AALBORG UNIVERSITY
Department of Electronic Systems

Low Pass ΔΣ Modulation for Highly Power-Efficient Transmission of Varying Envelope Signals

RISC - Master's Thesis

Rolf Hermansen, Jesper Johnsen & Peter Knudsen

2007

FOR PUBLIC RELEASE

AALBORG UNIVERSITY

Department of Electronic Systems
Fredrik Bajers Vej 7, A1
DK-9220 Aalborg East
Phone.: 9635 8600
Web: http://ies.auc.dk

PROJECT TITLE:
Low Pass $\Delta\Sigma$ Modulation for Highly Power-Efficient Transmission of Varying Envelope Signals

PROJECT TERM:
9th and 10th semester,
Sep. 4th 2006 - June 15th 2007

PROJECT GROUP: 1050

GROUP MEMBERS:
Rolf Thorsen Hermansen
Jesper Skaarup Johnsen
Peter Bloch Knudsen

SUPERVISORS:
Michael Nielsen
Ole Kiel Jensen

PAGES - MAIN REPORT: 91
PAGES - APPENDICES: 69
PUBLICATIONS: 7
FINISHED: June 15th 2007

ABSTRACT:

This master's thesis aims to prove the concept of an integrated transmitter architecture founded upon the principle of $\Delta\Sigma$ modulation. The architecture transforms a varying envelope signal into square waves allowing the successive PA to be driven as a switch, resulting in highly power-efficient transmission.
The design of the architecture is supported by comprehensive analysis on system and circuit level. Using behavioral models of sub-circuits, the impairment caused by non-ideal effects is evaluated, and requirements for each individual sub-circuit are specified. Circuit simulations show that the concept can be realized in an integrated solution.
The transmitter architecture is integrated in a CMOS 0.18 μm technology. Measurements are performed on the fabricated integrated circuits. However, a short-circuit exists, precluding verification of the architecture. Measurements performed on the individual sub-circuits show consistency with the simulation results, indicating that the simulated performance of the transmitter can be realized on chip.
Conclusively, the concept of the integrated transmitter architecture has not been proved, but simulation and measurement results indicate that it is possible.

Preface

This master's thesis is compiled during the period September 2006 to June 2007 and documents the project work of group 1050 during this period. The master's thesis is performed at the 9th and 10th semester of the specialization RF Integrated Systems and Circuits (RISC) at Department of Technology Platforms, Aalborg University Denmark. It is expected that the readers of this report have technical insight in communication systems, RF integrated circuit design and some knowledge concerning CMOS technology.

The target group for this thesis is the project supervisors and external examiner.

The main milestone in the project has been the tapeout in late February, at where the integrated circuit layout was submitted for fabrication. Hence, documentation is to a great extend carried out after tapeout, which deviates from normal project procedure. In addition, the shipment from the manufacturer was delayed, and the chip was received the 23^{rd} of may, reducing the available time for measurements.

Simulations are to some extend used as an integrated part of the design process. The presented simulations are performed in Matlab 7.2.0.232 (R2006a), Agilent Advanced Design System 2005A.400, ADS Momentum, and Cadence Design systems v5.1.41.

A DVD-ROM containing project related material is located in the back cover. In the root of the DVD the file index.html links to the list of all documented ADS simulations.

References to documented simulations performed in Matlab and ADS are made using `[Init ##]` and `[DIR###]` , respectively. Matlab code is found in Appendix H, while ADS refers to the HTML structure found on the DVD.

The chip is fabricated by UMC in their 0.18 μm CMOS technology.

Literature sources are referred to with square brackets [#,#], where the latter indication defines the specific page reference. If the citation is located before a period, it refers to the preceding sentence, and if located after a period, the reference applies to the preceding paragraph.
The first time an abbreviation is written in the thesis, it is written in full length, followed by the acronym in a bracket. The list of acronyms is located on Page 88.

Rolf Hermansen

Jesper Skaarup Johnsen

Peter Bloch Knudsen

Table of contents

1 Introduction

In the present chapter a short introduction to the project is given. The basic idea and motivation behind the project are presented followed by a project scope outlining the areas of interest and aims of the project work. Finally, fundamental theory on the applied concept is presented.

1.1 Motivation

An important sales parameter in any wireless mobile communication device is the talk time, which is the length of time the device may be engaged in transmission before it runs out of battery. A factor of decisive importance when dealing with the talk time, is the efficiency of the power amplifier in the transmitter.

High power efficiency is achieved by operating the power amplifier as a switch, as in the case of class D and E amplifiers. In theory an efficiency of 100 % is achievable. When operating the power amplifier as a switch however, it displays a strong non-linear transfer function, which limits its applicability to signals that are phase modulated. Consequently, less efficient class AB power amplifiers are used in modern wireless communication systems as these employ modulation schemes where both the phase and envelope of the message signal carry information.

One solution to improve the efficiency, is to combine the power amplifier with the concept of $\Delta\Sigma$ modulation, transforming the varying envelope signal into square pulses enabling usage of class D and E power amplifiers without distorting the envelope [Nielsen and Larsen, 2006]. The concept is illustrated in Figure 1.1.

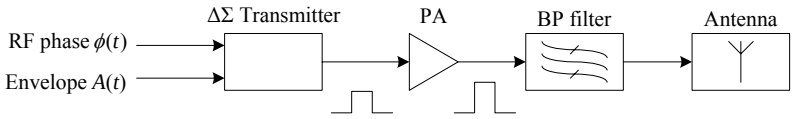


Figure 1.1: *Concept of combining $\Delta\Sigma$ modulation and power amplifier in transmitter.*

The $\Delta\Sigma$ Transmitter shown in Figure 1.1 is based on the concept of $\Delta\Sigma$ modulation and operates as a driver to the power amplifier. A $\Delta\Sigma$ Transmitter is defined as a $\Delta\Sigma$ Modulator that translates the input envelope to an RF frequency, whereas for a $\Delta\Sigma$ Modulator, or a $\Delta\Sigma$ ADC the output signal is located at baseband.

1.2 Project Objective and Scope

The primary objective of the project is to document whether the $\Delta\Sigma$ Transmitter architecture, as a concept, can be realized in an integrated solution.

The objective is approached through analysis, design, implementation and verification of the $\Delta\Sigma$ Transmitter architecture. The analysis is aimed at identifying critical performance issues, utilizing compre-

hensive simulations ranging from system level simulations to circuit simulations. From system level simulations requirements are specified, and the sub-circuits used in the $\Delta\Sigma$ Transmitter are separately designed. The sub-circuits are interconnected and the $\Delta\Sigma$ Transmitter is designed. Finally, the $\Delta\Sigma$ Transmitter is integrated in the 0.18 μm Complementary Metal-Oxide Semiconductor (CMOS) technology, and measurements are performed on the integrated circuit in order to verify its functionality.

With the objective of proving the concept of the $\Delta\Sigma$ Transmitter, focus is limited to transmission of a WLAN signal. WLAN signals carry information in both envelope and phase, and the standard is integrated in a vast variety of mobile devices. Hence, simulations are performed applying a WLAN signal, and the performance is evaluated mainly with respect to complying with the transmission mask specified in the IEEE 802.11g standard [IEEE, 802.11g].

The $\Delta\Sigma$ Transmitter is based on the fundamental theory of $\Delta\Sigma$ modulation, why an introduction to the topic is presented in the following.

1.3 $\Delta\Sigma$ Modulation

$\Delta\Sigma$ modulation is often used in digital signal processing systems, where it is suited for especially A/D conversion. For this reason it is often referred to as a $\Delta\Sigma$ ADC[1]. The modulator utilizes oversampling of the input signal combined with noise shaping. The advantage of the $\Delta\Sigma$ Modulator compared to unshaped modulators or A/D converters is a significant improvement in in-band Signal-to-Noise Ratio (SNR). This improvement in SNR is illustrated and described later. Initially the operational principle of the $\Delta\Sigma$ Modulator is presented.

1.3.1 Operational Principle

The operational principle of the $\Delta\Sigma$ Modulator is described based on the first order architecture illustrated in Figure $1.2a$.

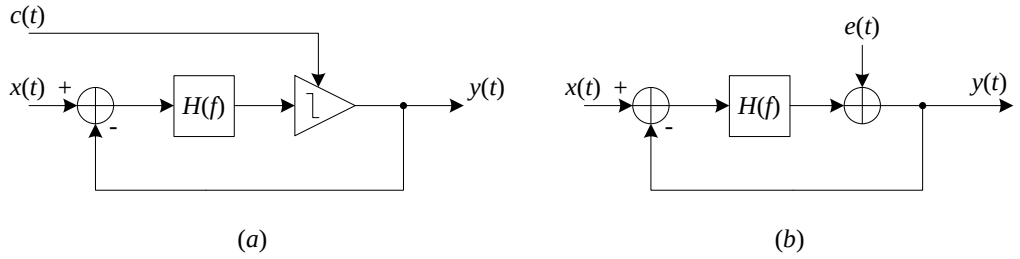


Figure 1.2: *Principle of the $\Delta\Sigma$ Modulator. (a) First order $\Delta\Sigma$ Modulator consisting of a loop filter and a 1 bit clocked comparator. (b) Linearized model assuming ideal clocked comparator represented by an additive noise source $e(t)$. [Cherry and Snelgrove, 2002, p. 2]*

The $\Delta\Sigma$ Modulator performs a quantization, controlled by $c(t)$, of the input signal $x(t)$, utilizing a 1 bit clocked comparator. As for any quantization process, noise is introduced at the output $y(t)$. Using negative feedback, the modulator continuously tries to equalize the difference between the output and the input, and hereby equalizing the quantization error over time. When utilizing a Low Pass (LP) filter in the forward loop, the modulator is able to shape the noise away from the signal band of interest, which increases the SNR.

[1] In the present project no A/D conversion is made, why the term $\Delta\Sigma$ Modulator is applied.

Linearized Model

In order to obtain an analytical description of the functionality, a linearized model of the ΔΣ Modulator is presented in Figure 1.2*b*. The model enables calculations of the modulator performance, by representing the comparator by an additive noise source $e(t)$. The quantization error $e(t)$ is assumed to be a random white noise process independent from the input signal. For these assumptions to be valid the noise sequence must be uncorrelated from sample to sample and the comparator must remain in the non-overload region, meaning that the input amplitude must not exceed the high output potential of the comparator. In a practical system, these conditions are not strictly valid, but yield good results. [Geerts et al., 2002, p. 12]

The linearized model contains two inputs; the input signal $x(t)$ and the quantization error $e(t)$. The output signal in the Z-domain $Y(z)$ is given by [Cherry and Snelgrove, 2002, p. 2]

$$Y(z) = \mathrm{STF}(z) \cdot X(z) + \mathrm{NTF}(z) \cdot E(z) \tag{1.1}$$

where $\mathrm{STF}(z)$ and $\mathrm{NTF}(z)$ are the signal and noise transfer functions, respectively. Applying basic feedback theory, the signal and noise transfer functions are given by

$$\mathrm{STF}(z) = \left.\frac{Y(z)}{X(z)}\right|_{E(z)=0} = \frac{H(z)}{1+H(z)} \tag{1.2}$$

$$\mathrm{NTF}(z) = \left.\frac{Y(z)}{E(z)}\right|_{X(z)=0} = \frac{1}{1+H(z)} \tag{1.3}$$

From the expressions for $\mathrm{STF}(z)$ and $\mathrm{NTF}(z)$, the principle of noise shaping is seen. In the frequency range where the gain of the loop filter is much greater than 1, the signal and noise transfer functions are given by

$$\mathrm{STF}(z) \approx 1 \tag{1.4}$$

$$\mathrm{NTF}(z) \approx \frac{1}{H(z)} \ll 1 \tag{1.5}$$

which show that the gain of the signal transfer function is close to 1, while the gain of the noise transfer function equals the reciprocal value of the filter gain in the given frequency range. The principle is illustrated in Figure 1.3.

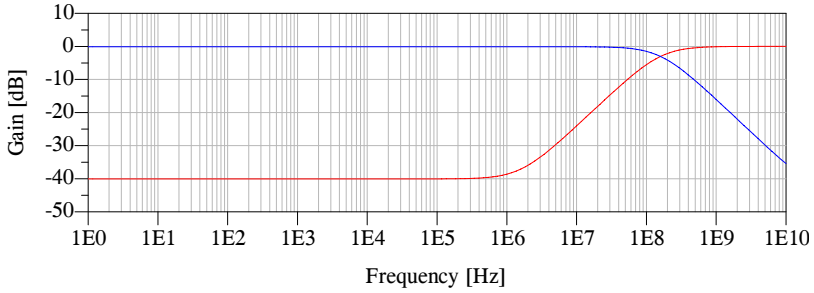


Figure 1.3: *STF (blue) and NTF (red) for a first order low pass ΔΣ Modulator, when applying the model of Figure 1.2b. The filter gain is set to* 40 dB *and cutoff frequency to* 1.6 MHz. `[DIR023]`

Within the frequency band the input signal is passed directly to the output. The white quantization noise is shaped and suppressed by the inverse characteristic of the loop filter.

The loop filter can be designed to have different filter characteristics. Low pass and band pass characteristics are commonly used [Geerts et al., 2002, p. 21].

1.3.2 Improvement in SNR

In order to illustrate how the $\Delta\Sigma$ Modulator performs in comparison with Nyquist rate modulation and oversampled modulation, the relative PSDs of the three modulation types are illustrated in Figure 1.4.

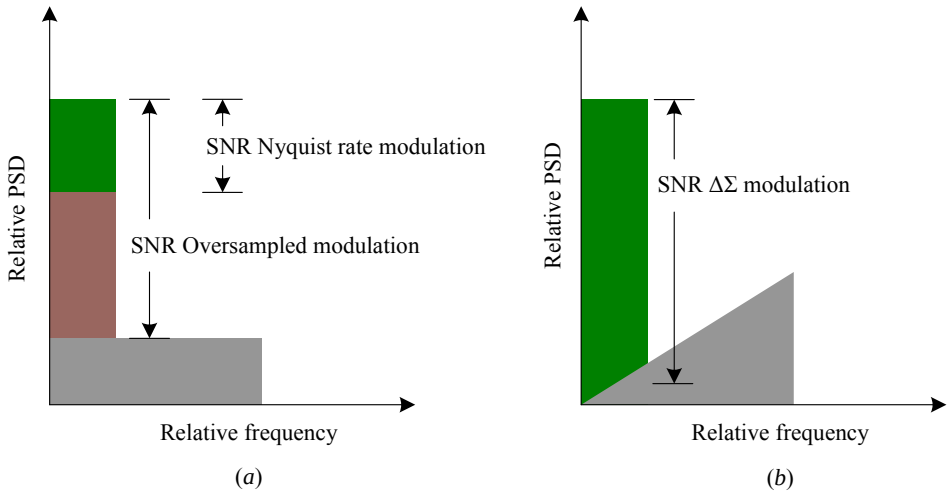


Figure 1.4: SNR *for three different types of modulation. (a) Nyquist rate modulation and oversampled modulation. (b) Low pass* $\Delta\Sigma$ *modulation. [Pervez Aziz, 1996, pp. 64-65].*

The figure depicts the power spectral density of the quantization noise and the input signal given three different modulation types. In the Nyquist rate modulation in Figure 1.4*a* all noise is produced in-band, while the oversampled modulator reduces the noise in the desired signal bandwidth (BW) by spreading the quantization noise power uniformly in a bandwidth from $-f_s/2$ to $f_s/2$, where f_s is the sampling frequency, resulting in increased SNR. Figure 1.4*b*, in which $\Delta\Sigma$ modulation is applied, illustrates how the uniformly distributed noise is shaped such that only a relatively small fraction of the total noise power falls in the signal bandwidth. The increased out-of-band noise is attenuated by a low pass filter.

The Nyquist rate f_N is defined as the minimum sampling frequency required to avoid aliasing, which is

$$f_N = 2f_b \tag{1.6}$$

where f_b is the BW of the sampled signal. The oversampling ratio (OSR) is defined as the ratio between the actual sampling frequency f_s and the Nyquist rate:

$$\text{OSR} = \frac{f_s}{f_N} = \frac{f_s}{2f_b} \tag{1.7}$$

The relations between the number of bits B in the quantizer, OSR, and the maximum achievable Signal-to-Noise ratio SNR_{max} for the three types of modulation are given in the following:

Nyquist rate modulation [Geerts et al., 2002, p. 14, Eqn. 2.8]:

$$\text{SNR}_{max} = B \cdot 6.02 + 1.76 \text{ [dB]} \tag{1.8}$$

Oversampled modulation [Geerts et al., 2002, p. 18, Eqn. 2.13]:

$$\text{SNR}_{max} = B \cdot 6.02 + 1.76 + 10\log(\text{OSR}) \text{ [dB]} \tag{1.9}$$

$\Delta\Sigma$ modulation utilizing integrators [Geerts et al., 2002, p. 27, Eqn. 2.33]:

$$\text{SNR}_{max} = \frac{3\pi}{2} \cdot (2^B - 1)^2 \cdot (2n+1) \cdot \left(\frac{\text{OSR}}{\pi}\right)^{2n+1} [\cdot] \tag{1.10}$$

where n in (1.10) denotes the order of the modulator. A higher oversampling ratio distributes the noise in a larger frequency band, resulting in less noise in the signal band. The higher order modulators shape the noise such that more noise is moved outside the signal band and hence reducing noise in the frequency band of interest. When the number of bits in the quantizer is increased, the quantization noise is reduced as the quantizer output is able to track the input signal much closer. [Geerts et al., 2002, p. 22]

Based on the presented theory concerning $\Delta\Sigma$ modulation, the principle of the $\Delta\Sigma$ Transmitter is presented in the following chapter.

2 $\Delta\Sigma$ Transmitter Architecture

This chapter describes the basic concept of the employed transmitter architecture. The chapter serves the purpose of describing the ideal functionality of the architecture on a system level.

2.1 Circuit Operation

As presented in the introduction the basic idea is to design a transmitter architecture that enables the use of a more efficient power amplifier. The architecture must modulate a signal that carries information in both the envelope and phase using square pulses.

This is realized by employing a $\Delta\Sigma$ Transmitter architecture consisting of a first order low pass $\Delta\Sigma$ Modulator, a delay element, and an AND-gate, where the $\Delta\Sigma$ Modulator is comprised of a low pass filter and a one-bit clocked comparator. The transmitter architecture is depicted in Figure 2.1.

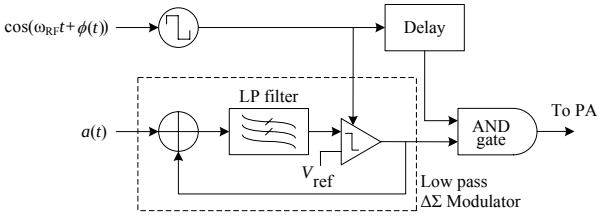


Figure 2.1: *$\Delta\Sigma$ Transmitter architecture [Nielsen and Larsen, 2006].*

The envelope $a(t)$ of the message signal is quantized into ones and zeros by the $\Delta\Sigma$ Modulator, whereby the envelope is represented by the average value of the ones and zeros. Both the clocked comparator and the AND-gate are controlled by a phase modulated clock, where the phase of the message signal $\phi(t)$ is added to the phase of the radio frequency (RF) carrier. When the $\Delta\Sigma$ Modulator output is AND'ed with a delayed version of the phase modulated clock, the phase information is passed to the output of the $\Delta\Sigma$ Transmitter, because the timing of the output pulses is controlled by the phase of the input message signal.

The delay element ensures that the output signal is always low for at least half a RF period, meaning that the output from the $\Delta\Sigma$ Modulator is translated to the RF carrier.

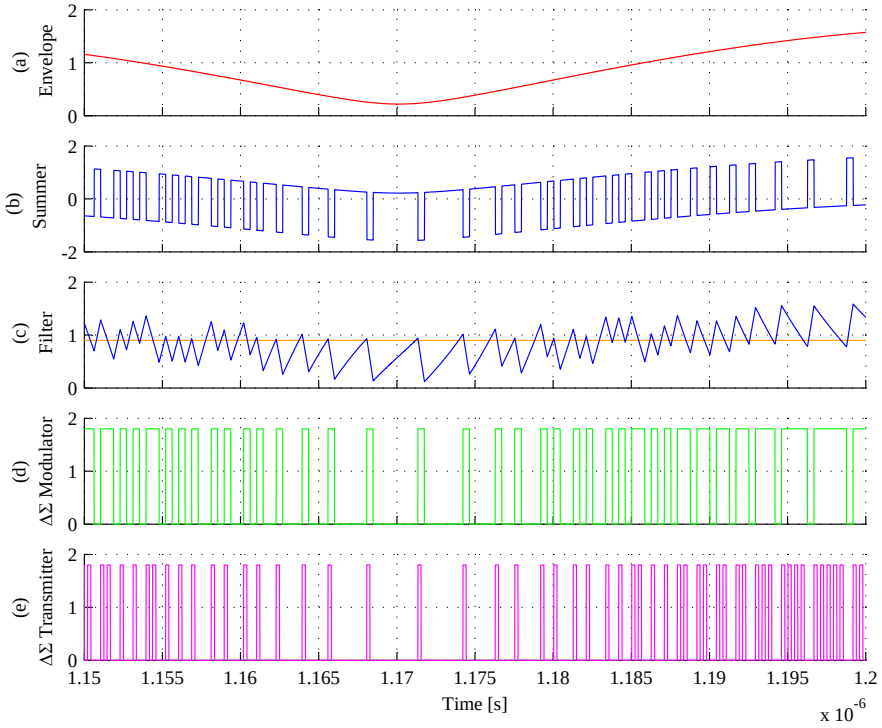


Figure 2.2: *Time domain representation of signals in the* $\Delta\Sigma$ *Transmitter. Duration is* 312.5 ns *and the sampling frequency is* 300 MHz. `[Init 3]`

Figure 2.2a illustrates a varying envelope signal in the time domain, applied to the input of the $\Delta\Sigma$ Transmitter. The summer output b, is the input signal where the output of the $\Delta\Sigma$ Modulator d is subtracted, which eventually means that the summer output presents the error introduced by quantization.

The filter output signal c is an accumulation of the error where the incline is controlled by the feedback pulses. The filter output is represented by either a low or high voltage at the output of the clocked comparator. Due to a sampling frequency much higher than the bandwidth of the input signal, the $\Delta\Sigma$ loop is able to constantly equalize the difference between the input a and the output d. This shapes the quantization noise away from the message signal, increasing the SNR of the modulator output.

Finally, the output of the $\Delta\Sigma$ Modulator d is AND'ed with the phase modulated clock, forming the actual output e, composed of a series of uniform square pulses. Comparing a with e reveals a correlation between the input signal level and the density of the output pulses.

To illustrate the effect from noise shaping a 2.4 GHz clock signal and a WLAN signal with a 16.6 MHz bandwidth are applied to the $\Delta\Sigma$ Transmitter. The output spectra of the $\Delta\Sigma$ Modulator and the $\Delta\Sigma$ Transmitter are depicted in Figure 2.3.

Figure 2.3a clearly shows how the noise is shaped away from the information band at the output of the $\Delta\Sigma$ Modulator. According to Equation (1.10) the obtained in-band SNR for a first order, one bit $\Delta\Sigma$ Modulator with an OSR $= {}^{2.4\text{ GHz}}/_{33.2\text{ MHz}} = 72.3$, equals

$$\text{SNR} = \frac{3\pi}{2} \cdot (2-1)^2 \cdot (2+1) \cdot \left(\frac{72.3}{\pi}\right)^{2+1} = 52.4 \text{ dB} \tag{2.1}$$

which is consistent with the simulated behavior. When expressing SNR in the following, it is done in reference to noise power close to the in-band signal.

Figure 2.3b illustrates how the signal is moved to the RF frequency as a result of the processing in the AND-gate. At the RF frequency an SNR of approximately 44 dB is obtained.

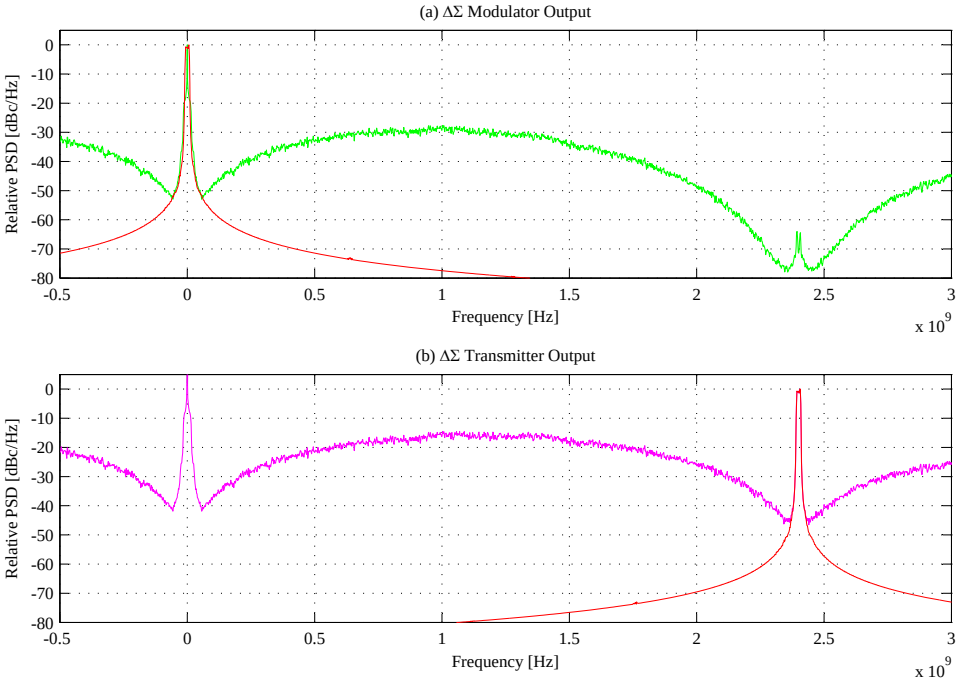


Figure 2.3: *Relative power spectral density. (a) Output of ΔΣ Modulator (green) and input signal (red). (b) Output of ΔΣ Transmitter (magenta) and shifted version of input signal (red).* `[Init 2]`

2.2 Mathematical Operation

For a mathematical analysis of the ΔΣ Transmitter architecture, it is appropriate to express the output signal from the ΔΣ Modulator as [Nielsen and Larsen, 2006]

$$s_{\Delta\Sigma}(t) = a(t) + q(t) \tag{2.2}$$

where $a(t)$ is the normalized envelope and $q(t)$ is the shaped quantization noise at the output of the ΔΣ Transmitter. The squared clock signal $s_{\mathrm{CLK}}(t)$ is expressed as a trigonometric Fourier series given by [Spiegel and Liu, 1999, p. 142, eq. 24.7]

$$s_{\mathrm{CLK}}(t) = \frac{1}{2} + \frac{2}{\pi} \sum_{n=1,3,\ldots}^{\infty} \frac{1}{n} \sin\left[n(\omega_0 t + \phi(t))\right] \tag{2.3}$$

in which the phase information $\phi(t)$ results in non-constant duty cycle of the clock.

The processing in the AND-gate corresponds to a multiplication of the quantized envelope $s_{\Delta\Sigma}(t)$ and the clock signal $s_{\mathrm{CLK}}(t)$. That is

$$s_{\mathrm{o}}(t) = s_{\Delta\Sigma}(t) \cdot s_{\mathrm{CLK}}(t) = \frac{1}{2}a(t) + \frac{1}{2}q(t) + \frac{2}{\pi}\left[a(t) + q(t)\right] \cdot \sum_{n=1,3,\ldots}^{\infty} \frac{1}{n} \sin\left[n(\omega_0 t + \phi(t))\right] \tag{2.4}$$

In the frequency domain this corresponds to a convolution that translates the quantized envelope signal to all the odd harmonics of the RF clock frequency while at the same time adding phase information.

From (2.4) it is identified that the output around DC and the odd harmonics is given by

$$s_{\mathrm{o,DC}}(t) = \frac{1}{2}a(t) + \frac{1}{2}q(t) \tag{2.5}$$

$$s_{\mathrm{o},f_0}(t) = \frac{2a(t)}{\pi}\sin(2\pi f_0 t + \phi(t)) + \frac{2q(t)}{\pi}\sin(2\pi f_0 t + \phi(t)) \tag{2.6}$$

$$s_{\mathrm{o},3f_0}(t) = \frac{2a(t)}{3\pi}\sin(6\pi f_0 t + 3\phi(t)) + \frac{2q(t)}{3\pi}\sin(6\pi f_0 t + 3\phi(t)) \tag{2.7}$$

and so on. This means that the correct modulated signal plus the quantization is found around f_0. This is depicted in Figure 2.4.

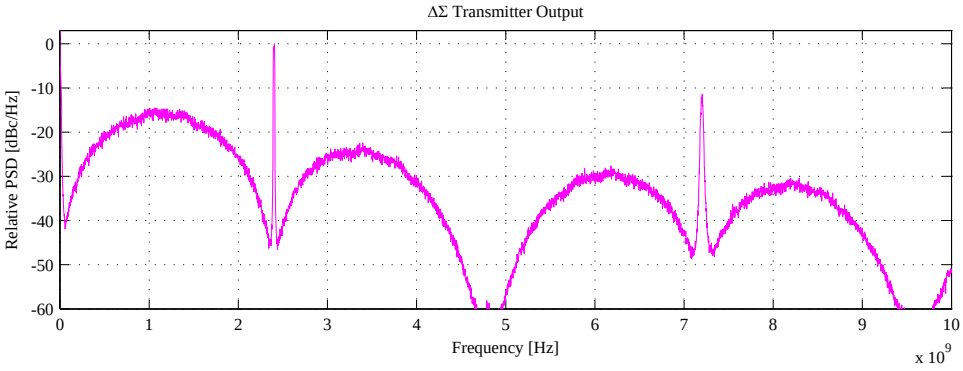


Figure 2.4: *ΔΣ Transmitter output PSD.* `[Init 4]`

As the quantization noise is shaped away from the information signal, the noise increases from DC to half f_0 and then falls until f_0. At half f_0 two significant noise contributors exist; that from DC and that from f_0. At $3f_0/2$ only the noise contribution from f_0 is significant. This explains why the quantization noise is not symmetrical around the carrier.

2.3 System Response

As a verification of correct mapping of envelope and phase from the input to the output of the ΔΣ Transmitter, an ideal simulation is performed in Matlab. A WLAN signal is applied, and the ΔΣ Transmitter output signal is direct down-converted and filtered with an ideal 25 MHz brick wall low pass filter. The time domain envelope and phase response is illustrated in Figure 2.5.

A compensation is made for constant error in both time, envelope and phase. Details on the procedure is found in Appendix D. The figure shows that both the envelope and phase at the output of the ΔΣ Transmitter are almost coinciding with their corresponding input.

Based on the presented ΔΣ Transmitter concept a basis for the following design phase is established. The design is approached via system level simulations, as described in the following chapter.

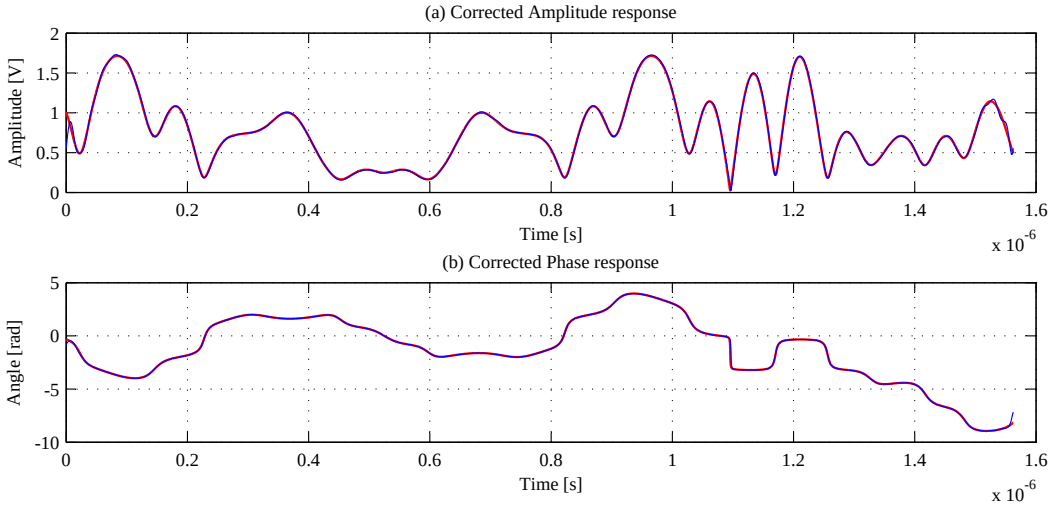


Figure 2.5: *Time domain envelope and phase response. Red: Reference signal. Blue: $\Delta\Sigma$ Transmitter output signal.* `[Init 5]`

System Level Simulation 3

The present chapter describes how the $\Delta\Sigma$ Transmitter architecture is implemented on a system level in Matlab. The implementation of each sub-circuit of the architecture is presented along with non-ideal effects that are expected to degrade system performance. An ideal simulation is performed, which is used for reference when evaluating the influence of non-ideal effects. This is followed by an investigation of how the system behaves, when the different non-ideal effects are applied.

3.1 Simulation Models

The design is based on both system level simulations and spice simulations on transistor level. The system level simulations are primarily performed in Matlab using behavioral models of each sub-circuit of the $\Delta\Sigma$ Transmitter architecture. The purpose of applying behavioral models is to model the behavior of each component adequately allowing for a fast analysis of the $\Delta\Sigma$ Transmitter. The model should have proper complexity to emphasize the critical parameters, but still be simple in order to minimize the simulation time.

Keeping the models simple is especially important when performing simulations of the $\Delta\Sigma$ Transmitter. The performance of the architecture depends on precise predictions of the transients in the clock signal. Since the clock is phase modulated, the duration of the RF clock period is non-constant, which makes true RF simulations a necessity. This makes a simulation of even a short time period, a time consuming affair.

In the following it is introduced how simulation models of each sub-circuit of the transmitter are implemented in Matlab. The implemented behavioral models include the following effects:

- Slew rate in the filter, clocked comparator, and AND-gate.
- Unity Gain Bandwidth (UGBW) of the operational amplifier (opamp) in the filter.
- Hysteresis in the clocked comparator.
- Delay in the clocked comparator.

The literature states that these effects are of significant importance in relation to the overall system performance when applying the concept of $\Delta\Sigma$ modulation [Cherry and Snelgrove, 2002, Chap. 3-4]. At the same time their complexity allow for a fairly straight forward implementation in Matlab.

The overall flow of the implemented Matlab simulation is illustrated in Figure 3.1.

The figure illustrates how the Matlab scripts are arranged. Each Matlab simulation documented in this thesis is associated with its own initialization file. The initialization file holds the setup for a specific simulation. The setup involves among other things, the type of signal to process, the non-ideal effects to include, and whether a specific parameter should be swept or not. The Matlab source code is found in Appendix H that also includes a list of the used initialization files.

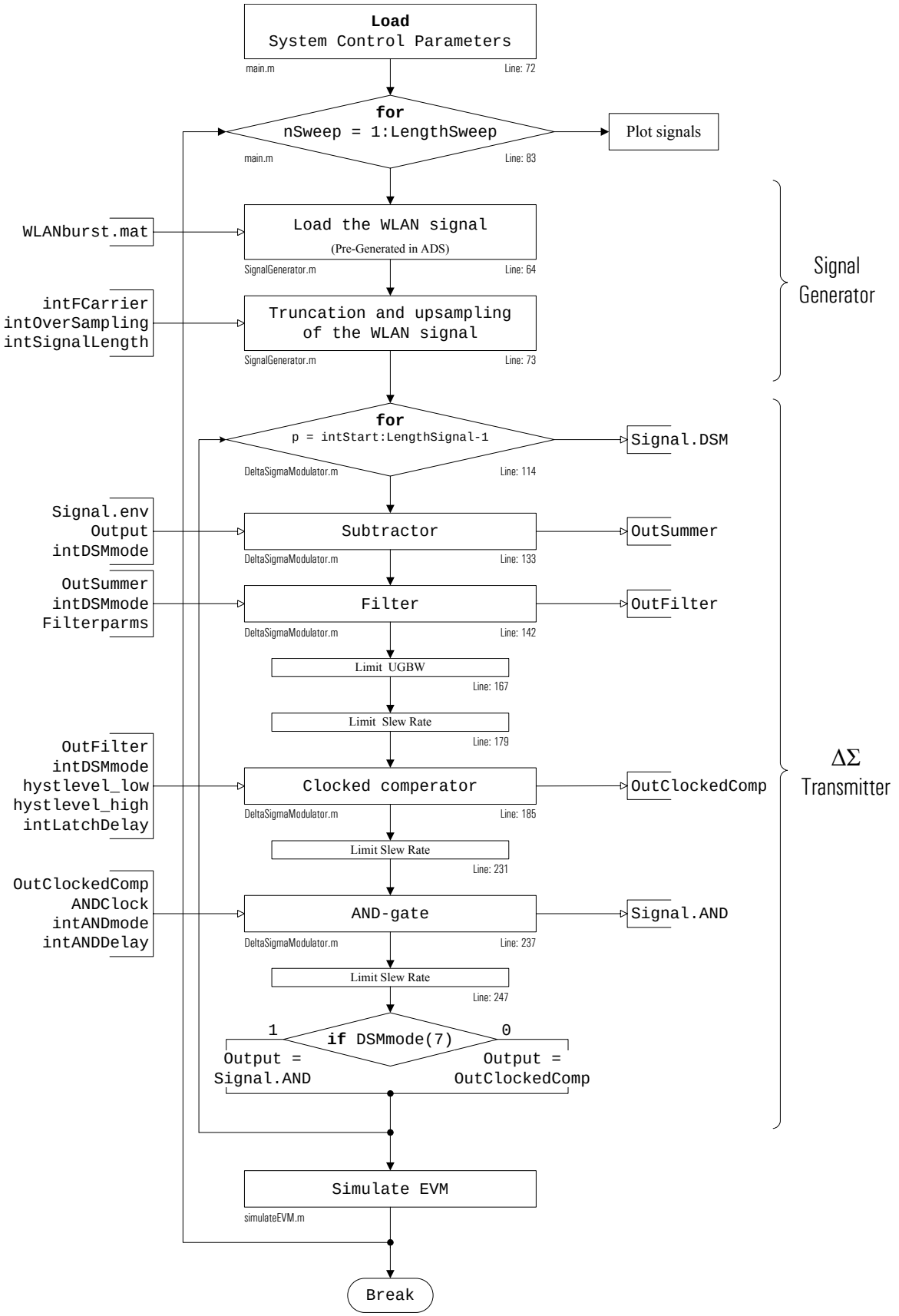


Figure 3.1: *Matlab system level implementation of the $\Delta\Sigma$ Transmitter.*

In the following an introduction to the script that generates both envelope and clock signals to be processed by the $\Delta\Sigma$ Transmitter, is given.

3.1.1 Signal Generator

Generation of the envelope and clock signals, used as system inputs, is carried out in the Matlab script `SignalGenerator.m`. The script offers the possibility to use either a sequence of random bits modulated using 16-QAM or a pre-generated OFDM modulated WLAN burst. The signal to use is specified during initialization.

Three additional parameters are used when generating the input signals; the desired carrier frequency, oversample rate, and signal length. The carrier frequency, is the frequency of the non-modulated RF clock. The oversample rate is defined as the number of samples in one RF clock period, and the signal length indicates the number of WLAN samples that are included in the simulation.

True RF oversampling and up-sampling

True RF oversampling is necessary when simulating the $\Delta\Sigma$ Transmitter, since the timing of the output pulses contains the phase information of the message signal. To visualize this issue, Figure 3.2 is presented.

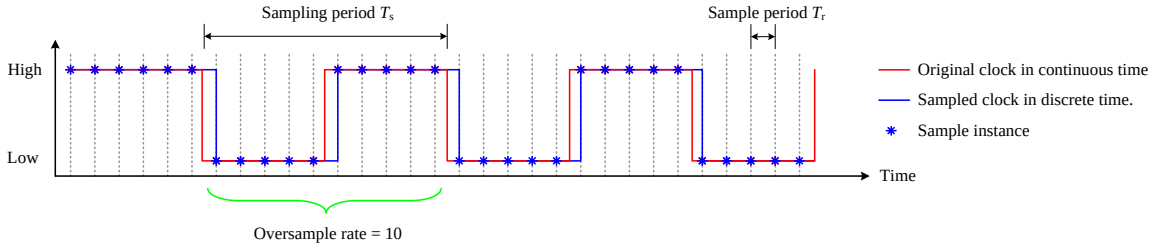


Figure 3.2: *The error introduced to the phase modulated clock due to sampling.*

The figure illustrates how the phase modulated clock (red) is sampled resulting in the discrete time clock signal (blue). Due to phase modulation, the period of the clock T_s is non-constant, why transients cannot be expressed as an integer multiple of the sample period T_r. This eventually introduces noise since the transients of the continuous time clock is different from the transients of the sampled clock. The noise caused by insufficient oversample rate must be kept at a level lower than that caused by non-ideal effects in various system blocks, in order to perform a correct evaluation of system performance based on simulations.

By increasing the oversample rate, the noise is reduced, but the downside is a proportional increase in computation time. When an appropriate oversample rate is chosen, the pre-generated WLAN signal must be up-sampled to the sample rate of the RF clock. Matlab performs the up-sampling based on the specified carrier frequency and oversample rate using interpolation and LP filtering.

3.1.2 Subtractor

Apart from a few initialization procedures, the subtractor is the first part of the larger $\Delta\Sigma$ Transmitter script `DeltaSigmaModulator.m`. The subtractor is realized by subtracting the previous output

sample $x_2[n-1]$ of the $\Delta\Sigma$M loop from the present input sample $x_1[n]$. That is,

$$y[n] = x_1[n] - x_2[n-1]. \tag{3.1}$$

This realizes negative feedback in the loop. No non-ideal effects are included for the subtractor. The subtractor is found on line 133.

3.1.3 Low Pass Filter

A simple integrator or a low pass filter of either first or second order is applied for filtering.

Integrator

The integrator is realized by an accumulation where the present input sample $x[n]$ is added to the previous output sample $y[n-1]$:

$$y[n] = y[n-1] + x[n]. \tag{3.2}$$

A block diagram representation of the integrator is given in Figure 3.3.

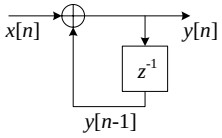


Figure 3.3: *Block diagram representation of the integrator.*

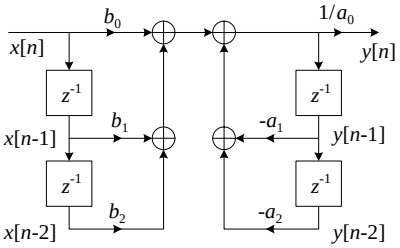


Figure 3.4: *Block diagram representation of second order difference equation.*

The integrator is found on line 145.

First and Second Order Filter

The filter is realized using the predefined Matlab functions. During initialization, the filter parameters such as order, gain, and 3 dB cutoff frequency are specified. Given the order and cutoff frequency the Matlab function `butter` returns the coefficients for a low pass Butterworth filter. Incorporating the gain, these coefficients are then passed as arguments to the function `impinvar` which returns the coefficients of a digital filter with the same impulse response as that of the analog filter. With the obtained filter coefficients a_k and b_k, the input signal $x[n]$ is filtered sample-wise using the standard difference equation [Oppenheim and Schafer, 1999, p. 37]:

$$y[n] = \sum_{k=0}^{M} \frac{b_\mathrm{k}}{a_0} x[n-k] - \sum_{k=1}^{N} \frac{a_\mathrm{k}}{a_0} y[n-k], \tag{3.3}$$

A block diagram representation is given in Figure 3.4.

The slew rate at the output of the filter is limited by using the function `LimitedSlewRate.m`. The filter is found from line 150.

Slew Rate

In a circuit implementation the driving capability of any device is limited and consequently the slew rate is limited. This effect is implemented on system level using the function `LimitedSlewRate.m`. The concept of limited slew rate is illustrated in Figure 3.5.

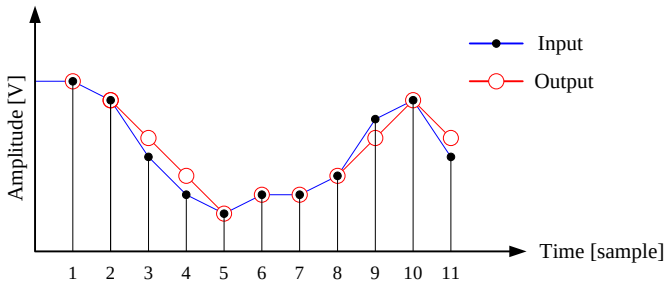


Figure 3.5: *Functionality of the function* `LimitedSlewRate.m`*. Input samples* $3, 4, 9$*, and* 11 *are modified due to slew rate limitations.*

The function compares its input sample to its previous output sample. If the input changes more rapidly than the predefined slew rate, the $n+1$'th output sample is limited to the value of the n'th output sample plus the maximum allowed increment or decrement per sample. When the change in the input signal is less than the predefined slew rate, the output follows the function input. The allowed slew rate of each block in the architecture is defined in the initialization file.

Unity Gain Bandwidth

When implementing the low pass filter, an opamp is used as active element. The amplitude response of the opamp is limited by its UGBW, defined as the frequency at which the gain of the opamp is unity. The consequence of a limited UGBW is that an additional pole is introduced in the filter response. The limited UGBW is included in the system level simulation as an additional low pass filter as illustrated in Figure 3.6.

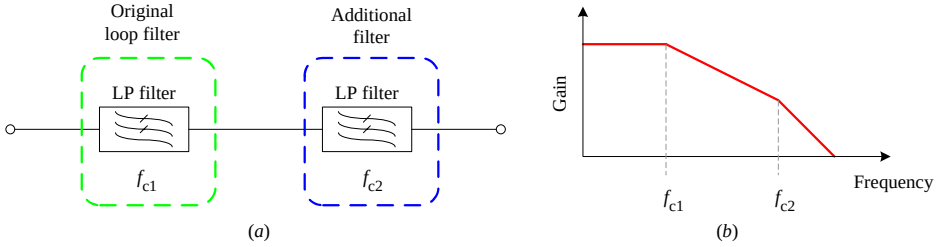


Figure 3.6: *(a) Original filter and filter with limited UGBW. (b) Corresponding amplitude response.* `[DIR023]`

To add the pole caused by the limited UGBW an additional filter is added in cascade with the loop filter as depicted in Figure 3.6*a*. The filter has a gain of unity and a cutoff frequency equal to the applied UGBW. The cascade of the two low pass filters realizes the amplitude response shown in 3.6*b*. Limited UGBW is found on line 167.

3.1.4 Clocked Comparator

The clocked comparator is realized using conditional statements. The outer condition detects any rising edge in the clock signal by comparing two adjacent samples. If $n+1$'th sample equals V_{DD} and the

n'th sample equals zero, a rising edge is present, and the next condition checks whether the input signal is below or above $V_{DD}/2$. The output sample is then assigned a low or high value accordingly. If the outer condition is not satisfied, the $n+1$'th output sample is assigned the same value as the n'th output sample.

Hysteresis

If the clocked comparator suffers from hysteresis the overall system performance is expected to degrade. This fact makes it relevant to include the effect of hysteresis in a system level simulation. The comparator exhibits hysteresis if two different input threshold voltages apply for prompting a change in output voltage. One threshold voltage V_{TH} applies when the output voltage is low and a lower threshold voltage V_{TL} applies when the output voltage is high, as illustrated in Figure 3.7.

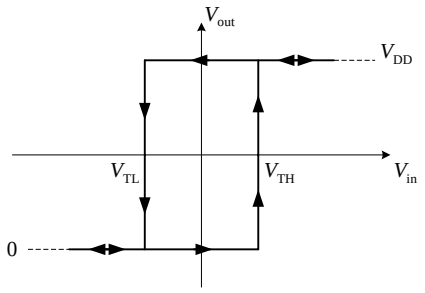


Figure 3.7: *Principle of hysteresis in the comparator.*

The hysteresis degrades the dynamics of the system, because the output of the filter has to accumulate for a longer period of time before the threshold is reached.

Comparator hysteresis is implemented using two different threshold values. On the condition that a rising edge is present in the clock signal, it is settled whether the present output at time n is high or low. In the first case the threshold is set to V_{TL} and in the second case V_{TH}. Then, if the input at time $n+1$ exceeds the given threshold value, the $n+1$'th output sample is set high, else low. The values of V_{TL} and V_{TH} are specified in the initialization file.

