FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Testability and Physical Security: The Cell-Level Approach

by

Jan Bělohoubek

A dissertation thesis submitted to
the Faculty of Information Technology, Czech Technical University in Prague,
in partial fulfilment of the requirements for the degree of Doctor.

Dissertation degree study programme: Informatics
Department of Digital Design

Prague, August 2021

Supervisor:

Assoc. Prof. Ing. Petr Fišer, Ph.D.
Department of Digital Design
Faculty of Information Technology
Czech Technical University in Prague
Thákurova 9
160 00 Prague 6
Czech Republic

Co-Supervisor:

Assoc. Prof. Ing. Jan Schmidt, Ph.D.
Department of Digital Design
Faculty of Information Technology
Czech Technical University in Prague
Thákurova 9
160 00 Prague 6
Czech Republic

Abstract and contributions

Reliability, testability, and security belong to the most significant digital design challenges. Notable testability and security problems originate at the physical level, while the solutions may be implemented at higher levels.

This dissertation thesis deals with low-level approaches addressing the high-level design problems, namely the problem of the offline test length and fault coverage, and the problem of the physical security. The proposed solutions are based on enhanced CMOS structures. This dissertation thesis includes also a vulnerability analysis of the conventional CMOS circuit static power and established dynamic power countermeasures such as WDDL or SecLib. A particular contribution touches also on the security-reliability interplay.

In particular, the main contributions of the dissertation thesis are as follows:

1. Conceptual design of the short-duration offline test. The proposed fast offline test may be incorporated into the normal computation flow and potentially replace the online test in many cases while reducing delay and area penalty at the same time.

2. A method for designing a system with increased reliability incorporating the proposed approach is described and its efficiency is shown.

3. A novel CMOS design thereat is described, its severity is proved by simulation, and feasible physical attack scenarios are described. The threat arises especially in redundant structures like voters.

4. The described threat endangers also the dynamic power balancing countermeasures like SecLib and other conventional dual-rail-based countermeasures in general.

5. CMOS circuit-level (standard-cell level) attack countermeasures are proposed and evaluated. The proposed standard cells may be used as a direct replacement of conventional CMOS cells in a standard design process.

Keywords:

combined attack, data-dependency, domino logic, dynamic logic, error detection, error correction, fault, fault coverage, fault model, invasive attack, offline test, optical beam induced current, online test, power analysis, power imprint, self-checking, side-channel, standard cell

Acknowledgements

First of all, I would like to express my gratitude to my dissertation thesis supervisors Petr and Jan for their constant help, encouragement, countless discussions, proofreading, excellent collaboration, and friendship.

I would like to express special thanks to my current and former colleagues from the Department of Digital Design at CTU in Prague, ASICentrum spol. s r.o., and the Department of Materials and Technology at UWB in Pilsen, who maintained a pleasant, flexible and stimulating environment, and for their friendly advice, fruitful discussion, and collaboration.

My greatest thanks go to my family, and especially to my wife Klára for her infinite patience, care, and love.

My research has been partially supported by the grant GA16-05179S of the Czech Grant Agency, "Fault-Tolerant and Attack-Resistant Architectures Based on Programmable Devices: Research of Interplay and Common Features" (2016 – 2018), by CTU grants SGS14/105/OHK3/1T/18, SGS15/119/OHK3/1T/18, SGS16/121/OHK3/1T/18, SGS17/213/OHK3/3T/18 and the OP VVV MEYS funded project CZ.02.1.01/0.0/0.0/16019/0000765 "Research Center for Informatics".

To my wife, Klára

Contents

List of Figures

List of Tables

Abbreviations

ATPG	Automated Test Pattern Generator
BIST	Built-In-Self-Test
CMOS	Complementary Metal–Oxide–Semiconductor
CPA	Correlation Power Analysis
DARPA	Defense Advanced Research Projects Agency
DFT	Design For Test
DPA	Differencial Power Analysis
DUT	Device Under Test
EDA	Electronic Design Automation
FPGA	Field-Programmable Gate Array
IDD	Supply current
IDDQ	Quiescent supply current (IDD)
LPA	Leakage Power Analysis
LSI	Large-Scale Integration
MOS	Metal Oxide Semiconductor
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
MPW	Multi-Project Wafer
MSI	Medium-Scale Integration
NDA	Non-Disclosure Agreement
NMR	N-Modular Redundancy
OBIC	Optical Beam Induced Current
PDK	Process Design Kit
PLS	Photoelectric Laser Stimulation
SCMOS	Scalable CMOS
SCR	Space Charge Region
SPICE	Simulation Program with Integrated Circuit Emphasis
SSI	Small-Scale Integration
TED	Time-Extended Duplex
TMR	Triple Modular Redundancy
TSC	Totally Self-Checking
TSMC	Taiwan Semiconductor Manufacturing Company
VLSI	Very Large-Scale Integration
WDDL	Wave Dynamic Differential Logic

Chapter 1

Introduction

This dissertation thesis represents a contribution to the testability and security aspects of CMOS circuits design. Several notable testability and security problems originate at the physical level, while the solutions might be addressed on higher levels. Higher-level modeling introduces a certain level of simplification, often enabling an elegant design. On the other hand, the low-level approach allows addressing the root of the problem leading to a precise and compact design. This dissertation thesis addresses the CMOS circuit basic building blocks and continues down to the circuit and technology levels.

1.1 Motivation

There are many challenges in digital design today. To the most significant challenges belongs the reliable, safe, and secure design of digital systems supported by threat and reliability analysis [66, 70, 131]. Different approaches may be employed at each level of abstraction separately or combined to enhance security or reliability of the digital device. Similarly, unintended side-effects of high-level design decisions or even algorithm properties may corrupt security or reliability of the digital device [69, 92, 135]. Several levels of abstraction may be addressed starting at the algorithm level, through the architecture level, down to the microarchitecture and circuit levels.

The high-level design of today's digital design is enabled by a careful design of the basic building blocks – CMOS *standard cells.* CMOS cells are conventionally designed according to functional, power, and delay or area requirements [131]. Less conventionally, system-level reliability, testability, or security requirements are propagated down to the circuit level. It is advantageous to have a compact and universal library of standard cells. Therefore, approaches using standard cells are often preferred [121]. However, some of the security challenges are hard to overcome without the circuit level optimization leading to CMOS cell library extension.

This dissertation thesis represents a contribution to the low-level *design for testability* and *design for security* methods and vulnerability analysis. The method enabling a *short-duration* offline test of a combinational circuit was developed. The method is enabled by a proposed CMOS cell-level design. Secondly, a new static-power-related vulnerability was identified and analyzed. This vulnerability potentially endangers the vast majority of today digital circuits, including the side-channel protected ones. At the same time, new CMOS cell-level approaches were developed to face the threat.

Some of the approaches presented in this dissertation thesis follow the known approaches, which experienced historically several times a great success in the digital design area. Successful examples of a strategy closely connected to the topic of this dissertation thesis are described below.

In the 1980s, the *domino logic* circuit design style was introduced to enhance high-performance CMOS circuits. In domino logic, system performance was increased by significant paradigm change at the lower design level(s). The domino logic optimizes the circuit delay by decreasing the in-circuit parasitic capacitances. In such a circuit, special – two-phase – dynamic logic gates with a footprint smaller than conventional CMOS gates are employed in connection with the dual-rail circuit style and the precharge clock in combinational logic. Naturally, the domino logic suffers also from several disadvantages and due to them it was not widely adopted in the *design mainstream*, however, it was a relatively popular design method of high-performance circuits in the 1980s.

Dynamic logic in general and domino-logic in particular, is a design style dedicated to high-performance circuit design [131]. It offers many nice properties and challenges [113]. Since its introduction, domino-logic attracts many improvement efforts [71, 85, 109, 120]. The nature of the domino logic was also one of the inspirations of the first part of this dissertation thesis, as the proposed domino-like structures are used to achieve increased circuit (pseudo-online) testability and therefore reliability.

In the 1990s, Kocher et al. introduced the implementation attacks [92, 93]. Since their introduction, implementation attacks remain a significant threat for digital circuits: many efforts have been made to increase digital circuit attack resistance and novel attacks and attack countermeasures are constantly under development [15, 42, 52, 84, 98, 111, 121]. The implementation attacks use the properties of the implementation technology to break the secret stored in the circuit instead of breaking the (mathematical) principle of the implemented algorithm. The attacks on CMOS *dynamic power*, including *Differential Power Analysis* (DPA) or *Correlation Power Analysis* (CPA), are the most cited and well addressed in the literature. The probably most widely used approach to fight against the dynamic power data dependency is called *hiding*. Hiding mechanisms are incorporated into many real designs [53, 61, 88]. Many hiding approaches are based on dual-rail encoding [52, 111, 121] introducing almost constant (data-independent) dynamic power consumption, at least at the cycle level. As the dynamic power data dependency is closely connected with gate parasitics, delays, or even hazards [52], the CMOS gate, or even cell-level hiding approaches were proposed. In the security area, the low-level design is conventionally used to increase system security [52, 121].

One of the less discovered aspects of implementation attacks are attacks on *static power* consumption [84]. Recently, we described a similar class of attacks on *photoinduced static current* [A.5, A.6, A.7]. As the challenges connected to static power in general, and photoinduced static current in particular, are originated in the nature of CMOS technology, it is convenient to use the CMOS cell-level optimization to increase the system security.

This dissertation thesis deals with *design for testability* and *design for security*. Novel CMOS structures employed to increase system security and reliability are presented and the CMOS vulnerability is presented and analyzed.

1.2 Problem Statement

Two challenges distinct at the system/device level are the subject of this dissertation thesis. Although the challenges are distinct at the system level, both are addressed at the circuit level in general and the CMOS cell-level in particular.

The first addressed problem is the on-line/off-line testability interplay problem. Online testing is employed in systems with an increased level of resilience to achieve a certain level of system reliability. The online test allows to detect system error and take action to mitigate error consequences or even mask the erroneous system output, while offline testing is applied to detect faults and identify faulty parts.

Online testing has two requirements: high error coverage and high test speed. This can be naturally achieved by a design employing area redundancy and checkers. Unfortunately, employing redundancy brings high area and/or delay overhead. Offline testing is conventionally less constrained by test time (compared to online testing) and involves smaller area overhead. On the other hand, it is conventionally not possible to guarantee the functional correctness of the device even when the test fault coverage is high.

The second addressed problem is related to the research of the security threat originating in the CMOS circuits first described by the author in [A.6]: the secret value processed by the CMOS combinational logic may be compromised by a combined attack employing power measurement and *Photoelectric Laser Stimulation* (PLS) of the target device. The severity of this vulnerability is significant, as it may be used to compromise even the trusted circuits with state-of-the-art attack protections. This is caused by the fact that the state-of-the-art physical attacks on CMOS devices are targeted on the circuit dynamic power, but the attack exploiting data-dependency of the PLS allows to overcome many industry-standard dynamic-power attack countermeasures.

1.3 Goals of the Dissertation Thesis

This dissertation thesis deals with reliability and security challenges. The CMOS-based design properties are enhanced by the tunning performed at the CMOS cell-level:

1. A fast *short-duration offline test*, enabled by special CMOS cell design, is proposed. The fast offline test may be incorporated into normal computation flow and potentially replace the online test in many cases while reducing delay and area penalty at the same time.

2. The method for designing a system with increased reliability incorporating the proposed approach is described and its efficiency is shown.

3. A novel CMOS threat is described, its severity is proved by simulation, and feasible physical attack scenarios are described.

4. CMOS circuit-level (standard-cell level) attack countermeasures are proposed and evaluated. The proposed standard-cells may be used as a direct replacement of conventional CMOS cells in the common design process.

1.4 Structure of the Dissertation Thesis

This dissertation thesis is organized into six chapters as follows:

1. *Introduction*: Describes the motivation and goals of the dissertation thesis.

2. *Background and State-of-the-Art*: Introduces the reader to the theoretical background and terminology, and surveys the fundamental state-of-the-art and related work.

3. *Design for Testability*: Provides an overview of the state-of-the-art related to the contribution of this dissertation thesis in the DFT area, and continues with its presentation: the principal explanation of the proposed DFT method enabling a really short offline test is provided.

4. *Side-Channel Attacks*: Provides an overview of side-channel attacks, summarizes the related work, and continues with the contribution of this dissertation thesis: the chapter describes how the proposed approach discloses processed data in a static power channel in a particular case.

5. *Balanced Standard Cells*: Describes the proposed standard cell structure and provides a case study on the AES SBOX block.

6. *Conclusions*: Summarizes the results of the dissertation thesis and suggests the possible directions in further work.

Chapter **2**

Background and State-of-the-Art

> *Transistor is not an Ice-Cream-Bar . . .*
>
> *Prof. Pinker, Introduction to Digital Systems*

This dissertation thesis deals with CMOS combinational circuits, their structure, and circuit-level structure consequences on testability, side-channel vulnerability, and attack resistance. A bit of distinct language is used by security and testing communities, even though obvious analogies exist: interestingly, both testing and (physical) security have a common playground – the aim is to extract *information from within the circuit. The design of the CMOS circuit basic building blocks –* standard cells *– may be exploited to simplify or complicate the extraction process according to the context and design requirements. This chapter summarizes the current state-of-the-art in both testing and security areas and provides the necessary introduction.*

Section 2.1 gives a summary of the theory related to CMOS technology and static and dynamic power channels, Section 2.2 provides a short introduction to the CMOS *Photoelectric Laser Stimulation* (PLS), Section 2.3 summarizes the state-of-the-art in the diagnostics, testing and design for test.

2.1 VLSI CMOS Technology

The early integrated circuits in the 1950s were realized exclusively in the *bipolar* transistor technology introduced by Bell Labs. Later, in 1960s, started the success story of the *unipolar Metal Oxide Semiconductor Field Effect Transistors* (MOSFETs) [131]. The predominant advantage of MOSFETs lies in their low power consumption in the idle state compared to bipolar transistors. The MOSFETs require (almost) no current to keep the device in the ON state compared to a bipolar transistor, where the transistor is controlled by its *base* current.

The MOSFET transistor comes in two types distinguished by the *majority carrier*: P-type (PMOS) where the majority of carriers are *holes* and N-type (NMOS) where the majority of carriers are *electrons*. The early commercially available processes utilizing MOSFETs to create integrated circuits manufactured by the *planar* process were *pMOS* and later *nMOS* processes [131].

The pMOS process allows the implementation of pull-up function (transistor structure connected to the supply rail) only[1], while nMOS allows pull-down function only (transistor structure connected to the ground rail). In the circuit implemented using pMOS or nMOS, the complementary function is implemented by a pull-down or pull-up resistance introducing idle current. Thus the nMOS and the pMOS process still introduce a significant power consumption in the idle state.

The growing size of integrated circuits and unacceptable power consumption and speed limitations of devices manufactured by mainstream processes caused the wide adoption of the more complex and more expensive *Complementary Metal–Oxide–Semiconductor* (CMOS) in the 1980s. The CMOS uses both flavors of MOSFET transistors on a single *substrate*. It allows implementing complementary pull-up and pull-down functions simultaneously, cutting the idle current drastically, while allowing higher circuit performance at the same time [131].

The degree of integration in the semiconductor industry is traditionally indicated by SSI, MSI, LSI, and VLSI acronyms. The small ICs, including first integrated circuits, are denoted as *Small-Scale Integration* (SSI), the term *Medium-Scale Integration* (MSI) denotes later circuits with hundreds of on-chip transistor, while the term *Large-Scale Integration* (LSI) denotes CMOS circuits with tens of thousands transistors. The LSI circuits were first produced during the 1970s. Since hundreds of thousands of transistors are integrated on a chip since the 1980s, the term *Very Large-Scale Integration* (VLSI) is used for advanced circuits up today [119].

Today digital devices are almost exclusively manufactured using a CMOS planar process. Nowadays, however, the importance of the idle consumption of the integrated circuits has araised. As the size of CMOS devices shrinks down, the *channel leakage* rises. When hundreds of thousands or even millions or billions of transistors are integrated on a single chip manufactured in a recent process, its idle or *static power* consumption becomes high.

[1] Negative supply voltage was also commonly used in pMOS, in this case, pMOS implements the topological pull-down function

2.1.1 CMOS Technology and Planar Process

The *Metal-Oxide Semiconductor* (MOS) devices are built on single crystals of silicon (Si) available as thin circular *wafers* of 15 – 30 cm diameter. As silicon is a group IV element with low conductivity, the group III and V *dopants* are used to increase silicon conductivity. The conductivity of doped silicon is given by missing (P-type) or added (N-type) valence electrons [131].

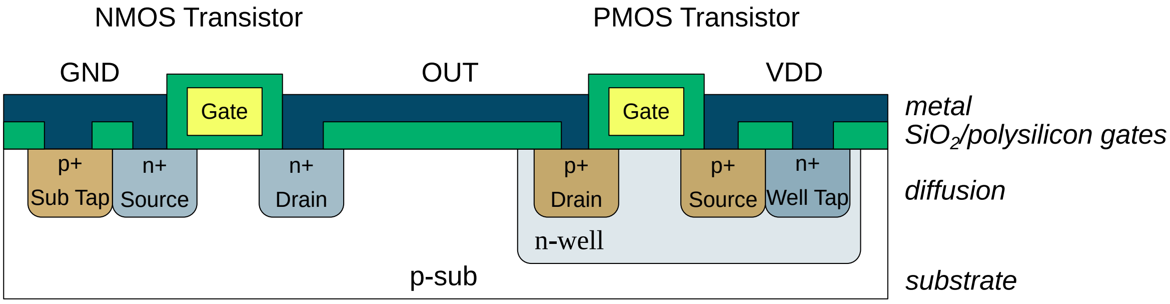


Figure 2.1: Conventional bulk CMOS inverter cross-section

The MOS structure is a sandwich-like *semiconductor* structure formed by conducting and insulating layers. The conventional *planar process* involves several chemical processing steps, including the introduction of dopants, oxidation of the silicon, and deposition and etching of metal layers [131].

The *Complementary MOS* (CMOS) process combines complementary P-type and N-type MOS devices on a single *substrate* (bulk). The complementary arrangement ensures that the idle current is conventionally negligible [131].

Figure 2.1 shows the CMOS device cross-section. The NMOS transistor is built on a P-type body with N-type *source* and *drain* regions, the body is conventionally grounded. The PMOS transistor is conventionally built on a N-type body in a speciall *well* with P-type *source* and *drain* regions, while its body is conventionally connected to the supply voltage. The control gates are insulated by the *silicon dioxide* (SiO_2) and are built of *polysilicon* or *metal* [131].

The voltage between the control electrode (gate) and transistor body affects the charge concentration in the thin body region under the gate. If the voltage is high enough, the *channel* is formed between the source and drain and the transistor becomes closed (ON) [131].

Materials used to manufacture the conventional CMOS transistors, the manufacturing processes, and the CMOS planar structure incorporating many PN-junctions enabled the great success of this technology in the past 50 years. The technology, however, determines also hidden properties of CMOS, influencing not only further technology scaling, but also typical manufacturing defect nature, or static and dynamic power data dependency, or the environment sensitivity in general and the light sensitivity in particular.

2.1.2 Static and Dynamic CMOS

Conventional CMOS circuits are *static*. Static CMOS gates use *complementary* PMOS and NMOS networks to compute output. The advantage of the static CMOS is namely the straightforward design, energy efficiency, and robustness. [131]. Alternative circuit families were developed to address speed, power, or area constraints. The approaches employed in CMOS *dynamic logic* influenced this dissertation thesis.

A dynamic gate works in two alternating phases: *precharge* and *evaluation*. In the first phase, the gate is forced to a defined state (by a dedicated control) and in the second phase, the gate inputs are evaluated.

Assume that the control signal, called *clock*, forces the gate output to `1` during precharge. In the evaluation phase, the output remains `1` or switches to `0`, depending on the input values, as shown in Figure 2.2a. This design style significantly reduces the load at the gate inputs and also the gate size compared to static CMOS, because the gate inputs drive NMOS transistors only.

Dynamic gates exist in two flavours distinguished by a presence of the NMOS *foot* transistor [131]: an *unfooted* gate is shown in Figure 2.2a, and a *footed* gate in Figure 2.2b.

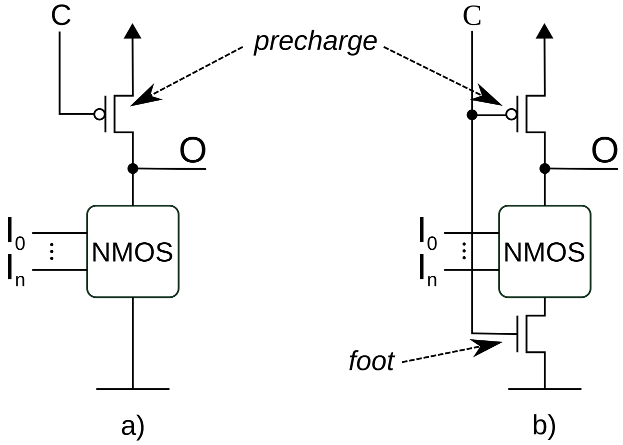


Figure 2.2: a) dynamic-logic gate and b) footed dynamic-logic gate

The disadvantage of the dynamic logic is that it employs a high fan-out clock signal. This disadvantage is much lower than one would expect, because:

- the clock controls only one (a single PMOS for unfooted) or two (one PMOS and one NMOS for footed gates) transistors per gate,
- clock-controlled transistors may be relatively small because the design tolerates longer rising delays (up to half of the computational cycle for 50% clock duty cycle),

thus the load caused by transistor gates is relatively low. The main issue is that there is the need for additional (balanced) metal wires to distribute the clock signal.

The main issue with the dynamic gates described above is that they require *monotonically rising* inputs during evaluation. The outputs of gates described above are *monotonically falling* (during evaluation) – this implies that those gates cannot be simply concatenated to form deeper circuits. Other notable issues connected with dynamic logic in general are *charge leakage* and *charge sharing* [131].

2.1.3 Domino Logic

The concatenation of dynamic gates is enabled by inserting a static CMOS inverter at the dynamic gate output – the design style employing static inverters is called *domino logic*. Domino logic gate outputs are *monotonically rising* during evaluation [131] – see Figure 2.3.

Domino logic is a dedicated logic family belonging to dynamic logic [131], recently popular for high-performance chip design [131].

The precharge function can be realized by a single PMOS transistor only. If it is not guaranteed that the gate inputs are always `0` during precharge, it may be necessary to add an additional NMOS *foot* transistor, as shown in Figure 2.2b.

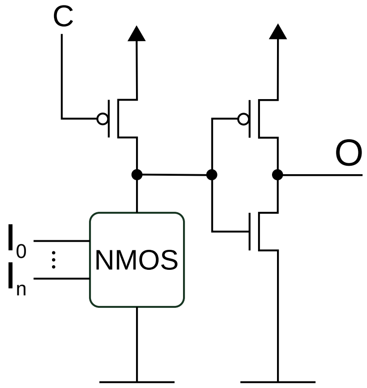


Figure 2.3: Domino-logic gate

The overall advantage of domino logic is the gate size and speed. The *mobility ratio* for holes/electrons is 2 – 3. This causes that PMOS transistors have to be bigger than the NMOS ones to achieve the same conductivity [131]. When the dynamic domino AND and OR gates with precharge to zero are used, the number of PMOS transistors is reduced significantly, compared to the number of NMOS transistors.

Domino logic thus represents a trade-off by providing faster and smaller gates with reduced static power and increased dynamic power.

Note that in domino logic, monotonicity is required, thus the circuit design conventionally employs dual-rail encoding to assure monotonicity.

2.1.4 Static and Dynamic Power

A *static CMOS logic gate* in general has an *NMOS pull-down transistor network* (N) and a *PMOS pull-up transistor network* (P) [131], as illustrated in Figure 2.4. The P/N parts arrangement ensures that only one of both parts is ON and the other is OFF in idle state for any combination of input values.

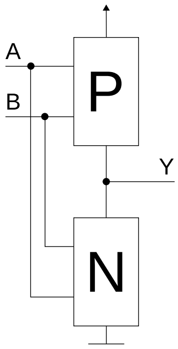


Figure 2.4: Generalized (2-input) CMOS gate structure

The static CMOS gate requires a significant amount of energy to change its output – it is called *dynamic power*. The dynamic power has a pulsed nature and it is composed mainly of the *load capacitance* charge/discharge and also by the *short current*. The short current is a short-duration part of the dynamic power arising from the simultaneously open PMOS and NMOS networks while switching. The dynamic power is commonly expressed by the following simplified (integral) equation [131]:

$$P_d = \alpha \cdot C \cdot V_{dd}^2 \cdot f, \tag{2.1}$$

where V_{dd} is the *supply voltage*, f is the *switching frequency*, C is the load capacitance being charged/discharged and α is the activity factor expressing the average frequency of the gate output change. In static CMOS, the activity factor for the datapath is in most cases up to 0.5 (one transition per cycle), while conventionally it is close to 0.1 [131]. The load capacitance of a single gate is only charged/discharged when the output of the gate changes from $0 \rightarrow 1$ or $1 \rightarrow 0$.

The other component of CMOS gate power consumption is called *static power* or *leakage*. The static power represents the CMOS power in case of no switching activity. Static power is technology-dependent, and ideally, it should be minimized. Static power can be expressed by the following simplified (integral) equation [131]:

$$P_s = I_s \cdot V_{dd}, \tag{2.2}$$

where V_{dd} is the *supply voltage* and I_s is the *static current*. It is not surprising that I_s for a particular gate depends on many variables including manufacturing *process parameters* and variability, logic gate *geometry* (parallel vs. serial connection of transistors), and size.

The static curent is composed of several components:

$$I_s = I_{sub} + I_{gate} + I_{junct}, \tag{2.3}$$

where I_{sub} is the *subthreshold* leakage, which is the most important source of leakage in technologies with thicker gate oxide layer; I_{gate} is the *gate* leakage experienced by transistors with gate oxide thinner than 2nm [131] – in recent technology nodes, this type of leakage may overcome I_{sub}; and I_{junct} is the *junction* leakage.

The I_{gate} is a minority leakage source compared to the subthreshold leakage in technologies above 100nm [58], but it increases significantly faster than the I_{sub} in recent technology nodes [72]. In a typical low-power CMOS process, care is taken to minimize I_{gate}, therefore the I_{sub} is dominant.

The junction leakage depends on the device geometry and junction biasing. The diodes formed on P-N junctions between diffusion and substrate or well might be sources of leakage if forward biased. But commonly, the substate is connected to VSS and well to VDD, ensuring that all junctions are reverse biased. The I_{junct} is normally below 0.1 fA, which is negligible compared to I_{sub} and I_{gate}.

Subthreshold Leakage

Subthreshold leakage is the dominant source of leakage in the majority of technologies. Subthreshold leakage represents the channel current between drain and source of the MOSFET when the transistor is in the OFF state (gate-to-source voltage is below the threshold voltage). For conservative technology nodes (above 100nm), subthreshold leakage is the dominant leakage source in CMOS devices. The subthreshold leakage depends on environmental parameters such as temperature and interestingly, it is also input-dependent. Figure 2.5 gives an idea about the impact of the subthreshold leakage to a 100k gate circuit in a conservative 180nm technology node under rising temperature.

The subthreshold leakage model and main parameters are described e.g. in [130, 131]. The following equation expresses the subthreshold leakage:

$$I_{sub} = I_{ds0} \cdot exp(\frac{V_{gs} - V_{t0} + \eta V_{ds} - k_\gamma V_{sb}}{n v_T}) \cdot (1 - exp(\frac{-V_{ds}}{v_T})), \tag{2.4}$$

where I_{ds0} is a process parameter defining the drain current at the threshold voltage $V_{gs} = V_t$ and the given temperature; the term $V_{t0} + \eta V_{ds} - k_\gamma V_{sb}$ expresses the *threshold* voltage dependency on V_{ds} and V_{sb} – see Figure 2.6; η is the *Drain-Induced Barrier Lowering* (DIBL) coefficient expressing the threshold voltage dependency on V_{ds} affecting strongly the short-channel transistors; k_γ is the body effect coefficient; v_T is the *thermal voltage* expressing the dependency on temperature. The v_T value is 26mV at room temperature and it is rising with rising temperature – Figure 2.7 illustartes the I_{sub} temperature dependency.

Gate Leakage

The gate leakage is caused by the *direct tunneling* effect experienced by transistors with gate oxide thinner than 2nm (20Å). PMOS transistors with the same gate thickness nor-

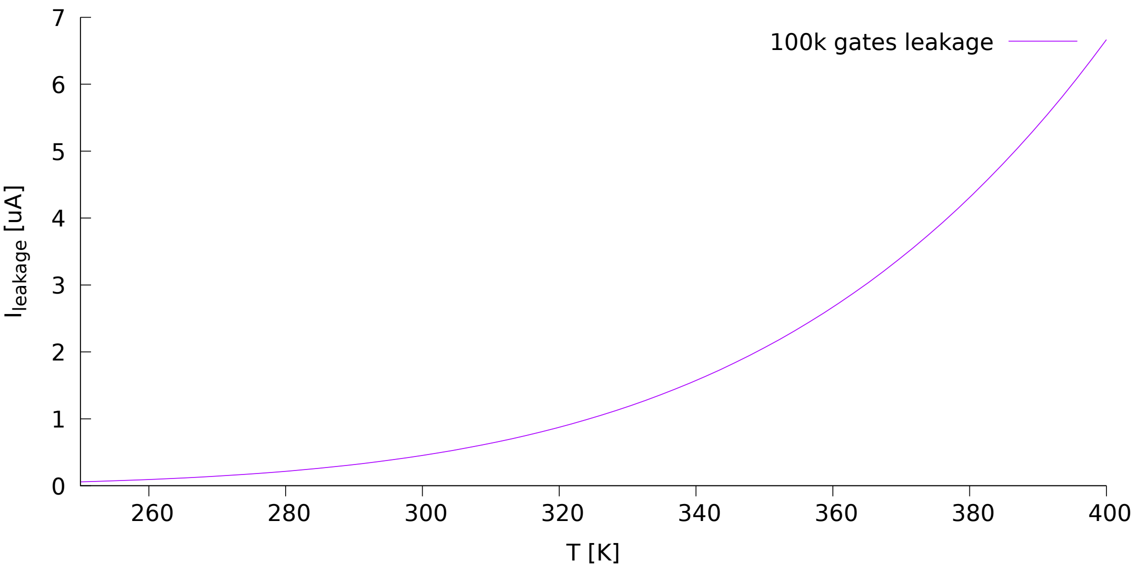


Figure 2.5: Temperature dependency for circuit composed of 100k gates in TSMC180nm – a simplified model based on a single inverter bahaviour where transistors are perfectly closed (NMOS $V_{gs} = 0$ and $V_{ds} = V_{DD}$)

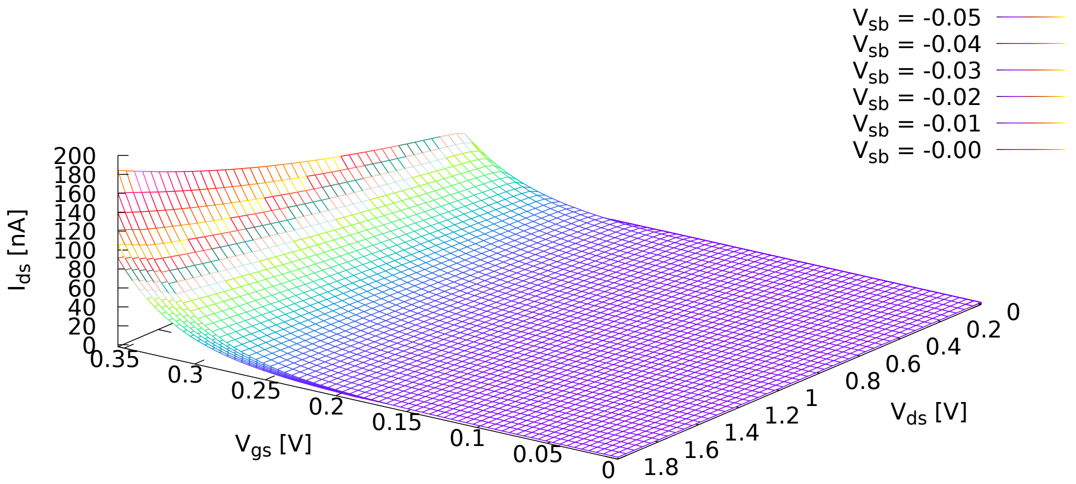


Figure 2.6: Subthreshold leakage current model for TSMC180nm at the room temperature: falling V_{sb} causes increase in the leakage as well as rising V_{gs} and V_{ds}

mally experience a significantly lower gate leakage, as the electrons tunnel from the conduction band but holes in PMOS tunnel from the valence band over a higher barrier [131].

