Turun yliopisto
University of Turku

NEAR-THRESHOLD COMPUTING

Mika-Petteri Kutila

2013
Master of Science (Technology) Thesis
University of Turku
Department of Information Technology

Supervisors:
D.Sc. (Tech.) Teijo Lehtonen
Adj.Prof., Ph.D. Juha Plosila

TURUN YLIOPISTO
Informaatioteknologian laitos

MIKA-PETTERI KUTILA: Near-Threshold Computing

Diplomityö, 87 sivua
Järjestelmäelektroniikka
Maaliskuu 2013

Valmistustekniikoiden kehittyessä IC-piireille saadaan mahtumaan yhä enemmän transistoreja. Monimutkaisemmat piirit mahdollistavat suurempien laskutoimitusmäärien suorittamisen aikayksikössä. Piirien aktiivisuuden lisääntyessä myös niiden energiankulutus lisääntyy, ja tämä puolestaan lisää piirin lämmöntuotantoa. Liiallinen lämpö rajoittaa piirien toimintaa. Tämän takia tarvitaan tekniikoita, joilla piirien energiankulutusta saadaan pienennettyä. Uudeksi tutkimuskohteeksi ovat tulleet pienet laitteet, jotka seuraavat esimerkiksi ihmiskehon toimintaa, rakennuksia tai siltoja. Tällaisten laitteiden on oltava energiankulutukseltaan pieniä, jotta ne voivat toimia pitkiä aikoja ilman akkujen lataamista.

Near-Threshold Computing on tekniikka, jolla pyritään pienentämään integroitujen piirien energiankulutusta. Periaatteena on käyttää piireillä pienempää käyttöjännitettä kuin piirivalmistaja on niille alunperin suunnitellut. Tämä hidastaa ja haittaa piirin toimintaa. Jos kuitenkin laitteen toiminnassa pystyään hyväksymään huonompi laskentateho ja pienentynyt toimintavarmuus, voidaan saavuttaa säästöä energiankulutuksessa.

Tässä diplomityössä tarkastellaan Near-Threshold Computing -tekniikkaa eri näkökulmista: aluksi perustuen kirjallisuudesta löytyviin aikaisempiin tutkimuksiin, ja myöhemmin tutkimalla Near-Threshold Computing -tekniikan soveltamista kahden tapaustutkimuksen kautta.

Tapaustutkimuksissa tarkastellaan FO4-invertteriä sekä 6T SRAM -solua piirisimulaatioiden avulla. Näiden komponenttien käyttäytymisen Near-Threshold Computing -jännitteillä voidaan tulkita antavan kattavan kuvan suuresta osasta tavanomaisen IC-piirin pinta-alaa ja energiankulusta. Tapaustutkimuksissa käytetään 130 nm teknologiaa, ja niissä mallinnetaan todellisia piirivalmistusprosessin tuotteita ajamalla useita Monte Carlo -simulaatioita. Tämä valmistuskustannuksiltaan huokea teknologia yhdistettynä Near-Threshold Computing -tekniikkaan mahdollistaa matalan energiankulutuksen piirien valmistaminen järkevään hintaan.

Tämän diplomityön tulokset näyttävät, että Near-Threshold Computing pienentää piirien energiankulutusta merkittävästi. Toisaalta, piirien nopeus heikkenee, ja yleisesti käytetty 6T SRAM -muistisolu muuttuu epäluotettavaksi. Pidemmät polut logiikkapiireissä sekä transistorien kasvattaminen muistisoluissa osoitetaan tehokkaiksi vastatoimiksi Near-Threshold Computing -tekniikan huonoja puolia vastaan. Tulokset antavat perusteita matalan energiankulutuksen IC-piirien suunnittelussa sille, kannattaako käyttää normaalia käyttöjännitettä, vai laskea sitä, jolloin piirin hidastuminen ja epävarmempi käyttäytyminen pitää ottaa huomioon.

Asiasanat: 6T SRAM, CMOS integroidut piirit, energiatehokkuus, FO4-invertteri, Near-Threshold Computing

UNIVERSITY OF TURKU
Department of Information Technology

MIKA-PETTERI KUTILA: Near-Threshold Computing

Master of Science (Technology) Thesis, 87 pages
Electronic Systems
March 2013

As the IC technology is advancing, larger amounts of transistors are fitted on single IC chips. More complicated chips are able to execute more calculations at a given time period, but higher activity uses more energy and that generates heat. The excessive heat limits the activity of the chips. Therefore, there is a continuous demand for techniques that enable the same operations with lower energy consumption. Very small devices that can be used to monitor human activities, buildings, bridges and so on are a new field of study which also needs low energy solutions, as the devices must be able to operate long periods of time without charging the batteries.

The Near-Threshold Computing is a technique for reducing the energy consumption of IC devices. The principle in the Near-Threshold Computing is to use lower supply voltage than the nominal, which the chip manufacturer has originally designed. This slows devices down and makes them unreliable. However, if these drawbacks can be tolerated energy savings can be achieved.

In this study, different aspects of the Near-Threshold Computing are discussed, first by exploring previous research in the literature, and then by conducting two case studies to research applying Near-Threshold Computing technique for two CMOS devices.

In the case studies, an FO4 inverter and a 6T SRAM were investigated by simulations. The behavior of these devices in the Near-Threshold Computing voltages can be considered covering a large portion of a conventional IC chip area and energy usage. A 130 nm technology was used. Actual manufacturing process products were modelled by running multiple Monte Carlo simulations. When this technology that has a low price point is combined with the Near-Threshold Computing technique, it is possible to produce reasonably priced low-power devices.

In this study, The Near Threshold Computing technique is shown to drop the energy usage significantly. On the other hand, the devices operate slower and the widely used 6T SRAM cells become unreliable. As countermeasures, longer paths in the logic circuits and larger transistor sizes in the memory structures are shown to be effective ways of compensating the downsides of the Near-Threshold Computing. The results give a basis for low-power IC circuit designing, if the normal supply voltage with no drawbacks should be used, or if reduced voltage levels should be used and the drawbacks tolerated somehow.

Keywords: 6T SRAM, CMOS integrated circuits, Energy efficiency, FO4 inverter, Near-Threshold Computing

Table of Contents

List of Figures iv

List of Tables vii

List of Symbols and Acronyms viii

1 Introduction 1

2 Background 5

2.1 Metal-Oxide-Semiconductor Field-Effect Transistors 5

2.1.1 Transistor 5

2.1.2 Field-Effect Transistor 6

2.1.3 Threshold Voltage 7

2.2 Complementary Metal-Oxide-Semiconductor 10

2.2.1 Complementary Metal-Oxide-Semiconductor Inverter 11

2.3 Static Random-Access Memory 12

2.3.1 Memory Column 14

2.3.2 Write Circuit 15

2.3.3 Read Circuit 15

2.3.4 The Future of The Static Random-Access Memory 19

3 Near-Threshold Computing 21

3.1 Motivation 21
3.2 Scaling Voltage Down 23
3.3 Variation and Mismatch 25
3.4 Performance Drop 26
3.5 Temperature Dependency Changes 27
3.6 Pipelining 28
3.7 Six-Transistor Static Random-Access Memory in Near-Threshold Computing 29
3.8 Sub-Threshold Region 30

4 CASE STUDIES 32
4.1 Process Corners 34
4.2 Monte Carlo Simulations 35

5 CASE STUDY: FO4 INVERTER 36
5.1 Propagation Delay 37
5.2 Propagation Delay Variation 39
5.3 Energy Consumption 42
5.4 Leak Current 46
5.5 Leak Energy Per Operation 47
5.6 Noise Tolerance Of the Inveter 50
5.7 Ring Oscillator 52

6 CASE STUDY: 6T SRAM 55
6.1 6T SRAM Test Arrangement 55
6.1.1 Input Signals 57
6.2 Errors In 6T SRAM 57
6.2.1 Pre-charging of the Bitlines 58
6.2.2 Minimum Size 6T SRAM Cell Simulations 60
6.2.3 Larger 6T SRAM Cell Simulations 62

6.3 Signal Noise Margin 63
6.4 Dynamic Energy Consumption 67
6.5 Leak Current 69
6.6 Propagation Delay 74

7 CONCLUSIONS 80

REFERENCES 85

List of Figures

2.1 Diagram symbols of MOSFETs . . . 8
2.2 Currents through MOSFETs, 1.2 V difference between *drain* and *source*, 27 °C temperature . . . 9
2.3 CMOS structure . . . 10
2.4 CMOS inverter . . . 12
2.5 Diagram of 6T SRAM cell structured of two inverters . . . 13
2.6 6T SRAM transistor diagram . . . 14
2.7 6T SRAM column with auxiliary circuits . . . 16
2.8 6T SRAM write operation . . . 17
2.9 Sense amplifier transistor diagram . . . 18
2.10 6T SRAM read operation . . . 19

3.1 MOSFET zero-bias threshold voltages V_{th0}, 130 nm technology, 10 000 simulations, TYP corners . . . 26

5.1 FO4 inverter test arrangement . . . 37
5.2 FO4 inverter propagation delays, maximum values, 1 000 Monte Carlo simulations . . . 39
5.3 FO4 inverter propagation delays, maximum values, 1 000 Monte Carlo simulations . . . 40

5.4 FO4 inverter propagation delay variations, note the different y axis scales 41
5.5 FO4 inverter propagation delay variation distributions, 1 000 Monte Carlo Simulations, SS corners , −25 °C temperature 43
5.6 FO4 inverter energy per operation mean values 44
5.7 FO4 inverter energy per operation mean values 45
5.8 FO4 inverter leak current maximum values, 1 000 Monte Carlo simulations 47
5.9 FO4 inverter leak current maximum values,, 1 000 Monte Carlo simulations, the worst-case FF corners 48
5.10 FO4 inverter leak energy per operation, 1 000 Monte Carlo simulations, the worst-case FF corners, note the different y axis 49
5.11 FO4 inverter noise tolerance test diagram 50
5.12 FO4 inverter noise tolerance test, *input* and *output* signals 51
5.13 FO4 inverter propagation delays from noise tolerance test, 162 different test situations, SS cornes, −25 °C temperature 52
5.14 FO4 inverter ring oscillator test arrangement 53
5.15 FO4 inverter propagation delays from the ring oscillator test, 1 000 Monte Carlo runs, worst-case SS corners and −25 °C temperature 54

6.1 6T SRAM test arrangement 56
6.2 6T SRAM test sequence, one square on the horizontal time axis represents 0.5 µs 57
6.3 6T SRAM transistor diagram 58
6.4 6T SRAM error counts, minimum size transistors, 10 000 Monte Carlo simulations 61
6.5 6T SRAM error counts, 300 nm inner transistor widths, 10 000 Monte Carlo simulations 63
6.6 6T SRAM logic error count comparison, 10 000 Monte Carlo simulations, FS corners 64

6.7 6T SRAM signal noise margin test bench, artificial noise sources added in the middle . . . 65
6.8 6T SRAM *read* operation butterfly curves, minimum signal noise margin square sizes, 100 Monte Carlo simulations . . . 66
6.9 6T SRAM signal noise margins, 100 Monte Carlo simulations . . . 68
6.10 6T SRAM energy consumption . . . 70
6.11 6T SRAM energy consumption, *read* operation, FS corners . . . 71
6.12 6T SRAM energy consumption deviations, 1 000 Monte Carlo simulations 72
6.13 6T SRAM node Q voltages during *read* operation, 500 Monte Carlo simulations, TYP corners, 25 °C temperature . . . 73
6.14 6T SRAM leak currents, 1 000 Monte Carlo simulations . . . 75
6.15 6T SRAM propagation delay maximum values, *read* operation, 1 000 Monte Carlo simulations, worst-case SS corners . . . 77
6.16 6T SRAM propagation delays, *write* operation, 1 000 Monte Carlo simulations, worst-case SS corners . . . 78
6.17 6T SRAM propagation delays, *read* operation, 1 000 Monte Carlo simulations, worst-case SS corners . . . 79

List of Tables

4.1 Specifications of the used technology (Circuits Multi-Projects, 2010) . . 34

6.1 Read error counts depending on the pre-charging value of the bitlines and the sense amplifier, 50 Monte Carlo simulations, TYP corners, 0 °C temperature 59

List of Symbols and Acronyms

σ	Standard deviation
V_G	Voltage on the *gate* terminal of a MOSFET
V_{SB}	Voltage over MOSFET terminals *source* and *base*
V_{SD}	Voltage over MOSFET terminals *source* and *drain*
V_{dd}	Supply voltage of a CMOS circuit. The *dd* comes from the word *drain*, as usually the supply voltage is connected to *drain* of a transistor.
V_{nth}	Threshold voltage of an NMOS transistor
V_{pth}	Threshold voltage of a PMOS transistor
V_{th0}	Zero-bias threshold voltage ($V_{SB} = 0$)
V_{th}	Threshold voltage
i_D	Current of the *drain* terminal of a MOSFET
i_S	Current of the *source* terminal of a MOSFET
t_p	Propagation delay
6T SRAM	Six-transistor SRAM
BJT	Bipolar junction transistor
BL	Bitline
CMOS	Complementary metal-oxide-semiconductor
DUT	Device under test, sometimes unit under test (UUT)
FET	Field-effect transistor

FinFET	Fin field-effect transistor
FO4	Fanout-of-four
IC	Integrated circuit
MOSFET	Metal-oxide-semiconductor field-effect transistor
NMOS	N-channel MOSFET
PMOS	P-channel MOSFET
SRAM	Static random-access memory
WL	Wordline, row address, a control signal of an SRAM

1 INTRODUCTION

The transistors continue to shrink in size. The first commercial 22 nm microprocessors were introduced in the year 2012. The smaller chips allow better energy efficiency and performance. Also, the number of transistors integrated on a single chip is increasing. The commercial products have exceeded 3 000 million transistor count on a single chip. When the complexity of a chip grows, the heat generation becomes the problem and the power consumption per transistor has to be lowered. If the manufacturing process technology is kept the same, this can be achieved by lowering the clock frequencies, lowering the supply voltages, or both. As a result the performance of the chip lowers as well, but it can be compensated to some extent with increased parallel processing. Also, different solutions are being invented to problems like leak currents and voltage variability, as these problems become more significant when the supply voltage is decreased. (K. Smith, Wang, & Fujino, 2012, pp. 9–10)

New technologies emerge continuously for fitting more transistors into a single chip. Two examples of these are the Fin Field-Effect Transistor (FinFET) (Islam, Akram, Imran, & Hasan, 2010; Carlson et al., 2010) which is a new structure for transistors, and the three-dimensional chips which are multiple layer chips. FinFETs are smaller, more energy efficient and they have less leak currents than the traditional transistors. The chip manufacturer Intel is implementing the 22 nm microprocessors with FinFETs (Turley, 2011).

Although both performance and energy efficiency are generally better with smaller technologies, their dependence on the technology size is not the same with the technologies under 65 nm. This is because the heat generation limits the power that the chip can handle (Dreslinski, Wieckowski, Blaauw, Sylvester, & Mudge, 2010, p. 254). The calculation power can not be increased any more by simply adding more transistors on a chip. The electric power consumption has to be considered as a limiting factor and the individual circuits inside a chip have to be designed to use less power than before.

The low power consumption is an important aspect in designing small carry-on devices which all benefit from a long battery life. It is important also in big data centers which need large powering and cooling systems. The designing of very small sensor-based devices that can be implanted in the body of a human and used as monitors or actuation medical devices is a new field of study. This kind of devices could also be used as environmental monitors, for example in building or bridge structures (Dreslinski et al., 2010, p. 254). One future vision is that small wireless sensor nodes could be used in any physical object. The nodes would form the so-called Internet-of-Things. The Internet Protocol Version 6 has enough address space for all of them. The Internet-of-Things would enable in today's standards incredible applications in multiple fields, as almost every object around people would be networked and they would adapt to people's needs. The number of devices would be huge and each device should be very energy efficient to enable long operational times without the dependency on charghing the battery. The needed energy should be harvested from the environment or the batteries should last several years. (Bol, J. De V., et al., 2013, p. 20; H. De M., 2005, p. 29)

Many application areas do not need high performance chips and they would be comfortable with low performance. Very low-power and low performance devices would be ideal for applications of this kind. Also, if they need to be active only short periods, they can save power by staying mostly idle.

The power usage is closely connected to the supply voltage level of an electric device. Lowering the supply voltage is an effective way of reducing the power consumption. There are several studies about using the digital circuits with lower supply voltages than they were originally designed for. Their goal is to achieve power savings while using the same manufacturing technologies as before.

The manufacturers of the Metal-Oxide-Semiconductor Field-Effect Transistors (MOSFETs) have each determined a nominal value of the voltage supply for each device. The nominal value has some margins inside which the supply voltage can hover while the transistor is considered behaving normally. Because the supply voltage has a significant influence on the CMOS circuit energy usage, the thought has risen that using supply voltages significantly lower than normally would result in power savings. However, the devices behave differently under the nominal voltages, but in the case these drawbacks are carefully considered and solutions to tolerate them are used the lower power consumption might be achieved. This kind of low-power technique could be applied to technologies with low price point to produce reasonably priced low-power devices. This would be beneficial as the high-end technologies that use less power by nature are very expensive if the manufacturing volumes are modest.

Each MOSFET has a threshold voltage. When the input voltage shifts over it, transistor shifts from non-conducting mode to conducting mode or vice versa. The Near-Threshold Computing is a technique, in which the supply voltage is reduced from the nominal region to the near-threshold region. This means that the supply voltage is reduced significantly, but not under the threshold voltage. In this study the Near-Threshold Computing is discussed. Also, simulations are conducted to study the behavior of a 130 nm technology with Near-Threshold Computing voltages. The results of this study build basis for the future development of low-power devices using the Near-Threshold Computing technique. The results can also be used as background when designing high level design flows.

This thesis is organized as follows. In the Chapter 2 the basic behavior of MOSFETs and the concept of the their threshold voltages are explained. Also, the CMOS technique, the mostly used technique of constructing logic devices from MOSFETs, is discussed. The structures of a simple inverter and a widely used memory cell, 6T SRAM, are presented as two examples of CMOS devices. Also, auxiliary devices needed for operating an array of 6T SRAM cells are presented. In the Chapter 3 the motivation and the characteristics of the Near-Threshold Computing technique are explained. The basis is in the previous studies and literature.

In the Chapter 4 the two case studies and the used technology are introduced. In the Chapter 5 test arrangement for simulations of FO4 inverter is presented. Also, the results of the simulations using Near-Threshold Computing voltages are presented and they are compared with the nominal supply voltage behavior. In the Chapter 6 the test bench and simulation tests for 6T SRAM cell are presented. Also the 6T SRAM is tested with different voltages and in different temperatures to find out its suitability to the Near-Threshold Computing. In the Chapter 7 conclusions about the Near-Threshold Computing are made based on the previous studies and the conducted simulations.

2 BACKGROUND

2.1 Metal-Oxide-Semiconductor Field-Effect Transistors

2.1.1 Transistor

Transistors are the basic building blocks of modern electronic devices. Their operation is based on the properties of semiconductor materials. A transistor has at least three terminals. The voltage across or the current through two of the terminals determines the current through another pair of terminals. The output current is in proportion to the input voltage or current, thus the input signal controls the magnitude of the output current. Because the power of the output signal can be larger than the power of the input signal, transistors can be used as amplifiers. Transistors can also be used as switches, to just control the output current on and off. The transistor switches are widely used to construct logic gates in digital computing.

There are two kinds of transistors: Bipolar Junction Transistors (BJTs) and Field-Effect Transistors (FETs). The BJTs are controlled by the current through the node called the *base*. The FETs on the other hand are controlled by the voltage on the terminal called the *gate*. The FETs have terminals called the *gate*, the *source* and the *drain*. The voltage at the *gate* controls the current between the *source* and the *drain*. Usually, FETs have a

fourth terminal called the *base* (also called the *body*, the *bulk* or the *substrate*).

The FET manufacturing process is nowadays simpler and the sizes of the devices are smaller than BJT's (Sedra & K. C. Smith, 2010, p. 232). Amongst some other things, these are the reasons the FETs are the mainstream transistor nowadays. This is why the BJTs are not discussed in further detail here.

2.1.2 Field-Effect Transistor

The operation of the FETs is based on a phenomenon, where the voltage on the *gate* terminal causes an electric field which controls the shape of the conductivity channel of charge carriers in the semiconductor material. The channel connects the *drain* and the *source* and allows current to flow between them. There are two kinds of FETs: N-channel FETs that use electrons and P-channel FETs that use holes as the charge carriers. (Sedra & K. C. Smith, 2010, p. 235)

FETs are used in digital devices in a way that there are no dynamic currents flowing through the transistors when they are in a standby state. During a state transition the current flows between the *drain* and the *source*, but the circuits are designed in a way that there is only briefly a short circuit between the voltage source and the ground. This makes FETs use less power than BJTs. Small leak currents, on the other hand, are present at any moment. They are getting more significant if compared with the dynamic currents when technologies shrink. Also, when supply voltages are downscaled under the threshold the leak currents grow.

A Metal-Oxide-Semiconductor Field-Effect Transistor (MOSFET) is a FET manufactured of layers of metal, oxide and semiconductor. The oxide layer acts as an insulator between the *gate* which is made of metal and the *body* which is made of semiconductor. MOSFET is the most widely used electronic device in Integrated Circuit (IC) design. The name

MOSFET is so popular that it is in many cases used also for the FETs that have the gate electrode made of something else than metal. The n-channel MOSFET is called the NMOS transistor and the p-channel MOSFET is called the PMOS transistor. (Sedra & K. C. Smith, 2010, p. 231–235)

2.1.3 Threshold Voltage

The threshold voltage V_{th} is such voltage that when applied to the *gate* terminal of a MOSFET the conducting channel is formed between the *drain* and the *source* (Sedra & K. C. Smith, 2010, p. 235).

Let us consider a situation where the *drain* and the *source* of a MOSFET are connected between the source voltage V_{dd} and the ground. This situation is demonstrated in the Figure 2.1 for an NMOS and a PMOS. Usually, the *drain* of the NMOS and the *source* of the PMOS is connected to the higher potential and that is the case also in the Figure 2.1. If the transistor is an NMOS, the current starts to flow between the *drain* and the *source* when the voltage on the *gate* terminal is over the threshold value V_{nth}. Thus, the current, which is i_D in the Figure 2.1, begins to flow when $V_G > V_{nth}$. V_G is the voltage applied to the *gate* terminal and V_{nth} is the threshold voltage value of the NMOS type transistor. In the case of a PMOS, the current is present when the voltage on the *gate* terminal is under the V_{dd} subtracted by the threshold voltage value V_{pth}. Thus, the current i_D is present when $V_G < (V_{dd} - V_{pth})$. V_G is the voltage applied to the *gate* terminal, V_{dd} is the supply voltage and V_{pth} is the threshold voltage value of the PMOS type transistor.

The voltage V_{GS} between the *gate* and the *source* is marked in the Figure 2.1(a) and the voltage V_{SG} between the *source* and the *gate* is marked in the Figure 2.1(b). These are the voltages that have to be over the threshold values, that is $V_{GS} > V_{nth}$ and $V_{SG} > V_{pth}$, for the current i_D to flow. Notable in the Figure 2.1 is, that the currents i_D and i_S are

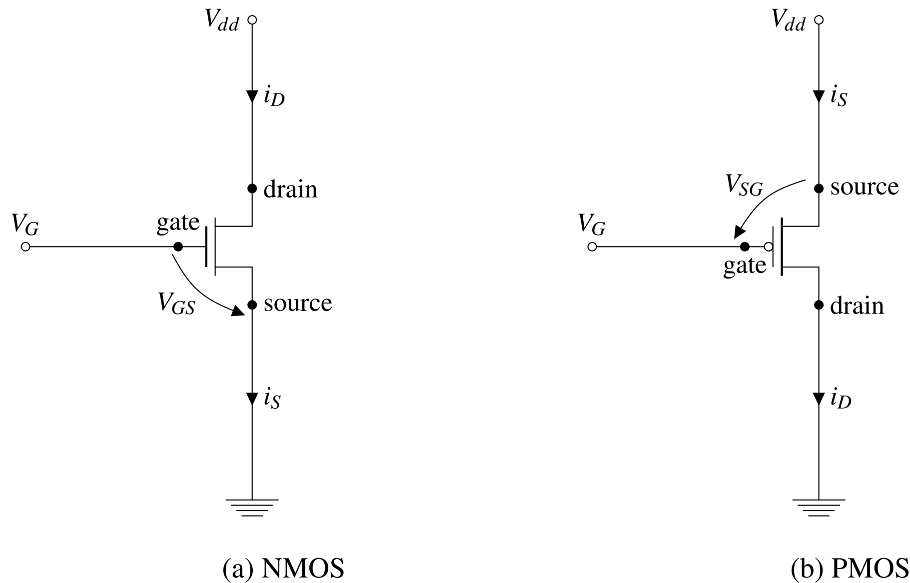
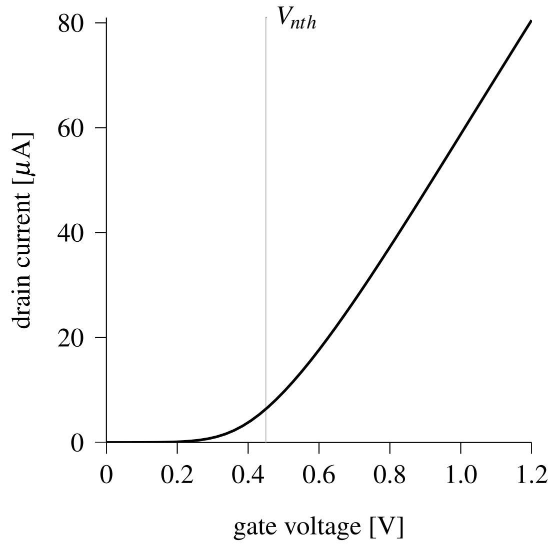


Figure 2.1: Diagram symbols of MOSFETs

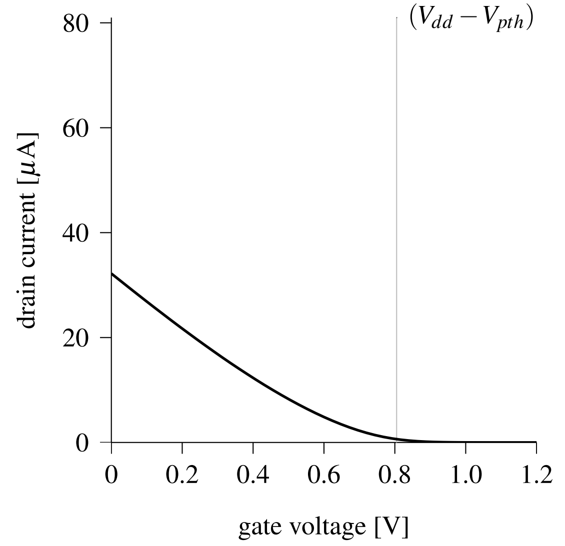
practically the same current that flows through the transistor. In the normal situations, there is practically no current through the *gate*.

The Figure 2.2 shows the behavior of the i_D in MOSFETs. These figures are from the simulations conducted in the case studies in the Chapter 5 and the Chapter 6. V_{nth} and V_{pth} are marked in the Figure 2.2 as the manufacturer has reported them. The simulated currents as functions of *gate* voltage are shown in the Figure 2.2(a) and the Figure 2.2(b). It can be seen from the figures that when the *gate* voltage is over V_{nth} (NMOS) or under $(V_{dd} - V_{pth})$ (PMOS) the current starts to flow between the *drain* and the *source*. The behavior of the NMOS current under V_{nth} and PMOS current over $(V_{dd} - V_{nth})$ is exponential. This can be seen from the Figure 2.2(c) and the Figure 2.2(c), where the y-axis is logarithmic. Here the straight line indicates exponential behavior.