Delay

Ideally, if a rising edge in the clock signal is present between samples n and $n+1$, a change is possible in the output signal, also between the two samples n and $n+1$. However, in a circuit implementation, such an instant reaction time between a change in the clock signal and a change in output signal is not possible. A delay is introduced, which degrades the performance of the $\Delta\Sigma$ Transmitter.

In the simulation, the delay is implemented during the assignment of a high or low value to an output sample. If an output change is prompted by samples n and $n+1$ in the clock signal, a corresponding change is made from the $n+1+d$'th output sample, where d represents the delay specified in terms of seconds in the initialization file. The Matlab script `SignalGenerator.m` automatically compensates for the delay in the clocked comparator.

Besides hysteresis and delay, limited slew rate can be included as a non-ideal effect using the function `LimitedSlewRate.m`. The clocked comparator is found on line 185.

3.1.5 AND-gate

The AND-gate functionality is simulated using a conditional statement. If two corresponding input samples are above 0.9 V the corresponding output sample equals V_{DD}. Else, the output sample equals zero. As for the clocked comparator, a delay can be specified for the AND-gate.

Limited slew rate can be included as a non-ideal effect using the function `LimitedSlewRate.m`. The decision is made in the initialization file. The AND-gate is found on line 237.

3.2 Simulation of Ideal $\Delta\Sigma$ Transmitter

The purpose of the following is to present the performance of the ideal $\Delta\Sigma$ Transmitter implemented in Matlab. The ideal performance is used as a reference when evaluating signal quality and thereby system performance in relation to non-ideal effects and/or performance critical parameters. Evaluation is made in terms of SNR of the output PSD and Error Vector Magnitude (EVM). Initially, the setup for the ideal simulation is described followed by a discussion of simulation results.

3.2.1 Simulation Setup

The setup for the ideal simulation includes an ideal first order low pass filter, an ideal clocked comparator, an ideal AND-gate, and a perfect receiver to determine the EVM of the output from the $\Delta\Sigma$ Transmitter.

Oversample Rate

As described in Section 3.1.1, the up-sampling of the time domain signals is decisive for the noise density level in the output spectrum. In connection with the ideal reference simulation it is established to what extend up-sampling is needed for minimizing its influence. For this purpose a sweep of the oversample rate is performed, while considering the output spectrum, as illustrated in Figure 3.8.

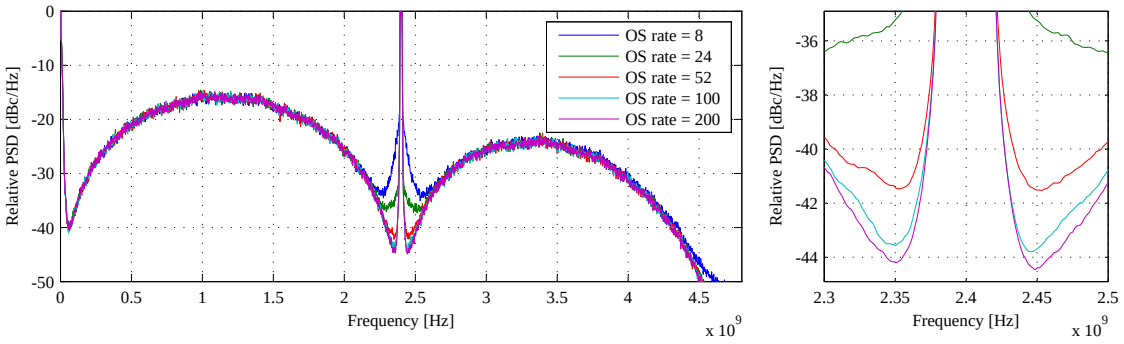


Figure 3.8: *PSD as a function of oversample rate.* `[Init 6]`

From the PSD in Figure 3.8 it is evident that an insufficient sample rate adds spectral noise. As the oversample rate is gradually increased, the noise level decreases near the carrier frequency. When reaching an oversample rate of 100 the change in the spectrum tends to level out. However, even at high oversample rates the signal band is widened.

Another mean of verifying signal quality is EVM. In the present case EVM is not calculated symbol-wise. Instead, any output sample that differs from the corresponding input sample, results in an error.

Consequently, errors will occur in an ideal simulation even though the quantization noise is suppressed sufficiently. For this reason the calculated EVM is not used to compare with a requirement specified in any standard. Details on how EVM is calculated is given in Appendix D.

Figure 3.9 shows how the degree of up-sampling affects the simulated system performance in terms of EVM.

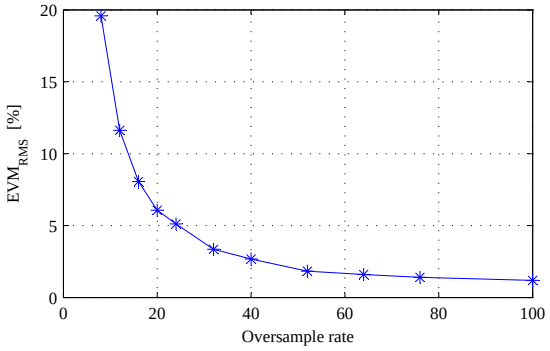


Figure 3.9: *EVM as a function of oversample rate.* `[Init 7]`

The result agrees with that of Figure 3.8. Due to the heavy computation procedure of determining the correlation between input and output, as described in Appendix D, an EVM simulation of an oversample rate of 200 is not possible. As a compromise between computation time and system performance an oversample rate of 100 is applied for successive simulations.

Filter Parameters

The two design parameters filter gain and cutoff frequency affect the system performance. In an ideal simulation setup these two parameters are swept one by one and the resulting output spectrum is used to decide which values to apply. For both simulations an oversample rate of 100 is applied. The results are presented in Figure 3.10 and 3.11.

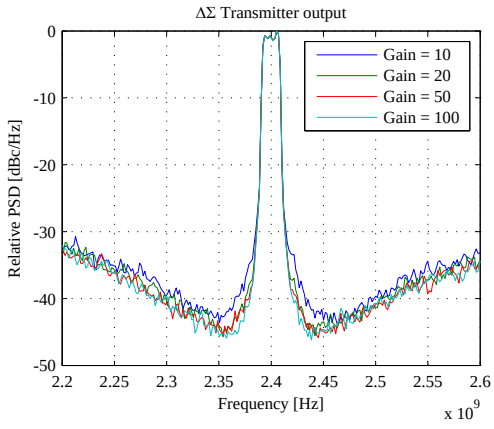


Figure 3.10: *Output PSD for four different gains in the filter. Cutoff frequency is set to* 1 MHz. `[Init 8]`

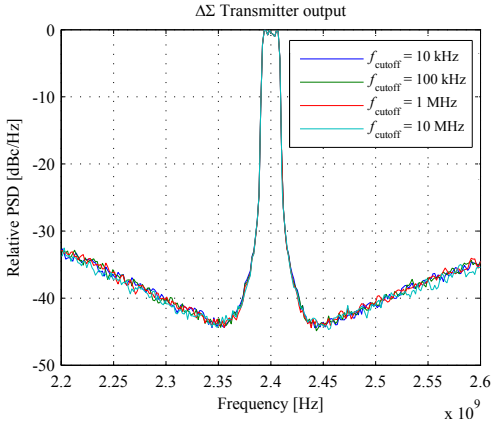


Figure 3.11: *Output PSD for four different cutoff frequencies. Filter gain is set to* 100. `[Init 9]`

Figure 3.10 shows an influence of the filter gain. The best result is obtained with a gain of 100 (40 dB). The filter cutoff frequency is not as decisive for the output spectrum, as seen in Figure 3.11. For the ideal reference simulation a gain of 40 dB and a cutoff frequency of 1 MHz are used.

3.2.2 Simulation Results

The ideal reference simulation is carried out applying the parameter values presented in the previous sections. The simulation results are considered as best case for a given input signal, an RF frequency of 2.4 GHz, and the already specified parameters. The input signal is a pre-generated WLAN burst of 20000 samples prior to up-sampling. The ideal reference spectra are shown in Figure 3.12.

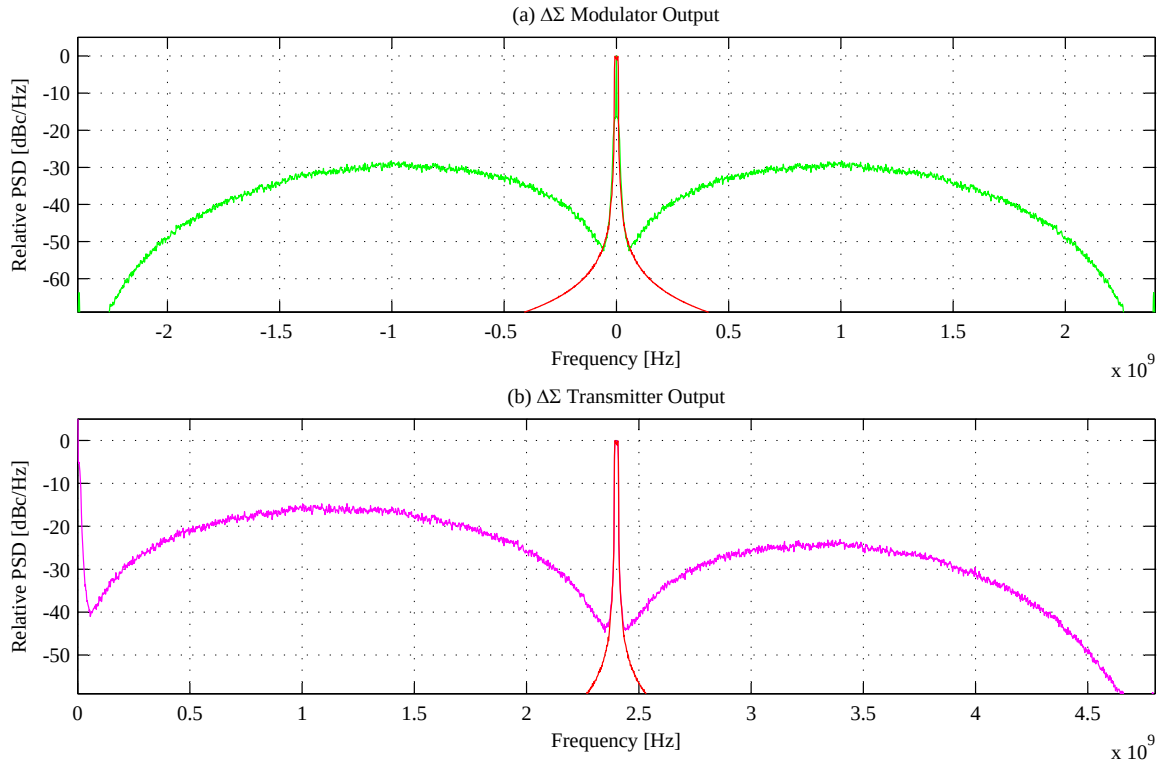


Figure 3.12: *Ideal Matlab reference simulation. Red: complex input signal. Green: $\Delta\Sigma$ Modulator output. Magenta: $\Delta\Sigma$ Transmitter output.* `[Init 10]`

Figure 3.12a illustrates the output spectrum of the $\Delta\Sigma$ Modulator and the spectrum of the input message signal. In Figure 3.12b the output of the $\Delta\Sigma$ Transmitter is depicted. In the same plot a frequency shifted version of the input signal is included. This is done to visualize that the $\Delta\Sigma$ Transmitter output spectrum represents the message signal lifted to a carrier frequency. In this case 2.4 GHz. The simulation of the EVM agrees with the visual inspection. The EVM associated with the ideal simulation is 1.17%.

3.3 Evaluation of Performance Critical Parameters

The system level simulation enables inspection of how certain parameters impair system performance. The parameters are those included in the behavioral models:

- Hysteresis in the clocked comparator.
- Delay in the clocked comparator.
- Slew rate in the filter, clocked comparator, and AND-gate.
- Opamp unity gain bandwidth.

The purpose is to establish how, and at what level, the mentioned parameters degrade system performance, mainly in terms of decreased SNR. The impact of each parameter is simulated using behavioral models in order to establish requirements to the $\Delta\Sigma$ Transmitter. The requirements are based on both system level simulations and literature on high performance $\Delta\Sigma$ modulation. This knowledge allows for a more focused circuit level design phase, as it is revealed which effects that are of significant importance.

In each of the following cases, one parameter is swept while the remaining are equal to those of the ideal reference simulation. The parameter of interest is swept within a range that tries to reveal at what point a degradation of system performance is evident. That is, at what level a distinct change in the output spectrum is seen. This also means that only little consideration is made regarding the level that is expected in an actual implementation.

3.3.1 Hysteresis

Figure 3.13 illustrates how hysteresis in the clocked comparator impairs the output spectrum.

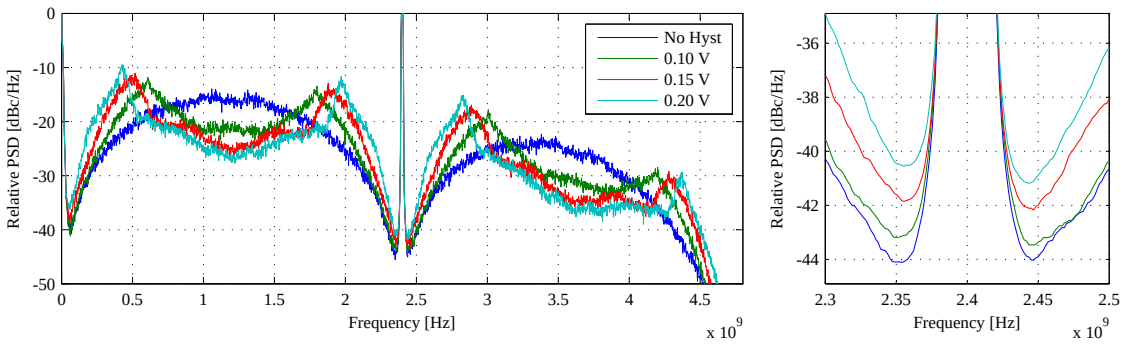


Figure 3.13: *Output spectra for various levels of hysteresis levels in the clocked comparator. Traces to the right are smoothed for easy comparison.* `[Init 11]`

The hysteresis levels are indicated as an offset from 0.9 V, meaning that the two thresholds V_{TH} and V_{TL} are placed symmetrically around 0.9 V. When inspecting Figure 3.13 it is seen that hysteresis influences the spectrum close to the message signal. Also out-of-band noise power is influenced, which aggravates requirements for the transmitter filter. Literature identifies the maximum allowed hysteresis to be in the order of 0.02 V to 0.18 V [Cherry and Snelgrove, 2002, p.65].

When hysteresis is introduced in the clocked comparator, the filter output must accumulate more to trigger a change of the comparator output. Since the filter output has to accumulate more, the time it takes the filter output to reach the threshold values is increased, which in principle increases the time constant of the loop. This changes the quantization noise in the system, which again changes the output spectrum of the $\Delta\Sigma$ Transmitter.

To reduce the impact from hysteresis, the voltage swing on the filter output must be increased. Methods to increase the filter output swing include increasing the filter gain, cutoff frequency, and/or UGBW. Increasing the cutoff frequency or UGBW, increases the frequency content that contributes to the accumulation. Increasing the gain of the filter makes the change in the output more significant. It must be noted that the swing of the filter output should not drive the filter into compression, which limits the possible filter output swing.

From the simulation it is concluded that the hysteresis influences the output spectrum, and in general the hysteresis should be kept below 0.1 V to minimize the degradation of the SNR.

3.3.2 Delay

The delay or Excess Loop Delay (ELD) of the clocked comparator is defined as the delay from the comparator clock to the comparator output [Cherry and Snelgrove, 2002, p. 75]. To some extend this delay is unwanted, but theory also claims that a certain amount of ELD improves the SNR of a typical $\Delta\Sigma$ ADC. Literature states that the best SNR is obtained with an ELD of 25% of the sampling period [Gosslau and Gottwald, 1988]. A simulation is performed in order to clarify whether ELD introduces a similar improvement in the presented $\Delta\Sigma$ Transmitter. The result is presented in Figure 3.14.

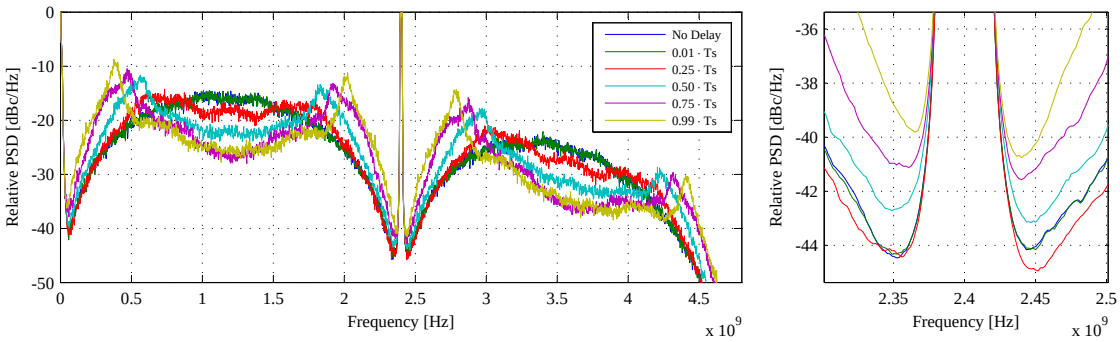


Figure 3.14: *Spectrum illustrating the impact from different degrees of Excess Loop Delay.* `[Init 12]`

Inspecting the simulation result depicted in Figure 3.14 yields the initial conclusion that ELD influences the output spectrum. More significantly, it is seen that the quantization noise can be attenuated approximately 1 dB on the right side of the spectrum if ELD equals $0.25 \cdot T_s$. However, if the delay is larger the SNR is degraded.

ELD influences the spectral behavior of the $\Delta\Sigma$ loop, because it introduces memory in the system. When the input to the comparator is sampled on a rising edge of the clock, the filter continues to integrate the previous comparator output for a duration equal to the ELD, which changes the quantization noise of the loop. The topic is covered in greater detail in Appendix C.

The presented simulations show that in the $\Delta\Sigma$ Transmitter, ELD must be kept below $0.5 \cdot T_s$ in order to minimize the degradation in SNR.

3.3.3 Limited Slew Rate

Figure 3.15 illustrates the simulated output spectra for various levels of Slew Rate (SR) in the filter.

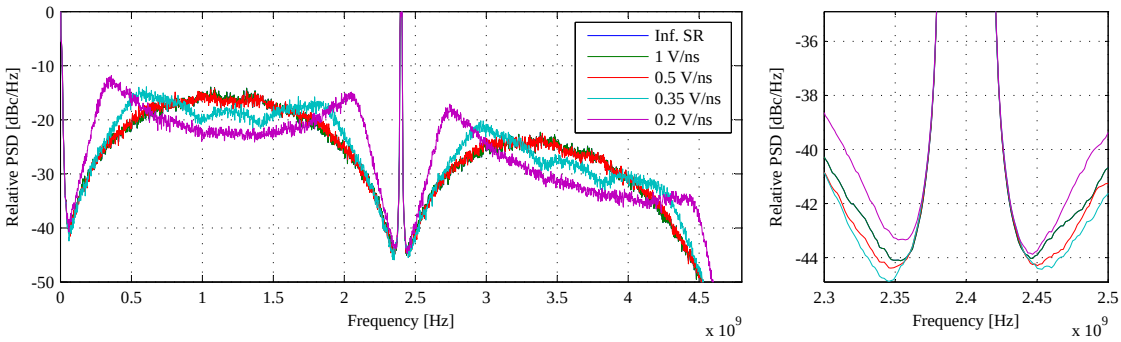


Figure 3.15: *Output spectrum for various levels of SR in the filter.* `[Init 13]`

From the figure it is seen that the slew rate of the opamp output influences the output spectrum. When the slew rate is greater than 0.5 V/ns the impact on the spectrum is insignificant. The literature states that a maximum limit is 0.8 V/ns, which corresponds fairly well with the simulated result [Cherry and Snelgrove, 2002, p. 61].

When the output of the filter is limited by its SR the accumulation at the filter output is slowed down, which is equivalent to introducing a delay in the loop. This statement is supported by the plot of SR $= 0.35$ V/ns, in which case the SNR is improved according to the theory on ELD.

The clocked comparator and the AND-gate are digital circuits. These often realize their functionality using a configuration of several stages of inverters, which source and sink current quickly. This results in a high output slew rate. Figure 3.16 illustrates how the output spectrum is impaired by a limited slew rate in the clocked comparator.

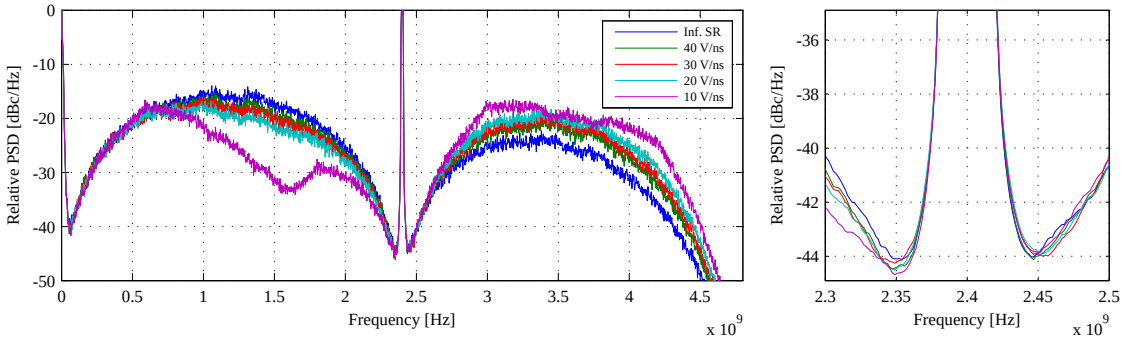


Figure 3.16: *Output spectrum for various levels of slew rate in the clocked comparator.* `[Init 14]`

In comparison to the filter, a much higher level of slew rate is required for the spectrum to coincide with that of the ideal simulation. The impairment caused by limited slew rate is seen in terms of increased out-of-band noise, which poses a harder requirement for the transmitter filter. For a slew rate close to 40 V/ns, no significant impairment is caused.

Figure 3.17 illustrates how the output spectrum is impaired by a limited slew rate in the AND-gate.

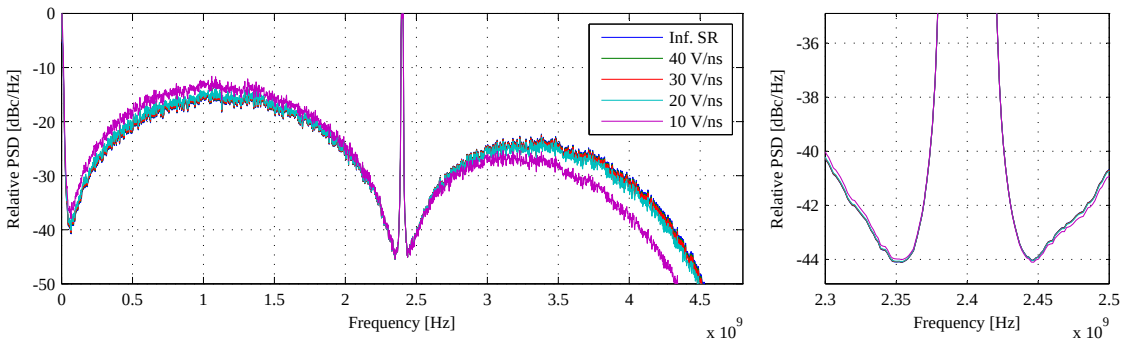


Figure 3.17: *Output spectrum for various levels of slew rate in the AND-gate.* `[Init 15]`

From the figure it is seen that the influence of limited slew rate in the AND-gate is negligible in a range that is expected in an actual implementation.

3.3.4 Opamp Unity Gain Bandwidth

Figure 3.18 illustrates how the output spectrum is impaired for various levels of the opamp UGBW in the filter.

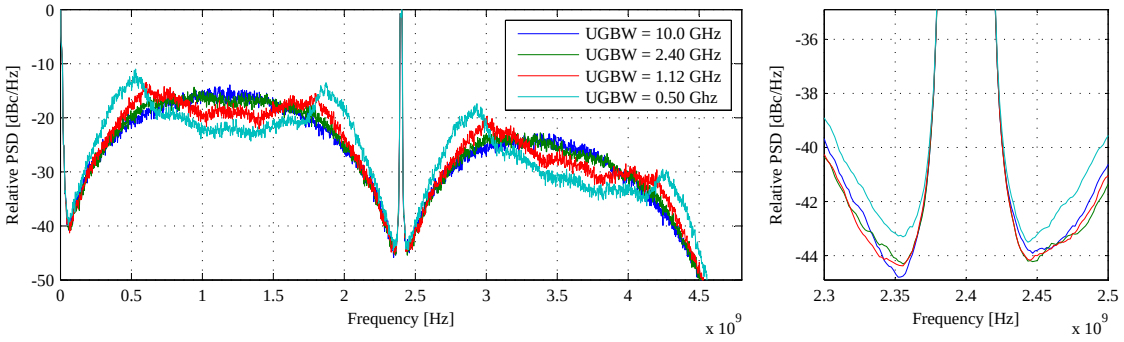


Figure 3.18: *The influence of finite opamp UGBW.* [Init 17]

From the figure it is seen that the UGBW of the filter influences the output spectrum of the $\Delta\Sigma$ Transmitter. The simulation results indicate that an UGBW of at least 2.4 GHz is adequate to minimize the degradation in SNR and avoid inappropriate shaping of the out-of-band noise. This is consistent with the literature that states that the UGBW in an opamp in a $\Delta\Sigma$ Modulator in general should be greater than the sampling frequency in order to maintain maximum SNR [Chan and Martin, 1992, p. 1301].

In order to clarify the influence of infinite UGBW, the output signal of the filter and the feedback is considered. If the output potential of the clocked comparator changes with every rising edge of the clock, the frequency of the feedback signal will be half the sampling frequency. The resulting signal at the filter output will be a saw tooth at that frequency. If the UGBW of the opamp is too low this saw tooth is attenuated. This eventually leads to an impairment of the $\Delta\Sigma$ Transmitter output PSD.

3.4 Ideal ADS Simulation Model

The behavior of the $\Delta\Sigma$ Transmitter is simulated on system level using the Matlab simulation models. When each block is designed on circuit level, a method to verify its performance and isolated impact on the spectrum, is needed. For that purpose an ideal $\Delta\Sigma$ Transmitter is constructed in ADS. Using the ideal ADS simulation model, a single ideal component can be replaced with its designed counterpart, and impact on the performance can then be determined through simulation.

3.4.1 Ideal ADS Circuit Simulation Model

This section describes the construction of each block in the ideal ADS circuit simulation model. The model is used as basis for every following ADS simulation, where a single or more designed blocks are inserted.

Operational Amplifier

An ideal opamp simulation model with the following parameters is included in the $\Delta\Sigma$ Modulator loop: DC gain of 100 dB, a SR of 1.2 V/ps, a UGBW of 10 GHz, and input and output impedances of 1 MΩ and 100 Ω, respectively.

Clocked Comparator

The clocked comparator simulation model is essentially a positive edge-triggered D-flip-flop circuit made up of 7 ideal NAND-gates [Wakerly, 2001, p. 540]. Each NAND-gate is realized using a 3-port logic device including mathematical expressions. The NAND-gate output is changed to 0 V when both inputs are above the threshold of 0.9 V, else the output is 1.8 V. The state of the output signal is changed with unlimited slew rate. The ideal clocked comparator model does not exhibit any delay or hysteresis.

AND-gate

The ideal AND-gate simulation model is realized by a 3-port logic device. Mathematical expressions are included in the device to specify algebraic relationships for the 3-ports. They state that the AND-gate output is 1.8 V if both input signal levels are above 0.9 V, else the output is 0 V. The state of the output signal is changed with unlimited slew rate and the AND-gate exhibits no delay.

3.4.2 ADS Test Bench

The ideal simulation models are included in a test bench where a WLAN signal with an oversample rate of 52 is loaded. The amplitude and phase of the WLAN signal are adapted and fed into the $\Delta\Sigma$ Transmitter. Timed data collectors are connected to the $\Delta\Sigma$ Transmitter in order to collect timed simulation data from selected nodes and save it to the simulation dataset. The test bench forms the basis for ADS simulations when comparing designed circuits with the ideal $\Delta\Sigma$ Transmitter. Figure 3.19 depicts the $\Delta\Sigma$ Transmitter output spectrum from both an ideal ADS simulation and corresponding Matlab simulation.

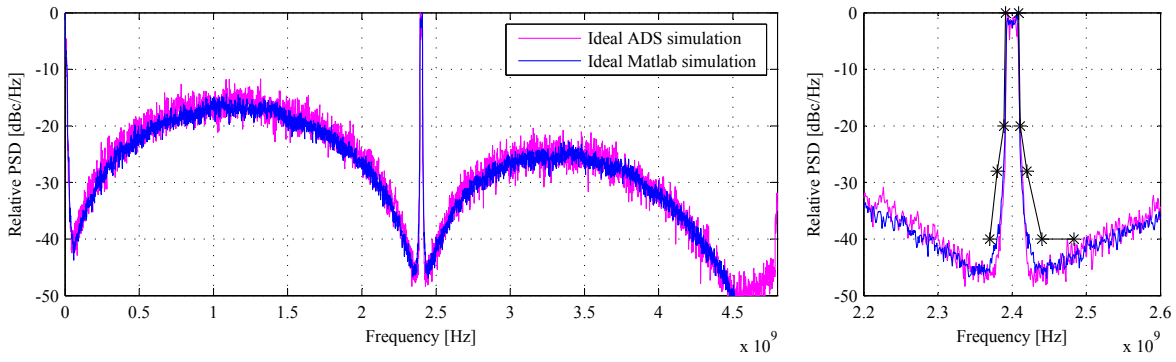


Figure 3.19: *$\Delta\Sigma$ Transmitter output spectrum for ideal ADS and Matlab simulations.* `[DIR000Ref]`

From the figure it is seen that the two spectra are almost identical.

3.5 Summary

The $\Delta\Sigma$ Transmitter is implemented in Matlab on a system level including the possibility to evaluate the influence of critical performance parameters in order to focus the successive circuit level design.

Simulation results reveal that hysteresis in the clocked comparator impairs the output spectrum. For appropriate performance it should be kept on a level below 0.1 V. Delay in same device tends to

impair performance. However a delay of a quarter RF period yields an improved SNR. Under all circumstances, the delay should be lower than half a RF period. The slew rate in the filter should not attain a value lower than 0.8 V/ns, while slew rate in the clocked comparator and AND-gate are of no significant importance. With respect to UGBW of the opamp a frequency of 2.4 GHz ensures adequate performance.

An ideal ADS simulation is presented. This is used as reference when evaluating performance of circuit level designs.

This concludes the Matlab system level simulations. In the following three chapters, the circuit level designs of the filter, clocked comparator, and AND-gate are carried out.

4 Filter

This chapter describes the analysis and design of the low pass filter, which primarily involves the design of an operational amplifier. Several simulations are carried out as part of the design of the opamp and filter. Stability is considered and performance of the filter is simulated. Initially, performance requirements are specified for the opamp and the low pass filter.

4.1 Performance Requirements

The filter uses an opamp as active element. The following section serves the purpose of specifying requirements to filter and opamp performance. Requirements are specified with reference to the literature and system level simulations carried out in the previous chapter. The most important parameters are the opamp gain, BW, and slew rate, but also the limited output signal swing is of importance [Cherry and Snelgrove, 2002, p. 59-62]. The simulated results pose less strict requirements compared to those given in the theory. To ensure adequate performance under all circumstances, requirements are specified mainly using the limits given in the literature. The requirements are summarized in Table 4.1.

Parameter	Requirement
Slew rate	> 0.8 V/ns
Gain	> 26 dB
Cutoff frequency	< 10 MHz
Unity gain bandwidth	> 2.4 GHz

Table 4.1: *Filter and opamp performance requirements.*

In the following two sections an appropriate opamp topology is presented and the design is carried out in terms of dimensioning.

4.2 Opamp Topology

The gain requirement can be met by applying cascoding to the input stage. Cascoding squares the voltage gain without lowering the UGBW. Unfortunately, the driving capabilities and output voltage swing of a cascoded stage is limited. Therefore, a two stage topology is a necessity in order to meet the specified requirements; a high gain input stage and an output stage with high driving capabilities. It is inappropriate to cascade more than two stages, because each stage adds at least one pole in the open loop response of the amplifier. To ensure stability when more than two poles are present in the opamp transfer function, a low frequency dominant pole is needed, which eventually reduces the UGBW of the opamp. [Razavi, 2001, p. 307-309]

A two stage topology is selected, and each stage should be optimized to deliver the highest possible UGBW, and still deliver adequate gain.

4.2.1 Selection of Input Stage

The input stage of an opamp is a differential amplifier. Using a two stage topology, the gain of the opamp must primarily be delivered by the differential stage. Since the needed gain exceeds the intrinsic gain of a Field Effect Transistor (FET) (approximately 20 dB), the gain of the stage must be increased using cascoding [Razavi, 2001, p. 297]. The principle is described in detail in Appendix B.1.

When applying a cascode differential stage, two topologies are commonly used. The telescopic amplifier and the folded cascode amplifier, as illustrated in Figure 4.1.

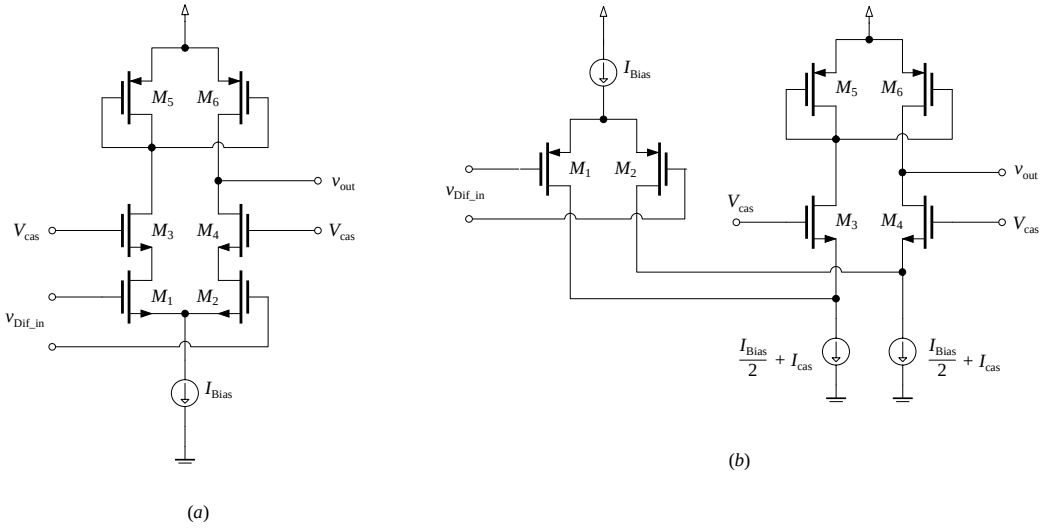


Figure 4.1: *Cascoded differential amplifiers. (a) The telescopic amplifier. (b) The folded cascode amplifier.*

The two amplifiers are composed of the input pair (M_1 and M_2) and their cascode device (M_3 and M_4). Both topologies are loaded by an active current mirror (M_5 and M_6) [Razavi, 2001, p. 150]. Bias current I_{Bias} is generated using a current mirror [Razavi, 2001, p. 298, 302].

One of the differences between the two topologies is that the telescopic amplifier uses a pair of NMOS transistors, where the folded cascode amplifier uses a pair of PMOS transistors as input devices. Literature states that this causes the telescopic amplifier to have a higher UGBW, compared to the folded cascode counter part [Razavi, 2001, p. 314].

In order to evaluate the performance of the two topologies, simulations are performed in ADS under equal bias conditions. Simulation results are depicted in Figure 4.2.

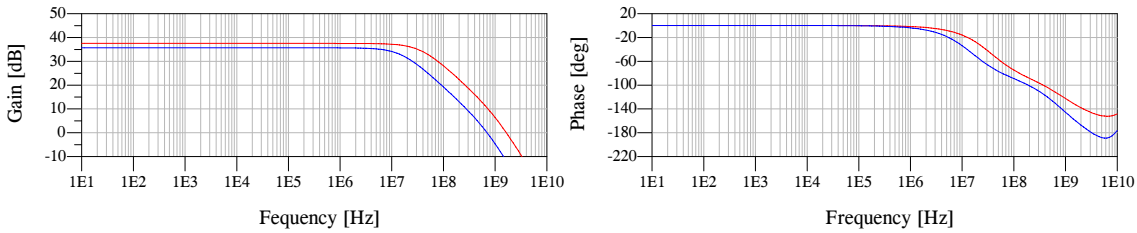


Figure 4.2: *The unloaded bode plots of the telescopic amplifier (red) and the folded cascode amplifier (blue).* `[DIR009]`

The simulation results show that the folded cascode amplifier has less gain and decreased UGBW compared to the telescopic amplifier. Based on the fact that both speed and gain are important parameters in the current application, the telescopic topology is selected as the input stage of the opamp.

4.2.2 Selection of Peripherals

For realizing proper functionality of the telescopic amplifier, peripheral circuits are needed. In Figure 4.1 it is seen that the amplifier uses both a current source and a voltage source. Both circuits are designed in the following paragraphs.

Current Source

Biasing of the telescopic differential amplifier is realized with a current source. The current source is implemented as a current mirror and a reference current generator as depicted in Figure 4.3a.

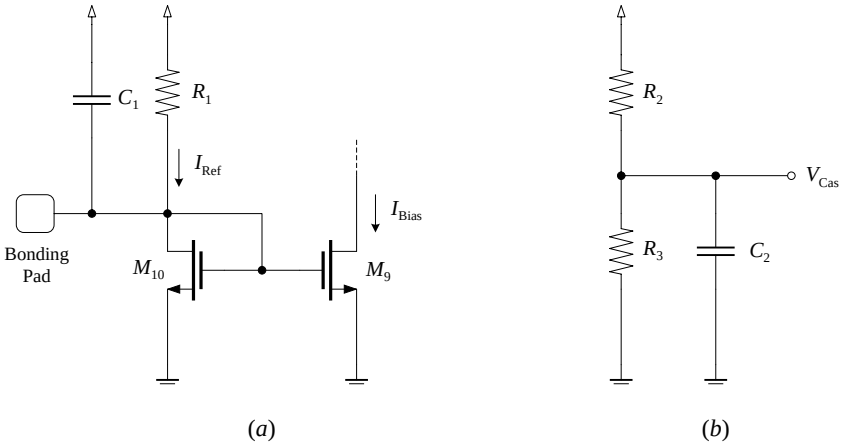


Figure 4.3: *The peripherals needed for the telescopic amplifier. (a) Current mirror using a resistor R_1 for generating the reference current I_{Ref}. (b) Voltage source generating V_{Cas}.*

The current mirror is realized by M_9 and M_{10}. In order to make M_{10} sink the current I_{Ref}, the gate-source voltage of the FET is set by the resistor R_1. As the gate-source voltage of M_9 is equal to that of M_{10}, I_{Bias} becomes equal to I_{Ref}, and the reference current is replicated, given that the two transistors are of equal size.

Voltage Source

For proper operation of the opamp, the voltage V_{Cas} is generated, using the circuit illustrated in Figure 4.3b. Simulations show that the frequency response of the amplifier is not influenced by the cascode voltage, as long as the voltage is larger than 1 V. The cascode voltage is implemented using a voltage divider and the node is decoupled using the capacitor C_2. Setting $R_2 = 2\ \text{k}\Omega$ and $R_3 = 5\ \text{k}\Omega$ gives a cascode voltage of 1.3 V.

4.2.3 Selection of Output Stage

The following section presents the three most promising output stages. In Figure 4.4 the three output stages, in their best achieved configuration, are shown together with simulation results.

Stage 1 is in principle a voltage buffer, as it is based on a common drain transistor pair [Boonyaporn and Kasemsuwan, 2002]. The stage has a low output impedance which makes it attractive since the voltage over the load capacitor can change rapidly. This fact is seen in the loaded step response of Figure 4.4.

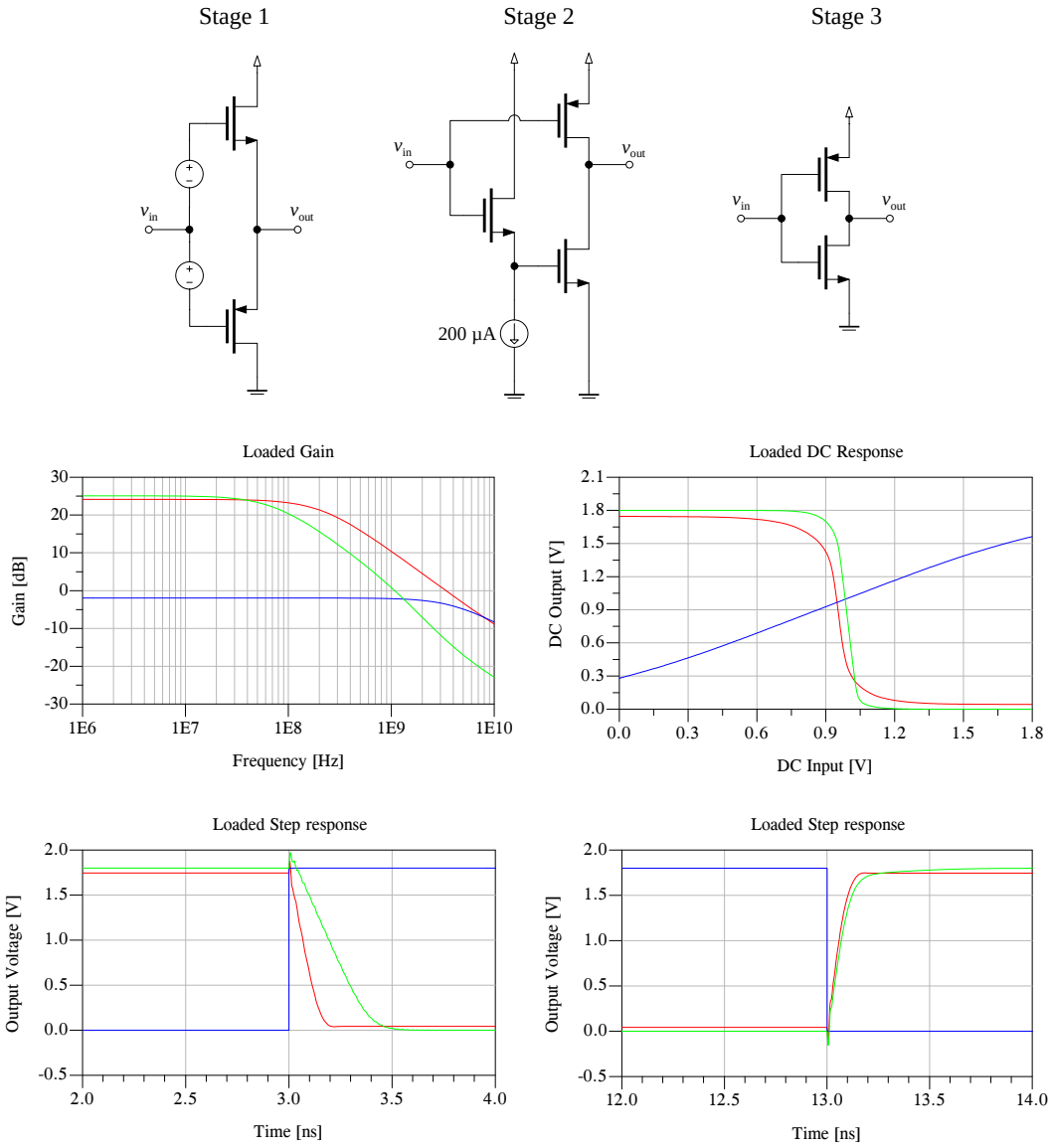


Figure 4.4: *Comparison of three different output stages described in Section 4.2.3. Circuits are loaded with* 100 kΩ *in parallel with* 1 pF.
Blue: Stage 1, Push pull source follower.
Green: Stage 2, Inverter w. source follower.
Red: Stage 3: Inverter. `[DIR011]`

Simulations are performed on designs found in the literature, that realize the common drain configuration. In general, a dominant pole at a few MHz is introduced, which reduces the UGBW of the opamp to an unacceptable level. Stage 1 is on this basis not an option as output stage for the opamp.

Stage 2 and stage 3 are both based on a push pull configuration. This sets the output impedance equal to r_o of the transistor.

Stage 2 uses the push-pull configuration and a source follower as driver to the NMOS transistor [Wang and Sánchez-Sinencio, 1997]. In Figure 4.4 it is seen that the stage delivers a slightly higher DC gain and an improved output swing, compared to stage 3. Unfortunately, the driver stage introduces a dominant pole in the transfer function, which reduces the UGBW of the stage.

Stage 3 is a simple inverter. It has the highest pole of the two, which maximizes the speed of the operational amplifier. Also in terms of SR stage 3 performs better than stage 2, as seen in the loaded step response of Figure 4.4. The linear output swing of the stage is approximately equal to $1.8 - 2V_t$.

Based on the facts, that stage 3 has the highest pole and exhibits a larger SR, it is selected as output stage.

4.3 Opamp Design

The topology of the opamp and the needed peripheral circuits is selected, and the resulting circuit is depicted in Figure 4.5.

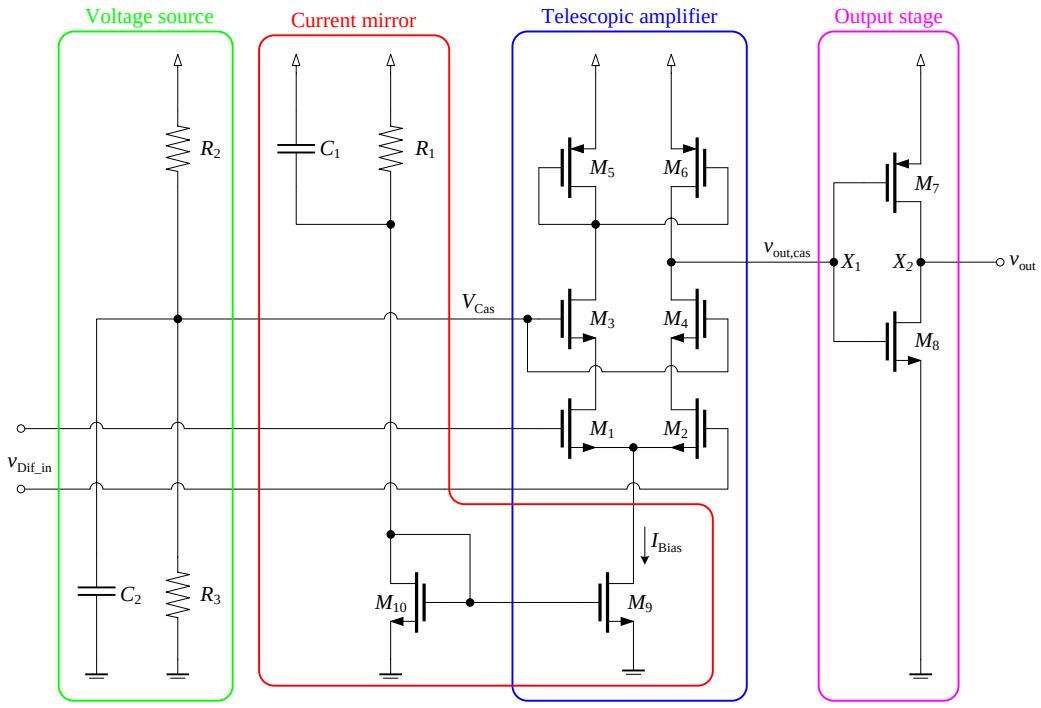


Figure 4.5: *The schematic of the opamp. The bond pads are not included in the schematic.*

The configuration of each block of the opamp is specified in the previous sections. The following sections are addressed to sizing the transistors of the two stages and the current mirror.

4.3.1 Design of Input Stage

The sizes of M_1 through M_4 should be minimal as this minimizes the parasitics of the transistors in the signal path, which results in high speed. The size of the transistors in the active current mirror M_5 and M_6 should also be small. This places the mirror pole at the highest possible frequency.

In general, making the transistors small enhances the UGBW, but on the downside limits the output signal swing of the stage. This is due to the fact that the voltage headroom needed to hold the transistors in saturation is increased. The size of M_1 through M_6 is set to 5 fingers which maximizes the speed of the input stage. The requirement for output swing must then be ensured by the output stage.

