



NTNU – Trondheim
Norwegian University of
Science and Technology

Current-Mode SAR-ADC In 180nm CMOS Technology

Bård Egil Eilertsen

Master of Science in Electronics
Submission date: June 2012
Supervisor: Trond Ytterdal, IET

Norwegian University of Science and Technology
Department of Electronics and Telecommunications

Problem description

The objective of this project is to design an energy efficient current-mode 50MS/s, 9-bit SAR-ADC in 180nm CMOS technology. Main performance parameters are power consumption, area and low switching-noise.

The project consists of the following tasks:

- Perform a literature survey of current-mode ADCs to establish current-state-of-the-art
- Analyze and compare published work
- Based on above, choose ADC architecture and the individual building blocks
- Design the ADC at transistor level and characterize the performance by simulation.
- If time allows, perform the layout of the circuit and study the effects of layout parasitics on ADC performance

The specifications are as follows:

- CMOS technology: AMS 180nm
- Supply voltage: 1.8V
- Maximum input current (FSR): TBD
- Input resistance: TBD
- Resolution: 9-bits
- ENOB: 8.5
- Sampling frequency: 50MHz

Acknowledgements

I would like to thank my supervisor Trond Ytterdal. His knowledge and excellent guidance has been very helpful during the work on this master thesis. His fellow Surya Sharma has also been to great help. Her know-how in layout technique has been crucial for my work.

Also, I would like to thank my fellow students, Kristine Aakre, Simen Dahl, Mehmet Altan and Eirik Steen-Hansen for all their help and encouragement through this work.

Abstract

This thesis presents a fully differential 9-bit current-mode successive approximation (SAR) ADC. The circuit is designed in $0.18\ \mu\text{m}$ technology with $1.8\ \text{V}$ supply voltage and has a current draw on $472\ \mu\text{A}$. The ADC has a sampling frequency on $50\ \text{MHz}$ and has a maximum ENOB on $8.42\ \text{bit}$. Because of non-linearity will ENOB be input frequency dependent and degrade to $6.87\ \text{bit}$.

The design is based on conventional current-mode SAR ADC operation, but with a new comparator design and time interleaving. Time interleaving is used to increase the sampling frequency 10 times.

The circuit needs a high degree of matching to work properly. Sub-threshold operation in several current sources gives a high degree of uncertainty in the current value. Thus several calibration circuits are presented, but are not implemented.

Sammendrag

Denne avhandlingen presenterer en differensial 9-bit strøm-modus suksessiv tilnærming (SAR) ADC. Kretsen er laget i 0,18 μm -teknologi med 1,8 V forspenning, og har et strømforbruk på 472 μA . ADCen har en samplingsfrekvens på 50 MHz, og har en maksimal ENOB på 8,42 bit. På grunn av ikke-linearitet vil ENOB være avhengig av inngangsfrekvens, og ENOB går ned til 6,87 bit.

Konstruksjonen er basert på konvensjonell strøm-modus SAR ADC operasjon, men med et nytt komparator-design og time interleaving¹. Time interleaving brukes til å øke samplingsfrekvensen 10 ganger.

Kretsen trenger en høy grad av matching for å fungere. Sub-terskel drift i flere strømkilder gir en høy grad av usikkerhet i strømstyrken. Dermed blir flere kalibrering kretser presentert, men uten å være implementert.

¹Kapittel 2.4.2 forklarer begrepet time interleaving

Contents

Acronyms	xv
1 Introduction	1
1.1 Motivation	1
1.2 Specification	2
2 Theoretical background	3
2.1 Tools	3
2.2 Current mode	3
2.3 Charge injection	3
2.3.1 Dummy switch	4
2.4 SAR ADC	4
2.4.1 Binary search algorithm	5
2.4.2 Time interleaving	5
2.4.3 INL and DNL	5
2.5 FOM	5
2.6 SINAD and ENOB	6
2.7 Current mirror	6
2.7.1 Simple current mirror	6
2.7.2 Cascode current mirror	7
2.7.3 Wide-swing current mirror	8

3	Schematic design	9
3.1	Comparator	9
3.1.1	Calibration circuit	9
3.2	SAR ADC	11
3.2.1	Current mirror array	12
3.2.2	Input mirror	14
3.3	Sampler	15
3.4	Digital control logic	16
4	Layout	17
4.1	Dog bone	17
4.2	Final layout	18
4.2.1	Comparator	18
4.2.2	Current ladder and current input mirror	18
5	Results	19
5.1	Calibration Circuit	19
5.2	Current mirrors	19
5.2.1	Sample and hold	19
5.3	Switch	20
5.4	SAR ADC	20
5.4.1	SNR	20
5.4.2	ENOB	20
5.4.3	Monte Carlo	21
5.4.4	Offset and gain error	21
5.4.5	INL and DNL	22
5.4.6	Performance summary	22
5.5	Layout	23
5.5.1	Comparator	23

5.5.2	Current ladder and input current mirror	24
6	Discussion	25
6.1	Calibration Circuit	25
6.2	Current mirrors	25
6.2.1	Sample and hold	25
6.3	Switch	26
6.4	SAR ADC	26
6.4.1	SNR	26
6.4.2	ENOB	26
6.4.3	Monte Carlo	27
6.4.4	Offset and gain error	27
6.4.5	INL and DNL	27
6.4.6	Performance summary	28
6.5	Layout	28
6.5.1	Comparator	28
6.5.2	Current ladder and input current mirror	29
6.6	Comparison with others	29
6.7	Future work	30
7	Conclusion	31
	References	34
A	Calculation of ENOB	35
B	Input and output impedance to a simple current mirror	37
C	Input and output impedance to a wide-swing current mirror	39
D	Layout	41

E	Calibration	43
F	Current draw	45
G	SAR ADC	47
G.1	Top file	47
G.2	SAR ADC	54
G.3	Input sampler	57
G.4	Comparator	59
G.5	NMOS switch	60
G.6	NMOS switch with dummy transistor	61
G.7	Inverter	62
H	MATLAB	63
H.1	Top code	63
H.2	Script	65

Acronyms

ADC - Analog to Digital Converter

AVG - AVeraGe

CMOS - Complementary Metal - Oxide - Semiconductor

DNL - Differential NonLinearity error

DR - Dynamic Range

ENOB - Effective Number Of Bits

FOM - Figure Of Merit

FSR - Full Scale Range

INL - Integral NonLinearity error

LSB - Least Significant Bit

MOS - Metal Oxide Semiconductor

MSB - Most Significant Bit

NOB - Number Of Bits

OTA - Operational Transconductance Amplifier

RMS - Root Mean Square

SAR - Successive AppRoximation

SINAD - Signal to Noise And Distortion

SNR - Signal to Noise Ratio

STD - STandard Deviation

Chapter 1

Introduction

This report presents and discusses the possibility of using current-mode design in a SAR ADC. The circuit shall convert a 25MHz input signal into a nine bit digital word. The goal is to make a fast, low noise ADC that also has a small current draw. Current-mode as a design technique is used because of its promising characteristics. No high impedance nodes in current-mode circuits will ensure that the circuit has high speed and low voltage swing. The SAR ADC has to work within a given specification.

This paper is organized as follows: Chapter two presents some of the theory which is useful to understand the text. Chapter three present the schematic design. Several challenges with the design as well as suggestions to calibration are also presented. Layout is described in chapter four followed by simulation result in chapter five. Discussions of the result are done in chapter six. The paper is finished with a conclusion in chapter seven.

1.1 Motivation

Development of new generations of CMOS technology pushes the supply voltage down, and thus reducing the dynamic range of voltage mode circuits. Simultaneously will the complexity in modern electronics increase and cause a larger, and more irregular current draw. Electronics devices made for a practical purpose will need cooling mechanism to wield and wireless devises have lower battery life. This is only two reasons for why it is important to focus more on making power efficient circuits. In this thesis one common part of an integrated circuit, the SAR ADC, is studied.

Today, most signal processing is done in the digital domain. Performance to the ADC, present on the border between the analogue and digital domains, are more

important than ever. SAR ADC is currently a very popular design technique to realize medium speed and medium accuracy ADC. Often will a SAR ADC use voltage as a signal and capacitors will grow larger for every new bit added. Current-mode can be seemed as an attempt to reduce area. By removing large capacitors and replacing them with small transistors the total area can be reduced.

The current draw to a voltage mode circuit depends on several factors. This will introduce a power noise and in turn will introduce noise to the entire system. Such noise will potentially affect the system negatively. Current-mode as a design technique can be used to counteract power noise since it has a constant current draw.

1.2 Specification

The specification to the SAR ADC is given in table 1.1.

Table 1.1: Specifications

Supply voltage	1.8V
Resolution	9-bits
ENOB	8.5-bits
Sampling frequency	50MHz
Technology	0.18 μ m
Power consumption	Minimize

Chapter 2

Theoretical background

The theoretical foundation to understand the text is laid in this chapter.

2.1 Tools

Questa ADMS is a design tool for simulation of analogue and mixed signal circuits. The software is developed by Mentor graphics [4]. The tool combines four simulation engines, Eldo, Questa, ADiT and ELDO RF. The waveform results from the simulation are displayed in EZwave.

Cadence design systems [3] with Austriamicrosystems HIT KIT [2] was used to create layout.

2.2 Current mode

Current-mode is a design technique that uses current as a signal rather than voltage. A benefit with this technique is that there are no capacitors or high impedance nodes that will degrade the speed. In new technologies where the supply voltage will reduce the signal swing this technique is well suited. This is because low voltage change on a gate can make a high current swing through a transistor.

2.3 Charge injection

Charge injection is an effect that emerges in transistors that is used as a switch [9]. Channel charge in a transistor is pushed out to the source and drain nodes

when the gate voltage is turned rapidly from V_{dd} to V_{ss} . In a NMOS transistor the channel charge will be negative. The amount of charge in a transistor channel is given in equation 2.1, and the total voltage change in the node that the transistor is connected to is given in 2.2 [9].

$$\Delta Q_{channel} = -W \cdot L \cdot C_{ox} \cdot V_{eff} \quad (2.1)$$

$$\Delta V = \frac{\frac{\Delta Q}{2}}{C} = \frac{-W \cdot L \cdot C_{ox} \cdot V_{eff}}{2 \cdot C} \quad (2.2)$$

In equation 2.2 the charge is divided on two since it is assumed that equal amount of charge goes to drain and source.

2.3.1 Dummy switch

To counteract the effect of charge injection in a switch a dummy switch is inserted [9]. The dummy have a 180 degree phase shift on the gate voltage compared to the switch so it can suck up the charge. Also the dummy have only half the width of the switch. This is because the dummy shall only cancel half the charge injected by the switch.

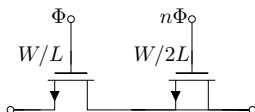


Figure 2.1: Switch with dummy transistor

2.4 SAR ADC

Downscaling of technology gives a lower supply voltage and less instinctive gain [23]. This lowered supply voltage makes OTA undesirable to produce since they will need a high current draw to get sufficient gain. SAR ADC is a technique that does not use OTA, but only comparators.

SAR ADC is often the architecture of choice for low cost ADC with medium resolution and with sample rates less than 5MS/s [22]. Normally a SAR ADC has a resolution from 8 to 16 bits and has low power consumption.

2.4.1 Binary search algorithm

A SAR ADC uses binary search to find the correct digital output word. The search algorithm works as follows [1]. First the input is sampled and the circuit guesses the input to be half of the input swing. The comparator will determine if the circuit has guessed to high or too low. The comparator output will also determine the first bit. In the next attempt the circuit guesses on a higher or lower value depending on the comparator output from the last comparison. This guessing continues until the ADC has run through all bits. When all bits are gone through the word is ready and the circuit can re-sample.

2.4.2 Time interleaving

To get higher speed in a ADC a technique called time interleaving is often used. The theory is that one ADC uses longer time to find the output word than the time it takes to collect one sample. By sampling more often and divide the samples on several different ADC that is arranged in parallel the total ADC will produce more output words and the sampling frequency can increased. One important aspect is that there is not any overlapping in sampling between the different ADC, since this will cause errors.

2.4.3 INL and DNL

A digital word will represent the input signal with quantified values. INL is defined to be the difference between a ADCs output value and a straight line.

$$I_q = I_{input} - I_{quantization}$$

In an ideal world the nearest quantified value will only be $\frac{1}{2}LSB$ from the input signal.

Each step in an ADC is ideally one LSB. DNL is defined to be the variation in step size away from one LSB [9]. An ideal ADC has a DNL equal to zero.

