Calibration of the Nauta Structure Differential OTA

Author:
Iberzanov, Artemij

Publication Date:
2015

DOI:
https://doi.org/10.26190/unsworks/18722

License:
https://creativecommons.org/licenses/by-nc-nd/3.0/au/
Link to license to see what you are allowed to do with this resource.

Downloaded from http://hdl.handle.net/1959.4/55490 in https://unsworks.unsw.edu.au on 2023-09-14
Calibration of the Nauta Structure
Differential OTA

by
Artemij Iberzanov

A thesis submitted in partial fulfilment for the
degree of Master of Engineering

in the
Faculty of Engineering
School of Electrical Engineering and Telecommunications
The University of New South Wales

August 2015

UNSW
AUSTRALIA
Abstract

This work identifies properties and behaviours of a differential operational transconductance amplifier (OTA) consisting of six inverters, first proposed by Bram Nauta, that are exploited to design a digital calibration technique for a modified, digitally-controllable version of the OTA. The calibration technique is designed to reliably tune the OTA for high gain in such a way that very little additional analogue hardware is required, that then could be implemented as an integrated on-chip solution.

Due to its simple topology the Nauta structure is suitable for operation with low supply voltages and can have a high bandwidth, while still exhibiting a high gain. Such properties make the Nauta structure exceptionally well suited to being scaled down and implemented in deep sub-micron technologies. Unfortunately the Nauta structure is also highly susceptible to device mismatch which is a major issue for deep sub-micron technologies and this shortcoming must be addressed to ensure the utility of the Nauta structure. To deal with the issue of mismatch a modified, digitally controllable Nauta structure is used. In this structure, conventional inverters are replaced with digital-to-transconductance (DTC) converters, whose transconductances can be controlled with a digital control code, this in turn allows the Nauta structure to be tuned for high gain and mitigate the effects of mismatch.

In the Nauta structure the gain is enhanced by the presence of positive feedback and excessive positive feedback is known to cause hysteresis. This work correlates the presence of hysteresis, or lack thereof and high gain. In this work a number of calibration approaches exploiting the presence of hysteresis are examined and their feasibility is discussed and investigated. Based on the gathered data, a synthesisable digital calibration algorithm is then presented.
Acknowledgments

I would like to thank my supervisors Tara Julia Hamilton and Torsten Lehmann, and also everyone that is a part of UNSW Microelectronics “Nanolab” group, in particular Andrew Peter Nicholson and Julian Jenkins for their support and advice. Without them I would not have been able to achieve my goals.

I would also like to thank my family and friends for their continuing support and encouragement during my studies at UNSW.
Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADC</td>
<td>Analogue to Digital Converter</td>
</tr>
<tr>
<td>CC</td>
<td>Cross-Coupled</td>
</tr>
<tr>
<td>CMRR</td>
<td>Common Mode Rejection Ratio</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>CRRD</td>
<td>Component Redundancy and Random Diversity</td>
</tr>
<tr>
<td>DMM</td>
<td>Digital Multimeter</td>
</tr>
<tr>
<td>DNL</td>
<td>Differential Nonlinearity</td>
</tr>
<tr>
<td>DTC</td>
<td>Digital to Transconductance Converter</td>
</tr>
<tr>
<td>FF</td>
<td>Feed-Forward</td>
</tr>
<tr>
<td>GPU</td>
<td>Graphics Processing Unit</td>
</tr>
<tr>
<td>ICMR</td>
<td>Input Common Mode Range</td>
</tr>
<tr>
<td>LSB</td>
<td>Least Significant Bit</td>
</tr>
<tr>
<td>MSB</td>
<td>Most Significant Bit</td>
</tr>
<tr>
<td>NAND</td>
<td>Negative (Not) And</td>
</tr>
<tr>
<td>NMOS</td>
<td>N-Type Metal Oxide Semiconductor</td>
</tr>
<tr>
<td>OTA</td>
<td>Operational Transconductance Amplifier</td>
</tr>
<tr>
<td>PMOS</td>
<td>P-Type Metal Oxide Semiconductor</td>
</tr>
<tr>
<td>PVT</td>
<td>Process Voltage Temperature</td>
</tr>
<tr>
<td>SC</td>
<td>Self-Coupled</td>
</tr>
<tr>
<td>SMU</td>
<td>Source Measure Unit</td>
</tr>
<tr>
<td>SNDR</td>
<td>Signal to Noise and Distortion Ratio</td>
</tr>
<tr>
<td>SRAM</td>
<td>Static Random Access Memory</td>
</tr>
<tr>
<td>TSMC</td>
<td>Taiwan Semiconductor Manufacturing Company</td>
</tr>
<tr>
<td>UGF</td>
<td>Unity Gain Frequency</td>
</tr>
<tr>
<td>VISA</td>
<td>Virtual Instrument Software Architecture</td>
</tr>
</tbody>
</table>
# Table of Contents

Abstract ......................................................................................................................... 1  
Acknowledgments ........................................................................................................ 2  
Abbreviations ................................................................................................................ 3  
Table of Contents ......................................................................................................... 4  
Table of Figures ............................................................................................................ 5  
Table of Tables ............................................................................................................. 8  
1. Introduction ............................................................................................................. 9  
   1.1 Motivation ........................................................................................................... 9  
   1.2 Aim .................................................................................................................... 10  
   1.3 Thesis Structure ................................................................................................ 10  
   1.4 Research Output ............................................................................................... 11  
2. Background ............................................................................................................ 12  
   2.1 Introduction ....................................................................................................... 12  
   2.2 Process variation ................................................................................................ 12  
      2.2.1 Introduction ................................................................................................. 12  
      2.2.2 Classification .............................................................................................. 12  
      2.2.3 Trends ........................................................................................................ 14  
      2.2.4 Conclusion ................................................................................................ 15  
   2.3 Digitally Assisted Circuits .................................................................................. 16  
      2.3.1 Introduction ................................................................................................. 16  
      2.3.2 Analogue To Digital ................................................................................... 16  
      2.3.3 Conclusion ................................................................................................ 19  
   2.4 Negative Conductance Gain Enhancement ....................................................... 20  
      2.4.1 Introduction ................................................................................................. 20  
      2.4.2 Performance and Downsides ...................................................................... 21  
      2.4.3 Conclusion ................................................................................................ 26  
   2.5 Nauta Structure .................................................................................................. 27  
      2.5.1 Introduction ................................................................................................. 27  
      2.5.2 Modifications and Applications ................................................................. 30  
      2.5.3 Conclusion ................................................................................................ 39  
   2.6 Digital to Transconductance Converter Based Nauta Structure ....................... 40  
      2.6.1 180nm Prototype ....................................................................................... 41  
      2.6.2 65nm Prototype ......................................................................................... 42  
3. Theory and Solutions ............................................................................................ 45  
   3.1 Introduction ........................................................................................................ 45  
   3.2 Requirements and Calibration Approaches ...................................................... 45  
   3.3 Detection of Hysteresis ...................................................................................... 48  
   3.4 Proposed Calibration Approaches ..................................................................... 50
Table of Figures

Figure 1 - Types of Variation in the Performance of Integrated Circuits [4] ........................................ 12
Figure 2 - a) Traditional and b) Digitally-Assisted Digital to Analogue Interface [9] .................. 16
Figure 3 - Power Consumption of ADCs and NAND Gates for Various Technologies [9] ......... 17
Figure 4 - Miller Compensation Block Diagram ............................................................ 20
Figure 5 - Conventional Telescoping Cascode Structure ....................................................... 20
Figure 6 - Basic Principles of Negative Conductance [15] .................................................... 21
Figure 7 - Negative Conductance Controlled by Output Conductance [15] ................................. 22
Figure 8 - Half Circuit of the Amplifier (Representation of One Input Path of a Differential Amplifier) [15] ...................................................... 23
Figure 9 - DC Gain, $A_v(s)$ Versus the Control Voltage ($V_{ctrl}$) [15] ................................. 23
Figure 53 - a) $g_m$ Cancellation Concept b) Design Example [17] ............................ 24
Figure 54 - Schematic of the Topology-[18] Implementation ..................................... 26
Figure 55 - Nauta Structure [19].......................................................................... 27
Figure 56 - Comparison of Band Pass Gain for 3 Different Designs (Where CC Inverter PMOS Widths Are 8.8µm, 8.9µm and 9.0µm) [20] © 2002 IEEE ....................... 29
Figure 57 - Various Approaches to Varying FF Inverter Dimensions [21] .................. 30
Figure 58 - Nauta Structure Topology-[22] ............................................................. 31
Figure 59 - Double CMOS Pair [23] ..................................................................... 33
Figure 60 - Tuneable Topology-[23] Nauta Structure .............................................. 33
Figure 61 - a) Topology-[24] Nauta Structure and b) Gyrorator Implementation [24] ... 34
Figure 62 - Positive Feedback Amplifier with CRRD [26] © 2012 IEEE ................. 35
Figure 63 - Contrast of Array Sizing [26] © 2012 IEEE........................................... 36
Figure 64 - Increase in Resolution Due to Increase in Array Size [26] © 2012 IEEE .... 37
Figure 65 - Performance of the Fabricated Circuit [26] © 2012 IEEE...................... 37
Figure 66 - Complementary Constant-$g_m$ Voltage Bias Circuit with Integrated Regulator [27] .......................................................................................... 38
Figure 67 - Percentage Deviation Over Temperature [27] ......................................... 38
Figure 68 - Tri-State Inverter [28] ......................................................................... 40
Figure 69 - Control Bits of the 180nm Prototype ...................................................... 41
Figure 70 - Simulated Op-Amp DC Gain versus Self-Coupled Code-Word, for Steps in Cross-Coupled (CC) Inverter Code-Word [28] © 2014 IEEE .................. 44
Figure 71 - Schmitt Trigger .................................................................................... 48
Figure 72 - Examples of Differential Input Sweep Results, Each of the Outputs Plotted Individually. A) No Hysteresis, No Offset B) No Hysteresis, With Offset C) Hysteresis, No Offset .......................................................................................... 49
Figure 73 - Waveform Distortion Due to Hysteresis .................................................. 50
Figure 74 - Output vs. Reference Hysteresis Detection ............................................. 51
Figure 75 - Output Reference Voltage Boundaries .................................................... 52
Figure 76 - Comparator Based Testing Setup ............................................................ 52
Figure 77 - Resistor Ladder ................................................................................... 53
Figure 78 - Naming Convention of the Nauta Structure ........................................... 56
Figure 79 - Enabled Inverters for Characterisation Sweeps ....................................... 57
Figure 80 - SC Characterization Sweep Measurements .............................................. 59
Figure 81 - FF Characterization Sweep Measurements ($V_{IN}=830mV$) .................. 60
Figure 82 - FF Characterization Sweep Measurements (Supplementary Die, $V_{IN}=830mV$) 60
Figure 83 - CC Characterization Sweep Measurements ($V_{IN} = 830mV$, as set by CC1 at code 63) ................................................................. 61
Figure 84 - SCCC Characterization Test Measurements (0 Offset) ............................ 62
Figure 85 - SCCC Characterization Test Measurements (0 Offset, Supplementary Die) .... 62
Figure 86 - SCCC Characterization Test Measurements (+1 Offset) .......................... 63
Figure 87 - SCCC Characterization Test Measurements (+2 Offset, Bit 2 Disabled) .... 64
Figure 88 - Simulation - Effects of Imbalance and Common Mode Gain on the Output DC Levels, $\Delta W_{CC}$ = PMOS-1µm/ NMOS-500nm ($\approx$3.3% of the Total Width) ........................................ 66
Figure 89 - Gain Tuning Measurements ($V_{INDC}$ = 830mV, $V_{INPP}$ = 1.12mV, $f$=10kHz) 67
Figure 90 - Simulation - Common Mode Sweeps - P/N Imbalance ......................... 69
Figure 91 - Simulation - Common Mode Sweeps - FF Imbalance .............................. 71
Figure 92 - Simulation - Increase in CC (Large Steps, Large Mismatch) ................. 72
Figure 93 - Simulation - Increase in CC (Small Steps, Small Mismatch) .................... 73
Figure 94 - Simulation - Hysteresis During Differential Sweep (CC Codes, No Mismatch) ... 75
Figure 95 - Simulation - Hysteresis During Differential Sweep (FF Codes, No Mismatch) ... 76
Figure 96 - Simulation - Differential Sweeps with PMOS/NMOS Imbalance (Increase NMOS, Top Half Only) .......................................................................... 78
Table of Tables

Table 1 - Parameters and Process Variation [5] ................................................................. 13
Table 2 - Process Variation Error for Different Technologies [1] ......................................... 13
Table 3 - Variation in Access Time [8] .................................................................................. 15
Table 4 - Power Efficiency of ADCs and Equivalent Number of Gates [10] ....................... 17
Table 5 - Performance of a Digitally Assisted Telescopic Amplifier [11] ............................ 18
Table 6 - Performance Comparison of Conventional and Proposed Amplifiers in Nominal Case (Typical-Typical Corner and T = 27°C) [17] .......................................................... 25
Table 7 - Performance Comparison of Conventional Amplifier and the Ones Examined in [17] and [18] ................................................................................................................ 25
Table 8 - Output Resistances of Nauta Structure [19] .......................................................... 28
Table 9 - Performance of the Modified Structure Compared to the Original Nauta Structure [22] ......................................................................................................................... 32
Table 10 - Performance of the Topology [25] Nauta Structure with Double CMOS Pair Inverters .......................................................... 35
Table 11 - Inverter Unit Sizes for 180nm Prototype .............................................................. 42
Table 12 - Inverter Sizing of 65nm Prototype .................................................................... 43
Table 13 - Mean and Standard Deviation of Transconductance Difference at Worst Case Transition Under Monte-Carlo Mismatch Analysis [28] ........................................ 43
Table 14 - Code Searching Approaches ................................................................................ 46
Table 15 - Gain Measurement Approaches ......................................................................... 47
Table 16 - Comparison of Tuning Approaches .................................................................. 55
Table 17 - Change in Output Voltage Due to Mismatch Between FF NMOS Transistors (FF1 & FF2) Width at Various Offsets ................................................................. 70
Table 18 - Change in Output Voltage Due to Mismatch Between FF Inverters (FF1 & FF2) Width at Various Offsets .................................................................................. 70
Table 19 - Change in Output Voltage Due to Increase in CC Inverter Width at Vin=876mV and a Constant CC1/CC2 Mismatch of NMOS=25nm/PMOS=50nm ...................... 72
Table 20 - Change in Output Voltage Due to Increase in CC Inverter Width at Vin=876mV and a Constant CC1/CC2 Mismatch of NMOS=5nm/PMOS=10nm ....................... 73
Table 21 - High Gain Settings ............................................................................................. 100
Table 22 - Tuning Algorithm Stage I .................................................................................... 103
Table 23 - Tuning Algorithm Stage II .................................................................................. 104
Table 24 - Tuning Example ................................................................................................. 105
Table 25 - Change in DC Gain Due to Decrease in FF Width (AC Simulation) and Change in Hysteresis Due to Increase in FF Width ......................................................... 107
Table 26 - Performance of Bandgap Voltage references .................................................... 109
Table 27 - Initial and Post Calibration Codes ..................................................................... 114
Table 28 - Calibration Results ............................................................................................ 114
1. Introduction

1.1 Motivation

Electronic technology is improving, seemingly at a continually increasing rate. This is especially evident in consumer electronic markets, as devices such as smart phones and tablets are becoming smaller and more powerful every year. This puts developers under enormous pressure to design and optimize integrated circuits to increase processing power while reducing power consumption.

Operational amplifiers are commonly found in critically important circuits such as digital to analogue or analogue to digital converters. As semiconductor fabrication technologies improve transistor sizes also decrease. These transistors become faster but must operate with lower power supply voltages. The circuits can now be designed to be more compact and consequently the devices using these circuits can also be reduced in size. Unfortunately the unwanted side effect of the transistor size reduction is that mismatch errors have now become a significant issue, especially in deep sub-micron sized transistors. The mismatch errors are caused by process variation during fabrication of the integrated circuits and are often impossible to predict and very hard to account for during the design stage. Mismatch errors cause variation in important transistor design parameters such as threshold voltage and channel length (see Ch. 2.2). As the transistors become smaller the errors due to process variation become a more significant proportion of the actual size of the transistor [1]. Transistor sizing is often critical to achieve expected performance levels and can even fundamentally alter the way the circuit operates.

Process variation was studied and well understood for older technologies. However for newer deep sub-micron technologies, process variation is becoming increasingly difficult to understand and characterise. Certain types of variation are systematic and layout dependent and are a lot easier to predict [2]. Other types of variations, at this point appear to be random and extremely hard to predict. Also attempts are constantly being made to identify the properties of transistors that are affected the most. This allows the circuit designers to develop circuits in such way that they do not rely on the most sensitive properties of transistors [3] or in ways that employ additional techniques that compensate for mismatch errors. This can lead to circuits being more complex, consuming more power or having a suboptimal performance.
Furthermore, traditional operational amplifier structures do not perform as well when scaled down for fabrication with deep sub-micron technologies as the intrinsic gain of transistors is reduced. Gain enhancement techniques, such as cascading of amplifier stages and cascoding of multiple transistors are being used. Unfortunately, using such techniques causes increased power consumption, requires more area or lowers the output voltage swing, and requires higher power supply voltages [3].

This work is primarily motivated by the need to overcome the challenges of working with scaled down, deep sub-micron technologies described above. Due to its simplicity the Nauta structure is more suited to operation with lower power supply voltages while having relatively high output voltage swing, unlike more traditional structures. Additionally, due to the lack of internal nodes and therefore parasitic poles and zeros, the structure has the potential to have a large bandwidth. As a result of having such properties, the Nauta structure is ideally suited for scaled down processes and is expected to perform equally well when implemented using the latest technologies. However, the main concern associated with scaling down the Nauta structure is its sensitivity to mismatch caused by process variation and this will be addressed in this thesis.

1.2 Aim

The aim of this work is to address the issue of mismatch that causes sub-optimal performance in the Nauta structure by developing a calibration procedure that will allow the structure to be tuned for high gain. The calibration procedure must be designed in such a way that allows it to be implemented on-chip, requires minimal amount of area and has low power consumption.

1.3 Thesis Structure

The following sections of this report will cover:

**Chapter 2**: Detailed background and literature review.
**Chapter 3**: Proposed design.
**Chapter 4**: Testing of the Nauta structure and proposed calibration procedures.
**Chapter 5**: HDL implementation of the calibration procedure.
**Chapter 6**: Conclusion and proposed future work.

**Appendix A**: Brief overview of the laboratory equipment used for testing.
**Appendix B**: Source code of the calibration algorithm and test benches.
1.4 Research Output


2. Background

2.1 Introduction

This section contains a literature review of the topics relevant to understanding the topic of the Nauta structure calibration. Firstly, types and effects of process variation are examined, as mismatch due to process variation is the primary impediment preventing the Nauta structure from operating with high gain reliably. Secondly, the use of digitally-assisted circuits and the possible benefits of using digital based solutions for improving the performance of the Nauta structure are examined. Thirdly, the negative conductance gain enhancement techniques are examined, as the same principle is exploited in the Nauta structure, in order to gain a better understanding of the potential upsides and downsides of this approach. Finally the Nauta structure itself and previously proposed solutions to achieving high gain with the Nauta structure will be examined in greater detail.

2.2 Process variation

2.2.1 Introduction

Process variation during fabrication causes mismatch errors, where the physical dimensions of transistors in a manufactured integrated circuit do not match the ideal dimensions specified during the design. In this section this problem and its trends will be examined in more detail to provide better understanding of how it complicates the task of achieving high gain with the Nauta structure.

![Variations Diagram](image)

Figure 1 - Types of Variation in the Performance of Integrated Circuits [4]

2.2.2 Classification

Mismatch errors can be introduced at various stages during manufacturing, but in general can be classified as random, systematic, inter-die and intra-die as seen in Figure 1 and Table 1[5]. Additionally, when classifying types of process variation it is common to consider the specific source of the variation such as lithography, doping and etching. It is also important to note the affected design parameters such as gate length, width and threshold voltage [5].
Systematic errors are a lot easier to predict, and for example might depend on the position of the circuit on the die. On the other hand, errors are classified as random if there is no apparent pattern that would allow such errors to be estimated. Intra-die errors refer to mismatch between the same components on the same die, inter-die errors refer to mismatch between components on different dies. The mismatch errors of a certain component property may be caused by different types of errors at the same time. This can be seen in Table 1, demonstrating the different forms of variation affecting some of the properties of a commercial 90nm process.

<table>
<thead>
<tr>
<th>Component</th>
<th>Form of Variation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel length</td>
<td>Inter-die systematic</td>
</tr>
<tr>
<td>Mean Threshold voltage</td>
<td>Intra-die systematic</td>
</tr>
<tr>
<td>Difference between device types</td>
<td>Intra-die random</td>
</tr>
<tr>
<td>Threshold voltage</td>
<td>Inter-die systematic</td>
</tr>
<tr>
<td>Mean metal R and C differences</td>
<td>Inter-die systematic</td>
</tr>
<tr>
<td>Between metal levels</td>
<td>Intra-die random</td>
</tr>
<tr>
<td>Voltage and Temperature</td>
<td>Intra-die systematic</td>
</tr>
<tr>
<td>NBTI, hot-e</td>
<td>Intra-die systematic</td>
</tr>
</tbody>
</table>

Table 1 - Parameters and Process Variation [5]

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Nominal Values</th>
<th>3σ Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>L_{eff} (nm)</td>
<td>250</td>
<td>180</td>
</tr>
<tr>
<td>T_{ox} (nm)</td>
<td>5</td>
<td>4.5</td>
</tr>
<tr>
<td>V_{dd} (V)</td>
<td>2.5</td>
<td>1.8</td>
</tr>
<tr>
<td>V_{th} (mV)</td>
<td>500</td>
<td>450</td>
</tr>
<tr>
<td>W (µm)</td>
<td>0.8</td>
<td>0.55</td>
</tr>
<tr>
<td>H (µm)</td>
<td>1.2</td>
<td>1.0</td>
</tr>
<tr>
<td>ρ (mΩ)</td>
<td>45</td>
<td>50</td>
</tr>
</tbody>
</table>

Table 2 - Process Variation Error for Different Technologies [1]
2.2.3 Trends

The major issue with process variation is that as the technology scales down, the errors caused by process variations do not scale down equally well. This can be seen in Table 2, where ideal values are compared to the error values. In Table 2, $L_{\text{eff}}$ – effective gate length, $T_{\text{ox}}$ – gate oxide thickness, $V_{dd}$ – power supply voltage, $V_{th}$ – threshold voltage, $W$ – width, $H$ – distance to the ground plane and $\rho$ – resistivity where examined. For simulation purposes $V_{dd}$ was assumed to have a tolerance of ±10% and temperature was set to vary between 25°C and 125°C. 100 simulations were conducted for each technology using Latin Hypercube sampling [6]. The errors are given as a proportion of the ideal parameter increase. This highlights the necessity for more robust circuits, or improving and enhancing existing circuits, such as the Nauta structure which is the focus of this research.

Alternatively, it is possible to attempt to lessen the effects of the process variation by using improved device models that try to account for it. Obtaining such models involves characterising a large number of devices. The initial step in characterising process variation is the design of suitable test structure [7]. A test structure must contain a sufficiently large number of individual components to be able to produce results that are statistically significant. For a test structure to produce the most reliable results it must be designed in such a way that allows each component to be accessed individually. If components are accessed as a part of an array of a number of components in series, the errors in variations may be averaged and produce a misleading result [2]. Additional issues worth investigating while designing a test structure may include identifying whether the results will be affected by the location of the component in the test structure array. For example it may be important to consider the voltage drop in the wires connected to the component if the wire length varies for different components. One way of dealing with this can be achieved by making sure it is possible to measure the voltage levels at multiple points in a path and adjust the measurement values if necessary [2]. Another problem may be due to leakage through non selected devices, and a possible way to combat this may be by clamping the terminals to specific voltages [2]. Furthermore it is important to consider designing a test structure that would allow the data to be gathered quickly and efficiently. Taking into account that the test structure may contain thousands if not tens of thousands of individual transistors that need to be tested, the time it takes to analyse each one can be crucial to quickly and reliably obtaining the required information [2].

One of the Nauta structure prototypes examined in this thesis was fabricated in 65nm technology. The process variation in 65nm technology was also examined in [8] by investigating the performance of SRAM cells under
process variation. Not unlike amplifier structures, the increase in process variation within newer technologies can also have a significant effect on memory circuits such as SRAM. Changes in threshold voltage and channel length can have an impact on the performance of such circuits and result in reduced operating frequency and increased number of faulty memory cells. Table 3 shows the worst case variation in access time for registers of several different sizes based on 100 run Monte-Carlo simulation. It can be seen that process variation causes a significant deviation from the expected circuit performance levels. Similar deviation in a fixed width, non-tuneable Nauta structure would almost certainly cause a significant loss of gain.

<table>
<thead>
<tr>
<th>S. No</th>
<th>Register File Size</th>
<th>Access Time Without PV</th>
<th>Access Time Under PV</th>
<th>Variation in Access Time</th>
<th>Read/Write Power</th>
<th>Leakage Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16 bit</td>
<td>71ps</td>
<td>96ps</td>
<td>35%</td>
<td>756µW</td>
<td>427µW</td>
</tr>
<tr>
<td>2</td>
<td>32 bit</td>
<td>73ps</td>
<td>100ps</td>
<td>37%</td>
<td>1.68mW</td>
<td>0.72mW</td>
</tr>
<tr>
<td>3</td>
<td>64 bit</td>
<td>75ps</td>
<td>119ps</td>
<td>59%</td>
<td>3.19mW</td>
<td>1.4mW</td>
</tr>
</tbody>
</table>

Table 3 - Variation in Access Time [8]

2.2.4 Conclusion

Overall it can be seen that with current trends where device sizes are constantly decreasing, mismatch caused by process variation is expected to make the design of analogue circuits an increasingly difficult task. Modelling of process variation is not a straightforward procedure and does not guarantee good results as there are many issues that need to be considered and overcome. A procedure for modelling the performance of transistors would need to be repeated for all new technologies, and even though it is likely that modelling will be conducted for all established processes by major fabricators, it is not guaranteed that it will be possible to use this information to design traditional circuits and expect high performance. Therefore it may be more feasible to focus on the development of more robust and adaptable circuits which can perform well regardless of what technology they are implemented in, and the Nauta structure would fit that description perfectly if a calibration procedure can be developed successfully.
2.3 Digitally Assisted Circuits

2.3.1 Introduction

As technologies are scaled down, usually the primary objective is to increase transistor density and reduce transistor switching times. The power supply voltages also have to be reduced so that the smaller transistors are not destroyed during operation, this also has a positive effect of decreasing the power consumption. Such changes are ideal for digital circuits and especially important for increasing processing power while decreasing the size of various processing units (CPUs, GPUs etc.). While improvements are seen in digital circuits, it is becoming increasingly harder to implement analogue circuits. A possible solution to the problem is to avoid the use of complex analogue circuits, and use low cost digital circuits to augment analogue circuits to achieve higher performance. This section will look at the effectiveness of such circuits and how that has contributed to the design of the tuneable Nauta structure.

2.3.2 Analogue To Digital

While digital circuits are what is primarily needed to process large amounts of data in a day to day operation of various computing devices it is still an inescapable reality that interfaces for converting analogue signals to digital are still required. Figure 2 shows an alternative approach to digital to analogue conversion where instead of solely relying on analogue circuits to do the conversion, simpler and less precise (highlighted in red/horizontal pattern) analogue designs are used along with additional digital post-processing circuits (highlighted in green/vertical pattern) to achieve comparable results [9].

![Figure 2 - a) Traditional and b) Digitally-Assisted Digital to Analogue Interface [9]](image-url)
There are a number of different ways digital circuits can be used to affect the performance of analogue circuits or provide alternative approaches for performing functions that traditionally were performed by the analogue circuits only. For example, as discussed previously and shown in Figure 2 this can be done through digital post processing of the signal to correct the errors of a less precise analogue circuit (such as the ADC). Alternatively digital control signals can be used to disable or enable certain portions of analogue circuits to save power or change its behaviour (such as the bandwidth of the filter). The advantages of choosing to use digitally assisted circuits are also evident when power consumption is considered. For example, power consumption of an ADC can be compared to the number of digital NAND gates that use equivalent power. It can be seen that as technologies are scaled down the number of gates that will cause the same levels of power consumption for any given ADC will increase exponentially. Figure 3 also

![Power Consumption of ADCs and NAND Gates for Various Technologies](image)

Figure 3 - Power Consumption of ADCs and NAND Gates for Various Technologies [9]

<table>
<thead>
<tr>
<th>SNDR (dB)</th>
<th>$E_{\text{ADC}}$</th>
<th>$E_{\text{ADC}}/E_{\text{NAND}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>21nJ</td>
<td>4,700</td>
</tr>
<tr>
<td>50</td>
<td>168nJ</td>
<td>38,000</td>
</tr>
<tr>
<td>70</td>
<td>1.35µJ</td>
<td>300,000</td>
</tr>
<tr>
<td>90</td>
<td>10.8µJ</td>
<td>2,400,000</td>
</tr>
</tbody>
</table>

Table 4 - Power Efficiency of ADCs and Equivalent Number of Gates [10]
demonstrates that this is especially evident in higher precision ADCs, where ADCs with 10 bit or higher precision consume at least as much power as 100,000 gates for sub 100nm technologies [9].

Similarly in Table 4 it can also be seen how power consumption of an ADC implemented in 90nm technology compares to the power consumption of digital NAND gates and how the equivalent number of gates changes with an increase in signal to noise and distortion ratio (SNDR) [10].

As discussed in the process variation section, the analogue circuits implemented in scaled down technologies suffer from mismatch issues. However, there are other negative effects that need to be considered during the design stages of ADCs. Such effects include thermal noise and the need for linear amplifiers that are becoming harder to design using newer, scaled down technologies [9]. Reducing the effects of high noise can be done by increasing the signal power compared to the noise floor. Increase in signal power comes at the cost of increased transconductance and higher power consumption. It is possible to overcome issues caused by non-linearity of an amplifier simply by increasing the gain. While the issue of noise is a fundamental one, the non-linearity is not. Ideally the same principle of using digital circuits to compensate for any shortcomings would be applied to the gain of the amplifiers to deal with the non-linearity. Nauta structure is a perfect candidate for such design due to its simple six-inverter structure that is expected to be able to achieve a high gain in scaled down technologies, as long as a digital calibration technique can be used to overcome any performance loss caused by process variation.

In addition to using a high gain amplifier there are a number of approaches where digital circuits can be exploited. Such approaches include digital linearization, digital correction of dynamic errors and system synergetic approaches [10]. When thinking of digital correction methods it is easy to

<table>
<thead>
<tr>
<th>Performance Parameters</th>
<th>Low Power Disabled</th>
<th>Low Power Enabled</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICMR-</td>
<td>0.56V</td>
<td>0.56V</td>
</tr>
<tr>
<td>ICMR+</td>
<td>1.08V</td>
<td>1.08V</td>
</tr>
<tr>
<td>ICMR</td>
<td>0.52V</td>
<td>0.52V</td>
</tr>
<tr>
<td>DC Gain</td>
<td>63.5dB</td>
<td>62.2dB</td>
</tr>
<tr>
<td>UGF</td>
<td>261.7MHz</td>
<td>120.4 MHz</td>
</tr>
<tr>
<td>DC Power</td>
<td>245µW</td>
<td>145µW</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>182.5kHz</td>
<td>94kHz</td>
</tr>
<tr>
<td>Phase Margin</td>
<td>61.8°</td>
<td>69.6°</td>
</tr>
<tr>
<td>Slew Rate</td>
<td>16.5 MV/µs</td>
<td>18.1 MV/µs</td>
</tr>
</tbody>
</table>

Table 5 - Performance of a Digitally Assisted Telescopic Amplifier [11]
falsely assume that these methods will not be complex enough to utilise a large number of gates whose power consumption can exceed the power consumption of solutions using analogue circuits. However, these approaches vary in complexity and in some cases may not be feasible for technologies with larger feature sizes. Furthermore the high complexity of the algorithms, especially those that employ statistical analysis may be too slow to dynamically adjust to changing conditions. Even simpler approaches are likely to result in some sort of performance degradation which needs to be considered when the viability of the technique is examined. For example in Table 5 the performance of a digitally assisted amplifier design is presented, showing performance parameters such as input common mode range (ICMR), unity gain frequency (UGF), DC gain and bandwidth. The architecture of the amplifier allows a certain portion of it to be disabled or enabled when needed by a digital control signal to conserve power. Table 5 compares the performance of the amplifier in and out of the low power mode. The most noticeable performance changes can be seen in bandwidth and power consumption parameters. While the power consumption is decreased significantly (from 245µW to 145µW) the UGF is also decreased from 261.7MHz to 120.4 MHz [11].

