratio 7
SRFF Set/Reset Flip-Flip 12, 13, 14, 21, 23, 25, 27, 28, 35

TDAQ trigger and data acquisition systems 1
TDC Time to Digital Converter 2, 4, 7, 8, 9, 12, 14, 15, 16, 17, 18, 19, 21, 25, 27, 28, 30, 32, 34, 35, 36, 48, 49

VCO Voltage Controlled Oscillator 3, 4

Symbols

C_L	Load Capacitance 12
CLK	Clock input signal 31
F_{max}	Maximum switching frequency 11, 25
f_s	Sampling Frequency 30, 50
I_D	Saturation current 6
Tp_{min}	Minimum Period 11, 25
Tp_{HL}	High to Low propagation delay 11
Tp_{LH}	Low to High propagation delay 11
V_{DD}	Circuit supply voltage 2, 10, 11, 13, 14, 21, 27, 31
V_{in}	Input signal 10, 11, 12, 28, 31, 32
V_{out}	Output signal 10, 11, 12
V_{th}	Threshold voltage 10, 13

1

Introduction

1.1 Context and Motivation

The Phase Locked Loop (PLL) is a basic building block in electronic circuits. It is constituted by a blend of analog and digital techniques, can operate with both linear and nonlinear behavior, and can function with continuous, sampled or chaotic analysis. PLL circuits maintain the phase of the local oscillator to the phase of an external signal. They were created in 1930 to implement a zero IF frequency synchronous receiver. [1]

Digital building blocks take advantage of the low device geometries in terms of area, power per functionality, and switching speed. Analog circuits on the other hand rely on the actual shape of the transistor characteristic. The technology scaling degrades these characteristics in relation to analog performance parameters. Under 100 nm technology the design of analog and mixed-signal circuits becomes more difficult. This applies more for low supply voltages equal or bellow 1 V. This increases the design effort and the power consumption. In digital circuits scaling makes the circuits faster, smaller and use less power. The increase in the switching speed decreases the circuits susceptibility to noise, like the flicker noise in the transistors. The main advantage of digital signal processing is the robustness of digital signals against disturbances and process variation. Unfortunately there are some disadvantages, like the generation of power supply or the lack of power supply rejection. Even considering this, the advantages prevail and suggest implementing as much components in the digital domain as possible. The solution however is not switching every part of the circuit to digital, because some functions are better done with analog circuits. [2]

Digital circuits (binary in specific) have more resolution on the time domain than on the amplitude domain. This resolution in the time domain is very important do acquire accurate time measurements in trigger and data acquisition systems (TDAQ) of high-energy physics (HEP) experiments. Time measuring devices are used in HEP experiments because they can be used in most parts of the experiment, for example in the synchronization and for time measurement purposes. [3]

1.2 Objectives and Results

The goal of this work is to study and design a Time to Digital Converter (TDC) for phase detection. The study focuses in analysing all components of the chosen TDC. The TDC design and simulation have to comply with the parameters in Table 1.1, using UMC L130 Mixed-Mode/RF Complementary metal–oxide–semiconductor (CMOS) technology.

Table 1.1: Parameters and Objectives for the circuit

Parameters	**Objectives**
V_{DD}	1.2 V
Resolution	$<$ 20 ps
Vin_{Low} - Vin_{High}	0 - 1.2 V
Frequency	100 MHz
Number of Bits	4
Jitter	$< 100\ fs$
INL error	$<$ 0.25 LSB
DNL error	$<$ 0.25 LSB

The developed TDC architecture was compared with state-of-the-art TDC architectures. An attempt to improve the circuits linearity was done with dummy inverters, but was not successful, section 5.1.6.

1.3 Document Organization

Chapter two of this work will present some background into PLLs, Noise and TDCs.

Chapter three is the State of the Art on TDCs and Thermometer to Binary Decoders. It presents a study of the basic principles and concepts behind the first and second generation TDCs, the advanced TDCs and the basic blocks that compose them. In addition it also presents the most common architectures for Thermometer to Binary Decoders.

Chapter four describes the chosen TDC topology, it gives an overview of the purpose of each block in the circuit. Then each block is described and dimensioned. After the blocks are sized the TDC is assembled.

Chapter five is the circuit simulations and results. It presents and describes the simulation techniques, including corner analysis.

At last chapter six is the conclusion that presents the results of the work, resumes and suggests future work to improve the circuits performance.

2

Background

2.1 Phase locked Loops

A basic PLL, Figure 2.1, is a closed loop system that consists of a phase detector, a loop filter, and a Voltage Controlled Oscillator (VCO). The phase and frequency of the output signal is tracked and feedback to the phase detector or comparator where it's compared to the input signal. The purpose of the PLL is to lock the phase and frequency of the output to that of the input signal.

The Phase detector or comparator block, compares the reference input signal to the feedback signal. With the difference between the signals, a proportional error signal is produced at the output.

The loop filter then converts the error signal into a proportional voltage, that voltage is then used as the control voltage that is fed to the VCO. [4]

The VCO generates a high frequency signal at the output based on the voltage from the loop filter.

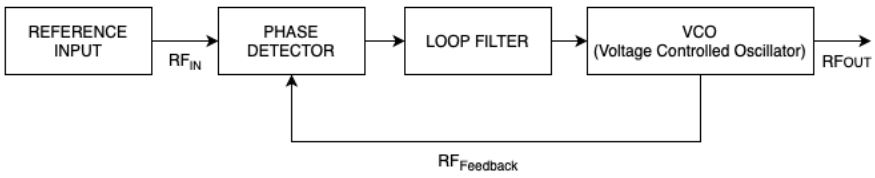


Figure 2.1: Basic Block diagram of a PLL

2.1.1 Analog PLL

The phase detector in an Analog PLL is made by combining a multiplier with a low pass filter. The multiplier block is connected to the reference input signal and the feedback signal. The multiplier then produces the output error signal that is proportional to the phase difference between the two input signals. This signal is fed to the low pass filter which translates the error signal into voltage. A frequency divider could be added

to the Feedback line, where the feedback frequency would be divided by N, but the value N in analog PLL is 1. Thus, a feedback divider is not needed. [4]

2.1.2 All-Digital PLLs

In a All Digital Phase Locked Loop (ADPLL) the phase detector is part of another block, the error detector, that consists of the phase detector and a charge pump. A TDC can be used as a phase frequency detector. The VCO is substituted by a Digitally Controlled Oscillator (DCO), and a frequency divider is added to the feedback line, Figure 2.2.

The output of the phase detector in an ADPLL is either high or low depending if the feedback frequency lags or leads in relation to the reference frequency.

This output is then fed into the charge pump that contains a positive and negative current source. If the phase detector output was high then the charge pump output will be a positive current, and if the phase detector output is low then the charge pump output will be a negative current.

The loop filter then removes any phase noise that might occur, and translates the current pulses form the charge pump into constant voltage. This voltage is then used to control the DCO. [4]

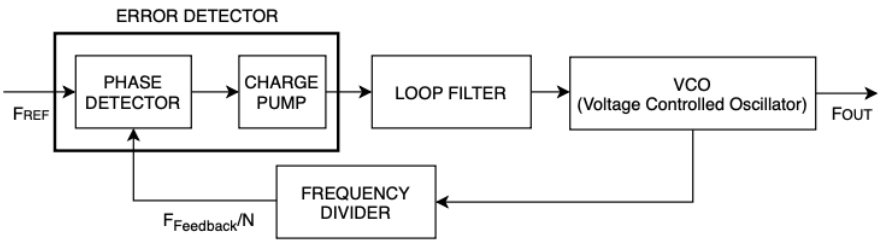


Figure 2.2: Block diagram of a ADPLL

2.2 Noise

Noise is a random process, and its instantaneous values in the time domain cannot be accurately predicted. And is one of the most important aspects to have in consideration when designing PLLs is the phase noise and the jitter. They can appear from electronic noise, and the supply and substrate noise.

2.2.1 Phase Noise

The phase noise, $\mathcal{L}(f)$, is the random variation of the output amplitude or phase. The phase noise can be calculated with equation 2.1, for a 1 Hz bandwidth with a specific

offset frequency, f_m, and a specific carrier frequency, f_0.[5]

$$\mathcal{L}(f) = \frac{P(f_m)}{P(f_0)} \tag{2.1}$$

This parameter can be measured easily with a spectrum analyzer, or with a phase or frequency demodulator. The measurement comes in dBc/Hz.

In Figure 2.3 it's possible to observe a typical output noise spectrum.

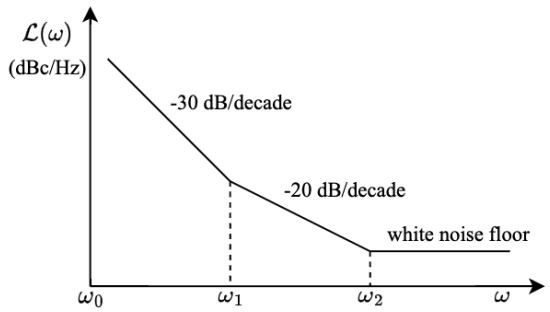


Figure 2.3: Phase Noise spectrum

The plot is divided in three different regions [5]:

- The first region with a -30 dB/decade slope or the $1/f^3$ region is the closest to the carrier frequency. It has a frequency between ω_0 and ω_1, and is the major noise contributor for the $1/f$ noise of the active devices.
- The second region with a -20 dB/decade slope or the $1/f^2$ region. The slope is due to the frequency modulation by white noise sources, and it has a frequency between ω_1 and ω_2.
- The third and last region has no slope, and is the furthest away from the carrier frequency. Its phase noise is the result of the white noise from the sources connected to external devices.

The band of integration for the phase noise in this work is from 10 kHz to 20 MHz. To avoid the flicker noise and to not include the full jitter.

2.2.2 RMS Jitter

The jitter used in this work is the rms jitter. This rms jitter is the displacement of the rise and fall time of a clock from its ideal value. While the phase noise is normally represented in the frequency domain, the jitter is represented in the time domain. It can be obtained through equation 2.2, using the phase noise. The division by $2\pi \cdot f_o$ transforms the jitter value from radians to seconds. [6]

$$J_{rms} = \frac{\sqrt{\int_{-\infty}^{\infty} S_{\phi n} \cdot f \cdot df}}{2\pi \cdot f_o} \tag{2.2}$$

The white noise is the main contributor to jitter and phase noise for an inverter stage,[7]. So considering the CMOS inverter, with one Negative Channel Metal Oxide Semiconductor (NMOS) and one Positive Channel Metal Oxide Semiconductor (PMOS). When only the NMOS is working the current noise has a spectral density of

$$S_{in,N} = 4KT\gamma gm = 8KT\gamma \frac{I_D}{V_{DD} - V_{th}} \tag{2.3}$$

considering I_D. The noise introduced by the NMOS during the switching phase adds to the previous noise, introduced by the the PMOS in the load capacitance. So the total jitter results in the equation 2.4. [6]

$$\sigma^2_{tdN} = \frac{4KT\gamma td}{I_D(V_{DD} - V_{th})} + \frac{KTC_L}{I_D^2} \tag{2.4}$$

Another contribution is the flicker noise that in represented in equation 2.5.[8]

$$\overline{V^2_{n,f}} = \frac{K_f}{C_{ox}WLf} \tag{2.5}$$

3

State of the Art

3.1 Analog TDCs

Time to digital Converters are highly precise stopwatches that convert the information on the time domain into a digital representation. They represent the fundamental block between time encoded data and the digital world. TDCs can be classified into two classes, delay line TDCs and oscillator based TDC. They are frequently used as phase frequency detectors in ADPLLs.

The challenges in TDCs appear when continuous measurement, long time intervals, high resolution, linearity, low power consumption, and robustness. The architectures to fix this one or more of these issues are becoming more complex at a very fast pace, this is because of the technology scaling to the ultra-deep sub-micron regime, which results in the decrease of the intrinsic gain of a single MOS transistor. [2]

There are the classical ways to fix the decreases in gain, like using Cascode transistors, but even with this possible it decreases the speed of the circuit and the signal swing. The last one particularly disadvantage because it reduces the Signal-to-Noise Ratio (SNR). [2]

The design of classical mixed-signal circuits, which have better performance, has become increasingly difficult and time consuming. Because of this the use of this circuits will continue to decrease, and the ones that are left will be supported by digital enhancement techniques and digital post-processing of imperfect data. [3]

The traditional TDC conversion approach starts by converting the time interval into voltage. Then converting the voltage to digital with an Analog-to-Digital Converter (ADC), Figure 2.1.

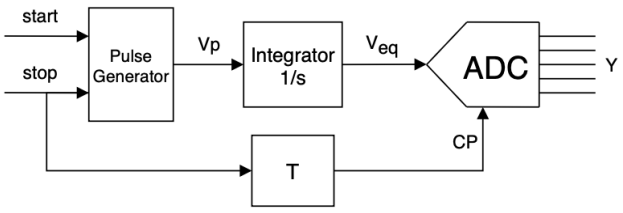


Figure 3.1: Block and Signal of a basic TDC

The start and end signals are used to form a pulse with a width that corresponds to the time interval to be measured. This pulse is then processed by a integrator that transforms the pulse into voltage. At last the ADC returns the equivalent digital word. The number of bits returned by theADC determines the dynamic range of the TDC.

$$DR = 2^n * Tlsb \tag{3.1}$$

Even though a basic TDC uses a simple conversion principle, all blocks need to meet the linearity demands of the overall TDC, it needs to have a basic integrator implementation that allows high-speed, because of the finite output resistance of the current source the linearity is weak, to improve this an active RC-integrator is used, and to finish the finite bandwidth of the Operational Amplifier (op-amp) limits the speed and the minimum time interval. [2]

A more elaborate TDC is necessary, because absolute time measurements need information on the current and the capacitance value, and this can't be done without calibration. The dual-slope TDC, Figure 3.2, is a reversed dual-slope analog-to-digital converter, in the ADC the unknown analog signal is integrated for a well-defined time, in the TDC a well-defined voltage (pulse height) is integrated for a time interval. An advantage of the dual-slope in relation to single-slope is that the absolute device values cancel out. [2]

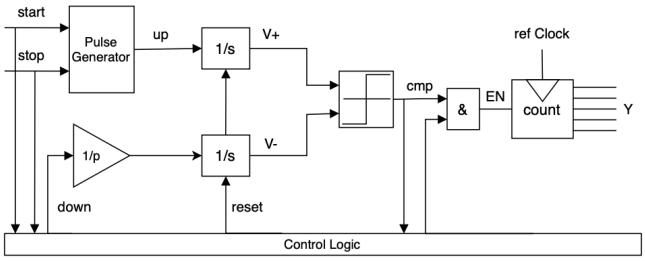


Figure 3.2: Analog TDC with Dual SLope analog-to-time interpolation

In both architectures the conventional ADC is the main mechanism, so their performance is highly dependent on the linearity and mismatch of their elements, and the resolution of the voltage-mode ADC. Unfortunately, the voltage mode ADC has a negative scaling effect with technology, so analog based TDCs will also not scale well with technologies.

3.2 Digital TDCs

The traditional TDCs consist mainly of an ADC, however the advantages of using the time domain are only applicable if there is no analog conversion step in the time-to-digital conversion. The simplest technique for a digital conversion, to quantize a time interval, is to count the cycles of a reference clock fitting into the respective measurement interval. The measurement accuracy can be increased by increasing the clock frequency,

unfortunately this results in a higher power consumption, and above a certain frequency CMOS based oscillators are not available and more expensive oscillators are required. A higher resolution can be achieved by subdividing one clock signal asynchronously into smaller time intervals. [2]

The subdivision of the interval is done by using multiple phases of the reference clock. This can be done, for a medium resolution, by using a ring oscillator that generates k equally spaced versions of the clock signal.

3.2.1 Delay line TDCs

To obtain a better resolution the original reference clock is delayed in a chain of digital delay elements. When the signal arrives all versions are read in parallel, with latches or flip-flops being used as sampling elements. The sampling element freezes and reads the state of the delay-line at that instance. This delay line TDC and its output signals can be seen in Figure 3.3. The resolution is limited by the delay of the delay element of the the delay line, in this case the buffer.

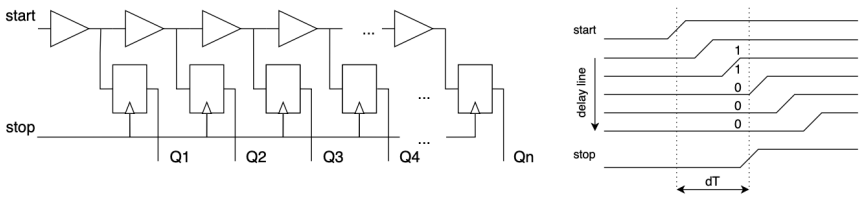


Figure 3.3: Delay line TDC and output signals

3.2.1.1 Inverter Based delay line

The resolution of delay-line based TDCs can be doubled by replacing the buffers with CMOS inverters. The use of inverters unfortunately means that both the rising and falling transitions are used for the measurement, which results in a difference between the delay time of consecutive inverters in the delay line. This happens because if the first inverter uses the NMOS device the following inverter will use the PMOS device. To correct this problem two delay lines are used to make the TDC more robust, and the inputs of the flip-flop are twisted in alternating stages, this way the flip-flop used can be asymmetrical, Figure 3.4.

3.2.2 Basic Blocks

3.2.2.1 Inverter

The most basic block in delay line TDCs is the inverter. A inverter is a basic circuit used in all digital circuits, but with down scaling has started to be used beyond digital

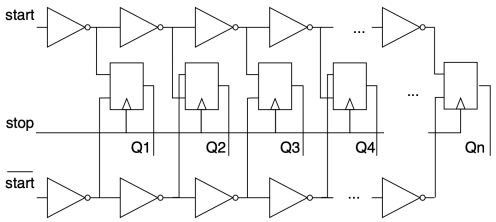


Figure 3.4: Inverter Delay Line TDC

circuits. It consists of a NMOS pull-down transistor, and a PMOS pull-up transistor, as shown in Figure 3.5.