4.3.2 Design of Current Mirror

Since the current mirror is not a part of the signal path, the size of the transistors has less impact on the frequency response of the circuit. The size of the transistors can therefore be made large without degrading UGBW. The size of M_9 and M_{10} is set to 21 fingers, which is the largest number of fingers a single transistor can have.

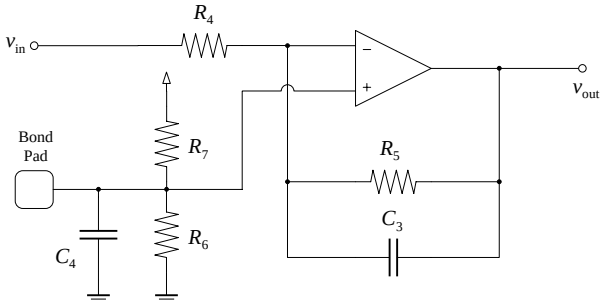
Through simulation the bias current I_{Bias} is set to 200 μA, which compromises neither speed nor output voltage swing. In this case the size of the resistor is 1.6 kΩ. In general the frequency response of the amplifier can be improved by increasing the bias current. This makes it appropriate to allow for external tuning of the current. For this purpose a bond pad is connected to the drain of M_{10} and the voltage is decoupled by C_1. If the reference current needs to be changed during measurements, the drain-source potential of M_{10} can be controlled externally, making it possible to tune I_{Bias}.

4.3.3 Design of Output Stage

The two transistors in the output stage are dimensioned as an inverter for a symmetric operation. Hence, the NMOS has 5 fingers and the PMOS has 18, as discussed in Appendix B.3. Still it is intended that the transistors are as small as possible to obtain high UGBW.

4.4 Filter Design

This section describes the design of the filter based on the designed opamp. The filter circuit is presented in Figure 4.6 and the equations setting the gain and cutoff frequency of the filter are given in (4.1) and (4.2) [Sedra and Smith, 1998, p. 902].



$$A_{\text{v,DC}} = -\frac{R_5}{R_4} \tag{4.1}$$

$$f_{\text{cutoff}} = \frac{1}{2 \cdot \pi \cdot C_3 \cdot R_5} \tag{4.2}$$

Figure 4.6: *Schematic of the filter circuit.*

The two equations predict the gain and cutoff frequency of the filter when using an ideal amplifier.

Since neither gain nor cutoff frequency are critical parameters, the equations are used and the deviation introduced by the non-ideal opamp is accepted.

When integrating the filter, the size of the passive components must be given some attention. As the capacitor C_3 is in the signal path a RF capacitor is needed.

Setting $R_5 = 100\,\text{k}\Omega$ and $C_3 = 1\,\text{pF}$, gives a good compromise between a small capacitor and a relatively small resistor in terms of physical size, while at the same time implementing the needed cutoff frequency. The DC-gain of the filter is then only defined by R_4. Setting it to $2\,\text{k}\Omega$ yields a gain of $34\,\text{dB}$.

These specifications are applied in the following. The final dimensioning is settled when the combined $\Delta\Sigma$ Transmitter is designed.

4.5 Stability Considerations

The stability of the filter is analyzed using both a step response and the Barkhausen's Criteria. A common method to evaluate the stability of a circuit is to use the Barkhausen's Criteria, which states that stability is ensured, if the closed loop gain is less than unity when the phase shift is greater than $180°$ [Razavi, 2001, p. 346]. When evaluating for stability using the Barkhausen's Criteria the bode plot of the feedback system is inspected. The Barkhausen's Criteria evaluates the small signal stability. If it must be ensured that the feedback system is stable when exposed to large signals, a large signal step response must be simulated. A large signal step response includes the non-linear circuit model, the change in bias point, and the change in frequency response of both the forward loop and feedback loop. [Razavi, 2001, p. 354] To evaluate the stability of the circuit, simulations are performed and results are presented in Figure 4.7.

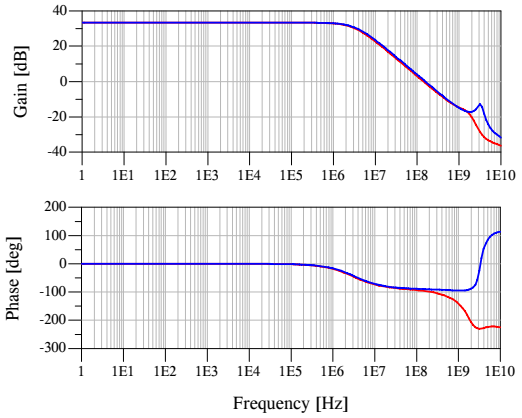


Figure 4.7: *Filter bode plots. Design in Figure 4.6 (blue) and added Miller capacitor (red). Phase is indicated in an offset from* $-180°$. `[DIR013]`

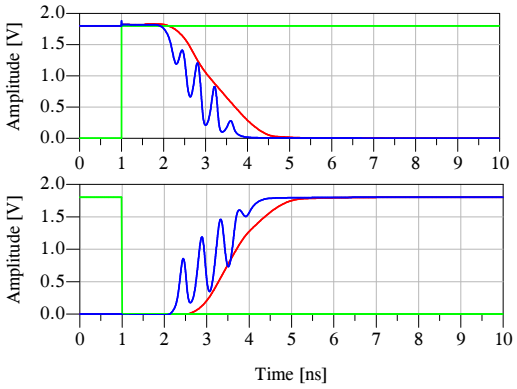


Figure 4.8: *Large signal step response of the filter. Input step (green), response of design in Figure 4.6 (blue), and response when Miller capacitor is added (red).* `[DIR013]`

From the bode plot it is seen that the Barkhausen's Criteria is met with a phase margin of $18\,°$, but that both the amplitude and phase response behave inappropriately around 3 GHz, indicating a potential problem. When the large signal step response in Figure 4.8 is inspected, ringing is clearly detected in the response.

In order to solve this problem, a dominant pole is included in the filter transfer function. When suffi-

cient gain margin and phase margin is obtained, ringing and instability is avoided. The pole is implemented by adding a Miller capacitor of 0.4 pF between node X_1 and X_2 in Figure 4.5. In Figure 4.7 and 4.8 it is seen that both the bode plot and the large signal step response indicate that the filter is stable in the case where the Miller capacitor is added.

The drawback of adding a dominant pole to the opamp is a reduced UGBW. The UGBW of the amplifier is decreased from 2.82 GHz to 1.12 GHz.

This concludes the design of the filter. In the following section, a simulated evaluation of the filter is presented.

4.6 Simulated Filter Performance

The opamp and filter circuits are simulated in order to determine whether requirements are met. The simulation results are given in Table 4.2.

Parameter	Simulated
Filter Slew rate High to Low	-0.83 V/ns
Filter Slew rate Low to high	1.02 V/ns
Filter gain	34 dB
Filter cutoff frequency	1.6 MHz
Phase margin	38.4 °
Gain margin	6.45 dB
Open Loop gain	56.78 dB
Unity gain bandwidth	1.12 GHz

Table 4.2: *Parameters of the simulated opamp and filter circuits.*

From the table it is seen that the requirement for DC-gain and SR is meet. The requirement for UGBW is not, which is expected to degrade SNR. Some possibilities exist in improving the UGBW when performing real measurements. The bias current can be increased, which eventually will increase the UGBW of the opamp. The layout is designed such that the Miller capacitor can be removed by laser. This will increase the UGBW, but also reduce the phase and gain margin, which could make the circuit unstable.

To clarify the performance degradation when including the filter in the $\Delta\Sigma$ Transmitter, an ideal ADS simulation is performed, where only the ideal opamp is replaced by its designed counterpart. The simulation result is depicted in Figure 4.9.

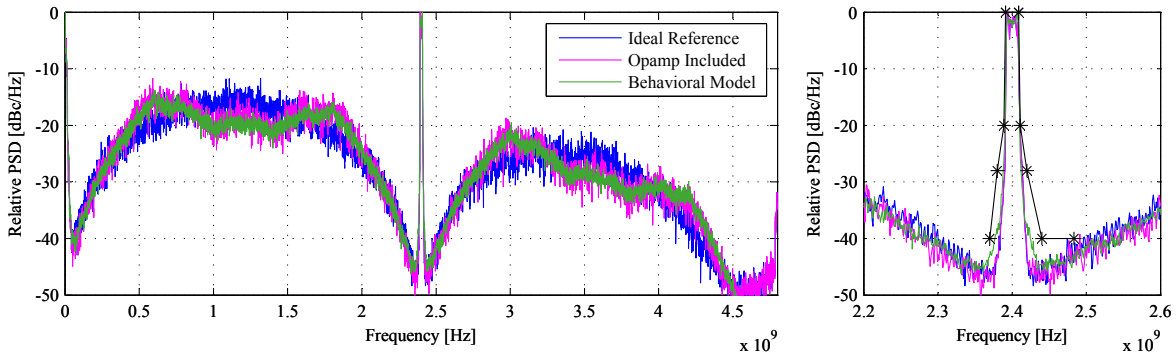


Figure 4.9: *Comparison between the ideal ADS simulation, simulation including the designed filter, and simulation of the behavioral model.* `[DIR001]` `[Init 21]`

From the figure it is seen that the output spectrum is influenced by the filter, because the UGBW of the opamp is insufficient. In addition it is seen that the behavioral model of the filter predicts the impairment satisfactory without significant deviations.

4.6.1 Process Variations

The United Microelectronics Corporation (UMC) models make it possible to simulate with three different types of transistor models. The fast, normal, and slow models. In order to ensure that the filter operates as intended when implemented on chip, the bode plot of the amplifier using the three different transistor models is simulated `[DIR012]` . Results show that the frequency response of the filter is almost unchanged when the transistor model is changed. The difference between the three simulations is found in the output DC level, which decreases, when the speed of the transistor increases.

4.7 Summary

A design of a operational amplifier based on a telescopic differential input stage, exploiting an inverter as output stage is given. To obtain stability the opamp is stabilized using a Miller capacitor. Finally the performance of the filter is simulated, and the results show that requirements are met, except for UGBW.

5 Clocked Comparator

This chapter documents the design of the clocked comparator. Circuit performance is simulated and influence on the overall system performance is discussed.

5.1 Basic Functionality

The clocked comparator serves the purpose of repeatedly translating the input signal to a low or high voltage at a given time. The sampling of the input signal is controlled by the phase modulated clock. On rising edges of the clock, the comparator delivers an output dependent on the voltage level of the input signal. Input levels above $V_{DD}/2$ at sampling instances yields a high output (V_{DD}), while an input level below $V_{DD}/2$ yields a low output (0 V).

With respect to functionality a CMOS positive edge-triggered D-flip-flop serves the purpose of sampling the quantized input signal [Wakerly, 2001, p. 540]. This flip-flop is a combination of different logic gates that all together consist of 40 transistors, which is not very space efficient and introduces an inappropriate large delay. With reference to Section 3.3, a large delay in the comparator must be avoided. More specifically, the delay should be below $T_s/2 = 208$ ps.

A more suitable solution that uses fewer transistors and reduces the delay is presented below.

5.2 Circuit Design

As illustrated in Figure 5.1 the circuit is based on a 9-transistor 3-stage configuration which demonstrates the basic functionality of a positive edge-triggered D-flip-flop.

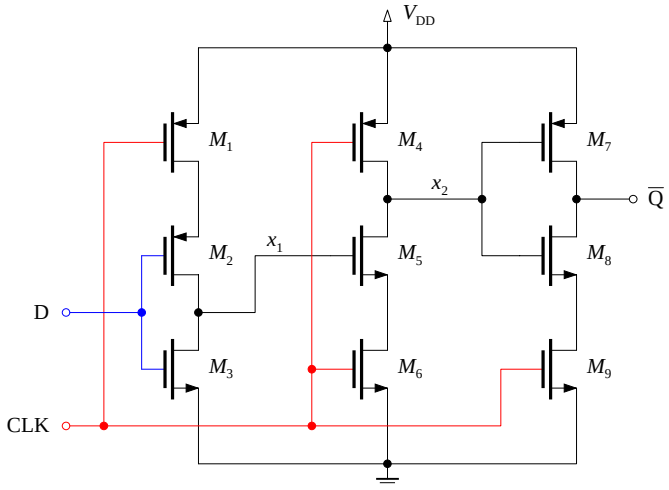


Figure 5.1: *Positive edge-triggered D-flip-flop [Huang and Rogenmoser, 1996]. Input D from filter (blue) and clock input CLK (red).*

The flip-flop is comprised by three stages as described below [Yuan and Svensson, 1989, p. 65]:

- Clocked PMOS latch (M_1 through M_3).
- NMOS pre-charge stage (M_4 through M_6).
- Clocked NMOS latch (M_7, through M_9).

The desired functionality can be realized due to the parasitic capacities of the MOSFETs. As an example, when a high signal is applied at the gate of a FET, its gate-source capacity is charged to that high voltage level. If then the gate node is isolated due to a change of state elsewhere in the circuit the node will remain at its high level for some time before the capacity is discharged.

For understanding the positive edge-triggering property, the NMOS pre-charge stage and the clocked NMOS latch of the circuit in Figure 5.1 are initially considered.

With a low clock signal (pre-charge/hold state) node x_2 is charged to a high level and M_7 and M_9 are off. At this time node $\overline{\text{Q}}$ is isolated from any supply and it therefore holds the result from the previous evaluation. The evaluation state is initiated as the clock switches from low to high. Transistor M_6 enables x_2 to change to low depending on the input x_1, as M_4 is switched off. Likewise, M_9 enables $\overline{\text{Q}}$ to change to low depending on x_2. If x_1 is high at the time of a rising clock, the pre-charged x_2 will change to low, changing $\overline{\text{Q}}$ to high. On the other hand, if x_1 is low, x_2 is isolated and will therefore remain at its pre-charged high, changing $\overline{\text{Q}}$ to low.

The actions above account for the desired functionality when the clock changes from low to high, but it is not adequate. As the system is needed to be edge-triggered the output should be insensitive to any changes in the input at all times, except when the clock exhibits a positive edge. Considering once again the NMOS pre-charge stage and the clocked NMOS latch of Figure 5.1 on the preceding page in the case where clock is high. Transistor M_4 will then be off and M_6 on, meaning that a change of x_1 from high to low will not have any effect on x_2 as this node is isolated from VDD. However, in the case where x_1 changes from low to high, x_2 will change from its pre-charged high to low as both M_5 and M_6 are on. This will in turn change $\overline{\text{Q}}$ from low to high, which is not intended.

In order to rectify this problem the clocked PMOS latch is employed. Transistor M_1 makes it impossible to realize a conduction path between V_{DD} and x_1 during a high clock. This ensures that x_1 cannot change from low to high during a high clock. A change of x_1 from high to low can still occur, but as mentioned, this does not affect the output.

5.2.1 Design Modifications

During the three stages described above the input signal D is inverted an odd number of times which necessitates the use of an inverter at the output (M_{13} and M_{14}), as illustrated in Figure 5.2.

Despite the fact that the presented circuit delivers the desired functionality it suffers from unfortunate behavior when subjected to a certain sequence of logic transitions [Huang and Rogenmoser, 1996, pp. 461]. Referring to Figure 5.1, if both D and CLK are low, x_1 will be high while x_2 is pre-charged to high. When the clock changes to high M_4 is turned off and M_6 and M_9 are turned on. This creates a discharge path to ground for x_2. However, as the node cannot discharge instantly, it leaves a chance for $\overline{\text{Q}}$ to discharge through M_8 and M_9 for a short moment until x_2 reaches a level low enough to turn M_7 on and M_8 off. The size of this glitch depends on dimensioning of the transistors involved, however it can be fully eliminated by modifying the circuit as described below.

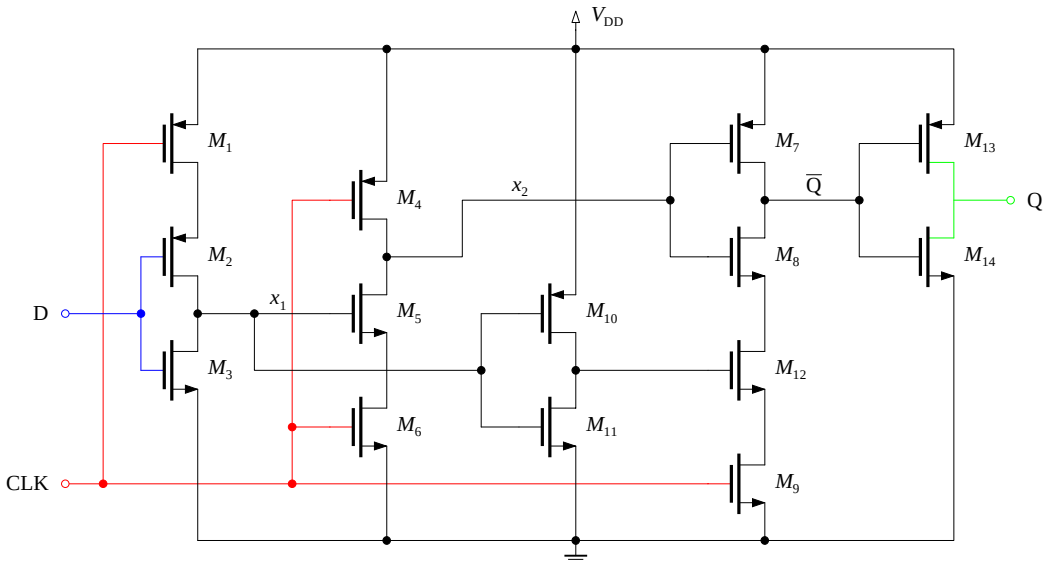


Figure 5.2: *Glitch-free positive edge-triggered D-flip-flop [Huang and Rogenmoser, 1996].*

Glitch Removal

To avoid the discharge of node $\overline{Q}$ when x_1 is high and the clock exhibits a positive edge, an NMOS transistor M_{12} is inserted in the discharge path of $\overline{Q}$, as illustrated in Figure 5.2. Via an inverter (M_{10} and M_{11}), M_{12} is turned off before the end of the pre-charge/hold state. This cuts off the discharge path before M_9 is turned on at evaluation.

The additional inverter stage does not add to the delay of the comparator as it is placed in parallel with the NMOS pre-charge stage.

5.2.2 Transistor Sizing

Sizing of the transistors in Figure 5.2 is done with one primary objective in mind:

- Minimizing the delay from a positive edge in the clock signal to the corresponding change in the output signal.

Obtaining high speed in a digital circuit involves finding the best compromise for the sizes of the different transistors in all state transitions [Huang and Rogenmoser, 1996, p. 459]. A trade-off must be made of the size of some, if not all, transistors, as their "role" changes from one state transition to another. A change of role means that in one situation a transistor functions as a driver for another transistor, while in the next situation the same transistor poses as a load. For high speed performance this duality forces conflicting constraints on the physical parameters of the transistor.

The gate capacitance of a MOSFET is approximately proportional to $C_{ox} \cdot W \cdot L$, where C_{ox} is the gate-oxide capacitance, W is the gate width, and L is the gate length [Huang and Rogenmoser, 1996]. When a transistor, or more precisely its gate capacitance, loads a given node, a relatively small gate capacitance is desired, as this results in a small time constant for charging the node. Hence, a small gate width W is needed. On the other hand, when the transistor is used to charge or discharge a node quickly, a large drain current is needed. As the drain current also is proportional to the gate width, conflicting requirements exist.

The sizing of the transistors is based on analysis of the different transitions in the circuit [Huang and Rogenmoser, 1996, pp. 457] assisted with a number of design iterations in terms of simulations. The size of each transistor, expressed in number of fingers, is given in Table 5.1.

Transistor	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}
No. of fingers	18	18	5	5	10	10	12	5	5	5	5	5	18	5

Table 5.1: *Transistor sizes in the clocked comparator.*

The performance of the designed clocked comparator is investigated through simulations in the following.

5.3 Circuit Simulations

The circuit simulations seek to describe the performance of the designed circuit in terms of functionality and non-ideal effects. Moreover, the comparator is included in the ideal ADS model in order to describe the impairment of system performance caused by the clocked comparator.

5.3.1 Functionality and Non-Ideal Effects

To verify the functionality of the comparator, the circuit design presented in Figure 5.2 on the previous page is simulated in ADS using square wave generators. The clock is switching at a frequency of 2.4 GHz and the input at 0.8 GHz with a duty-cycle above 50%. The result is presented in Figure 5.3.

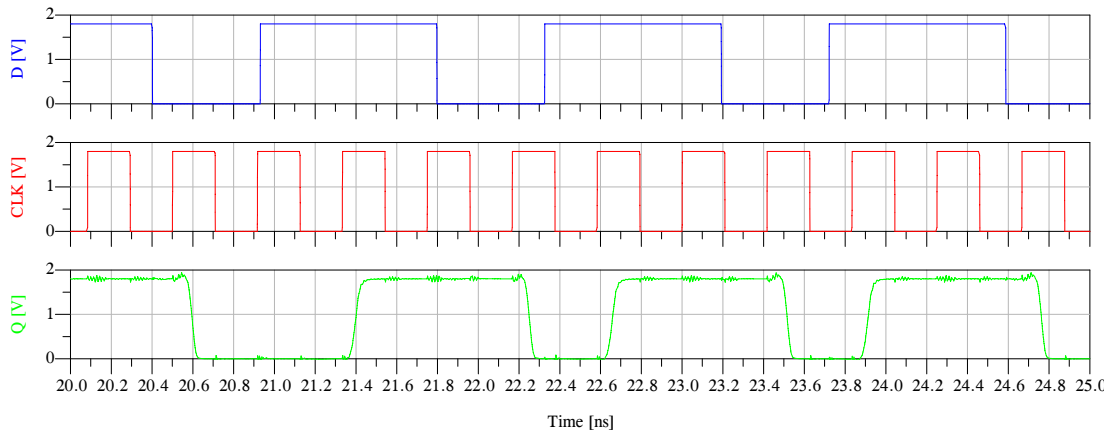


Figure 5.3: *Simulated functionality of clocked comparator* `[DIR004]`.

As shown in the figure, the desired functionality is realized. Furthermore it is seen that the comparator exhibits a delay from the clock to the output as expected. Also, slew rates of approximately 35 V/ns for rising edges and 40 V/ns for falling edges, are observed.

Another simulation is performed in order to determine the threshold voltages and the delay between the clock and the output of the clocked comparator.

Both the delay and threshold values vary as a function of the input signal. Literature defines this phenomenon as metastability, which causes spectral whitening and degrades the SNR [Cherry and Snelgrove, 2002, p. 108 and 136]. As the input signal has great effect on the non-ideal parameters

of the clocked comparator, the signal used when determining these parameters, should be a realistic WLAN signal, and not a square pulse.

By analyzing ADS based behavior of the clocked comparator, in terms of input and output signals, it is possible in Matlab to determine which values of the input signal, that triggers a change in the output signal. The size of the delay from the rising edge of the clock signal to the change in output signal can also be determined.

From this information, the threshold limits and a statistical approximation of the delay can be found, in the case where a given WLAN input signal is applied. Figure 5.4a shows the histogram used for determining hysteresis of the clocked comparator.

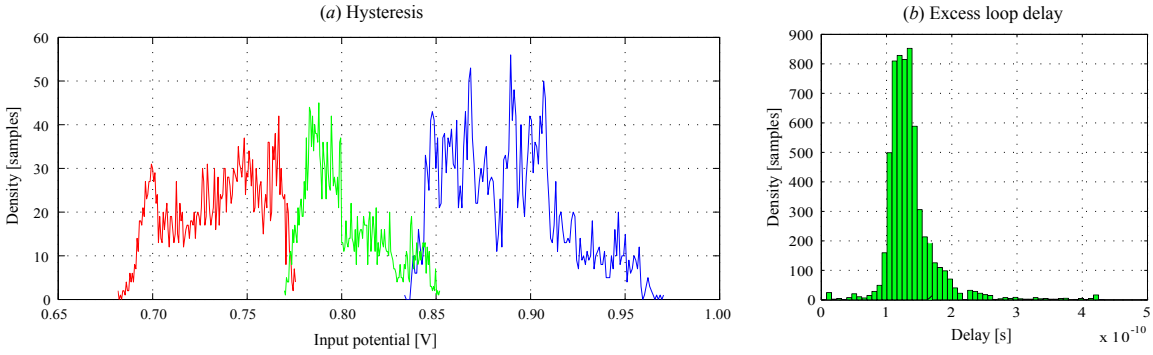


Figure 5.4: *Histograms used to determine non-ideal effects of clocked comparator. (a) Input levels that trigger a change in output: Change in output from high to low (red), wrongfully no change in output (green), and change in output from low to high (blue). (b) Histogram of delay from the clock to the comparator output.* `[DIR024]`

Several things are worth noticing. First, the lowest level that triggers a change from low to high and the highest value that triggers a change from high to low are not placed symmetrically around 900 mV. Instead, the mean value is close to 800 mV. This, however, does not impair system performance, as the comparator is a part of a feedback loop, which eventually will make the filter output settle around the mean of the two threshold levels. The explanation for the deviation could probably be found in the chosen size of the transistors. Secondly, the overlap of the green curve with the red and the blue indicates that the threshold levels change, depending on the input signal. As an example, an input level between 0.840 mV and 0.850 mV will only in some occasions trigger a change from low to high. Samples of the green curve are generated when the input crosses 0.8 V without generating a change in the output on a rising of the clock. The value of 0.8 V is the mean value of the red and blue samples. Lastly, the upper and lower threshold values that indisputably trigger a change in output are 0.85 V and 0.77 V respectively. These values represent the hysteresis of the clocked comparator, which is within the requirement of a maximum offset from mean of 0.1 V, specified in Section 3.3.1.

Each time a sampled value changes the output value of the clocked comparator, the delay between the rising edge of the clock to the output is computed. The output is considered to have changed state when the output potential reaches 0.9 V. In Figure 5.4b the histogram of the computed delay is plotted. It is seen that the density is high between 100 ps and 150 ps. A mean value of 137 ps is calculated, which is below the required 208 ps.

5.3.2 Comparator Included in Ideal Setup

To clarify the performance degradation caused by the designed clocked comparator, it is included in the ideal setup. The resulting output spectrum, and spectra used as reference, are illustrated in Figure 5.5.

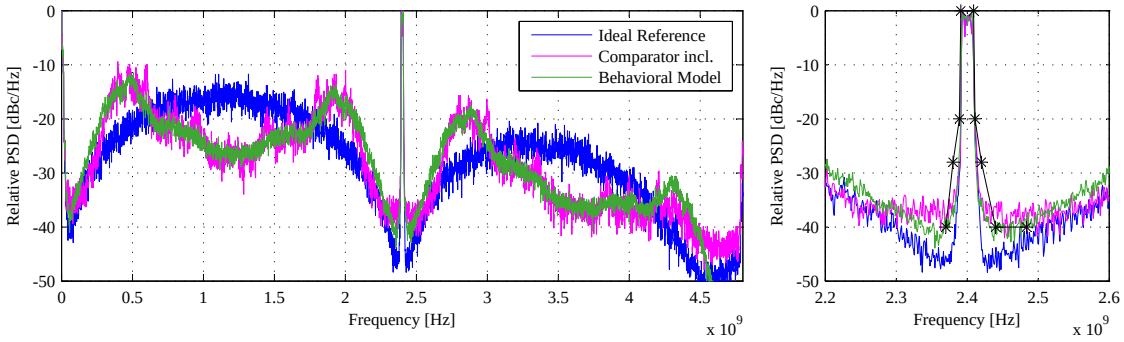


Figure 5.5: *Frequency domain representation of AND-gate output. Blue: Ideal reference simulation in ADS. Orange: The designed clocked comparator included in the ideal simulation. Red: Matlab simulation utilizing the extracted values of non-ideal effects.* `[DIR002]` `[Init 19]`

From the figure it is seen that when the designed clocked comparator is included in the simulation the quantization noise is increased significantly, compared to the results from the ideal ADS simulation.

The degradation in SNR is caused by the delay and hysteresis introduced by the comparator. Comparison with the Matlab simulation shows that the model accounts well for the behavior of the comparator, despite that this includes constant values of delay and hysteresis. The spectral whitening seen in the ADS simulation is as mentioned caused by metastability. A more comprehensive behavioral model would include the metastability that the comparator clearly exhibits.

5.4 Summary

In the sections above, the design of the clocked comparator is presented. The design is a modified D-flip-flop that in simulations exhibits an input dependent delay with a mean of 137 ps. Likewise, the comparator shows varying hysteresis with a worst-case of 0.05 V. However, both non-ideal effects are within the required limits.

6 AND-gate

This chapter documents the analysis and design of the AND-gate circuit located as the last sub-circuit in the $\Delta\Sigma$ Transmitter. Circuit performance is simulated and influence on overall system performance is discussed.

6.1 Basic Functionality

The AND-gate is located in the block diagram of the $\Delta\Sigma$ Transmitter in Figure 2.1 between the $\Delta\Sigma$ Modulator and the PA. The square signal from the $\Delta\Sigma$ Modulator is AND'ed with a delayed version of the clock in order to ensure that the output signal is low half of the time in each RF period. Basically, the AND-gate performs a frequency translation of the $\Delta\Sigma$ Modulator output signal, moving it to the carrier frequency.

6.2 Circuit Design

The functionality is realized using a 2-input CMOS AND-gate depicted in Figure 6.1.

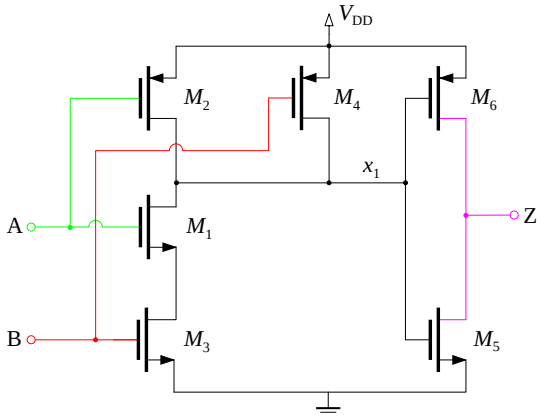


Figure 6.1: *CMOS AND-gate circuit diagram [Wakerly, 2001, p. 93].*

The AND-gate is comprised of three NMOS and three PMOS FETs. Basically, the circuit consists of a NAND-gate connected to an inverter stage at the output. The output of the NAND-gate is denoted x_1.

If both inputs of the AND-gate are high, the path from V_{DD} to x_1 is blocked, since transistors M_2 and M_4 are off. On the same time transistors M_1 and M_3 are on, connecting x_1 to ground, and thereby connecting Z to V_{DD} through M_6. If either input is low, the corresponding PMOS transistor of the NAND-gate (M_2 or M_4) is on, connecting x_1 to V_{DD}. This situation creates a path between Z and ground through M_5.

6.2.1 Sizing of Transistors

The sizes of the transistors are chosen in order to ensure that the rise- and fall times of the output signal are as equal as possibly achievable. The delay in the AND-gate is of less significance compared to the clocked comparator, since it is not included in the $\Delta\Sigma$ Modulator loop. It is important that the pulses in the AND-gate output signal have a constant delay, if not, the output signal is phase distorted. The delay is only significant in the case where the delay is a part of the $\Delta\Sigma$ Modulator loop. Therefore the procedure of choosing transistor sizes is based primarily on the output signal symmetry and secondly on the delay.

Based on simulations, the best achievable signal symmetry is obtained when using the chosen transistor sizes given in Table 6.1.

Transistor	M_1	M_2	M_3	M_4	M_5	M_6
No. of fingers	5	10	5	10	5	18

Table 6.1: *Chosen sizes of transistors in the AND-gate circuit.*

The sizes of M_5 and M_6 are supported by the description of the CMOS inverter given in Appendix B.3.

6.3 Circuit Simulations

Circuit simulations are performed with the purpose of determining the performance of the designed AND-gate in terms of functionality and non-ideal effects. Moreover, the AND-gate is included in the ideal ADS model in order to describe the impairment of system performance caused by the AND-gate.

6.3.1 Functionality and Non-Ideal Effects

The AND-gate with the chosen transistor sizes is simulated in order to verify that the AND-gate circuit operates satisfactory. The circuit in Figure 6.1 is implemented and simulated in a test bench applying square pulse inputs. The resulting time domain waveforms are presented in Figure 6.2.

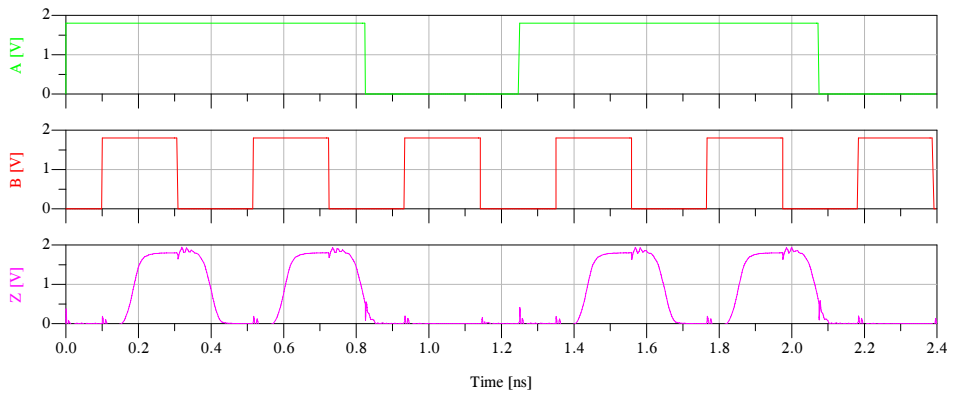


Figure 6.2: *Time domain waveforms of the two input signals and the resulting AND-gate output.* `[DIR020]`

The figure illustrates a successful AND'ing of the two square input signals. The output is delayed approximately 70 ps compared to the clock and the output pulse exhibits a slew rate of 36 V/ns. The output signal is rail-to-rail, since the high and low voltage levels are 1.8 V and 0 V, respectively.

Only the delay from clock signal to AND-gate output is of interest, since an edge of the comparator output signal never occurs during a clock pulse. The timing of the clock signal in the AND-gate is ensured when introducing a delay element, which is designed in the successive chapter.

6.3.2 AND-gate Included in Ideal Setup

To clarify the performance degradation caused by the designed AND-gate, it is included in the ideal simulation setup in ADS. The resulting output spectrum and spectra used as reference are illustrated in Figure 6.3.

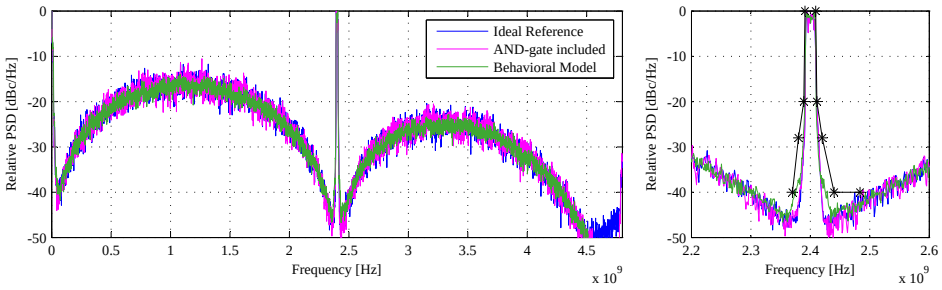


Figure 6.3: *Frequency domain representation of AND-gate output. Blue: Ideal reference simulation in ADS. Magenta: The designed AND-gate included in the ideal simulation. Green: Matlab simulation utilizing the extracted values of non-ideal effects.* `[DIR019]` `[Init 20]`

The figure shows that the designed AND-gate does not degrade system performance. When comparing with the spectrum from the ideal ADS simulation, the two spectra are almost coinciding. The delay and the slew rate of the AND-gate are applied in the Matlab system level simulation. The simulation result agrees with the one from ADS.

6.4 Summary

The design of 2-input CMOS AND-gate displaying a delay of 70 ps and a slew rate of 36 V/ns is presented above. Simulations show no impairment of system performance caused by the designed AND-gate.

This concludes the design of the three main blocks of the $\Delta\Sigma$ Transmitter. In the following chapter the three blocks are joined in the design of the $\Delta\Sigma$ Transmitter.

7 ΔΣ Transmitter Design

Based on the designs in the previous three chapters the sub-circuits are brought together to realize the ΔΣ Transmitter. Moreover, the delay element is designed and gain and cutoff frequency of the filter are settled. Lastly, performance is evaluated and significant imperfections are described.

7.1 Circuit Design

The ΔΣ Transmitter architecture presented in Figure 2.1 on page 7 is realized as illustrated in Figure 7.1.

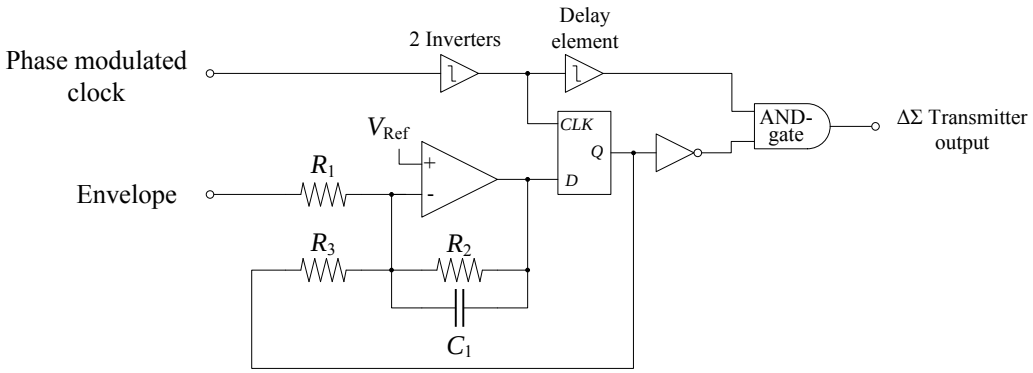


Figure 7.1: *Circuit implementation of ΔΣ Transmitter.*

The degree of feedback from the clocked comparator relative to the envelope input is given by the ratio between R_1 and R_3. In order to realize the Signal Transfer Function (STF) and Noise Transfer Function (NTF) given in Section 1.3, the gain in the ΔΣ Transmitter feedback network must be equal to one, and the forward gain must be much larger than unity. The two conditions are realized if $R_1 = R_3$ and the resistance of R_2 is much larger than R_1. When $R_1 = R_3$, the envelope and the feedback signals are applied equal gain in the filter, which means that the gain of the feedback network is unity and the forward gain of the feedback system is equal to the filter gain, which is given by the ratio between R_2 and R_1. The final dimensioning is performed later. An inverter is inserted between the clocked comparator and the AND-gate. This compensates for the inversion introduced in the loop filter.

In the following, the delay element is dimensioned in order to compensate for the delay introduced in the clocked comparator and the associated inverter.

7.1.1 Delay Element

The delay element must ensure that the timing of the two input signals to the AND-gate is correct. Figure 7.2 illustrates the impact from incorrect timing.

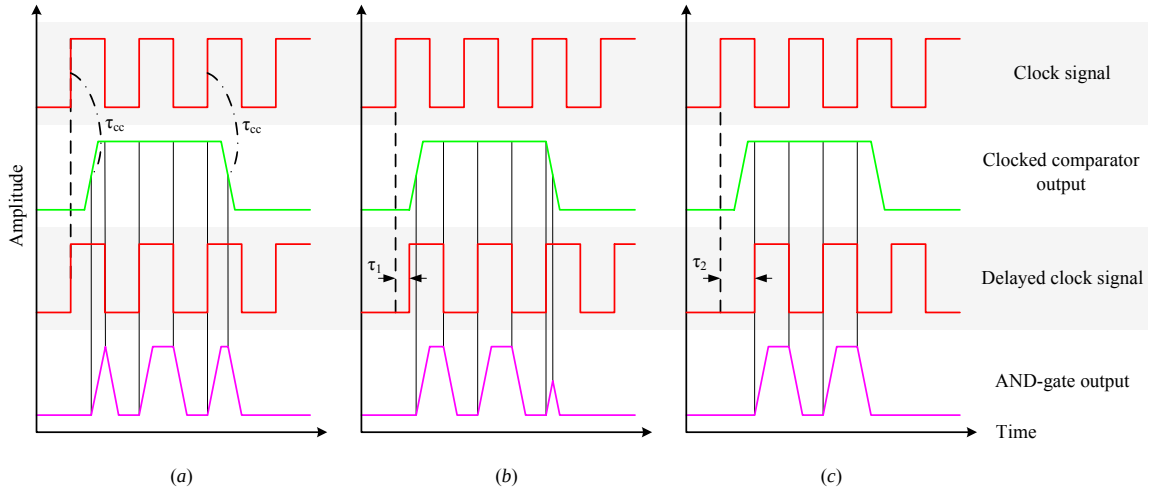


Figure 7.2: *The effect from in-accurate timing of the clock signal used in the AND-gate. (a) The clock to the AND-gate is not delayed, why the output pulse width becomes non-constant. (b) The clock to the AND-gate is delayed τ_{cc}, but the slew rate of the clocked comparator causes non-constant pulse width. (c) The clock signal is delayed τ_2 ensuring that the clocked comparator output is above the threshold level of the AND-gate, which results in pulses of equal widths.*

If the delay is set inadequately, the output of the AND-gate contains pulses with non-constant width, as depicted in Figure 7.2a and 7.2b. This reduces the SNR in the output PSD. In Figure 7.2 it is seen that the delay element must concurrently compensate for the delay and finite SR introduced in the clocked comparator and the added inverter in order to ensure that the output pulses are uniform. Naturally, the delay can be too large, making the clock pulses arrive too late in the comparator output, resulting in incorrect output pulses, equal to the situation depicted in Figure 7.2a.

The delay in the delay element must comply with equation (7.1) [Nielsen and Larsen, 2006].

$$\tau_2 \leq t_{\text{Delay}} \leq \tau_2 + \frac{T_s}{2} \tag{7.1}$$

where τ_2 is the introduced delay from the clock input of the clocked comparator to the input of the AND-gate, while T_s is a clock period.

In Section 5.3 the delay of the clocked comparator itself is determined to approximately 137 ps and simulations show that the delay in a single inverter is 35 ps. Consequently, the delay in the delay element should be between 172 ps and 380 ps. It is chosen to implement the delay using a cascade of inverter pairs, which is appropriate because the implemented delay then depends on the transistors. If the delay in the comparator increases due to process variation, so does the delay in the delay element. Applying six inverters in cascade introduces a delay of 216 ps `[DIR007]` , which complies with (7.1).

However, in the initial design phase, the total delay of the comparator and inverter was determined using square pulses, which resulted in a smaller and constant delay of 132 ps. Complying with (7.1) then only requires four inverters, why erroneously four inverters were integrated, resulting in an insufficient delay of 137 ps `[DIR007]` .

The output spectra of the ΔΣ Transmitter with four and six inverters are illustrated in Figure 7.3.

From the figure it is seen that when the introduced delay is insufficient, the power at higher frequencies is increased while the SNR close the carrier is degraded. Furthermore, the quantization noise power is uneven around the message signal when four inverters are used. This phenomenon is completely removed when using six inverters as delay element.

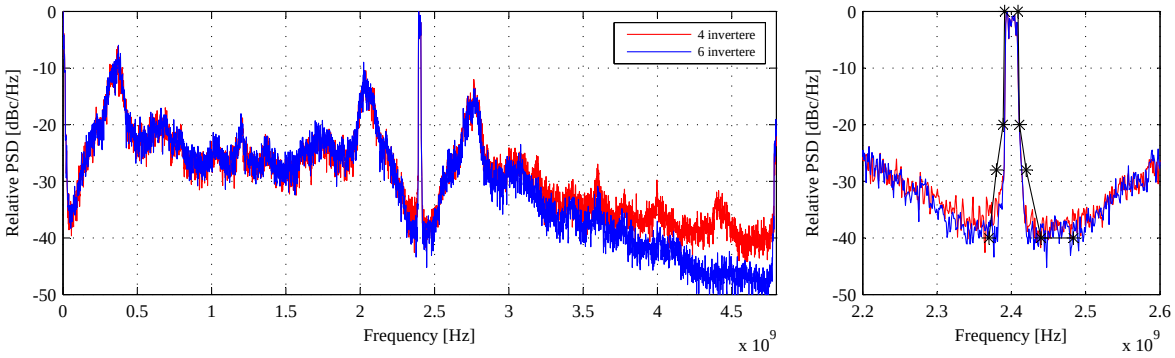


Figure 7.3: *$\Delta\Sigma$ Transmitter output spectrum when four and six inverters are used as delay element. The filter gain is* 25 *and the cutoff frequency is* 1.6 MHz. `[DIR006]`

In the following section the filter gain and cutoff frequency are determined.

7.1.2 Filter Gain and Cutoff Frequency

The final parameters to design are the filter gain and cutoff frequency. In system level simulations it is shown that these parameters are not of decisive importance, however the parameters influence the filter output swing. A large swing is desired as it minimizes the influence of hysteresis in the clocked comparator. However, the filter enters its non-linear region if the voltage swing is too large. Several simulations are performed on the circuit presented in Figure 7.1 with varying sizes of R_1, R_2, R_3, and C_1, for finding the best compromise. Figure 7.4 depicts the output spectra for the three best performing configurations of the filter.

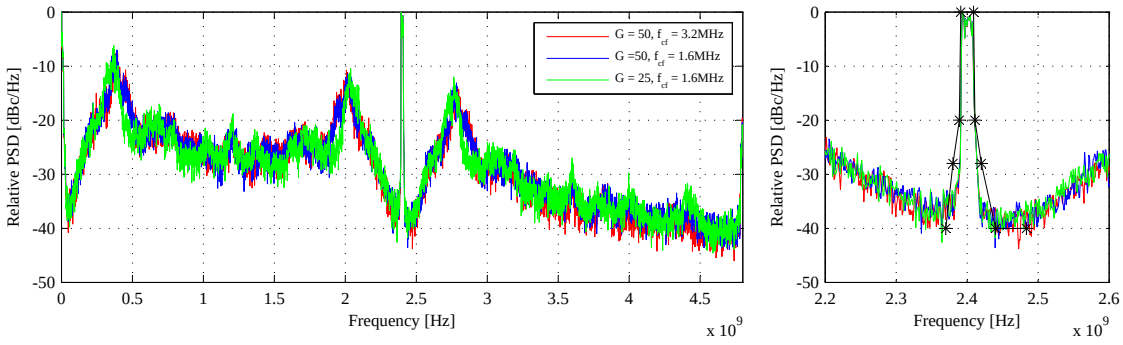


Figure 7.4: *$\Delta\Sigma$ Transmitter output PSD's with filter gain and cutoff frequency as parameters.* `[DIR006]`

The results presented in Figure 7.4 show only a small change in the PSD when varying the mentioned parameters. The final $\Delta\Sigma$ Transmitter is implemented with $R_1 = R_3 = 4\,\text{k}\Omega$, $R_2 = 100\,\text{k}\Omega$, and $C_1 = 1$ pF, resulting in a filter gain of 25 and a cutoff frequency of 1.6 MHz.

The given filter parameters along with all sub-circuits specifications and non-ideal effects of the $\Delta\Sigma$ Transmitter are included in the Matlab behavioral model. The resulting PSD is plotted and on that basis compared with the circuit performance simulated in ADS, as illustrated in Figure 7.5.

The two spectra indicate a good correlation between the implemented circuit and the Matlab behavioral model. The one parameter which is not included in the behavioral model is the metastability of the clocked comparator, and if the closeup PSD is inspected, a tendency for spectral whitening is seen in the ADS circuit simulation. With respect to the WLAN spectrum mask, the performance of the $\Delta\Sigma$ Transmitter is not quite adequate.

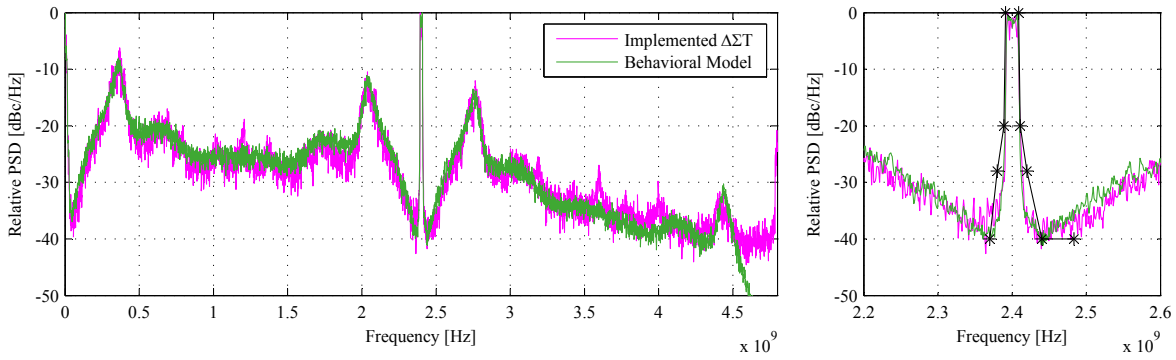


Figure 7.5: *ΔΣ Transmitter output spectra for implemented circuit and corresponding Matlab behavioral model.* `[DIR015]` `[Init 18]`

An investigation of sources of errors is carried out in the following.