2.3.3 Conclusion

In conclusion it can be said that digital circuits can be used to address certain shortcomings of the analogue circuits. The benefits can range from lower power consumption to higher precision in circuits such as ADCs. However care must be taken to ensure that the benefits are significant enough to outweigh the likely performance losses in other performance parameters. Such consideration also must be taken when developing a calibration technique for a digitally assisted, modified Nauta structure. As a result a simple calibration technique for the Nauta structure is preferable, so that implementation would not require the use of additional complex circuits that would increase the power consumption or slow down the operation of the circuits using the Nauta structure.
2.4 Negative Conductance Gain Enhancement

2.4.1 Introduction

Increasing the gain of operational amplifiers has always been of great importance to developing new and improved circuits due to their widespread use in a variety of different applications. Traditional techniques such as cascading of multiple amplifier stages or cascode-based approaches such as the regulated cascode [12] approach are becoming less and less appealing.

Various optimization techniques exist for the more traditional structures [13], however overall they still continue to suffer from the same fundamental shortcomings especially in low supply voltage environments. One example is a two stage Miller compensated amplifiers as seen in Figure 4. Such amplifiers can achieve high gain and high voltage swing by cascading multiple stages. However, by cascading two stages the phase margin is reduced and therefore this approach requires an additional capacitor \( C_M \) to compensate for the reduction of the phase margin and improve the phase margin. Improvement in phase margin is achieved by pole splitting. Pole splitting moves the pole with the lowest frequency to an even lower frequency and a pole with a higher frequency to an even higher frequency. When the high frequency pole is pushed beyond the unity gain frequency the phase margin and stability is improved. Unfortunately this approach also reduces the amplifier bandwidth and high frequency gain [14]. Another example is a differential telescopic cascode amplifier as seen in Figure 5. Telescoping cascode amplifier consists of
differential input pair transistors (M1 - M2), cascode transistors (M3 - M4) and current source load transistors (M5 - M8). Having a relatively large number of cascoded transistors at the outputs increases the output resistance and therefore the gain, however it also reduces the output voltage swing. A differential folded cascode amplifier improves the telescopic amplifier structure by reducing the number of transistors at the output and hence increases the output voltage swing. However, it requires an auxiliary amplifier stage to maintain high gain, which increase the overall power consumption [14].

Overall, it can be said that the low appeal of traditional structures is due to difficulties in integrating these structures in low power dissipation, low power supply, low area and high speed applications. Another gain enhancement technique exists that is based on negative conductance principles and has a potential to have the highest gain, lowest power dissipation, excellent high frequency performance while being able to operate with low voltage power supplies.

### 2.4.2 Performance and Downsides

The main disadvantage of negative conductance techniques is a consequence of the fact that they are extremely sensitive to the transconductance of the element from which the negative conductance is derived [3]. The problem becomes more apparent if you take the issue of process variation into account. Even though the negative conductance techniques in theory can perform extremely well, on the other hand when implemented the variations in gain can fluctuate wildly due to process variation. As a result the negative conductance techniques have not been used widely in commercial applications.

Figure 6 - Basic Principles of Negative Conductance [15]

\[ A_V(s) = \frac{v_o}{v_i} = \frac{-g_m}{sC_L + g_ds + \frac{1}{R_n}} \] (1)
Consider a schematic of a basic amplifier (a) and a small signal equivalent circuit (b) in Figure 6. A resistor $R_n$ is placed in parallel with the output conductance. The gain, $A_V(S)$, of the amplifier can be found by dividing the transconductance of the transistor, $g_{m}$, by the overall output conductance, which comprises the conductance of the load capacitor, $sC_L$, the conductance of $R_n$, $1/R_n$, and the output conductance of the transistor, $g_{ds}$, as seen in Eq. 1. When the total output conductance becomes zero, the gain of the amplifier will become infinite. For this to occur the condition of $g_{ds}+1/R_n = 0$ must be satisfied. This can only occur if $R_n$ has a negative value, implying that $R_n$ must be a theoretical “negative” resistor. In reality, no such thing as a negative resistor as a discrete component exists.

Effects of the negative resistance are usually achieved by circuits with an active power source that have equivalent properties where the power is being supplied rather than dissipated like it is in conventional resistors, even though there are some exceptions such as Tunnel and Gunn [16] diodes that possess negative differential resistance characteristics while only dissipating power. Commonly the transconductance of an element is used to determine and generate the negative conductance, but as mentioned earlier, due to the sensitivity of transconductance to process variation many researchers were led to investigate alternative solutions. One possible alternative is to use the conductance, $g_{ds}$, of an element rather than the transconductance to determine and generate a negative conductance [15]. This can be seen in Eq. 2, where the negative conductance component is calculated by $g_{dsn}(1-A)$, and therefore determined by the output conductance, $g_{dsn}$, of transistor $M_n$. In Figure 7 we can see a basic amplifier comprised of transistors M1 and M2 with a load capacitance of $C_L$. Furthermore, an additional transistor $M_n$ is connected to the output of the basic amplifier. Transistor $M_n$ is biased to always be in the saturation region and its gate source voltage is AC shorted through the use of a DC power supply $V_{xx}$. A low gain amplifier stage is placed between the

$$A_V(S) \approx \frac{-g_{m1}}{sC_L+g_{ds1}+g_{ds2}+g_{dsn}(1-A)}$$  \hspace{2cm} (2)
source and the drain of the transistor \( M_n \), as a result the conductance due to \( M_n \) at the output of the basic amplifier will be seen as \((1-A)g_{dsn}\). Similarly, the output gain, \( A_V(s) \), is found by dividing the output transconductance by the total output conductance as seen in Eq. 2 where \( g_{ds1}, g_{ds2} \) and \( g_{dsn} \) correspond to output conductances of transistors M1, M2 and \( M_n \) respectively. To achieve the overall output conductance of zero, where \( g_{ds1} + g_{ds2} + g_{dsn}(1-A) = 0 \), the gain \( A \) of the low gain amplifier stage must be greater than 1. A more precise cancelation of conductances can be achieved by finetuning \( g_{dsn} \), this can be done by biasing the current of transistor \( M_n \).

\[
\text{Figure 8 - Half Circuit of the Amplifier (Representation of One Input Path of a Differential Amplifier)} \ [15]
\]

The concept described and shown in Figure 7 allowed an amplifier to be designed, shown in Figure 8, and tuned in such a way that a gain of 83dB was achieved in a 500nm process as seen in Figure 9 [15]. In the design in [15], shown in Figure 8, the gain was dependent on a control voltage that determined a bias current through transistor \( M_n \) used for generating a negative conductance at the output. Changing the control voltage \( (V_{ctrl}) \) allowed the conductance to be varied and tuned in such a way that a very small overall

\[
\text{Figure 9 - DC Gain, } A_V(s) \text{ Versus the Control Voltage } (V_{ctrl}) \ [15]
\]
output conductance could be achieved and high gain could be observed. This again confirmed that a high gain still could not be achieved without further tuning and the issue of process variation could not be ignored.

Another method was proposed in [17], where it was attempted to eliminate the need for manual tuning and propose a technique that performed equally well across all process corners. This method also uses the conductance, in this case the output conductance, $g_{ds12}$ of transistor M12, rather than the transconductance of an element to generate a negative output conductance and makes this approach more robust when it comes to process and temperature variation. In this method differential signals from the cascode are sensed and a feedback signal is generated automatically to control the negative conductance. The concept diagram of this method can be seen in Figure 10.

The negative conductance $g_n$ and total output conductance $g_D$ can be found by Eq. 3a and 3b. In Eq. 3 $\eta$ is the ratio of body effect transconductance and transconductance ($g_{m_{b23}}/g_{m_{23}}$) of M23 that remains approximately constant under process and temperature variation. $A_L$ is a gain of a low gain amplifier. Since $g_D$ is found by a difference of $g_{ds12}$ and $g_n$, and $g_n$ is proportional to $g_{ds12}$,
the matching between $g_{ds2}$ and $g_m$, and therefore high gain can be maintained under process and temperature variation. The method proposed in [17] improves the gain of a conventional structure by at least 22dB when tested across all process corners in 130nm technology as seen in Table 6.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Conventional</th>
<th>Topology-[17]</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC Gain (dB)</td>
<td>51.9</td>
<td>79</td>
</tr>
<tr>
<td>Load C (pF)</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>GBW (MHz)</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>PM (°)</td>
<td>73</td>
<td>72</td>
</tr>
<tr>
<td>SR/SR- (V/µs)</td>
<td>38</td>
<td>38</td>
</tr>
<tr>
<td>Total Current (µA)</td>
<td>606</td>
<td>656</td>
</tr>
<tr>
<td>Supply Voltage (V)</td>
<td>1.2</td>
<td>1.2</td>
</tr>
<tr>
<td>Technology</td>
<td>IBM 0.13 µm CMOS</td>
<td>IBM 0.13 µm CMOS</td>
</tr>
</tbody>
</table>

Table 6 - Performance Comparison of Conventional and Proposed Amplifiers in Nominal Case (Typical-Typical Corner and T = 27°C) [17]

The output cancelation concept proposed in [17] was further explored and improved in [18] in order to simplify the design and reduce the number of required simulations. The design method was simplified to a point where essentially only the sizing of a single transistor needs to be considered. The schematic of the implementation can be seen in Figure 11. In Table 7 [18] it can be seen that this method achieves comparable performance to the method propose in [17] while the design procedure was simplified significantly. This highlights the need for easily implementable and easy to design solutions that can ideally be implemented in various technologies without much effort.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>DC Gain (dB)</td>
<td>87.4</td>
<td>134.6</td>
<td>131.4</td>
</tr>
<tr>
<td>Load C (pF)</td>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>GBW (MHz)</td>
<td>73.1/66.51</td>
<td>75.05/67.52</td>
<td>75.5/67.53</td>
</tr>
<tr>
<td>PM (°)</td>
<td>53.9</td>
<td>53.56</td>
<td>53.6</td>
</tr>
<tr>
<td>SR/SR- (V/µs)</td>
<td>51.2/51.3</td>
<td>51.7/52.3</td>
<td>51.3/51.9</td>
</tr>
<tr>
<td>1% Settling Time (ns)</td>
<td>50.8</td>
<td>40.0</td>
<td>40.0</td>
</tr>
<tr>
<td>Supply Voltage (V)</td>
<td>1.5</td>
<td>1.5</td>
<td>1.5</td>
</tr>
</tbody>
</table>

Table 7 - Performance Comparison of Conventional Amplifier and the Ones Examined in [17] and [18]
2.4.3 Conclusion

In the past, negative conductance approaches to gain enhancement have been explored but were rarely pursued commercially either due to the approach not being feasible due to process variation or it would require a calibration procedure not suitable for on-chip operation. Recently more robust approaches have been explored that perform a lot better even with process variation being taken into account. With the latest refinements it has become clear that negative conductance techniques can guarantee significant improvements in gain while being easy to design and not suffering significant drawbacks when compared to unmodified amplifier structures. This is a promising indication as it makes it likely that the Nauta structure will maintain high performance levels if an adequate tuning method is found.

Figure 11 - Schematic of the Topology-[18] Implementation
2.5 Nauta Structure

2.5.1 Introduction

The Nauta structure is an inverter based operational transconductance amplifier and was first proposed by Bram Nauta [19]. It relies on a negative conductance technique to maximise gain. The maximum gain is only achieved when “positive” output conductance is perfectly cancelled out with “negative” output conductance. This is impossible to achieve unless the actual device dimensions match the ideal design dimensions, which is unlikely to occur due to process variation. The Nauta structure derives the negative conductance from the transconductance of an inverter and as such can be affected by process variation to a significant degree as transconductance can be particularly sensitive to mismatch. As a result, it is critical to minimise the mismatch in the Nauta structure to achieve high gain.

Negative conductance or resistance is a property of a device that causes it to behave in an opposite way compared to a conventional resistor and was discussed in Ch. 2.4. This property is commonly exploited to achieve gain enhancement. A current in a negative resistor would decrease with an increase...
in voltage, power would be produced rather than consumed, and the increase in power can result in amplification of the signal. Within the Nauta structure, the negative conductance is only applied when differential inputs are supplied. To best understand this, consider a common mode voltage $V_c$ and differential voltage $V_{id}$, where at the first input $V_c + 0.5V_{id}$ is applied, and at the second, $V_c - 0.5V_{id}$ is applied. While $Inv1$ will be pushing the voltage down at the output node $V_{o1}$, $Inv2$ will be pushing the voltage up at output node $V_{o2}$. Similarly cross coupled inverters $Inv3$ and $Inv6$ will be pushing the voltage up at output node $V_{o2}$ and down at node $V_{o1}$ respectively, as a result the signals at both nodes will be amplified. This effectively allows inverters $Inv3$ and $Inv6$ to provide negative conductance at the outputs of the amplifier. Inverters $Inv4$ and $Inv5$ are connected in self-coupled configuration and as a result supply positive conductance at the outputs. The summary of output resistances as determined by the transconductances $gm_3$, $gm_4$, $gm_5$, $gm_6$ of inverters $Inv3$, $Inv4$, $Inv5$, and $Inv6$ respectively, can be seen in Table 8.

<table>
<thead>
<tr>
<th>Output Node</th>
<th>Common Resistance</th>
<th>Differential Resistance</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{o1}$</td>
<td>$\frac{1}{gm_5 + gm_6}$</td>
<td>$\frac{1}{gm_5 - gm_6}$</td>
</tr>
<tr>
<td>$V_{o2}$</td>
<td>$\frac{1}{gm_4 + gm_3}$</td>
<td>$\frac{1}{gm_4 - gm_3}$</td>
</tr>
</tbody>
</table>

Table 8 - Output Resistances of Nauta Structure [19]  

The overall DC gain of the Nauta structure can then be calculated using the following formula as seen in Eq. 4. The DC gain of an amplifier is determined by the ratio of transconductance (differential transconductance $g_{md}$ in the case of the Nauta structure) and output conductance $g_{out}$. The total output conductance of the Nauta structure is a sum of all output conductances of the inverters connected to the particular node ($g_{o1} + g_{o5} + g_{o6}$ for output node $V_{o1}$, corresponding to output conductances of $Inv1$, $Inv5$ and $Inv6$) as well as the transconductances of the self and cross coupled inverters ($Inv5$ and $Inv6$ respectively, for output node $V_{o1}$).

$$ A_0 = \frac{g_{md}}{g_{out}} = \frac{g_{md}}{g_{m5} - g_{m6} + g_{o1} + g_{o5} + g_{o6}} $$  \(4\)  

It can be seen that as long as the transconductance of inverter $Inv3$ and $Inv6$ is bigger than that of $Inv4$ and $Inv5$ respectively, and difference between them is equal to the sum of all conductances of all inverters connected to the
output node the DC gain can theoretically be infinite. That is impossible in practice and realistically the gain will be finite. The gain will be determined by the ratio of the transconductance of the feed forward inverters and the error between the ideal (required for maximum gain) and actual transconductance values (that are affected by mismatch) of self-coupled and cross-coupled inverters. The gain will decrease as the errors caused by mismatch get worse. At this point the importance of minimizing the mismatch errors to be able to achieve the maximum DC gain becomes obvious.

In his paper [19] Nauta suggests a possible way to deal with mismatch. He proposes a method that involves tuning the transconductance of inverters Inv3 and Inv6 by tuning the power supply of these inverters. This means that an additional tuneable power supply must be implemented for these inverters. Newer technologies already have a low headroom voltage therefore implementation of a tuneable power supply would result in an even lower usable voltage range for the amplifier, that would in turn affect the output voltage swing. This would essentially negate the inherent advantages of the Nauta structure that it has when being implemented in the latest technologies compared to more traditional amplifier topologies. As a result, while this method is plausible in theory it would have limited commercial applications.

However, if an alternative tuning method can be implemented, the Nauta structure can become very appealing. Its simplicity and the lack of internal nodes means that it will not have any parasitic poles and zeros that would influence its transfer function and it has the potential to have a relatively high bandwidth.

Figure 13 - Comparison of Band Pass Gain for 3 Different Designs (Where CC Inverter PMOS Widths Are 8.8µm, 8.9µm and 9.0µm) [20] © 2002 IEEE
2.5.2 Modifications and Applications

A very common use for the Nauta structure is as a transconductor in Gm-C filters. In [20] the viability of using the Nauta structure in band-pass filters with centre frequency of 3MHz and pass band of 1MHz was examined. The filters examined in [20] were implemented in 600nm and 450nm technologies. Even for these technologies mismatch and process variation was a significant concern and the authors explored the possibility to overcome this through the use of large components. Due to lack of internal nodes the viability of implementing low frequency filters with large transistors and large capacitors, which would in turn minimise the mismatch caused by process variation, was explored. This may be especially suitable for low frequency applications as the transconductance may be lower.

Figure 13 compares the designs when the CC inverter's PMOS widths were set to 8.8µm, 8.9µm and 9.0µm, while SC widths were kept constant at 8.8µm. The difference in performance between these setting is acceptably small, where the pass band gain difference for 8.8µm and 8.9µm settings is only 1.1dB. This makes the Nauta structure robust enough for low gain applications. However, this research may not be a good representation of the performance of the Nauta structure that may be implemented with even smaller technologies where the process variation may be a lot worse. Especially considering that the Nauta structure may not be all that robust with stricter, high gain and bandwidth requirements.

![Diagram of Nauta Structure](image)

Figure 14 - Various Approaches to Varying FF Inverter Dimensions [21]
In [21] an approach where the properties of the Nauta structure could be tuned through variable arrays of transistors was explored. In [21] the Nauta structure was used in a $g_m$-$C$ biquadratic low pass filter. The filter was designed in 130nm technology in such a way that the bandwidth could be tuned from 100kHz to 20MHz. The bandwidth could be tuned by adjusting the transconductance of the Nauta structures in the filter by varying the effective dimensions of the FF inverters.

Figure 14 shows the basic methods of adjusting the effective transistor width by enabling transistors in parallel and effective length by enabling transistors in series. The series transistors are controlled by connecting the drains to $V_{DD}$ (PMOS) or GND (NMOS) via a switch as seen in Figure 14b. To increase the effective length more switches need to be opened. Figure 14c shows the width control mechanism, where the width is adjusted by opening and closing the switches at the gate. The biggest disadvantages of this method are that it doesn't allow independent C tuning and input capacitance along with the ON resistance of the switch forms a parasitic pole. In Figure 14d a more advanced switching method is shown that allows a more precise control of the capacitance. If both D and S signals are high, transconductance ($g_m$) and capacitance (C) are increased. If only S signal is high only C is increased. When both signals, D and S are low, $g_m$ and C are largely unaffected. This occurs as when S is low, signal S' is high and therefore connects the source of the bottom transistor to $V_{DD}$ which results in a negative overdrive voltage and hence, lower input capacitance. Having a greater control over the input capacitance and hence the bandwidth of the structure may be beneficial in certain applications.

Overall this research demonstrates that it is entirely possible to effectively adjust the transconductance of the Nauta structure through the use of controllable arrays of inverters. However the tuning methods or their complexity are not examined in great detail, nor is the feasibility of tuning the structure for high gain.
In [22] the performance of the Nauta structure is examined further and a new topology with improved characteristics is proposed. The motivation behind the modifications is to improve the frequency performance and improve output common mode stability. The modifications include the addition of two inverters cross-connecting the inputs and the outputs as seen in Figure 15.

To be more specific, the authors point out that the common mode rejection ratio (CMRR) which in the case of the Nauta structure is determined by the ratio of common mode and differential transconductance gets worse at higher frequencies. Table 9 shows the performance of the topology-[22] Nauta structure compared to the original topology and a traditional wide swing fully differential OTA when simulated in 350nm technology with a 50 Ohm load. As predicted the modified structure exhibits a better CMRR. Additionally the modified structure has a high output voltage swing and high gain bandwidth product. Naturally the addition of the inverters comes at a cost of increased power consumption and increased input capacitance. The power consumption has increased by 100µW from 700µW of the original structure to 800µW of the modified structure.

Although the performance increases are significant it remains to be seen if the additional power consumption is justified for all applications, moreover it is important to consider that the addition of two inverters to a structure made of six inverters is a relatively substantial increase in complexity and could require a more complex tuning procedure if effects of process variation are taken into account.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Topology-[22] Nauta</th>
<th>Nauta</th>
<th>Traditional OTA</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOS-I(_{DS})-DC signal path</td>
<td>40</td>
<td>40</td>
<td>I(_{D})=40</td>
<td>µA</td>
</tr>
<tr>
<td>Differential Transconductance G(_{MD})(Z=50 Ω)</td>
<td>108.4</td>
<td>108.4</td>
<td>379</td>
<td>µΩ(^{-1}) (µS)</td>
</tr>
<tr>
<td>Common Mode Transconductance G(_{CM})(Z=50 Ω)</td>
<td>3</td>
<td>217</td>
<td>13.8</td>
<td>µΩ(^{-1}) (µS)</td>
</tr>
<tr>
<td>1° phase deviation of G(_{MD})</td>
<td>239</td>
<td>136</td>
<td>24</td>
<td>MHz</td>
</tr>
<tr>
<td>CMRR@100kHz</td>
<td>-31</td>
<td>3</td>
<td>-25.7</td>
<td>dBc</td>
</tr>
<tr>
<td>Differential Input Capacitance</td>
<td>85.15</td>
<td>48.9</td>
<td>9.65</td>
<td>fF</td>
</tr>
<tr>
<td>Output Current Swing</td>
<td>254</td>
<td>254</td>
<td>170</td>
<td>µA</td>
</tr>
<tr>
<td>Maximum Output Voltage Swing</td>
<td>2.27</td>
<td>1.67</td>
<td>2.5</td>
<td>V</td>
</tr>
<tr>
<td>Output Resistance R(_{O})</td>
<td>338</td>
<td>350</td>
<td>66</td>
<td>kΩ</td>
</tr>
<tr>
<td>Open Loop: DC Gain A(_{O})</td>
<td>31.3</td>
<td>27.8</td>
<td>28</td>
<td>dB</td>
</tr>
<tr>
<td>Open Loop: GBWP</td>
<td>3.56</td>
<td>0.94</td>
<td>7.08</td>
<td>GHz</td>
</tr>
<tr>
<td>Phase Margin</td>
<td>77</td>
<td>92</td>
<td>27</td>
<td>°</td>
</tr>
</tbody>
</table>

Table 9 - Performance of the Modified Structure Compared to the Original Nauta Structure [22]

In [23] an implementation of a band pass filter based on a Nauta structure is once again explored. This approach is to a certain degree similar to the one
examined in [21] as it also uses an array of components where a variable number of said components can be enabled or disabled to implement a tuneable transconductance element.

Figure 16 shows a CMOS double pair that is used in the proposed topology-[23]. A number of these cells can be switched on and off to vary the transconductance of the Nauta structure FF inverters as seen in Figure 17. The CMOS double pair can be considered a transconductance cell as long as all transistors operate in saturation. When a twin well process is used the bulks of the transistors can be connected to their own source terminals to eliminate the body effect. Eq. 5a – 5h approximates the behaviour of the cell using simplified square-law where channel length modulation is not taken into account. Eq. 5h shows that the transconductance of the cell can be varied using the bias voltages $V_{G1}/V_{G2}$.

\[
\frac{1}{\sqrt{K_{eff}}} = \sqrt{\frac{1}{K_n} + \sqrt{\frac{1}{K_p}}} \quad \text{(5a)}
\]

\[
I_0 = I_1 + I_2 - 2K_{eff} [V_{G1} + V_{G4} - \Sigma V_I] \Delta V_T + K_{eff} [V_{G1} + V_{G4} - \Sigma V_I] \quad \text{(5b)}
\]

\[
\Sigma V_I = V_{Tn1} + V_{Tn3} + |V_{TP2}| + |V_{TP4}|
\quad \text{(5c)}
\]

\[
\Delta V_T = (V_{Tn3} - V_{Tn1}) + (|V_{TP4}| - |V_{TP2}|) + (V_{G1} - V_{G4})
\quad \text{(5d)}
\]

\[
V_{Tn1} = V_{Tn3} = V_{Tn0}
\quad \text{(5e)}
\]

\[
V_{TP2} = V_{TP4} = V_{TP0}
\quad \text{(5f)}
\]

\[
I_0 = g_m V_I
\quad \text{(5g)}
\]

\[
g_m = -4K_{eff}(V_g - V_{Tn0} + |V_{TP0}|)
\quad \text{(5h)}
\]

As can be seen in Figure 17 a rather simple 4 element array is used, but the tuning precision is supplemented by variable biasing of these array elements. It is not clear however, if the use of additional biasing circuits is more beneficial area, complexity and power consumption wise, compared to having
more elements in the inverter array that would also allow the same tuning precision. Additionally, biasing circuits require a higher power supply voltage of ±1.5V compared to ±0.9V used for the rest of the circuit that was implemented in TSMC 180nm process. Using larger supply voltages for biasing circuits will not be feasible in even smaller, deep sub-micron, technologies due to voltage headroom restrictions.

In [24] another topology, as seen in Figure 18a, based on the original Nauta OTA is examined. The same concepts are then used to implement a 7th order Chebyshev low-pass filter.

![Figure 18 - a) Topology-[24] Nauta Structure and b) Gyrator Implementation [24]](image)

The topology-[24] Nauta structure (Figure 18a) is used for implementation of a gyrator (Figure 18b). A derivation of the gain for this topology shows that the differential gain is controlled by the cancelation of output conductances and transconductance of cross- and self-coupled transconductors as in other Nauta structures. Differential gain is primarily determined by cancelation of output conductances, of all transconductors, the transconductance of SC and transconductances of CC transconductors. However additionally, it is shown that the topology also achieves 0 common mode gain.

In [25] we start to see a merger of previously examined research. Here the Nauta structure topology first seen in [22] is examined once gain and used to implement a tuneable gm-C band pass filter. Here the tuning is achieved through the use of CMOS double pair inverters as seen in [23] and achieves similar results of varying effective dimensions of the inverter by enabling a varying number of individual components in the array seen in [21].
In [25] the Nauta structure was implemented in 180nm process and the performance was compared to its topology-[25] variant. Both variants were implemented with double CMOS pair inverters. As seen in Table 10, even with addition of the double CMOS pair inverters the performance improvements of the topology-[25] Nauta structure remained in the form of reduced common mode transconductance \( G_{CM} \) and improved CMRR, while the differential transconductance \( G_{MD} \) remained virtually identical. An improvement of 10dB in CMRR was seen at the cost of increased power consumption, which increased by 22.276\( \mu \)W from 56.05\( \mu \)W to 78.326\( \mu \)W when comparing traditional and modified Nauta structures.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Nauta OTA</th>
<th>Topology-[25] Nauta OTA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter / Type</td>
<td>Double CMOS</td>
<td>Double CMOS</td>
</tr>
<tr>
<td>( G_{MD} (\mu S) (Z=50\Omega) )</td>
<td>59.57</td>
<td>59.59</td>
</tr>
<tr>
<td>( G_{CM} (\mu S) (Z=50\Omega) )</td>
<td>117.75</td>
<td>10.27</td>
</tr>
<tr>
<td>CMRR (dB)</td>
<td>-5.92</td>
<td>15.27</td>
</tr>
<tr>
<td>PSRR (dB) @10MHz</td>
<td>35.76</td>
<td>46.09</td>
</tr>
<tr>
<td>Power (( \mu )W)</td>
<td>56.05</td>
<td>78.326</td>
</tr>
</tbody>
</table>

Table 10 - Performance of the Topology-[25] Nauta Structure with Double CMOS Pair Inverters

In [26] an approach based on a technique named “component redundancy and random diversity” (CRRD) is examined. Although this technique is not implemented with the Nauta structure, it is used to obtain high gain using the positive feedback principle. This research was not discussed in the previous

![Figure 19 - Positive Feedback Amplifier with CRRD][26] © 2012 IEEE
sections as it is far more relevant to assessing the feasibility of using variable arrays of parallel components, similar to those proposed in [21], [23] and [25].

The amplifier structure examined in [26] can be seen in Figure 19. Similarly to Nauta structure, the high gain is achieved by matching and cancelation of the overall output conductances. The circuit in Figure 19 is broken down into two parts, coarse load formed by transistors M2a and M2b, and a tuneable load, denoted as “-ve g\text{m} array”. In this case the gain is determined by the sum of output conductances of inverters M1 and M2 (g_{ds1} and g_{ds2}) as well as the difference between the conductance and transconductance of inverters in the tuneable array consisting of inverters Mf1-Mf8 (g_{dsf(1-8)} - g_{mf(1-8)}). If the total sum of the transconductances of the “-ve g\text{m} array” and conductances of both loads becomes close to zero, the amplifier will exhibit high gain.

\[
N = \log_2 \left( \frac{\phi^{-1}(\alpha) \sigma_{gres}}{g_{res} - g_{mf}} + 1 \right) \tag{6}
\]

In [26] it is suggested that it may be beneficial to embrace the variation of parameters in the components caused by process variation in order to achieve high gain. As seen in Figure 20, instead of trying to have an array of uniformly sized components it may be beneficial to use the components that are smaller due to process variation and as a result achieve a higher tuning resolution (N). N can be found using Eq. 6, where \( \alpha \) is a yield level guaranteeing high gain (and \( \phi \) represents standard Gaussian distribution function), g_{res} is the overall output conductance (and \( \sigma_{gres} \) is its standard deviation) of the coarse load and g_{mf} is the overall output conductance of the tuneable load. To guarantee that for any given number of array elements there

![Figure 20 - Comparison of Array Sizing](image)
will be enough of the smaller elements, not surprisingly, the total number of the elements needs to be increased. Figure 21 shows the increase in effective resolution as the number of elements in the array is increased.

![Figure 21 - Increase in Resolution Due to Increase in Array Size](image)

When the circuits were implemented in a 180nm process and 60 chips were tested the results were very promising. Without calibration of amplifiers a minimum of 38dB and an average of 42dB gain was achieved, after calibration the amplifiers were able to achieve a gain of at least 58dB. Although the gain would drop if the temperature changed, the amplifiers could still be successfully recalibrated to adapt to the new temperature as seen in Figure 22.

