

Arash Azimirad

Timing calibration for up-converting DAC

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 22.01.2015

Thesis supervisor:

Prof. Jussi Rynänen

Thesis advisors:

D.Sc. (Tech.) Kari Stadius

M.Sc. (Tech.) Jerry Lemberg

Author: Arash Azimirad

Title: Timing calibration for up-converting DAC

Date: 22.01.2015

Language: English

Number of pages: 6+50

Department of Micro- and Nanoscience

Professorship: Electrical Circuit Design

Code: S-87

Supervisor: Prof. Jussi Ryynänen

Advisors: D.Sc. (Tech.) Kari Stadius, M.Sc. (Tech.) Jerry Lemberg

This thesis deals with the timing error problem that appears in high frequency Digital to Analog Converters. Inequalities among signal paths in different branches and inaccuracies happened during fabrication, result in different time delays in different branches of a Digital to Analog Converter. The consequence of this inequality is having the data for different bits not arriving to the summation point at the same time. This timing error will create some glitches in the output analog signal

A new approach is introduced in this work that measures the timing error among branches of the DAC and corrects them through a calibration process. Being all the error measurement and its correction process done on chip, this approach can correct the errors created by both sources. This idea was implemented and tested in Eldo simulator. A timing error of 8pS was inserted to the MSB branch of a 10-bit binary coded DAC. After performing the calibration process on this DAC, the SFDR of the output signal was increased by about 3.2dB.

Keywords: timing error, RF-DAC, DAC, timing calibration, up-converting DAC

Preface

This work has been done at Aalto University, department of Micro- and Nanoscience. It was an independent student project.

I would like to thank Professor Jussi Rynänen for his supervision over my thesis. Also, my thanks go to Kari Stadius for his instructions during different phases of the thesis. In addition, I would like to thank Jerry Lemberg for his helpful advices and consultations about different details of the work.

Finally, I would like to thank all people without help of whom it was not possible to get this work done.

Otaniemi, 22.01.2015

Arash Azimirad

Contents

Abstract	ii
Preface	iii
Contents	iv
Symbols and abbreviations	vi
1 Introduction	1
2 Digital to Analog Converters, basic techniques and definitions	4
2.1 Ideal DAC	4
2.1.1 Non Return-to-Zero DAC	4
2.1.2 Return-to-Zero DAC	7
2.1.3 Nyquist-Shannon sampling theory	8
2.2 DAC topologies	8
2.2.1 Resistor DAC	9
2.2.2 Capacitor DAC	9
2.2.3 Current steering DAC	10
2.3 Techniques for improving performance of DACs	10
2.3.1 Differential topology	10
2.3.2 Thermometer coding	12
2.3.3 Noise Shaping	14
2.3.4 Deglitching	15
2.3.5 Delta-Sigma DAC	16
2.3.6 Dynamic Mismatch Mapping	17
2.4 Definition of performance figures	19
2.4.1 Static accuracy	19
2.4.2 Dynamic accuracy	21
3 Solutions	24
3.1 Introduction	24
3.2 Timing error detector unit	25
3.2.1 Theory	25
3.2.2 Simulation results	30
3.3 Delay unit	31
4 Implementation and simulation results	34
4.1 Implementation	34
4.2 Calibration process and simulation results	37
5 Conclusions	43
6 Appendix A- Dynamic flip-flops	44

7	Appendix B- Comparison between the output of RF-DAC transmitter and DCT	47
---	---	----

Symbols and abbreviations

Symbols

δ_K	Dirac sequence
f_{CLK}	Frequency of Clock signal
f_{LO}	Frequency of Local oscillator
T	Period

Abbreviations

ADC	Analog to Digital Converter
CLK	Clock
DAC	Digital to Analog Converter
DCT	Direct Conversion Transmitter
DMM	Dynamic Mismatch Mapping
DNL	Differential Non-Linearity
$E_{fm,i}$	Non-linearity error of the fundamental component at the i input
$E_{fm,code}$	Integral non-linearity error of the fundamental component at an input code
$E_{2fm,i}$	Non-linearity error of the second harmonic component at the i input
$E_{2fm,code}$	Integral non-linearity error of the second harmonic component at an input code
FS	Full Scale
FSR	Full Scale Range
$FUND_{actual}$	The actual fundamental component of the DAC output
$FUND_{ideal}$	The ideal fundamental component of the DAC output
$HC2_{ideal}$	The ideal second harmonic component of the DAC output
$HC2_{actual}$	The actual second harmonic component of the DAC output
IC	Integrated Circuit
INL	Integral Non-Linearity
LO	Local (signal/carrier)
LSB	least Significant Bit
MSB	Most Significant Bit
NRZ	Non Return to Zero
NTF	Noise Transfer Function
OSR	Over Sampling Ratio
PA	Power Amplifier
PSD	Power Spectral Density
RF	Radio Frequency
RF-DAC	Radio Frequency Digital to Analog Converter
RZ	Return to Zero
SFDR	Spurious Free Dynamic Range
SNR	Signal to Noise Ratio
ZOH	Zero Order Hold

1 Introduction

Transmitters were classically designed as depicted in Figure 1 [1]:

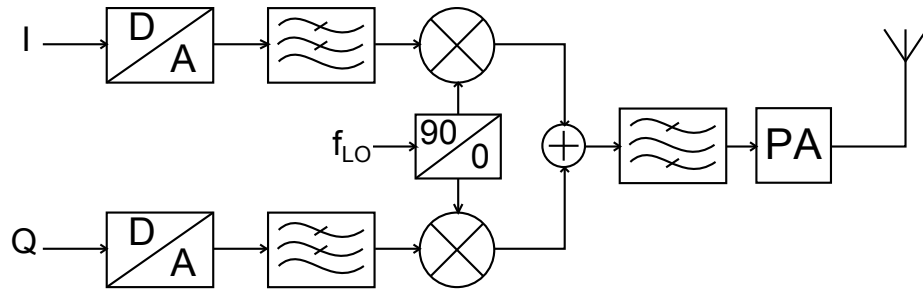


Figure 1: Direct Conversion Transmitter (DCT)

This kind of transmitter is called Direct Conversion Transmitter (DCT). As we can see, the digital data is converted to analog at the baseband and then, after filtering, that analog data is up converted by mixer.

About 10 years ago in [2] a new architecture called RF-DAC was introduced as a replacement for DCT [1]. Since that time various versions of RF-DAC have been designed. The general structure of a RF-DAC can be illustrated as in Figure 2:

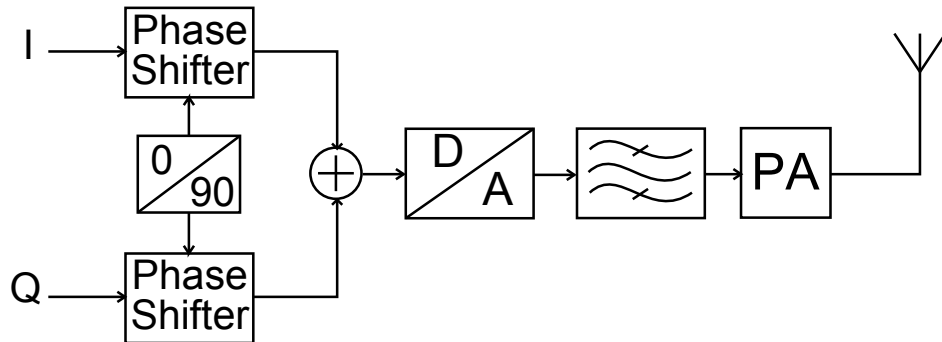


Figure 2: RF-DAC Transmitter

The input digital data is given to the transmitter with the frequency at which we want to send the signal to the air, i.e., the carrier frequency. So, there is no need for up-conversion (for reading about the draw back of RF-DAC transmitter compared to DCT, refer to Appendix B). Then after creating a 90-degree phase difference between the I and Q signals, they are added up together to make a unique digital signal. Then this signal is converted to analog.

We can see that the DACs used in RF-DAC transmitters work at frequencies much higher than the ones used in DCTs. This fact forces us to deal with a problem

called "Timing errors in DACs". Before continuing this argument, it is better to first explain what exactly this problem is. In Figure 3 the general structure of a DAC is illustrated:

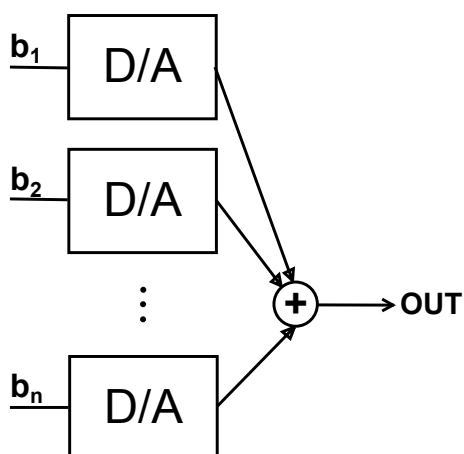


Figure 3: The general structure of a DAC

The output of this DAC in the ideal case will be as follows (assuming that the signal is a ramp, for example):

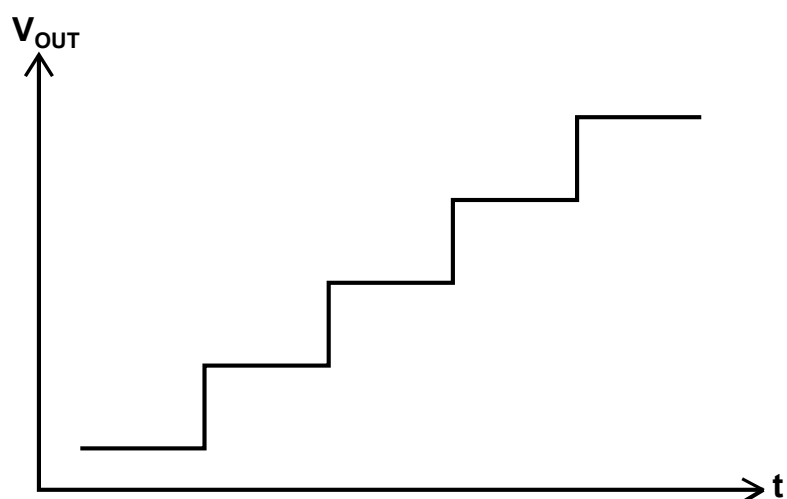


Figure 4: The output of an ideal DAC to a ramp input

However, in practice the signals at the branches of different bits of the DAC do not move forward simultaneously, due to different delays that they face in their paths (the source of these delays are factors like different path lengths, errors in transistor sizing during fabrication process, ...). These delays are modeled in the diagram of Figure 5:

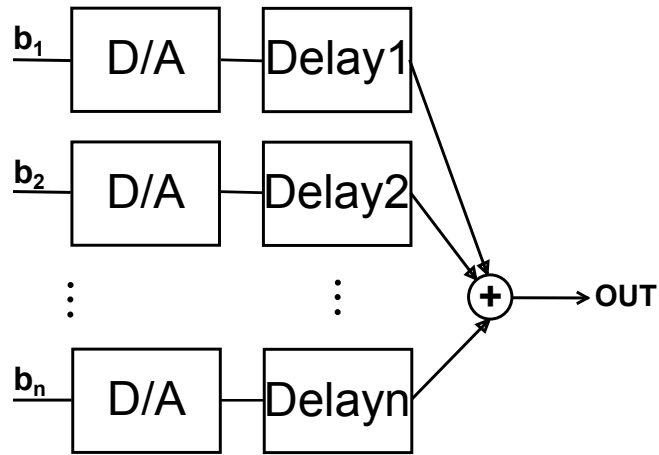


Figure 5: The realistic model of a DAC

There will be some glitches in the output of this DAC at the transition points of the signal, as illustrated in Figure 6:

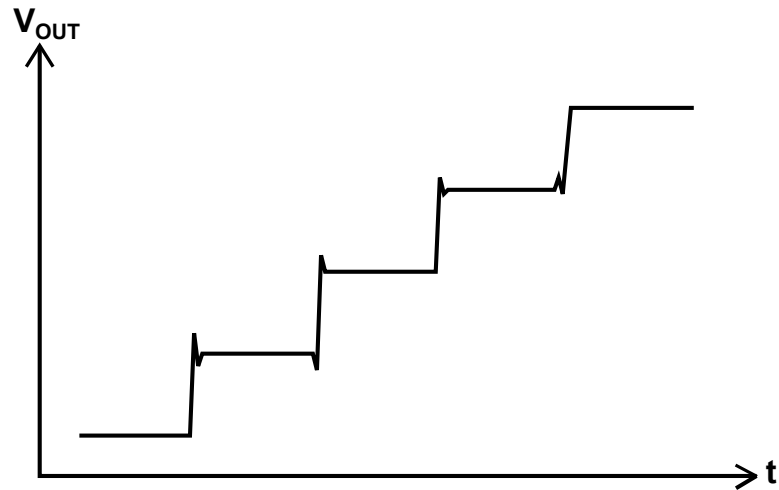


Figure 6: The output of a real DAC

Because the delay times are small, the frequency of these glitches are high. So, for the DACs not operating at high frequencies, these glitches are not a big problem and can be easily filtered out at the output. But, as the operating frequency of the DAC grows, filtering these glitches becomes more difficult.

Now, returning back to our earlier discussion about DCTs and RF-DAC transmitters, the DACs in the RF-DAC transmitter architecture usually deal with timing error problem. And as this structure is becoming more popular, the need for solutions for this problem is becoming more serious.

The object of this thesis is finding solutions for the timing error problem in RF-DACs. In this thesis we do not go through the design of the whole transmitter and just focus on the RF-DAC itself. The results can be useful for high-speed DACs which are used for applications other than transmitters, as well.

2 Digital to Analog Converters, basic techniques and definitions

In this section some basic concepts and definitions about Digital to Analog Converters are explained.

2.1 Ideal DAC

Any digital code can be expressed as follows [3]:

$$d = \sum_{i=0}^{B-1} 2^i \cdot b_i \quad b_i \in [0, 1] \quad (1)$$

By multiplying Equation 1 to a constant value like Δ we will have:

$$u = \Delta \cdot \sum_{i=0}^{B-1} 2^i \cdot b_i \quad (2)$$

This equation is the basis of the operation of Digital to Analog Converters. The values of the terms $\Delta \cdot 2^i$ are created either by voltage or current or electrical charge. Usually a reference source with the value of Δ is used for creating different $\Delta \cdot 2^i$ terms. The operation of the terms b_i is realized by the use of switches. We can see that equation 2 can take on only some specific discrete values. The step size between two consequent values is Δ . It means that if we want to convert an analog signal with its amplitude between $u_k = \Delta \cdot 2^k$ and $u_{k+1} = \Delta \cdot 2^{k+1}$ to digital, we should choose one of the values u_k and u_{k+1} . This unavoidable error created due to the digital to analog conversion is called "Quantization error". The value of quantization error is between $-\frac{1}{2}\Delta$ and $+\frac{1}{2}\Delta$. [3]

Another term defined for DACs is Full Scale Range (FSR). It shows the maximum amplitude that the DAC can sweep in its output. Considering equation 2, we can derive the relation for FSR as follows [3]:

$$FSR = u|_{b_0, b_1, \dots, b_{B-1}=1} - u|_{b_0, b_1, \dots, b_{B-1}=0} = \Delta \cdot \sum_{i=0}^{B-1} 2^i \cdot 1 - \Delta \cdot \sum_{i=0}^{B-1} 2^i \cdot 0 = \Delta \cdot (2^B - 1) \quad (3)$$

2.1.1 Non Return-to-Zero DAC

We define the Non-return-to-Zero DAC as a DAC with an impulse response with the shape of a $\text{rect}()$ function (Figure 7.a), as follows [3]:

$$\rho_{NRZ}(t) = \text{rect}\left(\frac{t - \frac{T}{2}}{T}\right) \quad (4)$$

The input data is a series of numbers, like d_k in Figure 7.b [3]. The output of the DAC in time domain can be derived by multiplying each instance of d_k to the $\text{rect}()$

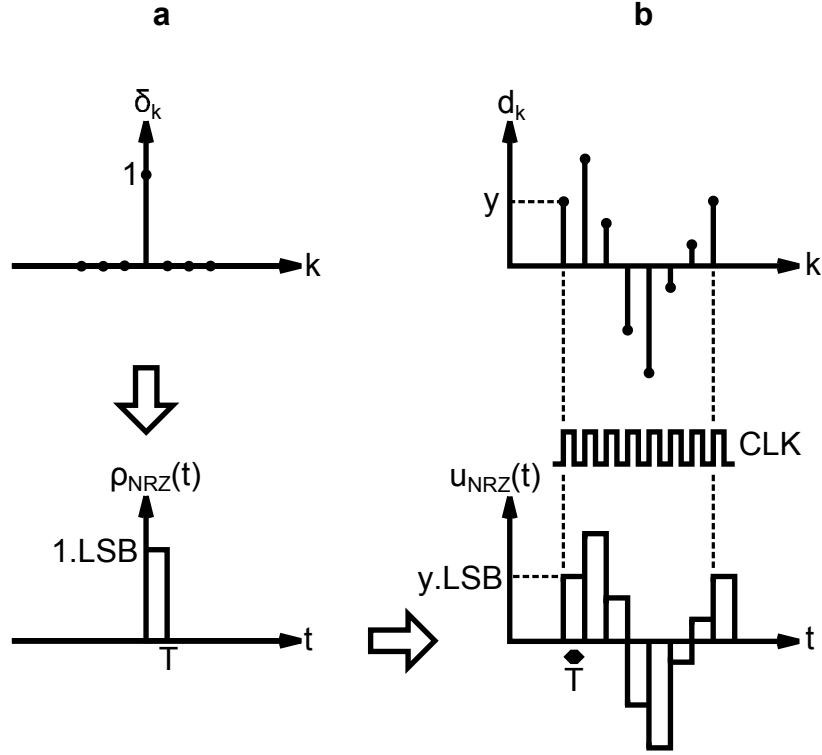


Figure 7: Ideal NRZ DAC input and output: (a) impulse response (b) typical digital input and the output of the DAC for it

function of 4 [3]:

$$u_{NRZ}(t) = \sum_k d_k \cdot u_{NRZ}(t - k.T) = \sum_k d_k \cdot \text{rect}\left(\frac{t - k.T - \frac{T}{2}}{T}\right) \quad (5)$$

For investigating the output of the NRZ-DAC in frequency domain, we should calculate the Fourier transform of 5. Equation 5 can be written as follows [3]:

$$u_{NRZ}(t) = [d(t) \cdot \sum_k \delta(t - k.T)] * \rho_{NRZ}(t) \quad (6)$$

Now for calculating the fourier transform of 5 we can multiply the fourier transform of each of the expressions on the sides of the convolution sign in 6. The fourier transform of $\rho_{NRZ}(t)$ is:

$$\rho_{NRZ}(t) = \text{rect}\left(\frac{t - \frac{T}{2}}{T}\right) \Rightarrow \rho_{NRZ}(j\omega) = T \cdot \frac{\sin \frac{\omega T}{2}}{\frac{\omega T}{2}} \cdot e^{-j\frac{\omega T}{2}} \quad (7)$$

So, the fourier transform of 6 will be [3]:

$$U_{NRZ}(j\omega) = \frac{1}{T} \sum_m D(j(\omega - m \frac{2\pi}{T})) \cdot \rho_{NRZ}(j\omega) \quad (8)$$

$|U_{NRZ}(j\omega)|$ and also $|D(j\omega)|$ and $|\rho_{NRZ}(j\omega)|$ are drawn in Figure 8 [3]. The frequency response of the NRZ-DAC is a sinc() function with zeros at even multiples of $\frac{2\pi}{T}$.

