NTNU

Norwegian University of
Science and Technology

Optimisation of a Pipeline ADC by using a low power, high resolution Flash ADC as backend.

Dag Sverre Høye

Master of Science in Electronics

Submission date: June 2008

Supervisor: Trond Ytterdal, IET

Co-supervisor: Terje Andersen, Arctic Silicon Devices

Norwegian University of Science and Technology
Department of Electronics and Telecommunications

Problem Description

In high speed data converters, the Flash architecture is widely used, both as a stand-alone converter, and as sub-converters in Pipeline ADCs. Both within the pipeline stage and at the end of the pipeline, Flash ADCs are used.

The tasks will be design and verification of 3-, 4-, and 5-bit Flash ADCs

An initial specification is established and a final specification will be derived by extracting the performance parameters using Matlab.

The motivation for this assignment is given by the possibility for a reduction in total power consumption when the resolution of the backend Flash ADC is increased. This will also lead to fewer Pipeline-stages and a shorter Pipeline-delay.

The focus will be on low power and good yield.

Teaching supervisor: Terje N. Andersen - Arctic Silicon Devices AS - terje.andersen@arcticsilicon.com

Suject teacher: Professor Trond Ytterdal - Department of electronics and telecommunications, NTNU - ytterdal@iet.ntnu.no

This thesis is a continuation of the project assignment with the title "Flash ADC", handed in at the Department of electronics and telecommunications 20.12.2007.

Assignment given: 15. January 2008
Supervisor: Trond Ytterdal, IET

Abstract

Flash ADCs with resolutions from 3 to 5 bits have been implemented on a transistor level. These ADCs are to be incorporated as the backend of a higher resolution Pipeline ADC. The motivation for this work has been to see how much the resolution of this backend can be increased before the power consumption becomes to high. This is beneficial in Pipeline ADCs because the number of Pipeline stages is reduced so that the throughput delay of the Pipeline ADC is also reduced. All the Flash ADCs are implemented with the same Capacitive Interpolation-technique. This technique was found to have several benificial properties as opposed to other power saving techniques applied to Flash ADCs in a project assignment done prior to this thesis. The results of the simulations show that the resolution of the backend can be increased to 5 bits both in terms of power and other static and dynamic performance parameters.

Preface

This master thesis was proposed and formulated by Arctic Semiconductor Devices. The purpose of the thesis was to implement one or more Flash ADCs suitable as back-end of a Pipeline ADC.

The motivation for the assignment was to have a low-power converter that would reduce the overall power consumption of the Pipeline ADC.

This work was a prolonging of a project assignment done by the same writer. the preceeding work included a litterature study and the building of a Matlab modell of the converters. The aim of this work was to implement the converters on a transistor level, and to simulate them to extract all the desired performance parameters.

The work has partly been done on campus at NTNU and partly at Arctic Silicon Devices' offices. Every other week there has been a meeting between the professors, Ph.D- and master-students at the department of telecommunications and electronics at NTNU to discuss problems and individual progress.

I would sincerely like to thank my teaching supervisor Terje N. Andersen and my subject teacher professor Trond Ytterdal for many fruitful conversations and for answering all of my good and not-quite-so-good questions. I would also like to thank the rest of the staff at Arctic Sillicon Devices and the staff at the analog electronics division at NTNU. It has been an inspiration to get to work with such experienced, world-class teams.

Contents

1

Introduction

1.1 The Flash Analog to Digital Converter

The Flash ADC architecture is the first choice for highest speed, low-to-medium resolution ADCs. It offers single clock cycle latency between inputs and outputs due to the simple parallel architecture. The Flash ADC is to be incorporated as the backend of a Pipeline ADC, this is explained in the next chapter.

In the project assignment done previous to this work [7] different power saving techniques for the Flash ADC as well as other applicable architectures were discussed and the most promising were modeled in Matlab for resolutions from 3 to 5 bits. The architecture of choice was a Capacitive Interpolation technique [3].

1.2 Initial specifications

Table 1.1 lists the specifications given by Arctic Silicon Devices (hereafter refered to as ASD) for the converter. A 3- 4- and 5-bit version of the Flash ADC will be implemented for comparison. Only the 5 bit specifications are listed. ENOB specifications are 2.75 bits for the 3-bit and 3.75-bits for the 4-bit converter.

The specifications for the power consumption and input capacitance will be a comparison with the current consumption of the most recent ASD Pipeline ADC, these are listed in table 1.2. Note that the numbers are from BEFORE the layout process of the ASD Pipeline ADC.

Compared to the Pipeline ADC, the 4 bit flash current consumption will be: 3-bit Flash + Pipeline stage $N = 605\mu A$ and for the 5-bit: 3-bit Flash + Pipeline stage N+ Pipeline stage $N - 1 = 985\mu$A. In addition to this the Pipeline ADC has some digital correction logic. The power consumption of this block is calculated in the Appendix.

Resolution Pipeline ADC	13 bit
Resolution Flash ADC	5 bit
ENOB 5 bit Flash ADC	~ 4.75 bits
V_{p-p} (diff. input)	2V
V_{CM} (input)	0.9V
V_{ref+}	1.4V
V_{ref-}	0.4V
f_{sample}	50 MSPS
f_{in}	15 MHz

Table 1.1: Pipeline and Flash ADC specifications

Description	Measurement
3-bit Flash	$185\mu A$
Flash input capacitance	$182fF$
Pipeline stage N	$420\mu A$
Pipeline stage N-1	$380\mu A$
Common Mode driver	$100\mu A$
Reference drivers	$2x100\mu A$

Table 1.2: Current consumption for a comparable portion of the ASD Pipeline ADC submodules.

1.3 Contents

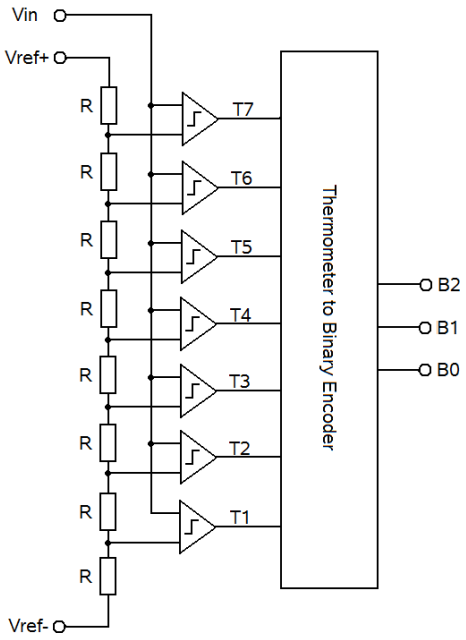
In the next chapter an introduction to the Flash and Pipeline ADC is presented along with a presentation of the Capacitive Interpolation technique. In the Design chapter the implemented converter is analyzed in-depth and the design process is described both from a top-down perspective and for each sub module. The Simulation chapter includes all the simulation results. The Remaining work chapter lists the work that wasn't finished because of too little time or because it wasn't in the scope of this work. The simulation results are discussed in the Discussion chapter and the work is concluded in the Conclusion chapter. The Appendix includes some derivations, plots and the Matlab-scripts used to simulate and extract the data from Cadence in Matlab.

2

The Flash and Pipeline ADCs

2.1 The Flash Analog to Digital Converter

The basic Flash ADC is shown in figure 2.1.



Conversion Table

$V_{in,LSB}$>	T7	T6	T5	T4	T3	T2	T1	B2	B1	B0
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0	0	1
2	0	0	0	0	0	1	1	0	1	0
3	0	0	0	0	1	1	1	0	1	1
4	0	0	0	1	1	1	1	1	0	0
5	0	0	1	1	1	1	1	1	0	1
6	0	1	1	1	1	1	1	1	1	0
7	1	1	1	1	1	1	1	1	1	1

Figure 2.1: The basic Flash ADC-architecture

The architecture can be divided into two main blocks. The first block converts the analog input to a digital thermometer code. A resistor ladder is

typically connected between two reference voltages. Thus when $2N$ (where N is the number of converter output-bits) resistors are connected, $2N - 1$ reference voltages are created. These references are connected to one of the inputs of a comparator and the converter input voltage is connected to the other input of the comparator. Thus, when the input voltage is higher than the reference, the output of the comparator is set "high". The second block of the converter is a pure digital thermometer to binary encoder providing the binary output.

2.2 The Pipeline Analog to Digital Converter

The basic Pipeline ADC is shown in 2.2.

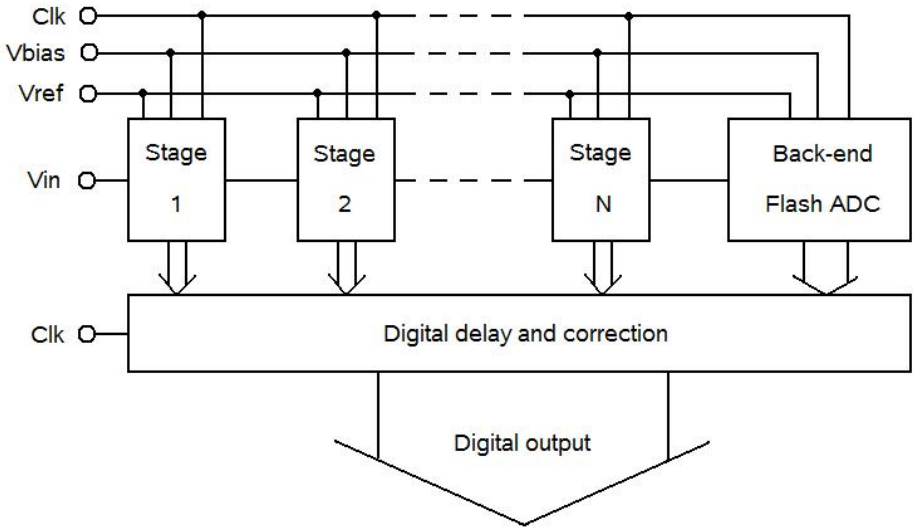


Figure 2.2: The basic Pipeline ADC-architecture

The first stage converts the MSB to binary. A DAC converts the digital value back to analog, and it is subtracted from the input of the stage. This residue is then amplified and sent to the next Pipeline stage where the process is repeated. Simultaneously the bits are sent to a series of latches which delays the outputs from the stages until all the stages are done converting. The stages can continuously work with a new sample such that the sample rate of the converter is kept very high. The Pipeline chain does on the other hand introduce a delay equal to the number of stages.

2.3 Flash ADC as the back-end of the Pipeline ADC

The last Pipeline stage is typically a Flash ADC [16]. The resolution of the Flash ADC is limited by power considerations and is therefore typically 2-3 bits. However a backend Flash ADC with greater resolution would decrease the number of Pipeline stages and therefore minimize the delay through the Pipeline. The main goal of this work is therefore to find out how much the resolution can be increased before the power consumption becomes to high.

An example of the capacitive interpolation technique chosen as the design to be implemented is shown in figure 2.3.

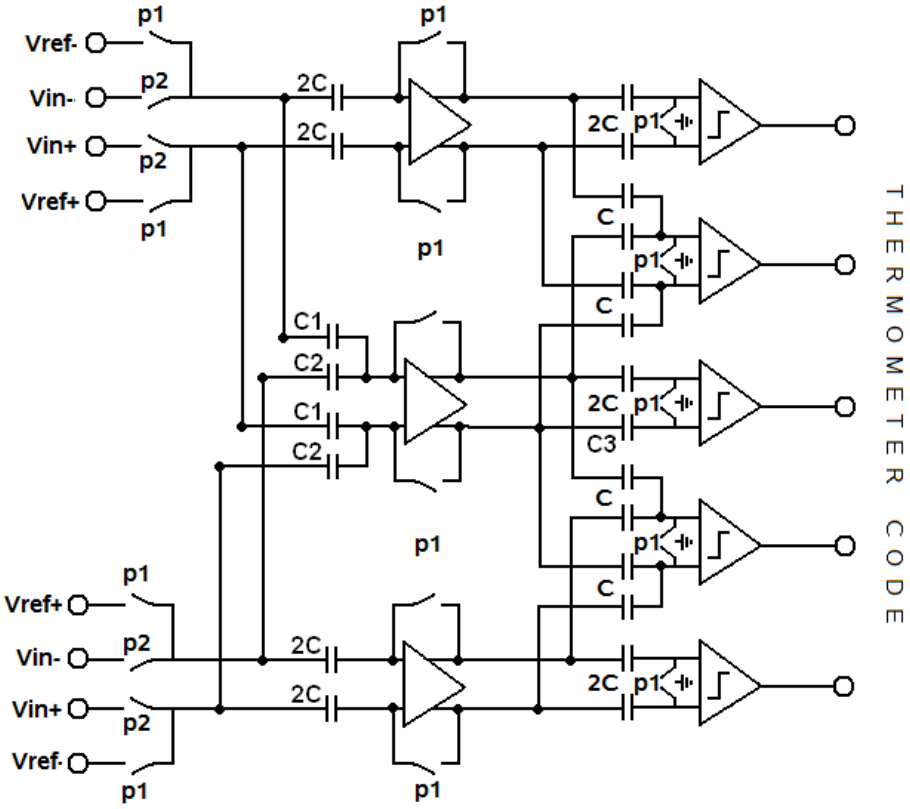


Figure 2.3: Capacitive Interpolation

A charge-transfer analysis of the V_{in+}-path in figure 2.3 follows.

When switches p1 are high, the reference voltages are loaded onto the left sides of the left capacitors. The amplifiers are shorted from input to output and therefore left in unity-gain. The common-mode plus offset voltages of the amplifiers (V_{CM+off}) are loaded onto the plates of the capacitors connected

to that node (here the middle-left amplifier):

$$q_{p_1,C_1} = C_1(V_{ref+} - V_{CM+off}) \tag{2.1}$$

$$q_{p_1,C_2} = C_2(V_{ref-} - V_{CM+off}) \tag{2.2}$$

Then switches p1 are opened and p2 closed, the inputs are connected and the difference voltage between the reference voltages and the input voltages appear on the inputs of the left amplifiers.

$$q_{p_2,C_1} = C_1(V_{in} - V_{amp}) \tag{2.3}$$

$$q_{p_2,C_2} = C_2(V_{in} - V_{amp}) \tag{2.4}$$

Where V_{amp} is the unknown input voltage of the amplifier. Then conservation of charge gives:

$$q_{p_1,C_1} + q_{p_1,C_2} = q_{p_2,C_1} + q_{p_2,C_2} \tag{2.5}$$

Solving for V_{amp}

$$V_{amp} = (V_{in} - (V_{ref-} + \frac{C_1}{C_1 + C_2}(V_{ref+} - V_{ref-}))) + V_{CM+off} \tag{2.6}$$

Thus the reference voltage is decided by the relative sizing of capacitors C_1 and C_2. Similarly solving for the input of the middle right comparator V_{comp}:

$$q_{p_1,C_3} = C_3 V_{CM+off} \tag{2.7}$$

$$q_{p_2,C_3} = C3(V_{amp} - V_{comp}) \tag{2.8}$$

gives:

$$V_{comp} = V_{amp} - V_{CM+off} \tag{2.9}$$

$$V_{amp} = V_{in} - (V_{ref-} + \frac{C_1}{C_1 + C_2}(V_{ref+} - V_{ref-})) \tag{2.10}$$

The offsets of the amplifiers are gone due to being sampled on both the input and output-capacitors during p_1. The only offset remaining is that of the comparators (not included in the equations). A good feature of the amplifiers is that if some gain is introduced, the offset of the comparators are made less significant since the gain is inversely proportional to the offset [7, section 6.1.2]. A similar analysis can be made on the V_{in-}-path. The

points where in which the differential voltages crosses each other will be the decision levels for the comparators.

The architecture of choice for the Thermometer-to-Binary encoder was the FAT-tree encoder [4]. This was due to very low power consumption becouse of very little static power being drawn at the same time as just a little part of the encoder currently working on a sample is activated as explained in [7, section 5.2].

The other traits of these architectures can be summed up: The removal of the resistor ladder also removs the static current being drawn through the references. The capacitive loading at the input is reduced to the sum of the input capacitors. The FAT-Tree encoder is made up of only OR-gates, this ensures good matching. Every throughput has the same amount of gates to propagate through, this means the propagation delay is the same for all. [7, Conclusions].

3

Design

This chapter will be an in-depth analysis of the converter-design that has been chosen. The specifications for the converter(s) given to me by ASD as well as the specifications derived from [7], will form a basis for the design both on a top-level point of view and for the smaller modules.

3.1 About the design process

The converters are implemented in a 0.18μm process from TSMC, and simulated with the Spectre simulator in Cadence 5.1.0. All the building blocks for the converters are custom designed from earlier designs, theory and specifications, with the exception of the comparator which is a design from ASD that may be modified to satisfy the minimum power requirement

3.2 Top level Design

A top level description. Known problems are recognized and solutions for these are chosen.

3.2.1 Overall description

A block diagram of the converters is presented in figure 3.1.

The first stages are the Capacitive Interpolation part of the converter as described in the introduction chapter. The number of these stages are decided by the resolution of the converter, the interpolation factor and the wanted gain. Since parallel converters with lower resolutions eases the demand on the comparators by accepting a higher offset (e.g. a 4 bit converter has twice as high tolerance on this as a 5 bit converter [7]), a lower gain

can be accepted and therefore fever stages are necessary. The interpolation factor is the number of interpolation points per input (for one stage). If one interpolation point is created between two inputs, the interpolation factor is 2. The interpolation factor can therefore be calculated as $\frac{N_{outputs}+1}{N_{inputs}}$, where N is an integer. As an example a 5-bit converter needs 31 taps for the comparators. If we choose a constant interpolation factor of 2, 5 interpolation stages will be needed!

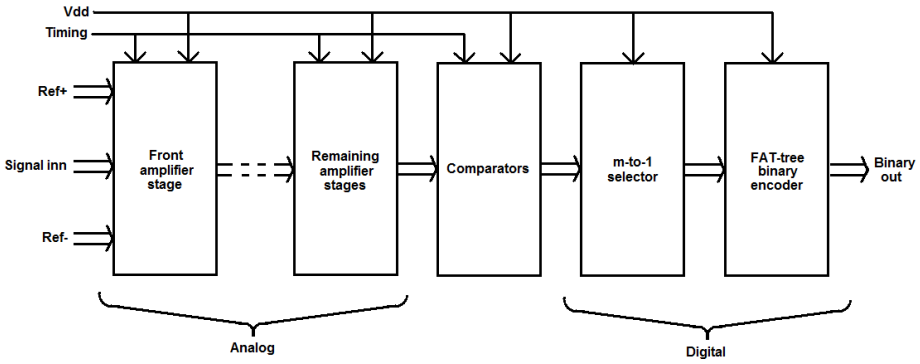


Figure 3.1: Layout of the converters

The next stage is the comparators. Because of the relatively low sample frequency (in Flash ADC-terms) and because the system is sampled, as little as possible static current is wanted. The obvious choice is then on dynamic comparators, since they are only drawing current for a short time during comparison. For a discussion on these comparators see [10], they will also be explained later in the section describing the comparators that are being utilized in this project.

In figure 3.1 the analog and digital parts of the converter are marked. The focus on the design has been on the analog part since the digital will not draw any static current and therefore will contribute a lot less to the overall power consumption. Some considerations have nevertheless been made when choosing the FAT-tree thermometer to binary-encoder as mentioned in the introduction and described in [4]. Figure 3.2 shows the 4-bit FAT-tree encoder.

The next rightmost stage in figure 3.1 shows the M-to-1 line selector [15] that is consisting of 2NAND-gates and inverters, and has the purpose of sending only the output from the comparator at the conversion point (where there are only zeroes above and ones below the point) of the thermometer code through to the encoder. It's functionality is shown in figure 3.3. An alternative to the 2NAND gate are 3NAND gates [17], in which two of the inputs are inverted and coming from the two comparators above the transition point. The 3NAND output is only high if two "0" are present over a

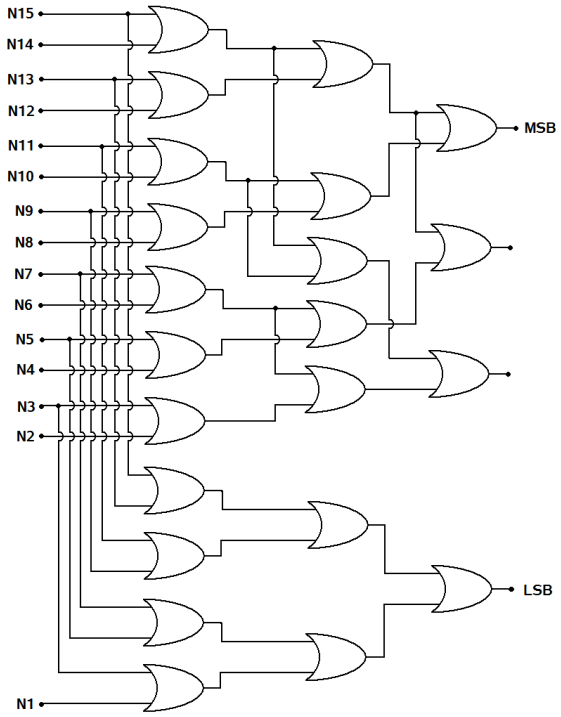
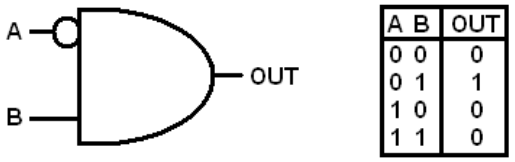


Figure 3.2: FAT-tree encoder

"1". This technique makes shure no "1" above a "0" is sent to the output of the converter (also known as bubble-errors). At a very low power cost this provides efficient digital correction.



