CARL VON OSSIETZKY universität OLDENBURG

Fakultät II – Informatik, Wirtschafts- und Rechtswissenschaften
Department für Informatik

RT-Level Power-Gating Models optimizing Dynamic Leakage-Management

Dissertation zur Erlangung des Grades eines
Doktors der Ingenieurwissenschaften

von

MSc Sven Rosinger

Gutachter:

Prof. Dr.-Ing. Wolfgang Nebel
Prof. Dr.-Ing. Jürgen Teich

Tag der Disputation: 27.09.2012

Abstract

Power-gating is the most promising run-time technique in order to reduce leakage currents in sub-$100nm$ CMOS devices but its application is associated with numerous problems. Overhead costs in terms of additional state transition costs occur, the targeted circuit is slowed down while being in the active state, additional interfacing circuits are necessary, and in general the total impact of the power-gating technique is hard to predict at early design stages.

The goal of this thesis is to develop power-gating models for functional units at RT-level to enable design tradeoffs and to optimize the high-level synthesis for the use of this design technique.

Main contributions of this work are

- Fast and accurate power-gating models for an estimation of the functional unit's energy demand during the static active and sleep state as well as during a state transition,
- Optimized scheduling, binding, and allocation approaches that are able to increase the profitability of a cycle-wise power-gating and to expand the design-space exploration, and
- A consistent design flow of the high-level synthesis decisions to subsequent design tools.

In this thesis, such an estimation and optimization framework is proposed. The models are characterized by circuit-level simulations and have been evaluated to lead to maximum errors of 15.7% at a standard deviation of 3.41%. They are used to estimate the energy reduction of functional RT-level components to be 46% in average. The optimized synthesis approaches can even further reduce the remaining energy demand by up to 43% at an average reduction of 19.8%.

Contents

1 Introduction

Embedded systems' complexity has risen dramatically due to the unbroken need for computing power. At the same time, non-ideal scaling of device parameters and supply/threshold voltages, as well as manufacturing variability have become major silicon complexity concerns for the design of an application specific integrated circuit (ASIC). The situation is predicted to get even worse as main parameters such as the ASIC's power consumption and its thermal heat will limit the previous progress of shrinking the area and increasing the number of transistor devices [ITR10]. The sector of mobile devices, predicted to have 7 trillion mobile devices connecting 7 billion people in 2017 [Dav08], is especially hard hit by this development.

To tackle this super-exponential increase of complexity, crosscutting challenges are needed to be addressed. One of these challenges is *dynamic power-management (DPM)*. DPM directly counteracts the power issue, it can be applied at different levels of abstraction during design, and even strengthens future re-use of circuit parts.

In the following, the ideas of DPM for low leakage design and the design at system-level are described and motivated in terms of its impact for this thesis. Afterwards the contribution of this thesis is worked out and summarized. The thesis outline concludes this introduction.

1.1 Low Leakage Design and Dynamic Power-Management

Power as a design criterion has become more and more relevant already in the nineties. At the beginning, approaches only optimized for dynamic power dissipated due to capacitive charge and short-circuit currents. In today's digital integrated circuits leakage currents are responsible for the dominating part of the total energy consumption. Especially in low utilized parts of the circuit, energy is wasted as a result of leakage currents. As a consequence, leakage has become the primary optimization goal.

Figure 1.1 presents a full-chip leakage power foresight of the international technology roadmap for semiconductors (ITRS) working group. Their leakage prediction is based on detailed parameter estimates done in [ITR10] and is normalized to the values of 2007. In comparison with the past years, the short-term foresight predicts a slackening increase of leakage currents. This is mainly due to development of high-k dielectric materials used in semiconductor manufacturing processes limiting gate-leakage. But for the long-term, other leakage sources will gain in importance and again considerable increases in leakage currents are predicted.

In the actual ITRS 2009 Design Edition and its 2010 update [ITR10], the working group appoints leakage to a *control requirement*. Further, dynamic power-management is considered as one of the cross-cutting challenges between silicon- and system-complexity concerns. However, *DPM* is more a collective term describing techniques at various levels of abstraction. All of them have in common to aim at

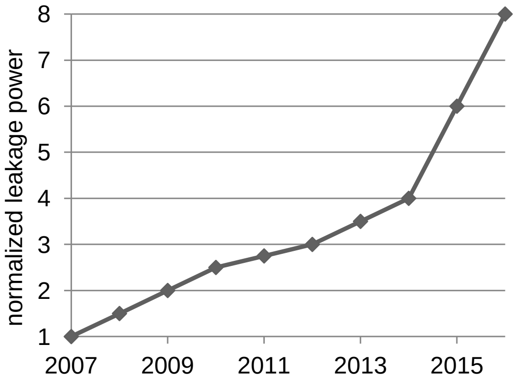


Figure 1.1: Full-chip leakage foresight of ITRS [ITR10] (normalized to 2007)

dynamic-power or leakage-power savings. In contrast to improvements of the semiconductor manufacturing process and the use of new materials these DPM techniques are integrated during design, and power savings are gained during runtime by setting circuit parts into power saving states.

1.2 System Level Design and the Need for High-Level Synthesis

At system-level a designer has greatest possible degrees of freedom in terms of the design optimization. Functional tasks have to be partitioned to either hardware or software and adequate structural objects such as processor cores, memories, busses, and intellectual property (IP)-components have to be selected or designed for their execution. A special focus at system-level is on reuse of legacy IP-components to ease the development and to reduce costs. If not available or not compliant with constraints on quality, execution time, or power, IP-components have to be created from scratch by synthesis approaches.

Power constraints at system-level that may be derived from requirements on battery run-time rise the need for application of DPM techniques. Questions thereby are often related to the granularity of its appliance and effectiveness. Additionally, runtime-effects that are induced by the technique and the influence of parameters such as surrounding temperature are of concern. Thereby, two main problems exist for that no manufacturable solutions are available [ITR10]. The first problem is to accurately analyze and estimate power-management techniques budgeting area, power, and timing. Secondly, methods and tools for an automated insertion of power-management structures are missing. Both issues are at odds with the reuse of IP-components that are not capable to power-manage and demand for PM-integrated IP-components. A supplementary insertion of internal PM-techniques is not possible in many cases and even an externally caused power-down may not be acceptable due to a state-loss of the component.

As a result, appropriate high-level synthesis (HLS) methods and tools to design DPM-aware IP components are needed. These tools have to comprise DPM-techniques directly during synthesis and have to offer PM-interfaces at system-level.

Table 1.1 shows a market survey on the most common available behavioral level synthesis tools. The

Vendor	Tool	I/O languages
Synopsys	SynphonyC Compiler	C, C++ / Verilog, VHDL
Forte	Cynthesizer	SystemC-TLM / Verilog
NEC	Behavioral Synthesizer	C / VHDL, Verilog
Bluespec	Bluespec Compiler	Sys. Verilog, SystemC / Verilog, SystemC
AutoESL	AutoPilot	C, C++, SystemC / Verilog, VHDL, SystemC
Mentor	Catapult C Synthesis	C, C++ / Verilog, VHDL
CebaTech	C-to-RTL-Compiler (C2R)	C / Verilog

Table 1.1: Market survey on behavioral synthesis tools

minority of the tools address leakage estimation. Further, none of the tools currently targets leakage optimization and is capable to implement dynamic power-management during synthesis. Thus all tools are candidates for the integration of the flow presented in this thesis.

1.3 Scope and Contributions of this Thesis

In this thesis the following terminology will be used, defining the terms device, component, system, and circuit.

- A single transistor is referred to as **device**. A single sleep transistor is denoted as device.
- **Components** refer to register transfer level (RTL) components. Examples to be mentioned here are adders, multipliers, registers, and multiplexers.
- More complex datapaths containing multiple RTL components including a dedicated controller are referred to as **system**.
- The term **circuit** is used in a lax manner and is not necessarily fixed to a level of abstraction. For example, several design techniques can be applied to single gates, components, or even systems. To signify this flexibility the term circuit is used.

Several research projects at the institute for information technology OFFIS aimed at the estimation and optimization of power in ASIC designs. At the beginning the focus was on dynamic power. Later, leakage power estimation and its *static optimization* gained importance. At the same time the level of abstraction raised from transistor- to behavioral-level.

The idea of this thesis is to logically continue previous work by including *DPM* techniques to target leakage optimization during runtime of a design. Thereby, the scope is on power-gating that is applied to functional RT-level units of ASIC designs as adders or multipliers. Figure 1.2 gives an overview on the four-folded flow that is proposed. It is divided into a *modeling-*, *estimation-*, *optimization-*, and *output generation*-part leading to a holistic IP-synthesis flow with the consideration of dynamic leakage-management.

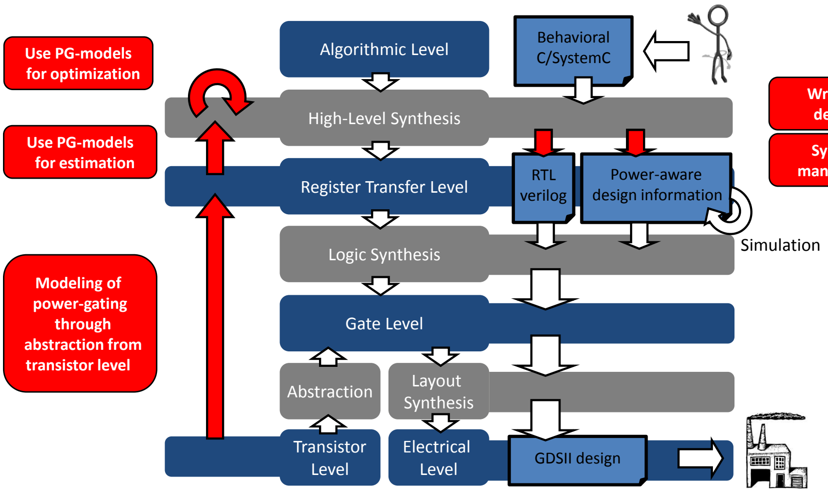

Use PG-models for optimization
Use PG-models for estimation
Modeling of power-gating through abstraction from transistor level
Algorithmic Level
Behavioral C/SystemC
High-Level Synthesis
Register Transfer Level
RTL verilog
Power-aware design information
Simulation
Logic Synthesis
Gate Level
Abstraction
Layout Synthesis
Transistor Level
Electrical Level
GDSII design
Write power-aware design information
Synthesize power-management controller

Figure 1.2: Visualization of proposed power-gating modeling, estimation, and optimization flow

The main contributions of the thesis are summarized in the following:

- **A discussion of power-gating estimation and its granularity problem.**
  Power-gating estimation implicates a huge bunch of parameters that impact resulting leakage savings. In contrast to simple cell-based gating approaches, in this thesis power-gating is applied to RTL components that even complicates its estimation. Transistor-level simulations are applied to identify important parameters and also to explore temporal restrictions on the application of power-gating.

- **The definition of requirements on power-gating RT-level models.**
  The superordinated goals of the models are to enable automated decisions on temporal granularity of power-gating application based on break-even times and high-level design tradeoffs. Therefore, it is necessary to predict leakage currents in active and standby state, wake-up time and costs, virtual supply or ground voltages in standby state, and area overhead under consideration of all important parameters.

- **A widely automated model characterization process.**
  A power-gating modeling process through abstraction from transistor level is implemented to perform an automated characterization for a given transistor technology and its parameters (see Figure 1.2). The set of derived models allows a holistic, cycle-accurate, and fast estimation of power gated RT-components. Combined with only few characterization data, this modeling approach builds a cornerstone of the following high-level leakage optimization.

- **An estimation flow for RT-level power-gating.**
  The power-gating models are completely integrated into the cycle-accurate power estimation of the PowerOpt® high-level synthesis tool that implements the algorithmic- to RT-level synthesis.

Thereby, the dimensioning of sleep transistors implies a tradeoff between faster computation and less leakage. To relieve the designer from decisions on transistor level, a delay-dependent sleep-transistor sizing approach enables a completely automated estimation.

- **A dynamic leakage-management methodology applied during high-level synthesis.**
  Two approaches are implemented to enhance the high-level synthesis in terms of the application of dynamic leakage-management techniques. Firstly, an operator binding and allocation approach that minimizes both, static and dynamic power is proposed. In contrast to existing works it incorporates different power modes into synthesis and optimizes for an efficient use of power-management.
  Secondly, the optimization criterion of the scheduling is changed into a heuristic for improving the effectiveness of techniques such as power-gating. The proposed ILP-formulated scheduling performs an operation clustering within the execution-time of the critical path and serves as a heuristic to minimize the amount of power state transitions. The resulting schedule further improves the binding and allocation phase and operations can be bound to resources cluster-wise. As a result, emerging idle times can be exploited by power-management techniques. Both, the low leakage allocation and the scheduling are integrated into OFFIS's PowerOpt® tool.

- **A power manager controller synthesis.**
  Power-management at architectural level raises the problem of its controllability during runtime. The datapath is expanded by control signals from the controller to the components. During synthesis, the controller is extended to a power manager by a static analysis of the fixed schedule. It is further shown that the adoption of the datapath is compliant to industrial standards and that the results of the *output processing* can be passed to subsequent tools (see Figure 1.2).

- **The evaluation of the model characterization.**
  The models are evaluated with different transistor technologies, types of power-gating, components, and parameters. Parameters are varied within their possible ranges, and error measures are applied. The models are further used to analyze possible savings at system-level.

- **The evaluation of the leakage-management methodology.**
  In the second part of the evaluation the models are applied within the new behavioral synthesis flow. The analysis bases on several practical examples. The results are compared to results of previous high-level synthesis algorithms known in literature and improvements of the power consumption are discussed.

1.4 Thesis Outline

The rest of this thesis is organized as follows: The next chapter briefly describes basic concepts and background information on leakage currents and its management technique applied in this thesis. Additionally, fundamental tasks of the high-level synthesis are presented. Chapter 3 contains a short presentation of existing power standards to unify the application of power-management across different levels of abstraction. It also includes a presentation of the state-of-the-art in modeling, estimation, and optimization from both, industry and research. The modeling and estimation flow is described in Chapter 4 including its simulation environment. The models are then applied within a leakage-minimizing high-level synthesis. Chapter 5 proposes the new leakage-optimized allocation, binding, and scheduling. Additionally, necessary enhancements of the controller synthesis are described to take an advantage of the power-manageable datapath. In Chapter 6 the models are evaluated based on a selection of technologies and the improvements of the power-management (PM)-aware synthesis are analyzed on different practical design examples. The thesis closes with a short summary, conclusion, and outlook in Chapter 7.

2 Basic Concepts and Background

In the following basic concepts, techniques, and background information are presented. Section 2.1 describes leakage currents with its sources and factors and gives an overview on techniques for its reduction. The most promising technique, power-gating, is described in detail in Section 2.2. Fundamentals of the high level synthesis, as it is being optimized in this thesis, are presented in Section 2.3. The following presentation explicitly is not an exhaustive description on these topics but summarizes aspects that are necessary as prerequisites for the subsequent chapters.

2.1 Leakage

The two states of a transistor are getting closer since the supply voltage swing as well as the threshold voltage are getting smaller and smaller. Additionally, transistor scaling reaches atomic scale and forces quantal effects. In a consequence, the picture of a transistor being a perfect digital switch with two clearly defined states developed more and more into the direction of becoming an analog and leaky switch.

Figure 2.1 [Hel09] shows a cross-section through an n-channel metal-oxide semiconductor (NMOS) transistor with different kinds of present leakage currents. While locking, drain is typically at a high potential and all other terminals are at low. In this state a subthreshold current I_{subth} flows through the channel, a gate tunneling induced current I_{gate} occurs from gate to source/drain/bulk, a punchthrough current I_{punch} flows from drain to source when both pn-junctions touch each other, an unavoidable current $I_{junction}$ flows though the pn junctions, and a current is induced from gate to source I_{gisl} and drain I_{gidl} in the gate-(drain/source) overlapping area. If the channel is conducting, source, drain, and gate have the same potential and thus only gate- and pn-junction leakage occur. During switching hot carrier injection occurs where electrons or positive charge carriers overcome the potential barrier and are injected into the gate dielectric leading to a current I_{hci} from gate to the channel.

From a historical point of view, these currents represented a vanishing part in the above-$100nm$ age but have developed to the significant contributor of the total power dissipation as analyzed and predicted in [KAB$^+$03], [MR03], and [ITR10]. Especially in many mobile applications large fractions of an ASIC idle most of the time and the leakage power is the only contribution during these idle times.

In the following a closer insight will be given for the two most important leakage types, subthreshold and gate leakage, in order to identify and highlight parameters of big impact. These parameters will serve as basis for the modeling process in Chapter 4. A far closer description including an exhaustive presentation of leakage currents and its parameter dependencies is given in [Hel09] and [RMMM03].

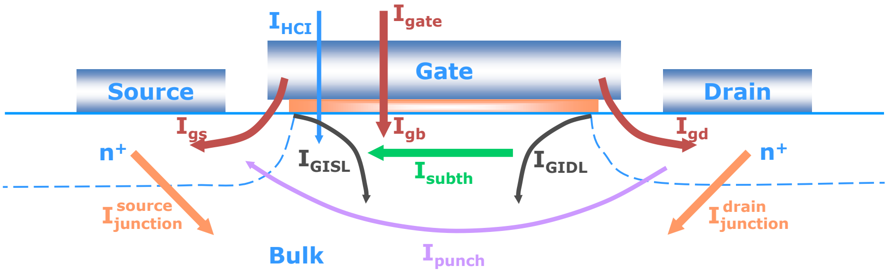


Figure 2.1: Cross-section through an NMOS transistor and different kinds of leakage currents in locking, conducting, and transient state [Hel09]

2.1.1 Subthreshold Leakage

Figure 2.2 ([Hel09]) shows an I_{SD}/V_{GB}-curve of an NMOS transistors manufactured in the $45nm$ and $130nm$ predictive technology model process of [IaASU]. This curve characterizes the conductivity of a transistor as a high I_{ON} current can flow through the channel in conducting state and a lower I_{OFF} current remains while locking. The kink in the curve defines the threshold voltage V_{TH} and the characteristic slope is given for the linear part. As it can be seen, I_{ON} is in the same order of magnitude for both transistors but the remaining subthreshold leakage (I_{OFF}) is 10^6 times higher for the $45nm$ technology.

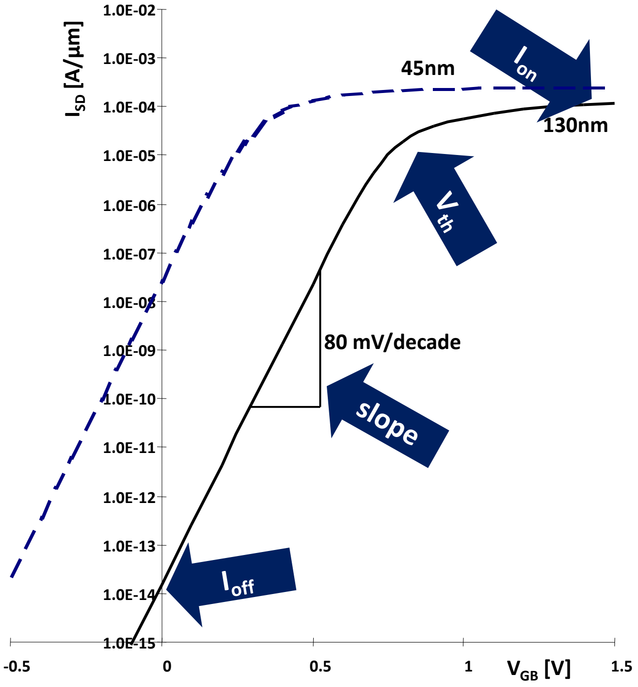


Figure 2.2: I_{SD}/V_{GB}-curve of an NMOS transistor based on $45/130nm$ PTM process [Hel09]

The equation, describing subthreshold currents in a transistor, as it is defined in the Berkeley short-channel IGFET model (BSIM) manual [DYX$^+$07] and an approximation for the locking state with a drain-source voltage close to V_{DD} is given in Equation 2.1. It can be seen that subthreshold leakage scales linearly with the design parameters channel width W and length L, exponentially (dominating the quadratic impact) with the inverse thermal voltage V_T (being itself linearly dependent on the temperature T), and again exponentially with the threshold voltage V_{TH} and gate-source voltage V_{GS}.

$$I_{subth} = kV_T^2 \frac{W}{L} e^{\frac{V_{GS}-V_{TH}}{nV_T}} (1 - e^{-\frac{V_{DS}}{V_T}}) \approx \alpha e^{-\beta \frac{V_{TH}}{T}} \tag{2.1}$$

2.1.2 Gate Leakage

Today's gate isolation layers have a height of few SiO molecules. Electrons can tunnel through the isolation layer, although the potential barrier energy is higher than an electron's energy. The tunneling probability rises exponentially while the isolation layer becomes thinner and resulting unwanted currents significantly contribute to the total power dissipation. These currents flow from gate to either source (I_{GS}) or drain (I_{GD}) (directly in the overlapping area or through the channel), or to bulk (I_{GB}) and occur in every transistor state. They are summarized as gate leakage.

$$I_{gate} \propto \frac{\lambda_S W V_{GS}^2 + LWV_T V_{GB} + \lambda_d W V_{GD}^2}{T_{ox}^2 e^{\beta T_{ox}(\alpha_0 + \alpha_1 V_{ox} + \alpha_2 V_{ox}^2)}} \tag{2.2}$$

An approximation of gate leakage currents is given in Equation 2.2. Beside the isolation layer thickness T_{ox}, gate leakage majorly depends on the gate-(source/drain) voltage V_{GS}/V_{GD} and the gate-drain/gate-source overlapping area size $W\lambda_d$ and $W\lambda_s$, respectively. A temperature dependency only exists for gate-to-bulk currents but compared to I_{GS} and I_{GD}, I_{GB} is negligible small.

2.1.3 Leakage-Management Methodologies Overview

Two different classes of leakage-management techniques can be separated: *static* and *dynamic* techniques. They are all considered and applied during the design but the temporal manner when leakage currents are reduced differs. Static techniques do not imply different power modes and as a consequence, beside leakage currents, they also reduce other parameters like the maximum frequency at all time. In contrast to this, dynamic techniques implicate different modes during runtime and offer more flexibility. Saving-modes might affect leakage currents, the dynamic power consumption as well as the circuits speed but if necessary, the design can operate at a normal mode. In contrast to static techniques, mode transitions become necessary and complicate the overall adoption of these techniques.

Static Leakage-Management Techniques

Voltage Island Timing constraints may differ between several circuit parts of a design. This observation or given demand in combination with the superlinear dependency between leakage currents and the assigned supply voltage can lead to a subdivided design. An assignment of different supply voltages to these islands is static and needs to be known during the design phase.

Static Body Biasing Beside the three functional terminals *drain*, *gate*, and *source* of a transistor the *body* terminal, typically connected to *source*, can be separated and loaded to an independent and different potential. This will significantly impact the channel conductivity and switching performance. It can either lead to an improved performance in the active mode if a forward-biasing is applied or to a reduced leakage current in an idle phase if a reverse-biasing is applied to the body. The latter reverse body biasing technique increases the threshold voltage V_{TH} that in turn reduces the performance and especially sub-threshold leakage currents. The static body biasing assigns fixed voltages during design in order to create low leakage or high performance circuits.

Dynamic Leakage-Management Techniques

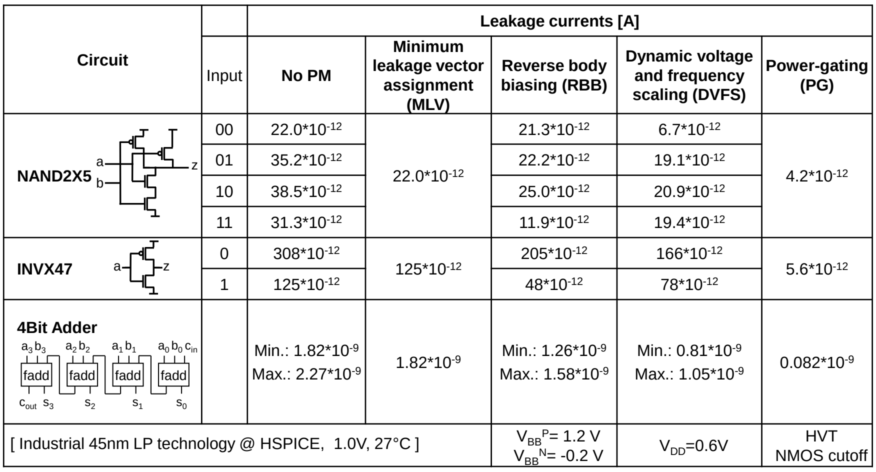
Dynamic Body Biasing Body voltage assignment can also be done dynamically. In this case two operating modes exist. The body voltage is forward-biased in the active mode and/or reverse-biased in the idle mode to meet the varying performance demands. For this technique, multiple voltages need to be provided and additional switches are necessary.

Power-Gating Power-gating (PG) is the most intuitive technique for reducing leakage currents. It introduces a power switch to temporarily power down unused parts of an ASIC design. In this state the difference in leakage currents $I_{ACTIVE} - I_{SLEEP}$ is saved. The switch is typically made of a single transistor and is also referred to as *sleep transistor*. As simple as the idea of power-gating sounds, it is complicated to implement and to consider its impact during the design because of an immense range of implementation artifacts. Most important design parameters are the type and size of a sleep transistor, the interfacing to neighboring components, as well as the component size to which power-gating is applied. A major drawback of power-gating is the loss of the internal state during sleep.

Minimum Leakage Vector Exploiting the input data dependency of leakage currents is the idea of the minimal leakage vector (MLV) technique. It assigns fixed and pre-defined data vectors to the input of the target component in order to reduce its internal leakage current. Main problems of this technique are the determination of the MLV for designs with a large input bitwidth, the diminishing effect of leakage controllability by just assigning the inputs, and the state retention. Like in the PG approach the internal state will be lost.

Dynamic Voltage and Frequency Scaling Similar to the static voltage island assignment, the dynamic voltage and frequency scaling (DVFS) technique exploits the dependency between leakage currents and the assigned supply voltage. The difference is that both the supply voltage and frequency of a component are reduced in times of a low utilization and are not statically assigned. Main problems of implementation arise from providing multiple supply voltages and clock frequencies to the chip areas where DVFS is applied to. In contrast to PG and MLV, the state will be preserved in the DVFS approach. It can be distinguished between approaches where the component is still functional at a lower voltage and frequency and a forced application where only the state is preserved but the component is no longer able to operate.

All mentioned techniques do not come free of costs. Large computational effort for finding the MLV or complicated design decisions for finding suitable voltages and frequencies meeting all requirements are only few examples. They also have in common that the ASIC's area increases due to the need for hardware to apply the MLV, the power-gating switch, and additional routing effort for multiple bias or supply voltages.


a
b
z
a
z
a_3 b_3
a_2 b_2
a_1 b_1
a_0 b_0 c_{in}
fadd
fadd
fadd
fadd
c_{out} s_3
s_2
s_1
s_0

Circuit		Leakage currents [A]				
	Input	**No PM**	**Minimum leakage vector assignment (MLV)**	**Reverse body biasing (RBB)**	**Dynamic voltage and frequency scaling (DVFS)**	**Power-gating (PG)**
NAND2X5	00	$22.0*10^{-12}$	$22.0*10^{-12}$	$21.3*10^{-12}$	$6.7*10^{-12}$	$4.2*10^{-12}$
	01	$35.2*10^{-12}$		$22.2*10^{-12}$	$19.1*10^{-12}$	
	10	$38.5*10^{-12}$		$25.0*10^{-12}$	$20.9*10^{-12}$	
	11	$31.3*10^{-12}$		$11.9*10^{-12}$	$19.4*10^{-12}$	
INVX47	0	$308*10^{-12}$	$125*10^{-12}$	$205*10^{-12}$	$166*10^{-12}$	$5.6*10^{-12}$
	1	$125*10^{-12}$		$48*10^{-12}$	$78*10^{-12}$	
4Bit Adder		Min.: $1.82*10^{-9}$ Max.: $2.27*10^{-9}$	$1.82*10^{-9}$	Min.: $1.26*10^{-9}$ Max.: $1.58*10^{-9}$	Min.: $0.81*10^{-9}$ Max.: $1.05*10^{-9}$	$0.082*10^{-9}$
[Industrial 45nm LP technology @ HSPICE, 1.0V, 27°C]				V_{BB}^P= 1.2 V V_{BB}^N= -0.2 V	V_{DD}=0.6V	HVT NMOS cutoff

Figure 2.3: Overview on possible leakage reductions induced by different leakage-management techniques

Figure 2.3 compares the different dynamic leakage-management techniques in terms of their potential leakage savings at different levels of granularity. They are all applied at a medium sized NAND-gate (smallest), a largest available inverter-gate (middle sized), and a $4bit$ adder component (largest circuit observed in this overview). All measurements were obtained by circuit-level simulations and refer to an industrial $45nm$ low power (LP) device technique at a supply voltage of $1V$ and an operating temperature of $27°\mathrm{C}$. At gate-level, leakage measurements are listed for all possible input-vectors, whereas at RT-level minimum and maximum currents are presented.

It can be seen that the leakage reduction due to MLV-assignment diminishes from $40 - 60\%$ at gate- to 20% at RT-level and will lead to the smallest possible savings. Reverse body biasing (RBB) and DVFS will result in savings of $30 - 70\%$ with an offset voltage of $0.2V$ for RBB and a reduced supply voltage of $0.6V$ for DVFS. Power-gating outperforms these improvements by far and will result in savings up to 95%. Each circuit is power-gated by one smallest available NMOS sleep transistor in the high threshold voltage (HVT) version. It can impressively be seen that the fine-grained appliance of power-gating limits the savings of the NAND-gate to about 87%. This is because the channel-width ratio between the sleep transistor and the transistors in the power-gated circuit is much bigger for a small circuit and thus power-gating is less effective. In turn, power-gating is more effective the larger the power-gated circuit is.

2.1.4 Summary

Leakage currents constitute in the same order of magnitude to the total power consumption of today's semiconductor designs as the dynamic power due to capacitance charging. In circuits with a bigger part of idleness the fraction of leakage currents is even higher and dominates the total power dissipation. The encouraging factors of leakage can be separated into dynamic and static parameters. Dynamic parameters may change during runtime and can be set from outside such as the surrounding temperature, supply or body voltage. Static parameters are fixed during manufacturing such as the gate oxide thickness, type and level of doping concentration, and are encapsulated in the semiconductor technology.

Beside manufacturing techniques such as high-k dielectrics (to counteract gate leakage currents) or future FinFET design (three-dimensional channels for a better channel blocking to reduce subthreshold currents), dynamic power-management gained the most important class of runtime techniques. As analyzed on example circuits, power-gating has the biggest impact on all kinds of leakage currents and may cut off more than 90% of them.

2.2 Power-Gating in Detail

Transistor stacking is one of the most important inventions for the design of digital integrated circuits. It has been introduced in 1963 in the complementary metal-oxide semiconductor (CMOS) technique where stacked and complementary switching pull-up and pull-down networks prevent a circuit from high short-circuit currents to flow from supply to ground through the transistor channels.

This principal of transistor stacking is also used for power-gating that has been invented by M. Horiguchi, T. Sakata and K. Itoh in 1993 [HSI93]. Beside the pull-up and pull-down network a third network is connected in series, either above the pull-up or below the pull-down network. It contains a functionally-redundant switch in terms of a p-type transistor header or n-type transistor footer device. While this transistor is closed, it is intended to be as transparently as possible regarding the functionality of the circuit. In the open state it controls the supply or ground voltage and thus power-gates a circuit. Its purpose is again to reduce currents but, in contrast to the CMOS methodology the focus is on leakage currents occurring as described in Section 2.1. During this state, the circuit is *sleeping* and not able to operate. Thus, power-gating can only be applied during times of disuse.

In the following, the impact of this additional transistor will be discussed in detail for the two static states as well as for the state transition. All observable and important currents, voltages, and other parameters are defined. Section 2.2.1 then shows different possible implementation schemes described in literature. The block size, to which power-gating is applied, is discussed in Section 2.2.2 followed by interfacing considerations in Section 2.2.3. Finally, Section 2.2.4 proposes more advanced and probably future concepts of power-gated designs.

Trace of a Power-Down Sequence

Figure 2.4 schematically illustrates a sequence of powering down and waking up a circuit block c_1 with a p-type header device. The circuit is controlled by a $sleep$ signal and two characteristic currents I_1 and I_2 are plotted over time. The sequence is divided into the following phases. $active_1$ describes a phase of idleness. At this time c_1 has constant in- and outputs and the overall system is stable. In the second phase $power\ down$, the switch is opened and the circuit c_1 powers down until it enters the following static $sleep$ phase. After a while, c_1 is woken up ($wake\ up$) and traverses a second active phase $active_2$.

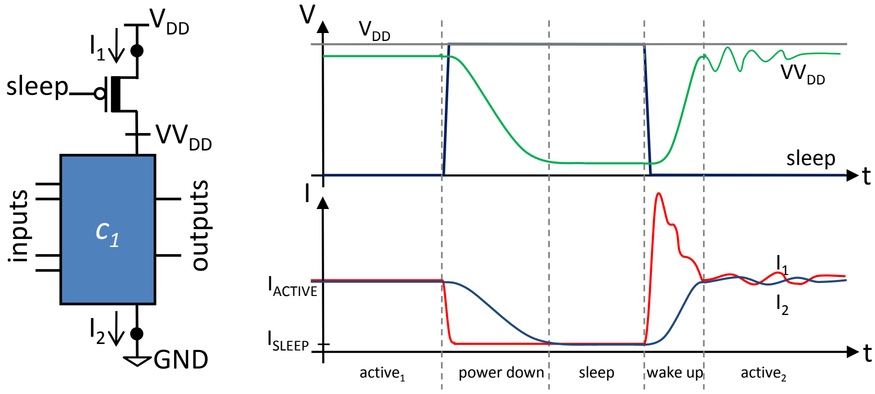


Figure 2.4: p-type power-gating functionality

Static active state In the active modes, the sleep transistor conducts but it cannot be prevented that a small IR drop $V^{ST}_{DROP\text{-}ON}$ occurs across the sleep transistor. This is because the transistor retains a small resistance $R_{ST}(t)$ while conducting. The series connection of sleep transistor and circuit acts as a voltage divider with a comparatively small sleep transistor resistance and high circuit resistance. Both resistances vary over time dependent on the data input signals of c_1 as they cause internal activations. As a consequence, the effective circuit's supply voltage reduces to a so-called virtual supply voltage VV_{DD}. If an NMOS-device is used for power-gating, the ground voltage is raised to a virtual ground level $VGND$. The voltage drop $V^{ST}_{DROP\text{-}ON}$ across the sleep transistor is thus defined as:

$$V^{ST}_{DROP\text{-}ON}(t) = \begin{cases} V_{DD} - VV_{DD}(t) & \text{for p-type power-gating} \\ VGND(t) & \text{for n-type power-gating} \end{cases} \tag{2.3}$$

In both cases, the effective supply voltage of c_1 is defined as:

$$V^{eff}_{DD}(t) = V_{DD} - V^{ST}_{DROP\text{-}ON}(t) \tag{2.4}$$

As an important consequence, the delay of a circuit increases because it is a function of the effective supply voltage.

Figure 2.4 distinguishes between two active scenarios. In $active_1$, the data inputs of the circuit are fixed and the overall system is in a stable idle state. As it can be seen, the virtual supply voltage VV_{DD} as well as the leakage current I_{ACTIVE} are constant. In $active_2$, the input signals of c_1 change their values. During this phase, switching activity propagates within c_1 until the stable state of $active_1$ is reached. It can be seen that an activity induced current flow impacts the division of VV_{DD} and $V^{ST}_{DROP\text{-}ON}$.

Transient power-down state From the moment of activating the power switch, VV_{DD} drops with time until it saturates at a voltage close to GND. At the same time I_1 and I_2 reduce to I_{OFF} that is limited to the subthreshold and gate leakage current through the sleep transistor. As shown, I_1 and I_2 diminish with a different rate. While I_1 is cut by a sharp edge as $sleep$ passes V_{TH}, I_2 decreases slowly and discharges the capacitances in c_1. During this phase, the state within memory elements of c_1 is lost if no further state-preserving techniques are applied as presented in Section 2.2.4. The time for powering down to the static sleep state is denoted as $t_{powerdown}$.

Static sleep state In the *sleep* state, I_1 only describes remaining leakage currents that are effectively reduced. In this stable state, I_1 defines I_{SLEEP} and depends on the saturated voltage level at VV_{DD} that in turns depends on the resistance ratio between sleep transistor and power-gated circuit. In the end, I_{SLEEP} is a function of several parameters such as transistor technology, power-gated circuit c_1, c_1's inputs, sleep transistor parameters (i.e. W_{ST}, V_{TH}, ...), or ambient temperature.

I_2 levels out at a slightly higher current compared to I_1 because gate-leakage currents from logically high inputs flow through the circuit as well.

Transient wake-up state At the beginning of this phase a falling edge occurs at the sleep signal and thus the sleep transistor conducts. The capacitances within c_1 are loaded and a high current is drawn during wake-up. The wake-up time that is necessary to obtain the steady active state with a virtual supply voltage close to V_{DD} is denoted as t_{wakeup}. During wake-up, the maximum current is limited to I_{ON} of the sleep transistor. The overall consumed energy due to capacitance charging is denoted as state transition energy E_{SW}^{RT} as defined in Equation 2.5.

$$E_{SW}^{RT} = \int\limits_{t_{wakeup}} V_{DD} \cdot I_1(t) \tag{2.5}$$

The previous trace is only slightly different for NMOS power-gating and all previous discussions are applicable for footer cells as well. Obviously, $sleep$ needs to be inverted. The most important difference regards the voltage levels that appear. In active state, the voltage across the circuit is again close to V_{DD} with a small IR drop across the sleep transistor. The circuit node between sleep transistor and circuit is now denoted as virtual ground with a voltage of $VGND$. During power down the circuit again continues to leak but this time the circuit's internal nodes saturate at V_{DD}. Defining I_1 as the current flowing through the sleep transistor, the E_{SW}^{RT}-definition holds for header- as well as footer-based power-gating.

Profitability of Power-Gating

If a sleep phase of a circuit is long enough to amortize the state switch costs, power-gating is a disbursing technique. With the definition of state transition energy in Equation 2.5 and known leakage currents in active and sleep state (I_{ACTIVE} and I_{SLEEP}) a break-even time t_{be} can be computed. Regrettably, some further costs occur that need to be considered in the t_{be} computation. For example, the header or footer transistor, necessary interfacing circuits, or buffers (the latter two will be described in Section 2.2.3) have inherent leakage currents and consume an inherent amount of energy during state transition. Thus, t_{be} is defined as shown in Equation 2.6 with $E_{OVERHEAD}$ and $I_{OVERHEAD}$ summing up all additional energies and currents respectively.

$$t_{be} = \frac{E_{SW}^{RT} + E_{OVERHEAD}}{(I_{ACTIVE} - (I_{OFF} + I_{OVERHEAD})) \cdot V_{DD}} \tag{2.6}$$

In Chapter 4 the overhead energies and additional currents will further be quantified.

2.2.1 Microarchitectural Implementation of Power-Gating

Figure 2.5 shows some exemplary types of microarchitectural implementations of power-gating. Each of these possible implementations is referred to as power-gating scheme (PGS) in the subsequent part of this thesis. In general, the first three schemes apply p-channel metal-oxide semiconductor (PMOS) gating whereas the last scheme implements NMOS gating. For powering down, PMOS sleep transistors are driven by a signal $sleep$ equal to the supply voltage whereas NMOS transistors are driven by the inverse signal $\overline{sleep}$. In the first scheme, a PMOS transistor that is built in a standard threshold voltage (SVT) process is used for powering down the circuit. The power-gated circuit is typically also made of SVT transistors to provide highest possible performance. This kind of power-gating implementation, also referred to as cutoff CMOS (CCMOS), is nearly outdated, but is considered in this thesis for the sake of completeness. The second scheme implements an HVT sleep transistor (indicated by the thick channel) to maintain a higher potential at the virtual rail and thus to push the suppression of leakage currents. Another advantage of HVT sleep transistors is a reduced inherent leakage current of the transistor itself being relevant because of its size as described in Section 2.2.2. A combination of HVT sleep transistors and SVT circuit is state-of-the-art in today's realization of power-gated designs. For this reason, power-gating is also often referred to as multiple threshold CMOS (MTCMOS) in literature although multi-threshold CMOS only indicates the availability of devices with different threshold voltages, typically a low and a high threshold device. [RMMM03] gives an overview on possibilities how different threshold voltages can be obtained during manufacturing either by body biasing or gate engineering.

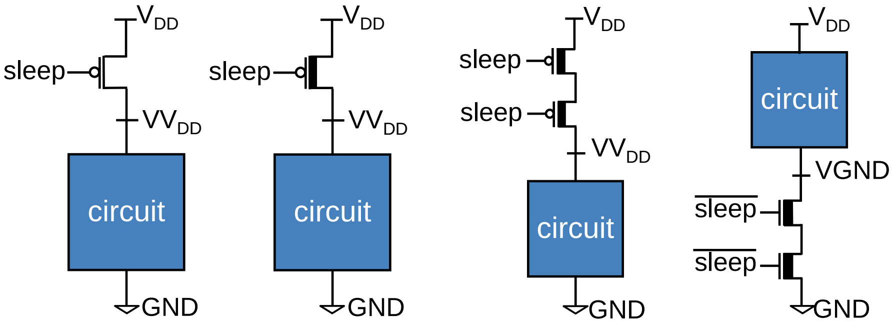


Figure 2.5: Microarchitectural implementation of different power-gating schemes

To enforce the leakage suppression, the $sleep$ input of a PMOS transistor can be driven by voltage values above V_{DD}. In literature, this technique is referred to as super cutoff CMOS (SCCMOS) [KNS00].

A problem that arises in doing so is a high voltage V_{GD} between the *gate-* and *drain*-terminal of the sleep transistor. This leads to a high voltage stress that may even result in a gate oxide breakdown of the transistor. To overcome this problem the third power-gating scheme applies two sleep transistors in series. A SCCMOS implementation then drives the supply-facing device with a sleep voltage above the supply voltage and the circuit-facing transistor with the supply voltage. Then, a voltage level arises between the two sleep transistors that is uncritical regarding the gate-to-drain voltage of the supply-facing transistor [MS02]. The transistor stack further leads to a higher voltage drop across the transistors in both, the sleep and active state. This will better suppress the leakage currents during sleep but will also further reduce the virtual supply voltage and thus the speed during operation.

Every PMOS gating scheme has a corresponding NMOS gating based counterpart. For example, the fourth scheme shown in Figure 2.5 is the NMOS gating counterpart of the one before. NMOS gating has significant advantages compared to PMOS gating because the on-current I_{ON} is about twice as high if the transistors are of the same size. Thus, an NMOS sleep transistor has less impact on the active state and on the state transition or can be smaller at the same performance.

Beside the sleep transistor selection, other critical design considerations exist in order to optimally power-gate. [SH06b, SH06a] summarize and trade off header vs. footer cells, grid vs. ring style placement and analyzes the sleep transistor efficiency based on gate length, width, and body voltage.

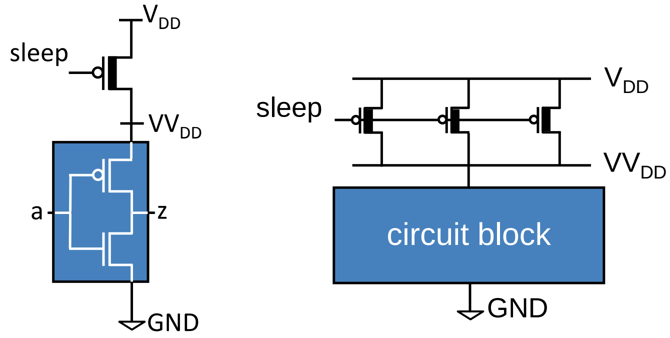
All mentioned techniques have in common that the saturated voltage level equalizes to either V_{DD} in NMOS case (and GND in PMOS case respectively) for the whole power-gated circuit. Thus, during wake-up of a PMOS gated circuit all nodes of the pull-up network conduct simultaneously. After a while, the voltage is close to V_{DD} and a big fraction of the pull-up network stops conducting. The circuit ripples until a steady state, defined by the inputs, is obtained. A specialized form of power-gating called zigzag SCCMOS (ZSCCMOS) can reduce this unnecessary dynamic power due to the rippling for a fixed and predefined state during power down [MS02]. PMOS and NMOS sleep transistors are mixed in this approach according to the input values of a gated circuit. For example, an inverter with a logic one as input is PMOS gated because then the virtual supply voltage level saturates to a voltage level above GND and even if high cutoff voltages are applied (see SCCMOS) the sleep transistor is not subjected to a high voltage stress. During wake-up, the capacitances are already precharged and much less power is consumed. In turn, this technique is not as powerful in reducing leakage currents because of its fine-grained application.

2.2.2 Granularity of Application

Beside the scheme of power-gating implementation, another important aspect is its granularity of application in both spatial as well as temporal meaning.

Spatial granularity considers the circuit size to be powered down. Approaches presented in literature range from fine-grained techniques with dedicated sleep transistors for each gate of a technology up to coarse-grained techniques where large IP components are power-gated via a single sleep transistor or a cluster of parallel sleep transistors. In the following, the pros and cons will be explained.

Fine-grained spatial power-gating duplicates every cell in the standard cell library and adds a single sleep transistor [KS04] as it is shown for an inverter cell in Figure 2.6(a). Thus, a fine-grained power-gating compatible technology typically contains twice as many cells. The sleep transistor size is individ-



(a) Fine grained power-gating at gate level

(b) Coarse-grained power-gating at block level

Figure 2.6: Spatial granularity of power-gating

ually matched to the gated cell and is fixed in the library. Thereby, sizing depends on the I_{MIC} drawn during active state. This highly decentralized approach implies some advantages. First of all, the timing and delay information can be analyzed during library characterization and thus be stored in the library files such as in the liberty file format. The same holds for the layout as every cell's layout is fixed and the area overhead is known. As no separate sleep transistors are required, the effort for placement will not be increased. The third advantage of the fine-grained approach is a compatibility with existing design tools and thus its seamless integration in existing tool flows. On the other hand, the huge number of sleep transistors implies some disadvantages. Since every sleep transistor needs to be controlled, the sleep signal distribution demands a high effort during interconnect routing. Secondly, every sleep transistor should be properly sized for its gate. Beside the pessimistic sizing for I_{MIC}, the lower bound for transistor width is limited by the technology size. From a global perspective this will lead to a far pessimistic design and thus a large area penalty. Additionally, the efficiency of power-gating is limited as already shown in the NAND-gate example of Figure 2.3 for small gates because the relation between I_{OFF}^{ST} and I_{ON}^{Gate} is too big. This effect will become even worse for smaller gates. Moreover, the library complexity is increased due to the number of gates and annotated timing information.

Approaches powering down more than a single gate by one or multiple sleep transistors connected in parallel are referred to as **coarse-grained spatial power-gating** techniques. They can further be divided into *cluster based sleep transistor design (CBSD) techniques* where each cluster of few gates has its own sleep transistor and *distributed sleep transistor network (DSTN) based design techniques* where several sleep transistors are shared to power down larger circuit parts [LH04] and are coarsely placed throughout the chip. Sharing means that there is one virtual supply or ground line among the sleep transistors. In today's industrial power-gating designs, DSTN design is state-of-the-art [SLJY08]. Figure 2.6(b) exemplarily shows a circuit block powered down by three shared sleep transistors.