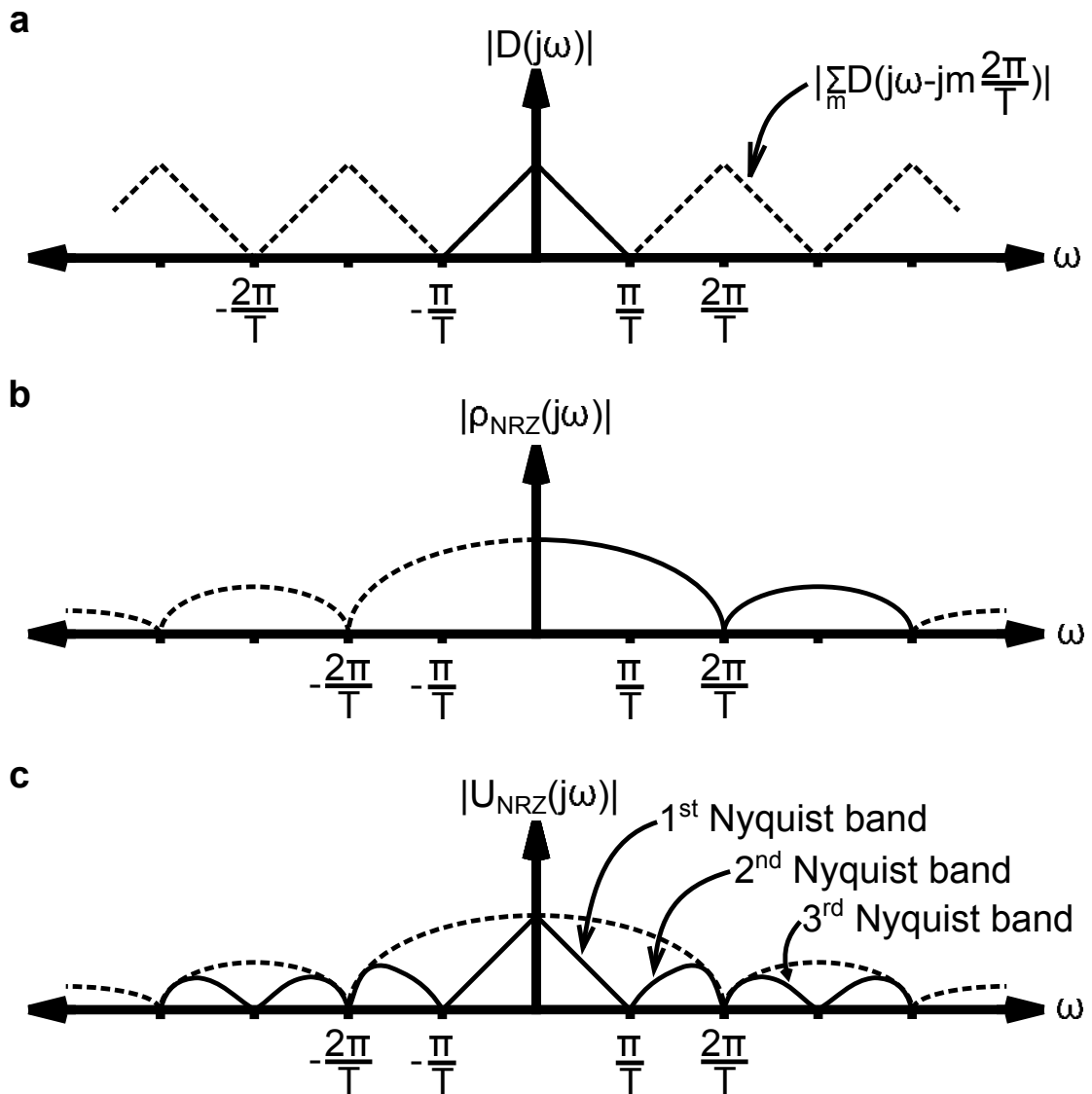


Figure 8: Spectrum of ideal NRZ DAC: (a) (repetitive) input spectrum, (b) NRZ DAC unit pulse spectrum, (c) NRZ DAC output spectrum

2.1.2 Return-to-Zero DAC

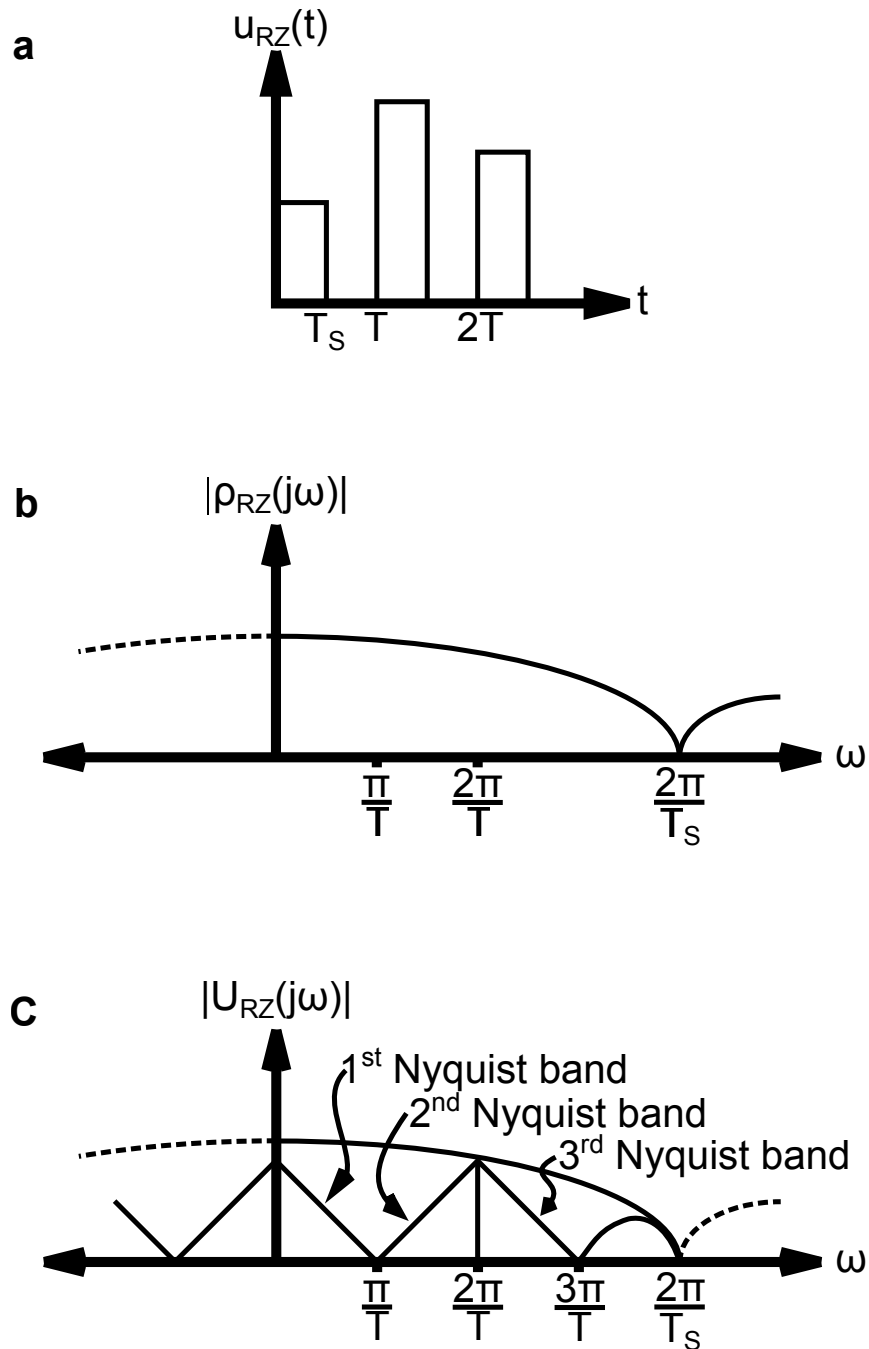


Figure 9: Spectrum of ideal RZ DAC: (a) ideal time domain output waveform (b) RZ DAC unit pulse spectrum (c) RZ DAC output spectrum

The time response of the Return-to-Zero (RZ) DAC is illustrated in Figure 9a [3]. We can see that the width of the pulse is equal to a portion of the sampling period. We call the fraction of $\frac{T_s}{T}$ "Duty factor" and show it with D . For getting

the frequency response of RZ-DAC we should change T in Equation 7 to T_S [3]:

$$\rho_{RZ}(t) = \text{rect}\left(\frac{t - \frac{T_S}{2}}{T_S}\right) \Rightarrow \rho_{RZ}(j\omega) = T_S \cdot \frac{\sin \frac{\omega T_S}{2}}{\frac{\omega T_S}{2}} \cdot e^{-j\omega \frac{T_S}{2}} \quad (9)$$

Continuing in a manner similar to what we did for NRZ-DAC, we can get the frequency response of RZ-DAC to a series of digital inputs, as follows:

$$U_{RZ}(j\omega) = \frac{1}{T} \sum_m D(j(\omega - m \frac{2\pi}{T})) \cdot \rho_{RZ}(j\omega) \quad (10)$$

The transfer function of the RZ-DAC and its output in frequency domain are illustrated in figures 9b and 9c. The transfer function of RZ-DAC, similar to NRZ-DAC, is a sinc(); With this difference that its zeros are at even multiples of $\frac{2\pi}{T_S}$, not $\frac{2\pi}{T}$. It means that its first zero is at bigger frequency, compared to NRZ-DAC. This will lead to less amplitude distortion [3]. However, the main reason for preferring RZ-DAC to NRZ-DAC in some applications, is due to its better dynamic performance. This feature is based on its hardware implementation [3]. But, this advantage comes with the drawback of having the the power of the base band signal weakened by the factor of $(\frac{T_S}{T})^2$ (compared to NRZ-DAC) [3].

2.1.3 Nyquist-Shannon sampling theory

Nyquist-Shannon sampling theory is one of the fundamental theories about Analog to Digital and Digital to Analog conversion. It can be explained as follows:

The minimum sampling frequency that is needed for sampling an analog signal with highest frequency component of f_H , in order to be able to reconstruct it from the samples later, without any data loss, is $2f_H$. [4]

The frequency $2f_H$ is called the Nyquist frequency.

"Aliasing" and "Over sampling" are two terms about sampling which are related to Nyquist frequency. They are defined as follows:

2.1.3.1 Aliasing If we have component with a frequency over Nyquist frequency at the input of an ADC, they will be folded back or replicated at frequencies above or below the Nyquist frequency. This phenomena is called "Aliasing". For preventing this issue, proper filtering should be done before giving the signal to the ADC [5].

2.1.3.2 Over sampling If we sample analog signal at frequencies much higher than Nyquist frequency, we have "oversampled" it. Oversampling decreases quantization noise and improves signal quality [5].

2.2 DAC topologies

There are three main topologies used for implementing DACs. They are briefly explained in the following parts.

2.2.1 Resistor DAC

This is the most simple implementation of DACs. In Figure 10 [6] a form of this topology called R-2R ladder is illustrated:

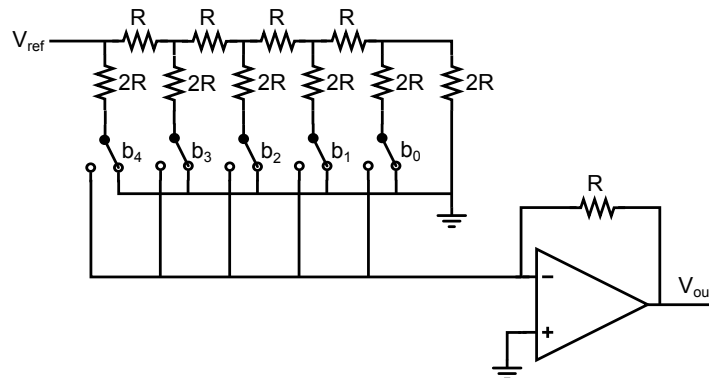


Figure 10: 5-bit R-2R ladder DAC

By changing the position of the bit switches, we can make all the analog voltages corresponding to different binary input codes.

For getting nice accuracy the ratio of resistors should be implemented accurately. Non linearities of the resistors and the buffer and also the buffer bandwidth and the large RC time constant of this topology degrade its linearity and speed. So, this structure is not convenient for high speed conversion applications. [6]

2.2.2 Capacitor DAC

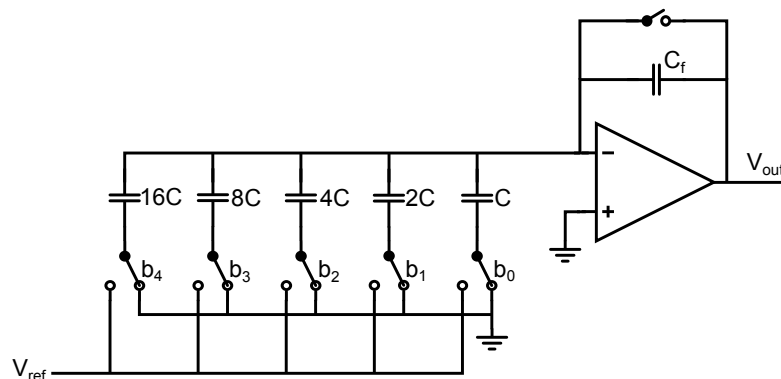


Figure 11: 5-bit switched capacitor DAC

You can see the schematic of this topology in Figure 11 [6]. The conversion is performed in two phases; In the first phase, the binary capacitors are connected either to the reference voltage or to ground, based on the value of their corresponding bits. In this phase the feedback capacitor is shorted. In the second phase, all the input switches connect one end of all binary capacitors to ground, and the switch around the feedback capacitor opens. In this way, the charge that all binary capacitors had gathered during the first phase, will be transferred to the output.

The advantage of this topology is its low power consumption. Similar to resistor DAC, the accuracy of the DAC depends on the accuracy of the values of the capacitors. Although it is faster than resistor DACs, but still speed and linearity are the main problems in this structure. [6]

2.2.3 Current steering DAC

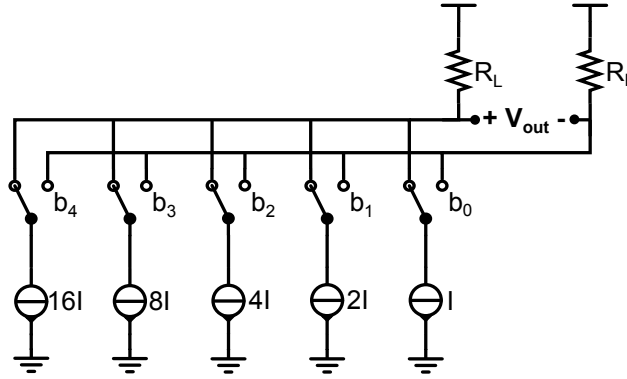


Figure 12: 5-bit current steering DAC

The general structure of a five bit binary coded differential current steering DAC is illustrated in Figure 12 [6]. The currents corresponding to different input bits will be added up together at the output node and change to voltage by flowing to the load resistance.

The acceptable performance of current steering DACs at high frequencies has made them the only practical choice in such applications [7]. Another advantage of this topology is having high output impedance and also having the output signal in the form of current. Due to this feature they can drive a 50Ω standard load without the need for buffer. [6]

2.3 Techniques for improving performance of DACs

2.3.1 Differential topology

Differential topology is usually preferred to single ended topology due to its better noise performance. In the differential topology the common mode noise will be eliminated. A single ended 10 bit DAC and a differential 10 bit DAC are illustrated in figures 13 and 14, respectively:

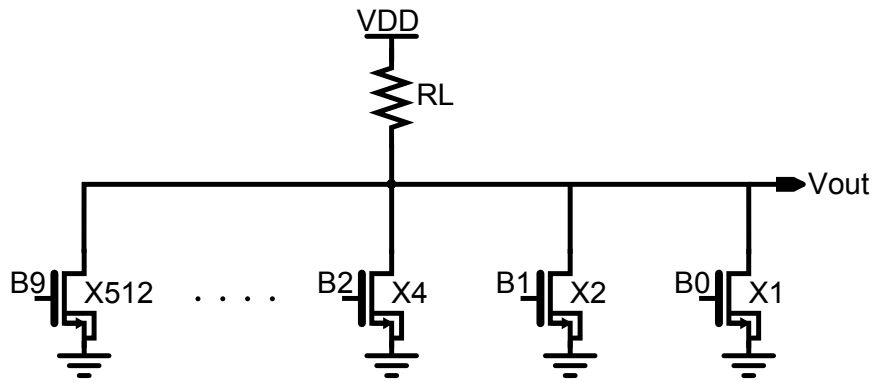
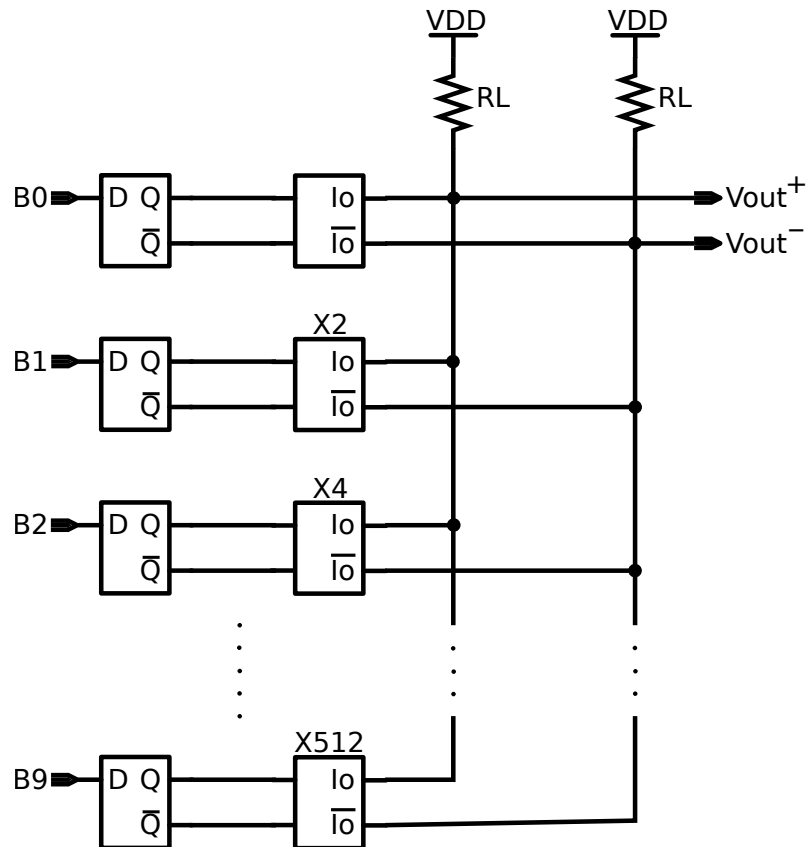


Figure 13: Single ended 10 bit DAC



Circuit diagram of the above blocks:

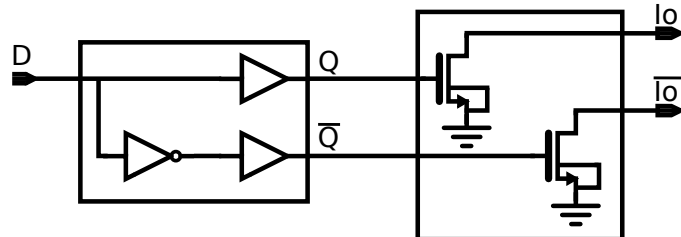


Figure 14: Differential 10 bit DAC

We can see that in the differential case, both nodes of the output are branched from signal line. It means that the noise existing on the input signal will be on both nodes of the output which will result in them canceling out each other. While it is not the case in the single ended mode. [1]

2.3.2 Thermometer coding

Thermometer coding or unary coding is a coding system for showing zero and natural numbers with 1s and 0s. This coding system for numbers 0 to 5 is described in Table 1:

Number	Thermometer code
0	0
1	10
2	110
3	1110
4	11110
5	111110

Table 1: Thermometer coding

This coding system changes the "rising and falling edge" glitches at the output of a binary coded DAC, produced due to the difference between "switching off" and "switching on" times of a transistor, to "just rising edge" or "just falling edge" glitches (depending on whether the signal value is increasing or decreasing).