7.2 Imperfections

In this section, the imperfections of the designed ΔΣ Transmitter are emphasized. The description is based on Figure 7.6, and references to the figure are denoted by encircled letters.

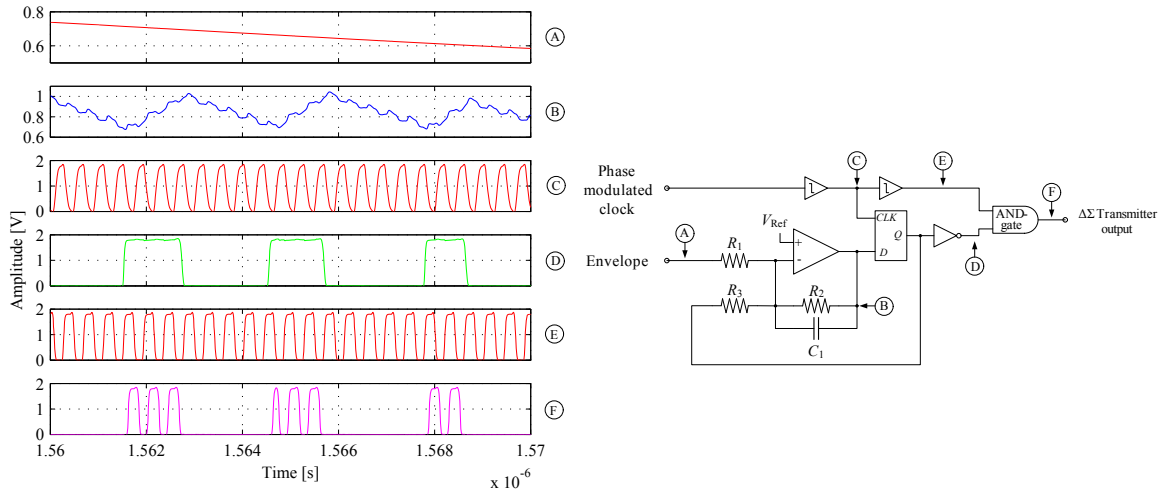


Figure 7.6: *Time domain waveforms of signals in the ΔΣ Transmitter. Signal definitions from top to bottom: Envelope, filter output, comparator clock, inverted comparator output, AND-gate clock, and AND-gate output. To the right, the ΔΣ Transmitter architecture is depicted with encircled letters referring to the time domain waveforms.* `[DIR008]`

The applied envelope signal Ⓐ and the feedback are integrated in the LP filter. The filter output Ⓑ is then sampled at the rising edges of the phase modulated clock Ⓒ in the clocked comparator. The output of the comparator is inverted Ⓓ to compensate for the negative integration in the low pass filter.

By comparing Ⓑ and Ⓓ it is evident that the output of the filter is a negative integration of the feedback pulses. The ripple on the filter output Ⓑ is introduced because the impedance of the comparator changes as a function of the phase modulated clock signal Ⓒ. This phenomenon could enhance the effect from the metastable behavior of the comparator Ⓒ.

The AND-gate delivers the sequences of square pulses Ⓕ, representing the input envelope translated

to the RF frequency. To ensure correct timing of input to the AND-gate, the clock signal is delayed in four inverters Ⓔ to compensate for the delay introduced by the clocked comparator and the inverter. The implemented delay is not always adequate, due to the changing delay in the clocked comparator. In Ⓕ, between 1.564 μs and 1.566 μs, the widths of the three output pulses are non-constant, which moves power away from the RF frequency.

Finally, it can be mentioned, that the modulator is not capable of providing a single pulse at $\Delta\Sigma$ Transmitter output, but always a sequence of two pulses or more. The reason for this deviation is probably caused by the insufficient UGBW or SR of the filter. If the filter is not able to change the output voltage sufficiently fast, the minimum number of pulses is increased.

The calculated sample-wise EVM is 4.48 % and the corresponding amplitude and phase responses are depicted in Figure 7.7.

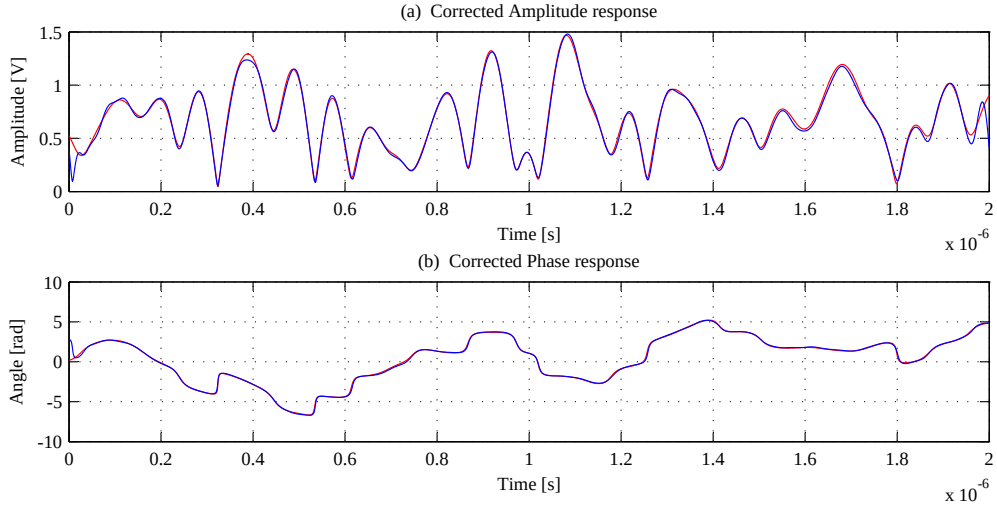


Figure 7.7: *The input phase and envelope, and the down converted output.* `[DIR040]`

The main challenge is not to modulate the input signal and translate it to the RF frequency, but to obtain sufficient attenuation of the quantization noise, which is primarily introduced by ELD and metastability in the clocked comparator.

7.3 Summary

From the circuit design of the filter, clocked comparator, and AND-gate, the design of the $\Delta\Sigma$ Transmitter is carried out. The delay element compensating for the delay in the clocked comparator is dimensioned and cutoff frequency and gain of the filter are settled. Erroneously, the delay in the implemented delay element is insufficient, why simulations show degraded performance. Further simulations conclude that a large delay in the clocked comparator constitutes a significant challenge when complying with the 802.11g transmission spectrum mask. This fact motivates for a modification of the $\Delta\Sigma$ Transmitter architecture. In Appendix C it is shown that the loop delay can be compensated for by using return-to-zero feedback pulses. This knowledge leads to the design of the $\Delta\Sigma$ Transmitter presented in the following chapter.

8 ELD Compensated $\Delta\Sigma$ Transmitter Design

This chapter presents an alternative $\Delta\Sigma$ Transmitter architecture based on compensation of excess loop delay. The concept is described and improved results are presented.

For the design of the $\Delta\Sigma$ Transmitter presented in Chapter 7 it is concluded that the output spectrum does not comply with the transmission mask specified for WLAN. It is emphasized that inadequate performance is partly caused by the large delay in the clocked comparator. The following section provides a short description of how a compensation of the loop delay can be made, which eventually leads to improved performance.

8.1 Excess Loop Delay Compensation

In the following, an ideal modulator, and a modulator that exhibits ELD are described with reference to Figure 8.1 in order to identify how ELD influences the performance of the modulator.

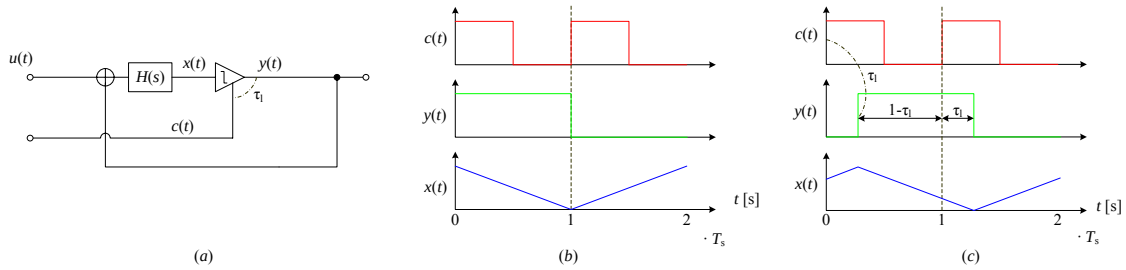


Figure 8.1: *Influence of ELD in time domain. (a) The general $\Delta\Sigma$ Modulator circuit. (b) The time domain waveforms of the modulator with no ELD. (c) The time domain signals for the modulator with ELD.*

Figure 8.1a illustrates a $\Delta\Sigma$ Modulator and nodes of interest, while 8.1b depicts the time domain waveforms in the case where no ELD exists. It is noted that the integration of the feedback signal $y(t)$ has just finished when the sampling instance occurs at one T_s. This is not the case when the modulator exhibits ELD as depicted in Figure 8.1c. The comparator output pulse $y(t)$ is forced into the following clock cycle, causing the input to the comparator at sampling instance to change, which influences the performance of the modulator.

In order to avoid that the feedback pulse is pushed into the adjacent clock period, the architecture depicted in Figure 8.2a is introduced.

The feedback is applied from the output of the AND-gate instead of the output of the comparator, which changes the feedback pulses. The pulses at the output of the AND-gate are high in the first half of the period and return to zero in the last half, why it is referred to as Return-to-Zero (RZ) pulses. Since the width of the pulse is cut by half, the feedback pulse is no longer forced into the adjacent

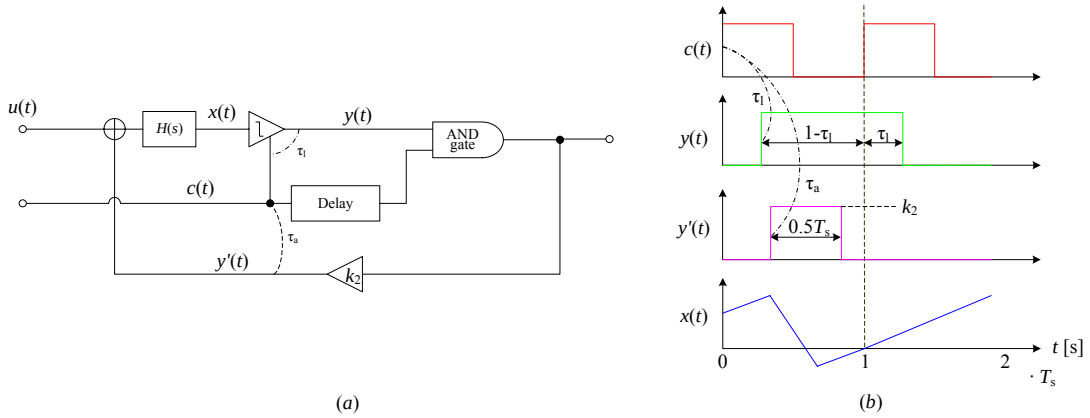


Figure 8.2: *Transmitter architecture compensating for ELD. (a) Architecture and significant nodes. (b) Corresponding time domain waveforms.*

clock period, as seen in Figure 8.2*b*. This eventually cancels out the influence from ELD, as the input to the comparator at sampling instances is equal to that of the system with no ELD.

The extended modulator has a larger delay because of the AND-gate and the gain k_2, which are added in the feedback path. However, as long the total delay τ_a is less than half a clocked period the effect of the overall delay can be removed.

Using RZ pulses reduces the area of the feedback pulses by a factor two. In order to compensate for the reduced area the pulses must be made twice as high, which in theory can be done by setting k_2 equal to two. However, in the implemented solution the pulses cannot attain a height of 3.6 V. Instead, it is necessary to divide the input by two in order to obtain an equivalent result. Consequently, the input is normalized with 0.9 V instead of 1.8 V.

In order to verify that the improvement from applying ELD compensation is sufficient to obtain an SNR of at least 40 dB, a system level simulation including ELD compensation is performed. Specifications from the initial ΔΣ Transmitter are applied in a system level simulation and the feedback is made from the AND-gate and not the clocked comparator. The simulation result is presented in Figure 8.3.

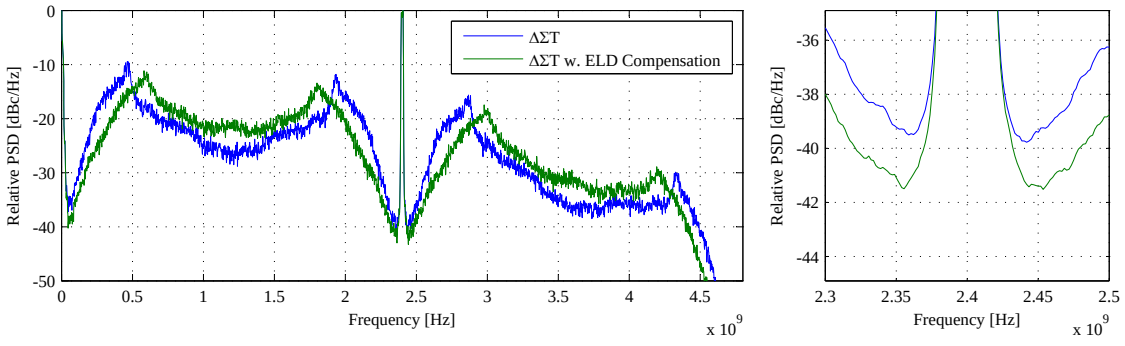


Figure 8.3: *System level simulation of initial ΔΣ Transmitter and alternative architecture compensating for excess loop delay.* `[Init 16]`

From the figure the improvement from ELD compensation is evident, as the quantization noise power density is reduced approximately 2 dB close to the carrier frequency, providing an SNR larger than 40 dB. The applied concept is covered in greater detail in Appendix C.

8.2 Transmitter Architecture

Based on the presented theory and the system level simulation presented above the concept is applied, resulting in the architecture presented in Figure 8.4.

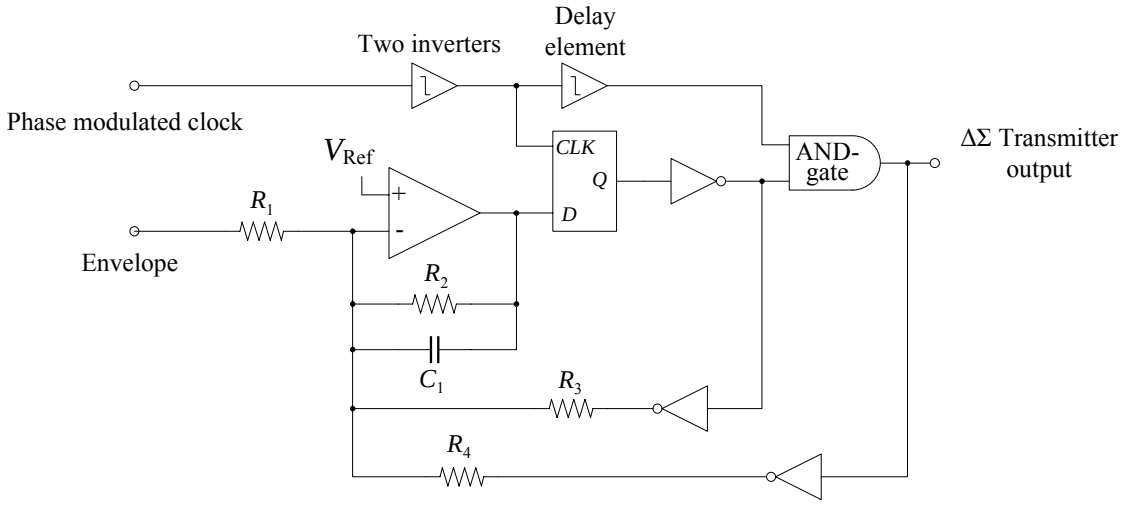


Figure 8.4: *ΔΣ Transmitter architecture employing compensation of excess loop delay.*

As earlier mentioned, the feedback in the modified architecture is applied from the AND-gate. However, simulations reveal that quantization noise is shaped further away from the carrier in the case where a small fraction of the total feedback originates from the clocked comparator `[DIR036]` . Appropriate values for the components are specified later.

In the following the voltage levels of the architecture are discussed.

8.2.1 Voltage Levels

The purpose of this section is to determine the value for the filter reference voltage V_{ref} that makes the ELD compensated ΔΣ Transmitter function as intended. As described in section 8.1, the input should be normalized to 0.9 V, in order to set k_2 equal to two. In the following it is shown that V_{ref} should be 0.9 V. The description is based on Figure 8.5.

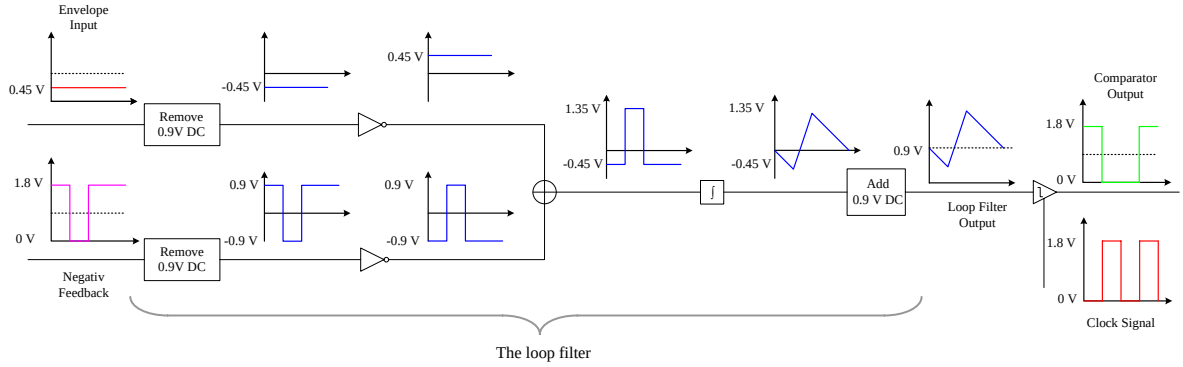


Figure 8.5: *Principle block diagram of operations performed by the ELD compensated ΔΣ Transmitter and corresponding waveforms.* `[DIR034]`

The figure depicts a theoretical block diagram of the loop filter and clocked comparator. For simplicity, the envelope input signal is a DC voltage of 0.45 V. The feedback signal is an inverted version of the

RZ pulses at the AND-gate output, as depicted in Figure 8.4. The reference for the integration is set by V_{ref}, so for illustrative purposes the DC of 0.9 V is removed. The integration in this case is performed around 0 V. Since the filter is designed as an inverting amplifier, both signals are inverted. The two signals are added resulting in RZ pulses between -0.45 V and 1.35 V. The positive integration occurs more rapidly than the negative, since the high pulse represents a larger area per time unit. At the output, the DC is finally added. The integration ends where it started, indicating that the output of the comparator is equally high and low, resulting in a mean voltage of 0.9 V. With an input DC of 0.45 V, the $\Delta\Sigma$ Transmitter consequently applies a gain of two.

Simulation of the envelope input-output relation of the $\Delta\Sigma$ Transmitter is documented in the following. An input DC voltage is increased in steps of 0.1 V in the range from zero to 1.8 V. For an appropriate duration the corresponding mean value of the $\Delta\Sigma$ Transmitter output is calculated. The results from both the initial $\Delta\Sigma$ Transmitter and the one discussed in the present chapter are shown in Figure 8.6.

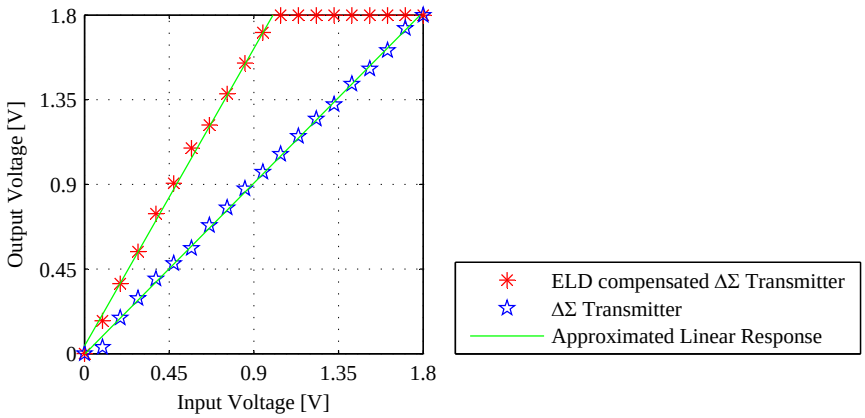


Figure 8.6: *Voltage mapping of the two $\Delta\Sigma$ Transmitters.* `[DIR017]`

From the figure it is seen that both $\Delta\Sigma$ Transmitters perform a linear mapping of the input envelope to the output. For the initial $\Delta\Sigma$ Transmitter the envelope is mapped in the ratio $1:1$, while for the ELD compensated $\Delta\Sigma$ Transmitter the ratio is $2:1$, why the input should be normalized to 0.9 V as described.

In the following section the simulation-based design is documented.

Transmitter Simulation and Design

Circuit level simulations are performed in order to optimize the performance in terms of an increased SNR. In reference to Figure 8.4, the filter is dimensioned such that $R_2 = 100\ \Omega$ and $C_1 = 0.5$ pF yielding a cutoff frequency of 3.2 MHz. With $R_1 = 2$ kΩ, a gain of 50 is realized.

The feedback resistors R_3 and R_4 determine the contribution from each feedback branch. The parallel coupling of the two resistors must be equal to R_1, as this ensures that the area of the feedback pulses of the system utilizing RZ pulses is equal to the non-Return-to-Zero (NRZ) pulses. The overall feedback is weighted such that $R_3 = 40$ kΩ and $R_4 = 2.19$ kΩ.

Besides the passive components, attention must be paid to the delay element. Obtaining improved results with the presented $\Delta\Sigma$ Transmitter involves further constraints when designing the delay element. The issue is clarified in the following.

8.2.2 Delay Element

The delay in the delay element must again ensure uniform square pulses at the output of the AND-gate, but concurrently the delay must be as small as possible to minimize the loop delay. If the delay in the delay element is too large, the ELD cannot be compensated for, or even worse, the performance is further degraded. The scenarios are depicted in Figure 8.7.

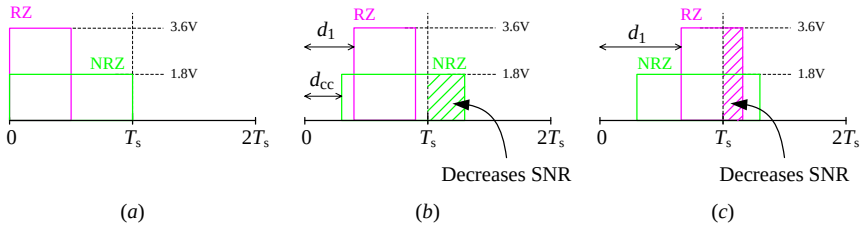


Figure 8.7: *Comparison between NRZ and RZ feedback. (a) In the case where no loop delay exists both NRZ and RZ pulses stay within one clock period. (b) ELD forces the NRZ pulse into the adjacent clock period, while the corresponding RZ pulse stays within the first clock period. (c) A too large delay in the delay element forces the RZ pulse into the adjacent clock period.*

In Figure 8.7a the feedback waveforms of the NRZ and the RZ feedback are illustrated. In the case depicted in 8.7a the loop does not exhibit any delay. The performance of the $\Delta\Sigma$ Transmitter is then not influenced by changing the feedback pulses from NRZ to RZ. In Figure 8.7b the delay d_{cc} is introduced in the clocked comparator, which forces the NRZ pulse into the adjacent clock period impairing the output spectrum. In the situation depicted in Figure 8.7b, the effect from ELD can be removed by changing the feedback to RZ pulses, since the entire RZ pulse is located in the current clock period, due to the fact that the sum of $T_s/2$ and the delay d_1 is less than T_s. The delay d_1 denotes the delay in the delay element plus the propagation delay in the AND-gate plus the delay of the inverter in the feedback path, as seen in Figure 8.4. In Figure 8.7c the duration $T_s/2 + d_1$ is greater than T_s, meaning that the RZ pulse is forced into the next clock period, which again influences the output spectrum. Since the RZ pulse is two times higher than the NRZ pulse, the impairment is more severe.

As previously described, the delay is realized by using an even number of inverters. At least four inverters, which introduce a delay of 137 ps, are necessary. Increasing the number of inverters to six, sets the delay to 216 ps. To settle which size of delay that results in the best ELD compensation, the area of the RZ pulse in the adjacent clock period is computed. The delay d_1, when using four inverters in the delay element, is given by

$$d_1 = d_{4\_inv} + d_{AND} + d_{1\_inv} = 137 \text{ ps} + 70 \text{ ps} + 35 \text{ ps} = 242 \text{ ps} \tag{8.1}$$

where d_{4_inv} is the delay in four inverters, d_{AND} is the delay in the AND-gate, and d_{1_inv} is the delay in a single inverter. When adding the width of the RZ pulse of $T_s/2 = 208$ ps, the RZ pulse is pushed 34 ps into the adjacent clock period. If implementing the delay element with six inverters, the situation is worsened . The delay d_1 is now given by

$$d_1 = d_{6\_inv} + d_{AND} + d_{1\_inv} = 216 \text{ ps} + 70 \text{ ps} + 35 \text{ ps} = 321 \text{ ps} \tag{8.2}$$

where d_{6_inv} is the delay in six inverters. This means that the RZ is forced 113 ps into the adjacent clock period.

In the design presented in Chapter 7, the NRZ pulse is, due to the delay in the clocked comparator, forced 135 ps into the adjacent clock period. Even though the area of the RZ pulse is double per time unit, an improvement is theoretically made when using only four inverters. In the case of six inverters, the area becomes too large.

Conclusively, if the use of RZ pulses is supposed to have any effect, the delay element must by composed of four inverters, which again yields the possibility of introducing a non-constant pulse width of the output signal.

8.2.3 Simulation Results

As previously mentioned, simulations reveal that the best system performance is achieved when applying both feedback from the AND-gate and the clocked comparator. The degrees of feedback and filter gain and cutoff frequency are dimensioned as described earlier and a delay element composed of four inverters is implemented. The simulation result is depicted in Figure 8.8.

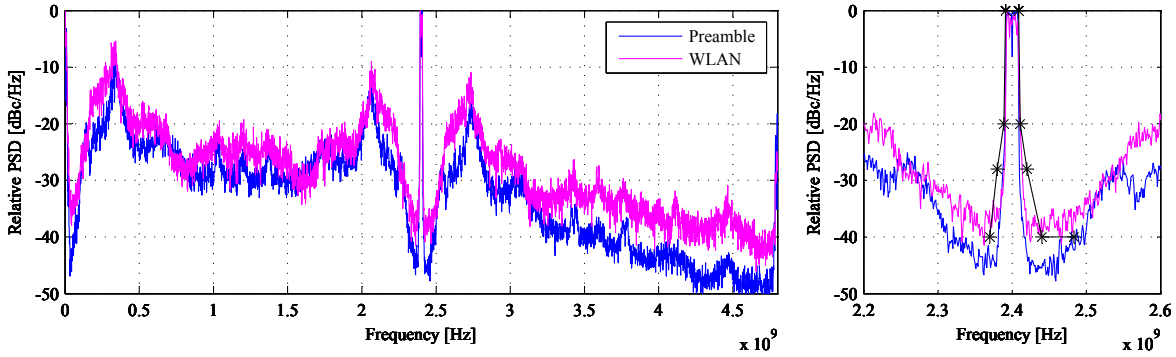


Figure 8.8: *Output PSD for ΔΣ Transmitter compensated for ELD. Two different inputs are applied.* `[DIR021]`

The blue trace represents the simulation result obtained during the design phase. At that point, the applied signal was a WLAN preamble signal, which exhibits a limited signal swing in comparison to a WLAN data signal. Consequently, the design phase aimed at obtaining adequate performance with that given input. More details on the applied input signals are given in Appendix A. From the blue trace the improvement in performance from applying ELD compensation is evident. The noise power density close to the message signal is approximately 5 dB below the required level specified in terms of the spectrum mask. A clear indication that the principle of the ΔΣ Transmitter can be applied for highly power-efficient transmission of a WLAN signal.

The magenta trace in Figure 8.8 represents the case where an actual WLAN data signal is applied as input, in which case performance is decreased considerably. Unfortunately the performance is compromised when the characteristics of the input signal change significantly. In order to obtain adequate performance, the initial design phase should include a WLAN data signal as reference.

In the following section, the imperfections of the final design are discussed.

8.3 Imperfections

This section is designated to emphasize the influence from using RZ pulses in the feedback. The description is based on Figure 8.9, and references to the figure are denoted by circled letters.

Initially, it is seen that the input envelope Ⓐ is quantized by a sequence of pulses with varying width at the comparator output Ⓓ. The pulses are translated to the RF frequency in the AND-gate Ⓕ. Because R_3 is much grater than R_4, the integration is controlled by the sum of Ⓐ and Ⓕ. Each pulse in the feedback Ⓕ, prompts a positive accumulation in the filter output Ⓑ. If Ⓑ and Ⓒ are compared, it is seen that the positive accumulation has ended when the clock comparator samples the input. This is an

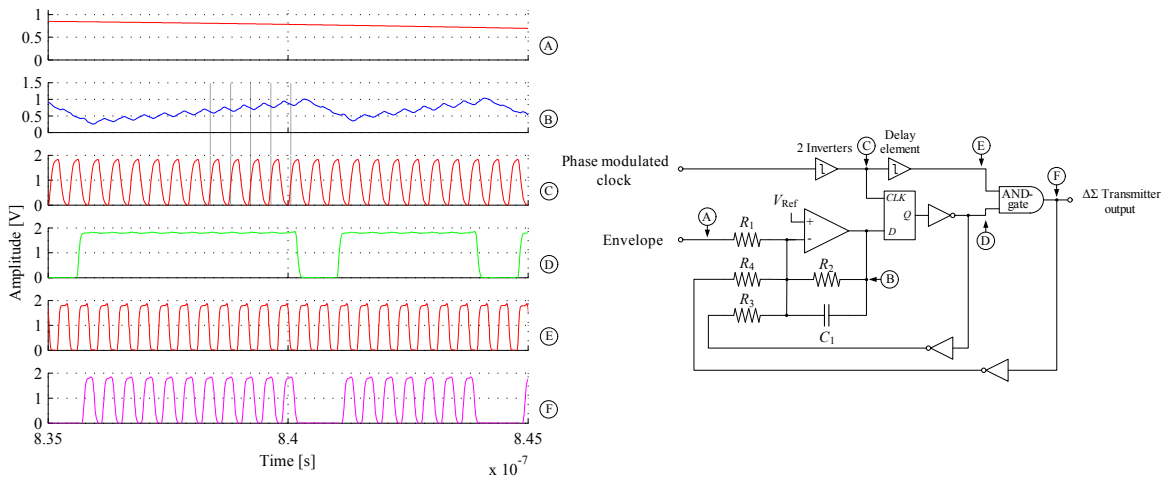


Figure 8.9: *$\Delta\Sigma$ Transmitter and the corresponding signals propagating through the modulator*

indication of that the principle of ELD compensation functions as intended.

The non-constant pulse width introduced by insufficient delay in the delay element has double effect for the presented architecture. The output spectrum is directly influenced by the non-constant pulse width, but also an error is introduced in the feedback loop, which is not the case for the initial $\Delta\Sigma$ Transmitter.

8.4 Summary

Based on the fact that excess loop delay impairs the performance of the $\Delta\Sigma$ Transmitter, a theoretical analysis leads to the design of a $\Delta\Sigma$ Transmitter architecture that compensates for excess loop delay. Simulations show that for a given input signal, the signal performance is improved in terms of an increased SNR of several dB, which eventually means that the modulated output complies with the IEEE 802.11g spectrum mask.

This concludes the design of the second $\Delta\Sigma$ Transmitter. The following chapter documents the process of realizing the integration of the designed circuits. Both $\Delta\Sigma$ Transmitters, along with the main sub-circuits, are integrated in a CMOS chip.

DUE TO CONFIDENTIALITY AGREEMENT WITH UMC,
CHAPTER 9 'SYSTEM INTEGRATION' HAS BEEN REMOVED.

10 Verification

The present chapter serves the purpose of documenting the verification of the circuits integrated on chip. Verification is made in terms of measurements. All measurements are documented in greater detail in Appendix G.

Due to a delayed delivery from the chip manufacturer, the presented verification represents a single measurement iteration. No time has been available for repeating measurements. The following documentation may to some extend reflect this circumstance.

10.1 Inverter

For verifying the performance in terms of gain and delay of the inverter, the S-parameters of the inverter test bench are measured. Detailed documentation on the measurement is found in Appendix G.4.

10.1.1 S-parameter Measurement

S-parameters are measured in order to determine the gain and delay of the inverter. Using a network analyzer and a S-parameter test set, the frequency response of the inverter test bench is measured in the frequency range 200 MHz to 10 GHz at a fixed input power level and DC bias voltage. The result of the measurement is presented in Figure 10.1.

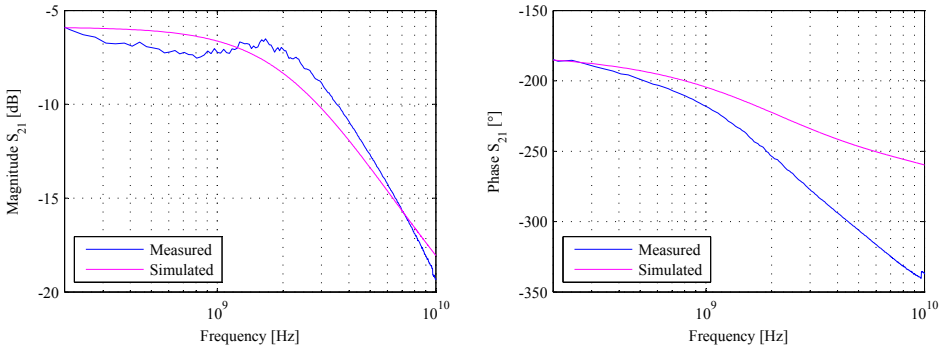


Figure 10.1: *Measured and simulated frequency response of AND-gate.* `[DIR037]`

In order to determine the gain a low power input signal is applied. The measured output is a twentieth of the inverter output. The attenuation is caused by the 1 kΩ on-chip resistor and the 50 Ω characteristic impedance of the measurement setup. Taking this into account an inverter gain of 20 dB is measured for input frequencies up to about 2 GHz. The phase response indicates a large delay of the inverter and at 2.4 GHz the measured delay is 90 ps. The low input power influences the delay of the inverter. However, the simulation result shown in Figure 10.1 is generated with an input power equal to that applied in the measurement, but still, the measured delay is significantly larger than the simulated. A simulation shows that a larger capacitive coupling to ground from the output pad can account for the

increased delay. However, no indications of this effect are found in the layout. Another simulation shows that for an input voltage amplitude of 0.6 V, the delay of the inverter is 29 ps.

10.1.2 Summary

The gain of the inverter is measured to 20 dB while the measured delay is 90 ps at 2.4 GHz. The large delay is partly caused by a small input voltage amplitude. A small input amplitude does not reflect the inverter as an application in the $\Delta\Sigma$ Transmitter, in which the inverter input is mainly square pulses. Also a capacitive coupling to ground contributes to a larger delay, however no indications of such are found in the layout.

10.2 AND-gate

Two measurements are performed on the AND-gate. An initial measurement verifying the functionality using a one tone input signal, and a second measurement determining the delay in terms of S-parameters. Detailed documentation is found in Appendix G.5.

10.2.1 Single Frequency Response

The AND-gate test bench is measured using a signal generator delivering a single tone and a spectrum analyzer measuring the output spectrum. One input of the AND-gate is tied to V_{DD} while the other is applied a fixed frequency of 2.4 GHz. Appropriate levels for input power and DC bias are found. The resulting spectrum in the range 0 to 12 GHz shows significant contributions at 2.4 GHz and its odd harmonics, suggesting that the output consists of square pulses at the frequency of 2.4 GHz as expected.

10.2.2 S-parameter Measurement

S-parameters are measured in order to determine the delay of the AND-gate. Using the network analyzer and the S-parameter test set, the frequency response of the AND-gate test bench is measured in the frequency range 200 MHz to 10 GHz at a fixed input power level and DC bias voltage. The result of the measurement is presented in Figure 10.2.

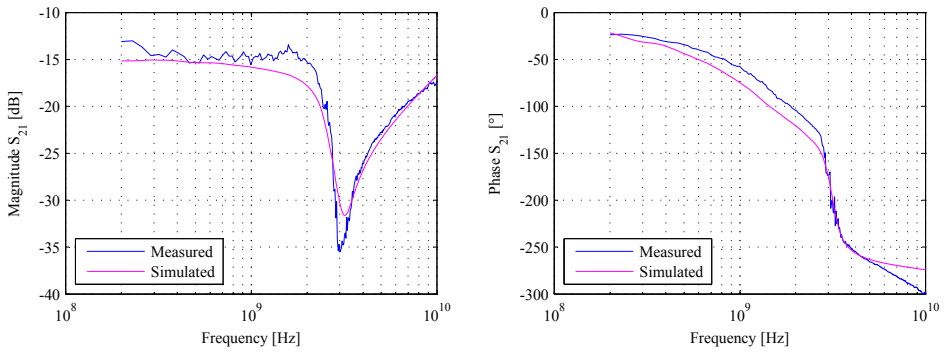


Figure 10.2: *Measured and simulated frequency response of AND-gate.* `[DIR038]`

The measured output is a twentieth of the AND-gate output. The attenuation is caused by the 1 kΩ on-chip resistor and the 50 Ω characteristic impedance of the measurement setup. This reduces the output

with 26 dB indicating a gain of the AND-gate of approximately 10 dB. The fact that the gain of the AND-gate is larger than 0 dB means that the input voltage in the measurement is not rail-to-rail. The low input voltage accounts for the early drop in the magnitude response between 2 GHz and 3 GHz. From the phase response, a delay of 144 ps at 2.4 GHz is measured. The immediate large delay is likewise caused by the reduced input voltage.

Capacitive Coupling

As seen in Figure 10.2, the frequency response changes unexpectedly above 3 GHz. Investigating the layout of the test bench reveals that metal routed in ME5 causes a capacitive coupling between input and output pads as illustrated in Figure 10.3.

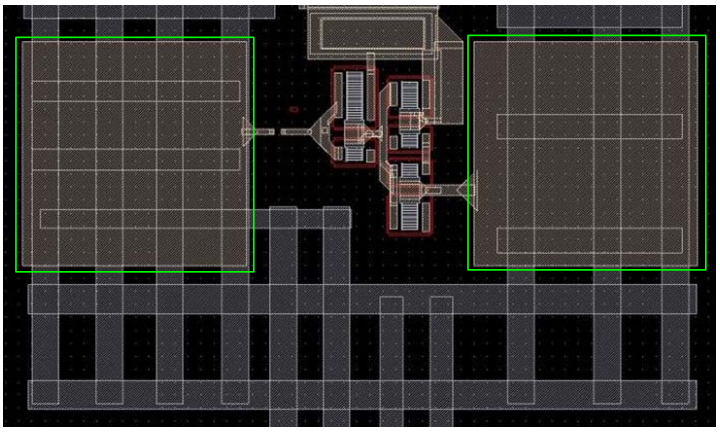


Figure 10.3: *Layout of AND-gate test bench illustrating capacitive coupling between input and output pads.*

The coupling is represented by a 60 fF capacitor in the simulation of the AND-gate measurement. The result is presented along with the measured data in Figure 10.2. The two results agree, which indicates that the inappropriate behavior of the AND-gate is caused by the capacitive coupling between input and output in the layout.

10.2.3 Summary

The functionality of the AND-gate is verified through two measurements. The delay of the AND-gate is measured to 144 ps at 2.4 GHz. A relatively large delay, which is caused by insufficient input power. In the layout of the AND-gate test bench, a capacitive coupling between the input and output pads is found. This compromises the functionality at high frequencies, but since the coupling only exists in the test bench, the $\Delta\Sigma$ Transmitter is unaffected.

10.3 Clocked Comparator

Two measurements are intended for verifying the implementation of the clocked comparator. An initial measurement is performed in order to verify the functionality, and furthermore a more comprehensive measurement is carried out with the purpose of determining the ELD of the clocked comparator. Details on the performed measurements are found in Appendix G.6.

10.3.1 Functionality Test

The measurement is performed using two signal generators and a spectrum analyzer. A 1.2 GHz and 2.4 GHz signal are applied in a DC offset of 0.9 V. The resulting output spectrum in the frequency range 1 GHz to 6.5 GHz is shown in Figure 10.4.

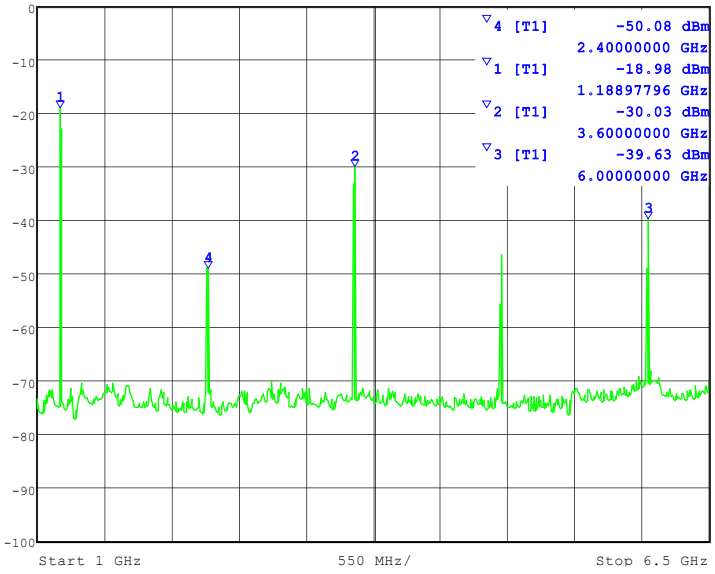


Figure 10.4: *Clocked comparator output spectrum.* `[DIR039]`

The significant contribution at 1.2 GHz and the odd harmonics indicate that the output time domain waveform is a 1.2 GHz square signal, as expected. A simulation of the measurement setup supports this conclusion.

10.3.2 Delay Measurement

The measurement of the ELD has not been performed; most likely due to limitations of the measurement equipment. However, the measurement procedure is verified through simulation. Within the available time frame, it has not been possible to obtain a distinct indication of whether the practical measurement setup allows for a determination of the delay in the clocked comparator.

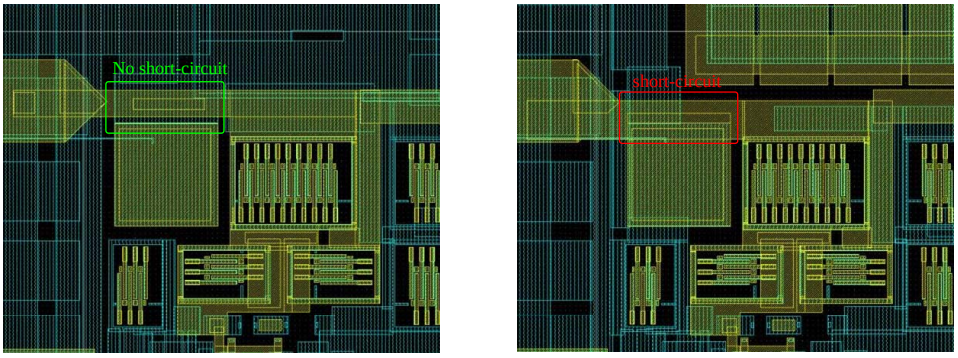
10.3.3 Summary

From the performed functionality test it is verified that the comparator functions as intended. No measurement of the delay is performed.

10.4 Filter

Probing on the filter test bench shows that an inappropriately large current is drawn from the power supply. By reviewing the Cadence layout, the source of error is located as described in the following in reference to Figure 10.5.

Figure 10.5a shows the layout at the first tapeout. The yellow path entering from the left is the opamp V_{DD} routed in ME2. The square just before the transistor is a connection to a decoupling capacitor



(*a*) (*b*)

Figure 10.5: *Two layout iterations of the filter. (a) Submitted for first tapeout. (b) Submitted for final tapeout.*

in the upper layers. The metal present in ME2 is connected to the bottom of the capacitor and to ME1 ground layer. Figure 10.5*b* illustrates the final submission. The ME2 carrying the opamp V_{DD} is connected to the ME2 underneath the capacitor, which causes a short circuit to ground. The error was made in between two submissions in connection with the re-routing of the MIM capacitors, as described in Section 9.3.2.

With the purpose of removing the short circuit, laser cutting has been performed. However, it has not possible to remove the short circuit between ME2 and ME1. For this reason no measurements are performed on the filter test bench.

Unfortunately, the layout of the filter is created and edited in a sub-circuit. The sub-circuit is included in the test bench as well as the two ΔΣ Transmitters. Consequently, the error is replicated in the two ΔΣ Transmitters.

10.5 ΔΣ Transmitters

As explained above, the short circuit in the filter makes it impossible to carry out the planned verification of the two ΔΣ Transmitters. Instead, probing is performed on the ΔΣ Transmitter, accepting the relatively large current of approximately 200 mA drawn from the supply.

A DC input and an RF clock are applied, while the output is connected to a spectrum analyzer. The resulting spectrum displays a significant contribution at the RF frequency and its odd harmonics.

As documented in Appendix F, the power supply of each sub-circuit in the ΔΣ Transmitters are isolated from one another. This means that even though the supply of the filter is shorted, V_{DD} is still applied on the clocked comparator, AND-gate, and various inverters. As the output from the filter is constantly low, the clocked comparator output is also low. Consequently, the input to the AND-gate is constantly high due to an inversion. The clock input is because of that mapped directly to the output of the AND-gate, which explains the measured behavior. This verifies that the sub-circuits in the ΔΣ Transmitters, besides the filter, exhibits proper functionality.

10.6 Summary

In the present chapter the inverter, AND-gate, and clocked comparator are verified through measurements. All measurement data matches sufficiently the data obtained from simulations replicating the measurement setups. A short-circuit is found in the filter, why the performance of this sub-circuit can-

not be verified. This also applies for the two $\Delta\Sigma$ Transmitters, as they include the filter as a sub-model.

This concludes the verification of the implemented circuits. The following chapter is dedicated to discussion of various issues related to the performance of the $\Delta\Sigma$ Transmitters.

11 Discussion

This chapter presents a discussion of issues that relate to the design of the two $\Delta\Sigma$ Transmitters. Mainly issues that limit system performance are addressed.

11.1 Clocked Comparator

The performance of the $\Delta\Sigma$ Transmitter is to a great extend related to the performance of the clocked comparator. Various issues related to the clocked comparator are discussed in the following.

11.1.1 Excess Loop Delay

The clocked comparator presented in this work realizes the functionality of a positive edge-triggered D-flip-flop using only 14 transistors. The clocked comparator introduces a delay of 137 ps or one third of a clock period at 2.4 GHz.

System level simulations show that the introduced delay causes an impairment of system performance. The influence of the delay in the clocked comparator is reduced by applying feedback from the AND-gate. Theory shows that it is possible to reduce the effect from ELD, however the delay in the designed sub-circuits do not allow for a complete equalization of the impairment. This fact makes it relevant to reduces the delay of the clocked comparator, why the issue would constitute a top priority in a revised design.

11.1.2 Metastability

Metastability of the clocked comparator causes the degree of delay and hysteresis to depend on the input signal. This leads to a random variation in the output pulse width, which eventually results in a whitening of the baseband spectrum, and hence a degradation of SNR [Cherry and Snelgrove, 2002, p. 138].

In order to resolve the issue of metastability, the input signal to the clocked comparator should be as large as possible. Ideally, the input should be a square waveform. This cancels out the effect of hysteresis all together and the delay would be constant.

Two immediate solutions exist; (i) increasing the gain and UGBW of the filter and/or (ii) implementing a preamplifier between the filter and clocked comparator. The first solution creates a larger signal swing at the filter output, while the second ensures a larger signal swing at the input of the clocked comparator and concurrently ensures that the filter output swing is small, which minimizes the added distortion. However, a large delay in the preamplifier could undermine its justification.

11.1.3 Return-to-Zero Behavior

When compensating for excess loop delay, the RZ pulses from the AND-gate are applied in feedback. If instead the RZ pulses are delivered directly by the clocked comparator, the additional delay introduced by the AND-gate could be removed, leading to further improvements. Another advantage of this modification is that the need for the disputed delay element is eliminated.