In these approaches only a few header or footer cells of different sizes are added to the library (as presented in [CPS$^+$07] for a $65nm$ technology node) and are characterized separately from the power-gated circuit. Due to the linear proportionality between W_{ST} and I_{ON} (and I_{OFF} respectively) they are then combined in order to meet the target size. Other advantages of the coarse-grained application arise from the sharing of sleep transistors. On the one hand the sleep control distribution is much easier. On the other hand the area overhead can be reduced significantly because the sleep transistors share the virtual supply or ground line and the I_{MIC} of a large circuit block is much lower than the aggregated maximum currents of its individual parts. Additionally, it is less sensitive to process-voltage-temperature (PVT) variation and, if the distributed sleep transistors are sized properly, it introduces less IR drop variation [SH06a]. But the sleep transistor size is also the main problem of this approach because it is now a parameter of the design trading off area overhead and timing degradation. Another drawback is the lack of electronic design automation (EDA) tool support for sleep transistor sizing and synthesis of power-gateable circuits.

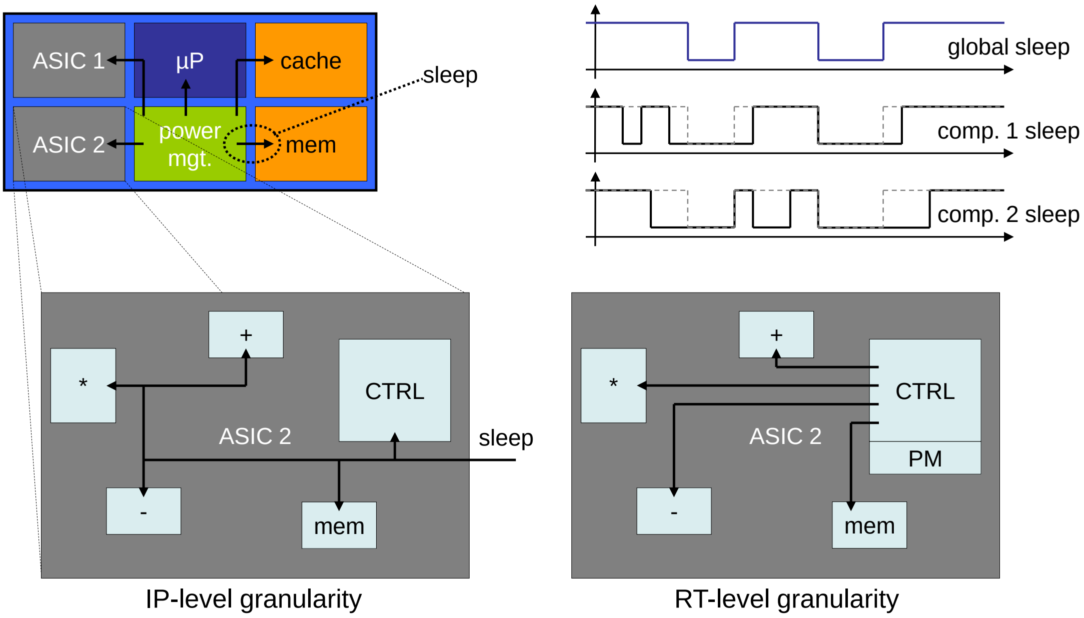


Figure 2.7: Temporal granularity of power-gating application

Within the class of coarse-grained spatial power-gating approaches an additional separation can be made when ASIC hardware is considered. Simple approaches only power-gate ASIC designs as a whole whereas more sophisticated approaches look into it and power down at register transfer (RT) level basis. Figure 2.7 compares these two possibilities. In the first implementation the ASIC does not contain power-management functionalities and is entirely controlled from outside via a global sleep signal. Shorter idle times of single RTL components are neglected. In the example shown, the sleep signal is distributed to every included RTL component everyone having its own power down mechanism. Alternatively, it could also be implemented in a way that the overall ASIC is powered down by a single power switch. In the second version, the ASIC's internal controller is extended by power-management functionalities. It either observes the workload or knows by its controller-states when RTL components can be put into sleep mode. This approach is more flexible as single RTL components such as adders or multipliers can be powered down for few cycles on an individual basis. This is especially an advantage if the workload-traces vary between components.

Of course, these two approaches can be combined in order to apply an autonomous control within the ASIC as well as providing the possibility for a global power down.

In all cases, the break-even time to amortize the state transition costs decides which granularity is the most suitable. Figure 2.8 shows break-even time computations based on measured I_{ACTIVE}, I_{SLEEP}, and E_{SW}^{RT} curves. The measurements have been performed by Synopsys HSPICE® simulations powering down an 8-bit adder by PMOS gating with a supply voltage of $1.0V$ assuming an operating temperature of 77°C. Two transistor technologies have been examined: the Nangate free $45nm$ open source digital cell library based on the predictive technology model (PTM) and an industrial $45nm$ low power technology. The sleep transistor width remained a parameter as indicated in the charts. The subsequently break-even time computation has been performed as defined in Equation 2.6.

The results clearly show the t_{be} sensitivity regarding different technologies. While the break-even time of the PTM technology ranges between $110 - 122ns$ what is equal to about 12 cycles at a $100Mhz$ clock speed, it is significantly higher for the industrial $45nm$ technology. Thus, a temporal fine-grained power down at RT level only makes sense for the PTM technology. In [SSCS10] the authors analyze the break-even time of several larger circuits implemented in an older and less leaky $65nm$ technologies to be in the order of one μs. This result underlines a sophisticated analysis of an application of power-gating.

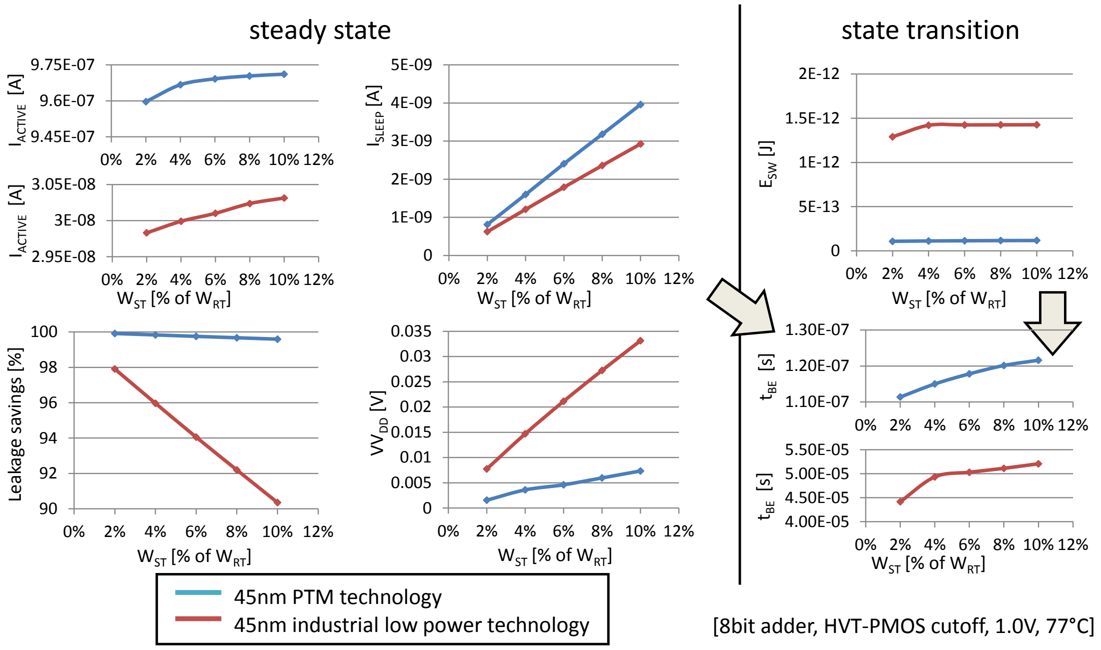


Figure 2.8: Break-even time computations of power-gating application

2.2.3 Interfacing to Power-Gated Circuits

Communicating with a power-gated circuit is a critical issue. First of all, the sleep signal that is typically provided by a controller needs to be lead to the sleep transistor in a sufficient strength. This is important because the spatial distance between sleep transistor and controller on the chip might be big and only small wires should be used for long distances due to the wire load. On the other hand, the sleep transistor size might be significantly larger compared to a standard transistor and thus a high drive is necessary to load its gate capacitance and obtain reasonable switching performance. For these reasons, a cascaded buffer chain is inserted in front of the sleep transistor input in order to stepwise amplify the sleep signal. Figure 2.9 shows such a stepwise tapered buffer chain. The buffers grow rate is the main characteristic to differentiate approaches in literature. It has been shown that an exponentially tapered buffer (i.e. with a ratio of e) provides the minimum system delay [JL75]. Every buffer in the chain thus has an input capacity that is e-times as large as the output capacity of the predecessor. More sophisticated approaches even take the local interconnect capacities between two consecutive stages into account and compute individually sized buffers [CF95a, CF95b, PPD$^+$98], delay-optimized [Bla96], or energy minimizing tapering factors [KAP03]. In practice, buffer sizing also depends on the availability of inverter standard cells.

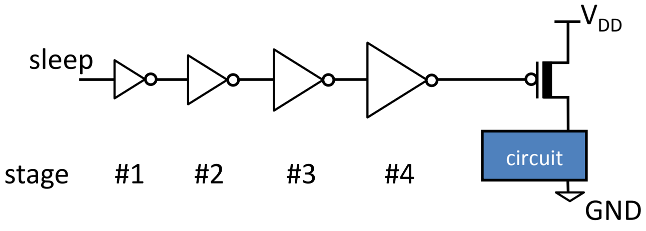


Figure 2.9: Stepwise buffer tapering

Secondly, data outputs of a power-gated circuit need to be isolated because they tend to float at intermediate potentials and cause large short-circuit currents in the subsequent circuit. For this purpose, voltage anchors have been introduced that pull the output voltage up or down to a legal value immediately before the sleep signal is raised. Voltage anchors are small devices and thus introduce a small area overhead of 6 to 14 minimally sized transistors [BBMM06]. An exemplary voltage anchor, forcing the output to the latest value, is shown in Figure 2.10. The voltage anchor activation is derived directly out of the sleep signal that is delayed and then used to power down the circuit.

In contrast to the data output, the inputs of a power-gated circuit are less critical and do not need to be handled carefully. Of course, they have an impact on the saturating voltage but no voltage anchors are needed since they cannot cause short circuit currents or a state-loss.

2.2.4 Advanced Concepts

As shown above, the main limitations for applying power-gating are the high state transition costs and the state-loss after a short time in the gated state.

To overcome the high dynamic power during a state transition, the concept of charge recycling has

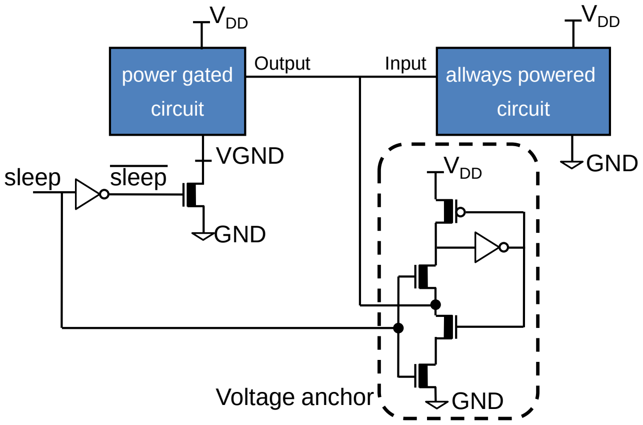


Figure 2.10: Exemplary voltage anchor holding the latest output value [BBMM06]

been proposed [PFP08]. It introduces a transmission gate (TG) as a charge sharing switch between the virtual supply of one and the virtual ground of another power-gated circuit. The approach requires a zigzag-application of power-gating with alternating NMOS and PMOS gating. In the constellation shown in Figure 2.11, the potential at G is very close to V_{DD} and at P it is very close to GND. Immediately before a rising edge occurs at the sleep signal, the TG closes for a short time and the potentials at G and P align each other. In this moment a part of the charged energy in c_1 is transferred to c_2 reducing the transition costs for both circuits.

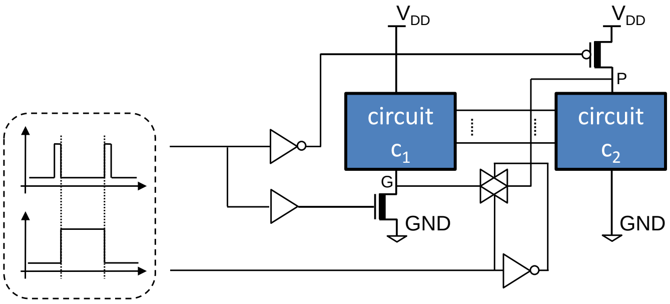
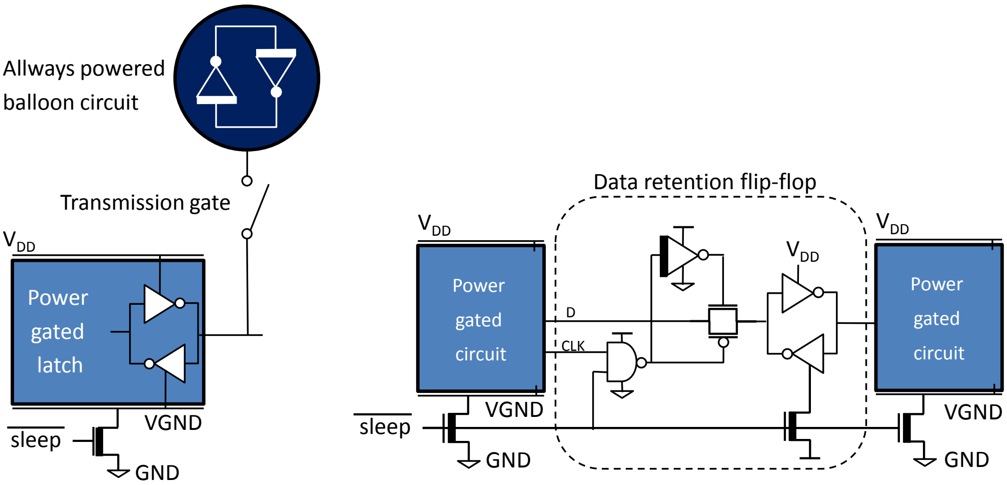


Figure 2.11: Charge-recycling for power-gated circuits [PFP08]

An energy saving analysis showed a reduction of transition energy up to 43% while the wake-up time is maintained. Additionally, this approach has a positive impact on the peak voltage drop and it reduces the ground bounce during wake-up. Nevertheless, this technique also introduces additional costs in terms of area for the TG.

Secondly, to overcome the state-loss of memories during power-gating, balloon latches have been introduced [SMM$^+$97] for state retention. A balloon circuit, as shown in Figure 2.12(a), contains two additional coupled inverters that are always powered on and take over the state from a standard latch or flip-flop via a transmission gate before powering down. During wake-up, the state is restored. While the



(a) Concept of classical balloon latch based data retention

(b) Advanced data-retention flip-flop

Figure 2.12: Different data retention techniques

memory element that should be powered down is implemented in the SVT process, the balloon latch is made of HVT devices, thus the difference in their leakage current is saved during sleep state.

In [MMR04], an even more advanced concept for data-retention is proposed. Figure 2.12(b) shows a data-retention flip flop with properly gated clock and data inputs that holds its state even during sleep mode. The internal clock and data gating needs to be powered on always in order to prevent a change of the stored data in the coupled transistors. By applying data-retention techniques to all state-containing memories, the overall datapath can be powered down and the level of granularity can be raised from RTL-wise gating to system-level gating.

Another possibility to prevent a state-loss is to clamp the virtual supply or virtual ground voltage within the power-gated design. This is done by inserting rail clamp devices such as diodes [KIY^{+}98], additional PMOS devices as diode [KKS04], or by a clever interconnect of the sleep transistor driving buffer [TNN06]. In all of these cases, the falling VV_{DD} or rising $VGND$ is limited to a state-preserving value.

A very new field of research is to find alternative power switches like microelectromechanical systems (MEMS) [HN10]. MEMS are physical levers that bend due to any kind of electrostatic, piezoelectric, thermal, or magnetic force and thus form a bonding. They are optimal switches because they cause negligible voltage drops, do not leak inherently and, due to the physical void, the standby leakage is zero. Drawbacks of MEMS are the large switching time of about $100\mu s$ and the complex integration into CMOS circuits.

In [WC10], spintronic memristors are proposed for use as power-gating switches. Memristors are variable resistors that depend on an integral of the current/voltage profile. To actively power control a circuit with a memristor connected in series, its memristance needs to be increased significantly. This will occur after a certain amount of energy that has been drawn. Due to the lack of external controllability, memristor switches can only be used in special cases such as power budgeting.

2.2.5 Summary

Although significant improvements in transistor design have been made, leakage remains a limiting factor in the *smaller, faster, and less power* mentality of today's semiconductor development. As shown, power-gating is **the** upcoming and seldom already applied technique to control leakage currents. It outperforms any other leakage reduction technique by far but for an automated use in the design of ASIC designs, still several design challenges exist:

- Choice of power-gating implementation scheme,
- Placement and sizing of sleep transistors,
- Automated generation and distribution of sleep signals,
- Sleep signal scheduling for wake-up noise reduction (ground bounce reduction),
- Mode transition energy (E_{SW}^{RT}) minimization, and
- State retention issues.

Today, coarse-grained MTCMOS approaches are state-of-the-art as they can be applied to existing IP. The most important drawback is the need for large sleep times to amortize the costs. In contrast, fine-grained MTCMOS is much easier to implement but offers only a limited leakage reduction. Thus, the future of power-gating will be in between a coarse- and fine-grained application to overcome these limitations. In all cases, many aspects need to be considered for creating models and to optimize designs for power-gating. Solutions to overcome the main problems exist or will be proposed in this thesis and can jointly be applied.

Worth reading summaries of the principles, history, and especially technical implementation details of power-gating techniques are also presented in [Hen07] (Chapter 5) and [SSCS10].

2.3 High-Level Synthesis

Rising design complexity, reuse of design entities, and the need for effective design tradeoffs force the demand for an automated synthesis. Within the process of an HLS (also referred to as *behavioral synthesis* and *algorithmic synthesis*) a formal specification of an algorithmic behavior is synthesized/compiled to hardware in terms of a fully timed microarchitectural description that implements that behavior. The HLS output can then directly be forwarded to conventional logic synthesis with an existing tool support of more than 25 years[1].

First generation HLS tools such as Synopsys®'s *Behavioral Compiler* based on behavioral Verilog or VHDL for design description but have not established because of the languages' insufficiency in modeling a behavior as well as the partial timing abstraction. Today's HLS tools base on standard languages such as ANSI C/C++ or SystemC for modeling behaviors.

Several hopes are related with an automated translation. Hardware could be built more efficiently, optimizations techniques can be adopted automatically, and design-space explorations can be examined

[1] Parts of the Synopsys® Design Compiler, being the first logic synthesis tool, can be dated back to 1986.

for any of the design targets area, performance, power (dynamic as well as static), and reliability. Beside these investigations, a verification of the hardware against its behavior is another important aspect of an HLS.

In general, an HLS consists of the following activities that can be implemented in different orders and with different algorithms:

- **Lexical processing** for representing the behavioral description in a graph notation. Typically, a control- and data-flow graph (CDFG) is used that combines a *data-flow-* with a *control-flow*-graph. Thus, nodes represent either functional operations ($+, -$ or $*$) or control-flow nodes (e.g. split or join) and edges represent either directed data flows (e.g. the output of an adder operation defines an input of a subsequent multiplication) or control edges expressing the successor relation in a sequential program flow.

- **Algorithmic optimization** can be performed onto the graph. For example, loops can be unrolled or merged and the number of stages in a filter design can easily be changed.

- **Data/Controlflow analysis** examines data- and controlflow-dependencies between the CDFG nodes. For example, unused variables can be identified by static analysis techniques and data dependencies may be removed to relax the total amount of constraints.

- **Resource allocation** constrains the number of each functional unit available in the subsequent synthesis phases and to be used in the microarchitectural datapath.

- **Module selection** decides on different implementation alternatives of a functional unit (FU). For example, different types of digital adders exist such as ripple carry, carry-lookahead, or carry-save adder. Furthermore, each component may exist in different variants such as in a fast or small version in order to satisfy precise demands.

- **Scheduling** assigns each operation to a cycle satisfying all data- and controlflow dependencies as well as memory hold-times. Furthermore, advanced scheduling aspects such as multicycling and chaining can be supported. Prominent examples of scheduling algorithms are as soon as possible (ASAP)/as late as possible (ALAP), the class of list-based scheduling techniques, and force directed scheduling (FDS) [PK89]. In the FDS approach, a global time constraint is set and the required resources are minimized heuristically. In contrast, list scheduling techniques minimize the total execution time under a given hardware constraint.

- **Functional unit binding** assigns each functional operation to an operator that is capable to execute it. Thereby, a resource can be shared by mapping multiple operations with non-overlapping execution times to it. In literature, several binding algorithms have been proposed. For example, in [Kru01] a functional unit binding for a low dynamic power dissipation is proposed that is also capable to trade off different resource allocations.

- **Register binding** is the binding counterpart for registers. After scheduling and FU binding, the number of interim results for each cycle is fixed that need to be stored in registers. The register binding maps the temporary results to available registers. As in the FU binding, registers can also be shared.

- **Controller synthesis and output processing** describes the last activity in a HLS. Based on a fixed resource allocation, module selection, schedule, FU and register binding, a controller for the resulting microarchitectural datapath is created. It controls the registers' enable and necessary multiplexer select signals. Furthermore, it reads control-flow related data to decide on conditional executions. In the end, the timed datapath and controller is outputted for the subsequent logic synthesis.

Figure 2.13 illustrates a simple behavior being synthesized to an RTL datapath. For simplicity and with a focus on functional units in this thesis, a data-dominated and pure sequential design is shown, neglecting control-flow dependencies and register mapping.

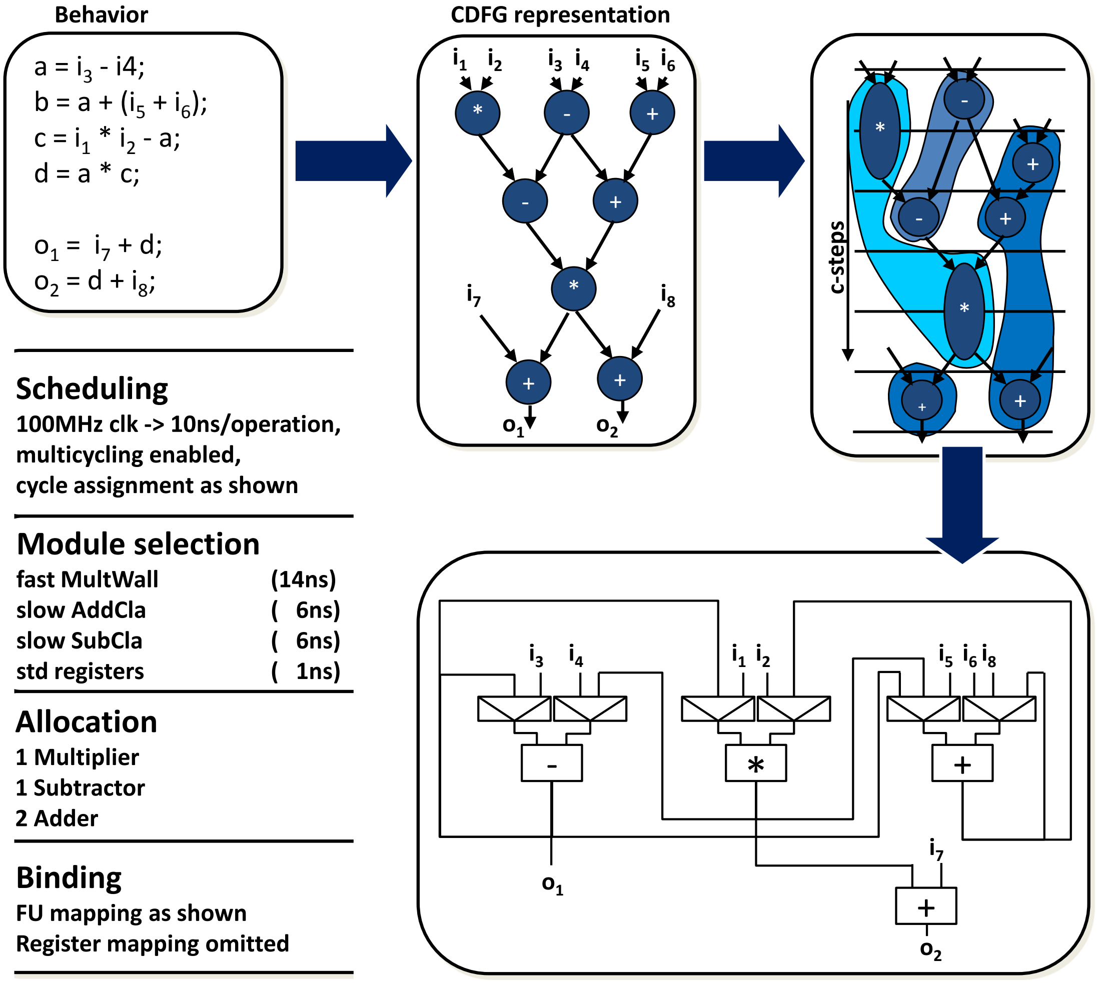


Figure 2.13: Exemplary high-level synthesis application

2.3.1 Summary

HLS can be summarized as the process of compiling a behavioral design description given in C/C++ or SystemC to a fully-timed microarchitectural datapath at RT-level described in Verilog or VHDL. The synthesis process covers a broad variety of decisions including scheduling, allocation, binding, and controller synthesis. It has its strengths in synthesizing data-dominated behavioral descriptions that mainly occur in the signal and image processing domain. These algorithms are widely distributed and can be characterized by a predominance of arithmetic operations.

3 Related Work

This chapter presents a focused and detailed cross-section on related work of the topics addressed in this thesis. Section 3.1 focuses on techniques and methods for modeling and estimation of power-gating techniques in research and scientific environment. Existing algorithms and methods for an automated leakage optimization during tasks of the high-level synthesis are considered in Section 3.2. The few existing toolflows being composed of industrial EDA tools with a consideration of the power-gating technique (in any fashion and level of abstraction) are presented in Section 3.3. The related work chapter closes with a presentation of the industrial power standards Common Power Format (CPF) and Unified Power Format (UPF) for a specification and an exchange of power-related meta information.

3.1 Modeling and Estimation for Power-Gating

In the following, existing works targeting an estimation of design- as well as runtime-characteristics of power-gated circuits are analyzed. Due to the diversity of facets, the following classification has been made for the literature search. At first sleep transistor sizing is considered as it is a main parameter for the runtime-behavior. Secondly, both steady states (on and off) are analyzed, followed by a closer literature research for estimating the transient state. At last, existing techniques for estimating overhead costs in power-gated designs are presented as they occur due to additional hardware as described in Section 2.2.3. This section closes with a short summary emphasizing the need for a holistic modeling approach.

3.1.1 Sleep Transistor Sizing

Sizing the channel width of a sleep transistor is a key issue of state-of-the-art coarse-grained MTCMOS circuits. Just as in a fine-grained application [KS04], the sizing trades off silicon area overhead, switching energy overhead, and performance. Additionally, it complicates timing estimation and affects signal integrity. The most pragmatic approach sums up transistor widths of a power-gated circuit and derives a sleep transistor size by a table lookup. This approach uses rule of thumb estimates and completely ignores dynamic voltage drops during operation and power-up caused by the sleep transistor. Thus, in the course of years more sophisticated sizing approaches have been introduced that all consider the delay degradation. They can be divided into two classes: *peak-current* and *average-current* driven approaches.

The general idea of peak-current driven approaches is to estimate the worst-case or maximum instantaneous current (I_{MIC}) through the sleep transistor due to switching activity in normal operation. Based on this current, the delay degradation is estimated and an appropriate transistor size is derived. [KCA97] introduces a gate-level simulator that dynamically adapts the gates delay on the base of the total number of simultaneously switching gates. This simulator then exhaustively searches for the input vectors leading to the worst-case currents. Of course, this approach cannot be applied for large RTL components with large input bitwidths. For this reason, the same authors propose an input-vector independent technique that first applies separate sleep transistors for each gate as it is done in a fine-grained application of power-gating but then merges the sleep transistors [KNC98, KC00]. This merging is based on the fact that not all gates in the circuit will switch simultaneously. Thus, they analyze mutual exclusive gate discharge patterns, merge sleep transistors accordingly, and can guarantee a performance level. Other possibilities to prevent an exhaustive input vector search in order to derive the maximum instantaneous current are to apply heuristics as summarized in [CCCY06] or to use static timing analysis as presented in [SCB+07].

The authors of [CCC09] tackle the problem of I_{MIC}-estimation in DSTNs in that multiple sleep transistors are shared via one virtual supply or ground line, balancing the I_{MIC} (see Section 2.2.2). Based on this I_{MIC} they apply an iterative greedy sizing approach to satisfy a maximum voltage drop constraint in DSTN designs. In [CCJC10] the same authors enhance their I_{MIC}-estimation by dividing the clock period into smaller time frames of variable lengths being more accurate.

Peak-current driven sleep transistor sizing approaches, as described so far, may be necessary to ensure signal integrity in critical designs and to prevent memory elements from loosing their state by falling below a certain operating voltage. But on the other hand, they are far too conservative since current peaks occur only for short compared to the time of a whole operation. The second class of approaches thus considers average currents occurring in an operational cycle. In [RZDP05, AAE03, WAA04] the authors propose sizing algorithms based on expected average currents that occur under a certain switching probability. The focus on the average case leads to significantly smaller sleep transistors.

The authors of [PP08] consider sleep transistor sizing as a delay budgeting problem. They distribute remaining timing slack to the sleep transistor of each gate row in a preplaced design. In contrast to the former techniques, they allow large temporary IR drops in single rows but they can guarantee an upper bound of delay penalty for the whole circuit.

3.1.2 On-State Estimation

On-state estimation for power-gating consists of two parts. At first the IR drop across the sleep transistor needs to be predicted. In a second step, this voltage drop can then be used to predict the on-state leakage currents through sleep transistor and circuit.

The resulting voltage drop across the sleep transistor $V^{ST}_{DROP\text{-}ON}$ can be computed by Ohm's law as $V^{ST}_{DROP\text{-}ON}(t) = R_{ST}(t) \cdot I(t)$ where $R_{ST}(t)$ is the sleep transistor resistance and $I(t)$ is the current through it. Most of today's works consider the sleep transistor resistance as a static value since it works in its linear region while the transistor conducts [KS04, RZDP05, CCCY06, PP08, CCJC10]. This assumption can be traced back to [KCA97] from 1997 and is also used in recent work of sizing the sleep transistor.

Leakage estimation techniques that have been developed for custom circuits also hold for power-gateable circuits being in the active state if the reduced supply voltage is taken into account. In the following, only a short overview on existing leakage modeling techniques is given. Since the focus in this thesis is on power-gating of RTL components, only leakage modeling techniques at this level will be summarized. An exhaustive presentation of leakage modeling techniques can be found in [Hel09].

The authors of [BS00, ZPS+03] propose complexity based models. To derive an RTL estimate they scale the leakage current per device with the number of devices and a design complexity metric. This metric is specific for a circuit topology and "accounts for effects like transistor sizing, transistor stacking and the number and relationship of NMOS and PMOS transistors in a circuit". Thus, it abstracts from dozens of device parameters, roughly covers data dependency, but, as a main drawback of this top-down approach, it is determined empirically.

In [LHB$^+$05] the authors consider power-gating and propose a data dependent leakage model to get upper- and lower-bound leakage estimates as well as the responsible input vectors. Based on these upper- and lower-bound currents, the min/max leakage saving can be computed and statements on the efficiency of power-gating are made. In their approach they apply a genetic algorithm to find the vectors for existing circuits and analyze the leakage currents by circuit simulations. The average leakage current per gate is then denoted as I_{leak}^{avg} and a leakage power estimate can be computed as shown in Equation 3.1 by scaling with the number of gates. Further parameters like the temperature dependency are not considered.

$$P_{leak} = \#_{gates} \cdot I_{leak}^{avg} \cdot V_{DD} \tag{3.1}$$

Another bottom-up leakage modeling technique has been proposed in [HHN06, HEN06, Hel09]. The model consists of four layers. At first, reference transistor models are built for NMOS and PMOS devices in conducting and non-conducting state. They capture technology dependency on static parameters such as channel length or oxide thickness as well as the dependence on dynamic parameters such as supply voltage, body voltage, and temperature. Secondly, at gate level, a state dependent characterization is performed and parameters are derived to fit each gate and state to a linear combination of the reference models. The third layer then performs RT-level zero-delay simulations in order to get the state of each gate within a RT component netlist for a given input vector. Again the lower-level models are scaled accordingly, leading to an RT hard-macro. In the last modeling stage the size/bitwidth and data dependency are abstracted to the final RT soft-macro model.

3.1.3 Off-State Estimation

Comparable to the on-state, the off-state estimation is also two-folded: the virtual supply/ground voltage needs to be estimated and the remaining leakage current is also of interest. The former $VV_{DD}/VGND$ defines the starting point for a subsequent state transition and impacts the wake-up time quadratically [SASN07]. Furthermore, the saturated voltage level is an important variable in order to estimate the remaining leakage current and to compute the effectiveness of leakage power reduction.

Simple off-state estimation approaches assume the virtual supply/ground voltage to be saturated to the steady state immediately after power down [SASN07, KKK$^+$07]. Although these approaches are pessimistic and neglect a potential saving in power-up energy of a recently power-gated circuit, they define a lower/upper bound for $VV_{DD}/VGND$ and thus a safe upper bound for the state transition energy. The authors of [XVJ08] model the virtual supply/ground voltage over time and thus the time since powering down is a model parameter. Their model is built bottom-up from transistor to RT-level and predicts $VV_{DD}/VGND$ with a maximum error below 4%.

Remaining leakage currents I_{OFF} in the off-state are readily neglected in existing works and $VV_{DD}/VGND$ are assumed to be equal to GND/V_{DD} [JMSN05]. But in fact, $VV_{DD}/VGND$ saturate at a level with identical currents through sleep transistor and gated circuit (neglecting minor gate-leakage currents through the circuits in- and output pins). In [SHSB07] the effect of a non-zero standby current is considered the first time. The focus in this work is on subthreshold operation where the ratio of on/off leakage is much higher compared to a regular operation.

Leakage currents of voltage anchors and of buffer chains are not considered in recent work.

3.1.4 Power-Up Current and Energy Estimation

The maximum instantaneous current I_{MIC} is used for sleep transistor sizing as described in Section 3.1.1. Another important peak current, denoted as maximum power-up current I_{MPC}, occurs while the transition from sleep to active state is taken. In contrast to the normal operation in active state, all gate capacitances are charged simultaneously because all gates are powered up by one PGS. For PMOS power-gated circuits the I_{MPC} is drawn from the supply and flows into the circuit, whereas NMOS power-gated circuits are at a high voltage level during sleep and the I_{MPC} flows from the circuit to ground. Obviously, the power-up current is limited by I_{ON} of the sleep devices but as its channel width can become very large, I_{ON} is just a pessimistic upper bound. A consideration of this current is necessary for two reasons: It affects the power-on time and may cause circuit failures, timing errors, or logic malfunctions e.g. caused by electromigration.

A maximum power-up current estimation is first addressed in [LH01, LHS02]. In contrast to the I_{MIC}, the authors emphasize this current as one-vector dependent because it only depends on the input vector defining the state after wake-up. In this approach, it is assumed that for PMOS gating the charging current is maximal if and only if the maximum number of gate outputs is high because then the most internal capacitances have to be loaded. Thus, they apply different automated test pattern generation (ATPG) techniques that implement greedy algorithms in order to find costly input vectors. A simulative validation showed up to 87% larger I_{MPC} currents compared to the corresponding I_{MIC}.

A drawback of the assumed proportionality between logical high gate outputs and power-up current is the negligence of hazards as they significantly contribute to the power-up current. For this reason, another simulative approach has been proposed in [LXHL03] that includes hazards into the cost function and applies a genetic algorithm in order to find worse input vectors. Beside these heuristic approaches the problem in finding the worst input vector has been formulated as integer linear program (ILP) in [SA06].

In a continued work, the authors of [LH01, LHS02] abandon the simulative approach and raise the level of abstraction in order to use the power-up current estimation during high-level synthesis [LHB$^+$04, LHB$^+$05]. They pre-characterize the I_{MPC} of each cell in the library by Spice simulations and calculate an averaged cell power-up current I_{MPC_avg} for existing designs by regression techniques. They further estimate the gate count $\#_{gates}$ of a power-gated circuit c and derive the overall I_{MPC} as shown in Equation 3.2.

$$I_{MPC}(c) = \#_{gates}(c) \cdot I_{MPC\_avg} \tag{3.2}$$

Compared to simulative and time-consuming gate-level power-up investigations, this approach has been evaluated with an average error of 21%.

With the knowledge of the I_{MPC}, concepts for minimizing the ground bounce have been proposed. For example, in [KKK03, SASN07] sleep transistors are turned on in a stepwise manner or [RDP07] proposes a mixed integer linear program (MILP) formulation for a ground bounce minimizing wake-up scheduling. All these optimization approaches trade off peak current versus power-on time.

Regarding the energy E_{SW}^{RT} of transitions from off- to on-state, existing modeling approaches base on the assumption that in average half of the capacitances within a power-gated circuit are charged during wake-up [JMSN05, HBS+04]. Thus, these approaches perform a static analysis of a circuit. Incomplete transitions or hazards during wake-up are neglected in these approaches.

3.1.5 Additional Overhead Cost Estimation

Beside the pure gating circuitry, additional overhead costs in terms of area and power occur due to interfacing circuits. As described in Section 2.2.3, buffers are used to amplify the sleep signal and voltage anchors are needed to isolate the outputs of a power-gated circuit. In literature, these overhead costs are neglected generally. For example, in [JMSN05] the authors describe the first system-level tradeoff of sleep-transistor-based power-gating techniques. Most of the costs in area, performance and power caused by the gating circuitry are analyzed and listed but interfacing components are not mentioned at all.

Since the fan-out of each buffer in the chain grows with a specific rate $e \approx 2,718$ as shown in Figure 2.9 (see Section 2.2.3 for a further explanation), the total transistor width in a 2-stage buffer chain, driving a sleep transistor with a width of W_{ST}, can be approximated as shown in Equation 3.3.

$$W_{buffer} = (\frac{2}{e} + \frac{2}{e^2})W_{ST} \approx 1.006 \cdot W_{ST} \tag{3.3}$$

As it can be seen, the area and thus the additional dynamic power and leakage currents are in the same order of magnitude in comparison to the sleep transistor. Beside the power and area, buffers further introduce a signal delay on top of the wake-up delay that needs to be considered.

Voltage anchors consist of 6 to 14 transistors (see Section 2.2.3) and for each output of an RT-level component a separate one is needed. A $4bit$ adder, synthesized with a $45nm$ industrial technology consists of 130 transistors and thus the overhead for voltage anchors is 18% to 43% in transistor count. [BBMM06] presents further details on the area and power overhead.

For these reasons, overhead costs due to interfacing circuits have to be taken into account during estimation and appropriate models have to be developed.

3.1.6 Summary

None of the aforementioned approaches of **sleep transistor sizing** is suitable for an application during high-level synthesis as addressed in this thesis. They all assume a gate level circuit description in order to derive discharge current traces or even assume placed logic and sleep transistor cells. Secondly, they do not allow design tradeoffs during synthesis.

One major inaccuracy in existing **on-state estimation** approaches is the approximation of the sleep transistor as a fixed resistant R_{ST}. Simple simulations using Synopsys HSPICE® indicate that this simplification leads to errors of 20% in comparison to the real voltage drop using transistor technologies of $90nm$ and below. As a consequence, the estimation will become inaccurate and it will even worsen the sizing of the sleep transistor if this is based on a faulty assumption. For leakage estimation during on-state the proposed models of [HHN06, HEN06, Hel09] are used in this thesis because they cover all necessary dynamic parameters and are available inhouse.

Adequate $VV_{DD}/VGND$ estimation techniques for characterizing the **off-state** exist. Additional effort has to be spent in order to estimate the remaining leakage currents because existing leakage models [HHN06, HEN06, Hel09] are only made for a certain supply voltage range and are not applicable for remaining supply voltages close to zero.

In scientific works on **power-up energy estimation** incomplete and spurious transitions during startup are neglected since they only base on the static circuit's internal capacitances. Synopsys HSPICE$^{®}$ circuit simulations show that the resulting error becomes 50% for a $4bit$ adder and even more than 100% for an $8bit$ multiplier. For this reason, a more accurate model needs to be developed.

Interfacing circuits are neglected in recent work of power-gating but may introduce significant power and area overhead.

As a result of the above mentioned reasons, a holistic modeling of sleep-transistor-based power-gated RT-components that could be used in automated EDA tool support for estimation and optimization is still not available and is thus presented in the remainder of this work.

3.2 Leakage-Management in High-Level Synthesis

Power optimizations in high-level synthesis mostly focus on dynamic power by minimizing switching activity during scheduling, binding, and allocation. [MDAM96, LRJD98] propose scheduling and constrained register sharing methodologies to force dynamic power savings by minimizing spurious switching activity (SSA). A sophisticated binding and allocation approach is presented in [KSJ^{+}99]. It considers data-dependencies, trades off different allocations but it ignores leakage power and dynamic power-management. With gaining importance of leakage currents, counteracting low-level techniques have been developed as described in Section 2.1.3. In the following, methods are summarized that apply these general techniques during the high-level synthesis and improve the HLS by including leakage currents into optimizing cost functions.

Using low- and high-V_{TH} MOSFET devices jointly in one circuit, biasing the body voltage with a fixed voltage as well as using multiple voltage islands are the most important static optimization techniques. [SP99, KJ00] propose leakage optimization techniques using multiple-threshold CMOS technology. Circuit parts (at gate and RT level) that idle most of the time are implemented in channel-leakage saving high-threshold transistors. Performance critical modules within the critical path are implemented in fast low-threshold transistors. [MKP08] simultaneously performs operation scheduling and resource binding using devices of different oxide thicknesses and thus has its focus on gate-leakage currents. The authors of [HMHN07] statically partition RTL-components into islands during synthesis and tune their supply and body voltage to exploit available register-to-register slacks defined by the clock cycle. In [NM06], the authors show that an allocation with the lowest number of resources is not the best allocation in all cases due to the electro-thermal coupling effect. They propose a leakage minimizing resource allocation that is thermally induced by distributing activity over a higher number of resources.

[HC09] introduces power-gating into the HLS and further assumes variable clock skews for registers.

Based on the resulting maximum allowable delays of functional units between the registers the authors size the sleep transistors in order to exploit the slack. Given a target clock cycle length the objective in their work is to schedule clock arrival times of registers and to find an FU resource binding, leading to a minimized standby leakage current. In contrast to the work of this thesis, they do not optimize the HLS for continuous idle times necessary for applying cycle-based power-gating. Further, all of the aforementioned approaches are of static nature.

Synthesis optimizations supporting dynamic leakage-management techniques have been proposed by Katkoori et al. for the first time. In [GK02, GK03b] they propose a resource allocation and binding algorithm based on clique partitioning. It optimizes a datapath for largest possible idle times between operations and thus reduces the amount of transitions between idle and active periods. Thereby, resource allocation bases on the minimal amount of necessary resources obtained by the scheduler and thus tradeoffs are not considered. Furthermore, the optimization approach abstracts from transition costs in terms of dynamic energy, transition delay, and real break-even times by expecting user-defined weights defining their cost function. Additionally, the binding and allocation is independent on data pattern and thus the dynamic power consumption is ignored completely. As scheduling algorithms they use an as-soon-as-possible (ASAP), as-late-as-possible (ALAP), and a force directed (FDS) scheduler [PK89]. In all cases, the scheduler does not optimize for operation clustering and thus the overall results are suboptimal. In [GK03a] and [GK04] the authors extend their leakage-aware high-level synthesis to select MTCMOS resources partially and to include scheduling into optimization using simulated annealing. Again, they do not regard allocation restrictions and ignore transition costs. Dynamic power-management is not applied on the datapath and an optimization for large idle times is not a topic of this work. Another leakage-driven behavioral synthesis approach is described in [GGK05] but it is limited to functionally pipelined datapaths.

The authors of [CSKS08] propose a high-level synthesis framework called HLS-pg, which optimizes the schedule in order to reduce the number of retention registers. When power-gating is applied, these registers become necessary to hold variables across cycle borders. Optimization for FUs is not addressed in their work.

The most recent and, at first sight, the most similar work compared to the proposed work in this thesis is presented in [DM09]. The authors target an optimization of the HLS for power-gating by powering down FUs for longest possible idle times. Their optimization criterion is to cluster active periods of FUs in order to minimize the power up/down frequency exactly as it is targeted in this thesis. At a closer look a lot of differences become visible. At first, a conventional list scheduling algorithm is used without power consideration. It is purely area driven and the minimal number of resources for shortest schedule is determined. The binding is thus restricted to the minimal number of resources and no allocation tradeoffs are possible (refer to [NM06] for the scope of this limitation). Components' bitwidths are also not reflected during binding. As a consequence an $8bit$ operation might be bound to a $32bit$ component introducing dynamic as well as static power overheads. In general, the dynamic power is neglected at all because the overall proposed HLS is performed data independent. As a further result, no data correlation is considered during binding, leading to inaccurate power estimates. In contrast to the individual component based implementation of power-gating, the authors of [DM09] further cluster the components to larger islands. This is less effective in reducing the power but positively limits the sleep signal distribution and controller logic overhead. Another significant inaccuracy is the negligence of break-even times. The authors assume components to be able to power-gate and wake up within one cycle and with overhead costs completely compensated by leakage savings in one cycle.

Up to now, the work presented in this thesis is the first on exploiting power modes of functional RT-units to optimize scheduling, binding, and allocation in order to reduce both, dynamic and static power.

3.3 Industrial Toolflows for Power-Gating Estimation

Power-down techniques addressed in this work are already used in some commercial chips, for example, to power down idling cores in recent multi-core processors. These cases of application are limited to manually optimized, general purpose processors. Tool support for the automated handling of leakage reduction techniques is not yet available (see Section 1.2). Emerging industrial toolflows to consider power-down techniques in ASICs already during design are presented and analyzed in the following.

Figure 3.1 illustrates a power-gating estimation-flow from RTL to graphical design station II (GDSII) that bases on a $90nm$ process of Semiconductor Manufacturing International Corporation (SMIC) [Wan08]. The flow entirely adopts tools of Synopsys® and consists of five steps. First, a power-simulation is done in Synopsys® VCS where the power-gating behavior is observed and verified in terms of correct control signals, data retention, and isolation. In the second step, the synthesizable RT-level design is enriched by power information as described in Section 3.4 and passed to the Synopsys® Design Compiler. Beside the classical RTL synthesis, the tool further maps retention registers and isolation cells to elements of the library. Floorplanning is the third step of the proposed flow. Header or footer cells are inserted, sized by ”rule of thumb” [Wan08] and placed to create voltage areas. Additionally, the (virtual) supply and ground network is planned and analyzed. This analysis gives first and rough estimates of IR drops and peak currents. Floorplanning as well as the following physical optimization are done by Synopsys® IC compiler. Output of the optimization step is a tapeout-ready GDSII design. In a last step, predictions on the efficiency of the overall power-gating implementation, IR drops, power-up rush currents and wake-up times can be done. Unless composite current source (CCS) power models are available for each standard, filler, and power-management cell, they have to be characterized preliminarily by Synopsys® PrimeRail using a Synopsys HSPICE® backend. This cell-characterization is necessary because standard .lib liberty files do not describe transient behavior of the driver waveform into a load. Synopsys® PrimeTime-PX in combination with PrimeRail then allows power analysis as wanted.

A second industrial effort of covering power-gating aspects with existing tools has been proposed by Sequence Design Inc.® (recently acquired by Apache Design Solutions®) and adopts their CoolPower® and CoolTime® solutions [FV07, CF07]. CoolPower® replaces non-MTCMOS cells of an existing netlist and placement by its MTCMOS derivate and inserts additional switches. CoolTime® is used for a power grid analysis in order to derive rush currents as well as wake-up time estimates. The overall flow outputs an updated netlist. Further considerations, such as an automated integration or pre-RTL analysis, are not proposed.

