UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM MICROELETRÔNICA

ROGER CAPUTO LLANOS

Voltage Scaling Interfaces for Multi-Voltage Digital Systems

Thesis presented in partial fulfillment
of the requirements for the degree of
Master of Microeletronics

Advisor: Prof. Dr. Ricardo Reis
Coadvisor: Prof. Dr. Marcelo Johann

Porto Alegre
December 2015

CIP — CATALOGING-IN-PUBLICATION

Llanos, Roger Caputo

Voltage Scaling Interfaces for Multi-Voltage Digital Systems / Roger Caputo Llanos. – Porto Alegre: PGMICRO da UFRGS, 2015.

119 f.: il.

Thesis (Master) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Microeletrônica, Porto Alegre, BR–RS, 2015. Advisor: Ricardo Reis; Coadvisor: Marcelo Johann.

1. Level Shifter. 2. Low Power. 3. Multiple Supply Voltage. 4. Power-Delay Product. 5. CMOS. 6. Dynamic Power. 7. Static Power. I. Reis, Ricardo. II. Johann, Marcelo. III. Título.

Para mi amada Lucy, compañera de ésta travesía y de todas las vidas.
Para mi madre y mis tías, a quienes debo el haber llegado hasta aquí.
A Lila, gracias por creer en mis “locuras”.
Gracias totales!

ABSTRACT

Multiple Voltage Digital Systems exploit the concept of voltage scaling by applying different supplies to particular regions of the chip. Each of those regions belongs to a power domain and may have two or more supply voltage configurations. Regardless of distinct energy levels on different power domains, the blocks shall process signals with coherent logic levels. In these systems, the Level Shifters (LS) are essential components that act as voltage scaling interfaces between power domains, guaranteeing the correct signal transmission. With the appropriate voltage scaling interface and its proper implementation, we can avoid excessive static and dynamic power consumption. Therefore, the design and implementation of level shifters should be a conscientious process and must guarantee the lowest overhead in size, energy consumption, and delay time. In this work, we study the main characteristics of voltage scaling interfaces and introduce an energy-efficient level shifter with reduced area, and suitable for *low-to-high* level conversion. We present the level shifters with the best performance that we found in the literature and categorize them into two main groups: Dual-rail and Single-rail, according to the number of power rails required. The proposed circuit was compared to the traditional topology of each group, *Differential Cascode Voltage Switch* (DCVS) and *Puri*'s level shifter respectively. Simulations on an IBM™ 130nm CMOS technology show that the proposed topology requires up to 93.79% less energy under certain conditions. It presented 88.03% smaller delay and 39.6% less *Power-Delay Product* (PDP) when compared to the DCVS topology. In contrast with the Puri's level shifter, we obtained a reduction of 32.08% in power consumption, 13.26% smaller delay and 15.37% lower PDP. Besides, our level shifter was the only one capable of working at 35% of the nominal supply voltage.

Keywords: Level Shifter. Low Power. Multiple Supply Voltage. Power-Delay Product. CMOS. Dynamic Power. Static Power.

Interfaces de escalonamento de tensão para Sistemas Digitais de Múltiplas Tensões

RESUMO

Os Sistemas Digitais de Múltiplas Tensões exploram o conceito de dimensionamento da tensão de alimentação através da aplicação de diferentes fontes para regiões específicas do chip. Cada uma destas regiões pertence a um domínio de energia e pode ter duas ou mais configurações de voltagens. Independentemente dos distintos níveis de energia em diferentes domínios de tensão, os blocos devem processar sinais com níveis lógicos coerentes. Nestes sistemas, os Conversores de Nível (LS do inglês *Level Shifters*) são componentes essenciais que atuam como interfaces de escalonamento da tensão entre domínios de energia, garantindo a correta transmissão dos sinais. Com a apropriada interface de escalonamento de tensão e sua correta implementação, pode-se evitar o consumo excessivo de potência dinâmica e estática. Portanto, a concepção e implementação de conversores de nível deve ser um processo consciente que garanta o menor sobrecusto no tamanho, consumo de energia, e tempo de atraso. Neste trabalho estudam-se as principais características das interfaces de escalonamento de tensão e se introduce um conversor de tensão com eficiência energética e área reduzida, adequado para a conversão de baixo a alto nível. Apresentam-se os conversores de nível com o melhor desempenho encontrados na literatura, os quais são categorizados em dois principais grupos: Dois trilhos (*Dual-rail*) e Único trilho (*Single-rail*), de acordo ao número de linhas de alimentação necessárias. O circuito proposto foi comparado com a topologia tradicional de cada grupo, o *Differential Cascode Voltage Switch* (DCVS) e o conversor de Puri respectivamente. Simulações na tecnologia CMOS 130nm da IBM^{TM} mostram que a topologia proposta requer até 93,79% menos energia em determinadas condições. Esta apresentou 88,03% menor atraso e uma redução de 39,6% no Produto Potência-Atraso (PDP), quando comparada com a topologia DCVS. Em contraste com o conversor Puri, obteve-se uma redução de 32,08% no consumo de energia, 13,26% diminuição no atraso e 15,37% inferior PDP. Além disso, o conversor de nível proposto foi o único capaz de trabalhar a 35% da tensão nominal de alimentação.

Palavras-chave: Conversores de Nível. Baixo Consumo. Múltiplas Tensões. Produto Potência-Atraso. CMOS. Potência Dinâmica. Potência Estática.

LIST OF ABBREVIATIONS AND ACRONYMS

ALC	Asynchronous Level Converters
AVS	Adaptive Voltage Scaling
CMOS	Complementary Metal-Oxide Semiconductor
CPU	Central Process Unit
CVS	Clustered Voltage Supply
DIBL	Drain Induced Barrier Lowering
DCVSL	Differential Cascode Voltage Switch Logic
DLS	Dual-rail Level Shifters
DVS	Dynamic Voltage Supply
DVFS	Dynamic Voltage and Frequency Scaling
ECVS	Extended Clustered Voltage Scaling
EDA	Electronic Design Automation
EM	Electromigration
EOP	Energy Per Operation
GECVS	Greedy Extended - Clustered Voltage Scaling
HDL	Hardware Description Language
IC	Integrated Circuit
IP	Intellectual Property
ITRS	International Technology Roadmap for Semiconductors
LCFF	Level Converting Flip-Flop
LS	Level Shifters
MOS	Metal-Oxide Semiconductor
MOSFET	Metal-Oxide Semiconductor Field-Effect Transistor
MDSV	Multiple Dynamic Supply Voltage

MVS	Multi-level Voltage Scaling
PDP	Power-Delay Product
ROI	Return-on-Investment
RTL	Register Transfer Level
SLS	Single-rail Level Shifters
SoC	System-On-Chip
STA	Static Timing Analysis
SVS	Static Voltage Scaling
VLSI	Very Large Scale Integration
VRM	Voltage Regulator Module
VTC	Volta-transfer Characteristic

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 INTRODUCTION

The design of complex chips has undergone a series of revolutions. From the introduction of language-based design and synthesis to the adoption of design reuse and IP as a mainstream practice. Each of these revolutions was the response to the challenges posed by evolving semiconductor technology and by the need for more efficient Very-Large Scale Integration (VLSI) systems. In the past decade, energy and power consumption has become one of the primary concerns in sophisticated systems-on-chip (SoC) (RABAEY, 2009). For its impact and importance in VLSI designs, we can consider low power as the most recent design revolution.

As technology shrank to 90nm and below, the leakage current increased dramatically, to the point where, in some 65nm designs, leakage is nearly as large as dynamic current (KEATING et al., 2007). These changes reflect on how we conceive the integrated circuits. For example, the power density of the highest performance chips grew to the point where it was no longer possible to increase clock speed as technology advanced. Subsequently, multi-processor chips emerged as an option instead of chips with a single, ultra-high speed processor (OLUKOTUN et al., 1996).

The continuous transistor downsizing has also empowered the development of portable devices. By using eight or more CPU cores, current smartphones match the computational capabilities of basic desktop computers and laptops. Despite its computing performance, a smartphone is a battery-powered system. Where, the available energy is fixed, and the rate of power consumption determines the lifetime of the battery or the time between recharges. It means that the energy supply is finite, and hence energy minimization is critical.

Rabaey (2009) showed that the battery capacity doubles approximately every ten years. Although this represents an improvement of 3 to 7% every year, the growth curve lags substantially behind Moore's law. This disparate evolution could be explained by the fact that the improvements in capacity are often related to new chemicals or electrode materials. Capacity and size limitations of batteries push further the energy savings of battery-powered devices.

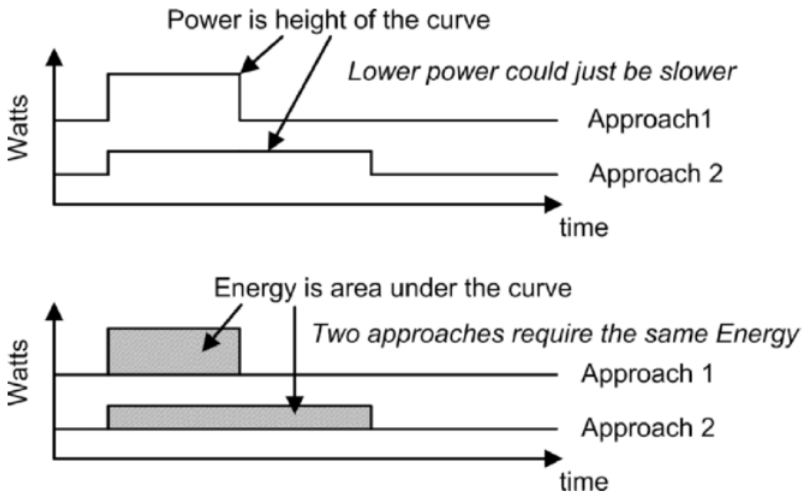
Smartphones are just one example of devices with limited power supply, which require low energy consumption. For a vast number of designs, low power is the hardest constraint or the central objective. Wearable devices, bioengineering, sensor networks, among others, are application areas that may require conscious low-power design. The technology scaling may help to address some of the power issues. Nevertheless, design solutions will be the primary mechanisms for keeping energy consumption under control or within bounds.

In today's design environment, optimizing for only one parameter (delay or energy) rarely makes sense. The design with the minimum propagation delay in general takes an exorbitant amount of energy, and, vice versa, the design with the minimum energy is unacceptably slow. Both represent extremes in an optimization space, where many other optimal operational points and metrics exist. Typically, one metric receives greater weight; for instance, energy reduces for a given maximum delay or delay is minimized for a given maximum energy.

1.1 Power, Energy, and Delay Basics

Energy and Power terms are almost used indistinctly, but they have particular definitions despite the direct relation between each one. Energy is the product of power and time, in other words, it is power integrated over time. We can measure power at any point in time, whereas energy has to be measured during a period. Figure 1.1 illustrates better the difference between both. Energy is the area under the curve, and Power is the height of the graph (KEATING et al., 2007).

Figure 1.1 – Power Vs Energy



Source: (KEATING et al., 2007)

The power consumption of a design determines how much energy it spends per operation, and how much heat the circuit dissipates. These factors influence a number of critical design decisions, such as the power-supply capacity, the battery lifetime, packaging and cooling requirements. Therefore, power dissipation is an important property of a VLSI design that affects the feasibility, cost, and reliability. With the increasing popularity of mobile and distributed computation, energy limitations put a firm restriction on the number of computations

at a given minimum time between battery recharges.

Power dissipation can be decomposed into static and dynamic components. The latter occurs only during transients when the gate is switching. It is attributed to the charging of capacitors and temporary current paths between the supply rails, and it is, therefore, proportional to the switching frequency. The higher the number of switching events, the higher the dynamic power consumption (RABAEY, 2009).

The charging and discharging of capacitances are the main sources of dynamic power dissipation. Other contributions are the parasitic effects of short-circuit currents and dynamic hazards or glitches. On the other hand, the static component is present even when no switching occurs. Static conductive paths between the supply rails or leakage currents are the main reasons for the static power consumption. It is always present, even when the circuit is in stand-by mode.

The propagation delay and the power consumption of a gate are related, the propagation delay is mostly determined by the speed at which a given amount of energy can be stored on the gate capacitors. The faster the energy transfer (or, the higher the power consumption), the faster the gate (RABAEY, 2009). The product of power consumption and propagation delay is generally constant. This product is the *Power-Delay Product* (PDP) and represents the energy consumed by the gate per switching event.

1.2 Dynamic Power Consumption

Switching power consumption occurs when logic switches from *low-to-high* (and vice versa) and capacitances are charged and discharged. In summary, each switching cycle (consisting of a *low-to-high* and a *high-to-low* transition) takes a fixed amount of energy (RABAEY; CHANDRAKASAN; NIKOLIC, 2003). This being equal to $C_L {V_{DD}}^2$.

The switching power consumption can be described as a function of the energy spent per transition at a specific clock frequency. Equation (1.1) represents the dynamic power consumption when a gate is switched *on* and *off* $f_{0\to1}$ times. C_L is the load capacitance and V_{DD} the supply voltage.

$$P_{dyn} = C_L {V_{DD}}^2 f_{0\to1} \tag{1.1}$$

f represents the frequency of energy-consuming transitions for static Complementary Metal-Oxide Semiconductor (CMOS)

Computing the dissipation of a complex circuit is complicated by the $f_{0\to1}$ factor. While

the switching activity is easily computed for an inverter, it turns out to be far more complex in the case of higher-order gates. From equation 1.1, we derive the expression 1.2, where f is the clock frequency, and α is the switching activity.