11.2 Variable Delay Element

The delay element in the present work is implemented using a cascade of four inverters. The immediate advantage of implementing a transistor based delay, instead of one based on passive components, such as an RC delay element, is that any change of delay due to process variations equals the change of delay in the clocked comparator. If the transistors in the clocked comparator exhibits a delay longer or shorter than designed for, the transistors in the delay element will be equally slower or faster.

The disadvantage is that the delay can only be an even multiple of the delay in an inverter. In the final design stage this proved to be a serious limitation. Especially in consideration to the $\Delta\Sigma$ Transmitter utilizing ELD compensation, where the delay advantageously can be set to the lower threshold of Equation (7.1) on page 50.

Due to the benefit from a minimum delay in the delay element, a tunable delay is preferred rather than a fixed delay as for the inverters. A tunable delay would make it possible to ensure uniform pulse width at the output of the $\Delta\Sigma$ Transmitter and concurrently the delay can be tuned to just meet the lower threshold of (7.1), which eventually minimizes the total delay in the loop.

11.3 Integration Technology

The speed of the 0.18 μm CMOS technology is a constraining factor in relation to the performance of the presented architecture. Much time has been allocated for optimizing the UGBW of the opamp, and to reduce the influence of the delay in the clocked comparator. Applying a faster technology, less optimization of the sub-circuits is necessary, leaving more time to optimize the $\Delta\Sigma$ Transmitter and its behavior. However, two immediate drawbacks exist if choosing a faster CMOS technology; (i) the smaller scale of transistors reduces the supply voltage, and (ii) manufacturing costs are increased.

11.4 Mismatch between the phase and envelope

An issue which is not covered in the present work, is the mismatch between the phase and envelope at the transmitter output. When designing a polar modulator it is important that the phase and envelope are synchronized in time [Larsen, 2006, mm 5, slide 25]. In order to synchronize phase and envelope, the input phase must be appropriately delayed. The size of the delay must be equal to the delay from the input of the $\Delta\Sigma$ Transmitter to the output of the AND-gate, where the delay in the clock path is taken into account.

System level simulations show that the introduced mismatch in the $\Delta\Sigma$Transmitters has little influence on both the EVM and the spectra, since the delays in the clock path and envelope path are very similar. Due to the limited impairment caused by mismatch, the motivation for performing a compensation is negligible for the presented architecture.

11.5 $\Delta\Sigma$ Transmitter versus $\Delta\Sigma$ ADC

Characterization of the $\Delta\Sigma$ Transmitter has been in great focus in this thesis. That is, characterizing the parameters which influence the performance and to what extend. For this analysis, literature on $\Delta\Sigma$ ADC theory is partly used as reference. From the performed system and circuit level simulations it is concluded that requirements specified when designing a $\Delta\Sigma$ Transmitter are equal to those of a $\Delta\Sigma$ ADC. Consequently, designing a $\Delta\Sigma$ Transmitter is in principle no different from designing a $\Delta\Sigma$ ADC, why the comprehensive literature and guidelines on the subject should be used intensively in the design of the $\Delta\Sigma$ Transmitter.

12 Conclusion

The primary objective of this master's thesis is to prove the concept of the $\Delta\Sigma$ Transmitter architecture in an integrated solution, which enables highly power-efficient transmission of varying envelope signals.

The objective is approached through comprehensive analysis of the effects of decisive importance for system performance, utilizing system level behavioral models. The analysis is followed by design and implementation of the architecture in the 0.18 μm CMOS technology. Finally, system performance is verified through measurements.

Initially, the $\Delta\Sigma$ Transmitter is implemented in Matlab on system level incorporating behavioral models of the sub-circuits in the architecture. This allows for time-efficient evaluation of the significance of non-ideal effects. Attention is devoted to the effects that in the literature are known to impair system performance. The system level simulations constitute a reference for the following circuit level design.

Circuit level design is carried out for the filter, clocked comparator, and AND-gate. The isolated impact of each sub-circuit is evaluated by including the sub-circuits one by one in an ideal $\Delta\Sigma$ Transmitter simulation setup in ADS. Similar simulations are likewise carried out on system level in Matlab, where the non-ideal effects for each sub-circuit are included in its corresponding Matlab behavioral model. Simulation results are compared and to a significant extent coinciding. This provides a distinct capability of modeling the specific behavior of the $\Delta\Sigma$ Transmitter sub-circuits.

The sub-circuits are joined in the design of the $\Delta\Sigma$ Transmitter. System simulations show an inadequate performance in terms of complying with the transmission mask, specified in the IEEE 802.11g WLAN standard. As for the sub-circuits, the behavior of the $\Delta\Sigma$ Transmitter is modeled on system level in Matlab with a satisfactory result.

In the preliminary analysis and design phase, evaluation of sub-circuit performance indicates that the delay introduced in the clocked comparator constitutes the most significant contributor in relation to impairment of overall system performance. The architecture is modified, changing the feedback strategy and hence reducing the influence of the delay of the clocked comparator. Simulation results conclude that improved performance is obtained for a specific input.

Finally, the two designed $\Delta\Sigma$ Transmitters are implemented in an integrated circuit. In addition, the three main sub-circuits and an inverter are individually included in the integration, enabling verification of each sub-circuit.

On-wafer measurements are performed in order to verify the performance of the system. Initial measurements reveal that a short-circuit exists in the two $\Delta\Sigma$ Transmitters, preventing a satisfactory verification. Measurements on the filter sub-circuit and review of the corresponding layout leads to the conclusion that the short-circuit originates from this sub-circuit.

In order to verify the remaining sub-circuits, measurements are performed. In addition the measurement setup is simulated validating the measurement results.

The functionality of the clocked comparator is verified based on measurements and simulations. Likewise, the functionality of the AND-gate is verified. From measured S-parameters, the delay of the

AND-gate is determined. The result agrees with that of the simulation. A similar measurement is performed on the inverter and a deviation in terms of a larger delay is observed, in comparison to simulation results.

The concept of the $\Delta\Sigma$ Transmitter presented in this master's thesis is not proved in an integrated solution. Due to an error in the integrated circuit layout, a final verification is not performed. However, the correlation between the measurements and corresponding simulations, indicates that the simulated performance of the $\Delta\Sigma$ Transmitter architecture can be replicated in a measurement.

This leads to the conclusion that the principle of transmitting varying envelope signals using an integrated $\Delta\Sigma$ Transmitter is possible.

List of Acronyms

BP	Band Pass
BW	bandwidth
CMOS	Complementary Metal-Oxide Semiconductor
CG	Common Gate
CS	Common Source
DUT	Device Under Test
DRC	Design Rule Check
ELD	Excess Loop Delay
ESD	Electrostatic Discharge
EVM	Error Vector Magnitude
FET	Field Effect Transistor
GSG	ground-signal-ground
ISS	impedance standard substrate
LP	Low Pass
LVS	Layout-Versus-Schematic
ME1	metal layer 1
ME2	metal layer 2
ME5	metal layer 5
ME6	metal layer 6
MIM	Metal-Insulator-Metal
MMC	Metal/Metal Capacitor
NRZ	non-Return-to-Zero
NTF	Noise Transfer Function
opamp	operational amplifier
OSD	open/short de-embedding
OSR	oversampling ratio
PCB	printed circuit board

RF	radio frequency
RNHR-RF	HR Non-silicided Poly resistor
RISC	RF Integrated Systems and Circuits
RZ	Return-to-Zero
STF	Signal Transfer Function
SMD	Surface Mounted Device
SNR	Signal-to-Noise Ratio
SOLT	short-open-load-thru
SR	Slew Rate
UGBW	Unity Gain Bandwidth
UMC	United Microelectronics Corporation

Bibliography

Agilent.
Applying Error Correction to Network Analyzer Measurements.
Agilent Technologies, 2002.
http://cp.literature.agilent.com/litweb/pdf/5965-7709E.pdf.

BHE, 2007.
http://www.bhe.dk/showpage.php?pageid=429.

P. Boonyaporn and V. Kasemsuwan.
A High Performance Class AB CMOS Rail-to-Rail Voltage Follower.
IEEE, 2002.

K. T. Chan and K. W. Martin.
Components for a GaAs Delta-Sigma Modulator Oversampled Analog-to-Digital Converter.
IEEE, 1992.

J. A. Cherry and W. M. Snelgrove.
Continuous-Time Delta-Sigma Modulators for High-Speed A/D conversion.
Kluwer Academic Publishers, 2002.
ISBN : 0-792-38625-6.

I. Doerr, L.-T. Hwang, G. Sommer, and H. Oppermann.
Parameterized Models for a RF Chip-To-Substrate Interconnect.
IEEE, 2001.

Y. Geerts, M. Steyaert, and W. Sansen.
Design of multi-bit Delta-Sigma A/D Converters.
Kluwer Academic Publishers, 2002.
ISBN : 1-4020-7078-0.

A. Gosslau and A. Gottwald.
Optimization of a Sigma-Delta Modulator by the use of a Slow ADC.
IEEE, 1988.

Q. Huang and R. Rogenmoser.
Speed Optimization of Edge-Triggered CMOS Circuits for Gigahertz Single-Phase Clocks.
1996.
IEEE Journal of Solid-State Circuits, Vol 31, No. 3, pp. 456 - 465.

IEEE.
IEEE, 802.11a.
IEEE Standard for Information technology, 2003.

IEEE.
IEEE, 802.11g.
IEEE Standard for Information technology, 2003.

O. K. Jensen.
Viewgraphs for Microwave Theories and Techniques mm6.
RISC group at Aalborg University, 2007.
http://cpk.auc.dk/risc/lecture/9sem-2006/RISC9-2/mm.6.slides.pdf.

T. E. Kolding.
On-Wafer Measuring Techniques for characterizing RF CMOS Devices.
Ph.D Dissertation, 1999.
RISC Group, Aalborg University.

T. Larsen.
Viewgraphs for Transceiver Fundamentals mm1-6.
RISC group at Aalborg University, 2006.
http://cpk.auc.dk/risc/lecture/8sem-2006/RISC8-2.

T. Lehmann and M. Cassia.
1-v power supply cmos cascode amplifier, 2001.
IEEE Journal of Solid-State Circuits, Vol 36, No. 7, pp. 1082 - 1086.

M. Nielsen and T. Larsen.
A Transmitter Architecture Based on Delta-Sigma Modulation and Switch-Mode Power Amplification.
Accepted for publication in TCAS-II, 2006.

A. V. Oppenheim and R. W. Schafer.
Discrete-Time Signal Processing.
Prentice-Hall, Inc., 1999.
ISBN: 0-13-083443-2.

H. W. Ott.
Noise reduction techniques in electronic systems.
John Wiley & Sons, LTD., 1988.
ISBN: 0-471-85068-3.

H. S. . J. S. Pervez Aziz.
An overview of sigma-delta converters, 1996.
IEEE Signal Processing Magazine, pp. 61 - 74.

B. Razavi.
Design of Analog CMOS Integrated Circuits.
McGraw-Hill, 2001.
ISBN: 0-07-118839-8.

A. Sedra and K. Smith.
Microelectronic Circuits.
Oxford University Press, 1998.
ISBN: 0-19-511690-9.

M. R. Spiegel and J. Liu.
Matematical Handbook of formulas and tables.
McGraw-Hill, 1999.
ISBN: 0-07-038203.

UMC.
Umc 0.18μm 1p6m salicide mixed-mode/rf cmos model, rev. 2d2.
Component Schematic Vs. Layout Descriptions, 2005a.

UMC.
Umc 0.18μm mixed mode and rfcmos 1.8v 1p6m mmc process electrical design rule, v1.4.
Electrical Design Rules, 2005b.

UMC.
Umc 0.18μm mixed mode and rfcmos 1.8v 1p6m mmc process topological layout rule, v2.7.
Topological Layout Rules, 2005c.

UMC.
Umc 0.18μm 1p6m salicide mixed-mode/rf cmos model, rev. 2d2.
RF CMOS Model Descriptions, 2003.

J. F. Wakerly.
Digital Design - Principles & Practices.
Prentice Hall, 2001.
ISBN: 0-13-089896-1.

M. Wang and E. Sánchez-Sinencio.
Simple CMOS Low-Voltage Op Amps with Constant-Gm Rail-to-Rail Input Stage.
IEEE, 1997.

N. H. Weste and D. Harris.
CMOS VLSI Design - A Circuits and Systems Perspective.
Pearson Education, Inc., 2005.
ISBN: 0-321-26977-2.

J. Yuan and C. Svensson.
High-speed cmos circuit technique, 1989.
IEEE Journal of Solid-State Circuits, Vol 24, No. 1, pp. 62 - 70.

IEEE 802.11 WLAN Standard

A

This appendix presents the communication standard that is used as reference for the design of the $\Delta\Sigma$ Transmitter architecture. Focus is aimed at the transmission spectrum mask, since complying with this, is known to comprise a great challenge of the specific architecture. Finally, signals used as simulation input are presented.

In the following, key specifications of the IEEE 802.11g WLAN Standard are presented. This is followed by the transmission spectrum mask, which is used throughout the report as a reference when evaluating system performance.

Key Specifications for WLAN 802.11g

In February 2003 IEEE established the 802.11g WLAN standard, which offers up to 54 Mbps of data rate in the frequency band $2.4 - 2.4835$ GHz . It is a merged standard of 802.11a and 802.11b and supports modulation schemes such as BPSK, OFDM and 64-QAM. WLAN 802.11g offers 7 channels of each 20 MHz bandwidths. In practice only 3 channels can be used simultaneously, which results in a total frequency bandwidth of 83.5 MHz (3 non-overlapping channels). [IEEE, 802.11g, p. 30 sec:19.5.4] which refers to [IEEE, 802.11a, pp. 29-30]

Transmission Spectrum Mask

The IEEE 802.11g WLAN transmission spectrum mask is depicted in Figure A.1.

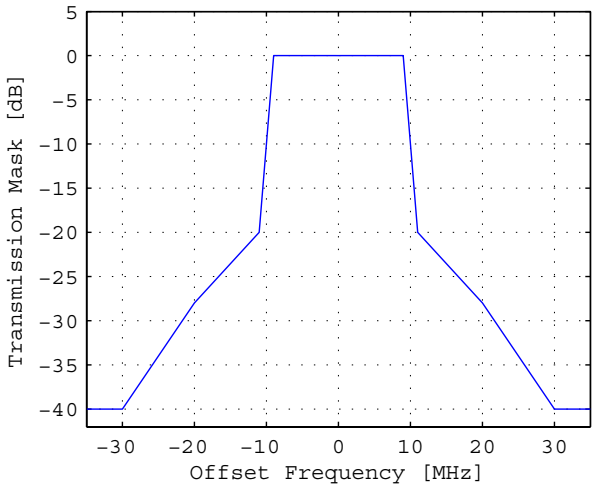


Figure A.1: *IEEE 802.11g transmission spectrum mask.*

Applied WLAN Signals

For simulation purposes, two types of WLAN signals are applied. A duration of 5 μs of each signal is depicted in Figure A.2.

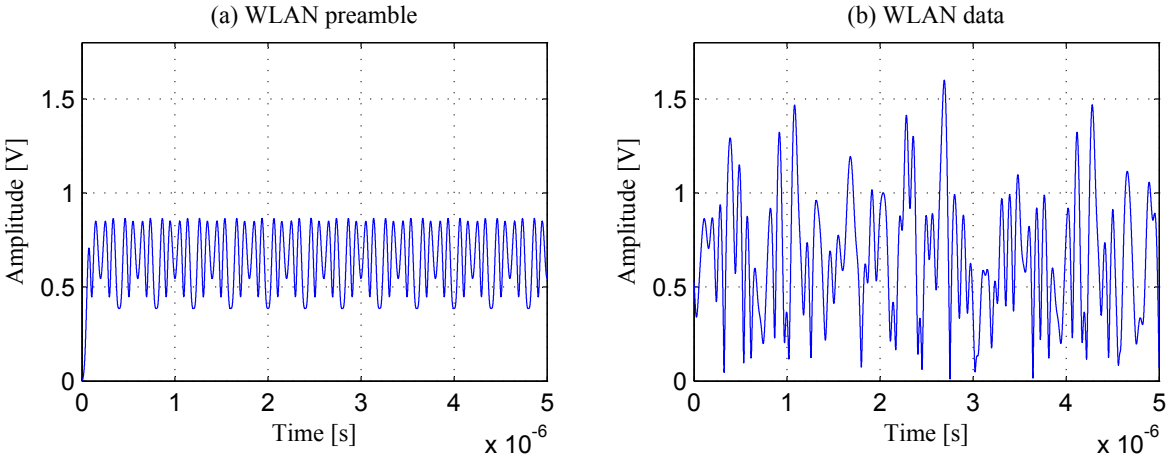


Figure A.2: *Applied WLAN signals. (a) WLAN preample. (b) WLAN data.*

The WLAN preample is periodic with a limited swing compared to the WLAN data signal. All documented simulations are performed using the WLAN data signal, except for the one presented in Figure 8.8 related to the design of the ELD compensated $\Delta\Sigma$ Transmitter.

B FET Building Blocks

B.1 Cascoding

The Common Source (CS) amplifier is capable of delivering a voltage gain of roughly 20 in the $0.18\ \mu$m CMOS technology. By applying a cascode device to the amplifier, the gain of the amplifier is almost squared. The cascade of a CS stage and a Common Gate (CG) stage is a cascode topology, in which the CG stage is defined as the cascode device and the CS stage is defined as the input device. [Razavi, 2001, pp. 52, 83, 87]. The topology is depicted in Figure B.1a.

In the following section, the principle is described in general terms using the small signal AC equivalent.

B.1.1 Output Impedance of the Cascode Stage

The cascode amplifier attains the voltage gain improvement by increasing the output impedance of the amplifier. When calculating the output impedance the small signal AC equivalent, depicted in Figure B.1b, is applied.

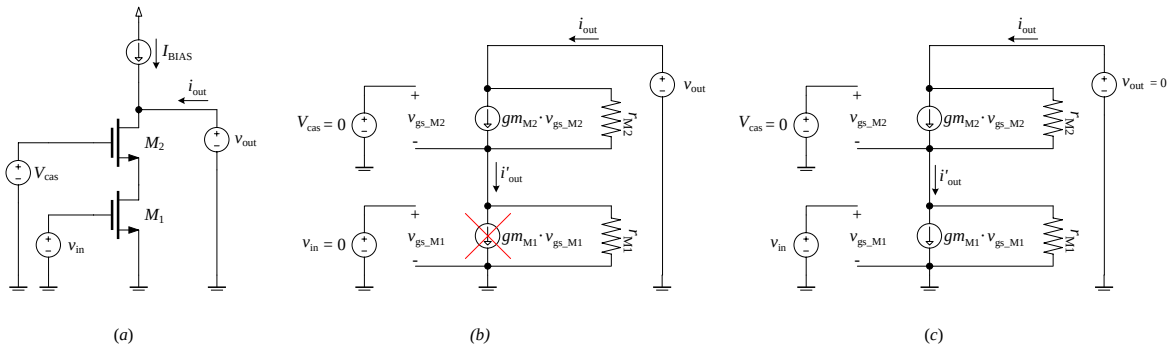


Figure B.1: *Cascode topology and circuit equivalents. (a) A cascode stage composed of the input device M_1 and the cascode device M_2. (b) Small signal AC equivalent applied for calculating circuit output impedance R_{out}. (c) Small signal AC equivalent applied for calculating circuit transconductance G_m.*

The two voltages V_{cas} and v_{in} are shorted to ground and the current source I_{BIAS} is replaced by an open. The first thing to notice is that v_{gs_M1} is zero, which means that the current through the voltage controlled current source is also zero. The result is that the resistor r_{M1} drains the current i_{out}, which makes it easy to calculate v_{gs_M2}.

$$v_{gs\_M2} = -i_{out} \cdot r_{M1} \tag{B.1}$$

From (B.1), the current drained by M_2 can be expressed, which makes it possible to calculate the current

through r_{M2}, because i_{out} is shared between the resistor and the voltage controlled current source.

$$\begin{aligned} i_{r\_M2} &= i_{out} - gm_{M2} \cdot V_{gs\_M2} \\ &= i_{out} \cdot (1 + gm_{M2} \cdot r_{M1}) \Big|_{v_{gs\_M2} = r_{M1} \cdot i_{out}} \end{aligned} \tag{B.2}$$

The current through the two resistors r_{M1} and r_{M2} is now known, which makes it possible to calculate both V_{out} and R_{out} [Razavi, 2001, p. 87 eqn. 3.119].

$$\begin{aligned} v_{out} &= r_{M1} \cdot i_{out} + r_{M2} \cdot i_{r\_M2} \\ R_{out} &= \frac{v_{out}}{i_{out}} \\ &= \frac{r_{M1} \cdot i_{out} + r_{M2} \cdot i_{out} \cdot (1 + gm_{M2} \cdot r_{M1})}{i_{out}} \\ &= gm_{M2} \cdot r_{M1} \cdot r_{M2} + r_{M1} + r_{M2} \end{aligned} \tag{B.3}$$

For expressing the voltage gain, the transconductance G_m must be determined.

B.1.2 Transconductance of the Cascode Stage

The transconductance is the relationship between the input voltage and the output current. From inspection of Figure B.1c it is seen that v_{in} generates a current in the corresponding voltage controlled source. This current is divided between r_{M1} and the parallel coupling between $\frac{1}{gm_{M2}}$ and r_{M2}. From this observation, the transconductance is calculated using the current-divider formula [Razavi, 2001, p. 87 eqn. 3.120].

$$i_{out} = gm_{M1} \cdot v_{in} \cdot \frac{r_{M1}}{r_{M1} + \frac{1}{\frac{1}{r_{M2}} + gm_{M2}}} \tag{B.4}$$

From (B.4) the transconductance is calculated [Razavi, 2001, p. 87 eqn. 3.121].

$$Gm = \frac{i_{out}}{V_{in}} = \frac{gm_{M1} \cdot gm_{M2} \cdot r_{M1} \cdot r_{M2} + gm_{M1} \cdot r_{M1}}{gm_{M2} \cdot r_{M1} \cdot r_{M2} + r_{M1} + r_{M2}} \tag{B.5}$$

B.1.3 Voltage Gain

Finally the voltage gain of the cascode stage is found [Razavi, 2001, p. 87 eqn. 3.123].

$$\begin{aligned} |A_v| &= Gm \cdot R_{out} \\ &= gm_{M1} \cdot gm_{M2} \cdot r_{M1} \cdot r_{M2} + gm_{M1} \cdot r_{M1} \end{aligned} \tag{B.6}$$

The result obtained in (B.6) reveals that the cascode stage increases the gain of the CS amplifier by $gm_{M1} \cdot gm_{M2} \cdot r_{M1} \cdot r_{M2}$, which is roughly equal to the square of the intrinsic gain given of the CS amplifier which is equal to $gm_{M1} \cdot r_{M1}$ [Razavi, 2001, p. 53].

B.1.4 Simulation of Cascode Amplifier

The circuit depicted in Figure B.1a is simulated in ADS and the result is depicted in Figure B.2.

The unloaded voltage gain of the stage is simulated to 51 dB, which is significantly more than the voltage gain of 20, delivered by the CS stage. Furthermore, it is noticed that the gain is decreasing with

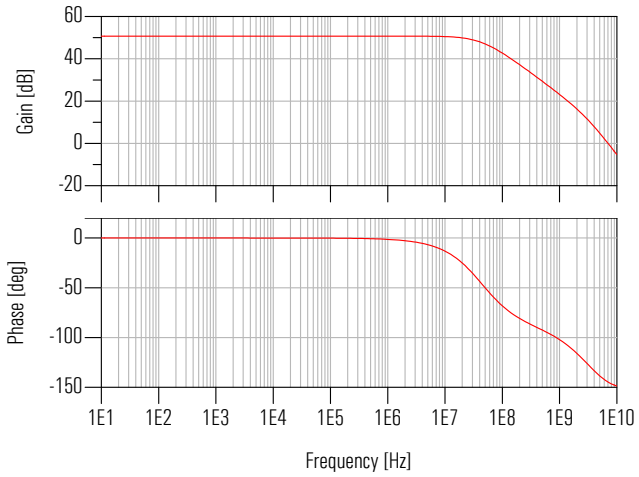


Figure B.2: *Simulated amplitude and phase response of the CS cascode amplifier. The phase response is relative to -180°. Transistor M_1 is biased at $V_{IN} = 0.4$ V and $I_{BIAS} = 100$ μA.* `[DIR032]`

20 dB/dec, indicating a single pole response. Therefore, the CS cascode amplifier can be considered as a single stage amplifier.

As theory predicts, the gain of the CS amplifier is squared. Since the cascode device is a CG amplifier, which is a wideband amplifier, the dominant pole in the transfer function is caused by the input device, and is not significantly reduced compared to the CS amplifier.

The voltage swing of the stage is reduced compared to the CS amplifier, since the cascode device must be kept in saturation for linear operation.

B.1.5 Cascode stage with PMOS Load

The purpose of this section is to determine the gain of the cascode stage in the case where it is loaded by a PMOS FET.To this end, the output impedance R_o and transconductance G_m of the circuit, are determined in analogy with the approach used in Section B.1.1. In Figure B.3a the circuit diagram of the PMOS loaded cascode amplifier is depicted.

In the following section the voltage gain of the circuit is derived. First, the output impedance R_o is determined using Figure B.3b. The PMOS transistor M_3 adds an impedance in parallel with the cascode stage. As the output impedance of the cascode stage is given in (B.3) and the output impedance of the PMOS load is given by r_{M3}, the resulting output impedance is given by

$$R_o = (gm_{M2} \cdot r_{M1} \cdot r_{M2} + r_{M1} + r_{M2} \| r_{M3}) \tag{B.7}$$

The transconductance of the circuit is determined by the cascode stage, and is not influenced by the added load, as illustrated in Figure B.3c. From the latter observations the voltage gain of the circuit depicted in Figure B.3a is found in (B.8) using the transconductance calculated in (B.5).

$$A_v = \underbrace{(gm_{M1} \cdot gm_{M2} \cdot r_{M1} \cdot r_{M2} + gm_{M1} \cdot r_{M1})}_{G_m} \cdot \underbrace{(gm_{M2} \cdot r_{M1} \cdot r_{M2} + r_{M1} + r_{M2} \| r_{M3})}_{R_o} \tag{B.8}$$

In the following section, the circuit depicted in Figure B.3a is simulated in order to determine how the load influences the gain and phase shift of the circuit.

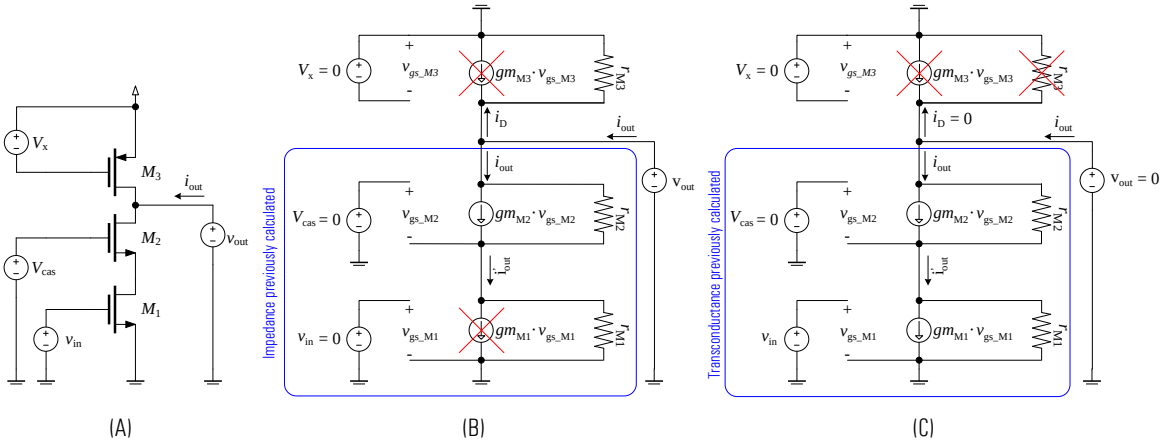


Figure B.3:
(a) Circuit diagram of the cascode stage with the PMOS load. (b) The components are replaced by their AC equivalents. The output voltage v_{out} is set to non zero, with the purpose of calculating the output impedance. (c) The input voltage v_{in} is set to non zero, enabling calculation of the transconductance of the circuit.

B.1.6 Simulation of Cascode stage with PMOS Load

The circuit depicted in Figure B.3a is simulated in ADS and the result is depicted in Figure B.4

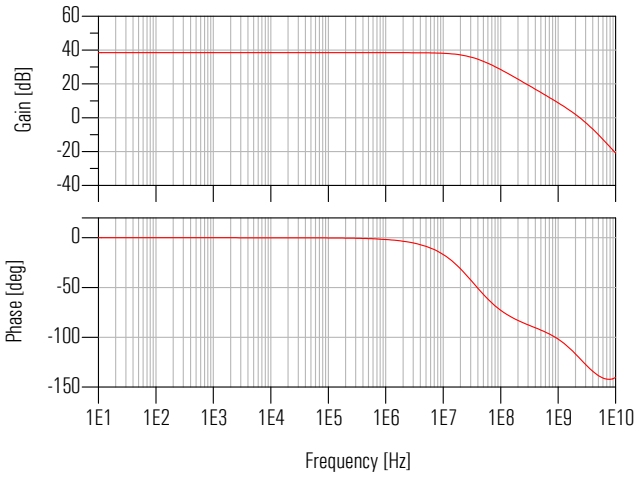


Figure B.4: *Simulated amplitude and phase response of the loaded cascode amplifier. The phase response is relative to -180°. Transistor M_1 is biased at $v_{in} = 0.45$ V and $I_{BIAS} = 100$ μA.* `[DIR033]` .

When comparing the obtained gain with the gain of the cascode stage, it is seen that the gain is decreased as expected. The loaded cascode stage delivers a gain of 38 dB and UGBW of 2.2 GHz.

B.2 Differential Amplifier

The differential input pair of an opamp is realized by a differential amplifier. This section shows how the gain of the differential amplifier is found and the used topology is chosen.

B.2.1 Basic Functionality of the Differential Pair

In the following the functionality of the basic differential pair, depicted in Figure B.5, is described.

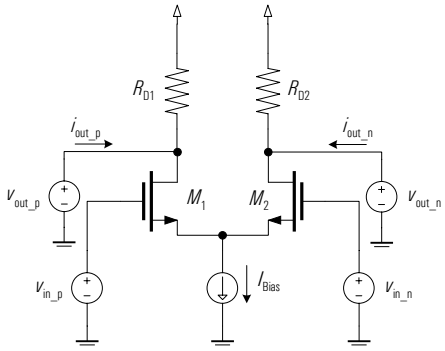


Figure B.5: *The basic differential pair.*

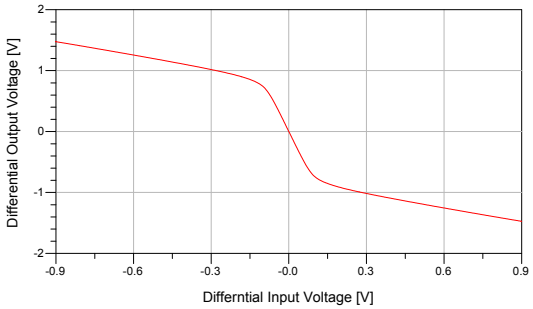


Figure B.6: *The relation between the differential input and output voltage.*

It is assumed that the two transistors M_1 and M_2 and resistors are identical. When the circuit is in equilibrium the two voltages $v_{\text{in\_p}}$ and $v_{\text{in\_n}}$ are equal, and both transistors drain $I_{\text{Bias}}/2$ through R_{D}. This is the bias point of the circuit.

If $v_{\text{in\_p}}$ is increased and $v_{\text{in\_n}}$ is decreased, the differential input ($v_{\text{in\_p}}$-$v_{\text{in\_n}}$) is increased. This makes the current through M_1 increase, and M_2 decrease, which further decreases $v_{\text{out\_p}}$, due to an increased voltage drop over R_{D1}, and visa versa for $v_{\text{out\_n}}$.

The output voltage is limited by the bias current and the size of the load resistor R_{D}. The differential stage is simulated in ADS and the result is depicted in Figure B.6. The slope of the curve is the differential voltage gain.

B.2.2 Differential Amplifier with Single Ended Output

The differential amplifier described in the previous section has a differential input and output. In the application of an opamp, a single ended output is needed. In order to realize a single ended output, an active current mirror is needed. The modified differential amplifier is depicted in Figure B.7.

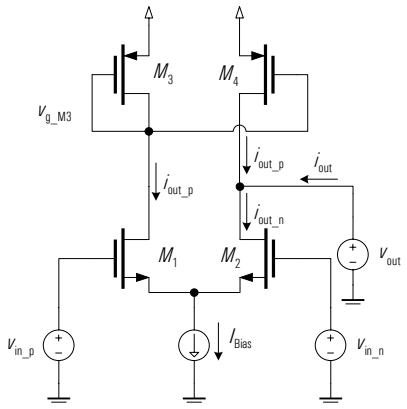


Figure B.7: *The active loaded differential pair [Razavi, 2001, p. 150]*

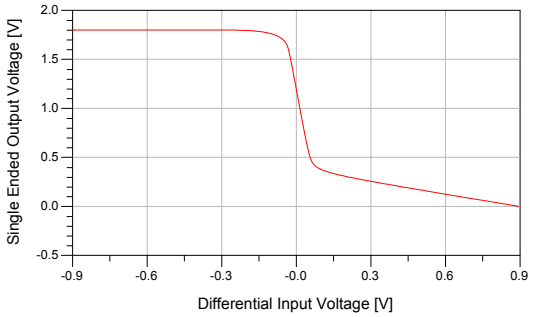


Figure B.8: *The relation between the differential input and the single ended output voltage.*

The voltage $v_{\text{in\_p}}$ makes M_1 drain the current $i_{\text{out\_p}}$. The transistor M_3 sources $i_{\text{out\_p}}$ by setting $v_{\text{g\_M3}}$. This voltage is also applied to the gate of M_4 making it source an identical current. The current caused by M_1 is because of that replicated, and sourced to the single ended output.

Gain of the active loaded differential stage

In the following section, a rough estimation of the gain is made. As previously shown, the voltage gain is given by.

$$|A_{\text{v}}| = G_{\text{m}} \cdot R_{\text{o}} \tag{B.9}$$

The gain is calculated by initially determining the transconductance and then the output resistance. If the current source which generates I_{Bias} is assumed to be ideal, it introduces no voltage drop, and the source nodes of M_1 and M_2 are connected to virtual ground. The gate-source potentials of the two transistors equal $v_{\text{in\_p}}$ and $v_{\text{in\_n}}$, respectively. If the current mirror realizes a perfect replication of $i_{\text{out\_p}}$, the transconductance is found by

$$\begin{aligned}
v_{\text{in_p}} &= -v_{\text{in_n}} \\
v_{\text{in}} &= v_{\text{in_p}} - v_{\text{in_n}} = 2v_{\text{in_p}} \\
i_{\text{out_p}} &= gm_{\text{M1}} \cdot v_{\text{in_p}} \\
i_{\text{out_n}} &= -gm_{\text{M2}} \cdot v_{\text{in_p}} \\
i_{\text{out}} &= i_{\text{out_p}} - i_{\text{out_n}} = gm \cdot v_{\text{in}}\Big|_{gm=gm_{\text{M1}}=gm_{\text{M2}}} \\
G_{\text{m}} &= \frac{i_{\text{out}}}{v_{\text{in}}} = gm
\end{aligned} \tag{B.10}$$

The output impedance is determined by shorting the input and calculating the output current i_{out} from the output voltage v_{out}. The circuit is depicted in Figure B.9.

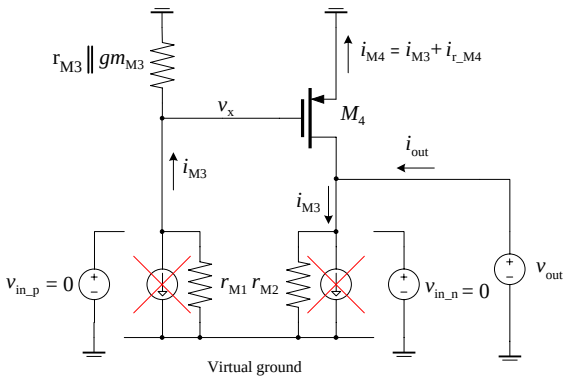


Figure B.9: *AC equivalent used for calculating the output impedance of the active loaded differential stage [Razavi, 2001, p. 152].*

Since the source of M_1 and M_2 are connected to virtual ground, and the gates of both transistors are shorted to ground, the current sources of the transistor AC equivalent models can be removed, leaving only the two resistors.

The current i_{out} is divided into three components; (i) the current i_{M3} drawn by r_{M1} and r_{M2}, (ii) the current $i_{\text{r\_M4}}$ drawn by r_{M4}, and (iii) the current drawn by the source of M_4. The latter current is a replicate of the current i_{M3} running through M_3. Based on these observations the output impedance is

calculated [Razavi, 2001, p. 152, eq. 5.27].

$$\begin{aligned} i_{\text{out}} &= 2 \cdot i_{\text{M3}} + i_{\text{r\_M4}} \\ &= 2 \cdot \frac{v_{\text{out}}}{r_{\text{M1}} + r_{\text{M2}} + \left(r_{\text{M3}} \| \frac{1}{gm_{\text{M3}}}\right)} + \frac{v_{\text{out}}}{r_{\text{M4}}} \\ &= 2 \cdot \frac{v_{\text{out}}}{2r_{\text{M2}} + \left(r_{\text{M3}} \| \frac{1}{gm_{\text{M3}}}\right)} + \frac{v_{\text{out}}}{r_{\text{M4}}}\Bigg|_{r_{\text{M2}}=r_{\text{M1}}} \end{aligned} \tag{B.11}$$

$$R_{\text{o}} = \frac{v_{\text{out}}}{i_{\text{out}}} = \frac{1}{\frac{2}{2r_{\text{M2}} + \left(r_{\text{M3}} \| \frac{1}{gm_{\text{M3}}}\right)} + \frac{1}{r_{\text{M4}}}} = (r_{\text{M2}} \| r_{\text{M4}})\Big|_{r_{\text{M2}} \gg r_{\text{M3}} \| \frac{1}{gm_{\text{M3}}}} \tag{B.12}$$

From (B.12) the voltage gain of the circuit is determined using (B.9) and (B.12).

$$|A_{\text{v}}| = gm \cdot (r_{\text{M2}} \| r_{\text{M4}}) \tag{B.13}$$

The result obtained in (B.13) is identical to the gain of the loaded CS amplifier. This is convenient since the gain of a differential amplifier can then be predicted using only the single ended version of the circuit.

B.3 CMOS Inverter

In logic circuitry the inverter is the most simple and most frequently used element [Sedra and Smith, 1998]. For understanding more complex logic circuitry, a basic knowledge of the inverter must therefore be established. The comparator in the $\Delta\Sigma$ Transmitter relies on logic circuitry in the process of sampling the input signal on the positive edges of the clock signal. This makes it relevant to investigate the operation of the CMOS inverter. Also the AND-gate is based on logic circuitry.

B.3.1 Simplified Operation of the Inverter

The CMOS inverter is composed of two matched transistors; one PMOS and one NMOS. By matched is meant that the two transistors realize a symmetric voltage transfer characteristic. The benefit of this matching and how it is realized are described later. The CMOS inverter circuit is illustrated in Figure B.10*a*.

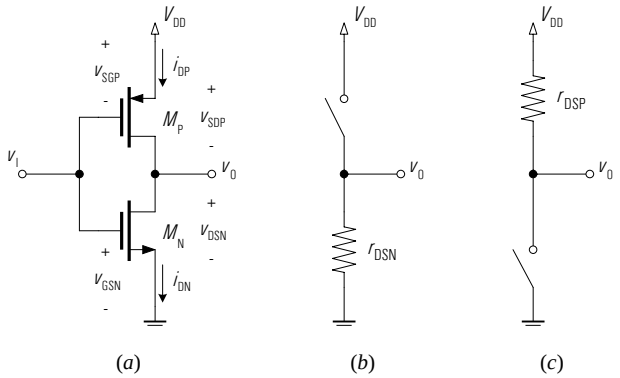


Figure B.10: *CMOS inverter. (a) Inverter circuit using one PMOS and one NMOS transistor. (b) Circuit equivalent for high input voltage. (c) Circuit equivalent for low input voltage.*

With v_{I} being a logic input signal with a voltage level of either 0 or V_{DD} volt, the operation of the inverter can be described as a switching circuit. When $v_{\text{I}} = V_{\text{DD}}$ volt, the voltage drop v_{SGP} from the

source to the gate of the PMOS transistor equals 0 V causing the transistor to open ($i_{DP} = 0$). At the same time the voltage drop v_{GSN} from the gate to the source of the NMOS transistor equals V_{DD} causing it to close. At this point the NMOS transistor pose a small resistance r_{DSN} between drain and source creating a low-resistance path between the output terminal and ground as illustrated in Figure B.10*b*. This results in zero output voltage. In Figure B.10*c* the opposite case is depicted. The input voltage v_I equals 0 V closing the PMOS transistor and opening the NMOS transistor, changing the output voltage to V_{DD} volt as a low-resistance path to the DC supply is made.

Studying the two equivalent circuits in Figure B.10 on the preceding page reveals that the static power dissipation is 0 W as no current flows in the two states described above. This is one of the attractive features of the CMOS inverter. Despite the fact that the quiescent current of the inverter is zero, its load-driving capability is very high [Sedra and Smith, 1998, p. 428]. With a high input voltage as in Figure B.10 (b), transistor M_N can sink a relatively large load current which quickly discharges the load capacitance. The transistor is said to pull the output voltage down toward ground, why it is referred to as a pull-down device. On the other hand the transistor M_P can source a relatively large load current when the input voltage is low. This current quickly charges up the load capacitance, pulling the output voltage close to V_{DD}, why it is referred to as a pull-up device.

B.3.2 Matching of the NMOS and PMOS Transistors

The description of the operation of the CMOS inverter in the section above is limited to the situation where the input voltage is either zero or V_{DD} volt. In the following the CMOS inverter is characterized for various input voltages leading to the explanation of how symmetric performance is achieved.

The current-voltage relationship is for the NMOS transistor in the triode region given by

$$i_{DN} = k'_n \left(\frac{W}{L}\right)_n \left[(v_{GSN} - V_{tn})v_{DSN} - \frac{1}{2}v_{DSN}^2\right] \qquad \text{for} \qquad v_{DSN} \leq v_{GSN} - V_{tn} \tag{B.14}$$

and in the saturation region

$$i_{DN} = \frac{1}{2}k'_n \left(\frac{W}{L}\right)_n (v_{GSN} - V_{tn})^2(1 + \lambda_n v_{DS}) \qquad \text{for} \qquad v_{DSN} \geq v_{GSN} - V_{tn} \tag{B.15}$$

where the factor $(1 + \lambda_n v_{DS})$ accounts for the channel length modulation, k'_n is the process transconductance, V_{tn} is the zero-bias threshold voltage and W and L are the gate width and length, respectively. The process transconductance is given by

$$k'_n = C_{ox} \cdot \mu_n \ [\mu\text{A/V}^2] \tag{B.16}$$

where C_{ox} is the oxide capacitance and μ_n is the electron mobility. Applying the notation of Figure B.10, the current-voltage relation is given by

$$i_{DN} = k'_n \left(\frac{W}{L}\right)_n \left[(v_I - V_{tn})v_O - \frac{1}{2}v_O^2\right] \qquad \text{for} \qquad v_O \leq v_I - V_{tn} \tag{B.17}$$

and

$$i_{DN} = \frac{1}{2}k'_n \left(\frac{W}{L}\right)_n (v_I - V_{tn})^2(1 + \lambda_n v_O) \qquad \text{for} \qquad v_O \geq v_I - V_{tn} \tag{B.18}$$

That is, $v_I = v_{GSN}$ and $v_O = v_{DS}$. For the PMOS transistor in the triode region the relation is given by

$$i_{DP} = k'_p \left(\frac{W}{L}\right)_p \left[(v_I - V_{DD} - V_{tp})(v_O - V_{DD}) - \frac{1}{2}(v_O - V_{DD})^2\right] \qquad \text{for} \qquad v_O \geq v_I - V_{tp} \tag{B.19}$$

and in the saturation region

$$i_{\mathrm{DP}} = \frac{1}{2}k'_{\mathrm{p}}\left(\frac{W}{L}\right)_{\mathrm{p}}(v_{\mathrm{I}} - v_{\mathrm{DD}} - V_{\mathrm{tp}})^2(1 + \lambda_n(v_{\mathrm{O}} - V_{\mathrm{DD}})) \qquad \text{for} \qquad v_{\mathrm{O}} \leq v_{\mathrm{I}} - V_{\mathrm{tp}} \tag{B.20}$$

That is, $v_{\mathrm{GS}} = v_{\mathrm{I}} - V_{\mathrm{DD}}$ and $v_{\mathrm{DS}} = v_{\mathrm{O}} - V_{\mathrm{DD}}$.

For a symmetric voltage-transfer characteristic, the inverter must be designed such that two conditions are met; (i) $V_{\mathrm{tn}} = |V_{\mathrm{tp}}|$ and (ii) $k'_{\mathrm{n}}\left(\frac{W}{L}\right)_{\mathrm{n}} = k'_{\mathrm{p}}\left(\frac{W}{L}\right)_{\mathrm{p}}$ [Sedra and Smith, 1998, p. 429], where k'_{p} is related to the hole mobility μ_{p} of the PMOS transistor, as in (B.16). The specified values for the electron mobility of the NMOS transistor and the hole mobility of the PMOS transistor are $\mu_{\mathrm{n}} = 332.1$ cm^2/Vs and $\mu_{\mathrm{p}} = 90$ cm^2/Vs.

In the first condition, the threshold voltage of the transistors is related to the manufacturing process, but can however be altered by forcing a constant current into the bulk terminal of the transistor in order to accommodate a desired threshold voltage [Lehmann and Cassia, 2001]. The threshold voltages of the two types of transistors are $V_{\mathrm{tn}} = 0.3075$ V and $V_{\mathrm{tp}} = -0.4325$. Whether this difference gives rise to adaptive measures depends on the specific design. The second condition can be accommodated by scaling the widths of the NMOS and PMOS transistor appropriately, such that $\frac{W_{\mathrm{p}}}{W_{\mathrm{n}}} = \frac{\mu_{\mathrm{n}}}{\mu_{\mathrm{p}}}$, while keeping the gate length of the two transistors the same. With the given values, the gate width of the PMOS transistor must be a factor

$$\frac{W_{\mathrm{p}}}{W_{\mathrm{n}}} = \frac{332.1 \text{ cm}^2/\text{Vs}}{90.0 \text{ cm}^2/\text{Vs}} = 3.69 \tag{B.21}$$

greater than that of the NMOS, for realizing a symmetric voltage-transfer characteristic.

Excess Loop Delay

C

ELD is defined as the delay from the quantizer clock to the Clocked Comparator output [Cherry and Snelgrove, 2002, p. 75]. In som extend this delay is unwanted, but theory also claims that some ELD improves the SNR of a $\Delta\Sigma$-ADC. In [Gosslau and Gottwald, 1988] it is shown that the best SNR is obtain with an ELD of 25% of the a sampling period. The simulation presented in [Gosslau and Gottwald, 1988, p. 2319] is replicated in Figure C.1, to clarify if ELD introduces an equal improvement in the presented $\Delta\Sigma$ Modulator.