A	B	OUT
0	0	0
0	1	1
1	0	0
1	1	0

Figure 3.3: M-to-1 line selector functionality

The FAT-tree is an array of solely OR gates that directs the input from the M-to-1 to the right converter output. Each signal passes an equal amount of gates such that the latency is the same for all. The homogenousity of the encoder is also favorable when it come to the layout.

3.2.2 Top level issues and considerations

Interpolation factor

The interpolation factor for each stage is limited by the minimum capacitive load that is tolerable on the output of each amplifier. By increasing the interpolation factor a smaller number of stages and thus a smaller number of amplifiers is needed, but since a heavier capacitive load is connected to the output, other issues arises: The amplifiers will have to draw more current to meet the bandwidth requirements. Increasing to a factor 4 (a factor of 3 is unwanted since the total amount of interpolated levels will not ad up to a power of two when used) leads to a 4 times higher capacitive load. Assuming that there are 3 converter input taps, 12 amplifiers have to be used to get to 31 outputs in the case of a 5 bit converter and a interpolation factor of 4. The same setup using the factor 2 leads to 34 amplifiers.

Due to the known fact that when doubling the capacitive load of an amplifier, twice as much power is needed to obtain the same bandwidth ($\frac{g_m}{C}$ [1]). An interpolation factor of 2 is then preferred since the amount of amplifiers is increased by 283% while the capacitive load is reduced to 25% as opposed to a factor 4. A minimum power transfer is thus achieved. In addition the larger capacitors used with higher interpolation factors takes up a lot more area on the chip as well as there can be a matching issue due to capacitors of many sizes are used.

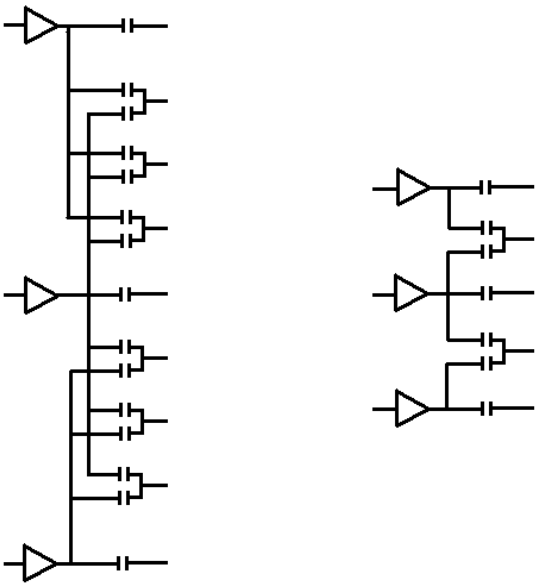


Figure 3.4: Sketch of interpolation factor 4 and 2 from left to right. A higher interpolation factor leads to fewer amplifiers and stages, but increases the power consumption due to a larger capacitive load.

Timing

Correct timing is important in all sampled systems. For the Capacitive Interpolation block there are two obvious clock phases as mentioned in the Introduction chapter. In the first clock phase (*clk* in figure 3.5) the references are sampled on the input capacitors and the amplifiers are in reset mode, meaning they are settling to the common-mode voltage which is, together with any additional offset voltage sampled on both input and output capacitors of the amplifiers. In the next clock phase (*clkinv*) the inputs are switched on the input capacitors and the amplifiers are put in open loop (amplification mode). When doing this it is important that the clock phases between reset and amplification mode do not overlap. The inputs of the amplifiers have to be floating before the input signal is switched on to ensure that no charge injection from the switches is directly affecting the signal. Therefore the clock signal that sets the amplifiers in reset mode (*clkamp*) turns low 0.5ns before *clk*. This is common in bottom plate sampling. The last clock phase is the comparator latch phase (*complatch*). This phase starts just before the amplification mode is finished and lasts until the outputs are retransmitted to whatever circuitry is beyond the outputs of the converter. For this converter the latch phase will last until the reset phase is done. The clocks that is going to be used during simulation has a transition time of 200ps to add some realism since it is impossible to design a clock with

infinitely low transition time. Figure 3.5 is a printout of the clock signals from Cadence.

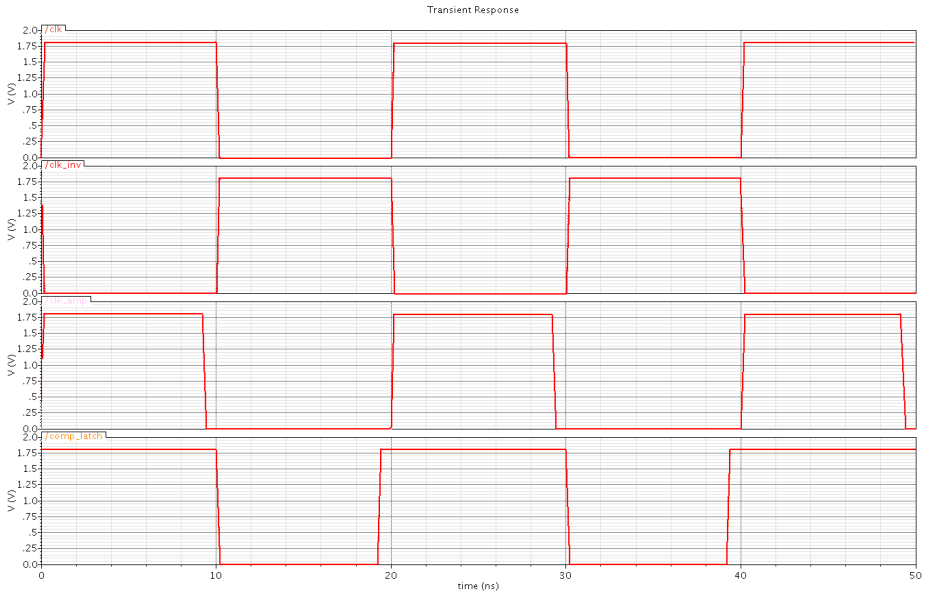


Figure 3.5: Clock tree extracted from Cadence. From top to bottom: *clk*, *clkinv*, *clkamp* and *complatch*

Input capacitance

The input capacitance is important because the power consumption of the last Pipeline stage before the Flash ADC is proportional to this parameter. The input capacitance of this type of converter is consisting of the capacitors connected directly to the inputs, the gate source capacitances of the input amplifiers and the source-drain, source-bulk and drain-bulk capacitances of the input switches. Since there is a number of input amplifiers the input capacitance also depends on the interpolation factor of the first stage. The cost of lowering the input capacitance is then to add an extra interpolation stage. A simulation of the actual input capacitance is included in the Simulations chapter.

Power consumption

The power consumption of the analog part of the converter is an issue that is difficult to estimate correctly before any work on each module is complete. The power consumption of the digital part of the converter is therefore estimated first. The power of the digital circuitry can be estimated by the

formula [18]:

$$P = mfE_{tr} + P_{stat} \tag{3.1}$$

Here, f is the sampling frequency times two and m is the number of gate-transitions per period. Looking at figure 3.2 which displays a 4 bit version of the FAT-tree encoder and remembering that only one line from the output of the comparators are active for each sample, three logical OR gates are required for each sample to traverse to the correct output. In the 3 and 5 bit versions the number of logical OR-gates are respectively 2 and 4. In addition the M-to-1 selectors consists of one NAND-gate with one inverted input. In standard CMOS OR-,AND- and NOT-gates (inverters), the number of transistors are 6, 6 and 2 [6, p 86].

$E_{tr} = CV_{DD}^2$ is the energy per gate transition with $C = WLC_{OX}$ being the capacitance of each gate and V_{DD} the supply voltage. A very pessimistic approach relative to the model specifications for the TSMC018 process (which is confidential information and thus can not be listed here) of 10fF will be used for the gate capacitance estimation.

$P_{stat} = N_g V_{DD} I_0$ is the static power consumption where N_g is the available gates in transition on average and I_0 is the quiescent current. I_0 which is essentially the total leakage current for the total amount of MOS-gates in the digital circuitry is considered small as opposed to the switching current that it can be omitted.

The estimated power consumption and corresponding current consumption of the digital part of the converter is listed in the table below.

Resolution	m	P	I
3 bit	20	$64\mu W$	$36\mu A$
4 bit	26	$84\mu W$	$47\mu A$
5 bit	32	$103\mu W$	$57\mu A$

Table 3.1: Estimated power consumption of the digital circuitry using the formula $P = mfV_{DD}^2C$; with $f = 100MHz$, $V_{DD} = 1.8V$ and $C = 10fF$

The analog part then has to consume less than about $900\mu A$ for the 5 bit converter. Giving a interpolation factor of 2, 34 amplifiers are needed as well as 31 comparators. In the calculations, the bias circuitry for the amplifiers also has to be counted in. This will be dealt with in the next section.

3.3 Design of the sub-modules.

A detailed discussion on the various sub-modules follows. Some theory and estimation as well as a description of the preferred solutions with plots.

3.3.1 Comparators

ASD provided the dynamic comparator that was used in this work. The comparator is shown in figure 3.6.

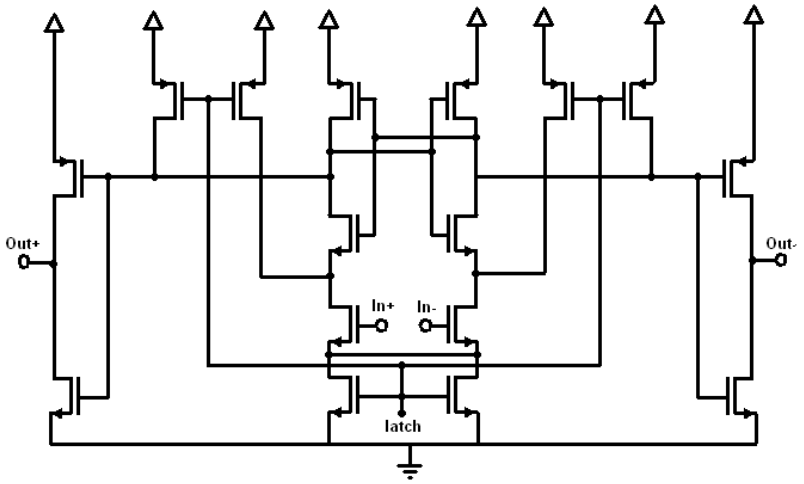


Figure 3.6: The comparator from ASD

The operation for this comparator is standard to any dynamic comparator [10]. In the reset phase the latch signal is connected to ground, and since the pMOS connected to the latch signal is conducting; the outputs of the input differential pair as well as the output transistors are forced to V_{DD}. The four nMOS-transistors connected to ground are turned off so no current is flowing. When *latch* goes high, the outputs as well as the latch are released from V_{DD}. The difference signal on the inputs are then regeneratively amplified through the latch and sent to the output stage which now serves as an inverter. When the transient diminish, the outputs are either connected to ground or V_{DD} and there is no current path left. From [7] a Matlab-modell of the converter was simulated and performance parameters were measured when the offset standard deviation of the comparators were altered:

The simulation results from the implemented comparator are shown in figure 3.7

Resolution	ENOB	σ	$\sigma, A = 5$
3 bit	2.75 bit	21mV	104mV
4 bit	3.75 bit	11mV	56mV
5 bit	4.75 bit	5.5mV	29mV

Table 3.2: Matlab simulations: The standard deviation for the comparator offset when the ENOB of the converter is approx 2.75, 3,75 and 4,75. In the right column, a gain of 5 in the capacitive interpolation-network were added

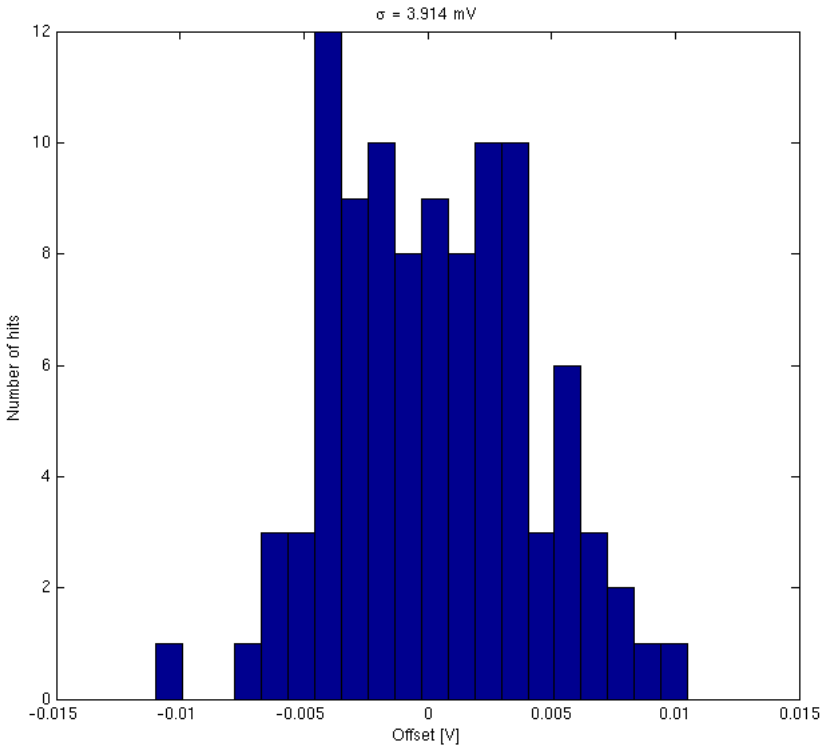


Figure 3.7: Offset of the implemented ASD-comparator. Monte Carlo 100 runs mismatch only.

The standard deviation is as can be seen $3.914mV$, which is a lot lower that the requirement from the Matlab simulation. Note that in the Matlab simulation all other non idealities are set to zero. This still shows that the gain of the preceding stage does not need to be high. The offset of the comparator is decided by the offset of the input differential pair which is calculated as [11]:

$$\sigma_{off} = \sqrt{\frac{A_{V_T}^2}{W*L} + \frac{(V_{GS}-V_T)^2}{4} * \frac{A_\beta^2}{W*L} + (\frac{\sigma_{R_L}}{A_0})^2} \tag{3.2}$$

V_T is the threshold voltage, A_β the relative current factor mismatch, σ_{R_L} the standard deviation of the load resistance mismatch and A_0 the gain of the stage. Since $V_{GS} - V_T$ is often small [11], the second term is nearly zero so the offset becomes a function of the threshold voltage, the load mismatch and the gain. The offset is then approximately inversely proportional to the root of the gate area of the differential pair transistors.

The current consumption of one comparator was measured in cadence by averaging the current through the V_{DD}-node for a few clock periods. The inputs were set so that the comparators positive and negative outputs went high for every other period. The current was measured to $5.31\mu A$, which gives a total comparator current consumption of about $165\mu A$ for the 5 bit, $80\mu A$ for the 4 bit and $37\mu A$ for the 3 bit.

The comparator delay was measured as 120ps. A plot of this is included in figure 6 in the Appendix.

3.3.2 Amplifiers

As in [3], differential pairs will be used as amplifiers (see figure 3.9) These simple, standard analog building blocks serve the purpose of amplifiers in the Capacitive Interpolation block.

To match the current consumption of the ASD Pipeline ADC, the current for the Capacitive Interpolation block ($I_{TOTAL,CONVERTER} - I_{DIGITAL} - I_{COMPARATOR}$) for the 5-bit, 4-bit and 3-bit has respectively to be about $760\mu A$, $480\mu A$ and $110\mu A$, which means that the maximum power allowed in each amplifier is $24\mu A$, $34\mu A$ and $22\mu A$ including the biasing circuitry if the first stage has an interpolation factor of 4. Since most of the power is spent in these amplifiers, this is where the focus on power saving will be.

In the section describing the comparators, it was noted that their offset were quite better than the specifications. Therefore the gain before the comparators does not need to be significantly large. The gain of each inter-

polation stage is also limited by the required linear range of the amplifiers. These demands are highest at the front, as sketched in figure 3.8.

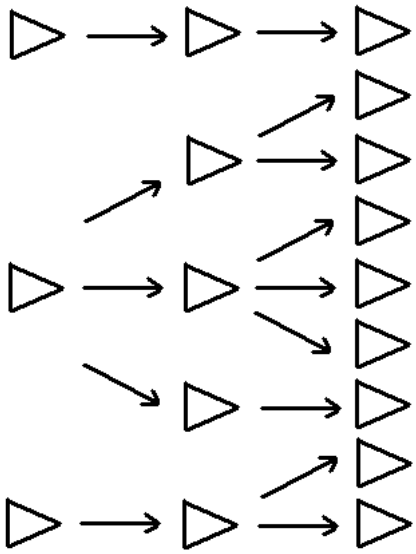


Figure 3.8: The sketch shows the demand on linear range for the amplifiers. A first stage with an interpolation factor of two will have to have a linear range of half the full input range, while the next stage (or similarly a first stage with interpolation factor four only needs to be linear for 1/4 of full input range.

The differential gain of the differential pair is $A = -g_m R_D$, or if a load is taken into account $A = -g_m(R_D||Z_L)$ [5, ch 3.8] [1, ch 7.2.1] Where g_m is the transconductance of any of the devices assuming they are similar and R_D is the resistor connected to each drain. When the current available for each differential pair is smaller than $20\mu A$ this resistor has to be around $\frac{1.8V}{20\mu A} = 90k\Omega$ and larger. This makes it difficult to keep the gain down. A solution to this is to use a source degenerating resistor R_S like in figure 3.9 (see e.g. [13]).

The introduction of R_S changes the gain of the amplifier to:

$$A_{SD} = -\frac{R_D||Z_L}{R_S + \frac{1}{g_m}} \tag{3.3}$$

Please see the Appendix for the analysis that results in equation 3.3. With both R_S and R_D much larger than $\frac{1}{g_m}$, a smaller gain (and larger linear range) is achieved by direct scaling of the two resistors. The bandwidth of the amplifier can be written in this form:

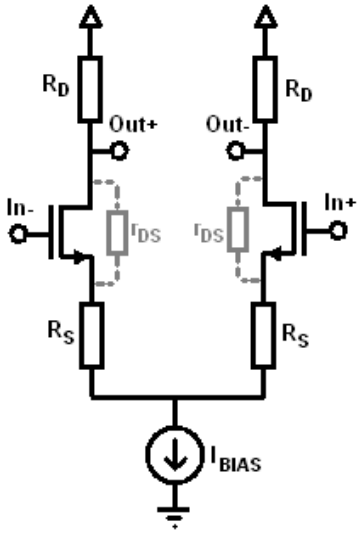


Figure 3.9: Differential pair with resistive source degeneration.

$$f_t = \frac{1}{2\pi\tau_t} \tag{3.4}$$

f_t is the 3dB-frequency and τ_t is the time constant. To find the exact bandwidth of each amplifier a 10kΩ test resistor is connected to the input of an interpolation stage with 40fF capacitors between the resistors and the input of the amplifiers. Then an AC simulation is performed to find the 3dB-frequency of the node between the resistors and the capacitors. For the amplifiers, R_D is 70kΩs and R_S is 7.5kΩ. The sizes of the transistors are W= 5μ and L= 180nm. The 3dB frequency was found to be at 1.59GHz. C_L is then found by using: ([5, page 155]):

$$|A_{3dB}| = \frac{1}{\sqrt{2}} = \frac{1}{\omega_{3dB} R_{test} C_L} \Rightarrow C_L = \frac{1}{\sqrt{(2)}\pi f R_{test}} \tag{3.5}$$

C_L was found to be 14.2fF. This is the load capacitance for each differential pair and is the series capacitance of the input 40fF capacitors and the gate capacitance of the differential pair nMOS.

AC analysis and transient analysis of the differential pair biased with 18μA is shown in figure 2 in the appendix. For the transient analysis the input is a ramp from V_{REF-} to V_{REF+} and the input of the AC analysis is a sine with a 0.9V offset and 0.5V amplitude. It can be seen that the common mode voltage of the output is $1.20V$ as opposed to $0.9V$ for the input. This is because the amp is biased with a cascode current mirror (as

described later in this section), as well as having the source degeneration resistor which pushes the common mode even higher. The dynamic range of the differential pair is therefore a little compressed. The gain at 100MHz is about 10.5dB.