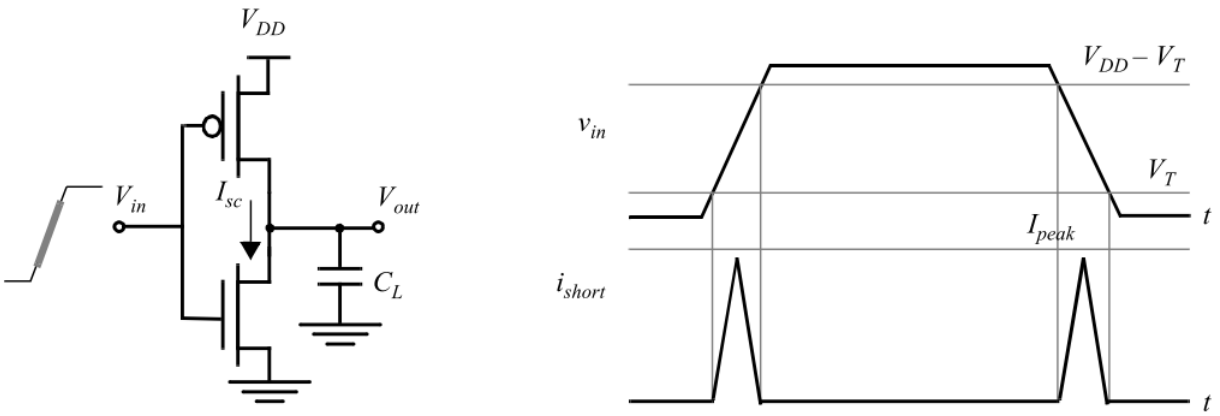
$$P_{dyn} = \alpha C_L {V_{DD}}^2 f \tag{1.2}$$

The switching activity is increased by glitches, which typically cause 15% to 20% of the activity in complementary static CMOS logic (CHINNERY; KEUTZER, 2007). In addition, short circuit power consists of the current that flows when both the NMOS and PMOS transistors of the CMOS inverter are on (Figure 1.2). As well as the current required to charge the internal capacitance (KEATING et al., 2007). This current creates temporary direct-path connecting V_{DD} and ground. Equation (1.3) estimates the short circuit current of an inverter without external load (CHINNERY; KEUTZER, 2007).

$$I_{sc} = c\mu \frac{\epsilon_{ox}}{t_{ox}} \frac{W}{L_{eff}} \frac{1}{V_{DD}} (V_{DD} - 2V_T)^3 s_{in} \tag{1.3}$$

c is a process-determined constant defined in (VEENDRICK, 1984); s_{in} is the input slew

Figure 1.2 – Short-circuit current during transients



Source: (RABAEY; CHANDRAKASAN; NIKOLIC, 2003)

If the rising and falling responses of the inverter are symmetric, the energy consumed per switching period due to direct-paths currents is (RABAEY; CHANDRAKASAN; NIKOLIC, 2003):

$$E_{dp} = V_{DD} \frac{I_{peak} t_{sc}}{2} + V_{DD} \frac{I_{peak} t_{sc}}{2} = t_{sc} V_{DD} I_{peak} \tag{1.4}$$

Where t_{sc} is the duration of the short circuit current, and I_{peak} is the total internal switching current (short circuit current plus the current required to charge the internal capacitance) (KEATING et al., 2007). The peak current is also a function of the ratio between input and output slopes (RABAEY; CHANDRAKASAN; NIKOLIC, 2003). The direct-path power dissipation is proportional to the switching activity, similar to the capacitive power dissipation. From equation (1.4), we obtain the average power consumption during short-circuit events (RABAEY; CHANDRAKASAN; NIKOLIC, 2003):

$$P_{dp} = t_{sc} V_{DD} I_{peak} f = C_{sc} {V_{DD}}^2 f \tag{1.5}$$

As long as the ramp time of the input signal is short, the short circuit current occurs for only a short time during each transition and the overall dynamic power is dominated by the switching power. Mainly because when the load capacitance is vast, the output fall time is significantly larger than the input rise time. Under those circumstances, the input moves through the transient region before the output starts to change. As the source-drain voltage of the PMOS device is approximately zero during that period, the device shuts off without ever delivering any current (RABAEY; CHANDRAKASAN; NIKOLIC, 2003). For this reason, the dynamic power is often reduced to the expression (1.2). Moreover, short-circuit power typically contributes less than 10% of the total dynamic power and increases with the escalation of V_{DD}, and with decreasing V_T (CHINNERY; KEUTZER, 2007).

1.3 Static Power Consumption

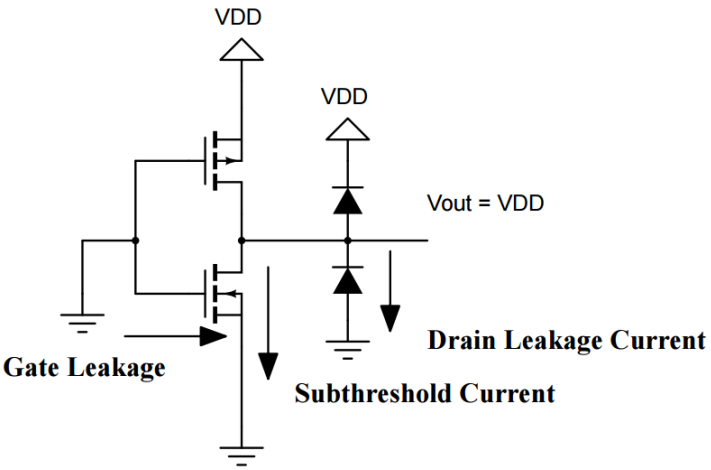
The current that flows between the supply rails in the absence of switching activity causes the Static power consumption. In equation (1.6), I_{stat} is ideally equal to zero, because PMOS and NMOS transistors of the CMOS inverter are never *on* simultaneously during static operation (RABAEY; CHANDRAKASAN; NIKOLIC, 2003).

$$P_{stat} = I_{stat} V_{DD} \tag{1.6}$$

From a digital perspective, an ideal MOS transistor should not have any currents flowing into the bulk, should not conduct any current between drain and source when off, and should have an infinite gate resistance (RABAEY, 2009). Actually, there is a small leakage current that contribute to static power dissipation (Figure 1.3). According to (KEATING et al., 2007), the four main sources of leakage current are:

- Subthreshold Leakage (I_{SUB}): the current that flows from the drain to the source of a transistor operating in the weak inversion region.
- Gate Leakage (I_{GATE}): it flows directly from the gate and occurs as a result of tunneling current through the gate oxide.
- *Gate-Induced* Drain Leakage (I_{GIDL}): flows from the drain to the substrate induced by a high field effect in the MOSFET drain caused by a high V_{DG}.
- Reverse Bias Junction Leakage (I_{REV}): caused by minority carrier drift and generation of electron/hole pairs in the depletion regions.

Figure 1.3 – Source of leakage currents in CMOS inverter



Source: Author

In deep submicron process with low threshold voltages, the dominant sources of leakage power are subthreshold leakage and gate leakage. Expression (1.7) shows an analytical model for the subthreshold leakage current (KESHAVARZI; KAO, 2002).

$$I_{SUB} = e^{1.8}\mu\frac{\epsilon_{ox}}{t_{ox}}\frac{W}{L_{eff}}\left[\frac{kT}{q}\right]^2 e^{q\frac{(V_{GS}-V_{T0}-\gamma V_b+\eta V_{DS})}{mkT}}(1-e^{-\frac{qV_{DS}}{kT}}) \tag{1.7}$$

Where k is the Boltzmann's constant; q is the charge of an electron; T is the temperature in Kelvins; V_{T0} is the zero-bias threshold voltage; V_b is the body bias voltage; γ is the linearized body effect coefficient; m is the subthreshold swing coefficient, and η is the Drain Induced Barrier Lowering (DIBL) coefficient. Keating (2007) shows a good approximation of (1.7):

$$I_{SUB} = \mu C_{ox}\frac{W}{L}\left[\frac{kT}{q}\right]^2 e^{\left[\frac{V_{GS}-V_T}{n\left(\frac{kT}{q}\right)}\right]} \tag{1.8}$$

Where (kT/q) is the thermal voltage ($25.9mV$ at room temperature) and n is a function

of the device fabrication process with ranges from 1.0 to 2.5. Equation (1.8) shows that the subthreshold leakage increases exponentially as the threshold voltage reduces. Furthermore, it increases linearly with gate size (W) and with the supply voltage. Therefore, as V_{DD} and V_T scale down (to limit dynamic power), leakage power gets proportionally worse.

Gate leakage occurs because of the tunneling current through the gate oxide. A high electric field across the thin transistor gate oxide originates an electrons flow. The gate leakage can be modeled as shown in (1.9) (CHINNERY; KEUTZER, 2007), where a, b, and c are constants defined in the BSIM4 model.

$$I_{GATE} = aL_{eff}e^{\left(bV_{GS}-ct_{ox}{}^{-2.5}\right)} + aL_{eff}e^{\left(bV_{GD}-ct_{ox}{}^{-2.5}\right)} \tag{1.9}$$

In previous technology nodes, subthreshold leakage dominated leakage current. However, starting with 90nm, gate leakage can be nearly $1/3$ as much as subthreshold leakage (KEATING et al., 2007). There are several approaches to minimizing leakage current. One technique, known as Multi-V_T, uses high V_T cells wherever performance goals allow, and low V_T where necessary to meet timing constraints.

1.4 Propagation Delay

The propagation delay is defined by the time between the 50% transitions of the input and the output. For the CMOS inverter, the propagation delay is the time it takes to charge and discharge the load capacitor (C_L) through the PMOS and NMOS transistors. A single inverter can theoretically drive an infinite number of gates (or have a vast fan-out); however, increasing the fan-out also increases the propagation delay. Therefore, a small load capacitor or reduced fan-out improves the propagation delay, enabling the realization of high-performance CMOS circuits.

One way to compute this delay is by integrating the capacitor current (RABAEY; CHANDRAKASAN; NIKOLIC, 2003). It results in the following expression, where i is the discharging current, v the voltage over the capacitor, and v_1 and v_2 the initial and final voltage:

$$t_p = \int_{v_1}^{v_2} \frac{C_L(v)}{i(v)} dv \tag{1.10}$$

An exact computation of this equation is impractical, as both $C_L(v)$ and $i(v)$ are non-linear functions of v. We rather use a simplified switch-model of the inverter to derive the propagation delay from the analysis of a first-order linear *RC*-network (RABAEY; CHAN-

DRAKASAN; NIKOLIC, 2003). Where the delay is proportional to the time-constant of the network, formed by a pull-down resistor and load capacitance. Hence, the propagation delay for the *high-to-low* transition becomes:

$$t_{pHL} = \ln(2) R_{eqn} C_L = 0.69 R_{eqn} C_L \tag{1.11}$$

Similarly, we can obtain the propagation delay for the *low-to-high* transition,

$$t_{pLH} = 0.69 R_{eqp} C_L \tag{1.12}$$

On (1.11) and (1.12), R_{eqn} and R_{eqp} are the equivalent on-resistance of the MOS transistor over the interval of interest. This analysis assumes that the equivalent load-capacitance is identical for both the *high-to-low* and *low-to-high* transitions. Then, the overall propagation delay of the inverter becomes the average of the two values.

$$t_p = \frac{t_{pHL} + t_{pLH}}{2} = 0.69 C_L \left(\frac{R_{eqn} + R_{eqp}}{2} \right) \tag{1.13}$$

1.5 Main motivation

The total power consumption of a SoC consists of dynamic power and static power. As explained before, dynamic power is mainly present when the device is active or in switching activity, and the static power represents the energy consumed when the device is inactive or in standby mode. There is a broad range of approaches for optimizing both components of power consumption. From systems with multiple voltages to power-awareness designs, and, at a lower abstraction level, transistor sizing and supply voltage scaling enable a grainy control of the circuit's consumption.

Optimizations at the architecture or system level can enable more effective power minimization and maintain performance at the same time. While circuit techniques may yield improvements in the 10-50% range, architecture and algorithm optimizations have reported greatest power reduction (RABAEY, 2009). Among the latter, we can mention designs that employ multiple voltages to optimize the power consumption.

Those multi-voltage designs exploit the concept of voltage scaling by applying different supplies to specific regions of the chip. Each region belongs to a power domain and may have two or more supply voltage configurations. Regardless of distinct energy levels on differ-

ent power domains, the blocks shall process signals with coherent logic levels. When driving signals between power domains with radically different power rails, the need for level converters is essential. These voltage scaling interfaces (also referred as *Level Shifters*) are located at the power domain boundaries, and they must ensure the correct communication between two regions. Therefore, the design and implementation of level shifters should be a conscientious process and must guarantee the lowest overhead in size, energy consumption, and delay time. Thus, the main motivation of this research is to study and to identify the best level shifter topologies for low power consumption and reduced area, as well as to propose a novel approach.

The following chapter describes some of the most common techniques to reduce the power consumption of SoC designs. Among them, Multi-Voltage approaches are versatile solutions that present considerable energy savings and are widely used in modern designs. Chapter 3 presents the basic Multi-Voltage techniques we found in the literature and explains some of their challenges, including the need for Level Shifters (LS). The reader can find more information about these voltage scaling interfaces in Chapter 4 of this document, which also summarizes the *state-of-the-art* Level Shifters. After analyzing the approaches with the best results, we introduce a novel topology of Level Shifter with low power consumption. The proposed circuit and its effects can be observed in Chapter 5 and subsequent section. Finally, our final considerations and future prospects conclude this work.

2 LOW-POWER OPTIMIZATION STRATEGIES

Designers dispose of several approaches to minimize the impact of the switching activity or leakage current of CMOS digital circuits. In a custom VLSI project, they can implement low-power optimizations almost at every step of the design process. From a *bottom-up* view, the most common techniques for optimizing power are: transistor-level optimizations; voltage scaling; clock gating; power-gating; logical optimizations; and architecture-level optimizations. An ideal design methodology employs low power techniques in all abstraction levels to obtain better results. The gain at each level contributes to greater energy savings.

2.1 Transistor-level Optimizations

Custom techniques used to achieve high-speed on designs may also be employed to reach low power constraints (THOMPSON et al., 2001). Designers can optimize the individual logic cells, the layout, the wiring between cells, and other aspects of the design as the number of transistors, their size and positions. A reduction in the number of transistors can impact both, static and dynamic power consumption. In (SCARTEZZINI; REIS, 2011), authors achieved better results in terms of power and delay, when using networks with reduced number of transistors instead of commercial standard cells.

2.1.1 Transistor Sizing

In some custom designs, engineers have the option to enlarge or reduce the width of the transistors channel to target the short-circuit power. Transistor Sizing is a useful technique to lessen the delay of a CMOS circuit. When the width of the channel enlarges, the current drive capability of the transistor increases, which reduces the signal rise/fall times at the gate output.

Transistor sizing technique becomes a trade-off between speed and power dissipation because varying the channel width affects both power and delay. On equation (2.1), if we increase the width (W), the resistance decreases allowing more current to flow through the transistor. This may reduce the delay of the transistor but augments the energy consumption instead.

$$R_n = \frac{1}{\mu C_{ox} \left(V_{GS} - V_T\right) 2} \left(\frac{L}{W}\right) \tag{2.1}$$

By changing W, we can control the gain factor of a CMOS inverter. Equation (2.2) represents the maximum short-circuit dissipation under *no-load* condition at the output of the inverter (VEENDRICK, 1984). Where τ is the input transition time and β is the gain factor of the transistor.

$$P_{sc} = \frac{\tau\beta}{12}(V_{DD} - 2V_T)^3 f \tag{2.2}$$

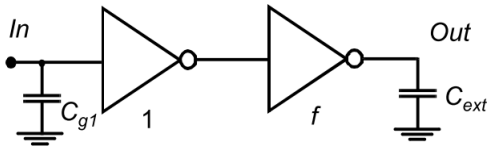
On equation (2.2), the gain factor of the gate (β) is determined by the width of the transistor (W) and the mobility of the carriers responsible for the transition (μ_p for a *low-to-high* transition and μ_n *for a high-to-low transition*). Expression (2.2) further reduces to (2.3), where k is a process and voltage dependent constant of proportionality. It clearly shows that the short-circuit power (P_{sc}) consumption is directly proportional to both width of the transistor and the input transition time.

$$P_{sc} = k\mu W\tau f \tag{2.3}$$

Borah, Owens, and Irwin (1996) presented an analytical derivation of the optimum value of W under high fan-out condition. They also showed a nonlinear (*U-shape*) curve representing transistor size versus average power dissipation. The *U-shape* curve indicates that there is an optimum point to design the channel width. A large channel width may increase the driving current but reduce the transition time, thus reducing the short-circuit power. Over-sizing the transistors beyond the optimal value comes at a hefty price in energy. On the other hand, a small channel width could decrease the driving current but result in a longer transition time, thus increasing the short-circuit power (YUAN; DI, 2005).

