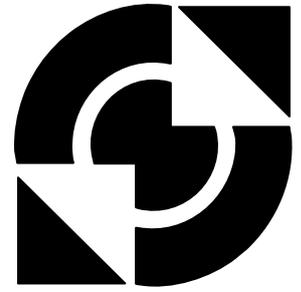


University of Twente

Faculty of Electrical Engineering,
Mathematics & Computer Science



Design of a 12bit 500Ms/s standalone charge redistribution Digital-to- Analog Converter

Frank B. Boschker
MSc. Thesis
January 2008

Supervisors:
prof. ir. A.J.M. van Tuijl
ir. F. van Houwelingen
dr. ir. R.A.R. van der Zee
Report number: 067.3245
Chair of Integrated Circuit Design
Faculty of Electrical Engineering,
Mathematics & Computer Science
University of Twente
P. O. Box 217
7500 AE Enschede
The Netherlands

1 Abstract

The subject of this Master Thesis is the design of a 12-bit 500Ms/s standalone charge redistribution Digital-to-Analog Converter. Digital-to-Analog Converters with very high analog bandwidth are mostly built with unitary and binary weighted current sources. Their performance is limited by the matching precision of these current sources. A new type of converter based on charge redistribution can easily be operated at very high sampling rates and is built with integrated capacitors. The matching accuracy of integrated capacitors is excellent.

The charge redistribution Digital-to-Analog Converter already shows fast and pretty linear settling but used as a standalone Digital-to-Analog Converter it must show a perfect 'hold' function and glitches between consecutive samples must be very small. In this report several possible architectures are presented that incorporate these features. Reducing the glitches and still keeping the total capacitance low are the important aspects that contributed to the choice of thermometer coding the four MSB bits and implementing the next eight bits in a binary weighted architecture with a split array. Calculations and simulations show that this architecture achieves the required matching accuracy and reduction of glitches with acceptable total capacitance. The static performance is measured and the Differential Nonlinearity (DNL) and Integral Nonlinearity (INL) are within $\pm 1/2$ LSB so monotonicity is guaranteed. The dynamic performance measured in the Signal-to-Noise Ratio (SNR) is close to the theoretical value of 74dB.

Overall it can be concluded that the architecture presented in this report is suitable for the specified requirements; 12 bits resolution at a clock frequency of 500Ms/s with reasonable total capacitance.

Future work should include further improvements in the linearity of the settling behavior, the cleaning up of the supply voltage and the design of an output buffer.

2 Table of Content

1 Abstract	2
2 Table of Content	3
3 Introduction.....	6
4 The Digital-to-Analog Converter (DAC).....	8
4.1 DAC specifications	8
4.1.1 Transfer curve	9
4.1.2 Least Significant Bit (LSB)	9
4.1.3 Most significant Bit (MSB)	10
4.1.4 Resolution	10
4.1.5 Differential nonlinearity.....	10
4.1.6 Integral Nonlinearity	10
4.1.7 Signal-to-Noise Ratio (SNR).....	11
4.1.8 Total harmonic Distortion (THD).....	11
4.1.9 Signal-to-Noise-and-Distortion (SINAD).....	12
4.1.10 Effective-Number-of-Bits (ENOB)	12
4.2 Types of DACs	13
4.2.1 Oversampling DAC	13
4.2.2 Binary Weighted DAC.....	13
4.2.3 R-2R Ladder DAC	13
4.2.4 Thermometer coded DAC.....	13
4.2.5 Segmented DAC	14
4.2.6 Hybrid DAC.....	14
4.3 Charge redistribution DAC.....	15
4.3.1 The Split Array	16
4.3.2 Thermometer coded DAC.....	18
4.3.3 Inverter as switch	20
4.3.4 The settling time	22
4.3.5 The deglitch capacitor.....	23
4.3.6 Standalone DAC	23
5 Design of the charge redistribution DAC	26
5.1 Inverter sizes	27
5.2 Calculating the component values	28
5.2.1 Determining the proportionality constant, A_C	28
5.2.2 Calculations for the minimal value of the unit capacitor.....	29
5.2.3 Discussion of the results	31
5.3 Final design.....	32
6 Simulation results.....	34
6.1 DNL and INL.....	34
6.2 Single tone test.....	35
6.3 Two tone test.....	41
6.4 THD	43
6.5 Area.....	44
6.6 Power consumption.....	45
6.7 MSB transition	46

7	Conclusions and recommendations.....	48
8	Appendix A.....	50
	8.1 MATLAB code for DNL and INL calculations.....	50
9	References.....	52
10	Bibliography	54

3 Introduction

Digital-to-Analog Converters with very high analog bandwidth are mostly built with a number of unitary currents for the MSB bits and with binary weighted currents for the LSB bits. The performance for low frequencies is limited by the matching precision of the current sources. Several methods are published to improve the matching behavior like calibration, Dynamic Element Matching (DEM) and sorting algorithms. Those methods may add a lot to the complexity but for low frequencies the results are satisfactory. For higher signal frequencies the matching of the timing of the current sources becomes the biggest challenge. Improving the matching behavior with methods comparable to the methods that improve the low frequency behavior is not very successful yet.

A new Digital-to-Analog Converter

In recent work on Successive Approximation Analog-to-Digital Converters a new Digital-to-Analog Converter was introduced. This new converter is based on charge redistribution and it can easily be operated at very high sampling rates. Furthermore the matching accuracy of integrated capacitors is excellent. Converters up to 12-bit or even more resolution are possible without any adjustments. DA Converters that are used in AD Converters are already adequate if they only show the right output value when the comparator action takes place. However used as a standalone DA Converter they must show a perfect 'hold'-function and glitches between consecutive samples must be very small. This report covers the design of such a converter.

Chapter four describes the Digital-to-Analog Converter in general; its specifications, the types of DA Converters and finally the charge redistribution DA Converter are introduced.

In the next chapter, chapter five, the design is discussed. Several important parts of the converter are explained and calculations are done for the components involved.

The results are presented in chapter six. The static and dynamic performance is measured and compared to the theory handled in the previous chapter.

Chapter seven presents the conclusions of this report and recommendations for future work.

4 The Digital-to-Analog Converter (DAC)

Probably the most popular digital-to-analog converter application is the digital audio compact disc player. Here digital information stored on the CD is converted into music via a high-precision DAC. Many characteristics define a DAC's performance. Each characteristic will be discussed [1].

4.1 DAC specifications

A block diagram of a DAC can be seen in Fig. 4.1. Here an N -bit digital word is mapped into a single analog voltage. Typically, the output of the DAC is a voltage that is some fraction of a reference voltage (or current), such that

$$V_{out} = FV_{ref} \quad (1)$$

$$F = \frac{D}{2^N} \quad (2)$$

where V_{out} is the analog voltage output, V_{ref} is the reference voltage, and F is the fraction defined by the input word, D , that is N bits wide. The number of input combinations represented by the input word D is related to the number of bits in the word by

$$\text{Number of input combinations} = 2^N$$

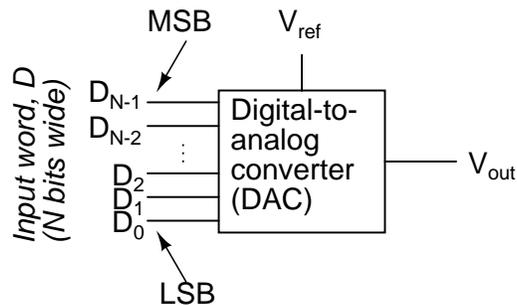


Figure 4.1: Block diagram of the digital-to-analog converter

4.1.1 Transfer curve

By plotting the input word, D , versus V_{out} as D is incremented from 000 to 111 for 3 bits, the *transfer curve* seen in Fig. 4.2 would be generated. The y-axis has been normalized to V_{ref} . Some important characteristics need to be discussed here. First, notice that the transfer curve is not continuous. Since the input is a digital signal, which is inherently discrete, the input signal can only have eight values that must correspondingly produce eight output voltages. If a straight line connected each of the output values, the slope of the line would ideally be one increment/input code value. Also note that the maximum value of the output is $7/8$. Since the case where $D = 000$ has to result in an analog voltage of $0V$, and a 3-bit DAC has eight possible analog output voltages, then the analog output will increase from $0V$ to only $7/8V_{ref}$.

This maximum analog output voltage that can be generated is known as *full-scale voltage*, V_{FS} , and can be generalized to any N -bit DAC as

$$V_{FS} = \frac{2^N - 1}{2^N} \cdot V_{ref} \quad (3)$$

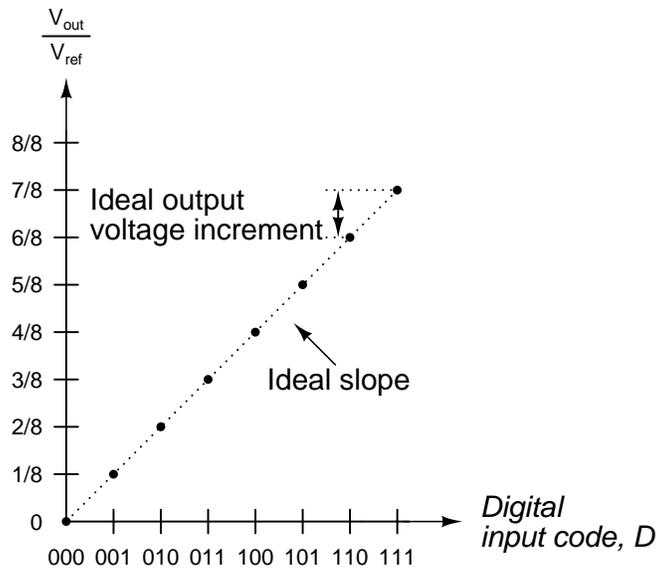


Figure 4.2: Ideal transfer curve for a 3-bit DAC

4.1.2 Least Significant Bit (LSB)

The *least significant bit (LSB)* refers to the rightmost bit in the digital input word. The LSB defines the smallest possible change in the analog output voltage. The LSB will always be denoted as D_0 . One LSB can be defined as

$$1LSB = \frac{V_{ref}}{2^N} \quad (4)$$

4.1.3 Most significant Bit (MSB)

The *most significant bit (MSB)* refers to the leftmost bit of the digital word, D . Generalizing to the N -bit DAC, the MSB would be denoted as D_{N-1} . The MSB causes the output to change by $\frac{1}{2}V_{ref}$.

4.1.4 Resolution

The term resolution describes the smallest change in the analog output with respect to the value of the reference voltage V_{ref} . Resolution is typically given in terms of bits and represents the *number of unique output voltage levels*, i.e., 2^N .

4.1.5 Differential nonlinearity

Nonideal components cause the analog increments to differ from their ideal values. The difference between the ideal and nonideal values is known as *differential nonlinearity*, or DNL and is defined as

$$DNL = \max \left| \frac{V(i+1) - V(i)}{V_{LSB}} - 1 \right|, \quad \forall i = 0..(2^N - 2) \quad (5)$$

The DNL specification measures how well a DAC can generate uniform analog LSB multiples at its output.

Generally, a DAC will have less than $\pm\frac{1}{2}$ LSB of DNL if it is to be N -bit accurate. If the DNL for a DAC is more than $\pm\frac{1}{2}$ LSB, then the DAC is said to be *nonmonotonic*, which means that the analog output voltage does not always increase as the digital input code is incremented. A DAC should always exhibit *monotonicity* if it is to function without error.

4.1.6 Integral Nonlinearity

Another important static characteristic of DACs is called *integral nonlinearity (INL)*. Defined as the difference between the data converter output values and a reference straight line drawn through the first and last output values, INL defines the linearity of the overall transfer curve and can be described as

$$INL = \max \left| \frac{V(i) - i \cdot V_{LSB}}{V_{LSB}} \right|, \quad \forall i = 0..(2^N - 1) \quad (6)$$

It is common practice to assume that a converter with N -bit resolution will have less than $\pm\frac{1}{2}$ LSB of DNL and INL.

Figure 4.3 shows both the DNL and INL graphically.

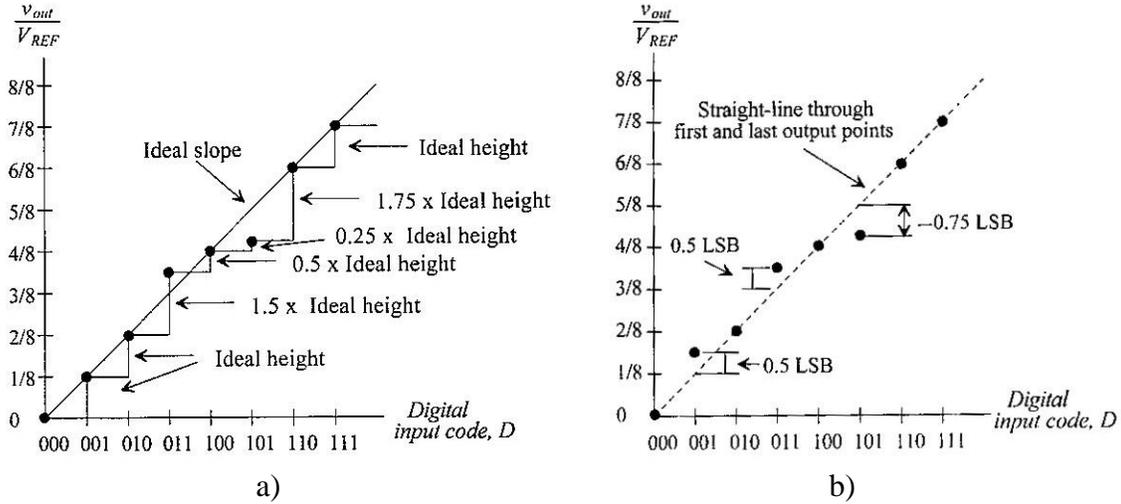


Figure 4.3: a) DNL and b) INL explained

4.1.7 Signal-to-Noise Ratio (SNR)

Signal-to-Noise Ratio (SNR) is defined as the ratio of the signal power to the noise at the analog output. In amplifier applications, this specification is typically measured using a sine wave input. For the DAC, a “digital” sine wave is generated through instrumentation or through an A/D. The SNR can reveal the true resolution of a data converter as the effective number of bits can be quantified mathematically.

$$\text{Error energy: } e_{rms}^2 = \frac{V_{LSB}^2}{12}$$

$$\text{Signal energy: } s_{rms}^2 = \frac{1}{T} \int_{t=0}^{t=T} A^2 \sin^2(\omega t) dt = \frac{A^2}{2} = \frac{2^{2N} V_{LSB}^2}{8}$$

$$\text{SNR} = \frac{\frac{2^{2N} V_{LSB}^2}{8}}{\frac{V_{LSB}^2}{12}} = \frac{3}{2} 2^{2N} \approx 6.02 \cdot N + 1.76 [dB]$$

4.1.8 Total harmonic Distortion (THD)

The total harmonic distortion is given by

$$\text{THD} = 10 \log \frac{A_{2f}^2 + A_{3f}^2 + A_{4f}^2 + \dots + A_{nf}^2}{A_{1f}^2} [dB] \quad (7)$$

Five to ten harmonics are included in the THD, the rest is considered “noise”

4.1.9 Signal-to-Noise-and-Distortion (SINAD)

Signal-to-Noise-and-Distortion (SINAD) is the ratio of the root-mean-square (rms) signal amplitude to the mean value of the root-sum-square (rss) of all other spectral components, *including harmonics*, but excluding dc. SINAD is a good indication of the overall dynamic performance of a DAC because it includes all components which make up noise and distortion.

$$SINAD = \frac{S}{N + D} \quad (8)$$

4.1.10 Effective-Number-of-Bits (ENOB)

SINAD is often converted to *effective-number-of-bits (ENOB)* using the relationship for the theoretical SNR of an ideal N -bit DAC: $SNR = 6.02N + 1.76[dB]$. The equation is solved for N , and the value of SINAD is substituted for SNR:

$$ENOB = \frac{SINAD - 1.76dB}{6.02} \quad (9)$$

4.2 Types of DACs

The most common types of electronic DACs are listed in the following paragraphs.

