Accurate Power Analysis of Integrated CMOS Circuits on Gate Level

Dissertation

zur Erlangung des Grades eines
Doktors der Ingenieurwissenschaften
am Fachbereich Informatik der
Carl von Ossietzky Universität Oldenburg

von

Dirk Rabe

Erstgutachter:	Prof. Dr.-Ing. Wolfgang Nebel
Zweitgutachter:	Prof. Dr.-Ing. Klaus D. Müller-Glaser
Tag der Einreichung:	20.04.2001
Tag der Disputation:	03.07.2001

© 2001 by the author

Accurate Power Analysis of Integrated CMOS Circuits on Gate Level

Dissertation

zur Erlangung des Grades eines
Doktors der Ingenieurwissenschaften
am Fachbereich Informatik der
Carl von Ossietzky Universität Oldenburg

von

Dirk Rabe

Erstgutachter:	Prof. Dr.-Ing. Wolfgang Nebel
Zweitgutachter:	Prof. Dr.-Ing. Klaus D. Müller-Glaser
Tag der Einreichung:	20.04.2001
Tag der Disputation:	03.07.2001

© 2001 by the author

Acknowledgement

The ideas, which are presented in this thesis, are the results of my research at the Carl von Ossietzky University Oldenburg from 1993 until 1998.

I am grateful to my supervisor, Prof. Dr.-Ing. Wolfgang Nebel, for enabling and supporting my research in an inspiring environment.

I would like to thank Prof. Dr.-Ing. Klaus D. Müller-Glaser for his effort to examine this dissertation.

Thanks are due to my former colleagues of the low power group Gerd Jochens, Lars Kruse and Bernd Timmermann for inspiring discussions.

I am thankful for the great work of the former students Boris Fiuczynsky, Malte Gaudig, Christiane Kill, Lars Kruse, Andree Martens, Gero Vögel and Andreas Welslau, who did numerous simulation runs and part of the implementation of GliPS and OCHATO.

I would also like to take the opportunity to thank my colleague Till Winteler for reviewing the manuscript.

Table of Contents

1 Introduction 1

1.1 Motivation 1
1.2 Overview of the scientific contribution 5
1.3 Structure of thesis 6

2 Trends in Microelectronics 7

2.1 Impact of technological advances on performance and power consumption 8
2.1.1 Constant electrical field scaling 9
2.1.2 Non constant electrical field scaling 14
2.1.3 Comparative impact of scaling on power consumption 15
2.1.4 Impact of silicon on insulator technologies on power consumption 19
2.2 Advances in battery technologies 19

3 Basics 21

3.1 Power consumption of standard cell CMOS designs 21
3.2 Signal modelling in digital circuits 23
3.2.1 Modelling of single transitions 25
3.2.2 Colliding and non-monotonous signal changes 31
3.2.3 Logic criteria for glitch generation and propagation 34
3.2.4 Dynamic glitch properties 40
3.3 Power consumption in CMOS circuits 49
3.4 Static power consumption in CMOS circuits 49
3.4.1 Leakage power consumption 49
3.4.2 Non ideal input voltages 51
3.4.3 Signal conflicts 52
3.4.4 Wired AND/OR topologies 53
3.5 Dynamic power consumption 53
3.5.1 Determination of capacitances 55
3.5.2 Capacitive power consumption 61
3.5.3 Short circuit power consumption 63

4 State of the Art 75

4.1 Gate level power analysis 76

4.1.1 Simulation with application specific pattern 76
4.1.2 Exhaustive simulation 76
4.1.3 Stochastic simulation 76
4.1.4 Statistical simulation 77

4.2 Simulation of delays 78

4.2.1 Zero delay model 78
4.2.2 Unit delay model 78
4.2.3 Transport delay model 79
4.2.4 Inertial delay model 80
4.2.5 Enhanced glitch models 81

4.3 Conclusions 91

5 The new Glitch-Model 93

5.1 Derivation 93

5.2 Evaluation of the new model 99

6 Gate Level Power Model 105

7 Simulation Algorithm and Implementation 109

7.1 Interfaces of GliPS 109

7.2 General simulation algorithm 110

7.2.1 Logic value system 110
7.2.2 Event driven simulation algorithm 111
7.2.3 Glitch handling 114
7.2.4 Impact of input event processing order on detection of unnecessary transitions 116

7.3 Control of the simulation 117

7.4 Library characterization 118

8 Evaluation 119

8.1 Practical results 120

8.2 Conclusions 126

9 Summary 127

10 References 129

11 Glossary 135

11.1 Terms 135
11.2 Expressions 139

Appendix A: Power Gain Budget 143

Appendix B: Personal Record 145

1 Introduction

1.1 Motivation

Microelectronic products are the essential key for products of much higher economic value, which have an increasing impact on everybody's life [VDE96]. The market driven progress of microelectronics in terms of increasing functionality per chip (respectively circuit complexity) and at the same time decreasing its costs is higher than in any other industrial field. The cost reduction per transistor is 25-30% per year throughout semiconductor industry's history [Sema97]. The total maximum number of transistors per chip will increase from 11 million (for MPUs[†]) in 1998 up to 1.4 billion in 2012 [Sema97] (38% per year) for leading-edge circuits mainly by

- decreasing feature sizes (10%-15% per year [Sema97,Bako90[††],Inte98[†††]]) and
- increasing die area (6% per year for MPUs (12% for DRAMs) [Sema97] and in the past even 19% per year [Bako90][††]).

In conjunction with the technological advances the factory and technology development costs are continuing to escalate [Chat93,Sema97]. These challenges afford a high degree of innovation for technical production and all fields of CAD (Computer Aided Design). On the one hand abstraction is needed to enable handling the large circuit complexities within the design process and on the other hand the number of low-level effects, which significantly influence chip-characteristics (like performance, power consumption and functionality), is increasing.

Within this thesis basically two topics are addressed:

- accurate *digital* gate level simulation and
- accurate gate level power calculation.

To be accurate in both topics, the simulation of the circuit behaviour has to be as close to the actual silicon behaviour as possible. Therefore an adequate delay model is required. Traditional delay models rarely fulfil this demand. Therefore a new delay model has been invented, which is as accurate as fast transistor level simulators (e.g. EPIC's PowerMill[††††]) but features more than one order of magnitude higher simulation performance.

The need for accurate simulation is obvious to ensure correct silicon behaviour. Besides this topic the need to fabricate and design ICs for low power has become an important topic within the past few years for CAD and technology. The motivation for this hot topic is now discussed in detail.

The power consumption per chip is continuing to increase for future technologies even though the supply voltage and feature sizes will be further scaled down (confer Chapter 2). For marketing, environmental and reliability reasons a low power consumption is gaining importance within a large number of application domains, e.g.:

- Portable applications: The maximum time of operation, during which portable applications operate independently from external power supplies, is limited by its energy consumption and battery capacities (respectively photoelectric cells). The amount of energy, which can be supplied to an application by batteries, is limited by the user requirements in terms of battery

---

[†] Microprocessor Units

[††] reference numbers are taken from years 1959 and 1983

[†††] reference numbers are taken from years 1972 and 1995

[††††] within comparisons EPIC's PowerMill version 5.1 has been used

size, weight and price. Hence the application's power consumption is important. Examples for battery powered applications are PDAs[†], notebooks, mobile phones, hearing aids, wrist watches and pacemakers [Nebe97,Chat93].

- High performance applications are typically powered by external power supplies, which are only limited due to environmental reasons and - in case of a power supply brake down - by battery capabilities. The electrical energy is turned into heat, which has to be transferred to the ambient. As a consequence cooling problems arise which dramatically influence the packaging and its costs (including heat sinks). The noise of heat sinks (e.g. forced air) also has a large impact on the user acceptance. Important examples for high performance applications are microprocessors and telecommunication applications (e.g. ATM switches).
- For contactless chip card ICs the energy needs to be transferred via electromagnetic fields onto the chip. As a consequence a low power consumption increases the maximum distance between the chip and the transmitter, which is an important marketing issue within that application field.

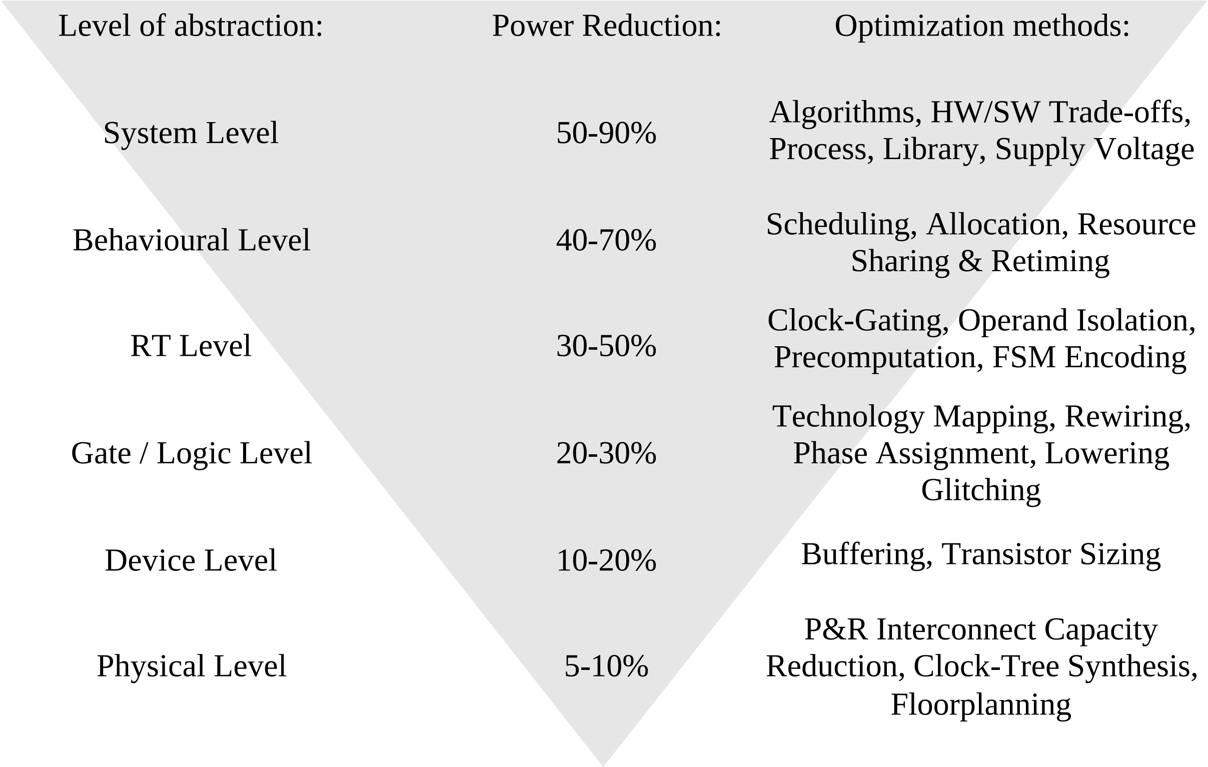
Besides the power dissipation itself, the respective power needs to be supplied to the circuit. The supply network on chip and on the boards need to cope with the resulting high current, which can be in the range of up to several 100A according to the projections for the next decade in Chapter 2.1.3.

The power consumption of an application can be reduced by technology improvements and/or design decisions for low power [Chan92, Chan95, Cha295, Cha395, Sing95, Alid94, Cong94, Tiwa93, Nebe97]. Both ways require a certain amount of financial investment. Utilizing design for low power has the highest return on investment (ROI) [Sing95], because changing a technology is typically very expensive and is rather a long term goal. The potential impact of design decisions on power consumption at different levels of abstraction is given in Figure 1 (for further expert's opinions about the power gain budget, refer to appendix A). It is obvious, that design decisions at high levels of abstraction have a larger impact on power consumption than design decisions on low levels (similar to other constraints like area and circuit performance). Even though potential power savings on high level of abstraction are more promising, the savings on low levels (gate-level and below) can be exploited much easier by *push button* tools than on higher levels. For a wide range of future low power applications it is mandatory, that the high demands on lowering energy consumption require the exploitation of all potential technological savings <u>and</u> all possibilities at all levels of design (confer Chapter 2).

Within design optimization respectively synthesis different design alternatives need to be validated. For validation a cost function is needed to trade off different design alternatives. This cost function typically contains variables like area, performance and power consumption. The requirements of the cost function's accuracy is closely related to the possible optimization gains, because it has to be ensured, that a certain design decision is better than an other tradeoff. I.e., for large potential gains the different design alternatives are more likely to be spread further apart from each other in the design space and hence the inevitably best solution can be determined even if the accuracy is relatively low. On gate level the possible power savings (20-30%) are lower than on RT[††] level and above. Hence for the gate level power estimate a minimum accuracy in the range of approximately 5-10% has to be guaranteed for evaluating different alternatives.

[†] Personal Digital Assistants

[††] Register Transfer level



Level of abstraction:	Power Reduction:	Optimization methods:
System Level	50-90%	Algorithms, HW/SW Trade-offs, Process, Library, Supply Voltage
Behavioural Level	40-70%	Scheduling, Allocation, Resource Sharing & Retiming
RT Level	30-50%	Clock-Gating, Operand Isolation, Precomputation, FSM Encoding
Gate / Logic Level	20-30%	Technology Mapping, Rewiring, Phase Assignment, Lowering Glitching
Device Level	10-20%	Buffering, Transistor Sizing
Physical Level	5-10%	P&R Interconnect Capacity Reduction, Clock-Tree Synthesis, Floorplanning

Figure 1: Possible power savings at different levels of abstraction: Data provided by Synopsys Inc. within 1998-low power training course material.

Proceeding in the design process from system to layout level, more and more details get available, which enable an increasingly accurate power calculation. However, this increasing accuracy typically has to be paid with a decrease in calculation performance. As a consequence - especially on the lower levels of abstraction - it is important to trade off accuracy against performance by considering the main important effects. A fast calculation of the cost function within circuit optimization respectively synthesis also enables a higher design space exploration within a given (commonly limited) time.

A large number of power estimation and modelling approaches have been proposed on different levels of abstraction: circuit-level [Deng94], gate-level [Burc93, VanO93, Saxe97, Geor94, Eise95, Ghos92, Burc88, Najm91, Metr95, Melc91], RT-/architectural/-behavioural level [Sven94, Powe92, Land93, Land95, Beni96, Masa92, Cha395, Mehr94, Bogl98, Wu98].

The cost function for low power does not necessarily have to contain absolute power values. Especially on high levels of abstraction absolute power numbers are hard to obtain due to missing information about the final implementation. Only if details about the design process towards the silicon implementation (synthesis process and target technology) respectively software implementation (target processor and algorithms) are considered in advance, absolute power numbers can be estimated.

Besides the validation of certain design solutions, tools on lower levels of abstraction are needed for characterizing higher level modules. For RT-module characterization tools on circuit- or gate-level can be used. Even though this characterization has to be done only once for a module library, the usage of SPICE-like tools is commonly not feasible due to the module's high complexity and the large number of stimuli, which need to be analysed. On the other hand a poor accuracy during the characterization process will decrease the simulation accuracy on higher levels.

Within this thesis highly accurate power evaluation on gate-level is addressed, which is applicable to module-characterization and full chip analysis of cell based semicustom designs. One way to achieve this high accuracy is to put high emphasis on power modelling, which will be discussed in detail. Power consumption is closely related to a circuit's net activities. The net activities are application specific and hence a power number is always a function of the circuit *and* the application specific stimuli. Within combinatorial parts of a circuit, signals may multiply switch within one computation cycle due to different path delays from the inputs (primary inputs and outputs of sequential cells) to internal circuit nodes and the outputs. Multiple transitions can be distinguished in hazards and glitches (refer to Definition 4 and 5). A few definitions, which are important in this context, are given next:

Definition 1: ***Transition:***
A transition T describes the process of a monotonously changing signal s. I.e., rising and falling transitions are distinguished[†]*. The changing signal is typically represented by a voltage in the domain of integrated CMOS circuits. The derivation of a falling (rising) transition's voltage waveform is lower (larger) than zero at the beginning of the transition and remains lower (larger) or equal than zero until its end is reached. The voltage at the end of the transition is either* V_{SS} *(*V_{DD}*) or an intermediate voltage in case of a glitch. Hence formally either one of the following two properties need to be fulfilled for a transition:*

$$\left.\frac{d}{dt}V(t)\right|_{t\,=\,t_{Start}} > 0 \wedge \left.\frac{d}{dt}V(t)\right|_{t_{End}\geq t\geq t_{Start}} \geq 0 \qquad or$$

$$\left.\frac{d}{dt}V(t)\right|_{t\,=\,t_{Start}} < 0 \wedge \left.\frac{d}{dt}V(t)\right|_{t_{End}\geq t\geq t_{Start}} \leq 0$$

A voltage range is typically associated with a logic value (e.g. 0,1,X).

Definition 2: ***Complete, incomplete and partial transition:***
If a signal's voltage is monotonously changing from V_{DD} *to* V_{SS} *or vice versa, a* ***complete*** *transition has occurred. In all other cases an* ***incomplete*** *respectively* ***partial*** *transition has occurred*[†]*. The potentials* V_{DD} *and* V_{SS} *are typically given by the driving gate's supply voltage.*

Definition 3: ***Useful and useless transition:***
If an odd number of signal transitions occurs within one computational cycle $[t_0,t_e]$ $(|V_s(t_0)-V_s(t_e)|=V_{DD})$*, one* ***useful*** *transition has occurred within this period. All additional transitions are* ***useless****.*
If an even number of signal transitions occurs within one computational cycle $(V_s(t_0)=V_s(t_e))$*, all transitions within this period are* ***useless****.*

Definition 4: ***Glitch:***
A ***glitch*** *consists of a pair of at least two partial signal transitions. Three or more consecutive partial transitions, which do neither reach* V_{DD} *nor*

[†] over- and undershots are neglected here

V_{SS} in between, define a ***dynamic glitch****.*

Definition 5: **Hazard:**
A pair of useless complete transitions within one computational cycle $[t_0,t_e]$ is defined as a ***hazard****. Three or more consecutive complete transitions define a* ***dynamic hazard****.*

Definition 6: **Event:**
An event is a change between two states, which belong to a well defined set of signal states. E.g., for Boolean signals a change from 0 to 1 and vice versa are possible events. In addition to voltage level dependent state definitions, driving strengths are commonly also considered.

Definition 7: **Net activity:**
The net activity α of a signal s is the average number of transitions per clock cycle (typically equivalent to computational cycle). Partial transitions are considered fractionally according to their voltage swing ΔV_s:

$$\alpha_s = \frac{1}{V_{DD} \cdot f} \cdot \lim_{\tau \to \infty} \frac{\sum_{\text{Transitions during the period } \tau} \Delta V_s}{\tau}$$

To properly estimate net-activity and power consumption it is inevitable to use accurate delay models. Conventional gate-level delay models (e.g. transport or inertial delay model) can not handle incomplete transitions accurately enough for all classes of circuits.

Unlike other cost functions, the calculation of a circuit’s power consumption requires the analysis of the dynamic circuit behaviour, which cannot be *accurately* accomplished with static algorithms.

1.2 Overview of the scientific contribution

Within my research I have been focusing on two major topics:
- accurate *digital* gate level simulation and
- accurate gate level power calculation.

The main stream of gate level power estimation research focused on dealing with simulation pattern complexity. This is important, because power consumption heavily depends on signal transitions of all circuit nodes. The signal transitions are caused by the external stimulation pattern (vectors). The theoretical number of different stimulation pattern of a FSM[†] is 4^n (n=number of a circuit’s top level pins plus the number of Flip Flops) and each pattern has a different probability to occur. The main stream research hence focused on pattern compression by stochastic and statitistical simulation. Only very few researchers have focused in detail on the impact of the delay model on power consumption to consider the impact of glitches (incomplete transitions) [Eise95, Metr95, Melc91], which is also important for detailed circuit validation. In this work the first fundamentally sophisticated model is presented to accurately and efficiently consider incomplete transitions, which have been found to be one of the main errors for power calculation of circuits with moderate to high circuit depths. The new simulator

---

[†] Finite State Machine

GliPS (Glitch Power Simulator) was implemented to exemplify the high accuracy and simulation performance of the new delay model. For the required characterization an automatic characterization procedure has been developed and implemented into the tool OCHATO (OFFIS Characterization Tool). Besides the glitch and timing information the power characterization is also taken care of in OCHATO.

1.3 Structure of thesis

Chapter 2 is devoted to trends in microelectronics and their impact on circuit performance and power consumption. In Chapter 3 the basics about gate level (power) simulation are dealt with. High emphasis is put on abstracting basic properties for digital simulation from circuit level CMOS characteristics. These observations are used in Chapter 4 to evaluate existing state of the art models. Besides these signal modelling issues some basic power estimation approaches on gate level are shortly dealt with (stochastic and statistical simulation). The new delay model is derived from basic CMOS characteristics (confer Chapter 3) in Chapter 5 and compared to the models, which have been presented in Chapter 4 in terms of accuracy. Besides a good delay model accurate power estimation requires a good power model, which is dealt with in Chapter 6. A new power simulation tool, which is based on the new delay model and the accurate characterization data, has been implemented (Chapter 7) and compared to other commercially available tools (Chapter 8). The last chapter contains a summary and an outline.

2 Trends in Microelectronics

The evolution of microelectronic technologies and products is the main challenge for tools and design methodologies to cope with. Hence it is very important to clearly analyse the needs of current and future integrated circuits to address the right issues within research. I am focusing on the impact of increasing transistor count per die and its impact on power consumption and performance in this chapter, to give further motivation for my research activities. To get a better feeling for the impact of scaling, basic equations and relations are first introduced. These basic relations are used together with data, which is provided by the publications [Sema97,Dava95], to discuss future trends. All trend projections clearly point out, that future design methodologies and tools will have to cope with increasing complexity and increasing power consumption problems. Besides these problems, a huge number of further problems will have to be solved on the way to the new nanometer generations. In this thesis the power consumption - more specifically its efficient calculation - and delay modelling are focused on, which is the essential key to evaluate different design alternatives.

The market driven progress in technology enables a doubling of transistor count per manufactured die every 18-26 months. Gorden E. Moore† made this observation in 1965 while preparing a speech by graphing data of past year's trends, just four years after the first planar integrated circuit was discovered. Moore's observation, now known as Moore's Law, describes a continuing trend, which is still remarkably accurate today (Figure 2) and which will continue until fundamental physical limits will be reached. In 1965 Moore did not really expect this law to be still true some 30 years later, but today he is confident, that it will be true for another 20 years [Moor97]. The period for doubling of transistor counts is approximately

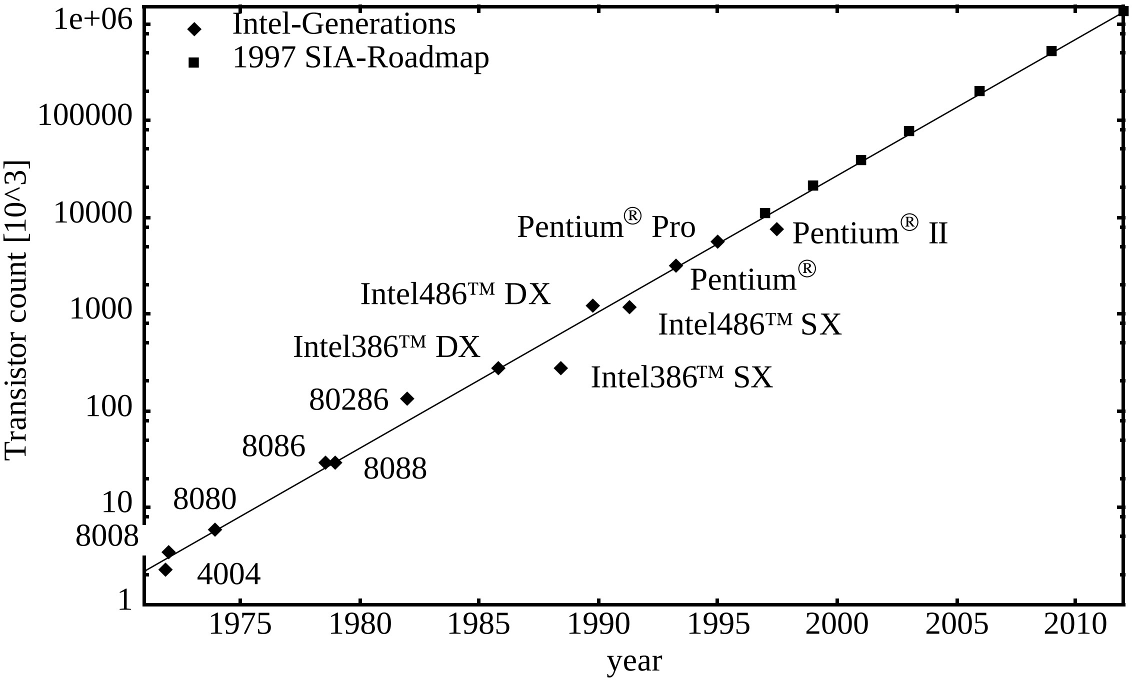


Figure 2: Transistor count over Time of existing intel-microprocessors and Future Trends [Inte98,Sema97].

---

† Dr. Gordon E. Moore co-founded Intel in 1968 and is Chairman Emeritus of Intel Cooperation today.

26 months (respectively one order of magnitude every 10 years - confer Figure 2) and the number of bits on a single DRAM die doubles approximately every 18 months [Sema97].

The main keys for these technological advances are improvements of circuit patterning technologies, which enable decreasing minimum feature-sizes (Figure 3), and increasing die areas.

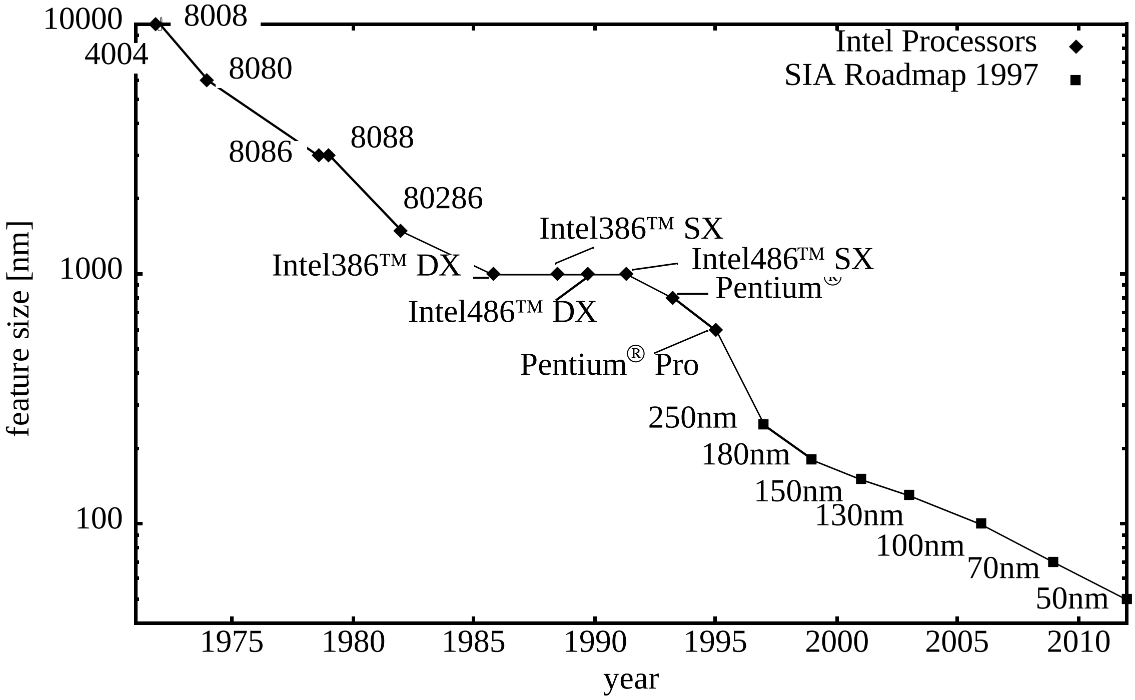


Figure 3: Minimum feature sizes over time of existing intel-microprocessors and Future Trends [Inte98,Sema97].

The shorter period for doubling the complexity of a single DRAM chip is achieved by increasing the die area much more aggressively than for MPUs and ASICs. The impact of reducing minimum feature sizes and increasing die areas on performance (i.e. clock frequency) and power consumption is discussed in the following Subchapter 2.1.

2.1 Impact of technological advances on performance and power consumption

The impact of scaling the minimum feature size by 1/S (S>1) and the die's edge by S_C (S_C>1) is discussed here in a simple way (i.e., quantitative short channel effects are only partly considered) [Dava95,Bako90]. The main purpose is to exemplify the impact of scaling transistor's, interconnect's and die's dimensions on performance and power. These results may be slightly degraded by other effects [Bako90].

A CMOS circuit's power consumption strongly depends on its supply voltage. The supply voltage has been kept constant with scaling technologies in the past. Today and in the future the supply voltage will be scaled in conjunction with the transistor's dimensions. However, the supply voltage will possibly be scaled less aggressively than S. As a consequence different scaling scenarios are distinguished in the Subchapters 2.1.1 and 2.1.2.

2.1.1 Constant electrical field scaling

It is assumed first, that an existing design is simply scaled down (i.e. without exploiting the additionally available area to integrate further functionality on the same die). All transistor dimensions (width, length and thickness of gate-oxide) are supposed to be scaled by 1/S (confer Figure 4). As a consequence the area needed for a single transistor is scaled by $1/S^2$ and the transistor density (transistors per area) is increased by S^2. The supply-voltage V_{DD} and threshold-voltages V_{TN}, V_{TP} are also scaled down by 1/S to keep the scalar value of the electrical field E in the gate-oxide constant. This way of scaling is referred to as constant electric field scaling (CE scaling). CE scaling also helps avoiding reliability degradation. The electrical field pattern are preserved within the silicon substrate by increasing the impurity doping with the factor S. The gate capacity C_g and the local interconnect capacity $C_{intlocal}$[†] can be expressed as given in Equation 1.

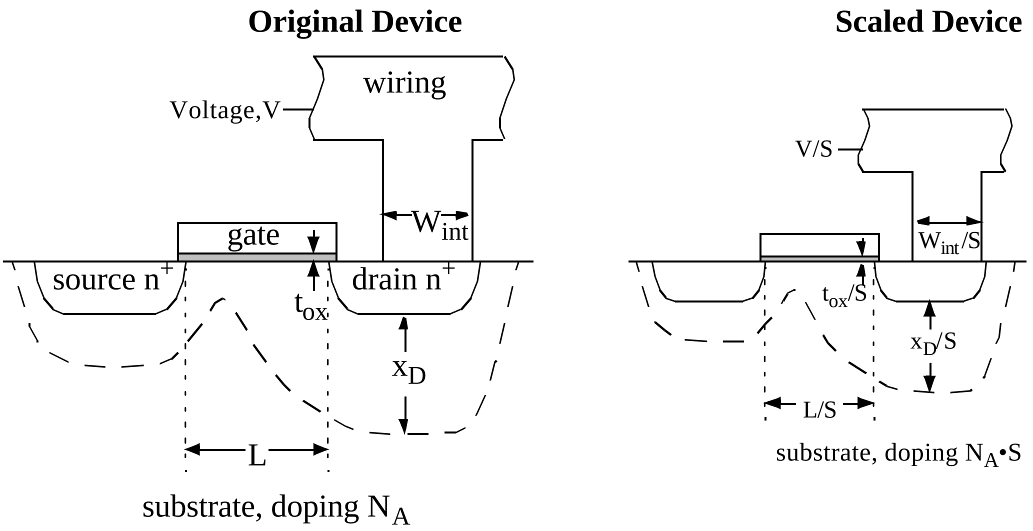


Figure 4: Principles of constant-electric-field scaling for MOS transistors and integrated circuits [Dava95].

$$
\begin{aligned}
C_g &= \varepsilon \cdot \frac{A}{t_{gox}} \sim \frac{1/S^2}{1/S} = 1/S \\
C_{intlocal} &= \varepsilon_{ox} \cdot \frac{W_{int} \cdot l_{int}}{t_{ox}} \sim 1 \cdot \frac{1/S \cdot 1/S}{1/S} = 1/S
\end{aligned}
\tag{1}
$$

The scaling of the drain-source current I_{DS} is given in Equation 2 for saturation:

$$
I_{DS} = \frac{C_g \cdot W \cdot \mu}{2 \cdot l_{tr} \cdot A} \cdot (V_{GS} - V_T)^2 \sim \frac{\frac{1}{S} \cdot \frac{1}{S} \cdot 1}{\frac{1}{S} \cdot \frac{1}{S^2}} \cdot \frac{1}{S^2} = \frac{1}{S}\ ^{\dagger\dagger}
\tag{2}
$$

[†] In this first scenario all interconnections are scaled by 1/S. Later the impact of increasing the die area on the interconnection length will lead to a distinction of local and global interconnections.

[††] Especially I_{DS} is degraded due to short channel effects, which are not properly considered here. However, at least (further) velocity saturation is avoided, as the supply voltage is scaled by 1/S. [Bako90]

The transistor's on-resistance R_{tr} remains constant ($R_{tr} \sim V_{DD}/I_{DS} = 1$). As a consequence the gate-delay τ scales by 1/S ($\tau = R_{tr} \cdot (C_{intlocal} + C_{fanin}) \sim 1 \cdot 1/S = 1/S$) and the clock frequency can be increased by S ($f \sim 1/\tau$). This ideal performance scaling is only valid if the interconnect resistance is much smaller than the transistor's on-resistance. While the transistor's on-resistance is approximately independent from scaling, the interconnect resistance is scaled by S (confer Table 5).

In CMOS integrated circuits power is turned into heat while charging capacitances (confer Chapter 3). The power consumption can be calculated according to Equation 3, where f is the circuit's frequency, n is the number of nodes within the circuit, C_i is the capacitive load of node i (i.e. interconnect plus driven gate capacities, confer Figure 5) and α_i is the average number of signal transitions per clock cycle at node i (confer Definition 7):

$$P = \frac{1}{2} \cdot V_{DD}^2 \cdot f \cdot \sum_n C_i \cdot \alpha_i = \frac{1}{2} \cdot V_{DD}^2 \cdot f \cdot C_{eff} \sim \frac{1}{S^2} \cdot S \cdot \frac{1}{S} = \frac{1}{S^2} \qquad (3)$$

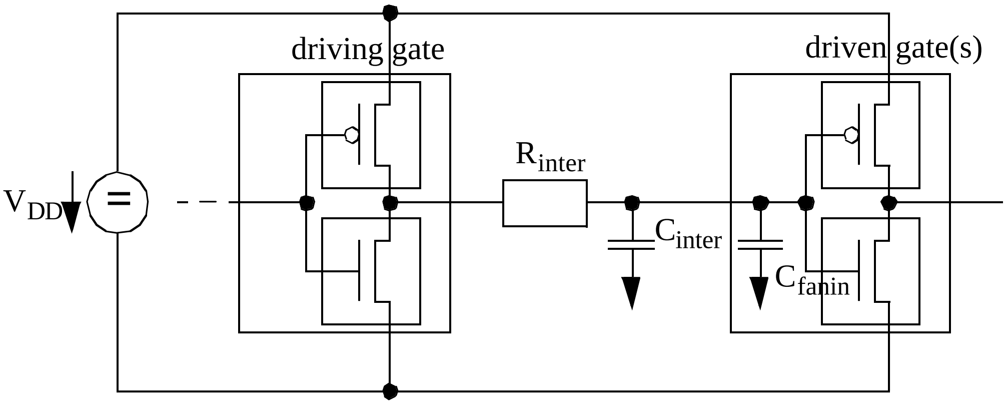


Figure 5: CMOS circuit situation.

$\sum C_i \cdot \alpha_i$ is defined as C_{eff} (effective -switching- capacitance). In total the power consumption is scaled by $1/S^2$ and the power-delay-product is even scaled by $1/S^3$. I.e., shrinking a design is not only attractive for decreasing the die area but also from the power consumption and performance point of view. The power consumption per area remains constant. I.e., if a circuit's feature sizes are scaled with 1/S, the circuit's complexity may be scaled by S^2 without increasing its power consumption. The scaling effects are summarized in Table 1.

Parameter	**Scaling Factor**
Dimensions (W, L, t_{gox})	*1/S*
Area per device A_{tr}	$1/S^2$
Voltages (V_{DD}, V_{TN}, V_{TP})	*1/S*
Electrical fields E (in gate oxide)	*1*
Gate capacity C_g	*1/S*
Drain-source current I_{DS}	*1/S*

Table 1: Impact of device scaling on power and delay (scaling of a given design).

Parameter	**Scaling Factor**
Transistor on-resistance R_{tr}	*1*
Gate Delay τ_g	$1/S$
Power consumption per gate P_g	$1/S^2$
Power consumption density P_g/A	*1*
Power-delay product per gate $P_g \cdot \tau_g$	$1/S^3$

Table 1: Impact of device scaling on power and delay (scaling of a given design).

Besides the reduction of feature sizes, which have been addressed so far in this chapter, technological improvements also allow to economically produce single chips with larger die-sizes. Taking S_C as the scaling factor of die edges, the die area is increased by S_C^2. I.e., when scaling of feature sizes and die sizes are considered, the maximum number of transistors on a single die scales with $S_C^2 \cdot S^2$. As the power consumption density is 1, the only contributor to increased power consumption is the die size scaling: $P \sim S_C^2$. These numbers are summarized in Table 2.

Parameter	**Scaling Factor**
Area per die - A_{global}	S_C^2
Number of transistors per die - N_{Total}	$S_C^2 \cdot S^2$
Total device capacity per die	$S_C^2 \cdot S$
Power consumption per die - P_{Total}	S_C^2

Table 2: Impact of device and die scaling on power and delay (the additionally available area is used).

For delay modelling it has been observed, that the impact of interconnects on total signal-delay is increasing [Bohr95,Sema97], because the growing number of transistors per chip require more and more routing resources, which are made available by increasing the number of metal layers by 0.75 per technology generation [Sema97]. The wire load’s contribution to the total fanout capacitance of a large standard cell block (>10mm^2) is increasing from 50% to 70% comparing a 350nm and a 150nm process [Veen98] (confer Table 3).

Technology	**ratio: wire load/fanin**
350nm	50/50
250nm	58/42
180nm	66/34
150nm	70/30

Table 3: Increasing interconnect dominance on delay[Veen98].

However, it is important to distinguish between local (i.e. short) and global (i.e. long) interconnections [Bako90,Dava95]. Assuming, that a chip is build up of a set of partitioned blocks, the gates within such a block are referred to as local interconnections. The different blocks are (typically) interconnected by global wires. The length of global interconnects grows with increasing die sizes and the local interconnection delays ($\approx R_{tr}C_{intlocal}$) are scaling down similar to the gate delays (assuming that the interconnect resistance is negligible).

The interconnect length, for which its resistance equals the NMOS on-resistance per square, will drop from 41mm for the 250nm generation (164000 λ[†]) to 7mm for the 50nm generation (140000 λ) [Sema97][††]. I.e., this interconnect length will only drop 15% (= 1-140000/164000) for local interconnects but 89% (= 1-7/(41•1.58[†††])) for global interconnects. The maximum interconnection length per gate[††††] will drop from 295µm (1180 λ) to 71µm (1420 λ). I.e., in average the situation won't become worse[†††††]. However, for global circuit communication across the chip the resistive impact will increase. The interconnect RC product characterizes the minimum delay if ideal drivers[††††††] are used. Besides the pure comparison of device and interconnection capacitances and resistances, a distributed RC wire model might become essential for increasing contributions of interconnection resistances. However, as long as the following condition is valid, the RC-modelling may be neglected [West93]:

$$l_{int} \ll \sqrt{\frac{2 \cdot \tau_g}{R_{int} \cdot C_{int}}} \tag{4}$$

From this equation the following conservative guidelines for ignoring RC wire delays can be derived [West93]:

Layer	**Maximum Length (λ)**
Metal 3	10000
Metal 2	8000
Metal 1	5000
Silicide	600
Polysilicon	200
Diffusion	60

Table 4: Guidelines for ignoring RC wire delays.

Hence only for global interconnections a RC model is needed. E.g. for clock lines the RC delay is of importance.

[†] λ is the minimum feature size of a technology

[††] these numbers include the change in interconnect aspect ratio and the decreasing effective resistance by choosing copper instead of aluminium

[†††] $S_C=1.58$

[††††] max. available routing resources divided by the number of gate-equivalents (1 gate-equ. = 4 transistors)

[†††††] besides the pure interconnection length the number of via-connections plays an increasing role for decreasing feature sizes and increasing number of metal layers (a deeper discussion is omitted here)

[††††††] ideal drivers are perfect switches: i.e. low on-resistance, no additional parasitic capacitances to be charged respectively discharged

Increasing numbers of metal layers are needed for two reasons: firstly, the metal pitch is scaled less than silicon structures [Bohr98] and secondly, higher functional integration on a single die requires more routing resources. For the global interconnections its resistance and capacities are increasing, resulting in a dramatically growing delay. It should be mentioned here, that within the higher layers of interconnections it is more likely to reduce its dimensions less than on the lower layers, in order to reduce the resistance of global interconnections and to increase its reliability.

The number of different clock frequencies on a chip will increase, in order to exploit more efficiently the possible performance margins.

The absolute minimum delay of a cross chip signal is limited by the speed of light. E.g., the minimum delay for a signal to propagate a distance of 6 cm is 0.2 ns. This delay will increase for a medium with a larger dielectric constant.

In Table 6 the interconnect scaling's impact on power and delay are given. The total device capacitance scales by $S_C^2 \cdot S$ and the total interconnection capacity scales by $S_C^2 \cdot S^2 \cdot S_l$. Hence the impact of interconnection capacity on power consumption is increasing. However, if the interconnection-capacity becomes dominant, the frequency will also be scaled by less than S and in total the power consumption scales with approximately S_C^2 for the CE model.

Parameter	**Local interconnects**	**Global interconnects**
Length of interconnects	$l_{intlocal} \sim 1/S$	$l_{intglobal} \sim S_C$
height, width, oxide-thickness	$1/S$	$1/S$
Interconnect capacity	$C_{intlocal} \sim 1/S$	$C_{intglobal} \sim S_C$
Interconnect resistance	$R_{intlocal} \sim S$	$R_{intglobal} \sim S_C \cdot S^2$
Interconnect Delay	$\tau_{intlocal} \sim 1/S^{‡}$	$\tau_{intglobal} \sim S_C^2 \cdot S^{2\,‡‡}$

Table 5: Interconnect scaling.

---

‡ assumption: $R_{intlocal} \ll R_{tr}$

‡‡ assumption: $R_{intglobal} \gg R_{tr}$

Parameter	**Scaling**
Number of metal layers	S_l
Total interconnect area	$S_C^2 \cdot S_l$
Total interconnect capacity $C_{inttotal}^{‡}$	$S_C^2 \cdot S \cdot S_l$
Average interconnect capacity per gate‡	S_l / S
Interconnect Delay τ_{intavg} for $R_{int} \ll R_{tr}^{‡}$	S_l / S
Total interconnect power consumption $P_{int}^{‡,‡‡}$	$S_C^2 \cdot S_l > Scaling(P_{int}) > S_C^2 \cdot S_l / S$

Table 6: Impact of interconnect scaling on power consumption.

The increasing impact of (global) interconnect delays may prevent the exploitation of possible maximum die sizes in the future. In [Flet94] it has been stated therefore, that the increasing number of transistors of (Intel-) microprocessors will therefore be achieved only by decreasing feature sizes while keeping the die size approximately constant.

2.1.2 Non constant electrical field scaling

So far CE scaling has been discussed. Within CE scaling all voltages are scaled together with all dimensions by 1/S. Even though it is not intended to go into the details of short channel effects, the scaling of subthreshold currents has a fundamental impact on the lower bounds of the voltages (V_{TH}, V_{DD}) and is therefore briefly introduced here.

The transistor’s behaviour within the subthreshold region does not scale in a linear way. The subthreshold leakage currents exponentially depend on the *absolute* threshold-voltage:

$$\left.\begin{array}{l} I_{DS} = K \cdot e^{\frac{V_{GS} - V_{TH}}{n \cdot V_T}} \cdot \left(1 - e^{-\frac{V_{DS}}{V_T}}\right) \\ V_T = 26\text{mV at room temperature } (25°\text{C}) \\ K, n\text{: function of technology} \\ \text{for } V_{DS} > 100mV \Rightarrow I_{DS} \approx K \cdot e^{\frac{V_{GS} - V_{TH}}{n \cdot V_T}} \end{array}\right\} \text{(for NMOS transistors)} \tag{5}$$

The voltage, which is required to drop the subthreshold current by one decade, is called the subthreshold slope S_{TH}. The values for the subthreshold slope are between 60 and 90mV at room temperature [Chan95]. A practical lower limit for the threshold voltage is approximately 300mV [Dava95]. If the threshold voltage will be reduced below this voltage, the subthreshold currents will be of concern from the low power perspective.

For high speed application it might be of interest to increase the supply voltage as high as possible. This increase in circuit performance (1/τ) can be partly achieved, if the scalar value of the electrical field in the gate-oxide is raised by a factor of ε_S. I.e., the supply voltage is scaled by ε_S/S (confer Table 7,8), However, on the one hand due to velocity saturation of electrons (respectively holes for PMOS transistors) the performance gain is less than $\varepsilon_S \bullet S$ for high V_{DD} (short channel effect [Bako90]) and on the other hand the upper limit of supply voltage is given by reliability considerations. One important reliability issue are hot electrons. If electrons gain sufficient dynamic energy within the transistor channel, they can overcome the interfacial barrier and get injected into the gate oxide, where they are trapped [Lebl93]. As the amount of trapped electrons increases with circuit life time, the threshold voltage shifts upwards, the channel resistance is increased and as a consequence the transistor’s performance is decreased.

[‡] It is not considered here, that the dielectric constant of the interlevel metal insulator will be reduced by a factor of approximately 2 within the next 15 years and that the aspect ratio (height/width) of interconnections will grow [Sema97]

[‡‡] The frequency, which is an important variable within the power formula (confer Equation 3), is determined by the sum of interconnect (~ S_l/S) and gate delay (~ $1/S$); as these two delay components do not scale the same way, a range is given for the total interconnect power consumption

Parameter	**CE Scaling Factor**	**NON CE Scaling Factor**
Dimensions (W, L, t_{gox})	$1/S$	$1/S$
Area per device A_{tr}	$1/S^2$	$1/S^2$
Voltages (V_{DD}, V_{TN}, V_{TP})	$1/S$	ε_S/S
Electrical fields E (in gate oxide)	1	ε_S
Gate capacity C_g	$1/S$	$1/S$
Drain-source current I_{DS}	$1/S$	ε_S/S
Transistor on-resistance R_{tr}	1	$1/\varepsilon_S$
Gate Delay τ_g	$1/S$	$1/(\varepsilon_S \bullet S)$
Power consumption per gate P_g	$1/S^2$	ε_S^2/S^2
Power consumption density P_g/A	1	ε_S^2
Power-delay product per gate $P_g \bullet \tau_g$	$1/S^3$	ε_S/S^3

Table 7: Impact of device scaling on power and delay (scaling of a given design).

Parameter	**CE Scaling Factor**	**NON CE Scaling Factor**
Area per die - A_{global}	S_C^2	S_C^2
Number of transistors per die - N_{Total}	$S_C^2 \cdot S^2$	$S_C^2 \cdot S^2$
Total device capacity per die	$S_C^2 \cdot S$	$S_C^2 \cdot S$
Power consumption per die - P_{Total}	S_C^2	$\varepsilon_S{}^2 \cdot S_C^2$

Table 8: Impact of device and die scaling on power and delay (the additionally available area is used).

In [Dava95] it is predicted, that - comparing the 70nm and the 900nm technologies - the power density will increase by a factor of 3.7 for the high performance and 2.0 for the low power scenario. The choice of supply voltage and threshold voltages is a major reason for the lower increase of power consumption for low power circuits.

2.1.3 Comparative impact of scaling on power consumption

Within this subchapter different scaling scenarios are compared. In general high performance and low power applications are distinguished. Scaling and physical data is either taken or derived from [Sema97,Dava95]. The following scaling scenarios are distinguished:

- CE scaling: the electrical field in the gate oxide is kept constant at the value given in [Sema97] for 1997[†] (confer Chapter 2.1.1, Table 1,2), the parameters S, S_C and the electrical values for 1997 are taken respectively are derived from [Sema97],

---

[†] The minimum feature size is 250nm in 1997

- NON CE scaling: the electrical field in the gate oxide is not kept constant, i.e. $\varepsilon_S \neq 1$ (confer Chapter 2.1.2, Table 7,8), the parameters S, S_C, ε_S and the electrical values for 1997 are taken respectively are derived from [Sema97]. The following scenarios are further distinguished:
    - Scaling according to Table 8: the degradation of the chip frequency and the effective capacitances C_{eff} due to the increasing impact of interconnect delays compensate each other within power consumption Equation 3 ($P \sim f \bullet C_{eff} \sim 1/(\varepsilon_S \bullet S^2)$) (short channel effects - e.g. velocity saturation - are neglected),
    - the clock frequency scaling is also taken from [Sema97 - Table 43] (including further degradation due to short channel effects),
- maximum power consumption data directly predicted by [Sema97],
- data provided by [Dava95]: for power trends the relative power density data from [Dava95] is multiplied with the increase in die area [Sema97] and the absolute maximum power consumption in 1997 [Sema97].

Within the following discussion and figures low power and high performance applications are distinguished.

In Figure 6 and 7 it is illustrated, that the supply voltage will continue to decrease in conjunction with the minimum feature size[†]. However, for the non CE scaling scenarios the decrease of supply voltage is less aggressive than the gate oxide thickness, which results in increasing electrical fields (for decreasing feature sizes) in total (even though the curves are not strictly monotonous). Within [Sema97] the supply voltage is considered to drop as low as 0.5V with a large increase in subthreshold currents, which are not taken into account within the following power figures (Figure 8 and 9). In [Dava95] the lower limit for the supply voltage is 1V. Within the low power scenario, performance ($\sim E_g$) is traded against power consumption. As a consequence the electrical field is larger for the high performance scenario. Further means for decreasing power consumption are design and lower area (~C). Practically, for MPUs the area reduction was achieved by lagging the main stream market and switching to the next generation technology [Flet94].

In [Dava95] it is stated, that the main limiter for gate electric field are defect density requirements rather than tunnelling effects. The upper limit was therefore projected to be 500 MV/m. The upper limit projections in [Sema97] are considerably higher. The electrical field is already 555 MV/m for today’s high performance applications.

The trend of increasing power consumption per die will continue for decreasing feature sizes within all above mentioned scenarios (confer Figure 8 and 9). The main contributors are the increasing die sizes ($S_C > 1$) and the change of the electrical fields ($P \sim \varepsilon_S^2$) (confer Table 7,8).

For high performance applications the maximum power consumption per die may increase within the next 15 years from 70W today (e.g. the Alpha 21264 consumes 60W for a 350nm-technology) to 174W-400W, if today’s design style is continued. Such a high power consumption (400W) would require an average supply current ($I = P/V_{DD} \sim \varepsilon_S \cdot S_C^2 \cdot S$) of 525A (in comparison to 28A today), which will lead to severe voltage-drop problems on the power rails.

[†] Please note, that the small feature sizes, which will be realized in the future, are printed on the left side of the diagram. A corresponding axis with the year of introducing the respective feature size would increase from the right to the left hand side of the diagram.

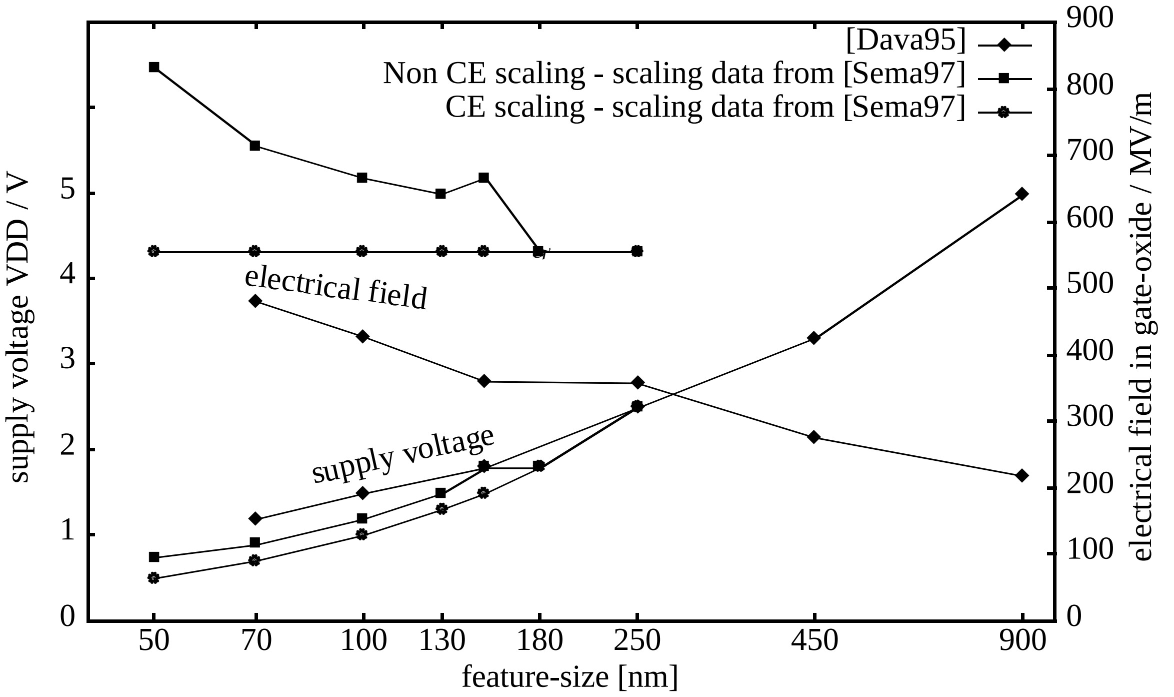


Figure 6: Scaling of supply voltage and electrical field in gate oxide for high performance applications [Sema97,Dava95].

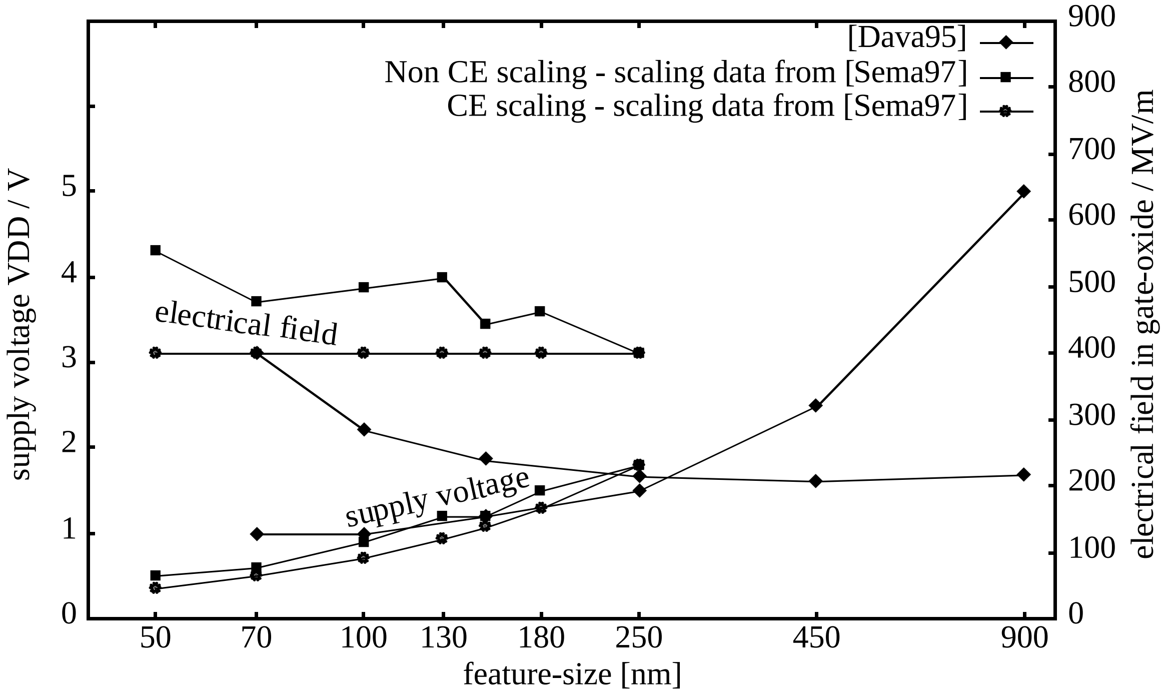


Figure 7: Scaling of supply voltage and electrical field in gate oxide for low power applications [Sema97,Dava95].