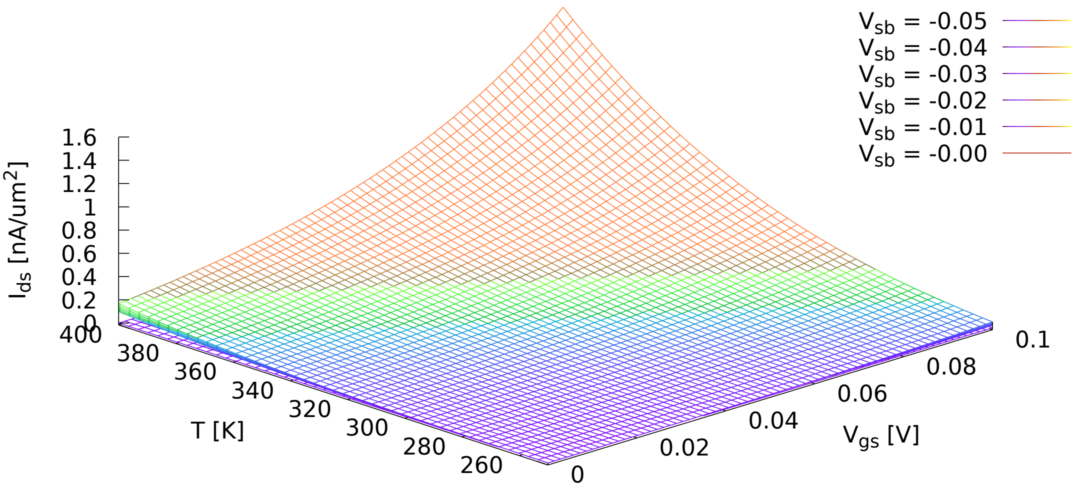


Figure 2.7: Temperature dependency subthreshold leakage current model for TSMC180nm; $V_{ds} = V_{DD} = 1.8V$; Channel area $= 0.2 \times 2.0\ \mu m$

In a conservative or low-power process, the PMOS gate leakage is typically negligible.

The gate leakage is strongly dependent on the insulator thickness and permittivity and loosely dependent on the supply voltage. The leakage current density depends on the supply voltage and the technology node for the most common insulator, which is the *silicon dioxide* (SiO_2), is depicted in Figure 2.8. For processes where the gate dielectric is thin, materials with higher permittivity must be used in place of silicon dioxide [67].

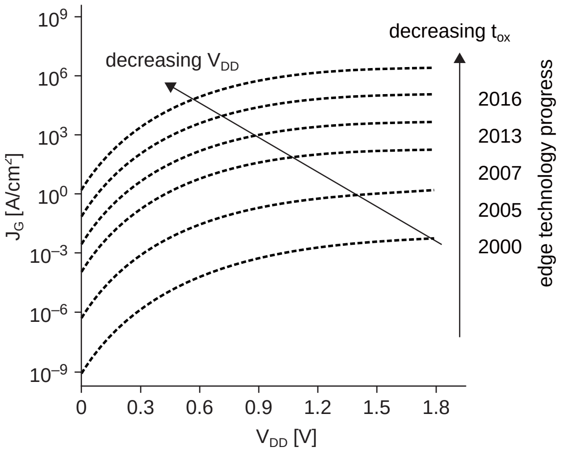


Figure 2.8: Gate leakage as a function of the supply voltage (V_{DD}), the SiO_2 gate oxide thickness (t_{ox}), and the technology generation [9, 131]

The gate leakage is also strongly connected with the state of the transistor – the area of the open channel region is extremely important – see Figure 2.9.

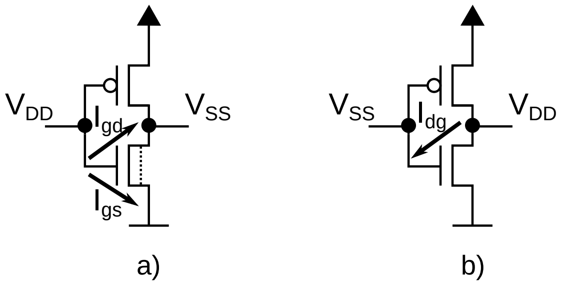


Figure 2.9: a) the closed NMOS transistor experiences a full gate leakage – tunneling to the channel region is signifficant; b) the open NMOS transistor experiences negligible reverse gate leakage, as the drain/gate overlap is small [72]

2.2 CMOS Photoelectric Laser Stimulation

The laser beam passing through silicon creates, as a result of energy absorption, electron-hole pairs along its path. In *Space Charge Regions (SCR)* of PN junctions, the generated electron-hole pairs are separated by the internal electric field, generating the *Optical Beam Induced Current* [56, 73, 107]. The principle behind OBIC is called a *photoelectric effect* – see Figure 2.10.

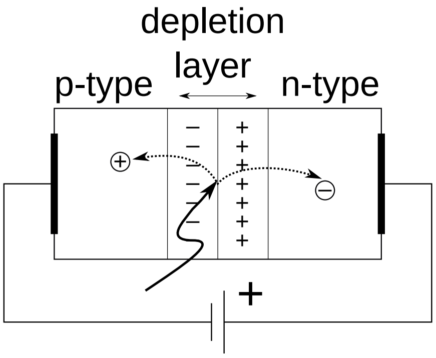


Figure 2.10: In the reverse-biased PN junction, the electron-hole pairs are seperated by the electric field in the depletion layer: OBIC is induced

The photoelectric effect is in the context of CMOS often used for diagnostics [54, 56] or fault injection with precise location control [66, 100].

Sarafianos et al. published a series of papers related to *Photoelectric Laser Stimulation* (PLS), incrementally describing the electrical model of the pulsed photoelectric laser stimulation of an NMOS and PMOS respectively, e.g., [104, 105, 107].

The Sarafianos et al. model includes photocurrents induced at the bulk CMOS PN junctions: *p+/n-well*, *n+/p-sub* and *p-sub/n-well*, as shown in Figure 2.1.

The photocurrent induced by a laser beam in any PN junction is modeled primarily by a voltage controlled current source – see Figure 2.11. The current amplitude is expressed by the following equation:

$$I_{laser} = (a \cdot V + b) \cdot \rho \cdot S, \tag{2.5}$$

where S is the surface of the sensitive zone ($[\mu m^2]$), a and b are fitting parameters expressing the laser power and technology parameters, V is the reversed bias voltage of the PN junction under laser illumination. Parameters a and b express the dependency on the laser power ($[mW]$) by using fitting parameters [107]:

$$a = p \cdot P_{laser}^2 + q \cdot P_{laser} \tag{2.6}$$

$$b = s \cdot P_{laser} \tag{2.7}$$

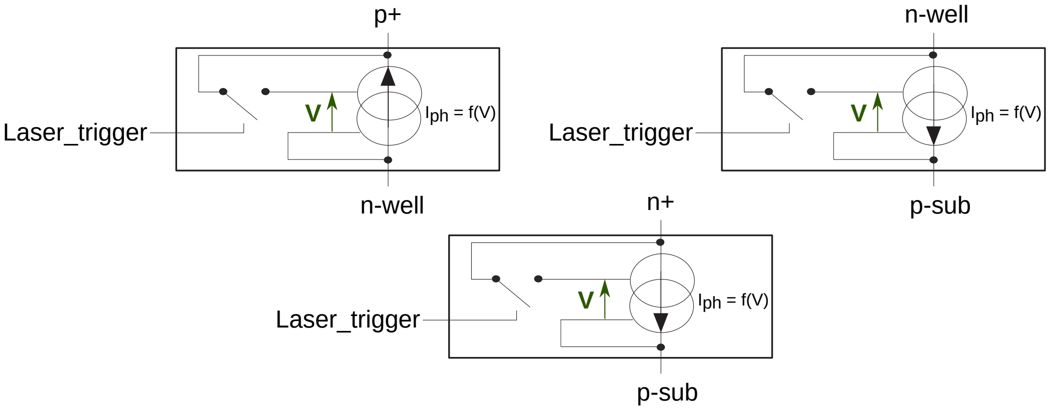


Figure 2.11: Voltage-controlled current sources representing photocurrent induced in certain PN junctions (see Equation 2.5) as used in SPICE models [104, 105, 107]. `Laser_trigger` signal is used to turn the laser in the simulation environment ON

The parameter s is speciffic for PMOS and NMOS. The parameter ρ is used to take into account the distance between the PN junction and the laser spot, as expressed in the following equation:

$$\rho = \beta \cdot exp(-\frac{d^2}{c_1}) + \gamma \cdot exp(-\frac{d^2}{c_2}), \tag{2.8}$$

where β and γ are the fitting parameters [107] and c_1 and c_2 express the influence of optical lens – a part of the measurement setup influencing the illuminated area and the energy density per square area.

Note that the equations above contain parameters specific for each PN junction: the *p-sub/n-well* junction uses different parameters to express the photocurrent than *p+/n-well* or *p-sub/n+*. The parameter values are reported in the referenced papers and are included in our models available online [20].

The equations 2.5, 2.6 and 2.7 disclose that the induced photocurrent is proportional to the PN junction areas – to the transistor size, to the reversed bias voltage, and to the energy density represented by the laser power.

In addition to the OBIC generated in each PN junction, the *parasitic bipolar* transistor might be activated. The NPN parasitic transistor in NMOS is activated when the transistor body voltage is increased enough. The simple bipolar model for NMOS transistor [106, 107] is in Figure 2.12. Similarly, the PNP parasitic bipolar transistor may be activated in PMOS. For lower laser powers, the PNP parasitic bipolar transistor was not observed [106] and thus we do not include it in our models [20].

When the described PN-junction models are combined and complemented by the MOS transistor model, the current dependency on the laser power may be obtained for both PMOS and NMOS – see Figure 2.13. The dependency of the laser-induced current on the laser power is analogous to the dependency of the subthreshold leakage on temperature,

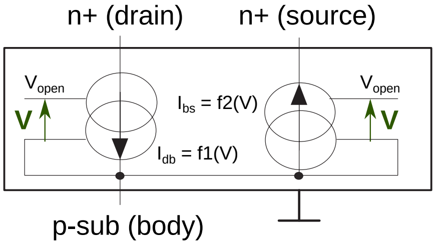


Figure 2.12: Parasitic bipolar transistor as modelled in SPICE model [105, 107]. `f1` and `f2` were fitted according to the measurements by Sarafianos et al. [105, 107]

while the induced current is several orders of magnitude higher – compare Figure 2.13 to Figures 2.7 and 2.5.

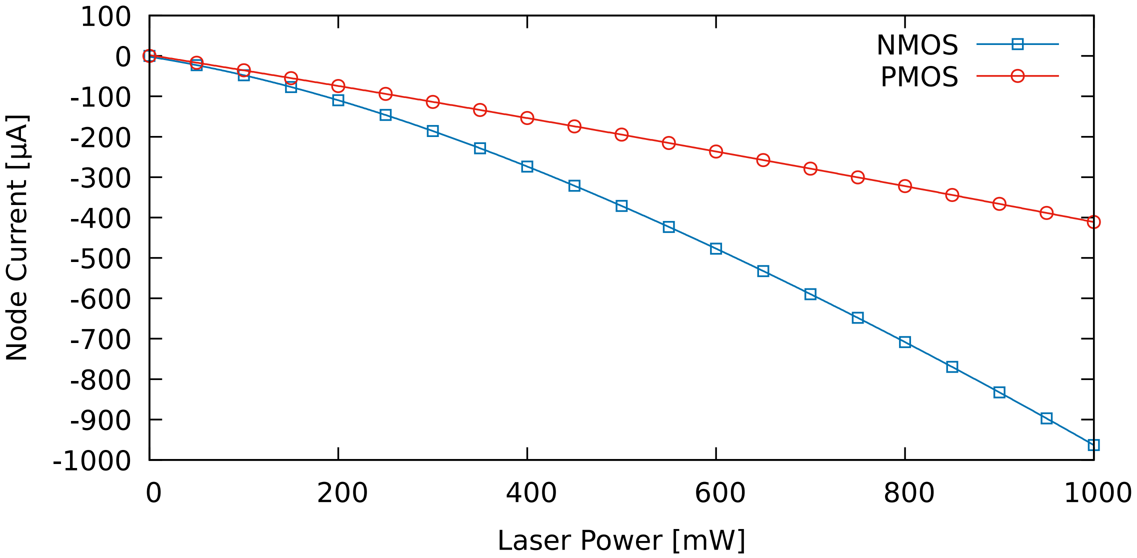


Figure 2.13: Optically induced current dependency on the illumination power for a single closed NMOS and PMOS transistors in TSMC180nm (NMOS = $0.2 \times 2.0\ \mu m$, PMOS = $0.2 \times 4.0\ \mu m$)

2.3 Digital Circuit Diagnostics

Circuit diagnostics aims at faulty parts detection. A *fault* in a digital circuit is the consequence of a *physical defect* or *enviromental effect* causing an unwanted *logic level* behavior possibly leading to *error*: the unintended change of the circuit output or the device state.

We can divide the error-correcting and detecting methods by the impact to the device performance to *online* and *offline* methods. Online methods do not affect the device latency significantly, while offline methods suspend the device.

Tests may deal with faults and/or errors. The offline tests – both manufacturing tests or in-the-field – commonly deal with faults (and output errors). The online tests commonly deal with errors only, as they are conventionally used to identify (and possibly induce a reaction to) errors.

The devices intended for operation in high-risk environments or in critical systems, such as flight control or Anti-lock Brake System (ABS), must be able to deal with errors arising during the device operation. For diagnostics in such systems, different approaches may be used depending on the device's needs. The in-the-field offline test could be performed periodically while the device is not loaded, if it is in the maintenance mode, or temporarily shut down. The other option is to employ the in-the-field *online test*. The online test is performed without the interruption of the device's normal operation, in the background.

The erroneous device output is caused by a fault at the physical level. From this perspective, both offline and online methods can detect fault presence. The main aspects determining the suitability of a given method are *fault coverage*, *error coverage*, *latency*, and area, delay, and power overhead.

In the literature, faults are often categorized according to their duration [70]. The faults, which occur and disappear are denoted as the *transient faults*. When a transient fault disappears, the function of the affected device is fully restored. The faults affecting the device permanently are denoted as *permanent faults*. These are often caused by physical defects [70].

A different fault classification is based on the fault nature. The *stuck-at faults* and *bridging faults* are those caused by a short; *open faults* are caused by a wire interruption and *delay faults* may be caused by transistor ageing, voltage drop, or by process variations (capacity or resistance). A high-energy particle may cause a *single-event-upset* (SEU) or a bit-flip.

2.3.1 Physical Defects and Fault Detection

The traditional approach for permanent fault detection in digital circuit diagnostics employs the *Automated Test Pattern Generator* (ATPG) to generate a huge amount of *test vectors* used for a complete circuit test [49]. A test vector is used to excite a fault and to propagate the possible *fault symptom* to the circuit's primary outputs, where it can be observed. If the fault symptom is observed at the circuit's primary output, the fault in the circuit is detected (and partially localized). Such a test is conventionally applied right after device manufacturing before packaging to discover defective parts as soon as possible.

The traditional yet widely used digital circuit testing approach is based on *fault simulation* using *fault models*. The fault simulation is a technique employing the fault model to derive the excitation input vector – the *test vector* – and its corresponding output set under the given fault. The test vector job is to excite a fault and propagate the fault symptom to the circuit outputs while ensuring that the fault symptom will be distinguishable.

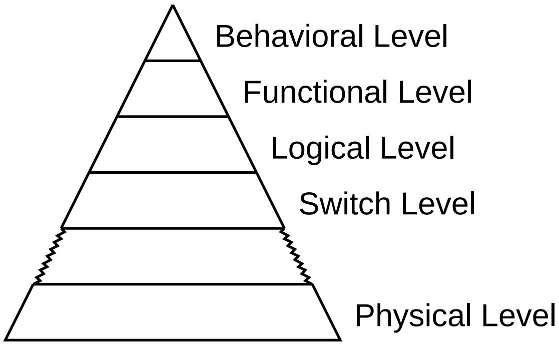


Figure 2.14: Circuit model abstraction level hierarchy

The fault simulation may operate on the several levels of abstraction [13] – see Figure 2.14. The ability to represent a real CMOS *defect* defects differs level-by-level [57, 60].

It has been shown that most defects are – at the *physical level* – caused by bridging faults [45, 76]. It is relatively hard to model bridging faults, thus the time and input invariant *logical level* models are widely used [2]. In both academy [70] and industry [13], the logical level *stuck-at fault* model or the *switch level stuck-open/stuck-on fault* models are widely used.

The mentioned models are preferred due to their simplicity [2, 13]. Both models are conventionally used to model a single fault case. The reason behind considering a single fault only is natural, as if more than a single fault are considered, the simulation, the test time, and mainly the test generation time rise. Here testing becomes impractical and unusable in practice.

Naturally, the test generation process employs methods for making the test set compact. The natural property is that the excitation vector may *cover* multiple modeled (single) faults at the same time. Another method employs fault collapsing: a fault *dominate* over the other fault when the test vectors of the dominating fault are the superset of the test vectors detecting the other fault. The dominant fault is detected by *implication*. These facts lead to the natural reduction of the test length. The test quality metric is the fault coverage expressing the share of modeled faults covered by the test:

$$c_f = \frac{\#\ faults\ detected}{\#\ faults\ modeled} \tag{2.9}$$

A principal problem of the current approaches is that a simplified expectation that only a single fault may occur, may lead to uncovered real circuit defects. The problem is that in reality, more faults might be excited by using a single vector, while their fault symptoms can eliminate each other while propagating to the circuit inputs. Although the single fault models oversimplify the true nature of the faults in today's CMOS processes, it has been shown that if the single stuck-at fault coverage is high, the case where the circuit is marked as fault-free while it is faulty is rare [74] but natural due to model inaccuracy [57].

2.3.2 Fault Models

Stuck-At-Fault Model

The stuck-at-fault model is a *gate-level* model. It comes from the deep history of digital systems when the main source of faults was the interconnection between logic gates [2]. This model considers two types of permanent faults – permanent logic one (s@1) and permanent logic zero (s@0) at the gate input or output. Today, the stuck-at-fault coverage is still widely used as a metric for the test quality even in industry [13].

It has been shown that the stuck-at-fault model may be used to detect some of the bridging faults [77, 81], and that a high stuck-at-fault coverage implies high bridging fault coverage [74].

Stuck-Open/Stuck-On Model

A more detailed model is the stuck-open/stuck-on fault model. It is a *transistor-level* model. It considers two types of faults – one corresponds to a permanently open transistor and the second corresponds to a permanently closed transistor. This model may also be defined as an extension of the stuck-at-fault model, where s@1 and s@0 are considered at every transistor gate [2, 18]. Thus, the set of modeled faults includes all faults from the stuck-at-fault, while additional faults are modeled [2].

Other Models

The most commonly used fault model for the bridging faults is the Wired-AND/Wired-OR fault model [2]. A single fault is represented as the AND/OR logic function. As the Wired-AND/Wired-OR fault model [77] does not reflect the behavior of all types of bridging faults, several models reflecting this behavior have been proposed [40, 74].

In [2], the voting model related to the *Byzantine generals problem* for bridging faults has been presented. It is based on the transistor-level comparison of pull-up and pull-down path conductivity.

Transient and Intermittent Faults

For the offline methods, the most problematic faults are those appearing randomly. A fault which occurs and disappears is denoted as the *transient fault* (sometimes called *soft-error*). When a transient fault disappears, the function of the affected device is fully restored. Some authors also distinguish *short-duration transient* and *long-duration transient* faults [99].

An example of a transient fault is a change in a memory cell caused by electromagnetic interference. Sometimes, the transient fault removal may require device reset or re-initialization; correction of a bit-flip in the configuration memory of an FPGA device may require the FPGA reconfiguration [16].

Sometimes, *intermittent faults* are mentioned. The intermittent fault never goes away entirely, but it sometimes affects the system function and sometimes is hidden. The intermittent faults are commonly connected with defects and they often tend to become permanent [70].

Delay Faults

Some defects do not change the logic function of the circuit but affect its timing. Some of the delay defects are covered by time-invariant models, while many delay faults are not [57, 60].

2.3.3 Error Detection And Correction

In applications where *error resilience* is required, some kind of *redundancy* has to be involved to enable in-the-field test. In most cases, the *time* (temporal) or *area* (spatial) redundancy is considered.

The quality metric for error detection and correction method is the error coverage:

$$c_e = \frac{\#\ errors\ detected}{\#\ errors\ occured} \tag{2.10}$$

The full error detection is conditioned by the *self-checking* property. The system is *self-checking* if an occurrence of a fault leads to a faulty output. The important sub-class of the self-checking circuits set are the *Totally Self-Checking* circuits (TSC). The TSC property means that any fault in the circuit may not cause an undetectable faulty output [95]. Thus, any architecture offering full error detection must be TSC to provide full *error coverage.*

The problem with the error coverage is that it is not possible to compute error coverage from the fault coverage (see equation (2.9)), as any fault model is a simplification and it does not reflect all defects and their consequences. It has been shown that if the fault coverage is high and the test passes, the device is most probably not defective [74]. However, using an offline test for error detection or correction is principally limited by the discrepancy between equations (2.9) and (2.10), and thus it cannot be employed in safety-critical systems.

Offline testing can be used to correct errors only if the test has significant and realistic *fault coverage.* If the offline test passes, the output of the device may be correct or not, depending on the test coverage and the fault model accuracy. On the other side, if the test does not pass, it is clear that for the set of input vectors, the device produces an erroneous output (but it can still produce correct outputs for another set of input vectors).

Offline testing can still be employed to detect faults appearing during the device mission. The in-the-field offline test is represented by the *Built-In-Self-Test* (BIST) approach. BIST can be executed during device startup or periodically. BIST is characterized by both spatial and temporal redundancy. The BIST approach is widely used for startup diagnostics.

However, BIST can only be used if the device alternates between operational states and states when it is not loaded: the maintenance mode, or temporarily shut down. If an uninterrupted operation is required, offline testing cannot be used. Additionally, offline testing is proactive; it can be used to identify only long-duration transient or permanent faults before entering the functional mode, not during the computation [70]. Short-duration transient faults are not covered along with hard-to-catch defects [57].

On the other hand, online testing can be used to detect or correct errors reactively during the computation without interruption for a cost of significant area and/or time redundancy.

The redundancy employed by the online test serves to detect or tolerate malfunctions in computational units, storages, or communication channels. Different types of redundancy may be employed, such as information (error detecting and correcting codes), software (N-version programming), *time* (recomputation, offline test), or *area* (concurrent computation in independent units – e.g. N-modular redundancy). Involving any type of redundancy brings additional design, manufacturing, and operating costs [59], thus the balance between costs and benefits has to be targeted. From the hardware point of view, all types of redundancy affect the *time* (latency and throughput) and/or the *area* domain.

From the *physical fault* point of view, the online area redundancy-based methods are well suitable for mitigation of errors caused by both *transient* and *permanent* faults. Computation repetition (i.e., time redundancy) can be efficiently used for mitigating errors caused by *transient* faults.

2.3.4 Area Redundancy Overview

A well-adopted approach employing the area redundancy for error detection and correction is the *N-modular redundancy* (NMR). The redundancy of NMR is in the area domain and the error detection (and correction) is performed online, while only a small delay penalty is caused by the voter circuit.

The simplest way to achieve *online error detection* is by duplicating the original module and thus creating the *duplex.* The joint output of two identical modules allows to distinguish a correct (outputs match – both outputs are correct) and an erroneous output (outputs are different – one of the outputs is faulty) in case when at least one of the duplex parts is fault-free, the TSC property holds. Duplex is also the simplest example of a self-checking system.

In the NMR family, the *online error correction* can be achieved by (at least) triplicating the original module. This is called a *Triple modular redundancy (TMR)* – see Figure 2.15. TMR can produce correct output if at least two of three identical modules are fault-free.

From another point of view, self-checking (error detecting) modules, in general, can be used to construct an error-correcting system – see Figure 2.16. A simple example is the *bi-duplex system.* It is an error-correcting system consisted of two self-checking duplex

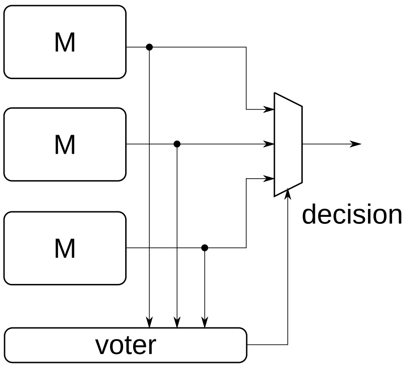


Figure 2.15: Conceptual scheme of an error-correcting TMR

modules [70]. The disadvantage of self-checking circuits is their size. The self-checking circuit size is typically close to the size of the duplex [70, 102].

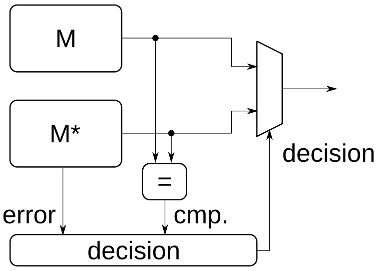


Figure 2.16: Conceptual scheme of an error-correcting duplex system with a self-checking module `M*`

The duplex system is not the only example of a self-checking circuit. Self-checking circuits are the subject of deep research in the area of asynchronous circuits [112]. The traditional asynchronous approach is based on *dual-rail* logic [29, 30, 102].

2.3.5 Time Redundancy Overview

Time domain methods can be used to eliminate programming errors and transient faults. The *N-version programming* offers immunity to errors caused by developers. Here the code – software or hardware description and test – is developed N-times by different teams. The software can be executed simultaneously in production, while the distinct hardware descriptions and tests are used for verification. It is supposed that the different teams do not make the same mistakes [26]

The straightforward *computation repetition* offers immunity to transient faults (some of them) but it is unable to overcome permanent faults, while specialized techniques employing coders might also detect permanent faults [86]. If the computation is repeated several times, both error detection and correction may be provided [70].

2.3.6 Time and Area Redundancy Combinations Overview

The online methods increase the costs (mainly) in the area domain and the offline methods in the time domain. Sometimes, it may be advantageous to combine both approaches. As applying the offline methods disables the system function for some time, it is efficient to use an online method for error detection and when an error is detected, then use an offline method for error correction. Thus, the online error detection is a trigger for the offline error correction.

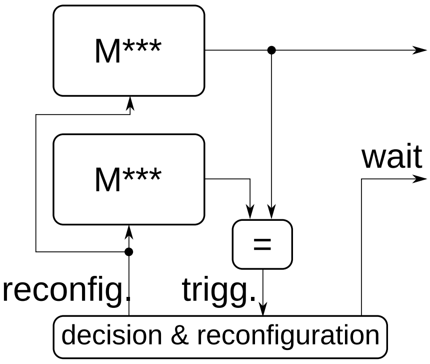


Figure 2.17: Conceptual scheme of an error-correcting duplex employing the reconfigurable modules M***

Works combining time and area redundancy often deal only with transient or *soft faults* like *single-event upsets* (SEU), e.g., [103]. The usability of presented solutions where not only transient faults are considered is problematical due to a delayed fault detection [14, 68]. The following paragraphs contain a list of approaches combining time and area redundancy for error correction.

The approach proposed for handling *bit-flips* in the configuration memory of FPGA devices uses a kind of duplex system to detect errors online and then reconfiguration is performed to repair the faulty parts [16, 43] – see Figure 2.17.

The approach presented in [27] relies on parts, which are not backed up and are considered to be reliable enough, while the unreliable part of the system is reconfigurable and thus allows fault recovery. The system was designed as radiation tolerant, and the approach is efficient for the given application. But this approach deals only with transient faults in the reprogrammable part. Additionally, it is not general and the system contains parts which may be denoted as a single point of failure.

The approach employing backup units may be used for periodical checking of the functional unit. In case of a fault, the functional unit is replaced by a backup unit [14, 68]. The significant disadvantages of this approach are that the fault detection is delayed significantly and that the unit output is not checked in every cycle. Some of the input vectors may disclose present faults but other vectors may produce correct results. Thus, the fault needs not be identified while the incorrect output is produced. This is why the practical usage of this approach is problematic.

To introduce some level of reliability into high-performance chips, the problem with an additional delay caused by checkers was studied years ago [118]. Although fast checkers are used, some delay is still introduced and additionally, the area overhead caused by high-performance checkers is large. To allow to use of lower performance checkers and mask the introduced delay, pipeline *micro rollback* was introduced in [118]. Similar approaches were presented later, e.g. in [11, 23, 129]. The presented pipeline rollback-based approaches are suitable for handling soft-errors.

The approach called *dynamic implementation verification architecture* (DIVA) presented in [11] is similar to the pipeline rollback. It is based on the concatenation of two pipelines. The first pipeline is more complicated and performs a speculative computation. It is implemented to be as fast as possible, and thus it is less reliable. The second pipeline checks the results of the first pipeline. Because in the second pipeline there are no slow inter-instruction dependencies, it is fast enough, although it is implemented in a robust technology.

Chapter 3

Design for Testability

> *Combinational logic optimized for function and fault symptom excitation and propagation at the same time ...*
>
> *Testers' dream*

Conventional design-for-testability methods resolve the problem of deep integration, making parts of the circuit accessible to the tester. Scan-based methods or even BIST allow to access the IC internal memory and integrated block interfaces to apply conventional tests. Our contribution goes a further step behind the traditional concepts: the proposed method of the combinational logic design allows replacing the long structural test by its short-duration alternative.