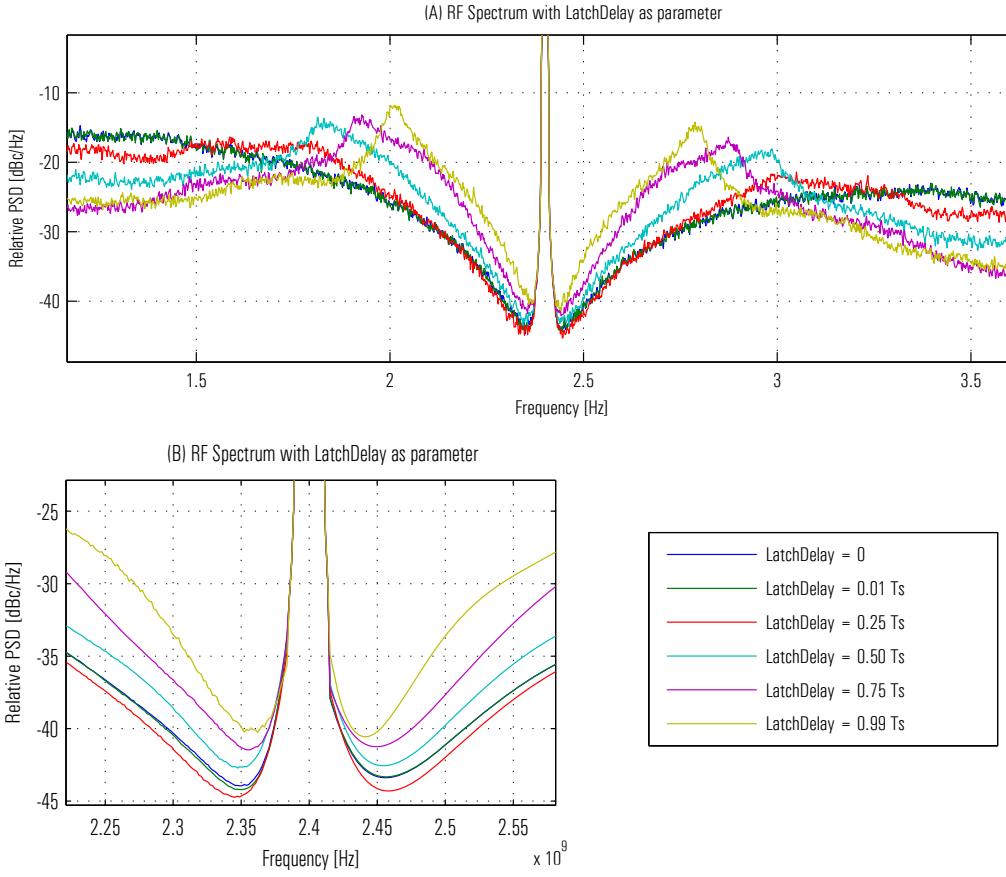


Figure C.1: *Influence of ELD on output PSD.*

From inspection of the simulation results it is concluded that ELD influences the output spectrum. In bottom window, it is seen that the quantization noise can be attenuated approximately 1 dB, if the ELD is $0.25 \cdot T_s$, but if the delay is too large the SNR is degraded several dB.

The following section present a method to remove the influence form ELD. Furthermore, it is presented how a $\Delta\Sigma$ Transmitter is designed to replicate the behavior of a system with a specific degree of ELD.

C.1 Compensating for Excess Loop Delay

In previous analysis it is established that excess loop delay can degrade the performance of the $\Delta\Sigma$ Transmitter. This section presents a method, not only for removing ELD from a first order $\Delta\Sigma$ Transmitter, but also for designing for a specific delay, which can enhance system performance.

C.1.1 Analysis Method

The presented method seeks to develop a circuit with excess loop delay, that performs identically with a circuit with no excess loop delay. For this purpose it must be defined what makes two $\Delta\Sigma$ Modulators or $\Delta\Sigma$ Transmitters equivalent.

As both the discrete time and continuous time $\Delta\Sigma$ ADC and the presented $\Delta\Sigma$ Transmitter contain a quantizer that generates the output signal, it is defined that two modulators are equivalent if, for the same input waveform, their quantizer input voltage at sampling instances are equal. This causes the quantizer to produce identical output signals. That is, if both input and output signals from the modulators are identical, the two modulators are equivalent. [Cherry and Snelgrove, 2002, pp. 29-31]

Analyzing what makes two modulators equivalent is most conveniently done in the z-domain. However, in the present case this involves some limitations.

Limitations

Theory shows that a continuous $\Delta\Sigma$ Modulator has a discrete time equivalent [Cherry and Snelgrove, 2002, pp. 29-33]. Unfortunately the $\Delta\Sigma$ Transmitter does not have a discrete time equivalent, because the sampling interval is non-constant, why the fundamental rules of the z-transformation are not valid.

However, to illustrate the influence of ELD and to show the principle of how its influence is removed, the z-domain representation is used. This is because the phase of the RF-clock changes much faster than the phase of the message signal. Therefore, the phase of the modulated clock is considered constant if the time of interest is sufficiently small. The following section describes the z-domain representation of a first order $\Delta\Sigma$ Modulator.

C.1.2 The Ideal $\Delta\Sigma$ Modulator

The present section presents a z-domain representation of a $\Delta\Sigma$ Modulator without ELD. Applying the method described above, the loop is opened around the quantizer, as illustrated in Figure C.2a and C.2b

If the input signal $u(t)$ is set to zero, the filter output $x(t)$ only depends on the quantizer output $y(t)$. Due to this observation the functionality of the $\Delta\Sigma$ Modulator is described only by the transfer function $H(s)$, as depicted in Figure C.2b [Cherry and Snelgrove, 2002, p. 30]. The output from the quantizer $y(t)$ is expressed as a sequence of square pulses. The general formula of a square pulse is given by

$$\hat{r}_{(\alpha,\beta)}(t) = \begin{cases} A, & \alpha \leq t < \beta \\ 0, & \text{otherwise} \end{cases} \tag{C.1}$$

If the amplitude and the sample period T_s are normalized to one, $y(t)$ is described by (C.2) [Cherry and Snelgrove, 2002, p. 77].

$$y(t) = \hat{r}_{(0,1)}(t) = \begin{cases} 1, & 0 \leq t < 1 \\ 0, & \text{otherwise} \end{cases} \tag{C.2}$$

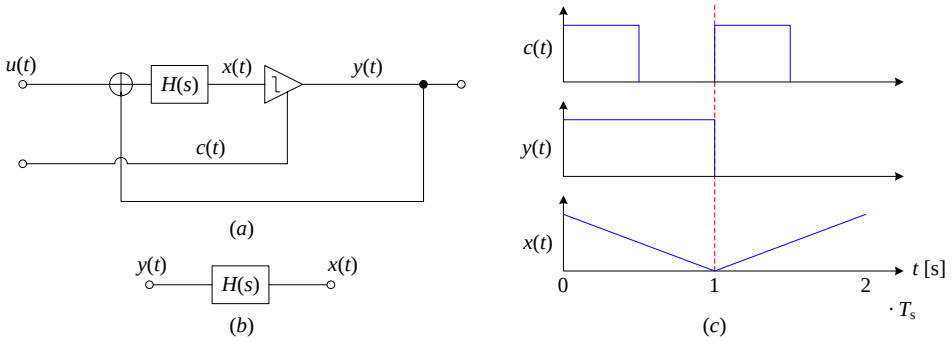


Figure C.2: *Ideal ΔΣ Modulator and principle waveforms.*

A z-domain representation of $y(t)$ and $H(s)$ is given in Equation (C.3) and (C.4). For simplicity an integrator is used as loop filter [Cherry and Snelgrove, 2002, p. 79].

$$y(z) = \mathcal{Z}\left\{y(t)\right\} = \mathcal{Z}\left\{\hat{r}_{(0,1)}(t)\right\} = 1 \tag{C.3}$$

$$H(z) = \mathcal{Z}\left\{\mathcal{L}^{-1}\left\{H(s)\right\}\right\} = \mathcal{Z}\left\{\mathcal{L}^{-1}\left\{\frac{1}{s}\right\}\right\} = \frac{1}{z} \tag{C.4}$$

$$x(z) = y(z) \cdot H(z) = \frac{1}{z} \tag{C.5}$$

From (C.5) it is seen that $x(z)$ is an integration of $y(z)$ as expected. By applying a similar procedure, expressions for $x(z)$ and $y(z)$ are derived in the case where the loop suffers from excess loop delay.

C.1.3 The influence of Excess Loop Delay

In Figure C.3, ELD is added to the system depicted in Figure C.2.

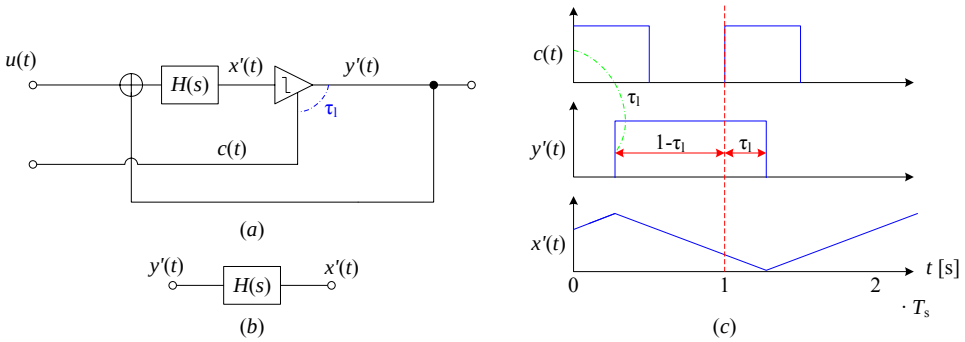


Figure C.3: *ΔΣ Modulator with excess loop delay.*

The pulse $y'(t)$ shown in Figure C.3c is described using two square pulses [Cherry and Snelgrove, 2002, p. 79]

$$y'(t) = \hat{r}_{(\tau,1+\tau)}(t) = \hat{r}_{(\tau,1)}(t) + \hat{r}_{(0,\tau)}(t-1) \tag{C.6}$$

$$\tag{C.7}$$

which in the z-domain gives

$$y'(z) = \mathcal{Z}\left\{\hat{r}_{(\tau,1)}(t) + \hat{r}_{(0,\tau)}(t-1)\right\} = (1-\tau) + z^{-1} \cdot \tau \tag{C.8}$$

From the input $y'(z)$, the output $x'(z)$ is given by

$$x'(z) = H(z) \cdot y'(z) = \frac{1-\tau}{z} + \frac{\tau}{z^2} \tag{C.9}$$

As seen in the equation, excess loop delay changes the quantizer input voltage $x'(z)$ at sampling instances, as (C.9) only equals (C.5) when $\tau = 0$. This results in a changed output bit sequence.

The following section presents a method to remove the influence of excess loop delay.

C.1.4 Canceling out Excess Loop Delay

The literature shows that the influence of excess loop delay can be removed by changing the shape of the pulse fed back to the filter [Cherry and Snelgrove, 2002, p. 96]. The principle is applied in the first order $\Delta\Sigma$ Transmitter, which results in the circuit depicted in Figure C.4a.

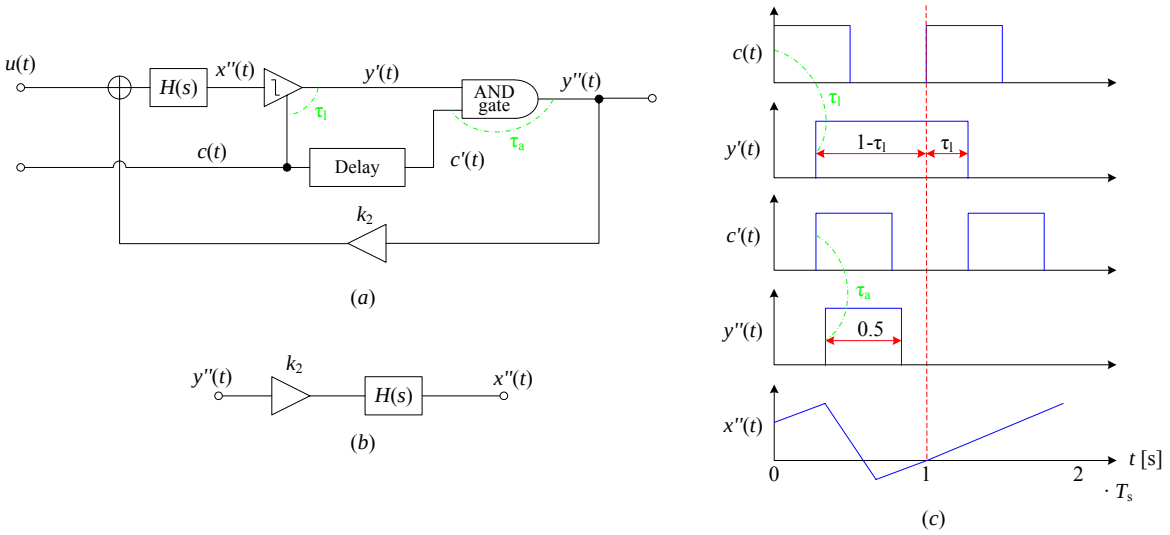


Figure C.4: *$\Delta\Sigma$ Transmitter used to remove the influence of excess loop delay and corresponding waveforms.*

Instead of taking the feedback from the output of the comparator, the output from the AND-gate, denoted $y''(t)$, is applied in feedback. A pulse from the AND-gate always stays within one clock period, why it is described by

$$y''(t) = \hat{r}_{(\tau_{la},\tau_{la}+0.5)}(t)\Big|_{\tau_{la}=\tau_l+\tau_a} \tag{C.10}$$

which in the z-domain gives

$$y''(z) = \mathcal{Z}\left\{\hat{r}_{(\tau_{la},\tau_{la}+0.5)}(t)\right\} = 0.5 \tag{C.11}$$

Applying the new feedback, the filter output $x''(z)$ is given by

$$x''(z) = H(z) \cdot y''(z) \cdot k_2 = \frac{\frac{1}{2} \cdot k_2}{z} = \frac{1}{z}\Big|_{k_2=2} \tag{C.12}$$

As seen in (C.12), the influence of ELD can be completely removed, if the gain factor k_2 is set equal to 2. The limitation of the method is that the total delay τ_{la} must be less than $0.5T_s$, otherwise the square pulse enters the adjacent clock period and the description of the pulse $y''(z)$ changes. Since the loop filter functions as an integrator, the area of the square pulse $y''(z)$ is computed. If the area of $y''(t)$ is equal to $y(t)$, the voltage signal at the quantizer input in Figure C.4 is equal to that in Figure C.3 at the sampling instance and the two systems are identical. In other words if the length of $y(z)$ is cut by half, the amplitude must be doubled in order to obtain the intended functionality, and the similarity is ensured as long the total delay τ_{la} is less than $T_s/2$.

C.1.5 Arbitrary Excess Loop Delay

Using a combination of feedbacks from the comparator and AND-gate, the system can be designed to exhibit a given loop delay, given the delays of the comparator and AND-gate are known. The presented system utilizes two feedback loops, as illustrated in Figure C.5a.

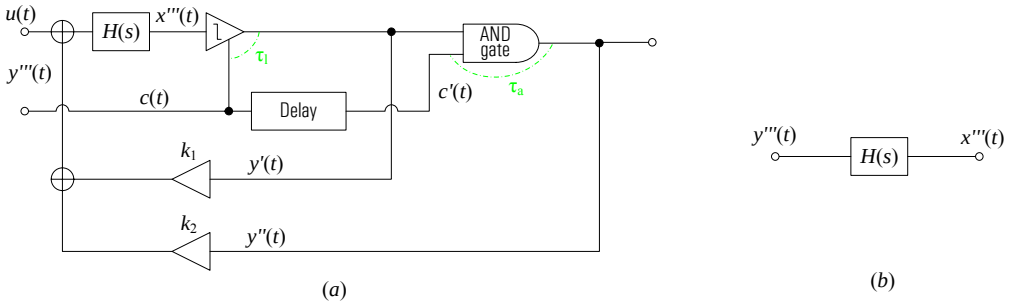


Figure C.5: *Circuit that enables to design for a specific excess loop delay.*

The two feedback signals $y'(t)$ and $y''(t)$ are summed, resulting in $y'''(t)$, which is given by

$$y'''(t) = k_1 \cdot \hat{r}_{(\tau_l,1)}(t) + k_1 \cdot \hat{r}_{(0,\tau_l)}(t-1) + k_2 \cdot \hat{r}_{(\tau_{la},\tau_{la}+0.5)}(t) \tag{C.13}$$

which transformed to the z-domain yields

$$y'''(z) = k_1 \cdot (1-\tau_l) + z^{-1} \cdot k_1 \cdot \tau_l + \frac{1}{2} \cdot k_2 \tag{C.14}$$

From the latter equation, the signal $x'''(z)$ at the comparator input is given by

$$x'''(z) = \frac{\frac{1}{2}k_2 + (1-\tau_l)\,k_1}{z} + \frac{\tau_l k_1}{z^2} = \frac{1-\tau_s}{z} + \frac{\tau_s}{z^2}\bigg|_{k_1=\frac{\tau_s}{\tau_l},k_2=2\left(1-\frac{\tau_s}{\tau_l}\right)} \tag{C.15}$$

where τ_l is the delay in the comparator and τ_s is the desired excess loop delay. If the delay in the comparator is known, and the total delay in the comparator and AND-gate is less than half T_s, the system can be designed to display the functionality of a $\Delta\Sigma$ Transmitter with any given excess loop delay between 0 and $T_s/2$.

The previous sections show that the excess loop delay impairs the performance of the $\Delta\Sigma$ Transmitter. As a solution, it is presented how ELD can be completely removed, or how the $\Delta\Sigma$ Transmitter can be designed to display a certain delay. Simulation results are presented in the following.

C.1.6 Simulation Results

The method presented in section C.1.4 is simulated in Matlab and the result is presented in Figure C.6

The top curve shows influence on the spectra when excess loop delay is added to the modulator. it is seen that power is moving closer to the message signal as the ELD increases. When the feedback is taken at the output of the AND-gate as presented in Section C.1.4, the system should not be sensitive to ELD. If the bottom plot of Figure C.6 is inspected, it is seen that the output spectrum do not change while the ELD is less than $T_s/2$. The simulation performed with ELD equal to $0.25T_s$ has identical performance to the system with no ELD. When the delay is made larger than $0.5T_s$ the delay influences the spectrum, this is because the pules from the AND-gate is present in the adjacent clock period. On this basis it can be concluded that the influence from ELD in theory can be removed by using the signal from the output of the AND-gate as feedback.

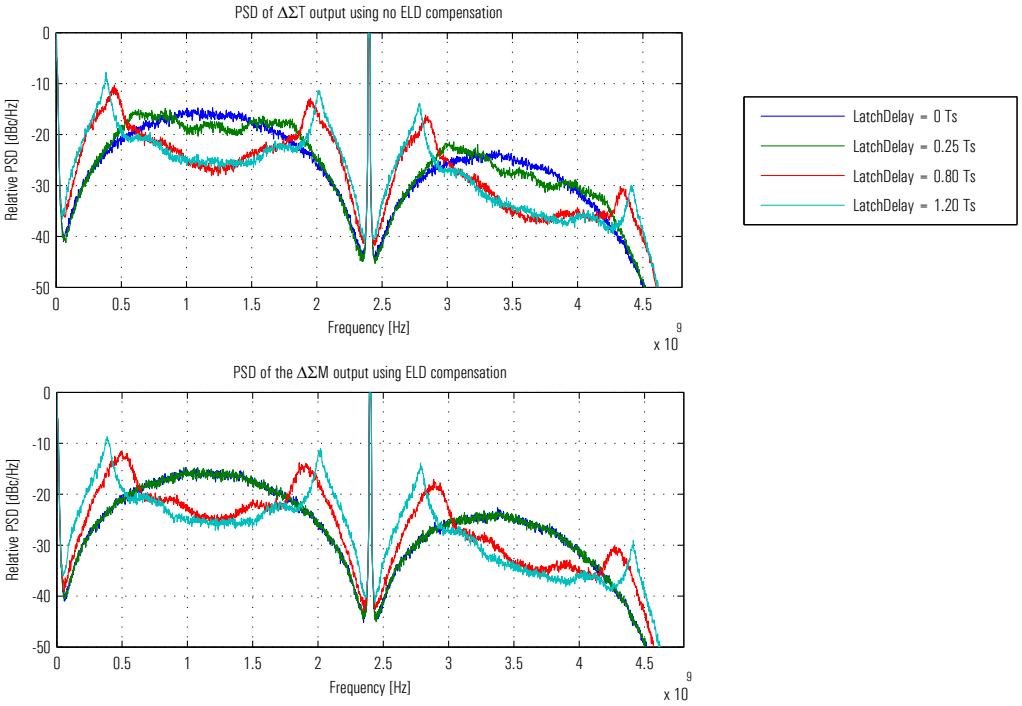


Figure C.6: *Comparison between $\Delta\Sigma$ Transmitter with and with out ELD compensation.*

Simulation of EVM

D

Besides the output spectrum, the EVM is used to evaluate the signal quality. The EVM is a measure for the amount of distortion a system adds to both the phase and amplitude of a signal. The calculation of EVM is implemented in Matlab and applied throughout the design phase in order to determine whether a design iteration yields an improved result.

D.1 Basic Concept

The basic concept of EVM is to compare the input and output signals of a system. Under appropriate conditions, the error can be found as the difference between the input and output signal, as depicted in Figure D.1.

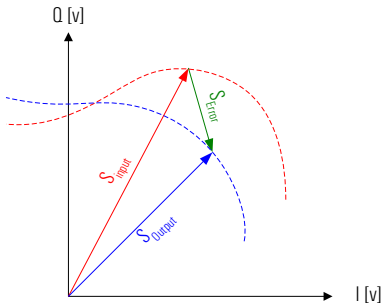


Figure D.1: *The error signal is the difference between the input and output signal.*

The error is measured when the constant time, phase and amplitude are equalized. Consequently, the error changes over time. This equalization is natural to perform as a receiver is capable to perform a constant error correction as a part of the demodulation.

When the constant error is removed the error signal as a function of time is given by

$$S_{\text{error}}(n) = S_{\text{output}}(n) - S_{\text{input}}(n) \tag{D.1}$$

n denotes the time reference in the discrete domain. From the error signal the RMS error is given by [Larsen, 2006, MM2, p.20]

$$EVM_{\text{RMS}} = \sqrt{\frac{\sum_n |S_{\text{Error}}(n)|^2}{\sum_n |S_{\text{Input}}(n)|^2}} \cdot 100 \quad [\%] \tag{D.2}$$

In the present application the output signal is translated to an RF frequency. Before a comparison with the input signal can be made, the frequency translation must be reversed.

D.2 Frequency Translation

The down conversion of the output signal is realized by applying a direct down converter and an ideal filter in an ideal receiver. The principle is depicted in Figure D.2.

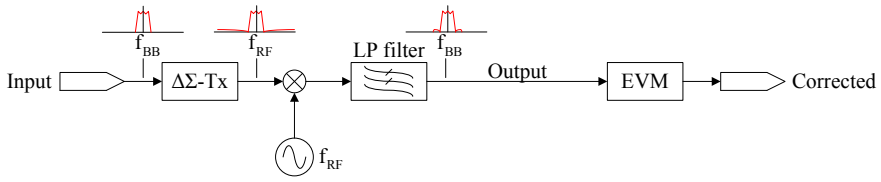


Figure D.2: *Principle of ideal down receiver.*

The input is passed through the $\Delta\Sigma$ Transmitter. Then, the RF output is down converted and filtered, and the signal is denoted S_{output}. The resulting signal is located at complex baseband while the image located at two times the RF frequency is removed.

Prior to calculating EVM, constant errors are removed.

D.3 Equalizing Constant Errors

Before the actual EVM can be measured, the constant errors must be removed. In the following sections methods to remove the constant errors are presented.

Constant Time Error

A compensation is made for constant delay in the $\Delta\Sigma$ Transmitter. The constant delay is determined by calculating the cross correlation between input and output. Where the cross correlation has its peak corresponds to the constant delay between input and output

Constant Phase Error

The constant phase error between the input and output signal is given by

$$\theta_{\text{error}} = \angle \left(\sum_n S_{\text{Output}}(n) \cdot S^*_{\text{Input}}(n) \right) \tag{D.3}$$

The mean of the phase error θ_{error} in a given time period represents the constant phase error.

The output signal corrected for constant phase error is given by

$$S_{\text{corrected}}(n) = S_{\text{Output}}(n) \cdot e^{-j\theta_{\text{error}}} \tag{D.4}$$

Constant Amplitude Error

The constant amplitude error is removed by ensuring that the input and output signals have equal mean amplitude. For this purpose the amplitude normalization factor given by

$$A_{\text{norm}} = \sqrt{\frac{\sum_n S_{\text{input}}(n) \cdot S^*_{\text{input}}(n)}{\sum_n S_{\text{output}}(n) \cdot S^*_{\text{output}}(n)}} \tag{D.5}$$

The output signal corrected for constant amplitude error is given by

$$S_{\text{corrected}} = S_{\text{output}} \cdot A_{\text{norm}} \tag{D.6}$$

D.4 Limitation

In many occasions EVM it is calculated symbol-wise, meaning that errors are calculated at points of decision, defined as the time instance where the ideal signal crosses the constellation points. As a pre-generated WLAN signal is applied there is no knowledge of the points of decision. Instead, errors are calculated sample-wise. Due to this alternative way of determining EVM the measure is not used for reference in relation to requirements specified in any standard.

Comparator Layout Simulation

E

Based on simulations, this appendix serves the purpose of documenting that proper behavior is realized in the implemented layout.

E.1 Circuit Simulation using Micro-strip

E.1.1 Procedure

Available simulation tools do not allow for a final simulation of the on-chip integrated circuit layout. The implemented $\Delta\Sigma$ Transmitter is simulated on system level in Matlab and on circuit level in ADS, but no layout simulations exist. As an alternative, all transistors of the clocked comparator are interconnected with micro-strip models in ADS that correspond to the interconnections found in the layout. The substrate definition is specified in accordance with the UMC 0.18 μm process. Moreover, in order to verify the substrate definition applied in ADS, S-parameters of a given micro-strip are simulated in ADS Momentum using the same substrate definition. The obtained S-parameters are then compared with those of an identical micro-strip in ADS.

E.1.2 Substrate Definition

The six layer UMC 0.18 μm substrate is modeled with a micro-strip substrate. Signal paths are, to extent possible, routed in the top metal layer ME6, why this layer is mapped as the micro-strip metal conductor. Ground plane is laid out in the bottom metal layer ME1, why this corresponds to the micro-strip ground plane. The remaining metal layers are not accounted for. The UMC 0.18 μm substrate specifies the metal thickness and conductivity, and substrate thickness, which are applied in the ADS micro-strip model.

E.1.3 Simulation Results

From the chip layout of the clocked comparator, width and length of each path connecting transistors are given. Where T-junctions and corners are used in the layout the corresponding models are used in ADS. The transistors are supplied with ideal ground and V_{DD}. Square signals are applied using an input frequency of 760 MHz and a clock frequency of 2.4 GHz. Simulation result is given in Figure E.1

The waveforms reveal no influence of the applied micro-strip model. This does not ensure proper performance of the total layout, but it indicates that the precise interconnections of the transistors are not of crucial importance.

I order to ensure that the micro-strip substrate is a fair model of the UMC 0.18 μm substrate, simulations in Momentum are performed, as described in the following.

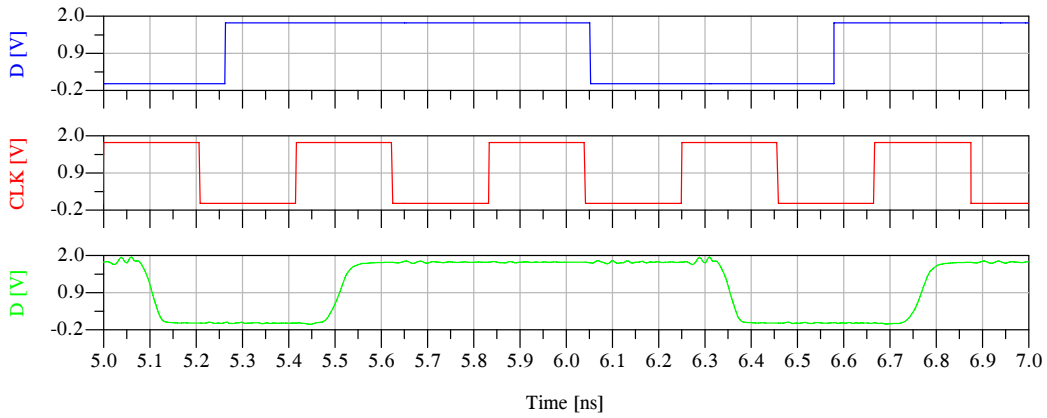


Figure E.1: *Simulated time domain representation of signals around the clocked comparator when connecting transistors with micro-strip.* `[DIR030]`

E.2 ADS vs Momentum Substrate Model

In Momentum, a substrate is defined applying the same specifications as in the ADS substrate definition. In the top metal layer ME6 a 400 μm long and 1.9 μm wide path is laid out. In the bottom layer a much larger ground metal is laid out. In a just as large, but very thin via layer, connection is made to a closed ground layer defined as a perfect conductor. This closed ground layer corresponds very well to the ground layer in the ADS micro-strip model, but it cannot be used as reference terminal when simulating S-parameters in Momentum. S-parameters are simulated in the frequency range 1 GHz to 10 GHz.

Likewise, S-parameters of a micro-strip of the same dimensions are simulated in ADS. The results of the two simulations are presented in Figure E.2.

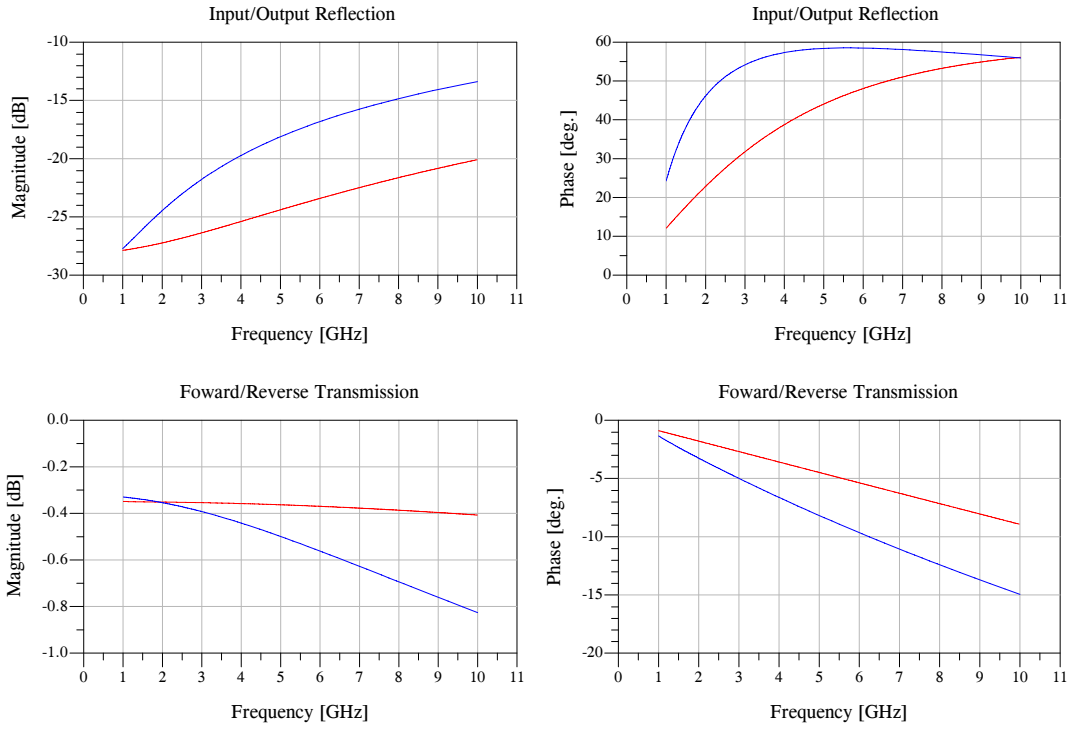


Figure E.2: *Comparison between S-parameters from ADS (red) and Momentum (blue).* `[DIR031]`

The simulated S-parameters show a good correlation between the two models. The ADS substrate model, however, varies less with frequency. In the Momentum model the loss of the path increases with frequency, where in ADS the loss remains almost constant in the frequency range. However, the variations in Momentum are on a very small scale, few tenths of one dB. A larger capacitive coupling to ground in the Momentum model could explain the increased loss at higher frequencies.

Power Supply Decoupling

F

In the case where varying load currents are present, the power supply must supply a nearly constant DC voltage to avoid noise coupling between the connected circuits [Ott, 1988, p. 122].

The voltage supply of every sub-circuit on the chip must be decoupled in order to short circuit high frequency noise to ground and to eliminate coupling between circuits. Power supply noise impairs performance and can degrade noise margins [Weste and Harris, 2005, p. 353]. This is especially caused by the fast switching circuits in the $\Delta\Sigma$ Transmitter. Especially the filter is vulnerable to voltage supply noise since it changes the quiescent point, that is specified by a voltage division.

Decoupling capacitors are included at the power supplies of the sub-circuits in the chip. The larger capacitive decoupling distributed in the integrated circuit, the more constant supply voltage. The available physical space on the CMOS chip limits the number of on-chip capacitors. If these on-chip capacitors cannot source the desired charge to a given sub-circuit, the voltage potential over the sub-circuit will fall and inject supply noise in terms of a voltage spike on the voltage supply. For the on-chip capacitors to maintain their voltage potential, they must be charged by another, and often larger capacitor, located off-chip on the PCB between the power supply and the chip. This large off-chip capacitor is denoted C_1 in Figure F.1, which illustrates the implemented decoupling strategy.

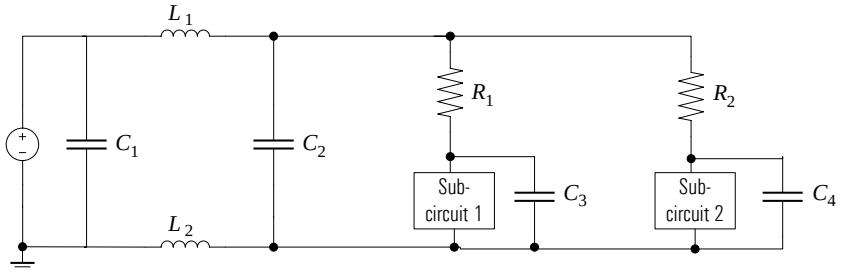


Figure F.1: *Principle of employing decoupling filters in order to isolate circuits and minimize power supply noise.*

To isolate the circuits and thereby minimize the power supply noise further, small resistors ($3 - 20\ \Omega$) are inserted between each circuit and the power supply. This functions as a low pass RC filter attenuating the high-frequency noise on the local supply. The inductors L_1 and L_2 represent the bonding wires between the PCB and the chip. The capacitor C_2 is a large decoupling capacitance located on-chip. The components R_1, C_3 and R_2, C_4 isolate one sub-circuit from the other. The voltage drop in the resistors will also decrease the power supply voltage. For that reason the external power supply voltage is increased in order to feed the desired voltage to the isolated circuit.

F.0.1 Simulation of the power supply decoupling

The capacitive decoupling needed for the designed circuits to operate appropriately, is simulated in ADS. The simulation is based on the $\Delta\Sigma$ Transmitter. This circuit is less decoupled than the other circuits on the designed chip, because of physical chip size constraints. If simulations show an appropriate decoupling of the $\Delta\Sigma$ Transmitter circuit, it should also ensure that the other circuits on the chip

	No decoupling [V_p]		Decoupling [V_p]	
	Max/min	Δ	Max/min	Δ
Power Supply	2.19 1.55	0.64	1.95 1.85	0.10
Delay inverters	2.15 1.49	0.66	1.84 1.75	0.09
Opamp	2.19 1.55	0.64	1.90 1.81	0.09
Clocked comparator	2.18 1.55	0.63	1.90 1.78	0.12
Inverter	2.19 1.55	0.64	1.90 1.76	0.14
And-gate	2.19 1.55	0.64	1.92 1.78	0.12

Table F.1: *Simulation results from power supply decoupling of $\Delta\Sigma$ Transmitter system. The specified values are peak values of the supply voltages. The intended voltage level of each sub-circuit is* 1.8 V . *The listed values are based on simulations located in* `[DIR028]` .

are decoupled appropriately.

The simulation is made as realistic as possible, modeling the signal and voltage sources, bonding wires, pads and paths as comparable to the layout as possible. Furthermore, the decoupling capacitors are simulated using the capacitances allowed by the available physical chip area. Further information on the simulation is located in `[DIR028]` .

The main power source in the simulation feeds a voltage of 1.9 V due to 100 mV voltage drops in the decoupling filters. Four 1 nH inductors represent the four bonding wires between the PCB and the chip. Micro-strip lines are included in the simulation in order to model the paths from the main on-chip power source to each sub-circuit. The models apply the same substrate definitions as those included in the clocked comparator simulation described in Appendix E. The dimensions of the simulated micro-strip lines are considered being worst case, with widths of 10 μm and lengths of 1000 μm and 1500 μm .

As an iterative process, the voltage supply at each sub-circuit is investigated and each sub-circuit is decoupled in order for the $\Delta\Sigma$ Transmitter system to function properly. An overview of the simulation results is listed in Table F.1.

The table depicts the simulated supply voltage at each sub-circuit of $\Delta\Sigma$ Transmitter and how the decoupling filters influence the supply voltage. The simulated external voltage supply is specified to 1.8 V in the case of no decoupling, and 1.9 V when decoupling is applied. For more illustrative purposes, the voltage supply of the opamp is depicted in Figure F.2.

This figure supports the peak voltage levels for the opamp listed in Table F.1. The plots from the simulations show an improvement in power supply stability when utilizing the decoupling filters. This tendency also applies for the remaining sub-circuits in the $\Delta\Sigma$ Transmitter system.

Since the decoupling of $\Delta\Sigma$ Transmitter is shown to be appropriate for required power supply stability, it is expected that all the sub-circuits on the designed chip are decoupled appropriately.

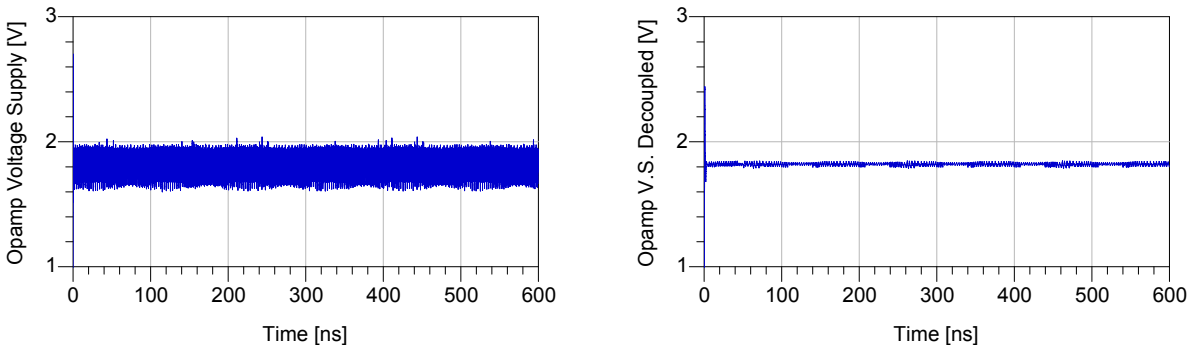


Figure F.2: *Time domain plots of the opamp voltage supply. Left figure illustrates the voltage supply when the $\Delta\Sigma$ Transmitter is not decoupled, and the left figure when $\Delta\Sigma$ Transmitter is decoupled. Plots are generated using* `[DIR028]` .

G Measurements

In the following sections the procedures of on-wafer probing, calibration of measurement equipment, and de-embedding are described. Hereafter, the measurements performed in this project are presented. The sections regarding measurements include lists of equipment, measurement setup, step-by-step measurement procedures, measurement results, and finally sources of errors. The data processing is found in the main report. The step-by-step measurement procedure guides the reader through every detail of the performed measurements. This makes it easy to reproduce the results obtained in the measurement, or perform the measurement again if the collected data was not useful. The reader should feel free to skip this section. Sections of the present chapter are listed below.

G.1 Measurement Equipment

Table G.1 contains a complete listing of the equipment used for the performed measurements.

No.	Device	Model	AAU-number
D1	Network analyzer	HP 8510C	LVNR 33792
D2	S-parameter test set	HP 8515A	LBNR 08242
D3	Synthesized sweeper	HP 83631B	LVNR 33793
D4	Analytical probe station	Cascade Summit 9000	LBNR 08952
D5	GSG probe	Cascade	56911, sn 406350
D6	GSG probe	Cascade	33828, sn 08829
D7	5-pin DC probe	Cascade	DCQ-05, sn 4766
D8	GSGSG probe	Cascade	56830, sn 4425R
D9	GSGSG probe	Cascade	56829, sn 4425Q
D10	Single tip DC needle	Cascade	33779
D11	ISS	Cascade 1995	005-016, sn 50294
D12	Bias feeder ZFBT-6G	Mini circuits	1270-00, A6-217-A-3
D13	Bias feeder ZFBT-6G	Mini circuits	1270-01, A6-217-A-3
D14	Power supply	B&O SN16A	52787, A6-211
D15	Power supply	B&O SN16A	52781, A6-215-B-4
D16	Multimeter	Fluke 189	56888
D17	Multimeter	Fluke 189	52781
D18	Coaxial cable (200 cm)	Sucoflex 104PB	1292-03, A6-222
D19	Coaxial cable (200 cm)	Utiflex	sn 00e06
D20	Coaxial cable (25 cm)	-	nr. 02, A6-217
D21	Coaxial cable (25 cm)	-	nr. 03, A6-217
D22	Torque wrench	Suhner	A6-217-A-6
D23	Vector signal generator	R&S SMIQ 06B	Inst8 52765
D24	Spectrum analyzer	R&S FSIQ 26	Inst8 52766
D25	Synthesized signal generator	MG3642A	LVNR 33704, A-212
D26	Signal generator	SMR 20	LVNR 52767,

Table G.1: *Measurement equipment used in the performed measurements.*

Through description of measurements, each device will be referred to using the label given in the column "No.":

The chip design makes it possible to verify the functionality of each sub-circuit separately. The following sections describe the procedures of on-wafer probing, calibration, and de-embedding. This is followed by descriptions of the performed measurements of each circuit.

G.2 Wafer Probing

When measurements are performed on the four test benches, on-wafer probing is carried out. The procedure of probing on a pad is illustrated in Figure G.1.

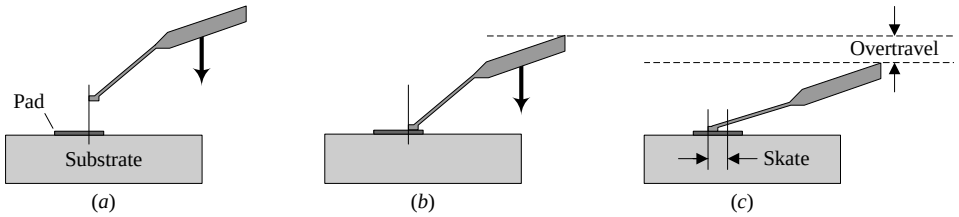


Figure G.1: *Side-view of a probe contacting an integrated structure. (a) Probe in the air. (b) Initial touchdown. (c) Probe overtravel and skate. [Jensen, 2007, p. 32]*

In Figure G.1a the probe is in the air close to the edge of the pad. Then the probe is vertically moved towards the pad until contact is made, G.1b. Continuing the downward motion, the probe tip skates on the pad, creating a well-defined contact to the pad, G.1c. The vertical movement from G.1b to G.1c is denoted overtravel and is defined as the amount of vertical movement of the probe after initial contact with the structure. When continuing the downward movement, the flexibility of the probe makes the tips move forward which is called skate.

Contact resistance

The contact issues of on-wafer probing constitute a large contributor to the overall measurement reliability and replicability [Kolding, 1999, p. 60].

The contact resistance is varying, due to different amounts of oxides and dirt on the contacting surfaces. The overtravel and skate combined, leads to a penetration into the top metal layer which establishes a low-ohmic contact to the pad. In order to obtain stable and repeatable low-ohmic contact, the probe tips must be pressed toward the chip with the same amount of pressure, using an equal amount of skate, at each measurement.

Contacting (or probing) is a destructive process. Each time a chip is measured, its pads are worn. The amount of skate is a trade-off between low contact resistance and wear on probes and pads [Jensen, 2007, p. 33]. On average $40 - 60$ μm of skate is a good compromise [Kolding, 1999, p. 49]. It can be difficult to re-probe if planarity of the probe has not been obtained. The marks, caused by skating on pads, indicate whether the probe tips are planar to the substrate, or if a planarity adjustment has to be done. If the marks from all the probe tips have equal skate size the probe is planar to the substrate.

G.3 Calibration and De-embedding

This section presents a method for performing on-wafer calibration and de-embedding used for two-port measurements. Initially, the utilized calibration and de-embedding methods are described, followed by the step-by-step procedures.

G.3.1 Calibration Method

Prior to measurements, the network analyzer is initialized and the S-parameter test set is calibrated, in order to establish a known reference plane in both the forward and reverse directions.

Systematic errors, like crosstalk and impedance mismatch, are caused by imperfections in test equipment and test setup. If they do not vary with time, they can be characterized through calibration and mathematically removed during the measurement process. [Agilent, 2002, p. 3].
The simple and widely used short-open-load-thru (SOLT) calibration method is performed using a ceramic impedance standard substrate (ISS) containing the high-precision calibration standards.

In the full SOLT 2-port calibration, each ground-signal-ground (GSG) probe is calibrated separately using reflection measurements with the open-circuit, short-circuit, and load standards. Both probes are then connected to the thru standard for a transmission calibration. It is important to apply the appropriate amount of skate in order to obtain a low-ohmic contact, as described in Appendix G.2 on the previous page.

The SOLT calibration procedure establishes a well-defined calibration reference at the probe tips. However, this calibration plane does not directly interface the Device Under Test (DUT), hereby creating the need for further post processing, called de-embedding.

G.3.2 De-embedding Method

The de-embedding creates a reference plane at the probing pads, and hereby reduces the parasitics introduced by the pads. Various de-embedding procedures can be applied. The widely used open/short de-embedding (OSD) is chosen because of the limited available chip area for test structures. The OSD method addresses the most considerable CMOS parasitics; namely pad admittance and contact impedance. The method is based on the assumption that the fixture parasitics leading to the DUT can be described by parallel admittances, Y_{p1} and Y_{p2}, as illustrated in Figure G.2.

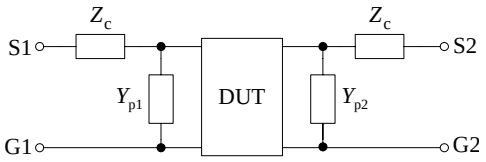


Figure G.2: *Fixture model topology for the OSD method.*

The contact impedances are represented by Z_c [Kolding, 1999, p. 120]. The OSD technique only requires two single-port de-embedding structures allocated on the chip; an open and a short GSGSG test structure [Kolding, 1999, p. 119-120]. An overview of the two de-embedding structures is given in Figure G.3.

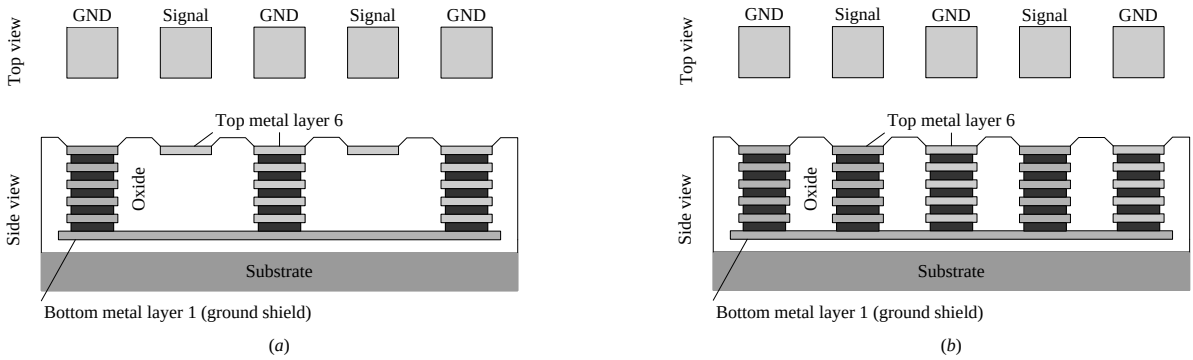


Figure G.3: *Deembedding structures. (a) Open standard. (b) Short standard.*

For the OSD procedure, the open and short fixtures are measured where the data represents the para-

sitics from the signal pads. Using a mathematical approach off-line, the pad parasitics are compensated for in the measured data [Kolding, 1999, pp. 120-121]. The mathematical computations are not described in detail here.

A necessary condition for a successful de-embedding is that the pads in the de-embedding fixtures are of same type as the pads utilized on the Chip. This is not the case for the two $\Delta\Sigma$ Transmitters, where ESD protected pads are used to enable bonding.

When Calibration and de-embedding is performed, the reference plane is at the probe pad edges as illustrated in Figure G.4. The pad parasitics are cancelled out by de-embedding.