On biasing

The simplest way of biasing is to use a current mirror. The goal of the biasing circuitry is to generate a stable current that is independent of what is going on at the output, this is achieved with a high output impedance. A good solution to this is the cascode current mirror, which will be used in this design.[1, chapters 6.3 and 6.12.1] Both mirror types are shown in figure 3.10. The current mirrors used to bias the amplifiers with 18μA in this work have a 1:6 ratio.

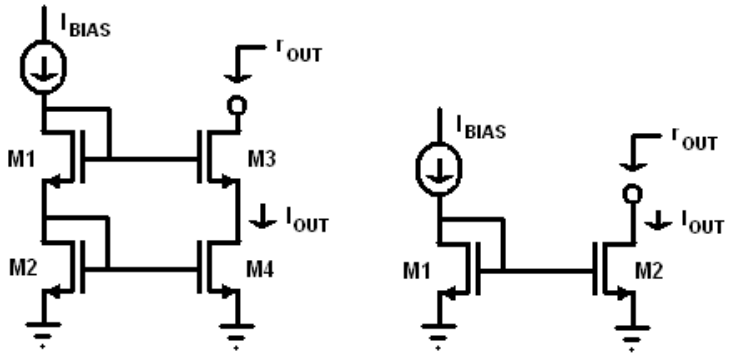


Figure 3.10: Basic current mirror to the right and cascode current mirror to the left. Output resistance of the basic current mirror is $r_{OUT} = r_{o2}$ and of the cascode current mirror $r_{OUT} = g_{M3}r_{o3}r_{o4}$.

Offset cancellation

Figure 3.11 shows some key parameters of the amplifiers as well as the impact of the amplifier offset cancelation feature of the Capacitive Interpolation technique described in the Introduction chapter. The circuit for this is shown in figure 3.12. The first amplifier is the actual differential pair which is biased at 18μA and used in the converter. This amplifier is built with the mismatch devices from the TSMC018, so that when running a Monte Carlo statistical analysis in Spectre the mismatch parameters differs in each run according to the specifications from TSMC. The second amplifier is similar to the first, but the device is non-mismatch sensitive. The setup is as follows. The differential input is a pulse altering between 0.8V and 1.0V so that the

difference voltage appears in clock phase 2 just as in the actual converter. This transient analysis is run 20 times Monte Carlo-vise. The sampling frequency is 50 MS/s. The bottom plot in figure 3.11 is the outputs from the first amplifier. Since these outputs are taken before the output capacitor, the offsets are not yet cancelled and it is clearly having an impact on the signal! The top plot is taken at the input of the second amplifier, where the offset canceling has occurred. The mismatch-induced offsets are now almost completely canceled out by the output sampling capacitor. The actual gain is about 2.6, but because of the mismatch it is varrying from about 2.2 to 2.8. This is going to have some impact on the overall resolution of the converter since there will be some variance in the generated reference voltages.

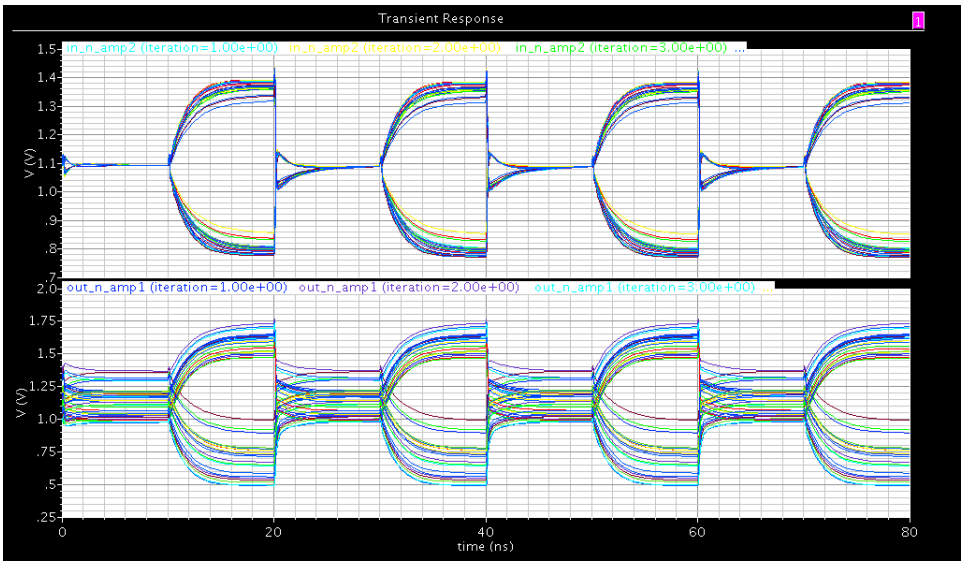


Figure 3.11: Transient analysis of the circuits shown in figure 3.12

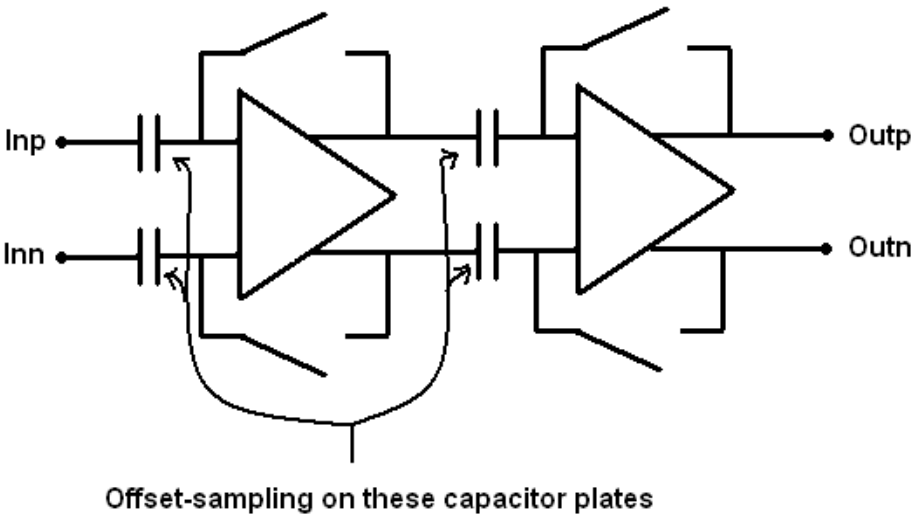


Figure 3.12: Circuit to test the amplifiers.

3.3.3 Digital

There are three types of logical gates that has to be implemented to fill the functions of the thermometer to binary encoder, the inverter, the NAND and the NOR gates. These will be implemented with minimum transistor gate area to decrease the gate capacitances and the total power consumption.

The delay of the digital circuitry is included in the table below. the time is measured between the outputs of the comparators and the outputs of the FAT-tree encoder. Plots of the fall and rise delay of the 5 bit converter is included in the appendix.

Converter	Rise delay	Fall delay
3-bit Flash	300ps	490ps
4-bit Flash	340ps	680ps
5-bit Flash	410ps	890ps

Table 3.3: Delay of the digital circuitry.

3.3.4 Switches

There are two different purposes for the switches in this design. The switches that shorts the inputs and the outputs of the amplifiers during the reset phase, and the switches that alternates the references and input signal being connected to the converter input during respectively the reset phase and the

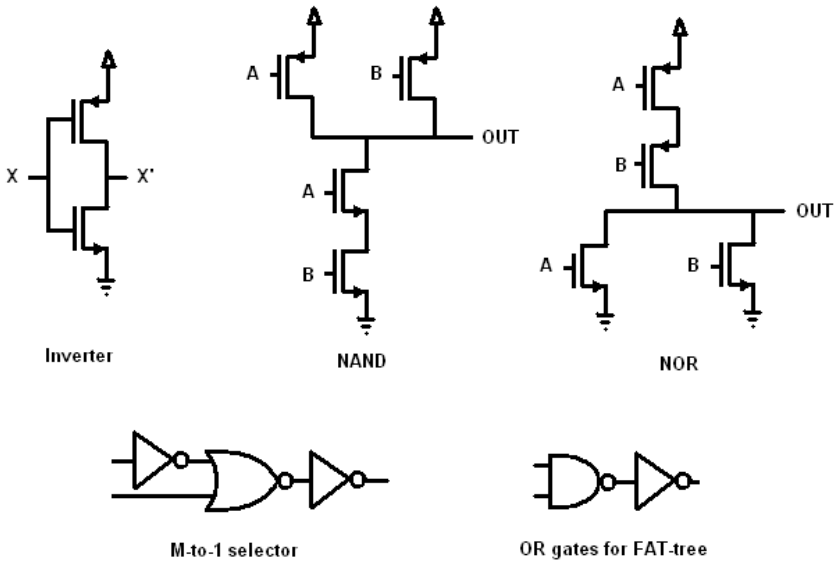


Figure 3.13: Transistor models of the logic gates used. All inverting logic. Slices of M-to-1 and FAT-tree below.

amplification (tracking) phase. The most important feature of these switches is that there are minimum charge injection when the switch turns off. The input switches also must have the range of the input signal. The CMOS-transmission gate is the most obvious choice for the input switch since it has the range needed. On the other hand there will be some charge injection [5, chapter 10]. One way to cancel the charge injection is to add a dummy transistor of half the width on one or both sides of the switching transistor [2] shown in figure 3.14.

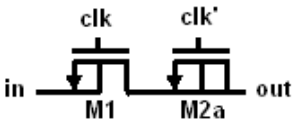


Figure 3.14: NMOS-switch with half size dummy that cancel the charge injection on output.

When a MOS switch turns off, half the charge held by the gate capacitance of a MOS (approx $C_G = C_{OX}WL$, when ignoring the overlap capacitance) is injected to the drain and source. Connecting a dummy with half the size of the switch transistor which turns on when the switching transistor

turns off and vice versa will cancel out the injected charge [2]. Figure 4 in the appendix shows the clock, input and output of a standard CMOS-gate and figure 5 in the appendix the same but a CMOS gate with dummy device charge injection cancellation. As can be seen, when the regular CMOS transmission gate turns off, the charge injected in the output node makes the voltage of that node a little lower that it should. This is almost eliminated when the dummy is introduced. The sizes of the switch transistors are the same for both the regular and the switches with dummies. In this design two types of switches are used depending on switching the signal directly or just shorting the amplifiers as said in the first paragraph of this subsection. The signal-switches will be nMOS and pMOS in parallel (CMOS), and the shorting switches will pe pMOS-only switches. The CMOS switches was measured to have a 412Ω on-resistance and the pMOS switch measured 1.72kΩ.

4

Simulations

In this chapter most of the simulation results are presented. Various power parameters, input capacitance, stage amplifier gain, differential and integral nonlinearity (DNL and INL) and a frequency analysis including e.g. signal to noise ratio (SNR), signal to noise-and-distortion ratio (SNDR) and the efficient number of bits (ENOB) are plotted. Explanations on each parameter analyzed is included at the start of each subsection. There are two main sections. The first section holds the results of the converters that have total power specifications predicted to be compareable with the specifications from ASD (Like in the design chapter, biased with 18μA). The second is a variant where as much power as possible is reduced in the Capacitive Interpolation block. All the converters was simulated with an interpolation factor of 4 in the input stage.

4.1 Simulations of converters with power predicted to be comparable with the ASD specifications

4.1.1 INL/DNL

These plots have been made the following way. A transient analysis with an input ramp signal was simulated in Spectre. The length of the analysis was calculated such that 16 samples per (ideal) output code was generated. This limits the resolution of this analysis to $\frac{1}{16}$ of an LSB. This is still a good enough resolution since $\frac{1}{16}$ of an LSB is small compared to the typical demand on the INL and DNL (typical $\frac{1}{2}$LSB and 1 LSB, see for instance [5])

The data is fed into Matlab where the data is converted to thermometer code V_{TERM} using an ideal DAC. This signal is compared to an ideal ramp signal V_{RAMP} (with the same length as V_{TERM}) generated in Matlab, and

the mean of the two nearest values of the ramp signal when a transition has occurred is loaded into a matrix.

If

$$V_{TERM}(s) > V_{TERM}(s-1) \tag{4.1}$$

then

$$V_{TR}(n) = \frac{V_{RAMP,(s-1)}(n) + V_{RAMP,s}(n)}{2}, n = 1, 2, ..., 2^N \tag{4.2}$$

$V_{TR}(n)$ is the value of the ideal ramp signal at transition n of the input thermometer code and s is the index of the ramp (and thermometer code) vector. The value is only added when the thermometer code changes value. This gives a good approximation of the value of the input for the output thermometer transitions.

The DNL and INL is calculated as in equations 4.3 and 4.4. Note that the INL is the best-fit straight line approach.

$$DNL(n-1) = \frac{V_{TR}(n) - V_{TR}(n-1)}{V_{LSB}} - 1, n = 2, 3, ..., 2^N \tag{4.3}$$

$$INL(m-1) = \frac{V_{TR}(n)}{V_{LSB}} - (n-1), n = 2, 3, ..., 2^N \tag{4.4}$$

N is the number of bits of the converter and V_{LSB} is the LSB voltage.

3 bit converter

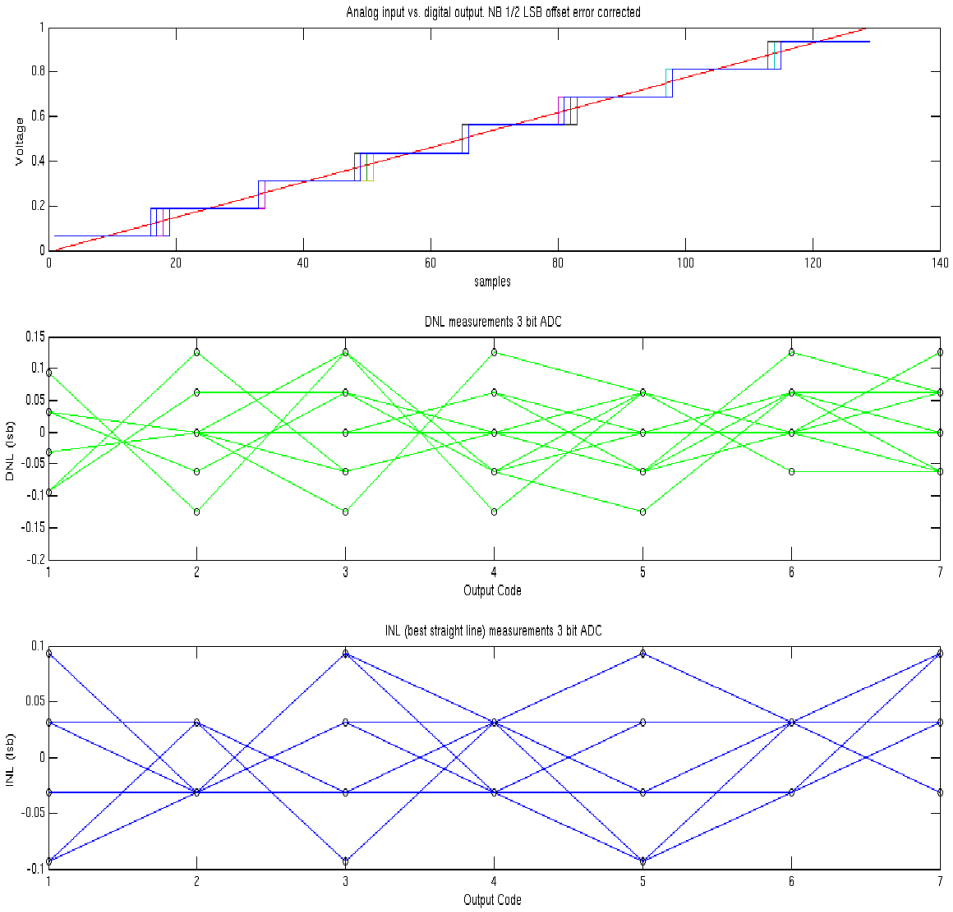


Figure 4.1: DNL and INL of the 3 bit converter.

4 bit converter

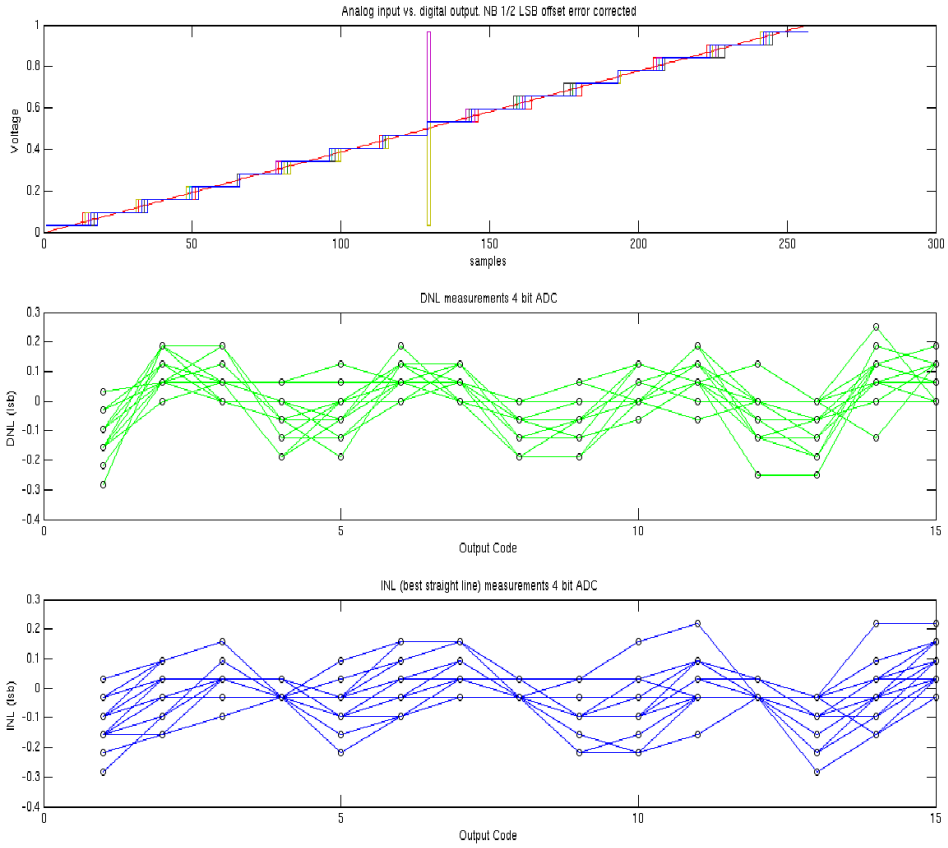


Figure 4.2: DNL and INL of the 4 bit converter

5 bit converter

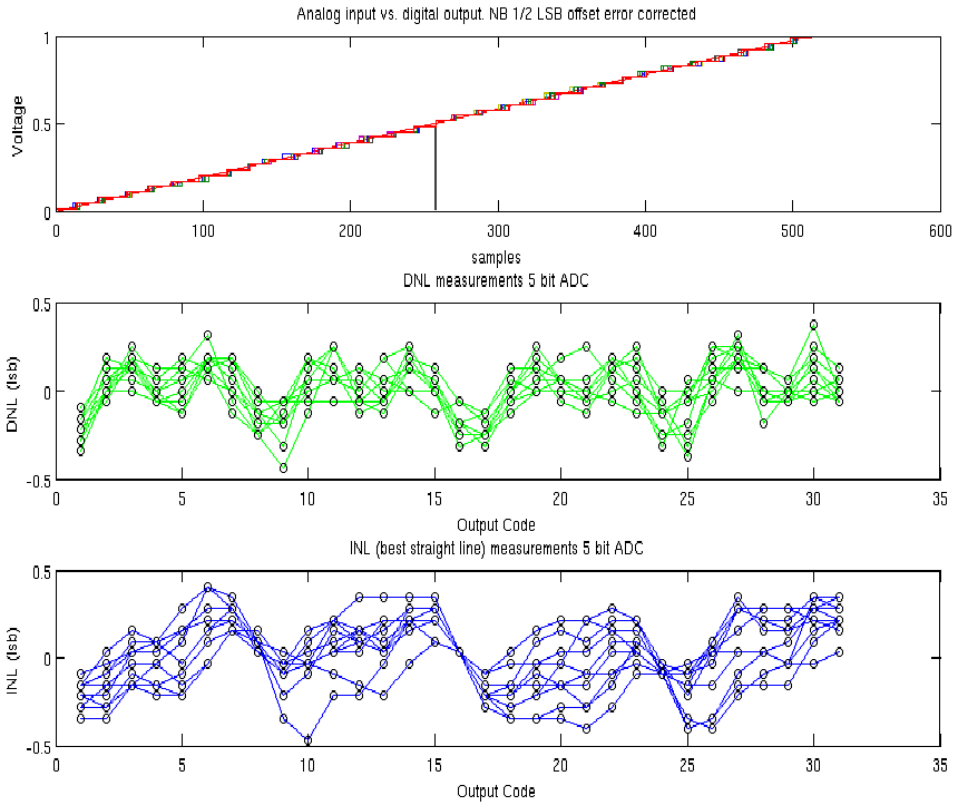


Figure 4.3: DNL and INL of the 5 bit converter

4.1.2 Frequency analysis with dynamic parameters

These simulations were done with sample frequencies F_S and input sine frequencies f_{in} according to the coherent sampling theorem [12] stating that:

$$\frac{f_{in}}{F_S} = \frac{N_{WINDOW}}{N_{RECORD}} \tag{4.5}$$

Where N_{WINDOW} is the number of input sine cycles and N_{RECORD} is the number of sampling points during the sampling window. N_{WINDOW} will have to be a odd number (most preferably a prime number) so that every possible quantization level has "hits" and is represented in the output vector. Please read the Maxim application note [12] for a thorough discussion on coherent sampling.