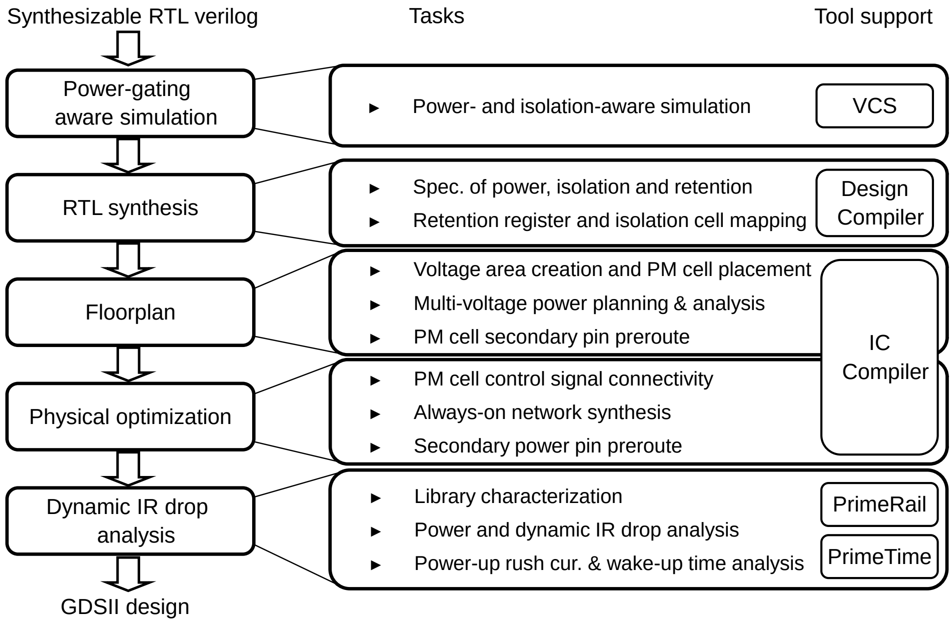

Synthesizable RTL verilog
Tasks
Tool support
Power-gating aware simulation
Power- and isolation-aware simulation
VCS
RTL synthesis
Spec. of power, isolation and retention
Retention register and isolation cell mapping
Design Compiler
Floorplan
Voltage area creation and PM cell placement
Multi-voltage power planning & analysis
PM cell secondary pin preroute
IC Compiler
Physical optimization
PM cell control signal connectivity
Always-on network synthesis
Secondary power pin preroute
Dynamic IR drop analysis
Library characterization
Power and dynamic IR drop analysis
Power-up rush cur. & wake-up time analysis
PrimeRail
PrimeTime
GDSII design

Figure 3.1: Synopsys power-gating design methodology [Wan08]

3.4 Industrial Power Standards

Two coexistent standards, the Unified Power Format (UPF 1.0, 2007) by Accellera [Mag07] and the Common Power Format (CPF 1.0, 2007) by the Silicon Integration Initiative (Si2) [Sil08] have been established in the recent past. Both standards allow a user to define (or a design tool to generate) script-based files describing the power-intent for electronic systems and electronic intellectual property. For example, they cover the definition of *static* design elements (referred to as ”design objects”) to describe hierarchies in a design, voltage domains, and power isolation domains as well as its interconnects with pads, pins, ports, and nets. Special library cells such as always-on-, isolation-, level-shifter-, power-clamp-, power-switch-, as well as state-retention-cells are available to describe advanced power-management techniques. CPF- and UPF-objects are then used to describe the *dynamic* capabilities and behavior of a design. For example, power modes and corresponding rules to change modes, power down an island, or retain its state can be defined.

Since the initial 1.0 releases of CPF and UPF, different dialects and enhancements have been released: CPF 1.0e, CPF 1.1, UPF 2.0. The latter UPF 2.0 also achieved formal standardization as IEEE 1801, ”Standard for Design and Verification of Low Power Integrated Circuits” [Uni09].

Both standards support various phases of the design including RTL, post-synthesis, and post-routing. The roadmap for future updates also indicates power-estimation capabilities, updates required to drive power optimizations, and system-level support. But although there is a large correspondence between CPF and UPF, and undertakings are done to build a converter [Ope], both standards are coexistent and tool-support varies from tool to tool.

3.5 Summary

In summary, a wide spectrum of work has been done for **modeling** the power-gating technique in order to **estimate** all relevant power issues, **optimizing** its application during high-level synthesis, and transferring design decisions to subsequent design tools. But none of the aforementioned work covers the holistic integration as it is the scope of this thesis.

Existing modeling approaches oversimplify relations between sleep transistors and their impact on different power modes. Overhead costs for state transition are handled stepmotherly and important parameters for leakage estimation such as temperature are neglected at all in the dominant majority of existing work. Synthesis considerations base on inaccurate models, are limited to single phases of the HLS, and are not aware of the overall power consumption being composed of static and dynamic power. Existing industrial toolflows for power-gating are at a very beginning. They do not allow design-space explorations, rely on empirical knowledge of experts, and require post-RTL analyses.

Today's **industrial tool support** for an automated integration of power-gating lacks in various aspects. The drawbacks directly become apparent when the needs of a designer, as presented in Section 1.2, are considered. The first limitation is that important and interesting design parameters have to be fixed inputs. Especially, power-ating granularity decisions as well as runtime-dynamics have to be fixed at RT-level without any pre-RTL analysis. The second limitation is the restriction to pure estimation flows. They have no exploration loop to trade off design alternatives. Even if the loop is done manually, high-level optimizations for power-gating are not possible due to a missing investigation of power-on energies and break-even times. Additionally, the flows are time-consuming and difficult to set up caused by its low level of abstraction.

Regarding the existing **power standards**, the flow in this thesis makes sure to support the industrial standards of Section 3.4 in order to be compliant with existing tools for subsequent design phases. In detail, power-aware design information will be provided out of the high-level synthesis as indicated in Figure 1.2 to reflect power-gateable domains in the supply grid, sleep signals, and the controllers interconnect.

4 Modeling and Estimation Flow for Power-Gating

Modeling the dominant effects of RT-level components under power-gating in order to get fast and accurate estimates and to explore the design-space of the HLS is one of the main contributions in this thesis. Figure 4.1 gives a very simplified overview on the modeling and estimation flow that will be further described in this chapter.

In the following, the characterization environment is described first, naming all tools and prerequisites to the modeling flow such as necessary technology files and formats. Secondly, the models, subdivided into the different working states of an RTL-component, are described in detail, motivating all model parameters and including all techniques that are applied to compact and ease the models.

Afterwards, the estimation flow is described in detail. Its main purpose is to get accurate estimates for a few main variables: leakage currents in the static on- and off-state, energy and timing overheads due to the state transition, and the break-even time. These values are obtained for each individual RTL component within the design and are then used beside the precise parameter values and activity patterns to get an estimation for its overall energy consumption.

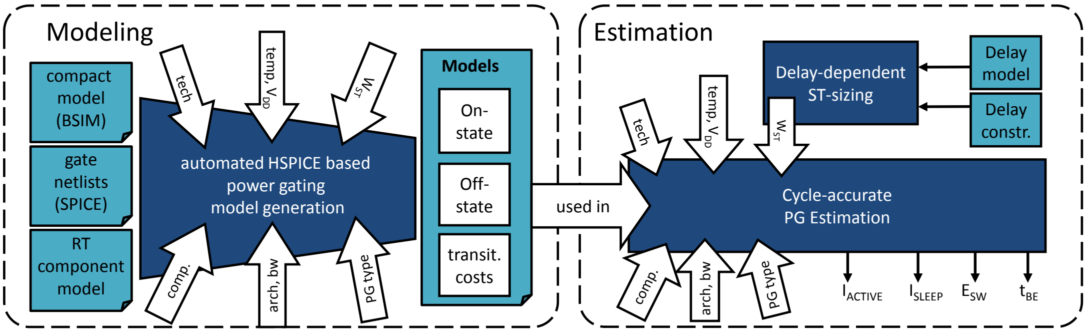


Figure 4.1: Overview on the modeling and estimation flow

4.1 Model Characterization Environment

The overall toolflow of the model characterization environment is presented in Figure 4.2. It combines different point-tools at various levels of abstraction in order to create models, satisfying the demands of their application in the design-space exploration (DSE) of Chapter 5.

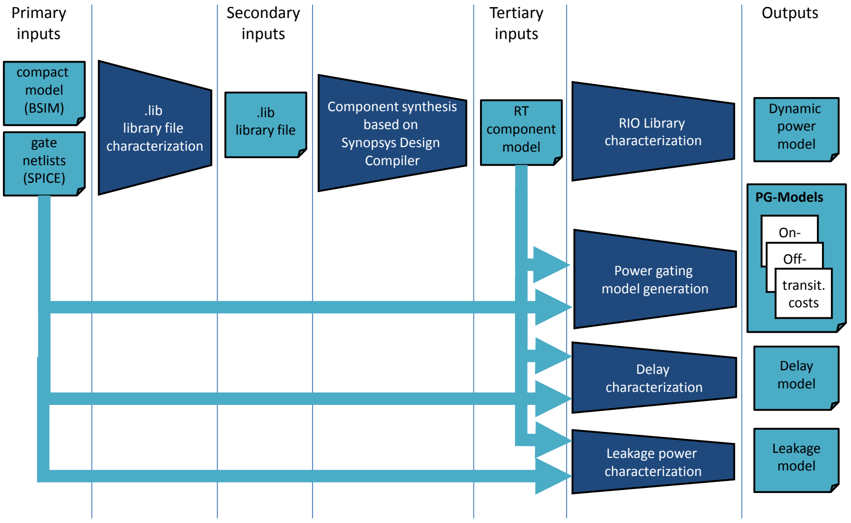


Figure 4.2: Overview on the model characterization environment

Primary inputs to the overall flow are semiconductor- and technology-specific files. The transistor modelcards of a semiconductor technology define all device model parameters such as doping concentration and geometry parameters and are specified in a BSIM-compatible format. Circuit simulators, like Spice, implement BSIM transistor models, read the transistor modelcards, and base their simulation on them. In addition to the modelcards, the proposed flow needs gate netlists describing all standard cells that are available in the considered technology. Thereby, the sizing of each transistor is of importance as each logic gate is described in different driving strengths. Beside the wired transistors, the gate description also includes decoupling capacitances. Again, the gate netlists need to be described in a Spice-compatible format.

Out of these primary inputs, a secondary input is built if it is not already available: the liberty formatted library file. This is done using a library characterizer like the Nangate Library Characterizer®. A liberty file (.lib) is an industry standard and contains a textual representation of timing and power information for each standard cell in a semiconductor technology. It is obtained by Spice-based simulations of the cells under various conditions such as temperature and input data.

The .lib library-file is then used as an input to synthesize a set of RTL-components using the Synopsys Design Compiler®. The outcome is a set of Verilog files describing a gate-level netlist for each RTL-component. These components now define the atomic blocks in the HLS. For each of them, a corresponding dynamic power model is created using the RIO library characterization engine of PowerOpt®. Additionally, these component netlists serve as input to the power-gating, leakage, and delay modeling tools. Since the latter all base on Synopsys HSPICE® simulations, an additional conversion is necessary to translate the Verilog-netlists to Spice-netlists.

The overall outcome that is used within the HLS then consists of four classes of models: the dynamic power models, power-gating models, delay models, and leakage models. The power-gating model characterization is one main contribution of this thesis and is described in Section 4.2. Nevertheless, a consistent model characterization for a certain semiconductor technology is important. For this reason and for the sake of completeness, a complete model overview including links to related work will be given in the following.

4.2 Power-Gateable Functional Units

Because of different characteristics of power-gateable functional units in each power-state, the following analysis is initially state-driven and differs between the on-state with conducting sleep transistor, the off-state with locking sleep transistor, and the transition from off to on. However, the subsequent model characterization is effect-driven, because the model characterization follows the idea of splitting the modeling of different effects. Thus, separated sub-models are created for the effects that have been identified in Section 2.2 to be necessary for a holistic estimation flow of power-gated designs. For example, dedicated models will be created for sleep transistor and buffer energy during state transition. The model generation has initially been proposed in [Ros06], been published in [RHN07], and extended to the final models of this thesis.

The model generation for each of these effects is done using the analog circuit simulation tool Synopsys HSPICE® at circuit level, adopts techniques for abstraction, and makes the models applicable at RT-level. Additionally, assumptions are made to simplify the models and to reduce the number of parameters. A detailed presentation of all sub-models and its parameters is given in the following. Afterwards, the sub-model characterization is presented.

4.2.1 Sub-Model Description

Based on the analysis of Section 2.2, a power-gateable FU consists of the functional RTL component itself (e.g. adder of multiplier), a sleep transistor (e.g. single PMOS or NMOS device), a buffer chain in front of it (e.g. a two-stage tapered buffer), and a voltage anchor for each output.

While being in the active state, the RTL component is working and a power model is necessary for predicting its dynamic power. The other circuit parts do not perform any operation at this time because the *sleep* signal is fixed. Beside the dynamic power, leakage occurs in each of the circuit parts, raising the demand for appropriate models. Beside power considerations, the voltage drop across the conducting sleep transistor needs to be known for determining the virtual supply voltage as described in Section 2.2. The problem is that the voltage drop depends on the actual current through the circuit and this current flow depends on the virtual supply voltage in turn. To avoid a complete new characterization of each RTL component with the sleep transistor size as parameter, the coupling is split up and a separated voltage drop model is proposed. In the estimation phase, both models are then combined and the operating point can be determined. At last, the component's delay is of interest. In summary, the following sub-models are necessary to fully cover the static active state:

- Dynamic power of an RTL-component (P_{DYN}^{RT}) due to activity,
- Leakage currents through an RTL-component (I^{RT}),
- Leakage currents of a conducting power-gating scheme (I_{ON}^{ST}),

- Leakage currents of a buffer (I^{BUF}),
- Leakage currents of a voltage anchor (I^{VA}),
- Voltage drop across a conducting power-gating scheme ($V^{ST}_{DROP\text{-}ON}$),
- Delay of an RTL component (D^{RT}).

In the static off state, the voltage drop across the sleep transistor is again of interest because it impacts the remaining leakage currents and the state transition energy. Buffer chain and voltage anchors are not affected from power-gating and continue to leak as in the on state. Thus, only two additional sub-models are required to fully cover the static sleep state:

- Voltage drop across a locking power-gating scheme ($V^{ST}_{DROP\text{-}OFF}$),
- Leakage currents of a locking power-gating scheme (I^{ST}_{OFF}).

The off-on switchover is initiated by an edge of the $sleep$ signal. It propagates through the buffer chain, opens the sleep transistor channel after clamping the FU's outputs with voltage anchors, and awakes the RTL component. In all parts a certain amount of energy is dissipated, significantly impacting the break-even time of a worthwhile application of power-gating. The overall wake-up time t_{wakeup} is composed of the buffer chain delay D^{BUF}, sleep transistor switching time D^{ST}, and the RTL component wake-up time t^{RT}_{wakeup}. Voltage anchors are connected in parallel to the RTL components outputs and thus do only marginally (about 1% in average and 2.5% in worst-case [BBMM06]) contribute to a delay and wake-up time increase. In total, the following seven sub-models are required to accurately estimate the state transition:

- Energy dissipation of a power-gating scheme (E^{ST}_{SW}),
- Energy dissipation of a buffer (E^{BUF}_{SW}),
- Energy dissipation of a voltage anchor (E^{VA}_{SW}),
- Energy dissipation of an RTL component state transition (E^{RT}_{SW}),
- Delay of a buffer (D^{BUF}),
- Sleep transistor switching time (D^{ST}),
- RTL component wake-up time (t^{RT}_{wakeup}).

Thus, a holistic estimation requires the aforementioned bunch of sub-models. In this thesis not all of the models are introduced or built from scratch. In fact, some of the sub-models come with the PowerOpt® component database library or are taken from literature. A presentation and a justification for the selection is given in the prerequisites Section 4.3.1 of the model application and estimation flow description.

As a consequence, the list of missing sub-models reduces to two leakage models, two voltage drop models, three state transition energy models and two delay models. Figure 4.3 summarizes all necessary models with the dependent input variables, whereas each sub-model is represented as a block. Obviously, the parameters and their ranges differ between the models and are carefully selected to provide accurate estimates. A detailed description of the parameter ranges is given in Section 6.1.

In the subsequent sections, the sub-models are quantified, the leakage-, voltage drop-, state transition energy-, and delay-models are presented, and the model parameters are explained in detail. For simplicity, the sub-models are referenced by their short model name. Furthermore, to maintain readability in equations, not all model parameters are named when the models are used. For example, I_{ON}^{ST} refers to the leakage current through a conducting PGS for a certain functional unit instead of terming $I_{ON}^{ST}(fu, PGS, W_{ST}, T, V_{DD})$.

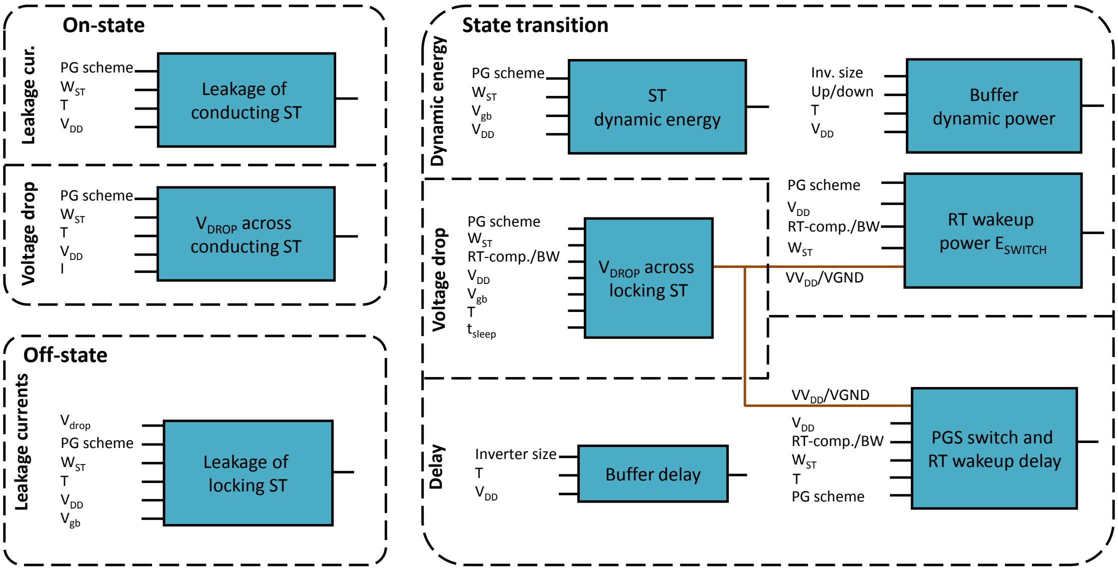


Figure 4.3: Sub-model and parameter overview

4.2.2 Analysis of Power-Gating Circuitry Overhead Costs

All of the mentioned sub-models are of varying importance for an overall assessment. In this section, exemplary measurements are shown in order to identify the dominant costs. Thereby, the main focus is on the relative composition of the state transition energy E_{SW}, idle leakage currents in the active state I_{ACTIVE}, and remaining leakage currents I_{SLEEP} during sleep state.

Figure 4.4 presents an absolute and relative composition of the state transition energy for an exemplary incrementer component in different bitwidths and for a broad spectrum of sleep transistor sizes. The incrementer is power-gated with an HVT PMOS sleep transistor and operates at a nominal voltage of $1.0V$. The total state transition energy is composed of E_{SW}^{RT} and $E_{OVERHEAD}$ which in turn consists of E_{SW}^{ST} and E_{SW}^{BUF}. As it can clearly be seen, the relative amount of buffer and PGS state transition energy increases with the sleep transistor size W_{ST}. This is because small sleep transistor widths only require a one-stage buffer chain, while the $32bit$ component with a relative W_{ST} size of 10% requires a four-stage

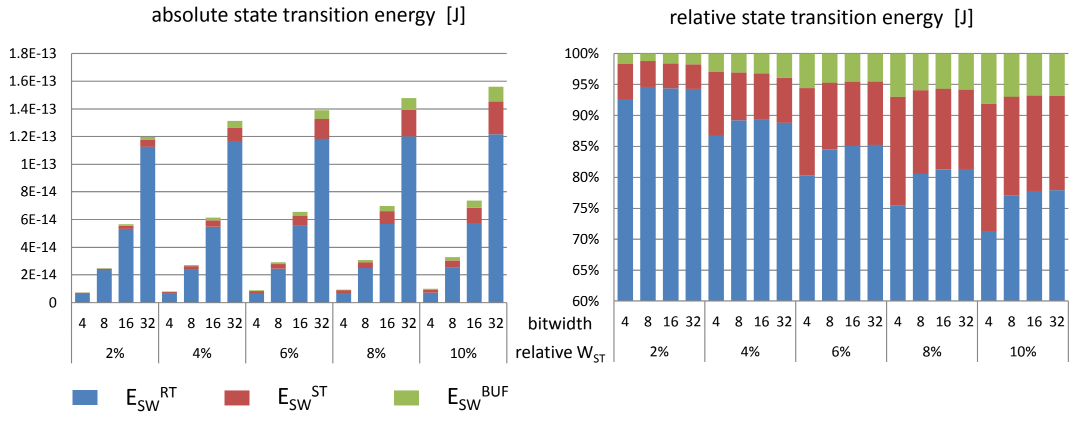


Figure 4.4: Composition of the state transition energy of an incrementer component

chain. The results show that with rising component size, E_{SW}^{ST} and E_{SW}^{BUF} account for up to 28% of the total state transition energy.

Next, the active state leakage current I_{ACTIVE} is analyzed on the base of the same component. It is composed of the current through the RTL component I^{RT}, the buffer chain leakage I^{BUF}, and the sleep device gate leakage. As shown in Figure 4.5, I^{RT} is independent on the buffer and dominates I_{ACTIVE}, although I^{BUF} increases up to 16% for large sleep devices. Gate leakage only plays a subordinated role in these examples, as it is three orders of magnitude smaller compared to sub-threshold leakage in the used Nangate free $45nm$ open source digital cell library technology. Nevertheless, I_{ON}^{ST} can be significantly larger in other technologies and is not omitted during modeling.

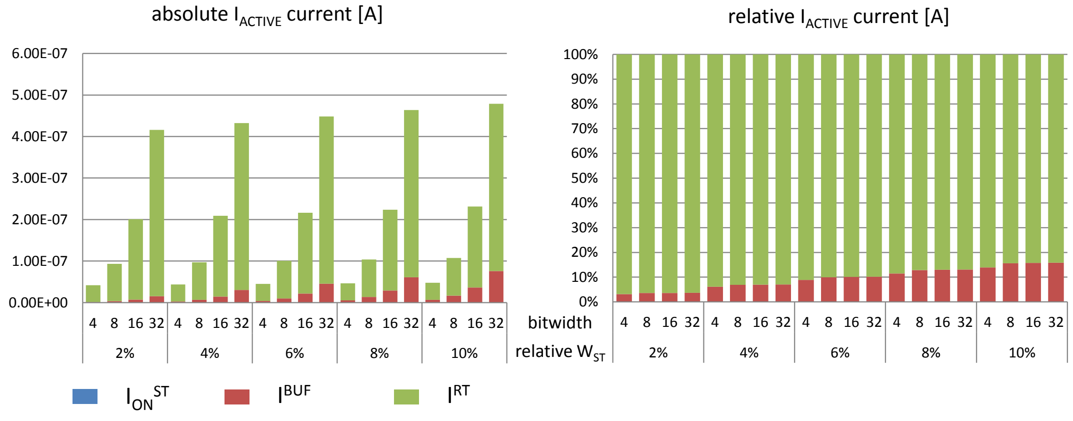


Figure 4.5: Composition of the leakage currents in the active state I_{ACTIVE}

Figure 4.6 presents the remaining leakage currents in the sleep state. It can be seen that I_{SLEEP} splits into I_{OFF}^{ST} and I^{BUF} at a relatively constant ratio of about 60-to-40. If a less leaky NMOS sleep device would have been used the ratio would be reverse.

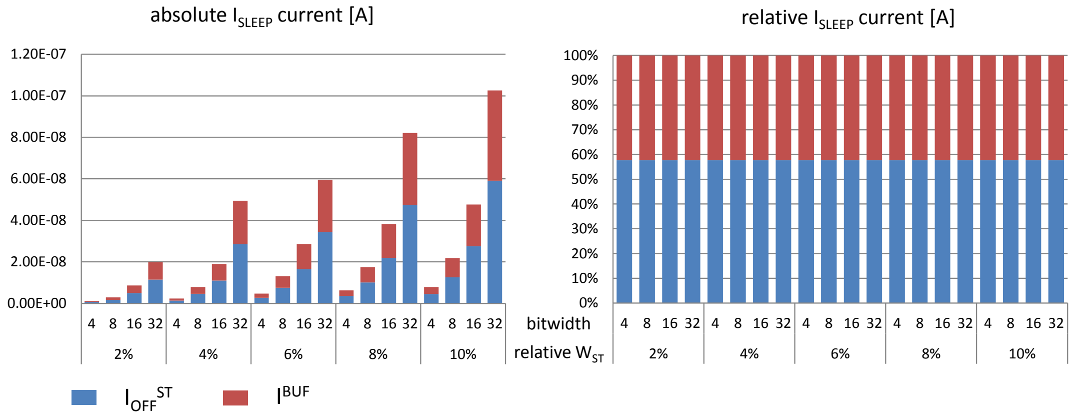


Figure 4.6: Composition of the remaining I_{SLEEP} leakage currents of an inverter component

4.2.3 Power-Gating Scheme Leakage Models

I_{OFF}^{ST} majorly consists of subthreshold- and gate-leakage and counteracts possible savings while sleeping. I_{ON}^{ST} introduces overhead costs of the gating circuitry. Because of the open channel, this current is dominated by gate-leakage currents.

The important dynamic parameters influencing the leakage currents through the gating circuit are the supply voltage, temperature, voltage drop, and sleep transistor width as it has been discussed in Sections 2.1 and 3.1.3. To support super cutoff techniques [KNS00] using gate-voltages higher than V_{DD} for PMOS sleep transistors (and lower than GND in the NMOS case), the gate voltage of the sleep transistor needs to be considered separately in the I_{OFF}^{ST} model. As doubling the transistor width will exactly double leakage currents and because of the fact that this linear dependency is independent of all other variables, W_{ST} can be separated. Thus, the model can be created for one reference width $W_{ST}^{ref} = 1\mu m$ only and can simply be scaled for any given W_{ST} later on. Furthermore, a voltage drop of zero can be assumed for the I_{ON}^{ST} model and a maximal voltage drop of $V_{DROP\text{-}OFF}^{ST} = V_{DD}$ across the sleep transistor is assumed for the I_{OFF}^{ST} model. The latter assumption leads to the worst remaining leakage and simplifies the overall model.

Beside the dynamic parameters, static parameters do also affect the models: the PGS, the semiconductor technology, selected process corner, and the HVT/SVT sleep device selection. For these parameters model splitting is applied for I_{ON}^{ST}, I_{OFF}^{ST}, and also for all remaining models. Thus, for each technology and each PGS separate models are created.

To model the remaining parameters ((V_{DD}, T, V_{Gate}) for I_{OFF}^{ST} and (V_{DD}, T) for I_{ON}^{ST} respectively), a 3 (2)-dimensional measuring field is created using Synopsys HSPICE® simulations. In these simulations all parameters are sampled in equidistant steps within their ranges. Thus, for any given combination of the dynamic parameters, the resulting currents are determined by Synopsys HSPICE® DC-analyses. These DC-analyses are very fast because only the steady operating point is computed and no transient analysis is required. Additionally, the PGS characterization is done independently of the power-gated circuit, the PGS is applied to. Figure 4.7 presents the test circuits and indicates the measured currents used for the model. As shown, I_{OFF}^{ST} is measured in between the PGS and power-gated circuit while the I_{ON}^{ST} current is purely measured at the PGS's gate connections. In this constellation, all occurring currents are measured.

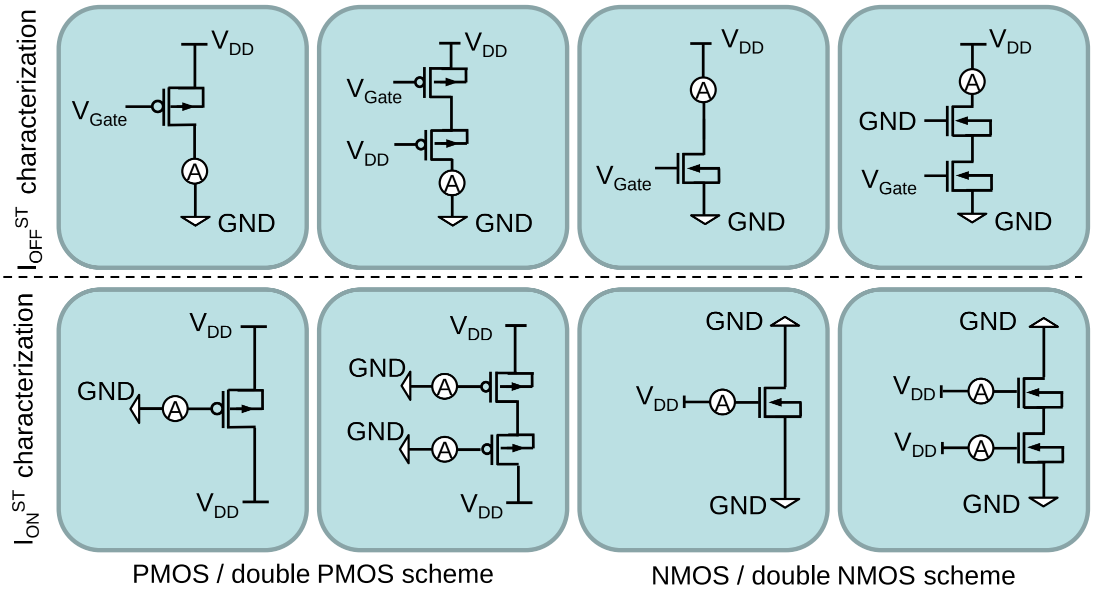

I_{OFF}^{ST} characterization
I_{ON}^{ST} characterization
V_{DD}
V_{Gate}
A
GND
PMOS / double PMOS scheme
NMOS / double NMOS scheme

Figure 4.7: Power-gating scheme leakage current measurements for I_{OFF}^{ST} and I_{ON}^{ST} model characterization

Once all data have been collected, the superlinear correlation between temperature/voltages and leakage currents is exploited for compression: the measuring field is logarithmized by logarithmizing every single value and a linear regression is applied to each dimension in order to reduce the amount of characterization data. The model then consists of few points and its usage consists of a linear 3 (2) dimensional interpolation, followed by a delogarithmization of the result and a linear scaling to W_{ST}.

4.2.4 Power-Gating Scheme Voltage Drop Models

$V_{DROP\text{-}ON}^{ST}$ Model Description

$V_{DROP\text{-}ON}^{ST}$ is responsible for a delay degradation of the RT component in on-state because it reduces the effective supply voltage. The proposed model characterization is again based on an isolated consideration of the PGS. As in the I_{ON}^{ST} model, important dynamic parameters are the transistor size, temperature, and supply voltage. Additionally, the dynamic power P_{DYN}^{RT} is of importance because it induces a high current flow through the gating circuitry.

W_{ST} has again a linear impact on $V_{DROP\text{-}ON}^{ST}$, because it linearly defines the PGS's resistance and Ohm's law postulates a linear dependency between the voltage drop and the resistance. Thus, W_{ST} can again be separated from the other parameters and the characterization is done for a reference width of $W_{ST}^{ref} = 1\mu m$.

For the remaining parameters test circuits are created as shown in Figure 4.8 and a three-dimensional measuring field stores the simulation results. As it can be seen in the test circuits, the voltage drop is applied to the PGS and the flowing current is measured instead of vice versa. Directly after measuring the I/V-relation is reversed. The reason for this is two-fold. At first, the parameter range of $V^{ST}_{DROP\text{-}ON}$ is known and bound to $[0, V_{DD}]$ whereas a maximum current flow would have to be pre-characterized first. Secondly, voltage sources can easier be set up in Synopsys HSPICE® than current sources.

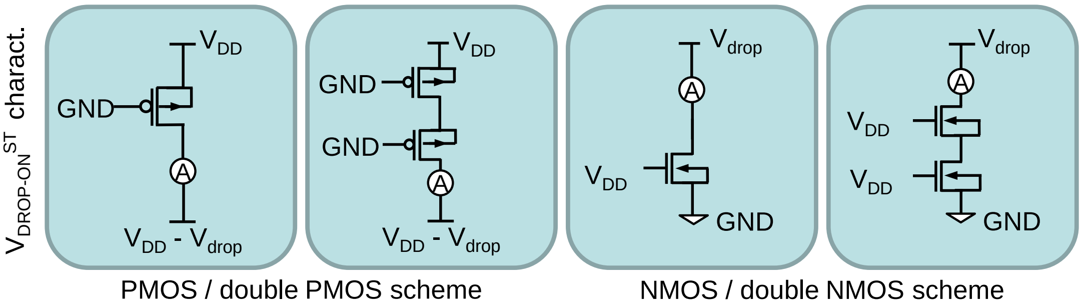


Figure 4.8: Power-gating scheme voltage drop measurements for $V^{ST}_{DROP\text{-}ON}$ model characterization

For this field, no fundamental physical dependency between the parameters could be found for compression, such as the exponential T/I^{ST}_{ON}-dependency in the I^{ST}_{ON} model. Thus, to further reduce the amount of characterization data, regression techniques were used. A non-linear regression technique is applied because the linear regression technique led to large model errors. In detail, the Levenberg-Marquardt fitting algorithm of [KYF04] is used to fit the influence of the current on the voltage drop. For fitting, a wide range of polynomial, exponential, logarithmical, and root functions have been tried. It showed up that the dependency can best be described with a polynomial of fourth order as shown in Equation 4.1. This observation clearly shows that the assumption to simplify sleep transistors as fixed resistances as done in the related work (see Section 3.1.2) is obsolete.

$$V^{ST}_{DROP\text{-}ON}(I) = \alpha_4 I^4 + \alpha_3 I^3 + \alpha_2 I^2 + \alpha_1 I^1 \tag{4.1}$$

Based on the fitting result, the three-dimensional field is simplified to a two-dimensional field storing a set of the four parameters $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ for each combination of temperature and supply voltage.

For any given temperature $\hat{t}$, supply voltage $\hat{v}$, and current $\hat{\imath}$ the model then first interpolates between the four adjacent polynomials (defined by the four α-sets of the neighboring points in the field) using bilinear interpolation shown in Equation 4.2. Afterwards the polynomial is used to compute the voltage drop under the given current $\hat{\imath}$. As a final step, the model result is sized in accordance to the sleep transistor width.

$$
\begin{aligned}
V^{ST}_{VDROP\text{-}ON}(\hat{t},\hat{v},\hat{i})) &= \alpha_4^{t,v}\cdot\hat{i}^4+\alpha_3^{t,v}\cdot\hat{i}^3+\alpha_2^{t,v}\cdot\hat{i}^2+\alpha_1^{t,v}\cdot\hat{i}^1\\
\alpha_n^{t,v} &= (1-\delta t)(1-\delta v)\cdot\alpha_{n(j,k)}+\delta t(1-\delta v)\cdot\alpha_{n(j+1,k)}+\\
&\quad (1-\delta t)\delta v\cdot\alpha_{n(j,k+1)}+\delta t\delta v\cdot\alpha_{n(j+1,k+1)}\\
\text{with}&\\
&\quad t_j\le\hat{t}\le t_{j+1},\;\; v_k\le\hat{v}\le v_{k+1}\\
&\quad \delta t=\hat{t}-t_j,\;\; \delta v=\hat{v}-v_k
\end{aligned}
\tag{4.2}
$$

$V^{ST}_{DROP\text{-}OFF}$ Model Description

$V^{ST}_{DROP\text{-}OFF}$ majorly impacts the switching energy E^{RT}_{SW} during wake-up. Test measurements have shown that a 20% misprediction of $V^{ST}_{DROP\text{-}OFF}$ results in a 50% misprediction of E^{RT}_{SW}. While the voltage drop is assumed to be equal to the supply voltage in the entire I^{ST}_{OFF} model, an analysis has shown that $V^{ST}_{DROP\text{-}OFF} = V_{DD}$ also holds for PGSs based on HVT devices and double gating schemes with an accuracy of above 96% for 4 of 5 transistor technologies under investigation. This is because HVT and stacked devices are such effective in gating (the latter are subjected to the body effect) even for very large W_{ST}. The only exception has been observed for the PTM $45nm$ transistor technology at fast process corner. In such devices, the threshold voltage is lowered and the overall speed is increased. At the same time I_{OFF} is increased and, in turn, $V^{ST}_{DROP\text{-}OFF}$ is lowered if these devices are used for PGS circuits. Thus, for a transistor technology at fast process corner, the approximation $V^{ST}_{DROP\text{-}OFF} = V_{DD}$ cannot be held and separate models need to be characterized for single- and double-based, as well as for SVT- and HVT-based power-gating schemes.

In contrast to the previous $V^{ST}_{DROP\text{-}ON}$ model, $V^{ST}_{DROP\text{-}OFF}$ further highly depends on the gated RTL component and thus cannot be characterized isolated from it. Beside the component and its bitwidth, additional parameters are the sleep transistor size W_{ST}, cutoff voltage V_{Gate}, temperature T, and the time passed after entering the sleep state t_{sleep}. The temperature is important because leakier sleep transistors reduce the voltage gap. The latter parameter t_{sleep} is ignored in this model and only the steady state after entering the sleep mode is modeled. As a consequence, E^{RT}_{SW} will be overestimated for short sleep periods.

For the remaining parameters again Synopsys HSPICE® DC-analyses are executed and the occurring voltage drop is measured as shown in the circuits of Figure 4.9. The model is then compressed by linear regression, stored as a multidimensional measuring field, and linear interpolation is used to get model estimates.

In future work the technique of [XVJ08] can be integrated in order to obtain a VV_{DD}-over-time model to not overestimate E^{RT}_{SW} during the model application. A short analysis of the VV_{DD}-lowering is presented in Figure 4.10. The measurements base on the adder component and the dynamic parameters used in the measurements of Figure 2.8 in Section 2.2.2. As indicated, the steady sleep state is reached at about $5\mu s$ (representing 500 cycles at a clock speed of $100MHz$) after the sleep transistor interrupts the power supply. In the meantime, the remaining virtual supply voltage is higher than in the steady state and if the circuit is powered on again, the state transition costs would be significantly lower.

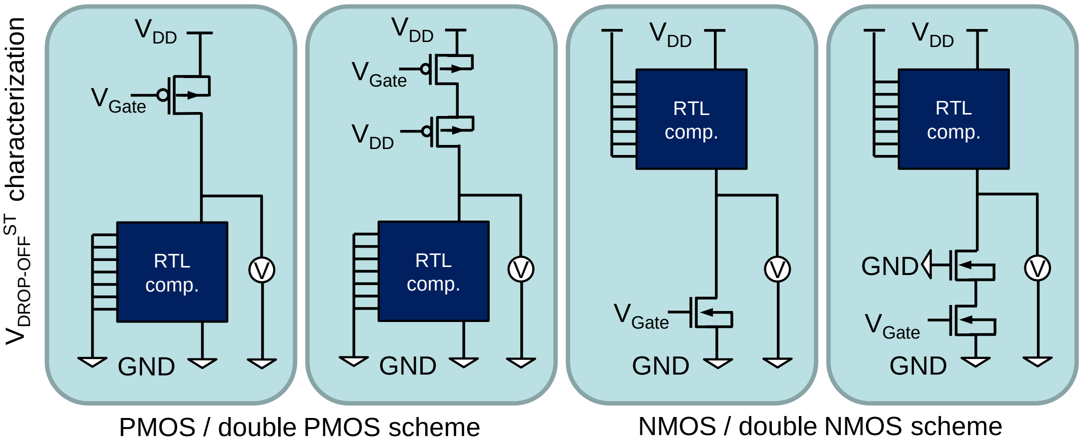

$V_{DROP\text{-}OFF}^{ST}$ characterization
V_DD
V_Gate
RTL comp.
V
GND
PMOS / double PMOS scheme
NMOS / double NMOS scheme

Figure 4.9: Power-gating scheme voltage drop measurements for $V_{DROP\text{-}OFF}^{ST}$ model characterization

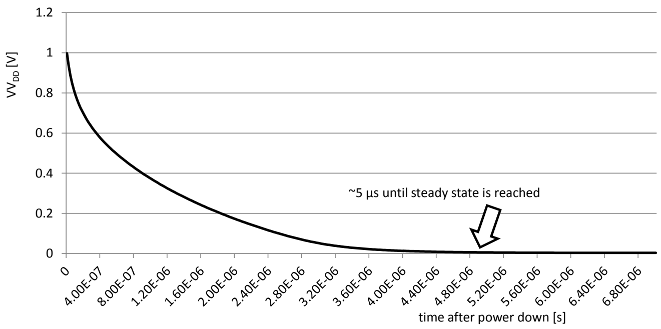


Figure 4.10: Lowering of VV_{DD} after power down

4.2.5 State Transition Energy Models

E_{SW}^{RT} Model Description

During state transition, dynamic power is drawn in the buffer chain, voltage anchors, sleep transistor, and RTL component. While the first three circuit parts are powered on all time and thus contribute classical dynamic power, the latter RTL component causes a significantly higher wake-up power. This leads to an energy overhead E_{SW}^{RT}, majorly impacting the break-even time after which a switched off component starts saving energy. As indicated in Section 3.1.4, dynamic energy estimation approaches basing on the charged capacitance cause unacceptable errors because incomplete and spurious transitions are not considered. These transitions highly correlate with the logical gate depth because more time passes for deeper gates until the blurred signal becomes static. Attempts in scaling capacity-based models in dependence on the logical gate depth of the circuit failed. For this reason, new E_{SW}^{RT} models are created for all RTL components in the library with the parameters PGS type, supply voltage, voltage drop, and sleep transistor size. The temperature is not a parameter in this model because it only affects the saturated virtual supply/ground voltage and this dependency is already encapsulated in the $V_{DROP\text{-}OFF}^{ST}$ model as described in Section 4.2.4 and shown in Figure 4.3.

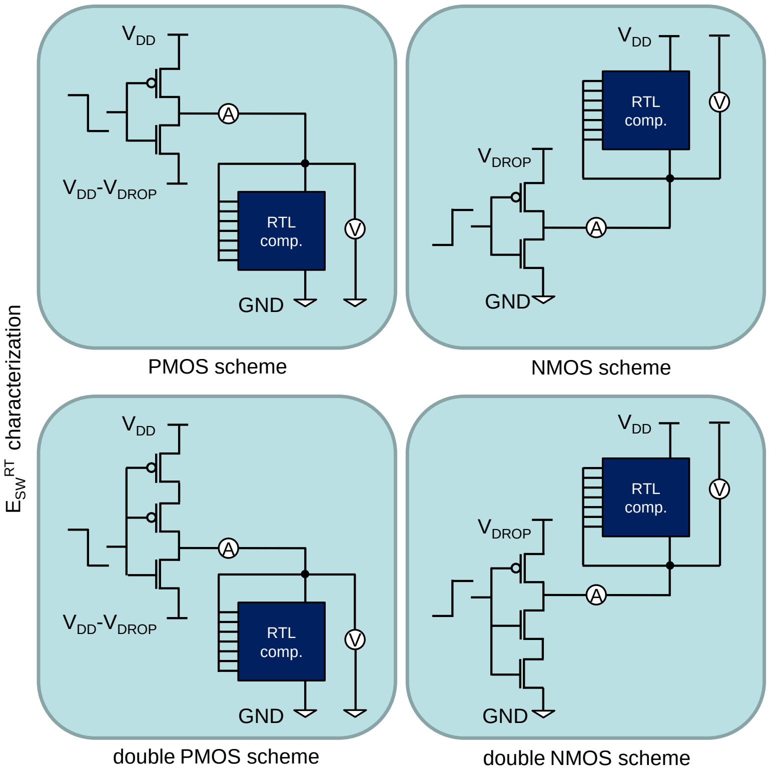


Figure 4.11: State transition energy measurements for E_{SW}^{RT} model characterization

Figure 4.11 shows the circuits that are used for transient Synopsys HSPICE® simulations. In contrast to DC-analyses in Synopsys HSPICE®, transient analyses simulate over time, are much slower, and thus necessary parameters need to be selected carefully to not explode the number of simulations. At the beginning of every simulation, an RTL component is in the sleep state at a remaining virtual supply voltage of $V_{DD} - V_{DROP\text{-}OFF}^{ST}$. Then, an edge occurs at an inverters input and the component is woken up. This edge is rectangular with a full voltage swing from V_{DD} to GND for PMOS schemes and GND to V_{DD} for NMOS schemes, respectively. It is flattened to a realistic slope by the inverter, charging the components internal capacitances. During the observed time the current and the virtual supply/ground voltage are sampled and its product is integrated over time to get an energy measurement ($E_{SW}^{RT} = \int^{t} I(t) \cdot VV_{DD}(t)dt$ for PMOS gating).

As shown in Figure 4.11 the RTL component input pins are connected to the virtual supply/ground line. Thus, their input capacitances are also charged during wake-up. Secondly, the maximal spurious transitions occur because the logic depth is largest. For both reasons E_{SW}^{RT} is overestimated. This simplification has been done after careful analysis to remove the input data dependency because an exhaustive simulation is far too time consuming. The induced error has been evaluated to be 13% maximally for an

$8bit$ component and it further diminishes with rising component size.

Figure 4.12 presents an exemplary analysis of the V_{DD}, W_{ST}, $V^{ST}_{DROP\text{-}OFF}$, and bitwidth parameter dependencies for a PMOS-gated $8bit$ adder. As indicated, $V^{ST}_{DROP\text{-}OFF}$ and W_{ST} are specified relatively to V_{DD} and W_{RT}. In the two upper charts $V^{ST}_{DROP\text{-}OFF}$ is fixed to $80\% \cdot V_{DD}$ and in the bottom charts W_{ST} is assigned to $2\% \cdot W_{RT}$. The downright chart further samples different bitwidths of the adder at a fixed voltage drop of $80\% \cdot V_{DD}$. Thereby, W_{RT} is defined as the sum of transistor channel widths within the pull-up and pull-down network of the gated component ($W_{RT} = \sum_{gates}(W_P + W_N)$). For all shown parameters the dependency can be approximated linearly and thus simple linear interpolation is used in the obtained E^{RT}_{SW} model. Especially the energy vs. bitwidth linearity for the adder component eases the model characterization as the characterization can be limited to a reference bitwidth and a vast number of costly transient Synopsys HSPICE® simulations are redundant.

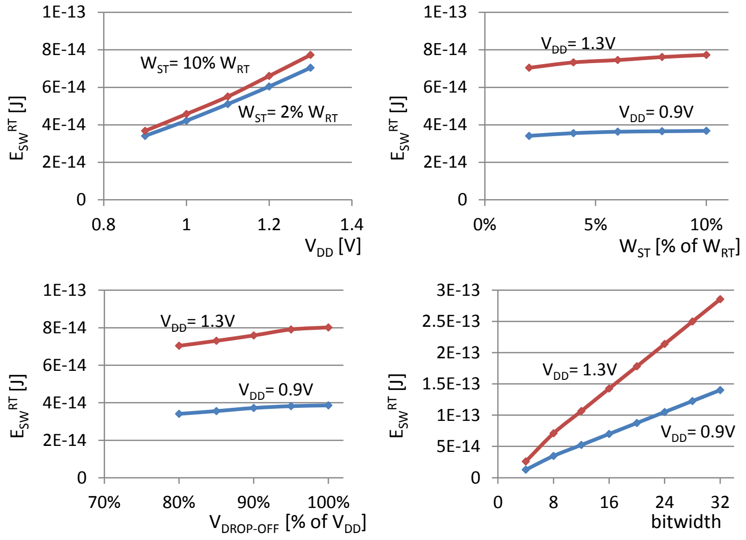


Figure 4.12: Parameter dependencies in the E^{RT}_{SW} model

Beside the adder component, the energy vs. bitwidth linearity has been evaluated to hold for every other component such as incrementer, decrementer, subtractor, and multiplier components. In all of these components, the ratio E^{RT}_{SW}/area is nearly constant.