3.1 Motivation

Testing fights with two challenges: the fault (i) *controlability* and the fault (ii) *observability.* Fault controllability is an ability to *excite* the fault to show up – to produce a *fault symptom.* And observability is an ability to *propagate* the fault symptom to the primary outputs, where it can be observed. The problem is that the fault excitation and symptom propagation must be performed by the same test vector (or by a short sequence of test vectors) leading to huge test sets to gain a high fault coverage. Another problem is that combinational circuits are incorporated in larger systems of a sequential nature. Testing of sequential circuits is impractical even for small circuits [132]. The current DFT (*Design-For-Testability*) methods enable system partitioning and allow access to the state of the sequential circuit to apply a test to the combinational part of the circuit. Conventionally, the test vector must be *transported* to the circuit inputs and the test response from the circuit outputs for checking [132]. Alternatively, the test is in part (or completely) realized on-chip, while its nature and length remain: only the communication bottleneck between an IC and the tester is removed [24, 90].

3.2 Related Art

Up today, number of design-for-testability methods were developed. The DFT methods mostly differ in the way how the test vector (and the response) is delivered to (and out of) the device (or block) under-test (DUT) and how complex this process is. The test vectors may be compressed or even generated on the chip, while the responses may be compacted [24, 90]. Even in the case when the test logic is placed on-chip, the number of test vectors required to achieve high fault coverage remains high [90], creating a principal bottleneck. Most of the current DFT methods are based on *test-per-clock* or *test-per-scan* or their combination [90]. Another approach is based on fault current monitoring – an increased current consumption might serve as a fault symptom: the manifestation of a fault.

3.2.1 Test-Per-Clock

The test-per-clock approach employs the circuits' *test mode.* In the test mode, all combinational logic inputs and outputs become drivable and observable, as they are connected to the test equipment: flip-flop outputs are disconnected and the circuit is driven by pseudo-primary inputs, while the flip-flop inputs are made observable as pseudo-primary outputs – see Figure 3.1. The disadvantage of this approach is that the number of inputs and outputs of the particular block is increased by the number of flip-flops, thus this approach is vital only for on-chip usage, typically for the Buil-In-Self-Test (BIST) [24, 90].

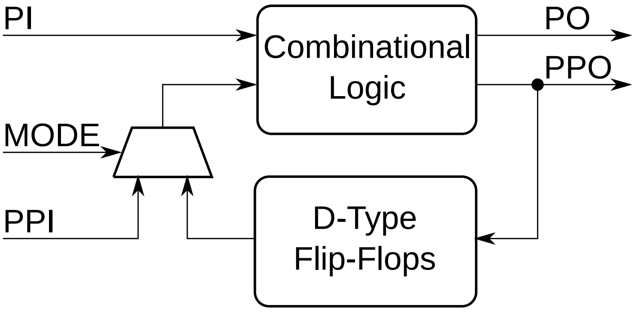


Figure 3.1: In the test-per-clock approach, the circuit input set (PI) is extended by pseudo-primary inputs (PPI), while the output set (PO) is extended by pseudo-primary outputs (PO)

3.2.2 Test-Per-Scan

The test-per-scan approach is a mature solution and the *design-for-testability* method based on *scan chains*, first introduced in 1972 [132]. The test-per-scan approach also employs a dedicated test mode, but in contrary to the test-per-clock, the memory elements are not isolated, but replaced by a shift register allowing to change and observe the current state of the circuits' memory – see Figure 3.2. Test-per-scan does not lead to a significant increase in the number of the circuit (or block) inputs and outputs. On the other hand, the test-per-scan increases the test time, as testing one vector requires a number of clock cycles to shift-in the test stimuli, and shift-out the response. Another disadvantage is the increased (and abnormal) switching activity leading to the DUT overheating [101, 128].

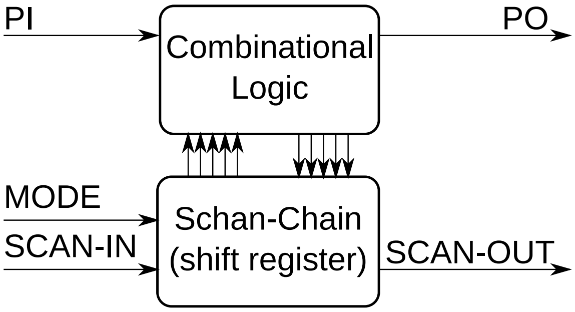


Figure 3.2: In the test-per-scan approach, only two additional inputs and one output is required to insert a test vector and get the response, while the test time is increased

The test-per-scan method can be combined with the *boundary-scan* [62] to access even the circuits' primary inputs in an analogous way to the internal memory elements – see Figure 3.3.

Commonly, the scan-chains or boundary-scan cells inserted into design add excitation, as well as observation points, and extend the number of accessible signals in the *test mode*.

3.2.3 Fault Current Monitoring

Fault current monitoring is a method commonly employed by industry [13]. It has been shown, that combining time-invariant fault models with the fault current monitoring increases the test coverage significantly [57]. Conventionally, *fault-current* monitoring[1] is used in connection with traditional test approaches [57]. The increased power consumption of the device-under-test serves as a secondary fault symptom. The combination of different fault symptoms allows to cover hard-to-detect faults and/or shortening of the test time.

Conventionally, the *fault-current* is monitored externally [127], [131], but in the past years, much work has been done also in the Built-In Current Sensors (BICS) area, starting

[1] Fault-current monitoring is also referred to as *quiescent supply current monitoring* or *IDDQ testing*.

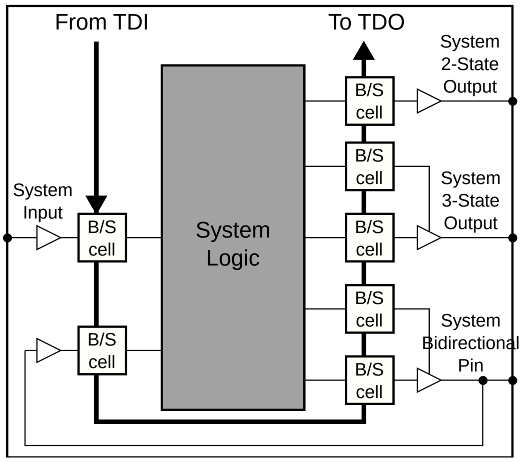


Figure 3.3: Boundary-scan cell placement (from [62]) – boudary-scan makes the logic of the System-on-Chip core or the System-on-Board component accessible for tester

from [10] in 1996 where the first BICS for deep sub-micron technologies has been presented. Recently, BICSs were proposed also for transient faults detection [94].

One BICS can monitor only a limited number of power rails due to a limited resolution and current load capacity. This implies using more parallel BICSs for the whole circuit [10].

The current example of the test methodology employing the fault symptom combination is represented by the AEC-Q100 standard defining the manufacturing test requirements for the automotive industry. The AEC-Q100 states [13] that in the production test, 97% stuck-at fault coverage is acceptable when combined with IDDQ, otherwise 98% stuck-at fault coverage is required. The trade-off is up to the circuit and test designers.

3.3 Short-Duration Offline Test

This section represents a contribution of this dissertation thesis to the combinational circuit testability field. To circumvent the problem of expensive test vectors generation and storing in memory, we propose a circuit design method, for which test vectors and test responses are easy to produce and check in hardware, while the test length is in orders of tens computational (clock) cycles only. We call such a test a *short-duration test*. If the fault coverage is 100% with respect to the given fault model, we call the test a *complete short-duration test*.

In [A.3, A.4, A.1], we proposed the circuit design method enabling the *short-duration test*. Our short-duration offline test works in such a way that in a fault-free circuit, an all-zero output is a response to the all-zero input and an all-one output is a response for the all-one input. If there is a fault in the circuit, the opposite logic value is propagated from the fault location up to the circuit outputs. In other words: the circuit is flooded by zeroes and subsequently by ones and any fault blocks the value propagation from the circuit inputs to the outputs. Such test vectors and test responses are easy to produce and check in hardware and the test controller is a simple state machine.

Naturally, such a test requires a paradigm shift in the CMOS circuit design and our method belongs to *design-for-testability* methods. In [A.3], we have shown that *monotonicity* is required for a short-duration offline test, as it ensures that the fault symptoms are not flipped (one to zero or zero to one) during the propagation to the circuit outputs, and thus simplifies the overall test.

Monotonic circuits are circuits containing no inverters. The $0 \to 1$ change at the circuit inputs may only cause $0 \to 1$ changes in the circuit and the $1 \to 0$ change at the circuit inputs may only cause $1 \to 0$ changes in the circuit. An example of a monotonic circuit is the dual-rail (`AND`/`OR` based) circuit.

3.3.1 Stuck-At-Fault Symptom Propagation

If every internal gate output in a monotonic circuit is connected to at least one OR gate and to at least one AND gate, the test with 100% fault coverage with respect to the stuck-at-fault model requires only two test vectors. These are *all-zero* and *all-one* vectors. It is intuitively clear that such two-vector-testable circuits are rare.

3.3.2 Conventional Monotonic Design

The simplest solution to transform any logic function to a monotonic function is to adopt a fully monotonic logic design, the *dual-rail* logic [112, 131]. In dual-rail logic, an inverter is represented as a wire-swap only and every signal is represented by a value on the complementary wires. Dual-rail encoding is conventionally used for completion detection [112]. Unfortunately, the area of this approach is approximately double compared to the original single-rail circuit.

3.3.3 Isolated Monotonic Logic Blocks

In our case, we require monotonicity for testing, thus we do not require full dual-rail implementation. To reduce the area (and power), we allow inverters (single-rail to dual-rail converters) at the primary inputs of the `M**` module. This allows to move from the dual-rail design to a structure we call *isolated monotonic* combinational logic blocks – see Figure 3.4.

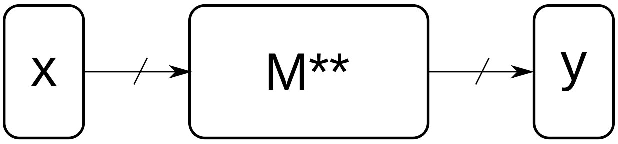


Figure 3.4: The isolated monotonic logic block (`M**`) contains AND/OR gates only – it is short-duration test ready. The arrays of inverters at its inputs/outputs are denoted `x` and `y`

In an isolated monotonic logic block, inverters are placed at the input or output of the monotonic logic block. When the monotonic logic is a dual-rail circuit, the input inverters transform the single-rail signals to dual-rail signals without disrupting the monotonicity of the isolated block – see blocks `x` , `y` and `M**` in Figure 3.4.

For our method, a single-rail output of the module `M**` is sufficient, thus only those internal signals should remain, which are required to compute the single-rail output. Therefore, we can remove half of the dual-rail circuit outputs from the dual-rail implementation (only the positive outputs remain). Circuit parts feeding only the removed outputs should also be removed – see Figure 3.6. Then the dual signals (originating from the dual-rail implementation) serve as inverters replacements only. The number of outputs of such a circuit is equal to the number of outputs of the original single-rail circuit (module denoted `M`), and the number of the `M**` inputs varies between once and twice the number of the `M` inputs.

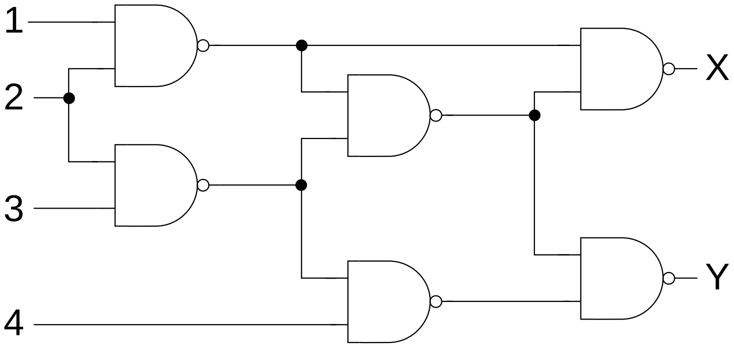


Figure 3.5: NAND-based circuit example

`NAND`-based circuit example

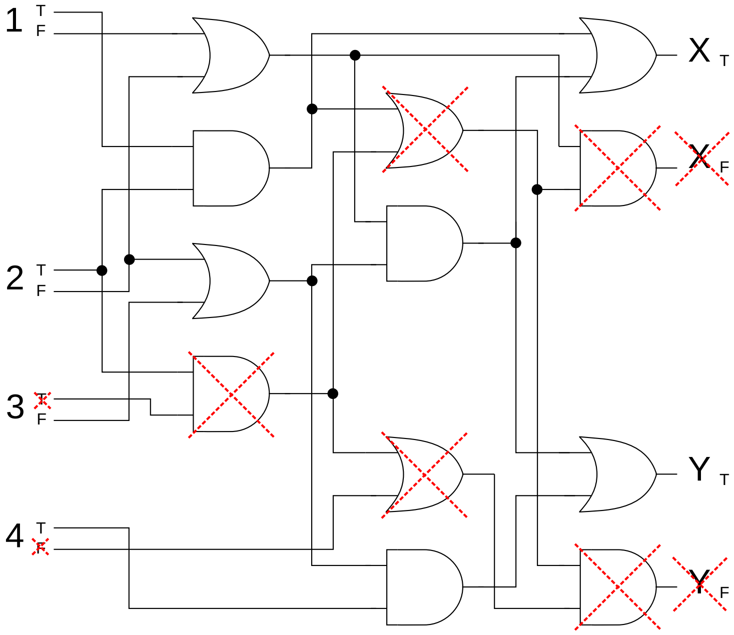


Figure 3.6: Dual-rail logic circuit derived from the circuit in Figure 3.5 – every NAND gate was replaced by an AND and OR gate pair. The crossed-out gates, inputs, and outputs are removed by the reduction (M**)

After reduction, the resulting circuit in Figure 3.6 is smaller, but it still has more gates than the original single-rail one. The number of outputs is the same, and the number of inputs is increased – it has 6 primary inputs instead of 4 in the original single-rail implementation in Figure 3.5 – both polarities of inputs 1 and 2 are required to compute outputs.

The reduction presented in [A.3] was evaluated on circuits from the *IWLS 2005* benchmark set [5]. Our results show that reduced circuits have about 60% of the area of the dual-rail circuits on average. The resulting area for all benchmark circuits was between 50% and 100% of the dual-rail circuit area. The extreme values were achieved for smaller circuits only. Large circuits were close to the average.

If inverted outputs are allowed – see the block of output inverters denoted y in Figure 3.4 – either polarity can be selected during reduction. Here, the reduction success depends not only on the circuit structure but also on the output polarity selection.

We developed five simple ways to achieve a high degree of reduction. The two simplest approaches take just the set of positive (as described above) or just the set of negative outputs. Another three approaches are greedy heuristics. All greedy heuristics start with the first output pair and continue with the other pairs. From each pair of the dual-rail circuit outputs, the output with a smaller additional cost is selected (e.g., selecting one polarity implies adding fewer gates than selecting the other polarity). The heuristics differ just in the cost function. The cost functions are the number of gates, circuit size, and delay.

We applied all the developed approaches to benchmark circuits. When the best result for every circuit was selected, we achieved only 3% area improvement on average compared to the approach taking just the set of positive outputs.

For the set of benchmark circuits, we additionally compared the heuristics with results of the *Monte-Carlo* method taking random output selections. We achieved no improvement compared to the best result given by one of the heuristics. Thus it can be concluded that all of the greedy heuristics give results close to optimum.

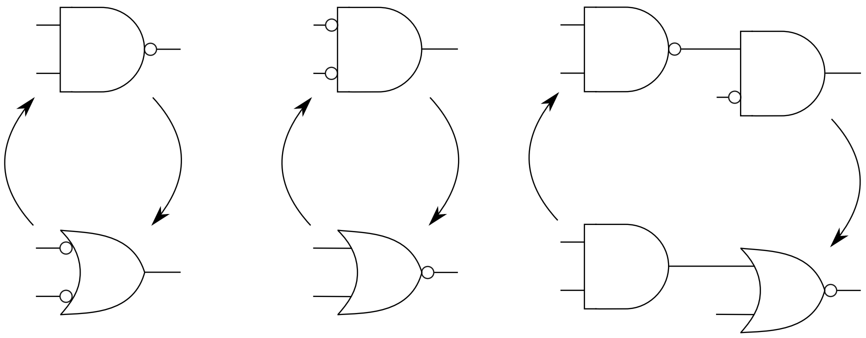


Figure 3.7: An example of gate transformations allowing to move inverters to the circuit primary inputs/outputs

The isolated monotonic logic blocks can also be created differently. It is possible to start from a single-rail circuit containing inverters and apply transformations preserving the logic function by moving inverters to circuit primary inputs/outputs, as shown in Figure 3.7. If some of the gate outputs are present in both forms – direct and inverted, the gate is duplicated – the first duplicate produces only the direct and the other only the inverted form. This heuristic produces also an internally monotonic circuit with inverters at circuit primary inputs/outputs and the number of circuit inputs is (usually) also greater than in the original single-rail circuit.

We performed a comparison of the heuristic we developed to perform the described transformations with the heuristics reducing the dual-rail circuits, and no improvement has been achieved. The heuristic transforming the single-rail circuit directly gave always worse, or (in the best case) the same results as the best heuristic used for the dual-rail reduction.

The overall algorithm for constructing the smallest isolated combinational logic block is thus the following: take the minimized single-rail circuit (Figure 3.5) and create its dual-rail equivalent (Figure 3.6). Then apply the proposed reduction heuristics (Figure 3.6) and select the best result.

3.3.4 Addressing Testability Limitations at the Cell Level

The monotonicity itself is not enough to ensure that all possible fault symptoms (one and zero) will be propagated up to the circuit's primary outputs. It ensures that these are without any change, but they may be still masked and thus not observable.

The short-duration test of `M**` requires a special gate design. The gate has to allow propagation of all possible fault symptoms (one and zero) up to the circuit's primary outputs without any change and with no masking. We propose a novel reconfigurable gate structure allowing propagation of both fault symptoms (zero and one), which is similar to the dynamic *domino logic* – see Section 2.1.3.

The proposed gate can be configured to: 1) propagate fault symptom one (like `OR` gate); 2) propagate fault symptom zero (like `AND` gate); 3) set its output to 1 or 0; 4) work as a one-bit capacitance-based memory.

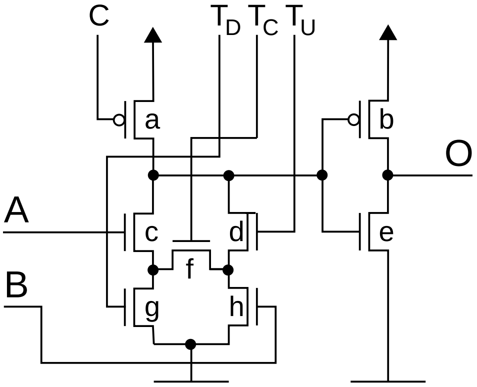


Figure 3.8: Proposed transistor-level structure

The proposed structure based on domino logic is shown in Figure 3.8 – this structure can realize both logic functions (AND/OR) depending on the control signals $\mathtt{T_U}$, $\mathtt{T_C}$ and $\mathtt{T_D}$. The proposed structure is still a domino logic gate. The novelty is in increased controllability of the gate, which is used for testability – during the test, the other functions of this structure are used.

As the described structure is domino logic-like (see Section 2.1.2), it operates in two phases: *precharge* and *evaluation*. The operation mode and the gate function (AND/OR) is set by control signals, as shown in Tables 3.1 and 3.2, where the output value is switched to `0` (↓) during precharge – depending only on the control (clock) signals. During evaluation, it preserves its value or is switched to `1` (↕) depending on both the gate inputs and control (clock) signals.

The additional *clock* signals are used for mode selection, during the test, and as the foot control. The load at these additional clock signals is significantly smaller than at the default domino logic clock signal because these signals control smaller NMOS transistors only. Additionally, for AND function, only $\mathtt{T_C}$ is switched during computation and $\mathtt{T_U}$ and $\mathtt{T_D}$ are permanently closed – the same applies for the OR gate function.

step	C	T_U	T_C	T_D	O
precharge	0	0	0	0	↓
evaluation	1	0	1	0	↕

Table 3.1: Control signals for AND gate function

step	C	T_U	T_C	T_D	O
precharge	0	0	0	0	↓
evaluation	1	1	0	1	↕

Table 3.2: Control signals for OR gate function

Other combinations of the control signals are used during the offline test to set specified signals to the desired value, to preserve a logic value for a small amount of time between few clock cycles, or to raise a fault symptom, when a specific fault is present in the gate. An example of other control signals combinations is setting all control signals to 1, which causes that the gate output is switched to 0; when all control signals are set to 0, then the output is switched to 1. A one-bit capacitance memory is realized by isolating the internal-node capacitance – T_C, T_U and T_D are set to 1 and C is set to 0.

The proposed structure allows performing a complete circuit test based on an accurate (but still time-invariant) transistor-level *stuck-open/stuck-on* fault model.

The offline test algorithm and the detailed TED design proposal are described in [A.1], while a detailed description with example is provided also in Appendix A.

3.4 Time-Extended Duplex

In Section 2.3.1, conventional fault detection and localization methods are described, while Section 2.3.3 deals with conventional error detection and correction methods. In this section, the new approach combining online error detection and offline fault localization is described. The presented approach, called *Time-Extended Duplex* (TED), allows designing a system with lower overhead than *Tripple-Modular Redundancy* (TMR) with comparable error recovery ability. In this dissertation thesis, only the high-level description of the TED is provided, for implementation details, please refer to [A.1].

The TED system is somehow similar to the TMR system – some of the TMR blocks are equivalent to parts of the TED system, and even the interface is very similar. The TED description provided in this section is partially based on comparison with the TMR system. The TMR system is shown in Figure 3.9.

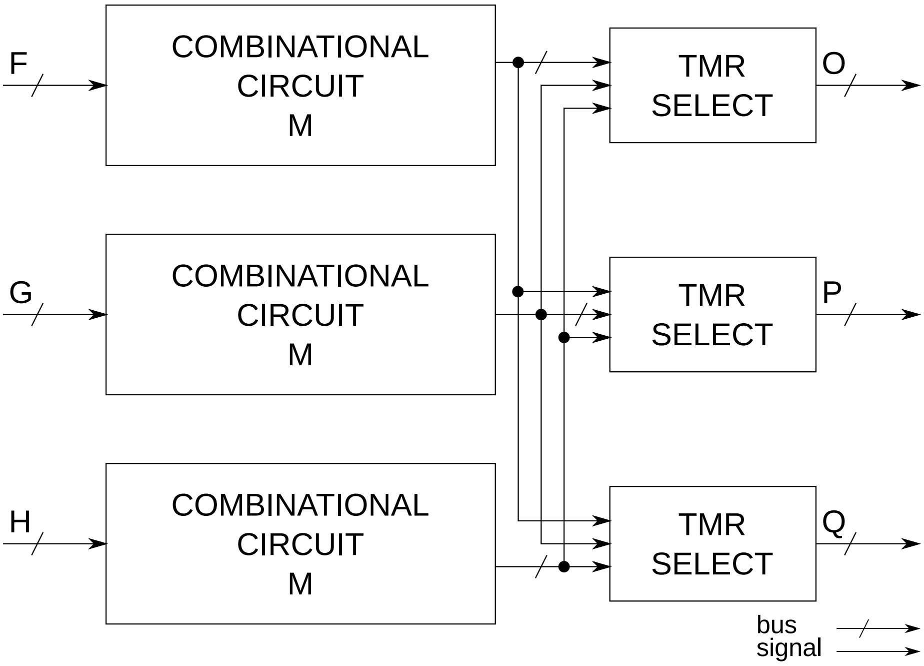


Figure 3.9: Detailed scheme of the TMR system

Similarly to the TMR system (Figure 3.9), the TED structure shown in Figure 3.10 processes three equivalent inputs (`I`, `J` and `K`) and offers three equivalent outputs (`OUTPUT SELECTs` produce `R`, `S` and `T`). However, it is composed of only two functionally equivalent combinational logic blocks (`M` and `M**`). This arrangement ensures: 1) that TED tolerates errors in a preceding logic by comparing three equivalent inputs; and 2) that the subsequent logic is able to select correct TED output in case of an error in the (triplicated) output logic.

The internal duplex arrangement allows error detection, not error masking. The error masking ability is allowed by the short-duration offline test. The offline test is triggered,

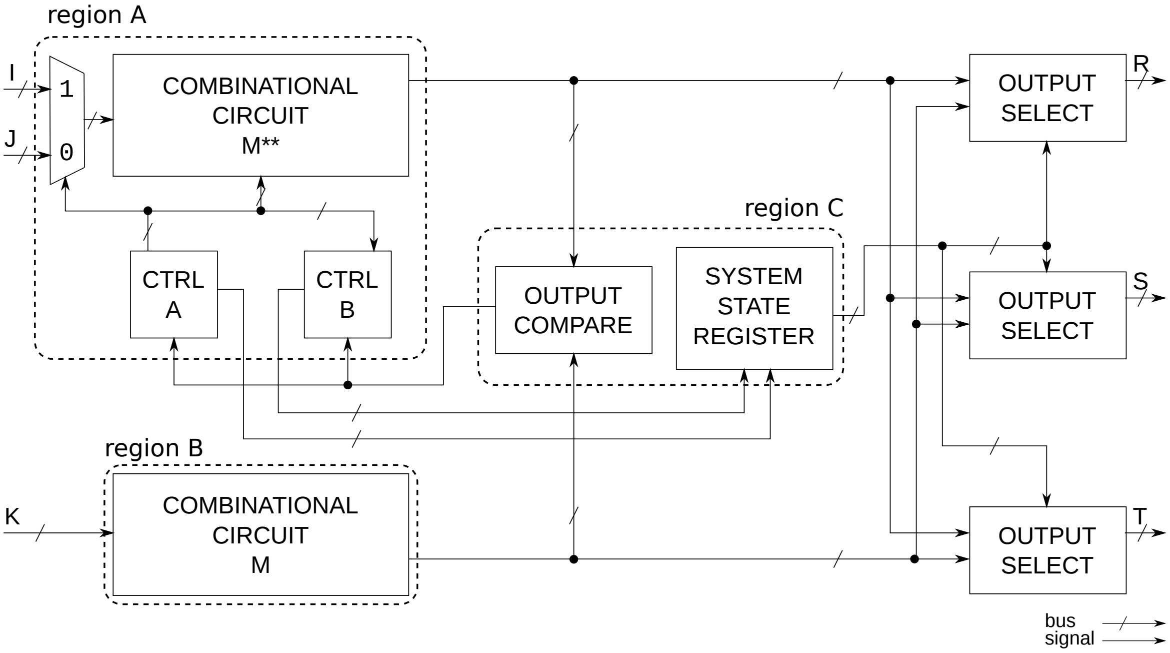


Figure 3.10: A high-level scheme of the Time-Extended Duplex

when the `OUTPUT COMPARE` block signalizes a mismatch of combinational logic outputs (`M` and `M**`). The test is able to detect any (modeled) permanent (or a long-duration transient) fault in `M**`. The rest of the logic in *region A* (test controller `CTRL` and input processing) is duplicated, thus error detection is ensured in the rest of the *region A* – see Figure 3.10.

If the offline test discloses a fault in `M**`, or a malfunction is detected (duplicated parts outputs are different) in the rest of the *region A*, the output of `M**` is assumed to be invalid and the output of `M` as valid. If no malfunction or fault is detected in *region A*, the output of `M**` is marked valid and the output of `M` as invalid. Note that an TED error caused by a fault located in *region C* cannot cause an erroneous output if both *region A* and *region B* are fault-free.

To be able to tolerate transient faults, which may also cause output mismatch, the TED uses the *recomputation*. A transient fault will trigger the offline test, but the offline test will be (with a high probability) not influenced by that fault. The offline test will always mark `M` as faulty independently of the transient fault location (because *region A* outputs are marked valid if no malfunction or fault is detected). Because – in case of the transient fault – it is not possible to state, which output is correct, the outputs must be recomputed after the offline test is performed. Unfortunately, it is not possible to distinguish between permanent and transient fault, thus the recomputation must be performed always.

To reduce a massive delay overhead introduced by a permanent fault causing a frequent mismatch in `M` and `M**` outputs, the test result memory represented by the `SYSTEM STATE REGISTER` is introduced. The `SYSTEM STATE REGISTER` holds the results of the last performed offline test. The content of the register is used for correct output selection, instead of performing the offline test (which is time-consuming). The offline test is performed only if the `SYSTEM STATE REGISTER` is empty.

Because transient faults may also cause output errors, the `SYSTEM STATE REGISTER` must be cleared periodically to recover from transient faults. The clearing period must be chosen to reflect the expected transient fault rate (the period must be much lower).

The arrangement with the `SYSTEM STATE REGISTER` ensures that the performance degradation is bounded by the `SYSTEM STATE REGISTER` clear period – this represents the worst case, as not every input of combinational logic necessarily reveals the actual permanent fault.

3.5 Discussion

In [A.1], we provide an area comparison of TED and TMR – the results of the comparison are provided in Figure 3.11. To provide a fair comparison, both TED and TMR are implemented as domino logic.

We have also defined an empirical pesimistic equation expressing the relation between the TED and TMR block sizes. If the following equation holds, TED is likely a more convenient from the area perspective, compared to TMR ($|\mathtt{A}|$ represents the area of $\mathtt{A}$):

$$\begin{gathered} if\ (\#inputs\ \approx\ \#outputs)\ and\ (\#outputs\ >\ 50):\\ 18 \cdot |\mathtt{TMR\ SELECT}|\ <\ |\mathtt{M}| \end{gathered} \tag{3.1}$$

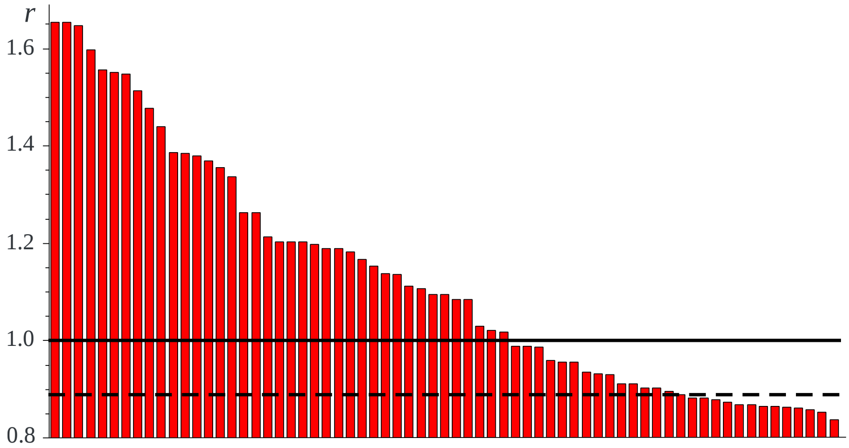


Figure 3.11: The comparison of the TED and TMR systems for 67 selected circuits (bigger benchmark circuits [5] with at least 50 IOs). The following ratio: $r = \frac{|\mathtt{TED}|}{|\mathtt{TMR}|}$ is shown. The circuits are shown descending ordered by r. The TED takes less area than TMR for circuits under the solid line. The equation (3.1) holds for circuits under the dashed line

The TED is a redundant combinational logic structure, however, it can be naturally used as a part of sequential logic. The usage of the TED is straightforward – the way to implement sequential logic using the TED is equal to using any other area-redundancy-based error-masking structure.