4.2.1 Oversampling DAC

Oversampling DACs such as the Sigma-Delta DAC use a pulse density conversion technique. The oversampling technique allows for the use of a lower resolution DAC internally. A simple 1-bit DAC is often chosen because the oversampled result is inherently linear. The DAC is driven with a pulse density modulated signal, created with the use of a low-pass filter, step nonlinearity (the actual 1-bit DAC), and negative feedback loop, in a technique called sigma-delta modulation. This results in an effective high-pass filter acting on the quantization (signal processing) noise, thus steering this noise out of the low frequencies of interest into the high frequencies of little interest, which is called noise shaping. The quantization noise at these high frequencies is removed or greatly attenuated by use of an analog low-pass filter at the output (sometimes a simple RC low-pass circuit is sufficient). Most very high resolution DACs (greater than 16 bits) are of this type due to its high linearity and low cost. Higher oversampling rates can either relax the specifications of the output low-pass filter or enable further suppression of quantization noise. Speeds of greater than 100 thousand samples per second (for example, 192kHz) and resolutions of 24 bits are attainable with Delta-Sigma DACs.

4.2.2 Binary Weighted DAC

The Binary Weighted DAC contains one resistor or current source for each bit of the DAC connected to a summing point. These precise voltages or currents sum to the correct output value. This is one of the fastest conversion methods but suffers from poor accuracy because of the high precision required for each individual voltage or current. Such high-precision resistors and current sources are expensive, so this type of converter is usually limited to 8-bit resolution or less.

4.2.3 R-2R Ladder DAC

This type of DAC is a binary weighted DAC that uses a repeating cascaded structure of resistor values R and $2R$. This improves the precision due to the relative ease of producing equal valued matched resistors (or current sources). However, wide converters perform slowly due to increasingly large RC -constants for each added R - $2R$ link.

4.2.4 Thermometer coded DAC

The thermometer coded DAC contains an equal resistor or current source segment for each possible value of DAC output. An 8-bit thermometer DAC would have 255 segments, and a 16-bit thermometer DAC would have 65,535 segments. This is perhaps the fastest and highest precision DAC architecture but at the expense of high cost. Conversion speeds of over 1 billion samples per second have been reached with this type of DAC.

4.2.5 Segmented DAC

The segmented DAC combines the thermometer coded principle for the most significant bits and the binary weighted principle for the least significant bits. In this way, a compromise is obtained between precision (by the use of the thermometer coded principle) and number of resistors or current sources (by the use of the binary weighted principle). The full binary weighted design means 0% segmentation, the full thermometer coded design means 100% segmentation.

4.2.6 Hybrid DAC

The hybrid DAC uses a combination of the above techniques in a single converter. Most DAC integrated circuits are of this type due to the difficulty of getting low cost, high speed and high precision in one device.

4.3 Charge redistribution DAC

Shown in Fig. 4.4, a charge redistribution DAC is a parallel array of binary-weighted capacitors, $2^N C$ in total. After initially being discharged, the digital signal switches each capacitor to either V_{ref} or ground, causing the output voltage, V_{out} , to be a function of the voltage division between the capacitors.

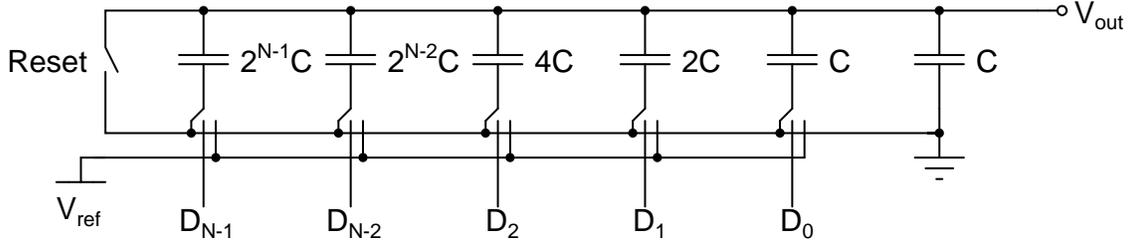


Figure 4.4: A charge-redistribution DAC

The capacitor array totals $2^N C$. Therefore, if the MSB is high and the remaining bits are low, then a voltage divider occurs between the MSB capacitor and the rest of the array. The analog output voltage, V_{out} , becomes

$$V_{out} = V_{ref} \cdot \frac{2^{N-1} C}{(2^{N-1} + 2^{N-2} + 2^{N-3} + \dots + 4 + 2 + 1 + 1)C} = V_{ref} \cdot \frac{2^{N-1} C}{2^N C} = \frac{V_{ref}}{2} \quad (10)$$

which confirms the fact that the MSB changes the output of a DAC by $\frac{1}{2}V_{ref}$. Fig. 4.5 shows the equivalent circuit under this condition.

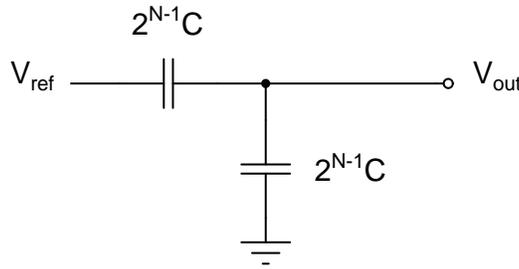


Figure 4.5: Equivalent circuit with the MSB = 1, and all other bits set to zero

The ratio between V_{out} and V_{ref} due to each capacitor can be generalized to

$$V_{out} = \frac{2^k C}{2^N C} \cdot V_{ref} = 2^{k-N} \cdot V_{ref} \quad (11)$$

where it is assumed that the k -th bit, D_k , is one and all other bits are zero. Superposition can then be used to find the value of V_{out} for any input word by

$$V_{out} = \sum_{k=0}^{N-1} D_k 2^{k-N} \cdot V_{ref} \quad (12)$$

4.3.1 The Split Array

The charge-redistribution architecture is very popular because of its simplicity and relative good accuracy. Although a linear capacitor is required, high resolution in the 10- to 12-bit range can be achieved. However, as the resolution increases, the size of the MSB capacitor becomes a major concern. For example if the unit capacitor, C , were $100fF$, and a 12-bit DAC were to be designed, the MSB capacitor would need to be

$$C_{MSB} = 2^{N-1} \cdot 100fF = 204.8pF \quad (13)$$

One method of reducing the size of the capacitors is to use a split array. A 6-bit example of the array is pictured in Fig. 4.6. This architecture is slightly different from the charge-redistribution DAC pictured in Fig. 4.4 in that the output is taken off a different node and an additional attenuation capacitor is used to separate the array into a LSB array and a MSB array. Note that the LSB, D_0 , now corresponds to the leftmost switch and that the MSB, D_5 , corresponds to the rightmost switch.

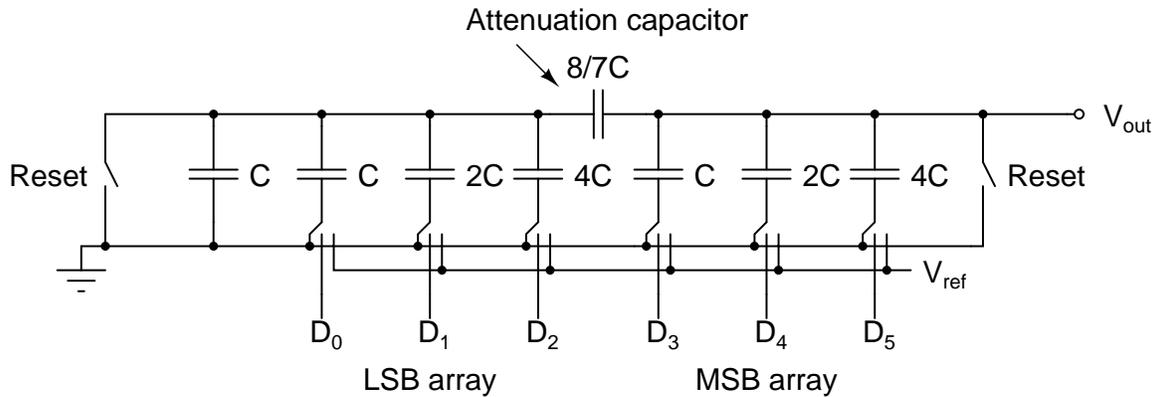


Figure 4.6: A charge-redistribution DAC using a split array

The value of the attenuation capacitor can be found by

$$C_{att} = \frac{\text{sum of the LSB array capacitors}}{\text{sum of the MSB array capacitors}} \cdot C \quad (14)$$

where the sum of the MSB array equals the sum of LSB capacitor array minus C . The value of the attenuation capacitor should be such that the series combination of the attenuation capacitor and the LSB array, assuming all bits are zero, equals C . To prove this a derivation is made, see formula (15). The output voltage is defined as the

attenuation factor times the LSB bits plus the MSB bits times the reference voltage. The attenuation factor is a capacitive divider between the attenuation capacitor and the sum of the LSB array capacitors. One can see that with some manipulation this is equal to formula (12)

$$\begin{aligned}
V_{out} &= \left(\frac{2^{N/2}}{2^{N/2} - 1} \sum_{k=0}^{N/2-1} D_k 2^{k-N/2} + \sum_{k=N/2}^{N-1} D_k 2^{k-N} \right) \cdot V_{ref} \\
&= \left(\frac{1}{2^{N/2}} \sum_{k=0}^{N/2-1} D_k 2^{k-N/2} + \sum_{k=N/2}^{N-1} D_k 2^{k-N} \right) \cdot V_{ref} \\
&= \left(\frac{2^{N/2}}{2^{N/2}} \cdot \sum_{k=0}^{N/2-1} D_k 2^{k-N} + \sum_{k=N/2}^{N-1} D_k 2^{k-N} \right) \cdot V_{ref} \\
&= \left(\sum_{k=0}^{N/2-1} D_k 2^{k-N} + \sum_{k=N/2}^{N-1} D_k 2^{k-N} \right) \cdot V_{ref} = \sum_{k=0}^{N-1} D_k 2^{k-N} \cdot V_{ref}
\end{aligned} \tag{15}$$

A drawback of the split array is that spreading in the attenuation capacitor affects all the capacitors after the attenuation capacitor. Therefore, care in the layout should be taken.

4.3.2 Thermometer coded DAC

The advantage of a binary-weighted DAC is its simplicity, as no decoding logic is required [2]. There are several major drawbacks, however, which are all associated with major bit transitions. At the mid-code transition (011 111 111 111 \rightarrow 100 000 000 000), the most significant bit (MSB) capacitor needs to be matched to the sum of all the other capacitors to within $\pm\frac{1}{2}\text{LSB}$. This is difficult to achieve. Because of statistical spread, such matching can never be guaranteed. Therefore this architecture is not guaranteed monotonic. Matching is an issue for all bit transitions, but the severity of the problem is proportional to the weight of the bit, resulting in a typical differential nonlinearity (DNL) plot as shown in Fig. 4.7a. In addition, the errors caused by the dynamic behavior of the switches (such as charge injection and clock feed through) result in glitches in the output signal as shown in Fig. 4.7b. This problem is most severe at the mid-code transition, as all switches are switching simultaneously. Such a mid-code glitch contains highly nonlinear signal components, even for small output signals and will manifest itself as spurs in the frequency domain.

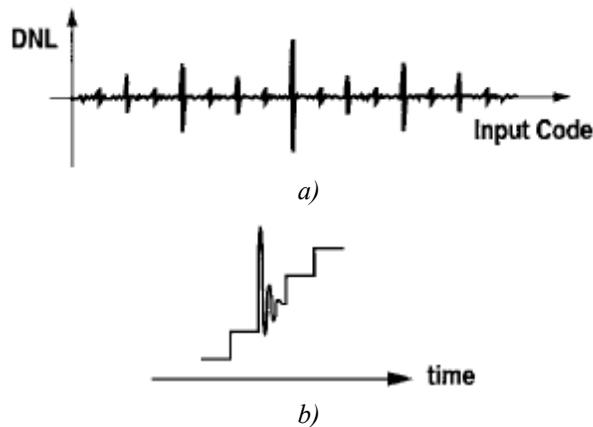


Figure 4.7: Matching and glitch problems of a binary-weighted DAC

Fig. 4.8 shows an example of a 3-bit thermometer-coded DAC. There are $2^3 = 8$ unit capacitors. Each unit capacitor is connected to a switch controlled by the signal coming from the binary-to-thermometer decoder.