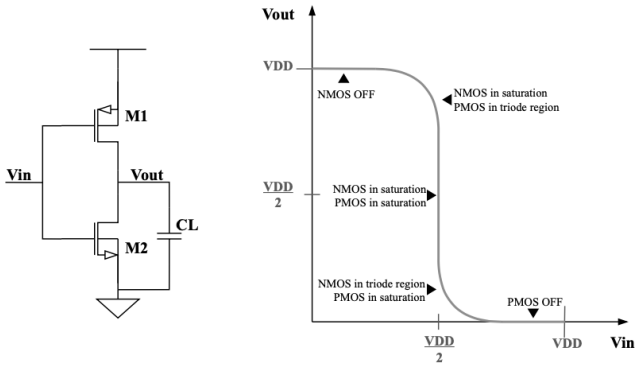


Figure 3.5: CMOS Inverter and Transfer Function

The transfer function, in Figure 3.5, is symmetric. This means that when V_{out} equals V_{in} both transistors are saturated, and that voltage is the V_{th}. It also means that the current in M1 is the same as in M2. So to calculate the V_{th} the equations 3.2 and 3.3, are used.

$$\frac{k_n}{2}(\frac{W}{L})_n(V_{th}-V_{tn})^2 = \frac{k_p}{2}(\frac{W}{L})_p(V_{DD}-V_{th}-|V_{tp}|)^2 \tag{3.2}$$

$$V_{th} = \frac{V_{DD}-|V_{tp}|+V_{tn}\cdot r}{1+r} \tag{3.3}$$

$$r = \sqrt{\frac{k_n(\frac{W}{L})_n}{k_p(\frac{W}{L})_p}} \tag{3.4}$$

For the transistors to be matched $k_n\ (W/L)_n = k_p\ (W/L)_p$, so r (3.4) is 1, and V_{tn}=$|V_{tp}|$. Switching these values in 3.3, the V_{th} is $V_{DD}/2$.

When V_{in} equals V_{DD} , the PMOS acts as an off switch and the NMOS as a low value resistor, R_{DSN}, so the V_{out} value will be given by the NMOS, therefore V_{out} will be zero volts. When V_{in} equals zero volts the NMOS acts as an off switch and the NMOS as a low value resistor, R_{DSP}. The formula for $R_{DS(N/P)}$ is given by [9], and can be found bellow.

$$R_{DS(N/P)} = \frac{1}{k_{n/p}\cdot(\frac{W}{L})_{n/p}\cdot(V_{DD}-V_{th})} \tag{3.5}$$

With both transistors in saturation the gain can be obtained through the small signal circuit (Figure 3.6). This gain is given by Equation 3.6.

$$Av = -(gm_n + gm_p)(ro_n \| ro_p) \tag{3.6}$$

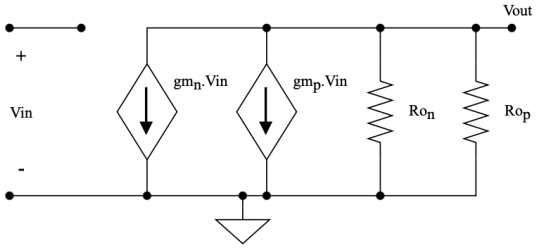


Figure 3.6: Inverter Small signals circuit

The propagation delay of the inverter is the time the output takes arriving to the expected result. If V_{in} equals V_{DD} then the time it takes for V_{out} to reach $V_{DD}/2$ is Tp_{LH}, the opposite is Tp_{HL}. The delay of the inverter is given by an average of these two, and can be found in the equation below.

$$Tp = \frac{Tp_{HL} + Tp_{LH}}{2} \tag{3.7}$$

The F_{max} is given by the inverse of the Tp_{min} of the inverter. Tp_{min} is two times the delay of the inverter.

The dynamic operation of the circuit can be analysed with Equation 3.8. The current I that goes through the capacitor C, during a time interval Δt, leaves a charge ΔQ on the capacitor, this causes the voltage in the capacitor to increase by ΔV.

$$I\Delta \cdot t = \Delta Q = C \cdot \Delta V \tag{3.8}$$

The previous equation can be adapted to to calculate Tp_{LH} or Tp_{HL}. The following equation shows the balance equation for Tp_{HL},

$$I_{avg} \cdot Tp_{HL} = C \cdot [V_{DD} - (\frac{V_{DD}}{2})] \tag{3.9}$$

that results in,

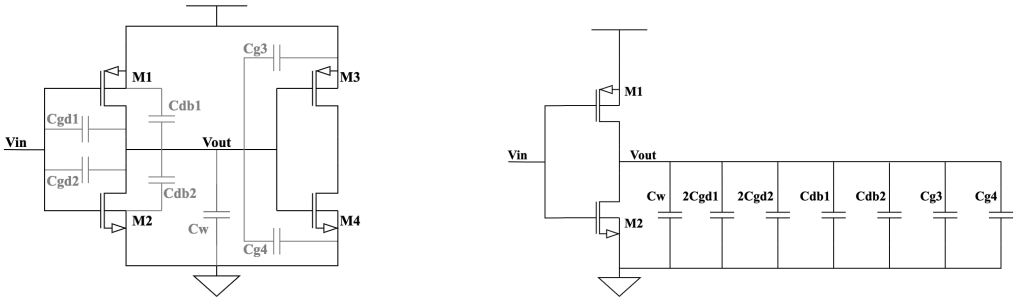
$$Tp_{HL} = \frac{C \cdot V_{DD}}{2I_{avg}} \tag{3.10}$$

This equation can be simplified and applied to both Tp_{HL} and Tp_{LH}, and results in the following equations

$$Tp_{HL} = 0.69 \cdot R_N \cdot C \tag{3.11}$$

and

$$Tp_{LH} = 0.69 \cdot R_P \cdot C \tag{3.12}$$



(a) Circuit with Parasitic Capacities (b) Equivalent Circuit

Figure 3.7: Inverter Circuit with Capacities

The resistances can be calculated through Equation 3.13, considering the empirical values of ro_n is 12.5 kΩ and ro_p is 30 kΩ.

$$R_{N/P} = (\frac{ro}{W/L})_{N/P} \tag{3.13}$$

With the resistance calculated, all that's left to calculate the time delay is determining the equivalent Load Capacitance, C_L. The inverter circuit with the respective capacities can be observed in Figure 3.7a, the inverter in the circuit is driving an equally sized transistor to better simulate real life conditions. The capacities in the circuit are the internal capacities of the CMOS transistors and the wiring capacity. In the equivalent circuit the capacities Cgd1 and Cgd2 are replaced by their double capacitance. Originally these capacities are not connected to the ground, but to V_{in} and V_{out}, as the voltage in V_{in} goes high the same amount that V_{out} goes low, the change in voltage in the capacitance is twice the amount. Therefore if they are to be connected between V_{out} and ground their values will be doubled. This happens because of the Miller effect, and the total load capacities can be seen in 3.7b. [9]

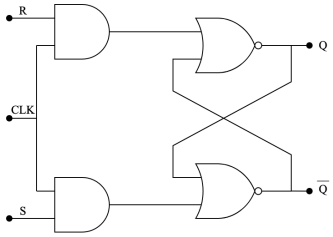
These internal capacities can be calculated using the equations A.1, A.2, A.3 and A.4, in appendix A.

With the previous equations it's possible to predict that the delay of the inverter will decrease as the width of the transistors increases, because the $R_{N/P}$ resistances will lower faster than the load capacitance of the circuit increases. It is also possible to predict that the noise will be lower with the increase of the width, while the power will increase.

3.2.2.2 Flip Flop

Flip-Flops work as time samplers in delay line TDCs, and can be static or dynamic logic circuits. The most basic flip-flops is the Set/Reset Flip-Flip (SRFF) and can be made by cross-coupling two NOR gates. In delay line TDCs the SRFF need to be clocked, to do this two AND gates are connected to the input of the NOR gates, the AND gates are connected through one of their inputs and this connection is where the clock is connected.

A basic SRFF can be observed in Figure 3.8a. When the clock is low no change will come to the circuit the same will happen when both the Set and Reset inputs are low.



(a) SRFF Schematic

CLK	R	S	Q_{n+1}
0	X	X	Q_n
1	0	0	
1	0	1	1
1	1	0	0
1	1	1	NA

(b) Truth Table

Figure 3.8: SR Flip Flop

When the clock is high ,the Q output will follow the value of the Set input and $\overline{Q}$ will follow the value of the input Reset, with the exception of when both the Set and Reset inputs are high then both outputs will give an undefined value. This can be observed in Figure 3.8b. This flip flop can also be implemented using only NAND gates, by inverting the Reset and Set inputs.[9]

Although the previous SRFF works well, it is a complex circuit when the gates are translated into their respective CMOS circuit. So a simplified circuit is needed, Figure 3.9. In this circuit the clock inputs form a AND function with the set and reset inputs, so the SRFF can only be set or reset when the clock is high. This circuit does not lose static power as there is no path between V_{DD} and the ground.

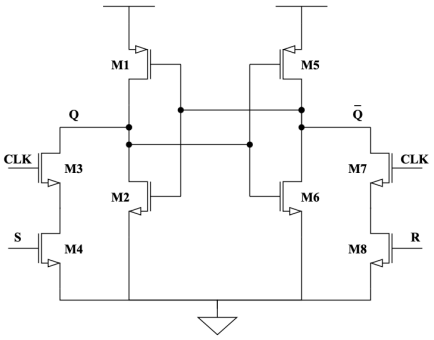


Figure 3.9: CMOS clocked SR flip-flop

This SRFF has the same truth table as a NAND gate only SRFF. As long as the clock is low the state will not change from the previous state, in case the this is the first state and there is no previous state to maintain the SRFF outputs will default to the V_{th}. When the clock is high, the Set input is high and the Reset is low the transistors M3 and M4 will conduct and pull the Q voltage down. When the Q voltage reaches V_{th} the inverter will activate and push $\overline{Q}$ up, this will then push Q even lower. The same will happen to the other side of the circuit if Set is low and Reset is high, do to the symmetry of the circuit. When all inputs are high both pairs of transistors M3/M4, and M7/ M8, pull respectively Q and $\overline{Q}$ down, therefore when they reach V_{th} neither can push the other up, so the outputs will both be V_{th}.

As for the sizing of the Flip Flop, the equivalent circuit needs to be analysed, Figure

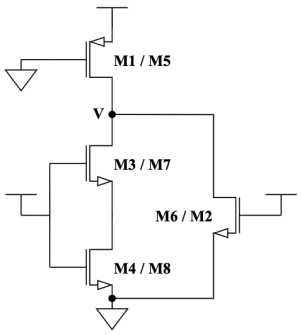


Figure 3.10: CMOS SR Flip-Flop equivalent circuit

3.10. Half of the SRFF is considered and the PMOS transistor is connected to ground and the NMOS transistors are connected to V_{DD}, to force the voltage V to $V_{DD}/2$. As can be seen in the circuit the M6/M2 is equivalent to the transistors in series, so M6/M2 needs to be half of their sizes. The transistors M3/M7 and M4/M8 also need to pull the voltage in Q down so they need to be bigger than M1/M5 to minimize the delay of the flip-flop. The relation between the sizes of the transistors in the inverters will be analysed further in this work. This sizing relations are shown in the equations 3.14 and 3.15.

$$(\frac{W}{L})_{3/4/7/8} = 2 \cdot (\frac{W}{L})_{2/6} \tag{3.14}$$

$$(\frac{W}{L})_{1/5} = x \cdot (\frac{W}{L})_{2/6}, \quad x < 2 \tag{3.15}$$

3.3 Advanced Topologies

3.3.1 Bipolar TDC

The previous TDCs have an asymmetrical architecture, the start signal arrives through the delay line, at the flip-flops before the stop signal, because the stop signal is what triggers the flip-flops and allows the time measurement. If the start signal arrives after the stop signal, the output would be null. So they can only measure positive time intervals.

As some applications need to measure negative time intervals, for example type II PLLs, the previous TDCs can be used by applying an offset in the stop signal. This technique has the disadvantage of an unknown offset error in the TDC characteristics, that is easily influenced by process variations and operation conditions, like supply voltage and temperature.

An advanced architecture that can be used to solve these issues is the bipolar TDC (Figure 3.11). Two standard TDCs are connected so that the forward TDC measures the time interval x to y and the reverse TDC measures the time interval y to x. This way neither of theTDCs has to measure negative time intervals. The output signal can be obtained subtracting the reverse TDC output from the forward TDC output.

The disadvantage of this architecture is the offset. There is an overlap region where both TDCs are measuring, as TDCs normally allow a slightly negative time measurement, so in the overlap region the gain is twice what is normally is. This results in non-linearity, which can be an issue in some implementations. [2]

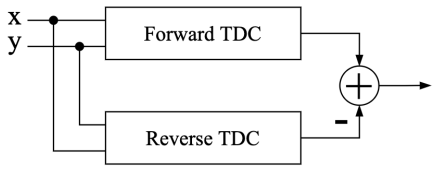


Figure 3.11: Basic Bipolar TDC

3.3.2 Looped TDC

The TDCs previously discussed where linear TDCs, as they don't have feedback. The length of these circuits grows with the maximum time interval to be measured, T_{Max}, so for long time intervals the circuit area is large, and so is the power consumption. The looped TDC architecture is a solution to this problem, the end of the delay line is connected to its beginning through a counter. This counter determines how many times the signal looped before the TDC is stopped. The value measured, B, is given by equation 3.16. B_{cnt} gives the number of cycles counted by the clock, M the number of elements in the delay line, and B_{TDC} the position of the start signal in the delay line when the stop signal arrives.

$$B = M \cdot B_{cnt} + B_{TDC} \tag{3.16}$$

The start signal enters a multiplexer, this multiplexer passes either the start signal or the feedback signal to the delay line. The circuit possesses a control unit that toggles the multiplexer. After the start signal passes the multiplexer only lets the feedback signal pass, until the arrival of the stop signal, then goes back to letting the start signal pass. The most important part in this circuit is the detection of the stop signal, because the start and stop signal are asynchronous.

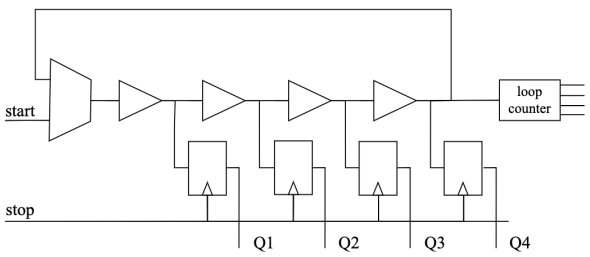


Figure 3.12: Basic Lopped TDC

Area consumption is not the only reason to use looped TDCs, increasing the arrival time uncertainty in delay-lines with local process variations. One disadvantage of this circuit is that because of its loop geometry it's very difficult to not cause layout asymmetries. To solve this problem the elements in the delay line can be substituted by multiplexers with one of the inputs used as a dummy. This causes the area of the circuit and power consumption to increase and the resolution to worsen. [2]

3.4 TDCs with sub-gate delay resolution

The resolution of the previously discussed TDCs is limited by the technology, the delay of the inverter. This is called technology delay, T_{tech}. A higher resolution can only be obtained with a faster technology or with new techniques. To be considered a TDC with sub-gate delay resolution, the resolution has to be better than the technology resolution. The relation between these two is the sub-gate delay interpolation factor and can be calculated with the following equation.

$$IF = \frac{T_{tech}}{T_{LSB}} \tag{3.17}$$

3.4.1 Parallel TDC

The technology delay of the inverters scales linearly withe the width of the transistor and the load capacitance. This delay can be calculated with the following equation, where p is the parasitic delay, g is the logical effort, and h is the fan-out of the gate ($h = \frac{C_{load}}{C_{in}}$). In delay lines each gate is loaded by an identical gate so h equals 1.

$$t_d = p + gh \tag{3.18}$$

So by configuring the gates in parallel, Figure 3.13, instead of in line, this technique allows to control the h parameter, by either changing width of the transistor or the load capacitance. So theTDC consists of an n number of gates in parallel, each with a different load capacitance. The load capacitance in each gate is the same capacitance n times all in parallel, this results in a very linear system output The parasitic delay contributes for the offset of the circuit but this does not affect the measurement accuracy.

Like the delay-line TDCs the stop signal activates the sampling elements. In this TDC however the start signal reaches all the delay elements at the same time. The delay of each delay element can be calculated with the following equation. T_d^0 refers to the technology delay, and the circuits offset, ΔT_d^0 to the delay variation introduced by each capacitance, and n to the number of capacitors in parallel with the delay element. [2]

$$t_{d,n} = t_d^0 + \Delta t_d \cdot n \tag{3.19}$$

The resolution of this circuit is given by $T_{LSB} = \Delta t_d$, and its range is given by $N \cdot T_{LSB}$

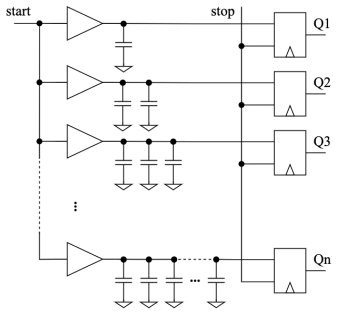
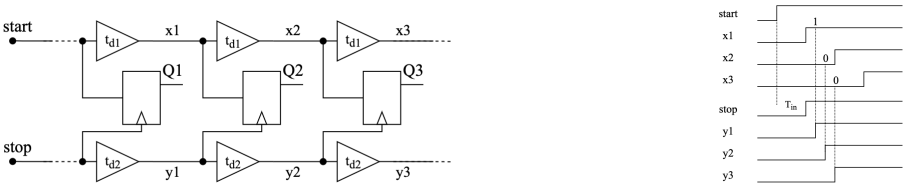


Figure 3.13: TDC with parallel scaled delay elements



(a) Cut out of a Vernier Delay Line

(b) Operating Principle

Figure 3.14: Vernier TDC

The area of the circuit, not counting wires can be calculated considering that there are N delay elements, N flip flops, and $\frac{1}{2}N(N+1)$ capacitors. So the area can be calculated using the following equation, considering that A^{inv}, A^{FF} and A^{cap}, are respectively the area of the delay element, the flip-flop, and the capacitor.

$$A_{parallel} = N(A^{inv} + A^{FF} + \frac{1}{2}(N+1)A^{cap}) \tag{3.20}$$

3.4.2 Vernier TDC

A TDC that has sub-gate delay resolution and is still a delay-line TDC is the Vernier TDC, Figure 3.14a. It is the most popular TDC technique for sub-gate delay resolutions. because it can be implemented in many architectures, like a simple delay-line, or in a looped structure. Another reason for its popularity is the very simple operating principle and high resolution.

A Vernier TDC has two delay lines, with the same number of elements, one for the start signal and one for the stop signal. The delay elements on the start signal delay line have a higher delay, t_{d1}, than the ones on the stop signal delay line, t_{d2}. The stop signal happens after the start signal, but delays analysed are smaller along the TDC, Figure 3.14b. When the stop signal catches up with the start signal, the measurement ends. The measurement is done with Early-Late-Detectors, ELDs, that are normally flip-flops. This instance can be calculated, with equation 3.21, where N is the number of stages that the signals passed through before the stop signal catches up with the start signal, and T_{in} is

the original delay between the start and stop signals.

$$T_{catch} = N \cdot t_{d1} = N \cdot t_{d2} + T_{in} \tag{3.21}$$

The resolution of this TDC, T_{LSB}, is the delay difference between the delay elements in the $(t_{d1} - t_{d2})$. So to determine the number of stages, N, needed for a certain time measurement, T, the time measurement needed to be divided by the circuits resolution $(\frac{T_{max}}{T_{LSB}})$. Considering that each stage has two buffers and a flip-flop, the area of the circuit without wires is

$$A_{Vernier} = N(4A^{inv} + A^{FF}) \tag{3.22}$$

Like classic delay lines the effects of mismatches, and temperature and voltage variations, have a huge impact on the TDCs dynamic range, that prevent it from scaling linearly. The Vernier in loop structure can solve this problem because it can have big time intervals using few delay elements. [2]

If the delay-line is big enough, and the measured sample is large compared to the resolution, it is possible to inject a new signal while the previous one is still running. This will require more control, to provide proper results, this is considered a multi-event time-to-digital converter.

