UNIVERSITY OF OSLO
Department of Informatics

Wireless Sensor Network Localisation Strategies

Master thesis

Håkon K. Olafsen

May 13, 2007

Abstract

The recent years Wireless Sensor Networks have had a tremendous growth in interest. Many see the huge potential in this technology and the vast possibilities with small wireless autonomous nodes. WSN nodes have a few limitations like their small size and limited power consumption. A network might exist for years without any major maintenance, putting tight restrictions on available power. The price is also an important aspect, and cheap production technologies like CMOS is preferred.

The applications vary from cargo tracking to medical implants, and the requirements are as diverse. One requirement is present in almost all applications, and that is the need of localisation information. The localisation problem is yet to be solved for all applications. Almost all present systems rely on rigid infrastructure with anchor nodes or super nodes with extra capabilities. In this thesis a suggested algorithm for WSNs with no anchor nodes is presented, operating without any infrastructure. It is tested using different position estimation techniques.

An important part of all localisation algorithms is the data acquisition phase. This is to gather range, angle or connectivity information about the network. In WSNs angle information is hard to acquire, so connectivity and range information is the viable option. Existing range estimation techniques rely on highly inaccurate and imprecise RSS measurements. To improve the precision and accuracy of ranging a Time Difference Measuring Circuit (TDMC) circuit was implemented. A digital real time circuit implemented i 90 nm CMOS technology capable of measuring sub-10-picosecond time differences.

Contents

Preface

This thesis is submitted as part of my degree *Master of Informatics* in Microelectronics at the Department of Informatics, University of Oslo. The work initiated in January 2006 and concluded the following year, in May 2007.

I worked closely with Nikolaj Andersen creating a chip, that gave us good results. The results are included in both thesis as a cooperative chapter. This work was challenging in many ways, regarding the design and measurements. However, I have learned a lot about the engineering of an ASIC and a PCB during the process.

The individual work was challenging in its own ways. I had to work independent for a considerable period of time, which was new for me. During this work I experienced the importance of a strict schedule and organised studying. Looking back I realised I have learned a lot about much more than WSNs and positioning, especially about the process of research and writing. Not to hide that I have acquired a good understanding of positioning systems and the challenges in WSNs.

University of Oslo, May 2007

Håkon K. Olafsen

Acknowledgements

I would like to thank my supervisor Professor Tor Sverre Lande (Bassen) for accepting me as his student, and excellent inspiration, encouragement and guidance throughout the work on this thesis. I should also mention Håkon A. Hjortland and thank him for all the valuable input regarding almost every aspect in this thesis.

Thanks to Nikolaj for the teamwork and countless discussions on every aspect in this thesis and everything else. Not to forget all the late nights and long days spent at IFI.

This thesis could not have been accomplished without Håvard Kolle Riis and Hans K. Otnes Berge and the invaluable help with the production of the chip and PCB. Hans also deserves a grate thanks for his tolerance and patient helping me with MATLAB and Cadence.

To all at MES, thank you all for a great learning environment and social happenings, including but not limited to Dag, Johannes, Lena, Mats, Philip, Snorre and Yngvar.

A big thanks to all the guys at the lab, including Dag, Eirik, Ole Brum, Håvard, Jan Erik, Jens Petter, Jostein, Kristian, Ole-Petter, Svein, Trygve and Øyvind. Thanks for all the lunch breaks and really great social environment.

The Veilabben crew, ProsIT idrett and others, thanks for all the leisure time!

Chapter 1

Introduction

Moore's Lawis still valid for the IC technology, and seems to be valid for several more years. The power consumption and size of ICs are reduced each year. Size and low power are key features of a Wireless Sensor Network (WSN). WSNs are a rather new application and will probably have a huge impact on several different commercial areas. Connecting small wireless sensor nodes together in self configuring networks open the door to many interesting new applications, and the possibility to improve existing systems. One of the main properties of a WSN is the location information in the system. This location information can be used for tracking or to determine where the sensed information originates form.

Networks of cooperating autonomous devices have existed for quite some time. Some of the first implementations were used for anti-submarine warfare like the Sound Surveillance System (SOSUS) and the successor Integrated Undersea Surveillance System (IUSS). DARPA sponsored the Smart-Dust program at Berkeley University of California, to come up with smart solutions. Smart-Dust is an example where the goal is to produce dust or sand grain sized cheap sensors. DARPA deserve credit for starting the WSN research when looking into military applications. Possible military application is the use of small sensors to create a sensing grid to track enemy movement, detect chemical agents or radioactivity.

1.1 The future

The recent years WSNs have gained a lot of attention due to its huge potential. People have great ideas for different commercial WSN applications ranging from smart homes, hospitals, disaster management and much more. A hospital in 10-15 years might have chosen sensor networks of tiny sensors inside and attached to the skin of the patients, Body Sensor Network

(BSN) [Yang 06], and abandoned large monitor equipment placed at the patients bed. All communication done wireless using low power Ultra-Wide Band (UWB) signals and made available for the staff at control posts. Tiny vessels circulating in the bloodstream, delivering precise doses of medicine where needed. Smart homes with integrated systems for artificial lighting, natural lighting (controlling electric window shades and draperies), heating, ventilation, air conditioning, intercoms, security, audio visual systems, kitchen appliances and more. One smart system integrated with a small handheld control unit with touch screen to control the complete environment in the house.

The transport industry might find smart RFID tags with built in temperature sensors useful. By monitoring the environment during transport it is possible to assure high quality. A small tag on each frozen chicken that contains a complete history of the different temperatures during transport and best before date, easily checked by anyone. Containers smart enough to know if anyone have been inside and tampered with the brand new car or the weapons destined for soldiers on the battlefield, and of course capable of relaying its accurate position anywhere in the world. [IBM 07]

Humanitarian organisations might also be interested in WSNs for use in operations where people might get hurt or during disasters. After an earth quake or a building collapse time is important when recovering survivors. Tiny robots [Pesc 07] running around in the ruins with sensors to detect humans connected together with each other forming a WSN can quickly give the disaster management crucial information. In our world remains of war are ever present, and 80 million land mines and sub-munitions from cluster bombs are still buried around the world. Small expandable sensors can be used to clear minefields, if sensors to detect explosives are developed. They can be used in WSNs and spread from a aircraft in a field where unexploded objects are suspected to be. Give the network some time to create a map with information about the concentration of explosives in the area and use this as a guidance when clearing it. If reliable enough this system could reduce the need for manpower and time to check large areas which could then be used for agriculture or settlements.

1.2 Motivation

The main inspiration for this thesis is the recent development of highly accurate ranging methods. With increasing ranging accuracy some of the problems with positioning are less noticeable, and the final location estimates are more accurate. Almost all present WSNs rely on some sort of infrastructure with anchor nodes at known locations. This thesis take a closer

look at an algorithm for a WSN without anchor nodes. The main idea is that any node in the network can be used as an anchor node when collecting the data from the network and mapping the positions to real world locations.

The setting in this thesis is a self configuring WSN with autonomous and identical nodes with very accurate ranging capabilities. Advantages and disadvantages with the approach is explored and discussed closer. The main focus is on localisation and especially the positioning part of the problem. In WSNs the main challenges are:

- WSN node size and complexity
- Self configuration
- Power consumption
- Price / technology

In this thesis the tmote^{TM} sky node is used as a comparison node. The tmote sky node is used as an example of typical limitations in a sensor node regarding computing power, memory and more. It is a small module consisting of a IEEE 802.15.4 compatible radio circuit, a micro controller, temperature and humidity sensor and built in antenna and expansion connectors.

1.3 Outline of thesis

In the first three chapters the world of WSNs and the localisation problem are introduced. Discussions about the positioning algorithm and position estimation techniques are treated in chapter 4, where an algorithm is presented. In chapter 5 a Time Difference Measuring Circuit is implemented and evaluated. Simulations is used to check the quality of the positioning algorithm before the thesis is concluded and future work suggested.

Chapter 1 gives an overview of the thesis, and the goal.

Chapter 2 gives an introduction to WSNs, topology, sensors, existing standards and the localisation problem in WSNs.

Chapter 3 gives an thoroughly introduction to localisation in WSN, and the different obstacles and restrictions. Existing localisation systems are presented and briefly discussed.

Chapter 4 presents a position algorithms used in WSNs without anchor nodes. Different techniques to estimate the positions are introduced together with the key restrictions for use in WSNs. How demanding on transmission, computing and so on.

Chapter 5 overview of the produced circuitry and the idea behind it. Measurements and more. Written in cooperation with Nikolaj Andersen, and is included in both thesis. [Ande 07]

Chapter 6 gives an overview of the expected performance of the presented position algorithm, using the measurements from the circuit to estimate ranging accuracy and precision.

Chapter 7 concludes the thesis and suggest future work.

Chapter 2

Wireless Sensor Networks

A WSN is a network made of numerous small independent sensor nodes with wireless communication capability. The network is self configuring and capable of reorganising at all times. Ad-hoc WSNs are characterised by the relative short life span and limited spatial distribution and the use of mesh networking. The WSN nodes are self contained units consisting of an energy source, RF-capabilities, computing power and an actuator or sensor.

2.1 Chapter overview

The chapter starts with the network and the properties and possibilities in a network of wireless sensor nodes. Introducing different topologies and connection arrangements and organisation of the network. Then the sensor nodes are presented with limits and possibilities before the localisation problem is introduced.

2.2 Wireless Sensor Network properties

A main property of WSNs is how they are deployed. They can either be part of an organised infrastructure or make up an ad-hoc network. Different deployment methods change the characteristics of the network. When WSNs are used as part of the infrastructure in smart homes or similar, the environment includes anchor nodes and is not completely random deployed. This is a highly mixed WSN with different types of nodes controlling and monitoring the environment. A disaster area management WSN on the other hand are randomly placed in an ad-hoc fashion. The key feature of ad-hoc networks is the limited lifetime and spatial extent. They can be dropped

from an aircraft or thrown out by hand or other means. Most ad-hoc WSNs are spontaneously made when needed and by the ones that need them.

Looking closer at ah-hoc network they operate as a mesh network. A characteristic of a mesh network is the ability to handle dynamics of the network. The main idea is that all nodes can communicate with each other by node hopping. The transmission range of the nodes is limited so hop routing is necessary to achieve communication across the network. The dynamics of a WSN also have a huge effect on the performance of the network. In mesh networking the nodes reconfigure on the fly and handle node break downs and discovery of new nodes. Due to the mesh networking WSNs are redundant. If a node runs out of energy the network adjusts and continues to operate.

An advantage of mesh networking is the reduced energy needed when transmitting over large distances. The energy needed to transmit wireless is a function of the squared distance ($E_t = \kappa d^2$) using direct communication. Using mesh networking and multi hop the energy grow linearly with the number of hops needed to travel a distance ($E_t = nE_{1\text{hop}}$).

WSN can be divided into two subclasses, stationary and dynamic networks. Stationary networks are deployed and left to solve a task. There are no movement of the nodes and the position can be estimated once. Dynamical networks include moving nodes and need to estimate the position and to reconfigure as time progress.

2.2.1 Network topologies

WSNs can be arranged in different topologies according to the type of network and application. Different topologies can be star network. peer-to-peer and combination of these, see figure 2.1. In a star topology one node has the responsibility for a small cluster of nodes and the communication to and from that cluster. Nodes and clusters communicate through these coordinators and not directly. This topology has weaknesses like high load on the coordinator nodes and ineffective communication. In some applications a peer-to-peer topology might be preferable, an example of such a topology is the cluster-tree network. Communication can go directly between all nodes in range of each other putting less strain on the coordinator, and if the node is not in range, it can be routed to the other via others. [IEEE 03, Flow 06].

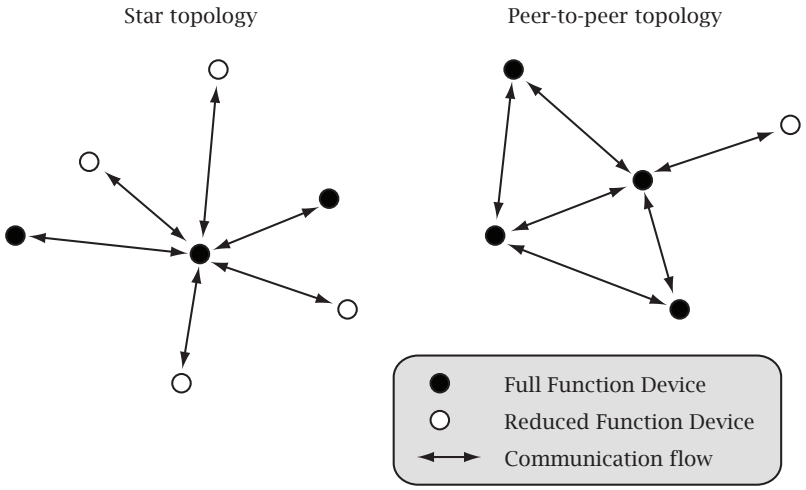

Figure 2.1: Star and peer-to-peer topologies.

2.2.2 Routing in WSN

Routing protocols in WSNs are primarily divided into *flat-based* routing, *hierarchical-based* routing and *location-based* routing. Flat-based and hierarchical-based routing do not utilise location information about a node when routing, but use other techniques. The location-based routing on the other hand uses and is dependant of location information about the target node. [Al K 04]

2.3 WSN nodes

A WSN consists of nodes with different capabilities. The sensor nodes in a WSN requires a variety of features suited for one specific application. The basic structure consists of a battery (or another power source), control and analogue electronics and an actuator or sensor, see figure 2.2. The actuator or sensor depend on the application of the node, while the three other blocks are required and can be standardised. In some application there is no need for more than the three fundamental blocks. If tracking or location is the desired information, no extra sensor block is needed.

A key feature for a WSN node is its communication capabilities. In IEEE 802.15.4 two types of nodes are defined, *Full-function device (FFD)* and *Reduced-function device (RFD)*. A FFD node implements the complete protocol set, while a RFD operates with a reduced or minimum implementation of the

protocol. The standard also discuss different roles linked to the type of topology the network has. Nodes can take on different roles based on their capabilities and position in the network. A centrally placed FFD can take on the role of *coordinator*, which is in charge of the local subset of nodes in the network. RFD nodes never takes on the role as coordinator, and usually communicates with only one coordinator node. In this thesis all nodes are considered to be FFDs.

Two other terms are used about nodes in this thesis. *Super node* is used to some extent. This is a node with extra computing capabilities and responsibility and is often a larger computer at a known location. The last term used about nodes is *Anchor node*. When mapping the estimated positions of a WSN, information about real world location for some nodes are need, these are called anchor nodes. Nodes that know their location in a predefined reference system.

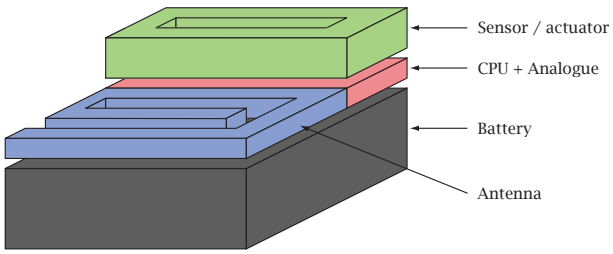

Figure 2.2: Principle sensor node.

2.3.1 Energy source

The energy source can vary depending on the application. Some actuator nodes only have a small capacitor as a battery and rely on energy harvested from the environment or from a control point to be activated. Other nodes can rely on a small battery which give them a life time of anything from hours to years depending on their consummation. Nodes that are part of a semi-fixed unit can often get energy from the power grid. Examples of these kind of nodes would be to control lights, heating, ventilation, air conditioning and audio & visual systems in a smart house. Tracking goods and creating emergency networks in a disaster area requires that the sensor node is equipped with a battery.

2.3.2 Sensing the environment

With advanced processing we are capable of creating Microelectromechanical systems (MEMS) sensors that can sense the real world and are placed on the same silicon die as CMOS-circuits. This means we can create extremely small sensors with built in control and communication capabilities. MEMS can be used to sense pressure, temperature, humidity, acceleration and biological and chemical substances. In CMOS we can create image sensors to detect light or even take pictures. Combining one or more of these sensing capabilities with the three standard blocks create a potent sensor with great value to a WSN.

2.3.3 Sensor node size & cost

Another key feature of a WSN is the cost per node. A WSN node should be generic enough to take advantage of mass production and produced using cheap technology. If the size and cost of the nodes are kept small it is easy and cheap to deploy a large amount of nodes to create a good and redundant network. When the cost of a node is small enough the node can be considered disposable.

Using CMOS as the basic technology has several advantages regarding mass production and reliability. CMOS technology also has the advantage of reduced energy consumption as the transistor size is reduced.

The size of the node is also highly dependant on the utilisation of the node. A BSN node should be very small and non intrusive, while a sensor node attached to a freight container can be much larger and still be negligible. The evolution of sensor networks is heading for smaller and smaller devices.

2.3.4 Typical sensor node

Looking to the tmote sky sensor node it is clear that the size is still noticeable. The device is approximately 3.2 cm wide, 6.5 cm long and 0.6 cm high without the battery pack consisting of two AA batteries. The onboard microcontroller run at maximum 8 MHz and has 10 kB of RAM. The program space is 48 kB and the flash is 1024 kB. These constrains limit the resources available for the localisation algorithm, and especially the computing heavy positioning task.

2.4 Localisation problem

All the information we can gather from smart sensor nodes can be of great value, if the system know where it originates from. A fundamental requirement of WSNs is the need for location information for the nodes. Either to use directly as the main source of information or to know where the sensed information originates from. Localisation information can be used in many ways. Apart from geographical origin of information it can be used for target tracking and during administration tasks such as maintenance and coverage information.

When the WSN is deployed in a smart house, it is possible to find the ID of each node and manually enter the position of the node into the system. A much easier solution would be that the node was located by the system, or found its own position. In other scenarios like a disaster area the position is not known and might even change during the operation. In all scenarios with movement of nodes and randomly distributed nodes the position of the nodes is crucial information that needs to be obtained by the network itself.

2.4.1 Location requirements

Some applications have high accuracy and precision requirements. If WSNs are to be used with industrial robots the accuracy need to be extremely high. The same applies for BSN applications where the patient has up to 128 nodes attached to the skin monitoring vitals and the accuracy has to be only a few millimetres.

In other applications the need for location information can be limited to different rooms, like the tracking of patients and nurses in an hospital. Tracking of equipment and smaller items need better accuracy, in the order of half a meter to a few meters.

2.4.2 Present

The localisation problem is a vastly explored field in recent research, and there are numerous solutions to this problem. All solutions have their limitations and capabilities that fit with the suggested application. Most of the existing systems need anchor points or beacons to get a location estimate for the network. [Bulu 00, Dohe 01, Sava 02, Kara 06] One reason for using anchor nodes is the direct link to real world location.

Not all applications need to relate itself to a real world location. If the goal is to use a small WSN as collision avoidance system in cars, where the

car is regarding longitude and latitude is not that interesting. It is more interesting if any obstacles or other cars are close and what direction they are moving in. GPS is not accurate enough to be a good alternative to a dedicated system in this case. Other scenarios might not need the longitude and latitude information, like a disaster area. The position of the node can be related to the command post, and not necessary to the absolute position.

The fundamental problem is to create a system that finds the position of the node, with a small error. This has been shown not to be an easy task. Current systems have an error estimate of about 0.5 m, which is good enough for lots of applications, but not all and there is still room for improvement.

Chapter 3

Localisation in WSN

Localisation in WSN is faced with several obstacles and problems. In this chapter the important aspects of localisation in a WSN is treated.

3.1 Chapter overview

This chapter is an overview of the localisation procedure including data acquisition and different properties and limitations regarding localisation in a WSN. Key aspects about localisation in WSNs and the limits are also discussed.

3.2 Localisation overview

In a WSN location information is one of the key elements. The procedure of locating a node is called localisation and is divided into three steps. First of all you need some sort of information about the network connectivity. Which nodes are connected, range measurements and more. The second step is to use this information to estimate a position of a node in a relative reference system. This can be done with different approaches depending on available information and the precision. Chapter 4 takes a closer look at how this is done. The third step is to map this information into real world locations, discussed in chapter 6.

First we need to define a few terms that are used in the thesis. The terms localisation, positioning and ranging have not yet gained a clear meaning in the literature. This thesis uses the definitions in *Understanding Ultra Wide Band Radio Fundamentals* [Bene 04] and some others defined here. They are as follow:

Localisation is the combination of data acquisition, position estimation and mapping.

Positioning is the process of estimating a position in a reference system.

Ranging is the action of estimating the distance between two nodes.

Position is an estimated position in a relative reference system.

Location is used for a position in the real world.

Mapping the procedure of linking the estimated positions to real world locations.

3.3 Accuracy & Precision

There are several different error sources in localisation. Both in the data acquisition and in the position estimation itself. The position estimation has two properties, the first is the grain size or the *accuracy*, this tells us the resolution of the localisation. In ranging the accuracy is the smallest distance possible to measure. The other property is the *precision* and this indicates how often the measurement is correct. The precision is given as a probability of the estimate to be within a small error. This can be found by looking at the standard deviation in a normal distributed variable. In ranging precision is expressed as; X% of the estimates are within $\pm Y$ m.

Localisation is limited by the accuracy of the data acquisition technique. If the ranging or Angle of Arrival (AoA) measurements lack precision the positioning algorithms can try to improve the result, but are always limited by the localisation data. Better accuracy and better precision in the data acquired result in better position estimates.

Accuracy is the grain size of the estimate.

Precision indicates the average error in the estimates.

3.4 Data acquisition

There is different methods to acquire localisation data about a WSN. Some information is available as soon as the first connection is established. If two nodes can communicate directly, they are within the radio range. Other than that it is possible to measure or estimate a few properties of the radio link. RF-signals propagate through air at a certain speed, with path loss and will arrive from one direction, which can all be measured.

3.4.1 Proximity

Some systems rely on proximity for localisation. This can be done with several techniques where the most common is to regard *in radio range* as in proximity. With small Radio Frequency Identification (RFID) tags and other passive nodes without battery that need power from the sensing device this can be a good solution. Other systems use pressure sensors in the floor, [Smar 07] detection of a sound, light or RF source in a room. These techniques are usually restricted to localisation within a room or proximity to a unit. The accuracy varies from a foot to several meters depending on the implementation and where it is used. This is too coarse for the positioning algorithms treated later in this thesis.

3.4.2 Angle of Arrival

Angle of Arrival (AoA) technique estimates an angle to a transmitter referenced to known axes. To determine the AoA you need at least two antennas at the receiver. It is more common to use antenna arrays called phased array. The basic principle rely on the fact that the phase of a RF signal changes over a distance. So when the signal arrives at the different antennas in the array they are out of phase with each other. The idea is then to change the delay from the different antennas in the array to a common amplifier, to maximise the input signal. Then the array of different delays can be used to determine the AoA.

Since this is based on the behaviour of RF signals this technique is not possible to use on UWB signals. Instead it is possible to use the Time Difference of Arrival (TDoA) for the signals at two antennas with known distance between them. This is done by NASA in a test system designed to track future lunar and Mars rovers. Two pairs of antennas are placed with centre 50 m apart and 15 m between the two antennas. They achieve a precision of 1% for ranges up to 610 metres. [Ni 06]

3.4.3 Ranging with RSS

Ranging is the action of estimating the distance between two nodes in communication range. This is usually done with Received Signal Strength (RSS) or Time of Flight (ToF). The most common method is RSS. RSS has a major drawback in lack of precision. The signal strength depends on the environment and range between the transmitter and receiver. When estimating a range from RSS several factors like path loss, the impulse response, multi

path and NLoS possibility needs to be considered. This is difficult for different environments. The unit needs tuning or use a generic model of the environment.

RSS has one important advantage over other systems, its simplicity. It is often available in a radio device as a measurement of the signal quality. In the GSM-system a strong signal from the base station indicates a short transmission range and the phone can reduce the transmitted power. RSS can be estimated during normal communication. The range can then be estimated without additional radio transmissions. Hence RSS is a good choice for devices where the accuracy is not crucial, but power consumption is.

3.4.4 Time of Flight

Another common ranging technique is to measure the ToF of the RF signal. In air RF-signals travel close to the speed of light. If the ToF can be accurately measured, the travelled distance can be calculated.

An existing system uses the ToF for ultra sound signals in cooperation with a RF signal to estimate the distance to a node.[Camb 07] Both signals are sent at the same time and since sound waves travel slower than RF signals they arrive at different times. In the following equation the time resolution required by the timing circuit of an ultra sound and RF system is estimated. c is the speed of light and v_s is the speed of sound in air.

$$c = 299\,792\,458 \text{ m/s} \approx 0.3 \text{ mm/ps} \tag{3.1}$$

$$v_s \approx 1238 \text{ km/h} \approx 344 \text{ m/s} = 0.344 \text{ mm/µs} \tag{3.2}$$

From this we see that a accuracy of 0.344 mm is achieved with a time resolution of 1 µs, which equals 1 MHz sampling frequency. If only RF signals are used the time resolution has to be 1 ps to achieve the same accuracy, a sampling frequency of 1 THz is not likely in any cheap process anytime soon.

Time of Arrival & Time Difference of Arrival

Time of Arrival (ToA) and Time Difference of Arrival (TDoA) are two special cases of ToF and are both depending on clock synchronisation of the reference nodes or the entire network. The ToA system requires all the nodes to have a common synchronised clock. The range between two nodes are found by sending a signal at a given time and then compare this with the time the signal arrives, hence *Time of Arrival*. TDoA on the other hand does

not require a synchronised clock at the node with unknown position, but the reference nodes do. Then the references send a signal simultaneously and the node doing the measurement detects the Time Difference of Arrival of the signals. Both techniques require synchronisation of clocks, which is extremely hard to achieve in a WSN.

Active Echo at a glance

The Active Echo scheme is a ToF measuring scheme proposed by Nikolaj Andersen. [Ande 07] The principle behind the technique is to measure the round trip time of a RF signal between a master and a slave node. The key feature is the continuous time processing in the slave node to maintain the accuracy. The ToF is measured in the master as the time difference between sent and received signal. There is no need for any synchronisation between the nodes involved in the range measurement since all the critical estimations are done in the master. Dedicated hardware is needed to implement this scheme, and is based on previous work by Håkon A. Hjortland. [Hjor 06]

This technique has the advantage of very high accuracy, good precision and no need for synchronisation.

3.4.5 Ranging over time

The ranging accuracy is improved by averaging in Active Echo. In applications with stationary or close to stationary nodes the range estimations can be averaged at a higher level too. Each range estimation has an error, by averaging the range estimations over time the error can be suppressed. Using the average range estimate higher precision can be achieved, resulting in better position estimates. For each range estimate the average has to be recalculated.

The average can be found by solving:

$$R_{avg} = \frac{NR_{avg} + R_i}{N+1}$$

By adjusting N it is possible to change the rate of decay for old estimates. $N = 1$ mean a very high decay, while increasing N reduces the decay.

When a node is moving this averaging results in less accurate range estimation. It can be hard to find a sensible algorithm for when to ignore changes in the range measurement and when to rely on them. One suggestion is to

reject the old R_{avg} when the connection to a node is lost or established. The impact of this technique should be further investigated, and is not included in the proposed algorithm.

3.4.6 Error in ranging

The accuracy and precision of ranging methods need consideration. Ranging using acoustic or RF signals have a common problem. The error in the distance measurement always results in longer distances than the real one. The signal can not be estimated to arrive earlier than the ToF time. This means that overshoot is a problem in most ranging schemes resulting in erroneous position estimates. This can be avoided with calibration as shown by Whitehouse and Culler. [Whit 02] With decent calibration the range estimates are expected to be centred around a mean with an average error.

3.5 Relative positioning

In localisation there are two fundamental different approaches to estimate a position using range information. The first rely on anchor nodes and try to estimate the range to the anchor and use this estimated range to estimate the position. The other is to use the position of the neighbours and the range to these to estimate the position. In recent literature a term used for this type of positioning has emerged, *relative location*. [Patw 03, Taub 05] The idea is to use the localisation data from neighbouring nodes and their position to estimate a position. With this technique the node does not rely on highly inaccurate estimates of ranging or other information to anchor nodes, but more accurate localisation information to neighbours. The collaborative multilateration and n-hop multilateration primitive in [Savv 02] are examples of this approach. This technique requires the localisation to propagate from the anchors to the other nodes.

When there are no anchor nodes in the network, a reference system needs to be established. This reference system is usually obtained by evolving from a *node-centred* coordinate system. This is a system where one node starts with the position (0,0) and defines the reference system as shown later in section 4.6.1.

3.5.1 Ranging with a twist

Several existing positioning algorithms do not rely on relative positioning and use coarse distance estimates. One of these add up the estimated

ranges for all the hops between anchor nodes and each node resulting in high overshoot estimates. The DV-hop algorithm by Niculescu and Nath [Nicu 01] uses an other approach which includes flooding the network with broadcasts. The main idea is to let anchor nodes estimate an average hop distance and count the number of hops from nodes to each anchor. Flooding is energy demanding and can even result in the same nodes receiving the same information more than once. This type of approaches suffer from energy dissipation and result in position estimates with large error. By using relative positioning these problems can be avoided.

3.6 Number of reference nodes required

There is a lower limitation for when it is possible to uniquely determine the position of a node. If the node knows the range to one neighbour node (N1) the position can be anywhere on a circle with radius equal to the range (r1) between them. When a second reference node (N2) is added a second circle (r2) intersects the first and the solution is the two points (A and B). For an uniquely determined position in 2D a third reference (N3) is required, and only one solution (B) remains, as seen in figure 3.1.

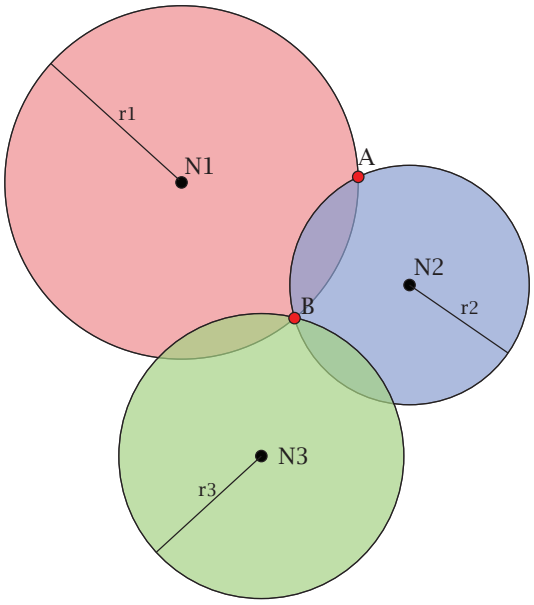

Figure 3.1: Number of nodes required for uniquely position estimation.

For a 3D position four references is needed. The first reference gives a

sphere, and the second sphere intercepts the first one and results in a circle. A third reference reduces the possible solution to two points on the circle, which in some cases might be enough. The GPS only need signal from 3 satellites to estimate the position on earth, the second solution will either be in space and or beneath the earths crust are not likely the correct one. If no knowledge about the scene is present a fourth reference is needed to uniquely estimate the position.