Figure 2.1 shows a static CMOS inverter driving an external capacitance (C_{ext}) and a minimum-sized device drives the inverter. The goal here is to minimize the energy dissipation of the complete circuit while maintaining the performance (RABAEY; CHANDRAKASAN; NIKOLIC, 2003). The propagation delay of the optimized circuit should be smaller or equal to the delay of a reference circuit, with size factor $f = 1$ and supply voltage $V_{DD} = V_{ref}$.

Figure 2.1 – CMOS inverter driving an external load



Source: (RABAEY; CHANDRAKASAN; NIKOLIC, 2003)

Expression (2.4) represents the propagation delay of the circuit. With (*effective fan-out*) $F = (C_{ext}C_{g1})$, the intrinsic delay of the inverter is the overall effective fan-out of the circuit t_{p0}.

$$t_p = t_{p0}\left[\left(1+\frac{f}{\gamma}\right)+\left(1+\frac{F}{f\gamma}\right)\right] \tag{2.4}$$

The energy dissipation for a single transition at the input of the circuit can be determined by:

$$E = {V_{DD}}^2 C_{g1}\left[(1+\gamma)(1+f)+F\right] \tag{2.5}$$

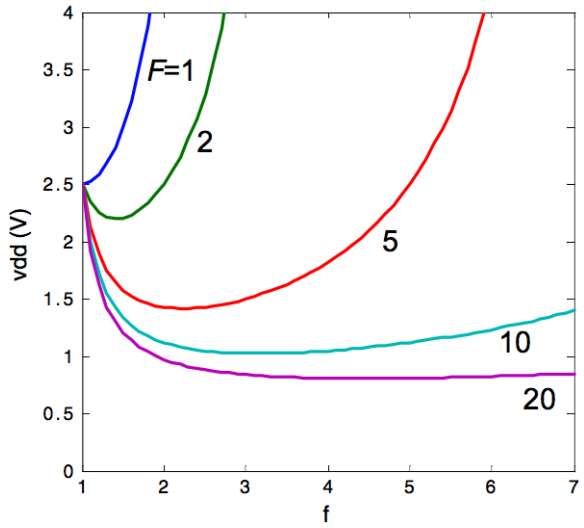
If the intrinsic output capacitance of the gate equals its gate capacitance ($\gamma = 1$), we have the following relation:

$$\frac{t_p}{t_{p_{ref}}} = \frac{t_{p0}\left(2+f+\frac{F}{f}\right)}{t_{p0_{ref}}(3+F)} = \left(\frac{V_{DD}}{V_{ref}}\right)\left(\frac{V_{ref}-V_{TE}}{V_{DD}-V_{TE}}\right)\left[\frac{2+f+\frac{F}{f}}{3+F}\right] \tag{2.6}$$

The previous equation establishes a relationship between the sizing factor f and the supply voltage. Figure 2.2 plots this correlation for different values of f. It shows that increasing the size of the inverter rises the performance until the sizing factor of $f = \sqrt{F}$ is reached. Further increases in the device size only intensify the self-loading factor, deteriorate the performance, and require an escalation in supply voltage. Therefore, over-sizing the transistors beyond the optimal value comes at a hefty price in energy.

Earlier approaches for transistor sizing concentrated on minimizing the area of the circuit subject to a particular delay constraint. Those methods were based on the assumption that the power consumption of a circuit was proportional to the active area (HOPPE et al., 1990; SAPATNEKAR et al., 1993). Later studies revealed that the power consumption of a CMOS circuit does not always reduce by minimizing the active area; but it can be improved by enlarging some of the transistors that drive large active loads (BORAH; IRWIN; OWENS, 1995; KO; BALSARA, 1995). In contrast to the existing assumption at the time; Borah, Owens, and Irwin (1996) showed that the power consumption of a circuit is a convex function of the area. They presented an analytical model based on the analysis of short-circuit power consumption by Veendrick (1984) and the gate delay model of Hedenstierna and Jeppson (1987). In which,

Figure 2.2 – Sizing of a CMOS inverter for energy-minimization



Supply voltage as a function of the sizing factor f for different values of the overall effective fan-out F

Source: (RABAEY; CHANDRAKASAN; NIKOLIC, 2003)

the curve of average power consumption versus transistors size attains a minimum when:

$$W_1^* = \frac{\sqrt{\phi \frac{\mu'}{\mu} C_L \left(\sum_{i=2}^{n} W_i f_i\right)}}{\sqrt{\left(\frac{k_1}{k} + \mu\tau\right) f_1}} \tag{2.7}$$

In (2.7), W_1^* is the *power optimal size* for the transistor driving various gates (first inverter on Figure 2.1). By substituting μ_p for μ and μ_n for $\mu^{'}$, we obtain the optimal size of the *p-channel* transistor. The optimal *n-channel* transistor size can be obtained by substituting μ_n and μ_p, respectively. This approach can be used in the situation where each transistor is scaled up uniformly.

In complex CMOS gates, the best method to scale the transistors is by progressive upsizing them. Basically, progressive transistor sizing approach reduces the transistor resistance while reducing capacitance (RABAEY; CHANDRAKASAN; NIKOLIC, 2003). Figure 2.3 shows an example of progressive sizing of transistors in a large transistor chains, each one driving a corresponding external load.

The delay of a chain of N transistors of width W is less than n times the delay of a single inverter of width W is given by (2.8) (BORAH; OWENS; IRWIN, 1996).

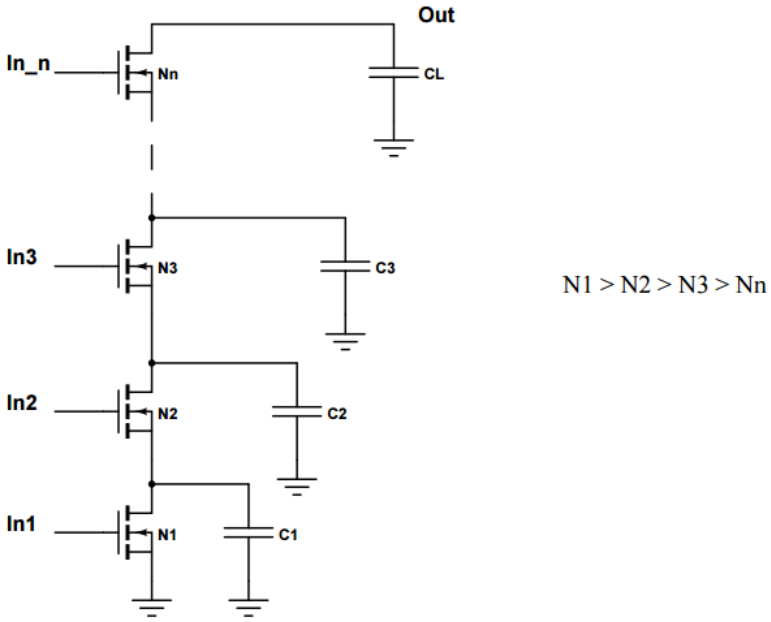
$$\frac{t_p N_{transistors}}{t_p inverter} = 1 + \varsigma\,(n-1) \tag{2.8}$$

Taking into account the fact that the total gate capacitance of the driver is proportional

to $N * W$, we have in (2.9) the power-optimal size of a transistor in a gate with N serially connected transistors (BORAH; OWENS; IRWIN, 1996).

$$W_N^* = \frac{\sqrt{\phi \frac{\mu'}{\mu} \left[1 + \varsigma \left(n-1\right)\right] C_L \left(\sum_{i=2}^{n} W_i f_i\right)}}{\sqrt{\left[\frac{k_1 N}{k} + \frac{\mu\tau}{1+\varsigma(n-1)}\right] f_1}} \tag{2.9}$$

Figure 2.3 – Progressive sizing of transistors of a n-input gate



Source: Author

After the recents *International Symposium on Physical Design* (ISPD) contests, power-driven gate sizing has aroused lot of interest. Since then, plenty of research papers have been published and new algorithms were proposed based on the ISPD 2013 contest formulation. (REIMANN; SZE; REIS, 2015) presented an approach to integrate a state-of-the-art *Lagrangian Relaxation*-based gate sizing method (FLACH et al., 2014) into a physical synthesis framework. Their sizing method achieved the first place in the *ISPD 2013 Discrete Gate Sizing Contest* with, on average, 8.78% better power results than the second place tool. This flow was also the first gate sizing method to report violation-free solutions for all benchmarks of the ISPD 2013 Contest.

2.1.2 Transistor Reorder

Often implemented together with Transistor Sizing, Transistor Reorder mainly targets on the capacitive switching power (P_{dyn}) of the transistors. "The fundamental idea of transistor reordering is to achieve lower switching power by adjusting the order of the transistors in a serial-connected CMOS chain, based on the behaviors of different inputs." (YUAN; DI, 2005).

Serially connected transistors are widely used to implement NAND/NOR gates, various complex gates, and PLA's in VLSI logic design. The preponderance of these structures in CMOS circuits produces a significant source of power dissipation. Mainly because the transition times at internal nodes are strongly dependent on the input signal characteristics and input positions. Therefore, a misguided transistor placement may increase the spurious transitions at internal nodes and result in extra power consumption. The goal of transistor reordering is to reduce the propagation delay as well as the charging and discharging of internal capacitances to achieve low power consumption.

The same input sequence could cause different power dissipation while applied to different orders of transistor chains. To determine the appropriate transistor order, we must have some information about inputs. First, one must determine the percentage of time that the specific input stays in a logic high and then determine how often this input makes a transition. Since the exact data of actual inputs is unknown beforehand, only probabilistic information is available (YUAN; DI, 2005). We can use signal probability and transition density to decide the order of transistors (SHEN; LIN; WANG, 1995; VEENDRICK, 1984). Expression (2.10) defines the signal probability; where p_i is the signal probability of input x_i, S is the total number of time slots during one clock cycle, and $x_i(k)$ is the value of x_i during the interval of time instances k and $k+1$.

$$p_i = \lim_{n\to\infty} \frac{\sum_{k=1}^{n\times S} x_i(k)}{n \times S} \tag{2.10}$$

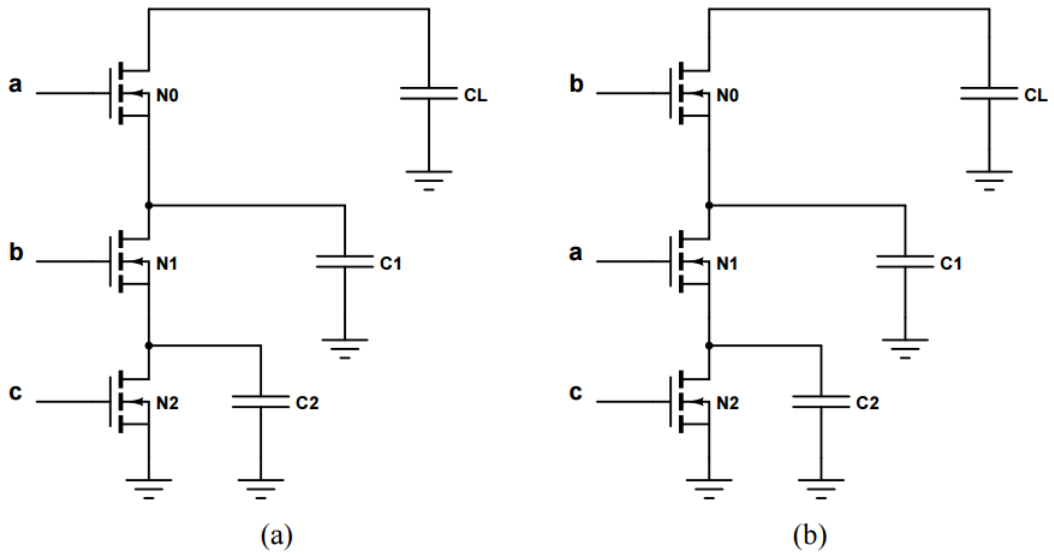
Equation (2.11) shows the transition density D_i for an input x_i.

$$D_i = \lim_{n\to\infty} \frac{\sum_{k=1}^{n\times S} \left[x_i(k)\overline{x_i(k+1)} + \overline{x_i(k)}x_i(k+1) \right]}{n \times S} \tag{2.11}$$

Figure 2.4 shows two alternate implementations of a three-input NAND gate. In both cases, node N_0 is the output of the gate, with two other internal nodes N_1 and N_2. On Figure 2.4 (a), if a vector "110" followed by "011" is applied to the inputs in alphabetic order, and all three inputs arrive simultaneously. The sequence causes the capacitances C_L, C_1 and C_2 to be

initially charged and then the capacitances C_1, and C_2 are discharged; whereas only capacitance C_2 in Figure 2.4 (b) discharges.

Figure 2.4 – Example of transistor reordering



Original three-input NAND gate (a). Gate with transistors reordered (b).

Source: Author

2.2 Voltage Scaling

Up to the mid-1990s, the standard supply voltage for all digital components was five volts, and a fixed-voltage scaling model was the norm. Back then, fixed-voltage scaling was an attractive proposition because it simplified the interfacing between different components (RABAEY, 2009). Around 1995, supply voltages dropped for the first time to 3.3V with the advent of 0.35μm technology. It dates the beginning of supply voltage scaling in correspondence with successive process nodes. On equation (1.2) is clear that the dynamic power is quadratically proportional to the supply voltage. Thus, despite the intrinsic increase in the delay, the supply voltage reduction is one of the most efficient ways to reduce power dissipation. Therefore, the scaling-down of the supply voltages has been kept until recent technology nodes. The *International Roadmap for Semiconductors* (low-power scenario) (ITRS, 2013) projects that the voltage will be reduced to 0.5V and then saturates. This may occur in the forthcoming years, when the transistors downscaling surpasses the 7nm of gate length.