3.5 TDCs Summary

The previous sections provided a brief introduction to different TDCs. Their working principles, their architectures, advantages and disadvantages were presented.

The first fully digital TDCs were delay-line TDCs, buffer or inverter based, they provide a good resolution and low power consumption. However the resolution is limited by the technology and the only way to increase the dynamic range is to increase the delay-line, this results in an increase in area, power consumption and non-linearities in the system performance. This comparison can be observed in Table B.1 in appendix B.

One of the easiest methods to achieve sub-gate resolution is using a parallel scaled delay-line. This increases the resolution of the TDC, and is based on scaled capacitors connected to the output of the parallel delay elements. This techniques disadvantage comes from the circuits area and the parasitic effects. This can be observed in Table B.2 in appendix B.

To achieve sub-gate resolution using a delay-line the Vernier TDC is used. It used different sized delay elements in two different delay-lines. However the high resolution and linearity, comes with a big area and power consumption. The comparison between the last two techniques can be observed in Table B.2 in appendix B.

3.6 Thermometer Decoders

As previously mentioned TDCs outputs come in thermometer code, so the next step to obtain an easiest too measure result is to transform the thermometer code into binary. The thermometer code is the sequence of high and low outputs that indicate where the measurement ends in the delay-line. The truth table is in Table 3.1.

Table 3.1: Thermometer to binary code converter truth table

Decimal Number	Thermometer Code	Binary Code
0	0000000	000
1	0000001	001
2	0000011	010
3	0000111	011
4	0001111	100
5	0011111	101
6	0111111	110
7	1111111	111

There are various approaches for thermometer to binary decoders, like ROM, Wallace-tree, fat tree decoder and logic based.

The ROM-based decoder is the most frequently used for flash ADCs, because of its parallel architecture There are two types of ROM decoders, Binary and Gray. This decoder has the advantage, of its simple structure. However it has a low speed conversion and a high power consumption. For increased data speeds it's also susceptible to bubble errors. Presented in Figure C.1 in appendix C.

The Wallace-tree processes the thermometer code directly. It counts the number of ones, and converts that number to binary code. This technique has global error suppression, and can be found in Figure C.2 in appendix C.

The Fat-Tree Decoder, uses two stages of conversion. First it converts the thermometer code to 1-of-N code, then this code is converted in to binary, Figure C.3 in appendix C.

The Fat-tree circuit is delayed $log_2 N$, the ROM circuit is delayed N, and the Wallace-tree is delayed $log_{1.5} N$. Being N the 2^{nbits}-1. So the faster decoder is the Fat-tree circuit. [10]

At last there is the logic-based decoder. It is an all digital decoder, that has the best performance against bubble errors, and consumes less area. To achieve high speeds with this architecture, there needs to be pipelining and storage elements, this results in a higher power consumption. A 4 bit logic-based decoder is shown in Figure C.4 in appendix C.

4

TDC Topology

In Chapter 3, various TDCs architectures were presented. In this chapter an overview of the chosen TDC architecture and all of its blocks will be made, along with the chosen thermometer.

4.1 Topology Overview

The chosen TDC is the inverter based TDC, for its simplicity and because the objective is to reach the best technology resolution. The end circuit will be a basic TDC easily adaptable to new functions by adding or excluding blocks.

The circuit will contain two same size delay-lines, SRFFs, and a thermometer. The number of elements of each delay-line and SRFFs will be the number of states for the chosen number of bits minus one ($2^{nbits} - 1$).

The only block that is not detrimental to the function of the circuit, and is only being used to aid measurements, is the Thermometer Decoder so it will be an ideal block.

To further test the circuit the V_{DD} ideal source will be substituted with a Low-dropout regulator (LDO).

4.2 Delay Line Sizing

The Delay lines are imperative to the circuits performance. Their main requisite in this circuit is to deliver a high resolution and a low phase noise, with a good linearity. The delay-lines are made up of same sized inverters.

4.2.1 Inverter Sizing

As was explained in section 3.2.2.1, for the transistors to be matched $(W/L)_p$ needs to be equal to k_n/k_p times $(W/L)_n$. Since the length will be the lowest possible, 120 nm, this translates to $W_p = k_n/k_p \cdot W_n$. So the first thing to do while sizing the inverters is to discover the k_n/k_p for this technology. Using the test bench in Figure D.1. With a dummy

inverter to simulate the delay-line, and ideal sources. For this technology it is expected that the value will be between 2 and 3, according to [9].

Doing a DC analysis varying the k_n/k_p and obtaining the transfer function, for which the results can be observed in Figure 4.1. So for this technology k_n/k_p is 2.65.

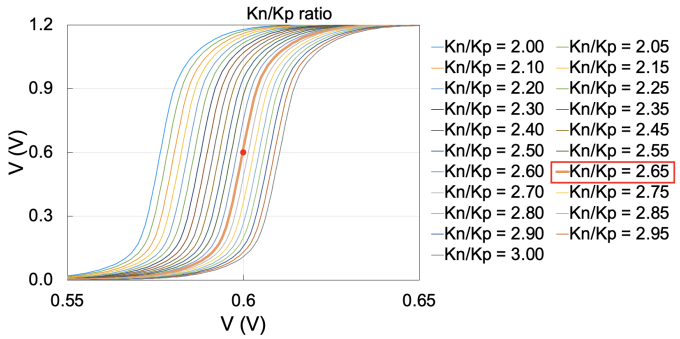


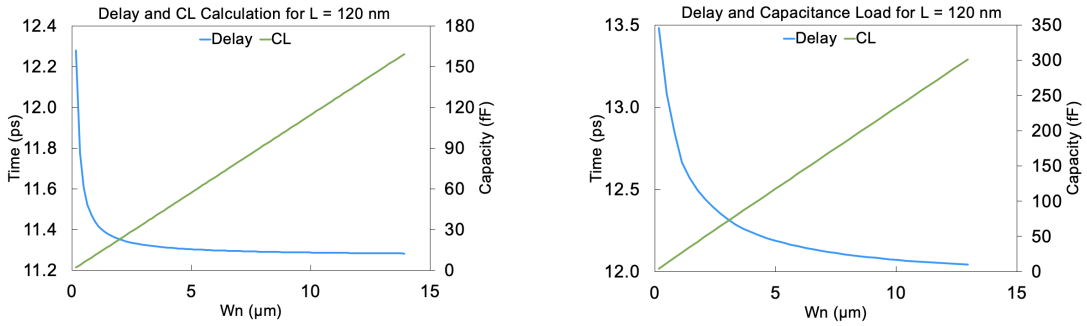
Figure 4.1: k_n/k_p ratio for technology

Now that the k_n/k_p ratio is defined, all that's left is to choose the W_n so that the delay is the minimum possible and matches the required resolution parameter presented in Table 1.1.

To study how the inverters delay changes, the code in listing F.1 was developed. The graphs in Figure 4.2a is the result. This graph confirms the results predicted in section 3.2.2.1. That the delay decreases as the capacitance load increases, and that the capacitance increases with the width.

A simulation was then done to confirm the values previously obtained. The simulation results are in Figure 4.2b. This values were obtained using the code in listing F.2 As can be observed the values are higher than the calculated this is due to the simplified equations and circuits used in the calculations.

Considering the graphs in Figure 4.2 the values 6 and 12 μm where chosen for W_n, because as was observed in the graphs these values don't show a significant difference between their delays, and don't have conversion issues when simulating. Two values where chosen to see the difference between their contribution to the circuits noise and



(a) Calculation

(b) Simulation

Figure 4.2: Inverter Delay Study

power consumption. It's expected that the circuit with the lower W_n will have a higher circuit noise and a lower power consumption than the circuit with the higher W_n. In the complete circuit the delay is expected to rise because the capacitance load will be higher in a delay line than in two inverters in a row.

4.3 Flip Flop Sizing

To keep the circuit as simple as possible the W_n sizes for the SRFF are the same as for the inverter, so 6 and 12 μm. However the relation between W_n and W_p is not the same as for the inverter, so the first step is to discover the optimal x value, from equation 3.15.

The test bench used is in Figure D.2 in appendix E, the sources are the only ideal component, and the W_n used for the simulation was 6 μm.

The results can be observed in Figure 4.3, where it's possible to observe that the Flip Flop is symmetrical. The x value which results in the lower delay for the rise transition is 1.5, and either of the outputs, Q or $\overline{Q}$, can be used.

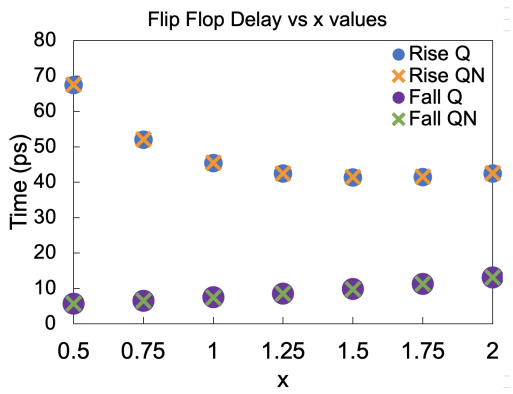


Figure 4.3: Flip Flop Delay in relation to x value

Through a transversal analysis it´s possible to confirm that the SRFF is working as expected, table 3.8b, with the difference that the R and S inputs are switched this does not affect the end result. This can be observed in Figure 4.4. The voltage peaks seen in the figure are due to the use of ideal sources in the simulation.

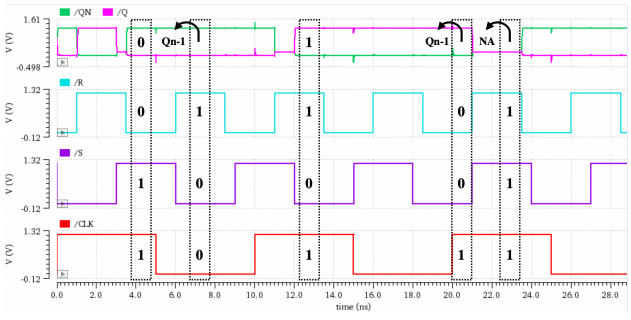


Figure 4.4: Flip Flop transversal signal with logic values

4.4 Thermometer-to-Binary Decoder

For this thesis the Thermometer decoder is an ideal block, so a logic-based decoder was chosen.

The block can be seen in Figure C.4, where ideal logic gates were used. In input i8, a two inverters were placed between the input and the output to separate them without changing the logic value.

The transient result of this thermometer can be observed in Figure 4.5. Where the inputs are equally spaced, low to high transitions, to simulate a thermometer code. As can be observed the binary output corresponds to the expected, shown in Table 3.1.

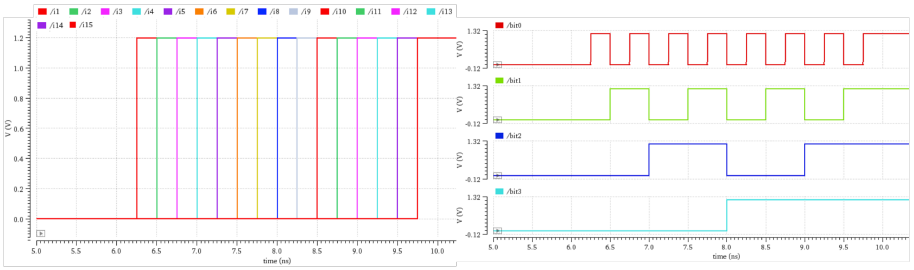
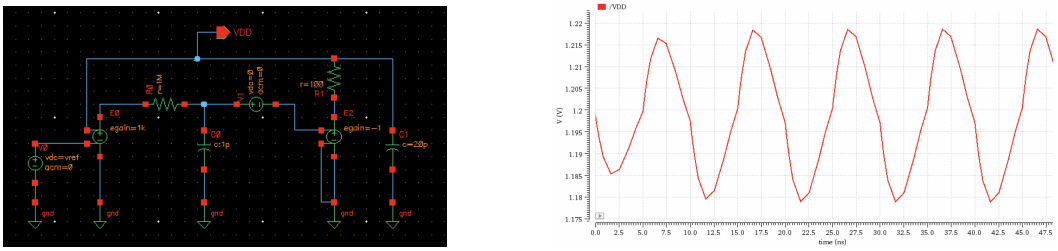


Figure 4.5: Logic-based Thermometer Decoder input and output transient response.

4.5 LDO sizing

The LDO block used can be observed in Figure 4.6a. The V0 source defines the output tension of the LDO, in this case it is 1.2 V.

R0 and C0 define the circuits bandwidth. E0 defines the low frequency gain for theLDO. R1 defines the output impedance. C1 defines the decoupling capacitance, this capacitor needs to be big enough so that the tension does not go lower than 10% of its initial value. V1 source is to be used in stability simulations. At last to test only the LDO a current source cam be added in parallel to the C1 capacity, to simulate the current pulled by the circuit. The circuits transient response can be observed in Figure 4.6b.



(a) Block

(b) Transient response

Figure 4.6: LDO circuit and simulation

4.6 TDC Architecture Summary

As the inverter based TDC is required to have 4 bits, it will have 15 phases and is represented in Figure 4.7. The TDC also has an initial flip flop to be able to measure the circuits offset, and a final inverter to improve the linearity by maintaining the same charge applied at the end of all inverters.

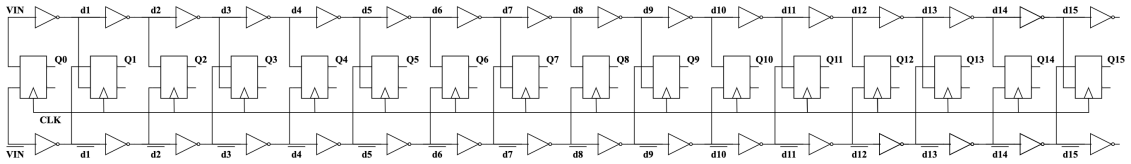


Figure 4.7: 4 bit TDC.

The circuit has 3 options of possible combination of components sizes. Size 1 where all the components are the smallest size, size 2 where the inverter W_n will be bigger than the SRFFs W_n, and size 3 where all the components are the biggest size. The SRFFs W_n can not be higher than the inverters W_n because the increase in the capacitance load will increase the delay. These sizes are presented in table 4.1, their calculations were defined in the previous sections. It is important to note that all the CMOS will have 2 fingers.

Table 4.1: TDC dimensions

	Size 1	**Size 2**	**Size 3**
L (nm)	120	120	120
Inverter			
W_n (μ m)	6.00	12.0	12.0
W_p (μm)	15.9	31.8	31.8
Area (μm^2)	0.72	1.44	1.44
Flip Flop			
$W_{M2/M6}$ (μ m)	6.00	6.00	12.0
$W_{M3/M4/M7/M8}$ (μ m)	12.0	12.0	24.0
$W_{M1/M5}$ (μ m)	9.00	9.00	18.0
Area (μ m^2)	9.36	9.36	18.7

In this work the maximum delay of the inverter is 20 ps, as was mentioned in Table 1.1, so Tp_{min} is 40 ps. As the chosen sizes have a delay close to 12 ps with two inverters in a row, their F_{max} is 42 GHz. As the working frequency of the circuit is 100 MHz, Table 1.1, the F_{max} does not limit the work of the circuit.

5

Circuit Simulations and Results

The simulation results for the previously chosen TDC will be presented and analysed in this chapter. Some simulation results have specific targets, these targets are in Table 1.1.

All the simulations were done in the test bench in Figure D.3 in appendix D, and the results analysed with cadence and Matlab.

5.1 TDC Simulation

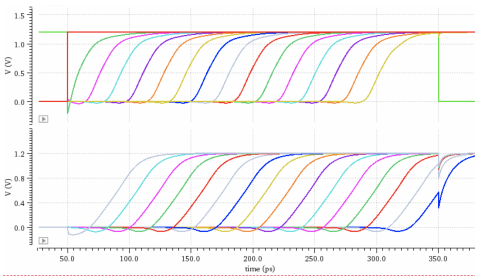
As the ideal size for the circuits components is yet to be defined the first step is to compare the three possible circuits. Doing the transient analysis of the circuits it was possible to obtain their transient responses, Figure 5.1. The delay response values where taken alternating delay lines.

In the figure it's possible to observe that the Size 2 TDC has a smaller delay than the other two. This happens because of the low capacitance load added from the flip flop. The offset of the circuits was measured with the output Q0, from the first SRFF, and is 28 ps for the first two sizes and 27 ps for Size 3.

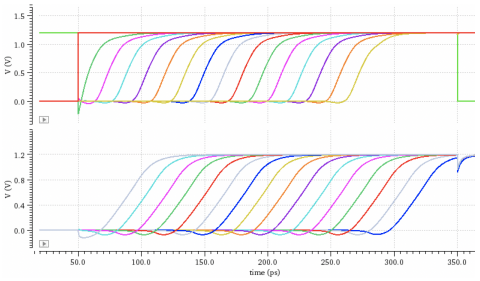
The data from Figure 5.1, processed with F.3 in appendix F, produced Figure 5.2. As can be observed along the delay line there's a rise and fall of the delay of the inverters, this happens because of the alternate use of either the PMOS or the NMOS transistors of the inverter, this was predicted previously and it's the reason the TDC has two delay lines. The first rise happens, because the first inverter is not affected by the charges of the previous inverters.

Then a Periodic Steady-State (PSS) and a Periodic Noise (PNOISE) analysis were performed, to obtain the average current, the average power consumed and the jitter noise. The specter analysis can be observed in Figure 5.3.