Some special cases exists where three references is not enough in 2D. This can be when the references are perfectly aligned or in the same point. These circumstances are not very likely in real world scenarios, but need to be considered. The problem with nodes in the same point can be avoided by not using nodes as references if the distance between them is too small.

3.6.1 On the edge

When a node is located on the edge of the network the number of connections might be too few for a unique position estimate. If only two nodes are connected two possible solutions exist. By considering the connectivity to the node with unknown position one of these estimates might be ruled out. In figure 3.2 one of the solution estimates make it highly probable that the node should have a connection to some other node, this estimate can then be discarded.

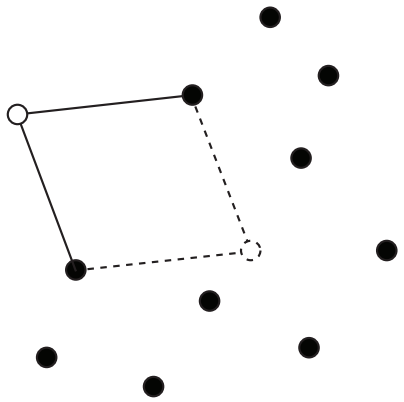

Figure 3.2: Nodes at the edge of the network.

This is just a proposed technique for handling nodes at the networks edge, this will not be implemented in the proposed algorithm.

3.7 Dynamic behaviour of a WSN

One of the important aspects about self-configuring WSNs is how the effect of disappearing nodes is handled. As long as the node coverage is good, the network continues to function as before. Losing a node reduces the ability to sense or interact with the real world at that position, but the overall network is still connected and continues to work as intended. If the node was in a critical position regarding routing in the network the impact can be severe, splitting the network or creating a bottleneck. The network also needs to handle a new node entering the network. This node also need location information related to the already existing network.

Movement of nodes is also an important aspect to handle in ad-hoc WSNs. In some applications tracking of moving objects can be an important feature in the network. Depending on the demand of tracking accuracy various techniques can be applied. If the tracking capabilities are critical it might be an idea to have the node broadcasting a beacon at predefined intervals and let the infrastructure handle the tracking. The ranging technology should also be fast enough so that the different range estimates originate from approximately the same location. If the speed is high, the different range measurements might occur on different physical locations and give a higher error in the final location estimate. If the position tracking is not that critical the node itself can compute the position and broadcast to its neighbours.

Applications like car collision avoidance need very high precision in the position estimation and real time processing of the possibility for a collision. If the WSN is used for meteorological measurements in a forest it is not critical to register the movement of nodes, but to estimate a new precise position when the node is stationary.

The dynamic changes in a network has an effect on the lifetime of the network. Solutions that use RSS for ranging have the advantage of no extra transmissions compared to dedicated ranging systems. A dedicated ranging system like Active Echo results in shorter lifetime for the node. Passive ranging methods like RSS are preferred when very long lifetime is considered more important than high accuracy localisation. Coarse localisation methods in a WSN can increase the lifetime by reducing the power consumption.

If the network is stationary the nodes should save energy by turning off the positioning part and reduce the range estimation interval. To determine when to recalculate the position the range estimates can be stored for later use and compared to the estimates used in the previous position phase. If the change in the ranges passes a threshold a new position can be estimated.

3.8 2-dimensional or 3-dimensional localisation

A WSN can work with 2D or 3D localisation and both have advantages and disadvantages. In a 2D system, the estimation process is less complex and require less energy and time. The equations used to solve the problem have fewer terms in 2D than 3D, and the equation system has one less equation as a minimum requirement. Using 2-dimensional algorithms on a system place in 3-dimensions can result in erroneous position mapping to the real world. When estimating a position in 2D from a 3D environment the position estimate is the point in the plane where the x and y coordinate are the same as the real position. When mapping these estimated positions back to the real world an error can occur if the reference system is thought to be aligned with the ground while it is not. Any angle between the reference plane and the ground result in an error during mapping if this is not considered. The error can be avoided by first aligning the reference system to ground by rotating it around the x- and y-axis. By using a localisation system for 3D this problem is eliminated completely.

3.9 Numerical limits

WSN nodes usually have a reduced microcontroller to save power. The tmote sky node has a ultra low power microcontroller from Texas Instruments with a 16 bit Arithmetic Logic Unit (ALU) and a 16×16 multiplier. This limits the numerical accuracy during the position estimation. A system implementing a high accuracy positioning algorithm using high accuracy ranging information need to consider the numerical limitations during calculations. Very large and small values should be avoided during the computation reducing any error caused by overflow or rounding. Floating point representation also has problems representing real values. To avoid any problem with numerical limits in the computation the equation used should be reduced or expanded and calculated in a carefully chosen order.

3.10 Existing localisation systems

There are numerous existing localisation systems commercially available. The best known localisation system is GPS. GPS is a good system for outdoor localisation in medium to large sized units. Relying on LoS to satellites orbiting the earth limits the indoor use of GPS. The complex and power consuming computations require a medium to large unit to have power resources to last for some time. These limitations make GPS unsuitable for use in a WSN.

Other systems also rely on infrastructure of some sort. The Active Badge system use diffuse infrared technology and a infrastructure of proximity sensors placed in the area to be monitored. [Want 92] By using infrared light the accuracy of position estimation is in the size of rooms. Ultrasound is used by the Active Bat system when measuring distances. [Camb 07] The distance to the badge is collected and sent to a central processing unit that estimate the position. The receivers are placed in a grid in the ceiling of the monitored area. Both these systems require extensive infrastructure to be deployed in the area of interest, which increase the cost. SpotON is based on the same principle with a infrastructure consisting of base stations at known locations. [High 00] By using RSS the system estimates the distance between the object and the base stations and sends it to a central computer. Ubisense is yet another implementation with fixed infrastructure that feeds information back to a central processing point. [Ubis 05] The main difference with Ubisense is the use of UWB signals to achieve high precision and accuracy.

The IEEE 802.15.4 standards includes localisation. Motorola has developed a dedicated hardware core for this task. It is a distributed system where each node estimates its own position. This hardware core is implemented in the CC2431 from Texas Instruments. Range information come from RSS measurements and a maximum likelihood algorithm is applied to estimate the positions. This positions is estimated through the optimisation of a cost function based on RSS relations. This system has an precision of 3 m. The implementation can handle between three and eight reference nodes. [Taub 05] By limiting the maximum number of reference nodes they effectively limits the precision.

Chapter 4

Positioning Algorithm

Localisation need a good and suited positioning algorithm. In this chapter an algorithm is proposed and different techniques regarding the position estimation is presented.

4.1 Chapter overview

In this chapter the scenario and a suited positioning algorithm are presented. The positioning algorithm describes the behaviour of the sensor nodes regarding localisation. Information like connectivity and range information is used by the algorithm to estimate a nodes position. A major part of the algorithm is the technique to estimate the position. Several techniques are discussed and reviewed in a WSN setting with the constraint this implies.

The chapter start with the limits and possibilities in WSNs before the scenario and suggested algorithm is presented. Then the position estimation techniques are discussed.

4.2 WSN considerations

Comparing positioning algorithms is not an easy task. Most algorithms are set in a certain environment where they perform good, while changes to the environment can reduce the quality. In this chapter the comparison is focused on computational complexity while the quality of position estimates is treated in chapter 6. The main limitations are the limited computing capabilities and energy resources, and the estimation techniques are

discussed based on this. If the technique is simple enough to be implemented in dedicated electronics this can reduce the power consumption and is briefly discussed.

Another important aspect of WSN localisation is distributed versus centralised algorithms. The energy cost of one transmission can be equal to several hundred computations in the node itself. Chen et al. show that this holds true for a set of conditions in *Source localization and beamforming*. [Chen 02] With this in mind any position algorithms based on gathering all information at a super node and then estimating the positions seems futile. The super node and its neighbouring nodes suffer from this kind of algorithm and use excessive energy relying information. This does in fact restrict the maximum number of nodes the system can handle. [Dohe 01] A solution is to let the nodes themselves estimate their position and only rely the final result when needed. The power consumption are spread around the network and the computing power in the nodes can be utilised. This can also be done with other tasks in the network, not only localisation.

Position algorithms can be very computational intensive requiring hundreds of floating point calculations. When the same computation is done in several nodes energy might be wasted. An idea is to broadcast the results of some computations with very high power consumption to reduce the energy dissipation in the WSN. For this to be a viable solution the energy required by the computation has to be larger than the energy used when transmitting the result. Time is also an issue in positioning algorithms when sharing results. Transmitting results take some time and adds a time delay for when the result is ready. Both the energy and time perspective are very hard to estimate and is outside the scope of this thesis.

4.2.1 Effect of more connections

Depending on the algorithm and estimation technique used, the number of connections to a node might have no or large impact on the accuracy and precision. If the technique is very simple and only uses the lowest number of connections needed to estimate the position, the number of connections do not have any impact. On the other hand it is possible to use Least Square Estimation (LSE)s to handle over determined systems, see section 4.7.4. The min-max in section 4.7.5 algorithm also utilise more than minimum number of connections.

By using more than three reference nodes the precision is shown to increase with the number of connections. The accuracy increase rapidly as the connectivity increases up to about 10–15 nodes. [Sava 02, Lang 03] In earlier literature and systems with anchor nodes, 15 connections seems to be the limit for when the error in position estimates steadies out.

Not all techniques can use more than three reference nodes when estimating a position. These techniques need some sort of quality check for the reference positions and the range measurements to determine which neighbours to use as references.

4.2.2 Arithmetic operations

Due to the limited computational power in small WSN nodes the positioning algorithm should reduce the number of arithmetic operations. The type of operation is also important. Addition and subtraction is easy to perform compared to multiplication and divisions. One exception is integers multiplied and divided by 2^n, which is easily done with bit shifts. Trigonometric functions, such as sine and cosine, needs to be estimated and are expensive operations. The more computational heavy operations can be implemented in dedicated hardware to reduce the strain on the microcontroller. Creating a dedicated circuit to handle the position estimation might the best solution to reduce the energy cost and computation time.

4.3 Scenario

The scenario for this thesis is a self configuring WSN with no prior localisation information at any node. There is no anchor nodes in the network, and no nodes with GPS or other form of localisation hardware. The nodes can measure the range to nodes within communication range with high accuracy and precision. The nodes only estimate their own position and broadcast it to their neighbours.

The reference system is created in the first node activated and spreads through the network as nodes connect and estimate their position, a fully relative reference system.

4.4 Unsuited algorithms

In earlier literature several algorithms have been proposed as solutions to the localisation problem. The first algorithms almost always rely on centralised computing and have high communication cost. Examples of these algorithms are the Convex positioning system by Doherty et. al [Dohe 01] and multidimensional scaling by Ji et. al [Ji 04]. Kalman filter solutions have also been suggested both as centralised [Zaid 05] and distributed systems. [Savv 02] Other methods used for localisation such as triangulation can also be used, but rely on AoA information.

All these algorithms are either based on the existence of anchor nodes or centralised computing. They are not suited for implementation in autonomous sensor nodes in ad-hoc WSNs. In this thesis algorithms and techniques that fit the scenario are evaluated.

4.5 Suggested algorithm

The suggested algorithm is event driven. If no changes have occurred with the neighbours, energy is conserved by not recalculating the position. When a neighbour node estimates a new position or a new neighbour node connects the node estimates a new position. New range measurements are also a trigger for estimating a new position. Not implemented here, but a test to see if the new range measurement is noticeable different from the last can reduce the power consumption by not estimate a new position.

Algorithm 4.1 Position Algorithm

```
if numberOfNeighboursWithPos > 3 then
  if notOwnPos or newNeighbourPos or newNeighbourRange then
    ownPos = estimatePos
  end if
  εn = |xn−1 − xn|
  if εi > εl then
    broadcastPosInfo
  end if
end if
```

The suggested technique fits the presented scenario and estimates a position for all nodes with three or more neighbours with position information. The new position estimate has to represent a significant change in position to be broadcasted. Nodes with low connectivity might not find a position, and the effect of radio range / number of connections is evaluated in chapter 6.

This algorithm need a reference system and at least 3 connected nodes with position estimates. Then any nodes connected to these three can estimate their position and broadcast it. Position awareness spread through the network originating from the three first nodes. To create this reference system the technique in the next section is used.

4.6 Reference system

The reference system is the coordinate system for the estimated positions and can be relative for the WSN or related to the real world. If anchor nodes are present in the network a reference system is present and can be used when locating other nodes. When no anchor nodes are present a relative reference system is needed. This system is created and maintained by the WSN. Savarese et. al suggested a technique to create local reference systems in 2001. [Sava 01] This system start from scratch with only range information about the local nodes.

4.6.1 Assumption Based Coordinates

Starting from scratch with only range information available the Assumption Based Coordinates (ABC) algorithm can create a reference system. ABC starts with one node in position (0,0). It then use one of the nodes in communication range as a node on the x-axis (D_{12},0). The position of the next node is assumed to be in positive y-direction and is calculated as shown below. Figure 4.1 show the idea.

1. n_0 located at (0,0,0)
2. n_1 located at (D_{01},0,0)
3. n_2 located at (x_3,y_3,0)
4. n_3 located at (x_4,y_4,z_4)

$$
\begin{aligned}
x_3 &= \frac{D_{12}^2 + D_{13}^2 - D_{23}^2}{2D_{12}} \\
y_3 &= \sqrt{D_{13}^2 - x_3^2} \\
x_4 &= \frac{D_{12}^2 + D_{14}^2 - D_{24}^2}{2D_{12}} \\
y_4 &= \frac{D_{14}^2 - D_{34}^2 + x_3^2 + y_3^2 - 2x_3x_4}{2y_3} \\
z_4 &= \sqrt{D_{14}^2 - x_4^2 - y_4^2}
\end{aligned}
$$

When the first three nodes are found, the next step is to find a position estimate for other nodes in range of the first three.

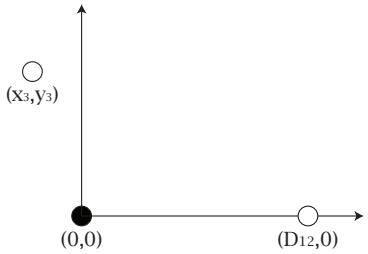

Figure 4.1: Show the principle of ABC.

4.7 Position estimation

When a reference system is established the remaining nodes need to estimate a position. The algorithm opens for different techniques as long as they rely on ranging information and the position of neighbours. All the following algorithms are generic in the way that they only expect the position information from three or more reference positions and the range to these. In this scenario only nodes within communication range can be used as reference nodes. When three neighbour nodes have estimated a position and the range to the node, a position can be estimated with on of the following techniques.

4.7.1 Lateration

Lateration is to use information about range between points to estimate a position. Looking at the simple case of two known points and the range to a unknown position the problem is solved using *Law of Cosines*. In figure 4.2 the position of point A and B and the sides a and b are known. By using the Law of Cosines the angle α can be found, equation 4.1.

$$\begin{aligned} c^2 &= a^2 + b^2 - 2ab\cos\alpha \\ \cos\alpha &= \frac{a^2 + b^2 - c^2}{2ab} \\ \alpha &= \cos^{-1}\left(\frac{a^2 + b^2 - c^2}{2ab}\right) \end{aligned} \tag{4.1}$$

When α is known it is easy to find the position of point C by solving:

$$\begin{aligned} x_i &= b\cos\alpha \\ y_i &= b\sin\alpha \end{aligned}$$

This result is not unique, as it yields two solutions in 2D and a circle in 3D. To uniquely determine the position a third reference point is needed to test which solution is correct. Since this technique often is used to solve problems where the scene is known, information from two reference points is enough. If a third reference point is available there are two dominant techniques to estimate a solution. They are called trilateration and multilateration, and the difference is the approach when solving the equation sets.

Position Algorithm	+	−	·	/	bit shift	other
Lateration	1	1	7	1	–	arccos + sin + scene

Table 4.1: Number of arithmetic operations involved in lateration.

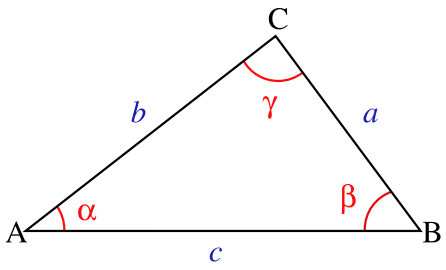

Figure 4.2: Triangle with notations.

All the arithmetic operations needed by lateration is listed in table 4.1. Each algorithm description includes a table like this and in section 4.9 the algorithms are compared. Lateration is a fairly simple algorithm with few operations, with arccos and sin standing out as the more demanding ones. There is no need for the cos function as this is the result before arccos, saving one operation. This type of reduction is also considered for the other algorithms.

4.7.2 Trilateration or Spherical estimation

Trilateration use ranging information to look at spheres in the 3-dimensional space. The range between the unknown node and a reference node creates

a sphere and the different spheres intersect and if the range information is accurate, resulting in one solution. In the 2-dimensional case circles drawn around the reference nodes with radii equal to D_{ni} are considered, see figure 4.3 from Wikipedia. The general equation is:

$$\begin{aligned} D_{ni} &= \sqrt{(x_n - x_i)^2 + (y_n - y_i)^2} \\ D_{ni}^2 &= (x_n - x_i)^2 + (y_n - y_i)^2 \end{aligned} \tag{4.2}$$

For the three reference nodes (1–3) the equations are:

$$\begin{aligned} D_{1i}^2 &= (x_1 - x_i)^2 + (y_1 - y_i)^2 \\ D_{2i}^2 &= (x_2 - x_i)^2 + (y_2 - y_i)^2 \\ D_{3i}^2 &= (x_3 - x_i)^2 + (y_3 - y_i)^2 \end{aligned}$$

To solve this problem, reference 1 is considered to be located at (0, 0), reference 2 at (x_2, 0) and reference 3 at (x_3, y_3). This reduces the equations as follows:

$$D_{1i}^2 = x_i^2 + y_i^2 \tag{4.3}$$
$$D_{2i}^2 = (x_2 - x_i)^2 + y_i^2 \tag{4.4}$$
$$D_{3i}^2 = (x_3 - x_i)^2 + (y_3 - y_i)^2 \tag{4.5}$$

The first step is to find x_i by substituting 4.3 for y_i^2 into 4.4.

$$\begin{aligned} D_{2i}^2 &= (x_2 - x_i)^2 + D_{1i}^2 - x_i^2 \\ x_i &= \frac{D_{1i}^2 - D_{2i}^2 + x_2^2}{2x_2} \end{aligned}$$

Then substituting 4.3 for y_i^2 into 4.5 to find y_i.

$$\begin{aligned} D_{3i}^2 &= (x_3 - x_i)^2 + y_3^2 - 2y_3y_i + D_{1i}^2 - x_i^2 \\ y_i &= \frac{D_{1i}^2 - D_{3i}^2 + x_3^2 + y_3^2 - 2x_ix_3}{2y_3} \end{aligned}$$

These three equations yields a single solution. This technique has problems with errors in the measured ranges. If the measured ranges are not precise the position found is not correct position. The error from this technique can be quite large, due to the fact that the range estimates are squared. The equations set can be extended to handle 3D.

A problem with this approach in WSNs is the assumption of reference placement. This means that the reference nodes need to rearrange their position to fulfil the placement to reduce the equations before estimating the position. This movement and possible rotation of the reference system adds more arithmetic operations to find an estimate. The process of rotating the coordinate system is costly as seen in section 4.8. The number of operations for trilateration is found in table 4.2.

Position Algorithm	+	−	·	/	bit shift	other
Trilateration	3	3	7	2	3	unaligned system

Table 4.2: Number of arithmetic operations involved in trilateration.

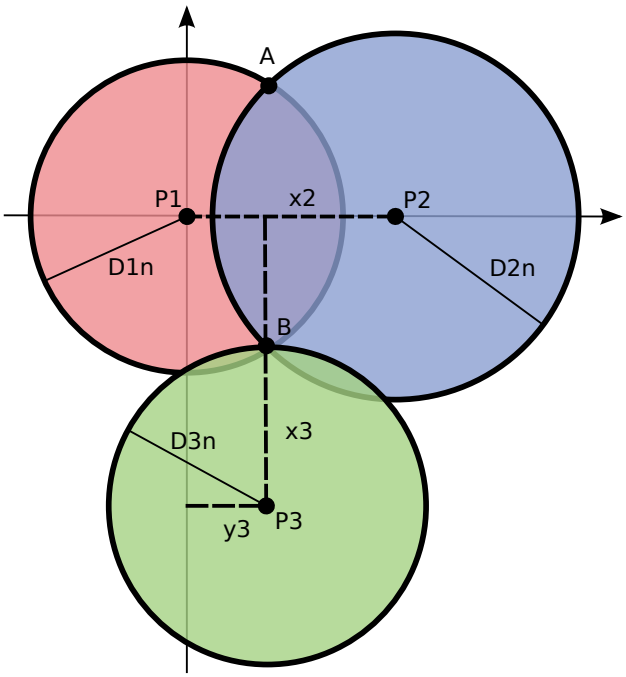

Figure 4.3: Show the circles used in 2D trilateration.

4.7.3 Multilateration or Hyperbolic positioning

This technique is originally used with TDoA measurements. The main difference between trilateration and multilateration is how you solve the equation sets. In multilateration you look at the difference in range between the unknown position and two reference nodes. Instead of circles or spheres the equations yields hyperbolas, see figure 4.4. If the range difference to two reference points is constant and the remaining distance is varied, it is easy to see why this yields hyperbolas. The following equations use the position of the reference nodes, and use no simplifying by expecting special positions of the reference nodes.

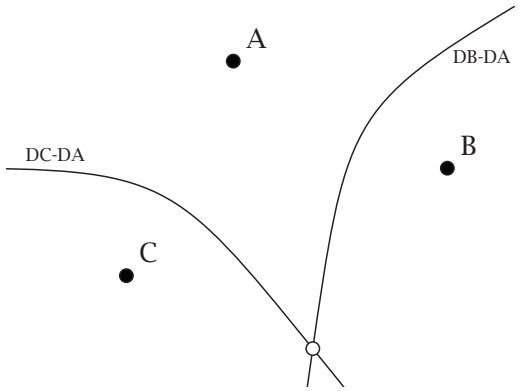

Figure 4.4: Show the hyperbolas used in 2D multilateration.

Starting with equation 4.2 two equations can be derived. By subtracting the equation for reference 2 from the equations for reference 1 and 3 we get:

$$(x_1 - x_i)^2 + (y_1 - y_i)^2 - (x_2 - x_i)^2 - (y_2 - y_i)^2 = D_{1i}^2 - D_{2i}^2 \quad (4.6)$$
$$(x_2 - x_i)^2 + (y_2 - y_i)^2 - (x_3 - x_i)^2 - (y_3 - y_i)^2 = D_{2i}^2 - D_{3i}^2 \quad (4.7)$$

Solving these two equations yields the estimated position. x_i is found as a function of y_i by rearranging 4.6 with respect to x_i.

$$\begin{aligned} x_i &= \frac{A + 2y_i(y_3 - y_1)}{2(x_1 - x_3)} \\ A &= (D_{3n}^2 - D_{1n}^2 - x_3^2 + x_1^2 - y_3^2 + y_1^2) \end{aligned} \quad (4.8)$$

Then 4.8 is substituted into 4.7 to find y_i.

$$
\begin{aligned}
y_i &= \frac{A(x_3 - x_2) + B(x_1 - x_3)}{2\left((x_1 - x_3)(y_2 - y_3) + (x_3 - x_2)(y_1 - y_3)\right)} \\
B &= (D_{3n}^2 - D_{2n}^2 - x_3^2 + x_2^2 - y_3^2 + y_2^2)
\end{aligned}
$$

When y_i is estimated, x_i is found by using y_i in 4.8. This technique is straight forward and does not rely on any additional arithmetic operations. It is computational heavy but does not include any square roots or trigonometrical operations. The operations needed are found in table 4.3.

Position Algorithm	+	−	·	/	bit shift	other
Multilateration	7	14	14	2	3	–

Table 4.3: Number of arithmetic operations involved in multilateration.

Multilateration and offsets

During range measurements done with TDoA or the Active Echo scheme an extra delay (δ) occur. In TDoA this error is caused when the received signal is processed, and all signals get the same error. This error originates from the receiver circuit, and also exists in the main device in the Active Echo Scheme. In Active Echo this delay also originates in the echo device and the delay between received and sent pulse. This delay should be the same for all nodes resulting in the same error. Since this delay is equal for all estimates, multilateration can compensate for this delay without any extra arithmetic operations. δ is cancelled due to the subtraction of the distance measurements, as shown below:

$$
D_{(n-1)i} - D_{ni} = c(\tau_{(n-1)i} + \delta) - c(\tau_{ni} + \delta) = c(\tau_{(n-1)i} - \tau_{ni})
$$

4.7.4 Least Square Estimation

Least Square Estimation is a general approach to find the best solution to over determined equation sets. This means that more than three reference nodes can be used when estimating a node position. By finding the best solution with more than three references should yield a better estimate. A set of k, where k is the number of references, multilateration equations is linearised and solved. This was first shown in [Sava 01] and then used in [Lang 03] and [Bene 04].

Starting with the equations for multilateration with n nodes:

$$\begin{aligned}
(x_1 - x_i)^2 + (y_1 - y_i)^2 - (x_n - x_i)^2 - (y_n - y_i)^2 &= D_{1i}^2 - D_{ni}^2 \\
(x_2 - x_i)^2 + (y_2 - y_i)^2 - (x_n - x_i)^2 - (y_n - y_i)^2 &= D_{2i}^2 - D_{ni}^2 \\
&\cdots \\
(x_{n-1} - x_i)^2 + (y_{n-1} - y_i)^2 - (x_n - x_i)^2 - (y_n - y_i)^2 &= D_{(n-1)i}^2 - D_{ni}^2
\end{aligned} \tag{4.9}$$

Looking at the general equation (4.9) and expanding it:

$$\begin{aligned}
D_{(n-1)i}^2 - D_{ni}^2 &= (x_{n-1} - x_i)^2 - (y_{n-1} - y_i)^2 \\
&- (x_n - x_i)^2 + (y_n - y_i)^2 \\
D_{(n-1)i}^2 - D_{ni}^2 &= x_{n-1}^2 - 2x_i x_{n-1} + x_i^2 + y_{n-1}^2 - 2y_i y_{n-1} + y_i^2 \\
&- x_n^2 + 2x_i x_n - x_i^2 - y_n^2 + 2y_i y_n - y_i^2 \\
D_{(n-1)i}^2 - D_{ni}^2 &= 2x_i (x_n - x_{n-1}) + x_{n-1}^2 - x_n^2 \\
&+ 2y_i (y_n - y_{n-1}) + y_{n-1}^2 - y_n^2
\end{aligned}$$

Rearranging the content of the parenthesis' and the equation:

$$\begin{aligned}
&-2x_i (x_{n-1} - x_n) - 2y_i (y_{n-1} - y_n) \\
&= D_{(n-1)i}^2 - D_{ni}^2 - x_{n-1}^2 + x_n^2 - y_{n-1}^2 + y_n^2
\end{aligned} \tag{4.10}$$

Defining the linear problem as:

$$A\mathbf{x}_i = \mathbf{b}$$

Where $A\mathbf{x}_i$ is the left hand side of equation 4.10, resulting in A and $\mathbf{x}_i$ equal to:

$$A = -2 \begin{bmatrix} (x_1 - x_n) & (y_1 - y_n) \\ (x_2 - x_n) & (y_2 - y_n) \\ \cdots & \cdots \\ (x_{n-1} - x_n) & (y_{n-1} - y_n) \end{bmatrix}$$

$$\mathbf{x}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

And the **b**-vector:

$$\mathbf{b} = \begin{bmatrix} D_{1i}^2 - D_{ni}^2 - x_1^2 + x_n^2 - y_1^2 + y_n^2 \\ D_{2i}^2 - D_{ni}^2 - x_2^2 + x_n^2 - y_2^2 + y_n^2 \\ \cdots \\ D_{(n-1)i}^2 - D_{ni}^2 - x_{n-1}^2 + x_n^2 - y_{n-1}^2 + y_n^2 \end{bmatrix}$$

This might be solved as follows:

$$\mathbf{x}_i = (A^T A)^{-1} A^T \mathbf{b}$$

This algorithm gives the same precision as multilateration when three references are used, and improves with the number of references. The position estimates converges and become more precise as long as the node is stationary. LSE has more demanding computations than multilateration, solving the linear problem increases the number of arithmetic operations drastically. Table 4.4 lists the operations before the linear problem is solved. The effect of increasing number of references (n) is also visible. To solve the linear problem the inverse of matrix A is needed. This can be done by QR-decomposition and LU-decomposition, but is outside the scope of this thesis. Eirik Kile has implemented QR-decomposition using Givens rotation in VHDL in his Master thesis. [Kile 06]. This thesis show that implementing matrix inversion of 25×25 matrices on FPGA is possible and fast.

Position Algorithm	$+$	$-$	$\cdot$	$/$	bit shift	other
LSE	$2(n-1)$	$5(n-1)$	$3n$	–	$2(n-1)$	solve linear equation

Table 4.4: Number of arithmetic operations involved in LSE.

4.7.5 Min-max

One of the drawbacks with lateration is the number of floating point operations required. Savvides et al. presented a simpler method as part of the n-hop multilateration approach. [Savv 02] The idea is to look at a bounding box around each reference node and determine the intersection of these

boxes. The estimated position is the centre of the intersection box, figure 4.5. The ranges from the reference nodes to the node with unknown position creates a bounding box around the references, seen as the corners. The intersection box is light grey and and estimated position, the estimated position is plotted as grey. Min-max scale with the number of references like LSE.

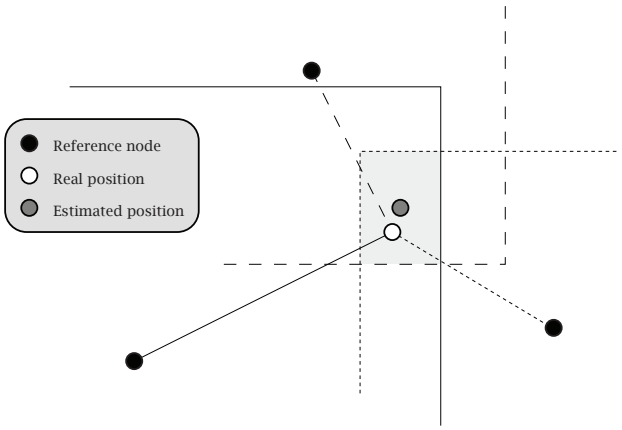

Figure 4.5: Show the principle of min-max.