Within the 400W scenario (confer Figure 8) important short channel effects (e.g. velocity saturation) are not considered for the clock frequency. Taking the frequency values, which are provided by [Sema97 - Table 43], a somewhat lower and more realistic outline is obtained for the 50nm-technology (thick line in Figure 8 - 270W for the year 2012). The high power consumption results in significant problems in the domain of thermal management. Today’s solutions

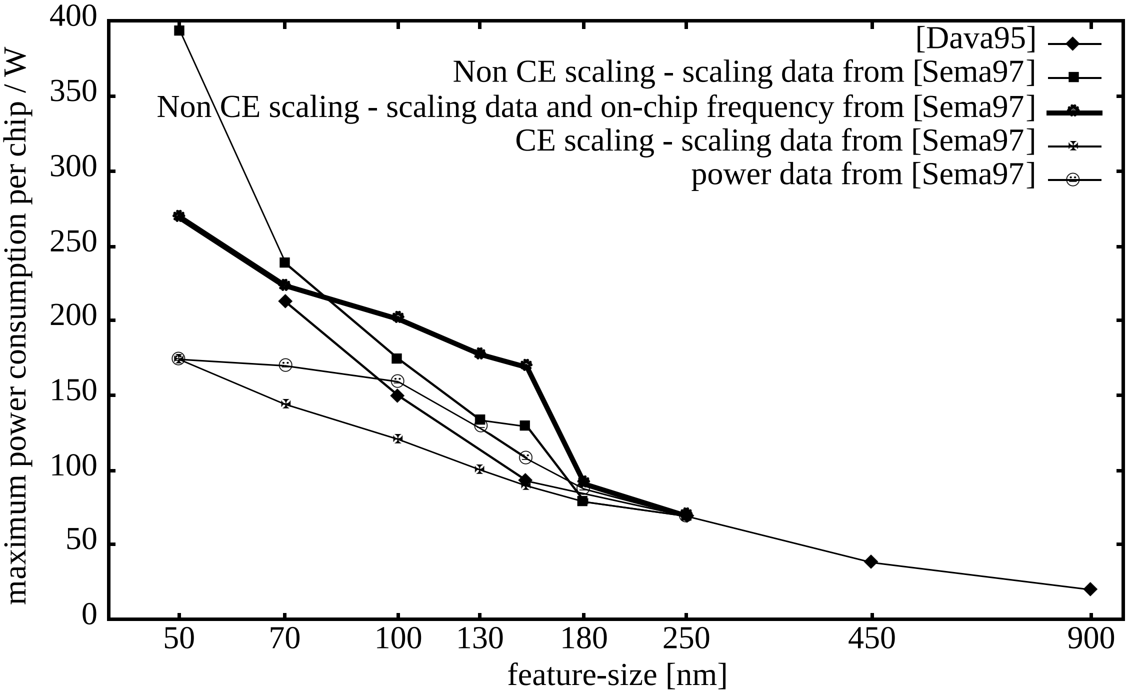


Figure 8: Impact of scaling on power consumption for high performance applications [Sema97,Dava95].

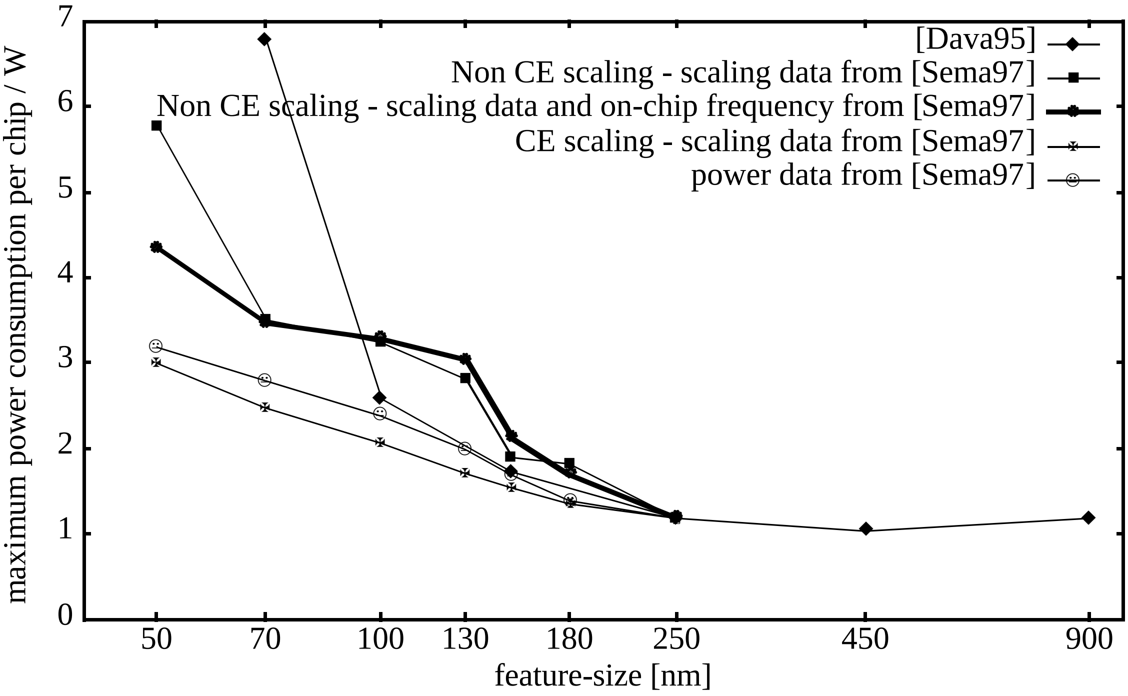


Figure 9: Impact of scaling on power consumption for low power applications [Sema97,Dava95].

are commonly based on forced air cooling. Existing technology solutions in cooling and heat sink design could become insufficient beyond 50 Watts per chip in applications where air cooling capabilities are limited, such as acoustic noise limits. Significant development and innovations will be needed for many applications in the high performance market. For a power dissipation in the range of 60-70W hot spots are of concern. It is expected, that at approximately 110-120 Watts per chip major innovations and solutions will be needed for cooling, as the heat sink size will become intolerable [Sema97].

The power consumption for low power applications (i.e. typically hand-held applications) is also increasing for decreasing feature sizes as illustrated in Figure 9. In [Dava95] the change in power consumption for lowering the minimum feature size from 100nm to 70nm is quite high because the supply voltage is not lowered below 1V. The maximum power consumption per die may increase within the next 15 years from 1.2W today to 3 to 5.8W, if today’s design style is continued. The advances in battery technology will hardly keep pace with the increase in power consumption (confer Chapter 2.2). Considering that the demands on long time battery operation for portable applications is growing, it is desirable to extend the time of operation for portable applications.

2.1.4 Impact of silicon on insulator technologies on power consumption

In common bulk technologies transistors are build into the main substrate. Within silicon on insulator (SOI) technologies these structures are grown on an isolating layer. As a consequence the transistors don’t have bulk connections. The SOI technology has the following physical advantages:

- lower parasitic transistor capacities,
- reduction of the body effect,
- sharper subthreshold slopes S_{TH} (hence enabling lower threshold voltages).

These physical effects can be used to either increase circuit performance by 1.5x to 2x (without changing power consumption) or to decrease power consumption by more than 3x (without changing performance) compared to bulk technologies with the same minimum feature sizes [Dava95].

However, SOI has some technical drawbacks (e.g. availability of low cost wafers with low defect density, floating body effects [Dava95]), which fortunately are becoming significantly less for supply voltages below 2.5V. Consequently SOI will gain importance in the future. Recently IBM has announced, that they will soon start high volume production of SOI logic ICs.

As the main stream production is still based on bulk technologies, a deeper discussion of SOI technologies is omitted here.

2.2 Advances in battery technologies

For low power applications the time of battery operation is an important marketing issue. In Chapter 2.1.3 the future trend of power consumption has been discussed. It is now investigated, how well battery technologies will cope with the increasing power consumption of integrated (low power) circuits.

Currently used battery-technologies are summarized in Table 9.

Tech- nology	Cell Voltage [V]	mAh	C Rate‡	Wh/liter	Wh/Kg	Recharge Cycles‡‡	Loss/ Month
NiCd	1.2	1000	10 C	150	60	1000	15%
NiMH	1.2	1200	2 C	175	65	500	20%
Li Ion (CoO_2)	3.6	500	C	225	90	1200	8%
Li/MnO_2	3.0	800	C/2	280	130	200	1%
Pb Acid	2.0	400	C	80	40	200	2%

Table 9: Characteristics of rechargeable AA-size batteries [Powe95].

---

‡ A discharge or charge current equal in amperes to the nominal ampere-hour capacity of the battery. E.g. a rate of 2C means, that a battery can be completely charged respectively discharged in 1/2 hour.

‡‡ The number of recharge cycles is defined as the number of recharge cycles until the storable energy drops to 80% of the brand new battery.

All of these batteries have their domains of application. E.g. for notebooks NiMH and Li Ion batteries are most common and Li Ion batteries are gaining importance. The maximum storable energy density of Li Ion batteries is expected to increase by a factor of approximately 2x in the next few years. They will provide 3-4x higher energy densities as NiMH batteries [Nebe97]. Besides the storable energy (per weight respectively per volume) further characteristics are given in Table 9, which are of varying importance for different applications. The different characteristics are discussed in detail in [Powe95].

For battery trends the roadmap is less precise than for semiconductors. However, it is obvious that in the next 15 years no break-through battery inventions are expected, which will satisfy all user requests in terms of portability and time of operation. Hence it is very important to continue the exploitation of all possibilities to reduce the power consumption of portable applications while meeting other constraints (e.g. performance).

3 Basics

Within this chapter the basics for gate level (power) simulation are dealt with.

In Chapter 3.1 the power consumption of standard cell based designs is introduced. As signal transitions within a CMOS circuit are the principle cause of power consumption, the analysis of dynamic circuit behaviour is a key task, which is discussed in Chapter 3.2. The different sources of power consumption in CMOS circuits are discussed in Chapter 3.3.

3.1 Power consumption of standard cell CMOS designs

A general standard cell based integrated CMOS circuit is built up of a number of instantiated library cells, which are connected with each other by electrical wires. The library cells are provided by the fabrication companies. Library cells are commonly available for basic combinational functions, buffers, tristate drivers, basic sequential elements (i.e. flipflops and latches) and pads. Several functionally equivalent cells with different driving capabilities are typically included in a single library. Within the common top down design flow, a high level circuit description is synthesized towards its final implementation. Within technology mapping, Boolean expressions and general storage elements are mapped on the available library cells. The logical composition and the choice of cells may be constraint driven. Typical constraints are area, delay (respectively circuit performance), testability and power consumption. After technology mapping the placement and routing has to be done in order to obtain the final layout. When gate level simulations respectively estimations are addressed within this thesis, mapped standard cell circuits with possibly available backannotation data are referred to.

An integrated circuit typically has a couple of input and output pins. In addition to these signal pins, supply pins are needed to connect the die with V_{DD} and V_{SS}[†]. The instantaneous electrical power *consumption* of the integrated circuit is given by the product of the supply voltage v(t) and the supply current i(t) (confer Figure 10):

$$\begin{aligned} P(t) &= v(t) \cdot i(t) \\ P(t)\big|_{v(t) = V_{DD}} &= V_{DD} \cdot i(t) \end{aligned} \tag{6}$$

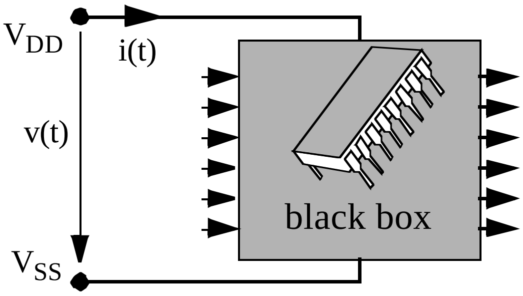


Figure 10:Abstract view of an integrated „black box“ circuit.

The supply voltage v(t) is typically approximately time invariant ($v(t) \approx V_{DD}$). The current i(t) depends on the supply voltage V_{DD}. The energy consumption of a circuit is calculated by integrating the instantaneous power over the referred time interval:

[†] V_{SS} is typically defined as reference for all voltages (0V). I.e., whenever only a voltage is given without an explicit definition of the reference potential, V_{SS} is the reference potential.

$$En = \int_T p(t)dt = V_{DD} \cdot \int_T i(t)dt = V_{DD} \cdot Q \tag{7}$$

The term power consumption typically refers to the time average of the instantaneous power consumption over a *certain* time interval:

$$P = \frac{En}{T} = \frac{V_{DD} \cdot \int_T i(t)dt}{T} = V_{DD} \cdot I = \frac{V_{DD} \cdot Q}{T} \tag{8}$$

Within Equations 6-8 the terms charge, instantaneous power consumption, energy and (average) power consumption have been defined. For the terms charge, energy (consumption) and (average) power consumption a mathematical relation is given in Equation 8 to calculate one term from another, if the supply voltage and the reference time interval are known. This is important, as these three terms are often used interchangeable in the domain of power analysis.

If power *consumption* or even energy *consumption* are referred to, this causes the impression, that power respectively energy is consumed, i.e. vanishes during the process of consumption. It is well known from physics, that energy cannot vanish. However energy can be turned into another form of energy. Within integrated circuits electrical energy, which is supplied to the circuit by the voltage source, is typically partly turned into heat and partly stored within capacitances. The electrically stored energy (in capacitances), however, cannot be returned to the voltage source, if common design style (except adiabatic design style) is used. I.e., strictly speaking, no energy nor power is consumed. However, the electrical energy, which is supplied to the circuit and eventually turned into heat, cannot be used and on contrary further efforts have to be done to transfer the heat to the ambient (confer Figure 11). Hence the energy, which is transferred to the circuit is *lost* from the circuit user's point of view and is referred to as energy, which is *consumed* by the circuit.

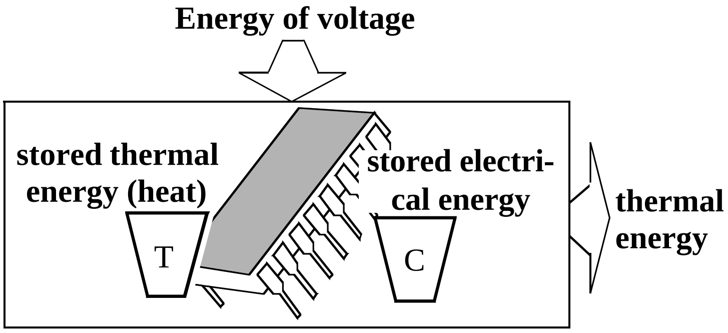


Figure 11:Physical equilibrium state of energy for an integrated circuit.

The power consumption of a complete circuit or part of it is calculated by adding up the power consumption of all included modules. For high level power modelling (RT level and above) these modules consist of (large) functional units. On gate level these modules are simple CMOS cells, which are part of the silicon provider's library. Within this thesis gate level power calculation is focused on.

An arbitrary CMOS cell typically consists of one or more interconnected CMOS stages. Each CMOS stage is build up of one pull up and one pull down network (confer Figure 12). If all

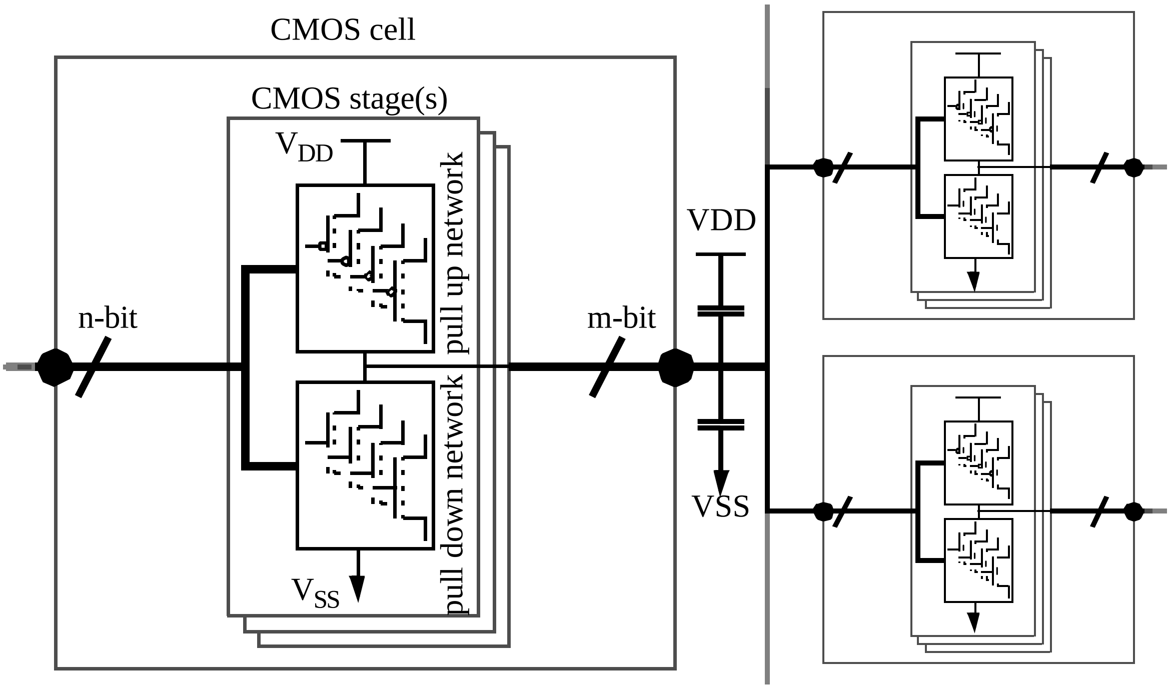


Figure 12:Interconnected CMOS cells and internal cell structure: the cell's functionality is realized by one or more interconnected CMOS stages.

input voltages of a CMOS stage are stable at V_{DD} or V_{SS} either the pulldown or the pullup network respectively both are in a high resistive state. For common Boolean stages either the pullup- or the pulldown network is conducting. Stages, for which both networks (pullup and pulldown) may be high resistive at the same time, are needed in tristate and sequential cells (confer example in Figure 13).

As a consequence, during static operation (all input voltages are at either V_{DD} or V_{SS}), no conducting path through a cell from V_{DD} to V_{SS} exists and hence in CMOS circuits ideally no (static) power consumption is occurring. However, due to non ideal transistor behaviour (leakage currents), input signal degradation and signal conflicts, static power consumption is possible (confer Chapter 3.4). The major portion of a well designed CMOS circuit's total power consumption is dynamic capacitive and short circuit power consumption. This may not be true for technologies with very low threshold voltages, which will be needed for low supply voltages in future technology generations.

3.2 Signal modelling in digital circuits

As the dynamic signal behaviour is the key for the major part of power consumption, basic signal modelling issues are presented here. First signal propagation through arbitrary elements[†] are dealt with, which are caused by complete input signal[††] transitions (Chapter 3.2.1). In the Chapters 3.2.2-3.2.4 these basic observations are enlarged to handle more general situations of simultaneous input transitions at different input pins or even glitches and hazards.

[†] an element is a system with possible memory, which transfers a given input signal to the system's output according to its system response. In this context typically cells are referred to.

[††] in this context signals are associated with the corresponding node voltages.

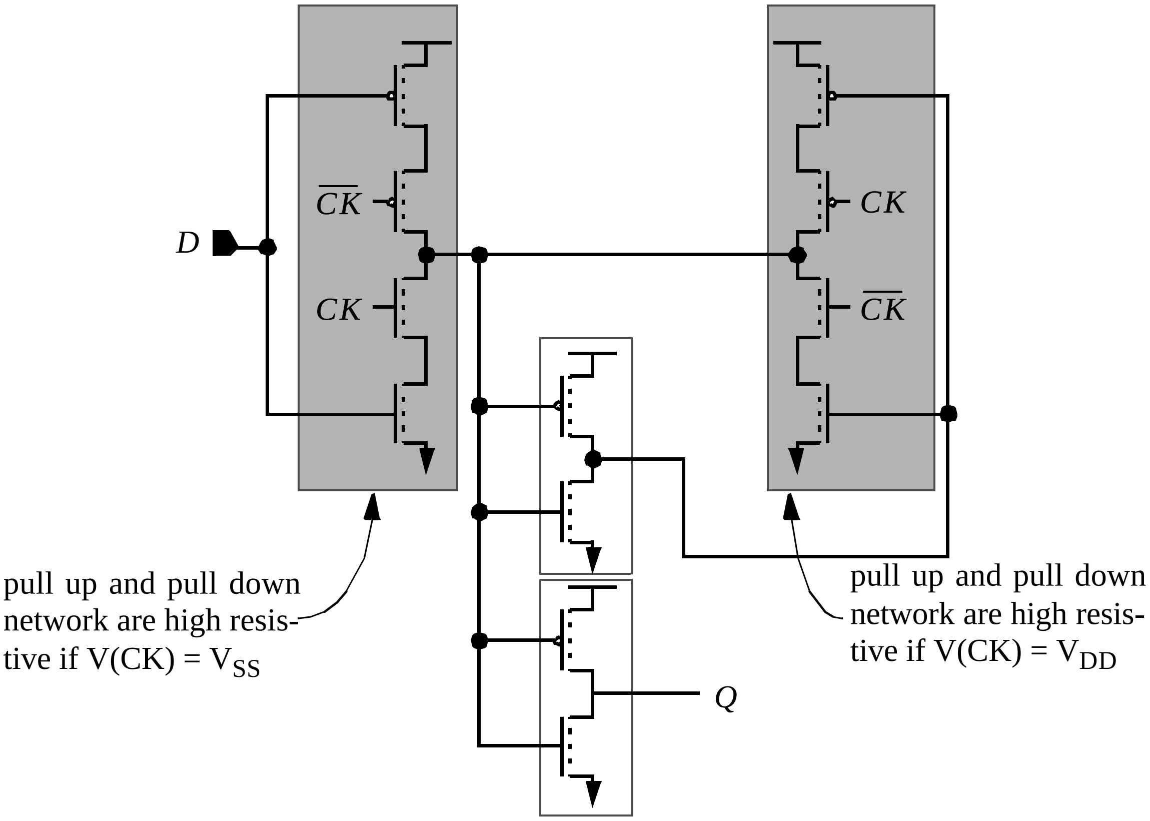

D
$\overline{CK}$
CK
CK
$\overline{CK}$
Q
pull up and pull down network are high resistive if V(CK) = V_{SS}
pull up and pull down network are high resistive if V(CK) = V_{DD}

Figure 13:Transistor netlist of a latch: the pullup and pulldown network of the shaded stages may be high resistive simultaneously.

Within Boolean algebra all signals are represented by the Boolean values {0,1}[†]. These Boolean values (bit values) are associated with electrical voltages (respectively ranges of voltages) in CMOS circuits. The reference potential within CMOS circuits is typically V_{SS} (=0V). The Boolean value ’0’ (’1’) is associated with voltages below V_L (above V_H). Signal voltages in the range $[V_L,V_H]$ represent undefined Boolean values (’X’). The values for V_L and V_H are typically derived from CMOS stages’ static operation point analysis. For practical purposes V_L and V_H are defined by constant fractions of V_{DD} for a whole cell library.

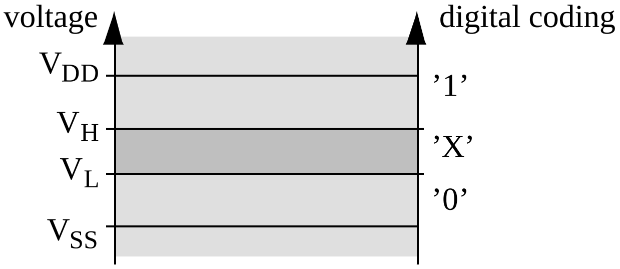


Figure 14:The Boolean values ’0’ and ’1’ are mapped on defined voltage ranges.

[†] the Boolean value ’0’ (’1’) is often referred to as ’L’ (’H’), i.e. Low (High); in [IEEE87] ’0’ (’1’) - forcing low (high) - and ’L’ (’H’) - weak low (high) - are distinguished.

3.2.1 Modelling of single transitions

If a binary signal changes its value at the input of an element (here CMOS cell), a resulting change of the output signal is delayed in causal systems. The signal propagation delay is defined by the time interval between the instants when a predefined input voltage and a predefined output voltage of the corresponding electrical signals are crossed. The predefined voltages must be identical for all cell characterizations to allow efficient event driven simulation. However, the predefined voltages for rising and falling transitions typically have different values:

- V_H: logic threshold voltage for delay characterizations of falling transitions
- V_L: logic threshold voltage for delay characterizations of rising transitions

3.2.1.1 Constraints for logic threshold voltage definitions

The choice of these threshold voltages has a major impact on the actual propagation delay values and their functional relations to influencing parameters (e.g. input slope and output load). Hence the following constraints should be taken into account [Lehm95]:

a) **Positive Delay Constraint**: only positive propagation delays can be efficiently handled within event driven simulators,

b) **Linearity**: the propagation delay's dependency on the input slope should be minimized and possibly be linear,

c) **Summability**: The propagation delay of a number of gates connected in series must equal the sum of the single propagation delays.

A range of possible threshold voltages (V_L,V_H), which ensure positive propagation delays, can be derived from a cell's static operation point analysis. In Figure 15 dynamic and static opera-

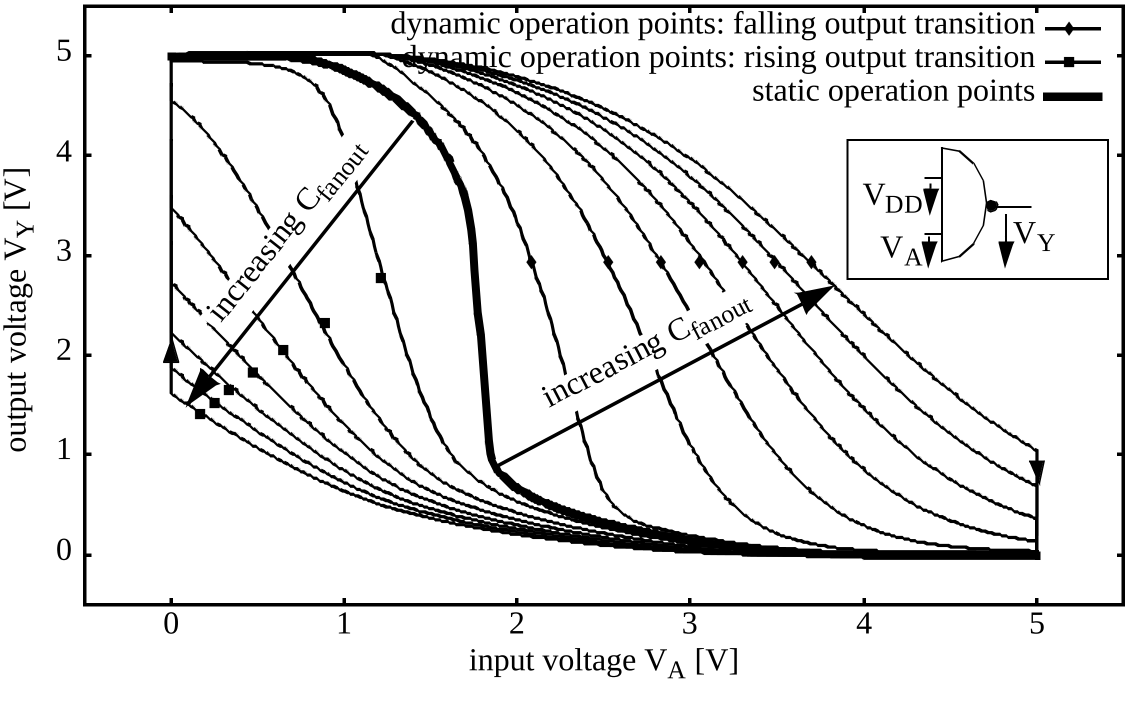


Figure 15:Static and dynamic operation points of a NAND2-gate (constant input slopes).

tion points are given for a NAND2 cell. One input is stable at V_{DD} and the other is connected to a rising respectively falling voltage signal.

For a falling input transition and a rising output transition the possible operation points cannot be below the static operation curve because the rising output signal is delayed due to charging and discharging of capacitances[†]. The exact operation points depend on the input slope and the output load.

Therefore different dynamic curves are plotted in Figure 15. On the Y-axis the output voltage V_Y is plotted as a function of the input voltage V_A(X-axis). The larger the fanout load, the more the output transition is delayed. The input voltage waveform $V_A(t)$ is the same for all output loads. I.e., for a rising (falling) input transition a delayed falling (rising) output transition's graph $V_Y(V_A)$ is located above (below) the static operation curve. The more the output transition is delayed (higher fanout load), the further away the graph is located from the static operation curve. The static operation curve is the limiting curve for all possible dynamic curves, which is reached for an input slope close to infinity and a low output load.

For a rising (falling) input transition and a falling (rising) output transition all voltage pairs (V_A,V_Y) on a graph in Figure 15 are reached exactly at the same time (no delay). If such a voltage pair (V_A,V_Y) is used for (V_H,V_L) $((V_L,V_H)$[††]), the delay would be zero for the specific output load. However, the goal is to guarantee a positive delay for all fanout loads (and input slopes). Choosing (V_H,V_L) $((V_L,V_H)$[††]) above (below) a graph means, that the output voltage V_Y is reached later than the input voltage V_A, which corresponds to a positive delay. Hence two conditions for positive delays exist:

- (V_H,V_L) must be above all dynamic operation graphs for a rising input and a falling output transition.
- (V_L,V_H) must be below all dynamic operation graphs for a falling input and a rising output transition.

The upper (lower) limit for all dynamic operation graphs is the static operation graph. All possible (V_L,V_H) values are plotted for both cases[†††] in the upper two diagrams of Figure 16. The axis of the right diagram (falling transition at input A, rising transition at output Y) are exchanged. If the diagrams for both cases are joined, all possible values for (V_L,V_H) are obtained (confer lower diagram of Figure 16).

For non negating cells (e.g. AND cells) possible V_H and V_L values can be determined independently of each other. In Figure 17 possible values for (V_L,V_H) are determined for an AND2-cell in a similar way as for the NAND2-cell in Figure 16. A rising input transition at input A leads to a rising transition at output Y ($V_B=V_{DD}$). Hence all input and output voltage combinations with a positive delay can be derived from the static operation curve (shaded area of the left diagram in Figure 17). The possible values for V_L are located on the black line. Similarly the right diagram exemplifies the situation for V_H (V_A and V_Y falling). These two diagrams are joined in the lower diagram of Figure 17.

[†] for low fanout loads and steep falling (rising) input slopes the operation points may be slightly below (above) the static operation curve due to input to output coupling

[††] (V_L,V_H) refers to a falling input transition at input A and rising transition at output Y.

[†††] first case: rising input transition, falling output transition,
second case: falling input transition, rising output transition.

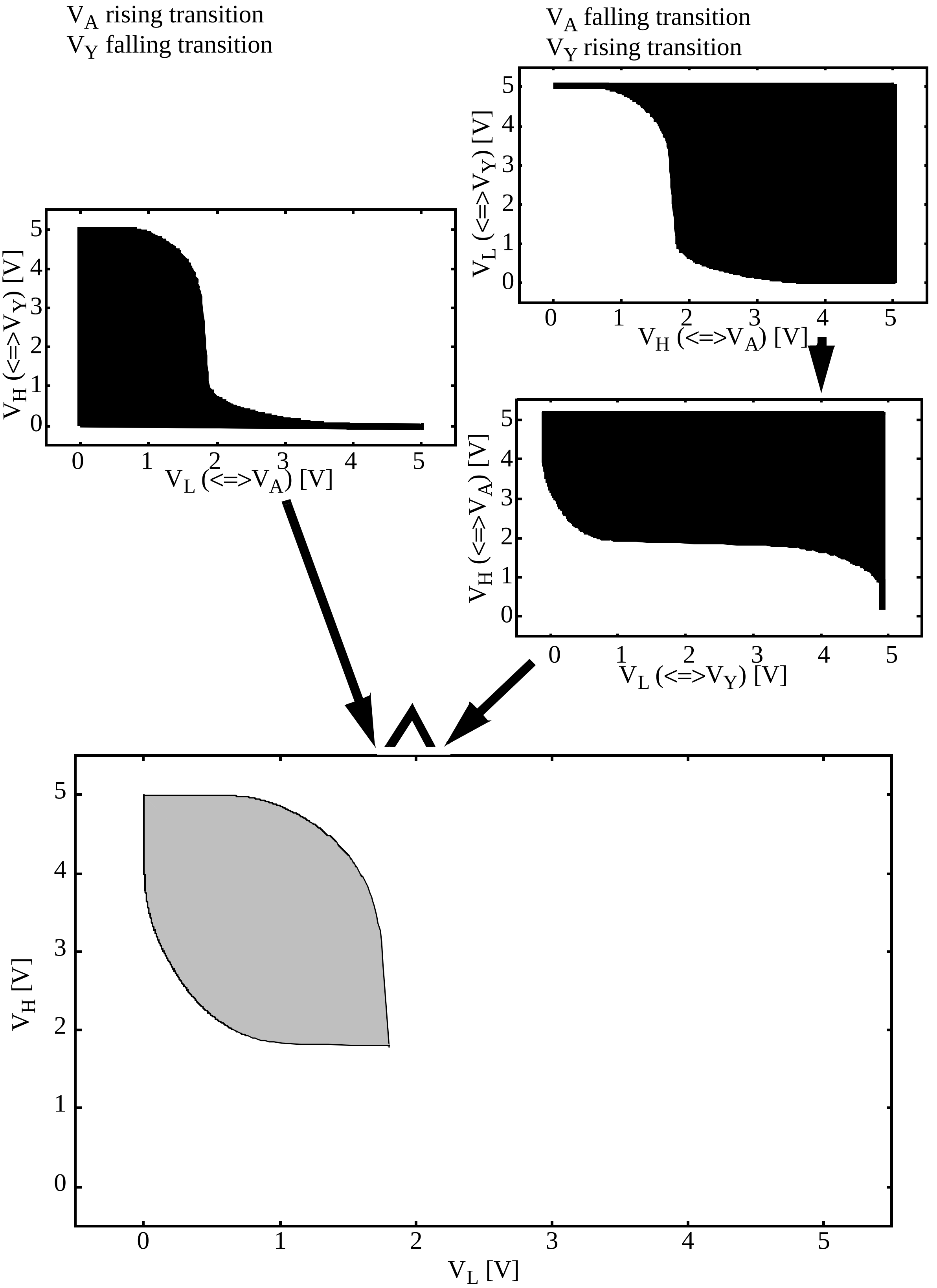


Figure 16:Possible values for V_L and V_H (derived from a cell's static operation curve).

The intersection of all possible threshold voltage sets for all cell's pin-to-pin combinations results in a safe set of possible V_H and V_L values. These tight limits may be relaxed, if the

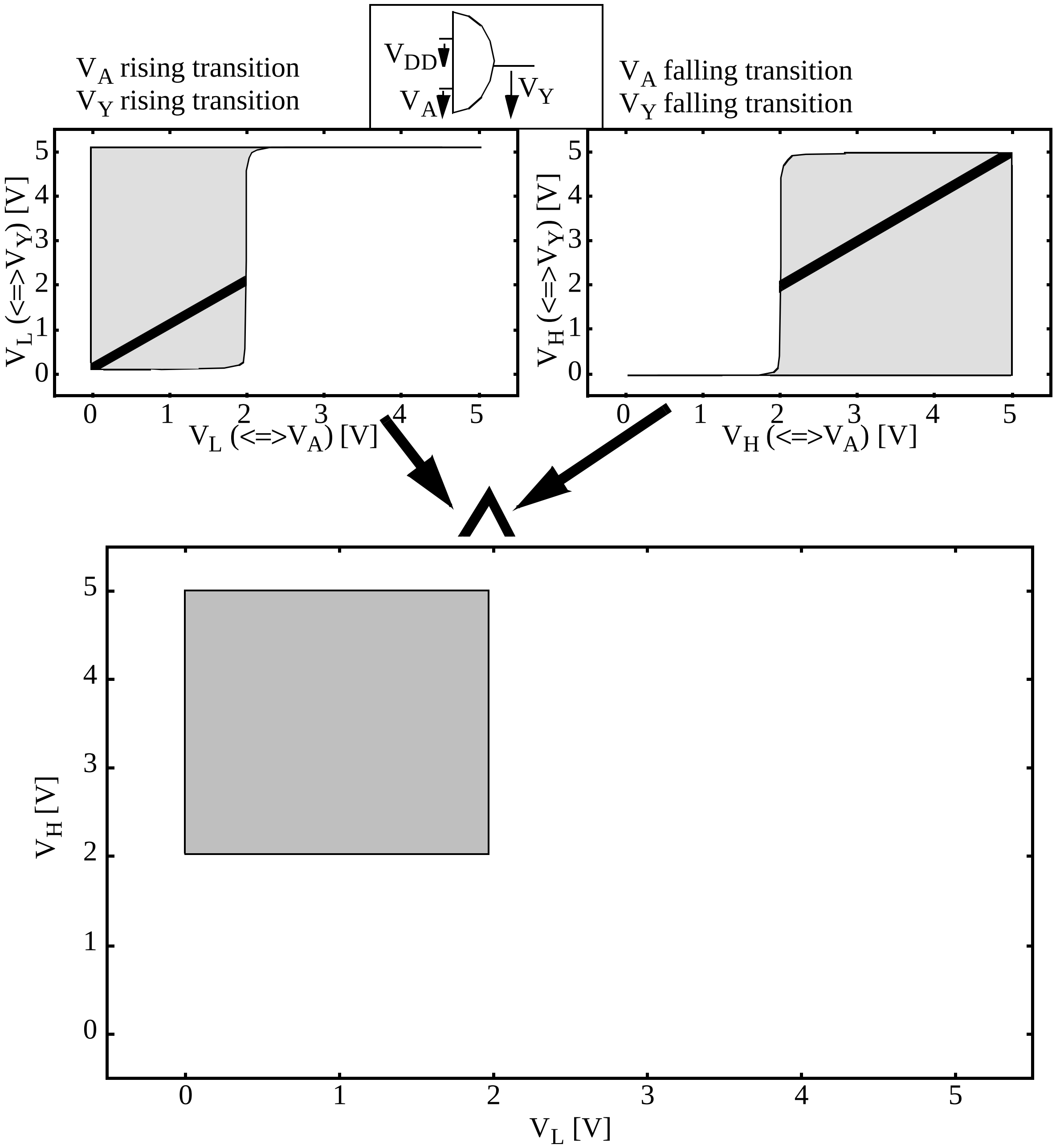


Figure 17:Possible values for V_L and V_H (derived from a AND-cell's static operation curve).

static operation curve is replaced by the worst-case dynamic operation curves for falling and rising output transitions. The worst-case dynamic operation curves are defined by choosing

- the slowest allowed input-slope and zero-load or
- the slowest allowed input-slope and a single fanout gate load.

The difference between the three regions is illustrated in Figure 18 for the NAND2 cell. From this figure it can be observed, that

- the values (V_L,V_H)=(2.5V,2.5V)=(50% V_{DD},50% V_{DD}) may result in negative propagation delays for all three cases
- and that the often used values (V_L,V_H)=(2V,3V)=(40% V_{DD},60% V_{DD}) are just inside the safe region of the no load capacitor's dynamic bound.

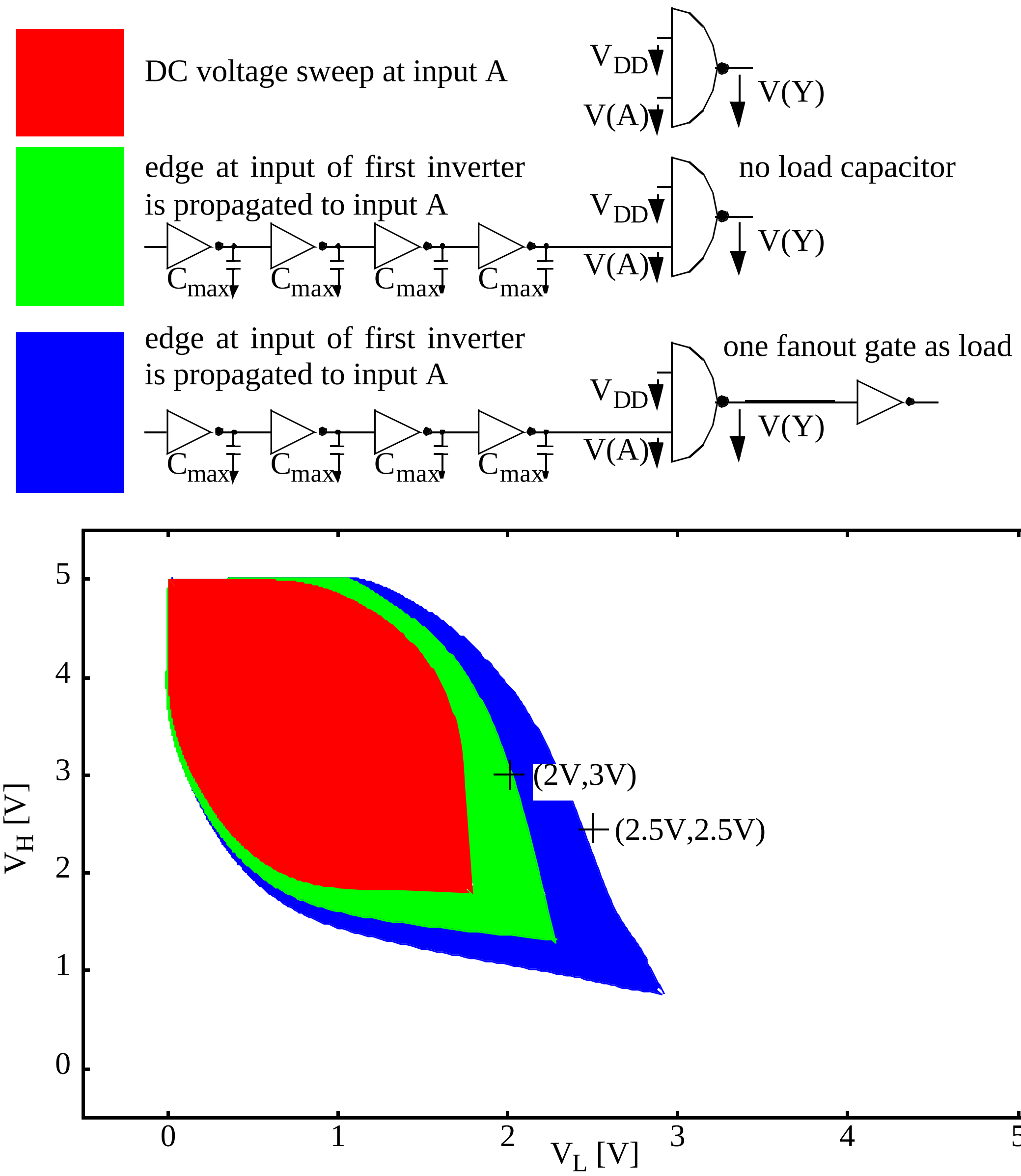


Figure 18:Possible values for V_L and V_H (derived from a cell's static respectively worst case dynamic operation curve).

Besides the *positive delay constraint* the impact of the input slope should be minimized. In [Lehm95] the threshold voltages (30% V_{DD},70% V_{DD}) are proposed to meet this constraint.

The summability constraint is simply fulfilled, if for all library cells the same V_L- and V_H-values are used.

3.2.1.2 Cell delay characterization

Within delay characterization typically all possible input-to-output *paths* are characterized. The path delays for rising and falling output events are commonly distinguished (for tristate gates the set of characterized events may be larger). For nand- and nor-stages, all stable input signals are assigned a non-controlling signal and are therefore unambiguously defined for a characterized path delay. For stages with a more general Boolean functionality, the stable input signals are not unambiguously defined, i.e. multiple possible paths from V_{DD} (V_{SS}) to the output exist for a rising (falling) output transition. In Figure 19 an example is given. For a falling transition at output Y, which is caused by a rising transition at input C, three combinations of static signals at input A and B are possible. If the NMOS transistors' dimensions Tr_{NA} and Tr_{NB} are similar (W/L ratio), mainly two situations should be distinguished:

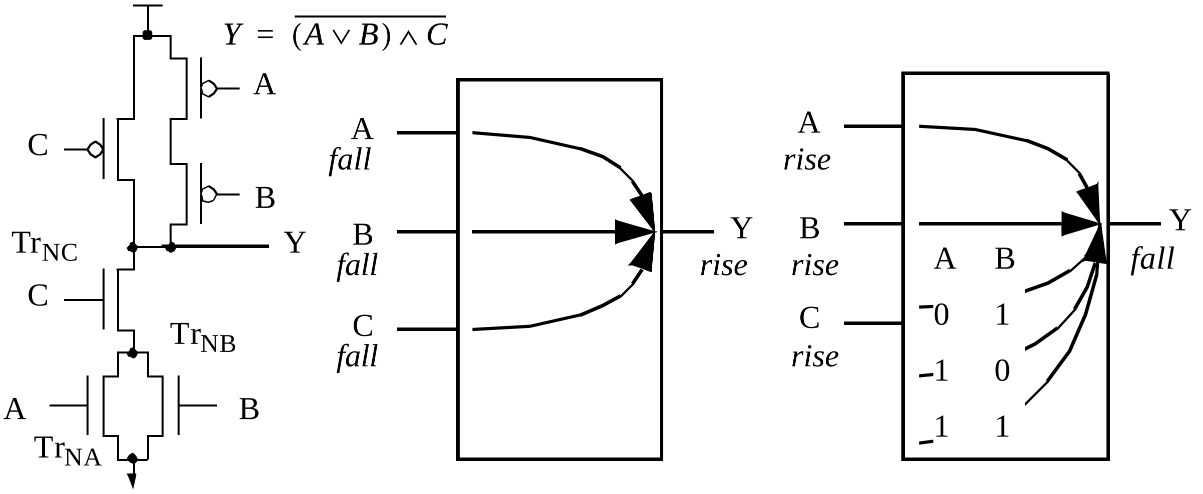


Figure 19:Example for a single stage Boolean function, with multiple possible paths from the input to the output.

- both transistors are on (A = B = 1),
- only one transistor is on (A⊕B = 1).

The on-resistance of the pull down network is approximately (assuming that all transistor's dimensions are similar)

- 2R if either Tr_{NA} or Tr_{NB} is conducting,
- 1.5R if Tr_{NA} and Tr_{NB} are conducting.

Hence the delay is reduced by roughly 25% if both transistors are on compared to the case with only one conducting transistor[†]. The practical delay reductions are somewhat lower, because Tr_{NC} is not immediately turned on. The slower Tr_{NC} is turned on in relation to the complete charging time, the lower is the delay reduction. Practical values for typical input slope and fanout configurations vary between 17.5% and 22% [Vöge98,ES2_07] for the above mentioned cell. The number of transistors in series is typically limited to 4 transistors and hence the maximum number of parallel transistors in CMOS stages is limited to the same number of transistors. I.e., the maximum delay reduction for practically used CMOS stages is 37.5% (100*(1-1.25R/2R)).

For gate level simulation the delay needs to be available for each possible instantiation of a cell and for different input signal waveforms. The fanout capacitance(s) of an instantiated cell has (have) the most important impact on the delay. The input signal waveform is commonly characterized by its slope, i.e. the time interval between crossing the voltage levels of $10\% \cdot V_{DD}$ and $90\% \cdot V_{DD}$. As a gate's output waveform serves as an input waveform for consecutive gates, a gate's output slope also needs to be characterized, if input slopes are taken into account for delay characterization. For delay and slope characterization two approaches are distinguished:

[†] A more detailed discussion about impacts of MOS specific characteristics on the delay differences is omitted here

- analytical equations,
- table look up approaches.

The parameters of analytical equations are typically derived directly from basic semiconductor equations in conjunction with technology informations or detailed circuit-level simulation results [West93].

The table look up approach is based on a number of circuit-level simulations with varying fanout capacitances and input slopes for each delay path. The characterized values can be applied to interpolation procedures in order to obtain delays and output slopes for fanout capacitances and input slopes which are not explicitly characterized. The interpolation is typically needed as it is not possible to characterize each pair of possible fanout capacitances and input slopes.

The advantage of using analytical equations is, that no explicit characterization of each library cell is needed. The table look up approach requires a characterization of each library cell with possible derating factors for process-, temperature- and supply voltage variations. However, if this characterization data is available, the calculation of the instance dependent delay calculations are typically faster and more accurate than the delays, which are calculated from analytical equations. Within this thesis a table look up approach is applied to simulation.

3.2.2 Colliding and non-monotonous signal changes

So far only non colliding input-to-output transitions have been addressed. Within this subchapter this concept is enlarged to more general transitions.

In order to precisely calculate energy-consumption and a circuit’s timing behaviour the logic node transitions must be examined carefully. It is important to note that useless transitions within one clock cycle significantly influence power consumption (typically around 15-20% but in arithmetic units up to 65%[Figu94] or according to my own experiments up to 82%). This influence strongly depends on the architecture. For synchronous circuits useless transitions are hazards or glitches. As an example a voltage waveform at an internal node of a 16x16bit multiplier (ISCAS’87 benchmark circuit c6288), to which one change of pattern at the primary inputs has been applied at 400ns, is shown in Fig. 20. Within this example the number of hazards is actually higher than the number of glitches. However, other nodes can be monitored, which contain more glitches than hazards. Accurate modelling of such hazards and glitches is also important for consecutive gates, where these useless transitions might be amplified respectively filtered. In order to be able to correctly handle such hazards or glitches for consecutive gates, the actual signal waveforms need to be modelled accurately.

A glitch is a special case of a signal propagation collision. A signal propagation collision is defined as follows:

Definition 8: ***Signal propagation collision:***
If two or more changing input signals impact a change of the output voltage waveform at the same time, the input signals collide while propagating through the cell.

In general, colliding input signals may have the following impact on the output signal:

- glitch (confer Figure 21 and Definition 4),
- hazard - similar to a glitch the output waveform is impact by more than one input transition before reaching the peak voltage, which is in contrast to a glitch either V_{DD} or V_{SS} (confer Definition 5),

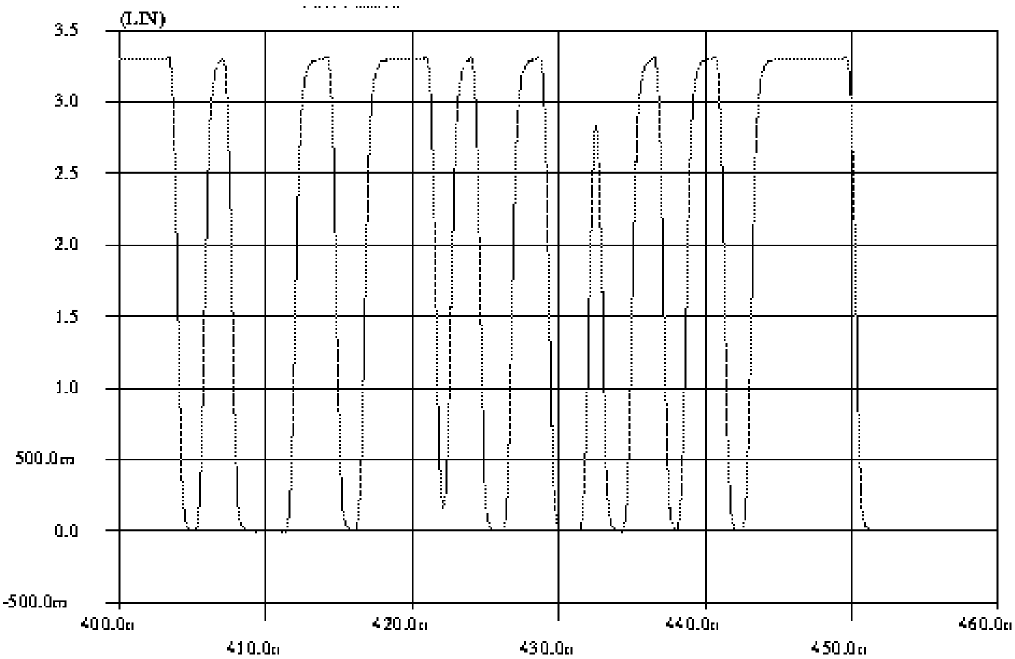


Figure 20:Example of a voltage waveform inside a 16x16bit multiplier (c6288).

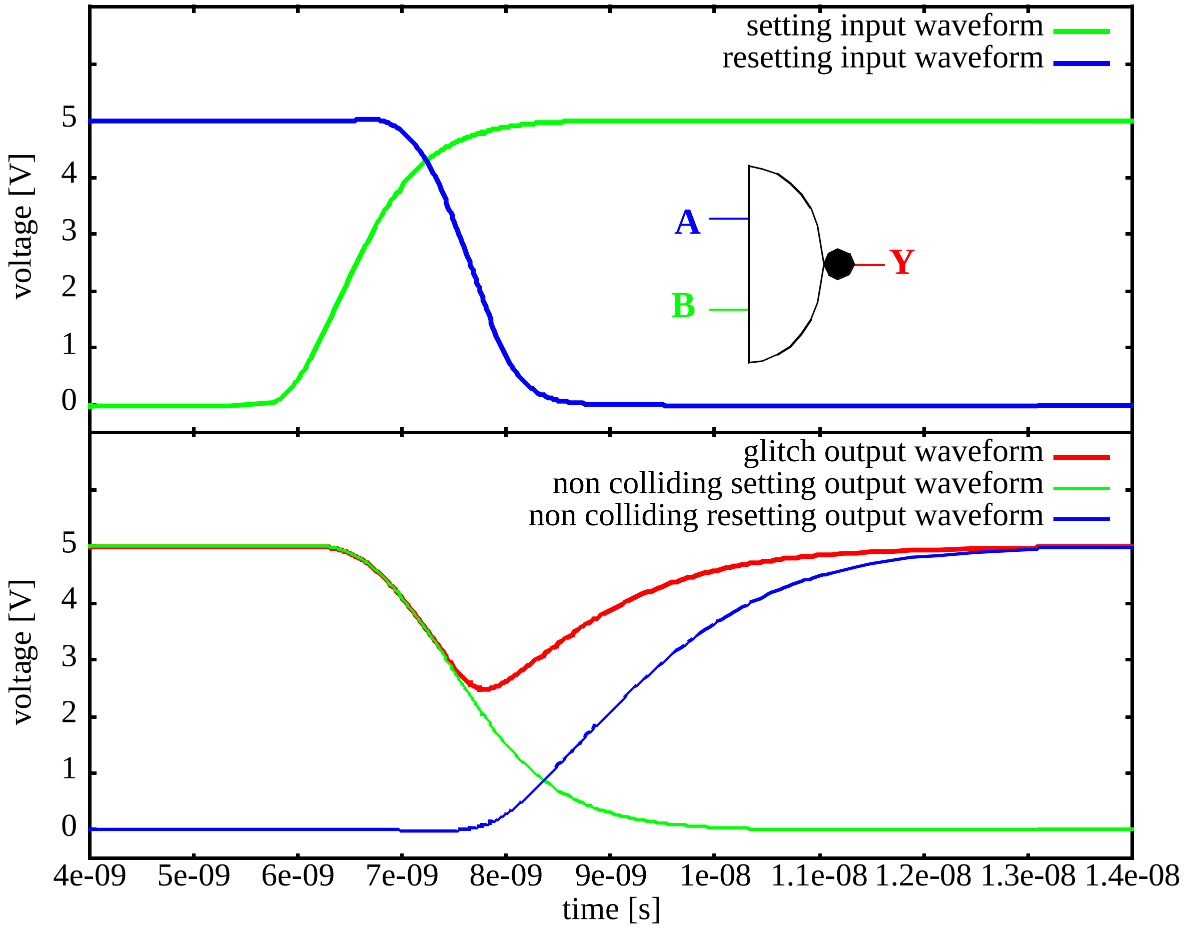


Figure 21:Example for a glitch: Two colliding input transitions result in a glitch (the non-colliding output waveforms are also plotted in the lower graph).

- speed-up transition - the output transition is faster, if it is caused by multiple input transitions instead of a single one - and
- slow-down transition - the output transition is slower, if it is caused by multiple input transitions instead of a single one.

Speed-up transitions, which are caused by two colliding input signals, generally occur, if the output signal transition only requires one of the two input signals to change (with the other remaining constant at the initial signal value). The structural condition is simply, that two parallel transistors are both turned on, which lowers the effective resistance for charging respectively discharging capacitances.

Slow-down transitions, which are caused by two colliding input signals, generally occur, if the change of the output signal requires both signal transitions. Hence, the structural condition requires two transistors in series to be turned on.

The most important of the above mentioned three collision effects from the power estimation's point of view are glitches and hazards. In Figure 21 an example for a glitch at a NAND2 gate's output is given. The glitch is generated by two input transitions in opposite directions. The *setting* input transition (rising) at input B causes the output voltage to drop. The falling *resetting* input transition at input A causes the output voltage to return to its initial value.

Definition 9: ***Setting and resetting transition:***
In case of a glitch generation or propagation, the setting input transition causes the first output transition and the resetting input transition causes the second output transition. The two output transitions have opposite directions.

Besides the glitch-waveform, the figure also contains the complete output-waveforms which would result from one input transition, if the other input signal is stable at 1 (respectively at V_{DD}). The glitch waveform is equivalent to the complete setting output-waveform until the resetting input-waveform becomes important. Afterwards the resetting input transition (input A) starts controlling the glitch waveform. As the voltage of the output waveform is higher than V_{SS} when the resetting input starts controlling the glitch, the fanout capacitances and the cell internal capacitances only need to be partly charged respectively discharged. As a consequence the resetting part of the glitch waveform is delayed less than the corresponding voltage levels of the non-colliding resetting output waveform. This *delay reduction* has a significant impact on signal propagation through consecutive gates, which are sensible to the glitching input signal.