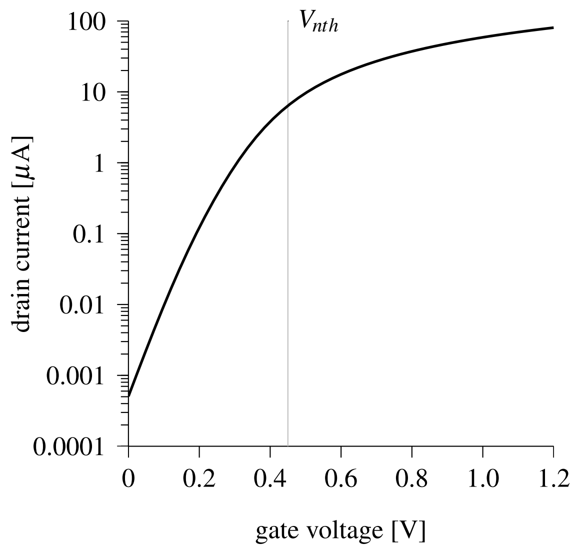
The threshold voltage of a transistor can be defined in many ways (Yu et al., 2010, p. 625). Some define it so, that when V_{GS} rises over it in an NMOS or V_{SG} falls under it in a PMOS, i_D is greater than some value, for example 1 µA. Some define it so that a straight line is drawn along V_G–i_D plot in the high V_G values in the Figure 2.2(a) or the low



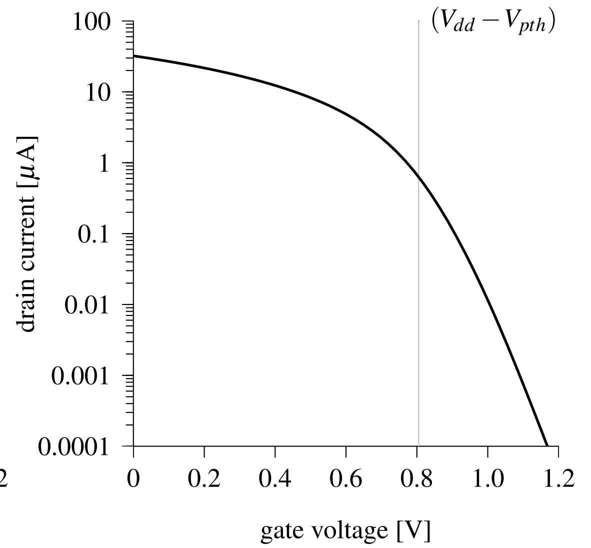
(a) NMOS, linear current scale



(b) PMOS, linear current scale



(c) NMOS, logatithmic current scale



(d) PMOS, logatithmic current scale

Figure 2.2: Currents through MOSFETs, 1.2 V difference between *drain* and *source*, 27 °C temperature

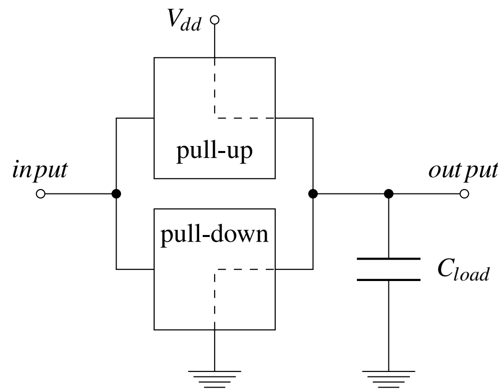


Figure 2.3: CMOS structure

V_G values in the Figure 2.2(b); the V_{th} would be the value of V_G where the line crosses its axis. These are the reasons why it is not easy to compare the results of the previous studies conducted with different technologies. Each technology should be carefully studied when considering scaling down the supply voltage.

2.2 Complementary Metal-Oxide-Semiconductor

Complementary Metal-Oxide-Semiconductor (CMOS) is a technique for constructing digital logic circuits from Field-Effect Transistors (FETs). Basically, the CMOS circuits have a separate pull-up and pull-down circuits as illustrated in the Figure 2.3. When a CMOS circuit is in a standby state, *output* of the circuit is connected either to V_{dd} through the pull-up circuit or to the ground through the pull-down circuit. The pull-up and the pull-down circuits are designed in the way, that the value of *input* signal determines where *output* is connected to.

When the circuit changes its state the short circuited connection of *output* through the pull-up or the pull-down circuit is opened and the other one is closed. Here open means that there is no connection, and closed means that a connection exists. *output* of the CMOS circuit is connected to the inputs of other circuits which are seen as load capacitances.

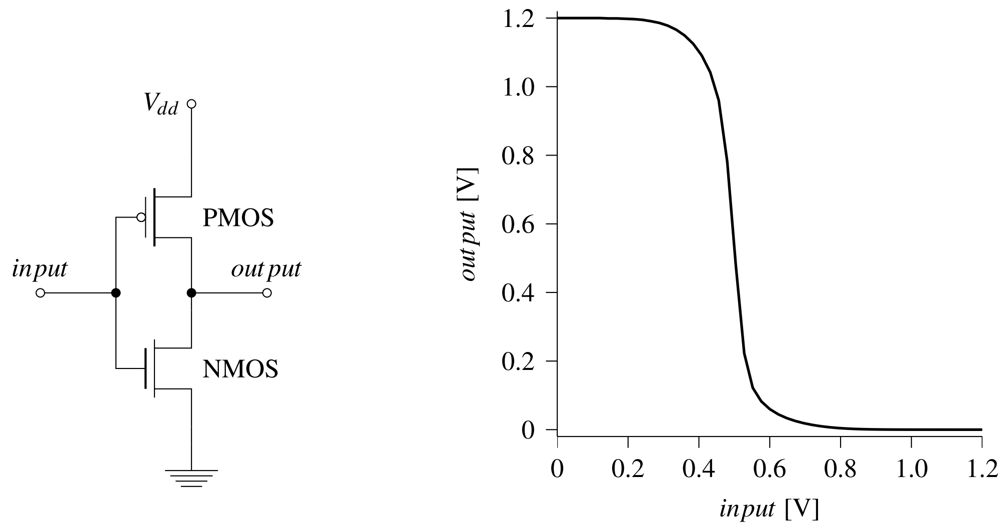
This being the case, C_{load} in the Figure 2.3 is just representing the following circuits that *output* signal is driving. During a state change, these load capacitances have to be charged or discharged through the pull-up or pull-down circuit. After the charging or discharging there is no current through the CMOS circuit besides the leakage which is always present at some extent.

2.2.1 Complementary Metal-Oxide-Semiconductor Inverter

The simplest CMOS circuit is an inverter that is constructed from one PMOS and one NMOS transistor. This kind of inverter is shown in the Figure 2.4(a). The PMOS transistor acts as the pull-up circuit and the NMOS transistor acts as the pull-down circuit. It is said that a transistor is closed when the *drain* and the *source* are short circuited, like a switch is closed when the current can flow through it. Mutually, a transistor is open when there is no connection between the *drain* and the *source*, like a switch is open when the current is not able to flow through it.

When *input* signal is connected to the ground the PMOS is closed and the NMOS is open. In this situation, the current can flow through the PMOS as *output* is connected to V_{dd}. The load on *output* starts charging and after the charging is done *output* signal is at the same potential as V_{dd} and there is no more any currents present. When *input* is at V_{dd}, the PMOS is open and the NMOS is closed. The output load discharges through the NMOS and *output* signal settles at the ground voltage.

When using nominal voltages, the leak currents are small if compared with the dynamic currents. Therefore, usually the dynamic currents that flow during the state change are the main cause of the inverter energy consumption. The same is true for all CMOS circuits and this is one of the benefits of using CMOS technique. Energy is mainly consumed only during the state changes.



(a) Transistor model

(b) Input-output relation, minimum size transistors, 1.2 Vsupply voltage

Figure 2.4: CMOS inverter

For a brief moment during the state change both the NMOS and the PMOS are conducting and V_{dd} and the ground are short circuited. The CMOS inverter behaves inherently so that this time stays short even if the input signal would be slow. An inverter of one PMOS and one NMOS transistor was simulated in the Chapter 5, and the result in the Figure 2.4(b) is an example of behavior of a CMOS inverter. The transition on *output* happens during a small range of *input* voltage change.

2.3 Static Random-Access Memory

As it was discussed in the Subsection 2.2.1, a CMOS inverter has low standby currents and low leak currents. Generally, these are desired characteristics of an electronic device. Therefore, a memory circuit, which has a construction of two CMOS inverters and two access transistors, has been widely used. That kind of memory is called the The Six-Transistor Static Random-Access Memory (6T SRAM) and the structure is illustrated in

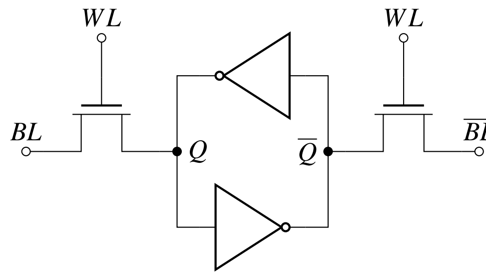


Figure 2.5: Diagram of 6T SRAM cell structured of two inverters

the Figure 2.5. Two inverters are connected to each other forming a loop. The node Q in the Figure 2.5 has the electric potential that represents the logic value stored to the memory cell; $\overline{Q}$ is the inversion of that. Two complementary wires called the bitlines BL and $\overline{BL}$ run on each side of a column of memory cells. The bitlines are used for transferring the values to and from the memory cells. Access transistor switches are used to connect a single SRAM node to the bitlines when needed. The wordline WL is the control signal that is used to make the connection.

The standby and the leak currents of 6T SRAM are small and the soft error tolerance is better than its predecessors, which had transistors but also resistors as their components. The resistor is a power hungry component when compared to a transistor. Soft error is a situation, where the memory cell occasionally behaves erroneously, even when it is not physically broken. (Sharma, 2003, pp. 19–21)

In the standby state WL is down, two NMOS access transistors are open, and the values of Q and $\overline{Q}$ are kept in the memory. When a logic value is read from the memory, WL is risen, the access transistors are closed, and the electric potential stored in the memory cell is spread to the bitlines BL and $\overline{BL}$. The bitlines are then read with an auxiliary circuit, which is called the *sense amplifier*.

When a logic value is written to the memory, the WL is risen, the access transistors are closed, and the wanted logic value and its inversion are driven to the bitlines BL and $\overline{BL}$. The drivers have to be strong enough to make the logic values inside the memory cell

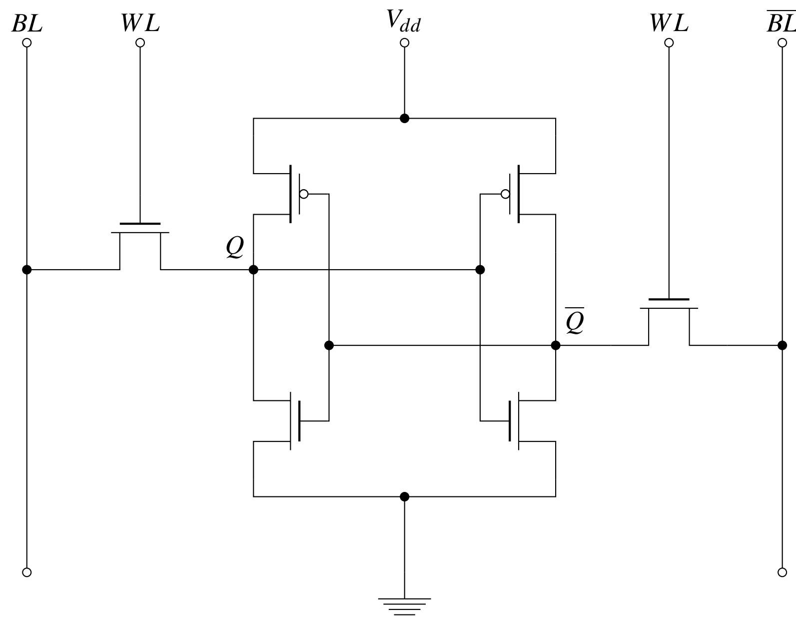


Figure 2.6: 6T SRAM transistor diagram

switch.

The transistor level structure of the 6T SRAM can be seen in the Figure 2.6. The voltage level of the node Q represents the value stored in the memory cell. In the standby state Q is connected to V_{dd} and $\overline{Q}$ to the ground, or vice versa. In this static state there are no currents present other than leakages.

2.3.1 Memory Column

A memory circuit is constructed from multiple columns and rows of memory cells. Basically, each column has multiple memory cells connected in parallel. To read from and to write to a memory cell some auxiliary circuits are needed. These would include at least logic for selecting the row and the column, the drivers for writing to the memory, the circuit for reading the values from the cells and the input/output logic of the circuit. Also, different circuits to handle pipelining, burst sequences or other functions might be needed. (Sharma, 2003, p. 21)

The basic operation of the memory can be understood by observing the behavior of a single memory column, which is shown in the Figure 2.7. A column is made of multiple 6T SRAM cells and one circuit for writing and one circuit for reading. Pre-charging of the bitlines is needed before *read* operation, so there is a pre-charger for each bitline. Each memory cell has its own WL control signal, that represents the row address. It is used to select the cell that is read or written to.

2.3.2 Write Circuit

The *write circuit* is basically a switch and a driver. The signal $write$ acts as an enable signal that lets the signal $input$ and its inversion through to the bitlines. The *write circuit* has to be able to drive the two bitlines and convert a state of a single memory cell. Therefore, the transistors of the drivers have to be stronger than the ones inside a single memory cell.

The Figure 2.8 shows the use of all the input signals when a *write* operation is conducted. Only the signals $write$ and WL are involved in the *write* operation.

2.3.3 Read Circuit

The *read* operation needs some preliminary actions. The bitlines and the inner nodes of the *sense amplifier*, the ones that are illustrated as $output$ and $\overline{output}$ in the Figure 2.9, have to be pre-charged to the same electric potential. This way the sum of the charges in the bitlines and in the memory cells is always the same before the *read* operation. Hence, the timing of the *read* operation does not have to variate each time depending on the initial situation. Yeknami (2008) uses V_{dd} as the bitline pre-charging voltage. Sharma (2003, p. 112) and Baker (2010) use $\frac{V_{dd}}{2}$ as the bitline pre-charging voltage. Alorda, Torrens, Bota, and Segura (2009, p. 4) test different bitline voltages and determine that a preferable voltage would be somewhere between $\frac{2}{3} \cdot V_{dd}$ and V_{dd}. Different pre-charging voltages are

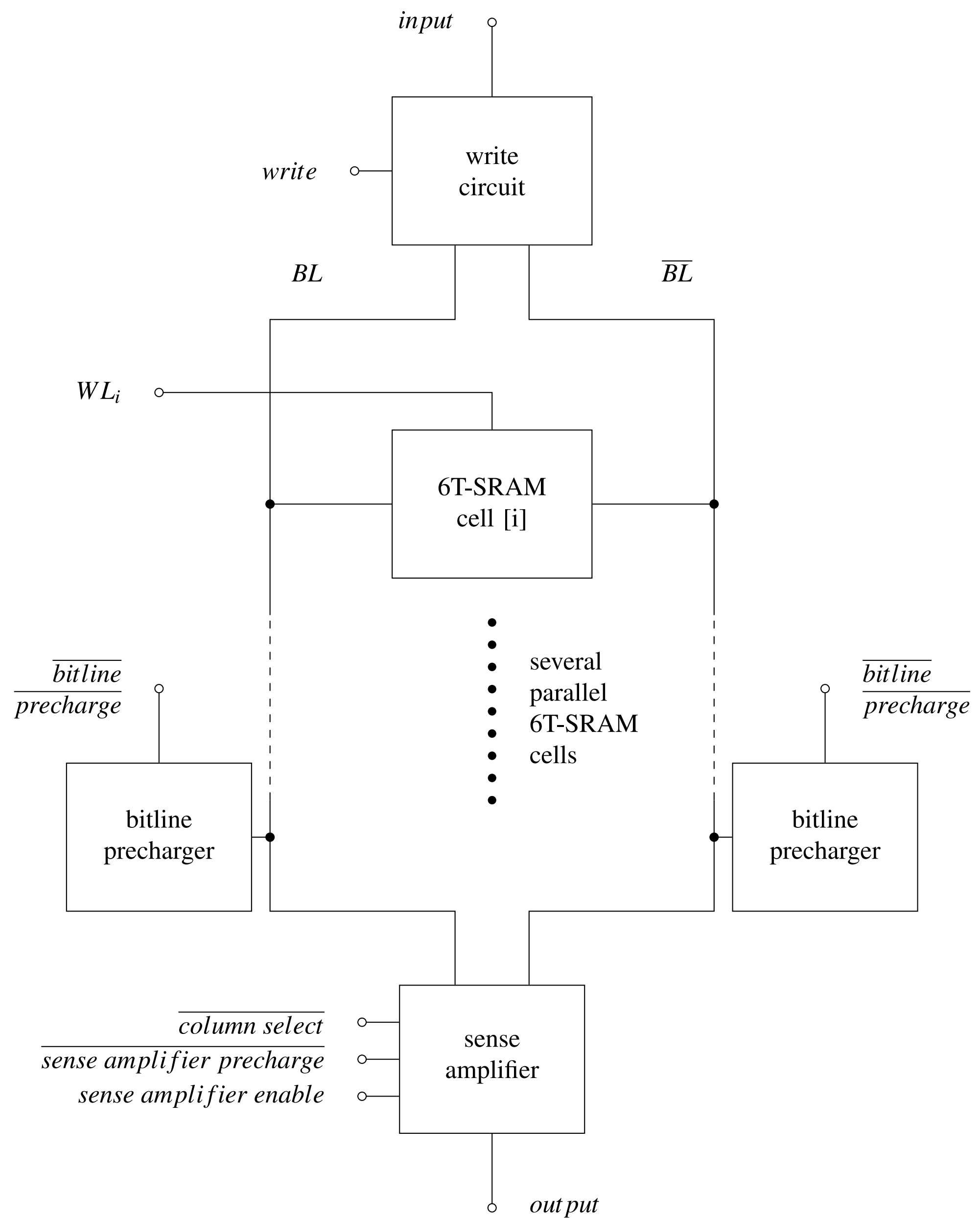


Figure 2.7: 6T SRAM column with auxiliary circuits

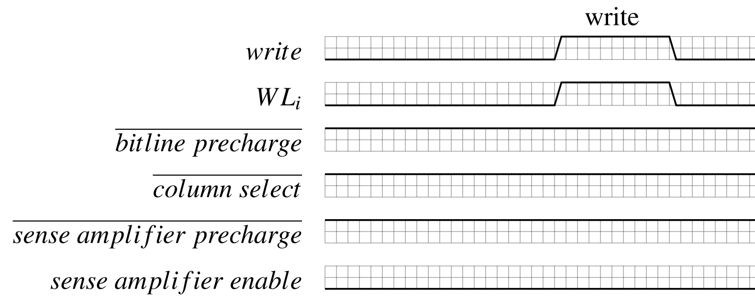


Figure 2.8: 6T SRAM write operation

tested in the Chapter 6.

After the pre-charghing, when the reading starts, the pre-chargers are turned off, and WL is raised to V_{dd}. While, the bitlines were pre-charged to V_{dd}, one of them starts to discharge through the 6T SRAM cell. The discharging takes place on that side of the 6T SRAM that has the logical 0 stored into. Because the transistors inside the 6T SRAM cell are designed to be small in size to save area and energy, the discharging happens slowly. It happens so slowly, that is has an effect on the overall performance of an SRAM circuit. This is why simple inverters, that would act as amplifiers, are not enough as the read circuit. A *sense amplifier* helps the memory cell in charging and discharging the bitlines during the *read* operation.

A *sense amplifier* speeds up the *read* operation significantly. It senses small voltage differences between the bitlines, and then it amplifies the signals so that the discharging happens, in addition to the SRAM cell, also through the sense amplifier. The transistors inside the sense amplifier can be bigger than the ones in the memory cells, because the overhead of the size of the *sense amplifier* is only a fraction per memory cell. If the transistors inside the memory cells were bigger, the overhead would be too big because of the great number of memory cells on circuits. Also, the overall energy consumption would be bigger.

The sequence that is used in the *read* operation can be seen in the Figure 2.10. After

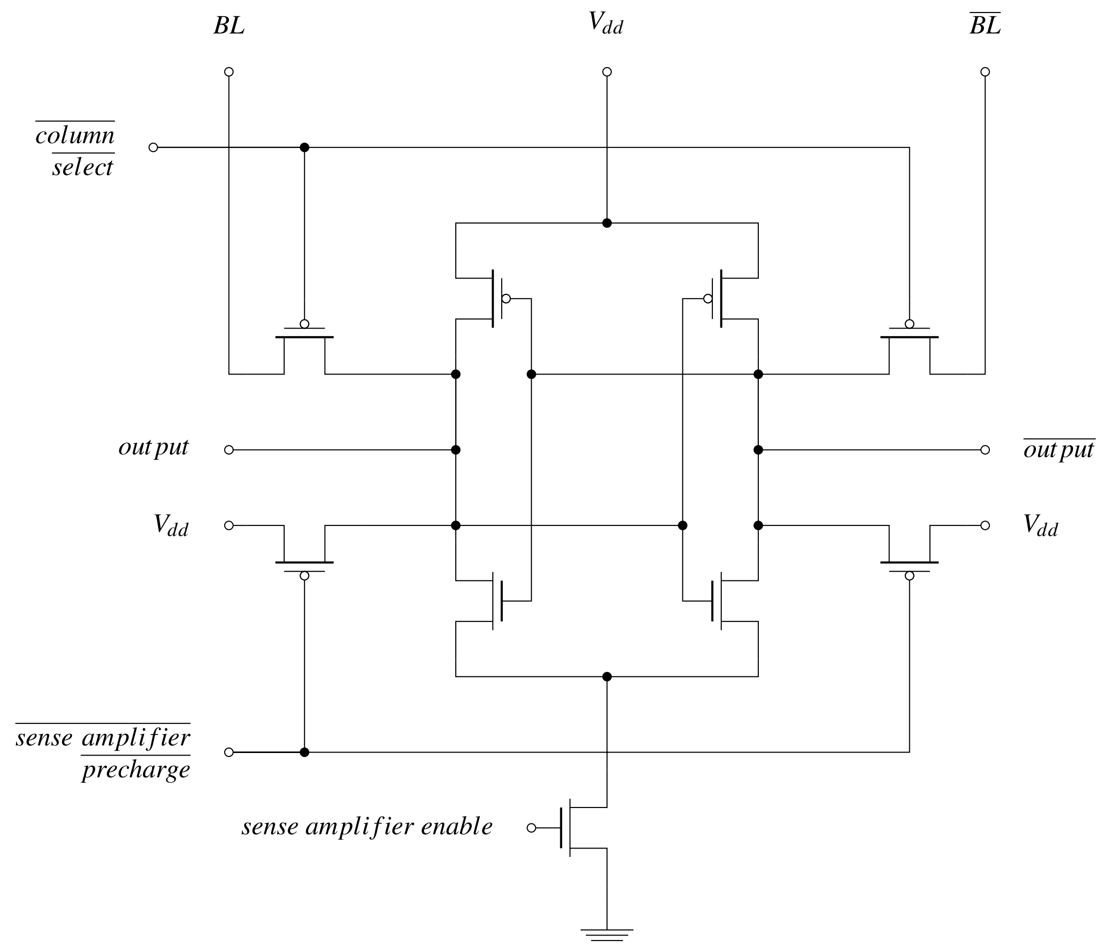


Figure 2.9: Sense amplifier transistor diagram

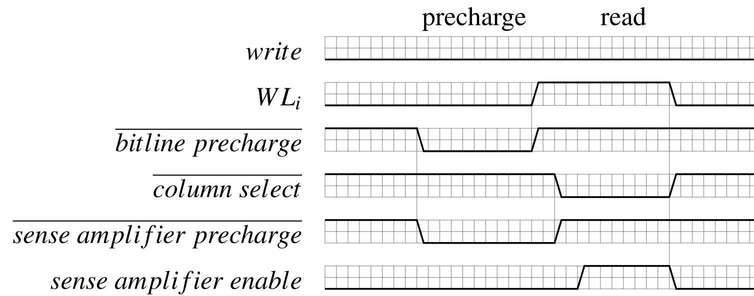


Figure 2.10: 6T SRAM read operation

the pre-charging of the bitlines and the *sense amplifier*, WL rises and a memory cell gets connected to the bitlines. The other bitline starts to discharge through the memory cell. After a small fraction of time, during which the bitlines have reached a voltage difference of some degree, pre-charge of the *sense amplifier* ends and $\overline{column\ select}$ signal is lowered. Now the electric potentials of the bitlines spread to the inner nodes of the *sense amplifier*. At this point the inverters inside the *sense amplifier* are not working as the circuit is not connected to the ground yet.

After still a small delay, during which the inner nodes of the *sense amplifier* have reached the same potentials as the bitlines, the *sense amplifier enable* signal is risen and the *sense amplifier* starts to work. At this point, the bitline that was discharging only through the memory cell discharges also through the *sense amplifier*. Because the bigger sizes of the transistors inside the *sense amplifier* the discharging is faster.

After this, the SRAM-like construct of the *sense amplifier* settles fast to a state where two output nodes have the electric potentials of V_{dd} and the ground. Now the output signals can be read by an external circuit.

2.3.4 The Future of The Static Random-Access Memory

SRAM is a widely used component in many applications. Following the Moore's law they have shrunk in size. Also the reliable operating voltage has gone lower, so the power

consumption has been decreasing. However, under 20 nm transistor size device variations caused by the manufacturing process limit both the shrinking of the transistor sizes and the decreasing of the operating voltages. (K. Smith et al., 2012, pp. 13–14)

High-K metal gates have been one solution. This has enabled thinner oxide layers and because of them also smaller transistors. Under 45 nm technologies high-K metal gates have been used to reduce V_{th} mismatch of the transistors. (K. Smith et al., 2012, pp. 13–14)

Design solutions as assisting circuits for *read* and *write* operations have been used under 32 nm technologies. These enable the same or lower supply voltages than before. (K. Smith et al., 2012, pp. 13–14)

The new transistor structure FinFET has emerged as a replacement for the traditional planar transistors. FinFET has a small semiconductor fin between the *drain* and the *source*. The *gate* terminal covers the fin from the top and from the sides. This way the inversion channel between the *drain* and the *source* is achieved more easily. FinFET is more energy efficient and has less leak currents. (Islam et al., 2010; Carlson et al., 2010; Turley, 2011)

3 Near-Threshold Computing

3.1 Motivation

In the history of IC technology, devices have always been shrinking in size. This has enabled the circuits to use smaller supply voltages and less power than previously. However, under the 65 nm technology the supply voltage levels have stayed somewhat the same. (Dreslinski et al., 2010, p. 254)

The dynamic power consumption of a CMOS circuit is a result of devices charging and discharging their load capacitances. The energy consumption of a single state transition follows the equation

$$U_E = \frac{1}{2} \cdot C_{load} \cdot V_{dd}^2, \tag{3.1}$$

where U_E is the potential energy that is stored in the load capacitance C_{load} when it is fully charged to V_{dd}. U_E is also the amount of work that the CMOS device has to do when changing state, because it has to charge or discharge C_{load} during each state transition. As the energy consumption is in relation to the square of V_{dd}, it can be significantly reduced by lowering V_{dd}. While the energy consumption has such dramatic reliance on V_{dd}, downscaling V_{dd} has become the main technique for reducing the dynamic energy consumption (Hanson et al., 2006, p. 469).