The bounding box of each node are found by finding the position of the lower left ($C_{B_{LL}n}$) and upper right corner ($C_{B_{UR}n}$) as follows:

$$\begin{aligned} C_{B_{LL}n} &= [x_n - D_{ni}, y_n - D_{ni}] \\ C_{B_{UR}n} &= [x_n + D_{ni}, y_n + D_{ni}] \end{aligned}$$

This can be found for each of the reference nodes. To find the intersection box the maximum of all the coordinate minima and the minimum of all the maximums are found. The lower left corner is constrained by the maximum of the x-axis minima and the maximum of the y-axis minima. Upper right corner is constrained by the minimum of the x-axis maxima and the minimum of the y-axis maxima.

$$\begin{aligned} C_{I_{LL}i} &= [\max(x_n - D_{ni}), \max(y_n - D_{ni})] \\ C_{I_{UR}i} &= [\min(x_n + D_{ni}), \min(y_n + D_{ni})] \end{aligned}$$

To find the final position estimate, the centre of the intersection box is found:

$$\begin{aligned} x_i &= \frac{\max(x_n - D_{ni}) + \min(x_n + D_{ni})}{2} \\ y_i &= \frac{\max(y_n - D_{ni}) + \min(y_n + D_{ni})}{2} \end{aligned}$$

This technique has a major advantage regarding computational complexity, no multiplications or divisions is required only two bit shifts. In table 4.5 the different arithmetic operations are listed and the effect of n reference nodes is visible. The accuracy suffers and is not as good as a position estimated using lateration.

Position Algorithm	+	−	·	/	bit shift	other
Min-max	$2n+2$	$2n$	−	−	2	2 max +2 min

Table 4.5: Number of arithmetic operations involved in min-max.

4.7.6 Refinement

When the nodes have an estimated position it is possible to use an iterative approach to improve the position estimates. The refinement algorithm look to the nodes neighbours when estimating its position. The positions of the neighbouring nodes and the range measurements to them are used to form an equation set. This is done with multilateration, solving the linearised equations $A\mathbf{x} = \mathbf{b}$, using the LSE algorithm. This new position is a better estimate in the reference system compared to the last one. This technique has the advantage that all the nodes in the entire system align and adjust their position and eventually have very good position estimates. [Sava 02] When the updated position estimate is close to the previous position estimate, $\varepsilon_n = \left|\mathbf{x}_{n-1} - \mathbf{x}_n\right| < \varepsilon_l$, refinement stops. ε_l is a predefined limit for when to stop the procedure.

4.8 Combining reference systems

When speaking of ad-hoc WSNs there is a chance that two networks connect while they both have good localisation information. When this happen there are two possible solutions. First of all you can discard the information from one of the networks and estimate the locations of the nodes in this system by using one of the position algorithms. The other solution is

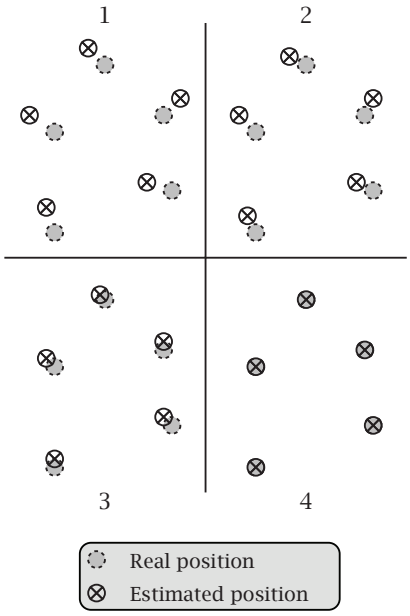

Figure 4.6: The principle of Refinement.

to transform the system so that the axes of two system align. Depending on the size of the two systems both solutions can be feasible. If a small network connects with a larger network, it might be more energy efficient to estimate the positions from scratch, while larger networks connecting can benefit from the following technique.

This algorithm aligns the axes of two known coordinate systems. The two systems need to be centred around two nodes (**i** and **k**) in range of each other, with positions in each others local reference system, and have a third node (**j**) known to both. When aligning two reference systems there are two possible cases in 2D. One is that you only need to rotate one of the systems to align the axes, and the other is that you need to rotate and mirror around one of the axis.[Capk 01]

In figure 4.7, from [Capk 01], the angles used to determine what case applies are seen. The angles from the reference systems are used as seen below to find the angle which to rotate one of the systems so that the axes align. When the axes are aligned, and the angles give us case two, the old system also needs to be mirrored around the y-axis to align completely. The angles $\alpha_{jk} = \alpha_j - \alpha_k$ and $\beta_{ji} = \beta_j - \beta_i$ are between 0 and 2π.

When the case is determined and the rotation angle for one of the systems is found, this can be broadcasted to all the nodes in the old system. To align the two systems the information needed is the position of the refer-

if $\alpha_j - \alpha_k < \pi$ and $\beta_j - \beta_i < \pi$
or $\alpha_j - \alpha_k > \pi$ and $\beta_j - \beta_i > \pi$
=> mirroring is necessary
=> the correction angle $\theta = \beta_i + \alpha_k$

if $\alpha_j - \alpha_k < \pi$ and $\beta_j - \beta_i > \pi$
or $\alpha_j - \alpha_k > \pi$ and $\beta_j - \beta_i < \pi$
=> mirroring is not necessary
=> the correction angle $\theta = \beta_i - \alpha_k + \pi$

Table 4.6: Angle tests to find the correct approach.

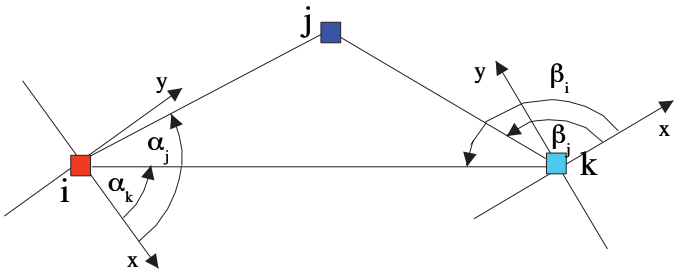

Figure 4.7: Angles used for aligning coordinate systems.

ence node for the old system in the old and new system and the rotation angle. The rotation around the reference node can be done by multiplying an Affine matrix with the position vector. Any mirroring is done with respect to the y-axis by changing the sign of the x-coordinate after rotation.

The matrix used in this affine transformation is given below:

$$A = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

The old position $I_{\mathbf{k}_n}$ of the node are given by:

$$I_{\mathbf{k}_n} = \begin{bmatrix} x_{\mathbf{k}_n} \\ y_{\mathbf{k}_n} \end{bmatrix}$$

First the old position of the reference node is subtracted, so the origin of the coordinate system is at that node. The new position in this system $I_{\mathbf{r}_n}$ can be found by doing the following matrix multiplication.

$$\begin{aligned} I_{\mathbf{r}_n} &= A \times I_{\mathbf{k}_n} \\ I_{\mathbf{r}_n} &= \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \times \begin{bmatrix} x_{\mathbf{k}_n} \\ y_{\mathbf{k}_n} \end{bmatrix} \\ I_{\mathbf{r}_n} &= [x_{\mathbf{r}_n} \quad y_{\mathbf{r}_n}] \\ x_{\mathbf{r}_n} &= x_{\mathbf{k}_n}\cos\theta + y_{\mathbf{k}_n}\sin\theta \\ y_{\mathbf{r}_n} &= -x_{\mathbf{k}_n}\sin\theta + y_{\mathbf{k}_n}\cos\theta \end{aligned}$$

Now the axes are aligned, and the only thing missing is to get the same origin as the other system. If we use the reference node **i** as the reference after the transformation the new position for the nodes in the system of reference **k** will be:

$$\begin{aligned} x_{\mathbf{i}_n} &= x_{\mathbf{r}_n} - x_{\mathbf{i}_k} \\ y_{\mathbf{i}_n} &= y_{\mathbf{r}_n} - y_{\mathbf{i}_k} \end{aligned}$$

$x_{\mathbf{i}_n}$ and $y_{\mathbf{i}_n}$ are the coordinates for node n in system $\mathbf{i}$. When this is completed all the nodes are in the same reference system. Table 4.7 lists the operations involved in merging two reference systems. Depending on the resources needed to transmit the result of solving $\cos\theta$ and $\sin\theta$ this can either be computed once and then broadcasted, or the angle is broadcasted and each node does the computation, hence the (n) in the table. In addition to the table the algorithm needs to control that all angles used in computations are between 0 and 2π. To find the angle θ one or two more operations are needed operations.

Position Algorithm	$+$	$-$	$\cdot$	$/$	bit shift	other
Aligning	n	$3n$	$4n$	–	–	$(n)\sin+(n)\cos+4\arctan$

Table 4.7: Number of operations involved in aligning systems.

4.9 Comparison

The presented algorithm is able to create a reference system and estimate the position to all well connected nodes in the network. The estimation techniques are all viable in the presented scenario. From previous work lateration techniques are known to produce precise positions estimated when the ranging precision is good. Trilateration as presented rely on the aligning of systems after the positioning and this adds more arithmetic operations, section 4.8. By using multilateration the position is estimated directly based on the reference nodes positions and reducing the overall number of arithmetic operations. Introducing the LSE algorithm can increase the precision of the position estimate at the cost of energy and time.

The min-max algorithm is a much simpler approach, but with expected lower precision. This technique is very interesting as it can be implemented directly in electronics as a dedicated circuit. The number of arithmetic operations is much lower than for any of the other algorithms, as seen in table 4.8.

Position Algorithm	$+$	$-$	$\cdot$	$/$	bit shift	other
Lateration	1	1	7	1	–	arccos + sin + scene
Trilateration	3	3	7	2	3	unaligned system
Multilateration	7	14	14	2	3	–
LSE	$2(n-1)$	$5(n-1)$	$3n$	–	$2(n-1)$	solve linear equation
Min-max	$2n+2$	$2n$	–	–	2	$2\max+2\min$
Aligning	n	$3n$	$4n$	–	–	$(n)\sin+(n)\cos+4\arctan$

Table 4.8: Number of operations involved in each positioning algorithm.

The LSE technique require some multiplications, and compared to the min-max technique it is expensive. Depending on the application and the de-

mand for precision both LSE and min-max are viable. In high precision application with limited life span the LSE is preferred. When precision is less critical but the WSN is expected to have a life span of several years the min-max method is the better choice. To evaluate the precision in both algorithms they are simulated in matlab on random WSNs in chapter 6.

Chapter 5

Circuit implementations

by Nikolaj Andersen and Håkon K. Olafsen

Body biasing of MOS transistors is an increasingly popular technique, since it allows adjustment of the speed and/or the static power consumption of the devices after production. In this chapter we will explore this technique by measurements on a ring oscillator. In addition a Time Difference Measuring Circuit utilizing the adjustable gate delay in a body biased inverter is implemented and examined.

5.1 Chapter overview

This chapter is split in to main parts. First a ring oscillator consisting of inverters with tunable gate delay is presented, complete with both simulation and measurement results. The tunable gate delay is achieved through Body Bias (BB).

Secondly, a novel Time Difference Measuring Circuit (TDMC), utilizing BB to create small delay differences in parallel delay chains is presented. The primary goal of this implementation is to investigate if the scheme is viable. The achievable time resolution will also be discussed.

The implemented circuits and the content of this chapter is a cooperative work by Nikolaj Andersen and Håkon Olafsen. Both authors has contributed to the entire process, from chip layout to PCB design and measurements. The chapter is also written in cooperation, and is included in both theses.

5.2 Ring oscillator

The tapped delay line is a relatively common structure, used in for instance rake receivers [Limb 05] and radars [Hjor 06]. A delay line is a series of delay elements, usually realized with an even number of inverters. To make the delay through these delay elements tunable after production is a desirable feature in many applications, in addition to being useful as a compensation for process variations. Varying the delay through an inverter by altering for instance the supply voltage or limiting the current is feasible, but by doing this we add additional elements to the signal path and/or reduce the signal swing of the circuit. Instead of adding extra devices to the circuitry, the body of the transistor, also known as the back gate, can be biased to tune the delay through the device.

5.2.1 Introduction

A ring oscillator is a simple structure consisting of an odd number of inverters connected to each other as a ring. It is often used to characterize technologies. This is done by measuring the gate delay through the inverter used in the circuit. It is a well known and widely used circuit when measuring gate delays, and has been used since the early days of integrated circuits [Forb 73, Fang 75].

Delay through digital gates is normally described by the average gate delay τ, which is the average of the rise time τ_{LH} and the fall time τ_{HL} of the output. The average gate delay of the inverters in a ring oscillator can be calculated from the oscillator's fundamental frequency by the following formula:

$$\tau = \frac{1}{2 \cdot N \cdot f} \tag{5.1}$$

where τ is the average gate delay, N is the number of inverter stages and f is the fundamental frequency of the oscillator. Figure 5.1 show a ring oscillator with 5 stages.

5.2.2 Motivation

The chip originally included two different TDMCs, one with analog output from an analog integrator (a capacitor), and one with purely digital outputs. Because of the high time resolution of the circuits, off chip generation of input signals with small enough time difference is a problem as the resolution is in the order of 10-20 ps. To provide the small time difference

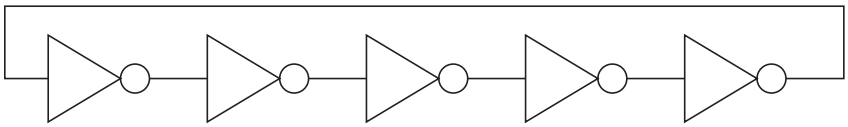

Figure 5.1: Typical ring oscillator with 5 stages

needed, inverters with programmable gate delay was included as initial delays to the analog integration circuit. The purely digital circuit use another type of initial delay, which is described in detail in a later section. To characterize the initial delay inverters, a ring oscillator was included.

A figure illustrating the TDMC with analog integrators can be found in figure 5.2. The main idea is that when a pulse propagate trough each of the delaylines, they will eventually appear at the same delay element at the same time and create a collision. The duration of the collision is stored as a charge in the capacitor. Collisions appearing frequently at the same spot are rewarded, while collisions appearing in random spot are given less weight.

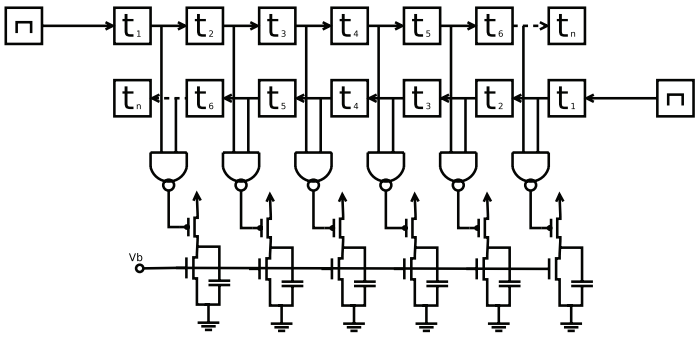

Figure 5.2: Schematics of TDMC with capacitors as analog integrators

To read out the voltages on the different capacitors, an analog multiplexer (figure 5.3) was implemented together with the circuit. Measurements proved that the selector circuit in the multiplexer did not work as expected, resulting in corrupted voltages at the output of the circuit. Because of limited

time, and because the second TDMC circuit has a higher potential resolution, no more efforts was put into getting more results from the circuit.

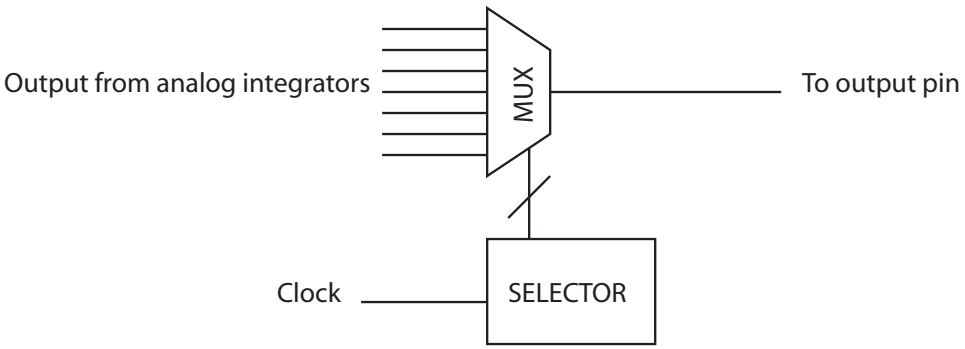

Figure 5.3: Top level of the analog multiplexer

The ring oscillator on the other hand gave promising results. Several aspects of the programmable ring oscillator are of great general interest, and in particular it is of major interest to the second TDMC presented in the last part of this chapter. The following part is devoted to the simulation and measurement results gained from this circuit.

5.2.3 Overview of the circuit

The implemented ring oscillator consists of 201 inverters. It is tapped through a buffer consisting of two inverters. This reduces the extra load on the ring oscillator to one extra gate and only adds a small error in gate delay estimation.

Programmability of the gate delay is achieved through BB of both the NMOS and PMOS transistors. When a positive voltage is applied to the body of a transistor, this is referred to as Forward BB (FBB). When a FBB is applied to the transistor, the gate delay can be decreased due to reduced threshold voltage [Oowa 98, Nare 03]. Similarly, applying a negative voltage to the body results in what is called Reverse Body Bias (RBB). When a RBB is applied, the threshold voltage increase, and thus also the gate delay is increased [Seta 95, Kesh 01]. To separate the body of the transistor from the substrate, the chip is realized in a triple well process. This allows a deep nwell to be used together with an nwell guard ring as isolation. In figure 5.4 a cross section of an NMOS transistor in a deep nwell is shown.

The body bias terminals for the NMOS and PMOS transistors are connected to separate pads on the chip. A transistor with body terminals is sometimes referred to as a four terminal device. The fourth terminal, or the body, of the transistor can be referred to as the back gate of the transistor, because

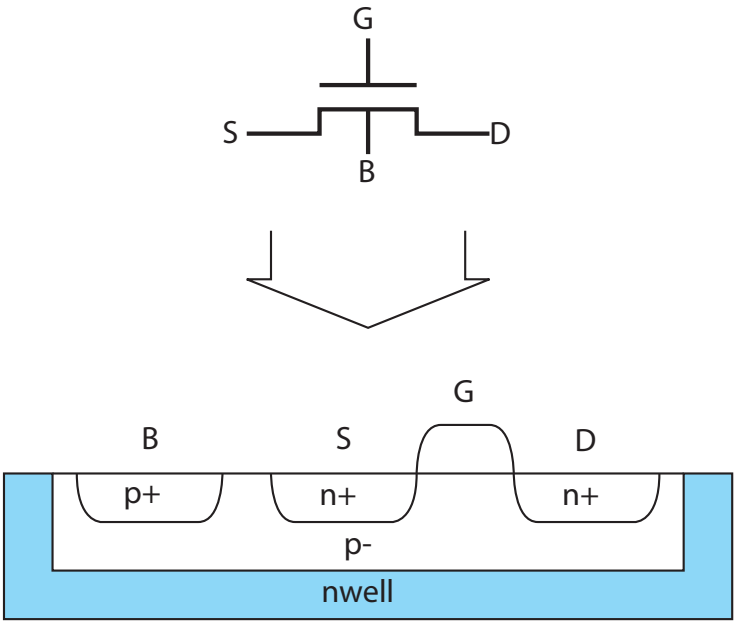

Figure 5.4: Sketch showing the cross section of an NMOS transistor placed in a deep nwell

it behaves similar to the front gate. A more detailed description of body biasing will be given in the following sections.

Body biasing

When applying bias to the transistor bodies, the bias can either be applied to both the NMOS and the PMOS transistors at the same time, or it can be applied to only one of the two. This depends on the intended application of the circuit, and the purpose of applying body biasing. In this circuit, the bias will be applied to both transistors at the same time in a differential manner, meaning that the PMOS bias voltage V_{PMOS} equals $V_{DD} - V_{NMOS}$.

Body biasing is often used to increase the speed or to decrease the off current in digital circuits. Some implementations do both, by applying RBB when the circuit is in standby to gain the best from both. This dynamic threshold voltage concept was introduced in [Seta 95].

It also offers advantages in analog applications, such as reducing output resistance in amplifiers, and it can be used in Voltage-Controlled Oscillator (VCO)s [Wann 00]. The use of the body as an active component is sometimes referred to as active well biasing.

Body biasing can also be used to compensate for process variations, see

[Mele 04, Brya 01, Gran 06].

As already mentioned, the reduced gate delay is a result of reduced threshold voltage. The threshold is roughly proportional to the square root of the body bias through the following formula

$$V_{th} = V_{t0} + \gamma \left[\sqrt{(V_{bb} + 2\phi)} - \sqrt{(2\phi)} \right] \tag{5.2}$$

where γ is a technology dependent constant often referred to as the body effect constant, ϕ is the Fermi potential and V_{bb} is the body bias voltage. Using the Shockley model [Shoc 52] as a first order approximation, the strong inversion source drain current of a transistor in the saturation region is given by

$$I_{DS} = \frac{\beta}{2}(V_{GS} - V_{th})^2 \tag{5.3}$$

$$\beta = \frac{\mu\epsilon}{t_{ox}}\frac{W}{L} \tag{5.4}$$

It has a quadratic relationship toward the effective gate voltage $V_{GS} - V_{th}$. The Shockley model does not model the behavior in weak inversion, but the so-called EKV model proposed by Enz et al. [Buch 98], models the current in weak inversion as an exponential function of the effective threshold voltage.

When a rising voltage beginning at 0 V is applied to an inverter, the NMOS transistor start off in cut off while the PMOS is in strong inversion. As the input voltage rise, the NMOS moves through weak inversion before it enter strong inversion, and for a while both transistors are in strong inversion. After a while the threshold of the PMOS is crossed and only the NMOS remain in strong inversion, while the PMOS enter weak inversion for a while before it reach cut off. As device lengths and supply voltages decrease, the strong inversion part of the total operating area decreases. When the threshold voltage is decreased, the strong inversion part is actually increased.

As the threshold voltage is decreased, the off-state leakage current increase. This current can negate some of the delay improvement from increased on-state current in the transistors in the inverter. It will also lead to higher static power dissipation.

Due to short channel effects in modern CMOS transistors such as velocity saturation and Drain Induced Barrier Lowering (DIBL) the first order Shockley model is inaccurate. In addition, when a positive bias voltage is applied

to the bulk, the effective channel length is increased due to decreased depletion regions around drain and source. This can actually suppress second order effect caused by short channels, and affect the threshold voltage considerably. More accurate models for threshold voltage, taking into account DIBL have been proposed [Liu 93].

Advanced Computer Aided Design (CAD) tools such as cadence use more advanced simulation models such as the BSIM4 model [Hu 03]. They model the body effect more accurately, taking short channel effects into account. In the BSIM4 model, as in most popular transistor models, the effect of body bias is modeled as a change in threshold voltage. In reality, the back gate acts much like the front gate, and inflict on the operation of the transistor in other ways. This leads to a suspicion that the behavior of a body biased transistor may not be estimated accurately by current computer simulation tools, at least not for the purpose we intend here.

The operating area of the back gate is limited compared to the front gate because of the body-drain and body-source junction diodes. The operating voltages of integrated circuits have declined rapidly the recent years, with modern circuits operating at 1 V and below. The threshold of the junction diode is however a constant determined by the material the chip is realized in (approximately 0.65 V for silicon). This means that the relative operation area of the back gate is higher for lower supply voltages.

As the bias voltage is increased, the power consumption is expected to increase [Liu 00]. This increase can be split into two regions. For low V_{bb}, the current consumption comes from off-state leakage and increased parasitic capacitances due to shallower depletion regions. Once V_{bb} reaches the threshold voltage of the bulk-drain and bulk-source diodes, which is approximately 0.6 volts, the power consumption will increase rapidly due to the forward biased diode current. This can also negate the delay improvement as the bulk-drain current increase and reduce the output swing of the inverter. For these reasons, the bias voltage V_{bb} should be kept below 0.6 volts [Wann 00]. In [Nare 03] an optimum bias voltage of 500 mV is found using measurements in a $0.18\mu m$ CMOS process. Further increase bias voltage only results in decreased performance and rapid increase in power consumption.

5.2.4 Schematic

The inverter used in the ring oscillator is based on a standard inverter from the STM 90nm library, with a fan-out of 1. At minimum length the gate delay is simulated to be under 14 ps with a single inverter as load. Increasing the length of the transistors drastically increases the delay, a times four increase in length leads to an increase of almost eight in gate delay.

Gate delay						
Length	0 mV	300mV	500mV	600mV	700mV	σ @ **0mV**
0.1	13.68	12.77	12.58	12.7	13.04	0.33886
0.2	32.55	30.09	29.25	29.23	29.6	0.509
0.3	61.2	56.77	55.02	54.75	54.96	0.824
0.4	100.43	93.3	90.36	89.75	89.63	1.18618
0.5	151.5	141.1	136.5	135.3	134.7	1.61685

Table 5.1: Simulated gate delay for different FBB voltages and transistor lengths. All delays are in ps

	length	**width**
NMOS	0.400	0.600
PMOS	0.400	0.800

Table 5.2: Transistor sizes used on the chip

In table 5.1 simulation results for different transistor lengths and FBB voltages are shown. For all transistor lengths, the NMOS width is $0.6um$ and the PMOS width is $0.8um$. As mentioned before, the inverters in the ringoscillator was originally designed to create inputs for a TDOA measuring circuit. Since the relative delay difference achieved in a long device is higher than for shorter devices, the implemented transistors were chosen longer than minimum. The same total delay difference could have been achieved using a higher number of shorter transistors.

In figure 5.5 delay through the implemented inverter as a function of body bias voltage is plotted, normalized to zero bias. From the plot, there are several interesting observations to be made. As expected, the delay decreases with increased bulk voltage. The decrease is not linear, it seems more like a quadratic relationship with bulk voltage. At about 0.7 volts, the decrease stops. This is also expected, because of increasing bulk drain current and leakage through the closed transistor.

There is however a major discrepancy between the results from [Nare 03] and the simulations provided here. Instead of an increase in delay after the minimum point, the simulated delay actually decrease. The most probable explanation of this is that the transistor model used in simulations does not model the behavior of the back gate correctly. Operation of the back gate of transistors in the area above the threshold voltage of the body-drain diode should in any case be avoided since it leads to excessive power dissipation.

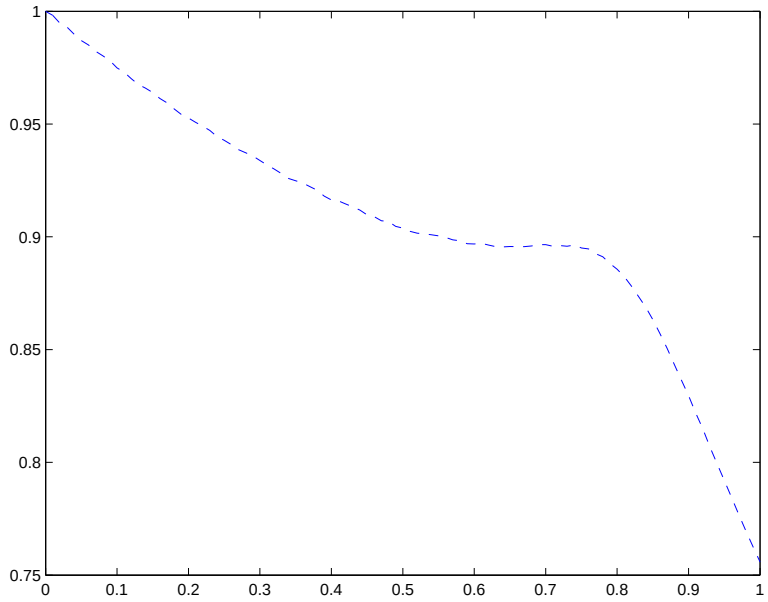

Figure 5.5: Simulated delay versus body bias, normalized to peak delay at zero bias

5.2.5 Layout

The layout of the circuit is fairly straight forward. As already mentioned, the NMOS transistors are placed in wells. These wells consist of a deep n-well, an n-doped layer beneath the p-doped substrate, forming the bottom of the well. The walls of the well are made of n-doped substrate, overlapping the deep n-well in the bottom to seal the well. See figure 5.6 for an illustration of the cross section of the inverter. The layout of the inverters and the complete ring oscillator circuit can be found in appendix A.

5.3 Ring oscillator measurements

5.3.1 Measurement setup

The measurement setup consist of a PCB, an oscilloscope and voltage sources for biasing and power. The oscilloscope is connected to the PCB via a coaxial cable, and the power supply through BNC connectors soldered to the PCB. During early measurements, we observed higher harmonic frequencies at odd levels 3-7 times higher than the expected oscillation frequency. As it turns out, this is a quite common problem, see for instance [Sasa 82] where harmonic frequencies is observed in a 101-stage ring oscillator. The solution is to slowly ramp the supply voltage in small steps.

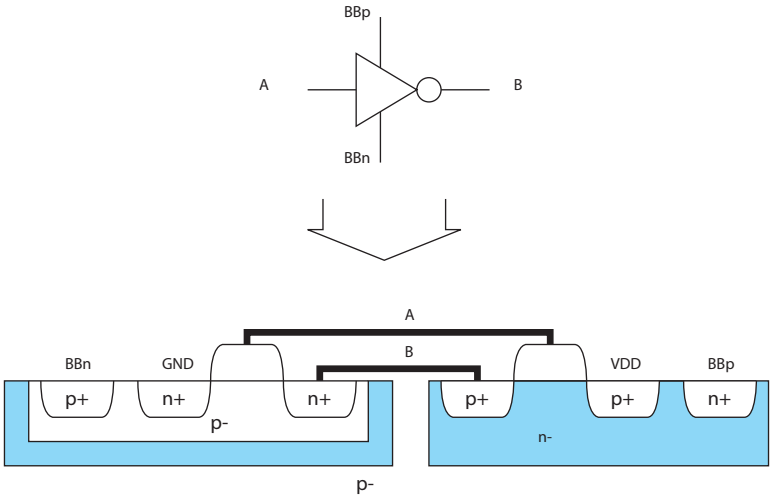

Figure 5.6: Sketch showing the cross section of the implemented inverter

Results are read out from the oscilloscope using GPIB with matlab.

Pictures of the PCB with the mounted chip is provided in appendix B.

5.3.2 Introduction

The measured waveform is shown in figure 5.7. The measurements were repeated on 20 different chips. The resulting measured frequencies are plotted in figure 5.8 for all chips. Two of the chips showed a major deviation in frequency from the rest. Since the chips all come from the same wafer, the most probable explanation to this is mismatching within the wafer.

In figure 5.9, the measured average gate delay is compared to the simulated results. The two deviating chips have been removed from the collection, leaving a remaining 18 chips to average over. The two most obvious differences from simulations is the rather big difference in simulated and measured frequency, and the way the frequency actually start to decrease after reaching a maximum. This is in contrast to the simulations, where the slope flattened, before it actually decreased further.