![Figure 22 - Performance of the Fabricated Circuit](image)

In [27] the idea of having a robust circuit capable of maintaining high performance levels across PVT variations was examined again. The proposed method involves tuning of the power supply voltage to adjust the differential transconductance of the Nauta structure much like in the original paper by B. Nauta. Here however a robust circuit that has to be tuned only once at production is proposed as seen in Figure 23. The proposed bias circuit sets the supply voltage VDD used to control the transconductance of the Nauta structure. Initial tuning of the VDD voltage can be achieved by trimming the values of the resistor R and the structure of the circuit will allow the VDD voltage to be set in such a way that the Nauta structure will maintain a constant transconductance across temperature variations.
The transconductance of the Nauta structure set by the VDD voltage of the bias circuit shown in Figure 23 is sensitive to mismatch between $K_n$ and $K_p$ (where $K = \mu C_{ox} W/L$ for NMOS and PMOS respectively) that vary across process, voltage and temperature changes. However the effect of the variations in the difference between $K_n$ and $K_p$ is minimal as the bias voltage and hence the transconductance of the Nauta structure is dependent on the difference between $K_n$ and $K_p$ quadratically, making it less sensitive to changes in the difference between $K_n$ and $K_p$. A Butterworth band pass filter with centre frequency and bandwidth of 2MHz was implemented in 180nm technology using Nauta structure with the proposed bias circuit. Although there was only 2% variation in centre frequency and bandwidth of the filter, as seen in Figure 24, the increase in power and area requirements compared to the basic Nauta structure are not considered.

![Complementary Constant-$g_m$ Voltage Bias Circuit with Integrated Regulator][27]

**Figure 23** - Complementary Constant-$g_m$ Voltage Bias Circuit with Integrated Regulator [27]

![Percentage Deviation Over Temperature][27]

**Figure 24** - Percentage Deviation Over Temperature [27]
2.5.3 Conclusion

Overall we can see that aside from Nauta structures implemented in older technologies with larger feature size the issue of mismatch caused by process variation can't be ignored. This issue is commonly tackled by having some sort bias circuit or having an array of elements that can be enabled or disabled to control the performance of the structure, or a combination of both. Both methods have proven to be effective in improving the performance. However it is often not clear if it is feasible to use the circuit in practical applications due to increased power consumption or the need for a tuning procedure that can't be implemented on-chip. As a result, further investigations are needed to devise a robust and easily implementable technique to maximise the performance of the Nauta structure.
2.6 Digital to Transconductance Converter Based Nauta Structure

To tune the Nauta structure for higher gain it needs to be possible to adjust the transconductance and output conductance of individual inverters. Instead of adjusting the supply voltages or biasing voltages of individual inverter it is possible to adjust the widths of the inverters. This is hard to achieve unless further modifications are made to the Nauta structure. One way of achieving this is by replacing each inverter in the original structure with an array of tri-state inverters as seen in Figure 25 [28].

A tri-state inverter consists of a regular inverter and an additional PMOS and NMOS control transistor inserted between the original inverter and the power supply and ground respectively. The widths of the control transistors are equal to the widths of the transistors in the original inverter. The gates of the control transistors are connected to a digital control signal, EN (where EN_bar is the EN signal after inversion), that can effectively enable or disable the inverter. If each inverter in Figure 12 (Inv1-Inv6) can be replaced by an array of tri-state inverters of varying sizes in parallel, where each individual tri-state inverter can be controlled by a separate control signal, a digital-to-transconductance converter will be formed [28]. Each control signal only needs to be toggled between high and low (VDD/GND) and therefore can be implemented using a binary/digital signal. Therefore a combination of all the individual control signals can be represented by a binary number or a control code. Output transconductance of an inverter is determined by the widths of the transistors. By changing the control code and enabling a varying number of inverters in parallel the overall transconductance will vary as well, in other words a digital control value is converted into a transconductance.

Figure 25 - Tri-State Inverter [28]
Digital to transconductance converters in turn allow the Nauta structure to be tuned for high DC gain and to deal with mismatch. The actual precision with which the gain can be tuned will depend on several properties of the arrays. Firstly, the size of the individual inverters, where smaller inverters would allow high tuning precision. And secondly, the overall number of inverters, where the more inverters you have the higher the likelihood that a high gain setting will be found. Increasing the number of controllable tri-state inverters will require a larger number of control signals and hence there will be a larger number of possible control code combinations.

For this work two different modified, TDC based, Nauta structure prototypes were made available for experimentation and testing (schematics and physical prototypes). One implemented in 180nm technology, and one implemented in 65nm technology [28]. The two prototypes use different digital-to-transconductance converter structures described in Ch. 2.6.1-2.6.2.

2.6.1 180nm Prototype

Figure 26 shows the general composition of the digital to transconductance converters used in the 180nm prototype, where all the inverters in each digital to transconductance converter (Inv1-Inv6) and their sizes can also be seen. There are 8 tri-state inverters in of each of the FF inverter arrays and 7 in SC and CC inverter arrays. Most inverters are controlled by a unique control signal, but some of the controls are shared to reduce the overall number of control codes, for example the same control bit is used to switch on both of the largest FF inverters (S1) and one control bit is used for all of the largest SC and CC inverters (S2), and finally the second largest SC and CC inverters are controlled by one control bit as well (S3 and S4 respectively).
Each tri-state inverter can be further broken down into unit inverters. Unit inverters can't be controlled individually and can only be switched on or off as a group. Using varying number of unit inverters allows tri-state inverters of varying sizes to be designed quickly and efficiently. For example, “×10” inverter consists of 10 unit inverters, the size of the unit inverter can be seen in Table 11, and therefore the “×10” inverter is 10 times as big as a unit inverter. The sizes of the other tri-state inverters can also be seen in Table 11. Additionally, the two smallest tri-state inverters are sized to be as big as a half and quarter of the unit inverter. Overall there are 38 distinct control signals that form a 38 bit control code that needs to be generated for this particular version of the Nauta structure. To program the structure each inverter from Inv1 to Inv6 is assigned a specific code that is simply an integer, where in a binary format each bit controls a single tri-state inverter. The LSB corresponds to the smallest block and MSB to the largest.

### 2.6.2 65nm Prototype

Just like the 180nm prototype, the 65nm version [28] also consists of digital to transconductance converters based on tri-state inverters. This version however, consists of equally sized DTCs, where all of the converters (FF, SC and CC) are made up of 10 individually controlled tri-state inverters. In this prototype there are also no shared control signals, so there is one control signal for each inverter. Each inverter is sized in such a way that the width is increased by a factor of approximately 1.8 (limited by the minimum width changes allowed for the TSMC 65nm technology used for implementation) starting with the smallest inverter with PMOS width of 240nm.

The actual width of inverters can be seen in Table 12. The primary advantages of the 65nm prototype over the 180nm one is greater controllability with no shared control bits. This is ideal for testing purposes even though it may not be necessary in the final version. Additionally 65nm prototype can achieve a higher tuning precision due to the reduced dimensions of the smallest inverter in the digital to transconductance converter arrays.
“×1.8” sizing of the elements in the array guarantees a continuous range of transconductance, even under process variation [28]. This is achieved by ensuring that when a single large component in the array is switched on, while at the same time a large number of smaller components are switched off (example: transitioning from control code 01111111 to 1000000000), the actual size of the large component isn’t larger than the width of the inverters that were switched off due to “×1.8 sizing”. If the component is in fact a lot larger due to process variation the digital to transconductance converter will have a range of transconductance values that it cannot produce. In other words the difference between two digital input values corresponds to unexpectedly large difference between two output transconductance values. Such errors are defined as differential nonlinearity (DNL) errors. It is also

<table>
<thead>
<tr>
<th>Control Bit</th>
<th>PMOS W (µ)</th>
<th>NMOS W (µ)</th>
<th># of fingers</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>48.64</td>
<td>24.32</td>
<td>16</td>
</tr>
<tr>
<td>1</td>
<td>26.88</td>
<td>13.44</td>
<td>16</td>
</tr>
<tr>
<td>2</td>
<td>14.72</td>
<td>7.36</td>
<td>16</td>
</tr>
<tr>
<td>3</td>
<td>8.32</td>
<td>4.16</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>4.62</td>
<td>2.31</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>2.56</td>
<td>1.28</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>1.40</td>
<td>0.70</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>0.78</td>
<td>0.39</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>0.43</td>
<td>0.215</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>0.24</td>
<td>0.12</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 12 - Inverter Sizing of 65nm Prototype

<table>
<thead>
<tr>
<th>Scaling Ratio</th>
<th>Mean (mS)</th>
<th>Std. Deviation (mS)</th>
<th>% Positive DNL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.7</td>
<td>2.56</td>
<td>1.32</td>
<td>2.6</td>
</tr>
<tr>
<td>1.8</td>
<td>2.43</td>
<td>1.30</td>
<td>3.0</td>
</tr>
<tr>
<td>1.9</td>
<td>1.86</td>
<td>1.15</td>
<td>5.2</td>
</tr>
<tr>
<td>1.95</td>
<td>1.04</td>
<td>0.98</td>
<td>14</td>
</tr>
<tr>
<td>1.975</td>
<td>0.67</td>
<td>0.96</td>
<td>24</td>
</tr>
</tbody>
</table>

Table 13 - Mean and Standard Deviation of Transconductance Difference at Worst Case Transition Under Monte-Carlo Mismatch Analysis [28]
important to note that only “positive” DNL errors are an issue, where the expected changes in transconductance are bigger than the ideal change. If the change is smaller, and therefore causes a “negative” DNL error, it may even be beneficial as it will allow the transconductance value to be tuned with higher precision by having wider ranges of control codes with overlapping transconductance values.

Table 13 shows the results of Monte-Carlo simulations for different scaling ratios of the components in the arrays. As can be seen for scaling ratio of 1.8, only 3% of simulation runs showed that there was a positive DNL error. In practice a realistic requirement is to be able to state that a product operates correctly with a 95% confidence level, making ×1.8 the highest scaling ratio to meet that requirement. The Nauta structure implemented using 10 element digital to transconductance converters with “×1.8” scaling ratio was simulated for 65nm technology. Figure 27 shows the feasibility of reliably obtaining high gain for various CC array settings by sweeping through a range of SC settings. For every CC code that was tested a gain of over 50dB could be achieved by sweeping through SC codes and in some cases even higher gain could be achieved.

![Figure 27 - Simulated Op-Amp DC Gain versus Self-Coupled Code-Word, for Steps in Cross-Coupled (CC) Inverter Code-Word](28) © 2014 IEEE

Overall it can be seen that the designs of the digital to transconductance converters can vary significantly and might yield different results and achieve different performance levels. One thing that is common to both designs is that a number of possible control codes is very large, even with shared control codes. This is likely to remain true in future implementations. To avoid testing all of the possible control code combinations and to ensure that it is feasible to use Nauta structure in commercial products, an efficient calibration technique is required, which will be the focus of the remainder of this thesis.
3. Theory and Solutions

3.1 Introduction

This chapter will cover the specifications and requirements of the calibration procedure. Based on the requirements a number of approaches that can be used for calibration will be compared and contrasted. Several of the strategies that are most likely to achieve the set requirements will then be proposed and described in more detailed. In subsequent chapters, through simulations and testing, the best strategy will be chosen.

3.2 Requirements and Calibration Approaches

The main goal of this thesis is to develop a calibration algorithm that can be used to tune the Nauta structure for high gain. Considering that the gain expected from a simple inverter is expected to be only approximately 20dB, a high gain will be defined to be in range from 100 to 1000 linear gain (40dB to 60dB). Anything above 60dB will be considered very high gain. The general requirements of the calibration algorithm are:

A. **Simplicity**: the algorithm must require as little additional hardware as possible to minimise power consumption and area requirements.

B. **Speed**: the tuning must be done as quickly as possible.

C. **Integration**: it must be possible to integrate the additional hardware with the Nauta structure and implement it on-chip.

In the previous section a number of designs that allow the Nauta structure to be tuned were examined. The types of the designs can be primarily split into two categories: biasing [23, 25, 27] and digitally controllable arrays [21, 26, 28]. It was shown that any modifications or calibration schemes come at a cost of increased power consumption and increased area requirements. Additionally, biasing techniques require a higher power supply voltage compared to the rest of the circuit to operate. That makes these biasing approaches to tuning unsuitable for implementation in deep sub-micron technologies that have to run on low power supply voltages to prevent breakdown of the devices. Considering that the Nauta structure becomes particularly attractive compared to traditional structures when implemented in deep sub-micron technologies it was decided to use a purely digitally controllable Nauta structure for implementation of the calibration procedure.

A number of different designs of a digitally controllable Nauta structure [21, 26, 28] were examined in the previous section as well, including the prototypes that will be used for implementation of the calibration algorithm [28]. The research examined there is primarily focused on asking how to design the structure that can potentially achieve high gain if tuned correctly and how precisely will the structure allows the parameters and gain to be tuned.
other words the research is focused on the design of the tuneable structure rather than the mechanism by which the structure can be tuned.

The tuning procedure can be split into two main components: code searching and gain measurements. A number of tuning approaches are summarised in Table 14 and Table 15. Table 14 summarises the code searching approaches and Table 15 summarises gain measurement approaches. A very straightforward approach to tuning is to go through every single control code and see which one has the highest gain. A 65nm prototype is controlled by 60 control bits, this translates to $2^{60}$ (or $>1.1 \times 10^{18}$) possible permutations. Even if we remove certain control code combinations, such as the lowest 10% of the settings (due to low overall transconductance and low tuning precision) or take into account that the CC codes would have to be greater than the SC codes (as discussed in Ch. 2.5.1), we would still be left with unreasonably high number of control codes to test. A complete opposite of searching all the codes, and an equally unrealistic approach, would be to statistically estimate a high gain code. This of course would defeat the purpose of having a controllable structure that can be tuned. A more reasonable approach would be to combine the previously mentioned approaches by conducting a limited sweep to search the codes around an estimated high gain code. This approach will enable the tuning procedure to be completed in a reasonable amount of time and exploit the advantages of a digitally controllable structure to achieve a higher gain.

<table>
<thead>
<tr>
<th>Search Method/Design Strategy</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistical estimation of a high gain code</td>
<td>Requires no searching time</td>
<td>Mismatch makes the likelihood of achieving high gain unlikely</td>
</tr>
<tr>
<td>Localised sweep around a statistically estimated code</td>
<td>Requires little searching time</td>
<td>Highest possible gain cannot be guaranteed</td>
</tr>
<tr>
<td>Search all codes combinations</td>
<td>Thorough, highest possible gain will be found</td>
<td>Will take an unreasonably long amount of time</td>
</tr>
</tbody>
</table>

Table 14 - Code Searching Approaches

Similarly, a number of approaches for measuring the actual gain need to be considered. Firstly the gain can be measured by simply applying a differential input to the Nauta structure and measuring the differential output. This is a straightforward approach, but would require an ADC which may have a high cost in terms of power consumption, area and design time. It may also be possible to measure output current and estimate the output conductance and transconductance of the structure and therefore measure the gain. Similarly however, this will require additional and precise circuitry to measure the current. Alternatively, approaches more suitable for digital circuitry can be employed instead. Firstly, changes in phase can be tracked and measured as the gain changes. Such approach would require a comparator and a high speed clock that could be used to measure the time for which the output voltage and the input voltage differ. In contrast to previous approaches, this one requires
less complex analogue hardware (in a form of a comparator) and uses more digital hardware instead, that has a potential be a lot more efficient when it comes to power consumption and area (as discussed in Ch. 2.3). Similarly, another approach utilising digital circuits relies on detection of hysteresis. A hysteresis can be detected by connecting the outputs of the Nauta structure to the inputs of a comparator. The output of the comparator will be different for low to high and high to low input sweeps if the hysteresis is present. This approach also requires less complex analogue hardware in a form of a comparator and voltage ladder for inputs. This approach will be discussed in greater detail.

<table>
<thead>
<tr>
<th>Gain Measurement Method</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voltage</td>
<td>Accurate</td>
<td>Requires complex circuits for measurement (i.e. ADC)</td>
</tr>
<tr>
<td>Current</td>
<td>Accurate</td>
<td>Requires complex circuits for measurement (i.e. ADC)</td>
</tr>
<tr>
<td>Hysteresis</td>
<td>Requires simple circuits for measurement (i.e. comparator)</td>
<td>Requires precise voltage references</td>
</tr>
<tr>
<td>Phase</td>
<td>Requires simple circuits for measurement (i.e. comparator)</td>
<td>Requires high-speed clock</td>
</tr>
</tbody>
</table>

Table 15 - Gain Measurement Approaches

Let’s examine a general approach to tuning that can be realistically completed in a short amount of time in laboratory conditions:

**A.** Choose a self coupled (SC) inverter code that will be held constant.

The choice of a starting SC code is a trade-off between tuning precision and power consumption. Having a higher SC code means that the smallest SC/CC inverters will become an even smaller proportion of the overall size of the structure and therefore allow higher tuning precision but more power will be consumed. A loose restriction on the choice of SC is also placed to minimise the common mode gain. If the common mode gain is to be kept below 1, the sum of the transconductances (or sizes) of SC \((gm_{sc})\) and CC \((gm_{cc})\) inverters must exceed the transconductance of FF \((gm_{ff})\) inverters as seen in Eq. 7a. If

\[
\left(\frac{V_{OC}}{V_{oc}}\right) = \frac{gm_{ff}}{gm_{sc} + gm_{cc}} < 1 \quad b) \quad A_0 = \frac{g_{md}}{g_{out}} = \frac{g_{md}}{g_{m5} + g_{m6} + g_{o1} + g_{o5} + g_{o6}} \quad (7)
\]

it becomes necessary to increase the gain or transconductance of the Nauta structure by increasing the size of FF inverters the size of SC inverters would have to be increased to maintain low common mode gain. The SC inverter will also have to be smaller than the CC inverters that will have to be used to cancel out the positive output conductance. Other than that there are no strict requirements for choosing a SC code as the tuning can be done with the CC and FF inverters.
B. Continuously increase cross coupled (CC) codes until high gain is achieved.

This is a straightforward sweep of CC codes that are greater than the chosen SC code. The codes should be swept continuously by increasing CC codes by 1 to achieve highest tuning precision. The code search times are reduced as we only search through CC codes, instead of CC and SC which would increase the number of codes needed to be searched quadratically.

C. Fine tune the gain by varying feed-forward (FF) inverter codes.

FF inverters can be used to fine-tune the gain as they only contribute their output conductance \( (g_{out}) \) to the calculation of the overall output conductance \( (g_{out}) \) as seen in Eq. 7b, which is significantly smaller compared to transconductance and conductance that SC \( (g_{m5} \) and \( g_{o5} \)) and CC \( (g_{m6} \) and \( g_{o6} \)) inverters contribute to the overall output conductance. Once the gain has been pre-tuned with CC inverters the gain can be further increased by reducing FF inverters and cancelling out the overall output conductance with higher precision. Although the gain is also determined by the transconductance of FF inverters, only a minor decrease in FF inverters will be required and the change in gain due to better output conductance cancelation will outweigh the loss in gain due to decrease in transconductance of FF inverters.

3.3 Detection of Hysteresis

![Figure 28 - Schmitt Trigger](image)

The issue of identifying a point when the structure is tuned to high gain can be solved by exploiting the positive feedback nature of the CC inverters. Consider a basic example of positive feedback being used in a Schmitt trigger design in Figure 28. Here an amplifier is being used as a comparator with positive feedback. The amplifier changes output voltages from positive saturation voltage to negative saturation voltage \((+V_s/-V_s)\) depending on the input voltage. A voltage divider comprised of R1 and R2 determines the
positive feedback voltage acting as a reference. As a result the hysteresis voltage is determined by Eq. 8.

\[
V_{+fb} = \pm V_s \frac{R_2}{R_1 + R_2}
\]  

(8)

As can be seen in Eq. 8 and Figure 28, increasing the positive feedback voltage by increasing \(R_2\), also increases the hysteresis width. Effectively the positive feedback causes the output to resist change. A similar principle can be applied to the Nauta structure. Increasing the width of the transistors in the CC inverters will also increase the positive feedback. Excessive increase of the width of CC inverters will cause outputs to resist change and will require higher input voltage for Nauta structure to invert the outputs. Therefore increasing the CC inverter widths will correspond to an increase in hysteresis. If the hysteresis can be detected and measured, and the width of the hysteresis can be correlated with the width of the transistors, it can be used as a measurement technique for measuring the unknown widths of the mismatched devices in the Nauta structure and therefore tune it.

The basic premise behind the proposed tuning solution is that if the size of CC inverters exceeds the size of SC inverters by a significant margin and exceeds the size needed for maximum gain, hysteresis will be observed as seen in Figure 29. If the hysteresis can be detected as soon as it appears while continually adjusting the effective width of CC inverters, and then revert to the last setting where no hysteresis was detected, it is reasonable to expect that the structure will exhibit a high DC gain, as the difference between ideal output conductance and the actual output conductance will be limited by a difference in output conductance caused by a change of 1 code.

![Figure 29 - Examples of Differential Input Sweep Results, Each of the Outputs Plotted Individually. A) No Hysteresis, No Offset B) No Hysteresis, With Offset C) Hysteresis, No Offset](image-url)
To exploit the presence of hysteresis to obtain high gain, methods of hysteresis detection need to be considered. Firstly considering that with hysteresis the structure will not behave as an ideal inverter, if a sinusoidal input signal is applied at the inputs, we can expect a distorted waveform at the output, as seen in Figure 30. The spectrum of the output signal can then be analysed for the presence of harmonics, and the presence of harmonics can be used as an indication for the presence of hysteresis. However this method may not meet the requirements that would allow it to be implemented on-chip. An on-chip signal generator would be required as well as a precise ADC with a high sampling rate. The addition of these circuits would likely require a significant amount of area and the signal analysis will likely require a significant amount of time to complete. In general, implementation will require an ADC, and having to implement an ADC just to enable the calibration of another amplifier (the Nauta structure) that could also be possibly used in an implementation of an ADC, could be considered counterproductive.

3.4 Proposed Calibration Approaches

Alternatively the basic principles of the methods described above can be distilled into something simpler. Instead of using a continuous test signal a small number of test voltages could be used instead that could be implemented with a simple voltage ladder. And instead of trying to analyse the spectrum of the output signal, changes in expected output voltage could be detected. Voltage changes detection could potentially be done with a simple comparator and a few reference voltages. A number of simple hysteresis detection approaches that will be investigated in greater detail to assess their feasibility are described below:

![Figure 30 - Waveform Distortion Due to Hysteresis](image-url)
A. Hysteresis Detection – Output vs. Reference Method

If the structure operates with no hysteresis when a differential input voltage is set to be 0 (both inputs are set to ideal common mode voltage) or to very small differential input voltage, the differential output voltage is expected to be 0 (both outputs are equal to ideal common mode voltage) or very small simply because there is nothing to amplify. On the other hand if the hysteresis is present for the same 0V differential input, the differential output voltage will still be high. Therefore hysteresis can be detected by comparing the output voltage (positive or negative) to a reference voltage. A possible test circuit for this is shown in Figure 31. A reference voltage would have to be chosen in such a way that it does not exceed the expected voltage levels with hysteresis present. However as seen in Figure 29c, if the structure also has offset, the measured output levels at 0V differential input could also be high even without hysteresis. Therefore the reference voltage has to be chosen from a limited range of voltages, so that it is greater than the lower boundary determined by expected voltage levels due to largest expected offset and high gain only, and lower than the upper boundary that is determined by expected voltage levels due to hysteresis. The boundaries are shown in Figure 32.
B. Hysteresis Detection – Forward-Backward Sweep Method

Both positive and negative outputs can be connected to a comparator as seen in Figure 33. If a rough input voltage sweep is conducted by sweeping from high (~VDD) to low (~GND), and back to high voltage, the comparator output will be different during the forward and backward sweep at the same input voltage if hysteresis is present. If the output of the comparator can be also recorded at a point where $V_{\text{TEST}} = V_{\text{CM}}$ and compared once both sweeps have finished the hysteresis could be easily detected.

While the first approach may be faster, as there wouldn’t be a need to waste time conducting two input sweeps, finding a correct value for the reference voltage may require additional design time and may vary from technology to technology.

Figure 32 - Output Reference Voltage Boundaries

Figure 33 - Comparator Based Testing Setup
Input test voltages can also be applied in different ways:

A. Input Scheme – Common Mode

If “Output vs. Reference Method” is employed for hysteresis detection, which doesn’t require a test voltage to be toggled between high and low, it may be possible to simplify the testing procedure by constantly holding both inputs at the same ideal common mode voltage. Using common mode voltage for testing will not require a reference voltage ladder at the inputs. On the other hand mismatch by its nature is unpredictable, and may cause the Nauta structure to have offset. And in extreme cases of very large offset and no way to adjust the input for different offsets, it may be hard to guarantee the reliability of the tuning procedure.

B. Input Scheme – Differential

A standard approach for testing differential amplifiers: one input can be held constant at $V_{CM}$ while the second input is connected to $V_{TEST}$ and is varied. Figure 33 shows a test circuit implementing this. As mentioned before, the Nauta structure is expected to have some offset. To detect hysteresis with offset more test voltages around common mode may be required. Therefore to achieve very small differential input voltages a very precise reference voltage ladder will need to be implemented (where in its most basic form it can be a simple resistor ladder as seen in Figure 34). This may be a more reliable testing approach but will be more difficult to implement.

![Resistor Ladder](image)

As discussed before, the mismatch can not only cause a drop in gain but cause other non-desirable behaviours in the Nauta structure such as an offset. Having an offset could complicate the hysteresis detection problem and should ideally be corrected or avoided if possible. Offset is likely to be caused by PMOS and NMOS device imbalance (deviation from ideal 2:1 ratio for the discussed 65nm/180nm processes) where the average PMOS/NMOS imbalance of FF (INV 1), SC (INV 5) and CC (INV 6) inverters of the first input path is different to the average imbalance of FF (INV 2), SC (INV 4) and CC (INV 3) inverters of the second input path. A number of different approaches can be considered to deal with this issue:
A. PMOS/NMOS Imbalance Correction – Characterisation Method

One way of avoiding this could be by identifying the DTC array inverters with high PMOS/NMOS imbalance and not using them during tuning. Individual array inverters in DTC can be isolated by disabling everything else other than the inverter being tested. With a perfectly balanced inverter, if an ideal common voltage is applied at the inputs the same voltage will be seen at the outputs. Any deviation from the ideal common mode value would indicate an imbalance between PMOS and NMOS transistors in the inverter. As a result it may be possible to exclude the inverters with the highest imbalance from being used for tuning. However, testing and characterising every inverter in the Nauta structure may take a significant amount of time to complete.

B. PMOS/NMOS Imbalance Correction – Direct Ratio Adjustment Method

Alternatively the PMOS/NMOS imbalance can be corrected by enabling or disabling individual transistors rather than whole inverters, and hence adjusting and correcting the PMOS/NMOS ratio. This is especially easy to achieve with tri-state inverters by applying different control signals to the NMOS and PMOS control transistors, an alternative switching mechanism would be required for an unmodified Nauta structure to achieve the same effect. This method could be used in conjunction with “Forward-Backward Sweep Method” used for hysteresis detection. A comparator connected to the outputs of the Nauta structure could be used to determine whether there is positive or negative offset with respect to the input voltage. This works as the output of the comparator would change depending on whether the voltage at the positive or negative output node of the Nauta structure is greater, which changes depending on whether a negative or positive differential input is applied. This method may allow the offset to be varied but it doesn't guarantee that perfect PMOS/NMOS balance will be achieved since changing PMOS/NMOS ratio in one inverter may be impossible to differentiate from doing so in the others.

3.5 Conclusion

The methods described above will be combined into 2 main tuning approaches summarised in Table 16 and discussed further in the subsequent Ch. 4. The output vs. reference hysteresis detection method in conjunction with common mode input scheme and characterisation method for PMOS/NMOS imbalance correction will be investigated using the 180nm prototype. Forward-backward sweep method for hysteresis detection in conjunction with differential input scheme and direct ratio adjustment for
PMOS/NMOS imbalance correction will be investigated using the 65nm prototype.

<table>
<thead>
<tr>
<th></th>
<th>Tuning Approach 1</th>
<th>Tuning Approach 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hysteresis Detection</strong></td>
<td>Output vs. Reference Method</td>
<td>Forward-Backward Sweep Method</td>
</tr>
<tr>
<td><strong>Input Scheme</strong></td>
<td>Common Mode</td>
<td>Differential</td>
</tr>
<tr>
<td><strong>PMOS/NMOS Imbalance Correction</strong></td>
<td>Characterisation Method</td>
<td>Direct Ratio Adjustment Method</td>
</tr>
<tr>
<td><strong>Advantages</strong></td>
<td>• Requires a lower number of test voltages</td>
<td>• Better PMOS/NMOS balance can be achieved</td>
</tr>
<tr>
<td></td>
<td>• Characterisation is time consuming</td>
<td>• Does not require extensive characterisation tests of the whole structure</td>
</tr>
<tr>
<td></td>
<td>• Characterisation doesn’t guarantee that PMOS/NMOS balance can be achieved</td>
<td>• Requires a greater amount of test voltages</td>
</tr>
<tr>
<td></td>
<td>• Choosing a suitable reference voltage for output testing is difficult</td>
<td>• Requires tight spacing of test voltages</td>
</tr>
</tbody>
</table>

Table 16 - Comparison of Tuning Approaches
4. Testing, Measurements and Implementation

4.1 Introduction

This chapter will cover the simulation and testing procedures that were used to evaluate a number of possible tuning approaches (Ch. 3.4) and ultimately examine the feasibility of implementing the best procedure on chip in greater detail. Two chips will be used for testing - 180nm and 65nm versions described in Ch. 2.6.1 and 2.6.2.

4.2 Naming Convention

In the following sections a number of naming conventions will be followed. Firstly the Nauta structure will be split into 2 parts: top and bottom. Each part encompasses inverters connected to the same output node as seen in Figure 35. Each inverter in the top half will be annotated with “1” (FF1, SC1, CC1) and inverters in the bottom half will be annotated with “2” (FF2, SC2, CC2). If inverters are referred to without any annotation (FF, SC, CC), it means that it is referred to both inverters (FF1 and FF2, SC1 and SC2 or CC1 and CC2) simultaneously.

4.3 Modified Nauta Structure - 180nm Prototype

![Figure 35 - Naming Convention of the Nauta Structure](image)

4.3.1 Introduction

The first prototype examined for this work was a modified Nauta structure implemented in Silterra 180nm process. The structure was based on the Digital to Transconductance converters described previously (in Ch. 2.6.1). As this was an early prototype there were some drawbacks and implementation strategies that did not result in improved performance of the prototype. This section will examine the structure of the 180nm prototype and its shortcomings in more detail, as well as a rudimentary tuning procedure that was examined initially, but ultimately was found to be not feasible.
4.3.2 PMOS/NMOS Ratio Imbalance

As mismatch is a primary issue for the Nauta structure in scaled down technologies, it was also an issue for a Nauta structure implemented in 180nm technology. Obtaining a high gain is prevented by mismatch between SC and CC array elements. However imbalance should also be expected between NMOS and PMOS transistors. If due to mismatch, the average PMOS and NMOS ratios of the inverters in the top and bottom half of the Nauta structure differ, differential and common mode offsets can be expected, and this can make it significantly more difficult to make open loop gain measurements. As a result the viability of running characterisation tests before tuning the Nauta structure was examined. Characterisation is one of two PMOS/NMOS Imbalance Correction methods proposed in Ch. 3.4, the other – direct PMOS/NMOS ratio adjustment method will be investigated in subsequent chapters.

Different parts of the structure can be isolated to observe the DC levels at each of the output nodes while sweeping through a wide range of codes for each inverter. This way an estimate of PMOS and NMOS imbalance in each inverter can be found. The minimum number of inverters that need to be enabled to test FF, SC and CC inverters with as little influence of other inverters in the structure as possible are highlighted in Figure 36.