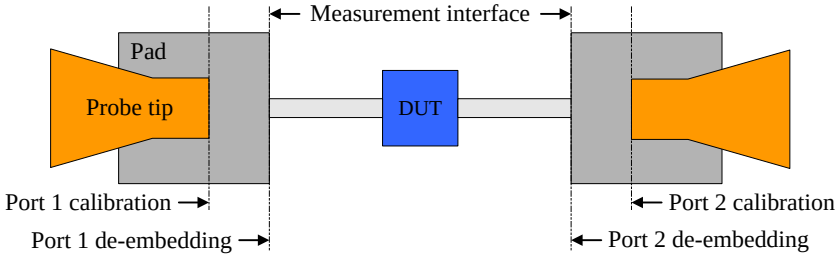


Figure G.4: *Reference plane is at the probe tips after the two-port SOLT calibration. The measurement interface is at the pad edges after the calibration and de-embedding.*

G.3.3 SOLT calibration procedure

The SOLT calibration method has been described and in the following a step-by-step procedure is presented. Prior to measurements of the four S-parameters of a DUT, the network analyzer is initialized and the measurement setup, from the S-parameter test set to the probe tips, is calibrated.

When doing a full 2-port SOLT calibration using the ISS print, each probe is separately calibrated with the open-circuit, short-circuit, and load standards. Then both probes are connected to the thru standard for a transmission calibration.

The following notation is used in the initialization and calibration procedures:

- A menu button on the network analyzer is presented as MENU.
- At the right side of the network analyzer screen, a series of softkeys are available. The function of each softkey is illustrated on the screen. In this chapter, each softkey is presented as <SOFTKEY>.

The calibration procedure is used prior to the following two measurements:

- Measuring S_{21} of the inverter. Measurement report in Appendix G.4.
- Measuring S_{21} of the AND-gate. Measurement report in Appendix G.5.

This step-by-step guide is used for all the measurements. The calibration procedure is similar in the three cases. The initialization process of the network analyzer prior to the measurements is almost similar. The only differences are the settings for frequency range and power level. The initialization specifications are defined in the respective measurement reports. As an example, the initialization for the inverter measurement is presented in the following.

Initialization of network analyzer

1. Connect the measuring equipment to a power outlet and turn it on in 'bottom-up' order. Allow it 30 - 60 minutes of warm up.
2. Preset the analyzer. [USER PRESET]
3. Set lower frequency limit. [DOMAIN] -> <FREQUENCY> -> [START] -> [2] [0] [0] [MHz].
4. Set higher frequency limit. [STOP] -> [1] [0] [GHz].
5. Set sweep type to stepped. Stimulus [MENU] -> <STEP>
6. Setting up the power level of the two ports of the test set.
   Stimulus [MENU] -> <POWER MENU> -> <POWER SOURCE 1> -> [-] [1] [0] [X1]
   <POWER SOURCE 2> -> [-] [1] [0] [X1].
7. Setting up number of points per measurement.
   Stimulus [MENU] -> <NUMBER OF POINTS> -> <801>

Now the network analyzer is adjusted to display the input signal in the desired frequency range of 200 MHz to 10 GHz sampled by 801 points and at a constant power level of -10 dBm. If the frequency range or power levels of the network analyzer are changed, the calibration procedure has to be redone.

Calibration procedure

The SOLT calibration is performed after the initialization of the network analyzer. The calibration procedure is the same for the three measurements. When probing on the ISS a skate of 50 μm is applied in order to obtain low contact resistance, as described in Appendix G.2.

1. Make sure the correct probes are mounted.
2. Set all DC voltages to zero.
3. Probe a contact substrate to ensure that probe tips are planar. If this is not the case, adjust planarity until scrape marks are similar.
4. Keep both probes lifted in the air.
5. Making a full 2-Port calibration. [CAL] -> <CAL 2 CSC GSG150> -> <FULL 2-PORT>
6. The three superior parameters for the full 2-port calibration are displayed. These parameters are <REFLECT'N>, <TRANSMISSION> and <ISOLATION>. One by one these parameters are calibrated. First the reflection is calibrated for S_{11} using the ISS while probe 2 is kept lifted in the air.
7. Calibrating reflection on port 1. Press softkey <Reflection>.
8. Keep probe 1 lifted in the air.
9. Press S_{11} <OPEN>. The open circuit data is measured, and the softkey <OPEN> is underlined.
10. Connect probe 1 to a short standard on the ISS.
11. Press S_{11} <SHORT>. The short circuit data is measured and the softkey <SHORT> is underlined.

12. Now connect probe 1 to a laser trimmed $50\ \Omega$ load standard on the ISS.
13. Press S_{11} <LOAD>. The load data is measured, and the softkey <LOAD> is underlined.
14. Lift probe 1 into the air.

Now, the reflection for S_{11} on Port 1 has been calibrated. Next, the reflection for S_{22} on port 2 is calibrated using the ISS. The procedure is done by repeating the steps for S_{11} shown above. Probe 2 is connected to the given standards and the S_{22} softkeys are pressed.

- Press <REFLECT'N DONE>.
- The reflection calibration coefficients have now been computed and stored on the instrument. The 2-Port cal menu is displayed, with the softkey <REFLECT'N> underlined.
- Now, the correction for transmission errors is calibrated.
- Press <TRANSMISSION>.
- Make a thru connection between port 1 and port 2. Turn the ISS 90° clockwise.
- Connect probe 1 to the left side of a thru standard in the ISS.
- Connect probe 2 to the right side of the same thru standard. Move the probes cautiously, to avoid skating into each other, causing damage to the probes.
- Press softkey <FWD.TRANS.THRU>. S_{21} frequency response is measured and the softkey is underlined.
- Press <FWD.MATCH.THRU>. S_{11} load match is measured and the softkey is underlined.
- Press <REV.TRANS.THRU>. S_{12} frequency response is measured and the softkey is underlined.
- Press <REV.MATCH.THRU>. S_{22} load match is measured and the softkey is underlined.
- Press <TRANS.DONE>.
- Raise both probes into the air.
- Press <ISOLATION> followed by <OMIT ISOLATION> and <ISOLATION DONE>. The correction for isolation is not required.
- Press <DONE 2-PORT CAL>.
- Press <SAVE 2-PORT CAL>. Save it at an appropriate cal set (cal set 1 - 8).

This completes the full 2-Port calibration procedure. Now the DUT can be connected and measured.

G.3.4 Open/Short de-embedding procedure

The OSD method has been described and in the following a step-by-step procedure is presented.

- Find the chip and place it on the probe station.
- While probe 2 is suspended in the air, connect probe 1 to the open standard.

- Measure the S-parameters and save the file. Name it e.g. Deemb_s11open.s2p.
- Now, connect probe 1 to the short standard.
- Measure the S-parameters and save the file. Name it e.g. Deemb_s11short.s2p.

The same procedure is done for probe 2, while probe 1 is suspended in the air. When all the four S-parameter files have been saved, the de-embedding measurements are completed. The saved files are used for the final post-processing of the measurement data.

G.4 Inverter Measurement

This section describes how the measurement on the inverter test bench is performed.

G.4.1 Purpose

The purpose of the measurement is to determine the amplitude and phase responses of the implemented inverter. From the phase response, the propagation delay of the inverter can be calculated. Simulated and measured data is found in `[DIR037]`.

G.4.2 Setup and Measurement Procedure

A sine wave with a DC offset of 0.9 V is applied at the input, and the output signal is measured. If the circuit has the intended functionality, the output is an inverted version of the input. The delay in the inverter is calculated from the phase response, when taking the $180°$ phase shift into account. The power of the input signal should be kept low enough to avoid compression in the inverter. The measurement setup is illustrated in Figure G.5.

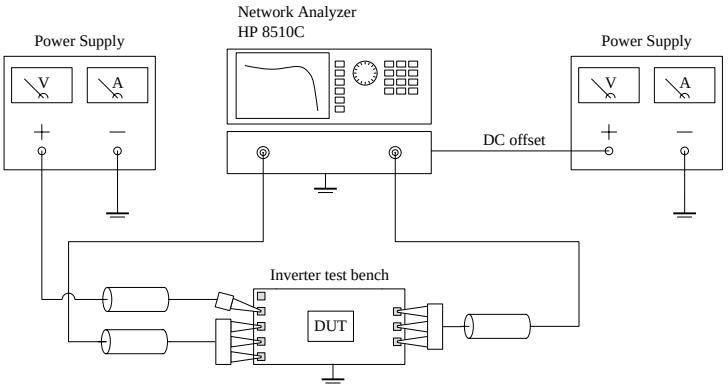


Figure G.5: *Setup used for testing the inverter.*

In order to perform calibration, de-embedding, and measurement, the S-parameter test set (Device D2), Network analyzer (D1), and synthesized sweeper (D3) are utilized. For probing, the probe station (D4) and two GSG probes (D5 & D6) are used. The calibration is performed using the ISS (D11). The inverter is power supplied by (D14) and V_{DD} is monitored using a multimeter (D16). The supply voltage V_{DD} of 1.8 V is injected directly onto the chip using a single tip DC needle (D10). The DC offset is generated by (D15) and is supplied through the S-parameter test set. The GSG probes are connected to each port of the S-parameter test set through two 200 cm and two 25 cm coaxial cables; (D18), (D19),(D20), and (D21), respectively.

After performing the initialization process, the measurement equipment is calibrated and de-embedding is performed, as described in the following. Details on these procedures are given in Appendix G.3.

- Turn on the measurement equipment.
- Turn the probe station slot levers all the way up and back.
- Mount GSG probes on the opposing mounting slots.
- Attach 25 cm cables to probes and clamp these to prevent tension on probes.
- Connect 200 cm cables between the measurement equipment and the 25 cm cables.

- Mount ISS on probe station and apply vacuum control to stabilize the wafer.
- Gently move levers toward the ISS.
- Verify planarity of the probes on the ISS.
- Initialize the network analyzer as described in Appendix G.3.3. The network analyzer is set to sweep the frequency from 200 MHz to 10 GHz at a constant power level of -10 dBm. The stepped frequency sweep is performed over 801 points. A bias voltage of 0.9 V is connected to the rear end of the network analyzer at 'Port 1 bias'. The bias is only applied during the measurement.
- Calibrate measurement equipment as described in Appendix G.3.3.
- Unmount ISS and mount chip.
- Verify planarity on the chip.
- Perform de-embedding as described in Appendix G.3.4.
- Connect probes to the inverter test bench (pay attention to skate).
- Apply V_{DD}, bias and input signal.
- Measure S-parameters and save them to file.
- Turn off input signal, bias and V_{DD}.
- Lift probes up and back.
- Unmount chip and shut down vacuum control.
- Disconnect cables.
- Gently unmount probes and place them in their protective boxes.
- Turn off measurement equipment.
- Place plastic dust cover over probe station.

G.4.3 Simulation of Measurement

As a reference a simulation is performed replicating the measurement setup. The simulation is performed in ADS using the test bench shown in Figure G.6.

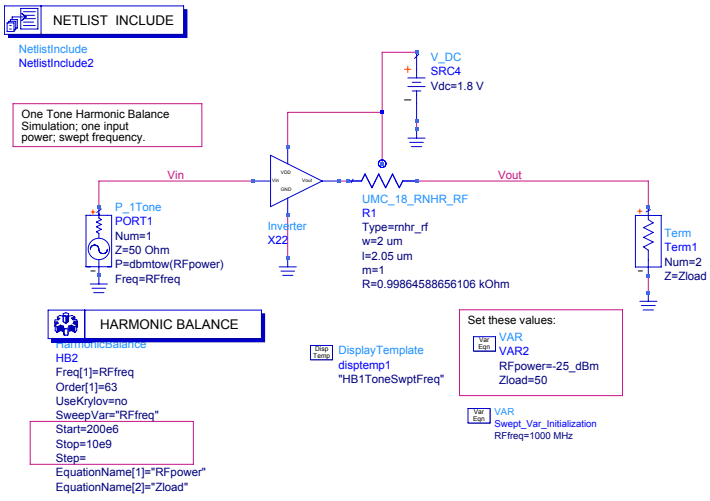


Figure G.6: *Simulated inverter measurement.*

An ideal $50\ \Omega$ source and load are used, which corresponds to the S-parameter test set. An input signal power of -25 dBm[1] and a 0.9 V DC are applied. The low input power ensures that the output is not forced into compression. Since calibration is performed, any influences from cables and probes are canceled out. Furthermore, the de-embedding procedure compensates for the parasitic effect of the pads, why this is not included in the simulation. Simulation results are presented along with measured results in the following.

G.4.4 Measurement Result

Following the step-by-step procedure given in section G.4.2, the measurement on the inverter test bench is performed. The result is presented in Figure G.7.

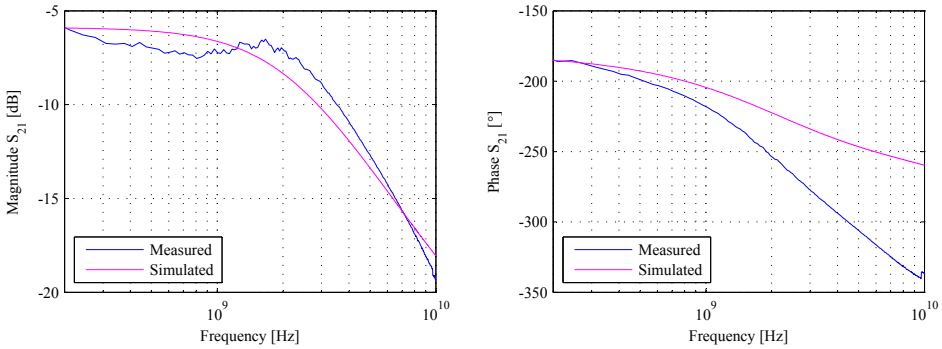


Figure G.7: *Simulated and measured results.*

The figure illustrates the simulated and measured magnitude and phase responses. The $1\ \text{k}\Omega$ resistor placed to minimize the load of the inverter, reduces the measured voltage over the 50 ohm load. Approximately one twentieth of the inverter output voltage is measured corresponding to a 26 dB reduction of the measured gain. This yields a gain of approximately 20 dB of the inverter. At 2.4 GHz the gain of the inverter is reduced with about 2 dB. The simulation shows a slightly decreased gain. Considering the phase response a phase shift of $-180\ ^\circ$ is present at low frequencies. The phase shift increases with frequency as the delay of the inverter represents a relatively larger percentage of an input period. At 2.4 GHz the phase shift is about $-260\ ^\circ$, which corresponds to a delay of 90 ps when taking the inversion of $-180\ ^\circ$ into account. The simulation result indicates a reduced slope of phase shift with frequency. This corresponds to a smaller propagation delay in the performed simulation. At 2.4 GHz the delay is simulated to 56 ps.

Adapted Simulation

With the purpose of accounting for the deviations between measurement and simulation, different capacitive effects are introduced in a simulation. In metal layer five, metal is laid out in order to fulfill minimum density requirement specified in the UMC design rules. This interconnecting metal is found under both the input and output pads, which creates a capacitive coupling between the two terminals. Applying a 60 fF capacitor yields appropriate results. However, obtaining a good match between simulation and measurement requires a large capacitive coupling to ground from the output terminal. A capacitance of 700 fF is applied in the simulation. However a coupling of this degree seems unrealistic in the actual layout. The adapted simulation result is compared with that of the measurement in Figure G.8.

[1] On the network analyzer a power level of -10 dBm is specified due to a 15 dB attenuation in the test set.

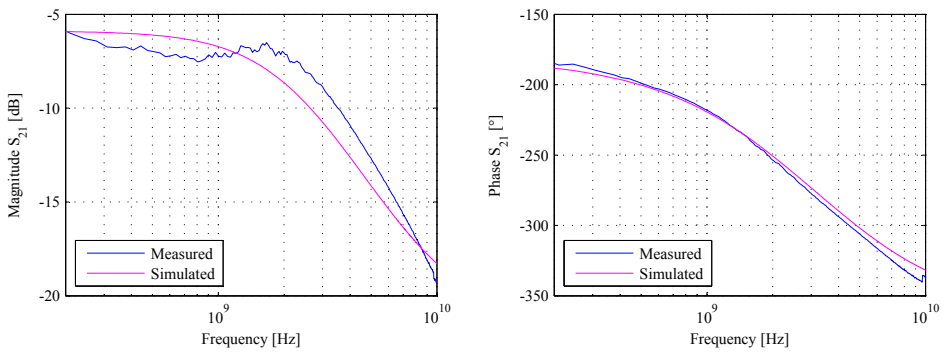


Figure G.8: *Adapted simulation and measured results.*

The simulation reflects very well the measured data, but as mentioned a coupling to ground of this size is not likely to exist. The extra metal beneath the pads corrupts the de-embedding principle, as the pads in the test bench do not match those of the de-embedding structures. For this reason de-embedding is not performed on the measured S-parameters.

G.4.5 Discussion

The measurement indicates a gain of the inverter of approximately 20 dB as expected. The large deviation in delay can be explained by the fact that delay is dependent on the input power level or voltage swing. Decreasing the power yields an increased delay. In the application of the $\Delta\Sigma$ Transmitter the delay when applying a square input signal is of interest. However, no equipment is available for performing a measurement with a high frequency square signal. Instead it would be appropriate to increase the input power level when measuring the S-parameters. This forces the output of the inverter in compression and decreases the gain to a level below unity. The measured delay would then be equivalent to the actual delay found in the application of the $\Delta\Sigma$ Transmitter. The described measurement is simulated in terms of an input power of 0 dBm. The result is presented in Figure G.9.

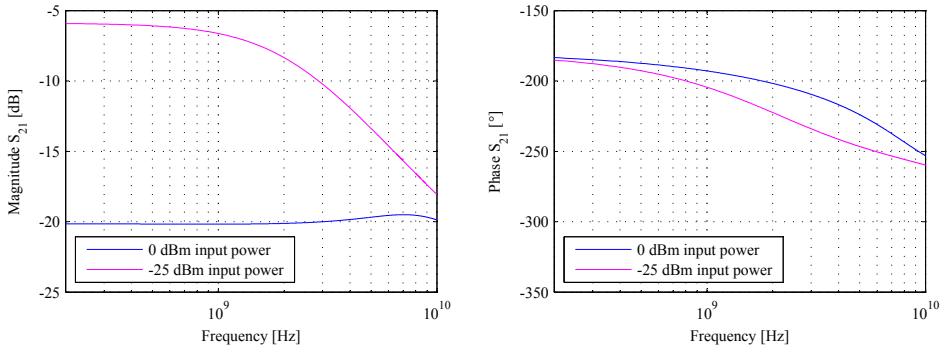


Figure G.9: *Simulation with two different input power levels.*

The delay is reduced in the case where the input power is increased to 0 dBm. At 2.4 GHz the simulated delay is 29 ps. An input power of 0 dBm corresponds to an input voltage amplitude of 0.6 V. A power of 0 dBm is just below the maximum output power level of the S-parameter test set.

G.4.6 Summary

At an input power level of -25 dBm the inverter measurement shows a gain of 20 dB as expected, however the phase response reveals a significant delay of 90 ps at 2.4 GHz. A simulation of the measurement setup shows likewise a significant delay, but not to the same extend as for the measurement. Another simulation shows that the large delay is obtained if a large coupling to ground exists from the output. However, this seems unlikely for the routed inverter test bench. Performing the measurement with a low input power ensures that the output is not forced into compression which is a condition for determining the gain of the inverter. On the downside, this eventually results in a large delay, which is not likely to exist for the inverter in the application of the $\Delta\Sigma$ Transmitter. A third simulation shows how the delay at 2.4 GHz is reduced to 90 ps, when an input power of 0 dBm is applied.

G.5 AND-gate Measurement

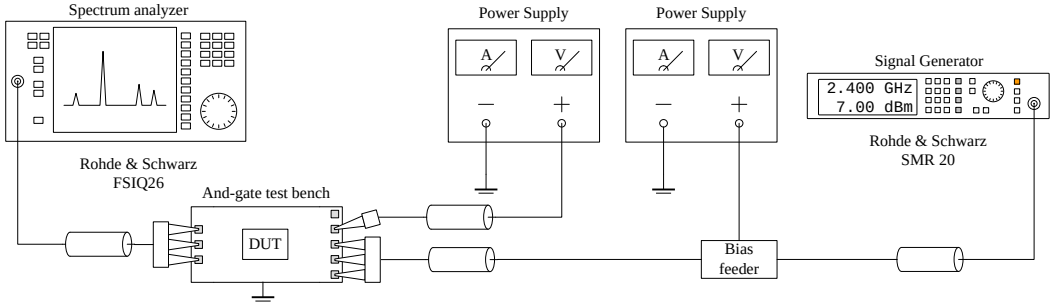
This section describes how measurements are performed on the AND-gate test bench.

G.5.1 Purpose

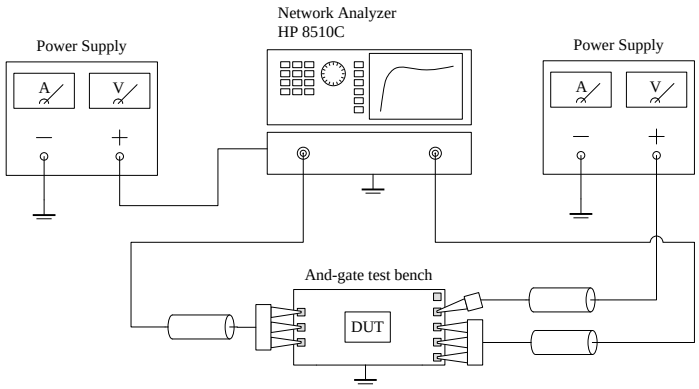
The purpose of the measurements is to determine the amplitude and phase responses of the implemented AND-gate. From the phase response, the propagation delay of the AND-gate can be calculated. Simulated and measured data is found in `[DIR038]` .

G.5.2 Setup and Measurement Procedure

Two measurements are performed on the AND-gate. One in which the input frequency is fixed and another where S-parameters are measured over a large range of frequencies. The two measuring setups are illustrated in Figure G.10.



(*a*)



(*b*)

Figure G.10: *The setup used when performing the AND-gate measurements.*

Initially, the AND-gate is connected to a spectrum analyzer (D24) for a simple verification of functionality as illustrated in G.10a. One input terminal is connected to a signal generator (D23) which applies a 2.4 GHz sine wave. Using a bias feeder (D12) a DC offset is applied. The other input terminal is tied to V_{DD}. The bias voltage and the power level of the input signal are swept while considering the output spectrum in the frequency range 0 GHz to 12 GHz. The two parameters are tuned until the fundamental tone, third and fifth order harmonics attain the highest achievable power levels.

In addition the S-parameters of the AND-gate test bench are measured. The setup for the measurement is illustrated in Figure G.10*b*. The equipment in the setup is the same as that used for the inverter measurement. The procedure of initializing the network analyzer and performing calibration is similar to that performed for the inverter measurement. The procedure is described in Appendix G.3.3. Only difference is that the power level is changed to 15 dBm, which corresponds to a test set output power of 0 dBm.

The network analyzer is set to sweep the frequency from 200 MHz to 10 GHz at a constant power level of 15 dBm, resulting in 0 dBm at the output of the S-parameter test set. The stepped frequency sweep is performed over 801 points. A bias voltage of 0.9 V is connected to the rear end of the network analyzer at 'Port 1 bias'. The bias is only applied during the measurement.

G.5.3 Simulation of the measurement

As a reference, a simulation is performed replicating the setup for the S-parameter measurement. The simulation is performed in ADS using a test bench similar to the one used for the inverter. The simulated circuit is illustrated in Figure G.11.

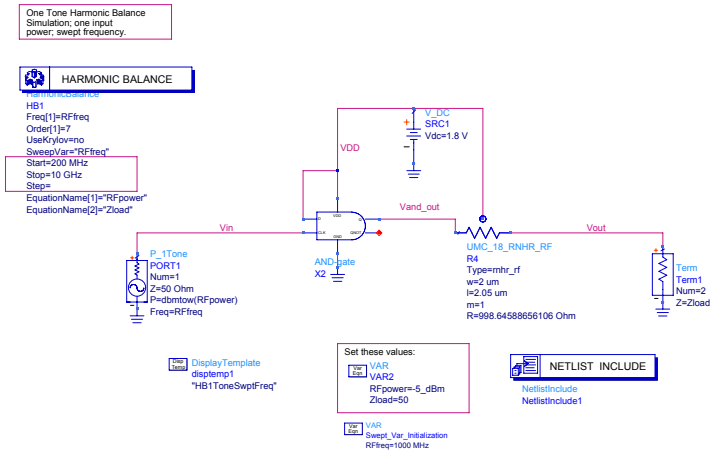


Figure G.11: *Simulated AND-gate measurement.*

The simulated results are presented along with measured results in the following.

G.5.4 Measurement Result

The results of the two measurements are presented in this section. In the first setup the input power level is, for best performance, adjusted to 7 dBm and the DC bias to 0.9 V at a fixed input frequency of 2.4 GHz. The result captured on the spectrum analyzer is shown in Figure G.12.

The significant odd harmonics of the input frequency suggest that the output is a 2.4 GHz square signal.

In Figure G.13 the result from the S-parameter measurement is presented and compared with the result obtained from simulation.

The figure illustrates the magnitude and phase response of the AND-gate test bench. As for the inverter the 1 kΩ resistor reduces the output voltage, but in this case it is of no importance. The magnitude response drops significantly between 2 GHz and 3 GHz due to insufficient input voltage swing. In the $\Delta\Sigma$ Transmitter the AND-gate inputs are square pulses, in which case the constant magnitude response is extended several gigahertz. A simulation shows a constant 0 dB gain of the AND-gate well above

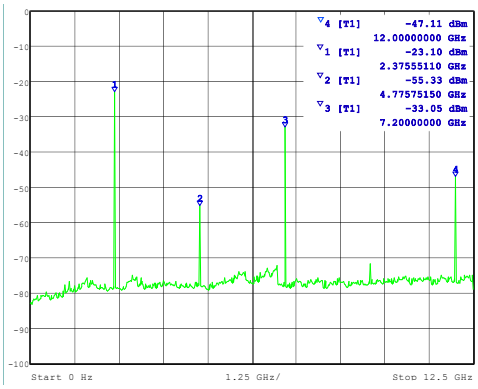


Figure G.12: *Measured output spectrum from the AND-gate when a* 2.4 GHz *signal is applied.*

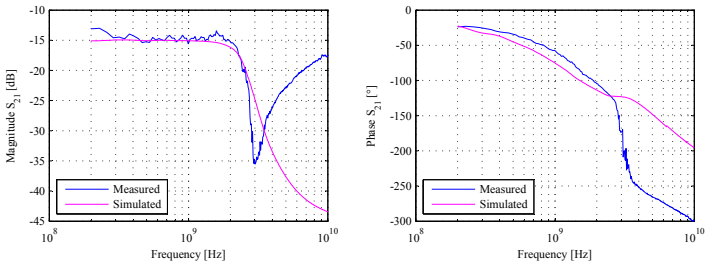


Figure G.13: *Results from simulated and measured S-parameters.*

4 GHz when applying an input power of 5 dBm corresponding to a sine wave with an amplitude of 0.9 V.

At 3 GHz the measured magnitude increases significantly in contrast to the simulated magnitude. Likewise in the phase response, the measured and simulated data agree fairly until 3 GHz. From the measured phase response a delay of 144 ps at 2.4 GHz is observed.

Adapted Simulation

With the purpose of accounting for the deviations between measurement and simulation, the known coupling between the input and output pads, found in both the layout of the inverter and AND-gate, is included in the simulation. For an appropriate match, a 60 fF capacitor is inserted between the input and output. When approximating the area of ME5 underneath each pad, the plate capacitance is equivalent to 40 fF . The simulation result is presented in Figure G.14.

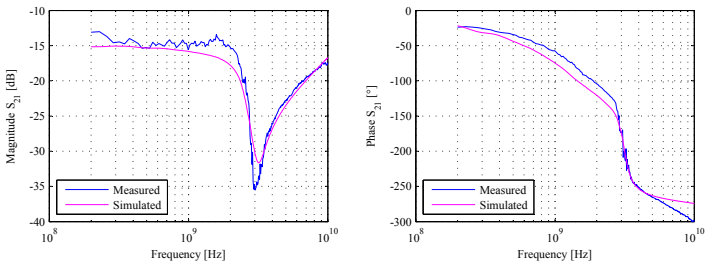


Figure G.14: *Adapted simulation and measured results.*

From the simulation, it is concluded that the inappropriate behavior of the AND-gate test bench is caused by the capacitive coupling between input and output. At high frequencies a large part of the

input signal couples directly to the output causing the significant increase in the magnitude response. As for the inverter, de-embedding is not performed, as the metal underneath the pads corrupts the de-embedding principle.

G.5.5 Summary

The functionality of the AND-gate is as expected, however a delay of 144 ps at 2.4 GHz is significantly larger than expected. As for the inverter, the reason for this is the fact that delay increases when a small voltage amplitude is applied.

Due to an inappropriate layout, a relatively large capacitive coupling exists directly from the input pads to the output pad of the AND-gate test bench. This yields great deviation between measured and simulated results at high frequencies.

G.6 Clocked Comparator Measurement

G.6.1 Purpose

The purpose of the measurement is (i) to measure the output spectrum of the comparator when a single tone is applied (ii) to measure the ELD of the clocked comparator.

G.6.2 Functionality Test

Initially the comparator is tested in order to verify that the circuit is functioning as intended. The measurement is performed using the setup depicted in Figure G.15.

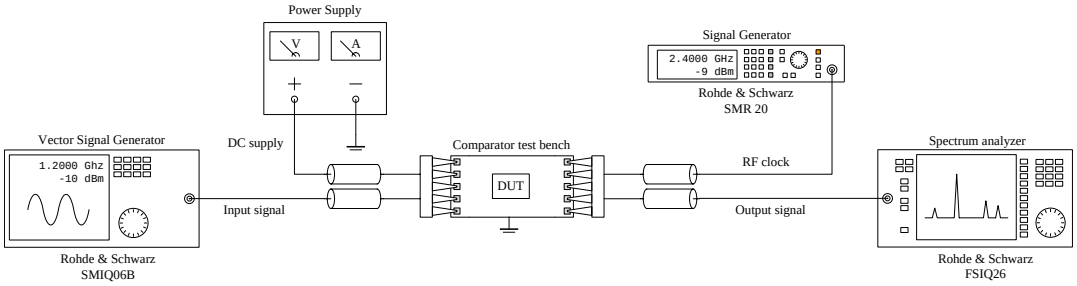


Figure G.15: *Measurement setup used in the functionality test.*

Two signals are applied; the clock signal of 2.4 GHz generated by D26 and the data signal presented by a sine wave of 1.2 GHz (D23). The power of the two tones is set to -9 dBm and -10 dBm, respectively, and they are both applied a DC offset of 0.9 V using bias feeders (D12 and D13). The output is measured using a spectrum analyzer (D24), where the fundamental and the harmonics are observed. The simulation results are presented in the following section.

G.6.3 Simulation of the Functionality Test

The measurement setup is replicated in ADS. The performed simulation is an Harmonic Balance simulation of 15^{th} order. The simulation setup is located in `[DIR029]` . The simulation result is reproduced in Figure G.16.

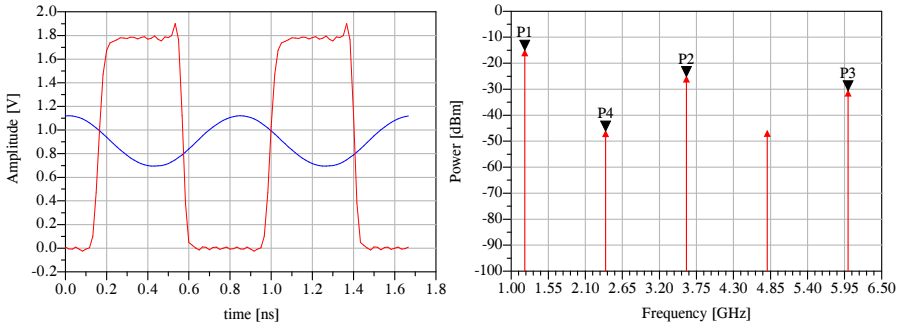


Figure G.16: *Simulation of clocked comparator measurement setup. Left: Time domain presentation of the output (red), and the data input (blue). Right: The fundamental and harmonics at the output of the simulation setup. Power levels: P1:*-15.0 dBm*, P2:*-25.1 dBm*, P3:*-30.6 dBm*, P4:*-46.1 dBm*.* `[DIR029]`

From Figure G.16 it is seen that the output signal from the comparator is a square wave as expected. This is also indicated in the spectrum where the odd harmonics clearly are present. The input signal is a small sine wave, which before it is sampled, is amplified in two inverters. It is important that the sine wave is added the 0.9 V DC potential.

G.6.4 Procedure and Setup for Measuring ELD

The following section describes how the delay of the clocked comparator can be measured using the HP8510 S-parameter test set. The principle of the measurement is explained with reference to Figure G.17.

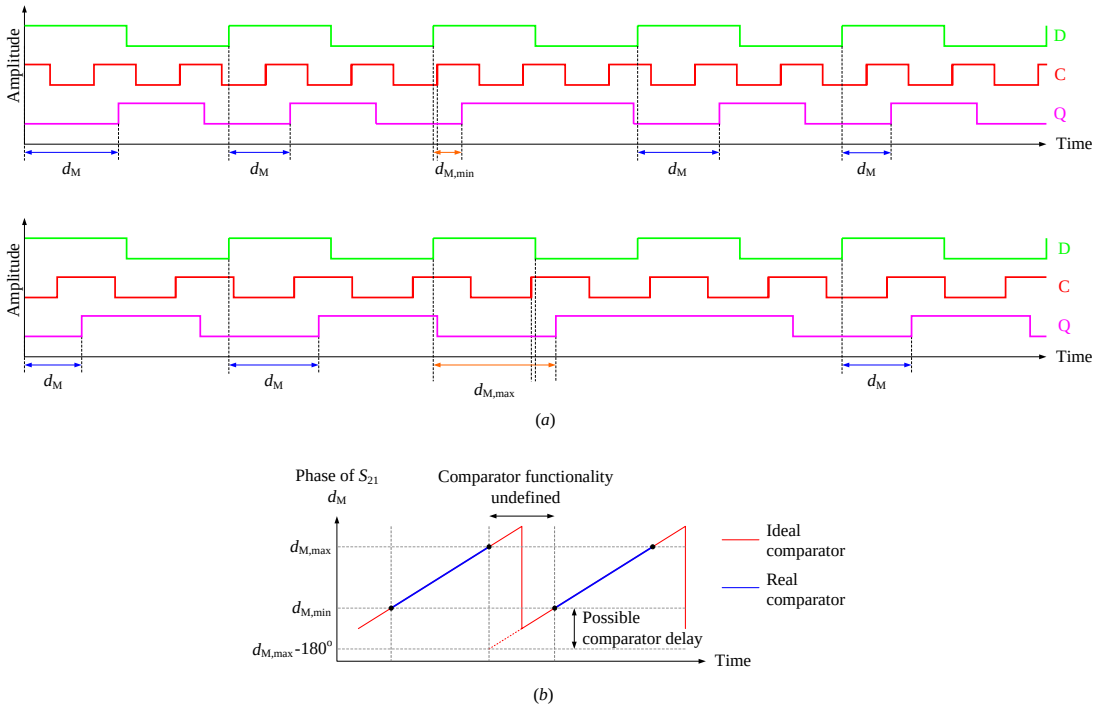


Figure G.17: *ELD measuring principle. (a) Time domain signal illustrating the principle of measuring the clocked comparator delay. (b) Time domain representation of the measured delay between D and Q, from which the minimum and maximum clocked comparator delays are determined.*

The input signal D is sampled on the rising edges of the clock. If the frequency of the clock is not an integer multiple of the input frequency, the sampling instance changes position in the input period, as depicted in Figure G.17a. This constantly changing delay causes the measured delay d_{M} to change in each period of the input signal, until a limit value is reached. The measured delay is the delay between D and Q, and the goal of the measurement is to make d_{M} equal to the delay between C and Q, which corresponds to the clocked comparator delay d_{CC}. The smallest measurable delay $d_{\mathrm{M,min}}$ occurs when the rising edge of the clock is placed just in the beginning of an input pulse, as indicated in the upper part of Figure G.17a. The largest measurable delay $d_{\mathrm{M,max}}$ occurs when the rising edge of the clock is placed just in the end of an input pulse, as illustrated in the lower part of Figure G.17a. In principle, the minimum delay $d_{\mathrm{M,min}}$ corresponds to the comparator delay d_{CC}. The maximum delay $d_{\mathrm{M,max}}$ corresponds to $180^\circ + d_{\mathrm{CC}}$, why the comparator delay d_{CC} is directly given from the measurements.

Unfortunately, the performance of the clocked comparator is compromised when a rising edge in C arrives too early, or too late, in a high input pulse. This limits the minimum and maximum delays that are possible to measure.

In the duration between the occurrence of the maximum and the minimum measurable delay, the performance of the real clocked comparator cannot be predicted, which is illustrated in Figure G.17*b*. However, the delay of the clocked comparator can be derived from the minimum and maximum measured delays, but the result will be associated with some tolerance. The best prediction of the delay given by this measurement is the mean of the two delays, when the $180°$ phase shift, and the two inverters placed prior to the clocked comparator are taken into account.

In Figure G.18 the measurement setup is depicted.

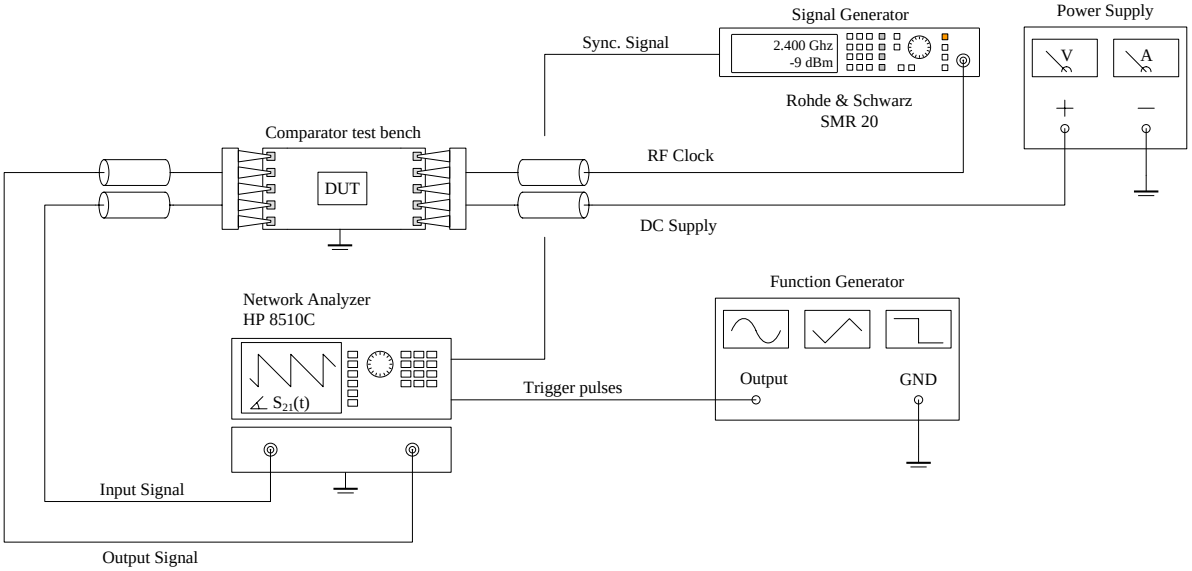


Figure G.18: *The setup used for testing the clocked comparator.*

Besides the DC voltage supplies, two input signals are needed; the input and the clock signals. The input signal is generated by the S-parameter test set and the frequency is set to 1.2 GHz. The clock signal is generated by another signal generator, which is synchronized to the S-parameter test set. Both are applied a DC offset of 0.9 V. By setting the frequency of the clock generator slightly higher than 2.4 GHz, the comparator delay can be determined from the phase of S_{21} if it is measured as a function of time. Finally an external trigger is connected to the S-parameter test set. Its purpose is described in the following section.

Measuring in time domain using the HP8510

The HP8510 is supposed to measure the S-parameters as a function of frequency and power. To perform a measurement in the time domain some tricks are necessary. The S-parameter test set can sweep three different parameters. The frequency, the power, and an external voltage. In our case the voltage will be swept in order to keep both the power and frequency levels constant. Now it is possible to measure at a constant frequency and power level, but it is still possible to perform a sweep of the maximum of 801 points.

Using an external trigger it is possible to control when the S-parameter test set initiates a measurement. The frequency of the trigger will then define the time reference. The maximum allowed frequency $f_{\text{trig,max}}$ of the trigger signal is measured to 85 Hz.

From the maximum trigger frequency, and the frequency of the clock and data signal, it is possible to determine the exact trigger frequency needed to measure the phase signal a certain number of times. The calculation of the trigger frequency is performed by the Matlab script located in `[DIR027]` ,

and is not described in further detail. If the frequency offset is 400 Hz and the needed resolution is 10 points per period of the phase signal, the trigger frequency should be 64.5 Hz.

G.6.5 Simulation of the ELD Measurement

To validate the method presented in Section G.6.4, the setup is simulated using ADS. The designed comparator is simulated using two square wave generators, and the output is stored and processed in Matlab. The simulation setup is depicted in Figure G.19.

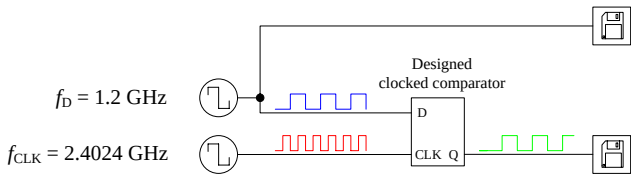


Figure G.19: *The simulation of the clocked comparator measurement is performed using the designed comparator and two square wave generators. The clocked comparator data input is given by a square wave signal of* 1.2 GHz*, and the clock signal is set slightly higher than twice the input frequency. The setup is simulated using a transient analysis.*

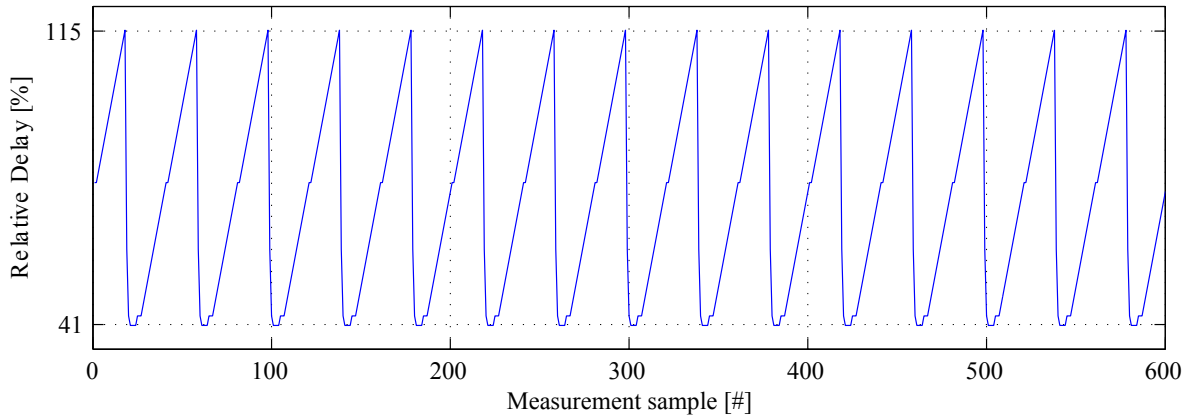


Figure G.20: *The simulated delay is plotted for each rising edge in the input. The delay is indicated relatively to a clock period. According to Section G.6.4, the delay is between* 15 % *and* 41 % *corresponding to* 63 ps *and* 171 ps*, respectively.* `[DIR027]`

The Matlab script calculates the time duration between a rising edge in the data input and the corresponding rising edge in the output, as described in Section G.6.4. The script then plots the simulated delay, which is presented in Figure G.20. The simulation shows that the comparator delay is between 63 ps and 171 ps. From the maximum and minimum simulated delay, a mean value of 112 ps is calculated. This corresponds fairly well with a mean delay of 95 ps obtained directly from the clock to the output in the simulation, why the measurement setup is validated.

G.6.6 Measurement Result

The output spectrum from the functionality test is depicted in Figure G.21.

The result is discussed in the following section.

It has not been possible to obtain a reasonable result in the measurement of the comparator delay. The reason for this is discussed in the following section.

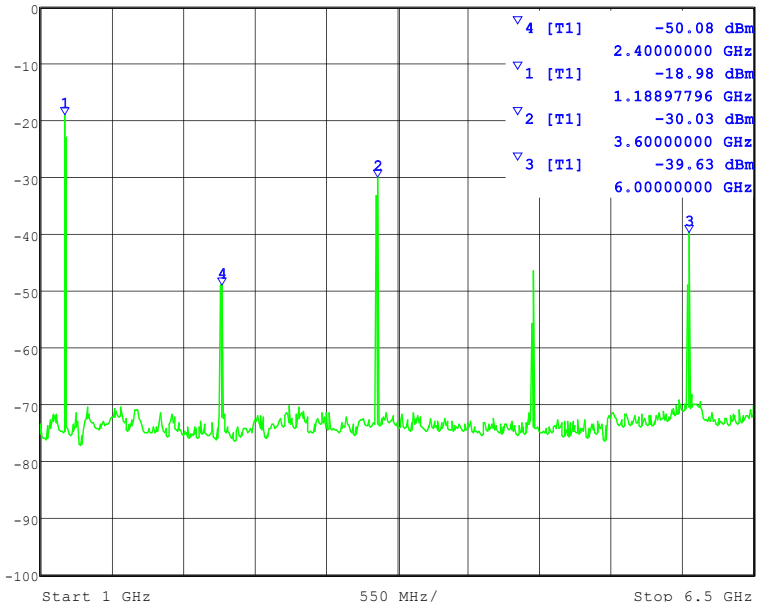


Figure G.21: *Measured output spectrum from the comparator when a* 1.2 GHz *data signal and a* 2.4 GHz *clock signal are applied.*

G.6.7 Discussion of Results

Functionality test

The measurement and the simulation result of the functionality test show good correlation. The simulated power is around 5 dB higher than the measured. The deviation is caused by the cables used to connect the comparator to the spectrum analyzer, which introduces an attenuation of approximately 2 dB. And the remaining 3 dB attenuation is probably introduced in the spectrum analyzer due to measurement tolerances, and the wide video and resolution bandwidths used for the measurement.

Measurement of ELD

It has not been possible to perform the measurement with satisfactory result. In the time allocated for the measurement it has not been possible to verify the measurement setup. During the measurement, the spectrum analyzer was connected to the output of the comparator instead of port 2 of the S-parameter test set. Here it was possible to get a fundamental tone where the power varied over time, indicating a valid measurement principle. But when the output was measured using the S-parameter test set the measured phase resembled white noise.

The measurement time of the S-parameter test set is the most likely source of error in this measurement. As previously described, the S-parameter test set performs a measurement every 11 ms, and due to the construction of the S-parameter test set, the signal at port 2 will be measured in some fraction of the 11 ms, but it has not been possible to determine the exact value. Due to the fact that the measurement time is not sufficiently short, and that the delay varies over time, another tolerance is added to the measurement. Furthermore, if the period of the phase signal is not sufficiently large compared to the 11 ms, that tolerance will corrupt the measurement.

If a drift of 10° can be accepted during one measurement, and the measurement time is 11 ms, then the period of the phase signal should be 2.5 Hz, corresponding to setting the clock frequency to 2.4 GHz + 1.25 Hz. It must be expected that the two signals are influenced by some phase noise which presumably

would be significantly larger than the frequency of just 1.25 Hz, which includes an additional tolerance to the measurement.

In order to verify that it is possible to lock two RF frequencies that close to each other and obtain the period of 400 ms of the phase signal, the following setup should be measured.

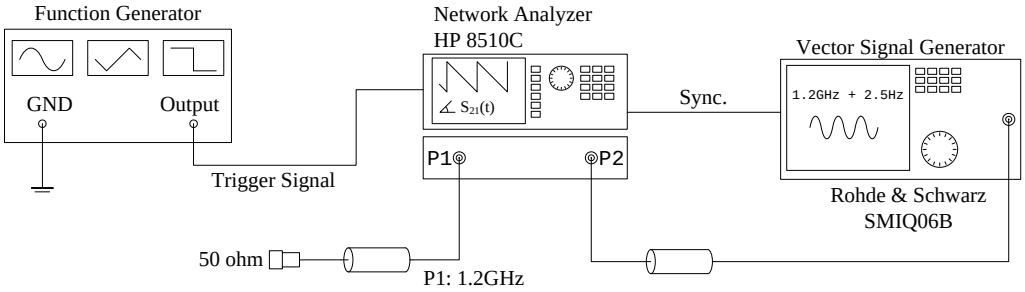


Figure G.22: *Simplified setup of the ELD measurement, used to determine if the actual measurement is possible to perform.*

The setup will result in the varying phase, as for the real measurement, but the setup is simplified compared to the setup presented in Figure G.18. The signal generated by the S-parameter test set is set to 1.2 GHz and terminated in a $50\ \Omega$ termination. The input at port 2 is supplied by the SMIQ. The frequency is 1.2 GHz plus the offset of 2.5 Hz. The result is a drifting delay, as in the real measurement, and the period of the phase signal will be 400 ms.