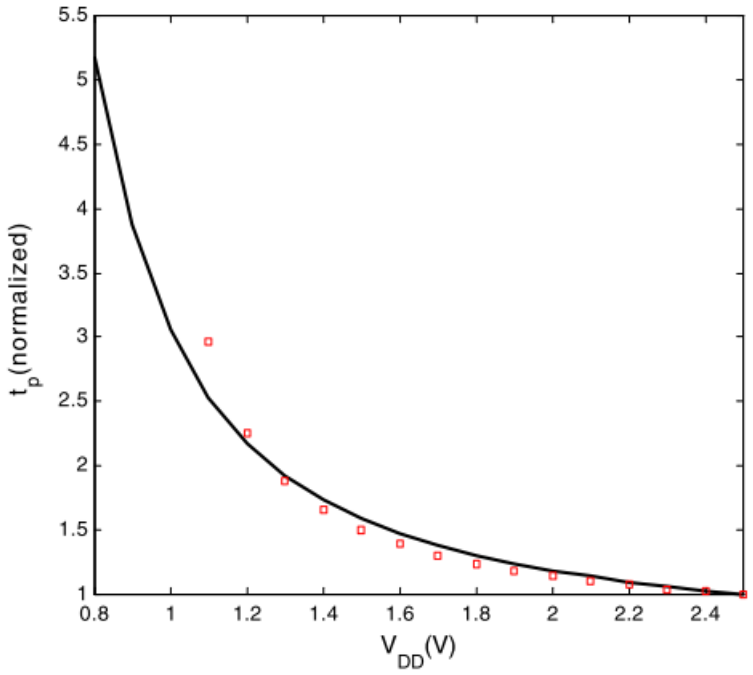
Scaling down the supply voltage requires a commensurate reduction in clock frequency

because signal propagation delays increase when the supply voltage reduces. The maximum clock frequency at which a transistor can operate is proportional to $[(V_{DD} - V_T)^{\alpha}]/(V_{DD})$, where V_T is the transistor threshold voltage and α is strongly dependent on the mobility degradation of the electrons (LI et al., 2005).

In Figure 2.5, if we move the curve to the left, the performance penalty for lowering voltage reduces. One way to address the delay problem of reduced supply voltage is by using low V_T cells or by combining cells with different V_{DD}. Moreover, dual V_{DD} (BENINI; DEMICHELI, 1998; CHANDRAKASAN; SHENG; BRODERSEN, 1992) and dual V_T (ATHAS, 1999; MERMET, 1997) techniques have been proposed. Thus, it is possible to reduce power while meeting delay constraints by using high V_{DD} with low V_T on delay critical paths, and low V_{DD} with high V_T where there is sufficient timing slack.

As supply voltage scales down, the subthreshold leakage and the leakage due to Drain Induced Barrier Lowering (DIBL) also decreases (RABAEY, 2009). The main reason for these undesired effects is the scaling of the supply voltage while keeping the threshold voltage constant. One way to address this performance issue is to scale the device threshold as well.

Figure 2.5 – Propagation delay of a CMOS inverter



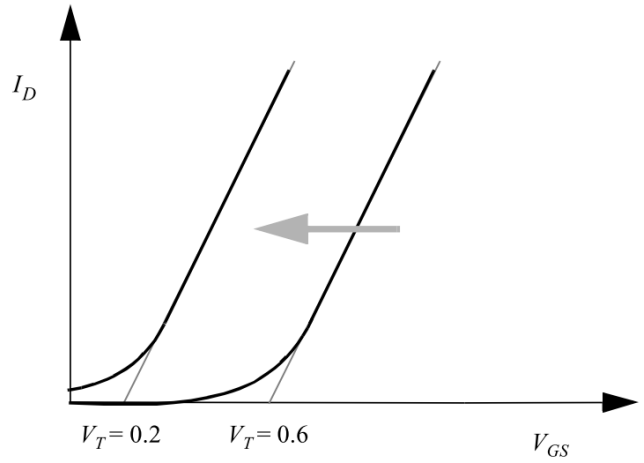
Propagation delay as a function of supply voltage (normalized with respect to the delay at 2.5 V).

Source: (RABAEY; CHANDRAKASAN; NIKOLIC, 2003)

In a CMOS inverter, the propagation delay also reduces with the decrement of the threshold voltage. Gonzalez, Gordon, and Horowitz (1997) showed that this propagation delay is

inversely proportional to $(V_{DD} - V_T)$. The dominant sources of leakage in deep-submicron process technologies with low threshold voltages are subthreshold leakage (Figure 2.6) and gate leakage. This additional leakage power dissipation reduced the benefits of supply voltage scaling.

Figure 2.6 – Subthreshold leakage



Decreasing the threshold voltage increases the subthreshold current at $V_{GS} = 0$.

Source: (RABAEY; CHANDRAKASAN; NIKOLIC, 2003)

The issue with lowering V_{DD} is that it tends to reduce I_{DS} as well, resulting in slower speeds. If we ignore velocity saturation and some of the other subtle effects that occur below 90nm, the I_{DS} for a MOSFET can be approximated by expression (2.12).

$$I_{DS} = \mu C_{ox} \frac{W}{L} \frac{(V_{GS} - V_T)^2}{2} \tag{2.12}$$

Several practical approaches have been proposed to minimize the effect of leakage current (ABDOLLAHI; FALLAH; PEDRAM, 2004; NARENDRA; CHANDRAKASAN, 2006; ROY; MUKHOPADHYAY; MAHMOODI-MEIMAND, 2003). On custom VLSI chip design, for example, *high*-V_T (slow) transistors are used to minimize the leakage current, they must be located on logical paths with enough positive slack. While *low*-V_T (fast) transistors increase the driving current and speed on critical paths. By combining supply voltage scaling optimizations and cells with different threshold voltages, we can control leakage, reduce delay and achieve significant power savings. From a design perspective, on (KEATING et al., 2007), the authors cite four different approaches for voltage scaling:

- **Static Voltage Scaling (SVS):** where different blocks or subsystems receive fixed supply

voltages.

- **Multi-level Voltage Scaling (MVS):** an extension of the static voltage scaling case where a block or subsystem is switched between two or more voltage levels. Only a few, fixed, discrete levels are supported for different operating modes.
- **Dynamic Voltage and Frequency Scaling (DVFS):** is considered as an extension of MVS where a larger number of voltage levels dynamically switches to follow changing workloads.
- **Adaptive Voltage Scaling (AVS):** an extension of DVFS where a control loop is used to adjust the voltage.

DVFS and AVS are two of the most commonly used approaches for dynamic power optimization we found in literature (BURD; BRODERSEN, 2000; BURD et al., 2000; POUWELSE; LANGENDOEN; SIPS, 2001; ZHAI et al., 2004; ELGEBALY; SACHDEV, 2007; DHAR; MAKSIRNOVI; KRANZEN, 2002; GUPTA et al., 2008). These techniques obtain the best results for voltage-frequency tradeoff.

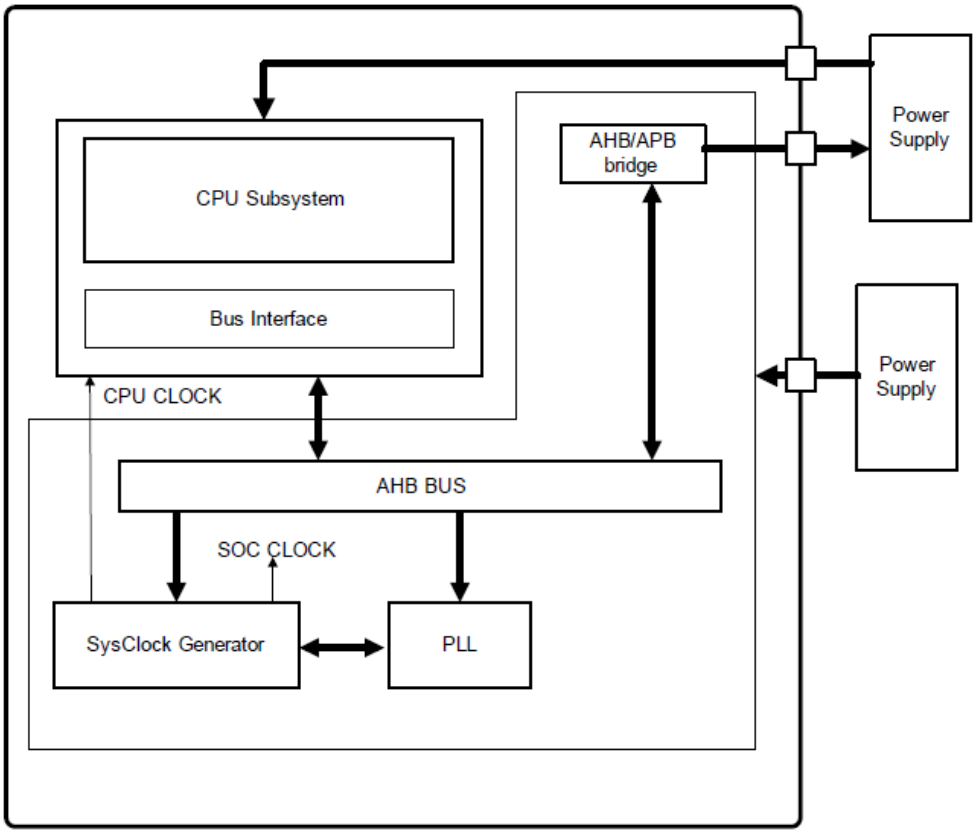
2.2.1 Dynamic Voltage and Frequency Scaling

Varying clocks and voltages during operation is a relatively new paradigm in design that demands determining which voltage and clock values to support, modeling timing and dealing with the settling time of clock generators, and power supplies (KEATING et al., 2007). Usually, the Dynamic Voltage and Frequency Scaling requires the circuit to be partitioned focusing on performance. Each area or part of the system have specific power constraints and target clock frequency. Figure 2.7 shows an example of a DVFS design. A programmable power supply drives the CPU sub-system. A fixed supply voltage powers the rest of the chip. Also, a Phase-locked Loop (PLL) provides a high-speed clock to the *SysClock Generator*, which uses dividers to generate the *CPU CLOCK* and the *SOC CLOCK*. For voltage and frequency scaling, a software must decide the minimum CPU clock speed that meets the workload requirements. It then determines the lowest supply voltage that will support that clock speed (KEATING et al., 2007).

During its operation, the system may face two possible situations: a high workload that requires an increase in frequency and other where a lower frequency could be used. If the target clock frequency is higher than the current frequency, the CPU programs the power supply to the new voltage and continues operating at the current clock frequency until the new voltage settles. The CPU can program either the SysClock Generator or the PLL to adjust the clock

frequency when required. On the other case, if the target clock frequency is lower than the current frequency, the CPU programs the new clock frequency and changes the configurations for the SysClock Generator, and the PLL if required. The CPU then selects the power supply to the new voltage and continues operating at the new clock frequency while the voltage settles to the new value.

Figure 2.7 – Block diagram of a DVFS example



Source: (KEATING et al., 2007)

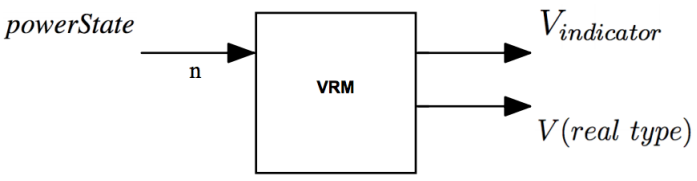
Most DVFS systems use a set of discrete voltage/frequency pairs. Determining which values to support is a crucial design decision, and it is highly application dependent. (YAO; DEMERS; SHENKER, 1995) presented an algorithm for voltage selection that takes into account the delay, the execution time limit and the number of CPU cycles required to perform a task. The technique they proposed, computes the speed of execution, determining the start and end times of each task. Another way to solve the problem of choosing the better voltage level was proposed in (SHIN; KIM; LEE, 2001); where the voltage is set during the execution of the task and the number of CPU cycles required to perform the task may change during execution.

2.2.2 Adaptive Voltage Scaling

At lower supply and threshold voltages, the delay and the energy become more sensitive to variations in V_{DD} and V_T. Thus, the control of both voltage levels is critical. A tough task because, on real circuits, there is always a small uncertainty in the value of the supply and threshold voltage. The main sources of uncertainty are the operation conditions of the circuit and the variability of the manufacturing process. These variations cause the delay and energy to be spread out over a range, reducing the overall efficiency of the circuit. One way to minimize the effect of uncertainty is by using adaptive techniques to regulate the supply and threshold voltages. That is, dynamically adjust the supply and/or threshold voltage such that the circuit meets the required specifications (DHAR; MAKSIRNOVI; KRANZEN, 2002; ELGEBALY; SACHDEV, 2007; GONZALEZ; GORDON; HOROWITZ, 1997; USAMI; HOROWITZ, 1995; VIERI et al., 1995).

Chinnery and Keutzer (2007) shows an example of adaptive voltage scaling. Figure 2.8 depicts a diagram of a simple Voltage Regulator Module (VRM) with the ability to control the voltage rails in an analog-like fashion. The output voltage $V(real\ type)$ is controlled by the input *powerState*. A binary voltage indicator ($V_{indicator}$) signals when the requested voltage is stable at the output. This model enables a continuous change in the voltage in real time. A new voltage level is sought by *powerState*, and then the VRM will set the indicator to "0" and move the output voltage to the required level. Once the VRM reaches the new voltage level, the indicator is set to "1". With this signal we can control other aspects of the design; for instance, it may assist with the control of clock frequencies. Under higher performance demands, the voltage raises followed by an increment in frequency. The voltage indicator can be used to signalize that is safe to increase the clock frequency once the requested voltage has been achieved and is stable.

Figure 2.8 – Voltage regulator module (VRM)



Source: (CHINNERY; KEUTZER, 2007)

2.3 Logic Optimizations

The EDA tools provide the designers with a variety of tweaking options to improve the performance of a system for low power scenarios. The technology-independent part of the logic synthesis process consists of a sequence of optimizations that manipulate the network topology to minimize delay, power, or area. On the physical phase, the object of technology mapping is to transform a technology-independent logic description into an implementation in a specific technology.

At the logic level, the EDA tools automatically choose the network topology to implement a particular function. Several optimizations can be carried out during the logic synthesis, most of them contribute to reducing the power consumption. For example, from a performance perspective, it is a good idea to connect the most critical signal to the input pin closest to the output node; on the NAND gate of Figure 2.4, it is the N0 transistor. This is just one example of the multiple modifications that the RTL code suffers during the logical synthesis phase. These optimizations include: factoring; restructuring to minimize spurious transitions; buffer insertion/removal for path balancing and timing constraints; and *don't-care* optimization.