The converters has been simulated twice with two different simulation lengths:

No of runs	N_{RECORD}	F_S	f_{in}	N_{WINDOW}	Mismatch
1	4096	49.9712 MS/s	15.0182 MHz	1231	No
15	256	49.9968 MS/s	15.0381 MHz	77	Yes

Table 4.1: Simulation setup. Using coherent sampling

The first 4096 points simulation was done to get a nice plot and has little value beyond that because the mismatch parameters were turned off. The 5 bit converter 4096 points run took about 24 hours, so to spare a lot of time mismatch was simulated with only 256 points in 15 runs. In the tables below the ENOBs for each run is presented and the worst case run is highlighted. Note that the sampling frequency in the frequency plots are normalized to 50MS so that the input frequency seems a little higher than it actually is. The Matlab script doing the fft-analysis is a modified script originally made by Carsten Wulff (See Appendix).

3 bit converter

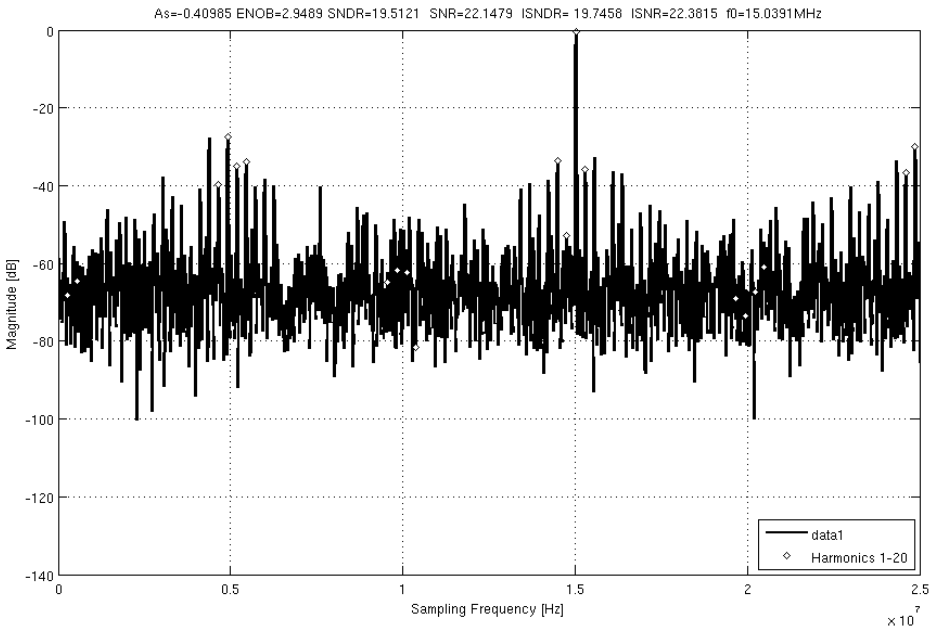


Figure 4.4: FFT plot showing the dynamic performance parameters of the 3 bit converter. No mismatch.

No of run	1	2	3	4	5
ENOB (bits)	2.90	2.90	2.89	2.88	2.89
No of run	6	7	8	9	10
ENOB (bits)	2.88	2.88	2.90	**2.82**	2.89
No of run	11	12	13	14	15
ENOB (bits)	2.90	2.84	2.89	2.87	2.86

Table 4.2: ENOBs for the 3 bit Flash ADC. 15 runs Monte Carlo simulation. Mismatch only. Amplifiers biased with 18μA

4 bit converter

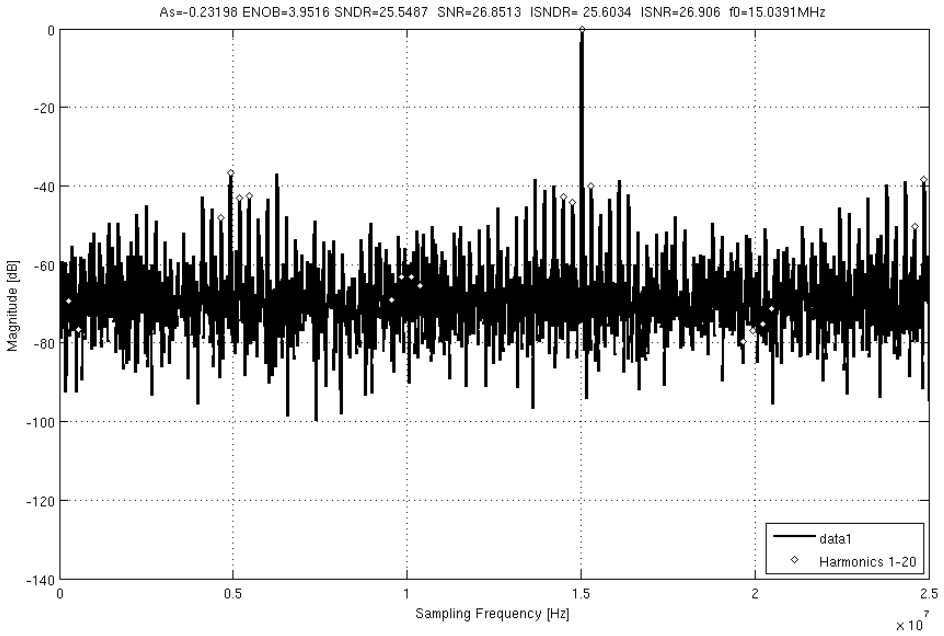


Figure 4.5: FFT plot showing the dynamic performance parameters of the 4 bit converter. No mismatch.

No of run	1	2	3	4	5
ENOB (bits)	3.94	3.92	3.91	3.92	3.88
No of run	6	7	8	9	10
ENOB (bits)	3.83	3.89	3.81	3.91	3.91
No of run	11	12	13	14	15
ENOB (bits)	3.90	3.91	3.95	3.94	**3.77**

Table 4.3: ENOBs for the 4 bit Flash ADC. 15 runs Monte Carlo simulation. Mismatch only. Amplifiers biased with 18μA

5 bit converter

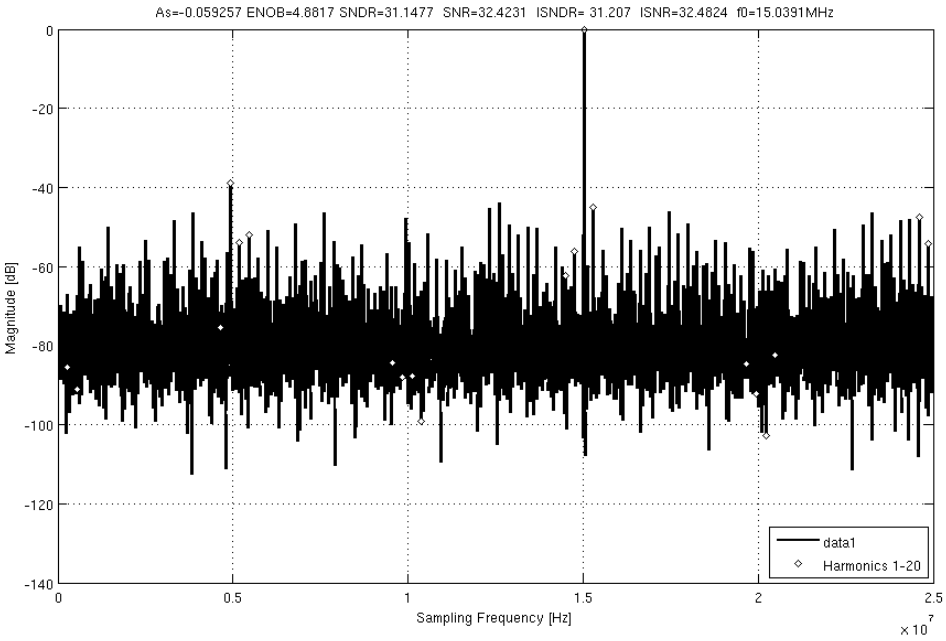


Figure 4.6: FFT plot showing the dynamic performance parameters of the 5 bit converter. No mismatch.

No of run	1	2	3	4	5
ENOB (bits)	4.82	4.88	4.70	4.86	4.77
No of run	6	7	8	9	10
ENOB (bits)	4.63	4.69	4.77	**4.53**	4.83
No of run	11	12	13	14	15
ENOB (bits)	4.77	4.84	4.85	4.82	4.83

Table 4.4: ENOBs for the 5 bit Flash ADC. 15 runs Monte Carlo simulation. Mismatch only. Amplifiers biased with 18μA

4.1.3 Power simulation

The power simulations were done with the same setup as for the 256 points frequency simulations. The current flowing from the V_{DD} node into each block was recorded and averaged using the Calculator tool in Cadence to get the data listed in the tables in this section.

3 bit converter

Block	Current consumption
Interpolation stage	100.4μA
Comparators	70.3μA
M-to-1	1.2μA
FAT-tree	2.5μA
Whole converter	174.3μA

Table 4.5: Power simulations of the 3 bit converter. Input is a full range 15.0381MHz sine. Sampling frequency 49.9968MS/s

4 bit converter

Block	Current consumption
1. interpolation stage	100.6μA
2. interpolation stage	180.5μA
Comparators	110.7μA
M-to-1	1.5μA
FAT-tree	4.7μA
Whole converter	398μA

Table 4.6: Power simulations of the 4 bit converter. Input is a full range 15.0381MHz sine. Sampling frequency 49.9968MS/s

5 bit converter

Block	Current consumption
1. interpolation stage	100.6μA
2. interpolation stage	180.8μA
3. interpolation stage	341.0μA
Comparators	206.6μA
M-to-1	1.9μA
FAT-tree	7.5μA
Whole converter	838.4μA

Table 4.7: Power simulations of the 5 bit converter. Input is a 15.0381MHz sine. Sampling frequency 49.9968MS/s

4.1.4 Input capacitance and required output resistance of the driving stage

The output of every Pipeline stage is an opamp. A large input capacitance for the back-end of the Pipeline ADC will have consequences for the last Pipeline stage since the driver (opamp) will have a larger g_m by:

$$R_{out} = \frac{1}{gm, min} \tag{4.6}$$

Where R_{out} is the output resistance and $g_{m,min}$ the minimum transconductance of the output driver of the last Pipeline stage. Further more we have that $\tau = R_{out}C_{in}$, where C_{in} is the input capacitance of the back-end Flash ADC. Which gives:

$$R_{out} = \frac{2ns}{C_{in}} \tag{4.7}$$

This result has been derived in the appendix.

Equation 3.5 for the cut-off frequency gives the input capacitance. A 10kΩ test-resistor was connected to the inputs and an AC-simulation was run with a AC signal with 0.5V amplitude and 0,9 V offset like a full-range input signal. The first simulation gave a relatively large input capacitance. This had to be investigated so the setup is simulated with different transistor-sizes of the input switches. The results are listed in table 4.8. In the bottom row the input switches has been removed. The plots are listed in the appendix as figure 8, 9, 10 and 11.

Input switch resistance	$R_{out,test}$	f_{3db}	C_{in}	R_{out}
412Ω	10kΩ	40.83MHz	551fF	3630Ω
2185Ω	10kΩ	70.94MHz	317fF	6309Ω
3274Ω	10kΩ	84.69MHz	265fF	7547Ω
-	10KΩ	152.00Mhz	148fF	13.5kΩ

Table 4.8: Measurement of the 3dB frequency with a 10kΩ resistor connected to the input to simulate the output resistance of the previous driver stage. The equivalent maximum output resistance was calculated using equation 4.7.

4.1.5 Gain of the Capacitive Interpolation block

This subsection includes a table with the gain of the amplifier stages relative to the converter input. The 3-bit converter is the same as one stage, the 4-bit the same as two stages and the 5-bit the same as 3 stages. The input is a 4 μs ramp. The t in the table below stands for the time where the gain is extracted. The input differential voltage at time t is therefore calculated as $V_{diff} = 1 - (0.5 * t)$ since the max amplitude of the differential input voltage is 1 volt.

Stage number	t	Δ V	Gain A
1	1.92μs	128mV	3.2
2	1.92μs	265mV	6.6
3	1.92μs	386mV	9.7

Table 4.9: Gain of the different stages of the different stages of the Capacitive Interpolation block. The 3-bit converter has 1 stage, the 4-bit has 2 stages and the 5-bit has 3 stages. The gain is aken out at time 1.92μs when the input differential voltage is 40mV.

4.2 Simulations of converters with minimized power consumption

In these simulations the amplifier bias current has been reduced to 12μA. This has been done to see how the resolution of the converters are affected by pushing the amplifiers closer to the unity gain frequency. The performance of the 3 and 4-bit converters were much better than both linearity demands and ENOB specifications so these simulations will be a better measure of these converters. $R_D = 90k\Omega$ and $R_S = 30k\Omega$ while the transistor sizes are the same as for the previous setup. A plot of the performance for these amplifiers is in figure 3 in the appendix.

4.2.1 INL/DNL

3 bit converter

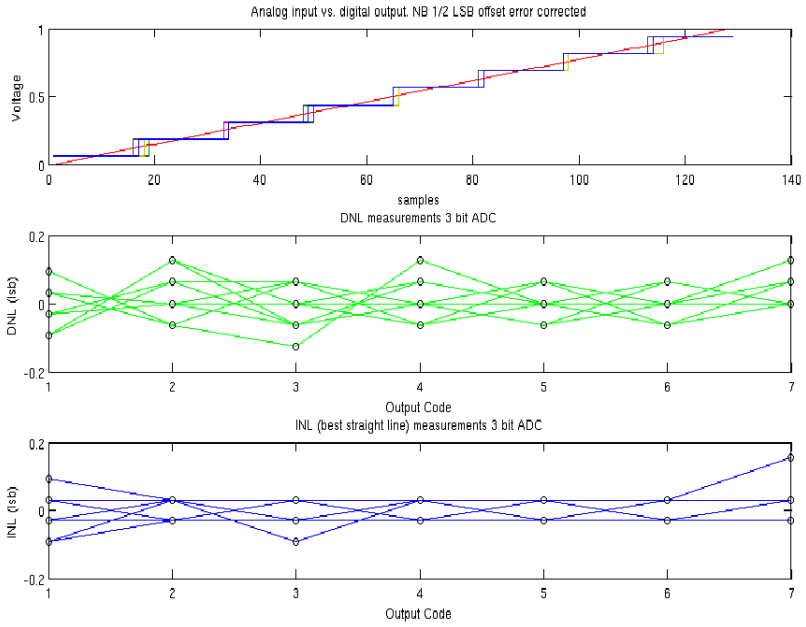


Figure 4.7: DNL and INL of the 3 bit converter. Amplifiers biased with 12μA

4 bit converter

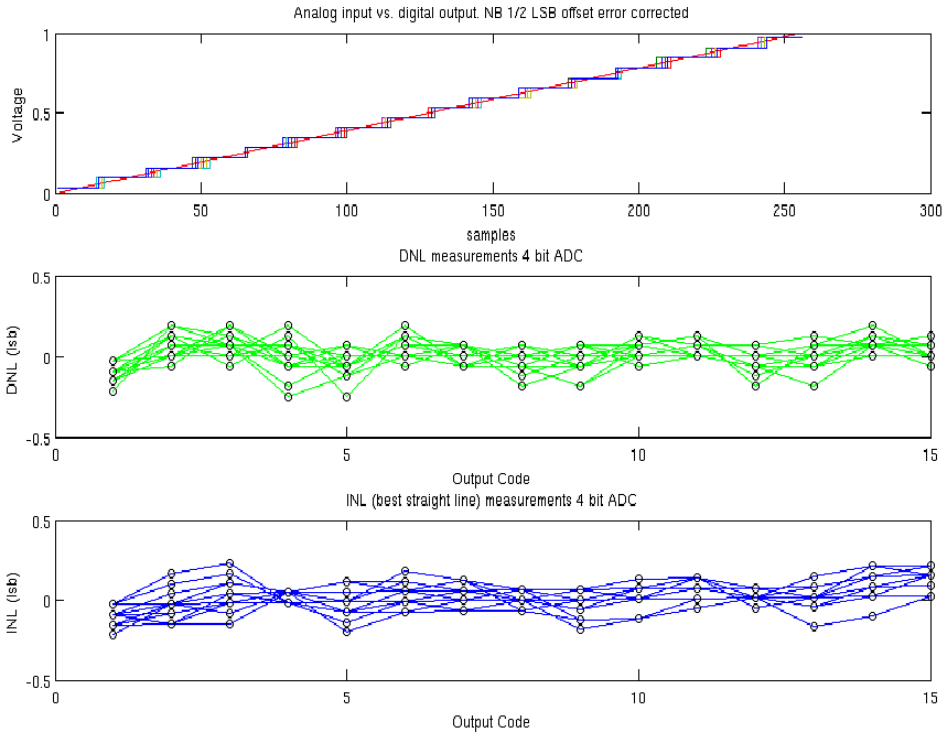


Figure 4.8: DNL and INL of the 4 bit converter. Amplifiers biased with 12μA

5 bit converter

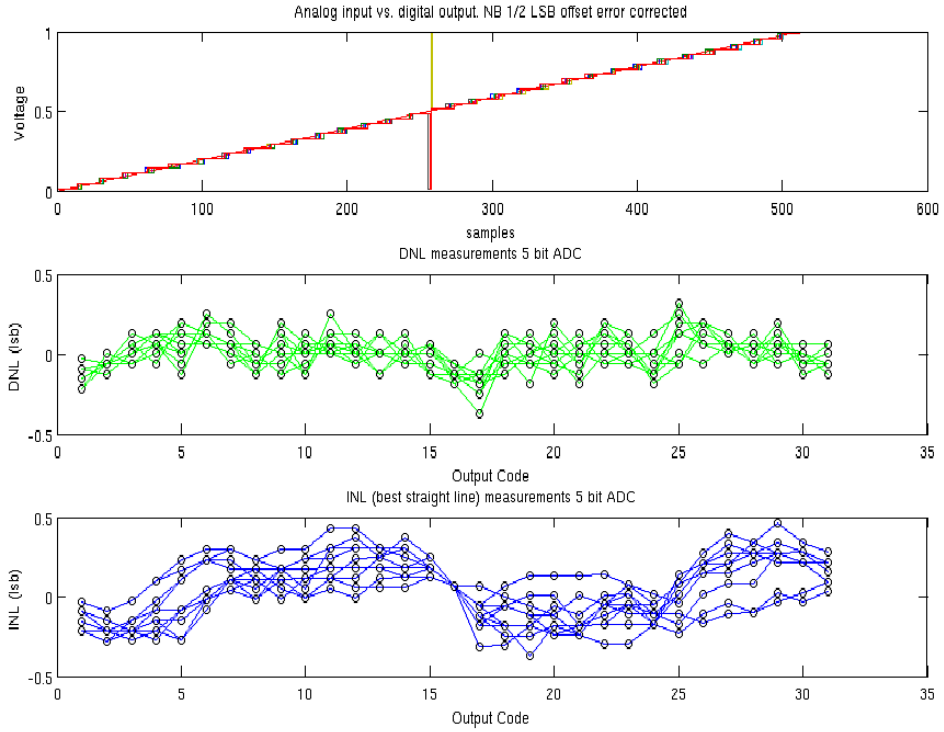


Figure 4.9: DNL and INL of the 5 bit converter. Amplifiers biased with 12μA

4.2.2 Frequency analysis with dynamic parameters

3 bit converter

No of run	1	2	3	4	5
ENOB (bits)	2.90	2.88	**2.84**	2.88	2.88
No of run	6	7	8	9	10
ENOB (bits)	2.88	2.88	2.86	2.87	2.86
No of run	11	12	13	14	15
ENOB (bits)	2.88	2.88	2.86	2.90	2.88

Table 4.10: ENOBs for the 3 bit Flash ADC. 15 runs Monte Carlo simulation. Mismatch only. Amplifiers biased with 12μA

4 bit converter

No of run	1	2	3	4	5
ENOB (bits)	3.91	3.93	3.91	3.95	3.94
No of run	6	7	8	9	10
ENOB (bits)	3.81	3.90	**3.81**	3.96	3.93
No of run	11	12	13	14	15
ENOB (bits)	3.89	3.96	3.93	3.91	3.92

Table 4.11: ENOBs for the 5 bit Flash ADC. 15 runs Monte Carlo simulation. Mismatch only. Amplifiers biased with 12μA

5 bit converter

No of run	1	2	3	4	5
ENOB (bits)	4.63	4.53	4.64	4.67	4.67
No of run	6	7	8	9	10
ENOB (bits)	4.71	4.62	**4.49**	4.55	4.67
No of run	11	12	13	14	15
ENOB (bits)	4.63	4.62	4.65	4.61	4.61

Table 4.12: ENOBs for the 5 bit Flash ADC. 15 runs Monte Carlo simulation. Mismatch only. Amplifiers biased with 12μA

4.2.3 Power simulation

3 bit converter

Block	Current consumption
Interpolation stage	65.1μA
Comparators	70.4μA
M-to-1	1.2μA
FAT-tree	2.5μA
Whole converter	146.8μA

Table 4.13: Power simulations of the 3 bit converter. Input is a full range 15.0381MHz sine. Sampling frequency 49.9968MS/s

4 bit converter

Block	Current consumption
1. interpolation stage	65.0μA
2. interpolation stage	117.0μA
Comparators	112.0μA
M-to-1	1.5μA
FAT-tree	4.7μA
Whole converter	316.9μA

Table 4.14: Power simulations of the 4 bit converter. Input is a full range 15.0381MHz sine. Sampling frequency 49.9968MS/s

5 bit converter

Block	Current consumption
1. interpolation stage	65.0μA
2. interpolation stage	117.0μA
3. interpolation stage	221.0μA
Comparators	209.0μA
M-to-1	1.9μA
FAT-tree	7.5μA
Whole converter	655.00μA

Table 4.15: Power simulations of the 5 bit converter. Input is a 15.0381MHz sine. Sampling frequency 49.9968MS/s

4.2.4 Gain of the Capacitive Interpolation block

This subsection includes a table with the gain of the amplifier stages relative to the converter input. The 3-bit converter is the same as one stage, the 4-bit the same as two stages and the 5-bit the same as 3 stages. The input is a 4 μs ramp. The t in the table below stands for the time where the gain is extracted. The input differential voltage at time t is therefore calculated as $V_{diff} = 1 - (0.5 * t)$ since the max amplitude of the differential input voltage is 1 volt.