2.5 FOM

A FOM often found in literature concerning current-mode SAR ADC is given in equation 2.3.

$$FOM = \frac{f_s * 2^{NOB}}{P_{Norm}} \quad (2.3)$$

Where NOB is the number of bits in the ADC, f_s is the sampling frequency and P_{Norm} is the dissipated power normalized to 8 bits. Put in mathematical terms P_{Norm} is given in equation 2.4.

$$P_{Norm} = \frac{P_{Diss} * 2^8}{2^{NOB}} \quad (2.4)$$

2.6 SINAD and ENOB

SINAD is the ratio between signal power to the noise and distortion power. Put in mathematical terms:

$$SINAD = 10 * \log \frac{|P_{SIGNAL}|}{|P_{NOISE}| + |P_{DISTORTION}|}$$

SINAD is used to find ENOB. ENOB is defined to be the number of bits the SINAD ratio describes.

$$ENOB = \frac{SINAD - 1.76dB}{6.02dB} \quad (2.5)$$

The calculation for equation 2.5 is given in appendix A.

2.7 Current mirror

Current mirror is a essential building block in current mode design. In the following section three common current mirror designs will be presented. Some parts of the current mirrors will also be analyzed.

In general will a low W/L ratio make the current mirror more accurate during mismatch simulation because it will give a lower variation in threshold voltage[23].

2.7.1 Simple current mirror

The mirror has a diode connected input transistor and a common-source output transistor. Calculation for the input and output resistance is done in appendix B. Input impedance is given as $1/gm$ and output impedance is given as $1/gds$. This means that the input impedance is low compared with the output impedance.

Another interesting aspect of this mirror is the frequency response. By analyzing the input and output node one can see which node that is critical in relation to

frequency. Equation 2.6 and 2.7 respectively displays equations for the frequency response to the input and output node. In equation 2.7 there is not taken into account the effect of the output load, but only the output transistor.

$$\omega_1 = \frac{g_m}{2C_{gs}} = \frac{I_D}{\left(\frac{2}{3}WLC_{ox} + WL_{ov}C_{ox}\right) * V_{eff}} \quad (2.6)$$

$$\omega_2 = \frac{g_{ds}}{C_{dg}} \quad (2.7)$$

Equation 2.6, which displays the node with the lowest pole frequency, show that the bias current play a big role in the speed of the current mirror. By biasing the mirror with a high input current the speed can be increased. This comes at the cost of current draw. Another interesting observation from equation 2.6 is that V_{eff} will also increase the frequency to the current mirror if it is chosen to be a low value.

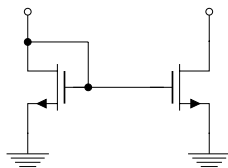


Figure 2.2: Simple current mirror

2.7.2 Cascode current mirror

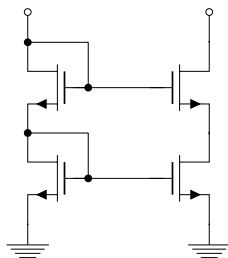


Figure 2.3: Wilson current mirror

In addition to increasing the output resistance the cascode current mirror showed in figure 2.3 will also reduce the output capacitance. This is because the mirror can have a small output transistor with a low C_{dg} . C_{dg} is given in [9] as $W \cdot L_{ov} \cdot C_{ox}$. The cascode transistor can have a small width regardless of the bottom transistor.

The frequency response will be similar to the simple current mirror.

2.7.3 Wide-swing current mirror

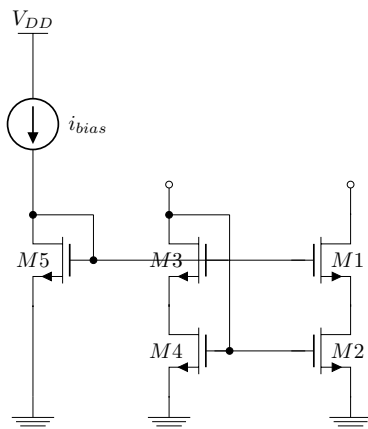


Figure 2.4: Wide-swing current mirror

This current mirror provides the circuit with a higher swing because the bottom transistor is connected to the drain of the cascode transistor. The input and output resistance is calculated in appendix C.

The recommended way to size the transistors from [9] are as follows.

- M1: $\frac{W/L}{n^2}$
- M2: W/L
- M3: W/L
- M4: $\frac{W/L}{n^2}$
- M5: $\frac{W/L}{(n+1)^2}$

Where n is an integer which normally is chosen to be one. This way of selecting the sizes is motivated by matching V_{eff} is such a way that everything is biased in the active region. The current source has the same value as the maximum input current to create the correct gate voltage.

Minimum allowed output voltage to ensure that all transistors are in active region is given in equation 2.8. To give a linear sampling over a wide swing of currents V_{eff} in transistor 2 and 3 is lowered [9].

$$V_{OUT} > (n + 1)V_{eff} \quad (2.8)$$

A drawback by using the wide swing current mirror is the extra branch from V_{dd} to ground. This increases the current consumption to the circuit.

Chapter 3

Schematic design

In this chapter the schematic design is presented. Different design choices are motivated and described.

3.1 Comparator

The comparator was designed in the project [8]. The schematic is reprinted in figure 3.1 and the SPICE code is given in appendix G.4. The circuit is fully differential with a negative input current and voltage on the output nodes A and B. The latch in the comparator needs to be reset every comparison. That is why it uses the clock as a reset signal. The clock is specified to run at 50MHz. The two branches are reset to V_{CM} to get a more constant current draw.

The diode connected input transistors give the input node a resistance and a capacitance given by the equations 3.1 and 3.2.

$$R = \frac{V_x}{i_x} = \frac{r_{ds}}{1 + gm * r_{ds}} \approx \frac{1}{gm} = 86.7K\Omega \quad (3.1)$$

$$C = c_{gs} = 2.15fF \quad (3.2)$$

Key performance characteristics is given in table 3.1.

3.1.1 Calibration circuit

The project report [8] stated in future work that a calibration circuit is needed to utilize the comparator. This comes from a high standard deviation after Monte

instead of analogue integration. This design is not implemented in the digital domain, but it is discussed here as a possibility.

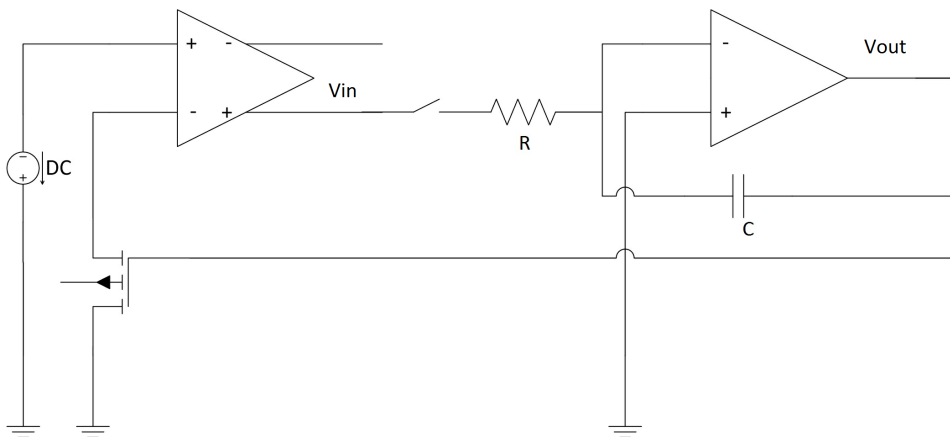


Figure 3.2: Calibration circuit

3.2 SAR ADC

The ADC is a time interleaved circuit with ten cells in parallel. Each cell has a 5MHz SAR ADC. Together they form a 50MHz ADC. Figure 3.3 shows how one cell looks like. This design originated from articles [17], [18] and [20]. The SPICE code for the circuit is given in appendix G.2. The circuit is a differential design in order to achieve 9 bit. The circuit was turned into a differential design to combat second harmonics in the frequency spectrum. One cell consists of two input current samplers, two current arrays and one comparator. The general functionality is as follows.

The first step is to sample the input current. The input can vary between $0\mu A$ up to $4\mu A$. If the bias current is included will the input vary from $2\mu A$ to $6\mu A$. After the sampling is completed the binary search can begin as described in chapter 2.4.1. The current array sets up nine different currents ranging from $2\mu A$ down to $\frac{2\mu A}{2^8}$. The MSB is first current to open. The two currents in to the comparator will determine which is largest and thus determine if MSB shall be set on or off. After the comparator has determined the largest the next bit is opened. This will continue until all bits is cycled and the cell can re-sample.

$$i_{in} + i_{bias} + i_{comparator} = i_{array} + 2 * i_{bias} \quad (3.4)$$

The sum node is indicated in figure 3.3 as the input node to the comparator. In

this node the input current, the comparator current and the bias current has to be equal to the current in the array and the bias current as indicated by equation 3.4. The only free variable is the comparator current.

The impedance the comparator sees on the sum node is critical to ensure high speed. Low time constant is necessary, and it depends on capacitance and resistance. The diode connected transistor in the comparator is connected in shunt with the input mirror and the current ladder. Since it has such a low resistance, it will dominate seen from the sum node. Both the input mirror and the current ladder has an output resistance which is proportional to $(\frac{1}{g_{ds}})^2$. By extracting g_m we can use equation 3.1 to determine that the resistance is $86.7K\Omega$.

Next step is to determine the capacitance. This is also dominated by the diode connected transistor in the comparator because of it is connected to the gate. It is also dependent on C_{dg} in the current array because of the sheer number of transistors. By using the result in equation 3.2 the time constant will be 303MHz as given in equation 3.5.

$$\begin{aligned}\tau &= R * (C_{gs} + C_{dg}) = 86.7K\Omega * (2.15fF + 36 * 1fF) = 3.3ns \\ \omega &= \frac{1}{\tau} = \underline{303MHz}\end{aligned}\tag{3.5}$$

In the current array a simple switch is used between the sum node and the current sources. This switch is connected to drain in three different transistors. The charge injection will go through the drain and not affect the operation.

3.2.1 Current mirror array

The current array consists of Wilson current mirrors as described in chapter 2.7.2. Wilson current mirrors are chosen over simple current mirror mainly to ensure low capacitance on the sum node. The bottom transistor mirrors the current while the top transistor acts like a buffer with small length and width. The dominate pole is the output node to the mirror. This would normally give a low pole frequency, but since the comparator has a diode connected transistor on the output it is connected to a low impedance node. Current-mode circuits are often made with no high impedance nodes to ensure high speed operation[7]. This will ensure that speed is not an issue.

The current sources are run with a low overdrive, which is a disadvantage with respect to mismatch. A variation in V_T will highly affect the circuit. This means that the current source that feeds the current array needs to be a variable current source that can be calibrated. A method of calibrating a current source is given in [9]. The circuit on the left, in figure 3.4, shows the calibration mode, and the circuit on the right shows work mode. Here only 10% of the current source is calibrated. The transistor in this particular calibration needs to be designed with a low g_m

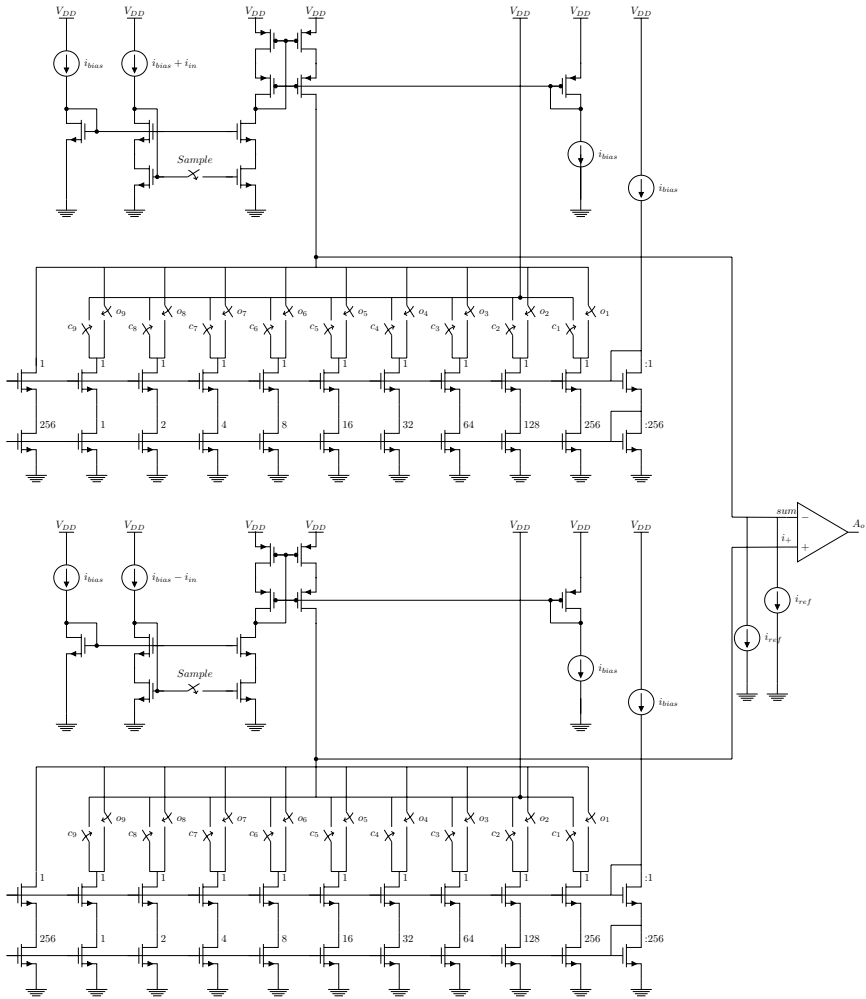


Figure 3.3: SAR ADC

to work properly. This is done to reduce charge injection errors, and so that the circuit will work properly even if some of the voltage on the gate leak out. The transistor current will be equal to the gate voltage multiplied with gm. A low gm will result in a higher gate voltage for the same transistor current.

