



Norwegian University of  
Science and Technology

# A sub- $1\mu\text{W}$ , 16kHz Current-Mode SAR- ADC for Neural Spike Recording

**Bård Haaheim**

Master of Science in Electronics

Submission date: August 2011

Supervisor: Trond Ytterdal, IET

Co-supervisor: Timothy Constandinou, Imperial College



# Problem description

The candidate is to develop a ultra-low-power analogue-to-digital converter (ADC) for single channel neuron spike recording using a commercially-available  $0.18\mu m$  CMOS technology. A key requirement is that the circuit occupies a compact silicon area to allow for scalability within an array. Specifically, it is envisaged that the analogue front-end will be confined to the footprint of a neural probe, which must be scalable up to 100's of channels.

For this project, the candidate must become familiar with the field of neurotechnology and specifically regarding neuron spike recording. A good understanding to the dynamics of neural signals, and challenges brought on by the electrode-electrolyte interface would give good insight into defining the ADC specifications and in revealing potential improvements.

Both the choice of topology and implementation are left to the candidate, although it is required to be Nyquist-rate to allow for an asynchronous system architecture. A successful outcome would have a performance comparable to the state-of-the-art with the design tolerant to process variation to achieve a  $3\sigma$  production yield. A full-custom layout will be submitted for fabrication in the target technology.



# Preface

In Autumn 2010 I was invited to work at Imperial College under Dr. Timothy Constandinou as a part of a research team aiming to develop an ultra low power implantable platform for the next generation neural interfaces. Recent advances in technology combined with a ground breaking concept for neural spike sorting on-chip was the foundation of this project motivated by the need of providing easy to use neuralgic tools paving the way for new treatments of conditions currently suffering from extensive clinical burdens. The project, spanning over several years, was in an initial phase emphasizing experimental design and creative thinking my participation concerned and has therefore dynamically changed while it has progressed. My work omitted the front-end analogue to digital converter started as the project of course TFE4540 at NTNU. For convenience, the main content of this project has been incorporated into this paper [22].



# Acknowledgements

I would like to direct my sincere thanks to Dr. Timothy Constandinou for all the encouragement and useful guidance through the project, but also for enabling this cooperation between colleges and letting me take part in his research team. Further I would like to thank the neural team consisting of Dr. Amir Eftekhar, Song Luan, Sivylla Paraskevopoulou and Ian Williams for the feedback, ideas and moral support. A special thanks goes to Song Luan for his many helps and discussions.

Also, I would like to thank my professor and advisor at NTNU, Trond Ytterdal for all guidance and past-on wisdom, but also for being extremely helpful and flexible, using his network to provide me a project abroad.

Last, but not least, I would like to thank my office sharers at NTNU, Daniel Aasbø, Torbjørn Løvseth Finnøy, Carolina Fiorella Inche Velezmoro, Kjetil Kvalø and Marius Lind Volstad for fun times, good discussions and moral support over the last years.



# Abstract

This thesis presents an ultra-low-power 8-bit asynchronous current-mode successive approximation (SAR) ADC for single channel neuron spike recording. The novel design exploits current mode operating in weak inversion for high power efficiency and is design to operate at a  $1.8V$  supply. The ADC is running at a  $16kHz$  sampling frequency using under  $1\mu W$  of power, though is adjustable using the featured calibration registers. A finished layout is presented, occupying less than  $0.078mm^2$ . Linear operation through mismatch and process variations is obtained using a current calibration circuit connected to both the current mode DAC and all the biases. This ensures  $INL < \pm 0.5$  and  $DNL < \pm 1$ , yielding no missing codes and a  $3\sigma$  production yield. Calibration is needed because of the relatively large mismatch caused by sub-threshold operation of the current mirrors. The design also offers a newly developed current comparator with high resolution and fast settling relative to the current level and is completable with other state-of-the-art solutions, though still feature some voltage scaling issues left for future work.



# Contents

<b>Abstract</b>	<b>VII</b>
<b>Contents</b>	<b>XI</b>
<b>List of Figures</b>	<b>XIV</b>
<b>List of Tables</b>	<b>XV</b>
<b>Acronyms</b>	<b>XVII</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Neural Science and ADC requirements</b>	<b>3</b>
2.1 Neuron Signals . . . . .	3
2.2 The next generation neuron interfaces . . . . .	5
2.3 Limitations . . . . .	5
2.4 This project . . . . .	6
2.5 ADC requirements . . . . .	7
<b>3 Background</b>	<b>9</b>
3.1 Topology . . . . .	9
3.2 Current Mirror . . . . .	11
3.2.1 Topologies . . . . .	11
3.2.2 Noise . . . . .	12
3.2.3 Frequency response . . . . .	12
3.2.4 Mismatch . . . . .	13
3.2.5 Absolute Accuracy and dynamic range . . . . .	15
3.3 Converter Performance Characteristics . . . . .	17
3.4 Current Comparator . . . . .	17
3.4.1 Implementations . . . . .	18
3.5 Current Mode DAC . . . . .	21
3.5.1 Performance Limitations . . . . .	23

<b>4</b>	<b>Design and Methods</b>	<b>25</b>
4.1	Estimations . . . . .	25
4.2	Current Mirror . . . . .	26
4.3	Current Mode DAC . . . . .	26
4.4	Current Comparator . . . . .	29
4.5	Calibration . . . . .	29
4.6	Logic . . . . .	30
<b>5</b>	<b>Implementation</b>	<b>31</b>
5.1	Control Logic . . . . .	31
5.2	Current Comparator . . . . .	34
5.3	Current-Mode DAC . . . . .	36
5.4	Calibration Module . . . . .	37
5.5	Calibration register . . . . .	39
5.6	Input current mirror . . . . .	40
<b>6</b>	<b>Layout</b>	<b>43</b>
<b>7</b>	<b>Results</b>	<b>47</b>
7.1	ADC Core . . . . .	47
7.2	Register . . . . .	49
7.3	Calibration Module . . . . .	50
7.4	iDAC . . . . .	53
7.5	iComp . . . . .	53
7.6	Logic . . . . .	54
<b>8</b>	<b>Analysis and Discussion</b>	<b>57</b>
8.1	Topology . . . . .	58
8.2	Architecture . . . . .	58
8.2.1	Current mode DAC . . . . .	58
8.2.2	Current Comparator . . . . .	59
8.2.3	Logic . . . . .	60
8.2.4	Calibration . . . . .	61
8.2.5	Calibration Register . . . . .	62
8.3	ADC core . . . . .	62
8.4	Future Works . . . . .	63
<b>9</b>	<b>Conclusions</b>	<b>65</b>
<b>A</b>	<b>Derivations</b>	<b>73</b>
A.1	DAC mismatch requirements . . . . .	73

<b>B Ocean Scripts</b>	<b>75</b>
B.1 Current Mirror . . . . .	75
B.2 ADC Core . . . . .	76
B.3 Calibration Register . . . . .	78
B.4 iDAC . . . . .	81
<b>C Matlab Code</b>	<b>83</b>
C.1 iDAC . . . . .	83
C.2 ADC Core . . . . .	84
C.3 iComp input impedance . . . . .	85
<b>D Pin overview</b>	<b>87</b>
<b>E Measurements</b>	<b>89</b>
E.1 Current Mirror . . . . .	89
E.1.1 DR . . . . .	91
E.1.2 AC . . . . .	92
E.1.3 Noise . . . . .	93
E.1.4 Leakage . . . . .	94
E.1.5 MC . . . . .	95



# List of Figures

2.1	Neural recording from a cat motor cortex using Utah Electrode Array and CMOS amplifier. . . . .	4
2.2	The 100-electrode silicon-based Utah Electrode array measuring $4 \times 4 \times 1.5 \text{mm}^3$ . . . . .	5
2.3	System overview . . . . .	7
3.1	Comparison of recent published ADCs in performance of interest	10
3.2	General SAR ADC system overview. . . . .	11
3.3	SCM with compensation resistor for bandwidth enhancement .	13
3.4	Principle of current comparison . . . . .	18
3.5	Multilevel input impedance current comparator . . . . .	20
3.6	Common implementation of a current mode DAC . . . . .	22
3.7	Current splitting technique I . . . . .	23
3.8	Current splitting technique II . . . . .	23
4.1	SCM Mismatch - 3D model . . . . .	27
4.2	SCM Mismatch - Side projection in log-scale . . . . .	28
5.1	System level design for the current mode SAR ADC. . . . .	32
5.2	Digital circuit used to realize the SAR algorithm and output registers. . . . .	33
5.3	Impulse Generator . . . . .	34
5.4	Threshold detecting amplifier . . . . .	35
5.5	Current Comparator . . . . .	36
5.6	Current Comparator . . . . .	37
5.7	Bias Generator . . . . .	38
5.8	Calibration circuit for MSB . . . . .	38
5.9	Calibration circuit for bit 2-6, though all with different bias-voltages applied . . . . .	39
5.10	calibration register . . . . .	40
5.11	Current input/output stage . . . . .	41

6.1	Final Layout . . . . .	44
7.1	Input/output characteristics of the ADC . . . . .	48
7.2	INL of the ADC . . . . .	49
7.3	DNL of the ADC . . . . .	49
7.4	Input impedance of the current comparator . . . . .	55
E.1	The derivative of DC simulations for CMs at input currents from 1pA to 1 $\mu$ A. . . . .	91
E.2	AC simulations of CMs for currents in range of 0.1nA to 1nA . . . . .	92
E.3	Noise simulations for CMs . . . . .	93
E.4	Leakage simulations of MCs . . . . .	94

# List of Tables

2.1	ADC system requirements . . . . .	8
3.1	Output resistance of the four most common current mirrors . .	16
5.1	Register overview . . . . .	40
6.1	Layout explanation . . . . .	45
7.1	Power Consumption . . . . .	50
7.2	Selected sweep parameters and their values for corner analysis	50
7.3	Static power consumption of the register module . . . . .	50
7.4	Mismatch and process deviation of the reference currents . . .	51
7.5	Calibration current specifications . . . . .	52
7.6	Performance summary of the iDAC-module . . . . .	53
7.7	logistic regression parameters . . . . .	54
7.8	Mismatch and process deviations of the iComp-module . . . .	54
7.9	Current comparator specifications . . . . .	54
7.10	Mismatch and process deviations of the ImpGen-module . . . .	55
8.1	Comparison of low power ADCs . . . . .	57
A.1	Assigned error to each iDAC branch . . . . .	73
D.1	ADC pin overview . . . . .	88
E.1	Summary of simulation results of the different topologies at scaling 1:1 at MSB . . . . .	95



# Acronyms

<b>ADC</b>	analogue-to-digital converter
<b>CCM</b>	cascoded current mirror
<b>CMOS</b>	complementary metal-oxide-semiconductor
<b>DAC</b>	digital-to-analogue converter
<b>DNL</b>	differential nonlinearity
<b>DR</b>	dynamic range
<b>DSP</b>	digital signal processor
<b>ENOB</b>	effective number of bits
<b>EEPROM</b>	Electrically erasable programmable read-only memory
<b>FOM</b>	figure of merit
<b>INL</b>	integral nonlinearity
<b>IWCM</b>	Improved Wilson current mirror
<b>KCL</b>	Kirchhoff's current law
<b>LNA</b>	low noise amplifier
<b>LPF</b>	local field potential
<b>MOSFET</b>	metal-oxide-semiconductor field-effect transistor
<b>OPAMP</b>	Operational amplifier
<b>OTA</b>	operational transconductance amplifier
<b>PGA</b>	programmable gain amplifier

<b>PTAT</b>	proportional to absolute temperature
<b>SNR</b>	signal-to-noise ratio
<b>SAR</b>	successive approximation
<b>SCM</b>	simple current mirror
<b>TDA</b>	threshold detecting amplifier
<b>WCM</b>	Wilson current mirror

# Chapter 1

## Introduction

With ever-advancing trends in microelectronics, researchers are now innovating analytical tools for the in-vivo measurements of neuron activity. They are developing next generation neural interfaces for totally-implantable devices that use bio-telemetries for the transmission of both power and data. Such links require digitised signals for efficient and reliable data transmission therefore introducing the need for on-chip analogue-to-digital converters (ADCs). Traditionally, such systems use single, multiplexed ADCs that are time-shared [7, 24, 41], whereas this thesis offers single-channel implementations allowing for an efficient and easily-scalable system. These solutions are also commonly using charge redistribution voltage-mode successive approximation (SAR) ADCs despite current mode converters are now demonstrating excellent characteristics and in particular, high resource efficiency (power and area) [1, 2, 17]. This thesis presents an ultra low power 8-bit asynchronous current-mode SAR ADC implemented in a commercially-available  $0.18\mu\text{m}$  CMOS technology intended for neural spike recording. Exploiting current mode, the requirement of small area is ensured featuring  $16\text{kHz}$  sampling frequency.

The paper is organized as follows: Chapter 2 explains the challenges and the nature of neural signal recording together with the conceptual technique of on-chip neural spike sorting. Chapter 3 covers the theory behind applied techniques and design, while chapter 4 describes the design process. Chapter 5 describes the final implementation while chapter 6 briefly presents the final layout. Chapter 7 presents all simulated data and obtained results, while chapter 8 makes an analysis and discussion of the project. Chapter 9 concludes the report, summarizing the whole project.



# Chapter 2

## Neural Science and ADC requirements

Neural spike recording is not a new scientific area, but has existed over half a century. Already conducted experiments yield profound insights into brain function and how this complex system is affected by neurological injuries and diseases while recent years research have brought systems like Deep Brain Stimulator and Cochlear Implants helping thousands of patients worldwide [37, 45]. However, despite considerable advances in electrode technologies, the ability to interface digital microelectronics with the brain at the level of individual neurons is at present severely limited. To be able to mimic complex brain functions like moving limbs or preventing seizures, individual neuron recording and stimulation in numbers of hundreds are required. This aims to make new therapeutic avenues available to neurologists battling until now untreatable conditions among Huntington's, Parkinson's and Alzheimer's disease, spinal cord injuries and epilepsy to name a few. To better understand the design challenges regarding neuron spike recording a brief presentation of signal characteristics and recording methods will be explained in the following.

### 2.1 Neuron Signals

The analogue voltage potential of a neuron exhibits larger fluctuations whenever the neuron fires a signal. This manifests as a voltage pulse usually in

range of  $50\text{--}200\mu\text{V}$  over a  $1\text{--}5\mu\text{s}$  period of time. Each neuron has its own signature within the  $300\text{Hz} - 5\text{kHz}$  signal range, though the noisy background makes them hard to distinguish. A typical signal is shown in figure 2.1 and is showing three spikes at  $\sim 100\mu\text{s}$ ,  $\sim 140\mu\text{s}$  and  $\sim 180\mu\text{s}$  respectively [25]. Because the neurons are in close proximity of each other surrounded in dielectric material, the DC potential is correlated to the global brain activity resulting in unpredictable variations in frequencies below  $300\text{Hz}$ , also known as local field potentials (LFPs). This can also be seen in the figure. Figure 2.2 on the next page is showing the recording probe array. The LFPs are easily detected by commercially available technologies and contain information of general activity in regions of the brain. Understanding and interacting with the neuron spikes on the other hand, though unclear, scientists widely believe that it will give access to complete control of the body, though still severely limited due to its complexity [19].

At the present, arrays of electronic probes inserted into the brain tissue have no guaranty to the proximity to the actual neurons, making single probes (referred to as a channel,) picking up signals from several neurons. The average is  $3.4 \pm 1.5$  per channel. This severely increases the complexity of systems intended for neuron spike detection as a single probe needs to be able to distinguish several neurons from each other to detect activity. The interval of which a neuron fires is also changing. While the average is around  $50\text{Hz}$ , burst up at  $300\text{Hz}$  are possible.

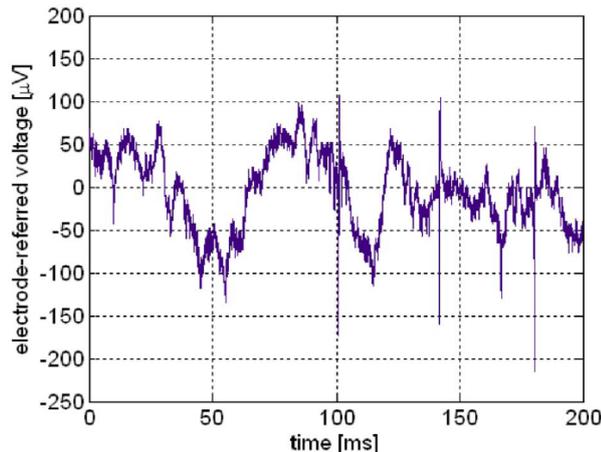


Figure 2.1: Neural recording from a cat motor cortex using Utah Electrode Array and CMOS amplifier.

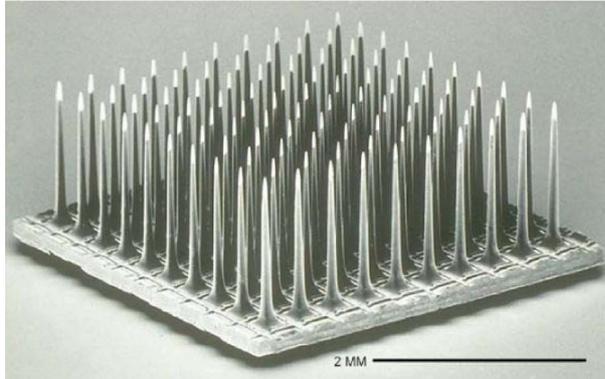


Figure 2.2: The 100-electrode silicon-based Utah Electrode array measuring  $4 \times 4 \times 1.5 \text{ mm}^3$

## 2.2 The next generation neuron interfaces

Several wired solutions for multi-channel recording and stimulation systems have been created in the past. However, the wiring limits movement, increases the risk of infections and severe complications in addition to increase tissue regeneration changing the characteristics. In all, due to the severe exposure of the system, supervision is required at all times in addition to limited and often corrupted data because of external factors interfering with the sensitive nodes. The key factor for the next generation neuron interfaces is therefore wireless telemetry for both power and signal transfer to totally isolate the system. The required number of channels are unknown, though in-vivo measurements have shown that tens to hundreds are required to monitor simple body functions like moving a limb. For future research, it is therefore important not to restrict the data acquiring, but to enable an as broad spectre as possible leaving the requirement of a highly scalable channel count. Later projects have encountered these challenges [7, 24, 41] while in-vivo measurements still have yet to be conducted, most projects in their early stages.

## 2.3 Limitations

The key challenges for the system implementation are power and area. Preliminary studies have shown that approximately  $10 \text{ mW}$  of power can safely

be dissipated using a system measuring  $6 \times 6 \times 2mm^3$  attached to the Utah array [25]. The main issue arising with limited power is the available bandwidth of the wireless link as the intended high channel count leads to raw data generation far higher than possible transmission rates. There exist several techniques for data reduction which exploit compression, signal redundancy and/or level of abstraction [19] though the next generation systems will need to provide high quality signals containing information about which neuron fired at a given time. The process to determine this information is complex and requires high computational force previously performed by powerful external computers.

Concerning area, the whole front-end is restricted to within the boundaries of the probing array. This will both ensure that multiple systems can be put in close proximity to each other while also minimizing invasion and unwanted side effects of electrical signals and heat in other parts of the brain. For the Utah array, 100 channels should be fitted within the  $16mm^2$  area. Other things to consider are tissue regeneration at the probes causing probe characteristics to change over time, leading to continuous calibration requirements.

## 2.4 This project

The strength of this project is the use of a digital spike sorting technique to digitize the spike events. This will reduce the requirements on the wireless link while providing high quality data and for the future support on-chip response to the brain activity, stimulating the nerves back to prevent seizures or similar if necessary. A simplified overview of the system is shown in figure 2.3 on the facing page. The capacitor  $C_x$  together with the inherent frequency response of the low noise amplifier (LNA) high-pass filters the input removing the unwanted frequency range from DC to  $300Hz$ . The LNA is of band-pass type implementing together with the low-pass filter(LP) also a 5kHz cut-off frequency before amplified by programmable gain amplifier (PGA) adjusting the signal level to the dynamic range (DR) of the ADC. The output of each channel is input into a rolling buffer in the digital signal processor (DSP) using template matching to find spike events. The events are transmitted over the wireless link coded with the channel number and a time-stamp. The templates are created in a calibration period using a high resolution high sampling frequency ADC which data are sent to an external computer for processing. The generated templates are later uploaded to the system. When the system is running in normal mode, the deviation of matches to the

templates are determining if a new calibration is required. However, tissue regeneration is only expected to require daily or weekly adjustments.

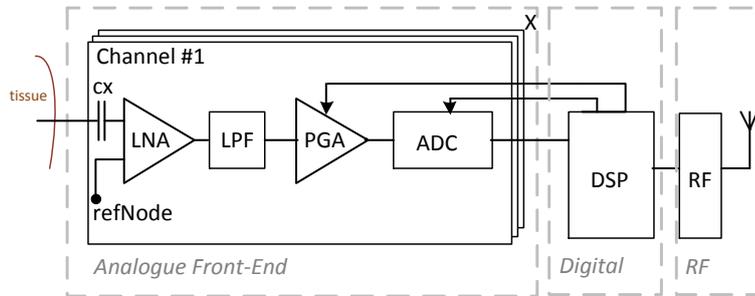


Figure 2.3: System overview

## 2.5 ADC requirements

For this system to work, internal preliminary simulations have shown that the analogue signals need to be digitized at  $16kSamples/s$  using 8-bit resolution for the DSP's algorithms to work with satisfactorily numbers of false-negatives and false-positives. The power consumption is restricted to  $2\mu W$  though less should be striven as to enable more computational power and options in other parts of the system. The area should be less than  $0.1mm^2$  to accommodate 100 channels within the boundaries of the Utah array including the reset of the front-end and the input DR should be  $0.9 \pm 0.5V$ , the specifications derived in compliance with the neural team. However, the PGA's output can also be a current, removing the last output stage making it a  $g_m$ -source. The sample and hold will be done in voltage-mode after amplification. Table 2.1 on the next page summarizes the requirements.