5.3.3 Gate delay measurements

Measured gate-delay

The simulations indicated a fundamental frequency of about 25 MHz at zero biasing, while the measurements show that the real frequency is ac-

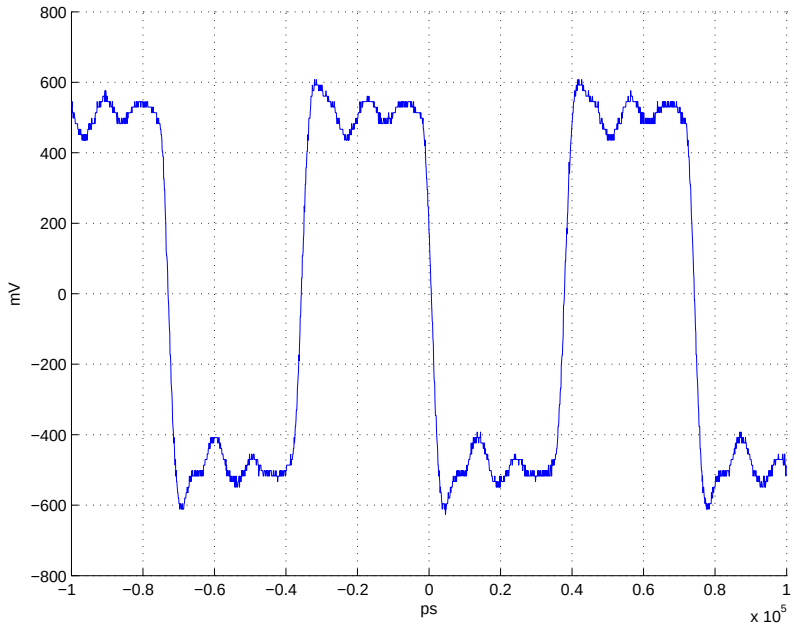

Figure 5.7: Measured waveform at the output of the ring oscillator

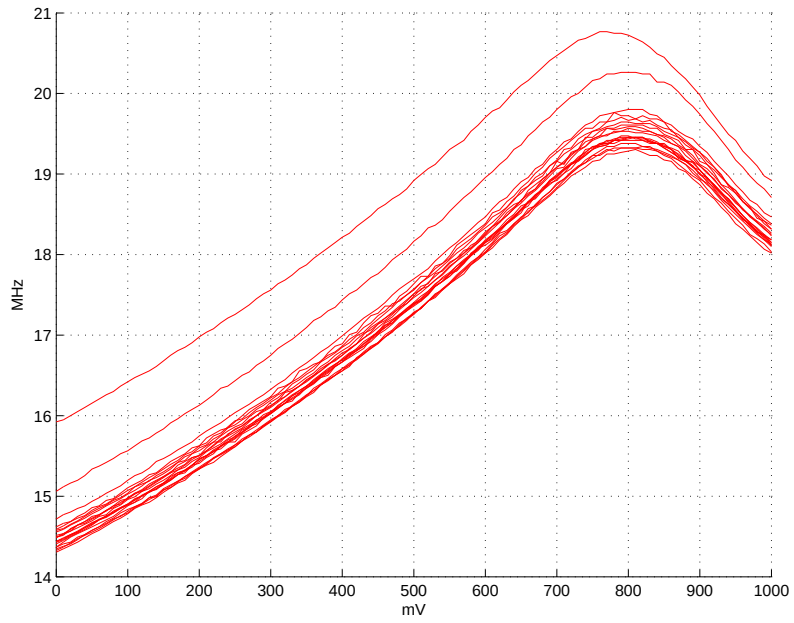

Figure 5.8: Measured frequency on the y-axis and BB voltage on the NMOS transistor along the x-axis

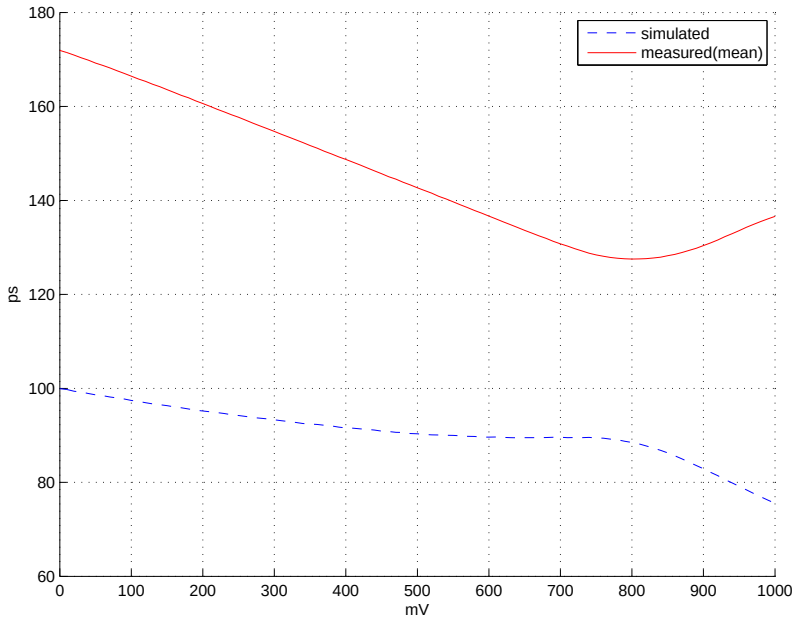

Figure 5.9: Average gate delay for measured and simulated behavior versus BB on the NMOS transistor

tually below 15MHz. This is a 40 percent drop in frequency. The simulations were done without any extracted parasitics, so that could explain at least some of this difference. To investigate the large difference in simulated and measured gate delay, the impact of load capacitance is investigated below. The estimations are based on the first order Shockley model [Shoc 52]. These are widely used in Metal-Oxide-Semiconductor Field-Effect Transistor (MOSFET) analysis, and are used in many textbooks [West 94, John 97]. From [West 94] we get that the transition time of a CMOS inverter can be approximated as:

$$t_{LH}, t_{HL} = k \cdot \frac{C_L}{\beta \cdot V_{DD}} \tag{5.5}$$

$$\tag{5.6}$$

where β is the gain coefficient of the transistors given by:

$$\beta = \frac{\mu\epsilon}{t_{ox}} \frac{W}{L} \tag{5.7}$$

$$\tag{5.8}$$

with ϵ being the permittivity of S_iO_2, t_{OX} the thickness of the gate insulation, and μ the electron or hole mobility in the channel of NMOS and PMOS

devices. k is a constant given by:

$$k = \frac{2}{1-n}\left[\frac{n-0.1}{1-n} + \tfrac{1}{2}ln(19-20n)\right] \tag{5.9}$$

with $n = V_{th}/V_{DD}$

The period of the oscillations in the ringoscillator is the sum of rise and fall time. Adding the two result in the following equation for oscillator fundamental period:

$$t_{period} = t_{HL} + t_{LH} = \frac{C_L}{V_{DD}}\left[\frac{k_n}{\beta_n} + \frac{k_p}{\beta_p}\right] \cdot N \tag{5.10}$$

Where N is the number of inverters. For our purposes the gate delay is more interesting. The gate delay is given by the average of the rise and fall time of the inverter:

$$t_{gate} = \frac{t_{period}}{2N} \tag{5.11}$$

Rearranging equation 5.11 gives the following equation for load capacitance as a function of gate delay:

$$C_L = \frac{t_{period}V_{DD}}{2}\left[\frac{k_n}{\beta_n} + \frac{k_p}{\beta_p}\right]^{-1} \tag{5.12}$$

Using this equation results in a mean load capacitance of approximately 21fF.

The layout of the ringoscillator is shown in figure A.2. The circuit was laid out as one long structure, with one inverter following the next with minimal wiring between the stages. However, the feedback wire is rather long, stretching the entire length of the oscillator, and is the largest single contributor to parasitic capacitance in the complete circuit. Extracting parasitics from cadence shows that connections between the inverters add about 2.2fF of parasitic load, while the long feedback wire add 27.5fF. Because the relationship between delay and load capacitance is linear, the large capacitance from the feedback wire can be split into smaller capacitors and distributed to all the inverters in the circuit. This makes the extracted values easier to compare with the measured mean value. Doing this adds only 0.14fF to each of the inverters, meaning the parasitic capacitance of the feedback wire adds little to the total delay.

Parameters	nMOS	pMOS
V_{th}	0.24	0.29
V_{DD}	1	1
ϵ	$3.9\epsilon_0$	$3.9\epsilon_0$
t_{ox}	16A	16A
μ	500	180

Table 5.3: Transistor parameter values

Another major contributor to additional delay is resistance in poly and metal wiring. It is not included in the estimation of load capacitance conducted above. Extraction of parasitic resistances in cadence reveal significant amounts of parasitic resistance, particularly in the contacts between the metal wiring and the gate poly and active drain area of the preceding transistors. This is also confirmed by a closer reading of the process parameters provided by STM. According to these parameters a typical metal to polycide contact has a resistance of 17 Ω

Noise in supply voltage due to noise generated by the oscillator when the inverters switch is also a potential error source. Because the on chip wiring and bonding wires also acts as parasitic inductors, sudden changes in current such as those generated by a switching inverter will induce a certain voltage drop, which in turn could affect the speed of the oscillation. In addition the voltage drop over the resistive elements in the on chip wiring might contribute. Adding better decoupling, both on chip and off chip, and putting more work in the supply rails wiring will reduce this problem. This effect is also known as IR-Drop [Khal 06].

Maximum oscillator frequency

As mentioned earlier, the frequency of the ring oscillator only increase to a certain point. In [Nare 03] this is pointed out, and an optimal, or maximum, bias voltage of 500 mV is found. Simulations with minimum length transistors in 90nm give a similar result. Simulations also indicate that longer transistors allow the FBB to be adjusted higher than for minimum length transistors. In both simulations and in [Nare 03], a maximum delay improvement of approximately 10% is found. Measurements however, show that the FBB can be even more effective, with a 25% increase in frequency at max.

In figure 5.9 the measured mean delay is plotted along with the simulated delay. When the two curves are compared (see figure 5.11 for a normalized plot) we see that the measured circuit display some interesting characteristics. The body can be tuned to over 750 mV(PMOS = 250 mV and the delay

still decrease. Since this is above the threshold of the bulk-source and bulk-drain diodes the power consumption is expected to increase rapidly. In figure 5.10 a plot of the measured current consumption of the chip is plotted with the NMOS bias voltage. The absolute current is not interesting here, since the V_{DD} and GND connections are shared by multiple circuits on the chip. It does however confirm the hypothesis that the circuits draw considerably more current when the bias reaches the diode threshold voltage.

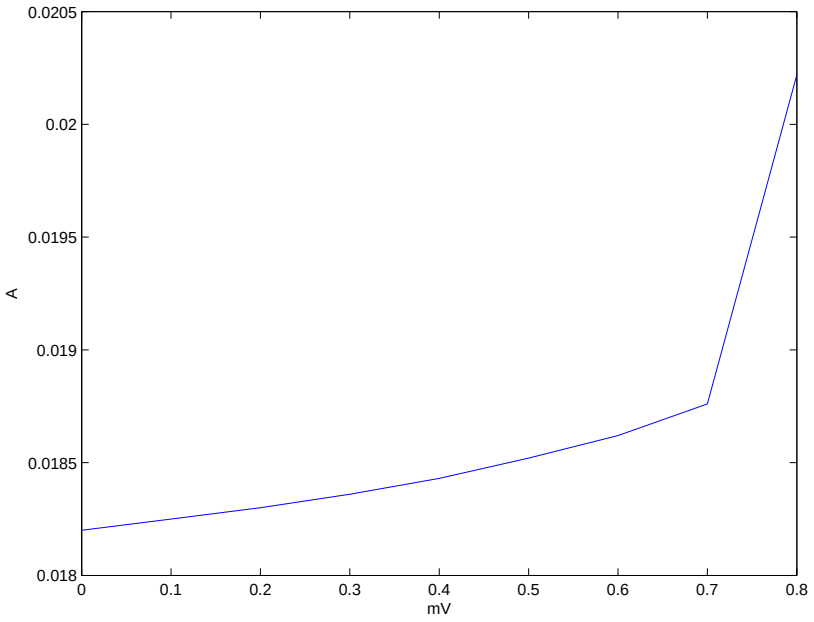

Figure 5.10: Current drawn by chip for different bias voltages. Notice the steep increase in current as the voltage pass the diode voltage of approximately 0.65 V

Process variations and mismatch

An interesting parameter that can be derived from the measurements is the amount of mismatch between the individual chips. A total of 20 chips were available for measurement, and as stated earlier two of them showed a significant deviation from the rest. They have been excluded from the samples, and the standard deviation in delay in the remaining 18 chips is found. In figure 5.12 the standard deviation is shown in seconds and percent of total delay as a function of body bias voltage. In figure 5.13 a linear fit to the standard deviation is shown, together with the residue of the fit. This is to show that the deviation seem to decay linearly with body voltage. Because of the low amount of current passing through the transistors in weak inversion compared to strong inversion, mismatching and process

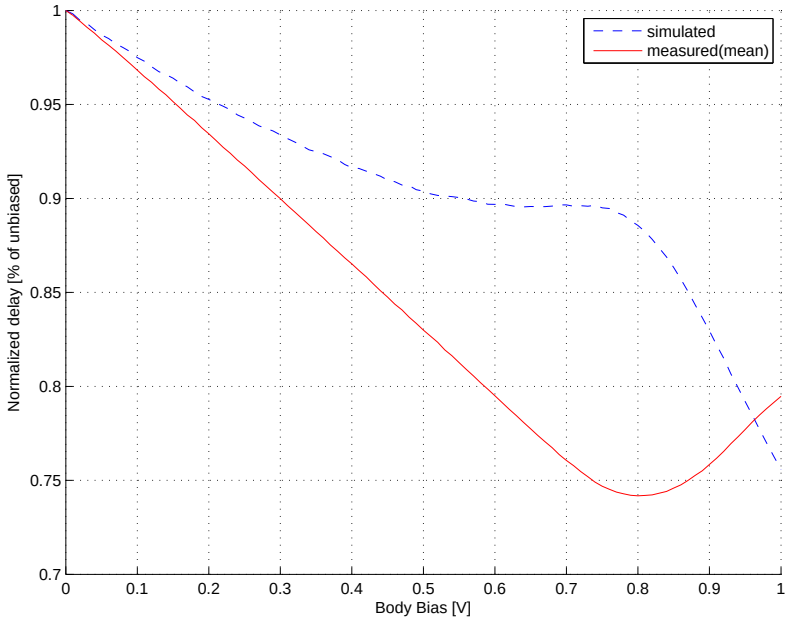

Figure 5.11: Comparison of simulation and measurement data, both curves normalized to their respective peak delay

variations are more pronounced in weak inversion. As explained earlier in this chapter, a body biased transistor has a larger strong inversion area, and this could explain the decay in standard deviation.

The standard deviation in seconds at zero bias is approximately 1.3 ps, which is not far from the simulated value at 1.2 ps. In percent of the total delay however, the deviation is far lower in the measurements (0.76%) than in the simulations (1.18%).

From a statistical point of view, 20 samples is a low count to base a conclusion upon. The simulated standard deviation is found through monte carlo simulations of mismatch with 1000 iterations. It does however give us an indication on the mismatch between the units. We see that the simulated deviations provide a rather good estimate of the actual deviations seen in measurements.

5.3.4 The oscillator as a VCO

When examining the first part of the measured curve in figures 5.11 and 5.14, it seems to display an almost linear behavior. In VCO's, this is often a desired property since it makes the VCO behavior predictable. In this section the linearity of the oscillator is explored.

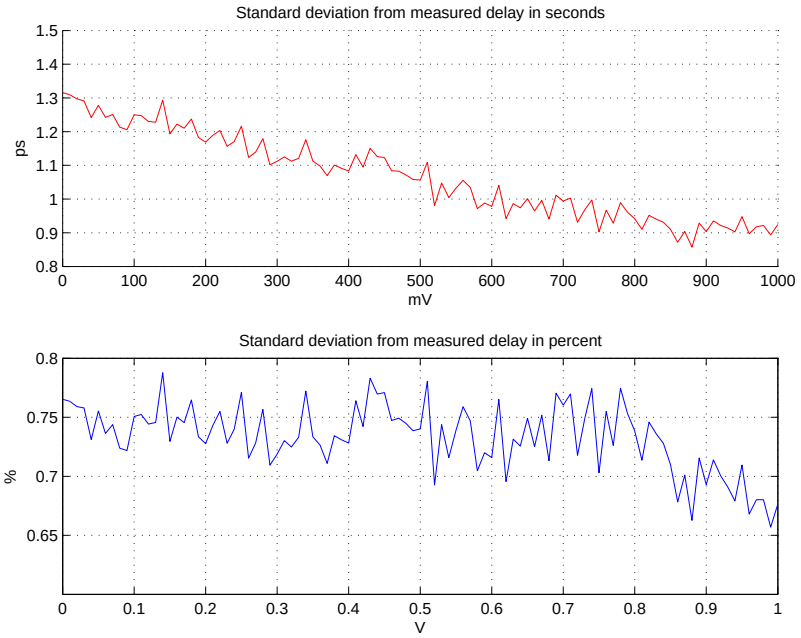

Figure 5.12: Standard deviation from measured delay in seconds, based on measurements on 18 different dies.

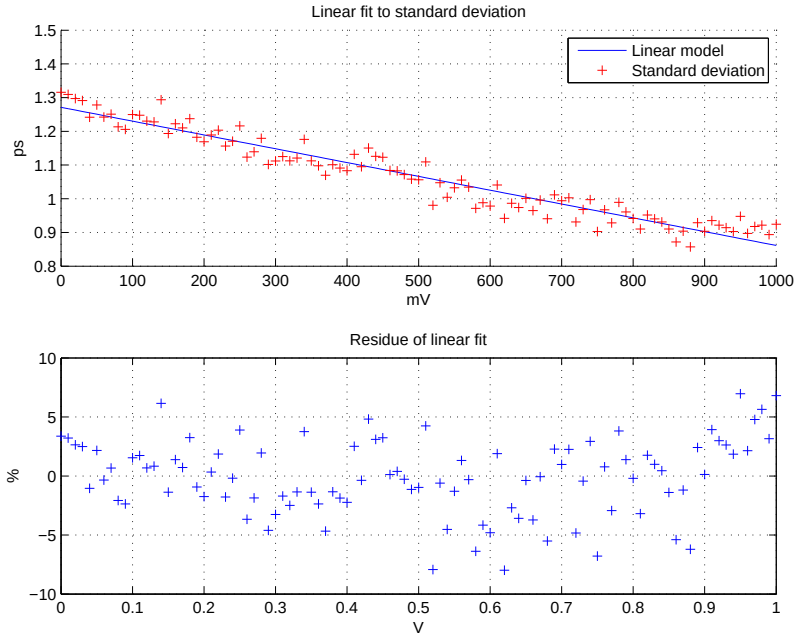

Figure 5.13: Linear fit to standard deviation in measurements

To investigate the linearity of the oscillator, a first degree polynomial fit was performed in matlab. Because of the nonlinear behavior displayed at high biases, the polynomial fit is performed only on the first part of the curve. In figure 5.14 a plot of the linear fit is shown together with the residue from the fit, showing the error of the fit. It is clear that the curve is not entirely linear. The shape of the error curve indicates that a higher degree polynomial might fit the curve better. A second degree polynomial fit is shown in figure 5.15 together with its corresponding error curve. The seemingly random distribution of error indicates that the second degree polynomial is the best fit, at least for lower bias voltages. The quadratic behavior shown by the VCO is however weak, since the error in the linear fit is 1.5% at most. This weak quadratic tendency can be related to the quadratic behavior of the drain source current in the saturation region. In fine pitch processes this quadratic behavior is significantly reduced due to velocity saturation, explaining the close to linear behavior.

From around 600 mV the curve seems to take another shape, but this is as expected since it is close to the diode threshold. One would typically aim at keeping the bias voltage below this threshold in an implementation to avoid excessive power consumption.

Another important property of VCOs is phase noise. Because of time constraints, the degree of phase noise in the implemented oscillator has not been measured. Ring oscillators do however often tend to inhibit a certain amount of phase noise. Further study of the VCO properties of the body biased ring oscillator is therefore required before a conclusive description of the performance of the oscillator as a VCO can be made.

5.3.5 Summary

The measurements have provided several interesting observations. First of all, the absolute gate delay of the inverters deviates considerably from the schematics simulations. Extracting parasites from layout improve the simulation considerably, but there are still significant discrepancies between measured and simulated delay. Some of this comes from other sources than just parasitic capacitors and resistors, with IRdrop as a potential candidate. Since no special care was taken during layout regarding routing of the supply rails, it is possible that it contributes to the decreased speed of the oscillator.

As mentioned in the beginning of this section, we suspected that there might be noticeable differences between the simulated and measured behavior of the back gate. The measurements confirmed these suspicions. With the effect shown in these measurements, the back gate can probably

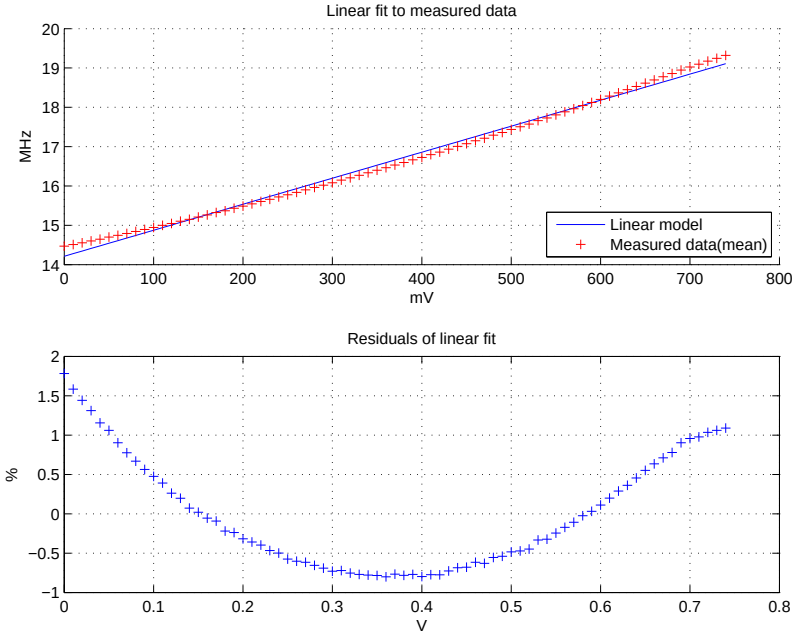

Figure 5.14: Linear fit to measured frequency

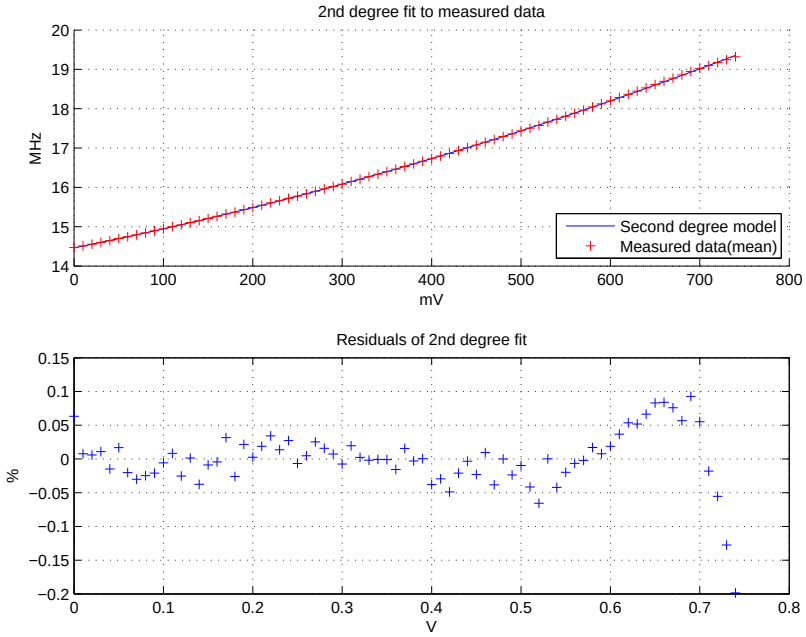

Figure 5.15: Second degree polynomial fit to measured frequency

be used in several applications. It can provide weak quadratic behavior in a VCO and it has great potential as a tunable delay element.

5.4 Time Difference Measuring Circuit

There are two fundamental techniques when measuring time. The first is to measure absolute time compared to a clock and the other is to use relative time measurements. Absolute time measurements rely on synchronized clocks, which is hard to achieve when the devices are spread over some distance. GPS use absolute clocks as references and is dependent on atomic clocks with high accuracy to stay synchronised. Implementing high precision clocks in WSN nodes is not a feasible option, making them rely on imprecise in-field synchronisation.

Relative time measurements are done by measuring the time difference between two or more signals. By using relative time measurement techniques, all the time critical parts can be placed in the same unit. This makes it easier to control the environment for the time measurement, and hence increase the accuracy.

The main idea with this Time Difference Measuring Circuit (TDMC) is to use a small relative delay difference in two delay elements to measure a relative time difference between two pulses. By using this relative delay difference we hope to achieve a time resolution smaller than the gate delay. The delay elements are ordered in two parallel tapped delay lines of N elements, one with *normal* delays and a second with slightly *faster* delays. The difference in delay is called $\Delta\tau$. Detecting a time difference is done by sending the first pulse through the slow tapped line and the second pulse through the fast tapped line. After n delays the second pulse will overtake the first. By determining the number of delays (n) before this happens and knowing the $\Delta\tau$ we can estimate the time difference (τ_i) between the two pulses. In figure 5.16 the two delay chains are shown.

To determine where the second pulse overtakes the first we need some sort of detector. The output of the detectors is a function of the three inputs τ_i, $\Delta\tau$ and n. The output of the detectors is represented by:

$$\begin{aligned} \text{for } n &\in [0, N] \\ f(n, \Delta\tau, \tau_i) &= \begin{cases} 1 & , \tau_i \geq \Delta\tau \cdot n \\ 0 & , \tau_i < \Delta\tau \cdot n \end{cases} \end{aligned}$$

This shows that the output of the first n detectors is $f = 1$, while the remaining $N - n$ is $f = 0$. In other words, the output is stored as a ther-

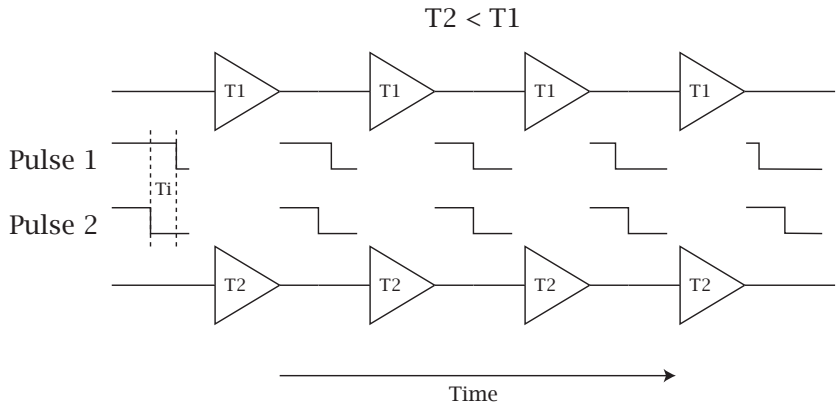

Figure 5.16: Relative delay difference in parallel chains.

mometer code. n_e is the thermometer output and relates to the inputs as follows:

$$\tau_i = \Delta\tau \cdot n_e \tag{5.13}$$

By solving this, the time difference between two incoming signals can be estimated

5.4.1 Motivation

In earlier literature scaling has been used to create the relative delay difference of two delay elements. [Abas 04] By doing this it is not possible to tune the circuit after production. By using Body Bias (BB) it is possible to tune and change the relative delay difference between two identically scaled devices. The possibility of tuning the delay in a delay element after production was the main motivation for creating this circuit. The goal is to investigate the achievable relative precision using this technique.

Measuring a time difference between two pulses can be used in several applications such as VCOs, to determine AoA or for ranging purposes. This chapter will treat ranging in particular.

Achievable ranging accuracy

Ranging using RF signals is based on the propagation of electromagnetic radiation. Electromagnetic radiation propagate with a speed close to 0.3 mm/ps

in vacuum, which in terms of ranging can be considered a worst case scenario. In the TDMC presented here, the accuracy is given by the difference in delay between two delay elements. If one element is tuned 5 ps faster than another, this yields a spatial resolution of:

$$\begin{aligned} c &= 0.3 \text{ mm/ps} \\ \Delta\tau &= c \cdot 5 \text{ ps} = 1.5 \text{ mm} \end{aligned}$$

The achievable ranging accuracy can potentially be in the order of a few millimeters, which is a significant improvement over existing ranging systems.

5.4.2 Overview of the system

The TDMC use two parallel tapped delay lines with BB tuning as the main feature. The detector consist of one d-flip-flop with the D-input tapped from the slow delay line and the Clk-input from the faster delay line. This implements the thermometer code where the first n d-flip-flop outputs are high, see figure 5.17.

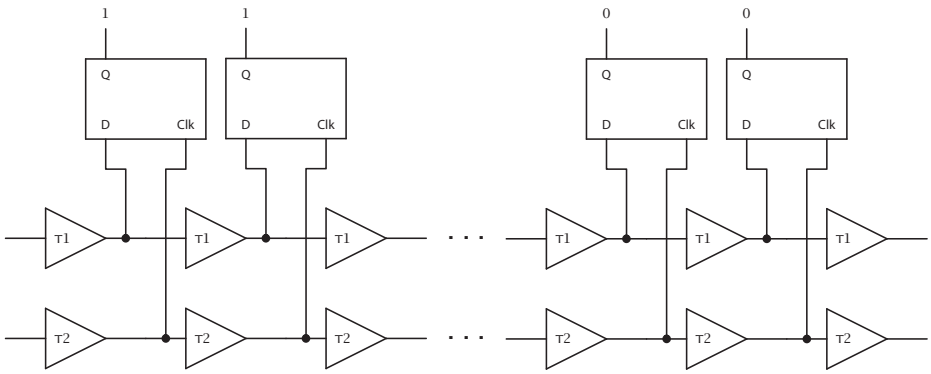

Figure 5.17: Show the implemented TDMC with detectors.

The delay chains and the d-flip-flops are the key elements in the TDMC. When the measurement is completed, the next step is to decode the thermometer code. On this chip there is no need for the information on-chip so it is shifted into a shift register which then can be shifted out on a single pad. By sending the output from the last d-flip-flop back into the first, the thermometer code can be read with an oscilloscope. The signal will have a period equal to the number of d-flip-flops times the input clock period.

The duty cycle of the signal represent the thermometer code. If the duty cycle is 50% then half the detectors have a high output.

A problem when characterising the TDMC is to create the two input pulses. Creating two pulses with a time difference in the order of picoseconds is very difficult off-chip. The solution is to create two pulses on-chip using a pulse generator circuit. The input is a single pulse, which is then split into two different pulses with a small time difference.

The complete structure for the TDMC with input and output systems is shown in figure 5.18.