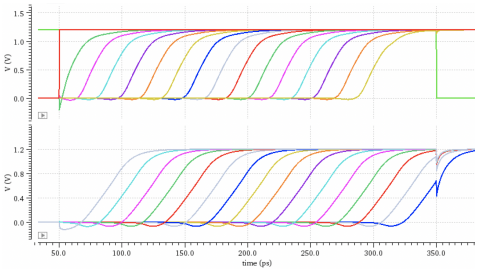
As can be seen in the Figures the the response is as expected, at 100 MHz is the maximum value, at 300 MHz a 1/3 of the maximum value, at 500 MHz a 1/5 of the maximum, and so on. The average current can also be obtained with a PSS analysis, by selecting the spectral analysis of the current in V_{DD}. The current is 712.5 μA in Size 1,



(a) Transient response Size 1



(b) Transient response Size 2



(c) Transient response Size 3

Figure 5.1: TDC Transient analysis

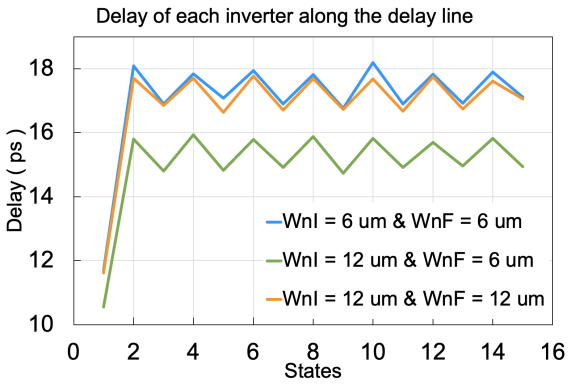


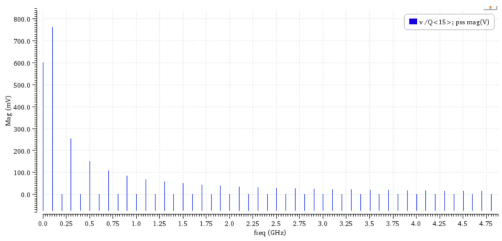
Figure 5.2: Delay of each inverter along the Delay line size comparison

926.9 μA in Size 2 and 1.429 mA in Size 3. The increase in current results in a increase in power, so the increase in size results in a increase in power, as was expected.

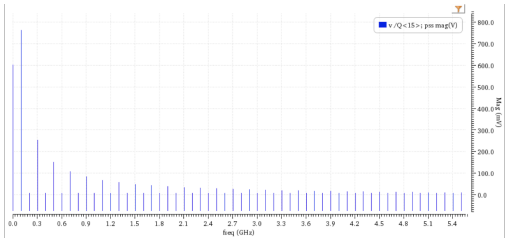
For the noise analysis, the simulation and formulas can be found in section E.2 in appendix E. The noise is measured on the last SRFF output, because it's where it's going to be the highest in the circuit. The jitter for the Size 1 is 23.9 fs, for Size 2 22.92 fsand for Size 3 16.76 fs. For all the sizes approximately 70% of the noise comes from the M1 PMOS on the last SRFF. As expected the jitter decreases with the increase in size.

5.1.1 TDC linearity

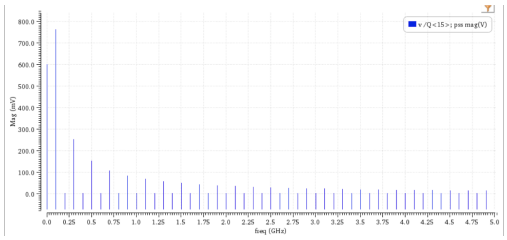
To obtain the Transfer function of the TDC and study its linearity, a transient parametric analysis was necessary, the delay of the clock in relation to V_{in} was varied from 0 to 300 ps with a 1 ps step. The transfer functions can be observed in Figure 5.4. Size 1



(a) PSS response Size 1



(b) PSS response Size 2



(c) PSS response Size 3

Figure 5.3: TDC PSS analysis

has a resolution of 17.4 ps and a range of 261 ps, size 2 of 15.3 ps and 229 ps, and size 3 of 17.13 ps and 257 ps.

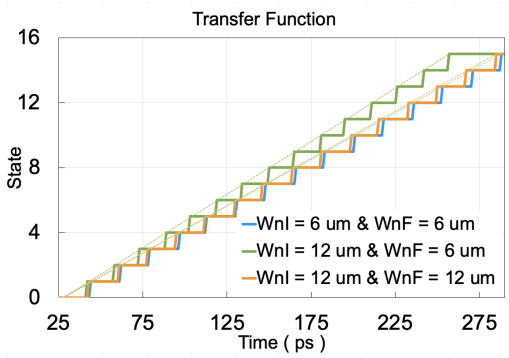
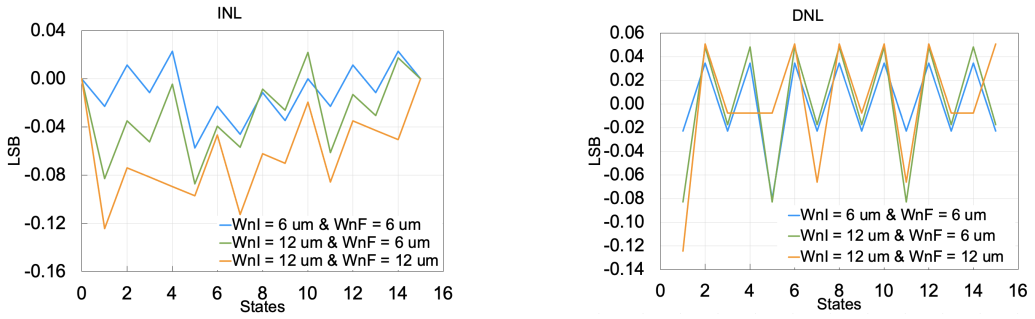


Figure 5.4: Transfer Function size comparison

The Integral Non Linearity (INL) is the deviation of the output from 1 Least Significant Bit (LSB), and the Differential Non Linearity (DNL) is the difference between two successive threshold points form 1 LSB. The INL and DNL error are, respectively, 0.08 LSB and 0.11 LSB for size 1, 0.13 LSB and 0.11 LSB for size 2, and 0.12 LSB and 0.18 LSB for size 3. All of the different size circuits have errors lower than 20% of the LSB, therefore meet the objectives in Table 1.1.

5.1.2 Final TDC size

As the linearity is good in all the circuits, the choice of the finals circuits size falls on the jitter noise, the circuits power and the resolution. The jitter noise increases with the increase in the circuits size while the power decreases. Therefore considering the middle value of these parameters and the lowest resolution the ideal circuit is the Size 2 circuit.



(a) INL size comparison

(b) DNL size comparison

Figure 5.5: Linearity size comparison

Adding 10 fF parasitic capacities in all the nodes the jitter noise and the power consumption are approximately the same, but the resolution increases to 16.6 ps and the INL and DNL decrease. So the circuit linearity gets better with the parasitic capacities.

A test of the maximum frequency and minimum voltage supply, was also done to determine the limits of the circuit. The maximum working frequency of the circuit is 5.5 GHz and the minimum voltage supply is 0.44 V.

The circuits area is the same with or without parasite capacities, 0.143 mm^2 was was obtained through the layout in section D.2 in appendix D, this layout is not optimized and is was done simply to obtain a rough area approximation.

As the circuit has all the states the Effective number of bits (ENOB) is 4 and the f_s is the inverse of the range. So the Figure-of-Merit (FOM) is 13.9 fJ/conv for the Final TDC, and 17.3 fJ/conv when parasite capacities are considered.

The chosen circuit parameters are in Table 5.1. As can be observed in the table all the parameters values meet the objectives set for the circuit.

Table 5.1: Final TDC parameters

Parameters	Final TDC	Final TDC w/ parasites	Objectives
Wn Inverter (μm)	12.0	-	
Wn Flip Flop (μm)	6.00	-	
Resolution (ps)	15.3	16.6	< 20.0
Offset (ps)	28.0	34.0	
Range (ps)	229.0	249.0	
Current (μA)	926.9	-	
Power (mW)	1.11	-	
Jitter (fs)	22.9	-	< 100
INL error (LSB)	0.13	0.07	< 0.25
DNL error (LSB)	0.11	0.06	< 0.25
Area (mm^2)	0.143	-	
FOM (fJ/conv)	13.9	17.3	

5.1.3 TDC performance with process corners

For the process corner simulation all the corners, in Table E.1, were simulated. After checking that the circuit was working in all the simulations, the worst two corners were analysed. This corners were the Slow corner at 120 ºC with V_{DD}, V_{in} and the CLK at 1.08 V, and the Fast corner at -40 ºC with V_{DD}, V_{in} and the CLK at 1.32 V.

In Figure 5.6 it's possible to observe the delay of each inverter along the delay line, of the Normal, the Fast and Slow corner. The results are as expected considering the transistor speed.

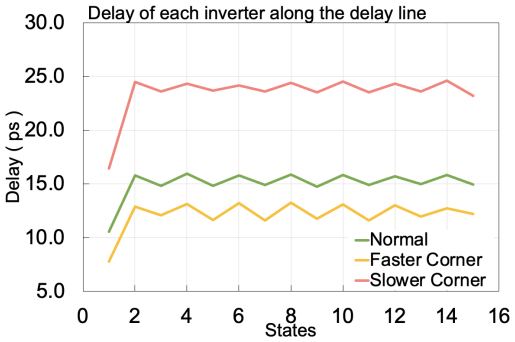


Figure 5.6: Delay of each inverter along the Delay line Corners comparison

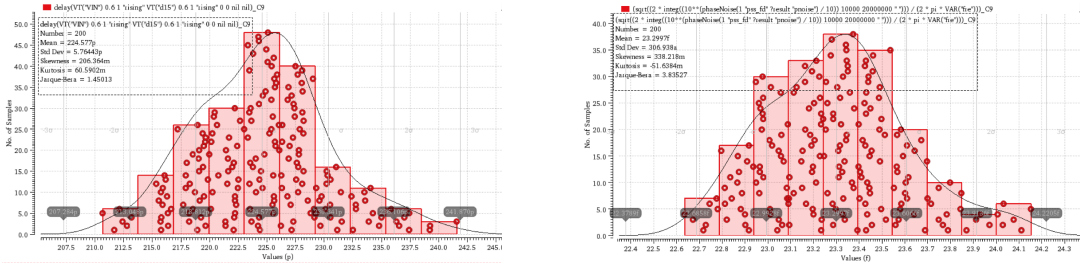
As ca be observed in the Table 5.2 the resolution of the Slow corner is 23.9 ps and of the Fast corner is 11.07 ps. The resolution is better for the Fast Fast corner, as expected, and the jitter is lower, however the there's an increase in power consumption and the linearity is worst. Even with the worst linearity it still meets the objectives. The resolution of the Slow Slow corner is the only parameter that does not meet the objectives.

Table 5.2: Final TDC Corners parameters

Parameters	Final TDC	Faster Corner	Slower Corner	Objectives
Resolution (ps)	15.3	11.1	23.9	< 20.0
Offset (ps)	28.0	18.0	50.0	
Range (ps)	229.0	166.0	359.0	
Current (μA)	926.9	1049.7	827.7	
Power (mW)	1.11	1.39	0.89	
Jitter (fs)	22.9	17.4	36.7	< 100
INL error (LSB)	0.13	0.13	0.08	< 0.25
DNL error (LSB)	0.11	0.18	0.13	< 0.25

5.1.4 TDC Monte Carlo simulation

For the Monte Carlo Simulation the simulation described in E.4 in Appendix E was used. As can be observed by Figure 5.7 the both simulations have a close to normal



(a) Full Delay line delay Variation

(b) Circuit Jitter variation

Figure 5.7: Monte Carlo Delay and Jitter simulation

distribution and the delay has a standard deviation of 5.8 ps and the jitter of 0.31 fs. The delay in Figure 5.7a refers to the time the signal takes to go through the fifteen inverters in the delay line. So as there is little result variation the circuit can function with process variations.

5.1.5 TDC performance with LDO

When using a LDO the tension source that feeds the circuit varies. This variation can be observed in Figure 5.8.

As was observed in the corners the circuit is slower with a lower supply voltage. The circuit with LDO has a resolution of 16.3 ps this is 1 ps slower than the circuit with an ideal source. The linearity and jitter also worsen, as is expected. The INL error goes from 0.13 LSB to 0.24 LSB, the DNL error from 0.11 LSB to 0.25 LSB, and the jitter from 22.9 fs to 26.6 fs. Even considering this differences the circuit still meets the requirements.

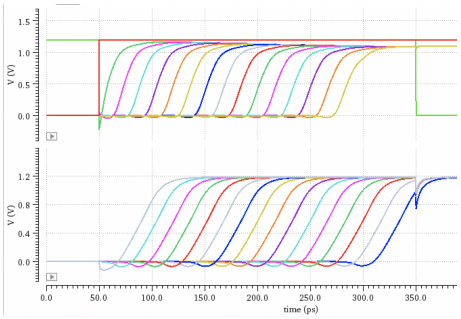


Figure 5.8: TDC with LDO Transient response

5.1.6 TDC Simulation with Dummy

In an attempt to increase the linearity, even though it already meets the objective, a dummy inverter was inserted between the input signal and the first inverter of both the delay lines, and inverting the V_{in} and $\overline{V_{in}}$. The new circuit is in Figure 5.9.

The objective of the dummy inverters is to avoid the rise in the delay of the first inverter of the delay line. The disadvantage of the TDC with dummies is the increase in

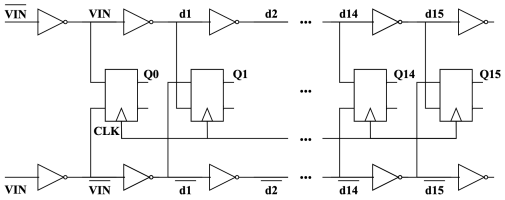


Figure 5.9: 4 Bit TDC with dummy

the power consumption and in the circuits offset.

As can be observed in Figure 5.10, instead of a rise in delay there's a smaller fall.

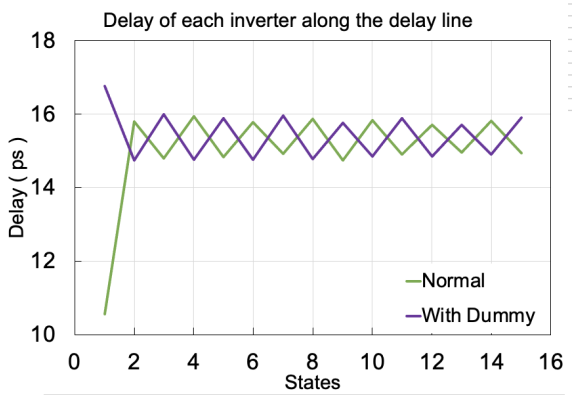


Figure 5.10: Delay of each inverter along the Delay line Dummy comparison

The comparison of parameters can be found in Table 5.3. The Resolution is higher with Dummy than without, and the linearity does not improve. Another thing to note is that the maximum working frequency is 1.7 GHz, lower than previously and the minimum voltage supply is 0.5 V, highers than previously. So the added inverters do not increase the circuits linearities, and as they increase the power consumption and area they are not a solution to increase the linearity.

Table 5.3: Final TDC Dummy parameters

Parameters	Final TDC	TDC with Dummy	Objectives
Resolution (ps)	15.3	15.4	< 20.0
Offset (ps)	28.0	42.00	
Range (ps)	229.0	231.0	
Current (μA)	926.9	958.4	
Power (mW)	1.11	1.26	
Jitter (fs)	22.9	22.9	< 100
INL error (LSB)	0.13	0.13	< 0.25
DNL error (LSB)	0.11	0.12	< 0.25

5.2 TDC Performance Summary

In section 5.1 the performance of the inverter based TDC was evaluated. Considering parasitic capacities and process corners. The circuits linearity was also tested with the extraction of the INL and DNL, from the transfer function, section 5.1.1.

The linearity analysis resulted in a INL and DNL error of 0.13 LSB and 0.11 LSB for the TT corner, 0.13 LSB and 0.18 LSB for the FF corner, and 0.08 LSB and 0.13 LSB for the SS corner. The corner with the worst linearity is the FF corner that has a temperature of -40 ºC and a supply voltage of 1.32 V. Even though the SS is the worst corner it's still within the objectives.

The circuits resolution is 15.3 ps and can rise to 23.9 ps in the SS corner, and decrease to 11.1 ps in the FF corner.

The core area of the circuit is 307 μ m^2 and the power consumption is 1.11 mW. The so the circuit has a low area and power consumption.

The FOM for the circuit is also good being 13.9 fJ/conv when the parasitic capacities are not considered and 17.3 fJ/conv when considering the parasitic capacities. It has a sampling frequency of 4.4 GHz, this being the reason has to why the maximum work frequency is 5.5 GHz, as the circuit needs at least 4.4 GHz to do a complete measure plus the offset. The circuits ENOB is 4 as there are no missing transition states.

When a dummy was inserted to try to increase the linearity, the DNL error increased and the INL error maintained the value. So as a solution to increase the linearity the dummy insertion is not a solution.

The Table 5.4 compares the designed TDC with other state-of-the art TDCs. The other TDCs in the table use the same technology, but use different techniques. The difference in techniques can be observed in the lower resolution and the increase in number of bits results in the increase of the FOM. Another difference is the area in comparison with the number of bits, the designed circuit has a big area compared to the others, because of the technique.

Table 5.4: TDCs performance comparison

Reference	**This work**	**[11]**	**[12]**	**[13]**	**[14]**
Technique	Inverter Based	Pulse Shrinking	Two Step PS	3D Vernier	Cyclic
Tech (nm)	130	130	180	130	130
Resolution (ps)	15.3	0.82	2.00	6.98	1.25
Nº of Bits	4.00	11.7	16.0	11.0	8.00
Power (mW)	1.11	7.50	18.0	0.33	4.30
Jitter (fs)	22.9	-	-	-	-
INL error (LSB)	0.13	4.51	4.20	1.50	4.00
DNL error (LSB)	0.11	1.78	1.50	0.80	1.40
Area (mm^2)	0.143	0.147	0.080	0.280	0.070
FOM (fJ/conv)	13.9	91.0	430	400	460

6

Conclusion and Future Work

6.1 Conclusion

The objective of this thesis was to design a low noise time to digital converter to use as a phase detector in ADPLL. Considering this objective, and after researching the literature, the circuits topology was chosen. The circuit was developed with CMOS 130 nm technology, and the main objective of this work was to make sure the TDC met the specifications.

Since TDCs are digital circuits, the most important parameters are going to improve with the technology scaling, parameters like power consumption, speed and silicon area.

The developed TDC is a simple modular structure, that can be easily adapted, by adding or replacing blocks, to a new functionality. With this in mind one of the simplest circuit architectures was chosen, the inverter based TDC, with technology resolution. This technique is an upgrade to the buffer based TDC as it can achieve double of its resolution. So two 15 inverter delay-lines were used to achieve a 4 bit TDC, with a 15.3 ps resolution. To sample the signal and discover where in the delay line it stops, 16 CMOS Set and reset Flip Flops where used, being the first SRFF used to measure offset. At last to facilitate the measurements, as the SRFFs outputs come in thermometer code, a 4 bit Thermometer to binary decoder was connected to the SRFFs outputs. This decoder is a ideal block made with ideal logic gates that uses a logic-based architecture. The core area of the TDC is 307 μm^2.

With the logic-based decoder the TDC has a 4 bit architecture, this results in 16 possible different codes. As mentioned in section 5.1.2 the TDC has an ENOB of 4 bits so there are no missing codes.