Table 2.1: ADC system requirements

PARAMETERS	CONDITIONS	MIN	TYP	MAX	UNIT
<b>Power Supply</b>					
Supply Voltage		1.75	1.8	1.85	V
Power				2	$\mu W$
<b>Area</b>					
Analogue front-end	per channel			0.175	$mm^2$
ADC estimate	per channel			0.1	$mm^2$
<b>Accuracy</b>					
Resolution			8		bit
INL			$\pm 0.5$		LSB
DNL			$\pm 1$		LSB
Offset			$\pm 0.5$		LSB
<b>Timing</b>					
Sampling Frequency			16.384		kHz
Conversion Time				60	$\mu s$
<b>Input/Output</b>					
Input Range	Voltage Mode	0.4		1.4	V
	Current Mode		Optional <sup>†</sup>		I
Output	Parallel		8		bit
<b>Other</b>					
Temperature <sup>††</sup>		20	37	42	$C^\circ$

<sup>†</sup> The output of the  $g_m$ -source will be designed after the ADC-requirements

<sup>††</sup> Includes testing environment for convenience. The actual requirement is  $37 \pm 5^\circ C$

# Chapter 3

## Background

In this chapter, theory and background material needed to understand this thesis will be explained. However, some basic theory will be assumed known to the reader and of such refers to general books of analogue design and ADCs [12, 26].

### 3.1 Topology

From previous work [22], the topology was chosen to be a current-mode SAR ADC. The choice was based on figure 3.1 on the following page showing a figure of merit (FOM) versus power plot for recent<sup>1</sup> IEEE publications [1, 2, 8, 9, 16, 17, 20, 27, 29, 32, 35, 36, 38, 42, 44] in addition to be supported by several published papers [17, 40]. The FOM is given by equation 3.1 on the next page, normalized by the term in the parenthesis for technology interdependency. The presented ADCs have resolutions around 8-bits and relatively low sampling frequencies as this is the performance of interest. The figure indicates that current mode SAR ADCs at the moment have the best performance for the given specifications. They are also small compared to their voltage mode counter parts using large capacitors for accurate matching, thus suitable for single channel implementation. For further argumentation

---

<sup>1</sup>As of October 2010

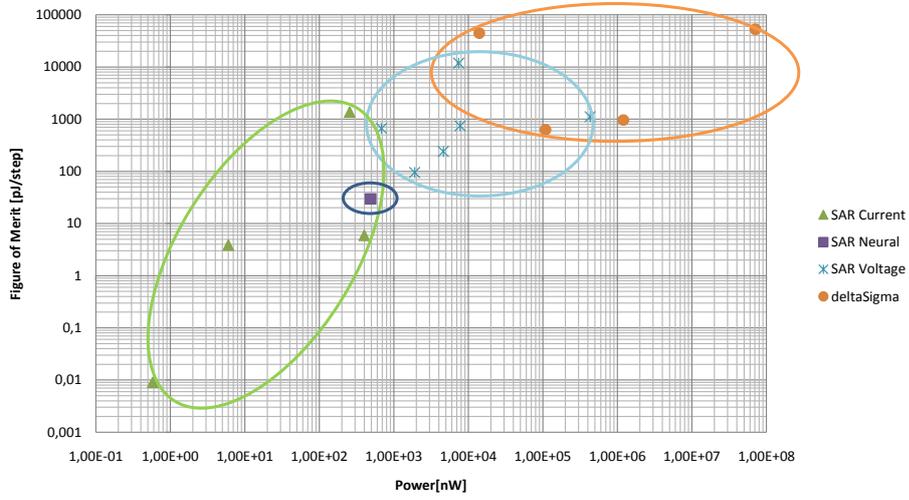


Figure 3.1: Comparison of recent published ADCs in performance of interest

of selected topology it is referred to the term paper.

$$FOM = \frac{Power}{2^{ENOB} \cdot F_s} \cdot \left( \frac{\gamma \cdot Area}{Technology^2} \right) \quad (3.1)$$

The main components within a current mode SAR ADC shown in figure 3.2 on the facing page are a current comparator, a sample and hold module, a current mode digital-to-analogue converter (DAC) and some logic. The topology is based on the same fundamentals as its commonly known voltage-mode counterpart (e.g. charge redistributed SAR ADC), though this only needs MOSFET devices in its implementation. This introduces the potential of: (1) reduce the area requirements, (2) reduce power consumption and (3) achieve high speed operation. The digital output is acquired by comparing the input current to a reference current in sequence, for each bit in the selected resolution, adding up to N-cycles per conversion (i.e. a binary search). The reference current is generated by a current mode DAC controlled by the digital logic and is calculated according to equation 3.2. Here,  $I_{bias}$  controls the input DR, which makes the power consumption of the DAC module directly proportional to the signal level and thus favourable for low energy signals. The current comparison feeds back to the logic which adjusts the reference current.

$$I_{DAC} = (b_n \cdot 2^n + b_{n-1} \cdot 2^{n-1} + \dots + b_1 \cdot 2 + b_0) \cdot I_{bias} \quad (3.2)$$

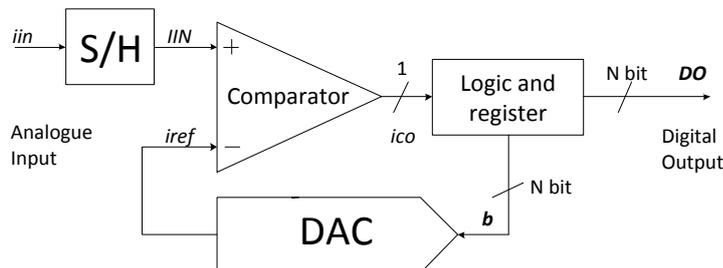


Figure 3.2: General SAR ADC system overview.

In the following, the different parts needed in an implementation of a current mode SAR ADC will be explained, starting with the current mirror as it is a fundamental building block. A note concerning all the analogue circuit parts is that the low overall power consumption results in a nano-ampere current range, yielding weak inversion operation for most devices.

## 3.2 Current Mirror

The main performance limitations of a current mirror are frequency response, absolute accuracy, noise, mismatch and DR which all are dependent on the topology, transistor size and current level. All will be explained in the following.

### 3.2.1 Topologies

There exist many topologies for current mirrors though for low power applications, four are most common selected due to accuracy, simplicity, low-voltage operation and widespread usage. These are (1) simple current mirror (SCM), (2) Wilson current mirror (WCM), (3) Improved Wilson current mirror (IWCM), (4) cascoded current mirror (CCM). Active current mirrors make use of amplifiers to drive compensations, feedback loops and/or regulators. This consumes extra power in respect to the data signal and is therefore to be avoided in low power applications if possible, though they have much better specifications otherwise. Other subsequent modes of the previously mentioned mirrors are the use of potential shifted sources of the transistors and extra biasing currents to set stable operation region. All techniques inherent different characteristics, thus trends and brief overviews will be given

for each in the next sections.

### 3.2.2 Noise

Noise can be a significant error source for many applications, especially when using low powered signals. However, for current mirrors considering small gain and low frequencies, the added noise is negligible, applying to all mentioned topologies. However, for small lengths and narrow bandwidths at low current levels one has to be careful as the flicker noise could prove significant. Methods to minimize noise are: (1) reduce the gain; (2) augmenting the output impedance; (3) decrease  $g_m$  by: (a) decreasing  $W$ ; (b) decreasing  $I_D$ ; (c) increasing  $L$ . Moreover, it may seem to be favourable using cascaded gain stages to achieve gain requirements, though for higher values the best way is still direct conversion [4].

### 3.2.3 Frequency response

Regarding frequency response, the behaviour of the topologies is very different. Mutual design strategies for maximizing the bandwidth are: (a) Increasing the quiescent current; (b) decrease  $W$  and/or  $L$  ( $L$  being the dominant parameter); (c) decrease the gain to move the dominant pole imposed by  $C_{gs}$  of  $M2$ , though the two first must be applied with care as decreasing the output impedance affects the accuracy. Also note that the current frequency response is often not restricted by the current mirror or the output load, but mainly due to the current to voltage(I-V) conversion performed by the load itself. This is setting strict requirements to the load for a given current. Concerning the topologies, both Wilson mirrors are performing much better than the others and are in general not as affected by larger gains because of their negative feedback compensation [3]. The CCM is the slowest of the presented topologies.

A technique of bandwidth enhancement is to add a resistor between the gates of the mirroring transistors demonstrated with a SCM in figure 3.3 on the facing page. This will effectively change the first-order low-pass filter function of the SCM given in equation 3.3 on the next page to a second-order low-pass mirror with one zero and two poles given by equation 3.4. By choosing  $R_C = 1/g_m$  and  $C_{gs1} = C_{gs2}$ , the zero is cancelled out by a pole yielding a first order system with twice bandwidth of the original circuit. However, on-chip

poly-silicon resistors have rather large variations in addition to be affected by temperature and are therefore unsuitable for this purpose. Exchanging it with the  $r_{ds}$  of a small transistor biased in a manner to compensate for global process variations and temperature drift is therefore necessary[21].

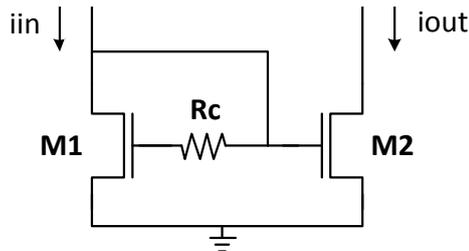


Figure 3.3: SCM with compensation resistor for bandwidth enhancement

$$\omega_0 = \frac{g_m}{2C_{gs}} \quad (3.3)$$

$$\omega_0 = \sqrt{\frac{g_{m1}}{R_C C_{gs1} C_{gs2}}} \quad (3.4)$$

### 3.2.4 Mismatch

Mismatch is a substantial part of the limitations considering all precision analogue design and is identified by the differential performance of two or more devices. For current mirrors, this is defined according to equation 3.5 and is affected by a numerous of processes. Commonly, the deviation of the output current of a current mirror is given by equation 3.6, moreover introducing the current ratio  $M : N$  results in the relative variance of the current ratio  $R$  given in equation 3.7. The error associated with edge variations is given in equation 3.8. The derivation of this is given in [12, p.75]. The later equation is stating that if the ratio is due to dimension sizes in the transistors, an additional offset is introduced and the use of equal sized parallel devices is recommended. Furthermore, the variance is widely determined by the smallest device making it important to keep ratios low so to make the minimum sized device as large as possible. It is also evident that increasing the area, preferable the length to also increase  $V^*(= V_{gs} - V_T)$  is the only way of reducing the mismatch for a given current. The foundry provides the

parameters for these equations, though including extra parameters for deviation in effective channel length and width to accommodate for mismatch to the measured performance. The mismatch scales with the inverse of the current level and is known to grow exponentially in sub-threshold region.

$$\delta_I = \frac{I_{in} - I_{out}}{I_{in}} \quad (3.5)$$

$$\frac{\sigma^2(I)}{I^2} = \frac{1}{WL} \cdot \left( A_\beta^2 + \frac{4AV_T^2}{V_{GS} - V_T} \right) \quad (3.6)$$

$$\frac{\sigma^2(R)}{R^2} = \frac{1}{WL} \cdot \left( A_\beta^2 + \frac{4AV_T^2}{V_{GS} - V_T} \right) \cdot \frac{1}{2} \left| \frac{1}{M} + \frac{1}{N} \right| \quad (3.7)$$

$$\varepsilon_{sys} \approx \delta W \left| \frac{1}{W_2} - \frac{1}{W_1} \right| + \delta L \left| \frac{1}{L_1} - \frac{1}{L_2} \right| \quad (3.8)$$

If cascoding the output of several current mirrors, the total standard deviation follows equation 3.9 though the covariance is hard to predict because the common gate voltage is affected by the mismatch of the input mirror. Other effects like channel length modulation will also affect the correlation, but to a smaller extent. To provide an approximation, equation 3.10 is given and holds via the Cauchy-Schwarz inequality [33, p.291]. It describes the maximum covariance for a finite set of variances and should be used to calculate the maximum mismatch. However, simulations will probably give better results as the system has only one of the transistors in common.

$$\sigma_{Tot} = \sqrt{\text{var}(X) + \text{var}(Y) + 2\text{cov}(X, Y)} \quad (3.9)$$

$$|\text{Cov}(X, Y)| \leq \sqrt{\text{Var}(X) \cdot \text{Var}(Y)} \quad (3.10)$$

Though the foundry's models have a good correlation with the real transistors, some deviation exists. The devices reported tested are of  $W/L \geq 1$  with the deviation between model and measurements increasing with a decreasing ratio. Extensive fundamental experiments reported in [18] yield  $V_T$  mismatch not following a simplistic  $1/\sqrt{\text{area}}$  function, especially for wide/short and narrow/long devices and comments on foundries attempts to accommodate

the otherwise anomalous scaling behaviour by using the effective length and width. Moreover, they claim that in higher Spice modelling, this is circumvented by creating local subsets fitting the model to the data whilst creating discontinuities in the model. This makes the accuracy of hand calculations regarding mismatch in many cases often inadequate and could lead to over design. Calculations should therefore only be used for rough estimations.

In the choice of PMOS versus NMOS for best matching properties there are in general no trends across technologies regarding voltage biased devices. However, using current bias, i.e. a diode connected transistor, the lower electron mobility of the PMOS results in a higher  $V^*$  effectively reducing the mismatch in accordance with equation 3.6/3.7. Another technique of enhancing mismatch similar to  $V^*$  incrementation, exploits changing the substrate voltage, introducing a bipolar action potential in the substrate-to-source junction for the forward biased transistor. This is well documented and works especially well in weak inversion and for small devices. However, increasing the  $V_{BS}$  voltage too much results in unpredictable effects and should be avoided [11]. The  $V_{BS}$  affects the effective threshold voltage according to equation 3.11, thus also  $V^*$ .

$$V_T = V_{T_0} + \gamma(\sqrt{2\phi_p + V_{SB}} - \sqrt{2\phi_p}) \quad (3.11)$$

Regarding layout, symmetry is of the essence concerning mismatch and the use of dummy devices is common to avoid under or over etching in the fabrication process. Common centroid layout or at least parallel or stripe pair patterns are to be used together with current flow in the same direction for matched devices. Further, large spacing and no routing over active area of all analogue components should be employed. For an extensive overview of layout guidelines and rules, it is referred to [5, 39] in addition to the foundry's own manual (confidential).

### 3.2.5 Absolute Accuracy and dynamic range

For the mean output current, the relative error is given by equation 3.12. Disregarding mismatch, asymmetrical properties are the sources of the deviation, either caused by an inherent asymmetry in the topology as for the WCM, or through channel length modulation caused by a non-zero differential input-output voltage which is a valid assumption in most cases. Equation

Current mirror	DC matching error $\varepsilon$	OVC
SCM	$\lambda(V_{out} - \frac{1}{2}V_{in})$	$\lambda$
WCM	$-\frac{\lambda V_{in}}{2(1+\lambda V_{in})} + \lambda^2 \sqrt{\frac{I_{in}}{2K}}(V_{out} - V_{in})$	$\lambda^2 \sqrt{\frac{I_{in}}{2K}}$
IWCM	$\lambda^2 \sqrt{\frac{I_{in}}{2K}}(V_{out} - V_{in})$	$\lambda^2 \sqrt{\frac{I_{in}}{2K}}$
CCM	$\lambda^2 \sqrt{\frac{I_{in}}{2K}}(V_{out} - V_{in})$	$\lambda^2 \sqrt{\frac{I_{in}}{2K}}$

Table 3.1: Output resistance of the four most common current mirrors

3.13 provides a quality factor for this, first defined in [43].

$$\varepsilon = \frac{I_{out}}{I_{in}} - 1 = \alpha - 1 \quad (3.12)$$

$$OVC = \frac{\Delta I_{out}/I_{in}}{\Delta V_{out}} \quad (3.13)$$

The channel length modulation effect, in resemblance to the early effect of the bi-polar transistor, is caused by different bias conditions changing the position of the depletion edges. For the MOSFET, different  $V_{ds}$  voltages causes a change in the effective channel length, thus also  $r_{ds}$  leading to changing output characteristics of the current mirror. Table 3.1 summarizes the characteristics of each of the common mirrors derived from their respective output resistances [43] where  $K = \mu C_{ox} \frac{W}{L}$  being the mobility, gate oxide capacitance per unit area, transistors width and length respectively.  $\lambda$  is the channel length modulation factor and  $V_{in} = 2\sqrt{\frac{2I_{in}}{K}} + V_T$ . However, the equations are only valid for the saturation region. The literature provides little if no information about weak inversion.

From the equations it is evident that the WCM withholds an extra term independent of  $V_{out}$  compared to the other cascoded mirrors which causes an imbalance in the input-output voltage, increasing the error. From the topology of the WCM, it is evident that the output resistance is different from the input resistance, and thus a non-unity transfer function is expected for a 1-to-1 mirror. Both the SCM, CCM and IWCM withhold symmetric properties and thus provides  $i_{in} = i_{out}$  given no mismatch and  $V_{in} = V_{out}$ . However, for most systems, the output voltage will deviate from the input voltage and therefore skew this balance. The channel length modulation is

dependent on region of operation, but cascoding devices are an efficient way of minimizing the effect regardless of this.

The DC-output value of a current mirror can be precisely designed for a given current level. However, if a wide range of input values are required, the mirror may change region of operation, giving non-linear behaviour.

### 3.3 Converter Performance Characteristics

To be able to compare and determine the performance of a data converter, it is referred to the commonly used terms described in [26], chapter 11.5. In short, the offset, gain error, absolute accuracy, integral nonlinearity (INL), differential nonlinearity (DNL), monotonicity/missing codes, sampling rate and DR are defined and explained there. This is needed both for the characterization of the whole system, but also for some of the sub-modules. However, INL and DNL need some further elaboration as they can be defined in several different ways. In a current mode approach, gain error and offset are easily cancelled and a linear best-least-square approximation curve is therefore to be used. INL and DNL are then defined as the maximum difference from the fitted line.

### 3.4 Current Comparator

The basic principle of the current mode SAR ADC is to compare two currents to determine which is the largest. To do this, a current comparator is needed and is easily achieved by joining the currents of opposite signs and measure the output in accordance with Kirchhoff's current law (KCL) yielding  $i_c = i_{ref} - i_{in}$  given figure 3.4 on the following page. As the figure indicates, there are two approaches to measure the residual current  $i_c$ .

In the capacitive method, if the  $i_c$  current is positive, it will charge the capacitor. If it is negative, it will discharge it. The capacitor is normally not explicitly defined in the design, but makes use of the parasitics associated with the summing node as increasing the capacitor value will linearly slow down the circuit in accordance with equation 3.14 on page 19. To give a numerical example; assuming  $0.2V$  is needed to determine the logic output, the parasitic capacitance is  $15fF$  and the residual current is  $0.1nA$ , the circuit

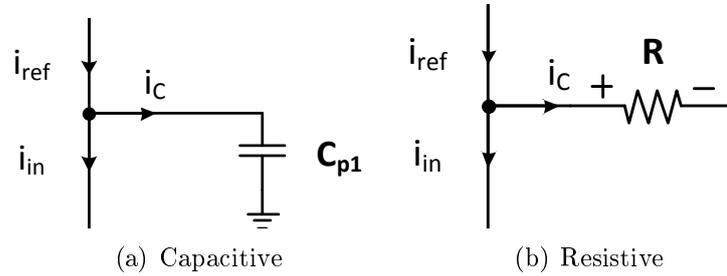


Figure 3.4: Principle of current comparison

will use  $100\mu s$  to reach the threshold corresponding to a speed of  $10kHz$ . It is therefore evident that for small currents, care must be taken to reduce the parasitics. Because the technique exploits the time domain as a parameter to determine the output, for an infinite amount of time the theoretical resolution would be infinite. As these parameters are inverse proportional to each other, speed can be sacrificed for accuracy and is the method of choice when high resolution is required. Feedback loops could be applied to increase transition time.

The resistive topology exploits the instant voltage drop/gain over the resistor as a result of the residual current flowing through it. For an ideal circuit, the speed would be infinite, though for real implementations parasitics will introduce limitations. However, the resistive method is widely used when high speed is required. On the other side, because the voltage supply is limited and the unit measuring the voltage over the resistor has a finite resolution, the current values through the resistor are limited and thus give rise to a finite resolution. A combination of the two techniques is also possible [10, 14] and will be explained later.

It is also common to distinguish between synchronous and asynchronous design. However, synchronous methods are not applicable for this ADC as there are no accurate clock sources available on-chip, thus auto-zeroing techniques for offset attenuation or propagation delay reduction though the use of clock-controlled circuitry are not possible.

### 3.4.1 Implementations

The easiest capacitive implementation is using an inverter which input is connected to the summing node employed by previously published works[2, 17].

However, if the residual current is small compared to the capacitor, the time where the input of the inverter is between the rails becomes significant. In this period, the current draw of the inverter is huge resulting in an energy inefficient system. Also, because the system is non-ideal, the input current sources will be affected by the summing voltage through the channel length modulation effect and will result in a finite voltage non-equal to the rail voltages. Especially if the two currents are near equal to each other, the voltage will not deviate much from common-mode, resulting in a huge power loss, also after settling. A solution to this may be to use a current starved inverter chain to control the current draw. Other methods involve using operational transconductance amplifiers (OTAs) or Operational amplifiers (OPAMPs) though requiring relative large bias currents for slew-rate requirements following the same simplistic equation in 3.14. Some existing amplifier topologies using feedback to control the output biasing, encounter this by reducing the bias when not shifting output value resulting in high slew-rate and low power operation [30, 34]. However, these techniques are often complicated and more suitable in higher power applications.

$$i_C = C \frac{dV}{dt} \Rightarrow \Delta t = \frac{C \Delta V}{i_C} \quad (3.14)$$

The resistive (low impedance) method is usually employed using an amplifier connected in a resistive feedback configuration allowing the residual current to flow to the amplifier's output node, using the amplifier to source or sink the current for equilibrium [28].