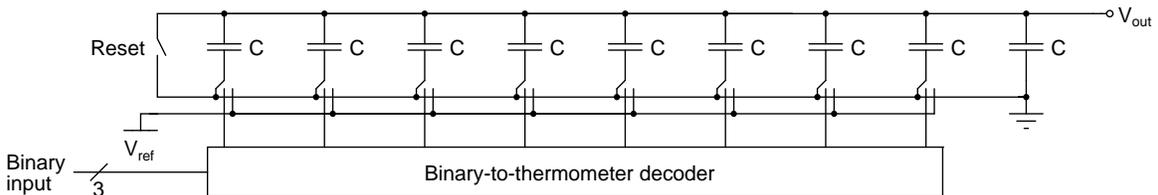


Figure 4.8: 3-bit thermometer coded DAC

When the digital input increases by 1LSB, one more capacitor is charged. The analog output is always increasing as the digital input increases. Hence, monotonicity is

guaranteed using this architecture. In addition, there are several other advantages for a thermometer-coded DAC compared to its binary-weighted counterpart. First, the matching requirement is much relaxed: 50% matching of the unit capacitor is good enough for a DNL of 0.5 LSB, as shown in Fig. 4.9a. At the mid-code, a 1LSB transition (011 111 111 111 \rightarrow 100 000 000 000), causes only one capacitor to charge as the digital input only increases by one. This greatly reduces the glitch problem. On top of that, glitches hardly contribute to nonlinearity in the thermometer coded architectures. This is because the magnitude of a glitch is proportional to the number of switches that are actually switching. So for small steps, the glitch is small, and for a large step, the glitch is large. Since the number of switches that switch is proportional to the signal step between two consecutive clock cycles, the magnitude of the glitch is directly proportional to the amplitude of the signal step. As an example, Fig. 4.9b shows that the glitch in the output signal of a step of 4LSB has exactly the same shape and duration as the glitch in the output signal for a 1LSB step and it is exactly 4 times larger in amplitude. If the glitch is strictly proportional to the signal step, it will not cause any nonlinearity in the DAC output signal.

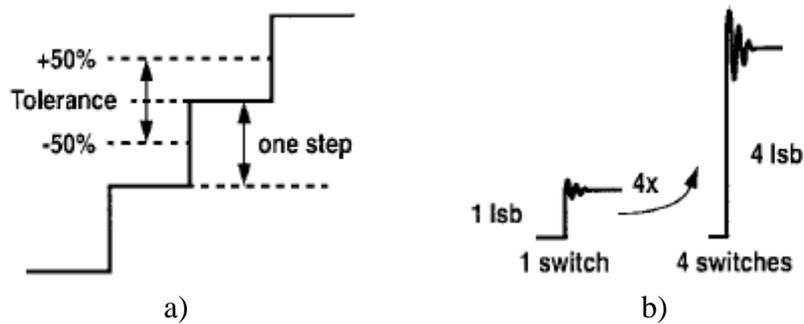


Figure 4.9: Matching and glitch advantages of a thermometer-coded DAC

Table 4.1 shows the conversion table from decimal to binary to thermometer code.

Decimal	Binary			Thermometer Code						
	b ₁	b ₂	b ₃	d ₁	d ₂	d ₃	d ₄	d ₅	d ₆	d ₇
0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	1
2	0	1	0	0	0	0	0	0	1	1
3	0	1	1	0	0	0	0	1	1	1
4	1	0	0	0	0	0	1	1	1	1
5	1	0	1	0	0	1	1	1	1	1
6	1	1	0	0	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1

Table 4.1: Conversion table

One major drawback of the thermometer coded DAC is area consumption, since for every LSB this architecture needs a capacitor, a switch, and a decoding circuit, as well as the binary to thermometer decoder.

4.3.3 Inverter as switch

The digital signal switches each capacitor to either V_{ref} or ground. The implementation of this switch can be realized by an inverter as it too can switch to either V_{ref} or ground. The input of the inverter is the digital input code D_N coming from the digital part of the converter. The output of the inverter is connected to the charging capacitor C_N , shown in Fig. 4.10.

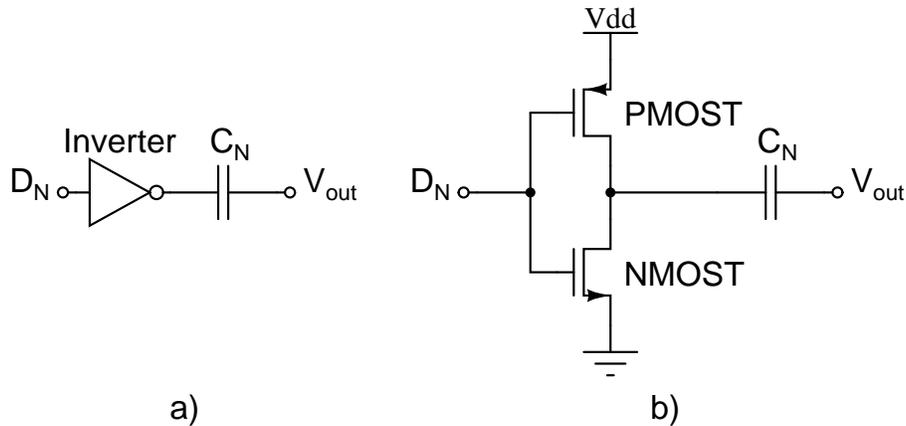


Figure 4.10: The inverter as switch, a) the symbol, b) transistor level

The inverter has an internal ON-resistance, which is the resistance that the current experiences flowing through the transistor when switched “ON”. When the digital input code is low, i.e. “0”, the top transistor, a PMOS transistor, is switched “ON” and the bottom transistor, a NMOS transistor, is switches off, see Fig. 4.11.

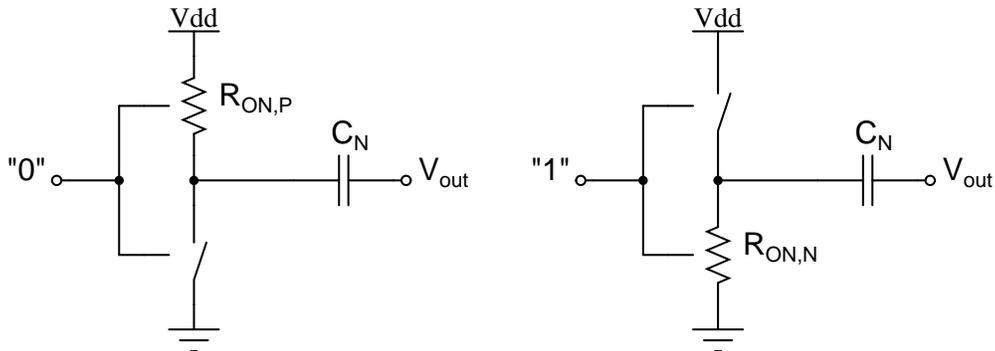


Figure 4.11: ON-resistance in an inverter

In the first situation the capacitor is charged to V_{dd} . In the other situation when D_N is high, i.e. “1”, the top transistor is “OFF” and the bottom transistor conducts, discharging the capacitor. The combination of the ON-resistance of the inverter and the charging capacitor delivers a RC-product (also called τ) which defines the time needed to charge a capacitor to 63% of full charge. This property defines the bandwidth and the settling and is discussed in the next paragraph.

One major drawback of the inverter is that the ON-resistance is not linear. As can be seen from Fig. 4.12, a plot made by ProMOST, the ON-resistance of each transistor depends on the gate voltage. The equivalent ON-resistance of the whole inverter is shown as the bold line. When the inverter switches, the ON-resistance changes during a transition. This affects the linearity of the settling which will be discussed in paragraph 4.3.6.

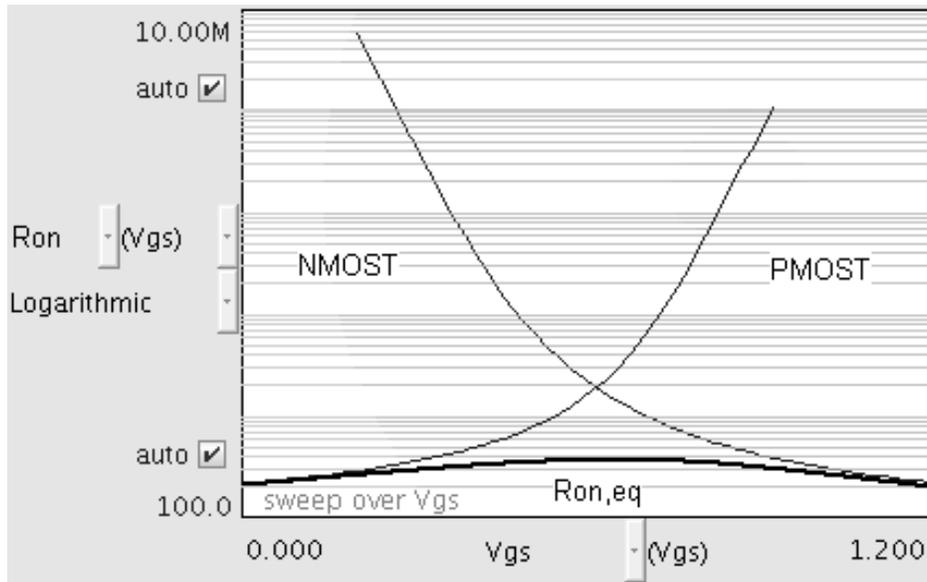


Figure 4.12: ON-resistance of a NMOS- and PMOS transistor and the equivalent ON-resistance of the inverter

4.3.4 The settling time

The charging of every individual capacitor that represents a bit value should occur within a certain time, the settling time. The settling time depends on the ON-resistance of the inverter, the capacitive load of the inverter (deglitch- and charging capacitor) and the accuracy demand. The accuracy is determined by the required resolution, which is 12 bits. The charging of a capacitor is defined by formula (16).

$$V_C = V_{ref} \left(1 - e^{-\frac{t}{RC}} \right) \quad (16)$$

Because of the 12-bit resolution the smallest step that can be made at the output is

$$V_{LSB} = \frac{V_{ref}}{2^N} \quad (17)$$

To settle within the required accuracy the minimal time needed will be

$$V_{ref} \left(1 - e^{-\frac{t}{RC}} \right) = V_{ref} - V_{LSB} = V_{ref} - \frac{V_{ref}}{2^N} = V_{ref} \left(1 - \frac{1}{2^N} \right) \quad (18)$$

$$e^{-\frac{t}{RC}} = \frac{1}{2^N} \quad (19)$$

$$t = RC \ln(2^N) \quad (20)$$

For a 12-bit DAC the settling time is $8.3 \cdot RC$.

The product of the ON-resistance and the capacitive load of the inverter determines the bandwidth. So if the sampling frequency is 500Ms/s then the time to charge and discharge the charging capacitor is equal to 2ns . Formula (16) describes only the charging of a capacitor, therefore the settling time has to occur in only half the period, in 1ns .

Substituting the settling time in formula (20) results in an expression for the RC-product.

$$RC = \frac{1\text{ns}}{\ln(2^N)} = 120\text{ps} \quad (21)$$

4.3.5 The deglitch capacitor

As discussed in paragraph 4.3.2 thermometer coded architectures have glitch advantages. But for the binary weighted part of the DAC glitches are still a problem. Therefore a deglitch capacitor is placed between the output of the inverter and the charging capacitor and ground, plotted in Fig. 4.13. The combination of the ON-resistance of the inverter and the deglitch capacitor is essentially a low-pass filter which filters out the high frequency glitches.

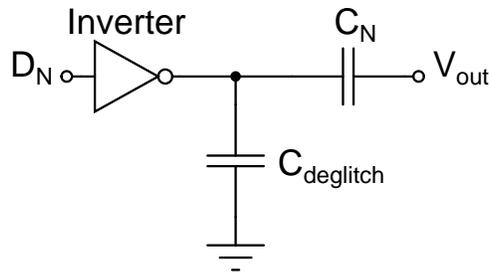


Figure 4.13: The deglitch capacitor

A second advantage is that the effect of the nonlinear ON-resistance of the inverter on the RC-time is weakened. This improves the linearity of the settling.

4.3.6 Standalone DAC

A standalone DAC should show a perfect hold function and glitches between consecutive samples must be very small. The perfect hold function produces a staircase signal at the output as can be seen in Fig. 4.14. This results in a frequency spectrum of the signal with its aliases attenuated by a sinc-function ($= \sin(x)/x$). The sinc-function is the Fourier transform of a rectangular pulse. The energy of the output is equal to the area under the signal. With a perfect hold function the area is proportional to the amplitude. The advantage of the hold function is that it does not add noise to the spectrum. The hard part is to implement such a 'hold' function. A solution is to use a first order hold function. The output is also shown in Fig. 4.14. The energy of the first order hold function is also equal to the area but suffers from a gain error. Furthermore, a first order hold function also does not add noise.

To implement this in the DAC every branch should have equal first order hold functions or RC-times. In paragraph 4.3.3 we saw that the ON-resistance was not linear. Paragraph 4.3.5 provided a partial solution by adding a deglitch capacitor, but still some nonlinear settling occurs. This will affect the dynamic performance of the DAC as will be seen in Chapter 6 in the simulations.

