Cairo University

Cairo University
Faculty of Engineering

Department of Electronics
and Communications

A 1 GS/s 6 bits Time-Based Analog-to-Digital Converter

By

Ahmed Ali El Sayed Ali
Ali El Hussien Ali Hassan
Maged Ali Ahmed Ahmed Ghazal
Mohammed Mostafa Mohammed Hassoubh
Nabil Mohammed Nabil Gomaa

Under the Supervision of

Dr. Hassan Mostafa
Dr. Mohamed Refky Amin

A Graduation Project Report Submitted to
the Faculty of Engineering at Cairo University
In Partial Fulfillment of the Requirements for the
Degree of
Bachelor of Science
in
Electronics and Communications Engineering

Faculty of Engineering, Cairo University

Giza, Egypt

July 2014

Table of Contents

List of Tables

List of Figures

List of Symbols and Abbreviations

ADC	Analog-to-Digital Converter
DAC	Digital-to-Analog Converter
TADC	Time-Based Analog-to-Digital Converter
CMOS	Complementary Metal Oxide Semiconductor
DSP	Digital Signal Processing
UWB	Ultra wide Band
4G	Fourth Generation
IC	Integrated Circuit
A/D	Analog-to-Digital Conversion
F_s	Sampling Frequency
F_m	Maximum Input Signal Bandwidth
ENOB	Effective Number of Bits
ERBW	Effective Resolution Bandwidth
LSB	Least Significant Bit
FFT	Fast Fourier Transform
MSB	Most Significant Bit
FS	Full Scale
DR	Dynamic Range
DNL	Differential Non-Linearity
INL	Integral Non-Linearity
SNR	Signal to Noise Ratio
RMS	Root Mean Squared Value
SNDR	Signal to Noise and Distortion Ratio
SFDR	Spurious Free Dynamic Range
THD	Total Harmonic Distortion
V_{pp}	Volts peak-to-peak
FOM	Figure of Merit
OSR	Over Sampling Ratio
SAR	Successive approximation register
S/H	Sample-and-Hold
DA	Delay Adjustment
VTC	Voltage-to-Time Converter
TSMP	Time Mode Signal Processing
TDC	Time-to-Digital Converter

Acknowledgments

First of all, we would like to thank all people who supported us and encouraged us to give more effort in order to reach better results.

We would like to express our appreciation and special thanks to our supervisors Dr. Hassan and Dr. Mohammed for all of their guidance, motivation, support, and help throughout the development of this project and documentation. They also opened our minds to explore new design ideas and search techniques till we reached our goal.

We would also like to present our special thanks and deepest gratitude to our Teaching Assistants of the Electronics and Communications department as they were always there when we needed their help, and support.

We wish to express our love and gratitude to our beloved families; for their understanding and endless love, encouragement and patience through the duration of our studies and our life.

Last but not least, the greatest debt and gratitude is for our dear God who provided us with the capabilities to complete our project and achieve our goal.

Abstract

The Ultra-Wide-Band (UWB) 4G (fourth generation) receivers are attracting many researchers recently. The main idea of these UWB receivers is to convert the noisy analog signal coming from the antenna directly to a digital signal removing all the analog processing blocks. Therefore, the signal will be converted directly to digital and following that a Digital Signal Processor (DSP).

With CMOS technology continued scaling down, designing very fast conventional ADCs is facing several challenges such as accuracy, resolution and power. These challenges make the conventional ADCs incapable of providing the high speed requirement of the UWB receivers front-end.

In the last three years, the Time-based ADCs (TADC) appear as the winning candidate to achieve the front-end ADCs high speed requirements. The TADCs converts the analog signal to a time delay or frequency representation through a circuit called Voltage-to-Time Converter (VTC). Then, the time-represented signal or the frequency-represented signal is converted to digital through a circuit called Time-to-Digital Converter (TDC). Processing the signal in the digital domain is foreseen to reduce the power consumption while keeping the high accuracy and resolution.

Our project is to design Time based ADC that comprises between speed and resolution, while maintaining minimum power consumption. Following that, the TADC chip will be fabricated using 65nm CMOS technology and tested.

We started from studying the concept of TADC and different implementation techniques for it, then we start to work in parallel in two different implementations to reach our goal.

Finally, we achieved from one of two implementation 500 MS/s 6-bit TADC with differential sinusoidal input of 540 mV peak-to-peak amplitude where we can boost the speed using time interleaving concept, While the other design trying to achieve 1 GS/s 6-bit TADC directly with smaller dynamic range equal to 172 mV.

Chapter 1: Introduction

1.1 Motivation

ADC is a device that converts continuous analog input signals to discrete output digital codes to interface analog and digital environments, where it plays very important role in our daily life as they are key components for many of electronic systems. For example, most of ICs today are mixed-signal (Analog part in addition to Digital part) as we need analog part for interfacing with external inputs as defined before and digital part for interfacing with Digital Signal Processing (DSP) blocks like Micro-Processors. This makes our new trend is searching for ADC architectures which offer minimum power consumption with more efficient functionality in order to be used in portable devices where we need to increase battery life as much as we can.

Figure 1 : Analog-to-Digital Conversion

With CMOS technology continued scaling down, designing very fast conventional ADCs is facing several challenges as supplying voltage decreases while the decrease in the transistor threshold voltage is relatively not in the same way, so cascoding transistors becomes more difficult where design of operational amplifiers (Op-Amps) which considered main block for many ADC architectures depends on cascading[1,2].

On the other hand, CMOS technology scaling improves the switching characteristics of MOS transistors offer superb timing accuracy at high frequencies, so we can use a new design approach, in which the time domain resolution of digital signal is represented either by rising or falling edge of input clock transitions which represent voltage resolution of an analog signal in a new way.

Using the advantage of CMOS technology scaling down in terms of area and gate delay, we need to increase digital part implemented on chip relative to analog part by removing pre-processing analog blocks like Low-Noise- Amplifier, Baseband filters, Variable-Gain-Amplifiers,..etc. This can be achieved using new approach of ADC design which called Time-Based ADC (TADC) [4].

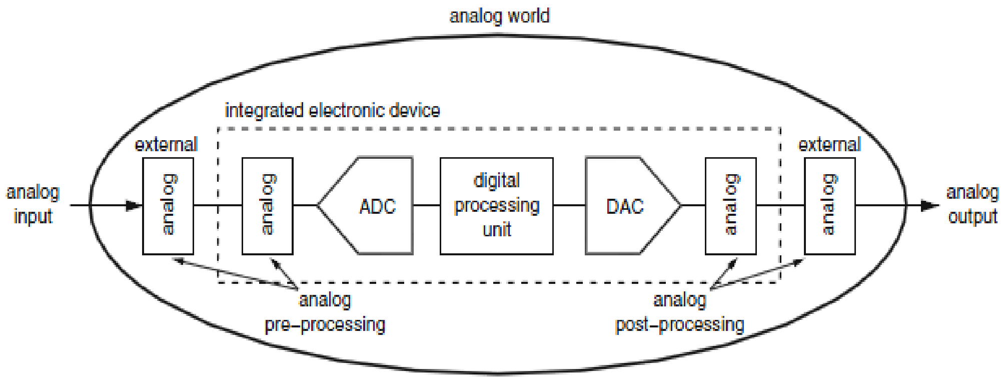


Figure 2 : Generic digital signal processing system

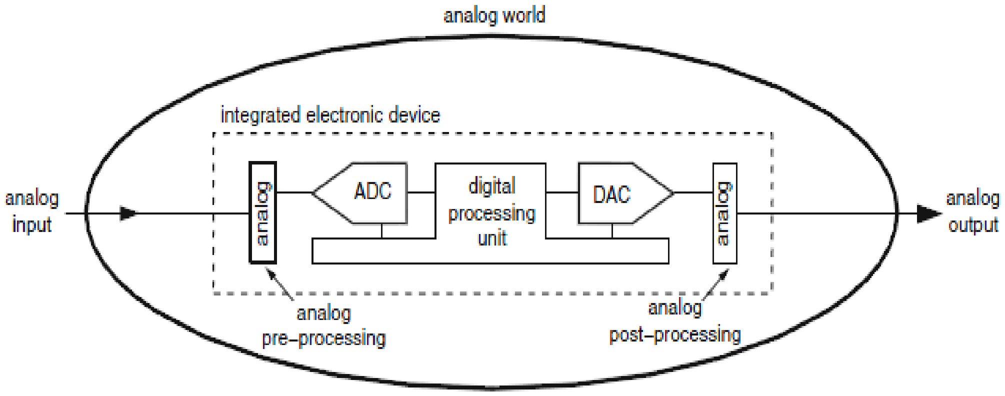


Figure 3 : Future digital signal processing system

Now, we can demonstrate our idea by simple comparison between processing in voltage domain and time domain in terms of Technology Process, Resolution, and Power Consumption, where we can notice that at 350 nm technology process there is a huge difference in power consumption while with 90 nm and 45 nm technology processes [5], voltage processing seems to be not affected by sizing scale down which we take about before, where this is shown in the following figure [4].

Here are the figures:

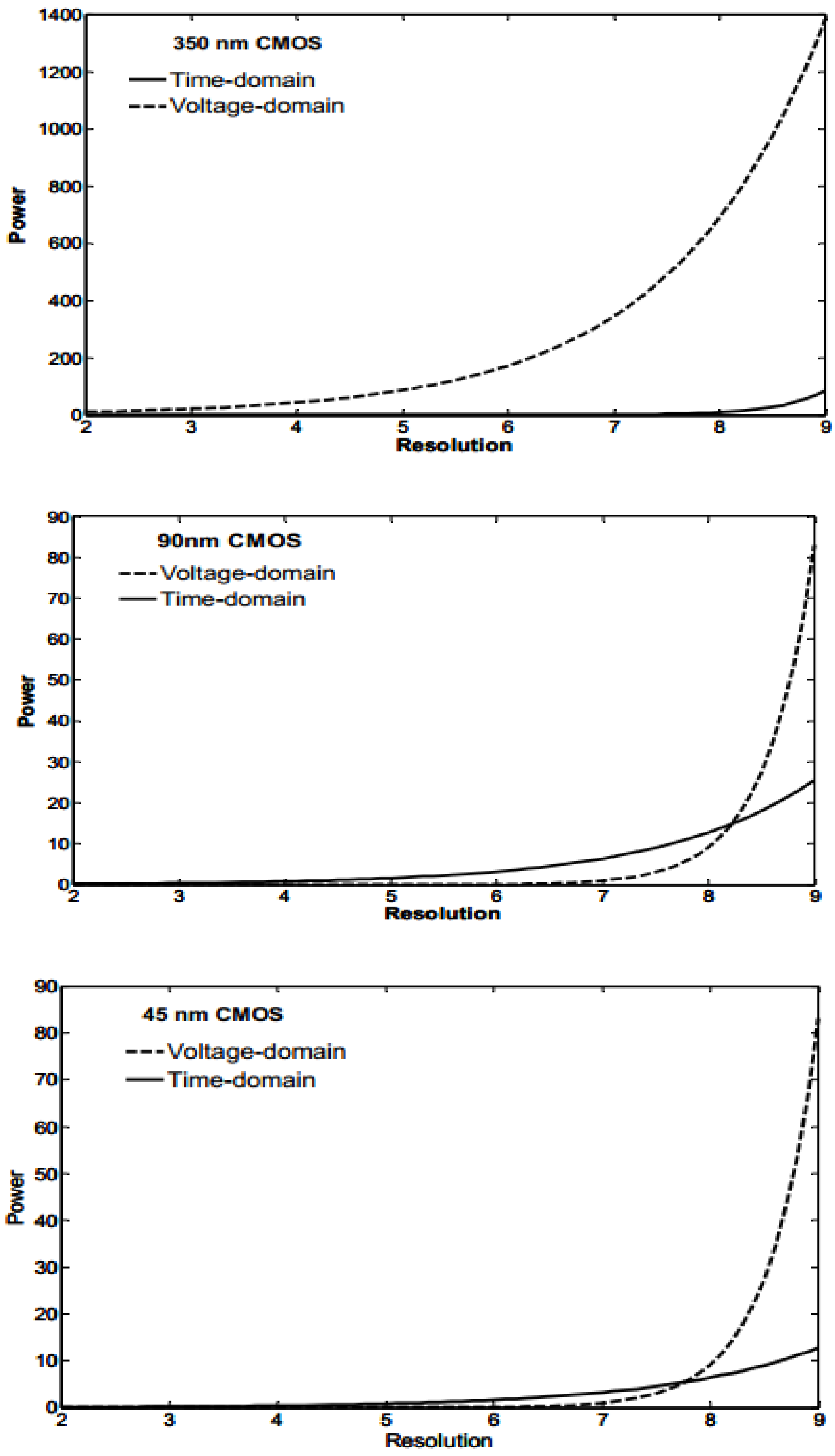


Figure 4 : Time Domain and Voltage Domain minimum power comparison

1.2 Background

A digital signal is discrete in terms of both the amplitude and the time. Thus, two main functions are necessary to obtain a digital waveform:

1. Sampling (To achieve the Discrete Time)
2. Quantization (To achieve the Discrete Amplitude)

In order to show the A/D conversion in an example, consider the simple analog waveform shown in Figure

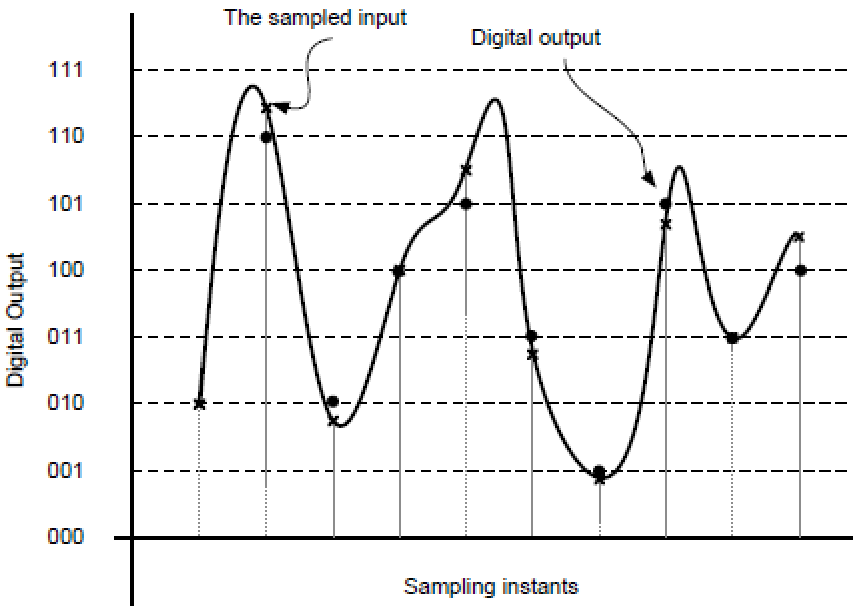


Figure 5 : 3 bits ADC

1.2.1 Sampling

The ADC samples the input signal with a rate called the sampling frequency(F_s). The sampling frequency must be at least equal to twice the maximum frequency appearing in the input signal Bandwidth (F_m) , this condition is referred to as the Nyquist criterion [6].

$$F_s \geq 2^* F_m$$

In case of sampling frequency was less than maximum input frequency, aliasing occurs and we can’t reconstruct signal again from its’ samples.

We can understand what is meant by aliasing if we take a look on frequency spectrum as shown in the following figures [6], and [7].

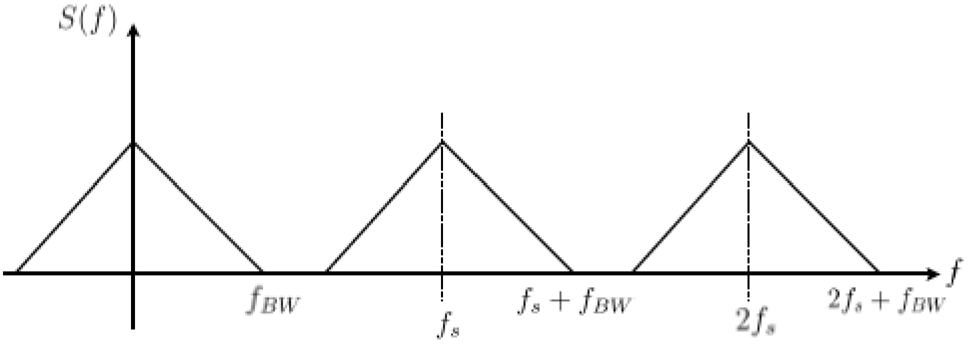


Figure 6 : Nyquist criterion is satisfied

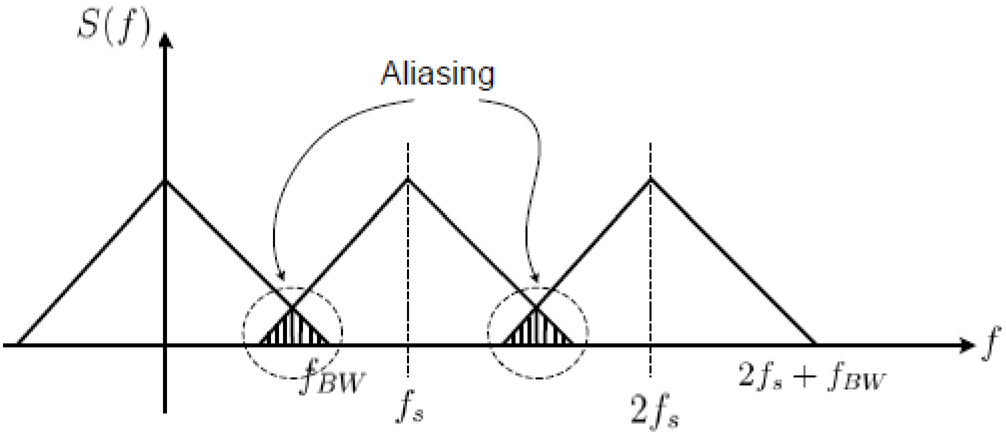


Figure 7 : Nyquist criterion is not satisfied

1.2.2 Quantization

Quantization is also necessary for analog-to-digital converters. Quantization is the process of assigning certain ranges of values from a continuous signal range to discrete values where step size is equal to the Least Significant Bit (LSB) = $\frac{FS}{2^N}$, where N is the number of bits in the digital word and FS is the Full Scale of the analog input. Due to this assignment, quantization errors occur. A quantization error is the difference between the quantized value and the original signal.

If we take an example for an analog input is swept from 0 to FS, the quantization error of a 3 bits ADC will look like a saw-tooth waveform as shown in figure [8].

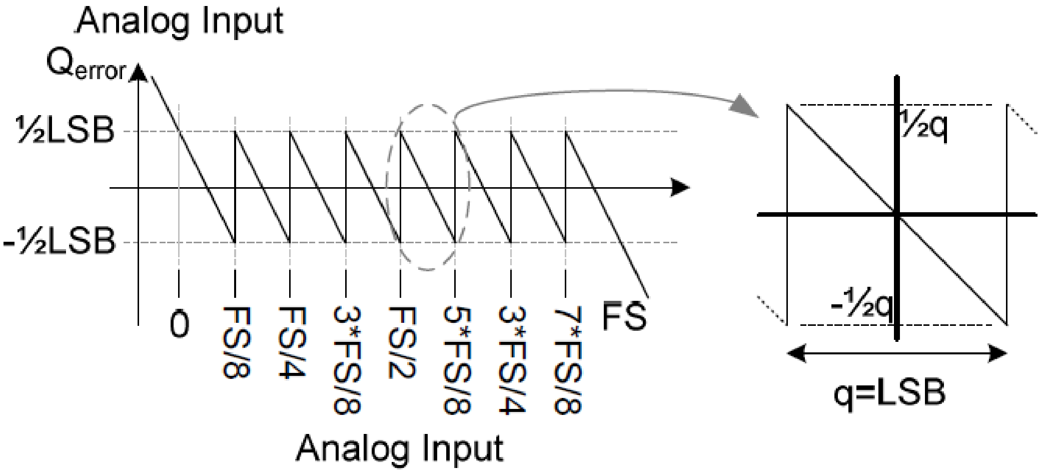


Figure 8 : Quantization error of a full-scale ramp input

For an ADC, quantization error is always in the range of

-0:5xLSB ≤ Quantization error ≤ 0.5xLSB

Quantization errors are directly related to the resolution of the ADC. An ADC that needs an accuracy within a very small margin of error is going to need more quantization levels. More levels require a larger number of digital bits to encode all the information. Higher resolution often comes at the cost of converter speed, so converters need to be optimized for required speeds and resolutions. This optimization depends greatly on the type of architecture chosen for the ADC design.

Chapter 2: ADC Specifications and Types

2.1 ADC Specifications

ADCs are characterized based on their application. Applications not requiring speed can be specified by static specifications. These include resolution, offset, linearity and gain error. However high speed applications like UWB receivers require ADCs to meet dynamic specifications. These include Clock Jitter, Signal to Noise Ratio (SNR), Signal to Noise and Distortion Ratio (SNDR), Spurious Free Dynamic Range (SFDR), and Total Harmonic Distortion (THD) [7].

2.1.1 Static Specifications:

1. Resolution:

Typically a converter with N-bit resolution should convert the input range of analog signal to 2^{N-1} discrete levels. Noise and non-linearity limits the resolution of a converter.

2. Offset:

Offset is simply a shift for zero input and it changes the transfer characteristic in such a way that all the steps are shifted by the amount of the offset which is generally expressed in LSB as shown in figure [9].

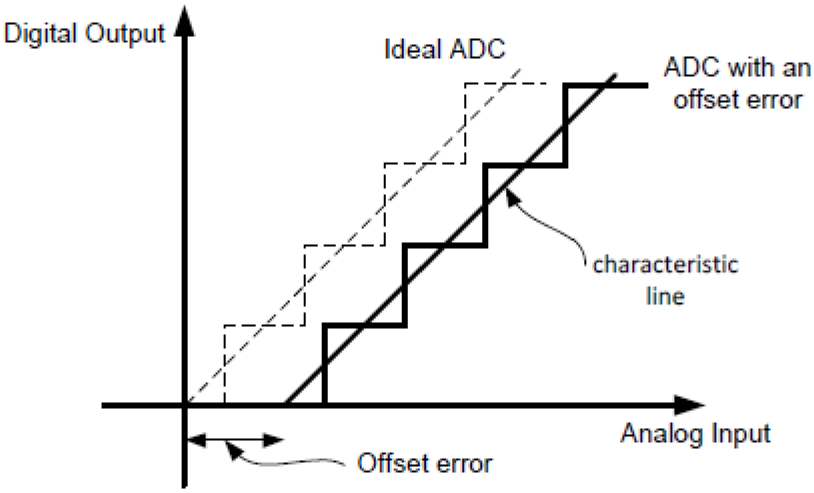


Figure 9 : Ideal transfer characteristic and the one with offset

3. Gain Error:

The gain of the transfer characteristic of an ADC can be calculated by drawing a straight line which interpolates the actual transfer curve as shown in figure [10].

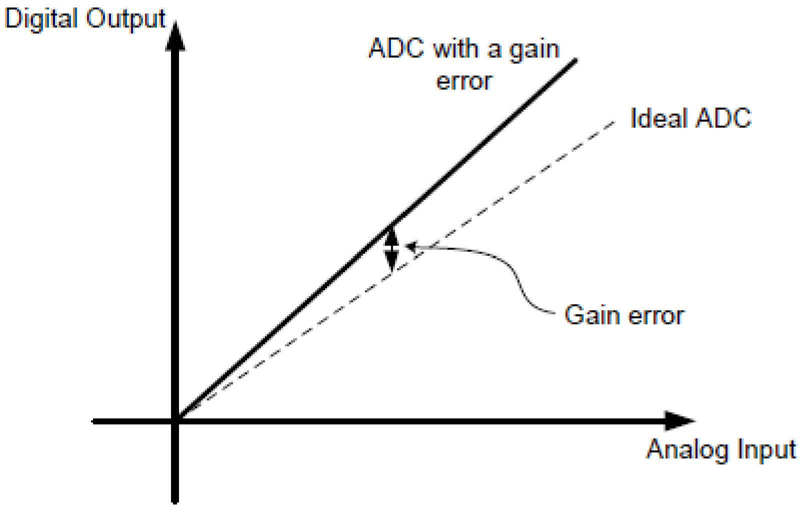


Figure 10 : Ideal transfer characteristic and the one with gain error

4. Differential Non-Linearity (DNL):

When the step size of an ADC's output is not equal to the ideal step size, the ADC is said to have differential nonlinearity. The DNL measurement for an ADC is classified based on amount of least significant bit (LSB) values that the actual transfer function deviates from the ideal transfer function. If the DNL is greater than 1 LSB, a non-monotonic transfer function will cause missing codes. Figure [11] shows deviation of actual transfer function from ideal one.

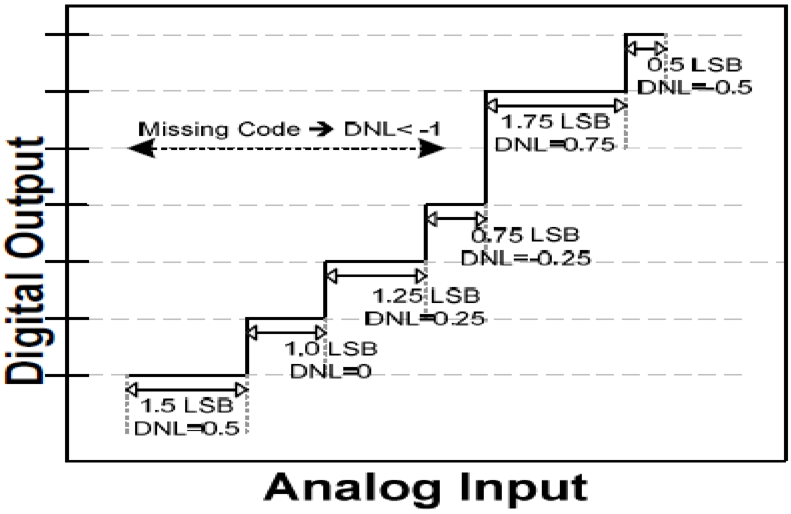


Figure 11 : DNL Error greater and less than 1 LSB

5. Integral Non-Linearity (INL):

INL is defined as the integral of the DNL errors, so good INL guarantees good DNL. The INL error tells us how much actual reading of ADC deviates from ideal one. Figure [12] shows two different methods for calculation INL.

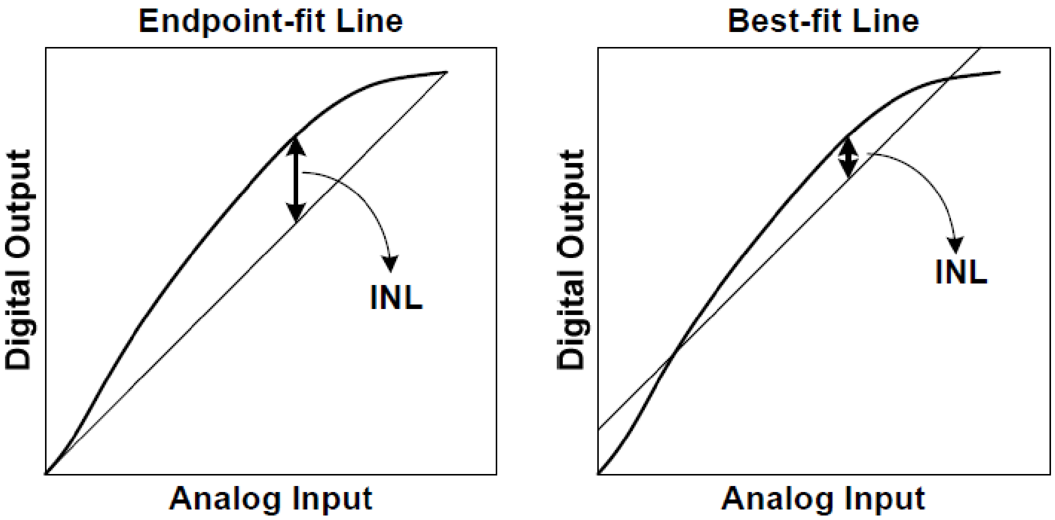


Figure 12 : Endpoint-fit and best-fit methods for INL calculation

2.1.2 Dynamic Specifications:

1. Clock Jitter:

Jitter is the timing variations of a set of signal edges from their ideal values. Jitters in clock signals are typically caused by noise or other disturbances in the system. Contributing factors include thermal noise, power supply variations, loading conditions, device noise, and interference coupled from nearby circuits. The predictable component of jitter in these circuits is called deterministic jitter, while the remaining components of jitter are called random jitter [11].

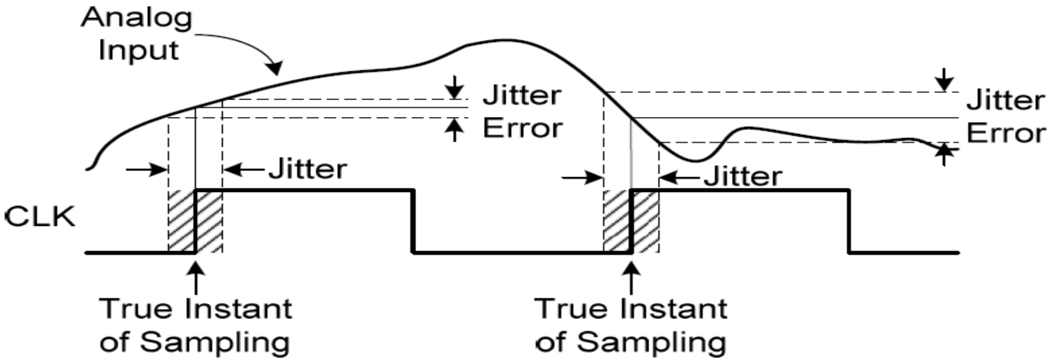


Figure 13 : Clock Jitter

2. Signal to Noise Ratio (SNR):

SNR is defined as the ratio of the power of the full scale input signal to the power of the noise at the output of the ADC and it is expressed by:

$$SNR = 10\log\left(\frac{Full\ scale\ input\ power}{Quantization\ noise\ power + circuit\ noise\ power}\right)db$$

3. Signal to Noise and Distortion Ratio (SNDR):

SNDR is the ratio between the RMS of the input signal and the RMS of the harmonic components in addition to the noise. Similarly, it depends on the frequency and the amplitude of the input signal as SNR.
It can be expressed by:

$$SNR = 10\log\left(\frac{Full\ scale\ input\ power}{Noise\ power + Distortion\ power}\right)db$$

Figure [14]: Shows how we can calculate SNR + SNDR using Fast Fourier Transform (FFT) with input sinusoidal wave [10].

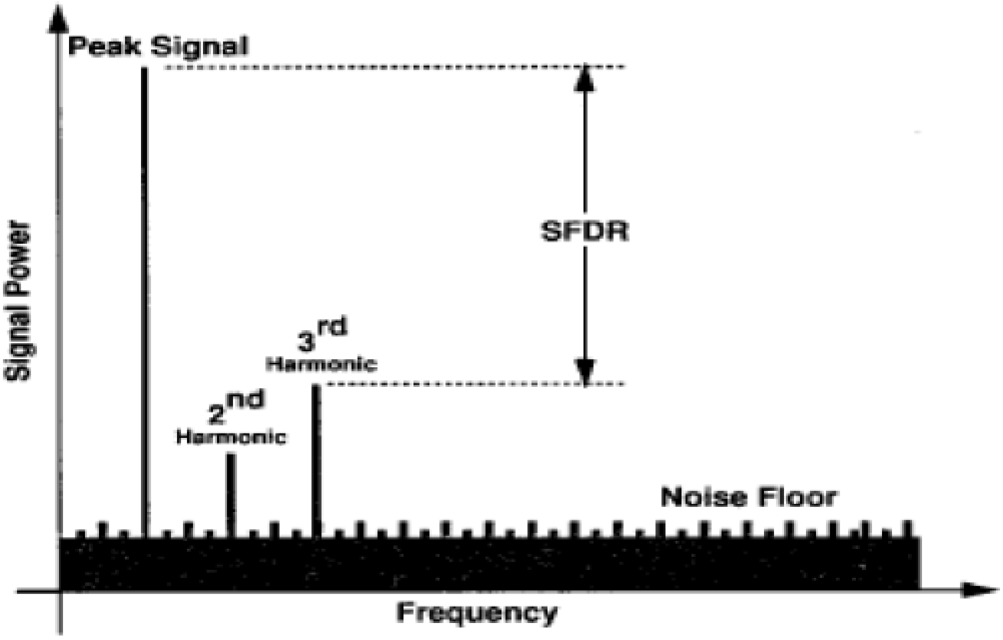


Figure 14 : Typical ADC Output Spectrum

4. Effective Number of Bits (ENOB):

The theoretical SNR expression is 6.02xN+1.76 dB for an N-bits ADC, so we ideally calculate ENOB by:

$$ENOB = \frac{SNDR_{db} - 1.76}{6.02}$$

5. Spurious Free Dynamic Range (SFDR):

It is defined as the ratio of the RMS value of the carrier frequency at the input of the ADC to the RMS value of the next largest harmonic distortion component (which is referred to as a “spurious” or a “spur”) at its output.

6. Total Harmonic Distortion (THD):

THD is the sum of rms of all the harmonic components (except the fundamental) of the output periodic signal to fundamental tone.

7. Figure of Merit (FOM):

FOM is generally used to compare the performances of different ADCs. It basically gives an idea on the power efficiency of the converter and it is expressed as:

i. $FOM_1 = \frac{Power}{2^{ENOB} * F_s}$ → [8]

ii. $FOM_2 = \frac{DR^2 * F_s}{Power}$ → [9]

2.2 ADC Types

2.2.1 Overview

They can be divided into types where first type directly converts the analog input into digital output which called conventional ADCs, while the other type does the conversion in an indirect way by first converting the analog signal into an intermediate representation such as time, then it converts this intermediate representation into digital code [3], e.g. TADC.

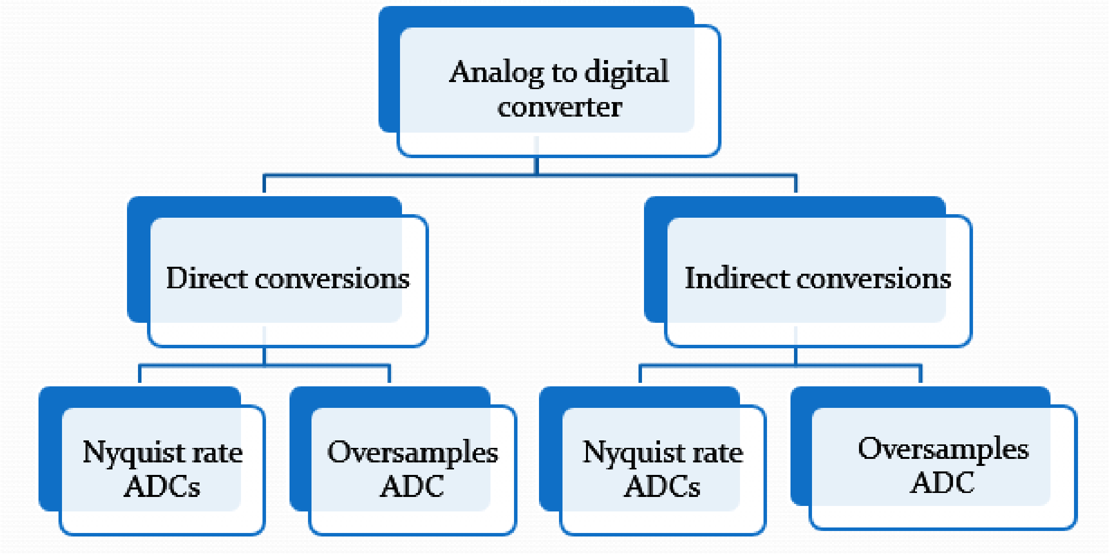


Figure 15 : ADC Types

A Nyquist rate ADC is an ADC type in which the sampling frequency is equal to twice the maximum frequency in the input signal BW, while Oversampling ADC is an ADC type in which there ratio between sampling frequency and maximum input frequency that’s called oversampling ratio (OSR).

OSR is an integer value, greater than 2 and it can reach 1000.

2.2.2 Conventional ADCs:

1. Flash ADCs

Flash ADC, which is the fastest and one of the simplest ADC architectures, is shown in Figure [16]. It performs $2^N - 1$ level quantization with an equal number of comparators. The reference voltages for the comparators are generated using a resistor ladder, which is connected between the positive (+*Vref*) and the negative (-*Vref*) reference voltage determining the full-scale signal range.

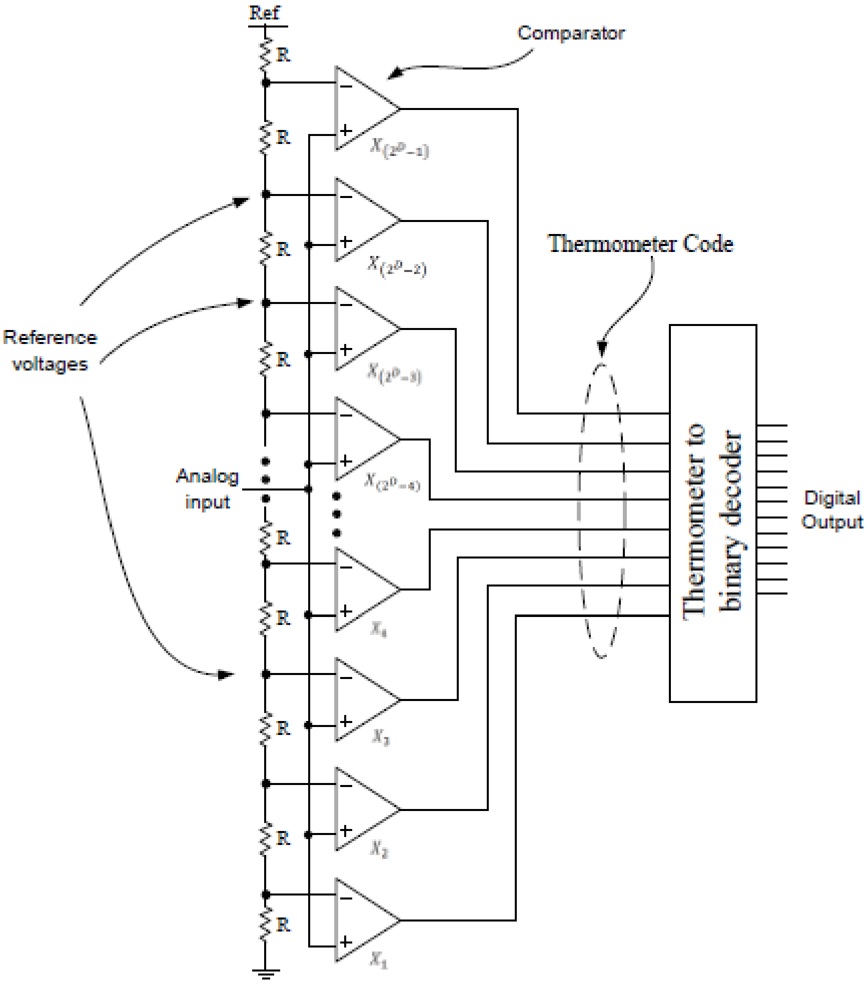


Figure 16 : Flash ADC

2. Successive Approximation Register (SAR) ADC

The Successive Approximation Register (SAR) ADC became a popular topology to implement ADCs in the 1970s with the availability of several logic ICs from companies such as AMD.

The algorithm used in Successive Approximation is based on a binary search algorithm, and thus is more component efficient than Flash ADCs which use a brute force approach to perform data conversion. Figure [17] illustrates how SAR ADC works.

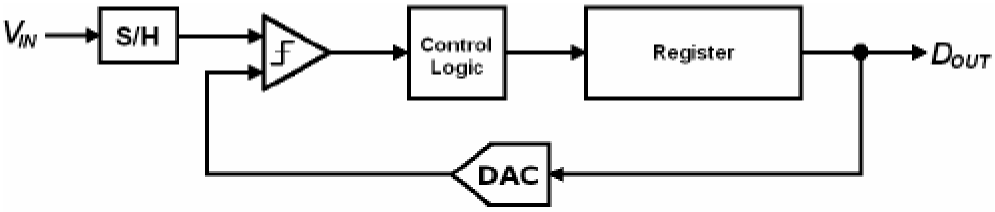


Figure 17 : SAR ADC

3. Delta-Sigma ADC

Delta-Sigma ADC use oversampling, modulator and digital filtering to achieve high resolution from a single bit quantizer as shown in Figure [18].

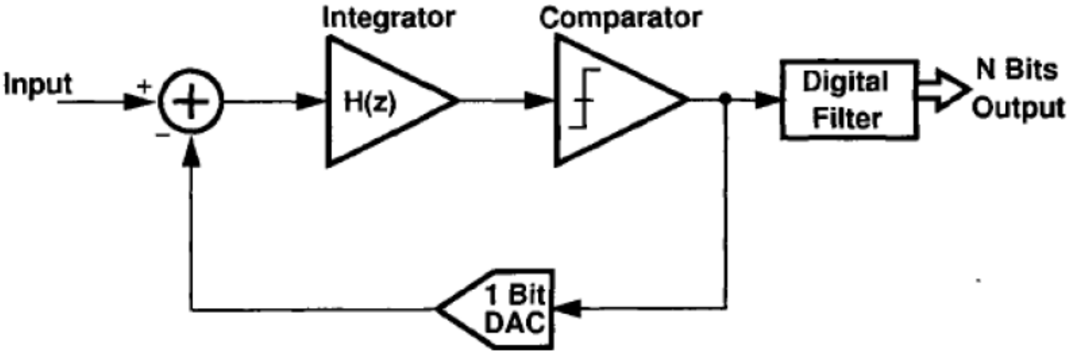


Figure 18 : Sigma-Delta ADC

4. Pipelined ADC

Although it is not common to see Pipelined ADCs in very high-speed, medium resolution applications, they will be briefly mentioned here since they can be considered as an extended version of subranging (or two-step) ADC. Basically, the operating principle of a subranging ADC can be extended to more than 2 ADCs to implement a Pipelined ADC as shown in Figure [19].