It seems obvious that a simpler logical expression would require less power as well. For instance, translating the function $f = a.b + a.c$ into its equivalent $f = a(b + c)$ seems a no-brainer, as it requires one less gate. However, this factorization may also introduce an internal node with substantially higher transition probabilities. Actually, this may increase the net power. Hence, power-aware logical synthesis must not only be aware of network topology and timing, but also should incorporate parameters such as capacitance and activity. In the end, the goal is to derive the *pareto-optimal* energy-delay curves (RABAEY, 2009). It is, we need to choose the network that minimizes power for a given maximum delay or reduces the delay for maximum power.

The algorithms for logic synthesis manipulate the logic equations to reduce the fan-in requirements and hence reduce the gate delay. For example, the quadratic dependency of the gate delay on fan-in makes a six-input NOR gate extremely slow. Partitioning the NOR-gate into two three-input gates results in a significant speed-up (RABAEY; CHANDRAKASAN; NIKOLIC, 2003). However, this new topology could increase total power consumption. Restructuring the logic network to have paths with similar lengths can reduce timing problems, power consumption, and spurious logic transitions. Also, dynamic hazards could be minimized by balancing the system in terms of timing (LAVAGNO; KEUTZER; SANGICIVANNI-VINCENTELLI, 1991).

Following the same approach, the network can also be balanced by the insertion or re-

moval of buffers. The buffers themselves add extra switching capacitance. Hence, as always, buffer insertion is a careful trade-off process. Analysis of state-of-the-art synthesis tools have shown that simple buffers handle a considerable part of the overall power budget of the combinatorial modules (RABAEY, 2009).

2.3.1 Technology Mapping

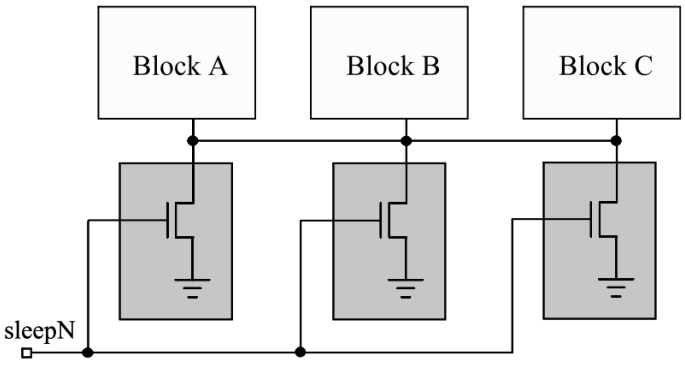
Technology mapping makes the link between logic synthesis and physical design. The object of technology mapping is to transform a technology-independent logic description into an implementation in a target technology. During this phase, the EDA tools select the gates and choose the appropriate size for them based on the design constraints. One of the key operations during technology mapping is to recognize logic equivalence between a portion of the initial logic description and an element of the target technology.

Different combinations of cells can implement a gate with distinctive switching activities, capacitance, power and delay (OZDAL; BURNS; HU, 2012). To implement an XOR2, an AO22 with inverters may be smaller and more energy-efficient, but slower. Refactoring to eliminate common subexpressions reduces the number of operations, therefore, the switching activity. Balancing path delays can decrease the glitch activity. Hence, the technology mapping for low power can take advantage of other approaches, as the ones cited previously, to improve the results. Also, during technology mapping, a gate sizing process chooses the best cells for meet timing constraints and minimize power consumption. To reduce delay, gates on critical paths are upsized, increasing their capacitance. In turn, the gates fan-in must be incremented to drive the larger capacitance. This result in oversized gates and buffer insertion on the critical paths (CHINNERY; KEUTZER, 2007).

2.4 Power Gating

Leakage power dissipation grows with every generation of CMOS process technology due to the scaling transistor threshold voltages. This leakage power is not only a serious challenge for battery-powered devices, but also an issue in tethered equipment such as servers, routers, and set-top boxes. To reduce the overall leakage power of the chip, it is desirable to add mechanisms to turn off blocks that are not required. In consequence, Power Gating provides two power modes: a low power (or inactive mode) and an active mode (KEATING et al., 2007).

Figure 2.9 – Global power gating topology



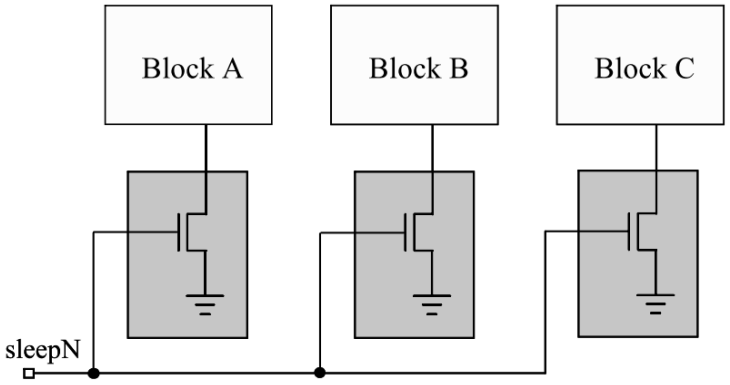
Source: (CHINNERY; KEUTZER, 2007)

Power gating consists of selectively powering down certain blocks in the chip while keeping other blocks powered up. The goal of power gating is to minimize leakage current by temporarily switching power off to blocks that are not required in the current operating mode. The use of this power optimization strategy is justified by the fact that on real systems, periods of activity are followed by relatively lengthy periods of inactivity. Moreover, while the power consumption during the active period is mostly due to the dynamic power, leakage power represents the major energy consumption during inactive or standby periods.

The most basic form of power gating control is an externally switched power supply. For example, on a CPU that has a dedicated *off-chip* power supply which provides power only to the CPU, we can shut down this power supply and reduce the leakage current in the CPU. This approach, though, takes the longest time and requires the most energy to restore power to a gated block. A better approach could be power gating internally, where internal switches control the power of selected blocks. Hence, a critical decision in power gating is how to switch power. In general, there are two approaches: fine grain power gating and coarse grain power gating (KEATING et al., 2007). In fine grain power gating the switch that controls the power is inside each standard cell in the library. In coarse grain power gating, a block of gates has its power switched by a collection of switch cells.

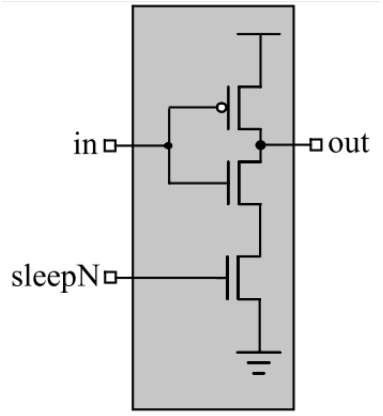
Based on these approaches, Power Gating can be implemented using several different topologies, such as: *global power gating*, *local power gating*, and *switch-in-cell power gating*. *Global power gating* refers to a logical topology in which multiple switches connect to one or more blocks of logic, and all the power-gated logic blocks share a single virtual ground. Figure 2.9 shows an example of this approach, where the three logic blocks share a virtual ground, and each sleep domain is controlled by a particular sleep enable signal. This topology is only effective for large logic blocks, and when a single sleep signal is enough to control all of them.

Figure 2.10 – Local power gating topology



Source: (CHINNERY; KEUTZER, 2007)

Figure 2.11 – Switch-in-cell power gating topology



Source: (CHINNERY; KEUTZER, 2007)

Local power gating specifies a logical topology in which each switch singularly gates its virtual ground connected to its own group of logic, as illustrated in Figure 2.10. Here, each logic block uses its particular switch, sharing the sleep signal.

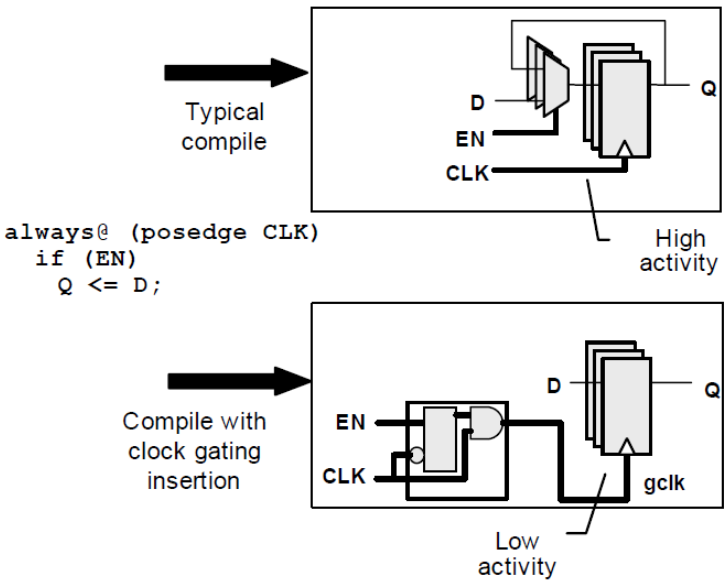
Switch-in-cell may be thought of as an extreme form of local power gating implementation. In this topology, each logic cell contains a specific switch transistor (as Figure 2.11 depicts). This approach has several notable advantages and disadvantages. As primary advantages, delay calculation is very straightforward since each cell is timing characterized with its dedicated internal switch. However, the additional transistor in the pulldown stack and the need to size up the previously existing logic transistors to compensate for the additional device in the stack, can increase the chip's area.

2.5 Clock Gating

In CMOS digital circuits, the sequential part is the major contributor of the total power dissipation. A significant fraction of the dynamic power is due to the distribution network of the clock. The clock is the only signal that switches all the time. It is also mostly likely to drive a heavy load. Clock buffers spend up to 50% or even more of the dynamic power (CHINNERY; KEUTZER, 2007). In addition, the flops receiving the clock dissipate some dynamic power even if the input and output remain the same. Thus, reducing the clock power is an efficient way to minimize the total power dissipation. The most common way to do this is by turning clocks off when they are not required. This approach is known as clock gating (YUAN; DI, 2005).

Modern design tools support automatic clock gating, they can identify circuits where clock-gating circuitry may be inserted without changing the function of the logic. Figure 2.12 exemplifies the insertion of clock gating. In the original RTL, the register updates independently of the variable (EN); when applied clock gating, this signal controls the state of the register.

Figure 2.12 – Clock Gating



Source: (KEATING et al., 2007)

This power optimization technique is transparent to the RTL designer. However, in the early days of RTL design, engineers would code clock-gating circuits explicitly in the RTL. This approach was error prone because it is very easy to create a clock gating circuit which glitches during gating (KEATING et al., 2007). Today, most libraries include specific clock

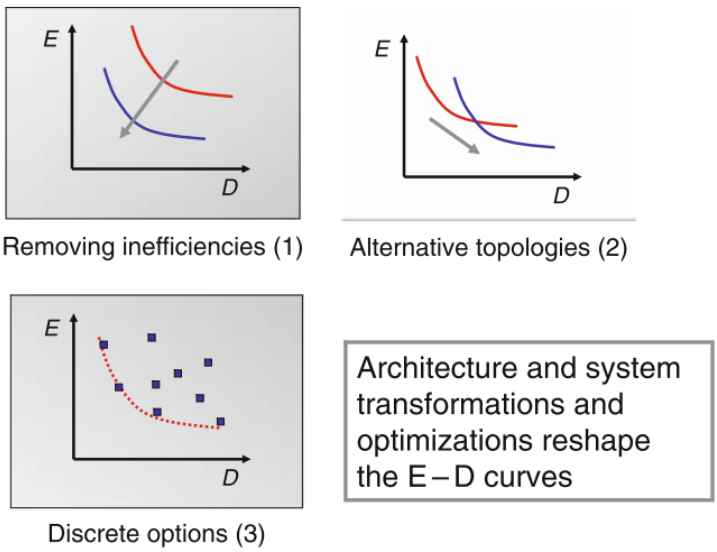
gating cells that the synthesis tool recognizes. The combination of explicit clock gating cells and automatic insertion makes clock gating a simple and reliable way of reducing power (WU; PEDRAM; WU, 2000).

2.6 Architecture-level Optimizations

Optimizations at the architecture or system level can enable more efficient power minimization while maintaining performance. At higher abstraction levels, design modifications tend to have a greater potential impact, allowing circuit level techniques such as voltage scaling (ZHAI et al., 2004) or gate sizing (FLACH et al., 2013) to be more effective. Moreover, it is well known that the *Return-on-Investment* (ROI) increases at higher levels of the design process. While circuit techniques may yield improvements in the 10-50% range, architecture and algorithm optimizations have reported power reduction by orders of magnitude (RABAEY, 2009).

At the circuit level, the optimization parameters are mostly continuous. We have to decide the optimum gate/transistor size or the best supply and threshold voltage levels for low power scenarios. At the architecture-level, the choices are rather discrete. For instance, we dispose of several adder topologies to choose the one that best fit the design constraints. These discrete choices contribute to expanding the energy-delay space.

Figure 2.13 – Energy-delay tradeoff



Source: (RABAEY, 2009)

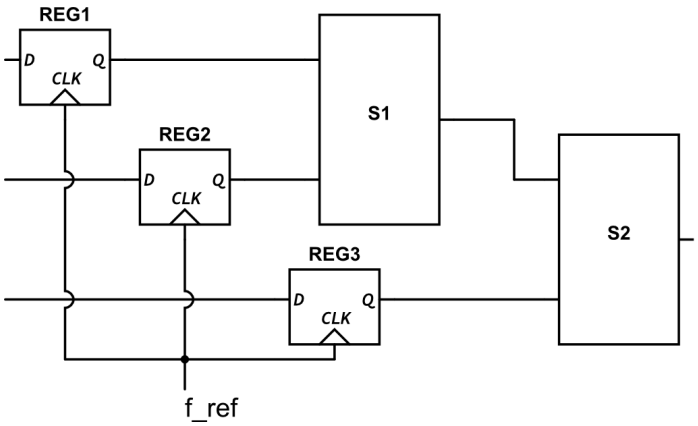
Figure 2.13 plots three scenarios for *energy-delay* tradeoff. Following the previous example, if two adder topologies are available, each of them comes with its optimal energy-delay curve (Figure 2.13, plot 2). When one topology is superior, offering the best energy-delay tradeoff, the selection process is quite straightforward (plot 1). Finally, a third scenario consists of many discrete options such as the sizes and the number of register files (plot 3).

2.6.1 Concurrency

Concurrency is a property of systems in which several operations are executing simultaneously, and potentially interacting with each other. A system that exploits parallel approaches, may have different clock domains and particular energy demands. Depending on the requirements of the system and its topology, we can enable aggressive supply voltage scaling. It also improves performance at a fixed *energy per operation* (EOP) (CHANDRAKASAN; SHENG; BRODERSEN, 1992).