Stage number	t	Δ V	Gain A
1	1.86μs	152mV	2.2
2	1.86μs	225mV	3.2
3	1.86μs	328mV	4.7

Table 4.16: Gain of the different stages of the different stages of the Capacitive Interpolation block. The 3-bit converter has 1 stage, the 4-bit has 2 stages and the 5-bit has 3 stages. The gain is taken out at time 1.86μs when the input differential voltage is 70mV.

4.2.5 Input capacitance and required output resistance of the driving stage

See the results of the previous section. Results are similar.

5

Remaining Work

This chapter lists the remaining work of the converters as well as suggestions on what may be the best way of dealing with these remaining issues. The remaining work can be divided into three categories:

- Minor work that has not been the focus of the design process because of limited time. The simulation process of these converters is time consuming, and it is of most importance not to add, subtract or alter any circuitry or parameters in the middle of a simulation. Therefore small adjustments can take a lot of time, so these parts which are considered minor just didn't make it to the final cut.
- The layout process which wasn't the scope of this assignment to begin with.
- During the work on this thesis an alternative approach involving changing the Flash ADC proposed to a two-step Subranging converter was planned, but never realized because off the limited time frame.

5.1 Clock circuitry

All the clocking circuitry of the converter has been realized as an ideal *stimulus* (name of the feature) when simulating in Cadence/Spectre. Because this is a subconverter in a higher resolution Pipeline ADC, it does not need any clock driver of it's own since the clocking is routed from the master clock. Any off-sample time clocking like *clkamp* and *complatch* (see the clocking section of the Design chapter) will have to be a shaped from this master clock. One way of doing this is to use digital delay circuitry [5, ch 10.1] to assure correct and non-overlapping clocks, this will however not be addressed here any further.

5.2 Kick-back from the comparators

The comparators produces a substantial amount of kick-back charge when going into latch. During transfer simulations of the converter the phenomena and its effect on the preceding stage was isolated by using ideal voltage controlled voltage sources with unity gain between the last capacitor-stage and the comparators. This way no charge escape back to the preceding stage. When these ideal devices are removed, there are some methods of minimizing either the kick-back charge or the effect of the kick-back charge. The most straight forward approach is to use buffers to isolate the charge . However, this is not a good approach since the buffers would be consuming a substantial amount of power since there has to be one buffer per comparator. In [14] some existing kickback reduction techniques are reviewed and new ones are proposed. One of the techniques suggested that suits dynamic comparators involves placing transmission gates before the input. This is a widely used technique (see e.g [8]). These switches are opened during the latch/regeneration phase, thus separating the comparator inputs from the preceding stage. A negative aspect of this technique is mentioned to be an increase in the offsets of the comparators due to mismatch charge injection from the switches. Using pMOS-switches and since the gain preceding the comparators are already minimizing the offsets of the comparators, this should therefore have minimal impact.

5.3 Layout

The layout is one of the most critical and comprehensive parts of the design process. E.g. parasitic and stray capacitances and noise from switching circuitry can have unforeseen consequences on the final performance of the converter. Since the Flash ADC is to be integrated in a larger converter, any layout work on the the converters is made difficult without having extensive knowledge on the Pipeline ADC and its layout.

5.4 The Subranging Converter

An alternative to the Flash/Interpolation (one-step) ADC is to use a multi-step architecture. A multi-step architecture converts the analog signal to binary in two or more clock cycles. This has many advantages as explained below. The Pipeline ADC is virtually a multistep architecture, but each Pipeline stage is usually converting just 1 bit. The motivation for a multi-bit multi-step ADC as the back end of the Pipeline is that if you can convert 5 bits in one clock cycle, why not convert $3 + 3 = 6$ bits in two clock cycles if the total power consumption of the Pipeline can be reduced and the resolution is good enough. An exciting prolonging of this study may therefore

be to look at possibilities for turning the Capacitive Interpolation ADC proposed in this work to an two-step architecture.

An appropriate approach for this is subranging. The basic subranging architecture using resistors is shown in figure 5.1.

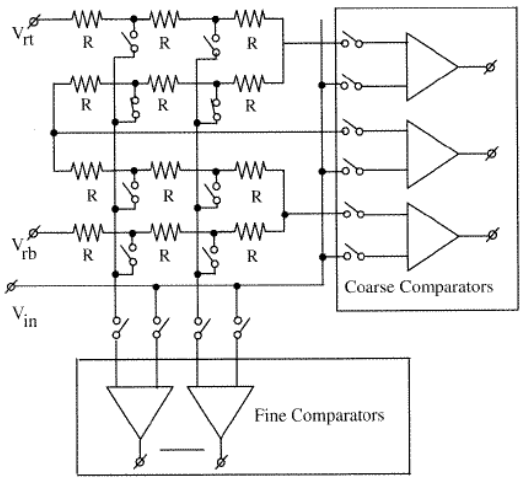


Figure 5.1: Subranging ADC [17, fig 3.32 s 149]

The basic building blocks for the converter are a reference ladder, coarse comparators and fine comparators. The reference ladder has $2^N - 1$ resistors. The course comparators are connected to 8 of the reference ladder taps (if this is an 6 bit converter), and then there are 7 resistors between each tap. First the 4 MSB's are decided. Then the MSB's are latched and at the same time a switching device decides which part of the reference ladder is going to be connected to the fine comparators for deciding the LSB's. In this architecture, no DAC is needed to create the LSB's as opposed to the two-step ADC.[17].

A proposed method for mixing the Subranging and Capacitive Interpolation techniques is shown in figure 5.2

This figure is an alternated version of figure 3.1. The top row is the coarse stage which converts the top 3 MSBs and the bottom row is the fine stage which converts the 3 LSBs. The middle box is basically a sample and hold system to get the fine stage to convert the right outputs from the coarse stage amplifiers. The converter is working the following way: First the coarse stage converts the MSBs in the same manner as the Capacitive Interpolation converter described in this thesis. When the comparator goes into latch-mode, the output from the M-to-1 line selector makes sure only the line at the conversion point of the thermometer code is high. This output

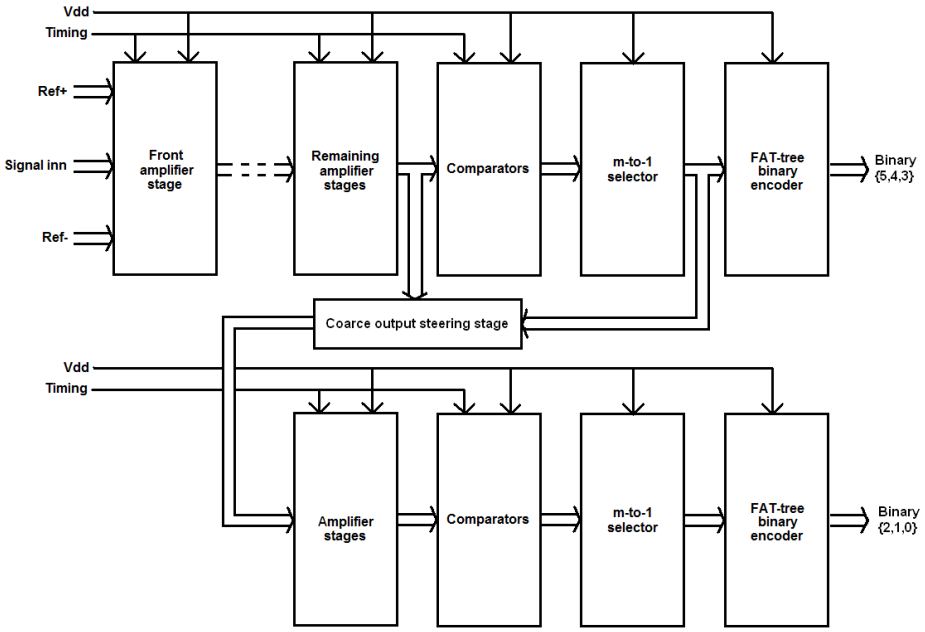


Figure 5.2: High level block diagram of the proposed subranging capacitive interpolation converter.

is then used as the control signal of the *Coarse output-steering stage*. The idea is that every output from the last amplifier stage is sampled and held, but only the amplifier outputs which the conversion point lies between is selected by the M-to-1 block and sent through. This is shown in figure 5.3

The Fine stage is the same capacitive interpolation type converter, so when looking from the input of the whole converter, through the amplifiers which have their outputs sent to the fine stage, and further into the fine stage a regular capacitive interpolation Flash ADC is basically seen! The most crucial parts of this converter will be the coarse stage and the sample and hold. The coarse stage has to have a gain close to 8 (but not over 8) to keep a maximum resolution on the inputs of the fine stage. The sample and hold will have to not add or subtract any information to the signal since this will deteriorate the resolution of the converter. the timing of this converter will also be a little more advanced than the basic Flash ADC since there are two stages that has to be synchronized. The main advantage of this converter is that it is basically two 3-bit converters, so the power consumption is reduced since less amplifiers and comparators are needed compared to a 6 bit and even a 5 bit Flash ADC.

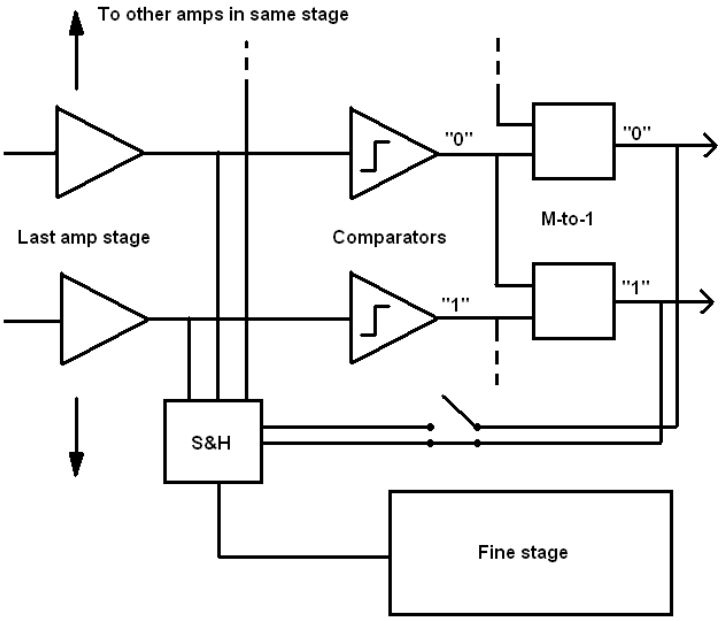


Figure 5.3: Diagram showing the suggested technique for sampling and switching the right coarse signal into the fine stage

6

Discussion

This chapter discusses the simulation results of this work.

6.1 DNL/INL analysis

These plots provides information regarding the accuracy of the converters. If the converters' INL error are staying within 0.5 LSBs and the DNL error within 1 LSB they are said to have no missing codes [5, chapter 11.5]. The maximum INL error occurs in the 5 bit converters where it is 0.47 LSBs for the high power converter and 0.46 LSBs for the low power converter. The DNL errors are all well within 1 LSB. Intuitively the maximum INL error of the lowest power converter should be higher than the higher power converter. The reason for this not happening may have two causes. First of all, the mismatch analysis had only 10 runs for the 5 bit converters. Such a little difference in maximum INL error may simply be because of chance since there is little probability that the actual maximum INL error was hit during these runs. Another reason for this to happen is because the linearity of the amplifiers that was scaled down wasn't the main contributor of the INL error. Some proof of this is found when looking at all the INL plots. For example: At sample 8, 16 and 24 of the 5 bit converters the INL error of the high power converter is at a minimum. These are the MSB and MSB-1 outputs and therefore the outputs that have gone through the capacitive interpolation block without being interpolated other than at the input. Therefore it seems like the capacitor (mis)matching is the main reason for the INL error in the higher power converters. In the lower power converter(s) this is not so much the case. The amplifier mismatch has more influence on the INL error for these converters.

6.2 Frequency analysis

The most interesting parameter of the frequency plots is the ENOB since it is one of the main parameters given in the specification that is saying something about the required resolution of the converter. Not having enough time for any more simulations limited the number of runs for each Monte Carlo-analysis to 15. This means that only 15 ENOBs are calculated for each ADC setup. This introduces a little uncertainty. Tables 6.2 and 6.2 lists statistical properties of the ENOBs obtained during the mismatch runs. The 95% confidence interval K for the population mean is estimated using the Student's t distribution [19], since the sample size is small (and) so the populations standard deviation and mean is unknown.

Resolution	Sample mean	s	95% K
3 bit	2.880 bits	0.023 bits	<2.867, 2.893> bits
4 bit	3.893 bits	0.051 bits	<3.864, 3.921> bits
5 bit	4.773 bits	0.097 bits	<4.712, 4.827> bits

Table 6.1: ENOB statistical properties for the converters with the higher power (18μA amp bias) capacitive interpolation block. s is the estimated standard deviation. 95% confidence interval K for population mean estimated by using Student's t distribution.

Resolution	Sample mean	s	95% K
3 bit	2.875 bits	0.016 bits	<2.867, 2.884> bits
4 bit	3.911 bits	0.045 bits	<3.885, 3.936> bits
5 bit	4.620 bits	0.060 bits	<4.590, 4.650> bits

Table 6.2: ENOB statistical properties for the converters with the lower power (12μA amp bias) capacitive interpolation block. s is the estimated standard deviation. 95% confidence interval K for population mean estimated by using Student's t distribution.

These results show that the 15 run analysis has been a good estimator on how the full population is like. For the 5 bit Flash ADC the ENOB mean has a 95% chance of lying within 0.1 bits, and it is even lower for the lower resolution converters. All the converters also has a standard deviation s $<$ 0.1 bits. The spread is therefore low, which is promising.

For the higher power converters, all the ENOBs of the 3-, and 4-bit converters lies above respectably 2.75 and 3.75 bits, but the 5-bit has 4 runs below 4.75-bits and has a mean lying between 4.71 and 4.83 with a 95%

probability. This means that the final ENOB specification for the 3-, and 4-bit converters are above the initial specification, but the 5-bit final ENOB specification is a bit below the initial specification since roughly 40% will lie below depending on where the population mean will be.

The lower power converters are designed to fit the power consumption of the 3- and 4-bit better than the 5-bit since they had some headroom regarding the static and dynamic performance parameters for the higher power converters. As can be seen the 3-, and 4-bit converters are still very good, suggesting that the amplifier biasing could be even lower for these converters. The 5-bit converters ENOB has dropped well below 4.75 bits.

6.3 Power Consumption

The table below lists the Flash ADCs power consumption as opposed to the Pipeline ADC. For the 4-, and 3-bit Flash ADCs all numbers include respectively the Nth and the Nth + (N-1)th ASD Pipeline stage. The power is found by multiplicating the current consumption (I_{total}) found with V_{DD}. Note that the power of the correction logic is included in the three rightmost columns. The last but one column is the ASD Pipeline ADC specs. with the ASD 3-bit Flash ADC.

Res.	Amp bias	I_{tot}	P_{tot}	$P_{tot+digcor}$	P_{pipe}	Diff %
3 bit	18μA	974μA	1.75mW	2.74mW	2.76mW	0.7%
4 bit	18μA	818μA	1.47mW	2.27mW	2.76mW	17.8%
5 bit	18μA	838μA	1.51mW	2.14mW	2.76mW	22.5%
3 bit	12μA	947μA	1.70mW	2.69mW	2.76mW	2.5%
4 bit	12μA	737μA	1.33mW	2.13mW	2.76mW	22.8%
5 bit	12μA	655μA	1.18mW	1.81mW	2.76mW	34.4%

Table 6.3: Table showing the total power consumption of the converters. Rs. is the resolution of the backend Flash ADC. The rightmost column is the power savings in %.

This table shows that the power savings can be substantial. Note that the ENOB of the 5-bit converter biased with 12μA was lower than the initial specifications. For the 3- and 4-bit converters the numbers include the corresponding Pipeline Stages, this means that for the numbers to be correct the input capacitance of the Flash ADC-converters has to be 182fF for the N'th Pipeline stage. The input capacitance is discussed in the next section.

6.4 Input Capacitance

The results of the input capacitance simulations (table 4.1.4) shows that the input capacitance is strongly dependent on the size of the input switches. The switches were in the first place chosen because of that they provide very little charge injection in the signal path. The width and length of the first switches tried out was 4μm by 180nm, with dummies having half the width. This gave a low on-resistance of 412Ω and an input capacitance of 551fF. The smallest input switch tried out was 500nm by 180nm which gave an on-resistance of 3274Ω and an input capacitance of 265fF. The input switches was also removed. The converter won't work when this is done, but it gives the opportunity to measure the input capacitance just being the input capacitors in series with the gate-capacitances of the first amplifier stage. The capacitance was measured to be 148fF.

These simulations show that the capacitances of the input switches will deteriorate the input capacitance of the Flash ADC extensively, so careful choosing of the right low-capacitance switches is a necessity for the driver of the backend not to consume a lot of power (as referred to the previous chapter). An alternative is to have an interpolation factor of 2 in the first stage of the Flash ADC. This means that tree amplifiers are in parallel instead of five as in the converters simulated. This will reduce the input capacitance at the expense of one more interpolation stage. This means that about 40μA is added to the current consumption of the Flash ADC in the 18μA amplifier bias version of the converter. If the switch problem isn't solved this will still be much less power added than to have a high power driving stage.

6.5 Gain of the Capacitive Interpolation Block

Optimally, the gain of cascading amplifier stages should multiplicate. As seen in tables 4.1.5 and 4.2.4 this is not the case in both the versions of the converters. This has a few explanations. First of all, the settling of the amplifiers of the later stages are sub-optimal as seen in figure 12 in the appendix. This can be explained by that there exists a gradual latency from stage to stage as can be seen in the figure.

The comparators of the Flash ADCs used in this work was deliberately having much better performance than needed. This was done to exclude them as a performance-degrading factor. When looking at the power consumption of the Flash ADCs isolated (tables 4.5, 4.6 and 4.7, and 4.13, 4.15 and 4.16), it can be seen that the comparators are using quite a large percentage of the power consumption. This is listed in table 6.4.

Resolution	Amplifier bias	Comparator power %
3 bit	18μA	39.2%
4 bit	18μA	27.8%
5 bit	18μA	24.6%
3 bit	12μA	47.8%
4 bit	12μA	35.4%
5 bit	12μA	31.9%

Table 6.4: The percentage of the comparator power consumption opposed to the respective Flash ADC. Not counting in Pipeline stages etc.

Introducing the gain of the preamplifiers of the interpolation block, this power can be substantially reduced. This decrease in power is limited by the input referred offset as described in the Design-chapter, but it is also limited by the sum of the delay of the comparator and the digital logic since the output bit has to propagate to the outputs of the converter in half a clock cycle. In the 5 bit case with a gain of 9.7, the input referred offset is 9.7 times smaller than the actual offset of the converters. Thus, according to equation 3.2 the area of the differential pair input stage of the comparator can be substantially reduced to reduce the switching current.