In the Equation 3.2, P_{total} is the total power dissipated by a CMOS device. $P_{dynamic}$ is the power used for charging and discharging the load capacitances in the device. $P_{short\ circuit}$ is the power used because the CMOS devices have a brief moment when there is a short circuit between V_{dd} and the ground. P_{static} is the power usage that is caused by the leak currents that are present all the time.

$$P_{total} = P_{dynamic} + P_{short\ circuit} + P_{static} \tag{3.2}$$

$$P_{dynamic} = \alpha \cdot f \cdot \frac{1}{2} \cdot C_{load} \cdot V_{dd}^2 \tag{3.3}$$

$$P_{short\ circuit} = \alpha \cdot f \cdot \frac{1}{2} \cdot (t_{I_{sc\ rise}} + t_{I_{sc\ fall}}) \cdot I_{sc\ max} \cdot V_{dd} \tag{3.4}$$

$$P_{leak} = I_{leak} \cdot V_{dd} \tag{3.5}$$

In the Equation 3.3 and the Equation 3.4, α is the activity of the device, or in the other words the average count of device state changes occurring at each clock cycle. f is the frequency of the clock signal. C_{load} in the Equation 3.3 is the load capacitance of the device. $t_{I_{sc\ rise}}$ and $t_{I_{sc\ fall}}$ in the Equation 3.4 are the short circuit current rise and fall times. $I_{sc\ max}$ is the maximum value that the short circuit current gets during a state transition. This equation approximates the short circuit current pulse as a triangular pulse. In the Equation 3.5, I_{leak} is the leak current through the device. As all of the terms in the Equation 3.2 are dependent on the supply voltage V_{dd}, they can all be reduced by lowering V_{dd}.

The low-power designs are an interesting field of study, because power savings are needed in many applications, for example in implantable medical devices, wireless sensor networks, self-powered RFID tags or whatever portable device (Wang, Calhoun, & Chandrakasan, 2006). Also, many applications are by nature such that they do not need high performance circuits. Therefore, they are multiple application areas for the low-power and low performance design techniques.

The performance drop caused by the low-power techniques can be compensated with

techniques like pipelining and parallelizing. This way also high-performance computing can benefit from low-power techniques. This is possible if the total calculation throughput can be increased while keeping the power usage the same (Wang et al., 2006), or if the calculation throughput stays the same and the power consumption can be decreased (Nam, Taeho, Bowman, De, & Mudge, 2005, p. 534). Also, using power saving methods only when the system is idle or just doing simple background routines is advantageous (Hanson et al., 2006, p. 469).

The new power saving techniques are also an opportunity to use older and cheaper technologies in low performance applications. The same older technologies could be used with lower frequencies and voltages than they were originally designed for, and energy savings could be achieved. This would be beneficial, when big manufacturing volumes are not needed or if the new technology is regarded too expensive.

3.2 Scaling Voltage Down

Even if the CMOS transistors are known to work in a correct way with much lower than the reported nominal voltages, usually the lowest operational voltage of a CMOS circuit is somewhere around 70% of the nominal voltage (Dreslinski et al., 2010, p. 255). Under that the performance and the reliability of the circuit begin to suffer.

While there is an opportunity to lower the supply voltage way under the threshold voltage V_{th} of the transistors, the lowest possible voltage might not be optimal. The application at hand might not be comfortable with such poor performance and especially the unreliability that the lowest voltages offer. The optimal energy per operation should be searched and the performance drop and the reliability issues should be estimated to be at a tolerable level for the application at hand. Here V_{th} is used as a common symbol for the NMOS and the PMOS threshold voltages.

In the MOSFETs of today, there is a cap between the 70 % of the nominal voltage and the threshold voltage V_{th}. The technology used in the Chapter 5 and the Chapter 6 has a theoretical cap of 0.39 V . This comes from equation

$$70\,\% \cdot 1.2\,\mathrm{V} - 0.45\,\mathrm{V} = 0.39\,\mathrm{V},$$

where 1.2 V is the nominal voltage and 0.45 V is the threshold voltage. So, there is some room to downscale the supply voltage if the performance loss is accepted and the reliability issues are tolerated or taken care of.

The Near-Threshold Computing is about finding an optimal combination of energy efficiency and reliability. The Near-Threshold Computing voltage regime lies significantly under the supply voltage, somewhere under $70\,\% \cdot V_{dd}$, but above V_{th}. When V_{dd} is dropped from the nominal value to the Near-Threshold Computing regime, energy used for a single operation is somewhat ten times lower. On the downside, also the delays of the device grow ten times higher. These numbers variate depending on the technology. (Dreslinski et al., 2010, p. 255)

When only the energy efficiency is concerned, the optimal point of the supply voltage is somewhere near V_{th}. The problem of using voltages around V_{th} is that the CMOS circuits behave differently under and over the threshold voltage (Harris, Keller, Karl, & Keller, 2010, p. 64). The techniques of managing low voltages is divided into two: under and over V_{th}. The technique is called the Near-Threshold Computing when the supply voltage is kept above V_{th}. The sub-threshold region computing does not have a well-established term.

The nominal use of FETs depends on the strong inversion in the semiconductor channel and the large gate overdrive voltages ($V_{GS} > V_{nth}$ on the NMOS and $V_{SG} < (V_{dd} - V_{pth})$ on the PMOS). In the Near-Threshold Computing the overdrive voltages are small, and the inversion is weak. Under V_{th} the current through a FET is exponentially dependent on

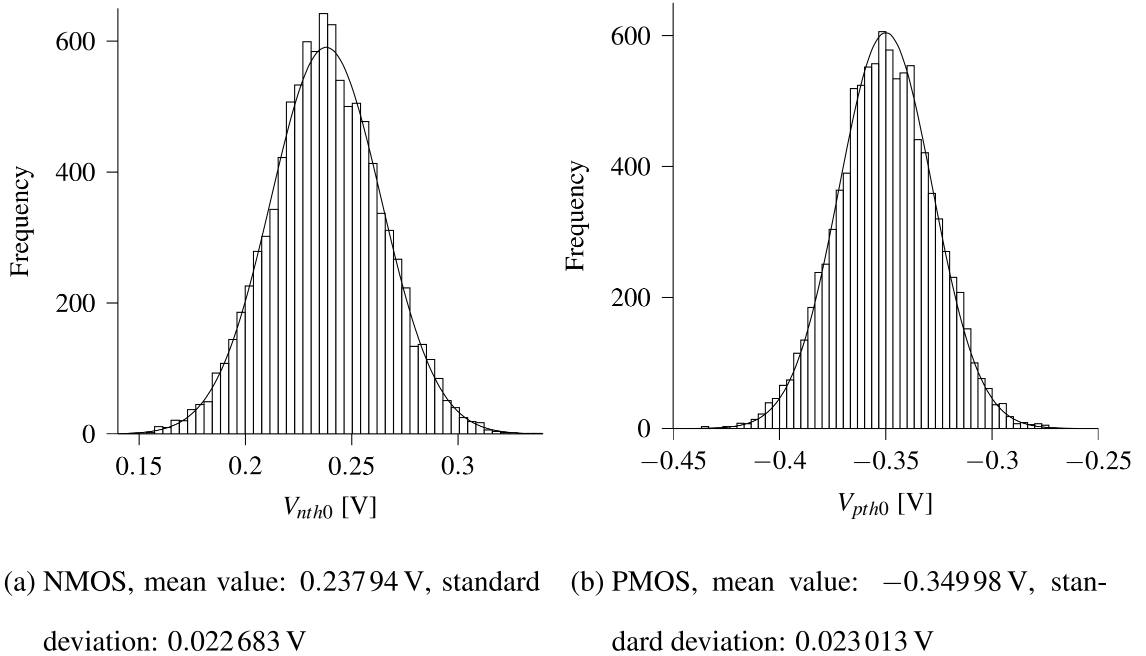
V_{dd} and V_{th} (Figure 2.2(c) and Figure 2.2(d)). (Hanson et al., 2006, p. 470)

It is proposed that the V_{dd} scaling method would not scale down together with the dimension scaling of the technologies. It is believed that for the traditional CMOS technologies 0.5 V is an optimal practical value of V_{dd}. This means that when that limit has been reached, some other energy efficiency techniques must be applied. (Chang et al., 2010, p. 218)

3.3 Variation and Mismatch

The manufacturing processes do not produce perfectly identical transistors. Their dimensions variate from one transistor to another. For example, the width and the length of the transistor channel, and the thickness of the oxide layer under the *gate* terminal, variate between each transistor. These size differences cause differences in the behavioral characteristics of the transistors. Especially when designing devices that use low voltages the variations in V_{th} have to be concerned. Experimental results have shown that V_{th} variates between transistors randomly and it has normal distribution. The variation is relative to the fluctuation of dopant atoms in the semiconductor channel material. The Figure 3.1 shows how V_{th} variation seems to be distributed normally also in the technology used in the simulations in the Chapter 5 and the Chapter 6. (Ding-Ming et al., 2006, p. 1)

The effect that the manufacturing process variability has on V_{th} is significantly bigger when V_{dd} is near or under V_{th}. This is because the behavior of a transistor depends exponentially on V_{dd} when V_{dd} is near or under V_{th}. This change in the behavior can be seen in the Figure 2.2(c) and the Figure 2.2(d). (Bo, Hanson, Blaauw, & Sylvester, 2005, p. 20)



(a) NMOS, mean value: 0.237 94 V, standard deviation: 0.022 683 V

(b) PMOS, mean value: −0.349 98 V, standard deviation: 0.023 013 V

Figure 3.1: MOSFET zero-bias threshold voltages V_{th0}, 130 nm technology, 10 000 simulations, TYP corners

3.4 Performance Drop

The performance of the device suffers from the drop of the supply voltage. The performance loss can be higher than 10 times in Near-Threshold Computinc voltages (Dreslinski et al., 2010, p. 254). Because different technologies have different V_{th} values, the performance drops in Near-Threshold Computing vary greatly depending on the technology (Yu et al., 2010, p. 625). Therefore the behavior of each design should be carefully investigated to ensure the characteristics in the Near-Threshold Computing or the subthreshold region.

Generally, dropping V_{dd} from over the 1.0 V regime to the Near-Threshold Computing regime to around 0.5 V can provide power efficiency improvements, even when the performance loss and unreliabilities are compensated with parallelism and other solutions. It is estimated that this kind of beneficial parallelism would increase the area of the de-

vice up to 4 times the original, and the power consumption would drop down to $\frac{1}{8}$ of the original. (Chang et al., 2010, p. 220)

When moving from the nominal source voltage to the Near-Threshold Computing region, the propagation delay variation grows. The global process variation has been reported causing the propagation delay to variate five times more in the Near-Threshold Computing region than in the nominal voltage region. In addition, also the variations in the temperature and in the source voltage level have more impact on the delay variation when the Near-Threshold Computing is used. (Dreslinski et al., 2010, p. 256)

3.5 Temperature Dependency Changes

The driving current of an NMOS is modeled in the equation

$$I_{on} \propto \begin{cases} \mu(T) \cdot e^{\frac{V_{GS}-V_{th}(T)}{S(T)}} & (V_{GS} < V_{th}) \\ \mu(T) \cdot (V_{GS} - V_{th}(T))^{\beta} & (V_{GS} \geq V_{th}) \end{cases}, \tag{3.6}$$

where I_{on} is the driving current (Yu et al., 2010, pp. 628). When the voltage on the *gate* is V_{dd}, the voltage on the *drain* is V_{dd} and the voltage on the *source* is zero, I_{on} is the current that flows through the transistor (Baker, 2010, p. 150). $\mu(T)$ is the carrier mobility in intrinsic to the process, β is the velocity saturation effect factor and $S(T)$ is the sub-threshold swing. V_{GS} is the voltage difference of *gate* and *source*. As in this case *source* is at the ground, V_{GS} is the *gate* voltage. $V_{th}(T)$ is the threshold voltage. Notable is, that $\mu(T)$, $S(T)$ and $V_{th}(T)$ are temperature dependent.

When the temperature decreases, $\mu(T)$ increases. This makes I_{on} increase. This is a well-known phenomenon with nominal voltages. However, with low voltages $S(T)$ decreases

and V_{th} increases when the temperature decreases, and because $S(T)$ and $V_{th}(T)$ have an exponential effect on I_{on}, they start to dominate it. This is why the low-voltage devices run slower in low temperatures.

When using Near-Threshold Computing, devices operate faster at higher temperatures. This is opposite to the behavior in the nominal voltage region. So, the worst-case situation in the Near-Threshold Computing is the low temperature. This must be kept in mind when Near-Threshold Computing designs are made and tested with worst-case conditions. The simulations conducted in the Chapter 5 show this phenomenon as well. (Yu et al., 2010, pp. 628–629)

3.6 Pipelining

The process variations have different effect on the propagation delay of each component. The delay variation of one component is somehow random, but when there is a longer chain of components in series the delay variations even each other out (Sangwon et al., 2012, p. 981). It is reported that the delay variations are significantly smaller in 50 inverter series than in 20 inverter series. This is because the combined delay variation of many inverters approaches the mean value of the variations. Thus, the propagation delay of a pipeline or other logic path follows the equation

$$\lim_{n \to \infty} \sum_{i=1}^{n} (\bar{t} + \Delta t_i) = n \cdot \bar{t}, \tag{3.7}$$

where n is the length of the pipeline, or the number of devices in series inside a pipeline stage, $\bar{t}$ is the average propagation delay of a single device, Δt_i is the amount that a single device propagation delay variates from the average. This means that the delay variations of a longer pipeline are more predictable than the delays of a shorter pipeline. Thus, the

propagation delay variability can be mitigated by using longer pipeline stages. (Mingoo et al., 2011, p. 45)

3.7 Six-Transistor Static Random-Access Memory in Near-Threshold Computing

SRAM memory structures can be produced with a large variety of manufacturing processes and they have fast access times. Therefore, they are a widely used technique when implementing embedded memory. The six-transistor SRAM (6T SRAM) behavior relies on the relative sizes of the transistors in the cell. The manufacturing process produces transistors that have some mismatch in the sizes and therefore also in the characteristics. In the worst-case, these mismatches may make a 6T SRAM cell behave incorrectly. The effects that the mismatches have on the characteristics are greater with smaller transistors and lower V_{dd} values. (Chang et al., 2010, pp. 221–222)

The 6T SRAM is a popular memory structure when implementing on-chip memory. Unfortunately, when lowering the supply voltage, the 6T SRAM devices suffer from logic errors more easily than other logic on the chip. The 6T SRAM structure is vulnerable to the manufacturing variations caused mainly by effects called the random dopant fluctuation and the line edge roughness. They cause mismatch in the transistor strengths inside the memory cell, and therefore the 6T SRAM cell might become more likely to settle at one state than the other. This makes the memory unreliable. Also, the usual large number of high density devices on a chip makes the 6T SRAM vulnerable. (Mingoo et al., 2011, p. 45; Dreslinski et al., 2010, p. 257)

When the minimum energy consumption is pursued, and the supply voltage is scaled down to achieve it, it is suggested that the 6T SRAM should operate at higher supply

voltage than the rest of the circuit (Hanson et al., 2006, p. 483). However, in some cases, it might not be justifiable to divide the circuit into different voltage islands.

Because of these reasons, the memory should get special attention when low-power circuits are designed. Many previous studies about low-power circuits have concentrated on the memory structures also because it is usually a power hungry part of a circuit. Although, the SRAM structures are sensitive to logical errors, their reliability can be improved by designing each individual transistor width of a memory cell carefully. Downside of this is, that all the transistors in the SRAM cell can not be of minimal size. This adds to the circuit area.

3.8 Sub-Threshold Region

Using low supply voltages is not a new concept. Circuits that use sub-threshold voltages, and depend on the weak-inversion in the semiconductor channel, have been introduced already in the 1970s (Vittoz & Fellrath, 1977; Tsividis, 2008). At first the markets of the sub-threshold region techniques were small. Only devices like wristwatches and hearing aids were using it. Generally, the first use cases of the sub-threshold voltages were analog circuits. Nowadays they are used for digital circuits as well. Circuits have been achieved to work even with voltages under 0.2 V. (Dreslinski et al., 2010, pp. 254–255)

For many circuits, the minimum energy consumption is achieved in the sub-threshold regime, and nowadays the sub-threshold designs have become an attractive solution to energy consumption problems (Hanson et al., 2006, p. 469). Although, the serious drawbacks in the performance and the reliability are still holding back the wide usage of the sub-threshold computing (Dreslinski et al., 2010, p. 253).

The Near-Threshold Computing is balancing between the lowest power, slow speed and

unreliability of the sub-threshold designs and the fast and reliable use of the nominal voltage. The goal is to get significant power savings if compared to the nominal voltages, but avoid the significant shortcomings of performance and variability of the sub-threshold region. (Dreslinski et al., 2010, p. 253)

4 CASE STUDIES

A low-power and low performance device in mind it would be interesting to choose somewhat larger technology over the small high-end technologies. This is because the new small technologies are relatively expensive if compared with the larger ones. Also, if the manufacturing volumes are not large, the cost per unit might become expensive with new technologies. Larger technologies tend to have thicker insulating oxide layers between the *gate* terminal and the semiconductor channel, and therefore the leak currents are smaller. If the components of the device have low activity, or if the device stays idle long times, the small leak currents are beneficial. If it is also assumed that the performance need is modest, the speed of the device is not concerned the main demand. Under these circumstances, a 130 nm technology was chosen under examination. As a reference, in the year 2010 130 nm technologies were typically used in low-power applications, because the manufacturing costs were low and the leakage powers were low. Also embedded memory was possible to implement with it with a low cost. (Bol, Boyd, & Dornfeld, 2011)

The characteristics of the chosen technology when using Near-Threshold Computing voltages are examined. The results are intended as guidelines when implementing low-power devices with a 130 nm technology with the help of the Near-Threshold Computing technique. Before manufacturing Near-Threshold Computing devices, it should be studied if the chosen technology really is suitable for the Near-Threshold Computing voltages. The downsides, and the costs of the countermeasures against them, should be evaluated. The

overall energy consumption, when all the countermeasures are taken into count, should be smaller than in the device that is working with nominal voltage.

The components used in this study are chosen to be the low leakage variants. Low leakage components are a good choice, because the leak currents through them are small. On the other hand, the speeds of the components are low. This is acceptable, because in this study it is presumed that there is no need for high speed operations.

In some applications, some parts of a circuit could be implemented with faster devices. This would be beneficial if the particular part of a circuit had high activity. This means that the CMOS devices in that part are going through massive amounts of state changes in comparison to other parts of the circuit. Then faster transistors would reduce the transition times, and therefore also the time that a CMOS device is short circuited between V_{dd} and ground. (Bol, J. De V., et al., 2013, p. 22)

In this study two kinds of devices are simulated. Firstly, in the Chapter 5 a fanout-of-four (FO4) inverter is simulated. It is thought to represent a logic element in logic design. The FO4 inverter is used in several other studies to find out characteristics of technologies or design principles. The propagation delay of an FO4 inverter is concerned as metric for the speed of the process technology. When CMOS devices are manufactured in different technologies, and they are run in different temperatures and with different voltages, the propagation delays could be normalized if divided by the propagation delay of FO4 inverter. Normalizing helps in evaluating and comparing the performance of a device at hand. (Bo et al., 2005; Dreslinski et al., 2010; Harris et al., 2010; Ho, Mai, & Horowitz, 2001; Nam et al., 2005, p. 535)

Secondly, in the Chapter 6 a 6T SRAM cell is simulated and studied. This choice is made because the memory is often one of the most energy hungry parts of a circuit, and the 6T SRAM is perhaps the most widely used memory structure.

Table 4.1: Specifications of the used technology (Circuits Multi-Projects, 2010)

TECHNOLOGY:	CMOS 130nm (HCMOS9GP)
Spec. process char.:	Gate length: 130nm (drawn), 130nm (effective) Triple well Power supply 1.2V Multiple Vt transistor offering (Low Leakage , High Speed) Threshold voltages (for 2 families above): VTN = 450/340mV, VTP = 395/300mV Isat (for 2 families above): TN @ 1.2V: 535/670uA/mic; TP @ 1.2V: 240/310uA/mic 6 metal layers in standard Low k inter-level dielectric MIM capacitances 2.5V-transistors option is also available WARNING: the 3.3V-transistors option and the Ultra Low Leakage option are no longer available

In the case studies the following characteristics are measured: leak currents, propagation delays, energy usage and noise tolerance. In the 6T SRAM, also the logic error probabilities are evaluated. These all are simulated with different V_{dd} values between the threshold voltage 0.45 V and the nominal voltage 1.2 V . Also different temperatures are used. The specifications of the used technology are presented in the Table 4.1.

The reported values of the threshold voltage values for the low leakage variant of the CMOS technology are V_{nth} of 450 mV for NMOS and V_{pth} of 395 mV for PMOS. The nominal supply voltage is 1.2 V. To stay above both of the threshold values only voltages equal of above 450 mV are used as V_{dd} and the voltages that represent the logical 1. Also, the symbol V_{th} is used in this study as a combined symbol for the two threshold voltages.

4.1 Process Corners

CMOS manufacturers evaluate the process variations for each transistor type they produce. The behavior and the speed of a single transistor depend on various parameters, which differ between the transistors. A single transistor can have the slowest, fastest or the typical speed within some probability. The parameters for these transistors are specified as corner values S, F and TYP respectively.

In the circuit simulations, corner values can be utilized when worst-case scenarios are studied. The worst-case corners for each transistor type might already be known or they can be found in preliminary simulations. By using the worst-case corners the circuit is more likely to malfunction, and so the weak points of the device are easier to distinguish.

Throughout this study, the corner values are denoted with two corner values, the first for the NMOS and the second for the PMOS transistors. FS for example means that the NMOS transistors have the fast corners and the PMOS transistors have the slow corners. TYP denotes that both of them have typical corners.

4.2 Monte Carlo Simulations

The case studies are based on the Monte Carlo simulation technique. It is based on the process variations the chip manufacturer provides. When each device is generated in the simulator, each measurement of it is randomly generated based on the provided variations. Therefore, every time the same component is generated in the simulator, it does not have exactly the same dimensions and characteristics. The differences in components in the simulator model the variations in the actual physical manufacturing process. In the Monte Carlo simulations, the device that is tested is generated multiple times, and the test bench sequence is run with each of them. The goal is to get data that represents real manufacturing process products. The results can be used to make statistical assumptions of the behavior of the device.

5 CASE STUDY: FO4 INVERTER

In this study one single inverter is simulated to find out the characteristics of the used technology, especially when using Near-Threshold Computing. Before the simulations it was expected that a single inverter works logically in a normal manner; just the propagation delay was expected to grow and variate more in the Near-Threshold Computing voltages.

In this case study, the basic test bench is constructed from a chain of three inverters in series. In the middle is the inverter that is tested, and it is called the device under test (DUT). The first inverter acts as the driver for the DUT. The driver takes an ideal input signal from the simulator environment, and produces a signal that is more realistic. This signal is fed to the DUT as its input. The third inverter acts as load for the DUT. There are also additional inverters as artificial loads for the driver and the DUT. This test arrangement is illustrated in the Figure 5.1. The inverters used in this study are the low leakage variants of the fanout-of-four (FO4) inverter in the component library.

Monte Carlo simulations are conducted to gather knowledge about the variation of circuit behavior that is caused by the manufacturing process variations in the physical device. In most cases 1 000 Monte Carlo simulations are conducted. This number should give good estimates about the behavior of the device. Different corner parameter values are used to get knowledge about the worst-case process parameters. This knowledge is beneficial

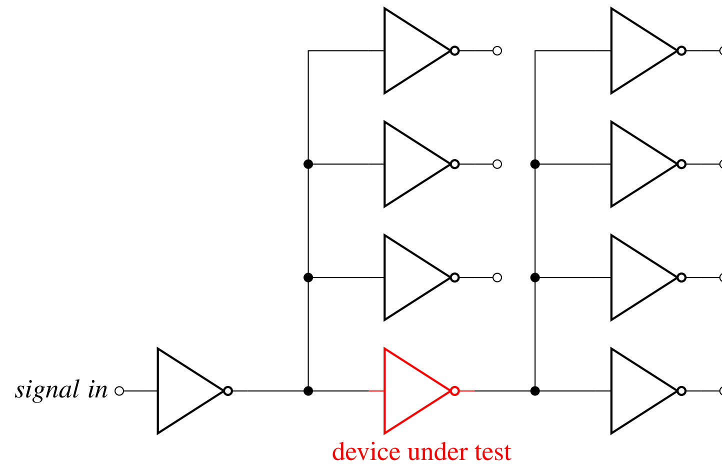
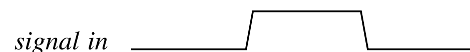


Figure 5.1: FO4 inverter test arrangement

when conducting further studies about the same technology. Different voltages between V_{th} 0.45 V and nominal voltage 1.2 V are used as the supply voltage V_{dd} and the logical 1 in the signals. Different temperatures between −25 °C and 125 °C are also used to gain knowledge about the influence of the environment.

The test sequence is as follows. The first column is the time in seconds. The second column represents the logical input to the inverter that acts as a driver; 0 is the ground voltage and 1 is V_{dd}.

```
time (s)        signal in
0               0
0.0100e-6       0
0.0101e-6       1
0.0200e-6       1
0.0201e-6       0
```



5.1 Propagation Delay

The propagation delay of the inverter is measured from the moment the input has changed 50 % of the voltage swing to the moment that the output also changes 50 % of the voltage swing. The voltage swing is the voltage difference between the used V_{dd} and the ground,

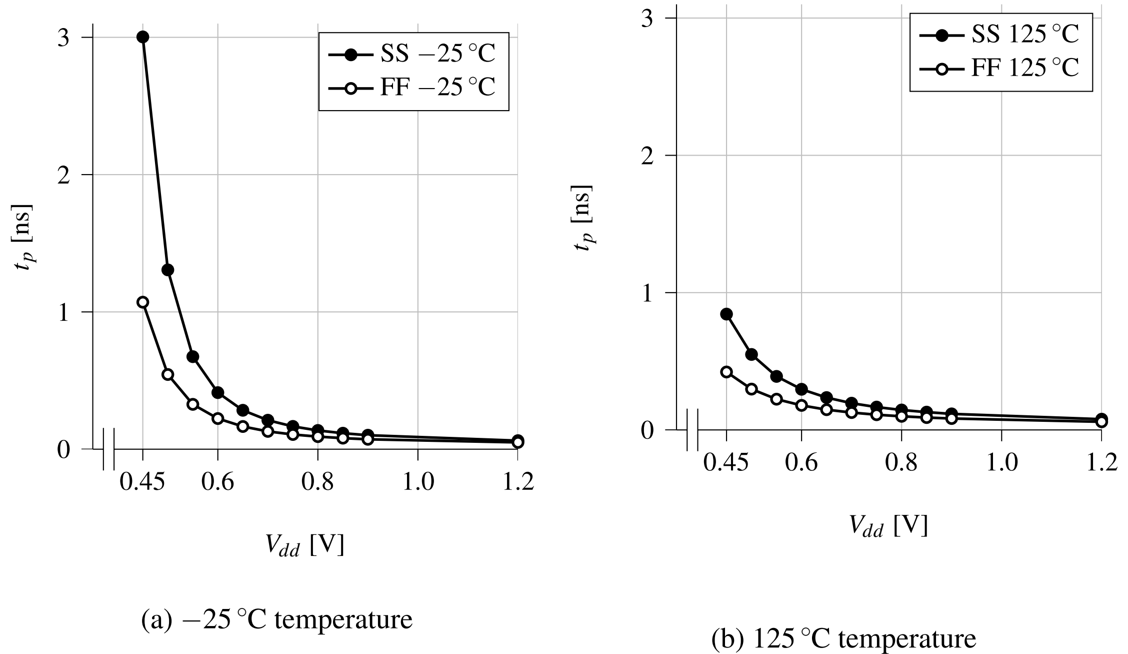
or the logical 1 and the logical 0. This is the conventional method for measuring the propagation delay.