The third option is to combine the two methods and has been successfully employed yielding high accuracy at high speeds relative to the power level [10, 14]. Figure 3.5 on the following page demonstrates this functionality. The circuit has two operating modes being either positive or negative  $i_C$  where the  $M1$  path is used for the negative currents and  $M2$  for the positive.  $M1$  and  $M2$  must be biased in a manner to make the  $r_{ds}$  of the respective transistor equal to high impedance relative to the input current near common-mode output voltage levels. Considering a positive  $i_C$  charging the parasitic capacitor between the input and the output, the input voltage  $V_C$  will start deviating from the common-mode voltage  $E$  and the inverting amplifier will switch transistor  $M2$  on as  $V_{gsM2}$  increases with decreasing output voltage. Turning  $M2$  on draws current to the input node equalizing the voltage to common-mode. Ideally this will keep the input voltage at common-mode and the amplifier's output will vary dependent on how much

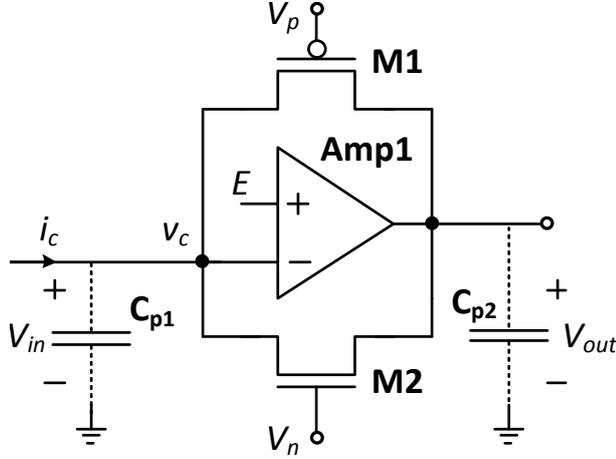


Figure 3.5: Multilevel input impedance current comparator

current needed to be sourced or sink from the input node. The output voltage will also need to further be amplified to make a logical value. However, this is not a problem as the sensitivity of this node can be adjusted with the preceding amplifier *Amp1*. It is similar for the negative currents. There exist other similar topologies, but as the parasitic capacitance  $C_{p1}$  is important to keep as small as possible, source follower structures in the resistive path are to be avoided. This is because the capacitance  $C_{gs}$  of a transistor tends to be large and in this configuration effectively forms a capacitive coupling from input to output, much alike the Miller effect in amplifier structures slowing down the circuit.

The settling time is given by equation 3.15 where  $GB$  is the gain-bandwidth product of the amplifier and  $J$  is the input current overdrive step, assuming  $[-J, J]$  step response [14]. It is evident that to increase settling time, a high gain-bandwidth product and/or a large input current change is required. Furthermore, the biasing of transistor  $M1$  and  $M2$  should just meet the requirements, though a margin must be assumed allowing for process deviations. Biasing conditions is given by the equations in 3.16 on the facing page.

$$T_D = \sqrt{\frac{2C_{in}}{GB} \frac{V_p - V_n + V_{tn} + |V_{tp}|}{J}} \quad (3.15)$$

$$\begin{aligned}
V_{in} &= V_o = E \\
V_n &\leq CM + V_{tn} \\
V_p &\geq CM - |V_{tp}|
\end{aligned}
\tag{3.16}$$

### 3.5 Current Mode DAC

Another major part of the system is the current reference circuit. This is commonly implemented as a binary weighted current splitting array using switches to steer the current [2, 17]. Figure 3.6 on the next page is showing a typical implementation. The scaling of the transistors to create the different current paths, is with the use of multiple equal sized transistors supported by the theory in section 3.2 on page 11. This is to create accurate current levels with the lowest amount of mismatch. Because the reference current is split into several current paths, the use of feedback current mirrors such as WCM and IWCM is not possible. A downside with this current mode DAC is that the current not used for the output reference is just dumped into the power supply. Turning the mirrors dynamically on and off are normally not possible for 8bit resolution DACs given sub-threshold operation as there will be settling issues limited by the frequency response of the mirrors. The required transistor dimensions for a low production yield are by great too large for a reasonable frequency response. The technique is however demonstrated with great success in [16] also offering multiple gain stages for increased minimum sized current mirrors, though the paper says nothing about mismatch.

Other topologies are R-2R based converters, dynamically matched current source DACs and hybrids together with different coding schemes [26, p.471-484]. Though the dynamically matched current source DAC features no inherent mismatch because it uses the same transistor for both input and output current, leakage over time will affect the accuracy of the output current and will therefore require regular calibration. Though the literature suggests several techniques of improvements [12, p.381], calibration through switched circuitry is still needed and thus not suitable for this project reasoned the asynchronous chip. Coding schemes like thermometer-code are known to reduce glitches when switching between the different output values in addition to support inherent monotonicity and good DNL errors[26, p.477]. Normally, care must be taken for proper timing of the control signals to avoid glitches causing either temporary low impedance between power

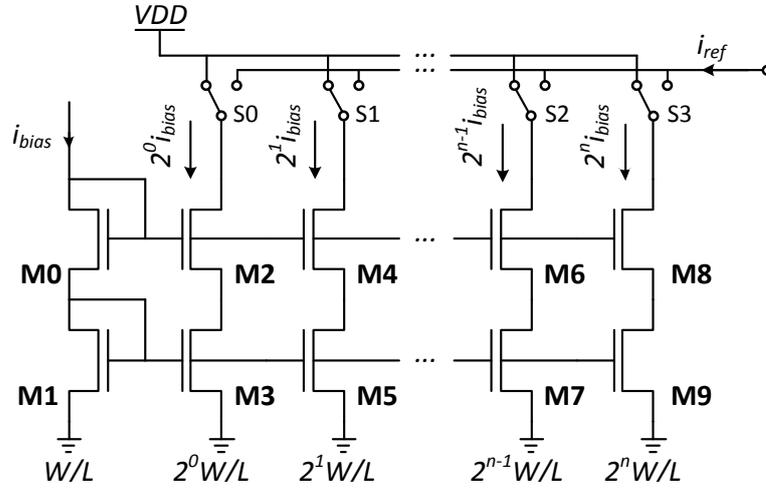


Figure 3.6: Common implementation of a current mode DAC

rails or high impedance nodes which slows down the circuit. This method requires extra logic for the binary-to-thermometer code conversion though. Using only one control signal per channel is another technique to minimize this problem [26, p.478].

There exist several different R-2R converters. They commonly use a larger reference current and divide it into several smaller current paths to generate the binary scaled current output array. A current division technique is described in [6, 15] and repeated in figure 3.7 on the next page for convenience. This is claimed to be independent of process variations and reference current level, though still subject to output voltage dependency and mismatch. The circuit achieves 103dB DR and -80dB THD at 10kHz using  $I_{max} = 100\mu A$  which corresponds to about 12bit operation. However, the circuit requires relative high supply voltage because of its many stacked transistors. It works as follows. Current  $i_{in}$  gets split in two equal portions because the resistance seen from the input node is equal in the two splitting branches given that  $V_a = V_b$ . The current through  $M3$  sees an equal resistance in  $M4 + M5$  and  $M6 + M7$  and gets further divided in two. This can go on, but terminated here in this example for simplicity. It is important that all current branches are active to balance the circuit. In [31] a slightly different scheme is implemented, re-printed for convenience in figure 3.8 on the facing page. It is reported to be able to generate currents down to pico-range with small mismatch, though the accuracy suffers due to the SCM-structure.

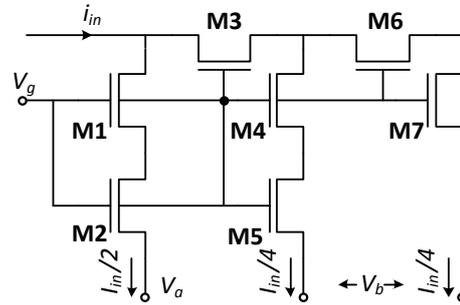


Figure 3.7: Current splitting technique I

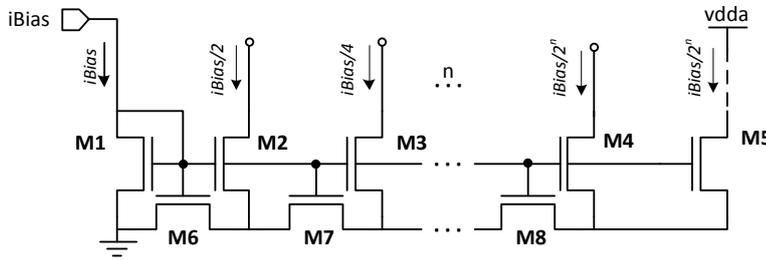


Figure 3.8: Current splitting technique II

### 3.5.1 Performance Limitations

The DR of the current mode DAC, regardless of chosen topology is easy to calibrate as it is proportional to the given reference current cf. equation 3.2. Also offset is easily removed as subtraction and addition in current mode is a matter of connecting nodes together. However, the relative accuracy between the current branches determines the obtainable resolution where absolute accuracy, mismatch and DR of the current mirrors are the determining factors.



# Chapter 4

## Design and Methods

This chapter will describe the design process and in detail give an account for the implementation decisions. The design takes basis in [16], holding the lowest reported power and area<sup>1</sup>.

### 4.1 Estimations

A power, area and error budget was set up regarding the three main modules; the current comparator, the current mode DAC and the logic. In addition, overhead was taken into account for the S/H-circuit not included in the design. Half the power and area and one third of the error budget were reserved this part, though the design of this was left to other members of the team. The estimations take basis in [16] and an overview can be found in the result section, table 7.1 on page 50 for comparison with the final result. Given this budget, the maximum reference current, i.e. the least significant bit(LSB), was estimated to be  $1nA$ .

The design process followed a bottom-up approach, putting the system together as the individual parts got finished.

---

<sup>1</sup>As of October 2010

## 4.2 Current Mirror

An extensive characterization of all mentioned mirrors was performed at the beginning of the project to better know the limitations. Already conducted experiments reported in the term project[22] served as the foundation, though a more complete and thorough approach was conducted for this paper. Width and length for the current interval  $[0.1nA, 1nA]$  were tested for the topologies in section 3.2.1 concerning mismatch, frequency response, DR, noise, leakage and absolute accuracy. An overview of the main results is given in appendix at page 89 with accompanying scripts from page 75. In general, the results showed a consistency with the theoretical principle, though the theory was unable to predict the absolute values within acceptable margin. This is, as section 3.2 suggests, because the common diode-equation for describing sub-threshold operation is too simplistic and requires more sophisticated modelling. The technique of bandwidth enhancement proved well functioning and should be used for the input current mirror as this is the only mirror the technique is applicable for.

## 4.3 Current Mode DAC

The results from the current mirror simulations suggest that the dynamical approach of adjusting the power consumption of the DAC with the input signal level described in section 3.5, is far from realistic. Even using the WCM and neglecting the accuracy and DR, the settling speed of  $\sim 10kHz$  for  $1nA$  bias is not enough to accommodate the sampling frequency requirement of  $16kHz$ , especially considering that three such mirrors are cascaded in this implementation. The numbers are obtained when the transistors are sized to give an acceptable mismatch, though they can be much improved by downsizing. If one disregards production yield, only focusing on the typical performance, this solution is possible and enabled through the use of post-fabrication trimming of the critical transistors. However, this is time consuming and expensive and therefore not suitable from an industrial point of view.

Deciding to use the common current mode DAC implementation described in 3.5, the frequency response requirement of the current mirrors disappears and arbitrary large transistors could be used to accommodate the other specifications as long as within the given area. It is decided to use PMOS current

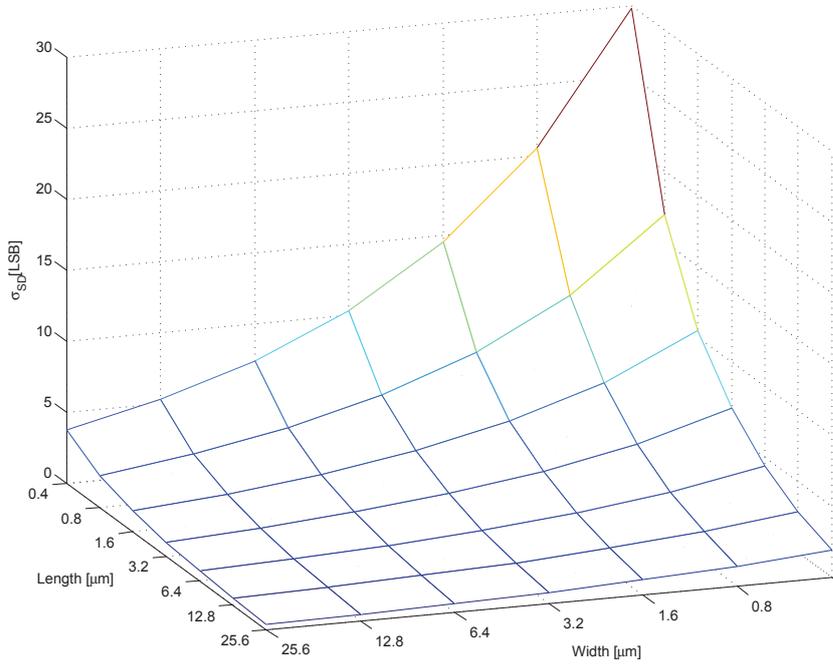


Figure 4.1: SCM Mismatch - 3D model

mirrors as they employ a significant reduction in standard deviation compared to the NMOS mirror. Looking into the mismatch for weak-inversion operation of the current mirrors though, it is evident that it would require impractical large transistors to give 8-bit resolution. The results presented in figure 4.1 and 4.2 on the following page are captured using a  $1nA$  input current with a 1 : 128-ratio current mirror using unit sized devices. The size is given by the  $x,y$ - coordinates and the standard deviation is normalized to the input current. The figures show an inverse logarithmic dependency of the standard deviation to the area.

Calculating the required mismatch for each current path in the 8-bit binary scaled DAC, makes it evident that the requirements can not be met without calibration. The derivation is given in appendix at page 73. From this, it is evident that the MSB requires a standard deviation less than  $\sim \frac{1}{12}nA$ . Figure 4.2 shows that a unit area much larger than  $25.6 \times 25.6\mu m^2$  is required to accommodate this. In fact, the area required will exceed the budget by far and therefore a calibration scheme is introduced. This will be dealt with in a later section.

Assuming that the mismatch will be calibrated within requirements, DR and

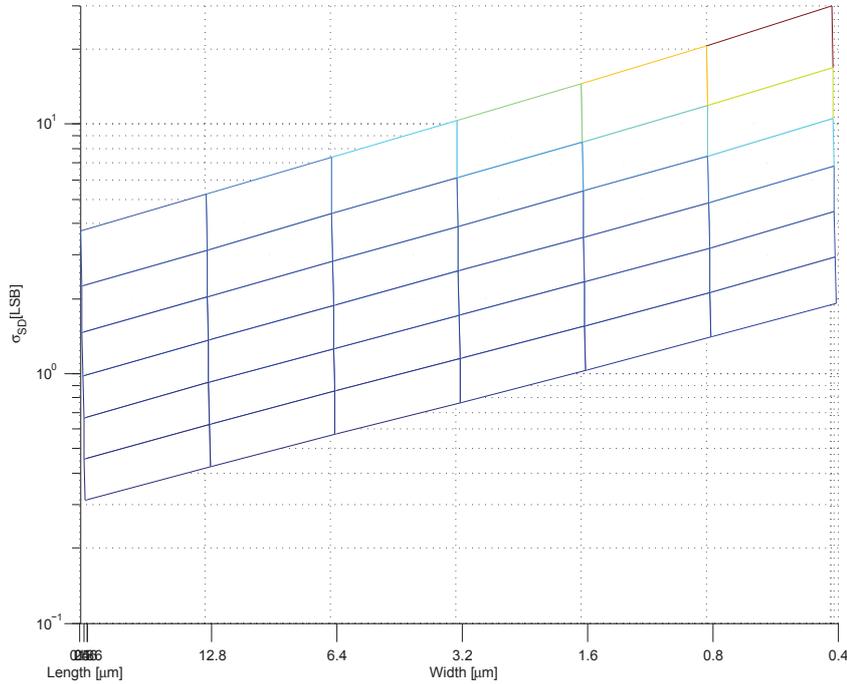


Figure 4.2: SCM Mismatch - Side projection in log-scale

accuracy are the next concern. An overview of the response from the different mirrors is attached in appendix at page 89 including frequency response, noise analysis and leakage. From this, it is shown that the CCM or IWCM is needed for the output current mirror, concerning the linearity over the DR. They also perform well in terms of accuracy which is convenient for easy transistor dimensioning. The simulations shows further that leakage is of no concern for neither of the mirrors, while the Wilson-type mirrors show significant increased noise levels. In frequency response, the Wilson-type mirrors perform best, though all mirrors for reasonable sizes are within specification. The CCM is therefore selected for all the full-range current mirrors needed. The inherent source degradation of the lower mirror in the CCM architecture also improves the mismatch, though to a small extent.

The DR and accuracy are not important for the individual current paths in the DAC as these currents will be DC and can be adjusted individually with different sizing. A SCM is therefore possible saving area compared to the others. However, this will require different sizing of all individual transistors and lead to larger mismatch because of asymmetry in the layout. Other concerns with this are the lack of robustness, in terms of being dependent on an accurate input bias for the right scaling ratio in addition to temperature

dependency. It was therefore chosen to use CCM for the DAC module as well, sacrificing area for increased robustness and a more consistent layout.

## 4.4 Current Comparator

Given the selected bias at  $1nA$ , the input range of the current comparator is from  $-128nA$  to  $128nA$ . With an error budget of  $\frac{1}{3}$  of the overall budget, the comparator will need to settle to  $\frac{1}{6}nA$  accounting for the requirement of INL less than  $\pm 0.5LSB$ . For the settling time, 50%, i.e.  $\sim 4\mu s$  was first assigned. All topologies mentioned in section 3.4 were tested. The simple inverter proved well functioning, but showed a very high average power consumption. Changing the inverter to a current starved inverter chain was briefly tested and reduced the power within budget though increased the signal path delay so that the total comparator delay became too large. The OPAMP method proved too slow because the transistor dimensions of the input stage needed to be large because of mismatch considerations, introducing a large gate capacitance coupled directly onto the comparing node. The multi-level input impedance comparator was tested using a voltage limited inverter as the amplifier introducing a  $\sim 2fF$  capacitance at the input. Estimates of the gain-bandwidth product needed by the amplifier were calculated from equation 3.15 on page 20 yielding around  $\sim 1MHzV/V$  using  $V_p = V_n$ ,  $V_{in} = |V_{ip}| = 0.35$  and were used to design the final amplifier. The multi-level input impedance comparator proved well functioning and selected for the final implementation.

## 4.5 Calibration

To make calibration currents, the need of creating binary scaled reference currents smaller than the supplied bias arose. For this, the second current division technique described in section 3.5 was used. The same argumentation for choosing CCM is used for this module as for the DAC. However, for this purpose it is further extended because the number of current copies made from the reference currents will vary dependent on the individual need of calibration for each chip after production. This will affect the reference current because of the sensitivity of pico-range current levels.

With reference currents for each binary division step down to  $31.25pA$ , cur-

rent mode DACs were made for each bit in the main DAC, adjusted to accommodate the individual channel's calibration needs.

## 4.6 Logic

The digital part of [16] which served as the basis for the initial project, is when realized very power hungry and seemingly a poor design. In detail, it is the pulse generating feedback loop consisting of an inverter chain that uses the largest amount of power. Three solutions were derived to improve this: (1) Using an differential two stage amplifier with large gain in a feedback configuration, (2) a power restricted inverter chain increasing the currents in later stages for sharpening of a digital signal while ensuring limited current draw, (3) a novel threshold detecting amplifier (TDA) using the inherent signal properties to drive a positive feedback path. All three methods significantly improved the power consumption, though the last option proved to be the best.

# Chapter 5

## Implementation

This chapter will in detail describe the final implementation of the ADC and is divided into four sections; Control Logic (SARalg), Current Comparator (iComp), Current Mode DAC (iDAC) and Calibration Module (iCalib) respectively. An overview of how all the modules are interconnected is shown in figure 5.1 on the following page. The ADC has some extra features including a sleep-mode (SLEEP) and a test-mode (MODE). The sleep-mode is used to turn off all bias currents leaving only leakage as the power consumption. This is a feature required by the neuron implementation to turn of defect channels. In test-mode, the current comparator is turned off and the DAC current is copied to the input pin making it a sourcing output pin. This enables the DAC module to be tested separate, thus allowing easier calibration and debugging if necessary. A pin overview is provided in appendix at page 87. A design overview is provided with figure 5.1 on the next page.

### 5.1 Control Logic

The digital logic (SARalg) used to realize the SAR algorithm is shown in figure 5.2 on page 33. The impulse generator (ImpGen) together with the phase generator (PhaseGen) establish the connection between the comparator output and the appropriate storage register for each cycle. The  $NOR_{X2}$  gates provide the control to the current mode DAC using a combination of the non-overlapping phases and the stored values. This design is similar to [16]. However, several enhancements concerning power consumption and reliability have been applied. First, the input from the comparator module

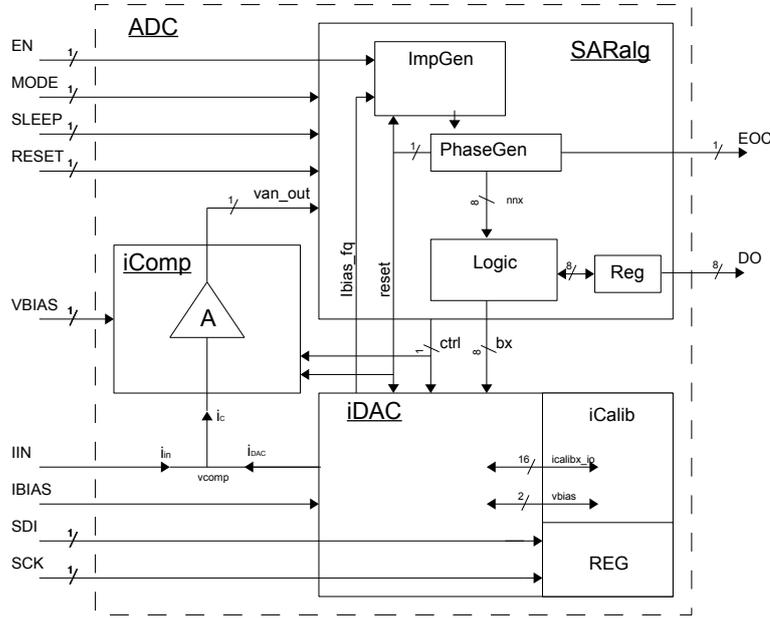


Figure 5.1: System level design for the current mode SAR ADC.

is for this application purely digital making the original storage capacitors redundant and are exchanged with digital flip-flops. This because in addition to voltage drop over time causing the consecutive stage to draw unnecessary power, the capacitors for this technology would be minimum 30fF causing heavy load for the comparator module. Instead the flip flops provided by the founder exhibiting 2fF input capacitance were used, adding up to 16fF load or the equivalent 8x driving strength for the preceding stage. It is also added output registers holding the acquired data until the next set of data is ready, avoiding accurate timing considerations and increasing the flexibility of the system. The  $L_x$  latches used in the phase generator module are also altered reducing power consumption by using positive feedback to drive logical signal fast to rail. The non-ideal logical voltages are caused by the on-resistance of switches  $T_{X1}$  in combination with parasitics leading to long turn-on time. The 2:1  $MUX_x$ s are used to explicitly control the iDAC in test mode.