For clarifying this issue, we consider the output of a binary coded DAC and also a thermometer coded DAC when their input data changes from the value of 2 to 5. In this example we imagine the "switching on" delay time of a transistor to be 2 time units and its "switching off" time to be 1 time unit.

In the binary coded DAC this transition will be from the code "010" to the code "101". The implementation of this transition in the related circuit is illustrated in Figure 15:

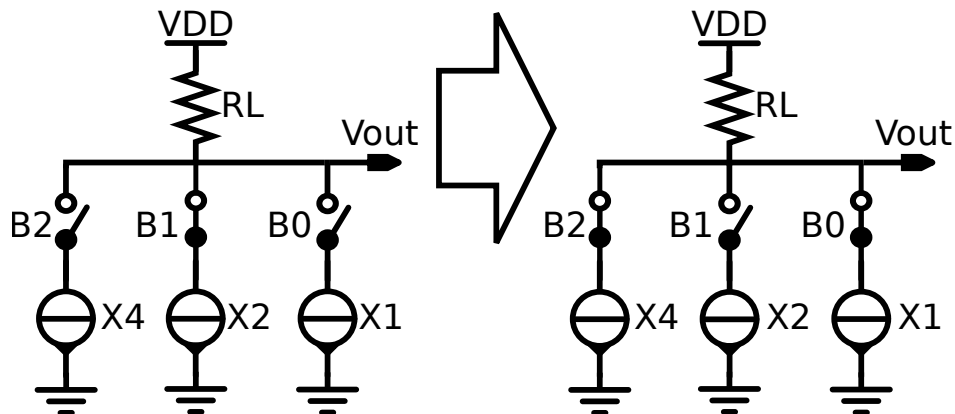


Figure 15: Transition from 2 to 5 in a binary coded DAC

Considering the delay times we defined above, the output wave of the DAC will be as follows:

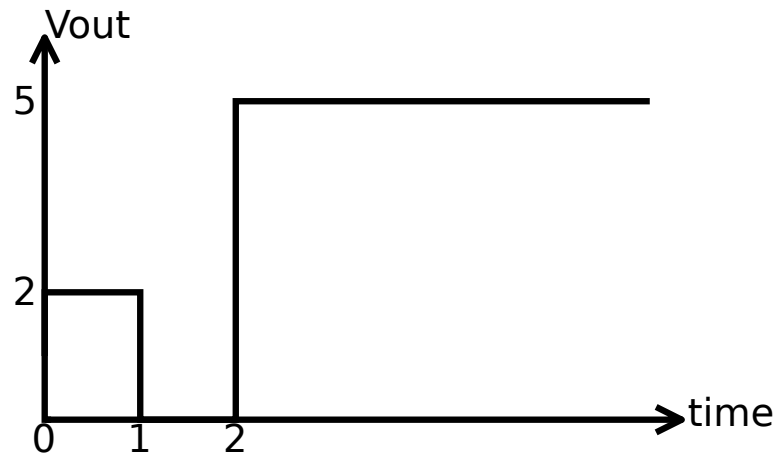


Figure 16: Output of the circuit in Figure 15

Now we consider this transition in a thermometer coded DAC. Thermometer code for 2 is "110" and for 5, it is "111110". Implementation of this transition in the circuit is illustrated in Figure 17:

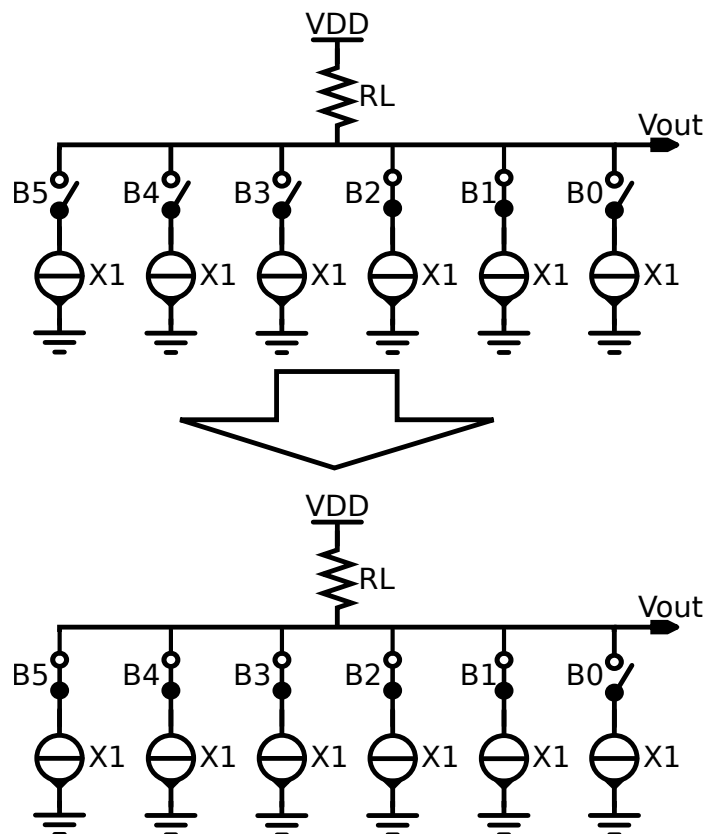


Figure 17: Transition from 2 to 5 in a thermometer coded DAC

By considering the delay times we defined above for transistor switching times, the output wave of this DAC will be as follows:

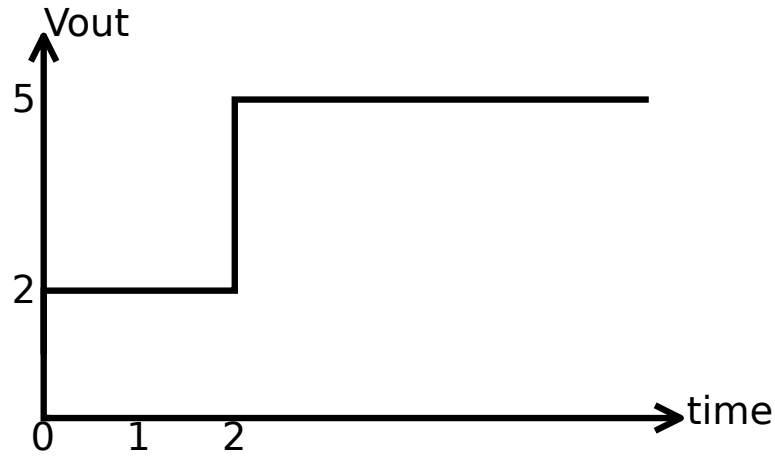


Figure 18: Output of the circuit in Figure 17

By comparing the waveforms of figures 16 and 18 we can see that the rising and falling edge glitches have changed into "just rising" glitches thanks to the thermometer coding. However, this advantage of thermometer coding comes with the drawback of more IC area consumption, and also more complex control signals.

2.3.3 Noise Shaping

The spectrum of the quantization noise of a DAC is normally as depicted in Figure 19 [3]:

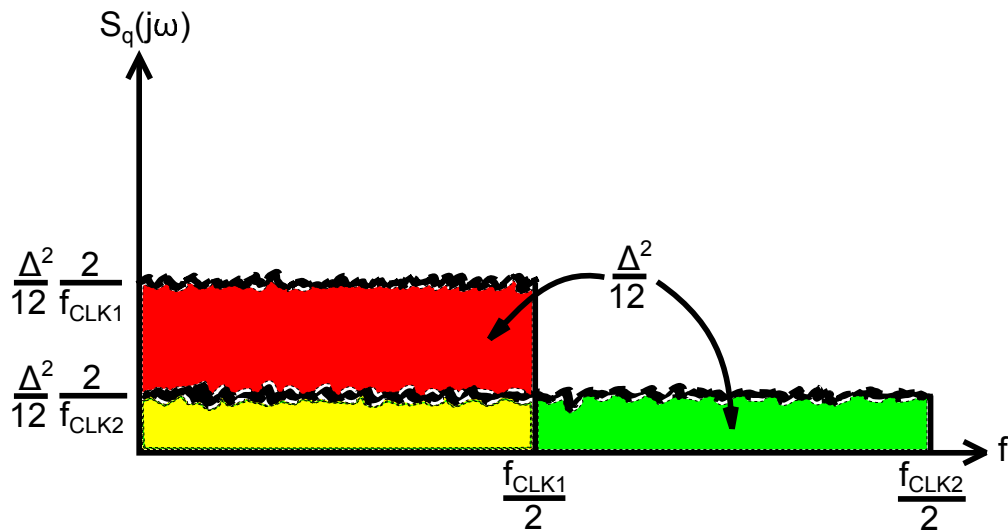


Figure 19: Quantization noise of Nyquist rate DAC

By means of some techniques, called "Noise Shaping", we can change the quantization noise spectrum of Figure 19 to something like Figure 20 [3]:

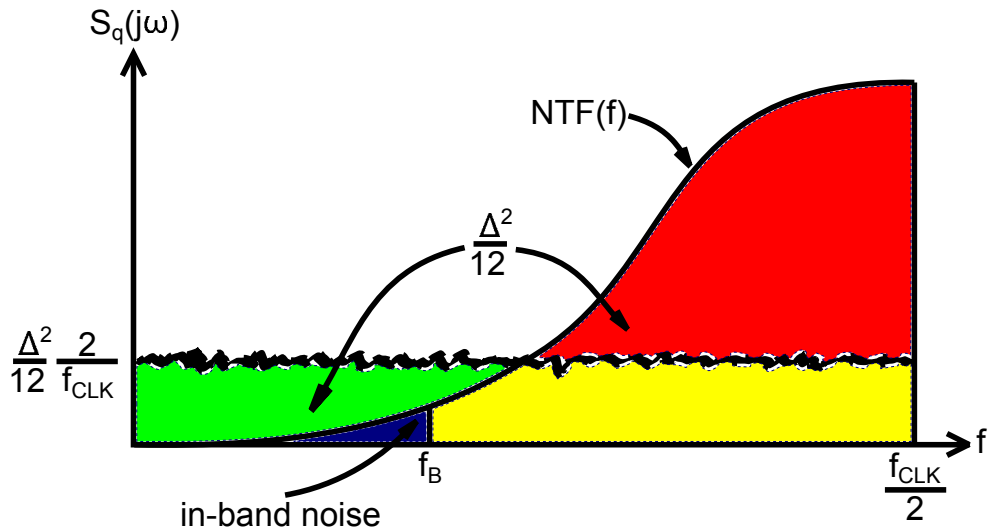


Figure 20: Noise shaped quantization noise

As we can see in Figure 20, the quantization noise at desired frequency has decreased significantly. And it has become much larger at out-of-band frequencies.

Noise shaping idea was first introduced by Cutler [8] in 1954 [9].

2.3.4 Deglitching

As explained in Introduction chapter, the timing error in DACs creates some glitches at the output of the DAC, as it was illustrated in Figure 6. One solution for overcoming this problem is "Deglitching". In this approach a unit called "track-and-hold" (T&H) is inserted between the output of the DAC and the output node. As illustrated in Figure 21 [10], there is a switch in T&H unit which can connect/disconnect the output of the DAC to/from the output node, by a control voltage. Also, a storage unit (like a capacitor) is connected at the output node in order to reserve (hold) the voltage of the output node, after the switch gets open.

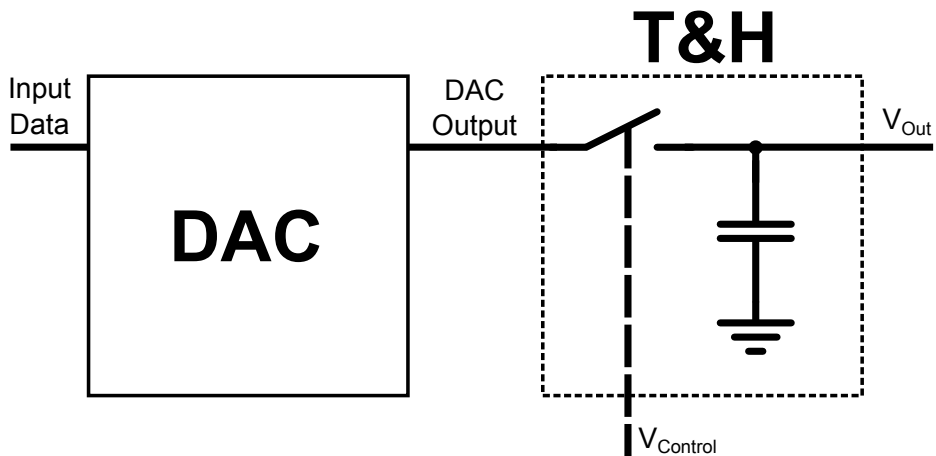


Figure 21: DAC and T&H unit

In Figure 22 [10], we can see the wave forms of the voltages of different nodes of Figure 21. The control voltage consists of two phases; Track phase and Hold phase. During Track phase the switch connects the output of the DAC to the output node. This phase should start after the glitches at the output of the DAC are settled down and it should finish before the glitches of the next cycle have started. During the Hold phase the switch gets open. This phase starts before the glitches happen at the output of the DAC, and it finishes after those glitches are settled down. During this phase the output node reserves its previous voltage until the next Track phase begins. As illustrated in Figure 22, through this approach we can prevent the glitches at the output of the DAC from appearing at output node. However, implementing the phases of the control voltage requires special care. In addition, distortion and noise of T&H can affect the performance of the DAC [10].

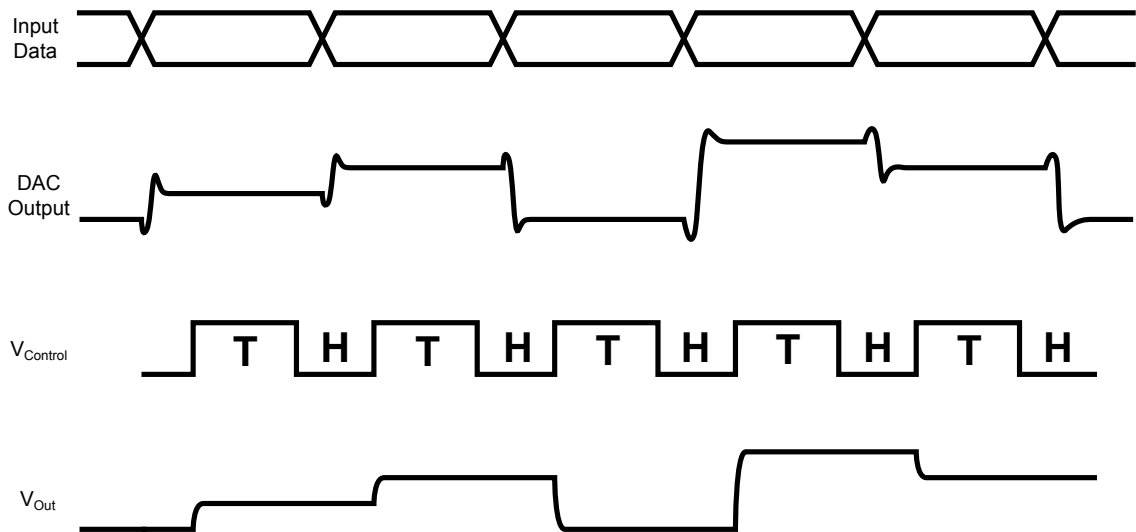


Figure 22: Waveforms of the circuit of Figure 21

2.3.5 Delta-Sigma DAC

Delta-Sigma is a modulation technique which is used in many electronic circuits, including DACs. A Delta-Sigma modulator converts an analog signal to a stream of bits. At each sample moment, instead of converting the analog signal amplitude to a digital code, it calculates the difference (Delta) of the current sample with the previous one. For making the original analog signal from these bits, we should add (Sigma) them together [11].

There are several motivations for using Delta-Sigma modulators in DACs; Delta-Sigma DACs have lower quantization noise in desired frequencies. The use of Delta-Sigma modulators in DACs results in cheaper circuitry with less power consumption [12]. If we want to achieve the resolution that we get with a Delta-Sigma DAC, with a conventional DAC, we should use much more accurate analog components [13].

Figure 23, taken from [14], shows the diagram of a typical Delta-Sigma DAC.

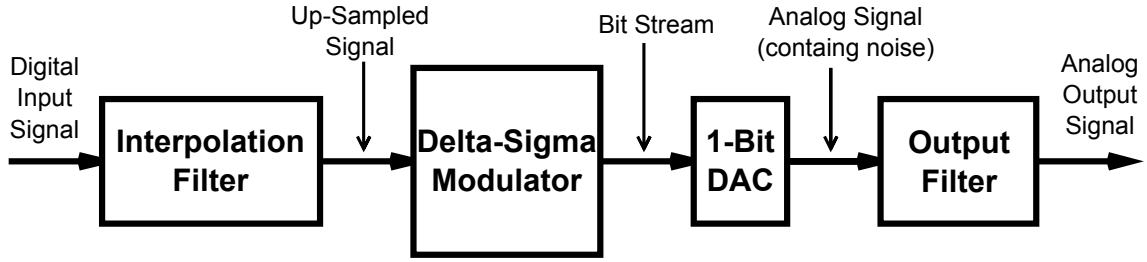


Figure 23: Diagram of a typical Delta-Sigma DAC

The output of Delta-Sigma modulator is one stream of 1s and 0s. But, the input digital signal to this Delta-Sigma DAC, consists of several of such streams (depending on the bit-count of the DAC). So, it means that in order to not lose any data we need to have the sample rate of Delta-Sigma modulator output be much more than the sample rate of input digital code. This increase in sample rate is done by help of an interpolation filter. Interpolation filter first inserts a fixed amount of zeros (depending on amount of desired increase in sample rate) between each two consequent samples. Then by passing the resulted signal through a low-pass filter, the zeros will become interpolated. Sigma-Delta modulator produces a stream of bits. For each sample at the digital input, there is a fixed number of bits at the output of Delta-Sigma modulator. We have a 1-bit DAC at the output of Delta-Sigma modulator that converts the bit stream to an analog signal. The produced analog signal contains the data of the input signal in its baseband and most of the quantization noise and the image of the input signal at higher bands. So, for extracting data a low pass filter is needed at the output, as illustrated in Figure 23 [14].