It might be appropriate to think again if this is the right way to measure the low-voltage devices, while the output port voltages may sometimes go over the 50 % boundary more than once before settling at the final value. This type of event could take place if there is disturbance or noise in the signal lines. In the basic simulations of this study these are not observed. When the noise tolerance is measured in the Section 5.6 one solution to this problem is used.

Following are the results of the propagation delay simulations of the FO4 inverter. The results in the Figure 5.2 show how the propagation delay t_p grows when V_{dd} is in the Near-Threshold Computing region. The corners SS and FF are chosen here because they represent the two extremes of an inverter behavior. SS corners produce the slowest, and FF corners produce the fastest inverter. The differences can be seen in the simulation results in the Figure 5.2. This is quite obvious, but these simulations confirm that the worst-case scenarios when observing the t_p can be arranged using the SS corners.

The t_p is less than 2.2 times longer with V_{dd} of 0.80 V, and less than 6.7 times longer with V_{dd} of 0.60 V. The t_p starts to grow significantly when V_{dd} is lower than 0.6 V. This gives an indication, that speed penalties are quite manageable, if the V_{dd} stays above 0.6 V value. Notable is, that the t_p stays almost at the same level when V_{dd} is dropped from 1.2 V to 0.8 V.

When the Figure 5.2(a) and the Figure 5.2(b) are compared, it seems that the temperature of the device has a significant effect on t_p only if V_{dd} is under 0.6 V. This indicates that the inverter tolerates differences in the temperature well, but only if V_{dd} is not too low. These results can be seen in the Figure 5.3(a) and the Figure 5.3(b) more clearly. In these figures the steep rise of t_p when the temperature reduces, can be seen when V_{dd} is lowered towards V_{th}. It is notable, that the rise does not take place until under 0.6 V. This indicates



(a) −25 °C temperature

(b) 125 °C temperature

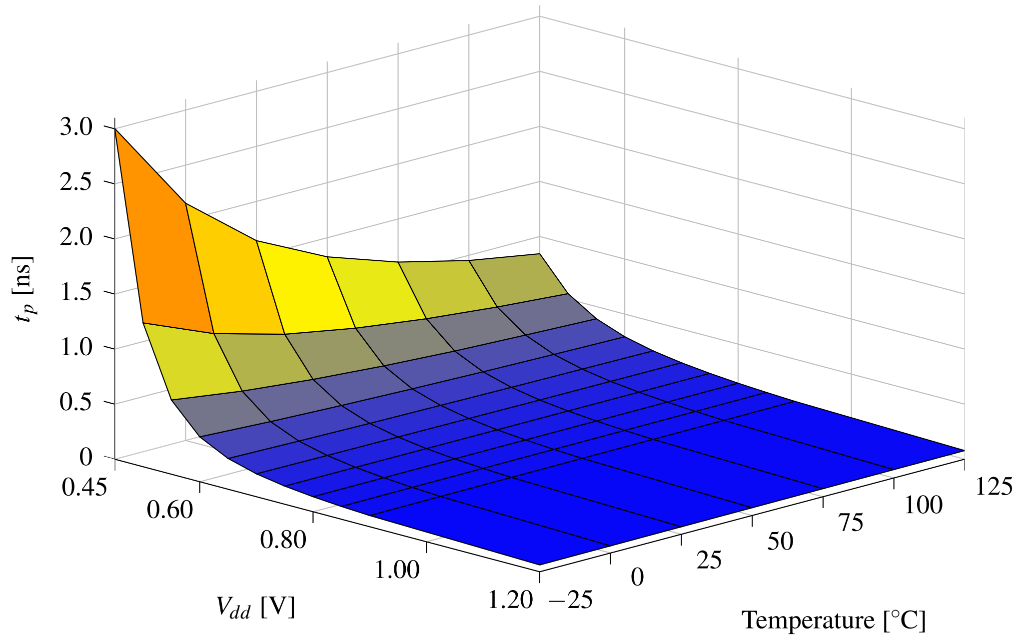
Figure 5.2: FO4 inverter propagation delays, maximum values, 1 000 Monte Carlo simulations

that the V_{dd} could safely be lowered significantly under the nominal 1.2 V.

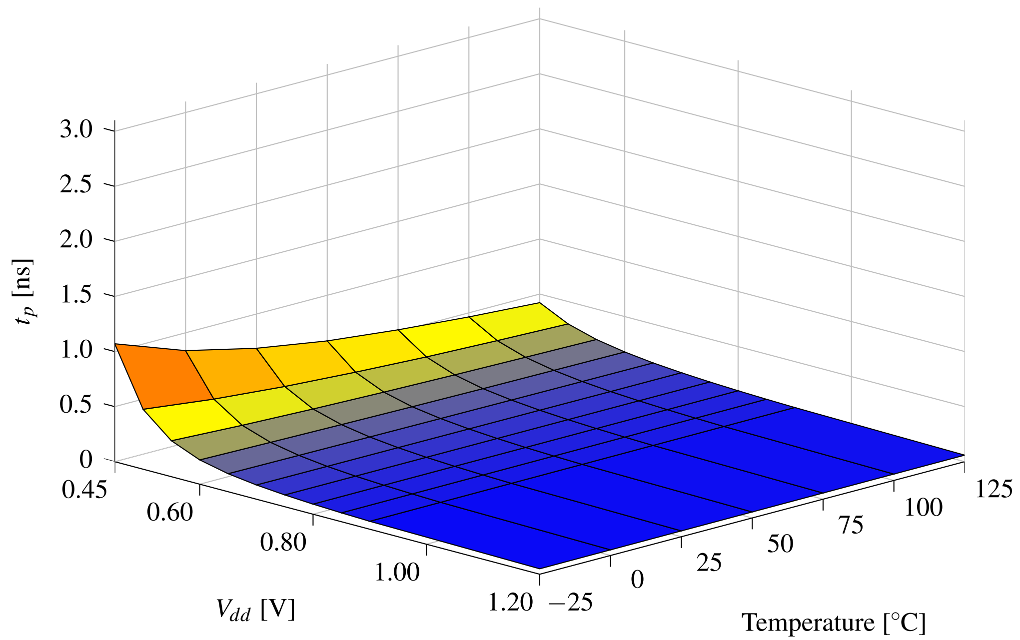
The results are showing also that the inverter runs faster at higher temperatures, when the Near-Threshold Computing voltage region is used. This phenomenon is discussed in the Section 3.5. (Yu et al., 2010, p. 628)

5.2 Propagation Delay Variation

In the Figure 5.2 and the Figure 5.3 the values of the propagation delays are the maximum values from 1 000 Monte Carlo simulations. The mean, standard deviation, minimum and maximum values are presented in the Figure 5.4. The deviations seem to be normally distributed. Noticeable is, that the propagation delay is variating more, when V_{dd} is lower. There is also a $5 \cdot \sigma$ line presented. That represents the barrier which statistically three values out of 1 000 million go over.

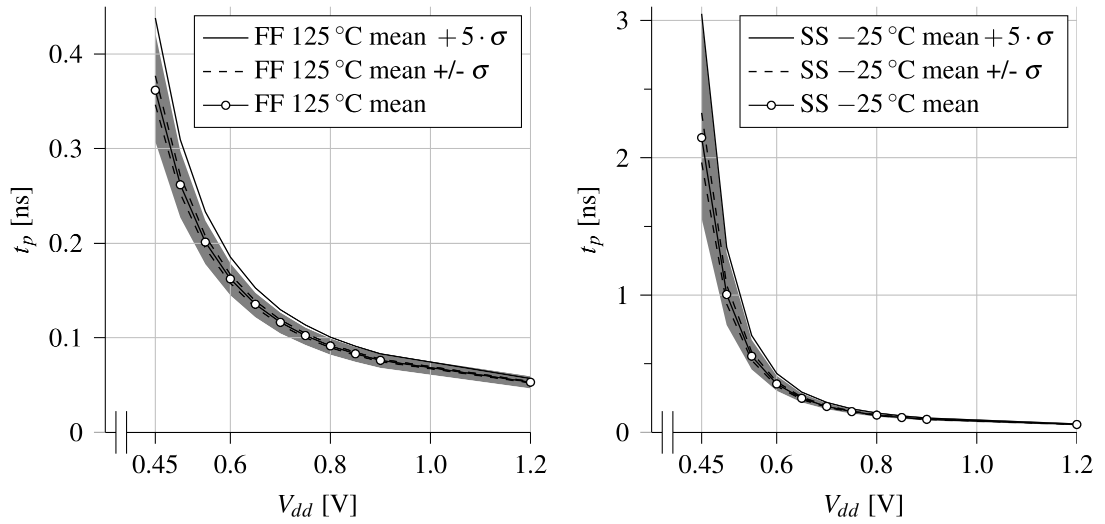


(a) SS corners



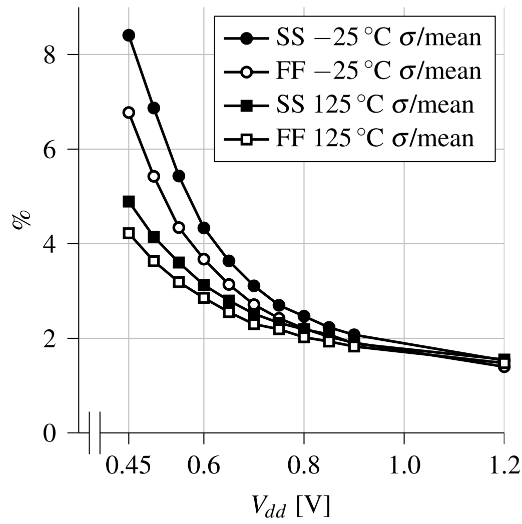
(b) FF corners

Figure 5.3: FO4 inverter propagation delays, maximum values, 1 000 Monte Carlo simulations



(a) The best case parameters, the gray area is the values between the minimum and the maximum

(b) The worst-case parameters, the gray area is the values between the minimum and the maximum



(c) The standard deviation compared with the mean value of the propagation delay

Figure 5.4: FO4 inverter propagation delay variations, note the different y axis scales

In the Figure 5.5(a) it seems that the delays are distributed symmetrically. This implicates to normal distribution. Also, the Figure 5.5(b) shows the delay to be distributed close to the Gaussian curve.

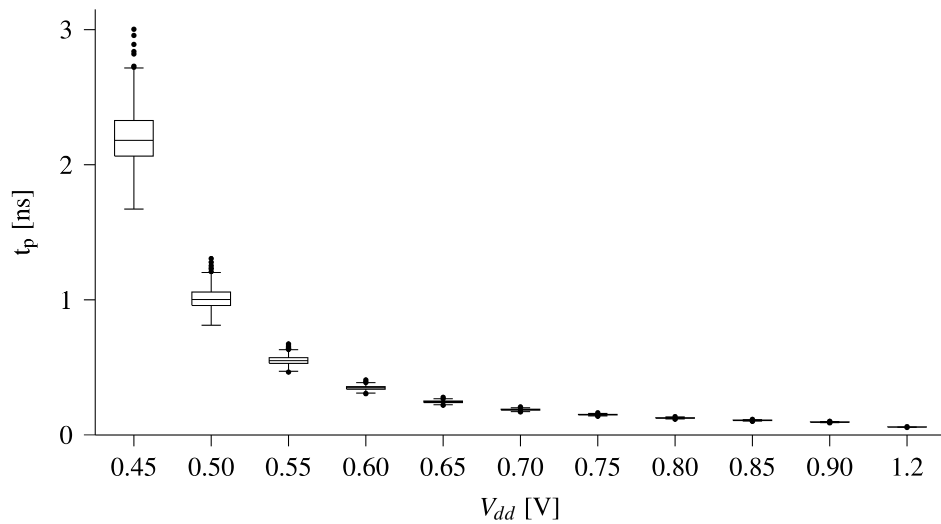
The variations proportional to the mean values are illustrated in the Figure 5.4(c). It is clear that the proportional variation gets bigger when V_{dd} gets lower. Also, the variation gets bigger as the magnitude of the propagation delay gets bigger. This can be seen when comparing the Figure 5.2(a) and the Figure 5.2(b) with the Figure 5.4(c).

5.3 Energy Consumption

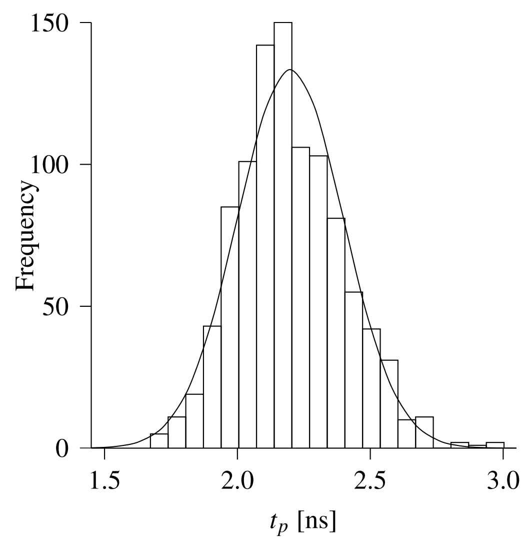
The *source-drain* current of the PMOS transistor is the most significant place of current flow whenever the inverter changes the *output* from the logical 0 to the logical 1. That is when the load capacitance gets charged from the voltage source through the PMOS transistor. The energy consumption during the *output* change from logical 0 to logical 1 was determined by measuring the current through the *source* terminal node of the PMOS transistor.

The *drain-source* current of the NMOS transistor is the most significant place of current flow whenever the inverter changes the *output* from the logical 1 to the logical 0. That is when the load capacitance gets discharged to the ground through the NMOS transistor. The energy consumption during the *output* change from logical 1 to logical 0 was determined by measuring the current through the *drain* terminal node of the NMOS transistor.

The currents were measured during a 30 ns period starting from the beginning of the state change event. During this period, the state transition has time to be completed, even with V_{dd} of 0.45 V. The measured current was integrated over the time period and the result was multiplied by the V_{dd}. The Equation 5.1 shows the calculation of the state transition

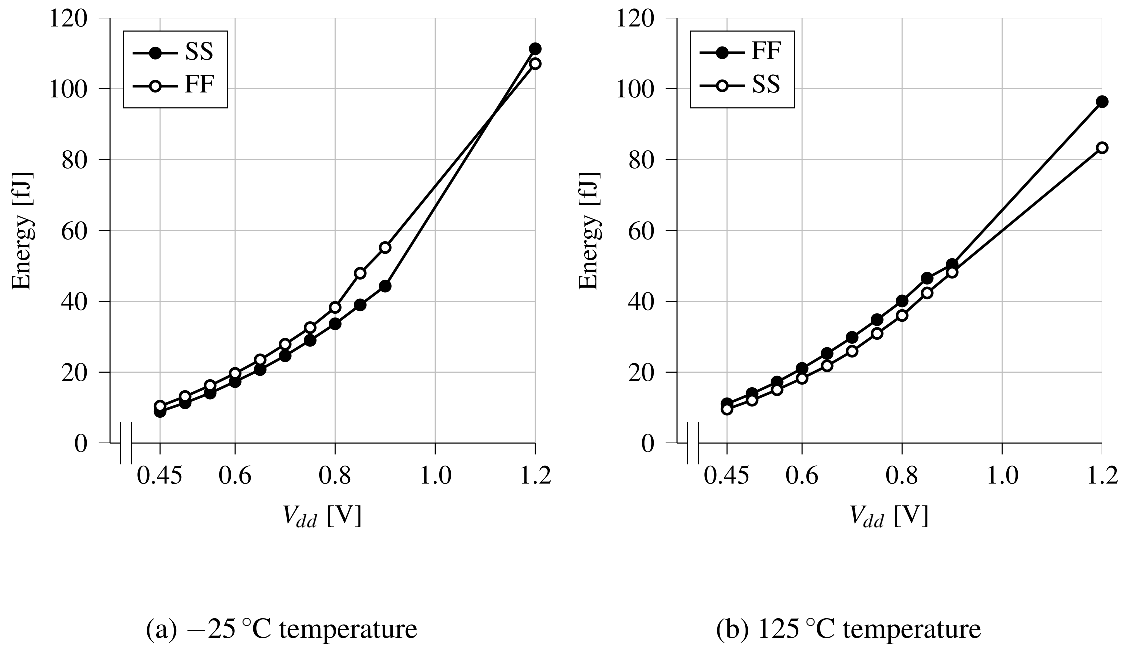


(a) The distributions



(b) The distribution at V_{dd} of 0.45 V, mean value: 2.1992 ns, σ: 0.1994 ns

Figure 5.5: FO4 inverter propagation delay variation distributions, 1000 Monte Carlo Simulations, SS corners , −25 °C temperature



(a) −25 °C temperature

(b) 125 °C temperature

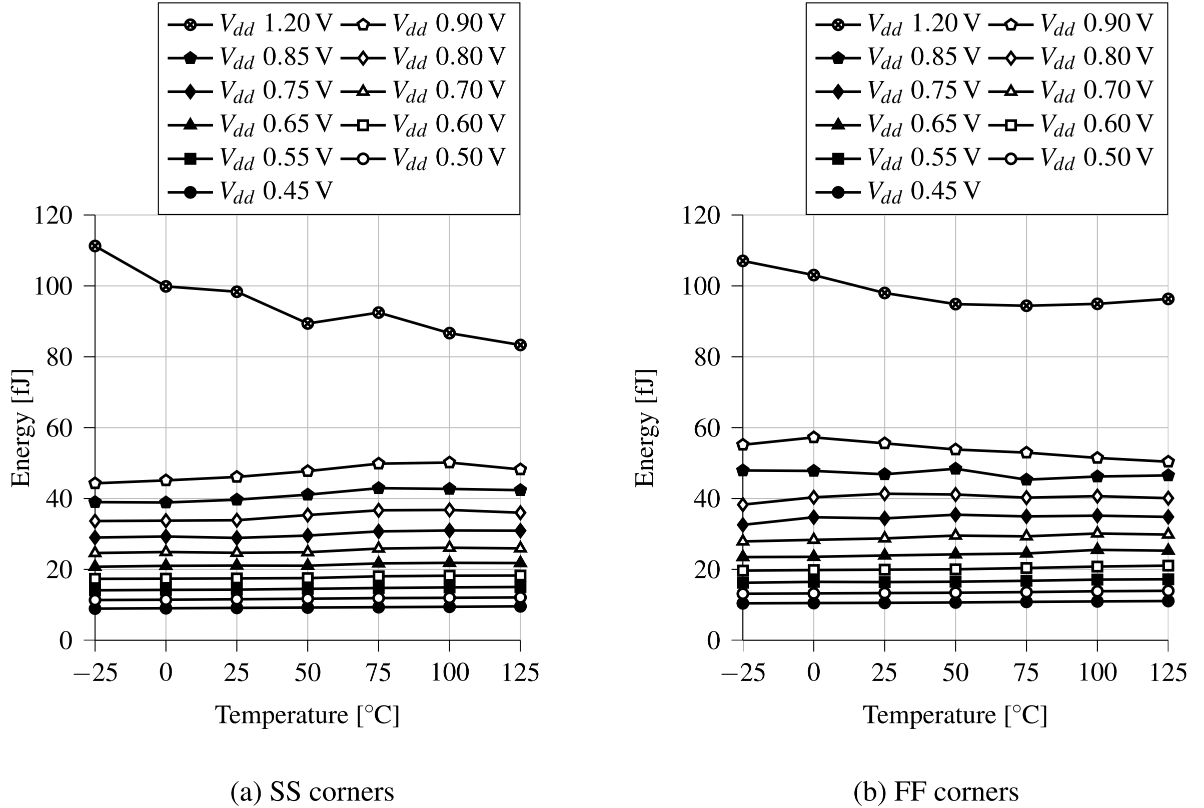
Figure 5.6: FO4 inverter energy per operation mean values

where the output rises from logical 0 to logical 1. The Equation 5.2 shows the calculation of the state transition where the output falls from logical 1 to logical 0.

$$E_{rise} \approx V_{dd} \int_{t_1}^{t_2} I_{source_{PMOS}} \tag{5.1}$$

$$E_{fall} \approx V_{dd} \int_{t_1}^{t_2} I_{drain_{NMOS}} \tag{5.2}$$

The energy consumption is reducing when V_{dd} is lowered. This is illustrated in the result figures of the calculations in the Figure 5.6. The drop in the energy consumption is significant. It drops to under 23 % when the V_{dd} reduces from 1.2 V to 0.8 V, and to under 45 % when the V_{dd} reduces from 1.2 V to 0.6 V. When combined this result with the propagation delay results in the Section 5.1 it seems a good idea to lower V_{dd} at least to somewhere around 0.7–0.8 V. There is almost no speed penalty and the energy consumption is half of the nominal usage.



(a) SS corners

(b) FF corners

Figure 5.7: FO4 inverter energy per operation mean values

The corner parameters do not seem to have significant effect on the energy consumption. Although there seems to be a slight difference to the direction that FF corner parameters are the worst-case when energy consumption is considered.

Also, the temperature does not seem to have a significant effect on the energy consumption. The changes in the energy consumption are illustrated in the Figure 5.7, where all the lines drawn between the energy measurement points are somewhat horizontal. The decrease in temperature does not seem to have almost any effect on the energy consumption. This might be due to two phenomena combined. Firstly, the propagation delays increase, when the temperature decreases. This alone would increase the energy consumption. Secondly, the leak currents decrease when the temperature decreases. The simulation results of the leak currents are presented in the Section 5.4. This alone would in turn decrease the energy consumption, when the temperature decreases.

5.4 Leak Current

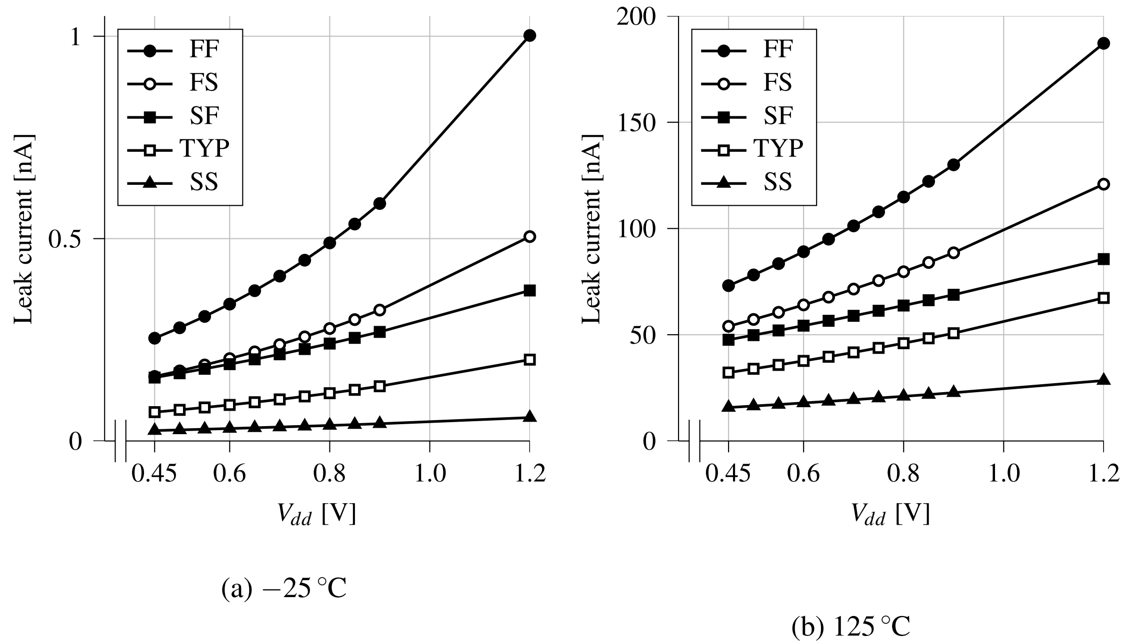
Whenever there are voltage differences across some parts of the circuit there are also leak currents present. The currents in the CMOS transistors across the terminals are observed in the simulations. Whenever the inverter component is in a static state almost all of the leak current flows from the *source* to the *drain* in the PMOS and from the *drain* to the *source* in the NMOS. This is observed by measuring currents through each of the terminals. The current through the *gate* is not significant if compared with others.

To measure leak currents through the inverter a slower test sequence is used. After a state change a long waiting period of 200 µs is added. This is more than enough for the signals to stabilize, and at the end of this waiting period the currents that are present are considered as the leak currents.

When V_{dd} is lower, it is presumable that also the leakage is lower. This is because the current through a component is proportional to the voltage over it. This relation can be seen in the Figure 5.8 and the Figure 5.9. The FF corner parameters seem to be the worst-case parameters when the leakage is concerned. This is intuitive, while a fast transistor is more sensitive to a deliberate closing, also the unintentional leak current flows more easily through it than through a slow transistor.

When comparing the y-axis of the Figure 5.8(a) and the Figure 5.8(b) it can be seen that the leakage is much bigger in the temperature 125 °C than in the -25 °C. This phenomenon is presented in the Figure 5.9 as well.

Overall, the results indicate, that lowering the V_{dd} from the nominal 1.2 V to somewhere under 0.8 V decreases leak currents to half.