![Figure 36 - Enabled Inverters for Characterisation Sweeps](image)

Firstly, everything but the SC inverters – SC1/SC2, are disabled, as shown in Figure 36a. The control codes of SC inverters are then swept from lowest to highest. The output values at large SC codes can be used to approximate the ideal common mode voltage. In effect, this is the same method used by Nauta to generate a common mode voltage at the input [19]. Secondly, all SC and CC inverters are disabled, while keeping FF inverters FF1/FF2 on, as seen in Figure 36b, and a test DC input voltage is set to the value approximated by the SC sweep. By sweeping through the FF codes (from lowest to highest) it is
possible to estimate the PMOS/NMOS imbalance in individual inverters of the FF arrays by comparing the output DC voltage to the input. If the difference between input and output voltage is small, it indicates that the PMOS/NMOS imbalance is small or is similar to the one in SC inverters, if the difference between input and output is large, it indicates that the imbalance is large. To test CC inverters, a SC inverter at the input of the particular CC inverter needs to be used as a constant voltage source. For example SC inverter SC2 can be used as a source of input voltage for CC inverter CC1, as seen in Figure 36c. SC2 code would be kept constant, while codes for CC1 would be swept through a range of values. Similarly, the test will allow the CC inverters with large PMOS/NMOS imbalance to be identified and possibly excluded from being used during tuning. Depending on the number of unique codes it may be more feasible to observe outputs of individual inverters only, rather than all possible combinations. To extrapolate the DC output for any given code, the size of the biggest enabled inverter as well as its output voltage level relative to any additional enabled inverters needs to be taken into account.

Similarly, it is possible to disable FF inverters and only keep SC and CC inverters on, as seen in Figure 36d. The purpose of this configuration is to observe the behaviour of the structure without any external influences and FF inverters. Although this particular test doesn't deal with PMOS/NMOS imbalance, it provides useful information about the mismatch of SC and CC inverters. Considering that this test could be implemented using the same circuits as the tests dealing with PMOS/NMOS imbalance it would be inefficient to not conduct SC/CC characterization. For this setup the number of possible code combinations becomes significantly larger, and to minimise the number of tests the codes of all inverters are set to the same value throughout the sweep and therefore equivalent width/transconductance. Alternatively codes of CC inverters can be set to always be smaller than codes for SC inverters by 1 or more, as codes for SC inverters are swept from lowest to highest values. The results of such tests are a good estimate for the true mismatch between the CC and SC inverters. High DC voltage difference between positive and negative outputs indicates that CC inverters are bigger than the SC ones, as it is likely to be caused by hysteresis. It is important to know the code ranges where sudden changes in mismatch occur. If for example, during a sweep, there is a sudden change from 0V difference between outputs to a large difference (while the SC and CC codes remain equal), it may indicate that there is a missing range of output transconductance values for the particular digital to transconductance converter, as it means that the mismatch between the SC and CC inverters increased due to a sudden change in width of either SC or CC inverter compared to the expected widths. Such deviations from expected widths may make it impossible to tune the Nauta structure for gain using that particular range of codes.
4.3.3 Case Study - 180nm Prototype Characterization

To gain a better understanding of the characterization tests described in the previous section, experiments with a fabricated Nauta structure were performed. Since the purpose of characterisation is to identify and minimise the effects of mismatch and process variation, by conducting the experiments on a real die, with real mismatch, allowed a more accurate evaluation of the usefulness of characterization procedures to be conducted. The experiments were primarily run on the same die, however some of the experiments were repeated on a secondary test die to examine and demonstrate the variability between different dies. Firstly the SC characterization test was run as seen in Figure 37. Due to the control signal design it wasn't possible to test a full range of codes as the largest control bit is shared with the largest CC inverter. As can be seen, the results demonstrate that there is significant amount of variability especially at lower codes corresponding to small elements in the inverter array like quarter, half and unit sized inverters. We can also see that there is a noticeable difference between the two SC inverters in the same Nauta structure. From this we can conclude that it will be more difficult to tune the structure for high gain reliably using lower code ranges. For low codes, a difference in output voltages of the SC inverters in some cases is measured to be over 5mV, this demonstrates that the PMOS/NMOS ratio of the two inverters is different and that in turn will introduce offset. If such codes can’t be avoided it will be necessary to adjust input test signals every time the control codes change and could complicate and make the tuning procedure more time consuming. On the other hand, at larger codes we can see that the output DC voltage generated by SC inverters averages out to some extent to around 830mV (consistent with $\sigma_{\text{mismatch}} = 1/\sqrt{(WL)}$ relationship [29], where mismatch is expected to reduce as area is increased). As a result, using higher control codes could minimise the number of adjustments that would need to be made to the input test signals to obtain reliable gain measurements, or ideally remove the need to make adjustments completely.

![Figure 37 - SC Characterization Sweep Measurements](image-url)
For the FF characterization test the input DC voltage was set to 830mV, as was found by SC characterization tests. The results of the FF characterization sweep can be seen in Figure 38. In the figure we can see that sudden changes in DC level can be seen at a point where control code transitions from a larger number of smaller array elements being active to a single larger element being active.

Figure 38 - FF Characterization Sweep Measurements (V\textsubscript{IN}=830mV)

Figure 39 - FF Characterization Sweep Measurements (Supplementary Die, V\textsubscript{IN}=830mV)
This experiment was also repeated on a second die where similar trends can be seen in Figure 39. There are also significant differences between the DC levels of the first and second die, especially at lower control code settings. This once again reinforces that significant intra-die variation should be expected. Additionally an upward trend in output voltages can be seen in both sweeps. This is likely caused by different rates of change in \( g_m \) to \( g_{ds} \) ratios in PMOS and NMOS transistors as the transistor sizes increase.

![Graph showing output voltage vs CC Code](image)

Figure 40 - CC Characterization Sweep Measurements (\( V_{IN} \approx 830\text{mV} \), as set by CC1 at code 63)

Figure 40 shows the results of the CC characterization test, once again due to some of the control bits being shared the number of control codes available for the sweep was limited. But it can still be seen that for control codes greater than 25 the output voltage and the PMOS/NMOS imbalance is averaged out to a large extent. This is likely due to the fact that larger inverters are comprised of identically sized unit inverters and as the number of switched on unit inverters is increased the PMOS/NMOS imbalance is reduced as it approaches an average value.

Overall, it can said that PMOS/NMOS characterisation tests can provide a lot of valuable information such as identifying codes where large changes in PMOS/NMOS imbalance occur and how large the codes need to be before the PMOS/NMOS imbalance begins to average out and decrease due to increases in the overall size of the inverters. However, it must be noted that characterising SC inverters is harder as their outputs are connected to the inputs and track the ideal common mode voltage that varies over a smaller range, compared to output voltages of FF and CC inverters that are connected to a constant input voltage and whose output voltage will vary over a greater range as any difference between the input voltage and ideal common mode voltage will be amplified. As a result, on-chip this would require a more precise circuit (such as an ADC) to conduct the characterisation.
Figure 41 shows the results of SCCC characterization test, the SC and CC codes were set to the same value throughout the sweep. From the large difference between output DC levels it can be seen that for most of those control codes effective CC inverter sizes may already exceed the sizes of SC inverters as discussed in Ch. 4.3.2. Large difference (of approximately 100mV or more) between the outputs implies that the hysteresis is present and for hysteresis to be present the transconductance of CC inverters must be greater than the transconductance, and therefore width, of SC inverters (as discussed in Ch. 2.5.1). The experiment was repeated on a supplementary test die with similar results, as seen in Figure 42. Certain amount of mismatch due to process variation is expected, and these results confirm the presence of mismatch.

Figure 42 - SCCC Characterization Test Measurements (0 Offset, Supplementary Die)

However the same experiment was repeated where the cross-coupled control code values lagged by 1 (SC Code = CC Code + 1), such that the settings for
control codes were CC=1/SC=2, CC=2/SC=3 etc. The results can be seen in Figure 43. The large difference between outputs is still seen for large ranges of codes. Similarly, the same experiment was repeated where the CC control codes lagged by 2 (SC Code = CC Code - 2), such that the settings for control codes were CC=1/SC=3, CC=2/SC=4 etc. This can be seen in Figure 44.

A number of observations can be made from the results above (Figure 41-Figure 43). Firstly we can see sudden reduction in voltage difference at certain code transitions, especially at transition from 63 to 64 (0111111 to 1000000). Which implies that there is a significant change in the width difference between the SC and CC inverters caused by mismatch and as such implies that the Nauta structure will have a missing range of output conductance values. Therefore it would be beneficial to avoid that transition during gain tuning sweeps. If such a code transition is encountered during gain tuning sweep, the range of the sweep and the duration of sweep will increase as alternative codes will need to be found for the missing output conductance range. This is not desirable for on-chip applications, where tuning must be done as quickly as possible. Additionally the output voltages appear to “oscillate”, where the output voltages are “swapped” for the positive and negative output terminals for certain code ranges. This behaviour is consistent with the presence of hysteresis since a large difference between the outputs is maintained regardless of changes in the magnitudes of individual outputs, and is likely to be caused by unpredictable voltage variations during reprogramming of the structure during code changes.

The “lag by 2” sweep was conducted while disabling one of the “×1Unit” inverters. Given that there are 2 “×1Unit” sized inverters control codes 1000

Figure 43 - SCCC Characterization Test Measurements (+1 Offset)
and 0100 correspond to the same width. Therefore cases where a larger code like 1000 (8) corresponding to a smaller width of 1 unit, compared to a smaller code 0110 (6) corresponding to a larger width of 1.5 units, would occur. To avoid that, the sweep was conducted where all codes using one of the “×1Unit” inverters corresponding to the 2nd control bit were removed, such that the codes used in the sweep where: 1,2,3, 8,9,10,11, 16,17,18,19, 24,25,26,27, 32,33,34,35 etc.

Here we start to see significant ranges of codes that result in little to no DC level variation. Overall however, the fact that the structure becomes unstable even when the control codes for SC and CC inverters are roughly the same, suggests that the conductance contributed to the calculation of the output gain by all the inverters is easily overcome by the variation of output transconductances of individual inverters. This is important to consider when choosing a CC sweep range values for gain tuning, and may require choosing a wider range, so that the codes that may have high gain aren’t left out of the sweep.

![Figure 44 - SCCC Characterization Test Measurements (+2 Offset, Bit 2 Disabled)](image)

Figure 44 - SCCC Characterization Test Measurements (+2 Offset, Bit 2 Disabled)

To recap, by analysing the results of the code sweep tests it is possible to identify the best usable ranges of codes that need to be used to tune the structure for maximum gain while minimising the effects of NMOS/PMOS imbalance and mismatch. The results of the SCC test can be used by observing the separation between the positive and negative output nodes. Sudden changes from low separation to higher separation indicates that for certain codes the effects of mismatch are more severe and that ideally a different range of CC and SC should be used for gain tuning. When it comes to CC and FF characterisation, only ranges of codes whose outputs did not
deviate from ideal common mode levels during the characterisation sweeps should be used. Overall, a lot of useful information can be extracted using characterisation tests, however the number of tests needed to characterise any given structure may be too high to be completed in a reasonable amount of time and there is no guarantee that a suitable range of codes can be found on every chip, even at code ranges corresponding to large inverter widths where the effects of mismatch and process variation appear to be less severe.
4.3.4 Case Study - 180nm Prototype Gain Tuning

Initially, in an effort to keep the tuning procedure as simple as possible, it was attempted to observe the output DC levels with common mode voltage at the input to identify codes with high gain (common mode input scheme and output vs. reference hysteresis detection method - first discussed in Ch 3.4). Since both inputs would be connected to the same voltage, this method would likely require a lower number (ideally only one) of tests voltages and the reference voltage circuits could possibly be implemented using a simpler topology as they wouldn’t need to be as precise, compared to for example having to implement a voltage reference for a sub-mV differential input needed for the differential input scheme. This sub-chapter will examine the use of the common mode input scheme and output vs. reference hysteresis detection in conjunction with characterisation procedure for PMOS/NMOS imbalance correction (described in Ch. 3.4 and 4.3.2), where the results of characterisation obtained in Ch. 4.3.3 will be used to tune the prototype for high gain.

During the common mode sweep (both inputs are held at the same voltage throughout the sweep) high gain condition is detected by detecting significant change in DC levels at the outputs. If the structure is perfectly balanced and there is no mismatch, the DC levels of both outputs will remain the same during a common mode sweep for all input values. However, if any mismatch was introduced during simulations (that almost certainly will be present in fabricated dies), a common mode gain was observed. If the effective width of CC inverters was increased past what is required for maximum gain, hysteresis could be observed resulting in large DC output voltage shifts if measured at the ideal common mode input voltage (of \( V_{DD}/2 \)).

![Figure 45 - Simulation - Effects of Imbalance and Common Mode Gain on the Output DC Levels, \( \Delta W_{CC} = \text{PMOS-1\mu m/ NMOS-500nm (≈3.3% of the Total Width)} \)](image-url)
Figure 45 shows the general behaviour of the Nauta structure during a common mode sweep when: it is balanced at high gain, it has a small amount of mismatch at high gain, and the structure has mismatch, while the maximum gain condition is exceeded. It can be seen that to detect hysteresis the outputs need to be observed at the ideal input common mode voltage, where even with a minor mismatch the output levels are likely to be close to ideal common mode voltage with no (or very little) common mode gain if there is no hysteresis. And a significant change will be observed when hysteresis occurs. However, that may not be the case if the mismatch causes significant PMOS/NMOS imbalance that will in turn result in common mode gain at a common mode input voltage that can possibly be confused with output DC level changes due to hysteresis. And that is why it is critical to conduct the characterization procedure for PMOS/NMOS imbalance correction (Ch. 3.4 and 4.3.2) to avoid common mode gain when input voltage is set to ideal common mode levels.

The tuning process itself can be best described as variation of the localised sweep tests described in Ch. 4.3.2. Suitable SC and CC inverters code ranges should be chosen such that there are at least several codes around the median codes that are bigger and smaller in size, and have a similar NMOS/PMOS imbalance. The size and code of FF inverters is then chosen in the same way, but based on the results of the FF characterisation sweep. During testing it was found that for this particular version of the prototype, inverter size difference corresponding to a difference of 4 CC codes was sufficiently greater than a possible error in size caused by mismatch. The CC values can therefore be swept within a maximum range of +/- 4 of the chosen SC code, if relatively small FF codes are used. The median CC code will need to be greater by up to

![Figure 46 - Gain Tuning Measurements (V\textsubscript{INDC} = 830mV, V\textsubscript{INPP} = 1.12mV, f=10kHz)](image_url)
several codes if particularly large FF codes are chosen. During the tuning sweep, as soon as a significant deviation of DC levels is detected the code is rolled back by one.

An example of this can be seen in Figure 46, where a SC code is set to a constant value of 65 and CC codes are swept from 55 to 76. The codes were chosen by manually pre-tuning the structure into a state close to high gain. The initial ranges of codes for the manual sweep were chosen based on the characterisation test results. This resulted in the use of FF codes from 129 to 160 that had relatively low deviation of PMOS/NMOS imbalance between FF1 and FF2 inverters, as seen in Figure 38. Similarly, the SC codes were chosen to be greater than 60, where PMOS/NMOS imbalance was also found to be relatively low, as seen in Figure 37. A CC sweep was chosen to have a wider range to demonstrate a number of different behaviours of the Nauta structure. At CC code 70 a significant change in the output DC levels and a drop in output peak to peak voltage can be seen. The CC code is then rolled back to 69 where there is little change in output DC levels and high peak to peak voltage. The procedure can then be repeated with FF codes. A preliminary tuning sweep with a larger range such as +/- 20 of an initial, rough FF code choice, and with larger increments of 4 codes is used to get the FF code closer to the ideal value. FF tuning is required because the gain is a lot less sensitive to changes in FF codes that contribute a conductance value rather than transconductance to the denominator of the gain equation (Eq. 7b), which maybe be up to 20 times smaller for 180nm technology. Finally, FF values are finetuned by sweeping codes with an increment of 1 code starting with the code generated by the preliminary FF sweep. For the examined prototype, it was possible to achieve open loop gains of 38-40dB at a test frequency of 10kHz. Overall, in most cases the tuning procedure should not exceed 26 measurement steps - 8 for CC tuning sweep, 10 for preliminary FF tuning sweep and 8 for FF finetuning. This number excludes measurements required for characterization. A minimum of 1 measurement step for every independent FF, SC and CC inverter will be required. The SC-CC characterisation sweep can be limited to a smaller range, but has to at the very least cover SC and CC code ranges used for gain tuning. Due to the nature of the architecture of the prototype certain codes correspond to equivalent effective array sizes. For demonstration purposes Figure 46 contains such codes. Ideally the repeating codes would be excluded from the tuning procedure based on the characterisation sweeps, where the blocks with more NMOS/PMOS imbalance would be excluded. In Figure 46 two codes 69 and 73 are equivalent in size with the only difference being that a different ×1 Unit block is used for each code. However, code 69 produces a higher gain with less variation in DC levels, highlighting the sensitivity of the structure to mismatch between different components. Additionally a range of codes with visibly large difference between the outputs consistent with large hysteresis can be seen. This occurs for codes under 63 and completely disappears after
code 64 (transition from 111111 to 1000000). This indicates that there is a large mismatch between a single large inverter and a larger number of smaller ones. Such transition points can be identified by SC-CC characterizations tests and should ideally be avoided during gain tuning sweeps.

In conclusion, it can be said that characterisation procedures (proposed in Ch. 3.4 and 4.3.2 and investigated in 4.3.3) were definitely of use in choosing the range of codes for gain tuning and in turn allowed the structure to be tuned for relatively high gain. It was possible to avoid ranges of codes with noticeably large mismatch and PMOS/NMOS imbalance. However, during gain tuning it was also shown that that the effects of process variation can't be avoided entirely as they are present to a lesser extent throughout all of the codes (such as the difference in measurements at equivalent CC codes of 69 and 73). As a result, the feasibility of using the characterisation along with the common mode input scheme and output vs. reference hysteresis detection method will be discussed further in the following Ch. 4.3.5.

4.3.5 Feasibility of Implementation

The calibration procedure described above worked relatively well enough to tune the gain in laboratory conditions. However if such a strategy was to be implemented as an on-chip solution, it is likely that it would not be sufficiently reliable for a consumer level product for a number of reasons. Firstly, it is likely that a significant amount of additional circuitry will be required to implement the tuning algorithm. Secondly, another issue with the proposed tuning procedure is the reliance on the characterization to deal with

![Figure 47 - Simulation - Common Mode Sweeps - P/N Imbalance](image-url)
PMOS/NMOS imbalance. By definition, it can only be used to characterise and help identify the PMOS/NMOS imbalance in the inverters, but not correct it. Ultimately, there is no guarantee that any given die will have a suitable range of codes that can be used to successfully tune the Nauta structure for high DC gain. If the characterization procedure is unsuccessful the gain tuning sweep could easily produce a false positive result.

In Figure 47 the structure was simulated and set to a code of FF=28, SC=51, CC=58 with effective PMOS widths being 16µm, 27µm, and 30µm respectively. Following that, common mode input sweeps were conducted showing both outputs individually. A difference in effective width of the two CC inverter arrays was introduced, while keeping PMOS/NMOS ratio at 2. The difference in PMOS/NMOS inverters was set to 2060nm/1030nm, following that a PMOS/NMOS imbalance was introduced to one of the CC inverters only by increasing the width of the NMOS transistor in steps of 50nm 3 times. If the output voltages are observed while inputs are set to the ideal common mode voltage of the setting with no PMOS/NMOS imbalance, a noticeable change of 44mV (for +ve output) and 47mv (for –ve output) would be seen for the case with 150nm mismatch between FF1 and FF2 NMOS transistors, as highlighted in Table 17. This would occur even though there is no hysteresis present and the structure is not at maximum gain.

| Width Difference Increase Between FF1 NMOS and FF2 NMOS Transistors (nm) |
|-----------------------------|-------------|-------------|-------------|
|                            | 0          | 50          | 100         | 150         |
| $V_{out}+(mV)$ @ Vin=876mV | 893        | 906         | 921         | 937         |
| $V_{out}+(mV)$ @ Vin=900mV | 874        | 887         | 901         | 916         |
| $V_{out}+(mV)$ @ Vin=924mV | 856        | 868         | 881         | 895         |
| $V_{out}+(mV)$ @ Vin=948mV | 838        | 849         | 861         | 874         |
| $V_{out}-(mV)$ @ Vin=876mV | 894        | 880         | 864         | 847         |
| $V_{out}-(mV)$ @ Vin=900mV | 900        | 886         | 872         | 855         |
| $V_{out}-(mV)$ @ Vin=924mV | 905        | 893         | 880         | 864         |
| $V_{out}-(mV)$ @ Vin=948mV | 910        | 899         | 887         | 873         |

Table 17 - Change in Output Voltage Due to Mismatch Between FF NMOS Transistors (FF1 & FF2) Width at Various Offsets

<table>
<thead>
<tr>
<th>Width Reduction of FF1 Inverter – NMOS (nm)/PMOS (nm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0/0</td>
</tr>
<tr>
<td>$V_{out}+(mV)$ @ Vin=828mV</td>
</tr>
<tr>
<td>$V_{out}+(mV)$ @ Vin=852mV</td>
</tr>
<tr>
<td>$V_{out}+(mV)$ @ Vin=876mV</td>
</tr>
<tr>
<td>$V_{out}-(mV)$ @ Vin=891mV</td>
</tr>
<tr>
<td>$V_{out}-(mV)$ @ Vin=900mV</td>
</tr>
<tr>
<td>$V_{out}-(mV)$ @ Vin=828mV</td>
</tr>
<tr>
<td>$V_{out}-(mV)$ @ Vin=852mV</td>
</tr>
<tr>
<td>$V_{out}-(mV)$ @ Vin=876mV</td>
</tr>
<tr>
<td>$V_{out}-(mV)$ @ Vin=891mV</td>
</tr>
<tr>
<td>$V_{out}-(mV)$ @ Vin=900mV</td>
</tr>
</tbody>
</table>

Table 18 - Change in Output Voltage Due to Mismatch Between FF Inverters (FF1 & FF2) Width at Various Offsets
Furthermore, it is possible to achieve similar common mode response/gain with even less mismatch. Figure 48 shows a common mode sweep response of a Nauta structure set with a control code of FF=28, SC=51, CC=58 as before. However, this time the mismatch was introduced in FF inverters. Starting with a balanced structure, firstly the size of the FF inverter was reduced in steps of NMOS=50nm/PMOS=100nm such that FF1 inverter is larger than FF2 inverter (Figure 48 - Top). Secondly the same inverter was then increased in steps of the same size such that FF2 inverter became larger than FF1 (Figure 48 - Bottom). The maximum mismatch in Figure 48 is NMOS=250nm and PMOS=500nm as opposed to NMOS=1030nm and PMOS=2060nm in previous example. Even accounting for the fact that for that particular setting FF inverters are half as small as SC and CC inverters, we can still see that a mismatch of approximately 3% (of $W_{\text{PMOS}}=16\mu\text{m}$ at code 28) causes significant common mode gain that could be easily confused with the presence of
hysteresis, even with only minor PMOS/NMOS imbalance, if only the DC levels are observed. This is highlighted in Table 18, where at an offset of 17mV ($V_{in}=876mV$) with mismatch of 250/500nm a DC voltage difference of 47mV compared to the output voltage of a balanced structure.

<table>
<thead>
<tr>
<th>Width Increase of CC Inverters (CC1 &amp; CC2) – NMOS (nm)/PMOS (nm)</th>
<th>0/0</th>
<th>50/100</th>
<th>100/200</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{out^+}(mV)$</td>
<td>891</td>
<td>976</td>
<td>1020</td>
</tr>
<tr>
<td>$V_{out^-}(mV)$</td>
<td>895</td>
<td>808</td>
<td>761</td>
</tr>
</tbody>
</table>

Table 19 - Change in Output Voltage Due to Increase in CC Inverter Width at $V_{in}=876mV$ and a Constant CC1/CC2 Mismatch of NMOS=25nm/PMOS=50nm

To further examine this problem we explore more of the possible situations that might result due to mismatch. Once again starting with a code of FF=28, SC=51, CC=58 corresponding to high gain, the CC inverters are then increased in steps of width NMOS=50nm/PMOS=100nm while a constant mismatch of is kept at NMOS=25nm and PMOS=50nm. In Figure 49 and Table 19 we can see that a rather large change in DC levels would be caused by hysteresis of approximately 85mV, during the common mode input sweeps. However, this could be considered a one of the more ideal situations. A less than ideal situation could be seen in Figure 50 and Table 20. In this case the CC inverters are increased in steps of width NMOS=10nm/PMOS=20nm while a constant mismatch is kept at NMOS=5nm and PMOS=10nm. There we could see a hysteresis where a change in voltage levels would only be approximately 30mV. As demonstrated by the example in Figure 47/Figure 48 this could be easily confused with common mode gain. Given the
unpredictable nature of mismatch it is likely that a variety of cases will be encountered during tuning, including cases where the tuning procedure will operate adequately. Nonetheless, it is clear that largely the procedure is far too unreliable for applications outside of laboratory conditions without any further modifications.

Table 20 - Change in Output Voltage Due to Increase in CC Inverter Width at \( V_{\text{in}}=876\text{mV} \) and a Constant CC1/CC2 Mismatch of NMOS=5nm/PMOS=10nm

<table>
<thead>
<tr>
<th>Width Increase of CC Inverters (CC1 &amp; CC2) – NMOS (nm)/PMOS (nm)</th>
<th>( V_{\text{out}+}(\text{mV}) )</th>
<th>( V_{\text{out}-(\text{mV})} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0/0</td>
<td>892</td>
<td>937</td>
</tr>
<tr>
<td>10/20</td>
<td>922</td>
<td>947</td>
</tr>
<tr>
<td>20/40</td>
<td>947</td>
<td>956</td>
</tr>
<tr>
<td>30/60</td>
<td>963</td>
<td>970</td>
</tr>
<tr>
<td>40/80</td>
<td>976</td>
<td>982</td>
</tr>
<tr>
<td>50/100</td>
<td>987</td>
<td>992</td>
</tr>
<tr>
<td>60/120</td>
<td>802</td>
<td>797</td>
</tr>
<tr>
<td>70/140</td>
<td>808</td>
<td>792</td>
</tr>
<tr>
<td>80/160</td>
<td></td>
<td></td>
</tr>
<tr>
<td>90/180</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100/200</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 50 - Simulation - Increase in CC (Small Steps, Small Mismatch)

Even if it was possible to choose a reference voltage that would guarantee a statistically low chance of confusing the common mode gain and hysteresis, the feasibility of implementing and running a characterization procedure on-chip needs to be considered. Measuring the DC levels using an oscilloscope paints a rather clear picture of PMOS/NMOS imbalance, replicating the same on-chip would require an ADC. Characterizing FF and CC inverters might be feasible with a low precision converter, however SC inverters will always operate at their ideal common mode voltage, given that in Figure 37 the range of common mode voltages was approximately 8mV, a high precision ADC will be required. This approach might still be applicable in the situation where a large number of structures on the same chip will need to be calibrated, but it is unlikely to be universally feasible, especially were a small number of structures will need to be calibrated. Alternatively, instead of trying to record the absolute value of the output voltage, a comparison of the output and a common mode reference voltage could be made (or a number of comparisons...
using multiple reference voltages close to common mode voltage) to get an approximate estimate of the PMOS/NMOS imbalance. Once again though, the precision of such procedure is likely to be far too low, and the procedure itself would take a lot of time run, to make characterization of any use.

4.3.6 Conclusion

A 180nm prototype of the digital Nauta structure was tested in laboratory conditions and simulated. A calibration procedure involving characterization and gain tuning steps was used to calibrate the Nauta structure for high gain. The testing revealed that the performance of the prototype was limited by the inverter array sizing combined with mismatch due to process variation. The simulations also revealed that the proposed tuning procedure may easily fail under certain mismatch conditions. Additionally trying to implement the tuning procedure on-chip would likely result in increased complexity, power and area requirements which would outweigh the gain and advantages of using the Nauta structure over more traditional amplifiers. As a result, the tuning procedure was investigated and refined further using the strategies such as differential input scheme and backward-forward sweep for hysteresis detection discussed in Ch. 3.4.
4.4 Modified Nauta Structure - 65nm Prototype

4.4.1 Introduction

A prototype of the Nauta structure fabricated in 65nm became available and was used for further testing and investigations in this chapter. The 65nm prototype has advantages over the 180nm prototype such as the superior inverter sizing scheme that allowed the structure to be tuned with higher precision. The design of the prototype was described in great detail in [28] and summarised in Ch. 2.6.2. The 65nm prototype was used to test some of the alternative calibration techniques described in Ch. 3.4 - differential input scheme and forward-backward sweep hysteresis detection, where the input voltage is swept from low to high and back to low while the outputs are connected to a comparator that would produce a different result during the forward and backward sweeps if the hysteresis is present. The advantages and disadvantages of the new calibration technique will be examined and the possibility of implementing the new technique on chip will be explored. The new technique no longer attempts to achieve complete PMOS/NMOS balance through characterisation and limiting the use of components with high PMOS/NMOS imbalance as a way of dealing with offset introduced by PMOS/NMOS imbalance. This is done in order to reduce the amount of additional hardware required for calibration, and instead the offset is corrected by adjusting the PMOS/NMOS ratio by using different control codes for PMOS and NMOS portions of the tri-state inverters (direct PMOS/NMOS ratio adjustment method described in Ch 3.4).

Figure 51 - Simulation - Hysteresis During Differential Sweep (CC Codes, No Mismatch)
4.4.2 65nm Prototype - Simulations

Due to the shortcomings of the calibration approach used for 180nm prototype, such as the reliance on characterisation to deal with PMOS/NMOS imbalance, which can be difficult to conduct on-chip and still does not guarantee reliable results (as discussed in Ch. 4.3.5), for the new prototype a different approach was examined. Instead of focusing on the testing with common mode input voltage, the feasibility of using differential input sweeps was investigated. One of the noticeable differences between the two approaches is that the common mode approach uses the same test voltage for both inputs and depends on mismatch to introduce common mode gain at which point it becomes possible to observe the hysteresis. On the other hand a differential sweep requires two test voltages, but the hysteresis will be observed even with perfect matching.

Figure 51 shows the results of the differential sweep simulations. The setup for the simulations was discussed in Ch. 3.4 and can be seen in Figure 33. The Nauta structure was set to the following code: SC = 512 (PMOS W=48.64µm), FF = 196 (PMOS W=23.82µm), CC = 601 (PMOS W=61.16µm) initially, this code was chosen as it sets the Nauta structure to be very close to high gain. This code was found through manual analysis of simulation results for a larger range of CC codes. Therefore, sweeping a small number of codes around the initial setting will demonstrate the transition from high gain to hysteresis. The CC codes were then swept from 601 to 605.

![Figure 52 - Simulation - Hysteresis During Differential Sweep (FF Codes, No Mismatch)](image-url)
The structure remained completely balanced throughout the sweep of all the CC codes. The width of the hysteresis at codes 603, 604 and 605 was measured to be 1.15mV, 1.80mV and 4.36mV respectively. A similar experiment was repeated by sweeping FF codes instead of CC codes (while keeping SC codes constant at 512, and CC at 602). The FF codes were swept starting with 192 and ending with 196. There results can be seen in Figure 52 and here a much narrower hysteresis can be observed at codes 192, 193 and 194. The hysteresis was simulated to be 136µV, 50µV and 10µV respectively. After examining these early simulation results, it started to become clear that a possible issue may arise with detection of extremely narrow hysteresis, and the consequences of that needed to be investigated further.

There would be no issue with detecting the hysteresis if it occurred where it is centred exactly at the point where \( V_{\text{IN}} = V_{\text{IN+}} \). However, that only occurs while the structure is completely balanced and there is no mismatch. If PMOS/NMOS imbalance is introduced the centre point of the hysteresis will also deviate from 0V differential input point. This in turn significantly complicates detection of hysteresis with width in sub mV range. The number of additional test voltages would be determined by the maximum expected hysteresis offset and the minimum hysteresis width that needs to be detected to achieve a high gain condition. With offset in 10s of mV and hysteresis in sub mV range the number of tests voltages could quickly become unreasonable. To understand this issue better, the expected offset and feasibility of correcting this offset to minimise the number of test voltages will be investigated further in the remainder of this thesis.