2.3.6 Dynamic Mismatch Mapping

Dynamic Mismatch mapping (DMM) is a new approach introduced in [6]. Two new parameters called Dynamic-DNL and Dynamic-INL are defined as follows:

$$dynamic - INL_{code} = \sqrt{\frac{|E_{fm,code}|^2 + |E_{2fm,code}|^2}{|FUND_{ideal,1LSB}|^2}} \cdot LSB, \quad code = 0 - full\ scale \quad (11)$$

$$dynamic - DNL_{code} = \sqrt{\frac{(|FUND_{code}|^2 + |HC2_{code}|^2) - (|FUND_{code-1}|^2 + |HC2_{code-1}|^2)}{|FUND_{ideal,1LSB}|^2}} - 1, \quad code = 0 - full\ scale \quad (12)$$

while

$$E_{fm,code} = FUND_{actual,code} - FUND_{ideal,code} = \sum_{i=1}^{code} E_{fm,i}, \quad code = 0 - full\ scale\ digital\ code \quad (13)$$

$$E_{2fm,code} = HC2_{actual,code} - HC2_{ideal,code} = \sum_{i=1}^{code} E_{2fm,i}, \quad code = 0 - full\ scale\ digital\ code \quad (14)$$

$FUND_{ideal}$: The ideal fundamental component of the DAC output

$FUND_{actual}$: The actual fundamental component of the DAC output

$HC2_{ideal}$: The ideal second harmonic component of the DAC output

$HC2_{actual}$: The actual second harmonic component of the DAC output

$E_{fm,i}$: Non-linearity error of the fundamental component at the i input

$E_{2fm,i}$: Non-linearity error of the second harmonic component at the i input

In this work ([6]) it is stated that parameters dynamic-INL and dynamic-DNL show both amplitude and timing error of a DAC. So, by minimizing them for the whole range of input codes we can minimize both amplitude and timing error of the DAC. The good point about this approach is that for calculating dynamic-INL and dynamic-DNL just we need to measure the output of the DAC at fundamental and second harmonic components. And designing a circuit for this purpose is much easier than designing a circuit for measuring small timing errors. In other words, through this approach, we can decrease the timing error of the DAC without a need for an accurate circuit that should measure small timing errors.

The solution that is offered in this work for decreasing dynamic-INL and dynamic-DNL of the DAC, is to arrange the order of the branches of the DAC in such a way that their errors cancel out each other as much as possible, as illustrated in Figure 24 [6]:

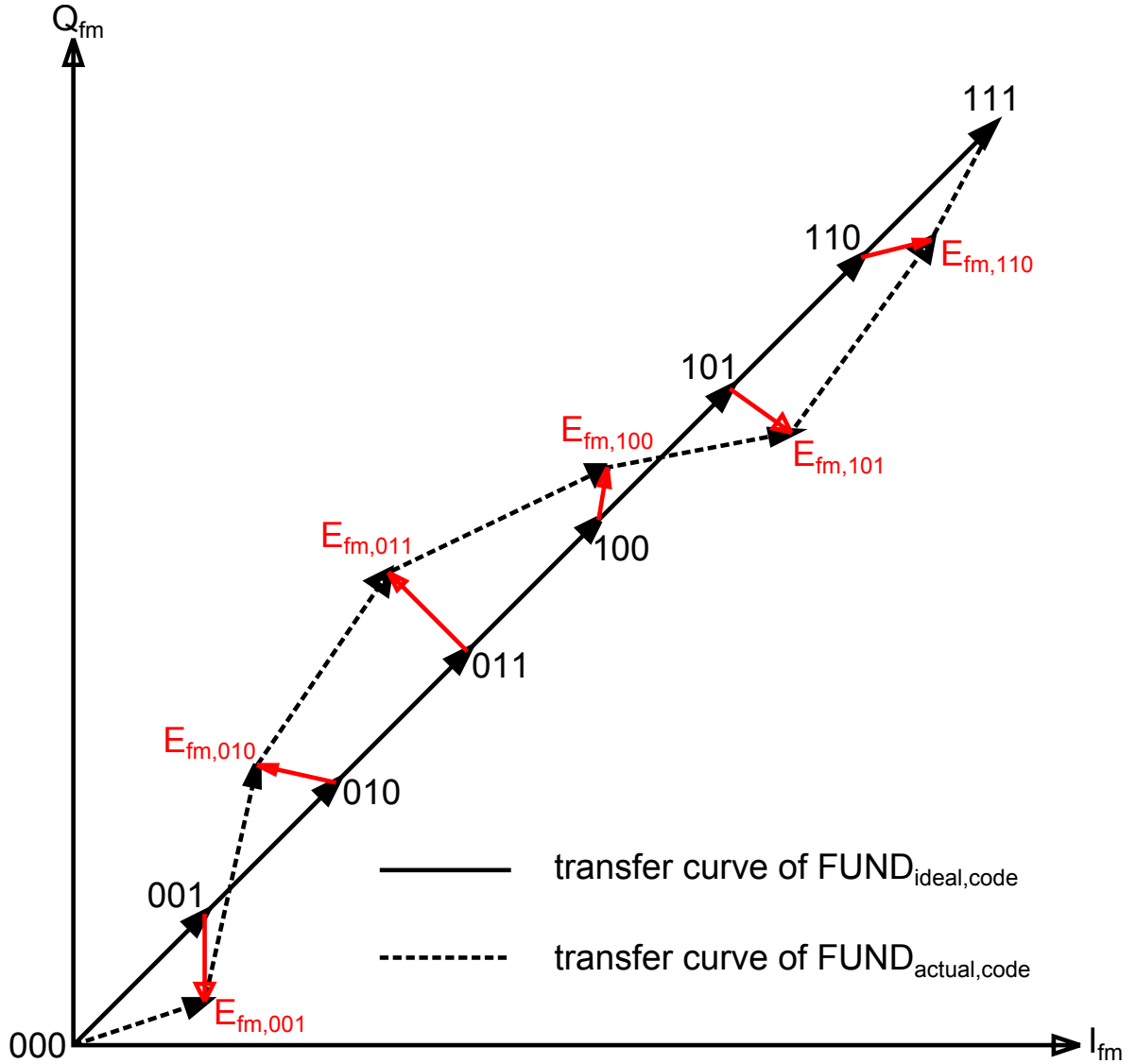


Figure 24: Two dimensional dynamic transfer curve of the fundamental component of the modulated DAC output

2.4 Definition of performance figures

2.4.1 Static accuracy

Static accuracy parameter describe the DC characteristic of the DAC [3]. The following figures are defined for measuring the static accuracy of a DAC.

2.4.1.1 Gain and offset error In Figure 25 [3] the ideal and measured characteristic of a typical DAC is illustrated:

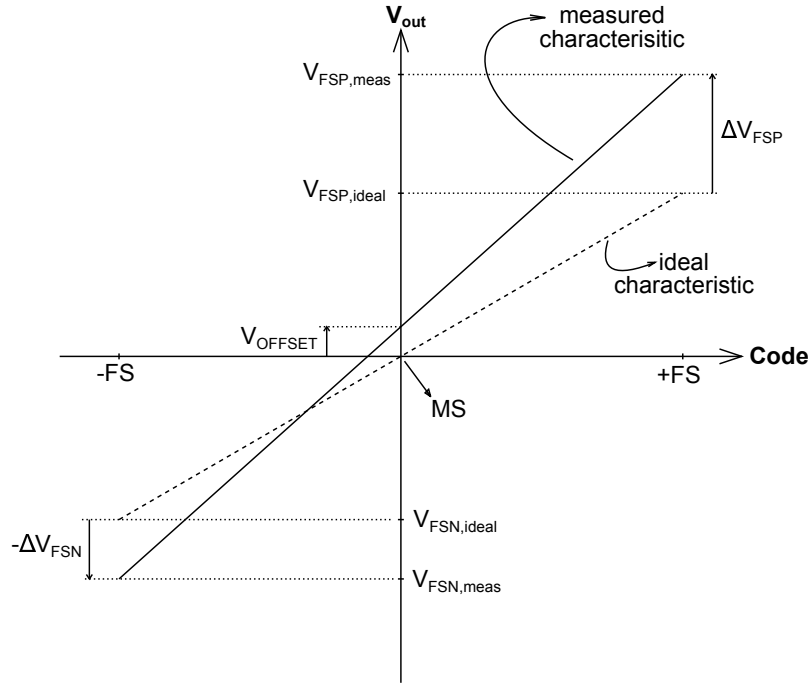


Figure 25: Gain and offset error

The midscale point (MS) corresponds to an input code for which the output voltage is zero. In a non-ideal DAC, the output for this code might be different from zero. In such a case we say that the DAC has some offset voltage. Offset voltage is the value of output for the input code corresponding to MS (see Figure 25)

The gain error of the DAC is defined as the difference of the full scale range of the ideal characteristic and the measured one, after the offset error is removed. So, according to Figure 25, the gain error normalized to LSB will be as follows [3]:

$$GE = \frac{\Delta V_{FSP} - \Delta V_{FSN} - V_{OFFSET}}{V_{LSB}} \quad (15)$$

2.4.1.2 Differential Nonlinearity Differential nonlinearity for an input code is defined as the difference between the height of the step corresponding to that code (with respect to the previous code) and the height of the step for that code in the ideal case (which is called LSB value). See Figure 26 [3]. Differential nonlinearity is normally defined with respect to the LSB height, so, the above value should be divided by LSB voltage value. For a DAC, usually the minimum and maximum value of DNL (for the whole range) is given. If the maximum value of DNL for a DAC is less than 1, that DAC is guaranteed to be monotonic. [3]

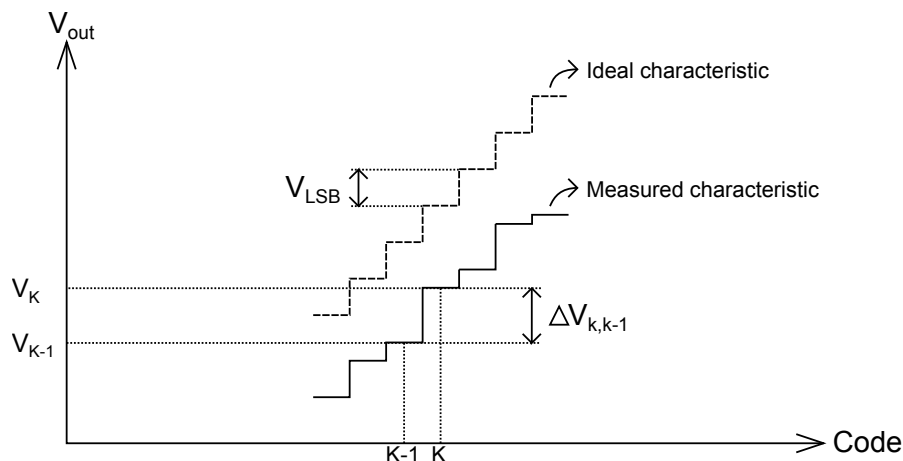


Figure 26: Differential Nonlinearity

2.4.1.3 Integral Nonlinearity Integral nonlinearity for each code is defined as the height distance of the actual DAC output for that code from the output characteristic of the ideal DAC. See Figure 27 [3].

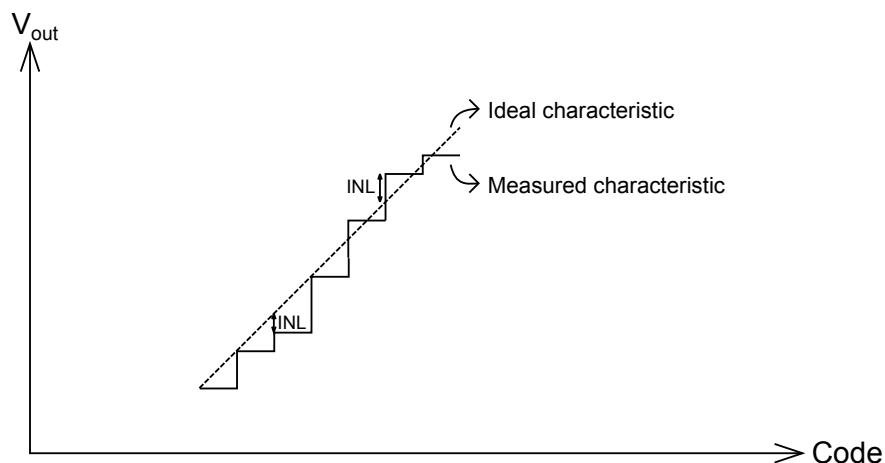


Figure 27: Integral Nonlinearity

By comparing the definition of DNL and INL we will find out that DNL for each code just depends on the accuracy of the DAC response for that code while in the case of INL, it depends not only on the accuracy of the DAC response for that code, but also on its accuracy for all previous codes. [3]

2.4.2 Dynamic accuracy

2.4.2.1 Spurious free dynamic range DACs are mixed signal circuits. So, except the harmonics of the carrier, we usually have some noise at other frequencies. Spurious free dynamic range is defined as the difference between the voltage level at the carrier frequency and the highest voltage level when not considering the carrier frequency. [15] See Figure 28 [3].

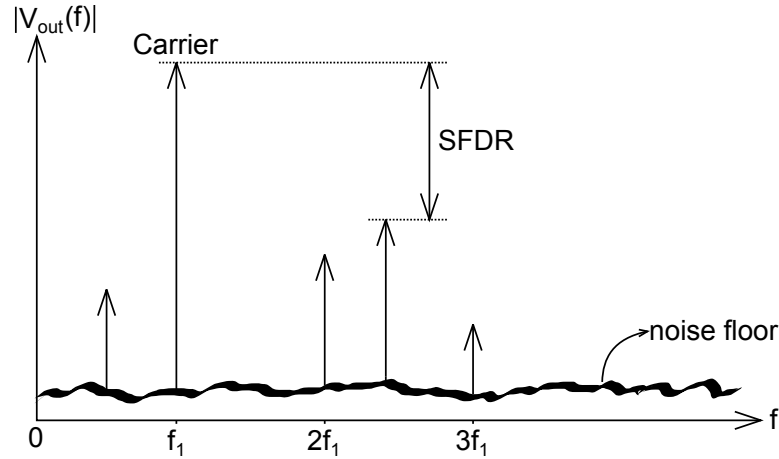


Figure 28: Spurious free dynamic range

2.4.2.2 Harmonic distortion If we give a tone signal to a non-linear circuit (which can be a DAC as well), at the output, we will have not only components at the frequency of the input tone, but also at integer multiples of the input frequency. These unwanted components at the output are called harmonic distortion components. (look at Figure 29 [3])

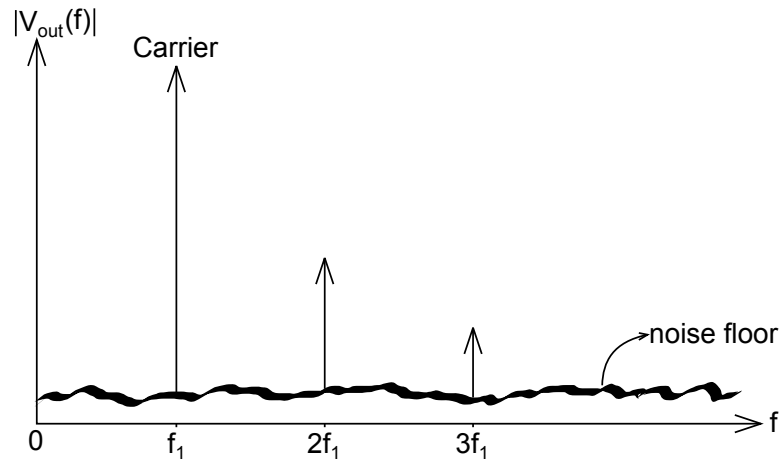


Figure 29: Harmonic distortion components

Harmonic distortion component at the $k.f_i$ frequency (f_i is the frequency of the input sinusoid) is defined as follows:

$$H_{Dk} = 20 \log_{10}(A_k/A_i) \quad (16)$$

where A_i is the magnitude of the output at the frequency f_i and A_k is the magnitude of the output at the frequency $k.f_i$. [3]

2.4.2.3 Intermodulation distortion If we give two sine waves with frequencies f_1 and f_2 to a nonlinear circuit (which can be a DAC as well), at the output not

only we will have components at f_1 and f_2 and their harmonics ($k.f_1$ and $k.f_2$, with k being an integer), but also at frequencies like $f_1 - f_2$, $2f_1 + f_2$, For clarifying this issue lets imagine a circuit with third order non-linearity as follow:

$$V_{out}(t) = k_1V_{in}(t) + k_2V_{in}^2(t) + k_3V_{in}^3(t) \quad (17)$$

By giving the input $V_{in}(t) = A_1\cos(\omega_1t) + A_2\cos(\omega_2t)$ to this circuit, the output will be as follows:

$$\begin{aligned} V_{out}(t) = & \frac{1}{2}k_2(A_1^2+A_2^2)+[k_1A_1+k_3\frac{3A_1^3+6A_1A_2^2}{4}]\cos(\omega_1t)+[k_1A_2+k_3\frac{3A_2^3+6A_1^2A_2}{4}]\cos(\omega_2t)+ \\ & \frac{1}{2}k_2A_1^2\cos(2\omega_1t) + \frac{1}{2}k_2A_2^2\cos(2\omega_2t) + \frac{1}{4}k_3A_1^3\cos(3\omega_1t) + \frac{1}{4}k_3A_2^3\cos(3\omega_2t)+ \\ & k_2A_1A_2[\cos(\omega_1t-\omega_2t)+\cos(\omega_1t+\omega_2t)]+\frac{3}{4}k_3A_1^2A_2\cos(2\omega_1t+\omega_2t)+\frac{3}{4}k_3A_1A_2^2\cos(\omega_1t+2\omega_2t)+ \\ & \frac{3}{4}k_3A_1^2A_2\cos(2\omega_1t - \omega_2t) + \frac{3}{4}k_3A_1A_2^2\cos(2\omega_2t - \omega_1t) \quad (18) \end{aligned}$$

The last two terms in the above expression (which are at frequencis $2f_1 - f_2$ and $2f_2 - f_1$) are the most problematic components. Because in the case that f_1 and f_2 are close together, these components will be placed at close neighboring of the wanted components of the output (i.e., components at f_1 and f_2). These components are called third order intermodulation products. [15]

3 Solutions

Most of previous works done in the area of DAC calibration deal with the problem of amplitude error in DACs; as some examples we can name [16], [17], [18], [19], [20], [21], [22]. Timing error calibration is a quite untouched area. The most serious work done in this area that we could find was [23]. That is a comprehensive work that tries to solve both amplitude and timing error in I-Q transmitters. In this thesis the focus is just on timing errors in DACs. A new approach is introduced in this work that is explained in this chapter.

3.1 Introduction

The general idea is illustrated in Figure 30. We choose one current cell as reference and calibrate other cells with respect to that, one by one. If the selected cell has more time delay than the reference cell, we decrease the delay in its delay unit. And, on the other hand, if it has less time delay than the reference cell, we increase the delay in its delay unit.