The only difference is that the output of the TED system may be delayed and thus the register write enable must be connected to the TED ready signal (TED signalizes the correct output).

The TED is comparable with the TMR in terms of delay and area only if: 1) the area overhead of the additional logic in TED is less than the area overhead caused by the third combinational logic module and 2) the delay introduced by the offline test is sufficiently small; 3) the offline test has high fault coverage.

The proposed test (see Appendix A) is unable to detect stuck-on faults at transistors ‘b’ and ‘e’ reliably (see Figure 3.8). The detectability of these faults depends on the fault nature. From the functional point of view, a fault causing an error at the gate output should be detectable by the presented tests. But in reality, it can behave as a transient fault if a short in the transistor causes that the output voltage is close to the next gate input threshold. Such a fault can cause errors on a random basis and may or may not be detected.

This can be solved by applying *fault-current* measurement. The Built-In Current Sensors (BICS) can be used to resolve this issue – see Section 3.2.3. One BICS can monitor only a limited number of power rails due to a limited resolution and current load capacity. This implies using more parallel BICSs for the whole circuit [10]. We propose to use BICSs just for fault detection at the output inverting stage of the proposed gate. Just one power rail has to be measured using BICS. Based on the previous sentences, this reduces the area overhead caused by using parallel BICSs. Additionally, the increased controllability of the circuit allows performing the required test by applying two test-vectors only – one vector to force the value `1` at the output of all gates and the second for the value `0`. The mentioned stuck-on faults are detectable using BICS at the end of *sub-test 2* and *sub-test 3*, therefore, no additional test cycles are required (although BICS tends to be slow and thus increase the test time).

As the used fault model does not fully reflect the bridging faults, it is advantageous to use BICS not only for uncovered stuck-on faults but also for the online detection of bridging faults located at the gate outputs.

Chapter 4

Side-Channel Attacks

> *Taste the cocktail and tell me the recipe ...*
>
> *Attackers' destiny*

Physical side-channel attacks, namely invasive, observation, and combined, represent a great challenge for today's digital design. A real CMOS circuit emits many symptoms connected to its configuration through so-called channels. The observation of the circuit emissions may compromise the circuit security while leaking the processed confidential data. Additionally, invasive methods allow stimulating side-channel emissions to bypass measurement limitations or implemented circuit protections. The contribution of this chapter is the analysis of the novel physical vulnerability of CMOS, and a description of the new combined attack potentially endangering most of today's CMOS designs.

4.1 Motivation

The real CMOS circuit may exhibit many distinct *configurations* (and configuration transitions). The configuration of a combinational circuit may be understood as a *state* of the circuit given namely by the circuit inputs and by its environment, and condition of its *elements*, like gates, transistors, interconnect, or the state of parasitics.

The circuit configuration (and configuration transitions) directly determines its *symptoms*: conventionaly circuit output(s), delay, and static power (leakage) or dynamic power, but also electromagnetic emissions (EMI), thermal emissions, and many other. In a bigger circuit, the symptoms produced by subcircuits are mixed together and *imprinted* to the *cocktail* of symptoms, which is *observable* through the so-called *channel*. The typical channels include the circuit *output set* – the *logical channel*, *power trace*, *EMI trace*, or the *delay record* – see Figure 4.1. The channels other than the logical channel are called *side-channels*, as their form is conventionally not dictated by the circuit designer, but it is implicated by the design and technology intersection as a *side-effect* [114].

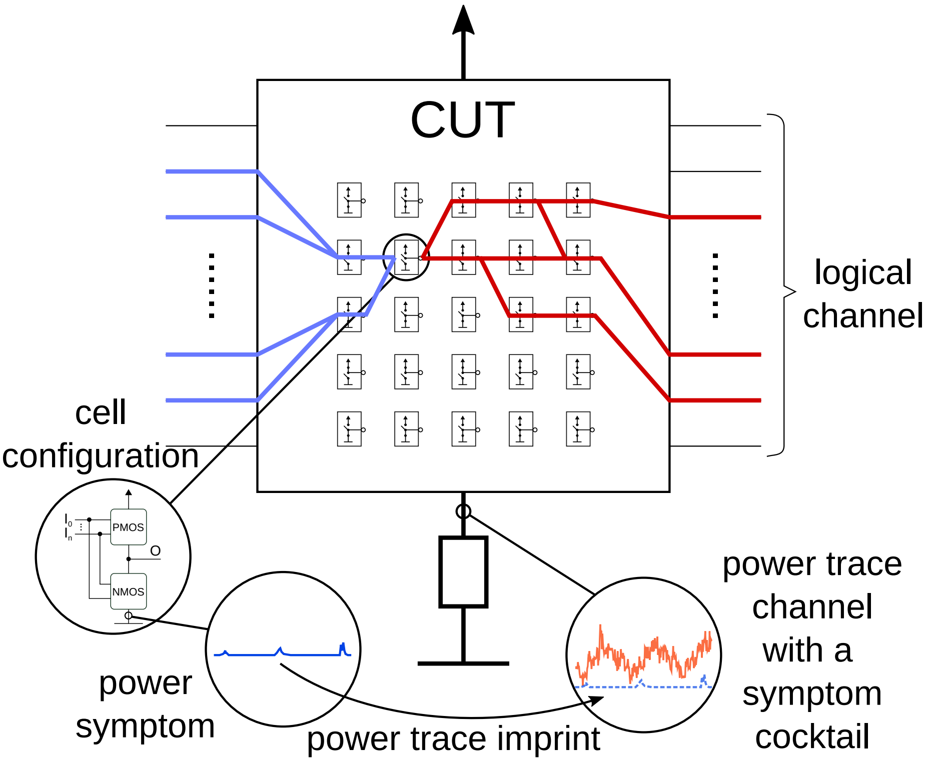


Figure 4.1: Illustration of how the symptom connected to a single cell imprints to the cocktail of symptoms observable through the power trace channel and how it is related to processed data

As the symptoms produced by the circuit into side-channels are influenced by processed data, the idea behind side-channel attacks is straightforward: deduce internal circuit variables by observing their symptoms in the side channels. The principle of the observation attack targeted on the power trace was first described by Kocher et al. in the late 1990s [93].

In this dissertation thesis, the physical vulnerability of CMOS is analyzed and a new combined attack potentially endangering most of today's CMOS designs is described. This work deals with observation power-related attacks, however, successful attack strategies exploiting other side channels were developed and are further studied [114, 137].

4.2 Related Art

This dissertation thesis provides an incremental contribution to combined side-channel power attacks. This section provides a brief overview of existing side-channel attack approaches and terminology. The physical side-channel attacks can be divided into observation [93], invasive [66], or combined [7].

The observation attacks are based only on the device activity and emissions *monitoring*. The most commonly used observation channels are power and EMI channels. The invasive attacks affect the device integrity, e.g. by decapsulation, and/or the device function, e.g. by inducing clock glitches, power spikes, or irradiating the device. Conventionally, the invasive attacks analyze the induced faulty outputs of the device to reveal the stored secret [66], while the so-called combined attacks analyze side-channels.

The following sections describe the matured power analysis techniques and conventional side-channel power channels.

4.2.1 Differential Power Analysis

The *differential power analysis* (DPA) was introduced by Kocher et al. [92, 93]. DPA was first demonstrated on a DES cipher, however, it is a common algorithm independent of particular hardware implementation under attack.

The DPA algorithm acts as follows:

1. The attacker observes m encryption operations, and captures power traces $T_{1..m}[1..k]$ with k samples each, and the attacker records encryption outputs (ciphertexts) $C_{1..m}$.

2. The attacker uses the knowledge of the encryption algorithm to define a *selection* function $I_i = D(C_i, K_g)$. The selection function defines the relation between the *n-bit* intermediate value I_i dependent on the secret value (key) guess K_g and a known ciphertext C_i.

3. For a given key guess K_g, the attacker computes a k-sample differential trace $\Delta_{1..k}$ for the selected bit (b) in the intermediate value I using all ciphertexts $C_{1..m}$. The Δ_j represents a difference of means of two subsets of the sample j over all power traces, distinguished by the selected bit in the intermediate value – see Figure 4.2.

4. The previous step is iterated for every key guess (and for several intermediate value bits).

The following equation holds, if the key guess is incorrect:

$$\lim_{m \to \infty} \Delta_j \approx 0 \tag{4.1}$$

The following equation holds, if the key guess is correct:

$$\lim_{m \to \infty} \Delta_j \approx i_b, \tag{4.2}$$

where i_b is the effect of the target bit (b) on the power consumption [93].
If the attacker can clearly distinguish i_b, the correct key guess is identified.

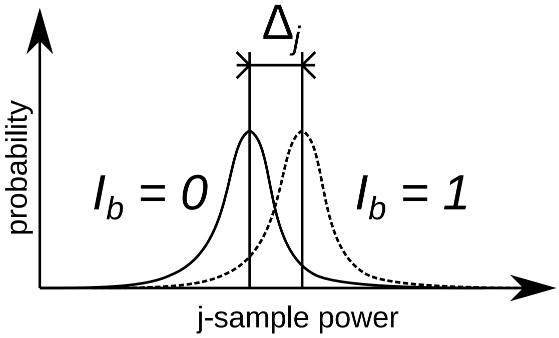


Figure 4.2: Two sample distributions distinguished by a correct key guess tend to have different means

4.2.2 Correlation Power Analysis

The *correlation power analysis* (CPA) studied e.g. in [17] is a bit newer statistical method employed in the side-channel analysis. CPA depends on the power model. In the literature, *Hamming weight* (HW) or *Hamming distance* (HD) power models are used [4, 17]. The models express the relationship between processed data and power consumption. The Hamming weight model expresses the power consumption as the number of asserted bits in the hidden variable, while the Hamming distance model captures the number of different bits between two consecutive values of the hidden variable.

The CPA algorithm acts as follows:

1. The attacker observes m encryption operations, and captures power traces $T_{1..m}[1..k]$ with k samples each, and the attacker records encryption outputs (ciphertexts) $C_{1..m}$.

2. The attacker uses the knowledge of the encryption algorithm to define a *selection* function $I_i = D(C_i, K_g)$. The selection function defines the relation between the n-bit intermediate value I_i dependent on the secret value (key) guess K_g and a known ciphertext C_i.

3. For all key guesses K_g and all ciphertexts C_i, the attacker computes the selection function D_{gi} and derives the power model: $H(D_{gi})$, which is a 2-D array representing (symbolicaly) a power consumption of the circuit for all key guesses and ciphertext.

4. Finally, the attacker computes correlation between the columns of $H(D_{gi})$ and $T_{1..m}[j]$ for every j: the most signifficant correlation indicates the leaking sample (j) and a correct key guess.

4.2.3 Dynamic Power Channel

Most of the successful side-channel attacks on physical devices in the past targeted dynamic power [114, 137]. Dynamic power data dependency is easier to measure and is straightforward to understand and model. As described in Section 2.1.4, the dynamic power depends on the parasitic capacitance being charged and discharged. If an internal variable in the circuit is changed – e.g. data on the microcontroller's bus change, a significant parasitic capacitance is being charged invoking the data-dependent current. This data-dependent current is additionally simple to model, as the charged parasitic capacitance is given by the number of bits changing their value.

Both DPA and CPA were originally developed for mounting attacks on dynamic power and both methods showed their usability in practical attacks [79, 80]. Dynamic power is often closely characterized by the output of the block, which is quite a realistic assumption when e.g. high-load bus wires are driven [4]. Thus the attacks on dynamic power are often driven by the output data of the target block.

There are several strategies to prevent attacks on dynamic power. A class of strategies is based on *hiding* by increasing the noise [65]. Another successful class of strategies adopted by industry allowing *hiding* data dependency of the side channel emissions in CMOS is based on balancing. Balancing aims to achieve a uniform and data-independent power consumption. Many techniques employing (parasitics) balancing were developed. An example of such a successful technique employing dual-rail complementary encoding is the conventional WDDL (*Wave Dynamic Differential Logic*) [121].

The dual-rail hiding approach is originally based on the dual-rail encoding extensively used in asynchronous circuits design [112]. In dual-rail, every signal is encoded on two wires and computation is divided into two alternating phases: *precharge* to the *spacer* and *evaluation*. The spacer is conventionally encoded as `00` and logic one and zero as `10` and `01`, respectively. The natural property of the dual-rail circuit is that there is a constant number of 0 to 1 (and 1 to 0) transitions in each (clock) cycle. This provides a high level of intrinsic parasitic balancing if complementary signals are well balanced.

4.2.4 Static Power Channel

Although static side-channel emissions are less significant compared to dynamic emissions, the recent research has shown that, at least in theory, exploiting data dependency in bulk CMOS, *static* power/leakage is possible [6, 42, 47, 55, 84].

As shown in section 2.1.4, both the subthreshold and gate leakage depend on the transistors configuration – the circuit configuration is *imprinted* to the circuit static power. An open transistor experiences dominantly the subthreshold leakage, while a closed transistor experiences the gate leakage – see Figure 4.3.

The CMOS data dependency was widely studied in the past because of leakage power reduction [136] and also in recent years because of the security consequences of the data dependency [6, 42, 47, 84].

The important effect employed in the data dependency is the so-called *stacking effect* affecting the subthreshold leakage [131, 136]. The stacking effect reduces the leakage in the serial transistor connections – stacks, as illustrated for two NMOS transistors in Figure 4.4. In deeper transistor stacks, the subthreshold leakage is reduced even more [136]. The resulting leakage in the transistor stack thus strongly depends on the number of open transistors in series.

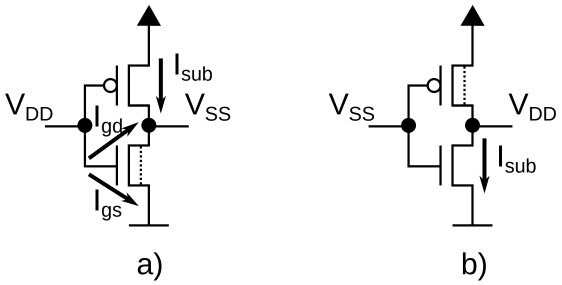


Figure 4.3: a) The closed NMOS transistor experiences a gate leakage and open PMOS experiences subthreshold leakage; b) the open NMOS transistor experiences subthreshold leakage; other leakage sources are negligible

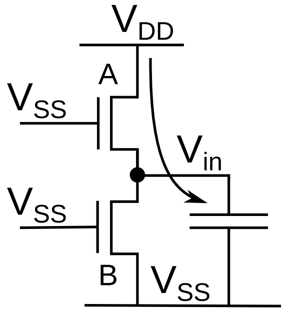


Figure 4.4: In the illustrated transistor stack, $V_{SS} < V_{in}$ while $V_{DD} >> V_{in}$, thus negative gate-to-source bias and the body effect increases V_t in transistor A; V_{ds} in transistor B is low, leading to V_t increase in transistor B [136]

A similar effect applies also to the gate leakage, as illustrated in Figure 4.5.

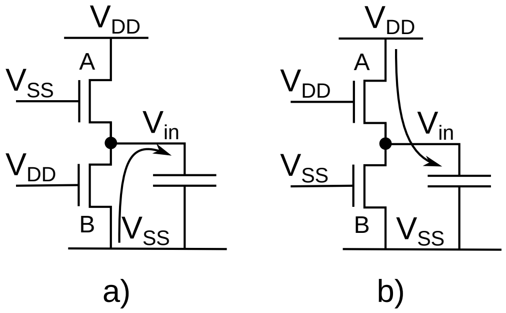


Figure 4.5: a) $V_{in} = 0$ and $V_{gs} = V_{DD}$ for transistor B, thus transistor B experiences full gate leakage, transistor A is closed, thus the gate leakage is negligible; b) $V_{in} = V_{DD} - V_t$ and $V_{gs} = V_t$ for transistor A, thus both transistors A and B experience negligible leakage [131]

Input Configuration	Estimated leakage [nA]		
	I_{gate}	I_{sub}	I_{tot}
000	0.2	0	0.2
001	0.4	0.8	1.2
010	0.4	0.4	0.4
011	6.8	1.5	8.3
100	0.4	0	0.4
101	3.7	0.8	4.5
110	3.7	0	3.7
111	32.0	2.3	34.3

Table 4.1: Leakage data dependency example on the input vector for the 3-input NAND gate in 90nm CMOS technology node with 1.7 nm gate oxide thickness, as reported in [72]

The combination of the subthreshold data dependency – stack effect – and the gate leakage data dependency leads to the dependency of leakage on the combinational input vector as reported for individual gates – see Table 4.1 – and also for benchmark circuits [1, 58, 72].

The older works did not consider this effect much important in general, as they targeted only the power consumption consequences, which were overcome by the process variations [1]. However, recent research has shown that, at least in theory, exploiting data-dependent leakage to compromise the (bulk) CMOS device is possible [6, 42, 47, 84].

Data presented in Table 4.1 for a single gate show a clear dependency between leakage and Hamming Weight (HW) of the gate input. Similarly, the dependency on Hamming weight might be observed for more complex CMOS structures enabling correlation and differential *leakage power attacks* (LPAs) [6, 55].

Static power data dependency countermeasures are less discovered compared to the dynamic power countermeasures. A significant contribution to the field is presented in [55] or [42]. This work represents also a contribution to the leakage countermeasure development.

4.3 Laser-Induced Static Power Channel

This Section represents a contribution of this dissertation thesis to the CMOS side-channel vulnerability research. Section 2.2 describes how the photocurrent is induced in bulk CMOS. The analogy between photocurrent and the subthreshold leakage is also described in Section 2.2: photocurrent depends on the illumination power, while the subthreshold leakage depends on environmental parameters, such as temperature. In past works, lasers were used to inject faults into CMOS, mostly affecting the sequential part of the circuit [66]. Our recent work lights up a novel area: we have shown that exploiting combinational logic without fault injection is possible.

The analogy between laser-induced static power and leakage continues in the data dependency area, as the laser-induced current in bulk CMOS is also data-dependent. In [A.5, A.6, A.7], we have shown that the static power data dependency in the CMOS integrated circuit may be manifested by using a (focused) laser beam. Compared to leakage, the data-dependent static current of the specific circuit part is increased by a factor 4–5: leakage currents are in the order of (tens of) nanoamps, but the data-dependent part of the static *Optical Beam Induced Current* (OBIC) may be in tens or even in hundreds of microamps for a single logic gate depending on the CMOS technology node and the *exposure energy* (laser power). Note that higher exposure energies induce high currents, thus to prevent device destruction, the current must be induced for a short time only (pulsed).

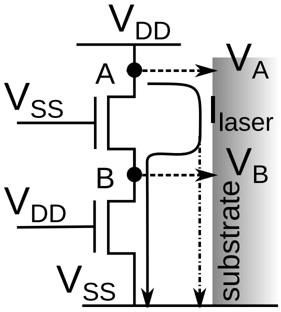
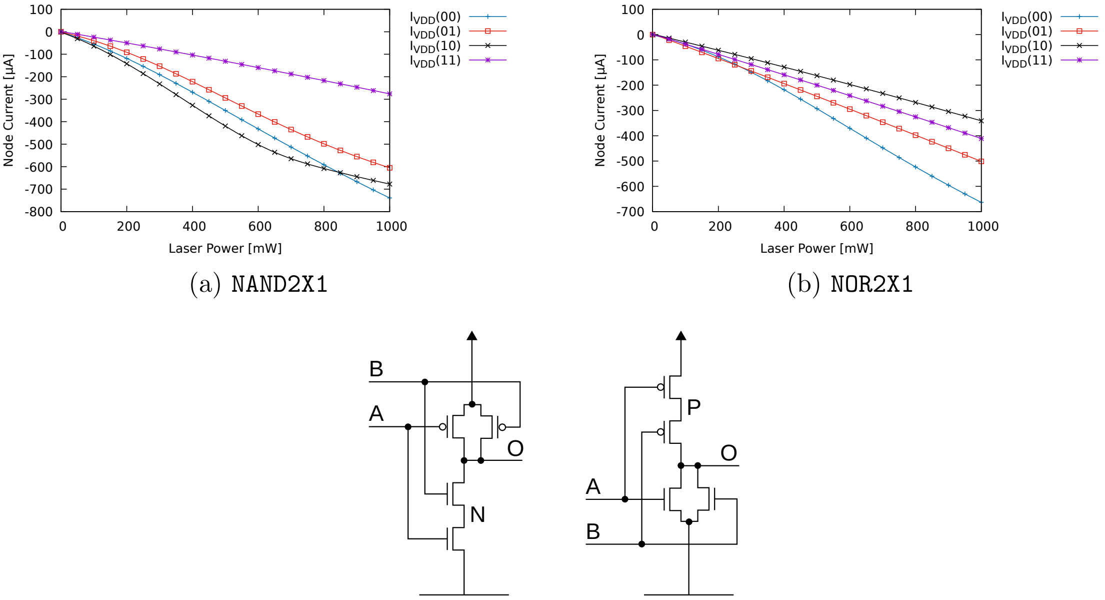


Figure 4.6: In the illuminated transistor stack, $V_B \approx 0$, as the lower transistor is closed; $V_A \approx V_{DD}$, as the upper transistor is open. As a result, the *n+/p-sub* junction in node A is reverse biased, and OBIC is generated. The current loop is closed through node B and substrate

Based on the NMOS/PMOS models under illumination, it is evident that to induce a significant OBIC, NMOS requires significantly less energy to induce similar OBIC than PMOS – see Figure 2.13. The OBIC may also be controlled by the gate voltage. The gate voltage determines the state of the MOS transistor channel, which directly influences the nodal voltage levels in the transistor stack, further affecting the photocurrent – see Figure 4.6. Both leakage and OBIC are thus correlated with the CMOS gate input configuration. This behavior is demonstrated for standard cells in Figure 4.7 for standard two-input gates.



(a) NAND2X1 (b) NOR2X1

(c) NAND2 (d) NOR2

Figure 4.7: The simulated power imprints for negative two-input standard cells in TSMC180nm technology library – the area of the standard cell is uniformly illuminated

In [A.7], we identified that when the PMOS stack arrangement is serial and NMOS arrangement is parallel (e.g. OR/NOR gates), there is lower data dependency compared to complementary gates (e.g. AND/NAND gates). This is caused by the fact that NMOS is more sensitive to illumination compared to PMOS and the NOR input configurations influence the nodal voltages similarly. The explanation of the current imprints from Figure 4.7 follows:

Case 00

In both NAND2X1 and NOR2X1, NMOS parts of the gate are open and PMOS parts are closed:

NAND2X1: $V_O \approx V_{DD}$; $V_N \to V_{SS}$

NOR2X1: $V_O \approx V_{DD}$; $V_P \approx V_{DD}$

In the NAND2X1 gate, the OBIC is generated by one *n+/p-sub* junction in node N and one in node O, while the contribution of the node O is dominant and the contribution of the node N is minor.

In the NOR2X1 gate, the OBIC is generated dominantly by two *n+/p-sub* junctions in node O.

Case 01

In this case, the input `A` is equal to V_{DD}, while `B` is equal to V_{SS}:

`NAND2X1`: $V_O \approx V_{DD}$; $V_N \approx V_{SS}$

`NOR2X1`: $V_O \approx V_{SS}$; $V_P \approx V_{SS} + V_t$

In the `NAND2X1` gate, the OBIC is generated almost exclusively only by the *n+/p-sub* junction in node `O`.

In the `NOR2X1` gate, the OBIC is generated by the *p+/n-well* junctions in nodes `P` and `O`.

Case 10

In this case, the input `A` is equal to V_{SS}, while `B` is equal to V_{DD}:

NAND2X1: $V_O \approx V_{DD}$; $V_N \approx V_{DD} - V_t$

NOR2X1: $V_O \approx V_{SS}$; $V_P \approx V_{DD}$

In the `NAND2X1` gate, the OBIC is generated by the *n+/p-sub* junctions in nodes `O` and `N`.

In the `NOR2X1` gate, the OBIC is generated by the *p+/n-well* junction in node `O` only leading to lower current compared to case `01`.

Case 11

In both `NAND2X1` and `NOR2X1`, PMOS parts of the gate are open and NMOS parts are closed:

`NAND2X1`: $V_O \approx V_{SS}$; $V_N \approx V_{SS}$

`NOR2X1`: $V_O \approx V_{SS}$; $V_P \to V_{DD}$

In the `NAND2X1` gate, the OBIC is generated by the *p+/n-well* junctions in node `O`.

In the `NOR2X1` gate, the OBIC is generated by the *p+/n-well* junctions in node `O`, while the contribution of *p+/n-well* junction in node `P` is small.

4.3.1 Dual-Rail is Not Safe

Dual-rail circuits are composed in such a way that every bit value is encoded using two wires and every gate is replaced by a pair of complementary gates to increase the robustness or introduce uniform power consumption.

As we have shown in [A.5], balancing approaches based on dual-rail logic are generally ineffective against attacks on OBIC in particular. Due to structural differences in complementary gates, most of the dual-rail balancing approaches are principally ineffective against attacks on static power in general.

Even when the circuit is designed as a dual-rail circuit and complementary gates are placed in such a way, so that it is impossible to target a single gate without affecting its complement, the OBIC can still be used to disclose the dual-rail gate input patterns, or at least decrease the entropy of the input pattern – see Figure 4.8.

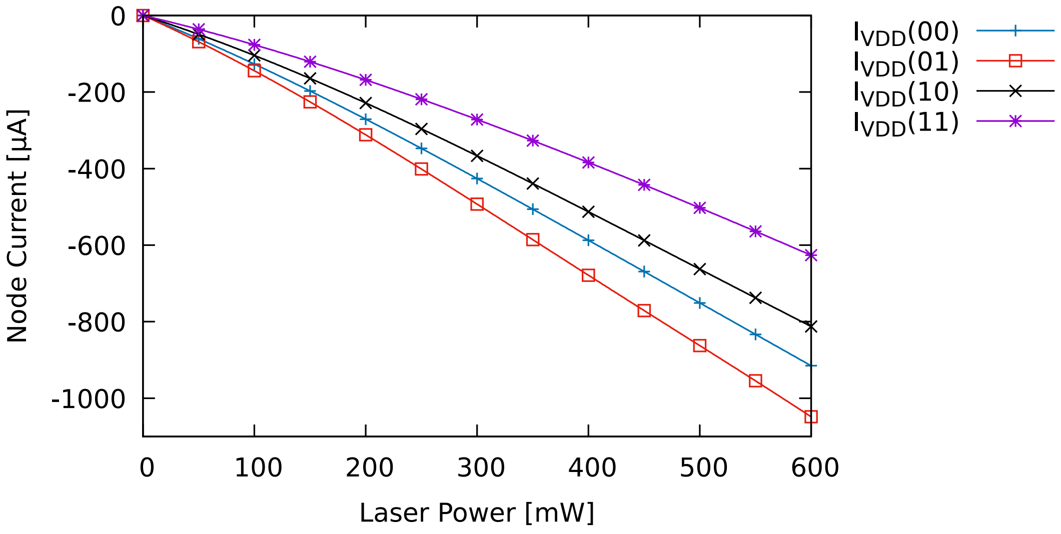


Figure 4.8: The photocurrent for the conventional WDDL `AND` gate composed of `AND2X1` and `OR2X1` gates for different input vectors and increasing laser power balanced with equal load capacitance

According to the best of our knowledge, many dual-rail circuit design styles (and actual designs) [15, 112, 121] will suffer from this behavior, because of differences in gate geometries that cannot be avoided.

In Section 5.5, the AES SBOX vulnerability evaluation results are provided: SBOX is protected using different dual-rail approaches, while all of them show significant and possibly exploitable variances in the simulated OBIC.

4.4 Proposed Attacks

Using OBIC for side-channel analysis brings several advantages, as described above, however, direct reading of processed bits might be possible in a special (and probably rare) case, and with sophisticated equipment only. In most cases, statistics should be employed – as for any conventional side-channel observation or combined attack.

The proposed attacks combine invasive and observation methods to compromise the circuit. The attack should be well targeted to an area of interest limiting the contribution of the surrounding logic to observed power traces.

4.4.1 Attacker Model

To compromise the circuit by a combination of CMOS illumination and static power consumption monitoring, the attacker must be able to decapsulate the circuit, while preserving it operational. This is possible with basic equipment [A.17, 41]. Next, the attacker must be able to synchronize the light source and the measurement equipment. Additionally, if the attacker can control the clock signal, it is a plus simplifying the attack.

Aligned with our published work [A.5, A.6, A.7], we assume that attackers considering light attacks on combinational logic would be of two kinds. We distinguish (i) a *sophisticated attacker* who has access to sophisticated equipment capable of targeted attacks to a small circuit area, who has constant power light source with the ability to perform repeatable experiments including short, area-constrained light pulses (e.g. sophisticated laser bench) and then (ii) a *mid-equipped attacker* with cheaper equipment allowing simpler attacks targeted to larger circuit areas, e.g., a poorly focused light source with limited repeatability, especially for short pulses. Both scenarios are possible, even if somehow challenging. Targeting an attack even to a very constrained circuit area by a laser beam is proven [66, 104, 105, 107, 108].

Based on the attacker abilities, we distinguish two attack scenarios: *Preciselly-Targeted Attack* and *Block-Targeted Attack*.

4.4.2 Preciselly-Targeted Attack

The sophisticated attacker can perform an attack *precisselly-trageted* to few standard cells, small CMOS structures, or even to a single CMOS cell area only.

If the attacker can determine the location of the cell or a small CMOS structure of interest and he can illuminate predominantly only the structure of interest, he may directly read the value at the moment present at the structure inputs. E.g., in [A.6], we have shown that conventional TMR voters represent significantly vulnerable structures. If the conventional 3-input voter operates in a fault-free environment, its inputs are equal. Then, the overall voter structure is driven by three equal inputs and works as an amplifier when illuminated: the difference between `000` and `111` inputs in the current induced by illumination is significant. Additionally, the area of several standard cells forming a voter circuit may be targeted simpler than the area of a single cell. This might be possible even in advanced technology nodes, where the voter area approaches the order of micrometers.

The algorithm of the attack to any (but small) CMOS structure – not only to a single standard cell, should be as follows:

1. Select the target CMOS structure of interest and create its SPICE model under illumination (e.g. using our published models).

2. Simulate all input patterns for the structure of interest under defined illumination intensity; alternatively use another device as the template and measure the current response of the target structure in the template device. The template device may have a different configuration, but inputs of the target structure must be known.

3. Sort the simulated or templated current responses by the input pattern of the target structure.

4. Stimulate the target structure in the device-under-attack for several input patterns: the device inputs are known, while the target structure inputs (depending on the secret) are wanted (unknown).

5. Intuitively: sort the current responses for the device under attack and assign them to structure input patterns according to the template or simulation to get the wanted input patterns of the structure under attack; or more formally and scalably: apply steps 3 and 4 of the algorithm described in Section 4.2.2, where power model comes from the simulation or template.

4.4.3 Block-Targeted Attack

The *mid-equipped attacker* can only target a wider circuit area (e.g. block) and can use limited illumination energy only. The illumination of the combinational logic block can be used to highlight its data-dependent power in the power trace of the device.