In section 5.1.3 the INL and DNL where analysed. The INL error in the TT corner is 0.13 LSB, in the SS corner, at 120 ºC and 1.08 V supply voltage, is 0.08, and in the FF corner, at -40 ºC with 1.32 V supply voltage, is 0.13, being the TT and FF corners the ones with the worst result. The DNL error in the TT corner is 0.11 LSB, in the SS corner is 0.13, and in the FF corner is 0.18, being the FF corner the worst result. So the worst corner in relation to linearity is the FF corner.

In relation to the resolution the TT corner has one of 15.3 ps, the SS and one of 23.9 ps, and the FF corner has one of 11.1 ps. So the worst corner is the SS, and is the only case parameter that does not meet the objectives, but it was tested with military grade parameters when used in other circumstances this will not be an issue.

The RMS Jitter is 22.9 fs for the TT corner, 36.7 fs for the SS corner and 17.4 fs for the FF corner. For this parameter the worst corner is the SS corner. However even being the worst value it still meets the objectives.

The circuit was tested with Monte Carlo simulations and has very little variation in the circuits parameters, as can be observed in section E.4.

When substituting the ideal voltage source with a LDO the parameters got worse, but still met the objectives.

To follow the expected values the worst case of Jitter noise will have the best power consumption and the best RMS Jitter will have the worst power consumption. So the corner with the worst power consumption is the SS corner, with 0.51 mW.

In an attempt to increase the circuits linearity, an inverter was inserted between the input signal and the first inverter of the delay line. This was done to try and correct the initial rise in delay. However the linearity didn't change significantly, and the area, the power consumption and the offset increased, so the addition of a dummy is not a solution to increase the linearity.

6.2 Future Work

To improve this project the most simple solutions would be to increase the number of bits of the TDC, and use smaller technology. The higher number of bits would make it so that the circuit could start to compete to the state of the art, and the smaller technology would guarantee better results is area and power consumption.

Maintaining the chosen topology, its parameters could be improved by changing the basic blocks architecture. There are multiple more complex architectures of inverters and Flip Flops that could be used to improve the circuit. However most of them would increase the power and area consumption.

Another improvement that could be done to the circuit is to transform the delay lines, to make the TDC cyclic, this way even when using the same technology, and most of the same blocks, the resolution could be increased substantially. The disadvantage of this technique is that it would require the design of more control blocks.

Bibliography

[1] P. V. Brennan. *Phase-Locked Loops: Principles and Practice*. McGraw-Hill Professional Publishing, 1996. ISBN: 9780070075689 (cit. on p. 1).

[2] S. Henzler. *Time-to-Digital Converters*. 1st. Springer Publishing Company, Incorporated, 2010. ISBN: 9048186277. DOI: 10.1007/978-90-481-8628-0 (cit. on pp. 1, 7–9, 15, 16, 18).

[3] S. Cadeddu et al. "A Time-to-Digital Converter Based on a Digitally Controlled Oscillator". In: *IEEE Transactions on Nuclear Science* 64.8 (2017), pp. 2441–2448. DOI: 10.1109/TNS.2017.2726822 (cit. on pp. 1, 7).

[4] A. Godave, P. Choudhari, and A. Jadhav. "Comparison and Simulation of Analog and Digital Phase Locked Loop". In: *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. 2018, pp. 1–4. DOI: 10.1109/ICCCNT.2018.8494198 (cit. on pp. 3, 4).

[5] L. A. B. G. de. "ANALYSIS AND DESIGN OF QUADRATURE OSCILLATORS". eng. PhD thesis. Jan. 2007 (cit. on p. 5).

[6] B. Razavi. *Design of CMOS Phase-Locked Loops: From Circuit Level to Architecture Level*. Cambridge University Press, 2020. DOI: 10.1017/9781108626200 (cit. on pp. 5, 6).

[7] A. Abidi. "Phase Noise and Jitter in CMOS Ring Oscillators". In: *IEEE Journal of Solid-State Circuits* 41.8 (2006), pp. 1803–1816. DOI: 10.1109/JSSC.2006.876206 (cit. on p. 6).

[8] C. Liu and J. McNeill. "Jitter in oscillators with 1/f noise sources". In: *2004 IEEE International Symposium on Circuits and Systems (ISCAS)*. Vol. 1. 2004, pp. I–773. DOI: 10.1109/ISCAS.2004.1328309 (cit. on p. 6).

[9] A. S. Sedra and K. C. Smith. *Microelectronic Circuits*. 7th. Oxford University Press, 2015. ISBN: 9780190649777 (cit. on pp. 10, 12, 13, 22).

[10] G. Madhumati, K. R. Rao, and M. Madhavilatha. "Comparison of 5-bit Thermometer-to-Binary Decoders in 1.8V, 0.18µm CMOS Technology for Flash ADCs". In: *2009 International Conference on Signal Processing Systems*. 2009, pp. 516–520. DOI: 10.1109/ICSPS.2009.160 (cit. on p. 19).

[11] R. Granja et al. "11.7b Time-To-Digital Converter with 0.82ps resolution in 130nm CMOS Technology". In: *2018 14th Conference on Ph.D. Research in Microelectronics and Electronics (PRIME)*. 2018, pp. 29–32. DOI: 10.1109/PRIME.2018.8430374 (cit. on p. 34).

[12] R. Enomoto et al. "A 16-bit 2.0-ps Resolution Two-Step TDC in 0.18- μ m CMOS Utilizing Pulse-Shrinking Fine Stage With Built-In Coarse Gain Calibration". In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 27.1 (2019), pp. 11–19. DOI: 10.1109/TVLSI.2018.2867505 (cit. on p. 34).

[13] Y. Kim and T. W. Kim. "An 11 b 7 ps Resolution Two-Step Time-to-Digital Converter With 3-D Vernier Space". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 61.8 (2014), pp. 2326–2336. DOI: 10.1109/TCSI.2014.2304656 (cit. on p. 34).

[14] Y.-H. Seo et al. "A 1.25 ps Resolution 8b Cyclic TDC in 0.13 μm CMOS". In: *IEEE Journal of Solid-State Circuits* 47.3 (2012), pp. 736–743. DOI: 10.1109/JSSC.2011.2176609 (cit. on p. 34).

A

Inverter Internal Capacitors

$$Cg_{N/P} = W_{N/P} \cdot (2CGSDO + L_{N/P} \cdot CGOX_{N/P}) \tag{A.1}$$

$$Cgd_{N/P} = W_{N/P} \cdot CGSDO \tag{A.2}$$

$$Cdb_{N/P} = (W_{N/P} + Sd_s) \cdot 2CJSW_{N/P} + W_{N/P} \cdot Sd_s \cdot CJ_{N/P} \tag{A.3}$$

$$CL = 2Cgd_1 + 2Cgd_2 + Cdb_1 + Cdb_2 + Cg_3 + Cg_4 + Cw \tag{A.4}$$

B

TDC SUMMARY

Table B.1: Performance summary of Buffer and Inverter Based delay-line TDCs

	Buffer Based Delay-line TDC	**Inverter Based Delay-line TDC**
Principle	Start signal propagates along a buffer delay-line. When the stop signal arrives the state is sampled by the flip-flops.	A differential start signal propagates along two inverter delay-lines. When the stop signal arrives the differential state is sampled by the flip-flops.
Resolution	$2 \cdot t_{inv}$	t_{inv}
Core Area	N $(2A^{inv} + A^{FF})$	$N(2A^{inv} + A^{FF})$
Loop Structure	possible	possible
Advantages	• Simple Structure • Low Power • Easy Control	• Simple Structure • Low Power • Technology Resolution
Disadvantages	• Low Resolution, two time the Technology resolution • Big offset	• Alignment of delay-lines • Resolution limited by technology

Table B.2: Performance summary of Parallel Scaled and Vernier TDCs

	Parallel Scaled TDC	**Vernier TDC**
Principle	The start signal reaches the parallel delay elements at the same time, each element with increasing propagation delays. When the stop signal arrives the state is sampled by the flip-flops.	Start and stop signals propagate along two delay-lines with slightly different delays.
Resolution	Tuned by transistor and capacitors	$t_{d1} - t_{d2}$
Core Area	$N[A^{inv} + A^{FF} + \frac{1}{2}(N+1)A^{cap}]$	$N(4A^{inv} + A^{FF})$
Loop Structure	not possible	possible
Advantages	• Sub-gate delay resolution • Good Linearity • Conversion time independent from resolution • Simple Structure	• Sub-gate delay resolution • Modular Structure • Good Linearity • High Dynamic Range
Disadvantages	• Susceptible to variation • Big offset • Not good for high dynamic range • Careful layout design	• Two long delay lines • Conversion time dependent on resolution • High Area and Power consumption • Difficult control and layout