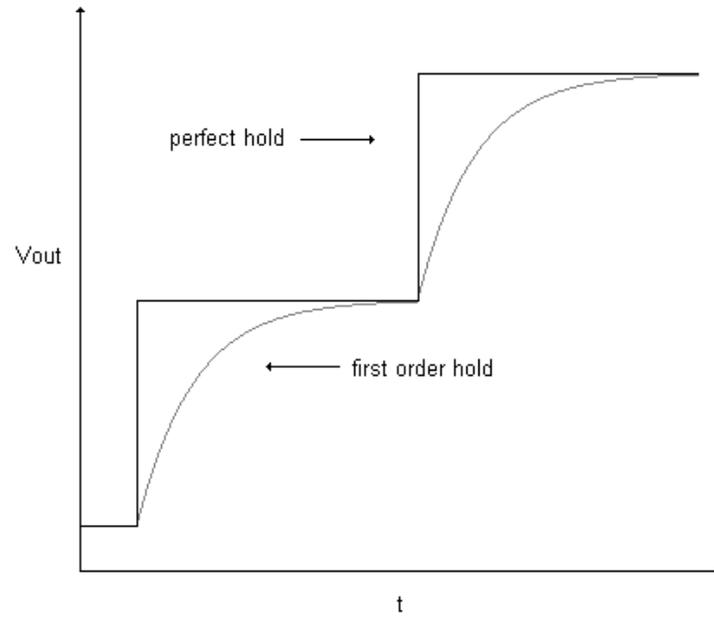


Figure 4.14: Linear settling

5 Design of the charge redistribution DAC

In this chapter the design of the charge redistribution DAC is discussed. There are several choices in architecture to be made before getting to the final design. The design starts off with a few specifications. The resolution of DAC should be 12 bits. The total chip area should be kept small, so a reasonable value for the total capacitance would be around $25pF$. The DAC is standalone and the sampling frequency is $500Ms/s$. With only the first two requirements it can be seen that a fully binary weighted architecture is not feasible. In paragraph 4.3.1 formula (13) stated the value of the MSB capacitor for a 12-bit DAC in case of a unit capacitor of $100fF$. The total capacitance is equal to the sum of all the bit capacitors and is given in formula (22).

$$C_{total} = \sum_{N=1}^{12} 2^{N-1} \cdot C_{unit} = (2^{12} - 1) \cdot C_{unit} = 4095 \cdot C_{unit} \quad (22)$$

If the total capacitance has to be around $25pF$ then the unit capacitor has to be around

$$C_{unit} = \frac{25pF}{4095} = 6.1fF \quad (23)$$

The spreading of these sized capacitors is relatively large, so sizing them up would be a solution, but that would immediately affect the total capacitance. Also the inverters that charge these capacitors should be inversely binary weighted in size. This produces difficulties as small transistors have relatively large parasitic capacitances. Furthermore, as was discussed in paragraph 4.3.2, monotonicity cannot be guaranteed for these high resolution binary weighted DACs. The solution is to design the first MSB bits in a thermometer coded architecture and the lower bits in a binary weighted architecture. In Table 5.1 an overview is given of possible combinations in architecture.

Thermometer coded bits	Binary weighted bits	Number of split arrays
2	10	1
	5, 5	2
3	9	1
	3, 3, 3	3
4	8	1
	4, 4	2
6	6	1
	3, 3	2

Table 5.1: Possible combinations in architecture

Because the DAC is standalone, which means it must show a perfect hold function and glitches between consecutive samples must be very small, the number of thermometer coded bits must be large, preferably more than 4. But more thermometer coding means more area consumption, so 6 bits would be too many. Also, a split array in the binary

weighted part is necessary to further reduce the total capacitance. In this design an architecture was chosen of a 4-bit thermometer coded part followed by a split array, which consist of a 4-bit binary weighted part, a split array and another 4-bit binary weighted part, as is highlighted in Table 5.1.

5.1 Inverter sizes

In paragraph 4.3.4 an expression was derived that shows the relation between the capacitor value and the required inverter ON-resistance. For the binary weighted part of the DAC binary weighted capacitor values will result in inversely binary weighted ON-resistor values for the inverters. This means that every inverter has to be differently sized. A drawback of this consequence is that different sized inverters have different ON-resistance and therefore different RC-times. The settling of each branch is then different, which will not make the DAC settle linearly. To alleviate this difficulty the deglitch capacitor could be sized in such a way that for every branch the RC product would be the same. Therefore, the inverters would all have the same size. A deglitch capacitor of the same size as the charging capacitor in the MSB branch of the binary weighted part of the DAC will be sufficiently large to reduce the glitches significantly. The deglitch capacitors for the next bits will always be bigger as can be seen in formula (24).

$$C_{deglitch} = 2C_{MSB} - C_N \quad (24)$$

The RC product for every branch will then become equal to

$$RC = R \cdot (C_{deglitch} + C_N) = R \cdot (2C_{MSB} - C_N + C_N) = 2RC_{MSB} \quad (25)$$

which results in equally sized inverters for every branch. The advantage is that equally sized inverters have better matching, which results in greater linearity in the settling.

5.2 Calculating the component values

A charge redistribution DAC can only achieve its desired resolution if the spreading of the different components and the resulting error in output voltage does not exceed the error in output voltage due to quantization. Bigger capacitors will spread less, but will contribute to a larger total capacitance. The aim is to derive an expression that states the relation between the output voltage error, the spreading of the capacitors and the total capacitance, so that a minimal total capacitance can be achieved without losing resolution. In the next two paragraphs an attempt will be made.

5.2.1 Determining the proportionality constant, A_C

The mismatch of a capacitor is proportional with the inverse square root of the capacitance. The constant A_C is the proportionality constant. This parameter is process dependent and its relation to the mismatch σ and the capacitance C is shown in formula (26).

$$A_C = \frac{\sigma_C}{\sqrt{C}} [F / \sqrt{F}] \quad (26)$$

To determine the proportionality constant a simulation is done in which different sized capacitors are charged and discharged by a sinusoidal voltage. The capacitors are of the 65nm process technology. A Monte Carlo simulation with 100 runs is performed and the node current of the voltage source is plotted. Next the third most extreme currents are measured to get 98% or 2.35σ of the capacitor deviation. These current measurements are used to calculate the capacitor values with the use of formula (27).

$$C = \frac{I_{pp}}{2\pi \cdot f_{in} \cdot V_{pp}} [F] \quad (27)$$

Table 5.2 shows the simulation results and the calculated mismatch per square root capacitance.

C [fF]	I [nA]	$C_{measured}$ [fF]	2.35σ [aF]	σ [aF]	σ_{av} [aF]	$A_{C,av}$ [F/ \sqrt{F}]
10	37.94	10.06	63.53	27.03	26.51	$2.651 \cdot 10^{-10}$
	37.47	9.934	61.09	26.00		
100	377.7	100.2	198.6	84.53	83.87	$2.652 \cdot 10^{-10}$
	376.3	99.80	195.5	83.20		
1000	3772	1001	628.3	267.4	265.3	$2.653 \cdot 10^{-10}$
	3768	999	618.4	263.1		

Table 5.2: Measurements for calculating A_C

The average proportionality constant is $2.652 \cdot 10^{-10} F / \sqrt{F}$ and will be used in the remainder of this report. For a $1fF$ capacitor the relative mismatch is

$$\frac{\sigma_C}{C} = \frac{A_C}{\sqrt{C}} \cdot 100[\%] = \frac{2.652 \cdot 10^{-10}}{\sqrt{1 \cdot 10^{-15}}} \cdot 100[\%] = 0.839\%$$

5.2.2 Calculations for the minimal value of the unit capacitor

The proposed design for the charge-redistribution DAC is built up by unit capacitors. A mismatch in the unit capacitor causes a deviation in the voltage level at the output. Smaller capacitors have greater deviations in their capacitor value. To determine the lower limit on the unit capacitor, a relation between the mismatch of the unit capacitor and the deviation in output voltage should be derived.

The master thesis report of Michiel van Elzakker [3] states a formula which shows the relation between the total capacitance C_{total} , the matching parameter M , the proportionality constant A_C , and the number of bits N , see formula (28).

$$C_{total} \geq \frac{2}{3} \cdot M^2 \cdot A_C^2 \cdot 2^{2N} \quad (28)$$

This formula was derived by calculating the error voltage as a result of quantization, σ_v , and the relative voltage error as a result of the capacitor mismatch σ_{Vrel} . The relative voltage error should be smaller than the error voltage. This results in a minimal value for the total capacitance.

The total capacitance can be expressed in terms of the unit capacitor. First the total capacitance of the LSB-branch is calculated.

$$C_{branch1} = C_{dummy} + C_0 + C_1 + C_2 + C_3 = C_{unit} + C_{unit} + 2C_{unit} + 4C_{unit} + 8C_{unit} = 16C_{unit} \quad (29)$$

Next, the equivalent capacitance of the attenuation capacitor in series with the total capacitance of the LSB-branch is calculated.

$$C_{eq1} = \frac{\frac{16}{15} C_{unit} \cdot 16C_{unit}}{\frac{16}{15} C_{unit} + 16C_{unit}} = \frac{\frac{256}{15} C_{unit}^2}{\frac{16}{15} C_{unit} + \frac{240}{15} C_{unit}} = \frac{\frac{256}{15} C_{unit}^2}{\frac{256}{15} C_{unit}} = C_{unit} \quad (30)$$

The same two steps for the next branch are applied.

$$C_{branch2} = C_{eq1} + C_4 + C_5 + C_6 + C_7 = C_{unit} + C_{unit} + 2C_{unit} + 4C_{unit} + 8C_{unit} = 16C_{unit} \quad (31)$$

$$C_{eq2} = \frac{\frac{16}{15}C_{unit} \cdot 16C_{unit}}{\frac{16}{15}C_{unit} + 16C_{unit}} = \frac{\frac{256}{15}C_{unit}^2}{\frac{16}{15}C_{unit} + \frac{240}{15}C_{unit}} = \frac{\frac{256}{15}C_{unit}^2}{\frac{256}{15}C_{unit}} = C_{unit} \quad (32)$$

The total capacitance of the whole DAC is found by adding up the second equivalent capacitance and the capacitors in the MSB-branch.

$$\begin{aligned} C_{total} &= C_{eq2} + C_8 + C_9 + C_{10} + C_{11} + C_{12} + C_{13} + C_{14} + C_{15} + C_{16} + C_{17} + C_{18} + C_{19} + C_{20} + C_{21} + C_{22} \\ &= C_{unit} + C_{unit} \\ &\quad + C_{unit} + C_{unit} + C_{unit} \\ &= 16C_{unit} \end{aligned} \quad (33)$$

Formula (28) can only be used for each segment individually, because it was written for a fully binary DAC. To arrive at a formula for a segmented DAC an attenuation factor a is introduced.

$$a = \frac{C_{att1}}{C_{att1} + C_{branch1}} = \frac{C_{att2}}{C_{att2} + C_{branch2}} = \frac{\frac{16}{15}C_{unit}}{\frac{16}{15}C_{unit} + 16C_{unit}} = \frac{1}{16} \quad (34)$$

$$C_{total} = C_{branch3} + \frac{C_{att2}}{C_{att2} + C_{branch2}} \left(C_{branch2} + \frac{C_{att1}}{C_{att1} + C_{branch1}} C_{branch1} \right) \quad (35)$$

$$C_{total} = C_{branch3} + a \cdot (C_{branch2} + a \cdot C_{branch1}) = 16 \cdot (1 + a + a^2) \cdot C_{unit} \quad (36)$$

Now that the total capacitance is written in terms of the unit capacitor the formula becomes

$$C_{total} = 16 \cdot (1 + a + a^2) \cdot C_{unit} \geq \frac{2}{3} \cdot M^2 \cdot A_C^2 \cdot 2^{2N} \Rightarrow C_{unit} \geq \frac{2}{3} \cdot \frac{M^2 \cdot A_C^2 \cdot 2^{2N}}{16 \cdot (1 + a + a^2)} \quad (37)$$

Calculating an actual minimum value requires numerical values for M , A_C , N and a . The value in (38) is based on 3σ matching, an A_C of $2.652 \cdot 10^{-10} F / \sqrt{F}$, a 12 bits converter and an attenuation factor of $\frac{1}{16}$.

$$C_{unit} \geq \frac{2}{3} \cdot \frac{3^2 \cdot (2.652 \cdot 10^{-10})^2 \cdot 2^{2 \cdot 12}}{16 \cdot \left(1 + \frac{1}{16} + \left(\frac{1}{16} \right)^2 \right)} \approx 415 \text{ fF} \quad (38)$$

5.2.3 Discussion of the results

The attenuation factor a is based on the attenuation capacitor and the branch below it. Mismatch in the attenuation capacitor is not taken into account in formula (38). An attempt to derive a formula that includes the mismatch in the attenuation capacitor was made, but it involved the calculation of quotients of component variances, which proved to be too complex. Instead, a code in MATLAB is written that models the DAC. For different values of the unit capacitor the DNL and INL can be computed. The value for the unit capacitor that keeps the DNL and INL within $\pm 1/2$ LSB will be chosen.

As can be seen from formula (38) the minimal value for the unit capacitor is quite large and therefore the total capacitance will be large, in the order of $200pF$, which is unacceptable. A solution to this problem was found in changing the proportions of the three branches. At the moment the attenuation factor is $1/16$ and the capacitor values of the next branch are the same as the previous branch. Simulations have shown that the MSB branch contributes the most to the error in the output voltage due to spreading and the other two branches significantly less. Therefore the capacitor values in the two branches that provide the eight lower bits are further scaled down to make room for the large capacitor values needed in the MSB branch. The unit capacitor value for the MSB branch, the first four bits, is set at $1024fF$. The unit capacitor value for the next four bits is set at $128fF$ and the unit capacitor value for the last four bits is set at $16fF$. The total capacitance is now dropped significantly to $25pF$.

In appendix A the MATLAB code can be found that provided the necessary insight. In the next chapter the DNL and INL are simulated using this MATLAB code.

5.3 Final design

In paragraph 4.3 the charge redistribution DAC was discussed and possible architectures were discussed. The choice was made for a segmented DAC with two split arrays. The first four bits were implemented in a thermometer coded way, followed by a split array, and the next eight bits in binary weighted form. The binary weighted part was also implemented as a split array. In this chapter calculations were performed to derive the component values. In Table 5.3 an overview of all the component values is given and Fig. 5.1 shows the complete DAC. The inverter sizes are determined by the use of ProMOST.