To better understand the effects of PMOS/NMOS imbalance and whether direct PMOS/NMOS ratio adjustment is a viable option for offset correction we introduce mismatch in a balanced structure by only increasing the control codes of the NMOS transistors in the inverters of the top half (FF1, SC1, and CC1) of the Nauta structure, while keeping control codes for PMOS constant. In Figure 53, starting with codes SC = 512 (PMOS W=48.64µm), FF = 196 (PMOS W=23.82µm), CC = 603 (PMOS W=61.59µm). The codes were then increased to 552 for SC NMOS, 236 for FF NMOS and 639 for CC NMOS (separately). The biggest differences in NMOS widths for those ranges were SC-3.01µm, FF-3.01µm, and CC-2.7µm. The offsets were found to be approximately SC-15.5mV, FF-12mV, and CC-15.75mV.
Figure 53 - Simulation - Differential Sweeps with PMOS/NMOS Imbalance
(Increase NMOS, Top Half Only)
Figure 54 - Simulation - Differential Sweeps with PMOS/NMOS Mismatch (Increase NMOS, Bottom Half Only)
To demonstrate the symmetry of the Nauta structure the same experiment was repeated in Figure 54, with the only difference being that control codes for NMOS were swept for the inverters on the bottom half of the Nauta structure. The offsets seen in Figure 53 are more or less of the same magnitude as the ones seen in Figure 54 with the only exception being that instead of negative offset, a positive offset can be seen.

![Figure 55 - Simulation - Hysteresis and Offset (CC and FF NMOS Sweeps)](image)

To get a better understanding of how the hysteresis and offset change on a code to code basis, in Figure 55 each DC sweep is represented by a line plot, with end points corresponding to the hysteresis boundaries. Furthermore, a 100 run Monte-Carlo simulation (for SC = 512, FF = 196, CC = 603 settings) is shown in Figure 56 which provides an estimate of the range of offset values to be expected.
Overall, from the Monte-Carlo results in Figure 56 it can be seen that for the particular set of control codes the expected offset falls within -15mV to 15 mV. Similar offset can also be achieved by introducing PMOS/NMOS imbalance of around 40 codes in any of the inverters. Therefore, it stands to reason that it may be viable to correct the offset caused by mismatch by compensating for physical PMOS/NMOS imbalance in transistors by adjusting the ratio of PMOS and NMOS control codes.

Figure 56 - 100 Run Monte-Carlo Simulation

Figure 57 - Simulation - CC Sweep (One Side Only, P/N Balanced)
In contrast a CC sweep can be repeated while keeping PMOS/NMOS ratio constant, but still sweeping the width of only one of the inverters. Figure 57 shows a sweep of the width of the top (CC1) inverter. Starting with a balanced structure the width of one of the CC inverters was increased by up to NMOS = 3.2µm/PMOS = 6.4µm. What can be seen is that the offset remains more or less constant throughout the sweep.

From this it can be concluded that for a calibration method that depends on a differential sweep to detect hysteresis to work it will be required that inter-channel PMOS/NMOS balance is achieved between the top and the bottom halves (where the average PMOS/NMOS imbalance of FF1, SC1 and CC1 is equivalent to the average imbalance of FF2, SC2 and CC2) of the Nauta structure. On the other hand balancing the mismatch between the FF, SC and CC inverter pairs can be considered a secondary objective, since it doesn’t have a significant effect on the hysteresis offset (<1mV offset for a change of 3.2µm/6.4µm (NMOS/PMOS)). That said it is still important to keep in mind that excessive mismatch will cause high common mode gain. Especially given that adjusting PMOS/NMOS ratio in any inverter in the same path, regardless whether it is FF, SC or CC will have the same effect on the offset and it might be possible to introduce more mismatch than necessary by only varying PMOS/NMOS ratio in only one of the inverters. A possible way to minimise this is by adjusting PMOS/NMOS ratio in all 6 inverters by smaller amounts, instead of introducing one large change in one inverter.

Figure 58 - Simulation - Differential DC Sweeps, No Offset a) Balanced, Matched b) Balanced, Mismatched
Figure 58a shows a differential sweep for a balanced (no PMOS/NMOS imbalance) and perfectly matched (no mismatch between inverters in the top and bottom halves of the structure). Figure 58b shows a differential sweep for a structure where individual inverters have a PMOS/NMOS imbalance, but on average cancel out to have 0V differential offset. Figure 59 shows the common mode sweep results for the same cases (dotted line for a perfectly balanced structure). Figure 59 shows that the unbalanced structure has a significant common mode gain due to mismatch between inverters in the top and bottom halves of the structure, compared to the ideal case with no common mode gain. Ideally, the common mode gain should be minimised.

![Common Mode Sweep](image)

**Figure 59 - Simulation - Common Mode Sweep (Balanced/Matched vs. Balanced/Mismatched)**

Since the effect of the mismatch is the most evident during a common mode sweep the feasibility of using common mode sweep was examined once again. This time however the objective is primarily to use the information to balance the structure and not to find a high gain setting.

![Common Mode Sweep](image)

**Figure 60 - Simulation - CM Sweep, FF PMOS Constant Difference, NMOS Difference Sweep**
In Figure 60 a common mode sweep test was conducted, where starting with a balanced structure SC = 512, FF = 196, CC = 603. A constant difference between the two FF PMOS transistors arrays was introduced equalling to 320nm. This was achieved by setting the width of one Bit 2 inverter to 14.4µm instead of 14.72µm. Similarly the NMOS portion of one Bit 2 inverter was swept from 2.56µm to 12.16µm. Not surprisingly, what could be seen is that the difference between the output voltages (+ve and −ve output nodes) is observed at low input voltage (~0) and high input voltage (~VDD), where it corresponds to the width difference between PMOS and NMOS transistors respectively. This can be confirmed by observing the constant difference (as annotated) between the two outputs at input voltage of 0, where plots for each output overlap (corresponding to a constant PMOS width difference). Similarly for all sweeps the difference in voltage between the two outputs is constantly varied at input voltage of VDD, and at a point where the NMOS width of an inverter in the top path matched the NMOS width of the inverter in the bottom path, the difference was zero. At first glance this might appear to be a useful source of information. If a comparator was connected to the two outputs, a point where the positive output of the structure changes from being greater than the negative to being less and vice versa, could be detected. Even though such an approach would not guarantee that perfect matching is achieved it would allow the mismatch to be minimised in cases were excessive mismatch is present.

Unfortunately, there is no obvious way of differentiating the common mode gain at the power supply voltages and ground caused by mismatch in FF, CC or SC. Furthermore, the mismatch between NMOS transistors of CC/SC inverters will cause the common mode gain at low input voltages instead of

![Figure 61 - Simulation - CM Sweep, CC PMOS Constant Difference, NMOS Difference Sweep](image-url)
high input voltages in the case of FF inverters. And this makes identifying the source of mismatch even more convoluted. Figure 61 shows a sweep of NMOS values of one CC inverter (sweep of one Bit 0 NMOS inverter from 23.36μm to 24.96μm, while PMOS difference of Bit 0 inverters is held at a constant 0.96μm).

To get a better understanding of the relationship between mismatch and the voltage difference between the outputs due to common mode gain, the common mode sweeps were repeated for a number of width differences between the corresponding inverters of the top and bottom half of the Nauta structure. The starting code was FF = 544, SC = 512, CC = 626. The sweeps were done for PMOS and NMOS individually and also repeated with CMOS/NMOS changes simultaneously. There were no discernible differences between individual and combined PMOS/NMOS sweeps.

![Figure 61](image1)

Figure 61 - Simulation - Output Voltages at Input of 0V (Left)/VDD (Right) vs. Changes in Width (CC)

Figure 62, Figure 63 and Figure 64 show the sweep results for FF1, CC1 and SC1 inverter width. The sweep range was from \( W_{\text{INITIAL}} \pm 1.6 \mu m \) (PMOS)/\( \pm 0.8 \mu m \) (NMOS). At the extreme ranges of the sweep the difference between two outputs were 24mV for NMOS sweep, 34mV for PMOS sweep for FF inverters. The difference was approximately 17mV for PMOS and 11mV for NMOS sweep for CC and SC inverters.

![Figure 62](image2)

Figure 62 - Simulation - Output Voltages at Input of 0V (Left)/VDD (Right) vs. Changes in Width (CC)

![Figure 63](image3)

Figure 63 - Simulation - Output Voltages at Input of 0V (Right)/VDD (Left) vs. Changes in Width (FF)
As can be seen, the voltage difference is substantial only with fairly large mismatch. With smaller mismatch the difference is significantly less in the order of 2-4mV which may be impossible to detect with anything other than the most precise comparator with very little offset. Although the difference between two outputs caused by common mode gain is noticeably larger (close to 50%) for FF inverters, this fact doesn’t aid in differentiating between the common mode gain caused by mismatch of FF, SC or CC inverters.

In conclusion, it can be said that the method discussed in this chapter can’t provide all the necessary information to perfectly balance the Nauta structure and that an alternative method would be required. Such alternative method would have to be used to match any two pairs of the inverters, which would then make it possible to match the remaining pair (one of either FF, SC or CC) by observing the voltage difference between the outputs with common mode input (0V differential) of either 0V or VDD.

Figure 64 - Simulation - Output Voltages at Input of 0V (Left)/VDD (Right) vs. Changes in Width (SC)
4.4.3 65nm Prototype – Testing

The behaviour of the Nauta structure and the feasibility of using PMOS/NMOS ratio adjustments to correct offset was examined using simulations in Ch. 4.4.2, but it is also important to have a good understanding of the behaviour of the real prototype, as ultimately any differences in behaviour compared to simulation results will affect the requirements of the tuning procedure. Testing will reveal the potential of the physical prototype to be tuned for high gain while being affected by mismatch and real environmental factors like temperature. Furthermore, if a quick and accurate testing methodology is found, it may be more time efficient to run tests on a physical prototype instead of lengthy and time consuming simulations. To summarise, the objectives that will be achieved in this sub-chapter include:

- Fast and efficient testing methodology will be found
- A large number of tests will be run to assess the feasibility of calibrating the Nauta structure to have a high gain
- Particular test results will be compared with simulation results (from Ch.4.4.2)

The first series of tests were conducted manually. Manual testing is a relatively time consuming procedure, gathering data on 10s of codes can easily take several minutes considering that for each code it may be required to take over a hundred (voltage or current) measurements. Further analysis of the data takes even longer. Overall this highlights the necessity for a more efficient automated on-chip calibration method. Once a high gain code was found, a sweep through a wider range of codes around it was conducted. Such

![Hysteresis Width Measurements](image)

Figure 65 - Hysteresis Width Measurements (For FF 283-333, SC 256, CC 329)
a range was found to be between codes 283 and 333 for FF inverters, while the SC code was set to a constant 256 and CC to a constant 329.

In Figure 65 we can see measurements of hysteresis width versus the FF codes. From the figure we can see that the hysteresis completely disappears past the FF code of 307. For codes 283 to 306 the hysteresis width fluctuates, but this can be attributed to several factors. Firstly the expected change in width of the inverters is not linear due to \(\times1.8\) scaling of inverter widths, secondly mismatch could cause some of the variations, that coupled with a limited resolution of the sweep (50\(\mu\)V, see Appendix A) some variation should almost certainly be expected.

![Figure 66](image.png)

**Figure 66 - Offset and Hysteresis Width Measurements (For FF 283-333, SC 256, CC 329)**

Figure 66 shows the hysteresis width and offset for the same code sweep (codes increase from bottom to top) as in Figure 65. For all codes the offset is contained within a 40mV range, while most of the codes are contained within 20mV range. The biggest change in offset is observed at code 328 which corresponds to Bit 6 inverter being switched on which appears to have significant PMOS/NMOS imbalance. However, based on Monte-Carlo simulations the offset range was expected to be approximately \(~35mV\) (Figure 56), which is only narrower by 5mV compared to the measured range. Given that there is no significant difference between offset variation during testing and simulations, and that the ability of offset adjustment through variation of PMOS/NMOS ratio was also shown in Figure 53/Figure 54, these results give credence to the proposed direct offset correction strategy in Ch. 3.4
Figure 67 - Input and Output CM Voltage Measurements (For FF 283-333, SC 256, CC 329)

Figure 68 - Measured Output Waveform For Codes with Narrow Hysteresis
Figure 67 shows the common mode voltages of the inputs and the outputs. The input common mode voltage was varied such that the difference between output common mode voltages of positive and negative outputs would be as small as possible. The measurements were taken while a low amplitude ($V_{in-pp} \approx 1.86\text{mV}$) sine wave used as an input. A large common mode difference was seen up to code 294, which is consistent with wide hysteresis, however in Figure 65 hysteresis persists until codes 306. For the remaining codes from 295 to 306 output waveform appears significantly different to the expected sine wave as seen in Figure 68. The test was also repeated with a larger FF code, allowing a waveform with even higher peak-to-peak voltage to be observed. As a result, under certain conditions such as very large FF control codes and a sufficiently large input voltage, if only the peak to peak voltage is observed, the results can still be interpreted as a high open loop gain while the structure is operating with hysteresis. However, the observed waveforms seen in Figure 68 appear to be distorted and this distortion is likely to be an indication of the presence of component frequencies other than the input frequency. Therefore, to guarantee that the structure is not operating with hysteresis the frequency spectrum of the signal should be analysed to confirm the lack of component frequencies.

![Graph showing expected hysteresis range and peak to peak voltage measurements](image)

**Figure 69 - Peak to Peak Voltage Measurements (Individual Outputs and Combined, For FF 283-333, SC 256, CC 329)**

In Figure 69 we can see the peak to peak measurements of the individual outputs and the combined output peak to peak voltage. Figure 70 shows the gain calculated by comparing input and output peak to peak voltages. Once again we can see the high gain measurements for codes 295 to 306 with hysteresis.
To further understand the issue the output signals were analysed by examining the spectrum. An FFT was used to measure the amplitude of the signal at the input frequency of 10kHz, but also at the harmonic frequencies of 20kHz, 30kHz, 40kHz and 50kHz. From Figure 71 it can be seen that harmonics with relatively high amplitude are present for codes 295 to 306. This therefore confirms that for these codes the output waveform is not an ideal sine wave and would appear distorted. While for the codes with no hysteresis the power of the harmonics is reduced to insignificant levels.

![Figure 70 - Differential Gain Measurements (For FF 283-333, SC 256, CC 329)](image1)

![Figure 71 - FFT Peak Measurements (For FF 283-333, SC 256, CC 329)](image2)
Overall it can be seen that using this testing approach to estimate the open loop gain is less than ideal. It is likely that the input amplitude is too large which allows an output waveform to be observed at the outputs even with some hysteresis present. At the same time it is possible that the input waveform might be exceeding the output range of the Nauta structure. It is possible to further reduce the input amplitude, however this might require additional circuitry at the inputs to achieve (as discussed in Appendix A) and it could become increasingly harder to accurately measure the actual amplitude (due to noise and limited resolution of the oscilloscopes used) of the input signals in order to get correct measurements of the gain. More importantly, to guarantee accurate results FFT analysis of the signal would be required, which would significantly increase the testing time, which would make it unsuitable for collecting the large amounts of data needed to produce statistically significant results. As a result, due to the extra effort required to measure the gain using input/output amplitude comparison, different measurement techniques were employed to get precise results quicker and with higher efficiency.

Using source measurement units (SMUs) it is possible to measure the output conductance $g_{ds}$ and transconductance $g_{mu}$, the open loop gain is found by dividing one by the other. Transconductance is found by connecting the two outputs via an SMU and setting the voltage across the outputs to 0V, and then by sweeping the input differential voltage while measuring the current across the outputs (Figure 72a). Dividing the change in current by the change in differential input voltage gives the transconductance of the structure. Similarly output conductance can be measured by setting inputs to a constant $V_{CM}$ voltage and sweeping the voltage difference between the outputs using the SMU while measuring the current (Figure 72b). Once again, plotting the current versus the output voltage difference and measuring the gradient of the plot allows the output conductance to be estimated. Overall, this method allowed measurements to be taken much quicker (as discussed in Appendix A) and in turn allowed larger amounts of data to be gathered.
Figure 73 shows $g_{ds}$ measurements taken using the SMU method. Here we can see that the change from positive to negative transconductance occurs, and hence the highest gain, around code FF 306. This means that the SMU test only produced 4 false-positive measurements where high gain was recorded for codes with hysteresis, compared to 12 false-positive measurements of the traditional transient waveform test using the signal generator.

Since the test described above only covers a short range of codes, a larger code space was explored to see if the Nauta structure is tuneable for high gain at any arbitrary code (that is also sufficiently large). Testing a larger code space and knowing the proportion of codes with high gain is critical for estimating the performance of the tuning procedure. Knowing the proportion of high gain codes in any given range of codes will help to estimate the number of codes that will have to be searched by the tuning algorithm before a high gain code is found, and hence the time it takes to find the high gain code. For the new test the following codes were used. FF = 986, a range of SC codes from 256 to 376 and a range of CC codes from 371 to 571. As a result a total of 24321 codes were tested. During the sweeps the FF code was held constant, although this means that the tuning of the structure won't be as precise, since only SC and CC inverters were used for tuning, the transconductance will remain constant. As such, the only factor in determining the gain that will vary is the output conductance. The transconductance was measured for one of the codes where FF code was set to 986 (SC = 256, CC = 424), this in turn allowed the gain to be calculated for all of the codes tested in this experiment.
Figure 74 shows a plot of the current between the two outputs as the differential input voltage is varied, while the difference between two outputs is set to 0 (as described previously, p.91). Finding the gradient of the plot allows the transconductance to be measured. Due to some fluctuation in measurements a line of best fit was used to average the errors. For this particular FF setting the $g_m$ was then found to be approximately 0.0135S. Code 986 corresponds to almost a maximum possible transconductance value of this particular prototype (where 1023 is the maximum setting), but also makes it possible for the structure to be finetuned further with codes 987-1023. As a result this code should provide a good approximation of the gains this prototype is capable of achieving.

Figure 75 shows a heatmap of the sweep results. Green is used to represent code combinations with excessively high output conductance of over 1S that would certainly not produce a high gain condition. Colours of light orange, orange, and red represent the ranges between 0.0004S and 0.0003S, 0.0003S and 0.0002S, 0.0002S and 0.000135S respectively. These ranges are used to indicate that the output conductance values are approaching the most optimal value required for high gain. Conductances for codes denoted by black fall in range between 0.000135S and 0.0000135S and are used to indicate a high gain condition: 100 to 1000 linear gain (40dB-60dB, for the measured $g_m$ of 0.0135S). Colour purple represents very high gain codes with conductance between 0.0000135S and 0S (gain above 60dB). Colours blue and white correspond to codes with negative output conductance or hysteresis. Blue colour represents codes with output conductance from 0 to -1S, white colour represents all codes with output conductance below -1S.
Figure 75 - Output Conductance Measurement Heatmap
Figure 76 - Output Conductance Measurement Heatmap (SC Sorted)
Due to the ×1.8 scaling of the tuneable Nauta structure the non linear nature of plot in Figure 75 is expected. To get a better understanding of the tuning capabilities the heatmap plot was sorted by SC codes in such a way that their expected width was used instead of control code. The results can be seen in Figure 76. As can be seen some of the “jaggedness” was removed by the sorting. Similarly the plot needs to be sorted by the width corresponding to the codes of CC inverters as well. Figure 77 shows the same data that has been rearranged by CC code width as well. There it can be seen that once again the plot was rearranged to closer resemble a linear pattern, but also that a significant number of codes are out of place. This is likely due to the mismatch in the Bit 0 inverter being switched on at CC code 512.

Figure 78 shows a plot where during sorting the width value of Bit 0 inverter was assigned a value of PMOS/NMOS 50.85µm/25.425µm instead of its ideal value of 48.64µm/24.32µm (a 4.5% change) at which point high gain codes, as denoted by black, follow the increase in CC and SC codes linearly (more or less, since it is still affected by mismatch in other inverters). Overall this experiment highlights the need for ×1.8 scaling as mismatch of 4.5% could easily result in loss of continuity of output transconductance values of the digital to transconductance converters. Also this shows that a very high gain of over 60dB can be achieved at large number of settings reliably, which means that having to go through an excessively large number of codes to find a high gain setting will likely be unnecessary, and hence doing only localised code sweeps to find high gain will be feasible. This then makes it possible for the on-chip calibration procedure to be completed quickly and meet one of its main requirements (discussed in Ch. 3.2).
Figure 77 - Output Conductance Measurement Heatmap (SC and CC Sorted)
Figure 78 - Output Conductance Measurement Heatmap (SC, CC Sorted, Mismatch Corrected)
4.4.4 65nm Prototype - Revised Calibration Procedure

In Ch. 4.4.2 and 4.4.3 the behaviour of the 65nm Nauta structure prototype and its potential to be tuned for high gain were shown through simulations and testing. In this section a possible on-chip calibration procedure will be examined and investigated in greater detail to see how well it can perform and whether the same performance levels can be reached with on-chip calibration as those that were achieved in laboratory conditions in Ch. 4.4.3 (gain > 60dB). In this section a number of high gain control settings will be examined. The codes were chosen by manually pre-tuning the structure to be at high gain, such that the gain was greater than 60 dB and with minor adjustments a hysteresis could also be observed. The settings were chosen to encompass a range FF inverter sizes, from small to very large, so that behaviours of the Nauta structure could be observed and compared using control setting that have different transconductances and different dimensions. The control codes and resulting dimensions can be seen in Table 21.

<table>
<thead>
<tr>
<th>Setting</th>
<th>FF Code</th>
<th>FF PMOS Width (µm)</th>
<th>SC Code</th>
<th>SC PMOS Width (µm)</th>
<th>CC Code</th>
<th>CC PMOS Width (µm)</th>
<th>Gain (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>195</td>
<td>23.82</td>
<td>512</td>
<td>48.64</td>
<td>602</td>
<td>61.35</td>
<td>62.56</td>
</tr>
<tr>
<td>Medium</td>
<td>544</td>
<td>53.26</td>
<td>512</td>
<td>48.64</td>
<td>626</td>
<td>64.57</td>
<td>69.74</td>
</tr>
<tr>
<td>Large</td>
<td>1006</td>
<td>105.7</td>
<td>512</td>
<td>48.64</td>
<td>690</td>
<td>70.97</td>
<td>78.25</td>
</tr>
</tbody>
</table>

The revised procedure was briefly described in Ch. 3.4 and employs a differential input scheme, and comparator based forward-backward sweep hysteresis detection (first discussed in Ch 3.4). Under ideal conditions this method would only require three test voltages and a comparator as seen in Figure 79. One of the inputs would be connected to $V_{CM} = V_{DD}/2$, while the other input, $V_{TEST}$ would be varied between $V_{DD}$, $V_{CM}$ and GND. The comparator output would then be recorded for the point at which $V_{TEST} = V_{CM}$. The process would then be repeated while $V_{TEST}$ is varied from GND to $V_{CM}$ to $V_{DD}$. If the comparator output did not change during the reverse sweep compared to the forward sweep, there would be no hysteresis. On the other hand if the comparator output did change, it would suggest that hysteresis is present and the CC code needs to be reduced to the last setting where no hysteresis was detected. If the hysteresis is detected as soon as it appears, reverting to the last code with no hysteresis will allow the Nauta structure to operate at high gain, where the gain will only be limited by the inverter width.

![Figure 79 - Comparator Based Testing Setup](image-url)
step size (the smaller step size, the better). Alternatively the gain can then be further fine-tuned by exploiting the fact that the FF inverters only contribute output conductance to the calculation of gain. Instead of reducing the code of CC inverters after detecting hysteresis, the FF code can be increased until hysteresis is no longer present. Since the output conductance in general is smaller than the transconductance, adjusting the FF inverters allows the gain to be tuned using much smaller steps and will get closer to the true maximum gain of the structure.

![Graph showing offset distribution](image)

**Figure 80 - Simulation - Offset Distribution for Medium FF setting (Monte-Carlo, 100 Runs)**

The calibration techniques described above assume that there will be no offset and that the hysteresis will always appear at $V_{\text{TEST}} = V_{\text{CM}}$. However, it was already shown that PMOS/NMOS imbalance will cause hysteresis offset (Ch. 4.4.2, Figure 53, Figure 54). To reiterate this fact 100 run Monte-Carlo simulation in TSMC 65nm process, showed that for the Medium-FF setting, where process and mismatch variation was simulated, the offset would likely fall within a range of +/- 10 mV as seen in Figure 80. If the hysteresis occurs at a large offset it will only be detected once the width of the hysteresis becomes exceedingly large, if there is only one test voltage at $V_{\text{TEST}} = V_{\text{CM}}$. This will result in the structure being tuned past the maximum gain point and will render the tuning procedure ineffective. However, in the previous section it was also shown that varying PMOS/NMOS ratio by varying the control code for only PMOS or NMOS parts of the inverter was possible.

Figure 81 shows how the PMOS/NMOS ratio affects the offset in a structure set to Medium-FF setting. It can be seen that in the worst case (for FF inverters) the offset range was 6.99mV for a change of 3.2µm in NMOS width. Considering that the NMOS width of the smallest inverter is 120nm,
expected change in width for that inverter is approximately 262µV. This is then the maximum excepted offset shift caused by a single code change in the ideal case, the number could increase with mismatch. This figure will vary for different settings as the smallest inverter size remains constant but the overall size of the Nauta structure increases, making the smallest inverter an even smaller proportion relative to the rest of the structure. The consequences of this become clearer once the design requirements for reference voltages are considered.

![Figure 81 - Simulation - Offset Caused by Variation of PMOS/NMOS Ratio](image)

To correct the offset and detect low width hysteresis simultaneously the number of test voltages needs to be increased by adding two more test voltages at \( V_{\text{CM}} + \Delta \) and \( V_{\text{CM}} - \Delta \). These voltages will set the bounds that will determined the maximum offset and hysteresis width and it will be possible to guarantee that the offset and hysteresis are within these bounds. As a result, these boundaries will have to be wide enough such that when tuning the offset it will always “pass” within the boundaries. For example, for a Medium-FF setting an offset tuning step of 262µV can be expected, this value can be easily rounded to 300µV if mismatch is taken into account. The boundaries then have to be wider than 300µV, if we liberally double this value then it can be ensured that the offset can be tuned, even in worst case scenario to be between \( V_{\text{CM}} + \Delta \) and \( V_{\text{CM}} - \Delta \).

The three test voltages will then be at \( V_{\text{CM}} - \Delta = 899.7\text{mV} \), \( V_{\text{CM}} = 900\text{mV} \), \( V_{\text{CM}} + \Delta = 900.3\text{mV} \). For each test voltage the comparator output will be recorded twice, during the forward sweep and during the backward sweep. The variables \( V_{\text{CM}} F - \Delta \), \( V_{\text{CM}} F \) and \( V_{\text{CM}} F + \Delta \) are used to record the comparator output during the forward sweep for test voltages \( V_{\text{CM}} - \Delta \), \( V_{\text{CM}} \) and \( V_{\text{CM}} + \Delta \)
respectively. Similarly, variables $V_{CM}B-\Delta$, $V_{CM}B$ and $V_{CM}B+\Delta$ are used to record the comparator output during a backward sweep. Consequently the complexity of the tuning algorithm will need to increase to iteratively correct for offset and gain as needed. The six recorded variables can then be analysed in two stages. The first stage is primarily used to detect the presence of hysteresis. To achieve this, the comparator output variables are grouped into pairs comprised of the forward and backward sweep variables for the same test voltage. The variables are then compared digitally, where 0 corresponds to forward and backward sweep results being equal and 1 corresponds to sweep results being different. Therefore 0 implies that no hysteresis was detected and 1 that hysteresis was found. Table 22 summarises the actions that will be taken for every possible result of forward and backward variable comparison.

<table>
<thead>
<tr>
<th>Variable Comparison</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{CM}F-\Delta$ &amp; $V_{CM}B-\Delta$</td>
<td>Check for offset (Stage II)</td>
</tr>
<tr>
<td>$V_{CM}F$ &amp; $V_{CM}B$</td>
<td>Decrease positive offset</td>
</tr>
<tr>
<td>$V_{CM}F+\Delta$ &amp; $V_{CM}B+\Delta$</td>
<td>Decrease CC array size, maximum gain point found</td>
</tr>
<tr>
<td>0 0 0</td>
<td>Check for offset (Stage II)</td>
</tr>
<tr>
<td>0 0 1</td>
<td>Decrease positive offset</td>
</tr>
<tr>
<td>0 1 0</td>
<td>Decrease CC array size and decrease positive offset</td>
</tr>
<tr>
<td>0 1 1</td>
<td>Decrease CC array size and decrease positive offset</td>
</tr>
<tr>
<td>1 0 0</td>
<td>Decrease negative offset</td>
</tr>
<tr>
<td>1 0 1</td>
<td>Impossible condition</td>
</tr>
<tr>
<td>1 1 0</td>
<td>Decrease CC array size and decrease negative offset</td>
</tr>
<tr>
<td>1 1 1</td>
<td>Decrease CC array size</td>
</tr>
</tbody>
</table>

Table 22 - Tuning Algorithm Stage I

If hysteresis is present, the first stage of algorithm will determine whether the hysteresis is too wide or is not centred correctly, and if there is also an offset present. If the hysteresis is narrow enough and has no offset, the algorithm will stop as this indicates that the maximum gain point is found. If no hysteresis is detected the algorithm will proceed to stage two. This stage will determine whether there is positive or negative offset present. This is done by comparing $V_{CM}F-\Delta$ and $V_{CM}F+\Delta$ (or $V_{CM}B+\Delta$ and $V_{CM}B+\Delta$, since no hysteresis was detected and both comparisons will yield the same result). Three different conditions are possible: variables are not equal, both variables are 1 or both are 0. If it is found that the variables are not equal, offset correction will not be required. Alternatively, if the variables are found to be equal, appropriate offset correction will be undertaken by varying the P:N ratio of the FF, SC or CC inverters by making corresponding changes to the control code set by a digital control circuit. The actions that can be taken during the second stage of the tuning algorithm are summarised in Table 23. This method only allows the average PMOS/NMOS ratios of inverters in the top and bottom halves to be matched. Using this method does not guarantee that PMOS/NMOS ratio will be the same within every inverter array and that inverter pairs FF1/FF2, SC1/SC2 and CC1/CC2 will be matched size-wise.
This, in turn, could still have negative effects, such as high common mode gain, on the performance of the structure. This technique, however, is still an excellent first step in ensuring that the Nauta structure operates at maximum gain. Also, the proposed offset correction approach doesn’t require a lot of additional hardware as it primarily relies on the fact that the transistors in the structure can be turned on or off individually, and without any sort of offset correction, a much larger number of reference voltages would be required.

<table>
<thead>
<tr>
<th>Variable Conditions</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V_{\text{CM}F+\Delta} \neq V_{\text{CM}F-\Delta} )</td>
<td>No offset correction needed, increase CC array size</td>
</tr>
<tr>
<td>( V_{\text{CM}F+\Delta} = V_{\text{CM}F-\Delta} = 0 )</td>
<td>Decrease negative offset</td>
</tr>
<tr>
<td>( V_{\text{CM}F+\Delta} = V_{\text{CM}F-\Delta} = 1 )</td>
<td>Decrease positive offset</td>
</tr>
</tbody>
</table>

Table 23 - Tuning Algorithm Stage II

Consider a possible tuning scenario shown in Figure 82 and Table 24.

**Figure 82a** - The initial condition. Since no hysteresis is present all comparison variables will be 0 and the algorithm will proceed to Stage 2. At Stage 2 the algorithm will determine that the comparator output is 0 at both test reference voltages, implying that there is negative offset and it needs to be corrected.

**Figure 82b** - The result of corrections done based on data gathered in (a). The action taken by the algorithm resulted in overcorrection and the structure now has small positive offset and hysteresis. The hysteresis is only detected at \( V_{\text{CM}+\Delta} \) which requires reduction of the size of CC inverters and reduction of positive offset. Since hysteresis is detected, there is no need to proceed to Stage 2.
**Figure 82c** - The result of corrections done based on data gathered in (b). Hysteresis is still present, but it has no offset. This is the required condition to trigger the end of the calibration. At this point no offset correction will be done, but the size of CC inverters will be reduced.