C

Thermometer to Binary Architectures

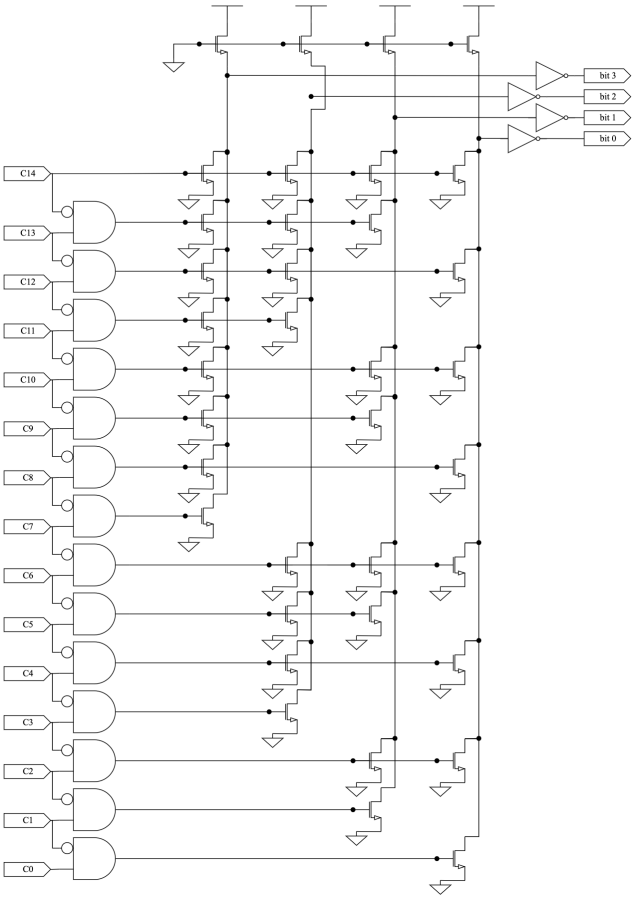


Figure C.1: Binary ROM based Decoder

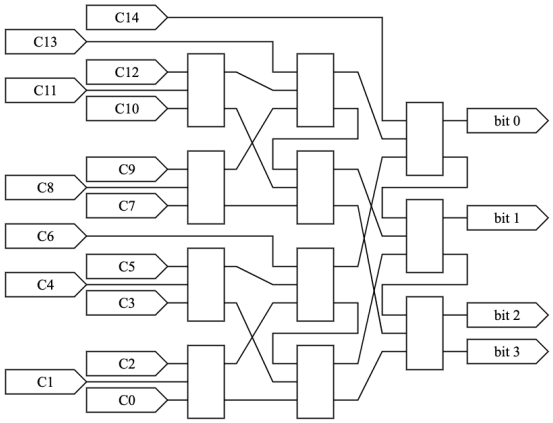


Figure C.2: 4 bit Wallace-Tree Decoder

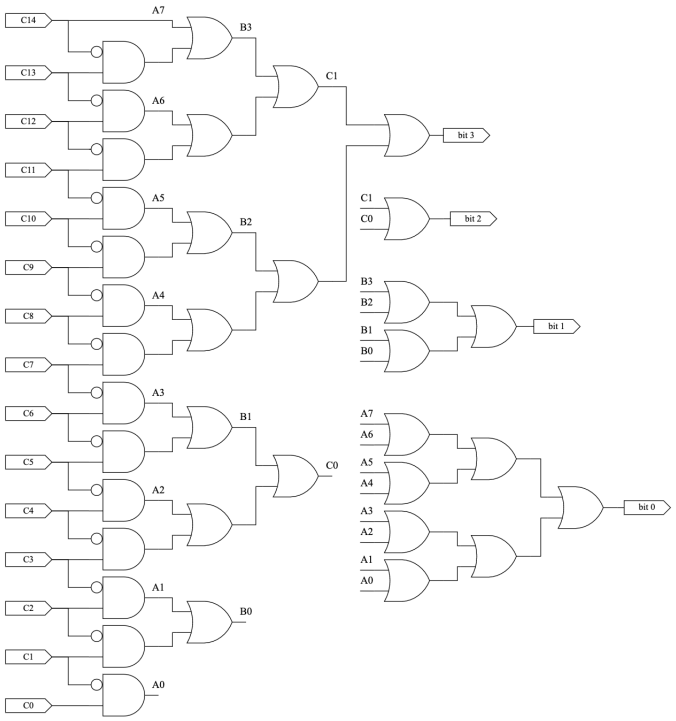


Figure C.3: Fat-Tree Decoder

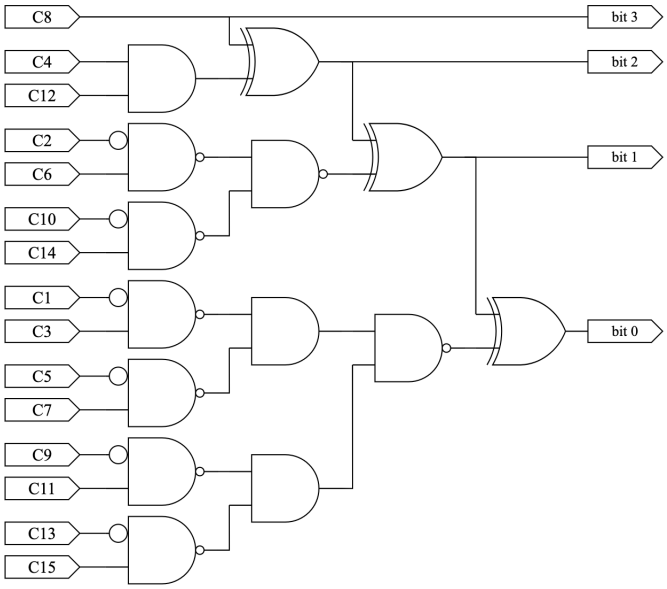


Figure C.4: Logic-based Decoder

D

Test Benches and Layout

D.1 Test Benches

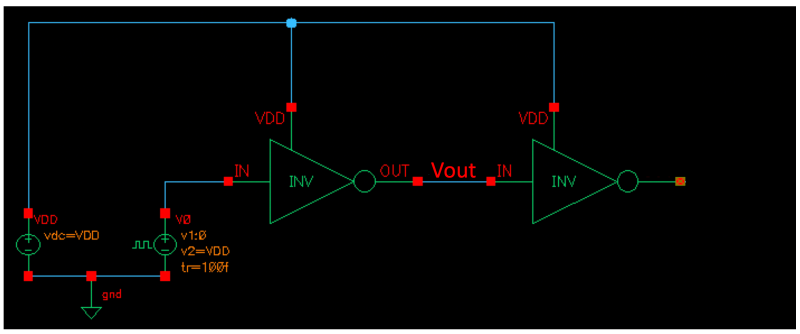


Figure D.1: Inverter Test Bench

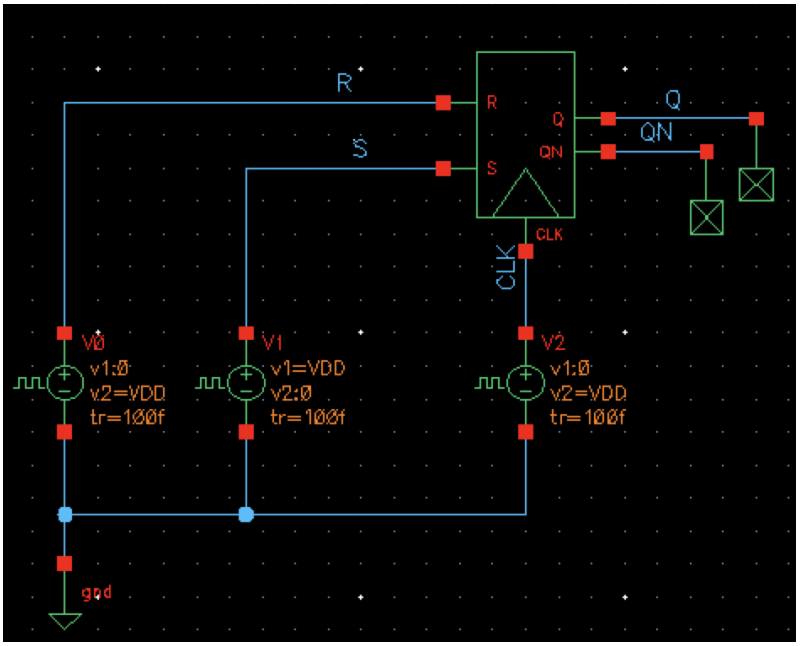


Figure D.2: Flip Flop Test Bench

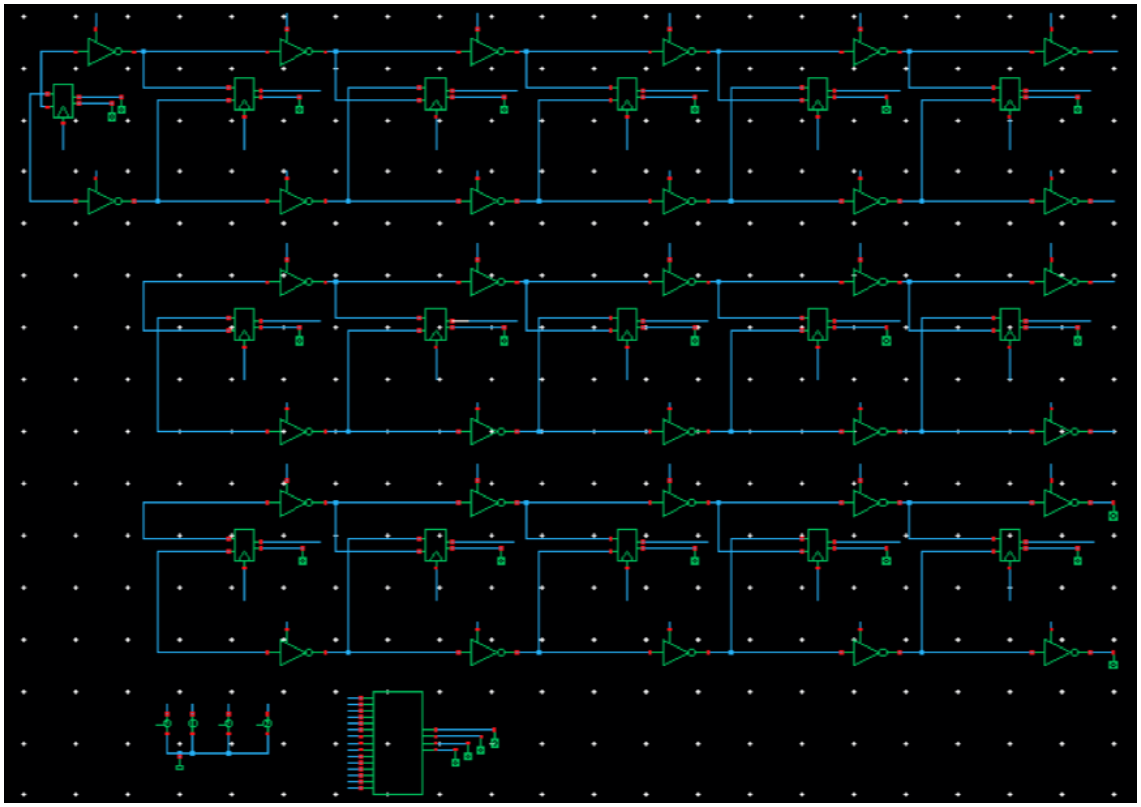


Figure D.3: TDC Test Bench

D.2 Layout

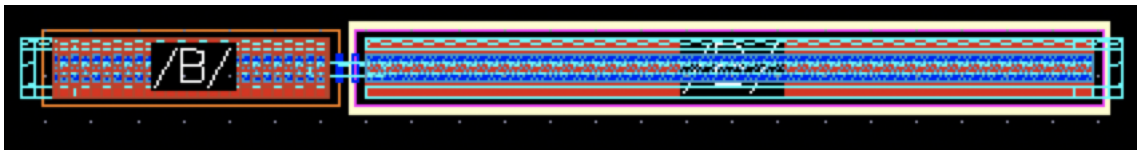


Figure D.4: Basic Connection Inverter Layout

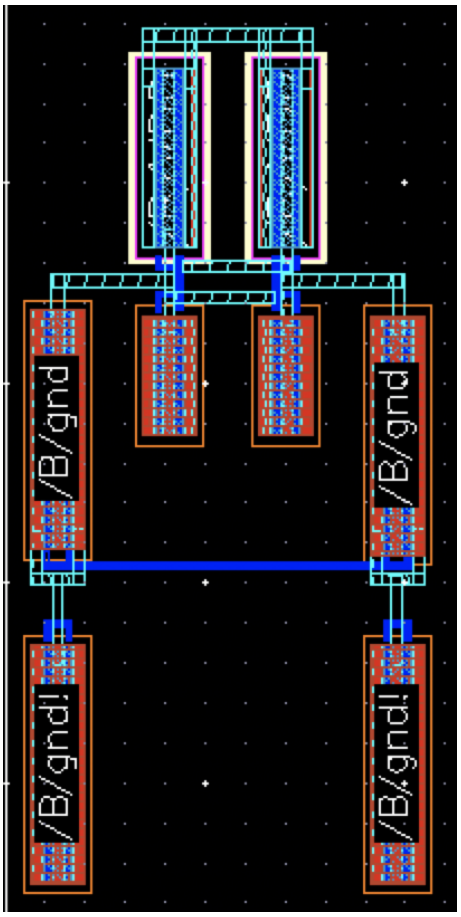


Figure D.5: Basic Connection Flip Flop Layout

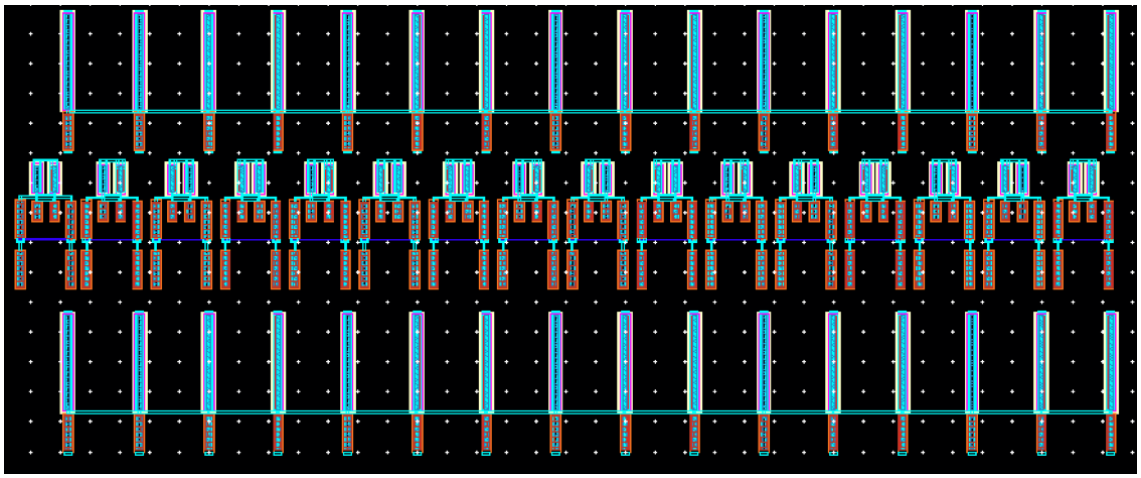


Figure D.6: Basic Connection TDC Layout

E

Simulation

E.1 Analysis Methods

E.1.1 Process Corners

The process corners study the influence of Process Voltage and Temperature (PVT) variations in the circuits performance. The main 4 corners are represented on Figure E.1. For the analysis all corners were simulated to confirm that the circuit worked on all. The linearity analysis was then done only on the extreme case corners. The Table E.1 shows all the tested process corners.

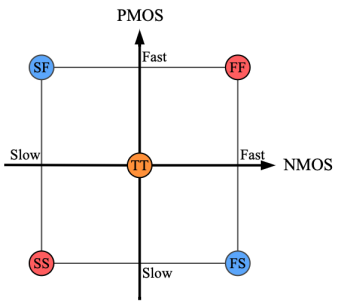


Figure E.1: Process Corners

E.1.2 INL and DNL

For the linearity analysis the INL and DNL were calculated. The ideal time at which each bit happens is called the input time, T_{in}, and is given by the following equation,

$$T_{in} = T_{LSB} \cdot N \tag{E.1}$$

where T_{LSB} is the resolution. T_{LSB} can be calculated by an average of all the TDCs steps. It's important to note that when the circuit has a significant expected offset that the offset value should be subtracted from the values from the transfer function, or added to the ideal input time before comparisons are made.

Table E.1: Tested Corners

Corner	Temperature (°C)	VDD (V)	Vin (V)	Clk (V)
TT	50	I	I	I
SS	-40	I	I	I
	120	I	I	I
SF	-40	I	I	I
	120	I	I	I
FF	-40	I	I	I
	120	I	I	I
FS	-40	I	I	I
	120	I	I	I

I = {1.08, 1.20 & 1.32}

The INL and DNL results can be obtained by applying the following equations to the transfer function. Considering that N follows the following rule, $0 < N < 2^N - 1$.

$$INL_i = \frac{T_{out_i} - T_{out_1}}{T_{LSB}} - N_i \tag{E.2}$$

$$DNL_i = \frac{T_{out_{i+1}} - T_{out_i}}{T_{LSB}} - 1 \tag{E.3}$$

The error for the INL and DNL are the minimum value subtracted from the maximum value.

E.1.3 Figure-of-Merit

The FOM is used to compare the overall performance of Time-to-Digital Converters by evaluating the amount of the power consumption per conversion step. This can be obtained with the followings equation, where P is the power consumption, ENOB is the effective number of bits and fs is the sampling frequency. The unit of the figure of merit is J/conv.

$$FOM = \frac{P}{2^{ENOB} \cdot f_s} \tag{E.4}$$

E.1.3.1 Effective Number of Bits

To obtain the FOM the ENOB needs to be calculated. The proposed TDC outputs a 4 bit binary word. This means that there should be $2^4 = 16$ different codes. The circuits ENOB is,

$$ENOB = log_2(states) \tag{E.5}$$

E.1.3.2 Sampling Frequency

The f_s is the time needed to perform each measurement. So considering the longest conversion time or the range, the maximum time is measured.

$$f_s = \frac{1}{Range} \tag{E.6}$$

E.2 PSS and Pnoise simulation

The first step to run a PNOISE analysis is to first run a PSS analyses. To run the PSS analyses the beat frequency needs to be correct so even if auto calculate is selected it's always better to confirm that the frequency is correct as well as the fundamental tones, Figure E.2.

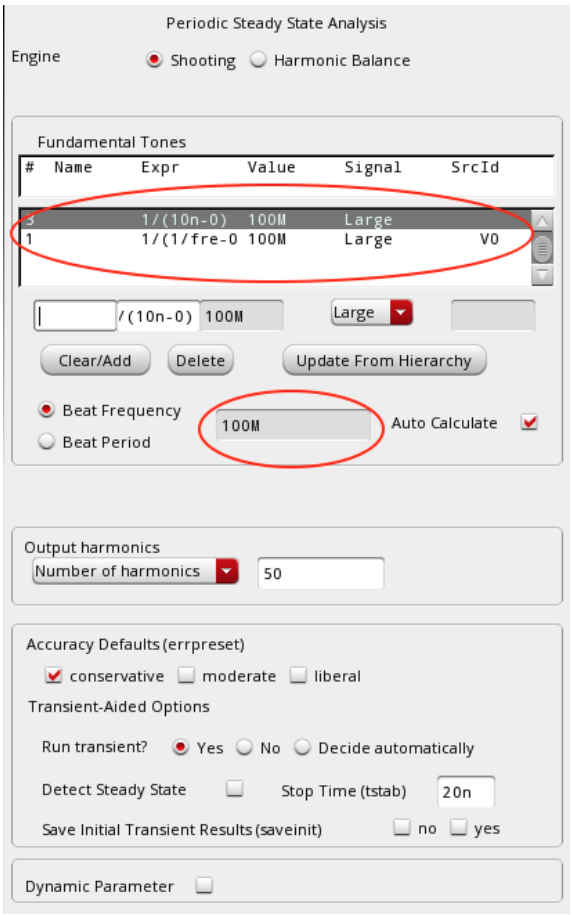
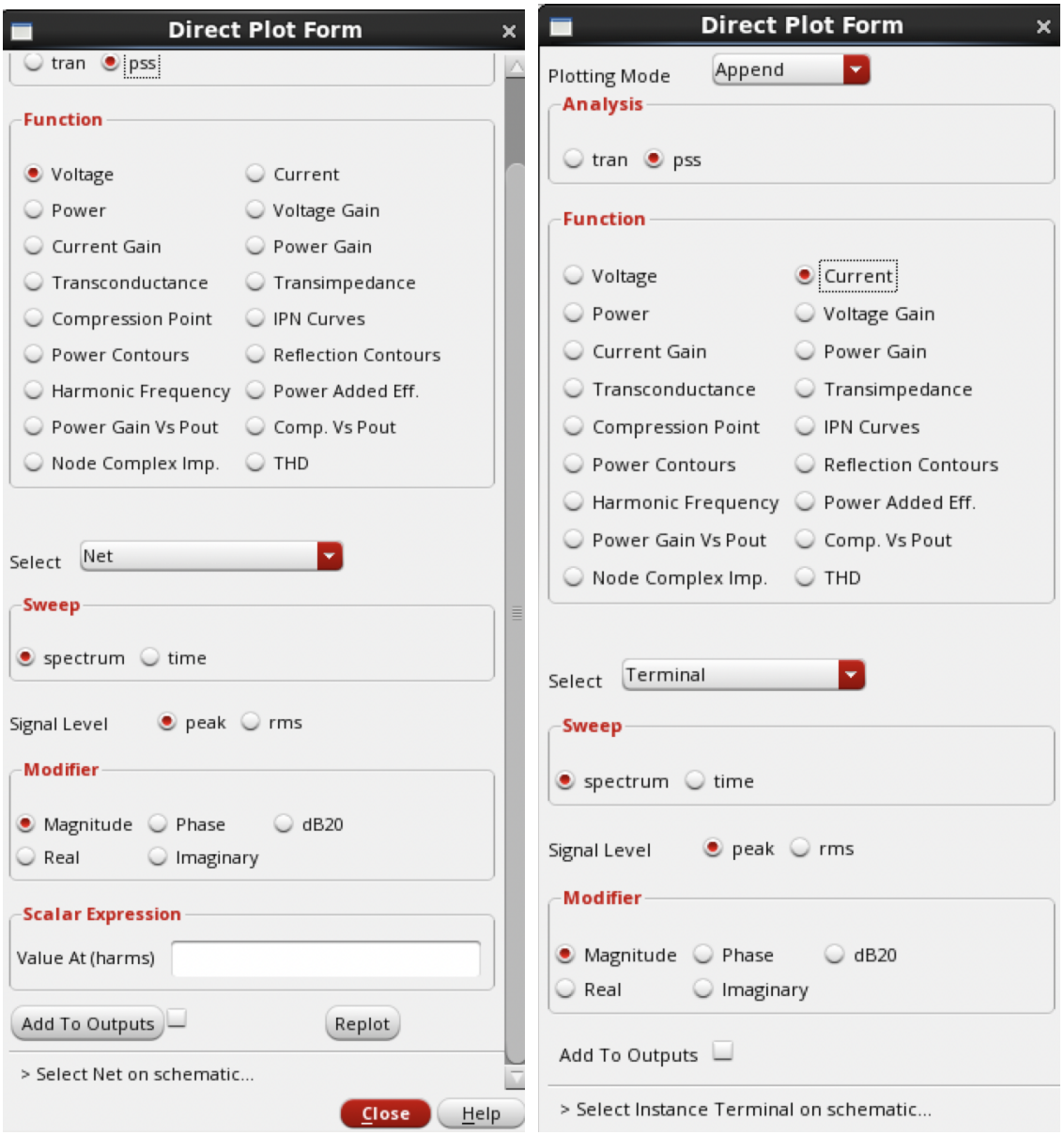


Figure E.2: PSS analyses definition

To obtain the spectral analysis for the circuit the PSS main form needs to be done. To obtain the spectral analysis of the circuits output the main form uses the voltage function, sweeping in spectrum and using magnitude as a modifier, Figure E.3a. To obtain the average current the main form uses the current function, selecting the voltage supply, sweeping in spectrum and using magnitude as a modifier, Figure E.3b. It is important



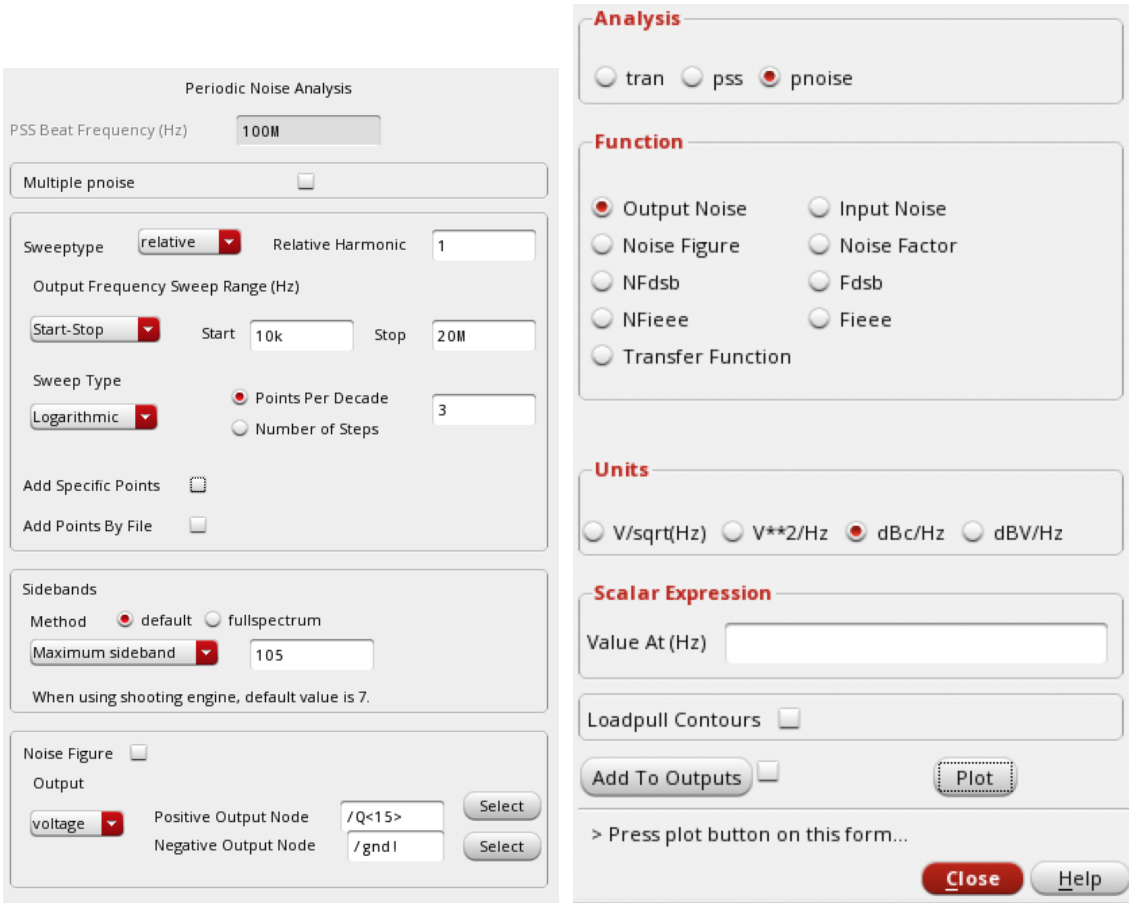
(a) PSS response Size 1 (b) PSS response Size 2

Figure E.3: PSS Main Form definition

not to select the voltage supply instance but its output pin. The first peak will give the circuits average current and the second the maximum current.

As for the PNOISE analyses, is done in relation to the first harmonic. The output Frequency Sweep Range in this work is from 10 kHz to 20 kHz. The Sweep Type is Logarithmic and it doesn't need a lot of points so three is sufficient. As for the Side bands, they use the default method, however do not use the default value. Their value is calculated using the following formula.

$$MaxSB = \frac{0.35}{tr \cdot freq} \tag{E.7}$$



(a) PNOISE analysis definition (b) Pnoise Main Form definition

Figure E.4: PNOISE definition

Where tr is the maximum rise delay time of the circuit and freq is the circuits frequency, preferably the one used in the PSS analyses. At last the positive output node is the node to measure and the negative is the ground. This can be observed in Figure E.4a.

The output noise graph can be obtained with the PNOISE main form in dBc/Hz and pressing the plot button, Figure E.4b.

To obtain the jitter value the calculator is used with the equation 2.2, and can be observed in Figure E.5.