Inverter	PMOST		NMOST		Resistance [Ω]
	width [μm]	length [μm]	width [μm]	length [μm]	
Inv ₈ to Inv ₂₂	5.725	0.060	2.090	0.060	117
Inv ₄ to Inv ₇	11.37	0.060	4.130	0.060	59
Inv ₀ to Inv ₃	1.390	0.060	0.550	0.060	469

a)

Charging capacitor	Capacitance [fF]	Deglitch capacitor	Capacitance [fF]
C ₈ to C ₂₂	1024		
C _{att2}	2048		
C ₇	1024	C _{d7}	1024
C ₆	512	C _{d6}	1536
C ₅	256	C _{d5}	1792
C ₄	128	C _{d4}	1920
C _{att1}	256		
C ₃	128	C _{d3}	128
C ₂	64	C _{d2}	192
C ₁	32	C _{d1}	224
C ₀	16	C _{d0}	240

b)

Table 5.3: Component values

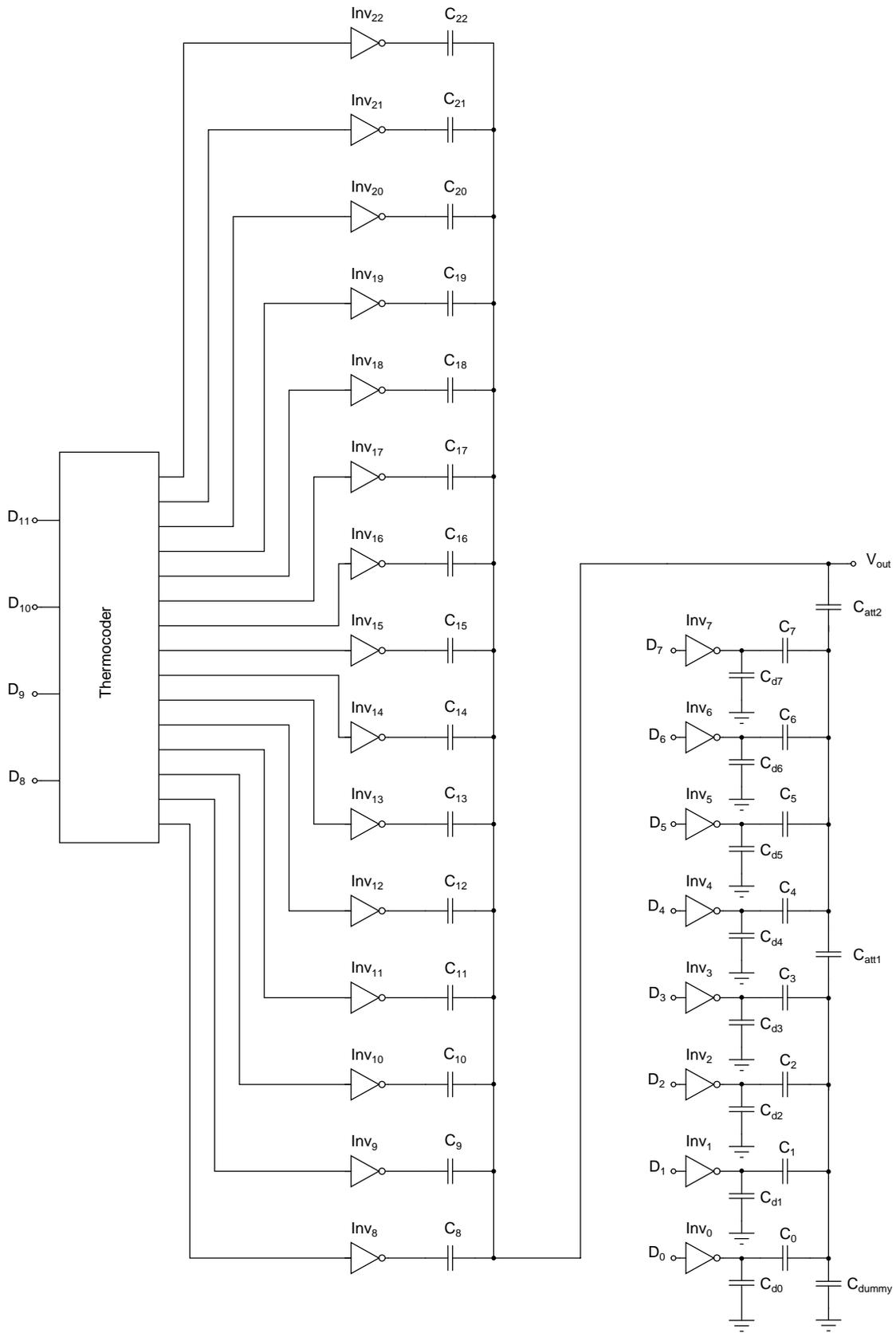


Figure 5.1: The complete DAC

6 Simulation results

In this chapter the final design is modeled in CADENCE and MATLAB and simulated. The simulation results are documented in the next paragraphs.

6.1 DNL and INL

The DNL and INL of the DAC are simulated by the MATLAB model of the DAC. The goal is to keep the DNL and INL smaller than $\pm 1/2$ LSB. Figure 6.1 shows the resulting plot. As can be seen in this plot the maximum DNL and INL occur at the MSB transition.

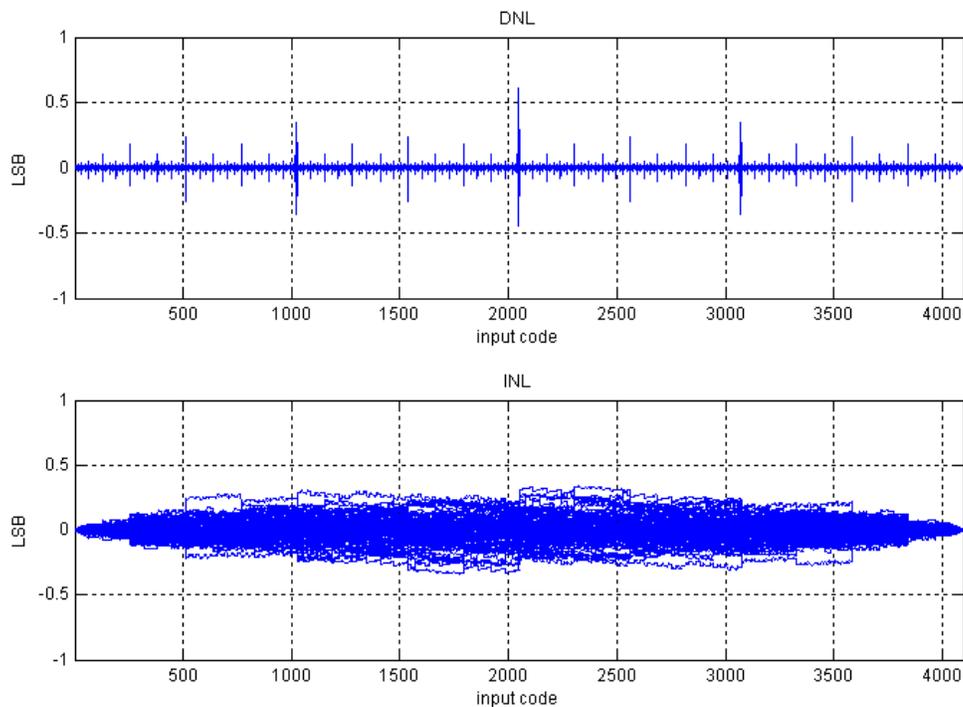


Figure 6.1: DNL and INL simulation results

The DNL is read from the plot and is equal to 0.61 LSB, a little bit more than the required maximum of $\pm 1/2$ LSB. The INL is kept within $\pm 1/2$ LSB and is equal to 0.34 LSB.

6.2 Single tone test

The Signal-to-Noise Ratio (SNR) of the DAC is found by applying a single tone to the input and sampling the output. The theoretical value for the SNR is calculated by formula (39), N is the number of bits.

$$SNR = 6.02 \cdot N + 1.76[dB] \quad (39)$$

Theoretical a 12bits converter can achieve $6.02 \cdot 12 + 1.76 = 74dB$ of SNR. This is limited by the quantization noise and depends on the resolution of the DAC. Figure 6.2 shows the magnitude plot of the DAC with input signal a sine wave with frequency $7.5MHz$.

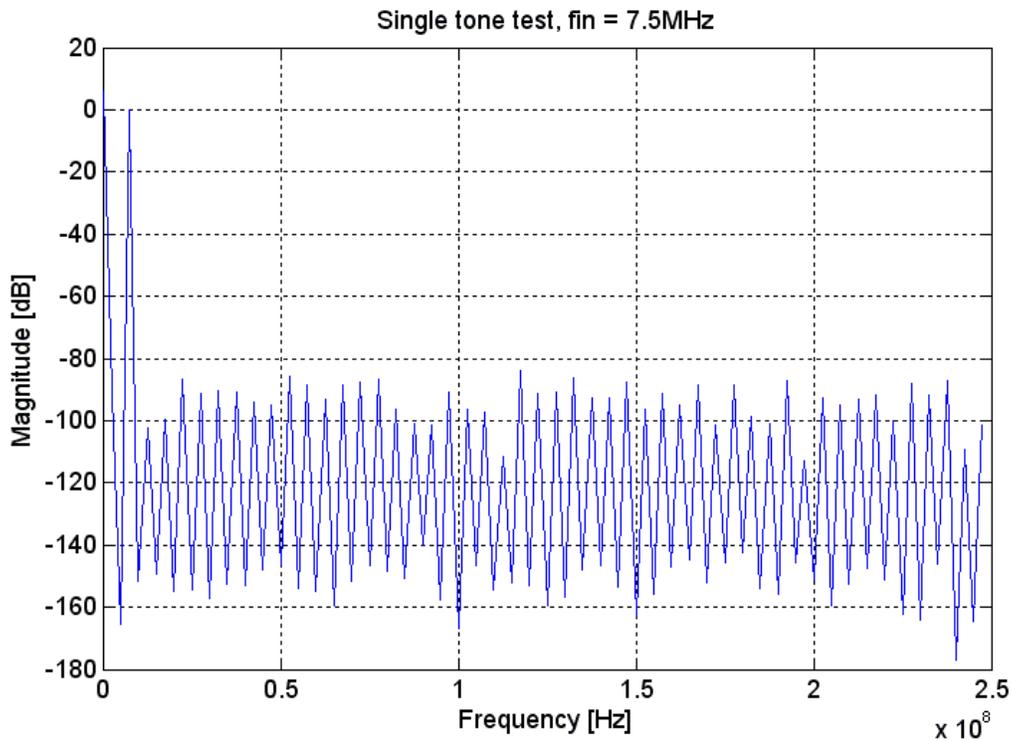
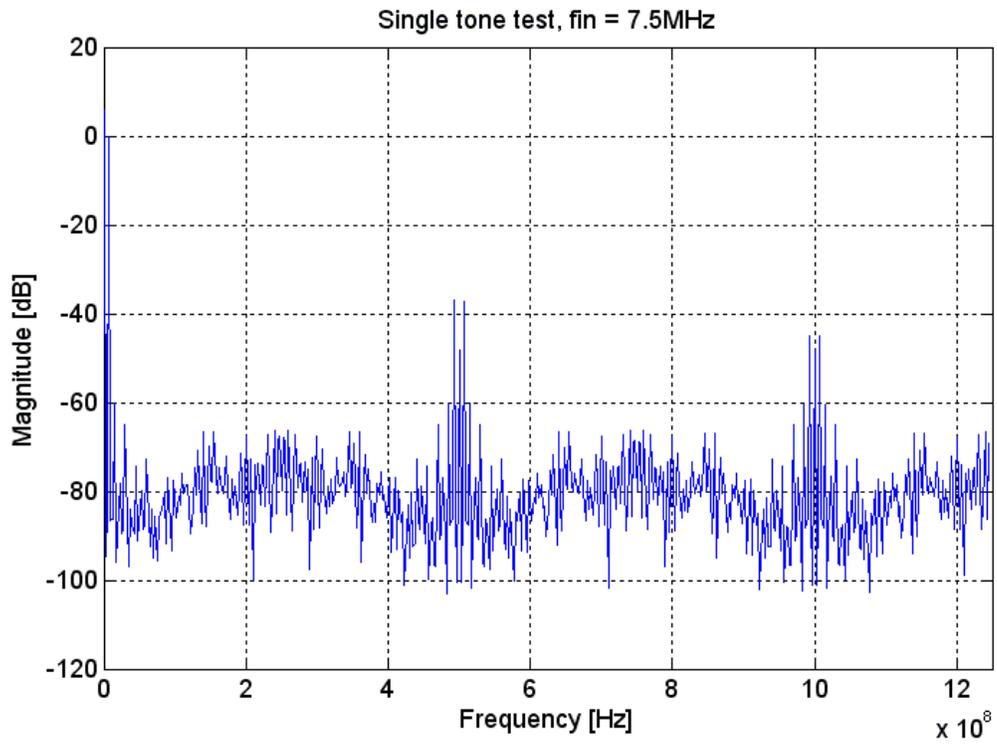
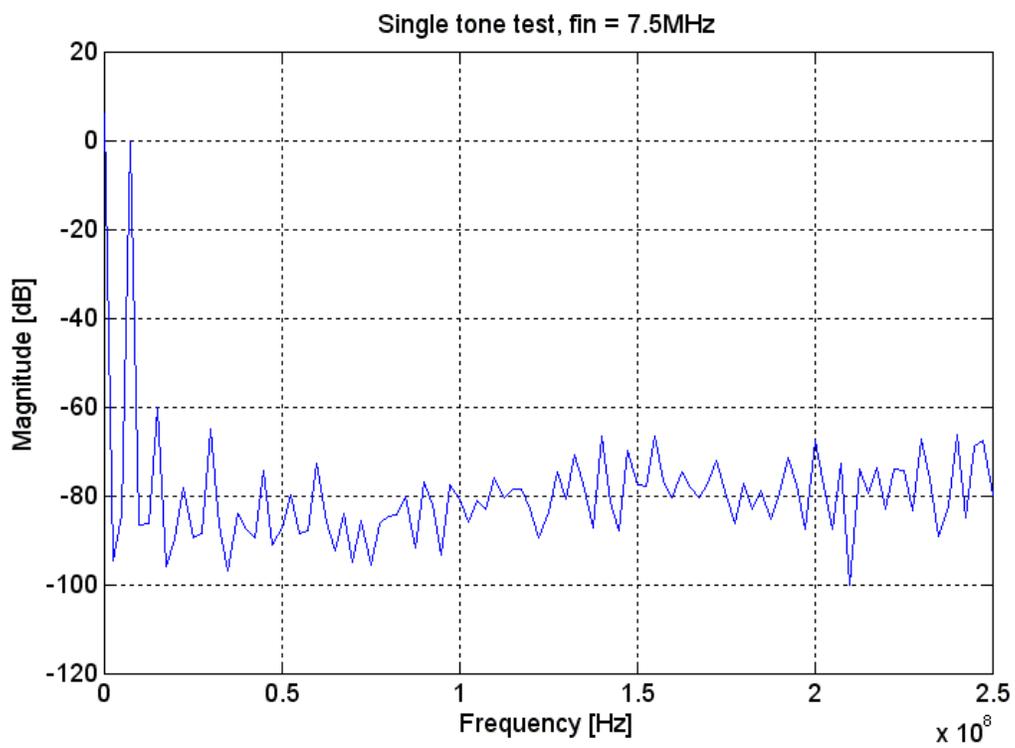


Figure 6.2: Magnitude plot of the single tone test at 500ms/s

The resulting SNR is $73.83dB$, close to the theoretical value. To investigate the settling behavior a higher sampling frequency is applied to the output. Figure 6.3 shows the output sampled at a five times higher sampling frequency at $2.5Gs/s$. The resulting SNR is equal to $54.16dB$, significantly lower than without higher sampling. The reason for this is nonlinear settling and glitching in the DAC. Linear settling only contributes to a gain error, which will be unnoticeable in the band of interest. Nonlinear settling adds unwanted frequency components in the band of interest, which show up as noise and deteriorate the SNR.



a)



b)

Figure 6.3: a) Magnitude plot of the single tone test at 2.5Gs/s, b) zoomed to 250MHz

Next as input signal a sine wave with frequency 102.5MHz is applied. The SNR is equal to 73.91dB , again close to the theoretical value. The magnitude plot is shown in Fig. 6.4.