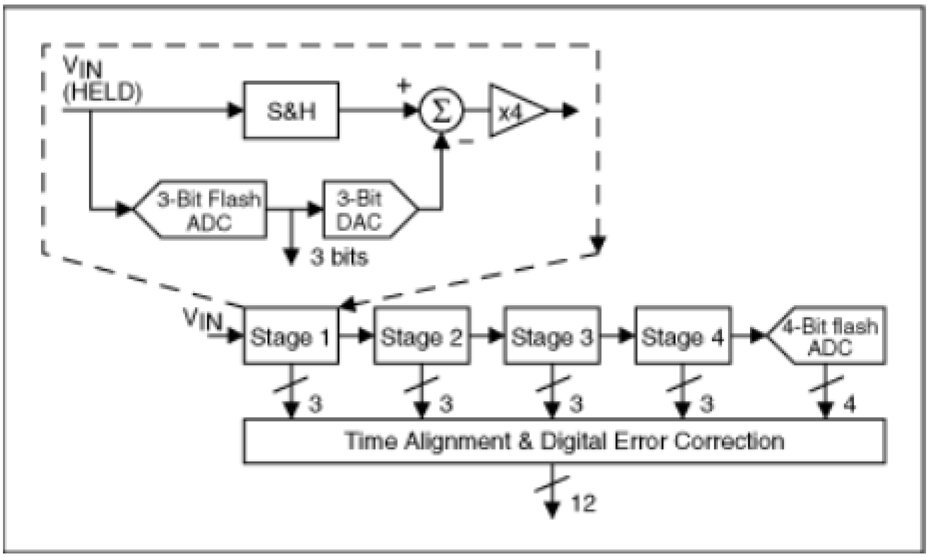


Figure 19 : Pipelined ADC with four 3-bit stages (each stage resolves two bits)

Comparison between different architectures:

Architecture	Speed	Resolution	Area
Flash	High	Low	High
SAR	Low-Medium	Medium-high	Low
Sigma-Delta	Low	High	Medium
Pipelined	Medium-high	Medium-high	Medium

Table 1 : Comparison between different architectures

2.2.3 Time-Based ADC (TADC)

1. Dual Slope TADC

It can be used for high accuracy, low data rate applications like Digital Multimeter which known as avometer.

It is named dual slope TADC, as it perform conversion through two phases:

During first phase has a fixed duration $T1$ controlled by the running of a counter for 2^N clock cycles. During this period, the integrator input is connected to the analog input sample and the integrator output starts to build up, where integrator output can be expressed by:

$$V_{out} = \frac{V_{in}T_1}{RC}$$

While during second phase, the input of the integrator is switched to the reference voltage $Vref$ .So, the slope is fixed during this phase, unlike the first phase which have variable slope, resulting in a variable duration $T2$ for the second phase. The integrator output starts to go down until it reaches zero, where $T2$ can be expressed by:

$$T_2 = T_1 \frac{V_{in}}{V_{ref}}$$

Here is shown simplified Figure [20] for Dual Slope TADC:

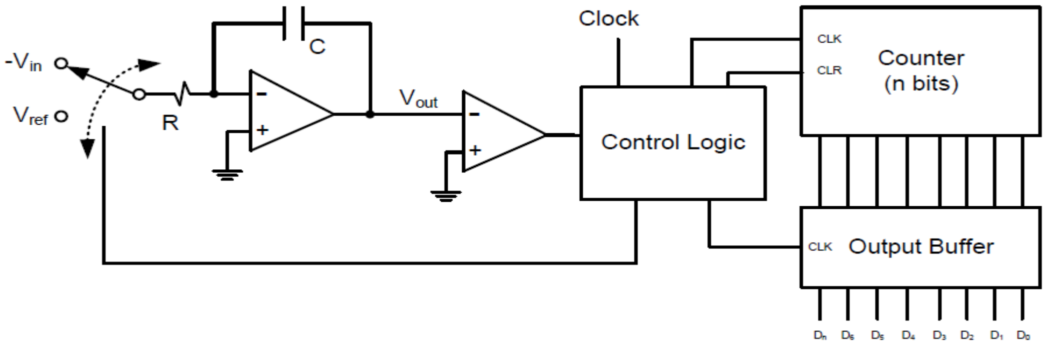


Figure 20 : Simplified Block Diagram for Dual Slope TADC

2. VTC/TDC based TADCs

It converts the analog signal to a time delay or frequency representation through a circuit called Voltage-to-Time Converter (VTC). Then, the time-represented signal or the frequency-represented signal is converted to digital through a circuit called Time-to-Digital Converter (TDC).

Here is shown in Figure [21] a simplified block diagram for this type:

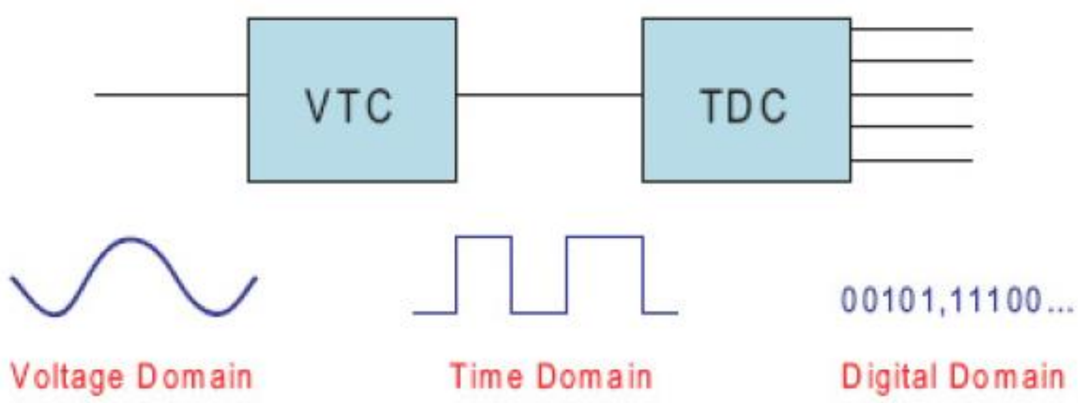


Figure 21 : TADC Architecture

Chapter 3: Delay Line TADC

3.1 Introduction

In this chapter we will introduce a delay-line-based analog-to-digital converter for high-speed applications. The basic idea of the ADC is to convert the sampled input voltage to a delay which controls the propagation speed of a digital pulse through the delay line. The generation of the output digital code is based on the propagation length of the pulse in the delay line in a fixed time window.

Time based signal processing is used in many applications such as time-of-flight measurements and digital phase-locked loops (DPLL) where a particular delay is measured accurately [12]-[13].

The analog to digital conversion can be done using the concept of time-to-digital quantization where the sampled input voltage is represented in time domain and then quantized using a delay line structure.

The proposed time based ADC can operate at high data rates [14]. There is distinct advantages of implementing the quantization process in time domain compared with implementing the quantization process in voltage domain. A 6 bit 500 MS/s delay-line data converter with no calibration is proposed.

3.2 Signal quantization of Delay-Line-Based:

Voltage comparison is not the only method of quantizing analog signals. While in the delay-line-based quantizer a time is quantized, which is proportional to analog signal. A delay line consists of a series of delay cells shown in Fig [22]. During a time window T_s, a digital pulse is applied to the beginning of the delay line and travel through it. The initial value of the output of the delay cell is set to “0” and then switched to “1” after the digital pulse propagate through it.

At the end of the time window T_s, the value of N is the number of triggered delay-cells that satisfies the equation:

$$\sum_{j=1}^{N} Dj < Ts < \sum_{j=1}^{N+1} Dj$$

Where D_j is the delay of the j^{th} delay cells. Where the line must be long enough so that the digital pulse does not reach the end of the delay line during the time window T_s.

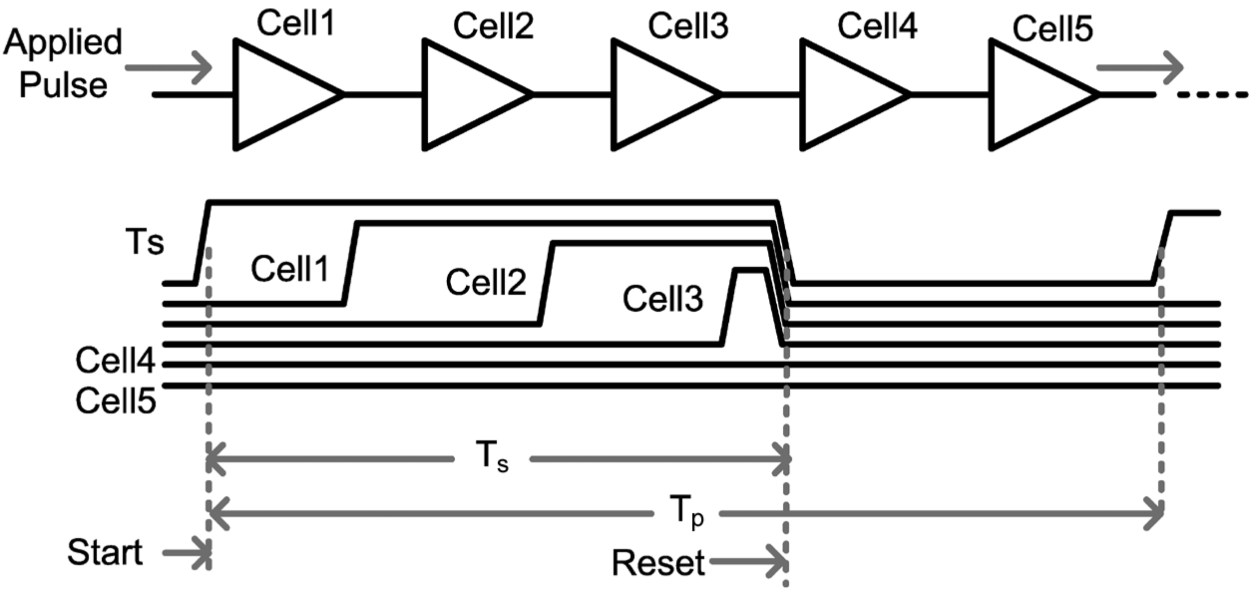


Figure 22

A digital pulse applied to the first delay cell of the delay line. The quantization of the propagation length of the digital pulse is done by collecting the digital outputs of the delay cells inside the delay line.

The time domain quantization can be done by two methods, changing the time window T_s or changing the delay of each delay cell. For simplicity we assume all the delay cells have the delay which equal to D. therefor N equal to

$$N = \frac{Ts}{D}$$

3.2.1 Time-based vs. Delay-based signal quantization:

In the time-based signal quantization, delay cells have a fixed delay while the time window T_s is not fixed and proportional to the input signal and this method is fit to applications such as digital phase locked loop (DPLL), while it is not fit to sampled mode circuit because the system will not work efficiently as the systems work with a fixed sampling rate needs a fixed processing time.

While in delay-based signal quantization, the time window T_s is fixed while the amount of delay of each delay cell is changing according to the analog input. A delay adjustment block (DA) is needed to apply the suitable delay to the delay cells according to the sampled voltage because we work on a voltage mode sampling, to make the pulse propagation speed proportional to the input signal the delay of each delay cell must be inversely proportional to the input.

3.3 Time-based ADC Architecture:

3.3.1 Delay cell

There are various implementation of variable delay cell such as current starved inverter and supply control [15],[16], and we should maintain the inverse relation between delay of each cell and the input signal for wide range of input voltages so delay cell shown in Fig [23].

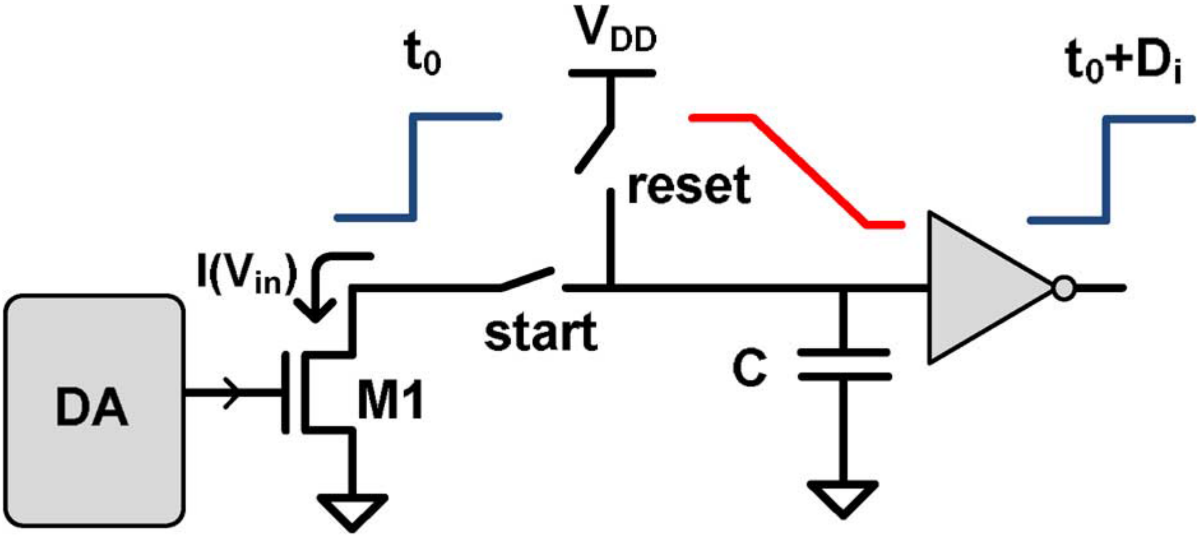


Figure 23

In the proposed controllable delay cell, when the reset switch is on, the capacitive node is charging and when the start switch is on, the capacitive node is linearly discharging using a controllable current source M1, and the delay cell is reset to its high level voltage at the end of time window T_s.

The resulting amount of delay of each delay cell is given by

$$D = \frac{C\,.Vt}{I(Vin)} + To$$

Where To is the additional delay of inverter , C is the capacitance at the charging node and I(vin) is proportional to input voltage (vin) through (DA), and Vt is threshold voltage of inverter, To limits the dynamic range and the linearity of delay cell so it should be small comparable to total delay.

3.3.2 Dual delay line:

In order to increase the resolution of time-based ADC and to decrease the effect of additional delay of inverter in delay cell, a dual delay line is proposed.

A differential DA is the block which control the two delay lines as a function of the sampled input voltage, all delay cells in both delay lines are similar but the amount of delay for each line is different , in other words, when N-cells are fast ,P-cells are slow and vice-versa.

The proposed dual delay line based ADC is shown in Fig [24]. The input pulse is applied to both p-cells and N-cells and propagate with different speeds in the two delay line, and at the end of time window T_s, the outputs of all delay cells in dual lines are latched separately and the encoded as the digital output.

In our design we use two delay lines, each consists of 32 delay cells, and two dummy delay cells at the beginning and one dummy delay cell at the end of the delay line.

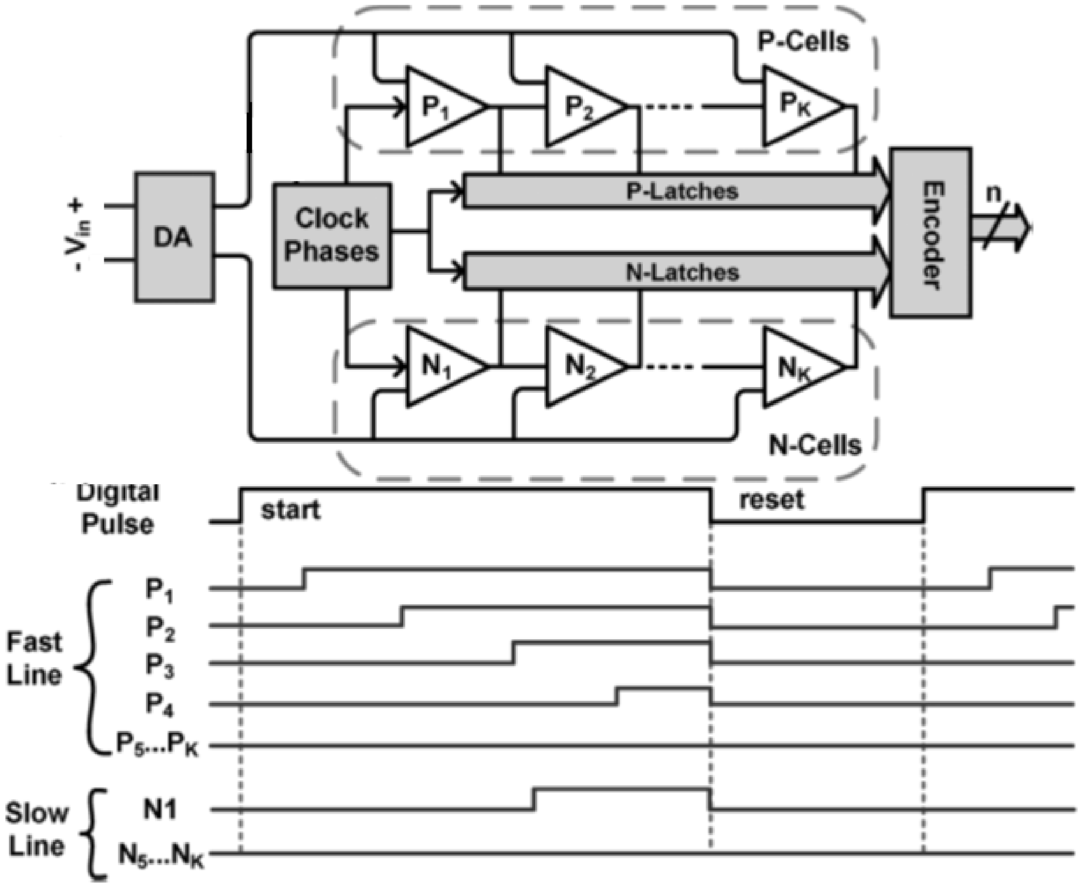


Figure 24

3.4 CIRCUIT DESIGN:

We design and simulate 6 bit 500 MS/s time-based ADC on a standard 65 nm CMOS process, and we will discuss the circuit blocks of the delay-line based ADC.

3.4.1 Delay Adjustment Circuit:

Delay adjustment is shown in Fig [25], it consists of degenerated differential pair where the PMOS transistors work as the input differential pair while the NMOS transistors are the load. The function of this block is to transform the differential input voltage to a differential current in M3 and M4. Based on the difference between the NMOS and PMOS current sources, the bias current of M3 and M4 is set. To decrease the overall power consumption, try to make the value of bias current low.

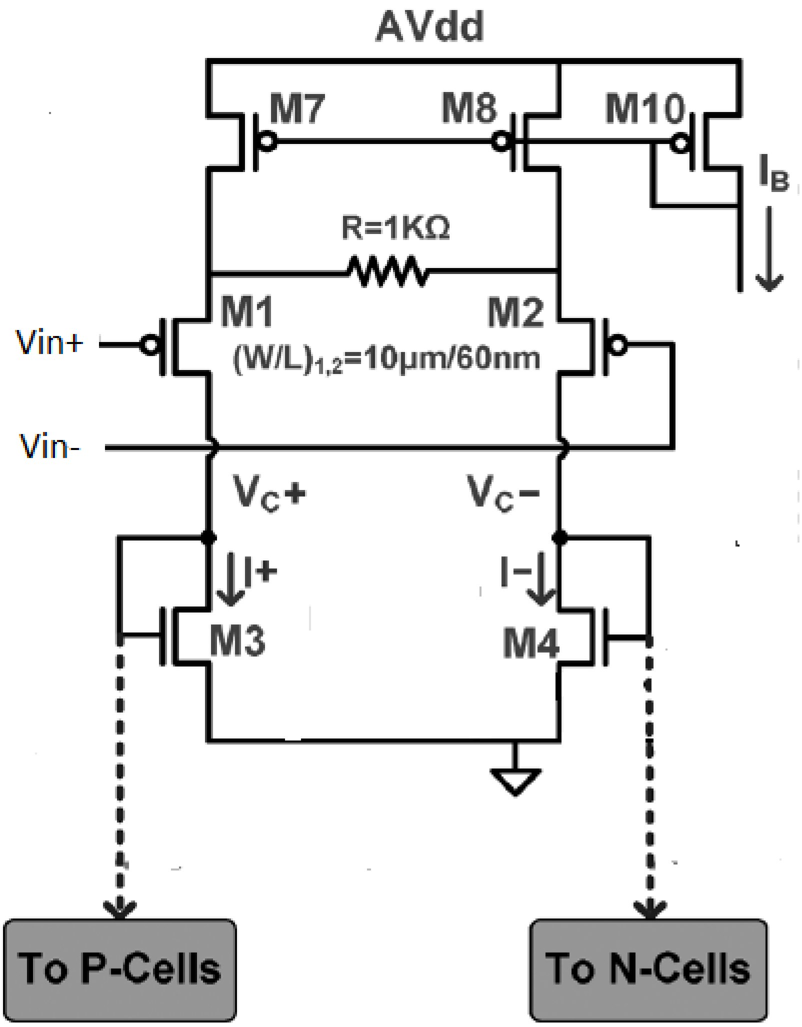

AVdd
M7
M8
M10
I_B
R=1KΩ
M1
M2
Vin+
$(W/L)_{1,2}$=10µm/60nm
Vin-
V_C+
V_C−
I+
I−
M3
M4
To P-Cells
To N-Cells

Figure 25

3.4.1.1 Input differential voltage vs. output differential voltage of DA:

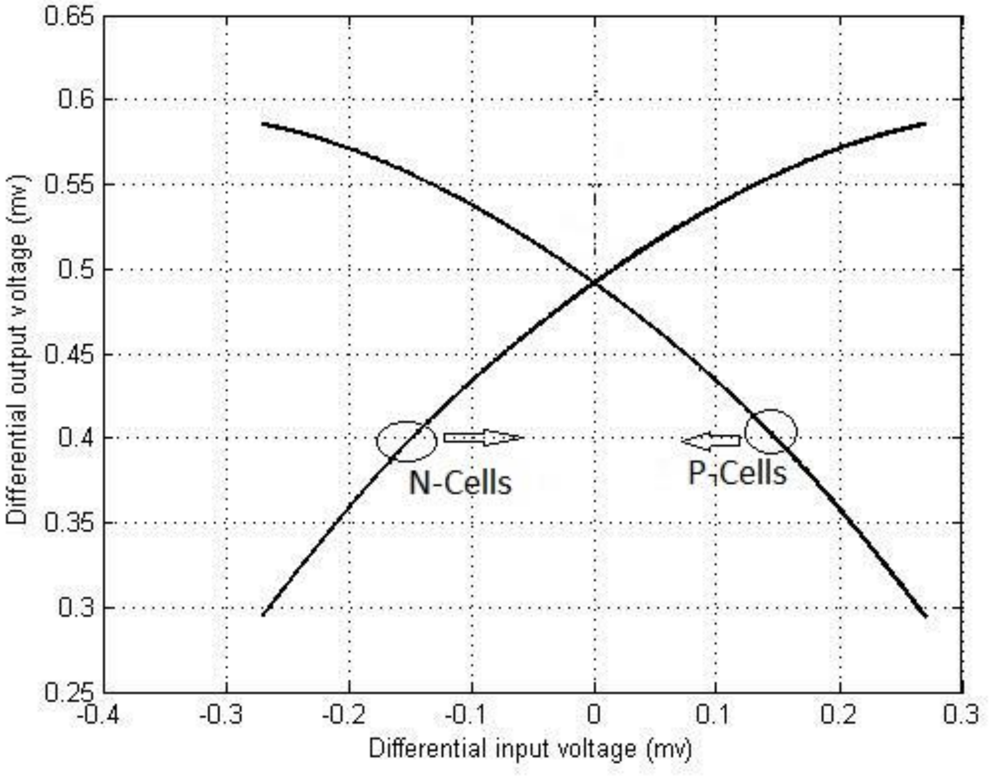


Figure 26

When input differential voltage is negative, the input voltage to P-Cells is big enough to make the digital pulse propagate fast in this delay line and triggered all delay cells in this delay line and output of all cells is “1”, while N-cells will generate thermometer code as shown in Fig. [26].

When input differential voltage is positive, the input voltage to N-Cells is big enough to make the digital pulse propagate fast in this delay line and triggered all delay cells in this delay line and output of all cells is “1”, while P-Cells will generate thermometer code.

The idea is to let the delay cells of one of the two delay lines generate thermometer code while the output of the delay cells of the other delay line is high according to differential input voltage is to increase the resolution of time-based ADC.

3.4.2 Delay cell:

Delay cell implementation is shown in Fig [27], where M1 is a controllable current source which determined by M3 and M4 transistors in DA for all delay cells in both delay lines. M2 is an NMOS switch and is triggered by the digital pulse from the prior stage. M3 is a PMOS switch which is set to "1" at the end of time window to reset delay cell. The succeeding inverter consists of M4 and M5 which supply the capacitance of the charging node of the delay cell.

Delay cell operation has two phases. During the propagation of the digital pulse through the delay line when the pulse arrives at the gate of M2 from the prior delay cell, this NMOS switch will turn on and the current source starts discharging the capacitive node of delay cell from VDD to ground, then inverter flips and the output of this delay cell is applied to the next delay cell. At the end of the time window T_s, the delay cells are charged sequentially by charging the capacitive node again through reset switch (M3) and at the same time the output of delay cell is latched to store the quantized output for the current sample.

During the pulse propagation, we should consider the charge leakage because the capacitive node is floating. Both M1 and M2 have gate lengths above the minimum length in order to decrease the charge leakage which can change the characteristics of the delay cell and lead to nonlinearity of delay cell, so the amount of charge leakage during the pulse propagation time has almost no effect on the ADC performance.

The latch at the output of each delay cell is designed to consume low power [17], but using latches brings the concern of metastability. The delay-cell outputs are high or low for most of the time but there is a chance for metastability to happen as shown in Fig [28]. This will happen if the time window T_s ends when a delay cell is in transition state so the final value of delay cell which is latched is determined randomly, this error in delay cell may happen and added to the quantization error. The transition time of delay cell must be small comparable to the minimum amount of delay of the delay cell in order to decrease metastability.

The output bits from latches in both delay lines are connected to a digital thermometer encoder to generate the final digital code.

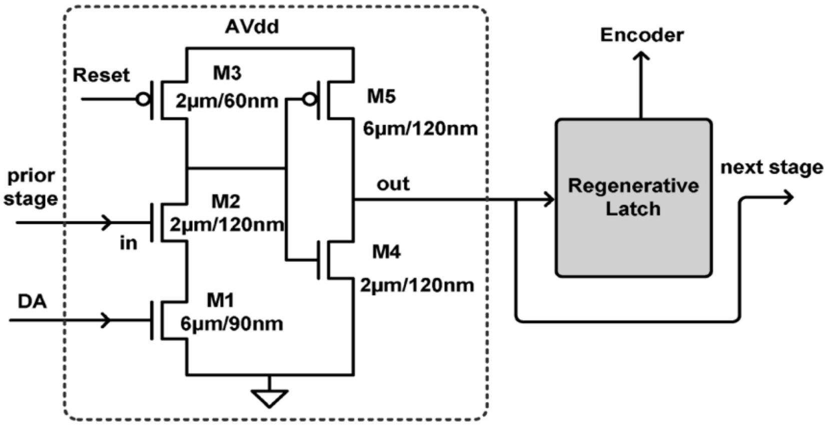


Figure 27

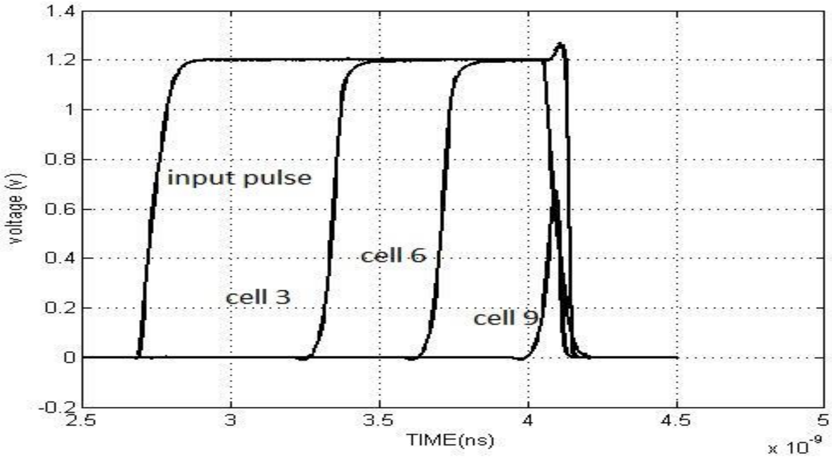


Figure 28

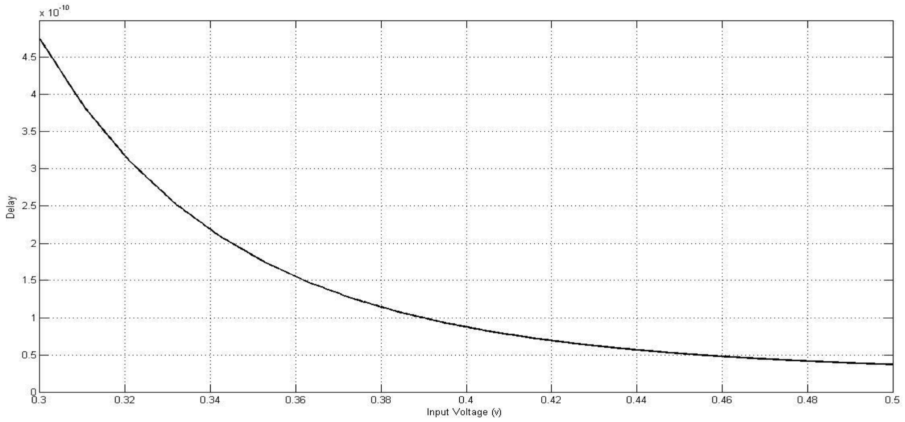


Figure 29 : Relation between input voltage to delay.

3.4.3 Clock phase generator:

There are two non-overlapping clock phases which the time-based ADC works on shown in Fig. [30], the first clock phase is the time window T_s at which the digital pulse propagate through delay line representing the input voltage. The input sets the current of M3 and M4 in DA which in turn decide the amount of delay of delay cells so propagation length vary from input to another. The second clock phase of time length T_r is the period at which the outputs of latches from both delay lines encoded to digital outputs using thermometer encoder.

To get high resolution, the larger part of the period should be dedicated to time window T_s and the remaining time of each period is time window T_r but T_r should be long enough to finish encoding and hold the digital output.

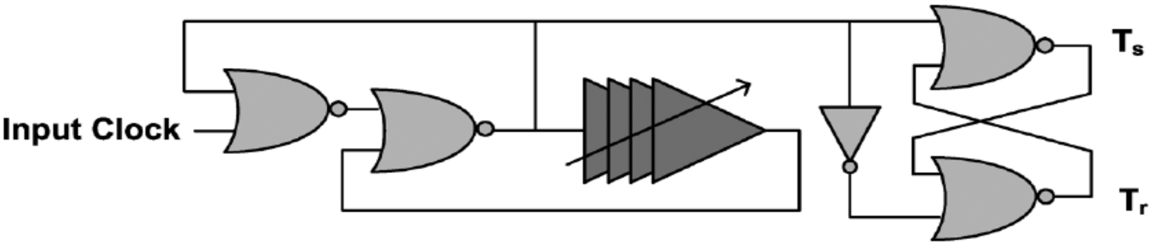


Figure 30 : clock phase implementation.

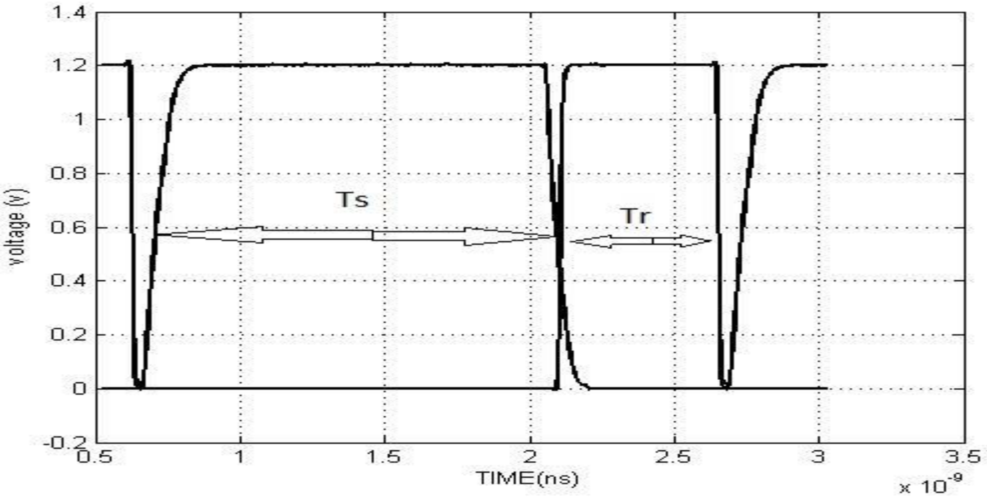


Figure 31

3.4.4 D Latch (transparent latch):

The D latch is able to store data as an internal storage shown in Fig [32], it is suitable for storing the binary information from the delay cell at the end of the time window T_S. When the enable of the D Latch is high, the binary information at the input of D Latch is transferred to Q output and the Q output will follow the input as long as enable is high. While when the enable is low, the binary information at the input at the time of transition is stored at the Q output until the enable is high again.

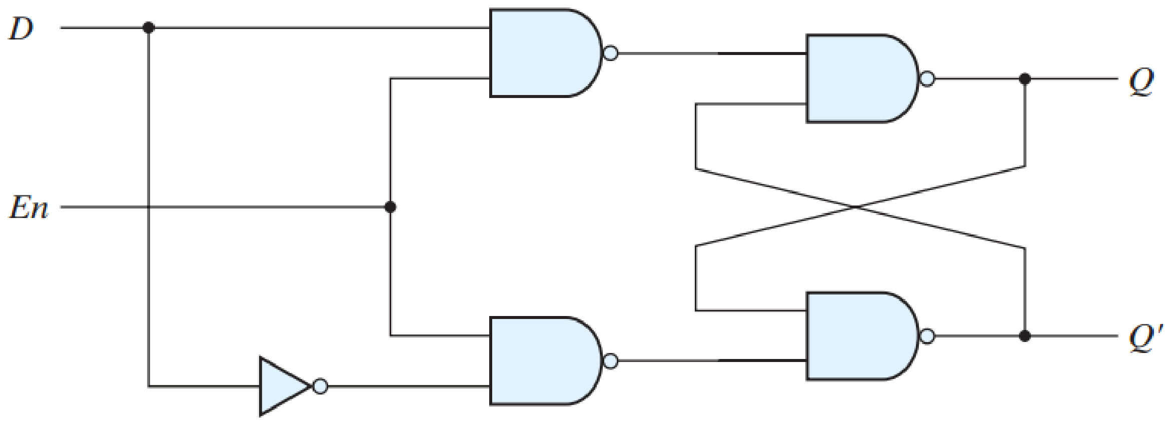
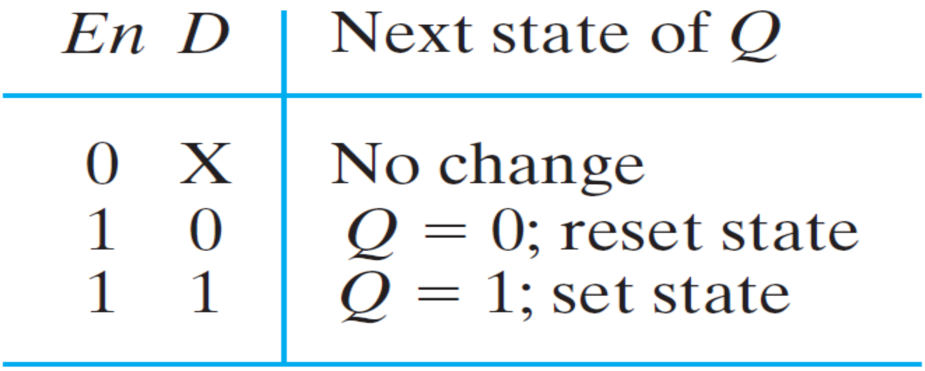


Figure 32 : D-Latch



En	D	Next state of Q
0	X	No change
1	0	$Q = 0$; reset state
1	1	$Q = 1$; set state

3.4.5 5-bits Thermometer encoder:

The proposed encoder is designed in a dynamic logic style, for ultra-low power consumption and can convert the thermometer code into binary code without any intermediate stage which decrease the number of transistors used in the design [19]. The P-latches and N-latches outputs represents the input signal in a thermometer code.

Pseudo dynamic CMOS circuit which consists of a PMOS transistor, pull down network of NMOS transistors, and inverter is not preferable as it has static power dissipation [18]. During the precharging operation, the output node is charged to a logic high through the PMOS transistor while the output is determined depending on the logic function implemented in the pull down network. When the pull up and pull down networks are on at the same time so the current find a path from VDD to ground which leads to static power dissipation and this happen if the clock signal is low and the NMOS transistors in the pull down network are on. This is a disadvantage of pseudo dynamic CMOS based circuit encoder.

To achieve ultra-low power dissipation without static power dissipation, the proposed encoder is done using dynamic CMOS logic style, where its operation is divided into two phases, precharging and evaluation.

During precharging phase (when CLK=0) output node is precharged to VDD by PMOS transistor while the pull down network of NMOS transistors are off because the NMOS which connected to CLK is off, this leads to minimizing the static power dissipation.

During evaluation phase (when CLK=1), the pull up transistor will be off while the NMOS transistor which connected to the CLK is on and the output is evaluated according to the NMOS transistors condition in the pull down network.

An advantage of dynamic CMOS logic over static CMOS is having faster switching speed because the number of transistors needed in implementation of dynamic CMOS logic is less than the number of transistors needed in implementation of static CMOS.

3.4.5.1 Truth Table:

The truth table gives the relation between the thermometer code and the corresponding 5-bit binary code.