This measurement replicates the input and output signals to the S-parameter test set, which will clarify if it is possible to perform the measurement on the comparator. If more time had been available this measurement would have been performed in order to validate the presented method.

G.6.8 Summary

From the performed functionality test it is verified that the comparator functions as intended, and the measured performance corresponds satisfactory with the simulated result. On this basis it is concluded that the comparator functions as intended. It has not been possible to measure the delay from the clock input to the output of the comparator. Due to time constraints it has not been possible to realize the intended measurement setup and complete the measurement with satisfactory result.

Matlab Source Code

H

This appendix contains the Matlab source code used for performing system level simulation of the $\Delta\Sigma$ Transmitter. It also includes functions used to generate various plots.

For performing a system level simulation, the script `main.m` is evoked. The user is subsequently requested to specify which initialization file to run. The initialization files used for the documentation are listed in Table H.1.

Reference	m-file
[Init 1] see page 147	`initialization.m`
[Init 2] see page 9	`FIG_DSMArchitecturePrinciple.m`
[Init 3] see page 8	`FIG_DSM_time_domain.m`
[Init 4] see page 10	`wideDSTout.m`
[Init 5] see page 11	`DST_principle.m`
[Init 6] see page 19	`spectrum_vs_intOversampling.m`
[Init 7] see page 20	`EVM_vs_intOversampling.m`
[Init 8] see page 20	`SpectrumvsFilterGain.m`
[Init 9] see page 20	`SpectrumvsFilterCutoffFreq.m`
[Init 10] see page 21	`IdealReferenceSimulation.m`
[Init 11] see page 22	`impairmentHysteresis.m`
[Init 12] see page 23	`impairmentELD`
[Init 13] see page 23	`impairmentFilterSlewrate.m`
[Init 14] see page 24	`impairmentCompSlewrate.m`
[Init 15] see page 24	`impairmentANDSlewrate.m`
[Init 16] see page 56	`EffectFromELDCompensation.m`
[Init 17] see page 25	`impairmentUGBW.m`
[Init 18] see page 52	`DST1_Model.m`
[Init 19] see page 44	`comparatorModel.m`
[Init 20] see page 47	`ANDgateModel.m`
[Init 21] see page 36	`filterModel.m`

Table H.1: *Matlab initialization files used to generate plots throughout the report.*

```
%%-------------------------------------------------------------------------------------------
%%  FILENAME: main.m
%%-------------------------------------------------------------------------------------------
clc
clear all
close all
addpath('functions')

%%-------------------------------------------------------------------------------------------
%% Choose the initialization file
%%-------------------------------------------------------------------------------------------

%% get default value
run('./init/default.m');

%% The created initialization files for simulation setup
Filename{1} =  'initialization';
Filename{2} =  'FIG_DSMArchitecturePrinciple';
Filename{3} =  'FIG_DSM_time_domain'; % Time domain plot for Transmitter Architecture
Filename{4} =  'wideDSTout'; % Illustrates unsymmetrical PSD
Filename{5} =  'DST_principle'; % Show the principle
Filename{6} =  'spectrum_vs_intOverSampling'; % Ideal Matlab PSD sweeping OS rate
Filename{7} =  'EVM_vs_intOverSampling'; % EVM and Spectra vs. OS
Filename{8} =  'SpectrumvsFilterGain'; % Spectrum and EVM vs. Filter Gain
Filename{9} =  'SpectrumvsFilterCutoffFreq'; % Spectrum and EVM vs. Filter Cut-off
Filename{10} = 'IdealReferenceSimulation'; % Ideal Reference Simulation
Filename{11} = 'impairmentHysteresis'; % Evaluates the effect of hysteresis
Filename{12} = 'impairmentELD'; % Evaluates the effect of ELD
Filename{13} = 'impairmentFilterSlewrate'; % Effect of limited slew rate in Filter
Filename{14} = 'impairmentCompSlewrate'; %limited slew rate in Comp.
Filename{15} = 'impairmentANDSlewrate'; % limited slew rate in AND-gate.
Filename{16} = 'EffectFromELDCompensation';% Evaluates effect from ELD comp.
Filename{17} = 'impairmentUGBW'; % Evaluates impairment of limited UGBW
Filename{18} = 'DST1_Model'; % DST1 behavioural model
Filename{19} = 'comparatorModel'; % Clocked comparator behavioral model
Filename{20} = 'ANDgateModel'; % AND-gate behavioral model
Filename{21} = 'filterModel'; %Filter behavioral model


fprintf('-------------------------------------------------------\n')
fprintf('Select the initialization file you want to excecute : \n')
fprintf('-------------------------------------------------------\n')

for n = 1:max(size(Filename));
    fprintf('FileNr %d: %s \n',n,Filename{n})
end

fprintf('-------------------------------------------------------\n')
fprintf('File to excecute [%d]?  ',DefaultValue)
FileEntry = input('');

if isempty(FileEntry) == 1
    FileEntry = DefaultValue;
else
    % Save the new default value
    preGen = fopen('.\init\default.m', 'w');
    fprintf(preGen,'DefaultValue = %d;',FileEntry);
    fclose(preGen);
end

% excecute the selected file
eval(sprintf('File = ''./init/%s'';',Filename{FileEntry}));
eval(sprintf('SweepFile = ''%s'';',Filename{FileEntry}));

fprintf('\n')
% Clear temp variables, and clear the workspace
clear n DefaultValue FileEntry preGen
clc


%%-------------------------------------------------------------------------
%% (0) Simulation initiated
%%-------------------------------------------------------------------------
fprintf('(0) Simulation initiated \n')
% run the chosen init file
run(File)


% Present the color syntax and define indexes for subplots
run('functions/ColorSyntax_Plots')

    %%-------------------------------------------------------------------------
    %% Setup the Sweeper
    %%-------------------------------------------------------------------------
    eval(sprintf('Sweep = %s;',SweepRange))
    LengthSweep = length(Sweep);

    for nSweep = 1:LengthSweep % The overall sweep loop
        if length(Sweep) > 1 % a sweep is performed
            clc             % clear the workspace
            clear Signal

            % Print information conserning the Sweep to the screen
            fprintf('(0) Simulation initiated \n')
```



```
 95              fprintf('    Sweeping %s \n', ParmeterToSweep)
 96              fprintf('    Sweep %d of %d \n', nSweep, LengthSweep)
 97
 98              % Correst the swept parameter value
 99              SweepData.SweepParm = ParmeterToSweep ;
100              SweepData.ParameterPlotName = ParameterPlotName;
101              eval(sprintf('SweepData.Value = %s;',SweepRange))
102              eval(sprintf('%s = %d',ParmeterToSweep,Sweep(nSweep)))
103          end
104
105  %%-------------------------------------------------------------------------
106  %% (1) Signal Generator
107  %%-------------------------------------------------------------------------
108  if intBreakLevel >= 1;
109      fprintf('\n(1) Signal Generator \n')
110
111      % Excecute m file
112      SignalGenerator
113  end
114
115  %%-------------------------------------------------------------------------
116  %% (2) Delta-Sigma Modulator
117  %%-------------------------------------------------------------------------
118  if intBreakLevel >= 2;
119      fprintf('\n(2) Delta-Sigma Modulator \n');
120
121      % Excecute m file
122      DeltaSigmaModulator;
123  end
124
125  %%-------------------------------------------------------------------------
126  %% (3) EVM Calc
127  %%-------------------------------------------------------------------------
128  if intBreakLevel >= 3;
129      fprintf('\n(5) TX-Filter \n');
130
131      % Excecute m file
132      DuplexFilter
133  end
134
135  if intBreakLevel < 3 && length(Sweep) > 1
136      %Sweep is on but we do not want to simlate EVM
137      EVM.EVMRMS = 0;
138  elseif intBreakLevel >= 3
139      simulateEVM
140  end
141
142  %%-------------------------------------------------------------------------
143  %% Save the EVM and the output Spectrum of one itteration
144  %%-------------------------------------------------------------------------
145  if length(Sweep) > 1
146      eval(sprintf('SweepData.EVM%d = EVM.EVMRMS;'             ,nSweep))
147      eval(sprintf('SweepData.PSD_DSM%d = Signal.PSD_DSM;'     ,nSweep))
148      eval(sprintf('SweepData.PSD_DSMf%d = Signal.PSD_DSMf;'   ,nSweep))
149      eval(sprintf('SweepData.PSD_AND%d = Signal.PSD_AND;'     ,nSweep))
150      eval(sprintf('SweepData.PSD_ANDf%d = Signal.PSD_ANDf;'   ,nSweep))
151      eval(sprintf('save ./signal/sweep%s', SweepFile))
152      clear EVM Signal
153  end
154
155  end % end of the sweep plan
156
157  %%-------------------------------------------------------------------------
158  %% Plot the result from the parameter sweep
159  %%-------------------------------------------------------------------------
160  if length(Sweep) > 1
161      PlotSweepData
162  end
163
164  %%-------------------------------------------------------------------------
165  %% Run the initialization file again, to modify plots
166  %%-------------------------------------------------------------------------
167  if exist('printFigureToFile') == 1
168      run(File)
169  end
```



```
%%-------------------------------------------------------------------------
% Help for input declarations: initialization
%%-------------------------------------------------------------------------
%
% In this file the needed input declarations are given.
% Every configurable/tunable parameter for the simulation is given here.
% The simulation will be based on different sets of files, which give
% different reproductable results.
if exist('initialization') == 0
fprintf('    Using "initialization.m" parameters \n')
end
%%-------------------------------------------------------------------------
%% Global Constants
%%-------------------------------------------------------------------------

intBreakLevel = 3; % Determines where the code execution breaks.
                   % 1 : Break after signal generator
                   % 2 : Break after DSM
                   % 3 : Break after EVM calculation (Used for Sweep)

%%-------------------------------------------------------------------------
% Variables for SignalGenerator
%%-------------------------------------------------------------------------

Vdd = 1.8;                  % Power supply
intFCarrier = 2.4e9;        % Carrier frequency [Hz]
intOverSampling = 200;      % Oversampling of Fc [gg]
intSignalLength = 1e3;      % Number of samples (pre-upsampling)
intFFT = 10;                % Minimum number of samples in
                            % signal bandwidth when performing FFT

%%-------------------------------------------------------------------------
%% Variables for the Delta-Sigma Modulator
%%-------------------------------------------------------------------------

%%-----------------------------------------------------------
%% Summer parameters:
%%-----------------------------------------------------------

SummerMAXout = Vdd;
SummerMINout = 0;

%%-----------------------------------------------------------
%% Filter 1 parameters:
%%-----------------------------------------------------------

% The normalized cut-off frequency ( f_cutoff / (intFs/2) )
Filterparms.Fn        = 1e6;
Filterparms.A         = 100;
Filterparms.MAXout    = Vdd-0.1;
Filterparms.MINout    = 0.1;

%%-----------------------------------------------------------
%% Edge triggered comparator parameters:
%%-----------------------------------------------------------

% Symetric hysteresis, arund 0.9 V
% hystlevel = 0;    % Remove If not used
% or Asymetric hysteresis, set offset from 0.9V both positive
hystlevel_low = 0;  % Remove If not used
hystlevel_high = 0; % Remove If not used

%%-----------------------------------------------------------
%% Slew Rate Settings
%%-----------------------------------------------------------
% Insert the Slew Rate in V/s
% No slew rate is applied if SR is equal to zero

Filter_SR_L_to_H = 0;%[V/s]
Filter_SR_H_to_L = 0;%[V/s]
ClComp_SR_L_to_H = 0;%V/s]
ClComp_SR_H_to_L = 0;%[V/s]
AND_SR_L_to_H    = 0;%[V/s]
AND_SR_H_to_L    = 0;%[V/s]

%%-----------------------------------------------------------
%% Delay Settings
%%-----------------------------------------------------------
InverterDelay = 0;          % Delay from input of comparator to latch
                            % Specified in seconds
LatchDelay = 0;             % Delay from latch to output of quantizer
                            % Specified in seconds
ANDDelay = 0;               % Delay in AND gate
                            % Specified in seconds
ClockDelay = 1e-12;         % The time difference between the
                            % Latch clock and the AND clock
PhaseDelay = 0;             % The delay of the Basband phase to
                            % keep the phase and envelope in sync.

%%-----------------------------------------------------------
%% For system excess loop delay compensated:
%%-----------------------------------------------------------
GainFeedbackLatch = 0;
GainFeedbackAND = 2;
```



```
 95
 96 %%-------------------------------------------------------------------------
 97 %% System control flags
 98 %%-------------------------------------------------------------------------
 99 intSGmode = [2];             % The selected modulationscheme.
100                               %   1 - Corresponds to a 16QAM signal,
101                               %       for illustration only (2000 samples)
102                               %   2 - WLAN burst
103                               %       Raw random signal, with no preamble
104 %-------------------------------------------------------------------------
105
106 intDSMmode = [1 2 1 1 1 0 0]; % First element - The summer
107                               %   1 - Ideal summer is used
108                               % Second element - The filter
109                               %   1 - An integrator is used
110                               %   2 - First order filter is used
111                               %   3 - Second order filter is used
112                               %   4 - Third order filter is used
113                               % Third element - The quantizer
114                               %   1 - An ideal Quantizer is used
115                               %   2 - Quantizer with hysteresis is used
116                               % Fourth element - Order of SD modulator
117                               %   1 - First order SD-modulator
118                               %   2 - Second order SD-modulator
119                               % Fifth element - The second filter
120                               %   1 - An integrator 2 is used
121                               %   2 - First order filter is used
122                               %   3 - Second order filter is used
123                               %   4 - Third order filter is used
124                               % 6 element - Limit Output Signal Swing (LOSS)
125                               %   0 - Do nothing
126                               %   1 - The Summer output is LOSS
127                               %   2 - The filter output is LOSS
128                               %   3 - Summer and filter is LOSS
129                               % 7 element - Modulator type selection flag
130                               %   0 - Modulator 1 feedback from latch
131                               %   1 - Modulator 2 feedback from both AND gate
132                               %       and Latch
133 %-------------------------------------------------------------------------
134 intANDmode = [2];            % Declares the AND gate model
135                               %   1 - Ideal AND gate is used
136                               %   2 - AND gate with finite fall and rise
137                               %       times are used
138 %-------------------------------------------------------------------------
139 intDFmode = [1];             % Declares the Duplex Filter model
140                               %   1 - Ideal DF is used
141 %-------------------------------------------------------------------------
142
143 %-------------------------------------------------------------------------
144 % Declare Destination for the plot.
145 %-------------------------------------------------------------------------
146
147 % Each plot is created from a vector with 4 elements:
148 %    1 - The Figure window where the curve is plotted.
149 %    2 - The Subfigure, where the plot is made.
150 %    3 - Number of rows in the figure window.
151 %    4 - Number of columns in the figure window.
152
153 %% Plots from the signal generator
154 plotSG = [transpose([1 1 4 1])...  % Amplitude of the input signal
155           transpose([1 2 4 1])...  % Phase of the input signal
156           transpose([1 3 4 1])...  % The phase modulated clock signal
157           transpose([3 1 3 1])];   % The PSD of the complex input signal
158 %% Plots fom the Delta-Sigma modulator
159 plotSDM = [transpose([2 1 5 1])... % The input signal to the DSM
160            transpose([2 2 5 1])... % The summer output signal
161            transpose([2 3 5 1])... % The filter output signal
162            transpose([2 4 5 1])... % The output signal from the comparator
163            transpose([0 1 1 1])... % The quantization error
164            transpose([3 2 3 1])... % The Delta-Sigma modulator output PSD
165            transpose([0 1 1 1])];  % The filter impulse response.
166 %% Plots from the AND gate
167 plotAND = [transpose([2 5 5 1])... % The time domain output from AND gate
168            transpose([3 3 3 1])];  % The AND gate output PSD
169
170 %% Plots from the TX filter
171 plotTX =  [transpose([0 1 1 1])];
172 %% Plots from the EVM
173 plotEVM = [transpose([4 1 3 2])... % Simulated VS Ideal Amplitude response
174            transpose([4 2 3 2])... % Simulated VS Ideal Phase response
175            transpose([4 3 3 2])... % Corrected Amplitude response
176            transpose([4 4 3 2])... % Corrected phase response
177            transpose([4 5 3 2])... % EVM as a function of time
178            transpose([4 6 3 2])... % Plot the PSD of the demodulated signal
179            transpose([4 6 3 2])];  % Plot the filter transferfunction
180 % Plots from the Sweep
181 plotSWE = [transpose([0 1 3 1])...
182            transpose([0 2 3 1])...
183            transpose([0 3 3 1])...
184            transpose([0 3 3 1])];
185
186 % add smoothed, PSD to powerspectrum
187 plotIPSD = [transpose([0 0 0 0])...
188             transpose([0 0 0 0])];
```



```
%%-------------------------------------------------------------------------------------------
%% FILENAME: SignalGenerator.m
%%-------------------------------------------------------------------------------------------

fprintf('    The signal is generated based on: \n', intOverSampling)
fprintf('    intOverSampling = %d \n', intOverSampling)
fprintf('    intSignalLength = %d \n', intSignalLength)

% Definition
intFs  = intFCarrier * intOverSampling; %The sampling frequency

%%-------------------------------------------------------------------------------------------
%% 16QAM modulation is selected
%%-------------------------------------------------------------------------------------------

if intSGmode == 1

    % Load pregenerated 16QAM signal of 2000 samples
    temp = load('.\signal\QAM16.mat');
    Signal.CMSignal = transpose(temp.QAM);

    %Specify signal BW
    intBW = 40e3;

    % Clearing temp variables
    clear temp
end

%%-------------------------------------------------------------------------------------------
%% WLAN burst with no preamble is selected.
%%-------------------------------------------------------------------------------------------

% Source: dspwmwlan.m written by Michael Nielsen
% Signal Length 51249 samples

if intSGmode >= 2
    % First it is cheked if a signal with the equal properties has been
    % generated previously

    % Predict filename from the defined constants
    filename{1} = sprintf('.\\signal\\Signal_%d_%d_%d_%d.mat'...
                          ,intSGmode,intFs,intSignalLength,intOverSampling);
    % Check if the signal has been generated previously
    preGen = fopen(filename{1}, 'r');

    if preGen ~= -1    % If the signal has been generated, load it
        %%---------------------------------------------------------------------------------
        %% Load Previously generated signal
        %%---------------------------------------------------------------------------------
        fprintf('    Loading previously generated signal\n')

        fclose(preGen);                 % Close the file before loading
        load(filename{1});              % Load the previously generated signal
        Signal.CMSignal = CMSignal;     % Store it in the signal struct
        clear CMSignal                  % Clear unused variables


    else % If the signal does not exist, generate the signal
        fprintf('    Loading "WLANBurst.mat" \n');
        fprintf('    The signal is 1 WLAN burst, with no preamble\n');
        fprintf('    The signal is upsampled to the correct Delta t\n');

        %%---------------------------------------------------------------------------------
        %% Load the WLAN signal
        %%---------------------------------------------------------------------------------

        load('./signal/WLANburst.mat');     % loads the signal into CMSignal
                                            % predefined in the file
        load('./signal/WLANBurstTime.mat')  % Loads the signal into Time
                                            % predefined in the file

        %%---------------------------------------------------------------------------------
        %% Upsampling and truncation of the signal
        %%---------------------------------------------------------------------------------

        % Calculate the timestep
        Deltat = Time(2)-Time(1);

        % Find the Upsampling factor of the WLAN signal, that gives the
        % specifed sample frequency of the final oversampled signal

        Upsampling_wlan = intFs*Deltat; % The needed upsampling factor

        % Check if the upsampling ratio of CMSignal is an integer
        if(mod(Upsampling_wlan,1)~=0)
            % Find nearest higher upsampling factor
            new_Upsampling_wlan = ceil(Upsampling_wlan);
            % Calculate corresponding sampling freq.
            new_intFs = new_Upsampling_wlan/Deltat;
            % Calculate corresponding oversampling ratio
            new_intOverSampling = new_intFs/intFCarrier;

            % Inform user
            fprintf('\n    The constant Upsampling_wlan is %f\n'...
```



```
95                  ,Upsampling_wlan);
96              fprintf('    This is not an int!\n');
97              fprintf('    Selecting an oversampling of %f results in:\n'...
98                  ,new_intOverSampling);
99              fprintf('    Upsampling: %f\n',new_Upsampling_wlan);
100             fprintf('    Sample rate: %f\n',new_intFs);
101             fprintf('    Please make the necessary adjustment.\n');
102             break
103         else % the upsampling ratio is an integer
104             fprintf('    The original WLAN signal is upsampled by a ');
105             fprintf('factor %d\n',Upsampling_wlan);
106         end
107
108     % Upsampling and truncation of CMSignal
109     % Interpolation and lowpass filtering is used
110     CMSignal = Vdd*interp(CMSignal(1:intSignalLength),Upsampling_wlan);
111     Signal.CMSignal = CMSignal;
112     save(filename{1},'CMSignal') % Store the result for recycle.
113     end
114
115     intBW = 16.6e6;      % WLAN bandwidth
116
117     % Clear temporary variables and data
118     clear n p SignalUp Upsampling_wlan Deltat fid text Time
119     clear new_intOverSampling new_intFs CMSignal Time
120 end
121
122 % Determine length of window used for computing PSD
123 % Calculate frequency resolution [Hz/sample]
124 intDeltaF = intBW/intFFT;
125 % Determine window length required [samples]
126 intWindowLength = ceil(intFs/intDeltaF);
127
128 %%-------------------------------------------------------------------------------------------
129 %% Compute polar signal
130 %%-------------------------------------------------------------------------------------------
131
132 Signal.AbsSignal = abs(Signal.CMSignal);
133 Signal.PhaseSignal = angle(Signal.CMSignal);
134
135 %%-------------------------------------------------------------------------------------------
136 %% Compute time refference
137 %%-------------------------------------------------------------------------------------------
138
139 % Construct a time vector from the length of Signal.CMSignal
140 Signal.t = transpose((0:length(Signal.CMSignal)-1)).*(1/intFs);
141
142 % Calculate sample time
143 Signal.Deltat = 1/intFs;
144
145 %%-------------------------------------------------------------------------------------------
146 %% Compute the delay in samples
147 %%-------------------------------------------------------------------------------------------
148
149 % Number of samples corresponding to delay in the Inverter:
150 intInverterDelay = ceil(InverterDelay/Signal.Deltat);
151
152
153 if sum([ClComp_SR_L_to_H ClComp_SR_H_to_L]) > 0
154     intLatchSRDelay = ceil((Vdd/max([ClComp_SR_L_to_H ClComp_SR_H_to_L]))*intFs);
155 else
156     intLatchSRDelay = 0;
157 end
158
159 % Number of samples corresponding to delay in the Latch
160 % half the SR delay is substracted meaning that the delay is between the
161 % rising edge in the clock to the latch output reaches 0.9V, as simulated
162 % when the parameter is substracted
163 intLatchDelay = ceil(LatchDelay/Signal.Deltat)-intLatchSRDelay;
164
165 % Number of samples corresonding to delay in clock
166 intClockDelay = ceil(ClockDelay/Signal.Deltat);
167
168 % Number of samples corresonding to delay in the Phase Signal
169 intPhaseDelay = ceil(PhaseDelay/Signal.Deltat);
170
171 % Number of samples corresonding to delay in AND gate
172 intANDDelay = ceil(ANDDelay/Signal.Deltat);
173
174 %%-------------------------------------------------------------------------------------------
175 %% Generate RF modulated clock
176 %%-------------------------------------------------------------------------------------------
177
178 % Delay the phase information, to keep the phase signal and envelope signal
179 % in sync.
180 PhaseSignal = transpose([1:length(Signal.CMSignal)].*0);
181 PhaseSignal(intPhaseDelay+1:end) = ...
182                             angle(Signal.CMSignal(1:end-intPhaseDelay));
183
184 % Generate the modulated cosine clock
185 Signal.CosClock = 0.99*sin((2*pi*intFCarrier).*Signal.t+PhaseSignal);
186
187 % The high and low potentials of the clock signal
188 ClockLow = 0;
```



```
189 ClockHigh = Vdd;
190
191 % Create the clock signal from Signal.CosClock.
192
193 Signal.Clock = ceil(Signal.CosClock).*Vdd;
194
195 %%-------------------------------------------------------------------------------------------
196 %% Generate The delayed clock signal
197 %%-------------------------------------------------------------------------------------------
198
199 if(intClockDelay+intLatchDelay+intLatchSRDelay>0)
200     fprintf('    Generating delayed clock signal\n')
201
202     % Delay the orignal clock signal intClockDelay+intLatchDelay samples
203     tempClock(1:length(Signal.Clock)) = 0;
204     tempClock(intClockDelay+intLatchDelay+intLatchSRDelay+1:...
205         length(Signal.Clock)) = Signal.Clock(1:...
206         length(Signal.Clock)-intClockDelay-intLatchDelay-intLatchSRDelay);
207
208     % store the signal
209     Signal.ANDClock = transpose(tempClock);
210     clear tempClock
211 else
212     Signal.ANDClock = Signal.Clock;
213 end
214
215 %%-------------------------------------------------------------------------------------------
216 %% The Remaning Code is not included, and is only used for plot generation
217 %%-------------------------------------------------------------------------------------------
```



```
%%-------------------------------------------------------------------------------------------
%% FILENAME: DeltaSigmaModulator.m
%%-------------------------------------------------------------------------------------------

%%-------------------------------------------------------------------------------------------
% Print information to the screen
%%-------------------------------------------------------------------------------------------

if intDSMmode(7) == 0
    fprintf('    Simulating a first order Delta-Sigma Loop:\n')
elseif intDSMmode(7) == 1
    fprintf('    Simulating a first order Delta-Sigma Loop with ELD compensation:\n')
end

%%-------------------------------------------------------------------------------------------
%% Calculation of constants
%%-------------------------------------------------------------------------------------------

% Set the length of the simulation
LengthSignal = length(Signal.AbsSignal);

% Set the high potential of the digital logic
QuanHigh = Vdd;

%%-------------------------------------------------------------------------------------------
%% Prepare Hysteresis
%%-------------------------------------------------------------------------------------------
% The simulator can only sweep one parameter.
% Becuase of that the hystlevel, is used when sweeping the hysteresis

if exist('hystlevel') == 1
        % Thresholds symmetrical around 0.9 V
    hystlevel_low = hystlevel;
    hystlevel_high = hystlevel;
end

%%-------------------------------------------------------------------------------------------
%% Prepare Slew Rate Limitation
%%-------------------------------------------------------------------------------------------
% The simulator can only sweep one parameter.
% Becuase of that the XXXEqualSR, is used when sweeping the SR

% Filter:
if ( Filter_SR_L_to_H > 0 && Filter_SR_H_to_L == 0 )||( exist('FilterEqualSR')== 1 )
    FilterEqualSR = 1;
    Filter_SR_H_to_L = Filter_SR_L_to_H     % Equal SR for rising and falling edge
end

% Clocked Comparator:
if ( ClComp_SR_L_to_H > 0 && ClComp_SR_H_to_L == 0 )||( exist('ClCompEqualSR')== 1 )
    ClCompEqualSR = 1;
    ClComp_SR_H_to_L = ClComp_SR_L_to_H;    % Equal SR for rising and falling edge
end

% AND-gate:
if ( AND_SR_L_to_H > 0 && AND_SR_H_to_L == 0 )||( exist('ANDEqualSR')== 1 )
    ANDEqualSR = 1;
    AND_SR_H_to_L = AND_SR_L_to_H;          % Equal SR for rising and falling edge
end

%%-------------------------------------------------------------------------------------------
%% Calculation of the filter coefficients
%%-------------------------------------------------------------------------------------------

if intDSMmode(2)>=2
  if intDSMmode(2) == 4
    FilterOrder = 1;
      fprintf('    Using 1. Order Butterworth filter, with Limited UGBW\n')
  else
    FilterOrder = intDSMmode(2)-1;
      fprintf('    Using %d. Order Butterworth filter\n',FilterOrder)
  end
    % Generates the butterworth filter transferfunction H(s)

    [b,a] = butter(FilterOrder,Filterparms.Fn*2*pi,'s');
    % Add gain to the normalized filter
    b = b * Filterparms.A;
    % Calculate the digital representation of H(s)
    [Filterparms.b,Filterparms.a] = impinvar(b,a,intFs);
else
    fprintf('    Using Integrator 1\n')
end

if intDSMmode(2)==4
    % Generates the butterworth filter transferfunction H(s)
    [b,a] = butter(1,Filter_UGBW*2*pi,'s');
    % Calculate the digital representation of H(s)
    [UGBW.b,UGBW.a] = impinvar(b,a,intFs);
    UGBW.PreviousOutput = 0;
end

%%-------------------------------------------------------------------------------------------
%% Initialization of the Delta Sigma Loop
%%-------------------------------------------------------------------------------------------
```



```
 95
 96 % Initial conditions and preallocation of variables
 97 OutSummer(1:LengthSignal) = 0;                        % Output from the Summer
 98 FilterInt(1:LengthSignal) = Vdd/2;                    % Filter output including gain
 99 FilterInt2(1:LengthSignal) = Vdd/2;                   % Filter output including gain
100 OutFilter(1:LengthSignal) = Vdd/2;                    % w. Limited voltage swing
101 InClockedComp(1:LengthSignal+intInverterDelay) = 0; % Input to the clocked comparator
102 OutClockedComp(1:LengthSignal+intLatchDelay) = 0;    % Output from the clocked comparator
103 Output(1:LengthSignal) = 0;                          % DSM output
104 Signal.AND(1:LengthSignal) = 0;                      % The signal from the AND gate
105 OutClockedCompOut(1:LengthSignal) = 0;               % The signal from the CC
106
107 % Draw the progress bar
108 fprintf('    Starting the Delta-Sigma loop:\n')
109 fprintf('    0    10   20   30   40   50   60   70   80   90  100 \n')
110 fprintf('    |    |    |    |    |    |    |    |    |    |    |\n    •')
111 PPHcomputed = 0;
112
113 %%-------------------------------------------------------------------------------------
114 %% The Delta-Sigma Loop
115 %%-------------------------------------------------------------------------------------
116 tic
117 % determine start sample
118 if intDSMmode(2) > intDSMmode(5)
119     intStart = intDSMmode(2);
120 else
121     intStart = intDSMmode(5);
122 end
123
124 for p = intStart:LengthSignal-1; % The DS loop
125
126     % Part of the progress bar, Indicates activity of the loop
127     if p>2*((LengthSignal)/100)+PPHcomputed;
128         fprintf('•')
129         PPHcomputed = PPHcomputed+0.02*LengthSignal;
130     end
131
132     %%-------------------------------------------------------------------------------
133     %% Summer models
134     %%-------------------------------------------------------------------------------
135
136     %% Ideal Summer
137     if intDSMmode(1) == 1;
138         OutSummer(p+1) = (Signal.AbsSignal(p+1) -  Output(p));
139     end
140
141     %%-------------------------------------------------------------------------------
142     %% Filter Options
143     %%-------------------------------------------------------------------------------
144
145     %% Ideal Integrator
146     if intDSMmode(2) == 1;
147         FilterInt(p+1) = FilterInt(p) + OutSummer(p+1); %* (1/(intFs))
148     end
149
150     %% 1st Order Lowpass Filter
151     if intDSMmode(2) == 2 || intDSMmode(2) == 4;
152         FilterInt(p+1) = (     Filterparms.b(1)*OutSummer(p+1)...
153                              + Filterparms.b(2)*OutSummer(p)...
154                              - Filterparms.a(2)*FilterInt(p)...
155                            ) / Filterparms.a(1);
156     end
157
158     %% 2nd Order Lowpass Filter
159     if intDSMmode(2) == 3;
160         FilterInt(p+1) = (     Filterparms.b(1)*OutSummer(p+1)...
161                              + Filterparms.b(2)*OutSummer(p)...
162                              + Filterparms.b(3)*OutSummer(p-1)...
163                              - Filterparms.a(3)*FilterInt(p-1)...
164                              - Filterparms.a(2)*FilterInt(p)...
165                            ) / Filterparms.a(1);
166     end
167     % Apply UGBW to the model
168     if intDSMmode(2) == 4;
169         FilterInt2(p+1) = (     UGBW.b(1)*FilterInt(p+1)...
170                               + UGBW.b(2)*FilterInt(p)...
171                               - UGBW.a(2)*FilterInt2(p) ...
172                             )  / UGBW.a(1);
173         OutFilter(p+1) = FilterInt2(p+1);
174     else
175     % Store the filter output
176     OutFilter(p+1) = FilterInt(p+1);
177     end
178
179     % Add Slew Rate to the filter output
180         OutFilter(p+1) = LimitedSlewRate(...
181                                          OutFilter(p+1), OutFilter(p), ...
182                                          Filter_SR_H_to_L, Filter_SR_L_to_H, intFs);
183
184     %%-------------------------------------------------------------------------------
185     %% Clocked comperator
186     %%-------------------------------------------------------------------------------
187
188     %% Ideal clocked comperator
```



```
189     if intDSMmode(3) == 1;
190         % Sample if rising edge is present, else hold previous value
191         if(Signal.Clock(p+1) == Vdd && Signal.Clock(p) == 0)
192             if OutFilter(p+1) > Vdd/2                    % Threshold voltage
193                 OutClockedComp(p+1) = QuanHigh;          % Set ouput High
194             else
195                 OutClockedComp(p+1) = 0;                 % Set output Low
196             end
197         else
198             OutClockedComp(p+1) = OutClockedComp(p);     % Else hold
199         end
200     end
201
202     %% Clocked comparator with hysteresis and delay
203     if intDSMmode(3) == 2;
204         %% Delaying signal the defined number of samples
205         InClockedComp(p+1+intInverterDelay) = OutFilter(p+1);
206         % Sample if rising edge is present, else hold previous value
207         if(Signal.Clock(p+1) == Vdd && Signal.Clock(p) == 0)
208             if OutClockedComp(p) == QuanHigh
209                 % Clock Signal is going from high state to low
210                 complevel = 0.9 - hystlevel_low;
211             else
212                 % Clock Signal is going from from low state to high
213                 complevel = 0.9 + hystlevel_high;
214             end
215
216             if InClockedComp(p+1) > complevel
217                 % Set the output High
218                 OutClockedComp(p+1+intLatchDelay) = QuanHigh;
219             else
220                 % Set the output low
221                 OutClockedComp(p+1+intLatchDelay) = 0;
222             end
223         else % Else hold and do nothing
224             OutClockedComp(p+1+intLatchDelay) = OutClockedComp(p+intLatchDelay);
225         end
226     end
227
228     % Save the DSM output.
229     OutClockedCompOut(p+1)=OutClockedComp(p+1);
230
231     % Add Slew Rate to the Clocked Comparator output
232     OutClockedCompOut(p+1) = LimitedSlewRate(...
233                                 OutClockedCompOut(p+1), OutClockedCompOut(p),...
234                                 ClComp_SR_H_to_L, ClComp_SR_L_to_H, intFs);
235
236     %%-------------------------------------------------------------------------------------
237     %% AND gate
238     %%-------------------------------------------------------------------------------------
239
240     %% AND'ning including delay
241     if OutClockedCompOut(p+1)>0.9 && Signal.ANDClock(p+1)> 0.9
242         Signal.AND(p+1+intANDDelay)=Vdd;
243     else
244         Signal.AND(p+1+intANDDelay)=0;
245     end
246
247     % Apply SR to the output signal of the AND gate
248     Signal.AND(p+1) = LimitedSlewRate(...
249                                 Signal.AND(p+1), Signal.AND(p), ...
250                                 AND_SR_H_to_L, AND_SR_L_to_H, intFs);
251
252
253     if intDSMmode(7) == 1; % The DST w. ELD compensation is selected
254         Output(p+1) = GainFeedbackLatch * OutClockedCompOut(p+1) ...
255                                     + GainFeedbackAND*Signal.AND(p+1);
256     else    % The DST is selected
257         Output(p+1) = OutClockedCompOut(p+1);
258     end
259
260 %%-----------------------------------------------------------------------------------------
261 %% Terminate the Delta Sigma Loop
262 %%-----------------------------------------------------------------------------------------
263 end
264
265 fprintf('•\n') % Ends the progress bar
266 Signal.SDM = transpose(Output); % Stores the output
267 toc
268
269
270 %%-----------------------------------------------------------------------------------------
271 %% The Remaning Code is not included, and is only used for plot generation
272 %%-----------------------------------------------------------------------------------------
```



```
%%-------------------------------------------------------------------------------------------
%% FILENAME: simulateEVM.m
%%-------------------------------------------------------------------------------------------

%%-------------------------------------------------------------------------------------------
%% System variables
%%-------------------------------------------------------------------------------------------

% The crossover frequency used when filtering the downconverted baseband
% signal
EVM.filterWn = 100e6;

% the number of samples, which is not used in the beginning and end when
% computing the EVM
if exist('intExcludeSamples') == 0
    intExcludeSamples = ceil(length(Signal.Out)*.1)
end

fprintf('    Computing EVM\n')
%%-------------------------------------------------------------------------------------------
%% Direct Down-convertion
%%-------------------------------------------------------------------------------------------

%create time refference vector and RF carrier
SignalLength = length(Signal.Out);

%create the carrier
SignalRFcarrier = 2*cos(2*pi*intFCarrier*Signal.t);

%direct down conversion. from RF to DC
EVM.Sdc = Signal.Out .* SignalRFcarrier;

%%-------------------------------------------------------------------------------------------
%% Low Pass filter
%%-------------------------------------------------------------------------------------------

% compute the fft, and the linear frequence axis of the downconverted signal
f = linspace(-intFs/2, +intFs/2, SignalLength);
EVM.f_SDC = fftshift(fft(EVM.Sdc));

% allocate memory for the filtered signal
EVM.f_SDCF(1:SignalLength) = min(EVM.f_SDC);

% Remove frequency components outside EVM.filterWn
SamplesInBW = find(f > -EVM.filterWn/2 & f < EVM.filterWn/2);
EVM.f_SDCF(SamplesInBW) = EVM.f_SDC(SamplesInBW);

% calculate the inverse fft, and return to the time domain
% the signal is used as Input for the actual EVM algorithm
EVM.Sin = ctranspose(ifft(fftshift(EVM.f_SDCF)));

% the reffernce signal is stored.
EVM.Sref = Signal.CMSignal;

%%-------------------------------------------------------------------------------------------
%% Compensate for constant time delay
%%-------------------------------------------------------------------------------------------

% calculate the cross correlation between the input and the reference
EVM.Xcoor_Sref_Sin = xcorr(EVM.Sref,EVM.Sin);

% locate the peak in the crosse correlation, indicating the delay
EVM.Optimum_delay = SignalLength - ...
    find(abs(EVM.Xcoor_Sref_Sin) == max(abs(EVM.Xcoor_Sref_Sin))) + 1;

% change time refference to obtain max crosscorrelation
EVM.StC = EVM.Sin(EVM.Optimum_delay:end);
EVM.Sreft = EVM.Sref(1:end-EVM.Optimum_delay+1);
EVM.t = Signal.t(1:end-EVM.Optimum_delay+1);

%%-------------------------------------------------------------------------------------------
%% Compensate for constant phase error
%%-------------------------------------------------------------------------------------------

% Because the phase of the output signal is supposed to be different
% from the input signal, a phase correctrion is performed
EVM.Pcorr = angle(ctranspose(EVM.Sreft(intExcludeSamples:end))*...
                             (EVM.StC(intExcludeSamples:end)));
EVM.SptC = EVM.StC*exp(-i*EVM.Pcorr);

%%-------------------------------------------------------------------------------------------
%% Amplitude normalization
%%-------------------------------------------------------------------------------------------

% Because the amplitude of the output signal is supposed to be different
% from the input signal, an correction normalization is performed

Anormalizationfactor =  mean(abs(EVM.SptC(intExcludeSamples:end)))...
                       \mean(abs(EVM.Sreft(intExcludeSamples:end)));


% Normalize the input signal
EVM.SaptC = EVM.SptC .* Anormalizationfactor;

```



```
 95 %%-------------------------------------------------------------------------------------------
 96 %% Calculate EVM
 97 %%-------------------------------------------------------------------------------------------
 98 % see formulas in T.Larsen mm2. slide 20
 99
100 %Calculate the error vector containing both the phase- and magnitude error.
101 EVM.EV = EVM.SaptC - EVM.Sreft;
102
103 % convert the error vector to error vector magnitude.
104 EVM.EVM = abs(EVM.EV);
105
106 % Calculate the RMS EVM in part per hundred,
107 % removing the increased error in the beginning and end of the signal
108 intStart = intExcludeSamples;
109 intStop = SignalLength-intExcludeSamples;
110 EVM.EVMRMS = sqrt(...
111     sum(abs(EVM.EVM(intStart:intStop)).^2)/...
112     sum(abs(EVM.Sreft(intStart:intStop)).^2)...
113                       )*100;
114 % print the transmitter EVM
115 fprintf('    EVM = %6.2f %%\n',EVM.EVMRMS)
116
117 %%-------------------------------------------------------------------------------------------
118 %% The Remaning Code is not included, and is only used for plot generation
119 %%-------------------------------------------------------------------------------------------
```



```
function Vo = LimitedSlewRate(V1, V2, SR_HtoL, SR_LtoH, Fs)

%-------------------------------------------------------------------------------------------
% Limited Slew rate function
%-------------------------------------------------------------------------------------------
%
% the function takes the following as input
%
% V1         :   predicted output
% V2         :   previous output
% SR_HtoL    :   Slew rate of a negative incline
% SR_LtoH    :   Slew rate of a positive incline
% Fs is the  :   Samplings frequency
%
% The function then returns the actual, and slew rate limited output
% Setting one of of the slew rate values to zero cancles the function

if(SR_HtoL ~= 0 && SR_LtoH ~= 0)        % Skip if SR is zero
    if V1 > V2                          % The incline is positive
        if V1-V2 > (SR_LtoH/Fs)         % The change exceeds the SR limit
           Vo = V2 + (SR_LtoH/Fs);
        else                            % The change is below the SR limit
            Vo = V1;
        end
    elseif V1 < V2                      % the incline is negativ
        if V2-V1 > (SR_HtoL/Fs)         % The change exceeds the SR limit
           Vo = V2 - (SR_HtoL/Fs);
        else                            % The change is below the SR limit
            Vo = V1;
        end
    else
        Vo = V1;
    end
else
    Vo = V1;
end

end
```



```
function VoLimited = LimitSignalSwing(Vo, VoMAX, VoMIN)
%-------------------------------------------------------------------------------------
% Limits the signal swing to the predefined limits.
%-------------------------------------------------------------------------------------
% Input Parameters
%
% Vo           :    The output signal if not limited
% VoMAX        :    The maximum allowed output voltage
% VoMIN        :    The minimum allowed output voltage
%
% Output Signals
%
% VoLimited    :    The limited output voltage
%
%

if Vo < VoMIN               % The output is below the threshold and is limited
    VoLimited = VoMIN;
elseif Vo > VoMAX           % The output is above the threshold and is limited
    VoLimited = VoMAX;
else                        % The output is valid
    VoLimited = Vo;
end

end
```



```
function [PSD_IN PSD_INf] = ADSPSD(FileName)
%--------------------------------------------------------------------------
%  ADS_PSD help screen
%--------------------------------------------------------------------------
%
% Input
%
% FileName       : The name of the m-file contaning the data.
%                  Data type: String
%                  The data must be structured as below
%                  [time(1) data(1)
%                   time(2) data(2)
%                         ...
%                   time(n) data(n)];
%
% Example:
%
% ADS_PSD('ads',)
%--------------------------------------------------------------------------
fprintf('This is ADSPSD\n')
% Load the data from the input file
run(FileName)
Signal = ans;
time = Signal(:,1);
Signal = Signal(:,2);

intFs = 1/(time(2)-time(1)); % determine the input sample frequence

intBW = 20e6;
intFFT = 20;
intDeltaF = intBW/intFFT;

% Determine window length required [samples]
intWindowLength = ceil(intFs/intDeltaF);

% calculate the FFT, using pwelch
[P_FFT PSD_INf] = pwelch(Signal,intWindowLength,[],intWindowLength,intFs,'twosided');

% generate the linear frequency axis
PSD_INf = linspace(-intFs/2, intFs/2, length(PSD_INf));

% Normalization the spectrum to peak in the signal band
ind = find(PSD_INf>2e9,1);
ind2 = find(PSD_INf<3e9,1,'last');
P_FFT = fftshift(P_FFT);
MaxPower = max(P_FFT(ind:ind2));

% calculate the power spectrum
PSD_IN = 10*log10(P_FFT/MaxPower);
end
```


Annexes

A1 Email Correspondence

Subject: [RISC CAD] PIN LVS problem solved for UMC018
From: Jan Hvolgaard Mikkelsen <jhm@es.aau.dk>
Date: Tue, 27 Feb 2007 14:46:53 +0100
To: 07gr1050@kom.aau.dk, risc@kom.aau.dk

```
Hi

Daniel cracked the code ....

/Jan

--
Please avoid sending me unnecessary Word or PowerPoint attachments.
See http://www.gnu.org/philosophy/no-word-attachments.html
```

Subject: PIN LVS problem solved
From: "Daniel Sira" <ds@es.aau.dk>
Date: Tue, 27 Feb 2007 13:26:11 +0100
To: "Jan Hvolgaard Mikkelsen" <jhm@es.aau.dk>
CC: <sg@es.aau.dk>

Hi Jan,

after several days I succeed to solve the PIN problem in LVS. It is quite simple – you have to just put a TEXT!!! label on a wire in appropriate metal layer. So – if you would like to put on some wire pin – instead of creating pin you just have to create text label in the same layer as is the metal wire. I have checked LVS in my layout and it completely matches J

Regards,

Daniel

Aalborg University
Department of Electronic Systems
Technology Platforms Section
Niels Jernes Vej 12, A6 - 118
DK - 9220 Aalborg
Denmark

http://es.aau.dk/sections/technology_platforms_tps/
Phone: +45 96 35 86 83

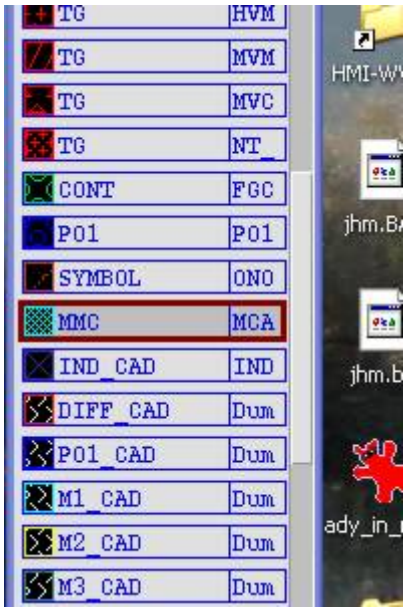
PIN LVS problem solved	**Content-Type:** message/rfc822 **Content-Encoding:** 7bit

Subject: [UMC018] Info related to capacitors (MCAP and M1 issue)
From: Jan Hvolgaard Mikkelsen <jhm@es.aau.dk>
Date: Fri, 23 Feb 2007 11:45:33 +0100
To: Daniel Sira <ds@es.aau.dk>
CC: 07gr1050@kom.aau.dk, Tian Tong <tt@kom.aau.dk>

Hi

This is the reply I got reg. the MCAP/M1 issue. I have attached a screen grap of the LSW for you to see what layer it is that he is referring to.

> You see this MCAP layer as MMC - MCAP in your LSW window (gds-nr. 112 - datatype 36) and only in the MIMCAPM_RF capacitors. This MCAP is nothing more than a pwell blockage, so if you have M1 used as shielding also, the MCAP will have no influence, except that the capacitor will not behave according to the model anymore. It also gives many DRC violations. For a dedicated maskset, you can discuss with UMC about this, but for MPW this is notp ossible. UMC just rejects any design with too many violations. We can not take this risk with 8 other circuits on the same frame, offcourse. Instead of deleting, it is maybe easier to fix the DRc violations (more spacing to unrelated difusion for example).



new-1.jpg	**Content-Type:** image/jpeg **Content-Encoding:** base64