Single vs. double gating has also an impact on the state transition energy. In comparison to single gating schemes, the I_{ON} current through stacked transistors is lower, reducing the spurious transitions within the component. Measurements showed up to 20% variance between single and double gating techniques. For this reason, both single and double PGS techniques are modeled separately. Further, the threshold-type of devices in the PGS impacts the state transition energy. SVT-devices are faster in waking up and lead to an increased amount and strength of spurious transitions. The energy difference between SVT- and HVT-based PGS has been evaluated to be up to 40% and thus the two types are also modeled separately.

Channel widths of the driving transistors in Figure 4.11 are of importance because they correspond to different W_{ST} and their I_{ON} limits the I_{MPC} and spurious transitions during wake-up. Their sizes need to be selected properly for any given RTL component and the model needs to cover a wide range for W_{ST}. For comparison, W_{RT} of a small $4bit$ adder in a $45nm$ technology is about $16\mu m$, a fast $32bit$ multiplier is about 483 times as big ($W_{RT} = 7729\mu m$), and a $32bit$ divider components is even about 1000 times bigger than the adder. In the characterization, W_{ST} is varied in the range of $[0\%; 10\%]$ of W_{RT} in order to simulate differently sized sleep transistors. $10\% \cdot W_{RT}$ has been chosen as upper bound because sleep transistors of this size will result in wake-up times below $1ns$ for the dominant majority of observed components, being equivalent to one cycle in a $1GHz$ clocked design.

E_{SW}^{ST} and E_{SW}^{BUF} Model Descriptions

Voltage anchors and sleep transistors will be represented as standard cells in future industrial semiconductor technologies. As a consequence, dynamic power estimates will directly be available for these cells in the .lib files and can be used for a holistic power-gating estimation. For example, standard cell voltage anchors are described in [BBMM06] and a family of sleep transistor cells is designed in [CPS$^+$07]. Because of a lack of availability of these technologies and due to the wide spectrum of PGSs supported in this thesis, an interim model for E_{SW}^{ST} has been built. It covers the dynamic energy of the sleep transistor and is used during evaluation. For this model, the parameters power-gating scheme, sleep transistor size, gate voltage, and supply voltage are used for the measurements. The characterization bases on the circuits shown in Figure 4.13. The RTL component is approximated by an equivalent resistance R_{RT} that is derived by measurements. Typical transitions for the $sleep$ signal are assured by placing an inverter in front of the PGS. Thereby, the transistors are sized in accordance to a tapering factor of e as it is typical for buffers. To this inverter a rising and falling edge is applied to cover both possible transitions. Further, the virtual supply voltage occurring at the drain connector is neglected by fixing it to V_{DD}/GND. Due to this simplification unrealistic gate-leakage currents may be measured but compared to the high dynamic power they only play a subordinated role. An ampere meter is placed at the PGS input and measures the current for charging the gate capacitances. Multiplied with the gate voltage and integrated over time, E_{SW}^{ST} is obtained. For double gating PGSs two energy estimates are summed up. Like in the other models, W_{ST} is separated and linear regression is performed to compress the model.

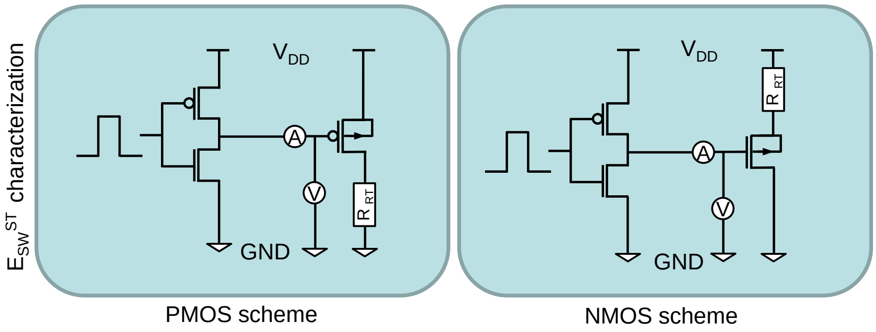


Figure 4.13: State transition energy measurements for E_{SW}^{ST} model characterization

Buffer elements are already standard cells in recent technologies and are also characterized as components in the component database (CDB) library. However, an interim model E_{SW}^{INV} has been built for a single inverter to easily estimate variable buffer chains with a given tapering factor. The model characterizes a single inverter with the same method used for E_{SW}^{ST}. It uses the same model parameters and measuring circuits shown in Figure 4.13, solely replacing the single transistor and its RT-equivalent resistance by an inverter and by further capturing the short-circuit currents through the inverter.

E_{SW}^{VA} is neglected in the recent implementation as only small transistors are used and the dynamic power is negligible small compared to E_{SW}^{BUF}. An explicit analysis of the voltage anchor induced dynamic power overhead is done in [BBMM06]. In summary, the power overhead is less than 0.02% in average. Merely, the area of functional units increases by 6% in average.

4.2.6 State Transition Delay Models

D^{BUF} Model Description

D^{BUF} represents a typical delay model. Similar to the corresponding power models, delay information for buffers will be included in future standard cell descriptions. For the use in this thesis, an interim model D^{INV} has been built for a single inverter to easily estimate variable buffer chains. The circuit used for model characterization is shown in Figure 4.14. A typical slope is outputted by a first and passed to the second inverter that is been characterized. For determining the delay, the output potential is measured over time and within this trace the point in time is found where the dropping/increasing potential crosses $50\% \cdot V_{DD}$.

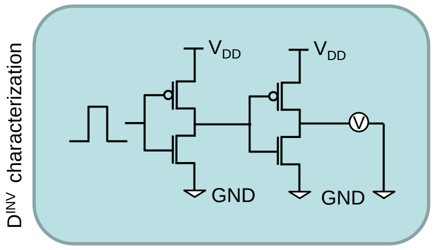


Figure 4.14: Delay measurements for D^{INV} model characterization

t_{wakeup}^{RT} Model Description

The PGS switch time (D^{ST}) and component wake-up time are considered together in one t_{wakeup}^{RT} model, that comes nearly for free during the E_{SW}^{RT} model generation, in that all necessary simulations are already executed. Thus, the wake-up times are also measured while awaking the components in the circuits of Figure 4.11. In contrast to the approximately linear impact of the supply voltage and sleep transistor size on E_{SW}^{RT}, the wake-up delay depends nonlinearly on them as shown in Figure 4.15. The voltage drop, defining the start point of simulation, remains its linear impact.

Stacked transistors of double gating schemes significantly slow down the wake-up delay because of their reduced I_{ON} current. In the left chart of Figure 4.16 t_{wakeup}^{RT} measurements are shown for adders

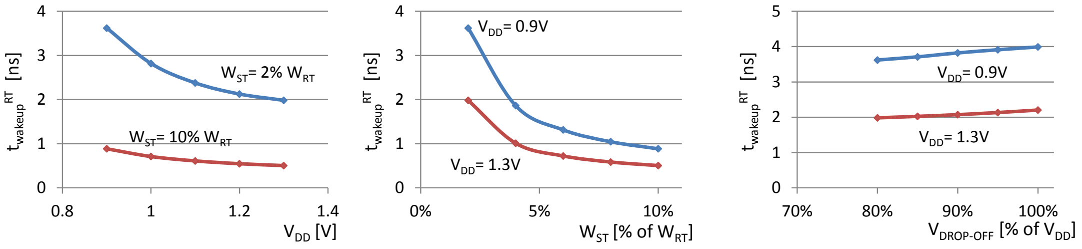


Figure 4.15: Parameter dependencies in the t_{wakeup}^{RT} model

of different bitwidths that are gated by single and double gating schemes. In all cases W_{ST} is relatively sized to $2\% \cdot W_{RT}$ and all other parameters are fixed. The measurements indicate a factor of $1.5 - 2.5$ in delay increase, when a second power-gating device is connected in series. Furthermore, t_{wakeup}^{RT} is significantly smaller for NMOS gating because of its higher I_{ON} current. As a consequence, all schemes need to be modeled separately. Furthermore, the chart indicates that t_{wakeup}^{RT} is nearly constant for a given PGS and constant W_{ST}/W_{RT} ratio. Beside the example adder component, this observation holds for all RTL components with a linear dependency between bitwidth/area and the state transition energy such as adders, subtractors, incrementers, and decrementers. This dramatically simplifies the model, since the characterization only needs to be done for a single reference bitwidth bw_{REF} and, comparable to E_{SW}^{RT}, the model can linearly be scaled to any W_{RT}. Regrettably, t_{wakeup}^{RT} varies too much between different components as shown in the right chart of Figure 4.16. Especially for small W_{ST} the delay variance is large and thus each RTL component is modeled separately.

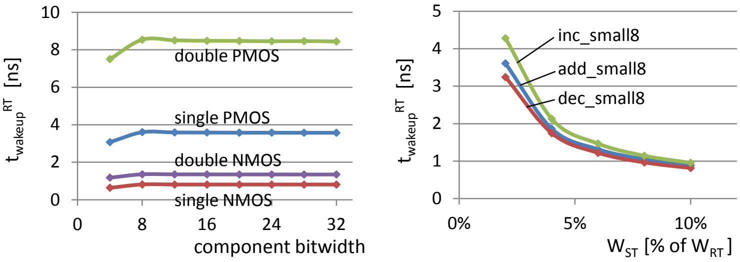


Figure 4.16: Bitwidths and component dependencies in the t_{wakeup}^{RT} model

Multiplier components have a quadratic dependency between bitwidth and area as well as between bitwidth and E_{SW}^{RT}. Although W_{ST} is sized relatively to the component size, the resulting delay increase is not compensated. In other words, the wake-up time is not as constant for different bitwidths as it is for the adder component in Figure 4.16. Thus, in the t_{wakeup}^{RT} model of multiplier components, the delay of a reference component is further scaled as shown in Equation 4.3. Thereby, a is a technology-specific slope that is determined during characterization and describes the wake-up delay increase.

$$t_{wakeup}^{RT} = t_{wakeup}^{REF} + \underbrace{t_{wakeup}^{REF} \cdot a \cdot (\frac{BW - BW_{REF}}{BW_{REF}})}_{\text{corrective additive term}} \tag{4.3}$$

For the Nangate technology a has been characterized to 0.41 denoting a 41% wake-up delay increase if the bitwidth of a multiplier is doubled. For the multiplier component in the reference bitwidth the corrective term in Equation 4.3 evaluates to zero. If $bw_{REF} = 8$ and an estimate should be obtained for a $16bit$ multiplier then $t_{wakeup}^{REF} \cdot 0.41$ is added to the wake-up time prediction of the reference $8bit$ multiplier. It is reasonable to conjecture that the matrix structure of multipliers with, in average, longer delay paths is responsible for the wake-up time increase.

The threshold type also influences the wake-up time as HVT devices have been evaluated to wake up a component up to two times slower than SVT devices because of their inherently slower switching. Thus, two separated models are necessary. Furthermore, and in contrast to the state change energy, the delay is impacted by the temperature as a higher temperature increases the delay. To not blow up the model complexity and number of costly transient Synopsys HSPICE® simulations, the worst-case temperature is assumed during characterization. Then, the model overestimates the delay in all cases or, in other words, the estimated delay holds for every ambient temperature.

4.2.7 Summary

Two leakage models, two voltage drop models, three state transition energy models, and two delay models have been proposed to provide estimates for the dominating effects of a power-gated RTL component and its interfacing circuitry. For all of these models, relevant parameters have been identified, their impact has been analyzed, and the modeling techniques have been proposed. The models of [Ros06] have been extended by HVT device and process corner support. Furthermore, the improved HSPICE® simulation capabilities now allow accurate state transition delay and energy models. In combination with already existing dynamic power (PowerOpt® CDB library), leakage ([Hel09]), and delay models ([HHN07]) all costs can now be predicted for a given transistor technology in a consistent manner.

4.3 Model Application and Estimation Flow

In Figure 4.1 the power-gating models are passed over to the estimation part. Assuming a given microarchitectural datapath at RT-level including power-gateable FUs and its power-management controller with a fixed schedule, they can then be used for estimating its power consumption in a cycle-by-cycle manner. This estimation depends on the dynamic and static parameters (T, V_{DD}, PGS, semiconductor tech., ...) as indicated. Beside these, further information regarding the implementation of the PGS circuit, more precisely the sleep transistor size W_{ST}, is necessary.

In the following, prerequisites of the flow are presented first. Section 4.3.2 then describes a cycle-accurate RTL estimation followed by a system-level utilization of the models in Section 4.3.3.

4.3.1 Prerequisites

Beside the proposed sub-models, the holistic flow to estimate RT-level units under power-gating requires additional models as mentioned. In the following, these prerequisites are explained, requirements are defined, and adequate selections from literature are justified. All models have in common that they have to be characterized for the targeted transistor technologies. Beside this requirement, the traceability and availability of model characterization also impacted its selection.

The first prerequisite is a RT-level dynamic power model P_{DYN}^{RT}. In detail, the dynamic power models of the PowerOpt® tool-internal component database (CDB) library files are used that are characterized by the PowerOpt® RIO library characterization engine. Beside dynamic power, the CDB library also consists of delay and leakage models but these models have not been pre-characterized for single transistors, have only been characterized for a fixed voltage and are not aware of virtual supply voltages, or their parameter ranges do not match with those of a power-gated component.

The RTL component delay D^{RT} strongly depends on the supply voltage and thus a voltage-dependent delay-model is required in the proposed estimation flow. In [SN90] the α-power law MOS model is introduced to describe carrier velocity saturation effects that are eminent in short-channel MOSFETs. Using this model, a closed-form expressions for the maximum frequency (and as reciprocal value for the delay) can be derived as shown in Equation 4.4 whereas α is the velocity saturation index of the technology, V_{TH} is the threshold voltage, K is the average switching activity and $f_{max}(V_{DD\_max})$ is the maximal frequency at which the resource can run under the maximal supply voltage V_{DD_max}.

$$f_{max}(V_{DD}) = K\frac{(V_{DD} - V_{TH})^{\alpha}}{V_{DD}} f_{max}(V_{DD\_max}) \tag{4.4}$$

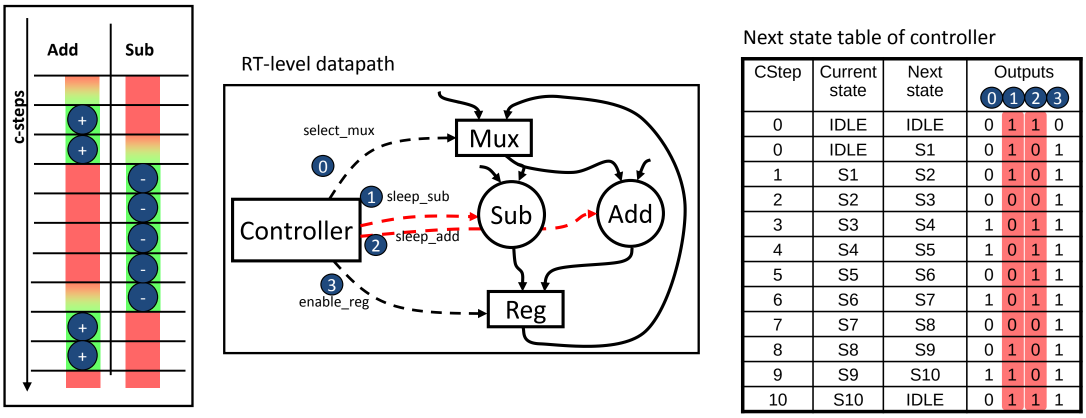
In practice, this model could have been applied in this work by fitting the parameters α and K in Equation 4.4 for the target technology. This could be done by measuring several (V, f)-points of an inverter-chain. Nevertheless, this approach was developed to model the delay of inverters and neglects data dependencies at RT-level. Additionally, the impact of the temperature is not covered in this model. For these reasons, the RT-level delay-model of [HHN07] is applied for D^{RT} in this thesis that fulfills all requirements. Furthermore, an automated model characterization is available.

Thirdly, leakage models are required to get estimates of the FUs during ungated state and of the ungated interface circuitry. The main requirement on these models is to depend on the same parameters as the power-gating models do and thus not to restrict their application. In this thesis the leakage model of [Hel09] is used since it meets all requirements. In detail, I^{RT}, I^{BUF}, and I^{VA} base on the approach of [Hel09]. Again, the characterization flow is available inhouse and provides accurate models as they have been evaluated in the corresponding literature.

4.3.2 Simulation-based Cycle-accurate RTL Estimation

Figure 4.17 illustrates a simple datapath containing a subtractor, an adder, a multiplexer, a register, and a corresponding controller. Beside the multiplexer select *select_mux* and the register enable *enable_reg* signals, the controller commands the sleep signals *sleep_sub* and *sleep_add*. On the left, the fixed schedule is shown for the two functional units in the design. Furthermore, power-management is color-coded in the schedule. In green tagged controller steps cs, the corresponding FU needs to be powered on, while it can sleep in red tagged controller steps. Being powered on means the component is either actively executing an operation or idling. Between on and off, the component needs to take the state transition from sleep to awake. The transition from awake to sleep is assumed to finish immediately, overestimating the state transition energy as described in Section 4.2.4. In this example a wake-up time of $t_{wakeup} < t_{cycle}$ is assumed and thus the state transition completes within one cycle. For each functional unit fu and each simulated cycle c, one of these four power modes is valid as defined in Equation 4.5. On the right of Figure 4.17 the next state table of the controller with its eleven controller states ($\#_{cstates} = 11$) is shown. For each current/next controller state tuple an allocation for the controller output is given. Beside the controller, a precondition of the estimation flow is to have fixed parameters of the PGS circuit, including the precise type of switch and its size.

$$pm(c, fu) \in \{active, idle, sleep, wakeup\} \tag{4.5}$$



Next state table of controller

CStep	Current state	Next state	Outputs			
			0	1	2	3
0	IDLE	IDLE	0	1	1	0
0	IDLE	S1	0	1	0	1
1	S1	S2	0	1	0	1
2	S2	S3	0	0	0	1
3	S3	S4	1	0	1	1
4	S4	S5	1	0	1	1
5	S5	S6	0	0	1	1
6	S6	S7	1	0	1	1
7	S7	S8	0	0	0	1
8	S8	S9	0	1	0	1
9	S9	S10	1	1	0	1
10	S10	IDLE	0	1	1	1

Figure 4.17: Simple datapath with power-gateable functional units and PM-aware controller

Estimating the overall energy consumption of such a system s is then examined in a cycle-by-cycle manner. With given testbench stimuli a functional simulation of this datapath will result in a pattern trace $pt = (p_1, p_2, ...)$ for each functional unit of the design. This trace is an ordered list of data pattern p whereas each pattern may either be a single data input for unary FUs as incrementers or a tuple for binary FUs as adders or multipliers. $cycle(p)$ denotes the simulation cycle the pattern is applied to the FU.

Assuming a small testbench with only one invocation of the example design, two traces exist as follows: $pt^{add} = (p_1^{add}, p_2^{add}, p_3^{add}, p_4^{add})$ and $pt^{sub} = (p_1^{sub}, p_3^{sub}, p_3^{sub}, p_4^{sub}, p_5^{sub})$. After the invocation the controller will enter the controller step cs_0, power-gate all FUs and wait for its next invocation. For this pure sequential design with a single invocation, the number of simulated cycles $\#_{cycles}^{sim}$ is equal to the number of controller steps $\#_{csteps}$. In general, $\#_{cycles}^{sim}$ is much higher for realistic testbenches.

The overall energy estimate of all functional units can now be derived by combining the set of sub-models as shown in Equation 4.6, $FU(s)$ being the set of power-gateable functional unit(s) in system s

and c a simulated cycle. As shown, the total energy estimate E_{total} is derived by summing up the cycle energy estimates E_c^{fu} for each functional unit fu and cycle c.

$$E_{total}^{FU(s)} = \sum_{fu \in FU(s)} \sum_{c=1}^{\#_{cycles}^{sim}} E_c^{fu} \tag{4.6}$$

The cycle energy estimate E_c^{fu} is specified in Equation 4.7. It differentiates the four power states and applies the required sub-models as shown.

$$E_c^{fu} = \begin{cases} (P_{DYN}^{RT}(p_k) + (I^{RT} + I_{ON}^{ST} + I^{BUF} + I^{VA}) \cdot V_{DD}) \cdot t_{cycle} \\ \qquad \text{with } cycle(p_k) = c \\ \qquad \text{for } pm(c, fu) = active \\ ((I^{RT} + I_{ON}^{ST} + I^{BUF} + I^{VA}) \cdot V_{DD}) \cdot t_{cycle} \\ \qquad \text{for } pm(c, fu) = idle \\ ((I_{OFF}^{ST} + I^{BUF} + I^{VA}) V_{DD}) \cdot t_{cycle} \\ \qquad \text{for } pm(c, fu) = sleep \\ (E_{SW}^{RT} + E_{SW}^{ST} + E_{SW}^{BUF} + E_{SW}^{VA}) \cdot (\lceil \frac{t_{wakeup}}{t_{cycle}} \rceil)^{-1} \\ \qquad \text{for } pm(c, fu) = wakeup \end{cases} \tag{4.7}$$

For simplicity, $t_{powerdown}$ is neglected and the power down is assumed to be free of costs although the buffers, PGS, and voltage anchors also toggle in the meantime and, for NMOS gated components, E_{SW}^{RT} is already drawn from the supply during power down. Instead, the complete E_{SW}^{RT}, E_{SW}^{ST}, E_{SW}^{BUF}, and E_{SW}^{VA} costs are considered during wake-up. Further, for simplicity, the state transition energies are averaged throughout all wake-up cycles. Despite all of these simplifications, the overall energy estimate remains the same.

In contrast to all other sub-models, the dynamic power model P_{DYN}^{RT} cannot be applied directly because the effective virtual supply voltage VV_{DD} is a model parameter that in turn depends on the voltage drop estimate $V_{DROP\text{-}ON}^{ST}$. A simple calculation of the working point is not possible because the models are not available in a closed analytical form. To handle this interdependency an iterative approach is used. Figure 4.18 shows the *voltage drop vs. current* characteristics of the PGS and the RTL component each for its own. Assuming a voltage drop of $V_{DROP\text{-}ON}^{ST} = 0V$ the averaged current through the PGS is zero and the current through the RTL component becomes largest. Assuming a voltage drop of $V_{DROP\text{-}ON}^{ST} = V_{DD}$ it is the other way round.

The iterative working point determination starts with a full supply voltage ($V_{DROP\text{-}ON}^{ST} = 0V$) leading to the maximum cycle-averaged current during operation $I_{CYCLE\_AVG}^{MAX}$. Using this current as input to the voltage drop model, a first rough approximation of the voltage drop can be obtained. By iterating this procedure the fixed working point can be determined easily. It is defined by the intersection of the two graphs because the current through the RTL component must be equal to the current through the sleep transistor. This voltage drop is then used to compute the virtual supply voltage and for the P_{DYN}^{RT} estimation. As shown in Figure 4.19, the iteration almost reaches the steady state after five iteration steps.

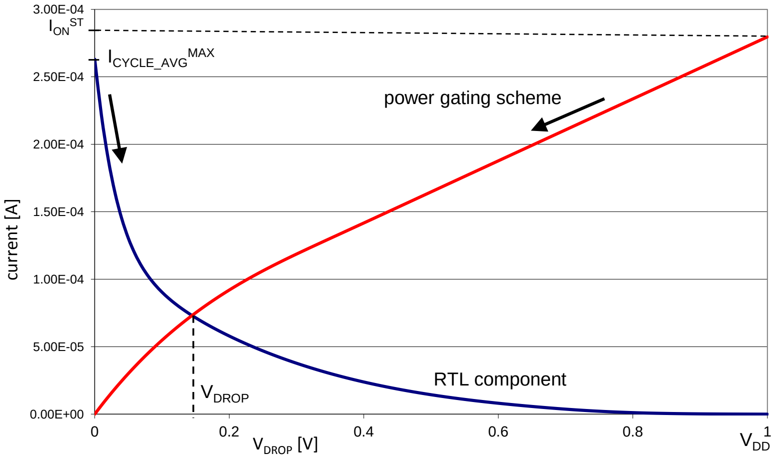


Figure 4.18: Voltage drop vs. current characteristics of power-gating scheme and RTL component.

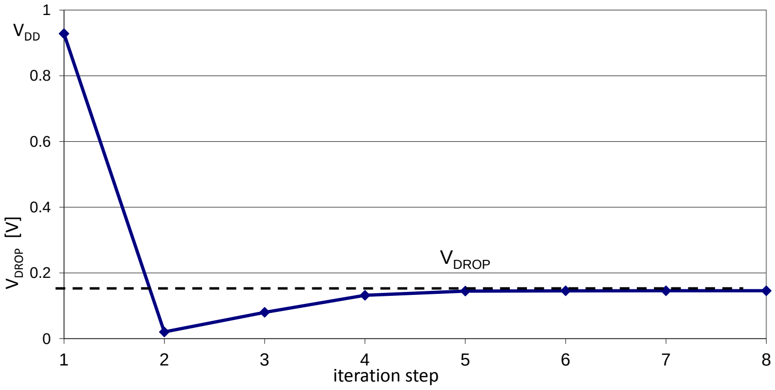


Figure 4.19: Cycle-averaged voltage drop in each iteration step.

Beside the models, the proposed functional unit estimation is also integrated into PowerOpt®. For further parts of the datapath such as registers, multiplexers, memories, and the controller, the PowerOpt® internal estimation is retained for a complete system energy estimate.

4.3.3 System-Level Modeling

In some cases an application of power-gating is not worthwhile for a small number of cycles. Section 2.2.2 points out the role of the break-even time and the influence of a semiconductor technology. Furthermore, highly utilized FUs or even pipelined designs prevent a cycle-based power down. Nevertheless, at system level, most designs are idling and awaiting a new invocation frequently and for many cycles. During these periods power-gating may be applied from outside to a system in order to reduce occurring leakage currents by gating all internal components simultaneously. This can be applied without any state loss if only the FUs are power-gated and the memory components remain ungated.

Denoting I_{idle}^{NoPG} as the idle/leakage current of a system without any power-gating, the increased idle current I_{idle}^{PG} of the same system with power-gateable functional units can be described by Equation 4.8. It regards the overhead costs of buffers, voltage anchors, and power-gating schemes.

$$I_{idle}^{PG} = I_{idle}^{NoPG} + \sum_{fu \in FU(s)} (I_{ON}^{ST} + I^{BUF} + I^{VA}) \tag{4.8}$$

$I_{sleep}^{PG}(s)$, as specified in Equation 4.9, then describes the remaining current of the same system while all functional units are power-gated.

$$I_{sleep}^{PG} = I_{idle}^{NoPG} + \sum_{fu \in FU(s)} (I^{BUF} + I^{VA}) - \sum_{fu \in FU(s)} (I^{RT} - I_{OFF}^{ST}) \tag{4.9}$$

In total, the difference $I_{idle}^{NoPG} - I_{sleep}^{PG}$ of these currents is saved by power-gating the functional units. The state transition energy is negligible for $t_{sleep} >> t_{wakeup}$.

4.3.4 Summary

The proposed models of Section 4.2 are supplemented by appropriate leakage, delay, and dynamic power models taken from literature. It is shown, how they are used altogether to cycle-accurately estimate the energy consumption of power-gateable FUs within a given datapath. Secondly, it is shown that the models are also applicable to estimate highly utilized and pipelined systems that are only worthwhile to be power-gated at system-level.

4.4 Summary

In summary, a characterization technique has been proposed that can be used to predict functional units' power consumption under power-gating. It bases on model splitting and proposes sub-models for each relevant effect for each steady and transient state. Beside the state, the models are aware of the surrounding temperature and of hardware overhead. As power-gating circuitry several schemes are supported, covering PMOS/NMOS sleep transistors in single and stacked realization, in high- and standard-threshold version as well as advanced super cutoff techniques. All models are automatically created by C++ programs triggering Synopsys HSPICE® simulations, gathering and interpreting the measurements in order to create small model files that are outputted and integrated in the PowerOpt®-internal CDB library.

The model use is two-folded. On the one hand, they can directly be used to predict the power consumption cycle-accurately of any given RTL datapath with power-gateable functional units, its corresponding controller, and for fixed parameters of the power-gating scheme. Due to its cycle-accuracy, the estimation can be applied component-wise as well as datapath-wise. On the other hand, the models serve as foundation for synthesis optimizations proposed in Chapter 5.

5 Dynamic Leakage-Management during High-Level Synthesis

The models and estimation techniques of Chapter 4 are used to make substantiated decisions during the high-level synthesis. In this thesis, the consideration of power-gating bases on and is integrated into the HLS-engine of PowerOpt®. Figure 5.1 schematically summarizes the sequential flow within the tool from the behavioral design description in C or SystemC to a synthesizable RTL Verilog output and a power estimate. At first, the design is internally represented as a CDFG as described in Section 2.3. Then, in a fixed sequence, scheduling, allocation and binding, power-management controller synthesis, and datapath generation are executed and the final design is outputted. Out of this flow, the scheduling and allocation/binding phases have been chosen for an optimization as they offer opportunities for a worthwhile integration of power-gating.

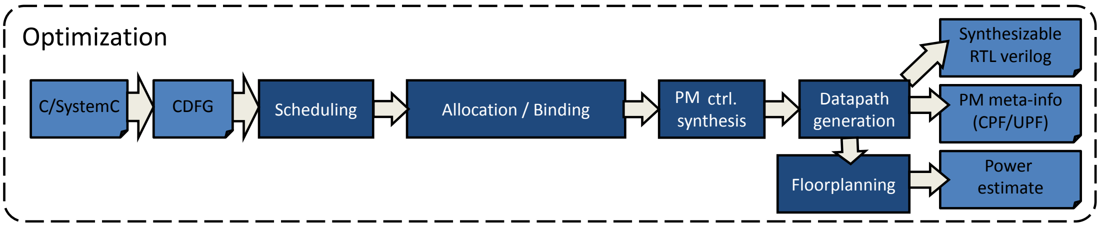


Figure 5.1: Overview on the optimization flow

Before focusing on these synthesis phases, Section 5.1 presents the applied delay-dependent sleep transistor sizing approach in order to automatically constrain the size and to relieve the user from this issue. The remainder of this chapter follows the tool flow. Thereby, the existing separation of scheduling and binding/allocation in PowerOpt® remains in this thesis because solving an explicit power minimization under the consideration of power-gating, combining both HLS tasks into one problem formulation, has been evaluated to fail due to the exploding design-space. The evaluation in [Sch08] impressively shows that only small designs of a few cycles can exhaustively be analyzed. Section 5.2 presents an ILP-based scheduling that has initially been developed in [Sch08]. It creates continuous idle phases between operations without a prolongation of the schedule. The continuous idle phases are then transferred to RTL components by binding the operations in a power-optimized manner under the consideration of power-gating. This functional unit binding and allocation approach is presented in Section 5.3. Both approaches have been published in [RSN09]. Section 5.4 then describes the power-management controller synthesis followed by a summary in Section 5.5.

5.1 Delay-dependent Sleep Transistor Sizing

Sleep transistor sizing is mandatory for estimating power-gated RTL components. It is further a design parameter and enlarges the design-space of any HLS exploration. As described in Section 3.1.1, two classes of approaches can be separated. In this thesis an *average-current* driven sizing approach is used as is leads to significant smaller sleep transistors. Thereby, *average-current* refers to the cycle-averaged current I_{CYCLE_AVG} induced by the maximum possible switching activity within one operation cycle for a given RTL component. Furthermore, it is delay-constrained as the user specifies a maximum delay increase caused by the PGS.

For predicting I_{CYCLE_AVG} the model P^{RT}_{DYN} is used. Since in this model the data dependency is abstracted to the 1-normalized hamming distance $Hd_1(p_i, p_{i+1})$ of two consecutive input patterns (p_i, p_{i+1}), the maximal dynamic power within one cycle is obtained by using the worst case Hd_1 value. I_{CYCLE_AVG} further depends on the leakage current I_{ACTIVE} as shown in Equation 5.1 but I_{ACTIVE} is almost independent on input vectors at RT level as figured out in [Hel09]. In total, this leads to an upper bound average current, an upper bound voltage drop, and to an upper bound delay degradation.

$$I_{CYCLE\_AVG} = \frac{P^{RT}_{DYN}(Hd_1)}{V_{DD}} + I_{ACTIVE} \tag{5.1}$$

Based on this current, the sleep transistor is sized to maintain a maximum user-defined delay increase. For this purpose, W_{ST} is initialized to the smallest possible value defined by the node size of the semiconductor technology and $V^{ST}_{DROP\text{-}ON}$ is used to compute the occurring voltage drop. The remaining virtual supply voltage $(VV_{DD} = V_{DD} - V^{ST}_{DROP\text{-}ON})$ is then used to compute the new and higher delay D^{RT} with the model of [HHN07]. Based on nested intervals, W_{ST} is iteratively enlarged until the delay constraint is met.

Figure 5.2 exemplarily shows the sizing approach for a $32bit$ adder component. Without power-gating the delay is $D^{RT} = 3.08ns$. In the presence of a sleep transistor the delay increases as a function of W_{ST}. With a defined delay increase of 20%, the delay is allowed to increase to $D^{RT} = 3.696ns$, leading to a sleep transistor size of $W_{ST} = 2\mu m$. As mentioned, the delay is user-constrained in the recent implementation but during synthesis the sleep transistor sizing may further be restricted by the clock speed. Assuming a clock speed of $300Mhz$, the delay is not allowed to exceed $3.333ns$. In turn, the sleep transistor size for this adder needs to be $3.2\mu m$ at least.

5.2 Low Leakage Functional Unit Scheduling

The task of high level scheduling is to assign operation nodes within a CDFG to fixed control steps. In doing so, several constraints have to be satisfied and remaining degrees of freedom can be used for a design-space exploration. In terms of a worthwhile application of power-gating, a scheduler strategy should address two aspects that are closely related. On the one hand, it should create maximal continuous idle periods between operation nodes of the same type that exceed the break-even time t_{be} of the power-gating technique. On the other hand, if t_{be} cannot be exceeded, the scheduler should cluster operations in order to enlarge periods of idleness before and after the operation cluster.

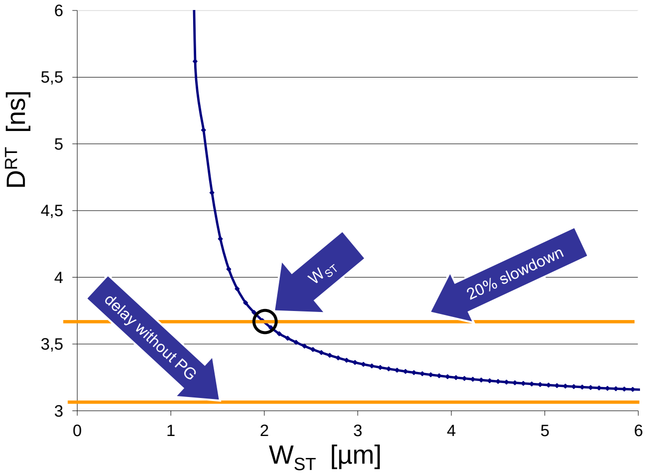


Figure 5.2: Sleep transistor sizing for a $32bit$ adder and 20% delay increase

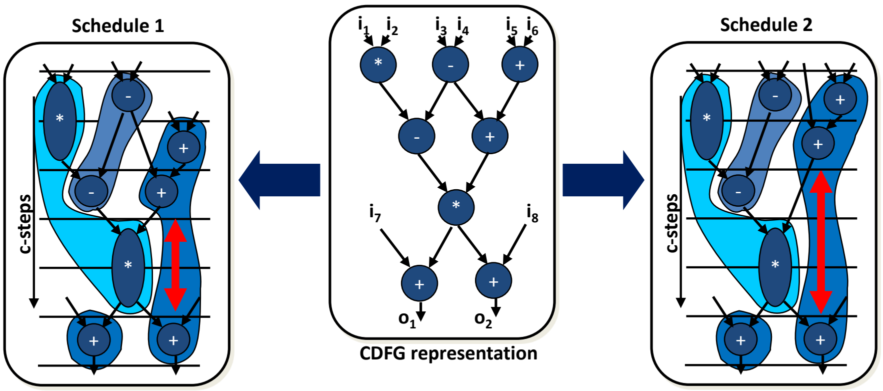


Figure 5.3: Alternative schedules for a given CDFG

Figure 5.3 shows two schedules for the HLS example design of Section 2.3. In this example, two additions are preponed in the second schedule by one cycle and the emerging idle period is lengthened. Obviously, power-gating can benefit from this schedule only if the binding and allocation are as shown. For this reason, an isolated investigation of the scheduling is not capable to improve the adoption of power-gating. Nevertheless, it can serve as a heuristic optimization for a subsequent DPM-aware binding and allocation phase of the synthesis that is presented in Section 5.3.

In this thesis, an integer linear program formulation is proposed for the scheduler that aims the indicated operation clustering. The use of integer linear programming to solve a scheduling problem has been introduced in [HP81] the first time and since then ILPs were commonly used for high-level synthesis tasks. The problem formulation in this thesis consists of three kinds of equations. These constrain the control step assignment, preserve data dependencies, and build the cost function. Sections 5.2.1 to 5.2.3 give a detailed presentation.

5.2.1 Constraining the Control Step Assignment

Each operation op has to be assigned to a unique control step cs within the operation-dependent timeframe $[ASAP(op), ALAP(op)]$ that depends on the control- and data-flow and can be determined statically. For example, the $ASAP$ and $ALAP$ times are constrained by data dependencies and loop initializations. To formulate a control step assignment constraint in the ILP, a Boolean variable $b_{cs,op}$ is introduced for each operation op and cstep $cs \in [ASAP(op), ALAP(op)]$. The variables are constrained as shown in Equation 5.2.

$$\forall op : \sum_{cs=ASAP(op)}^{ALAP(op)} b_{cs,op} = 1 \tag{5.2}$$

By this constraint, only one Boolean variable $b_{cs,op}$ can evaluate to 1 for each operation. The assigned cstep $cstep(op)$ can be obtained as an integer value using Equation 5.3.

$$\forall op : cstep(op) = \sum_{cs=ASAP(op)}^{ALAP(op)} cs \cdot b_{cs,op} \tag{5.3}$$

The $ASAP$ and $ALAP$ bounds of each operation are not necessarily required and $[0, \#_{csteps}]$ can be used instead whereas $\#_{csteps}$ denotes the number of cycles within the critical path. Nevertheless, they constrain the control step assignment, reduce the amount of variables within the ILP, and thus speed up its solving.

5.2.2 Preservation of Data Dependencies

Next, data-flow dependencies between each pair of operations op_s and op_p need to be constrained. Thereby, flow-, anti-, and out-dependencies are differentiated as shown in Equations 5.4 to 5.6. In the case of a flow-dependency, an operation op_p has to compute (and store) the output before a subsequent operation op_s can read it. Thus, the predecessor operation op_p has to be completed first. $\lceil \frac{D^{RT}(type(op_p), bw_{max})}{t_{cycle}} \rceil$ denotes the amount of cycles that are necessary to execute the operation op_p on a maximally sized functional unit.

$$cstep(op_s) \geq cstep(op_p) + \left\lceil \frac{D^{RT}(type(op_p), bw_{max})}{t_{cycle}} \right\rceil \tag{5.4}$$

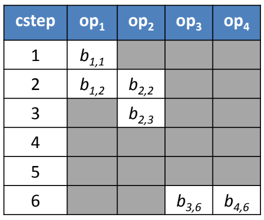
An anti-dependency constrains an operation op_p to read a value before a subsequent operation op_s is allowed to overwrite the value.

$$cstep(op_s) + \left\lceil \frac{D^{RT}(type(op_s), bw_{max})}{t_{cycle}} \right\rceil > cstep(op_p) \tag{5.5}$$

At last, an out-dependency constrains two write accesses to the required order.

$$cstep(op_s) + \left\lceil \frac{D^{RT}(type(op_s), bw_{max})}{t_{cycle}} \right\rceil > cstep(op_p) + \left\lceil \frac{D^{RT}(type(op_p), bw_{max})}{t_{cycle}} \right\rceil \tag{5.6}$$

Example: Considering the four adder operations within the unscheduled CDFG of the example design, the corresponding $ASAP$ and $ALAP$ times are constrained by relevant data dependencies as shown in Figure 5.4. For this example only the variables $b_{1,1}, b_{1,2}, b_{2,2}, b_{2,3}, b_{3,6}$ and $b_{4,6}$ are introduced for possible control step assignments. op_3 and op_4 need to be scheduled in cstep 6 whereas op_1 and op_2 have $ASAP/ALAP$ time windows of $[1, 2]$ and $[2, 3]$. The flow-dependency from op_1 to op_2 will further constrain $b_{1,i}$ and $b_{2,j}$ by $j > i$.



cstep	op$_1$	op$_2$	op$_3$	op$_4$
1	$b_{1,1}$			
2	$b_{1,2}$	$b_{2,2}$		
3		$b_{2,3}$		
4				
5				
6			$b_{3,6}$	$b_{4,6}$

Figure 5.4: Constraints on control step assignment

5.2.3 Cost Functions

Based on the constraints on control step assignment and preservation of data dependencies, two cost functions for different intentions of optimization are proposed. At first, a cluster-building formulation is presented. Its inherent cost function serves as a cluster-building heuristic by minimizing the amount of csteps with a higher demand on resources in comparison to the preceding csteps.

Afterwards, a second cost function is defined in order to achieve the minimal amount of necessary resources as it is also the optimization criterion in the force-directed scheduling approach. But in contrast to the FDS, it further minimizes the sizes of necessary resources by handling different input bitwidths and it is a non-heuristic problem-solving approach. It will be used during evaluation in order to show that the cluster building heuristic can outperform any minimal concurrency approach.

Operation Clustering Cost Function

Clusters of operations as well as idle periods in between are only meaningful for power-down sequences if the operations are of the same type $t \in (add, sub, mult, dec, inc)$. Thus, each operation type is considered separately in the following but due to control- and data-flow dependencies their schedule remains strongly interdependent.

For every control step cs and operation type t a Boolean variable $extraOps_{t,cs}$ is introduced. These variables are constrained as shown in Equation 5.7. The variable for cstep zero of the global idle state is initialized as $opsCount_{t,0} = 0$.

$$\forall t, \forall cs : extraOps_{t,cs} \geq \frac{opsCount_{t,cs} - opsCount_{t,cs-1}}{\#_{operations}} \tag{5.7}$$

Within this equation the variables $opsCount_{t,cs}$ indicate the number of operations as defined in Equation 5.8. The complete fraction thus indicates whether in cstep cs more operations are scheduled than in the previous cstep $cs - 1$ or not. If this is the case the Boolean variable $extraOps_{t,cs}$ evaluates to 1. Only if $opsCount_{t,cs} \leq opsCount_{t,cs-1}$ it can evaluate to 0.

$$\forall t, \forall cs : opsCount_{t,cs} = \sum_{i=1}^{\#_{operations}} b_{cs,op_i}, \quad type(op_i) = t \tag{5.8}$$

The cost function that is minimized by the solver is then formulated as follows:

$$costs = \sum_{t} \sum_{cs=1}^{\#_{csteps}} extraOps_{t,cs} \cdot (I_{ACTIVE}(t, bw_{max}) - I_{SLEEP}(t, bw_{max})) \tag{5.9}$$

The interior sum first counts the number of control steps in which more operations are scheduled than in the previous control step. Each of these csteps exert an influence on the cost function that is weighted with the difference in leakage current between $active$ and $sleep$ state. Thereby, $I_{ACTIVE}(t, bw_{max})$ and $I_{SLEEP}(t, bw_{max})$ depend on the component type t and the maximum bitwidth bw_{max} is assumed for its estimation. The weighting is important because potential savings in leakage power significantly differ between component types. For example, a clustering of multiplier operations may be much more effective than a clustering of incrementer operations. The costs are determined for all component types and the total sum is built, defining the cost function that is minimized by the ILP solver.

Recent leakage-aware synthesis approaches assume that the best results are obtained by minimal concurrency heuristics and thus they reduce the necessary amount of RTL components. This assumption counteracts an effective use of different power modes as argued in Section 3.2. Thus, by default, the proposed ILP scheduler is not limiting the number of components. Nevertheless, the available chip area may be limited and demand for an upper allocation constraint on the number of resources r_t as described in Equation 5.10.

$$\forall t, \forall cs : opsCount_{t,cs} \leq r_t \tag{5.10}$$

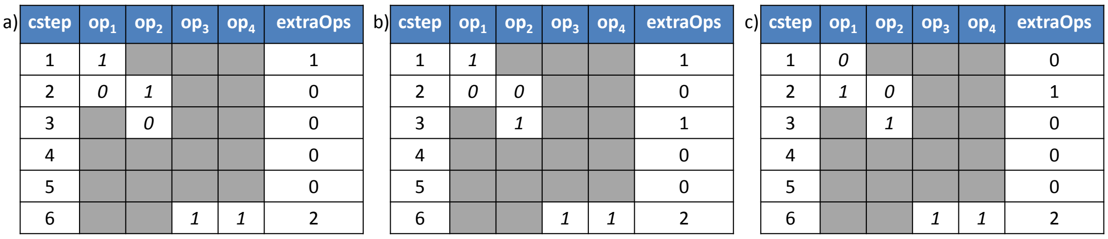
The complexity of the proposed ILP formulation in terms of the worst-case number of variables is described in Equation 5.11. As the number of different component types $\#_{types}$ in a design is highly limited, the number of variables is dominated by $\#_{csteps} \cdot \#_{operations}$.

$$\#_{variables}(ILP) = \underbrace{\#_{csteps} \cdot \#_{operations}}_{\text{Boolean } b_{cs,op}} + \underbrace{\#_{csteps}}_{\text{integer } cstep(op)} + \underbrace{\#_{csteps} \cdot \#_{types}}_{\text{integer } opsCount_{t,cs}} + \underbrace{\#_{csteps} \cdot \#_{types}}_{\text{integer } extraOps_{t,cs}} \tag{5.11}$$

Example: Consider again the example of Figure 5.4. In total, three legal schedules exist as shown in Figure 5.5. Thereby, Equation 5.7 constrains the $extraOps$ variables as shown in the right columns. It can be seen that schedule $a)$ and $c)$ perform a clustering because there is no idle cycle between op_1 and op_2. In these cases the cost function of Equation 5.9 also evaluates to the minimum costs of $costs = 3 \cdot (I_{ACTIVE}(t, bw_{max}) - I_{SLEEP}(t, bw_{max}))$. In $b)$ one additional $extraOps$ variable evaluates to 1 and the cost function evaluates to $costs = 4 \cdot (I_{ACTIVE}(t, bw_{max}) - I_{SLEEP}(t, bw_{max}))$.