The attack algorithm we propose for this kind of attack follows the CPA attack described in Section 4.2.2. The difference from a standard CPA or LPA attack is in the power model only. Conventional attacks use simple models like Hamming Distance (HD) or Hamming Weight (HW) of the data [55], but for the attack exploiting OBIC, we recommend a more precise power model. Simple models are unable to characterize the OBIC of the CMOS logic well enough, even though they are conventionally used for LPA [55]. Ideally, the power model should be created by SPICE simulation of the target structure under all input vectors. Our SPICE models can be used even for similar bulk CMOS technologies.

As an alternative to SPICE simulation, a template-based model [25] may be used analogously to the simulation in case of both *Precisely-Targeted* and *Block-Targeted* attacks.

4.4.4 Simplified Power Model of Complex Structures

The attacks exploiting OBIC require a relatively precise power model. In the case of the *Block-Targeted* attack, exhaustive simulation of bigger CMOS circuits might be required. To further ease the attack, a simplified composite power model can be used.

The simplified power model does not require electrical simulation of large circuits. From the attacker's point of view, simpler approaches employing open tools only potentially reduce the attack cost.

The simplified power model for a given circuit is a *tuple* containing the OBIC for each circuit input vector. To construct this power model, only standard cells (single gates) must be pre-simulated in SPICE (or characterized another way). Their responses are then placed in the *tuple* P, and are used in connection with the knowledge of the circuit configurations under all given input vectors to *compose* the complete power model. The circuit configuration extraction under the given input vector is a straightforward task[1]producing the *tuple* N. The power model for a given input vector is the sum of data-dependent OBICs of all standard cells (pre-simulated in SPICE) in the given circuit configuration. As a result, only the standard cells used in the circuit are to be simulated in SPICE, instead of the whole circuit.

The circuit configuration for a given input vector can be described by the following *tuple*:

$$\begin{aligned} N = \{g_0(0..0), g_0(0..1), g_0(1..0), g_0(1..1), \ \dots \\ \dots \ g_{n-1}(1..0), g_{n-1}(1..1)\}, \end{aligned} \tag{4.3}$$

where the respective g_i's represent the numbers of gates in the respective configuration, and n represents the number of gate types used in the circuit. The pre-simulated power for each gate in the circuit is organized in the following *tuple*:

$$\begin{aligned} P = \{p_0(0..0), p_0(0..1), p_0(1..0), p_0(1..1), \ \dots \\ \dots \ p_{n-1}(1..0), p_{n-1}(1..1)\}, \end{aligned} \tag{4.4}$$

where the respective p_i's represent the pre-simulated power for the respective gates in the given configuration.

The power model for the j-th circuit input vector is given by the following sum:

$$m_j = \sum_{i \in [0, n-1]} p_i \cdot g_i, \tag{4.5}$$

while the complete power model for the circuit is a *k-tuple*, a lookup table, composed of power models for every circuit input:

$$M = \{m_0, \dots m_{2^{k-1}}\}, \tag{4.6}$$

where k is the number of circuit inputs.

[1] Circuit configuration extraction is provided by input vector simulation in the TSaCt2 framework [19]

To illustrate the power model generation, we use the AES SBOX circuit composed of 866 2-input NAND gates only ($n = 1$) as an example. The SBOX circuit has 8 inputs ($k = 8$). Figure 4.9 shows the resulting power model for all 256 distinct circuit configurations related to 256 different circuit input vectors.

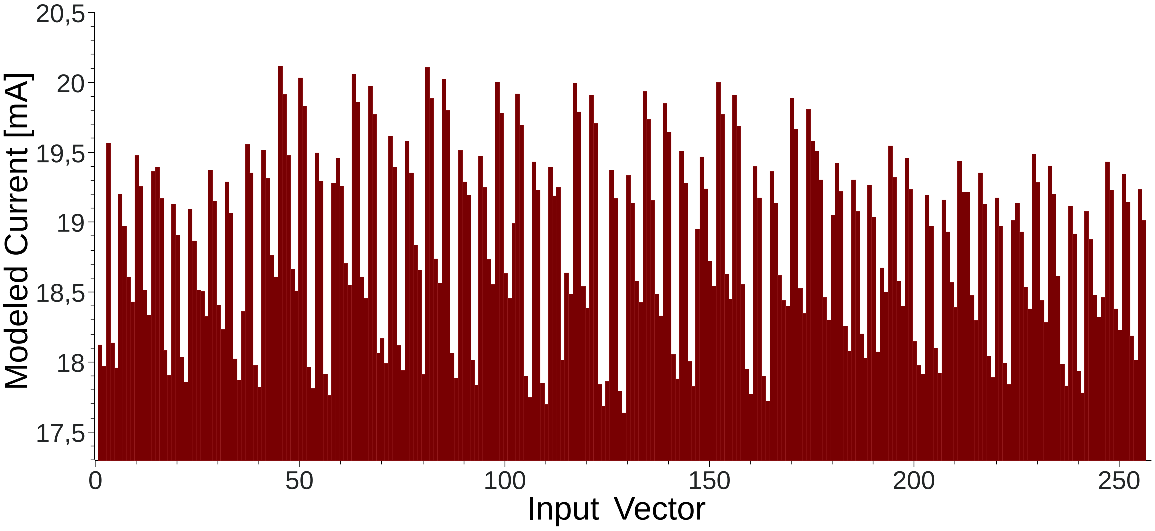


Figure 4.9: Simplified power model of the example AES SBOX circuit composed of 866 NAND gates illuminated by 50mW of the equivalent power – the power model is related exclusively to the *time of illumination* – the attacker's point-of-interest

The advantage of the look-up table-based approach is that it can be mount on a bigger circuit without the need for time-intensive simulation or templating and only a standard cell characterization is required. Its disadvantage is that it becomes less accurate with rising illumination power, as higher illumination powers cause voltage drops in the circuit, affecting the induced currents – the simple look-up table reflects reality less accurately [A.5].

Chapter 5

Balanced Standard Cells

> *Mixing spiced, single-flavored, or randomly-flavored cocktails . . . cause unpleasant conditions to enjoy a drink.*
>
> *Low-level countermeasures*

The existing power side-channel attack countermeasures conventionally use masking by randomization to break the relation between processed data and the power trace or hiding by increasing noise or by limiting the power trace variations. To diminish the data dependency and enhance attack resistance, we designed a novel hiding approach employing CMOS standard cells with decreased variability in power imprint.

5.1 Motivation

In Chapter 4, we described the novel vulnerability potentially affecting most of today's CMOS designs. This chapter addresses the problem of static power data dependency at the circuit level. To diminish the data dependency and enhance attack resistance, several strategies described in Chapter 4 could be used.

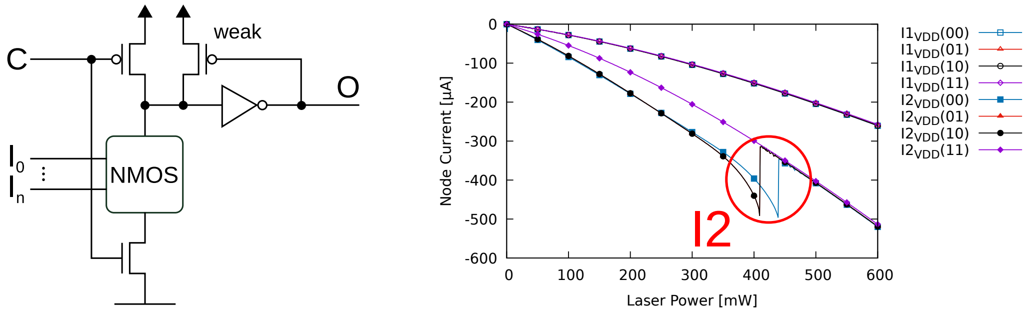
One of the strategies, which could be employed, is circuit-level hiding. The advantage of the hiding strategy is that it is easy to employ without affecting the overall design flow. In this chapter, we propose a novel hiding approach for the CMOS standard cell design with decreased variability in power imprint. The proposed structures were compared to existing alternatives and their value was confirmed by simulation.

5.2 Related Art

The existing design techniques providing a significant level of data-independence include namely dynamic logic, where the PMOS stack is reduced to a single transistor [131], and methods employing SecLib gates [51, 52]. Their relevance was first identified in [A.7].

5.2.1 Domino Logic Employing Single Precharge PMOS

We identified the dynamic logic circuit design styles as promising, as they limit the data dependency in the PMOS stack by replacing the stack with a data-independent precharge transistor.



(a) Footed domino logic gate employing standard *weak keeper*

(b) Domino logic two-input AND gate without keeper (I1) and with the standard weak-keeper (I2) power imprints

Figure 5.1: Domino logic gate structure and power imprint example

The structure of conventional dynamic (domino) gates provides natural hiding. The hiding is given by the *charge leakage* [131], as the precharged internal node in the domino gate tends to discharge fast even for small illumination energy, leading to a (almost) constant state of the gate under any input pattern – see I1 in Figure 5.1b. In dynamic logic, the charge leakage is often compensated by a *weak keeper* [131] – see Figure 5.1a. The domino gate with a weak keeper has commonly a data-dependent power imprint with a notable drop in the current characteristics at the characteristic illumination energy which can change the output of the gate – see I2 in Figure 5.1b. Fortunately, by altering the sizes of the weak keeper and the output inverter, the data-dependency may be decreased significantly.

5.2.2 Symmetric SecLib Gates

One of the known approaches employing classical static CMOS with increased symmetry – at both schematic and layout levels – is called SecLib. The symmetry is achieved by following the SecLib gate design guidelines described in [51, 52]. The original SecLib dual-rail AND gate is shown in Figure 5.2a.

Originally, we assumed that the perfect SecLib symmetry leads to perfect balancing, however, we discovered a hidden asymmetry in the area-efficient SecLib version.

If the SecLib gate is designed to be area-efficient, it should employ *dynamic C-elements*. Unfortunately, dynamic C-elements suffer from *charge leakage* [131] if C-element inputs are not equal. Charge leakage in combination with circuit illumination (even for small energies) turns the C-element output to `1`. This uncovers a hidden asymmetry in SecLib: the second OR gate is fed by one C-element only, while two other inputs are grounded and the grounded inputs are not affected by the charge leakage.

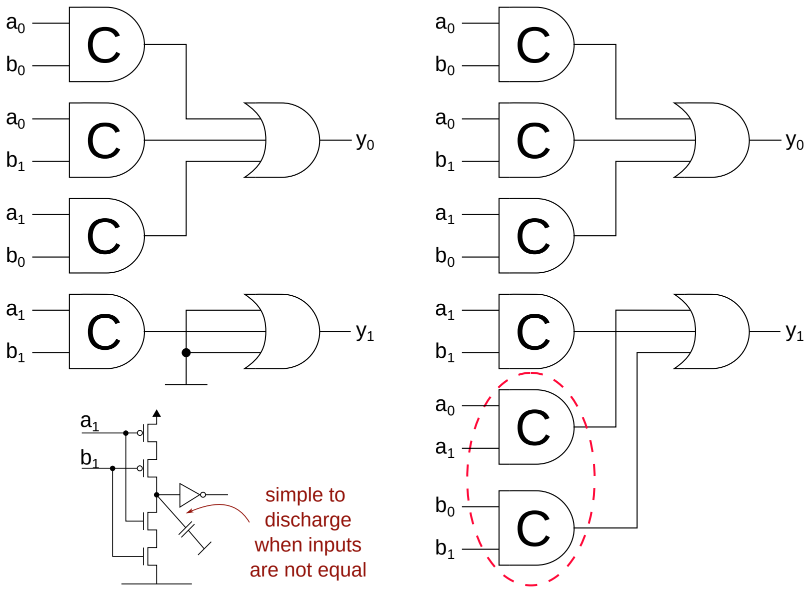
In most cases, both outputs of the SecLib gate under illumination are `1`, as the charge leakage causes at least one input of both ORs to be `1` – the C-element parasitic capacitance shown in Figure 5.2a is discharged when C-element inputs are not equal. However, when the input of the dual-rail SecLib gate is `00` ($a_1 = 0$, $b_1 = 0$, $a_0 = 1$ and $b_0 = 1$), the bottom C-element does not experience charge leakage, as both of its inputs match and it produces output equal to `0`. The other inputs of the lower OR gate are also `0`, thus the OR gate output is `0` not `1`. This allows distinguishing the `00` input of the SecLib gate – the gate experiences a data-dependency: the voltage output is data-dependent and subsequently also the power imprint is imbalanced. The output data-dependency is shown in Figure 5.3a. The lower OR gate turns its output when the illumination power is high enough to cause the C-element to change its output – see Figure 5.3a – the input pattern is masked for high energy only.

Although we identified this vulnerability in connection with OBIC, it can arise in different contexts as well and it enables fault-attacks in general. In a different context, it might be less obvious but it is still present. The output of the dynamic C-element depends on the charge *stored* in the parasitic capacitance – see Figure 5.2a. Even a small increase in subthreshold leakage caused, e.g., by temperature, may have a similar effect [131]. Also the traditional fault-injection techniques [66] targeted at the SecLib gate area will cause that the SecLib gate output for the `00` input pattern will be different than the output of other input patters. This is a principal vulnerability that can be exploited in a fault or combined attack [66]. As a result, the SecLib structure employing dynamic C-elements should be considered vulnerable in general.

A possible solution to the problem with SecLib asymmetry is shown in Figure 5.2b: two additional C-elements producing constant 0 are added. These C-elements *are* prone to charge leakage, as their inputs during the dual-rail evaluation phase do not match. It is possible to omit one of the added C-elements and still obtain a data-independent output and subsequently also increased balance in power imprint for photocurrent, as shown in Figures 5.3b and 5.4b respectively. On the other hand, by omitting one of the C-elements, the SecLib gate becomes unbalanced from the dynamic power perspective: loads of all input signals will not be equal. Thus, using both C-elements is recommended.

Note that even SecLib optimized by added C-element(s) experiences a power imprint imbalance – see Figure 5.4. The remaining imbalances are given mostly by asymmetries in the serial CMOS transistor stacks.

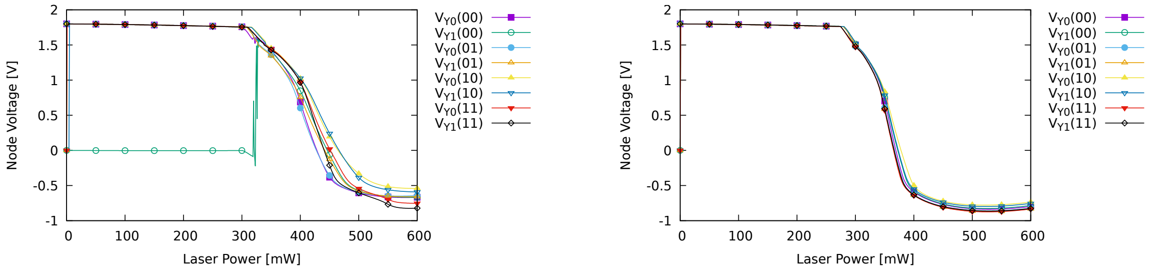
To further decrease the power imprint data dependency of SecLib, mainly the asymmetries connected with signal ordering in the serially arranged transistor stacks should



(a) Original SecLib gate and the dynamic C-element structure

(b) Optimized gate providing better OBIC balancing and consistent output

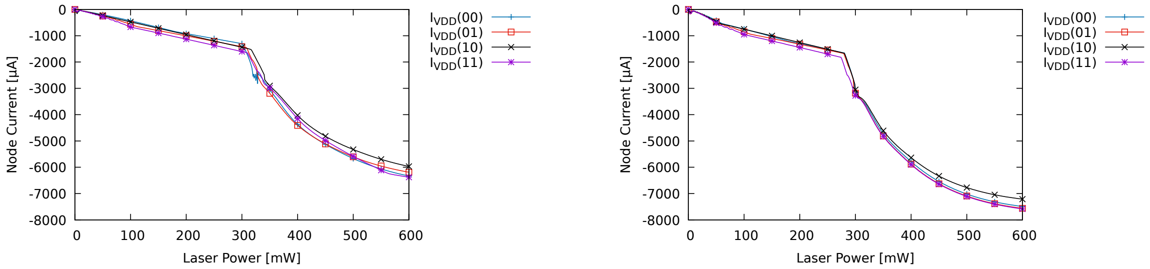
Figure 5.2: Secured 2-input AND gate schematics: all input combinations at C-element inputs are represented; if not illuminated, one C-element output is always equal to 1 and remaining C-element outputs are always equal to 0



(a) SecLib

(b) Optimized SecLib

Figure 5.3: Output voltages for SecLib and the optimized SecLib



(a) SecLib

(b) Optimized SecLib

Figure 5.4: Induced photocurrents for SecLib and the optimized SecLib

be removed. The parallel arrangement of duplicated CMOS stacks with permuted inputs would lead to further suppression of the data dependency, for a price of further increase of the SecLib gate area.

Both domino logic and SecLib share a nice property: they were designed to support dual-rail encoding computation, thus (if employed in a dual-rail circuit) provide (at least) a basic level of dynamic power attack resistance. On the other hand, they suffer from significant disadvantages. SecLib suffers mainly from a large gate size. The increased gate size influences not only the circuit static power or delay but also the circuit security [A.5]. On the other hand, domino logic provides a small area footprint, but it suffers from general dynamic logic disadvantages including the need for careful clocking or increased dynamic power [131].

5.3 Structures Enabling Static CMOS Current Balancing

Compared to the dynamic logic, and domino logic, in particular, we propose compact static cells that counter the attack by balancing and may complement SecLib as a countermeasure against attacks on light-modulated static power described in Section 4.4.

The circuit vulnerability connected with the light-induced static current (OBIC) may be compensated only when – in the case of the illumination attack – the entire balanced structure is exposed to the same light intensity. This natural requirement may not be guaranteed for bigger structures: for larger exposure areas, a precise attack employing imbalancing becomes feasible. The size of the balanced CMOS gate is extremely important.

In this section, we describe approaches to balance traditional CMOS gates to decrease data dependency between leakage or OBIC and gate input patterns. The severity of OBIC data-dependency is more significant, and thus the emphasis is on breaking OBIC data-dependency. The approaches employing inverter balancing and constant current source approximation are – according to the best of our knowledge – novel in the security context.

5.3.1 Inverter Balancing

The first approach originates in the fact that two equally sized cascaded inverters work with complementary values, and thus may provide constant (mutually balanced) power imprint. It is simple to balance a two-inverter chain resulting in a buffer with constant static power imprint – see Figure 5.5a. Note that the equally sized inverters working with complementary input values at the same time also provide leakage balancing.

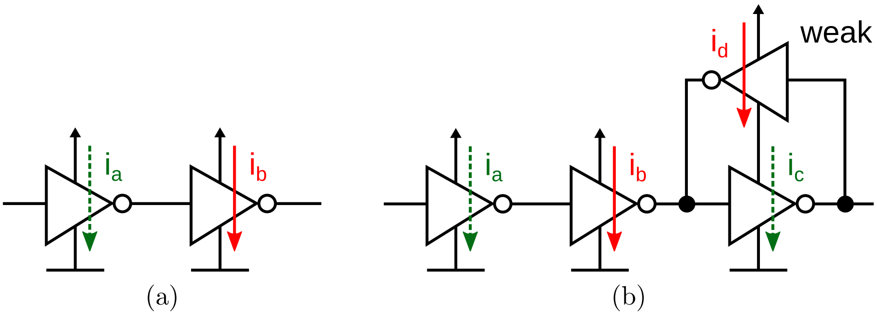


Figure 5.5: Two-inverter chain (a) uses complementary power consumption to obtain a constant power imprint: $i_a + i_b = const.$; three-inverter chain with feedback weak inverter (b) uses the same principle

If an inverting structure in general, and inverter cell in particular, should be balanced, an odd number of inverters in the linear chain is required. A straightforward balancing strategy is to alter inverter sizes in the chain to provide balancing. An alternative option is to employ a three-inverter linear chain equipped with an in-the-cell compensation feedback inverter. From the integration perspective, it is important to ensure that the balanced cell provides a high-impedance input, thus using only a single inverter with a feedback is not possible.

Note that the output inverter may also be used for (at least partial) balancing of the power consumption of arbitrary negative CMOS structures – such as NAND or NOR gates in particular. This is the reason why static CMOS positive gates (e.g. AND, OR) provide a limited intrinsic level of balancing, overcoming the negative gates. We found that it is relatively simple to enhance the balancing efficiency by an output inverter scaling, which provides an opportunity to increase the circuit attack resistance by a small modification of existing positive gates, namely in circuits employing only positive gates (e.g. some dual-rail circuit families).

In general, any approach employing positive gates, where the output inverter is comparable to the negative part of the gate, provides an intrinsic balancing. It is also the reason why the mentioned domino logic provides a good level of power compensation by design.

5.3.2 Constant Current Source Approximation

The other group of current-balancing approaches we developed is based on the idea of modification of a CMOS gate to mimic the *Constant Current Source* structure, as shown in Figure 5.6a.

In the case of a generic static CMOS gate, the data-dependent load resistance is represented by PMOS and NMOS stacks. The standard approach in the constant current source approximation is to employ a fixed serial resistor significantly bigger than the load resistance. To avoid data-dependent power imprints of the gate, the resistance of data-dependent components of the CMOS stack should be decreased compared to the fixed (data-independent) components. To achieve this goal, we developed three approaches at transistor level:

Adding Serial Transistor

A straightforward approach is shown in Figure 5.6b. A single – normally closed – transistor with constrained channel width is added in series with the NMOS or PMOS part of the circuit. This arrangement converts the parallel arrangement of the PMOS or NMOS blocks to quasi-serial, limiting the data dependency. This behavior is conditioned by the CMOS stack arrangement introducing a new (separated) voltage node in the transistor stack. This transistor is related to $\mathtt{R_{SP}}$ and $\mathtt{R_{SN}}$ in Figure 5.6a. Although it is normally closed, it does not affect leakage data dependency and it reduces the OBIC data-dependency in case of an illumination attack.

Adding Light-Sensitive Parallel Transistor

The second approach is shown in Figure 5.6c. The parallel connection of normally-open PMOS and NMOS transistors to PMOS and NMOS stacks has no significant effect in the case of normal operation (except for the increased leakage). The added parallel structures

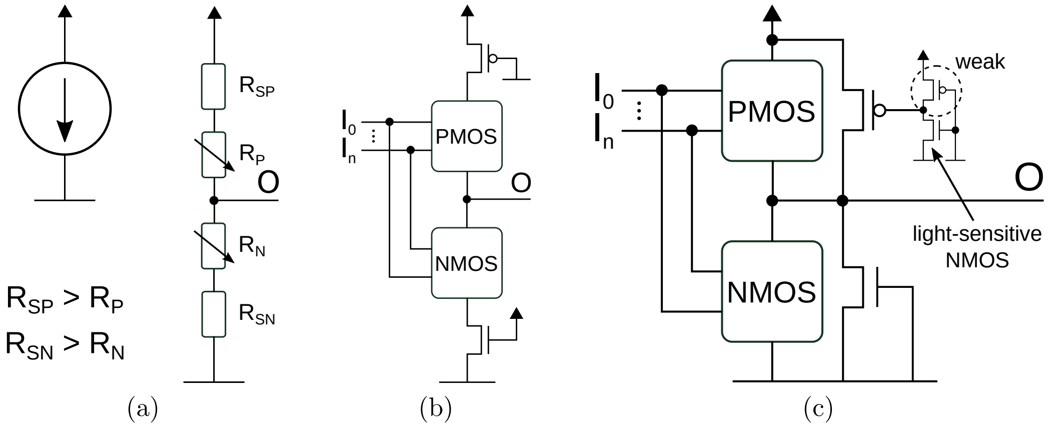


Figure 5.6: Constant current source approximation (a) by small (data-dependent) potentiometers and larger fixed serial resistors; serial transistors (b) are employed to increase the fixed component of the resistance; and the parallel transistors (c) are used to decrease the data-dependent component of resistance

should be significantly bigger than the data-dependent part of the PMOS and NMOS stacks.

In the case of low illumination intensity, added parallel structures are the dominant source of the photocurrent. For low illumination intensity, the main component of OBIC depends on one of two states of the gate output `O` (`0` or `1`). The OBIC does not directly depends on four gate inputs (`00`, `01`, `10` or `11`). The state of the gate output is given by the gate input configuration (it is data-dependent), however, the OBIC is determined by the gate output, thus variations are significantly decreased. The dominance of parallel structures helps to mimic the inverter behavior of this structure. If such a structure is connected to the input of an inverter of similar size, as described in Section 5.3.1, almost perfect balancing is achieved.

When the illumination intensity grows, the parallel transistor structures can be closed: it causes that the structure depends only on the illumination intensity, not on input data. We have noticed that for higher illumination intensity, NMOSes themselves can be efficiently used as light sensors (for higher illumination intensity), as their conductivity under illumination grows rapidly. The NMOS parallel transistor control can thus be realized only by grounding its gate. On the other hand, the PMOS requires control by a dedicated light sensor.

Our simple light sensor is an ordinary CMOS inverter, whose light sensitivity is increased by strengthening its NMOS part and weakening its PMOS part – see Figure 5.6c. This arrangement ensures that the light sensor is easy to integrate into a CMOS cell, it requires no additional process tuning, and has a very small footprint.

The resulting geometry employing parallel transistors decreases the relative significance of the data-dependent resistance of the PMOS and NMOS blocks under illumination.

Disconnecting Rail

The third approach is shown in Figure 5.7. When the CMOS circuit is under attack, disconnecting one or both of the rails feeding the NMOS and PMOS parts decreases the data dependency significantly. The same light-sensitive inverter is a source of the first control signal (C1) – the first control signal may be employed for parallel PMOS control and to disconnect the VSS rail, however, one additional inverter is required to generate the second control signal (C2) to disconnect the VDD rail. This approach represents a significant increase of the serial resistance denoted R_{SP} and R_{SN} in Figure 5.6a.

Connecting the second inverter to the light sensors makes two control signals available in the CMOS cell. These two signals can be used to control all additional serial and parallel transistors in the cell, thus gates of all transistors can be controlled by the single light sensor, not by multiple independent transistors with grounded gates. The resulting behavior of the cell is then more deterministic.

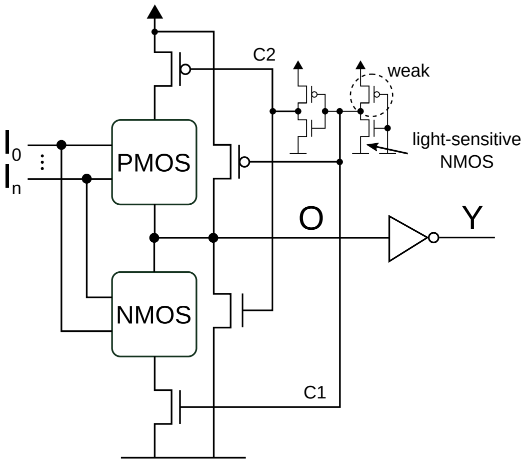


Figure 5.7: Completely balanced positive gate: the output inverter serves for power balancing and as the output voltage filter at the same time

5.3.3 Output Voltage Filtering

The careful design of the secured cell also includes the output voltage to be without significant variations and far from the *intermediate voltage* region. The output inverter serves not only for preceding structure OBIC balancing (as described in Section 5.3.1) but also as a voltage filter separating the internal node O suffering from voltage drops and variations, which can fall into *intermediate voltage* level. In the case of previous approaches used, the internal node voltage is strongly influenced by the added balancing logic and the importance of the inverter as an output voltage filter is increased.

5.3.4 Increasing Transistor-Level Symmetry

The last approach related to balancing leakage and OBIC currents is dedicated to balancing asymmetries in the transistor stack. This modification aims to increase the similarity when equivalent input patterns (e.g. `01` and `10` for `NAND2X1`) are at the gate inputs. Although such patterns should intuitively imply equivalent leakage and OBIC currents, they do not, due to the effects similar to the *stack effect* asymmetric behavior in NMOS described in Sections 4.2.4 and 4.3. A simple approach based on transistor duplication and signal swapping can be used to fight asymmetry: the approach is not new – it is described in classical literature [51]. The symmetrized standard cells may be provided in the standard cell library – see [A.7] for further description.

5.4 Proposed Standard Cells

Conventional CMOS design utilizes standard building blocks called *standard cells* – see Figure 5.8. The cells are carefully designed to optimize the circuit area and performance. Our standard cells are designed with an additional dimension in mind: the constant power imprint under illumination for enhanced security.

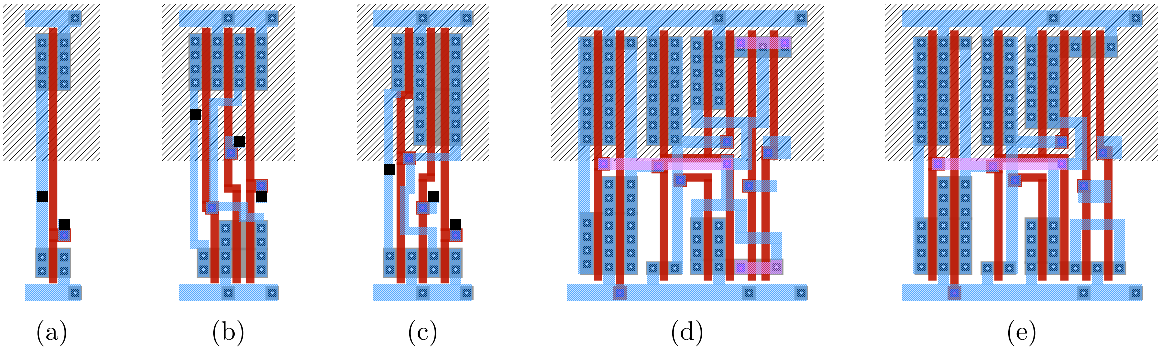


Figure 5.8: Standard cells in TSMC180nm: (a) `INVX1`, (b) `AND2X1` and (c) `OR2X1` from the TSMC180nm library provided by Oklahoma State University (OSU); and proposed: (d) `PAND2X1` and (e) `POR2X1`

The proposed standard cells (see Figure 5.8d and 5.8e) follow the principles described in Section 5.3 and are designed according to the MOSIS SCMOS rules and passed all DRC checks provided by Magic [34]. The cells employ all approaches described in Section 5.3 except the symmetrization approach, as it conflicts with the design rule 5 formulated below. The cells extend the OSU TSMC180nm cell library and are compatible with other library cells. The proposed cells are optimized according to SPICE models reflecting the data-dependent behavior of the CMOS under illumination.