7

Conclusion

A table of the performance parameters are listed in table 7. The numbers are based on the Flash ADC backends only. The amp bias column refers to the capacitive interpolation amplifiers being biased with either 18μA or 12μA. The Figure of Merit (FoM) is calculated using equation 7.1

$$FoM = \frac{Power}{2^{ENOB} * f_s} \tag{7.1}$$

Parameter	18μA amp bias			12μA amp bias		
Res. (bits)	3	4	5	3	4	5
DNL (LSB)	<0.2	<0.3	<0.5	<0.2	<0.3	<0.5
INL (LSB)	<0.1	<0.3	<0.5	<0.2	<0.3	<0.5
ENOB* (bits)	2.88(2.82)	3.89(3.77)	4.77(4.53)	2.88(2.84)	3.91(3.81)	4.62(4.49)
FoM (pJ/conv.)	0.85	0.97	1.11	0.72	0.76	0.96

Table 7.1: Final specifications. * Mean of ENOB, worst run in parenthesis.

Table 6.3 in the Discussion chapter shows that the resolution of the back-end of the Pipeline converter can be extended up to at least 5 bits while using the Capacitive Interpolation technique in terms of power consumption. The power can be lowered even further since some gain is introduced in the Capacitive Interpolation block. However, this depends somewhat on the input capacitance of the Flash ADC. The input capacitance simulations showed that this capacitance is highly dependable on the sizes of the input switches. Small devices with low capacitance must be used for the last Pipeline stage not to consume a lot of power. This will also be the case for the reference voltage drivers since they are connected the same way as the inputs.

In terms of the dynamic performance, the 4 bit converters show great performance and reduced power consumption. The 5-bit Flash ADC did not reach the ENOB requirement of 4.75 bits. Even though the mean was 4.77 bits, 4 of the 15 mismatch-runs on the 5-bit flash ADC was below the ENOB requirement. To get a higher score, the bias current of the amplifiers will need to be even higher than the estimated 18μA used in the simulations. Compared to the equivalent power consumption of the last 5 bits of the ASD Pipeline ADC the power consumption is already reduced by 24.6% (table 6.3), and a lot more power can be reduced since the gain of the amplifiers reduces the demands on the offset of the comparators so that the sizes of the input stage of the comparators and hence the switching current, can be reduced. Therefore it is safe to conclude that a 5-bit backend can replace the 3-bit backend currently implemented in the Pipeline ADC without raising the power consumption. The 4 bit converter show great performance and reduced power consumption.

To go higher than 5 bits of resolution the linearity of the converters will worsen and the increasing power consumption can make the Flash architecture a bad choice. The natural extension of this thesis will be to implement and simulate a 6 bit converter to see how much power will be spent on reducing the linearity/performance issues this converter will face. Another very interesting approach is the subranging converter that is outlined in the Remaining Work chapter. When moving up to 6 bits, the converter isn't only interesting as a backend ADC, but it also has various other applications as a stand-alone converter (see e.g [9]).

Bibliography

[1] Kenneth C. Smith Adel S. Sedra. *Microelectronic Circuits.* Oxford University Press, 2004.

[2] Walter Guggenbühl Christoph Eichenberger. Dummy transistor compensation of analog mos switches. *IEEE Journal of Solid-State Circuits*, 24, 1989.

[3] Andreas Santner Thomas Hartig Franz Kuttner Christoph Sandner, Martin Clara. A 6bit, 1.2gsps low-power flash-adc in 0.13μm digital cmos. *IEEE Journal of Solid-State Circuits*, 40(7), 2005.

[4] Kyusun Choi Jahan Ghaznavi Daegyu Lee, Jincheol Yoo. Fat tree encoder design for ultra-high speed flash a/d converters. *Circuits and Systems, 2002. MWSCAS-2002. The 2002 45th Midwest Symposium*, 2, 2002.

[5] Ken Martin David A Johns. *Analog Integrated Circuits.* Univ. of Toronto, 1997.

[6] Daniel D. Gajski. *Principles of Digital Design.* Prentice Hall, 1997.

[7] Dag Sverre Høye. *Flash ADC, project assignmen, fall 2007.* Department of electronics and telecommunications, 2007.

[8] Sangsuk Kim and Minkyu Song. An 8bit 200msps cmos a/d converter for analog interface module of tft-lcd driver. *IEEE*, 2001.

[9] Michiel S. J. Steyaert Koen Uyttenhove. A 1.8-v 6-bit 1.3-ghz flash adc in 0.25μm cmos. *IEEE Journal of Solid-State Circuits*, 38(7), 2003.

[10] Väinö Hakkarainen Kari Halonen Lauri Sumanen, Mikko Waltari. Cmos dynamic comparators for pipeline a/d converters. *Circuits and Systems, 2002 IEEE International Symposium*, 5, 2002.

[11] Willy Sansen Libin Yao, Michiel Steyaert. A 1.8-v 6-bit flash adc with rail-to-rail input range in 0.18μm cmos. *5th International Converence on ASIC*, 2003.

[12] Maxim. Coherent sampling vs. window sampling, application note 1040. http://www.maxim-ic.com/appnotes.cfm/appnotenumber/1040.

[13] Giuseppe Scotti Alessandro Trifiletti Pietro Monsurrò, Salvatore Pennisi. Linearization technique for source-degenerated cmos differential transconductors. *IEEE transactions on circuits and systems II*, 2007.

[14] Joãl C. Vital Redro M. Figueiredo. Kickback noise reduction techniques for cmos latched comparators. *IEEE transactions on circuits and systems*, 53, 2006.

[15] C. Morandi S. Padoan, A. Boni and F. Venturi. A novel encoding schemes for the rom of parallel adcs, featuring reduced conversion noise in the case of single bubbles in the thermometer code. *Electronics, Circuits and Systems, 1998 IEEE International Conference*, 2, 1998.

[16] Atle Briskemyr Frode Telstø Johnny Bjørnsen Thomas E. Bonnerud Øystein Moldsvor Terje Nortvedt Andersen, Bjørnar Hernes. A cost-efficient high-speed 12-bit pipeline adc in 0.18-μm digital cmos. *IEEE Journal of Solid-state Circuits*, 40(7), 2005.

[17] Rudy J. van de Plassche. *CMOS Integrated Analog-to-digital Converters.* 2003.

[18] Eric A. Vittoz. Low power design: Ways to approach the limits. *ISSCC*, 41, 1994.

[19] WolframMathWorld. Student's t-distribution. http://mathworld.wolfram.com/Studentst-Distribution.html.

List of Figures

Appendices

.1 Digital correction logic

In addition to the submodules listed in table 1.2, the Pipeline ADC has some digital correction logic. The current consumption of all this correction logic is 550μ. This current consumption is dominant by delay elements. The addition of one pipeline stage is propotional to the square of the added number of these delay elements. The Pipeline ADC has a total resolution of 13 bits and the backend Flash ADC of this converter is 3 bits. When there is a variable resolution of the Flash ADC the current consumption of this digital correction logic then becomes approximately:

$$I_{digcor} = 550\mu A * (\frac{13 - N_{Flash}}{10})^2 \tag{2}$$

Where N_{Flash} is the Flash ADC number of bits. The resulting current consumption is listet in the table below.

Resolution	Current consumption
3-bit	550μA
4-bit	446μA
5-bit	352μA

Table 2: Current consumption of digital correction logic of the Pipeline ADC with regard to the backend Flash ADC resolution

.2 Analysis of source degenerated differential pair.

This section includes calculations of the differential gain of the source degenerated differential pair.

.2.1 Differential gain of source degenerated differential pair.

This analysis is made a little simpler if the input is the common mode-voltage, so that half the bias current $\frac{i}{2}$ is running through each transistor, and the output voltages are equal. The small signal equivalent circuit is shown in figure 1

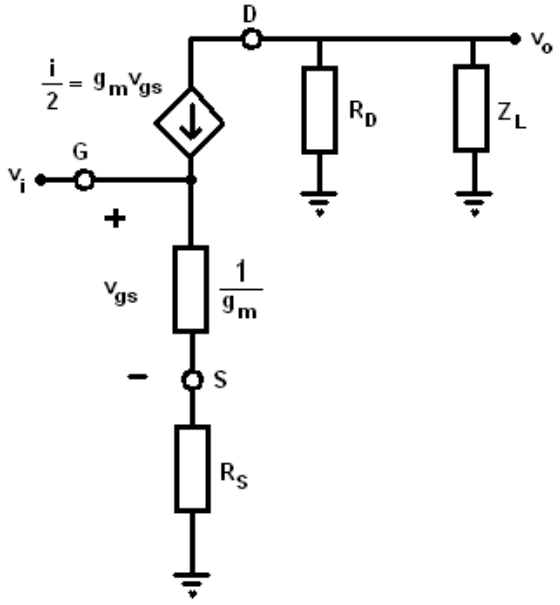


Figure 1: Small signal model of one side of the differential pair similar to source degenerated common source amplifier. Z_L is the load impedance

At the output we have:

$$v_o = -g_m v_{gs} R_D || Z_L \tag{3}$$

Where v_o is the output voltage, g_m the transconductance, v_{gs} the gate-source voltage, R_D the load resistance and Z_L any other load impedance (e.g the load capacitance). This is the same as for the basic differential pair. At the input however:

$$v_{gs} = \frac{\frac{1}{g_m}}{\frac{1}{g_m} + R_S} v_i = \frac{v_i}{1 + g_m R_S} \tag{4}$$

which put into 3, gives:

$$A_{SD} = \frac{v_o}{v_i} = -\frac{gm(R_D||Z_L)}{1 + g_m R_S} = -\frac{R_D||Z_L}{\frac{1}{g_m} + R_S} \tag{5}$$

.3 Deriving the right expressions for the input capacitance and driver output resistance

This section is an analysis of the restrictions on R_{out} of the output driver for the last Pipeline stage. An estimation of this parameter is important since a very small R_{out} means that the g_m of the driver will be large. Since g_m is proportional to the current consumption of the driver, we want g_m to be as small as possible.

The SNR of an ideal converter is expressed as:

$$SNR = 6.02N + 1.76 \tag{6}$$

For the 5-bit converter the equation gives $SNR_{5b} \approx 32dB$ which means that the settling of the input has to be better than $-32dB$. Adding some headroom to take into account the unidealities of the converter this settling is set to $-40dB$.

Further more we have that $\tau = R_{out}C_{in}$ where τ is the time constant and C_{in} is the Flash ADC input capacitance.The settling can be written:

$$e^{-M} = 0.01 = -40dB \tag{7}$$

Given that M = $\frac{t_{settle}}{\tau}$ we get

$$M = \frac{t_{settle}}{\tau} = 4.61 \approx 5 \Rightarrow t_{settle} = 5\tau \tag{8}$$

Rounding to 5 for ease. Since the sampling frequency of the converter is 50MS/s and the tracking time of the inputs are half a clock-cycle we get $t_{settle} = 10ns$ and $\tau = 2ns$. This means that the converter must settle to 63% of the new input sample within 2ns. For a fixed C_{in} this gives a minimum R_{out} (or maximum g_m).

$$R_{out} = \frac{2ns}{C_{in}} \tag{9}$$

.4 Plots from cadence

This section includes some of the plots from Cadence that had to be to large to be put directly in the text since they became very hard to interpret.

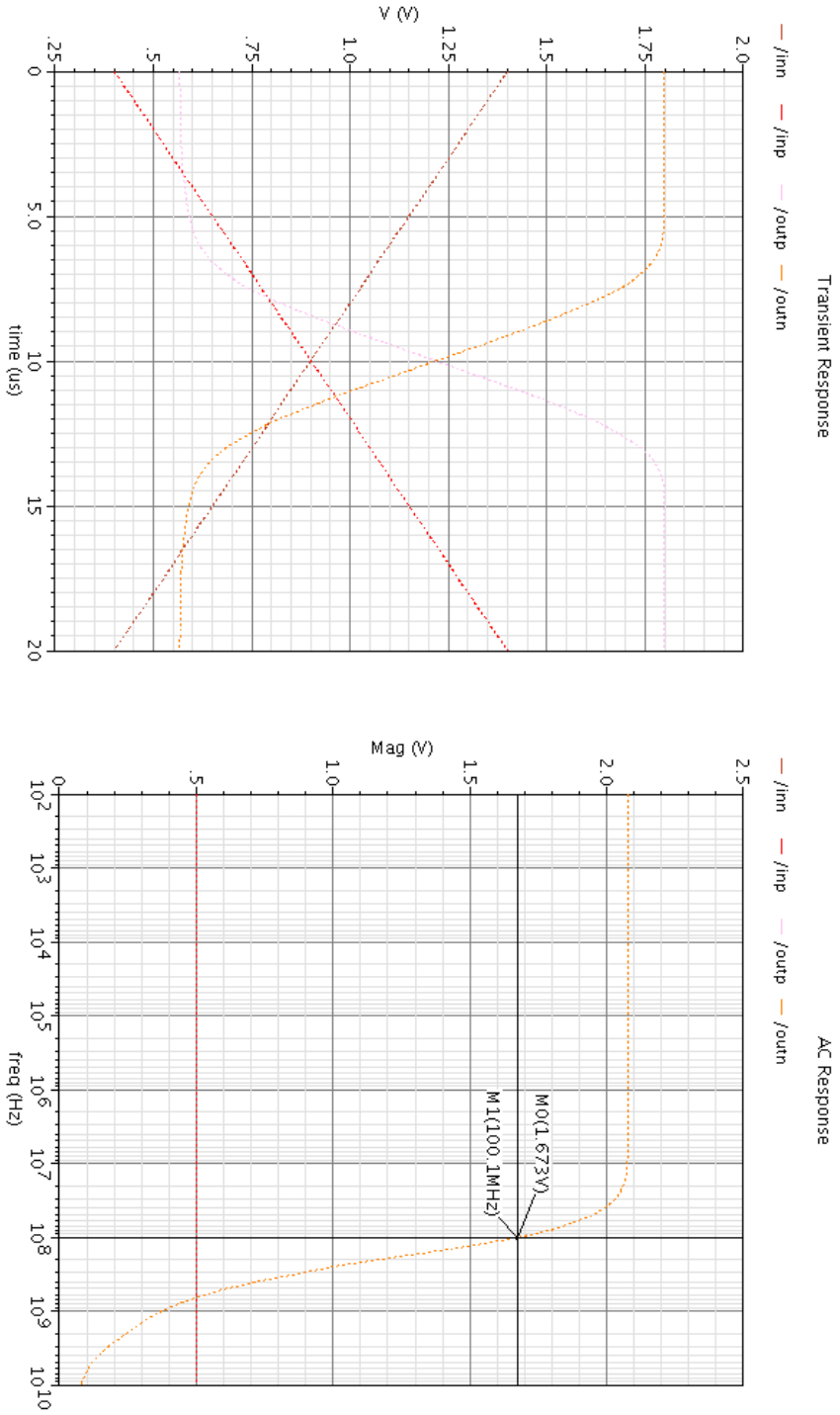


Figure 2: Simulations of a differential pair biased with 18μA. Transient analysis to the left and AC analysis to the right.

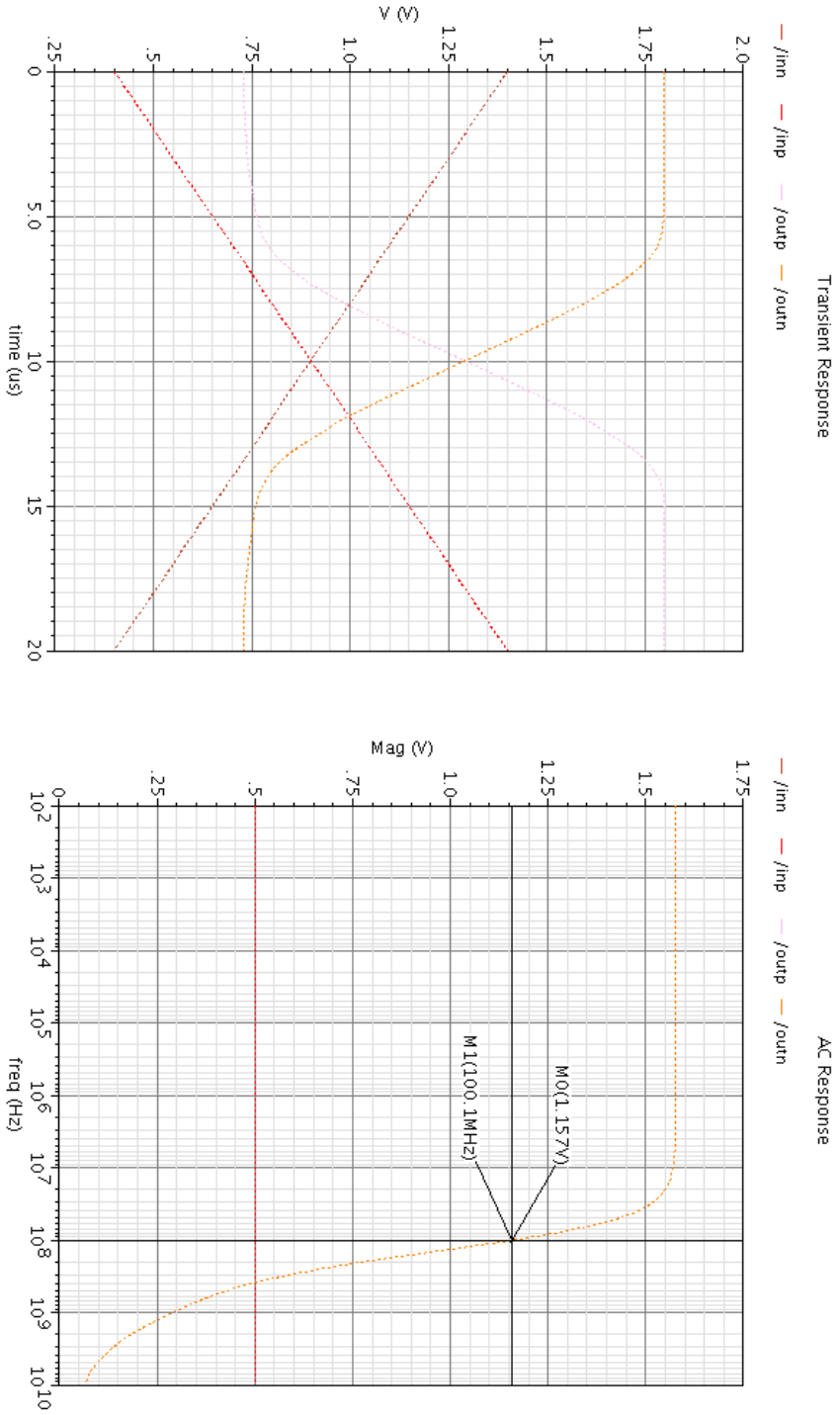


Figure 3: Simulations of a differential pair biased with 12μA. Transient analysis to the left and AC analysis to the right.