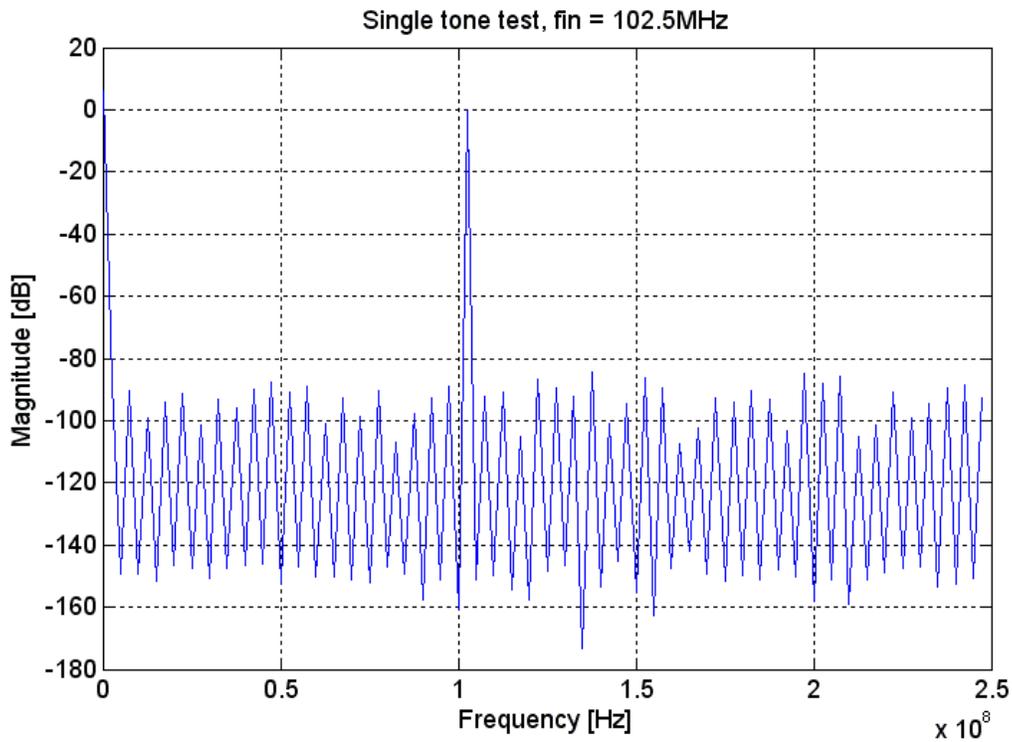
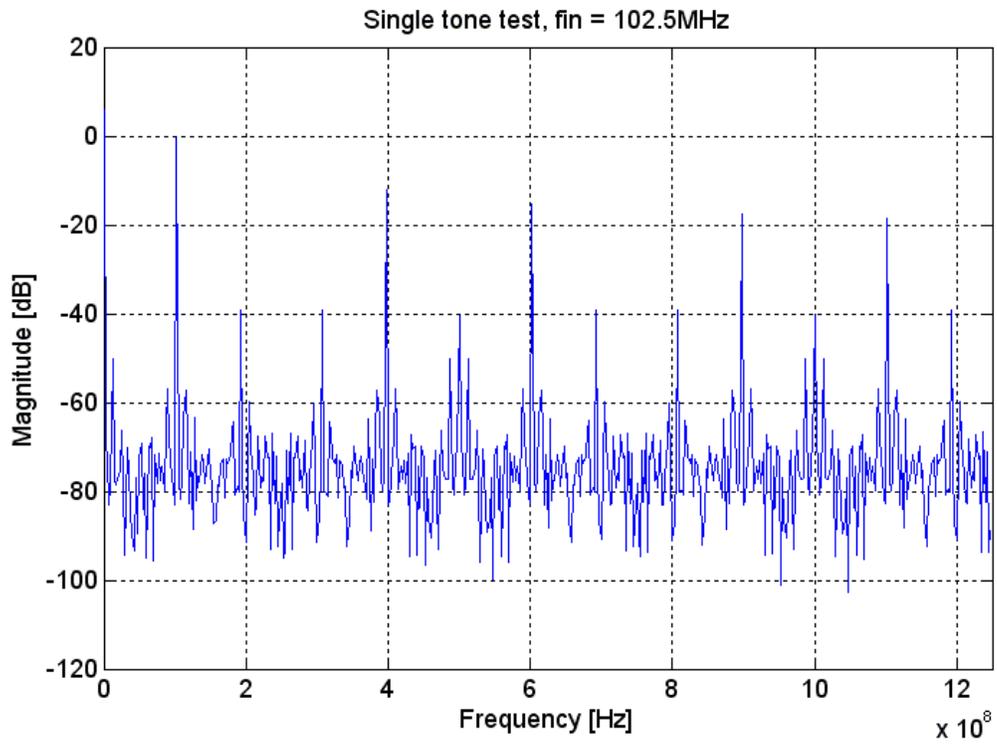
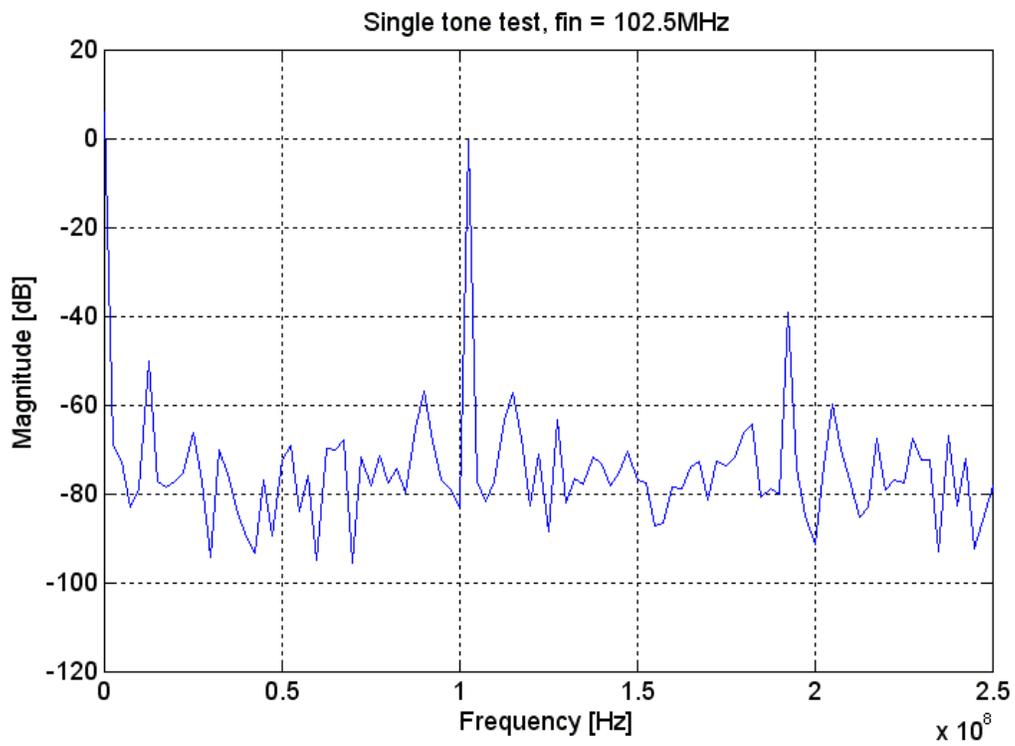


Figure 6.4: Magnitude plot of the single tone test at 500ms/s

For this input frequency also more samples were taken to reveal the settling behavior. And again, as can be seen in Fig. 6.5, the SNR deteriorates. Now the SNR is down to 37.88dB . The harmonics of the input signal can clearly be seen and are the main source of noise.



a)



b)

Figure 6.5: a) Magnitude plot of the single tone test at 2.5Gs/s, b) zoomed to 250MHz

Finally as input signal a sine wave with frequency 242.5MHz is applied. The SNR is measured and equal to 73.78dB . The magnitude plot is shown in Fig. 6.6.

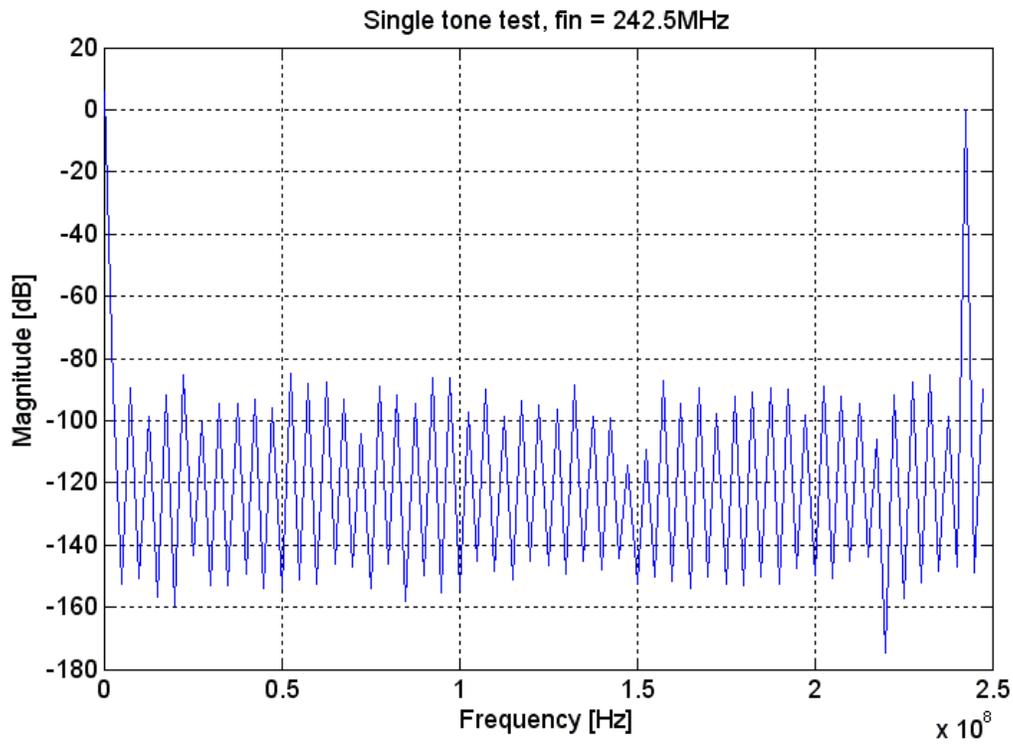
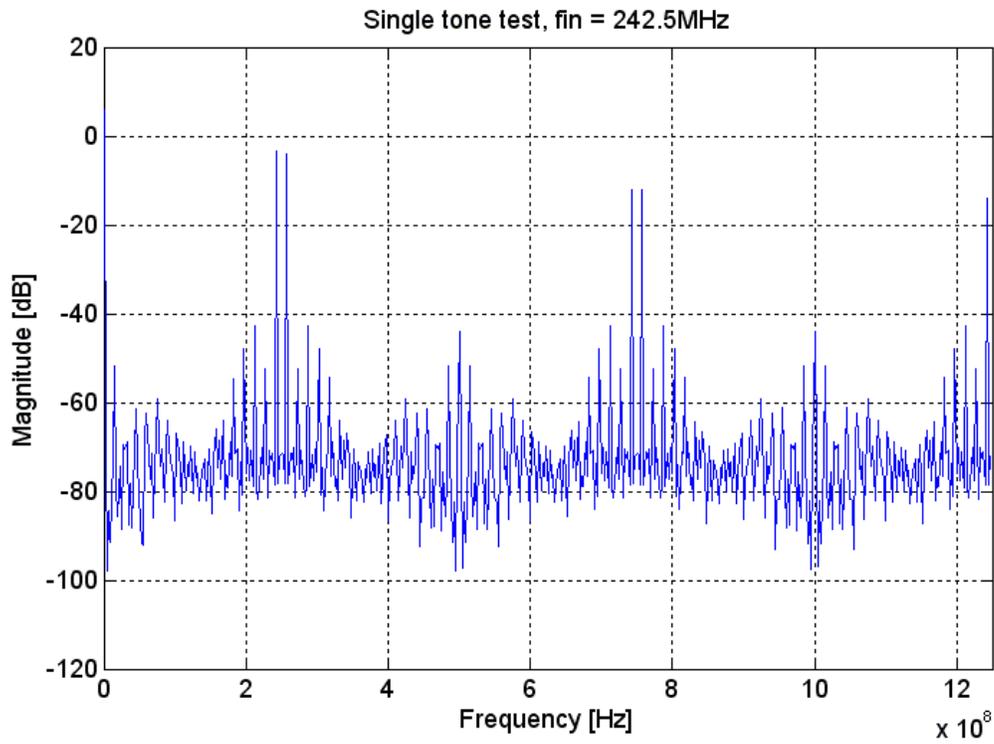
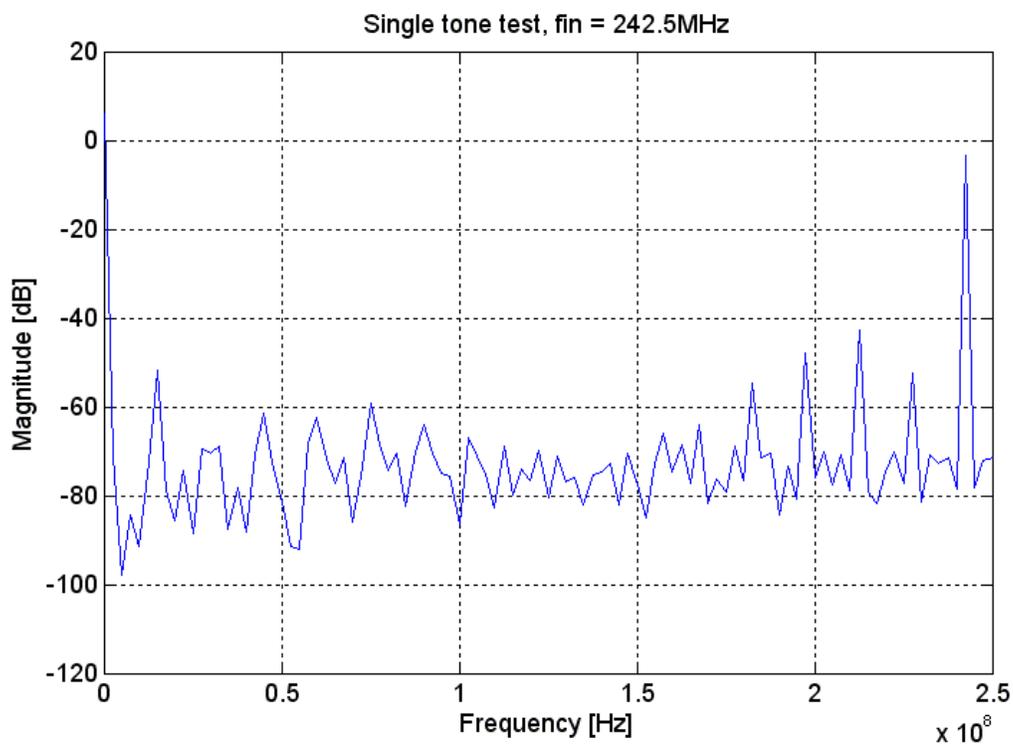