(a) −25 °C

(b) 125 °C

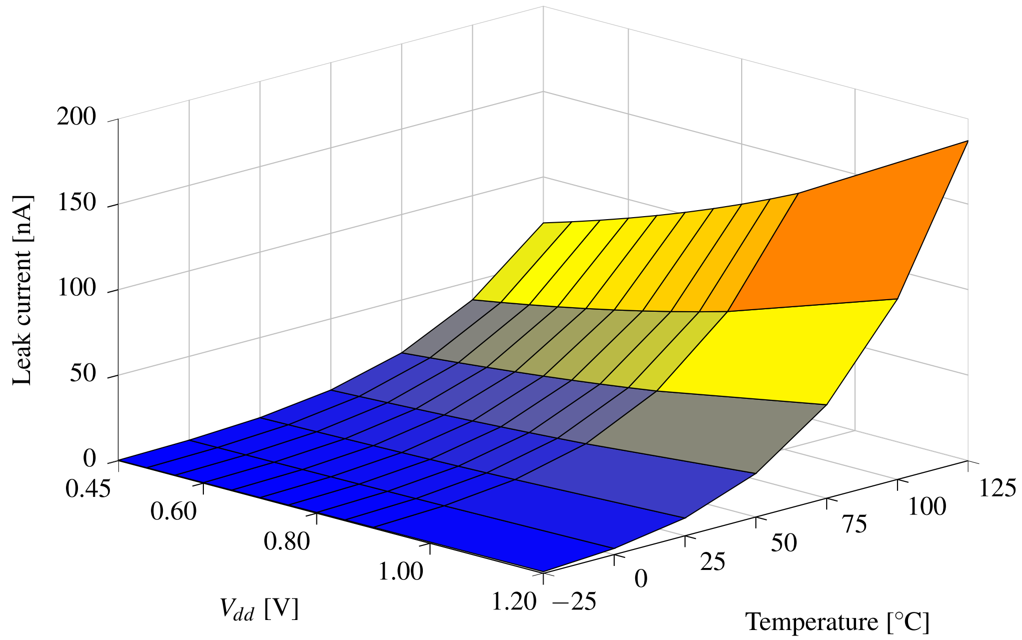
Figure 5.8: FO4 inverter leak current maximum values, 1 000 Monte Carlo simulations

5.5 Leak Energy Per Operation

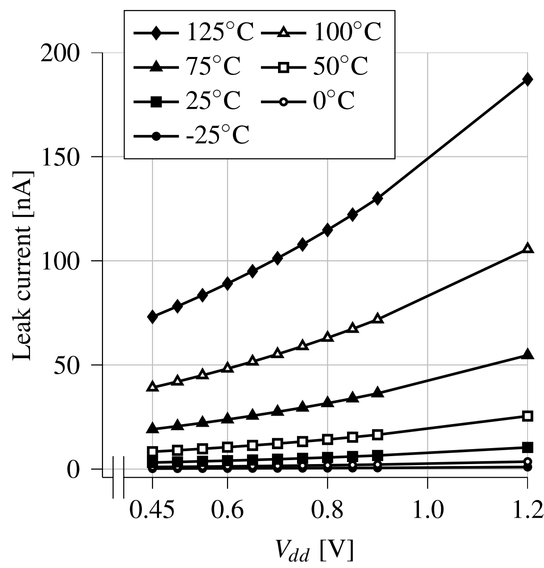
The portion of energy consumption, which is due to the leak currents, can be calculated by multiplying the leak power by the propagation delay of a single operation. The leak power is the leak current multiplied by V_{dd}. The results of these calculations with the leakage worst-case corner parameters of FF are illustrated in the Figure 5.10.

When observing the leak energy per operation, there is a minimum in between supply voltages 0.6 V and 0.8 V. This implicates, that V_{dd} in this region is beneficial if the goal is to minimize the leak energy consumption.

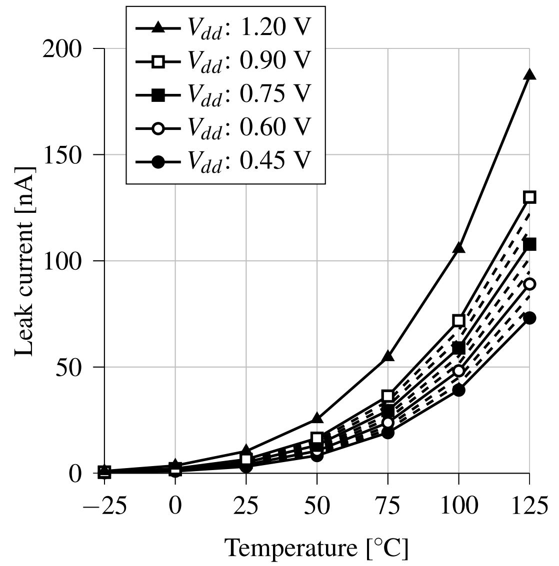
The magnitude of the energy consumption of the FO4 inverter is 10 to 100 fJ and the portion caused by the leakage is under 10 aJ. The leak energy is seemingly very small if compared with the total energy consumption. Nevertheless, if the circuit is such, that it stays long times in the idle state and if it has to stay operational autonomously long



(a) 3D presentation of the leak currents

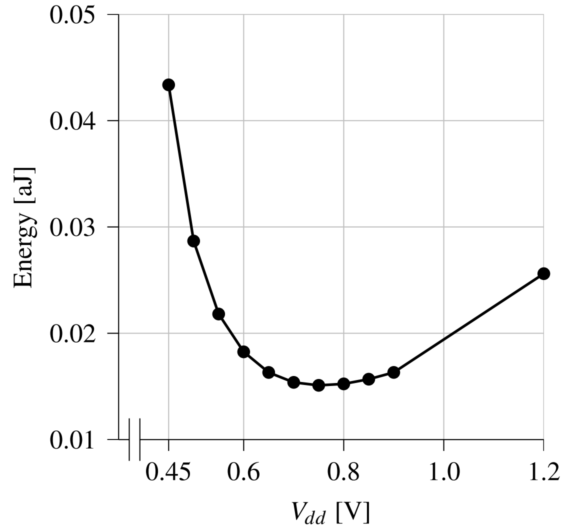


(b) V_{dd} changes

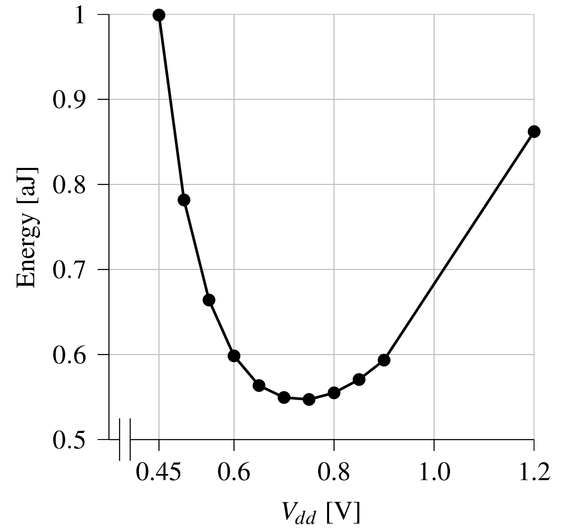


(c) Temperature changes

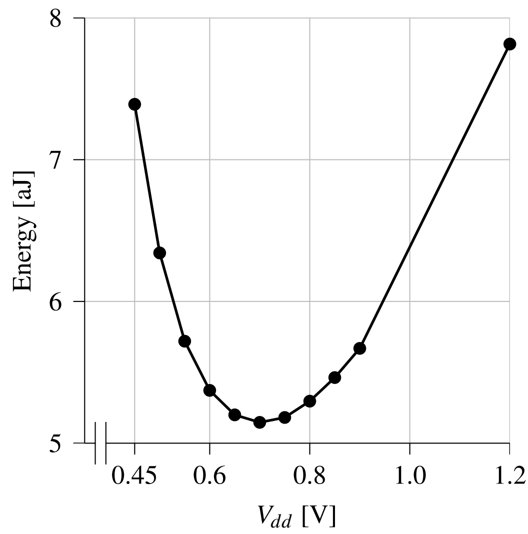
Figure 5.9: FO4 inverter leak current maximum values,, 1 000 Monte Carlo simulations, the worst-case FF corners



(a) −25 °C temperature



(b) 50 °C temperature



(c) 125 °C temperature

Figure 5.10: FO4 inverter leak energy per operation, 1 000 Monte Carlo simulations, the worst-case FF corners, note the different y axis

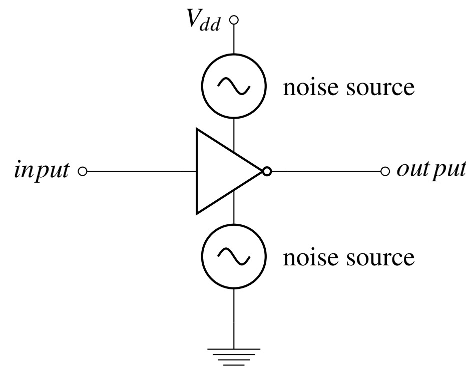


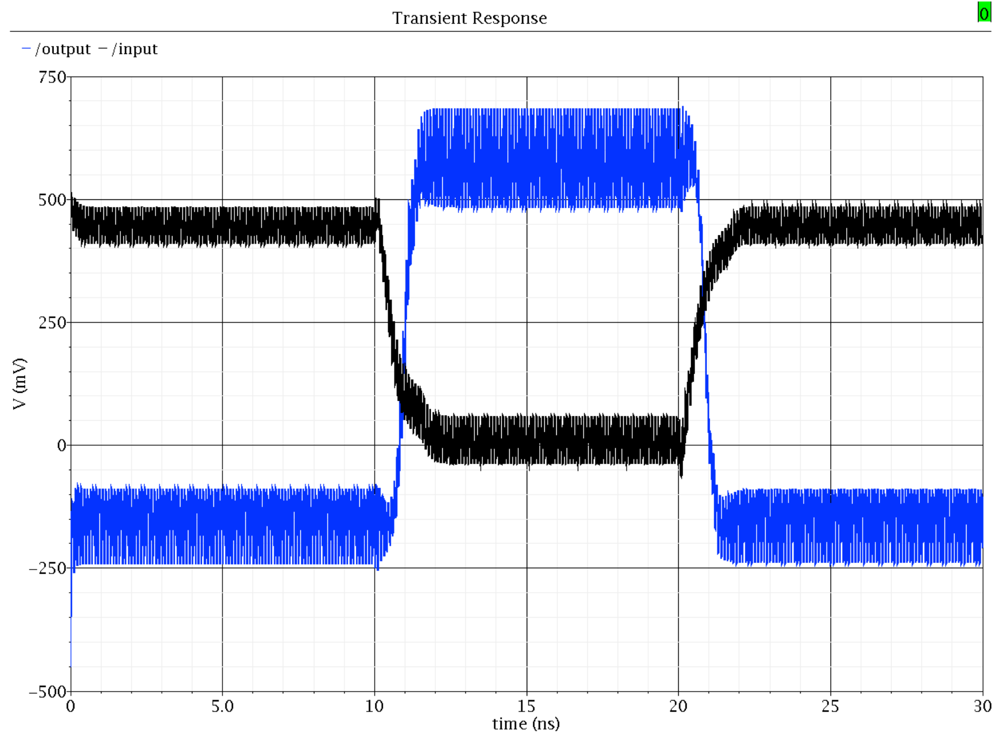
Figure 5.11: FO4 inverter noise tolerance test diagram

times without additional energy source, the leak energy has to be considered an important hazard in circuit designs.

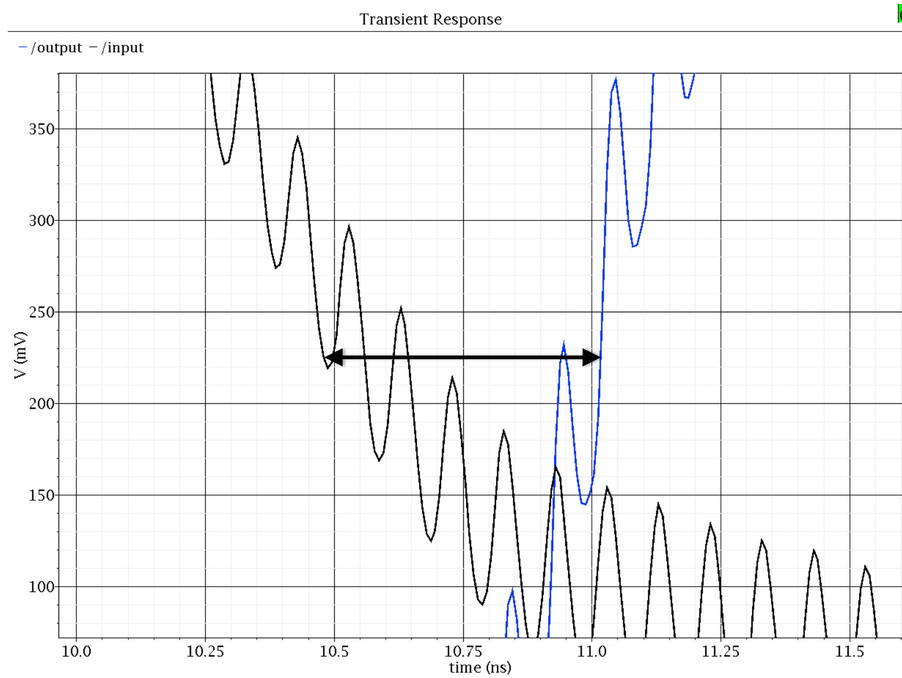
5.6 Noise Tolerance Of the Inveter

For testing the dynamic noise tolerance two noise sources were placed to V_{dd} and the ground ports of the inverter. This arrangement is illustrated in the Figure 5.11. This construction is then placed in the same test arrangement as in the Figure 5.1. This arrangement simulates a situation, where a FO4 inverter is a part of a logic path, where the devices are near each other in series. Therefore there is only little noise in the *input* and *output*. When considering the propagation delay, the worst-case temperature in the research done in the Section 5.1 is $-25\,^{\circ}\mathrm{C}$, and the worst-case corner parameter values are SS. These parameters are chosen for the noise simulation.

The noise sources both generate a sine wave with a frequency of 10 GHz and an amplitude of V_{dd}. The amplitude is exaggeratedly large to make the possible malfunctions stand out more clearly. The noise signal phases are variated independently. The purpose of this is to simulate two different independent noise sources. The propagation delays are calculated so that if the output of the DUT crosses multiple times the 50 % boundary of the voltage swing, the last one of the crossings is taken into account. This is illustrated in



(a) Noise tolerance test sequence



(b) Zoomed in on the section where the propagation delay measurement is done

Figure 5.12: FO4 inverter noise tolerance test, *input* and *output* signals

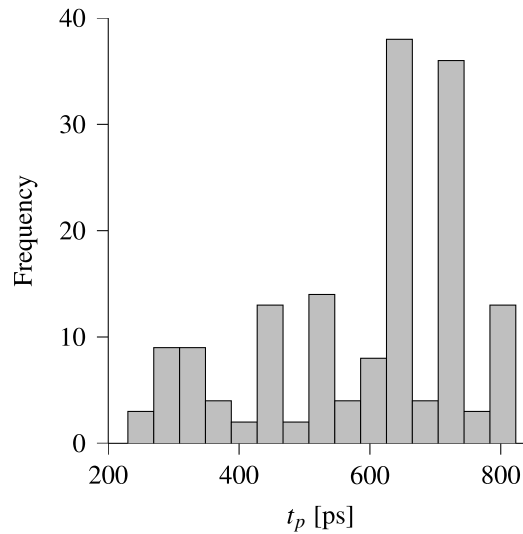


Figure 5.13: FO4 inverter propagation delays from noise tolerance test, 162 different test situations, SS cornes, −25 °C temperature

the Figure 5.12.

When the arrangement is simulated with 9 different initial phase values of each of the noise sources, there is 81 different test situations for each two state changes. The propagation delays of the tests are illustrated in the Figure 5.13. Monte Carlo technique was not applied to this test. The resulting delays seem to be smaller than those in the Section 5.1. When comparing these results with the Figure 5.5(b), the t_p values with the noise sources attached are under 1 ns, while the mean value with the same conditions but without the noise sources is 2.2 ns. The results indicate that the FO4 inverter has high tolerance to noise in V_{dd} and the ground voltage lines.

5.7 Ring Oscillator

The *de facto* standard of measuring and comparing circuit delays is the ring oscillator (Rabaey, 1996, p. 117). This technique is good for comparing circuits with others. However, it should not be used to actually determine the maximum frequency of the circuit.

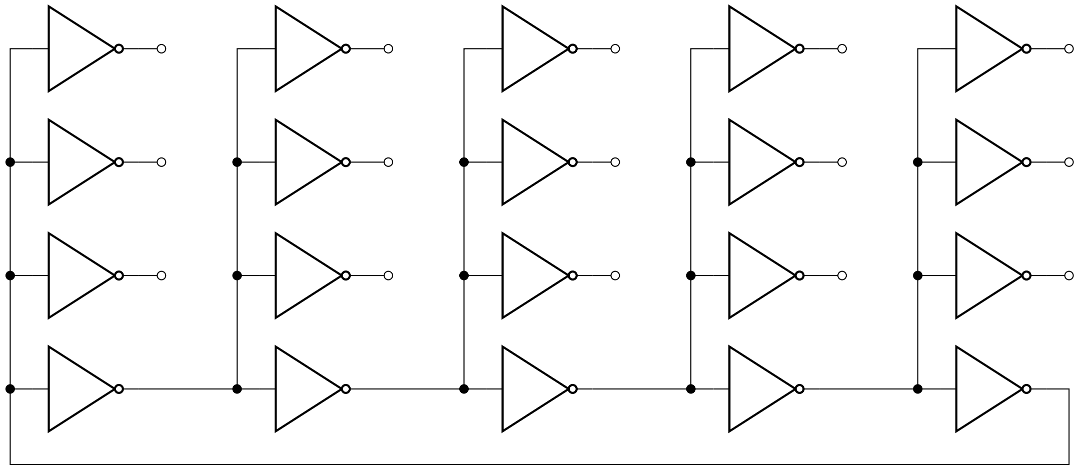


Figure 5.14: FO4 inverter ring oscillator test arrangement

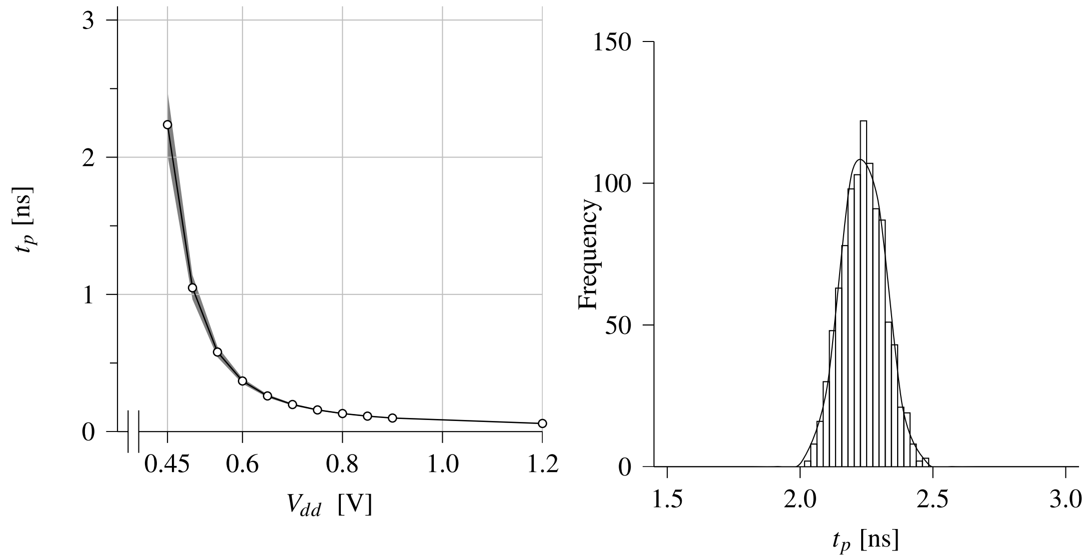
A ring oscillator is constructed from an odd number of inverters in series. In this experiment, a series of five inverters is used. Each inverter has a load of four inverters as illustrated in the Figure 5.14.

The voltages in the wires connecting the inverters oscillate. The theoretical maximum of the frequency is determined from the equation

$$f_{max} = 2 \cdot n \cdot f_{measured}, \tag{5.3}$$

where the n is the number of inverters in the series and the $f_{measured}$ is the frequency of the signal in one wire connecting two of the inverters. The measured frequency has to be multiplied by two times the number of inverters in the series, because during one period two events occur: the rising and the falling edges of the signal.

Even if the results of the ring oscillator simulations are not fully comparable with the results from other kind of test arrangement, some observations of their differences are interesting. When comparing the Figure 5.5(b) and the Figure 5.15(b), the mean values seem to be almost the same, but there are significant differences in the deviations. This is mainly due to the chain of multiple inverters in series. These five inverters have different



(a) The thin gray area representing all the values

(b) The distributions at 0.45 V V_{dd}, mean: 2.2373 ns, σ: 0.079843 ns

Figure 5.15: FO4 inverter propagation delays from the ring oscillator test, 1000 Monte Carlo runs, worst-case SS corners and −25 °C temperature

process variations, and they average each other out in the manner that is discussed in the Section 3.6. Longer pipelines seem to be a feasible way of compensating large variations between components.

6 Case Study: 6T SRAM

In many cases, the memory is a significant energy consumer on an IC chip and it would be beneficial to lower its energy needs. The purpose of this case study is to find out the behavior characteristics of an IC memory circuit when it is used with Near-Threshold Computing voltages. A single 6T SRAM memory cell is studied by simulating it with different temperatures and manufacturing parameters. Presumption based on the literature is that 6T SRAM becomes unreliable with low V_{dd} values. (Alorda et al., 2009; Bol, J. De V., et al., 2013; Dreslinski et al., 2010)

6.1 6T SRAM Test Arrangement

The test arrangement of the 6T SRAM is illustrated in the Figure 6.1. The test bench is constructed to resemble a 1 024-bit column of a 6T SRAM circuit. A memory column is illustrated in the Figure 2.7. The test bench construction is different from that in a way that there is only one 6T SRAM memory cell, and other cells are replaced with two capacitors connected to the bitlines. The locations of the capacitors can be seen in the Figure 6.1. The capacitors model the capacitance of 1 023 memory cells in parallel. The simulations run faster with this construction than with 1 024 SRAM cells. To determine the sizes of the capacitors, a memory cell *read* and *write* simulation is run with 1 024 SRAM cells and

with just the capacitors. By comparing the rise and the fall times in the bitline signals BL and $\overline{BL}$ between each simulation the sizes of the capacitors are adjusted. The size of each capacitor C_{load} in the Figure 6.1 is 185 fF.

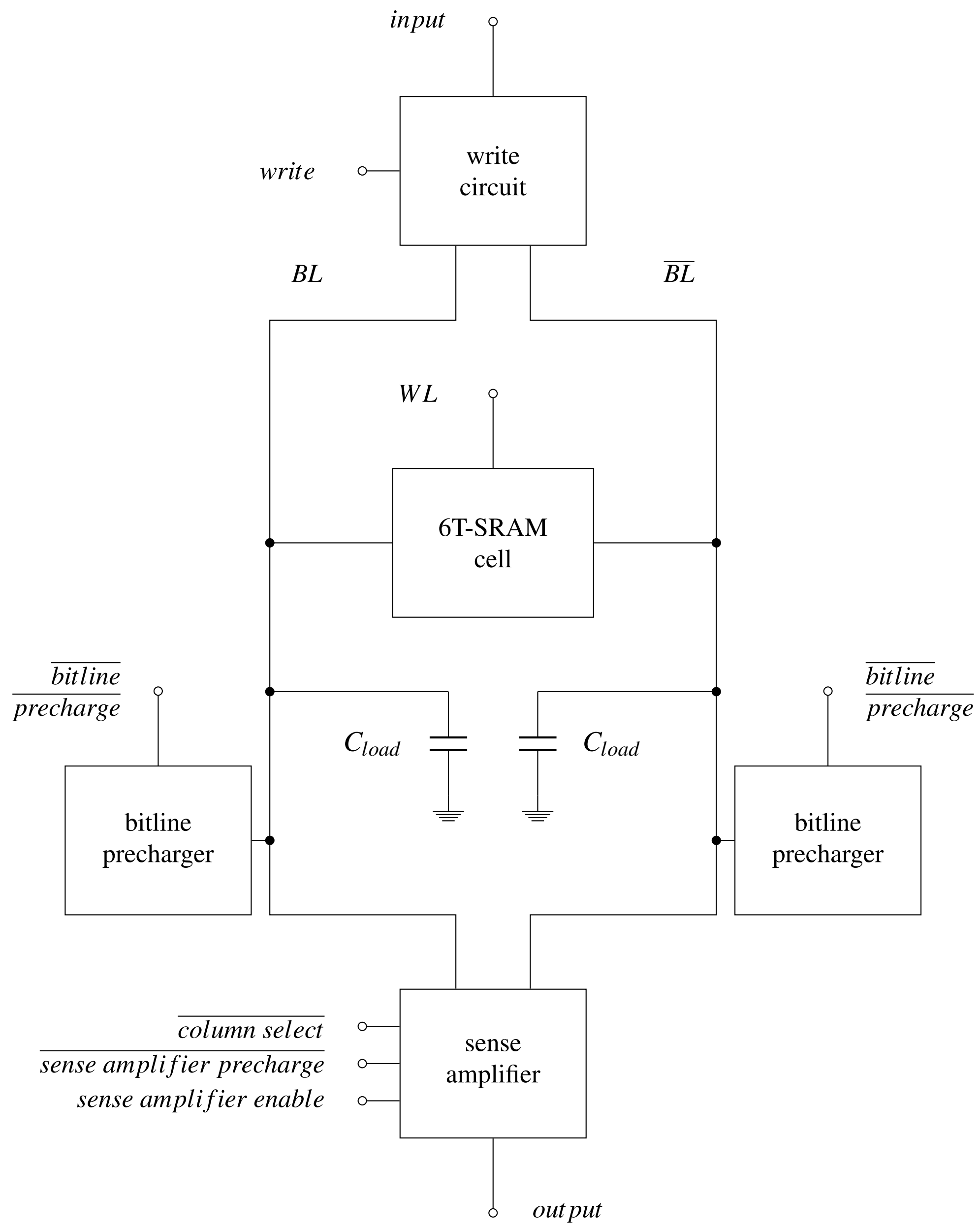


Figure 6.1: 6T SRAM test arrangement

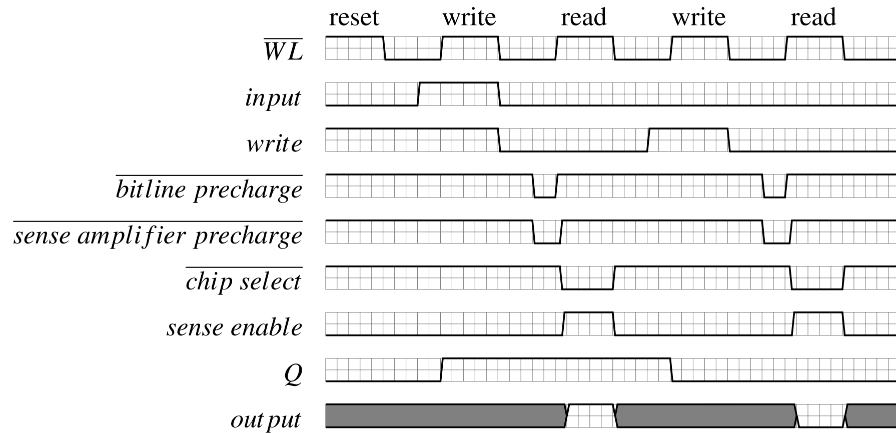


Figure 6.2: 6T SRAM test sequence, one square on the horizontal time axis represents 0.5 µs

6.1.1 Input Signals

The control signals are not taken straight from the simulator, but through inverters to make the signal transitions more realistic. The simulation sequence is designed as follows. At first, a reset is executed by writing 0 to the memory cell. At the time 5 µs value 1 is written to the memory cell. At the time 10 µs value 1 is read from the memory cell. At the time 15 µs value 0 is written to the memory cell. At the time 20 µs value 0 is read from the memory cell. The control signals as well as the assumed value stored in the memory cell Q are illustrated in the Figure 6.2.

6.2 Errors In 6T SRAM

The most important features of SRAM memory circuit are the ability to store given data, and keep it stored. When data is written to memory, the circuit should store it correctly. When data is read from the memory, the output should correspond to the stored data, and the stored data should not change during the *read* operation.