An input collision which results in a hazard simply slows down the first (complete) output transition, because the second input transition has an opposite logical impact on the output. The second output transition is possibly influenced by the setting input transition, because the end of the setting input transition has not necessarily been reached when the resetting input transition starts influencing the output waveform. In conclusion, the hazard's peak voltage waveform is less steep.

Within this work the contribution to glitches which are caused by cross-talk is not dealt with. Even though it should be mentioned that the influence of cross-talk on power consumption will increase with the growing number of metal layers, growing aspect ratios (height/width of metal lines) decreasing metal pitches and the enhancing chip complexity due to shrinking transistor sizes growing die sizes.

Another source for glitch generation are gate-internal charge sharing effects. An example for a glitch generation due to charge sharing is given in Figure 22. When signal A rises, the internal

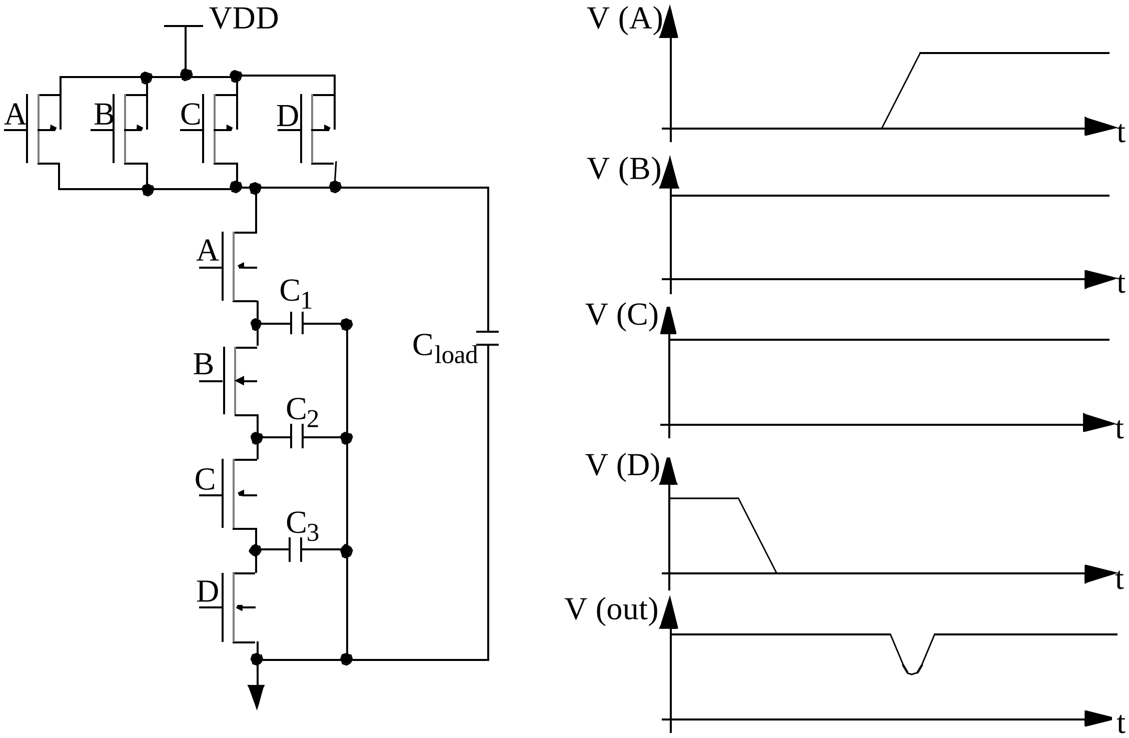


Figure 22:Gate-internal charge sharing effects may also cause glitches.

capacitances C_1, C_2 and C_3 are charged, resulting in a glitch at the output. The glitch peak voltage at the output can be significant especially for small fanout capacitances (C_{load}). These glitches are not considered within this thesis.

In Chapter 3.2.3 the Boolean conditions for glitch generation and propagation are presented. These Boolean conditions determine some logic properties for a glitch to be generated or propagated. Besides these logic properties the temporal relation of the colliding transitions determine the dynamic properties of the glitch, which are discussed in Chapter 3.2.4.

3.2.3 Logic criteria for glitch generation and propagation

In general a glitch is caused by at least 2 transitions at one or different input-pins. The events itself can be full or partial transitions with respect to V_{DD}. At least two transitions have to cause transitions at the gate's output with opposite directions. Two categories of glitches are distinguished:

- glitch generation: n events at n different input-pins cause a glitch and
- glitch propagation: more than one transition at the same input-pin, which may either represent a glitch or a hazard, cause a glitch at the output.

An example for glitch generation and propagation is given in Figure 23.

Non-monotonous gates (confer Definition 10) may have further sources for glitches at a gate's output-node or internal output-nodes of CMOS-stages, which originate from a single input transition due to differing internal path delays. I.e., that a non colliding single input transition

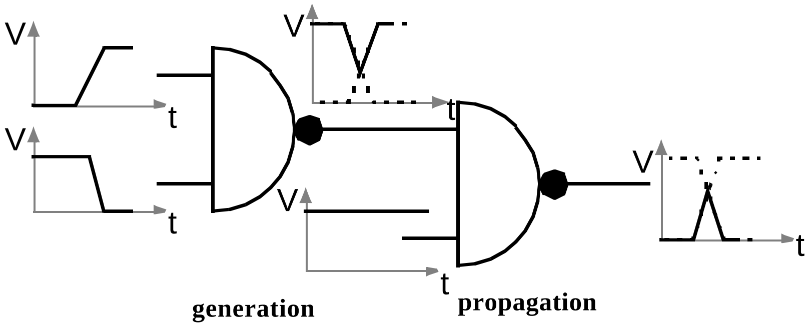


Figure 23:Example for glitch generation and propagation.

may cause glitches. This later category of glitches should be eliminated by library designers. However, if these glitches occur their power-consumption should be calculated correctly.

The following investigations will focus on glitches, which are caused by exactly two transitions. All known models (confer Chapter 4.2 and 5) can be extrapolated on glitches which are caused by more than two transitions by applying the model on pairs of consecutive transitions.

3.2.3.1 Glitch generation caused by two transitions

This subchapter deals with general gates first. Simplifications for special (monotonous) gates are derived afterwards.

Definition 10: ***Monotonous, non monotonous gates:***
*For a **monotonous** gate the direction of a potential output transition is unambiguously defined by the direction of the causing input transition.*

Examples for monotonous gates are AND-, NAND-, OR- and NOR-gates. An EXOR-gate is an example for a non monotonous gate.

Definition 11: ***Inverting and non inverting monotonous gates:***
***Inverting** and **non inverting** monotonous gates are further distinguished. For inverting gates a rising (falling) input transition causes a falling (rising) output transition (e.g. NAND and NOR gates). For non inverting gates a rising (falling) input transition causes a rising (falling) output transition (e.g. AND and OR gates).*

Definition 12: ***Monotonous primitive gates:***
*For **monotonous primitive gates** the input assignment is free from the functional point of view. Such gates' pull down respectively pull up networks either consist of transistors in series or in parallel. All AND-, OR-, NAND- and NOR-gates are monotonous primitive gates.*

a) General Gates

Logical criteria for glitch-generation are introduced here. It is assumed for all cases that two input signals transitions x_i and x_j at two different inputs collide in such a way, that a glitch is possible (from the timing point of view). Two different sorts of glitches are distinguished:

- a transition at input i causes a *falling* edge at the output and a transition at input j causes a *rising* edge or

- a transition at input i causes a *rising* edge at the output and a transition at input j causes a *falling* edge.

The first sort of glitches is called VDD-VMIN-VDD-glitch and the other is called VSS-VMAX-VSS-glitch.

For simplicity logical events from 0 to 1 and vice versa are associated with the two input transitions. In the following equations some terms are used, which are now defined:

symbol	**meaning**
i, j	Inputs of an arbitrary gate, at which two colliding transitions occur.
$x_i(t)$, $x_j(t)$	Signal at input i respectively j as a function of time.
$\underline{x}(t)$	The whole input vector as a function of time.
t_i, t_j	Instant, when a logical event of input i respectively j occurs.
x_i, x_j	Value, at an input (time independent); value may be a Boolean representation of a voltage.
-	Don’t care: the respective variable is removed from the Boolean expression f(x(t)). E.g.: $f(\underline{x}) = x_2 \wedge x_1 \wedge x_0 \qquad f(\underline{x})\|_{x_1 = -} = x_2 \wedge x_0$ $f(\underline{x}) = x_1 \oplus x_0 \qquad f(\underline{x})\|_{x_1 = -} = x_0$
$f(\underline{x}(t))$	Boolean Function of the investigated gate.
t_i^-, t_i^+, t_j^-, t_j^+	The instant immediately before (after) the event t_i respectively t_j is denoted with the superscript - (+).
$\overline{f(\underline{x})}$, $\overline{x_i}$	Negation of Boolean function respectively value.
$x_{iHL}(t_i)$, $x_{iLH}(t_i)$	Signal x_i performs a High->Low respectively Low->High transition at time t_i.

The logical behaviour can be expressed by a Boolean equation as follows:

$$\mathrm{Gl} = \begin{array}{lc} \left(f(\underline{x}(t_i^-)) \wedge \overline{f(\underline{x}(t_i^+))}\right)\Big|_{x_j = -;\; x_i(t_i^-) = \overline{x_i(t_i^+)}} & \begin{array}{c}\text{falling slope caused}\\ \text{by event at x}_i\end{array} \\ \wedge & \\ \left(\overline{f(\underline{x}(t_j^-))} \wedge f(\underline{x}(t_j^+))\right)\Big|_{x_i = -;\; x_j(t_j^-) = \overline{x_j(t_j^+)}} & \begin{array}{c}\text{rising slope caused}\\ \text{by event at x}_j\end{array} \\ \vee & \\ \left(\overline{f(\underline{x}(t_i^-))} \wedge f(\underline{x}(t_i^+))\right)\Big|_{x_j = -;\; x_i(t_i^-) = \overline{x_i(t_i^+)}} & \begin{array}{c}\text{rising slope caused}\\ \text{by event at x}_i\end{array} \\ \wedge & \\ \left(f(\underline{x}(t_j^-)) \wedge \overline{f(\underline{x}(t_j^+))}\right)\Big|_{x_i = -;\; x_j(t_j^-) = \overline{x_j(t_j^+)}} & \begin{array}{c}\text{falling slope caused}\\ \text{by event at x}_j\end{array} \end{array} \tag{9}$$

$$Gl = \begin{array}{c} \left(\underbrace{x_{iHL}(t_i) \wedge f(\underline{x})\big|_{x_i = 1} \wedge \overline{f(\underline{x})}\big|_{x_i = 0}}_{x_i \,=\, \text{falling} \Rightarrow f(\underline{x}) \;=\; \text{falling}} \vee \underbrace{x_{iLH}(t_i) \wedge f(\underline{x})\big|_{x_i = 0} \wedge \overline{f(\underline{x})}\big|_{x_i = 1}}_{x_i \,=\, \text{rising} \Rightarrow f(\underline{x}) \;=\; \text{falling}}\right)\Bigg|_{x_j = -} \\ \wedge \\ \left(\underbrace{x_{jLH}(t_j) \wedge \overline{f(\underline{x})}\big|_{x_j = 0} \wedge f(\underline{x})\big|_{x_j = 1}}_{x_j \,=\, \text{rising} \Rightarrow f\underline{x} \;=\; \text{rising}} \vee \underbrace{x_{jHL}(t_j) \wedge \overline{f(\underline{x})}\big|_{x_j = 1} \wedge f(\underline{x})\big|_{x_j = 0}}_{x_j \,=\, \text{falling} \Rightarrow f\underline{x} \;=\; \text{rising}}\right)\Bigg|_{x_i = -} \\ \vee \\ \left(\underbrace{x_{iLH}(t_i) \wedge \overline{f(\underline{x})}\big|_{x_i = 0} \wedge f(\underline{x})\big|_{x_i = 1}}_{x_i \,=\, \text{rising} \Rightarrow f\underline{x} \;=\; \text{rising}} \vee \underbrace{x_{iHL}(t_i) \wedge \overline{f(\underline{x})}\big|_{x_i = 1} \wedge f(\underline{x})\big|_{x_i = 0}}_{x_i \,=\, \text{falling} \Rightarrow f\underline{x} \;=\; \text{rising}}\right)\Bigg|_{x_j = -} \\ \wedge \\ \left(\underbrace{x_{jHL}(t_j) \wedge f(\underline{x})\big|_{x_j = 1} \wedge \overline{f(\underline{x})}\big|_{x_j = 0}}_{x_j \,=\, \text{falling} \Rightarrow f(\underline{x}) \;=\; \text{falling}} \vee \underbrace{x_{jLH}(t_j) \wedge f(\underline{x})\big|_{x_j = 0} \wedge \overline{f(\underline{x})}\big|_{x_j = 1}}_{x_j \,=\, \text{rising} \Rightarrow f(\underline{x}) \;=\; \text{falling}}\right)\Bigg|_{x_i = -} \end{array} \tag{10}$$

This Boolean expression cannot be transformed into a Boolean difference in general because it has to be ensured that x_i causes a transition at the output opposite to that of x_j.

Definition 13: ***Boolean difference:***
The Boolean difference defines the condition for f($\underline{x}$) to be sensitive on a

$$\textit{change of input } x_i\text{:}\ \frac{\partial f(\underline{x})}{\partial x_i} = f(\underline{x})\big|_{x_i = 0} \oplus f(\underline{x})\big|_{x_i = 1}$$

The Boolean differences $\left.\frac{\partial f(\underline{x})}{\partial x_i}\right|_{x_j = -}$ and $\left.\frac{\partial f(\underline{x})}{\partial x_j}\right|_{x_i = -}$ are necessary but not sufficient conditions for a glitch. Equation 10 can be simplified for monotonous gates.

b) Monotonous gates

For monotonous gates the direction of an input transition unambiguously defines the direction of a possible resulting output transition. Consequently only one term per line of Equation 10 remains.

For inverting monotonous gates (e.g., single stage gates) the terms for falling (rising) output-slopes which are caused by falling (rising) input-slopes are always logically zero. For non inverting monotonous gates the rising (falling) output-slopes which are caused by falling (rising) input-slopes are impossible. Hence Equation 10 can be simplified for inverting monotonous gates as follows:

$$Gl = \begin{array}{c} \left.\left[f(\underline{x})\big|_{x_i = 0} \wedge \overline{f(\underline{x})}\big|_{x_i = 1}\right]\right|_{x_j = -} \wedge \left.\left[\overline{f(\underline{x})}\big|_{x_j = 1} \wedge f(\underline{x})\big|_{x_j = 0}\right]\right|_{x_i = -} \\ \wedge \\ \left[x_{iHL}(t_i) \wedge x_{jLH}(t_j) \vee x_{iLH}(t_i) \wedge x_{jHL}(t_j)\right] \end{array} \tag{11}$$

The terms $[f(\underline{x})|_{x_i = 1} \wedge \overline{f(\underline{x})}|_{x_i = 0}]$ and $[\overline{f(\underline{x})}|_{x_j = 0} \wedge f(\underline{x})|_{x_j = 1}]$ are both false for inverting monotonous gates. Hence Equation 11 can be modified as follows:

$$Gl = \begin{array}{c} \left.\left[f(\underline{x})\big|_{x_i = 0} \oplus f(\underline{x})\big|_{x_i = 1}\right]\right|_{x_j = -} \wedge \left.\left[f(\underline{x})\big|_{x_j = 1} \oplus f(\underline{x})\big|_{x_j = 0}\right]\right|_{x_i = -} \\ \wedge \\ \left[x_{iHL}(t_i) \wedge x_{jLH}(t_j) \vee x_{iLH}(t_i) \wedge x_{jHL}(t_j)\right] \end{array} \tag{12}$$

Equation 12 also holds for non inverting monotonous gates.

c) Monotonous primitive gates

For monotonous primitive gates Equation 12 can be further simplified:

$$\begin{array}{c} \left.\left(f(\underline{x})\big|_{x_i = 1}\right)\right|_{x_j = -} = \left.\left(f(\underline{x})\big|_{x_j = 1}\right)\right|_{x_i = -} \\ \left.\left(f(\underline{x})\big|_{x_i = 0}\right)\right|_{x_j = -} = \left.\left(f(\underline{x})\big|_{x_j = 0}\right)\right|_{x_i = -} \end{array} \tag{13}$$

Combining Equations 12 and 13 the following relation can be derived:

$$Gl = [x_{iHL}(t_i) \wedge x_{jLH}(t_j) \vee x_{iLH}(t_i) \wedge x_{jHL}(t_j)] \wedge [f(\underline{x})|_{x_i = 0} \oplus f(\underline{x})|_{x_i = 1}]\Big|_{x_j = -} \tag{14}$$

All monotonous primitive gates only have either one minterm or one maxterm. Hence the Boolean difference within Equation 14 consists of one minterm only. I.e., that a glitch can only be caused by the following input transitions:

- input x_i falls and input x_j rises or
- input x_i rises and input x_j falls.

One of the glitch causing input-signals of monotonous primitive gates always changes

- from a logically *controlling* to a logically *non-controlling* signal and
- the other input-signal from a logically *non-controlling to* a logically *controlling* signal.

Definition 14: ***Controlling and non-controlling signal:***
An input value x_i is controlling the output, if the Boolean difference of all other inputs is FALSE:

$$\left.\frac{\partial f(\underline{x})}{\partial x_n}\right|_{(x_i = \{0,1\}),\, n \neq i} \equiv 0 \Rightarrow x_i \text{ controls } f(\underline{x})$$

If this property is not fulfilled, x_i is a non-controlling input value.
A controlling input value clearly defines the output value f($\underline{x}$).

Example NAND gate: the value 0 is a controlling input value (output = 1):

$$f(\underline{x}) = \overline{\prod_n x_j}$$

$$\left.\frac{\partial f(\underline{x})}{\partial x_n}\right|_{(x_i = \{0,1\}),\, n \neq i} = f(\underline{x})|_{x_i = 0,\, x_n = 0} \oplus f(\underline{x})|_{x_i = 0,\, x_n = 1} = 1 \oplus 1 = 0 \tag{15}$$

I.e., that *before* and *after* the glitch the gate is driven by a *controlling* input-pattern. Hence for monotonous primitive gates glitches can only be generated at either logical high- or low-level. In particular this means for the following gates:

gate	only possible generated glitches
NAND	VDD-VMIN-VDD glitch
AND	VSS-VMAX-VSS glitch
NOR	VSS-VMAX-VSS glitch
OR	VDD-VMIN-VDD glitch

3.2.3.2 Glitch propagation caused by two transitions

Glitches or hazards at a single input pin may be caused by *either hazards or glitches* of the driving gate. It is assumed, that two consecutive input signal transitions at the same input collide in such a way, that a glitch is possible (from the timing point of view). The logical criterion for a glitch propagation is, that the Boolean difference $\partial f(\underline{x}) / \partial x_i$ (x_i is the causing input-pin) is TRUE.

3.2.4 Dynamic glitch properties

Within this subchapter the basic electrical behaviour of glitches is analysed. These results are used to evaluate other state of the art glitch-models in Chapter 4 and in Chapter 5 a new accurate sophisticated gate-level glitch model is derived. Within this subchapter, it is assumed, that a logical criterion for glitch generation respectively propagation is fulfilled (confer Chapter 3.2.3).

The dynamic and static operation points of a gate for non colliding signals have already been discussed in detail (Chapter 3.1). Two colliding input waveforms, which cause a glitch generation at a single stage gate's output, are discussed now (conferFigure 24):

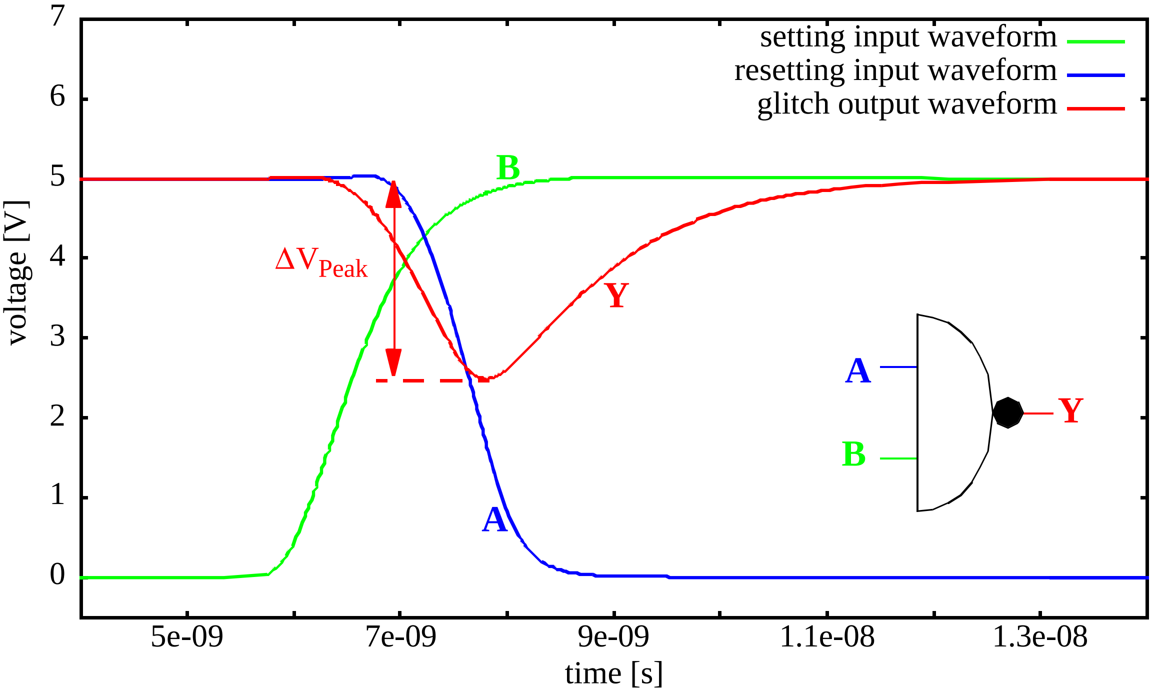


Figure 24:Example for a glitch generation at a NAND2 gate.

- the setting input transition (rising slope at input B) causes the output voltage to leave its initial value,
- the resetting input transition (falling slope at input A) first slows down the change of the output voltage and finally causes it to return to its initial value.

The resulting glitch waveform is also shown in Figure 24. When the glitch peak is reached the voltage waveform's derivation is zero, i.e. the fanout capacity is neither charged nor discharged. Hence at this instant the gate's dynamic operation point is approximately equal to the respective static operation point. I.e., approximately the same output voltage would occur, if the input voltage at both inputs at the glitch peak instant, were applied statically. During the glitch peak instant the gate is in an equilibrium state, which will be further investigated.

The equilibrium state generally depends on both input voltages: the setting and the resetting input voltage at the glitch peak instant. However, for glitches with a reasonable peak voltage, the rising (falling) setting input voltage has usually passed the V_{IHMIN} (V_{ILMAX}) voltage level (confer Figure 25) - due to the gate’s inertia - when it gets into the equilibrium state. This is a very important observation, because the static output voltage $V_{output}(V_{inSet}(t_{glitch}))$ in this region of the static operation curve is approximately V_{SS} (V_{DD}). Hence the setting input waveform effects the equilibrium state only very little except for gates with extremely low fanout loads and slow setting slopes[†]. I.e., the main impact on the equilibrium state has the resetting input voltage $V_{inReset}(t_{glitch})$. This assumption is now investigated experimentally.

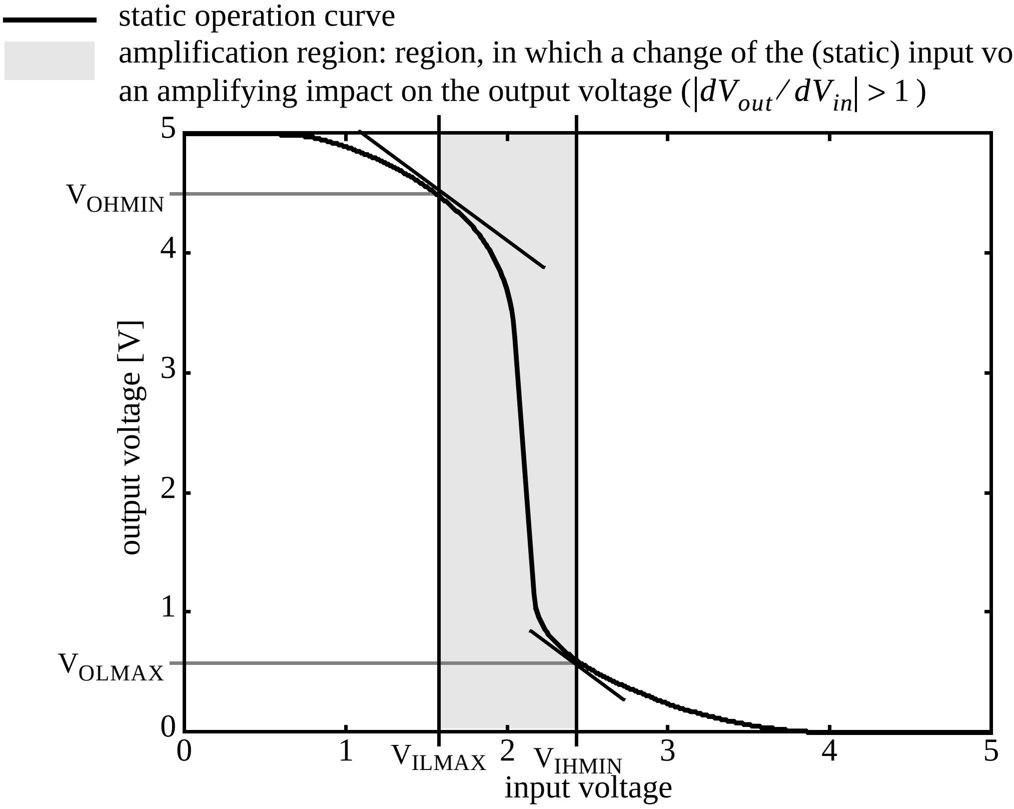


Figure 25:Static operation curve.

Therefore the dynamic and the static operation points ($V_{inSet}(t_{glitch})$, V_{Peak}) were analysed for a couple of gates by means of circuit level simulation (HSPICE) of layout extracted standard cell netlists. The setting and resetting input transitions were applied to different pairs of input pins (glitch generation). The testbench is shown in Figure 26. The fanout loads and the

---

[†] This situation is called *worst case*, because the observations of the glitch behaviour - which will be derived - are not very accurate for this extreme case.

fanin loads were varied to generate glitches with different voltage waveforms. The following terms are used within the testbench explanations:

term	**meaning**
DUT	Device under test (here a NAND4 gate).
C_{fanin}	Fanin capacitance, which is the fanout load of the driving gate: the sum of the intrinsic fanin capacitance (taken from the data sheets) and an explicit capacitor between the DUT's input node and V_{SS} ($C_{faninReset}$ and $C_{faninSet}$ are distinguished - see below).
$C_{faninReset}$	Fanin capacitance of the input at which the resetting input transition is applied.
$C_{faninSet}$	Fanin capacitance of the input at which the setting input transition is applied.
$C_{maxDriver}$	Maximum capacitive load of the driver cell, which is specified in the library datasheets; these numbers are typically derived from maximum delay- respectively output slope constraints during library characterization.
C_{fanout}	Explicit capacitor which is connected between the DUT's output node and V_{SS}.
C_{maxDUT}	Maximum capacitive load of the DUT, which is specified in the library datasheets; these numbers are typically derived from maximum delay- respectively output slope constraints during library characterization.
$V_{inSet}(t)$	Voltage waveform at the setting input pin.
$V_{inReset}(t)$	Voltage waveform at the resetting input pin.
t_{glitch}	Time when the glitch peak is reached.
V_{peak},DV_{peak}	Absolute voltage of the glitching waveform at $t=t_{glitch}$; ΔV_{peak} is the absolute voltage change with relation to its initial value (immediately before the setting output waveform) and V_{peak}.

In this experiment basically the following situations were analysed (confer Figure 26):

a) varying the fanout load with constant input slopes:
$C_{fanout} = \{20\%C_{maxDUT}, 40\%C_{maxDUT}, \ldots, 200\%C_{maxDUT}\}$, $C_{faninReset} = C_{faninSet}$

b) varying the input slopes by different fanin capacitances $C_{faninReset}$ and $C_{faninSet}$ as follows:

	$C_{faninReset}$	$C_{faninSet}$	C_{fanout}
1)	$C_{faninDUT}$	$200\%C_{maxDriver}$	$50\%C_{maxDut}$
2)	$20\%C_{maxDriver}$	$180\%C_{maxDriver}$	$50\%C_{maxDut}$
3)	$40\%C_{maxDriver}$	$160\%C_{maxDriver}$	$50\%C_{maxDut}$
4)	$60\%C_{maxDriver}$	$140\%C_{maxDriver}$	$50\%C_{maxDut}$
5)	$80\%C_{maxDriver}$	$120\%C_{maxDriver}$	$50\%C_{maxDut}$
6)	$100\%C_{maxDriver}$	$100\%C_{maxDriver}$	$50\%C_{maxDut}$
7)	$120\%C_{maxDriver}$	$80\%C_{maxDriver}$	$50\%C_{maxDut}$
8)	$140\%C_{maxDriver}$	$60\%C_{maxDriver}$	$50\%C_{maxDut}$
9)	$160\%C_{maxDriver}$	$40\%C_{maxDriver}$	$50\%C_{maxDut}$
10)	$180\%C_{maxDriver}$	$20\%C_{maxDriver}$	$50\%C_{maxDut}$
11)	$200\%C_{maxDriver}$	$C_{faninDUT}$	$50\%C_{maxDut}$

c) worst case scenario: $C_{faninSet} = 200\%C_{maxDriver}$, $C_{faninReset} = C_{faninDUT}$, $C_{fanout} = 0$

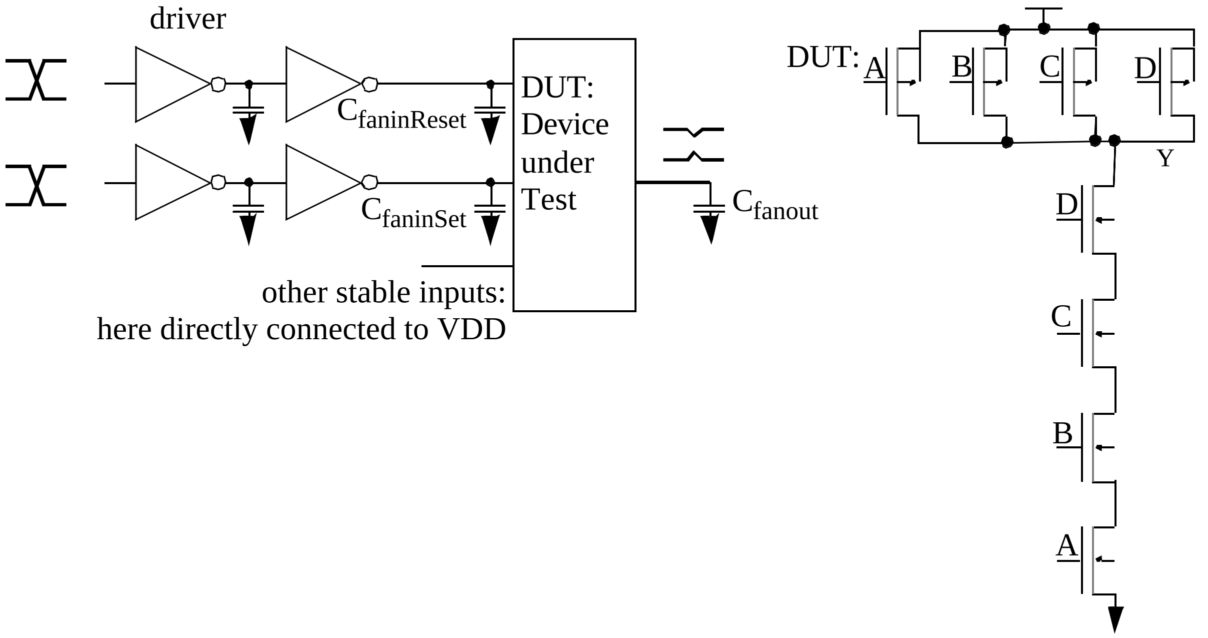


Figure 26:Testbench for glitch analysis of DUT (Device under test).

The input skew of the two input transitions is automatically changed in such a way, that glitches with peak voltages $\{10\%V_{DD}, 20\%V_{DD}, ..., 90\%V_{DD}\}$ are generated.

Some representative simulation results are shown in Figure 27 for a NAND4-gate (setting input transition at D and resetting input transition at A). For the NAND gate all resetting input transitions are falling and all setting input transitions are rising transitions. Each operation

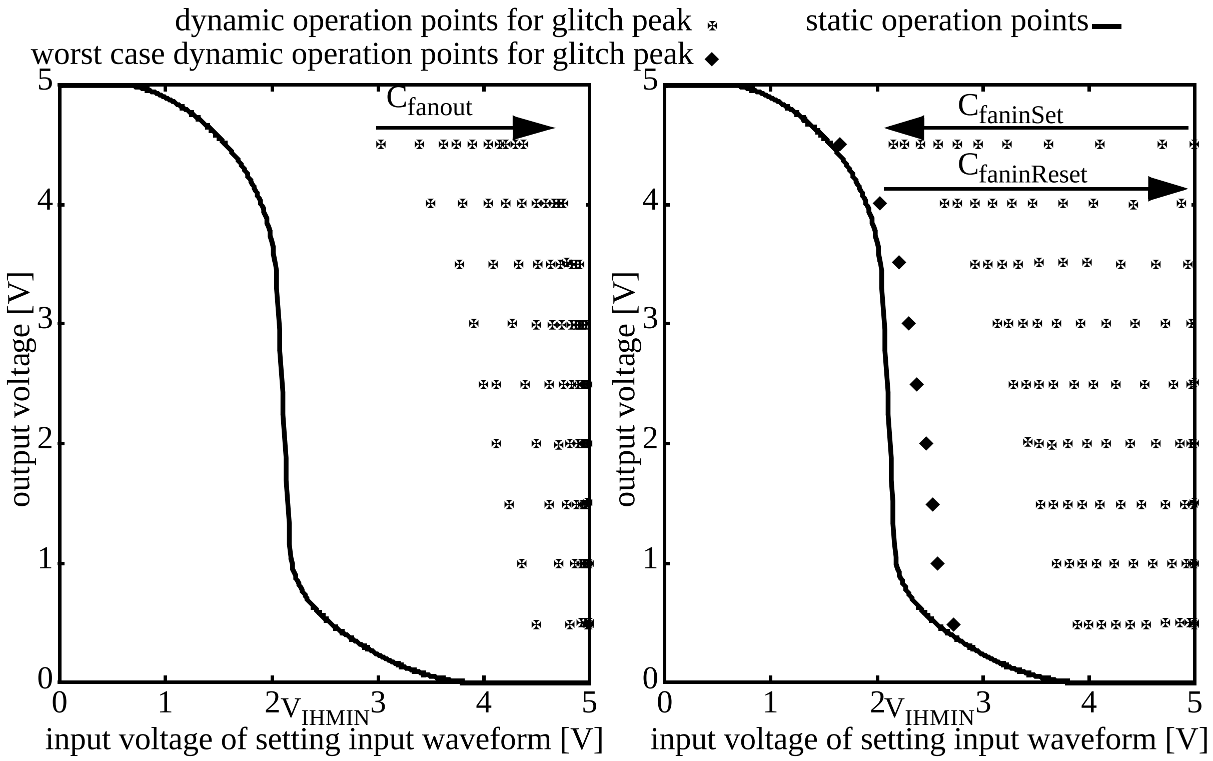


Figure 27:Glitch peak operation points (output voltage versus input voltage of setting input transition) for NAND4-gate.

point ($V_{inSet}(t_{glitch})$, V_{Peak}) is displayed by a small dot. Additionally the static operation curve for $V_A = V_B = V_C = V_{DD}$ is shown. The operation points ($V_{inSet}(t_{glitch})$, V_{Peak}) are to the right of the static operation curve. The distance between the static operation points ($V_{inSet}(t_{glitch})$, $V_{outStatic}(V_{inSet}(t_{glitch}))$) and ($V_{inSet}(t_{glitch})$, V_{Peak}) is a measure for their impact on the glitch operation point (confer Figure 28). The closer the distance the higher is the impact of the setting input transition. Similarly the distance between the static operation points ($V_{inSet}(t_{glitch})$, V_{Peak}) and ($V_{inStatic}$, V_{Peak}) can be used as a measure. For the given example in Figure 28 the equilibrium state's operation point ($V_{inSet}(t_{glitch})$, V_{Peak}) is so far away from the actual gate's static operation points that the setting transition can not be the main important contributor to the equilibrium state of the glitch.

Only the setting input waveforms of the worst case operation points have a significant impact on the equilibrium state. However, this case is very unlikely to occur.

In conclusion, it should be kept in mind, that within some extreme situations the setting input transition may have an impact on the equilibrium state but for common cases the impact is negligible.

Hence, the output voltage during the equilibrium state is mainly a function of the resetting input voltage $V_{inReset}(t_{glitch})$:

$$V_{out}(t_{glitch}) = f(V_{inSet}(t_{glitch}), V_{inReset}(t_{glitch})) \approx f(V_{inReset}(t_{glitch})) \tag{16}$$

For a given glitch peak voltage at the gate's output the input voltage of the resetting input waveform therefore can be read directly from the static voltage figure when the equilibrium

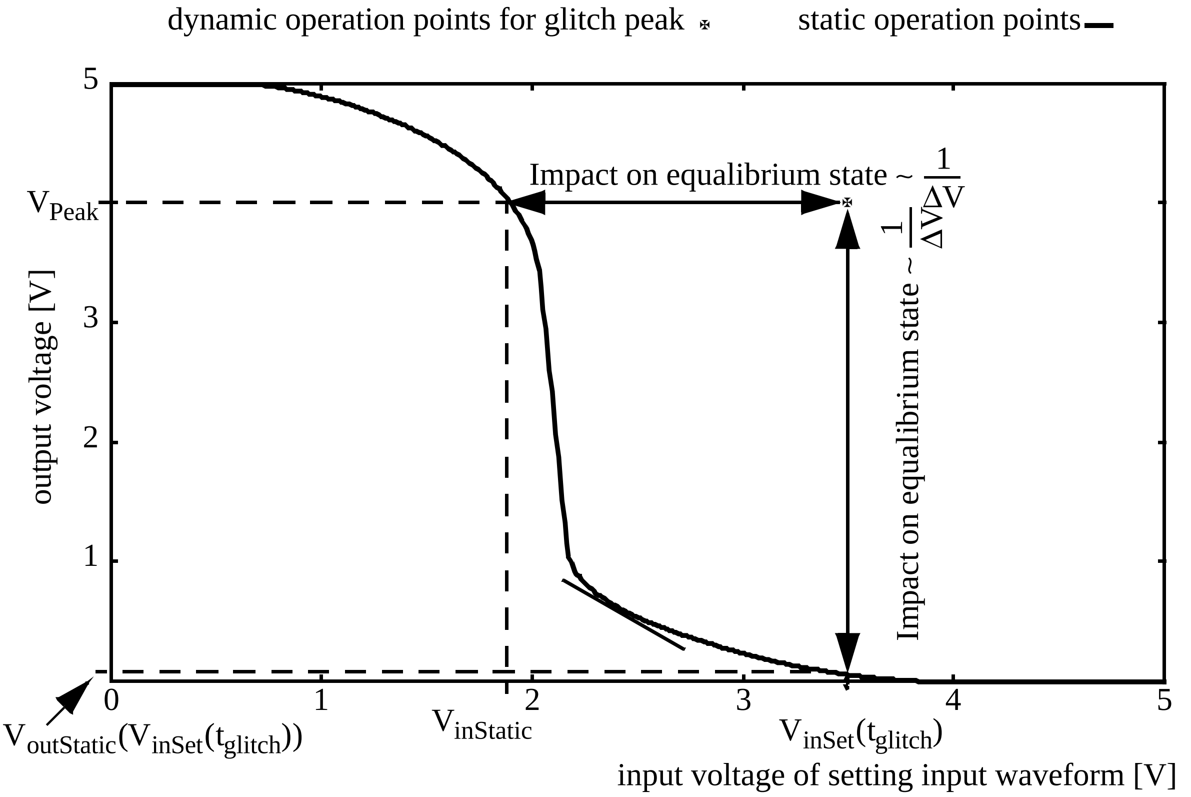


Figure 28:Distance between glitch operation point and the static operation curve is a measure for the impact on the equilibrium state.

state is reached. This observation is illustrated in Figure 29. A glitch is generated at a

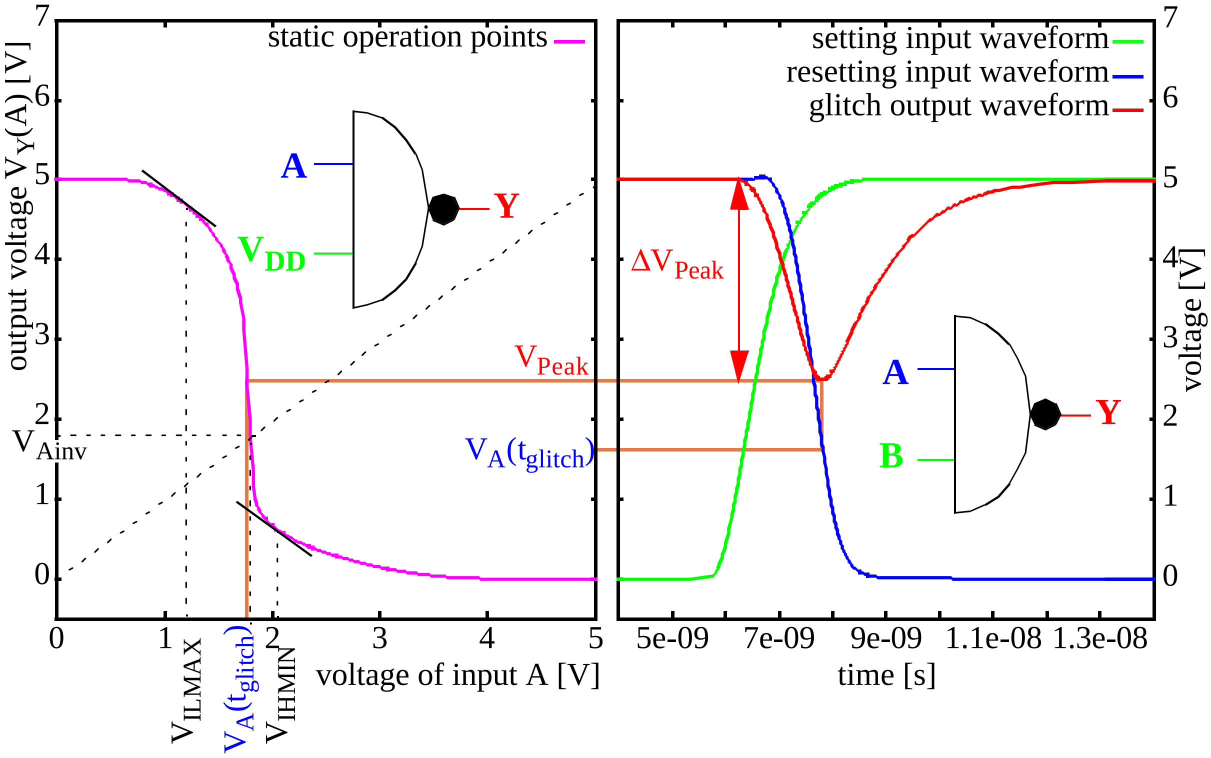


Figure 29:Static operation points of CMOS NAND2-Gate and glitch operation points.

NAND2's output by a rising setting transition at input B and a resetting transition at input A. The voltage of the resetting input waveform $V_A(t_{glitch})$ is highlighted within the dynamic simulation results (right part of the figure). The left part of Figure 29 contains the static operation curve of the NAND2-gate for the path A->Y (i.e., the voltage at input B is kept constant at V_{DD} while the voltage at input A is sweeped from V_{SS} to V_{DD}). The operation point ($V_A(t_{glitch})$, V_{Peak}) belongs to the static operation curve as illustrated in Figure 29.

So far, the equilibrium state was considered as a strict static operation point. As the glitch causing input voltages are not steady at glitch peak time (t_{glitch}), the equilibrium state is slightly degraded due to (dynamic) capacitive input to output coupling. Within the above mentioned circuit level experiments, the dynamic and the static operation points ($V_{inReset}(t_{glitch})$, V_{Peak}) were compared for a couple of gates and a couple of different input-event combinations.

The simulation results are exemplified for a NAND4 gate in Figure 30. The different curves represent the operation points (resetting input voltage versus glitch output voltage) for the above defined dynamic cases a,b (confer page 42).

It can be easily observed from Figure 30, that in the *glitch peak region* all operation curves, which belong to the same glitch peak voltage case, touch each other. I.e., the dynamic impact in the glitch peak region on the operational points is very low. The actual operation points for the glitch peak voltage are pointed out by black dots. The two input waveforms have an opposite dynamic impact on these operation points:

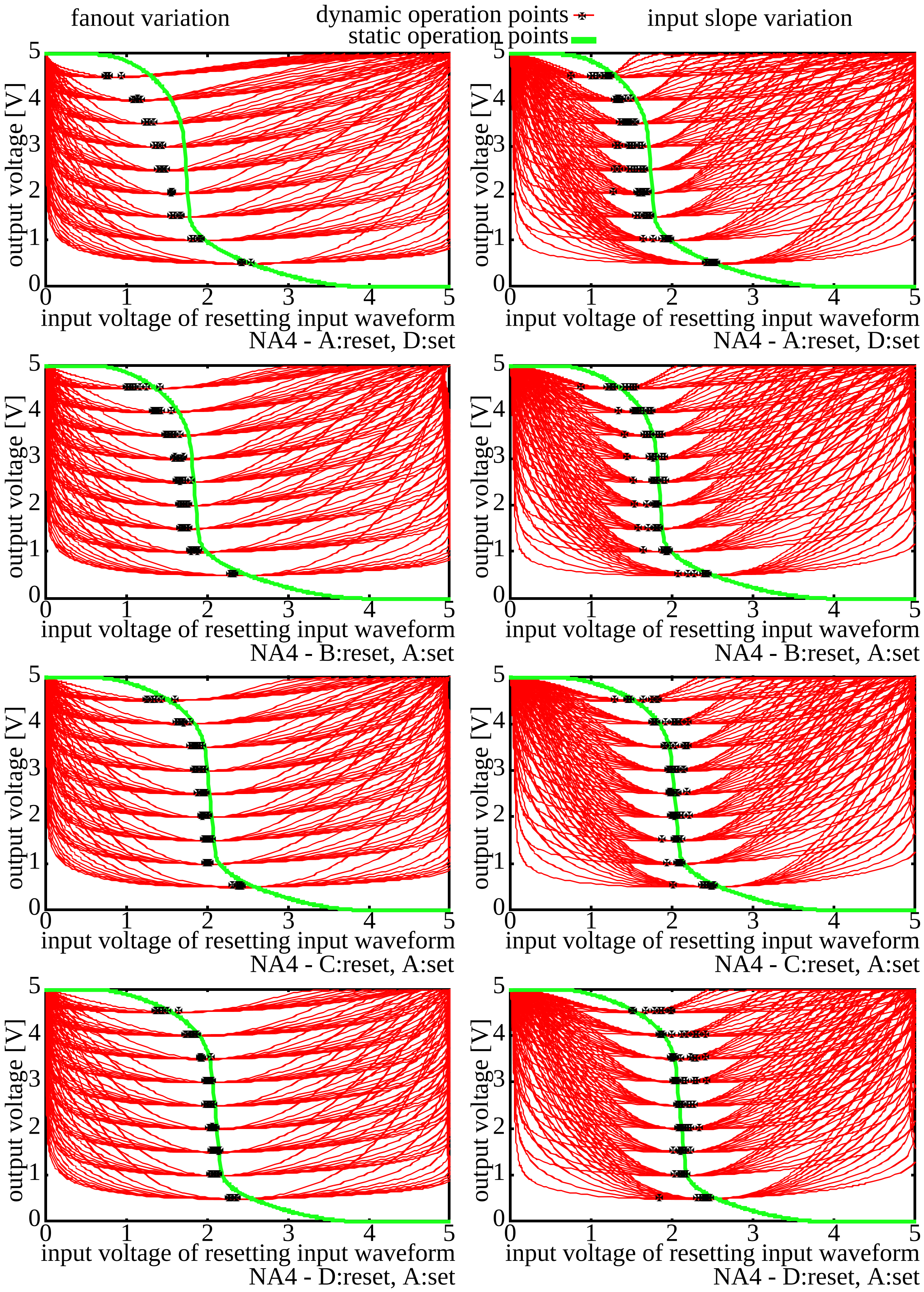


Figure 30:Operation points for glitches (output voltage versus input voltage of resetting input transition) for different input-output combinations.

- the input to output coupling of the rising (setting) input waveform demands lower input voltages at the resetting (falling) input waveform at glitch peak time for compensation; i.e., the points of operation for the glitch peak are to the left of the static operation curve,
- the input to output coupling of the falling (resetting) input waveform accelerates the falling output waveform (before the glitch peak is reached) and as a consequence the points of operation for the glitch peak are to the right of the static operation curve.

The impact of the input to output coupling depends on the position of the driven transistors within the cell’s transistor netlist. The less resistive a driven transistor’s source and drain are connected to the gate’s output, the higher is the input to output coupling. I.e. the resistive connection between the gate’s output and the switching transistor influences how sensitive the output is to input to output coupling. The resistive path between the switching transistor and V_{DD} respectively V_{SS} also has an impact on the sensitivity.

For the investigated NAND4 cell, input D is connected to the NMOS transistor, which is closest to the output in the pull down network (confer Figure 26). In the upper two plots of Figure30 the points of operation are displayed for a setting (rising) transition at input D and a resetting (falling) transition at input A. The setting (rising) input transition is better coupled to the output than the resetting input transition. As a consequence, the actual operation points for the peak voltage are to the left of the static operation curve. For the larger glitch peak voltages ΔV_{Peak} the derivation of the setting input transition is lower and consequently the difference between the static and glitch peak operation points is decreasing.

In the lower two plots of Figure 30 the glitch peak operation points are plotted for a setting (rising) transition at input A and a resetting (falling) transition at input D. Hence the falling (resetting) input transition at input D is much better coupled to the output than the setting transition. As a consequence, the actual operation points for the peak voltage are to the right of the static operation curve for varying input slopes.

Within the other four plots in the middle of Figure 30 input pin D is not involved in the output glitch and hence the operation points for the peak voltage are much closer to the static operation curve.

For the diagrams on the left, the input slopes were kept constant and only the fanout capacitor was varied. Comparing the corresponding plots of the two experiments the operation points at glitch peak time are much more scattered for the experiment with different input slopes (diagrams on the right of Figure 30). The reasons for the (small) variation of the operation points for the varied fanout capacity experiment are:

- The larger the fanout capacitance is, the less is the impact of the input transition (input to output coupling).
- For a large fanout load, the setting input slope is closer to V_{DD} (smaller static impact) at glitch peak time than for a small fanout load.
- For the experiment with fanout variations, the input slopes are equal.

From the circuit level analysis results it can be concluded, that the dynamic impact on the glitch peak operation point generally is quite low. The main contributors to dynamic dependencies are caused by capacitive input to output coupling.

Unfortunately the information, which is important from the glitch modelling point of view, is not the voltage level of the resetting input waveform at glitch peak time but the glitch peak voltage at the gate’s output and the glitch peak time itself. However, due to the characteristic static behaviour of CMOS gates, output voltages in the large range between V_{OHMIN} and V_{OLMAX} belong to a small range of input voltages between V_{IHMIN} and V_{ILMAX} (confer the amplification region in Figure 25). I.e. the time, when the equilibrium state is reached, can be approximated to a first extend by the instant when the resetting input waveform crosses a *typical* voltage value between V_{IHMIN} and V_{ILMAX}. This observation will be used in Chapter 5 to derive the new glitch delay model and to compare previous approaches in Chapter 4.

A generated glitch or hazard is usually applied to consecutive gates through which it might be propagated or not, depending on the state of other input pins and its waveform (confer Figure 23). The general behaviour of glitch propagation is the same as for glitch generation:

- the setting part of the input glitch respectively hazard (before its peak is reached) causes the output voltage to leave its initial value,
- the resetting part of the input glitch (after its peak is reached) first slows down the change of the output voltage and finally causes it to return to its initial value.

The equilibrium state of the output waveform is reached during the resetting part of the input glitch respectively hazard. Generally propagated glitches are decreased if the input glitch’s peak voltage doesn’t cross the voltage V_{jinv}[†] (confer left part of Figure 29). For larger input-glitches it depends on the gate’s dynamic behaviour whether the glitch is amplified or decreased.

During the discussion about the dynamic properties of a gate’s glitch peak operation point, it has been observed, that the small dynamic degradation of the equilibrium state is generally caused by the derivation of the changing input voltage(s) (dV/dt - input to output coupling). For glitch generation generally two different input pins and for glitch propagation only one input pin are involved. I.e., the number of sources for dynamic degradation of the static glitch characteristics are larger for glitch generation. However, for glitch generation the two colliding input transitions partly compensate each other.

The basic glitching behaviour, which has been described in this chapter, holds for single stage gates only. For multi stage gates (e.g. AND-, OR- and EXOR-gates) glitches may be generated at internal nodes and then possibly be propagated to their entity ports. As the external glitch reaches its equilibrium state later than the internal glitch(es), the gate’s internal dynamic behaviour should be considered. The most accurate way to do this, is to divide the gate into its underlying CMOS stages and to analyse them separately. For all other alternatives, which keep treating the gate as a black box, accuracy is sacrificed.

Pass transistor and transmission gate logic are rather seldomly used within ASIC libraries and hence have not yet been discussed [ES2_07,ES2_10]. If a cell contains this kind of logic, commonly buffers are used to strengthen the output signals. Pass transistor and transmission gate logic are not discussed here.

3.3 Power consumption in CMOS circuits

In Chapter 3.2 the dynamic behaviour of voltage waveforms in static CMOS gates was analysed, because the dynamic power consumption within CMOS circuits is typically the dominant component. Within the following subchapters different origins for power consumption in CMOS integrated circuits are discussed:

[†] index j refers to the input pin

- static power consumption:
  - leakage power consumption,
  - non ideal input voltages,
  - signal conflicts (if a signal is driven by more than one driver),
  - wired AND/OR topologies.
- dynamic power consumption:
  - capacitive power consumption,
  - dynamic short circuit power consumption.

Within gate-level power calculation all power components are typically associated with single gates. The power consumption of the whole circuit or part of it is calculated by summing up all gate’s contributions.

Within this thesis static complementary MOS circuits are focused on. Some of the mentioned origins for power consumption are restricted to certain design styles, which are not dealt with in detail. The main important sources for power consumption for static and dynamic CMOS gates in today’s technologies are the dynamic components.

3.4 Static power consumption in CMOS circuits

Static power consumption is occurring independently of the circuit’s dynamic behaviour. The above mentioned four origins are discussed in Subchapters 3.4.1 to 3.4.4. The leakage power consumption is a technological component, which can hardly be influenced by the design style. The other three static components can be avoided by a good design style.

3.4.1 Leakage power consumption

For leakage currents three different components may be distinguished (confer Figure 31):

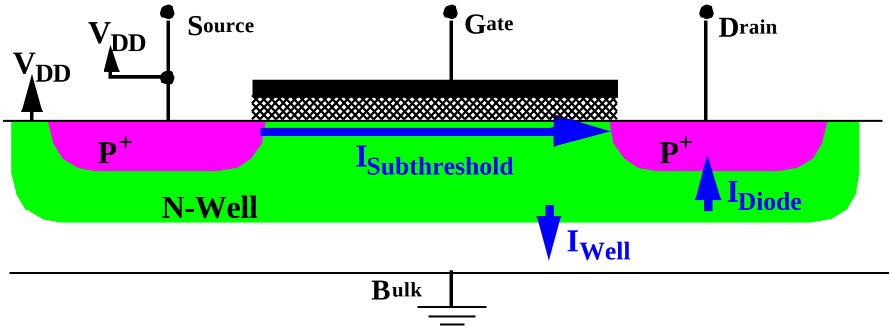


Figure 31:Leakage current in MOSFETs.

- $I_{Subthreshold}$: even without a channel between source and drain, a small current occurs similar to bipolar transistors, if $V_{DS} \neq 0$,
- I_{Diode}: current through reverse biased drain-well respectively drain-substrate diode,
- I_{Well}: the substrate-well diode is always reverse biased and hence only small current densities are possible; however, the diffusion area is very large.