During the standard cell design, we carefully iterated through the design space and we formulated design rules for our method. Some of the rules extend the original proposals presented in [A.7]. The rules can be used to design a custom cell according to our method:

1. the light-sensitive inverter N-channel width approaches the allowed maximum for the given standard cell height and the width of the P-channel approaches the minimum;

2. the control inverter connected to the output of the light-sensitive one is designed with opposite channels widths;

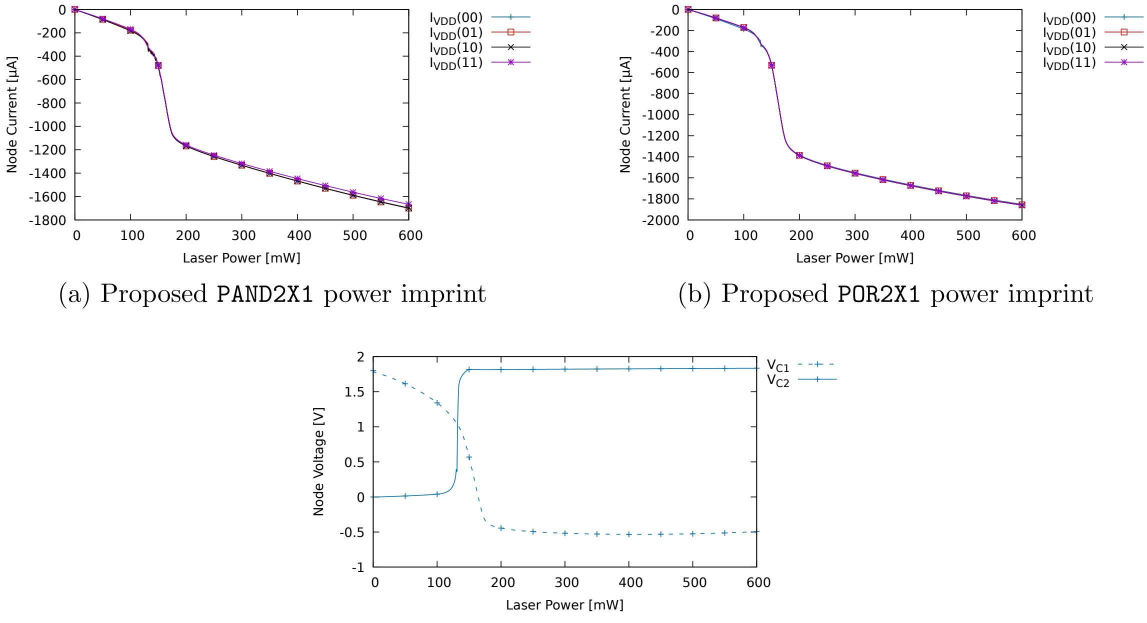
3. serial transistors are used to disconnect only the functional PMOS and NMOS part of the CMOS cell, not to disconnect parallel transistors;

4. parallel transistors are connected directly between the internal node and GND or VDD respectively;

5. the size of transistors controlled by the gate inputs is as small as possible[1]

6. the output inverter is optimized to balance the power imprint for lower light energies only, while its size ensures low light-sensitivity and acceptable load capacity for normal circuit operation;

Rule 1 allows to increase the sensitivity of the light sensor and increases the falling edge slope of the first control signal and Rule 2 helps to increase the rising edge slope of the second control signal – see Figure 5.9c.



(a) Proposed `PAND2X1` power imprint

(b) Proposed `POR2X1` power imprint

(c) Control signals

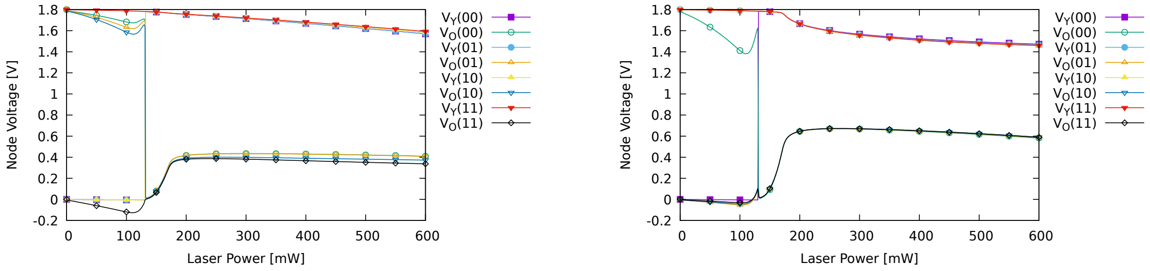
Figure 5.9: `PAND2X1` and `POR2X1` power imprints and control signals in the point-of-interest

Rules 3 – 6 represent a *Divide-and-Conquer* approach and make the overall CMOS structure more robust and easier to optimize: the parallel transistors are open for lower illumination intensities (laser powers), thus the power imprint of the structure is given only by the combination of power imprints of the PMOS and NMOS parts extended by the serial and parallel transistors and the output inverter, which is relatively easy to balance for lower illumination intensity.

For higher illumination intensity, the NMOS and PMOS parts are completely disconnected, while the parallel transistors are closed bringing the structure into a *short* – this fact further restricts longer light pulses with high energy, as it would lead to CMOS structure destruction. The internal node voltage is fixed to a value close to logic zero and the

[1] The size requirement may collide with the Transistor-symmetrization described in Section 5.3.4, as the symmetrization increases the minimal area of the transistor stack

gate output is fixed to logic one implying a constant power imprint for any input pattern. The size of the output inverter must also ensure low voltage drops even for high energies to minimize affecting subsequent circuit levels – near-threshold voltage values may lead to imbalances.



(a) Proposed `PAND2X1` node voltages

(b) Proposed `POR2X1` node voltages

Figure 5.10: `PAND2X1` and `POR2X1` internal node (`O`) and output node (`Y`) voltages

For the purpose of evaluation, we employed open tools and resources described in Appendix B: we used the ngSPICE simulation of TSMC180nm technology node standard cells. The resulting power imprints are presented in Figures 5.9a and 5.9b. The behavior below and above the illumination power ≈150mW in Figure 5.9 and the voltage changes shown in Figure 5.10 clearly distinguishes the two standard cell operational and one transient region. The regions are shown in Figure 5.11. For lower illumination powers, the gate performs a normal operation (region (i)). For higher powers, the gate output is constant and the gate experiences a short circuit (region (ii)) – see Figure 5.10.

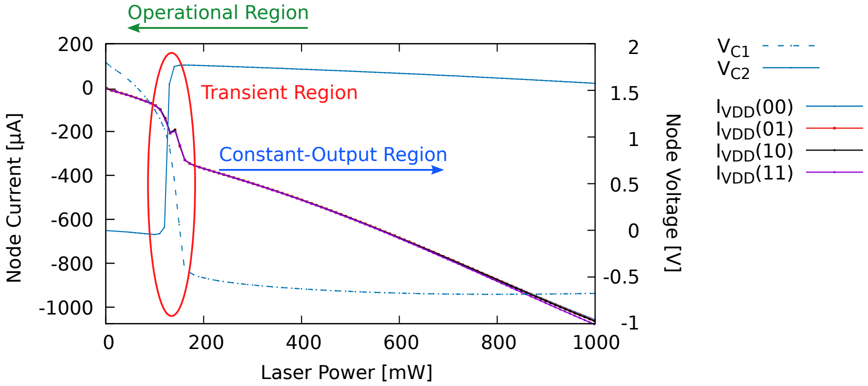


Figure 5.11: Proposed cell operation regions: (i) normal-operation region; (ii) constant-value-output region and (iii) transient region

The proposed cell comparison with the standard – unprotected – library cells and SecLib cells composed of library cells[2] is provided in Table 5.1. The protected cell size is increased

[2] SecLib approach uses library cells, custom C-elements were drawn for TSMC180nm library

Table 5.1: Comparison of Proposed Cells, SecLib Cells and their Standard Counterparts

Standard Cell	Area	Delay	Input Load	Drive Strength
Protected AND (`PAND2X1`)	≈260%	≈250%	≈30%	≈200%
Protected OR (`POR2X1`)	≈260%	≈280%	≈20%	≈200%
Dual-Rail Cell	**Area**	**Delay**	**Input Load**	**Drive Strength**
Protected (`PAND2X1` + `POR2X1`)	≈260%	≈280%	≈25%	≈200%
SecLib	≈434%	≈250%	≈400%	≈100%
Optimized SecLib	≈525%	>250%	≈600%	≈100%

compared to the unprotected standard cells, however, the delay remains acceptable and additionally, the input load of the protected cells is in general lower, and the output drive strength of the proposed cells is increased compared to standard cells due to the requirements given by the optimization process and design rules presented above, allowing lower delay penalty in a real circuit. The competing dual-rail SecLib gates are much bigger and are affected by a great increase of the input load. Additionally, the proposed gate power imprint is balanced even in the single-rail arrangement.

5.5 Case Study: SBOX Vulnerability Evaluation

To analyze the proposed structures in detail, we synthesized a larger combinational circuit implementing a crypto-function, namely the AES SBOX [28]. We synthesized four different circuit variants of the SBOX combinational function. One unprotected single-rail implementation and four implementations employing dual-rail encoding as a dynamic attack countermeasure:

- *singleRail* variant employs only two-input NAND gates (`NAND2X1` and `INVX1`)
- *dualRailAS* variant is a non-conventional dual-rail implementation with alternating spacer [111] employing only two-input NAND and NOR gates and inverters (`NAND2X1`, `NOR2X1` and `INVX1`) allowing lower overhead
- *dualRail* variant is a conventional dual-rail implementation [112, 121] employing only two-input AND and OR gates (`AND2X1` and `OR2X1`)
- *pDualRail* variant is a conventional dual-rail implementation employing only proposed two-input AND and OR gates (`PAND2X1` and `POR2X1`)
- *secLibDualRail* variant is a protected implementation employing SecLib gates based on six dynamic C-elements and library cells (`INVX1` and `NOR3X1`)

The SBOX was described in Verilog, then synthesized and optimized by *Yosys* [134] and *Berkeley ABC* [83] respectively and finally mapped by a custom tool TSaCt2 [19] to obtain netlists for all variants under evaluation. The following Yosys script was used:

```
# read design
read_verilog sbox.v

# elaborate design hierarchy
hierarchy -check -top sbox

# the high-level stuff
proc; opt; fsm; opt; memory; opt

# mapping to internal cell library
techmap; opt

write_blif sbox.blif
```

The following ABC script was used to optimize the netlist and produce the *aiger* output:

```
# read the library of standard
# 2-input gates for the map command
read_library 2-gates.genlib

# read blif produced by yosys
read_blif sbox.blif

# transform to AIG
strash

# synthesis script
dch; map; mfs; balance
...
[20 iterations]
...
dch; map; mfs; balance

# write output
write_aiger sbox.aig
```

The mapped netlists for all variants were then placed by *GrayWolf* [50] and routed by *QRouter* [35]. *Magic* [34] was used as a primary VLSI layout tool, while custom scripts were used for model extraction, simulation control, and data processing.

For evaluation, we employed the toolchain setup described in Appendix B: we used the ngSPICE simulation of TSMC180nm technology node standard cells: the layout models were simulated in ngSPICE. The largest netlists (pDualRail, secLibDualRail) were partitioned employing a custom procedure to enable a step-by-step simulation, as an en-bloc simulation in ngSPICE was not possible. The partitioning guarantees that the simulated dynamic power given by the load capacitance is pessimistic; however, the influence of glitches must not be preserved in all cases. The static light-induced power simulation accuracy is not affected significantly.

Table 5.2: Area/Delay overhead comparison of different SBOX implementations

SBOX implementation	Area [mm^2]	Delay [ns]
singleRail	0.038 (100%)	≈ 9 (100%)
dualRailAS	0.057 ≈150%	≈ 11 (≈120%)
dualRail	0.066 (≈170%)	≈ 11 (≈120%)
pDualRail	0.158 – 0.196 (≈400% – 530%)	≈ 12 (≈130%)
secLibDualRail (optimized)	0.294 – 0.431 (≈780% – 1150%)	≈ 15 (≈ 160%)

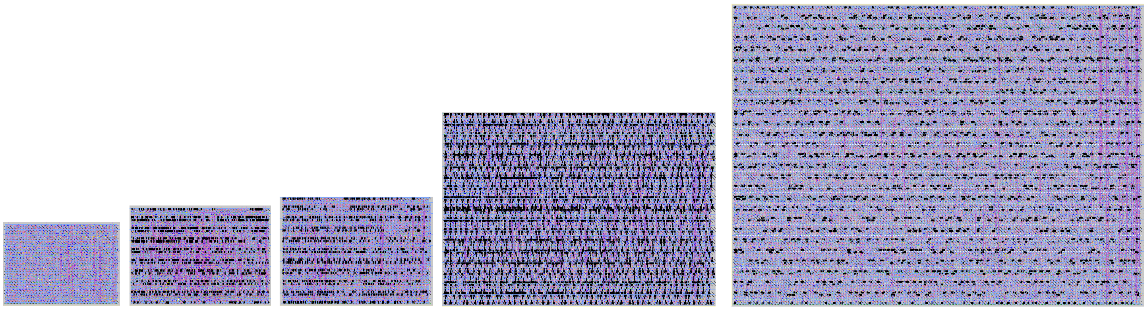


Figure 5.12: Layout size comparison of different SBOX implementations. From left to the right: singleRail, dualRailAS, dualRail, pDualRail (proposed), secLibDualRail (optimized)

The disadvantage of proposed gates is that they utilize two metal layers compared to a single metal layer used by simple library cells. The complexity of the routing inside the proposed cells is closer to, e.g. XOR gate than to AND or OR gates.

Yet, our results in Table 5.2 should be considered pessimistic. The used open-source QRouter is not a state-of-the-art router: it provides significantly worse results in complex designs than up-to-date commercial alternatives[3]. In our case, the router has problems with dense local interconnect. In the SBOX variants denoted pDualRail and secLibDualRail, we have to add increased cell spacing for successful routing.

Even if QRouter is not able to route densely placed designs, state-of-the-art routers could succeed. The number of failed nets for dense placement is low and the manual layout inspection has shown that there is room to finish the routing job. Therefore, we report the dense layout area as the optimistic data in Table 5.2.

Figure 5.12 is provided for illustartion only – it shows the pesimistic results reported in Table 5.2: preview of routed layouts of the five SBOX implementations.

As reported in Table 5.2, the circuit variant pDualRail, which is composed of proposed standard cells, brings only a small delay penalty compared to the area-efficient circuit variants and has a lower delay, and introduces a significantly smaller area overhead than the SecLib-based secLibDualRail circuit variant.

Our approach is a masking approach decreasing the measurement SNR [84] by decreasing the variability in the data-dependent current component. The simulation results presented in Figure 5.13 show the variability of the data-dependent current at the time when the circuit is illuminated (in the point-of-interest). We evaluated different SBOX implementations employing different kinds of countermeasures. To visualize the data-dependent current variability, we use the statistical *probability density function* (PDF). Figures 5.13a – 5.13d show the power imprint variability for selected illumination powers for all circuit variants, while Figure 5.13f shows the power imprint variability for the dynamic power, and Figure 5.13e for the static power consumption (subthreshold leakage only was included). The variability of the power imprint is directly connected with circuit

[3] See the maintainers' note in [35]

vulnerability: more variability in power traces decreases the attack cost. More variability in power traces also enables a successful attack to be performed by a less sophisticated attacker: the number of power traces required for a successful attack is lower, or simpler equipment might be used to obtain the power trace set of the required quality.

The routing procedure used is not able to balance complementary signals in dual-rail implementations. This does not significantly affect static circuit behavior, which is the focus of our research. In practice, dynamic-power imbalances in dual-rail implementations can be further reduced by careful routing. For comparison, the dynamic behavior of all circuit variants is presented.

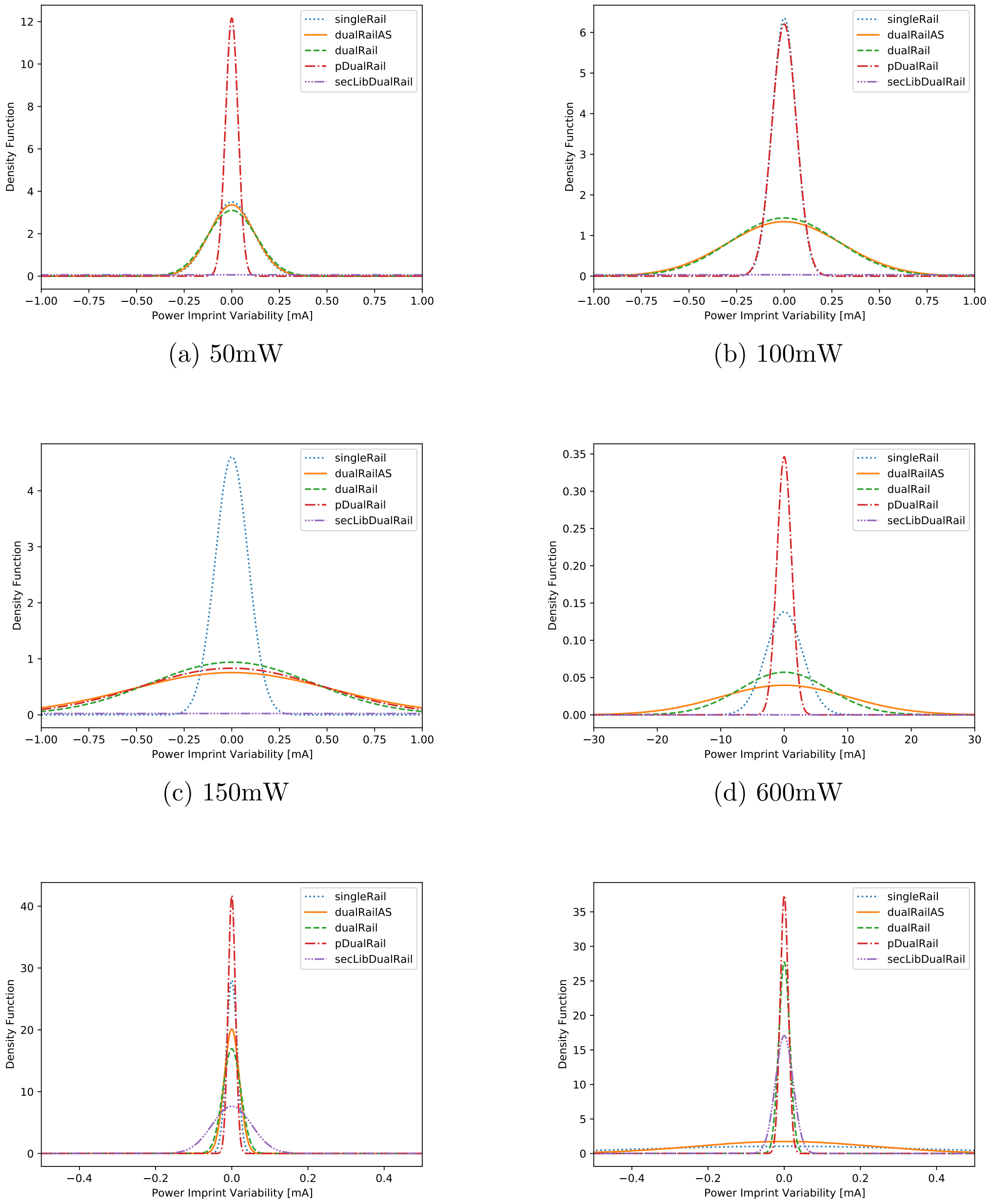
The simulation results show that our dualRailAS circuit version is significantly worse in dynamic power balancing compared to the other dual-rail implementations. In our comparison, we expect a more sophisticated attacker than the original dualRailAS authors. Nevertheless, our implementation still brings a little improvement compared to the single-rail circuit[4]. Figure 5.13f shows that dualRail, secLibDualRail and pDualRail implementations are balanced competitively from the dynamic power point of view. The implementation of the proposed pDualRail cells, however, offers much lower variability in power imprints induced by illumination at low illumination intensity.

Interestingly, the static power variability, presented in Figure 5.13e, follows the size of the implementation – smaller circuits are less vulnerable – except for the proposed implementation. The proposed implementation offers the lowest variability thanks to the size reduction of the input-controlled parts of the CMOS stack.

Figure 5.13 represents a serious issue for state-of-the-art protected dual-rail implementations (dualRail, dualRailAS, and even secLibDualRail). By delivering 50mW to 100mW of equivalent power to the area of the protected SBOX, the variability of the power trace set is increased to the level observed for dynamic power of the unprotected implementation (singleRail).

Although the attack setup is more complex (it needs to control the clock or to synchronize illumination and power measurement), it can be effective. The static power trace set delivered has a variability comparable to the variability of the dynamic power traces obtained from an unprotected single rail implementation. From an alternate perspective, we observed that about an order of magnitude lower measurement resolution is required for an illumination attack as for a dynamic power attack on protected implementation to obtain the power trace set with a comparable variability. The simulation shows that the illumination attack has the potential to circumvent the dynamic power countermeasures based on balancing.

[4] dualRailAS circuit variant employs alternating spacer and it provides – in theory – the best dynamic power balancing when the attacker is only able to observe the integral dynamic power over following spacers. We considered only transition from the first (00) spacer to evaluation phase (as for the other dual-rail circuit variants), which disadvantages this version compared to the theoretical assumptions and other dual-rail variants – see [111] for details



(a) 50mW

(b) 100mW

(c) 150mW

(d) 600mW

(e) Static Power

(f) Dynamic power

Figure 5.13: Selected density functions (PDF) for power imprints of all implementations - a narrower curve means a better protection. The proposed implementation overcomes its competitors significantly, except the *transient region*, where the results are comparable and smaller circuit size is an advantage

5.6 Discussion

The proposed structures, in general, affect the standard cell size and performance. On the other hand, only some of the approaches described in Section 5.3 may be employed to find out the trade-off between attack resistance and design cost – e.g. a smaller cell may be designed for lower illumination energy balancing only.

The advantage of the protections presented in Section 5.3 is that the protection mechanisms exploit natural properties of the CMOS technology, and thus all added transistor structures may be constructed accordingly to the original gate transistors – no process tuning is required. Although doping changes may increase the sensitivity of light sensors or increase/decrease the conductivity of added parts, good performance may be obtained by tuning transistor sizes only. This fact may simplify the protection mechanisms adoption.

Note that the light-sensitive structure may be shared between several standard cells to decrease the area overhead; however, any light-sensitive structure must be placed close to the protected structures to ensure that they will be exposed to the same light intensity in case of light-attack; in case of shared light-sensitive structures, considering bigger, while more sensitive structures [75] is possible.

The advantage is that *inverter balancing* may be applied using standard cells only, without the need for custom CMOS cell design. This increases the practical impact of this balancing technique.

All of the presented approaches increase the data independence of the induced OBIC, but the inverter balancing approach and the size reduction of the input-controlled transistors in a protected gate reduce the data-dependent part of the leakage significantly.

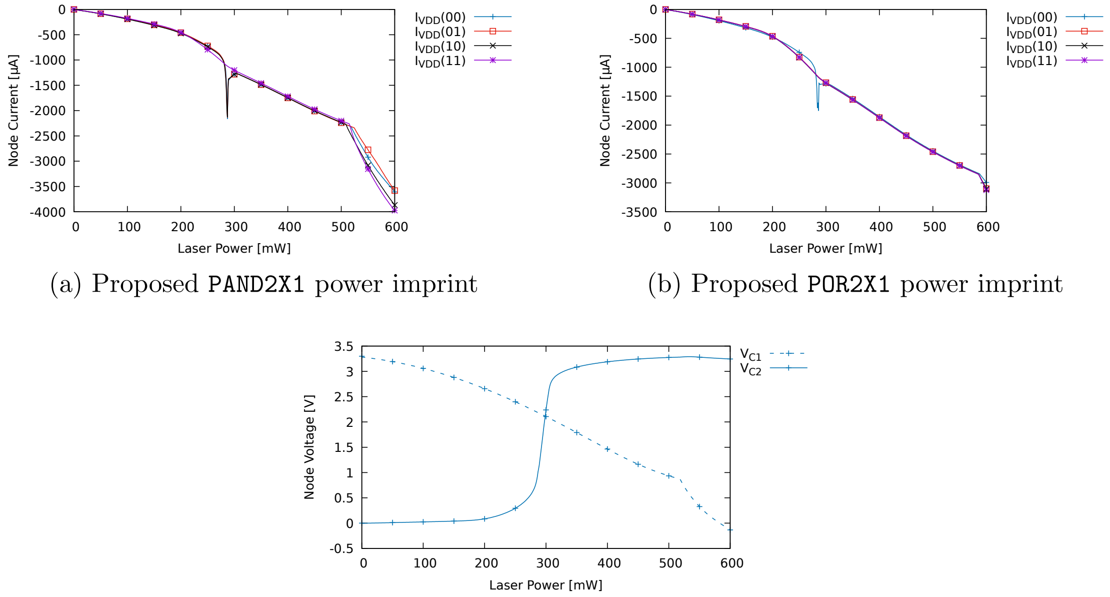
The performance degradation and the introduced area overhead are much smaller compared to the best static CMOS alternative, which is up today the SecLib [A.7].

As the proposed standard cells operate in two main regions (the operational and the constant value output), we believe that the whole circuit may be illuminated only by lower light intensities – in the operational region. Inducing a great current in a wider area would lead to circuit destruction: the cells out of the operational region are de-facto shorted and only short light pulses guarantee that the circuit will survive. Additionally, the presented experimental results provide only the data-dependent part of the light-induced current – the light attack induces also data-independent current contributing to possible circuit destruction.

The disadvantage of our approach is that it can potentially simplify the fault-injection attacks. As the proposed cell output is a constant fixed value outside the operational region, it can be used to induce a fixed-value fault into a combinational circuit. This may lead to glitches or even register value changes depending on the attack timing. Even this kind of attack requires a sophisticated setup and it is possible to induce a fixed value fault to a selected location even with classical CMOS cells [66, 107, 108], the proposed cells make the attack simpler, however, sophisticated equipment is still required. However, inducing random faults by illumination may also be simplified; at least, the attacker can use the fact that the probability that the induced fault is (close to) 1. This kind of attack in general requires system-level countermeasures.

The proposed structure is most vulnerable – power imprint imbalances occur – when the illumination intensity is in the transient region, which is given by the supply voltage and illumination intensity.

Higher supply voltage may also increase imbalances in the transient region, as it affects the slope of the first control signal produced by the light-sensitive inverter – see Figure 5.14. A possible solution of this issue is adding other inverters into the inverter chain generating the control signals, to correct the control signals slope. Two inverters are recommended, as better results are obtained only if the first control signal precedes the second control signal. This simple approach helps with narrowing the transient region, however, the imbalance is still present. As the other source of imbalances in a bigger circuit composed of proposed cells, we identified the voltage drops at the gate inputs deep in the illuminated circuit. The proposed standard cells were carefully designed to provide almost perfectly constant power imprint also under-voltage drops at the cell inputs, however, induced imbalances may still occur, especially in the transient region.



(a) Proposed `PAND2X1` power imprint

(b) Proposed `POR2X1` power imprint

(c) Control signals for `PAND2X1`

Figure 5.14: `PAND2X1` and `POR2X1` power imprints and control signals in the point-of-interest under the increased supply voltage

We have also a surprising result connected with the secLibDualRail circuit version. We originally expected that the secLibDualRail will provide the best protection, however, the sensitive region of the secLibDualRail circuit, which represents the standard SecLib approach, is in the lowest illumination power region. This fact, connected with the huge area of the SecLib implementation, potentially enables a simpler attack to be performed, as bigger structures naturally lead to increased data-dependent variances observable in the power trace.

Chapter 6

Conclusions

Testability, reliability, and security aspects of CMOS circuit design are important and extensively researched areas. A new approach in design for testability is the first of two main contributions of this dissertation thesis, while a novel static power-related vulnerability with a proposed countermeasure is the other. This dissertation thesis also touches the security-reliability interplay and points to newly discovered issues with state-of-the-art dynamic power attack countermeasures.

Topics and goals of this dissertation thesis were introduced in Chapter 1. Chapter 2 introduced the reader to the theoretical background in the research areas touched in this dissertation thesis, and provided a state-of-the-art overview.

Chapter 3 provided an overview of the state-of-the-art closely related to the proposed design for testability approach and described the proposed short-duration offline test, and proposed the Time-Extended Duplex (TED) as an alternative to Tripple-Modular Redundancy (TMR).

The state-of-the-art closely related to side-channel analysis was provided in Chapter 4. This chapter included the novel static power-related vulnerability analysis and it is concluded by the description of the proposed attacks.

Chapter 5 provided an overview of design styles providing a certain level of intrinsic immunity concerning the newly described CMOS vulnerability. Then, structures enabling the design of the attack-resistant cells were described and proposed cells were evaluated. The Chapter was concluded by a case study on the AES SBOX combinational logic block showing, how the proposed approach overcomes the current dynamic power state-of-the-art balancing approaches.

6.1 Summary

A new approach to design for testability, called short-duration offline test, was proposed and its impact was evaluated by incorporating into the Time-Extended Duplex concept, a potential alternative to the Tripple-Modular Redundancy.

A novel vulnerability related to the CMOS static power was analyzed and its impact on several design methods was evaluated. It has been shown that the state-of-the-art dual-rail-based balancing countermeasures are ineffective against the presented vulnerability, and a new issue related to the SecLib approach was identified. A novel standard cell design method was proposed as a countermeasure and its applicability was confirmed by simulation and the case study.

All results were presented and discussed in the scientific community. In the first place, this work was published in proceedings of five international conferences, and several workshops and local events. The article concluding the first part of this dissertation thesis (Chapter 3) was presented in the journal on Microprocessors and Microsystems, while the article concluding the second part of this dissertation thesis (Chapters 4 and 5) was presented in the journal on Microelectronics Reliability. The proposed CMOS structures described in Chapter 5 are subject to the patent application, while the national patent [A.12] was already confirmed.

6.2 Contributions of the Dissertation Thesis

- A short-duration offline test: The proposed fast offline test may be incorporated into the normal computation flow and potentially replace the online test in many cases, while reducing delay and area penalty at the same time. The short-duration offline test is enabled by proposed CMOS structures. For details, see Section 3.3.

- A method for designing a system with increased reliability incorporating the proposed short-duration offline-test: A Time-extended Duplex (TED) system concept was described and evaluated. For details, see Section 3.4 and Appendix A.

- A novel CMOS design threat: an attack combining combinational logic illumination and static power measurement was described, and its severity was proved by simulation. The threat arises especially in redundant structures like voters, but it also endangers dynamic power countermeasures based on balancing, including SecLib or WDDL. For details, see Sections 4.3 and 4.4.

- CMOS circuit-level (standard-cell level) attack countermeasures: the proposed standard cells may be used as a direct replacement of conventional CMOS cells in the common design process. Properties of proposed cells were shown by simulation and a case study on the AES SBOX design was provided. For details, see Sections 5.3, 5.4, 5.5, and 5.6.

6.3 Future Work

The author of the dissertation thesis suggests to explore the following:

- It would be interesting to investigate the properties of the monotonic circuit transformation allowing 100% stuck-at-fault, or stuck-open/stuck-on fault coverage, which is based on added static gates, not custom dynamic gates. The proposed approach is based on a custom domino logic cell.

- It is necessary to validate the proposed attacks and countermeasures on a real hardware implementation to confirm simulations.

- It would be beneficial to study and evaluate the impact of illumination detector sharing as well as the impact of more sophisticated illumination detectors in combination with proposed protected CMOS cells.

- It would be interesting to continue with a study of the usability and practical impact of *selective aging* processes in CMOS: eg. the selective ionizing irradiation of sensitive CMOS (sub)circuits might have a permanent and severe impact on the circuits' static power and security analogous to the described vulnerability connected with (selective) circuit illumination.