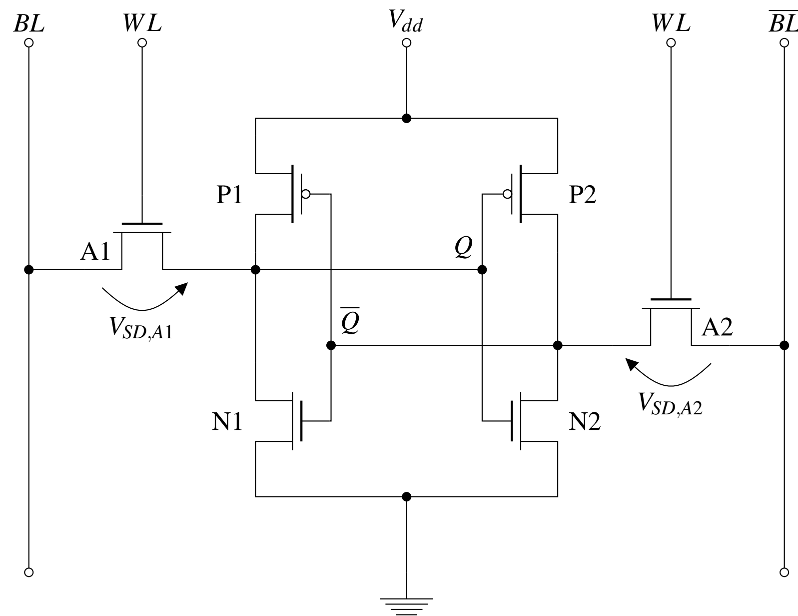


Figure 6.3: 6T SRAM transistor diagram

The reliability of 6T SRAM is tested by reading the voltage levels of nodes Q and $\overline{Q}$ after 1 µs after each *write* and *read* operation starts. The test sequence is presented in the Figure 6.2. Q and $\overline{Q}$ are illustrated in the Figure 6.3. Q represents the bit value that is being stored to the memory cell, and $\overline{Q}$ is the inversion of that. The behavior of the 6T SRAM is interpreted as logically erroneous, when the value stored in the memory cell is wrong after *read* or *write* operation. The errors of this kind happen when the wrong bit gets stored during *write* operation or when the state of the memory changes during a *read* operation.

6.2.1 Pre-charging of the Bitlines

The bitlines are pre-charged before each *read* operation to ensure that the previous remaining charges in the bitlines do not have an effect on the operation. This way, the conditions before each *read* operation are the same. The pre-charging to $\frac{V_{dd}}{2}$ would seem logical, as only equalization of the charges already in the bitlines, which have the potentials V_{dd} and 0, would be needed. Only little energy would be consumed in this operation

Table 6.1: Read error counts depending on the pre-charging value of the bitlines and the sense amplifier, 50 Monte Carlo simulations, TYP corners, 0 °C temperature

bitline pre-charge value	$\frac{V_{dd}}{2}$	$\frac{V_{dd}}{2}$	V_{dd}	V_{dd}
sense amplifier pre-charge value	$\frac{V_{dd}}{2}$	V_{dd}	$\frac{V_{dd}}{2}$	V_{dd}
read 1, SRAM cell switched	4	4	0	0
read 1, wrong value in the array output	15	10	0	0
read 0, SRAM cell switched	2	2	0	0
read 0, wrong value in the array output	13	9	0	0

because of no need for importing any charge to the bitlines. (Baker, 2010, p. 437)

Some testing is made with pre-charging the bitlines to $\frac{V_{dd}}{2}$ and V_{dd}. 50 runs of Monte Carlo simulations is run with each of them. The values stored in the memory cell are observed after the read operations. Observations are made after reading logical 0 and logical 1. The transistors inside the 6T SRAM cell are of minimum size. V_{dd} of 0.45 V was used as the operation near V_{th} is of interest. The behavior is interpreted as erroneous if the 6T SRAM does not keep its logical state throughout the read operation. Also, the *sense amplifier* is pre-charged before every *read* operation. Values of $\frac{V_{dd}}{2}$ and V_{dd} are used for it too to see if there is some affect to the read operation. The results are illustrated in the Table 6.1. The pre-charge value of $\frac{V_{dd}}{2}$ causes the memory cell to switch its state and the sense amplifier output to give wrong output values.

After pre-charging of the bitlines there is a voltage difference of V_{dd} over the other access transistor of the 6T SRAM cell. This voltage difference is labeled as $V_{SD,A1}$ or $V_{SD,A2}$ in the Figure 6.3. If there is 1 stored in the 6T SRAM, the $V_{SD,A1}$ is 0 and the $V_{SD,A2}$ is the same as V_{dd}. If there is 0 stored in the 6T SRAM, the $V_{SD,A1}$ is the same as V_{dd} and the $V_{SD,A2}$ is 0. So the voltage difference over the access transistor is on the side that has 0 stored at. Over the other access transistor there is no voltage difference, as there is potential of V_{dd} on both sides of it.

When WL signal is risen and the access transistors are closed, the access transistor that has 0 stored on its side lets the bitline discharge through it. The current flow is significant if there is a voltage difference greater than V_{nth} between the *source* and the *drain* nodes of the NMOS transistor. If V_{dd} is small, the current through the access transistor is also small. However if V_{dd} is smaller than V_{nth}, the bitline will not be able to discharge through the access transistor.

The relationship of the current through the access transistor I_{access} to V_{dd} is shown in the Equation 6.1 and Equation 6.2. If the bitlines are pre-charged to $\frac{V_{dd}}{2}$, and V_{dd} is smaller than two times V_{nth}, the voltage over the access transistors will not be over V_{nth}, and current can not flow through. This is why the bitlines must be pre-charged to full V_{dd} when V_{dd} is as low as in the Near-Threshold Computing. On these grounds, pre-charge value of V_{dd} is used throughout this study.

$$I_{access} \begin{cases} \sim 0 & \text{, when } V_{SD,access} < V_{nth} \\ > 0 & \text{, when } V_{SD,access} > V_{nth} \end{cases} \tag{6.1}$$

$$V_{SD,access} \begin{cases} < V_{nth} & \text{, when } V_{dd} < 2 \cdot V_{nth} \\ > V_{nth} & \text{, when } V_{dd} > 2 \cdot V_{nth} \end{cases} \tag{6.2}$$

6.2.2 Minimum Size 6T SRAM Cell Simulations

To make reasonable assumptions about the reliability of minimum size 6T SRAM memory large count of simulations are conducted. Up to 10 000 Monte Carlo simulation runs are run to find out the probabilities of logical errors occurring. Notable is, that every *write* operation of every simulation succeeded without errors. Therefore, all the error counts

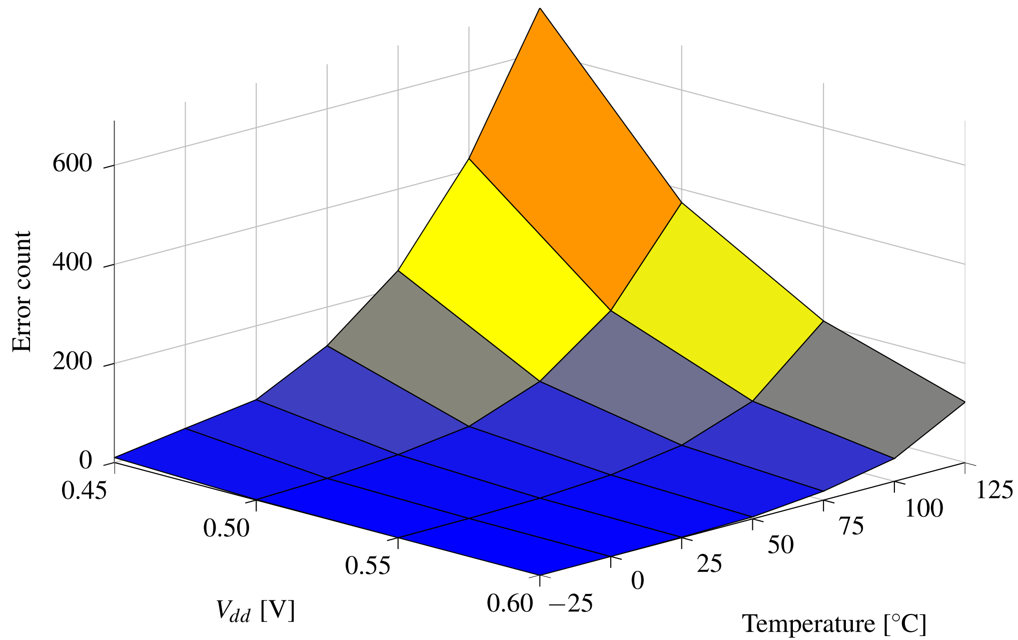


Figure 6.4: 6T SRAM error counts, minimum size transistors, 10 000 Monte Carlo simulations

presented here are detected after *read* operation. According to the preliminary simulation tests the worst-case corner parameters are FS. The weak NMOS transistors are the cause of the unreliability during *read* operation. The pre-charghed bitlines are able to switch the 6T SRAM state, if the N1 and N2 transistors are too weak or if they have inconvenient difference in characteristics due to the manufacturing process. Also in the previous studies the FS has been found to be the worst-case corner parameters. (Yeknami, 2008, p. 21)

The results in the Figure 6.4 show that the high temperature and low V_{dd} are the worst-case when the reliability is concerned. When the temperature is 125 °C and V_{dd} is 0.45 V the number of errors is 690. This means that approximately 6.9 % of the memory cells change their state during a *read* operation. This is too unreliable behavior for a memory circuit and some adjustments must be made to the minimum size 6T SRAM to make it usable.

6.2.3 Larger 6T SRAM Cell Simulations

If the 6T SRAM cell is constructed only from transistors of the same minimum size, the state of the memory has a significant possibility to change during a *read* operation. Obviously, during a *read* operation the state should not change for the memory cell to operate correctly. The sizes of the transistors should be carefully designed to make the cell more reliable. Yeknami (2008) uses double width NMOS transistors in the two inverters of the memory cell. Those are the NMOS transistors N1 and N2 that are connected to the ground in the Figure 6.3. This makes his memory cell operate more reliably with low V_{dd} values. By making the transistors of the 6T SRAM cell bigger the variability reduces and therefore the reliability of the memory improves. (Mingoo et al., 2011, p. 45)

In the Subsection 6.2.2 the minimum-sized 6T SRAM is tested, as presumably the minimum size memory cell is more energy efficient. The technology that is used has the channel length of 130 nm and the minimum width of 150 nm. 150 nm is the default size of a transistor when it is selected from the library.

As an improvement to the 6T SRAM cell reliability, the widths of the inner NMOS transistors, N1 and N2 in the Figure 6.3, are doubled. 10 000 Monte Carlo simulations are run on this modified 6T SRAM cell. The results of 6T SRAM with the larger 300 nm N1 and N2 transistors are illustrated in the Figure 6.5. The larger 6T SRAM is significantly more reliable; the error count is lower than the error count of the minimum-size 6T SRAM.

In these 10 000 Monte Carlo simulations, the largest error count is 33 and it corresponds to 0.33 % error rate at the worst-case parameters. Notable is, that when V_{dd} is 0.50 V there is only 1 error in the 10 000 simulations, and it occurs only in temperatures 100 °C and 125 °C. With the temperatures of 75 °C and under and with V_{dd} of 0.55 V and over no errors occur. The error counts of 6T SRAM with two differently sized N1 and N2 transistors can be compared in the Figure 6.6.

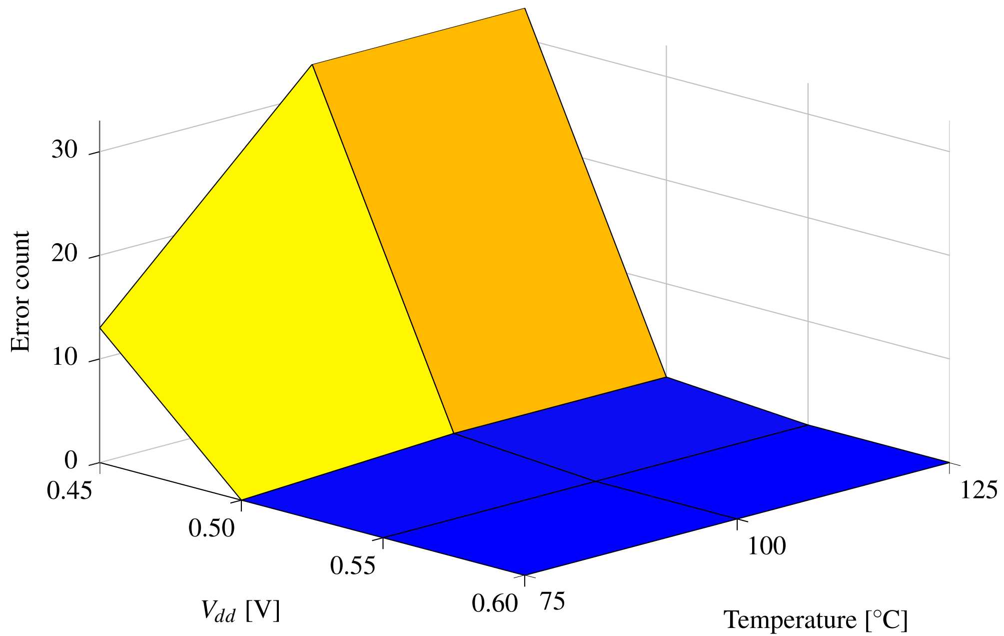
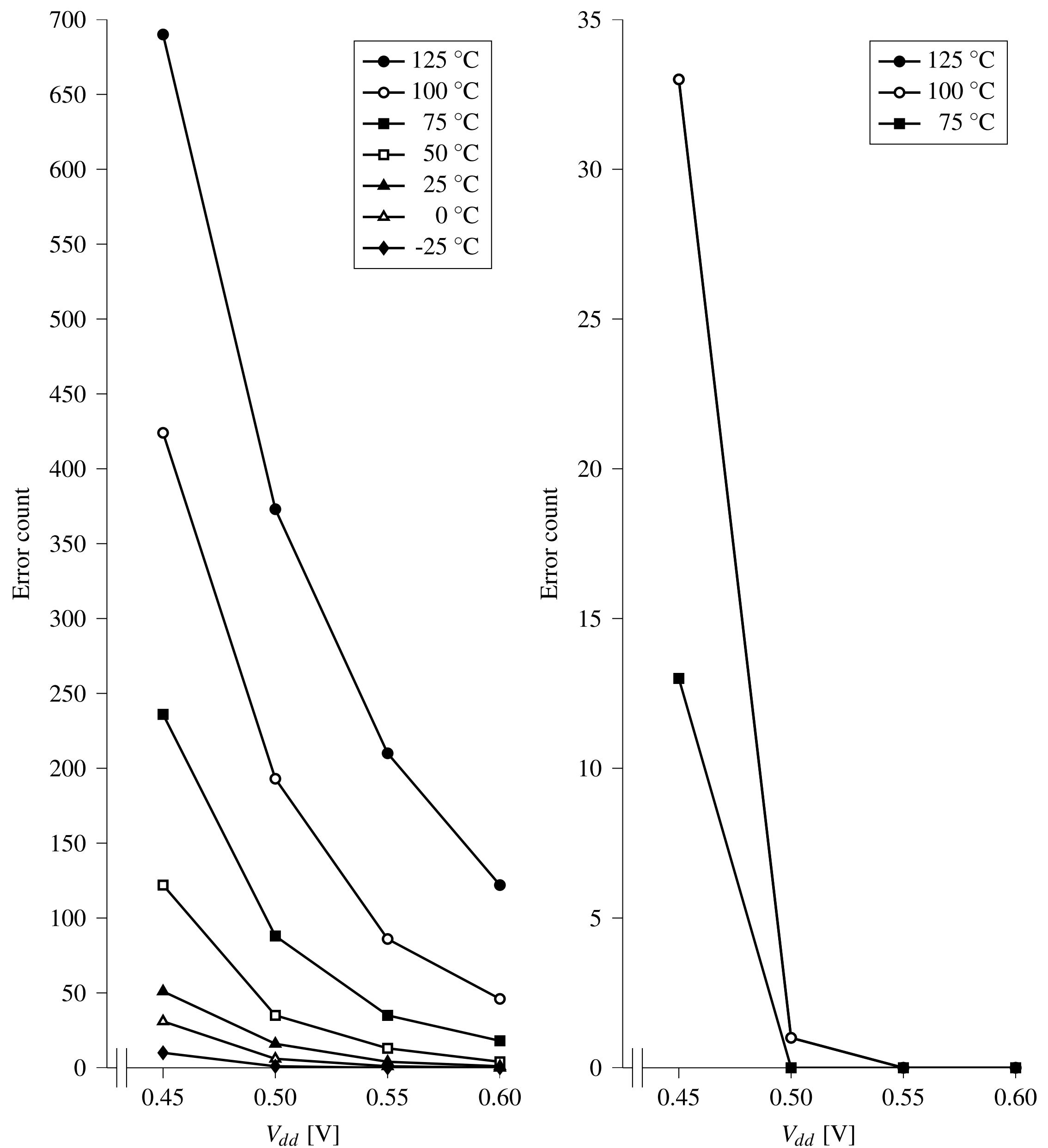


Figure 6.5: 6T SRAM error counts, 300 nm inner transistor widths, 10 000 Monte Carlo simulations

6.3 Signal Noise Margin

The instability of 6T SRAM cell can be explained by observing the behavior of the signal noise margin. It tells how much the signals in the 6T SRAM cell can deviate from the normal values without changing the state of the cell. The signal noise margin of 6T SRAM cell is different depending on the operation that is executed on it. It is the smallest when the memory cell is in its weakest state. This is during the *read* operation. In the simulations of this study, the error states in the 6T SRAM do not occur at all when a *write* operation is executed. (Alorda et al., 2009, p. 2)

The signal noise margin value can be determined by putting two artificial noise sources between the two inverters inside 6T SRAM. This arrangement is illustrated in the Figure 6.7. The noise sources are identical and their value is swept simultaneously from 0 to V_{dd}. The value stored in the 6T SRAM cell is disturbed by the noise sources and the cell changes during the sweep. The value of Q is drawn to the axis and it is mirrored over the



(a) 150 nm inner NMOS width

(b) 300 nm inner NMOS width, note that 100 °C and 125 °C results are identical

Figure 6.6: 6T SRAM logic error count comparison, 10 000 Monte Carlo simulations, FS corners

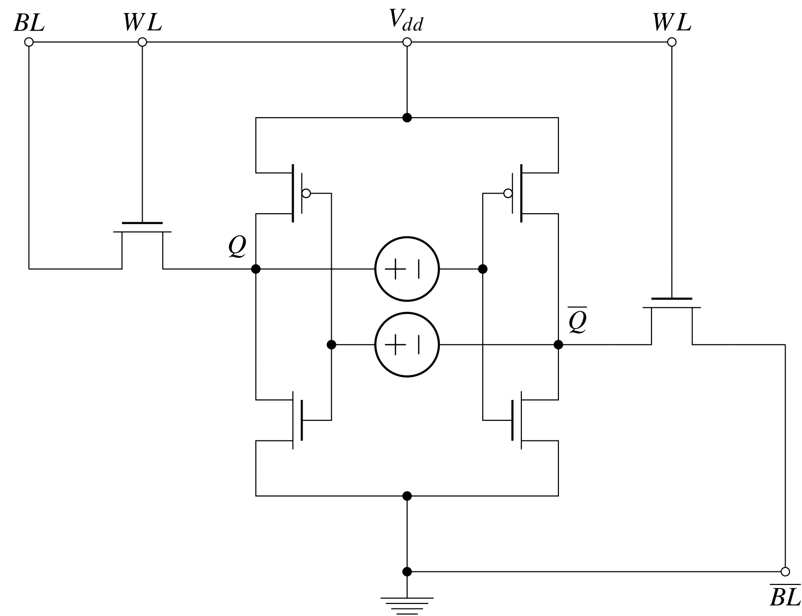
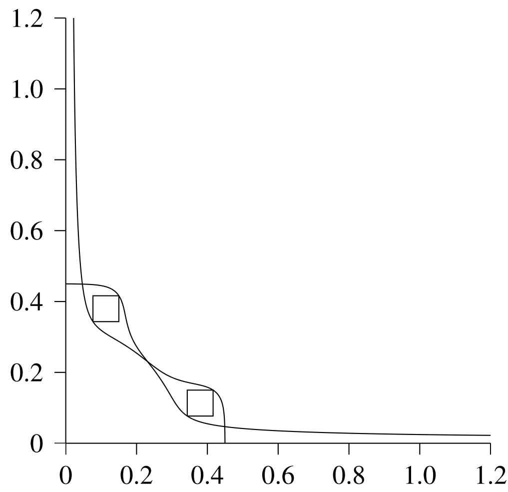


Figure 6.7: 6T SRAM signal noise margin test bench, artificial noise sources added in the middle

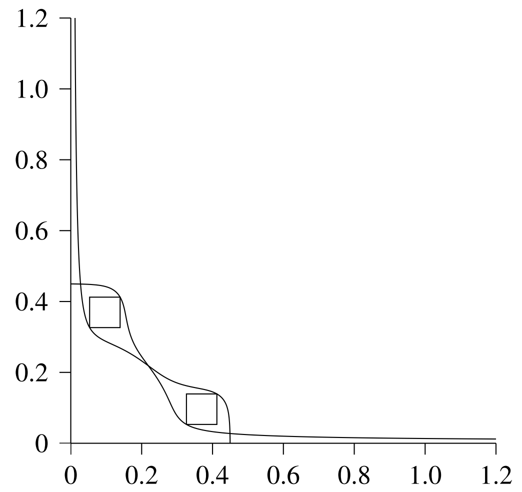
$x = y$ line. These two curves create a so-called butterfly curve. A square as big as possible is then fitted between the two curves, inside the wing of the butterfly. The side of the square is the measurement for the signal noise margin. This is a widely used technique for calculating the signal noise margin values of 6T SRAM cells. (Anami, Yoshimoto, Shinohara, Hirata, & Nakano, 1983; Lohstroh, Seevinck, & de, 1983; Seevinck, List, & Lohstroh, 1987)

According to the reliability results in the Section 6.2 the worst-case parameters when the reliability is concerned, are the FS corner parameters and the high temperature. These parameters are used to measure the signal noise margin of the 6T SRAM cell. The test bench is only the memory cell, where WL and BL are connected to V_{dd} and $\overline{BL}$ to the ground. The noise sources are identical and they are swept simultaneously from 0 V to 1.2 V. 100 Monte Carlo simulations are run, and the minimum signal noise margin butterfly curves of V_{dd} values 0.45 V and 1.20 V are illustrated in the Figure 6.8.

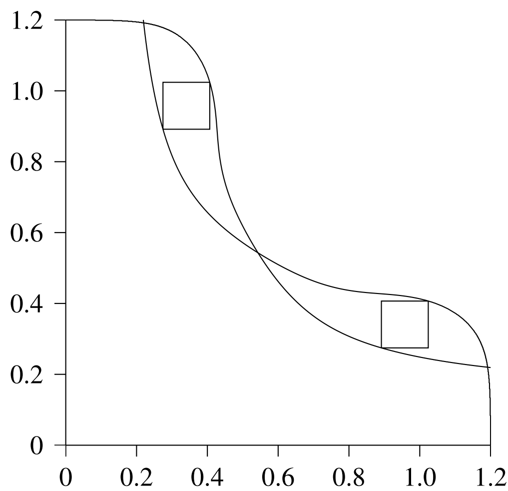
All the results of the signal noise margin simulations are plotted in the Figure 6.9. The



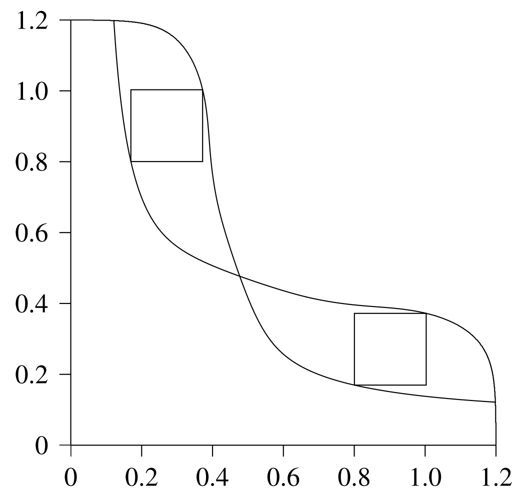
(a) 150 nm inner NMOS widths, 0.45 V V_{dd}



(b) 300 nm inner NMOS widths, 0.45 V V_{dd}



(c) 150 nm inner NMOS widths, 1.2 V V_{dd}



(d) 300 nm inner NMOS widths, 1.2 V V_{dd}

Figure 6.8: 6T SRAM *read* operation butterfly curves, minimum signal noise margin square sizes, 100 Monte Carlo simulations

signal noise margin is smaller when V_{dd} is lower. This implicates, that the 6T SRAM cell is more vulnerable to the noise in the signals and variations between transistors with lower voltages. Also, the 300 nm NMOS cell tolerates noise better, as the signal noise margin values of it in the Figure 6.9(b) are higher than in the minimum-size 6T SRAM signal noise margin values in the Figure 6.9(a). For the larger 6T SRAM cell, also the deviations of the signal noise margins are smaller at each V_{dd} value.

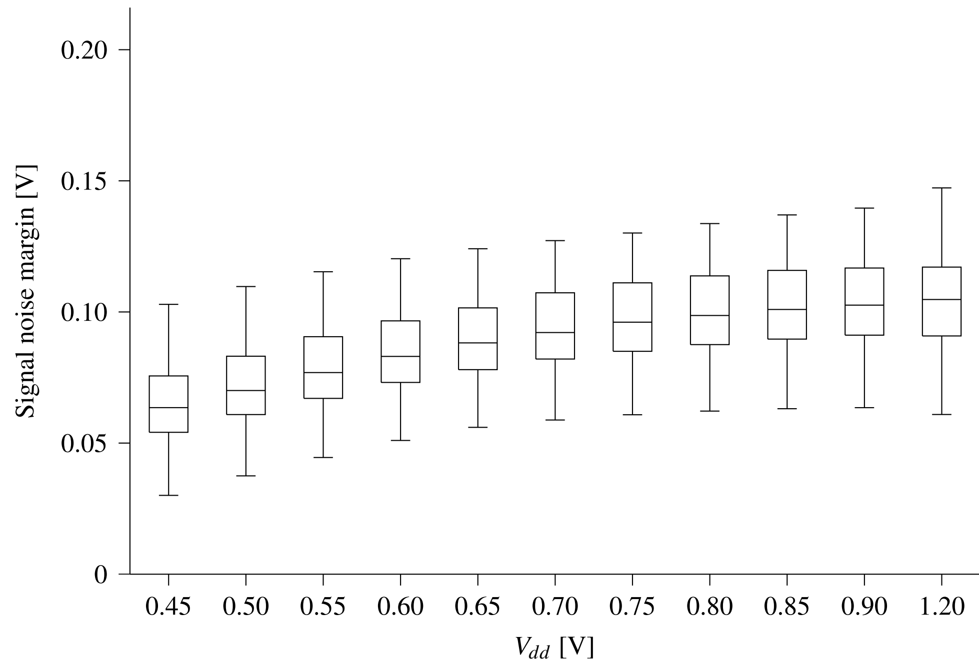
6.4 Dynamic Energy Consumption

The dynamic energy consumption of 6T SRAM is determined by measuring the currents from the voltage supply to the 6T SRAM cell. The measurements are made from the *source* nodes of the P1 and P2 transistors. The consumed energy can be calculated from equation

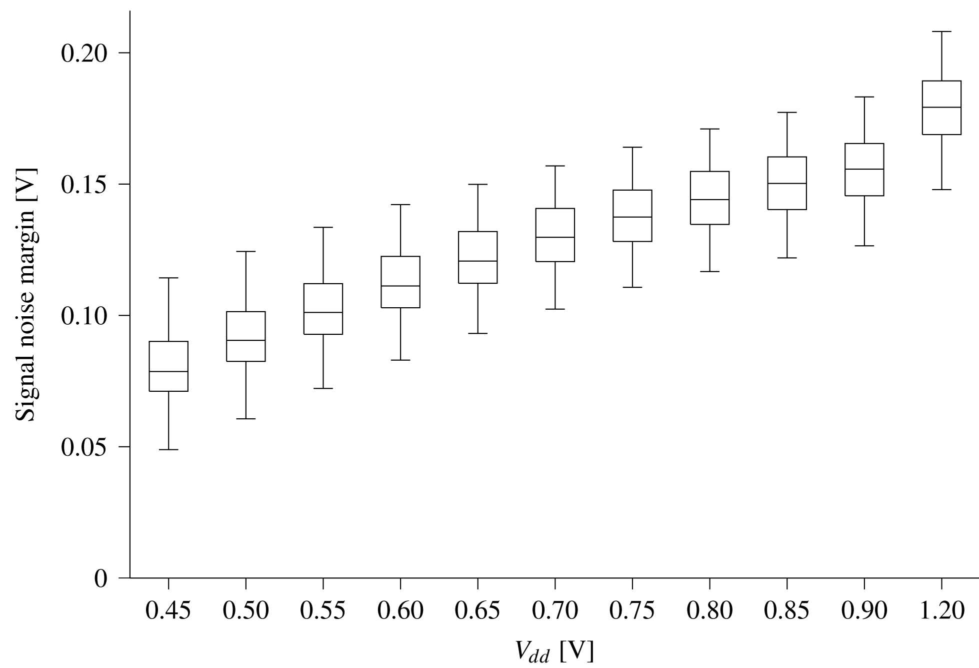
$$E_{operation} \approx V_{dd} \cdot \int_{t_1}^{t_2} (I_{P1,source} + I_{P2,source}), \tag{6.3}$$

where $E_{operation}$ is the amount of energy that 6T SRAM uses when the particular operation is executed. V_{dd} is the supply voltage. $I_{P1,source}$ and the $I_{P1,source}$ are the currents between the nodes *source* and *drain* of the transistors P1 and P2. P1 and P2 are illustrated in the Figure 6.3. The two currents are added together, and integrated over 5 ns time period after the beginning of each *write* and 2.5 ns time period after the beginning of each *read* operation. In the Equation 6.3 t_1 is the time when the operation starts, and t_2 is the time when the operation ends. The result is the amount of charge that flow through the memory cell during each operation. After this, the charges are multiplied by V_{dd} and the result is the consumed energy.