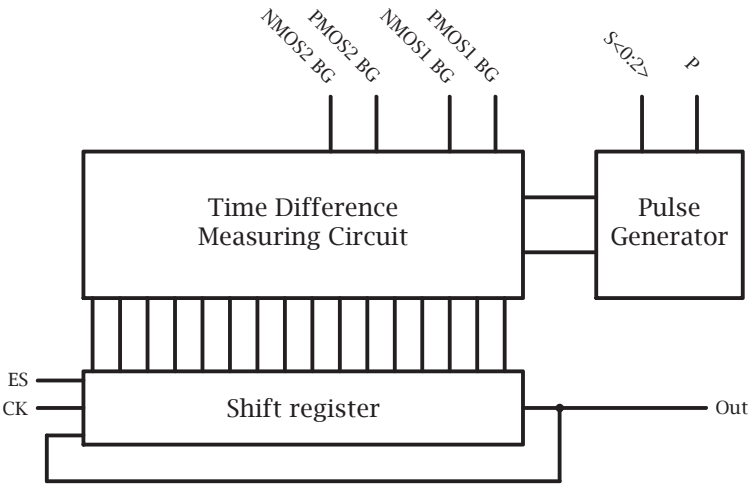

Figure 5.18: Schematic for complete circuit.

5.4.3 Implementation

The implementation of the TDMC is focused on proving the concept using tapped delay lines and d-flip-flops as detectors. Throughout our simulations and measurements the slow tapped delay line will not be biased while the faster will be. When BB is used later it will always refer to the biasing of the faster tapped line.

The dynamics of the TDMC is the combination of $\Delta\tau$ and the number of elements in the line and the measurable time range achieved. By using two tapped delay lines of length $N = 32$ the dynamics are good. In figure 5.18 the S<0:2> input to the pulse generator is to a decoder that selects one of eight different time differences. The time differences created by the pulse generator have a range of about 50 ps, see table 5.5. With this input range

of τ_i the dynamics of the TDMC can be thoroughly tested. The implemented system consists of the pulse generator, two tapped delay lines with 32 elements, 32 detectors and a 32 bit shift register. The different parts will be discussed shortly.

Figure 5.19 show the timing diagram for the entire circuit. This generic timing diagram show how the inputs and output is related for a given S<0:2> and BB combination.

CK is the clock input to the shift register.

P_{in} is the pulse input to the pulse generator.

ES is the enable shift signal to the shift register.

Out is the output from the shift register.

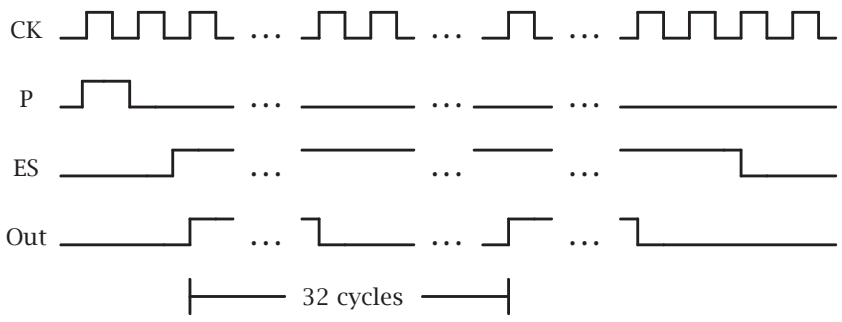

Figure 5.19: Timing diagram for the implemented TDMC test circuit.

Pulse generator

Creating a very small time difference between two pulses is not that easy in CMOS. One solution would be to use two tapped delay lines and then select the pulses after M elements and send this into the TDMC. This would result in delay elements both creating and measuring the time difference. M elements in the pulse generator would result in N elements in the TDMC. For this to be useful the two chains would have to be scaled differently or biased differently. Any error in the TDMC would be correlated and suppressed by an error in the pulse generator. Without any earlier experience with Body Biasing this seemed like a bad idea, instead a solution using RC delays is implemented. Combining buffers and varying RC delay to

create the different time delays make sure the generator and TDMC are independent. This also means the pulse generator does not need any biasing reducing the number of external pads.

In the final pulse generator a long unsilicided N+ poly resistor is used in the RC delay. The capacitor is the parasitic capacitance from the resistor to the bulk. The different time delays are tapped after different lengths of this poly resistor, creating some non ideal effects. These effects come from non-linear changes in the resistor caused by temperature changes and process mismatch and the output from RC delays.

Delay element

The delay element is of utmost importance to the performance of the TDMC. A good delay element for the TDMC will have a small absolute standard deviation caused by mismatch. Preliminary simulations showed that smaller delay elements would yield smaller standard deviations.

In table 5.1 we see that an inverter can have a standard deviation of 0.34 ps. The standard deviation would transform directly into the ranging precision. From this we can conclude that shorter inverters are good for the precision. Since this circuit merely is a proof of concept, longer transistors are used. The minimum length of transistors would require very small time differences from the pulse generator. By lengthening the transistors and increasing the effect of the body bias the constraints on the pulse generator is reduced.

In table 5.4 simulations for different delay elements are listed. Each delay element consists of two inverters scaled as input and output stage, both listed in the table. The different delay elements have different sizes and number of fingers.

All the simulations have been performed with 0 V input to the back gate using BSIM3 model.

Name	Sizes (L · W) in µm				Monte Carlo	
	Input stage		Output stage			
	nMOS	pMOS	nMOS	pMOS	μ	σ
delay01	0.5 · 4	0.5 · 6	0.5 · 10	0.5 · 14	357.16 ps	1.9 ps
delay01wbg	0.5 · 4	0.5 · 6	0.5 · 10	0.5 · 14	345.13 ps	1.9 ps
delay04wbg	0.3 · 2.4	0.3 · 3.6	0.3 · 6	0.3 · 9	145.4 ps	1.26 ps
delay04	0.3 · 2.4	0.3 · 3.6	0.3 · 6	0.3 · 9	156.58 ps	1.56 ps
delay04wbghvt	0.3 · 2.4	0.3 · 3.6	0.3 · 6	0.3 · 9	229.96 ps	2.41 ps
delay04hvt	0.3 · 2.4	0.3 · 3.6	0.3 · 6	0.3 · 9	239.53 ps	2.5 ps

Table 5.4: Delay element sizes & Monte Carlo using BSIM3.

With all this in mind the delay element *delay04wbg* was chosen to be used in our implementation. The schematic for the delay element is shown in figure 5.20.

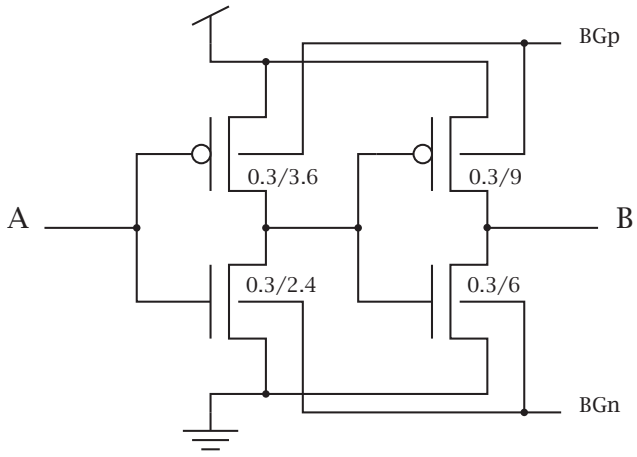

Figure 5.20: Schematic of the delay element with sizes.

D-flip-flop – the detector

The detector, a d-flip-flop in this implementation, is also a critical component. A edge triggered flip-flop is the architecture used in our implementation. It has to be edge triggered to work, since the goal is to detect the edge of the pulse. The timing is the critical property since the main goal is to accurately determine the number of delay elements before the pulses pass each other. The required setup time for the data signal should be as small as possible and controllable. If the two pulses arrive at a d-flip-flop at exactly the same time the d-flip-flop should ideally go high. Because of the setup time, this will not happen in the implemented system. The setup time for the signal will add a small negative time error to the estimate and should be corrected when estimating the time difference.

Any mismatch on the same die should also be avoided. The flip-flop architecture should be robust enough to handle mismatch, and this should be considered during the layout. This was taken lightly in this implementation and is a source of errors in measurements. Due to time limitations an existing d-flip-flop implemented by Hans Berge was used.

5.4.4 Expected results

With the main focus on body biasing combined with a proof of concept, and time limitation during production, extensive simulations were not possible. Some preliminary simulations were done on the different components and a few simulations on the entire circuit to test the maximum and minimum inputs. Based on these simulations the circuit is expected to work with all inputs as long as $BB/\Delta\tau$ is large enough.

After the production more thoroughly simulations have been completed and some unexpected results were found. First the pulse generator was not tested enough and have some weaknesses found with Monte Carlo simulations. Secondly the d-flip-flop was not simulated as a detector, but it was characterised to handle a clock frequency of approximately 2.5 GHz by Hans Berge. In this section the different components are studied closer. At the end the expected achievable accuracy is discussed.

Pulse generator

Simulation results from the pulse generator schematics are seen in figure 5.21. The time difference between the pulses can be varied from approximately 125 ps to 175 ps, a variation of 50 ps. The Monte Carlo results are found in table 5.5, where the first weakness is seen. Looking at the figure and the table we see that the generated delays are not linear. That means special consideration most be taken when evaluating the simulations and measurements. By using these simulation results as the estimated points along the x-axis the results is adjusted regarding the x-axis and more readable. This is not one hundred percent accurate, but it is a better approach than using linearly spaced points along the x-axis.

This phenomena is caused by the use of RC delay creating a ramp function. By looking at the different taps this ramp function can be compared to different time constants. The time constants will increase further down the line but the increase is not linear. The changes are smaller for larger τ_i. This will increase the impact of noise on the signal and should result in an increased standard deviation in measurements for larger τ_i.

The second problem with this implementation is the standard deviation found running Monte Carlo simulations. In table 5.5 a standard deviation of approximately 6 ps is observed for all the created time differences. These Monte Carlo simulations are done with 250 iterations and mismatch only. Simulations with process variations and mismatch showed a standard deviation $\sigma =$ 10-12 ps, which is large compared to the accuracy we hope to measure.

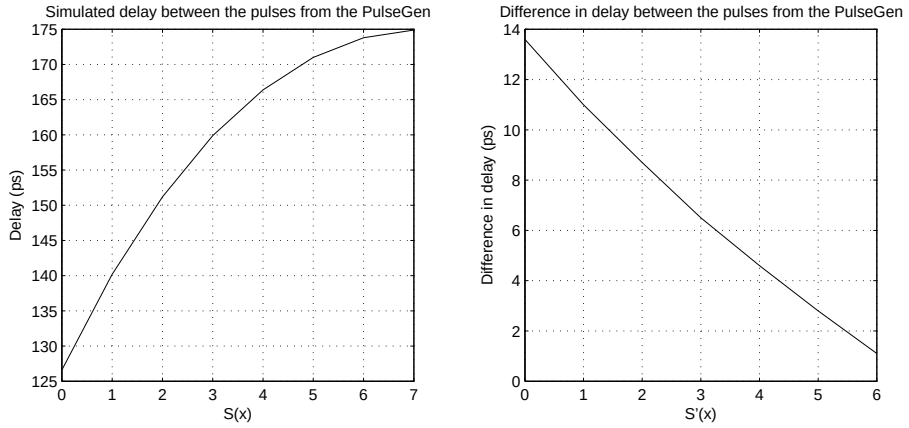

Figure 5.21: Simulation results for the pulse generator.

S<0:2>	τ_i	σ	$\Delta\tau_i$
000	126.8 ps	6.27 ps	–
001	140.5 ps	6.01 ps	13.7 ps
010	151.5 ps	5.85 ps	11 ps
011	160.4 ps	5.91 ps	8.9 ps
100	166.3 ps	6.06 ps	5.9 ps
101	171.2 ps	6.00 ps	4.9 ps
110	173.3 ps	5.76 ps	2.1 ps
111	174.8 ps	5.78 ps	1.5 ps

Table 5.5: Monte Carlo simulations for the pulse generator with 250 iterations and mismatch only, using BSIM4.

Delay element

Simulation results for the used delay element can be seen in table 5.6 and figure 5.22. The table show the effect of the back gate on the average delay and the standard deviation. The results deviate from the first simulations in table 5.4 since a new improved model (BSIM4) is used. The BSIM4 simulations differ from the BSIM3 simulations with lower $\Delta\tau$ but significantly higher σ. By sweeping BB from 0 V to 0.5 V the mean delay is reduced by 8.74% and the standard deviation by 18.65%. This shows that the mismatch will decrease when the body bias is increased, as expected. This is also shown in earlier works by Narendra et al.[Nare 03]

The Monte Carlo simulations used mismatch and process variations and 1000 runs on the BSIM4 model.

BB	μ	σ	$\Delta\tau$
0.0 mV	115.882 ps	5.00024 ps	0 ps
0.1 mV	112.918 ps	4.73094 ps	2.964 ps
0.2 mV	110.46 ps	4.51298 ps	5.422 ps
0.3 mV	108.433 ps	4.33084 ps	7.449 ps
0.4 mV	106.882 ps	4.1847 ps	9.000 ps
0.5 mV	105.748 ps	4.06785 ps	10.134 ps

Table 5.6: Monte Carlo simulations for the delay element with 1000 iterations, mismatch and process variations, using BSIM4.

As we can see in figure 5.22 the change in delay is close to linear with back gate input. We also see that the simulations show that we could achieve an accuracy of only a few picoseconds by biasing the delay chains with about 0.1 V difference.

Grey-maps

When processing the thermometer code the number n_e is the key information. When measuring the completed circuit, noise is inevitable. The noise is expected to be white Gaussian noise. If the same measurement is done several times, integrating over time, the result should be a Probability Density Function (PDF). The left plot in figure 5.23 is the results from the entire circuit with a BB $= V_{bulk} = 0.1$ V and sweeping the different input time differences. The y-axis is the number of delay elements (n_e) before the second pulse overtake the first. Values along the x-axis are the simulated delay difference between the two pulses from the pulse generator. Starting with equation 5.13 the expression for n_e can be found to be $\frac{\tau_i}{\Delta\tau}$, and the effect of

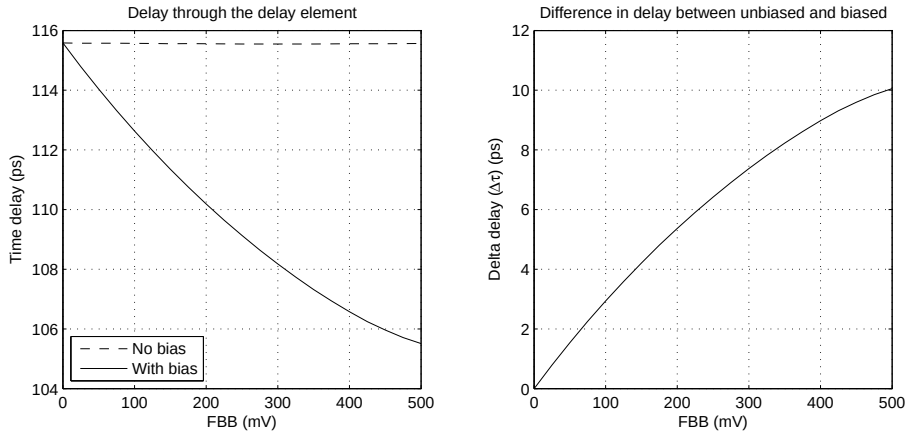

Figure 5.22: Simulation results for the delay element used in the TDMC.

τ_i and $\Delta\tau$ is easily seen. Each column represents a PDF for that time difference τ_i. In this example we see that almost all the measurements are at the same n_e, the black area. A few measurements indicates one more or one less n_e and are indicated by grey, white areas indicates no registered measurement. Light area equals few or none registered measurements, while darker area indicates more measurements. The complete plot is called a grey-map.

The second grey-map indicates the behaviour as a function of the BB input. From this it is possible to se how n_e behaves as a function of BB. This can be compared to the measurements of the ring oscillator and the simulations of the $\Delta\tau$.

Complete circuit

Simulations of the entire circuit were done without Monte Carlo to reduce simulation time. This means only one data set is available for the simulation results which again make them somewhat less informative. All the eight time differences τ_i were simulated and the BB input were stepped by 0.025 V from 0 V to 0.5 V. The result of this simulation is seen in figure 5.24, 5.25 and 5.26. In the 3D plot we see the expected tendency for the circuit. Increasing BB, larger $\Delta\tau$, results in fewer n_e, as seen by the declining slope from left to right. With less than 0.1 V as the BB input all the detectors are activated and the generated τ_i is larger than the TDMC can detect resulting in buffer overflow. In figure 5.25 the effect of changing τ_i is easily seen. For all the different BB inputs increasing τ_i results in larger n_e as expected. It is also possible to see the reduced accuracy when increasing BB by the declin-

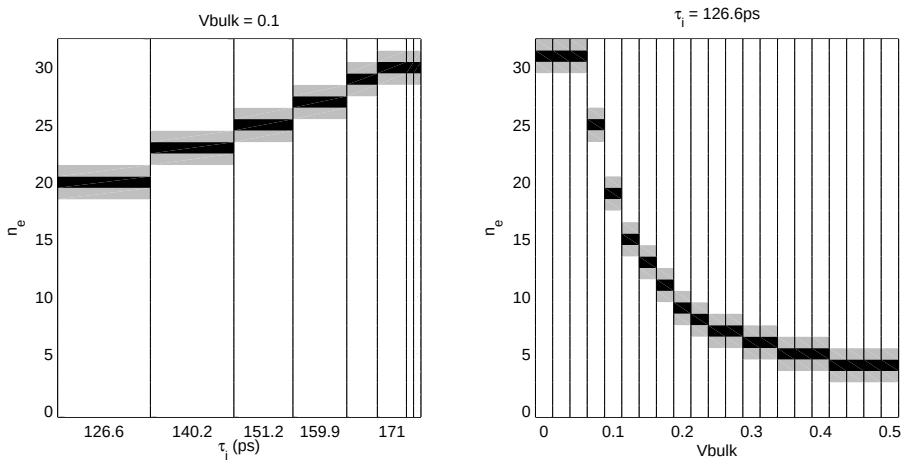

Figure 5.23: Example of a grey-maps for the TDMC. Showing the PDFs for different inputs.

ing slope. In figure 5.26 the n_e behaves as expected from the simulations of the delay elements.

Figure 5.25 show that the circuit behave linearly except for the largest τ_i, clearly seen for Vbulk = 0.15–0.2 V. One effect caused by the Body Biasing is increasing gate capacitance of the transistors. In figure 5.27 simulated input capacitance is plotted as a function of BB. The gate capacitance increase with increasing BB and will continue to increase until BB is approximately 0.225 V before it starts to decline. Both the tapped delay lines are buffered by an inverter as a last stage before entering the line. These inverters charge different loads, resulting in different time delays. This again means that the generated delay from the pulse generator will increase some. Simulations show that this increase is about 30 ps at the maximum BB = 0.225 V. When BB is increased past this point the capacitance is reduced and the extra time delay is also reduced. This is assumed to be the cause of the small dip found for the largest τ_i, at Vbulk = 0.15–0.2 V

The gate capacitance was determined through simulations. Several delay elements was connected serially, and the input was an ideal signal generator charging the first gate through a 100 kΩ resistor. Simulating with different BB inputs and measuring the time to reach 0.632 V on the gate gave the RC delay. Then solving $t = RC$ for C give the simulated gate capacitance. The increasing gate capacitance is caused due to the reduced depletion region beneath the gate. By applying a positive BB on the bulk, the depletion regions will shrink around the source, drain and the channel beneath the gate will be shallower. Most of the gate capacitance lies between the gate

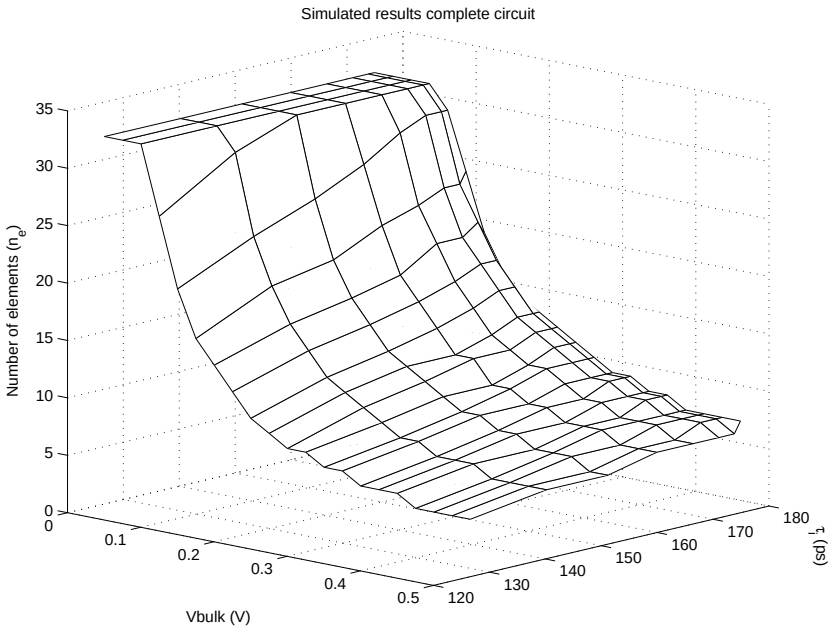

Figure 5.24: 3D view of the simulation results for the TDMC.

through the gate oxide and channel to the bulk. A thinner channel will reduce the distance between the gate and the bulk and hence increase the capacitance. Why this effect declines after approximately 0.225 V is unknown. The BSIM4 models used for BB is already shown to be inaccurate at best, so these results should be further investigated, but are outside the scope of this thesis.

In table 5.7 three BB inputs with outputs are shown. Starting with a BB of 0.4 V ($\Delta\tau = 9.00$ ps) the lack of accuracy is seen in n_e. When the τ_i increases there is almost no change in n_e. The first four τ_i inputs result in almost no change of n_e. All four should result in an increased output by one, but the change only happens twice. When the BB is reduced to 0.3 V a larger change of n_e is seen. The first increase of τ_i should result in an increase of n_e by two compared to the first, which we do not see. This can happen if the first two are close to the edge for the difference $\Delta\tau$ samples. The first can be towards $n_e = 6$ while the second are closer to $n_e = 9$. In measurements this would result in two grey areas in the grey-maps when noise effects the decision in either direction. The overall accuracy with 0.2 V BB seems to be around the simulated $\Delta\tau$.

With a BB of 0.2 V we see indications of even better accuracy. Here the first changes to τ_i seems to result in changes to n_e as expected with $\Delta\tau = 5.42$ ps. Correlation between the $\Delta\tau$ and changes in τ_i are better with smaller changes

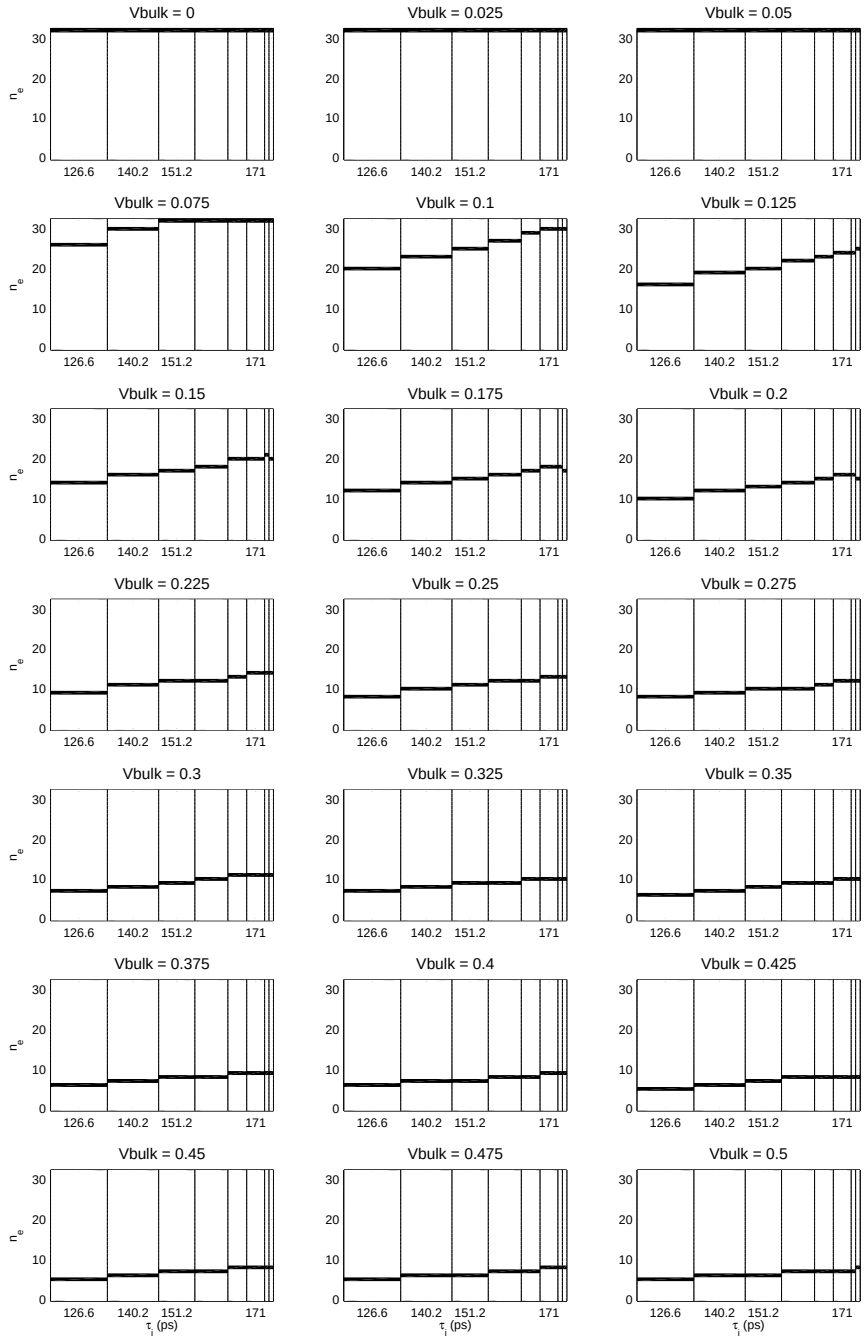

Figure 5.25: Grey-map of simulations as a function of generated delay for the TDMC.

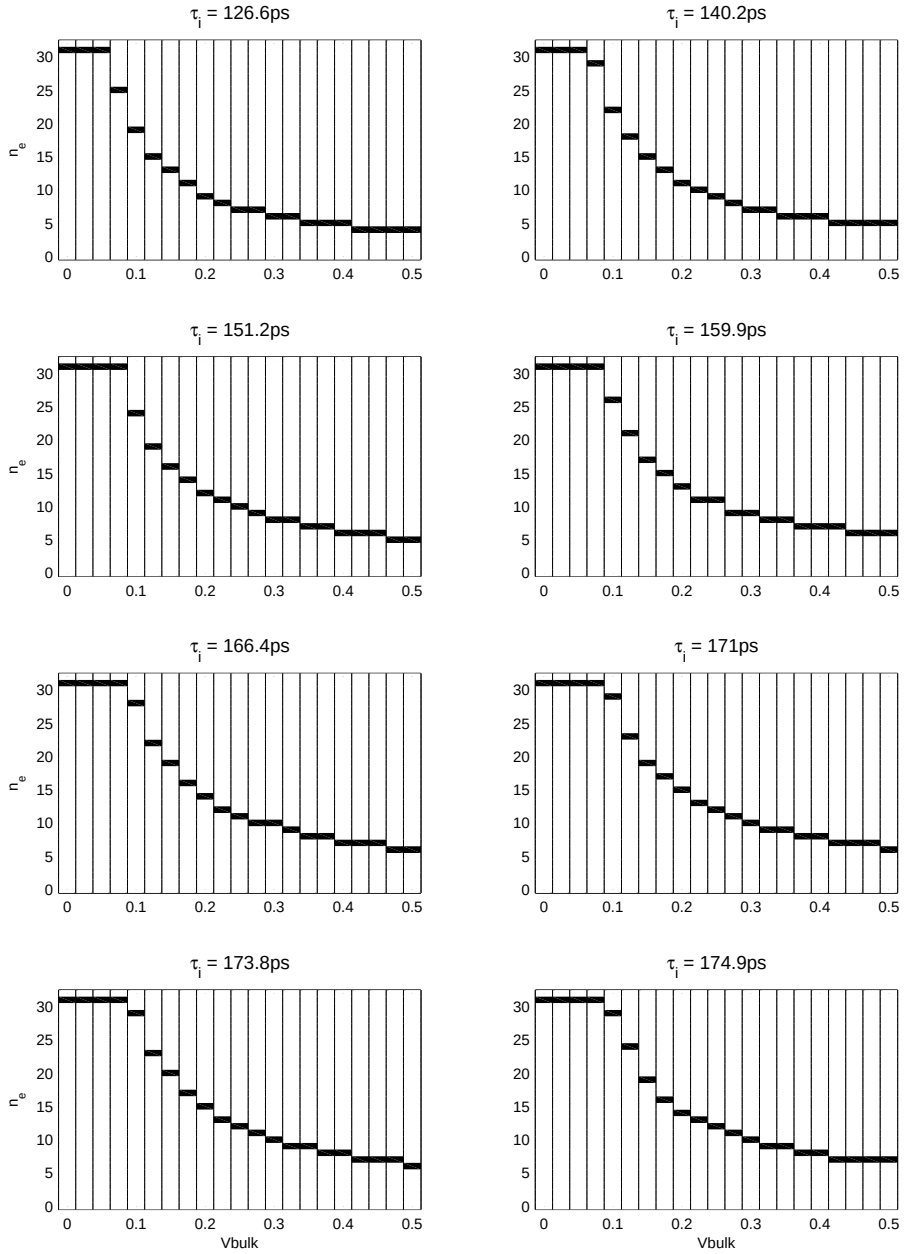

Figure 5.26: Grey-map of simulations as a function of BB for the TDMC.

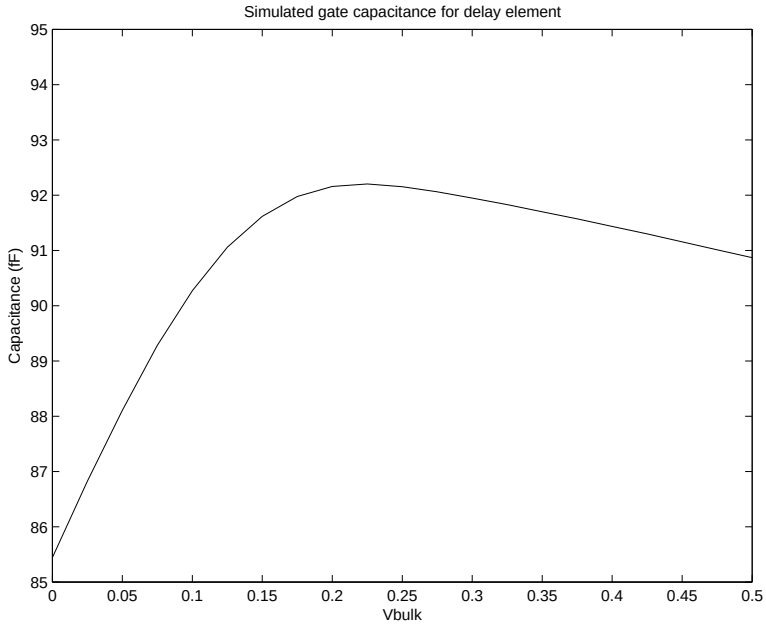

Figure 5.27: The effect of BB on gate capacitance for the delay element used in the TDMC.

to τ_i and indicates accuracy of around 5 ps!