B4	**B3**	**B2**	**B1**	**B0**	**Thermometer Code**
0	0	0	0	0	0000000000000000000000000000000
0	0	0	0	1	0000000000000000000000000000001
0	0	0	1	0	0000000000000000000000000000011
0	0	0	1	1	0000000000000000000000000000111
0	0	1	0	0	0000000000000000000000000001111
0	0	1	0	1	0000000000000000000000000011111
0	0	1	1	0	0000000000000000000000000111111
0	0	1	1	1	0000000000000000000000001111111
0	1	0	0	0	0000000000000000000000011111111
0	1	0	0	1	0000000000000000000000111111111
0	1	0	1	0	0000000000000000000001111111111
0	1	0	1	1	0000000000000000000011111111111
0	1	1	0	0	0000000000000000000111111111111
0	1	1	0	1	0000000000000000001111111111111
0	1	1	1	0	0000000000000000011111111111111
0	1	1	1	1	0000000000000000111111111111111
1	0	0	0	0	0000000000000001111111111111111
1	0	0	0	1	0000000000000011111111111111111
1	0	0	1	0	0000000000000111111111111111111
1	0	0	1	1	0000000000001111111111111111111
1	0	1	0	0	0000000000011111111111111111111
1	0	1	0	1	0000000000111111111111111111111
1	0	1	1	0	0000000001111111111111111111111
1	0	1	1	1	0000000011111111111111111111111
1	1	0	0	0	0000000111111111111111111111111
1	1	0	0	1	0000001111111111111111111111111
1	1	0	1	0	0000011111111111111111111111111
1	1	0	1	1	0000111111111111111111111111111
1	1	1	0	0	0001111111111111111111111111111
1	1	1	0	1	0011111111111111111111111111111
1	1	1	1	0	0111111111111111111111111111111
1	1	1	1	1	1111111111111111111111111111111

Table 2 : Truth Table of Thermometer to Binary Encoder

3.4.5.2 Design equations:

$$B4 = I15$$

$$B3 = I7.\overline{I15} + I23$$

$$B2 = I3.\overline{I7} + I11.\overline{I15} + I19.\overline{I23} + I27$$

$$B1 = I1.\overline{I3} + I5.\overline{I7} + I9.\overline{I11} + I13.\overline{I15} + I17.\overline{I19} + I21.\overline{I23} + I25.\overline{I27} + I29$$

$$\begin{aligned} B0 = I0.\overline{I1} + I2.\overline{I3} + I4.\overline{I5} + I6.\overline{I7} + I8.\overline{I9} + I10.\overline{I11} + I12.\overline{I13} + I14.\overline{I15} \\ + I16.\overline{I17} + I18.\overline{I19} + I20.\overline{I21} + I22.\overline{I23} + I24.\overline{I25} + I26.\overline{I27} \\ + I28.\overline{I29} + I30 \end{aligned}$$

3.4.5.3 Design Implementation:

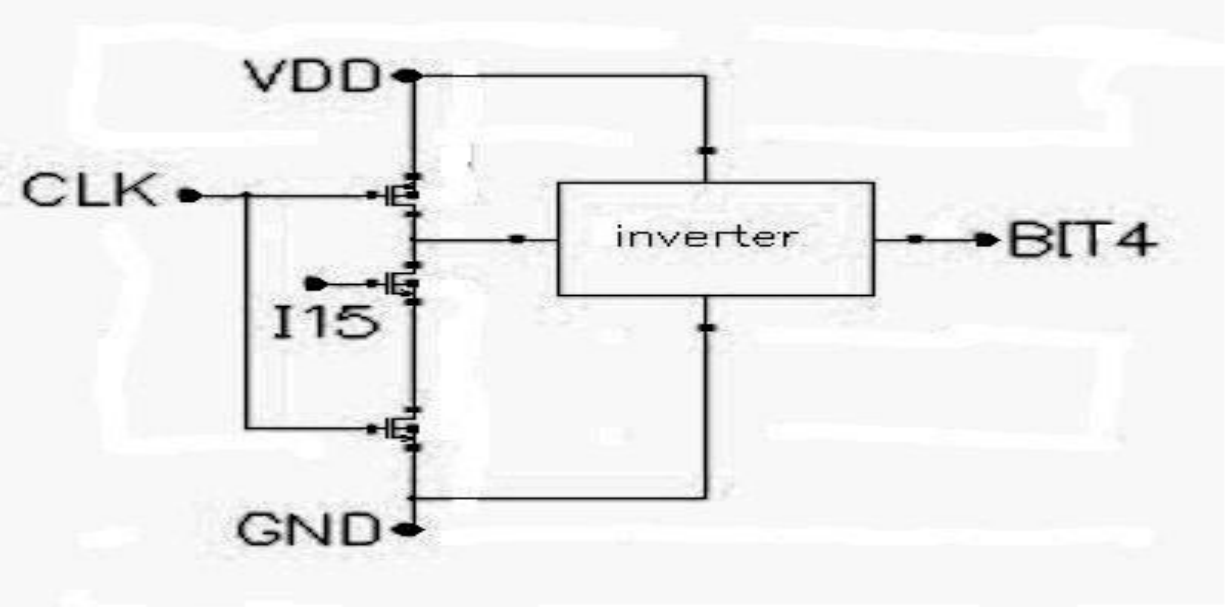


Figure 33 : Binary Code Bit 4 Generation Circuit

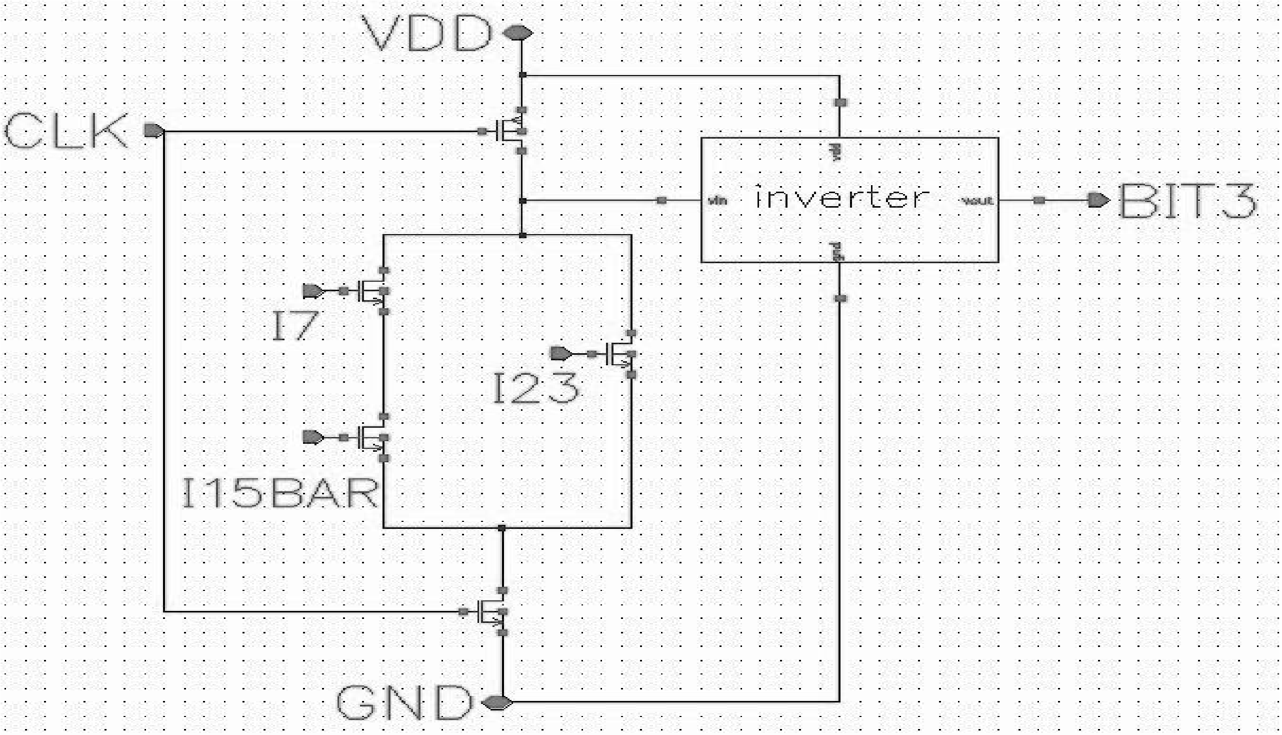


Figure 34 : Binary Code Bit3 Generation Circuit

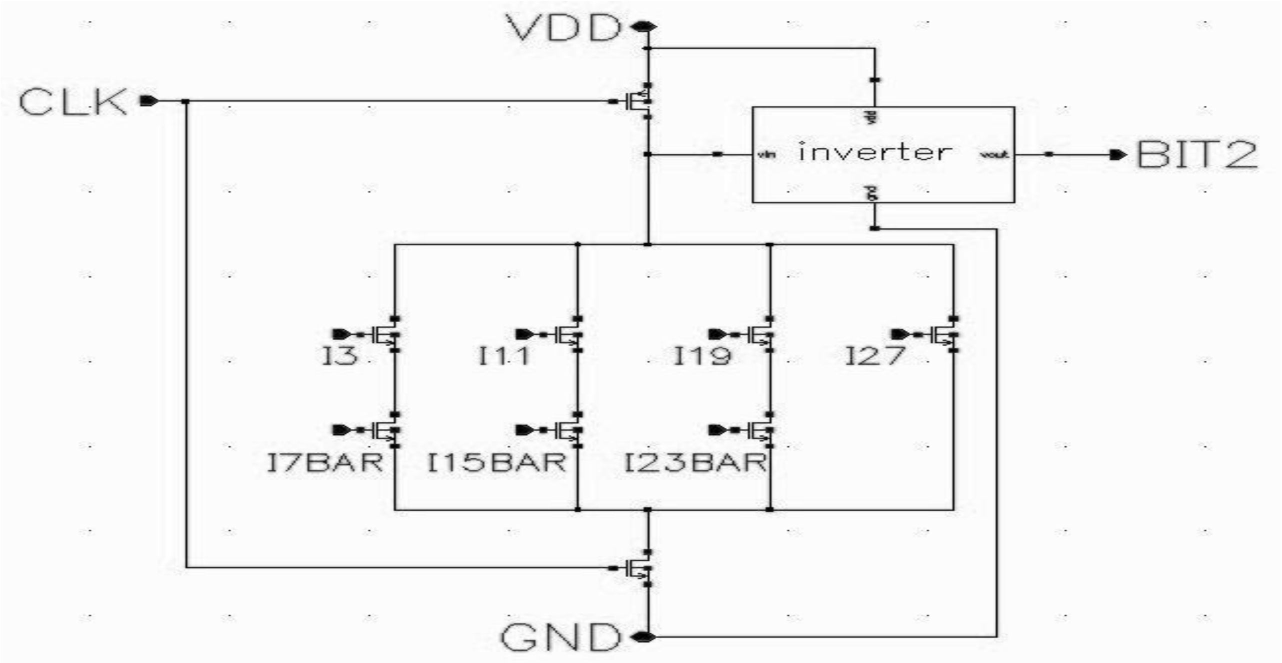


Figure 35 : Binary Code Bit2 Generation Circuit

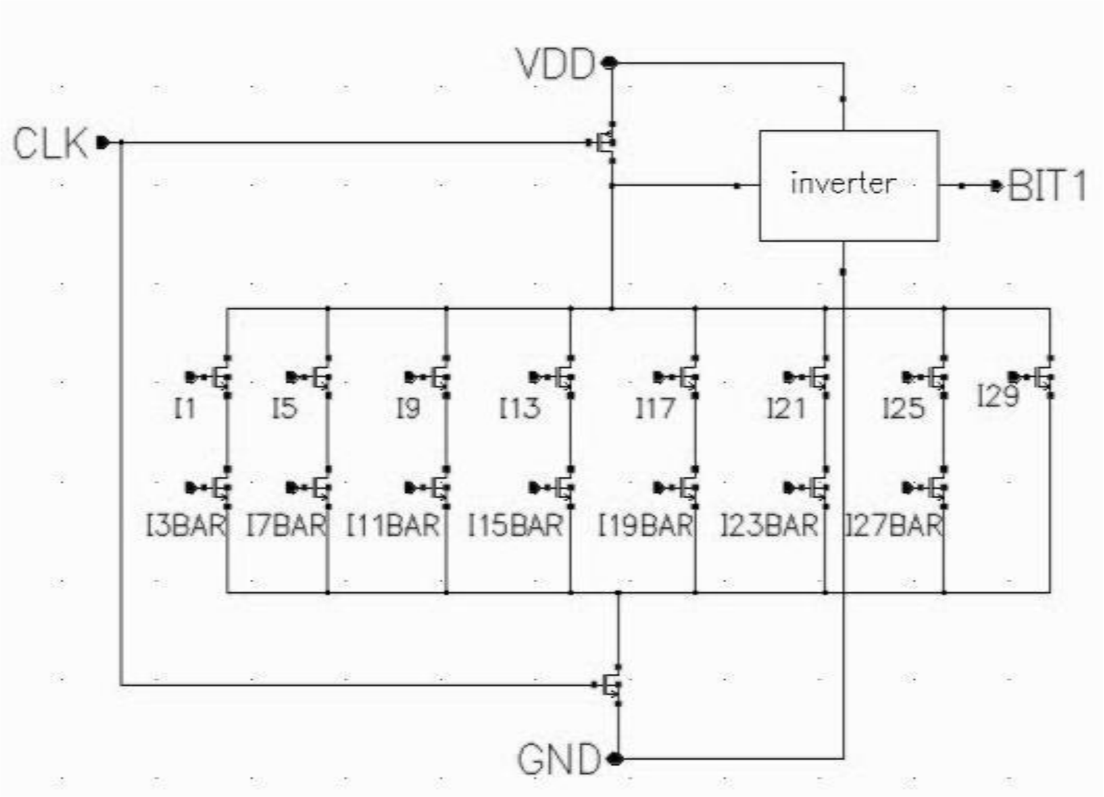


Figure 36 : Binary Code Bit 1 Generation Circuit

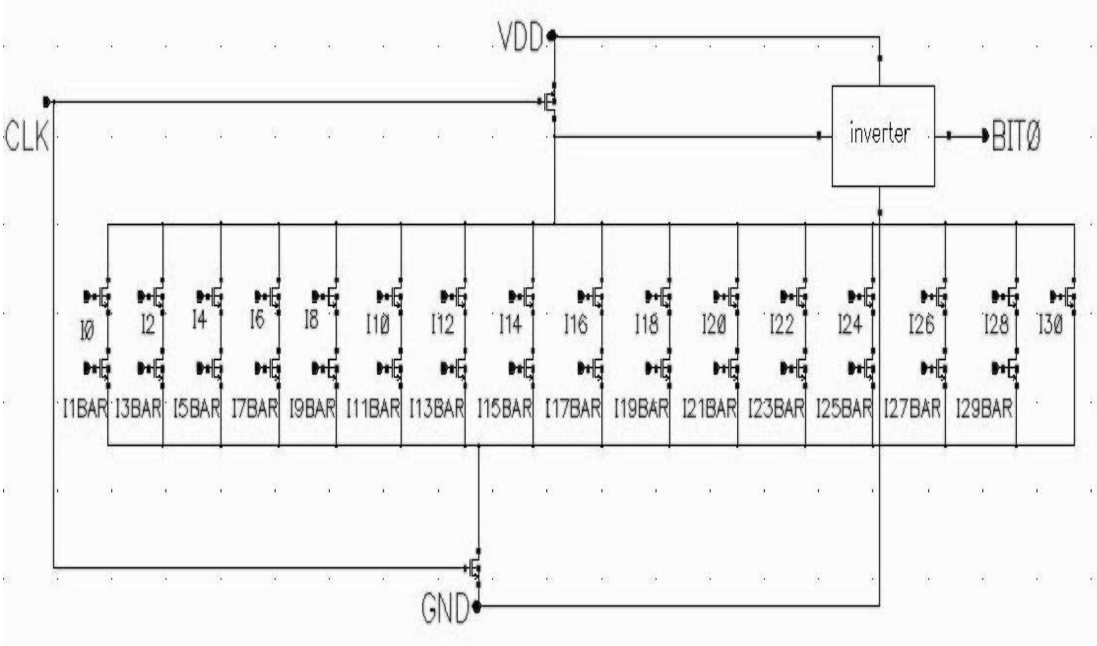
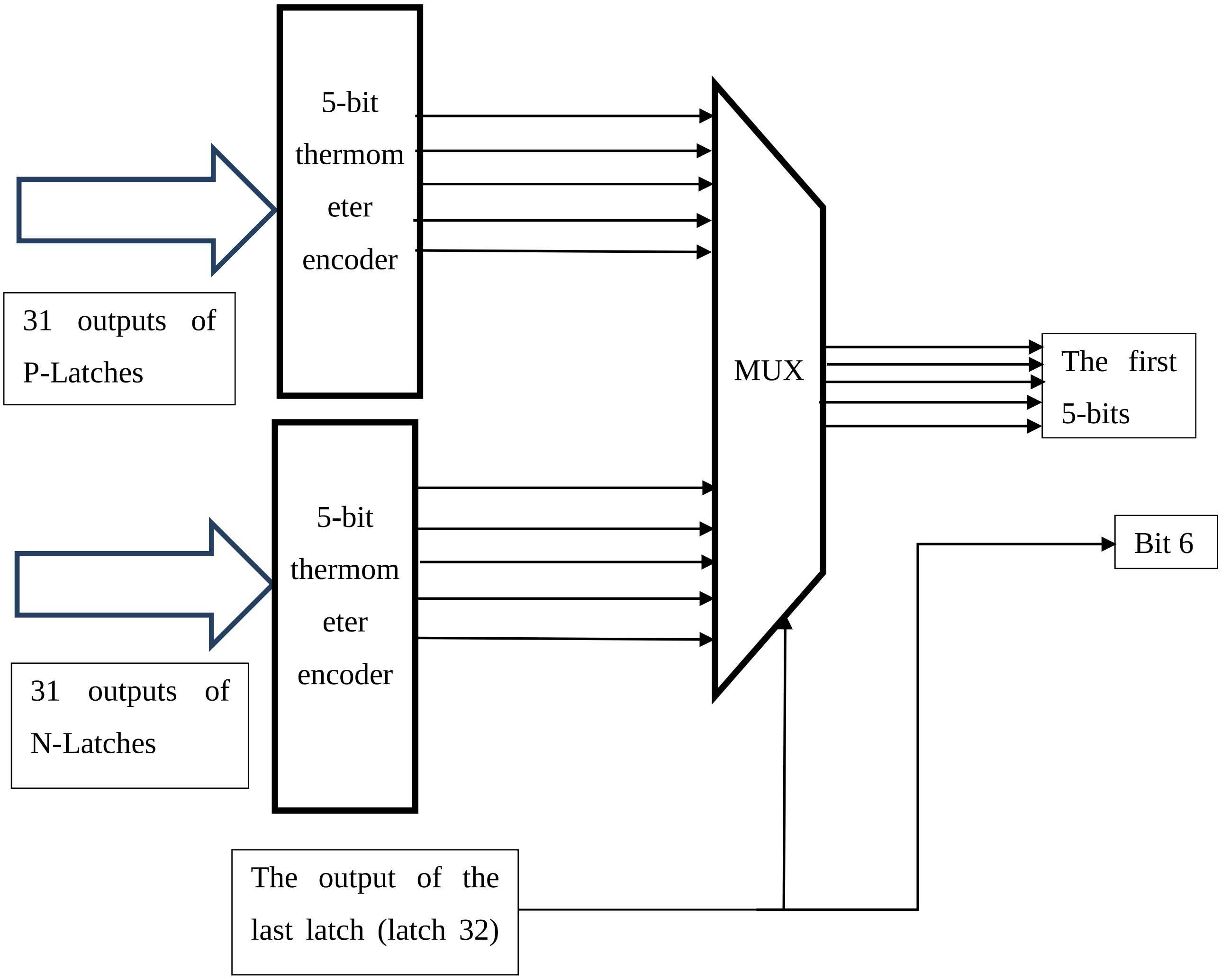


Figure 37 : Binary Code Bit 0 Generation Circuit

3.5 Encoding method of the outputs of P-Latches and N-latches:


5-bit thermometer encoder
31 outputs of P-Latches
5-bit thermometer encoder
31 outputs of N-Latches
MUX
The first 5-bits
Bit 6
The output of the last latch (latch 32)

In our design we use two 5-bit thermometer encoder, one for each delay line which consists of 32 delay cells, followed by a multiplexer. The selection of the multiplexer is set by the output of the last latch (latch 32) of the N-latches. The first 5 bits of the ADC digital output is generated from the multiplexer while the last bit is generated from the output of the last latch (latch 32) of the N-latches.

Chapter 4: Optimization for First TADC

4.1 Introduction:

We will introduce a modification to the time-based ADC design discussed in chapter (3). The main blocks of the design is the same such as delay cell and DA. The number of delay cells is decreased by forming a loop and take into account the number of the rotations of the digital pulse to achieve the same resolution.

4.2 Operation of ADC:

The delay line consists of 8 delay cells, instead of 32 delay cells. A two bit up saturated counter counts the number of rotation of the digital pulse through delay line. During the time window (T_s) if the digital pulse reaches the end of the delay line (8th delay cell), reset needed to be done for all delay cells to start a new loop. Consequently, resetting delay cells is done at the end of time window T_s and when the digital pulse reaches the 8th delay cell as shown in Fig [36].

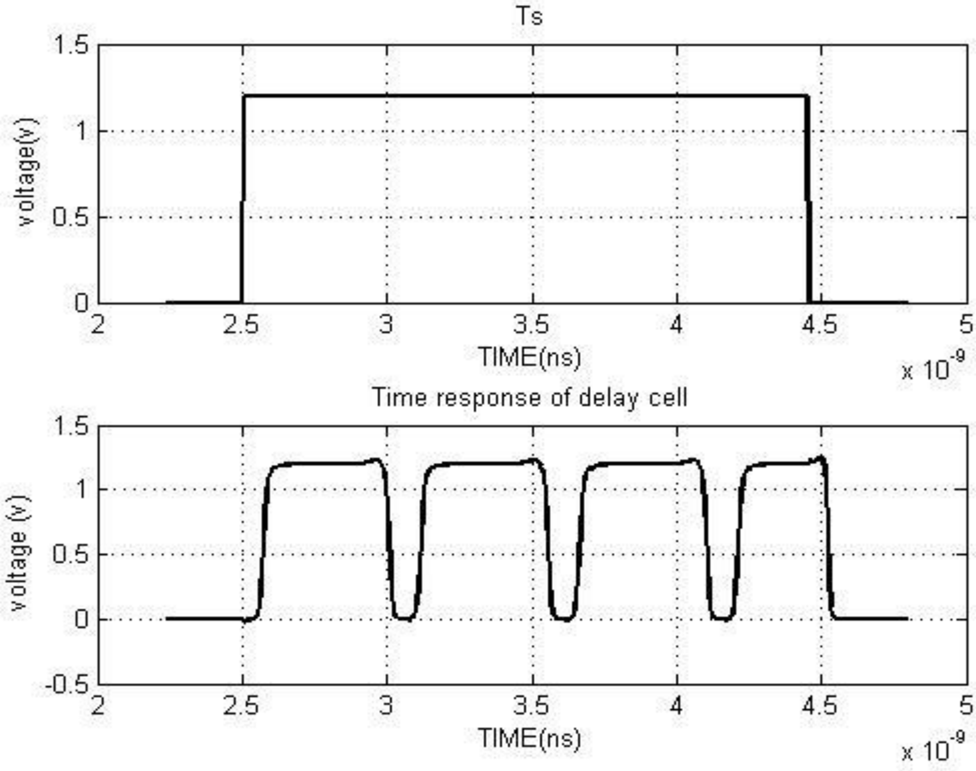


Figure 38

4.3 Implementation of TADC:

This is shown as in Figure [37]

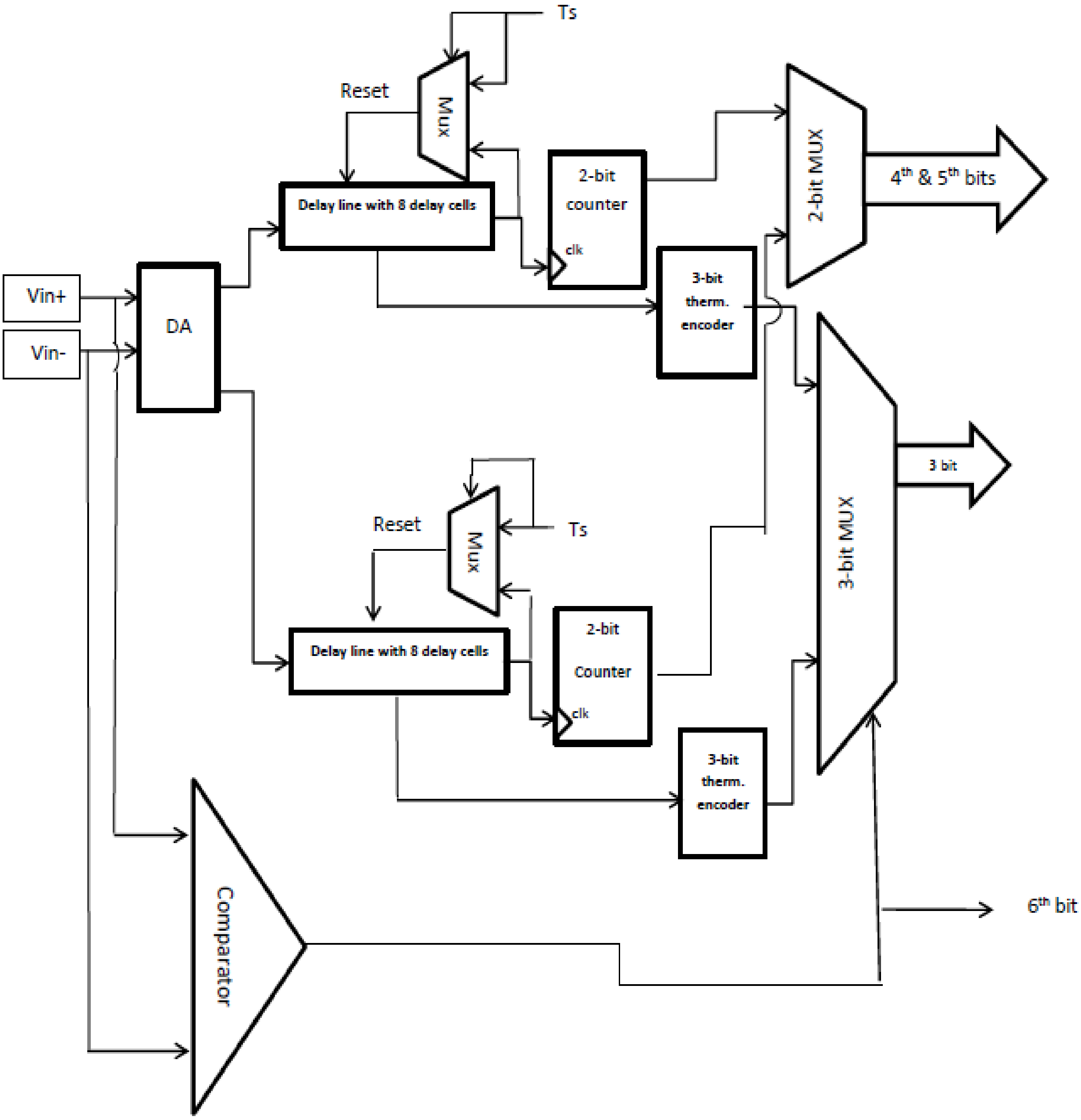


Figure 39

4.4 Circuit Design:

4.4.1 2-bit saturated up counter:

The first step is to construct the truth table of a synchronous circuit - its Transition Table. The transition table shows the state output values after the clock pulse (next) as a function of the input and state output values before the clock pulse (now). Since for a D type flip-flop the output (Q) after the clock pulse is equal to the input (D) before the clock pulse, the transition table becomes a simple input/output truth table.

Now		Next	
Q1	Q2	Q1	Q2
0	0	0	1
0	1	1	0
1	0	1	1
1	1	1	1

Q2

		0	1
Q1	0	0	1
	1	1	1

D1 (Q1 (next))

$$D1 = Q1 + Q2.$$

Q2

		0	1
Q1	0	1	0
	1	1	1

D2 (Q2 (next))

$$D2 = Q1 + \overline{Q2}.$$

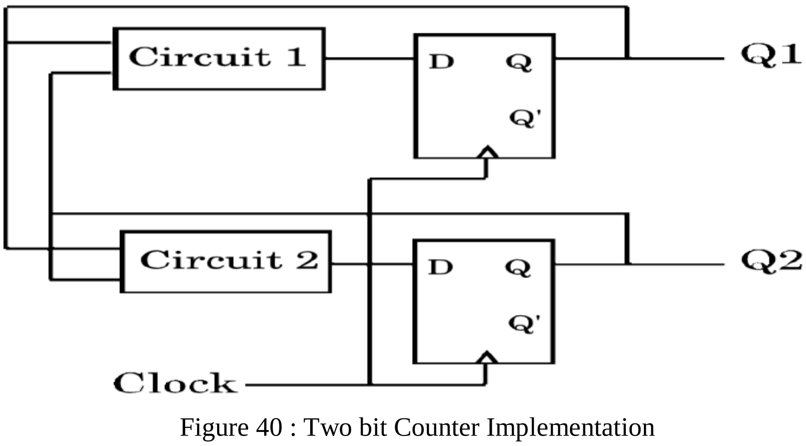


Figure 40 : Two bit Counter Implementation

4.4.2 Magnitude Comparator:

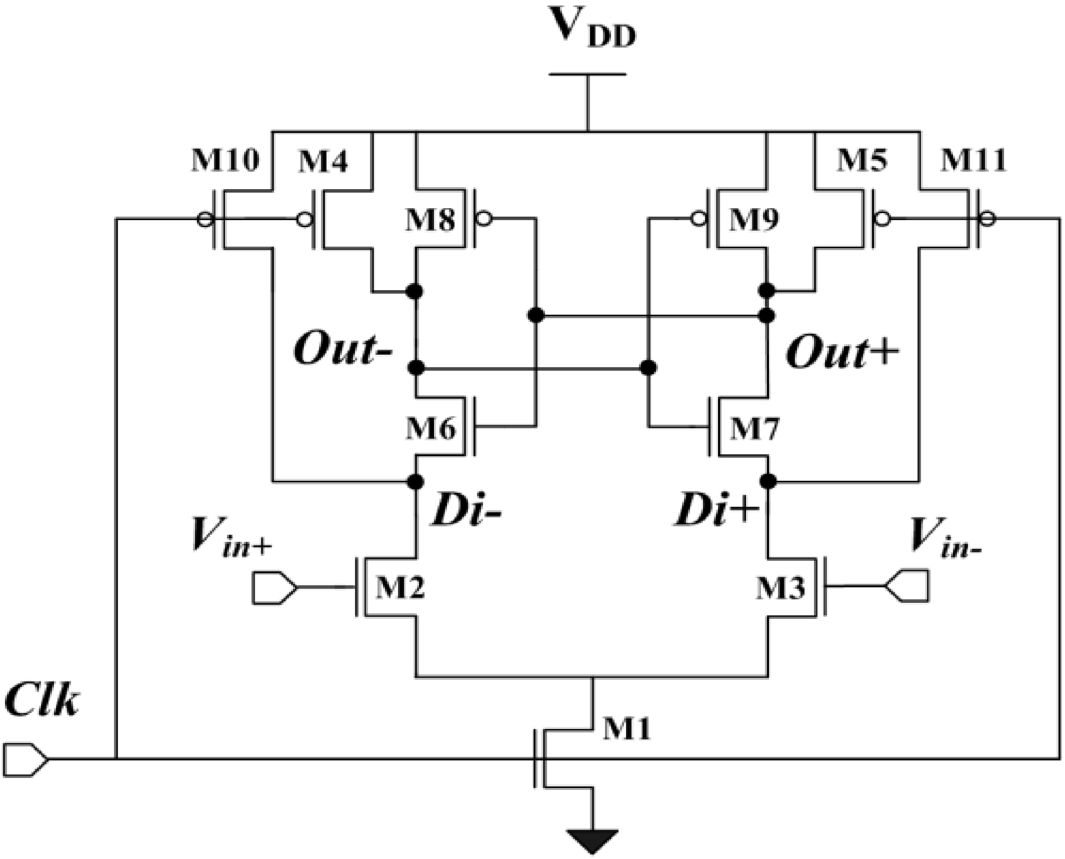


Figure 41 : Implementation of Magnitude Comparator

The operation of the magnitude comparator shown in Fig. [39] can be divide into two phases. First, the reset phase (clk=0), the nodes out+ and out- are reset to VDD through the reset transistors M4 and M5. Second, the evaluation phase (clk=1), the NMOS transistor M1 is turned on. The input transistors M2 and M3 starts to discharge Di+ and Di- nodes voltage with a different time rate proportional to the each applied input voltage from VDD to 0V. When one of Di+ or Di- nodes voltages drops around VDD−Vtn, then the NMOS transistors of the cross-coupled inverters M6 and M7 turn on. When one of Out+ or Out- nodes voltage reaches around VDD−|Vtp|, the PMOS transistors of the inverters also turn on and enabling the regeneration of a small differential voltage ΔVin to a full swing differential output.

Chapter 5: Voltage-to-Time Converter in 65 nm CMOS

This chapter will present an analysis of the voltage-to-time converter (VTC) designed in 65nm CMOS process by TSMC including an introduction of the time mode signal processing, an explanation of the physical operation of the circuit, a derivation of the output delay using hand-analysis, optimization of the output linearity, a design procedure for both single-ended and differential VTC variations. The differential input is composed of a DC bias voltage coupled to each of the two RF differential signals through bias tees (off-chip). The range of the VTC is tunable to correct for process and voltage, and the tunability can also be used to operate at clock frequencies other than 1GS/s.

In the TADCs, the conversion from an analog signal to a digital representation is done on two steps using time mode signal processing (TMSP) as an intermediate step. The first step, known as a voltage-to-time converter (VTC) which a series of pulses is delayed. The delay of each pulse depends on the analog input at the moment the pulse was created. In the second step the time representation is processed and then converted into a digital representation using time to digital converter (TDC).

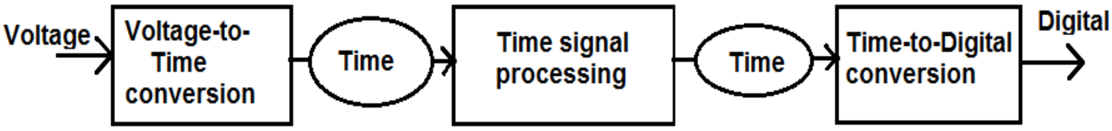


Figure 42 : Analog to Digital Conversion via Time Mode Signal Processing

5.1 Time-mode signal processing

TMSP may be defined as the detection, storage, and manipulation of sampled analog information using time-difference variables. Moreover, we define a time-difference variable ΔT, as the quantity of time between two events.

5.1.1 Time-difference

In this work, a time-difference variable refers to the time interval between two digital clock edges. Figure [43] illustrates two digital clock signals, $\varphi 1$ and $\varphi 2$, and the time-difference variable,ΔT represented as the time interval between the two clock edge transition times t1and t2.

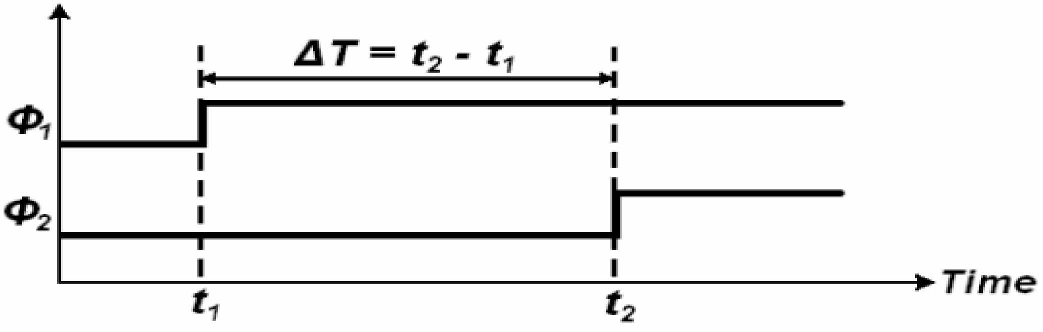


Figure 43 : Timing Diagram illustrate time-difference bet, 2 input clock edges

5.1.2 Voltage-controlled delay unit

A voltage-controlled delay unit is a circuit that proportionally delays an input time event with respect to a sampled input voltage. Figure [44] illustrates the block diagram of the voltage controlled delay unit. The implementation of it has two inputs; a reference even $\emptyset_{CLK}$, and the sampled input voltage $V_{IN}(n)$that controls the delay of $\emptyset_{CLK}$.

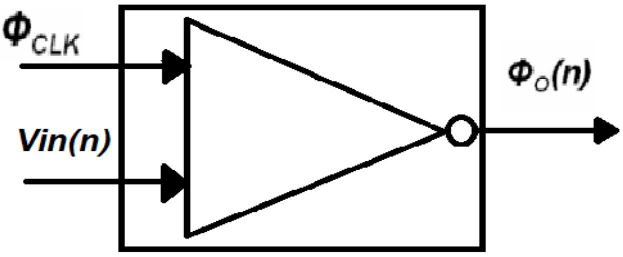


Figure 44 : Voltage Controlled delay unit block diagram

The voltage-controlled delay unit output is the time-difference variable $\Delta T_o(n)$. The time difference variable $\Delta T_o(n)$ contains the signal information. Figure [43] shows an example timing diagram of the voltage-controlled delay unit operation. The output $\Delta T_o(n)$ is proportional to the input voltage $V_{IN}(n)$ with the voltage-to-time conversion factor G∅.

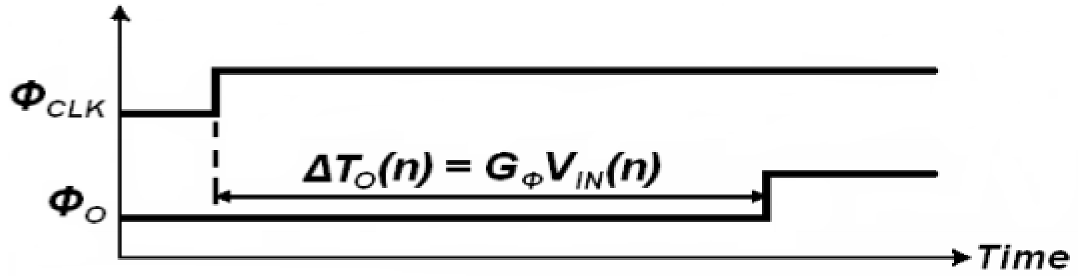


Figure 45 : Voltage controlled unit Timing Diagram

The relationship between voltage $V(t)$and time t is given by

$$I(t) = C\frac{dV(t)}{dt}$$

Where C is some nodal capacitance and $I(t)$ is current. Time can be obtained as the dependent variable from equation (previous), such as

$$d(t) = \frac{C}{I(t)}dV(t)$$

From equation (previous), there are two ways to introduce time-mode variables: via voltage control or current control.

The equation describing the conversion process is given by

$$\Delta T_o(n) = \frac{C}{I_{IN}}V_{IN}(n) = G_\varphi V_{IN}(n)$$

Where the voltage-to-time conversion factor $G_\varphi$is equal to C/I_{IN}.

A common implementation for this classification of VCDU is often referred to as a current-starved inverter [20] – [26].

5.2 Current-starved inverter

The process of Voltage-to-time conversion is facilitated using a voltage-controlled delay unit. The voltage-controlled delay unit is referred to as a current-starved inverter, as shown in Figure [46]. Current-starved inverter is an implementation for controlling the current.

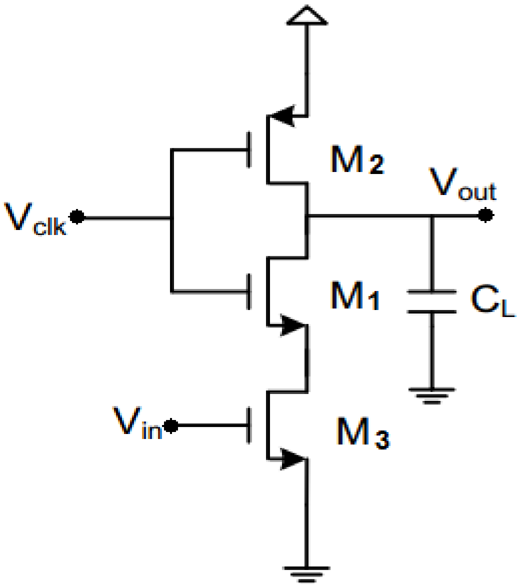


Figure 46 : Current Starved Inverter

Current starved inverter cell consists of three transistors. The upper two transistors (M1 and M2) form the standard inverter, while M3 is placed to control the flow of current in the inverter when V_{CLK} is equal toVDD, the discharging current of CL.

The input voltage V_{IN} controls the delay of the falling edge of the clock signal,V_{CLK} through the inverter (Transistors M1 and M2) by controlling the discharging current of transistor M3.

The conversion is done by using the analog input signal to control the rate at which the capacitor connected to the output of the inverter is charged or discharged.

5.3 The analysis of the VTC (Theory) operation

Voltage-to-time conversion process is the process of sampling an analog voltage and converting it into an analog time-difference variable.

A schematic diagram of the VTC is shown in Fig. [47]

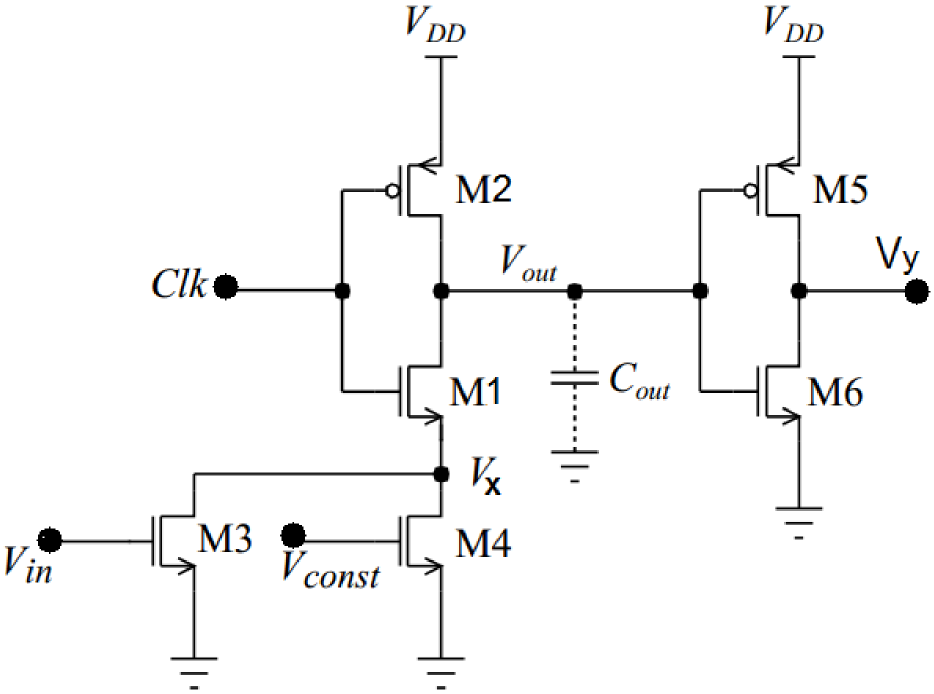


Figure 47: VTC schematic

5.3.1 Explanation of Operation

The VTC operation can be explained by assuming idealized transistor models with the addition of parasitic capacitor C_{out}, which is mainly formed from the drain capacitances of M1 and M2 and the gate capacitances of M5 and M6. It is assumed that all other node capacitances are significantly less than C_{out}.

In the figure, M5-M6 forms a standard CMOS inverter, while M1-M4 makes up a current-starved inverter. The gate input to M3 is the input signal to the VTC V_{IN}. The gate input to M4,$Vconst$ is a DC bias voltage used to tune the gain and linearity of the VTC.

Since the M1-M2 inverter has starving devices between M2 and ground but not between M1 and VDD, rising edges of CLK will be slowed down by the starved inverter, depending on the value of V_{IN}. However, falling edges of CLK will be passed through to V_{out}.

The VTC operation summary is as follows: When a rising edge occurs on $CLK(t)$, $V_{out}(t)$ begins to ramp downwards from VDD at a rate dependent on V_{IN}. When this ramping signal reaches the threshold of the M5-M6 inverter, a rising edge is triggered on the inverter output.

The delay on this edge, and how it varies withV_{IN}, is what we are interested in analyzing. All device lengths and threshold voltages are assumed equal.

5.3.1.1 Sequence of operation

0. Initial Conditions

In steady state, $CLK(t)$ and $V_x(t)$ equal to zero and $V_{out}(t)$equals toVDD. In the starved inverter, devices M1, M3 and M4 are in deep triode while M2 is in cut-off. In the second inverter, M5 is in cut-off while M6 is in deep triode.

1. Rising edge on clock input

Since $CLK(t)$goes high, M2 enters cut-off. M1 turns on, entering the saturation region. M3 and M4 are still in deep triode so conduct very little current. Charge stored on C_{out} flows through M1.The result is a rapid voltage increase of $V_x(t)$.

2. M3 and M4 enter saturation

When $V_x(t)$ is increasing rapidly, it can be assumed that M3 and M4 enter saturation mode simultaneously. Therefore, they limit the current flowing through M2 to a constant amount, I_{max}. This causes V_{out} to linearly ramp down at a constant rate.

The fixed current can be calculated as

$$I_{max} = I_3 + I_4$$

$$= \frac{1}{2} KP_3 [V_{in} - V_T]^2 + \frac{1}{2} KP_4 [V_{const} - V_T]^2$$

Where

$$KP_3 = \mu C_{ox} \frac{W3}{L} \qquad \text{and} \qquad KP_4 = \mu C_{ox} \frac{W4}{L}$$

and the ramp rate is

$$R_{ramp} = \frac{C_{out}}{I_{max}}$$

The constant current through M1 sets the gate-source voltage of M1 to a constant value. Since the gate is held constant at VDD, the voltage of $V_x(t)$ is fixed at

$$V_{x,max} = V_{DD} - V_{GS1}$$

$$= V_{DD} - (\sqrt{\frac{W_3[V_{in}-V_T]^2+W_4[V_{const}-V_T]^2}{W_2}} + V_T)$$

3. M1 enters triode

When $V_{DS1} \leq V_{GS1}-V_T$, which will correspond to $V_{out}(t)$ dropping by VT. After this point, V_{GS2} will no longer be constant, causing V_x to decrease. However, the current through M2 will still be I_{max}, so the linear ramp on $Vout$ is maintained. It is expected that $V_{out}(t)$ will reach the threshold of the M5-M6 inverter during this step, triggering a rising edge on the inverter output. $V_{out}(t)$ will no longer affect the VTC output after this switching point.