Bibliography

[1] Emrah Acar, Anirudh Devgan, and Sani R Nassif. Leakage and Leakage Sensitivity Computation for Combinational Circuits. *Journal of Low Power Electronics*, 1(2): 172–181, 2005.

[2] J. M. Acken and S. D. Millman. Fault model evolution for diagnosis: Accuracy vs precision. In *Custom Integrated Circuits Conference, 1992., Proceedings of the IEEE 1992*, pages 13.4.1–13.4.4, May 1992.

[3] Tutu Ajayi and David et. al Blaauw. Openroad: Toward a self-driving, open-source digital layout implementation tool chain. In *Proceedings of Government Microcircuit Applications and Critical Technology Conference*, 2019.

[4] Mehdi-Laurent Akkar, Régis Bevan, Paul Dischamp, and Didier Moyart. Power Analysis, What Is Now Possible... In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 489–502. Springer, 2000.

[5] C. Albrecht. IWLS 2005 Benchmarks. Technical report, June 2005.

[6] M. Alioto, L. Giancane, G. Scotti, and A. Trifiletti. Leakage Power Analysis attacks: Well-defined procedure and first experimental results. In *2009 International Conference on Microelectronics - ICM*, pages 46–49, Dec 2009. ISSN 2159-1660.

[7] F. Amiel, K. Villegas, B. Feix, and L. Marcel. Passive and active combined attacks: Combining fault attacks and side channel analysis. In *Fault Diagnosis and Tolerance in Cryptography, 2007. FDTC 2007. Workshop on*, pages 92–102, Sept 2007.

[8] Inc. Analog Devices. LTSpice, 1999 – 2021. URL `https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html`.

[9] Semiconductor Industry Association. 2001 International Technology Roadmap for Semiconductors: Process Integration, Devices and Structures and Emerging Research Devices. 2001.

[10] S.P. Athan, D.L. Landis, and S.A. Al-Arian. A novel built-in current sensor for IDDQ testing of deep submicron CMOS ICs. In *Proceedings of 14th VLSI Test Symposium, 1996.*, pages 118–123, Apr 1996. ISSN 1093-0167.

[11] T. M. Austin. DIVA: a reliable substrate for deep submicron microarchitecture design. In *Microarchitecture, 1999. MICRO-32. Proceedings. 32nd Annual International Symposium on*, pages 196–207, 1999. ISSN 1072-4451.

[12] SkyWater PDK Authors. SkyWater SKY130 PDK's documentation, 2020 – 2021. URL `https://skywater-pdk.readthedocs.io/en/latest/`.

[13] Automotive Electronics Council: Mark A. Kelly, Jean Clarac, Brian Jendro, Hadi Mehrooz, Robert V. Knoell, et al. AEC-Q100: Failure Mechanism Based Stress Test Qualification For Integrated Circuits: AEC-Q100-007: Fault Simulation and Test Grading, 2007. URL `http://www.aecouncil.com/`. Rev-B.

[14] M. Balaz and S. Kristofik. Generic self repair architecture with multiple fault handling capability. In *Euromicro Conference on Digital System Design (DSD), 2015*, pages 197–204, Aug 2015.

[15] S. Bhasin, J. L. Danger, F. Flament, T. Graba, S. Guilley, Y. Mathieu, M. Nassar, L. Sauvage, and N. Selmane. Combined SCA and DFA Countermeasures Integrable in a FPGA Design Flow. In *2009 International Conference on Reconfigurable Computing and FPGAs*, pages 213–218, Dec 2009. ISBN 978-1-4244-5293-4. ISSN 2325-6532.

[16] J. Borecký. *Dependable Systems Design Methods for FPGAs*. PhD thesis, the Faculty of Information Technology, Czech Technical University in Prague, 8 2015.

[17] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In *International workshop on cryptographic hardware and embedded systems*, pages 16–29. Springer, 2004.

[18] J.A. Brzozowski and K. Raahemifar. Testing C-elements is not elementary. In *Proceedings of Second Working Conference on Asynchronous Design Methodologies, 1995*, pages 150–159, May 1995.

[19] Jan Bělohoubek. TSaCt2, 2015 – 2021. URL `https://github.com/DDD-FIT-CTU/TSaCt2`.

[20] Jan Bělohoubek. Photoelectric Laser Stimulation of Combinational Logic, 2019 – 2021. URL `https://github.com/DDD-FIT-CTU/CMOS-PLS/`.

[21] Jan Bělohoubek. Open CMOS SPICE Model Collections, 2019. URL `https://github.com/DDD-FIT-CTU/CMOS-SPICE-Model-Collections`.

[22] Inc. Cadence Design Systems. Custom IC/Analog/RF Design, 1988 – 2021. URL `https://www.cadence.com/en_US/home/tools/custom-ic-analog-rf-design.html`.

[23] K. A. Campbell, P. Vissa, D. Z. Pan, and D. Chen. High-level synthesis of error detecting cores through low-cost modulo-3 shadow datapaths. In *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6, June 2015. ISSN 0738-100X.

[24] Krishnendu Chakrabarty, Brian T Murray, and Vikram Iyengar. Built-in Test Pattern Generation For High-Performance Circuits Using Twisted-Ring Counters. In *Proceedings 17th IEEE VLSI Test Symposium (Cat. No. PR00146)*, pages 22–27. IEEE, 1999.

[25] Suresh Chari, Josyula R Rao, and Pankaj Rohatgi. Template attacks. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 13–28. Springer, 2002.

[26] Liming Chen and A. Avizienis. N-version programming: A fault-tolerance approach to reliability of software operation. In *The Twenty-Fifth International Symposium on Fault-Tolerant Computing, Highlights from Twenty-Five Years, 1995*, pages 113–, Jun 1995.

[27] D.R. Czajkowski, P.K. Samudrala, and M.P. Pagey. SEU mitigation for reconfigurable FPGAs. In *Aerospace Conference, 2006 IEEE*, pages 7 pp.–, 2006.

[28] Joan Daemen and Vincent Rijmen. The Rijndael block cipher: AES proposal. In *First candidate conference (AeS1)*, pages 343–348, 1999.

[29] Ilana David, Ran Ginosar, and Michael Yoeli. Self-timed is self-checking. *Journal of Electronic Testing*, 6(2):219–228, 1995. ISSN 0923-8174. URL http://dx.doi.org/10.1007/BF00993088.

[30] Al Davis and Steven M Nowick. An introduction to asynchronous circuit design. Technical report, Technical Report UUCS-97-013, September 1997.

[31] Albert Davis. Gnucap: The gnu circuit analysis package users manual, September 2006.

[32] Florian Echtler and Maximilian Häußler. Open Source, Open Science, and the Replication Crisis in HCI. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–8, 2018.

[33] R. Timothy Edwards. IRSIM, 2002 – 2020. URL http://opencircuitdesign.com/irsim/.

[34] R. Timothy Edwards. Magic VLSI, 2004 – 2021. URL http://opencircuitdesign.com/magic/.

[35] R. Timothy Edwards. Qrouter, 2011 – 2021. URL http://opencircuitdesign.com/qrouter/.

[36] R. Timothy Edwards. Qrouter, 2011 – 2021. URL `http://opencircuitdesign.com/qflow/`.

[37] R Timothy Edwards, Mohamed Shalan, and Mohamed Kassem. Real Silicon Using Open Source EDA. *IEEE Design & Test*, 2021.

[38] Efabless.com. Efabless.com, 2014 – 2021. URL `https://www.efabless.com`.

[39] Efabless.com. OpenLANE Project, 2019 – 2021. URL `https://github.com/efabless/openlane`.

[40] John M Emmert, Charles E Stroud, and James R Bailey. A new bridging fault model for more accurate fault behavior. In *Proc. of the Design Automation Conf*, pages 481–485, 2000.

[41] Hirohiko Endoh and Takuya Naoe. Copper Wire Bonding Package Decapsulation Using the Anodic Protection Method. *Microelectronics Reliability*, 55(1):207–212, 2015. ISSN 0026-2714.

[42] Bijan Fadaeinia, Thorben Moos, and Amir Moradi. BSPL: Balanced Static Power Logic. *IACR Cryptol. ePrint Arch.*, 2020:558, 2020.

[43] P. Fiser, P. Kubalik, and H. Kubatova. An efficient multiple-parity generator design for on-line testing on fpga. In *11th EUROMICRO Conference on Digital System Design Architectures, Methods and Tools, 2008, DSD'08*, pages 96–99, Sept 2008.

[44] Inc. Free Software Foundation. GNU Electric, 1983 – 2017. URL `http://www.gnu.org/software/electric/`.

[45] J. Galiay, Y. Crouzet, and M. Vergniault. Physical versus logical fault models MOS LSI circuits: Impact on their testability. *IEEE Transactions on Computers*, C-29(6): 527–531, June 1980. ISSN 0018-9340.

[46] Ahmed Ghazy and Mohamed Shalan. Openlane: The open-source digital asic implementation flow. In *Workshop on Open-Source EDA Technology (WOSET)*, 2020.

[47] Jacopo Giorgetti, Giuseppe Scotti, Andrea Simonetti, and Alessandro Trifiletti. Analysis of data dependence of leakage current in CMOS cryptographic hardware. In *Proceedings of the 17th ACM Great Lakes symposium on VLSI*, pages 78–83. ACM, 2007.

[48] Symbiotic GmbH. Symbiotic EDA, 2019 – 2021. URL `https://www.symbioticeda.com`.

[49] Prabhakar Goel. An Implicit Enumeration Algorithm to Generate. *IEEE transactions on Computers*, 30(3), 1981.

[50] Graywolf contributors. Graywolf – a fork of TimberWolf 6.3.5, 2014 – 2021. URL `https://github.com/rubund/graywolf5`.

[51] S. Guilley, P. Hoogvorst, Y. Mathieu, R. Pacalet, and J. Provost. CMOS structures suitable for secured hardware. In *Proceedings Design, Automation and Test in Europe Conference and Exhibition*, volume 2, pages 1414–1415 Vol.2, Feb 2004. ISSN 1530-1591.

[52] Sylvain Guilley, Florent Flament, Yves Mathieu, and Renaud Pacalet. Security evaluation of a balanced quasi-delay insensitive library (seclib). In *Conference on Design of Circuits and Integrated Systems*, pages 6–pages, 2008.

[53] Sylvain Guilley, Laurent Sauvage, Jean-Luc Danger, Tarik Graba, and Yves Mathieu. Evaluation of Power-Constant Dual-Rail Logic as a Protection of Cryptographic Applications in FPGAs. In *2008 Second International Conference on Secure System Integration and Reliability Improvement*, pages 16–23. IEEE, 2008.

[54] Donald H Habing. The Use of Lasers to Simulate Radiation-Induced Transients in Semiconductor Devices and Circuits. *IEEE Transactions on Nuclear Science*, 12(5): 91–100, 1965.

[55] Basel Halak, Julian Murphy, and Alex Yakovlev. Power Balanced Circuits for Leakage-Power-Attacks Resilient Design. In *2015 Science and Information Conference (SAI)*, pages 1178–1183. IEEE, 2015.

[56] Koshi Haraguchi. Microscopic Optical Beam Induced Current Measurements and their Applications. In *Conference Proceedings. 10th Anniversary. IMTC/94. Advanced Technologies in I & M. 1994 IEEE Instrumentation and Measurement Technolgy Conference (Cat. No. 94CH3424-9)*, pages 693–699. IEEE, 1994.

[57] Charles F Hawkins, Jerry M Soden, Alan W Righter, and F Joel Ferguson. Defect Classes - An Overdue Paradigm for CMOS IC Testing. In *Proceedings., International Test Conference*, pages 413–425. IEEE, 1994.

[58] Domenik Helms, Eike Schmidt, and Wolfgang Nebel. Leakage in CMOS circuits – an introduction. In *International Workshop on Power and Timing Modeling, Optimization and Simulation*, pages 17–35. Springer, 2004.

[59] John L. Hennessy and David A. Patterson. *Computer Architecture: A Quantitative Approach, Fifth Edition*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 5th edition, 2011. ISBN 012383872X, 9780123838728.

[60] K. Heragu, J. H. Patel, and V. D. Agrawal. Segment Delay Faults: A New Fault Model. In *Proceedings of 14th VLSI Test Symposium*, pages 32–39, 1996.

[61] David D Hwang, Kris Tiri, Alireza Hodjat, B-C Lai, Shenglin Yang, Patrick Schaumont, and Ingrid Verbauwhede. AES-Based Security Coprocessor IC in 0.18-umCMOS With Resistance to Differential Power Analysis Side-Channel Attacks. *IEEE Journal of Solid-State Circuits*, 41(4):781–792, 2006.

[62] IEEE. IEEE Standard for Test Access Port and Boundary-Scan Architecture – Redline. *IEEE Std 1149.1-2013 (Revision of IEEE Std 1149.1-2001) - Redline*, pages 1–899, 2013.

[63] Accellera Systems Initiative. Verilog-AMS Language Reference Manual, May 2014. URL `https://www.accellera.org/images/downloads/standards/v-ams/VAMS-LRM-2-4.pdf`.

[64] Nanoscale Integration and ASU Modeling (NIMO) Group. Predictive Technology Model (PTM), 2001 – 2012. URL `http://ptm.asu.edu//`.

[65] Najeh Kamoun, Lilian Bossuet, and Adel Ghazel. Correlated Power Noise Generator as a Low Cost DPA Countermeasures to Secure Hardware AES Cipher. In *2009 3rd International Conference on Signals, Circuits and Systems (SCS)*, pages 1–6. IEEE, 2009.

[66] D. Karaklajić, J. Schmidt, and I. Verbauwhede. Hardware Designer's Guide to Fault Attacks. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 21 (12):2295–2306, Dec 2013. ISSN 1063-8210.

[67] Nam Sung Kim, Todd Austin, David Baauw, Trevor Mudge, Krisztián Flautner, Jie S Hu, Mary Jane Irwin, Mahmut Kandemir, and Vijaykrishnan Narayanan. Leakage current: Moore's Law Meets Static Power. *computer*, 36(12):68–75, 2003.

[68] T. Koal, M. Scholzel, and H.T. Vierhaus. Combining fault tolerance and self repair at minimum cost in power and hardware. In *17th International Symposium on Design and Diagnostics of Electronic Circuits Systems*, pages 153–158, April 2014.

[69] Paul Kocher, Jann Horn, Anders Fogh, Daniel Genkin, Daniel Gruss, Werner Haas, Mike Hamburg, Moritz Lipp, Stefan Mangard, Thomas Prescher, et al. Spectre Attacks: Exploiting Speculative Execution. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 1–19. IEEE, 2019.

[70] Israel Koren and C. Mani Krishna. *Fault-Tolerant Systems*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2007. ISBN 0120885255, 9780080492681.

[71] Volkan Kursun and Eby G Friedman. Domino Logic With Variable Threshold Voltage Keeper. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 11(6): 1080–1093, 2003.

[72] Dongwoo Lee, Wesley Kwong, David Blaauw, and Dennis Sylvester. Analysis and Minimization Techniques for Total Leakage Considering Gate Oxide Leakage. In *Proceedings of the 40th annual Design Automation Conference*, pages 175–180, 2003.

[73] R Llido, A Sarafianos, O Gagliano, V Serradeil, V Goubier, M Lisart, G Haller, Vincent Pouget, Dean Lewis, Jean-Max Dutertre, et al. Characterization and TCAD simulation of 90 nm technology transistors under continous photoelectric laser stimulation for failure analysis improvement. In *19th IEEE International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA) 2012*, pages 1–6. IEEE, 2012.

[74] S. Ma, I. Shaik, and R. S. Fetherston. A comparison of bridging fault simulation methods. In *Proceedings of International Test Conference, 1999.*, pages 587–595, 1999. ISSN 1089-3539.

[75] Marinet, Fabrice and Fort, Jimmy and Sarafianos, Alexandre and Mercier, Julien. Device for Detecting a Laser Attack in an Integrated Circuit Chip, December 9 2014.

[76] Edward J. McCluskey and Chao-Wen Tseng. Stuck-fault tests vs. actual defects. In *Proceedings of the 2000 IEEE International Test Conference*, ITC '00, pages 336–. IEEE Computer Society, Washington, DC, USA, 2000. ISBN 0-7803-6546-1.

[77] K. C. Y. Mei. Bridging and stuck-at faults. *IEEE Transactions on Computers*, C-23 (7):720–727, July 1974. ISSN 0018-9340.

[78] IPL Alliance Members. Interoperable PDK Libraries (IPL), 2007 – 2021. URL `https://www.iplnow.com`.

[79] Thomas S Messerges. Using Second-Order Power Analysis to Attack DPA Resistant Software. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 238–251. Springer, 2000.

[80] Thomas S Messerges, Ezzy A Dabbish, and Robert H Sloan. Investigations of Power Analysis Attacks on Smartcards. *Smartcard*, 99:151–161, 1999.

[81] S. D. Millman and E. J. McCluskey. Detecting bridging faults with stuck-at test sets. In *Proceedings the 1988 International Test Conference, New Frontiers in Testing*, pages 773–783, Sep 1988. ISSN 1089-3539.

[82] A Mishchenko et al. ABC: A system for sequential synthesis and verification. `http://www.eecs.berkeley.edu/~alanmi/abc`, 2012.

[83] Alan Mishchenko. ABC: A System for Sequential Synthesis and Verification, 2005 – 2021. URL `https://people.eecs.berkeley.edu/~alanmi/abc/`.

[84] Thorben Moos, Amir Moradi, and Bastian Richter. Static Power Side-Channel Analysis – An Investigation of Measurement Factors. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2019.

[85] Farshad Moradi, Tuan Vu Cao, Elena I Vatajelu, Ali Peiravi, Hamid Mahmoodi, and Dag T Wisland. Domino logic designs for high-performance and leakage-tolerant applications. *Integration*, 46(3):247–254, 2013.

[86] K. S. Morgan, D. L. McMurtrey, B. H. Pratt, and M. J. Wirthlin. A Comparison of TMR With Alternative Fault-Tolerant Design Techniques for FPGAs. *IEEE Transactions on Nuclear Science*, 54(6):2065–2072, 2007.

[87] Mosis. MOSIS Scalable CMOS (SCMOS), 2018. URL `https://web.archive.org/web/20190921192303/https://www.mosis.com/files/scmos/scmos.pdf`. rev. 8.0.

[88] Julian Murphy and Alex Yakovlev. An Alternating Spacer AES Crypto-processor. In *2006 Proceedings of the 32nd European Solid-State Circuits Conference*, pages 126–129, 2006.

[89] Laurence William Nagel. SPICE – Simulation Program With Integrated Circuit Emphasis. *Memo No.. ERL-M382, Electronics Research Laboratory, Univ. of California, Berkeley*, 1973.

[90] Ondrej Novak and H Nosek. Test-per-Clock Testing of the Circuits with Scan. In *Proceedings Seventh International On-Line Testing Workshop*, pages 90–92. IEEE, 2001.

[91] Oklahoma State University (OSU). MOSIS SCMOS: Standard Cells for AMI 0.6um, AMI 0.35um, TSMC 0.25um, and TSMC 0.18um, 1999 – 2016. URL `https://vlsiarch.ecen.okstate.edu/flows/MOSIS_SCMOS`.

[92] P. Kocher, J. Jaffe and B. Jun. Introduction to Differential Power Analysis and Related Attacks, 1998. URL `http://www.cryptography.com/dpa/`. technical report.

[93] P. Kocher, J. Jaffe and B. Jun. Differential power analysis. In *Annual International Cryptology Conference*, pages 388–397. Springer, 1999.

[94] R. Possamai Bastos, J.-M. Dutertre, and F. Sill Torres. Comparison of bulk built-in current sensors in terms of transient-fault detection sensitivity. In *CMOS Variability (VARI), 2014 5th European Workshop on*, pages 1–6, Sept 2014.

[95] D. K. Pradhan and J. J. Stiffler. Error-Correcting Codes and Self-Checking Circuits. *Computer*, 13(3):27–37, March 1980. ISSN 0018-9162.

[96] GHDL Project. GHDL, 2002 – 2021. URL `http://ghdl.free.fr/`.

[97] KLayout Project. KLayout, 2006 – 2021. URL `https://klayout.de/`.

[98] Francesco Regazzoni, Luca Breveglieri, Paolo Ienne, and Israel Koren. Interaction between fault attack countermeasures and the resistance against power analysis attacks. In *Fault Analysis in Cryptography*, pages 257–272. Springer, 2012.

[99] Marc Renaudin, P. Bastos, Rodrigo, G Sicard, and F. Kastensmidt. Asynchronous circuits as alternative for mitigation of long-duration transient faults in deep-submicron technologies. *Microelectronics Reliability*, 50(9–11):1241–1246, 09 2010. 21st European Symposium on the Reliability of Electron Devices, Failure Physics and Analysis.

[100] C. Roscian, A. Sarafianos, J. Dutertre, and A. Tria. Fault Model Analysis of Laser-Induced Faults in SRAM Memory Cells. In *2013 Workshop on Fault Diagnosis and Tolerance in Cryptography*, pages 89–98, Aug 2013.

[101] Paul Rosinger, Bashir Al-Hashimi, and Krishnendu Chakrabarty. Rapid generation of thermal-safe test schedules. In *Design, Automation and Test in Europe*, pages 840–845. IEEE, 2005.

[102] Jian Ruan, Zhiying Wang, Kui Dai, and Yong Li. Design and test of self-checking asynchronous control circuit. In *Integrated Circuit and System Design. Power and Timing Modeling, Optimization and Simulation*, volume 4644 of *Lecture Notes in Computer Science*, pages 320–329. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-74441-2.

[103] P.K. Samudrala, J. Ramos, and S. Katkoori. Selective triple Modular redundancy (STMR) based single-event upset (SEU) tolerant synthesis for FPGAs. *IEEE Transactions on Nuclear Science*, 51(5):2957–2969, Oct 2004. ISSN 0018-9499.

[104] A. Sarafianos, O. Gagliano, M. Lisart, V. Serradeil, J. Dutertre, and A. Tria. Building the electrical model of the pulsed photoelectric laser stimulation of a PMOS transistor in 90nm technology. In *Proceedings of the 20th IEEE International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA)*, pages 22–27, July 2013. ISSN 1946-1550.

[105] A. Sarafianos, R. Llido, O. Gagliano, V. Serradeil, Mathieu Lisart, et al. Building the electrical model of the pulsed photoelectric laser stimulation of an NMOS transistor in 90nm technology. In *38th International Symposium for Testing and Failure Analysis, (ISTFA) 2012*, pages 5B–5, 2012.

[106] Alexandre Sarafianos. *Injection de fautes par impulsion laser dans des circuits sécurisés*. PhD thesis, Saint-Etienne, EMSE, 2013.

[107] Alexandre Sarafianos, Olivier Gagliano, Valérie Serradeil, Mathieu Lisart, Jean-Max Dutertre, and Assia Tria. Building the electrical model of the pulsed photoelectric laser stimulation of an NMOS transistor in 90nm technology. In *IEEE International Reliability Physics Symposium (IRPS), 2013*, pages 5B–5. IEEE, 2013.

[108] Sergei P Skorobogatov and Ross J Anderson. Optical fault induction attacks. In *International workshop on cryptographic hardware and embedded systems*, pages 2–12. Springer, 2002.

[109] Hendrawan Soeleman, Kaushik Roy, and Bipul Paul. Sub-Domino Logic: Ultra-Low Power Dynamic Sub-Threshold Digital Logic. In *VLSI Design 2001. Fourteenth International Conference on VLSI Design*, pages 211–214. IEEE, 2001.

[110] Siemens Digital Industries Software. Calibre Design Solutions, 2021. URL `https://eda.sw.siemens.com/en-US/ic/calibre-design/`.

[111] Danil Sokolov, Julian Murphy, Alexander Bystrov, and Alexandre Yakovlev. Design and analysis of dual-rail circuits for security applications. *IEEE Transactions on Computers*, 54(4):449–460, 2005.

[112] Jens Sparsø and Steve Furber. *Principles of Asynchronous Circuit Design: A Systems Perspective*. Kluwer Academic Publishers, Boston, 1st edition, 2001. ISBN 0-7923-7613-7.

[113] P. Srivastava, A. Pua, and L. Welch. Issues in the Design of Domino Logic Circuits. In *Proceedings of the 8th Great Lakes Symposium on VLSI (Cat. No.98TB100222)*, pages 108–112, 1998.

[114] François-Xavier Standaert. Introduction to Side-Channel Attacks. In *Secure integrated circuits and systems*, pages 27–42. Springer, 2010.

[115] Victoria Stodden, Jonathan Borwein, and David H Bailey. Setting the Default to Reproducible. *computational science research. SIAM News*, 46(5):4–6, 2013.

[116] Inc. Synopsys. Synopsys Custom Design Platform, 1986 – 2021. URL `https://www.synopsys.com/implementation-and-signoff/custom-design-platform.html`.

[117] Inc. Synopsys. Europractice : MPW Prototyping : ASICS, 1995 – 2021. URL `https://europractice-ic.com/mpw-prototyping/asics/`.

[118] Y. Tamir and M. Tremblay. High-performance fault-tolerant VLSI systems using micro rollback. *IEEE Transactions on Computers*, 39(4):548–554, Apr 1990. ISSN 0018-9340.

[119] Andrew S Tanenbaum. *Structured computer organization*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 5th edition, 2016.

[120] Kris Tiri, Moonmoon Akmal, and Ingrid Verbauwhede. A Dynamic and Differential CMOS Logic with Signal Independent Power Consumption to Withstand Differential Power Analysis on Smart Cards. In *Proceedings of the 28th European solid-state circuits conference*, pages 403–406. IEEE, 2002.

[121] Kris Tiri and Ingrid Verbauwhede. A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation. In *Proceedings Design, Automation and Test in Europe Conference and Exhibition*, volume 1, pages 246–251. IEEE, 2004.

[122] Arisona State University. ASAP7: 7-nm Predictive PDK, 2016 – 2021. URL `http://asap.asu.edu/asap/`.

[123] North Carolina State University. FreePDK, 2008 – 2021. URL `https://www.eda.ncsu.edu/wiki/FreePDK`.

[124] UC San Diego VLSI CAD Laboratory. OpenROAD Project, 2018 – 2021. URL `https://theopenroadproject.org/`.

[125] Holger Vogt, Marcel Hendrix, Paolo Nenzi, and Dietmar Warning. Ngspice users manual version 34, 2021.

[126] VTR Developers. Verilog to Routing (VTR) Project, 2012 – 2021. URL `https://github.com/verilog-to-routing/vtr-verilog-to-routing`.

[127] Laung-Terng Wang, Cheng-Wen Wu, and Xiaoqing Wen. *VLSI Test Principles and Architectures: Design for Testability (Systems on Silicon)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2006. ISBN 0123705975.

[128] Seongmoon Wang and Sandeep K Gupta. ATPG for Heat Dissipation Minimization During Test Application. In *Proceedings., International Test Conference*, pages 250–258. IEEE, 1994.

[129] C. Weaver and T. Austin. A fault tolerant approach to microprocessor design. In *International Conference on Dependable Systems and Networks, DSN 2001, 2001*, pages 411–420, July 2001.

[130] Liu Weidong, Jin Xiaodong, Xi Xuemei, et al. BSIM3v3.3 MOSFET model users' manual. *Berkeley, CA: The Regents of the University of California*, 2005.

[131] Neil Weste and David Harris. *CMOS VLSI Design: A Circuits and Systems Perspective*. Addison-Wesley Publishing Company, USA, 4th edition, 2010. ISBN 0321547748, 9780321547743.

[132] Michael John Yates Williams and James B Angell. Enhancing Testability of Large-Scale Integrated Circuits via Test Points and Additional Logic. *IEEE Transactions on Computers*, 100(1):46–60, 1973.

[133] Stephen Williams. Icarus Verilog, 1998 – 2020. URL `http://iverilog.icarus.com/`.

[134] Claire Wolf. Yosys Open SYnthesis Suite, 2012 – 2021. URL `http://www.clifford.at/yosys/`.

[135] Bo Yang, Kaijie Wu, and Ramesh Karri. Scan Based Side Channel Attack on Dedicated Hardware Implementations of Data Encryption Standard. In *2004 International Conferce on Test*, pages 339–344. IEEE, 2004.

[136] Yibin Ye, Shekhar Borkar, and Vivek De. A New Technique for Standby Leakage Reduction in High-Performance Circuits. In *1998 Symposium on VLSI Circuits. Digest of Technical Papers (Cat. No. 98CH36215)*, pages 40–41. IEEE, 1998.

[137] YongBin Zhou and DengGuo Feng. Side-channel attacks: Ten years after its publication and the impacts on cryptographic module security testing. *IACR Cryptology ePrint Archive*, 2005:388, 2005.

Reviewed Publications of the Author Relevant to the Thesis

[A.1] J. Bělohoubek, P. Fišer and J. Schmidt *Error Masking Method Based On The Short-Duration Offline Test.* Microprocessors and Microsystems (MICPRO), Elsevier, vol. 52, pp. 236-250, ISSN: 0141-9331, July 2017.

The article has been cited in:

○ R. Panek, J. Lojda, J. Podivinsky, and Z. Kotasek *Partial Dynamic Reconfiguration in an FPGA-based Fault-Tolerant System: Simulation-based Evaluation*, IEEE East-West Design & Test Symposium (EWDTS) 2018, Kazan, Russian Federation, 2018.

○ Arista Networks Inc. *Logic Buffer for Hitless Single Event Upset Handling.* Inventors: D. A. Cananzi, E. B. Van Hartingsveldt, M. Romain. U.S. Patent No 10,997,011 B2, 2021.

[A.2] J. Bělohoubek, P. Fišer and J. Schmidt *Optically Induced Static Power in Combinational Logic: Vulnerabilities and Countermeasures.* Microelectronics Reliability, Elsevier, vol. 124, ISSN: 0026-2714, September 2021.

[A.3] J. Bělohoubek, P. Fišer and J. Schmidt *Novel C-Element Based Error Detection and Correction Method Combining Time and Area Redundancy.* Euromicro Conference on Digital System Design (DSD), 2015, Funchal, Madeira – Portugal, 2015.

The paper has been cited in:

○ J.-P. Anderson *Duplicate with Choose: Using Statistics for Fault Mitigation* Dissertation, Brigham Young University, BYU Scholars Archive, 2016.

[A.4] J. Bělohoubek, P. Fišer and J. Schmidt *Error Correction Method Based On The Short-Duration Offline Test.* 2016 Euromicro Conference on Digital System Design (DSD), Limassol, Cyprus, 2016.

[A.5] J. Bělohoubek, P. Fišer and J. Schmidt *CMOS Illumination Discloses Processed Data.* 22nd Euromicro Conference on Digital Systems Design (DSD), Kallithea - Chalkidiki , Greece, 2019.

[A.6] J. Bělohoubek, P. Fišer and J. Schmidt *Using Voters May Lead to Secret Leakage.* 2019 22nd IEEE International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS), Cluj-Napoca, Romania, 2019.

[A.7] J. Bělohoubek, P. Fišer and J. Schmidt *Standard Cell Tuning Enables Data-Independent Static Power Consumption.* 23rd IEEE International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS), Novi Sad, Serbia, 2020.