All in all the circuit seems to perform well and should yield reasonable measurements.

	BB	0.4 V	0.3 V	0.2 V
τ_i	$\Delta\tau_i \backslash \Delta\tau$	9.00 ps	7.45 ps	5.42 ps
126.8 ps	–	6	7	10
140.5 ps	13.7 ps	7	8	12
151.5 ps	11 ps	7	9	13
160.5 ps	8.9 ps	8	10	14
166.3 ps	5.9 ps	8	11	15
171.2 ps	4.9 ps	9	11	16

Table 5.7: A few simulation values from the TDMC.

5.4.5 Layout considerations

Significant efforts were used to get the time critical parts to match up regarding wire lengths and parasitic capacitance. Creating the eight buffers and pass gates in the pulse generator as identical as possible meant placing

them in a fix grid and using the same length of wires to connect everything together. The last stage before entering the TDMC for both pulses were identical inverters placed to minimise any parasitics.

In the TDMC tiny inverters are used when tapping the signal from the delay lines. By doing this the added capacitance is kept at a minimum trying to reduce the added delay. This also meant that the signals are buffered one more time before driving the d-flip-flop inputs. The delay elements in the delay line are symmetrical around the detector placed between them. This was done to reduce any mismatch in the circuit created by parasitics.

5.5 TDMC measurements

In this section the measurements for the TDMC are presented.

5.5.1 Measurement setup

A small PCB was created so it was easy to test the circuit manually and from MATLAB using the GPIB interface for the necessary instruments. The measurement setup consists of several voltage sources, an oscilloscope and a signal generator. The voltage sources were controlled from MATLAB together with readout from the oscilloscope making it possible to automate the measurements. By automating the measurements it was possible to do several hundred measurements on each chip so good estimates could be made

The measurements were done with the same inputs as the simulation of the complete circuit. BB was swept from 0 V to 0.5 V for each of the eight generated time differences τ_i. The measurements are conducted on eight chips with 100 measurements or more for each data point.

5.5.2 Measurements

During the initial measurements a problem with the output arose. The period of the output signal were not 160 µs as expected with an input clock of 200 kHz and the 32 bit long shift register. To separate these inconsistent measurements from the others, all measurements with a period other than 160 ± 1 µs are discarded. The result is a very low number of measurements for low values of BB. When a new and better signal generator were used to create the clock input to the shift register a drastic improvement was observed. Lack of clock buffering internally is suspected to be the cause. To determine how reliable the measurements are, figure 5.28 show in percent

how many of the measurements gave a valid output. All the 1100 measurements for each data point is considered, and a BB of 0.2 V or higher will result in very reliable measurements. The plot show the average number of measurements obtained at a data point.

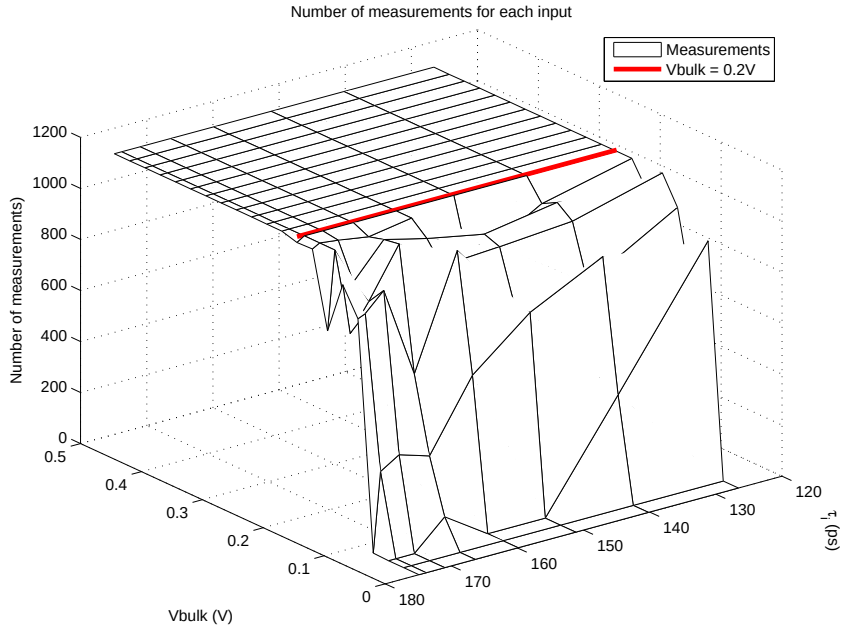

Figure 5.28: 3D view that show the number of valid measurements from the TDMC for all inputs.

Figure 5.29 show the mean measurements at each data point for all the chips. The behaviour is as expected, n_e increases with larger τ_i and smaller $\Delta\tau$. When compared to the simulated results in figure 5.24, the surface of the measured results is smoother. The reason for this is the averaging of the measurements and the effect of noise on the measured n_e. The noise will result in small changes in the pulse ramp and cause the threshold to be crossed at slightly different times. With enough noise and two pulses passing each other between two delays the TDMC will detect different n_e and the average will yield a better estimation than one measurement or the median.

To better see the difference between the simulation and the mean measurement $n_{diff} = n_{e_s} - n_{e_m}$ is plotted in figure 5.30. It clearly show that the simulation deviates more from the measurements at lower BB, and have a slight increase in difference with higher τ_i. but over all the difference is small and stable. The increase in error for lower BB is also seen in figure 5.9. The absolute difference between simulated and measured BB effect is larger

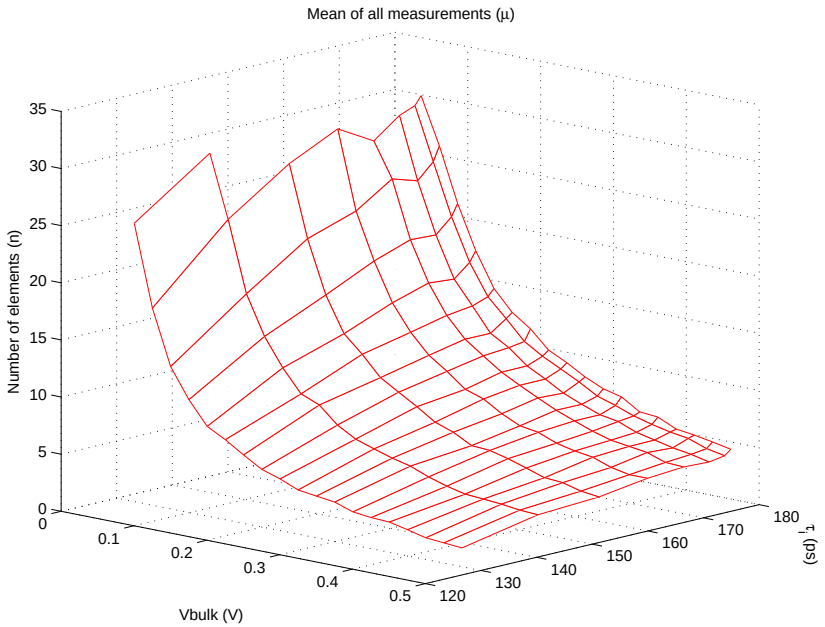

Figure 5.29: 3D view that show the mean measurements from the TDMC.

for lower BB and hence results in larger difference between simulation and measurements.

In figure 5.31 the simulations, mean measurements and n_{diff} are plotted together. The measurements follow the simulations well, but detect slightly smaller n_e. This can be explained by the inaccurate BB modeling of the transistor models shown by the ring oscillator measurements provided earlier. Since the biased delay elements actually behave faster in reality than predicted by simulations, the number of elements before the slower pulse is overtaken is lower in measurements.

Process variations and noise

We will start by looking at the deviations caused by noise in the individual chips. Since several measurements were performed on each chip, the average of all these measurements, and the corresponding standard deviation (σ) from this average from each of the considered chips can be found. The exact causes of this noise is hard to determine, but most of it probably resides from noise in the pulse generator.

In figure 5.32 the resulting standard deviation for all chips has been averaged. It looks fairly random, but it is possible to see the tendency of higher

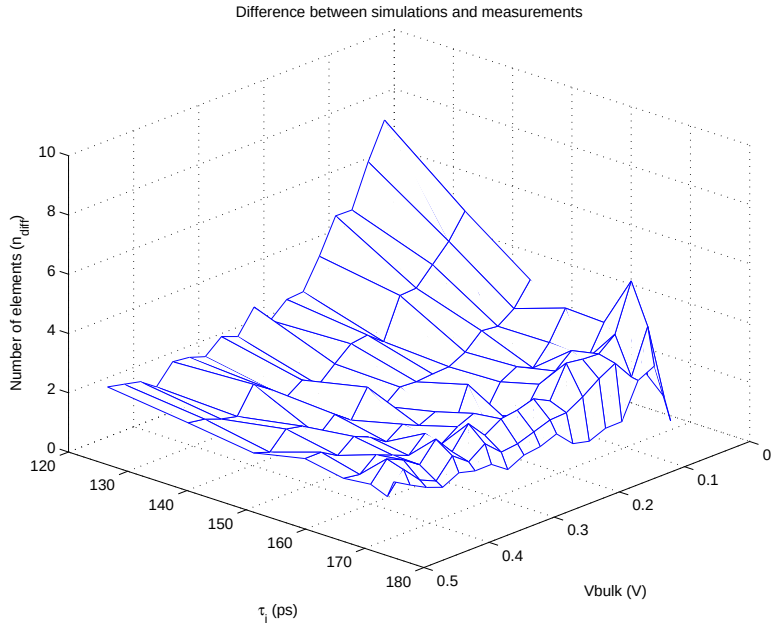

Figure 5.30: 3D view that show the difference between simulations and mean measurements for the TDMC.

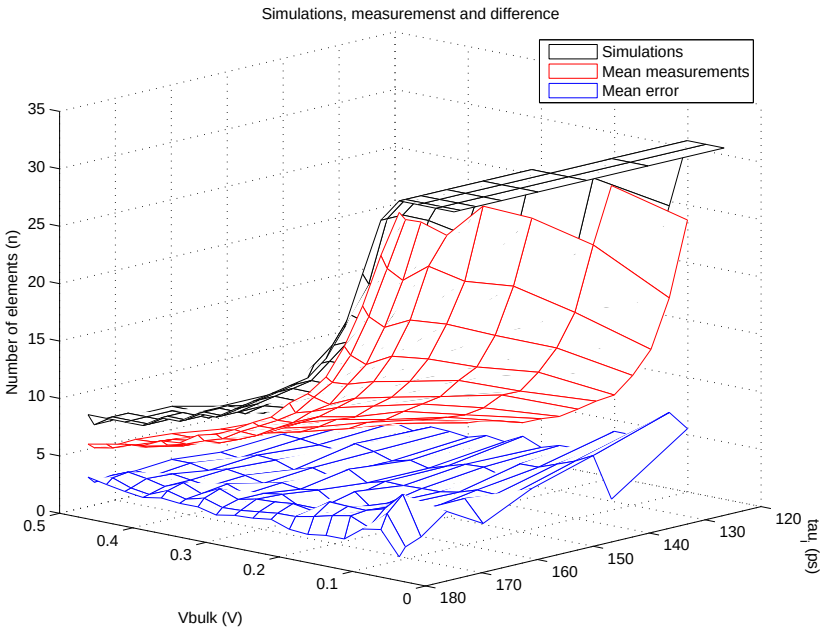

Figure 5.31: 3D view that show the simulations, mean measurements and the mean error for the TDMC.

standard deviation at lower BB and higher τ_i as expected. The increased σ for larger τ_i is caused by the reduced Signal-to-Noise Ratio (SNR). The increase in τ_i is smaller for higher values than for lower values, and thus the noise is more prominent. A $\sigma < 0.5\Delta\tau$ (n_e) indicates a precision of about $1\Delta\tau$ for 95% of occurrences (a confidence interval of 95%).

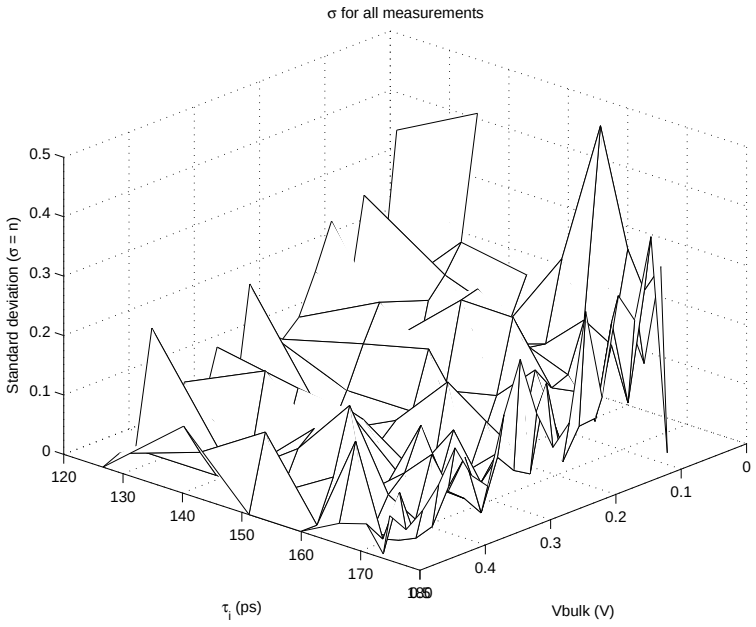

Figure 5.32: 3D view that show the average standard deviation of n_e from the TDMC, showing the effect of noise.

In figure 5.33, all the measurements are considered together. This plot therefore includes process variations in addition to just noise between individual measurements. The deviation is much higher here, meaning process variations contribute considerably compared to the noise in the individual chips. The deviation is actually almost four times $\Delta\tau$. The τ_i of 166.4 ps stands out with higher σ than its neighbors. This is caused by the fact that one chip deviates from the rest at this τ_i, as explained in following section.

Grey-maps

Figures 5.34 and 5.36 are grey-maps of all the 1100 measurements, while figure 5.35 and 5.37 are the measurements of a single chip. Comparing the single chip with the sum of all chips it is clear that the circuit is susceptible to process variations. The single chip show unambiguous n_e measurements

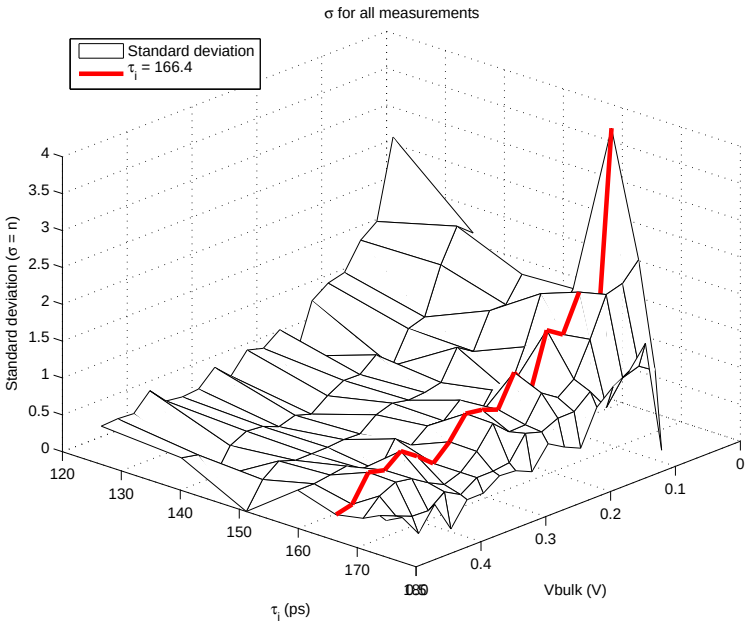

Figure 5.33: 3D view that show the standard deviation of n_e for all inputs from the TDMC, including process variations.

while the sum gives much more diverse results. Looking at the grey-maps with regard to BB the same tendency is seen.

For τ_i at 166.4 ps and 171 ps it is clear that one chip deviates noticeably from the mean at each level. In figure 5.38 all chips are plottet together, and the two chips can be identified. These differences originate from the pulse generator and most likely the threshold of the buffer after the RC delay. This is the reason for the noticeable higher standard deviation at $\tau_i = 166.4$ ps for all the measurements, figure 5.33.

The average results for a single chip are found in table 5.8. With a BB of 0.2 V it starts out with a jump of 3 which fits well with $\Delta\tau$ and the change in τ_i. The next changes in τ_i do not result in the expected change in n_e. The jump from $n_e = 11$ to $n_e = 13$ when τ_i changes with 5.9 ps also seems strange with that difference in delay. Increasing the BB to 0.3 V or 0.4 V does not change the lack of correlation between $\Delta\tau$ and n_e as a function of simulated τ_i.

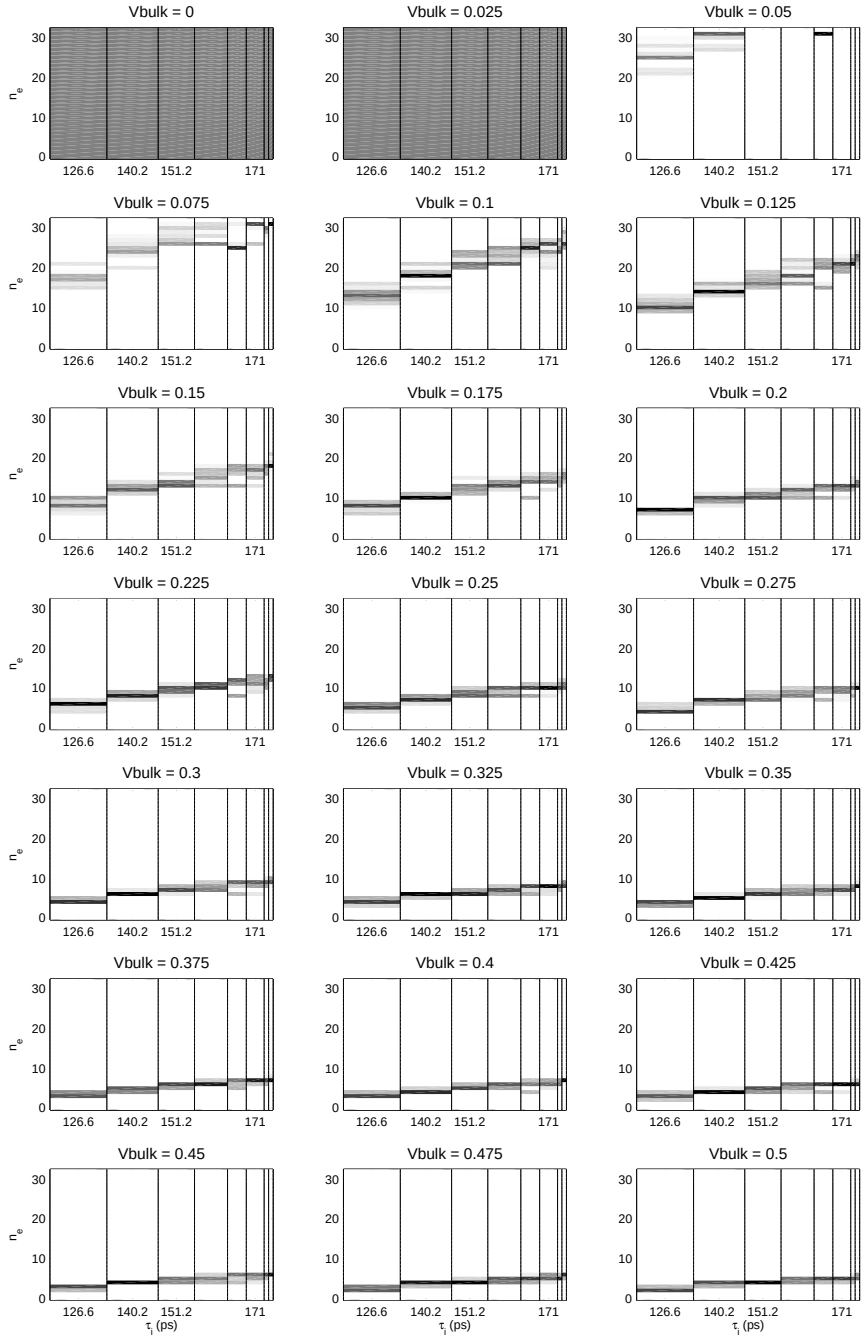

Figure 5.34: Grey-map of all the measurements from the TDMC as a function of generated delay.

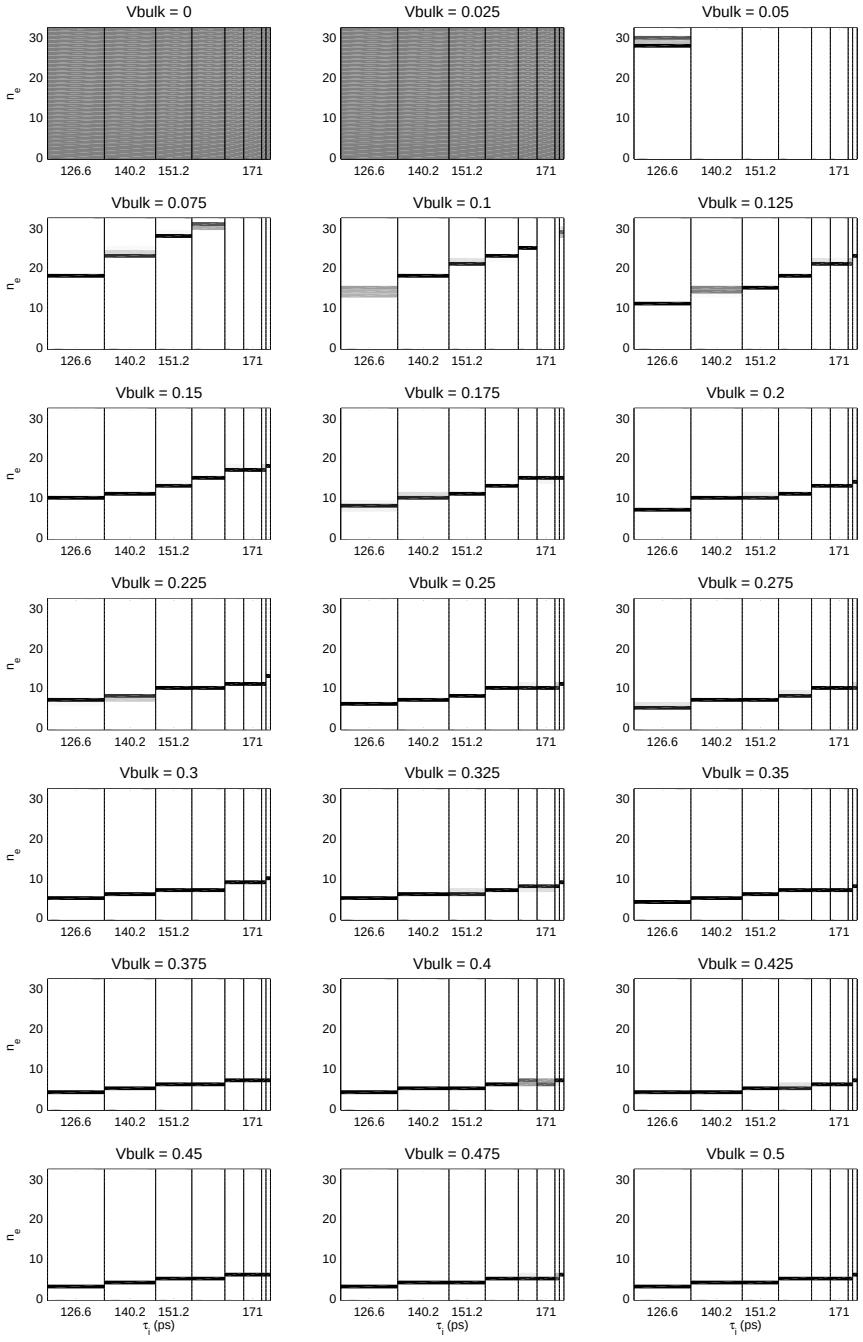

Figure 5.35: Grey-map of a single chip as a function of generated delay.

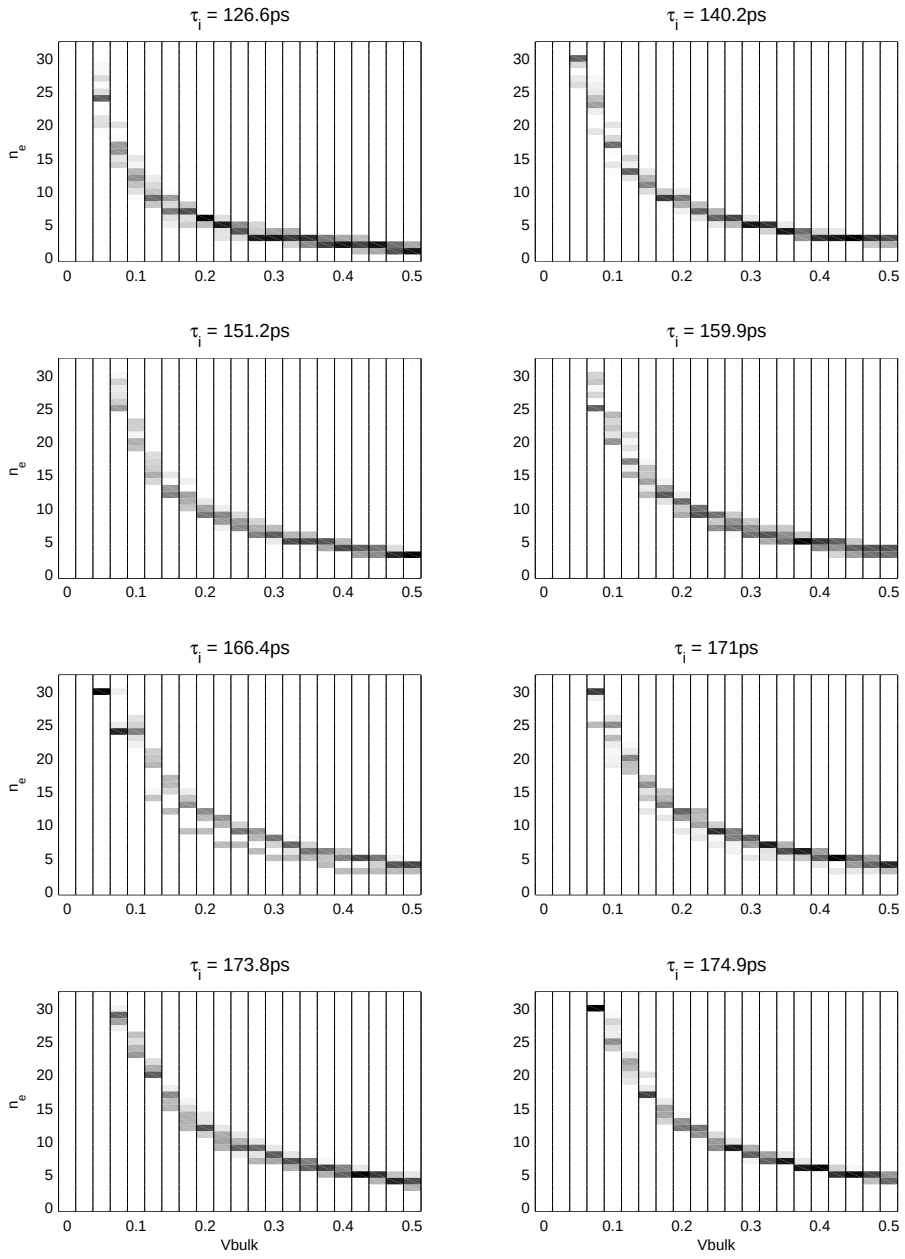

Figure 5.36: Grey-map of all the measurements from the TDMC as a function of BB.

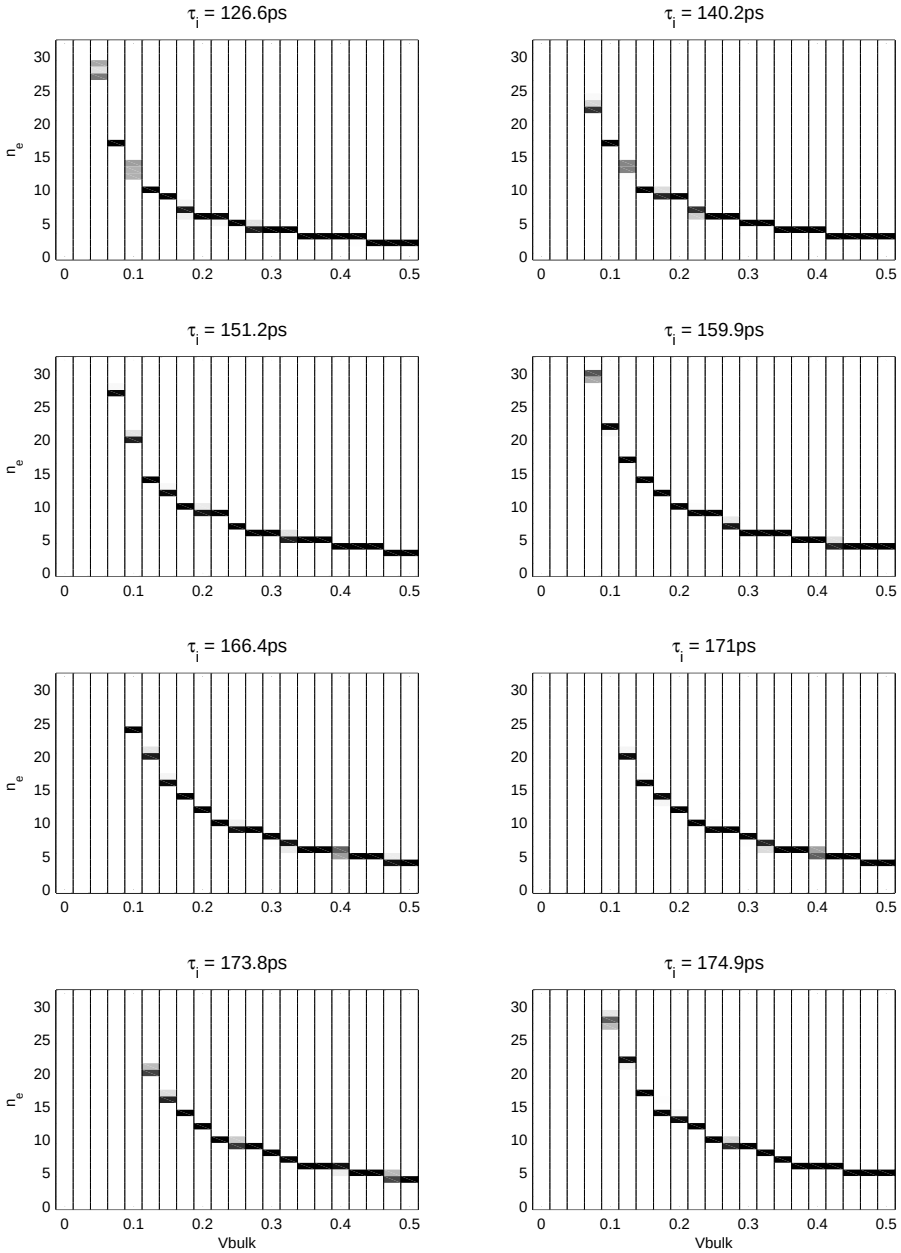

Figure 5.37: Grey-map of a single chip as a function of BB.