4. M3 and M4 enter triode

When $V_x(t)$ reaches the greater of $V_{IN} - V_T$ and V_{const} - V_T, either M3 or M4 enters triode, the current through M2 will begin to decrease, and the ramp on V_{out} will no longer be linear. It can be concluded that Vin and V_{const} should each be no higher than $\frac{1}{2}V_{DD} - V_T$ to ensure that M3 and M4 remain saturated until after the M5-M6 inverter is triggered.

5. System returns to steady state

$V_{out}(t)$And $V_x(t)$continue to ramp down until they reach 0.

5.3.2 Derivation of VTC delay

Using the previous analysis, the VTC delay can be derived. This will be the delay from the rising clock edge of M3 and M4 when enter saturation to the point where V_{out} reaches the threshold level for the M5-M6 inverter, which can be estimated as $\frac{1}{2}V_{DD}$. The VTC delay will simply be the time it takes for V_{out} to ramp linearly from V_{DD} to $\frac{1}{2}V_{DD}$.

The resulting expression is

$$delay = R_{ramp}(\ V_{DD} - \frac{1}{2}V_{DD})$$

$$= \frac{\frac{1}{2}V_{DD}C_{out}}{\frac{1}{2}KP_3[V_{in} - V_T]^2 + \frac{1}{2}KP_4[V_{const} - V_T]^2}$$

$$= \ \frac{V_{DD}C_{out}}{KP_3[V_{in} - V_T]^2 + KP_4[V_{const} - V_T]^2}$$

We can express $V_{in} = V_B + V_{amp}\sin(t)$ where V_B is the DC bias voltage is and V_{amp} is the amplitude of the sinusoidal input signal.

The optimum SNDR can be calculated:

1) For single-ended VTC optimization

$$SNDR_{single,opt} = (16(\frac{V_B - V_T}{V_{amp}})^2 - 3)^2$$

2) For differential VTC optimization

$$SNDR_{diff,opt} = \frac{50}{13}(4(\frac{V_B - V_T}{V_{amp}})^4 - 1)^2$$

Equations show that the linearity will be improved by making V_{amp} small and using a large overdrive voltage $V_B - V_T$.

5.3.3 Range and Absolute Delay - Single-Ended and Differential VTC

The main considerations for the design of VTC are the total range and the maximum absolute delay. The total range (t_{range}) is defined as the difference between the output delay for maximum input and for minimum input. This range should be as large as possible to relax the resolution requirements on the TDC, but it is limited by the clock period and pulse width being used. Since the maximum input is $V_B + V_{amp}$ and the minimum input is$V_B - V_{amp}$, the range can be expressed analytically as

1) For single-ended VTC optimization

$$t_{range} = \frac{V_{DD}C_{out}}{KP_3[V_B - V_{amp} - V_T]^2 + KP_4[V_{const} - V_T]^2} - \frac{V_{DD}C_{out}}{KP_3[V_B + V_{amp} - V_T]^2 + KP_4[V_{const} - V_T]^2}$$

It is simplified to

$$t_{range} = \frac{V_{DD}C_{out}V_{amp}}{4KP_3[V_B - V_T]^3}$$

1) For differential VTC optimization

$$t_{range} = \frac{2V_{DD}C_{out}}{KP_3[V_B - V_{amp} - V_T]^2 + KP_4[V_{const} - V_T]^2} - \frac{2V_{DD}C_{out}}{KP_3[V_B + V_{amp} - V_T]^2 + KP_4[V_{const} - V_T]^2}$$

It is simplified to

$$t_{range} = \frac{2V_{DD}C_{out}V_{amp}}{KP_3[V_B - V_T]^3}$$

The other important consideration is maximum absolute delay (t_{max}) which is defined as the VTC delay for the smallest possible input. Since only the falling edge is delayed, when the VTC delay increases it shrinks the pulse width of the VTC output. It must therefore be limited to a safe level for robust VTC operation. We will define the nominal absolute delay (t_{abs}) as being that produced by the common-mode output, $V_{IN} = V_B$.

1) For single-ended VTC optimization

$$t_{abs} = \frac{V_{DD}C_{out}}{KP_3[V_B - V_T]^2 + KP_4[V_{const} - V_T]^2}$$

The range can be expressed

$$t_{range} = \frac{V_{amp}t_{abs}}{(V_B - V_T)}$$

To find the maximum delay, we will assume the VTC is relatively linear.

$$t_{max} = t_{abs} + \frac{1}{2} t_{range}$$

2) For differential VTC optimization

$$t_{abs} = \frac{V_{DD}C_{out}}{KP_3[V_B - V_T]^2 + KP_4[V_{const} - V_T]^2}$$

$$= \frac{V_{DD}C_{out}}{2KP_3[V_B - V_T]^3}$$

The range can be expressed

$$t_{range} = \frac{4V_{amp}t_{abs}}{(V_B - V_T)}$$

To find the maximum delay, we will assume the half- VTC is relatively linear on its own.

$$t_{max} = t_{abs} + \frac{1}{4} t_{range}$$

5.3.4 Design Procedure

The first step is to determine the fixed parameters and design specifications. First, the technology-dependent parameters μC_{ox} and V_T can be estimated from the simulator using standard techniques.

The value of C_{ox}, is highly dependent on the input capacitance of the inverter following the VTC. The size of this inverter can be the same size as VTC devices M1 and M2, or it can be made larger. The larger inverter will increase C_{ox}, which the current needed will be increased to achieve a given slope on the output node during VTC ramping operation. This larger current requires larger widths of M3 and M4, which will improve their matching and noise performance at the expense of increased power consumption.

Based on design specifications, we choose V_{amp}. The desired range and maximum absolute delay can be chosen based on clock period (T_{clk}).

Based on the minimum acceptable pulse width, the maximum absolute delay (t_{max}) can be determined. If the standard 50% duty cycle clock is used, the minimum pulse width will be $\frac{1}{2}T_{clk} - t_{max}$. However, the clock duty cycle can be adjusted prior to the VTC in order to be able to adjust the absolute delay.

Based on the optimization equations, the unknown parameters (V_B,$V_{const,}$ KP_3 and KP_4) can be obtained.

First, V_B is set to achieve the desired range and maximum absolute delay.

1) For single-ended VTC optimization

$$V_B = V_T + \frac{V_{amp}\left(t_{max} - \frac{1}{2}t_{range}\right)}{t_{range}}$$

$$= V_T + V_{amp}(\frac{t_{max}}{t_{range}} - \frac{1}{2})$$

2) For differential VTC optimization

$$V_B = V_T + \frac{4V_{amp}\left(t_{max} - \frac{1}{4}t_{range}\right)}{t_{range}}$$

$$= V_T + 4V_{amp}(\frac{t_{max}}{t_{range}} - \frac{1}{4})$$

The next step is to find KP_3

1) For single-ended VTC optimization

$$KP_3 = \frac{4V_{DD}C_{out}(V_B - V_T)V_{amp}}{t_{range}[16(V_B - V_T)^4 + 4(V_B - V_T)^2(V_{amp})^2 + (V_{amp})^4]}$$

2) For differential VTC optimization

$$KP_3 = \frac{8V_{DD}C_{out}(V_B - V_T)V_{amp}}{t_{range}[4(V_B - V_T)^4 + (V_{amp})^4]}$$

The width of M3 can be determined fromKP_3.

The choice of KP_4 and V_{const} has an available extra degree of freedom.

1) For single-ended VTC optimization

$$V_{const} = V_T + \sqrt{\frac{3KP_3}{KP_4}}(V_B - V_T)$$

2) For differential VTC optimization

$$V_{const} = V_T + \sqrt{\frac{KP_3}{KP_4}}\ (V_B - V_T)$$

The width of M4 is suggested to be kept fairly small to improve noise performance and reduce parasitic capacitance on the node connected to the drain of M3 and M4, although not so small that V_{const}exceeds the limit of $\frac{1}{2}V_{DD} + V_T$.

5.4 VTC Half-Cell

The core VTC is a pseudo-differential circuit consists of two half-cells. Each half cell is fed by the same clock and bias voltages with complementary RF inputs.

5.4.1 The schematic of the VTC half-cell

The schematic is composed of 8 CMOS inverters, two of which have additional starving devices between the inverter and ground. A bias tee (LC) is used (off-chip) to couple the AC input signal with a DC bias voltage. Standard CMOS inverter used to sharpen the edges of the signal $V_{out}(t)$. The half-cell schematic is shown in Figure [48].

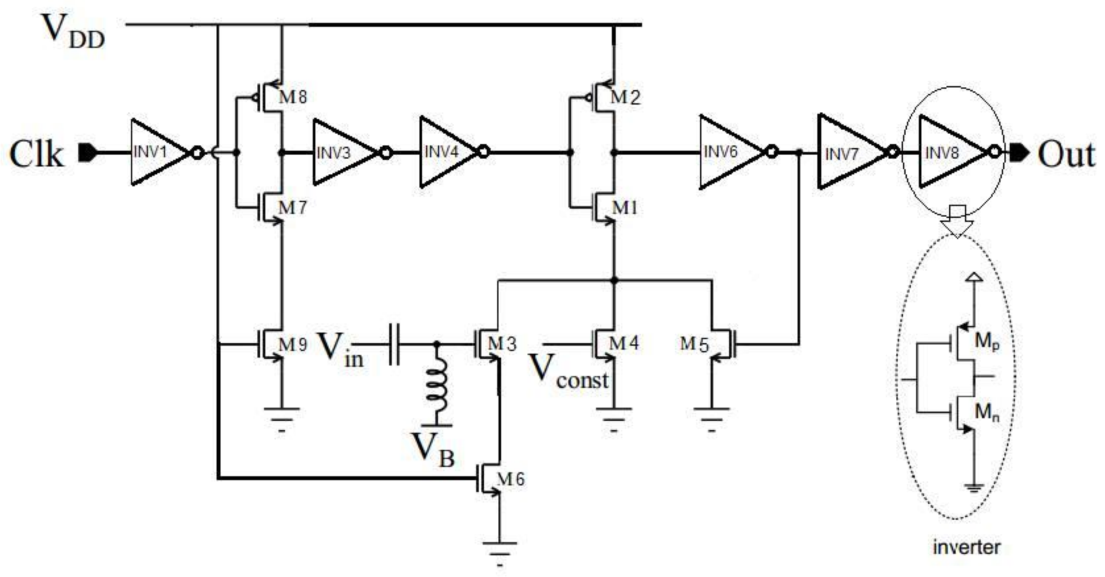


Figure 48 : VTC Half Cell Schematic

5.4.2 Duty-Cycle Adjustment Circuit

The first starved inverter, composed of M6-M8, delays the falling edge of the output signal. This device is used to adjust the duty cycle of the clock to allow greater conversion time. The starving device M5 is biased with the full supply voltage, adds a fixed delay to the falling edge of the output signal.

The second starved inverter, composed of M1-M5, delays only the rising edge of the output signal. The additional pulse width allows additional time for the VTC to complete each conversion cycle. Since the VTC delays the rising edge of the output, the pulse width of the output is variable.

5.4.3 VTC Core

The functionality of VTC core is provided by the starved inverter composed of M1-M5. This is the VTC circuit that was analyzed in detail. M5 is a small additional NMOS starving device with its gate connected to the output of the INV6 inverter that comes after the VTC inverter. M5 is attached only to ensure that the node at the source of M1 is fully discharged every cycle, even at the slow process corner. M5 has not any role in the delay process because it remains off until the INV6 inverter has already switched. It can be noted that the duty-cycle adjustment circuit is the second inverter in the chain and the VTC core is the fifth.

Depending on whether the delay is applied to one or both edges of the input clock pulses, A VTC can be referred to as either a pulse position modulator (PPM) or pulse width modulator (PWM) [15].

5.4.4 System Description

The VTC blocks are the same circuits used when not calibrating. These clocks must be provided from off-chip. The Figure [49] shows the VTC system block diagram. It is expected that a mature realization of the VTC would include internally generated clocks using a phase-locked-loop. For calibration, the analog VTC inputs $V_{in}P$ and $V_{in}N$ must be set to DC levels corresponding to the maximum and minimum values, respectively, that will be used in normal operation. So

$$V_{in}P = V_{bias} + V_{amp}$$

$$V_{in}N = V_{bias} - V_{amp}$$

Where V_{bias} and V_{amp} are the DC bias and AC amplitude of the VTC input used in normal operation.

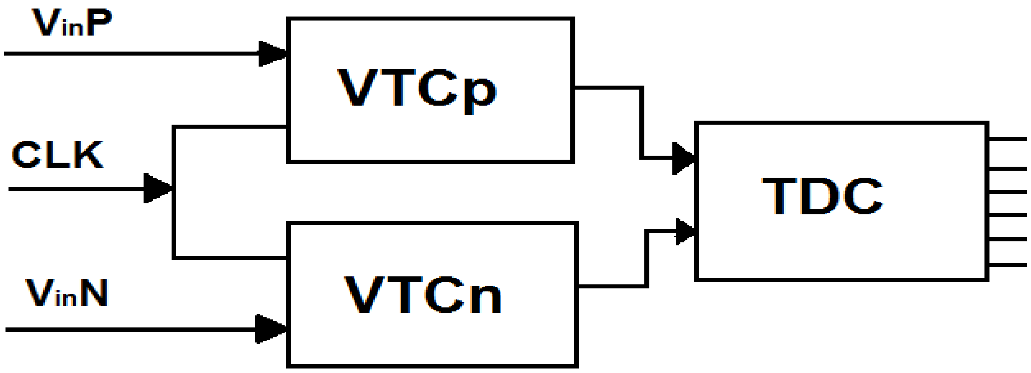


Figure 49 : VTC system Block Diagram

5.5 Linearity and Voltage Sensitivity

Linearity is an important property of any converter. There are various metrics for quantifying linearity, including integral non-linearity (INL), differential non-linearity (DNL), total harmonic distortion (THD), signal to noise and distortion ratio (SNDR), and effective number of bits (ENOB). Of these, ENOB is the most commonly used for data converters. The main problem with this delay cell is the nonlinearity between the controlled voltage (V_{IN}) and the delay value. This nonlinearity results in introducing distortion.

5.5.1 Linearization method

The main current starving device M3 is linearized by using source degeneration implemented with M6. Several current starving devices with different gate bias voltages were used in parallel with M3. This eases the compression of the pulse delay time versus input voltage characteristic at high input voltages. The additional parallel current starving devices also increase the voltage sensitivity of the VTC. Simulated results show the proposed linearization scheme improves the linearity and sensitivity of the VTC.

5.5.2 Voltage Sensitivity and Linearity

The voltage sensitivity and linearity of the VTC were simulated by sweeping the DC input of the VTC and measuring the clock pulse delay time using a transient analysis.

The linear range is significantly lower for the VTC with no linearization. Although the linearity of the VTC linearized with degeneration only is comparable to that of the VTC with the enhanced linearization scheme, the voltage sensitivity of the VTC with the enhanced linearization scheme is much higher. Using only source degeneration for linearization and increasing the width of the main current starving devices, M3, does not result in a VTC as sensitive to the input voltage as the VTC with the enhanced linearization scheme [26].

5.6 Simulated Results

The VTC was designed to accept differential input voltages range.

The transient output of the VTC were simulated by sweeping the DC input of the VTC and measuring the clock pulse delay time using a transient analysis with process in a commercial RF simulator.

The results are shown in Fig. [50], where the input voltage controls the delay of the rising edges of the clock pulses.

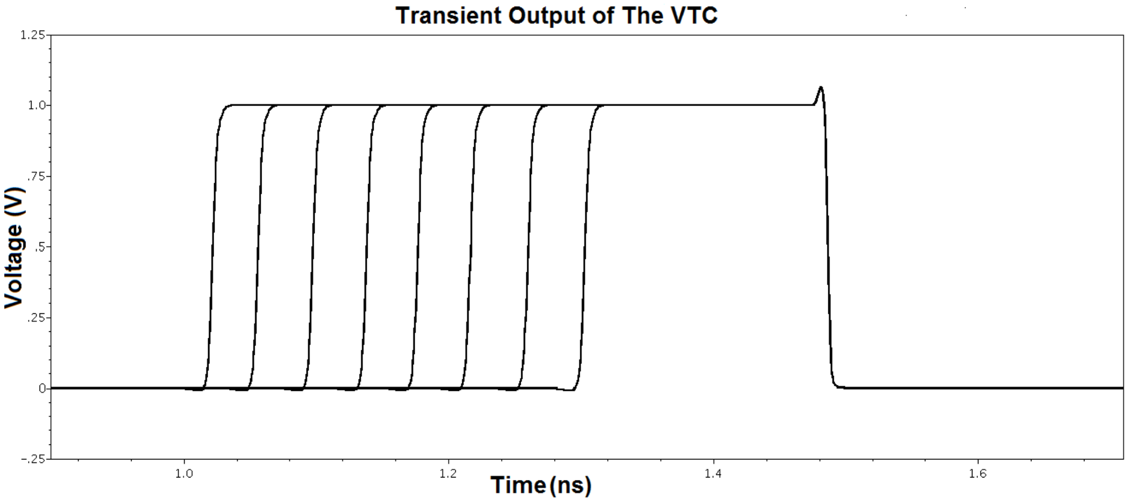


Figure 50 : VTC Transient Output

The circuit was first tested with DC inputs and a 1GS/s clock signal. Fig. [51] shows the results over a full input range. The output delay is the time between a rising edge on the VTC output and a rising edge on the output clock signal.

The highlighted area shows the dynamic range. Outside of this range the curve is highly non-linear with delay saturation for very low and very high voltages.

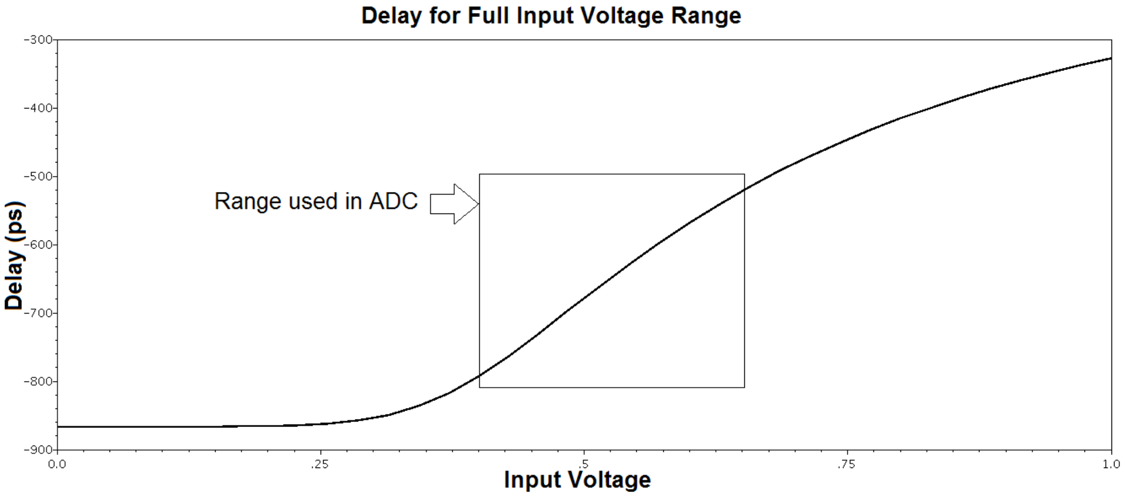


Figure 51 : VTC Output Delay

To illustrate the effect of the tuning voltage, Fig. [52] shows the output delay curves for various input voltages (dynamic range). The input voltage in the plot is in addition to the constant DC bias voltage.

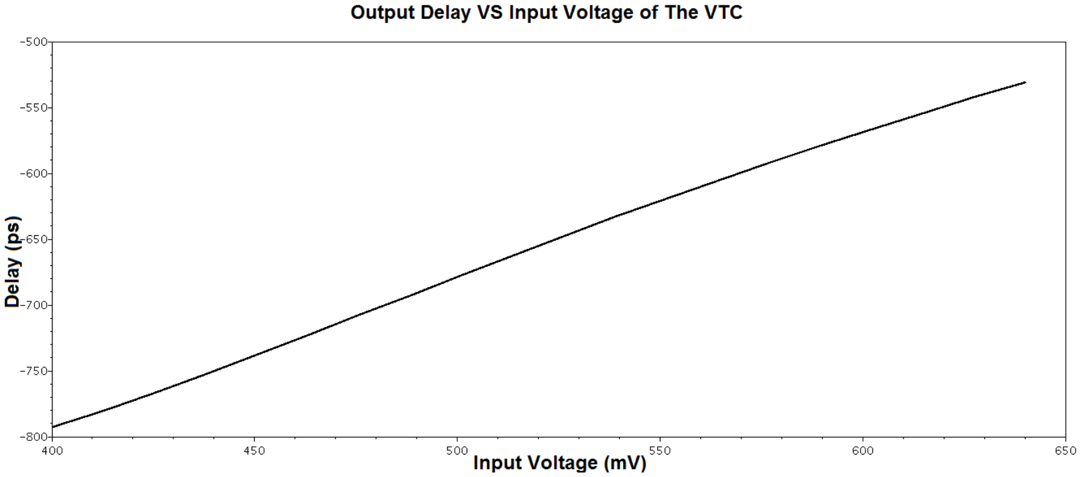


Figure 52 : Linear Operation Range

Chapter 6: Vernier Delay line TDC in 65nm CMOS

6.1 TDC Core

TDC is time to digital converter, which converts the delay in time to digital output. There several types of TDC such as Vernier delay line and Flash. Here in our design we used Vernier delay line TDC.

6.1.1 6-bit Vernier line TDC

Shown in Fig. [53], the design of the 6-bit VDL-based TDC. The inputs lnP (positive line) and InN (negative line) are pulse trains in time that represent sampled information as the difference between the rising edges of the two signals (Δtin). The TDC is designed to operate with time delay step=tδ.

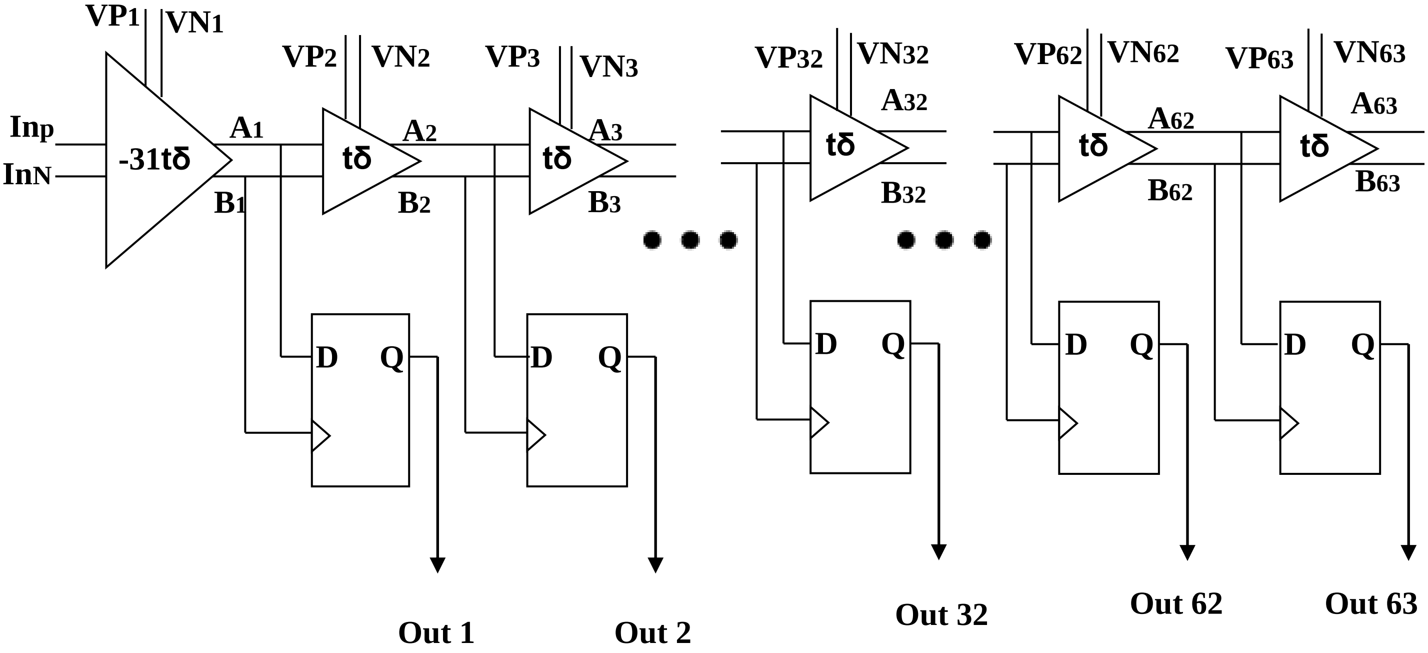


Figure 53 : 6-bit Vernier Delay Line

The TDC consists of 63 stages, each stage consists of a tunable delay cell and a flip-flop. The delays are tuned using control signals VPi and VNi. The first delay is designed to delay the negative input relative to the positive by 31tδ. All subsequent delay cells delay the inputs in the opposite direction, delaying the positive input relative to the negative input by tδ.

After each delay, a flip-flop decides according to which input's rising edge occurs first, and stores the output (e.g. Out1, Out2, etc.). The flip-flops use a sense-amplifier design.

The flip-flop outputs act as an input to 64-bit thermometer code representation of the TDC output.

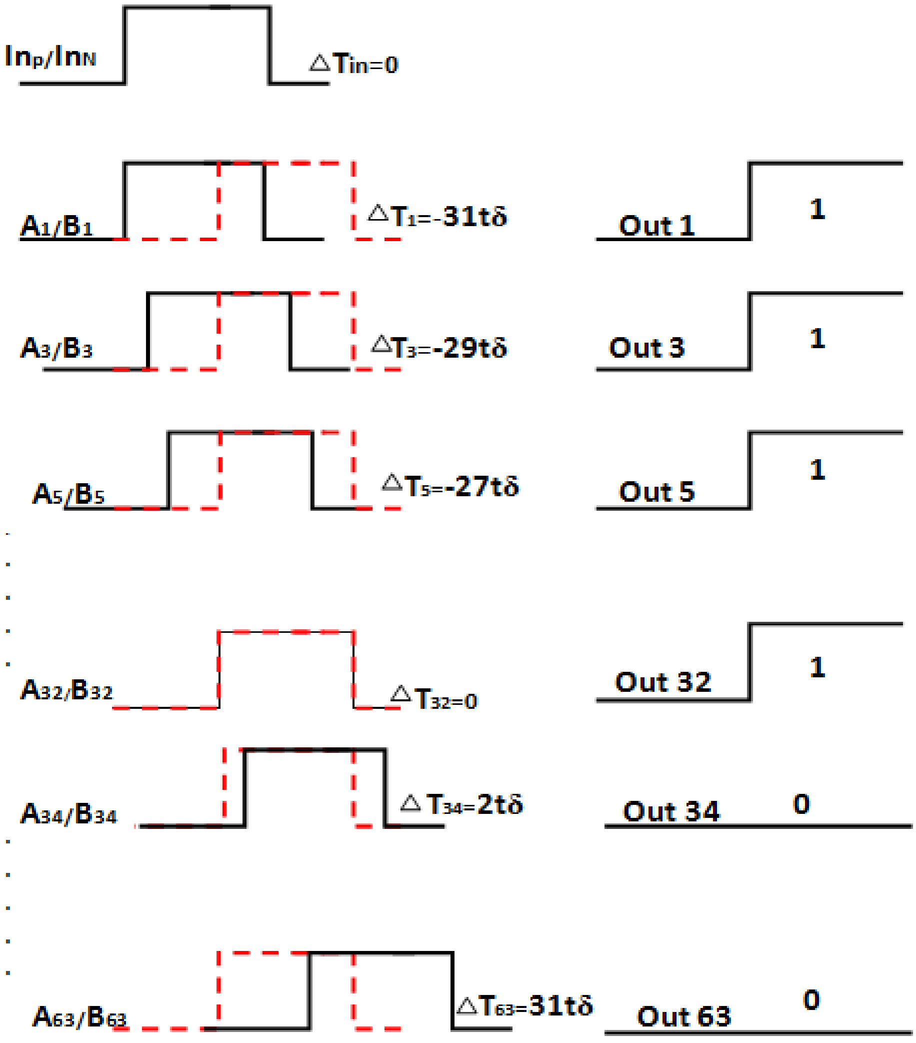


Figure 54 : Simulated TDC waveforms

Fig. [54] shows waveforms of simulated TDC for a single input pulse. In this example, signals InP and InN arrive at the TDC input at the same time. After passing through the first delay stage, a delay of −31tδ is introduced between the signals A1 and B1. The other following delay stages (2 through 63) each adds a positive tδ (tδ=3.125ps) to the delay between the two signals. In the figure, we gave only some examples to illustrate the idea also in order to avoid clutter.

It can be seen that the delay between the two signals changes in the range of -31tδ to +31tδ as the signals travel through the VDL. Also shown in Fig. [53] that, the labels of the outputs of the flip-flops, Out1, Out3 and so on. The flip-flop output will be ‘0’ when its clock signal (Bi) comes before its input signal (Ai), or in other words when Δti > 0. The output will be ‘1’ when the data comes before the clock, or when Δti < 0. In this way, the thermometer code output is built up.

In our design the restriction on the speed of the overall ADC was that to be equal 1GS/s. In the 6-bit VDL TDC it's obvious that at the delay stage number 32, the two input signal returns to their initial state (the state with which they arrived the TDC with respect to each other), as in the example the two signals at stage 32 returned aligned (signals were input exactly aligned). This is because that at the first stage, the positive input (InP) is delayed -31tδ with respect to the negative input (InN). Then at each following delay stage, the InP is delayed tδ with respect to the InN, so at stage number 32 the effect of the first stage will be eliminated. So, we used this property in our VDL to increase it’s speed by changing it from one series delay line to two a parallel delay line.

The first line consists of the first 32 delay stages, and the other parallel line consists of the remaining 31 delay stages. When the input signals arrive the TDC they enter the both parallel lines, and they travel in the two parallel lines simultaneously, so the speed is almost doubled Fig. [55].

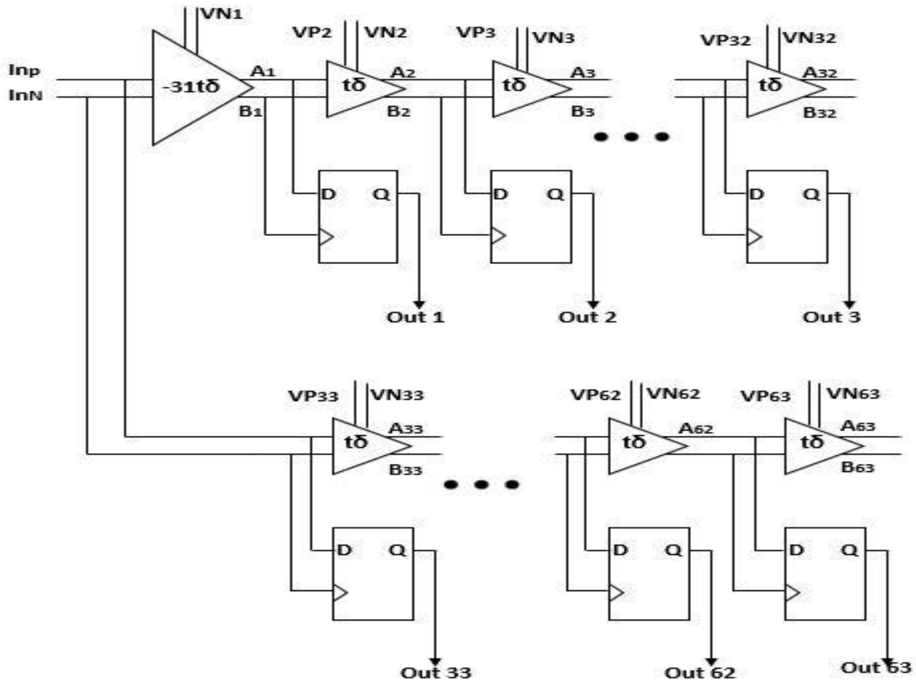


Figure 55 : 6-bit parallel VDL

6.2 Delay Blocks

We produced the variable delays needed for the VDL by the delay block shown in Fig. [56]. The core of the delay block is made by M4 and M5 in a standard CMOS inverter configuration, but using voltage-controlled current-starving devices M3 and M6, we are limiting the maximum current through the inverter. By tuning the voltages VgP and VgN we adjust the delay of the rising and falling edges, respectively, of the inverter. Devices M1 and M2 are small devices (one quarter width) compared to the rest devices, to ensure that a minimum amount of current is able to flow even if M3 and M6 enter cut-off mode. This ensures that an output will be produced regardless of the values of VgP and VgN.

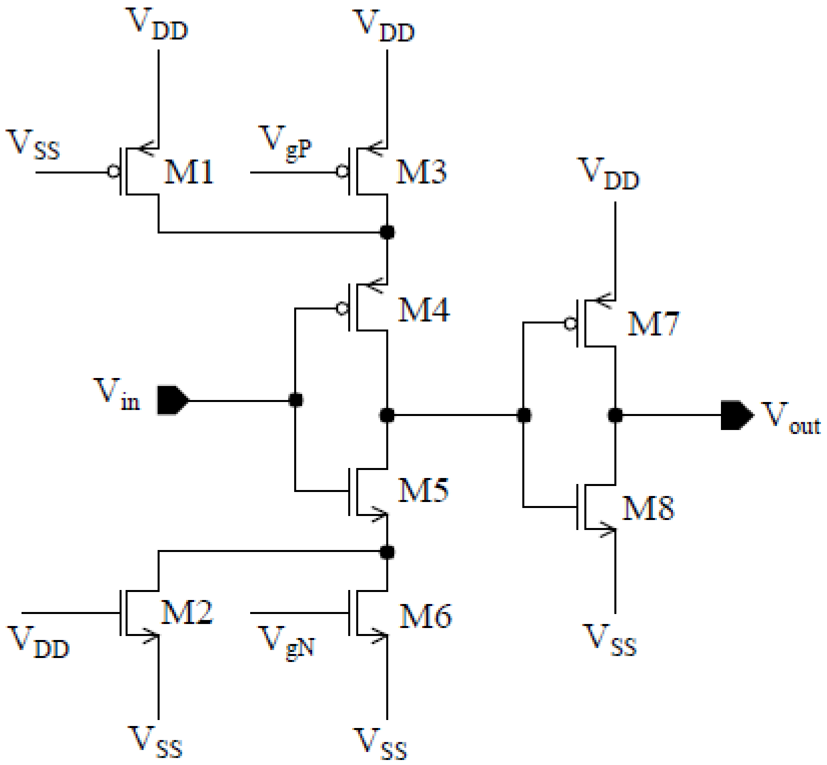


Figure 56 : Single Delay Block Schematic

Devices M7 and M8 make a standard CMOS inverter which sharpens the edge transitions and negates the inversion of the first portion of the delay block. By this way, the output rising edges continue to correspond to the input rising edges, and likewise for the falling edges.

6.2.1 Delay Tuning

As mentioned above, the DC voltages VgP and VgN can be tuned to change the delay of the circuit. We need to find a relationship between these voltages so that we can treat them as a single tuning parameter delay both the rising and falling by the same amount. To find this relationship, first consider the design of a standard CMOS inverter. From [27] an approximation for the propagation time for the rising edge (tpLH) and falling edge (tpHL) are

$$t_{pLH} = \frac{1}{2}\frac{V_{DD}C_L}{I_{DP}}$$
$$t_{pHL} = \frac{1}{2}\frac{V_{DD}C_L}{I_{DN}}$$

VDD is the supply voltage, CL is the output load capacitance, and IDN and IDP are the drain currents of the NMOS and PMOS transistors respectively during switching. The propagation time is defined as the time for the output to fall or rise to 50% of the full scale voltage when excited by a voltage step (negative or positive). Since both transistors are charging or discharging the same load capacitance, to equalize the rising and falling edge propagation times we must equate the NMOS and PMOS drain currents. Using simple Level 1 models for transistors in saturation, the currents are

$$I_{DN} = \frac{\mu_n C_{ox}}{2}\frac{W_N}{L}(V_{DD} - V_{TN})^2 \qquad \text{(NMOS)}$$
$$I_{DP} = \frac{\mu_p C_{ox}}{2}\frac{W_P}{L}(V_{DD} - |V_{TP}|)^2 \qquad \text{(PMOS)}$$

where µp and µn are the hole and electron mobilities respectively, Cox is oxide capacitance per unit area, WP and WN are the PMOS and NMOS gate widths, L is the gate length, and VTP and VTN are the absolute PMOS and NMOS threshold voltages. VTP and VTN are similar, so all that is left is to adjust the device widths so that the ratio $\frac{Wp}{Wn}$ is equal to $\frac{\mu n}{\mu p}$. This makes the propagation delays for the falling and rising edges to be equal. Simulations show that the correct ratio for the 65nm process is 2.1472.

For the voltage-controlled current-starving transistors (M3 and M6 in Fig. [56]), the current produced is

$$I_{DN} = \frac{\mu_n C_{ox}}{2}\frac{W_N}{L}(V_{gN} - V_{TN})^2 \qquad \text{(NMOS)}$$
$$I_{DP} = \frac{\mu_p C_{ox}}{2}\frac{W_P}{L}(V_{DD} - V_{gP} - |V_{TP}|)^2 \qquad \text{(PMOS).}$$

Once again the ratio $\frac{Wp}{Wn}$ is equal to $\frac{\mu n}{\mu p}$ so to provide the same current drive from each device. The gate inputs VgP and VgN must then be adjusted together according to the equation

$$V_{gN} = V_{DD} - V_{gP}.$$

This relationship ensures that the rising and falling edges are delayed nearly by equal amounts, prevent the output pulse width from growing or shrinking relative to the input pulse width.

6.2.2 Simulated Results

Fig. [57] Shows the delay produced by the block in Fig. [54], as we sweep VgN and VgP from 0 to VDD. This is the absolute delay of the block; that is, the time between the input rising edge crossing the 50% threshold and the output rising edge crossing the same threshold.

For the VDL, the more important delay is the differential delay shown in Fig. [58]. This is difference between the delays of the rising edge of the delay block of Fig. and a fixed delay block (the same circuit but with VgN and VgP connected to VDD and Ground, respectively). And it's obvious that the differential enhances the resolution in time. We target the delay to be at normal operation equal to tδ or 3.125ps as mentioned above. However the additional tuning range enables the TDC to be used at lower data rates with increased timing resolution. Tuning resolution decreases for higher delays however, due to the steeper slope of the delay curve for VgN<0.6.

Fig. [59] shows that the VgP=VDD-VgN relationship is effectively keeps the pulse width within +0.5/1.5ps of the nominal 500ps over the entire tuning range. When the circuit is biased for 3.125ps differential delay the pulse widths range from 500.44ps to 501.5ps.