Figure 6.6: Magnitude plot of the single tone test at 500ms/s

With a higher sampling frequency the SNR reduces to 36.80dB . Even clearer than in the previous single tone test the harmonics can be seen. Fig. 6.7 shows the magnitude plot.



a)



b)

Figure 6.7: a) Magnitude plot of the single tone test at 2.5Gs/s, b) zoomed to 250MHz

6.3 Two tone test

The two tone test is performed by applying two signals to the input, $f_{in1} = 242.5\text{MHz}$, $f_{in2} = 247.5\text{MHz}$, and sampling the output. Figure 6.8 shows the output sampled at 500Ms/s . The Signal-to-Noise Ratio (SNR) is 70.54dB , close to the theoretical value of 74dB minus 3dB , because of the input signals that both have half the maximal amplitude to avoid clipping. This proves that for a two tone test the DAC settles perfectly to the wanted end value.

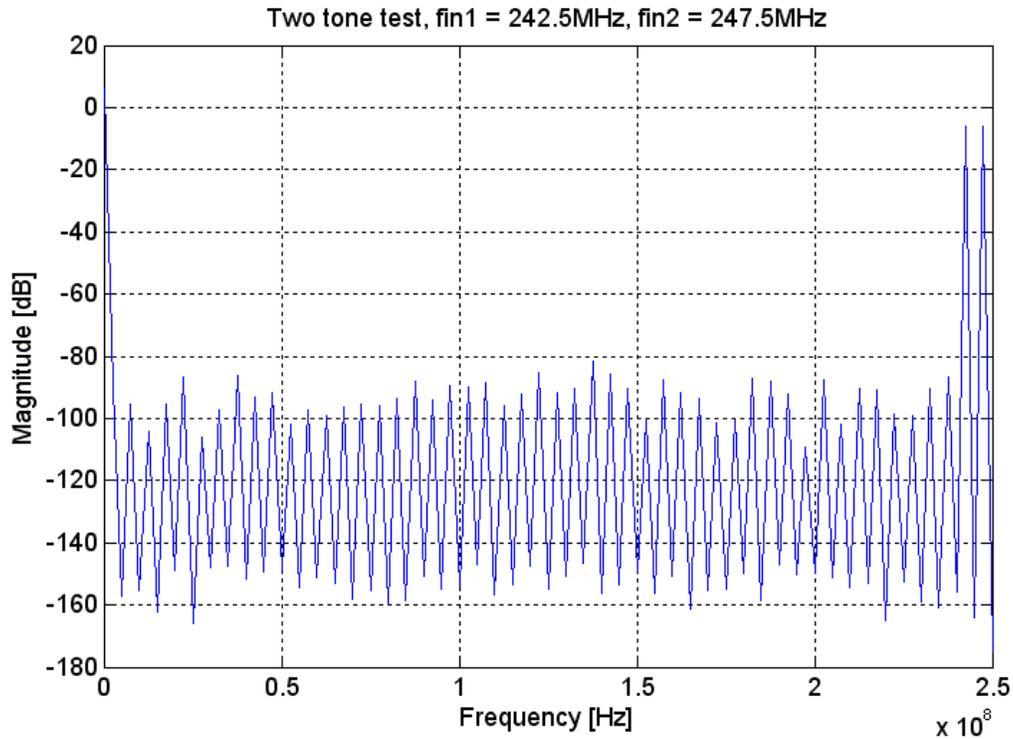
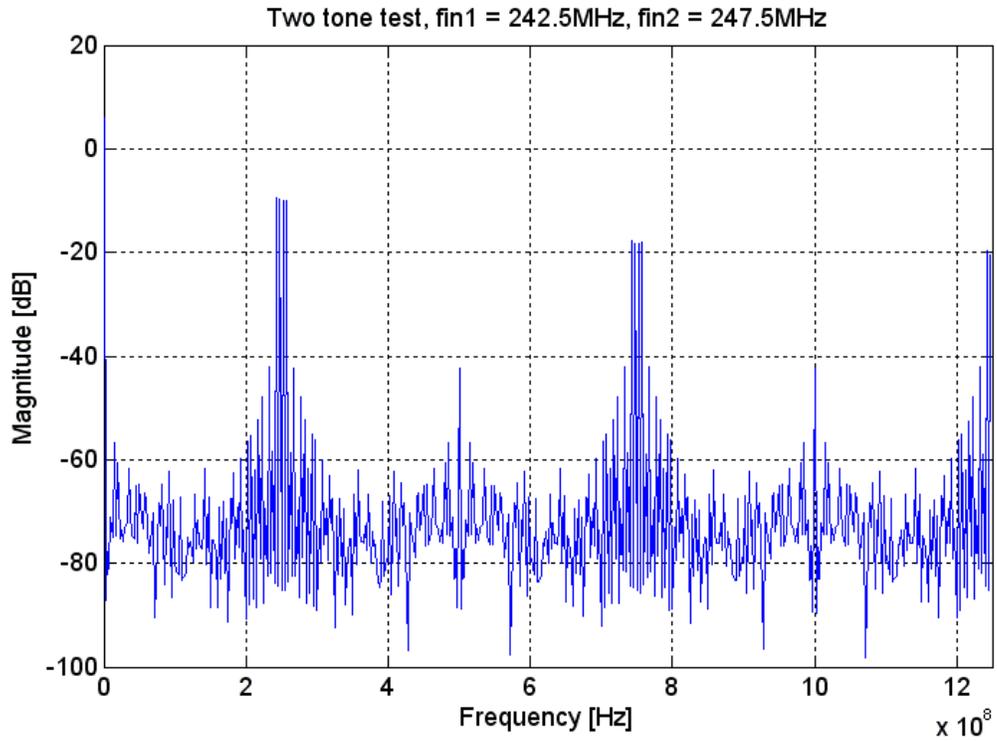
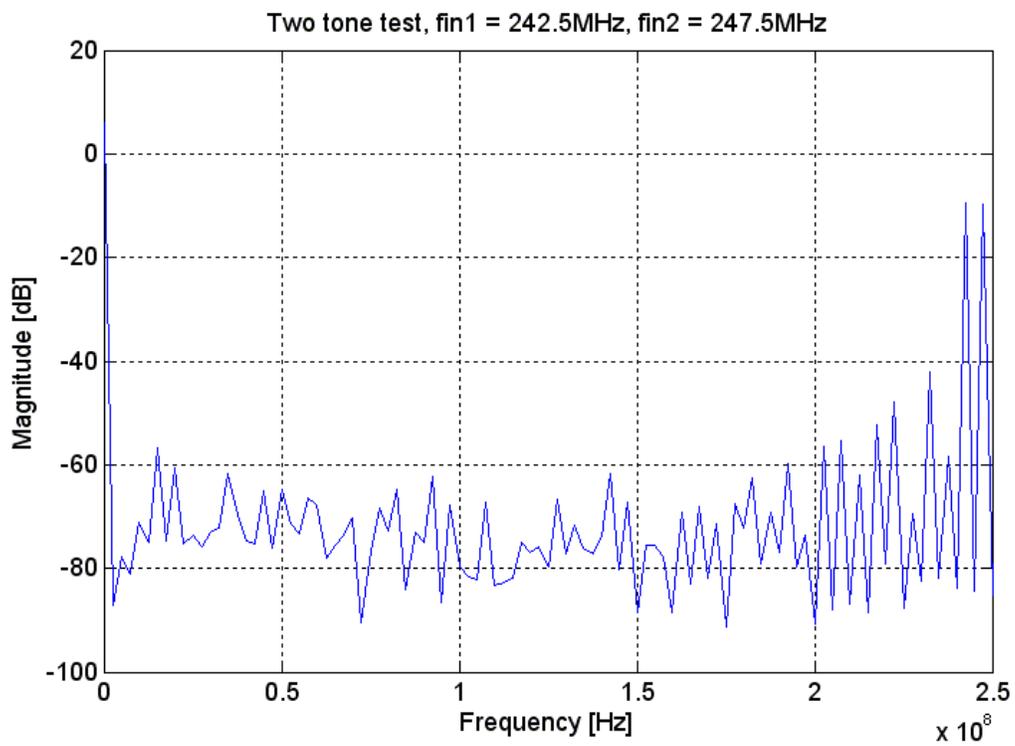


Figure 6.8: Magnitude plot of the two tone test at $f_s = 500\text{Ms/s}$

To investigate the settling behavior a higher sampling frequency is applied to the output. Figure 6.9 shows the output sampled at a five times higher sampling frequency, 2.5Gs/s . The resulting SNR is equal to 33.36dB , significantly smaller than without higher sampling.



a)



b)

Figure 6.9: a) Magnitude plot of the two tone test at 2.5Gs/s, b) zoomed to 250MHz

6.4 THD

The Total Harmonic Distortion is calculated by dividing the fundamental frequency by its harmonics. To determine the harmonics a sine wave with a frequency of 27.5MHz is applied to the input. To get a clearer plot a longer sampling period is taken and the sampling frequency is set at 2.5Gs/s. Fig. 6.10 shows the resulting magnitude plot.

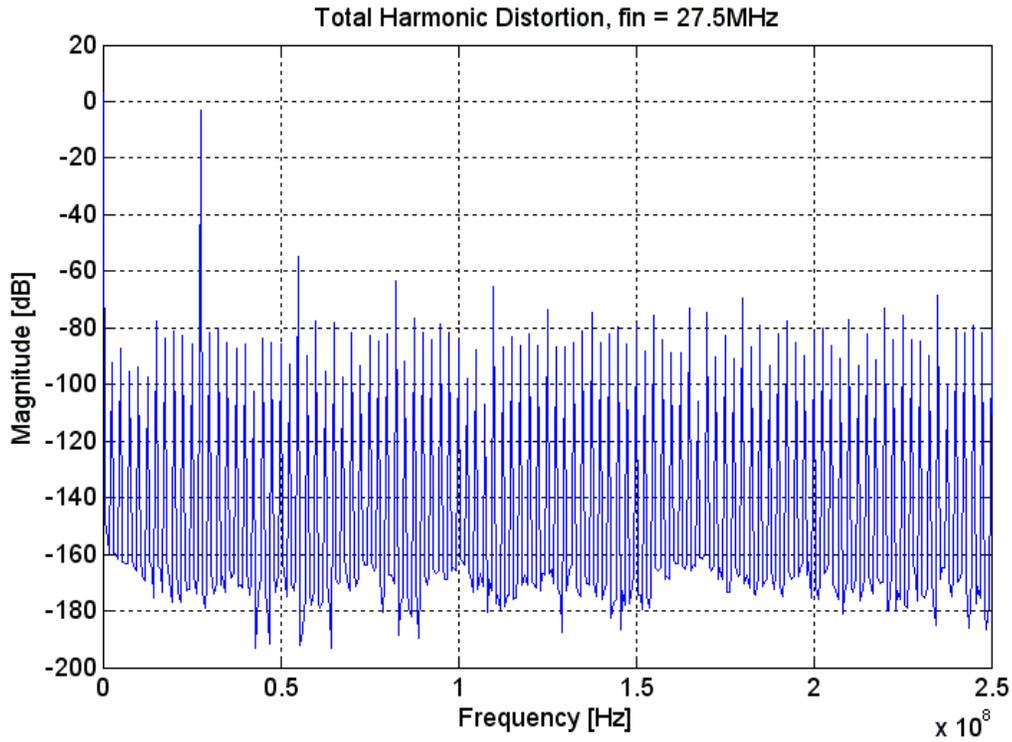


Figure 6.10: Magnitude plot for calculating the THD

The harmonics are read from the magnitude plot and written down in Table 6.1.

Frequency component	Frequency [MHz]	Magnitude [dB]	Power [W]
fundamental	27.5	-3.056	0.495
second harmonic	55.0	-54.967	$3.186 \cdot 10^{-6}$
third harmonic	82.5	-63.241	$4.741 \cdot 10^{-7}$
fourth harmonic	110.0	-65.643	$2.727 \cdot 10^{-7}$
fifth harmonic	137.5	-74.626	$3.447 \cdot 10^{-8}$
sixth harmonic	165.0	-73.144	$4.848 \cdot 10^{-8}$

Table 6.1: Magnitude and amplitude of the fundamental frequency and its harmonics

The THD is calculated in Formula x.

$$THD = 10 \log \frac{A_{2f}^2 + A_{3f}^2 + A_{4f}^2 + \dots + A_{nf}^2}{A_{1f}^2} = -50.91dB \quad (40)$$

6.5 Area

The area of all the components is listed in Table 6.2 below.