Energy consumption is the largest with the FS corner parameters. This is illustrated in



(a) 150 nm inner NMOS widths



(b) 300 nm inner NMOS widths

Figure 6.9: 6T SRAM signal noise margins, 100 Monte Carlo simulations

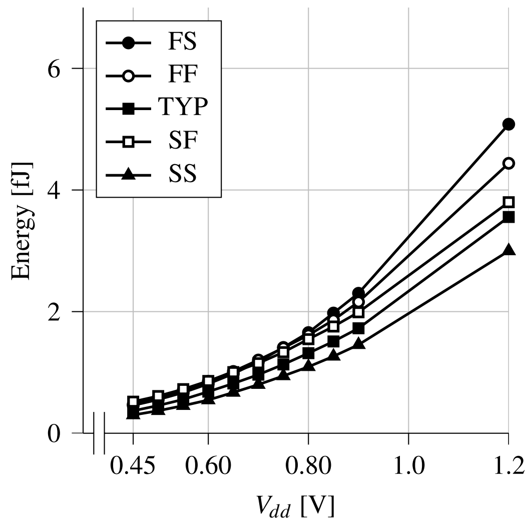
the Figure 6.10, where the energy consumption with different corner parameters are compared. The amount of energy consumed by the 6T SRAM cell with the worst-case corners during a single operation is illustrated in the Figure 6.11. The energy consumption is lower with lower V_{dd} values. This is expected as the energy consumption depends greatly on the V_{dd}. Higher temperatures are the worst-case when the energy consumption is concerned.

The energy consumption of the worst-case corners FS and temperature 125 °C is illustrated in more detail in the Figure 6.12. The energy consumption seems to have a large variation, but still a significant drop in it can be observed when V_{dd} is lowered from the nominal 1.2 V to the region under 0.8 V.

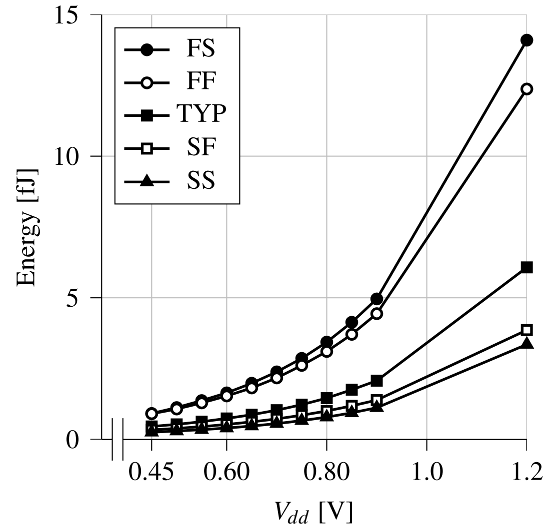
The Figure 6.10, the Figure 6.11 and the Figure 6.12 all show that the energy consumption is measured to be larger with the minimum size 6T SRAM cell. The Q node voltages during the *read* 0 operation are illustrated in the Figure 6.13. In the ideal situation, the node Q should stay at ground potential. However, the pre-charged bitline pulls the Q up slightly. When comparing the Figure 6.13(a) and the Figure 6.13(b) it is notable that the Q rises higher and stays up a longer time in the minimum size 6T SRAM case. While the Q is connected to the *gate* nodes of the N2 and P2 transistors, they let some amount of current flow through the wrong side of the 6T SRAM cell unintentionally. This implicates that the energy efficiency of the 6T SRAM can be improved by using larger transistors. This is counterintuitive and surprising and the causes should be further researched.

6.5 Leak Current

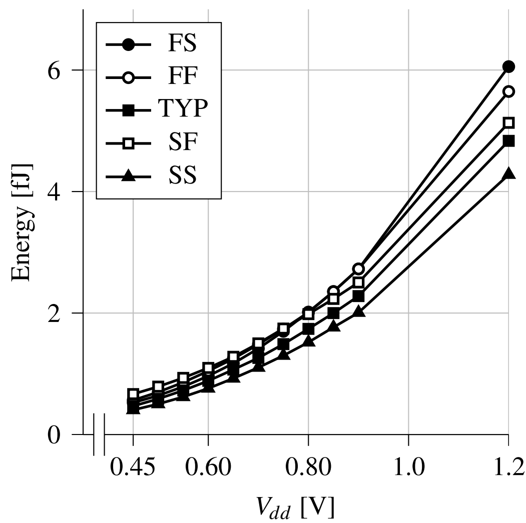
At the standby stage there are always leak currents flowing through 6T SRAM cell. That is because there is always some leakage, if there is a voltage difference over some device, and during the standby state of 6T SRAM there are always voltage differences over some



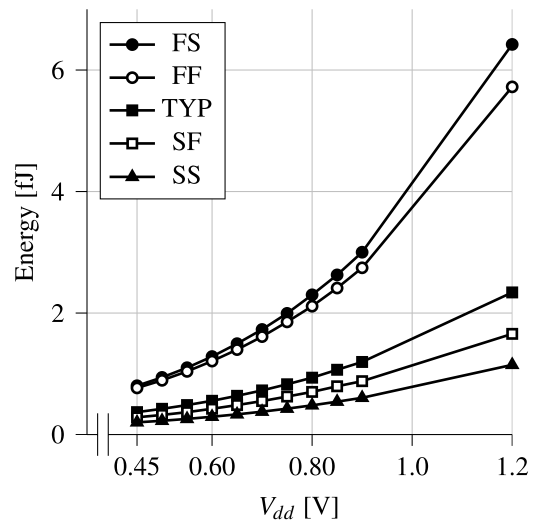
(a) *Write* operation, 150 nm inner NMOS widths



(b) *Read* operation, 150 nm inner NMOS widths

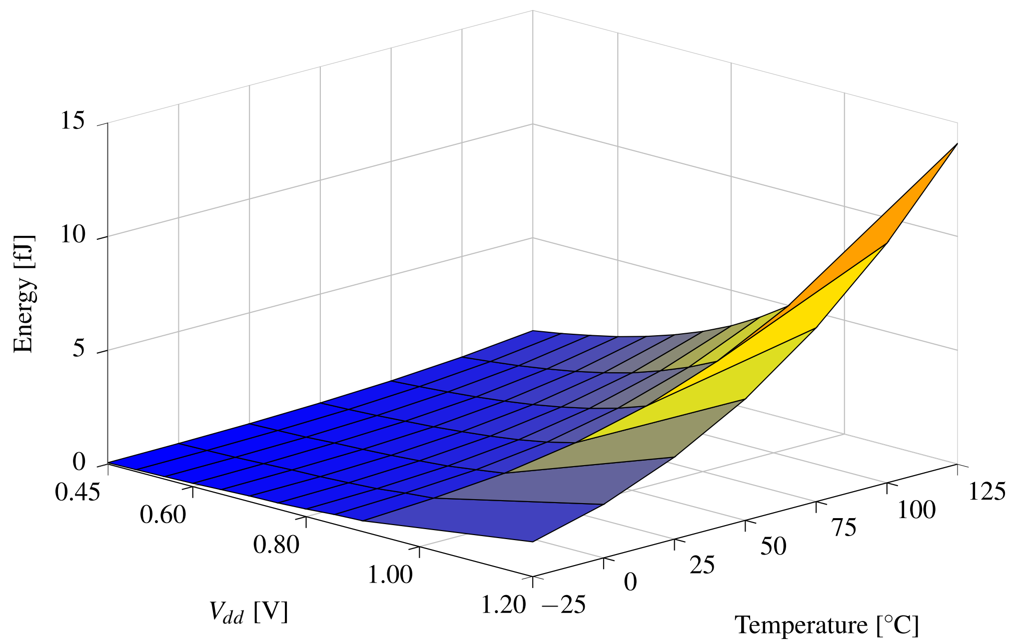


(c) *Write* operation, 300 nm inner NMOS widths

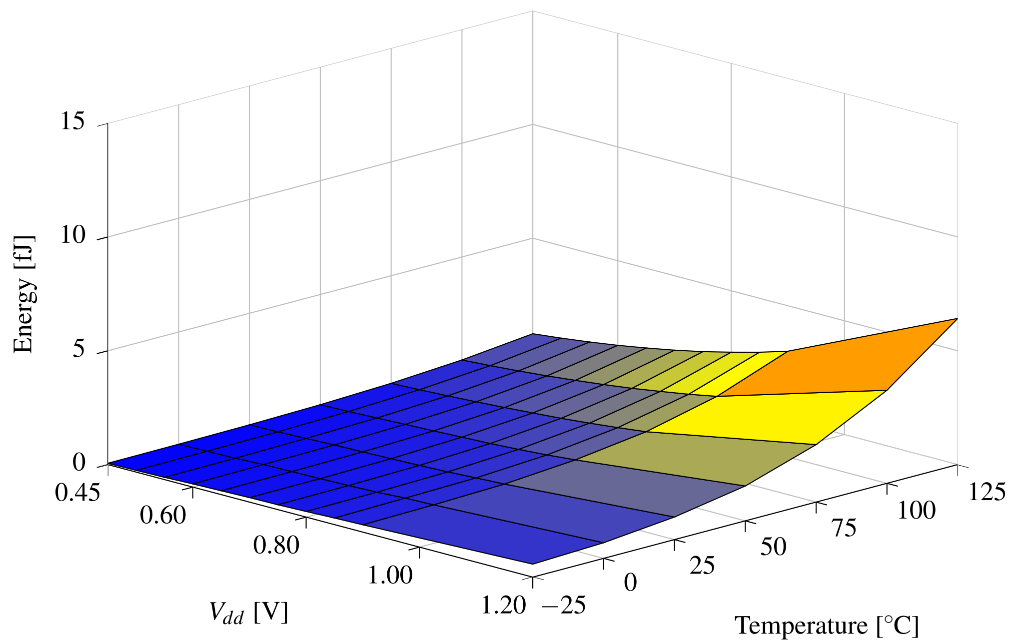


(d) *Read* operation, 300 nm inner NMOS widths

Figure 6.10: 6T SRAM energy consumption

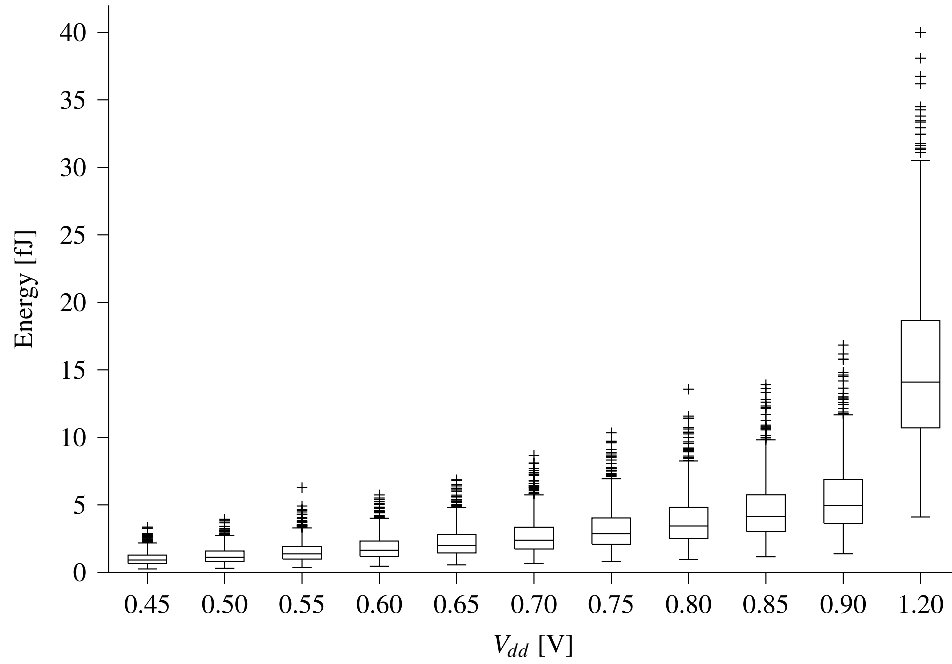


(a) 150 nm inner NMOS widths

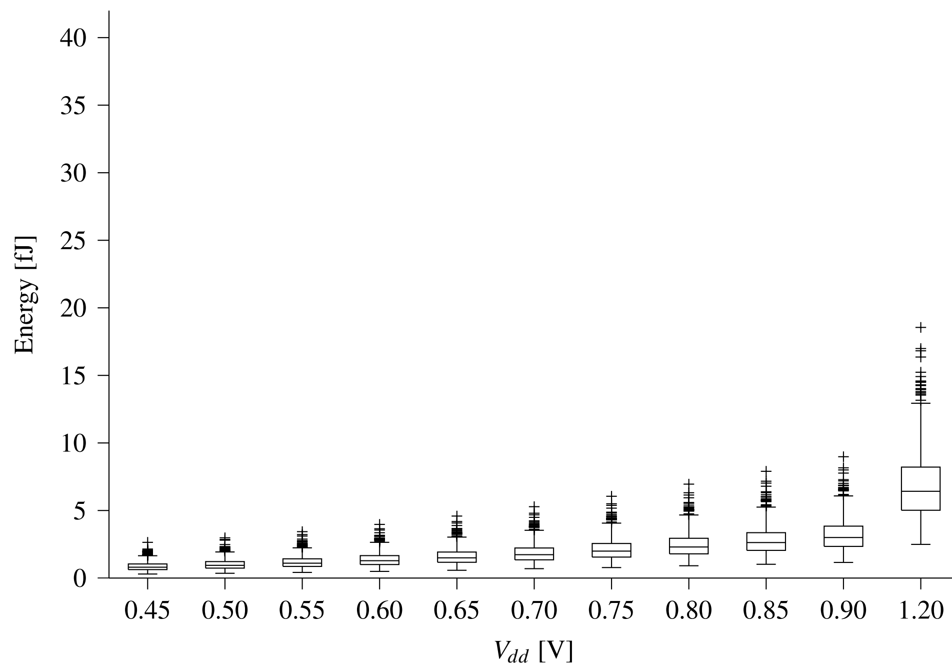


(b) 300 nm inner NMOS widths

Figure 6.11: 6T SRAM energy consumption, *read* operation, FS corners

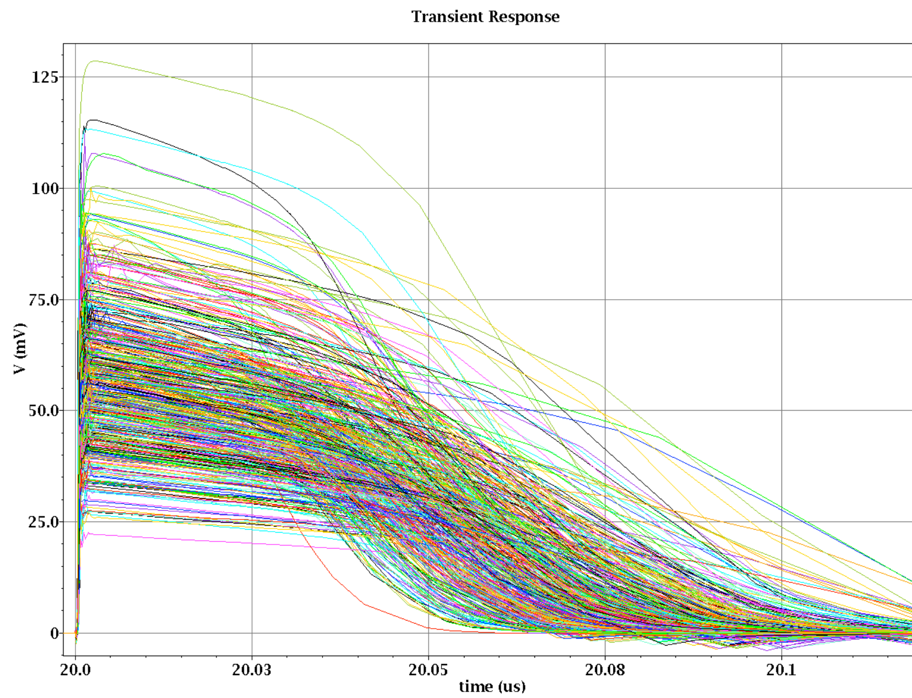


(a) 150 nm inner NMOS widths

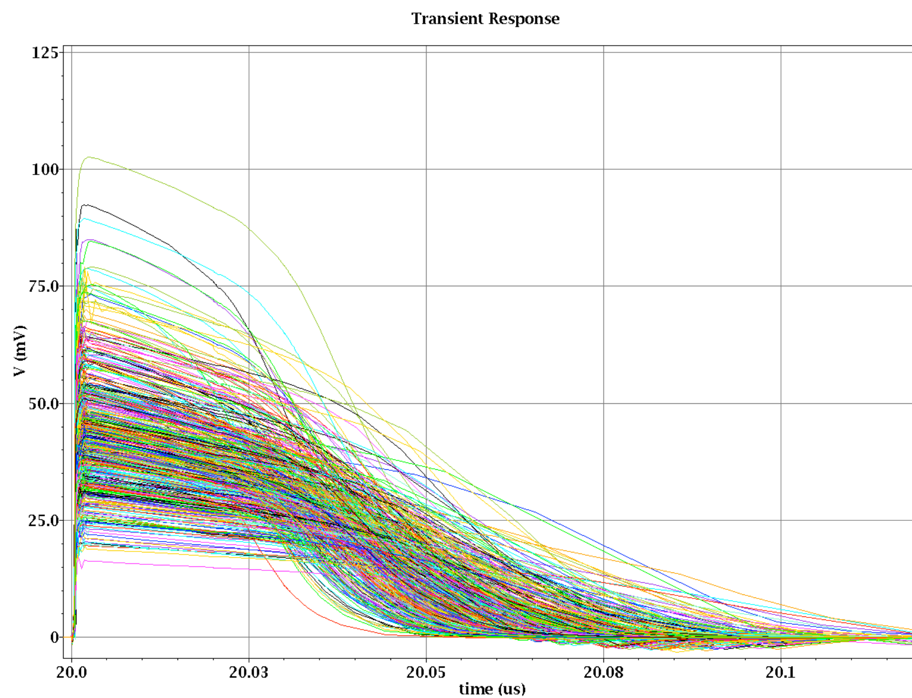


(b) 300 nm inner NMOS widths

Figure 6.12: 6T SRAM energy consumption deviations, 1 000 Monte Carlo simulations



(a) 150 nm inner NMOS widths



(b) 300 nm inner NMOS widths

Figure 6.13: 6T SRAM node Q voltages during *read* operation, 500 Monte Carlo simulations, TYP corners, 25 °C temperature

transistors. The leak currents can be measured by putting the 6T SRAM to the standby state, and waiting a long enough time for the circuit to stabilize. All the currents that are left, are leakage. In this case study currents through all six transistors of the memory cell are measured and added together, and the result is divided by two. The result has to be divided by two, because the added sum contains the currents that go in to the 6T SRAM cell and the currents that come out from the 6T SRAM cell.

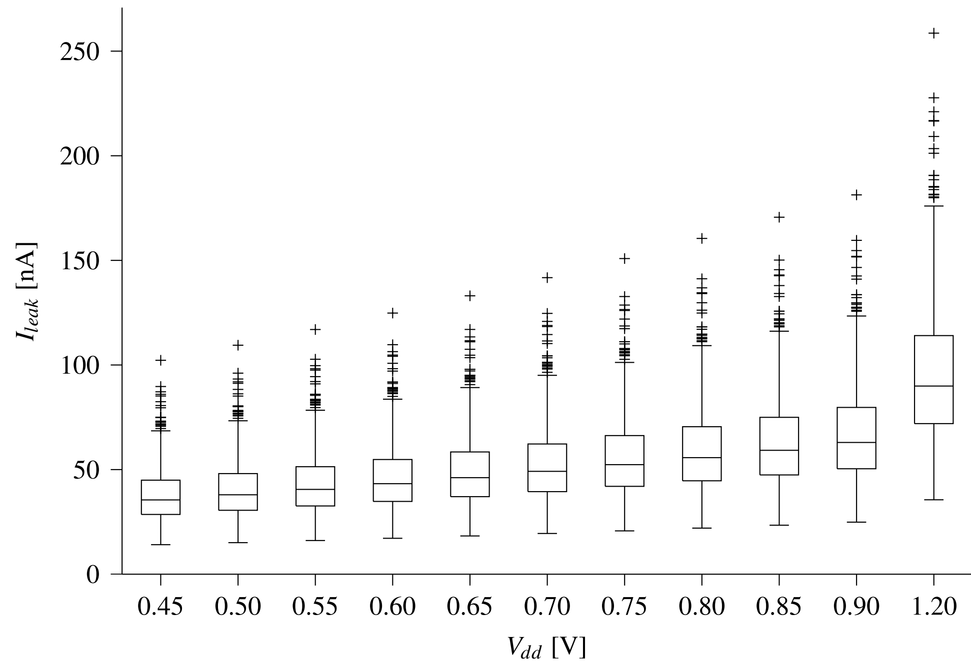
The results are illustrated in the Figure 6.14. The different sizing of the N1 and N2 transistors does not seem to have an effect on the leakage. This implicates that the minimum size PMOS transistors limit the leakage through the circuit.

6.6 Propagation Delay

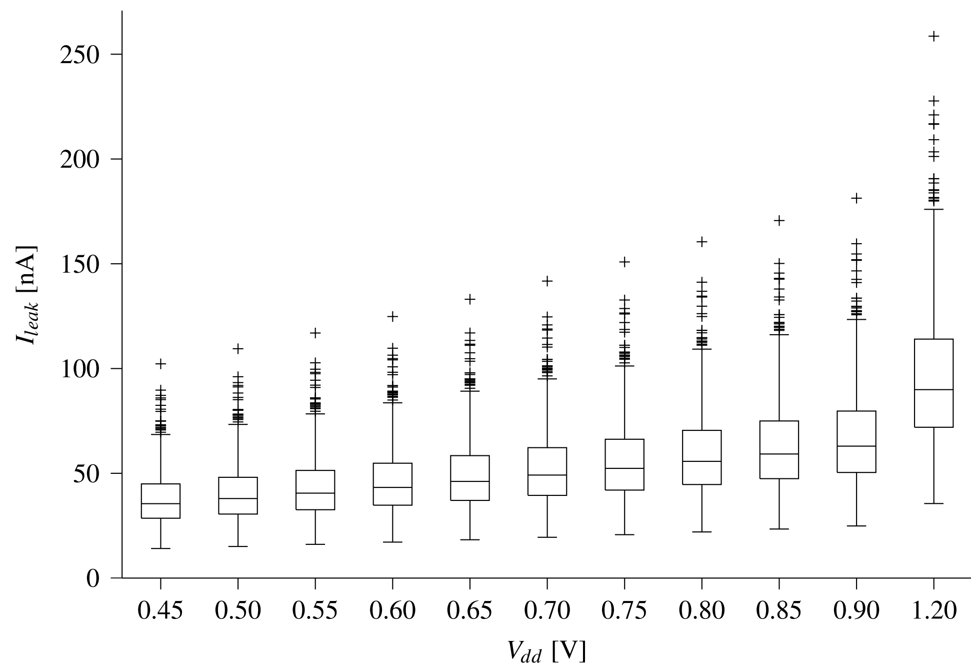
The highest *read* and *write* frequency of a 6T SRAM memory circuit is determined not only by the delays inside a single memory cell, but also by the size of the memory column, as longer bitlines have bigger capacitances. Also, the peripheral circuits and their delays have to be considered. In this study, the delays are measured from the single 6T SRAM cell. This gives a perspective to the delay differences with different parameters.

When writing bit value 1 to the memory, the propagation delay of the 6T SRAM cell is measured starting from the moment that the $\overline{WL}$ rises over $\frac{1}{2} \cdot V_{dd}$ to the moment that the memory cell node Q has risen over $\frac{1}{2} \cdot V_{dd}$. When writing bit value 0 to the memory, the measurement is starting from the moment that the $\overline{WL}$ rises over $\frac{1}{2} \cdot V_{dd}$ and ends when the memory cell node Q has fallen under $\frac{1}{2} \cdot V_{dd}$.