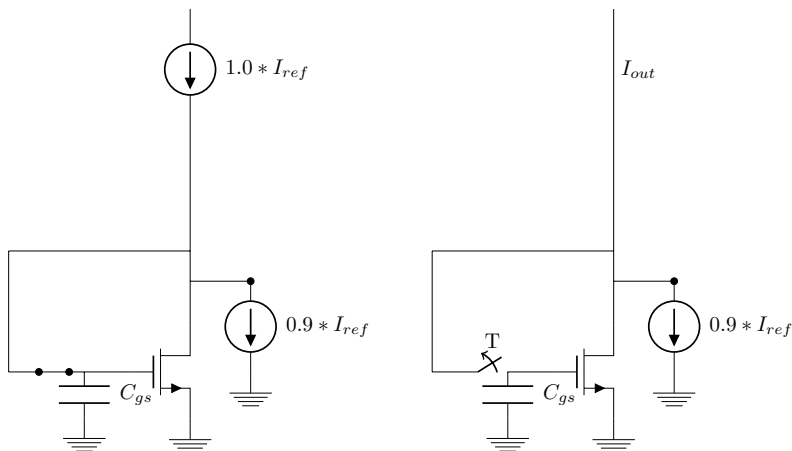


Figure 3.4: Current source calibration

3.2.2 Input mirror

The input current mirror is a wide swing current mirror to get good input current sampling across the input swing. Simple current mirror and Wilson current mirror was not adequate as a input sampler.

The wide-swing current mirror present on the input of the SAR ADC cell needs to be biased in order to work properly. The length and width need also to be designed. This current mirror is not meant for low power applications since it needs its own bias circuit.

In theory the width and length ratio is determined on basis on the square law current formula which is displayed in equation 3.6 [9]. This formula is not correct with reality so it is only used as an initial pointer.

$$i_D = \frac{1}{2} \mu_n c_{ox} \frac{W}{L} (V_{gs} - V_T)^2 (1 + \lambda(V_{DS} - V_{eff})) \quad (3.6)$$

If the input current mirror does not work properly the whole circuit suffers.

3.3 Sampler

The input sampler needed to be able to sample a 25MHz current signal. It also needs to sample a signal with an error less than one LSB. This means that the sampler must remain linear over a wide range of bandwidth and signal amplitudes. Figure 3.5 illustrate the sampler and how it splits between all ten SAR ADC cells. The code for the sampler is given in appendix G.3.

From chapter 2 it is showed that the two dominate poles to the current mirror are at the gate to the output transistor and at the output node. The pole with the lowest frequency is the internal pole since it will have the same resistance as the pole at the load, but have double the capacitance.

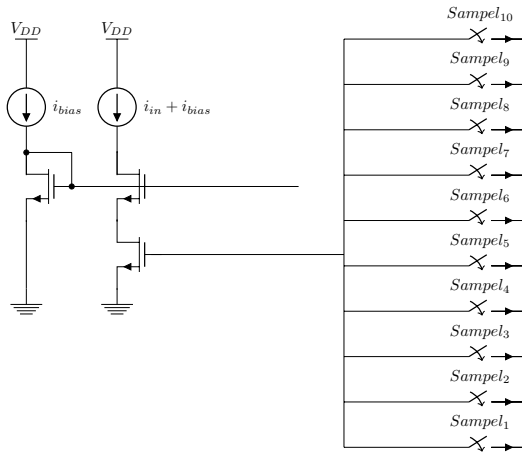


Figure 3.5: Sampler

The dummy switch is only inserted into the switch in the input sampling circuit and not in the current source array. The reason for this is the two nodes the switch is connected to. In the input sampling the switch is connected to a gate. Charge injection will change the gate voltage and in turn change the input current through the transistor. This will seriously degrade the circuit, especially since the change in gate voltage will be input current dependent. Equation 3.7 displays V_{eff} to the switch and how it is input voltage dependent. This V_{eff} can be put into equation 2.2 to calculate the charge injection.

$$V_{eff} = V_{gs} - V_t = V_g - V_{in} - V_t \quad (3.7)$$

3.4 Digital control logic

The control logic is written in VHDL code. The code has state machine behavior with ten states. Each state gives out all control signals needed. Every cell in the ADC has to go through ten stages in order to produce the output word. One is a reset and sample stage, and nine compare stages. These stages are for all cells overlapped in the state machine in such a way that no cells are in the same stage at the same time. In every state one cell is in reset when all the others are in a different compare stages.

The only delay in the code is between samples on the input sampler to ensure non-overlapping samples. The rest of the code has no delay in it because it is assumed that the analogue circuit has largest delay.

Chapter 4

Layout

In this chapter the layout is presented. General design techniques and changes to the design because of design rules and layout practice is discussed.

4.1 Dog bone

One change to the design was done to remove dog bone transistors. The W/L ratio was kept but W was increased to match the width of the via down to the transistor channel. Figure 4.1 shows two transistors. The one on the left has a dog bone structure. The one on the right is increased to remove the dog bone. The problem with dog bone transistors is the production. When it is produced the width will be continual and thus dependent on the y-axis which is not in any way beneficial.

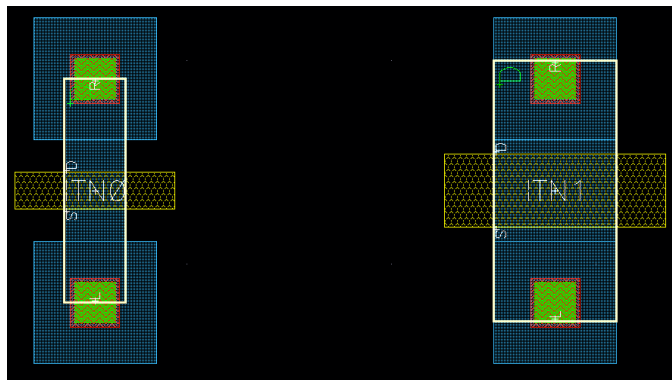


Figure 4.1: Dog bone removed

4.2 Final layout

4.2.1 Comparator

A picture of the final layout of the comparator is given in figure D.1 in appendix D. It is designed in a similar way as the schematic in figure 3.1. The area is $14.75 \mu\text{m}$ times $9 \mu\text{m}$.

4.2.2 Current ladder and current input mirror

The final layout is showed in figure D.2 in appendix D. The layout consists of ten current sources connected together. Over every current source there is a switch. In addition there is the pmos input current mirror. The area of the current array with the input mirror is $50 \mu\text{m}$ times $490 \mu\text{m}$.

Small amount of the layout is simulated at the time. This is because it will be easier to isolate errors. By simulating small portions of the circuit the origin of errors are found easily. Simulation of layout is done by connecting the parasitic extracted netlist into existing test benches.

Since one cell consist of two current ladders and the SAR ADC consist of ten cells the total area is estimated to be around $1000 \mu\text{m}$ times $490 \mu\text{m}$. In addition to this comes all wiring.

Chapter 5

Results

This chapter presents all simulation results. Simulation set-up is also described. Results will be discussed in the next chapter.

5.1 Calibration Circuit

In appendix E there is a picture from the simulation of the calibration circuit. It shows that after turned on, the calibration circuit uses $2.4 \mu\text{s}$ to match the two input currents. When it has found the correct value the circuit stops integrating and holds the result.

5.2 Current mirrors

5.2.1 Sample and hold

The input current mirror in the sampler is simulated with a dc test bench. This test is done without a dummy switch between the two branches. A linear increasing current is applied on the input, and the output current is measured. The results are given in the table below.

Table 5.1: Input current mirror

Input current: $\mu\text{[A]}$	2	3	4	5	6
Output current: $\mu\text{[A]}$	2.02289	3.02103	4.02111	5.02406	6.02968
Deviation:	1.145%	0.701%	0.5278%	0.481%	0.495%

An ac simulation is also done in a similar manor. The current source on the input had rail to rail amplitude with an increasing frequency. The result of the simulation is in table 5.2.

Table 5.2: Input current mirror

Input frequency: [MHz]	0	10	25	40	50
Output current: μ [A]	2.00296	2.00182	1.99548	1.98205	1.96782
Deviation:	0.148%	0.091%	0.226%	0.897%	1.609%

5.3 Switch

Two kinds of nmos switches were used in the design, switches with and without a dummy switch. Both were designed with minimum length and width to ensure that speed was not an issue.

Switches with dummy device were used in the sampling circuit to remove charge injection. The charge injected from a single switch was 0.158m[V]. Test with a dummy transistor showed that 100% off all charge was removed.

On and off-resistance was simulated to be respectively 5.6k Ω and 500M Ω . To test the resistance the transistors drain and source node was connected to a common node while the gate was switched between V_{ss} and V_{dd} .

5.4 SAR ADC

5.4.1 SNR

SNR in the circuit is about 50dB regardless of input frequency. Theoretical maximum from equation A.5 with nine bit is 56dB.

5.4.2 ENOB

To test ENOB a single frequency sinusoid was applied as an input. This input gave a corresponding output in form of a nine bit digital word. With help of the MATLAB script given in appendix H, SNR and SINAD could be calculated from a frequency response. From the SINAD result, ENOB could be calculated. The results of the ENOB simulation for different input frequencies is given in table 5.3. The amplitude to the input signal was rail to rail.

Table 5.3: ENOB

f_{in} [MHz]	1	2	3	4	5	8	16	20	25
ENOB [bit]	8.22	8.42	8.28	8.04	8.04	7.89	7.00	6.87	6.87

5.4.3 Monte Carlo

All current sources in the current ladder were simulated. Results from the Monte Carlo simulation is given in table 5.4. The values for ideal current are rounded so that there are three significant figures. The other numbers are extracted from Questa ADMS. There is also a column for simulated currents without Monte Carlo simulation.

Table 5.4: Monte Carlo simulation

	AVG: V_t [V]	Ideal I_d [μ A]	Sim.[μ A]	AVG: I_d [μ A]	STD: V_t [V]	STD: I_d [μ A]
Bias:	0.348	2.00	1.99692	2.000	0.0194	0.402
Bit1:	0.348	2.00	1.99692	2.027	0.0206	0.539
Bit2:	0.349	1.00	1.0023	1.022	0.0180	0.277
Bit3:	0.344	0.500	0.5026	0.564	0.0207	0.128
Bit4:	0.348	0.250	0.25189	0.259	0.0176	0.0586
Bit5:	0.348	0.125	0.12621	0.130	0.0184	0.0316
Bit6:	0.347	0.0625	0.06323	0.0656	0.0187	0.0120
Bit7:	0.349	0.0313	0.3168	0.0317	0.0188	0.00779
Bit8:	0.348	0.0156	0.01588	0.0162	0.0202	0.00375
Bit9:	0.349	0.00781	0.00796	0.00794	0.0189	0.00205

Plots of a frequency response from a Monte Carlo run showed a second harmonic at -25.63dB and a third harmonic at -18.78dB.