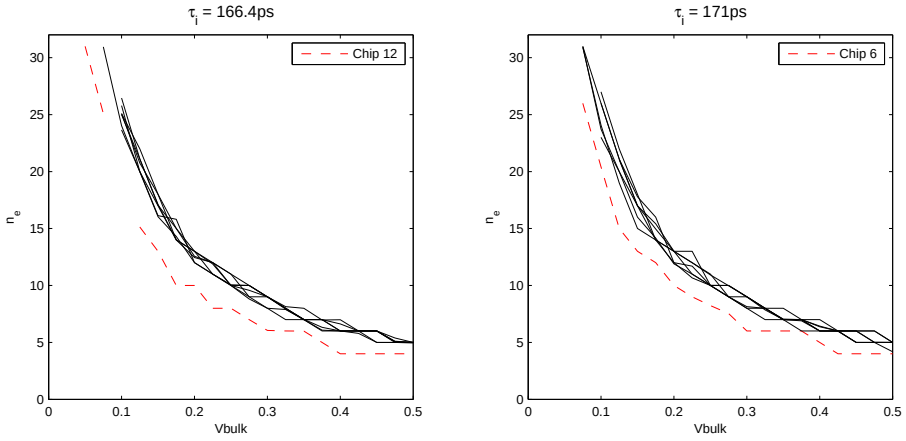

Figure 5.38: Mean measurements for all chips as a function of BB, with two standing out from the mean.

	BB	0.4 V	0.3 V	0.2 V
τ_i	$\Delta\tau_i \backslash \Delta\tau$	9.00 ps	7.45 ps	5.42 ps
126.8 ps	–	4	5	7
140.5 ps	13.7 ps	5	6	10
151.5 ps	11 ps	5	7	10
160.5 ps	8.9 ps	6	7	11
166.3 ps	5.9 ps	6.6	9	13
171.2 ps	4.9 ps	6.4	9	13

Table 5.8: Mean of measurements from the TDMC for an single chip.

5.5.3 Summary

The idea of using body biasing to obtain relative time difference measurements is interesting because it allows post production tuning of the circuit. This is important, first of all because it means the range and accuracy can be changed in-field, but also because it can be used to compensate for process variations.

The measurements were promising and provided important observations. The pulse generator has some issues though, so the actual performance of the circuit in terms of accuracy is hard to determine. Measurements indicate that an accuracy of 5–10 ps can be achieved. The reader should however keep in mind that the chip was produced to provide a proof of concept, in which we have succeeded.

When we look at the grey-maps of a single chip, we see a clear tendency in the behaviour of the circuit. For low bias voltages, the circuit fails. This is hardly surprising, since is only means $\Delta\tau$ is too small to be detectable within the range of the implemented circuit. As the bias voltage is increased, the results become more interesting. In figure 5.35 we see a close to linear tendency in n_e as a function of τ_i for bias voltages higher than approximately 100 mV.

As with the ring oscillator, the sample count is too low, from a statistical point of view, to base a conclusion upon. The measurements does however indicate that the circuit is vulnerable to process variations. When comparing the standard deviation between individual measurements on a single chip to the standard deviation of the total collection of samples, it is obvious that variations between the chips is prominent. A deviation of up to 4 times $\Delta\tau$ is significant, and is an issue that needs attention in a future implementation.

Obtaining precision and accuracy in the order of a few millimetres using RF signals is hard to accomplish using traditional techniques, since it means measuring time differences in the order of a few picoseconds. The circuit described here solves this without the need for high precision clocks, and is implementable in all triple well CMOS processes. Although the implemented circuit has some issues, most of these resides in the pulse generator and not in the measurement circuitry. The measurements therefore provide a solid proof of concept, but before the exact precision and accuracy achievable with this scheme can be found, there are some practical issues that needs to be solved.

One of the challenges in future realisations is how to avoid the added delay when entering the delay line. Simulations estimated this extra delay to about 30 ps, which would add 9 mm when used in ranging. One solution

to this problem can be to tune both the delay lines to reduce the difference in gate capacitance. Another is to use really large buffers as the last gate before the delay line. The idea of biasing both the delay lines is an interesting thought. This will reduce the effect of mismatch on both delay lines resulting in higher precision. The extra delay due to higher gate capacitance at the first gate is reduced and it will also be possible to create a very small $\Delta\tau$. Biasing both lines should be tried in any future work. The effect of reversed (negative) body bias should also be investigated.

Chapter 6

Performance

A computationally light or heavy algorithm might result in good or bad position estimates. It is time to determine which approach is the best.

6.1 Chapter overview

This chapter start with a discussion about how the quality of a positioning algorithm can be evaluated. Then the simulation setup and approach is presented. At the end the suggested positioning algorithm is evaluated with different estimation techniques during the two phases.

6.2 Evaluation of quality

To evaluate a positioning algorithm some sort of quality measurement is needed. One approach is to use the Root Mean Square (RMS) error in the distances from the estimated position to the references compared to the measured ranges. This can be used as a sanity check on the estimated position in the node. If this error is larger than the communication range, the estimated position can be discarded as erroneous. The equation for the range error approach is:

$$\sigma_{\mathbf{D}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\sqrt{(x_n - x_i)^2 + (y_n - y_i)^2} - D_{ni}\right)^2} \tag{6.1}$$

To determine the quality of the mapping it is common to use the RMS position error. This is to find the average error of the position estimates compared to the real locations. This is the quality measurement used when

evaluating the algorithm with different positioning techniques. To estimate the RMS position error the following equation is solved:

$$\sigma_{\mathbf{p}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left|\mathbf{x}_{real_i} - \mathbf{x}_{est_i}\right|^2} \tag{6.2}$$

$$\tag{6.3}$$

6.3 Useable position information

Since the algorithm does not rely on any form of anchor nodes, it is not straight forward to evaluate it. The position estimates need to be mapped to the same reference system as the original coordinates. There are two approaches to determine the quality of the position estimates and mapping, one is to determine a theoretical estimated quality. The other is to look at it in a real world scenario where the information in the WSN is to be mapped to real world locations.

The theoretical estimated quality is valid when a handheld node is used moving around in the WSN. There is no need to map the estimated positions to real world locations. The position of nodes is found relative to the hand held device. In application where the position estimates are linked to real world locations errors emerge. The application determine if mapping to real world locations is needed.

6.3.1 Relative localisation

Finding the position error in a relative setting means finding the infimum of the RMS error. This is a theoretical best fit of the mapping to real world locations, reducing $\sigma_{\mathbf{p}}$ to a minimum. This is a very good quality measurement of the positioning algorithm, but it is hard to find. The problem can be described as:

$$\sigma_{\mathbf{p}min} = \inf\left(\sigma_{\mathbf{p}}\right)$$

To find this minimum the four freedoms in 2D have to be tweaked. Considering positions in a plane the error come from either the movement of the entire plane in the x- and y-directions, mirroring, rotation around a point or scaling. An error in x- and y-direction is easily corrected and adds the same error to all the mapped positions. If the system is mirrored, it is not

hard to correct by mirroring around one axis and then move the system back in place. The error originating from wrong scaling increases linearly as the distance to the origin of the coordinate system increases. The effect of the rotation error also increases as the distance to origin increase. To find the infimum of $\sigma_{\mathbf{p}}$ a reduction algorithm has to be applied.

6.3.2 Real world mapping

Looking at the scenario where the estimated positions are mapped to real world locations there are degrees of freedom. In a real world scenario some of the nodes are used as anchor nodes when mapping the positions. Three nodes are needed in the 2-dimensional scenario to uniquely map the location estimates to the real world. Depending on which nodes used as anchors the mapping can result in both good and bad approximations to the real world locations.

The error created by the mapping of the estimated positions to real world locations come from the four freedoms explained in previous section. The most severe error from mapping is the rotation error, due to how it affects the positions. If nodes close to each other are used as references, a small estimated location error can result in very large mapped real world location error for nodes far away. In figure 6.1 the effect of using nodes close together, lower left corner, as anchors when mapping to the real world locations is seen. The error when using nodes far apart, lower left, upper left and upper right corners as anchors is significant smaller. The diamonds ($\diamond$) represent the true position, the x sign represent references close together, while the + represents references far apart.

This shows that choosing the reference nodes for the mapping is of great importance. It is desired to used nodes far apart as references when mapping the positions to the real world.

6.3.3 Mapping used

In the evaluation of the positioning algorithm the real world mapping approach is used before the RMS positioning error is found. The three reference nodes are chosen to be in the lower left, upper right and upper left corners to reduce the mapping error from rotation. To perform the mapping the algorithm presented in section 4.8 is applied.

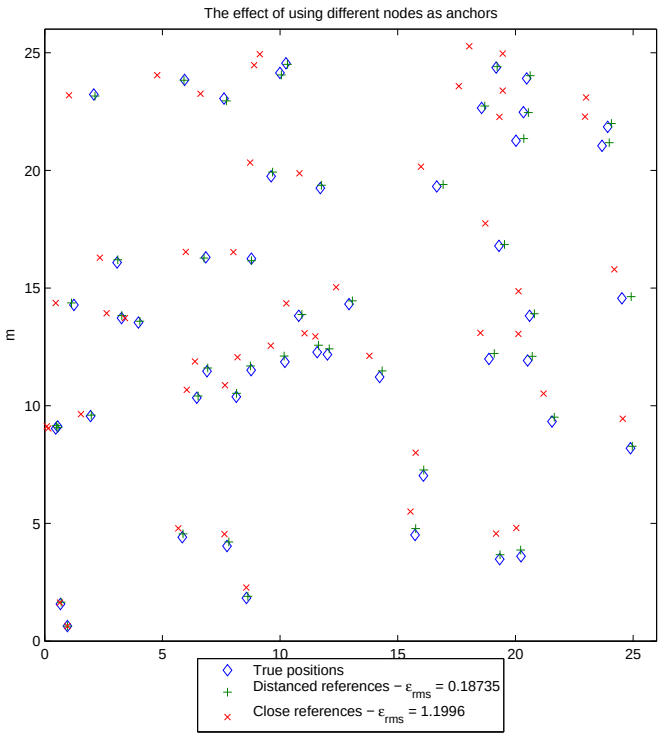

Figure 6.1: Error as a function of the range between references.

6.4 Simulation environment

The simulations are conducted in MATLAB. OMNeT++ has been used by Langendoen et al. to perform a quantitative comparison in [Lang 03]. This is a general discrete event simulation environment and could have been used for these simulations. MATLAB was easily available and is well known to the author and was chosen to be the platform for the simulations. A real WSN implementation of the algorithm is event driven, which is not easy to perform in MATLAB. The algorithm is implemented as MATLAB scripts and the events are reduced to loop structures.

Since the events are reduced to loops it is hard to simulated the propagation of positioning awareness in the network. The position algorithm is divided into two main phases. The first phase estimates a position for all well connected nodes, while the second phase refines this estimate. The first phase has to consider the propagation of position awareness throughout the network, while the second phase does not in these simulations.

To ease the scripting no communication protocol is implemented. This means all communication and information is available for a node during positioning estimation. During one iteration of the refinement phase the same erroneous range estimates are used. Only local broadcast is expected, only neighbour nodes within the communication range receive the position estimate from a node. The communication range does not consider any obstacles or non ideal antennas and is modulated as a circle around the node.

To make the simulations repeatable random seeds are fed to the random generators at the start. The WSN is created in a 100 by 100 unit square. The unit is generic and can represent 1 meter. 225 nodes are randomly placed with a uniform distribution throughout the area. One of these 225 nodes is randomly selected to be the starting node for the ABC algorithm.

When estimating the position, erroneous range measurements are used. A small normally distributed error with $\sigma = 0.1$ is used. This would represent a precision of 10 cm if one unit equals 1 m. In chapter 5 a precision of 10 ps (3 mm) is indicated, a precision of 10 cm is not unrealistic.

The input for a simulation is the communication range and the number of refinement iterations to perform. Each simulation returns the average number of connections, the coverage in the network and the average position error. Any nodes without a position estimate is excluded from the position error. To reduce the randomness in the simulation each input is repeated several times. Due to simulation time and problems with MATLAB, simulations were only conducted with 30 iterations for each input.

6.4.1 Implemented algorithms

Since the algorithm is reduced from event driven to loop structures slight changes are made. First the randomly selected starting node and two neighbours create the local reference system. For each node a list of neighbours with known position is maintained. By using a infinite while loop this list is used to find nodes without position estimates and three or more neighbours with known position. The while loop only breaks when all nodes have a position or no nodes found their position during the last iteration.

Algorithm 6.1 Implemented first position estimate algorithm

```
randomly pick a starting node
apply the ABC algorithm on the starting node with neighbours
while nodeFoundPosition and nodesWithoutPos do
  nodeFoundPosition = false
  for k = 1 to numberOfNodesWithOutPos do
    if numberOfNeighboursWithPos ≥ 3 then
      pos(k) = estimatePos
      if sanePos then
        update the list of known nodes for all neighbours
        nodeFoundPosition = true
      end if
    end if
  end for
end while
```

The next step in the algorithm is the refinement phase. Two for loops are used, the first perform the required number of refinements for all nodes. The next cycles through all nodes with known position and refines their position.

Algorithm 6.2 Implemented refinement algorithm

```
for k = 1 to numberOPfRefinements do
  ranges = errRanges(orgRanges)
  for l = 1 to numberOfNodesWithPos do
    pos(l) = refinePos
  end for
end for
```

The algorithm is simulated using two different approaches for each phase. For the first phase of estimating a start position multilateration and a combination technique called *Find Position* (FP) is used. The min-max technique is not used during phase one due to lack of performance. The multilateration approach uses the three neighbouring nodes furthest away as refer-

ence nodes, trying to reduce the relative range error. Range error in this scenario with dedicated ranging system is unrelated to the distance, larger range give smaller relative error. The nodes furthest away are used to minimise the effect of the ranging error. The FP technique is a combination of multilateration and LSE. During MATLAB simulations the LSE algorithm was a source for different problems. Problems involved singular matrices and other linear algebra problems. To solve this multilateration was used to estimate positions where LSE failed. During the refinement phase the FP and min-max techniques are used.

6.5 Simulation results

The connectivity in the network is important to the quality of the position estimates and the coverage of position estimates. The randomness of the generated networks mean some nodes are scattered around with few neighbours. To estimate a position at least three neighbours need a position estimate and range measurements to the node with unknown position. By altering the communication range for the nodes during simulation the connectivity is changed. Larger communication range means more nodes are directly connected. In figure 6.2 the effect of increasing communication range is seen for the average number of neighbours and the coverage. With a communication range of 15 the average connectivity was 13.9. This gives a coverage of almost 100% for the FP technique while multilateration only have a coverage of about 90%.

Numerical methods sometime yield large errors so all position estimates went through a sanity check. The RMS range error was found using equation 6.1. If this error was larger than the communication range, the position estimate was discarded. By doing this multilateration had trouble estimating a position to some nodes resulting in lower coverage.

In figure 6.3 a 3D plot of the position error is shown. The z-axis is the RMS position error, as presented in equation 6.2. The combination of phase one and two is seen in the legend. For short communication range, the position error increases as long as the connectivity increase. Low connectivity give low coverage and smaller networks yield smaller position error. After the coverage reaches maximum, the position error decrease as the connectivity continues to increase. Higher connectivity yields better position estimates. The effect of the refinement period is not that easily seen in this plot.

The first phase is to estimate a first position for all nodes in the network. Simulations shown that the min-max technique was unsuited during this phase. The average position error was too large, as seen in figure 6.4. Multilateration also seems to have some problem achieving good position esti-

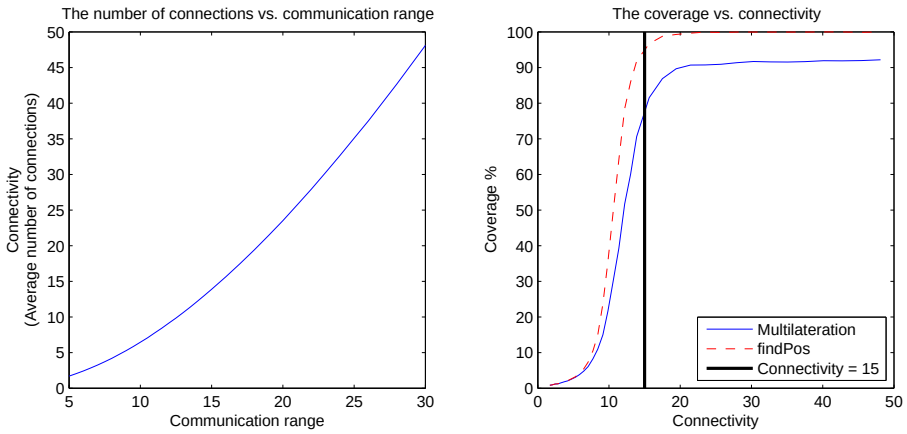

Figure 6.2: The effect of communication range on the connectivity and coverage in the network.

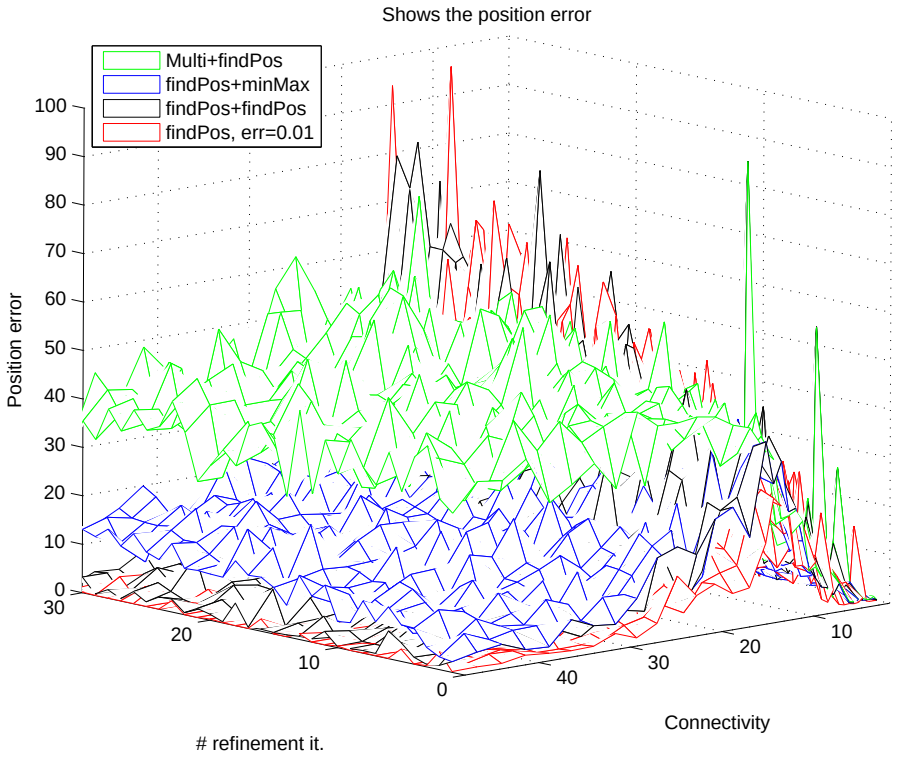

Figure 6.3: 3D plot of the position error for all inputs to the position algorithm.

mates. The FP technique on the other hand yields good position estimates when the connectivity is high. A simulation of the FP technique with an error of 0.01 is also seen. The increase precision in the ranging reduce the position error as expected, and the effect is noticeable. The coarse shape of the plots come from the limited number of simulations performed.

In figure 6.5 the effect of the refinement phase is enhanced. Only minor improvement is observed for the multilateration and FP combination and the double FP combination. The min-max technique increase the position error during refinement phase. These simulation show that the min-max technique is unsuited as a refinement method.

6.6 Summary

In this chapter an evaluation of a position algorithm using several position estimation techniques for WSNs with no anchor nodes have been conducted. All position estimates are relative to the network itself and generated by the network. This reduce the need for infrastructure and information prior to deployment.

Three combinations of phase one and two have been tested. It is easy to see that the connectivity is by far the most important parameter in this type of WSNs. By increasing the connectivity, either by adding more nodes to the area or increased communication range the accuracy of the position estimates increase. Simulations also show that the effect of the refinement phase is very limited when no anchor nodes are present. In a network with anchor nodes the position estimates are adjusted relative to the known positions. In this scenario the refinement phase lack these positions and the effect diminish.

The *Find Position* technique using LSE as the main method and relying on multilateration when LSE fails is the most accurate technique. This is a computational heavy technique, but yields good position estimates even in WSNs with no anchor nodes. The simpler min-max technique is found unsuited in this scenario, with too large position error.

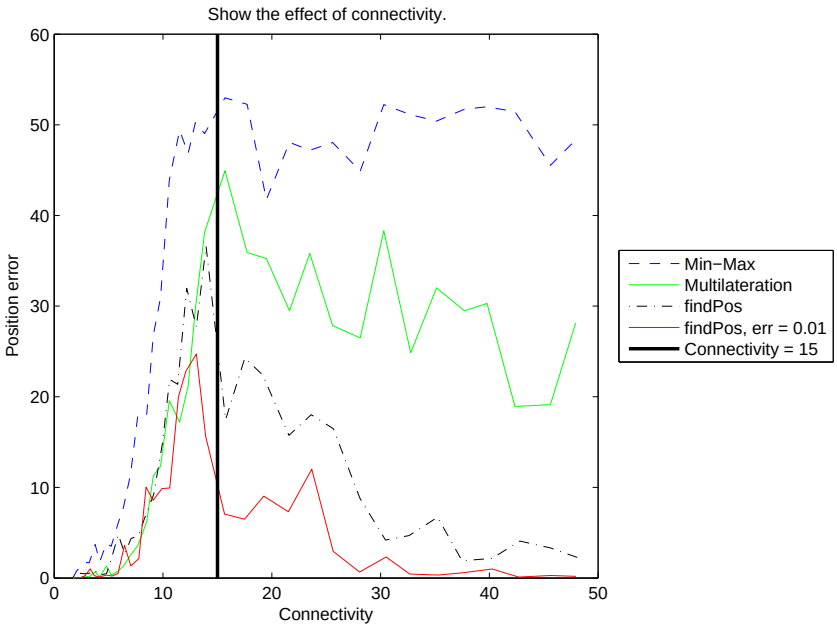

Figure 6.4: The effect of the connectivity on the position error.

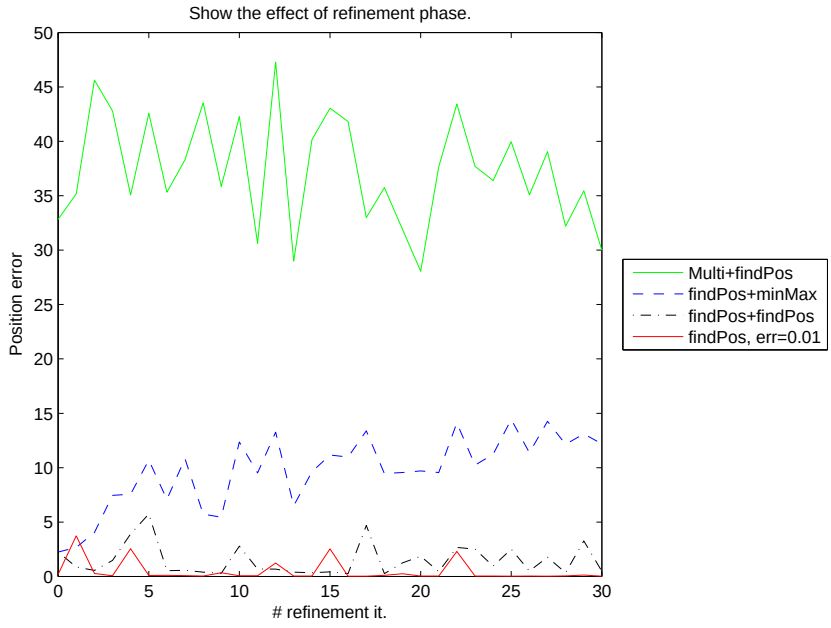

Figure 6.5: The effect of the refinement period on the position error.

Chapter 7

Conclusion

In this thesis a positioning algorithm for WSNs without anchor nodes is presented. By removing the anchor nodes the network is self configuring and easy to deploy. The main problem in these networks is the lack of definite localisation information. The network creates and maintains a reference system and the nodes are positioned in this reference system. This problem has been investigated by simulations of random networks in a square area. The simulations were performed with variations in the communication range (connectivity) and the number of refinement iterations.

The second part of the thesis is dedicated to the development and implementation of a TDMC. The circuit was implemented to test parallel tapped delay lines as accurate and precise time measuring devices. By using identical delay elements with body biasing the circuit is tuneable after production and during operation. In operations the tuning can be used to change the accuracy and observed distances of a ranging circuit.

7.1 Results

The TDMC worked as expected and seems to be very precise. The grey-map of one chip gives clear definite results, with very low variation. A precision of about 1 $\Delta\tau$ is achieved. Due to the lack of measurements of the pulse generator the accuracy is not all that clear. Combining the measurements with the simulations indicates an accuracy of 5-10 ps, equal to 1.5–3 mm. These results implies that high precision ranging is possible.

Using high precision ranging the performance of the suggested positioning algorithm were tested with simulations. Different computation techniques were used and characterised. Simulation showed that it is possible to achieve good position estimates in WSNs without anchor nodes, if the

connectivity is high enough. The FP technique was the only technique that gave good results, while min-max and multilateration clearly struggled, even with high connectivity.

Connectivity is by far the most important parameter in this scenario, and a connectivity of about 30 give good position estimates. In an real WSN with average communication range of 10 m this mean approximately one node for every 10 m^2 if spread evenly. With such high connectivity the LSE problem is quite large, and solving this problem cost energy and time. Even the arithmetic operations before solving the LSE problem sums up to substantially amount of operations: 58 additions, 145 subtractions, 90 multiplications and 58 bit shifts. There is a trade off between the power consumption and how many reference nodes used in the positioning, and consequently the precision in the final position estimate.

The refinement phase has almost no effect on position error in WSNs without anchor nodes. This means that the algorithm should consider to only refine the phase one or two times after the initial position estimate to conserve energy. It also means that refinement can not be used as a substitute for the connectivity, which would have reduced the complexity of the LSE problem.

Simulations also showed that the simple mix-max technique is unsuited both as a first coarse estimation and during refinement phase in this scenario. The implemented FP technique is the only option to achieve reasonable position estimates for the presented scenario.

7.2 Future work

Regarding the positioning algorithm future work should include event driven simulations to verify the performance. Letting the network expand from the first node and estimate the position when enough information is available. Sweeping the range error during simulations to see the effect is also recommended to see the effect of higher precision in the ranging.

Since the circuit needs to handle large LSE problems further investigation should be done to reduce the complexity of the calculations and reduce the power consumption. Implementing the positioning technique in dedicated hardware is also an option to reduce the power consumption. The numerical instabilities need attention. If the positioning technique is vulnerable to certain inputs these should be found. Large partial results in the computation should also be avoided to reduce problems with accuracy in floating point numbers.

WSNs without anchor nodes have interesting possibilities especially regarding the lack of infrastructure and complete ad-hoc structure. Positioning nodes in these networks is challenging, but the suggested algorithm give promising results and is worth a closer look in future work.

The next step regarding the TDMC is implementation in an application specific circuit. This can range from VCOs to ranging systems like the Active Echo scheme. A few problems should be addressed before implementing it, like the extra delay when entering the delay chain and the detector. In a ranging system the extra delay can be compensated for by using the multilateration or LSE technique, as explained in section 4.7.3. A new detector circuit is also needed for a ranging system with integration. Apart from these issues the presented TDMC is certainly worth further investigation and research, and is a promising approach for high accuracy and precise ranging circuits.

Appendix A

Layout

This section contain the layout diagrams of the implemented circuits.

A.1 Ring Oscillator

Figure A.1 show the layout of the inverter used in the ring oscillator. Notice the N-well and the deep N-well surrounding the NMOS transistor. The ring oscillator is shown in figure A.2. Because the oscillator consist of 201 equal inverters, the mid section is left out. Not shown in the plot is the long feedback wire which runs on the right hand side of the structure. On top of the oscillator the output buffer can be seen.

A.2 Time Difference Measuring Circuit

Figure A.3 show the delay element used in the TDMC. Notice the deep N-well around the NMOS at the bottom and the large distance requirement between the two transistors due to this. The d-flip-flop used as a detector is seen in figure A.4. The somewhat awkward shape is from the reshaping to make it fit between two delay elements. In figure A.5 two delay elements, two inverters and a d-flip-flop are packed together to create a cell with both delay lines and the detector. This made it easy to adjust the number of elements in the tapped delay line. The RC-delays and the output buffer of the pulse generator is seen in figure A.6. The resistor runs as a snake and the different taps are visible.