**Figure 82d** - The final state of the Nauta structure. If the algorithm performs as expected, the structure will have no hysteresis and have a high gain instead. Little to no offset (bound by reference voltages $V_{CM-\Delta}$ and $V_{CM+\Delta}$) will be present.

In general, the tuning procedure can be implemented relatively easily, however whether it can successfully calibrate the structure for high gain requires further investigation. As such, how the change in width correlates to an increase in hysteresis and gain will be examined. This is important to consider because once the hysteresis is detected the CC inverter size needs to be decreased (or the FF inverters need to be increased) to put the structure

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Recorded Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Stage 1</td>
</tr>
<tr>
<td></td>
<td>$V_{CMF-\Delta}$ &amp; $V_{CMB-\Delta}$</td>
</tr>
<tr>
<td>(a)</td>
<td>0</td>
</tr>
<tr>
<td>(b)</td>
<td>0</td>
</tr>
<tr>
<td>(c)</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 24 - Tuning Example

![Figure 83 - Simulation - Change in Hysteresis Due to Variation in CC Inverters](image)
into a high gain mode. In a non ideal case, the algorithm may detect hysteresis that is significantly less than the distance between the reference voltages and roll back inverter width by the equivalent width that guarantees the hysteresis will be reduced by the difference in the reference voltages and the structure will be stable in all cases.

Figure 83 shows a change in hysteresis width as the width of CC inverters is increased. A large increase in hysteresis width can be seen, as a result using CC inverter for fine tuning the gain is not optimal. CC inverter can be used to get the structure near high gain setting without affecting the transconductance and then FF inverters can be used to fine tune the gain, with only minimal change in transconductance. Figure 84 shows how the change in the width of FF inverters affects the hysteresis. The change in hysteresis is a lot more gradual, this therefore confirms the ability of FF inverters to tune the gain with higher precision.

![Figure 84 - Simulation - Change in Hysteresis Due to Variation in FF Inverters](image)

Table 24 shows the comparison of the change in gain due to an increase in the width of FF inverts and the change in hysteresis due to a decrease in FF inverter widths. Let’s examine the example discussed previously where for the Medium FF setting the reference voltages were set to be 600µV apart. In a worst case scenario a hysteresis with very narrow width will be detected. But to account for a possible hysteresis of just under 600µV the width will still have to be rolled back by approximately 3.2µm (“roll-back” width) since that corresponds to a change in width of just over 600µV (678µV). A corresponding drop in gain will be from 69.74dB to 44.63dB as seen in Table 25. That is certainly a relatively large drop in gain, but this value still falls within high
gain range, and it may be possible to optimise the reference voltage selection and other aspects of the tuning procedure further to improve the worst case performance. On the other hand the above results are based on simulations and mismatch may degrade the performance.

<table>
<thead>
<tr>
<th>Change in W(PMOS/2*Nmos) of FF Inverters (µm)</th>
<th>Large FF (dB)</th>
<th>Large FF Hysteresis (µV)</th>
<th>Medium FF (dB)</th>
<th>Medium FF Hysteresis (µV)</th>
<th>Small FF (dB)</th>
<th>Small FF Hysteresis (µV)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>78.25</td>
<td>0</td>
<td>69.74</td>
<td>0</td>
<td>62.56</td>
<td>0</td>
</tr>
<tr>
<td>0.64</td>
<td>63.06</td>
<td>24</td>
<td>56.56</td>
<td>48</td>
<td>49.72</td>
<td>92</td>
</tr>
<tr>
<td>1.28</td>
<td>57.88</td>
<td>83</td>
<td>51.63</td>
<td>145</td>
<td>44.94</td>
<td>356</td>
</tr>
<tr>
<td>1.92</td>
<td>54.69</td>
<td>152</td>
<td>48.57</td>
<td>312</td>
<td>42.01</td>
<td>757</td>
</tr>
<tr>
<td>2.56</td>
<td>52.38</td>
<td>233</td>
<td>46.35</td>
<td>478</td>
<td>39.91</td>
<td>1220</td>
</tr>
<tr>
<td>3.20</td>
<td>50.58</td>
<td>333</td>
<td>44.63</td>
<td>678</td>
<td>38.30</td>
<td>1820</td>
</tr>
<tr>
<td>3.84</td>
<td>49.11</td>
<td>438</td>
<td>43.21</td>
<td>929</td>
<td>36.99</td>
<td>2480</td>
</tr>
</tbody>
</table>

Table 25 - Change in DC Gain Due to Decrease in FF Width (AC Simulation) and Change in Hysteresis Due to Increase in FF Width

Figure 85 and Figure 86 show how simulation results compare to test results conducted on a prototype. Not surprisingly an overall lower gain (by approximately 5dB) was measured as well as some fluctuation in hysteresis due to mismatch (and possibly due to measurement resolution) could be seen. This will likely have a negative effect on the calibration procedure, but there is no indication that with some refinement the procedure would not perform well since it was still possible to achieve a gain in excess of 65dB.
In summary, it can be said that the proposed technique is feasible based on the simulation results. Testing results also confirm that the structure exhibits the same behaviour, with some deviation from ideal results due to mismatch. However, the performance is highly dependent on the spacing of reference voltages used for testing, where tighter spacing can guarantee higher worst case gain. On the other hand, if the noise levels will exceed the difference between the reference voltages the reliability of the calibration will be compromised.

4.5 Voltage References and Comparator

As discussed in Ch. 4.4, voltage references are a critical part of the tuning procedure and the implementation of the reference voltages can have a significant impact on the performance of the tuning circuit. Any discrepancy in reference voltage spacing from the ideal values will change the expected worst case gain. A number of voltage reference implementation techniques can be considered, such as a simple resistor ladder or a bandgap reference. A resistive ladder implementation in its simplest form would consist of a number of resistors in series. The overall resistance can be found using Eq. 9. Where L is the length, $\rho$ is resistivity and $S$ is cross-section area of the resistor. Given that the layer thickness will be uniform (poly, metal, etc), the resistance of any given resistor will be determined by its length and width. The width will also have a minimum value for any given technology and therefore to increase the resistance the overall length and therefore area of the resistor will have to be increased. The benefits of using larger resistors is reduced power consumption, however using exceedingly large resistors will negate all the advantages gained in overall area reduction of the OTA through the use of a digitally assisted Nauta structure. Additionally, the noise levels would have to be taken into account. Eq. 10 shows the factors affecting the thermal noise.
levels E, where $T$ is the temperature, $R$ is the resistance, $B$ is the bandwidth and $k$ is the Boltzmann constant. As a result, increasing the resistance will also increase the noise. Overall the choice of resistance values for the reference voltage ladder is a compromise between area, noise and power consumption.

$$R = \frac{L \rho}{S}$$

(9)

$$E = \sqrt{4kTRB}$$

(10)

As discussed in Ch. 4.4 and shown in Table 25 the spacing of reference voltage may have to be very tight (<100μV for a guaranteed very high gain detection) as a result noise is an important factor to consider when designing the reference voltages. On the other hand, given that it is possible to adjust the PMOS/NMOS ratio of the inverters in the Nauta structure and adjust the overall offset of the structure, the shift of reference voltages (where the central reference voltage is shifted away from the ideal $V_{DD}/2$ voltage) due to mismatch is not major issue. Nonetheless, mismatch will affect the overall spacing of reference voltage (by making the spacing wider in the worst case scenario) and the worst case gain estimations will have to account for that.

<table>
<thead>
<tr>
<th>Technology</th>
<th>30</th>
<th>31</th>
<th>32</th>
<th>33</th>
<th>34</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noise (μV, 0.1-10Hz)</td>
<td>9.1 rms</td>
<td>6.1 rms</td>
<td>53 pp</td>
<td>5.5 rms</td>
<td>0.15 rms</td>
</tr>
<tr>
<td>Supply Voltage (V)</td>
<td>1.4</td>
<td>1.8</td>
<td>1.8</td>
<td>1.2</td>
<td>3.4</td>
</tr>
<tr>
<td>Current (μA)</td>
<td>116</td>
<td>55</td>
<td>100</td>
<td>416</td>
<td>443</td>
</tr>
<tr>
<td>Area (mm²)</td>
<td>1.17</td>
<td>0.12</td>
<td>1.22</td>
<td>0.086</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Table 26 - Performance of Bandgap Voltage references

Bandgap reference is another common type of voltage reference and the performance of various implementations is summarised in Table 26. Although implementation of bandgap references is relatively costly in terms of increase in power consumption and area requirements, low noise levels can be achieved (1-60μV Peak-to-Peak) which makes the use of closely spaced references (<1mV) feasible. Negating the effects of noise could also be done by taking more than one measurement for every test point and averaging the results. However, taking multiple measurements would result in longer tuning time.

Comparator is another circuit that will need to be implemented for the tuning to work. One of the more common shortfalls of a comparator is offset. In theory it might be possible that the reference voltages are spaced so closely that due to the offset comparison at different reference voltages will yield the same result even if small changes in the output voltages of the Nauta structure occur. This is however not likely to occur due to previously discussed limitation of placing reference voltages too closely together. While exceedingly
large offset will prevent the tuning procedure from working correctly, having a small offset can be beneficial. In case where identical or near identical voltages are compared the results may be unpredictable if the comparator has no offset. For example, it is possible that the voltages may be compared during backward and forward sweeps before the voltages could settle to their final value, if that occurs, the comparison may yield different results during forward and backward sweeps. This would result in a false-positive detection of hysteresis.
5. HDL Implementation

A preliminary HDL version of the calibration algorithm was implemented in Verilog and tested alongside the TSMC 65nm prototype of the Nauta structure.

Figure 87 - HDL Implementation Diagram

Figure 87 shows a diagram of the operation of the calibration module. State control module determines the actions that should be performed by the calibration circuits. The calibration procedure operates by alternating between hysteresis detection (Stage I), offset correction (Stage II) or being idle when the calibration is complete, but the module can be instructed to re-tune the structure if necessary (for example, to recalibrate due to temperature changes). The pseudo code for hysteresis detection and offset correction modules can be seen in Figure 88 and Figure 89 (see Appendix B for full code). After the hysteresis and offset detection stages the request to increase or decrease calibration codes is sent to the register control block, here the actual changes in codes can be determined if it is not a straight forward

```plaintext
set_gnd: Set input voltage to ground
set_vccm: Set input voltage to Vcm-A
VCMm: Record comparator output as cmpm
set_vcmp: Set input voltage to Vcm+∆VCMm:
VCMp: Record comparator output as cmpp

if (cmpm != cmpp)
    Increase CC inverter size
if (cmpm == 1'b1 && cmpp == 1'b1)
    Decrease positive offset
    Increase SC1 or CC1 NMOS / Decrease SC2 or CC2 NMOS
if (cmpm == 1'b0 && cmpp == 1'b0)
    Decrease negative offset
    Increase SC2 or CC2 NMOS / Decrease SC1 or CC1 NMOS
```

Figure 88 - Offset Correction Pseudo Code
change by one, but for example, requires to account for non-linear scaling of widths. While the changes to the control codes are being made, the register control module signals the state control module to halt the operation of hysteresis detection and offset correction modules. The hysteresis and offset blocks can operate in SC-CC or FF modes. In SC-CC the tuning is done with self-coupled and cross-coupled inverters only, SC-CC tuning is done first, once that is done the tuning procedure runs again using only feed-forward inverters to fine-tune and maximise the gain. Once the appropriate adjustments by the register control module are made to the Nauta structure control codes, it signals the state control module that the calibration can continue.

```plaintext
set_gnd:    Set input voltage to ground
set_vcmfm:  Set input voltage to Vcm-Δ
VCMFm:      Record comparator output as VCMFm
set_vcmf:   Set input voltage to Vcm
VCMF:       Record comparator output as VCMF
set_vcmfp:  Set input voltage to Vcm+Δ
VCMFp:      Record comparator output as VCMFp
set_vdd:    Set input voltage to VDD
set_vcmdp:  Set input voltage to Vcm+Δ
VCMBP:      Record comparator output as VCMBP
set_vcom:   Set input voltage to Vcm
VCMB:       Record comparator output as VCMB
set_vcmbm:  Set input voltage to Vcm-Δ
VCMBm:      Record comparator output as VCMBm

if (VCMFp != VCMBP)
    cmpp <= 1'b1;
if (VCMF != VCMB)
    cmp <= 1'b1;
if (VCMFm != VCMBm)
    cmpm <= 1'b1;
if (cmpm == 1'b0 && cmp == 1'b0 && cmpp == 1'b0)
    Check for offset (Stage II)
if (cmpm == 1'b0 && cmp == 1'b0 && cmpp == 1'b1)
    Decrease gain and decrease positive offset
    Increase SC1 NMOS / Decrease CC2 NMOS
if (cmpm == 1'b0 && cmp == 1'b1 && cmpp == 1'b0)
    Decrease CC inverter size by a “roll-back” width
    Maximum gain point found
if (cmpm == 1'b0 && cmp == 1'b1 && cmpp == 1'b1)
    Decrease gain and decrease positive offset
    Increase SC1 NMOS / Decrease CC2 NMOS
if (cmpm == 1'b1 && cmp == 1'b0 && cmpp == 1'b0)
    Decrease gain and decrease negative offset
    Increase SC2 NMOS / Decrease CC1 NMOS
if (cmpm == 1'b1 && cmp == 1'b1 && cmpp == 1'b0)
    Decrease gain and decrease negative offset
    Increase SC2 NMOS / Decrease CC1 NMOS
if (cmpm == 1'b1 && cmp == 1'b1 && cmpp == 1'b1)
    Decrease CC inverter size
```

Figure 89 - Hysteresis Detection Pseudo Code
Simulations were run as proof of concept, as such further refinements to the test bench and the code will be required in the future. Specifically, during the simulations ideal voltage controlled voltage sources were used to set up voltage references and the comparator. Furthermore the “roll-back” width (change in width required to guarantee that no hysteresis will be present) during the calibration was not optimised at this stage, since the tuning can generate a different code for every one of the four transistors in the CC or FF pairs, simply reducing all of them by the same width could introduce unnecessary offset. As a result extensive Monte-Carlo simulations may be required to optimise the “roll-back” width estimation. A placeholder strategy of simply reducing all codes until a smallest possible reduction in width is achieved was used, as a result the hysteresis could still be observed for most of the codes generated by the calibration module. Following that the gain was measured (see Appendix B for test bench) once the width of FF inverters was rolled back by at least 3.2µm/1.6µm (PMOS/NMOS), the actual rollback width may vary and be greater due to limited width adjustment resolution limited by the step sizes.

The simulations were conducted by manually introducing an imbalance in initial code, the Medium-FF (FF-544, SC-512, CC-626) setting was used as a basis and various inverter control codes were changed as seen in Table 27, where they are highlighted in red. The right side of Table 27 shows the codes that were generated by the tuning algorithm. As in the previous example reference voltages were set to 599.7mV, 600mV (V_{CM}) and 600.3mV. The clock period was set to 1µs. Additionally a 0.5mV offset was introduced into the ideal comparator as it was found that in the situation where no hysteresis exists false positive results could be retuned if the voltages of both outputs were very close, but due to insufficient settling time one would appear to be greater than the other during the forward sweep and the reverse would be true during the backward sweep (the test condition for the presence of hysteresis). For example, during the calibration the structure may in fact be tuned so precisely that it has no or negligible differential offset, and therefore there will be little to no difference between the voltages of the two outputs. But, as mentioned previously, due to limited settling time the voltages may appear to be different during the measurements done during forward and backward sweeps, even if the difference is minute. Additionally, it is likely that a real implementation of the comparator will have some offset, the fact that the calibration procedure benefits from comparator having offset, may in fact simplify the design process of the comparator due to not having to design a comparator with no offset or requiring extreme precision.
### Table 27 - Initial and Post Calibration Codes

<table>
<thead>
<tr>
<th>#</th>
<th>FF</th>
<th>SC</th>
<th>CC</th>
<th>FF</th>
<th>SC</th>
<th>CC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>P</td>
<td>N</td>
<td>P</td>
<td>N</td>
<td>P</td>
<td>N</td>
</tr>
<tr>
<td>2</td>
<td>544</td>
<td>544</td>
<td>544</td>
<td>512</td>
<td>512</td>
<td>512</td>
</tr>
<tr>
<td>3</td>
<td>544</td>
<td>544</td>
<td>544</td>
<td>512</td>
<td>512</td>
<td>512</td>
</tr>
<tr>
<td>4</td>
<td>544</td>
<td>544</td>
<td>544</td>
<td>512</td>
<td>512</td>
<td>512</td>
</tr>
<tr>
<td>5</td>
<td>550</td>
<td>544</td>
<td>544</td>
<td>512</td>
<td>512</td>
<td>512</td>
</tr>
<tr>
<td>6</td>
<td>530</td>
<td>544</td>
<td>544</td>
<td>512</td>
<td>512</td>
<td>512</td>
</tr>
<tr>
<td>7</td>
<td>544</td>
<td>544</td>
<td>544</td>
<td>518</td>
<td>512</td>
<td>512</td>
</tr>
<tr>
<td>8</td>
<td>544</td>
<td>544</td>
<td>544</td>
<td>518</td>
<td>512</td>
<td>512</td>
</tr>
<tr>
<td>9</td>
<td>544</td>
<td>544</td>
<td>529</td>
<td>512</td>
<td>512</td>
<td>512</td>
</tr>
<tr>
<td>10</td>
<td>544</td>
<td>544</td>
<td>544</td>
<td>425</td>
<td>512</td>
<td>512</td>
</tr>
<tr>
<td>11</td>
<td>548</td>
<td>544</td>
<td>544</td>
<td>517</td>
<td>512</td>
<td>512</td>
</tr>
<tr>
<td>12</td>
<td>544</td>
<td>544</td>
<td>547</td>
<td>517</td>
<td>512</td>
<td>512</td>
</tr>
<tr>
<td>13</td>
<td>530</td>
<td>544</td>
<td>544</td>
<td>425</td>
<td>512</td>
<td>512</td>
</tr>
<tr>
<td>14</td>
<td>555</td>
<td>530</td>
<td>555</td>
<td>425</td>
<td>512</td>
<td>512</td>
</tr>
<tr>
<td>15</td>
<td>544</td>
<td>544</td>
<td>544</td>
<td>512</td>
<td>512</td>
<td>512</td>
</tr>
</tbody>
</table>

### Table 28 - Calibration Results

<table>
<thead>
<tr>
<th>#</th>
<th>Offset (µV)</th>
<th>Hysteresis (µV)</th>
<th>Offset (µV)</th>
<th>Hysteresis (µV)</th>
<th>Time to Tune (µS)</th>
<th>Gain @ FF 3.2µm Rollback (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>720</td>
<td>200</td>
<td>40</td>
<td>0</td>
<td>250</td>
<td>39.77</td>
</tr>
<tr>
<td>2</td>
<td>-450</td>
<td>0</td>
<td>-50</td>
<td>50</td>
<td>340</td>
<td>41.45</td>
</tr>
<tr>
<td>3</td>
<td>800</td>
<td>250</td>
<td>-100</td>
<td>150</td>
<td>475</td>
<td>43.17</td>
</tr>
<tr>
<td>4</td>
<td>-550</td>
<td>0</td>
<td>40</td>
<td>200</td>
<td>765</td>
<td>43.06</td>
</tr>
<tr>
<td>5</td>
<td>1170</td>
<td>0</td>
<td>150</td>
<td>140</td>
<td>610</td>
<td>41.89</td>
</tr>
<tr>
<td>6</td>
<td>-1750</td>
<td>50</td>
<td>-50</td>
<td>50</td>
<td>570</td>
<td>41.41</td>
</tr>
<tr>
<td>7</td>
<td>1550</td>
<td>0</td>
<td>25</td>
<td>300</td>
<td>1010</td>
<td>44.85</td>
</tr>
<tr>
<td>8</td>
<td>-1225</td>
<td>0</td>
<td>25</td>
<td>0</td>
<td>615</td>
<td>39.60</td>
</tr>
<tr>
<td>9</td>
<td>-1950</td>
<td>50</td>
<td>-125</td>
<td>190</td>
<td>780</td>
<td>43.79</td>
</tr>
<tr>
<td>10</td>
<td>-3450</td>
<td>0</td>
<td>20</td>
<td>340</td>
<td>1380</td>
<td>44.28</td>
</tr>
<tr>
<td>11</td>
<td>1550</td>
<td>200</td>
<td>-80</td>
<td>100</td>
<td>600</td>
<td>41.69</td>
</tr>
<tr>
<td>12</td>
<td>2250</td>
<td>0</td>
<td>-60</td>
<td>50</td>
<td>760</td>
<td>41.50</td>
</tr>
<tr>
<td>13</td>
<td>-23400</td>
<td>0</td>
<td>-150</td>
<td>200</td>
<td>5370</td>
<td>43.31</td>
</tr>
<tr>
<td>14</td>
<td>15600</td>
<td>0</td>
<td>270</td>
<td>0</td>
<td>3100</td>
<td>41.62</td>
</tr>
<tr>
<td>15</td>
<td>-2610</td>
<td>1480</td>
<td>150</td>
<td>100</td>
<td>720</td>
<td>43.51</td>
</tr>
</tbody>
</table>

114
The results of the simulations can be seen in Table 28. As expected there was some hysteresis present for the calibrated settings, however none of them exceeded the 600µV threshold set by the difference of the reference voltages and all offsets were corrected to be within the threshold voltage boundaries, which confirms the correct operation of the algorithm. The calibration was run with a 1µs clock. The calibration times were affected by initial offset and were completed on average at a rate of 3.4mV/ms (offset corrected in 1ms). For the range of codes used in the simulations the offset is expected to be within ±10mV based on Monte-Carlo simulations. This means that the tuning can be expected to be completed in or under approximately 34ms. Overall, it can be seen that the algorithm performs as expected and with further optimisations could be a viable solution to calibration of the Nauta structure.
6. Conclusion and Future Work

A tuning procedure was explored that exploits the positive feedback nature of the CC inverters of the Nauta structure that causes hysteresis if tuned incorrectly. Initially a procedure relying on observing the changes in DC levels that could be caused by hysteresis was examined, but was found to be unreliable and unfeasible as it required a number of characterisation steps to be conducted that could not be realistically implemented as an on-chip solution. The procedure was then further revised to detect hysteresis by comparing the outputs of the Nauta structure using a comparator during a forward differential input DC sweep and a backward differential input DC sweep with a minimal number of test voltages. And instead of characterising the structure to identify sections of inverter arrays with low PMOS/NMOS imbalance, it was proposed to attempt to counter the effects of mismatch by varying the PMOS and NMOS widths, made possible by the nature of the tri-state inverters used in the modified Nauta structure.

The feasibility of the proposed method was examined in great detail through simulations and testing of a prototype. The reliability of detection of hysteresis as an indicator of high gain was demonstrated and the overall ability of the Nauta structure to be tuned for high gain was shown through simulations and testing. Furthermore, extensive simulations were conducted to identify a relationship between the change in inverter dimensions, hysteresis and gain to determine the requirements for the supplementary hardware required for the tuning to be done (reference voltages, comparator). The simulation results have shown that a noticeable gain drop can occur in a worst case scenario (such as a drop from 69.74dB to 44.63dB, but the figures will vary depending on the overall sizing of the structure and the implementation of the reference voltages). The test results have shown that the presence of mismatch is likely to affect the outcome of calibration as well. Additionally, initial simulations of the digital control module have confirmed that the Nauta structure does operate as expected when implemented in a calibration loop with a comparator and reference voltages.

A number of issues can be addressed and overcome to optimise the performance of the proposed technique. Firstly a suitable comparator implementation will have to be found. Secondly, voltage reference ladder will have to be implemented. Considering the tight spacing of the voltage references this is likely to be a challenging task. Furthermore, to improve the performance additional voltage references may be required in addition to the one described in this work. For example instead of 3 voltages, 5 reference voltages can be used, by inserting another reference voltage between $V_{CM}$ and $V_{CM} + \Delta$, and also between $V_{CM}$ and $V_{CM} - \Delta$. This will make it possible to use any 3 of the 5 reference voltages for detection of hysteresis. Tighter spacing of
voltages will improve the worst case scenario when it comes to achieving high gain, but it will also make the reference voltage ladder harder to implement.

The changes to the digital tuning algorithm that can be made include the “roll-back” width optimisation. This will likely have to take into account the mismatch and all possible changes in codes and widths introduced during calibration, rather than just rely on the width estimation based on the ideal simulations. Additionally, changes in the algorithm will have to be made if the number of reference voltages is increased.
Appendix A - Test Equipment

Introduction

This chapter will provide a brief overview of test equipment used during the testing of the Nauta structure and the methodology used to gather the required data. The advantages and disadvantages of the equipment and testing techniques will be examined. The instruments listed below were the most critical for testing and data gathering:

- Tektronix AFG3102 - Signal Generator
- Tektronix TDS2024C - Oscilloscope
- NI PXI-4072 - Digital Multimeter
- NI PXI-4130 - SMU
- NI PXI-4132 - SMU

AFG3102

AFG3102 is dual channel signal generator. Since the Nauta structure is a differential amplifier, having two outputs capable of generating a differential input signal makes this particular model, for the most part, suitable for testing the Nauta structure. Without access to this signal generator the tests initially had to be limited to common mode and differential DC sweeps. Some of the key specifications include:

- Bandwidth: 100MHz (Sine Wave)
- Minimum Input Voltage: 40 mV_{P-P}
- DC Offset Resolution: 1mV
- Phase Adjustment: -180° to 180°
- USB Interface

The simplest way of measuring the open loop gain of the structure is to apply a low amplitude signal at the inputs and observes the outputs. The limiting factor of such tests would be the minimum input amplitude, as at expected gains (of greater than 40dB) the output would exceed the difference between power supply rails (1.8V for 180nm technology and 1.2V for 65nm technology). To overcome this, a simple circuit modification was used as seen in Figure 90.
A low value resistor is added between the two inputs which essentially forms a voltage divider for the differential input signal, this results in the input signal being attenuated (in the example the attenuation factor would be approx. 100). Since the bandwidth of the signal generator is only 100 MHz measuring the unity gain in open loop configuration would also be impossible if the expected unity gain frequency is in GHz range. A closed loop configuration would need to be used to estimate unity gain frequency.

**TDS2024C**

TDS2024C is a 4 channel digital oscilloscope, this allows the inputs and the outputs of the Nauta structure to be monitored simultaneously. Some of the key specifications include:

- Bandwidth: 200MHz
- Sample Rate: 2 GS/s (per channel)
- 2500 point record length
- USB Interface

Both AFG3102 and TDS2024C are compatible with Virtual Instrument Software Architecture (VISA) and can be easily programmed and controlled via Python scripts. The devices can perform all the functions that can be executed using the manual front panel controls through the scripts and in some cases some of the functions can only be accessed by programming the device. For example, voltage per division scale can be set to arbitrary values instead of just the predefined values that can be set using the front panel knob.
To demonstrate the functionality of the devices and the advantages of exploiting the programmability of the said devices a function used for automating the open loop testing will be described in more detail. The purpose of the function is to for any given code find an ideal common mode input voltage (the point where output DC levels are the same for both outputs of the Nauta structure) and ensure that all the signals are displayed on the screen correctly. This entails ensuring that the scale (voltage per division) is adjusted in such a way that the signal can be seen clearly and recorded with high resolution regardless of the actual amplitude.

```plaintext
while: DC voltage of the outputs are not equal
    o-scope query: min. and max. values of +ve and -ve outputs
    compare the differences of min. and max. values
    fn-gen set: increase or decrease common mode voltage
```

The pseudo code above demonstrates some of the functionality of the both function generator and the oscilloscope that can be used simultaneously in a loop to adjust the common mode input voltage and hence obtain the most accurate measurement of the open loop gain.

```plaintext
while: waveform is not centred vertically on screen
    o-scope query: capture on screen waveform
    check: if the max. of the waveform exceeds 32000 or min. -32000
    if min and max are exceeded: reduce scale (V/div)
    if max is exceeded: reduce oscilloscope offset
    if min is exceeded: increase oscilloscope offset
```

The pseudo code above shows how it is possible to centre a wave form on a screen to capture the data with highest resolution possible. This function is primarily used to overcome the disadvantages of using the built in autoset function. The autoset function often tries to display the zero voltage level on screen if the oscilloscope channel is in DC mode. However if a signal with DC voltage of V_{DD}/2 and a relatively small amplitude needs to be displayed a lot of screen real estate is wasted and the captured data will not be suitable for precise measurements. The custom function operates by analysing the raw data provided by the oscilloscope. For each waveform the oscilloscope returns data points from -32768 to 32767 where 0 corresponds to the centre of the screen rather than returning the actual voltage levels. If we set a boundary for the waveform that is slightly lower than the maximum (for example 32000) we can detect when the waveform is likely to go off screen and return false measurement, and make appropriate scale and zero offset adjustments.
**PXI-4130, PXI-4132 and PXI-4072**

To measure the output conductance and transconductance SMUs (Source measurement Unit) - PXI-4130 and PXI-4132 were used. To conduct high resolution DC sweeps DMM (Digital Multimeter) PXI-4072 was also used. Some of the key specifications of the above modules can be seen below:

- **PXI-4072:**
  - 1uV Resolution @ 1V Range
  - 100 S/s @ 6.5 Digits/22 bits Precision

- **PXI-4130:**
  - 0.05mV Resolution, +/-10V Range (Setting)

- **PXI-4132 Output 0:**
  - 0.12mV Resolution, 0V-6V Range (Setting)

- **PXI-4132 Output 1:**
  - 0.10mV Resolution, +/-6V Range (Setting)
  - 100nA Resolution, 2mA Range (Measurement)

A different approach was needed to obtain performance measurements to overcome the disadvantages of using a digital oscilloscope. The disadvantages became more apparent when the performance measurements needed to be conducted at a large number of codes. The data transfer from the oscilloscope was found to be prohibitively slow. Calibrating multiple oscilloscope channels, especially if the signal amplitude for the previous code was significantly different could easily take over 10 seconds, where the bulk of the time is taken up by repeated transfers of the signal data points (of which ideally would be 2500). On the other hand it was found that SMUs could take approximately 180 measurements per second reliably. This in turn could allow the performance at a particular code to be consistently characterised (by measuring output conductance) in 2-3 seconds. Similarly DC sweeps could be

![Figure 91 - SMU Noise](image-url)
conducted in a reasonable amount of time with high resolution steps (of 50uV), this is significantly more precise than the 1mV steps of the signal generator. A minor issue with SMU was observed, where there could be seen a significant amount of noise at the output as seen in Figure 91, however a simple passive R-C low pass 60Hz filter was sufficient to eliminate the noise.

It was also found that the oscilloscope measurements were not suitable for high accuracy. Figure 92 shows an example of DC sweep, including backward and forwards sweep, as measured by an oscilloscope. The jagged plots indicate that there may be a significant error range for each measurement. Additionally
significant differences could be seen between forward and backward sweeps. Alternatively a DMM could be used to measure the difference between the two outputs. Figure 93 shows an example of a sweep recorded by a DMM. Although some inconsistencies still can be seen between the forward and backward sweep, it was found that on average results obtained using a DMM would provide a much “smoother” plot where backward and forward sweep tests would match a lot closer compared to the oscilloscope results. The benefits of this were especially apparent when narrow hysteresis needed to be detected.