5.4.4 Offset and gain error

Offset error is how well the actual transfer function matches with the ideal function in a single point. In an ideal converter the first transition point is 0.5 LSB above zero. Simulation shows that the first transition happens when the input current is 15.625nA. Equation 5.2[9] calculates the offset error.

$$I_{LSB} = \frac{I_{bias}}{2^9} = \frac{2\mu A}{2^9} = 3.90625nA \quad (5.1)$$

$$E_{offset} = \frac{I_{0...01}}{I_{LSB}} - \frac{1}{2}LSB = \frac{15.625nA}{3.90625nA} - \frac{1}{2}LSB = 3.5LSB \quad (5.2)$$

Gain error is how well the slope of the transfer function matches with the slope of the ideal function. For this ADC the gain error is calculated in equation 5.3[9].

$$\begin{aligned}
 E_{gain} &= \left(\frac{I_{1\dots1}}{I_{LSB}} - \frac{I_{0\dots01}}{I_{LSB}} \right) - (2^N - 2) \\
 &= \left(\frac{1.98828125\mu A}{3.90625nA} - \frac{15.625nA}{3.90625nA} \right) - (2^9 - 2) = -5LSB \quad (5.3)
 \end{aligned}$$

5.4.5 INL and DNL

INL is found by comparing an ideal straight line with the output from the SAR ADC when gain and offset errors are removed. DNL is found by using the standard histogram test. Results of the tests are given in figure 5.1 and 5.2.

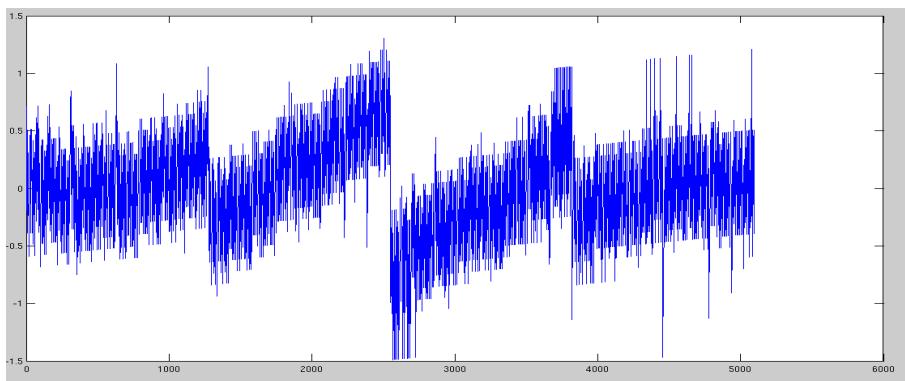


Figure 5.1: INL

5.4.6 Performance summary

Main performance characteristics to the ADC are summarized in table 5.5. The frequency used to simulate the harmonics is 20 MHz. The current draw is the ADCs with the comparator. The comparator uses $22 \mu A$ and the circuit uses $24 \mu A$. Ten in parallel and two biasing branches for the current mirror gives a total current consumption on $472 \mu A$.

The area is estimated based on the space used by the elements in the layout.

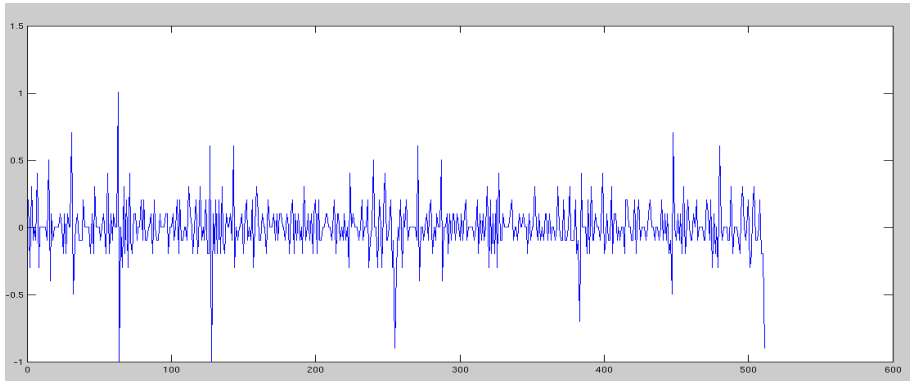


Figure 5.2: DNL

Table 5.5: Summary of performance

Technology	$0.18\mu\text{m}$
Supply voltage	1.8V
Current draw	$472\mu\text{A}$
Area	$490\text{n}m^2$
Sample rate	50 M[Hz/s]
DR	$0\text{[A]} \text{ to } 4\mu\text{[A]}$
Fundamental frequency	0.0 dB
Second harmonic	-72.1 dB
Third harmonic	-44.3 dB
INL	$1.25/-1.5$
DNL	$1.0/-1.0$
FOM	182.85
Input resistance	$22.84\text{ k}\Omega$

5.5 Layout

5.5.1 Comparator

The comparator has a behavior that is similar to the schematic simulation. Average current draw is $20\mu\text{A}$ with a $2\mu\text{A}$ bias current. A $2\mu\text{A}$ bias current will also make the circuit fast enough for a 50 MHz clock. One issue with the layout is that there is an offset between the two inputs. The output toggles when one input is 90 nA lower than the other.

5.5.2 Current ladder and input current mirror

All sources for the current ladder are simulated. The currents in the simulation are given in table 5.6. All currents have a deviation from ideal less than or about 1 %.

Simulation of the current draw from the circuit showed a constant current, except a few glitches when the circuit was toggling. A simulation of the current draw is given in figure F.1. The figure shows one current ladder toggling through one input current.

Table 5.6: Simulated currents with parasitics

Source:	Current	Deviation
Bias:	2.0002 μA	0.010%
MSB:	2.0001 μA	0.005%
b2:	1.0027 μA	0.270%
b3:	0.5025 μA	0.496%
b4:	0.2518 μA	0.712%
b5:	0.1261 μA	0.856%
b6:	63.1220 nA	0.995%
b7:	31.5860 nA	1.075%
b8:	15.7940 nA	1.082%
LSB:	7.9039 nA	1.169%

The input sampler was also simulated with a dc simulation. The result of the simulation is given in table 5.7.

Table 5.7: Input current mirror

Input current: $\mu\text{[A]}$	2	3	4	5	6
Output current: $\mu\text{[A]}$	2.017	3.015	4.016	5.017	6.020
Deviation:	0.85%	0.50%	0.40%	0.34%	0.33%

The ac simulation to the input current is given in talbe 5.8.

Table 5.8: Input current mirror

Input frequency: MHz	0	10	25	40	50
Output current: $\mu\text{[A]}$	1.99813	1.99368	1.97182	1.93445	1.90163
Deviation:	0.094%	0.316%	1.409%	3.277%	4.919%

Chapter 6

Discussion

In this chapter, all result from the previous chapter will be discussed as well as some design choices.

6.1 Calibration Circuit

The immediate drawback with this design is that it is difficult to say how long time it is going to use to calibrate the circuit before it has started. This is why the circuit has to be switched on for a long enough time for any initial condition. To minimize the number of times the calibration circuit has to be switched on the transistor which convert the output voltage into a current needs a low gm. A low gm will make sure that the current will not change when the output voltage have a small change.

6.2 Current mirrors

6.2.1 Sample and hold

The dc simulation showed a more or less constant current deviation. This deviation was about 20 to 30 nA. This can lead to a offset error in the circuit, but since the error is constant over the entire range, and the circuit is differential, will this error be cancelled out. Second harmonic in the frequency domain is counteracted by using a differential structure. For high input currents the error looks slightly higher. This can be the source to over harmonics since high currents will be clipped away in the ADC. Clipping is a source of over harmonics, and the circuit displays both second and third harmonic.

The ac simulation was simulated with a constant ac source on the input with an amplitude on $2 \mu\text{A}$. This showed a frequency dependency, and the amplitude decreased 35nA over the simulated region. The value went from a positive deviation to a negative deviation. This can be some of the explanation for why the circuit gets lower ENOB for higher frequencies.

The input current mirror is frequency dependent partly because of a dummy-switch is connected to the input. The present of the dummy switch lowers the pole frequency by introducing a higher capacitance on the input node.

6.3 Switch

The dummy switch worked excellent when it removed all charge injection produced by the switch. No trace of the charge injection was seen. The switch was also fast enough so that the circuit could settle within the specified time.

The normal switch also worked well when it isolated well when it was turned off, and shorted without issue when turned on. All parasitic introduced by the simple switch did not affect the circuits behavior.

6.4 SAR ADC

6.4.1 SNR

SNR is calculated as described in appendix A. From the calculations the SNR should be 56dB , while the simulation shows an SNR on 50dB . Simulation suggests that 3dB SNR is lost when the current sources are changed from ideal sources to non-ideal sources. This can be seen from table 5.4. The simulated values of the current sources deviate from the ideal values. 3dB SNR is assumed to be lost in the input mirrors because of uncertainty in sampling across amplitude and frequency.

6.4.2 ENOB

ENOB is decreasing over increasing input frequency. The only part of the design that can affect this is the input sampler since it is the only part that is frequency dependent to the input signal. The main reason for the degradation of ENOB is the increase of third harmonic with increasing frequency.

The fact that third harmonic is so large can be an indication that clipping is a problem in the sampler. Third harmonic in a differential circuit say that there is a different between the two branches. A high signal will be clipped and a low is not affected.

6.4.3 Monte Carlo

The current in all sources except the bias source showed an average value higher than the ideal value. This can be explained by assuming the current sources have been connected to V_{dd} in the simulation. All sources except the bias source have a switch at the top which switches between the sum node and V_{dd} . The square law model for a transistor in equation 3.6, suggests that the current will increase with a higher drain voltage. The result is presented here because the deviation is small. The drain voltage could be thought to be higher in general, but this will not explain why the bias current had an average current equal to the ideal current.

The Monte Carlo runs showed a high variation in the threshold voltages to the current sources which in turn showed a high variation in the currents. This large variation will cause the two branches of the ADC to be different. This will in turn result in high amount of over harmonics, both second and third. This big variation makes the circuit more or less useless, so calibration is necessary.

6.4.4 Offset and gain error

The offset error in the circuit is assumed to be connected with the offset error in the input sampler. Because of offset will the lowest input current be above zero. This means that the first digital value occurring in the circuit is 000000100 (4). This causes the offset error on 3.5 LSB.

The gain error is simulated to be -5LSB. This, like the offset error, comes from the offset voltage in the input sampler which will cause $I_{1...1}$ to occur with a lower input current.

6.4.5 INL and DNL

INL will ideally be between minus one half LSB to one half LSB. This ideal criteria is for the most part fulfilled except for the middle region. In figure 5.1 there is a discontinuity when MSB and MSB_{-1} are toggled. This is a clear indication that the largest current sources are not exactly matched to the sum of the rest of the current sources. This can also be seen in table 5.4. By summing up the values of the lowest current sources in the Sim. column they will not match with the value of MSB and MSB_{-1} . Equation 6.1 shows that the difference is about 1.23 LSB to the MSB. This calculation matches well with the simulation. What this means is that the circuit will jump from 011111110 (254) to 100000000 (256) since the first one will represent a higher input value.

$$MSB - \sum_{i=LSB}^{MSB_{-1}} = 1.99692\mu A - 2.00175\mu A = -4.83nA \simeq -1.23LSB. \quad (6.1)$$

DNL had an average value about zero as expected, but in some areas the value spiked. The spikes varied from minus one to one. An interesting observation is that these spikes appear the same places as the three MSB changes. This can be explained in a similar manner like INL. The circuit will completely or partially jump over some values because the sum of the lower current source will be greater than the largest source.

6.4.6 Performance summary

Several non-low power techniques were used in the design. For instance the WSCM. Another design aspect that increased current consumption is that the current sources were switches between vdd and the sum node. This means that the current sources that was not in use was switched to vdd. This is a waste of power, but it is done to ensure a constant current draw.

I_{bias} can in theory be $1\mu\text{A}$ since LSB will be as low as 1.95nA , which is higher than the lowest input current the comparator can sense. From [8] the lowest input current to the comparator can be 1.6nA . This bias current was raised to increase the speed of the circuit. It is clear from equation 2.6 that an increase in bias current will increase the pole frequency. This is also beneficial because it will give a higher DR and making it easier for the comparator to separate two different input currents. The trade-off is that the current consumption will increase.

After the design was changed from a single ended design to a differential design the power consumption doubled. The benefit by having differential design is that the second harmonic was reduced. With a single ended design second harmonic was an issue, but from table 5.5 it is clear that second harmonic is gone, and third harmonic poses now as the new concern.

6.5 Layout

6.5.1 Comparator

The layout simulation was compared with the simulation of the schematic. The current draw was about the same as the schematic design. Test for the comparator showed that there was an offset current in the comparator which caused the output to toggle at a lower current value than expected. This can come from parasitic in the layout that has its origin from a not 100% symmetrical layout, as can be seen in figure D.1. Effects like this and mismatch will reinforce the need for calibration.