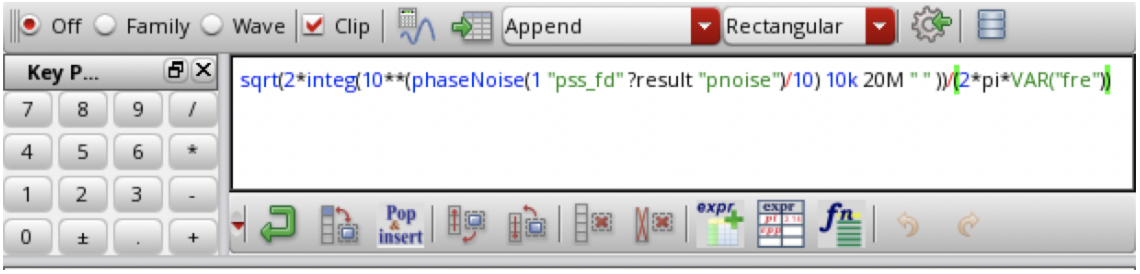


Figure E.5: Jitter Cadence Calculator Formula

E.3 Corner Simulation

The Corner analysis is done in Analog Design Environment (ADE)XL. So first step is to make sure the global variables are defined correctly. Then the corners can start to be defined, in add corner. First step is to add five corners after nominal. Then define the temperature and design variables variation, this variation is -40 ºC and 120 ºC, and 1.08 V, 1.2 V and 1.32 V respectively. There may be more than one design variable and they should be added and defined as previously mentioned. At last in the model files select the respective model for the CMOS used, in this work is the L130E_HS12_V241.lib.scs. Then in each of the five corners select tt, ff, ss, fnsp and snfp. This can be observed in Figure E.6. To run the analysis first the tests to run have to be selected and then just run the simulation.

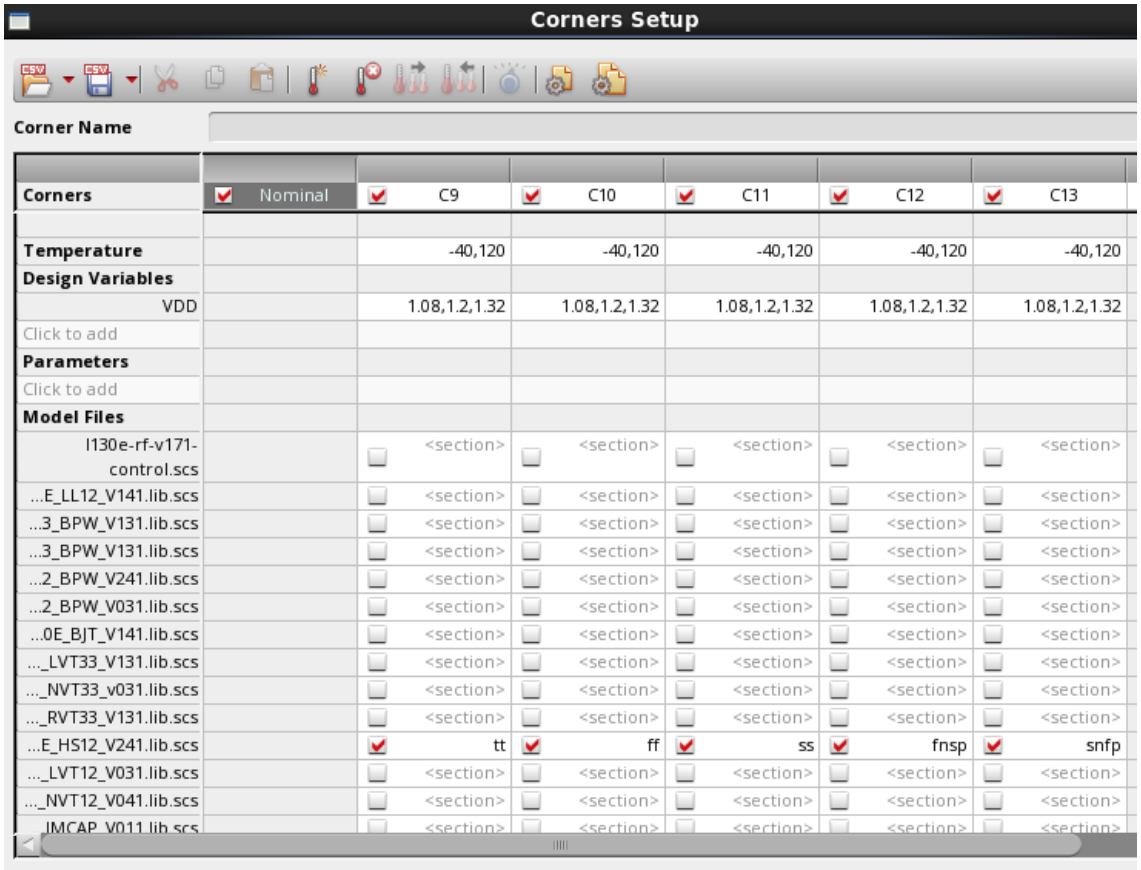


Figure E.6: Corners Definition

E.4 Monte Carlo Simulation

The Monte Carlo Simulation runs in ADEXL. As can be observed in Figure E.7, the first step is to select the Monte Carlo Sampling option in ADEXL.

Then select the dark green run symbol, highlighted in Figure E.7, and in Statistical

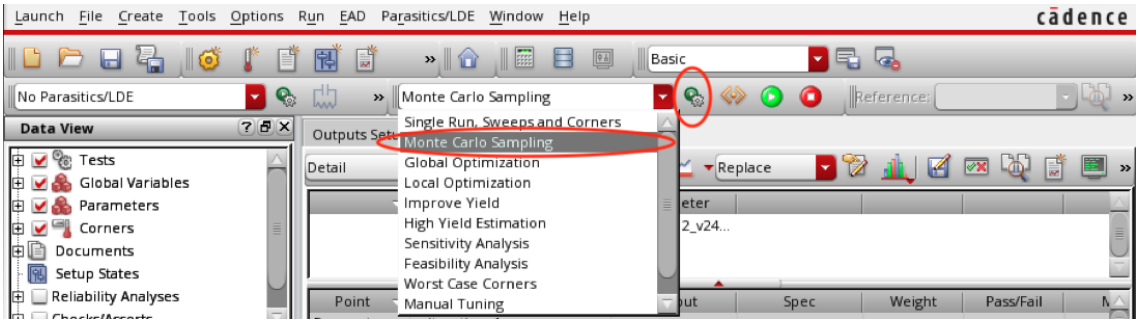


Figure E.7: Monte Carlo ADEXL simulation selection

Variation select all, in sampling method use Low-Discrepancy Sequence and 200 points, at last select save process data and mismatch data. This can be observed in Figure E.8 Then add the variable sigma equal to 3 in the global variables.

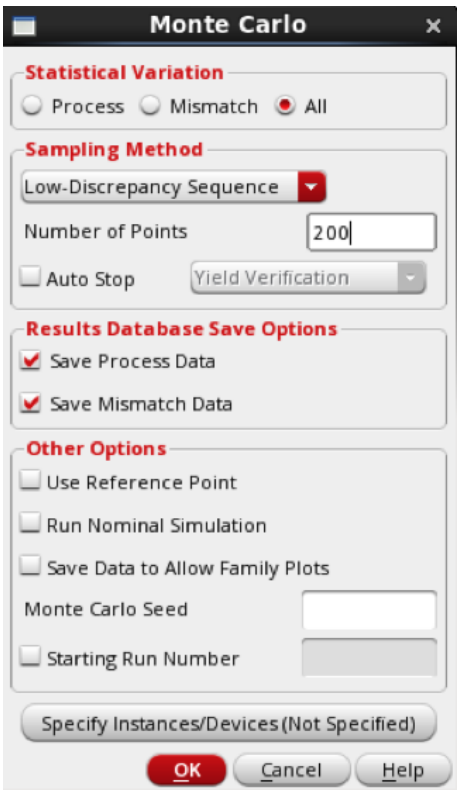


Figure E.8: Monte Carlo ADEXL simulation definition

Next the Monte Carlo Corner needs to be added. This corner doesn't need Temperature or design variables variation. However the model file can't be the same as the one used for the corner simulations. In this work the file used for the Monte Carlo Simulation was the L130E_HS12_V241_mc_corner.lib.scs, and it can be found in /eda/technologies /umc/130nm/20160202/_G_01-MIXED_MODE_RFCMOS13-1P8M-MMC-FSG-L130E /Designkits/Cadence_IC6/Models/Spectre/Monte_Carlo. Then after selecting the model file select mc in the corner definition, where the ff, ss, fnsp and snfp were defined.

After running the simulation the outputs can be seen by selecting detail, and the graph can be obtained by pressing the post processing button and selecting Histogram, Figure E.9.

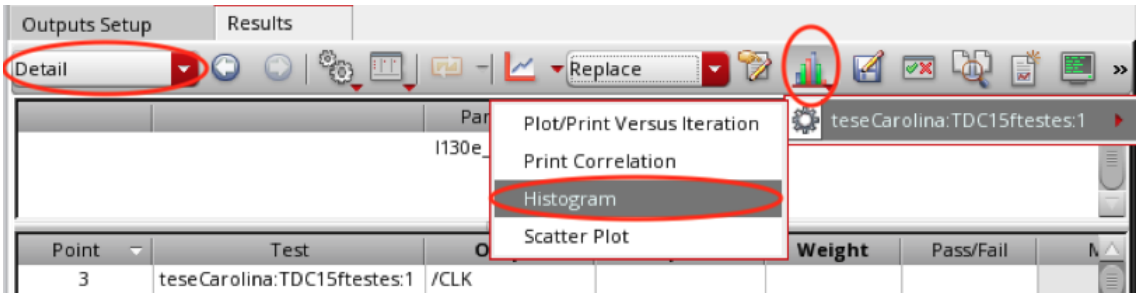


Figure E.9: Monte Carlo ADEXL results

F

Matlab Code

F.1 Inverter Study

```
%Inverter Delay Study

%Variables
Wn = 0.16 : 0.16 : 14; %um
Wn = Wn * 10^-6;
lengthW = length(Wn);
WN = transpose(Wn);

L = 120; %28 : 20 : 120 ;%test with multiple L sizes %nm
L = L * 10^-9;
lengthL = length(L);

%Technology Definitions

CGSDO = 275 * 10^-12; %F/m

CGOXN = 12.6547 * 10^-3; %F/m^2
CGOXP = 12.0795 * 10^-3; %F/m^2

Sd_s = 0.4 * 10^-6; %m

CJN = 722 * 10^-6; %F/m^2
CJP = 953 * 10^-6; %F/m^2

CJSWN = 99.3 * 10^-12; %F/m
CJSWP = 105 * 10^-12; %F/m

Wp = 2.65 * Wn;

%CL calculation

CL (lengthW,lengthL) = 0;
tHL (lengthW,lengthL) = 0;
tLH (lengthW,lengthL) = 0;
```

```
Tp (lengthW,lengthL) = 0;
TP (lengthL,lengthW) = 0;

CgdN(lengthW,lengthL) = 0;
CgdP(lengthW,lengthL) = 0;
CdbN(lengthW,lengthL) = 0;
CdbP(lengthW,lengthL) = 0;
CgN(lengthW,lengthL) = 0;
CgP(lengthW,lengthL) = 0;

Rn(lengthW,lengthL) = 0;
Rp(lengthW,lengthL) = 0;

figure(1)
title('Inverter L = 120 nm')
yyaxis left

for j=1:lengthL
    for i=1:lengthW

        CgdN(i,j) = Wn(i) * CGSD0; %F

        CgdP(i,j) = Wp(i) * CGSD0; %F

        CdbN(i,j) = (Wn(i)+Sd_s)*2*CJSWN+Wn(i)*Sd_s*CJN; %F

        CdbP(i,j) = (Wp(i)+Sd_s)*2*CJSWP+Wp(i)*Sd_s*CJP; %F

        CgN(i,j) = 2*CGSD0*Wn(i)+Wn(i)*L(j)*CGOXN; %F

        CgP(i,j) = 2*CGSD0*Wp(i)+Wp(i)*L(j)*CGOXP; %F

        CL (i,j)= CdbN(i,j) + CdbP(i,j) + 2*CgdP(i,j) + 2*CgdN(i,j) + CgN (i,j
    ) + CgP (i,j); %F

        Rn(i,j)=12.5*10^3*L(j)/Wn(i); %ohm
        Rp(i,j)=30*10^3*L(j)/Wp(i); %ohm

        tHL(i,j)= CL(i,j)*0.69*Rn(i,j); %s
        tLH(i,j)= CL(i,j)*0.69*Rp(i,j); %s

        Tp(i,j)=(tHL(i,j)+tLH(i,j))/2; %s

    end

    TP(j,:) = transpose(Tp(:,j));
    plot(WN * 10^6,TP(j,:) * 10^12,'DisplayName','Delay')
    hold on
    xlabel('Wn (um)')
    ylabel('Time (ps)')
```

```
84
85      yyaxis right
86      plot ( WN * 10^6 , CL(:,1) * 10^15 , "DisplayName" , 'CL' )
87
88      xlabel ( 'Wn (um)' )
89      ylabel ( 'Capacity (fF)' )
90  end
91
92  hold off
93  legend show
```

Listing F.1: Inverter Delay Study

```
1  %INVERSOR Study Simulation
2
3  VOUT = readmatrix('VoutInversor.csv',"Range","A2:CD570"); %Vout Signals
4  Vin = readmatrix('vinINV.csv',"Range","A2:B568"); %Vin Signal
5
6  X = 1 : 2 : 100;
7  Y = 2 : 2 : 100;
8  Wn = 0.16 : 0.32 : 13; %Wn Width
9  L = 120 * 10^-9; %Length
10
11 SizeVin = size(Vin);
12 SizevOut = size(VOUT);
13 Sample = ( SizevOut(1,2) / 2 );
14
15 %DELAYS
16
17 values = SizeVin(1,1);
18
19 d0(SizeVin(1,2)/2+1) = 0;
20 time(SizeVin(1,2)/2+1,1) = 0;
21 delay(SizeVin(1,2)/2+1,1) = 0;
22 Rn(SizeVin(1,2)/2+1,1) = 0;
23 Rp(SizeVin(1,2)/2+1,1) = 0;
24 CL(SizeVin(1,2)/2+1,1) = 0;
25
26 for i = 1 : Sample
27
28     for j = 1 : values
29
30         if Vin(j,X(1)) >= 4.995*10^-9 && Vin(j,X(1)) <= 5.06*10^-9
31
32             d0(i) = Vin(j,Y(1));
33             time(1,1) = 5.00005*10^-9;
34
35         end
36
37     end
38 end
```

```
39
40
41 for i = 2 : Sample + 1
42    for k = 1 : SizevOut(1,1)
43
44         if VOUT(k,X(i-1)) >= 4.995*10^-9 && VOUT(k,X(i-1)) <= 5.06*10^-9
45
46             d0(i) = VOUT(k,Y(i-1));
47
48             if d0(i) >= 0.55 && d0(i) <= 0.65
49                 time(i,1) = VOUT(k,X(i-1));
50
51             else
52             end
53         else
54         end
55
56
57     end
58
59     delay(i-1,1) = time(i,1) - time(1,1);
60
61     %CL
62     Rn(i-1,1) = 6.26 * 10^3 * L / (Wn(i-1) * 10^-6);
63     Rp(i-1,1) = 17.8 * 10^3 * L / (Wn(i-1) * 2.65 * 10^-6);
64     CL(i-1,1) = delay(i-1,1) / (0.69 * (Rn(i-1,1) + Rn(i-1,1)));
65
66 end
67
68 figure(1)
69 title ( 'Delay & CL for L = 120 nm' )
70 yyaxis left
71
72 plot ( Wn(:) , delay (:,1) * 10^12 / 2 , "DisplayName" , 'Delay' )
73
74 xlabel ( 'Wn (um)' )
75 ylabel ( 'Time (ps)' )
76
77 yyaxis right
78 plot ( Wn(:) , CL(:,1) * 10^15 , "DisplayName" , 'CL' )
79
80 xlabel ( 'Wn (um)' )
81 ylabel ( 'Capacity (fF)' )
82
83 legend show
```

Listing F.2: Inverter Delay Study Simulation

F.2 TDC Study

```
1
2  %Delay, Transfer Functions & Linearity
3
4  Clk12i12f = (20 : 1 : 300) ; %ps
5  Clk12i6f = (20 : 1 : 300) ; %ps
6  Clk6i6f = (20 : 1 : 300) ; %ps
7
8  Vin = 50.05 ; %ps
9  bits = 4 ;
10 states = 0 : 1 : 2^bits-1;
11
12 X = 1 : 2 : 800;
13 Y = 2 : 2 : 800;
14
15 L = 120 * 10^-9;
16
17 D12i12f = readmatrix('12uI12uFdelay.csv',"Range","A2:AD2336"); %delay line
18
19 sample12i12f = width(D12i12f)/2;
20 size12i12f = length(D12i12f);
21
22 D12i6f = readmatrix('12uI6uFdelay.csv',"Range","A2:AD2336"); %delay line
23
24 sample12i6f = width(D12i6f)/2;
25 size12i6f = length(D12i6f);
26
27 D6i6f = readmatrix('6uI6uFdelay.csv',"Range","A2:AD2336"); %delay line
28
29 sample6i6f = width(D6i6f)/2;
30 size6i6f = length(D6i6f);
31
32 %Delay FFD
33
34 delayFFD12i12f = 28; %ps
35 delayFFD12i6f = 28; %ps
36 delayFFD6i6f = 27; %ps
37
38 %Delay line
39
40 d112i12f (sample12i12f) = 0;
41 time12i12f (sample12i12f) = 0;
42 delay12i12f (sample12i12f) = 0;
43 delayinv12i12f (sample12i12f) = 0;
44
45 for i = 1 : sample12i12f
46    for k = 1 : size12i12f
47
48        if D12i12f(k,X(i)) >= 0 && D12i12f(k,X(i)) <= 1*10^-9
```

```
49
50              d112i12f(i) = D12i12f(k,Y(i));
51
52              if d112i12f(i) >= 0.55 && d112i12f(i) <= 0.65
53                  time12i12f(i) = D12i12f(k,X(i));
54
55              else
56              end
57          else
58          end
59
60     end
61
62      delay12i12f(i+1) = time12i12f(i) - Vin*10^-12;
63      delayinv12i12f (i) = delay12i12f (i+1) - delay12i12f (i);
64 end
65
66 figure(1)
67
68 plot (states(2:16) , delayinv12i12f*10^12 ,'DisplayName','WnI = 12 um / WnF =
       12 um','LineStyle','-')
69 hold on
70
71 d112i6f (sample12i6f) = 0;
72 time12i6f (sample12i6f) = 0;
73 delay12i6f (sample12i6f) = 0;
74 delayinv12i6f (sample12i6f) = 0;
75
76 for i = 1 : sample12i6f
77     for k = 1 : size12i6f
78
79          if D12i6f(k,X(i)) >= 0 && D12i6f(k,X(i)) <= 1*10^-9
80
81              d112i6f(i) = D12i6f(k,Y(i));
82
83              if d112i6f(i) >= 0.55 && d112i6f(i) <= 0.65
84                  time12i6f(i) = D12i6f(k,X(i));
85
86              else
87              end
88          else
89          end
90
91     end
92
93      delay12i6f(i+1) = time12i6f(i) - Vin*10^-12;
94      delayinv12i6f (i) = delay12i6f (i+1) - delay12i6f (i);
95 end
96
97
```

```
plot (states(2:16) , delayinv12i6f * 10^12 ,'DisplayName','WnI = 12 um / WnF = 6 um','LineStyle','-')
hold on

d16i6f (sample6i6f) = 0;
time6i6f (sample6i6f) = 0;
delay6i6f (sample6i6f) = 0;
delayinv6i6f (sample6i6f) = 0;

for i = 1 : sample6i6f
   for k = 1 : size6i6f

        if D6i6f(k,X(i)) >= 0 && D6i6f(k,X(i)) <= 1*10^-9

            d16i6f(i) = D6i6f(k,Y(i));

            if d16i6f(i) >= 0.55 && d16i6f(i) <= 0.65
                time6i6f(i) = D6i6f(k,X(i));

            else
            end
        else
        end

   end

    delay6i6f(i+1) = time6i6f(i) - Vin*10^-12;
    delayinv6i6f (i) = delay6i6f (i+1) - delay6i6f (i);
end

plot (states(2:16) , delayinv6i6f*10^12 ,'DisplayName','WnI = 6 um / WnF = 6 um','LineStyle','-')
hold on

title ('Delay 1 Inverter')
ylabel('time (ps)')
xlabel('Estados')

set(gca,'XTick',0:1:15)
legend show
grid on
hold off

%BITS

B012i12f = readmatrix('12uI12uFb0.csv',"Range","A2:UP2441");
B112i12f = readmatrix('12uI12uFb1.csv',"Range","A2:UP2441");
B212i12f = readmatrix('12uI12uFb2.csv',"Range","A2:UP2441");
B312i12f = readmatrix('12uI12uFb3.csv',"Range","A2:UP2441");