Conclusion

Overall it is critical to reiterate the importance of choosing the best instrument for any given job. The signal generators and oscilloscopes are necessary to observe a transient response. On the other hand other measurements such as DC voltage and current measurements can be taken a lot faster with higher precision SMUs and multimeters. Additionally, the benefits of interfacing with the instruments and automating as much of the testing procedures as possible can't be understated. This is especially evident when 10s of thousands of tests need to be conducted, each involving 100s of individual measurements.
Appendix B – Source Code and Test Benches

Test Bench Schematics

Figure 94 - $\frac{gm}{gds}$ Gain Simulation (Test Bench)
Figure 95 - gm/gds Gain Simulation (Calculation)
Figure 96 - Calibration Test Bench
Verilog Code of the Calibration Modules

//Verilog HDL for "test_65_2", "nt_cal_code_reg" "code"

module nt_cal_code_reg ( ff_code_n1_out, cc_code_n1_out, sc_code_n1_out, ff_code_p1_out, cc_code_p1_out, sc_code_p1_out, ff_code_n2_out, cc_code_n2_out, sc_code_n2_out, ff_code_p2_out, cc_code_p2_out, sc_code_p2_out, reg_cont_out, clk, reset, ack_offset_corr, ack_hyst_detect, ff_code_h_n1_in, cc_code_h_n1_in, sc_code_h_n1_in, ff_code_h_p1_in, cc_code_h_p1_in, sc_code_h_p1_in, ff_code_h_n2_in, cc_code_h_n2_in, sc_code_h_n2_in, ff_code_h_p2_in, cc_code_h_p2_in, sc_code_h_p2_in, ff_code_o_n1_in, cc_code_o_n1_in, sc_code_o_n1_in, ff_code_o_p1_in, cc_code_o_p1_in, sc_code_o_p1_in, ff_code_o_n2_in, cc_code_o_n2_in, sc_code_o_n2_in, ff_code_o_p2_in, cc_code_o_p2_in, sc_code_o_p2_in, vdd, gnd);

input [1:0] sc_code_h_p1_in;
input [1:0] cc_code_o_p2_in;
input [1:0] ff_code_h_n1_in;
input [1:0] sc_code_o_n2_in;
output [9:0] cc_code_p1_out;
input [1:0] cc_code_h_p2_in;
input [1:0] ff_code_h_p1_in;
output [9:0] cc_code_p2_out;
output [9:0] sc_code_n2_out;
input vdd;
input [1:0] sc_code_o_n1_in;
input reset;
input [1:0] cc_code_o_n2_in;
input [1:0] ff_code_o_p2_in;
output [9:0] cc_code_p1_out;
output [9:0] cc_code_n1_out;
input [1:0] cc_code_o_n1_in;
output [9:0] ff_code_n1_out;
output [9:0] cc_code_n2_out;
input [1:0] cc_code_h_n1_in;
input [1:0] cc_code_o_p1_in;
output [9:0] sc_code_p2_out;
input ack_hyst_detect;
input [1:0] sc_code_h_p2_in;
input [1:0] ff_code_h_n2_in;
input clk;
output [9:0] ff_code_p2_out;
output [9:0] sc_code_p1_out;
input [1:0] ff_code_h_p2_in;
input [1:0] ff_code_p1_out;
input gnd;
input ack_offset_corr;
output reg_cont_out;
output [9:0] sc_code_o_n1_out;
input [1:0] sc_code_o_p2_in;
input [1:0] sc_code_o_p1_in;
input [1:0] ff_code_o_n2_in;
input [1:0] cc_code_h_n2_in;
input [1:0] sc_code_h_n2_in;
input [1:0] cc_code_h_n1_in;
input reg_cont;
reg [23:0] update;
reg [23:0] change;
reg [9:0] ff_code_n1;
reg [9:0] cc_code_n1;
reg [9:0] sc_code_n1;
reg [9:0] ff_code_p1;
reg [9:0] cc_code_p1;
reg [9:0] sc_code_p1;
reg [9:0] ff_code_n2;
reg [9:0] cc_code_n2;
reg [9:0] sc_code_n2;
reg [9:0] ff_code_n1_tmp;
reg [9:0] cc_code_n1_tmp;
reg [9:0] sc_code_n1_tmp;
reg [9:0] ff_code_p1_tmp;
reg [9:0] cc_code_p1_tmp;
reg [9:0] sc_code_p1_tmp;
reg [9:0] ff_code_n2_tmp;
reg [9:0] cc_code_n2_tmp;
reg [9:0] sc_code_n2_tmp;
reg [9:0] ff_code_p2_tmp;
reg [9:0] cc_code_p2_tmp;
reg [9:0] sc_code_p2_tmp;

assign ff_code_n1_out = ff_code_n1;
assign cc_code_n1_out = cc_code_n1;
assign sc_code_n1_out = sc_code_n1;

assign ff_code_p1_out = ~ff_code_p1;
assign cc_code_p1_out = ~cc_code_p1;
assign sc_code_p1_out = ~sc_code_p1;

assign ff_code_n2_out = ff_code_n2;
assign cc_code_n2_out = cc_code_n2;
assign sc_code_n2_out = sc_code_n2;

assign ff_code_p2_out = ~ff_code_p2;
assign cc_code_p2_out = ~cc_code_p2;
assign sc_code_p2_out = ~sc_code_p2;

assign reg_cont_out = reg_cont;

parameter inv_1   = 4864,
      inv_2   = 2688,
      inv_4   = 1472,
      inv_8   = 832,
      inv_16  = 462,
      inv_32  = 256,
      inv_64  = 140,
      inv_128 = 78,
      inv_256 = 43,
      inv_512 = 24,
      rollback = 320;
parameter MEM_SIZE = 10'b1111111111;
reg [13:0] mem [0:1023];

initial begin
  $readmemb("/data/TSMC65LP/IC61_v2/work/artemij/test_65_2/nt_cal_code_reg/code/mem.dat", mem);
end

always @(posedge clk) begin
  if (reset == 1'b1) begin
    ff_code_n1 <= 10'b1000100000;
    cc_code_n1 <= 10'b1001110111;
    sc_code_n1 <= 10'b1000000000;
    ff_code_p1 <= 10'b1000100000;
    cc_code_p1 <= 10'b1001110010;
    sc_code_p1 <= 10'b1000000000;
    ff_code_n2 <= 10'b1000100000;
    cc_code_n2 <= 10'b1001110111;
    sc_code_n2 <= 10'b1000000000;
    ff_code_p2 <= 10'b1000100000;
    cc_code_p2 <= 10'b1001110010;
    sc_code_p2 <= 10'b1000000000;
    ff_code_n1_tmp <= 10'b1000100000;
    cc_code_n1_tmp <= 10'b1001110111;
    sc_code_n1_tmp <= 10'b1000000000;
    ff_code_p1_tmp <= 10'b1000100000;
    cc_code_p1_tmp <= 10'b1001110010;
    sc_code_p1_tmp <= 10'b1000000000;
    ff_code_n2_tmp <= 10'b1000100000;
    cc_code_n2_tmp <= 10'b1001110010;
    sc_code_n2_tmp <= 10'b1000000000;
    ff_code_p2_tmp <= 10'b1000100000;
    cc_code_p2_tmp <= 10'b1001110111;
    sc_code_p2_tmp <= 10'b1000000000;
    reg_cont <= 1'b1;
    change <= 24'b000000000000000000000000;
    update <= 24'b111111111111111111111111;
    //needs one cycle after changes
    //but must always be ready to accept data
  end else begin
    if (reg_cont == 1'b0 && change != 24'b000000000000000000000000 && (update ^ change) ==
      24'b111111111111111111111111) begin
      reg_cont <= 1'b1;
      ff_code_n1 <= ff_code_n1_tmp;
      cc_code_n1 <= cc_code_n1_tmp;
      sc_code_n1 <= sc_code_n1_tmp;
      ff_code_p1 <= ff_code_p1_tmp;
      cc_code_p1 <= cc_code_p1_tmp;
      sc_code_p1 <= sc_code_p1_tmp;
    end
  end
end

endmodule
ff_code_n2 <= ff_code_n2_tmp;
cc_code_n2 <= cc_code_n2_tmp;
sc_code_n2 <= sc_code_n2_tmp;
ff_code_p2 <= ff_code_p2_tmp;
cc_code_p2 <= cc_code_p2_tmp;
sc_code_p2 <= sc_code_p2_tmp;
end else if (reg_cont == 1'b1 && change != 24'b000000000000000000000000 && (update ^ change) == 24'b111111111111111111111111) begin

change <= 24'b000000000000000000000000;
update <= 24'b111111111111111111111111;
ff_code_n1 <= ff_code_n1_tmp;
cc_code_n1 <= cc_code_n1_tmp;
sc_code_n1 <= sc_code_n1_tmp;
ff_code_p1 <= ff_code_p1_tmp;
cc_code_p1 <= cc_code_p1_tmp;
sc_code_p1 <= sc_code_p1_tmp;
ff_code_n2 <= ff_code_n2_tmp;
cc_code_n2 <= cc_code_n2_tmp;
sc_code_n2 <= sc_code_n2_tmp;
ff_code_p2 <= ff_code_p2_tmp;
cc_code_p2 <= cc_code_p2_tmp;
sc_code_p2 <= sc_code_p2_tmp;
end else begin

if (ff_code_h_n1_in == 2'b10 && mem[ff_code_n1] < mem[ff_code_n1_tmp+1] && update[0] == 1'b1) begin
  ff_code_n1_tmp <= ff_code_n1_tmp + 1;
  update[0] <= 1'b0;
  change[0] <= 1'b1;
end else if (ff_code_h_n1_in == 2'b10 && mem[ff_code_n1] >= mem[ff_code_n1_tmp+1] && update[0] == 1'b1) begin
  ff_code_n1_tmp <= ff_code_n1_tmp + 1;
  reg_cont <= 1'b0;
  change[0] <= 1'b1;
end else if (ff_code_h_n1_in == 2'b01 && mem[ff_code_n1] > mem[ff_code_n1_tmp-1] && update[0] == 1'b1) begin
  ff_code_n1_tmp <= ff_code_n1_tmp - 1;
  update[0] <= 1'b0;
  change[0] <= 1'b1;
end else if (ff_code_h_n1_in == 2'b01 && mem[ff_code_n1] <= mem[ff_code_n1_tmp-1] && update[0] == 1'b1) begin
  ff_code_n1_tmp <= ff_code_n1_tmp - 1;
  reg_cont <= 1'b0;
  change[0] <= 1'b1;
end else if (ff_code_h_n1_in == 2'b11 && (mem[ff_code_n1]+rollback) < mem[ff_code_n1_tmp+1] && update[0] == 1'b1) begin
  ff_code_n1_tmp <= ff_code_n1_tmp + 1;
  update[0] <= 1'b0;
  change[0] <= 1'b1;
end else if (ff_code_h_n1_in == 2'b11 && (mem[ff_code_n1]+rollback) >= mem[ff_code_n1_tmp+1] && update[0] == 1'b1) begin
  ff_code_n1_tmp <= ff_code_n1_tmp + 1;
  reg_cont <= 1'b0;
  change[0] <= 1'b1;
end else if (ff_code_h_n1_in == 2'b11 && (mem[ff_code_n1]+rollback) < mem[ff_code_n1_tmp+1] && update[0] == 1'b1) begin
  ff_code_n1_tmp <= ff_code_n1_tmp + 1;
  update[0] <= 1'b0;
  change[0] <= 1'b1;
end else if (ff_code_h_n1_in == 2'b11 && (mem[ff_code_n1]+rollback) >= mem[ff_code_n1_tmp+1] && update[0] == 1'b1) begin
  ff_code_n1_tmp <= ff_code_n1_tmp + 1;
  reg_cont <= 1'b0;
  change[0] <= 1'b1;
end else if (ff_code_h_n1_in == 2'b00) begin
    //update <= 1'b0;
end

if (cc_code_h_n1_in == 2'b10 && mem[cc_code_n1] < mem[cc_code_n1_tmp+1] && update[1] == 1'b1) begin
    cc_code_n1_tmp <= cc_code_n1_tmp + 1;
    update[1] <= 1'b0;
    change[1] <= 1'b1;
end else if (cc_code_h_n1_in == 2'b10 && mem[cc_code_n1] >= mem[cc_code_n1_tmp+1] && update[1] == 1'b1) begin
    cc_code_n1_tmp <= cc_code_n1_tmp + 1;
    reg_cont <= 1'b0;
    change[1] <= 1'b1;
end else if (cc_code_h_n1_in == 2'b01 && mem[cc_code_n1] > mem[cc_code_n1_tmp-1] && update[1] == 1'b1) begin
    cc_code_n1_tmp <= cc_code_n1_tmp - 1;
    update[1] <= 1'b0;
    change[1] <= 1'b1;
end else if (cc_code_h_n1_in == 2'b01 && mem[cc_code_n1] <= mem[cc_code_n1_tmp-1] && update[1] == 1'b1) begin
    cc_code_n1_tmp <= cc_code_n1_tmp - 1;
    reg_cont <= 1'b0;
    change[1] <= 1'b1;
end else if (cc_code_h_n1_in == 2'b00) begin
    //update1 <= 1'b0;
end

if (sc_code_h_n1_in == 2'b10 && mem[sc_code_n1] < mem[sc_code_n1_tmp+1] && update[2] == 1'b1) begin
    sc_code_n1_tmp <= sc_code_n1_tmp + 1;
    update[2] <= 1'b0;
    change[2] <= 1'b1;
end else if (sc_code_h_n1_in == 2'b10 && mem[sc_code_n1] >= mem[sc_code_n1_tmp+1] && update[2] == 1'b1) begin
    sc_code_n1_tmp <= sc_code_n1_tmp + 1;
    reg_cont <= 1'b0;
    change[2] <= 1'b1;
end else if (sc_code_h_n1_in == 2'b01 && mem[sc_code_n1] > mem[sc_code_n1_tmp-1] && update[2] == 1'b1) begin
    sc_code_n1_tmp <= sc_code_n1_tmp - 1;
    update[2] <= 1'b0;
    change[2] <= 1'b1;
end else if (sc_code_h_n1_in == 2'b01 && mem[sc_code_n1] <= mem[sc_code_n1_tmp-1] && update[2] == 1'b1) begin
    sc_code_n1_tmp <= sc_code_n1_tmp - 1;
    reg_cont <= 1'b0;
    change[2] <= 1'b1;
end else if (sc_code_h_n1_in == 2'b00) begin
    //update2 <= 1'b0;
end

    ff_code_p1_tmp <= ff_code_p1_tmp + 1;
    update[3] <= 1'b0;
    change[3] <= 1'b1;
end else if (ff_code_h_p1_in == 2'b10 && mem[ff_code_p1] >= mem[ff_code_p1_tmp+1] && update[3] == 1'b1) begin
    ff_code_p1_tmp <= ff_code_p1_tmp + 1;
    reg_cont <= 1'b0;
    change[3] <= 1'b1;
end else if (ff_code_h_p1_in == 2'b01 && mem[ff_code_p1] > mem[ff_code_p1_tmp-1] && update[3] == 1'b1) begin
    ff_code_p1_tmp <= ff_code_p1_tmp - 1;
    update[3] <= 1'b0;
    change[3] <= 1'b1;
end else if (ff_code_h_p1_in == 2'b01 && mem[ff_code_p1] <= mem[ff_code_p1_tmp-1] && update[3] == 1'b1) begin
    ff_code_p1_tmp <= ff_code_p1_tmp - 1;
    reg_cont <= 1'b0;
    change[3] <= 1'b1;
    ff_code_p1_tmp <= ff_code_p1_tmp + 1;
    update[3] <= 1'b0;
    change[3] <= 1'b1;
    ff_code_p1_tmp <= ff_code_p1_tmp + 1;
    reg_cont <= 1'b0;
    change[3] <= 1'b1;
end else if (ff_code_h_p1_in == 2'b00) begin
    //update3 <= 1'b0;
end

if (cc_code_h_p1_in == 2'b10 && mem[cc_code_p1] < mem[cc_code_p1_tmp+1] && update[4] == 1'b1) begin
    cc_code_p1_tmp <= cc_code_p1_tmp + 1;
    update[4] <= 1'b0;
    change[4] <= 1'b1;
end else if (cc_code_h_p1_in == 2'b10 && mem[cc_code_p1] >= mem[cc_code_p1_tmp+1] && update[4] == 1'b1) begin
    cc_code_p1_tmp <= cc_code_p1_tmp + 1;
    reg_cont <= 1'b0;
    change[4] <= 1'b1;
end else if (cc_code_h_p1_in == 2'b01 && mem[cc_code_p1] > mem[cc_code_p1_tmp-1] && update[4] == 1'b1) begin
    cc_code_p1_tmp <= cc_code_p1_tmp - 1;
    update[4] <= 1'b0;
    change[4] <= 1'b1;
end else if (cc_code_h_p1_in == 2'b01 && mem[cc_code_p1] <= mem[cc_code_p1_tmp-1] && update[4] == 1'b1) begin
    cc_code_p1_tmp <= cc_code_p1_tmp - 1;
    reg_cont <= 1'b0;
    change[4] <= 1'b1;
end else if (cc_code_h_p1_in == 2'b00) begin
    //update4 <= 1'b0;
end

if (sc_code_h_p1_in == 2'b10 && mem[sc_code_p1] < mem[sc_code_p1_tmp+1] && update[5] == 1'b1) begin
    sc_code_p1_tmp <= sc_code_p1_tmp + 1;
    update[5] <= 1'b0;
    change[5] <= 1'b1;
end else if (sc_code_h_p1_in == 2'b10 && mem[sc_code_p1] >= mem[sc_code_p1_tmp+1] && update[5] == 1'b1) begin
    sc_code_p1_tmp <= sc_code_p1_tmp + 1;
    reg_cont <= 1'b0;
    change[5] <= 1'b1;
end else if (sc_code_h_p1_in == 2'b01 && mem[sc_code_p1] < mem[sc_code_p1_tmp-1] && update[5] == 1'b1) begin
    sc_code_p1_tmp <= sc_code_p1_tmp - 1;
    update[5] <= 1'b0;
    change[5] <= 1'b1;
end else if (sc_code_h_p1_in == 2'b01 && mem[sc_code_p1] <= mem[sc_code_p1_tmp-1] && update[5] == 1'b1) begin
    sc_code_p1_tmp <= sc_code_p1_tmp - 1;
    reg_cont <= 1'b0;
    change[5] <= 1'b1;
end else if (sc_code_h_p1_in == 2'b00) begin
    //update5 <= 1'b0;
end

if (ff_code_h_n2_in == 2'b10 && mem[ff_code_n2] < mem[ff_code_n2_tmp+1] && update[6] == 1'b1) begin
    ff_code_n2_tmp <= ff_code_n2_tmp + 1;
end
update[6] <= 1'b0;
change[6] <= 1'b1;
end else if (ff_code_h_n2_in == 2'b10 & mem[ff_code_n2] >= mem[ff_code_n2_tmp+1] & & update[6] == 1'b1) begin
  ff_code_n2_tmp <= ff_code_n2_tmp + 1;
  reg_cont <= 1'b0;
  change[6] <= 1'b1;
end else if (ff_code_h_n2_in == 2'b01 & mem[ff_code_n2] == mem[ff_code_n2_tmp-1] & & update[6] == 1'b1) begin
  ff_code_n2_tmp <= ff_code_n2_tmp - 1;
  update[6] <= 1'b0;
  change[6] <= 1'b1;
  ff_code_n2_tmp <= ff_code_n2_tmp - 1;
  reg_cont <= 1'b0;
  change[6] <= 1'b1;
end else if (ff_code_h_n2_in == 2'b00) begin
  //update6 <= 1'b0;
end

if (cc_code_h_n2_in == 2'b10 & mem[cc_code_n2] < mem[cc_code_n2_tmp+1] & & update[7] == 1'b1) begin
  cc_code_n2_tmp <= cc_code_n2_tmp + 1;
  update[7] <= 1'b0;
  change[7] <= 1'b1;
end else if (cc_code_h_n2_in == 2'b10 & & mem[cc_code_n2] == mem[cc_code_n2_tmp-1] & & update[7] == 1'b1) begin
  cc_code_n2_tmp <= cc_code_n2_tmp - 1;
  update[7] <= 1'b0;
  change[7] <= 1'b1;
end else if (cc_code_h_n2_in == 2'b01 & mem[cc_code_n2] < mem[cc_code_n2_tmp-1] & & update[7] == 1'b1) begin
  cc_code_n2_tmp <= cc_code_n2_tmp - 1;
  reg_cont <= 1'b0;
  change[7] <= 1'b1;
end else if (cc_code_h_n2_in == 2'b00) begin
  //update7 <= 1'b0;
end

if (sc_code_h_n2_in == 2'b10 & & mem[sc_code_n2] < mem[sc_code_n2_tmp+1] & & update[8] == 1'b1) begin
  sc_code_n2_tmp <= sc_code_n2_tmp + 1;
  update[8] <= 1'b0;
  change[8] <= 1'b1;
end
end else if (sc_code_h_n2_in == 2'b10 && mem[sc_code_n2] >= mem[sc_code_n2_tmp+1] && update[8] == 1'b1) begin
    sc_code_n2_tmp <= sc_code_n2_tmp + 1;
    reg_cont <= 1'b0;
    change[8] <= 1'b1;
end else if (sc_code_h_n2_in == 2'b01 && mem[sc_code_n2] > mem[sc_code_n2_tmp-1] && update[8] == 1'b1) begin
    sc_code_n2_tmp <= sc_code_n2_tmp - 1;
    update[8] <= 1'b0;
    change[8] <= 1'b1;
end else if (sc_code_h_n2_in == 2'b01 && mem[sc_code_n2] <= mem[sc_code_n2_tmp-1] && update[8] == 1'b1) begin
    sc_code_n2_tmp <= sc_code_n2_tmp - 1;
    reg_cont <= 1'b0;
    change[8] <= 1'b1;
end else if (sc_code_h_n2_in == 2'b00) begin
    //update8 <= 1'b0;
end

    ff_code_p2_tmp <= ff_code_p2_tmp + 1;
    update[9] <= 1'b0;
    change[9] <= 1'b1;
    ff_code_p2_tmp <= ff_code_p2_tmp + 1;
    reg_cont <= 1'b0;
    change[9] <= 1'b1;
    ff_code_p2_tmp <= ff_code_p2_tmp - 1;
    update[9] <= 1'b0;
    change[9] <= 1'b1;
    ff_code_p2_tmp <= ff_code_p2_tmp - 1;
    reg_cont <= 1'b0;
    change[9] <= 1'b1;
    ff_code_p2_tmp <= ff_code_p2_tmp + 1;
    update[9] <= 1'b0;
    change[9] <= 1'b1;
    ff_code_p2_tmp <= ff_code_p2_tmp + 1;
    reg_cont <= 1'b0;
    change[9] <= 1'b1;
end else if (ff_code_h_p2_in == 2'b00) begin
    //update9 <= 1'b0;
end

    cc_code_p2_tmp <= cc_code_p2_tmp + 1;
    update[10] <= 1'b0;
    change[10] <= 1'b1;
end else if (cc_code_h_p2_in == 2'b10 && mem[cc_code_p2] >= mem[cc_code_p2_tmp+1] && update[10] == 1'b1) begin
    //update10 <= 1'b0;
end

134
cc_code_p2_tmp <= cc_code_p2_tmp + 1;
reg_cont <= 1'b0;
change[10] <= 1'b1;
end else if (cc_code_h_p2_in == 2'b01 & mem[cc_code_p2] > mem[cc_code_p2_tmp-1] & update[10] == 1'b1) begin
  cc_code_p2_tmp <= cc_code_p2_tmp - 1;
  update[10] <= 1'b0;
  change[10] <= 1'b1;
end else if (cc_code_h_p2_in == 2'b01 & mem[cc_code_p2] <= mem[cc_code_p2_tmp-1] & update[10] ==
1'b1) begin
  cc_code_p2_tmp <= cc_code_p2_tmp - 1;
  reg_cont <= 1'b0;
  change[10] <= 1'b1;
end else if (cc_code_h_p2_in == 2'b00) begin
  //update10 <= 1'b0;
end

  sc_code_p2_tmp <= sc_code_p2_tmp + 1;
  update[11] <= 1'b0;
  change[11] <= 1'b1;
  sc_code_p2_tmp <= sc_code_p2_tmp + 1;
  reg_cont <= 1'b0;
  change[11] <= 1'b1;
1'b1) begin
  sc_code_p2_tmp <= sc_code_p2_tmp - 1;
  update[11] <= 1'b0;
  change[11] <= 1'b1;
1'b1) begin
  sc_code_p2_tmp <= sc_code_p2_tmp - 1;
  reg_cont <= 1'b0;
  change[11] <= 1'b1;
end else if (sc_code_h_p2_in == 2'b00) begin
  //update11 <= 1'b0;
end

if (ff_code_o_n1_in == 2'b10 & mem[ff_code_n1] < mem[ff_code_n1_tmp+1] & update[12] == 1'b1) begin
  ff_code_n1_tmp <= ff_code_n1_tmp + 1;
  update[12] <= 1'b0;
  change[12] <= 1'b1;
end else if (ff_code_o_n1_in == 2'b10 & mem[ff_code_n1] >= mem[ff_code_n1_tmp+1] & update[12] ==
1'b1) begin
  ff_code_n1_tmp <= ff_code_n1_tmp + 1;
  reg_cont <= 1'b0;
  change[12] <= 1'b1;
end else if (ff_code_o_n1_in == 2'b01 & mem[ff_code_n1] > mem[ff_code_n1_tmp-1] & update[12] ==
1'b1) begin
  ff_code_n1_tmp <= ff_code_n1_tmp - 1;
  update[12] <= 1'b0;
  change[12] <= 1'b1;
end else if (ff_code_o_n1_in == 2'b01 & mem[ff_code_n1] <= mem[ff_code_n1_tmp-1] & update[12] ==
1'b1) begin
  ff_code_n1_tmp <= ff_code_n1_tmp - 1;
  reg_cont <= 1'b0;
  change[12] <= 1'b1;
end else if (ff_code_o_n1_in == 2'b00) begin
  //update12 <= 1'b0;
end
if (cc_code_o_n1_in == 2'b10 && mem[cc_code_n1] < mem[cc_code_n1_tmp+1] && update[13] == 1'b1)
begin
  cc_code_n1_tmp <= cc_code_n1_tmp + 1;
  update[13] <= 1'b0;
  change[13] <= 1'b1;
end else if (cc_code_o_n1_in == 2'b10 && mem[cc_code_n1] >= mem[cc_code_n1_tmp+1] && update[13] == 1'b1)
begin
  cc_code_n1_tmp <= cc_code_n1_tmp + 1;
  reg_cont <= 1'b0;
  change[13] <= 1'b1;
end else if (cc_code_o_n1_in == 2'b01 && mem[cc_code_n1] > mem[cc_code_n1_tmp-1] && update[13] == 1'b1)
begin
  cc_code_n1_tmp <= cc_code_n1_tmp - 1;
  update[13] <= 1'b0;
  change[13] <= 1'b1;
end else if (cc_code_o_n1_in == 2'b01 && mem[cc_code_n1] <= mem[cc_code_n1_tmp-1] && update[13] == 1'b1)
begin
  cc_code_n1_tmp <= cc_code_n1_tmp - 1;
  reg_cont <= 1'b0;
  change[13] <= 1'b1;
end else if (cc_code_o_n1_in == 2'b00)
begin
  //update13 <= 1'b0;
end

if (sc_code_o_n1_in == 2'b10 && mem[sc_code_n1] < mem[sc_code_n1_tmp+1] && update[14] == 1'b1)
begin
  sc_code_n1_tmp <= sc_code_n1_tmp + 1;
  update[14] <= 1'b0;
  change[14] <= 1'b1;
end else if (sc_code_o_n1_in == 2'b10 && mem[sc_code_n1] >= mem[sc_code_n1_tmp+1] && update[14] == 1'b1)
begin
  sc_code_n1_tmp <= sc_code_n1_tmp + 1;
  reg_cont <= 1'b0;
  change[14] <= 1'b1;
end else if (sc_code_o_n1_in == 2'b01 && mem[sc_code_n1] > mem[sc_code_n1_tmp-1] && update[14] == 1'b1)
begin
  sc_code_n1_tmp <= sc_code_n1_tmp - 1;
  update[14] <= 1'b0;
  change[14] <= 1'b1;
end else if (sc_code_o_n1_in == 2'b01 && mem[sc_code_n1] <= mem[sc_code_n1_tmp-1] && update[14] == 1'b1)
begin
  sc_code_n1_tmp <= sc_code_n1_tmp - 1;
  reg_cont <= 1'b0;
  change[14] <= 1'b1;
end else if (sc_code_o_n1_in == 2'b00)
begin
  //update14 <= 1'b0;
end

begin
  ff_code_p1_tmp <= ff_code_p1_tmp + 1;
  update[15] <= 1'b0;
  change[15] <= 1'b1;
begin
  ff_code_p1_tmp <= ff_code_p1_tmp + 1;
  reg_cont <= 1'b0;
  change[15] <= 1'b1;
begin
  ff_code_p1_tmp <= ff_code_p1_tmp - 1;
  update[15] <= 1'b0;
  change[15] <= 1'b1;
begin
  ff_code_p1_tmp <= ff_code_p1_tmp - 1;
  //update15 <= 1'b0;
end