6.5.2 Current ladder and input current mirror

Simulation of the current ladder showed that all current sources had the expected value close to the simulated values. The largest deviations were for the lowest current sources with a deviation on 1.16%. Small errors in the small current cells will play a bigger part in the total current produced by the cell, compared to small errors in big current cells.

Current draw was also as expected. Since every current source produced the correct value the current draw was expected. The only change in current draw was glitches when the circuit toggled. This was somehow expected when the drain voltage to the current cells was changed during switching and the current cell had to adjust to the change in environment. To counteract this can a resistor be placed from V_{dd} to the switch causing a smaller voltage differences between the two nodes the switch are connected to.

Simulation of the input sampling circuit to the current ladder showed a positive behavior. The offset current was smaller and more constant over input currents. The ac simulation on the other hand was more severe. The deviation between the input current and the sampled current increased faster with frequency. At 25MHz the error was 1.409% which is equivalent with SNR equal to 70dB. This input sampler will by all likelihood degrade ENOB at higher frequencies.

6.6 Comparison with others

This work is compared with other ADC's in table 6.1. All articles describe current-mode ADC's realized with different techniques like SAR ADC, pipeline and time interleaved SAR ADC. These articles present their results in either hertz or samples per second. Chapter 2.5 describes a FOM where the object is go get highest number. It is worth mentioning that this FOM does not take into account effects like DR, INL or DNL which is important characteristics in an ADC.

This work lies as the third best. The two works above has 8 bit instead of 9 as this work, and has much lower power consumption. The biggest different compared with these two is that the sampling frequency of this work is much larger.

There are two articles which describe a ADC with a higher sampling frequency than this work. These two articles also has a much higher power consumption, and because of that a lower FOM.

Table 6.1: Comparisson with others

Ref.	Process	Res.	f_s	Power norm.	Year	V_{DD} [V]	FOM
[10]	CMOS $3\mu m$	8	450 [kHz]	$500 \mu W$	1991	5	0.23
[6]	CMOS $0.18\mu m$	6	125 [kHz]	$6 \mu W$	2005	0.65	1.33
[17]	CMOS $0.18\mu m$	1-8	2 [MHz]	$560 nW$	2006	0.55	914.28
[19]	CMOS $0.18\mu m$	8	1 [MHz]	$400 nW$	2007	0.6	640
[21]	CMOS $0.13\mu m$	8	1 [kS/s]	$255 nW$	2005	1	1
[13]	CMOS $2.5\mu m$	8	10 [MHz]	$5 mW$	1998	5	0.512
[14]	CMOS $1.2\mu m$	8	50 [kS/s]	$0.34 mW$	2000	1	0.037
[16]	CMOS $3.0\mu m$	6	500 [kHz]	$78.35 mW$	1990	5	$408 \cdot 10^{-6}$
[25]	CMOS $0.25\mu m$	10	8.3 [MS/s]	$930 \mu W$	2008	5	9.13
[5]	CMOS $0.18 \mu m$	6	125 [kHz]	$24 \mu A$	2005	0.65	1.33
[11]	CMOS 65 nm	10	2.6 [GS/s]	$120 mW$	2011	1.2 - 1.6	22.18
This work	CMOS $0.18\mu m$	9	50 [MHz]	$236 \mu W$	2012	1.8	108.47

6.7 Future work

Future work with this thesis includes finishing calibration circuits and including them into the layout simulation. This will make it possible to simulate the entire system and calculating ENOB and other performance characteristics like INL and DNL for the layout.

Another aspect of the work which needs some work is the input sampler. This sampler needs a current mirror which is more power efficient and more linear over frequency and voltage.

Chapter 7

Conclusion

A fully differential current-mode SAR ADC operating at a sampling frequency of 50 MHz is designed. The ADC is designed with 9 bit and has an ENOB specification on 8.5 bit. At 90% of the maximum input frequency ENOB is 6.87 bit, but almost reaches specification at 2 MHz with an ENOB on 8.42 bit. Degradation in ENOB is mainly because the input sampling is not linear enough in frequency, and input current amplitude for 9 bit.

Mismatch simulation of the ADC illustrated the need for calibration. In addition to the variation in the current sources which will give wrong output, will mismatch also separate the two identical branches. Simulation showed that both second and third harmonic increased significantly. To counteract these difficulties calibration is needed. Both the comparator and the current sources need to be calibrated.

Layout of this circuit also showed that calibration is important since it ensure correct behavior. The comparator showed an offset current on the input and the current sources in the current ladder had a deviation from ideal values, especially the sources with the lowest values.

Comparison with other current-mode SAR ADC showed that this work is competitive with other current-mode ADC's out there. It has a sampling frequency which is higher than most and a current consumption which is lower than most.

Reasons for why current-mode design is not used more is unclear, but there have been published papers where simulated results differ from measured results, for example [15]. The author speculates that this has something to do with the high degree of matching needed in a current-mode circuit for them to work properly. Sub-threshold operation in several current sources gives a high degree of uncertainty in the current value.

Bibliography

- [1] http://en.wikipedia.org/wiki/binary_search_algorithm.
- [2] <http://www.ams.com/eng>.
- [3] <http://www.cadence.com/products/cic/pages/default.aspx>.
- [4] http://www.mentor.com/products/fv/advance_ms/.
- [5] S Sonkusale A Agarwal, Y B Kim. Low power current mode adc for cmos sensor ic. 2005.
- [6] S. Sonkusale A. Agarwal, Y.B. Kim. Low power current mode adc for cmos sensor ic. May 2005.
- [7] David J. Allstot. Current-mode analog signal processing. 1991.
- [8] B. Eilertsen. Design of low power analog circuits for miniaturized ultrasound cameras. December 19, 2011.
- [9] David Johns and Ken Martin. *Analog Integrated Circuit Design*. John Wiley & Sons, Inc, 1997.
- [10] Kim Soo-Won Kim Seong-Won. Current-mode cyclic adc for low power and high speed applications. May 1991.
- [11] Claudio Nani Athon Zanicopoulos Kostas Doris, Erwin Janssen and Gerard van der Weide. A 480 mw 2.6 gs/s 10b time-interleaved adc with 48.5 db snr up to nyquist in 65 nm cmos. 2011.
- [12] Marko Kosunen. Digital signal processing and digital-to-analog converters for wide-band transmitters. November 2006.
- [13] D. Stoppa L. Ravezzi and G.-F. Dalla Betta. Current-mode a/d converter. 1998.
- [14] Siamak Morteza pour and Edward K. F. Lee. A 1-v, 8-bit successive approximation adc in standard cmos process. 2000.

- [15] Y. Sugimoto N. Yoshii, K. Mizutani. A current-mode adc with current exchanging and averaging capabilities by switching the currents and calculating data in the digital domain. 2007.
- [16] David G. Nairn and C. Andre T. Salama. Current -mode algorithmic analog-to-digital converters. 1990.
- [17] Kris Iniewski Rafal Dlugosz. Flexible architecture of ultra-low-power current-mode interleaved successive approximation analog-to-digital converter for wireless sensor networks. 2006.
- [18] Kris Iniewski Rafal Dlugosz. Ultra low power current-mode algorithmic analog-to-digital converter implemented in 0.18 um cmos technology for wireless sensor network. 2006.
- [19] Kris Iniewski Rafal Dlugosz, Vincent Gaudet. Flexible ultra low power successive approximation analog-to-digital converter with asynchronous clock generator. 2007.
- [20] Reza Lotfi Salim Al-Ahdab and Wouter A. Serdijn. A 1-v 225-nw 1ks/s current successive approximation adc for pacemakers.
- [21] Reza Lotfi Salim Al-Ahdab and Wouter A. Serdijn. A 1-v 225-nw 1ks/s current successive approximation adc for pacemakers. 2005.
- [22] Eulalia Balestrieri Luca De Vito Steven J. Tilden Solomon Max Sergio Rapuano, Pasquale Daponte and Jerome Blair. Adc parameters and characteristics, part 6 in a series of tutorials in instrumentation and measurement. 2005.
- [23] Eric A. Vittoz. Micropower techniques. 1994.
- [24] Ye Xu. A low-offset dynamic comparator using bulk biasing technique in digital 65nm cmos technology. 2010.
- [25] Zheng Yang and Jan Van der Spiegel. A 10-bit 8.3ms/s switched-current successive approximation adc for column-parallel imagers. 2008.

Appendix A

Calculation of ENOB

The quantization noise is assumed to be white. With this assumption we can calculate the current noise.

$$I_{QRMS} = \sqrt{\int_{-\infty}^{\infty} x^2 * f(x) dx} = \sqrt{\frac{1}{\delta} \int_{-\delta/2}^{\delta/2} x^2 dx} = \frac{\delta}{\sqrt{12}} = \frac{I_{ref}}{2^N * \sqrt{12}} \quad (\text{A.1})$$

Where $f(x)$ is the probability distribution to the quantization noise. $f(x)$ is normally distributed between $-\delta/2$ and $\delta/2$, and have the amplitude $1/\delta$. δ is one LSB. We assume that the input signal is a sinusoid with a amplitude that goes from peak to peak.

$$I_{inn} = \frac{I_{ref}}{2} \sin(2\pi ft) \quad (\text{A.2})$$

RMS value to the input amplitude is given by:

$$I_{innRMS} = \frac{I_{ref}}{2\sqrt{2}} \quad (\text{A.3})$$

SNR is then given by:

$$SNR = \frac{I_{innRMS}}{I_{QRMS}} = \frac{\frac{I_{ref}}{2\sqrt{2}}}{\frac{I_{ref}}{2^N * \sqrt{12}}} = 2^N \sqrt{\frac{3}{2}} \quad (\text{A.4})$$

where N is the number of bits. SNR given in dB :

$$SNR = 20 \log(2^N \sqrt{\frac{3}{2}}) = 6.02N + 1.76 [dB] \quad (A.5)$$

Now we want to calculate ENOB. Using equation A.5 and exchanging SNR by SINAD and the number of bits with ENOB. We can write the equation for ENOB as:

$$ENOB = \frac{SINAD - 1.76}{6.02} [dB] \quad (A.6)$$

Appendix B

Input and output impedance to a simple current mirror

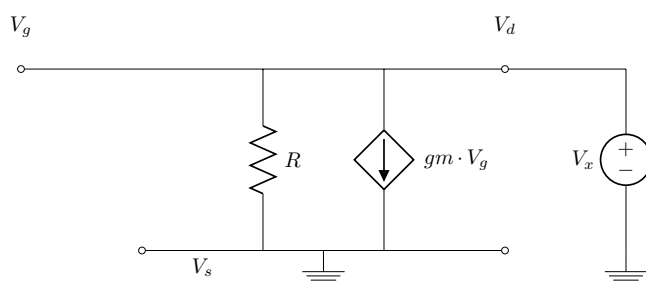


Figure B.1: Small signal model for the input transistor in a simple current mirror

To calculate input resistance a test source, V_x , is applied on the input. All other voltage sources are grounded, and all current sources are opened. First, the equation for the loop containing the test source and the small signal model:

$$V_x = i_x * R - gm * R * V_x$$

Rewritten:

$$V_x(1 + gm * R) = i_x * R$$

By defining R_{in} as the test voltage divided in the test current we can then calculate R_{in} as:

$$R_{in} = \frac{V_x}{i_x} = \frac{R}{1 + gm * R} \approx \frac{1}{gm}$$

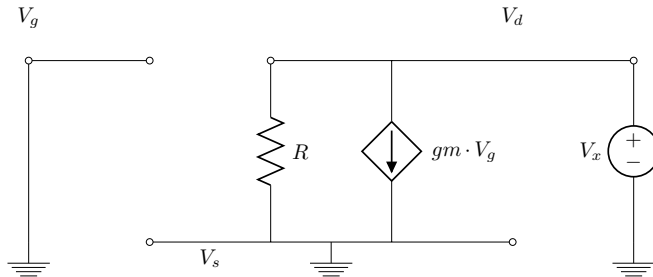


Figure B.2: Small signal model for the output transistor

Since V_g is grounded, the controlled current source has zero current through it. The equation is simply:

$$V_x = i_x * R$$

$$R_{out} = \frac{V_x}{i_x} = R \approx \frac{1}{g_{ds}}$$

Appendix C

Input and output impedance to a wide-swing current mirror

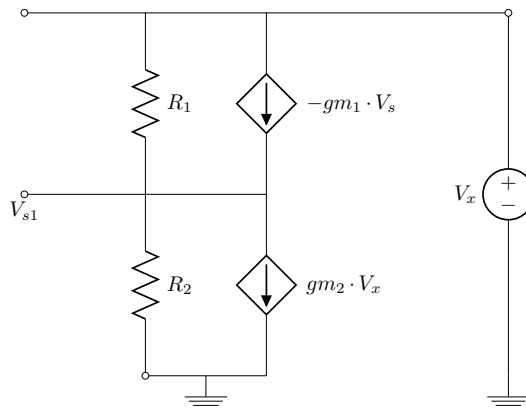


Figure C.1: Small signal model for the input transistor

An equation containing the test voltage V_x , and the small signal model is:

$$V_x = (gm_1 * V_{s1} + i_x)R_1 + (i_x - gm_2 * V_x)R_2$$

V_{s1} is calculated to be:

$$V_{s1} = (i_x + gm_2 * V_x)R_2$$

$$V_x = (gm_1 * (i_x + gm_2 * V_x)R_2 + i_x)R_1 + (i_x - gm_2 * V_x)R_2$$

$$V_x(1 + gm_1 * gm_2 * R_1 * R_2 + gm_2 * R_2) = i_x(gm_1 * R_1 * R_2 + R_1 + R_2)$$