The impulse generator (ImpGen) as a part of the logic is shown in figure 5.3 on page 34. Current mirror  $M1, M2$  is turned on by the latched enable signal causing capacitor  $C_2$  to slowly charge in accordance to equation 3.14 on page 19. Transistors  $M3 - M5$  are used to turn the module on and off. The TDA is used to trigger a reset function whenever the threshold voltage is reached, starting the integrating process over again using an internal reset loop ( $I2 - I5 + OR$ ). The output is driven by a  $12x$  strength inverter to ac-

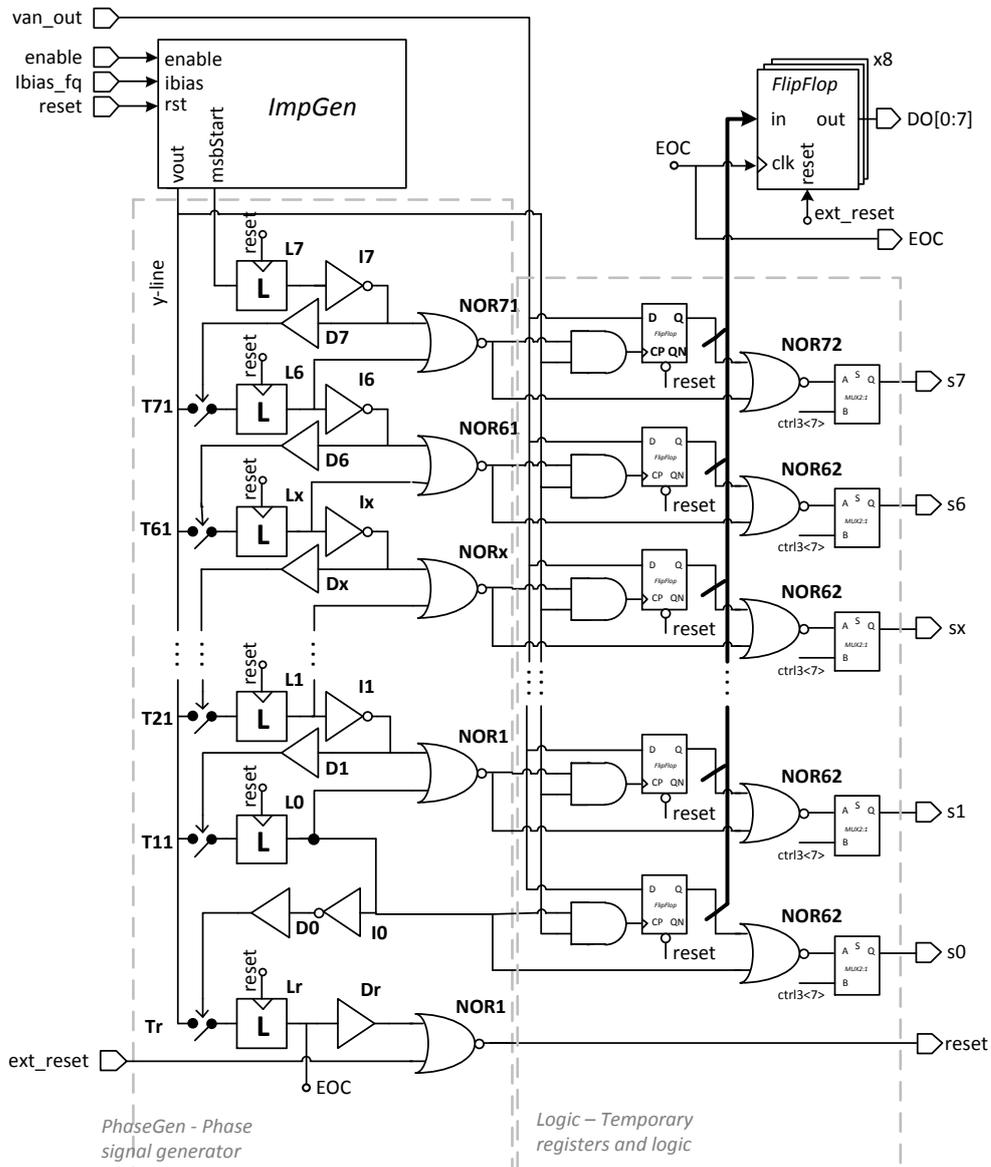


Figure 5.2: Digital circuit used to realize the SAR algorithm and output registers.

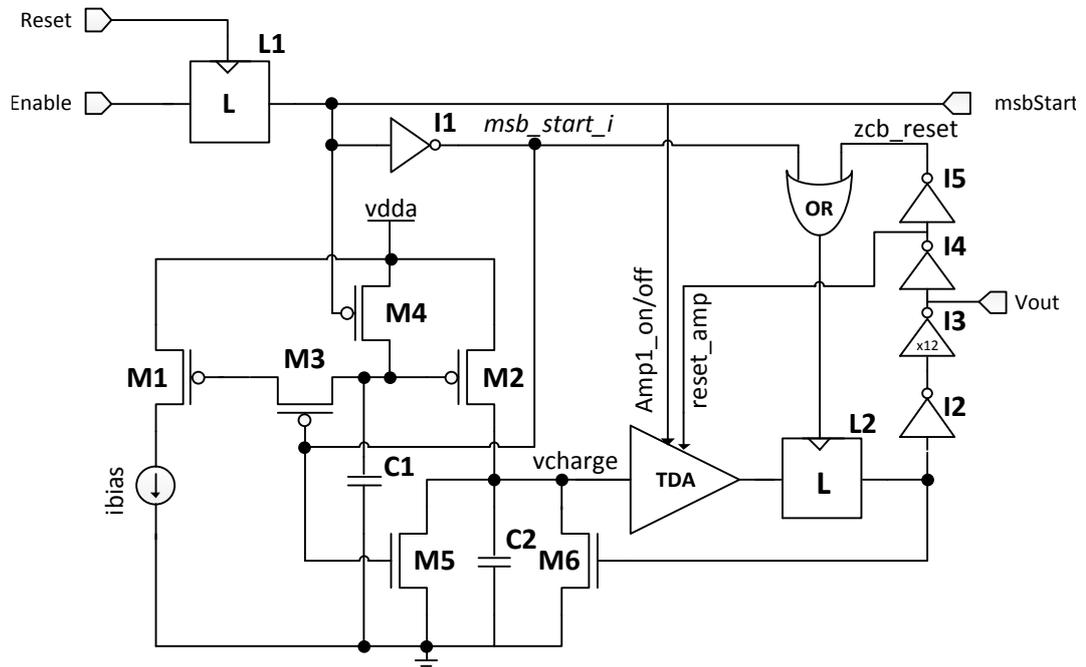


Figure 5.3: Impulse Generator

company the  $\sim 20\text{fF}$  load. This module also supplies the system with control signal *msbStart* telling if the ADC is in a conversion or not.

The TDA is shown in figure 5.4 on the next page. To ensure a fast transition of the amplifier (inverter  $M_1 - M_2$ ) output, positive feedback (with a gain of two) is applied. When the inverter ( $M_1 - M_2$ ) starts changing value, the current through it increases causing a voltage drop over  $M_3$ , thus turning on  $M_5$  [13]. This speeds up the switching process and therefore ensures low power operation by reducing the transition time.

## 5.2 Current Comparator

The current comparator (iComp) is shown in figure 5.5 on page 36. It consists of three stages; the current comparator, an amplifier and a buffer. The input node *v1* is very sensitive to capacitance and was therefore designed to minimize parasitics. The transistor gate area of the input amplifier was therefore minimized using a small sized inverter as the amplifier and avoiding



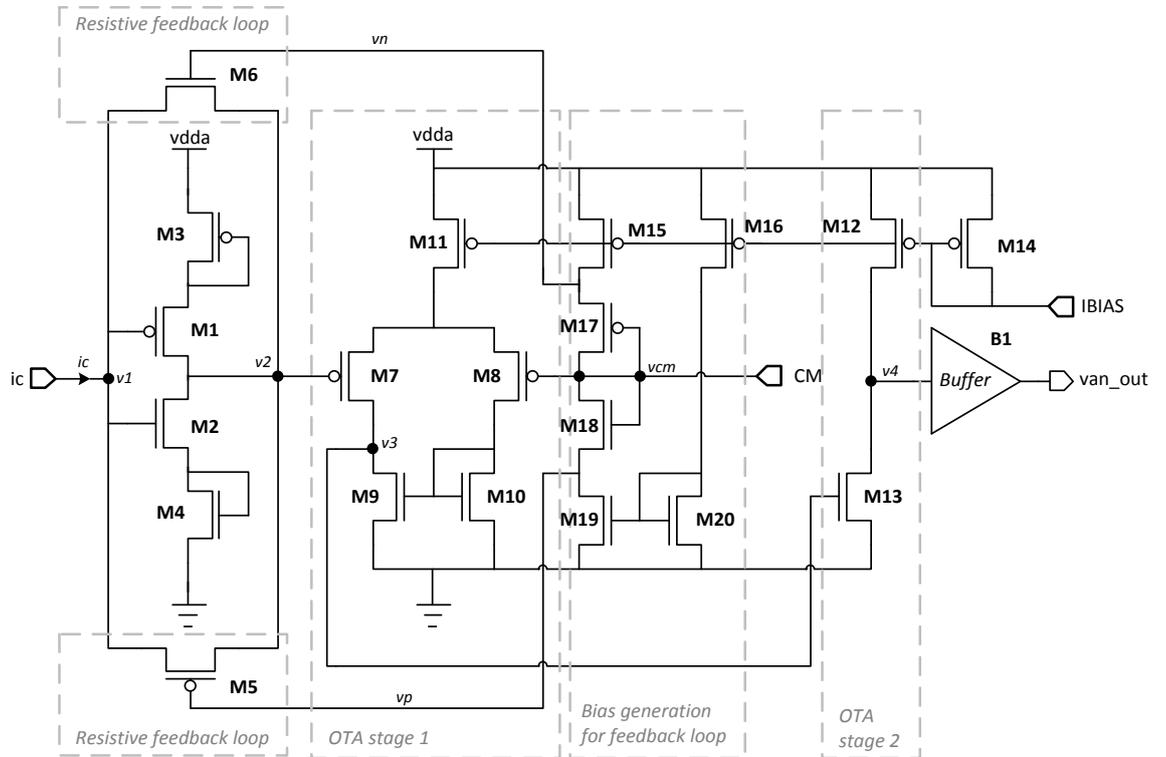


Figure 5.5: Current Comparator

restrict the diode connection. The current is designed to be less than one nano using as small sized devices as possible as the accuracy of the current is of little concern.

### 5.3 Current-Mode DAC

The current-mode DAC is shown in figure 5.6 on the next page. Transistors  $M2 - M13$  in the figure make up the current branches, biased by the diode connected transistors  $M0 - M1$ . The current steering is implemented using transistors  $M8 - M13$  as switches to guide the current. Transistors  $M14 - M17$  are used to create a constant bias current for the input current mirror and transistors  $M18 - M20$  together with inverter  $I1$  are used to turn off the module if the ADC for some reason has to be disabled. Transistors  $M21 - M22$  are used to increase the voltage of the current dumping node to approximate the current output voltage. This ensures minimum change in

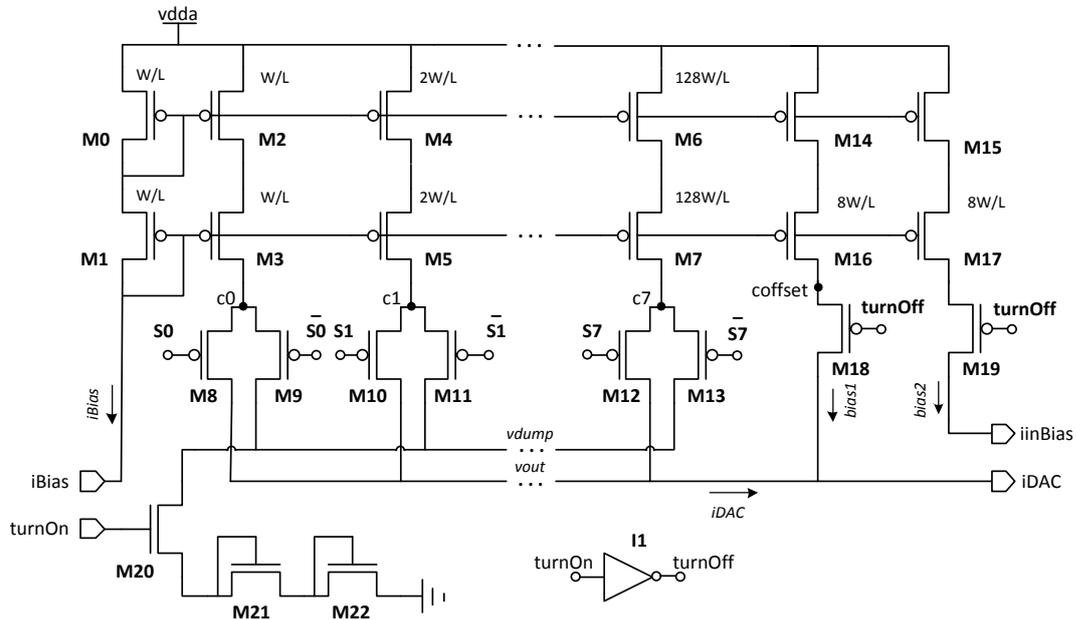


Figure 5.6: Current Comparator

current level through the branches while shifting between the outputs. For testing purposes, the current through transistors  $M21 - M22$  is copied using them as the input of a cascoded current mirror. Using additional switches, this current is guided to the input current pin of the ADC, disconnecting the input mirror, and makes the input to an output if test-mode is enabled. This is done to be able to characterise the DAC and make an automated calibration circuit in addition to support better debugging.

## 5.4 Calibration Module

The calibration circuitry consists of several modules. To be able to generate calibration currents, bias voltages for currents in pico-ampere range have to be generated. This is generated with the  $iBias$  module shown in figure 5.7 on the following page. All devices are of same size except transistor  $M10 - M12$  which has different length to accommodate for operating in weak inversion. All diode connected devices provides a voltage reference to use in the current mode calibration DAC modules.

For the MSB, a 6 bit calibration current with the ability to switch direction

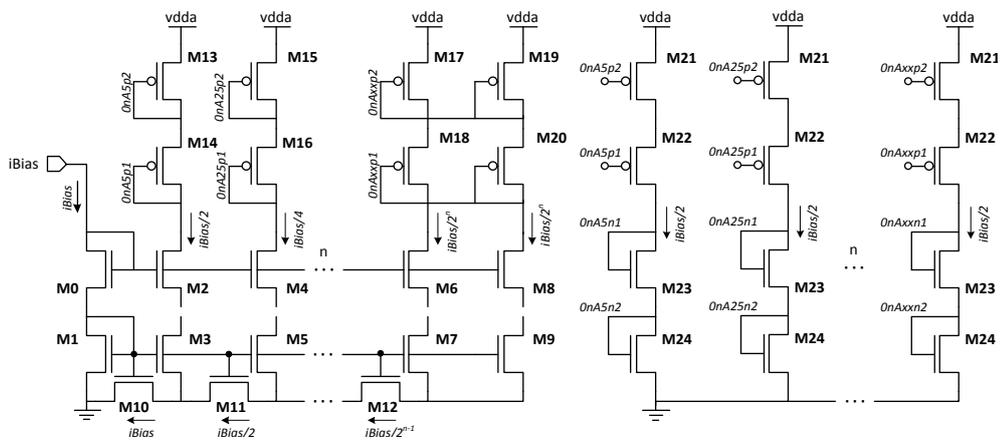


Figure 5.7: Bias Generator

was chosen. This is over-design, but because the only resource adding from 5 bit and up to 6 bit is using more area (equivalent to about one thousand of the total design), this was done to ensure a large margin and for DAC testing purposes as it can give a better indication of how well matching can be done for future design in this technology. For bits 2 to 6, the calibration circuit in figure 5.9 on the next page were used, applying different bias voltages for the different bits.

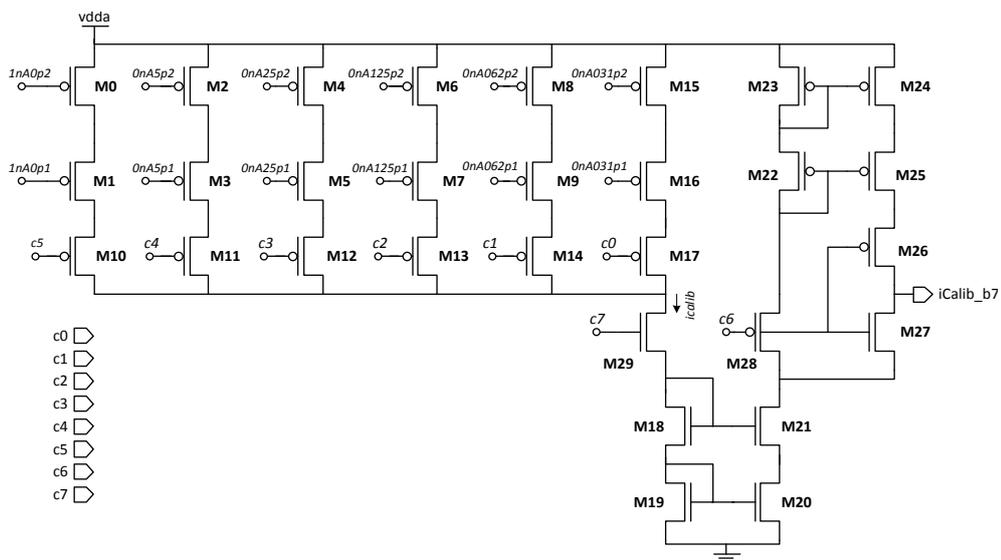


Figure 5.8: Calibration circuit for MSB

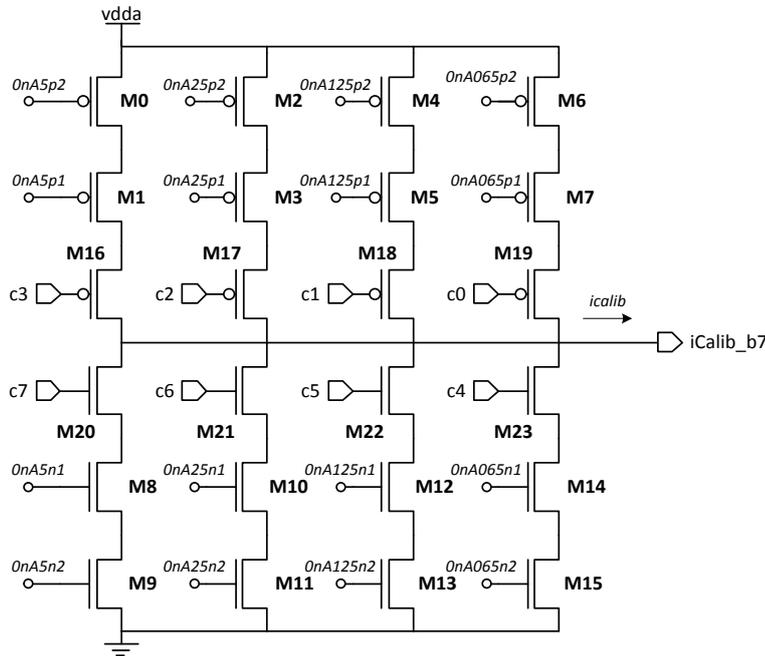


Figure 5.9: Calibration circuit for bit 2-6, though all with different bias-voltages applied

## 5.5 Calibration register

To be able to use the calibration circuits, some kind of storage mechanism has to be added. The original idea was to burn fuses offering a one-time calibration using no extra power and small area. However, because this technology is very new, neither fuse blocks or EEPROM modules are available from the founder yet. Though fuse blocks will be the final goal, a multi-time programmable memory would be handy for testing purposes, and thus the creation of the register shown in figure 5.10 on the following page. To save pins, the register is implemented as a serial shift register using D-flip-flops. The inverters are added to the clock path to ensure right timing. The register offers a solution to only update the *Ctrl3* register. This function is used to be able to set the DAC value explicitly when in test-mode while keeping the calibration data. Table 5.1 on the next page provides an overview of the usage of the registers.

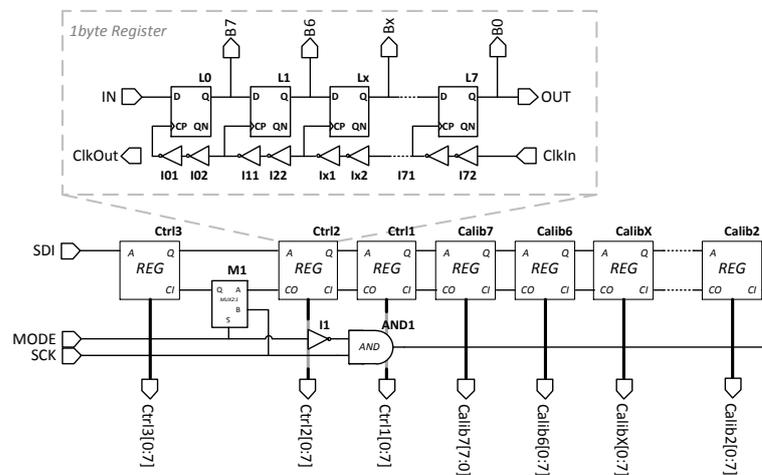


Figure 5.10: calibration register

Table 5.1: Register overview

Register	7	6	5	4	3	2	1	0	
ctrl3	DAC								
ctrl2	ONOFF	SIGN	OFFSET						
ctrl1	ONOFF	SIGN	FREQ						
icalib7	ONOFF	SIGN	VALUE						
icalib[6:2]	NEG				POS				

- DAC** When in test mode, the DAC register specifies the output level of the current mode DAC.
- ONOFF** Turns on or off the specific module.
- SIGN** Specifies if the module should source or sink the current.
- OFFSET** Set the value of the offset.
- FREQ** Set the speed of the conversion.
- VALUE** Set the calibration current.
- NEG** Set the level of current sinking.
- POS** Set the level of current sourcing.