Even if a MOS transistor gets into the subthreshold region ($V_{GS} < V_{TN}$ for NMOS transistors respectively $V_{GS} > V_{TP}$ for PMOS transistors), the source drain current is not abruptly turned off. In the subthreshold region the MOS transistor behaves like a bipolar transistor. The transistor’s source corresponds to the emitter, the drain is equivalent to the collector and the channel region corresponds to the base. The subthreshold current can be expressed for an NMOS transistor as follows [Chan95]:

$$\begin{aligned} I_{Subthreshold} &= K \cdot e^{\frac{V_{GS}-V_{TH}}{n \cdot V_{\text{Temperature}}}} \cdot \left(1 - e^{-\frac{V_{DS}}{V_{\text{Temperature}}}}\right) \\ V_{\text{Temperature}} &= \frac{k\vartheta}{q} \\ V_{\text{Temperature}}\Big|_{\vartheta = 300K} &\approx 26mV \\ I_{Subthreshold}\Big|_{V_{DS} > 100mV} &\approx K \cdot e^{\frac{V_{GS}-V_{TH}}{n \cdot V_{\text{Temperature}}}} \sim e^{V_{GS}-V_{TH}} \end{aligned} \tag{17}$$

Within Equation 17 K and n are technology dependent constants, k is the Boltzmann-constant and q is the elementary charge of one electron. The subthreshold current hence exponentially depends on the threshold voltage. Equation 17 can be easily adapted to PMOS transistors by simply negating the exponents. As the supply voltage is reduced for future technologies, the threshold voltage is also reduced to compensate the disadvantages on the circuit performance. Hence it is obvious, that the subthreshold current is gaining importance within future technologies.

The reverse operating drain-well respectively drain-substrate diodes and well-substrate diodes also contribute to the static leakage currents. Additionally the source-well respectively source-substrate diodes may also contribute depending on the source voltage. The well known equation for diode currents exponentially depends on the diode voltage:

$$\begin{aligned} I_{Diode} &= I_S \cdot \left(e^{\frac{V_{Diode}}{V_{\text{Temperature}}}} - 1\right) \\ I_{Diode}\Big|_{V_{Diode} < -100mV} &\approx -I_S \end{aligned} \tag{18}$$

For back biased operation, V_{Diode} is negative. For the well-substrate diode the voltage is -VDD and hence I_{Well} can be approximated by $-I_S$. The saturation current I_S is a function of the diffusion area and the saturation current density:

$$I_S = A_{Diffusion} \cdot J_S \tag{19}$$

The saturation current density is technology dependent. At room temperature (300 K) it is in the range of $1pA/\mu m^2$ to $5pA/\mu m^2$. The current density strongly depends on the temperature. It doubles with an increase of approximately 9 K.

For technologies above 1µm the diode components $I_{\text{Diode}}+I_{\text{Well}}$ dominated the leakage currents. For submicron technologies the subthreshold currents become increasingly important [Chan95].

Within Equations 17 and 18 it is obvious, that the leakage current also depends on the voltages V_{DS} and V_{Diode}. For a zero-voltage, no current is occurring. I.e., the leakage current depends on the state of each instance (but not directly the dynamic behaviour).

For today's technologies the power consumption due to leakage current is only a small fraction of the total power consumption. For this reason, this component is not further considered within this thesis. However, there is no conceptual blocking point to extend the proposed model to also consider these static contributors.

3.4.2 Non ideal input voltages

If a CMOS gate is statically driven by degraded input voltages the blocking transistor's resistance is reduced and as a consequence, the static current increases. A logically high signal, which is fed to a gate's input via a NMOS pass transistor is degraded by V_{TN}. As a consequence the driven PMOS transistor is turned off less as for a non degraded input voltage (confer Figure 32). According to Equation 17 the drain source current can be approximated with the technological constant K ($V_{GS} \approx V_T$). The input of the gate may even be driven by a worse input voltage, if the pass transistor is turned off. The voltage is further dragged down by the falling gate(-source) voltage due to capacitive coupling.

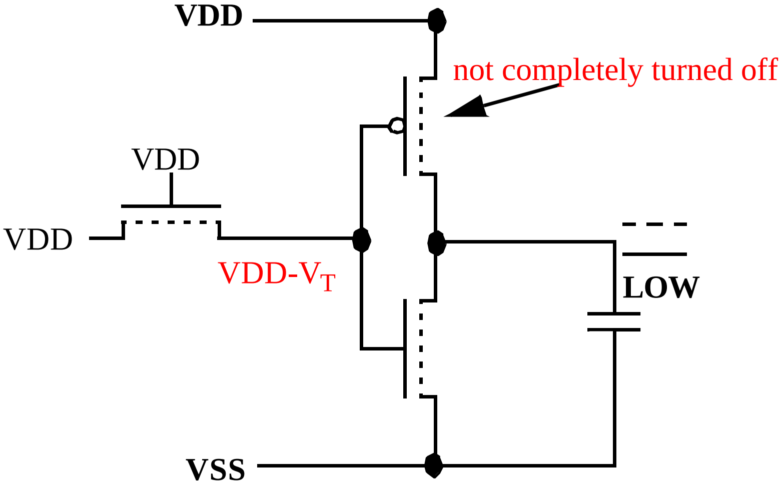


Figure 32:Signal degradation by using an NMOS pass transistors.

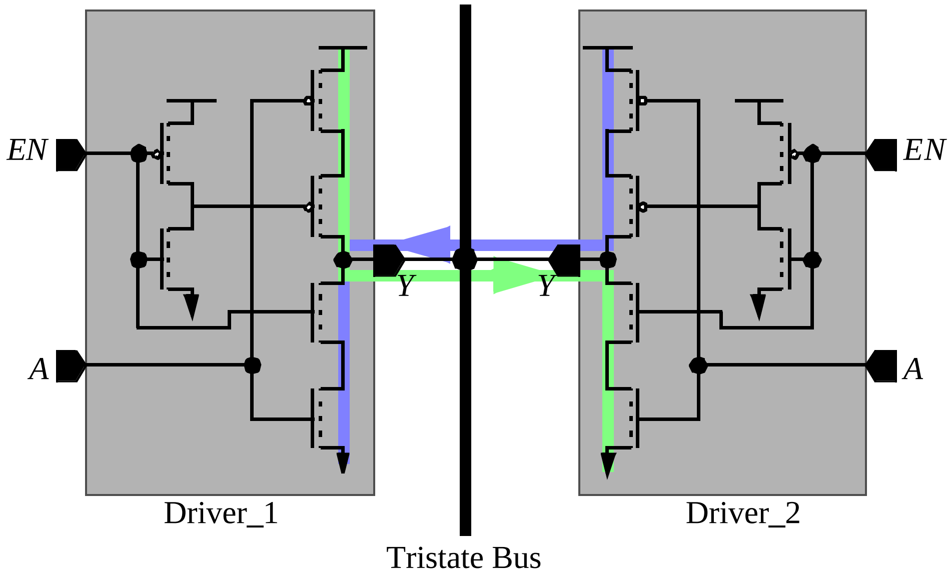
As NMOS pass transistors only degrade a logically high signal, the resulting power consumption depends on the logical input signal and hence is pattern dependent. However, in contrast to the dynamic components (refer to Chapter 3.5) this component occurs statically if the NMOS pass transistor drives a logically one.

This kind of signal degradation can be partly avoided by replacing all pass transistors by transmission gates. Pass transistors should be completely avoided within circuits for low power. Transmission gates, which are not transparent, may also supply consecutive gates with bad signal voltages (capacitive input to output coupling). Additionally, stored charges may drift away over time due to leakage currents, if a signal is high resistive.

The occurrence of non ideal input voltages can be avoided or at least be minimized by a good design style especially for low power applications. For this reason, this power consumption contributor is not further dealt with in this thesis.

3.4.3 Signal conflicts

Within bus systems several tristate drivers may operate as drivers for one net. In well designed circuits, it should be ensured, that the signal is driven by only one driver at a time. If the signal is driven by more than one driver, a resolution function may be used to evaluate the signal strengths and deliver a logic signal value. In this case however, a static low resistive path from VDD to VSS may exist, which results in very high currents. An example is given in Figure 33.



Possible conflict situations:

Driver_1/A	Driver_1/EN	Driver_1/Y	Driver_2/A	Driver_2/EN	Driver_2/Y
0	1	1	1	1	0
1	1	0	0	1	1

Figure 33:Example for a signal conflict.

Two inverting tristate drivers are connected to a tristate bus. The instantiated tristate drivers are transparent, if the enable signal *EN* is high. If both drivers are transparent and the signals at their inputs A differ, a conflict on the tristate bus occurs. The statically conducting paths are highlighted by green and blue arrows in Figure 33 for the two conflict cases. The current and the voltage on the tristate bus depend on the transistor dimensions.

The following two sources for such signal conflicts exist:

- logical design errors (the logical equation for the busdrivers' enable signals may become true for more than one driver at the same time) and
- clock skew problems (the delay for enabling a driver is shorter than disabling the previously enabled driver - this aspect is rather a dynamic effect).

Such signal conflicts should be avoided within the design process. Even though such conflicts are easy to detect by the simulations, they are not further dealt with in this thesis.

3.4.4 Wired AND/OR topologies

Similar to the former NMOS technology in wired and/or topologies the output is always connected to VDD via a resistor. If the output is drawn towards VSS by a conducting NMOS transistor, a permanent conducting path between VDD and VSS exists, which causes a static current flow. In Figure 34 an example is given. If input A is logically one, the Product-Term P1 is dragged towards VSS and a permanent current is flowing through the pull up resistor and the NMOS transistor. Similarly a permanent current is flowing in the OR Plane, if one of the connected product terms evaluates to logically 1.

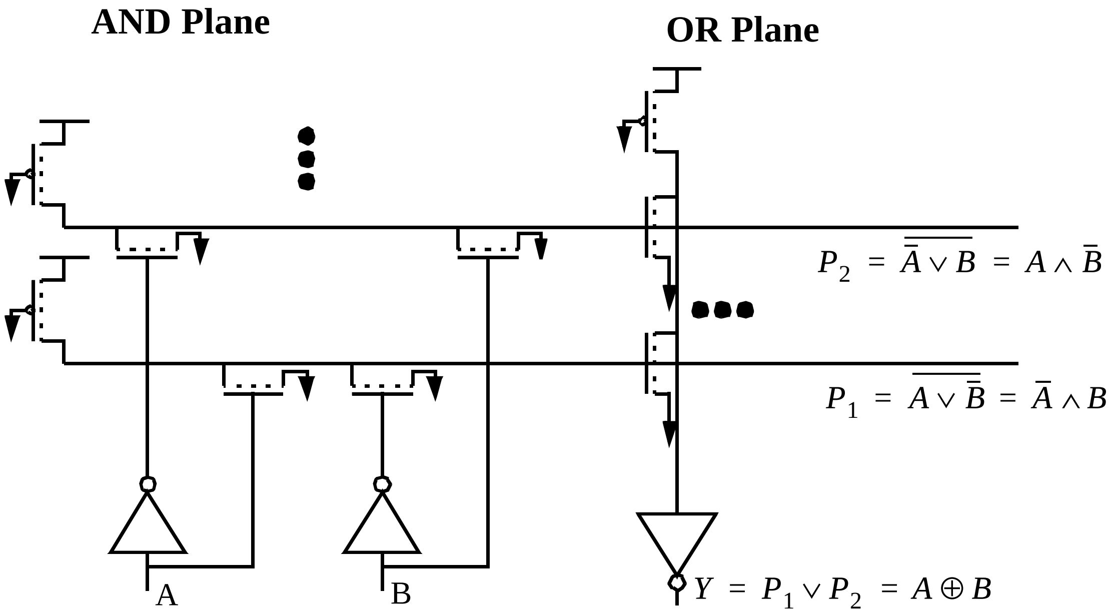

AND Plane
OR Plane
$P_2 = \overline{\bar{A} \vee B} = A \wedge \bar{B}$
$P_1 = \overline{A \vee \bar{B}} = \bar{A} \wedge B$
A
B
$Y = P_1 \vee P_2 = A \oplus B$

Figure 34:Example for a wired AND respectively a wired OR structure.

Such kind of Wired AND/OR topologies are typically used in PLAs but not in ASICs. Hence this source of power consumption is not considered in this thesis.

3.5 Dynamic power consumption

In Chapter 3.4 static power components have been discussed. In this chapter *dynamic* power components are focused on. The term *dynamic* refers to *changing* node voltages of an integrated circuit. I.e., these components only occur *during* switching. The following dynamic power components are distinguished:

- Short circuit power consumption: *During* switching (of a CMOS stage) a conducting path through the pull up and pull down network of a gate is present and as a consequence a short circuit current is occurring.
- Capacitive power consumption: The charging and discharging of capacitances results in power consumption due to the current flow through resistances (transistors and further parasitic resistances).
- Further dynamic contributors are signal conflicts due to different delays for enabling and disabling busdrivers (confer Chapter 3.4.3). However, these contributors are not focused on here.

All dynamic components, which can be observed at a gate’s interface nodes, are shown in Figure 35. The gate is embedded between (a) driving gate(s) and (a) driven gate(s). Driven gates are modelled as capacitances towards V_{DD} and V_{SS}. For a rising transition at the gate’s output and a falling transition at the causing input (confer upper part of Figure 35), the following charges flow through the gate:

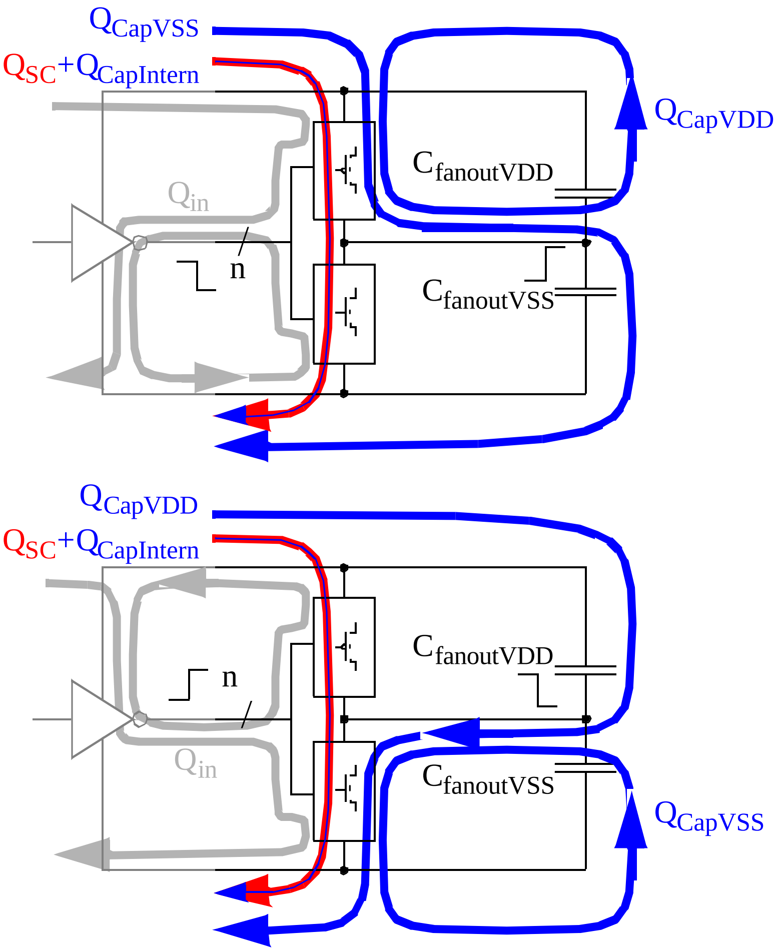


Figure 35:Dynamic charge flow within a single stage gate.

- Q_{CapVSS} for charging the capacitance $C_{fanoutVSS}$,
- Q_{CapVDD} for discharging the capacitance $C_{fanoutVDD}$,
- $Q_{CapIntern}$ for charging gate-internal capacitances (including junction capacitances of reverse biased diodes),
- Q_{in} for charging respectively discharging capacitances between the gate’s inputs and gate-internal nodes respectively V_{DD} or V_{SS} - these charges are also flowing through the driving gate, where they are considered as fanout charge,
- the short circuit charge Q_{SC}.

For a falling transition at the gate’s output and a rising transition at the causing input (confer lower part of Figure 35), the same charge components do occur as for the above discussed case. The main difference is, that the capacitances which have been charged for the above case, are now discharged respectively vice versa.

In a real cell layout further capacitances between arbitrary nodes can be extracted. In the above model (confer Figure 35) all capacitances are located between an arbitrary node and either V_{DD} or V_{SS}. If the input voltage of a gate changes from V_{DD} to V_{SS}, the actual voltage changes across the capacitively coupled nodes are different from V_{DD}. This fact will be further investigated in Chapter 3.5.1 for fanin capacitances.

For a complete cycle (a complete falling and a complete rising output transition), which is caused by complete transitions at the same input, most capacitances are once charged and once discharged. For the electrical energy *consumption* two basic definitions can be distinguished:

a) the energy is consumed as soon as it is drawn from the voltage supply, because even the part, which is stored inside the circuit's capacitances, won't be returned to the power supply later for common circuit design styles (i.e., except adiabatic circuits),

b) the energy is consumed, when it is actually turned into heat - i.e., part of the energy during the charging and the remaining part during the discharging process.

If the analysed time interval contains the same number of complete falling and rising transitions at a gate's output, which are caused by the same input, the energy consumption is equivalent for both definitions. For a single output transition of one gate, the energy consumption according to the above two definitions differ significantly. However, if the energy consumption of a large part of a circuit over a long time interval is focused on, these differences tend to *average out*. On the one hand the total number of energy causing transitions at a specific circuit-node (respectively net) during the whole simulation interval is typically high and on the other hand the circuit contains quite a few energy consuming gates.

If the distinction between the two definitions of energy consumption is not important, the two lumped capacitances towards V_{DD} and V_{SS} can be joined to a single capacitance between the output node and V_{SS} (confer Figure 36).

The charge Q_{in} is considered as fanout-charge for the driving gate and hence is not associated with the analysed gate.

3.5.1 Determination of capacitances

The lumped capacitance C_{fanout} basically consists of consecutive gate's fanin capacitances, diffusion capacitances of the drain regions connected to the output and interconnection (routing) capacitances.

The interconnection capacitance may approximately be regarded as a fixed capacitance, which can be extracted from layout. Fringing fields, that occur at the edges of the conductor due to its finite thickness, may degrade accuracy [West93]. The accurate extraction of capacitances between wires on the same layer is also a complicated task. As within delay and power models these capacitances are lumped into capacitances towards V_{SS}, some inaccuracies result from cross talk capacitances, if the signals of coupled interconnects change simultaneously in the same direction.

Besides the interconnection capacitance the fanin capacitances (of consecutive gates) contribute to the fanout capacitances. The physical device capacitances of MOS transistors depend on their operation points, as the channels may only serve as plates within the capacitance model, if the channels actually exist. I.e., during a transition the capacitance, which a driving gate has to

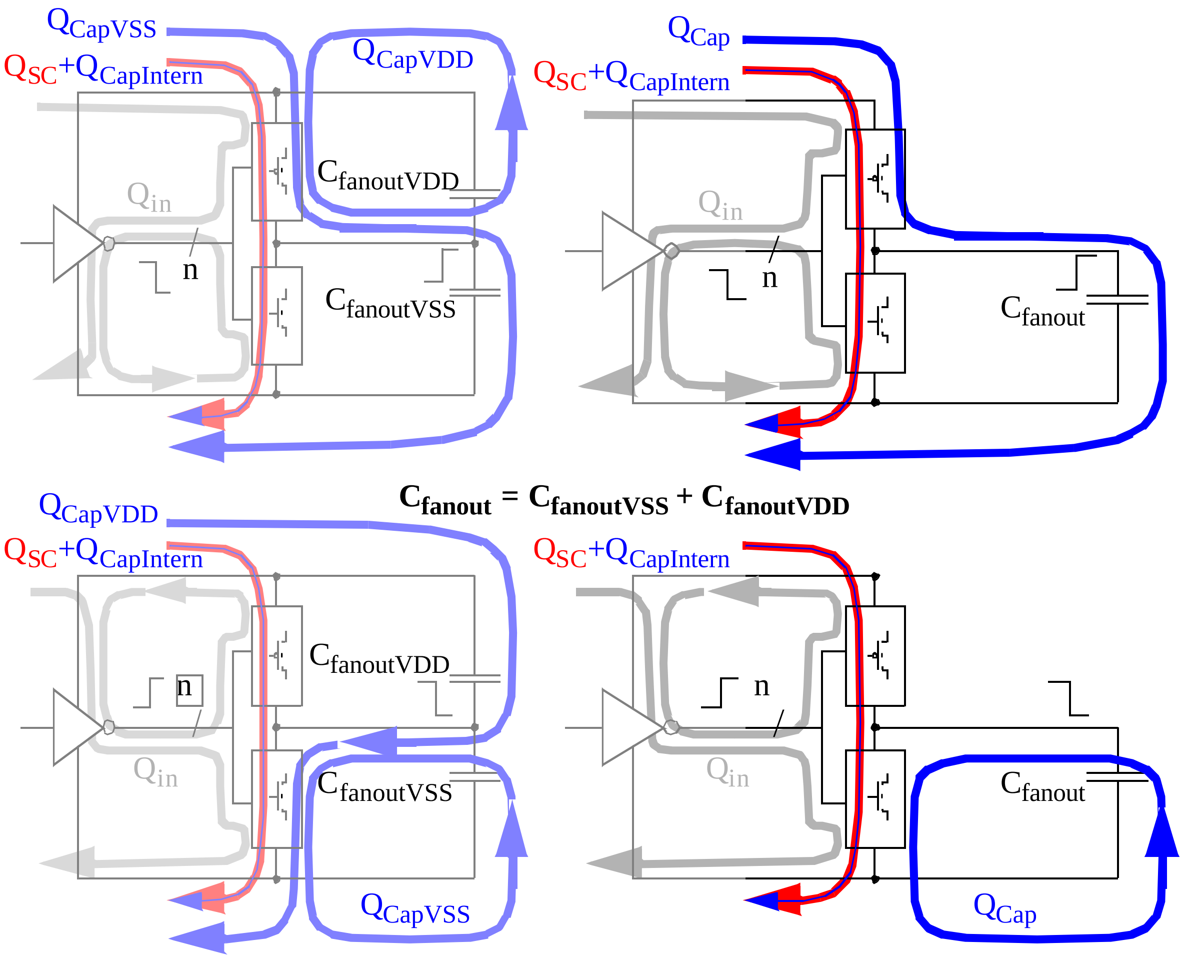


Figure 36:The two lumped capacitance towards V_{DD} *and* V_{SS} *may be joined.*

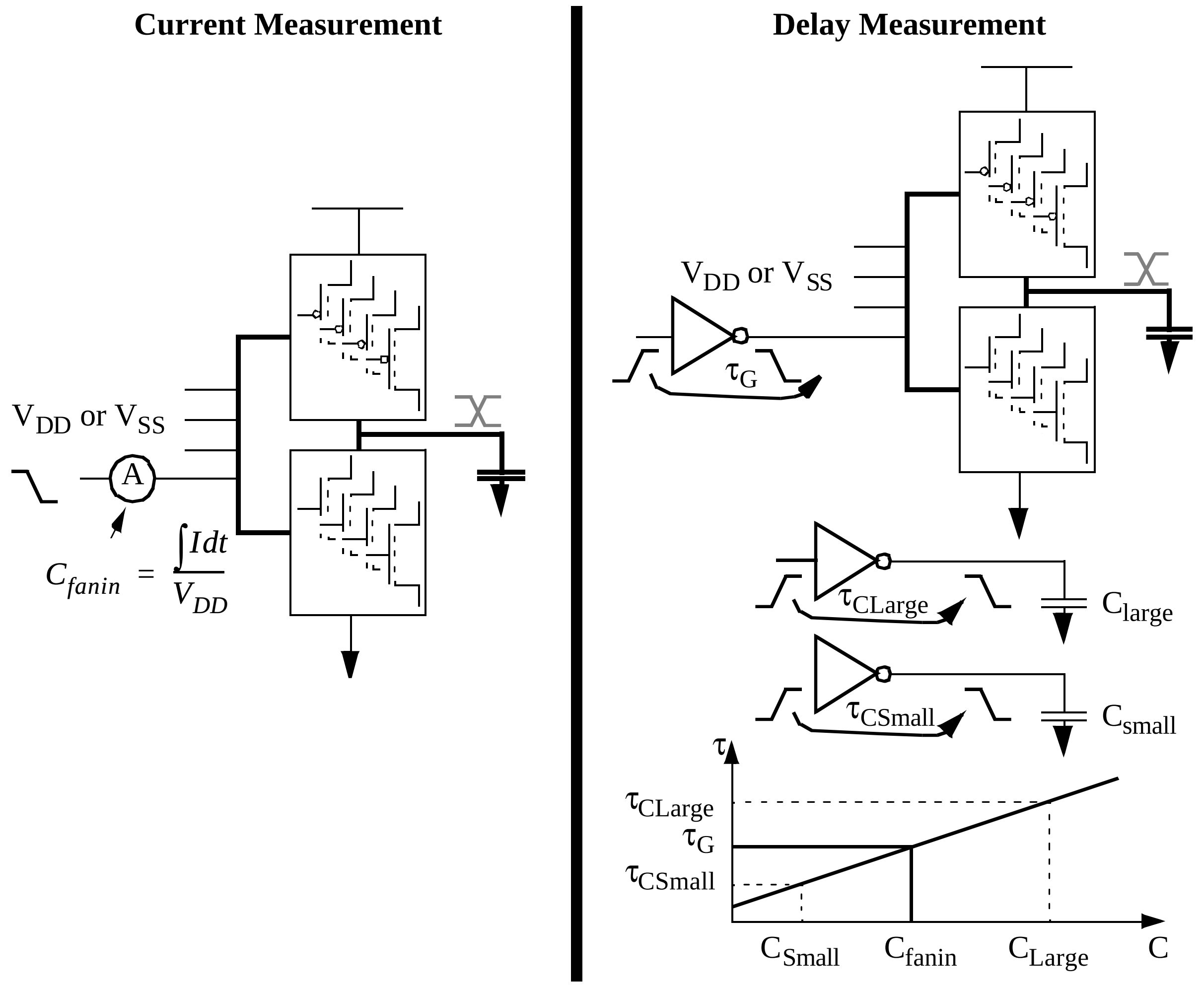
charge respectively discharge, is varying during the transition, which impacts the driver's output waveform and as a consequence its delay [West93].

The variation of fanin capacitances have been investigated by means of circuit level simulation. The characterization/testbench is introduced now. Within fanin characterization two different methodologies are distinguished (confer Figure 37):

- Current Measurement: the charge through the input terminal is measured (circuit level simulation) and divided by V_{DD},
- Delay Measurement: the fanin capacitance is derived from delay measurements of an arbitrary gate for different fanout loads:
  - large fanout capacitor C_{large},
  - small fanout capacitor C_{small},
  - the gate's input pin under test.

For the delay measurement method it is assumed, that a linear relation between the gate's fanout capacitance and its delay exists. Hence the delay measurement of two known fanout capacitances defines the parameter of a linear function. After measuring the delay of the gate's input under test, the corresponding (effective) fanin capacitance can be read from the diagram (confer Figure 37) respectively calculated from the functional description.

The two alternative characterization methods deliver different results. The current measurement method is best suited for power characterization and the delay measurement method is



Current Measurement

Delay Measurement

Figure 37:Fanin characterization methods.

best suited for delay characterization. Within this thesis the current measurement method is used for the testbench of the following fanin capacitance analysis and library characterization.

Besides these physical variations of the fanin capacitance, the simple black box approach assumes a constant fanin capacitance, which is lumped towards a fixed potential (typically V_{SS}, confer right part of Figure 36). The voltage swing across the actual physical capacitances commonly differs from V_{DD}. The exact voltage swings depend on the state of other transistors, which are part of the CMOS stage, and possibly their history. These effects are exemplified using the NOR3-cell transistor schematic, which is shown in Figure 38. This model includes capacitances between almost each pair of nodes. The sizes of the capacitances are derived from layout extraction.

A very big portion of the internal capacitances results from the connection of the various transistors on the diffusion layer. The diffusion can be modelled as a diode that is commonly not conducting, i.e. the diodes behave like non-linear capacitances. The capacitances C_{T2}, C_{T4}, C_{outVSS} and C_{outVDD} mainly consist of these connection-diffusion-diodes. The gate-capacities of the MOS transistors are covered by the transistor-models.

For this discussion the following assumptions are made:

- a transition at a cell's output is caused by a transition of a single input pin (i.e. glitches are not taken into account)

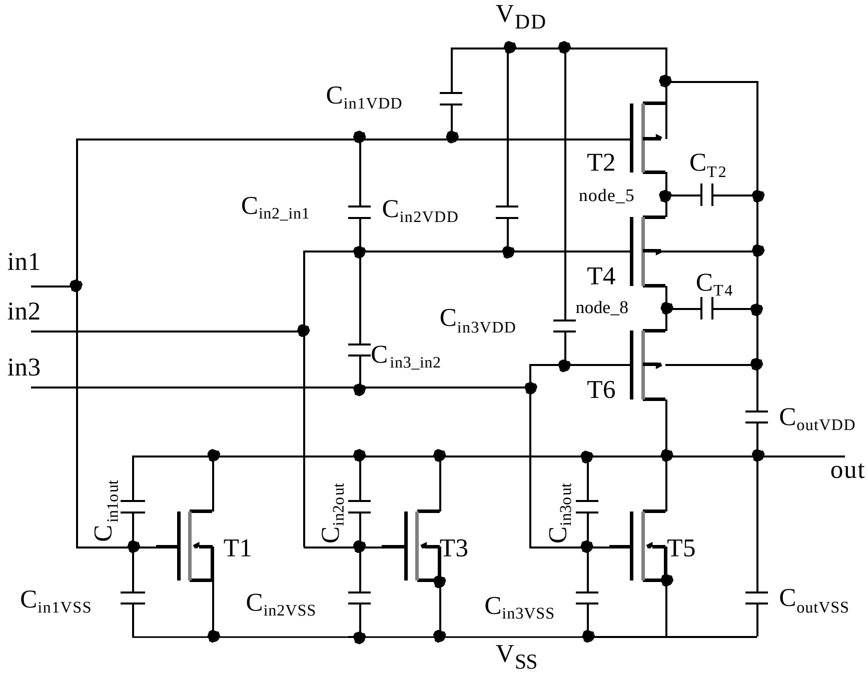


Figure 38:Transistor schematic of a NOR3-cell.

- the voltages at the inputs are always VDD or VSS when they are in a steady state,
- the voltage VDD and VSS is constant over time and
- each instance of a circuit is supplied with the same voltage.

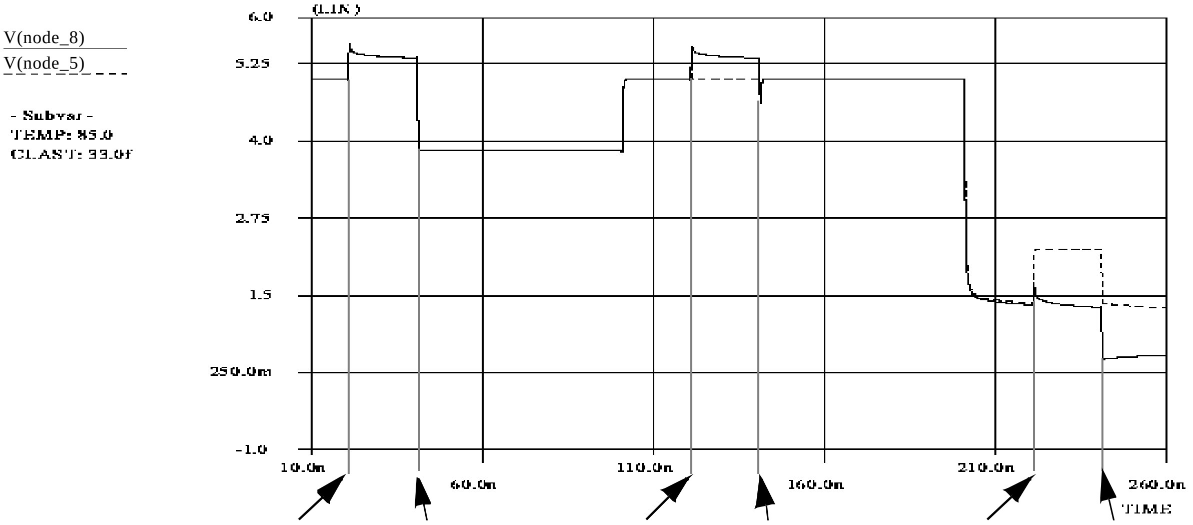
To assess the relevance of different effects, the fanin capacitance of a NOR3-cell’s input pin in2 is analysed in detail (confer Figure 38). The cell is taken from an industrial library of an 800nm technology. The fanin-capacity is specified with 41fF in the library datasheets. In Figure 39 the voltage at the internal nodes node_5 and node_8 is shown for different situations. Neither of these 6 situations (confer Figure 39) leads to a transition at the output. Under these conditions the fanin-capacity varies up to 30% and hence needs further consideration.

The total fanin-capacity can be divided into three components:
- fanin capacity of the n-block (pull down network),
- fanin capacity of the p-block (pull up network) and
- cross capacity towards the other input pins.

These contributions are observed by measuring the charge through the VDD-, the VSS- and the input pin-terminals within the SPICE simulation.

For the n-block contribution of this example only the transistor T3 is switching. Both, V_{out} and of course V_{SS} are constant for these 6 situations. Hence the drain- and source-voltage of T3 is not changing and its contribution to the total fanin-capacity is the same for each case (16.5fF).

During the first two transitions of in2 the transistors T2 and T6 are not conducting and hence node_5 and node_8 are only coupled with V_{DD} via C_{T2} and C_{T4} while T4 is switched off. The rising input slope at 20ns raises the voltage in the channel and at node_5 and node_8 up to about 5.5V - a higher voltage is not possible because the diffusion-diodes will start conducting. This voltage-level slowly decreases to a stable value of about 5.3 V and the gate capacity remains partly charged. The falling input slope at 40ns draws the voltage at node_5 and node_8 immediately down to about 3.9V. Hence the gate-capacity is not discharged completely. C_{fanin} respectively the contribution of the capacity towards VDD is about 4.5fF smaller



V(in2)		rising	falling	rising	falling	rising	falling
V(in1)		V_{DD}	V_{DD}	V_{SS}	V_{SS}	V_{DD}	V_{DD}
V(in3)		V_{DD}	V_{DD}	V_{DD}	V_{DD}	V_{SS}	V_{SS}
fanin-Capacitances in fF	total	40.5	36	40.5	40.5	30.5	30
	to VDD	17.5	13	16	16	5.5	7
	to VSS	16.5	16.5	16.5	16.5	16.5	16.5
	to in1+in3	7	8	6	6	6.5	6.5

Figure 39:Fanin-Capacities in dependence of different internal situations.

than for the rising slope at 20ns. It has to be noted, that the situation before 20ns (all p-transistors turned off and 5V at node_5 and node_8) cannot be reached during operation. If a node is isolated from V_{DD} by a transistor turning off, its voltage is drawn down by the falling input-slope and hence its voltage is always lower than V_{DD}. This situation only occurred due to the static initialization by the circuit level simulator at 0ns.

During the transition of in2 at 120ns and 140ns the transistor T2 is conducting and T6 is not. The voltage-swing of node_8 is the same as before at 20ns. The falling transition draws the voltage at node_8 down for a short period (until the channel of T4 is build up). Because the voltages at node_5 and node_8 are the same before the rising and after the falling slopes, all measured capacities are the same.

During the transition of in2 at 220ns and 240ns the transistor T6 is connecting the output with node_8 and T2 is not conducting. The capacities measured for these transitions are much smaller than before. This is due to the non-linear gate-capacity, which is much smaller if the voltage-level in the channel is less than V_{DD}. The total fanin's contribution towards VDD is very small, because only the gate-source-capacity of T4 is discharged via V_{DD} and the gate-drain-capacity not.

If an other input (in1 or in3) toggles, the number of differing voltage-combinations at source- and drain is much smaller. Hence the fanin-capacities almost don't vary at all.

The capacitances between the inputs are given in the last row of Figure 39. They don't vary significantly for different cases.

All situations, that have been discussed so far, don't result in a transition at the output of the cell. The fanin-loads of the switching NOR3-cell are significantly higher (48.5fF for in2) for the following reasons:

- the capacities between the switching input pin and the output pin is charged by a voltage-swing of 2-times V_{DD} and
- all gate-capacities of the pull up network (for the NOR-3 cell) are at least completely charged respectively discharged

For a rising output-slope of the NOR3-cell, the n-transistors' drain-voltage of the switching transistor is at V_{SS} before the transition and at V_{DD} afterwards. Hence the gate-drain capacitance (Miller capacitance) of the switching transistor is charged with a voltage swing of $2*V_{DD}$. Similarly the drain of the switching p-transistor is charged by a swing of $2*V_{DD}$ respectively $2*V_{DD}-V_T$.

The maximum possible range of fanin capacitances is between 30fF and 48.5fF. I.e., if the worst case (i.e. maximum) capacitance is taken as a fixed fanin capacitance the actual capacitance may be up to 38% lower for some cases. This large deviation is important for power and delay calculation. As a fixed value is typically taken for the fanout load of a gate, this is a source of error when comparing gate level simulation results to circuit level simulation results. However, for future designs the contribution of input capacitances to the total fanout capacitance of a gate is decreasing [Veen98].

From the power consumption's point of view, it is also important, that a changing internal node voltage may results in charge flow through not transitioning input pins. An example is given within the above discussed NOR3 testcase. At 100 ns (confer Figure 39) a falling transition is applied to input in2. Consequently node_5 and node_8 are connected to V_{DD} and the voltage rises. Hence the gate-drain and the gate-source capacitances of T4 are charged.

The exact fanin capacitance for delay determination and power calculation is hard to obtain. The reason is, that the fanin capacitance is typically considered as part of the fanout capacitance of a driving gate. This has the following implication for a correct modelling of fanout capacitances:

- The correct consideration of all consecutive gates' fanin capacitances as part of a fanout capacitance requires the knowledge of all node voltages (internal and external) of the driven gates,
- the fanin capacitance's contribution to a fanout capacitance cannot be determined a priori, as the node voltages of consecutive gate's are not constant and consequently the fanout capacitance needs to be calculated on the fly.

Even if a simulator allows the evaluation of signals in consecutive gates, the consideration of these signals requires a lot of effort and would undoubtedly significantly slow down the simulation.

In conclusion it has to be kept in mind, that the fanin contribution to a gate's fanout capacitance may vary significantly (up to approximately 38% for the analysed example, if the maximum fanin capacitance is characterized). From the power consumption's point of view, charges through an input pin may even be caused by transitions at other input pins. Within the library characterization, which was needed for the simulator GliPS, the worst case fanin capacitances were characterized (switching output).

3.5.2 Capacitive power consumption

After the discussion of how to determine a gate's fanout capacitance respectively the fanin capacitance of consecutive gates, it is now analysed how much energy is consumed to charge them.

Capacitive power is consumed, if the voltage over a capacitance changes. A simple switch model of a single CMOS stage is illustrated in Figure 40. The MOS transistors are each replaced by a switch and a resistor. The pull up (pull down) switch is open (closed) before t_0 and closes (opens) for a period Δt at $t = t_0$. This causes the output voltage $V_{out}(t)$ to rise from 0 to ΔV. Hence, the charge flow Q can be calculated as follows:

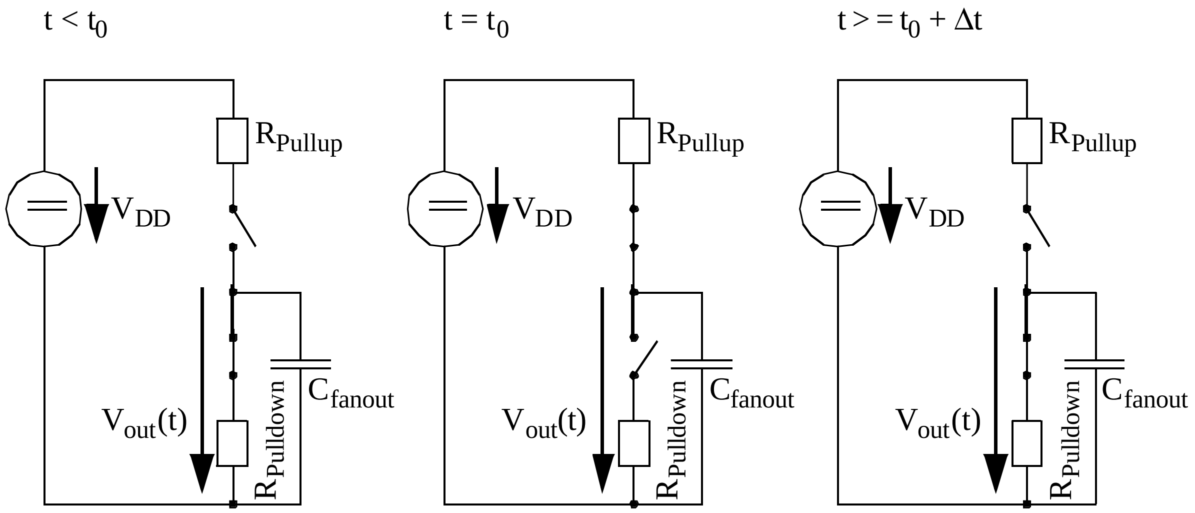


Figure 40:Switch modelling of a CMOS stage.

$$Q = C_{fanout} \cdot |\Delta V| \tag{20}$$

The total energy, which is supplied by the voltage source, is given by

$$En = Q \cdot V_{DD} = C_{fanout} \cdot |\Delta V| \cdot V_{DD} . \tag{21}$$

Part of the energy is stored within the capacitance C_{fanout} (En_{cap}) and part of it is turned into heat within the pull up resistance ($E_{Rpullup}$):

$$\begin{aligned} En_{cap} &= \frac{1}{2} \cdot C_{fanout} \cdot \Delta V^2 \\ En_{Rpullup} &= C_{fanout} \cdot |\Delta V| \cdot V_{DD} - \frac{1}{2} \cdot C_{fanout} \cdot \Delta V^2 = C_{fanout} \cdot \Delta V^2 \cdot \left(\frac{V_{DD}}{|\Delta V|} - \frac{1}{2}\right) \end{aligned} \tag{22}$$

At $t = t_0 + \Delta t$ the pull up switch is opened and the pull down switch is closed again. The energy En_{cap}, which was previously stored within the fanout capacitance, is now turned into heat within the pull down resistance ($En_{Rpulldown}$).

$$En_{Rpulldown} = \frac{1}{2} \cdot C_{fanout} \cdot \Delta V^2 \tag{23}$$

Hence during a whole cycle the energy En (Equation 21) is drawn from the voltage source and turned into heat.

For a complete V_{DD} swing ($|\Delta V| = V_{DD}$) the energy En_{cap}, $En_{Rpullup}$ and $En_{Rpulldown}$ are equivalent:

$$En_{Rpullup}\big|_{\Delta V = V_{DD}} = En_{cap}\big|_{\Delta V = V_{DD}} = En_{Rpulldown}\big|_{\Delta V = V_{DD}} = \frac{1}{2} \cdot C_{fanout} \cdot V_{DD}{}^{2} \tag{24}$$

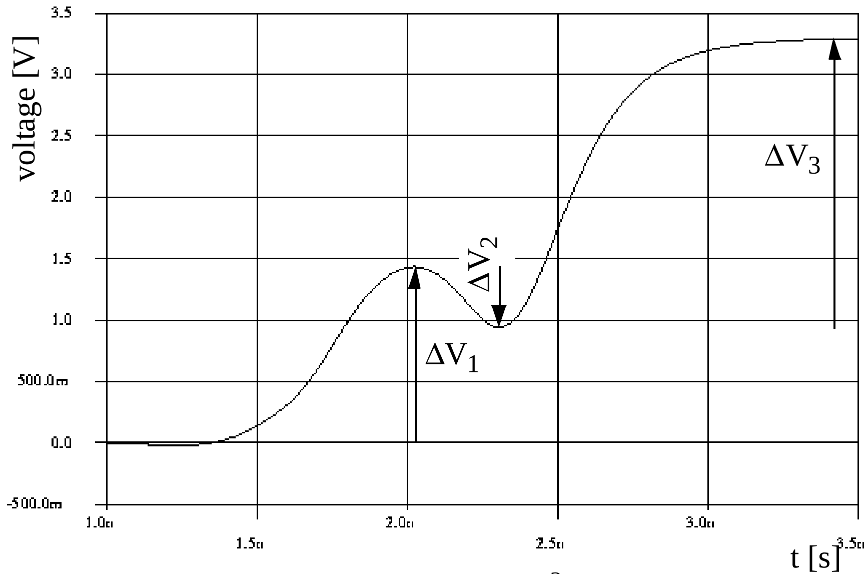
The actual energy, which is turned into heat respectively drawn from the voltage supply for charging and discharging the fanout capacitance, is equivalent to

$$En = En_{Rpullup} + En_{Rpulldown} = C_{fanout} \cdot |\Delta V| \cdot V_{DD}. \tag{25}$$

According to the above discussion, the energy consumption of this whole cycle can approximately be divided into two equal parts which are associated with each voltage swing ΔV:

$$En_{Rpullup} \approx En_{Rpulldown} \approx \frac{1}{2} \cdot C_{fanout} \cdot |\Delta V| \cdot V_{DD} \tag{26}$$

This equation can also be applied to calculate the capacitive energy consumption of more general waveforms (confer Figure 41).



$$En = \frac{1}{2} \cdot C_{fanout} \cdot V_{DD} \cdot \sum_{i=1}^{3} |\Delta V_i|$$

Figure 41:Example for a dynamic glitch.

The energy, which is consumed during a given period τ is given by:

$$En = \frac{1}{2} \cdot C_{fanout} \cdot V_{DD} \cdot \sum_{T} |\Delta V_i| = \frac{1}{2} \cdot C_{fanout} \cdot V_{DD}^{2} \cdot \sum_{\tau} \frac{|\Delta V_i|}{V_{DD}} \tag{27}$$

The term $\sum_{T} |\Delta V_i|$ is the sum of all voltage changes within the period T.

The capacitive power calculation is straight forward:

$$P_{Cap} = \frac{1}{2} \, x \, V_{DD} \, x \, C_{fanout} \, x \lim_{\tau \text{ ® } \text{¥}} \frac{\sum_{i} |DV_i|}{\tau} \tag{28}$$

Let α be the average number of transitions within one clock cycle (glitches are counted according to their fractional voltage swing with respect to V_{DD}):

$$\alpha = \frac{1}{V_{DD}} \cdot \lim_{\tau \to \infty} \frac{\sum_i |\Delta V_i|}{\tau} \cdot T = \frac{1}{f} \cdot \lim_{\tau \to \infty} \frac{\sum_i \frac{|\Delta V_i|}{V_{DD}}}{\tau} \tag{29}$$

The sum $\sum|\Delta V_i|$ respectively the term α can be obtained by logic simulation over a *sufficient* time-interval (confer Chapter 4.1.4) using an appropriate model. By combining Equations 28 and 29, the following common term for capacitive power consumption is obtained:

$$P_{Cap} = \frac{1}{2} \cdot V_{DD}^2 \cdot C_{fanout} \cdot \alpha \cdot f = \frac{1}{2} \cdot V_{DD}^2 \cdot C_{eff} \cdot f \tag{30}$$

- α is the average number of transitions within one clock cycle,
- f is the clock frequency and
- C_{eff} is the effective switched capacitance per clock cycle (unit is F).

The capacitive power consumption of a complete integrated circuit or a specific part of it is calculated by summing up the power consumption of each capacitive contributor:

$$P_{Cap\_total} = \frac{1}{2} \cdot V_{DD}^2 \cdot f \cdot \sum_i C_{fanout\_i} \cdot \alpha_i = \frac{1}{2} \cdot V_{DD}^2 \cdot f \cdot \sum_i C_{eff\_i} \tag{31}$$

Sometimes the term effective capacitance is defined to represent the total switched capacitance of the analysed part of the circuit (i.e. $\sum_i C_{eff\_i}$). However, here the definition is used as introduced above.

Within this thesis high emphasis is put on correctly considering glitches for power calculation. Therefore the partial voltage swings must be considered in the power formula.

3.5.3 Short circuit power consumption

During switching (of a CMOS stage) a conducting path through the pull up and pull down network of a gate is present and as a consequence a short circuit current is occurring. The time interval during which a short circuit current occurs depends on the voltage waveform of the switching input signal. A short circuit current is occurring for input voltages in the range $[V_{THN}, V_{DD}+V_{THP}]$ (confer Figure 42). From [Veen84] Equation 32 can be derived for the short circuit energy of a single complete transition under the following assumptions for an inverter:

- the inverter is symmetrical,
- the inverter's fanout load is zero and
- the input voltage is linearly rising:

$$\begin{aligned} En_{SC}(1\ \text{transition}) &= \frac{\beta}{24} \cdot (V_{DD} - 2V_T)^3 \cdot \tau_{SI} \\ \beta &= \frac{C_g \cdot W \cdot \mu}{l_{tr} \cdot A} \quad \text{(gain factor of a MOS transistor)} \end{aligned} \tag{32}$$

τ_{SI} rise or fall time of the input signal (here from 0 to 100% of V_{DD})

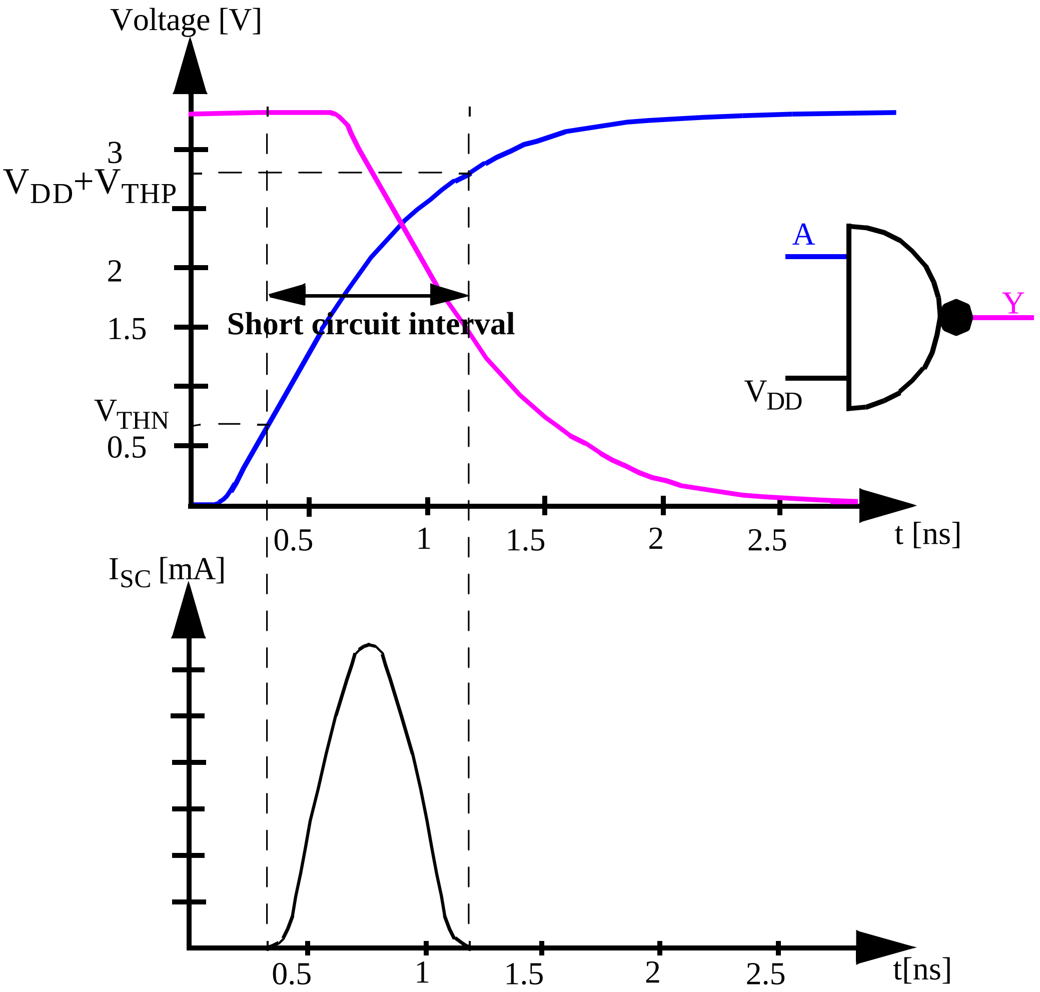


Figure 42:Short circuit current as a function of the input waveform.

Similar to Equation 30 the short circuit power consumption can be calculated for equal rise and fall times:

$$P_{SC} = \frac{\beta}{24} \cdot (V_{DD} - 2V_T)^3 \cdot \tau_{SI} \cdot \alpha_{FullTr} \cdot f \qquad (33)$$

The term α_{FullTr} is used, to point out, that the equation does not hold for glitches. The main reasons are:

- the short circuit current is associated with the time, in which the input voltage (and not the output voltage) is in the short circuit interval (see Figure 42) and
- the short circuit current is a non linear function of the input voltage.

Further effects will be discussed in the following subchapters.

The short circuit power consumption of a complete integrated circuit or a specific part of it is calculated by summing up the power consumption of each contributor:

$$P_{SC\_total} = \sum_i P_{SCi} \qquad (34)$$

In general the above assumptions are not met in a real circuit. The most important impact has the capacitive charging and discharging current waveform through the block, which is turning on (for fanout load > 0). For this reason the short circuit current is hard to determine within a simple expression like Equation 32. If capacitive effects are considered, according to [Hede87] the short circuit power is only 30% compared to not considering these effects for typical cases (i.e. equal input and output slopes). Within this thesis this component has been investigated by means of circuit level simulations of complete gates.

3.5.3.1 Testbench for short circuit charge extraction

Within this subchapter a testbench is introduced for extracting the short circuit charge from the flowing charges monitored at a cell’s terminal nodes. The cells’ circuit level descriptions which are used within this investigations were extracted from an industrial 0.5µm-CMOS library’s layout (V_{DD}=3.3V). The extraction of the short circuit component is not trivial because of the cell’s internal effects and the high number of overlying currents. For this reason special emphasis is put on this subject.

The charge flows, which are associated with a single gate have already been discussed above (confer Figure 35 respectively Figure 43). The pull up and pull down network contain numerous capacitors, diodes, MOS transistors which are part of the cell’s circuit-level description. The fanout capacitance is lumped into a single capacitor C_{fanout}. The different charges have already been explained above. Please note, that discharging currents are not drawn from the power supply. However they must be considered, when Q_{SC} is extracted from the charges through the V_{DD}- or the V_{SS}-terminal by means of circuit-level simulation.

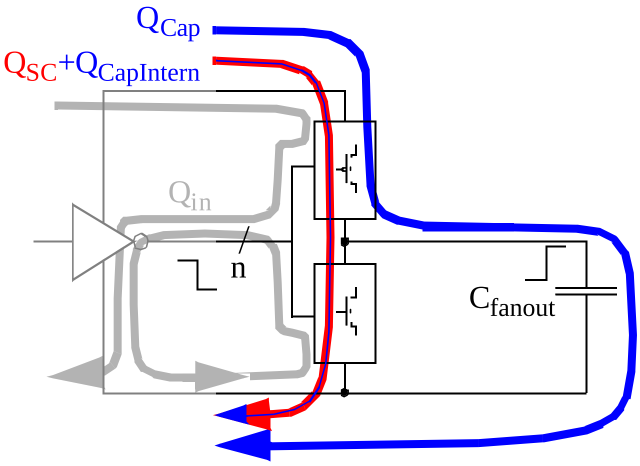


Figure 43:Abstract view of a single stage static CMOS cell.

The short circuit charge Q_{SC} is extracted by the following procedure (the subscripts $_{\_fall}$ and $_{\_rise}$ refer to the respective output transition of the device under test:

a) Determine Q_{ref_fall} and Q_{ref_rise} by monitoring Q_{VDD} for a falling and a rising output transition by using an extremely fast input-slope and a typical fanout capacitor:
$Q_{ref\_fall} = Q_{VDD}$
$Q_{ref\_rise} = Q_{VDD} - C_{fanout} \cdot V_{DD}$

b) Determine Q_{SC} by monitoring Q_{VDD} for the desired cell configuration (input-slope and fanout capacitor C_{fanout}):
$Q_{SC\_fall} = Q_{VDD} - Q_{ref\_fall}$
$Q_{SC\_rise} = Q_{VDD} - Q_{ref\_rise} - C_{fanout} \cdot V_{DD}$

In step a) the term $Q_{capIntern}$+Q_{inVDD} is determined (for an extremely fast input-slope the short circuit charge is close to zero). This charge is approximately identical for all circuit configurations in terms of input slope and output load, because all voltages before and after the transition are identical. Small deviations may appear due to differing overshoots (caused by input- to output-coupling). In step b) the short circuit charge is determined for the desired circuit configuration by monitoring Q_{VDD} and subtracting the other capacitive components, which have been determined in step a). The actual testbench is given in Figure 44. The charge Q_{VDD} is monitored at the cell's V_{DD}-terminal. The inverter is used to get realistic input-waveforms. Its slope is changed by choosing different values for C_{in}. However, for the reference simulation the input of the gate under test is directly driven by an extremely fast input-slope (about 50ps rise respectively fall time).