By defining R_{in} as $\frac{V_x}{i_x}$ and simplifying the equation, the result can be put as:

$$R_{in} = \frac{V_x}{i_x} = \frac{gm_1 * R_1 * R_2 + R_1 + R_2}{1 + gm_1 * gm_2 * R_1 * R_2 + gm_2 * R_2} \approx \frac{1}{gm_2}$$

The input resistance to the wide-swing current mirror is given by $\frac{1}{gm_2}$

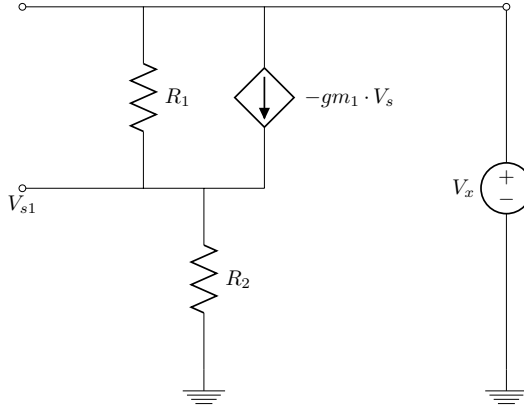


Figure C.2: Small signal model for the output transistor

$$V_x = (i_x - gm_1 * V_{s1})R_1 + i_x * R_2$$

$$V_{s1} = i_x * R_2$$

$$V_x = (i_x - gm_1 * i_x * R_2)R_1 + i_x * R_2 = i_x(R_1 + R_2 + R_1 * R_2 * gm)$$

$$R_{out} = \frac{V_x}{i_x} = R_1 + R_2 + R_1 * R_2 * gm \approx \underline{R_1 * R_2 * gm}$$

Output resistance to the wide-swing current mirror is $R_1 * R_2 * gm$.

Appendix D

Layout

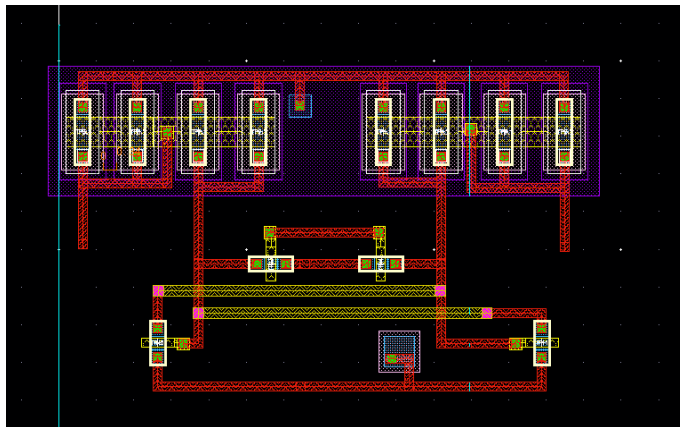


Figure D.1: Final layout of the comparator

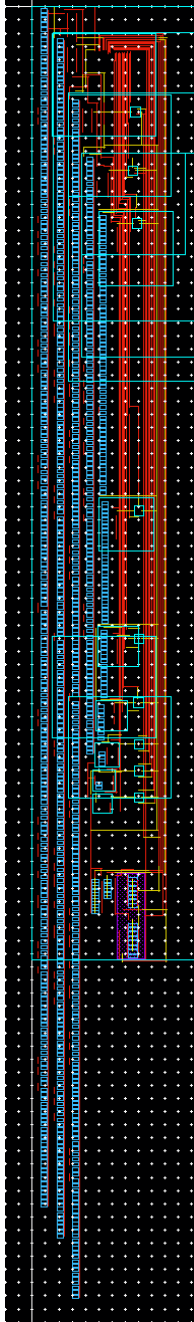


Figure D.2: Final layout of the current array

Appendix E

Calibration

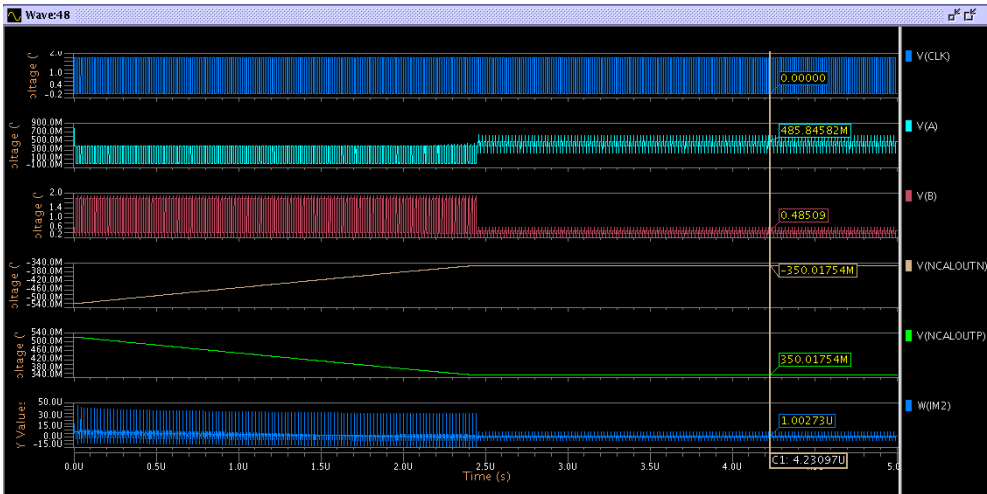


Figure E.1: A calibration run

Appendix F

Current draw

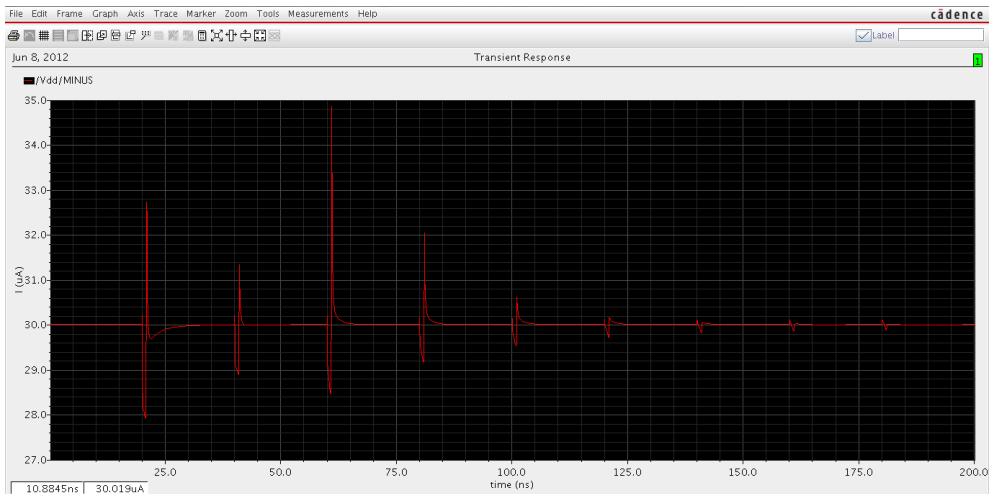


Figure F.1: Simulation of variation in current draw

Appendix G

SAR ADC

G.1 Top file

```
*****
*
*      Author: Baard Eilertsen
*
*
*      Date: 17.2.2012
*
*
*****

*****
* Include *
*****

.include comparator.cir
.include calibration_circuit_v02.cir
.include sar_adc.cir
.include insamp.cir
.include inverter.cir
.include nmos_switch.cir
.include nmos_switch_dummy.cir
.include pmos_switch.cir

.option ADMSMCDEVSEV=warning

.opt aex

.subckt mswitch a c z ron=switch_ron
r a z value={eval(v(c)>0.5?ron:1g)}
```

```

rc c 0 lt
.ends

.compat
.lib /eda/eldo_models/ams_hk.3.78/design.inc
.lib /eda/eldo_models/ams_hk.3.78/fixed_corner_lv.inc
    lvtt
.lib /eda/eldo_models/ams_hk.3.78/passive_corner_lv.inc
    lvrtt
.lib /eda/eldo_models/ams_hk.3.78/passive_corner_lv.inc
    lvctt
.lib /eda/eldo_models/ams_hk.3.78/passive_corner_lv.inc
    lvdtt
.lib /eda/eldo_models/ams_hk.3.78/process_lv.inc.dos
    lvstats
*.lib /eda/eldo_models/ams_hk.3.78/pfet.inc
*.lib /eda/eldo_models/ams_hk.3.78/nfet.inc
.lib ../sar_adc/fet/nfet.inc
.lib ../sar_adc/fet/pfet.inc
.endcompat

.subckt nfeti d g s b w=0 l=0 nf=1 gcon=1 rsx=50 dtemp=0 rf=0
    lstis=2 par=1
    .param
    *
    * some constant for mos
    + yc=0.52e-6 yo=0.43e-6 yn=0.26e-6 offside=0.06e-6
    *
    + nd='(floor((nf+1)/2))' ns='(floor(nf/2)+1)'
    + ndc='(nd-1+(ns-nd))' ndo='(1-(ns-nd))'
    + nsc='(ns-1-(ns-nd))' nso='(1+(ns-nd))'
    *
    + zad2='(1/nf*(ndc*(w/nf-offside)*yc+ndo*(w/nf-offside)*
        yo))'
    + zas2='(1/nf*(nsc*(w/nf-offside)*yc+nso*(w/nf-offside)*
        yo))'
    + zpd2='(1/nf*(nd*2*w/nf+2*(ndc*(yc-offside)+ndo*(yo-
        offside))))'
    + zps2='(1/nf*(ns*2*w/nf+2*(nsc*(yc-offside)+nso*(yo-
        offside))))'
    + znrd2='(yn/(w/nf-offside))'
    + znrs2='znrd2'
    *
    xnfet d g s b nfet l=1 w=w nf=nf ad=zad2 as=zas2 pd=zpd2
        ps=zps2 nrd=znrd2 nrs=znrs2
.ends

.subckt pfeti d g s b w=0 l=0 nf=1 gcon=1 rsx=50 dtemp=0 rf=0
    lstis=2 par=1
    .param

```



```

*
* some constant for mos
+ yc=0.52e-6 yo=0.43e-6 yn=0.26e-6 offside=0.06e-6
*
+ nd='(floor((nf+1)/2))' ns='(floor(nf/2)+1)'
+ ndc='(nd-1+(ns-nd))' ndo='(1-(ns-nd))'
+ nsc='(ns-1-(ns-nd))' nso='(1+(ns-nd))'
*
      + zad2='(1/nf*(ndc*(w/nf-offside)*yc+ndo*(w/nf-
      offside)*yo))'
+ zas2='(1/nf*(nsc*(w/nf-offside)*yc+nso*(w/nf-offside)*
  yo))'
+ zpd2='(1/nf*(nd*2*w/nf+2*(ndc*(yc-offside)+ndo*(yo-
  offside))))'
+ zps2='(1/nf*(ns*2*w/nf+2*(nsc*(yc-offside)+nso*(yo-
  offside))))'
+ znr2='(yn/(w/nf-offside))'
+ znrs2='znr2'
*
  xpfet d g s b pfet l=1 w=w nf=nf ad=zad2 as=zas2 pd=zpd2
    ps=zps2 nrd=znr2 nrs=znrs2
.ends

*****
* Parameters *
*****

.param vdd=1.8
.param vss=0
.param vmc=0.39
.param ibias=2u
.param wp=0.3u
.param lp=0.18u
.param wn=0.3u
.param ln=0.18u
.param switch_ion=100

*****
* Sources *
*****

vdd vdd vss vdd
vss vss 0 vss
vcm vcm vss vcm

*****
* Circuit *
*****

.model d2a_eldo d2a mode=bit