## 5.6 Input current mirror

The implementation of the input current mirror is given in figure 5.11 on the facing page. Bandwidth enhancement through the use of a pole-compensating





# Chapter 6

## Layout

The layout was done in accordance with section 3.2.4 following the guidelines for the selected technology. Figure 6.1 on the following page shows the final result with the accompanying table 6.1 which summarizes the indicated sub-modules marked with colour coded boxes. The whole design was put within a pad ring measuring  $0.4\mu\text{m} \cdot 0.4\mu$  for isolation from the rest of the modules on the test chip, though the final design only occupies  $378\mu\text{m} \cdot 207\mu\text{m} \approx 0.078\text{mm}^2$ . Dummy devices were used at the edges of every current mirror for matching enhancement. Unfortunately there was no time for post layout simulations.

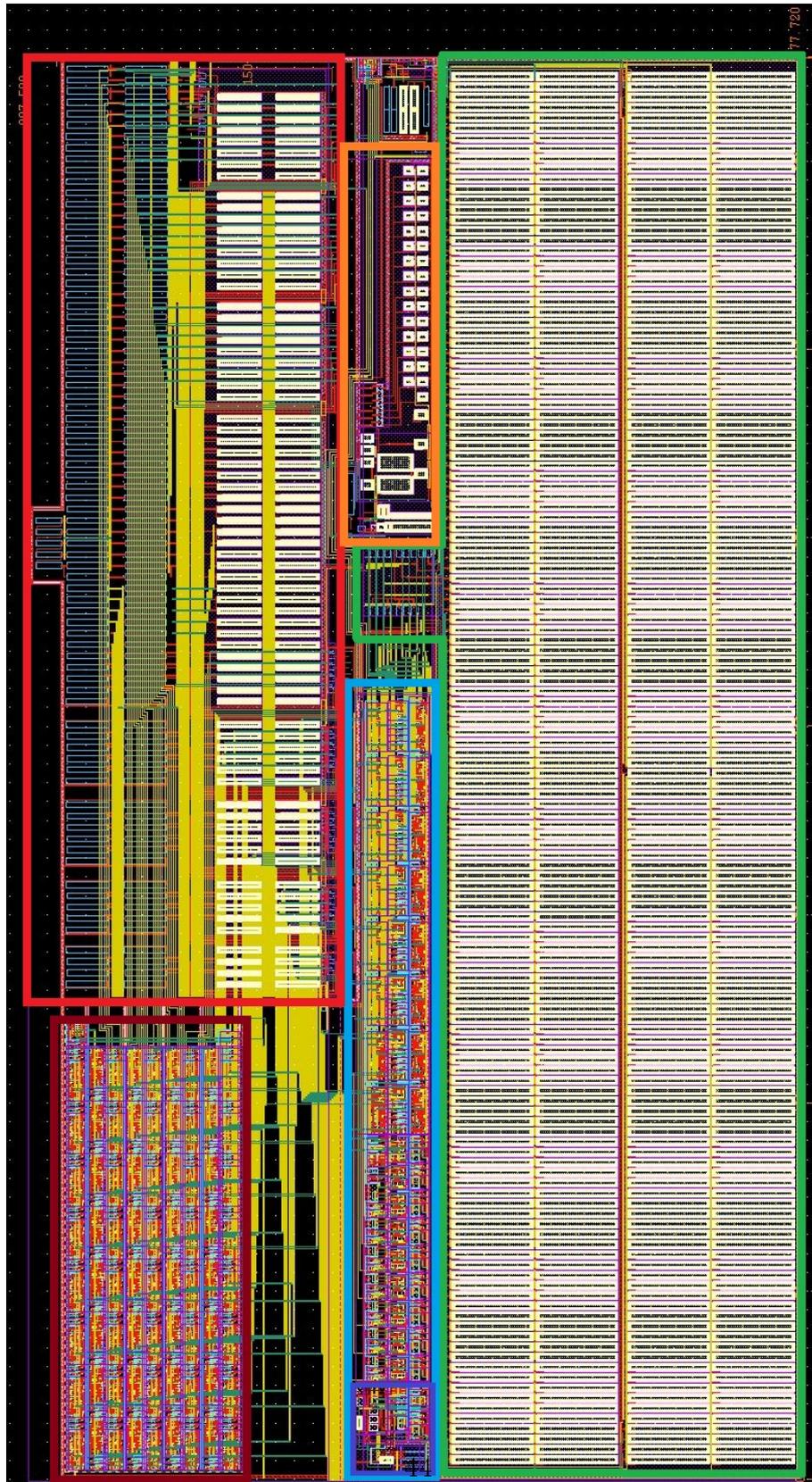


Figure 6.1: Final Layout

Table 6.1: Layout explanation

Module	Colour	Note/Description
iDAC	Green	Laid out using an adjusted common centroid method. Reference bias in the middle, each bit equally spread on both sides with the exception of MSB which is the entire lower row.
iComp	Orange	Path from CM-DAC to input made as short as possible. Wide spread transistors for small interconnected interference.
Logic	Blue	Dark blue area marks the Impulse Generator
Calib	Red	Includes calibration of CM-DAC, sampling frequency bias and offset
Register	Brown	Serial shift register including logic for test-mode operation



# Chapter 7

## Results

This chapter will present the obtained results documenting the functionality and robustness of the design. The first section will only present the top level to provide proof of the overall performance while the next sections will in detail account for each major sub-module.

### 7.1 ADC Core

As every module in the design are implemented as analogue structures, the whole design including calibration and registers is only simulated to verify the connectivity. For the testing of the performance of the ADC, the calibration circuitry was replaced with ideal current sources to save simulation time. This has no impact to the results of the performance as a linear best-least square approximation for the characterisation was used. The ideal calibration served therefore only to ensure that all digital codes were captured with a minimum input current range.

The functionality of the ADC was tested in normal operation mode using a  $1nA$  input bias and a  $1.5nA$  sampling frequency bias corresponding to a  $F_S = 17.5kHz$  leaving a margin for the  $16kHz$  *enable*-signal as the test-bench not implemented a handshake functionality with the ADC. Offset and gain error were calibrated to  $\sim$ zero before the simulation using the provided calibration nodes. An input current stepped with a  $0.25LSB$  resolution in the interval  $[0nA - 256nA]$  was then applied giving a total of  $\sim 63ms$  simulation time. Ideally a smaller step time to increase the resolution of the

measurement should have been used, but was limited to  $\frac{1}{4}LSB$  because of large simulation time. (The selected simulation has a duration of  $\sim 5$  days on the fastest server available.) An Ocean script was developed for automated output generation to minimize memory usage while simulating and for fast and reliable conversion to an output file. The script is attached at page 76. The result of the simulation is shown in the next three figures where the attached Matlab script at page 84 is used for the calculation of INL and DNL. Table 7.1 on page 50 summarizes the power consumption including a comparison with the estimated values.

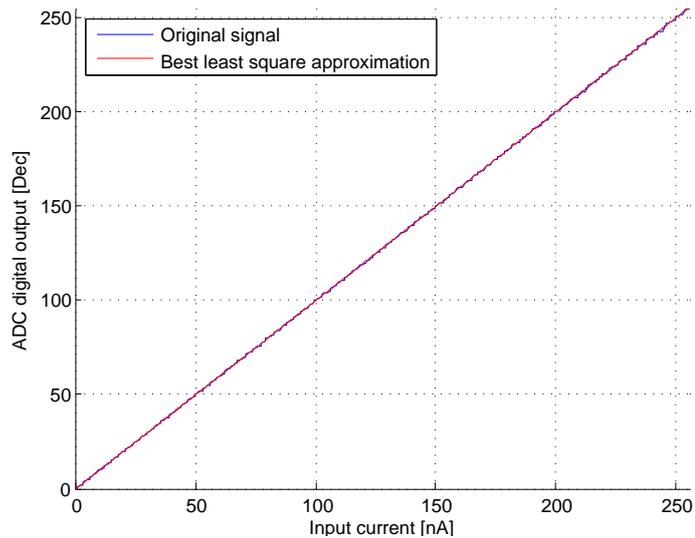


Figure 7.1: Input/output characteristics of the ADC

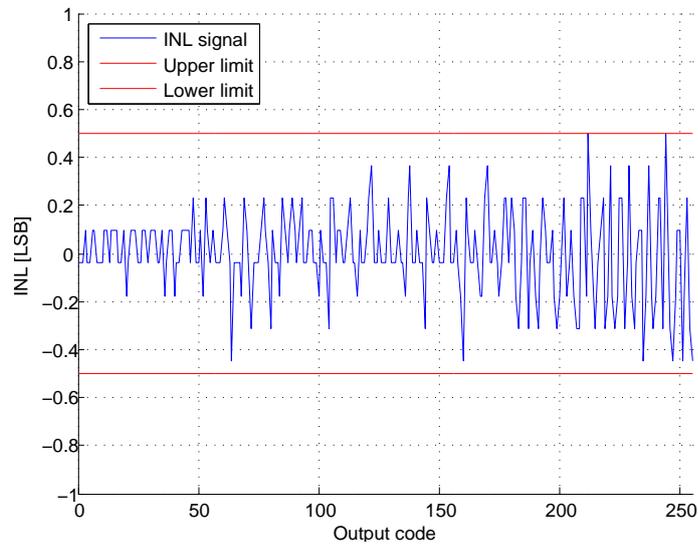


Figure 7.2: INL of the ADC

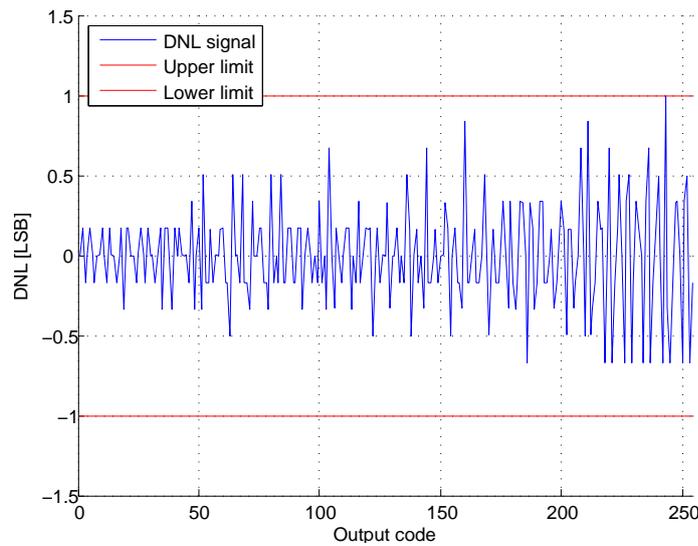


Figure 7.3: DNL of the ADC

## 7.2 Register

The register was simulated using an Ocean script provided in appendix at page 78. This automatically checks all register values for each clock-cycle

Table 7.1: Power Consumption

Module	Power[nA]	Estimated Power[nA]	Deviation
iDAC	263.93	256	+3.1%
iComp	161.79	200	-19.1%
Logic	105.76	100	+5.8%
<b>Total</b>	<b>531.49</b>	<b>556</b>	<b>-4.4%</b>

also accounting for the test-mode functionality. Operation through process corners and temperature was not performed because of the inherent robustness of the digital circuitry. However, power consumption was tested for the corners given in table 7.2 and is summarized in table 7.3.

Table 7.2: Selected sweep parameters and their values for corner analysis

Parameter	Values
Temperature	[20°C, 27°C, 40°C]
Supply Voltage	[1.2V, 1.8V, 2.0V]
FETs	[tm, wp, ws, wo, wz, fff, ssf]

Table 7.3: Static power consumption of the register module

Power	Min	Avg.	Max
Normal Mode	9.9n	16.1n	34.4n

## 7.3 Calibration Module

The bias generator module (iBias) was tested for all corners in table 7.2 and Monte Carlo analysis were performed for the typical process corner. Table 7.4 on the next page summarizes the mismatch and process simulations. A simultaneous process and mismatch simulation were not performed as this is a too pessimistic approach, assuming that there will be process deviations between closely placed transistors. All Monte Carlo analysis are performed using an average of 100 runs. Further, the specifications of the calibration currents are given in table 7.5 on page 52.

Table 7.4: Mismatch and process deviation of the reference currents

Reference Current	Mismatch		Process	
	$\mu[pA]$	$\sigma[pA]$	$\mu[pA]$	$\sigma[pA]$
$1nA$	997.8	42.9	999.5	0.4
$0.5nA$	504.1	20.8	502.2	0.3
$0.25nA$	253.2	11.8	251.4	0.3
$125pA$	125.6	6.4	125.4	0.2
$62.5pA$	62.1	3.3	61.7	0.1
$31.25pA$	29.7	1.7	30.1	0.1
biasDAC	999.7	35.4	1000.0	0.2
biasFq	999.2	38.3	1000.0	0.2
biasOta	30k0	912	30k	5.0

Table 7.5: Calibration current specifications

Reference	Parameter	$MSB$	$b6$	$b5$	$b4$	$b3$	$b2$	$DAC$	$F_s$	$Offset$
Max Current	$\mu_{Process}$	3.88n	1.79n	0.81n	0.40n	0.21n	89p	45p	3.89n	3.84n
	$\mu_{Mismatch}$	3.89n	1.83n	0.81n	0.4n	0.20n	91p	48p	3.90n	3.85n
	$\sigma_{Process}$	0.3p	77p	34p	20p	12p	5.0p	3.1p	0.5p	0.8p
	$\sigma_{Mismatch}$	172p	32p	15p	21p	12p	3.0p	5.0p	168p	180p
Zero Current	$\mu_{Process}$	12p	56p	12p	9.8p	10p	1.2p	6.1p	13p	12p
	$\mu_{Mismatch}$	6p	23p	11p	10p	6.1p	3.8p	3.5p	7.1p	8p
	$\sigma_{Process}$	0.0p	57p	24p	10p	10p	4.8p	3.4p	0.1p	0.0p
	$\sigma_{Mismatch}$	13p	12p	6.9p	6.4p	2.3p	1.1p	0.0p	10p	12p
Min Current	$\mu_{Process}$	-3.87n	-1.84n	-0.82n	-0.40n	-0.21n	-90p	-37p	-3.87n	-3.88n
	$\mu_{Mismatch}$	-3.90n	-1.87n	-0.83n	-0.40n	-0.22n	-83p	-43p	-3.81n	-3.87n
	$\sigma_{Process}$	1.2p	44p	20p	10p	7.1p	4.2p	1.2p	1.1p	0.8p
	$\sigma_{Mismatch}$	139p	44p	22p	11p	6.7p	2.9p	1.4p	128p	112p
INL		0.09LSB	0.10LSB	0.09LSB	0.11LSB	0.03LSB	0.13LSB	0.14LSB	0.09LSB	0.12LSB
DNL		0.13LSB	0.16LSB	0.10LSB	0.09LSB	0.05LSB	0.09LSB	0.09LSB	0.08LSB	0.12LSB
Gain error		0.08LSB	0.01LSB	0.00LSB	0.02LSB	0.01LSB	0.02LSB	0.02LSB	0.01TBD	0.01LSB
Offset		-10p	-23p	-11p	-5.6p	-2.2p	-11p	4.1p	-2.4p	5.0p

## 7.4 iDAC

The current mode DAC is tested using several test-benches. Ocean scripts in combination with Matlab scripts were developed for automated process of determining output characteristics and an example can be found in appendix at page 81 and 83 respectively. The static performance, i.e. INL, DNL, gain error and offset were tested using a diode-connected cascoded transistor load, the result given in table 7.6. Average power consumption was also measured during the simulation. The data were captured stepping the digital values from zero to maximum using  $4\mu s$  intervals. The output values were averaged between  $2\mu s$  and  $4\mu s$  using the approximation of a linear voltage change. Leakage when put in *SLEEP*-mode and turn on time was also tested. Process and Monte Carlo analysis similar to explained earlier of all the individual current paths have been performed, summarized in table 7.7 on the following page. This makes basis for the needed calibration range calculated using equation 3.9 and is given in the same table. Derivations could be found in appendix at page 73.

Table 7.6: Performance summary of the iDAC-module

Parameter	Measurement	Requirement[LSB].	Deviation
INL	0.14LSB	1/6	-16%
DNL .	0.01LSB	1/3	-97%
Gain error .	-1.67LSB	NA	NA
Offset	0.00LSB	NA	NA
Leakage in <i>SLEEP</i> -mode	$1.32nA$	NA	NA
Turn on time	$\sim 15ms$	NA	NA

## 7.5 iComp

To characterise the comparator module, DC-stepped transient simulations with resolution of 0.25LSB were used to derive the static performance similar to what was done with the ADC Core module. In addition, process and Monte Carlo simulations of the two-stage OTA as well as the whole module were performed using DC analysis. The result of this is given in table 7.8 on the next page. DC-characterisation of the comparator input impedance was

Table 7.7: Calibration current specifications

Parameter	MSB	b6	b5	b4	b3	b2	b1	LSB	Bias
$\mu_{Process}$ [nA]	128.00	64.00	32.00	16.00	8.00	4.00	2.00	1.00	8.00
$\mu_{Mismatch}$ [nA]	128.01	64.00	32.00	16.00	8.00	4.00	2.00	1.00	8.00
$\sigma_{Process}$ [pA]	28	3	0	5	0	0	0	0	0
$\sigma_{Mismatch}$ [pA]	1340	672	331	173	84	50	26	16	85
Calib range[nA]	$\pm 24.1$	$\pm 12.1$	$\pm 6.0$	$\pm 3.1$	$\pm 1.5$	$\pm 0.9$	$\pm 0.5$	$\pm 0.3$	$\pm 1.5$
# bits <sup>†</sup>	5+1	4+1	3+1	2+1	1+1	0	0	0	1+1

<sup>†</sup> Extra bit added for sign implementation, hence the +1

also tested to confirm the right levels for the multi-level input impedance stage in respect to the input current range, shown in figure 7.4 on the facing page. The Matlab code for this is in appendix at page 85.

Table 7.8: Mismatch and process deviations of the iComp-module

Module	Parameter	$\mu$	$\sigma$
iComp	Mismatch	-0.004LSB	3.9pA
	Process deviation	-0.002LSB	5.0pA
OTA	BW	3.97kHz	0.6kHz
	DC-gain	8.62kV/V	0.9kV/V

Table 7.9: Current comparator specifications

Parameter	Measurement[LSB]	Requirement[LSB]	Deviation
INL	0.15	1/6	-10%
DNL	0.11	1/3	-67%

## 7.6 Logic

The logic was verified through the simulation of the whole design in addition to each sub-module being simulated alone, results not included. The DC

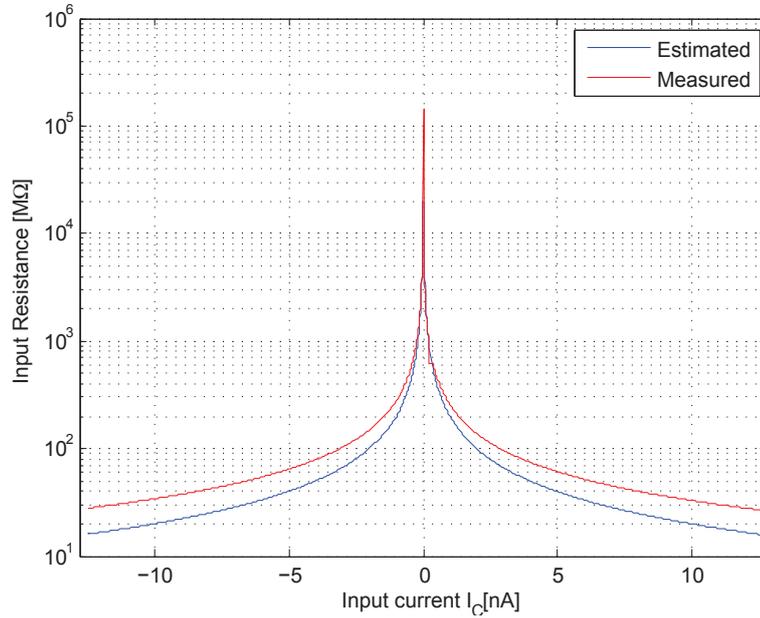


Figure 7.4: Input impedance of the current comparator

characterisation and average power (transient) were checked over process variations. The impulse generator was simulated using transient analysis over both mismatch and process variations to find required calibration range for the input bias. Frequency and MSB duration were extracted from the simulations, reported in table 7.10.

Table 7.10: Mismatch and process deviations of the ImpGen-module

Variable	Mismatch		Process variation	
	$\mu$	$\sigma$	$\mu$	$\sigma$
MSB Delay	$8.5\mu s$	$0.34\mu s$	$8.5\mu s$	$1.06\mu s$
Delay	$6.3\mu s$	$0.24\mu s$	$6.4\mu s$	$0.94\mu s$
ImpGen Power	24nA	1nA	24nA	4.24nA
Logic Power	105nA	2.3nA	104nA	12nA



# Chapter 8

## Analysis and Discussion

This section will analyse and discuss the design choices and obtained results before a reasoning towards the conclusion is made. However, a comparison of this ADC with several state-of-the-art low power, low area converters is provided in table 8.1 to give an idea of functionality and performance compared to other works.