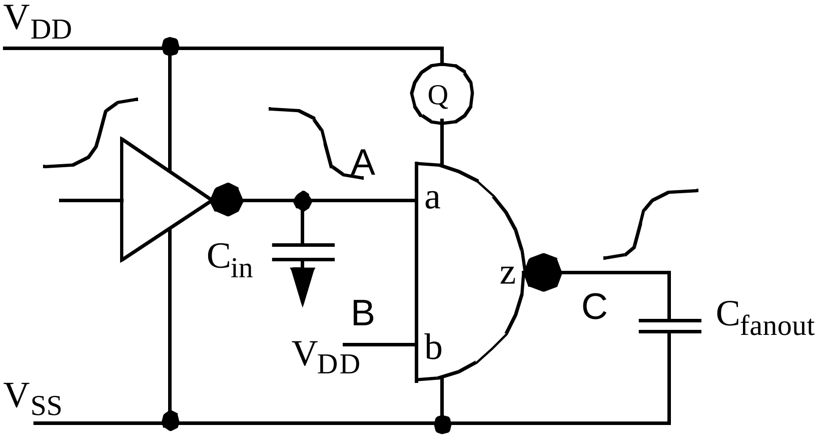


Figure 44:Testbench for single complete transitions.

The extraction of the short circuit charge for glitches is similar to the extraction for single complete transitions. However, the reference charges need to be determined for each glitch-peak voltage at the output. It is not necessary to determine the reference charge as a function of the internal node-voltages if they are identical for the reference and investigated simulation configuration before and after the glitch. Small deviations are again possible for different overshoots due to input to output coupling.

For extracting the short circuit charge Q_{SC} of glitches the following procedure is done:

a) determine Q_{ref} as a function of the glitch-peak voltage at the cell's output by monitoring Q_{VDD} using extremely fast input-slopes with different skews and a typical output load:
$Q_{ref}(\Delta V) = Q_{VDD} - C_{fanout} \cdot \Delta V$

b) determine Q_{SC} by measuring Q_{VDD} for the desired simulation configuration:
$Q_{SC} = Q_{VDD} - Q_{ref}(\Delta V) - C_{fanout} \cdot \Delta V$

For glitch propagation the simulation configuration are the width of the incoming glitch, its peak voltage and the cell's fanout load. The width of the glitch is defined as time between crossing the voltage V_{DD}+V_{thp} for VDD-VMIN-VDD-glitches and V_{SS}+V_{thn} for VSS-VMAX-VSS-glitches. If these points are not crossed, no short circuit current will occur. The above procedure, which has been introduced for determination of short circuit charges due to generated glitches, also holds for propagated glitches if no glitches at internal cell-nodes occur. This is true for cases in which the incoming glitch controls the input that has its MOS transistor's drain directly connected to the cell's output (for single stage gates). I only investigated cases here, which do not deliver internal glitches because this would only complicate the short circuit charge extraction without delivering significantly different results.

The testbench for glitch generation is shown in Figure 45 (left part of the figure). At the output of the gate under test (NAND-2) different glitches are generated by varying C_a and C_b and the skew between the two colliding input transitions at a and b. The charge Q_{VDD} is monitored. However, for the reference simulations the inputs of the gate under test are directly driven by two extremely fast input-slopes (about 50ps rise respectively fall time), whose skew is varied to achieve different glitches at the output.

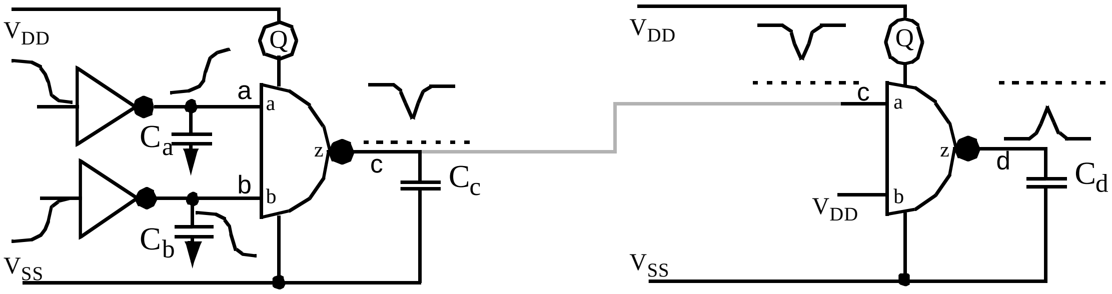


Figure 45:Testbench for generated glitches (left figure) and for propagated glitches (testbench on the left drives the gate on the right).

For glitch-propagation an incoming glitch is needed. This glitch is generated by the testbench for glitch generation and (possibly) propagated through the gate under test (confer Figure 45). The charge Q_{VDD} is monitored. For the reference simulations one rising and one falling extremely fast complete input-slope are applied to the input pin. The skew between these two input slopes is varied to achieve different glitches at the output d.

As a typical example for simulation purposes I focus on a NAND2-gate of an industrial 3.3V-library. Using the testbenches and the procedures introduced in this subchapter short circuit charges were extracted for glitch free cases, glitch generation and glitch propagation. The results are presented and discussed in the following subchapters. The behaviour of other single stage CMOS gates is similar, because in case of output transitions a pair of n- and p-channel transistors switches. Any other transistor in parallel (series) will have higher (lower) impedances during switching, because otherwise the output would not switch.

3.5.3.2 Simulation results for glitch free cases

Generally the time a short circuit path is present through the pull up and pull down network is proportional to the input-slope's steepness. On the other hand the charge and discharge of fanout capacitance C_{fanout} and internal capacitances limits the short circuit charge Q_{SC}. I.e. for a fixed input-slope the short circuit charge decreases with an increasing fanout capacitor C_{fanout}. This is exemplified by the simulation results in Figure 46. The short circuit charge's contribution to the total charge, which indicates the error if the short circuit power consumption is neglected within power calculation, is shown in Figure 47. For this comparison the capacitance C_{fanout} was divided into 2 equal parts: one towards V_{DD} and one towards V_{SS}, which is more realistic than lumping the whole capacitance towards V_{SS}. Hence only half of C_{fanout} is charged during a falling respectively a rising output-slope. The short circuit's charge-contribution ranges from 90% for slow input-slopes and C_{fanout}=0 to approximately 0% for fast input-slopes and/or high values of C_{fanout}. The cases with equal input-rise and output-fall

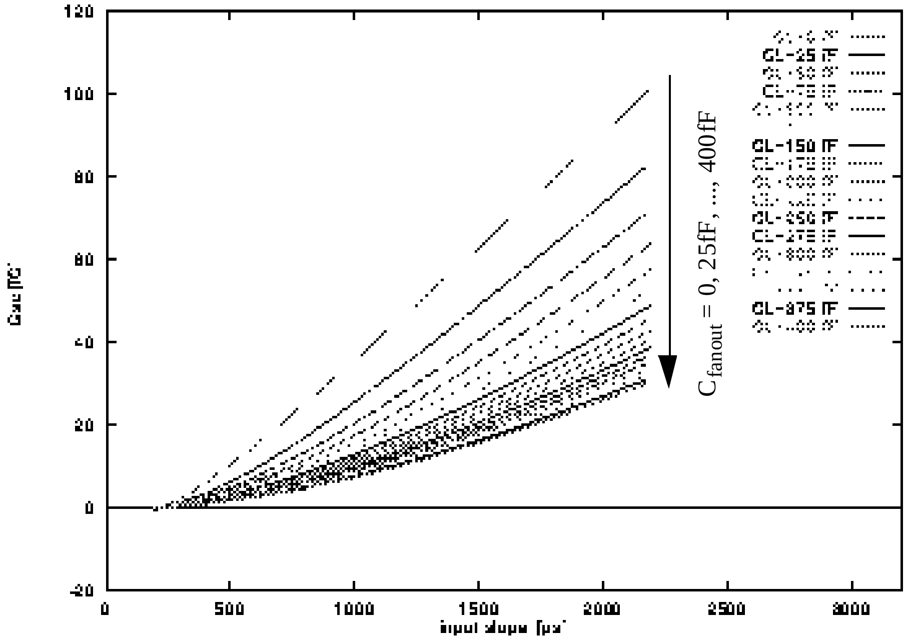


Figure 46:Short-circuit charge for a falling output-slope caused by input A.

times result in contributions of about 3% of the total charge. These results are in agreement with [Hede87,Veen84]. Hence the short circuit power consumption is negligible for *well designed* cases with equal input- and output- rise and fall times. However, it should be noted, that in combinational circuits it is impossible to always ensure such well designed cases due to different fall- and rise-times of a gate, different input-slopes at different input-pins and so on. Hence short circuit power consumption may not always be neglected.

3.5.3.3 Simulation results for generated glitches

For generated glitches the influence of the short circuit behaviour dramatically depends on the glitch peak voltage and the resetting input-slope. The characteristic short circuit behaviour is discussed on the basis of the simulation results (Figure 48) of the testbench (Figure 45 - left). The case C_a=100fF, C_b=300fF and C_c=150fF would result in approximately equal input-rise (input-fall) and output-fall (output-rise) times if no collision occurs.

The basic difference of the short circuit behaviour between the glitch- and glitch-free case is the role of the capacitive charging current. For glitches the resetting input-slope is in the *short circuit region* ($V_{thn}<V_b<V_{DD}+V_{thp}$) while no significant charging-/discharging current of the output load C_{fanout} is occurring, i.e. the output-voltage does not significantly change (cf. time interval τ_{SC} in Figure 49). Hence during this time the short circuit current is not limited by any charging/discharging current. For this reason the short circuit current is even higher for glitches than for the two respective non-colliding input transitions.

This basic behaviour can also be observed from the plots in Figure 48. The plot on the right shows that short circuit power consumption almost doesn’t depend on the capacitive load for glitches with glitch-peak voltages lower than 2.5 V. The great impact of the resetting input slope can be observed from the plot on the left. The impact of the setting input transition is much lower (confer lower plot). Within all plots glitches with their peak-voltage between about 1V and 2.5V have an almost constant short circuit charge (respectively power) consumption. For glitch-peak voltages lower than 1V the resetting input-slope starts before the setting

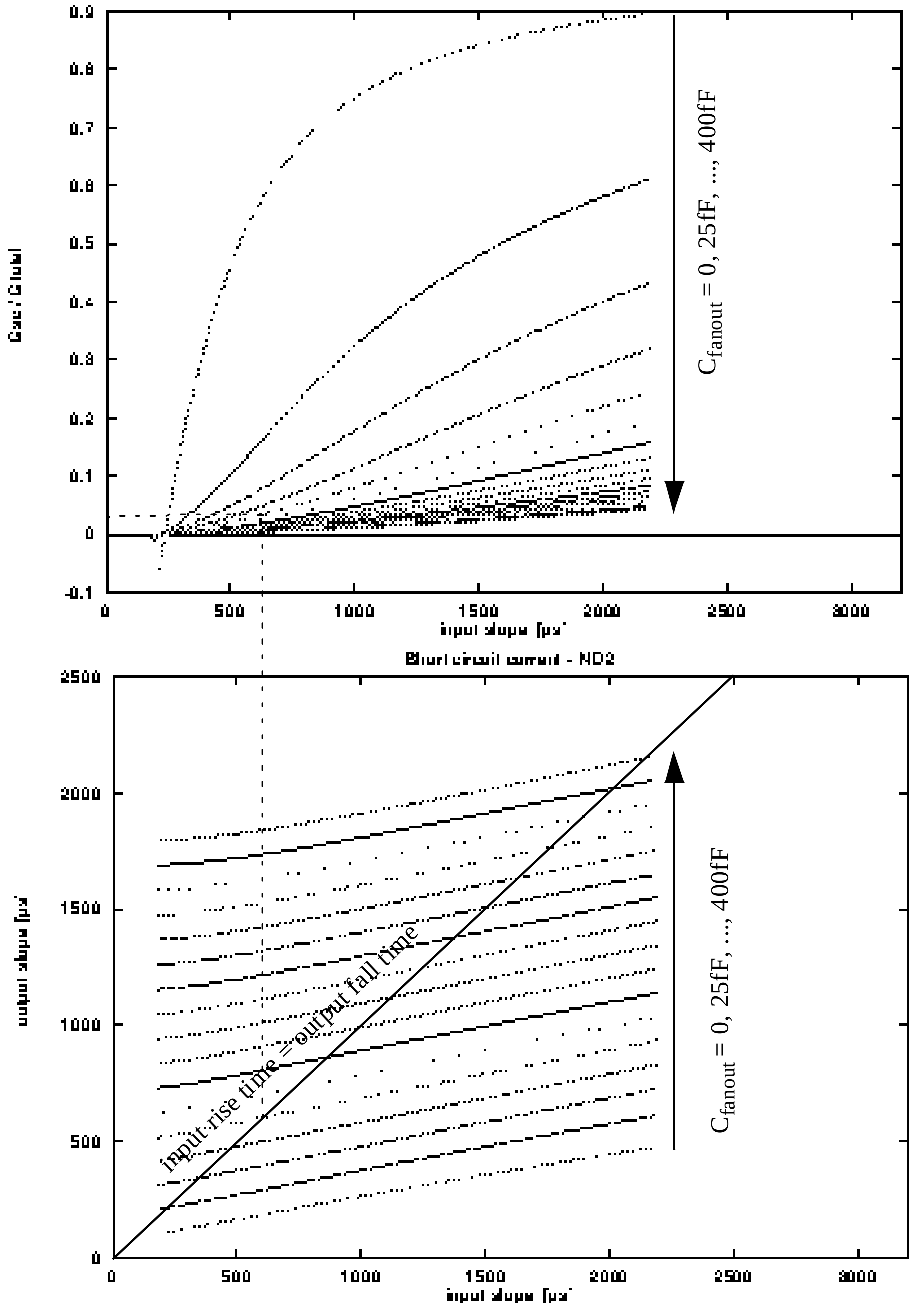


Figure 47:Short-Circuit charge's contribution to the total charge for a falling output-slope caused by input A and the output fall time as a function of the input rise time.

input-slope has reached V_{DD}+V_{thp}. I.e., that the effective impedance of the pull up network remains relatively large, because one of the two transistors in series of the pull up network is turned on and at the same time the other transistor is already turned off.

In Figure50 the relative contribution of the short circuit current is plotted for the *typical case*. The total power consumption is approximated by Q_{VDD}. The short circuit's contribution is hence significantly higher than for non colliding signal transitions (confer Chapter 3.5.3.2).

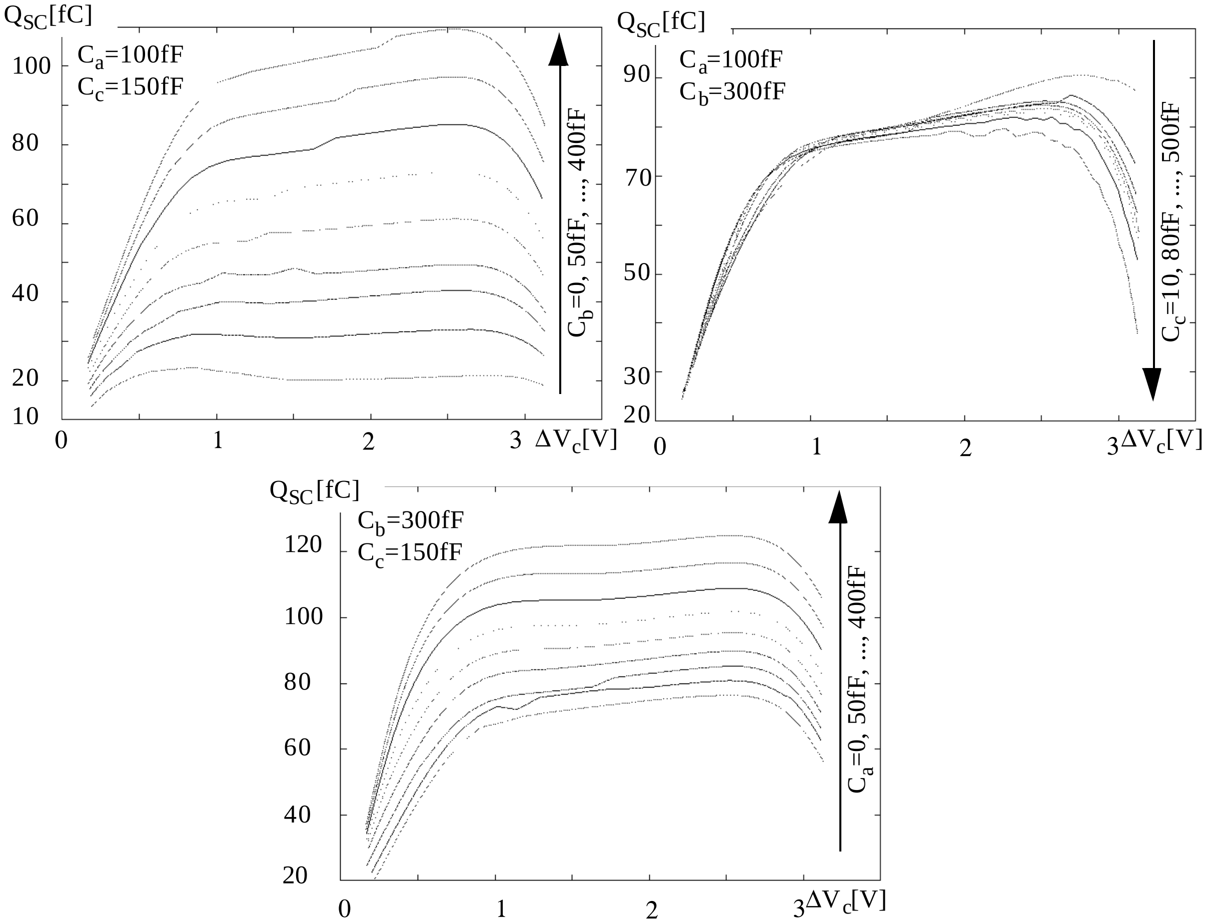


Figure 48:Short circuit charge as a function of generated glitch peak voltage.

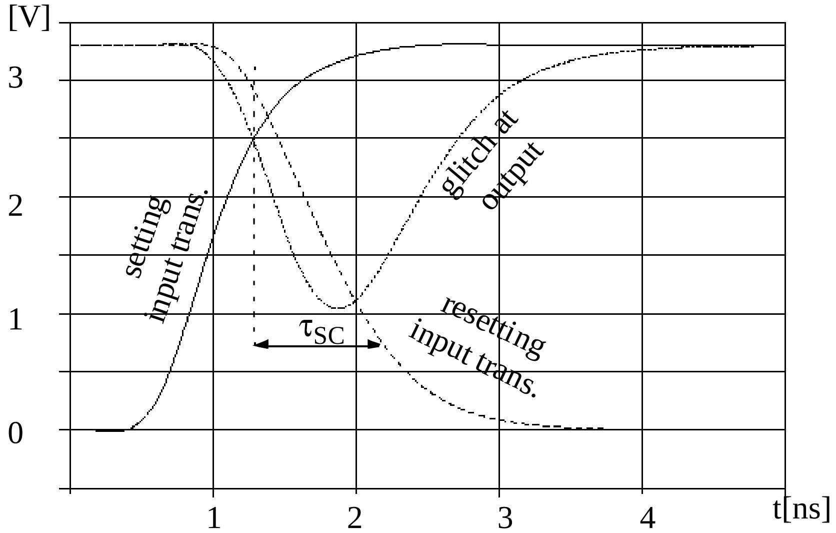


Figure 49:Region of possible short circuit currents caused by the resetting input transition.

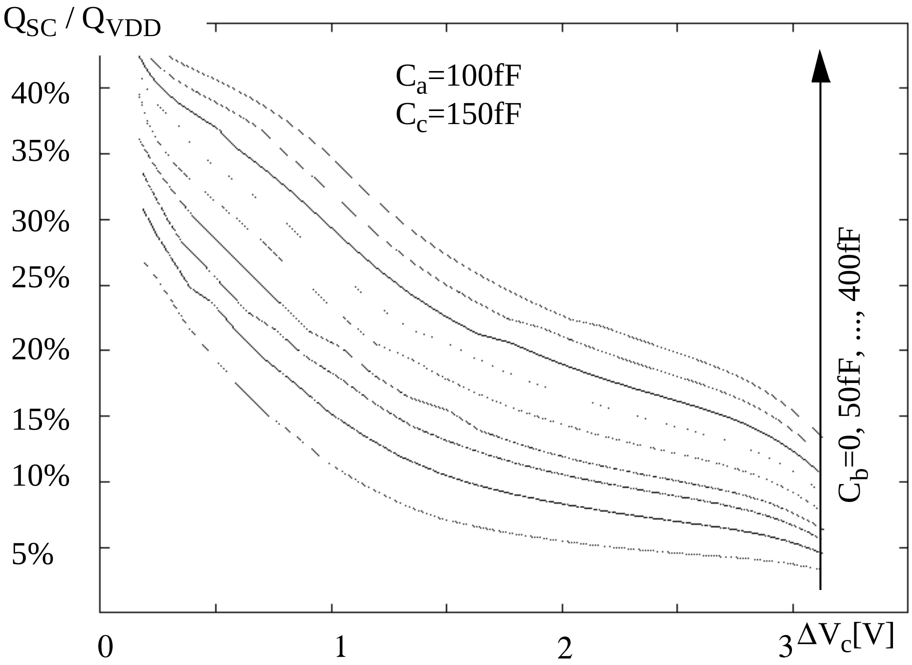


Figure 50:Relative contribution of the short circuit charge to the total charge Q_{VDD}.

3.5.3.4 Simulation results for propagated glitches

There are two reasons why the short circuit charge of a propagated glitch might be significantly higher than for two complete transitions at the input-pin. On the one hand the input-voltage might be - due to its commonly flat glitch-peak waveform - in the short circuit region ($V_{thn}<V_b<V_{DD}+V_{thp}$) for a comparatively long time. On the other hand the capacitive current, which is commonly limiting the short circuit current for non colliding input transitions, is quite low, when the glitch-peak voltage at the gate's output is reached. Hence the short circuit charge strongly depends on the input-voltage when the glitch-peak is reached at the gate's output. This later effect was already observed for glitch-generation in Chapter 3.5.3.3.

These basic glitch-propagation characteristics are exemplified by the following simulation results. In Figure 51 the short circuit charge is plotted over the input glitch peak for different output loads C_d. As the load capacitor C_c is constant the glitch waveforms are all equal for the same input peak voltage. The maximum short circuit charge is reached for each value of C_d, if the input voltage is in the most critical short circuit region, i.e. the sum of pull up and pull down impedance has a minimum value, when the output glitch reaches its peak voltage. Hence for high load capacitors C_d the maximum short circuit charge is reached for large input glitches. Note that the most critical short circuit region is always reached when the input glitch's voltage returns to its initial value (resetting input transition).

The high impact of the input voltage when the output reaches its maximum is also visible in Figure52. In this figure the load capacitor C_d is fixed for all curves and the width of the input glitch is varied by different values for C_c. The maximum short circuit charge is not reached for the cases in which the input glitch is in the short circuit region for the longest time but for the cases in which the output glitch is reached while the input glitch is in the most critical part of the short circuit region.

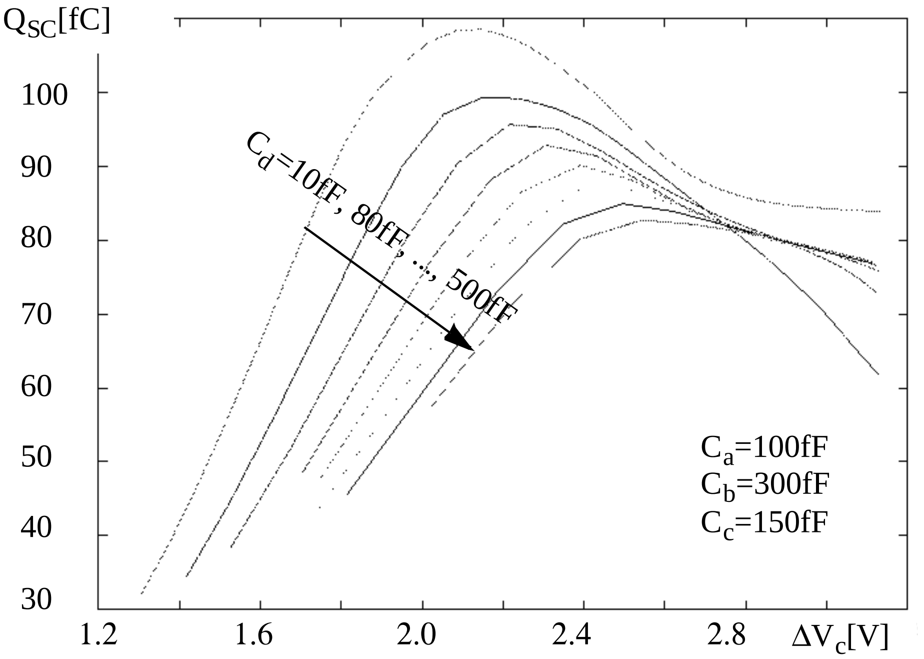


Figure 51:Short-circuit charge over input-glitch peak-voltage for equal input-glitch waveforms and different output loads.

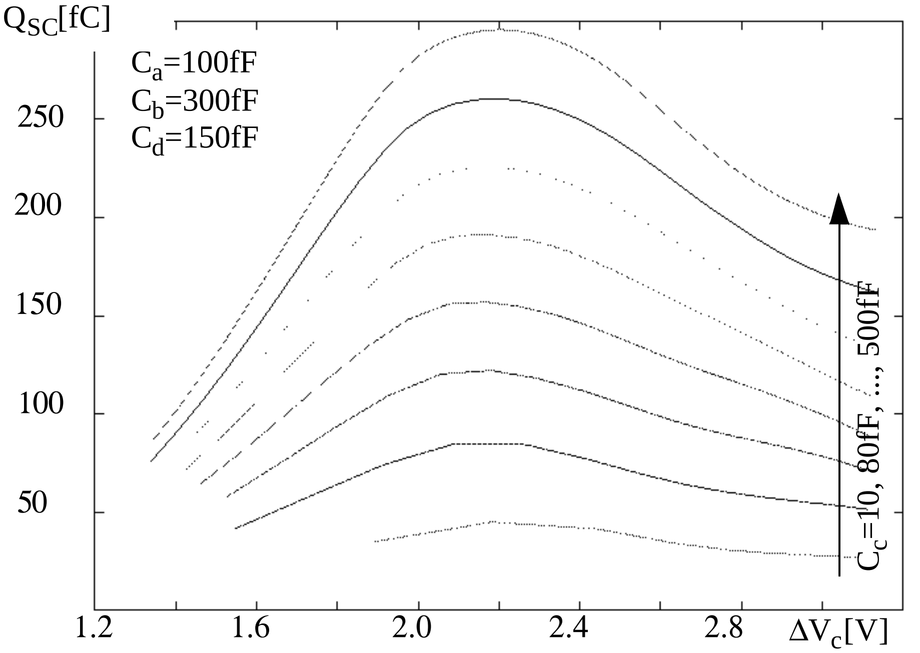


Figure 52:Short circuit charge over input glitch peak voltage for different input glitch waveforms and equal output loads.

In Figure 53 the relative contribution of the short circuit charge to the total consumed charge is plotted again for typical circuit configurations. Depending on the input- respectively output glitch peak the short circuit's contribution is in the range of 60% for small glitches and 15% for large glitches.

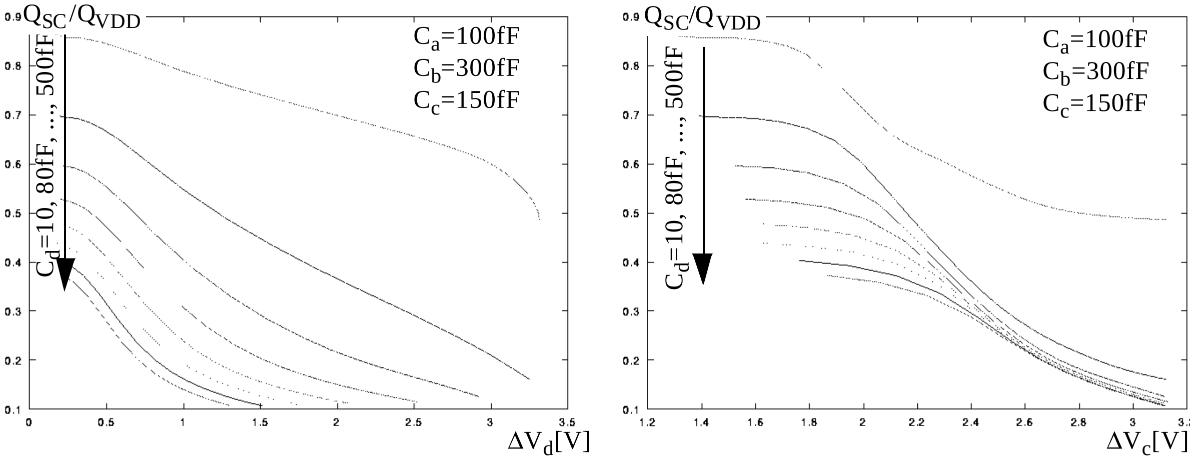


Figure 53:Relative contribution of the short circuit charge to the total charge Q_{VDD}.

3.5.3.5 Conclusions: Relevance of short circuit power consumption

The impact of the short circuit power consumption has been investigated for glitch free and glitch cases. For glitch free cases the short circuit's contribution to the total power consumption of a transition is about 3% and hence may be neglected for *well designed* circuit configurations (i.e. equal input and output rise/fall times). However, for cases with slower input slopes than output slopes its contribution can be up to 90%. For glitches the short circuit's contribution to the total power consumption is between 10% and 60% for typical cases (confer Figure50 and Figure 53) and hence the correct consideration significantly increases the accuracy of glitch power calculation. However, further efforts are needed to derive an appropriate model from these observations, which accurately takes the short circuit charge contribution into account for glitches. On the one hand simulation accuracy could be improved by introducing such a model, on the other hand, this would probably make the power calculation process more complicated and hence slow down the power calculation. Within this thesis such a model is not derived. The short circuit charge contribution is scaled similarly to the capacitive component. This is a possible source for power calculation errors.

4 State of the Art

In Chapter 3 the dynamic power component has been introduced as the major contributor of CMOS ICs’ power consumption. The dynamic power consumption has been split into three contributors:

- Capacitive Power Consumption,
- Short Circuit Power Consumption,
- Signal Conflicts.

The most common contributors are the first two components. Both of these two components depend on the dynamic behaviour of all signals in the circuit (conferEquations 31 and 33):

$$P_{SCi} \sim \alpha_i\text{, i refers to a specific output of a gate}$$

$$P_{Cap\_total} = \frac{1}{2} \times V_{DD}^2 \times f \times \sum_i C_{fanout\_i} \cdot \alpha_i \tag{35}$$

I.e., a major task for power analysis is to calculate α_i for all circuit nodes. Four basic approaches are distinguished here:

- simulation with application specific pattern,
- exhaustive simulation,
- stochastic simulation and
- statistical simulation.

These four approaches will be discussed in Chapter 4.1 in terms of simulation complexity and accuracy. In general synchronous systems are discussed here. Therefore the primary inputs and the internal states may change once per clock cycle. Hence the following different stimulation situations can occur for a state machine (Mealy or Moore):

- m possible current states,
- 2^n (n is the number of primary inputs) possible current input vectors $(s_{n-1}(t), s_{n-2}(t), \ldots, s_0(t))$ and
- 2^n possible consecutive input vectors at t+1 $(s_{n-1}(t+1), s_{n-2}(t+1), \ldots, s_0(t+1))$.

In total $m \cdot 2^{2n}$ different situations are possible. It is a major challenge to deal with the pattern complexities. It also has to be pointed out, that each of the $m \cdot 2^{2n}$ different situations has an application specific probability which needs to be considered. I.e., the node activities - and consequently the power consumption - vary for different circuit applications. Hence, the power consumption of a circuit is not only a function of the circuit but also of its application specific stimulation. A certain power optimized circuit solution is not necessarily the best choice for all applications. In [Schn96] the difference of the switching activity was analysed for different application stimulations. For datapath circuits the total activity[†] varied up to 35% and the signal activities[††] varied up to 150%. The variation of analysed controller circuits was much less (total activity: 5% and signal activity 15%).

[†] The total activity is the sum of all node activities within the analysed circuit. The actual error is calculated for the sum of these activities.

[††] The signal activity describes the activity of a certain node in the circuit. The error of the signal activities is the sum of the individual error’s absolute value.

Besides the simulation technique, the way how to simulate delays also has an impact on simulation accuracy and performance. The following models are discussed in Chapter 4.2:

- Zero-delay model,
- Unit-delay model,
- transport delay model,
- inertial delay model and
- glitch models.

The conclusions of Chapter 4.1 and 4.2 are summarized in Chapter 4.3.

4.1 Gate level power analysis

The major task within gate level power analysis is to determine the activity α_i at all circuit nodes i. The activity is a function of the circuit stimulation pattern.

4.1.1 Simulation with application specific pattern

Explicit application specific stimulation pattern are sometimes available. In these cases the pattern can be simulated by common logic simulators to obtain the circuit node activities. The number of stimulation pattern are typically quite expensive in terms of simulation time. Hence trade-offs are typically needed.

4.1.2 Exhaustive simulation

For an exhaustive stimulation all possible stimulation pattern are applied to a circuit for all possible states. For a state machine with m possible states and n primary inputs $m \cdot 2^{2n}$ different situations can be distinguished. In addition to these logical situations, the skews of the input signals and the clock skew of the state flipflops also have an impact on the dynamic behaviour of the circuit in terms of glitch and hazard power consumption. The number of different situations is far too large for practical cases.

Despite the feasibility problem, the results are only partly usable for the calculation of the average power consumption. Each of the simulated situations has to be weighted according to its application specific probability of occurrence.

4.1.3 Stochastic simulation

Within the simulation techniques, which have been discussed so far, logic values are propagated through the circuits. Instead of these discrete logic values, stochastic simulation techniques use probabilities of signal values and switching probabilities are propagated through the circuit. These probability values represent a large number of possible logic values and logic transitions.

Definition 15: ***Signal probability:***
Let s be a logical signal, to which the logical values {0,1} can be assigned. The signal probability is the probability of this signal to be logically 1 at a specific time:

$$
\begin{aligned}
& p(s(t) = 1) \; = \; p(s(t)) \\
& \text{assuming stationarity:} \\
& p(s(t)) \; = \; p(s) \; = \; \lim_{\tau \to \infty} \frac{\int_{t=0}^{\tau} s(t)dt}{\tau} \\
& p(s = 0) \; = \; p(\bar{s}) \; = \; 1 - p(s)
\end{aligned}
\tag{36}
$$

Definition 16: ***switching probability:***
The probability for a (rising or falling) transition at time t is defined as switching probability. The probability for a temporally uncorrelated signal s at time t can be calculated as follows:

$$
\begin{aligned}
& t^{-} < t < t^{+} \\
& p_{SW}(s(t)) \; = \; \underbrace{p(s(t^{-})) \cdot (1 - p(s(t^{+})))}_{\text{falling transition}} + \underbrace{p(s(t^{+})) \cdot (1 - p(s(t^{-})))}_{\text{rising transition}} \\
& p_{SW}(s(t)) \; = \; p(s(t^{-})) + p(s(t^{+})) - 2 \cdot p(s(t^{-})) \cdot p(s(t^{+})) \\
& \text{assuming stationarity: } p(s(t^{-})) \; = \; p(s(t^{+})) \; = \; p(s(t)) \; = \; p(s) \\
& p_{SW}(s) \; = \; 2 \cdot p(s) \cdot (1 - p(s))
\end{aligned}
\tag{37}
$$

Stimulating the circuit with such probability values and propagating them through the circuit is in the first glance a very attractive alternative to the conventional logic simulation of large numbers of vectors. The major drawback of this probabilistic technique is the fact, that within the simple straight forward approach no signal correlations are considered. If the simple approach is enhanced to take correlations into account the increase of accuracy has to be paid by an increase in computational effort and typically the consideration of correlation is limited. Basically two stochastic approaches can be distinguished:

- Probability waveform [Burc88,Najm89] and
- transition density approach [Najm91].

In the probability waveform approach the signal and switching probabilities are extracted at each node as a function of time. The transition density D(s) is the number of transitions of a signal s per time. This is equivalent to the product of the switching probability $p_{SW}(s)$ and the frequency f. The signal transition densities can be propagated through integrated circuit netlists very efficiently by using the Boolean difference. Both approaches have been enhanced by considering correlations up to a certain extend. A more detailed survey on this topic is given in [Nebe97 - Chapter 4.3].

4.1.4 Statistical simulation

The idea of this approach is to apply input vectors to a circuit until a stopping criterion is fulfilled. The applied pattern either originate from application specific pattern or random pattern generation. Within random pattern generation stochastic properties - including spatio temporal correlations - of the input pattern can be considered [Rade96]. Important works on statistical simulation have been published in [Huiz90,Burc93,VanO93,Saxe97]. A more detailed survey is published in [Nebe97 - Chapter 4.3].

4.2 Simulation of delays

So far simulation techniques have been introduced to deal with application specific stimulation pattern. Within one clock cycle glitches and hazards can occur at internal circuit nodes due to different delay paths from the state flipflops and primary input pins. The used simulation model has a large impact on the calculation accuracy of these power contributors.

Within this thesis it is assumed, that delays are assigned to each instance of a standard cell. Each instance has pin-to-pin and rise/fall delay definitions. These definitions may be instance specific delay values (e.g. from a SDF[†]) or characterization values, which are translated into delay values on the fly.

In the next subchapters different delay models from literature are compared with respect to the glitch and hazard modelling capability respectively the accuracy limitations. Generally the non real delay models (zero and unit delay model) and the real delay models (transport, inertial and enhanced glitch models) can be distinguished. For the real delay models the difference is the applied simulation algorithm for event filtering.

In general a good knowledge of accurate delays is an inevitable assumption for an accurate activity analysis with respect to hazard and glitch contributions. With the decreasing feature sizes more and more attention must be devoted to the interconnects' contributions to the delays (confer Chapter 2.1.1, Table 3). Consequently, in the prelayout phase good floorplanning and wiring estimators are needed. The most accurate data is available after the layout phase from the extraction process.

Within this chapter only transitions between the two logic values 0 and 1 are discussed. Further transitions are possible, which include transitions from or to other logical values (e.g. X or Z).

4.2.1 Zero delay model

For the zero delay model all circuit gates switch immediately without any delay. Under the zero delay model the circuit node changes are calculated only once per clock cycle when the clock signal switches and the new state is calculated. By ordering the circuit in a levelized way the events can be propagated very efficiently through the network. Due to the lack of timing information it is obvious, that no hazard nor glitch power is considered. For this reason the power is underestimated when using a zero delay model.

4.2.2 Unit delay model

The unit delay assumes a unique delay for all gates in the design. Using this delay model hazards at internal nodes are possible. However, the real gate delays vary significantly according to the gate-, capacitive load and input slope characteristics. I.e., the resulting hazard activities from the assumed unit delays for all gates tend not to match with the real hazard activities at a specific node. Glitches and the resulting delay reduction of the resetting slope (confer Chapter 3.2.2) cannot be handled. Due to the missing glitch handling characteristic, the node activity factors α_i tend to be overestimated. But due to the delay inaccuracies activity errors in both directions - over and underestimations - are possible.

[†] SDF: Standard Delay File

4.2.3 Transport delay model

The transport delay model is the most simple of the group of real delay models. In general no glitch or hazard filtering is applied. There is only one exception, which is essential to guarantee the correct logical behaviour within the simulation. Events have to be cancelled (i.e. filtered), if an event is scheduled earlier than events already scheduled for a gate’s output. E.g., suppose an inverter with different rise and fall delays (τ_{LH} = 3 units, τ_{HL} = 6 units) within a common logical simulation (confer Figure 54). For a rising input event at t = 5 a falling output event is scheduled at t = 11. A falling input event at t = 7 would result in a falling output event at t = 10. At that instant the output signal is actually still high and the scheduled falling event at t = 11 needs to be cancelled in order to ensure a correct logical high-value after t = 11. Using a more accurate continuous waveform simulation (displaying v(t)) a glitch or hazard would have been observed at the output Y.

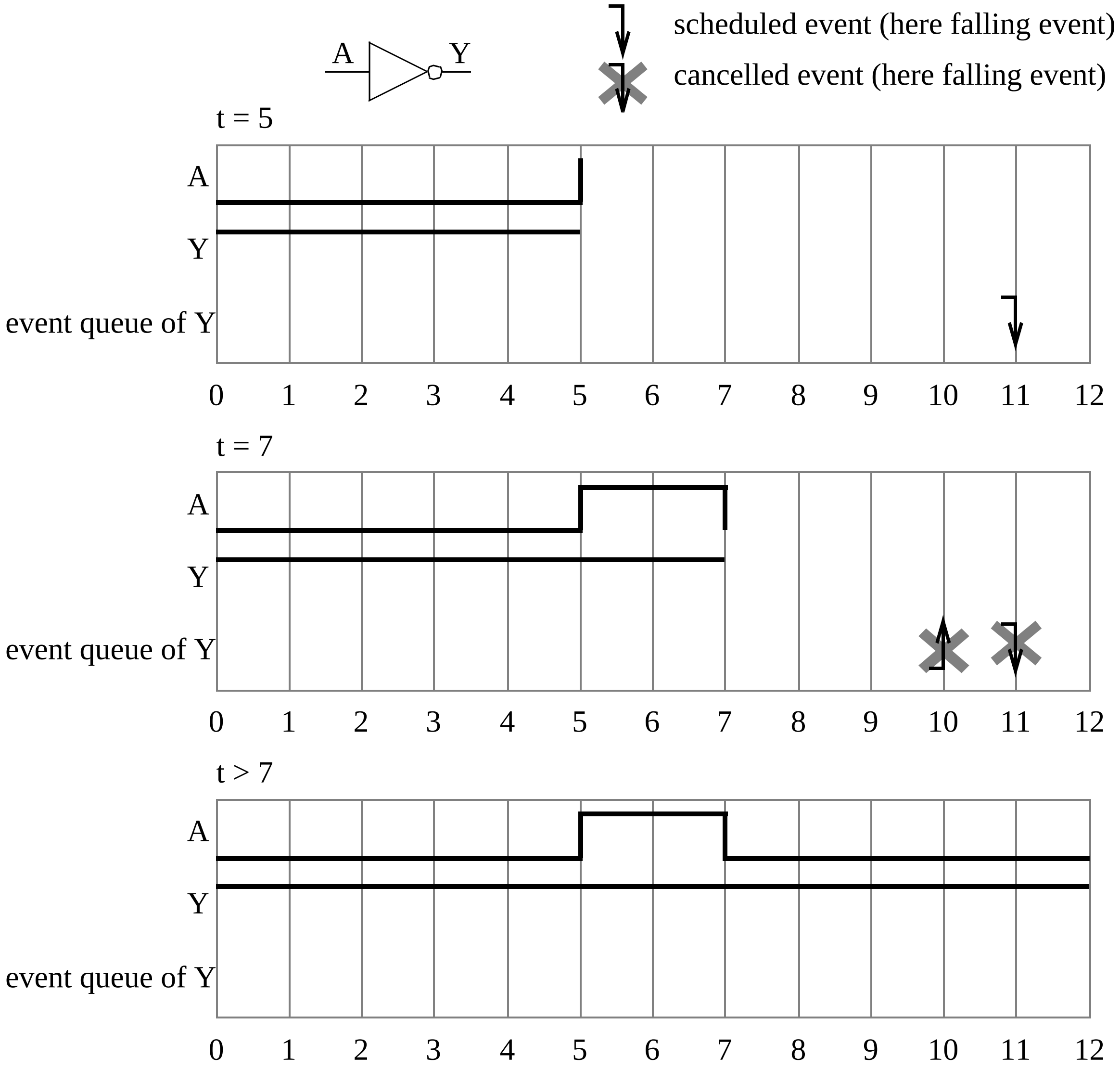


Figure 54:Example for essential event filtering within the transport delay model.

In practical applications the transport model is generally known to generate too high activity values. Glitches are much more likely to occur without being filtered than the above mentioned filtering case. The transport delay model does not consider the dynamic delay reduction of resetting output transitions.

4.2.4 Inertial delay model

The inertial delay is also a member of the real delay models. In addition to the filtering mechanism of the transport delay model (confer Chapter 4.2.3), pulses of shorter duration than the element’s delay are generally not passed through an instance. In practice this means, that a scheduled event is cancelled whenever the gate’s inputs change in such a way that an event would be generated. The cancellation is done as long as the first event is in the event queue (i.e., until the event time is reached within the simulation).

A more general inertial delay model may consider event cancellation up to a certain time after insertion into the event queue. In Verilog-XL [Cade97 - Chapter 12] a percentage of the module path delay can be defined for rejecting events. However, it should be mentioned, that within Verilog-XL this feature is defined as an enhanced transport delay feature for setting the pulse control (+pulse_r/m command line option). Within this thesis, I classify this kind of pulse control feature as an enhanced inertial delay model feature.

Using this kind of simulation model, glitches and hazards are filtered in more cases than for the transport delay model and hence the number of glitches and hazards is reduced. However, even relatively small glitches may be generated and propagated by the inertial delay model. The delay reduction of the resetting output event is not considered correctly so that the pulse width of the modelled glitch is too large. This will result in a pessimistic filtering characteristic in consecutive gates.

This characteristic behaviour is illustrated within an example (confer Figure 55). Consider a rising edge at input A and shortly afterwards a falling edge at input B of a 2 input NAND gate. In circuit level simulation a glitch with a glitch peak voltage ΔV of around 2.5V is generated. In the upper part of Figure 55, the continuous voltage waveforms are given for the two input waveforms. In the middle diagram the simulated glitch output waveform (from HSPICE) and the non colliding setting and resetting output waveforms are shown. Simulating the same case with a common logical simulator using the inertial delay model would propagate two events to the gate’s output. Within the illustrated example (lower diagram of Figure 55) the logic threshold voltages V_L and V_H are defined with 2V (40%V_{DD}) respectively 3V (60%V_{DD}). As the inertial delay model does not consider the delay reduction of the resetting output event, the resetting output event is generated much too late. Hence the generated pulse width is too large, which results in a pessimistic glitch respectively hazard behaviour in consecutive gates. If the resetting input event occurred before the (scheduled) setting output event (difference Δt_{AYset}[†] < 0) the setting output event would have been cancelled and no event would have occurred at the output within the logical simulation.

The generated node activities are lower than for the transport delay model but not necessarily lower than the real physical behaviour. An activity overestimation typically results from the inaccurate consideration of a resetting output event. An activity underestimation typically results from an optimistic (meaning too high) filtering of glitches and hazards. The actual

---

[†] $\Delta t_{AYset} = t_A - t_{Yset}$

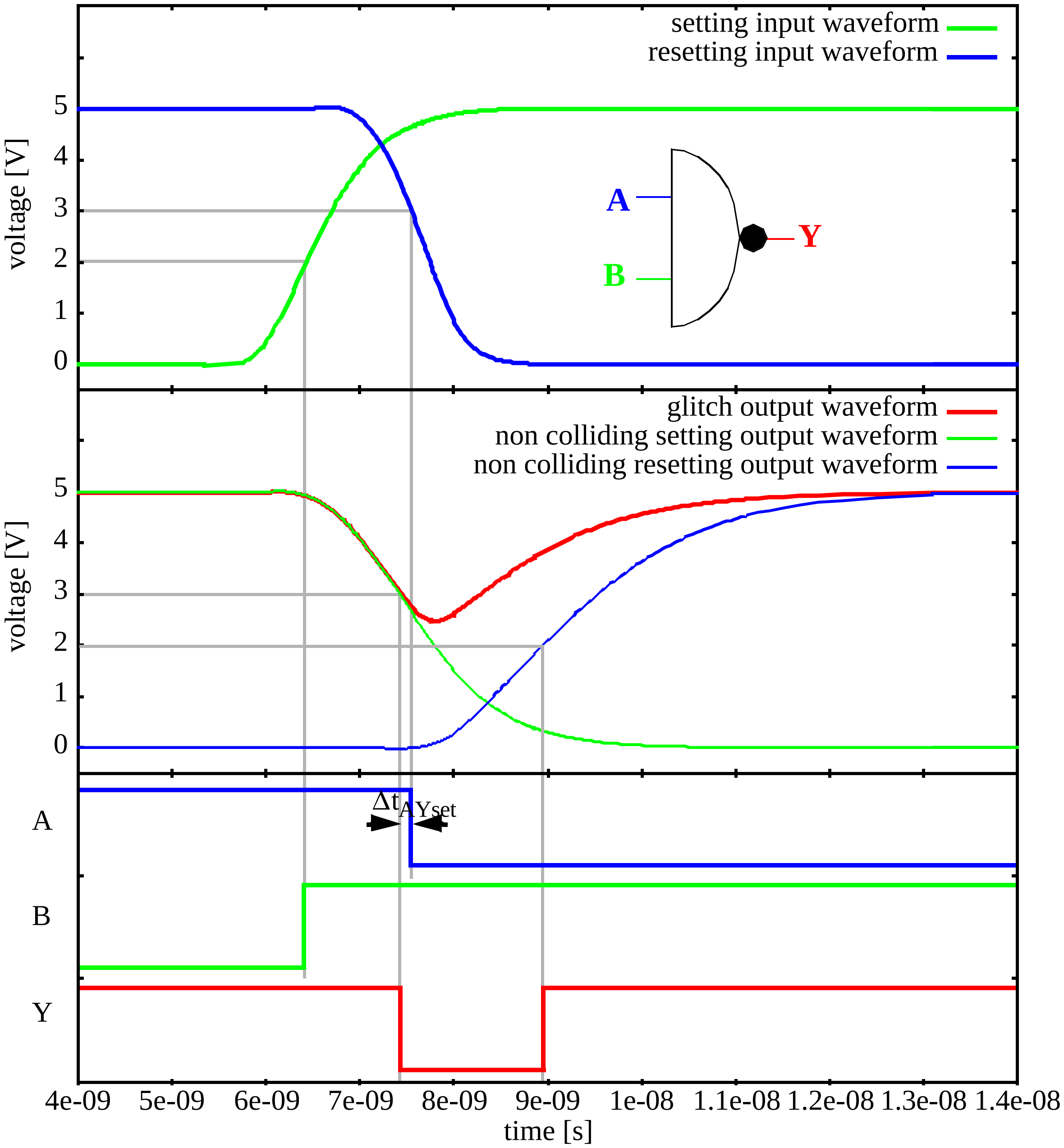


Figure 55:Example for glitch generation under an inertial delay model.

physical structure of a simulated gate is the main reason for underestimation. For single stage gates input transitions may directly influence a setting output ramp at a gate's output. For multistage gates input transitions first have to propagate to the last stage of the gate before having an impact on the output in reality. Within the inertial delay model this internal propagation effect is not considered. This characteristic of the inertial delay model is also observed by the practical simulation results in Chapter 8.1.

4.2.5 Enhanced glitch models

The major contributor to the power consumption of CMOS integrated standard cell circuits is the capacitive dynamic power component (confer Chapter 3.5.2):

$$
\begin{aligned}
P_{Cap\_total} &= \frac{1}{2} \times V_{DD}^2 \times f \times \sum_{i \in \text{ all circuit nodes}} C_{fanout\_i} \cdot \alpha_i \\
\alpha_i &= \frac{1}{f} \cdot \lim_{\tau \rightarrow \infty} \frac{\sum_{j \in \text{ all voltage swings at node i}} \frac{|\Delta V_{i,j}|}{V_{DD}}}{\tau}
\end{aligned}
\tag{38}
$$

The transport and inertial delay model have the following limitations on the accuracy of the activity factor α_i:

a) limited and inaccurate glitch filtering capabilities,

b) no consideration of reduced delays for the resetting output event (pessimistic glitch filtering in consecutive gates) and

c) no consideration of glitch peak voltages.

Already in the past some efforts have been done to develop models to overcome (some) of these problems [Melc91,Metr95,Eise95].

Based on the presented basic CMOS glitching behaviour in Chapter 3.2.4 these three models are presented here. The timing behaviour of glitches is dealt with by all models. In addition the models [Melc91,Metr95] also focus on determining the glitch peak voltage.

Within [Melc91,Metr95] the glitch output waveform is modelled by merging single complete output waveforms, from which a virtual glitch representation is obtained (confer Figure 56).

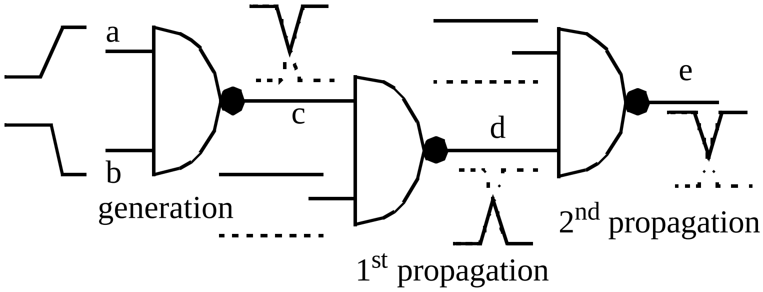


Figure 56:glitch representation by merged single complete output waveforms.

I.e., that the problem of glitch characterization is simplified to characterizing single complete transitions, which only depend on the instance parameters input slope, fanout capacity (and possibly initial internal charges). The skew and (in principal) each input slope's impact on the output waveform are derived from the single (i.e. non colliding) output voltage waveforms.

4.2.5.1 Waveform approximation

The model [Metr95] is based on linearly approximated voltage waveforms (i.e. ramps) in order to ease the model's usage for gate level simulation. The concept of linear approximated waveforms would need to be added to model [Melc91] in a similar way as [Metr95] in order to make it usable for gate level simulation. A linearized single output ramp is derived from its causing input ramp in [Metr95] as illustrated in Figure 57. The meaning of the voltage levels can be seen from Figure 25. The value of τ_{HL} must be characterized for different circuit situations (i.e. input slope and fanout load). The other voltages are characterized from the static

$$V_Y = \frac{V_{jinv} + V_{jILMAX}}{2}$$

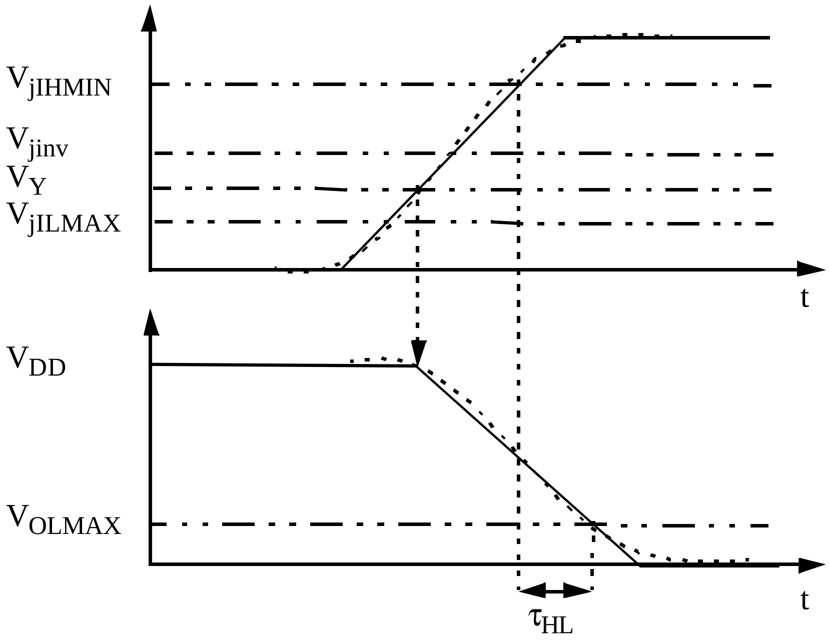


Figure 57:Waveform approximation in [Metr95].