```

```
B012i6f = readmatrix('12uI6uFb0.csv',"Range","A2:UP2441");
B112i6f = readmatrix('12uI6uFb1.csv',"Range","A2:UP2441");
B212i6f = readmatrix('12uI6uFb2.csv',"Range","A2:UP2441");
B312i6f = readmatrix('12uI6uFb3.csv',"Range","A2:UP2441");

B06i6f = readmatrix('6uI6uFb0.csv',"Range","A2:UP2441");
B16i6f = readmatrix('6uI6uFb1.csv',"Range","A2:UP2441");
B26i6f = readmatrix('6uI6uFb2.csv',"Range","A2:UP2441");
B36i6f = readmatrix('6uI6uFb3.csv',"Range","A2:UP2441");

nclk12i12f = width(B012i12f)/2 ;
sizeB12i12f = length(B012i12f);

db012i12f (nclk12i12f) = 0;
bit012i12f (nclk12i12f) = 0;
db112i12f (nclk12i12f) = 0;
bit112i12f (nclk12i12f) = 0;
db212i12f (nclk12i12f) = 0;
bit212i12f (nclk12i12f) = 0;
db312i12f (nclk12i12f) = 0;
bit312i12f (nclk12i12f) = 0;
Btotal12i12f(nclk12i12f) = 0;

for i = 1 : nclk12i12f

    for j = 1 : sizeB12i12f

        if B012i12f(j,X(i)) >= 1*10^-9 && B012i12f(j,X(i)) <= 1.2*10^-9

            db012i12f(i) = B012i12f(j,Y(i));

            if db012i12f(i) > 1
                bit012i12f(i) = 1;
            else
                bit012i12f(i) = 0;
            end
        else
        end

        if B112i12f(j,X(i)) >= 1*10^-9 && B112i12f(j,X(i)) <= 1.2*10^-9

            db112i12f(i) = B112i12f(j,Y(i));

            if db112i12f(i) > 1
                bit112i12f(i) = 2;
            else
                bit112i12f(i) = 0;
            end
        else
        end
```

```
196
197         if B212i12f(j,X(i)) >= 1*10^-9 && B212i12f(j,X(i)) <= 1.2*10^-9
198
199             db212i12f(i) = B212i12f(j,Y(i));
200
201             if db212i12f(i) > 1
202                 bit212i12f(i) = 4;
203             else
204                 bit212i12f(i) = 0;
205             end
206         else
207         end
208
209         if B312i12f(j,X(i)) >= 1*10^-9 && B312i12f(j,X(i)) <= 1.2*10^-9
210
211             db312i12f(i) = B312i12f(j,Y(i));
212
213             if db312i12f(i) > 1
214                 bit312i12f(i) = 8;
215             else
216                 bit312i12f(i) = 0;
217             end
218         else
219         end
220
221     end
222     Btotal12i12f(i)=bit312i12f(i)+bit212i12f(i)+bit112i12f(i)+bit012i12f(i);
223 end
224
225 nclk12i6f = width(B012i6f)/2 ;
226 sizeB12i6f = length(B012i6f);
227
228 db012i6f (nclk12i6f) = 0;
229 bit012i6f (nclk12i6f) = 0;
230 db112i6f (nclk12i6f) = 0;
231 bit112i6f (nclk12i6f) = 0;
232 db212i6f (nclk12i6f) = 0;
233 bit212i6f (nclk12i6f) = 0;
234 db312i6f (nclk12i6f) = 0;
235 bit312i6f (nclk12i6f) = 0;
236 Btotal12i6f(nclk12i6f) = 0;
237
238 for i = 1 : nclk12i6f
239
240     for j = 1 : sizeB12i6f
241
242         if B012i6f(j,X(i)) >= 1*10^-9 && B012i6f(j,X(i)) <= 1.2*10^-9
243
244             db012i6f(i) = B012i6f(j,Y(i));
245
```

```
246             if db012i6f(i) > 1
247                 bit012i6f(i) = 1;
248             else
249                 bit012i6f(i) = 0;
250             end
251         else
252         end
253 
254         if B112i6f(j,X(i)) >= 1*10^-9 && B112i6f(j,X(i)) <= 1.2*10^-9
255 
256             db112i6f(i) = B112i6f(j,Y(i));
257 
258             if db112i6f(i) > 1
259                 bit112i6f(i) = 2;
260             else
261                 bit112i6f(i) = 0;
262             end
263         else
264         end
265 
266         if B212i6f(j,X(i)) >= 1*10^-9 && B212i6f(j,X(i)) <= 1.2*10^-9
267 
268             db212i6f(i) = B212i6f(j,Y(i));
269 
270             if db212i6f(i) > 1
271                 bit212i6f(i) = 4;
272             else
273                 bit212i6f(i) = 0;
274             end
275         else
276         end
277 
278         if B312i6f(j,X(i)) >= 1*10^-9 && B312i6f(j,X(i)) <= 1.2*10^-9
279 
280             db312i6f(i) = B312i6f(j,Y(i));
281 
282             if db312i6f(i) > 1
283                 bit312i6f(i) = 8;
284             else
285                 bit312i6f(i) = 0;
286             end
287         else
288         end
289 
290     end
291     Btotal12i6f(i)=bit312i6f(i)+bit212i6f(i)+bit112i6f(i)+bit012i6f(i);
292 end
293 
294 nclk6i6f = width(B06i6f)/2 ;
295 sizeB6i6f = length(B06i6f);
```

```
296
297 db06i6f (nclk6i6f) = 0;
298 bit06i6f (nclk6i6f) = 0;
299 db16i6f (nclk6i6f) = 0;
300 bit16i6f (nclk6i6f) = 0;
301 db26i6f (nclk6i6f) = 0;
302 bit26i6f (nclk6i6f) = 0;
303 db36i6f (nclk6i6f) = 0;
304 bit36i6f (nclk6i6f) = 0;
305 Btotal6i6f(nclk6i6f) = 0;
306
307 for i = 1 : nclk6i6f
308
309     for j = 1 : sizeB6i6f
310
311         if B06i6f(j,X(i)) >= 1*10^-9 && B06i6f(j,X(i)) <= 1.2*10^-9
312
313             db06i6f(i) = B06i6f(j,Y(i));
314
315             if db06i6f(i) > 1
316                 bit06i6f(i) = 1;
317             else
318                 bit06i6f(i) = 0;
319             end
320         else
321         end
322
323         if B16i6f(j,X(i)) >= 1*10^-9 && B16i6f(j,X(i)) <= 1.2*10^-9
324
325             db16i6f(i) = B16i6f(j,Y(i));
326
327             if db16i6f(i) > 1
328                 bit16i6f(i) = 2;
329             else
330                 bit16i6f(i) = 0;
331             end
332         else
333         end
334
335         if B26i6f(j,X(i)) >= 1*10^-9 && B26i6f(j,X(i)) <= 1.2*10^-9
336
337             db26i6f(i) = B26i6f(j,Y(i));
338
339             if db26i6f(i) > 1
340                 bit26i6f(i) = 4;
341             else
342                 bit26i6f(i) = 0;
343             end
344         else
345         end
```

```
346
347         if B36i6f(j,X(i)) >= 1*10^-9 && B36i6f(j,X(i)) <= 1.2*10^-9
348
349             db36i6f(i) = B36i6f(j,Y(i));
350
351             if db36i6f(i) > 1
352                 bit36i6f(i) = 8;
353             else
354                 bit36i6f(i) = 0;
355             end
356         else
357         end
358
359     end
360     Btotal6i6f(i)=bit36i6f(i)+bit26i6f(i)+bit16i6f(i)+bit06i6f(i);
361 end
362
363
364 figure(2)
365
366 plot(Clk12i12f,Btotal12i12f,'DisplayName','WnI = 12 um / WnF = 12 um')
367 axis([delayFFD12i12f 303 0 15 ])
368 hold on
369
370 plot(Clk12i6f,Btotal12i6f,'DisplayName','WnI = 12 um / WnF = 6 um')
371 axis([delayFFD12i12f 303 0 15 ])
372 hold on
373
374 plot(Clk6i6f,Btotal6i6f,'DisplayName','WnI = 6 um / WnF = 6 um')
375 axis([delayFFD12i12f 303 0 15 ])
376 hold on
377
378 %Discovers the time at wich bit happens
379 math12i12f(length(Btotal12i12f))=0;
380 code12i12f(2^bits) = 0;
381 dTcode12i12f(2^bits) = 0;
382 dTcode12i12f(1) = delayFFD12i12f;
383 t=1;
384
385 for i=1:(length(Btotal12i12f)-1)
386
387     math12i12f(i) = Btotal12i12f(i+1) - Btotal12i12f(i);
388
389     if math12i12f(i) == 0
390     else
391         code12i12f(t+1) = Btotal12i12f(i+1);
392         dTcode12i12f(t+1) = Clk12i12f(i+1);
393         t=t+1;
394     end
395
```

```
396 end
397
398
399 delaycode12i12f(length(dTcode12i12f)-1)=0;
400
401 for i=1:length(dTcode12i12f)-1
402     delaycode12i12f(i) = dTcode12i12f(i+1) - dTcode12i12f(i);
403 end
404
405 Tlsb12i12f = sum(delaycode12i12f)/(2^bits-1);
406
407 TIDEAL12i12f=(0:15)*Tlsb12i12f+dTcode12i12f(1);
408
409 plot(TIDEAL12i12f,code12i12f,'LineStyle','--')
410 hold on
411 axis([delayFFD12i12f 300 0 15 ])
412
413 contas12i6f(length(Btotal12i6f))=0;
414 code12i6f(2^bits) = 0;
415 dTcode12i6f(2^bits) = 0;
416 dTcode12i6f(1) = delayFFD12i6f;
417 t=1;
418
419 for i=1:(length(Btotal12i6f)-1)
420
421     contas12i6f(i) = Btotal12i6f(i+1) - Btotal12i6f(i);
422
423     if contas12i6f(i) == 0
424     else
425         code12i6f(t+1) = Btotal12i6f(i+1);
426         dTcode12i6f(t+1) = Clk12i6f(i+1);
427         t=t+1;
428     end
429
430 end
431
432
433 delaycode12i5f(length(dTcode12i6f)-1)=0;
434
435 for i=1:length(dTcode12i6f)-1
436     delaycode12i5f(i) = dTcode12i6f(i+1) - dTcode12i6f(i);
437 end
438
439 Tlsb12i6f = sum(delaycode12i5f)/(2^bits-1);
440
441 TIDEAL12i6f=(0:15)*Tlsb12i6f+dTcode12i6f(1);
442
443 plot(TIDEAL12i6f,code12i6f,'LineStyle','--')
444 hold on
445 axis([delayFFD12i6f 268 0 15 ])
```

```
446
447 contas6i6f(length(Btotal6i6f))=0;
448 code6i6f(2^bits) = 0;
449 dTcode6i6f(2^bits) = 0;
450 dTcode6i6f(1) = delayFFD6i6f;
451 t=1;
452
453 for i=1:(length(Btotal6i6f)-1)
454
455     contas6i6f(i) = Btotal6i6f(i+1) - Btotal6i6f(i);
456
457     if contas6i6f(i) == 0
458     else
459         code6i6f(t+1) = Btotal6i6f(i+1);
460         dTcode6i6f(t+1) = Clk6i6f(i+1);
461         t=t+1;
462     end
463
464 end
465
466
467 delaycode6i6f(length(dTcode6i6f)-1)=0;
468
469 for i=1:length(dTcode6i6f)-1
470     delaycode6i6f(i) = dTcode6i6f(i+1) - dTcode6i6f(i);
471 end
472
473 Tlsb6i6f = sum(delaycode6i6f)/(2^bits-1);
474
475 TIDEAL6i6f=(0:15)*Tlsb6i6f+dTcode6i6f(1);
476
477 plot(TIDEAL6i6f,code6i6f,'LineStyle','--')
478 hold on
479 axis([delayFFD6i6f 300 0 15 ])
480
481 title('Transfer Function')
482 set(gca,'YTick',0:1:15)
483 grid on
484 legend show
485
486 xlabel('time (ps)')
487 ylabel('Estados')
488 hold off
489
490 %Linearidade
491
492 %DNL
493
494 DNL12i12f(2^bits-1) = 0;
495
```

```
for i=1:2^bits-1
    DNL12i12f(i)=(dTcode12i12f(i+1)-dTcode12i12f(i))/Tlsb12i12f - 1;
end

%INL
INL12i12f(2^bits) = 0;

for i=1:2^bits
    INL12i12f(i)= (dTcode12i12f(i) - dTcode12i12f(1))/Tlsb12i12f - code12i12f(
    i);
end

erroINL212i12f=max(INL12i12f)-min(INL12i12f);
erroDNL12i12f=max(DNL12i12f)-min(DNL12i12f);

linearity_INL12i12f=4-log(max(INL12i12f)-min(INL12i12f))/log(2);
linearidty_DNL12i12f=4-log(max(DNL12i12f)-min(DNL12i12f))/log(2);


DNL12i6f(2^bits-1) = 0;

for i=1:2^bits-1
    DNL12i6f(i)=(dTcode12i6f(i+1)-dTcode12i6f(i))/Tlsb12i6f - 1;
end

%INL
INL212i6f(2^bits) = 0;

for i=1:2^bits
    INL212i6f(i)= (dTcode12i6f(i) - dTcode12i6f(1))/Tlsb12i6f - code12i6f(i);
end

erroINL212i6f=max(INL212i6f)-min(INL212i6f);
erroDNL12i6f=max(DNL12i6f)-min(DNL12i6f);

linearity_INL12i6f=4-log(max(INL212i6f)-min(INL212i6f))/log(2);
linearity_DNL12i6f=4-log(max(DNL12i6f)-min(DNL12i6f))/log(2);

%DNL

DNL6i6f(2^bits-1) = 0;

for i=1:2^bits-1
    DNL6i6f(i)=(dTcode6i6f(i+1)-dTcode6i6f(i))/Tlsb6i6f - 1;
end

%INL

INL26i6f(2^bits) = 0;

```

```
545 for i=1:2^bits
546     INL26i6f(i)= (dTcode6i6f(i) - dTcode6i6f(1))/Tlsb6i6f - code6i6f(i);
547 end
548
549 erroINL26i6f=max(INL26i6f)-min(INL26i6f);
550 erroDNL6i6f=max(DNL6i6f)-min(DNL6i6f);
551
552 linearity_INL6i6f=4-log(max(INL26i6f)-min(INL26i6f))/log(2);
553 linearity_DNL6i6f=4-log(max(DNL6i6f)-min(DNL6i6f))/log(2);
554
555 disp('- - - - - - - - - - - - - - - - ')
556
557 disp(['Lin 12i12f INL = ' num2str(linearity_INL12i12f) ' / DNL = ' num2str(
    linearidty_DNL12i12f)])
558 disp(['Lin 12i5f INL = ' num2str(linearity_INL12i6f) ' / DNL = ' num2str(
    linearity_DNL12i6f)])
559 disp(['Lin 5i5f INL = ' num2str(linearity_INL6i6f) ' / DNL = ' num2str(
    linearity_DNL6i6f)])
560
561 figure(3)
562 subplot(2,1,1)
563
564 plot(code12i12f,INL12i12f,'DisplayName','12i12f')
565 hold on
566 plot(code12i6f,INL212i6f,'DisplayName','12i5f')
567 hold on
568 plot(code6i6f,INL26i6f,'DisplayName','5i5f')
569 hold on
570
571 title('INL')
572 set(gca,'XTick',0:1:15)
573 grid on
574 ylabel('LSB')
575 xlabel('Estados')
576 legend show
577 hold off
578
579 subplot(2,1,2)
580
581 plot(code12i12f(2:16),DNL12i12f,'DisplayName','12i12f')
582 hold on
583 plot(code12i6f(2:16),DNL12i6f,'DisplayName','12i5f')
584 hold on
585 plot(code6i6f(2:16),DNL6i6f,'DisplayName','5i5f')
586 hold on
587
588 title('DNL')
589 set(gca,'XTick',0:1:15)
590 grid on
591 ylabel('LSB')
```

```
592 xlabel('Estados')
593 legend show
594 hold off
595
596 disp(['TLSB 12i12f = ' num2str(Tlsb12i12f) ' ps'])
597 disp(['TLSB 12i5f = ' num2str(Tlsb12i6f) ' ps'])
598 disp(['TLSB 5i5f = ' num2str(Tlsb6i6f) ' ps'])
599
600 disp(['Erro 12i12f inl = ' num2str(erroINL212i12f) ' LSB / dnl = ' num2str(
    erroDNL12i12f) 'LSB'])
601 disp(['Erro 12i6f inl = ' num2str(erroINL212i6f) ' LSB / dnl = ' num2str(
    erroDNL12i6f) 'LSB'])
602 disp(['Erro 6i6f inl = ' num2str(erroINL26i6f) ' LSB / dnl = ' num2str(
    erroDNL6i6f) 'LSB'])
603
604 disp('- - - - - - - - - - - - - - - - - ')
```

Listing F.3: TDC Simulation Study