Branch	Inverter 1	Inverter 2	Total [$(\mu m)^2$]
	Area [$(\mu m)^2$]	Area [$(\mu m)^2$]	
Inv ₈ to Inv ₂₃	$15 \cdot 0.469 = 7.035$	$15 \cdot 0.938 = 14.070$	21.105
Inv ₄ to Inv ₇	$4 \cdot 0.930 = 3.720$	$4 \cdot 1.857 = 7.428$	11.148
Inv ₀ to Inv ₃	$4 \cdot 0.116 = 0.464$	$4 \cdot 0.223 = 0.932$	1.396
			33.649

a)

Charging capacitor	Area [$(\mu m)^2$]	Deg glitch capacitor	Area [$(\mu m)^2$]	Total [$(\mu m)^2$]
C ₈ to C ₂₃	$15 \cdot 5720 = 85800$			85800
C _{att2}	11441			11441
C ₇	5720	C _{d7}	5720	11441
C ₆	2860	C _{d6}	8581	11441
C ₅	1430	C _{d5}	10011	11441
C ₄	715	C _{d4}	10726	11441
C _{att1}	1430			1430
C ₃	715	C _{d3}	715	1430
C ₂	357	C _{d2}	1072	1430
C ₁	179	C _{d1}	1251	1430
C ₀	89	C _{d0}	1341	1430
	89			89
				150244

b)

Table 6.2: Component area of a) the inverters and b) the capacitors

The total area of all the inverters and capacitors is equal to $0.150 \mu m^2$.

6.6 Power consumption

Total power consumption of the DAC is measured by applying a rectangular pulse to the input and measuring the power at the node of the supply voltage. Because the DAC consumes no power if there is no input, the power consumption is code dependent, as can be seen from Fig. 6.11.

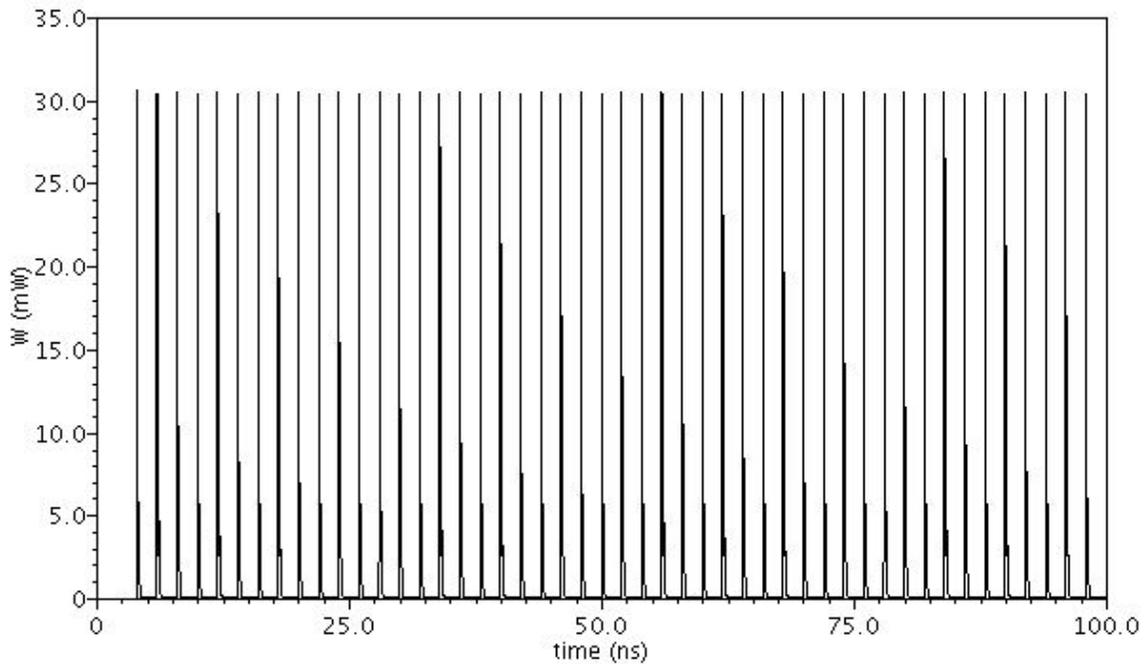


Figure 6.11: Dissipated power

The total power consumption is calculated by integrating the power over the period and normalize the outcome to one second. The resulting total power consumption is equal to 1.15mW.

6.7 MSB transition

To investigate the DNL of the DAC from CADENCE a Monte Carlo simulation of the MSB transition is performed. The resulting plot is shown in Fig. 6.12. The maximal deviation from the ideal step (bold line) is measured and should be smaller than $\pm\frac{1}{2}V_{LSB} = \pm 146\mu V$. The deviations are $178\mu V$, $-139\mu V$, $148\mu V$ and $-118\mu V$. Some are a bit over the maximum of $\pm 146\mu V$, which confirms a DNL greater than $\pm\frac{1}{2}V_{LSB}$.

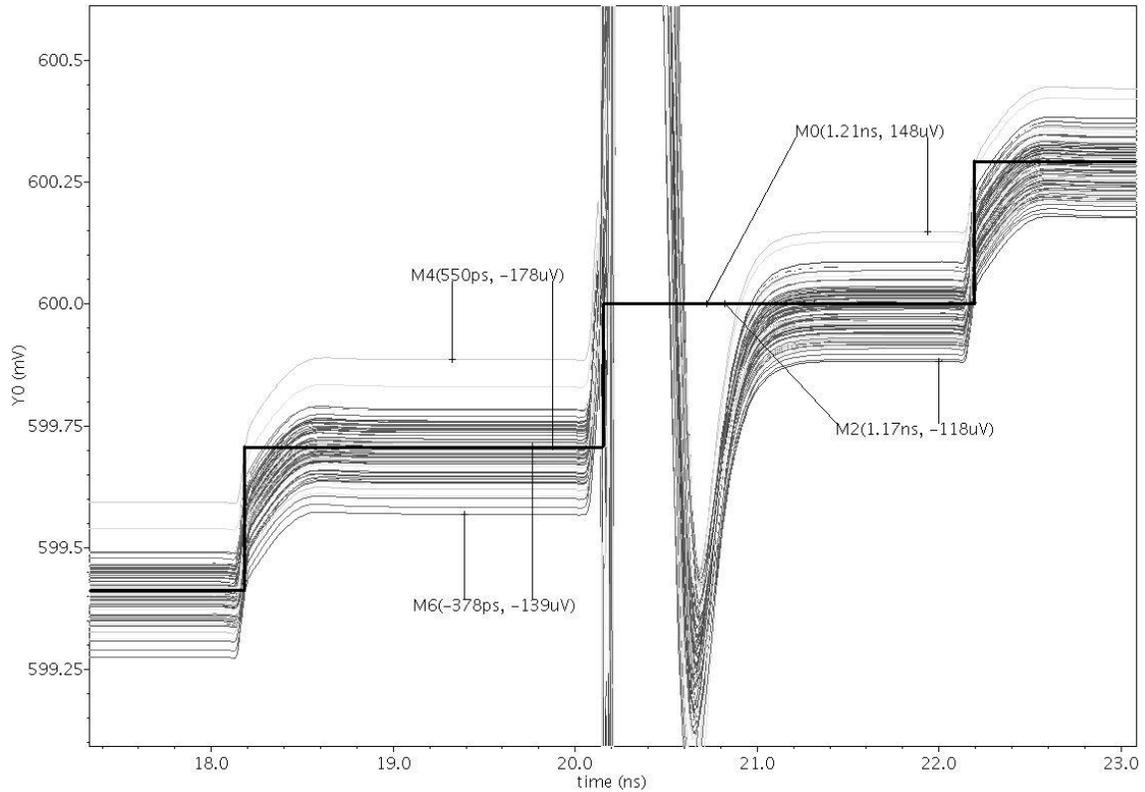


Figure 6.12: Monte Carlo simulation of MSB transition

7 Conclusions and recommendations

In paragraph 4.3 several possible architectures were presented. A fully binary weighted implementation of the DAC proved to be having too much total capacitance and monotonicity could not be guaranteed. Therefore thermometer coded and split array architectures were investigated. The choice was made to implement the first four bits in a thermometer coded part, followed by a split array and the binary weighted part. In the binary weighted part a second split array was implemented.

Next, calculations were done that resulted in the value for the minimal unit capacitor needed to ensure monotonicity. A model of the complete DAC was coded in MATLAB and from there the theoretical DNL and INL were computed.

The simulation results based on the calculated component values showed that this DAC architecture can achieve the required resolution at the sampling. Linear settling could not be achieved, mainly because of the nonlinear ON-resistance of the inverters.

A recommendation for future work is to investigate the influence of noise in the supply voltage on the overall performance. Also an output buffer is needed because at the moment the DAC cannot be loaded by an external unknown load.

8 Appendix A

8.1 MATLAB code for DNL and INL calculations

```
clf;
clear all;
N=12; %number of bits
Vref=1.2; %reference voltage
LSB=Vref/2^N; %LSB voltage
Ac=2.652e-10; %proportionality constant

for k=1:100 %calculate DNL and INL 100x
    Cunit1=16e-15; %Cunit for the LSB branch
    Cunit2=128e-15; %Cunit for the middle branch
    Cunit3=1024e-15; %Cunit for the MSB branch
    C1=[ Cunit1 Cunit1+Ac*sqrt(Cunit1)*randn(1); %ideal and with spreading
        2*Cunit1 2*Cunit1+Ac*sqrt(2*Cunit1)*randn(1);
        4*Cunit1 4*Cunit1+Ac*sqrt(4*Cunit1)*randn(1);
        8*Cunit1 8*Cunit1+Ac*sqrt(8*Cunit1)*randn(1)];
    C2=[ Cunit2 Cunit2+Ac*sqrt(Cunit2)*randn(1);
        2*Cunit2 2*Cunit2+Ac*sqrt(2*Cunit2)*randn(1);
        4*Cunit2 4*Cunit2+Ac*sqrt(4*Cunit2)*randn(1);
        8*Cunit2 8*Cunit2+Ac*sqrt(8*Cunit2)*randn(1)];
    C3=[ Cunit3 Cunit3+Ac*sqrt(Cunit3)*randn(1);
        Cunit3+Cunit3 Cunit3+Ac*sqrt(Cunit3)*randn(1)+Cunit3+Ac*sqrt(Cunit3)*randn(1);
        Cunit3+Cunit3+Cunit3+Cunit3
        Cunit3+Ac*sqrt(Cunit3)*randn(1)+Cunit3+Ac*sqrt(Cunit3)*randn(1)+Cunit3+Ac*sqrt(Cunit3)*ra
        ndn(1)+Cunit3+Ac*sqrt(Cunit3)*randn(1);
        Cunit3+Cunit3+Cunit3+Cunit3+Cunit3+Cunit3+Cunit3+Cunit3
        Cunit3+Ac*sqrt(Cunit3)*randn(1)+Cunit3+Ac*sqrt(Cunit3)*randn(1)+Cunit3+Ac*sqrt(Cunit3)*ra
        ndn(1)+Cunit3+Ac*sqrt(Cunit3)*randn(1)+Cunit3+Ac*sqrt(Cunit3)*randn(1)+Cunit3+Ac*sqrt(Cun
        it3)*randn(1)+Cunit3+Ac*sqrt(Cunit3)*randn(1)+Cunit3+Ac*sqrt(Cunit3)*randn(1)];
    Catt1=[2*Cunit2 2*Cunit2+Ac*sqrt(2*Cunit2)*randn(1)]; %attenuation capacitor1
    Catt2=[2*Cunit3 2*Cunit3+Ac*sqrt(2*Cunit3)*randn(1)]; %attenuation capacitor2
    Cdummy=[Cunit1 Cunit1+Ac*sqrt(Cunit1)*randn(1)]; %dummy capacitor

    Ctotal1=sum(C1)+Cdummy; %total capacitance branch 1
    Ctotal2=sum(C2)+Catt1.*Ctotal1./(Catt1+Ctotal1); %total capacitance branch 2
    Ctotal3=sum(C3)+Catt2.*Ctotal2./(Catt2+Ctotal2); %total capacitance branch 3

    for i=1:2^N %feed in all codes
        A(i,:)=dec2binvec(i-1,N); %12bit words
        Cbranch1(i,1)=A(i,1:4)*C1(:,1); %the bits times capacitance
        Cbranch1(i,2)=A(i,1:4)*C1(:,2);
        Cbranch2(i,1)=A(i,5:8)*C2(:,1);
        Cbranch2(i,2)=A(i,5:8)*C2(:,2);
        Cbranch3(i,1)=A(i,9:12)*C3(:,1);
        Cbranch3(i,2)=A(i,9:12)*C3(:,2);

        Vout(i,1)=Vref/Ctotal3(1)*(Cbranch3(i,1)+Catt2(1)/(Catt2(1)+Ctotal2(1))*(Cbranch2(i,1)+Ca
        tt1(1)/(Catt1(1)+Ctotal1(1))*Cbranch1(i,1)); %Vout without spreading
        Vout(i,2)=Vref/Ctotal3(2)*(Cbranch3(i,2)+Catt2(2)/(Catt2(2)+Ctotal2(2))*(Cbranch2(i,2)+Ca
        tt1(2)/(Catt1(2)+Ctotal1(2))*Cbranch1(i,2)); %Vout with spreading
    end

    for j=1:2^N-1
        DNL(j,1)=(Vout(j+1,2)-Vout(j,2))/LSB-1; %DNL calculations
        INL(j,1)=(Vout(j,2)-(j-1)*LSB)/LSB; %INL calculations
    end

    figure(1); %plot DNL and INL
    subplot(2,1,1),stairs(DNL),grid,title('DNL'),axis([1 4096 -1 1]),xlabel('input
code'),ylabel('LSB'),hold on;
    subplot(2,1,2),stairs(INL),grid,title('INL'),axis([1 4096 -1 1]),xlabel('input
code'),ylabel('LSB'),hold on;
    k
end
```


9 References

- [1] Baker, R. Jacob, CMOS circuit design, layout, and simulation, 2nd Ed., IEEE Press etc., 2005
- [2] Lin, Chi-Hung and Bult, Klaas, A 10-b 500-MSample/s CMOS DAC in 0.6 mm², Solid-State Circuits, IEEE Journal of, 1998
- [3] Elzakker, Michiel van, Low-power analog-to-digital conversion, Msc. Thesis, University of Twente, 2006

10 Bibliography

- [1] Johns, David A. and Martin, Ken, Analog integrated circuit design, Wiley & Sons, Inc., 1997
- [2] Razavi, Behzad, Principles of data conversion system design, IEEE Press, 1995
- [3] Allen, Philip E., Hollberg, Douglas R., CMOS analog circuit design, Oxford University Press, 2002
- [4] Gustavsson, M., Wikner J.J., and Tan, N.N., CMOS Data Converters for Communications, Kluwer, 2000.