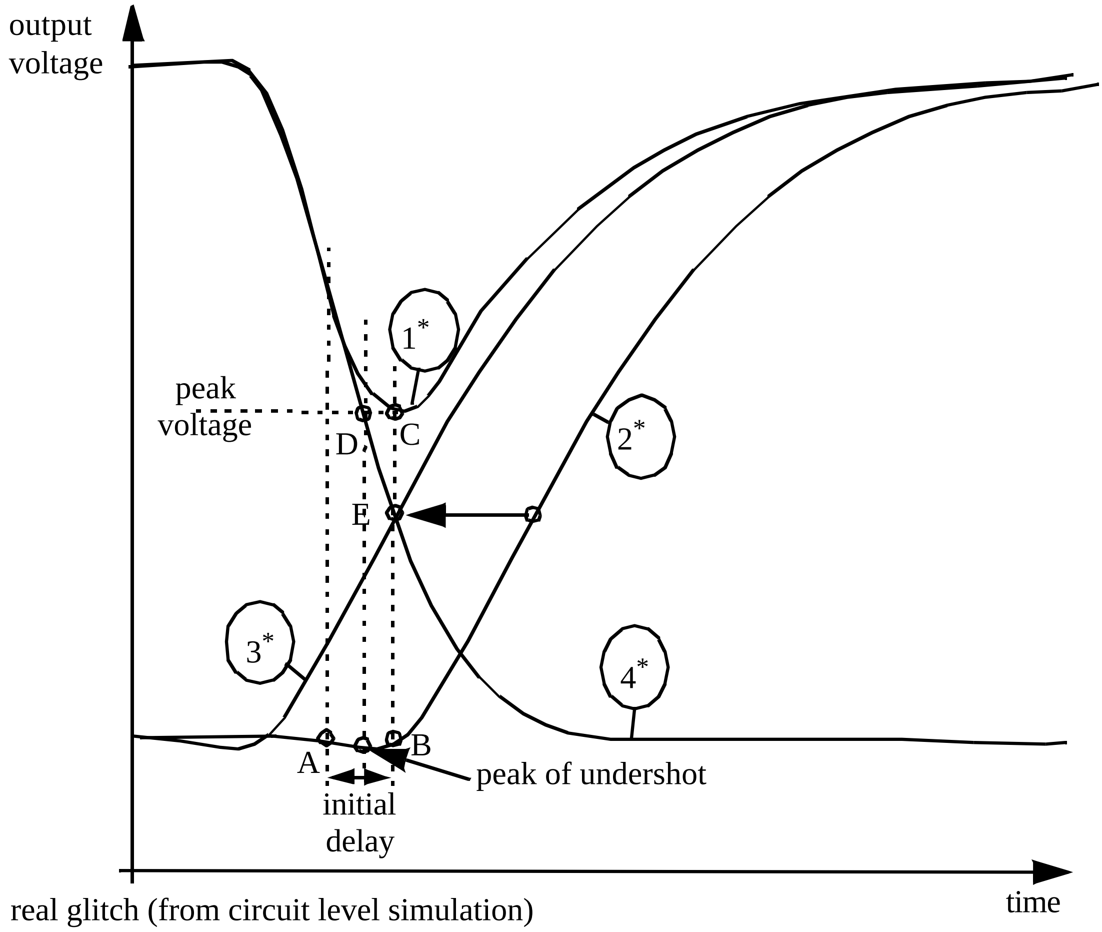
operation curve. For falling input ramps and rising output ramps V_Z (instead of V_Y) and τ_{LH} are used correspondingly.

An important requirement for the linearized waveform is, that it should be a good approximation of the non-linearized waveform within the whole transition. The choice of linearization method is rather a question of which characterizations are already available for target libraries. Typically it is hard to persuade a library vendor to characterize the same subject twice. It should be emphasized in this context, that for [Metr95] choosing higher (lower) reference voltages for τ_{HL} characterisation instead of V_{OLMAX} (V_{OHMIN}) has a significant impact on how well the non-linearized waveform is approximated by the linearization.

4.2.5.2 Glitch peak voltage modelling

In [Melc91] the glitch peak voltage is modelled by the voltage of the single setting output waveform when the overshot of the (non colliding) resetting output waveform reaches its peak (point D) (confer Figure 58).

In Chapter 3.2.4 it has been observed, that the glitch equilibrium state is mainly a function of the resetting input voltage. In [Melc91] the peak of the resetting output waveform's overshot is taken. These two modelling alternatives are not right away contradictory. Output voltages at the beginning of the output waveform are approximately independent of the capacitive load and the input slope. I.e. the output voltage values in this region belong to approximately fixed voltage values of the causing input waveform (this is also the basic for the linear waveform approximation in [Metr95]). In Chapter 3.2.4 it has been further observed, that the equilibrium state of most glitches (absolute peak voltages in the range [V_{OLMAX},V_{OHMIN}]) is reached for $V_{inReset}$ = [V_{ILMAX},V_{IHMIN}]. Hence the model [Melc91] assumes, that for the time, when the peak of the non colliding resetting waveform's overshot is reached, the resetting input waveform's voltage has to be in the region [V_{ILMAX},V_{IHMIN}]. To verify this assumption, I analysed the relation between the input voltage of the causing input waveform at the instant of the out-



1* real glitch (from circuit level simulation)
2* non-colliding resetting output ramp (i.e. V_{in}(a) = V_{DD})
3* dynamically scheduled resetting output ramp 2* (2* has been shifted)
4* non-colliding setting output ramp (i.e. V_{in}(b) = V_{DD})

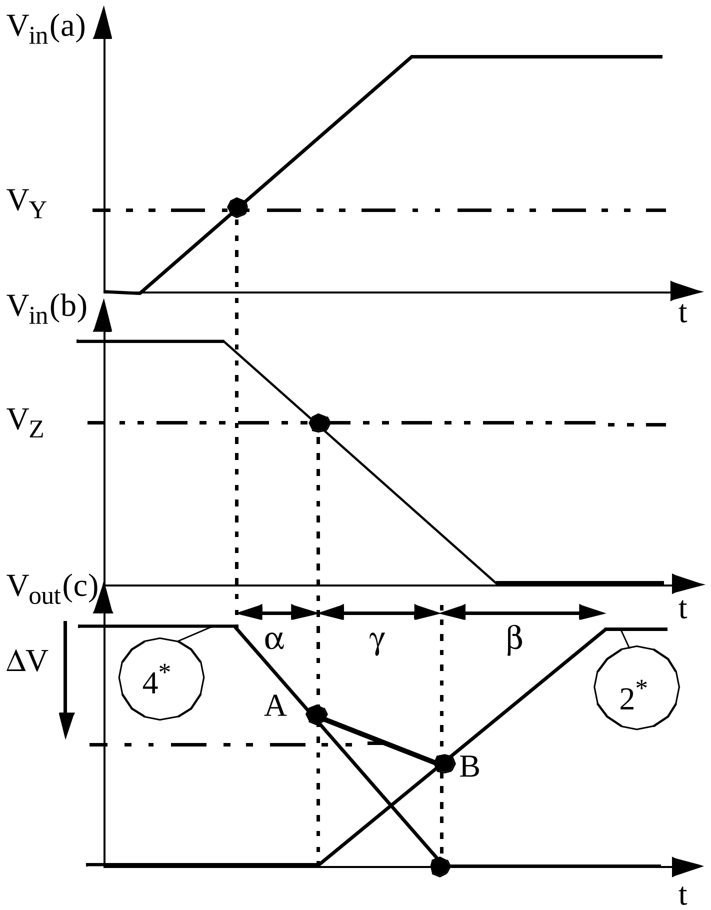
Figure 58:Glitch handling model of [Melc91].

put's undershot by some circuit level simulations (HSPICE). The resulting voltage was for rising (falling) input waveforms in most cases significantly lower (higher) than expected. I.e., the instant when reading the peak voltage (confer point D in Figure 58) is typically much too early and consequently the peak voltage is underestimated by [Melc91] for most cases.

In [Metr95] three regions are identified (confer Figure 59):

- region α: the resetting input ramp's voltage is above (below) V_Z (V_Y[†]) and the setting input ramp's voltage is above (below) V_Y (V_Z[†]) and therefore the output glitch is clearly dominated by the setting output ramp.
- region γ: both input ramps have a significant impact on the glitch while the resetting input ramp's voltage is between VSS and V_Z (V_Y and VDD[†]) and the single setting output ramp has not reached VSS (VDD[†]).
- region β: the single setting output ramp has reached its final state VSS (VDD[†]) and therefore the glitch is dominated by the resetting output ramp.

[†] The voltages in the bracket are for a falling setting and a rising resetting input ramp. In the discussed example (confer Figure 59) the setting input ramp is rising and the resetting input ramp is falling.



2^* non-colliding resetting output ramp (i.e. $V_{in}(a) = V_{DD}$)
4^* non-colliding setting output ramp (i.e. $V_{in}(b) = V_{DD}$)

Figure 59:Glitch handling model of [Metr95].

The actual peak voltage is approximated by (V(A) + V(B)) / 2. Point A is very closely related to the basic CMOS behaviour which was introduced in Chapter 3.2.4. However, the resetting output ramp (confer 2^* in Figure 59) needs to be dynamically scheduled, if the glitch is supposed to be virtually described by it, which is not taken into account in [Metr95]. I.e. the smaller the glitch, the less this resetting output ramp virtually represents the glitch. Point B is on the (non-dynamically scheduled) single resetting output ramp and hence its relevance for the determination of the glitch peak voltage is not obvious.

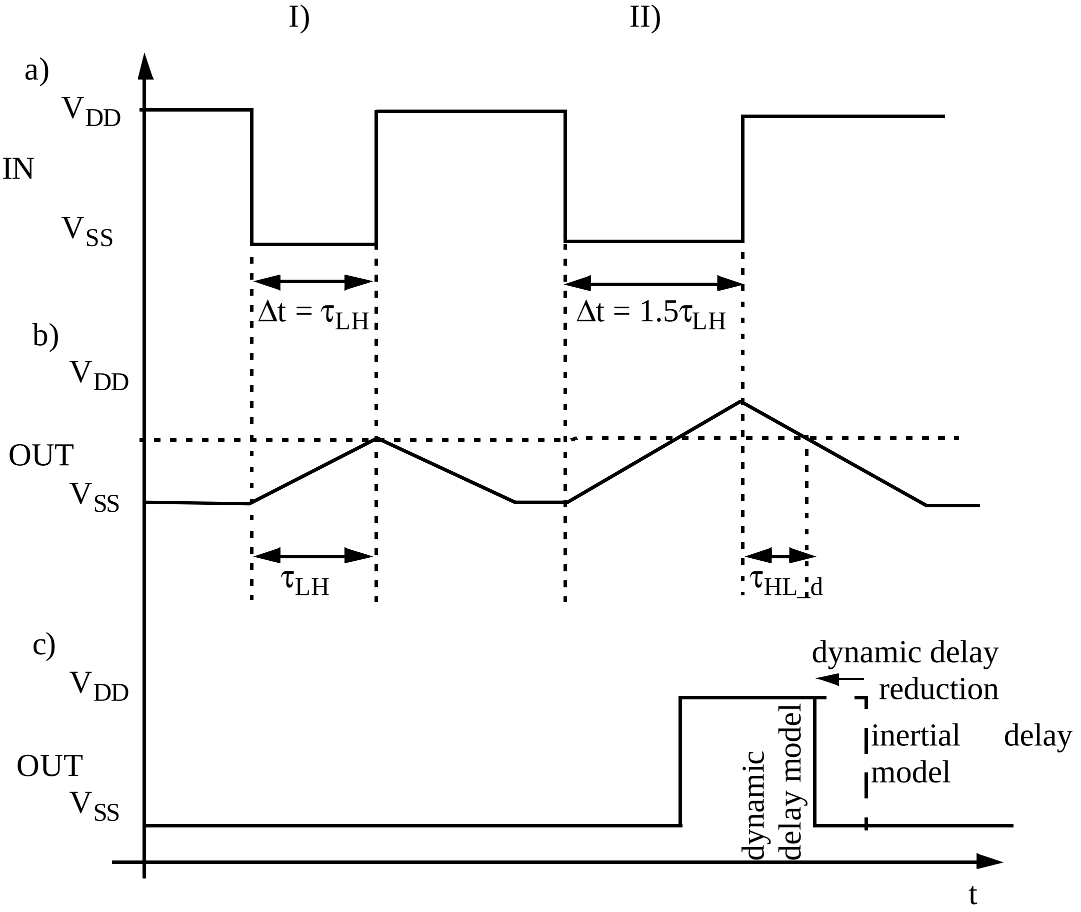
In [Eise95] the glitch peak voltage is not determined. Even though this model can be extended to also handle glitch waveforms as linear ramp approximations. But as the original model does not handle glitch peak voltages, the model and the glitch peak voltage extension are presented in the next subchapter.

4.2.5.3 Glitch representation for possible propagation

A generated glitch is represented by two virtual ramps (setting and resetting), which are used in consecutive gates for propagation. The initial voltage of the resetting part of the glitch waveform is its peak voltage and hence it is not equal to the initial voltage of the single resetting output ramp (either VDD or VSS). For this reason a dynamic scheduling mechanism for the resetting part is needed [Melc91,Eise95]. In [Metr95] the glitch is represented by its virtual single setting and non-dynamically scheduled resetting ramps, which leads to a loss of precision.

In [Melc91] the glitch is virtually represented by the unchanged single setting ramp and by the time shifted single resetting ramp (confer Figure 58). The time shift is defined in such a way, that the setting and resetting ramp cross each other at glitch peak time, which is defined as the end of the resetting output ramp's overshot (confer point B of Figure 58).

In [Eise95] a delay model is presented which dynamically calculates delays for input pulses (i.e. glitches) whose width is between the gate's propagation delay and twice the propagation delay. The model is illustrated in Figure 60. Part a) of Figure 60 shows the logic input transitions, part b) the resulting linearly approximated output waveform and part c) the logic output transitions using the inertial (intermittent line) and the dynamic delay model (solid line). For the second input pulse the dynamic delay model considers that V_{DD} is not reached and hence the second event occurs earlier than for the inertial delay model. This way of modelling focuses on glitch propagation. The model [Eise95] takes into account, that glitches might disappear during propagation and hence it is better than the inertial delay model. Even though the authors didn't explicitly focus on waveform modelling the basic idea is very similar to [Melc91] with the following interpretation (confer Figure 60):



τ_{HL_d}: dynamically reduced propagation delay

Figure 60:Glitch handling model of [Eise95].

- each single output ramp starts, when the input ramp crosses the logic threshold voltage (here: $V_H = V_L = 50\%V_{DD}$, confer Figure 61),

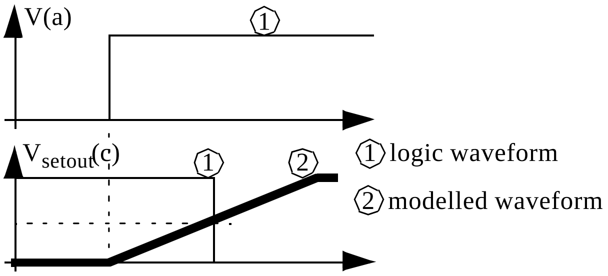


Figure 61:Ramp construction to make peak voltage estimation possible for [Eise95].

- the output ramp reaches the voltage level V_H respectively V_L according to the propagation delay $\Delta t = t_{LH}$ respectively $\Delta t = t_{LH}$ (confer Figure 61),
- the glitch peak time is defined by the instant when the single resetting output ramp starts, i.e. when the resetting input ramp switches,
- glitches with peak voltages less than V_H respectively V_L are absorbed; even for these glitches a peak voltage might be calculated for power calculation, but no glitch will be propagated.

A very important feature of this model [Eise95] is that besides the common delay characterization no additional characterization is needed.

4.2.5.4 Comparison of different glitch models

All three models [Melc91,Metr95,Eise95] are compared with respect to circuit level simulation by using a small benchmark circuit (confer Figure 62 - the driving inverters for signal a and b are not shown). The following parameters were varied:

- skew: in steps of 60ps
- two different slopes at both inputs a and b
- four different loads at c, d and e

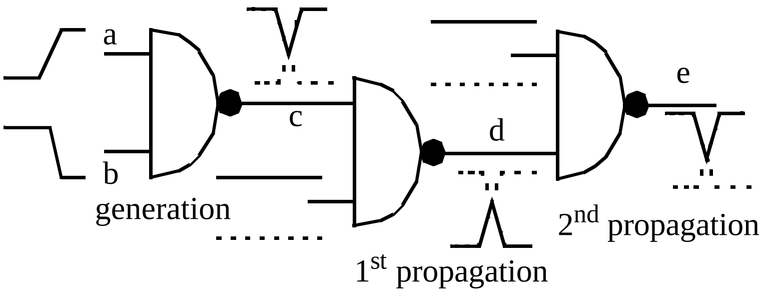


Figure 62: Benchmark circuit for evaluation of glitch model.

For glitch generation analysis (at node c) six further slopes for both inputs a and b and four further loads at node c were investigated.

Only cases which produce glitches for at least one of the models at the respective level (c, d and e) were considered (except the glitch peak voltage errors in Figure 63). In total approximately 17800 different cases were examined. The delays and the slopes were directly determined by circuit level simulation for each case (i.e. the focus is on glitch modelling and not on delay modelling of non-glitching transitions). Characteristic glitch parameters, which are needed by the models were determined before.

From the simulation results the glitch peak voltage and peak time for each simulation run were extracted. The difference between the circuit level simulation and each model $(\text{value}_{\text{circuit-level}} - \text{value}_{\text{model}})$ were investigated statistically for signals c, d and e (i.e. the mean value and the standard deviation). For model [Eise95] two different logic threshold voltages were investigated:

logic threshold voltage for	Model [Eise95]	Model [Eise95] modified
falling ramps V_H	50% V_{DD}	60% V_{DD}
rising ramps V_L	50% V_{DD}	40% V_{DD}

Some important details about the statistical processing of the data are given next:

- in the circuit level simulation glitches were considered in the glitch peak voltage range $4\%V_{DD} < \Delta V < 96\%V_{DD}$ ($V_{DD} = 3.3V$),
- for the glitch peak time statistics only cases can be considered, where the respective model and the circuit level reference case produced a glitch (i.e., for models with large errors the number of unusable simulation results for the glitch peak time is considerable) and
- the results for the glitch peak voltage were based on the cases where at least one of the models or the circuit level simulation result in a glitch (i.e., if for a model a hazard- or no transition is correctly detected, this case is taken into the statistics, if at least one of the other models predicts a glitch; consequently the data can be used only for relative comparisons).

The results are show in Figure 63, 64 and 65. The results show the above mentioned model characteristics (confer Chapter 4.2.5.2 and 4.2.5.3):

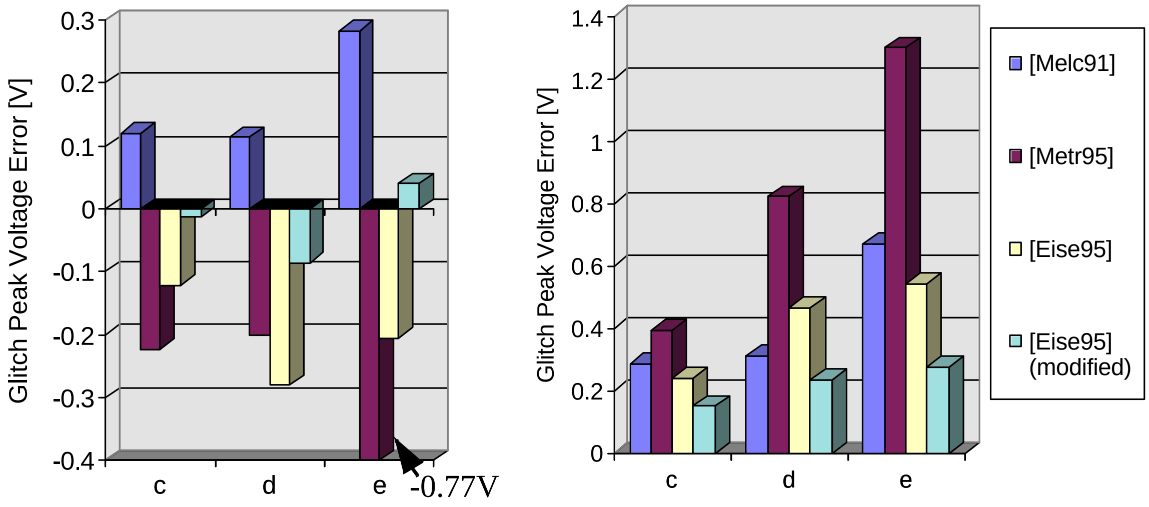


Figure 63:Glitch peak voltage error: mean value (left), standard deviation (right).

- for model [Melc91] the glitch peak time is estimated too early and the glitch peak voltage too low, these errors increase slightly during propagation together with the standard deviation,
- for model [Metr95] the glitch peak time is estimated too late and the glitch peak voltage too large, the errors increase significantly during propagation.

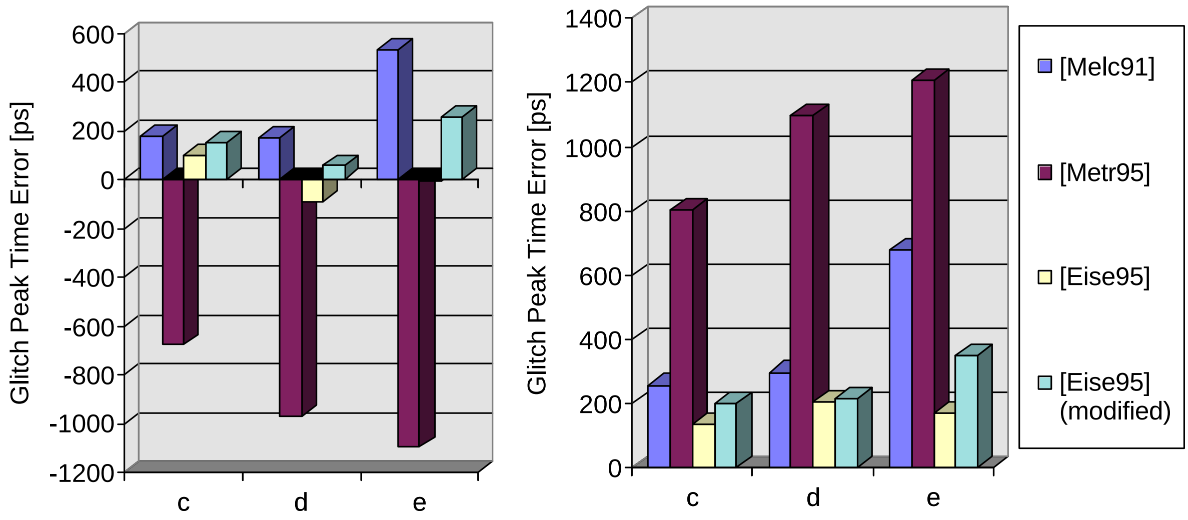


Figure 64:Glitch peak time error: mean value (left), standard deviation (right)

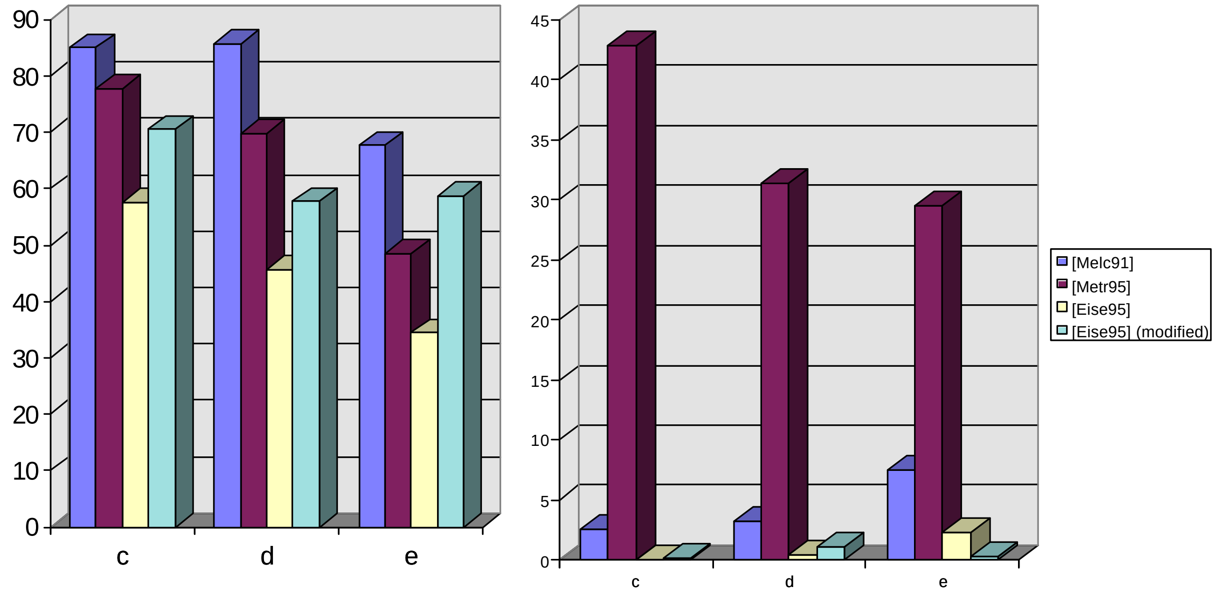


Figure 65:Relative amount of simulated glitches on circuit level which are detected by the gate level models (left), relative amount of detected glitches by the gate level models which are no glitches on circuit level (right).

In the left part of Figure 65 the relative amount of simulated glitches on circuit level which are detected by the gate level models[†] are illustrated:

- The number of detected glitches is quite low for model [Eise95], because small glitches are not considered within the model. The usage of the modified logic threshold voltages improves the model.
- The number of detected glitches by [Metr95] decreases for the propagated glitches (many glitches of the circuit level simulation occur as hazards within the model).

[†] A glitch is detected by a model if the glitch peak voltage is in the range [4%V_{DD} ,96%V_{DD}] **and** the virtual ramps cross each other

- The number of detected glitches by [Melc91] model is quite high. One important reason for this high accuracy is the usage of the single non-colliding continuous output waveform for extracting the projection instant of the potential glitch peak voltage and peak time. A ramp approximation, as it is considered for the model [Eise95], is especially inaccurate at the beginning and at the end of a single complete transition. The ramp approximated waveform starts later than the continuous waveform and reaches its final voltage earlier. Consequently small and large glitches tend not to be recognized as glitches by a linear approximated model. To efficiently use the model [Melc91] within a logic level simulator would require some sort of waveform simplification. In addition the actual modelling of the output waveform's undershot (respectively overshot) also affords further investigations. In conclusion, the accuracy results, which are illustrated in the above figures are not directly comparable to the other models.

The right part of Figure 65 shows the relative amount of detected glitches by the gate level model which are no glitches on circuit level:

- Model [Metr95] detects a lot of glitches, which actually are filtered in circuit level simulation due to the missing dynamic scheduling of the resetting ramp.
- Only a moderate number of additional glitches are obtained by model [Melc91].
- The number of additional glitches detected by model [Eise95] is quite low, but the number of found glitches is also quite low.

The characteristic features and limitations of the three models, which have been exemplified in the previous subchapters, are summarized in Table 10.

Model	Features	Obvious accuracy limitations
[Melc91]	• Glitch peak voltage calculation is considered, • dynamic delay reduction of resetting output slope is taken into account, • the actual mapping of the model on an efficient gate level model remains an open issue.	• The projection on the setting output waveform for glitch peak voltage extraction is done too early, which results in too low peak voltages ΔV.
[Metr95]	• Glitch peak voltage calculation is considered, • dynamic delay reduction of the resetting output slope is **not** taken into account.	• The missing dynamic delay reduction results in low accuracy of propagated glitches • the usage of the non colliding setting output ramp for the determination of the glitch peak voltage is a source of error
[Eise95]	• Glitch peak voltage calculation is **not** considered (but model can be extended to deal with peak voltage calculation), • dynamic delay reduction of resetting output slope is taken into account.	• Glitches below the logic threshold voltage are not taken into account, • the logic threshold voltage has a large impact on the model's accuracy.

Table 10:Overview of the glitch model's features and accuracy limitations [Melc91,Metr95,Eise95].

The experiments within this chapter only cover single stage gates. The characteristic glitching behaviour, which has been derived in Chapter 3.2.4, is based on single stages. For multi stage gates (e.g. non inverting CMOS gate like AND-, OR-gates) the most accurate way is to tread each stage separately. The relation between the resetting input slope and the equilibrium state at the output is obviously not transferable to multi stage gates:

a) On the one hand a glitch may occur at the output of an internal stage, which is hard to consider by a model which does not split the gate into stages.

b) On the other hand for glitches, which are generated at the last stage of the gate, the resetting input transition of the last stage's gate is delayed from the gate's primary input.

The above compared models [Melc91,Eise95,Metr95] are not capable to consider glitches at internal nodes (Point a).

In addition model [Eise95] directly uses the resetting input transition within the glitch model (disadvantage for Point b). In model [Melc91] it is generally not dealt with how to predict the overshot's waveform at the gate output. In model [Metr95] the resetting input waveform is used indirectly for the glitch model, because the start of the linear approximated output ramp is derived from a fixed input voltage V_Y respectively V_Z (confer Chapter 4.2.5.2). This assumption is fairly accurate for a single stage. For a multistage gate the main source of error is the variation of (primary) input slopes, which results in a variation of the internal input slopes. This assumption shall be exemplified using the example in Figure 66. A non colliding input waveform is propagated through an AND2-gate, which is typically build up of a NAND2- and an INVERTER-stage. The internal load capacitor at node d has a fixed value. A rising input ramp at input a results in a falling input ramp at the internal node d. The internal ramp starts, when the input ramp at input a crosses the voltage V_{Y_Stage1}. Finally the output ramp at output c starts when the ramp at node d crosses V_{Z_Stage2}. The question is, how accurate the start of the ramp at output c can be modelled by the value V_{Y_gate}. The main source of error is the variation of Δt with the assumption, that the start of a single stage's output ramp is modelled accurate enough by the model. As the capacitance at node d is fixed for the AND gate, the variation of Δt may only be caused by the input slope at a. Hence the main question is, how much the input slope of a stage can influence the output slope of the stage. A deeper numeric analysis of this question is omitted here.

4.3 Conclusions

In Chapter 4.1 different approaches to handle the pattern complexity have been introduced. The probabilistic approach is a good choice for propagating a large amount of logical pattern within one step through the whole circuit and obtain the desired node activities. However, this approach has only limited capabilities to consider spatio temporal correlations, which are automatically taken into account within logical simulation based approaches. Especially for utilizing the delay reduction of resetting output events it has to be known whether the setting and the resetting event originate from the same logical pattern or not. In other words, the exact consideration of the temporal correlation between a possible setting and resetting event are mandatory for an accurate glitch respectively hazard analysis. As this accuracy is hard to achieve for practical applications, the logical simulation based approaches are better suited for the targeted accurate activity analysis.

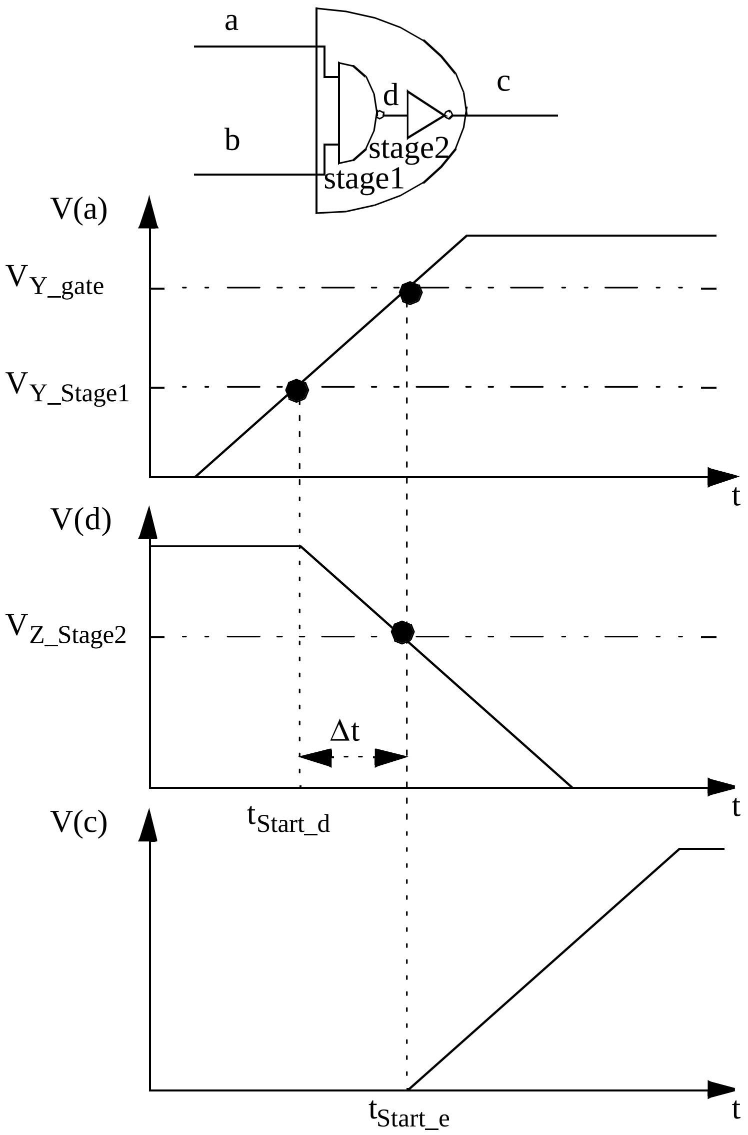


Figure 66:Modelling of a multi stage gate’s output ramp in [Metr95].

In Chapter 4.2 different delay and simulation models have been introduced. The simple zero and unit delay models are no candidates for an activity analysis, which accurately takes unnecessary transitions into account. The traditional real delay simulation models are the transport and the inertial delay models. Neither of them is capable to consider the dynamic delay reduction or the determination of a glitch peak voltage in case of a glitch. Some enhanced simulation models have been invented so far. None of these models [Melc91,Metr95,Eise95] gives an accurate solution to both of these limitations. The goal of the new approach, which is the topic of Chapter 5, is to overcome these problems and enable an efficient implementation within a simulator.

5 The new Glitch-Model

From the basic glitch properties, which have been introduced in Chapter 3, the new proposed model is derived. The model is discussed in terms of accuracy for the glitch peak voltage and the glitch peak time. In Chapter 7 it is shown, that this model can easily be implemented into an event driven logic simulator, which gives good results for complete benchmark circuits in terms of accuracy and simulation performance (confer Chapter 8).

5.1 Derivation

The basic idea of the proposed model is to represent a glitch by two or more linearly approximated ramps similar to [Melc91,Metr95] (confer Figure 67). The ramps can be easily derived from delay and slope information. A pair of a (colliding) setting and resetting ramp always represents a glitch or part of it (in case of more complex glitches). If a glitch is detected, the resetting ramp is scheduled into the event queue considering the dynamic delay reduction.

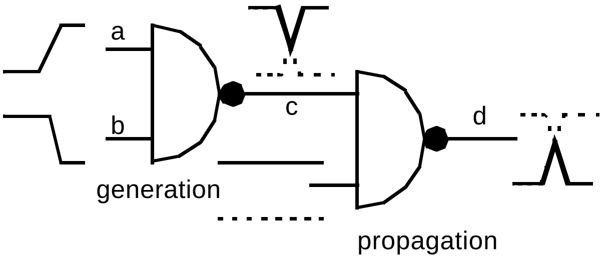


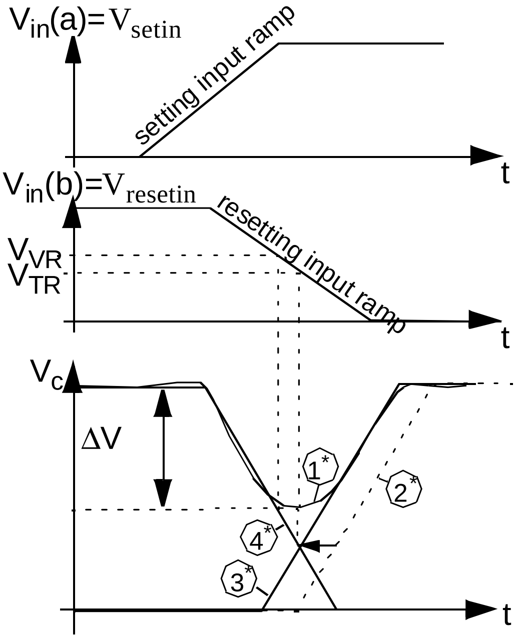
Figure 67:Representation of glitches by linear approximated ramps.

The remaining question is how to schedule a resetting ramp into the simulator’s event queue and how to predict the glitch peak voltage. As the gate is in the equilibrium state (confer Chapter 3.2.4) when the glitch peak is reached, the gate’s dynamic operation point ($V_{resetin}(t_{glitch})$, $V_{setin}(t_{glitch})$, $V_{1*}(t_{glitch})$[†]) (confer Figure 68) is approximately equal to the respective static operation point. The static characteristics neither depend on a gate’s input slope nor on its fanout load. In Chapter 3.2.4 it has been further observed that the impact of $V_{setin}(t_{glitch})$ on the equilibrium state can be neglected. I.e. $V_{setin}(t_{glitch})$ can be approximated with V_{DD} (V_{SS}) for rising (falling) setting transitions.

Within the new model four characteristic voltage values are introduced for each input-to-output-pin combination (confer Figure 68):

- V_{TR}: Voltage of the falling resetting input slope (index *R* refers to the resulting *r*ising output slope) at the instant when the glitch peak is reached at the stage’s output .
- V_{VR}: Voltage of the falling resetting input slope at the instant when the glitch peak voltage of the non-colliding setting output ramp is reached.
- V_{TF}: same as V_{TR} except that the resetting input slope is rising (i.e. the resulting output slope is *f*alling).
- V_{VF}: same as V_{VR} except that the resetting input slope is rising.

[†] $V_{1*}(t)$ refers to the (real) continuous glitch waveform at a gate’s output, which is obtained from circuit level simulation (confer Figure 68).



1^* real glitch (from circuit-level simulation)

2^* non-colliding resetting output ramp (i.e. $V_{in}(a) = V_{DD}$)

3^* dynamically scheduled resetting output ramp 2^*

4^* non-colliding setting output ramp (i.e. $V_{in}(b) = V_{DD}$) - also refered to as V_{setout}

Figure 68:Glitch model and its characteristic voltages for a NAND2-gate (cf. Fig. 67).

Each cell needs to be characterized with respect to these voltages.

The V_{TF}- and V_{TR}-values are used for scheduling the resetting ramp at the gate's output. It is scheduled in such a way that it crosses the setting output ramp when the resetting input ramp reaches V_{TF} respectively V_{TR} and the *real glitch* (1^*) its maximum ΔV. The glitch is represented by the two ramps for possible glitch propagation. The instant when an input ramp crosses the respective V_T-value (V_{TF} or V_{TR}) is called the projection time for a possible glitch peak time t_{pt}.

The V_{VF}- and V_{VR}-values are used to predict the glitch peak voltage which is needed to calculate the corresponding glitch power consumption (confer Equations 29 and 30). The instant when an arbitrary input transition crosses the respective V_V-value (V_{VF} or V_{VR}) is called the projection time for a possible glitch peak voltage t_{pv}.

The effect, that the setting (non-colliding) output ramp (4^*) and the real glitch (1^*) diverge the more the resetting input ramp takes control of the glitch, is modelled by taking different values for V_V (V_{VF} respectively V_{VR}) and V_T (V_{TF} respectively V_{TR}).

Due to the diverging waveforms of the real glitch and the non-colliding setting output ramp
- neither ($V_{resetin}(t_{pv})$, $V_{setin}(t_{pv})$, $V_{setout}(t_{pv})$)
- nor ($V_{resetin}(t_{pt})$, $V_{setin}(t_{pt})$, $V_{setout}(t_{pt})$)

exactly are the same as the triple of the real glitch reaching the equilibrium state. Hence the parameters do have a small dependency on the gate's output load and its input slope. This dependency has been analysed by means of circuit-level simulation for various single stage gates.

As a typical example a NAND2-gate of an industrial 0.5µm-CMOS library (V_{DD} = 3.3V) is discussed here. It was analysed within the testbench shown in Figure 69. An inverter is used to get realistic input slopes at the inputs A and B of the GuT (gate under test). The input slopes of a and b are modified by additional loads of the inverters. The capacitor between c and V_{SS} represents the GuT's load. For various combinations of capacities glitches with different peak

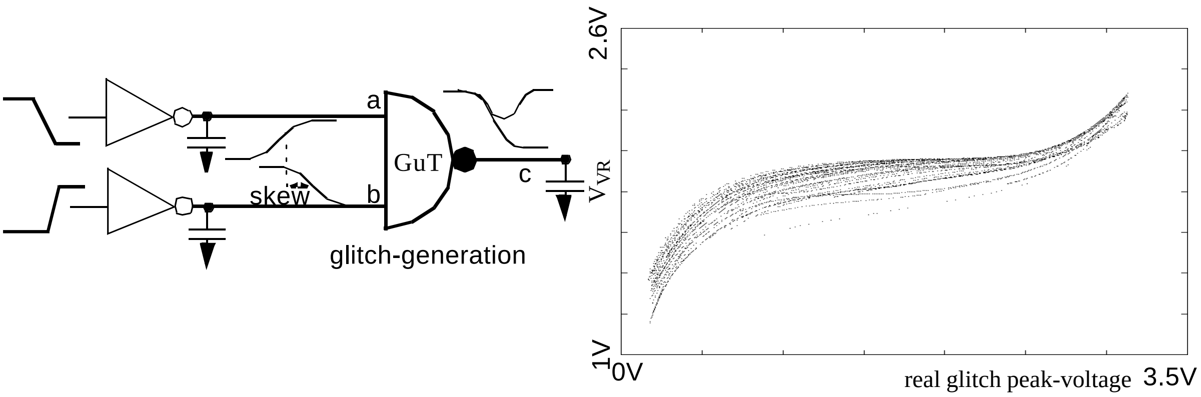


Figure 69:Investigation of characteristic voltage values (left), V_{VR} of a NAND2-gate for different circuit configurations (right).

voltages were generated by varying the input skew. The simulation results are shown in Figure 69. The ordinate axis contains the values for V_{VR} and the coordinate axis the glitch peak voltage. The different curves correspond to a variety of different capacitor configurations. Small glitches result in smaller V_{VR}-values than bigger ones. This characteristic behaviour has been explained in Chapter 3.2.4 (static CMOS behaviour). The impact of the setting input ramp (at node a) is very low as it has reached a voltage level close to V_{DD} for most cases when the glitch reaches its peak. The few curves which are not within the *curve bundle* belong to very small loads at node c and slow setting input ramps of the GuT. For these cases the voltage of the setting input ramp is comparatively small when the glitch reaches its peak. However, the affected glitches result in very little power consumption and are no candidates for glitch propagation as the load is smaller than the smallest fanin capacity of a gate within the library.

For the model two approximation alternatives of the characteristic values (V_{VF}, V_{TF}, ...) are discussed:

- using constant values (confer Figure 70),
- using a piecewise linear approximation (confer Figure 71).

By using a constant value for V_{VR} the projection on the setting output slope is done too early for small glitches respectively too late for large glitches. I.e., for small glitches the peak voltages are underestimated and for large glitches overestimated.

This can be exemplified using Figure 70. Consider a falling resetting input slope. For small glitches (left part of the diagram) the correct V_{VR} value is smaller than the actually used one. I.e., the input projection on the setting output ramp (resetting input ramp is falling) is done too early and hence the peak voltage will be too small. For large glitches (right part of the diagram) the correct V_{VR} value is larger than the actually used one. I.e., in this case the projection is done too late, which results in too large glitch peak voltages.

The curve bundle for V_{TR} have a similar shape. The corresponding values are approximately 0.4V lower than the V_{VR} values because the glitch peak is reached later than the projection instant for the glitch peak voltage (confer Figure 68).

The overestimation of large glitches and the underestimation of small glitches can be avoided if the waveform of the V_{VR} values is used by the model. A practical simplification is to model the waveform by a piecewise linear approximation (confer Figure 71). The needed part of the piecewise linear waveform (PWL) can be determined according to Equation 39 (also refer to Figure 72):

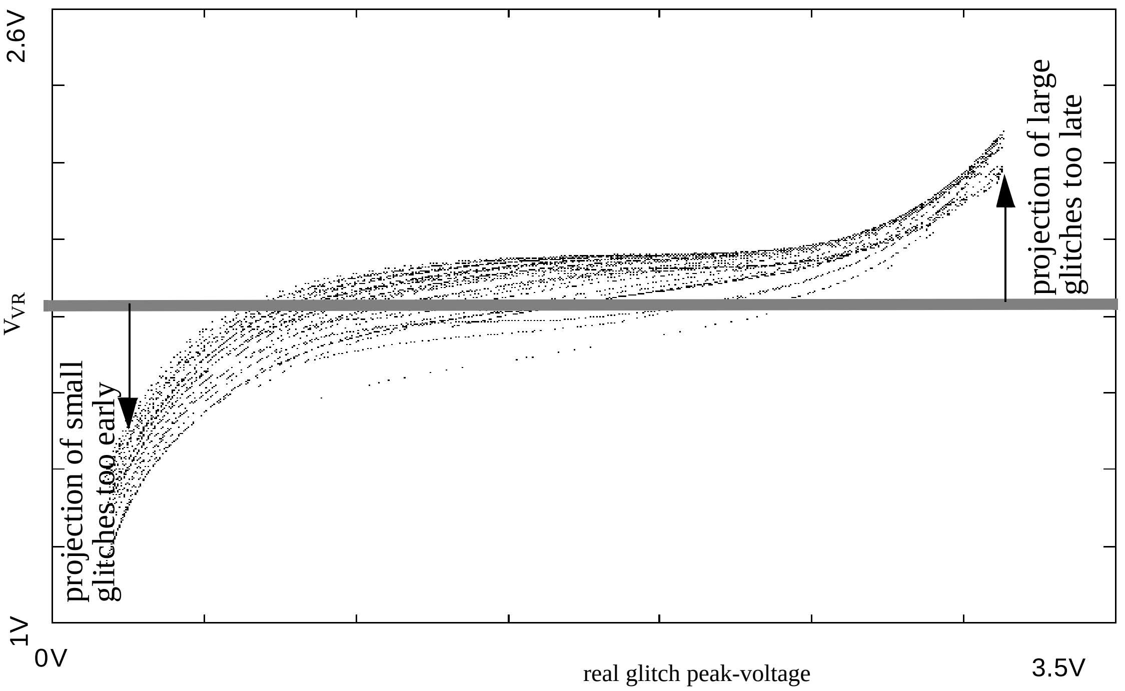


Figure 70:Approximation of V_{VR} by one constant value.

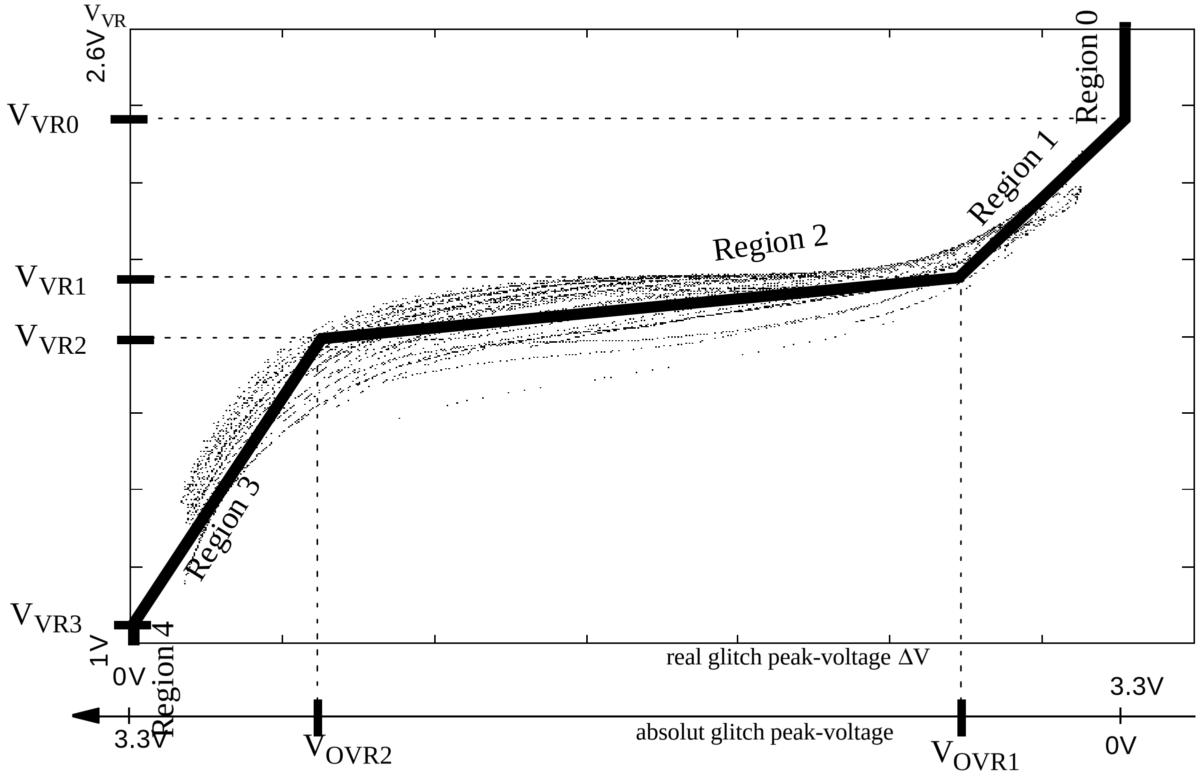


Figure 71:Piecewise linear approximation of V_{VR}.

$$\Delta V_{Glitch} = \begin{cases} 0 \text{ (i.e. in Region 4), if } t_{start\_setOut} \geq t(V_{resetin} = V_{VR3}) \\ VDD \text{ (i.e. in Region 0), if } t_{end\_setOut} \leq t(V_{resetin} = V_{VR0}) \\ \text{in Region 0, 1 or 2, if } t(V_{resetin}=V_{VR2}) \geq t(V_{setout}=V_{OVR2}) \\ \text{in Region 0 or 1, if } t(V_{resetin}=V_{VR1}) \geq t(V_{setout}=V_{OVR1}) \end{cases} \tag{39}$$

The instant t_{start_setOut} refers to the time when the linear approximated ramp starts. Correspondingly t_{end_setOut} refers to its end.

The calculation of the glitch peak voltage for the PWL approximation of V_{VR} can be illustrated as follows (confer Figure 72):

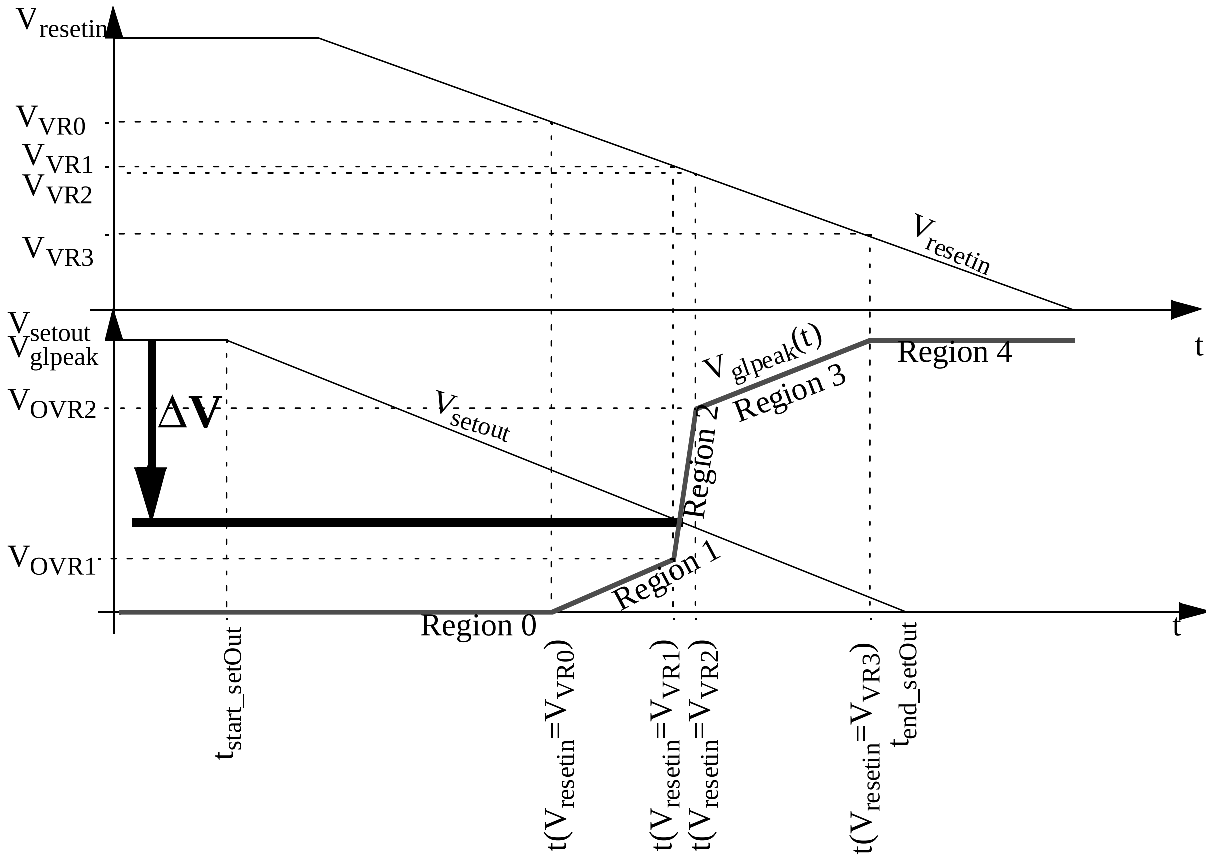


Figure 72:Determination of the glitch peak voltage using a PWL waveform for V_{VR}.

a) determine the different regions from the linear approximated resetting input waveform (upper part of figure),

b) calculate the PWL for the possible glitch peak voltages $V_{glpeak}(t)$ (lower part of the figure) and

c) determine the crossing of this PWL and the setting output’s waveform.

If a constant value is chosen for V_{VR}, the graph $V_{glpeak}(t)$ is a step function (confer Figure 73). The time interval for detecting glitches is shortened by this simplification. Hence it is obvious, that some glitches cannot be detected. On the other hand the computational effort to determine the glitch properties (glitch peak time and glitch peak voltage) is significantly lower and consequently the simulation performance is higher.

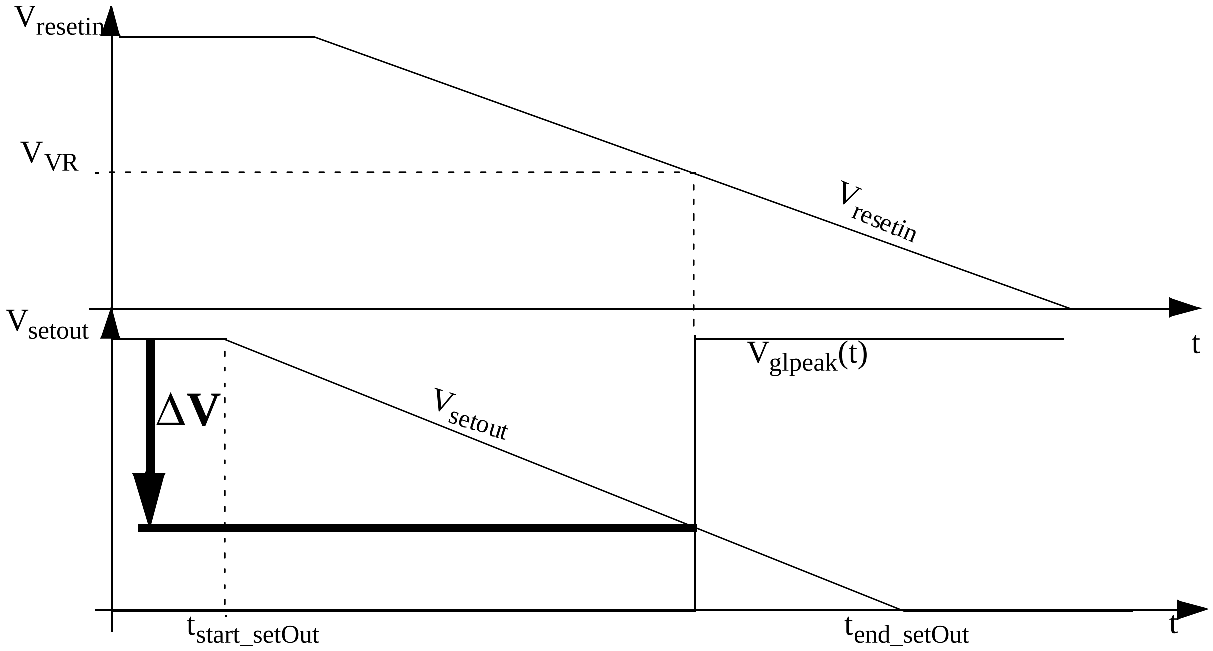


Figure 73:Determination of the glitch peak voltage using a constant value for V_{VR}.

Besides the modelling issues of the model parameters V_{VR}, V_{VF}, V_{TR} and V_{TF} a further source for errors is the approximation of the output waveform by a ramp. This ramp approximation is especially inaccurate for the beginning and the end of a transition waveform (the derivation is lower in this range). Hence further large and small glitches are not detected due to the ramp approximation of the output waveform.

The glitching behaviour strongly depends on the characteristic non-linearity of each CMOS stage. On the cost of accuracy the proposed model can be adapted to multi stage gates by defining a time offset between the *start* of the linearly approximated resetting output ramp and the projection time t_{pt} respectively t_{pv} (for glitch peak voltage and glitch peak time modelling). In Figure74 an example for an AND-2 gate is given. Two input ramps at each primary input result in a hazard at the internal node (i.e. the output of the NAND-2 stage). This hazard turns into a glitch after propagating through the second stage (the inverter). Within the glitch model the following two projection points are used:

- t_{pt}: start of the resetting (non rescheduled) output ramp plus T_{TF},
- t_{pv}: start of the resetting (non rescheduled) output ramp plus T_{VF}.

The projection point t_{pt} is used to determine the crossing time of the two colliding slopes (which is important for dynamic scheduling of the resetting ramp). t_{pv} is used to determine the glitch peak voltage (confer Figure 74).

The main idea is, that the start of the output's waveform is approximately independent from the fanout load. The glitch peak time is very close to the start of the *complete* resetting output waveform (also confer Chapter 4.2.5.4 on pages 91-92). From this observation can be concluded, that the time, when the resetting input (of the last stage within the gate) crosses its characteristic voltage V_{VF}, V_{VR}, V_{TF} respectively V_{TR} is close to the start of the complete resetting output ramp.