```

```

+vhi=vdd vlo=0 trise=50p tfall=50p
.defhook d2a_eldo

.model a2d_eldo a2d mode=bit
+vth='vdd/2' cin=10f
.defhook a2d_eldo

.model sar_adc_control(sar_adc_control) macro lang=vhdlams

yctrl sar_adc_control(sar_adc_control)
+ port: o11 c11 o12 c12 o13 c13 o14 c14 o15 c15 o16 c16 o17 c17
  o18 c18 o19 c19 samp1
+ o21 c21 o22 c22 o23 c23 o24 c24 o25 c25 o26 c26 o27 c27 o28
  c28 o29 c29 samp2
+ o31 c31 o32 c32 o33 c33 o34 c34 o35 c35 o36 c36 o37 c37 o38
  c38 o39 c39 samp3
+ o41 c41 o42 c42 o43 c43 o44 c44 o45 c45 o46 c46 o47 c47 o48
  c48 o49 c49 samp4
+ o51 c51 o52 c52 o53 c53 o54 c54 o55 c55 o56 c56 o57 c57 o58
  c58 o59 c59 samp5
+ o61 c61 o62 c62 o63 c63 o64 c64 o65 c65 o66 c66 o67 c67 o68
  c68 o69 c69 samp6
+ o71 c71 o72 c72 o73 c73 o74 c74 o75 c75 o76 c76 o77 c77 o78
  c78 o79 c79 samp7
+ o81 c81 o82 c82 o83 c83 o84 c84 o85 c85 o86 c86 o87 c87 o88
  c88 o89 c89 samp8
+ o91 c91 o92 c92 o93 c93 o94 c94 o95 c95 o96 c96 o97 c97 o98
  c98 o99 c99 samp9
+ o101 c101 o102 c102 o103 c103 o104 c104 o105 c105 o106 c106
  o107 c107 o108 c108 o109 c109 samp10
+ com1 com2 reset load1 load2 load3 load4 load5 load6 load7
  load8 load9 load10 rst clk
+ Aout1 Bout1 Aout2 Bout2 Aout3 Bout3 Aout4 Bout4 Aout5 Bout5
  Aout6 Bout6 Aout7 Bout7 Aout8 Bout8 Aout9 Bout9 Aout10
  Bout10

xcomp1 vdd vss clk vcm ipos1 ineg1 Aout1 Bout1 comparator
xcomp2 vdd vss clk vcm ipos2 ineg2 Aout2 Bout2 comparator
xcomp3 vdd vss clk vcm ipos3 ineg3 Aout3 Bout3 comparator
xcomp4 vdd vss clk vcm ipos4 ineg4 Aout4 Bout4 comparator
xcomp5 vdd vss clk vcm ipos5 ineg5 Aout5 Bout5 comparator
xcomp6 vdd vss clk vcm ipos6 ineg6 Aout6 Bout6 comparator
xcomp7 vdd vss clk vcm ipos7 ineg7 Aout7 Bout7 comparator
xcomp8 vdd vss clk vcm ipos8 ineg8 Aout8 Bout8 comparator
xcomp9 vdd vss clk vcm ipos9 ineg9 Aout9 Bout9 comparator
xcomp10 vdd vss clk vcm ipos10 ineg10 Aout10 Bout10 comparator

xsar_adc1 vdd vss s01 sd01 nb2 nb21 iin1 diin1 ipos1 ineg1 o11
  c11 o12 c12 o13 c13 o14 c14 o15 c15 o16 c16 o17 c17 o18 c18
  o19 c19 sar_adc

```

```

xsar_adc2 vdd vss s02 sd02 nb2 nb21 iin2 diin2 ipos2 ineg2 o21
  c21 o22 c22 o23 c23 o24 c24 o25 c25 o26 c26 o27 c27 o28 c28
  o29 c29 sar_adc
xsar_adc3 vdd vss s03 sd03 nb2 nb21 iin3 diin3 ipos3 ineg3 o31
  c31 o32 c32 o33 c33 o34 c34 o35 c35 o36 c36 o37 c37 o38 c38
  o39 c39 sar_adc
xsar_adc4 vdd vss s04 sd04 nb2 nb21 iin4 diin4 ipos4 ineg4 o41
  c41 o42 c42 o43 c43 o44 c44 o45 c45 o46 c46 o47 c47 o48 c48
  o49 c49 sar_adc
xsar_adc5 vdd vss s05 sd05 nb2 nb21 iin5 diin5 ipos5 ineg5 o51
  c51 o52 c52 o53 c53 o54 c54 o55 c55 o56 c56 o57 c57 o58 c58
  o59 c59 sar_adc
xsar_adc6 vdd vss s06 sd06 nb2 nb21 iin6 diin6 ipos6 ineg6 o61
  c61 o62 c62 o63 c63 o64 c64 o65 c65 o66 c66 o67 c67 o68 c68
  o69 c69 sar_adc
xsar_adc7 vdd vss s07 sd07 nb2 nb21 iin7 diin7 ipos7 ineg7 o71
  c71 o72 c72 o73 c73 o74 c74 o75 c75 o76 c76 o77 c77 o78 c78
  o79 c79 sar_adc
xsar_adc8 vdd vss s08 sd08 nb2 nb21 iin8 diin8 ipos8 ineg8 o81
  c81 o82 c82 o83 c83 o84 c84 o85 c85 o86 c86 o87 c87 o88 c88
  o89 c89 sar_adc
xsar_adc9 vdd vss s09 sd09 nb2 nb21 iin9 diin9 ipos9 ineg9 o91
  c91 o92 c92 o93 c93 o94 c94 o95 c95 o96 c96 o97 c97 o98 c98
  o99 c99 sar_adc
xsar_adc10 vdd vss s010 sd010 nb2 nb21 iin10 diin10 ipos10
  ineg10 o101 c101 o102 c102 o103 c103 o104 c104 o105 c105
  o106 c106 o107 c107 o108 c108 o109 c109 sar_adc

xinsamp vdd vss nb2 iin iind sampl samp2 samp3 samp4 samp5 samp6
  samp7 samp8 samp9 samp10 iin1 iin2 iin3 iin4 iin5 iin6 iin7
  iin8 iin9 iin10 diin1 diin2 diin3 diin4 diin5 diin6 diin7
  diin8 diin9 diin10 insampling

iin vdd iin sin('2*ibias ' '1*ibias ' 5.029296875meg 0 0 0)
iind vdd iind sin('2*ibias ' '1*ibias ' 5.029296875meg 0 0 180)

vclk clk vss pulse(vss vdd 0 1p 1p 10n 20n)
vrst rst vss pwl(0 vss 9n vss 10n vdd 29n vdd 30n vss 2000n vss)

*
*SAR ADC
*
xwideswingbias nb21 nb21 vdd vdd pfeti w='wp' l='5*lp' m=2
iwideswingbias nb21 vss '3*ibias '
*
*Sampler
*
xwideswingbias2 nb2 nb2 vss vss nfeti w='wn' l='8*ln' m=2
iwideswingbias2 vdd nb2 '3*ibias '

```

```
ipos21 vdd s01 ibias
ipos22 vdd s02 ibias
ipos23 vdd s03 ibias
ipos24 vdd s04 ibias
ipos25 vdd s05 ibias
ipos26 vdd s06 ibias
ipos27 vdd s07 ibias
ipos28 vdd s08 ibias
ipos29 vdd s09 ibias
ipos210 vdd s010 ibias
```

```
idpos21 vdd sd01 ibias
idpos22 vdd sd02 ibias
idpos23 vdd sd03 ibias
idpos24 vdd sd04 ibias
idpos25 vdd sd05 ibias
idpos26 vdd sd06 ibias
idpos27 vdd sd07 ibias
idpos28 vdd sd08 ibias
idpos29 vdd sd09 ibias
idpos210 vdd sd010 ibias
```

```
ineg1 ineg1 vss ibias
ineg2 ineg2 vss ibias
ineg3 ineg3 vss ibias
ineg4 ineg4 vss ibias
ineg5 ineg5 vss ibias
ineg6 ineg6 vss ibias
ineg7 ineg7 vss ibias
ineg8 ineg8 vss ibias
ineg9 ineg9 vss ibias
ineg10 ineg10 vss ibias
```

```
ipos1 ipos1 vss ibias
ipos2 ipos2 vss ibias
ipos3 ipos3 vss ibias
ipos4 ipos4 vss ibias
ipos5 ipos5 vss ibias
ipos6 ipos6 vss ibias
ipos7 ipos7 vss ibias
ipos8 ipos8 vss ibias
ipos9 ipos9 vss ibias
ipos10 ipos10 vss ibias
```

```
*****
* Simulation *
*****
```

```
.tran 1n '1*20.94u'
.probe v
```

```
.probe i
```

```
.mc 30 irun=1 nonon
```

G.2 SAR ADC

```
*****
*
*       Author: Baard Eilertsen
*
*
*       Date: 11.4.2012
*
*
*****

.subckt sar_adc vdd vss s0 sd0 sbias nb21 iin diin sum dsum o1
      c1 o2 c2 o3 c3 o4 c4 o5 c5 o6 c6 o7 c7 o8 c8 o9 c9

xm7 s0 s0 s00 vss nfeti w='wn' l='3*ln' m=4
xm77 s00 s00 vss vss nfeti w='wn' l='5*ln' m=256

xdm7 sd0 sd0 sd00 vss nfeti w='wn' l='3*ln' m=4
xdm77 sd00 sd00 vss vss nfeti w='wn' l='5*ln' m=256

xbias sum0 s0 nbias vss nfeti w='wn' l='3*ln' m=4
xbias2 nbias s00 vss vss nfeti w='wn' l='5*ln' m=256
xb1 s1 s0 s11 vss nfeti w='wn' l='3*ln' m=4
xb11 s11 s00 vss vss nfeti w='wn' l='5*ln' m=256
xb2 s2 s0 s22 vss nfeti w='wn' l='3*ln' m=4
xb22 s22 s00 vss vss nfeti w='wn' l='5*ln' m=128
xb3 s3 s0 s33 vss nfeti w='wn' l='3*ln' m=4
xb33 s33 s00 vss vss nfeti w='wn' l='5*ln' m=64
xb4 s4 s0 s44 vss nfeti w='wn' l='3*ln' m=4
xb44 s44 s00 vss vss nfeti w='wn' l='5*ln' m=32
xb5 s5 s0 s55 vss nfeti w='wn' l='3*ln' m=4
xb55 s55 s00 vss vss nfeti w='wn' l='5*ln' m=16
xb6 s6 s0 s66 vss nfeti w='wn' l='3*ln' m=4
xb66 s66 s00 vss vss nfeti w='wn' l='5*ln' m=8
xb7 s7 s0 s77 vss nfeti w='wn' l='3*ln' m=4
xb77 s77 s00 vss vss nfeti w='wn' l='5*ln' m=4
xb8 s8 s0 s88 vss nfeti w='wn' l='3*ln' m=4
xb88 s88 s00 vss vss nfeti w='wn' l='5*ln' m=2
xb9 s9 s0 s99 vss nfeti w='wn' l='3*ln' m=4
xb99 s99 s00 vss vss nfeti w='wn' l='5*ln' m=1

xdbias dsum0 sd0 ndbias vss nfeti w='wn' l='3*ln' m=4
xdbias2 ndbias sd00 vss vss nfeti w='wn' l='5*ln' m=256
xdb1 sd1 sd0 sd11 vss nfeti w='wn' l='3*ln' m=4
xdb11 sd11 sd00 vss vss nfeti w='wn' l='5*ln' m=256
xdb2 sd2 sd0 sd22 vss nfeti w='wn' l='3*ln' m=4
xdb22 sd22 sd00 vss vss nfeti w='wn' l='5*ln' m=128
xdb3 sd3 sd0 sd33 vss nfeti w='wn' l='3*ln' m=4
```

```

xdb33 sd33 sd00 vss vss nfeti w='wn' l='5*ln' m=64
xdb4 sd4 sd0 sd44 vss nfeti w='wn' l='3*ln' m=4
xdb44 sd44 sd00 vss vss nfeti w='wn' l='5*ln' m=32
xdb5 sd5 sd0 sd55 vss nfeti w='wn' l='3*ln' m=4
xdb55 sd55 sd00 vss vss nfeti w='wn' l='5*ln' m=16
xdb6 sd6 sd0 sd66 vss nfeti w='wn' l='3*ln' m=4
xdb66 sd66 sd00 vss vss nfeti w='wn' l='5*ln' m=8
xdb7 sd7 sd0 sd77 vss nfeti w='wn' l='3*ln' m=4
xdb77 sd77 sd00 vss vss nfeti w='wn' l='5*ln' m=4
xdb8 sd8 sd0 sd88 vss nfeti w='wn' l='3*ln' m=4
xdb88 sd88 sd00 vss vss nfeti w='wn' l='5*ln' m=2
xdb9 sd9 sd0 sd99 vss nfeti w='wn' l='3*ln' m=4
xdb99 sd99 sd00 vss vss nfeti w='wn' l='5*ln' m=1

xm9 nb nb21 net1 vdd pfeti w='wp' l='1.5*lp' m=6
xm10 sum nb21 net2 vdd pfeti w='wp' l='1.5*lp' m=6
xm99 net1 nb vdd vdd pfeti w='wp' l='6*lp' m=8
xm100 net2 nb vdd vdd pfeti w='wp' l='6*lp' m=8

xm12 nb sbias net3 vss nfeti w='wn' l='6.25*ln' m=6
xm122 net3 iin vss vss nfeti w='wn' l='6*ln' m=11

xdm9 ndb nb21 ndet1 vdd pfeti w='wp' l='1.5*lp' m=6
xdm10 dsum nb21 ndet2 vdd pfeti w='wp' l='1.5*lp' m=6
xdm99 ndet1 ndb vdd vdd pfeti w='wp' l='6*lp' m=8
xdm100 ndet2 ndb vdd vdd pfeti w='wp' l='6*lp' m=8

xdm12 ndb sbias ndet3 vss nfeti w='wn' l='6.25*ln' m=6
xdm122 ndet3 diin vss vss nfeti w='wn' l='6*ln' m=11

xtgabias vdd vss sum vdd sum0 nswitch
xdtgabias vdd vss dsum vdd dsum0 nswitch

xtga1 vdd vss sum o1 s1 nswitch m=1
xtgb1 vdd vss vdd c1 s1 nswitch m=1
xtga2 vdd vss sum o2 s2 nswitch m=1
xtgb2 vdd vss vdd c2 s2 nswitch m=1
xtga3 vdd vss sum o3 s3 nswitch m=1
xtgb3 vdd vss vdd c3 s3 nswitch m=1
xtga4 vdd vss sum o4 s4 nswitch m=1
xtgb4 vdd vss vdd c4 s4 nswitch m=1
xtga5 vdd vss sum o5 s5 nswitch m=1
xtgb5 vdd vss vdd c5 s5 nswitch m=1
xtga6 vdd vss sum o6 s6 nswitch m=1
xtgb6 vdd vss vdd c6 s6 nswitch m=1
xtga7 vdd vss sum o7 s7 nswitch m=1
xtgb7 vdd vss vdd c7 s7 nswitch m=1
xtga8 vdd vss sum o8 s8 nswitch m=1
xtgb8 vdd vss vdd c8 s8 nswitch m=1
xtga9 vdd vss sum o9 s9 nswitch m=1