136
reg_cont <= 1'b0;
change[15] <= 1'b1;
end else if (if_code_o_p1_in == 2'b00) begin
  //update15 <= 1'b0;
end

if(cc_code_o_p1_in == 2'b10 && mem[cc_code_p1] < mem[cc_code_p1_tmp+1] && update[16] == 1'b1) begin
  cc_code_p1_tmp <= cc_code_p1_tmp + 1;
  update[16] <= 1'b0;
  change[16] <= 1'b1;
end else if (cc_code_o_p1_in == 2'b10 && mem[cc_code_p1] >= mem[cc_code_p1_tmp+1] && update[16] ==
1'b1) begin
  cc_code_p1_tmp <= cc_code_p1_tmp + 1;
  reg_cont <= 1'b0;
  change[16] <= 1'b1;
end else if (cc_code_o_p1_in == 2'b01 && mem[cc_code_p1] > mem[cc_code_p1_tmp-1] && update[16] ==
1'b1) begin
  cc_code_p1_tmp <= cc_code_p1_tmp - 1;
  update[16] <= 1'b0;
  change[16] <= 1'b1;
end else if (cc_code_o_p1_in == 2'b01 && mem[cc_code_p1] <= mem[cc_code_p1_tmp-1] && update[16] ==
1'b1) begin
  cc_code_p1_tmp <= cc_code_p1_tmp - 1;
  reg_cont <= 1'b0;
  change[16] <= 1'b1;
end else if (cc_code_o_p1_in == 2'b00) begin
  //update16 <= 1'b0;
end

if (sc_code_o_p1_in == 2'b10 && mem[sc_code_p1] < mem[sc_code_p1_tmp+1] && update[17] == 1'b1) begin
  sc_code_p1_tmp <= sc_code_p1_tmp + 1;
  update[17] <= 1'b0;
  change[17] <= 1'b1;
end else if (sc_code_o_p1_in == 2'b10 && mem[sc_code_p1] >= mem[sc_code_p1_tmp+1] && update[17] ==
1'b1) begin
  sc_code_p1_tmp <= sc_code_p1_tmp + 1;
  reg_cont <= 1'b0;
  change[17] <= 1'b1;
end else if (sc_code_o_p1_in == 2'b01 && mem[sc_code_p1] < mem[sc_code_p1_tmp-1] && update[17] ==
1'b1) begin
  sc_code_p1_tmp <= sc_code_p1_tmp - 1;
  update[17] <= 1'b0;
  change[17] <= 1'b1;
end else if (sc_code_o_p1_in == 2'b01 && mem[sc_code_p1] <= mem[sc_code_p1_tmp-1] && update[17] ==
1'b1) begin
  sc_code_p1_tmp <= sc_code_p1_tmp - 1;
  reg_cont <= 1'b0;
  change[17] <= 1'b1;
end else if (sc_code_o_p1_in == 2'b00) begin
  //update17 <= 1'b0;
end

if (ff_code_o_n2_in == 2'b10 && mem[ff_code_n2] < mem[ff_code_n2_tmp+1] && update[18] == 1'b1) begin
  ff_code_n2_tmp <= ff_code_n2_tmp + 1;
  update[18] <= 1'b0;
  change[18] <= 1'b1;
end else if (ff_code_o_n2_in == 2'b10 && mem[ff_code_n2] >= mem[ff_code_n2_tmp+1] && update[18] ==
1'b1) begin
  ff_code_n2_tmp <= ff_code_n2_tmp + 1;
  reg_cont <= 1'b0;
  change[0] <= 1'b1;
end

if (ff_code_o_n2_in == 2'b00 && mem[ff_code_n2] < mem[ff_code_n2_tmp+1] && update[18] == 1'b1) begin
  ff_code_n2_tmp <= ff_code_n2_tmp + 1;
  update[18] <= 1'b0;
  change[18] <= 1'b1;
end else if (ff_code_o_n2_in == 2'b10 && mem[ff_code_n2] >= mem[ff_code_n2_tmp+1] && update[18] ==
1'b1) begin
  ff_code_n2_tmp <= ff_code_n2_tmp + 1;
  reg_cont <= 1'b0;
  change[0] <= 1'b1;
end else if (ff_code_o_n2_in == 2'b01 && mem[ff_code_n2] > mem[ff_code_n2_tmp] && update[18] == 1'b1) begin
    ff_code_n2_tmp <= ff_code_n2_tmp - 1;
    update[18] <= 1'b0;
    change[18] <= 1'b1;
end else if (ff_code_o_n2_in == 2'b01 && mem[ff_code_n2] <= mem[ff_code_n2_tmp] && update[18] == 1'b1) begin
    ff_code_n2_tmp <= ff_code_n2_tmp - 1;
    reg_cont <= 1'b0;
    change[18] <= 1'b1;
end else if (ff_code_o_n2_in == 2'b00) begin
    //update18 <= 1'b0;
end

if (cc_code_o_n2_in == 2'b10 && mem[cc_code_n2] < mem[cc_code_n2_tmp+1] && update[19] == 1'b1) begin
    cc_code_n2_tmp <= cc_code_n2_tmp + 1;
    update[19] <= 1'b0;
    change[19] <= 1'b1;
end else if (cc_code_o_n2_in == 2'b10 && mem[cc_code_n2] >= mem[cc_code_n2_tmp+1] && update[19] == 1'b1) begin
    cc_code_n2_tmp <= cc_code_n2_tmp + 1;
    reg_cont <= 1'b0;
    change[19] <= 1'b1;
end else if (cc_code_o_n2_in == 2'b01 && mem[cc_code_n2] > mem[cc_code_n2_tmp-1] && update[19] == 1'b1) begin
    cc_code_n2_tmp <= cc_code_n2_tmp - 1;
    update[19] <= 1'b0;
    change[19] <= 1'b1;
end else if (cc_code_o_n2_in == 2'b01 && mem[cc_code_n2] <= mem[cc_code_n2_tmp-1] && update[19] == 1'b1) begin
    cc_code_n2_tmp <= cc_code_n2_tmp - 1;
    reg_cont <= 1'b0;
    change[19] <= 1'b1;
end else if (cc_code_o_n2_in == 2'b00) begin
    //update19 <= 1'b0;
end

if (sc_code_o_n2_in == 2'b10 && mem[sc_code_n2] < mem[sc_code_n2_tmp+1] && update[20] == 1'b1) begin
    sc_code_n2_tmp <= sc_code_n2_tmp + 1;
    update[20] <= 1'b0;
    change[20] <= 1'b1;
end else if (sc_code_o_n2_in == 2'b10 && mem[sc_code_n2] >= mem[sc_code_n2_tmp+1] && update[20] == 1'b1) begin
    sc_code_n2_tmp <= sc_code_n2_tmp + 1;
    reg_cont <= 1'b0;
    change[20] <= 1'b1;
end else if (sc_code_o_n2_in == 2'b01 && mem[sc_code_n2] > mem[sc_code_n2_tmp-1] && update[20] == 1'b1) begin
    sc_code_n2_tmp <= sc_code_n2_tmp - 1;
    update[20] <= 1'b0;
    change[20] <= 1'b1;
end else if (sc_code_o_n2_in == 2'b01 && mem[sc_code_n2] <= mem[sc_code_n2_tmp-1] && update[20] == 1'b1) begin
    sc_code_n2_tmp <= sc_code_n2_tmp - 1;
    reg_cont <= 1'b0;
    change[20] <= 1'b1;
end else if (sc_code_o_n2_in == 2'b00) begin
    //update20 <= 1'b0;
end

    ff_code_p2_tmp <= ff_code_p2_tmp + 1;
end
begin
    ff_code_p2_tmp <= ff_code_p2_tmp + 1;
    reg_cont <= 1'b0;
    change[21] <= 1'b1;
    ff_code_p2_tmp <= ff_code_p2_tmp - 1;
    update[21] <= 1'b0;
    change[21] <= 1'b1;
  end else if (ff_code_o_p2_in == 2'b00) begin
    //update21 <= 1'b0;
  end
  if (cc_code_o_p2_in == 2'b10 && mem[cc_code_p2] < mem[cc_code_p2_tmp+1] && update[22] == 1'b1) begin
    cc_code_p2_tmp <= cc_code_p2_tmp + 1;
    update[22] <= 1'b0;
    change[22] <= 1'b1;
  end else if (cc_code_o_p2_in == 2'b01 && mem[cc_code_p2] >= mem[cc_code_p2_tmp-1] && update[22] == 1'b1) begin
    cc_code_p2_tmp <= cc_code_p2_tmp - 1;
    update[22] <= 1'b0;
    change[22] <= 1'b1;
  end else if (cc_code_o_p2_in == 2'b00) begin
    //update22 <= 1'b0;
  end
  if (sc_code_o_p2_in == 2'b10 && mem[sc_code_p2] < mem[sc_code_p2_tmp+1] && update[23] == 1'b1) begin
    sc_code_p2_tmp <= sc_code_p2_tmp + 1;
    update[23] <= 1'b0;
    change[23] <= 1'b1;
  end else if (sc_code_o_p2_in == 2'b01 && mem[sc_code_p2] >= mem[sc_code_p2_tmp-1] && update[23] == 1'b1) begin
    sc_code_p2_tmp <= sc_code_p2_tmp - 1;
    update[23] <= 1'b0;
    change[23] <= 1'b1;
  end else if (sc_code_o_p2_in == 2'b00) begin
    //update23 <= 1'b0;
module nt_cal_hyst_detect ( ack, stop, ff_mode, ff_code_n1_out, cc_code_n1_out,
sc_code_n1_out, ff_code_n2_out, cc_code_n2_out, sc_code_n2_out, ff_code_p1_out,
cc_code_p1_out, sc_code_p1_out, ff_code_p2_out, cc_code_p2_out, sc_code_p2_out,
clk, state, reset, vout_cmp, reg_cont, vdd, gnd, ref_ctrl_out );

input state;
input reg_cont;
input gnd;
output [1:0] sc_code_p2_out;
input clk;
output [1:0] sc_code_n1_out;
output [1:0] cc_code_p1_out;
output stop;
output [1:0] sc_code_n2_out;
output [1:0] cc_code_p2_out;
output [1:0] cc_code_n2_out;
output [1:0] sc_code_p1_out;
output [1:0] ff_code_p2_out;
output [1:0] cc_code_n1_out;
input vout_cmp;
input vdd;
output ff_mode;
output [1:0] ff_code_p1_out;
output ack;
input reset;
output [1:0] ff_code_n2_out;
output [1:0] ff_code_n1_out;
output [4:0] ref_ctrl_out;

reg ack_reg;
reg ack_reg_flag;
reg stop_reg;
reg stop_reg_flag;

reg VCMFm_reg;
reg VCMF_reg;
reg VCMFp_reg;
reg VCMBp_reg;
reg VCMB_reg;
reg VCMBm_reg;

reg cmpm;
reg cmp;
reg cmpp;

reg [1:0] ff_code_n1;
reg [1:0] cc_code_n1;
reg [1:0] sc_code_n1;

reg [1:0] ff_code_p1;
reg [1:0] cc_code_p1;
reg [1:0] sc_code_p1;

endmodule

//Verilog HDL for "test_65_2", "nt_cal_hyst_detect" "code"

// Registers and assignments to corresponding wires
reg [1:0] ff_code_n2;
reg [1:0] cc_code_n2;
reg [1:0] sc_code_n2;
reg [1:0] ff_code_p2;
reg [1:0] cc_code_p2;
reg [1:0] sc_code_p2;
reg [1:0] n_shift_ccsc;
reg [1:0] p_shift_ccsc;
reg n_shift_ccsc_wh; //wide hysteresis counters
reg p_shift_ccsc_wh;
reg ff_mode_reg;
reg n_shift_ff;
reg p_shift_ff;
reg wait_reg;
reg [4:0] ref_ctrl;

assign ack = ack_reg;
assign stop = stop_reg;
assign ff_mode = ff_mode_reg;
assign ff_code_n1_out = ff_code_n1;
assign cc_code_n1_out = cc_code_n1;
assign sc_code_n1_out = sc_code_n1;
assign ff_code_p1_out = ff_code_p1;
assign cc_code_p1_out = cc_code_p1;
assign sc_code_p1_out = sc_code_p1;
assign ff_code_n2_out = ff_code_n2;
assign cc_code_n2_out = cc_code_n2;
assign sc_code_n2_out = sc_code_n2;
assign ff_code_p2_out = ff_code_p2;
assign cc_code_p2_out = cc_code_p2;
assign sc_code_p2_out = sc_code_p2;
assign ref_ctrl_out = ref_ctrl;

parameter OFF = 0,
ON = 1;
parameter idle = 0, //set to gnd
set_gnd = 1,
set_vcmfm = 2,  //set to vcm-
VCMFm = 3,
set_vcmf = 4,  //set to vcm
VCMF = 5,
set_vcmfp = 6,  //set to vcm+
VCMFp = 7,
set_vdd = 8,  //set to vdd
set_vcmbp = 9,  //set to vcm+
VCMBp = 10,
set_vcmb = 11,  //set to vcm
VCMB = 12,
set_vcmbm = 13, //set to vcm-
VCMBm = 14,
calc = 15,
set_sccc = 16,
set_ff = 17,
update = 18;

integer var;

always @(posedge clk) begin
    if (reset == 1'b1) begin
        ack_reg_flag <= 1'b0;
        ack_reg <= 1'b0;
        stop_reg <= 1'b0;
        stop_reg_flag <= 1'b0;
        var <= idle;
        n_shift_ccsc <= 2'b00;
        p_shift_ccsc <= 2'b00;
        n_shift_ccsc_wh <= 1'b0;
        p_shift_ccsc_wh <= 1'b0;
        ff_mode_reg <= 1'b0;
        n_shift_ff <= 1'b0;
        p_shift_ff <= 1'b0;
        ff_code_n1 <= 2'b00;
        cc_code_n1 <= 2'b00;
        sc_code_n1 <= 2'b00;
        ff_code_p1 <= 2'b00;
        cc_code_p1 <= 2'b00;
        sc_code_p1 <= 2'b00;
        ff_code_n2 <= 2'b00;
        cc_code_n2 <= 2'b00;
        sc_code_n2 <= 2'b00;
        ff_code_p2 <= 2'b00;
        cc_code_p2 <= 2'b00;
        sc_code_p2 <= 2'b00;
        wait_reg <= 1'b0;
        ref_ctrl <= 5'b00000;
    end else begin
        if (state == ON && reg_cont == 1'b1) begin
            case (var)
                idle: begin
                    ff_code_n1 <= 2'b00;
                    cc_code_n1 <= 2'b00;
                    sc_code_n1 <= 2'b00;
                    ff_code_p1 <= 2'b00;
                    cc_code_p1 <= 2'b00;
                    sc_code_p1 <= 2'b00;
                    ff_code_n2 <= 2'b00;
                    cc_code_n2 <= 2'b00;
                    sc_code_n2 <= 2'b00;
                    ff_code_p2 <= 2'b00;
                    cc_code_p2 <= 2'b00;
                    sc_code_p2 <= 2'b00;
                    cmpp <= 1'b0;
                    cmp <= 1'b0;
                    cmpm <= 1'b0;
                end
            endcase
        end
    end
end
var <= set_gnd;
ref_ctrl <= 5'b00000;
end

set_gnd: begin
ref_ctrl <= 5'b00001;
var <= set_vcmfm;
end

set_vcmfm: begin
ref_ctrl <= 5'b00010;
var <= VCMFm;
end

VCMFm: begin
VCMFm_reg <= vout_cmp;
var <= set_vcmf;
end

set_vcmf: begin
ref_ctrl <= 5'b00100;
var <= VCMF;
end

VCMF: begin
VCMF_reg <= vout_cmp;
var <= set_vcmfp;
end

set_vcmfp: begin
ref_ctrl <= 5'b01000;
var <= VCMFp;
end

VCMFp: begin
VCMFp_reg <= vout_cmp;
var <= set_vdd;
end

set_vdd: begin
ref_ctrl <= 5'b10000;
var <= set_vcmbp;
end
end

set_vcmbp: begin
    ref_ctrl <= 5'b01000;
    var <= VCMBP;
end

VCMBp: begin
    VCMBP_reg <= vout_cmp;
    var <= set_vcmb;
end

set_vcmb: begin
    ref_ctrl <= 5'b00100;
    var <= VCMB;
end

VCMB: begin
    VCMB_reg <= vout_cmp;
    var <= set_vcmbm;
end

set_vcmbm: begin
    ref_ctrl <= 5'b00010;
    var <= VCMBm;
end

VCMBm: begin
    VCMBM_reg <= vout_cmp;
    var <= calc;
end

calc: begin
    /*
     * if (VCMFP != VCMBP) begin
     *     cmpp <= 1'b1;
     *     end
     *
     * if (VCMF != VCMB) begin
     *     cmp <= 1'b1;
     *     end
     *
     * if (VCMFM != VCMBM) begin
     *     cmpm <= 1'b1;
     *     end
     *
    */
    if (VCMFP_reg != VCMBP_reg) begin
        cmpp <= 1'b1;
    end

    if (VCMF_reg != VCMB_reg) begin
        cmp <= 1'b1;
    end

    if (VCMFM_reg != VCMBM_reg) begin
        cmpm <= 1'b1;
    end

end /*calc*/
if (VCMFm_reg != VCMBm_reg) begin
  cmpm <= 1'b1;
end

if (ff_mode_reg == 1'b0) begin
  var <= set_sccc;
end

if (ff_mode_reg == 1'b1) begin
  var <= set_ff;
end
end

set_sccc: begin

if (cmpm == 1'b0 && cmp == 1'b0 && cmpp == 1'b0) begin
  //Check for offset (Stage II)
  ack_reg_flag <= 1'b1;
end else if (cmpm == 1'b0 && cmp == 1'b0 && cmpp == 1'b1) begin
  //Decrease positive offset <=================
  //if +ve offset: increase cc1/sc1 nmos decrease cc2/sc2 nmos (negative shift)
  if (n_shift_ccsc_wh == 1'b0) begin
    cc_code_n2 <= 2'b01;
  end else if (n_shift_ccsc_wh == 1'b1) begin
    sc_code_n1 <= 2'b10;
  end
  n_shift_ccsc_wh <= n_shift_ccsc_wh + 1'b1;
end else if (cmpm == 1'b0 && cmp == 1'b1 && cmpp == 1'b0) begin
  //Decrease CC array size, maximum gain point found
  //stop_reg <= 1'b1;
  ff_mode_reg <= 1'b1;
  cc_code_n1 <= 2'b01;
  cc_code_p1 <= 2'b01;
  cc_code_n2 <= 2'b01;
  cc_code_p2 <= 2'b01;
end else if (cmpm == 1'b0 && cmp == 1'b1 && cmpp == 1'b1) begin
  //Decrease CC (gain) array size and decrease positive offset
  //Increase sc1n / Decrease cc2n
  if (n_shift_ccsc_wh == 1'b0) begin
    cc_code_n2 <= 2'b01;
  end else if (n_shift_ccsc_wh == 1'b1) begin
    sc_code_n1 <= 2'b10;
  end
  n_shift_ccsc_wh <= n_shift_ccsc_wh + 1'b1;
end else if (cmpm == 1'b1 && cmp == 1'b0 && cmpp == 1'b0) begin

end
//Decrease negative offset  
//if -ve offset: decrease cc1/sc1 nmos increase cc2/sc2 nmos (positive shift)

if (p_shift_ccsc_wh == 1'b0) begin
    cc_code_n1 <= 2'b01;
end else if (p_shift_ccsc_wh == 1'b1) begin
    sc_code_n2 <= 2'b10;
end

p_shift_ccsc_wh <= p_shift_ccsc_wh + 1'b1;

end else if (cmpm == 1'b1 && cmp == 1'b0 && cmpp == 1'b1) begin
    //Impossible condition
end else if (cmpm == 1'b1 && cmp == 1'b1 && cmpp == 1'b0) begin
    //Decrease CC array size and decrease negative offset
    //Increase sc2n / Decrease cc1n
    if (p_shift_ccsc_wh == 1'b0) begin
        cc_code_n1 <= 2'b01;
    end else if (p_shift_ccsc_wh == 1'b1) begin
        sc_code_n2 <= 2'b10;
    end

    p_shift_ccsc_wh <= p_shift_ccsc_wh + 1'b1;

end else if (cmpm == 1'b1 && cmp == 1'b1 && cmpp == 1'b1) begin
    //Decrease CC array size
    cc_code_n1 <= 2'b01;
    cc_code_p1 <= 2'b01;
    cc_code_n2 <= 2'b01;
    cc_code_p2 <= 2'b01;
end

var <= update;
end

set_ff: begin
    //ff adjustements
    if    (cmpm == 1'b0 && cmp == 1'b0 && cmpp == 1'b0) begin
        //Check for offset (Stage II)
        ack_reg_flag <= 1'b1;
    end else if (cmpm == 1'b0 && cmp == 1'b0 && cmpp == 1'b1) begin
        //Decrease positive offset
        //if +ve offset: increase ff1 nmos decrease ff2 nmos (negative shift)
        end else if (cmpm == 1'b0 && cmp == 1'b1 && cmpp == 1'b0) begin

//Increase FF array size, maximum gain point found
stop_reg_flag <= 1'b1;

ff_code_n1 <= 2'b11;
ff_code_p1  <= 2'b11;
ff_code_n2  <= 2'b11;
ff_code_p2  <= 2'b11;

end else if (cmpm == 1'b0 && cmp == 1'b1 && cmpp == 1'b1) begin

//Increase FF (gain) array size and decrease positive offset
//Increase ff1n

ff_code_n1 <= 2'b10;

end else if (cmpm == 1'b1 && cmp == 1'b0 && cmpp == 1'b0) begin

//Decrease negative offset
//if -ve offset: decrease f1 nmos increase ff2 nmos (positive shift)

p_shift_ff <= p_shift_ff + 1'b1;

end else if (cmpm == 1'b1 && cmp == 1'b0 && cmpp == 1'b1) begin

//Impossible condition
end else if (cmpm == 1'b1 && cmp == 1'b1 && cmpp == 1'b0) begin

//Increase FF array size and decrease negative offset
//Increase ff2n

ff_code_n2 <= 2'b10;

end else if (cmpm == 1'b1 && cmp == 1'b1 && cmpp == 1'b1) begin

//Increase FF array size

ff_code_n1 <= 2'b10;
ff_code_p1 <= 2'b10;
ff_code_n2 <= 2'b10;
ff_code_p2 <= 2'b10;

end

var <= update;

end

update: begin

if (ack_reg_flag == 1'b1) begin
    ack_reg <= 1'b1;
end

if (stop_reg_flag == 1'b1) begin
    stop_reg <= 1'b1;
end

var <= idle;

end
end
endcase

// set code

end else if (state == OFF) begin

// no action
ref_ctrl <= 5'b00000;
ack_reg_flag <= 1'b0;
ack_reg <= 1'b0;
stop_reg <= 1'b0;
stop_reg_flag <= 1'b0;
ff_code_n1 <= 2'b00;
cc_code_n1 <= 2'b00;
sc_code_n1 <= 2'b00;
ff_code_p1 <= 2'b00;
cc_code_p1 <= 2'b00;
sc_code_p1 <= 2'b00;
ff_code_n2 <= 2'b00;
cc_code_n2 <= 2'b00;
sc_code_n2 <= 2'b00;
ff_code_p2 <= 2'b00;
cc_code_p2 <= 2'b00;
sc_code_p2 <= 2'b00;

end
end
end
module nt_cal_offset_corr (ack, ff_code_n1_out, cc_code_n1_out, sc_code_n1_out, 
ff_code_n2_out, cc_code_n2_out, sc_code_n2_out, ff_code_p1_out, cc_code_p1_out, 
sc_code_p1_out, ff_code_p2_out, cc_code_p2_out, sc_code_p2_out, clk, state, 
reset, vout_cmp, reg_cont, ff_mode, gnd, vdd, ref_ctrl_out);

input state;
input reg_cont;
input gnd;
output [1:0] sc_code_p2_out;
input clk;
output [1:0] sc_code_n1_out;
output [1:0] cc_code_p1_out;
output [1:0] sc_code_n2_out;
output [1:0] cc_code_p2_out;
output [1:0] cc_code_n2_out;
output [1:0] cc_code_n1_out;
input vout_cmp;
input vdd;
input ff_mode;
output [1:0] ff_code_p1_out;
output ack;
input reset;
output [1:0] ff_code_n2_out;
output [1:0] ff_code_n1_out;
output [2:0] ref_ctrl_out;

reg ack_reg;
reg ack_reg_flag; // Registers and assignments to corresponding wires

reg cmpm;
reg cmpp;

reg [9:0] ff_code_n1;
reg [9:0] cc_code_n1;
reg [9:0] sc_code_n1;

reg [9:0] ff_code_p1;
reg [9:0] cc_code_p1;
reg [9:0] sc_code_p1;

reg [9:0] ff_code_n2;
reg [9:0] cc_code_n2;
reg [9:0] sc_code_n2;

reg [9:0] ff_code_p2;
reg [9:0] cc_code_p2;
reg [9:0] sc_code_p2;

reg [1:0] n_shift_ccsc;
reg [1:0] p_shift_ccsc;

reg n_shift_ff;
reg p_shift_ff;
reg shift_ccsc;

reg [2:0] ref_ctrl;

assign ack = ack_reg;
assign ff_code_n1_out = ff_code_n1;
assign cc_code_n1_out = cc_code_n1;
assign sc_code_n1_out = sc_code_n1;
assign ff_code_p1_out = ff_code_p1;
assign cc_code_p1_out = cc_code_p1;
assign sc_code_p1_out = sc_code_p1;
assign ff_code_n2_out = ff_code_n2;
assign cc_code_n2_out = cc_code_n2;
assign sc_code_n2_out = sc_code_n2;
assign ff_code_p2_out = ff_code_p2;
assign cc_code_p2_out = cc_code_p2;
assign sc_code_p2_out = sc_code_p2;
assign ref_ctrl_out = ref_ctrl;

parameter OFF = 0,
ON = 1;
parameter idle = 0,
set_gnd = 1, //set to gnd
set_vcmn = 2, //set to vcm-
VCMn = 3,
set_vcmpp = 4, //set to vcm+
VCMp = 5,
set_sccc = 6,
set_ff = 7,
update = 8;

integer var;
always @(posedge clk) begin
if (reset == 1'b1) begin
ack_reg <= 1'b0;
ack_reg_flag <= 1'b0;
var <= idle;
  n_shift_ccsc <= 2'b00;
p_shift_ccsc <= 2'b00;
n_shift_ff <= 1'b0;
p_shift_ff <= 1'b0;
shift_ccsc <= 1'b0;
  ff_code_n1 <= 2'b00;
c_code_n1 <= 2'b00;
sc_code_n1 <= 2'b00;
ff_code_p1 <= 2'b00;
cc_code_p1 <= 2'b00;
sc_code_p1 <= 2'b00;
ff_code_n2 <= 2'b00;
c_code_n2 <= 2'b00;
sc_code_n2 <= 2'b00;
ff_code_p2 <= 2'b00;
c_code_p2 <= 2'b00;
sc_code_p2 <= 2'b00;
ref_ctrl <= 3'b000;
end else begin
if (state == ON && reg_cont == 1'b1) begin // && ack_reg == 1'b0) begin
  case (var)
    idle: begin

end

end

150
ff_code_n1 <= 2'b00;
cc_code_n1 <= 2'b00;
sc_code_n1 <= 2'b00;
ff_code_p1 <= 2'b00;
cc_code_p1 <= 2'b00;
sc_code_p1 <= 2'b00;
ff_code_n2 <= 2'b00;
cc_code_n2 <= 2'b00;
sc_code_n2 <= 2'b00;
ff_code_p2 <= 2'b00;
cc_code_p2 <= 2'b00;
sc_code_p2 <= 2'b00;

var <= set_gnd;
ref_ctrl <= 3'b000;
end

set_gnd: begin
ref_ctrl <= 3'b001;
var <= set_vcmm;
end

set_vcmm: begin
ref_ctrl <= 3'b010;
var <= VCMm;
end

VCMm: begin
cmpm <= vout_cmp;
var <= set_vcmp;
end

set_vcmp: begin
ref_ctrl <= 3'b100;
var <= VCMp;
end

VCMp: begin
cmpp <= vout_cmp;
if (ff_mode == 1'b0) begin
var <= set_sccc;
end
if (ff_mode == 1'b1) begin
var <= set_ff;
end
end
set_sccc: begin

if (cmpm != cmpp) begin

//ack_reg <= 1'b1;

//Increase cc

if (shift_ccsc == 2'b00) begin
    cc_code_n1 <= 2'b10;
    cc_code_p1 <= 2'b10;
end else if (shift_ccsc == 2'b01) begin
    cc_code_n2 <= 2'b10;
    cc_code_p2 <= 2'b10;
end

shift_ccsc <= shift_ccsc + 1'b1;
end

if (cmpm == 1'b1 && cmpp == 1'b1) begin
//decrease possitive offset

if (n_shift_ccsc == 2'b00) begin
    cc_code_n1 <= 2'b10;
end else if (n_shift_ccsc == 2'b01) begin
    sc_code_n1 <= 2'b10;
end else if (n_shift_ccsc == 2'b10) begin
    cc_code_n2 <= 2'b10;
end else if (n_shift_ccsc == 2'b11) begin
    sc_code_n2 <= 2'b10;
end

n_shift_ccsc <= n_shift_ccsc + 1'b1;
end

if (cmpm == 1'b0 && cmpp == 1'b0) begin
//decrease negative offset

if (p_shift_ccsc == 2'b00) begin
    cc_code_n1 <= 2'b01;
end else if (p_shift_ccsc == 2'b01) begin
    sc_code_n1 <= 2'b01;
end else if (p_shift_ccsc == 2'b10) begin
    cc_code_n2 <= 2'b01;
end else if (p_shift_ccsc == 2'b11) begin
    sc_code_n2 <= 2'b01;
end

p_shift_ccsc <= p_shift_ccsc + 1'b1;
end

ack_reg_flag <= 1'b1;
//ack_reg <= 1'b1;
var <= update;
end

set_ff: begin

if (cmpm != cmpp) begin

end
//ack_reg <= 1'b1;

//Decrease FF
ff_code_n1 <= 2'b01;
ff_code_p1 <= 2'b01;
ff_code_n2 <= 2'b01;
ff_code_p2 <= 2'b01;

end

if (cmpm == 1'b1 == cmpp == 1'b1) begin

//decrease positive offset
if (n_shift_ff == 1'b0) begin
    ff_code_n1 <= 2'b10;
end else if (n_shift_ff == 1'b1) begin
    ff_code_n2 <= 2'b01;
end

    n_shift_ff <= n_shift_ff + 1'b1;
end

if (cmpm == 1'b0 && cmpp == 1'b0) begin

//decrease negative offset
if (p_shift_ff == 1'b0) begin
    ff_code_n1 <= 2'b01;
end else if (p_shift_ff == 1'b1) begin
    ff_code_n2 <= 2'b10;
end

    p_shift_ff <= p_shift_ff + 1'b1;
end

ack_reg_flag <= 1'b1;
//ack_reg<= 1'b1;
var <= update;
end

end

update: begin

if (ack_reg_flag == 1'b1) begin
    ack_reg <= 1'b1;
end

var <= idle;
cmpm <= 1'b0;
cmpp <= 1'b0;

end

endcase

end else if (state == OFF) begin

//no action
ref_ctrl <= 3b000;
ack_reg <= 1'b0;
ack_reg_flag <= 1'b0;

ff_code_n1 <= 2'b00;
cc_code_n1 <= 2'b00;
sc_code_n1 <= 2'b00;
ff_code_p1 <= 2'b00;
cc_code_p1 <= 2'b00;
sc_code_p1 <= 2'b00;
ff_code_n2 <= 2'b00;
cc_code_n2 <= 2'b00;
sc_code_n2 <= 2'b00;
ff_code_p2 <= 2'b00;
cc_code_p2 <= 2'b00;
sc_code_p2 <= 2'b00;

end
end

endmodule
module nt_cal_state ( ctrl_hyst_detect_w, ctrl_offset_corr_w, complete_w, clk, reset, ack_hyst_detect, ack_offset_corr, stop, re_tune, reg_cont, gnd, vdd );

input reg_cont;
output ctrl_offset_corr_w;
input gnd;
input ack_offset_corr;
input clk;
output ctrl_hyst_detect_w;
output complete_w;
input stop;
input re_tune;
input vdd;
input reset;
input ack_hyst_detect;

reg ctrl_hyst_detect;
reg ctrl_offset_corr;
reg complete;

assign ctrl_hyst_detect_w = ctrl_hyst_detect;
assign ctrl_offset_corr_w = ctrl_offset_corr;
assign complete_w = complete;

//states

parameter s_idle_pre_tune = 0,
    s_hyst_detect = 1,
    s_offset_corr = 2,
    s_idle_post_tune = 3;

integer state;

parameter OFF = 0,
    ON = 1;

always @(posedge clk) begin

    if (reset == 1'b1) begin
        state <= s_idle_pre_tune;
        ctrl_hyst_detect <= OFF;
        ctrl_offset_corr <= OFF;
        complete <= 1'b0;
    end else if (reg_cont == 1'b1) begin

        case (state)
            s_idle_pre_tune: begin
                ctrl_hyst_detect <= OFF;
                ctrl_offset_corr <= OFF;

                //default state after reset
                state <= s_hyst_detect;
            end

            s_hyst_detect: begin
                if (ack_hyst_detect == 1'b1 && stop == 1'b0) begin
                    state <= s_offset_corr;
                end

            end

        endcase
    end
end
ctrl_hyst_detect <= OFF;
ctrl_offset_corr <= ON;
end else if (ack_hyst_detect == 1'b0 && stop == 1'b1) begin
    state <= s_idle_post_tune;
    ctrl_hyst_detect <= OFF;
    ctrl_offset_corr <= OFF;
end else begin
    ctrl_hyst_detect <= ON;
    ctrl_offset_corr <= OFF;
end
end

s_offset_corr: begin
    if (ack_offset_corr == 1'b1) begin
        state <= s_hyst_detect;
        ctrl_hyst_detect <= ON;
        ctrl_offset_corr <= OFF;
    end else begin
        ctrl_hyst_detect <= OFF;
        ctrl_offset_corr <= ON;
    end
end

s_idle_post_tune: begin
    ctrl_hyst_detect <= OFF;
    ctrl_offset_corr <= OFF;

    if (re_tune == 1'b1) begin
        state <= s_idle_pre_tune;
    end else begin
        complete <= 1'b1;
    end

    //state for after tuning
end

endcase
end
References