When dealing with single stage gates, the projection points (t_{pt} and t_{pv}) are defined by the resetting input ramp. For multi stage gates the resetting output ramp is focused on. The linear ramp-representation of a transition waveform is more accurate for voltages which are close to

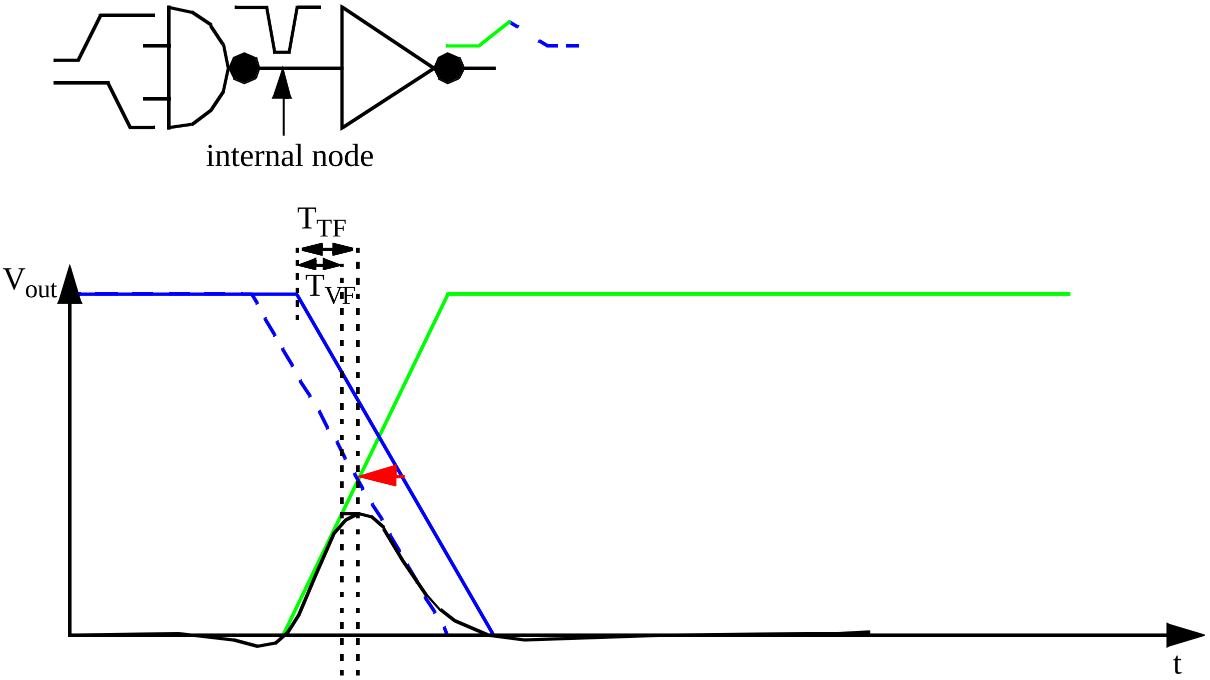


Figure 74:Sketch of modelling glitches for multi stage gates (e.g. an AND-2 gate).

the logic threshold voltage (40% V_{DD} for rising slopes and 60% V_{DD} for falling slopes within the used library). I.e., the start of a ramp, which is used for multi stage gates, is less accurately modelled by the linear ramp representation, leading to a further loss in accuracy for the multi stage glitch model.

For multi stage gates the values T_{TF}, T_{VF}, T_{TR}, T_{VR} are characterized instead of V_{VF}, V_{VR}, V_{TF}, V_{TR}. The V-values are used to obtain the projection points from the resetting input slope for single stage gates and the T-values are used to obtain the projection points from the resetting output slope for multi stage gates.

5.2 Evaluation of the new model

The proposed model has been analysed in the same way as the state of the art models in Chapter 4.2.5.4. For the proposed model two alternatives are distinguished:

- proposed model: using constant values for the glitch modelling parameters V_{VR}, V_{VF}, V_{TR}, V_{TF} or
- enhanced proposed model: using PWL waveforms for the glitch modelling parameters.

Instead of really using PWL waveforms (confer Figure 71), the parameters were determined directly from the resetting input waveform of the circuit level simulation for each simulation case. I.e., the only source for inaccuracies is the approximation of the waveforms by ramps. The simulation results are illustrated in Figures 75-77.

The glitch peak voltage estimation (Figure 75) of the proposed and the enhanced proposed model are the most accurate for all three nodes (mean value). The standard deviation of the results is similar to the results of the modified [Eise95] model. The characteristic glitch parameters for the proposed model are obtained by a single characterization run of a single circuit situation. The mean value of the glitch peak voltage and glitch peak time error can be further

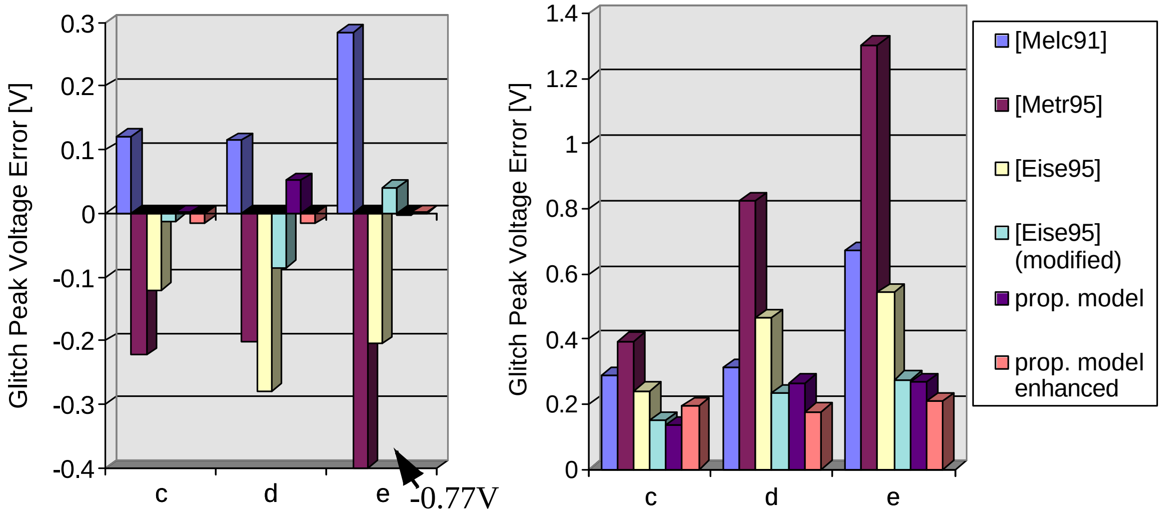


Figure 75:Glitch peak voltage error: mean value (left), standard deviation (right).

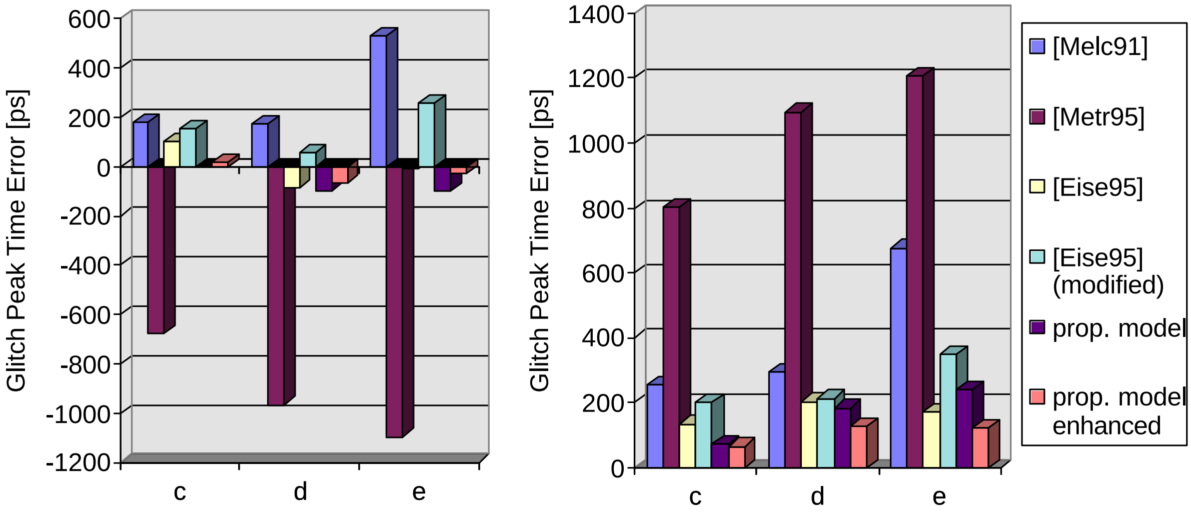


Figure 76:Glitch peak time error: mean value (left), standard deviation (right).

optimized by averaging between several characterization runs. The increase in accuracy by the enhanced proposed model over the simple proposed model is only very small.

The accuracy of the proposed model's glitch peak time estimation is in the same range as the [Eise95] model. However, for the [Eise95] model only glitches with its peak voltage above the logic threshold were considered. I.e., for glitch peak voltages the modified [Eise95] model gives better results than the [Eise95] model and for the glitch peak time it is the other way around. In contrast to the [Eise95] model the proposed model uses different projection times to model the glitch peak voltage and the glitch peak time. The benefit is, that the proposed model gives good results for both glitch characteristics.

Using the enhanced gate level glitch model, more glitches from the circuit level simulation reference are detected than for the simple proposed model. The reasons have been discussed in the previous subchapter. This high detection rate of the enhanced proposed model is paid by a

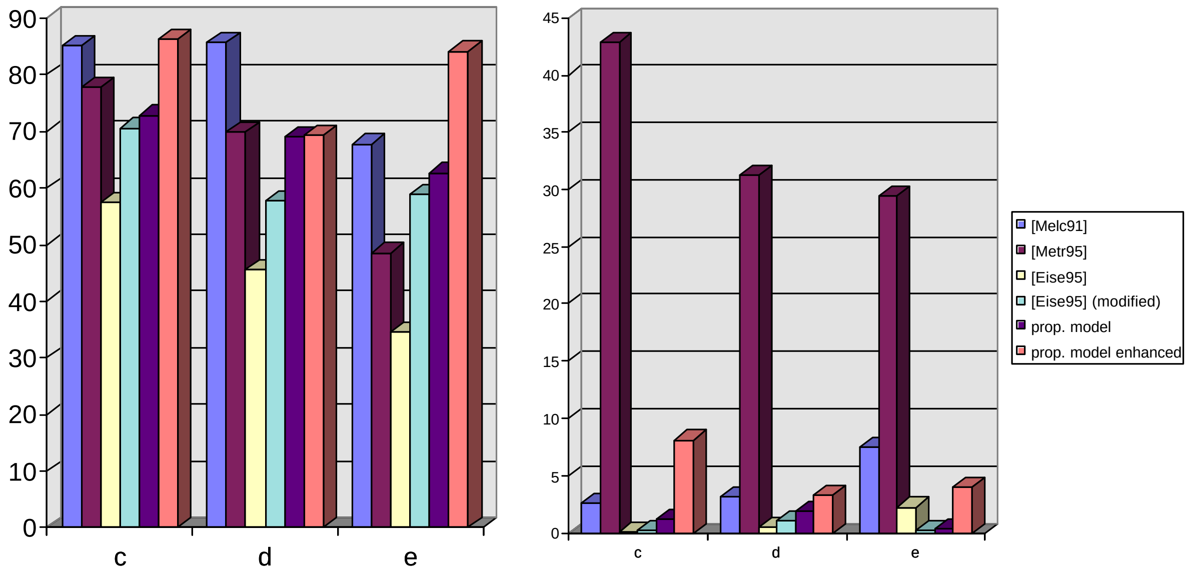


Figure 77:Relative amount of simulated glitches on circuit level which are detected by the gate level models (left), relative amount of detected glitches by the gate level models which are no glitches on circuit level (right).

slightly higher detection rate of glitches which are no glitches within the reference circuit level simulation. The reason is, that the enhanced proposed model detects more small and large glitches around 4% respectively 96% V_{DD} of the reference circuit level simulation than the simple proposed model. Small variation in this area may result in cases, where the circuit level simulation is just outside the glitch detection margin [4%V_{DD},96%V_{DD}] but the model still detects a glitch.

Another characteristic, which can be observed from the left part of Figure 77, is that the missing dynamic scheduling mechanisms result in lower detection rates deeper in the circuit [Metr95]. The lower accuracy of the non modified [Eise95] model for the glitch peak time also results in a decreasing amount of detected glitches for the nodes d and e.

Now the benefit of using different glitch parameters within the proposed model for each input to output pin combination is discussed. The model [Eise95] uses a similar algorithm with fixed glitch parameters. The importance shall be exemplified by the results of a single stage NAND4 analysis. As it has been discussed above, the input to output coupling has the largest impact on the glitch characteristics, if the switching transistors are connected close to the output. The discussed NAND4 gate has the 4 inputs A, B, C, D and the output Y. Input D is connected to the NMOS transistor, which has its drain connected to its output and input A is connected to the NMOS transistor whose source is connected to V_{SS} (confer Figure 78). Hence, the following cases are the extreme cases:

- The setting input slope is applied to input A and the resetting input slope to input D.
- The setting input slope is applied to input D and the resetting input slope to input A.

A typical input slopes is chosen for both glitch causing input transitions. The fanout load was varied: 0%, 20%, 40%, ..., 200% of C_{max}. The glitch parameters V_{VR} and V_{TR} were extracted for glitches of the following glitch peak voltages: 0.5V, 1V, 1.5V, ..., 4.5V. The glitch parameters were obtained by projecting on the non linearized complete resetting input transition. The results are shown in Figure 78. The variation of the parameters due to the different fanout loads

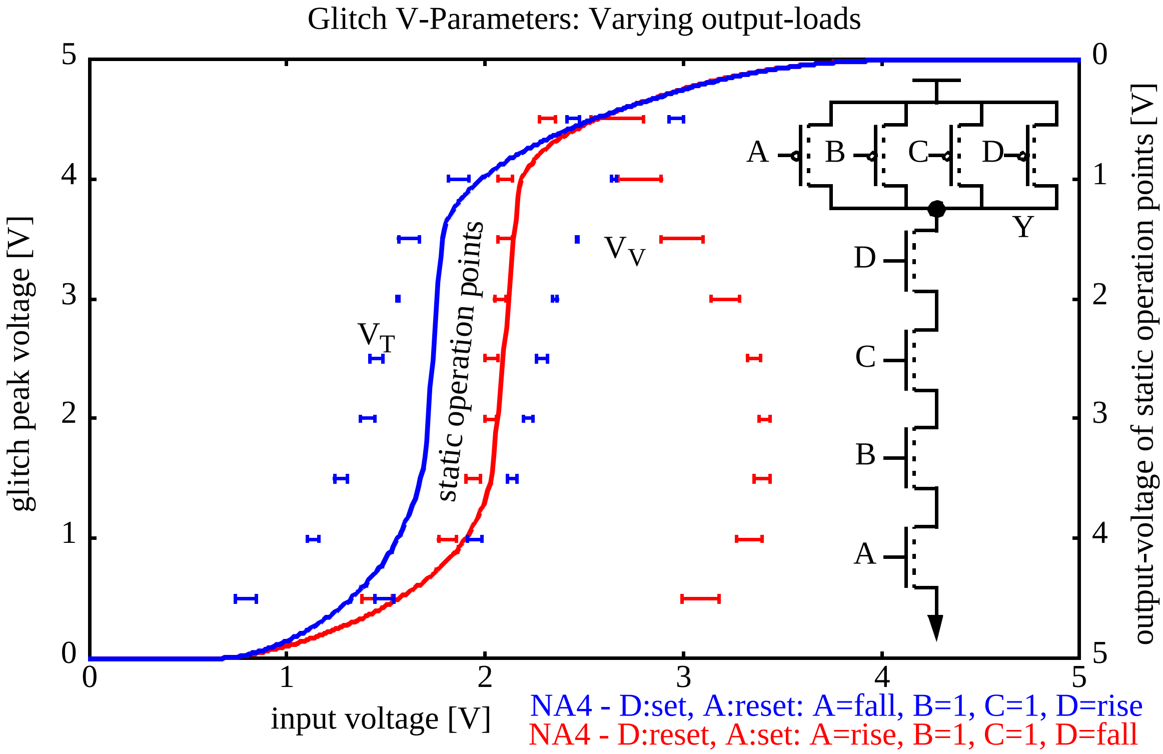


Figure 78:Static operation points and glitch parameter V_V and V_T for a NAND4-gate.

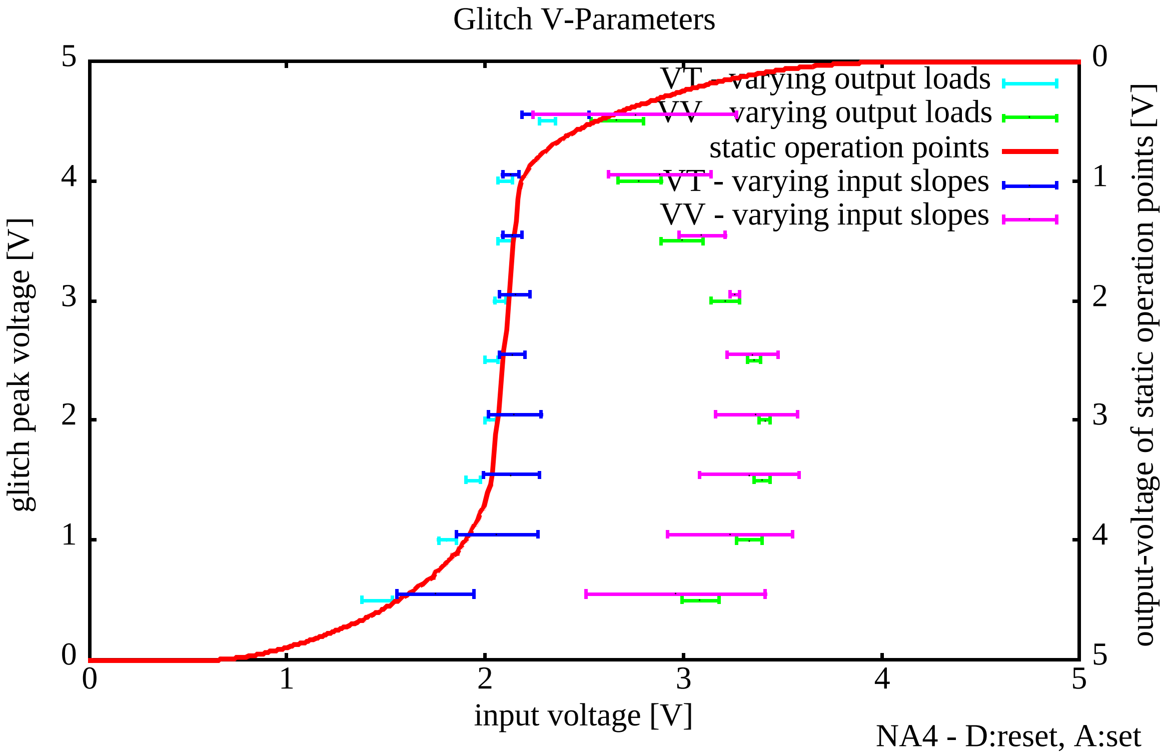


Figure 79:Static operation points and glitch parameter V_V and V_T for a NAND4-gate with varying input slopes and output load.

is plotted for each glitch peak voltage. In addition to the glitch parameters the static operation curves are also plotted for input A and D (all other inputs connected to V_{DD}). The reason for the difference of the two static operation curves is the body effect. Consequently, the body effect also has an impact on the equilibrium state of glitches and the glitch parameters of different pins.

The input- to output coupling influences the V_V parameter of input D significantly. The falling input transition drags the glitching output waveform significantly stronger down than other inputs due to the topological location of the NMOS transistor, to which input D is connected. By the time, when the resetting input waveform crosses the V_V voltage, the gate is significantly before reaching the equilibrium state. Consequently the actual glitch peak voltage on the complete setting waveform (projection time) is reached before crossing the static operation curve (confer Figure 78). Hence the impact of the input slope on the variation of the V_V parameters is significantly higher for input D than for any other input. The V_T parameter describes the voltage for the resetting input voltage, which defines the glitch peak time. This instant is significantly less influenced by dynamic coupling effects, because it is closer to the equilibrium state than the projection instant for the glitch peak voltage. In addition to impact of fanout variations on the glitch parameter (Figure 78), the impact of input slope variations are shown for input D in Figure 79. The dynamic impact of input slope variations is more significant. For other input pins the variations are smaller.

For library characterization - including glitch characterization for the proposed model - the automatic characterization tool OCHATO (*O*ffis *Cha*racterization *To*ol) was implemented [Vöge97,Vöge98]. For glitch characterization glitches and hazards are applied to the corresponding inputs for $\Delta V \approx V_{DD}/2$. The characterization results of the glitch parameters is shown in Table 11 for NAND-gates with different numbers of inputs and driving strengths.

Cell Name	**Pin Combination**	V_{VR} [V]	V_{VF} [V]	V_{TR} [V]	V_{TF} [V]
LIBNA2	A->Y	2.7950	1.1927	1.8445	2.0786
	B->Y	3.3667	1.2387	2.0154	2.1799
LIBNA2D	A->Y	3.1683	1.2549	2.0154	2.1799
	B->Y	3.1642	1.2578	2.0154	2.1799
LIBNA3	A->Y	2.4522	1.2120	1.6341	2.1278
	B->Y	2.8205	1.2588	2.0239	2.1799
	C->Y	3.4068	1.2776	2.2117	2.1814
LIBNA4	A->Y	2.2975	1.2532	1.4608	2.1814
	B->Y	2.4522	1.2737	1.7275	2.1814
	C->Y	2.9018	1.3096	2.2117	2.3988
	D->Y	3.3901	1.2714	2.4073	2.1814

Table 11:Glitch characterization data of an industrial 0.7mm library (V_{DD}=5V) for a variety of NAND-gates.

The LIBNA2D gate consists of 4 NMOS transistors with 2 transistors in a row each. The location of the transistors is shown in Figure 80. Due to this topology, the input characteristics of the two inputs A and B are approximately equivalent, which is also visible from Table 11. The variation of the parameters V_{VR} and V_{TR} for different inputs is significantly higher than for the parameters V_{VF} respectively V_{TF} (exception LIBNA2D).

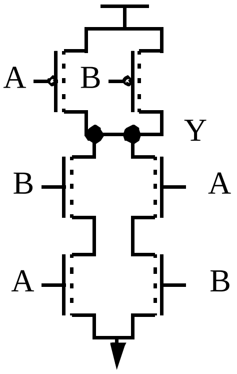


Figure 80:Transistor Schematic of the library cell LIBNA2D (stronger than common LIBNA2).

In order to accurately consider such input pin variations, different glitch parameters are used for each input to output combination by the proposed model. The comparison of the state of the art glitch models with the proposed model has been presented for a single testbench. By exchanging the gate's input pins of the testbench, (in contrast to the proposed model) the accuracy of the model proposed by [Eise95] varies. The reason is, that the model is not based on different glitch parameters for each input pin.

In conclusion the new model has been introduced as a robust and accurate model. The accuracy has been exemplified in comparison with other state of the art models. The high flexibility to consider different gate characteristics within the gate level model makes it robust. In Chapter 7 the efficient implementation of the model within a simulator is dealt with and the simulator's performance and accuracy is dealt with in Chapter 8.

6 Gate Level Power Model

The power model is targeted for library based CMOS circuits. The main contributor to power consumption of today’s CMOS technologies is dynamic power consumption. For this reason only the dynamic contributor is focused on. The omission of static components is no limitation of the model, but only a practical simplification. I.e., no blocking points exist to include the static power component into the model respectively into the implemented power simulator.

Changing voltages at cell’s internal and external cell nodes are the cause for dynamic power consumption (capacitive and short circuit power consumption). To take these node transitions into account, power triggers are defined.

Definition 17: ***Power trigger:***
A power trigger describes a transition in a certain direction of a dedicated physical or a combination of physical signals, which cause a certain amount of power consumption. Summing up all these power contributors within a circuit or part of it gives its total power consumption.

Typical power triggers are output transitions, which are caused by a certain input transition. These power triggers are the same as the delay paths, which are used for delay characterization (confer Chapter 3.2.1.2). For more general Boolean gates

- the resistive path within the cell for charging and discharging capacitances and
- the capacitances to charge respectively to discharge

depend on the state of further (typically stable) cell nodes (confer Chapter 3.2.1.2 and Figure 81). In addition to Chapter 3.2.1.2 for a rising output transition 3 cases are distinguished

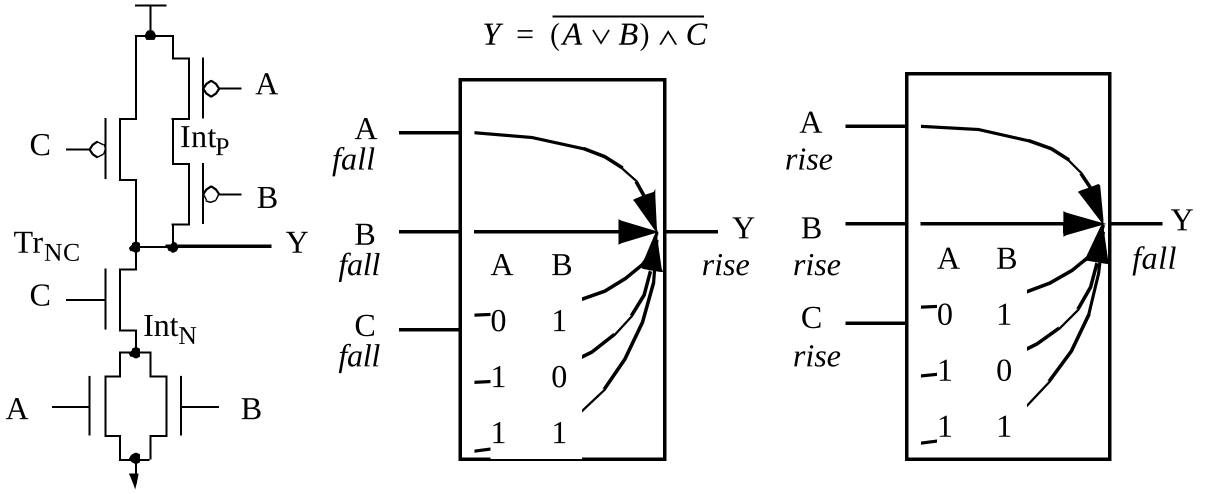


Figure 81:Example for a single stage Boolean function, with multiple possible paths from the input to the output.

from C to Y, which have been neglected for the discussion of the delay. Such a Boolean condition may be included within the combination of physical signals, which define a power trigger. For the example given in Figure 81 the following power triggers are defined:

Case	**Y**	**A**	**B**	**C**	**remark**
1	falling	rising			implicit conditions are B=0, C=1
2	falling		rising		implicit conditions are A=0, C=1
3	falling	0	1	rising	voltage of node Int_P remains unchanged[‡]
4	falling	1	0	rising	voltage of node Int_P changes
5	falling	1	1	rising	node Int_P is floating and effective resistance of NMOS network is lower than for case 3 and 4
6	rising	falling			implicit conditions are B=0, C=1
7	rising		falling		implicit conditions are A=0, C=1
8	rising	0	1	falling	voltage of node Int_P remains unchanged[‡]
9	rising	1	0	falling	voltage of node Int_P changes
10	rising	1	1	falling	node Int_P is floating and effective resistance of NMOS network is lower than for case 8 and 9

Table 12:Possible power triggers for the cell $Y = \overline{(A \vee B) \wedge C}$ *(refer to Figure 81).*

[‡] The internal node voltage remains only approximately unchanged, because the capacitive coupling with other switching nodes may result in a small change in the range of $[V_{DD}+V_{th}, V_{DD}]$ for internal nodes within the PMOS network (respectively $[V_{SS}, V_{SS}+V_{th}]$ within the NMOS network).

Within this example only external pins have been used to define power triggers. Additionally internal nodes can be defined (e.g. for the above example: Int_P and Int_n). Such internal nodes are especially important for sequential cells. An example of a flip-flop schematic is given in Figure82. During the clock-low phase all transitions at input D cause power consumption at

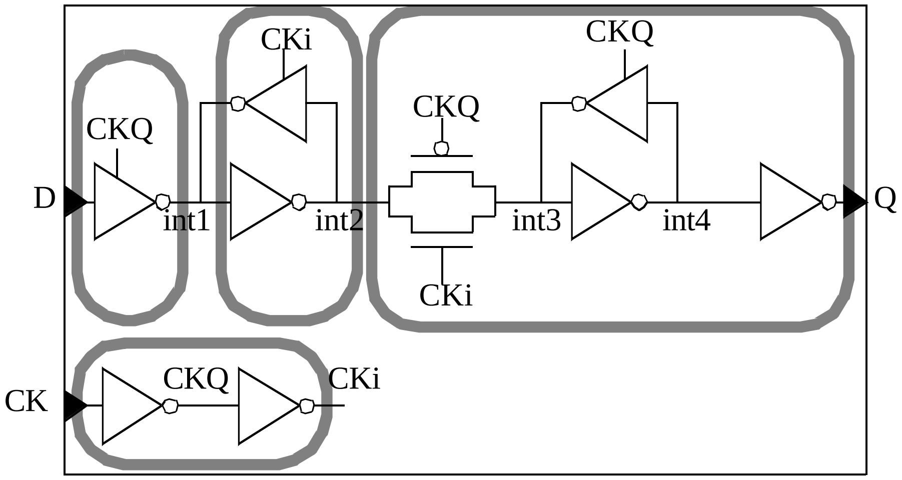


Figure 82:Schematic of a positive edge flip-flop.

the first two inverters with its outputs connected to nodes int1 and int2. Another very important contributor to power consumption is a clock transition (at input pin CK), which also switches nodes CKQ and CKi. Caused by a positive clock transition, the internal nodes int3, int4 and the output Q of the slave latch may also switch depending on their previous state. Consequently the following power triggers should be used for modelling the flip-flop as a single cell:

Case	**Q**	**CK**	**D**	**int2**	**remark**
1	falling	(rising)			falling transition at output of slave (int3 and int4 also switch)
2	rising	(rising)			rising transition at output of slave (int3 and int4 also switch)
3		falling		rising	input D has changed its value with respect to the last rising CK transition and the change is now stored in the master
4		falling		falling	
5		0		rising	a transition at input D is latched into the master
6		0		falling	
7[‡]		rising			inverter chain connected to the CK pin switches
8[‡]		falling			

Table 13:Possible power triggers for the cell $Y = \overline{(A \vee B) \wedge C}$ *(refer to Figure 81).*

[‡] The power contribution of case 7 and 8 must be excluded from the cases 1-4 in order to not count it twice.

Glitches may occur only at the internal nodes int1 and int2 for cases 5 and 6 (under the common assumption, that the clock is glitch free). In order to properly take them into account, the flip-flop may be separated into subcircuits as indicated by the grey boxes in Figure 82. However, as glitches cannot propagate through flip-flops, only the relatively small amount of internal power consumption would be taken into account, which is practically negligible (internal interconnection are very short compared to interconnected interface pins).

A certain amount of power consumption is associated with each power trigger. To determine the actual amount further parameters like input slopes and output load need to be considered. For the output load of a pin a fixed value can be determined from the circuit topology. The sources for inaccuracies of the capacitive value have been dealt with in Chapter 3.5.1. Input slopes may vary for different propagation paths. Therefore input slopes are considered dynamically during simulation, when a power trigger becomes true. For each library cell the power triggers are characterized for a number of different circuit situation in terms of input slope and output load. As not each possible circuit situation can be characterized a priori, linear interpolation is used to calculate the actual load consumption from a set of characterized circuit situations. The more characterization values are available, the less is the interpolation error but the more data need to be searched for the right reference data, which results in loss of simulation performance.

In case of incomplete transitions at a cell's output node i the load consumption of the corresponding complete transition is scaled by $|\Delta V_i| / V_{DD}$:

$$Q_{\text{glitch transition}} = Q_{\text{complete transition}} \cdot \frac{|\Delta \mathrm{V}|}{\mathrm{V}_{\mathrm{DD}}} \tag{40}$$

The energy consumption is $E = Q \cdot \mathrm{V}_{\mathrm{DD}}$ for a single gate. Hence, under the assumption that all gates are operating with the same supply voltage, the power consumption of the whole circuit is

$$\mathrm{P} = \lim_{T \to \infty} \frac{\sum_{\text{all gates}} \mathrm{E}_{\text{gate}}}{T} = \mathrm{V}_{\mathrm{DD}} \cdot \lim_{T \to \infty} \frac{\sum_{\text{all gates}} Q_{\text{gate}}}{T} = \mathrm{V}_{\mathrm{DD}} \cdot \overline{\mathrm{I}} \,. \tag{41}$$

The short circuit current, which has a high impact on glitch power consumption (confer Chapter 3.5.3), is also scaled by $|\Delta \mathrm{V}_{\mathrm{i}}| / \mathrm{V}_{\mathrm{DD}}$. This is a further source of errors within the power formula. The reasons for not properly handling the short circuit power of glitches are the additional characterization effort, the loss in simulation performance and the lack of an appropriate model.

7 Simulation Algorithm and Implementation

The integration of the new glitch model (confer Chapter 5) into a logic simulation algorithm and its implementation is focused on in this chapter. The implementation was realized in a stand-alone simulator named GliPS (Glitch Power Simulator). Only the aspects which are important from the glitch handling point of view are discussed in detail. The correct consideration of other aspects like for example resolution functions and flipflops is dealt with in [Mart97].

First the simulator's interface is described. The general simulation algorithm is dealt with in Chapter 7.2. In Chapter 7.2.3 different glitch situations are analysed and the handling of some special situations is discussed in detail. The control of the simulator is specified in Chapter 7.3.

On the base of the specified glitch situations and control flow, GliPS was implemented. Simulation results are given in Chapter 8.

The main feature of this simulation algorithm and its implementation is the enhanced handling of glitches. The only purpose of the implementation is to practically validate the simulation algorithm's accuracy and its simulation performance with respect to other algorithms, which are implemented in commercially available simulators.

7.1 Interfaces of GliPS

In Figure 83 the interfaces of GliPS are shown. A netlist is written out of the Cadence Design Framework Environment via a customized netlister. The generated netlist is easy to parse into the simulator's datastructure. The actual parasitic capacitances are also read in and stored in the datastructure. Within a user supplied option file control informations are defined. The input stimuli can either be specified as complete waveforms in the option file or can be defined within a stimuli file in a more efficient (i.e. more user friendly) way. The stimuli file contains the logic input pattern which are applied to the circuit within a defined strobe. This data is used for the simulation by GliPS.

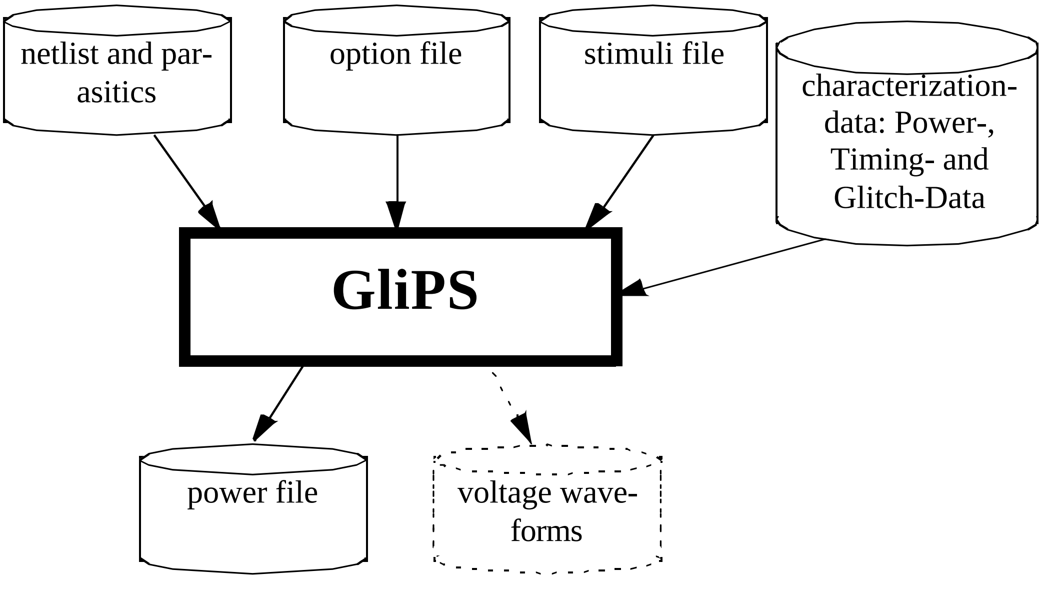


Figure 83:Interfaces of GliPS.

As a result the power data is calculated for each gate's output node. The power file contains power data for the whole circuit and each instance. For graphic representation of user defined voltage waveforms a file can be generated.

7.2 General simulation algorithm

An event driven simulation algorithm is used for simulation. First the logic value system is explained and in the consecutive subchapter the event handling algorithm is dealt with.

7.2.1 Logic value system

Unlike in most logic simulators the logic value system is predefined within the implemented simulator to keep it simple. The following 5-value logic system is used:

value	meaning
0	strong low: Associated voltage is V_{SS}
1	strong high: Associated voltage is V_{DD}
X	unknown: either non initialized or busconflict
L	weak low: Associated voltage is V_{SS}
H	weak high: Associated voltage is V_{DD}

Table 14:5-value logic system.

The value X is the default value, which is initially assigned to all circuit nodes. A driving conflict of 0 and 1 respectively L and H is resolved to X. The resolution tables are hard coded for the used primitives on which the functional description of the gates is based on.

The values L and H are used to model the following two aspects:

- Weak driver, which can be overdriven by a strong driver:
  Important examples are busholders and weak drivers for feedback loops.
- Signals, which were driven by a 0 respectively 1 value before the driver(s) has respectively have been switched off:
  Physically, such signals represent the voltage at drivers' output pins and the connected wire. All output pins are floating and the last driven voltage is capacitively stored. Due to coupling and leakage effects the voltage may drift away. This drifting effect could be taken into account by assigning the X value to such a signal after a certain decay time (similar as in Verilog XL). This feature is not implemented in GliPS.

Most logic simulators respectively the underlying hardware description languages support user defined logic value systems. The association of distinct voltage levels to these logic values is also very important to be able to model glitches.

Dynamic power triggers are associated with signals changing its value. Basically only falling and rising transitions are considered. The mapping of all possible transitions on these two transitions is given in Table 15:

current value **next value**	**0**	**1**	**X**	**L**	**H**
0	-	fall	(fall)	-	fall
1	rise	-	(rise)	rise	-
X	(rise)	(fall)	-	(rise)	(fall)
L	-	fall	(fall)	-	fall
H	rise	-	(rise)	rise	-

Table 15:Mapping of all possible transitions in the 5 value system on rising (rise), falling(fall) and no transition(-) (transitions which are written in brackets are considered as transitions with 50% probability).

The X-values do not occur for well designed circuits after the initialisation phase. However, if they occur, the power contribution is approximated with 50% of a complete rising respectively falling transition. The propagation delays are considered according to the transition direction.

7.2.2 Event driven simulation algorithm

Within conventional simulation algorithms, events are scheduled for the time, a library defined logic threshold voltage is crossed and the signal change is modelled as a sharp edge (refer to Chapter 4.2). The slowly changing voltage-waveform of the signal, which lies behind this model, is not considered. As long as the real signal is between V_{SS} and V_{DD}, a possible additional input event might lead to a glitch at a gate's output event. Therefore the signal output waveforms are modelled as linear approximated ramps (confer Figure 84).

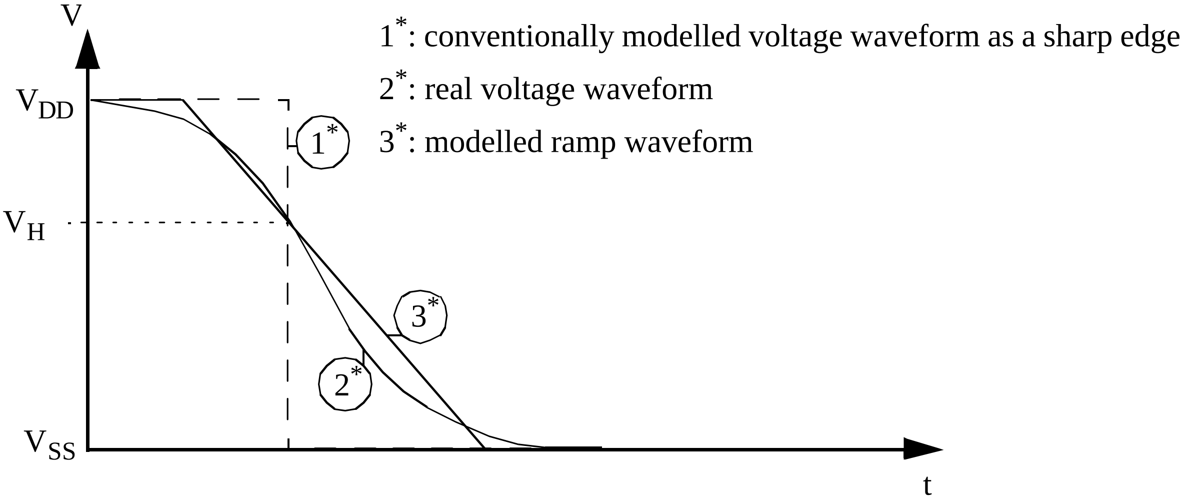

V
V_{DD}
V_H
V_{SS}
t
1^*
2^*
3^*
1^*: conventionally modelled voltage waveform as a sharp edge
2^*: real voltage waveform
3^*: modelled ramp waveform

Figure 84:Modelling of signal changes.

The ramp is represented by two events:

- the *begin* event and

- the *end* event.

The first is the *begin* event for the instant when the ramp becomes active. When the *begin* event is processed by the simulator the ramp is queued again as an *end* event for the instant when the ramp becomes inactive. The term *active* refers to the time, during which the modelled voltage-waveform is between V_{SS} and V_{DD} for single transitions. In case of glitches the active part of a ramp is defined by the period, it describes the voltage waveform of the glitch. E.g. in Figure85 the active parts of the output ramps are indicated by the bold part of the solid lines.

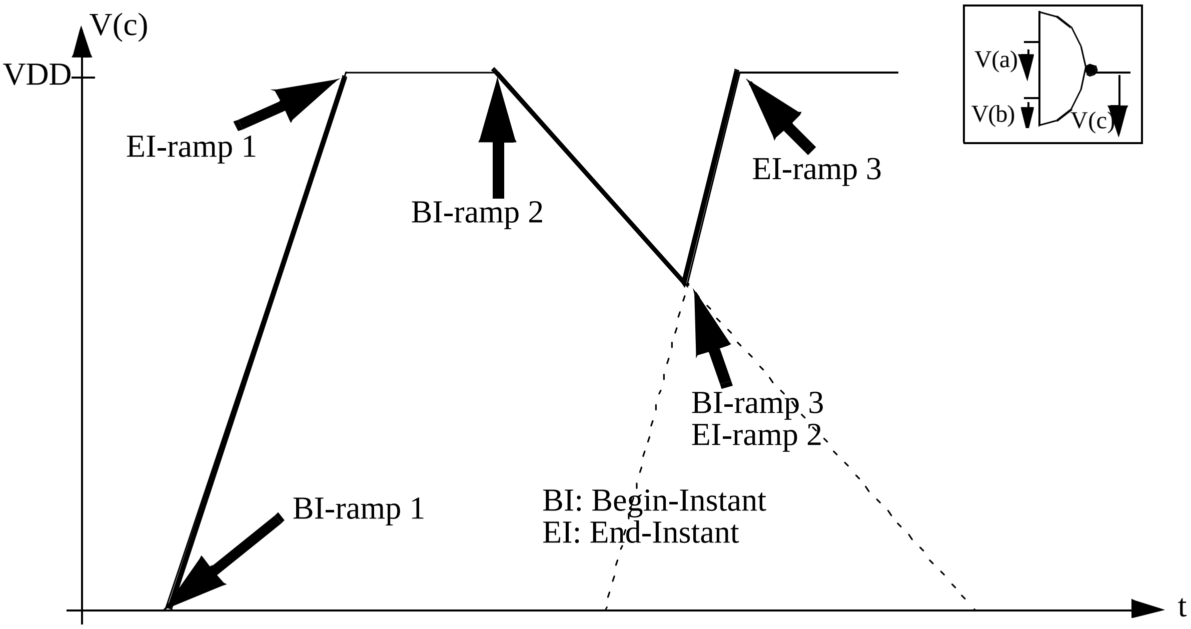


Figure 85:Ramp handling within local output event queues.

I.e., the instant when a ramp becomes active is either

- the time when leaving VDD respectively VSS or
- in case of a resetting ramp as part of a glitch the time when crossing the setting ramp.

The deactivation of a ramp takes place either

- when reaching VDD respectively VSS or
- in case of a setting ramp as part of a glitch when crossing the resetting ramp.

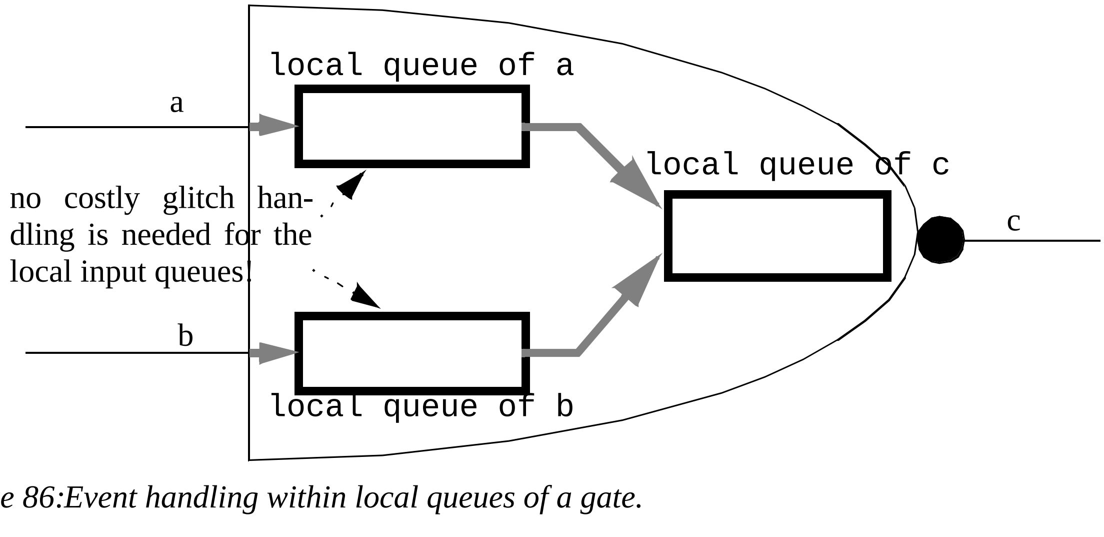

local queue of a
a
local queue of c
c
no costly glitch handling is needed for the local input queues!
b
local queue of b

Figure 86:Event handling within local queues of a gate.

For the three ramps of the example in Figure 85 the *begin* events and *end* events are indicated by *BI-ramp n* respectively *EI-ramp n* (n={1,2,3}).

Generally the simulation algorithm is based on two types of event queues:

- local event queues: for a signal all events are organized within its local queue,
- global event queues: the global event queue ensures the execution of the events from the local event queues in the right time order.

For input and output pins local event queues are used (confer Figure 86). Actually two separate global event queues are implemented for input events and output events.

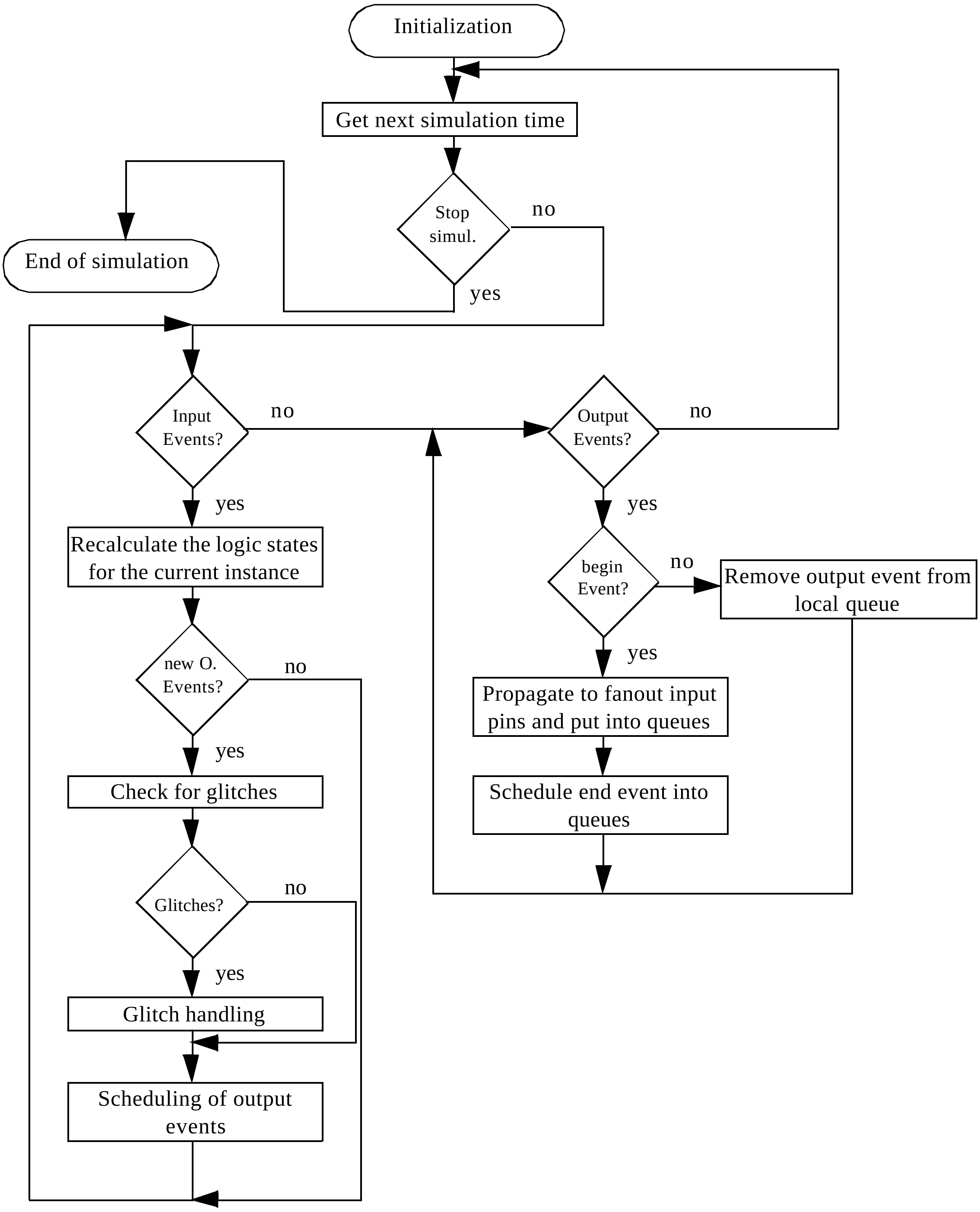

Initialization
Get next simulation time
Stop simul.
no
yes
End of simulation
Input Events?
no
yes
Output Events?
no
yes
Recalculate the logic states for the current instance
begin Event?
no
Remove output event from local queue
yes
new O. Events?
no
yes
Propagate to fanout input pins and put into queues
Check for glitches
Schedule end event into queues
Glitches?
no
yes
Glitch handling
Scheduling of output events

Figure 87:Flowchart of the basic simulation algorithm.

The general simulation algorithm is described by the flowchart in Figure 87. After initialization of the circuit, the first simulation time is read from the global event queues.

For that simulation time all input events are processed first. For each input transition the new signal value is assigned (one at a time) to the input and the gate behaviour is recalculated. In case of a new event it must be checked, whether the new event is a resetting event of a glitch. In case of a glitch, the glitch handling algorithms must be applied to correctly consider the dynamic scheduling time and glitch peak voltage. The new output event is then scheduled into the respective local and global output queues. As soon as no further input event is pending for the current simulation time, all output events are processed next.

Begin events at a gate's output-queue(s) are scheduled into the input queues of driven input pins of consecutive gates. The event time is the Begin-Instant of the ramp. To properly represent the ramp each event holds the following attributes: the slope, the start- and end-voltage. If the presented output event is not a begin event, it is simply removed from the local queue. After processing all events for the current simulation time, the next simulation time is read from the global event queues. As *begin* events at a gate's output have been propagated to fanout input queues, the described flow is typically run once more for the same simulation time.

The simulation continues until a user defined stopping criterion, which is not further discussed here, is satisfied.

In causal systems the transfer through a physical gate is always delayed. As the actual logic evaluation is done when processing a begin event of an input, it cannot have an impact before that instant. When choosing other predefined logic threshold voltages for events, negative delays may result for conventional simulation algorithms (confer Chapter 3.2.1). As the proposed simulation algorithm is intended to model this physical behaviour accurately, delays lower or equal to zero can not occur. If such cases occur during simulation, they are based on inaccurate delay characterization data and can be set to zero.

7.2.3 Glitch handling

The storage of all output ramps in the output queue for the whole activation time enables the application of the proposed glitch model (confer Chapter 5). Within this subchapter the different interactions of more than one event on a single output are discussed.

For the glitch algorithm projection times are used for
- dynamic scheduling of a resetting ramp (t_{pt}) and
- glitch peak voltage determination (t_{pv}).

I focus on an output queue here, for which a possible new event is generated. All events which have been previously inserted with a begin-instant later than that of the current ramp are deleted first. For efficiency reasons these events are only marked as deleted and dequeued from the local event queue in order to avoid the search in the global event queue for deletion. The actual memory is freed when processing the respective event from the global event queue. Within the local event queue at least one event is remaining which has an opposite edge direction as the new event. Hence the last event in the local event queue is possibly a setting event and the new event a resetting event. The currently last event in the local event queue is referred as setting event in the following discussion.

Within this subchapter the following instants are important:

- $t_{beginset}$ (used abbreviation: t_{bs}): instant when the active setting output ramp (last event in local event queue) starts; i.e., if the setting ramp is a resetting ramp of a previous glitch, $t_{beginset}$ is the crossing time with the corresponding setting ramp.
- t_{endset} (used abbreviation: t_{es}): instant when the setting output ramp ends.
- $t_{beginreset}$ (used abbreviation: t_{br}): instant when the complete resetting output ramp begins; i.e., the instant refers to the start of the ramp before glitch handling.
- $t_{endreset}$ (used abbreviation: t_{er}): instant when the complete output resetting ramp ends.
- t_{cross} (used abbreviation: t_c): crossing instant of the setting and resetting ramp (before glitch handling is done); if the voltage, the ramps cross each other is not within the interval $[V_{SS},V_{DD}]$, t_{cross} is defined as $t_{beginreset}$ - i.e.:

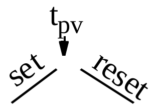
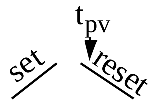
$$t_c = \begin{cases} \text{crossing time}, & \text{if } t_{es} \geq t_{br} \\ t_{beginreset}, & \text{if } t_{es} < t_{br} \end{cases} \tag{42}$$

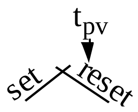
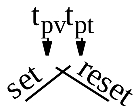
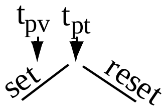
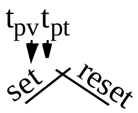
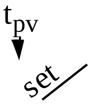
- $t_{simulationtime}$ (used abbreviation: t_{st}): current simulated time

By the role in which these 6 instants and the projection-instants t_{pv} and t_{pt} occur, different simulation cases are defined. In principle 6! (720) combinations (i.e. different simulation cases) are obtained. It would be hard to consider each of these cases within the simulation algorithm. Fortunately, the following constraints, which can be used to reduce the number of possible cases of interest, can be applied:

- $t_{bs} < t_{es}$,
- $t_{br} < t_{er}$,
- $t_{br} \leq t_c < t_{er}$,
- $t_{pv} < t_{pt}$ (basic glitch characteristic),
- $t_{pt} < t_{er}$ - if this constraint is not met within the simulation due to non accurate characterization data, simply the following assignment is done: $t_{pt}=t_{er}$; if further the constraint $t_{er} > t_{pv}$ is not met, the following assignment is done: $t_{pv}=t_{er}$,
- $t_{st} \leq t_{er}$,
- $t_{st} \leq t_{bs}$,
- $t_{st} \leq t_{pv}$ - if this constraint is not met within the simulation due to non accurate characterization data, simply the following assignment is done: $t_{pv}=t_{st}$; if further the constraint $t_{st} \leq t_{pt}$ is not met, the following assignment is done: $t_{pt}=t_{st}$.