Resource and Size Minimizing Cost Function

In addition to the heuristic force directed scheduling, a non-heuristic approach is proposed leading to the smallest and least leaky design. As different word lengths may occur in a behavioral-level design



a)

cstep	op_1	op_2	op_3	op_4	extraOps
1	*1*	▨	▨	▨	1
2	*0*	*1*	▨	▨	0
3	▨	*0*	▨	▨	0
4	▨	▨	▨	▨	0
5	▨	▨	▨	▨	0
6	▨	▨	*1*	*1*	2

b)

cstep	op_1	op_2	op_3	op_4	extraOps
1	*1*	▨	▨	▨	1
2	*0*	*0*	▨	▨	0
3	▨	*1*	▨	▨	1
4	▨	▨	▨	▨	0
5	▨	▨	▨	▨	0
6	▨	▨	*1*	*1*	2

c)

cstep	op_1	op_2	op_3	op_4	extraOps
1	*0*	▨	▨	▨	0
2	*1*	*0*	▨	▨	1
3	▨	*1*	▨	▨	0
4	▨	▨	▨	▨	0
5	▨	▨	▨	▨	0
6	▨	▨	*1*	*1*	2

Figure 5.5: Possible schedules and extraOps variable computation

and leakage currents depend on the component size, the cost function of this ILP incorporates the operation's bitwidths. For this reason, integer variables $r_{t,bw}$ are introduced that indicate the maximum needed number of components of type t and bitwidth bw across all control steps. A compact description for the determination of $r_{t,bw}$ is shown in Equation 5.12.

$$\forall t, \forall bw : r_{t,bw} = max\left(\bigcup_{cs=1}^{\#csteps}\left\{\sum_{i=1}^{\#operations} b_{cs,op_i}\right\}\right), \quad type(op_i) = t,\ bitwidth(op_i) = bw \tag{5.12}$$

In this equation, the inner sum counts the number of operations for an actual bitwidth and control cstep. This is done for each cstep and the maximum function determines the integer value of $r_{t,bw}$.

Since ILP solver do not support maximum functions directly, the implementation of Equation 5.12 is done by the constraints in Equation 5.13. As it can be seen, the lower bound of $r_{t,bw}$ is constrained separately for each control step. The upper bound needs to be left unconstrained in this equation to not define conflicting constraints. Nevertheless, the variables $r_{t,bw}$ reflect linearly in the cost function that is minimized. Thereby, the upper bound is constrained and Equation 5.13 in addition to the cost function implements Equation 5.12.

$$\forall t, \forall bw, \forall cs : r_{t,bw} \geq \sum_{i=1}^{\#operations} b_{cs,op_i}, \quad type(op_i) = t,\ bitwidth(op_i) = bw \tag{5.13}$$

Computing of resource numbers is also done independently of the bitwidth in order to determine the overall maximum component number per type r_t. Equation 5.14 constrains these variables for every control step and component type.

$$\forall t, \forall cs : r_t \geq \sum_{i=1}^{\#operations} b_{cs,op_i}, \quad type(op_i) = t \tag{5.14}$$

After the number of components has been determined, the cost function can be formulated as shown in Equation 5.15. The variables $I_{ACTIVE}(t, bw)$ quantify leakage currents of a component of type t and bitwidth bw in the active state. bw_{max} is the maximum bitwidth within the design. Thus, this cost function sums up $active$-state leakage currents of different component types and bitwidths.

The first term $r_{t,bw_{max}} \cdot I_{ACTIVE}^{t,bw_{max}}$ defines the leakage current of the components with the largest bitwidths in the design. The second term then describes how many components of smaller bitwidths are necessary in addition to the largest in order to execute all parallel scheduled operations.

$$costs = \sum_{t} \left(r_{t,bw_{max}} \cdot I_{ACTIVE}(t, bw_{max}) + \sum_{bw_1=1}^{bw_{max}-1} \left(r_t - \sum_{bw_2=bw_1+1}^{bw_{max}} r_{t,bw_2} \right) \cdot I_{ACTIVE}^{t,bw_1} \right) \tag{5.15}$$

Example: Assume a schedule of two control steps only comprising adder operations. Three $16bit$ and one $8bit$ operations are scheduled in cstep 1 and two $16bit$ and three $8bit$ operations are scheduled in cstep 2. Equation 5.13 then constrains $r_{add,8}$ and $r_{add,16}$ to $r_{add,16} \geq 3$ and $r_{add,8} \geq 3$. r_{add} is constrained to $r_{add} \geq 5$.

$$costs = \sum_{t} \left(r_{t,16} \cdot I_{ACTIVE}(t, 16) + (r_t - r_{t,16}) \cdot I_{ACTIVE}(t, 8) \right) \tag{5.16}$$

The cost function for the example design is then described in Equation 5.16 and can be interpreted as follows. At least $r_{add,16} = 3$ $16bit$ adders have to be used because of the three operations in the first cstep. Operation binding is not included in the ILP formulation but an implicit maximum binding is assumed. This means that the remaining $16bit$ adder in the second cstep is used for an $8bit$ operation as well and only two more $8bit$ adders ($r_{add} - r_{add,16} = 5 - 3 = 2$) have to be instantiated.

5.2.4 Summary

The proposed ILP scheduling performs a simple operation clustering. Its cost function reduces the number of idle phases and thereby it clusters operations of the same type to enlarge idle times between these clusters. Additional equations guarantee the successor-predecessor relationship for every pair of operations due to data-dependencies. The scheduling serves as a heuristic pre-optimization to the subsequent binding and allocation phase. In addition, a second non-heuristic cost function is defined for evaluation that minimizes concurrency in the schedule and further reduces the bitwidth size of components.

5.3 Low Leakage Functional Unit Allocation and Binding

The exploitation of existing idle periods within a schedule needs to be done by a functional unit binding under an appropriate allocation. In the following, a power-management aware low power binding and allocation approach is presented. It mainly bases on a modified cost function of the approach in [Kru01] and preceding works [KSJ+99, KSJN99, KSJ+00a, KSvCJ+01, KSJ+00b]. The authors present a sophisticated binding and allocation approach that optimizes for the dynamic power only. In contrast, its cost function is modified in this thesis to cover the dynamic power due to activity pattern, state-dependent static power, as well as state transition costs. Thus, it optimizes the overall power-consumption being aware of the power-gating technique, its induced costs, and its break-even time.

The approach consists of four steps. At first, an energy cost matrix is computed for each component type t in Section 5.3.1. This cost metric regards all the aforementioned power estimates and applies the models of Chapter 4. In a second step, a non-heuristic lower bound low-power binding is determined in Section 5.3.2 under a given resource constraint. This binding may potentially still contain conflicting constraints and thus it may be invalid. For this reason, relaxation and improvement heuristics, being summarized in Section 5.3.3, are used to find a valid near optimal binding with no conflicting constraints.

In the last step, the overall resource constrained binding approach is iterated in order to trade off different resource allocations. Steps two to four are only summarized in this thesis as they are described in detail in the given literature.

5.3.1 Energy Cost Representations

Each RTL component instance consumes a particular amount of energy that depends on the operations mapped to the resource. Thereby, the components overall energy can be split to individual energy values $E(op_i, op_j)$ for every tuple of consecutive operations (op_i, op_j). In this tuple op_j is the successor operation of op_i. Each individual energy value $E(op_i, op_j)$ is caused by switching activity due to changing data pattern at the inputs of the component, time due to leakage currents in between the operations, and possible power-gating state transitions.

Thought the other way around, if the binding should be optimized, $E(op_i, op_j)$ estimates for every possible combination of operations can be combined in an energy-optimized manner as it is proposed in [Kru01]. All of these costs are summarized into one cost matrix per component type. This matrix is independent from resource constraints and it has to be computed only once for each component type and design. On the base of this matrix, energy-optimized bindings can be derived and allocation tradeoffs can be made.

In order to compute the energy values, two cases are distinguished. Operations within the diagonal of the matrix $(i = j)$ are exclusively mapped to a hardware resource of type t with the necessary bitwidth bw. In these cases, resource sharing is not used. The switching activity of each resulting component only depends on data pattern belonging to the operation op_i. As leakage power at RT-level is nearly independent of data pattern ([HEN06]), the consumed energy only depends on the elapsed time. Nevertheless, a cycle-wise power-management complicates the computation of leakage energy and may introduce an additional switching energy $E^{RT}_{SW}(t, bw)$.

The specific amount of energy $E(op_i, op_i)$ of this kind of operations during runtime is defined in Equation 5.17.

$$\begin{aligned} E(op_i, op_i) &= \sum_{n=1}^{N-1} (P^{RT}_{DYN}(t, bw, p_n, p_{n+1}) \cdot t_{cycle} \\ &\qquad + switch(t, bw, \Delta t) \cdot E^{RT}_{SW}(t, bw) \\ &\qquad + E_{LEAK}(t, bw, \Delta t)\ ) \\ whereas & \\ \Delta t &= (cycle(p_{n+1}) - cycle(p_n)) \cdot t_{cycle} \end{aligned} \tag{5.17}$$

The energy value $E(op_i, op_i)$ is computed by an estimation of all corresponding data pattern $pt(op_i) = \{p_1, p_2, ...p_N\}$ that are executed on the resource and the time in between. Function $P^{RT}_{DYN}(t, bw, p_n, p_{n+1})$ provides the data-dependent dynamic power that is consumed by the resource of type t and bitwidth bw for every pair of consecutive executed data patterns (p_n, p_{n+1}). It is multiplied with the cycle time t_{cycle} for the respective energy value. The second summand $switch(t, bw, \Delta t) \cdot E^{RT}_{SW}(t, bw)$ adds dynamic energy caused by a possible state transition from $active$ to $sleep$ state and back. State transitions are assumed to occur when the time between the two consecutive patterns Δt exceeds the break-even time t_{be} of the component. As defined in Equation 5.18, $switch$ evaluates to 1 if t_{be} (defined in Equation 2.6) is exceeded, else it is 0. The timestamp $cycle(p_n) \cdot t_{cycle}$ of a data pattern p_n is known because the simulation trace of the scheduled CDFG is fixed for a given benchmark and a user-defined clock speed.

$$switch(t, bw, \Delta t) = \begin{cases} 0 & \text{for } \Delta t \leq t_{be}(t, bw) \\ 1 & \text{for } \Delta t > t_{be}(t, bw) \end{cases} \tag{5.18}$$

The last summand $E_{LEAK}(t, bw, \Delta t)$ delivers the leakage energy of the overall cost metric. It is defined in Equation 5.19 and depends on the elapsed time $(cycle(p_{n+1}) - cycle(p_n)) \cdot t_{cycle}$ between two consecutive data pattern p_n and p_{n+1} that are executed on the same resource. The computation differs between leakage power in $active$ and $sleep$ state and it also covers the overall wake-up time $t_{wakeup}(t, bw)$ and the execution time $D^{RT}(t, bw)$. In the second case of Equation 5.19, leakage power of the $active$ state $I_{ACTIVE}(t, b)$ is used for the execution time of the first pattern and the wake-up time before the next pattern occurs. A remaining and decreased leakage power $I_{SLEEP}(t, b)$ is used for the time in between.

$$E_{LEAK}(t, bw, \Delta t) = \begin{cases} I_{ACTIVE}(t, bw) \cdot V_{DD} \cdot \Delta t \\ \qquad \text{for } switch(t, bw, \Delta t) = 0 \\ I_{SLEEP}(t, bw) \cdot V_{DD} \cdot (\Delta t - D^{RT}(t, bw) - t_{wakeup}(t, bw)) \\ \qquad + I_{ACTIVE}(t, bw) \cdot V_{DD} \cdot (D^{RT}(t, bw) + t_{wakeup}(t, bw)) \\ \qquad \text{for } switch(t, bw, \Delta t) = 1 \end{cases} \tag{5.19}$$

$E(op_i, op_i)$ is also called inter-iteration energy because each pair of two consecutive pattern p_n and p_{n+1} is executed in two consecutive passes or iterations of the CDFG.

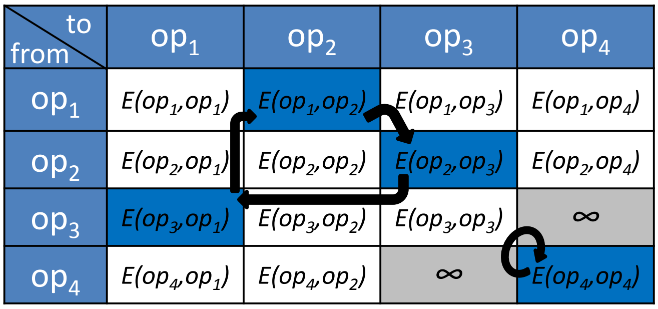
Within the rest of the matrix, resource sharing is assumed and two different operations op_i and op_j can be mapped to a single hardware resource. If op_i and op_j are consecutive operations the corresponding data pattern traces $pt_i = \{p_1^i, p_2^i, ..., p_N^i\}$ and $pt_j = \{p_1^j, p_2^j, ..., p_N^j\}$ are interleaved to a new trace of pattern tuples $pt = \{(p_1^i, p_1^j), (p_2^i, p_2^j), ..., (p_N^i, p_N^j)\}$. In these cases, no operations are executed in a control step between op_i and op_j. Again, each operation tuple causes a specific amount of energy $E(op_i, op_j)$ that is defined in Equation 5.20.

$$\begin{aligned} E(op_i, op_j) &= \sum_{n=1}^{N} (\ P_{DYN}^{RT}(t, bw, p_n^i, p_n^j) \cdot t_{cycle} \\ &\qquad + switch(t, bw, \Delta t) \cdot E_{SW}^{RT}(t, bw) \\ &\qquad + E_{LEAK}(t, bw, \Delta t)\ ) \\ whereas & \\ \Delta t &= (cycle(p_{n+1}) - cycle(p_n)) \cdot t_{cycle} \end{aligned} \tag{5.20}$$

With the use of resource sharing, $E_{LEAK}(t, bw, \Delta t)$ describes the leakage energy in dependence of the elapsed time Δt between two data pattern p_n^i and p_n^j of two operations op_i and op_j.

In general, $E(op_i, op_j)$ is not equal to $E(op_j, op_i)$ because of different idle periods between the operations and a different correlation of the data pattern. If op_j is scheduled after op_i, $E(op_i, op_j)$ is called intra-iteration energy because it is consumed within one execution of the CDFG. Operations that are scheduled in the same cstep are called *incompatible*. They cannot be bound to the same component and the corresponding energy values are set to infinite.

Example: For the given schedule of the design example, an energy cost matrix is computed for each operation type. Figure 5.6 represents the matrix for the adder operations within the example. The matrix also denotes an allocation of two adders and a possible binding. $E(op_3, op_4)$ and $E(op_4, op_3)$ are set to infinite because op_3 and op_4 are both scheduled in cstep 6.



to from	op_1	op_2	op_3	op_4
op_1	$E(op_1,op_1)$	$E(op_1,op_2)$	$E(op_1,op_3)$	$E(op_1,op_4)$
op_2	$E(op_2,op_1)$	$E(op_2,op_2)$	$E(op_2,op_3)$	$E(op_2,op_4)$
op_3	$E(op_3,op_1)$	$E(op_3,op_2)$	$E(op_3,op_3)$	∞
op_4	$E(op_4,op_1)$	$E(op_4,op_2)$	∞	$E(op_4,op_4)$

Figure 5.6: Cost matrix containing intra- and inter-energy costs for the adder operations in the example design.

In this example, the total cost for this binding and allocation is then given in Equation 5.21.

$$\begin{aligned} E_{total} &= \underbrace{E(op_1, op_2) + E(op_2, op_3) + E(op_3, op_1)}_{resource_1} \\ &\quad + \underbrace{E(op_4), op_4)}_{resource_2} \end{aligned} \tag{5.21}$$

Although conditional executions and loops within the CDFG are not represented in the matrix, the pattern wise energy determination delivers an accurate and fast foundation for the optimization in Sections 5.3.2 to 5.3.4.

Data-Independent Cost Function

Beside this complex pattern-dependent determination of energy estimates, a second very simple cost function has been developed that purely bases on the time in between two operations that can statically be derived out of the schedule. In this approach, the cost matrix contains cost values $C(op_i, op_j)$ as defined in Equation 5.22.

$$C(op_i, op_j) = \begin{cases} \frac{\Delta t}{t_{be}} & \text{for } \Delta t \leq t_{be}(t, bw) \\ \frac{\Delta t}{t_{be} - \#_{csteps} \cdot t_{cycle}} + (1 - \frac{t_{be}}{t_{be} - \#_{csteps} \cdot t_{cycle}}) & \text{for } \Delta t > t_{be}(t, bw) \end{cases} \tag{5.22}$$

$$whereas \;\; \Delta t = (cstep(op_j) - cstep(op_i)) \cdot t_{cycle}$$

This piecewise function is visualized in Figure 5.7. The main idea of this cost function is that the time Δt in between two operations should either be minimized (operations are clustered) or maximized (power-gating will lead to large leakage savings). The idle time is worst if it is equal to the break-even time. In this case, power-gating will not lead to any saving and the cost function evaluates to the highest cost of 1.

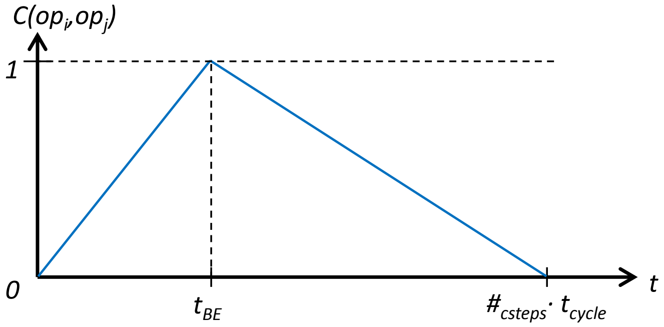


Figure 5.7: Data-independent cost function $C(op_i, op_j)$ based on the time in between op_i and op_j.

This approach has a significant advantage as it is independent on data pattern. Thus, the whole synthesis can be performed independent on testbenches and is much faster. Nevertheless, it has also drawbacks. At first, the data-dependency is neglected at all and an unfavorable binding may lead to poor data correlation with an increased dynamic power consumption. Secondly, component bitwidths cannot be handled in this approach.

5.3.2 Resource Constrained Lower Bound Computation

Based on the cost matrix, the approach of [KSJN99] is used to compute a lower bound binding for a given resource constraint. Due to the modified cost function, now the approach does not only determine the lower bound on the switching activity but on the overall energy demand.

The main idea of the approach is to cast the allocation and binding problem into a graph problem $G = (V, A)$ being $V = \{op_1, ..., op_n\}$ and $A = V \times V$. Each arc $(op_i, op_j) \in A$ is labeled with the cost value $E(op_i, op_j)$ as defined in Section 5.3.1. The optimization problem is then to cover all nodes with disjoint cycles with a minimum total cost under the constraint that each cycle contains exactly one backward arc. A backward arc describes an inter-iteration cost that is consumed between consecutive executions of the CDFG and can occur only once for each component. Each cycle then represents a set of operations bound to the same resource. The total cost is determined by the sum of all arc costs.

The authors show that the problem can be relaxed to the bipartite weighted matching problem which is solvable in polynomial time $O(n^3)$ by the Hungarian Method [PS82] where n is the number of functional units. A proof for the applicability of the Hungarian Method is also given by the authors. The relaxation is done by omitting the restriction to one backward arc. As a drawback, this allows invalid cycles such as $op_1 \rightarrow op_3 \rightarrow op_2 \rightarrow op_1$ in the example of Figure 5.6.

5.3.3 Relaxation of Lower Bound Binding

The binding defined by the lower bound may represent an invalid binding because of neglected precedence constraints and thus cannot be used as synthesis result. Nevertheless, this binding can be used as initial guess and the patching heuristic of [KSJ$^+$99] is able to relax conflicts and to find a valid binding.

The proposed patching heuristic splits all cycles with more than one backward arc into cycles with exactly one backward arc. In the given example, the invalid cycle $op_1 \rightarrow op_3 \rightarrow op_2 \rightarrow op_1$ would be split into $op_1 \rightarrow op_3 \rightarrow op_1$ and $op_2 \rightarrow op_2$.

In a second step, cycles are again patched together until the constrained number of resources is achieved. Each patching step selects those two cycles that will lead to the least increase in energy under the premise of only one backward arc. In the given example, cycles $op_2 \rightarrow op_2$ and $op_1 \rightarrow op_3 \rightarrow op_1$ might be patched to cycle $op_1 \rightarrow op_2 \rightarrow op_3 \rightarrow op_1$ and the contained operations are bound to the same resource. The patching heuristic might fail if there is no possibility to patch two cycles without including two incompatible operations. In such a case, random split and join steps are used to obtain a valid binding. The splitting is linear in the number of operations and the complexity of patching is in $O(n^2)$. The authors show that this heuristic serves as a good starting point for further improving heuristics and the final low energy binding is within 5% to the optimum.

5.3.4 Allocation Optimization

As already stated in the motivation, the best resource allocation is not automatically obtained by using the least necessary number of resources. The most obvious reasons are poor data correlation, unnecessary large input bitwidths caused by resource sharing, and increased multiplexer/interconnect costs. Low data correlation, for example, will introduce many switching bits and will increase the dynamic power. Power-gating may outweigh the overhead for an additional resource instance.

In the PowerOpt® synthesis framework the minimal necessary and maximal reasonable number of resources is first extracted from the scheduled CDFG for each resource type. The allocation optimization then computes the low power binding of Sections 5.3.1 to 5.3.3 for each possible resource constraint independently and the best solution is used in further synthesis steps.

5.3.5 Summary

The proposed functional unit binding and allocation applies the approach of [Kru01] under two power-gating aware cost functions. The first is data-dependent and considers dynamic, static, and state transition energy costs that occur in a trace of a simulated testbench. Thereby, operation clustering is only done if it is worthwhile in terms of an overall energy reduction. If a clustering would lead to an unfavorable binding with an increased dynamic power due to a poor data correlation, power-gating will not be applied. In contrast, the second cost function abstracts from data-dependencies and purely optimizes the operation clustering in order to improve the use of power down techniques.

5.4 Power-Management Controller Synthesis

Powering down processor cores or other system-level design parts implies the need for an idle time detection and suitable policies for entering the sleep mode such as time-based or branch-misprediction-guided approaches [HBS$^+$04]. In contrast, powering down RTL-components with a known and static schedule is easier. Knowing the utilization sequence of a functional unit as a result of the scheduling, allocation, and binding of Sections 5.2 and 5.3 as well as the power-gating models of Chapter 4, the cycle-accurate controller of the synthesized datapath can be extended to the power-management functionality. It will operate transparently to its environment as every FU is power-gated and woken up without an external control. Its policy is t_{be}-driven. Thus, every idle period in between two consecutive operations that exceeds the break-even time is exhausted for energy savings.

In general, a controller is defined as a Moore-based finite-state machine $A = (CS, \Sigma, \Omega, \delta, \lambda, cs_{idle})$ being CS the set of controller states, Σ and Ω the input and output alphabet describing bitvectors, δ the state transition function ($\delta : CS \times \Sigma \to CS$), λ the output function ($\lambda : CS \to \Omega$), and cs_{idle} the initial state. Each state $cs \in CS$ of the controller corresponds to a cstep in the schedule that is denoted as $cstep(cs)$ in the following. This function is a surjection because multiple controller states may belong to the same cstep. In each state, the Moore machine assigns its output bitvector and thereby controls the datapath. A transition to another state is conditionally executed. For the $sleep$-signal assignment analysis only the static structure of the controller is of importance. Thus, only the set of states CS and the set of possible transitions $T \subseteq CS \times CS$ are used in the analysis.

During synthesis of the controller output signals, a $sleep$ signal $sleep\_fu$ needs to be added for each power-gateable functional unit fu. This signal is connected to the sleep input of the PGS and outputs a logic-one (logic-zero) for a PMOS-based (NMOS-based) PGS. In the global idle state of the controller cs_{idle}, all FUs are power-gated to reduce the idle leakage of the overall datapath. Since the FUs need some time for wake-up, it has to be assured that the earliest operation will be no earlier than in the $(\lceil \frac{t_{wakeup}(fu)}{t_{cycle}} \rceil + 1)$-th controller state of every possible path traversing the states. If this is not the case in the recent controller, additional wait-states are inserted directly after the global idle state. Equation 5.23 defines the number of necessary wait states ws. $bind(fu)$ denotes the set of operations that are bound to this functional unit. The equation computes the maximum number of necessary wait states across all functional units and operations.

This is the only case of a possible schedule prolongation but it is necessary to guarantee sufficient time for powering up all datapath components.

$$ws = max\left(0, max\left(\bigcup_{fu \in FU}\left(\left\lceil \frac{t_{wakeup}(fu)}{t_{cycle}} \right\rceil + 1 - min\left(\bigcup_{op \in bind(fu)} cstep(op)\right)\right)\right)\right) \tag{5.23}$$

Every $sleep$ signal is initialized to power-gate the corresponding FU in every controller state as defined in Equation 5.24 .

$$\forall fu \in FU, \forall cs \in CS : sleep\_fu(cs) = ctrl\_sleep \tag{5.24}$$

Then, the functional unit is set to $active$ during times of operation. Thus, $sleep\_fu(cs)$ is overwritten with the active value for every operation $op \in bind(fu)$ for that $cstep(op) = cstep(cs)$ holds. If the operation delay exceeds the cycle time, op is a multicycling operation. In these cases, the FU needs to be ungated in the subsequent controller states of all possible paths of the control-flow too. Equation 5.25 gives a formal definition of these assignments.

$$\begin{aligned}
&\forall fu \in FU, \\
&\forall op \in bind(fu), \\
&\forall cs \in (cs_0, cs_1, ..., cs_n) \mid cstep(cs_0) = cstep(op) \wedge \\
&\qquad\qquad n = \left(\left\lceil \frac{D^{RT}(fu)}{t_{cycle}} \right\rceil - 1 \right) \wedge \\
&\qquad\qquad (cs_i, cs_{i+1}) \in T \ : \\
&\quad sleep\_fu(cs) = ctrl\_active
\end{aligned} \tag{5.25}$$

In addition, the wake-up time needs to be considered because a FU needs to be powered on early enough before it can be used. $\lceil \frac{t_{wakeup}(fu)}{t_{cycle}} \rceil$ denotes the number of cycles (and states) for that the controller needs to output the active state before an operation is executed. Since multiple paths can lead to the controller state of execution, the controller needs to power on the FU in all cases as described in Equation 5.26.

$$\begin{aligned}
&\forall fu \in FU, op \in bind(fu), \\
&\forall cs \in (cs_0, cs_1, ..., cs_n) \mid cstep(cs_n) = cstep(op) \wedge \\
&\qquad\qquad n = \left(\left\lceil \frac{t_{wakeup}(fu)}{t_{cycle}} \right\rceil - 1 \right) \wedge \\
&\qquad\qquad (cs_i, cs_{i+1}) \in T \ : \\
&\quad sleep\_fu(cs) = ctrl\_active
\end{aligned} \tag{5.26}$$

Now, every functional unit is available for times of operation. Nevertheless, sleep periods with a duration below the t_{be}-threshold will limit the energy saving. Thus, the controller needs to remain functional units ungated for these periods. Again, all paths the controller can traverse need to be considered as described in Equation 5.27.

$$\begin{aligned}
&\forall fu \in FU, \\
&\forall op_i, op_j \in bind(fu), \\
&\forall cs \in (cs_0, cs_1, ..., cs_n) \mid cstep(cs_0) = cstep(op_i) \wedge \\
&\qquad\qquad cstep(cs_n) = cstep(op_j) \wedge \\
&\qquad\qquad (cs_k, cs_{k+1}) \in T \wedge \\
&\qquad\qquad t_{be}(fu) > t_{cycle} \cdot \left(n - 1 - \left\lceil \frac{t_{wakeup}(fu)}{t_{cycle}} \right\rceil - \left\lceil \frac{D^{RT}(fu)}{t_{cycle}} \right\rceil \right) \ : \\
&\quad sleep\_fu(cs) = ctrl\_active
\end{aligned} \tag{5.27}$$

Note the definition of $ctrl\_sleep$ and $ctrl\_active$ in Equation 5.28 and 5.29.

$$ctrl\_sleep = \begin{cases} 0 & \text{for NMOS-based PGS} \\ 1 & \text{for PMOS-based PGS} \end{cases} \tag{5.28}$$

$$ctrl\_active = \overline{ctrl\_sleep} \tag{5.29}$$

With this definition, the controller is extended to a power-manager that enables FUs for the execution time of all bound operations. The example controller of the example in Figure 4.17 is pure sequential. To visualize the controller synthesis for control-flow dominated designs, a more complex example is given in the following.

Example: t_{wakeup}, t_{be}, and D^{RT} are assumed to be one cycle for this example. The controller state machine is shown in Figure 5.8. As it can be seen, the control-flow contains a branch in state s_1 that may result from an *if-then-else* clause of the behavioral description. The cycle $s_2 \rightarrow s_3 \rightarrow s_4 \rightarrow s_2$ denotes a loop due to a *for* or *while* statement. This loop may be exited prematurely from state s_3 as a *break* statement can terminate loops in a behavioral description. The controller power manages an adder component and the resulting $sleep\_add$ signal is color-coded for each controller state. For simplicity, multiplexer select and register enable signals are not visualized.

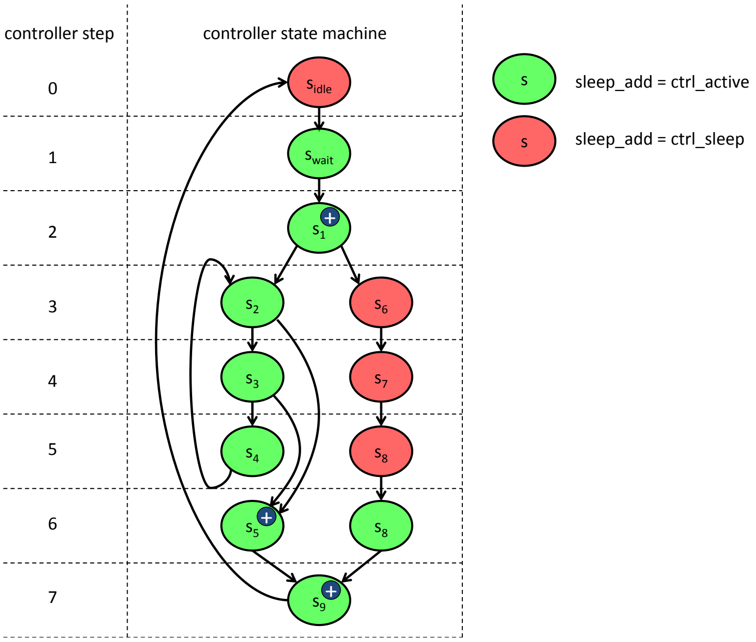


Figure 5.8: Exemplary controller state machine and $sleep\_add$ assignment.

As defined by Equation 5.23, one wait state s_{wait} has been inserted in between s_{idle} and s_1 in order to wake up the adder component duly for its operation in state s_1. Beside this operation, two more are bound to the adder that are executed in s_5 and s_9, requiring $sleep\_add$ to be active for these three states. As a consequence of Equation 5.26, $sleep\_add$ further needs to be active for the states s_{wait},s_2,s_3, and s_8 since all of these are direct predecessor states. At last, the adder remains active for the short idle period in state s_4 as defined by Equation 5.27.

5.5 Summary

An automated consideration of the power-gating technique during the high-level synthesis requires an automated sleep transistor sizing. Section 5.1 presents a delay-dependent sizing approach that uses the power-gating models of Chapter 4 in addition to the delay model of [HHN07]. The approach finds the smallest necessary sleep transistor size to meet a user-constrained delay increase.

The power-gating models are further used for heuristic optimizations of the high-level synthesis tasks. An ILP-based scheduler has been proposed in Section 5.2 that implements an operation clustering. Its cost function reduces the number of control steps with an increased resource demand. In doing so, operations of the same type are scheduled closely together and continuous idle times are created in between the operation clusters. The subsequent binding of Section 5.3 maps the operations to physical resources. This binding also heuristically clusters operations if it is worthwhile from an energetic point of view. Allocation tradeoffs are also executed by iterating the binding approach with different resource constraints.

In the end, the controller synthesis is extended to power-management capabilities in Section 5.4. Thereby, idle times of functional units are analyzed and appropriate sleep signals are created. Thus, the controller exploits the idle periods by power-gating functional units on an individual basis.

6 Experimental Environment and Assessment

The purpose of the power-gating characterization is the model application during high-level synthesis. Thereby, thousands of model invocations are necessary for a design-space exploration. Thus, the model estimation needs to be fast. Secondly, the DSE requires a high relative accuracy for providing meaningful tradeoffs. The relative accuracy reflects in the variation of model errors and is even more important than the absolute accuracy. Nevertheless, the latter is of importance, too, because the models are used in a concatenated manner. Thus, the overall error chain has to be kept small to not escalate overall estimation errors.

The experimental assessment of the model accuracy and synthesis improvement need a fixed and well defined environment. For this reason, at first a technology selection is done for which the evaluation is performed and all model parameters are constrained to a set of discrete values or a continuous range. The following evaluation then distinguishes between the pure model evaluation, a presentation of the power-management adoption at system level, and a presentation of the synthesis improvements caused by its power-management awareness. At the end of this chapter the compliance of the power-management integration with industrial power standards is presented and a summary is given.

6.1 Technology Selection and Parameter Ranges

To validate the correctness of the modeling approaches and to prove its universality, a selection of technologies and parameters has been made. Beside different technology node sizes, it is important to cover different process corners. Additionally, MTCMOS technologies should be considered in order to cover sleep transistor implementations in both standard- and high-threshold design.

Technology name	Size	Process corners			Threshold voltage	
		slow-slow	typical-typical	fast-fast	Standard V_{TH}	High V_{TH}
Nangate $45nm$ [Inc]	45nm	x	x	x	x	x
Industrial LP $45nm$	45nm	0	x	0	x	x
Industrial LP $65nm$	65nm	0	x	0	x	x

Table 6.1: Semiconductor technology selection

Table 6.1 lists three different technologies for which the characterization was done. The Nangate free $45nm$ open source digital cell library technology [Inc] is a general purpose (GP) technology based on predictive technology modelcards of the NIMO Group, Arizona State University [IaASU]. It is freely available and is widely used in the scientific context. It offers three *even* process corners (slow-slow, typical-typical, and fast-fast) that are all evaluated separately. *Even* means that both PMOS and NMOS devices are equally affected by variations of fabrication parameters. Further, it is an MTCMOS technology and thus it includes both, standard- and high-V_{TH} transistors. The industrial $45nm$ and $65nm$ technologies are also MTCMOS technologies but their process corner is restricted to the typical case in this evaluation. Additionally, and in contrast to the Nangate free $45nm$ open source digital cell library technology, they are both LP specialized technologies. As analyzed in Section 2.2.2, these LP techniques inherently have lower leakage currents and the resulting power-gating break-even time is in another order of magnitude.

Table 6.2 lists all parameters of the characterization process and its parameter ranges. The supply voltage is constrained by the technology whereas the surrounding temperature is constrained by reasonable values. The gate voltage of the sleep devices that is used in SCCMOS techniques to enforce a cutoff is specified as an offset to the supply or ground voltage. It is in the range $0V$ to $0.1V$ and thus the sleep signal is in the range of $[V_{DD}; V_{DD} + 0.1V]$ for PMOS-based PGSs and $[GND; GND - 0.1V]$ for NMOS-based PGSs. The sleep transistor width is constrained to a maximum of 10% of the gated component size W_{RT} as argued in Section 4.2.5. The characterization is also constrained to functional RTL units that are available and supported by OFFIS's PowerOpt®. Their bitwidths ranges from 4 to 32 bits in 4 bit steps.

Parameter	Symbol	Ranges
supply voltage	V_{DD}	$[0.9V; 1.3V]$
temperature	T	$[27°\text{C}, 127°\text{C}]$
gate voltage offset	V_{GB}	$[0.0V; 0.1V]$
sleep transistor width	W_{ST}	$[0\%; 10\%]$ of gated component size W_{RT}
RT-level components	RT	$add\_fast, add\_small, dec\_fast, dec\_small, inc\_fast,$
		$inc\_small, sub\_fast, sub\_small, mult\_fast, mult\_small$
bitwidth	bw	$4, 8, 12, 16, 20, 24, 28, 32$

Table 6.2: Parameter ranges

6.2 Power-Gating Model Evaluation

During model generation, a lot of methods have been used to compact and ease the resulting models. This includes compression of lookup tables, exhaustive interpolations in multiple dimensions, parameter separation, (non)-linear regression techniques, and simplifications to speed up the model generation. For this reason, the evaluation has to show the quality and the performance improvements compared to reference estimates. Since silicon measurements are not available, the reference estimates are obtained by Spice-based analog circuit simulation measurements. This is an established approach in the scientific as well as industrial area. Setting and measuring voltages and currents at all transistor and component connections is done as described in the appropriate modeling sections.

The entire characterization is done via Synopsys HSPICE® version A-2008.03-SP1 and is executed on a general purpose Intel Core2Duo machine at $3Ghz$. It lasts about one day per semiconductor technology whereas transient simulations of the E_{SW}^{RT} and t_{wakeup}^{RT} models make up 98% of the time. Of this, more than 50% is attributable to the two multiplier components. This illustrates the limits of circuit simulations and underlines the hardness of predicting the application of power-gating for huge components.

For presenting the absolute and relative accuracy of the models, a Monte-Carlo evaluation has been applied covering all parameters in the aforementioned ranges and three error measures have been computed: the maximum relative error for over- and underestimation $XRE \uparrow$ and $XRE \downarrow$ (to state the absolute model accuracy), the mean absolute relative error $MARE$, and the relative standard deviation σ_{rel} (to state the relative model accuracy). In the following, the evaluation results of the models are presented.

6.2.1 Evaluation Results of the Sleep Transistor Leakage Models

In the I_{OFF}^{ST} model the supply voltage range is sampled with a rate of $0.1V$, the temperature with 20°C, and the gate voltage with a rate of $0.1V$, resulting in a total of $5 \cdot 6 \cdot 2 = 60$ sampling points for each PGS and technology. Furthermore, the characterization has been done for an isolated PGS circuitry with a channel width of $1\mu m$. Figure 6.1 shows the model errors.

As it can be seen, the remaining gate- and subthreshold-leakage currents can be predicted with an average $MARE$ below 1% and a maximum error of 6.5%. On top of this error, the model simplification $V_{DROP\text{-}OFF}^{ST} = V_{DD}$ will induce an additional error in terms of an overestimation of up to 15%.

Conducting sleep transistors are again modeled at a supply voltage sampling rate of $0.1V$, whereas the gate voltage disappears as a parameter. Since pure gate-leakage currents of I_{ON}^{ST} do only slightly depend on the temperature, a wider sampling step of $50°\mathrm{C}$ can be used for this model, leading to a total of $5 \cdot 3 = 15$ sampling points. Nevertheless, the temperature remains a parameter during modeling as it may gain importance in future semiconductor technologies because of increasing pn-junction leakage currents being more dependent on the temperature.

Figure 6.2 presents the I_{ON}^{ST}-model evaluation results. The $MARE$ is about 4% for the Nangate free $45nm$ open source digital cell library and 1% for the two industrial technologies. In all cases, the model tends to overestimate the gate-leakage currents because of the quadratic impact of V_{GS} and V_{GD} (cf. Equation 2.2) while the model linearly interpolates between two adjacent sampling points. Increasing the supply voltage sampling rate would reduce this overestimation but also enlarge the model. Additionally, the maximum error is only 18% for the Nangate and even below 4% for the industrial technologies. As argued in Section 6.1 and also demonstrated in Figure 2.8, the industrial LP technologies have inherently lower leakage currents. Since for all models the same number of sampling points is used, the model interpolation of the LP technologies needs to cover a smaller dynamic range and results in a better model evaluation.

6.2.2 Evaluation Results of the Voltage Drop Models

Figure 6.3 presents the maximum, mean, and standard deviation errors of the $V_{DROP\text{-}ON}^{ST}$ model. The parameters temperature and supply voltage are sampled with a step width of 20°C and $0.05V$. As presented in the charts, the occurring voltage drop can be predicted with an average error of $1-5\%$ with maximum overestimates of 25%. Secondly, the errors of HVT- and double-gating schemes are larger than those of SVT- and single-gating schemes because these schemes have higher on resistances and increase the

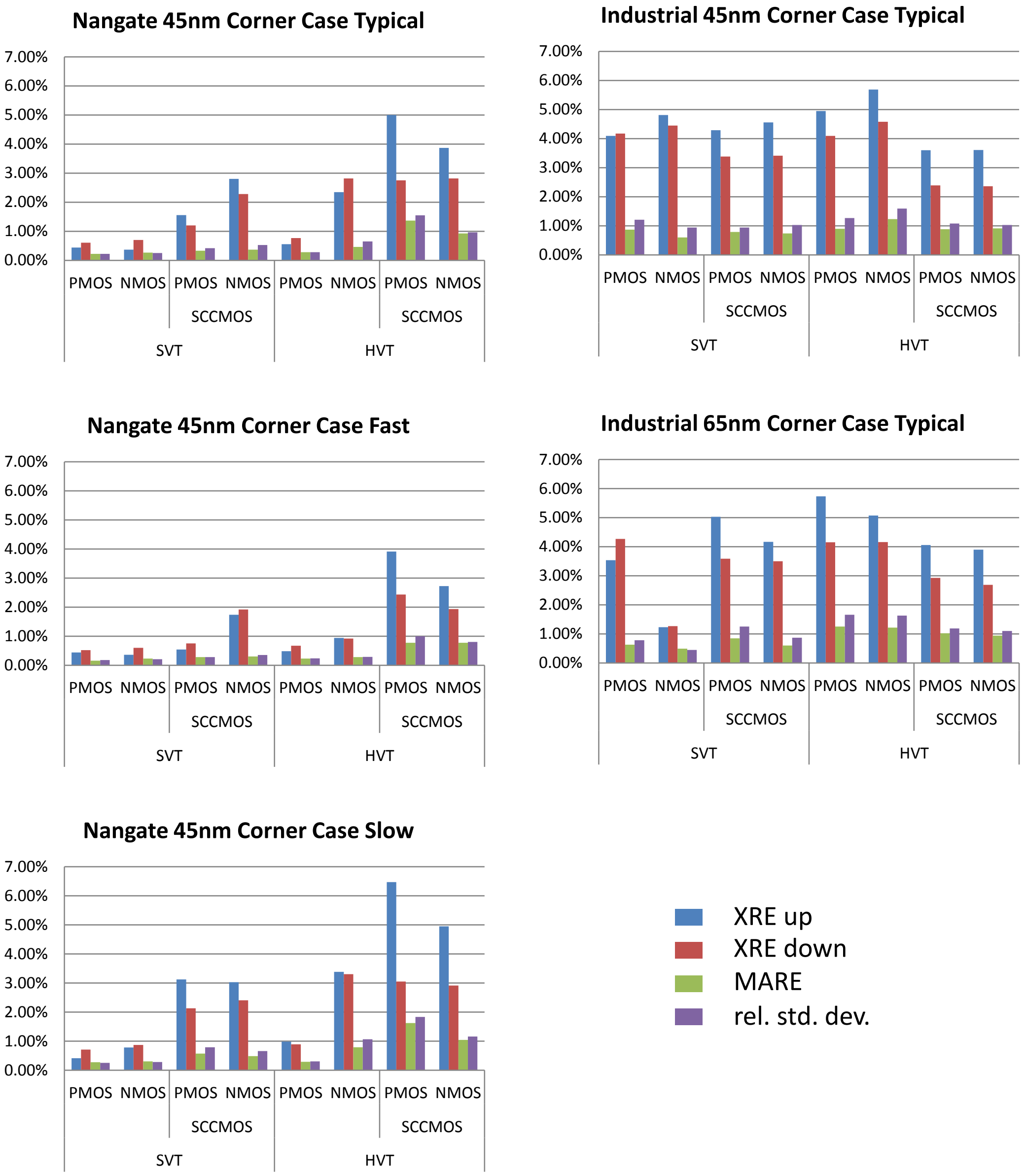


Figure 6.1: Errors of the gate- and subthreshold-leakage model I_{OFF}^{ST} for locking sleep devices

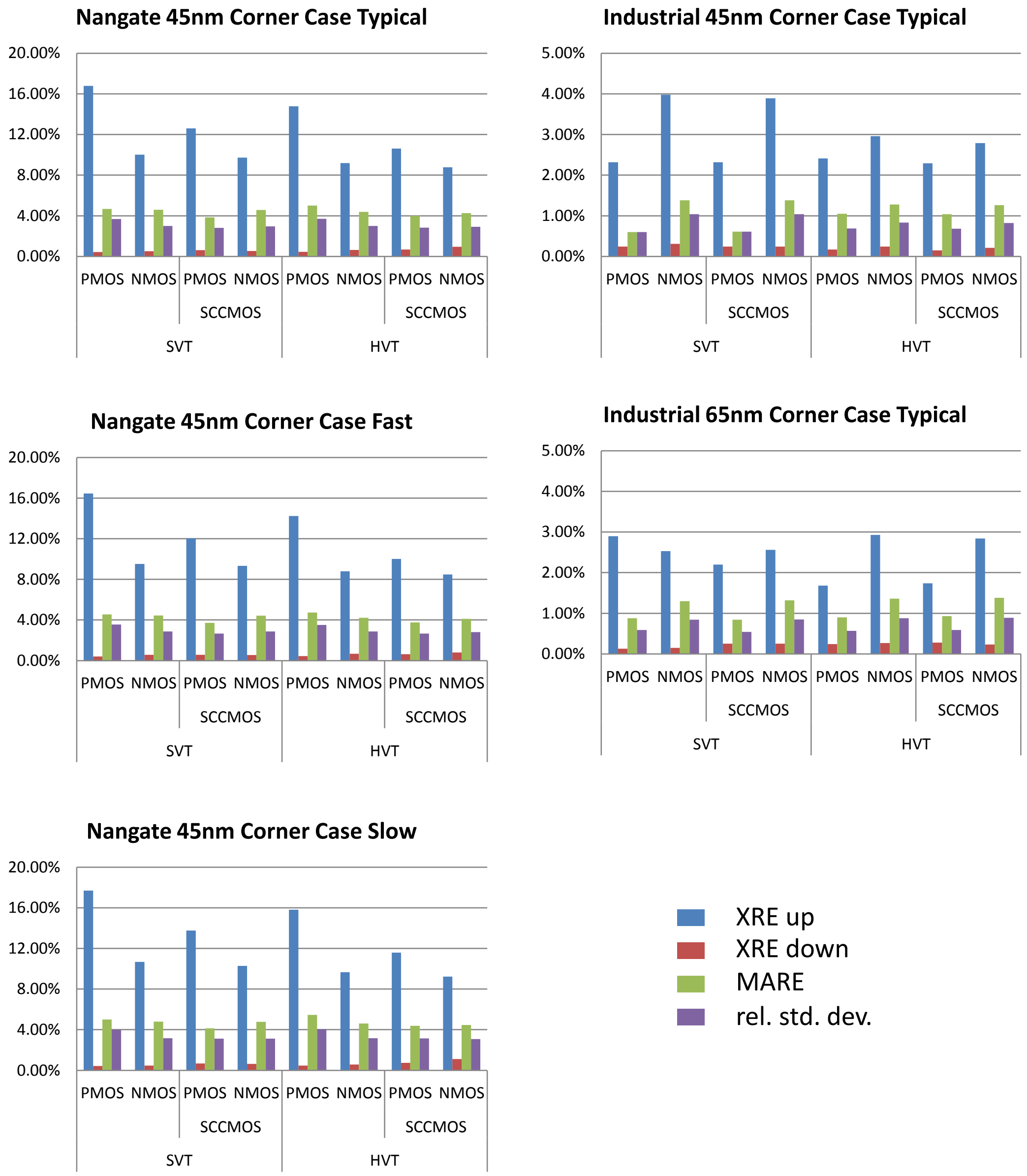


Figure 6.2: Errors of the gate-leakage model I_{ON}^{ST} for conducting sleep devices

voltage drop dynamic that needs to be interpolated by the model. Underestimates that would play down the presence of sleep devices, are limited to 5% maximum.