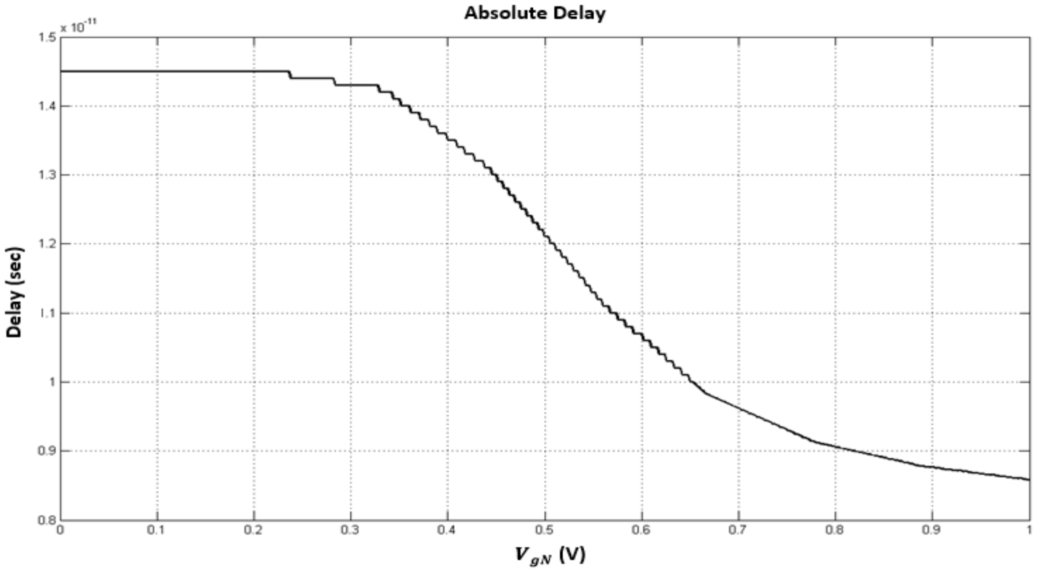


Figure 57 : Absolute Delay

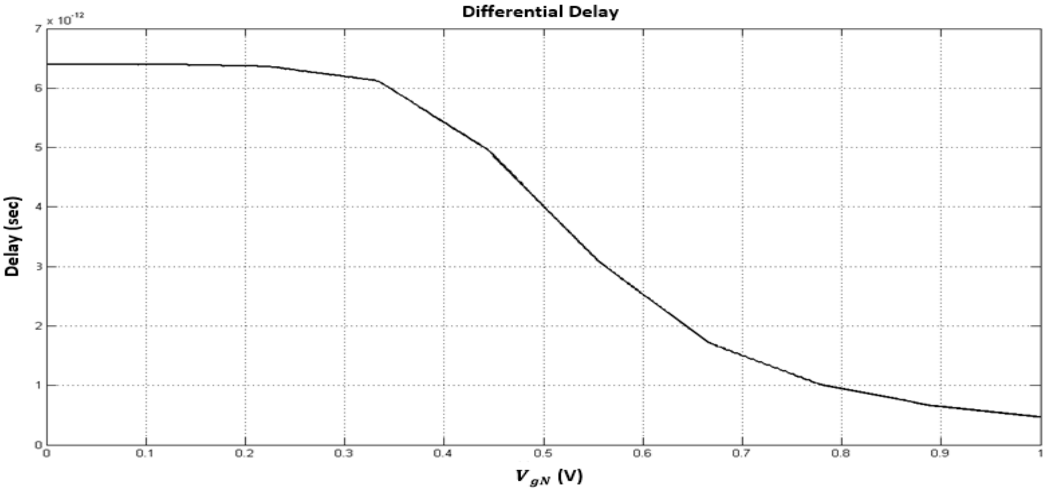


Figure 58 : Differential Delay

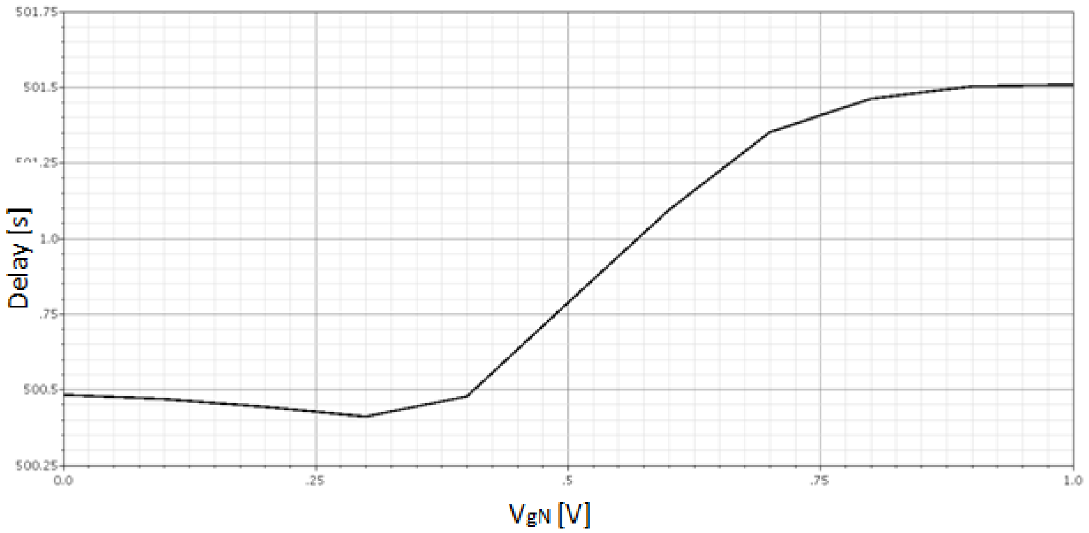


Figure 59 : Pulse Width variation with VgN

6.2.3 Complete Differential Delay Blocks

As described previously, the single tδ delays are generated using a differential circuit with one path has a variable delay that can be controlled and the other has a fixed delay. Fig. [60] (a) shows this circuit. The lower path with its starving devices biased to generate maximum current, producing the minimum delay possible for the circuit. The upper path introduces an additional delay, controlled by the VgN and VgP inputs. For the delay of -31tδ, the configuration of Fig. [60] (b) is used. The same individual delay elements are used (that of Fig. [56]), but with 2 elements in series. In this case, the upper path exhibits minimum delay while the bottom path is delayed using VgN and VgP, producing a delay in the opposite direction of the tδ circuit. Using 2 elements to generate 31 times the delay means the circuits have to be biased further up the delay curve. The resolution of the tuning of this block is therefore lower due to both the factor of 2 and the increase that happens in the slope as delay increases (see Fig. [58]. Using 2 elements rather than 31 decreases the power consumption, noise generation and layout area for the circuit. It could be possible to save much more area by using a single delay element in the bottom path and no element in the top path. However, this was considered very risky as any process variation beyond what the simulator predicts would make it impossible to reach the desired delay.

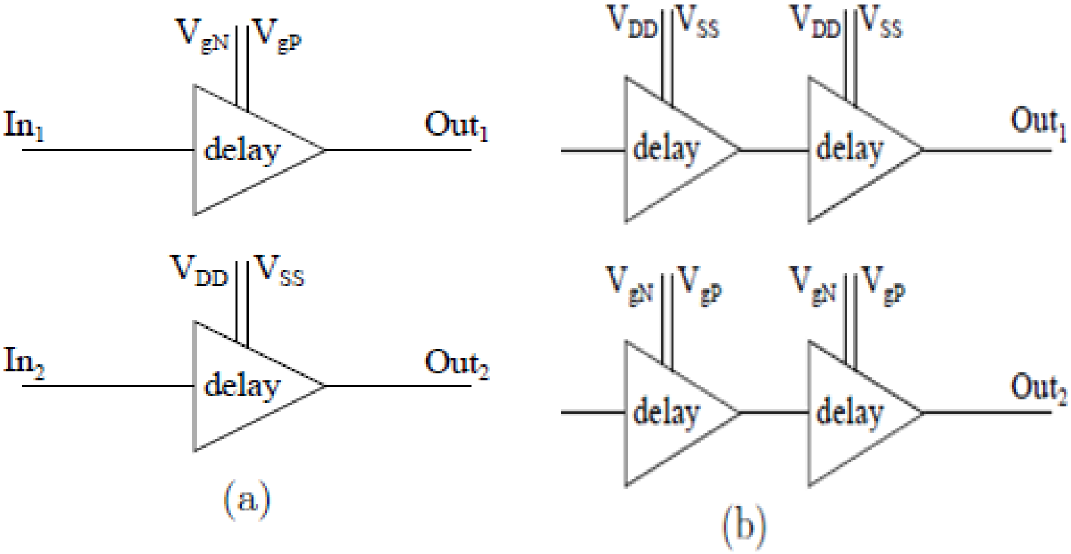


Figure 60 : Differential Delay Blocks for generating (a) tδ (b) -31tδ

In order to reach with the ADC speed to be equal 1GS/s, it was desirable to make the delay of the bottom line small as possible, as to get the required time resolution (tδ=3.125ps) from the differential delay blocks with the minimum delay between the input and the output of the delay cell. So we swept the bottom line delays over the widths of the transistors M4, M5, M7 and M8 (see Fig. [56]) Fig. [61], and chose the width that approaches the minimum delay.

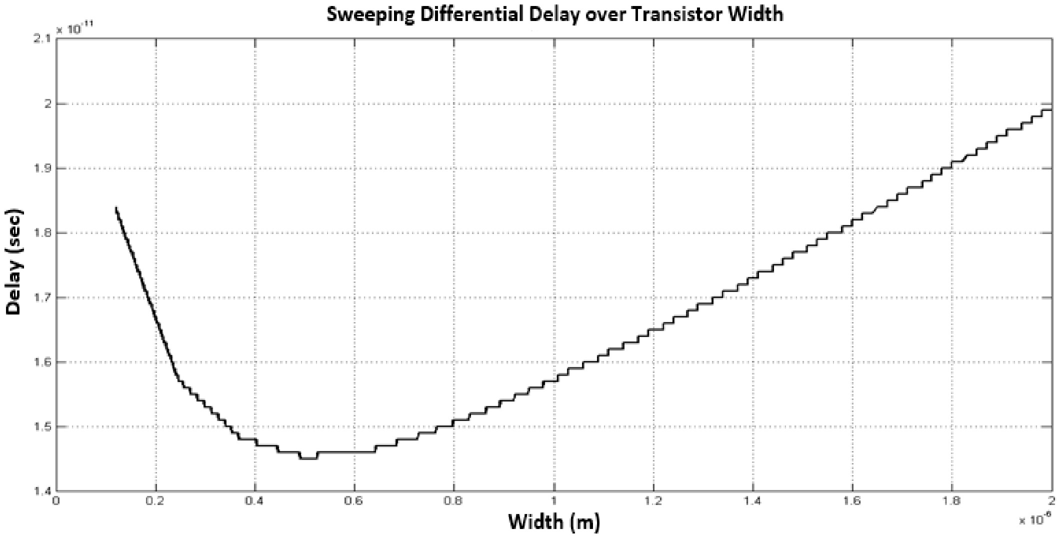


Figure 61 : Sweeping the differential delay over transistor width

6.3 Sense Amplifier D-flip-flop

Sense Amplifier flip-flops (SAFF) has been used in high performance and low- power digital systems. Recently some modification were added to the SAFF exhibit a very small delay which is calculated as the sum of setup time and clock to output delay in high speed applications. The output of the SAFF is differential while the input may be single or differential. The SAFF acts with a very small load on the clock since only three transistors are connected to the clock [28, 29, 30]. The first design of the SAFF was based on the sense amplifier in the first stage and the NAND based cross coupled SR latch in the second stage Fig. [62] Modifications were added to the SAFF to improve the output stage and to reduce the overall delay [30].

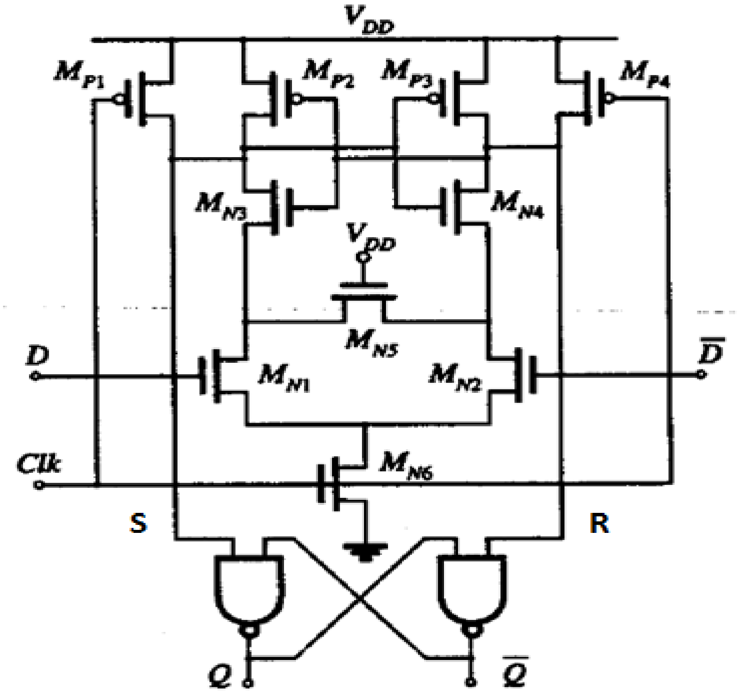


Figure 62 : Sense Amplifier D-FlipFlop

6.3.1 Analysis of operation

The sense amplifier stage [28] is precharged during as long as the clock signal is low. The high state of S and R keeps MN3 and MN4 on, there sources are being charged up to VDD - VtN because there is no way to the ground as the clocked transistor MN6 is off. Since MN1 or MN2 is on, the common node of MN1, MN2 and MN6 is also precharged to VDD – VTN. So, before the rising clock edge all the capacitances in the differential tree are precharged. Rising edge of the clock makes a falling transition on only one of the sense amplifier outputs, S or R.

This negative pulse is being captured by the SR latch and is being held until the end of the cycle. AS in [28] SAFF didn't include the transistor MN5. MN5 is added to allow static operation, providing a path to the ground even after the data changed. This to prevent the potential charging of the low output of the sense amplifier stage, due to the leakage currents.

S and R can’t be zero at the same time. When the clock is logic low, both S and R are precharged to logic high. During the transition of the clock from low to high, signals S an R evaluate new logic values.

Ideally, when clock (CLK) changing from 0 to 1, and D=0, the value of the S node should be unchanged, as well as the value of R during the rising of the CLK and D=1, as the flip-flop works properly.

(a) When CLK changes from logic 0 to logic 1, causing the triggering of the flip-flop. With D is logic 1, branch S is pulled to logic 0.

(b) When CLK=1 and D=0, in this case the sense amplifier output R is forced low at the leading edge of the clock, leakage currents could charge this node, and at the end changes the state of the flip-flop.

This problem was noticed in [29], and added modification shown in Fig. [62]. we added transistor MN5 in Fig. [62], to allow static operation, as it provides a path to the ground even after the data is changed. This is to prevent eventual charging of the low sense amplifier output, due to leakage currents, to the value when it could trigger the latch.

6.4 Output Decoding using Thermometer Binary Encoder

Encoder performance is critical for the design of ADC. High speed encoders are employed in very fast ADCs such as our TADC, which is used in application such as UWB communication receivers involving medium to high resolution and high speed [31]. In our design the $2^n - 1$ -63- (where n is the number of bits in our ADC = 6) delay stage (delay cells and D-flip-flop see Fig. [53] or [55]) in our 6-bit ADC generate the thermometer code in combination of series of ones then zeroes. The encoder circuit converts the thermometer code output from the delay stages into a binary code.

6.4.1 MULTIPLEXER BASED ECNCODER

Multiplexer based encoder is a very fast encoder that exhibits a comparatively small delay which is convenient with our 1GS/s ADC.

Multiplexer based encoder idea is based on binary search algorithm and uses only 2:1 multiplexers. According to bits level of significance they are generated [32,33]. For n-bit encoder (2^n – 1 inputs), the center bit acts as the most significant bit (MSB). The MSB is logic 1 only if half of the inputs of the encoder are high. To find the next lower significant bit (MSB - 1), the thermometer code is divided into two parts separated by MSB. The center bit of the two divided parts is decoded using 2:1 multiplexer where the previous MSB acts as the select signal to the multiplexer. The output of the multiplexer is used as next lower significant bit (MSB - 1) code. For the next lower significant bit, the codes are again divided into two parts and multiplexed recursively till only 2:1 multiplexer remains.

Due to the regular structure of this encoder it can be easily expanded to be used in a system of higher resolution than 4-bits as in our system we used 6-bit multiplexer based encoder consists of 5 stages, each stage consists of 31, 15, 7, 3 and 1 multiplexer respectively.

Also the multiplexer based encoder has regular physical layout due to it's regular structure.

In Fig. [63], 4-bit multiplexer based encoder is shown instead of our 6-bit encoder to avoid clutter but both with the same concept.

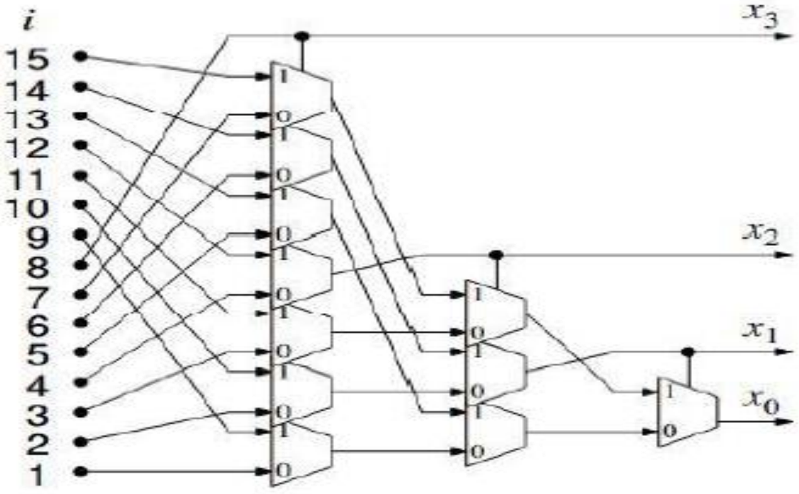


Figure 63 : 4-bit MUX based Encoder

Chapter 7: Layout

7.1 Full Custom IC Design Flow

As shown in Figure [64], that we first define our specifications at system level, then we try to achieve it at schematic level, if we succeeded to achieve required specifications we move to more advanced level that called layout, where we can make physical verification tests in order to tape out our IC design.

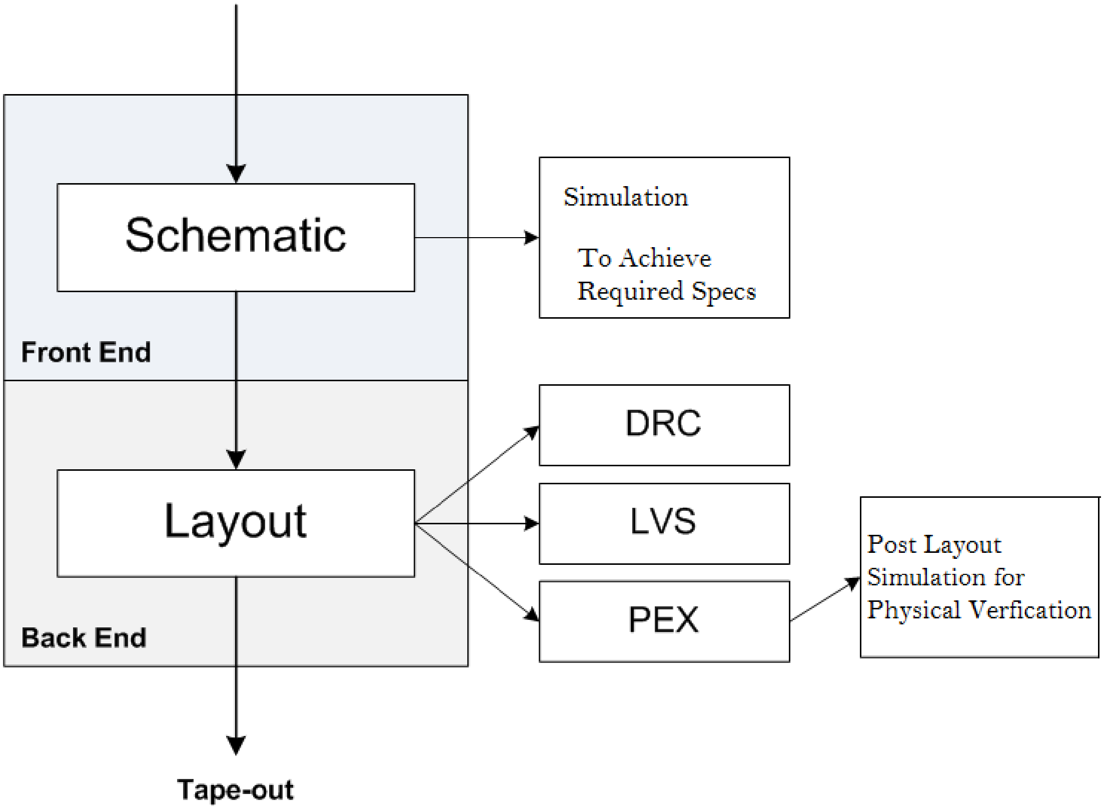

Schematic
Simulation
To Achieve Required Specs
Front End
DRC
Layout
LVS
Post Layout Simulation for Physical Verfication
PEX
Back End
Tape-out

Figure 64 : Analog IC Design Flow

7.1.1 Physical Verification

1. Routing between components, then main blocks.
2. DRC (Design Rule Check) to detect if found any design rule violation.
3. LVS (Layout Versus Schematic) to check the interconnection of blocks from both layout and schematic.
4. PEX (Parasitic Extraction Using XCalibre) to simulate parasitic effects and see how system specifications are affected.

7.2 Layout of Main Blocks

7.2.1 VTC Layout

1. Inverter

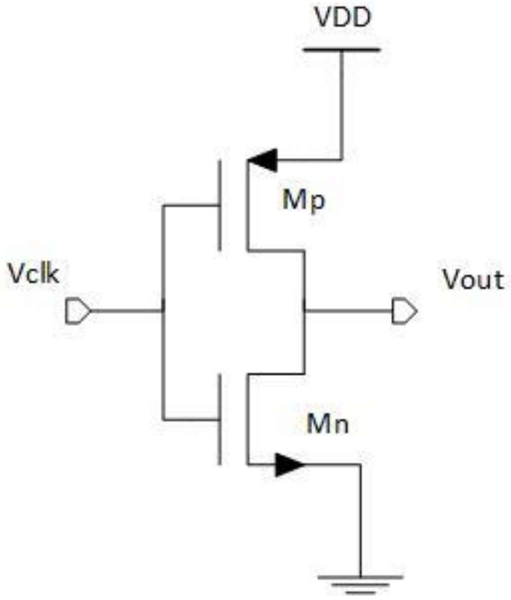


Figure 65 : Inverter Schematic

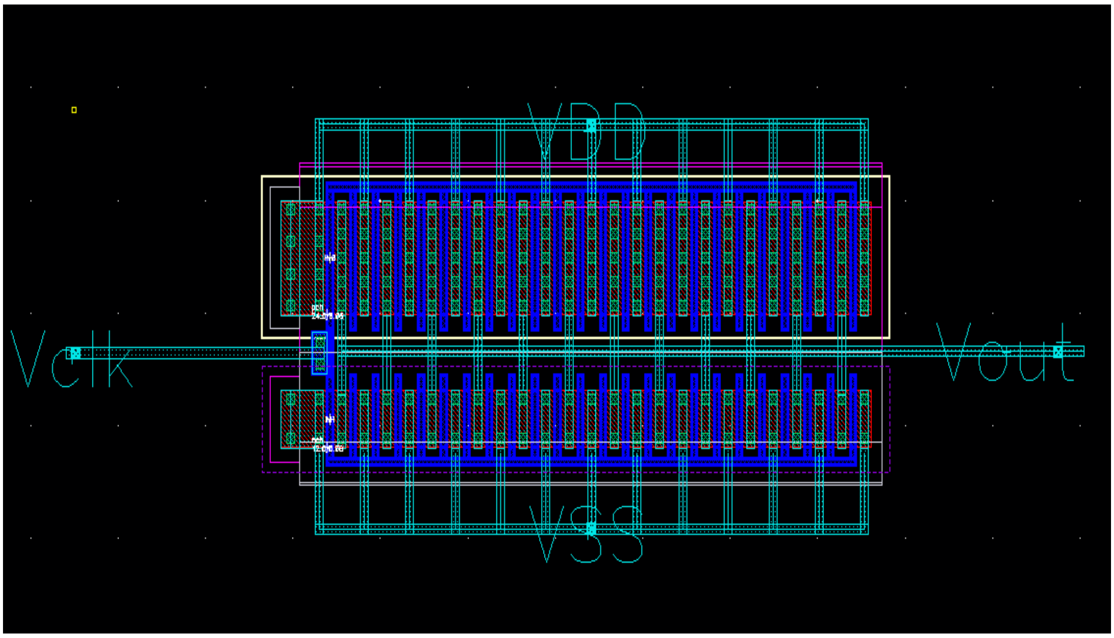


Figure 66 : Inverter Layout

2. Buffer

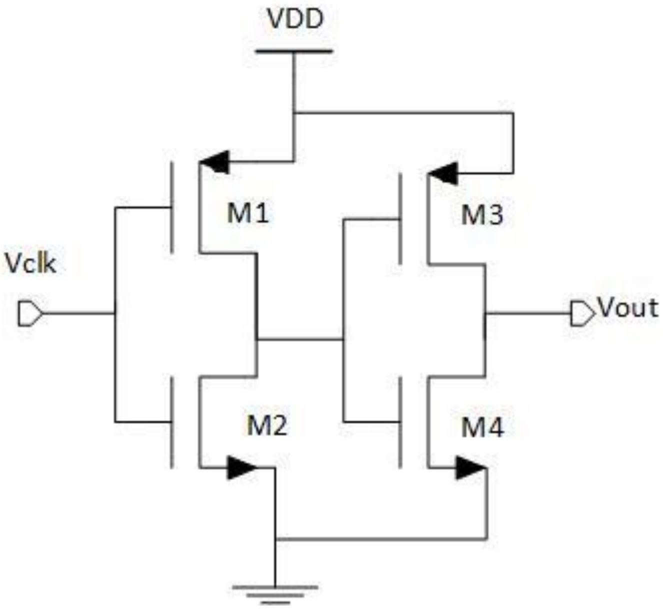


Figure 67 : Buffer Schematic

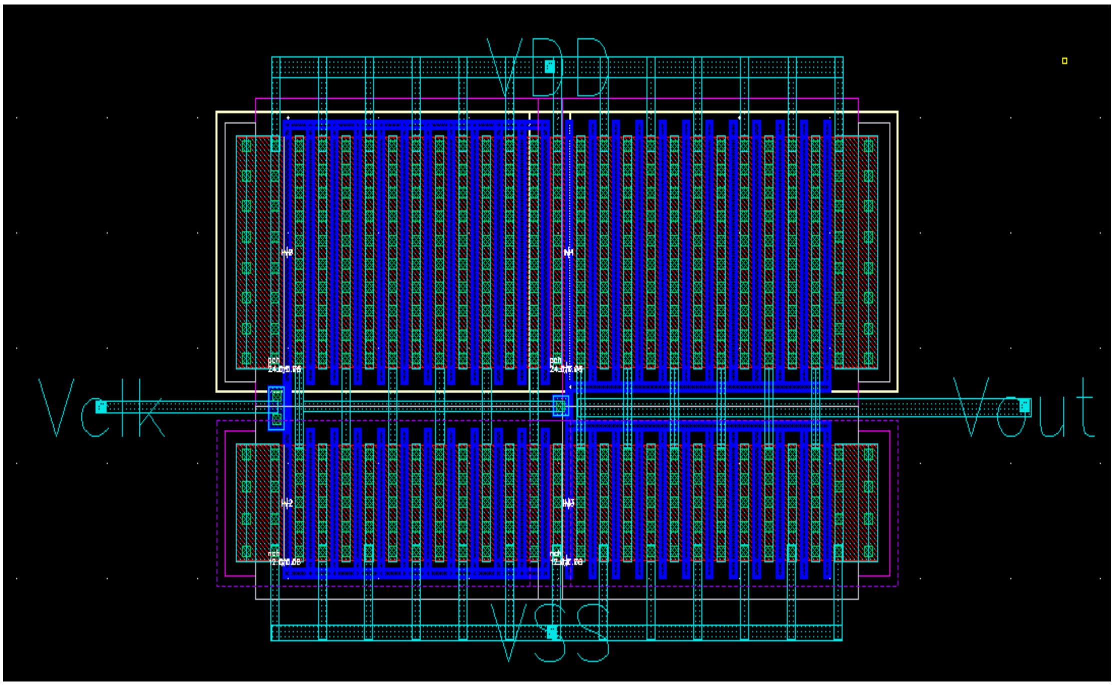


Figure 68 : Buffer Layout

3. Current Starved 1

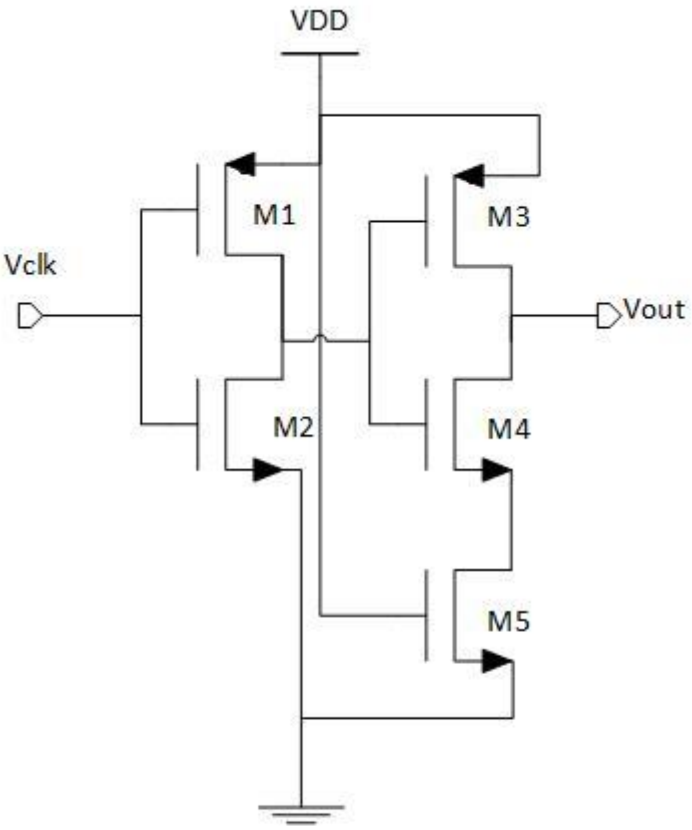


Figure 69 : Starved 1 Schematic

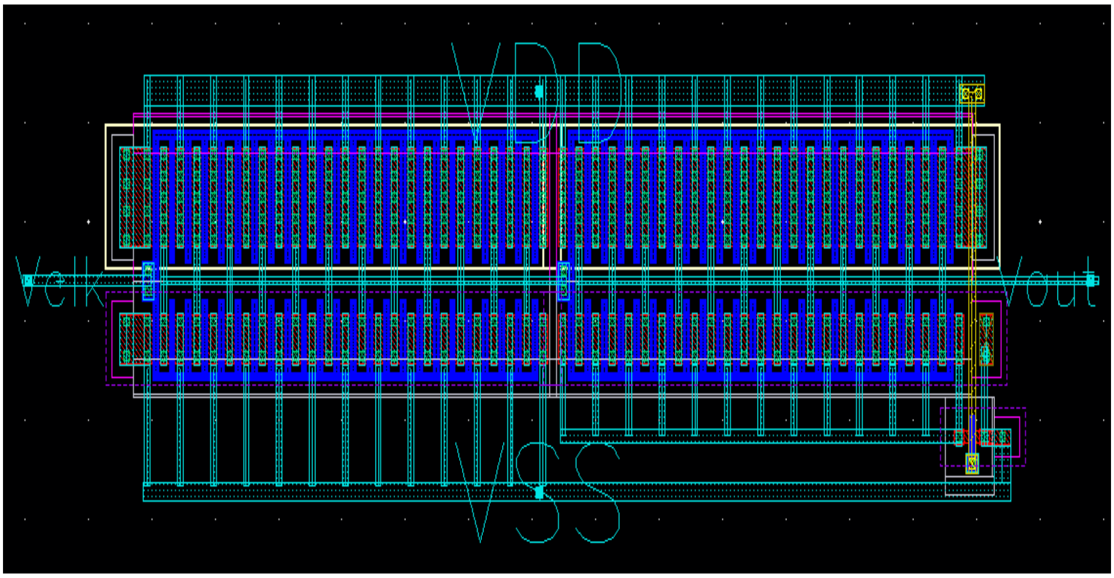


Figure 70 : Starved 1 Layout

4. Current Starved 2

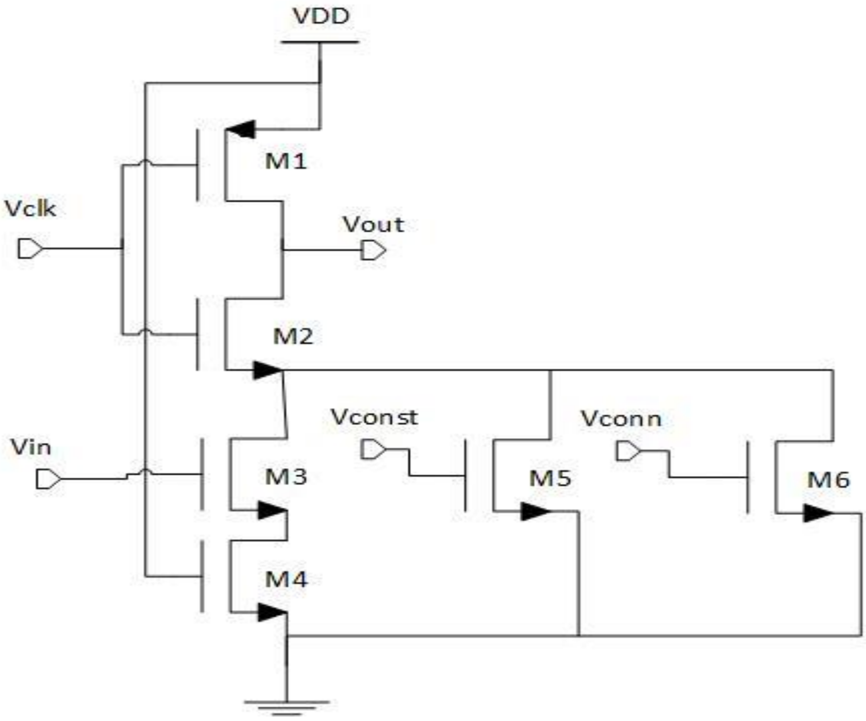


Figure 71 : Starved 2 Schematic

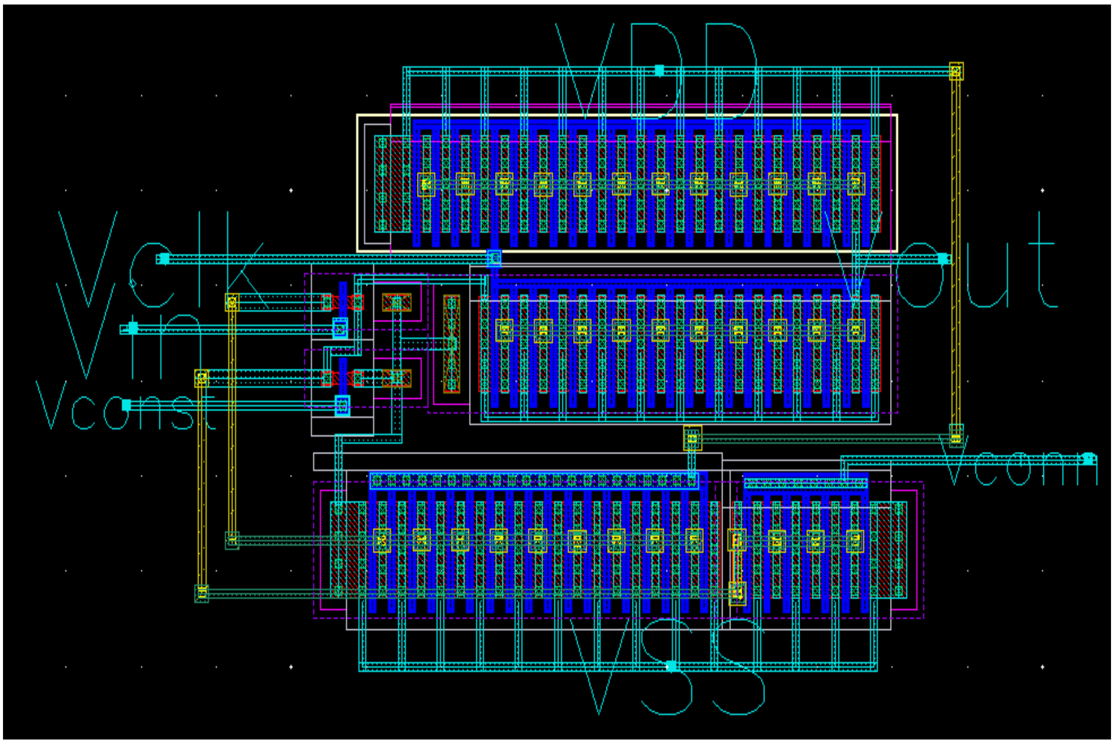


Figure 72 : Starved 2 Layout

5. VTC Core

Schematic is shown in Figure [48]

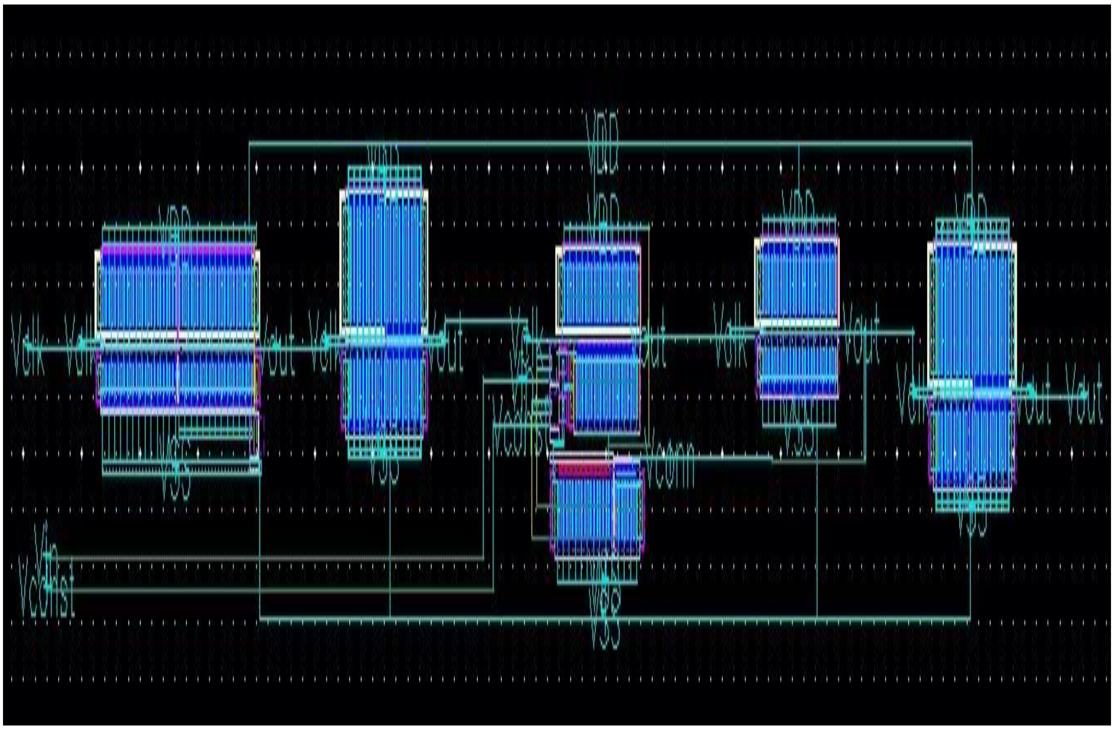


Figure 73 : VTC Half Cell Layout

7.2.2 TDC Layout

1. Big Delay Cell

Schematic is shown in Figure [56]

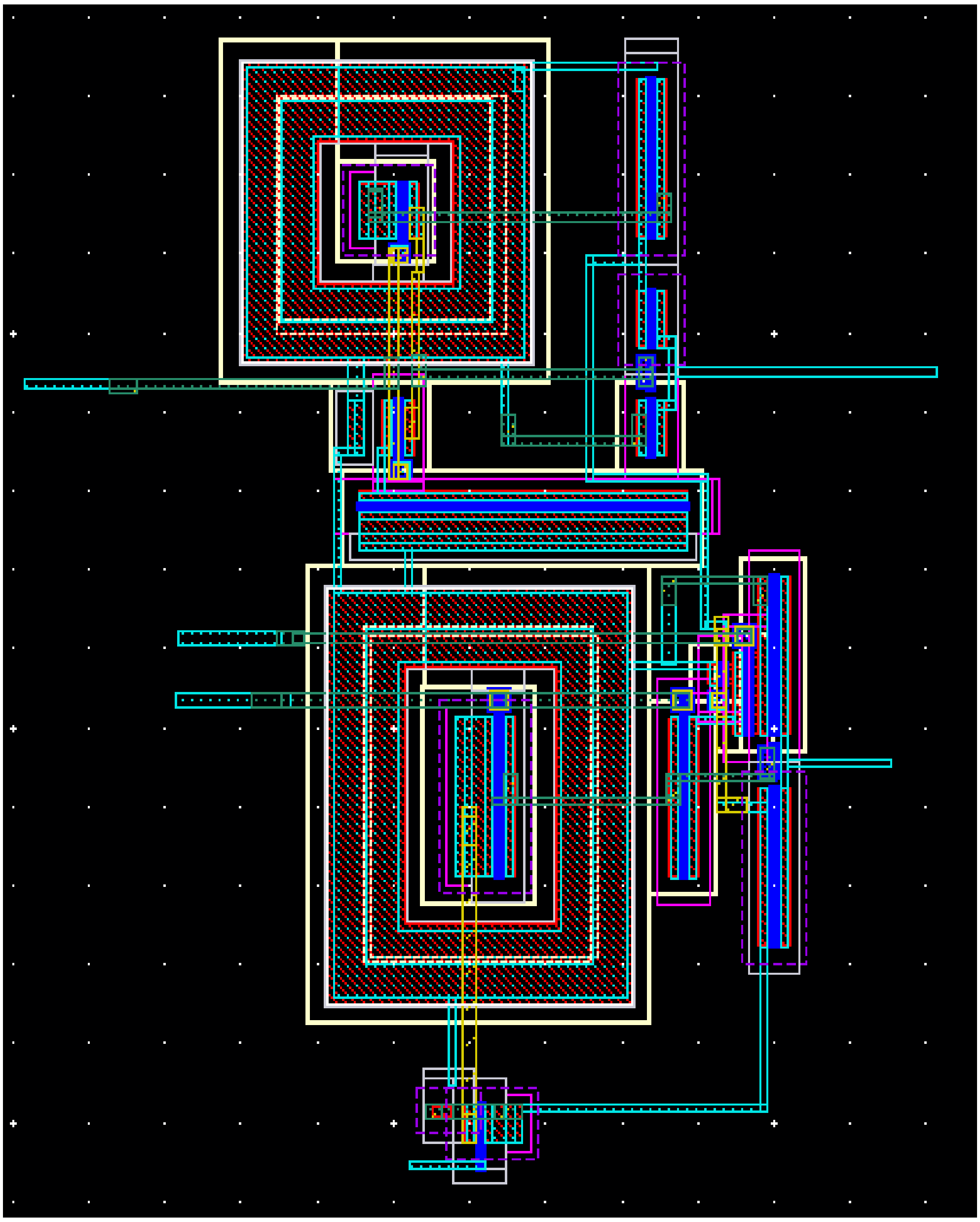


Figure 74 : Big Delay Cell Layout

2. Delay Cell

Same schematic with different Sizing

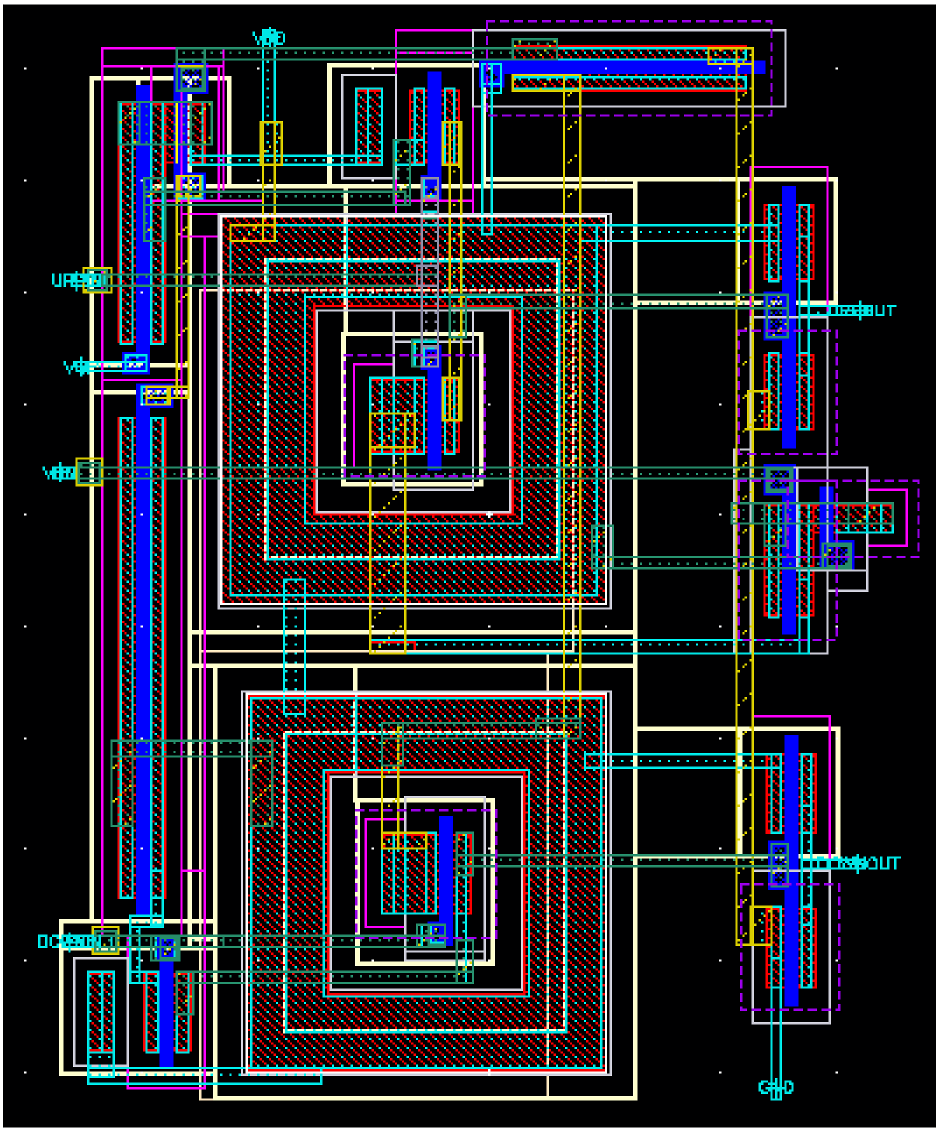


Figure 75 : Delay Cell Layout

3. D-Flip Flop

Schematic Shown in Figure [62]