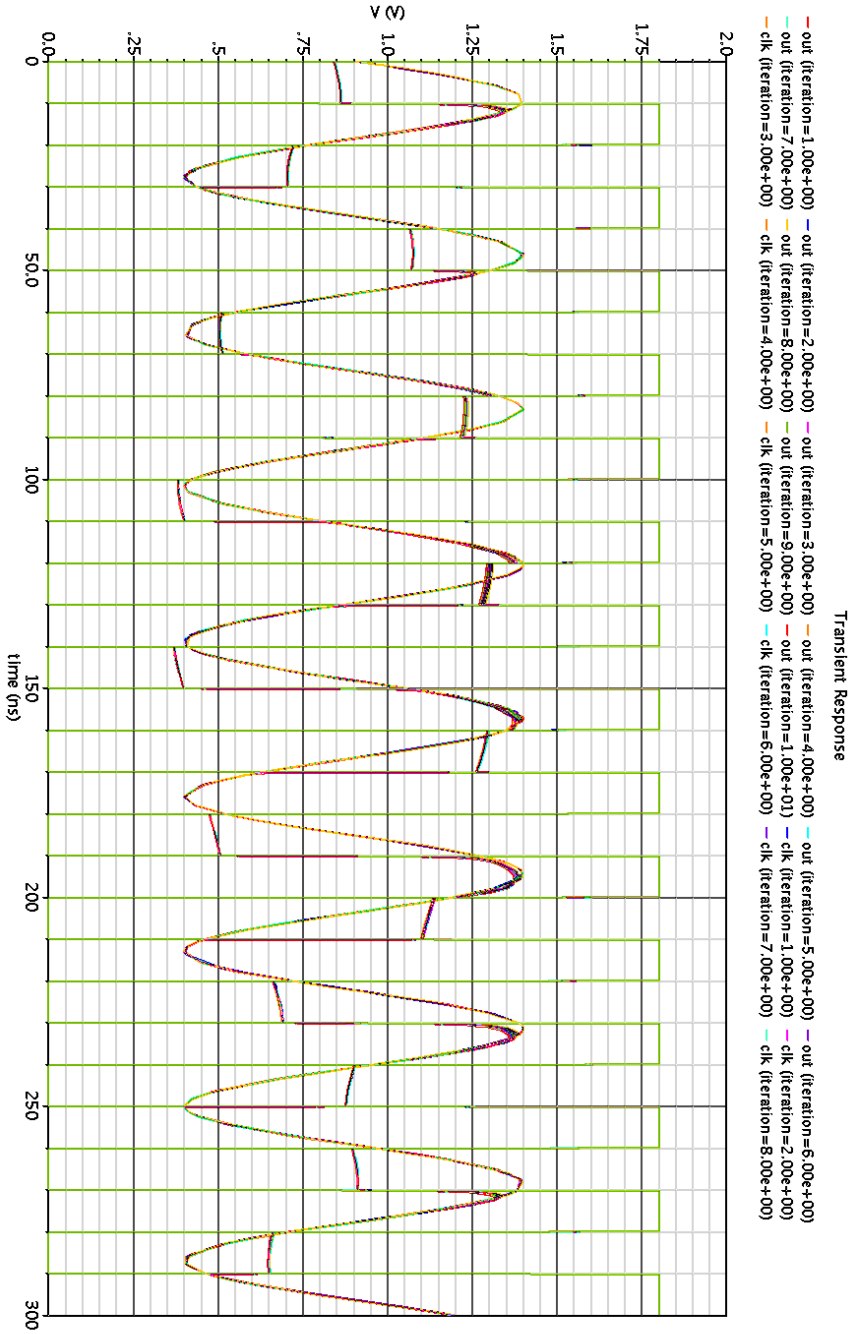


Figure 4: 15 run Monte Carlo transient analysis of regular CMOS transmission gate. The clock, input and output is plotted. Input is a 27MHz sine and clock sampling rate is 50MS

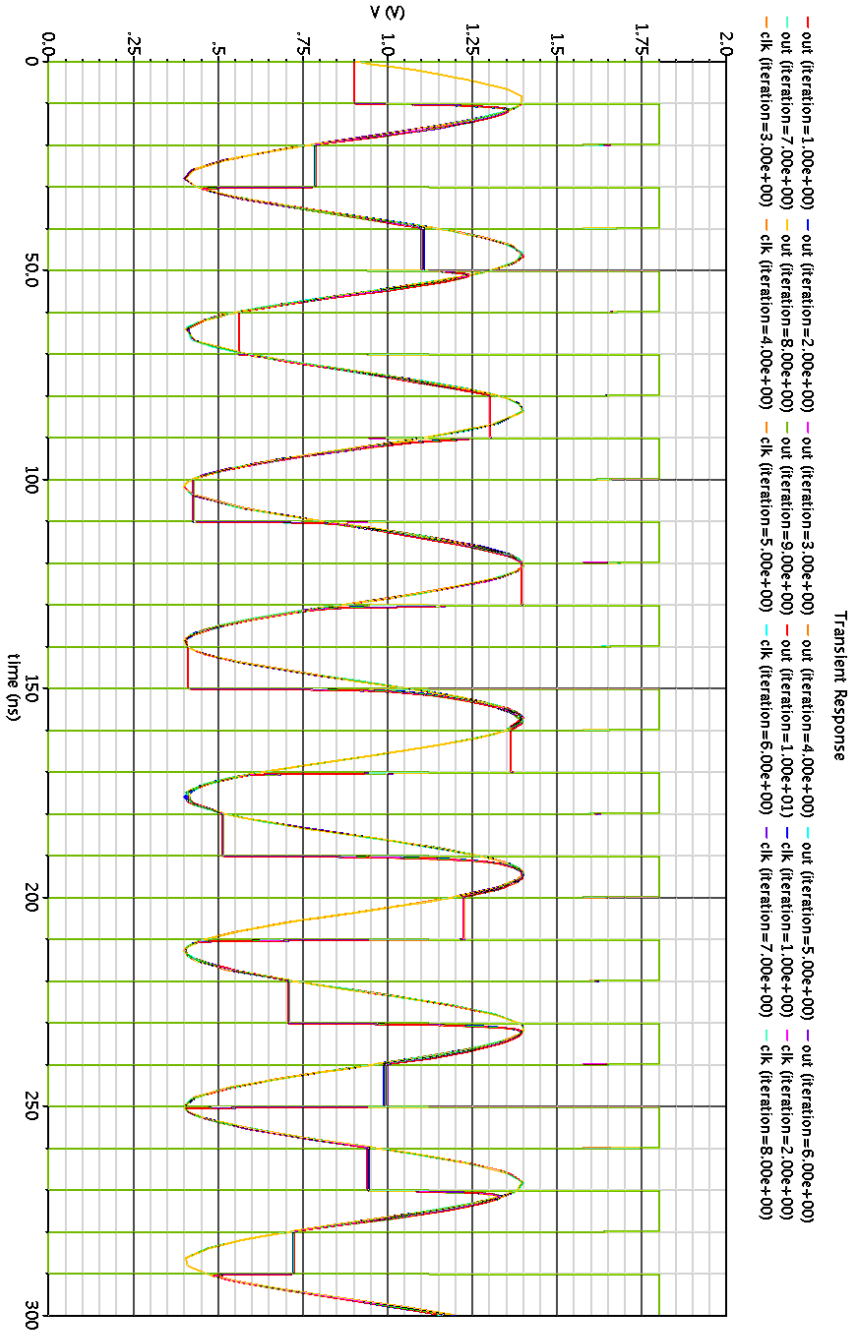


Figure 5: 15 run Monte Carlo transient analysis of CMOS transmission gate with dummy at input and output. The clock, input and output is plotted. Input is a 27MHz sine and clock sampling rate is 50MS

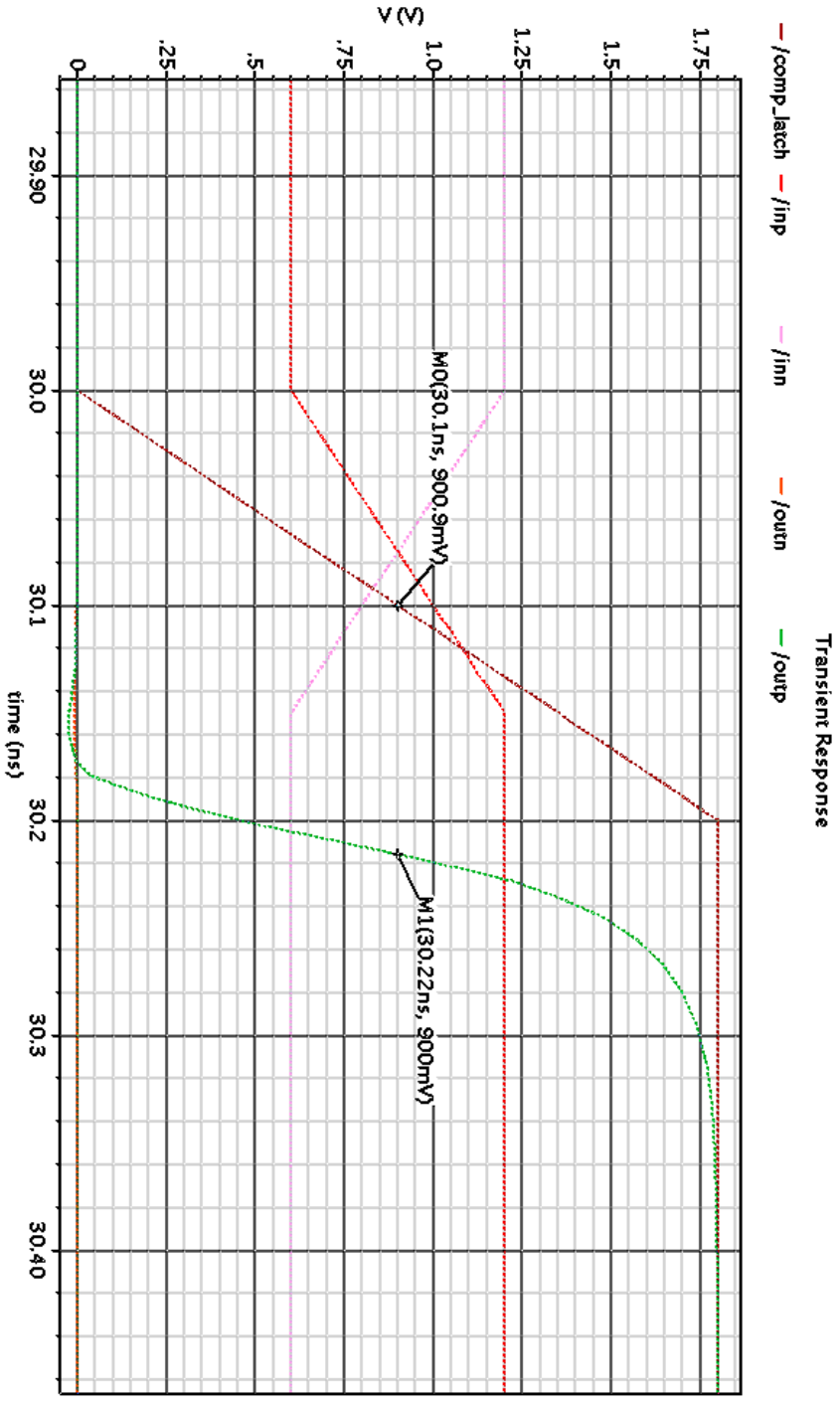


Figure 6: Comparator delay. Measured to be 120ps.

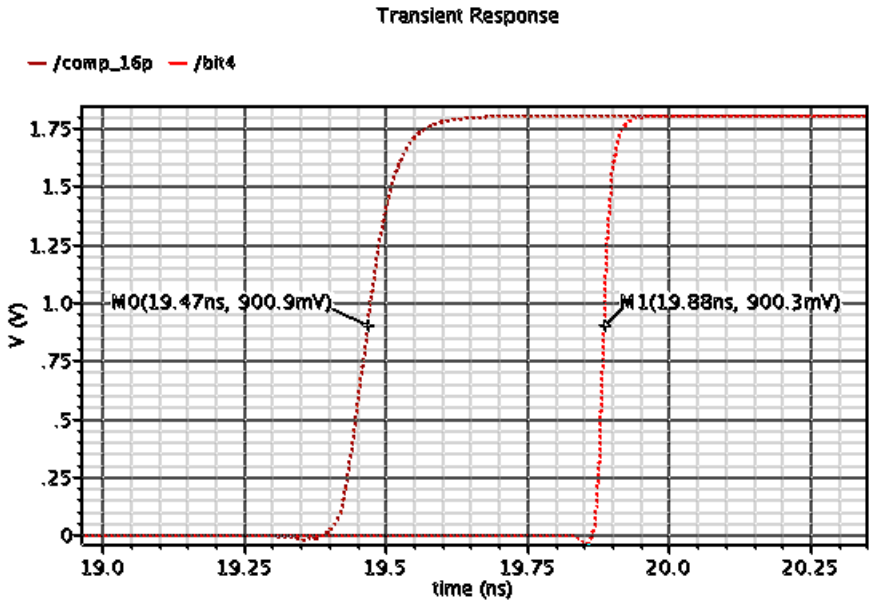


Figure 7: Rise delay of the digital circuitry

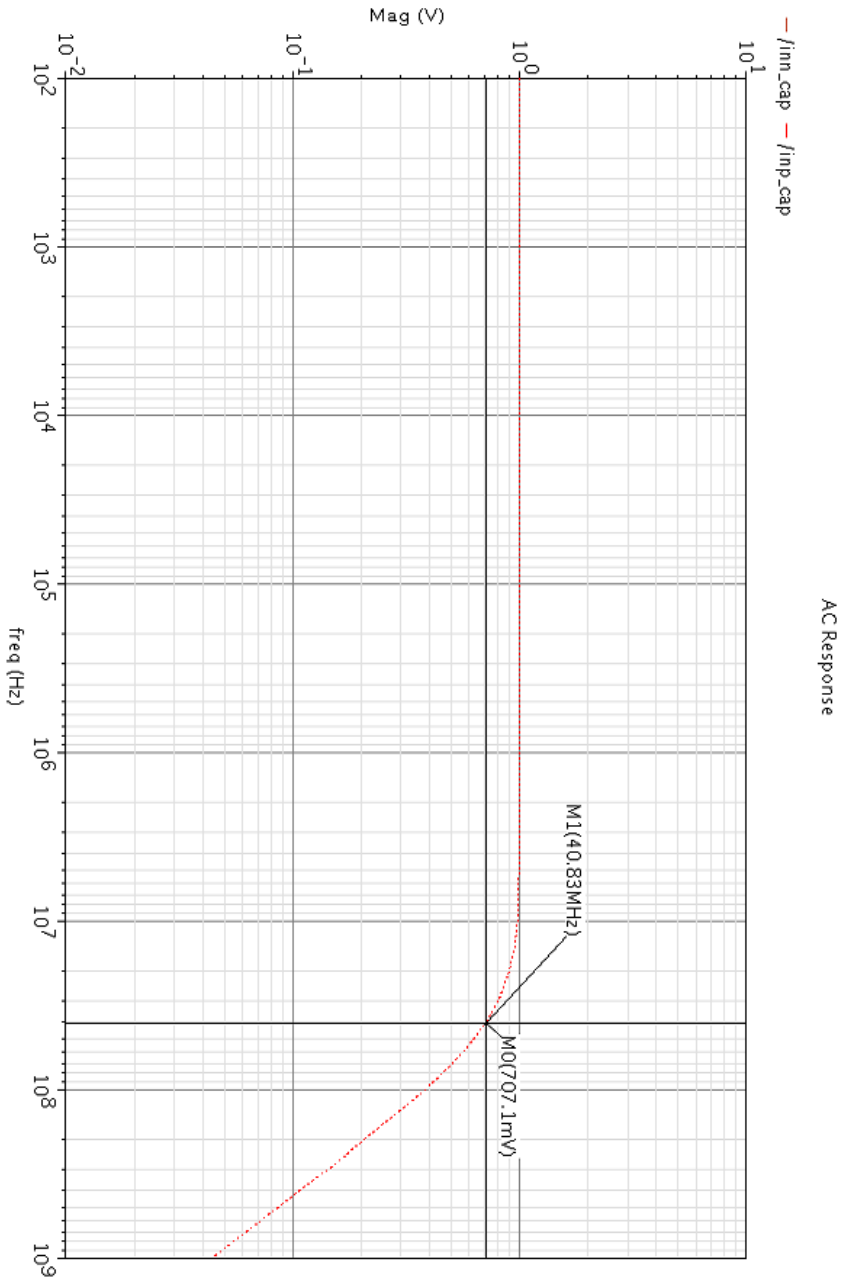


Figure 8: Measurement of the 3dB frequency of the input with a 10KΩ resistor connected to the input to simulate the output resistance of the previous driver stage. The 3dB frequency is at 40.83MHz which gives an input capacitance of 550fF. The on-resistance of the input-switches is 412Ω

The equivalent maximum resistance according to equation 4.7 becomes 3630Ω.

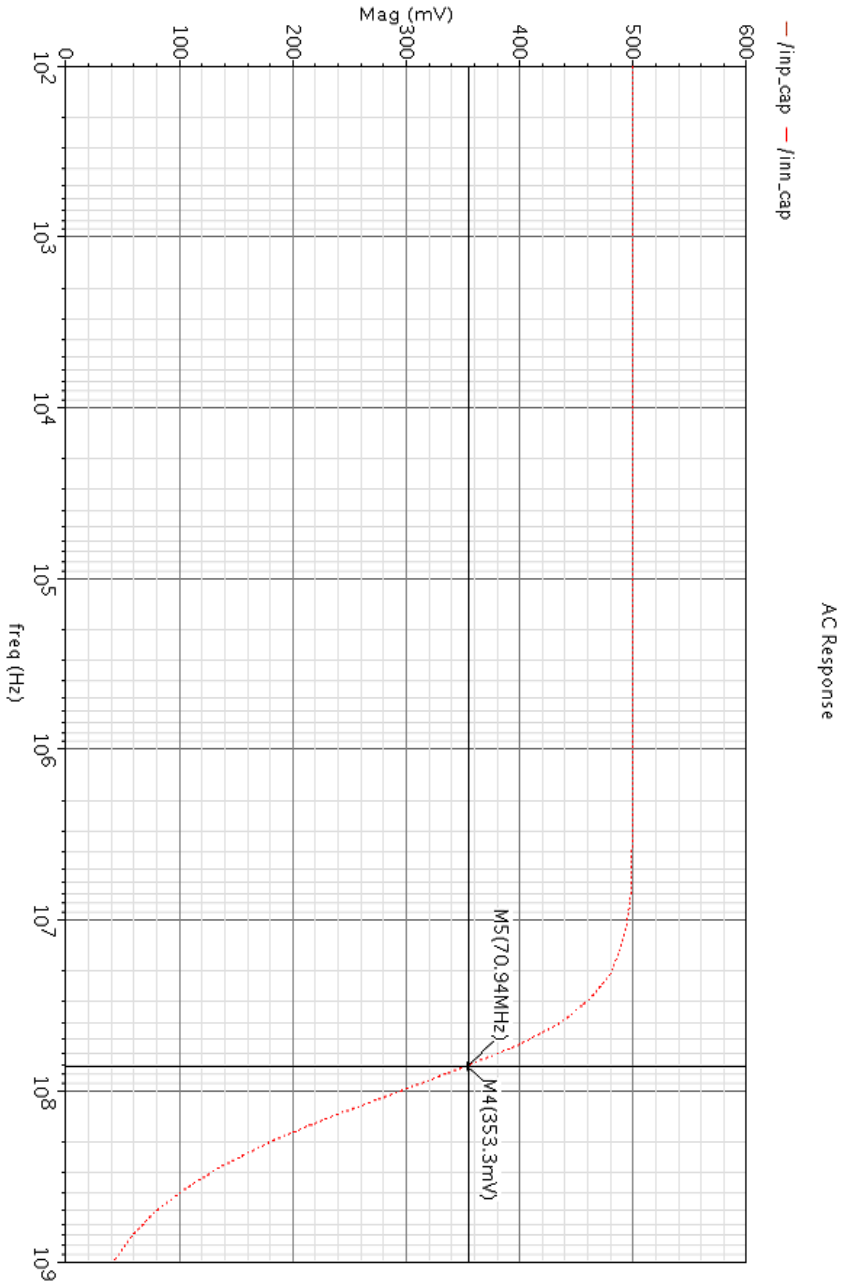


Figure 9: Measurement of the 3dB frequency of the input with a 10KΩ resistor connected to the input to simulate the output resistance of the previous driver stage. The 3dB frequency is at 70.94MHz which gives an input capacitance of 317fF. The on-resistance of the input-switches is 2185Ω . The equivalent maximum output resistance according to equation 4.7 becomes 6309Ω.

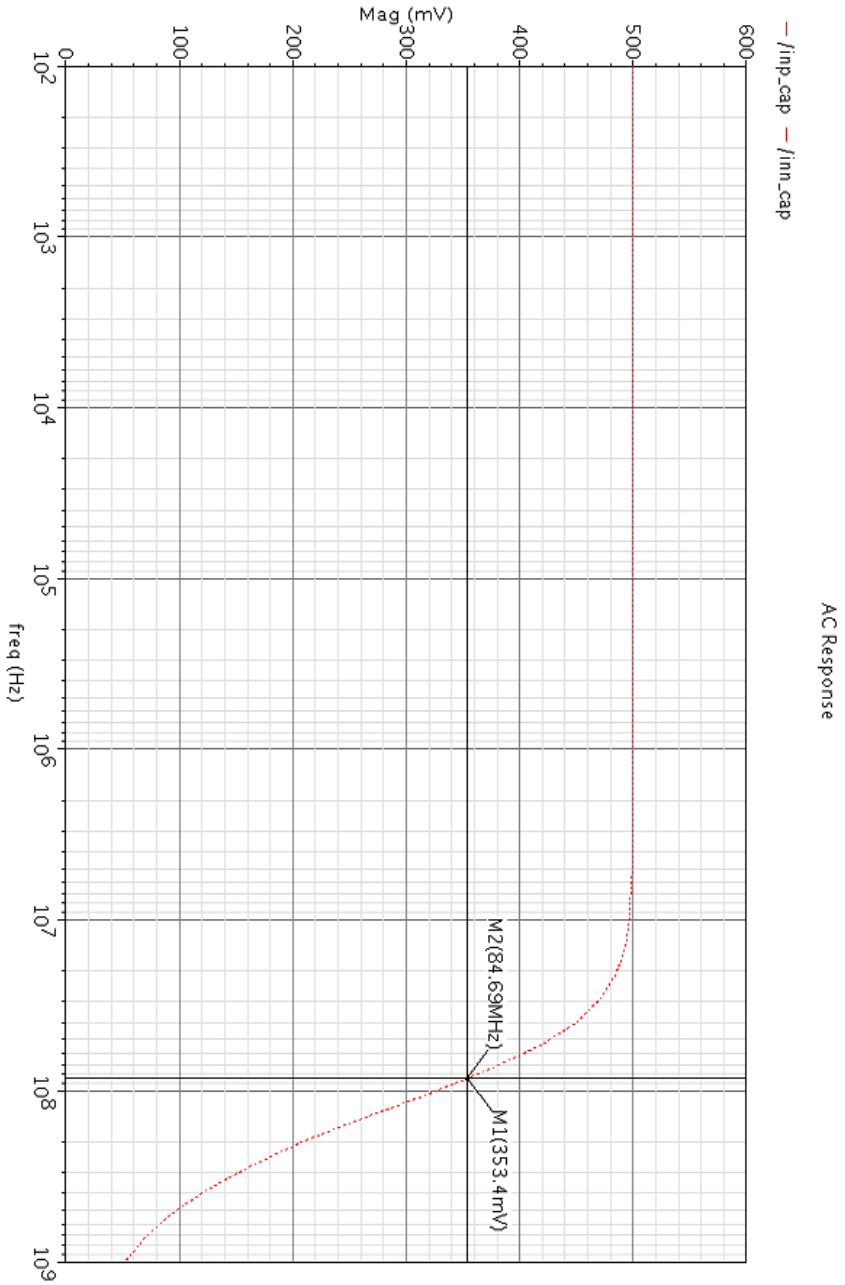


Figure 10: Measurement of the 3dB frequency of the input with a 10KΩ resistor connected to the input to simulate the output resistance of the previous driver stage. The 3dB frequency is at 84.69MHz which gives an input capacitance of 265fF. The on-resistance of the input-switches is 3274Ω . The equivalent maximum output resistance according to equation 4.7 becomes 7547Ω.