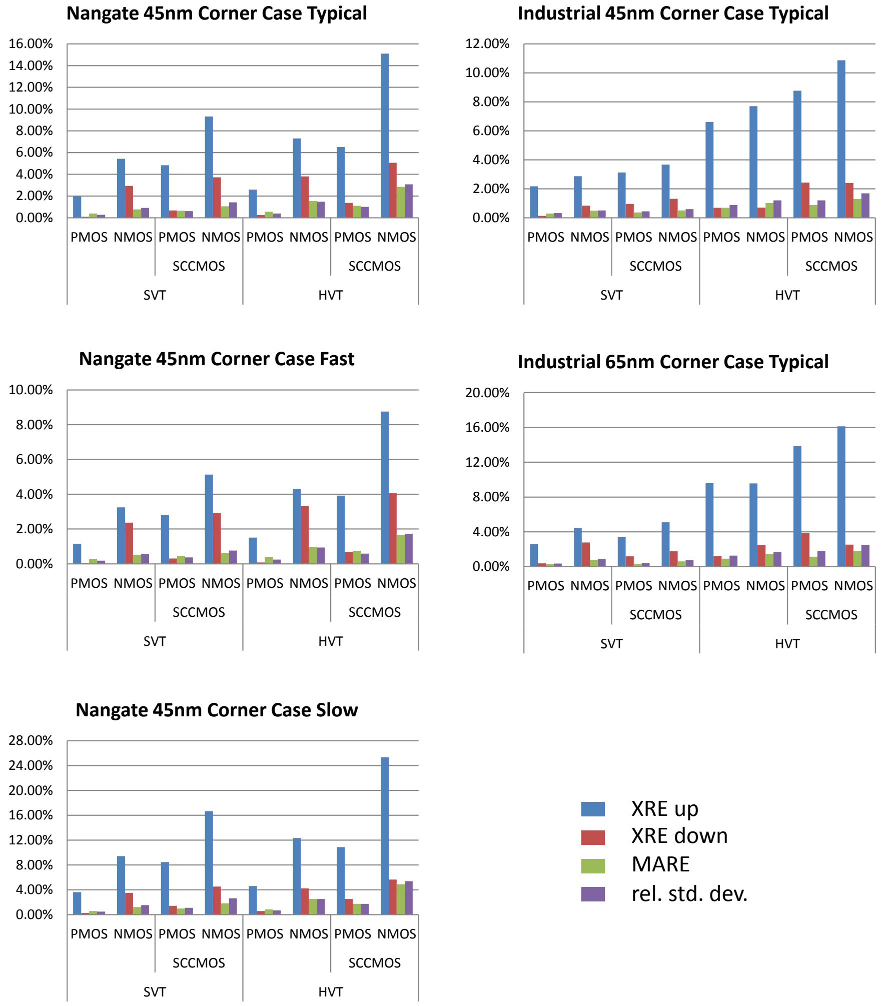


Figure 6.3: Errors of the voltage drop model $V_{DROP\text{-}ON}^{ST}$ for conducting sleep devices

$V_{DROP\text{-}OFF}^{ST}$ is evaluated as presented in Figure 6.4. For the parameters V_{DD}, T, V_{Gate}, and W_{ST} the model consists of a $5 \cdot 2 \cdot 3 \cdot 6 = 180$-point measuring field. With a mean absolute relative error below 1.5% and a relative standard deviation of 2.1% in maximum across all technologies, the accuracy of the model is very high. However, this accuracy is also necessary because the estimates serve as input to the E_{SW}^{RT}-model and highly impact its prediction as analyzed in Section 4.2.4.

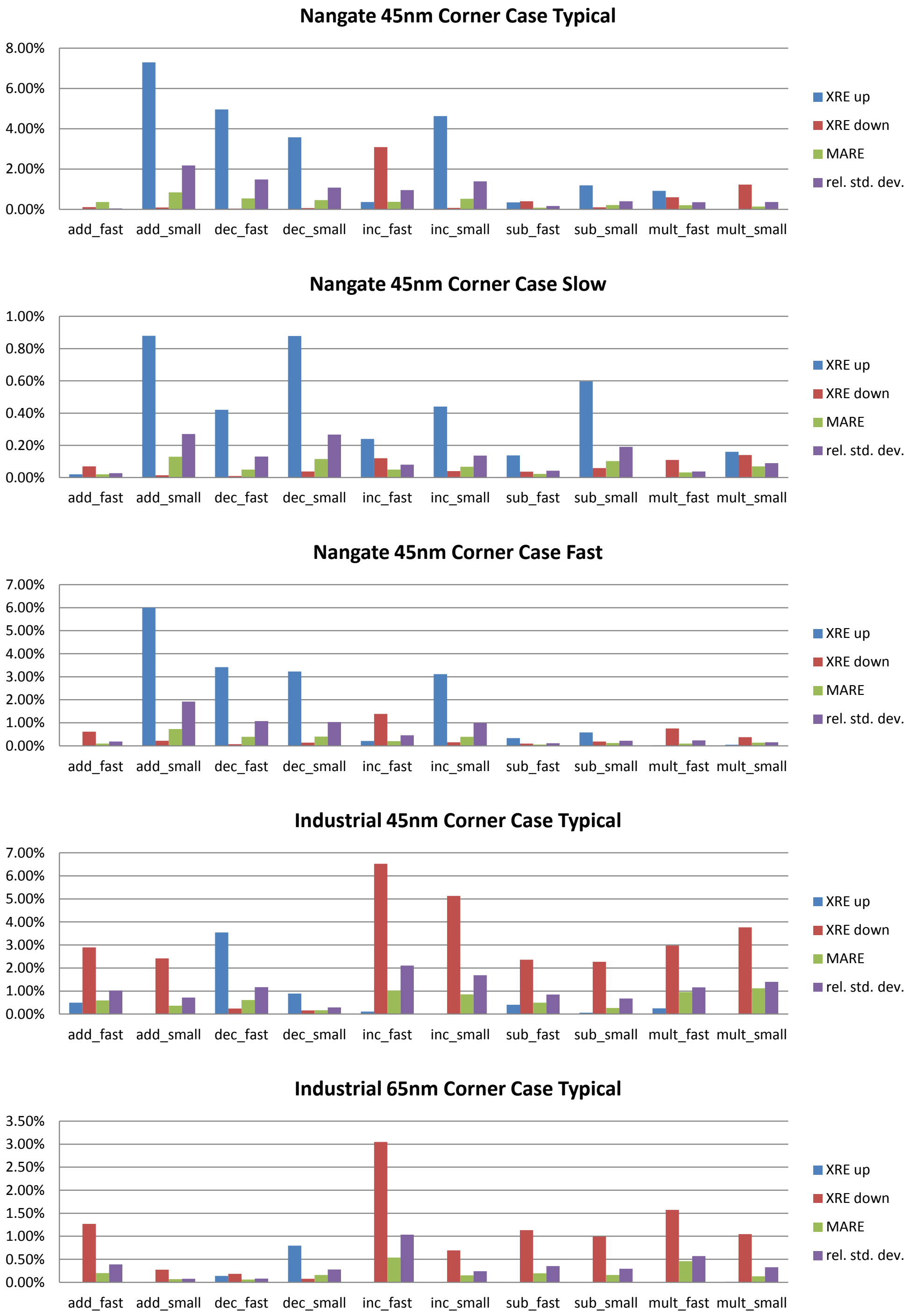


Figure 6.4: Errors of the voltage drop model $V^{ST}_{DROP\text{-}OFF}$ for locking sleep devices

6.2.3 Evaluation Results of the State Transition Energy Models

The state transition energy models E_{SW}^{ST} and E_{SW}^{INV} are characterized at fine-grained supply- and gate-voltage sampling rates and are then compressed to step widths of $0.2V$ and $0.1V$. Thus, the models consist of only 6 sampling points for each PGS. As it can be seen in the E_{SW}^{ST} evaluation of Figure 6.5, maximum errors of 7% occur in the industrial technologies while mean errors are throughout below 2%. In the Nangate technology, maximum errors are even below 2.5%.

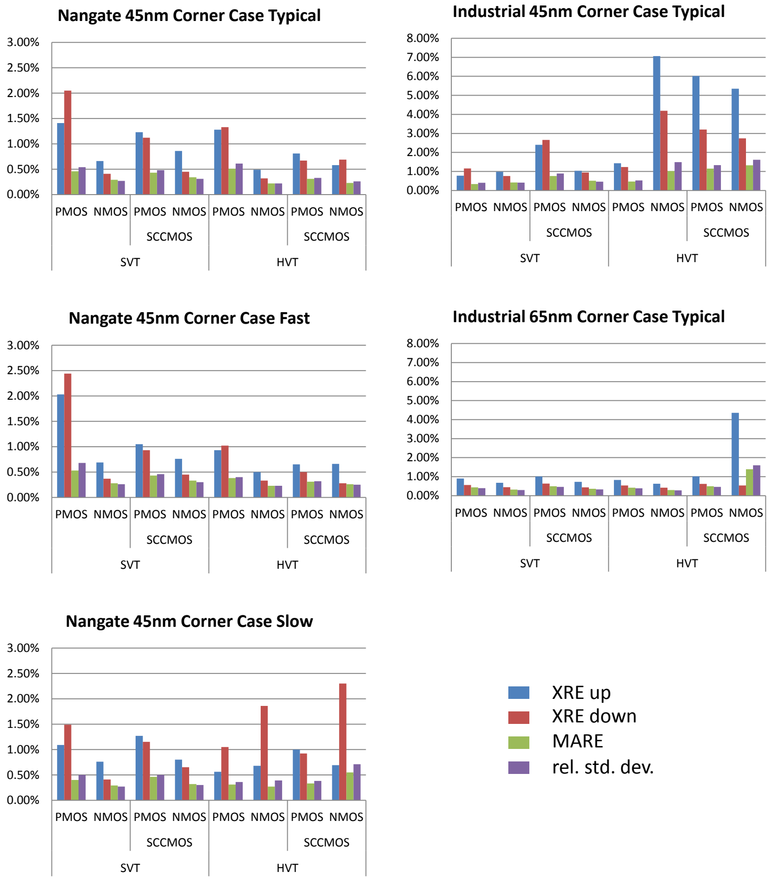


Figure 6.5: Errors of state-transition induced sleep transistor dynamic-energy E_{SW}^{ST} model

The inverter state transition energy model E_{SW}^{INV} for predicting buffer chains is evaluated in Figure 6.6 for up and down transitions. As shown, the errors are throughout below 3% for all technologies.

The most effort for model evaluation has been spent for the E_{SW}^{RT} model because some large multiplier components are not simulatable in high bitwidths or in combination with some PGSs. In these cases, Synopsys HSPICE® fails in simulating the circuits due to a high memory demand and failing convergence analyses. To provide a meaningful analysis of the model, a Monte-Carlo based evaluation performs a total of 1000 randomly chosen transient simulation runs, lasting about two weeks of computation time. The presented errors base on about 93% of the simulation runs that have been finished successfully and include

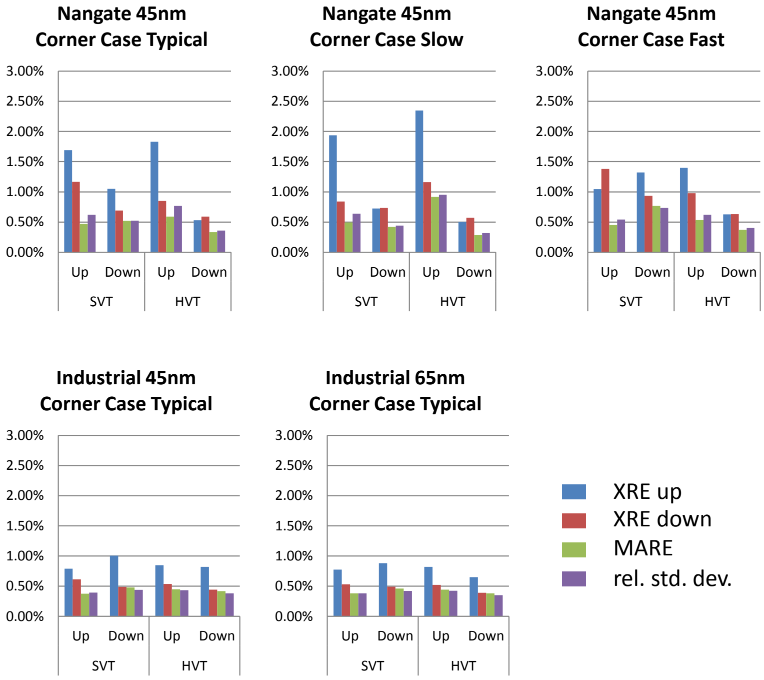
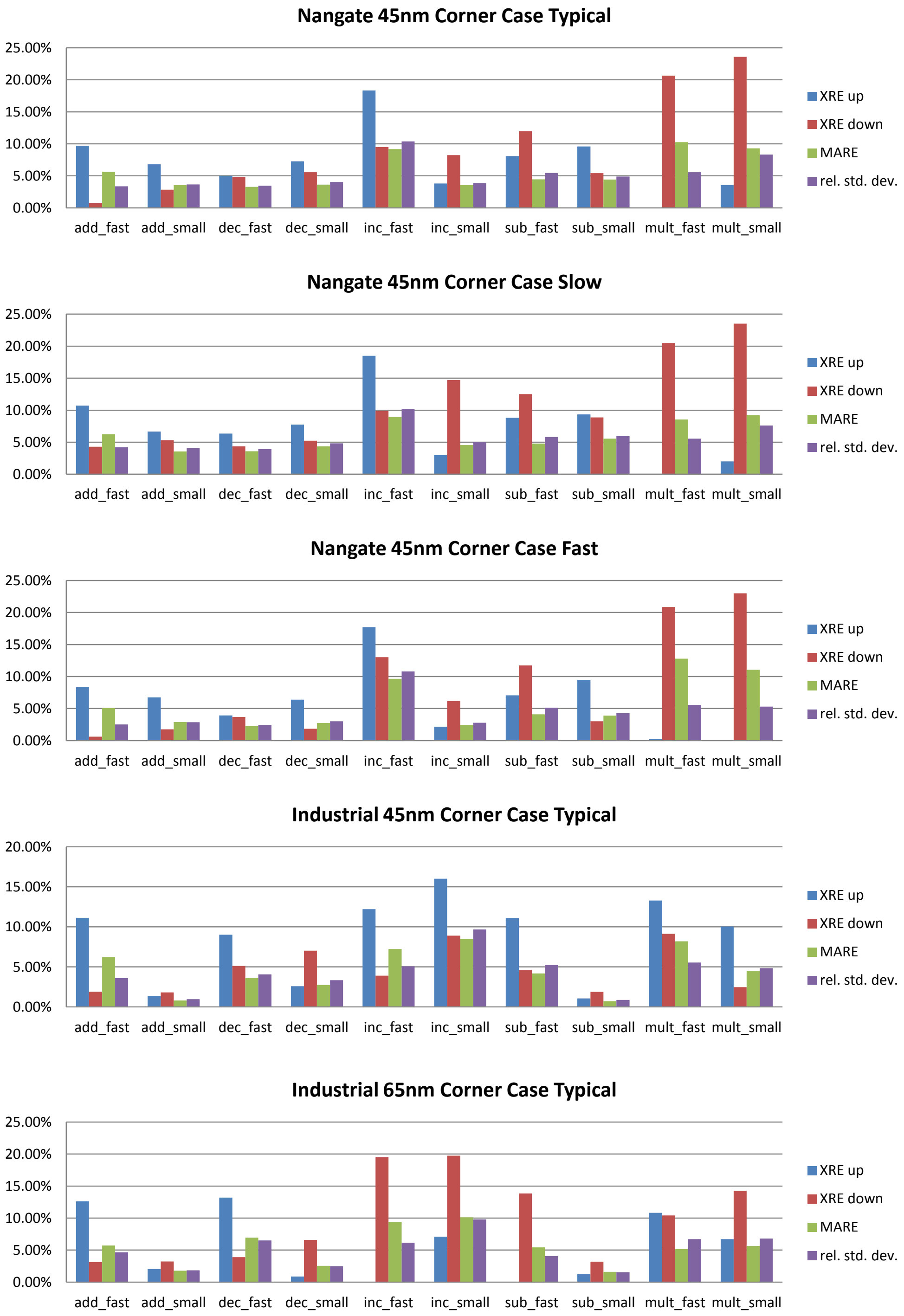


Figure 6.6: Errors of state-transition induced inverter dynamic-energy E_{SW}^{INV} model

all model errors induced by the model representation and required interpolation. Especially, the bitwidth-scaling and PGS selection is reflected in the evaluation. Furthermore, the $V_{DROP\text{-}OFF}^{ST}$-model cascading effect is covered in this evaluation. It is therefore not surprising that peak E_{SW}^{RT} errors have been observed at peak voltage drop errors because of their super linear dependency.

Figure 6.7 summarizes the evaluation results per technology and RTL-component. Mean absolute relative errors below 10% and mostly even below 5% have been determined for the dominant part of components. Nevertheless, the quality varies. For example the incrementer component inc_fast in the Nangate technology is conspicuous with its higher peak errors and standard deviations. Secondly, the model tends to underestimate the state transition energy for the two multiplier components in different technologies. This suggests the conjecture that, analogous to the non-linear wake-up delay of multipliers, the matrix structure also causes super linearly increasing wake-up energies. Nonetheless, the maximum errors are reasonable below 25% and no further modeling effort has been spent for these components. Moreover, the temperature is set to the upper bound during characterization. Thus, the models do only predict upper bound estimates.

The interpolation table size of the model is $5 \cdot 2 \cdot 5 = 50$ points for the model parameters supply voltage, voltage drop, and sleep transistor size.


Nangate 45nm Corner Case Typical

Nangate 45nm Corner Case Slow

Nangate 45nm Corner Case Fast

Industrial 45nm Corner Case Typical

Industrial 65nm Corner Case Typical

Figure 6.7: Errors of state-transition induced dynamic energy model E_{SW}^{RT} for a set of RTL-components

6.2.4 Evaluation Results of the State Transition Delay Models

The buffer chain delay is estimated by triggering the D^{INV} model multiple times in accordance to the buffer chain structure. The delay model errors are presented in Figure 6.8. The two dimensional model table only consists of 10 sampling points and due to the super linear delay increase but linear interpolation, the model tends to overestimate. Nevertheless, the errors are 7% at the maximum and below 2% in average.

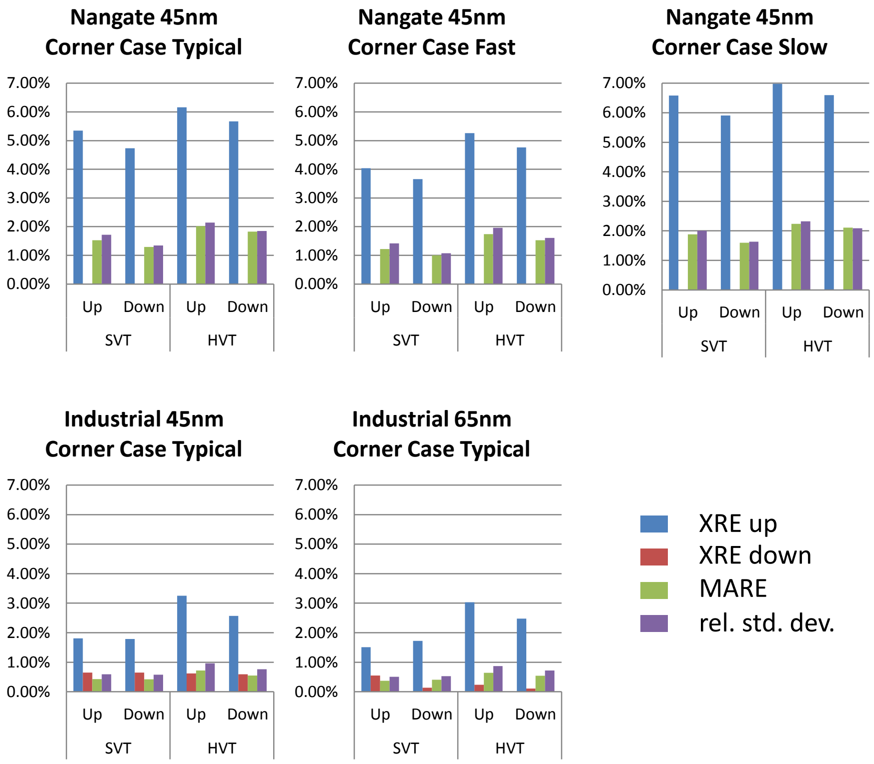
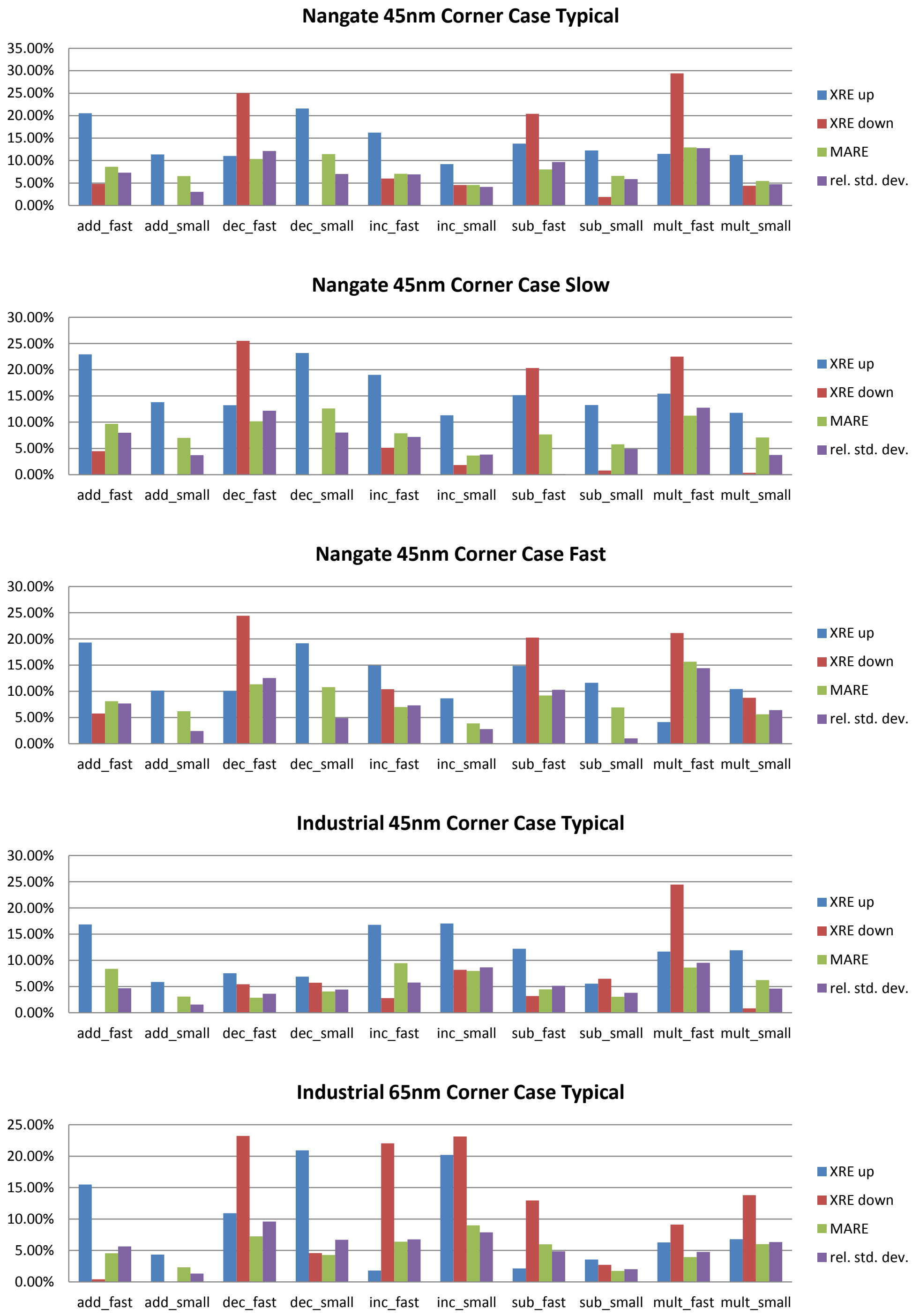


Figure 6.8: Errors of state-transition induced delay model D^{INV} for an inverter within the buffer chain

Figure 6.9 presents the wake-up time model evaluation. As it can be seen, the mean average errors are mainly below 10% but peak errors vary a lot and range up to 29% for the multiplier component in the typical process corner of the Nangate technology. Especially the wake-up delay prediction for the small-type components performs better compared to the fast-type components throughout all technologies.

The interpolation table size of the model is as small as in the E_{SW}^{RT} model because it bases on the same characterization runs.


Nangate 45nm Corner Case Typical

Nangate 45nm Corner Case Slow

Nangate 45nm Corner Case Fast

Industrial 45nm Corner Case Typical

Industrial 65nm Corner Case Typical

Figure 6.9: Errors of state-transition induced wake-up model t^{RT}_{wakeup}

6.2.5 Coupled Model Evaluation

Most of the proposed sub-models contribute to a design's energy demand. In order to combine the sub-model errors to an overall error metric their weighting is of importance. Assuming an imaginary design that is manufactured in a semiconductor technology with no leakage currents and only containing functional units with a workload of 1, the overall error is completely defined by the dynamic power model. In contrast, the error of a leaky design with a negligible workload is dominated by the leakage model error.

As the state transition energy model has shown the largest error of 25% for multiplier components, this can also be seen as a pessimistic upper bound error for all models proposed in this thesis that are necessary for estimating a design's energy consumption. However, typical designs do not solely consist of multiplier components and the other models have an impact, too. As a result of this model composition, the overall expected maximum error depends on the design and on its execution trace defined by the testbench.

In order to weight the $XRE \uparrow$ and $XRE \downarrow$ errors of the FUs state transition models, the datapath composition within the set of benchmarks being proposed in Section 6.3 has been analyzed. In the considered designs, adder components contribute 13% to the total state transition energy, subtractors 5.5%, incrementers 4.2%, and multipliers contribute the dominating part of 77.3%. For the components that are manufactured in the Nangate technology at the fast process corner (see Figure 6.7 for evaluation results) this results in a weighted maximum $XRE \uparrow$ and $XRE \downarrow$ error of 1.5% and 18.5% for the E_{SW}^{RT} model. The models' standard deviations are combined by the propagation of uncertainty law. Equation 6.1 computes the combined standard deviation $\sigma_{rel\_f} = 4.13\%$ for the E_{SW}^{RT} model and for the considered benchmarks.

$$\sigma_{rel\_f}(E_{SW}^{RT}) = \sqrt{(\sigma_{rel}(E_{SW}^{add\_small}) \cdot 13\%)^2 + (\sigma_{rel}(E_{SW}^{sub\_small}) \cdot 5.5\%)^2 + (\sigma_{rel}(E_{SW}^{inc\_small}) \cdot 4.2\%)^2 + (\sigma_{rel}(E_{SW}^{mult\_small}) \cdot 77.3\%)^2} = 4.13\% \tag{6.1}$$

To provide a statement on the coupled model error the averaged sub-model composition has been analyzed as shown in Figure 6.10. It can be seen that 30.4% of the total energy consumption is caused by classical dynamic power. 38.7% of the total energy is due to leakage currents and 30.9% is caused by power-gating state transition costs. As shown, the leakage-induced fraction is further divided into four contributors and the state transition costs are divided into RTL-component-, sleep transistor-, and buffer-costs.

If the maximum sub-model errors are scaled by this distribution the weighted $XRE \uparrow$ and $XRE \downarrow$ errors of all involved sub-models are 1.7% and 15.7%. Together with the maximum leakage model error of 25% [Hel09] and the maximum dynamic power model error of 34% [JKSN99] this results in an even higher weighted maximum $XRE \uparrow$ and $XRE \downarrow$ errors of 19.8% and 24.6%.

At last the standard deviations of the proposed sub-models are combined by the propagation of uncertainty law resulting to a standard deviation of $\sigma_{rel\_f} = 3.41\%$. Thus, the models are as accurate as the leakage models of [Hel09] that have been evaluated with a relative standard deviation of $3.6\% - 6.9\%$

In addition, many of the model parameters such as the supply voltage, the temperature, and sleep transistor size are fixed for different synthesis iterations. For this reason, the relative accuracy will even be higher and design tradeoffs will be very accurate.

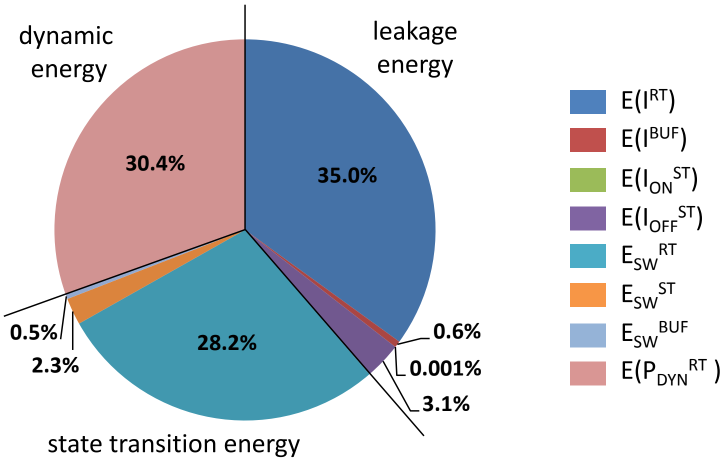


Figure 6.10: Composition of the total energy estimation to all involved sub-models

6.2.6 Evaluation of Process Variation on Power-Gating

The Nangate semiconductor technology offers circuit level device models of three process corners. These corners represent the extremes of parameter variations within which a circuit must operate correctly. Thus, the corners cover the overall spectrum from slowest to fastest possible devices. In this section, the impact of process variation on power-gating is evaluated exemplarily for a single RTL component.

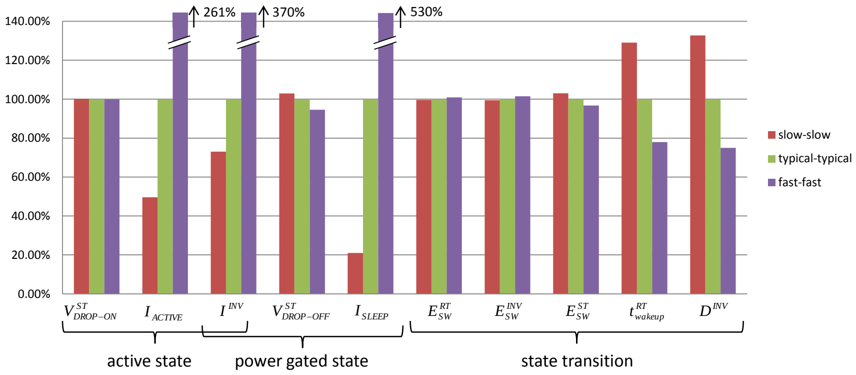


Figure 6.11: Normalized model estimates for different process corners to analyze the process variation impact on power-gating

Figure 6.11 presents model estimates for power-gating relevant parameters that are normalized to the typical operating case. As it can be seen, the voltage drop across the sleep transistor and the state transition energies do only slightly change. This is completely different for the leakage currents and timing behavior. As expected, power-gated components that are fabricated at the fast process corner wake up faster but on the other hand they cause a lot more leakage currents. In relative terms, the I_{ACTIVE} current of the fast process corner is 2.6 times as high as of the typical corner but, while being power-gated, the remaining leakage current I_{SLEEP} is even 5.3 times as high. But in absolute terms, the amount of reduced leakage is much higher for the fast corner. Together with the almost constant state change energies (E^{RT}_{SW}, E^{ST}_{SW}, and E^{INV}_{SW}), power-gating becomes even more advantageous for designs fabricated at the fast and less advantageous at the slow process corner.

The break-even time analysis of Figure 2.8 for the Nangate $45nm$ technology at fast process corner results in t_{be} times of below $50ns$ which is less than half of the typical-case break-even time.

6.2.7 Summary

The model evaluation has shown mean average relative errors and relative standard deviations at a low single-digit range throughout all leakage and voltage drop models. This observation also holds for the state transition energy and delay models except for few components that show mean errors of up to 12%.

The absolute accuracy has been evaluated by maximum errors of 18% for the I^{ST}_{ON} leakage model and 25% for the $V^{ST}_{DROP\text{-}ON}$ voltage drop model. As these errors represent overestimates, the predicted efficiency of power-gating may be worsened but not improved. The models I^{ST}_{OFF}, $V^{ST}_{DROP\text{-}OFF}$, E^{ST}_{SW}, and E^{INV}_{SW} show maximum over- and underestimates below 8% throughout all technologies and PGSs. The highest maximum errors of 25% and 29% showed up in the E^{RT}_{SW} and t^{RT}_{wakeup} models for isolated RTL components. As argued in Section 6.2.3 these models provide upper bound estimates as they have been characterized with the upper bound temperature.

To sum up, it can be said that the efficiency of power-gating or the synthesis improvements are not beautified by the proposed models. For the purpose of high-level tradeoffs both, the absolute and relative accuracy, are perfectly adequate and the speed improvement is a major model feature. In comparison to a single analog circuit simulation that may take up to several hours, the pre-characterized models can provide thousands of estimates per second as required by the design-space exploration.

6.3 System-level Power-Management Evaluation

Every RTL component within a datapath contributes a small fraction to the I_{ACTIVE} and I_{SLEEP} currents of a design and has its individual wake-up energy and time. Further, at RT-level, each component has its own break-even time. At system-level, all of these parameters merge to one overall effectivity-metric of power-gating and result in one global break-even time that has to be exceeded if all components are cut off simultaneously. This section will evaluate this system-level view of power-management in relative comparisons and absolute numbers against the background of possible savings, impact of parameters, and overhead costs in terms of area and power.

Table 6.3 lists design examples and characteristic parameters such as their functional unit datapath composition required by the minimal allocation after synthesis and amount of controller steps. To all of the designs power-gating has been applied with HVT NMOS sleep devices that are most common in today's practice. The fourth and fifth column of Table 6.3 show absolute I_{ACTIVE} and I_{SLEEP} numbers of the designs at a fixed supply voltage of $1.0V$, an ambient temperature of 27°C, and on the base of the Nangate $45nm$ technology at typical process corner. I_{SLEEP} and I_{ACTIVE} are restricted to the functional

Design name	Minimum allocation	$\#_{csteps}$	I_{ACTIVE} of FUs	I_{SLEEP} of FUs
FDCT	4 add, 3 sub, 8 mult	7 csteps	$93.1\mu A$	$2.2\mu A$
JPEG this	1 add, 1 inc, 1 mult	69 csteps	$28.9\mu A$	$0.7\mu A$
JPEG columns	2 add, 2 sub, 2 mult	36 csteps	$61.9\mu A$	$4.2\mu A$
JPEG rows	2 add, 2 sub, 1 mult, 1 inc	50 csteps	$32.9\mu A$	$2.2\mu A$
Wavelet3	1 inc, 1 add, 1 mult	47 csteps	$30.2\mu A$	$2.1\mu A$
AES cipher	4 add, 1 mult	116 csteps	$29.9\mu A$	$0.5\mu A$

Table 6.3: Design examples and the efficiency of power-gating in a global sleep state

units of the designs because of the focus within this thesis. Nevertheless, the FUs make up the dominating part of the total energy consumption. For example, in the *FDCT* benchmark the FUs contribute 68% of the total energy consumption whereas the remaining 32% split up for multiplexers, registers, controller, and the clock tree. As the results show, I_{ACTIVE} is effectively reduced to I_{SLEEP} throughout all benchmarks.

In the following, a deeper analysis of the *FDCT* benchmark is examined in order to show the impact of the continuous parameters temperature and supply voltage as well as the discrete parameters process corner and PGS selection. For this analysis the Nangate $45nm$ technology has been chosen in typical and fast process corner. Furthermore, the HVT version has again been selected for sleep devices and the sleep device sizes have been fixed to $2\% \cdot W_{RT}$ for each RTL component. HVT devices require a higher supply voltage. Thus, its range is constrained to $[1.1V; 1.3V]$ whereas the temperature is examined across its whole range of [27°C, 127°C]. Figure 6.12 then shows the gating-switch efficiency as a ratio of I_{SLEEP}/I_{ACTIVE} and the break-even time of the overall FDCT design in nanoseconds.

At first, it can be seen that the efficiency of power-gating has only a small variance across the parameter ranges. It becomes only slightly less effective in suppressing leakage currents if the temperature increases. The supply voltage has also only a marginal impact on the effectivity. Additionally, there is only a small variation between 2% and 4% among the different PGSs. In other words, leakage is reduced by $96 - 98\%$ in all cases and, from the point of pure leakage saving, the PGS selection is not particularly interesting if all surrounding parameters are identical.

Secondly, the break-even time is presented. Unlike the gating effectivity, the t_{be} diminishes with increasing temperature and supply voltage. This is because the wake-up time is much lower and less incomplete transitions occur during the state transition. With a factor of up to four, the variance is also much higher. Furthermore, the PGS selection highly impacts the break-even time. As it can be seen, PMOS schemes have up to two times higher break-even times. Comparing the two process corners, t_{be} is also about twice as big for the typical process corner as that of the fast process corner.

The wake-up time at system-level is given by the maximum RTL component wake-up time if the supply grid is assumed to be sufficiently dimensioned. Figure 6.13 shows the wake-up time of the *FDCT* benchmark in dependence on the temperature and supply voltage parameter for the aforementioned gating types and process corners.

It can be observed that t_{wakeup} shows a very small variance in the parameter ranges. It slightly decreases with increasing supply voltage and increases with a raising temperature. Furthermore, at the fast process

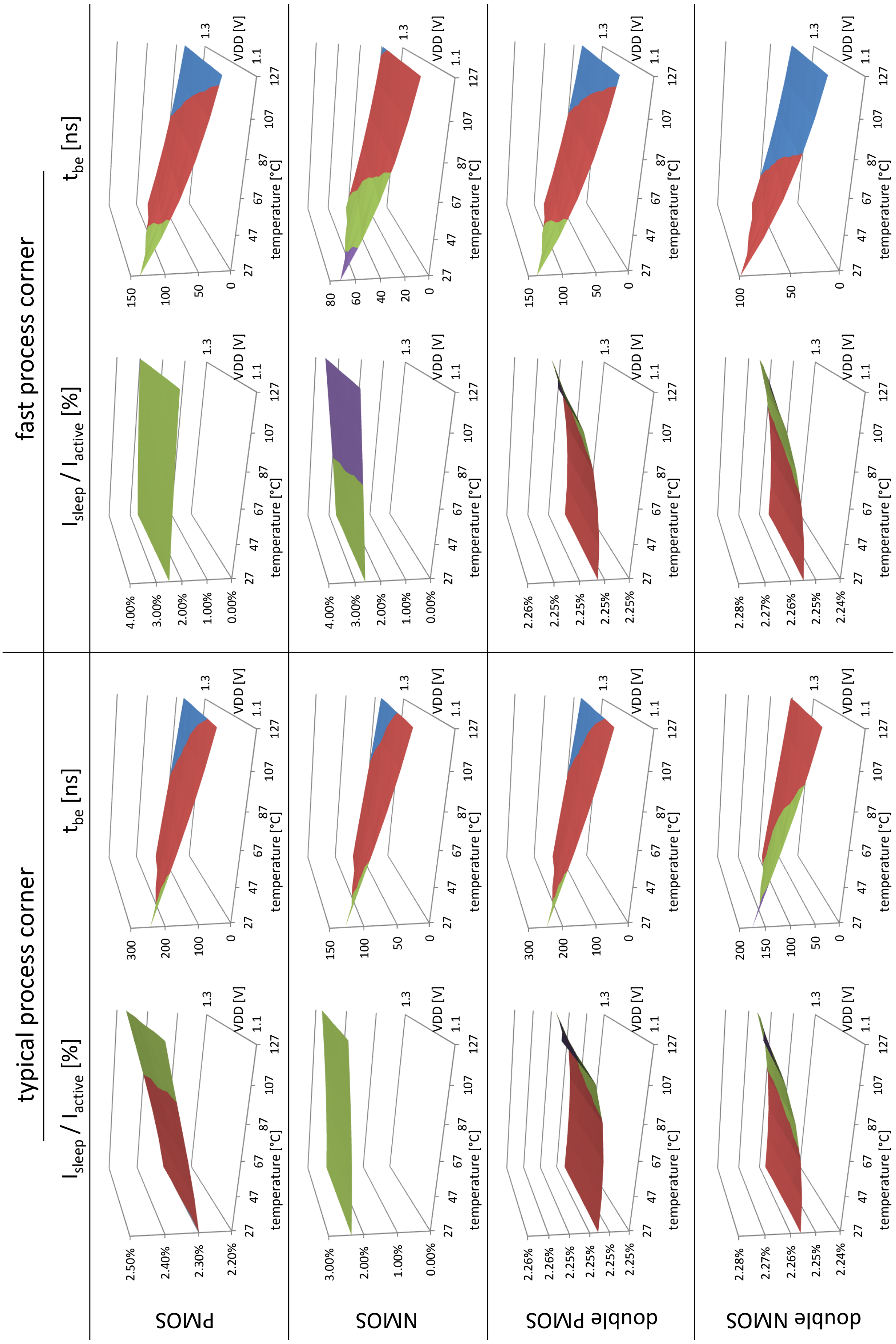


Figure 6.12: Comparison of PGS efficiency and dynamic parameter impact

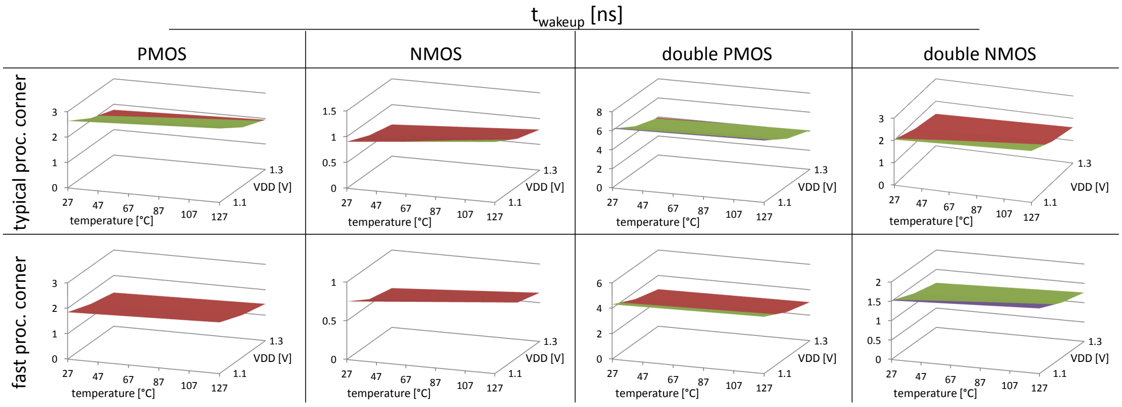


Figure 6.13: Wake-up time evaluation of the *FDCT* design

corner, it is about $20 - 30\%$ smaller as it is at the typical process corner. A comparison of the PMOS and NMOS gating schemes shows that NMOS schemes are about three times faster in waking up.

Next, the impact of sleep transistor sizing on the overall power and area demand is analyzed on the exemplary design. Figure 6.14 lists four different synthesis passes of the *FDCT* behavioral description. The first column holds the results for a synthesis without power-gating and the other three include power-gating during synthesis process. The difference between them is the maximum allowed performance degradation (delay increase) for each arithmetic unit. This constraint is set by the user and is used as an input for the sleep transistor sizing algorithm of Section 5.1. It influences several estimation results including the area, leakage currents during active operation, as well as quiescent leakage currents during the global sleep state.

Smaller delay degradation

	Without power gating	Power gating, 5% performance degradation	Power gating, 1% performance degradation	Power gating, 0.1% performance degradation
Area estimation	42120 µm²	43920 µm² +4.3 %	44608 µm² +5.9 %	46368 µm² +10.1 %
Energy estimation of all arithmetic units	8.15 nWs	8.38 nWs +2.7 %	8.39 nWs +2.9 %	8.42 nWs +3.2 %
Leakage increase of				
Adder		+2.4 %	+2.7 %	+2.8 %
Subtractor		+6.7 %	+7.6 %	+11.6 %
Multiplier		+8.1 %	+9.1 %	+12.5 %
Quiescent current	2.37e-4 A	3.60e-6 A	1.67e-5 A	4.03e-5 A
Leakage reduction in global sleep state		98.5 %	92.9 %	82.9 %

Figure 6.14: Power-gating of FDCT design with IP-level granularity

As it can be seen, the area of the design without power-gating is the smallest and the size increases with falling performance degradation. This is because the smaller the allowed delay increase is, the larger the sleep transistor has to be sized to guarantee a worst-case delay. In total, the area after rough RT-level floorplanning increases by 4.3% to 10.1% for the overall design. The second row holds the total energy estimation result of all arithmetic units within the design. The pure leakage increase for each component is given in the third row. The overhead that is caused by the additional power-management hardware composed of PGS, buffer chain, and voltage anchor rises up to 3.2%. This increase is due to additional leakage currents of the power-gateable adders, subtractors, and multipliers within the design. For example, an increase in leakage current of up to 12.5% can be observed for a multiplier component compared to a non power-gateable multiplier. Beside the overhead costs, power-gating can lead to enormous savings in the global sleep state. In this state the controller waits for an activation and all arithmetic components consume only small remaining quiescent leakage currents. The *FDCT* example shows reductions of the quiescent current of up to 98.5% if a 5% performance degradation is acceptable. In this case, the energy overhead of 2.7% will be amortized by the 98.5% savings if the sleep-time vs. active-time ratio exceeds 3%. If the performance degradation is limited to 0.1%, the overhead increases and the leakage reduction reduces. But even this case is profitable as soon as the entire *FDCT* design is power-gated for at least 4% of the total time.

6.4 Power-Management aware High-Level Synthesis Evaluation

The general adoption of a cycle-accurate power-gating that has been proposed in Section 4.3 as well as the synthesis optimization methodologies presented in Chapter 5 have been implemented into the behavioral-level power optimization framework of PowerOpt®.

Its evaluation starts with an analysis of the benefit of the cycle-wise power-down in Section 6.4.1. Subsequently, the power-management aware binding and allocation approaches are analyzed. Section 6.4.2 gives a comparison of the state-transition cost-aware binding (*STB*) as well as of the data-independent idle-time optimized binding (*ITB*) to the power optimized binding approach (*POB*). *POB* implements the binding of [KSJ+99] with an extension to cover leakage currents. This modification is necessary because otherwise the results would not be comparable as leaving out leakage currents during the allocation would result in a multiple of component instances.

As an isolated consideration of the scheduling is not worthwhile for the application of power-gating, its evaluation is presented afterwards. Section 6.4.3 presents the results for the ILP-based schedulers jointly applied with the proposed binding and allocation approaches. Thereby, a comparison of the cluster-building ILP scheduler (*CBILP*) and of the non-heuristic resource minimizing ILP scheduler (*RMILP*) to the force-directed scheduler (*FDS*) of [PK89] is given. In total, the following techniques are compared and the results are discussed:

- *NoPG+FDS+POB*: Initial implementation in PowerOpt®. Power-gating is not applied at all, the force-directed scheduler and the power optimized binding and allocation is used.

- *PG+FDS+POB*: Power-gating is applied but neither the scheduling nor the binding and allocation are optimized for it.
- *PG+FDS+STB*: The binding cost function includes state-dependent leakage currents and state transition costs.
- *PG+FDS+ITB*: The binding solely optimizes for lengthy consecutive idle times.
- *PG+RMILP+STB*: The non-heuristic resource-minimizing scheduler is used. The binding optimizes for the overall energy consumption. Regarding the scheduler, this synthesis approach is a non-heuristic implementation of *PG+FDS+STB*.
- *PG+CBILP+STB*: The scheduler is replaced by the cluster-building one. The binding optimizes for the overall energy consumption and is state-aware.
- *PG+CBILP+ITB*: The same scheduler as before is used but the binding optimizes for continuous idle times.