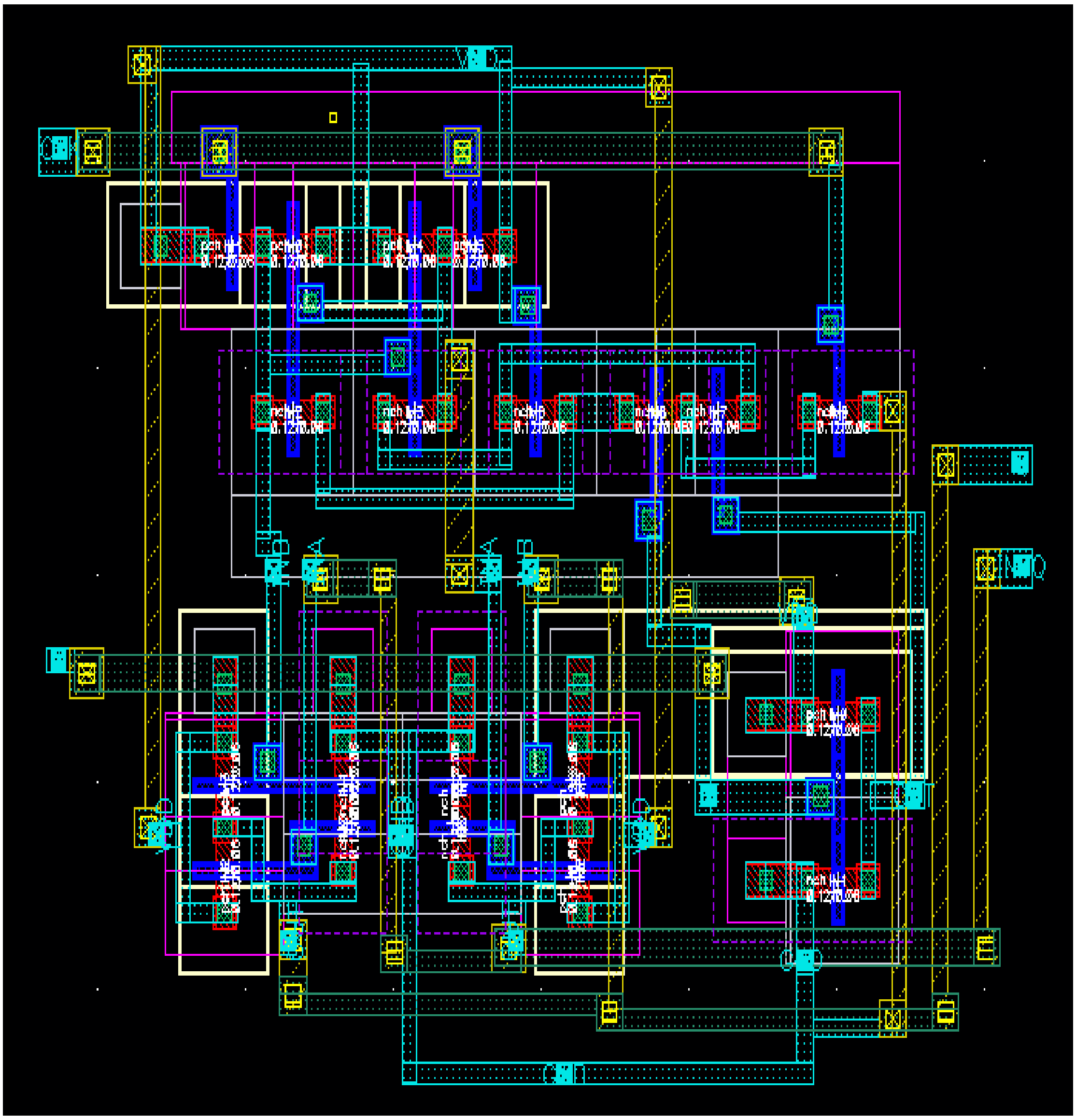


Figure 76 : D-FF Layout

4. 14 Delay Cell connected to 14 D-Flip Flop

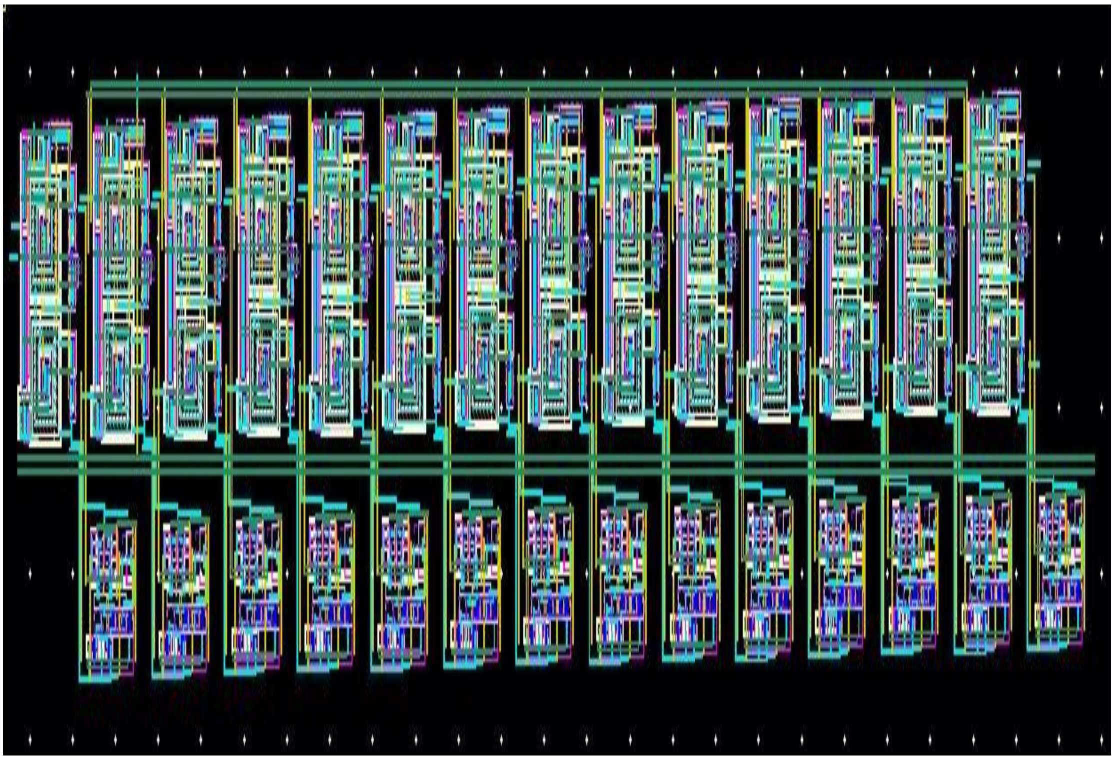


Figure 77 : TDC Main Block Layout

5. MUX based Thermometer Encoder

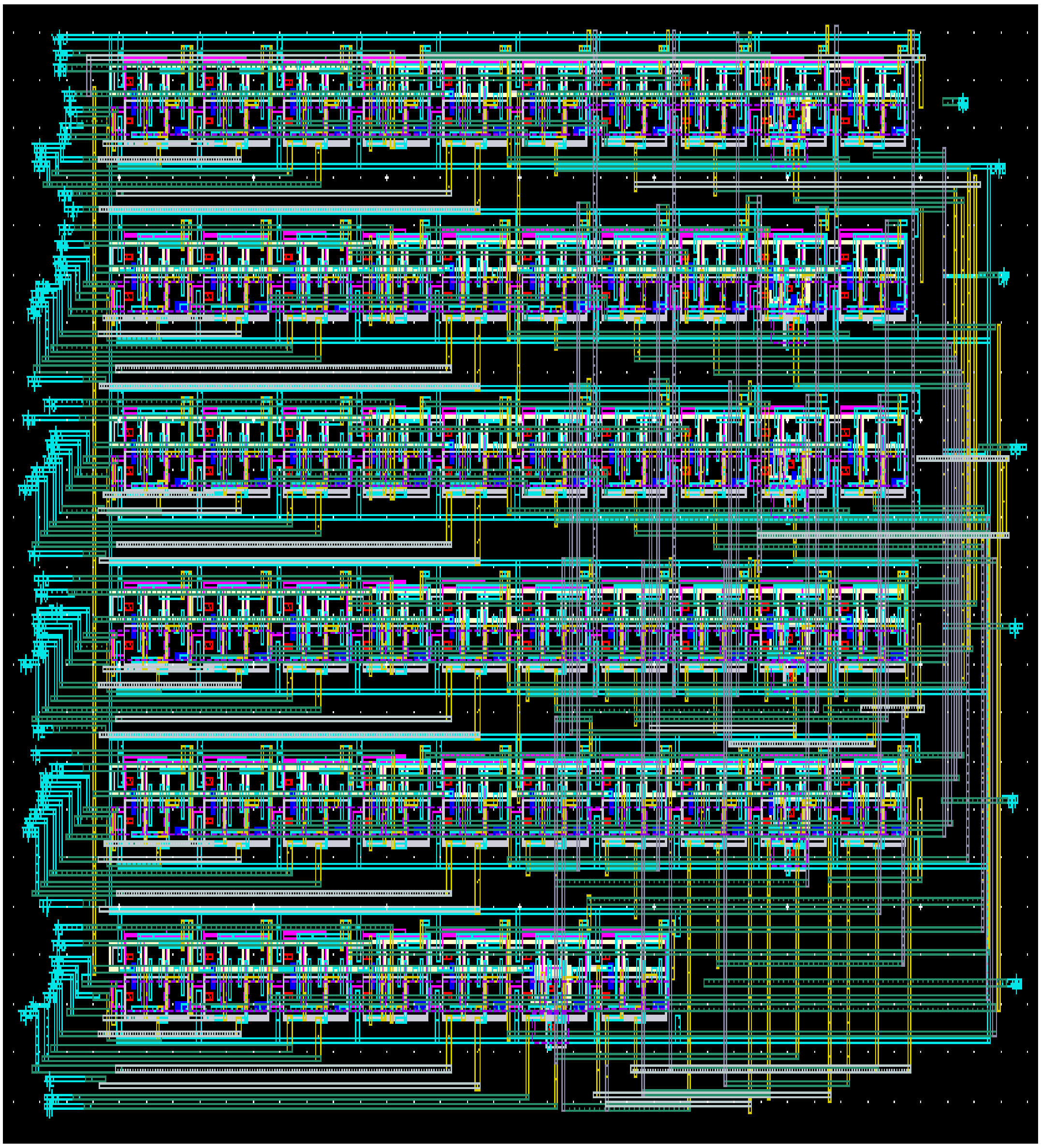


Figure 78 : Encoder Layout

7.2.3 Overall Chip Design Layout

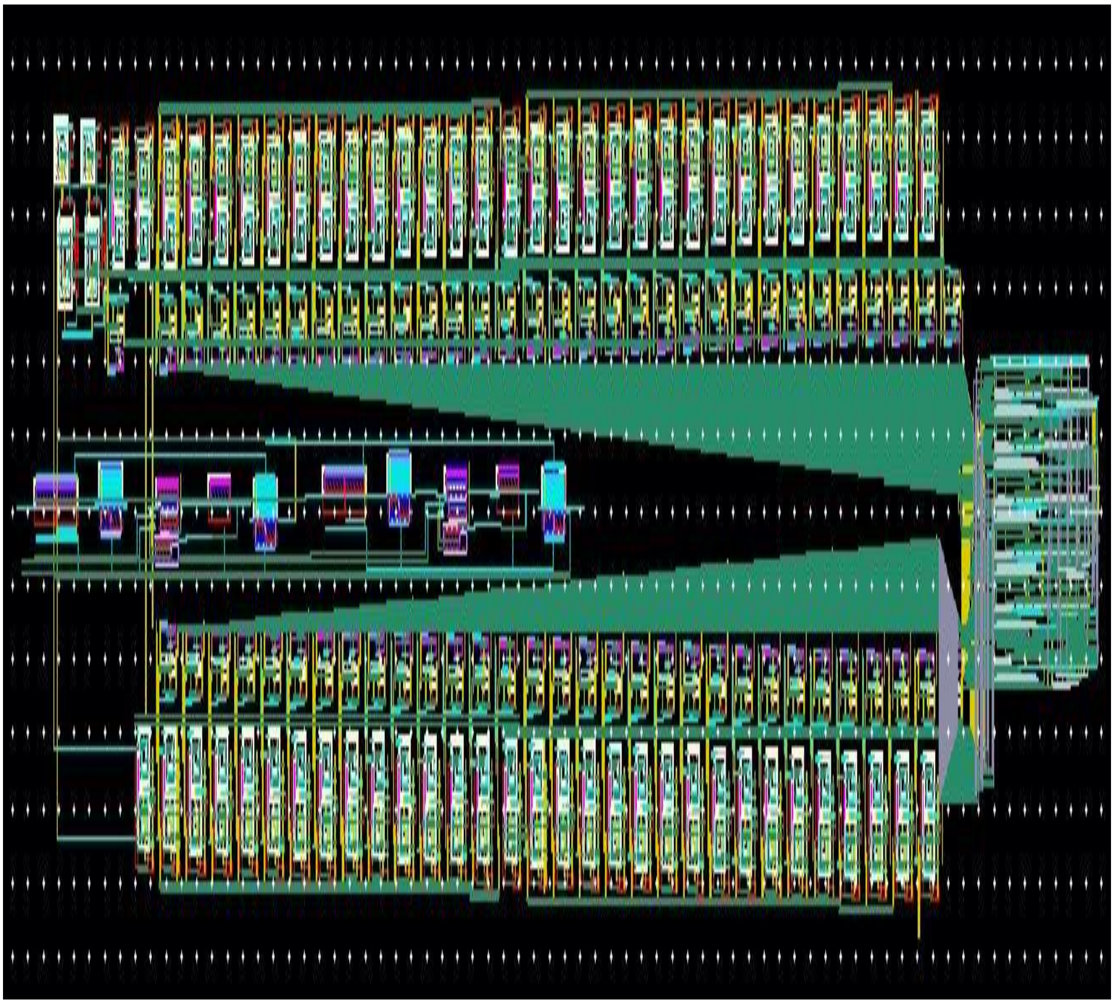


Figure 79 : Layout of overall system

Chapter 8: Conclusion

8.1 Summary

8.1.1 Simulation Results

1. First design achieved results

Technology	TSMC 65nm CMOS
Resolution	6 bits
ENOB	5.4 bits
B.W.	150 MHZ
Sampling rate	500 MS/s
Power supply	1.2 V
Input Dynamic Range	1080 mV Differential
Power Consumption	1.8 mW

Table 3 : Results of First Implementation of TADC

2. Second design achieved results

Technology	TSMC 65nm CMOS
Resolution	6 bits
ENOB	5.4 bits
B.W.	225 MHZ
Sampling rate	1 GS/s
Power supply	1 V
Input Dynamic Range	172.2 mV Single Ended
Power Consumption	1 mW

Table 4 : Results of Second Implementation of TADC

3. Optimization of area for First design

Technology	TSMC 65nm CMOS
resolution	6 bits
ENOB	3.6 bits
B.W.	80 MHZ
Sampling rate	400 MS/s
Power supply	1.2 V
Input Dynamic Range	1200 mV Differential

Table 5 : Results of optimization of First Implementation of TADC

8.1.2 Comparison between this work and others

Reference	Architecture	Technology Process	Power	Resolution	F_s	FOM_1 PJ/conv. step
[14]	Delay-Line	65 nm	2 mW	4	1.2 GS/s	0.143
[34]	Flash	90 nm	2.5 mW	4	1.25 GS/s	0.154
[35]	TI-SAR	65 nm	1.8 mW	5	500 MS/s	0.22
[36]	Folding Flash	90 nm	2.2 mW	5	1.75 GS/s	0.064
[37]	Flash	180 nm	78 mW	4	4 GS/s	1.76
[38]	Flash	130 nm	180 mW	6	1.6 GS/s	2.6
[39]	Two-Step Flash	130 nm	49 mW	6	1 GS/s	1.24
[40]	SAR	65 nm	6.7 mW	6	1 GS/s	0.21
[41]	SAR	65 nm	1.2 mW	5	250 MS/s	0.223
[42]	SAR	130 nm	32 mW	6	1.25 GS/s	0.46
[43]	Delay-Line	65 nm	1 mW	4	1 GS/s	0.105
TADC 1	Delay-Line	65 nm	1.8 mW	6	500 MS/s	0.085
TADC 2	VTC/TDC	65 nm	1 mW	6	1 GS/s	0.024

Table 6 : Comparison between different architectures

8.2 Future Work

We should take many points in our consideration in order to optimize and enhance the performance of TADC such as:

- Jitter Analysis, as it affects ENOB directly in both designs.
- Time Interleaving for first TADC Implementation.
- Resolve problems affecting ENOB of optimized implementation of first TADC implementation.
- Development and Implementation for hardware of digital calibration of second TADC implementation.
- For second implementation, it needs TDC programming for providing the circuit with accurate biasing voltages.
- Optimize Layout and overall area

References

[1] A. Annema, B. Nauta, R. van Langevelde, and H. Tuinhout, “Analog circuits in ultra-deep-submicron CMOS,” IEEE Journal of Solid-State Circuits, vol. 40, no. 1, pp. 132 – 143, January 2005.

[2] A. Matsuzawa, “Design challenges of analog-to-digital converters in nanoscale CMOS, IEICE Transactions on Electronics,” vol. 90, pp. 779–785, 2007.

[3] C. Taillefer, “Analog-to-digital conversion via time-mode signal processing,” Canada: McGill University, p. 186, 2007.

[4] Stephan Henzler, “Time-to-Digital Converters, Springer Series in Advanced Microelectronics,” vol. 29, 2010.

[5] Shahrzad Naraghi, Matthew Courcy, and Michael P. Flynn, “A 9-bit, 14 μW and 0.06 mm^2 Pulse Position Modulation ADC in 90 nm Digital CMOS,” IEEE JOURNAL OF SOLID-STATE CIRCUITS, vol. 45, no. 9, SEPTEMBER 2010.

[6] A. Oppenheime and A. Willsky, Signal and System, 2nd ed. Prentice Hall signal processing series, 2004.

[7] Sergio Rapuano, Pasquale Daponte, Eulalia Balestrieri, Luca De Vito, Steven J. Tilden, Solomon Max, and Jerome Blair, “ADC Parameters and Characteristics, Instrumentation & Measurement Magazine,” IEEE, vol. 8 Issue: 5, pp. 44 - 56, 23 January, 2006.

[8] Frank Goodenough, “Analog technology of all varieties dominate.” ISSCC, Electronic Design, vol. 44 Issue 4, p96, February 19, 1996.

[9] R. H. Walden, “Analog to Digital Converter Survey and Analysis, IEEE Journal on Selected Areas in Communications,” Vol. 17, pp.539550, 1999.

[10] S. Weaver, B. Hershberg, and Un-Ku Moon, “ENOB Calculation for ADCs with Input-Correlated Quantization Error Using a Sine-Wave Test,” Microelectronics ICM December 19, 2010.

[11] Howard Johnson, “Random and Deterministic Jitter,” EDN Magazine, June 2002.

[12] S. Lee, Y. Seo, H. Park, and J. Sim, “A 1 GHz ADPLL with a 1.25 ps minimum-resolution sub-exponent TDC in 0.18 m CMOS,” IEEE J. Solid-State Circuits, vol. 45, no. 12, pp. 2874–2881, Dec. 2010.

[13] L. Vercesi, A. Liscidini, and R. Castello, “Two-dimensions vernier time-to-digital converter,” IEEE J. Solid-State Circuits, vol. 45, no. 8, pp. 1504–1512, Aug. 2010.

[14] Yahya M. Tousi, Ehsan Afshari, “A Miniature 2 mW 4 bit 1.2 GS/s Delay-Line-Based ADC in 65 nm CMOS,” Solid-State Circuits, IEEE Journal, vol. 46, Issue 10, Oct. 2011.

[15] T.Watanabe, T.Mizuno, and Y. Makino, “An all-digital analog-to-digital converter with 12- V/LSB using moving-average filtering,” IEEE J. Solid-State Circuits, vol. 38, no. 1, pp. 120–125, Jan. 2003.

[16] M. Maymandi-Nejad and M. Sachdev, “A monotonic digitally controlled delay element,” IEEE J. Solid-State Circuits, vol. 40, no. 11, pp. 2212–2219, Nov. 2005.

[17] M. Morris Mano and Michael D. Ciletti, Digital Design, 4th ed., Dec. 2006.

[18]. Vinayashree Hiremath, Saiyu Ren, “An Ultra High Speed Encoder for 5GSPS Flash ADC,” IEEE Conference on Instrumentation and Measurement Technology, pp 136-141, May 2010.

[19] Yatish Lavania, George Tom Varghese and K. K. Mahapatra, “An Ultra-Low Power Encoder for 5 Bit Flash ADC,” IEEE, Jan. 2013.

[20] Gyudong Kim, Min-Kyu Kim, Byoung-Soo Chang, and Wonchan Kim, “ A Low-Voltage, Low-Power CMOS Delay Element,” IEEE Journal of SolidState Circuits, vol.31, no. 1, July 1996.

[21] John G. Maneatis, “Low-Jitter Process-Independent DLL and PLL Based on Self-Biased Techniques,” IEEE Transactions on Solid-State Circuits, vol. 31, no. 11 November 1996.

[22] Woogeun Rhee, A Low Power, “Wide Linear-Range CMOS Voltage- Controlled Oscillator,” IEEE International Symposium on Circuits and Systems, vol. 2, pp. 85-88, June 1998.

[23] N.R. Mahapatra, S.V. Garimella, and A. Tareen, “An Empirical and Analytical Comparison of Delay Elements and a New Delay Element Design,” Proceeding of the IEEE Computer Society Workshop on VLSI, pp. 81-86, April 2000.

[24] Dean A. Badillo and Sayfe Kiaei, “Comparison of Contemporary Ring Oscillators, IEEE Radio Frequency Integrated Circuits Symposium,” pp. 281-284, 2004.

[25] M. El-Hage and F. Yuan, “An Overview of Low-Voltage VCO Delay Cells and a Worst-Case Analysis of Supply Noise Sensitivity,” Canadian Conference on Electrical and Computer Engineering, vol. 3, pp 1785-1788, May 2004.

[26] H. Pekau, A. Yousif, and J.W. Haslett, “A CMOS Integrated Linear Voltage-to-Pulse-Delay-Time Converter for Time Based Analog-to-Digital Converters,” IEEE International Symposium on Circuits and Systems, pp. 2373-2376, May 2006.

[27] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, Digital integrated circuits-A design perspective. Prentice Hall, 2^{nd} ed, 2004.

[28] M. Matsui, “A 200MHZ 13 mm^2 2-D DCT Macrocell Using Sense-Amplifying Pipeline Flip-Flop Scheme,” IEEE journal of Solid circuits, vol. 29, issue 12, pp.1482-1490, Dec.1994.

[29] D.Dobberpuhl, “the Design of a high performance Low Power Microprocessor, Proceeding International Symposium on Low-Power Electronics and Design,” August 1996.

[30] B.Nikolic' V. Stojanovic' V.G. Oklobdz'ija W. jia, j .Chiu, M.Leung, “Sense Amplifier-Based Flip-Flop,” 1999. International Solid-State Circuits Conference, Digest of Technical Papers, ISSCC99 February 1999.

[31] A.P .Chandrakasan F.S.Lee,D.D. Wentzloff, V.Sze, B.P. Ginsburg, P.P.Mercier ,D.C. Daly, and R. Blazquez, "Low-Power Impulse UWB Architectures and Circuits," Proceedings of the IEEE, vol.97,pp.332-352,2009.

[32] E. Sall and M. Vesterbacka, “Comparison of two thermometer-to-binary decoders for high-performance flash ADCs,” in NORCHIP Conference, 23rd, pp.253-256, 2005.

[33] E.Sall and M.Vesterbacka, “A Multiplexer based decoder for flash Analog-to-Digital Converter,” in Proc. Tencom, pp.21-24, 2004.

[34] G. Van der Plas, S. Decoutere and S. Donnay, “A 0.16pJ/Conversion-Step 2.5mW 1.25GS/s 4b ADC in 90nm Digital CMOS Process,” ISSCC Dig. Tech. Papers, pp. 566-567, Feb. 2006.

[35] B. P. Ginsburg and A. P. Chandrakasan, “500-MS/s 5-bit ADC in 65-nm CMOS with Split Capacitor Array DAC,” IEEE J. Solid-State Circuits, vol. 42, pp. 739-747, April. 2007.

[36] B. Verburggen, J. Cranincks, M. Kuijk, P. Wambacq and G. V. Plas, “A 2.2mW 5b 1.75GS/s Folding Flash ADC in 90nm Digital CMOS,” ISSCC Dig. Tech. Papers, pp. 252-253, Feb. 2008.

[37] S. Park, Y. Palaskas, and M. P. Flynn, “A 4-GS/s 4-bit flash ADC in 0.18 µm CMOS,” IEEE J. Solid-State Circuits, vol. 42, no. 9, pp.1865–1872, Sep. 2007.

[38] A. Ismail and M. Elmasry, “A 6-bit 1.6-GS/s low-power wideband flash ADC converter in 0.13 µm CMOS technology,” IEEE J. Solid-State Circuits, vol. 43, no. 9, pp. 1982–1990, Sep. 2008.

[39] H. Chen, I. Chen, H. Tseng, and H. Chen, “A 1-GS/s 6-bit two-channel two-step ADC in 0.13 µm CMOS,” IEEE J. Solid-State Circuits, vol. 44, no. 11, pp. 3051–3059, Nov. 2009.

[40] J. Yang, T. L. Niang, and R. W. Broderson, “A 1 GS/s 6 bit 6.7 mW successive approximation ADC using asynchronous processing,” IEEE J. Solid-State Circuits, vol. 45, no. 8, pp. 1469–1478, Aug.2010.

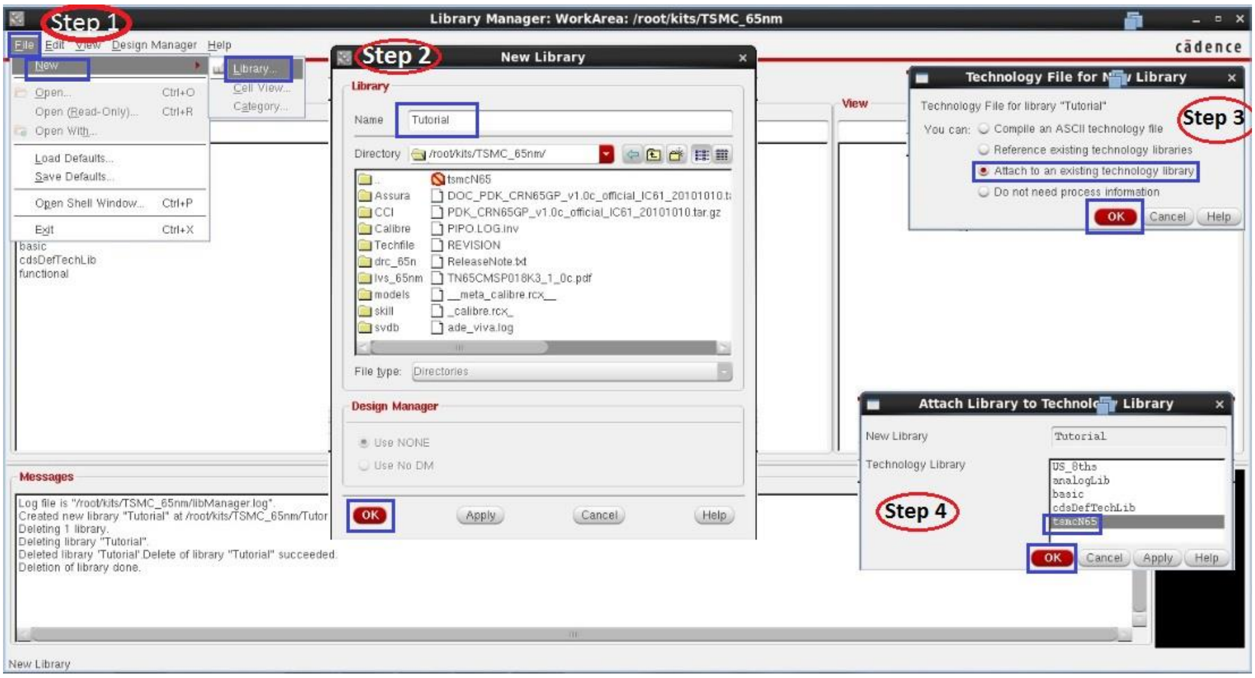
[41] B. P. Ginsburg and A. P. Chandrakasan, “Highly interleaved 5-bit, 250-MSample/s, 1.2 mW ADC with redundant channels in 65-nm CMOS,” IEEE J. Solid-State Circuits, vol. 43, no. 12, pp. 2641–2650, Dec.2008.

[42] Z. Cao, S. Yan, and Y. Li, “A 32 mW 1.25 GS/s 6 bit 2 b/Step SAR ADC in 0.13 µm CMOS,” IEEE J. Solid-State Circuits, vol. 44, no. 3, pp. 862–873, Apr. 2009.

[43] Yahya M.Tousi, Guansheng Li, Arjang Hassibi, and Ehsan Afshari, “A 1mW 4b 1GS/s Delay Line based Analog-to-Digital Converter,” Circuits and Systems, 2009.

Appendix A: Steps towards tape-out of an IC

1. In order to create new library in cadence 6.1x:



New Library

Library

Name: Tutorial

Directory: /root/kits/TSMC_65nm/

..	tsmcN65
Assura	DOC_PDK_CRN65GP_v1.0c_official_IC61_20101010.t
CCI	PDK_CRN65GP_v1.0c_official_IC61_20101010.tar.gz
Calibre	PIPO.LOG.inv
Techfile	REVISION
drc_65n	ReleaseNote.txt
lvs_65nm	TN65CMSP018K3_1_0c.pdf
models	__meta_calibre.rcx__
skill	_calibre.rcx_
svdb	ade_viva.log

File type: Directories

Design Manager

● Use NONE

○ Use No DM

Messages

```
Log file is "/root/kits/TSMC_65nm/libManager.log".
Created new library "Tutorial" at /root/kits/TSMC_65nm/Tutor
Deleting 1 library.
Deleting library "Tutorial".
Deleted library 'Tutorial'.Delete of library "Tutorial" succeeded.
Deletion of library done.
```

New Library

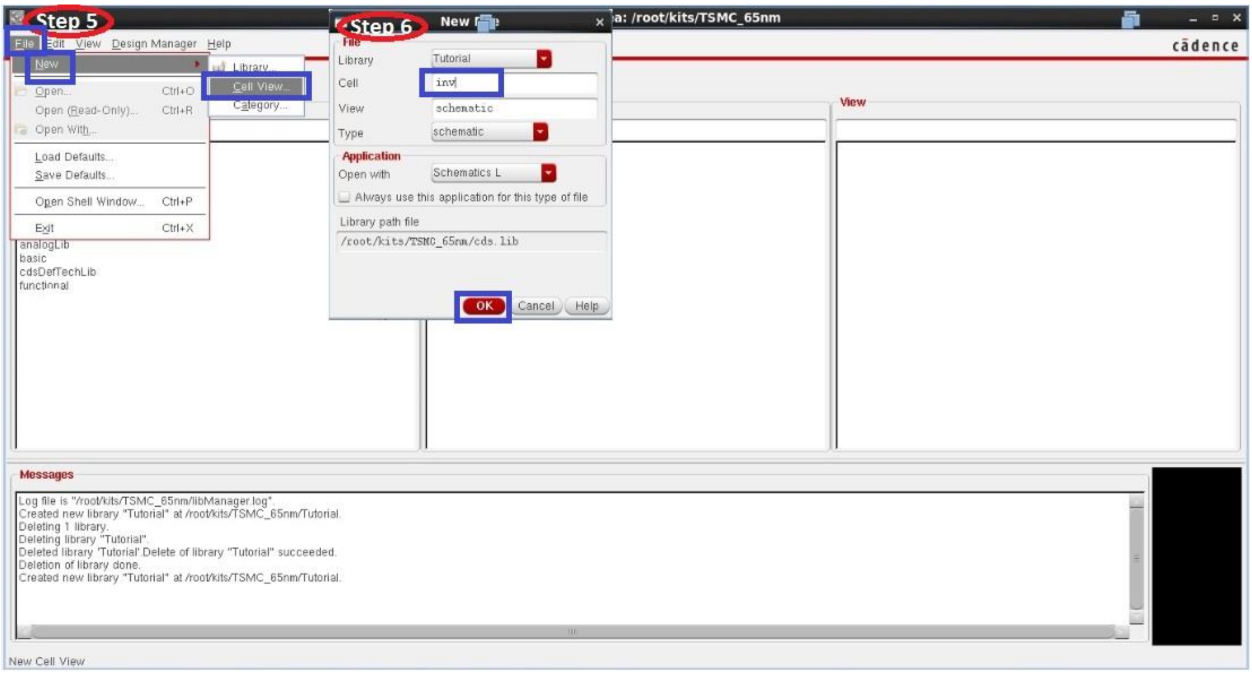
Technology File for library "Tutorial"

You can:
○ Compile an ASCII technology file
○ Reference existing technology libraries
● Attach to an existing technology library
○ Do not need process information

New Library: Tutorial

Technology Library:
US_8ths
analogLib
basic
cdsDefTechLib
tsmcN65

2. In order to create new cell view to draw your own schematic:



File

Library: Tutorial

Cell: inv

View: schematic

Type: schematic

Application

Open with: Schematics L

□ Always use this application for this type of file

Library path file: /root/kits/TSMC_65nm/cds.lib

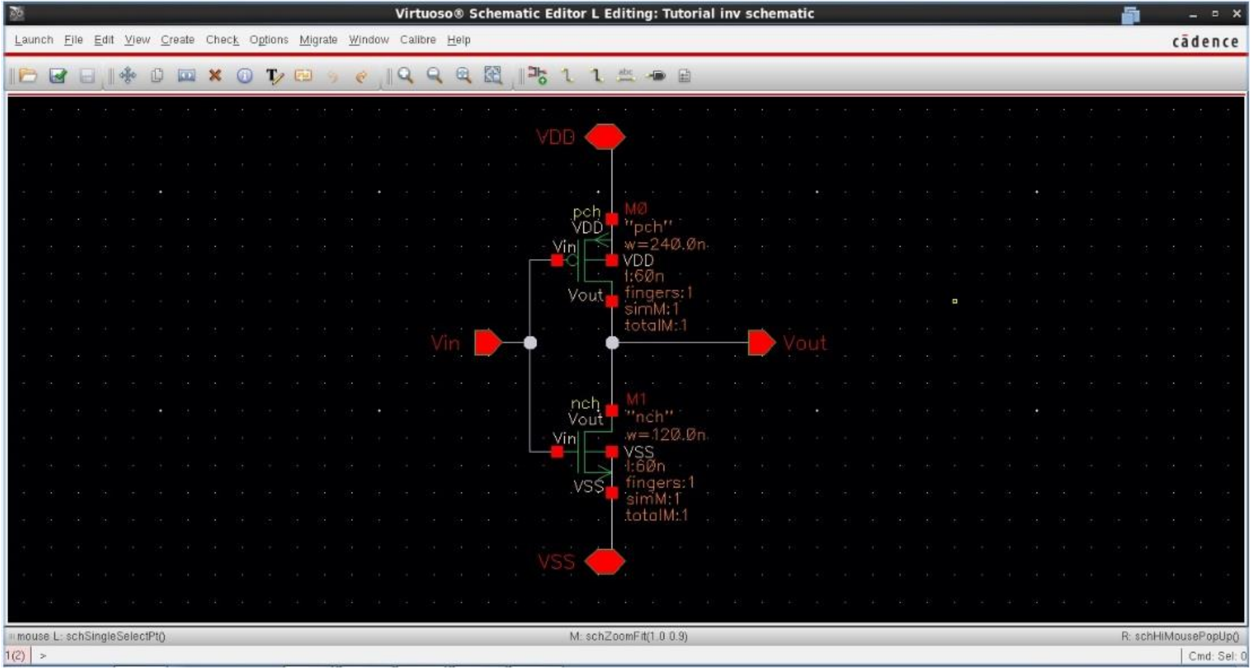
Messages

```
Log file is "/root/kits/TSMC_65nm/libManager.log".
Created new library "Tutorial" at /root/kits/TSMC_65nm/Tutorial.
Deleting 1 library.
Deleting library "Tutorial".
Deleted library 'Tutorial'.Delete of library "Tutorial" succeeded.
Deletion of library done.
Created new library "Tutorial" at /root/kits/TSMC_65nm/Tutorial.
```

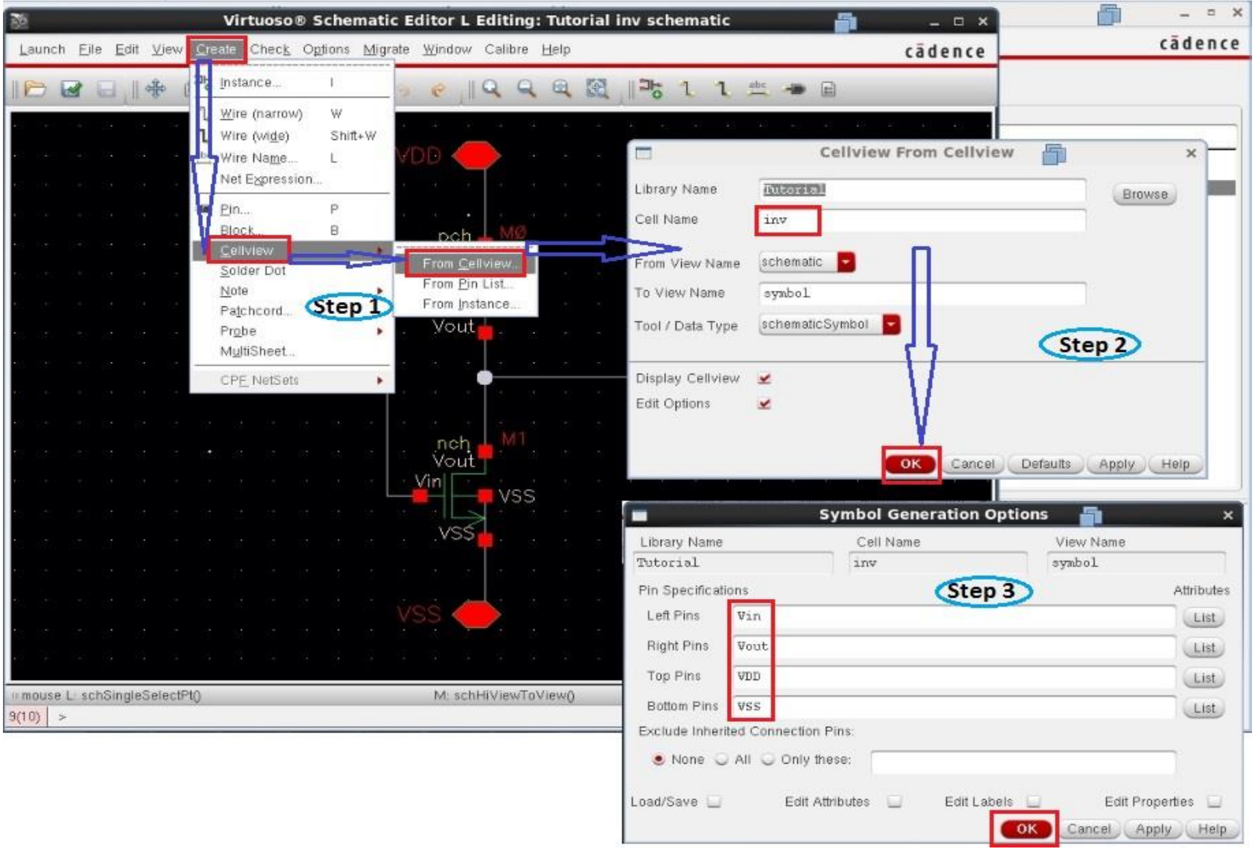
New Cell View

3. Begin by drawing schematic:

Inserting components → press “i”
Change component parameter → press “Q”
For wiring between different components → press “w”
For make I/O pins → press “p”
Here is an inverter shown as example