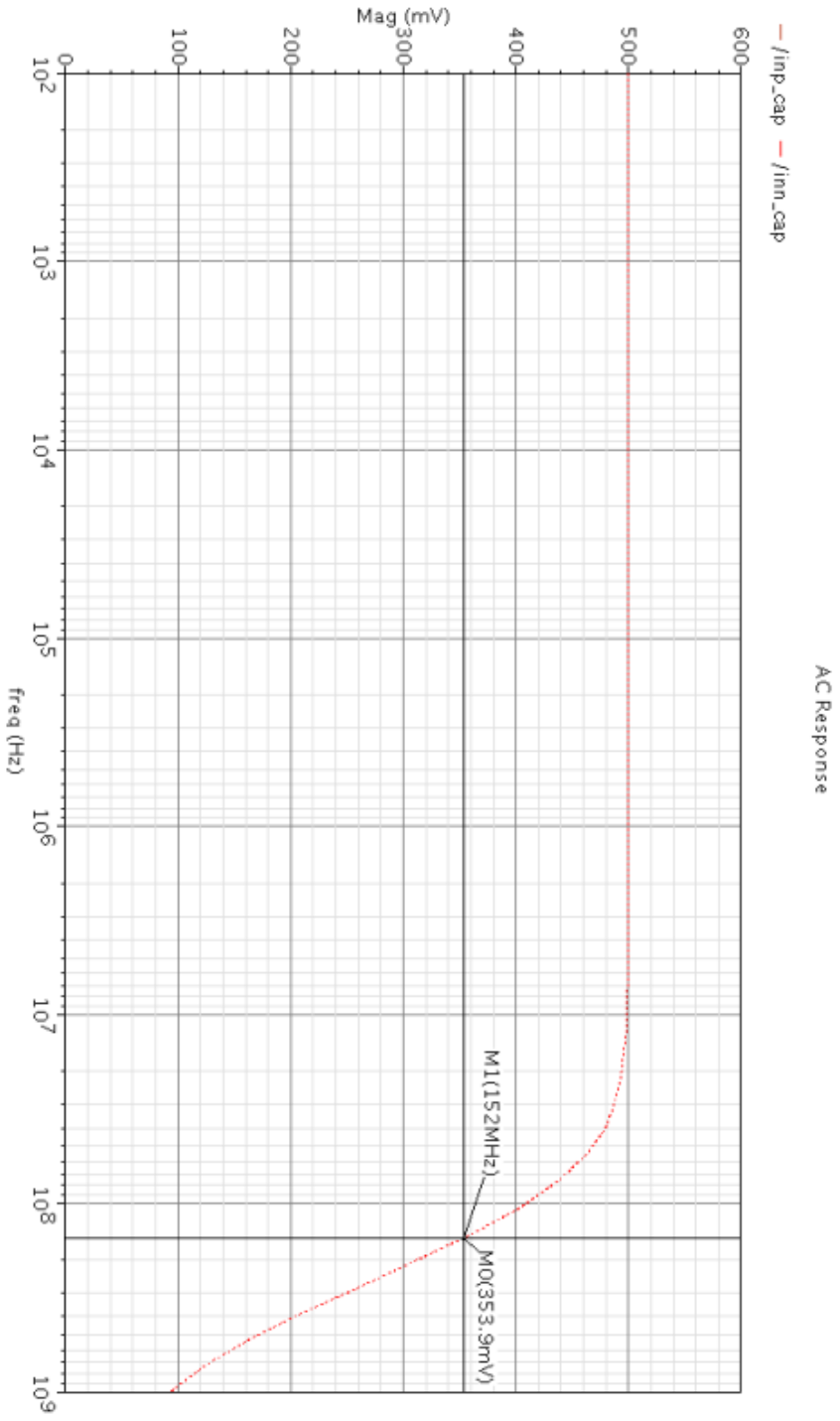


Figure 11: Measurement of the 3dB frequency of the input with a 10KΩ resistor connected to the input to simulate the output resistance of the previous driver stage. The 3dB frequency is at 152MHz which gives an input capacitance of 148fF. The input switches has been removed. The equivalent maximum output resistance according to equation 4.7 becomes 13.5kΩ.

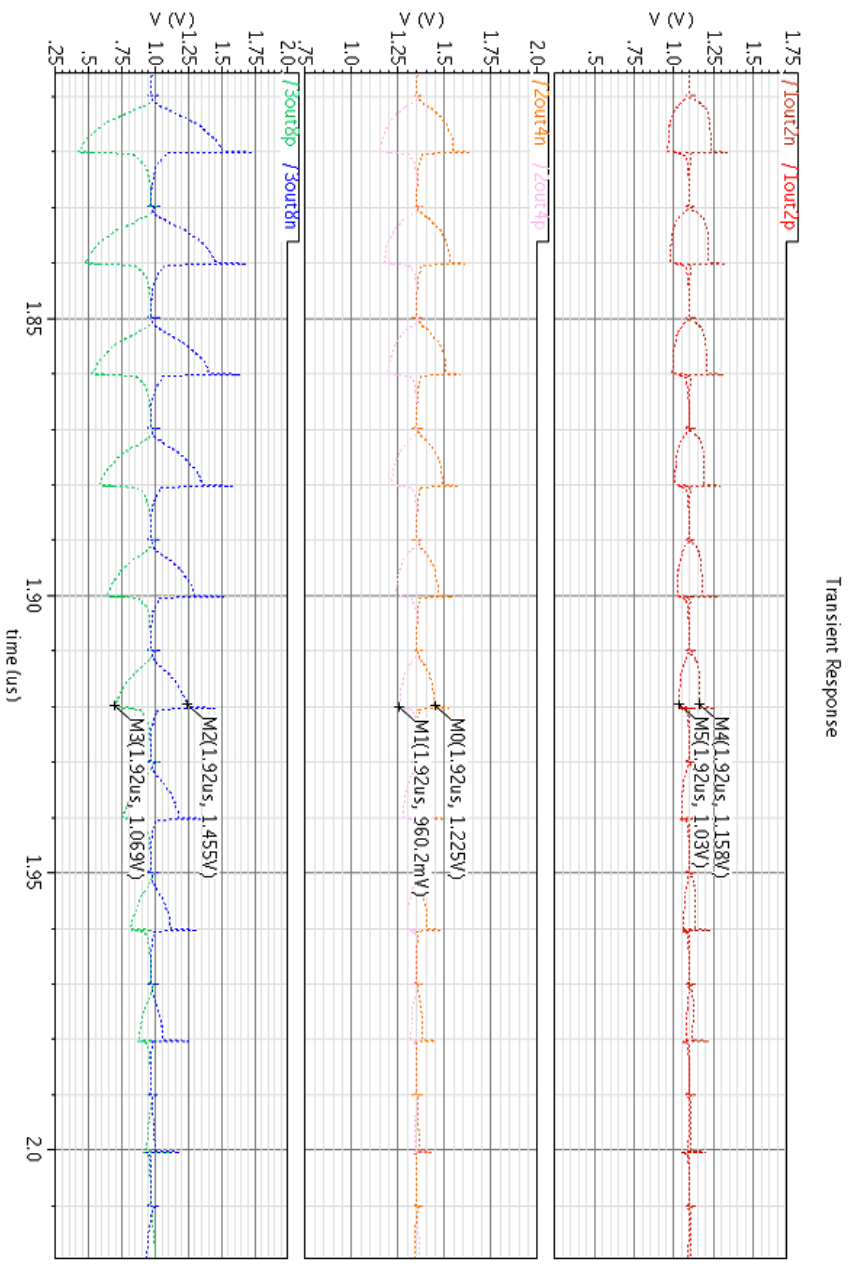


Figure 12: Gain of the 3 stages of the 5 bit converter with 18μA bias of the amplifiers. The plot shows the bad settling in the later stages.

.5 Matlab scripts

This section includes the Matlab scripts made for simulations. Only the 5-bit scripts are included. 3 and 4-bit scripts are similar.

.5.1 Comparator simulations

```
%%  comparator_testbench.m
%%
%%  Comparator Offset Monte Carlo Testbench
%%
%%  Dag Sverre Hoye - 2008
%%

% Reads comma-separated table from Cadence
infile = csvread('MCcomp100.csv',1,1);


[inrow,incol] = size(infile);

% Cleans up the input

for i=1:incol
    for j=1:inrow
        if infile(j,i) > 0.9
            infile(j,i) = 1;
        else infile(j,i) = 0;
        end
    end
end

% Finds the correct place where the comparator goes from 0 -> 1

for i=1:incol
    for j=1:inrow
        if infile(j,i)==1 & infile(j-1,i)==0
            place(i)=j;
        else
        end
    end
end

% Assign the correct deviation

value = (place - 80)*5e-4;

comp_std = std(value);

% Plots the offset in a histogram

hist(value,20)
```

```
xlabel('Offset [V]');
ylabel('Number of hits');
title(['\sigma = ', num2str(1000*comp_std),' mV']);
```

.5.2 Frequency analysis

```
%% testbench5b.m
%%
%% Frequency analysis testbench for the 5 bit converter.
%%
%%  Dag Sverre Hoye 2008
%%

clear all;

lsb = 1/2^5;     % Least significant bit
lsim = 256;      % Number of points in simulation
x_amp = 0.5;     % Input amplitude
hann = 0;        % Hanning window on or off
doplot = 0;      % Make frequency plot
fs = 50e6;       % Sampling frequency
txt = '';        % Additional information on plot

% Retrieve simulation data

binn = dataharv5b;
size(binn);

% Do 15 fft simulations. Retrieve ENOB for each run

g=1;
for g=1:15;
dacout = dac5bfft(binn,lsb,lsim,g);

[y,sndr,menob] = oct_dofft(dacout,x_amp,hann,doplot,fs,txt);

enobs(g)=menob;
end
enobs
```

```
%%  dataharv5b.m
%%
%%  Collects the data from the Cadence-simulations and puts them
    in the
%%  Matrix "binary". Used for the frequency analysises.
%%
%%  Dag Sverre Hoye - 2008
%%

function binary = dataharv5b

clear;

%Reads comma-separated table from cadence

input = csvread('5bit_coherent.csv',1,1);

[row,col] = size(input);

% Rounds the input to either zeroes or ones.

for i=1:col
    for j=1:row
        if input(j,i) > 0.9
            input(j,i) = 1;
        else input(j,i) = 0;
        end
    end
end

binary = input;

return
```

```
%%  dac5b.m
%%
%%  Takes the data fetched from Cadence and returns a thermometer
%%  code (Like a DAC). This code is used for frequency analysis
%%
%%  Dag Sverre Hoye - 2008
%%


function quantified = dac5b(binn,lsb,lsim,g)

%  Returns 10 thermometer code in the vector "quantified".

    for i=1:lsim
        quantified(i)=16*lsb*binn(i,g) + 8*lsb*binn(i,15+g) + 4*
            lsb*binn(i,30+g) + 2*lsb*binn(i,45+g) + lsb*binn(i,60+
            g); %#ok<AGROW>
    end


return
```

```
%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Foct_dofft.m
%% Description:
%% Author:        Carsten Wulff <wulff@iet.ntnu.no>
%% Created at:    Fri Sep 29 14:56:11 2006
%% Modified at:   Wed Mar 26 07:04:18 2008
%% Modified by:    Carsten Wulff <wulff@iet.ntnu.no>
%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%function [y,sndr,menob,fbin,Amp]=dofft(x,x_amp,hann,doplot,fs,txt
    )
% y: FFT Magnitude normalized to full scale
% sndr: signal to noise and distortion ratio
% menob: effective number of bits
% fbin: Which frequency bin the signal is in
% x: data input
% x_amp: full scale amplitude that the FFT should be normalized to
% hann: use a hanning window (hann=1), do not use a window (hann
    =0)
% doplot: show plot (doplot=1), do not show plot (doplot=0)
% fs: Sampling frequency, set to 1 for normalized frequency
% txt: Text to append to the plot title
function [y,sndr,menob,fbin,Amp,figure1]=oct_dofft(x,x_amp,hann,
    doplot,fs,txt)

%Resize down to 2^x points
mod(log2(length(x)),1)
if mod(log2(length(x)),1) ≠ 0
  M = length(x);
  N = 2^floor(log2(M));
  x = x(M-N+1:M);
end

%Remove DC
x = x - mean(x);

%Get length
N = length(x);

%Apply a hanning window if wanted
if hann==1
  w = x .*hanning(N)';
else
  w = x;
end

%Do the FFT and extract 0->Nyquist, adjust for the power from
    Nyquist
%to FS
y = abs(fft(w));
```

```
y(1) = 0;
y = y(1:N/2+1)*2;

%Normalize to maximum power
if hann==1
  sigpow = 1.2;
  y = y/(x_amp*N/2);
else
  sigpow = 1;
  y = y/(x_amp*N);
end

%Find the signal frequency, applies correction for hanning window
[u,fbin] = max(y);
spec = y;
fbin = fbin;
if fbin < 3
  fbin = 3;
end

if fbin > length(y)-10
  fbin = length(y) - 10;
end

if hann==1
  spread = [ -1 0 1];
else
  spread = [0];
end
signalBins = fbin+spread;

signalBins;

%Convert to dBFs
ya = 20*log10(y);

%Find the harmonics
sp = (1:20);
myind = mod((fbin-1).*sp,N);
k=1;
harmonics = ones(1,length(sp)*length(spread));
for i=1:length(myind)
  if(myind(i) > N/2)
    myind(i) = N - myind(i);
  elseif(myind(i) == 0)
    myind(i) = myind(i) + 1;
  end
  for z=1:length(spread)
    harmonics(k) = myind(i)+spread(z)+1;
    if harmonics(k) == 0 || harmonics(k) > length(y)
      harmonics(k) = 1;
    end
    k = k+1;
  end
```

```
end

%Remove the signal from the noise bins
noiseBins = 1 : length(y);
noiseBins(signalBins) = [];

noharmBins = 1:length(y);
noharmBins(harmonics) = [];

%Calculate signal power
s = norm(spec(signalBins));

%Calculate noise power
n = norm(spec(noiseBins));

h = norm(spec(noharmBins));


try
if n== 0
  sndr = Inf;
  menob = 0;
else

  nsndr = 20*log10(sigpow/n);
  sndr = 20 * log10(s/n);
  snr = 20*log10(s/h);
  isnr = 20*log10(sigpow/h);
  menob = (sndr-1.76)/6.02;
end
catch
  nsndr=0;
  sndr = 0;
  snr = 0;
  isnr = 0;
  menob = 0;
end

myind = myind+1;

%Generate X
x = linspace(0,0.5,length(y))*fs;

%Find signal and harmonics amplitude
Amp = ya(myind);

if doplot > 0
% Create figure

%MATLAB
figure1 = figure('XVisual',...
    '0x22 (TrueColor, depth 24, RGB mask 0xff0000 0xff00 0x00ff)',
        ...
```

```
    'Color',[1 1 1],'Name','FFT');

%OCTAVE
%figure1 = figure(   'Color',[1 1 1]);

% Create axes
axes1 = axes('Parent',figure1,'YGrid','on','XGrid','on');
box('on');
hold on;
axis([ 0 max(x) -140 0])

% Create plot
plot(x,ya,'Parent',axes1,'LineWidth',2,...
    'Color',[0 0 0]);


%Works in MATLAB
% Create scatter
scatter(x(myind),ya(myind),'MarkerFaceColor',[1 1 1],'
    MarkerEdgeColor',[0 0 0],...
    'Marker','diamond',...
    'DisplayName','Harmonics 1-20',...
    'Parent',axes1);

% Create xlabel
xlabel('Sampling Frequency [Hz]');

% Create ylabel
ylabel('Magnitude [dB]');

%Works in MATLAB
% Create legend
legend1 = legend(axes1,'show');
set(legend1,'Location','SouthEast');

title(['As=',num2str(max(ya)),' ENOB=',num2str(menob),' SNDR=',
    ...
       num2str(sndr),'  SNR=',num2str(snr) '  ISNDR= ',num2str(
           nsndr), ...
       '  ISNR=',num2str(isnr),'  f0=',num2str(fbin/N*fs/1e6),'MHz
           ',txt]);

menob

end
```

.5.3 INL/DNL analysis

```
%% testbenchINLDNL5b.m
%%
%% INL/DNL analysis testbench for the 5 bit converter.
%%
%%  Dag Sverre Hoye 2008
%%

clear all;

lsb = 1/2^5;  % Least significant bit

inn = dataharvID5b;  % Get the inputs from Cadence

lsim = length(inn);  % Length of simulation

dacout = dac5b(inn,lsb,lsim);

dacout = dacout + lsb/2   % Correct for 1/2 LSB inherent offset
    error.

analog = 0:1/(lsim-1):1;        % Analog rampsignal

% Finds the analog voltage for every output code transition.

trans(1)=0;

for g=1:10
    j=2;
    for i=2:lsim
        if dacout(i,g) > dacout(i-1,g)
            trans(j,g)=(analog(i)+analog(i-1))/2;
            j=j+1;
        else
        end
    end
end

% Calculate the differential non-linearity error

for m=1:10
    for k=2:32
        DNL(k-1,m) = ((trans(k,m)-trans(k-1,m))/lsb) - 1;
    end
end

% Calculate the integral non-linearity error

for n=1:10
    for l=2:32
        INL(l-1,n) = (trans(l,n)/lsb) - (l-1);
```

```
    end
end

% Plots

subplot(3,1,1);

plot(analog,'-r');

hold on;

stairs(dacout);

xlabel('samples');
ylabel('Voltage');
title('Analog input vs. digital output. NB 1/2 LSB offset error
    corrected');


subplot(3,1,2);

plot(DNL,'-og','MarkerEdgeColor','black');

xlabel('Output Code');
ylabel('DNL (lsb)');
title('DNL measurements 5 bit ADC');

subplot(3,1,3);

plot(INL,'-ob','MarkerEdgeColor','black');

xlabel('Output Code')
ylabel('INL (lsb)');
title('INL (best straight line) measurements 5 bit ADC');
```

```
%%  dataharvID5b.m
%%
%%  Collects the data from the Cadence-simulations and puts them
    in the
%%  Matrix "binary". Used for the DNL/INL analysis.
%%
%%  Dag Sverre Hoye - 2008
%%

function binary = dataharvID5b

clear;

% Reads comma-separated table from cadence

input = csvread('5bit_INL_DNL_MC.csv',1,1);

[row,col] = size(input);

% Rounds the input to either zeroes or ones.

for i=1:col
    for j=1:row
        if input(j,i) > 0.9
            input(j,i) = 1;
        else input(j,i) = 0;
        end
    end
end

binary = input;

return
```

```
%%  dac5b.m
%%
%%  Takes the data fetched from Cadence and returns a thermometer
%%  code (Like a DAC). This code is used for DNL/INL analysis.
%%
%%  Dag Sverre Hoye - 2008
%%

function quantified = dac5b(binn,lsb,lsim)

% h is the number of the Monte Carlo run. Returns 10 thermometer
    codes in
% the matrix "quantified".

for h=1:10
    for i=1:lsim
        quantified(i,h)=16*lsb*binn(i,1) + 8*lsb*binn(i,h+10) + 4*
            lsb*binn(i,h+20) + 2*lsb*binn(i,h+30) + lsb*binn(i,h
            +40); %#ok<AGROW>
    end
end

return
```