The evaluation focuses on the Nangate semiconductor technology at the fast process corner because it has been evaluated to have the largest leakage currents and it is the most appropriate technology because of its small break-even times. As analyzed and discussed in Section 2.2.2 the industrial low power specialized $45nm$ and $65nm$ technologies have significantly higher break-even times and are not suitable for a temporal fine-grained power down. All synthesis approaches have been applied to the benchmarks of Table 6.3 and the evaluation results are presented in the following.

6.4.1 Evaluation of Cycle-Accurate RTL Estimation

Figure 6.15 presents the evaluation results of the cycle-wise power down of RTL components within the benchmarks' datapaths and during runtime. It relatively compares the energy consumption of the designs under a given testbench execution for the *NoPG+FDS+POB* and *PG+FDS+POB* synthesis approaches. All synthesis passes use the single NMOS-cutoff technique at a supply voltage of $1.0V$ and are constrained to a maximum delay increase for each component of 8%.

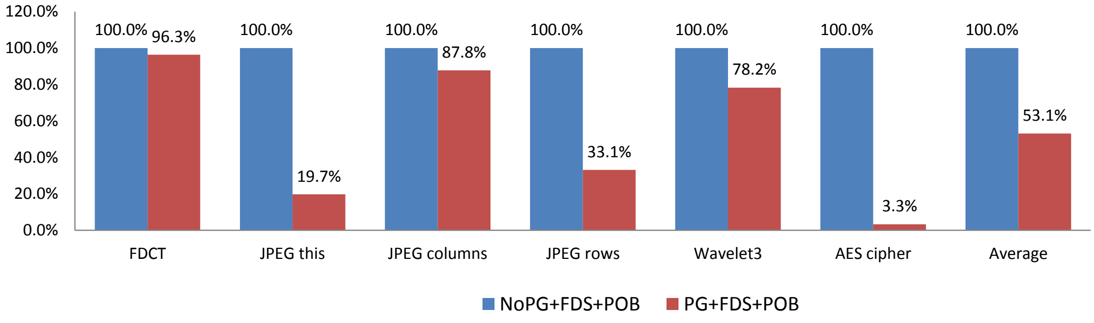


Figure 6.15: Benefit of a cycle-accurate power-gating application in terms of the functional unit energy demand reduction

It can be seen that power-gating can drastically reduce the energy consumption but its degree of reduction varies a lot between the benchmarks. The *FDCT* benchmark, for example, is almost ineligible for a cycle-wise application of power-gating and only 3.7% savings can be obtained. This is due to the small schedule length and high workload of the components. In contrast, the *AES cipher* benchmark has a very long and control-flow intensive schedule that contains only a small number of operations. This circumstances result in a reduction of up to 96.7% that is close to the theoretical minimum defined by the quiescent current during sleep as analyzed in Table 6.3. In average, the energy demand reduces by about 47%.

6.4.2 Binding and Allocation Evaluation

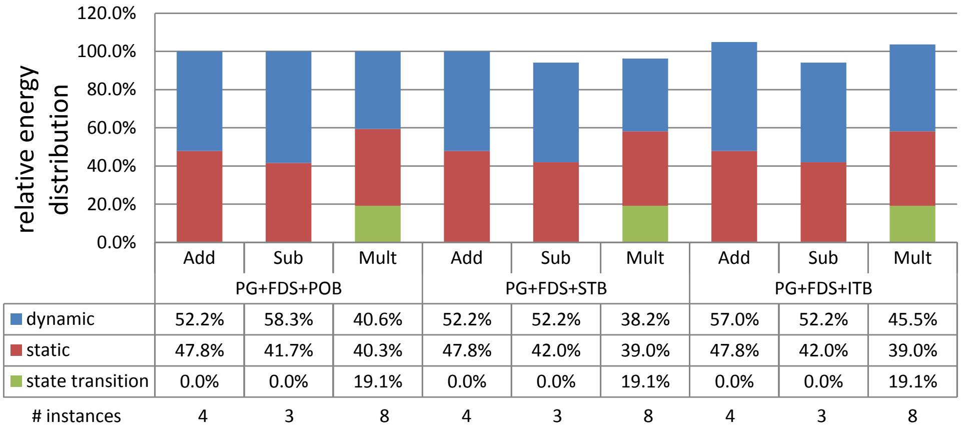
Figure 6.16 presents relative energy distributions within the *FDCT* benchmark. For each component type the total energy consumption is divided into a dynamic, static, and state transition part. *Dynamic* refers to the fraction of energy that has been consumed due to switching activity. *Static energy* occurred due to leakage currents and *state transition energy* refers to the switching costs for entering and leaving the sleep state. The first three bars belong to the FU estimates of the *PG+FDS+POB* synthesis approach. In this case, power-gating is applied but the synthesis does neither take care of state transition costs nor optimizes for continuous idle times. All other synthesis passes are relatively scaled to these estimates to clearly indicate the benefit of synthesis improvements.

In the *FDCT* design, the adder and subtractor components are subjected to a high workload and a cycle-wise application of power-gating is not worthwhile for these components at all. For this reason, these components are never power-gated during runtime and no state transition energy occurs. Their overall energy demand even increases compared to the synthesis without power-gating because of the overhead costs in terms of the sleep transistors, buffer chains, and voltage anchors. The multiplier components also have a workload but its energy consumption can slightly be reduced by 4.1% in the *PG+FDS+POB* synthesis compared to *NoPG+FDS+POB* run. Using the *STB* binding, its energy consumption can be reduced by additional 3.7%.

In all cases, the estimates refer to the best allocation for each FU that has been found during synthesis. While the allocation is constant in the *FDCT* benchmark, it will vary in some of the following designs.

Figure 6.17 presents the evaluation results for three JPEG processes. They all belong to a JPEG encoder and decoder suite and represent disjoint algorithms. In contrast to the *FDCT* benchmark, the *JPEG* algorithms are memory intensive and contain a lot of memory accesses on the critical path. Thus, the FU's workloads are much lower and power-gating is more effective. In the *JPEG this* benchmark for example, the multiplier and incrementer components have a workload of 0.015 and 0.058. As a result, their energy demand can be reduced by 84.2% and 62.7% compared to the *NoPG+FDS+POB* synthesis. A similar observation could be done for the *JPEG columns* and *JPEG rows* benchmarks.

The *PG+FDS+STB* and *PG+FDS+ITB* binding approaches can further reduce the overall energy demand in the *JPEG this* benchmark by about 1% in contrast to the *PG+FDS+POB* approach. This reduction is small because in most of the cases only one component instance is used in the allocation and the binding does not have any remaining degrees of freedom for these cases. In contrast, the multiplier and adder energy in the *JPEG columns* benchmark can be reduced from 100% in the *PG+FDS+POB*



	PG+FDS+POB			PG+FDS+STB			PG+FDS+ITB		
	Add	Sub	Mult	Add	Sub	Mult	Add	Sub	Mult
🟦 dynamic	52.2%	58.3%	40.6%	52.2%	52.2%	38.2%	57.0%	52.2%	45.5%
🟥 static	47.8%	41.7%	40.3%	47.8%	42.0%	39.0%	47.8%	42.0%	39.0%
🟩 state transition	0.0%	0.0%	19.1%	0.0%	0.0%	19.1%	0.0%	0.0%	19.1%
# instances	4	3	8	4	3	8	4	3	8

Figure 6.16: Relative energy distributions of the *FDCT* benchmark for different binding and allocation approaches

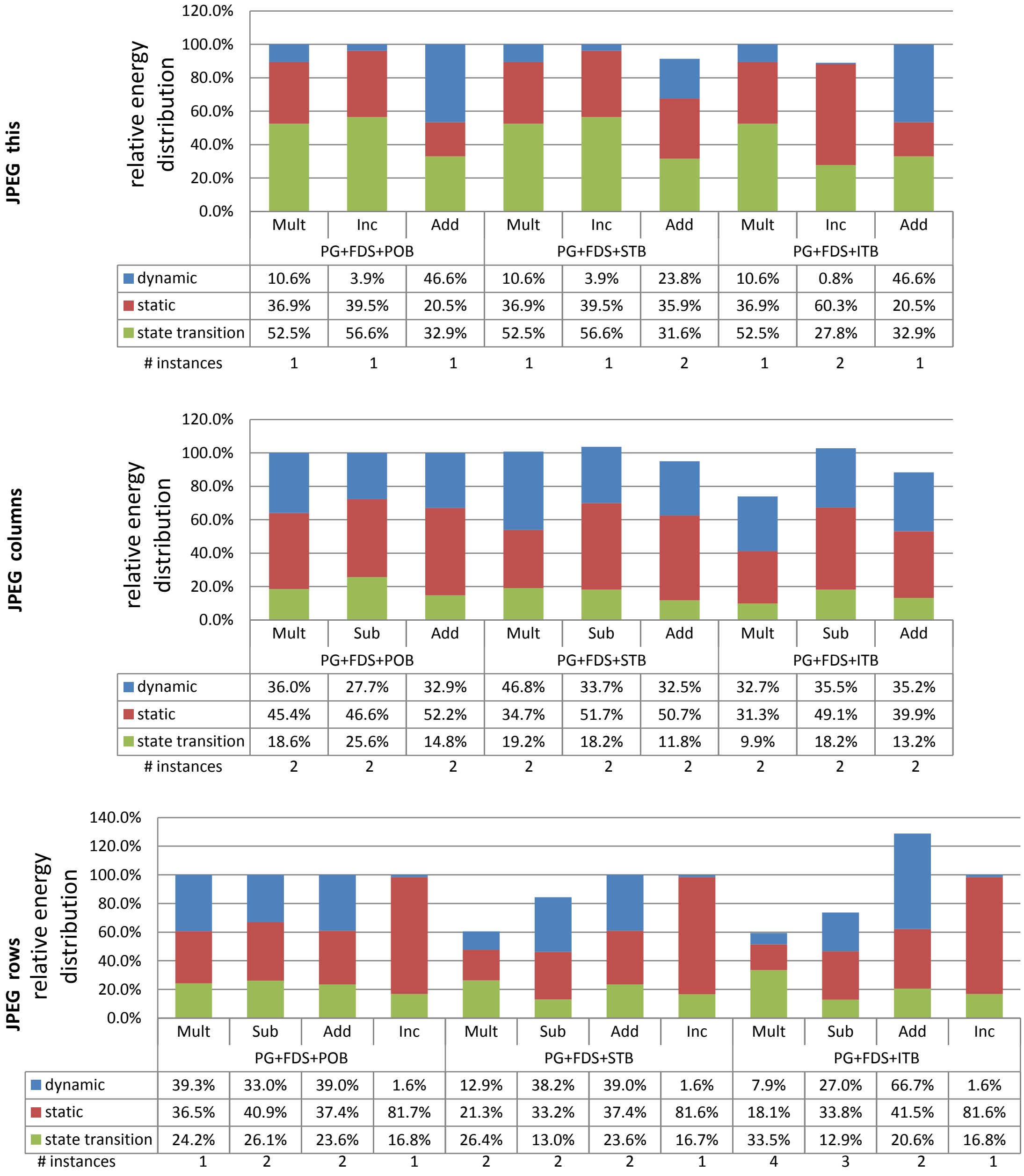
synthesis to 73.9% and 88.3% in the *PG+FDS+ITB* approach.

In the *JPEG rows* design, the *PG+FDS+STB* synthesis reduces the multiplier and subtractor energy by 39% and 15%. These savings have been obtained due to pure synthesis optimizations by considering the power-gating costs during binding. The *PG+FDS+ITB* synthesis can even further reduce the multiplier and subtractor costs. In comparison to the *PG+FDS+POB* synthesis their overall energy demand reduces by 40.5% and 26.3%. A drawback of the simple idle-time based binding arises with the negligence of switching activity as it can clearly be seen using the example of the adder components. Their dynamic energy demand rises by 70% and exceeds smaller savings of the state transition costs.

Figure 6.18 trades off different allocations for the multiplier components within the *JPEG rows* benchmark. The algorithmic description contains five multiplications whereas each of them is scheduled in another controller step. Thus, one to five multipliers can be used during synthesis. In the *NoPG+FDS+POB* synthesis, power-gating is not enabled and the leakage currents increase with the number of multiplier instances. While a single multiplier needs to be instantiated in the largest necessary bitwidth, the second to fifth instance can be smaller. For this reason, the aggregated leakage energy does not increase fivefold. Obviously, the best allocation uses one multiplier component. In the *PG+FDS+ITB* synthesis approach, multiplier instances are power-gated during idle periods and costs of additional instances are limited. As it can be seen, the state transition costs increase with the number of multiplier instances but these costs are compensated by leakage savings and the best allocation has been found using four multipliers.

The cycle-wise power-gating, being applied to the *Wavelet3* benchmark, reduces the energy demand of the multiplier, incrementer, and adder components by 21.8% in average as shown in Figure 6.15. The *STB* and *ITB* binding approach do not lead to further improvements due to the small amount of component instances.

Consisting of 116 controller steps and multiple nested loops, resulting in hundreds of cycles for one design invocation, the *AES cipher* benchmark is the largest benchmark regarded in the evaluation. Its multiplier component is required in only one controller step and thus it idles hundreds of consecutive



JPEG this

	Mult	Inc	Add	Mult	Inc	Add	Mult	Inc	Add
	PG+FDS+POB			PG+FDS+STB			PG+FDS+ITB		
🟦 dynamic	10.6%	3.9%	46.6%	10.6%	3.9%	23.8%	10.6%	0.8%	46.6%
🟥 static	36.9%	39.5%	20.5%	36.9%	39.5%	35.9%	36.9%	60.3%	20.5%
🟩 state transition	52.5%	56.6%	32.9%	52.5%	56.6%	31.6%	52.5%	27.8%	32.9%
# instances	1	1	1	1	1	2	1	2	1

JPEG columns

	Mult	Sub	Add	Mult	Sub	Add	Mult	Sub	Add
	PG+FDS+POB			PG+FDS+STB			PG+FDS+ITB		
🟦 dynamic	36.0%	27.7%	32.9%	46.8%	33.7%	32.5%	32.7%	35.5%	35.2%
🟥 static	45.4%	46.6%	52.2%	34.7%	51.7%	50.7%	31.3%	49.1%	39.9%
🟩 state transition	18.6%	25.6%	14.8%	19.2%	18.2%	11.8%	9.9%	18.2%	13.2%
# instances	2	2	2	2	2	2	2	2	2

JPEG rows

	Mult	Sub	Add	Inc	Mult	Sub	Add	Inc	Mult	Sub	Add	Inc
	PG+FDS+POB				PG+FDS+STB				PG+FDS+ITB			
🟦 dynamic	39.3%	33.0%	39.0%	1.6%	12.9%	38.2%	39.0%	1.6%	7.9%	27.0%	66.7%	1.6%
🟥 static	36.5%	40.9%	37.4%	81.7%	21.3%	33.2%	37.4%	81.6%	18.1%	33.8%	41.5%	81.6%
🟩 state transition	24.2%	26.1%	23.6%	16.8%	26.4%	13.0%	23.6%	16.7%	33.5%	12.9%	20.6%	16.8%
# instances	1	2	2	1	2	2	2	1	4	3	2	1

Figure 6.17: Relative energy distributions of the *JPEG* benchmarks for different binding and allocation approaches

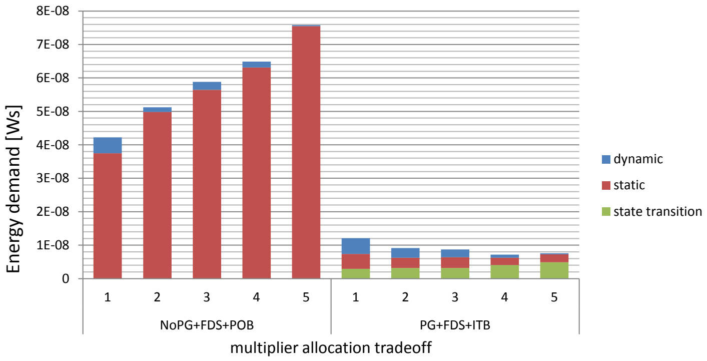
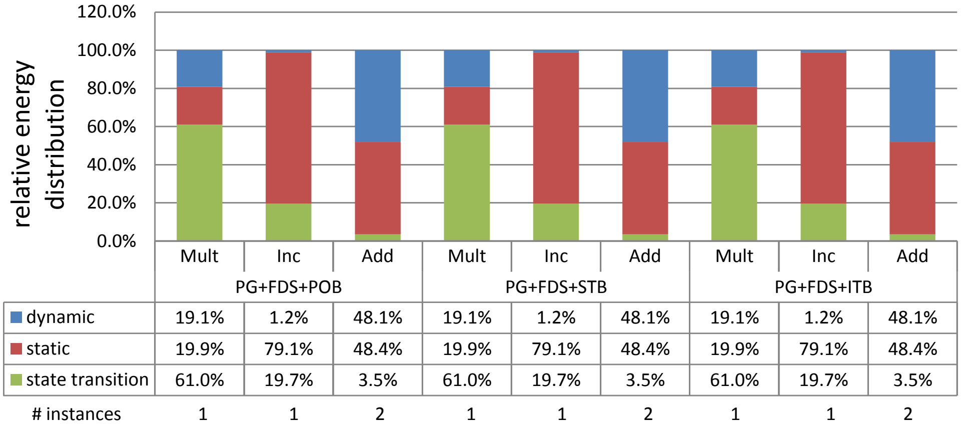


Figure 6.18: Allocation tradeoff for the multiplier components within the *JPEG rows* benchmark

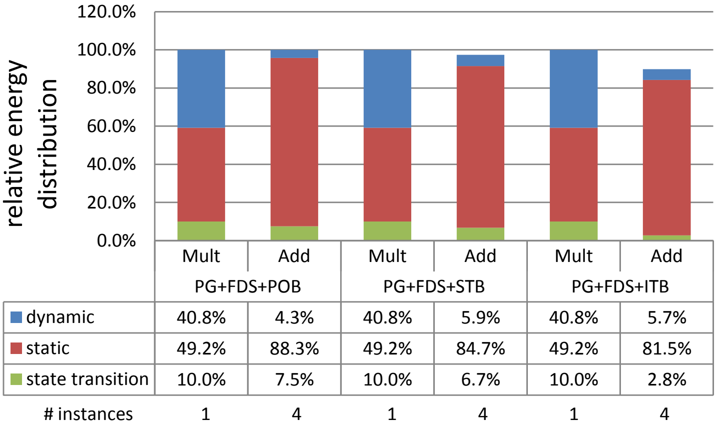
cycles during execution. Thus, power-gating can reduce its energy consumption by 96.7% as stated in Figure 6.15. Regarding the adder components, the reduction can even be increased by the *STB* and *ITB* binding.



	Mult	Inc	Add	Mult	Inc	Add	Mult	Inc	Add
	PG+FDS+POB			PG+FDS+STB			PG+FDS+ITB		
🟦 dynamic	19.1%	1.2%	48.1%	19.1%	1.2%	48.1%	19.1%	1.2%	48.1%
🟥 static	19.9%	79.1%	48.4%	19.9%	79.1%	48.4%	19.9%	79.1%	48.4%
🟩 state transition	61.0%	19.7%	3.5%	61.0%	19.7%	3.5%	61.0%	19.7%	3.5%
# instances	1	1	2	1	1	2	1	1	2

Figure 6.19: Relative energy distributions of the *Wavelet3* benchmarks for different binding and allocation approaches

To summarize the results, the highest energy reductions have been achieved for components with low workloads under the consideration of the idle-time optimizing synthesis approach. Furthermore, the *PG+FDS+ITB* approach results in a significant speedup of the synthesis because of the small amount of necessary model estimates. In average, the *PG+FDS+ITB* synthesis approach has been evaluated to be $15 - 20\%$ faster than the *POB*-based synthesis and even speeds up the synthesis by a factor of 2 compared to the *PG+FDS+STB* synthesis. Regarding the energy minimization the *PG+FDS+STB* synthesis remains slightly behind the *PG+FDS+ITB* results but it regards the overall energy estimates and prevents the dynamic energy from dominating the costs as it occurred for the adder components in the *JPEG rows* benchmark.



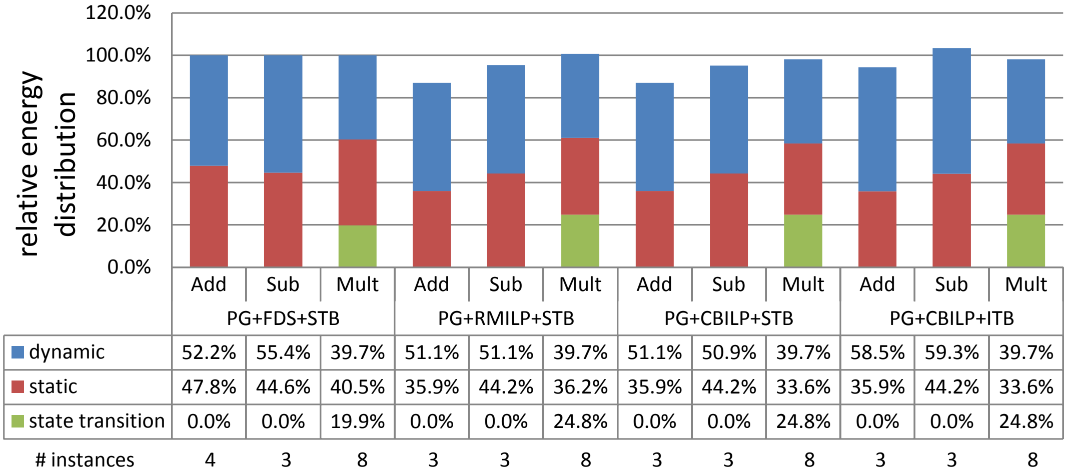
	Mult	Add	Mult	Add	Mult	Add
	PG+FDS+POB		PG+FDS+STB		PG+FDS+ITB	
🟦 dynamic	40.8%	4.3%	40.8%	5.9%	40.8%	5.7%
🟥 static	49.2%	88.3%	49.2%	84.7%	49.2%	81.5%
🟩 state transition	10.0%	7.5%	10.0%	6.7%	10.0%	2.8%
# instances	1	4	1	4	1	4

Figure 6.20: Relative energy distributions of the *AES cipher* benchmarks for different binding and allocation approaches

6.4.3 Scheduler Evaluation

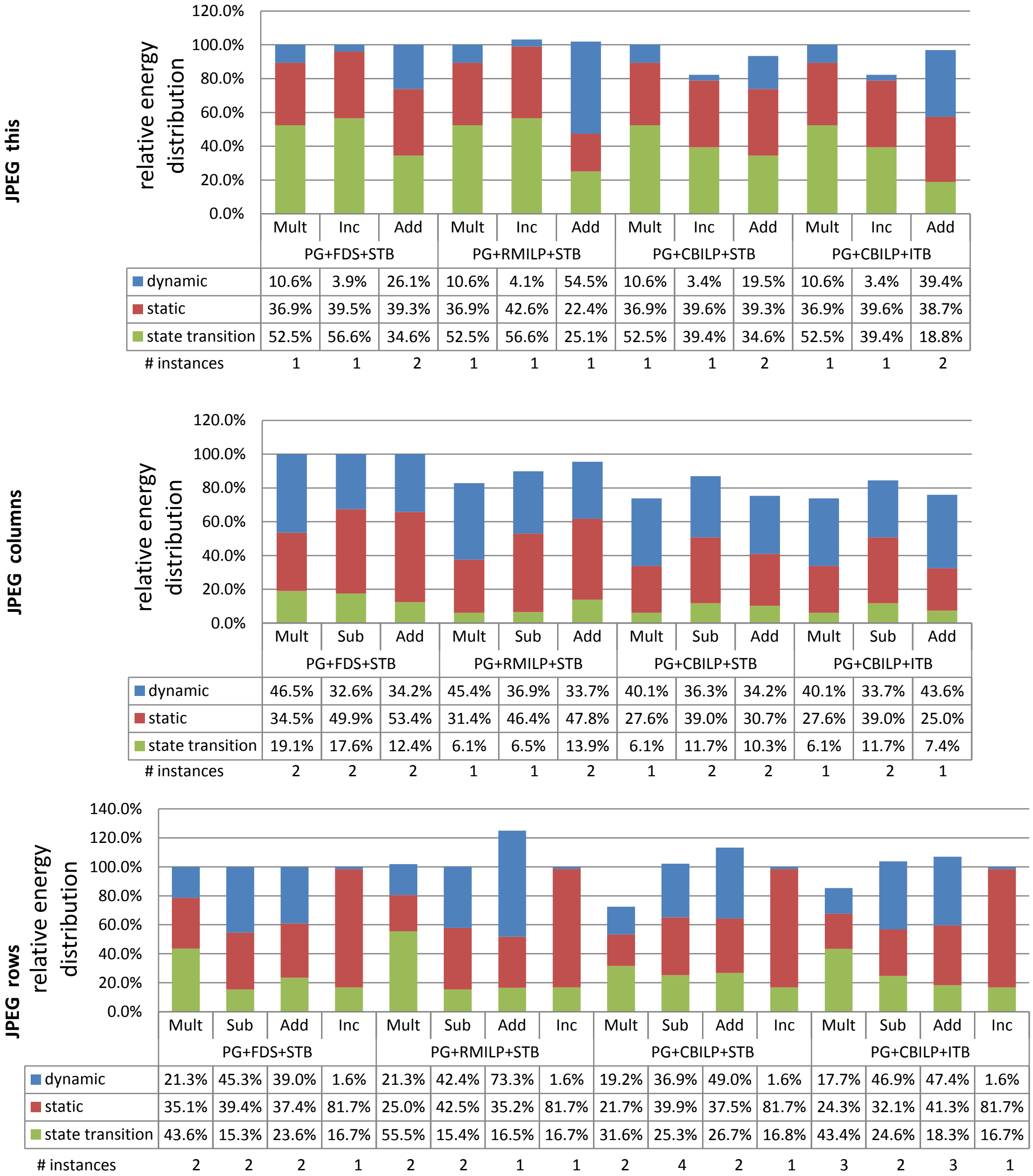
The integer linear programs of Section 5.2 have been formulated as templates using the Zimpl language [Kon11] and Solving Constraint Integer Programs (SCIP) [Ach04] was used for solving the ILPs. These templates were automatically filled with design specific data by PowerOpt®.

As it has been expected, the short schedule of the *FDCT* benchmark has only a small degree of freedom for an optimization. In contrast to the FDS approach the non-heuristic resource-minimizing scheduling *RMILP* was able to reduce the number of adder components by one. Beside the associated leakage reduction, energy differences mainly result out of variations in the dynamic power and may slightly reduce or increase the total energy consumption.



	Add	Sub	Mult	Add	Sub	Mult	Add	Sub	Mult	Add	Sub	Mult
	PG+FDS+STB			PG+RMILP+STB			PG+CBILP+STB			PG+CBILP+ITB		
🟦 dynamic	52.2%	55.4%	39.7%	51.1%	51.1%	39.7%	51.1%	50.9%	39.7%	58.5%	59.3%	39.7%
🟥 static	47.8%	44.6%	40.5%	35.9%	44.2%	36.2%	35.9%	44.2%	33.6%	35.9%	44.2%	33.6%
🟩 state transition	0.0%	0.0%	19.9%	0.0%	0.0%	24.8%	0.0%	0.0%	24.8%	0.0%	0.0%	24.8%
# instances	4	3	8	3	3	8	3	3	8	3	3	8

Figure 6.21: Relative energy distributions of the *FDCT* benchmark for different scheduling approaches



JPEG this

	PG+FDS+STB			PG+RMILP+STB			PG+CBILP+STB			PG+CBILP+ITB		
	Mult	Inc	Add	Mult	Inc	Add	Mult	Inc	Add	Mult	Inc	Add
🟦 dynamic	10.6%	3.9%	26.1%	10.6%	4.1%	54.5%	10.6%	3.4%	19.5%	10.6%	3.4%	39.4%
🟥 static	36.9%	39.5%	39.3%	36.9%	42.6%	22.4%	36.9%	39.6%	39.3%	36.9%	39.6%	38.7%
🟩 state transition	52.5%	56.6%	34.6%	52.5%	56.6%	25.1%	52.5%	39.4%	34.6%	52.5%	39.4%	18.8%
# instances	1	1	2	1	1	1	1	1	2	1	1	2

JPEG columns

	PG+FDS+STB			PG+RMILP+STB			PG+CBILP+STB			PG+CBILP+ITB		
	Mult	Sub	Add	Mult	Sub	Add	Mult	Sub	Add	Mult	Sub	Add
🟦 dynamic	46.5%	32.6%	34.2%	45.4%	36.9%	33.7%	40.1%	36.3%	34.2%	40.1%	33.7%	43.6%
🟥 static	34.5%	49.9%	53.4%	31.4%	46.4%	47.8%	27.6%	39.0%	30.7%	27.6%	39.0%	25.0%
🟩 state transition	19.1%	17.6%	12.4%	6.1%	6.5%	13.9%	6.1%	11.7%	10.3%	6.1%	11.7%	7.4%
# instances	2	2	2	1	1	2	1	2	2	1	2	1

JPEG rows

	PG+FDS+STB				PG+RMILP+STB				PG+CBILP+STB				PG+CBILP+ITB			
	Mult	Sub	Add	Inc	Mult	Sub	Add	Inc	Mult	Sub	Add	Inc	Mult	Sub	Add	Inc
🟦 dynamic	21.3%	45.3%	39.0%	1.6%	21.3%	42.4%	73.3%	1.6%	19.2%	36.9%	49.0%	1.6%	17.7%	46.9%	47.4%	1.6%
🟥 static	35.1%	39.4%	37.4%	81.7%	25.0%	42.5%	35.2%	81.7%	21.7%	39.9%	37.5%	81.7%	24.3%	32.1%	41.3%	81.7%
🟩 state transition	43.6%	15.3%	23.6%	16.7%	55.5%	15.4%	16.5%	16.7%	31.6%	25.3%	26.7%	16.8%	43.4%	24.6%	18.3%	16.7%
# instances	2	2	2	1	2	2	1	1	2	4	2	1	3	2	3	1

Figure 6.22: Relative energy distributions of the *JPEG* benchmarks for different scheduling approaches

The *JPEG* benchmarks results are presented in Figure 6.22. Again, the resource-minimizing scheduling reduces the number of components in all designs but the overall energy consumption is only reduced in the *JPEG columns* benchmark. It can impressively be seen that the allocation varies a lot throughout the synthesis approaches and in average more component instances are used in the *CBILP* approaches. Although more components are instantiated, the cluster-building scheduler *CBILP* can reduce the energy consumption by up to 27% for the multiplier components in the *PG+CBILP+STB* synthesis approach. As the multipliers contribute the largest part of energy, the scheduling forces the multiplication operation clustering due to the weighting in the cost function as defined in Equation 5.9 in Section 5.2.3.

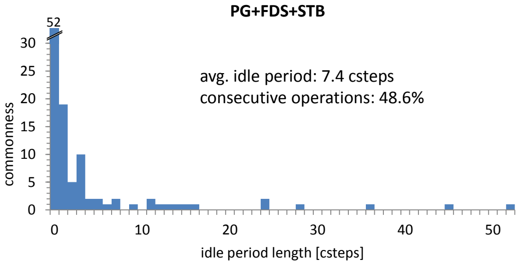


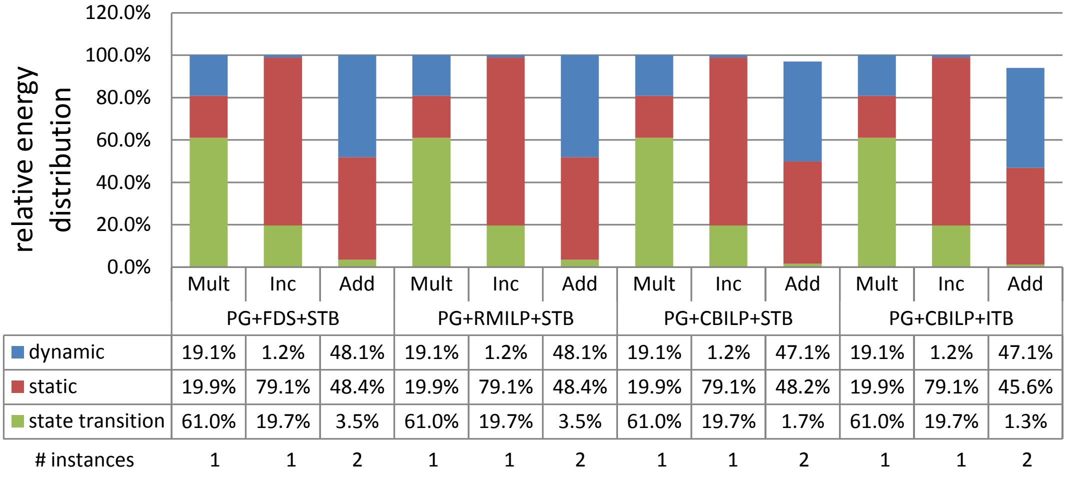
Figure 6.23: Histogram of idle period lengths within all *JPEG* benchmarks

In the following the operation clustering is analyzed. Figure 6.23 shows histograms of the idle period lengths in between each pair of successive operations. Idle periods with a length of zero csteps are optimal as they represent two consecutively executed operations. The force-directed scheduler results in 48.6% of consecutive operations. The majority of the remaining idle period lengths is situated in the single-digit range as it can be seen in the *PG+FDS+STB* histogram. These periods are mostly below the break-even time and cannot be exploited for a power down. The average idle period length is 7.4 csteps. The cluster-building ILP scheduler in the *PG+CBILP+STB* synthesis results in a much higher rate of 78% consecutive operations. The average idle period length rises to 16 controller steps.

The *Wavelet3* benchmark does not show significant energy reductions because the $[ASAP, ALAP]$ intervals of the operations are tightly constrained. Only the adder energy can be reduced by 3% and 6% using the *PG+CBILP+STB* and *PG+CBILP+ITB* approaches.

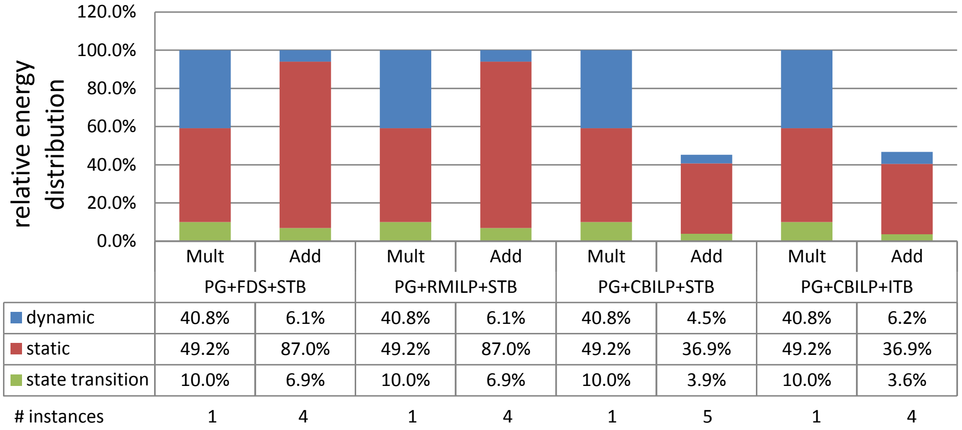
Similar to the *Wavelet3* design, the operations within the *AES cipher* benchmark are also tightly constrained by the control- and dataflow because they are placed in nested loops. But in this benchmark, the loop-unrolling technique of PowerOpt® can relax the constraints leading to considerable results. The adder energy reduces by 54% due to the clustered scheduling. Of course the loop unrolling has been applied in all synthesis passes to obtain comparability.

Table 6.4 gives a runtime comparison for the overall synthesis time for all benchmarks and synthesis approaches. The cycle-wise considerations of the power-gating technique as well as the *STB* binding approach have the most impact on the synthesis runtime. The cluster-building scheduler only slightly slows down the synthesis. It needs to be mentioned that the prototype integration of the models has not been optimized for access and response time and a drastic reduction should be possible.



	PG+FDS+STB			PG+RMILP+STB			PG+CBILP+STB			PG+CBILP+ITB		
	Mult	Inc	Add	Mult	Inc	Add	Mult	Inc	Add	Mult	Inc	Add
🟦 dynamic	19.1%	1.2%	48.1%	19.1%	1.2%	48.1%	19.1%	1.2%	47.1%	19.1%	1.2%	47.1%
🟥 static	19.9%	79.1%	48.4%	19.9%	79.1%	48.4%	19.9%	79.1%	48.2%	19.9%	79.1%	45.6%
🟩 state transition	61.0%	19.7%	3.5%	61.0%	19.7%	3.5%	61.0%	19.7%	1.7%	61.0%	19.7%	1.3%
# instances	1	1	2	1	1	2	1	1	2	1	1	2

Figure 6.24: Relative energy distributions of the *Wavelet3* benchmark for different scheduling approaches



	PG+FDS+STB		PG+RMILP+STB		PG+CBILP+STB		PG+CBILP+ITB	
	Mult	Add	Mult	Add	Mult	Add	Mult	Add
🟦 dynamic	40.8%	6.1%	40.8%	6.1%	40.8%	4.5%	40.8%	6.2%
🟥 static	49.2%	87.0%	49.2%	87.0%	49.2%	36.9%	49.2%	36.9%
🟩 state transition	10.0%	6.9%	10.0%	6.9%	10.0%	3.9%	10.0%	3.6%
# instances	1	4	1	4	1	5	1	4

Figure 6.25: Relative energy distributions of the *AES cipher* benchmark for different scheduling approaches

	Synthesis and estimation runtime [s]					
	NoPG+	*PG+*				
	FDS+POB	*FDS+POB*	*FDS+STB*	*FDS+ITB*	*CBILP+STB*	*CBILP+ITB*
FDCT	8.20s	10.40s	22.66s	8.86s	23.49s	10.60s
JPEG this	19.89s	29.30s	49.95s	26.55s	51.81s	27.23s
JPEG columns	1.78s	3.57s	5.88s	2.76s	6.05s	2.94s
JPEG rows	7.36s	12.70s	29.89s	10.25s	30.39s	10.44s
Wavelet3	30.52s	46.44s	57.73s	41.60s	62.49s	43.31s
AES cipher	78.10s	128.31s	149.32s	118.01s	155.92s	124.81s
avg. change		+60.6%	+174.0%	+37.2%	+184.0%	+45.8%

Table 6.4: Runtime comparison of synthesis approaches

6.4.4 Summary

A cycle-accurate power down of RTL components has been evaluated in Section 6.4.1 to result in significant reductions of the FUs energy consumption. For the proposed benchmarks, the overall energy demand could be reduced with the power-gating technique by 46.8% in average using the unoptimized *PG+FDS+POB* synthesis approach.

Synthesis optimizations in terms of the new *STB* and *ITB* binding approaches as well as the proposed *CBILP* scheduler can lead to further reductions as analyzed in Section 6.4.2 and Section 6.4.3. In these, Figures 6.16 to 6.25 contain relative changes in the energy consumption to describe the impact of the cycle-wise power-gating technique and the binding/allocation and scheduling approaches for separated functional units. Figure 6.26 presents aggregated estimates for all benchmarks and in average.

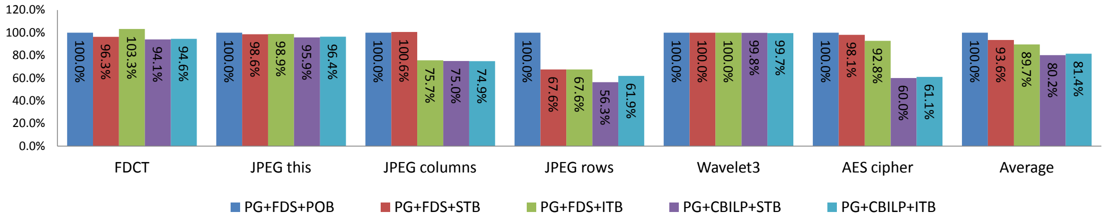


Figure 6.26: Aggregated energy estimate reductions for each benchmark and in average

It has been shown that the combination of the *CBILP* scheduling and the *STB* binding results in an average overall energy reduction of 19.8%. Due to the data independency, the *ITB*-based synthesis is slightly behind and results to an average reduction of 18.6%. Nevertheless, the *ITB* based approaches are much faster as the amount of model invocations is reduced drastically.

In general, the evaluation showed that the state transition energy makes up a huge fraction of the overall energy demand in temporal fine-grained power-gated designs. As the E_{SW}^{RT} model assumes all capacitances within a component to be completely discharged instantly after power down, the overall savings due to power-gating will even be much higher than the pessimistic estimates presented in this evaluation.

6.5 Compliance with Industrial Power Standards

During high-level synthesis, a lot of decisions about the power-gating integration are done. This includes decisions for which components it is profitable to include a power switch and to add a sleep signal to the controller. For each power-gateable component also information about when to put it into a sleep state and wake it up again are defined. These decisions have to be passed to subsequent tools for further synthesis of the RTL-design to even lower levels of abstraction. The timing information for each sleep signal is thereby directly contained in the controller, is thus specified in the RTL synthesis output in Verilog or VHDL, and, in this thesis, is outputted by the PowerOpt® internal RTL-writer. In contrast to this, the decisions regarding the power switch, available power domains, and their interconnect cannot be formulated in hardware description languages. For this purpose, the *Unified-Power Format* and the *Common-Power Format* have been introduced by the industry to specify this information in a metafile. In this thesis a CPF-writer has been developed and integrated into OFFIS's PowerOpt®. The integration has been done in accordance to the official CPF 1.0 specification [Sil08]. This section aims to describe the power-format metafile and to substantiate the consistency of the developed power-gating methodology and tool integration.

In the following, the design of Figure 4.17 will again be used to examine the power-management tool-integration and the resulting CPF file. This small example design has been chosen because it consists of only two functional units and its controller, its datapath, and its CPF-file is manageable for a full presentation in this section. Figure 6.27 shows a PowerOpt® screenshot of the scheduled CDFG consisting of operations, each bound to an adder or a subtractor, and the associated data dependencies. During synthesis, PMOS-gating has been selected and the break-even time has been evaluated to be within one clock cycle for simplicity. Thus, power-gating is worthwhile for both functional units and they are both equipped with a separate PGS.

Figure 6.28 visualizes the corresponding microarchitectural datapath that has been created by PowerOpt® and also been exported by the RTL-writer. It consists of input ports X, output ports Y, registers, the controller, and the two arithmetic units. The multiplexer select and register enable signals are visualized as dotted lines. The arithmetic units obtained sleep inputs in addition to the data inputs. These are connected to and controlled by the controller and are highlighted in red.

As a part of the HLS the controller managing the datapath is automatically synthesized. Listing 6.1 shows a list of the controller inputs and outputs. As power-management was considered during synthesis, it contains sleep outputs $SLEEP0$ and $SLEEP1$ to control the corresponding adder and subtractor.

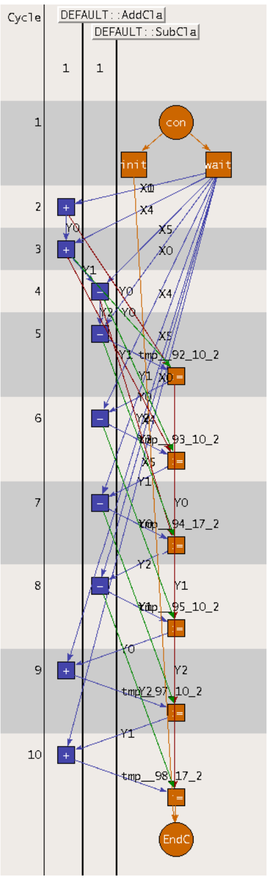


Figure 6.27: Control- and data-flow graph of an example design

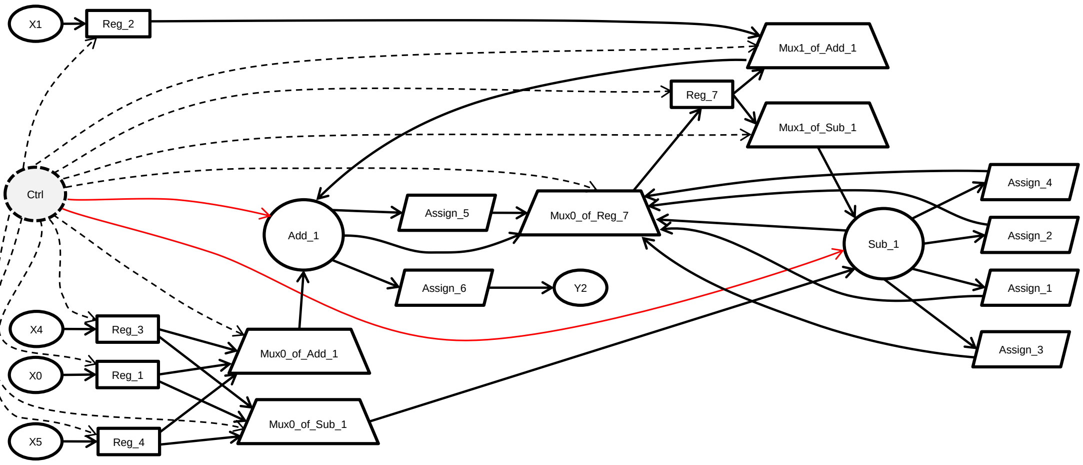


Figure 6.28: RTL datapath of example design including the controller sleep signals

```
Item Output Bits Details
 1      RDY    1  Ready signal
 2      BRK    1  Break signal
 3      RET    1  Return signal
 4   SLEEP0    1  Add_1
 5     MUX1    2  Mux0_of_Add_1
 6     MUX2    1  Mux1_of_Add_1
 7      RE3    1  Reg_1
 8      RE4    1  Reg_2
 9      RE5    1  Reg_3
10      RE6    1  Reg_4
11      RE7    1  Reg_7
12     MUX8    3  Mux0_of_Reg_7
13   SLEEP9    1  Sub_1
14    MUX10    2  Mux0_of_Sub_1
15    MUX11    1  Mux1_of_Sub_1
```

Listing 6.1: Controller inputs and outputs

The temporal behavior of the controller is presented in a state transition table in Listing 6.2 defining the output values for each output signal. A comparison of the scheduled operations in the CDFG of Figure 6.27 and the sleep signal values in Listing 6.1 shows the correlation between them. In the global idle state, the adder and the subtractor are put to the sleep state because the sleep signal is high using a PMOS-based PGS. Dependent on the wake-up time of one cycle, the adder is switched on while entering controller-step 1 and it remains on till cstep 3. While entering cstep 4, it is switched off again because the following idle time exceeds the break-even time. In cstep 8 it is switched on again for two consecutive additions in cstep 9 and 10.

```
[Finite state machine] ************************
Item CStep Input State Next     Output
 1     0    _0_   IDLE IDLE 0001___________1___
 2     0    _1_   IDLE   S1 0000___11110___1___
 3     1    ___     S1   S2 000000000000___1___
 4     2    ___     S2   S3 0000011000000010___
 5     3    ___     S3   S4 0001___000000010000
 6     4    ___     S4   S5 0001___000000000010
 7     5    ___     S5   S6 0001___000000010010
 8     6    ___     S6   S7 0001___000000010000
 9     7    ___     S7   S8 0000___000000000010
10     8    ___     S8   S9 0000011000010011___
11     9    ___     S9  S10 0000011000000011___
12    10    ___    S10 IDLE 1001___________1___
```

Listing 6.2: Next state table of synthesized controller and power manager of the example design in Figure 6.27

Listing 6.3 now shows the Common Power Format file for the example design. At first, the *set_design* command specifies the name of the hardware description language (HDL) module, to which the power information in the file applies. In PowerOpt® the process name is taken because each process is synthesized separately to a Verilog module and results in a dedicated CPF file. The command *create_ground_nets* then defines a list of ground nets. In the current example only one ground net GND with a ground potential of $0.0V$ exists. Analogous to the ground net, the command *create_power_nets* specifies a list of power nets. Again, the current example has only one power net V_{DD} with a supply potential of $1.0V$.