```

```
xtgb9 vdd vss vdd c9 s9 nswitch m=1

xdtga1 vdd vss dsum c1 sd1 nswitch m=1
xdtgb1 vdd vss vdd o1 sd1 nswitch m=1
xdtga2 vdd vss dsum c2 sd2 nswitch m=1
xdtgb2 vdd vss vdd o2 sd2 nswitch m=1
xdtga3 vdd vss dsum c3 sd3 nswitch m=1
xdtgb3 vdd vss vdd o3 sd3 nswitch m=1
xdtga4 vdd vss dsum c4 sd4 nswitch m=1
xdtgb4 vdd vss vdd o4 sd4 nswitch m=1
xdtga5 vdd vss dsum c5 sd5 nswitch m=1
xdtgb5 vdd vss vdd o5 sd5 nswitch m=1
xdtga6 vdd vss dsum c6 sd6 nswitch m=1
xdtgb6 vdd vss vdd o6 sd6 nswitch m=1
xdtga7 vdd vss dsum c7 sd7 nswitch m=1
xdtgb7 vdd vss vdd o7 sd7 nswitch m=1
xdtga8 vdd vss dsum c8 sd8 nswitch m=1
xdtgb8 vdd vss vdd o8 sd8 nswitch m=1
xdtga9 vdd vss dsum c9 sd9 nswitch m=1
xdtgb9 vdd vss vdd o9 sd9 nswitch m=1

.ends
```


G.3 Input sampler

```

*****
*
*       Author: Baard Eilertsen
*
*
*       Date: 11.4.2012
*
*
*****

.subckt insampling vdd vss nb2 input inputd sampsar1 sampsar2
    sampsar3 sampsar4 sampsar5 sampsar6 sampsar7 sampsar8
    sampsar9 sampsar10 insar1 insar2 insar3 insar4 insar5 insar6
    insar7 insar8 insar9 insar10 dinsar1 dinsar2 dinsar3
    dinsar4 dinsar5 dinsar6 dinsar7 dinsar8 dinsar9 dinsar10

*****
* Circuit
*****

xm9 input nb2 net01 vss nfeti w='wn' l='6.25*ln' m=6
xm99 net01 input vss vss nfeti w='wn' l='6*ln' m=11

xdm9 inputd nb2 ndet01 vss nfeti w='wn' l='6.25*ln' m=6
xdm99 ndet01 inputd vss vss nfeti w='wn' l='6*ln' m=11

xn1 vdd vss input sampsar1 insar1 nswitch_dummy
xn2 vdd vss input sampsar2 insar2 nswitch_dummy
xn3 vdd vss input sampsar3 insar3 nswitch_dummy
xn4 vdd vss input sampsar4 insar4 nswitch_dummy
xn5 vdd vss input sampsar5 insar5 nswitch_dummy
xn6 vdd vss input sampsar6 insar6 nswitch_dummy
xn7 vdd vss input sampsar7 insar7 nswitch_dummy
xn8 vdd vss input sampsar8 insar8 nswitch_dummy
xn9 vdd vss input sampsar9 insar9 nswitch_dummy
xn10 vdd vss input sampsar10 insar10 nswitch_dummy

xdn1 vdd vss inputd sampsar1 dinsar1 nswitch_dummy
xdn2 vdd vss inputd sampsar2 dinsar2 nswitch_dummy
xdn3 vdd vss inputd sampsar3 dinsar3 nswitch_dummy
xdn4 vdd vss inputd sampsar4 dinsar4 nswitch_dummy
xdn5 vdd vss inputd sampsar5 dinsar5 nswitch_dummy
xdn6 vdd vss inputd sampsar6 dinsar6 nswitch_dummy
xdn7 vdd vss inputd sampsar7 dinsar7 nswitch_dummy
xdn8 vdd vss inputd sampsar8 dinsar8 nswitch_dummy
xdn9 vdd vss inputd sampsar9 dinsar9 nswitch_dummy
xdn10 vdd vss inputd sampsar10 dinsar10 nswitch_dummy

```

.ends

G.5 NMOS switch

```
*****
*
*      Author: Baard Eilertsen      *
*
*
*      Date: 04.03.2012            *
*
*
*****

.subckt nswitch vdd vss iin oc iout

xnmMos iin oc iout vss nfeti w='wn' l='ln' m=1

.ends
```

G.6 NMOS switch with dummy transistor

```
*****
*
*      Author: Baard Eilertsen      *
*
*
*      Date: 18.03.2012            *
*
*
*****

.subckt nswitch_dummy vdd vss iin oc iout

xnmMos iin oc iout vss nfeti w='wn' l='ln' m=2

xinV vdd vss oc noc inverter
xmdummy iout noc iout vss nfeti w='wn' l='ln' m=1

.ends
```

G.7 Inverter

```
*****
*
*      Author: Baard Eilertsen      *
*
*
*      Date: 04.03.2012             *
*
*
*****

.subckt inverter vdd vss vin vout

xnmMos vout vin vss vss nfeti w='wn' l='ln' m=1
xpmos vout vin vdd vdd pfeti w='wn' l='ln' m=1

.ends
```

Appendix H

MATLAB

H.1 Top code

```
clear all;
clc;

%Data from spice simulation
data = bin2dec(num2str(load('output.dat')));

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

%Remove the mean value(DC) from data
data=data-mean(data);

%Set analysis parameters
fs = 50e6; %Sampling frequency
fsa = 512; %Full scale amplitude: Amplitude
            which the spectrum shall be normalized to
window = true; %Shall the function be windowed
nb = 1; %Normalized baseband: Between 0 and
         1, how large are of the spectrum is used in the calculation

%Perform analysis of the signal
dyn_param(data, fsa, window, nb, fs);
nb = 1; %Normalized baseband: Between 0 and
```

1, how large are of the spectrum is used in the calculation

H.2 Script

```
function dyn_param(sequence, fullscale_amplitude, window,
    normalized_baseband, sampling_frequency)
%Rename to shorter names to ease code reading
seq = sequence;
fsa = fullscale_amplitude;
nb = normalized_baseband;
Fs = sampling_frequency;

%Calc length of signal
L = length(seq);

%Hanning window
if window == true
    win=hann(L);
    seq=seq.*win;
    wgain=sum(win)/L;
    seq=seq/wgain;
end

%Decide how much of the signal we want to plot
f = Fs*nb/2*linspace(0,1,(L/2)*nb);

%Take the fourier transform
Y_fft = fft(seq)/(L*fsa);

seq_fft=2*(abs(Y_fft(1:(L/2)*nb)));

seq_fft_db = 20*log10(seq_fft);

seq_fft(1)=0;
seq_fft(end)=0;

%Find the maximum value and its index in sequence
[max_value, index_value]=max(seq_fft)

%Store the values of the signal and harmonics and then remove
them from fft.
hd2_bin = mod((index_value-1).*2,L);
if hd2_bin > L/2
    hd2_bin = L - hd2_bin;
elseif hd2_bin == 0
    hd2_bin = hd2_bin + 1;
end
hd2_bin = hd2_bin + 1;
```

```

hd3_bin = mod((index_value-1).*3,L);
if hd3_bin > L/2
    hd3_bin = L - hd3_bin;
elseif hd3_bin == 0
    hd3_bin = hd3_bin + 1;
end
hd3_bin = hd3_bin + 1;

if window
    seq=seq_fft(1:index_value+1);
    seq_fft(index_value-1:index_value+1)=0;
    %Find the spur within the seq_fft
    spur=max(seq_fft);

    %Do the same for the seconds and third harmonic
    hd2=seq_fft(hd2_bin-1:hd2_bin+1);
    hd3=seq_fft(hd3_bin-1:hd3_bin+1);
    seq_fft(hd2_bin-1:hd2_bin+1)=0;
    seq_fft(hd3_bin-1:hd3_bin+1)=0;
else
    seq=seq_fft(index_value);
    seq_fft(index_value)=0;

    %Find the spur within the seq_fft
    spur=max(seq_fft);

    %Do the same for the seconds and third harmonic
    hd2=seq_fft(hd2_bin);
    hd3=seq_fft(hd3_bin);
    seq_fft(hd2_bin)=0;
    seq_fft(hd3_bin)=0;
end
%Fix DC offset
seq_fft(1:4)=0;

%The noise has been extracted from the original signal and can
    be used in
%calc
noise=seq_fft;

%Find the power of the different signals
Ps=sum(seq.^2);
Pn=sum(noise.^2);
Phd2=sum(hd2.^2);
Phd3=sum(hd3.^2);

%Calculate signal to noise ratio: SNR
SNR=10*log10(Ps/Pn)

%Calculate signal to noise and distortion ratio: SINAD

```

```

SINAD=10*log10(Ps/(Pn+Phd2+Phd3))

%Calculate effective number of bits: ENOB
ENOB=(SINAD-1.76)/6.02

%Calculate spurious free dynamic range: SFDR
SFDR=10*log10(Ps/spur^2)

%Calculate the total harmonic distortion: THD
%OLD: THD=10*log10(Ps/(Phd2+Phd3))
THD=10*log10(1/(Phd2+Phd3)) %Normalisert med Ps

%Print out second harmonic
%OLD: HD2=hd2
HD2=10*log10(Phd2/Ps) %Normalisert med Ps samt fikset dB. Gjorde
    det samme med HD3.

%print out third harmonic
HD3=10*log10(Phd3/Ps)

%Plot
figure(1)
plot(f, seq_fft_db)
title({'Discrete FFT of a ADC signal.'; ['SNR=', num2str(SNR), ', '
    SINAD=', num2str(SINAD), ', ENOB=', num2str(ENOB)]})
xlabel('Frequency (Hz)')
ylabel('Magnitude(Seq(f))')

end

```