The following table contains all remaining possible cases which might occur during simulation and need to be distinguished. The columns entitled by 1, 2, 3, 4 give the order of occurring events. The situation is described and the treatment is given:

#	1	2	3	4		situation: description and treatment
a	t_{es}	t_{pv}	t_{br}			Hazard: i.e., no dynamic scheduling of the new ramp is done, the new event performs a full VDD-swing (so far)
b	t_{es}	t_{br}	t_{pv}			

#	1	2	3	4		situation: description and treatment
c	t_c	t_{es}	t_{pv}			Glitch, no dynamic scheduling of the new ramp, scheduling instant is t_c, glitch peak voltage is given by the voltage at $t=t_c$
d	t_{bs}	t_{pv}	t_c	t_{pt}		Glitch, no dynamic scheduling of the new ramp, scheduling instant is t_c, glitch peak voltage is given by the voltage at $t=t_{pv}$
e	t_{bs}	t_{pv}	t_{pt}	t_c	 or 	Glitch: dynamic scheduling of the new ramp, glitch peak voltage is given by the voltage at $t=t_{pv}$
f	t_{pv}	t_{bs}				Glitch-filtering: new event is not inserted into the event queues and the last event in the local event queue is deleted

7.2.4 Impact of input event processing order on detection of unnecessary transitions

Input events are evaluated by assigning the new logic value to the input at the *begin*-instant of the ramp. Possible output events are calculated under consideration of all other current gate-signals. Glitch handling is included within the calculation. In case of a glitch, the glitch-projection times are calculated to determine the glitch peak voltage and the glitch peak time.

The current implementation of the glitch algorithm has the limitation, that a currently processed input event can only be a resetting event for the last event with opposite transition direction in the output queue. This limitation can result in missing some hazards or more likely glitches. The overall functionality is not degraded for combinational logic blocks, if such a missed unintended glitch or hazard is not applied to an enable input of a latch. For such a case a wrong value might be latched in. However, in well designed circuits only clock synchronised signals are applied to enable pins.

To exemplify such a glitch missing case, an example is given in Figure 88. The ramp at input a causes a change at output c even though it starts later than the falling ramp at b, which is slower as the one at a. The actual waveform at the gate's output is a glitch. Within the simulation the event at input b occurs before the event at input a and is consequently processed first. As the event at input b does not cause a transition at the output, no event is scheduled. When processing the event at input a, the evaluation of the gate's reaction also results in no output event and the glitch is not detected by the simulation.

This glitch missing problem can be easily solved as follows: In case of a transaction at the gate's output without a resulting event, the actual change of input value is assigned when reaching the glitch projection time t_{pv}.

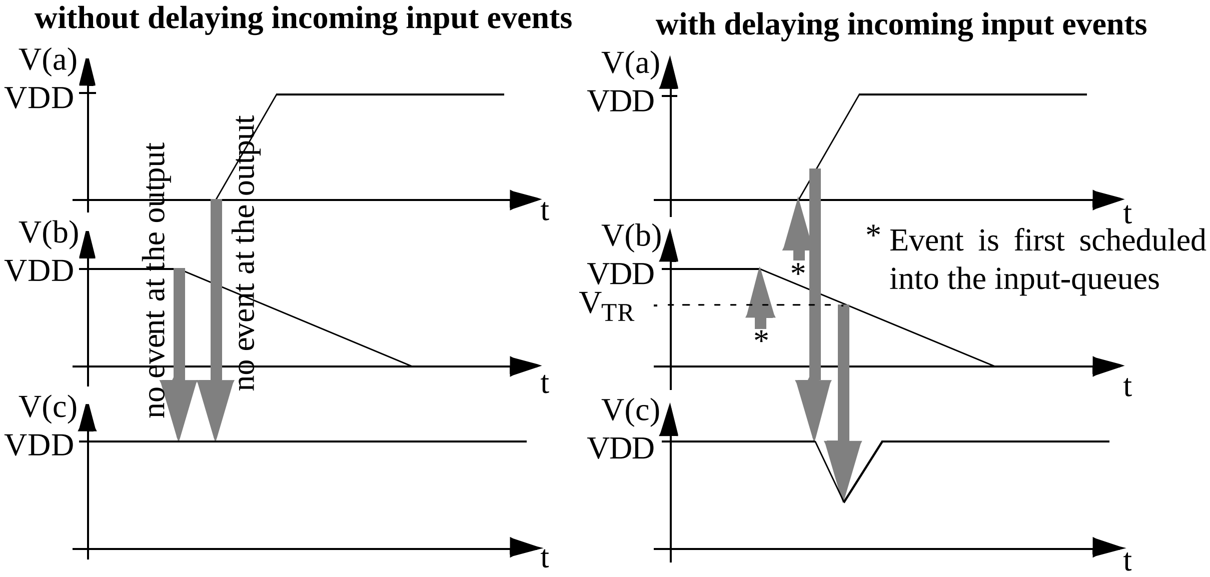

without delaying incoming input events
with delaying incoming input events
V(a)
VDD
t
V(b)
VDD
t
V(c)
VDD
t
no event at the output
no event at the output
V(a)
VDD
t
V(b)
VDD
V_TR
*
*
t
V(c)
VDD
t
* Event is first scheduled into the input-queues

Figure 88:Importance of delaying input events before propagating them to the output: AND-2.

However, this would result in a dramatic increase in events, because all input events, which do not result in an event at an output, would have to be processed twice. Further the probability of such cases is on the one hand not very high and on the other hand the peak voltage of such a missed pair of transitions is typically very small and will be filtered within the consecutive gates. Due to these reasons the currently implemented algorithm does not consider such cases.

7.3 Control of the simulation

The static initialization of the circuit (refer to Figure 87) simply assigns an X value to all circuit nodes. The actual initialization is done dynamically by the applied input events. Up to a user specified simulation time no power contributions are considered and no unnecessary events are counted.

The simulation is related to a clock definition. This clock definition is also needed for non-sequential circuits. The clock is specified within the option file. It is advised to use a sufficiently long clock period, to ensure, that no timing violations may occur. For shorter clock-periods (with equal clock slopes) the power consumption can simply be scaled by the relation f_clock_fast/f_clock_simulator under the assumption that even with a faster clock no timing violations occur. Hence it is proposed to use a conservative clocking scheme.

In relation to the clock events, new input events can be assigned and power convergency checks are done. For the input waveform definition the user has two possibilities:

- non-clock related waveforms can be defined within an option file for each primary input-pin,
- within a patternfile for all primary input-pins sets of stimuli can be defined; one new set of stimuli is assigned to the primary inputs during each clock period.

So far, the following event types have been introduced:

- *begin*-event: marks a beginning ramp,
- *end*-event: marks an active ramp,
- *deleted* event: marks a ramp which has been deleted from the local event queue, the event is

still referenced within the global event queue.

For control purposes the following new event types have been introduced, which are only used within the global event queue:

- CHECK_CONV event: a power convergency check is done,
- CLK event: a new CHECK_CONV event is scheduled after a user defined time and new stimuli from a possible stimuli file are applied after a user defined time.

During two consecutive CLK events multiple transitions at circuit nodes (actually for all output pins) are counted to give information about the glitch and hazard statistics.

7.4 Library characterization

The simulation is based on the logic behaviour of the instanced library cells and their characterization with respect to power consumption, delays and glitches. For the used library cells HSPICE simulations are used for parameter extraction.

The characterization data is put into a characterization file. Besides the actual characterization data also the gate functionality (based on primitives) is described in this file.

For semiautomatic characterization OCHATO (OFFIS CHAracterization TOol) was implemented. The tool is capable to invoke the required HSPICE simulations and extract the required data from the simulation results. For simple combinational gates also the possible power trigger and delay paths are extracted from the SPICE netlists. The results are written into the characterization file, which is needed for simulating the circuit with GliPS.

8 Evaluation

In this chapter the proposed glitch model and its implementation into a simulation tool is evaluated in terms of power simulation accuracy and simulation performance.

So far all model evaluations were based on the analysis of glitches for a small testbench, which is build up of 3 gates in series (confer Chapter 5.2). The analysis focused on the accuracy of the proposed glitch model in terms of glitch peak voltage and the representation of the glitch waveform by virtual ramps.

In this chapter the implemented simulation algorithm is evaluated for complete benchmark circuits. The impact of using the advanced glitch model on the accuracy of the power estimation is analysed. As reference the following simulators were used:

- HSPICE (Version 93a) by META-Software, now owned by AVANT!: HSPICE is the industry standard for accuracy in circuit simulation and is used for sign-off by most of the world's IC foundries. HSPICE is used as reference for accuracy of power consumption and simulator performance.
- PowerMill (Version 5.1) by EPIC, now owned by SYNOPSYS: PowerMill is also a circuit level simulation tool. In contrast to HSPICE the transistor model is simplified and the transistor characteristics are stored in tables. Further algorithms like circuit partitioning are used to further speed up simulation.
- TPS (Toggle Power Simulator) by OFFIS: TPS calculates power from simple toggle-count informations, which are extracted from VERILOG-XL simulations [Joch97]. TPS takes output loads and precharacterized gate internal power consumption into account. Within the VERILOG-XL simulation an inertial delay model and a SDF (considering slope-effects) are used.

A wide range of accuracy and simulator performance is covered by these three simulators. The target for GliPS is to fit well into the matrix of these 3 simulators. It is supposed to be significantly faster than the transistor based simulators HSPICE and PowerMill. The accuracy should be as close as possible to these two simulators. TPS features high simulation performance,

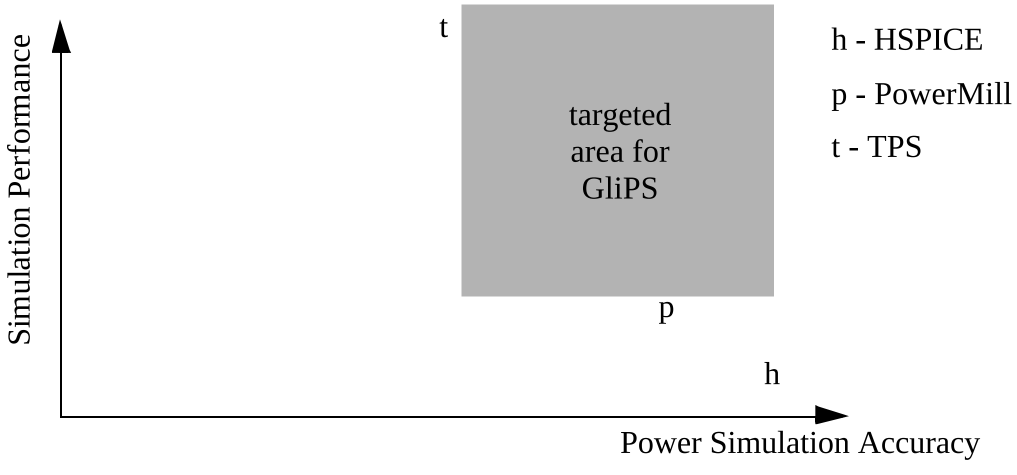


Figure 89:Simulator characteristics in terms of power simulation performance and power simulation accuracy.

which is paid by lower power simulation accuracy. GliPS is supposed to be more accurate than TPS. In Figure 89 this context is illustrated.

8.1 Practical results

The evaluation is based on the benchmark circuits given in Table 16.

library	module name	function	no. of simul. random pattern	no. of primary inputs	no. of prim. outputs	circuit depth[†]	no. of cells	layout-size [mm^2]
Design-Ware	ash	arithmetic shifter	1997	19	16	12	181	0.3
	mult	multiplier	1942	16	16	24	207	0.42
	sin	combinatorical sine	2000	8	8	25	196	0.36
ISCAS’85 bench-marks	c17	-	1056	5	2	4	6	0.008
	c499	ECAT	1000	41	22	13	202	0.38
	c1355	ECAT	1000	41	32	26	546	0.65
	c3540	ALU & contr.	500	50	22	48	1669	2.40
	c6288	16bit mult.	500	32	32	123	2406	3.21

Table 16:Used benchmark circuits for evaluation.

The first 3 benchmark circuits were generated from Synopsys’ DesignWare and contain complex gates like for example Full-Adders. The ISCAS’85 benchmarks consist of basic gates (like AND, OR, NAND, NOR, EXOR) only. The designs are mapped on Atmel ES2’s 1.0µm process and layout extracted data is available. Interconnects were modelled by single capacitors within HSpice and PowerMill.

For TPS delay calculation was done using the Cadence Delay Calculator (SDF enhanced wire delay model), which statically considers input-slope and output-load effects.

Within PowerMill the transistor-characterizations were run in advance (not included in the performance data) and two alternatives were distinguished:

- accurate mode: the following options were applied: `set_sim_spd 0.2` and `set_powr_acc 1`,
- default mode: no user defined options were used.

In Table 17 the achieved accuracies of charge-consumption and in Table 18 the simulator performances are reported with HSPICE as reference. It was not possible to simulate all pattern within one simulation run using HSPICE. As a consequence the simulations were split into several runs (each including initialization time).

module name	**Hspice**	**TPS**	**PowerMill**		**GLiPS**
	Q/Tr.		**accur.**	**def.**	
	[pC]	**ε_Q [%]**	**ε_Q [%]**	**ε_Q [%]**	**ε_Q [%]**
ash	72.7	24.6	13.7	26.3	10.7
mult	184.1	15.3	3.85	13.7	-0.23
sin	77.7	21.2	9.77	21.8	11.9
$\varnothing=\Sigma\|\varepsilon_Q\|$		20.4	9.11	20.6	7.61
c17	1.6	11.5	1.2	10.5	-6.7
c499	102	1.63	0.91	10.7	7.33
c1355	197	30.6	1.73	12.6	-0.97
c3540	1049	18.1	-6.89	0.66	2.79
c6288	3232	112	1.64	9.89	8.20
$\varnothing=\Sigma\|\varepsilon_Q\|$		34.8	2.47	8.87	5.20

Table 17:Accuracy of charge consumption.

module name	**HSpice time / pattern [s]**	**TPS**	**PowerMill**		**GliPS**
			accur.	**def.**	
		speed up	**speed up**	**speed up**	**speed up**
ash	148.78	45638	261	401	7051
mult	438.62	122863	332	533	13969
sin	116.08	42520	183	292	6036
c17	4.29	2258	223	263	3830
c499	170	31036	214	321	6540
c1355	283.2	32036	230	350	4240
c3450	3760	115984	743	1146	15732
c6288	9869	73108	360	535	3579
$\varnothing=\Sigma\|\varepsilon\|$		58180	318	480	7622

Table 18:Simulator Performance.

The TPS accuracies are better than 31% for all circuits except c6288, which has a large circuit depth[†] of 123 gates. The average speed-up of TPS is 58180. The low speed-up of the small

---

[†] circuit depth is referred to as the longest path from the primary inputs respectively Flip-Flop outputs to a gate (number of gates on the path are counted).

benchmark circuit c17 occurs, because the initialization of Verilog-XL takes a severe portion of the complete simulation time.

The PowerMill deviations are below 6.9% for the ISCAS'85 benchmarks (including c6288) in accurate mode. In average the accuracy is 2.47%. Using the default mode the accuracies for the ISCAS benchmarks are worse by a factor of 3.6 (average deviation is 8.87%). The inaccuracies of the DesignWare benchmarks raise up to 13.7% (26.3%) in accurate (default) mode. The speed-up is 318 (480) in average. I.e., the accuracy improvement of the accurate mode has to be paid by a loss of performance by 51%.

The GliPS deviations are below 12% for the DesignWare benchmarks and below 8.2% for the for the ISCAS'85 benchmarks. In average the accuracy is 7.6% (5.2%) for the DesignWare (ISCAS) benchmarks. Hence for the ISCAS benchmarks the accuracy of the GliPS results are between the results of the two used PowerMill modes. For the DesignWare benchmarks the results are even better than for both PowerMill modes. The speed-up over PowerMill is more than one order of magnitude. GliPS is approximately 16 times (24 times) faster than PowerMill for the accurate (default) mode. Comparing GliPS to TPS, the accuracy is significantly higher (7.6% versus 20.4% for the DesignWare benchmarks and 5.2% versus 34.8% for the ISCAS benchmarks). The gain in accuracy of GliPS over TPS is paid by a 7.6 times lower simulation performance.

As stated in the introduction (Chapter 1) the possible power savings for gate level optimizations are in the range of 20-30%. From these numbers a minimum accuracy of 5-10% was concluded, in order to be able to correctly choose between different design alternatives. This accuracy level is reached by GliPS, but not by TPS.

Two sources of error in accuracy can be distinguished:
- Errors in activity estimation and
- errors due to the power model.

In Table 19 the activity accuracy of the default PowerMill mode, TPS and GliPS is compared to the accurate PowerMill mode results. In PowerMill and GliPS all rising (falling) transitions, which cross the 30% (70%) V_{DD} voltage level, were counted. In TPS the logic threshold voltage is 50% V_{DD} for both types of transitions. I.e., if all simulators would perfectly model all transitions, the activity of TPS has to be lower than or equal to the activity values of GliPS respectively PowerMill, because the rising (falling) transitions crossing the 30% V_{DD} (70% V_{DD}) but not the 50% V_{DD} voltage level due to glitches would not be considered in TPS.

For TPS the characteristics of the used inertial delay model become obvious when comparing the activity results of the Designware multiplier with the ISCAS multiplier. Within the Designware multiplier a lot of complex multistage gates are used and within the ISCAS multiplier mainly nor2-gates, some inverter and a few AND2-gates are used. Hence the inertial delay model filters more events for the Designware multiplier than for the ISCAS multiplier, because the gate internal delay is larger than for single stage gates. The missing dynamic time shift of resetting transitions in TPS results in a pessimistic glitch filtering for the ISCAS multiplier. The inaccuracy increases with circuit depth. By accident the higher glitch filtering rate and the missing dynamic time shift of resetting transitions compensate each other for the DesignWare multiplier fairly good, so that the TPS activity results are surprisingly accurate (-1.25%). For other benchmark circuits (depending on circuit depth and the instantiated cells) these two effects do not compensate each other (refer to Table 19).

module name	**PowerMill**		**TPS**	**GliPS**
	accurate mode	**default mode**		
	net-activity	ε_{act} **[%]**	ε_{act} **[%]**	ε_{act} **[%]**
ash	0.524	0.953	15.28	-1.38
mult	0.646	1.44	-1.25	3.48
sin	0.450	0.942	13.3	-2.41
$\varnothing=\Sigma\lvert\varepsilon_{act}\rvert$		1.11	9.94	2.43
c17	0.481	0	3.72	-0.72
c499	0.514	1.37	12.7	-4.33
c1355	0.526	0.7	19.7	-9.23
c3540	0.638	1.23	17.5	0.58
c6288	2.597	3.15	105	1.58
$\varnothing=\Sigma\lvert\varepsilon_{act}\rvert$		1.29	31.7	3.28

Table 19:Accuracy of activity estimation.

The decrease in accuracy is dramatic especially for the circuits with high circuit-depth (c6288, c3540, c1355). Hence the main source of errors of TPS is the activity estimation inaccuracy. This observation is also documented in Figure 90 and Figure 91, where the activity inaccuracies are plotted as a function of circuit depth for the ISCAS multiplier. In Figure 90 the absolute activity is plotted for the accurate PowerMill, the GliPS and the TPS simulations (the decrease of activity for circuit-depth positions above 100 is due to the circuit structure). The relative activity accuracies per net are indicated by the dots in Figure 91. The deeper a net is located within the circuit, the higher the activity estimation errors are for TPS. The spread of the TPS activity results per net also indicates, that the maximum charge estimation error per gate is significantly higher than the above discussed average value. I.e. local cost functions for synthesis may lead to wrong optimization decisions. The activity-values of the GliPS simulation clearly show the gain in accuracy by the new glitch model. The spread of activity errors is also quite low. For GliPS over- and underestimation exist for different nets independent of circuit depth. I.e., no systematic errors (dynamic delay calculation of resetting ramps) like in TPS occur.

The high accuracy of GliPS simulation results in terms of activity also indicates a high accuracy of the delays themselves. The activity values can only be that accurate for the large number of different paths through the circuit to nodes, which are located deep in the circuit, if the delays are very accurate. Hence the accurate delay calculation is the major key for accurate activity values. Explicit delay values have not been analysed.

The above given accuracy data refer to simulations of a large set of random pattern. The error of charge consumption for single changes of input pattern is typically much higher. The error *may* average out if a large pattern sequence is analysed. In Table 20 the maximum deviation of charge consumption is given. The maximum error of TPS for a single change of input pattern is above 100%. The maximum error of GliPS are in the same range as the PowerMill results (default mode). The more complex gates which are used for the DesignWare modules, were modelled as black box components within GliPS and TPS. Figure 92 contains a plot with the number of pattern in a certain error interval for the ISCAS benchmark circuit c1355. The deviation in average and the variation of the data are significantly higher for TPS than for GliPS.

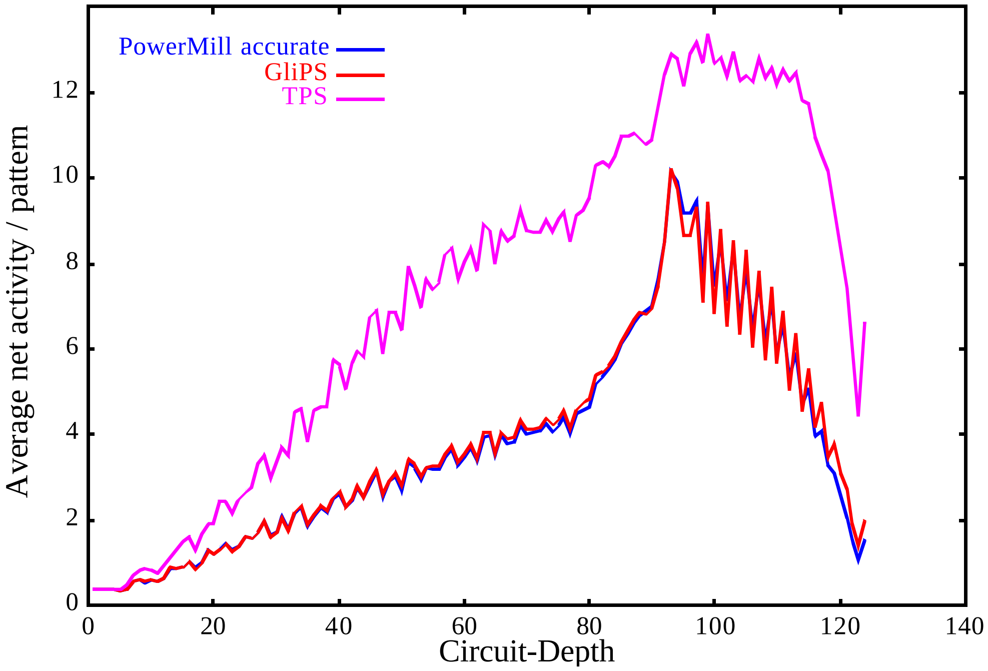


Figure 90:Net-activity as function of circuit depth position.

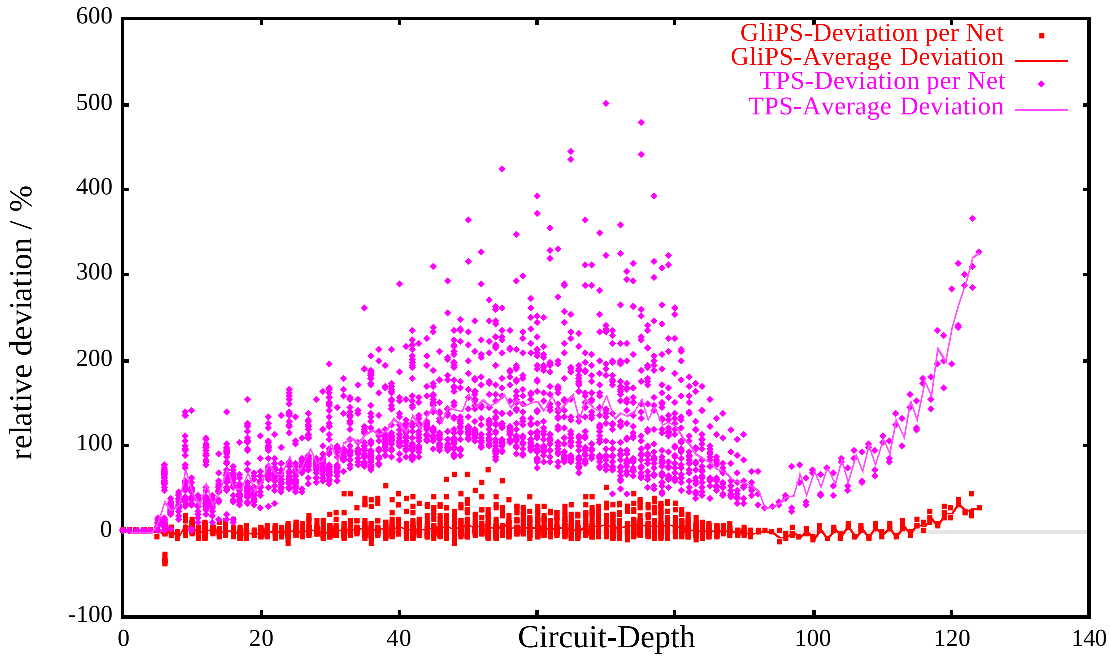


Figure 91:Relative accuracy of net-activity as function of circuit depth position.

RT-level power models may contain a large number of parameters, which need to be characterized using lower level simulators (i.e. gate- or transistor-level). Commonly only a subset of the complete set of input-pattern can be used to characterize a specific RT-level power model parameter, which is more error prone than the charge estimation of the whole pattern sequence.

module name	**PowerMill**		**TPS**	**GliPS**
	accurate	**default**		
ash	29	45	67	38
mult	41	34	69	65
sin	84	103	272	109
∅	51	61	136	71
c17	39	84	178	51
c499	23	37	54	62
c1355	24	43	84	54
c3450	17	22	56	24
c6288	15	19	147	27
∅	24	41	104	44

Table 20:Maximum deviation of charge consumption per pattern in %.

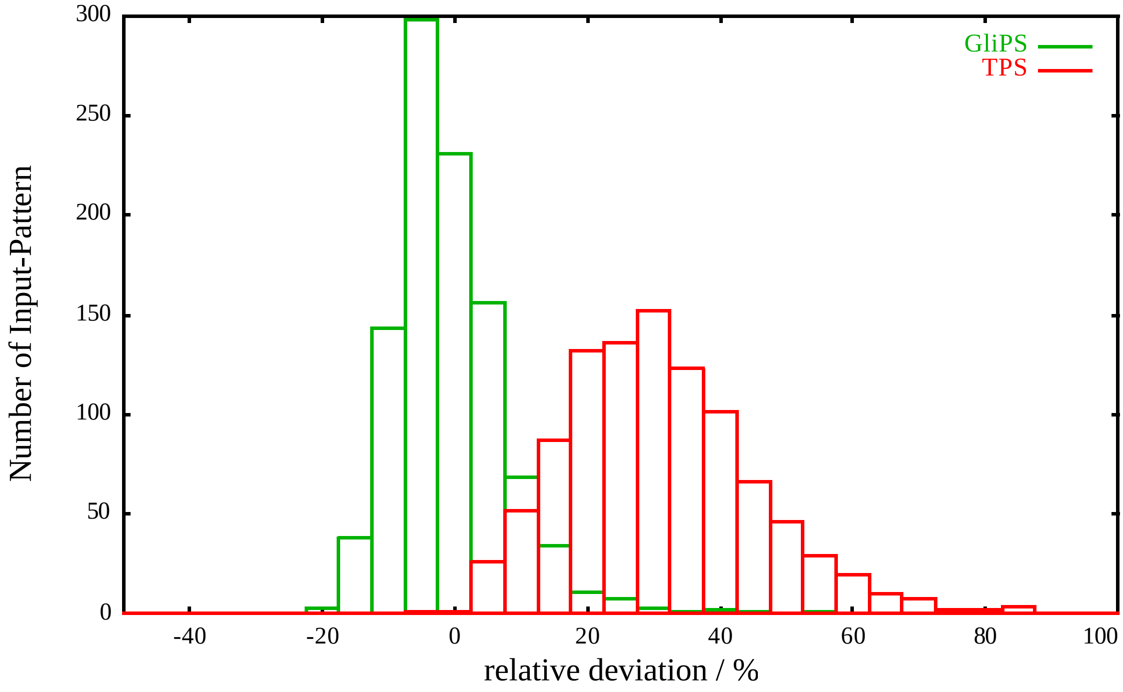


Figure 92:Power accuracy per pattern (c1355).

Within the common power formula, the dynamic power consumption depends linearly on the activity:

$$\begin{aligned} P_{SCi} &\sim \mathrm{a}_i, \text{ i refers to a specific output of a gate} \\ P_{Cap\_total} &= \frac{1}{2} \,\mathrm{x}\, V_{DD}^2 \,\mathrm{x}\, f \,\mathrm{x}\, \sum_i C_{fanout\_i} \cdot \mathrm{a}_{\boldsymbol{i}} \\ P_{dynamic} &= P_{SC} + P_{Cap\_total} \sim \mathrm{a} \end{aligned} \tag{43}$$

The actual charge consumption deviations do not completely match with the charge calculation numbers for the following reasons:

- The charge consumptions were compared to the HSpice results and the activity results to the PowerMill (accurate mode) results.
- The net activities need to be weighted with the actual characterized charge values for each power trigger before summing them up. The average net activities are not weighted by the characterized charge values.
- The analysed activity data do not correspond with the actual activity definition according to Definition 7 on page 5, because partial transitions are not counted with their fractional voltage change ΔV. This is a source of error for GliPS.
- Fanin capacities are no constant values as it has been explained in Chapter 3.5.1. This effects the TPS and GliPS results.
- For glitches the short circuit power consumption is scaled with the fractional voltage swing at a gate’s output by GliPS. This is not the actual physical behaviour (confer Chapter 3.5.3).
- For the calculation of situation dependent (input slope and output load) charge and delay values linear interpolation of precharacterized table entries are used. The interpolation error may lead to errors for GliPS and TPS.

The exact analysis of each possible further source of error is omitted here, because the activity has been identified as the major contributor for conventional logic simulators like TPS. A further improvement of the GliPS simulation algorithms is not absolutely necessary, because the target accuracy of 5-10% is achieved with the exception of 2 DesignWare modules. The results of the DesignWare modules, which use more complex gates, could be easily improved by splitting the multistage gates into single stage gates. Such a modification would transfer the simulated circuit topologies into a topology, which only contain single stage gates. Consequently the simulation results would improve to a similar level as the ISCAS benchmark circuits.

8.2 Conclusions

Within this chapter different power estimation methodologies at different levels of abstraction have been compared. The main important task is to find a good compromise between accuracy and simulation performance for the given constraints. As key point it was observed, that the activity estimation plays a major role. Simple toggle count based gate-level simulators (like TPS) deliver acceptable accuracy (in terms of power and activity) only for circuits with small logical depth. For large and moderate logical depth circuits the delay needs to be modelled more accurately. This is possible using the new gate-level power estimation tool GliPS, which is based on the proposed enhanced glitch model. Its accuracy is comparable to transistor level simulators running more than one order of magnitude faster.

9 Summary

In the introduction and the trend analysis it has been pointed out, that a circuit's power consumption is an important issue for today's integrated circuits, which will become even more important in the future. A circuit's power consumption can be optimized during the design process by considering the power consumption within a cost function, which has to be minimized. The cost function itself relies on accurate power estimates. As the possible power gains depend on the actual circuit design phase, each level of abstraction requires different accuracy margins for the power estimate. In this thesis gate level power calculation is focused on, for which a target accuracy margin of 5-10% needs to be met, in order to be able to make the right optimization decisions.

Different sources of errors for gate level power estimations have been identified. As the power consumption of a circuit depends on its stimulation, simulation based approaches need to be applied to obtain the needed activity numbers. Different approaches exist to (partly) solve the pattern complexity problem (confer Chapter 4.1). The most accurate method is statistical simulation, which is based on logical simulation.

Basic sources for inaccuracies on gate level are identified within this thesis (confer Chapter 3). Besides the accurate power assignment to complete transitions, accuracy can be significantly improved by correctly considering glitches and hazards within the power formula and the simulation algorithm. Conventional simulation algorithms - like the transport and inertial delay model - do not consider accurate glitch handling, which leads to activity overestimation for circuits with medium to large circuit depths. This lack can be overcome with the presented enhanced glitch model.

The glitch model is derived from basic physical CMOS characteristics. Its accuracy and robustness has been exemplified by comparing it to other existing approaches (Chapter 4 and 5) for a small benchmark circuit with 3 gates in series. The integration of the new glitch model into an appropriate simulation algorithm and its implementation into a simulator (named GliPS) is described in Chapter 7. Additional characterization data, which are needed for the glitch model, are automatically generated from circuit level simulations by the tool OCHATO.

The efficiency in terms of accuracy and simulation performance of the new model has been exemplified for common benchmark circuits. The results show, that this approach closes the gap between accurate circuit level and conventional gate level simulation tools. The simulation accuracy of GliPS is in the same range as the results of the circuit level simulator PowerMill (default mode, refer to Chapter 8), featuring more than one order of magnitude speed up. The simulation results of GliPS are 3-7 times more accurate than the results of TPS, which is based on a conventional inertial delay model. The simulation performance of GliPS is less than one order of magnitude below TPS.

10 References

[Alid94] Alidina, M.; Monteiro, J.; Devadas, S.; Ghosh, A.; Papaefthymiou, M.. *Precomputation-Based Sequential Logic Optimization for Low Power*. IEEE Transactions on VLSI Systems, Vol. 2, No. 4. pp. 426-436. 1994

[Bako90] Bakoglu, H.B.. *Circuits, Interconnections, and packaging for VLSI.* Addison-Wesley. 1990

[Beni96] Benini, L.; Bogliolo, A.; Favalli, M.; De Micheli, G.. Regression models for behavioural power estimation. PATMOS. pp. 179-187. 1996

[Bogl98] Bogliolo, A.; Benini, L.. *Robust RTL Power Macromodels*. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. Volume 6. pp. 578-581. December 1998

[Brgl89] Berglez,F.; Bryan, D.; Kozminzki, K.. Combinational profiles of Sequential Benchmark Circuits. Proceedings IEEE International Symposium on Circuits and Systems. pp. 1929-1934. 1989

[Bohr95] Bohr, M.T.. *Interconnect Scaling - The Real Limiter to High Performance ULSI.* Technical digest of International Electron Devices Meeting. 1995. pp. 10.1.1-10.1.4

[Bohr98] Bohr, M.T.. *Silicon Trends and Limits for Advanced Microprocessors*. Communications of the ACM. pp. 80-87. March 1998

[Burc88] Burch, R.; Najm, F.; Yang, P.; Hocevar, D.. *Pattern-Independent Current Estimation for Reliability Analysis of CMOS Circuits*. Proceedings of ACM/IEEE 25th Design Automation Conference. pp. 294-299. 1988

[Burc93] Burch, R.; Najm, F.; Yang, P.; Trick, T.N.. *A Monte Carlo Approach for Power Estimation*. IEEE transactions on very large scale integration systems, Vol. 1, No. 1. pp. 63-71. 1993

[Cade97] Cadence Design Systems, Inc.. *Verilog-XL Reference Manual*. 1997

[Chat93] Chatterjee, P.K.; Larrabee, G.B.. *Gigabit Age Microelectronics and Their Manufacture*. IEEE transactions on very large scale integration systems, Vol. 1, No. 1. pp. 7-21. 1993

[Chan92] Chandrakasan, A.P.; Sheng, S.; Brodersen, R.W.. *Low-Power CMOS Digital Design*. IEEE Journal of Solid-State Circuits, Vol. 27, No. 4. pp. 473-484. 1992

[Cha295] Chandrakasan, A.P.; Brodersen, R.W.. *Minimizing Power Consumption in Digital CMOS Circuits*. Proceedings of the IEEE, Vol. 83, No. 4. pp. 498-523. 1995

[Cha395] Chandrakasan, A.P.; Potkonjak, M.; Mehra, R.; Rabaey, J.; Brodersen, R.W.. *Optimizing Power Using Transformations*. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 14, No. 1. Jan. 1995. pp. 12-31

[Chan95] Chandrakasan, A.P.; Brodersen, R.W.. *Low Power Digital CMOS Design*. Kluwer Academic Publishers. 1995

[Cong94] Cong, J.; Koh, C.K.. *Simultaneous Driver and Wire Sizing for Performance and Power Optimization*. IEEE Transactions on VLSI Systems. pp. 408-425. 1994

[D&T95] *A D&T Roundtable: Low-Power Design*. IEEE Design & Test of Computers, pp. 84-90. Winter 1995

[Dava95] Davari, B.; Dennard, R.H.; Shahidi, G.G.. *CMOS Scaling for High Performance and Low Power - The Next Ten Years*. Proceedings of the IEEE, Vol. 83. pp. 595-606. April 1995

[Deng94] Deng, A.-C.; *Power Analysis For CMOS/BiCMOS Circuits*; International Workshop on Low Power Design. pp. 3-8. 1994

[Eise95] Eisele, M.; Berthold, J.. *Dynamic Gate Delay Modelling for Accurate Estimation of Glitch Power at Logic Level*. PATMOS. pp. 190-201. 1995

[ES2_07] ES2 Data Sheets 0.7µm

[ES2_10] ES2 Data Sheets 1.0µm

[Figu94] Ortega, M.A.; Figueras, J.. *Extra Power Consumed in Static CMOS Circuits due to Unnecessary Logic Transitions*. Proceedings of IV International Design Automation Workshop, Russia. pp. 40-42. 1994

[Flet94] Fletcher, T.D.. *Microprocessor Technology Trends.* Proceedings of the IEEE International Electron Devices Meeting (IEDM), pp. 10.1.1-10.1.3. 1994

[Geor94] George, B.J.; Gossain, D.; Tyler, S.C.; Wloka, M.G.; Yeap, G.K.H.. *Power analysis and characterization for semi-custom design*. International Workshop on Low Power Design. pp. 215-218. 1994

[Ghos92] Ghosh, A.; Devadas, S.; Keutzer, K.; White, J.. *Estimation of Average Switching Activity in Combinational and Sequential Circuits*. Proceedings of the 29th Design Automation Conference. pp. 253-59. 1992

[Hede87] Hedenstierna, N.; Jeppson, K.O.. *CMOS Circuit Speed and Buffer Optimization*. IEEE Transactions on Computer-Aided Design. Vol. CAD-6. NO. 2. pp. 270-281. 1987

[Huiz90] Huizer, C.M.. *Power Dissipation Analysis of CMOS VLSI Circuits by Means of Switch-Level Simulation*. Proceedings European Solid State Circuits Conference .pp. 61-64. 1990

[IEEE87] IEEE. *IEEE Standard VHDL Language Reference Manual*. IEEE Std 1076-1987

[Inte98] Intel Corporation. *Processor - Hall of Fame - Technical Specifications*. WWW page http://www.intel.com/intel/museum/25anniv/Hof/tspecs.htm

[Joch97] Jochens, G.; Kruse, L.; Nebel, W.. *Application of Toggle-Based Power Estimation to Module Characterization.* Proceedings of PATMOS. pp. 161-170. 1997

[Land93] Landman, P.E.; Rabaey, J.M.. *Power Estimation for High Level Synthesis*. Proceedings of EDAC'1993. pp. 361-366

[Land95] Landman, P.E.; Rabaey, J.M.. *Black-Box Capacitance Models for Architectural Power Analysis*. International Workshop on Low Power Design'95. pp. 93-98

[Lebl93] Leblebici, Y.; Kang, S.-M.. *Hot-Carrier Reliability of MOS VLSI Circuits*. Kluwer Academic Publishers. 1993

[Lehm95] Lehmann, G.; Nagel, P.; Müller-Glaser, K.D.. *An Approach to Detailed Modelling of Digital CMOS Gates for Logic Simulation Using VHDL*. International Journal of Electronics and Communications. pp. 81-90. 1995

[Mart97] Martens, A.. *Entwurf und Implementierung eines Verluststromanalysewerkzeugs*. Master-Thesis, Carl von Ossietzky University of Oldenburg, Germany. 1997

[Masa92] Masaki, A.. *Deep-Submicron warms up to High Speed Logic*. IEEE Circuits and Devices Magazine. pp. 18-24. November 1992

[Mehr94] Mehra, R.; Rabaey, J.. *Behavioral Level Power Estimation and Exploration*. IEEE International Workshop on Low Power Design. pp. 197-202. 1994

[Melc91] Melcher, E.; Dana, M.; Jutand, F.. *PATMOS: Report on Cell Assembly and Multi-Level Modelling*. BRA No 3237 Second Periodic Progress Report Appendix 1. pp. 1-21. December 1991

[Metr95] Metra, C.; Favalli, M.; Riccò, B.. *Glitch Power Dissipation Model.* PATMOS. pp. 175-189. 1995

[Moor97] Moore, G.. *The Continuing Silicon Technology Evolution Inside the PC Platform*. Intel Platform Solutions On-line News for Developers. October 1997. (http://developer.intel.com/solutions/archive/issue2/feature.htm)

[Najm89] Najm, F.. *Probabilistic Simulation For Reliability, Analysis of VLSI Circuits*. PHD Thesis at the University of Illinois at Urbana-Chapaign. 1989

[Najm91] Najm, F.. *Transition Density, a Stochastic Measure of Activity in Digital Circuits*. Proceedings of the 28th Design Automation Conference. pp. 439-450. 1991

[Nebe97] Nebel, W.; Mermet, J.. *Low Power Design in Deep Submicron Electronics*. Kluwer Academic Publishers. 1997

[Powe92] Powell, S.R.; Chau, P.M.. *Power Dissipation of VLSI Array Processing Systems.* Journal of VLSI Processing. Vol. 4. pp. 199-212. 1992

[Powe95] Powers, R.A.. *Batteries for Low Power Electronics*. Proceedings of the IEEE, Vol. 83. pp. 687-693. April 1995

[Rade96] Radetzki, M.; Timmermann, B.; Rabe, D.; Nebel, W.. *Generation of Binary Patterns with Given Spatiotemporal Correlations*. PATMOS. pp. 199-208. 1996

[Roy94] Roy, K.; Prasad, S.. *Logic Synthesis for Reliability - An Early Start to Controlling Electromigration and Hot Carrier Effects*. Proceedings of European Design Automation Conference. pp. 136-141. 1994

[Saxe97] Saxena, V.; Najm, F.N.; Hajj, I.N.. *Monte-Carlo Approach for Power Estimation in Sequential Circuits*. European Design and Test Conference. pp. 416-420. 1997

[Schn96] Schneider, P.H.; Krishnamoorthy, S.. *Effects of Correlations on Accuracy of Power Analysis - An Experimental Study*. International Symposium on Low Power Electronics and Design. pp. 113-116. 1996

[Sema94] SIA Semiconductor Industry Association. *The National Technology Roadmap for Semiconductors - Technology Needs*. 1994

[Sema97] SIA Semiconductor Industry Association. *The National Technology Roadmap for Semiconductors - Technology Needs*. 1997

[Sing95] Singh, D.; Rabaey, J.M.; Pedram, M.; Catthoor, F.; Rajgopal, N.; Sehgal, N.; Mozden, T.J.. *Power Conscious CAD Tools and Methodologies: A Perspective*. Proceedings of the IEEE, Vol. 83, No. 4. pp. 570-594. 1995

[Sven94] Svensson, C.; Liu, D.. *A Power Estimation Tool and Prospects of Power Savings in CMOS VLSI Chips*. International Workshop on Low Power Design. pp. 171-176. 1994

[Tiwa93] Tiwari, V.; Ashar, P.; Malik, S.. *Technology Mapping for low Power*. Proceedings of 30th Design Automation Conference. pp. 74-79. 1993

[VanO93] Van Oostende, P.; Six, P.; Vandewalle, J.; DeMan, H.. *Estimation of typical power of synchronous CMOS circuits using a hierarchy of simulators*. IEEE Journal of Solid-State Circuits. Vol. 28. pp. 26-39. 1993

[VDE96] GMM VDE/VDI-Gesellschaft Mikroelektronik, Mikro- und Feinwerktechnik, Fachausschuß Trendanalyse. *Trends der Mikroelektronik und ihrer Anwendungen 1995-2000*. 1996

[Veen84] Veendrick, H.. *Short-Circuit Dissipation of Static CMOS Circuitry and Its Impact on the Design of Buffer Circuits*. IEEE Journal of Solid-State Circuits. pp. 468-473. August 1984

[Veen98] Veendrick, H.. *Digital Goes Analog.* Proceedings of the 24th European Solid-State Circuits Conference. pp. 44-50. September 1998

[Vöge97] Vögel, G.. *Erweiterung eines Werkzeugs zur Charakterisierung von CMOS-Standardzellenbibliotheken bezüglich Verluststromdaten*. Studienarbeit, Carl von Ossietzky University of Oldenburg, Germany. 1997

[Vöge98] Vögel, G.. *Entwurf und Implementierung von Verfahren zur detaillierten Verlustleistungsanalyse auf Gatterebene*. Master-Thesis, Carl von Ossietzky University of Oldenburg, Germany. 1998

[West93] Weste, N.H.E.; Eshraghian, K.. *Principles of CMOS VLSI Design - A Systems Perspective*. Second Edition. Addison/Wesley Publishing Company. pp. 175-257. 1993

[Wu98] Wu, Q.; Qiu, Q.; Pedram, M.; Ding, C.S.. *Cycle-Accurate Macro-Models for RT-Level Power Analysis*. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. Volume 6. pp. 520-528. December 1998

[Wund97] Wunder, B.. *Ein neues Konzept zur Modellierung und Timing-Simulation von VLSI Systemen*. PhD-Thesis, Institut für Technik der Informationsverarbeitung (ITIV), Universität Karlsruhe. 1997

[Wuyt94] Wuytack, S.; Catthoor, F.; Franssen, F.; Nachtergaele, L.;DeMan, H.. *Global communication and memory optimzing transformations for low power systems*. International Workshop on Low Power Design. pp. 203-208. 1994

11 Glossary

11.1 Terms

Abbreviation / Term	**Definition**
activity	refer to net activity
Boolean difference	The Boolean difference defines the condition for f(x) to be sensitive on a change of input x_i: $\frac{\partial f(\underline{x})}{\partial x_i} = f(\underline{x})\|_{x_i = 0} \oplus f(\underline{x})\|_{x_i = 1}$
CE scaling	constant electrical field scaling: the electrical field in the gate isolation is kept constant
circuit complexity	number of transistors per chip
complete transition	refer to partial transition
controlling input signal	If an input signal $x_i(t)$ at time t determines a function’s output signal independently from other input signals, it controls the output signal (controlling input signal). If this property is not fulfilled, $x_i(t)$ is a non-controlling input signal at time t. Example NAND gate: if a signal $x_i(t)$ at time t is logically 0, the output signal $x_y(t)$ is logically 1 independent from any other input. I.e., the input signal controls the output at time t.
die	part of a wafer, which is used for a single chip
distributed delay	a delay is assigned to each of the components, of which a module is built up; in contrast to distributed delays module path delays may be used [Cade97]
DRAM	dynamic read access memory
edge	synonym for a transition
event	An event is a change between two states, which belong to a well defined set of signal states. In addition to voltage level dependent state definitions, driving strengths are commonly also considered. Examples for event definitions are: • an arbitrary change of voltage level is defined as an event within EPIC’s PowerMill tools, • a change of logic states {U,X,0,1,Z,W,L,H,-} (IEEE 1164).
GliPS	Glitch Power Simulator: The proposed enhanced glitch modelling algorithm is implemented into this stand alone simulator.

Abbreviation / Term	Definition
glitch	A **glitch** consists of a pair of at least two partial signal transitions. Three or more consecutive partial transitions, which do neither reach VDD nor VSS in between, define a **dynamic glitch**.
hazard	A pair of unnecessary complete transitions within one computational cycle $[t_0,t_e]$ is defined as a **hazard**. Three or more consecutive complete transitions define a **dynamic hazard**.
incomplete transition	refer to partial transition
colliding input signals (transitions)	refer to signal propagation collision
low-level effects	effects which can only be exactly determined late in the design flow (at low levels of abstraction); such effects typically have a significant effect on a circuit's power consumption and performance
module path delay	delays are assigned to different paths through a module; the delays may be conditional; in contrast to module path delays distributed delays may be used [Cade97]
monotonous gates	the direction of a potential output event is uniquely determined by the direction of the causing input event for a monotonous gate; structurally each input is connected to one NMOS- and one PMOS-transistor, which both belong to the same CMOS stage
MPU	Microprocessor Unit
net activity	The net activity α of a signal s is the average number of transitions per clock cycle (typically equivalent to computational cycle). Partial transitions are considered fractionally according to their voltage swing ΔV_s: $\alpha_s = \frac{1}{V_{DD} \cdot f} \cdot \lim_{\tau \to \infty} \frac{\sum_{\text{Transitions during the period } \tau} \Delta V_s}{\tau}$
partial transition	If a signal's voltage is monotonously changing from V_{DD} to V_{SS} or vice versa, a **complete** transition has occurred. In all other cases an **incomplete** respectively **partial** transition has occurred[‡]. The potentials V_{DD} and V_{SS} are typically given by the driving gate's supply voltage.
resetting transition	refer to setting transition / resetting transition

Abbreviation / Term	**Definition**
setting transition / resetting transition	In case of a glitch generation or propagation, the setting input transition causes the first output transition and the resetting input transition causes the second output transition. The two output transitions have opposite directions.
signal	object with a past history of values [IEEE87]; the values may be digital or continuous (within CMOS circuit typically voltages)
signal propagation collision	If two or more changing input signals impact a change of the output voltage waveform at the same time, the input signals collide while propagating through the cell.

Abbreviation / Term	Definition
slope	The slope of a transition describes its steepness. The slope is typically the time interval between the instant when a signal's voltage crosses 10%V_{DD} and 90%V_{DD}. Sometimes the term slope is also used as a synonym for a transition.
TPS	TPS calculates power from simple toggle-count informations, which are extracted from VERILOG-XL simulations [Joch97]. TPS takes output loads and precharacterized gate internal power consumption into account. Within the VERILOG-XL simulation an inertial delay model and a SDF (considering slope-effects) are used.
transaction	A transaction is an assignment of a state to a signal, which belongs to a well defined set of signal states. While events only consider a change of a state, a transaction can also be an assignment of the same state. I.e., events are a subtype of transactions.
transition	A **transition** T describes the process of a monotonously changing signal s. I.e., rising and falling transitions are distinguished[‡]. The changing signal is typically represented by a voltage in the domain of integrated CMOS circuits. Hence formally either one of the following two properties need to be fulfilled for a transition: $\frac{dV}{dt} \geq 0 \qquad or \qquad \frac{dV}{dt} \leq 0$ A voltage range is typically associated with a logic value (e.g. 0,1,X).
useful/useless transition	If an odd number of signal transitions occurs within one computational cycle $[t_0,t_e]$ ($\lvert V_s(t_0)-V_s(t_e)\rvert=V_{DD}$), one **useful** transition has occurred within this period. All additional transitions are **useless**. If an even number of signal transitions occurs within one computational cycle ($V_s(t_0)=V_s(t_e)$), all transitions within this period are **useless**.

[‡] over- and undershots are neglected here

11.2 Expressions

Abbreviation	**meaning**
A	area, typically transistor area is referred A_{tr}
A_{tr}	area occupied by one transistor
α	average switching activity per clock cycle
C_{eff}	effective switched capacitance per clock cycle ($\sum_i C_i \cdot \alpha_i$)
C_g	total capacitance per gate
C_{fanin}	effective capacitance of a gate's input pin; two definitions are distinguished: • for delay characterization: an explicit load capacitance, which results - if connected to an arbitrary driving gate's output - in the same delay, • for power consumption: actual charge flowing through the input while switching divided by the supply voltage. Within a test circuit the term fanin capacitance is also used as the fanout load of the driving gate: the sum of the intrinsic fanin capacitance (taken from the data sheets) and an explicit capacitor between the DUT's input node and V_{SS}
$C_{faninReset}$ / $C_{faninReset}$	fanin capacitance of the pin to which the setting / resetting input transition is applied to generate a glitch at the gate's output
C_{fanout}	capacitive load, which is connected to a gate's output; the fanout capacitance typically consists of interconnection and fanin capacitances of consecutive gates
C_{int}	total capacitance for an interconnection
$C_{intglobal}$	total capacitance for a global (i.e. long) interconnection
$C_{intlocal}$	total capacitance for a local (i.e. short) interconnection
C_{max}	due to delay and/or slope constraints for each gate a maximum fanout capacitance is defined
E	electrical field (e.g. gate-channel area or drain-source)
E_g	electrical field in the gate-oxide
En	Energy
e	scaling factor for electrical field in gate-oxide
f	clock frequency of a synchronous circuit or part of it, if multiple clocks are used on a chip
I_{DS}	drain-source current

Abbreviation	**meaning**
l	transistor length
λ	minimum feature size of a technology
l_{int}	interconnection length
$l_{intglobal}$	length of global interconnection
$l_{intlocal}$	length of local interconnection
P	Power Consumption of a circuit or part of it
P_g	Power Consumption of a single gate (i.e. cell)
Q	electrical charge ($Q = \int I dt$)
Q_{Cap}	charge for charging the fanout capacitor of a gate
$Q_{CapIntern}$	gate internal capacitors are charged
Q_{SC}	short circuit charge through a gate, which flowing while the input voltage switches
$R_{intglobal}$	resistance of global interconnection
$R_{intlocal}$	resistance of local interconnection
R_{tr}	on-resistance of a MOS transistor
S	scaling factor for feature sizes
S_C	scaling factor of die edges (i.e. die area is scaled by S_C^2), which allows an economical IC production
S_{TH}	voltage, which is required to drop the subthreshold current by one decade
T	period time ($=1/f$)
J	Temperature typically in Kelvin
t_{end_setOut}	instant, when the linear approximated ramp ends
τ_g	gate delay
t_{glitch}	time when the glitch peak is reached
τ_{HL}	propagation delay through an element with a falling slope at it's output
$\tau_{intglobal}$	delay associated with global interconnection
τ_{LH}	propagation delay through an element with a rising slope at it's output
t_{ox}	thickness of gate oxide

Abbreviation	**meaning**
t_{start_setOut}	instant, when the linear approximated ramp starts
V_{DD}	supply voltage, provided by an external voltage source (i.e. battery or permanent supply)
V_H	logic threshold voltage for delay characterizations of falling edges; in this thesis the Boolean value ’1’ is associated with voltages above $\mathrm{V_H}$
$V_{inReset}(t)$	glitch causing input voltage waveform of the resetting input transition
$V_{inSet}(t)$	glitch causing input voltage waveform of the setting input transition
V_L	logic threshold voltage for delay characterizations of rising edges; in this thesis the Boolean value ’0’ is associated with voltages below $\mathrm{V_L}$
V_{Peak} / ΔV_{peak}	absolute voltage of the glitching waveform at t=t_{glitch}; ΔV_{peak} is the absolute voltage change with relation to its initial value (immediately before the setting output waveform) and V_{peak}
V_T	threshold voltage of a MOS transistor, $\mathrm{V_{TN}}$ respectively $\mathrm{V_{TP}}$ refer to NMOS and PMOS transistors
V_{TN}	threshold voltage of a NMOS transistor
V_{TP}	threshold voltage of a PMOS transistor
W	transistor width
W_{int}	interconnection width

Appendix A: Power Gain Budget

The potential impact of design decisions on power consumption at different levels of abstraction is given in table21. The table contains the estimated impact of some well known experts in the low power domain. Even though the absolute numbers of potential power savings diverge among the different experts, it is obvious, that design decisions at high levels of abstraction have a greater impact on power consumption than design decisions on low levels (similar to other constraints like area and circuit performance).

	K. Keutzer[‡]	**P. Landman**[‡‡]	**L. Gal**[‡‡‡]	**U. Ko**[‡‡‡‡]
Layout		< 20%		
Circuit Level	10%		< 2x	30%
Gate/Logic Level	15-20%	15-50%		40-50%
Architectural Level	50%	10-90%		5-10x
System Level			> 10x	
Algorithm / Software	4x	10-100x	> 10x	
CAD			10x	

Table 21:Power Gain Budget[‡‡‡‡‡]

[‡] Synopsys, Inc.

[‡‡] Texas Instruments

[‡‡‡] Motorola

[‡‡‡‡] manager of the Low-Power Center of Excellence in Texas Instruments’ Application Specific Products, data taken from [D&T95]

[‡‡‡‡‡] The data (except that of Uming Ko) was presented in the panel entitled „*Which Has Greater Potential Power Impact: High-Level Design and Algorithms or Innovative Low Power Technology?*“ at *1996 International Symposium on Low Power Electronics and Design*; Monterey, California

Appendix B: Personal Record

Name:	Dirk Rabe
Birth:	June, 22nd 1968 in Bremen
Nationality:	German
Family Status:	married since September, 24th 1993

August 1974 until July 1978:	Grundschule Barrien
August 1978 until July 1980:	Orientierungsstufe Syke
August 1980 until June 1987:	Gymnasium Syke

July 1987 until September 1988:	Military service at the technical army school in Eschweiler and at the anti aircraft defence regiment in Achim

October 1988 until May1993:	Study of Electronics at the University of Bremen
May, 27th 1993:	Graduation with degree Dipl.-Ing. (Univ.)
June 1993 until May 1998:	PHD student and scientific assistant at the Carl von Ossietzky University of Oldenburg
since July 1998:	Electronic Designer at the Chipcard Division of Siemens Semiconductors, which has been spin off to Infineon Technologies in 1999