4. Create symbol from this schematic:



Virtuoso® Schematic Editor L Editing: Tutorial inv schematic

Step 1

Step 2

Step 3

Cellview From Cellview

Library Name	Tutorial
Cell Name	inv
From View Name	schematic
To View Name	symbol
Tool / Data Type	schematicSymbol
Display Cellview	☑
Edit Options	☑

Symbol Generation Options

Library Name	Cell Name	View Name
Tutorial	inv	symbol

Pin Specifications

Left Pins	Vin
Right Pins	Vout
Top Pins	VDD
Bottom Pins	VSS

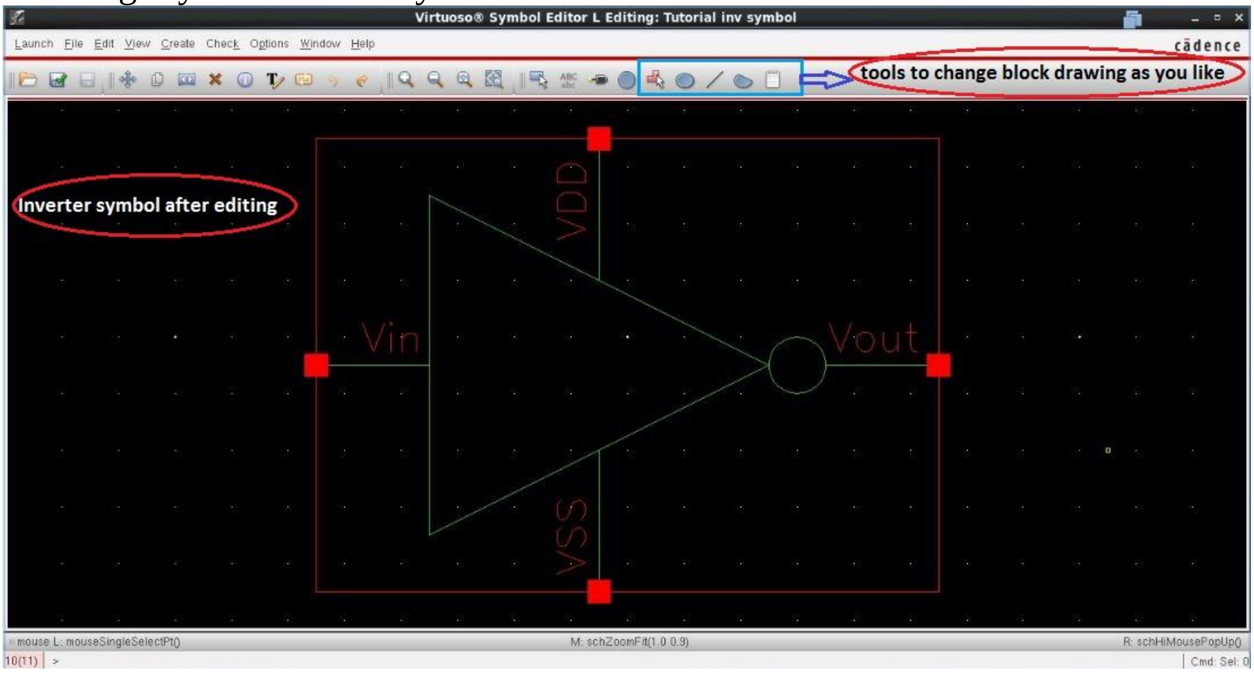
Exclude Inherited Connection Pins: ● None ○ All ○ Only these:

Load/Save □ Edit Attributes □ Edit Labels □ Edit Properties □

mouse L: schSingleSelectPt() M: schHiViewToView()

9(10) >

5. Change symbol block as you like as shown:



Virtuoso® Symbol Editor L Editing: Tutorial inv symbol

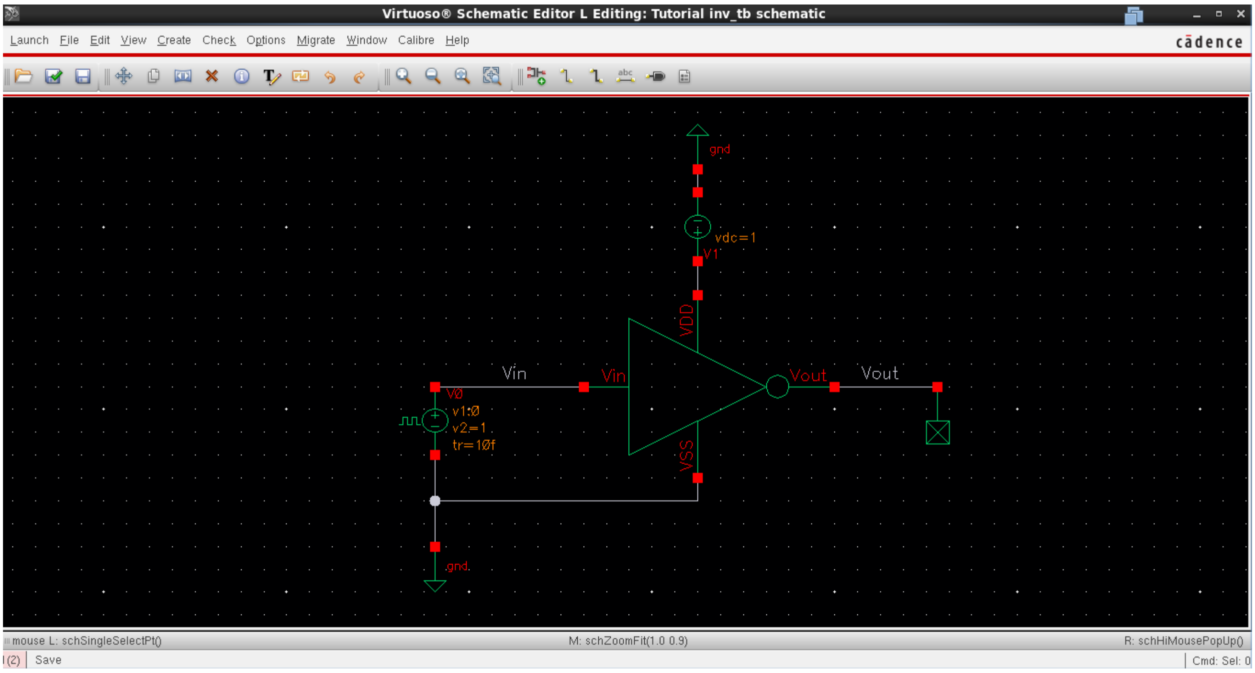
tools to change block drawing as you like

Inverter symbol after editing

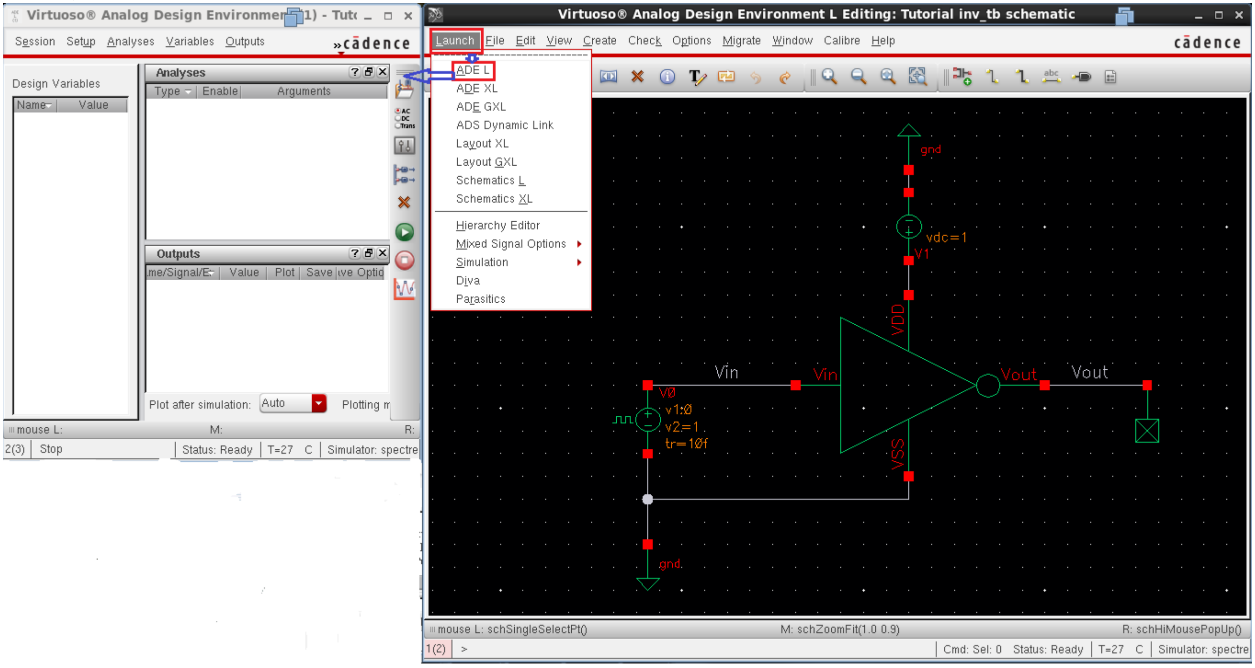
mouse L: mouseSingleSelectPt() M: schZoomFit(1.0 0.9) R: schHiMousePopUp()

10(11) > Cmd: Sel: 0

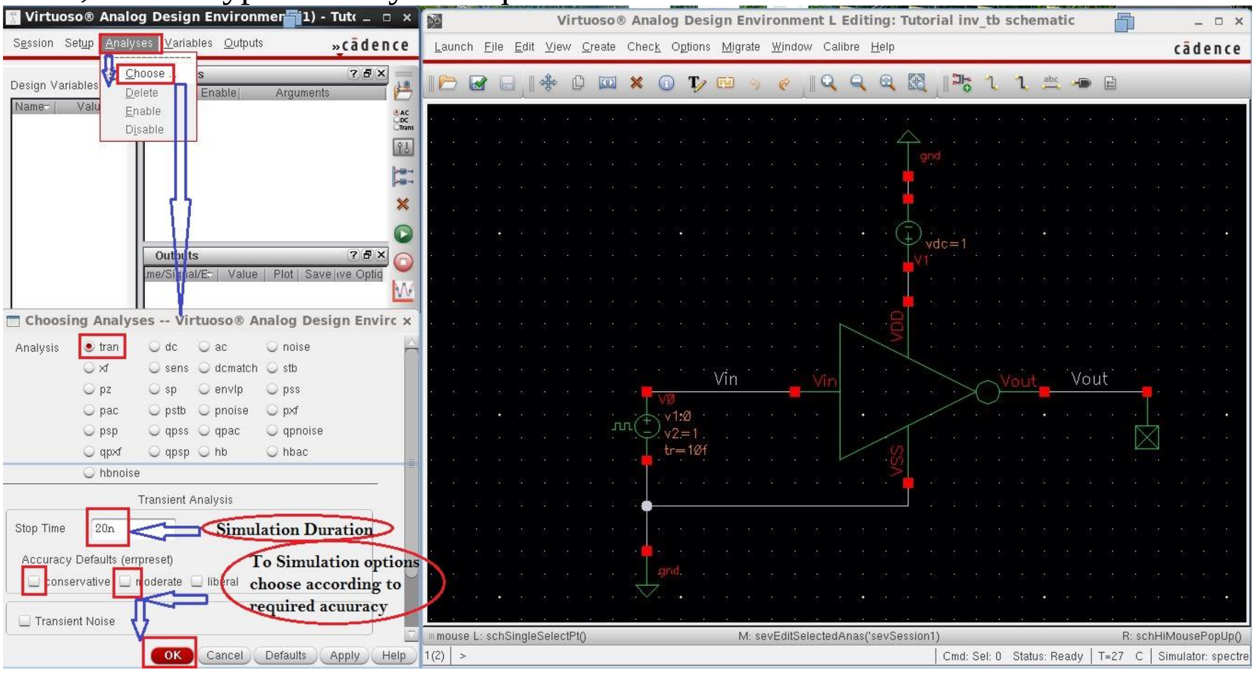
6. Now, we want to make a testbench file, so we will create new cellview as shown:
and we can pick ideal voltage sources from analoglib as vpulse,vdc,vsin,..


Virtuoso® Schematic Editor L Editing: Tutorial inv_tb schematic
Launch File Edit View Create Check Options Migrate Window Calibre Help
cādence
gnd
vdc=1
V1
VDD
Vin
Vin
Vout
Vout
V0
v1:0
v2=1
tr=10f
VSS
gnd
mouse L: schSingleSelectPt()
M: schZoomFit(1.0 0.9)
R: schHiMousePopUp()
1(2) Save
Cmd: Sel: 0

7. for simulation of this cell view:
First, Launch ADE L as shown:


Virtuoso® Analog Design Environment (1) - Tut
Session Setup Analyses Variables Outputs
»cādence
Design Variables
Name Value
Analyses
Type Enable Arguments
Outputs
Name/Signal/Expr Value Plot Save Save Options
Plot after simulation: Auto
Plotting mode
mouse L:
M:
R:
2(3) Stop
Status: Ready T=27 C Simulator: spectre
Virtuoso® Analog Design Environment L Editing: Tutorial inv_tb schematic
Launch File Edit View Create Check Options Migrate Window Calibre Help
cādence
ADE L
ADE XL
ADE GXL
ADS Dynamic Link
Layout XL
Layout GXL
Schematics L
Schematics XL
Hierarchy Editor
Mixed Signal Options
Simulation
Diva
Parasitics
gnd
vdc=1
V1
VDD
Vin
Vin
Vout
Vout
V0
v1:0
v2=1
tr=10f
VSS
gnd
mouse L: schSingleSelectPt()
M: schZoomFit(1.0 0.9)
R: schHiMousePopUp()
1(2) >
Cmd: Sel: 0 Status: Ready T=27 C Simulator: spectre

Then, choose type of analyses required:



Virtuoso® Analog Design Environment (1) - Tutc
Session Setup Analyses Variables Outputs
cādence
Choose ...
Delete
Enable
Disable
Design Variables
Name Value
Enable Arguments
Outputs
Name/Signal/Expr Value Plot Save Save Options
Virtuoso® Analog Design Environment L Editing: Tutorial inv_tb schematic
Launch File Edit View Create Check Options Migrate Window Calibre Help
gnd
vdc=1
V1
VDD
Vin
Vin
Vout
Vout
V0
v1:0
v2=1
tr=10f
VSS
gnd
mouse L: schSingleSelectPt()
M: sevEditSelectedAnas('sevSession1)
R: schHiMousePopUp()
1(2) >
Cmd: Sel: 0 Status: Ready T=27 C Simulator: spectre

Choosing Analyses -- Virtuoso® Analog Design Environment

Analysis ● tran ○ dc ○ ac ○ noise ○ xf ○ sens ○ dcmatch ○ stb ○ pz ○ sp ○ envlp ○ pss ○ pac ○ pstb ○ pnoise ○ pxf ○ psp ○ qpss ○ qpac ○ qpnoise ○ qpxf ○ qpsp ○ hb ○ hbac ○ hbnoise

Transient Analysis

Stop Time 20n

Simulation Duration

Accuracy Defaults (errpreset)

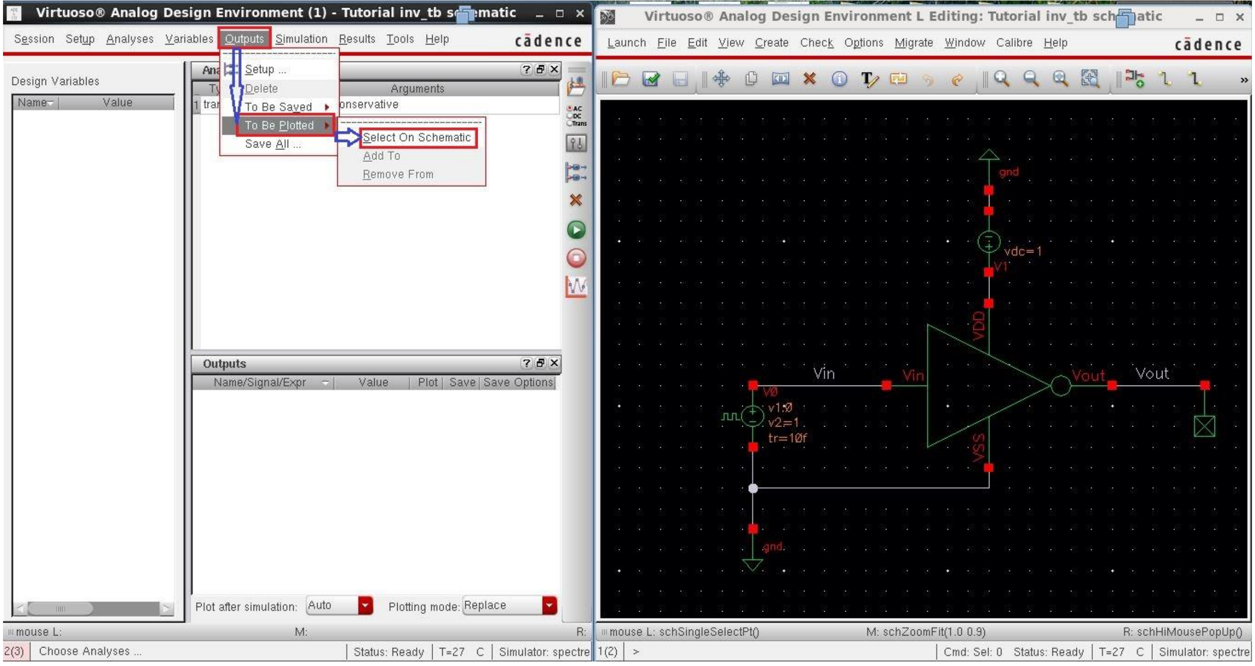
□ conservative □ moderate □ liberal

To Simulation options choose according to required acuuracy

□ Transient Noise

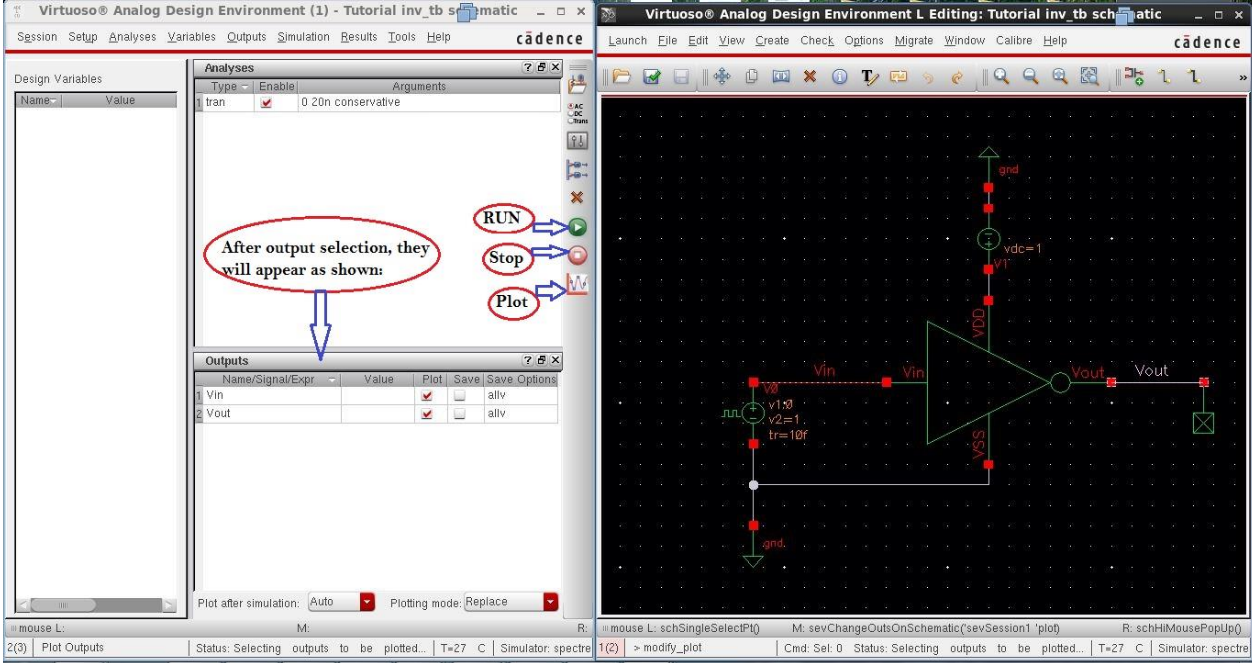
OK Cancel Defaults Apply Help

After that, choose outputs need to be plotted as shown:



Virtuoso® Analog Design Environment (1) - Tutorial inv_tb schematic
Session Setup Analyses Variables Outputs Simulation Results Tools Help
cādence
Design Variables
Name Value
Setup ...
Delete
To Be Saved
To Be Plotted
Save All ...
Select On Schematic
Add To
Remove From
Arguments
onservative
tran
Outputs
Name/Signal/Expr Value Plot Save Save Options
Plot after simulation: Auto
Plotting mode: Replace
mouse L:
M:
R:
2(3) Choose Analyses ...
Status: Ready T=27 C Simulator: spectre
Virtuoso® Analog Design Environment L Editing: Tutorial inv_tb schematic
Launch File Edit View Create Check Options Migrate Window Calibre Help
gnd
vdc=1
V1
VDD
Vin
Vin
Vout
Vout
V0
v1:0
v2=1
tr=10f
VSS
gnd
mouse L: schSingleSelectPt()
M: schZoomFit(1.0 0.9)
R: schHiMousePopUp()
1(2) >
Cmd: Sel: 0 Status: Ready T=27 C Simulator: spectre

Finally, you can simulate your testbench file:



Virtuoso® Analog Design Environment (1) - Tutorial inv_tb schematic

Analyses

	Type	Enable	Arguments
1	tran	✓	0 20n conservative

After output selection, they will appear as shown:

Outputs

	Name/Signal/Expr	Value	Plot	Save	Save Options
1	Vin		✓	□	allv
2	Vout		✓	□	allv

Plot after simulation: Auto   Plotting mode: Replace

Status: Selecting outputs to be plotted... | T=27 C | Simulator: spectre

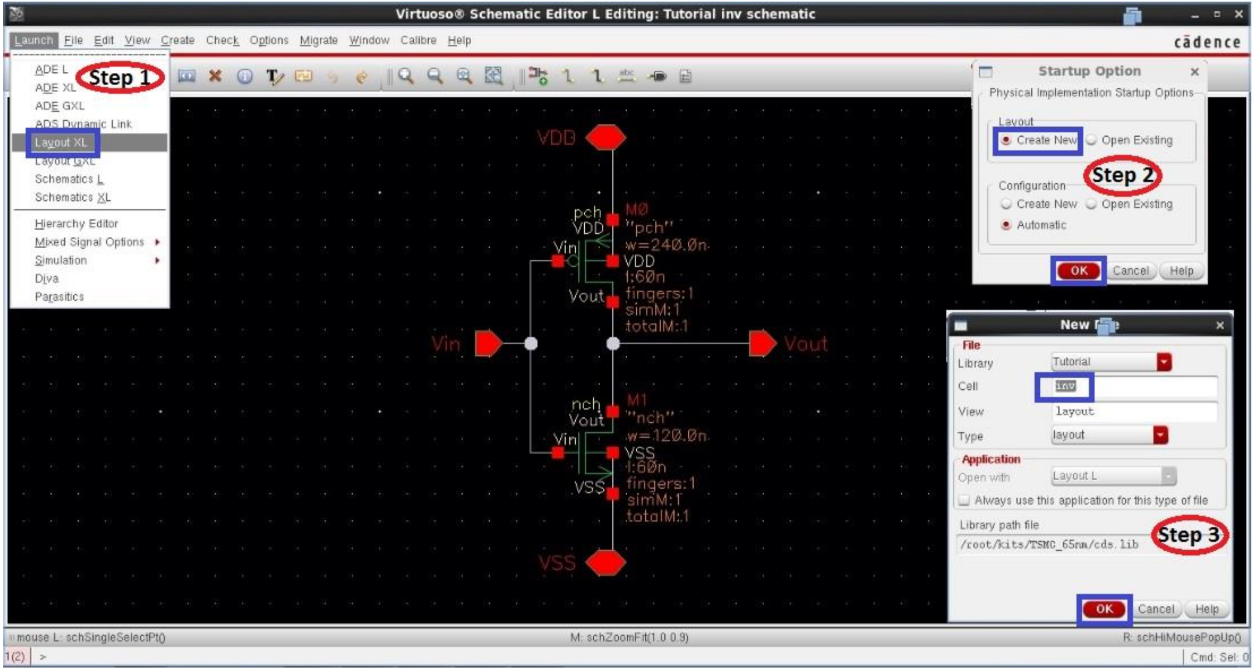
Virtuoso® Analog Design Environment L Editing: Tutorial inv_tb schematic

mouse L: schSingleSelectPt()   M: sevChangeOutsOnSchematic('sevSession1 'plot)   R: schHiMousePopUp()

Cmd: Sel: 0   Status: Selecting outputs to be plotted... | T=27 C | Simulator: spectre

Session Setup Analyses Variables Outputs Simulation Results Tools Help
cadence
Design Variables
Name Value
RUN
Stop
Plot
mouse L:
M:
R:
2(3) Plot Outputs
Launch File Edit View Create Check Options Migrate Window Calibre Help
gnd
vdc=1
V1
VDD
Vin
Vin
Vout
Vout
V0
v1:0
v2=1
tr=10f
VSS
gnd
1(2) > modify_plot

8. After, Creating symbol to use inverter as building block like any other component, we need to make layout for this schematic as shown:



Virtuoso® Schematic Editor L Editing: Tutorial inv schematic

Step 1: Launch → Layout XL

Startup Option

Physical Implementation Startup Options

Layout: ● Create New ○ Open Existing

Step 2

Configuration: ○ Create New ○ Open Existing ● Automatic

OK

New File

File
Library: Tutorial
Cell: inv
View: layout
Type: layout

Application
Open with: Layout L
□ Always use this application for this type of file

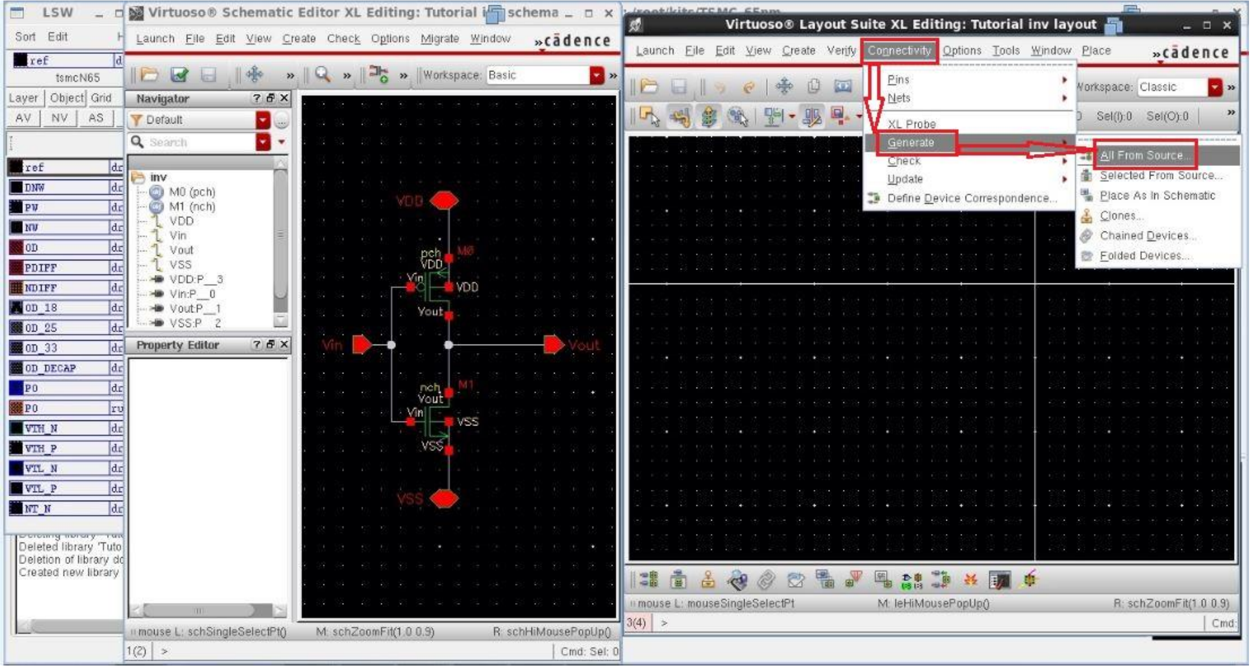
Library path file: /root/kits/TSMC_65nm/cds.lib

Step 3

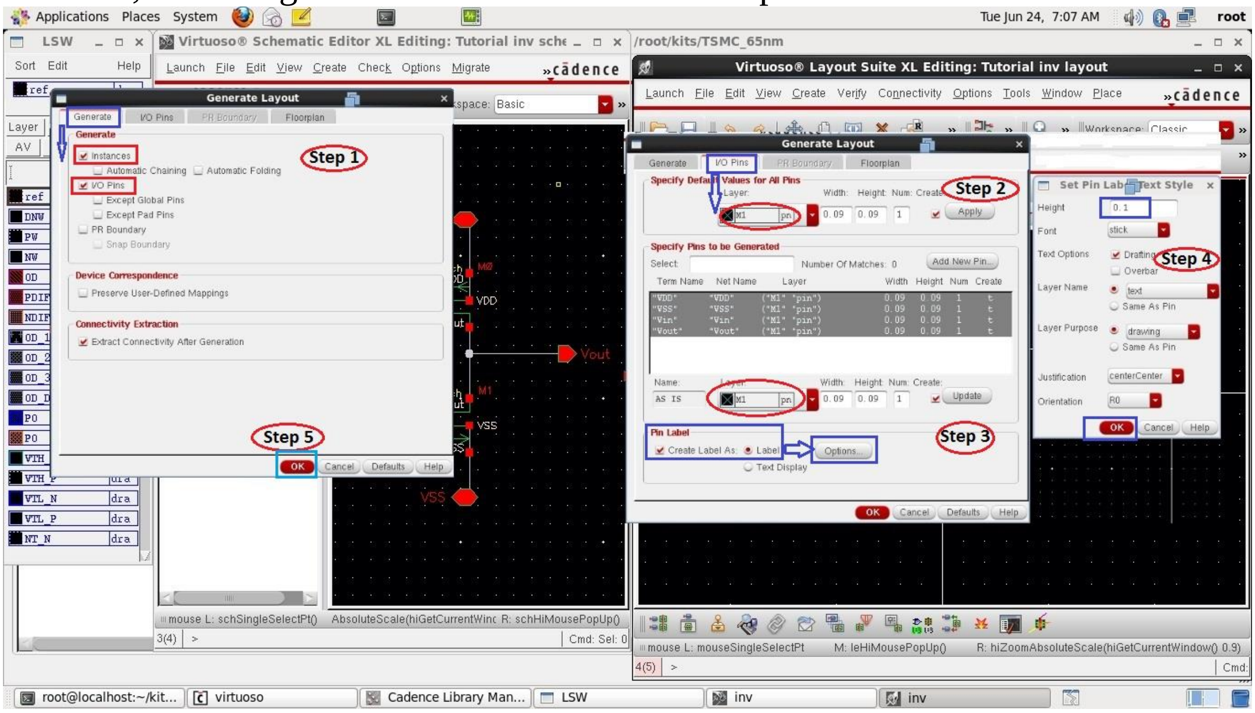
OK

mouse L: schSingleSelectPt()   M: schZoomFit(1.0 0.9)   R: schHiMousePopUp()

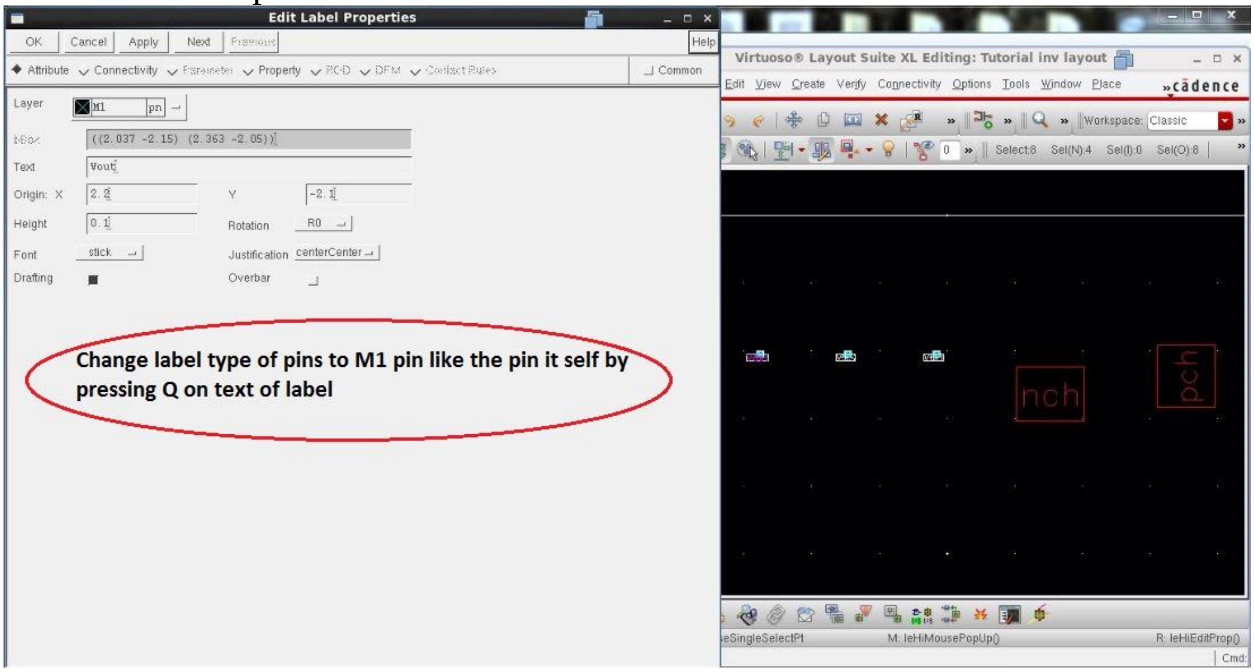
9. After, Layout extractor appear as shown we need to generate components like next step:



10. Now, We will generate our transistors and I/O pins



11. Label of I/O pins still need to be modified as shown:



Edit Label Properties

Layer: M1 pn

bBox: ((2.037 -2.15) (2.363 -2.05))

Text: Vout

Origin: X 2.2 Y -2.1

Height: 0.1 Rotation: R0

Font: stick Justification: centerCenter

Drafting ■ Overbar □

Change label type of pins to M1 pin like the pin it self by pressing Q on text of label

12. Transistor internal structure is hidden,
In order to make it appear → press "shift + f"
And you will find some difficulty while moving components, so need to edit display options as shown:

13. To make interconnections between transistors with each other and with I/O pins, we use following steps:

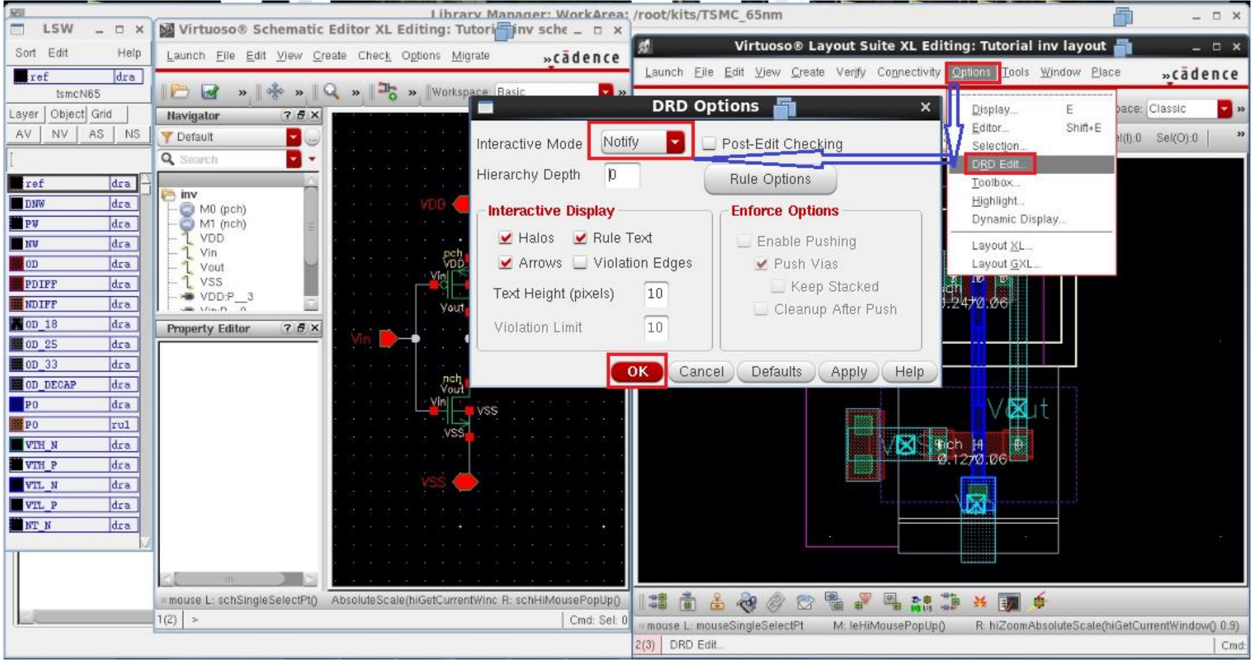
I. To make a path from node to another make check that you select required metal layer and beside it written that is used for drawing "dra" ,
Then press → "p" and make your path

II. To zoom on components press → "cntrl + z"

III. To stretch edge of path just press → "s" followed by one left click, then start to move

IV. For ruler to check spacing press → "k",
after finishing press → "shift + k "

V. To be notified in case of DRC violation takes place, activate this option as shown:



DRD Options

Interactive Mode: Notify ☐ Post-Edit Checking

Hierarchy Depth: 0 Rule Options

Interactive Display: ☑ Halos ☑ Rule Text ☑ Arrows ☐ Violation Edges

Text Height (pixels): 10

Violation Limit: 10

Enforce Options: ☐ Enable Pushing ☑ Push Vias ☐ Keep Stacked ☐ Cleanup After Push

OK Cancel Defaults Apply Help

Library Manager: WorkArea: /root/kits/TSMC_65nm
LSW
Virtuoso® Schematic Editor XL Editing: Tutorial inv schematic
Virtuoso® Layout Suite XL Editing: Tutorial inv layout
Launch File Edit View Create Verify Connectivity Options Tools Window Place
Display... E
Editor... Shift+E
Selection...
DRD Edit...
Toolbox...
Highlight...
Dynamic Display...
Layout XL...
Layout GXL...
cadence
Navigator
Property Editor
VDD
Vin
Vout
VSS
2(3) DRD Edit...

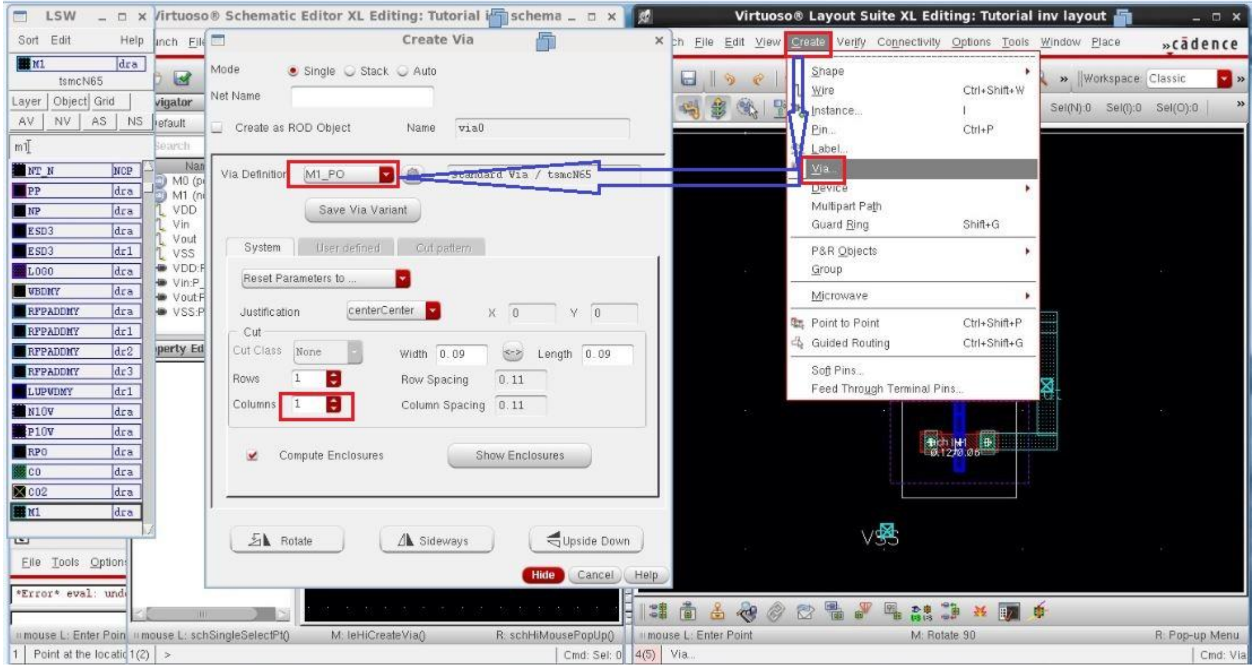
VI. To make via follow shown figure as we need to make via in different cases:

→ Between to different metal layers

→ Between Metal 1 and Substrate: to create bulk terminal of NMOS transistor

→ Between Metal 1 and N-Well: to create bulk terminal of PMOs transistor

→Between Metal layer and poly



Create Via

Mode: ● Single ○ Stack ○ Auto

Net Name:

☐ Create as ROD Object Name: via0

Via Definition: M1_PO Standard Via / tsmcN65

Save Via Variant

System | User defined | Cut pattern

Reset Parameters to ...