The *read* operation is measured from the 50 % change in the WL to the 50 % change in the other bitline. As the bitlines are pre-charged before the *read* operation, the BL is changing when 0 is being read and the $\overline{BL}$ is changing when the 1 is being read. The *read* operation



(a) 150 nm NMOS widths



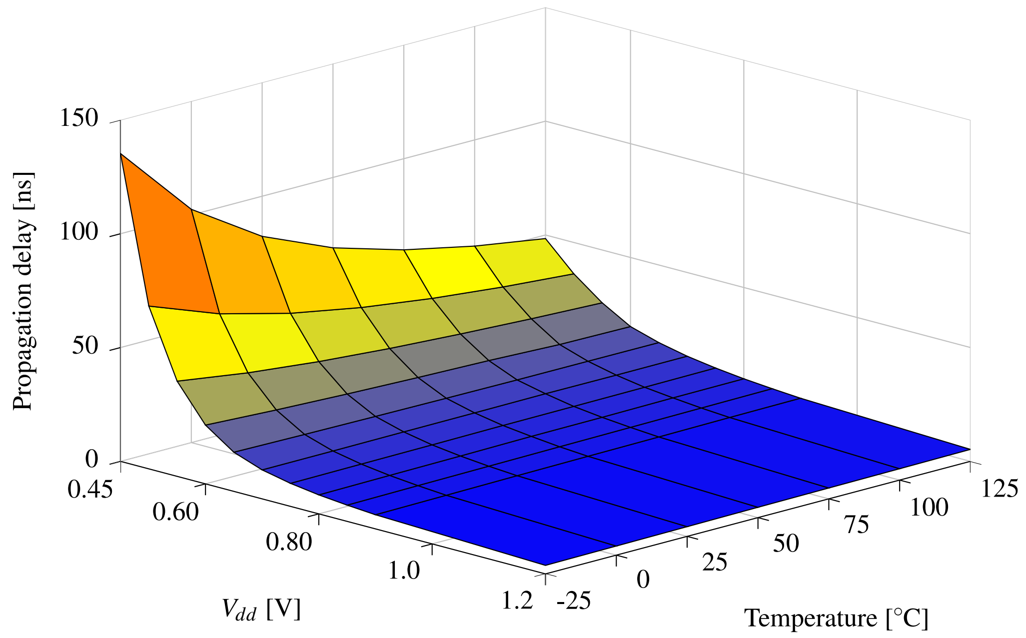
(b) 300 nm NMOS widths

Figure 6.14: 6T SRAM leak currents, 1 000 Monte Carlo simulations

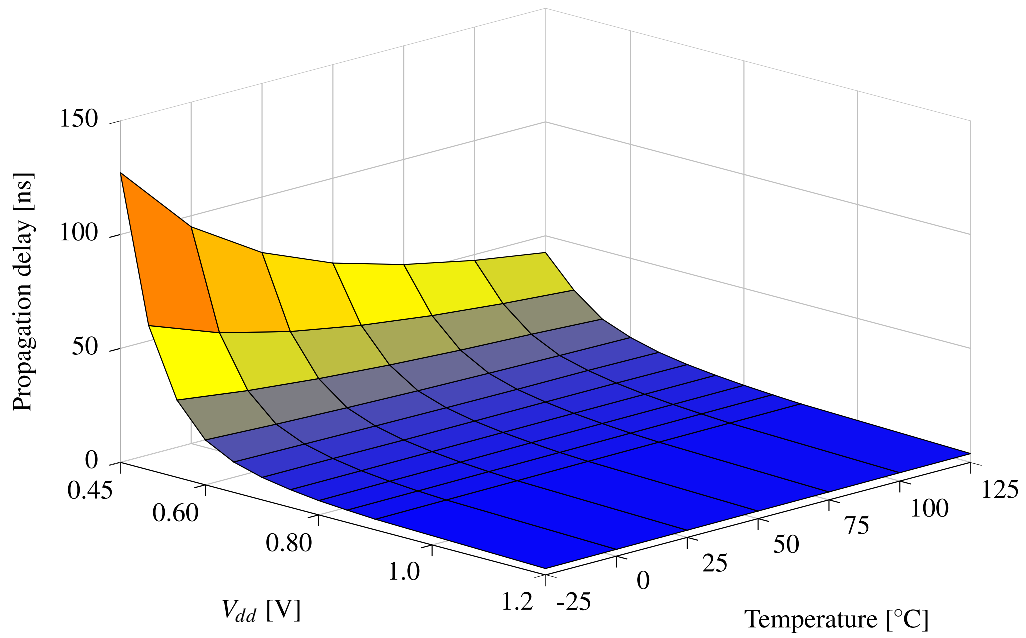
is guided by the timing of the control signals of the sense amplifier circuit. In this study, the *read* operation timing is designed to work with all V_{dd} values from 0.45 V to 1.20 V. Therefore there is some potential to improve the *read* operation, if it could be designed for a smaller voltage spread.

The worst-case corner values when the delays are concerned are the SS parameters. This is because then all the transistors are mainly slower than average. The Figure 6.15 illustrates the propagation delays when the SS corner values are used. The *read* operation is much slower than the *write* operation. This is because during the *write* operation the bitlines drive the memory cell, and because the capacitance in the bitlines is large if compared with the sizes of the transistors inside the memory cell, the operation is fast. During the *read* operation, the memory cell has to discharge one of the bitlines, which has a large capacitance in comparison to the transistor sizes in the memory cell, and therefore the operation is slow. Notable is, that the sense amplifier is capable of producing the correct output before the other bitline is discharged. The accelerated output generation is the main function of the sense amplifier.

As is shown in the Figure 6.15, the propagation delays are larger when the temperature is lower. The $-25\,^{\circ}$C simulations with the worst-case corner parameters SS are illustrated in more detail in the Figure 6.16 and the Figure 6.17. The propagation delays behave in the same way as with the FO4 inverter in the Figure 5.5(a); the variation is larger with lower V_{dd} values. Only, the magnitudes of the delays are larger. 6T SRAM cells with different sized N1 and N2 transistors seem to have almost the same propagation delay. The smaller cell is a little faster in *write* operation, and the larger cell is a little faster in *read* operation. The delay variation grows significantly when V_{dd} gets smaller. This is illustrated in the Figure 6.16 and the Figure 6.17. These box plot figures tell that the t_p deviations are almost symmetrical.

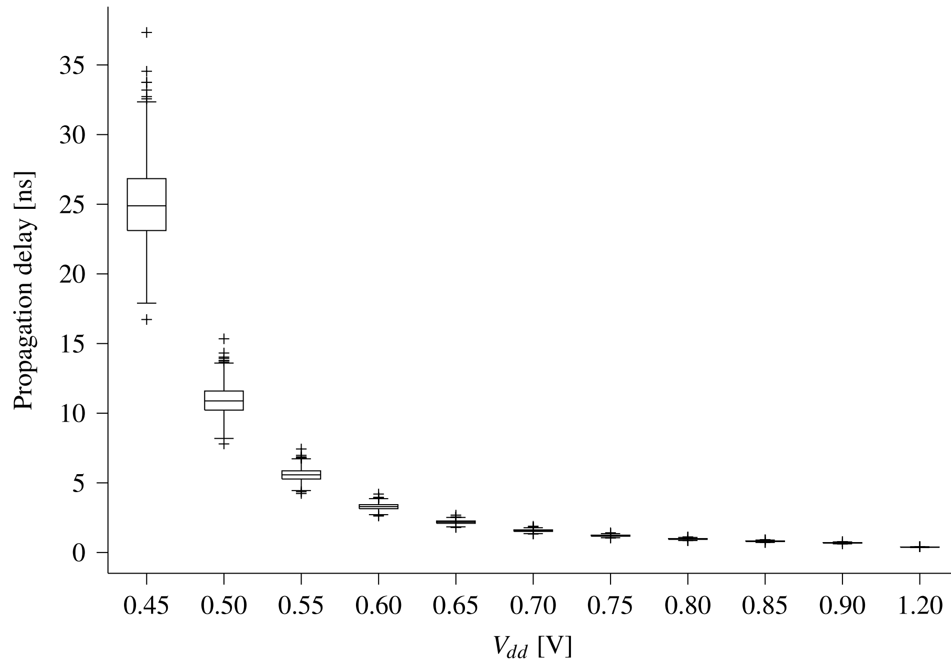


(a) 150 nm inner NMOS widths

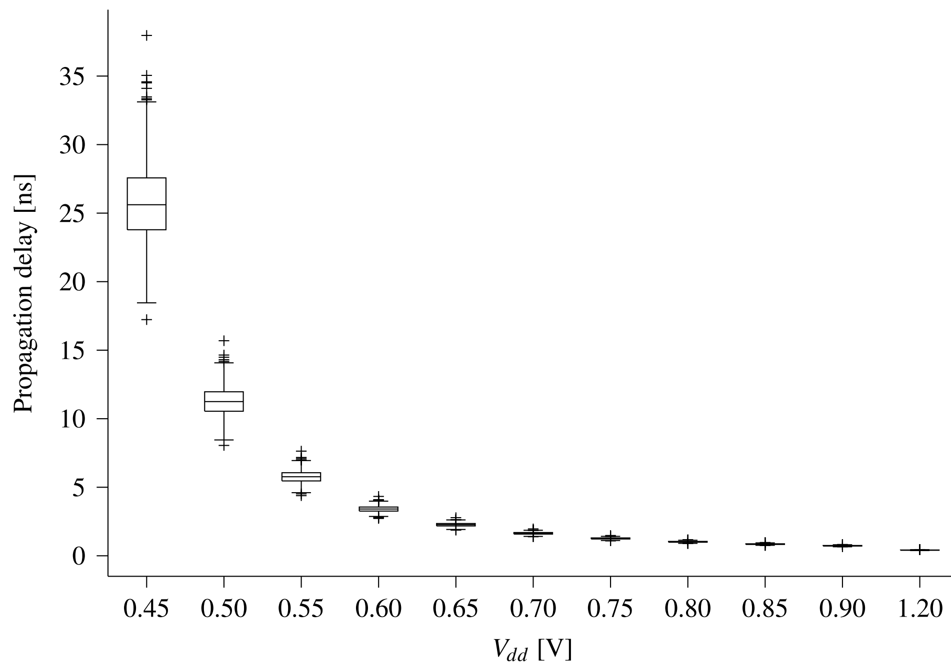


(b) 300 nm inner NMOS widths

Figure 6.15: 6T SRAM propagation delay maximum values, *read* operation, 1 000 Monte Carlo simulations, worst-case SS corners

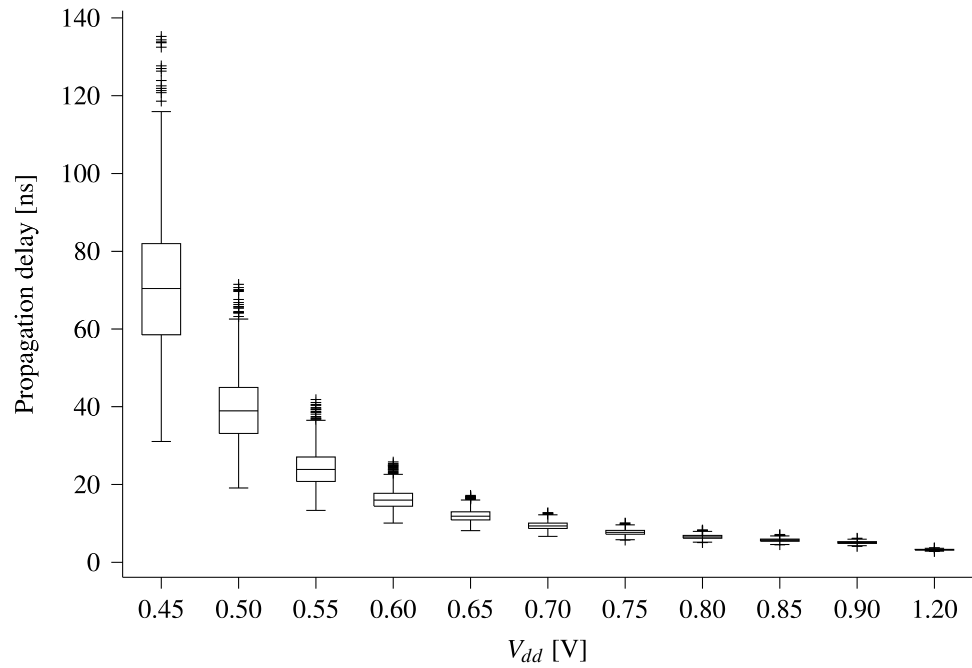


(a) 150 nm NMOS width

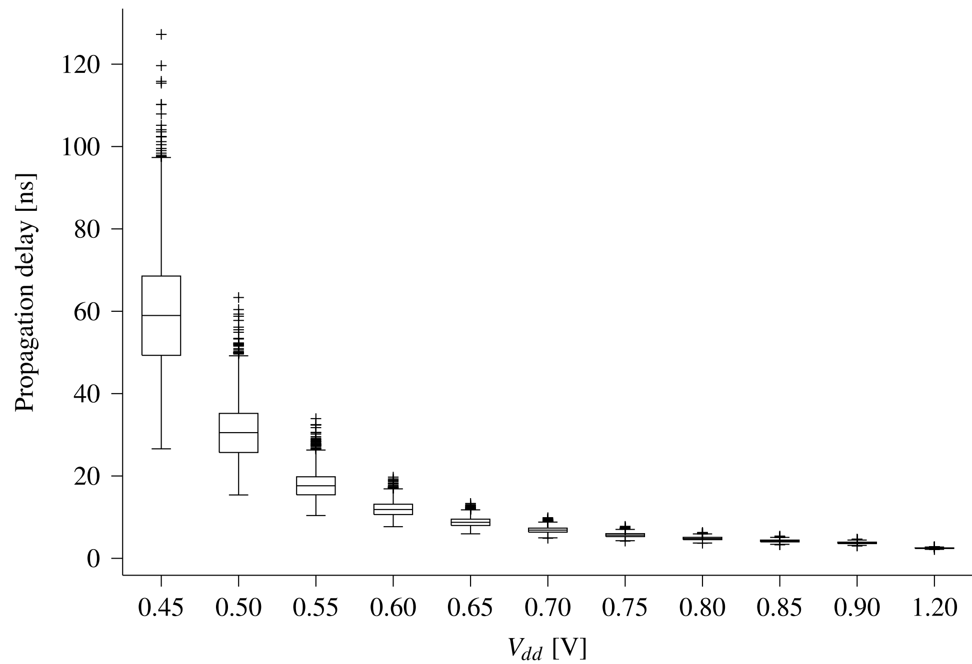


(b) 300 nm NMOS width

Figure 6.16: 6T SRAM propagation delays, *write* operation, 1 000 Monte Carlo simulations, worst-case SS corners



(a) 150 nm inner NMOS widths



(b) 300 nm inner NMOS widths

Figure 6.17: 6T SRAM propagation delays, *read* operation, 1 000 Monte Carlo simulations, worst-case SS corners

7 Conclusions

As the IC technology is advancing, larger amounts of transistors are fitted on a single IC chip. The power consumption increases and as a consequence more heat is generated. The heat generation limits the power usage, and therefore it limits the frequency of operations executed on a single chip. There is a continuous demand for lower energy consumption devices that would make more complicated IC chips possible. Very small environment monitoring sensor-based devices that can be used to monitor human activities, buildings, bridges and so on are also a new field of study that needs low energy solutions. The Near-Threshold Computing is a technique that can be used to reduce the power consumption of IC devices.

In this study, the characteristics of the Near-Threshold Computing were reviewed. Also, the behavior of a 130 nm IC technology was examined to find out the possibilities of applying the Near-Threshold Computing to an actual device. It is possible to produce low-power devices at reasonable costs, if the Near-Threshold Computing is utilized to a low price point technology. This study examined a 130 nm technology by simulating two different CMOS devices, FO4 inverter and 6T SRAM cell. These devices can be considered representing a large portion of area and energy consumption of a conventional IC circuit.

After the introduction in the Chapter 1, in the Chapter 2 technical background matters

which relate to the Near-Threshold Computing were looked into. The basic behavior of a MOSFET was discussed and the concept of threshold voltage was explained. The CMOS, which is the most widely used technique for logic device construction, was also presented. The 6T SRAM type memory structure and the peripheral circuits that are needed for reading from it and writing to it, were also presented.

In the Chapter 3 the principles of the Near-Threshold Computing were explained. The previous studies and literature were used as a basis to explain different aspects of the Near-Threshold Computing, and how IC devices behave when the supply voltage is reduced to the Near-Threshold Computing region.

The characteristics of the Near-Threshold Computing on a specific 130 nm IC manufacturing technology were researched by studying two different cases: an FO4 inverter and a 6T SRAM cell. These case studies were introduced in the Chapter 4. The FO4 inverter is a good representation of a logic device. The behavior of it can generally be scaled to larger CMOS constructs. In the Chapter 5 FO4 inverter was tested for Near-Threshold Computing by using different voltages. The 6T SRAM on the other hand is a widely used on-chip memory structure, and in the Chapter 6 it was tested for the Near-Threshold Computing usage.

The case studies were executed by generating multiple copies of the devices with real life manufacturing variations. This was conducted by running multiple Monte Carlo simulations and applying different variations to the devices at each simulation. The simulations gave a good representation of multiple actual manufactured devices. Each of the generated devices were simulated through various use cases and their behavior was observed.

The simulations were conducted by using different supply voltages starting from the threshold voltage of the used technology 0.45 V and ending at the nominal use voltage 1.2 V. Also, different temperatures were used from $-25\,^{\circ}\mathrm{C}$ to $125\,^{\circ}\mathrm{C}$. Different characteristics of the devices were analyzed. These were the propagation delays, energy con-

sumption, leak currents and error occurrence in the memory.

Basically, if the supply voltage is reduced, the energy consumption of the circuit becomes significantly lower. This was shown to be true in each simulation conducted in this study. This phenomenon is the grounds of the Near-Threshold Computing. The downside is that with lower voltages circuits become more sensitive to the manufacturing process variations of the transistors. This phenomenon is reflected as larger variations in the logic gate delays and unreliability in memory cell structures, which are based on carefully designed transistor sizes. In short, the behavior of the circuits became to some extent unpredictable.

The unpredictability of devices can be reduced with different countermeasures, for example designing longer logic paths or using larger transistors. In this study, the longer logic paths and larger transistors were both shown to be effective ways to make circuits work correctly and more predictably in the Near-Threshold Computing voltages. In addition, there is potential to using 130 nm technology in Near-Threshold Computing applications, as there is considerable margin to reduce the V_{dd} without harming the behavior of the device significantly.

In both case studies, the energy consumption of the devices dropped to under 45 % of the nominal when V_{dd} was decreased from 1.2 V to 0.80 V. It dropped even further to under 25 % of the nominal when V_{dd} was decreased from 1.2 V to 0.60 V. The energy savings are significant already with V_{dd} of 0.8 V. This gives the voltage levels a good margin to variate, and still stay well above the threshold voltage, which is 0.45 V.

In the case studies, the propagation delays of the devices were less than 2.7 times longer when V_{dd} was decreased from 1.2 V to 0.80 V. The propagation delays were less than 10.4 times longer when V_{dd} was decreased from 1.2 V to 0.60 V. These numbers are in line with the mentions about the Near-Threshold Computing in the literature. These longer delays are not significant if a low-power and low performance device is designed.

The studies about FO4 inverter in the in the Chapter 5 give a good general view of how logic blocks behave with lower voltages. The results can be used as guidelines when considering larger Near-Threshold Computing devices that contain logic blocks.

NAND logic gates for example can be used to construct many other logic gates. NAND and NOR circuits would be one choice to continue the Near-Threshold Computing research in the future. As registers are an essential part of a synchronous circuit, further examination could also focus on them.

There are many variations of SRAM structures. They take different amounts of area and they consume different amounts of energy. If Near-Threshold Computing memory is designed, these should be considered and compared. The observations made in this study on 6T SRAM give guidelines if Near-Threshold Computing memory is designed. The results have built ground for further investigation and comparisons to other memory cell types, for example the 8T-SRAM.

It is said that 8T-SRAM would not have a significant addition to circuit area if compared with 6T SRAM. 8T-SRAM cells would also tolerate more processing variability during *read* operation, and therefore they would be more usable with low V_{dd}. It would be useful to test how the 8T-SRAM behaves in the Near-Threshold Computing region. Maybe the small addition in the area and the power consumption would give some benefits in reliability. As the results in the Chapter 6 show, a Near-Threshold Computing 6T SRAM can not be constructed from only minimum size transistors; it has to be made more reliable by increasing the sizes of at least two of the transistors. Therefore, if two transistors in 6T SRAM cell were doubled, the memory cell area would be even closer to the area of a minimum-sized 8T-SRAM cell. If the *read* operation of the 8T-SRAM cell would be more reliable with just minimum transistors, the size difference between a reliable 6T SRAM and 8T-SRAM might be so small, that the 8T-SRAM could be a more reasonable option for memory. Testing this would be imperative if Near-Threshold Computing memory

blocks are designed.

In this study, there was no distinct implication of the 6T SRAM being totally unusable in the Near-Threshold Computing. Especially, as the presumption that the minimum size 6T SRAM cell would consume the least energy, was shown to be wrong in the particular case in the Chapter 6; in this case the energy consumption decreased as the sizes of two inner NMOS transistors increased. This phenomenon was surprising and it should be further researched. For these reasons, further investigation is appropriate. Also, inventions that have been made for the sub-threshold computing in mind could be used in the Near-Threshold Computing memory designs.

REFERENCES

Alorda, B., Torrens, G., Bota, S. & Segura, J. (2009). Static-noise margin analysis during read operation of 6t sram cells. *Conference on design of circuits and integrated systems, DCIS proceedings*.

Anami, K., Yoshimoto, M., Shinohara, H., Hirata, Y. & Nakano, T. (1983). Design consideration of a static memory cell. *IEEE Journal of Solid-State Circuits*, *18*(4), 414–418.

Baker, R. J. (2010). *Cmos circuit design, layout, and simulation*. Wiley-IEEE Press.

Bo, Z., Hanson, S., Blaauw, D. & Sylvester, D. (2005). Analysis and mitigation of variability in subthreshold design. *Proceedings of the 2005 international symposium on low power electronics and design 2005, ISLPED '05.* 20–25.

Bol, D., Boyd, S. & Dornfeld, D. (2011). Application-aware lca of semiconductors: life-cycle energy of microprocessors from high-performance 32nm cpu to ultra-low-power 130nm mcu. *IEEE international symposium on sustainable systems and technology (ISSST), 2011*, 1–16.

Bol, D., De, J., Vos, Hocquet, C., Botman, F., Durvaux, F., Boyd, S., ... Legat, J. (2013). SleepWalker: a 25-MHz 0.4-v sub-mm^2 7-μW/MHz microcontroller in 65-nm LP/GP CMOS for low-carbon wireless sensor nodes. *IEEE Journal of Solid-State Circuits*, *48*(1), 20–232.

Carlson, A., Guo, Z., Balasubramanian, S., Zlatanovici, R., King, T.-J., Liu & Nikolic, B. (2010). Sram read/write margin enhancements using finfets. *IEEE Trans. VLSI Syst. 18*(6), 887–900. Retrieved from http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5229332

Chang, L., Frank, D., Montoye, R., Koester, S., Ji, B., Coteus, P., ... Haensch, W. (2010). Practical strategies for power-efficient computing technologies. *Proceedings of the IEEE*, *98*(2), 215–2236.

Circuits Multi-Projects, M.-P. C. (2010). Technical specifications for cmos 130nm (hcmos9gp). Retrieved on 12. 09. 2012, from http://cmp.imag.fr/products/ic/?p=STHCMOS9

De, H., Man. (2005). Ambient intelligence: gigascale dreams and nanoscale realities. *2005 IEEE international solid-state circuits conference, 2005. digest of technical papers. ISSCC.* 29–35 Vol. 1.

Ding-Ming, K., Ching-Hua, H., Chung-Ping, K., Chi-Hsien, C., Min-Chung, H., Yi-Chun, C., … Yung-Fa, C. (2006). Sram cell current in low leakage design. *2006 IEEE international workshop on memory technology, design, and testing, 2006. MTDT '06.* 6 pp.

Dreslinski, R., Wieckowski, M., Blaauw, D., Sylvester, D. & Mudge, T. (2010). Near-threshold computing: reclaiming Moore's law through energy efficient integrated circuits. *Proceedings of the IEEE*, *98*(2), 253–2266.

Hanson, S., Zhai, B., Bernstein, K., Blaauw, D., Bryant, A., Chang, L., … Sylvester, D. M. (2006). Ultralow-voltage, minimum-energy cmos. *IBM Journal of Research and Development*, *50*(4.5), 469–4490. Retrieved from http://web.engr.oregonstate.edu/p%CC%83chiang/classes/ibm_t_j_watson.pdf

Harris, D., Keller, B., Karl, J. & Keller, S. (2010). A transregional model for near-threshold circuits with application to minimum-energy operation. *International conference on microelectronics (ICM), 2010*, 64–667.

Ho, R., Mai, K. & Horowitz, M. (2001). The future of wires. *Proceedings of the IEEE*, *89*(4), 490–4504.

Islam, A., Akram, M., Imran, A. & Hasan, M. (2010). Energy efficient and process tolerant full adder design in near threshold region using finfet. *2010 international symposium on electronic system design (ISED)*, 56–560.

Lohstroh, J., Seevinck, E. & de, J., Groot. (1983). Worst-case static noise margin criteria for logic circuits and their mathematical equivalence. *IEEE Journal of Solid-State Circuits*, *18*(6), 803–8807.

Mingoo, S., Chen, G., Hanson, S., Wieckowski, M., Blaauw, D. & Sylvester, D. (2011). Cas-fest 2010: mitigating variability in near-threshold computing. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, *1*(1), 42–449.

Nam, S. K., Taeho, K., Bowman, K., De, V. & Mudge, T. (2005). Total power-optimal pipelining and parallel processing under process variations in nanometer technology. *IEEE/ACM international conference on computer-aided design, 2005, ICCAD-2005*, 535–540.

Rabaey, J. M. (1996). *Digital integrated circuits : a design perspective / Jan M. Rabaey.* Upper Saddle River ; New Jersey : Prentice-Hall.

Sangwon, S., Dreslinski, R., Woh, M., Yongjun, P., Charkrabari, C., Mahlke, S., … Mudge, T. (2012). Process variation in near-threshold wide simd architectures. *2012 49th ACM/EDAC/IEEE design automation conference (DAC)*, 980–9987.

Sedra, A. S. & Smith, K. C. (2010). *Microelectronic circuits*. Oxford University Press.

Seevinck, E., List, F. & Lohstroh, J. (1987). Static-noise margin analysis of mos sram cells. *IEEE Journal of Solid-State Circuits*, *22*(5), 748–754.

Sharma, A. K. (2003). *Advanced semiconductor memories: architectures, designs, and applications*. Wiley-IEEE Press.

Smith, K., Wang, A. & Fujino, L. (2012). Through the looking glass: trend tracking for ISSCC 2012. *IEEE Solid-State Circuits Magazine*, *4*(1), 4–20.

Tsividis, Y. (2008). Eric Vittoz and the strong impact of weak inversion circuits. *Solid-State Circuits Newsletter, IEEE*, *13*(3), 56–558.

Turley, J. (2011). Let's get small, Intel's 22nm technology announcement is big, but not that big. Retrieved on 11. 12. 2012, from http://www.eejournal.com/archives/articles/20110511-small/

Wang, A., Calhoun, H., Benton & Chandrakasan, P., Anantha. (2006). *Sub-threshold design for ultra-low power systems*. Boston, MA : Springer Science Business Media, LLC.

Vittoz, E. & Fellrath, J. (1977). Cmos analog integrated circuits based on weak inversion operations. *Solid-State Circuits, IEEE Journal of*, *12*(3), 224–231.

Yeknami, A. F. (2008). Design and evaluation of a low-voltage, process-variation-tolerant sram cache in 90nm cmos technology. *Master's thesis, Electronic Devices, Dept. of Electrical Engineering at Linköpings Universitet*, 106. Retrieved from http://liu.diva-portal.org/smash/record.jsf?pid=diva2:18497%5C&searchId=null

Yu, P., Xin, Z., Huang, J., Muramatsu, A., Nomura, M., Hirairi, K., … Sakurai, T. (2010). Misleading energy and performance claims in sub/near threshold digital systems. *Ieee/acm international conference on computer-aided design (ICCAD), 2010*, 625–6631.