To better understand the advantages of concurrency, consider the design of Figure 2.14, which operates at a nominal supply voltage ($V_{DD_{ref}}$) and a specific frequency f_{ref}. S1 and S2 are combinational logic blocks, and C_{ref} is the average switched capacitance. A parallel implementation of the same design, as the one on Figure 2.15, replicates the design such that parallel branches process interleaved input samples; then a multiplexer recombines the outputs.

Figure 2.14 – Concurrency example design



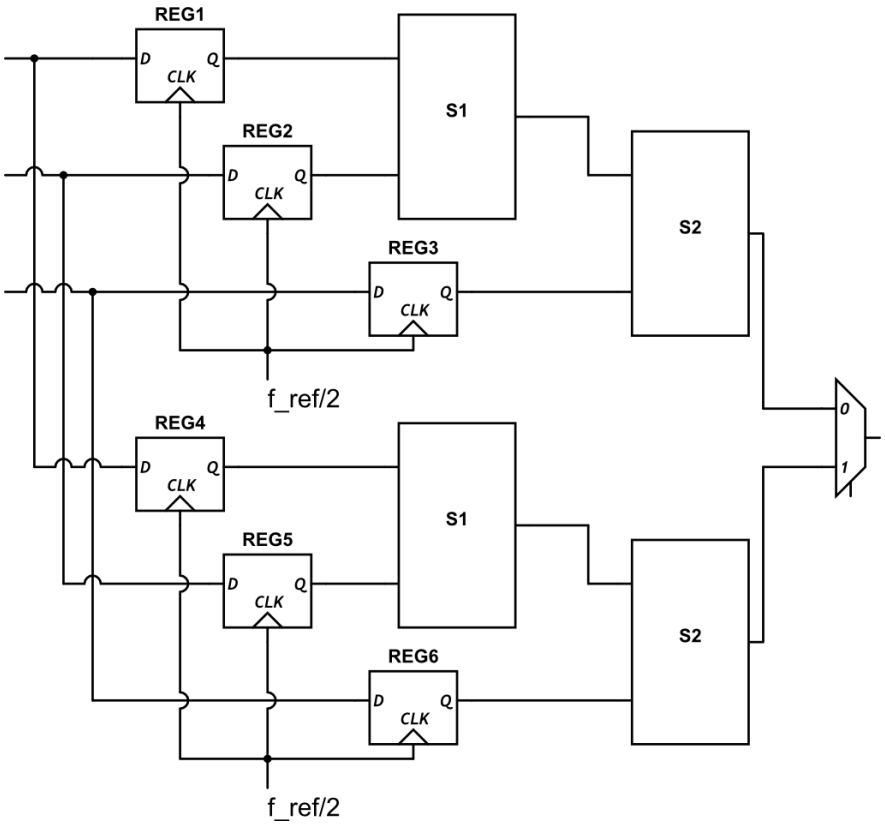
S1 and S2 contain adders, ALUs, etc.
The total power is $P_{ref} = C_{ref}(V_{DD_{ref}})^2 f_{ref}$

Source: (CHANDRAKASAN; SHENG; BRODERSEN, 1992)

Thanks to the parallelism, the new design can operate at half the speed; hence the fre-

quency becomes $f_{par} = f_{ref}/2$. Loosen the delay constraint enables a reduction of the supply voltage by a ϵ_{par} factor, and the total power is now $P_{par} = C_{ref}(V_{DD_{ref}}/\epsilon_{par})^2 f_{par}$. With this implementation, the overhead of switching capacitance is minimal but the increase in area is substantial. The impact of introducing concurrency to reduce EOP for a fixed performance hinges on the relationship between the supply voltage and delay as explained before.

Figure 2.15 – Parallel Implementation



Source: (CHANDRAKASAN; SHENG; BRODERSEN, 1992)

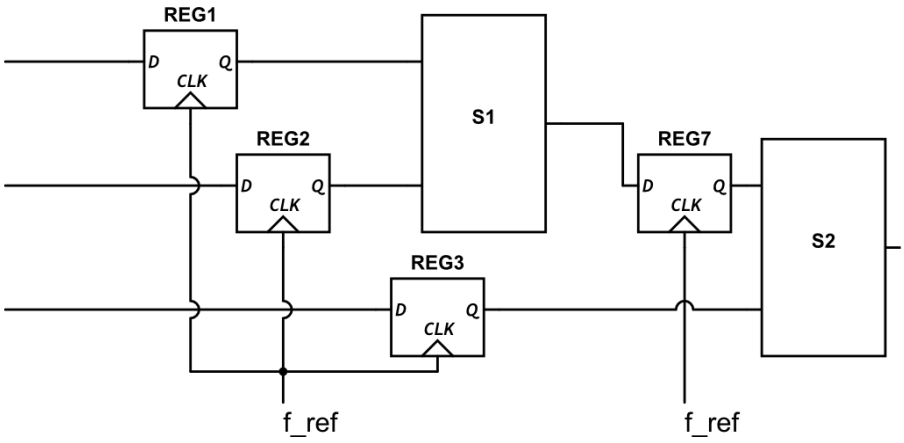
2.6.2 Pipelining

Another form to introduce concurrency in a design is with pipelining; it can be equally effective in reducing the supply voltage, and hence reducing power dissipation. Pipelining improves throughput at the cost of latency by inserting extra registers between logic gates. It employs the registers to break up a datapath into multiple stages, each stage storing the intermediate results.

Figure 2.16. depicts a possible pipelining implementation of the previous example. The area overhead of pipelining is much smaller than that of parallelism, the only cost being the

extra registers, compared to replicating the design and adding multiplexers (RABAEY, 2009). The switching power in registers is the major part of power dissipation and pipelined implementations typically come with a higher switched capacitance than parallel designs.

Figure 2.16 – Pipelining Implementation



Source: (CHANDRAKASAN; SHENG; BRODERSEN, 1992)

An ideal low-power design methodology employs some or all previously presented techniques, with the objective to reduce power in all abstraction levels and obtain greater energy savings.

With pipelining and parallelism, digital systems can achieve the same performance at lower clock frequencies. The timing slack is useful to reduce the power with the aid of a lower supply voltage, a higher threshold voltage, and reduced gate sizes. Some parts of the chip may operate at lower supply voltage, and those that require high-performance will maintain a higher voltage level. In consequence, concurrency also enables the possibility to use multiple voltages for controlling the power of each region independently. Systems with multiple supplies require a special design methodology and considerations as we can see in the next chapter.

3 MULTI-VOLTAGE DESIGNS

On systems that exploit concurrency, the logic blocks can operate at different frequencies. Each one having peculiar energy demands to fulfill the clock speed requirements. In addition, some of those blocks do not operate at all when the system is in standby mode. Several approaches take advantage of these characteristics to implement distinct voltage level regions (KULKARNI; SRIVASTAVA; SYLVESTER, 2004; LI et al., 2005; TERRES et al., 2014).

Multi-Voltage techniques are widely used inside high-performance systems such as modern processors. Where the CPU runs at a high voltage because of its performance demands and the rest of the chip can run at lower voltage without affecting overall system performance. Multi-Voltage takes advantage of the existence of different blocks inside the IC, with different target performance, objectives, and constraints. The energy demand of those blocks also varies under certain conditions; thus, the internal logic is partitioned into multiple voltage regions or power domains, each with its power supply. Mixing blocks at different V_{DD} adds some complexity to the design; from the floor-planning to the routing phase, Multi-Voltage approaches generate new design variables and, in some cases, they become a constraint.

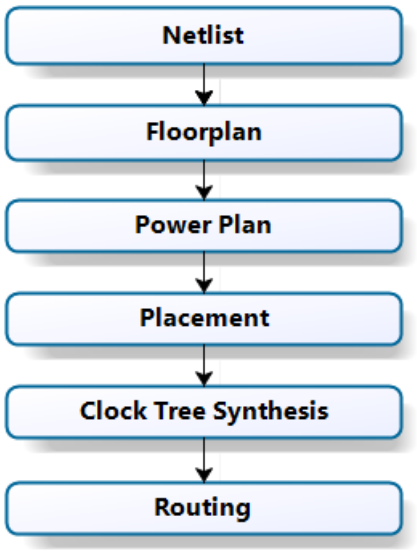
The projection of a distribution network that supplies the appropriate power and ground nets to all the instances of the design is known as the *Power Planning*. During this phase, the power grid is designed. As Figure 3.1 shows, Power Planning carries out before the placement of the cells. In a typical back-end flow, it is an intermediate phase between the Floor-planning and the Placement.

The output of the Power Plan (the power grid) contains information about the pads that supply power to the chip; the rings around the periphery of the die that carry power to the standard cells and macros; and the rails and trunks that cross the entire die or section of the die. With the help of automated tools, the designers must guarantee that the chip is robust enough to resist and to mitigate power-related issues such as:

- The voltage drop across a chip's power network caused by current and resistance, or *(IR) drop*.
- *Electromigration* (EM), the mechanical failure of metal wires because of metal atoms migrating over an extended period of time due to high current densities (RABAEY; CHANDRAKASAN; NIKOLIC, 2003).
- Crosstalk-induced delay caused by lower supply voltages.

In the previous chapter, we talked about some optimization strategies for low power

Figure 3.1 – Typical back-end flow



Source: Cadence ™

consumption. Among them, power gating and voltage scaling rely on moving away from the traditional approach of using a single, fixed supply rail for all of the gates in the design. As explained before, a reduction in V_{DD} considerably lowers the power dissipation of the circuit. Accordingly, Multi-Voltage design exploits this concept to reduce power consumption. Most of the works we found in the literature employ two levels of supply voltage, one for low voltage and other for high or normal voltage. Moreover, some designs may include a third level for the *off* or *standby* state (0V); thus, we adopt the following terminology for the three supply levels:

- $\boldsymbol{V_{DDH}}$: Normal supply voltage. Value depends on the technology.
- $\boldsymbol{V_{DDL}}$: Relative lower supply voltage. 50% of the normal supply voltage ($\boldsymbol{V_{DDH}}$) (KULKARNI; SRIVASTAVA; SYLVESTER, 2004).
- $\boldsymbol{V_{off}}$: Supply voltage equals zero (0V).

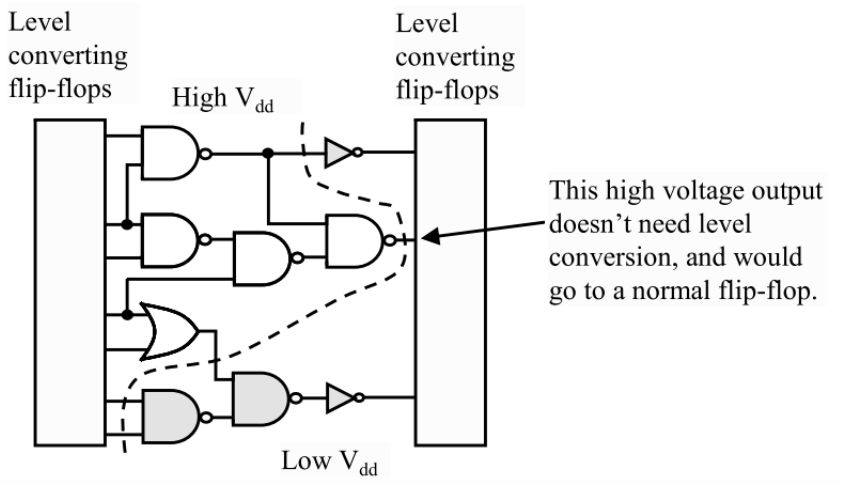
During the power planning, the power rails are distributed according to the design requirements. So that the cells that belong to critical paths are supplied with a nominal voltage (V_{DDH}) and the cells that are not part of the critical paths will have a lower supply voltage (V_{DDL}). In the same way, cells that are not being used can be associated with the standby mode caused by applying a neutral voltage (V_{off}). We found in the literature two major algorithms for V_{DD} assignment they are the *Clustered Voltage Scaling* (CVS) (USAMI; HOROWITZ, 1995) and the *Extended Clustered Voltage Scaling* (ECVS) (USAMI et al., 1998) algorithms.

3.1 Clustered Voltage Scaling

The starting point of the Clustered Voltage Scaling algorithm is a design with all cells supplied with nominal voltage V_{DDH}. Then, it assigns lower supply voltage (V_{DDL}) on the gates that do not belong to the critical paths based on the available timing slack. One restriction of the CVS voltage assignment is that V_{DDL} cells should not directly feed V_{DDH} cells; therefore, level conversion is implemented only at sequential boundaries (e.g., flip-flops). The level conversion functionality can be embedded into the flip-flops that delimit a logic cell and is referred to as a *Level Converting Flip-Flop* (LCFF) (CHINNERY; KEUTZER, 2007).

Figure 3.2 depicts a final topology attained by CVS. The algorithm partitioned the circuit into two clusters, one having only V_{DDH} cells and the cluster formed with V_{DDL} cells. The resulting netlist has no V_{DDH} cells driven by V_{DDL}, respecting the primary CVS constraint.

Figure 3.2 – Clustered Voltage Scaling



Source: (CHINNERY; KEUTZER, 2007)

CVS maintains a list (referred to as L) of candidate cells that can be assigned to lower supply voltage. The initial implementation of CVS (USAMI; HOROWITZ, 1995) used a heuristic that ordered the cells in the list L on the basis of their slack. The set of gates that drive the circuit primary outputs initializes the list. This initialization process is the step "SET V_{DDL}" of the pseudo-code presented in (1). As the CVS algorithm proceeds, new cells are added to the L list. The step "CONSTRAINED TOPOLOGY" guarantees that there will be no V_{DDL} feeding directly a V_{DDH} driven gate. Producing a final netlist with reduced number of voltage converters.