Justification: centerCenter X 0 Y 0

Cut

Cut Class: None Width 0.09 Length 0.09

Rows: 1 Row Spacing 0.11

Columns: 1 Column Spacing 0.11

☑ Compute Enclosures Show Enclosures

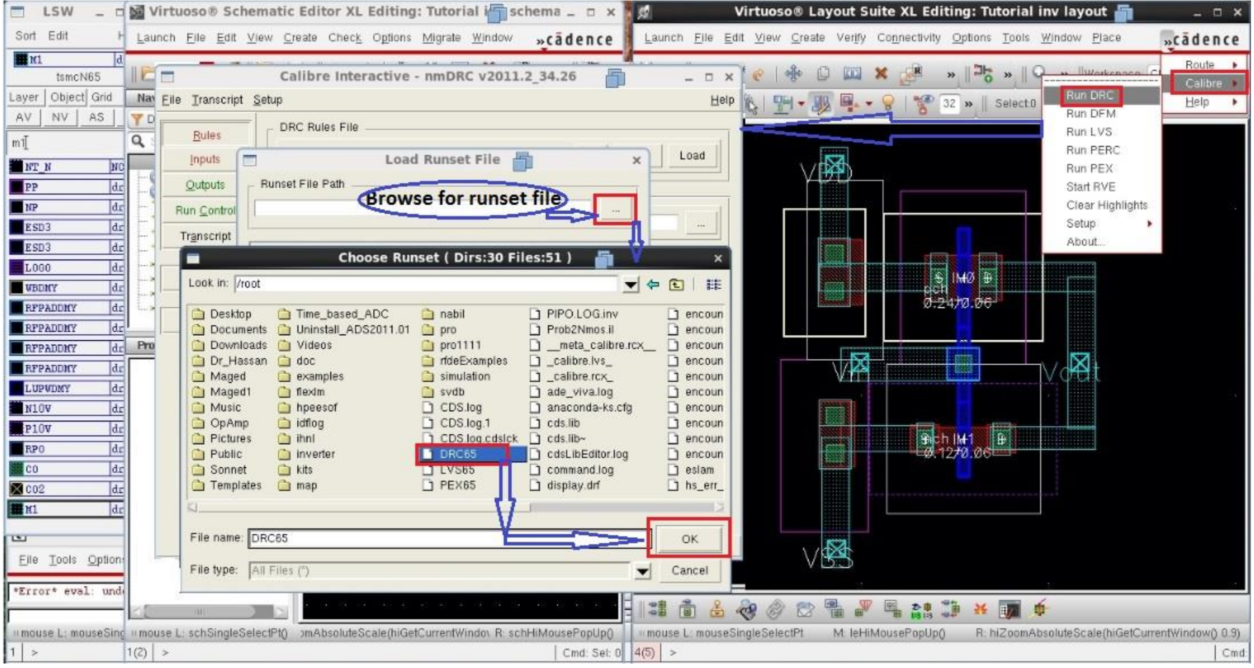
Rotate Sideways Upside Down

Hide Cancel Help

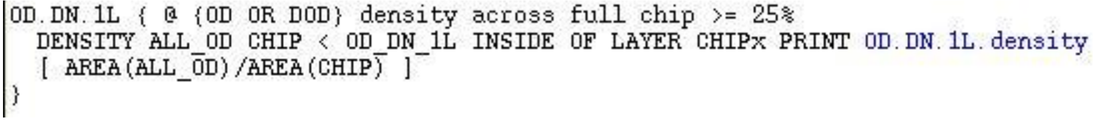
LSW
Virtuoso® Schematic Editor XL Editing: Tutorial inv schematic
Virtuoso® Layout Suite XL Editing: Tutorial inv layout
File Edit View Create Verify Connectivity Options Tools Window Place
Shape
Wire Ctrl+Shift+W
Instance... I
Pin... Ctrl+P
Label...
Via...
Device
Multipart Path
Guard Ring Shift+G
P&R Objects
Group
Microwave
Point to Point Ctrl+Shift+P
Guided Routing Ctrl+Shift+G
Soft Pins...
Feed Through Terminal Pins...
cadence
Workspace: Classic
4(5) Via...

14. Now, after finish drawing layout we need to run different checks that we mentioned before:

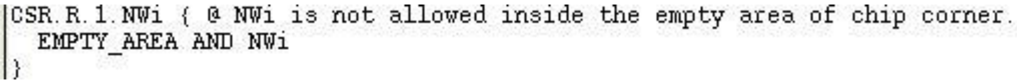
First, DRC is done as shown figure:



We will talk about common errors that would appear with DRC:
1-Errors that can be neglected like the following
Anything contain “Density or CSR”

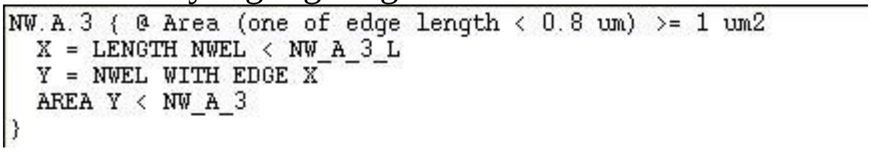


```
OD.DN.1L { @ {OD OR DOD} density across full chip >= 25%
  DENSITY ALL_OD CHIP < OD_DN_1L INSIDE OF LAYER CHIPx PRINT OD.DN.1L.density
  [ AREA(ALL_OD)/AREA(CHIP) ]
}
```



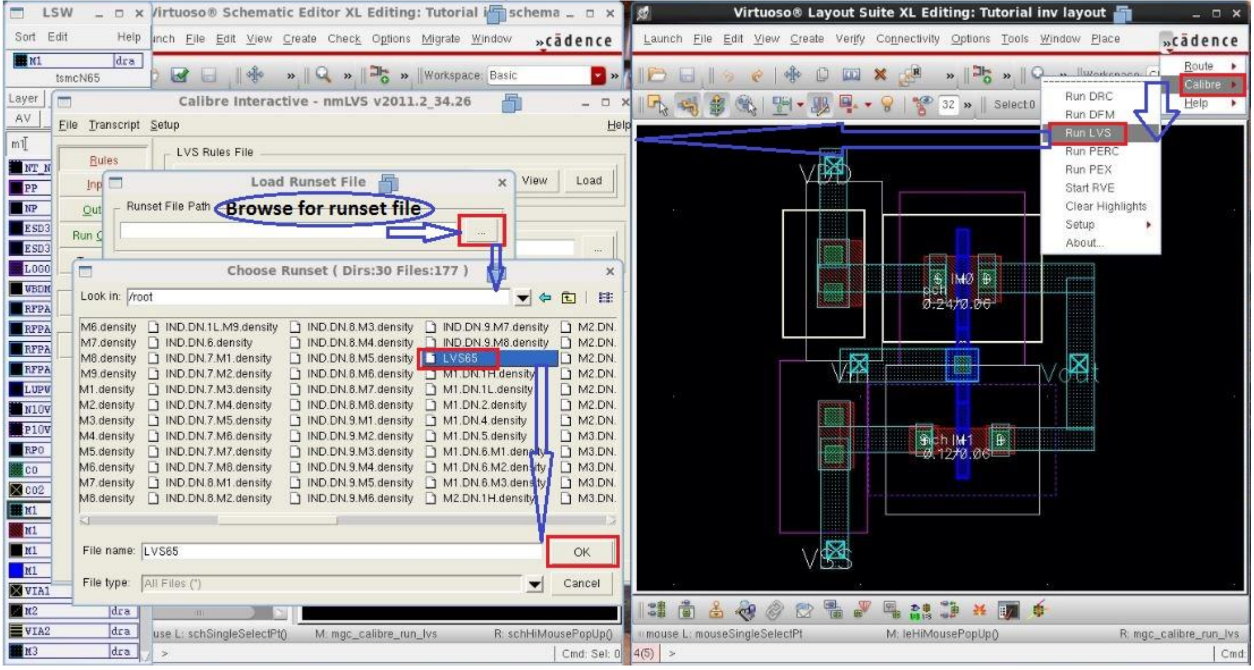
```
CSR.R.1.NWi { @ NWi is not allowed inside the empty area of chip corner.
  EMPTY_AREA AND NWi
}
```

2-Errors that is related to dimensions like following can’t be neglected just close layout view and press on error → layout viewer will open automatically highlighting on error to be resolved.

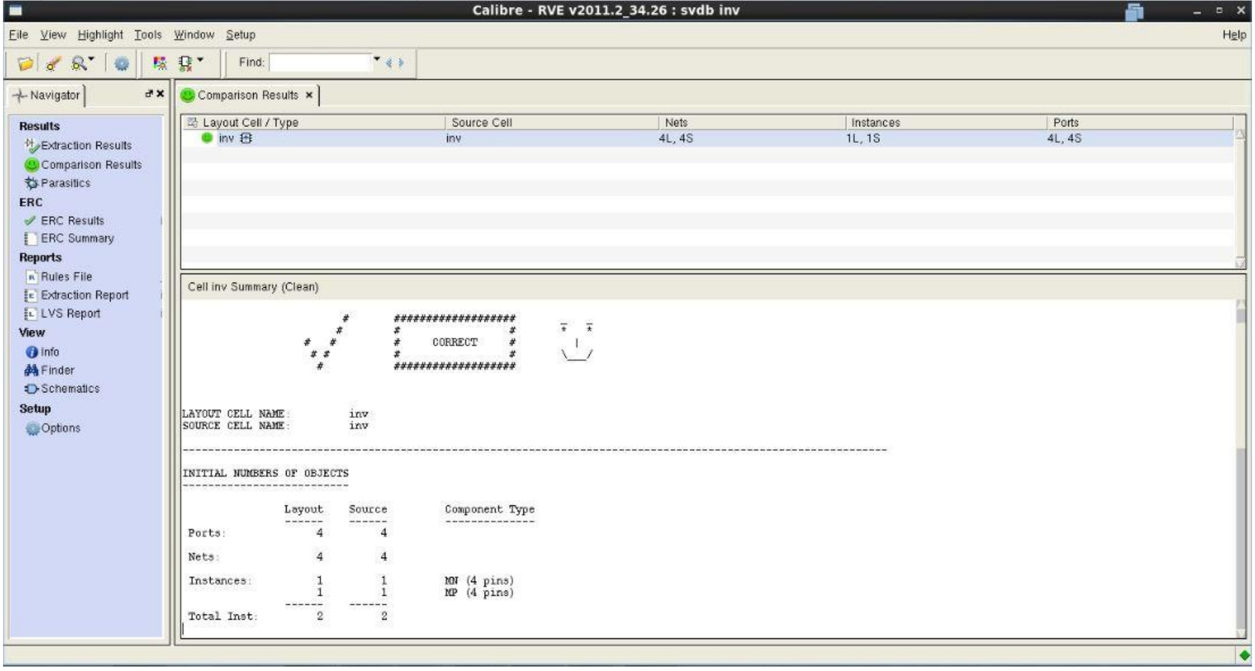


```
NW.A.3 { @ Area (one of edge length < 0.8 um) >= 1 um2
  X = LENGTH NWEL < NW_A_3_L
  Y = NWEL WITH EDGE X
  AREA Y < NW_A_3
}
```

Second, LVS is done as shown figure:



If your connections were right, you get this smiley face as shown:

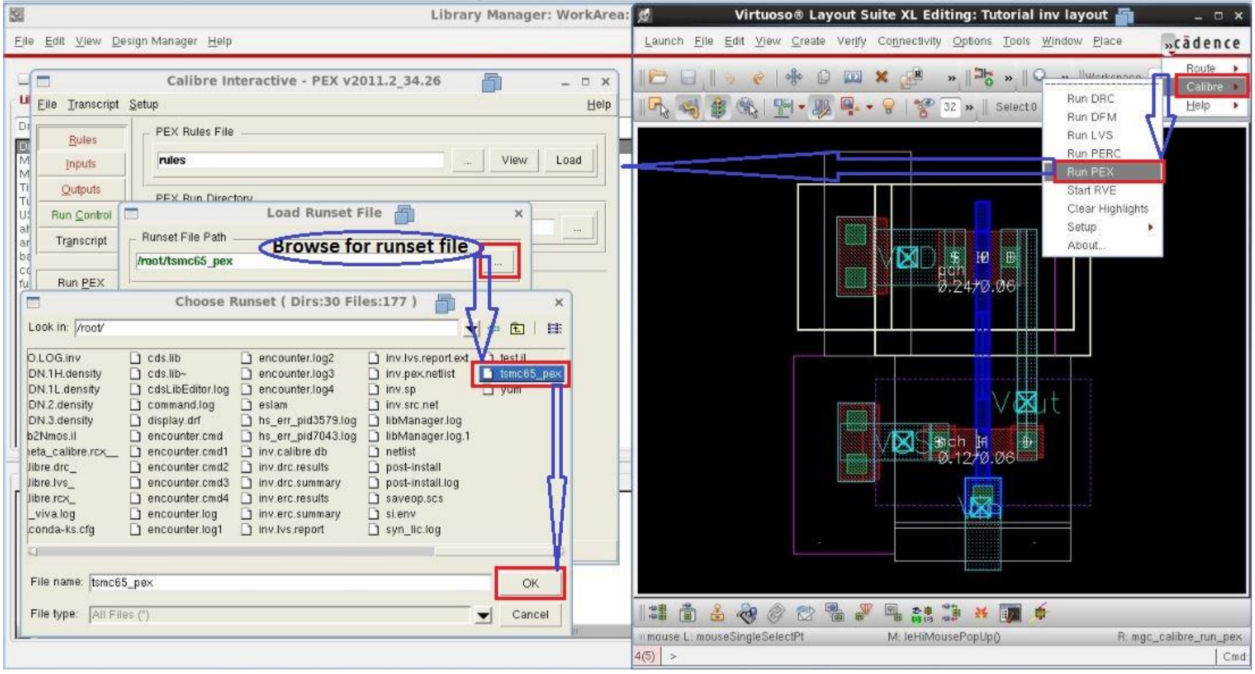


Layout Cell / Type	Source Cell	Nets	Instances	Ports
inv	inv	4L, 4S	1L, 1S	4L, 4S

Cell inv Summary (Clean)

```
           #         ###################      _   _
          #          #                 #      *   *
     #   #           #     CORRECT     #        |
      # #            #                 #      \___/
       #             ###################


LAYOUT CELL NAME:         inv
SOURCE CELL NAME:         inv

--------------------------------------------------------------------------------------------

INITIAL NUMBERS OF OBJECTS
--------------------------

                Layout     Source          Component Type
                ------     ------          --------------
 Ports:              4          4

 Nets:               4          4

 Instances:          1          1          MN  (4 pins)
                     1          1          MP  (4 pins)
                ------     ------
 Total Inst:         2          2
```

Third, PEX is done as shown figure:


Library Manager: WorkArea:
File Edit View Design Manager Help
Calibre Interactive - PEX v2011.2_34.26
File Transcript Setup Help
Rules
Inputs
Outputs
Run Control
Transcript
Run PEX
Load Runset File
Choose Runset (Dirs:30 Files:177)
Virtuoso® Layout Suite XL Editing: Tutorial inv layout
Launch File Edit View Create Verify Connectivity Options Tools Window Place
cādence
Route
Calibre
Help
Run DRC
Run DFM
Run LVS
Run PERC
Run PEX
Start RVE
Clear Highlights
Setup
About...

PEX Rules File: rules

Runset File Path: /root/tsmc65_pex

Browse for runset file

Look in: /root/

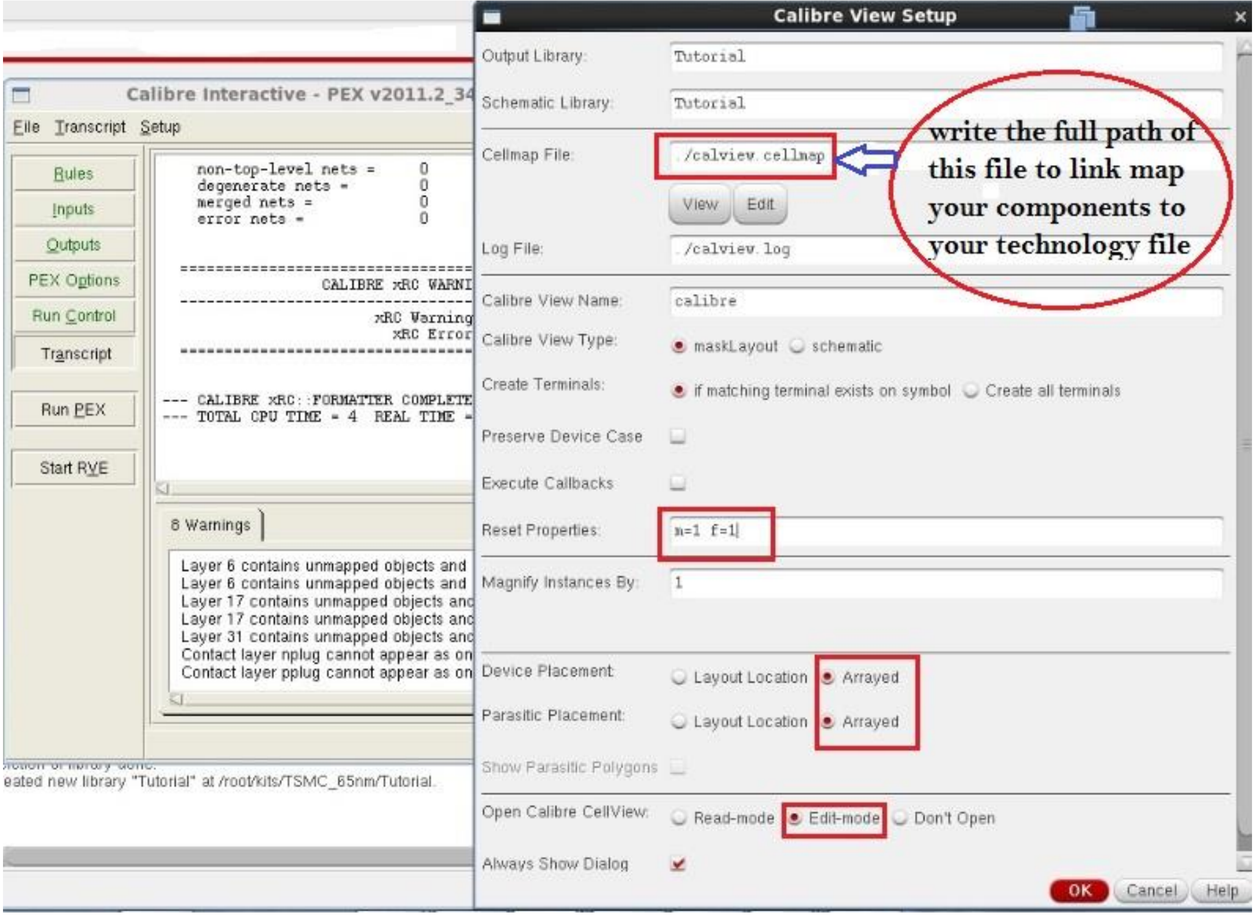
```
O.LOG.inv        cds.lib           encounter.log2      inv.lvs.report.ext   test.il
DN.1H.density    cds.lib~          encounter.log3      inv.pex.netlist      tsmc65_pex
DN.1L.density    cdsLibEditor.log  encounter.log4      inv.sp               yum
DN.2.density     command.log       eslam               inv.src.net
DN.3.density     display.drf       hs_err_pid3579.log  libManager.log
b2Nmos.il        encounter.cmd     hs_err_pid7043.log  libManager.log.1
eta_calibre.rcx__ encounter.cmd1   inv.calibre.db      netlist
libre.drc_       encounter.cmd2    inv.drc.results     post-install
libre.lvs_       encounter.cmd3    inv.drc.summary     post-install.log
libre.rcx_       encounter.cmd4    inv.erc.results     saveop.scs
_viva.log        encounter.log     inv.erc.summary     si.env
conda-ks.cfg     encounter.log1    inv.lvs.report      syn_lic.log
```

File name: tsmc65_pex [OK]

File type: All Files (*) [Cancel]

mouse L: mouseSingleSelectPt   M: leHiMousePopUp()   R: mgc_calibre_run_pex

After, you run PEX → you get this window:



```
non-top-level nets =      0
degenerate nets =         0
merged nets =             0
error nets =              0

================================
              CALIBRE xRC WARNI
--------------------------------
                    xRC Warning
                      xRC Error
--------------------------------

--- CALIBRE xRC::FORMATTER COMPLETE
--- TOTAL CPU TIME = 4   REAL TIME =
```

```
Layer 6 contains unmapped objects and
Layer 6 contains unmapped objects and
Layer 17 contains unmapped objects and
Layer 17 contains unmapped objects and
Layer 31 contains unmapped objects and
Contact layer nplug cannot appear as on
Contact layer pplug cannot appear as on
```

eated new library "Tutorial" at /root/kits/TSMC_65nm/Tutorial.

Output Library: Tutorial

Schematic Library: Tutorial

Cellmap File: ./calview.cellmap

write the full path of this file to link map your components to your technology file

Log File: ./calview.log

Calibre View Name: calibre

Calibre View Type: ● maskLayout ○ schematic

Create Terminals: ● if matching terminal exists on symbol ○ Create all terminals

Preserve Device Case □

Execute Callbacks □

Reset Properties: m=1 f=1

Magnify Instances By: 1

Device Placement: ○ Layout Location ● Arrayed

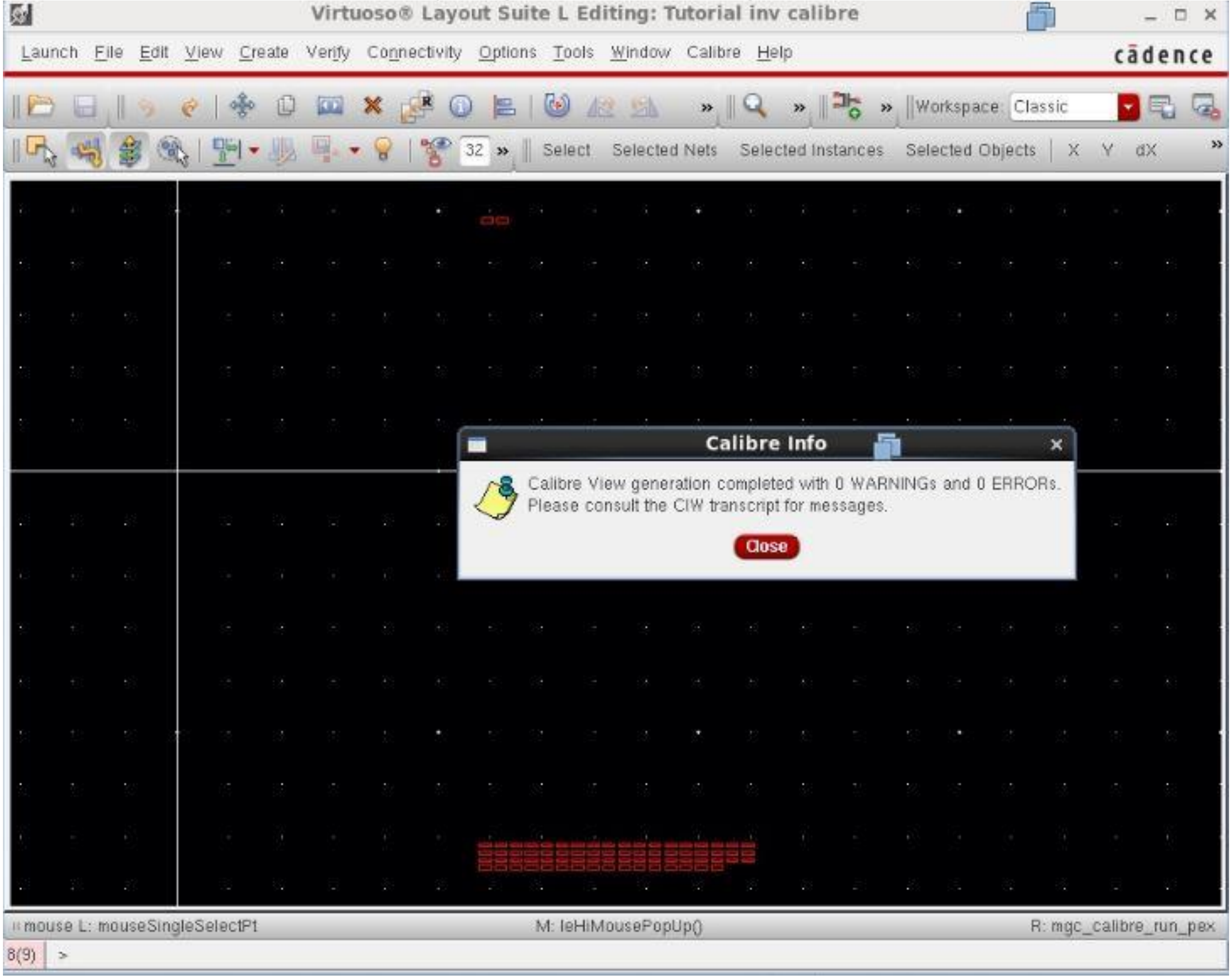
Parasitic Placement: ○ Layout Location ● Arrayed

Show Parasitic Polygons □

Open Calibre CellView: ○ Read-mode ● Edit-mode ○ Don't Open

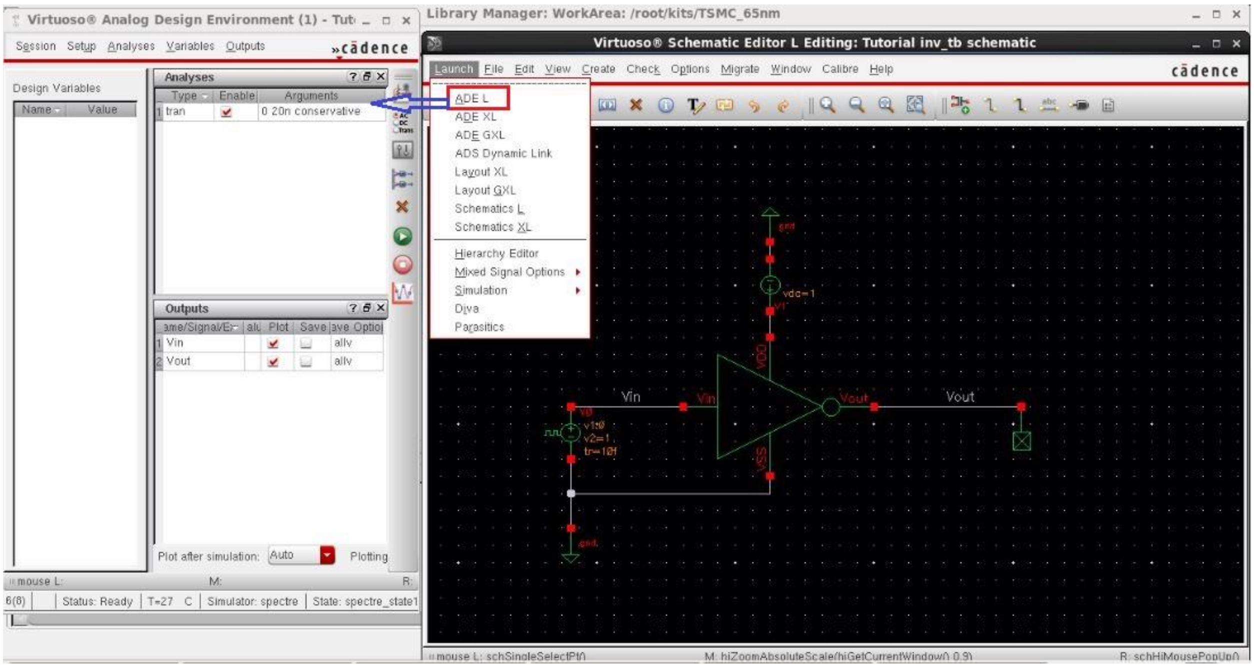
Always Show Dialog ☑

After PEX finish its work → you get this shown figure:

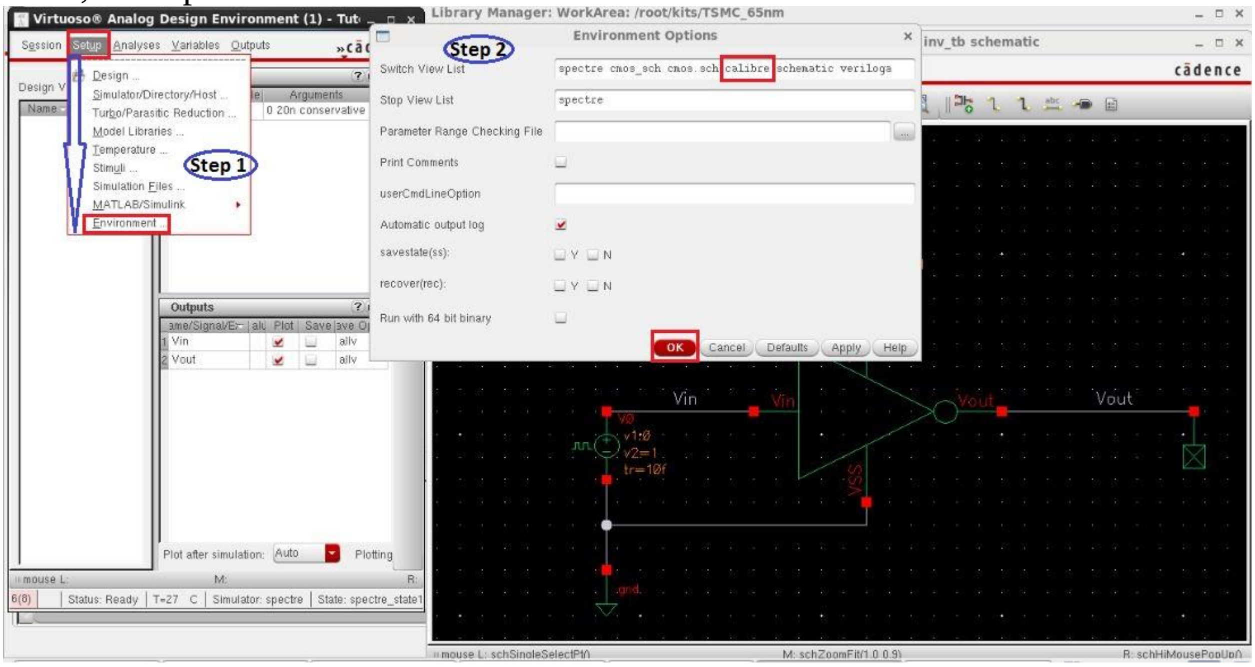

Virtuoso® Layout Suite L Editing: Tutorial inv calibre
Launch File Edit View Create Verify Connectivity Options Tools Window Calibre Help
cādence
Workspace: Classic
Select Selected Nets Selected Instances Selected Objects X Y dX
Calibre Info
Close
mouse L: mouseSingleSelectPt
M: leHiMousePopUp()
R: mgc_calibre_run_pex

Calibre View generation completed with 0 WARNINGs and 0 ERRORs.
Please consult the CIW transcript for messages.

15. Now, We need to simulate parasitic components extracted from drawn layout to know its effect on the circuit functionality:
Now we will test a file called calibre view for old testbench file as shown:


Virtuoso® Analog Design Environment (1) - Tut
Session Setup Analyses Variables Outputs
cādence
Design Variables
Name Value
Analyses
Type Enable Arguments
1 tran 0 20n conservative
Outputs
Name/Signal/Expr Plot Save Save Options
1 Vin allv
2 Vout allv
Plot after simulation: Auto
Plotting
Status: Ready T=27 C Simulator: spectre State: spectre_state1
Library Manager: WorkArea: /root/kits/TSMC_65nm
Virtuoso® Schematic Editor L Editing: Tutorial inv_tb schematic
Launch File Edit View Create Check Options Migrate Window Calibre Help
ADE L
ADE XL
ADE GXL
ADS Dynamic Link
Layout XL
Layout GXL
Schematics L
Schematics XL
Hierarchy Editor
Mixed Signal Options
Simulation
Diva
Parasitics
gnd
vdc=1
Vin
Vout
v1=0
v2=1
tr=10f
mouse L: schSingleSelectPt()
M: hiZoomAbsoluteScale(hiGetCurrentWindow() 0.9)
R: schHiMousePopUp()

Then, edit options as shown:



Switch View List: spectre cmos_sch cmos.sch calibre schematic veriloga

Stop View List: spectre

Parameter Range Checking File:

Print Comments: □

userCmdLineOption:

Automatic output log: ☑

savestate(ss): □ Y □ N

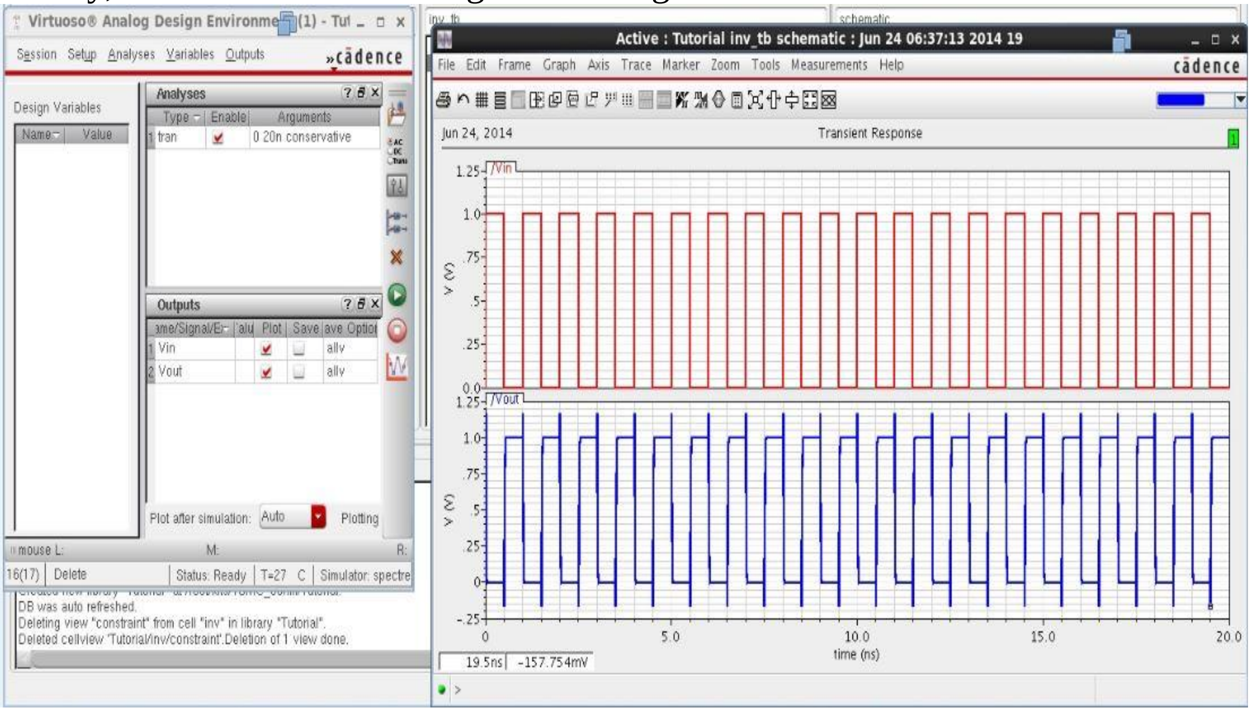
recover(rec): □ Y □ N

Run with 64 bit binary: □

v1:0 v2=1 tr=10f

Status: Ready | T=27 C | Simulator: spectre | State: spectre_state1

Finally, we run simulation and get shown figure:


Virtuoso® Analog Design Environment (1) - Tut
Session Setup Analyses Variables Outputs
cādence
Design Variables
Name Value
Analyses
Type Enable Arguments
Outputs
Name/Signal/Expr Value Plot Save Save Options
Plot after simulation: Auto
Plotting
Active : Tutorial inv_tb schematic : Jun 24 06:37:13 2014 19
File Edit Frame Graph Axis Trace Marker Zoom Tools Measurements Help
Jun 24, 2014
Transient Response
/Vin
/Vout
1.25
1.0
.75
.5
.25
0.0
0
-.25
V (V)
0
5.0
10.0
15.0
20.0
time (ns)

Type	Enable	Arguments
tran	☑	0 20n conservative

	Name/Signal/Expr	Value	Plot	Save	Save Options
1	Vin		☑	□	allv
2	Vout		☑	□	allv

16(17) | Delete | Status: Ready | T=27 C | Simulator: spectre

```
DB was auto refreshed.
Deleting view "constraint" from cell "inv" in library "Tutorial".
Deleted cellview 'Tutorial/inv/constraint'.Deletion of 1 view done.
```

19.5ns | -157.754mV

Appendix B: VTC Linearity test

1- Plot Output Delay Signal Vs Input Voltage

2- Differentiate output Delay Signal by cadence calculator

3-Calculate Average and Non-Linearity Error of differentiated curve:

$$\text{Average} = \frac{Max-Min}{2} \text{ \& Non-Linearity Error} = \frac{Max-Avg.}{Avg.} \text{ x } 100\ \%$$

4-We can use also Matlab to check Non-Linearity Error:

First: By importing Output Delay Signal Vs Input Voltage to Matlab

Then, Run this script:

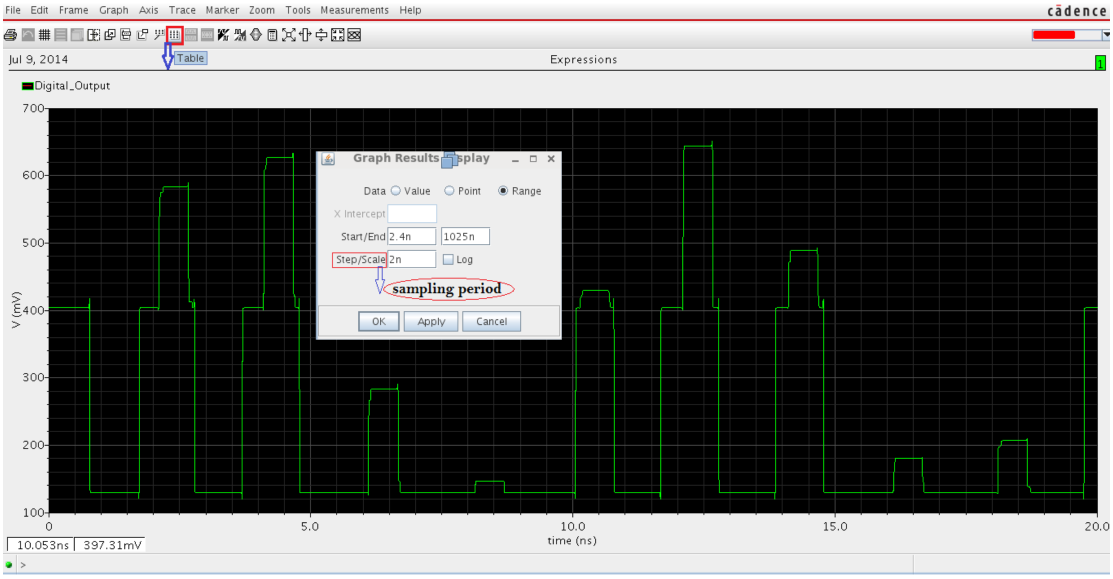
```
% Imported Delay Vs Input Voltage .csv file from cadence
%n: order for fitting
n=1; % to be linear
x=x*10^3; % Input Voltage is converted to mV
y=y*10^12; % Time Delay is converted to psec
p=polyfit(x,y,1);
y2=p(:,1)*x+p(:,2);
slope = (max(y)-min(y))/350;
error=abs(y2-y);
nonlin=( max(error)/slope ) / (350/2^n-bits) ;
% of nonlin
plot(x,y)
hold on
grid
plot(x,y2,'r')
```

Appendix C: ENOB calculations

1. We will simulate our TADC at least for 512 cycle using sinusoidal input source with $F_m = \frac{F_s}{5}$, then save output data stream from cadence after calculation of this equation which model DAC (Decimal Weighting for each bit):

$$\text{Digital Output} = A_0 \text{ x } \frac{DR}{2^{N-bits}{}_{-1}} + 2 \text{ x } A_1 \text{ x } \frac{DR}{2^{N-bits}{}_{-1}} + 4 \text{ x } A_2 \text{ x } \frac{DR}{2^{N-bits}{}_{-1}} + \ldots\ldots$$

→ Importing data from cadence as shown:



Data ○ Value ○ Point ● Range

X Intercept

Start/End 2.4n 1025n

Step/Scale 2n □ Log

sampling period

→ Then, save output table as .csv file to import it to Matlab.

2. We will import data to Matlab, and use Delta-sigma toolbox built-in functions to calculate ENOB using the following code:

```
data_stream = Cadence_Output*63/1.08;
dc_offset = sum(data_stream)./length(data_stream);
data_stream = data_stream-dc_offset;
N = length(data_stream);
%Length of your timedomain data
w=hann(N)/(N/4);     %Hann window
bw=150e6;            %Base-band
Fs=500e6;            %Sampling frequency
f=110e6/Fs;          %Normalized signal frequency
%choose f&N so that f*N be an integer number
fB=N*(bw/Fs);        % Base-band frequency bins
%the BW you are looking at
[snr,ptot,psig,pnoise]=calcSNR(data_stream,f,3,fB,w,N);
Rbit=(snr-1.76)/6.02;    % Equivalent resolution in bits
```