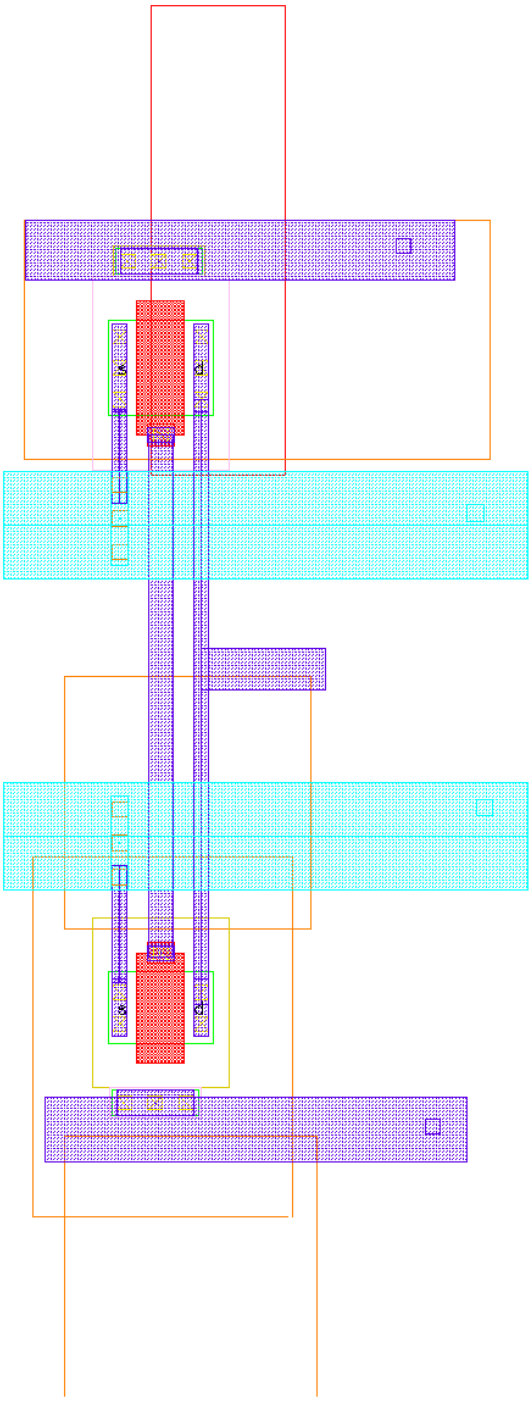

Figure A.1: Layout of inverter used in the ring oscillator

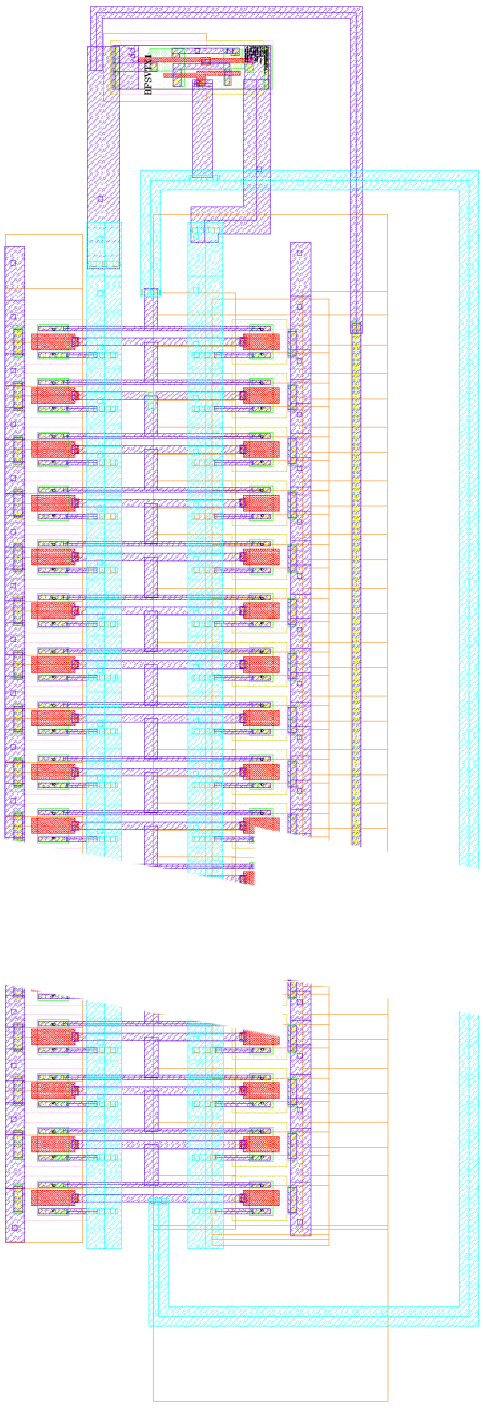

Figure A.2: Complete layout of ringoscillator. To make the picture more comprehensable, the midsection of the structure has been left out since the entire ring oscillator consist of 201 equal inverters.

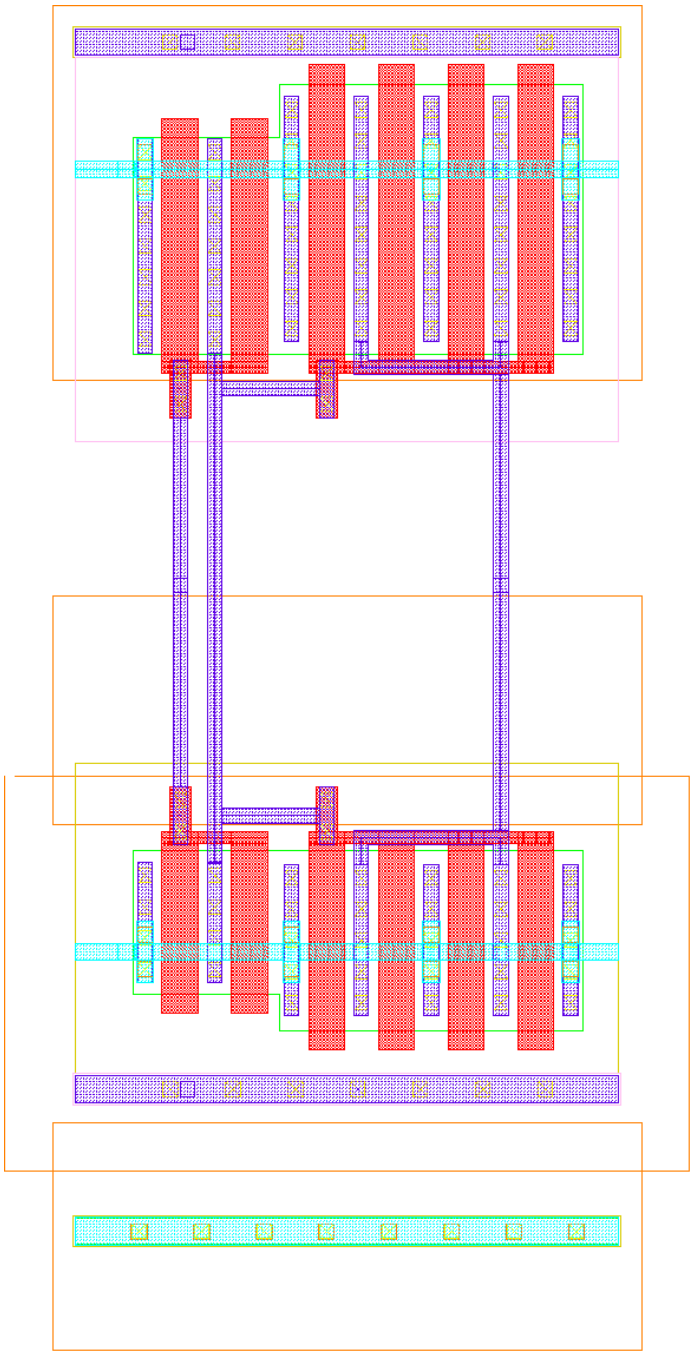

Figure A.3: Layout of the delay element implemented in the TDMC. The input is the left blue wire, and the right is the output. The PMOS-transistor is located at the top, while the NMOS-transistor in a p-well is located below.

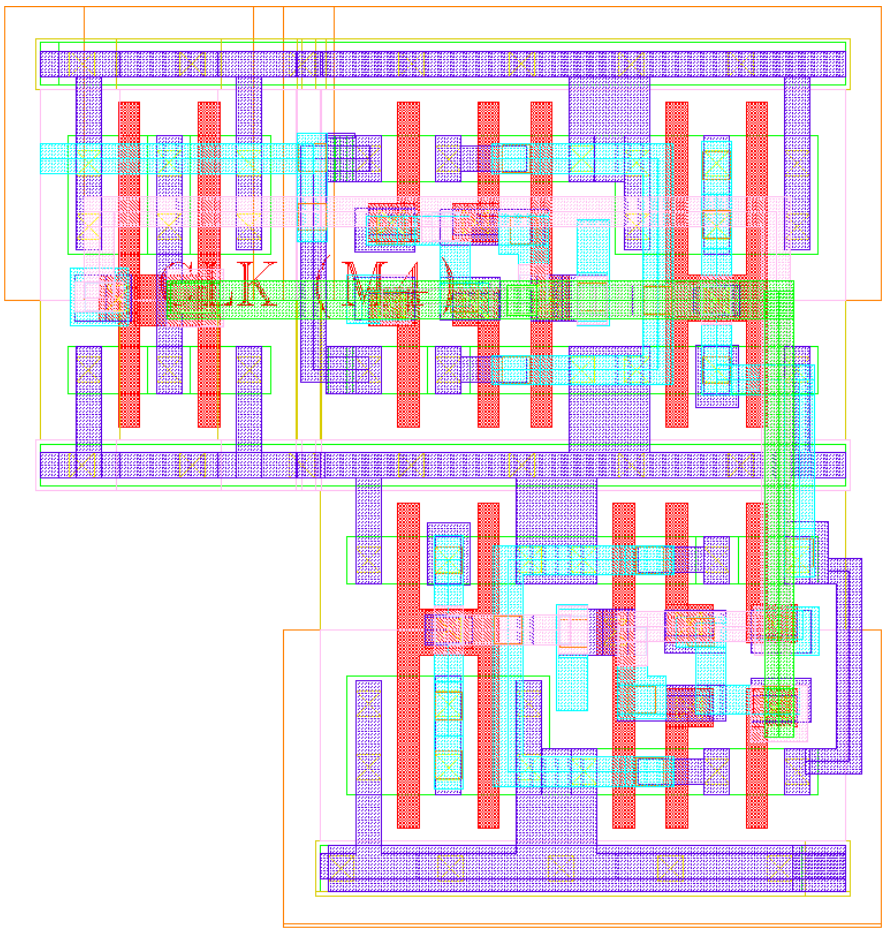

Figure A.4: The d-flip-flop used as a detector. The inputs are seen as **Clk** and **M1**, and the output is cyan wire to the left of the lower half. The shape is chosen to fit between two delay elements.

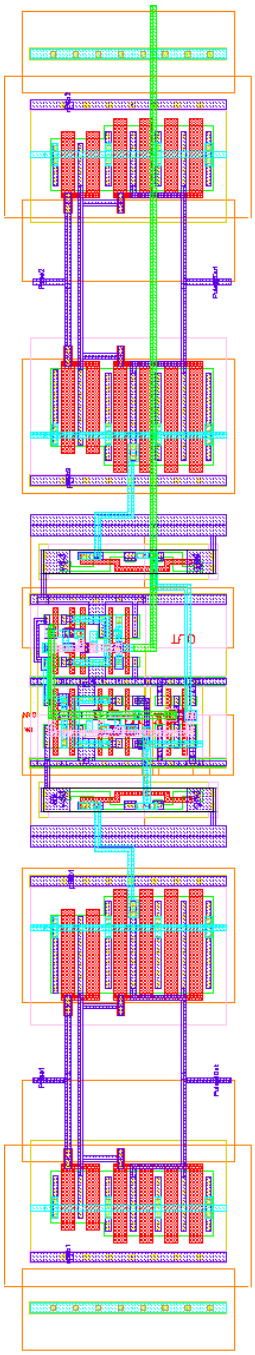

Figure A.5: Two delay elements, two inverters and a d-flip-flop connected as a cell. These cells were stacked to create the delay line. The delay line inputs are on the left side, and the d-flip-flop output is the green wire running to the top.

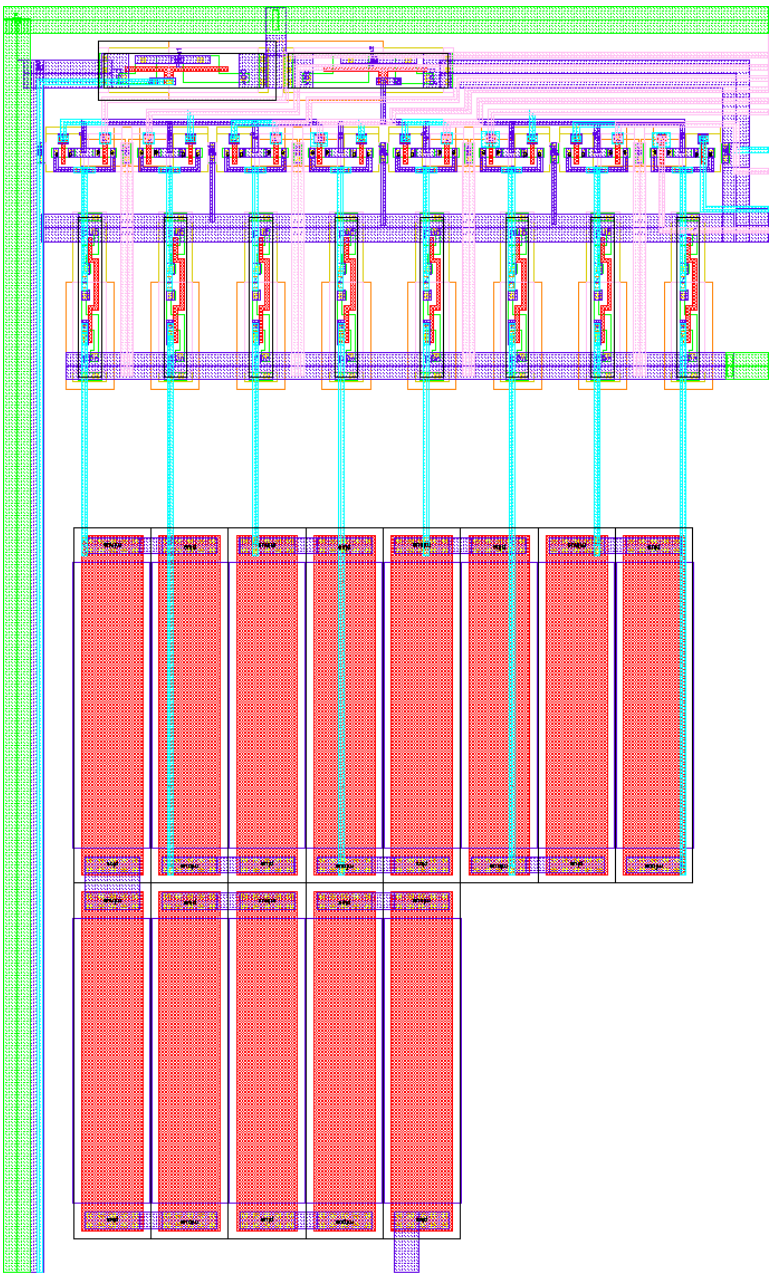

Figure A.6: The RC-delay in the pulse generator. The output inverters and pass gates are seen at the top. The two pulses are entering this part in the two blue wires at the bottom, and the delayed pulses leave at the two inverters at the top. The signals from the decoder are entering from right at the top. The thick lines are the supply lines.

Appendix B

PCB and Measurement setup

B.1 Measurement setup

Pictures of the measurement setup of the two circuits are included in figures B.1 and B.2. The ring oscillator PCB in figure B.1 is fairly straight forward, consisting only of decoupling capacitors and inputs and outputs. The TDMC PCB in figure B.2 use push button switches and schmitt triggers as inputs to the pulse generator. Both PCBs are connected to an oscilloscope through the shown coaxial cable connector.

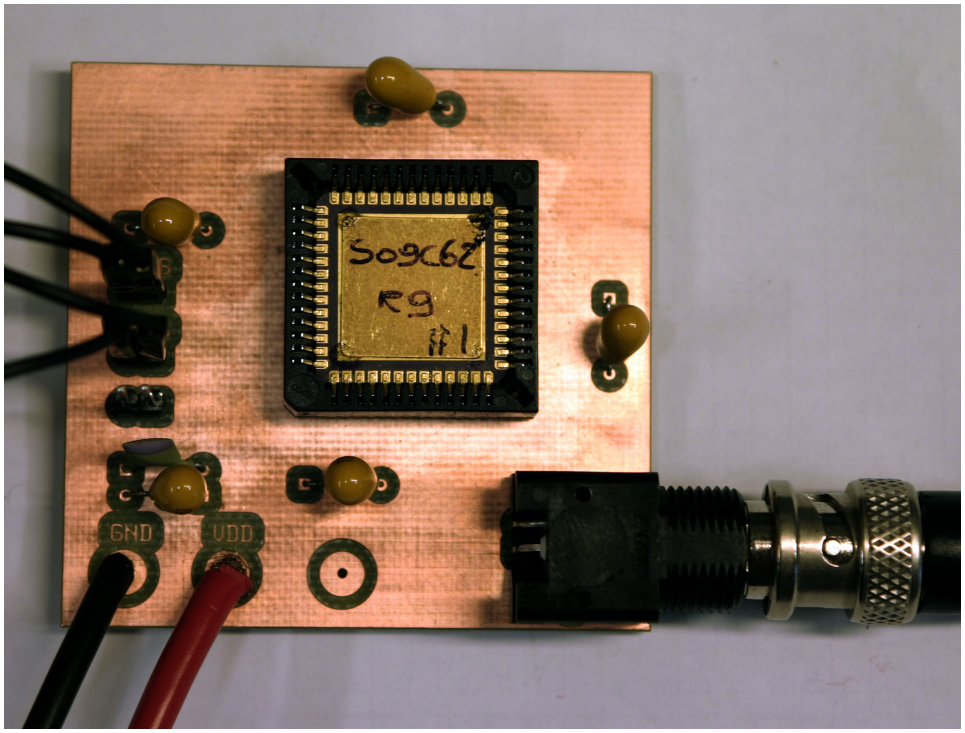

Figure B.1: Picture of the measurement setup for the ring oscillator

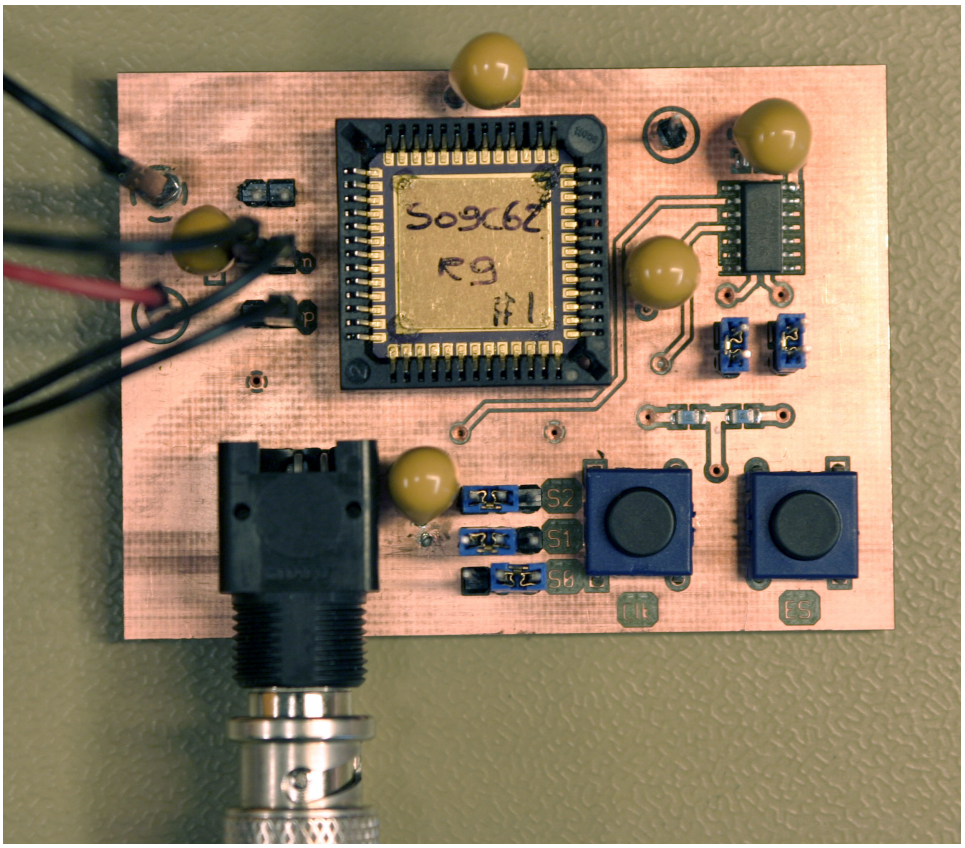

Figure B.2: Picture of the measurement setup for the TDMC

List of Figures

Acronyms

ALU	Arithmetic Logic Unit
AoA	Angle of Arrival
BB	Body Bias
BSN	Body Sensor Network
CAD	Computer Aided Design
CMOS	Complementary Metal-Oxide Semiconductor
DIBL	Drain Induced Barrier Lowering
FBB	Forward BB
FFD	Full-function device
GPIB	General Purpose Interface Bus
GPS	Global Positioning System
IC	Integrated Circuit
LSE	Least Square Estimation
LoS	Line of Sight
MEMS	Microelectromechanical systems
MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistor
NLoS	Non Line of Sight
NMOS	N-channel MOSFET
PCB	Printed Circuit Board
PDF	Probability Density Function

PMOS	P-channel MOSFET
RBB	Reverse Body Bias
RF	Radio Frequency
RFD	Reduced-function device
RFID	Radio Frequency Identification
RMS	Root Mean Square
RSS	Received Signal Strength
SNR	Signal-to-Noise Ratio
TDMC	Time Difference Measuring Circuit
TDoA	Time Difference of Arrival
ToA	Time of Arrival
ToF	Time of Flight
UWB	Ultra-Wide Band
VCO	Voltage-Controlled Oscillator
WSN	Wireless Sensor Network

Bibliography

[Abas 04] M. A. Abas, G. Russel, and D. Kinniment. "Design of sub-10-picoseconds on-chip time measurement circuit". In: *Design, Automation and Test in Europe Conference and Exhibition. Proceedings*, February 2004.

[Al K 04] J. N. Al-Karaki and A. E. Kamal. "Routing techniques in wireless sensor networks: a survey". *Wireless Communications, IEEE*, Vol. 11, pp. 6–28, December 2004.

[Ande 07] N. Andersen. *Active Echo High Precision Ranging in Wireless Sensor Networks*. Master's thesis, University of Oslo, May 2007.

[Bene 04] M.-G. D. Benedetto and G. Giancola. *Understanding Ultra Wide Band Radio Fundamentals*. Prentice Hall PTR, 1 Ed., 2004.

[Brya 01] A. Bryant *et al.* "Low-Power CMOS at Vdd=4kT/q". In: *Device Research Conference*, pp. 22–23, jun 2001.

[Buch 98] M. Bucher *et al.* "The EPFL-EKV MOSFET Model Equations for Simulation". Tech. Rep., Electronics Laboratories, Swiss Federal Institute of Technology (EPFL), jul 1998.

[Bulu 00] N. Bulusu, J. Heidemann, and D. Estrin. "GPS-less Low Cost Outdoor Localization for very small devices". *Personal Communications Magazine, IEEE*, Vol. 7, No. 5, pp. 28–34, October 2000.

[Camb 07] U. of Cambridge. "The Bat System". March 2007. *http://www.cl.cam.ac.uk/research/dtg/research/wiki/BatSystem*.

[Capk 01] S. Capkun, M. Hamdi, and J.-P. Hubaux. "GPS-free Positioning in Mobile Ad Hoc Networks". In: *Proceedings of the 34th Annual Hawaii International Conference on System Sciences, 2001*, January 2001.

[Chen 02] J. C. Chen, K. Yao, and R. E. Hudson. "Source localization and beamforming". *Signal Processing Magazine, IEEE*, Vol. 19, No. 2, pp. 30–39, March 2002.

[Dohe 01] L. Doherty, K. S. J. Pister, and L. E. Ghaoui. "Convex Position Estimation in Wireless Sensor Networks". In: *Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE (INFOCOM 2001)*, pp. 1655–1663, April 2001.

[Fang 75] F. F. Fang and H. S. Rupprecht. "High performance MOS integrated circuit using the ion implantation technique". *IEEE Journal of Solid-State Circuits*, Vol. sc-10, No. 4, pp. 205–211, Aug 1975.

[Flow 06] D. Flowers, K. Otten, and N. Rajbharti. "Microchip Stack for the ZigBee™Protocol". 2006.

[Forb 73] L. Forbes. "N-channel ion-implanted enhancement/depletion FET circuit and fabrication technology". *IEEE Journal of Solid-State Circuits*, Vol. sc-8, pp. 226–230, Jun 1973.

[Gran 06] K. Granhaug *et al.* "Body-bias Regulator for Ultra Low Power Multifunction CMOS Gates". In: *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1255–1258, may 2006.

[High 00] J. Hightower, G. Borriello, and R. Want. "SpotON: An Indoor 3D Location Sensing Technology Based on RF Signal Strength". Tech. Rep., University of Washington, February 2000.

[Hjor 06] H. A. Hjortland. *UWB impulse radar in 90nm CMOS*. Master's thesis, Department of Informatics, University of Oslo, 2006.

[Hu 03] C. Hu *et al. BSIM4.3.0 MOSFET Model - User's Manual*. University of California, Department of Electrical Engineering and Computer Sciences, 2003.

[IBM 07] IBM. "IBM and Maersk Logistics provice real-time cargo monitoring for global supply chain optimization". March 2007. *http://www.ibm.com/industries/government/doc/content/news/pressrelease/1383667109.html*.

[IEEE 03] IEEE. "IEEE Std 802.15.4™". May 2003.

[Ji 04] X. Ji and H. Zha. "Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling". In: *Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies. (INFOCOM 2004)*, pp. 2652–2661, March 2004.

[John 97] D. Johns and K. Martin. *Analog Integrated Circuit Design*. John Wiley & Sons, 1997.

[Kara 06] T. C. Karalar and J. Rabaey. "An RF ToF Based Ranging Implementation for Sensor Networks". In: *IEEE International Conference on Communications, 2006. (ICC '06)*, pp. 3347–3352, June 2006.

[Kesh 01] A. Keshavarzi *et al.* "Effectiveness of Reverse Body Bias for Leakage Control in Scaled Dual Vt CMOS ICs". *International Symposium on Low Power Electronics and Design*, pp. 207 – 212, aug 2001.

[Khal 06] D. Khalil *et al.* "Optimum Sizing of Power Grids for IR Drop". In: *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 481–484, may 2006.

[Kile 06] E. Kile. *Matriseinvertering på FPGA bed hjelp av QR-dekomponering*. Master's thesis, University of Oslo, December 2006.

[Lang 03] K. Langendoen and N. Reijers. "Distributed localization in wireless sensor networks: a quantitative comparison.". *Computer Networks: The International Journal of Computer and Telecommunications Networking*, pp. 499–518, November 2003.

[Limb 05] C. Limbodahl. *A spatial RAKE Receiver for Real-Time UWB-IR Applicatins*. Master's thesis, University of Oslo, 2005.

[Liu 00] X. Liu and S. Mourad. "Performance of Submicron CMOS Devices and Gates with Substrate Biasing". In: *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 9–12, may 2000.

[Liu 93] Z.-H. Liu *et al.* "Threshold Voltage Model for Deep-Submicrometer MOSFET". *IEEE Transactions on electron devices*, Vol. 40, pp. 86– 95, jan 1993.

[Mele 04] L. A. Melek *et al.* "Body-Bias Compensation Technique for Sub-Threshold CMOS Static Logic Gates". In: *Symposium on Integrated Circuits and Systems Design (SBCCI)*, pp. 267–272, sep 2004.

[Nare 03] S. Narendra *et al.* "Forward Body Bias For Microprocessors on 130-nm technology generation and beyond". *IEEE Journal of Solid-State Circuits*, Vol. 38, No. 5, pp. 696–701, May 2003.

[Ni 06] J. Ni, D. Arndt, P. Ngo, C. Phan, and J. Gross. "Ultra-Wideband Two-Cluster Tracking System Design with Angle of Arrival Algorithm". In: *IEEE Conference on Radar*, 2006.

[Nicu 01] D. Niculescu and B. Nath. "Ad hoc positioning system (APS)". In: *Global Telecommunications Conference, 2001. IEEE (GLOBECOM '01)*, November 2001.

[Oowa 98] Y. Oowaki *et al.* "A Sub-0.1um Circuit Design with Substrate-over-Biasing". In: *Digest of Technical Papers,IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 88–89, feb 1998.

[Patw 03] N. Patwari, A. O. Hero, M. Perkins, N. S. Correal, and R. J. O'Dea. "Relative location estimation in wireless sensor networks". In: *IEEE Transactions on Signal Processing*, pp. 2137–2148, August 2003.

[Pesc 07] D. Pescovitz. "Lab Notes: Research from the Berkeley College of Engineering". March 2007. *http://www.coe.berkeley.edu/labnotes/0903/pister.html*.

[Sasa 82] N. Sasaki. "Higher harmonics generation in CMOS/SOS Ring Oscillators". *IEEE transactions on Electron Devices*, Vol. 29, pp. 280–283, Feb 1982.

[Sava 01] C. Savarese, J. M. Rabaey, and J. Beutel. "Locationing in distributed ad hoc wireless sensor networks". In: *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01).*, pp. 2037–2040, IEEE, Piscataway, NJ, May 2001.

[Sava 02] C. Savarese, J. M. Rabaey, and K. Langendoen. "Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks". In: *Proceedings of the General Track: 2002 USENIX Annual Technical Conference*, May 2002.

[Savv 02] A. Savvides, H. Park, and M. B. Srivastava. "The bits and flops of the n-hop multilateration primitive for node localization problems". In: *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pp. 112–121, ACM Press, New York, NY, USA, 2002.

[Seta 95] K. Seta *et al.* "50% Active-power saving without speed degradation using standby power reduction (SPR) circuit". In: *Digest of Technical Papers, IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 318–319, feb 1995.

[Shoc 52] W. Shockley. "A unipolar "Field effect" Transistor". In: *Proceedings of the Institute of Radio Engineers*, pp. 1365–1376, Nov 1952.

[Smar 07] SmartFloor. "FCE Smart Floor". March 2007. *http://www3.cc.gatech.edu/fce/smartfloor/*.

[Taub 05] D. Taubenheim *et al.* "Distributed Radiolocation Hardware Core for IEEE 802.15.4". Tech. Rep., Motorola, 2005.

[Ubis 05] Ubisense. "Hardware Datasheet". February 2005.

[Wann 00] C. Wann, J. Harrington, R. Mih, and S. Biesemans. "CMOS with active well bias for low-power and RF/analog applications". In: *Symposium on VLSI Technology, Digest of Technical Papers*, pp. 158–159, jun 2000.

[Want 92] R. Want, A. Hopper, V. Falcão, and J. Gibbons. "The active badge location system". *ACM Trans. Inf. Syst.*, Vol. 10, No. 1, pp. 91–102, 1992.

[West 94] N. H. E. Weste and K. Eshraghian. *Principles of CMOS VLSI Design, A systems Perspective*. Addison-Wesley-Longman, 2 Ed., 1994.

[Whit 02] K. Whitehouse and D. Culler. "Calibration as parameter estimation in sensor networks". In: *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pp. 59–67, ACM Press, New York, NY, USA, 2002.

[Yang 06] G.-Z. Yang, Ed. *Body Sensor Networks*. Springer-Verlag London Limited, 1 Ed., 2006.

[Zaid 05] Z. R. Zaidi and B. L. Mark. "Real-time mobility tracking algorithms for cellular networks based on Kalman filtering". In: *IEEE Transactions on Mobile Computing*, pp. 195–208, March 2005.