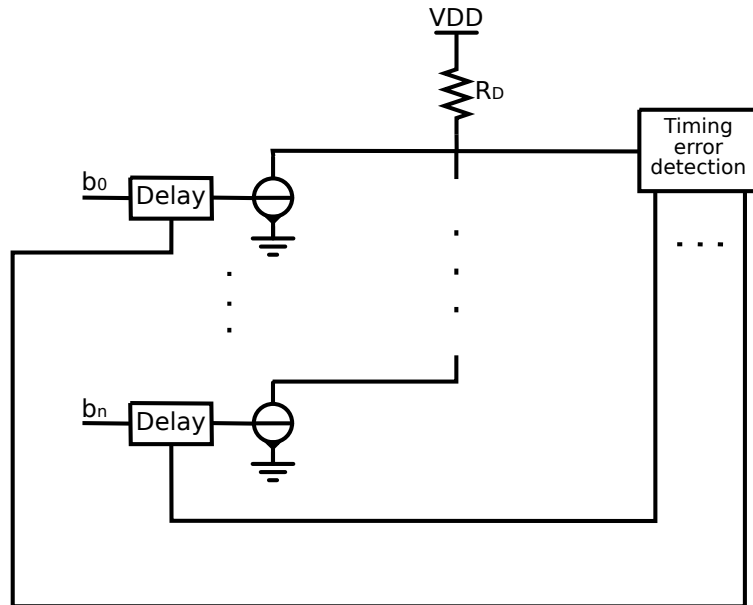


Figure 30: The general idea of the proposed approach

In the following the structure of the timing error detector and the delay unit are explained.

3.2 Timing error detector unit

3.2.1 Theory

The circuit diagram of the error detector unit can be found inside Figure 31.

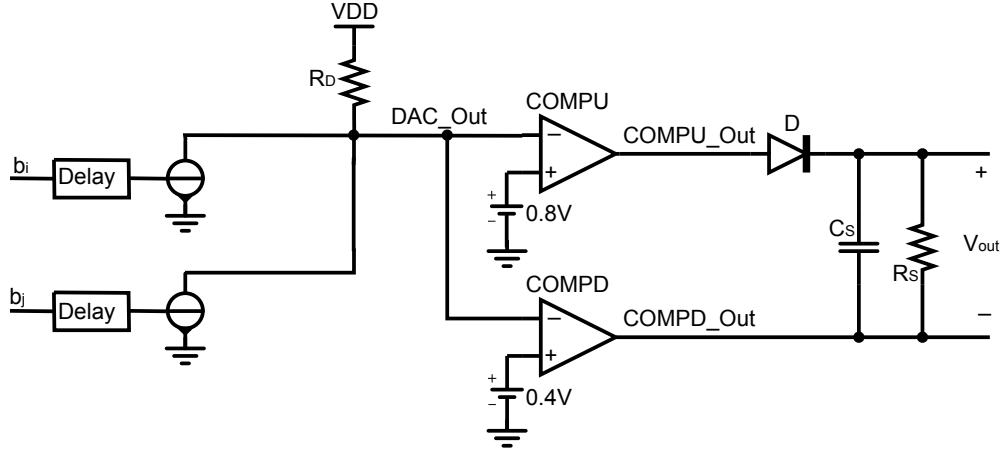


Figure 31: Illustration of error detector

For measuring the timing error between two selected cells, we give identical (with the same phase) square wave pulses to their input (The input to the other cells is zero). If there is a timing difference between these cells, the signal at the output of DAC will be as follows:

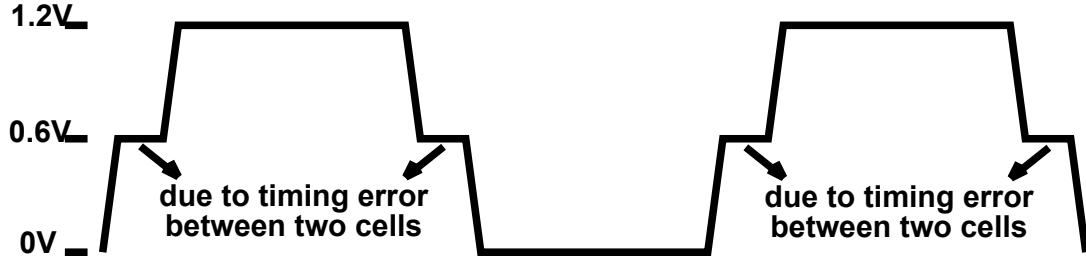


Figure 32: The output of an uncalibrated DAC to measuring signals

In Figure 32, the assumption is that the voltage that each cell contributes to the output, when being on, is 0.6V. We know that in practice this voltage is much less than this value. This assumption is used in here for the sake of simplicity. However, we agree that if this voltage is so small, the job of the error detector unit will be more difficult. But, even in that case, the solution is not difficult; We can have two resistors in place of R_D which can be selected by switches. One with a larger value that will be used for the calibration process and the other one with a smaller value that will be used during the normal operation of the DAC. Or another solution is the insertion of an amplifier stage at the output of the DAC.

In Figure 33, the signals in some nodes of the error detector unit are illustrated:

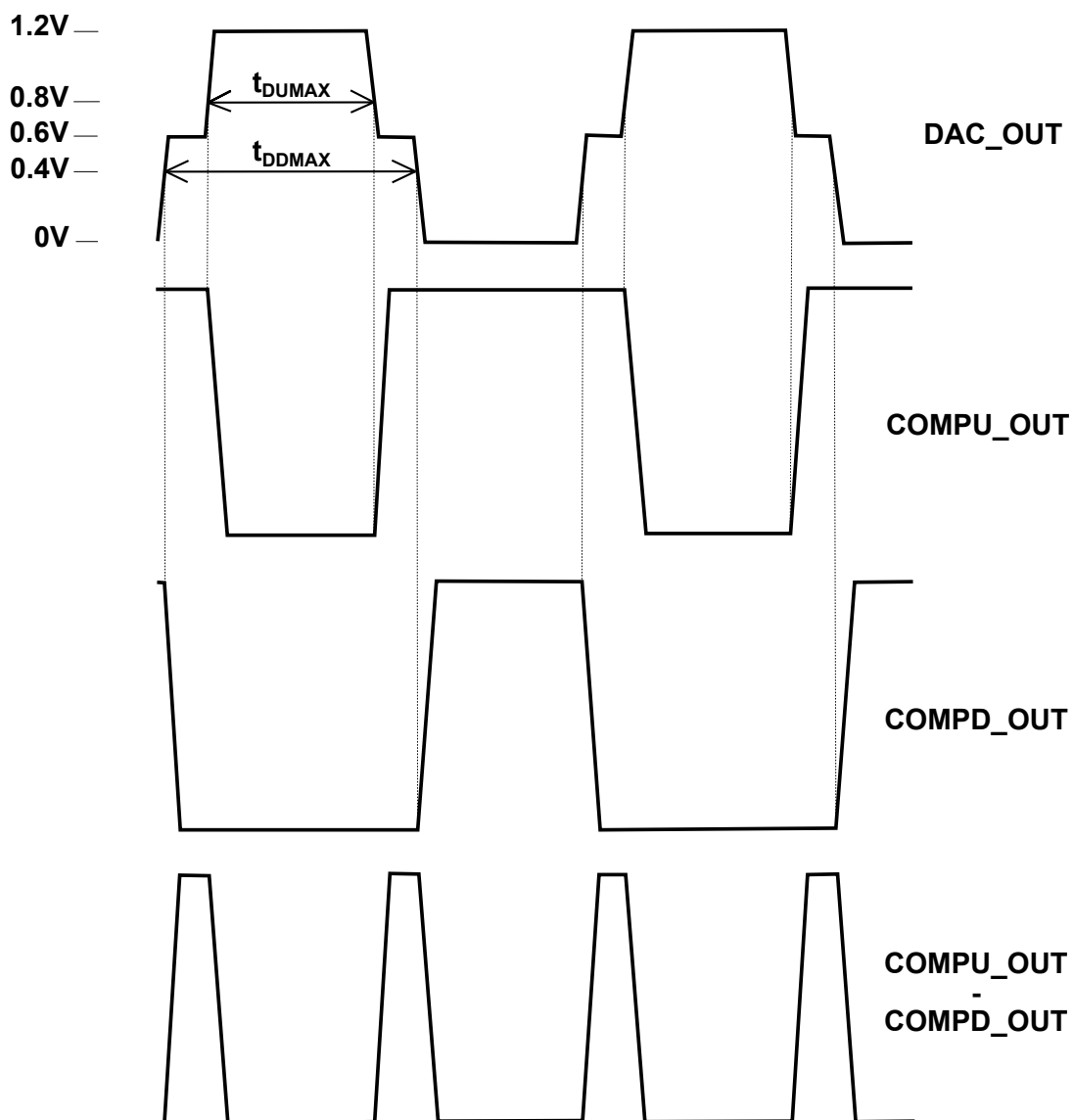


Figure 33

We can see that the duty cycle of the COMPU_OUT - COMPD_OUT signal is in direct relation with the timing difference of the two cells. So, if we can measure the power of this signal by some means, then all we need to do will be changing the delay at the input of the target cell (the cell that is under comparison with the reference cell) to minimize this power. The power measurement task is done by converting this signal to DC with a diode and a capacitor (Then this DC level can be measured with an ADC. This part is not considered in this work).

In Figure 33, for the sake of simplicity, the comparators are assumed to be without delay (This delay is taken into account in the simulation part). However, we know that in practice comparators have some tens of picoseconds response delay. In Figure 33 we can see that the maximum delay that the COMPU can have is equal to t_{DUMAX} and this value for the COMPD comparator is equal to t_{DDMAX} . So, by assuming the use of identical comparators for both branches, the maximum delay

that the comparator is allowed to have is t_{DUMAX} . By decreasing the frequency of the measurement signal we can avoid the problems that the response delay of the comparator creates. On the other hand, a higher frequency for this signal will result in a higher DC level at the output of the error detector circuit and it will enable us to detect smaller timing errors (the resolution will increase). So, it is a wise idea not to decrease the frequency more than what is needed. In the simulations performed in the next part, the response delay of the comparators is assumed to be 50ps and a frequency of 5GHz is selected for the measurement signals.

In Figure 31 we can see that the timing error detector circuitry has two branches. Even if we use identical comparators for both branches and design the layout such that the signal paths for both branches have exactly the same lengths, still there is this probability that some timing difference is created between them during fabrication process. And in the case of having some timing difference between two branches, the signals will not be exactly as what is illustrated in Figure 33. Now the question is whether it affects the functionality of this circuit or not. Fortunately, even in this case, still the power of the output signal is in direct relation with the timing error of the two cells under measurement. This issue is considered in the following:

There might be two cases; either the branch including COMPU comparator has some delay with respect to the other branch, or vice versa. The wave forms for these cases are illustrated in figures 34 and 35 respectively. In each case we have investigated the problem for two possibilities; either this timing difference between two branches is less than the timing error between two cells under measurement, or it is larger than that. The output of the error detector circuitry for the first possibility is illustrated in figures 34-e and 35-e. We can see that although the signal is not symmetric like what we had in Figure 33, but still by increasing the timing error between two cells (i.e. D), the width of both pulses will increase. The output of the detector circuitry for the second possibility is illustrated in figures 34-g and 35-g. The negative pulse will be eliminated by the diode and only the positive pulse will contribute to the level of the DC voltage at the output. We can see that the width of this pulse ($d_2 + D$) is in direct relation with the timing error of the two cells (D). However, in this case we are losing the contribution of one of the pulses which, at least theoretically, will decrease the resolution of the error detector unit. So, although this error detector circuit will work even in the case that there is some timing difference between its two branches, for getting the best result, the target at the design level should be to not have any timing difference between two branches.

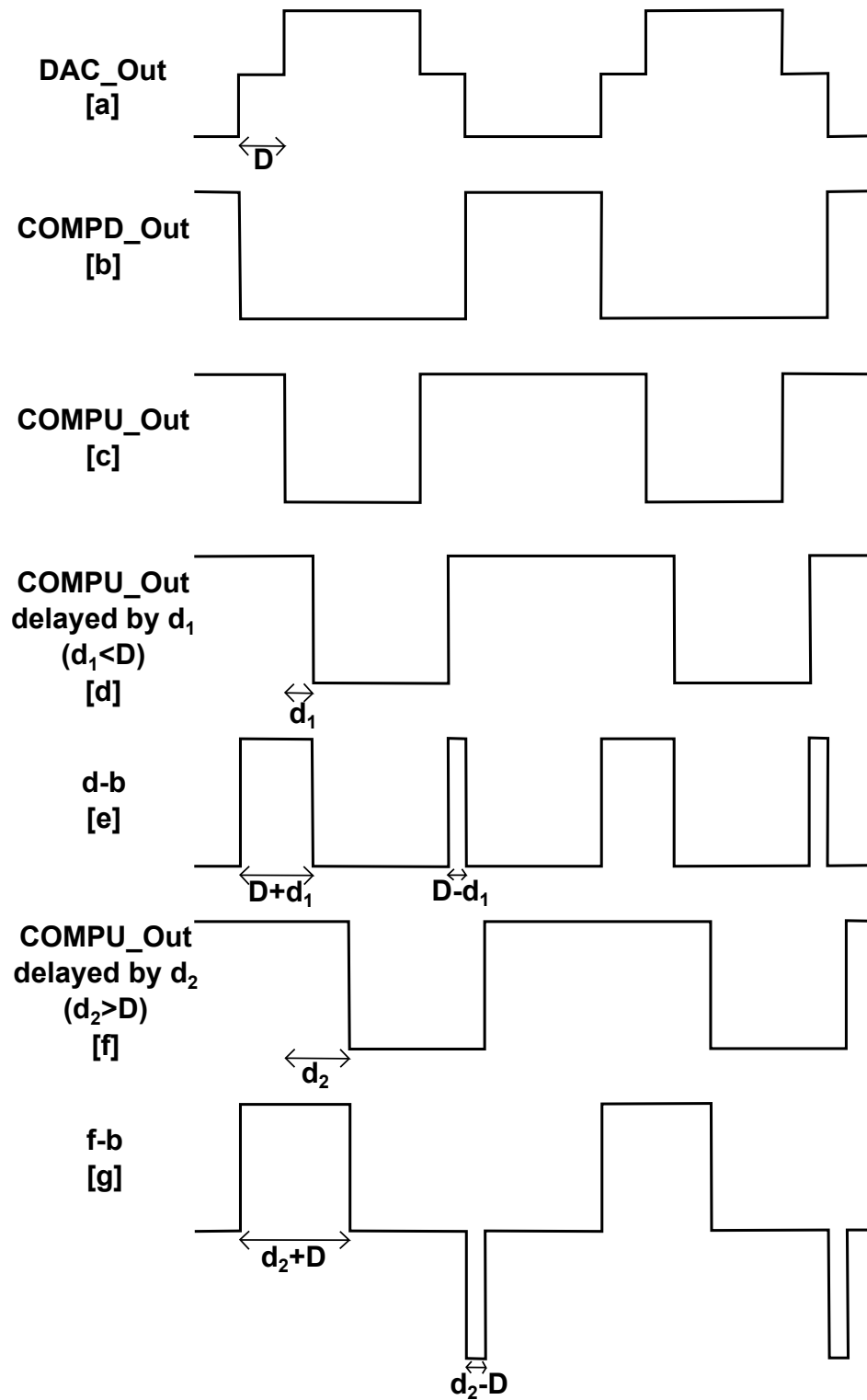


Figure 34: the waveforms by assuming the upper branch of the error detector has some delay with respect to the downer branch (D : timing difference between two cells of the DAC. d_1 and d_2 : the amount of delay that the upper branch has with respect to the downer branch)

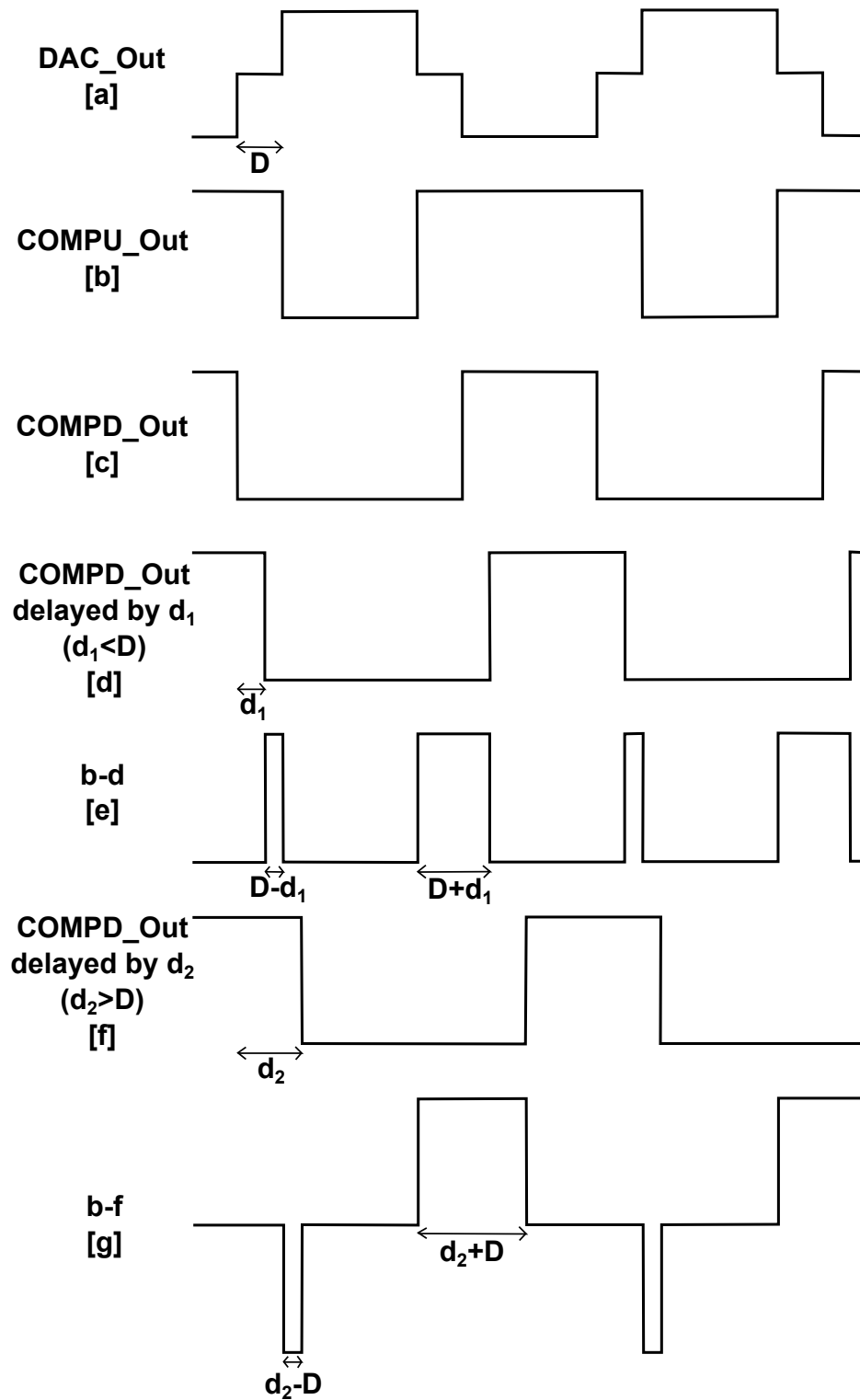


Figure 35: the waveforms by assuming the downer branch of the error detector has some delay with respect to the upper branch (D : timing difference between two cells of the DAC. d_1 and d_2 : the amount of delay that the downer branch has with respect to the upper branch)

3.2.2 Simulation results

The error detector in Figure 31 was simulated in Eldo. The input signal to the detector was a 5GHz signal with a timing error of 5ps between two cells. A response delay of 50ps was selected for the comparators. A value of 30pF was given to CS and 10KOhms to RS. The simulation results are presented in figures 36 and 37:

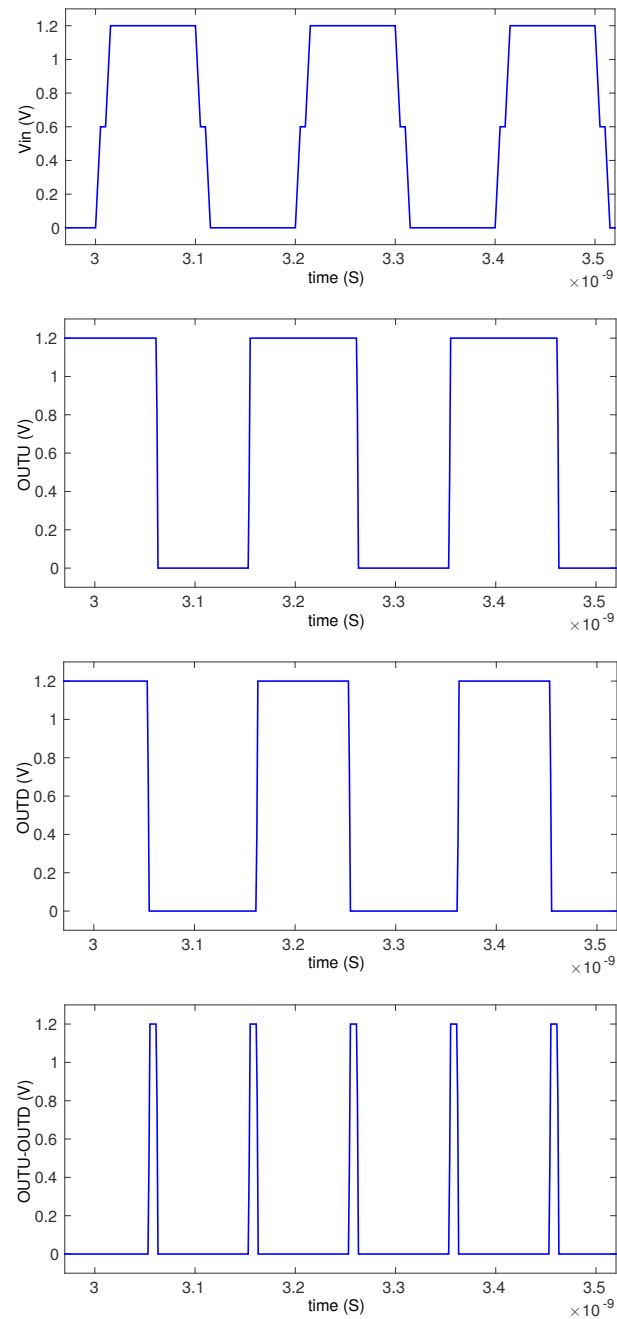


Figure 36: Simulation results for the circuit of Figure 31, with 5pS timing error between two cells

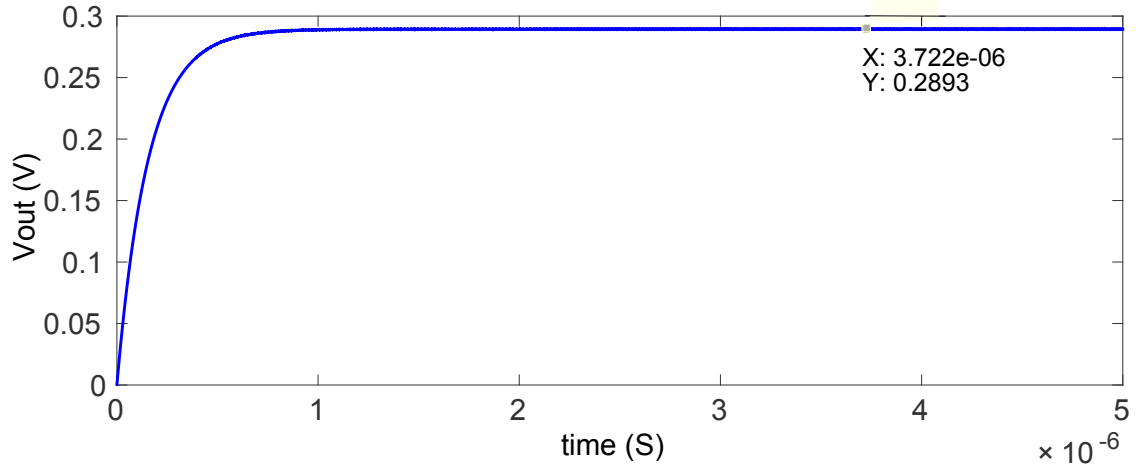


Figure 37: The DC voltage at the output of the error detector

This simulation was done for some other values of timing error as well. The results are presented in Table 2:

Timing error between two cells	The DC voltage at the output of error detector
1 PS	191 mV
2 PS	216 mV
3 PS	243 mV
4 PS	274 mV
5 PS	289 mV

Table 2: Simulation results for the error detector unit

3.3 Delay unit

The first structure that we tried for the delay unit is illustrated in Figure 38:

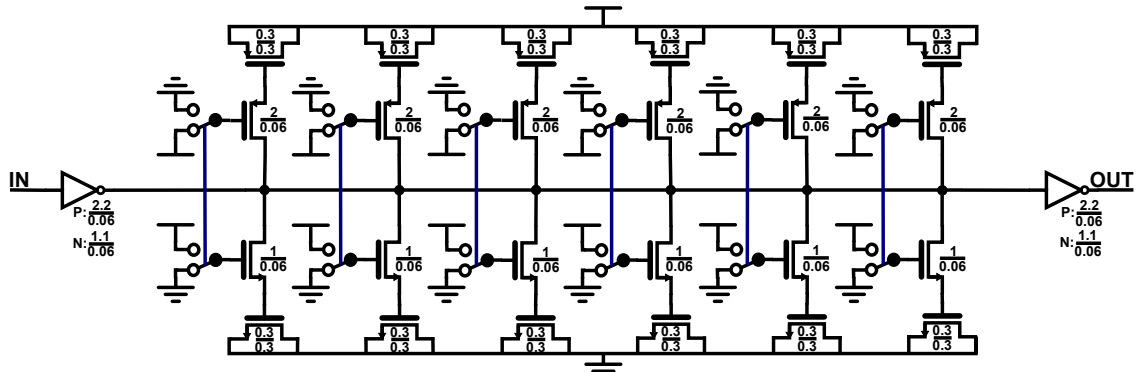


Figure 38: The primary idea for the delay unit structure

In order to investigate the performance of this structure, we gave a rectangular pulse train to its input and measured the time delay it would take for the pulse to appear at the output. We measured this time delay for different states of the switches. The pulses were fluctuating between 0 and 1.2V. We defined the reference point for measuring the time delays, the moment when the pulse passes the midpoint, i.e., 0.6V. We measured the time delay for both the rising and falling edges of the pulse. This issue is illustrated in Figure 39:

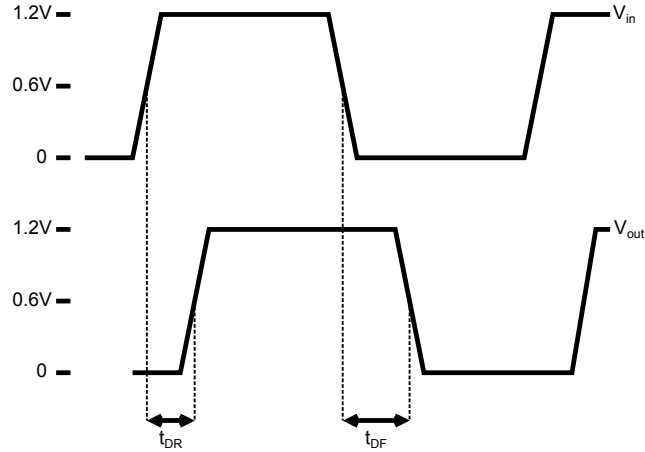


Figure 39: Time delay definitions for investigating the performance of the delay unit

The measured time delays are presented in Table 3:

Number of capacitor pairs connected	0	1	2	3	4	5	6
Delay added to the rising edge of the pulse; t_{DR} (pS)	32	35	38	42	46	50	54
Delay added to the falling edge of the pulse; t_{DF} (pS)	31	34	37	39	42	45	48
$t_{DR} - t_{DF}$ (pS)	1	1	1	3	4	5	6

Table 3: Measured delay times of the delay unit of Figure 38

We can see that for the case of having 0, 1, and 2 capacitor pairs connected the delay for the rising edge is just 1pS more than for the falling edge. But, as the number of capacitor pairs becomes more, this difference increases, such that for the case of having all capacitor pairs connected, this number is 6pS. This difference is something not acceptable and makes problems in the performance of the DAC. We expect the delay unit to add the same delay to both edges of a signal, otherwise that signal will be changed.

The cause for this behavior of this delay unit, is the wide range of the load for the first inverter; which varies between 0 to 6 capacitor pairs. The inverter is not capable of maintaining a linear behavior for such a large range. In order to solve this problem, we added two more inverters between the inverters of the delay unit in Figure 38, as illustrated in Figure 40:

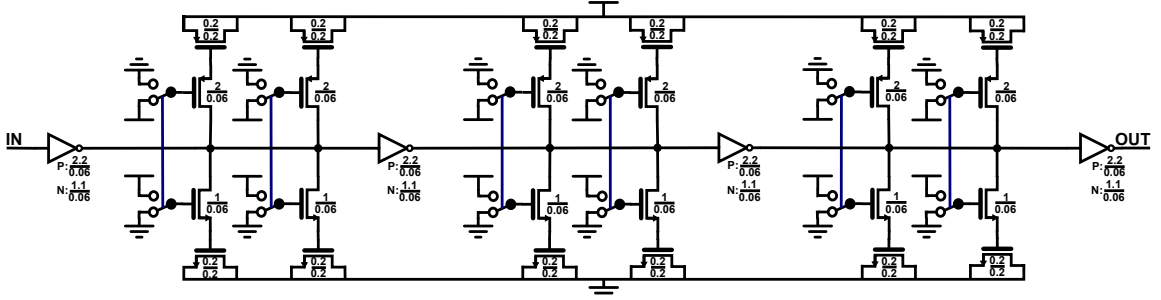


Figure 40: Improved delay unit

In this way, the range of the load for the first three inverters varies between 0 to 2 capacitor pairs, which is much more smoothed compared to the case of previous structure. The delay times created by this structure for different states of the switches, was measured. The results are illustrated in Table 4 ¹:

Number of capacitor pairs connected	0	1	2	3	4	5	6
Delay added to the rising edge of the pulse; t_{DR} (pS)	58.6	62.4	66.7	70.9	74.7	78.7	81.9
Delay added to the falling edge of the pulse; t_{DF} (pS)	59.1	63	66.9	71.2	75.7	78.5	80.8
$t_{DR} - t_{DF}$ (pS)	-0.5	-0.5	-0.2	-0.3	-1	0.2	1.1

Table 4: Measured delay times of the delay unit of Figure 40

We can see that the performance of the circuit has improved significantly. In most cases the difference between the delay added to the rising edge and the falling edge is less than 1pS. The worst case which is 1.1pS, belongs to the state that all 6 capacitor pairs are connected. Having all capacitor pairs connected means that both capacitor pairs before the final inverter are also connected. This fact decreases the slop of the pulse at the input of the final inverter, and just one inverter does not improve it well. In other words, by adding another inverter (or two, for having the same pulse at the output as it is at the input), after the final inverter in Figure 40, we can even decrease this 1.1pS to less than 1pS.² In addition, tuning the size of transistors might improve the accuracy of the circuit even more.

The resolution of the delay unit of Figure 40 is about 4 pS. This can be changed by changing the size of capacitors.

¹The results in this table are not rounded as it was in Table 3, and are written in the tenth of picoseconds resolution

²We simulated this idea and the results were approving it

4 Implementation and simulation results

This chapter includes two part; "Implementation" and "Calibration process and simulation results". In the first part the circuit diagram of the implemented DAC with all the details is presented. In the second part after giving an explanation about the calibration process and performed simulation, the simulation results are presented.

4.1 Implementation

In figures 41 to 46 you can see the diagram of the designed DAC:

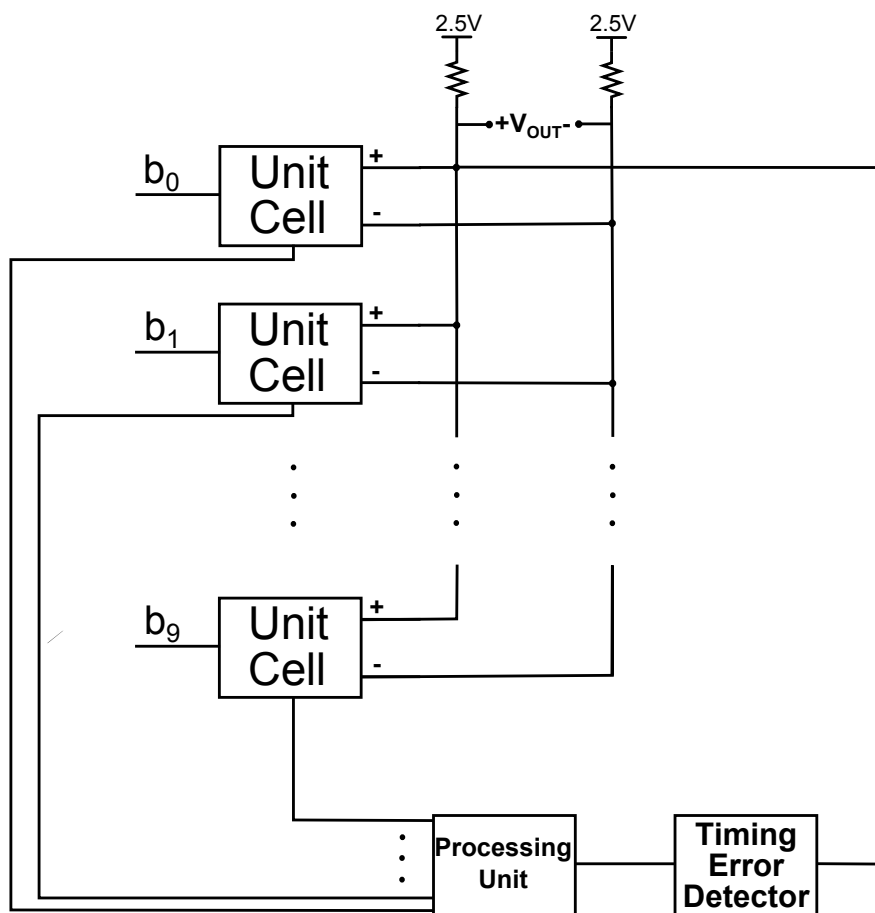


Figure 41: The whole DAC and error correcting unit

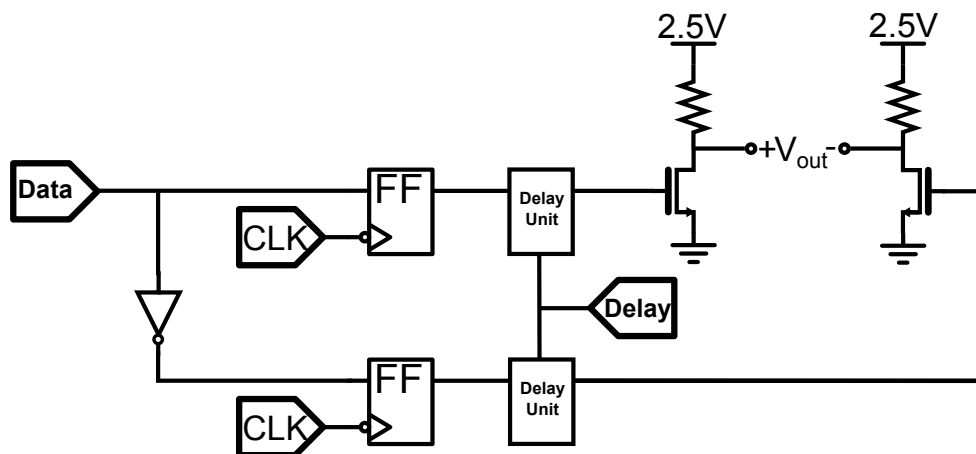


Figure 42: Unit cell

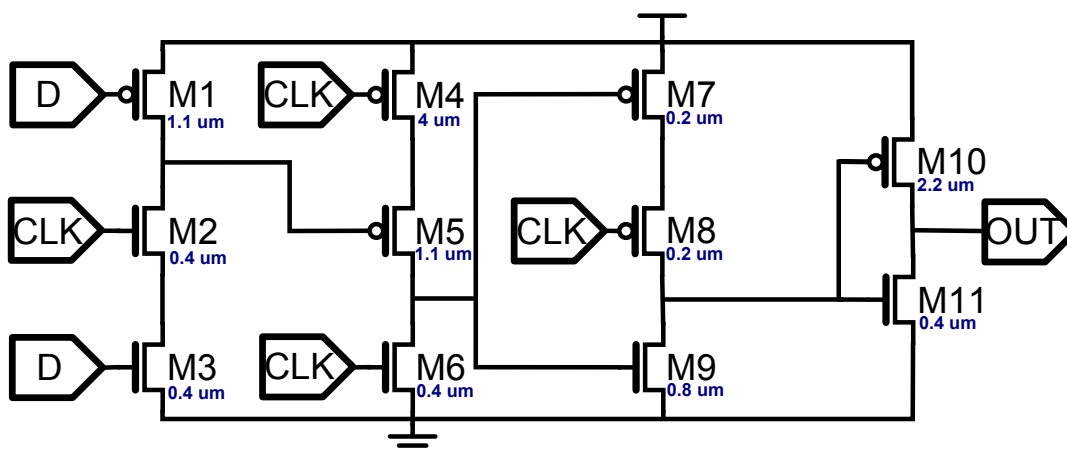


Figure 43: The flip-flops inside unit cell (the length of all transistors is 0.06um)

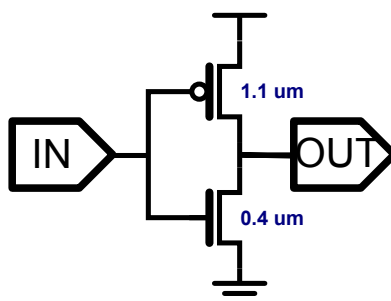


Figure 44: The inverter inside unit cell (the length of all transistors is 0.06um)