Afterwards, the logic structure for all power domains is specified. At first, power domains are created and named with the *create_power_domain* command. Further, all component instances, boundary ports, and pins that belong to this power domain are specified. The example design consists of three domains. $PD0$ is the default power domain and contains all RTL components that are not power-manageable and thus are on all the time. This includes the controller, registers and multiplexers. For every manageable RTL component an individual power domain is created and the components are defined as an instance within the domain. Furthermore, a shutoff condition is specified, connecting a power domain to a sleep signal output pin of the controller. This directly spans a connection to the functional Verilog module. Finally, the *update_power_domain* command specifies an internal power or ground net for each power-gateable power domain using a unique net name. Since PMOS-based power-gating is applied in the design, local virtual supply nets are specified.

The *create_nominal_condition* command specifies operating conditions with a supply voltage. In the example design only one nominal condition exists but if DVFS techniques would have been applied, additional conditions (e.g. low, $medium$, and $high$) are defined with this command. An entire power shutoff, equivalent to a supply voltage of $0V$, does not need to be specified separately. Next, all possible power modes are listed, each specified by a *create_power_mode* command line. Thereby, a power mode is defined as a mode-allocation for the power domains. All valid allocations are statically derived out of the controller definition leading to four allocations in the example design. In the default power mode, the default domain is operating at the default nominal condition whereas power domains $PD1$ and $PD2$ are power-gated. In the power mode $PM1$, the power domain $PD1$ is powered on as well whereas $PD2$ remains power-gated. This mode is valid in the controller-step 1. $PM2$ solely power-gates $PD1$ (valid in cstep 10 and 11) and in $PM3$ all domains are powered on as it is the case in the controller step 2. The use of several power-manageable components leads to a huge amount of different power modes dependent on valid states of the controller. Nevertheless, the number of modes in a typical design is significantly lower than the theoretical upper bound of $2^{\#\mathit{functionalunits}}$.

Next, the $create\_isolation\_rule$ command defines a rule for adding isolation cells (voltage anchors) to every power-manageable domain. The command specifies a list of pins to be isolated, the isolation condition, and the isolation output. The isolation condition parameter specifies the condition when the specified pins should be isolated. The isolation output parameter controls the value at the output of the isolation gates when the isolation condition is true. In the current example, the isolation condition is the corresponding sleep signal of the controller and the isolation output *hold* indicates that the output value should be stuck to the value right before the isolation condition is activated. Then, the $update\_isolation\_rules$ command appends the specified isolation rules with implementation information. For example, names of the voltage anchor library cells that should be used as isolation cells can be defined by a $cells$ parameter when they are known in later synthesis steps.

The command *create_power_switch_rule* specifies how a single power switch connects the external and internal power of ground nets for the power domains. In the current example, the external power net is defined because PMOS power-gating is used as power-management technique. At last, the command *update_power_switch_rule* appends a specified rule for power switch logic with implementation information. For example, names of PGS library cells that should be used as power switch cells can be defined. For the lack of semiconductor technologies with contained PGS cells, this information need to be added via a *cells* parameter in later synthesis steps.

```
# Define example design
#---------------------------------------------------------------
set_design example

# Define ground and power nets
#---------------------------------------------------------------
create_ground_nets -nets { GND } -voltage 0.0
create_power_nets -nets { VDD } -voltage 1.0

# Set up logic structure for all power domains
#---------------------------------------------------------------
create_power_domain -name PD0 -default -instances {
          Ctrl Mux0_of_Add_1 Mux1_of_Add_1 Mux0_of_Sub_1
          Mux1_of_Sub_1 Mux0_of_Reg_7 Reg_1 Reg_2 Reg_3
          Reg_4 Reg_7 }
create_power_domain -name PD1 -instances { Add_1 }
                              -shutoff_condition { Ctrl.SLEEP0 }
update_power_domain -name PD1 -internal_power_net VDD_VIRT_PD1
create_power_domain -name PD2 -instances { Sub_1 }
                              -shutoff_condition { Ctrl.SLEEP9 }
update_power_domain -name PD2 -internal_power_net VDD_VIRT_PD2

# Define operating condition under which the design performs
#---------------------------------------------------------------
create_nominal_condition -name default -voltage 1

# Define power modes
#---------------------------------------------------------------
create_power_mode -name PM0 -default -domain_condition { PD0@default }
create_power_mode -name PM1 -domain_condition { PD0@default
                                                PD1@default }
create_power_mode -name PM2 -domain_condition { PD0@default
                                                PD2@default }
create_power_mode -name PM3 -domain_condition { PD0@default
                                                PD1@default
                                                PD2@default }
```

```

# Define required isolation for all power domains
#----------------------------------------------------------------
create_isolation_rule -name ir1 -from PD1
                      -isolation_condition { Ctrl.SLEEP0 }
                      -isolation_output hold
create_isolation_rule -name ir2 -from PD2
                      -isolation_condition { Ctrl.SLEEP9 }
                      -isolation_output hold
update_isolation_rules -names { ir1 ir2 } -location from

# Define required power switch rules for all power domains
#----------------------------------------------------------------
create_power_switch_rule -name psr1 -domain PD1
                         -external_power_net VDD
create_power_switch_rule -name psr2 -domain PD2
                         -external_power_net VDD
update_power_switch_rule -name psr1
update_power_switch_rule -name psr2

end_design
```

Listing 6.3: Common Power Format file of the example design

6.6 Discussion and Summary

The experimental assessment covers industrial as well as academic semiconductor technologies, different node sizes and process corners, several power-gating schemes as well as typical ranges for all model parameters.

All power-gating sub-models have been evaluated separately and statistical error propagation methods are applied for determining an overall error metric. During evaluation it has been shown that mispredictions are mainly constrained to overestimations, limiting the field of application and reducing the efficiency of power-gating. The evaluation has also shown that the models are applicable to be used in the DSE in order to provide design tradeoffs.

At system-level, where all functional units are put to sleep simultaneously, power-gating reduces the leakage currents by up to 98%. The impact of the process corner on this reduction and on the break-even time has been analyzed as well as the advantages of a NMOS-based power-gating scheme have been shown. Furthermore, the effects of the delay-dependent sleep transistor sizing on the area, energy estimation, and quiescent leakage currents have been demonstrated.

The cycle-accurate adoption of the power-gating technique to the out-of-the-box synthesis result of PowerOpt® leads to an energy reduction of about 46% in average for the evaluated benchmarks. Nevertheless, without a dedicated and sophisticated consideration of the power-gating technique during synthesis possible savings may be outweighed for some functional units. Further improvements have been made with the proposed scheduling and binding approaches. It has been shown that the remaining energy demand can again be reduced by up to 43% at an average reduction of 19.8%. Except for the increased runtime, these synthesis improvements result out of the design-space exploration and come for no extra costs.

In addition, the consistency of the developed power-gating methodology and tool integration to the industrial CPF standard has been shown for an exemplary design. This includes the datapath, the power management controller including its temporal behavior and static interconnect as well as the resulting Common Power Format file.

7 Summary and Conclusion

Power-gating has been analyzed to be the most effective leakage reduction technique during runtime. In this thesis it has been used to convert idle times of functional RT-level components into leakage savings.

A set of models has been developed in order to cover the dominating costs of power-gated RTL components and its interfacing circuitry. For all models, relevant parameters have been identified, their impact has been analyzed, and adequate modeling techniques have been proposed. They are supplemented by appropriate leakage, delay, and dynamic power models taken from literature in order to provide a holistic estimation framework covering all operation modes. The evaluation has shown that the model characterization holds for multiple transistor-level implementation types of power-gating and semiconductor technologies in different node sizes and process corners. A coupled model error analysis has shown maximum overestimations of 1.7%, underestimations of 15.7%, and the statistical propagation of uncertainty results in a standard deviation of $\sigma_{rel\text{-}f} = 3.41\%$. In comparison to circuit-level simulations the estimation time reduces from up to several hours to milliseconds.

Beside the pure estimation, the models have also been built to enable an automated optimization of the high-level synthesis. This level of abstraction in combination with the variety of considered costs as well as model parameters differ from existing approaches. The automated insertion of sleep transistors to RTL components requires a sizing approach. Thus, a method has been proposed in order to find the smallest necessary sleep transistor size still satisfying a constrained delay increase. Based on this, heuristic optimizations of the high-level synthesis have been developed with the target of putting components to the sleep state for longer continuous times and thus to reduce the overall energy consumption. At first, an ILP-based scheduler has been developed. Its cost function reduces the number of idle phases and thereby it clusters operations of the same type to enlarge idle times between these clusters. The proposed scheduling serves as a heuristic pre-optimization to the subsequent synthesis phase.

On this basis, two approaches have been developed for the functional unit binding and allocation. The first is data-dependent and considers dynamic, static, and state transition energy costs. Thereby, operation clustering is only done if it is worthwhile in terms of an overall energy reduction. In contrast, the second cost function results in a much faster synthesis as it abstracts from data-dependencies and purely optimizes the operation clustering in order to improve the use of power down techniques. For evaluation all models and synthesis approaches have been integrated into the PowerOpt$^{®}$ high-level synthesis tool. Both binding techniques as well as the proposed scheduling have been evaluated in various combinations and improvements have been discussed on a variety of benchmarks. It has been shown that the power-gating technique is able to reduce the energy consumption of functional units by 46% in average and that the optimized synthesis approaches further reduce the remaining energy demand by up to 43% at an average reduction of 19.8%. Except for an increased runtime, these synthesis improvements result out of the design-space exploration and come for no extra costs.

To propagate the synthesis results to subsequent design phases and tools, an automated power-manager synthesis is formally defined and has been implemented to extend the datapath controller. Thereby, idle times of functional units are analyzed and cycle-accurate sleep signals are created. In the end, the power-manager is exported along with the synthesized datapath in a Verilog file. Furthermore, non-functional meta-information such as the selected power switch, available power domains, and their interconnect are outputted in the industrial Common-Power Format to show the consistency of the developed power-gating methodology and the tool integration.

7.1 Outlook

A vast number of extensions are imaginable in order to improve the estimation and optimization techniques presented in this thesis. In the following most obvious and promising ones are shortly summarized.

- **Transient dynamic virtual ground/supply voltage estimation**
  Right after powering down a circuit, the voltage level at the dynamic virtual ground/supply line increases/decreases to a level close to V_{DD}/GND. In this thesis, for simplicity, it is assumed that the voltage level is saturated immediately. As a consequence, an overestimation occurs as described in Section 4.2.3. Additionally, if a wakeup occurs while the voltage level has not saturated yet, E_{SW}^{RT} is much lower. Future work could integrate more sophisticated approaches as proposed in [XVJ08] to accurately model the virtual ground/supply voltage over time making the optimization techniques even more profitable or applicable for shorter break-even times. An analysis in Section 4.2.4 identified the power-down time to be in the range of microseconds and thus in most of the cycle-wise power-down phases the state transition energy is highly overestimated in this thesis.

- **Functional unit clustering**
  In this thesis functional units are power-gated on an individual basis. In order to reduce the overhead of control signals (but also being less effective in reducing the power) a further clustering could be incorporated as proposed in [DM09].

- **Power-Management Controller Estimation**
  In the recent version of PowerOpt®, the controller is functionally synthesized to a conditional state machine and exported to synthesizable Verilog. Its power and area estimation is based on empiric power models that are scaled in accordance to the controller state count. A more sophisticated controller power estimation approach is necessary to analyze the overheads induced by the power-management functionality.

Symbols

$active$	Active state.
$ALAP$	Function to determine the latest possible time of an operation in a schedule.
$ASAP$	Function to determine the earliest possible time of an operation in a schedule.
$bind$	Function to determine a set of operations that are bound to a functional unit.
bw	Bitwidth of a component.
CS	Set of controller states.
cs	Controller state.
cs	Controller step within a schedule.
$cstep$	Function to determine the controller step of a controller state or of an operation.
c	Cycle.
$cycle$	Function to determine the simulation cycle of a given data pattern.
D^{BUF}	Delay of a buffer.
Δt	Timespan (dependent on context either between two operations or between two data pattern).
δ	State transition function of a Moore machine.
D^{INV}	Delay of an inverter.
D^{RT}	Delay of a RTL component (in active state) (Also: execution time for an operation).
D^{ST}	Delay of sleep transistor.
E	Energy dissipation.
$E_{OVERHEAD}$	Overhead energy.
E_{SW}^{BUF}	Energy dissipation of a buffer.
E_{SW}^{INV}	Energy dissipation of an inverter.
E_{SW}^{RT}	State transition energy of a RTL component.

E_{SW}^{ST}	Energy dissipation of a sleep transistor.
E_{SW}^{VA}	Energy dissipation of a voltage anchor.
FU	Set of functional units.
fu	Functional unit that is capable to execute operations.
GND	Ground voltage.
Hd_1	1-normalized hamming distance.
I_{ACTIVE}	Current in active state.
I^{BUF}	Leakage currents of buffer.
I_{CYCLE_AVG}	Cycle-averaged current.
$idle$	Idle state.
I_{gate}	Gate leakage in general.
I_{GB}	Gate-to-bulk leakage.
I_{GD}	Gate-to-drain leakage.
I_{gidl}	Gate-induced drain leakage.
I_{gisl}	Gate-induced source leakage.
I_{GS}	Gate-to-source leakage.
I_{hci}	Hot carrier injection leakage.
$I_{junction}$	Junction leakage.
I_{MIC}	Maximum instantaneous current during an operation of a component.
I_{MPC}	Maximum power-up current.
I_{OFF}	Channel current of locking transistor (=subthr. leakage).
I_{OFF}^{ST}	Leakage currents of locking sleep transistor.
I_{ON}	Channel current of conducting transistor.
I_{ON}^{ST}	Leakage currents of conducting sleep transistor.
$I_{OVERHEAD}$	Overhead currents.
I_{punch}	Punchthrough leakage.
I^{RT}	Leakage currents of RTL-component (in active state).
I_{SLEEP}	Current in sleep state.
I_{SD}	Source-to-drain leakage.
I_{subth}	Subthreshold leakage.
I^{VA}	Leakage currents of voltage anchor.
λ	Output function of a Moore machine.
L	Channel length.
$MARE$	Mean absolute relative error.

$\#_{cstates}$	Number of states in a controller.
$\#_{csteps}$	Number of controller steps in a schedule.
$\#_{functionalunits}$	Number of functional units.
$\#_{gates}$	Number of gates in a circuit.
$\#_{operations}$	Number of operations.
$\#^{sim}_{cycles}$	Number of simulated cycles.
$\#_{types}$	Number of different component types in a design.
Ω	Output alphabet of a Moore machine.
op	Operation within a schedule.
p	Data pattern (either unary or binary).
pt	Data pattern trace.
P^{RT}_{DYN}	Dynamic power dissipation of RTL component due to activity.
pm	Function to determine the power mode of a given functional unit and simulated cycle.
R_{ST}	Resistance of sleep transistor.
Σ	Input alphabet of a Moore machine.
$sleep$	Sleep state.
σ_{rel}	Relative standard deviation.
s	System (defined as datapath with RTL components and dedicated controller).
t_{be}	Break-even time.
t_{cycle}	Clock cycle time.
T	Temperature.
T_{ox}	Thickness of transistor gate oxide.
$t_{powerdown}$	Power-down time.
t_{sleep}	Sleep time.
t_{wakeup}	Overall wake-up time composed of buffer delay, sleep transistor delay, and RTL component wake-up time.
t^{RT}_{wakeup}	Wake-up time of RTL component from sleep to active state.
t	Operation type.
$type$	Function to determine the operation type of a given operation.

V_{DD}	Supply voltage.
$V^{ST}_{DROP\text{-}OFF}$	Voltage drop accross locking sleep transistor.
$V^{ST}_{DROP\text{-}ON}$	Voltage drop accross conducting sleep transistor.
V_{DS}	Voltage between drain and source of a transistor.
V_{Gate}	Voltage between gate of a transistor and ground.
V_{GB}	Voltage between gate and bulk of a transistor.
V_{GD}	Voltage between gate and drain of a transistor.
$VGND$	Virtual ground voltage.
V_{GS}	Voltage between gate and source of a transistor.
V_{T}	Thermal voltage.
V_{TH}	Threshold voltage of a transistor.
VV_{DD}	Virtual supply voltage.
$wakeup$	Wake up state.
W	Channel width.
W_{RT}	Sum of PMOS and NMOS channel widths of a whole RTL component.
W_{ST}	Sleep transistor channel width.
$XRE\downarrow$	Maximum relative error of under-estimation.
$XRE\uparrow$	Maximum relative error of over-estimation.

Acronyms

ALAP	as late as possible
ASAP	as soon as possible
ASIC	application specific integrated circuit
ATPG	automated test pattern generation
BSIM	Berkeley short-channel IGFET model
CBSD	cluster based sleep transistor design
CCMOS	cutoff CMOS
CCS	composite current source
CDB	component database
CDFG	control- and data-flow graph
CMOS	complementary metal-oxide semiconductor
CPF	Common Power Format
DPM	dynamic power-management
DSE	design-space exploration
DSTN	distributed sleep transistor network
DVFS	dynamic voltage and frequency scaling
EDA	electronic design automation
FDS	force directed scheduling
FU	functional unit
GDSII	graphical design station II
GP	general purpose
HDL	hardware description language
HLS	high-level synthesis
HVT	high threshold voltage
ILP	integer linear program

IP	intellectual property
ITRS	international technology roadmap for semiconductors
LP	low power
MEMS	microelectromechanical systems
MILP	mixed integer linear program
MLV	minimal leakage vector
MTCMOS	multiple threshold CMOS
NMOS	n-channel metal-oxide semiconductor
PG	power-gating
PGS	power-gating scheme
PM	power-management
PMOS	p-channel metal-oxide semiconductor
PTM	predictive technology model
PVT	process-voltage-temperature
RBB	reverse body biasing
RT	register transfer
RTL	register transfer level
SCCMOS	super cutoff CMOS
SCIP	Solving Constraint Integer Programs
Si2	Silicon Integration Initiative
SMIC	Semiconductor Manufacturing International Corporation
SSA	spurious switching activity
SVT	standard threshold voltage
TG	transmission gate
UPF	Unified Power Format
ZSCCMOS	zigzag SCCMOS

Bibliography

[AAE03] Mohab Anis, Shawki Areibi, and Mohamed I. Elmasry. Design and optimization of multi-threshold CMOS (MTCMOS) circuits. *IEEE Transactions on CAD of Integrated Circuits and Systems*, pp. 1324–1342, 2003.

[Ach04] T. Achterberg. Scip - a framework to integrate constraint and mixed integer programming. Technical report, Zuse Institute Berlin, 2004. Technical Report 04-19,`http://www.zib.de/Publications/abstracts/ZR-04-19/`.

[BBMM06] Pietro Babighian, Luca Benini, Alberto Macii, and Enrico Macii. Enabling fine-grain leakage management by voltage anchor insertion. *Proc. of the Design, Automation and Test in Europe Conference and Exhibition 2006 (DATE)*, pp. 868–873, 2006.

[Bla96] Gerard M. Blair. CMOS buffer tapering with interconnect capacitances. *Electronics Letters*, vol. 32, pp. 1984–1985, 1996.

[BS00] J. Adam Butts and Gurindar S. Sohi. A static power model for architects. In *Intl. Symposium on Microarchitecture*, pp. 191–201, 2000.

[CCC09] De-Shiuan Chiou, Shih-Hsin Chen, and Shih-Chieh Chang. Sleep transistor sizing for leakage power minimization considering charge balancing. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, nr. 9, pp. 1330–1334, September 2009.

[CCCY06] De-Shiuan Chiou, Shih-Hsin Chen, Shih-Chieh Chang, and Chingwei Yeh. Timing driven power gating. *Proc. of the 43rd annual conference on Design automation*, pp. 121–124, 2006.

[CCJC10] De-Shiuan Chiou, Yu-Ting Chen, Da-Cheng Juan, and Shih-Chieh Chang. Sleep transistor sizing for leakage power minimization considering temporal correlation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, pp. 1285–1290, August 2010.

[CF95a] Brian S. Cherkauer and Eby G. Friedman. Design of tapered buffers with local interconnect capacitance. *IEEE Journal of Solid-State Circuits*, vol. 30, pp. 151–155, 1995.

[CF95b] Brian S. Cherkauer and Eby G. Friedman. A unified design methodology for CMOS tapered buffers. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 3, pp. 99–111, 1995.

[CF07] Ken Choi and Jerry Frenkil. An analysis methodology for dynamic power gating. Technical report, Sequence Design Inc., 2007.

[CPS$^+$07] Andrea Calimera, Antonio Pullini, Ashoka Visweswara Sathanur, Luca Benini, Alberto Macii, Enrico Macii, and Massimo Poncino. Design of a family of sleep transistor cells for a clustered power-gating flow in 65nm technology. *Proc. of the 17th ACM Great Lakes symposium on VLSI*, pp. 501–504, 2007.

[CSKS08] Eunjoo Choi, Changsik Shin, Taewhan Kim, and Youngsoo Shin. Power-gating-aware high-level synthesis. *Proc. of the 2008 Int'l Symposium on Low Power Electronics and Design (ISLPED)*, pp. 39–44, 2008.

[Dav08] Klaus David, editor. *Technologies for the Wireless Future: Wireless World Research Forum (WWRF)*, vol. 3. John Wiley and Sons, 2008.

[DM09] Deniz Dal and Nazanin Mansouri. Power optimization with power islands synthesis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, nr. 7, pp. 1025–1037, July 2009.

[DYX$^+$07] Mohan V. Dunga, Wenwei Yang, Xuemei Xi, Ali M. Niknejad, and Chenming Hu. BSIM4.6.1 MOSFET model - user's manual. Technical report, Department of Electrical Engineering and Computer Sciences University of California, Berkeley, 2007.

[FV07] Jerry Frenkil and Srini Venkatraman. *Closing the Power Gap between ASIC & Custom: Tools and Techniques for Low Power Design*, chapter 10: Power gating design automation, pp. 251–280. Springer, 2007.

[GGK05] Ranganath Gopalan, Chandramouli Gopalakrishnan, and Srinivas Katkoori. Leakage power driven behavioral synthesis of pipelined datapaths. *Proc. on IEEE Computer Society Annual Symposium on VLSI*, pp. 167–172, 2005.

[GK02] Chandramouli Gopalakrishnan and Srinivas Katkoori. Behavioral synthesis of datapaths with low leakage power. *Proc. of IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 4, pp. 699–702, 2002.

[GK03a] Chandramouli Gopalakrishnan and Srinivas Katkoori. Knapbind: An area-efficient binding algorithm for low-leakage datapaths. *Proc. on the 21st International Conference on Computer Design (ICCD)*, pp. 430–435, 2003.

[GK03b] Chandramouli Gopalakrishnan and Srinivas Katkoori. Resource allocation and binding approach for low leakage power. *Proc. on 16th International Conference on VLSI Design*, pp. 297–302, 2003.

[GK04] Chandramouli Gopalakrishnan and Srinivas Katkoori. Tabu search based behavioral synthesis of low leakage datapaths. *Proc. on IEEE Computer society Annual Symposium on VLSI*, pp. 260–261, 2004.

[HBS+04] Zhigang Hu, Alper Buyuktosunoglu, Viji Srinivasan, Victor Zyuban, Hans Jacobson, and Pradip Bose. Microarchitectural techniques for power gating of execution units. *Proc. of the 2004 Int'l Symposium on Low Power Electronics and Design (ISLPED)*, pp. 32–37, 2004.

[HC09] Shih-Hsu Huang and Chun-Hua Cheng. Timing driven power gating in high-level synthesis. *Proc. of the Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 173–178, 2009.

[Hel09] Domenik Helms. *Leakage Models for High-Level Power Estimation*. PhD thesis, University of Oldenburg, Department for computer science, 2009.

[HEN06] Domenik Helms, Günter Ehmen, and Wolfgang Nebel. Analysis and modeling of subthreshold leakage of RT-components under PTV and state variation. *Proc. of the 2006 Int'l Symposium on Low Power Electronics and Design (ISLPED)*, pp. 220–225, 2006.

[Hen07] Stephan Henzler. *Power Management of Digital Circuits in Deep Sub-Micron CMOS Technologies*. Advanced Microelectronics. Springer, 1-4020-5080-1, 2007.

[HHN06] Domenik Helms, Marko Hoyer, and Wolfgang Nebel. Accurate ptv, state, and abb aware rtl blackbox modeling of subthreshold, gate, and PN-junction leakage. *Proc. of the 2006 Int'l Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, vol. 4148/2006, pp. 56–65, 2006.

[HHN07] Marko Hoyer, Domenik Helms, and Wolfgang Nebel. Modeling the impact of high level leakage optimization techniques on the delay of RT-components. *Proc. of the 2007 Int'l Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, pp. 171–180, September 2007.

[HMHN07] Domenik Helms, Olaf Meyer, Marko Hoyer, and Wolfgang Nebel. Voltage and ABB island optimization in high level synthesis. In *Intl. Symposium on Low Power Electronics and Design (ISLPED)*, pp. 153–158, August 2007.

[HN10] Michael B. Henry and Leyla Nazhandali. From transistors to mems: throughput-aware power gating in CMOS circuits. In *Proc. of the Design, Automation and Test in Europe Conference and Exhibition 2010 (DATE)*, pp. 130–135, 2010.

[HP81] Louis Hafer and Alice C. Parker. A formal method for the specification, analysis, and design of register-transfer level digital logic. *Proc. of the 18th conference on Design automation (DAC)*, pp. 846–853, 1981.

[HSI93] Masashi Horiguchi, Takeshi Sakata, and Kiyoo Itoh. Switched-source-impedance CMOS circuit for low standby subthreshold current giga-scale LSI's. *IEEE Journal of Solid-State Circuits*, vol. 28, nr. 11, pp. 1131–1135, November 1993.

[IaASU] Nanoscale Integration and Modeling (NIMO) Group at Arizona State University. Predictive technology model (PTM).

[Inc] NanGate Inc. Nangate 45nm open cell library. http://www.nangate.com.

[ITR10] ITRS Working Group. *INTERNATIONAL TECHNOLOGY ROADMAP FOR SEMICONDUCTORS: Design*, 2010.

[JKSN99] Gerd Jochens, Lars Kruse, Eike Schmidt, and Wolfgang Nebel. A new parameterizable power macro-model for datapath components. *Proc. of the Design, Automation and Test in Europe Conference and Exhibition 1999 (DATE)*, pp. 29–36, 1999.

[JL75] R.C. Jaeger and L.W. Linholm. Comments on 'an optimized output stage for mos integrated circuits' [and reply]. *IEEE Journal of Solid-State Circuits*, vol. 10, nr. 3, pp. 185–186, June 1975.

[JMSN05] Hailin Jiang, Malgorzata Marek-Sadowska, and Sani R. Nassif. Benefits and costs of power-gating technique. *Proc. of the 2005 Int'l Conference on Computer Design*, pp. 559–566, 2005.

[KAB$^+$03] Nam Sung Kim, Todd Austin, David Blaauw, Trevor Mudge, Krisztián Flautner, Jie S. Hu, Mary Jane Irwin, Mahmut Kandemir, and Vijaykrishnan Narayanan. Leakage current: Moore's law meets static power. *IEEE Computer*, pp. 68–75, December 2003.

[KAP03] Chang Woo Kang, Soroush Abbaspour, and Massoud Pedram. Buffer sizing for minimum energy-delay product by using an approximating polynomial. *Proc. of the 13th ACM Great Lakes symposium on VLSI*, pp. 112–115, 2003.

[KC00] James T. Kao and Anantha P. Chandrakasan. Dual-threshold voltage techniques for low-power digital circuits. *IEEE Journal of Solid-State Circuits*, vol. 35, pp. 1009–1018, 2000.

[KCA97] James T. Kao, Anantha P. Chandrakasan, and Dimitri Antoniadis. Transistor sizing issues and tool for multithreshold CMOS technology. *Proc. of the 34th. Design Automation Conference (DAC)*, pp. 409–414, 1997.

[KIY$^+$98] Kouichi Kumagai, Hiroaki Iwaki, Hiroshi Yoshida, Hisamitsu Suzuki, Takashi Yamada, and Susumu Kurosawa. A novel powering-down scheme for low vt CMOS circuits. In *Symposium on VLSI Circuits. Digest of Technical Papers*, pp. 44–45, 1998.

[KJ00] Kamal S. Khouri and Niraj K. Jha. Leakage power analysis and reduction during behavioral synthesis. *Proc. on 2000 International Conference on Computer Design*, pp. 561–564, 2000.

[KKK03] Suhwan Kim, Stephen V. Kosonocky, and Daniel R. Knebel. Understanding and minimizing ground bounce during mode transition of power gating structures. In *Proc. of the 2003 Int'l Symposium on Low Power Electronics and Design (ISLPED)*, pp. 22–25, August 2003.

[KKK$^+$07] Suhwan Kim, Stephen V. Kosonocky, Daniel R. Knebel, Kevin Stawiasz, and Marios C. Papaefthymiou. A multi-mode power gating structure for low-voltage deep-submicron CMOS ICs. *IEEE Transactions on Circuits and Systems II: Express Briefs*, nr. 7, pp. 586 –590, July 2007.

[KKS04] Suhwan Kim, Stephen V. Kosonocky, and Daniel R. Knebeland Kevin Stawiasz. Experimental measurement of a novel power gating structure with intermediate power saving mode. In *Proc. of the 2004 Int'l Symposium on Low Power Electronics and Design (ISLPED)*, pp. 20–25, August 2004.

[KNC98] James T. Kao, Siva Narendra, and Anantha P. Chandrakasan. MTCMOS hierarchical sizing based on mutual exclusive discharge patterns. *Proc. of the 35th annual conference on Design automation (DAC)*, pp. 495–500, 1998.

[KNS00] Hiroshi Kawaguchi, Koichi Nose, and Takayasu Sakurai. A super cut-off CMOS(SCCMOS) scheme for 0.5-v supply voltage with picoampere stand-by current. *IEEE Journal of Solid-State Circuits*, vol. 35, pp. 1498–1501, 2000.

[Kon11] Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB). *Zuse Institute Mathematical Programming Language (Zimpl)*, 3.2.0 edition, 2011.

[Kru01] Lars Kruse. *Estimating and Optimizing Power Consumption of Integrated Macro Block at the Behavioral Level.* PhD thesis, University Oldenburg, 2001.

[KS04] Vishal Khandelwal and Ankur Srivastava. Leakage control through fine-grained placement and sizing of sleep transistors. *Proc. of the 2004 IEEE/ACM International conference on Computer-aided design*, pp. 533–536, 2004.

[KSJ$^+$99] Lars Kruse, Eike Schmidt, Gerd Jochens, Ansgar Stammermann, and Wolfgang Nebel. Low power binding heuristics. *Proc. of the 1999 Int'l Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, pp. 41–50, 1999.

[KSJ$^+$00a] Lars Kruse, Eike Schmidt, Gerd Jochens, Ansgar Stammermann, and Wolfgang Nebel. Lower bound estimation for low power high-level synthesis. *Proc. of the 13th International Symposium on System Synthesis*, pp. 180–185, 2000.

[KSJ$^+$00b] Lars Kruse, Eike Schmidt, Gerd Jochens, Ansgar Stammermann, and Wolfgang Nebel. Lower bounds on the power consumption in scheduled data flow graphs with resource constraints. *Proc. of the Design, Automation and Test in Europe Conference and Exhibition 2000 (DATE)*, pp. 737–742, 2000.

[KSJN99] Lars Kruse, Eike Schmidt, Gerd Jochens, and Wolfgang Nebel. Lower and upper bounds on the switching activity in scheduled data flow graphs. *Proc. of the International Symposium on Low Power Electronics and Design (ISLPED)*, pp. 115–120, 1999.

[KSvCJ$^+$01] Lars Kruse, Eike Schmidt, Gerd von Cölln (Jochens), Ansgar Stammermann, Arne Schulz, Enrico Macii, and Wolfgang Nebel. Estimation of lower and upper bounds on the power consumption from scheduled data flow graphs. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 9, pp. 3–15, 2001.

[KYF04] Christian Kanzow, Nobuo Yamashita, and Masao Fukushima. Levenberg-marquardt methods for constrained nonlinear equations with strong local convergence properties. *Journal of Computational and Applied Mathematics*, vol. 172, pp. 375–397, 2004.

[LH01] Fei Li and Lei He. Maximum current estimation considering power gating. *Proc. of the 2001 international symposium on Physical design*, pp. 106–111, 2001.

[LH04] Changbo Long and Lei He. Distributed sleep transistor network for power reduction. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, pp. 937–946, 2004.

[LHB$^+$04] Fei Li, Lei He, Joseph M. Basile, Rakesh J. Patel, and Hema Ramamurthy. High-level area and power-up current estimation considering rich cell library. *Proc. of the 2004 conference on Asia South Pacific design automation: electronic design and solution fair*, pp. 899–904, 2004.

[LHB$^+$05] Fei Li, Lei He, Joseph M. Basile, Rakesh J. Patel, and Hema Ramamurthy. Leakage current aware high-level estimation for vlsi circuits. *IEE Proc. on Computers and Digital Techniques*, vol. 152, pp. 747–755, 2005.

[LHS02] Fei Li, Lei He, and Kewal K. Saluja. Estimation of maximum power-up current. *Proc. of the 2002 conference on Asia South Pacific design automation/VLSI Design*, pp. 51–56, 2002.

[LRJD98] Ganesh Lakshminarayana, Anand Raghunathan, Niraj K. Jha, and Sujit Dey. A power management methodology for high-level synthesis. *Proc. of the Eleventh International Conference on VLSI Design: VLSI for Signal Processing*, pp. 24–29, 1998.

[LXHL03] Zuying Luo, Yongjun Xu, Yinhe Han, and Xiaowei Li. Maximum power-up current estimation of power-gated circuits. In *Proc. of the 5th International Conference on ASIC*, vol. 2, pp. 1243–1246, October 2003.

[Mag07] Magma, Mentor Graphics, and Synopsys. *Unified Power Format (UPF) Standard*, 1.0 edition, 2007.

[MDAM96] José Monteiro, Srinivas Devadas, Pranav Ashar, and Ashutosh Mauskar. Scheduling techniques to enable power management. *Proc. of the Design Automation Conference*, pp. 349–352, 1996.

[MKP08] Saraju P. Mohanty, Elias Kougianos, and Dhiraj K. Pradhan. Simultaneous scheduling and binding for low gate leakage nano-CMOS datapath circuit behavioral synthesis. *IET Computers and Digital Techniques (CDT)*, vol. 2, nr. 2, March 2008.

[MMR04] Hamid Mahmoodi-Meimand, , and Kaushik Roy. Data-retention flip-flops for power-down applications. *Proc. of the 2004 International Symposium on Circuits and Systems (ISCAS)*, vol. 2, pp. 677–680, May 2004.

[MR03] Saibal Mukhopadhyay and Kaushik Roy. Modeling and estimation of total leakage current in nano-scaled-CMOS devices considering the effect of parameter variation. In *Intl. Symposium on Low Power Electronics and Design (ISLPED)*, pp. 172–175, August 2003.

[MS02] Kyeong-Sik Min and Takayasu Sakurai. Zigzag super cut-off CMOS (ZSCCMOS) scheme with self-saturated virtual power lines for subthreshold-leakage-suppressed sub-1v-vdd lsi's. *Proc. of the 28th European Solid-State Circuits Conference*, pp. 679–682, 2002.

[NM06] Min Ni and Seda Ogrenci Memik. Thermal-induced leakage power optimization by redundant resource allocation. *Proc. of the 2006 IEEE/ACM international conference on Computer-aided design*, pp. 297–302, 2006.

[Ope] Open Source Community. A upf script 2 cpf script converter. http://upf2cpf.sourceforge.net/.

[PFP08] Ehsan Pakbaznia, Farzan Fallah, and Massoud Pedram. Charge recycling in power-gated CMOS circuits. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, pp. 1798–1811, 2008.

[PK89] Pierre G. Paulin and John P. Knight. Force-directed scheduling for the behavioral synthesis of asics. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 8, pp. 661–679, June 1989.

[PP08] Ehsan Pakbaznia and Massoud Pedram. Coarse-grain MTCMOS sleep transistor sizing using delay budgeting. *Proc. of the Conference on Design, automation and test in Europe*, pp. 385–390, 2008.

[PPD$^+$98] Satyamurthy Pullela, Rajendran V. Panda, Abhijit Dharchoudhury, G. Vijayan, and David T. Blaauw. CMOS combinational circuit sizing by stage-wise tapering. *Proc. of the Design, Automation and Test in Europe Conference and Exhibition 1998 (DATE)*, pp. 985–986, 1998.

[PS82] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall Inc., New Jersey, 1982.

[RDP07] Anand Ramalingam, Anirudh Devgan, and David Z. Pan. Wakeup scheduling in MTCMOS circuits using successive relaxation to minimize ground bounce. *Journal of Low Power Electronics*, vol. 3, pp. 28–35, 2007.

[RHN07] Sven Rosinger, Domenik Helms, and Wolfgang Nebel. RTL power modeling and estimation of sleep transistor based power gating. *Proc. on Int'l Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, pp. 278–287, September 2007.

[RMMM03] Kaushik Roy, Saibal Mukhopadhyay, and Hamid Mahmoodi-Meimand. Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits. *Proc. of the IEEE*, vol. 91, pp. 305–327, February 2003.

[Ros06] Sven Rosinger. Modeling the impact of power management on the energy- and timing-behavior of RT components (original title: Modellierung des Einflusses von Power Management auf das Energie- und Zeitverhalten von RT-Komponenten). Master's thesis, University of Oldenburg, Department for computer science, November 2006.

[RSN09] Sven Rosinger, Kiril Schröder, and Wolfgang Nebel. Power management aware low leakage behavioural synthesis. *Proc. of the 12th Euromicro Conference on Digital System Design, Architectures, Methods and Tools (DSD)*, pp. 149–156, 2009.

[RZDP05] Anand Ramalingam, Bin Zhang, Anirudh Devgan, and David Z. Pan. Sleep transistor sizing using timing criticality and temporal currents. *Proc. of the 2005 conference on Asia South Pacific design automation*, pp. 1094–1097, 2005.

[SA06] Assim Sagahyroon and Fadi Aloul. Maximum power-up current estimation in combinational CMOS circuits. In *IEEE Mediterranean Electrotechnical Conference (MELECON)*, pp. 70–73, May 2006.

[SASN07] Harmander Singh, Kanak Agarwal, Dennis Sylvester, and Kevin J. Nowka. Enhanced leakage reduction techniques using intermediate strength power gating. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, nr. 11, pp. 1215–1224, November 2007.

[SCB$^+$07] Ashoka Visweswara Sathanur, Andrea Calimera, Luca Benini, Alberto Macii, Enrico Macii, and Massimo Poncino. Efficient computation of discharge current upper bounds for clustered sleep transistor sizing. *Proc. of the Design, Automation and Test in Europe Conference and Exhibition 2007 (DATE)*, pp. 1–6, 2007.

[Sch08] Kiril Schröder. Development of a leakage reducing heuristic, optimizing the scheduling at the RT synthesis (Original title: Entwicklung einer Heuristik zur Reduktion der Leckströme durch Optimierung des Schedulings vor der RT-Synthese). Master's thesis, University of Oldenburg, Department for computer science, April 2008.

[SH06a] Kaijian Shi and David Howard. Challenges in sleep transistor design and implementation in low-power designs. *Proc. of the 43rd annual conference on Design automation*, pp. 113–116, 2006.

[SH06b] Kaijian Shi and David Howard. Sleep transistor design and implementation - simple concepts yet challenges to be optimum. *International Symposium on VLSI Design, Automation and Test*, pp. 1–4, 2006.

[SHSB07] Mingoo Seok, Scott Hanson, Dennis Sylvester, and David Blaauw. Analysis and optimization of sleep modes in subthreshold circuit design. *Proc. of the 44th annual conference on Design automation*, pp. 694–699, 2007.

[Sil08] Silicon Integration Initiative. *Si2 Common Power Format Specification (CPF)*, 1.1 edition, 2008.

[SLJY08] Kaijian Shi, Zhian Lin, Yi-Min Jian, and Lin Yuan. Simultaneous sleep transistor insertion and power network synthesis for industrial power gating designs. *Journal of Computers*, vol. 3, nr. 3, pp. 6–13, 2008.

[SMM$^+$97] Satoshi Shigematsu, Shinichiro Mutoh, Yasuyuki Matsuya, Yasuyuki Tanabe, and Junzo Yamada. A 1-v high-speed MTCMOS circuit scheme for power-down application circuits. *IEEE Journal of Solid-State Circuits*, vol. 32, nr. 6, pp. 861–869, June 1997.

[SN90] Takayasu Sakurai and A. Richard Newton. Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas. *IEEE Journal of Solid-state Circuits*, vol. 25, pp. 584–594, 1990.

[SP99] Vijay Sundararajan and Keshab K. Parhi. Low power synthesis of dual threshold voltage CMOS VLSI circuits. *Proc. on 1999 International Symposium on Low Power Electronics and Design*, pp. 139–144, 1999.

[SSCS10] Youngsoo Shin, Jun Seomun, Kyu-Myung Choi, and Takayasu Sakurai. Power gating: Circuits, design methodologies, and best practice for standard-cell vlsi designs. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 15, pp. 1–37, 2010.

[TNN06] Akira Tada, Hiromi Notani, and Masahiro Numa. A novel power gating scheme with charge recycling. *IEICE Electronics Express*, vol. 3, pp. 281–286, 2006.

[Uni09] Unified Power Format technical committee, editor. *1801-2009 - IEEE Standard for Design and Verification of Low Power Integrated Circuits*. IEEE Standards Association, March 2009.

[WAA04] Wenxin Wang, Mohab Anis, and Shawki Areibi. Fast techniques for standby leakage reduction in MTCMOS circuits. In *Proc. of the IEEE International SOC Conference*, pp. 21–24, September 2004.

[Wan08] Eugene Wang. Synopsys power-gating design methodology based on smic 90nm process. Technical report, Synopsys and Semiconductor Manufacturing International Corporation (SMIC), 2008.

[WC10] Xiaobin Wang and Yiran Chen. Spintronic memristor devices and application. In *Proc. of the Design, Automation and Test in Europe Conference and Exhibition 2010 (DATE)*, pp. 667 –672, 2010.

[XVJ08] Hao Xu, Ranga Vemuri, and Wen-Ben Jone. Dynamic virtual ground voltage estimation for power gating. *Proc. of the thirteenth international symposium on Low power electronics and design (ISLPED)*, pp. 27–32, 2008.

[ZPS$^+$03] Yan Zhang, Dharmesh Parikh, Karthik Sankaranarayanan, Kevin Skadron, and Mircea Stan. Hotleakage: A temperature-aware model of subthreshold and gate leakage for architects, March 2003.

List of Figures

List of Tables

List of Listings

Sven Rosinger

Curriculum vitae

Sven Rosinger
Magnolienring 15
26129 Oldenburg
Germany
✉ *svenrosinger@me.com*

Personal Data

Place, Date of birth Nordhorn, April, 29th 1982

Professional Career

11/2006 – today **Scientific assistant**, *OFFIS e.V.*, Division Transportation, Oldenburg.

Academic Studies

10/2002 – 08/2005 **Bachelor-Studium Informatik an der Carl von Ossietzky-Universität Oldenburg**, *Schwerpunkt "Eingebettete Systeme und Mikrorobotik"*, B.Sc. degree.

Bachelor thesis ***Reduktion von fallunterscheidungsinduzierten Bäumen arithmetischer Ausdrücke zur Verbesserung der Feld-Datenabhängigkeitsanalyse für die Stromverbrauchsoptimierung integrierter Schaltungen***, *Advisors: Prof. Dr.-Ing. Wolfgang Nebel and Dipl.-Inf. Mark Hillers.*

10/2005 – 11/2006 **Master-Studium Informatik an der Carl von Ossietzky-Universität Oldenburg**, *Schwerpunkt "Eingebettete Systeme und Mikrorobotik"*, M.Sc. degree.

Master thesis ***Modellierung des Einflusses von Power Management auf das Energie- und Zeitverhalten von RT-Komponenten***, *Advisors: Prof. Dr.-Ing. Wolfgang Nebel and Dipl.-Inf. Mark Hillers.*

Publications

[1] **Sven Rosinger, Domenik Helms, Wolfgang Nebel**, *RTL Power Modeling and Estimation of Sleep Transistor Based Power Gating*, Int'l Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS), 2007.

[2] **Domenik Helms, Marko Hoyer, Sven Rosinger, Wolfgang Nebel**, *RT Level Makro Modelling of Leakage and Delay under Realistic PTV Variation*, First International Workshop on the Impact of Low-Power (LPonTR), 2008.

[3] **Sven Rosinger, Kiril Schröder, Wolfgang Nebel**, *Power Management aware Low Leakage Behavioural Synthesis*, 12th Euromicro Conference on Digital System Design (DSD), 2009.

[4] **Sven Rosinger, Domenik Helms, Wolfgang Nebel**, *RTL Power Modeling and Estimation of Sleep Transistor Based Power Gating*, Journal of Embedded Computing, 2009 (Journal publication of [1]).

[5] **Kim Grüttner, Kai Hylla, Sven Rosinger, Wolfgang Nebel**, *Towards an ESL Framework for Timing and Power Aware Rapid Prototyping of HW/SW Systems*, Forum on Specification & Design Languages (FDL), 2010.

[6] **Sven Rosinger**, *Optimisation of Dynamic Leakage Management in IP Synthesis*, DATE 2010 SIGDA/EDAA PhD Forum, 2010.

[7] **Kim Grüttner, Kai Hylla, Sven Rosinger, Phillip A. Hartmann, Wolfgang Nebel**, *Enabling Timing and Power Aware Virtual Prototyping of HW/SW Systems*, Workshop on Micro Power Management for Macro Systems on Chip (uPM2SoC), 2011.

[8] **Sven Rosinger, Malte Metzdorf, Domenik Helms, Wolfgang Nebel**, *Behavioral-Level Thermal- and Aging-Estimation Flow*, 12th Latin-American Test Workshop (LATW), 2011.

[9] **Sven Rosinger, Malte Metzdorf, Patrick Knocke**, *High-Level Thermal Estimation Flow*, IEEE Elearning Library Courses, 2012.

[10] **Kim Gruttner, Philipp Hartmann, Kai Hylla, Sven Rosinger, Carlo Brandolese, William Fornaciari, Gianluca Palermo, Davide Quaglia, Wolfgang Nebel, Chantal Ykman-Couvreur, Francisco Ferrero, Raul Valencia, Fernando Herrera, Eugenio Villar**, *COMPLEX - COdesign and power Management in PLatform-based design space EXploration*, Digital System Design (DSD), 2012.

[11] **Sven Rosinger, Wolfgang Nebel**, *Sleep-Transistor Based Power-Gating Tradeoff Analyses*, Int'l Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS), 2012.

[12] **Kim Grüttner, Kai Hylla, Sven Rosinger, Wolfgang Nebel**, *Rapid Prototyping of HW/SW Systems using a Timing and Power aware ESL Framework*, System Specification and Design Languages - Selected Contributions from FDL 2010, 2012.

[13] **Reef Eilers, Malte Metzdorf, Sven Rosinger, Domenik Helms, Wolfgang Nebel**, *Phase space based NBTI model*, Int'l Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS), 2012.

Oldenburg, September 27, 2012
Place, Date

Sven Rosinger
Sven Rosinger

Sven Rosinger
Magnolienring 15
26129 Oldenburg

Ehrenwörtliche Erklärung zu meiner Dissertation mit dem Titel:
RT-Level Power-Gating Models optimizing Dynamic Leakage-Management

Hiermit erkläre ich, diese Arbeit ohne fremde Hilfe und nur unter Verwendung der angegebenen Quellen verfasst zu haben. Die Inhalte habe ich weder an der Carl von Ossietzky Universität Oldenburg noch an einer anderen Universität im Rahmen einer Diplomarbeit oder im Rahmen einer Promotion verwendet.

Oldenburg, September 27, 2012
Ort, Datum

Sven Rosinger
Sven Rosinger