Table 8.1: Comparison of low power ADCs

Parameter	[2]	[16]	[46]	[23]	This Work
$V_{DD}$	1V	0.6V	1V	1V	1.8
Technology	0.13 $\mu m$	0.18 $\mu m$	0.35 $\mu m$	90nm	0.18 $\mu m$
Mode	I-Mode	I-Mode	V-Mode	V-mode	I-Mode
Resolution	8	8	12	8	8
ENOB	7.6bit	8	10.2	7.8	8
Area [ $mm^2$ ]	0.005	0.004	1 <sup>†</sup>	0.054	0.078
Sampling rate	1kS/s	100kS/s	1kS/s	10MS/s	16kS/s
FOM (fJ/conv)	1314	16	435	12	242
Power [nA]	255	667	512	26k	550

<sup>†</sup> Area includes ECG circuit

From the table, it is evident that the performance of the ADC is not the best, nor the worst in any of the performance criteria. The FOM is a clear indication of this which ends up in the middle. Also looking at the fundamental

criteria power, this work again comes middle. Area, is though a bit larger. However, it should be noted that it is compared with the best ADCs regarding low power operation found at the moment. Another note regarding [16] which seems superior to the others, is that mismatch is not mentioned in the publication (section 4.3) and is suspected of using post-fabrication trimming to obtain the resolution.

## 8.1 Topology

Looking into the choice of topology, it is evident that current mode SAR ADC at the moment performs best when power and area is the criteria for an 8-bit resolution. However, if higher resolutions are required, significant power or area increments are inevitable because mismatch in sub-threshold operation is, as shown in section 4.2, exponential and thus much higher bias currents, larger calibration circuitry and/or much larger transistors are needed.

## 8.2 Architecture

Each sub-module and design choices will in the following be commented and discussed.

### 8.2.1 Current mode DAC

From the result section of the *iDAC* it is clear that a well functioning converter has been made. The INL and DNL error requirements are with good margin within the limits of the budget and will therefore presumably function well for the intended task. Further, both gain error and offset are negligible because of the selected topology and calibration module. It is therefore expected that the full DR will be available, and thus an effective number of bits (ENOB) close to the selected resolution. A further evaluation of table 7.6 reveals a substantial turn-on time for the module. This is because of the low current levels and high parasitic capacitances associated with the large transistors used for the current scaling. However, because the sleep-function is only intended to disable malfunctioning channels to save power, this has no impact. On the other hand, when turned off, a leakage current of  $1.32nA$  is

still drawn from the supply. This is a high value, considering that the leakage of a current mirror for the selected technology is in pico-range. Because of limited time however, further investigation of this was not conducted. The source of this is though likely to be a result of relatively short channel lengths of the implemented switches. The reason for choosing small sized switches was attributed the desire to implement low voltage drop in an on-state to enable lowering of the power supply down to  $1.2V$ . Because the switches are used for current steering, the difference in the impedance level is the only decisive factor for the current flow, resulting in low  $R_{off}$  requirements. If for the future, lower sleep-mode power consumption is required, this should be further investigated.

Converters are often tested dynamically to check operation over different input frequencies confirming the signal-to-noise ratio (SNR) is within specification for the required input frequency range. However, such simulations are very time consuming because of the high number of conversions required to make a good statistical approximation and were therefore not performed. However, because of the low levels of noise recoded in the simulation of the single current mirrors it was made no further attempts to test this. Concerning the static performance, because of the inherent switching employed by the SAR-algorithm, only one path will be turned on at a time. This will reduce some of the switching noise, presumably making the actual static performance better than the obtained results.

Looking at table 7.7, the mean value of each current path is right on spot and the deviations caused by process variations is almost negligible. The  $3\sigma$  current variation is therefore mostly due to mismatch. The large current deviations results in the MSB needed to be calibrated with 6 bits and could be claimed to be a disproportionate value in regard of the ADC's 8-bit resolution. However, due to the large mismatch using sub-threshold operation, this is a consequence and must be dealt with one way or another. Post-fabrication trimming using lasers could have removed the need of a calibration module, but was not an option due to the assignment requirements.

## 8.2.2 Current Comparator

From the results of the current comparator module it can be seen that the DC error for determining the digital output value has a increasable small deviation around the mean value. This confirms the theory in section 3.4, yielding high resolution at the expense of speed for high input impedance

current comparators. It also signals that the mismatch is dependent on the feedback loop and not the mismatch between the PMOS and NMOS transistor in the amplifier(inverter). When doing transient analysis, the resolution decreases significantly, but remains within the specifications of the error budget. Looking at figure 7.4 on page 55, the correlation between the theoretical input impedance and the implementation is clear. The multi-level impedance feedback loop forms a  $> 10G\Omega$  impedance for currents around zero, making currents as small as  $0.1nA$  able to create a  $1V$ -voltage drop over the resistor. Because of the exponential drop in resistance for increasing currents, the output will never go into saturation. However, a problem arise for larger currents as the amplifier used for the feedback has a final maximum input/output current and for this circuit design to be  $\sim 2^{N-1}nA$ . However, a small overhead was taken to allow some variation in the input bias.

The novelty of this module is high, applying a newly derived technique with never before seen modifications. This means that the architecture is not well tested and could prove withholding non-idealities not foreseen. A problem which arose under development was the supply voltage dependency, which makes the voltage supply range less flexible. Normally, very low power supplies can be applied to current mode circuits to reduce the power consumption, thus all the other circuit parts in this design were originally design for handling down to  $1V$  supplies. Adaptive biasing for the amplifiers could provide a solution to this, but is left to future works. However, since the neuron-chip supplies  $1.8V$ , this was only intended for testing purposes to see how small power consumption was obtainable. The architecture still leaves some room for lowering the supply voltage, though not confirmed below  $1.6V$ .

### 8.2.3 Logic

The circuit implementing the SAR-algorithm proved efficient, using half the current reported in [2]. The improvements compared to the original circuit [16] is difficult to predict as the reference does not mention the digital power consumption. However, from own simulations, the applied improvements reduced the power consumption significant, though a last minute change turned out to increase the supply current by  $\sim 40\%$ , up to the reported values in this thesis. The change was made due to layout considerations, as a digital library became available including a finished layout. The changes increased the leakage current significant due to under minimum sized transistors lengths. This was done to finish the layout in time.

From table 7.10 it can be seen that the impulse generation has a large standard deviation and thus need calibration. The large deviations are caused by using a nano-range current to charge a minimum sized capacitor, though the capacitor's contribution is the largest. The capacitor value is implemented to be  $30fF$  which is in close range of the parasitics. Using minimum size also increases edge effects, hence the large deviation. Because the linearity of the ADC is affected by the sampling frequency, it is important to get an as close match to the required speed as possible. This is to leave as little margin as possible to ensure that the sampling is finished in time. However, the asynchronous operation of the ADC enables a hand-shake implementation with the succeeding module which relaxes this requirement. Nevertheless, a full 6-bit calibration module was added to be able to test a large range of  $LSB/F_S$ - ratios for testing purposes.

## 8.2.4 Calibration

The calibration module turned out to work very well. The bias generation part using current division to create pico-range current sources had low deviations. From table 7.4 it can be seen that an approximate halving of the standard deviation for each current splitting and that the process deviations is in practice negligible. The absolute values do vary from perfect binary division, but is still within good margin of half of the minimum current. However, the mismatch of the  $1nA$  reference is larger than the value of the smallest current. Fortunately, for most of the chips, only small corrections will have to be made. In addition, overhead is taken to accommodate this.

The performance of the current generating modules is summarized in table 7.5. Here it can be seen that the range of the calibration currents corresponds well with the requirements of the DAC module. All the calibration currents modules have also a linear response to the control input. However, a note should be made out of the process deviation of the MSB branch being smaller than for bit 6, even though it has larger currents and DR. The reason for this is unclear, but could be due to leakage current deviations for the later topology. In terms of mismatch, the natural order is observed, scaling with DR relative to the smallest current. For the bit 2-6 topology, the current calibration extends further by having several extra bit combinations resulting the same output value. Deviations will therefore lead to very small steps between these values, and could be used for fine tuning. However, using values from both the positive (POS) and negative (NEG) register will lead

to extra power consumption and should be avoided for other than testing purposes. Using either sourcing or sinking currents will add no extra power consumption to the ADC as this only adjust the main DAC closer to an ideal value.

### 8.2.5 Calibration Register

The calibration register turned out to use more static power than anticipated, though operating well. The leakage current is high, and is to some extent caused by the digital library, used for the layout considerations. It is clear that the flip-flops provided with this package sacrifices power for space, which if looking at the layout at page 44 is unfortunate. However, this was done because of limited time. For future work, this should be implemented using low leakage devices.

When the selected technology matures and more standard components are offered from the foundry, exchanging the calibration module with fuses should be considered, as this will both minimize area and power consumption.

## 8.3 ADC core

The combination of the current mode DAC, current comparator and logic together implements an ADC using less than  $1\mu W$  at  $1.8V$ . Looking at table 7.1 shows that the power estimations were close to the actual result. The linearity of the module is within specifications looking at the figures in section 7.1, though it is evident that a 8-bit resolution approaches the limit for this topology and given technology. The implementation offers flexibility as both DR and  $F_s$  are dependent on selected bias, though not documented. Ripple in the bias was also tested because the PTAT band-gap reference current was reported to exhibited  $5pA$  interference at  $26MHz$  due to the wireless telemetry. This had no effect on the performance and is reasoned the low frequency response of the *iBias*-module.

## 8.4 Future Works

In addition to all suggested improvements or areas of investigation, future work will include testing the compliance between simulations and the actual chip. It is also suggested to further investigate the relation between resolution, bias current, DR and  $F_S$  to see if simple changes to the system can incorporate a higher resolution, higher sampling rate ADC for the neuron chip's calibration phase at the expense of increased power. Also the DAC functionality of the system should be investigated as the neuron chip requires a DAC for stimulation purposes in the future. If the module could be used for the digital interface of both recording and stimulation, much area and power will be saved.



# Chapter 9

## Conclusions

An 8-bit current mode SAR ADC operating at a  $16kHz$  sampling frequency using under  $1\mu W$  of power has been developed. A finished layout has been presented and the total circuit occupies less than  $0.078mm^2$ . The nominal operating voltage is  $1.8V$  although lower voltages can be applied for power reduction.  $INL < \pm 0.5$  and  $DNL < \pm 1$  is obtained, yielding no missing codes and thus linear operation. The production yield is less than  $3\sigma$ , obtained by a current calibration circuit connected to both the current mode DAC and all the biases. This cancels the relatively large mismatch caused by weak inversion operation of the current mirrors.

A newly developed current comparator is presented offering high resolution and fast settling relative to the current level by using an inverting amplifier connected in a multi-level impedance feedback loop. The design features some voltage scaling issues, but is well functioning and completable with state-of-the-art current comparators. All this, in combination with a low power, asynchronous SAR-algorithm implementation make up an ADC within the given and partly derived specifications.



# References

- [1] A. Agarwal, Y.B. Kim, and S. Sonkusale. Low power current mode adc for cmos sensor ic. In *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*, pages 584 – 587 Vol. 1, may 2005.
- [2] S. Al-Ahdab, R. Lotfi, and W.A. Serdijn. A 1-v 225-nw 1ks/s current successive approximation adc for pacemakers. *Proc. IEEE PRIME*, pages 1–4, 2010.
- [3] L.N. Alves and R.L. Aguiar. Frequency behavior of classical current mirrors [cmos]. In *Electronics, Circuits and Systems, 2002. 9th International Conference on*, volume 1, pages 189 – 192 vol.1, 2002.
- [4] L.N. Alves and R.L. Aguiar. Noise performance of classical current mirrors. In *Electronics, Circuits and Systems, 2002. 9th International Conference on*, volume 1, pages 277 – 280 vol.1, 2002.
- [5] R.J. Baker. *CMOS: Circuit Design, Layout, and Simulation*. IEEE Press Series on Microelectronic Systems. John Wiley & Sons, 2010.
- [6] K. Bult and G.J.G.M. Geelen. An inherently linear and compact most-only current division technique. *Solid-State Circuits, IEEE Journal of*, 27(12):1730 –1735, dec 1992.
- [7] Moo Sung Chae, Zhi Yang, M.R. Yuce, Linh Hoang, and W. Liu. A 128-channel 6 mw wireless neural recording ic with spike feature extraction and uwb transmitter. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 17(4):312 –321, aug. 2009.
- [8] Sun-Il Chang and Euisik Yoon. A low-power area-efficient 8 bit sar adc using dual capacitor arrays for neural microsystems. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 1647 –1650, sept. 2009.

- [9] You-Kuang Chang, Chao-Shiun Wang, and Chorng-Kuang Wang. A 8-bit 500-ks/s low power sar adc for bio-medical applications. In *Solid-State Circuits Conference, 2007. ASSCC '07. IEEE Asian*, pages 228–231, nov. 2007.
- [10] Lu Chen, Bingxue Shi, and Chun Lu. A high speed/power ratio continuous-time cmos current comparator. 2:883–886, 2000.
- [11] Ming-Jer Chen, Jih-Shin Ho, and Dang-Yang Chang. Optimizing the match in weakly inverted mosfet’s by gated lateral bipolar action. *Electron Devices, IEEE Transactions on*, 43(5):766–773, may 1996.
- [12] J.B. Hughes C.Toumazou and N.C. Battersby. *Switched-Currents an analogue technique for digital technology*. Peter Peregrinus Ltd, SG1 2AY, UK, 1993.
- [13] E. Culurciello, R. Etienne-Cummings, and K.A. Boahen. A biomorphic digital image sensor. *IEEE Journal of Solid-State Circuits*, 38(2):281–294, 2003.
- [14] R. del Rio-Fernandez, G. Linan-Cembrano, R. Dominguez-Castro, and A. Rodriguez-Vazquez. A mismatch-insensitive high-accuracy high-speed continuous-time current comparator in low voltage cmos. *Proc. IEEE Analog and Mixed IC Design*, pages 111–116, 1997.
- [15] T. Delbruck and A. van Schaik. Bias current generators with wide dynamic range. In *Circuits and Systems, 2004. ISCAS '04. Proceedings of the 2004 International Symposium on*, volume 1, pages I–337 – I–340 Vol.1, may 2004.
- [16] R. Dlugosz, V. Gaudet, and K. Iniewski. Flexible ultra low power successive approximation analog-to-digital converter with asynchronous clock generator. *Proc. IEEE CCECE*, pages 1649–1652, 2007.
- [17] R. Dlugosz and K. Iniewski. Ultra low power current-mode algorithmic analog-to-digital converter implemented in 0.18um cmos technology for wireless sensor network. *Proc. IEEE MIXDES*, pages 401–406, 2006.
- [18] P.G. Drennan and C.C. McAndrew. Understanding mosfet mismatch for analog design. *Solid-State Circuits, IEEE Journal of*, 38(3):450 – 456, mar 2003.
- [19] A. Eftekhar, S. E. Paraskevopoulou, and T. G. Constandinou. Towards a next generation neural interface: Optimizing power, bandwidth and data quality. *Proc. IEEE BioCAS*, pages 122–125, 2010.

- [20] M. El Mustapha Ait Yakoub, M. Sawan, and C. Thibeault. A neuromimetic ultra low-power adc for bio-sensing applications. In *Circuits and Systems and TAISA Conference, 2009. NEWCAS-TAISA '09. Joint IEEE North-East Workshop on*, pages 1–4, 28 2009-july 1 2009.
- [21] Maneesha Gupta, Prashant Aggarwal, Pritender Singh, and Naveen Jindal. Low voltage current mirrors with enhanced bandwidth. *Analog Integrated Circuits and Signal Processing*, 59:97–103, 2009. 10.1007/s10470-008-9241-2.
- [22] Baard Haaheim. An 8bit cm-sar adc for neural spike recording. [http://www.haaheim.org/papers/2010/term\\_project.pdf](http://www.haaheim.org/papers/2010/term_project.pdf), 2010.
- [23] P.J.A. Harpe, C. Zhou, Yu Bi, N.P. van der Meijs, Xiaoyan Wang, K. Philips, G. Dolmans, and H. de Groot. A 26  $\mu$  w 8 bit 10 ms/s asynchronous sar adc for low energy radios. *Solid-State Circuits, IEEE Journal of*, 46(7):1585–1595, july 2011.
- [24] R. R. Harrison, P. T. Watkins, R. J. Kier, R. O. Lovejoy, D. J. Black, B. Greger, and F. Solzbacher. A low-power integrated circuit for a wireless 100-electrode neural recording system. *Solid-State Circuits, IEEE Journal of*, 42(1):123–133, jan. 2007.
- [25] R.R. Harrison. The design of integrated circuits to observe brain activity. *Proc. IEEE*, 96(7):1203–1216, 2008.
- [26] David Johns and Ken Martin. *Analog integrated circuit design*. Wiley, New York, 1997.
- [27] Cao Junmin, Chen Zhongjian, Lu Wengao, and Zhao Baoying. A low power 12-bit 20msamples/s pipelined adc. In *2009 International Conference on Signal Processing Systems*, pages 77–80, may 2009.
- [28] S. Khucharoensin and V. Kasemsuwan. Robust high-speed low input impedance cmos current comparator. In *Circuits and Systems, 2004. MWSCAS '04. The 2004 47th Midwest Symposium on*, volume 1, pages I–93–6 vol.1, july 2004.
- [29] Jinseok Koh, Yunyoung Choi, and G. Gomez. A 66db dr 1.2v 1.2mw single-amplifier double-sampling 2nd-order delta; sigma; adc for wcdma in 90nm cmos. In *Solid-State Circuits Conference, 2005. Digest of Technical Papers. ISSCC. 2005 IEEE International*, pages 170–591 Vol. 1, feb. 2005.

- [30] Chi-Hung Lin and M. Ismail. A low-voltage cmos rail-to-rail class-ab input/output opamp with slew-rate and settling enhancement. In *Circuits and Systems, 1998. ISCAS '98. Proceedings of the 1998 IEEE International Symposium on*, volume 1, pages 448–450 vol.1, may-3 jun 1998.
- [31] B. Linares-Barranco and T. Serrano-Gotarredona. On the design and characterization of femtoampere current-mode circuits. *IEEE Journal of Solid-State Circuits*, 38(8):1353–1363, 2003.
- [32] D. McDonagh, O. Eljamaly, and A.J. Burdett. 1v 14uw switched-opamp delta; sigma; adc for bioelectric data aquisition. In *Medical Devices and Biosensors, 2007. ISSS-MDBS 2007. 4th IEEE/EMBS International Summer School and Symposium on*, pages 147–150, aug. 2007.
- [33] R.B. Millar. *Maximum Likelihood Estimation and Inference: With Examples in R, SAS and ADMB*. Statistics in Practice. John Wiley & Sons, 2011.
- [34] K. Nagaraj. Cmos amplifiers incorporating a novel slew rate enhancement technique. In *Custom Integrated Circuits Conference, 1990., Proceedings of the IEEE 1990*, pages 11.6/1–11.6/5, may 1990.
- [35] J.H. Nielsen and E. Bruun. A low-power 10-bit continuous-time cmos sigma; delta; a/d converter. In *Circuits and Systems, 2004. ISCAS '04. Proceedings of the 2004 International Symposium on*, volume 1, pages I–417–20 Vol.1, may 2004.
- [36] S. O'Driscoll and T.H. Meng. Adaptive adc design for neuro-prosthetic interfaces: base adc cell. In *Circuit Theory and Design, 2005. Proceedings of the 2005 European Conference on*, volume 1, pages I/301–I/304 vol. 1, aug.-2 sept. 2005.
- [37] D. Panescu. Emerging technologies. *Engineering in Medicine and Biology Magazine, IEEE*, 27(5):100–113, september-october 2008.
- [38] Pierre-Yves Robert, Benoit Gosselin, Amer Elias Ayoub, and Mohamad Sawan. An ultra-low-power successive-approximation-based adc for implantable sensing devices. In *Circuits and Systems, 2006. MWSCAS '06. 49th IEEE International Midwest Symposium on*, volume 1, pages 7–11, aug. 2006.
- [39] C. Saint and J. Saint. *IC mask design: essential layout techniques*. McGraw-Hill professional engineering. McGraw-Hill, 2002.

- [40] M.D. Scott, B.E. Boser, and K.S.J. Pister. An ultralow-energy adc for smart dust. *Solid-State Circuits, IEEE Journal of*, 38(7):1123 – 1129, july 2003.
- [41] F. Shahrokhi, K. Abdelhalim, D. Serletis, P.L. Carlen, and R. Genov. The 128-channel fully differential digital integrated neural recording and stimulation interface. *IEEE Trans. Biomedical Circuits and Systems*, 4(3):149–161, 2010.
- [42] M. van Elzakker, E. van Tuijl, P. Geraedts, D. Schinkel, E. Klumperink, and B. Nauta. A 10-bit charge-redistribution adc consuming 1.9  $\mu\text{w}$  at 1 ms/s. *Solid-State Circuits, IEEE Journal of*, 45(5):1007–1015, may 2010.
- [43] Z. Wang. Analytical determination of output resistance and dc matching errors in mos current mirrors. *Circuits, Devices and Systems, IEE Proceedings G*, 137(5):397–404, oct 1990.
- [44] M.H. Zarifi and J. Frounchi. A low-power, small-area 1 msample/sec adc for neural-signal recording systems in 0.35-  $\mu\text{m}$  cmos. In *Neural Engineering, 2009. NER '09. 4th International IEEE/EMBS Conference on*, pages 391–394, 29 2009-may 2 2009.
- [45] Fan-Gang Zeng, S. Rebscher, W. Harrison, Xiaoan Sun, and Haihong Feng. Cochlear implants: System design, integration, and evaluation. *Biomedical Engineering, IEEE Reviews in*, 1:115–142, 2008.
- [46] Xiaodan Zou, Xiaoyuan Xu, Libin Yao, and Yong Lian. A 1-v 450-nw fully integrated programmable biomedical sensor interface chip. *IEEE Journal of Solid-State Circuits*, 44(4):1067–1077, 2009.



# Appendix A

## Derivations

### A.1 DAC mismatch requirements

From simulations, the standard deviation approximate halves for each bit towards LSB for the given current level. This leads to the following relation

$$\sigma_{MSB} \simeq 2\sigma_{bit7} \simeq 2^n \sigma_{bit(N-n-1)} \quad (\text{A.1})$$

Assuming full correlation, each current branch in the iDAC will need to be calibrated within a margin argued as follows.