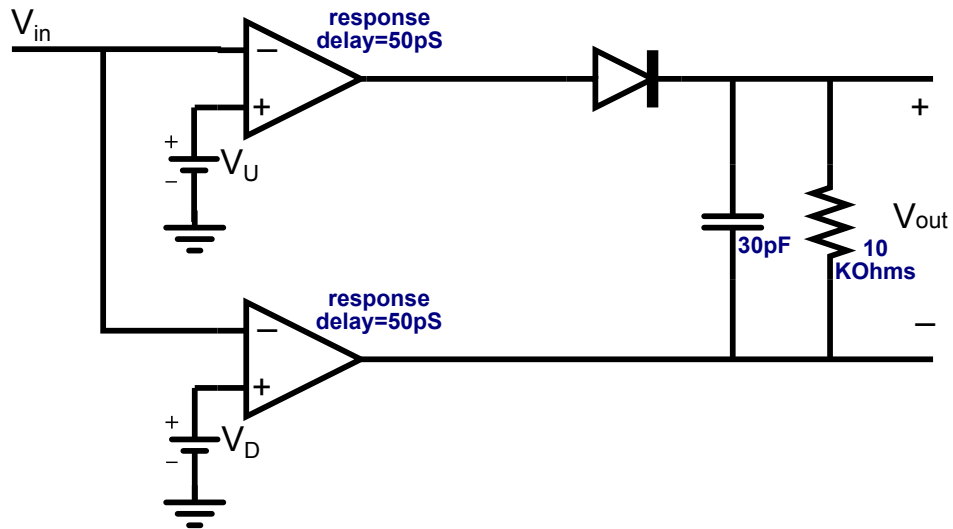
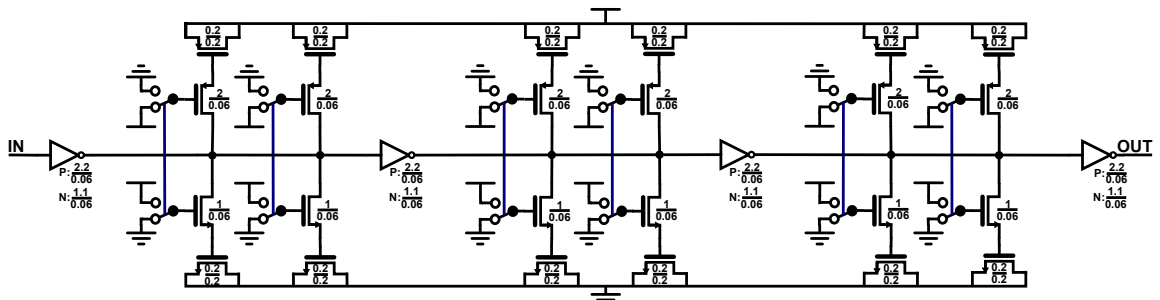
Figure 45: Timing Error Detector³

Figure 46: The delay unit

³If the DAC structure is thermometer coded, V_U and V_D will be constant for any pair of branches that are to be calibrated. But, if it is binary coded, the optimum value of V_U and V_D will be different for any selected pair of branches. In the case of the simulations done in this work, the DAC was binary coded and the calibration was done between branches 8 and 9. The values selected for V_U and V_D were 2.3V and 1.9V, respectively

4.2 Calibration process and simulation results

For calibrating the DAC, a calibration process should be performed at start up of the IC. Then the position of the switches of the delay units of all branches will be saved in a memory and will be applied to the switches during normal work of the DAC.⁴ The calibration process is as follows:

One branch will be selected as the reference branch. Half of the capacitor pairs of the delay unit of this branch will be included in the circuit and half of them will be disconnected. In the case of this design it means 3 pairs of capacitors. Now, other branches should be calibrated with respect to this branch one by one. So, we choose one of the other branches and after calibrating that we continue with the remaining branches in the same way. We disconnect all the capacitors of the delay unit of the branch that we are going to calibrate. Then we apply two synchronized pulses to this branch and the reference branch (The input to other branches are zero). Now, we measure the DC voltage at the output of the timing error detector circuit and record it. Then we involve one pair of the capacitors of the delay unit of the branch that is under calibration and record the DC voltage at the output of the timing error detector unit again. We continue like this until all six pairs of capacitors of the delay unit of the branch that is to be calibrated are connected. Now, by comparing the recorded DC voltages, we choose the state as the best calibrated state that has resulted in minimum DC voltage. We do the same procedure for the rest of the branches.

We designed the circuit of Figure 41 in eldo, in 65nm CMOS technology. The DAC was intended to work with 2GHz, binary coded input. In order to show the performance of the design, we inserted a delay of 8pS in branch 9, as depicted in the figure below:

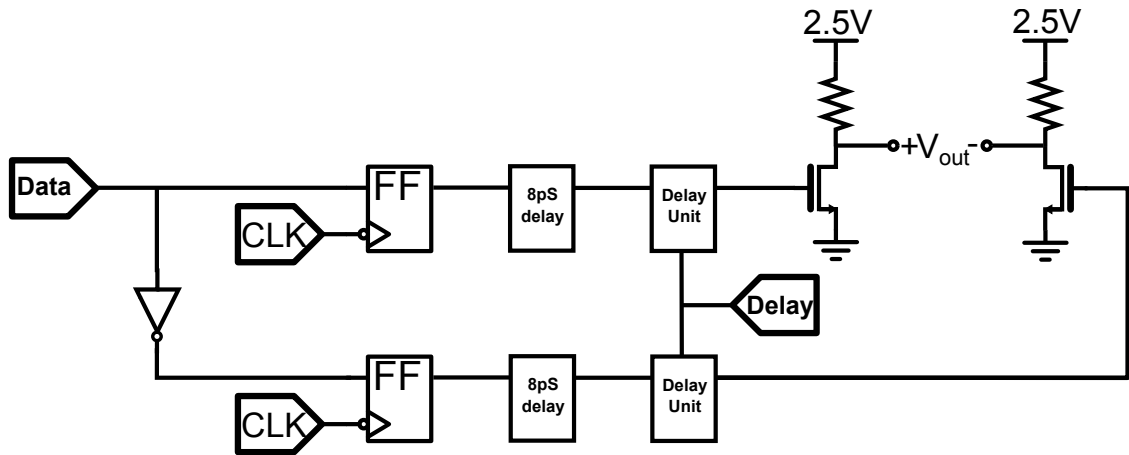


Figure 47: Unit cell of branch 9 with 8pS inserted delay

Then by choosing branch 8 as the reference branch, we tried to calibrate branch 9 with respect to it. So, we connected 3 of 6 capacitor pairs of the delay unit of

⁴Obviously, in this simulation we have not designed the memory and the task for this part is done manually

branch 8. In this simulation for the case of simplicity, we didn't insert timing error to branches 1 to 8. So, they are already all calibrated with respect to each other. So, in order to not ruin their calibration, we connected 3 of 6 capacitor pairs of each of the delay units of branches 1 to 7 as well.

By referring to Table 4 we can see that delay units of branches 1 to 8 add a delay of approximately 71pS to the signal flow. For compensating the 8pS delay that is inserted in branch 9, the delay that the delay unit of this branch adds, should be 8pS less than other branches, i.e., 63pS. In Table 4 we can see that for the case of having one capacitor pair connected, the delay created by the delay unit will be approximately 63pS. So, we expect the calibration process to return minimum DC voltage for the case of one capacitor pair connected. The calibration process was done. Its results are tabulated below:

Pairs of capacitors connected in the delay unit of branch 9	The DC voltage at the output of the error detector
0	139 mV
1	129 mV
2	145 mV
3	146 mV
4	144 mV
5	165 mV
6	162 mV

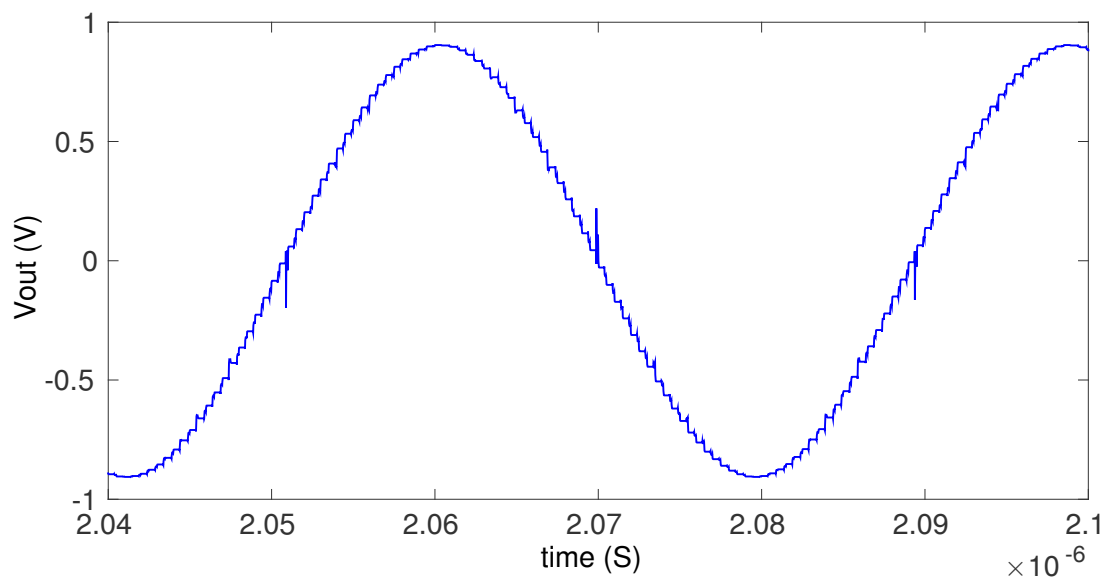
Table 5: Calibration process results

As it was expected, we can see that we have got the minimum DC voltage for the case of having one capacitor pair connected.

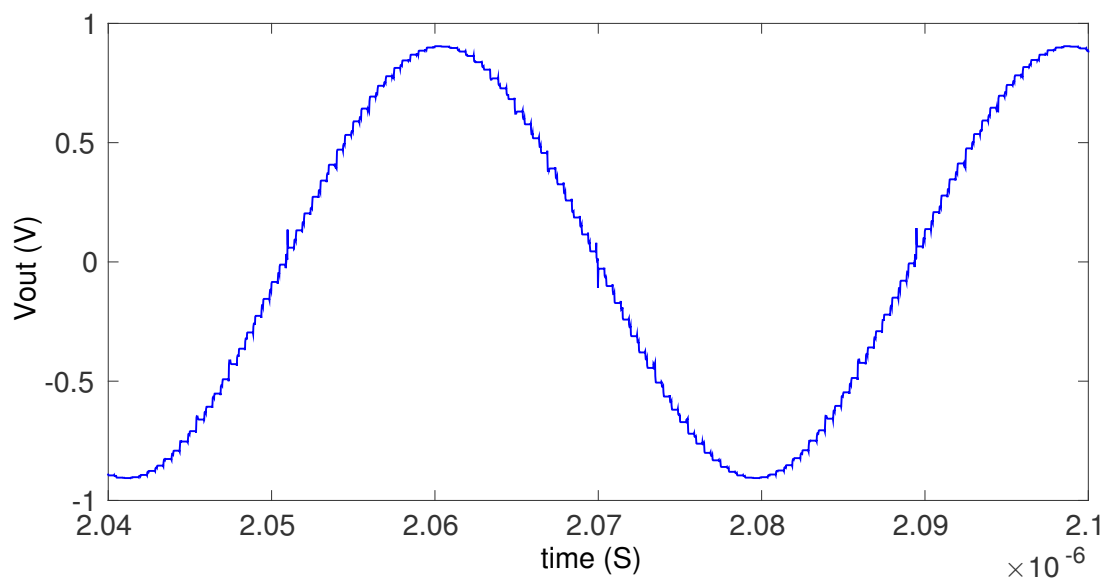
In order to show the performance of the circuit, we gave two signals as input to the DAC and measured the output before and after calibration.

One of the inputs was a 26MHz sinusoid with a digital data rate of 2GHz (2 G Samples/Second). In Figure 48 you can find the output of the DAC to this signal in time domain, before and after calibration. We can see how the calibration has decreased the amplitude of the glitches caused by timing error in branch 9. In Figure 49 the output of the DAC to this signal in frequency domain, before and after calibration, is illustrated. We can see how the noise floor has decreased after calibration.

The other input was a 20MHz wide-band signal at the carrier frequency of 2GHz. The output of the DAC for this case, both before and after calibration, is illustrated in figures 50 and 51. The first one is in time domain and the second one, in frequency domain. In time domain, we have zoomed in a short range of time in order to give a clearer view of the glitches. You can see that in the frequency response figure, the amplitude of the signal at the first harmonic and the second biggest component (excluding harmonics) is illustrated. So, the SFDR before and after calibration is 64.39dB and 67.55dB , respectively. So, we can see that calibration has resulted in 3.16dB increase in SFDR.

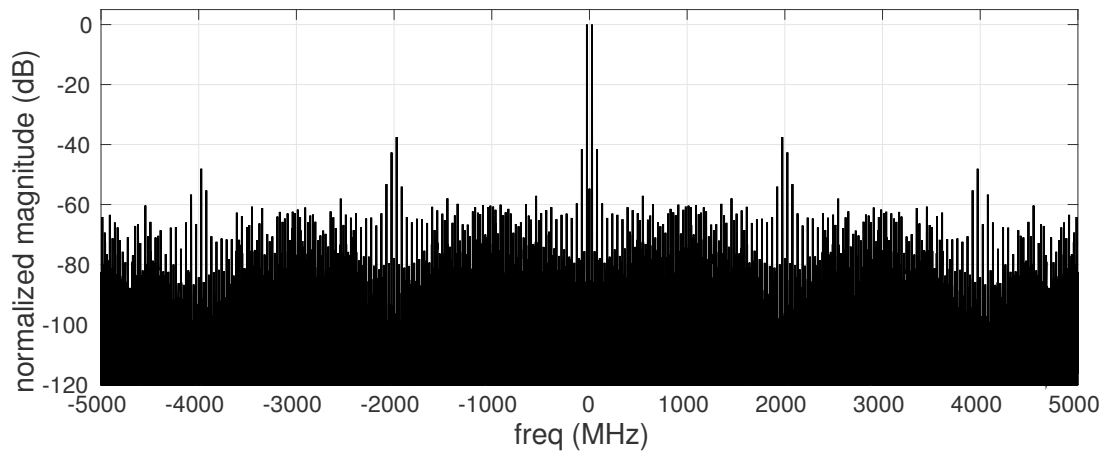


(a)

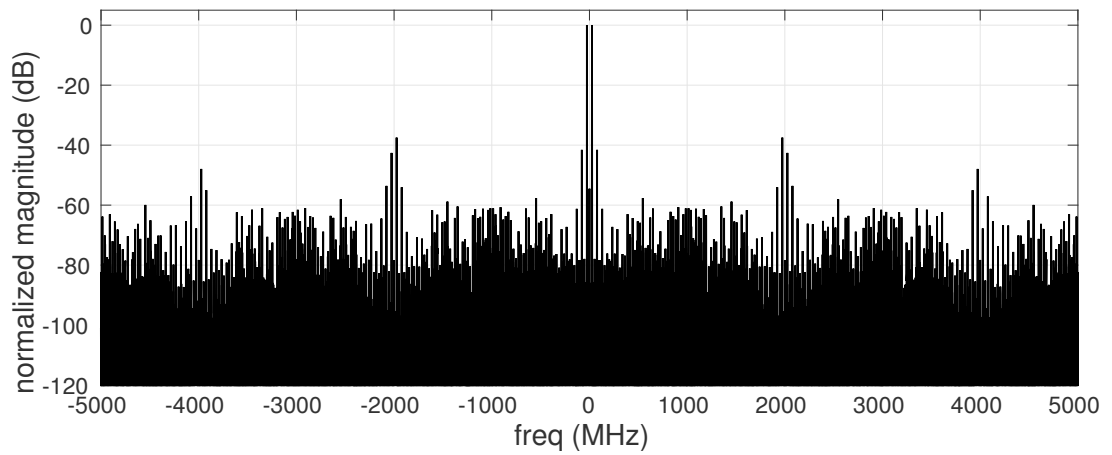


(b)

Figure 48: Time response of the DAC to a tone; (a) Before calibration (b) After calibration

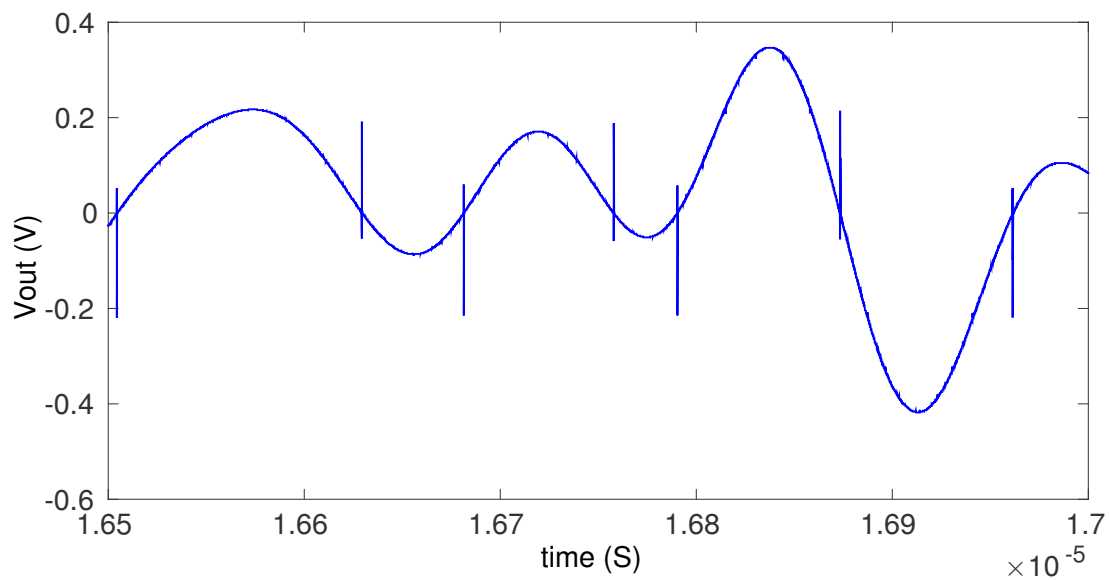


(a)

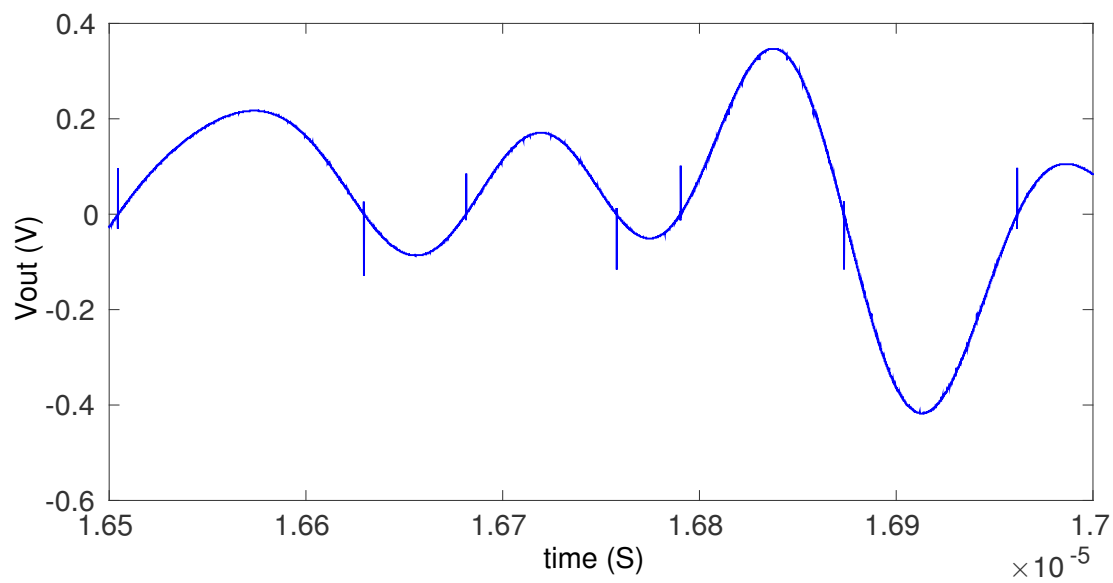


(b)

Figure 49: Frequency response of the DAC to a tone; (a) Before calibration (b) After calibration



(a)



(b)

Figure 50: Time response of the DAC to the wide-band signal; (a) Before calibration
(b) After calibration