The CVS considers only two possible voltage levels, V_{DDH}, and V_{DDL}. Restricting the

design cells to operate at maximum performance or low-power mode. Usami and Horowitz (1995) tested the algorithm on a microprocessor and, combining it with the gate sizing technique, they achieved an economy of 10-20% in power.

```
CVS() {
minimum power found = power of initial V_DDH circuit ;
Best configuration = all V_DDH assignment ;
L = gates that only drive circuit primary outputs;
while (L is non-empty) do
    STEP: "SET V_DDL" ;
    Select candidate A from L ;
    Remove A from L ;
    Set the supply voltage of A to V_DDL ;
    if (A drives a primary output) then
        insert an LCFF ;
    end
    Check timing ;
    if (circuit still meets timing constraints) then
        STEP: "CONSTRAINED TOPOLOGY" ;
        Add to L gates that fan into A but not into any V_DDH gate ;
        Check power consumption ;
        if (power < minimum power found) then
            minimum power found = power ;
            Best configuration = current V_DDL assignment ;
        end
    else
        Remove any added LCFFs ;
        Set the supply voltage of A back to V_DDH;
    end
end
}
```

Algorithm 1: Pseudo-code for the CVS algorithm (USAMI; HOROWITZ, 1995)

3.2 Extended - Clustered Voltage Scaling

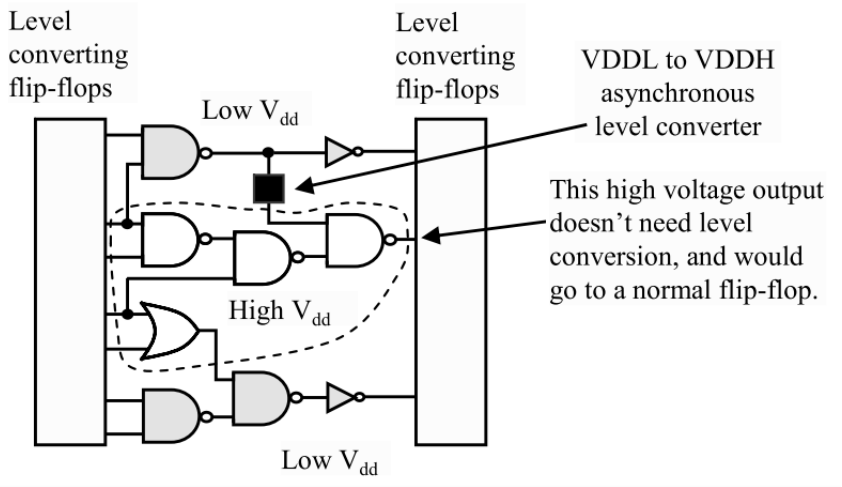
The Extended Clustered Voltage Scaling technique presented in (USAMI et al., 1998) is an extension of the CVS algorithm. Like its predecessor, the ECVS aims at using the available timing slack in a circuit by applying a lower supply voltage on gates that are off the critical paths. However, ECVS differs from CVS in the policy it follows to make the power supply assignment. Subsequently, the final structure of the resulting netlists also diverges.

ECVS relaxes the topological constraint of the CVS and allows a V_{DDL} driven cell to feed a V_{DDH} cell. This is not a straightforward process because those new interfaces require a

level conversion. The level converters impose penalties in terms of their delay, power, and area. On the contrary, ECVS may lead to higher energy savings because bigger portions of the circuit can be clustered into low voltage regions. By optimizing the insertion of level converters, it can increase the number of gates becoming V_{DDL} without significant increase in the number of level converters.

Figure 3.3 shows the final topology of a given circuit after applied ECVS. The resulting netlist has low-power cells driving V_{DDH} cells with the aid of voltage scaling interfaces known as *Asynchronous Level Converters* (ALC).

Figure 3.3 – Extended Clustered Voltage Scaling



Source: (CHINNERY; KEUTZER, 2007)

The ECVS algorithm (2) maintains a list L of candidate cells that will have V_{DDL} supply. The set of gates that drive the circuit primary outputs initialize the list and are connected to Level Converting Flip-Flops. As the execution of the ECVS algorithm continues, new cells are added to the L list. The algorithm then determines if the fan-out of the current gate contains V_{DDH} gates to insert Asynchronous Level Converters between them. The major difference of the ECVS algorithm, when compared with the CVS, is the step that restrict the final topology. It allows the interspersing of V_{DDL} and V_{DDH} cells with ALCs as interfaces.

On their work, (USAMI et al., 1998) describe a completely automated technique to reduce power that consists of structure synthesis, placement, and routing. After selecting the cells with V_{DDH} and those that belong to the non-critical paths. The placement and routing tool assigns either the reduced voltage or the normal one to each row so as to minimize the area overhead. The reduced supply voltage is also exploited in a clock tree to reduce power. They applied the combination of these techniques to a media processor chip, achieving a power reduction of

47% in random-logic modules and 73% in the clock tree, while keeping the performance.

```
ECVS() {
minimum power found = power of initial V_DDH circuit ;
Best configuration = all V_DDH assignment ;
L = gates that only drive circuit primary outputs;
while (L is non-empty) do
    STEP: "SET V_DDL" ;
    Select candidate A from L ;
    Remove A from L ;
    Set the supply voltage of A to V_DDL ;
    if (A drives a primary output) then
        Insert an LCFF ;
    end
    for (each gate B ∈ fan-outs[A]) do
        if (supply of B = V_DDH) then
            Insert an ALC on the path from A to B ;
        end
    end
    Check timing ;
    if (circuit still meets timing constraints) then
        STEP: "LEVELIZED" ;
        Add to L gates that fan into A but not into other gates that have already
        been considered or primary outputs ;
        Check power consumption ;
        if (power < minimum power found) then
            minimum power found = power ;
            Best configuration = current V_DDL assignment ;
        end
    else
        Remove any added LCFFs or ALCs ;
        Set the supply voltage of A back to V_DDH;
    end
end
}
```

Algorithm 2: Pseudo-code for the ECVS algorithm (USAMI et al., 1998)

3.3 Greedy Extended - Clustered Voltage Scaling

In order to reduce the number of level converters employed by the Extended - Clustered Voltage Scaling, the group of V_{DDL} cells driving V_{DDH} cells must be reduced. When this happens, power and area are further optimized. Clustered approaches are most effective when they can find "groups" or "clusters" of connected gates to assign the lower supply. This is

true since the arrangement will require fewer Asynchronous Level Converters and minimize their resulting overhead. To reduce the ALCs number, we can implement a sensitivity measure that uses the information available in the slack distribution of the circuit and the power savings attainable before finalizing each V_{DDL} assignment (CHINNERY; KEUTZER, 2007).

$$Sensitivity(x) = \frac{\Delta Power \times Slack\ at\ gate\ output}{\Delta Delay} \tag{3.1}$$

GECVS() {
minimum power found = power of initial V_{DDH} circuit ;
Best configuration = all V_{DDH} assignment ;
L = gates that only drive circuit primary outputs;
while *(there are feasible moves)* **do**
  for *each V_{DDH} gate 'A'* **do**
    Set A to V_{DDL} ;
    if *A drives a primary output* **then**
      Insert an LCFF ;
    end
    update_vicinity() ;
    Calculate sensitivity for A using Equation 3.1 ;
    Set A back to V_{DDH} ;
    update_vicinity() ;
  end
  Select the maximum sensitivity gate 'B' that meets timing ;
  Check power consumption ;
  if *power < minimum power found* **then**
    minimum power found = power ;
    Best configuration = current V_{DDL} assignment ;
  end
end
}

Algorithm 3: Pseudo-code for the GECVS algorithm (CHINNERY; KEUTZER, 2007)

The Greedy Extended - Clustered Voltage Scaling (GECVS) algorithm (KULKARNI; SRIVASTAVA; SYLVESTER, 2004) includes a sub-routine (*update_vicinity( )* of Algorithm 3) to remove ALC in particular non-critical logical paths with enough timing. This sub-routine also changes the slack of various paths in the circuit. The overall power dissipation of the circuit will also change as a result of the move (AC/LCFF insertion or removal). Equation 3.1 represents the sensitivity for a gate *x*, where $\Delta Power$ is the change in total power due to a move and $\Delta Delay$ represents the variation in arrival time at the gate output. The slack term in the sensitivity computation acts as *a weighting factor to encourage V_{DDL} assignment* for gates with more slack. Evaluating this sensitivity for a gate only requires the rise/fall transition and arrival times at the inputs of the gates that feed it (CHINNERY; KEUTZER, 2007). The sensitivities

for all gates that can undergo V_{DDL} assignment are evaluated at every iteration of the algorithm and then the algorithm selects the move with the best result (maximum sensitivity). It enables the possibility to choose the move that gives *the best power savings per unit delay penalty* (CHINNERY; KEUTZER, 2007).

The problem with the GECVS algorithm is that, in the long run, it allows negative moves (negative sensitivity), uncovering better solutions. Its main characteristic is the possibility of generating V_{DDL} cell groups from the beginning of the paths to the end of a chain of combinational logic. In contrast, the CVS and ECVS algorithms tend to create most of the clusters near the primary outputs of the circuit and continue backwards with the V_{DDL} assignment. This makes GECVS more flexible than CVS or ECVS. The vicinity calculations are not exact (as compared to static timing analysis), but this filtering significantly reduces the number of gates to be tried in the final timing check. The final timing checks are with static timing analysis over the fully affected region, to confirm that the delay target is met. (CHINNERY; KEUTZER, 2007) presents in Table 3.1 the power consumption results for two V_{DDL} power supplies of different circuits after applying CVS and GECVS.

Table 3.1 – Comparison of power savings using CVS and GECVS versus the original design with all gates being at V_{DDH} and low V_T

	$V_{DDL} = 0.6$V		$V_{DDL} = 0.8$V	
Circuit	**CVS**	**GECVS**	**CVS**	**GECVS**
c432	1.0%	1.5%	0.8%	0.8%
c880	8.2%	10.3%	15.0%	21.3%
c1355	0%	0%	0%	1.0%
c1908	4.3%	7.7%	3.4%	8.4%
c2670	21.1%	25.5%	16.5%	25.0%
c3540	3.2%	8.3%	2.9%	9.7%
c5315	7.6%	19.0%	8.3%	22.0%
c7552	14.9%	20.2%	22.0%	28.8%
Huffman	6.6%	12.7%	6.7%	14.4%
Average	**7.4%**	**11.7%**	**8.4%**	**14.6%**

Source: (CHINNERY; KEUTZER, 2007)

(KULKARNI; SRIVASTAVA; SYLVESTER, 2004) compares the three algorithms (CVS, ECVS, and GECVS). The authors made a comprehensive analysis of achievable power savings accounting the overhead due to level conversion. For the CVS, they studied various combinational benchmark circuits and thus incorporated the LCFF penalties by considering them to consume a fixed portion of the total delay budget (cycle time). In the case of ECVS, Asyn-

chronous Level Converters (ALCs) are required in addition to LCFFs. Table 3.2 and Table 3.3 summarize the dynamic power achieved by the described algorithms for the various benchmark circuits with V_{DDL}=0.6V and 0.8V. The values are presented in percentage with reference of the initial design (all cells at V_{DDH}) and obtained from two situations of the minimum achievable delay (10% and 20% timing relaxation referred as "backoff points"). We observe that the GECVS algorithm outperforms both algorithms (CVS and ECVS) for all benchmarks, values of VDDL, and delay *backoff*. Also, both ECVS and GECVS perform significantly (in some circuits providing approximately twice the power savings or more) better than CVS because of the greater V_{DDL} assignment. On average, circuits optimized with GECVS show 6-8% lower power than with ECVS; and 11-16% lower power than those with CVS. ALCs consume on average 8-10% of total power across benchmarks.

Table 3.2 – Comparison of CVS, ECVS, and GECVS algorithms power savings. Backoff = 20%

	V_{DDL}= 0.6V				V_{DDL}= 0.8V			
Benchmark	**CVS**	**ECVS**	**GECVS**		**CVS**	**ECVS**	**GECVS**	
				ALC				ALC
c880	24.1	28.4	35.2	9.8	20.8	24.2	27.3	8.0
c190	7.3	8.6	13.8	7.2	6.9	7.0	12.1	5.6
c2670	20.6	28.9	32.1	9.2	16.3	23.3	26.3	7.9
c3540	4.2	23.0	30.7	9.3	3.3	18.9	24.6	5.4
c5315	27.4	35.9	39.3	10.8	22.1	28.0	31.8	8.2
c7552	33.9	39.8	44.2	12.3	26.4	31.2	35.4	9.6
ALU64	56.2	62.9	65.7	12.7	46.2	49.5	51.5	9.1
HUFFMAN	18.7	19.9	30.2	14.8	14.9	17.6	25.2	12.5
Average	24.0	30.9	36.4	10.8	19.6	25.0	29.3	8.3

All numbers are percentage reductions relative to initial all V_{DDH} design, except ALC column.
Source: (KULKARNI; SRIVASTAVA; SYLVESTER, 2004)

3.3.1 GECVS Optimization

Chinnery and Keutzer (2007) presented an improved version of the GECVS algorithm called GVS. It includes gate sizing, dual V_T optimization, and two major heuristic modules for optimizations. The first module seeks to increase the V_{DDL} assignment in the circuit (referred

Table 3.3 – Comparison of CVS, ECVS, and GECVS algorithms power savings. Backoff = 10%

	V_{DDL}= 0.6V				V_{DDL}= 0.8V			
Benchmark	**CVS**	**ECVS**	**GECVS**		**CVS**	**ECVS**	**GECVS**	
				ALC				ALC
c880	21.1	26.3	30.6	9.2	17.2	20.3	24.8	6.5
c190	5.8	6.9	10.7	5.2	4.6	5.4	8.8	6.0
c2670	16.7	24.4	26.9	5.7	13.1	18.1	23.5	7.9
c3540	3.5	14.0	24.3	8.3	2.9	12.3	20.8	7.1
c5315	22.7	31.8	35.5	10.3	18.9	25.3	28.4	8.2
c7552	30.4	35.5	39.4	10.5	24.1	27.5	31.0	9.0
ALU64	54.6	61.2	63.6	13.2	45.0	48.1	49.9	8.8
HUFFMAN	11.9	16.3	25.6	13.5	10.6	15.5	21.0	11.9
Average	20.8	27.1	32.1	9.5	17.0	21.6	26.0	8.2

All numbers are percentage reductions relative to initial all V_{DDH} design, except ALC column.
Source: (KULKARNI; SRIVASTAVA; SYLVESTER, 2004)

to as 'Assign-V_{DDL}' and shown in (4) by employing the gate sizing technique to create slack. The second module tries to increase the *high*-V_T assignment (referred to as 'Assign-*high*-V_T').

At the end of GECVS, any slack remaining in the circuit is not sufficient to support additional V_{DDL} assignments for power reductions; any further V_{DDL} assignments will either cause the circuit to fail timing or increase power consumption (CHINNERY; KEUTZER, 2007). To optimize this, the 'Assign-V_{DDL}' heuristic attempts to increase the number of V_{DDL} assignments by upsizing specific gates in order to create slack.

The 'Assign-V_{DDL}' step considers as candidates for V_{DDL} assignment only those V_{DDH} gates that do not drive any other V_{DDH} gates. This condition is necessary to avoid the insertion of ALCs. After identifying the candidates for V_{DDL} assignment, GVS evaluates the sensitivities as the GECVS algorithm does, by using Equation 3.1. Once the sensitivities of all candidates have been evaluated, the gate with the maximum sensitivity is assigned to V_{DDL}. When the gate with the best sensitivity has been assigned to V_{DDL}, the circuit no longer meets timing and then the algorithm resizes gates on critical paths to meet timing. Once sensitivities for all the gates are evaluated, the gate with the maximum sensitivity is selected and sized up.