Looking at both sides of the normal distributed standard deviation, a  $6\sigma$  current range must be within the range of the calibration current. In addition, the calibration must ensure a resolution of the  $1/3 \cdot 0.5LSB$  as this defines the error budget. Because of the halving of the standard deviation towards LSB, the error for each current branch is scaled similar, shown in table A.1 (Original). However, to relax the requirements for the lower bits, the MSB error is halved, implemented as an extra calibration bit. The next equations then define the needed number of calibration bits.

Table A.1: Assigned error to each iDAC branch

	MSB	b7	b6	b5	b4	b3	b2	LSB
Original	1/2	1/4	1/8	1/16	1/32	1/64	1/128	1/256
Relaxed	1/4	1/4	1/8	1/8	1/16	1/16	1/16	1/16

$$6\sigma = range \cdot resolution \quad (A.2)$$

$$= range \cdot \frac{LSB}{2} \frac{1}{3} \quad (A.3)$$

$$range = \frac{24\sigma}{LSB} = \frac{24\sigma}{1n} [A] \quad (A.4)$$

The following needs to be satisfied to ensure  $3\sigma$  production yield and leads to the requirements in table 7.7 on page 54.

$$2^{bit} > range \quad (A.5)$$

# Appendix B

## Ocean Scripts

### B.1 Current Mirror

```
1 ; -----
2 ; CM - dc mismatch
3 ;
4 ; Loop L, W and current
5 ; (or other variables) for MC analysis
6 ; to gain extensive data sets.
7 ; -----
8
9 ; -----
10 ; START EDIT
11 ; -----
12 out=outfile("~/SimRes/cm/mc.dat" "w")
13 nob =5
14 lengder = 4
15 bredder = 4
16 declare( nob[nob])
17 nobvalues = '(0.25n 0.5n 1.0n 10.0n 100.0n)
18
19 ; -----
20 ; STOP EDIT
21 ; -----
22
23 for( aa 0 length(nob)-1
24     nob[aa] = nth(aa nobvalues)
25 )
26 fprintf(out,"ibias , length ,width , mu, sd\n")
27 for(bias 0 nob-1
28
29     for(tt 1 lengder
30
31         for(t2 1 bredder
32
33             lengde = 15*tt
34
35             ocnWaveformTool( 'wavescan )
36             simulator( 'spectre )
37             design( "/ibe/users/bard/h18a4/Sim/test_cm/spectre/schematic/netlist/netlist")
38             resultsDir( "/ibe/users/bard/h18a4/Sim/test_cm/spectre/schematic" )
39             modelFile(
40                 '/usr/local/cadence/kits/ams/3.77/spectre/h18/soac/design.scs' "noimc")
41                 '/usr/local/cadence/kits/ams/3.77/spectre/h18/soac/res.scs' "restm")
42                 '/usr/local/cadence/kits/ams/3.77/spectre/h18/soac/cap.scs' "capm")
43                 '/usr/local/cadence/kits/ams/3.77/spectre/h18/soac/bip.scs' "bipm")
44                 '/usr/local/cadence/kits/ams/3.77/spectre/h18/soac/esddi.scs' "esddimc")
45                 '/usr/local/cadence/kits/ams/3.77/spectre/h18/soac/cmos73.scs' "cmosmc")
46             )
47             definitionFile(
48                 "/usr/local/cadence/kits/ams/3.77/spectre/h18/soac/mcparams.scs"
```

```

49 )
50 analysis('dc ?saveOppoint t )
51 desVar( "wid" 500n )
52 desVar( "w" t2 )
53 desVar( "len" 180n )
54 desVar( "l" lengde )
55 desVar( "ibias" float(nobs[bias]) )
56 option( 'temp "37.0" )
57
58 save( 'i "/V18/PLUS" )
59 temp( 37.0 )
60 monteCarlo( ?numIters "50" ?startIter "1"
61             ?analysisVariation 'mismatch ?sweptParam "None"
62             ?sweptParamVals "27" ?saveData t
63             ?nomRun "yes" ?append nil
64             ?saveProcessParams t
65 )
66 monteExpr( "out" "IDC(\\"/V18/PLUS\\")" )
67 monteRun()
68 fprintf(out "%5.12f" float(nobs[bias]))
69 fprintf(out ",")
70 fprintf(out "%d" lengde)
71 fprintf(out ",")
72 fprintf(out "%d" t2)
73 fprintf(out ",")
74 fprintf(out "%5.14f" average(i("/V18/PLUS" ?result "mc1_dcOp-montecarlo")))
75 fprintf(out ",")
76 fprintf(out "%5.14f" stddev(i("/V18/PLUS" ?result "mc1_dcOp-montecarlo")))
77 fprintf(out "\n")
78 )
79 )
80 )
81 close(out)

```

Listing B.1: Ocean code for mismatch data capturing

## B.2 ADC Core

```

1 ; -----
2 ; ADC Core - trans
3 ;
4 ; Simulate the ADCCore module using a ramping
5 ; input current.
6 ; -----
7 ocnWaveformTool( 'wavescan )
8 simulator( 'spectre )
9 ; local drive for fast simulations
10 design( "/special/baard/tb_adc_core/spectre/schematic/netlist/netlist" )
11 resultsDir( "/special/baard/tb_adc_core/spectre/schematic" )
12 modelFile(
13     "/usr/local/cadence/kits/ams/3.77/spectre/h18/soac/design.scs" "noitm")
14     "/usr/local/cadence/kits/ams/3.77/spectre/h18/soac/res.scs" "restm")
15     "/usr/local/cadence/kits/ams/3.77/spectre/h18/soac/cap.scs" "captm")
16     "/usr/local/cadence/kits/ams/3.77/spectre/h18/soac/bip.scs" "bipm")
17     "/usr/local/cadence/kits/ams/3.77/spectre/h18/soac/esddi.scs" "esdditm")
18     "/usr/local/cadence/kits/ams/3.77/spectre/h18/soac/cmos73.scs" "cmostm")
19 )
20 definitionFile(
21     "/usr/local/cadence/kits/ams/3.77/spectre/h18/soac/mcparams.scs"
22 )
23
24 ; -----
25 ; START EDIT
26 ; -----
27
28 nop = 1024 ; Number of data points
29 test_input = 0 ; Offset
30 testmode = 0 ; Enables test mode if sat to 1.8
31 bias = 1n ; LSB/input bias level
32 step = 0.25n ; Step size
33 bias_fq = 1.5n ; sampling frequency bias
34 vsupply = 1.8 ; voltage supply
35 temperature = 37.0 ; temperature
36

```

```

37 ; -----
38 ; STOP EDIT
39 ; -----
40 ; set up simulation
41 ana_time = nop*61u+11u
42 analysis( 'tran ?stop ana_time )
43 desVar( "test_input" test_input )
44 desVar( "testmode" testmode )
45 desVar( "vsupply" vsupply )
46 desVar( "bias_fq" bias_fq )
47 desVar( "bias" bias )
48 desVar( "step" step )
49 option( 'temp' temperature
50 )
51 ; save only selected nodes to save memory
52 saveOption( 'currents "selected" )
53 saveOption( 'pwr "none" )
54 saveOption( 'save "selected" )
55 ; enable turbo and 12 parallel CPUs
56 option( ?categ 'turboOpts 'proc_affinity "0-11"
57 'uniMode "Turbo"
58 )
59 ; select current and voltages to save
60 save( 'v "/DUT/vcomp" "/DUT/van_out" "/DUT/LOGIC/yline" "/DO7" "/DO6" "/DO5" "/DO4" "/
DO3" "/DO2" "/DO1" "/DO0" )
61 save( 'i "/DUT/iin" "/DUT/IDAC/iout" "/DUT/TN0/D" "/DUT/ICOMP/iin" "/DUT/ICOMP/VDDA" "/
DUT/LOGIC/VDDA" "/DUT/IDAC/VDDA" "/VANALOG/MINUS" "/VDDCORELIB/MINUS" )
62
63 temp( 37.0 )
64 run()
65
66 ; PRINT to FILE
67 ; data.dat - DO of the ADC together with a timestamp and the input current
68 ; power.dat - short resume of power consumption
69 ; diagnostics.dat - prints the selected nodes at every lus
70
71 selectResult( 'tran )
72 out=outfile( "~/SimRes/adc/data.dat" "w" ) ; the output file
73 fprintf(out, "time, b7, b6, b5, b4, b3, b2, b1, b0, iin\n" )
74 for( tt 1 nop
75 time = (61*tt*0.000001) + (10*0.000001)
76 fprintf(out, "%5.7f" time)
77 fprintf(out, ",")
78 fprintf(out, "%5.3f" value(VT("/DO7"), time))
79 fprintf(out, ",")
80 fprintf(out, "%5.3f" value(VT("/DO6"), time))
81 fprintf(out, ",")
82 fprintf(out, "%5.3f" value(VT("/DO5"), time))
83 fprintf(out, ",")
84 fprintf(out, "%5.3f" value(VT("/DO4"), time))
85 fprintf(out, ",")
86 fprintf(out, "%5.3f" value(VT("/DO3"), time))
87 fprintf(out, ",")
88 fprintf(out, "%5.3f" value(VT("/DO2"), time))
89 fprintf(out, ",")
90 fprintf(out, "%5.3f" value(VT("/DO1"), time))
91 fprintf(out, ",")
92 fprintf(out, "%5.3f" value(VT("/DO0"), time))
93 fprintf(out, ",")
94 fprintf(out, "%5.14f" value(IT("/DUT/iin"), time-61u))
95 fprintf(out, "\n")
96 )
97 close(out)
98
99 power=outfile( "~/SimRes/adc/power.dat" "w" )
100 fprintf(power, "ADC - Power\n")
101 tot = average( IT("/VANALOG/MINUS") ) * 1G + average( IT("/VDDCORELIB/MINUS") ) * 1G
102 fprintf(power, "\nTOTAL: \t\t,")
103 fprintf(power, "%5.3f" tot)
104 fprintf(power, "\nTotal cmhv7sf: \t\t,")
105 fprintf(power, "%5.3f" average( IT("/VANALOG/MINUS") ) * 1G)
106 fprintf(power, "\nTotal CORELIB: \t\t,")
107 fprintf(power, "%5.3f" average( IT("/VDDCORELIB/MINUS") ) * 1G)
108 fprintf(power, "\niCOMP cmhv7sf: \t\t,")
109 fprintf(power, "%5.3f" average( IT("/DUT/ICOMP/VDDA") ) * 1G)
110 fprintf(power, "\niDAC cmhv7sf: \t\t,")
111 fprintf(power, "%5.3f" average( IT("/DUT/IDAC/VDDA") ) * 1G)
112 fprintf(power, "\nLogic cmhvsf: \t\t,")
113 fprintf(power, "%5.3f" average( IT("/DUT/LOGIC/VDDA") ) * 1G)
114 close(power)
115
116 diag=outfile( "~/SimRes/adc/diagnostics.dat" "w" )
117 selectResult( 'tran )

```

```

118 m0=value(IT("/DUT/iin") 0.9u ?period 1u ?xName "time")
119 m1=value(IT("/DUT/TN0/D") 0.9u ?period 1u ?xName "time")
120 m1=value(IT("/DUT/IDAC/iout") 0.9u ?period 1u ?xName "time")
121 m2=value(IT("/DUT/ICOMP/iin") 0.9u ?period 1u ?xName "time")
122 m3=value(VT("/DUT/vcomp") 0.9u ?period 1u ?xName "time")
123 m4=value(VT("/DUT/van_out") 0.9u ?period 1u ?xName "time")
124 m5=value(VT("/DUT/LOGIC/yline") 0.9u ?period 1u ?xName "time")
125 ocnPrint( ?output diag ?numberNotation 'scientific ?numSpaces 1 m0 m1 m2 m3 m4 m5)
126 close(diag)

```

Listing B.2: Ocean code for ADC Core module

## B.3 Calibration Register

```

1 ; -----
2 ; REG - trans
3 ;
4 ; Simulate the REG module using a pseudo
5 ; random input signal, clocking the data in.
6 ; The data is compared with the expected values.
7 ; The test includes testing for test-mode as well
8 ; -----
9 ocnWaveformTool( 'wavescan )
10 simulator( 'spectre )
11 design( "/ibe/users/bard/h18a4/Sim/tb_reg_top/spectre/schematic/netlist/netlist" )
12 resultsDir( "/ibe/users/bard/h18a4/Sim/tb_reg_top/spectre/schematic" )
13 modelFile(
14     '( "/usr/local/cadence/kits/ams/3.77/spectre/h18/soac/design.scs" "noitm" )
15     '( "/usr/local/cadence/kits/ams/3.77/spectre/h18/soac/res.scs" "restm" )
16     '( "/usr/local/cadence/kits/ams/3.77/spectre/h18/soac/cap.scs" "capm" )
17     '( "/usr/local/cadence/kits/ams/3.77/spectre/h18/soac/bip.scs" "bipm" )
18     '( "/usr/local/cadence/kits/ams/3.77/spectre/h18/soac/esddi.scs" "esdditm" )
19     '( "/usr/local/cadence/kits/ams/3.77/spectre/h18/soac/emos73.scs" "cmosm" )
20 )
21 definitionFile(
22     "/usr/local/cadence/kits/ams/3.77/spectre/h18/soac/mcparams.scs"
23 )
24 ; -----
25 ; START EDIT
26 ; -----
27
28 shifts = 32
29 simtime = shifts*200n
30 vsupply = 1.8
31 temperature = 27.0
32
33 ; -----
34 ; STOP EDIT
35 ; -----
36
37 analysis( 'tran ?stop simtime ?errpreset "liberal" )
38 desVar( "vsupply" vsupply )
39
40 option( ?categ 'turboOpts 'psrFmax 1
41     'psrOption "Default"
42     'numThreads 1
43     'mtOption "Auto"
44     'errorLevel "Do not override"
45     'turboSwitch t
46 )
47 saveOption( 'currents "selected" )
48 saveOption( 'pwr "none" )
49 saveOption( 'save "selected" )
50 save( 'v "/input" "/clk" "/mode" "/ctrl1<7>" "/ctrl1<6>" "/ctrl1<5>" "/ctrl1<4>" "/ctrl1
    <3>" "/ctrl1<2>" "/ctrl1<1>" "/ctrl1<0>" "/calib4<0>" "/calib4<1>" "/calib4<2>" "/
    calib4<3>" "/calib4<4>" "/calib4<5>" "/calib4<6>" "/calib4<7>" )
51 save( 'i "/VDD/MINUS" )
52
53 temp( temperature )
54 ;run()
55 selectResult( 'tran )
56
57 out=outfile( "~/SimRes/digital/reg_verification.dat" "w" )
58 fprintf(out "-----\n")

```

```

59 fprintf(out "Register Check.\n")
60 fprintf(out "-----\n")
61 status = 0
62
63 ;Ctrl1<7> Check
64 for(tt 1 shifts
65
66     now = 200n*tt-50n
67     prev = 200n*tt-100n
68
69     test = value(VT("/ctrl1<7>"),now)
70     test2 = value(VT("/input"),prev)
71
72     printf( "Checking \"/ctrl1<7>\n" at %5.0fns " now*1G)
73     printf( "%5.1f vs %5.0fns %5.1f - " test prev*1G test2 )
74
75     if( round(test) != round(test2) then
76         printf( "Failed\n")
77         fprintf(out "Reg:\t %f \tat: %5.1f ns\n" test now*1G)
78         fprintf(out "Input:\t %f \tat: %5.1f ns\n" test2 prev*1G)
79         status = 1
80     else
81         ;Everything is OK
82         printf( "OK\n")
83     )
84 )
85 ;Ctrl1 Check
86 for(jj 0 6
87     for(tt 1 shifts
88         now = 200n*tt-50n
89         prev = 200n*tt-100n
90
91         testName = strcat("/ctrl1<" sprintf(nil "%d" jj) ">")
92         testName2 = strcat("/ctrl1<" sprintf(nil "%d" jj+1) ">")
93         test = value(VT(testName),now)
94         test2 = value(VT(testName2),prev)
95
96         printf( "Checking %L at %5.0fns " testName now*1G)
97         printf( "%5.1f vs %5.0fns %5.1f - " test prev*1G test2 )
98
99
100     ;printf( "Checking %L at %5.0f ns - " testName now*1G)
101     ;printf( "%5.1f vs %5.1f - " test test2 )
102
103     if( round(test) != round(test2) then
104         printf( "Failed\n")
105         status = 1
106         fprintf(out "Reg:\t %f \tat: %5.1f ns\n" test now*1G)
107         fprintf(out "Input:\t %f \tat: %5.1f ns\n" test2 prev*1G)
108     else
109         printf( "OK\n")
110     )
111 )
112 )
113 ;

```

---

```

114 ;calib4<7> Check
115 for(tt 1 shifts
116     now = 200n*tt-50n
117     prev = 200n*tt-100n
118
119     if( value(VT("/mode") now) > vsupply/2 then
120         ;test mode, the bit should not change
121         testName = strcat("/calib4<7>")
122         testName2 = strcat("/calib4<7>")
123         test = value(VT(testName),now)
124         test2 = value(VT(testName2),prev)
125         printf( "Checking %L at %5.0fns " testName now*1G)
126         printf( "%5.1f vs %5.0fns %5.1f - " test prev*1G test2 )
127
128         if(round(test) != round(test2) then
129             ;error, the bit is not stored
130             printf( "Failed(2)\n")
131             status = 1
132         else
133             printf( "OK\n")
134             fprintf(out "Reg:\t %f \tat: %5.1f ns\n" test now*1G)
135             fprintf(out "Input:\t %f \tat: %5.1f ns\n" test2 prev*1G)
136         )
137     )
138 else
139     test = value(VT("/calib4<7>"),now)

```

```

140     test2 = value(VT("/ctrl1<0>"),prev)
141     printf( "Checking \"/calib4<7>\\" at %5.0fns " now*1G)
142     printf( "%5.1f vs %5.0fns %5.1f - " test prev*1G test2 )
143     if( round(test) != round(test2) then
144         printf( "Failed\n")
145         fprintf(out "Reg:\t %f \tat: %5.1f ns\n" test now*1G)
146         fprintf(out "Input:\t %f \tat: %5.1f ns\n" test2 prev*1G)
147         status = 1
148     else
149         ;Everything is OK
150         printf( "OK\n")
151     )
152 )
153 )
154 ;calib7 Check
155 for(jj 0 6
156     for(tt 1 shifts
157         now = 200n*tt-50n
158         prev = 200n*tt-100n
159
160         if(value(VT("/mode") now) > vsupply/2 then
161             ;test mode, the bit should not change
162             testName = strcat("/calib4<" sprintf(nil "%d" jj) ">")
163             testName2 = strcat("/calib4<" sprintf(nil "%d" jj) ">")
164             test = value(VT(testName),now)
165             test2 = value(VT(testName2),prev)
166             printf( "Checking %L at %5.0fns " testName now*1G)
167             printf( "%5.1f vs %5.0fns %5.1f - " test prev*1G test2 )
168
169             if(round(test) != round(test2) then
170                 ;error, the bit is not stored
171                 printf( "Failed (2)\n")
172                 status = 1
173             else
174                 printf( "OK\n")
175                 fprintf(out "Reg:\t %f \tat: %5.1f ns\n" test now*1G)
176                 fprintf(out "Input:\t %f \tat: %5.1f ns\n" test2 prev*1G)
177             )
178         else
179
180             testName = strcat("/calib4<" sprintf(nil "%d" jj) ">")
181             testName2 = strcat("/calib4<" sprintf(nil "%d" jj+1) ">")
182             test = value(VT(testName),now)
183             test2 = value(VT(testName2),prev)
184
185             printf( "Checking %L at %5.0fns " testName now*1G)
186             printf( "%5.1f vs %5.0fns %5.1f - " test prev*1G test2 )
187
188
189             ;printf( "Checking %L at %5.0f ns - " testName now*1G)
190             ;printf( "%5.1f vs %5.1f - " test test2 )
191
192             if( round(test) != round(test2) then
193                 printf( "Failed (1)\n")
194                 status = 1
195                 fprintf(out "Reg:\t %f \tat: %5.1f ns\n" test now*1G)
196                 fprintf(out "Input:\t %f \tat: %5.1f ns\n" test2 prev*1G)
197             else
198                 printf( "OK\n")
199             )
200         )
201     )
202 )
203 ;

```

---

```

204 ; PRINT STATUS
205 if( status==0 then
206
207     printf("\nALL CHECKS PASSED\n")
208     fprintf(out " - Check: OK\n")
209
210 else
211     printf("\n** ERROR REPORT WRITTEN TO FILE **")
212     fprintf(out "-----\n")
213     fprintf(out " - Check: Failed\n")
214 )
215 fprintf(out "-----\n\n")
216 close(out)