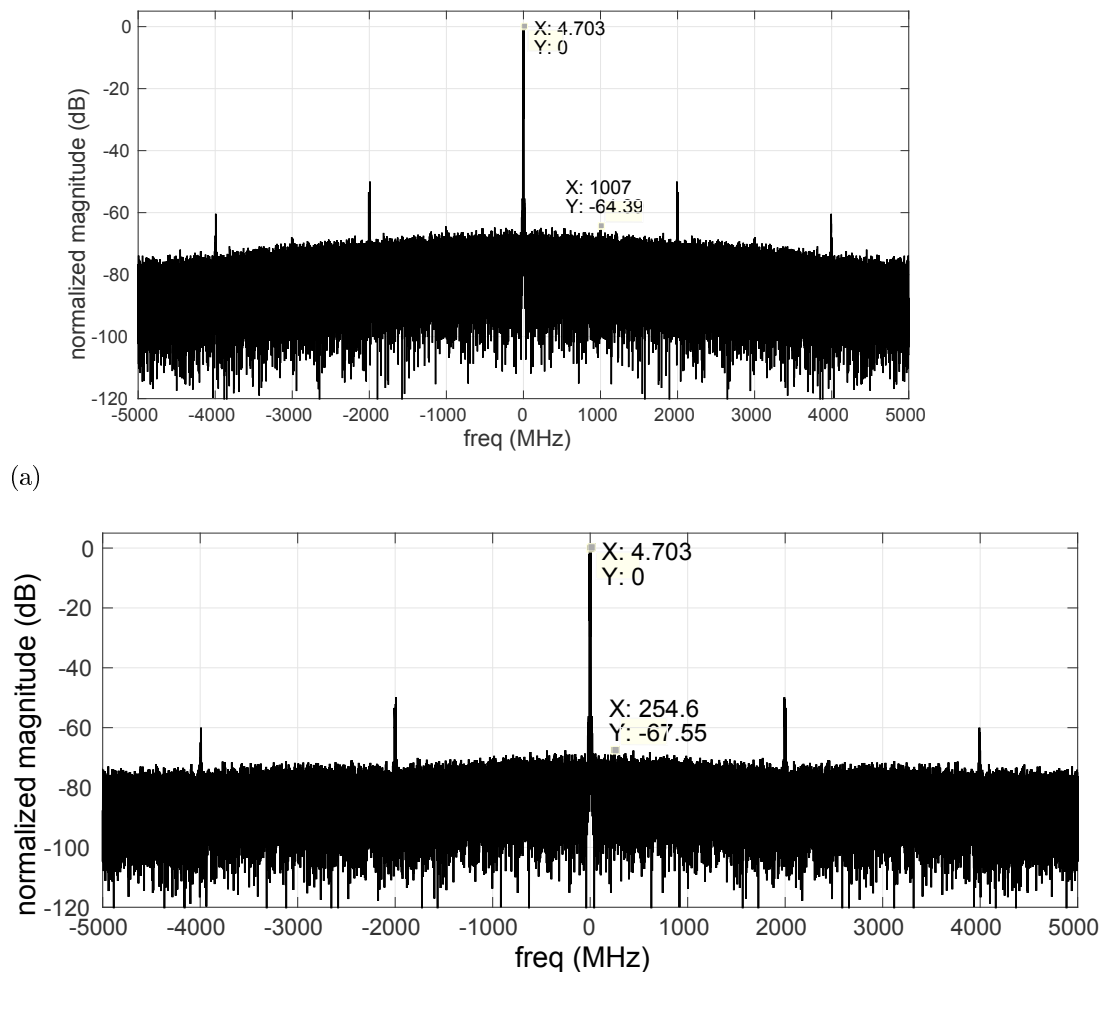


Figure 51: Frequency response of the DAC to the wide-band signal; (a) Before calibration (b) After calibration

5 Conclusions

The question of this thesis was finding a solution for the problem of glitches that appear at the output of the DACs that work at high frequencies, due to the timing error among different branches. A solution for this problem was introduced in this thesis. This approach can be briefly explained as follows:

We insert configurable delay blocks in all branches. The delay these blocks generate can be changed by a digital control voltage. Through a calibration process we give specific pulses to the branches and measure their timing error with respect to each other. This timing error measurement is done through a block called timing error detector. The circuit designed for timing error detector unit has a proper resolution and accuracy and is designed in a way that its performance is not affected by the timing error it might generate itself. Then by changing the control voltage to the delay blocks of the branches we try to minimize their timing error.

A 10 bit binary coded DAC was designed in 65nm technology in eldo simulator. We created a timing error of 8pS in the MSB branches and calibrated it through the approach explained above. We tried to check the efficiency of this approach by giving a 26MHz tone and a 20MHz wide-band signal to the DAC, before and after calibration. The carrier frequency of both signals was 2GHz. In both cases the amplitude of the glitches in time domain had decreased significantly after calibration. There was a notable decrease in the noise floor of the output for the tone after calibration and the SFDR of the output for the wide-band signal was increased by about 3.2dB after calibration.

6 Appendix A- Dynamic flip-flops

The circuit diagram of the flip-flop used in this DAC is illustrated in Figure 52.

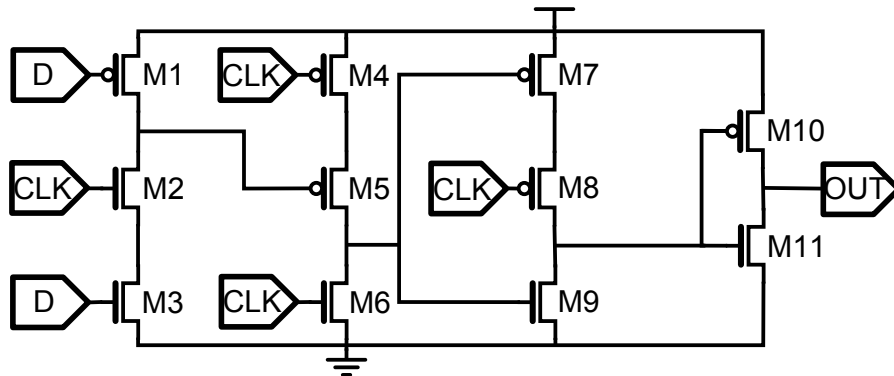


Figure 52: The circuitry of the flip-flop used in this thesis

It is a True Single Phase Clock (TSCP) flip-flop; meaning that it doesn't need the invert of clock for its operation [24]. It assures us from not suffering any problems caused by the timing delay between CLK and CLK'. We can see that this flip-flop, in contrast with classic flip-flops, has no feedback. So, the question that might arise is "how does it memorize the previous state?". It memorizes the previous state by storing it in the parasitic capacitors of the circuit. These kinds of flip-flops are called dynamic flip-flops. The advantage of them over flip-flops with feedback is their higher speed. But, these flip-flops do not work correctly at very low clock frequencies. It is because their parasitic capacitors will lose their charge [25]. In the following we have analyzed how this flip flop works.

An appropriate way for analyzing dynamic flip-flops is to first analyze the flip-flop in different stable states and then investigate what happens during transitions from one state to another one. In the case of this flip-flop, considering the values of Data and Clock inputs, we will have four different states. They are illustrated and analyzed in figures 53 to 56.

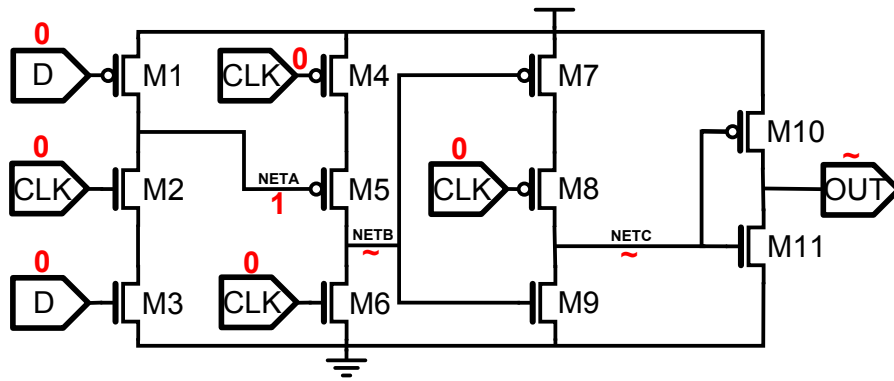


Figure 53: Data=0, CLK=0

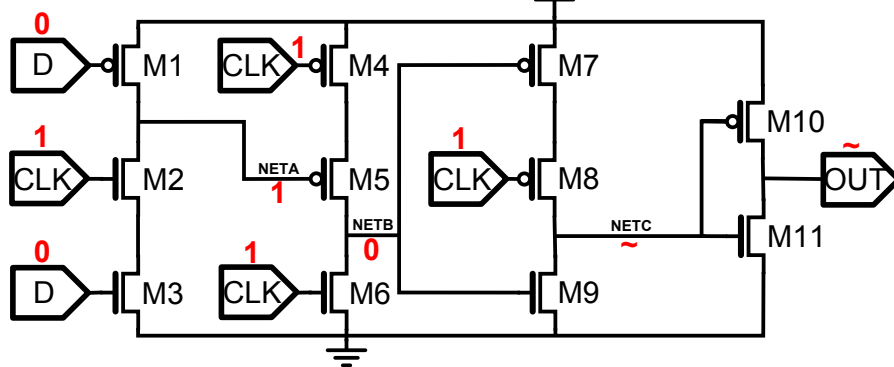


Figure 54: Data=0, CLK=1

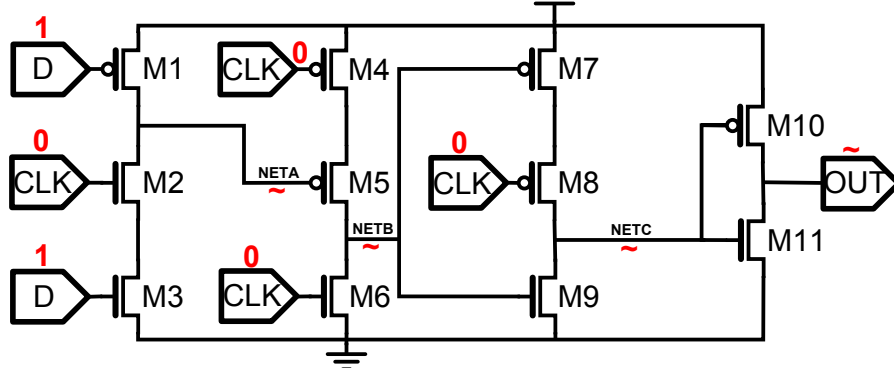


Figure 55: Data=1, CLK=0

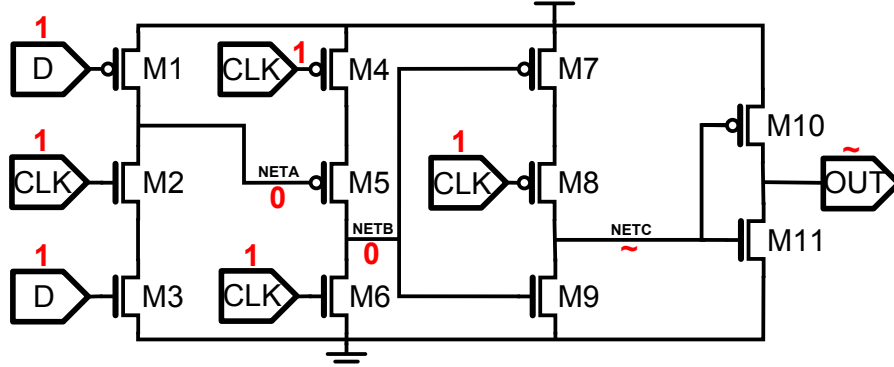


Figure 56: Data=1, CLK=1

In these figures the sign \sim stands for a floating voltage. Nodes specified with this sign are in a condition that the circuit is not forcing any specific voltage value to them. So, due to the parasitic capacitors existing at these nodes, they will reserve the voltage value that they had in the previous state of the circuit.

Now, for example, let's consider the case that the input goes from 0 to 1 while the clock is 1. This flip-flop is a falling edge flip-flop. So, for this transition we expect to see no change at the output. This transition means the change of state from Figure 54 to 56. We can see that in Figure 56 the voltages of the output and NETC are float. So, the voltages of these nodes will be reserved from the previous state, which means the output will be 0 and NETC will be 1. Now, if a falling edge on clock signal happens, we expect the output to be updated from 0 to 1. This transition means the change of state from Figure 56 to 55. We can see that in Figure 55 the voltage of NETA is float. So, it will inherit the voltage of this node in the previous state, i.e., 0. This will cause the transistor M5 to conduct and change the voltage of NETB to 1. Having 1 at NETB, M9 will conduct which will result in the change of NETC to 0 and consequently, output to 1.

7 Appendix B- Comparison between the output of RF-DAC transmitter and DCT

The drawback of the RF-DAC architecture, compared to DCT is that the output will have more high frequency components. So, nice filtering is needed. The reason is that in RF-DAC we are creating pulses as the carrier signal while in DCT it is sinusoid. In figures 57 and 58 the output of a DCT and RF-DAC for the case of one dimensional modulation (i.e., we have just the I signal) is presented.

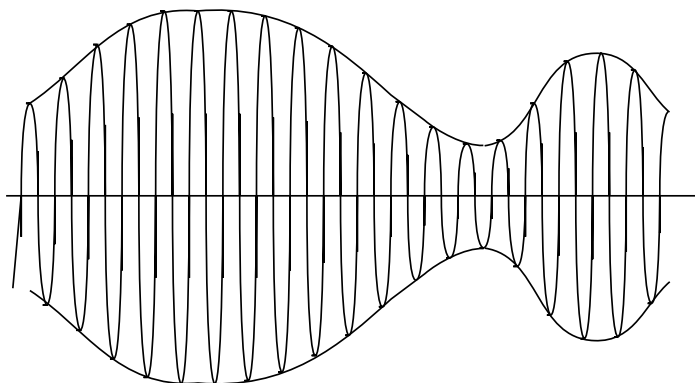


Figure 57: Amplitude modulated signal by DCT

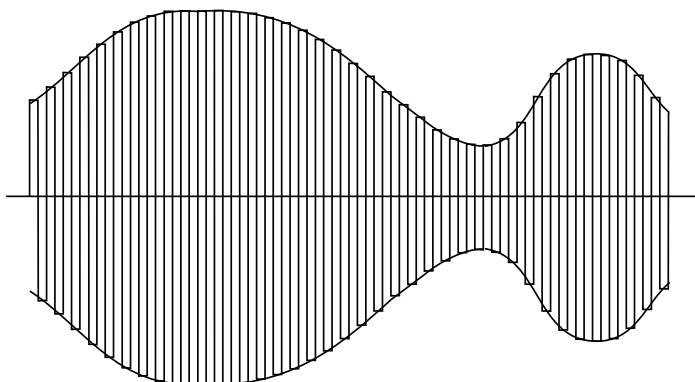


Figure 58: Amplitude modulated signal by RF-DAC

References

- [1] Jerry Lemberg *An Upconverting Digital to Analog Converter* Master thesis at the department of Micro- and Nanotechnology, Electrical Engineering school, Aalto University
- [2] S. Luschas, R. Schreier, and H.-S. Lee *Radio frequency digital-to-analog converter* Solid-State Circuits, IEEE Journal of, vol. 39, no. 9, pp. 1462-1467, sept. 2004.
- [3] Martin Clara *High-Performance D/A-Converters, Application to Digital Transceivers*
- [4] Abhishek Yadav *Digital Communication*
- [5] Maxim Integrated *Application notes, ID NO. 641*
- [6] Yongjian Tang, Hans Hegt, Arthur van Roermund *Dynamic-Mismatch Mapping for Digitally-Assisted DACs*
- [7] Marko Kosunen *Digital signal processing and digital-to-analog converters for wide-band transmitters* Doctoral dissertation, Helsinki university of technology, Electronic circuit design laboratory- Espoo 2006
- [8] Cassius C Cutler *Transmission systems employing quantization* Patent US 2927962 A, 26 April 1954
- [9] John Watkinson *Art of Digital Audio* Analog Devices, Application note 283
- [10] Franco Maloberti *Data Converters*
- [11] George I Bourdopoulos, Aristodemos Pnevmatikakis, Vassilis Anastassopoulos, Theodore L Deliyannis *Delta-Sigma Modulators- Modeling, Design, and Applications* Imperial College Press
- [12] Kyehyung Lee *High Efficiency Delta-sigma Modulation Data Converters*
- [13] Kaveh Hosseini, Michael Peter Kennedy *Minimizing Spurious Tones in Digital Delta-Sigma Modulators*
- [14] *Sigma-Delta ADCs and DACs* Analog Devices, Application note 283
- [15] Behzad Razavi *RF Microelectronics, 2nd edition*
- [16] Alex R. bugeja, member IEEE, and Bang-Sup Song, fellow, IEEE *A self-trimming 14-b 100-MS/s CMOS DAC* IEEE Journal of solid state circuits, VOL. 35, NO. 12, DECEMBER 2000

- [17] John Hyde, member, IEEE, Todd Humes, member, IEEE, Chris Diorio, member, IEEE, Mike Thomas, and Miguel Figueroa *A 300-MS/s 14-bit Digital to analog Converter in Logic CMOS* IEEE Journal of solid state circuits, VOL. 38, NO. 5, MAY 2003
- [18] Yonghua Cong, Randall L. Geiger *A 1.5V 14b 100MS/s self-calibrated DAC* ISSCC 2003/Session 7/DACs and AMPs/Paper 7.2
- [19] Qiuting Huang¹, Pier Andrea Francese¹, Chiara Martelli¹, Jannik Nielsen *A 200MS/s 14b 97mW DAC in 0.18 \hat{I}_4^1 m CMOS* ISSCC 2004 / SESSION 20 / DIGITAL-TO-ANALOG CONVERTERS / 20.3
- [20] K. P. Sunil Rafeeqe and Vinita Vasudevan *A New Technique for On-Chip Error Estimation and Reconfiguration of Current-Steering Digital-to-Analog Converters* IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS-I: REGULAR PAPERS, VOL. 52, NO. 11, NOVEMBER 2005
- [21] Georgi I. Radulov, Patrick J. Quinn, Hans Hegt, and Arthur van Roermund *An on-chip self-calibration method for current mismatch in D/A Converters* Proceedings of ESSCIRC, Grenoble, France, 2005
- [22] Tao Chen, Member, IEEE, and Georges G. E. Gielen, Fellow, IEEE *A 14-bit 200-MHz Current-Steering DAC With Switching-Sequence Post-Adjustment Calibration* IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 42, NO. 11, NOVEMBER 2007
- [23] Yongjian Tang, Joost Briaire, Kostas Doris, Robert van Veldhoven, Pieter C. W. van Beek, Hans Johannes A. Hegt, Senior Member, IEEE, and Arthur H. M. van Roermund, Senior Member, IEEE *A 14 bit 200 MS/s DAC With SFDR 78 dBc, IM3 < -83 dBc and NSD < -163 dBm/Hz Across the Whole Nyquist Band Enabled by Dynamic-Mismatch Mapping* IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 46, NO. 6, JUNE 2011
- [24] Dr. Paul D. Franzon *Latches and Flip-Flops* NC state University
- [25] Jiren Yuan, Christer Svensson *New single-clock CMOS Latches and FlipFlops with Improved Speed and Power Savings* IEEE Journal of Solid State Circuits, VOL. 32, NO. 1, JANUARY 1997