The paper has been cited in:

- F. Bijan, T. Moos and A. Moradi *BSPL: Balanced Static Power Logic*, IACR Cryptology ePrint Archive, 2020.

[A.8] J. Bělohoubek *Novel Error Detection and Correction Method Combining Time and Area Redundancy.* Počítačové architektury a diagnostika 2015, Zlín, Czech Republic, 2015.

[A.9] J. Bělohoubek *Využití rychlého offline testu v systému se schopností maskování jedné chyby.* Počítačové architektury a diagnostika 2016, Kraví Hora - Bořetice, Czech Republic, 2016.

[A.10] J. Bělohoubek *Error Correction Method Based on the Efficient Offline Test.* A Doctoral Study Report submitted to the Faculty of Information Technology, Prague, Czech Republic, 2016.

[A.11] J. Bělohoubek *Zvyšování spolehlivosti a bezpečnosti číslicových obvodů na úrovni mikroarchitektury.* Počítačové architektury a diagnostika 2018, Churáňov, Czech Republic 2018.

Granted Patents of the Author Relevant to the Thesis

[A.12] Czech Technical University in Prague *Connection of a standard CMOS cell with reduced data dependence of static consumption.* Inventors: J. Bělohoubek, P. Fišer and J. Schmidt. Czech Republic. Patent No CZ 308895 B6, 2021.

Remaining Publications of the Author Relevant to the Thesis

[A.13] J. Bělohoubek and J. Schmidt *Fully asynchronous QDI implementation of DES in FPGA.* Cryptographic architectures embedded in reconfigurable devices (CryptArchi), Annency, France, 2014 (unpublished lecture).

[A.14] J. Bělohoubek *Novel gate design method for short-duration test.* POSTER 2015, Prague, Czech Republic, 2015.

[A.15] J. Bělohoubek *The Design-Time Side-Channel Information Leakage Estimation.* Cryptographic architectures embedded in reconfigurable devices (CryptArchi), Smolenice, Slovakia 2017 (unpublished lecture).

[A.16] J. Bělohoubek, P. Fišer and J. Schmidt *Effect of Power Trace Set Properties to Differential Power Analysis.* TRUDEVICE 2018, Dresden, Germany, 2018 (poster).

[A.17] J. Bělohoubek and R. Vik *Low-Cost CMOS Power Consumption Data Dependency Demonstrator Concept.* The 7th Prague Embedded Systems Workshop, Roztoky u Prahy, Czech Republic, 2019.

[A.18] J. Bělohoubek and J. Schmidt *CMOS Illumination Enables Observation of Processed Data in Power Traces.* Workshop on Practical Hardware Innovations in Security Implementation and Characterization (PHISIC), Gardanne, France, 2019 (poster).

[A.19] J. Bělohoubek *Modulated CMOS Static Power is Data Dependendent and Observable.* Cryptographic architectures embedded in reconfigurable devices (CryptArchi), Pruhonice, Czech Republic 2019 (unpublished lecture).

[A.20] J. Bělohoubek, P. Fišer and J. Schmidt *Standard Cell Design For Data-Independent Static Power Under Illumination.* The 9th Prague Embedded Systems Workshop, Roztoky u Prahy, Czech Republic, 2021.

Remaining Publications of the Author

[A.21] J. Bělohoubek, J. Čengery, J. Freisleben, P. Kašpar, A. Hamáček *KETCube – the Universal Prototyping IoT Platform.* 21st Euromicro Conference on Digital System Design (DSD), Prague, Czech Republic, 2018.

The paper has been cited in:

- Alsukayti, Ibrahim S. *An Internet-of-Things Educational Platform*, International Journal of Computer Science and Network Security (IJCSNS) 2019, Seoul, South Korea, 2019.
- S. Douglas, K. Gary and S. Sohoni *Impact of a Virtualized IoT Environment on Online Students*, IEEE Frontiers in Education Conference (FIE) 2020, Uppsala, Sweden, 2020.

[A.22] J. Bělohoubek *Smart re-use of hardware peripherals for better software UART.* The 3rd Prague Embedded Systems Workshop, Roztoky u Prahy, Czech Republic, 2015.

[A.23] J. Bělohoubek *KETCube – the Prototyping and Educational Platform for IoT Nodes.* The 6th Prague Embedded Systems Workshop, Roztoky u Prahy, Czech Republic, 2018.

[A.24] L. Menšík, R. Vik, S. Pretl, J. Bělohoubek, T. Syrový, L. Syrová, L. Kubáč, L. Menšík *Možnosti uplatnění internetu věcí (IoT) v precizním zemědělství v ČR.* Úroda 12/2019, pp. 341-350., ISSN: 0139-6013, 2019.

Appendix **A**

Proposed Offline Test

This Appendix provides an in-detail description of the Time-Extended Duplex (TED), which is conceptually described in Section 3.4. Figure A.1 shows a detailed overview of the TED, where all important implementation details are provided.

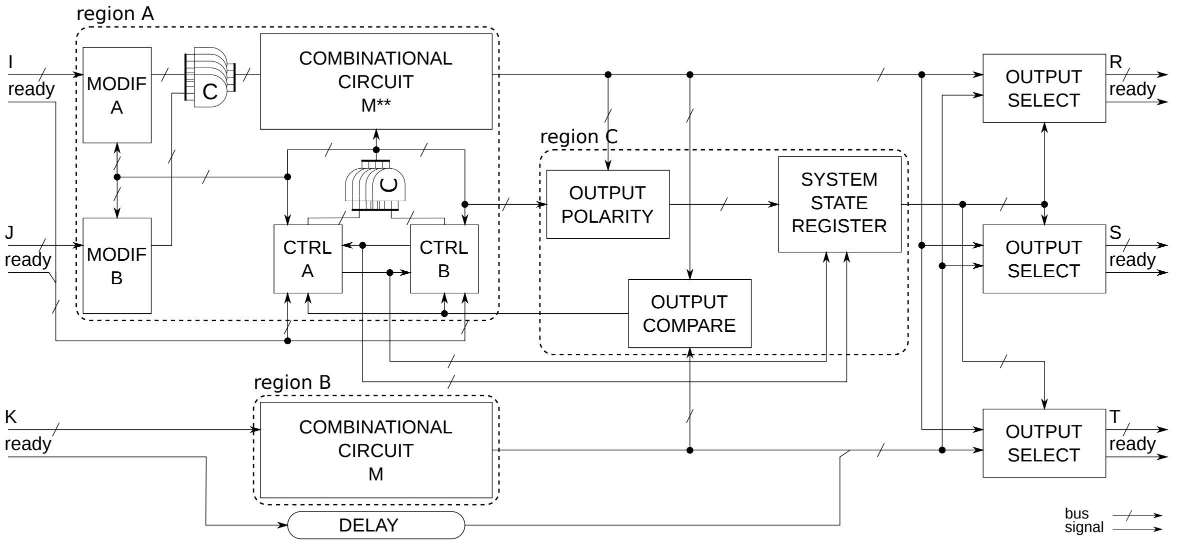


Figure A.1: Detailed Scheme of the Time-Extended Duplex

The TED arrangement in Figure A.1 ensures, that a fault in any block is detected. The offline test of the TED structure is executed in case of `M` and `M**` output mismatch. The test is optimized for the given structure and it is composed of several sub-tests. Each sub-test is designed to cover a set of faults in `M**` or errors at the outputs of other modules in TED.

M** Inputs Test

At the beginning of the test, the *inputs sub-test* is performed. MODIF B is used to propagate the output of MODIF A thru the C-elements. This is performed in two steps: the output of MODIF B is set to all-one – this propagates all ones from the MODIF A output to the C-elements output. Then the output of MODIF B is set to all-zero – this propagates zeroes from the MODIF A output to the C-elements output. After that, the output of M** is computed by using the MODIF A output only. The output is then compared with the M output.

The same steps are then repeated for the MODIF B output and the result is also compared with the output of M. If one of the two M** outputs matches the M output and the other does not match, erroneous MODIF has been detected. If no output matches with the M output, the test continues to the next sub-test.

M** Test

The main part of the test is a short-duration test of the module M**. The following sub-tests are performed level by level – the control signals of gates with the same gate level are joined to form a single control signal, driven by the test control logic. The *gate level* is defined as the maximal path length (number of gates) from the circuit primary inputs. The *circuit depth* is the maximum of the gate levels. The primary inputs are at level 0.

The term *primary input* is used in all sub-tests and refers to physical, not the logical circuit inputs. In the reduced dual-rail logic (Section 3.3.3), one circuit input is represented by one or two signals (primary inputs).

The test of M** is inspired by ideas described in Section 3.3 – the circuit is periodically flooded by a single value (1 and 0 alternate), and the flood propagation can be disrupted by faults. As this happens level-by-level, a fault in a lower level will cause the same fault symptom at higher levels. During the test, the control signals are used to excite and propagate the fault symptoms. This is the core idea of the short-duration test.

For example, if the gate preset to 0 is performed, then a stuck-open in an NMOS transistor of a gate at the first level will inhibit transition to 1, and thus cause that a zero value will occur at an input of a gate (configured as AND) at level two. This value – the fault symptom – is propagated up to the circuit outputs.

The proposed short-duration test of M** itself is divided into 3 sub-tests. Every sub-test is described in a dedicated table (Tables A.1, A.2 and A.3) as the sequence of iterations over the circuit levels. For every step of each sub-test, the values of control signals C, T_U, T_C and T_D are defined for each circuit level. The value of the gate output (signal O) is defined in the last column – arrows are used for transitions caused by the control signal setting ($0 \rightarrow 1$ or $1 \rightarrow 0$) in case of fault-free behavior.

The sub-test 1 (Table A.1) and the sub-test 3 (Table A.3) were designed to detect stuck-open faults and the sub-test 2 (Table A.2) to detect stuck-on faults. Additionally, the tests are able to detect some faults of the other type as a side-effect. Stuck-opens are generally relatively simple to detect because the gate is unable to change the output (the

step	C	T_U	T_C	T_D	O
1	set circuit primary inputs to 0				
2	start in level $i = 1$				
3	in all levels:				
	0	0	0	0	↓
4	in level i:				0
	1	1	1	1	↑
5	in level i:				1
	1	0	0	0	1
6	in levels other than i:				0
	1	0	1	0	0
7	set circuit primary inputs to 1				↑
8	Check if the circuit output is *all-one*				1
9	if (++i $\leq$ depth) then goto 3				1

Table A.1: The test sequence of the *sub-test 1*

step	C	T_U	T_C	T_D	O
1	set circuit primary inputs to 0				
2	1	1	1	1	↑
3	0	0	0	0	↓
4	start in level $i = 1$				
5	in all levels:				0
	1	0	0	0	0
6	in level i:				0
	1	0	1	1	0
7	in level i:				0
	1	1	1	0	0
8	in level i:				0
	1	1	0	1	0
9	if (++i $\leq$ depth) then goto 5				0
10	Check if the circuit output is *all-zero*				0

Table A.2: The test sequence of the *sub-test 2*

gate output retains its previous value). Every sub-test contains a cycle with the number of iterations equal to the circuit depth. A detailed example of sub-test 1 for a fault-free circuit is in Figure A.2 and for a faulty circuit in Figure A.3.

Table A.4 shows, which sub-test detects a stuck-open/stuck-on fault for a given transistor (see transistor labels in Figure 3.8).

In sub-test 1, the output is checked in every iteration because the precharge function of

step	C	$\mathtt{T_U}$	$\mathtt{T_C}$	$\mathtt{T_D}$	O
1	0	0	0	0	↓
2	set circuit primary inputs to 1				0
3	1	0	0	0	0
4	start in level $i = 1$				
5	in level i:				0
	1	0	1	0	↑
6	in level i:				1
	1	0	0	0	1
7	if (++i ≤ depth) then goto 5				1
8	Check if the circuit output is *all-one*				1

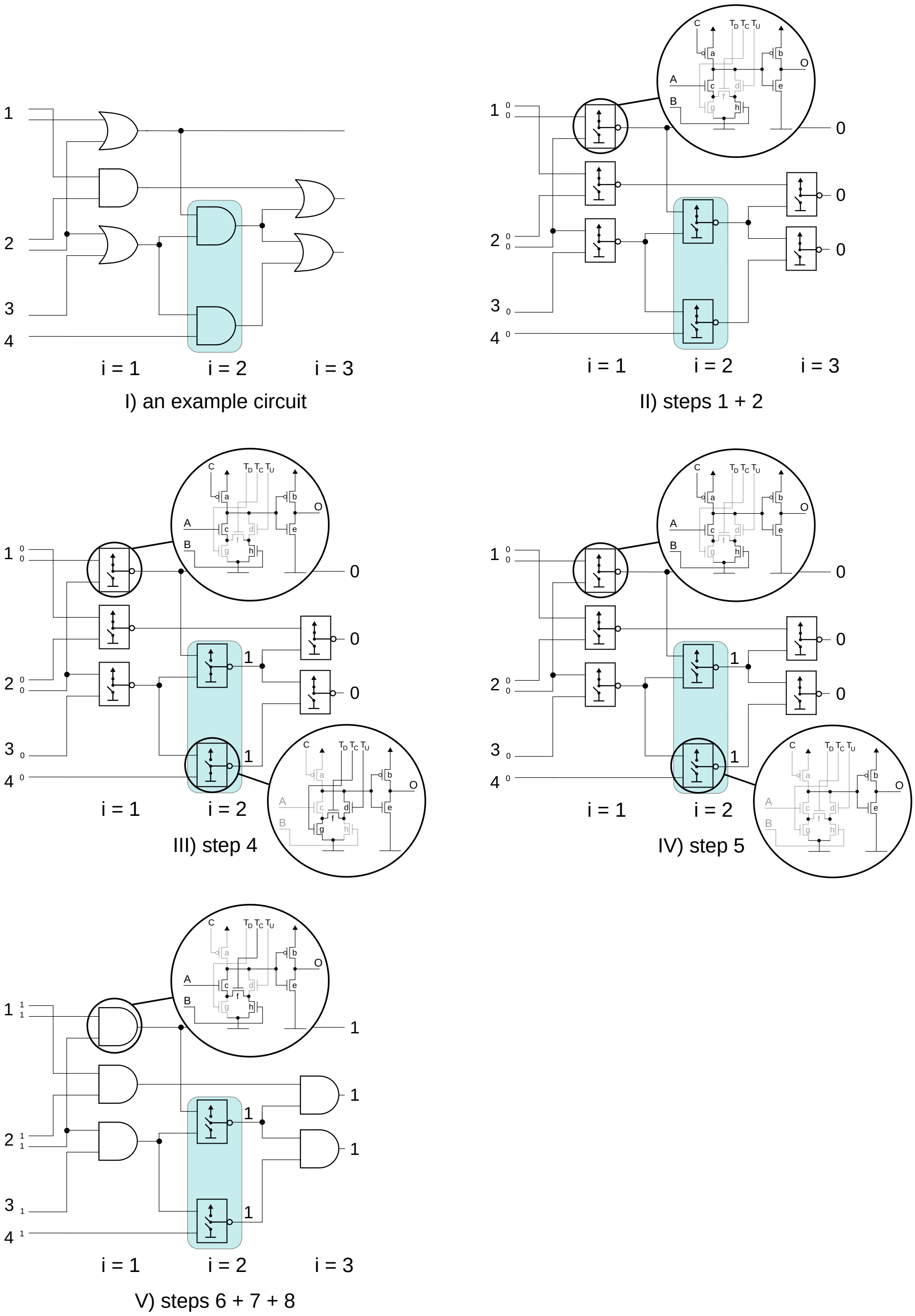
Table A.3: The test sequence of the *sub-test 3*

	tests covering faults	
transistor	stuck-on (short)	stuck-open
a	3	2
b	2*	1, 3
c	2	3
d	2	1
e	1*, 3*	2
f	2	1, 3
g	2	1
h	2	3

Table A.4: Sub-tests covering the faults

gates in the targeted level is tested – the level-by-level fault-symptom propagation is not possible. In this case, the function of gates in the targeted level is checked and the other gates are configured to propagate fault symptoms up to the circuit outputs.

In other tests, the output is tested only once at the end of each sub-test. The tests principle is that the value at the faulty gate output is flipped even if it should stay constant during the test. The value flip in the lower level causes that a pull-down path in the following level becomes conductive even if it should be closed (for sub-test 2) or vice-versa (for sub-test 3). In this way, a possible fault syndrome is propagated up to the primary outputs.



I) an example circuit

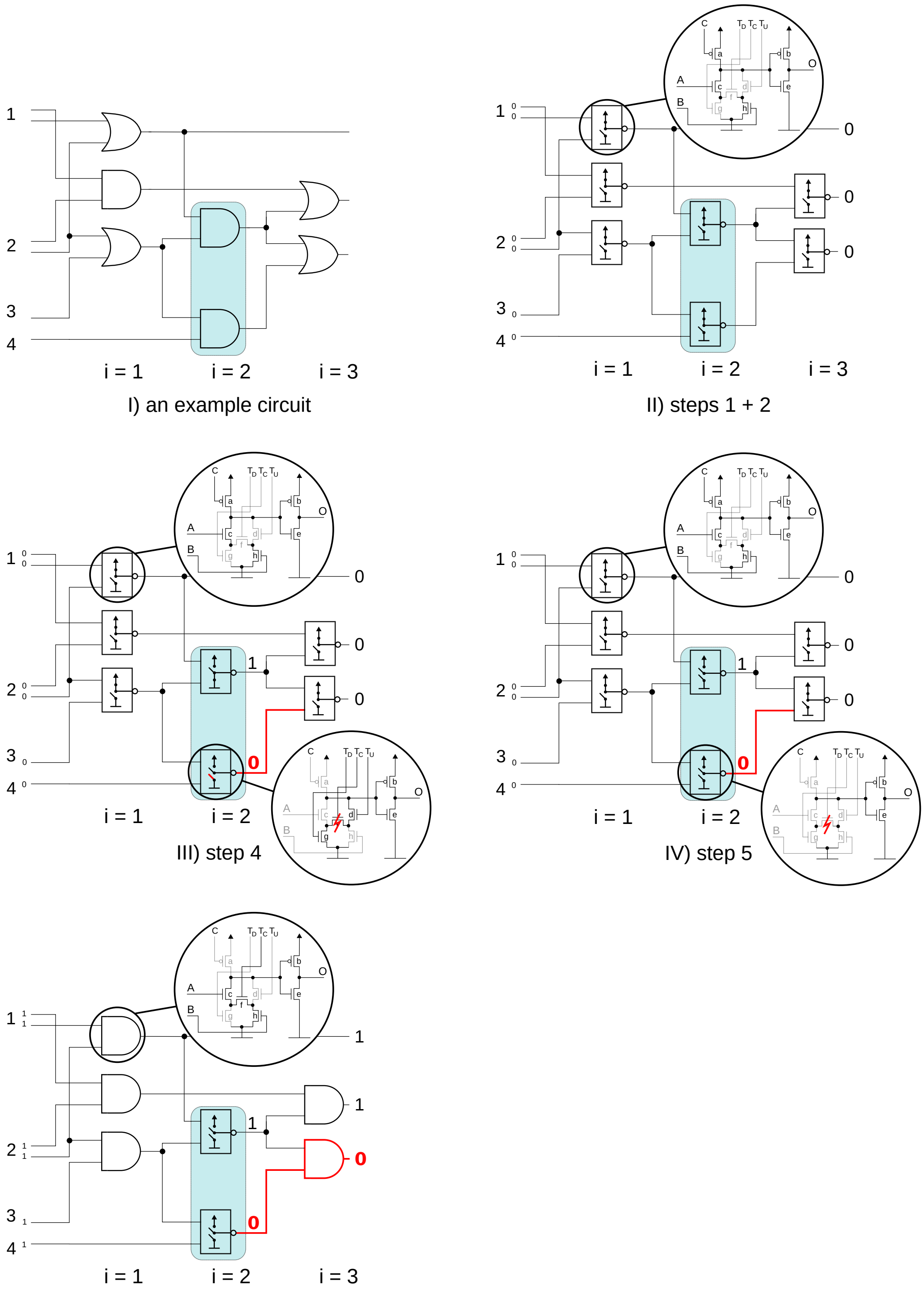
II) steps 1 + 2

III) step 4

IV) step 5

V) steps 6 + 7 + 8

Figure A.2: An iteration of the sub-test 1 for a fault-free circuit for level 2 (i = 2). Compare with the column “step” in Table A.1.



I) an example circuit

II) steps 1 + 2

III) step 4

IV) step 5

V) steps 6 + 7 + 8

Figure A.3: An iteration of the sub-test 1 for a faulty circuit for level 2 (i = 2). Compare with the column "step" in Table A.1. A behavior for a stuck-open fault in transistor f is shown. Stuck-open faults in transistors g, d or b in the same gate expose equivalent fault symptom.

Appendix B

Open Source VLSI EDA Tools and Resources

This section provides a short overview of the open-source tools and resources suitable for research and educational purposes in the field of VLSI design. The advantage of using open tools and resources in research is in removing a significant obstacle for replicability [32, 115]. The majority of results published in this dissertation thesis were realized using open-source tools and other publicly available resources.

The field of advanced digital VLSI EDA tools is nowadays partitioned between several major players supplying expansive toolchains covering design, verification, and test of integrated circuits. Many VLSI *Electronic Design Automation* (EDA) tools were created during the past about 50 years also in academia. The academic EDA tools constitute a diverse ecosystem. The open-source tools are rarely integrated into consistent and complete toolchains compared to their commercial counterparts. However, a number of the open-source EDA tools include industry-grade properties while being actively developed. Several notable projects also entered the field in the last few years.

The important part of the VLSI development process is the connection with the manufacturing technology. Every design must be designed and verified according to the design rules specific to the given technology. Additionally, specific simulation models and cell libraries must be available. The package of the design rules and libraries ready-to-use for a supported toolchain is called *Process Design Kit* (PDK).

B.1 Digital Design Toolchains

The major commercial supplyers of commercial EDA tools are *Siemens EDA* (formelly *Mentor Graphics*) with it's flagship *Calibre* [110]; *Cadence* with their *Virtuoso* [22]; and *Synopsys* with its *Custom Design Platform* [116].

The common limitation of most completely open toolchains is the lack of support for advanced technology nodes (below 100nm). In spite, the advanced VLSI design is today hard-to-imagine without expensive commercial tools, startups like *efabless.com* [38] or *Symbiotic* [48] recently raised around open digital toolchains. Similarly, the VLSI-related research can be (in part) covered by open tools.

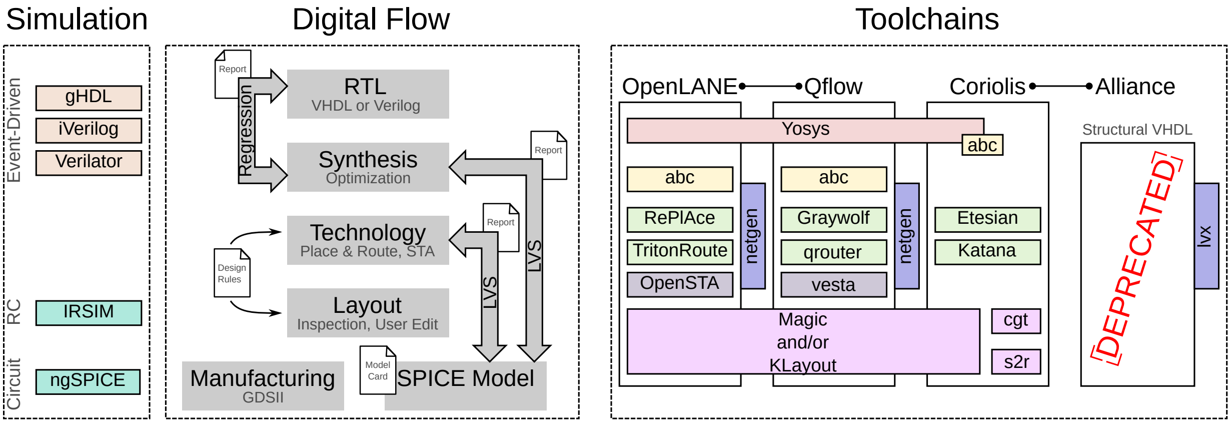


Figure B.1: Typical Digital Design Flow, notable open simmulation tools, and notable open digital toolchains

The typical digital design flow components are shown in Figure B.1, where the design is *synthesized* from the RTL to structural and finally to physical description. The design and also the overall design synthesis process must be verified – typical verification involves simulation at different levels and layout-vs-schematic checks – see Figure B.1.

The available open RTL simulators cover both major design and simulation languages used in digital design – VHDL and Verilog. Notable examples of widely used simulators are GHDL [96] or Icarus-Verilog [133]. Fast layout simulation might be performed at the switch-level using RC simulators. The ancient, while still maintained RC simulator is IRSIM [33].

The circuit-level simulation is typically performed in one of the SPICE programs (Simulation Program with Integrated Circuit Emphasis). All the SPICE programs have a common predecessor in the Berkeley SPICE [89]. The direct open-source successor of the Berkeley SPICE is ngSPICE [125] originally based on the codebase of the last release of Berkeleys' SPICE, Spice3f5 [125]. ngSPICE is competitive with the commercial-grade SPICE programs in common tasks. Additionally, it offers compatibility modes with widely used commercial SPICE programs: Cadences' PSPICE and LTSPICE [8]. Another open-source spice-like simulation program is Gnucap [31]. Even Gnucap suffers mostly by incompatibility with the latest CMOS models (BSIM4 is not supported), smaller developer- and userbase, it comes with a more modern design (the codebase is not ancient compared to ngSPICE) and includes better support for more recent technologies e.g. the subset of *Verilog-AMS* [63].

Yosys [134] and alternatively, ODIN-II [126] are nowadays predominantly used for RTL synthesis. For logic optimization, the Berkeley ABC [82, 83] is used. The lower levels of the digital flow are typically toolchain-dependent. Figure B.1 shows the notable open digital toolchains including the most important tools used as their building blocks.

The most complete open digital toolchains were for years QFlow [36] developed by Tim Edwards, and the toolchains developed by Sorbonne University: Alliance and its successor Coriolis. The digital circuit could also be designed using long-term living GNU Electric [44].

The most complete VLSI layout tools incorporated into many open toolchains are Magic [34] and KLayout [97].

In 2018, the OpenROAD [124] project has been started [3]. The aim of this project supported by DARPA (Defense Advanced Research Projects Agency) is to create an automated (AI *co-driven*) design flow for advanced technology nodes targeting 16/14nm technology node [3]. At the time of writing this dissertation thesis, the stable release date of OpenROAD is approaching. In the OpenROAD framework, several game-changing open tools were created – some of them were already used for real designs [37]. The open-ROAD tools were incorporated into OpenLANE [39] toolchain replacing and enhancing the long-standing, but less powerful tools originally used in QFlow [37, 46]. OpenLANE is developed by Efabless and it is already proven by real designs [37].

All the open toolchains are *fighting* with design resources [3, 37], as the process-related data remain predominantly closed (available under NDA), there are only a few available PDKs. On the other hand, the closed PDKs apparently are the base of the business model for open-source toolchain developers like Efabless allowing them to monetize their unique know-how and design-proven PDKs for open tools.

Currently, QFlow slowly becomes deprecated, as OpenLANE became more powerful and contains notable extensions like DFT or a more powerful router. However, like this dissertation thesis overlap with OpenLANE and OpenROAD development, the toolchain used through this dissertation thesis is QFlow: GrayWolf [50] is used for placement, QRouter [35] for routing and Magic [34] is used as a primary VLSI layout tool.

B.2 Process Design Kits

The *Process Design Kits* (PDKs) are developed by or for the silicon foundries for their proprietary processes. Typically, PDKs are available for a subset of major toolchains. In the past years, the interoperable PDK (iPDK) [78] initiative was introduced to increase tool interoperability, however, only specific tools are still often supported by *foundries* for their processes. The PDKs are available for developers typically under NDA only, to protect the foundry's know-how. A comprehensive overview of technologies and PDKs available for European researchers is provided by Europractice [117].

The PDKs for open toolchains created for academic projects are typically kept in-house, as they also rely on NDAs signed with foundries. An overview of notable publicly available PDKs is in Figure B.2. The most widely used and publicly available design rule set is the *MOSIS Scalable CMOS* [87]; unfortunately, MOSIS has cut down the foundry support significantly – in 2018, MOSIS accepted designs created using *Scalable CMOS* (SCMOS)

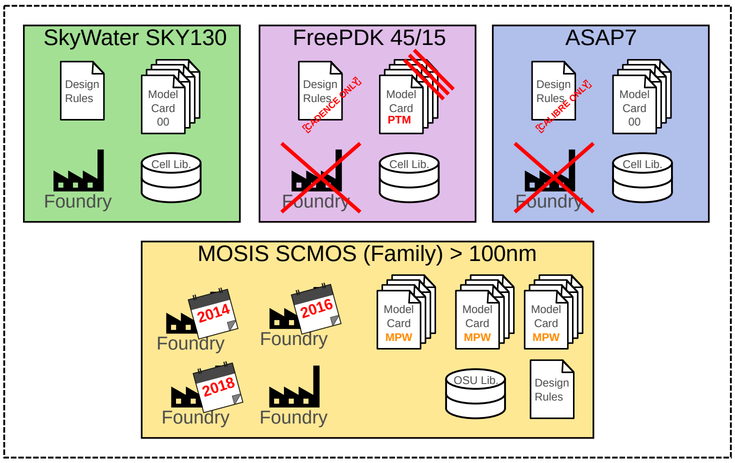


Figure B.2: Open PDK Overview

for only one fabrication process [87]. There are also not many open standard cell libraries. The notable exception is the standard cell library provided by Oklahoma State University (OSU) [91] used throughout this dissertation thesis.

Last, but not least, the full-featured PDK must include SPICE models. There is in general a great lack of open SPICE models for VLSI. One option is the *Predictive Technology Model* (PTM) collection developed by the Nanoscale Integration and Modeling (NIMO) Group at ASU [64]. The disadvantage of this model collection is the lack of corner models taking process variations into account. The models are *virtual*, meaning that they are not fitted to an existing manufacturing process.

As an alternative, the author of this dissertation thesis compiled a SPICE model collection [21] using archived MOSIS MPW data for several processes scattered on the web. The collection of models gives an idea of process variations and can be employed in Monte-Carlo simulation as an alternative if corner models are not available.

The complete open PDK intended for research and education is the FreePDK [123] for 45nm and 15nm processes. Even though this PDK is free, it is developed exclusively for Cadence tools. The other disadvantage is that the PDK is *virtual*, meaning that it has no connection to the manufacturing process, nor a complete SPICE model collection is available – virtual PTM models are used.

Similarly, the ASAP7 [122] PDK targeting the 7nm *virtual node* was created by Arizona State University in cooperation with ARM Research. Compared to FreePDK, ASAP7 requires Calibre tools. The notable advantage of this PDK is that it includes transistor corner models enabling more accurate SPICE-level analysis.

The most complete effort in this area is a fresh work-in-progress open PDK project *SkyWater SKY130 PDK* [12], which is a joined effort of Google and SkyWater Technology – the foundry. This project includes design files for open toolchains and it also utilizes the OSU library as an option.