```

Listing B.3: Ocean code for testing of REG module

## B.4 iDAC

```
1 ; -----
2 ; iDAC - trans
3 ;
4 ; Transient analysis of iDAC ramping digital input
5 ; -----
6 ocnWaveformTool( 'wavescan' )
7 simulator( 'spectre' )
8 design( "/ibe/users/bard/h18a4/Sim/tb_idac/spectre/schematic/netlist/netlist" )
9 resultsDir( "/ibe/users/bard/h18a4/Sim/tb_idac/spectre/schematic" )
10 modelFile(
11     "/usr/local/cadence/kits/ams/3.77/spectre/h18/soac/design.scs" "noitm" )
12     "/usr/local/cadence/kits/ams/3.77/spectre/h18/soac/res.scs" "restm" )
13     "/usr/local/cadence/kits/ams/3.77/spectre/h18/soac/cap.scs" "capm" )
14     "/usr/local/cadence/kits/ams/3.77/spectre/h18/soac/bip.scs" "biptm" )
15     "/usr/local/cadence/kits/ams/3.77/spectre/h18/soac/esddi.scs" "esdditm" )
16     "/usr/local/cadence/kits/ams/3.77/spectre/h18/soac/cmos73.scs" "cmostm" )
17 )
18 definitionFile(
19     "/usr/local/cadence/kits/ams/3.77/spectre/h18/soac/mcparams.scs"
20 )
21 ; -----
22 ;
23 ; START EDIT
24 ; -----
25 nop = 256 ; Number of data points
26 step_time = 6u ; conversion time
27
28 read_time = 5.9u ; read out data time 2
29 read_time2 = 3.9u ; read out data time 1
30 lsb = 1n ; LSB
31 temperature = 20.0 ; temperature
32 vsupply = 1.8 ; supply
33
34 ; -----
35 ; STOP EDIT
36 ; -----
37
38 ; Start Simulation
39 sim_time = nop*step_time
40
41 analysis( 'tran ?stop sim_time' )
42 desVar( "vsupply" vsupply )
43 desVar( "tid" step_time )
44 desVar( "bias" lsb )
45 option( 'temp temperature' )
46 )
47 ; option( ?categ 'turboOpts' 'psrFmax 1
48 ; 'psrOption "Default"
49 ; 'numThreads 1
50 ; 'mtOption "Auto"
51 ; 'errorLevel "Do not override"
52 ; 'turboSwitch t
53 ; )
54 saveOption( 'currents "selected" )
55 saveOption( 'pwr "none" )
56 saveOption( 'save "selected" )
57 save( 'i "/DUT/VDDA" "/V13/PLUS" "/DUT/TP50/S" "/VCORELIB/MINUS" )
58 temp( temperature )
59
60 out=outfile("~/SimRes/DAC/data.dat" "w")
61 run()
62
63 ; -----
64 ;
65 ; Print to file
66 ; data.dat - time, value1, value2
67 ; info.txt - info file for auto-detection in matlab
68 ; -----
69 selectResult( 'tran' )
70 fprintf(out "time, valuet1, valuet2\n")
71 for(tt 1 nop
72     time = (step_time*tt) - ((step_time-read_time)*0.000001)
73     time2 = time - (read_time-read_time2)
74     fprintf(out "%5.14f" time)
75     fprintf(out ",")
76     fprintf(out "%5.14f" value(IT("/V13/PLUS"),time))
77     fprintf(out ",")
78     fprintf(out "%5.14f" value(IT("/V13/PLUS"),time2))
79     fprintf(out "\n")
80 )
```

```

81 close(out)
82
83 out=outfile("~/SimRes/DAC/info.txt" "w")
84 fprintf(out "Bias[A], %5.14f\n" lsb)
85 fprintf(out "DC-bias[A], %5.14f\n" value(IT("/DUT/TP50/S"),step_time))
86 fprintf(out "Time[s], %5.14f\n" step_time)
87 fprintf(out "1_Read_out_time[s], %5.14f\n" read_time)
88 fprintf(out "2_Read_out_time[s], %5.14f\n" read_time2)
89 fprintf(out "Temp[ Celcius ], %5.3f\n" temperature)
90 fprintf(out "Analog Power[A], %5.14f\n" average(IT("/DUT/VDDA")))
91 fprintf(out "Digital Power[A], %5.14f\n" average(IT("/VCORELIB/MINUS")))
92 fprintf(out "Total Power[A], %5.14f\n" average(IT("/DUT/VDDA"))+average(IT("/VCORELIB/
MINUS")))
93 fprintf(out "Vsupply[V], %5.2f\n" vsupply)
94 fprintf(out "Module: CCM\n")
95 close(out)

```

Listing B.4: Ocean code for DAC module

# Appendix C

## Matlab Code

### C.1 iDAC

```
1 %% Test iDAC module for INL and DNL
2 % - standard input is:
3 % inl.csv containing [time, output current]
4 % having InA LSB and step size.
5
6 % Select and load data similar to ADC
7 % - uses folder structure to select dac data
8 if get_data(select_data())
9     return; % No file = no point in doing anything:(
10 end
11
12 % Store data to proper variables
13 t = data(:,1);
14 y1 = abs(data(:,2));
15 y2 = abs(data(:,3));
16 y3 = ((y1+y2)/2-iin_correction); % Average over 4us
17 x = (0:2^bit-1)'; % Digital input (DAC)
18 y = y3;
19
20 % Start Computation.
21 % Small check to check if I have control over the input data.
22 % if the check fails, the scripts terminates.
23 if lsb > 1.6*(y(2)) || lsb < 0.7*(y(2))
24     fprintf('test.m-->\n\tWarning\n\tLSB_check_failed\n');
25     fprintf('\t%g->%g->%g, not true.\n', 1.6*(y(2)), lsb, 0.7*(y(2)))
26     fprintf('\tFlipping input\n');
27     y = flipud(y);
28     if lsb > 1.6*(y(2)) || lsb < 0.7*(y(2))
29         fprintf('test.m-->\n\tError\n\tLSB_check_failed\n');
30         fprintf('\t%g->%g->%g, not true.\n', 1.6*(y(2)), lsb, 0.7*(y(2)))
31         return
32     end
33     fprintf('\tInput_OK_(but_corrected)\n');
34 end
35
36 offset = y(1)/lsb; % Caluclate offest in lsb
37 gainerror = (y(length(y))-y(1))/lsb-2^bit-1; % Calculate gainerror in lsb
38
39 % Remove offset from signal
40 y = y-offset*lsb;
41
42 % Remove Gain error from signal
43 y=y.*(2^bit-1)*lsb)/y(end);
44
45 inl = zeros(1,length(y));
46 dnl = zeros(1,length(y)-1);
47
48 % Endpoint INL
```

```

49 endpoint = lsb*x;%-0.5*lsb;
50 inl_ep = (y-endpoint)/lsb;
51
52 % Best fit INL
53 bf_params = polyfit(x,y,1);
54 bestfit = bf_params(1)*x+bf_params(2);
55 inl_bf = (y - bestfit)/lsb;
56
57 %Calculate DNL
58 for i=0:length(y)-2
59     dnl(i+1)=inl_bf(i+2)-inl_bf(i+1);
60 end
61
62 fprintf('\n-----\n')
63 fprintf('DAC Performance\n')
64 fprintf('-----\n')
65 fprintf('INL_bf: \t %5.2f_LSBs \nINL_ep: \t %5.2f_LSBs \nDNL: \t \t %5.2f_LSBs \nGain_
error: \t %5.2f_LSBs \nOffset: \t %5.2f_pA \n', max(abs(inl_bf)),max(abs(inl_ep)), max
(abs(dnl)), gainerror, offset)
66 fprintf('-----\n')
67
68 % Select data for output to run module.
69 inl = inl_bf;
70 codes = 0:length(inl)-1;

```

Listing C.1: Calculate static iDAC performance

## C.2 ADC Core

```

1 %% Test ADC data
2 % - Calculate INL, DNL, gain error and offset
3 % Note: Because the resolution of the captured data is fairly low, the
4 % calculations are somewhat untraditional, but will give an approximation
5 % of the parameters.
6
7 % Select and load data
8 % - provides an interactive interface for the selecting
9 % - Written to accept auto-detection of info-files
10 if get_data(select_data())
11     return; % No file = no point in doing anything:(
12 end
13
14 % store the imported data to proper variables
15 t = data(:,1);
16 y = zeros(1,length(t));
17 for i=1:length(t) % Convert binary adc output to dec.
18     bin_value = zeros(1,bit);
19     for j=1:bit
20         if data(i,j+1) > 0.5
21             bin_value(j) = 1;
22         end
23     end
24     y(i) = (bin2dec(num2str(bin_value, '%i')));
25 end
26 x = abs(data(:,end))'; % Import and convert input current (The last column of data)
27 x = x-stepSize; % AUTO FIX, variable provided in info file if used.
28
29 % Find DNL
30 dnl = zeros(1,max(y)-1);
31 for i=1:max(y)%length(dnl)
32     index_v2 = find(y>=i,1);
33     index_v1 = find(y>=(i-1),1);
34     dnl(i) = ((x(index_v2)-x(index_v1))/lsb)-1;
35 end
36
37 % Calculate gain error
38 lsb_avg = mean(dnl+1);
39 gainerror = 2^bit*(lsb_avg-1);
40
41 % Calculate INL
42 inl = zeros(1,max(y));
43 for i = 1:max(y)+1%length(inl)
44     index_vd = find(y>=(i-1),1);
45     inl(i) = (x(index_vd)/(lsb))-(i-1);

```

```

46 end
47
48 in1 = in1 - mean(in1);
49 codes = 0:length(in1)-1;
50 offset = (x(find(y==1,1))-8e-9-1sb)/1sb;
51
52 fprintf('\n-----\n')
53 fprintf('\nDAC Performance\n')
54 fprintf('\nResolution: 1/%i LSBs\n', (1sb/stepSize))
55 fprintf('-----\n')
56 fprintf('INL: \t\t %5.2f LSBs\nDNL: \t\t %5.2f LSBs\nGain error: \t\t %5.2f LSBs\nOffset: \t\t %5.2f LSBs\n', max(abs(in1)), max(abs(dnl)), gainerror, offset)
57 fprintf('-----\n')

```

Listing C.2: Calculate static ADC Core performance

### C.3 iComp input impedance

```

1 %% Small script of testing input impedance of the iComp module
2 % imports a measurement of the resistance, and compares it
3 % with an ideal resistor
4 clear
5 clc
6 fileToProcess = '../Measurements/v2.0/iComp/2011-06-09/v02/res.csv'
7 import_data(fileToProcess);
8 current = data(:,1);
9 rmeasured = data(:,2);
10 l = length(rmeasured)
11 i = -12.5e-9:0.01e-9:13.1e-9; % Input current interval to look at
12
13 % selecting bin for the measured data and calculates the resistance value.
14 values_pr_bit = ceil(length(i)/256);
15 bin = (floor(length(i)/2+1)-round(values_pr_bit/2)):1:(floor(length(i)/2+1)+round(values_pr_bit/2));
16 i(bin) = i((floor(length(i)/2+1)-round(values_pr_bit/2))-1);
17 u = 0.2;
18 r = abs(u./i); % U=RI
19
20 % print figure
21 figure(1)
22 semilogy(i*1e9, r/1e6)
23 hold on
24 grid on
25 semilogy(i*1e9, rmeasured/1e6, 'r')
26 hold off
27 axis([-128e-1 128e-1 10^1 10^6])
28 xlabel('Input current I_C [nA]')
29 ylabel('Input Resistance [M\Omega]')
30 legend('Estimated', 'Measured')

```

Listing C.3: Compare input impedance of iComp with a resistor



# Appendix D

## Pin overview

Table D.1: ADC pin overview

PIN #	NAME	I/O				DESCRIPTION
		TYPE	DIR	MDOE	DRIVE	
1	EN	D	I	V	NA	Start conversion
2	RESET	D	I	V	NA	External reset
3	SLEEP	D	I	V	NA	Active low, for turning of analogue components. Turn on time $\sim 15ms$
4	MODE	D	I	V	NA	0 - Normal Mode 1 - Test Mode
5	IN	A		I	0-256nA	
	MODE=0		I			, Input current
	MODE=1		O			, DAC Output current
6	IBIAS	A	I	I	1nA	Input bias current
7	VBIAS	A	I	V	0.8V	Input bias voltage
8	EOC	D	O	V	1x	End of conversion
9	DO7	D	O	V	1x	Data out MSB
10	DO6	D	O	V	1x	Data out bit 7
11	DO5	D	O	V	1x	Data out bit 6
12	DO4	D	O	V	1x	Data out bit 5
13	DO3	D	O	V	1x	Data out bit 4
14	DO2	D	O	V	1x	Data out bit 3
15	DO1	D	O	V	1x	Data out bit 2
16	DO0	D	O	V	1x	Data out bit LSB
17	SDI	D	I	V	1x	Serial data line for calibration bits and DAC test input
18	SCK	D	I	V	1x	Clock for SDI
19	VDD	IP	NA	V	NA	Power
20	GND	IP	NA	V	NA	Ground

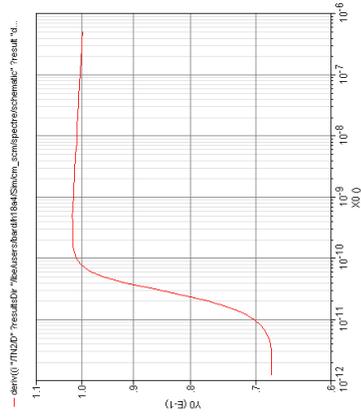
# Appendix E

## Measurements

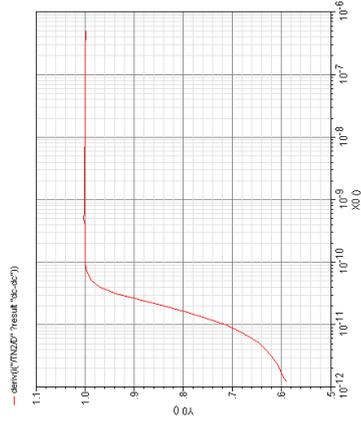
### E.1 Current Mirror



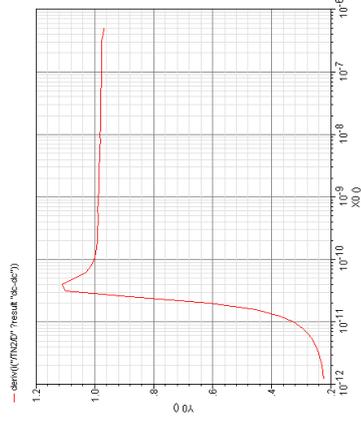
## E.1.1 DR



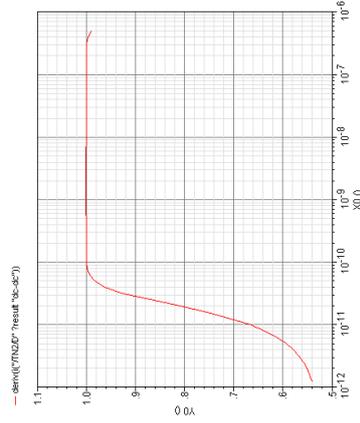
(a) SCM



(b) CCM



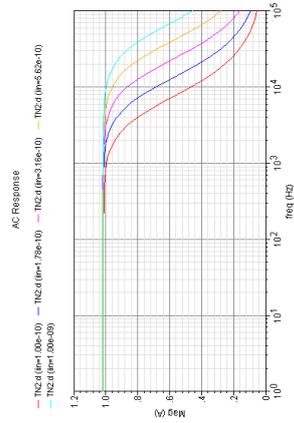
(c) WCM



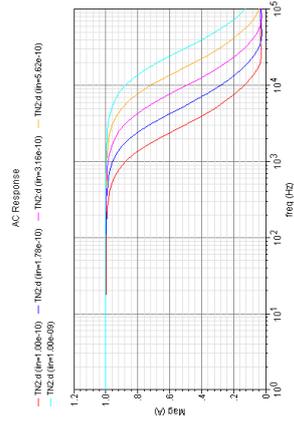
(d) IWCM

Figure E.1: The derivative of DC simulations for CMs at input currents from 1pA to 1μA.

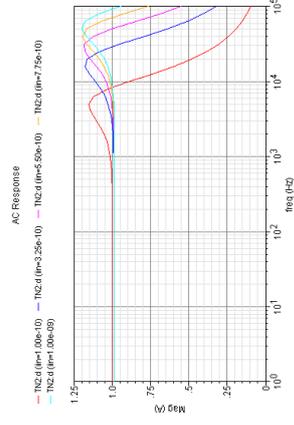
## E.1.2 AC



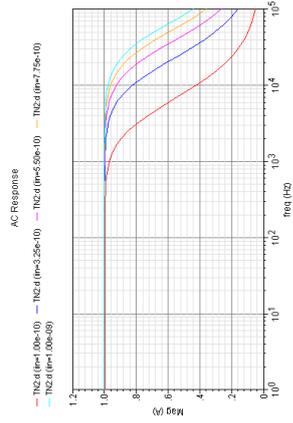
(a) SCM



(b) CCM



(c) WCM



(d) IWCM

Figure E.2: AC simulations of CMCs for currents in range of 0.1nA to 1nA

## E.1.3 Noise

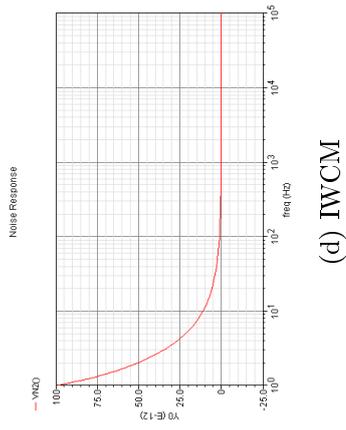
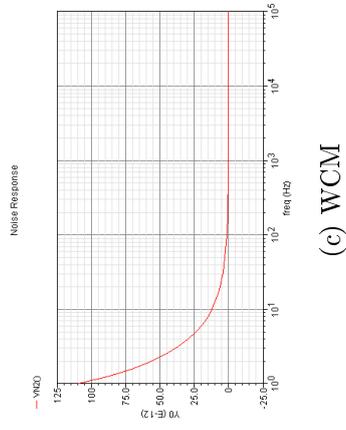
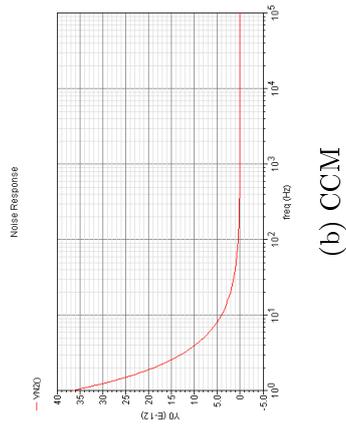
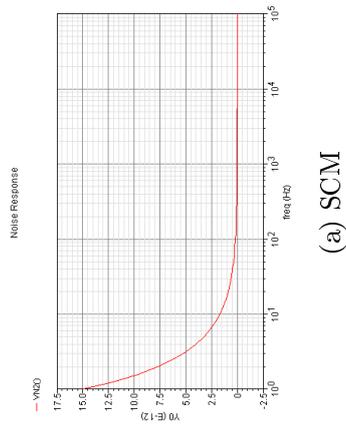


Figure E.3: Noise simulations for CMs

## E.1.4 Leakage

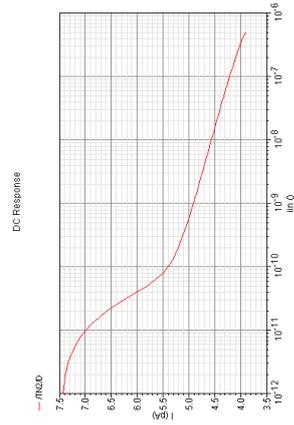
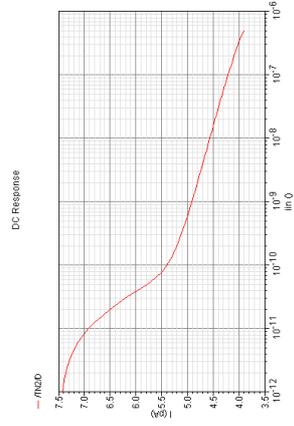
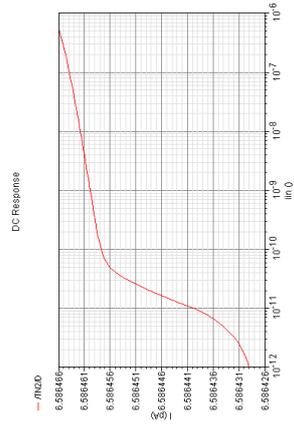
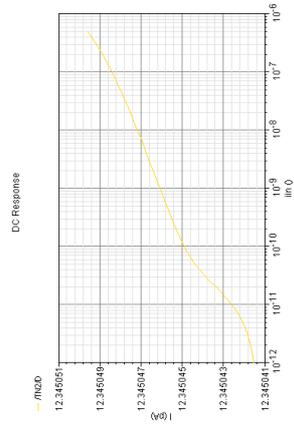


Figure E.4: Leakage simulations of MCs

## E.1.5 MC

L[ $\mu\text{m}$ ]	W/L	SCM				CCM				IWCM				WCM	
		$\mu$ [%]	$\sigma$ [%]	Area[ $\mu^2$ ]	$\mu$ [%]	$\sigma$ [%]	Area[ $\mu^2$ ]	$\mu$ [%]	$\sigma$ [%]	Area[ $\mu^2$ ]	$\mu$ [%]	$\sigma$ [%]	Area[ $\mu^2$ ]	$\mu$ [%]	$\sigma$ [%]
2	0.1	7.85	8.24	4.0	-1.33	8.59	8.0	-1.34	8.59	8.0	-18.99	7.14	6.0		
	0.2	7.39	8.60	3.2	-1.37	8.97	6.4	-1.38	8.97	6.4	-18.81	7.51	4.8		
	0.3	6.91	8.93	2.4	-1.40	9.33	4.8	-1.41	9.33	4.8	-18.37	7.91	3.6		
	0.4	6.48	9.13	1.6	-1.39	9.54	3.2	-1.39	9.54	3.2	-17.31	8.30	2.4		
	0.5	6.52	9.26	0.8	-1.21	9.61	1.6	-1.21	9.61	1.6	-14.99	9.03	1.2		
5	0.1	3.92	3.13	25.0	-0.63	3.34	50.0	-0.63	3.34	50.0	-9.62	2.79	37.5		
	0.2	3.61	3.39	20.0	-0.67	3.62	40.0	-0.67	3.62	40.0	-9.72	3.04	30.0		
	0.3	3.23	3.69	15.0	-0.71	3.94	30.0	-0.71	3.94	30.0	-9.79	3.34	22.5		
	0.4	2.74	4.00	10.0	-0.76	4.28	20.0	-0.76	4.28	20.0	-9.74	3.67	15.0		
	0.5	2.09	4.10	5.0	-0.81	4.39	10.0	-0.82	4.38	10.0	-9.21	3.90	7.5		
10	0.1	2.28	1.16	100.0	-0.34	1.28	200.0	-0.34	1.28	200.0	-5.38	1.02	150.0		
	0.2	2.09	1.32	80.0	-0.36	1.45	160.0	-0.36	1.45	160.0	-5.48	1.18	120.0		
	0.3	1.85	1.52	60.0	-0.39	1.67	120.0	-0.39	1.67	120.0	-5.60	1.37	90.0		
	0.4	1.52	1.77	40.0	-0.43	1.94	80.0	-0.43	1.94	80.0	-5.71	1.62	60.0		
	0.5	1.01	2.00	20.0	-0.53	2.19	40.0	-0.56	2.18	40.0	-5.79	1.88	30.0		

Table E.1: Summary of simulation results of the different topologies at scaling 1:1 at MSB