Although assigning a gate to *high*-V_T will undoubtedly slow it down, the gate input pin capacitances also reduce somewhat (SIRICHOTIYAKUL et al., 1999), speeding up gates that fan into it. When the 'Assign-V_{DDL}' step concludes, some slack may remain in the circuit. Then,

```
Assign-V_DDL() {
L = Candidate gates, V_DDH gates not fanning out to V_DDH gates ;
while (list L is non-empty) do
    Calculate sensitivity of gates in L to changing to V_DDL ;
    STEP: "SET V_DDL" {
    Select candidate A with maximum sensitivity from L ;
    Remove A from L ;
    Set the supply voltage of A to V_DDL ;
    if (A drives a primary output) then
        Insert an LCFF ;
    end
    }
    STEP: "UPSIZING" {
    while circuit fails timing and number of upsizing moves is < 10% of total
    number of gates in the circuit do
        Calculate sensitivity of all gates to upsizing with Equation 3.1 ;
        Upsize gate with maximum sensitivity to the next higher size available in
        the library ;
    end
    Check timing ;
    Check power consumption ;
    if circuit meets timing and power increase < hill-climbing tolerance then
        STEP: "CLUSTERING" {
        Add to L gates that fan into A but not into any V_DDH gate ;
        if power < minimum power found then
            minimum power found = power ;
            Best configuration = current V_DDL, V_T & sizing assignment ;
        end
        }
    else
        Undo upsizing moves ;
        Remove any added LCFFs ;
        Set the supply voltage of A back to V_DDH ;
    end
    }
end
}
```

Algorithm 4: Pseudo-code for the Assign-V_{DDL} algorithm (CHINNERY; KEUTZER, 2007)

the 'Assign-*high*-V_T' heuristic attempts to convert this slack into power savings by converting gates from *low*-V_T to *high*-V_T. The algorithm employs the sensitivity measure on expression to identify the gates that will be upsized or will be set to V_{DDH}.

$$Sensitivity(upsizing/set - V_{DDH}) = \frac{\Delta Delay}{\Delta Power} \tag{3.2}$$

This sensitivity enables the algorithm to choose the gate giving the best delay improvement per unit power penalty. Once all sensitivities have been computed, the gate with the largest sensitivity is set to V_{DDH} or sized accordingly, creating slack.

3.4 Multiple Dynamic Supply Voltage

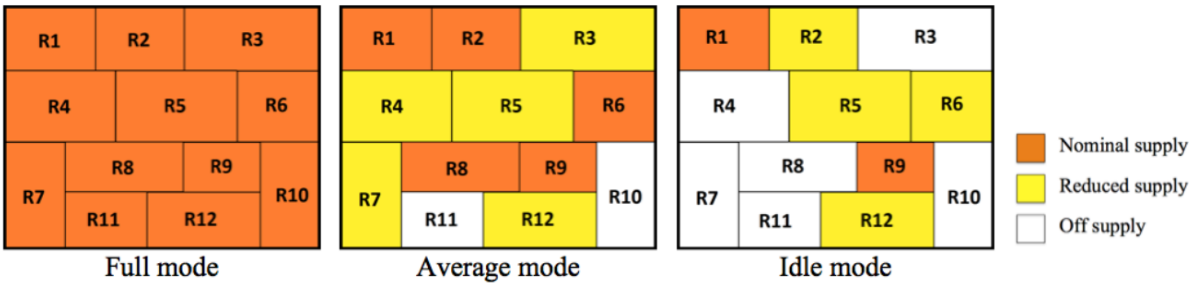
CVS-like techniques are considered Static Voltage Scaling (SVS) approaches (see section 2.2). The final clusters receive fixed supply voltages and the clustered cells can only operate at one of the available voltage levels (V_{DDH} or V_{DDL}). Therefore, the cells will always work at their highest V_{DD} level even when they are not necessary for the system operation. This will increase the switching activity of some cells and the charging, and discharging of load capacitances as well. Consequently, the power consumption increases, as can be seen on equation 3.3, where α is the switching activity factor, f the clock frequency and $(V_{DD} - V_{SS})$ the voltage across the capacitor when it is charged.

$$P = \frac{1}{2}\alpha f C(V_{DD} - V_{SS})^2 \tag{3.3}$$

To avoid the energy dissipation produced by the undesired switching activity, Multiple Dynamic Supply Voltage (MDSV) introduces the possibility to change the supply voltage dynamically in the same ways as Dynamic Voltage and Frequency Scaling (DVFS) does. But with MDSV, the circuit's cells are clustered in power domains, each one assigned to different voltage supplies.

Typically, MDSV employs the three supply levels explained before (V_{DDH};V_{DDL} and V_{off}). Depending on the operation requirements, the supply voltages of the system change *on-the-fly* to accomplish performance or energy-saving profiles. Moreover, MDSV can turn off some unused clusters to further power savings. Figure 3.4 illustrates a hypothetical example of a circuit using MDSV. It has three modes of operation and twelve power domains for which the supply voltages were assigned randomly. If the circuit requires maximum performance, the *Full mode* is selected and all the cells will be driven by the nominal supply voltage. An

Figure 3.4 – MDSV operation example



Source: (TERRES et al., 2013)

Average mode represents a situation of low-power where some cells will operate at reduced supply voltage, and few will be turned off. The third operation mode is *Idle mode*, in this standby situation most of the cells are deactivated, achieving the greatest power saving.

The MDSV technique can achieve outstanding results in terms of low-power, but it adds more complexity to the routing and placement phases of the IC designs (CHINNERY; KEUTZER, 2007). Moreover, must be considered the insertion of complementary circuitry like voltage scaling interfaces, buffers, and transistors for power gating; which increase area and overall power consumption.

3.4.1 Challenges in Multi-Voltage Designs

Even the most basic multi-voltage design presents the designer with some challenges. For instance, the power planning requires more careful and detailed floor-planning to support multiple power domains, turning the power grid a more elaborated structure. And, besides, designers must consider the power gating approaches for switching *off* and *on* the system, as well as sequential blocks to avoid data loss.

With a sole supply voltage for the entire chip, static timing analysis (STA) can be done at a single performance point. Typically, the EDA tools characterize the libraries for this point and then perform the timing analysis in a straightforward manner. With multiple blocks running at different voltages, timing analysis becomes much more complex increasing the execution time because of the corner cases that must be considered.

The main system level issue is that of power sequencing. In most instances, it will not be practical to bring up all the different power supplies at precisely the same time. Thus, it may be useful to plan an explicit power sequence, so that the different power domains come up in a well-defined order that assures correct function (KEATING et al., 2007).

Most of the complexity of using multiple voltages shows up on the boundaries of the power domains. Often, the signals that go between blocks of different power domains require additional circuits for voltage scaling. Besides, standard cell libraries are characterized to operate with a clean, fast input that goes rail to rail. Failure to meet this requirement may result in signals exhibiting significant *rise*- or *fall*-time degradation leading to timing closure problems (KEATING et al., 2007). Hence, we must ensure that each domain gets the voltage swings that it expects. We do this by providing level shifters between any domains that use different voltages.

4 VOLTAGE SCALING INTERFACES - LEVEL SHIFTERS

When using Multi-Voltage Designs, where supply voltages may change during operation, we have to guarantee the correct signal transmission and avoid excessive static power consumption between different power domains. Hence, the use of level converters is essential. The insertion of these interfaces impacts the projection and synthesis of the design. As well as the back-end phases like *Power Planning* and *Placement*. Moreover, the designer must architect and partition the design such that voltage domains have a defined relation to neighboring domains (KEATING et al., 2007).

Like most logic gates, the level shifters must be robust and insensitive to noise disturbances. It means that the noise margins of the LS should be larger as possible. These margins represent the level of noise that a gate can sustain (RABAEY; CHANDRAKASAN; NIKOLIC, 2003). In equation 4.1, *noise margin low* (NM_L) and *noise margin high* (NM_H) represent the size of the legal "0", and "1" intervals of a CMOS inverter respectively.

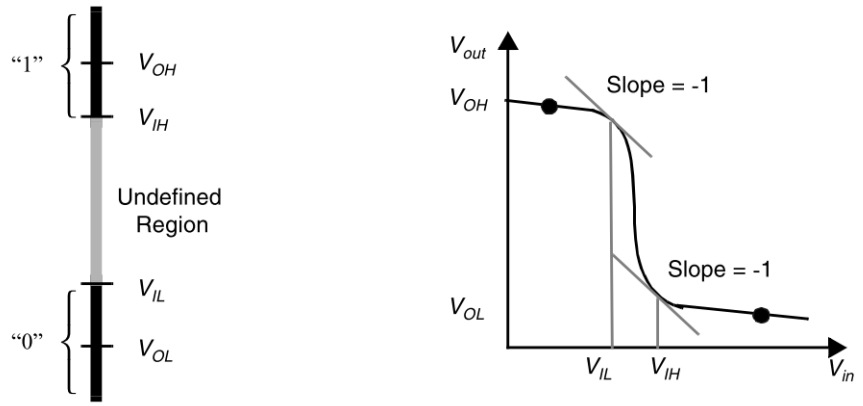
$$\begin{aligned} NM_L &= V_{IL} - V_{OL} \\ NM_H &= V_{OH} - V_{IH} \end{aligned} \tag{4.1}$$

The acceptable high and low voltages are delimited by the V_{IH} and V_{IL} voltage regions. These represent the points where the gain (dV_{out}/dV_{in}) of the *voltage-transfer characteristic* (VTC) equals -1, as Figure 4.1b shows. Where V_{OH} and V_{OL} are the high and low output voltage levels. The region between V_{IH} and V_{IL} is the *transition width* (RABAEY; CHANDRAKASAN; NIKOLIC, 2003). For proper circuit operation, this region should be avoided. Therefore, to interface two power domains successfully, certain requirements must be met:

- The V_{OH} of the driver cell must be greater than the V_{IH} of the receiver power domain.
- The V_{OL} of the driver must be less than the V_{IL} of the receiver.
- The output voltage from the driver must not exceed the I/O voltage tolerance of the receiver.

Level shifters do not affect the functionality of the design; from a logical perspective they are just buffers. For this reason, modern implementation tools can automatically insert level shifters where needed. Many EDA tools allow the designer to specify a level shifter placement strategy; to place the level shifter in the lower power domain, in the higher domain, or between them. Besides, the designer can specify which blocks lack of voltage level conversion, or establish a minimum voltage difference that requires level shifter insertion.

Figure 4.1 – Mapping logic levels to the voltage domain



(a) Relationship between voltage and logic levels (b) Definition of V_{IH} and V_{IL}

Source: (RABAEY; CHANDRAKASAN; NIKOLIC, 2003)

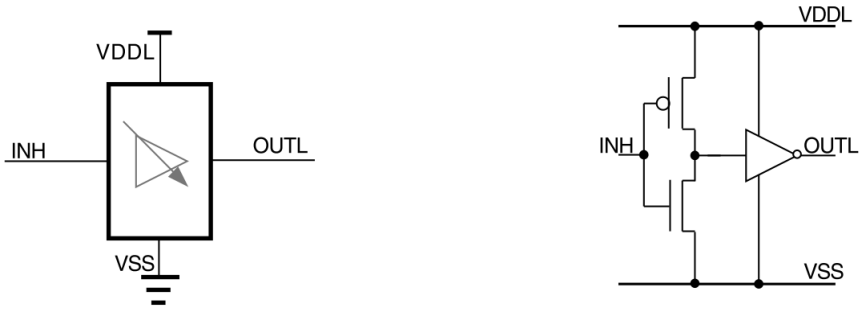
In multi-voltages designs, the distribution of the power domains create two contrasting situations that demand level shifters. One in which the signals travel from a domain with high voltage levels to a low power domain (*high-to-low*). And a second case in which the communication is between cells that belong to V_{DDL} region and logic blocks in a higher power domain (*low-to-high*).

4.1 High-to-Low Level Shifter

When a portion of the circuit, that operates at high frequency, communicates with other slower cells (i.e., low clock frequency as consequence of V_{DDL}), we require a *High-to-Low* Level Shifter between the two power domains. The cells that belong to a V_{DDH} domain have faster logic *high* or *low* switching levels compared to the V_{DDL} cells. Therefore, specialized high-to-low level shifters are provided for timing closure. If not, the entire library would have to be recharacterized to allow accurate static timing analysis.

High-to-low level shifters can be quite simple; in essence, a buffer with low supply voltage meets the requirements. Figure 4.2 shows an example of this circuit formed by two inverters in series. They are connected to a single power rail, which is the one from the lower or destination power domain.

Figure 4.2 – High-to-Low Level Shifter



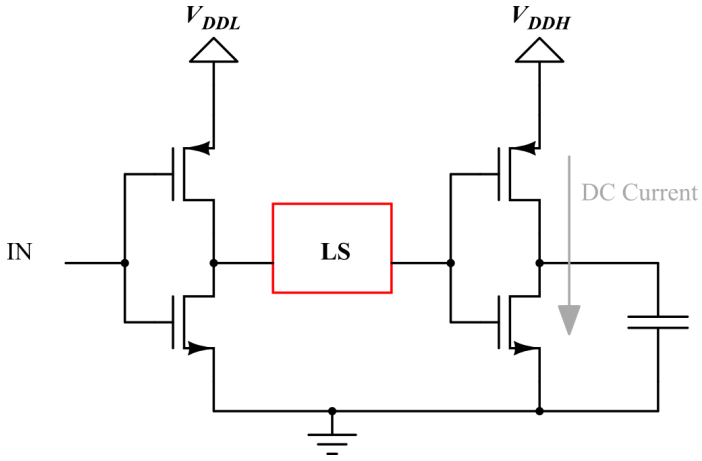
Source: (KEATING et al., 2007)

4.2 Low-to-High Level Shifter

The most common level shifter is the one that drives logic signals from a low supply rail to a cell on a higher voltage domain. *Low-to-High* level conversion is a more critical problem because under-driven signal degrades the rise and fall times at the receiving inputs. This can lead to higher switching currents and reduced noise margins. A slow transition time means that the signal spends more time near the threshold voltage, causing the short circuit current to last longer than necessary.

In the ECVS algorithm, a cell from the V_{DDL} list can feed a V_{DDH} driven cell if the signal goes through a level conversion. Figure 4.3 depicts an example of this situation. Where the level shifter (LS) provides fast, full-rail signals to the higher voltage domain. Without the use of a level shifter, the resulting DC current will produce high static power dissipation.

Figure 4.3 – Low-to-High Level Shifter



Source: Author

Power and timing overheads regulate the insertion of LS, because, low-to-high level shifters introduce a significant delay compared to the simple buffer delays of high-to-low level shifters. The time it takes for the input signal to reach the V_{DDH} level, increases the total delay of a logic path with voltage conversion. It complicates the setup and hold timing verification across such path. When the voltage difference between two power domains is large enough, the transistor in the input stage of a standard gate at the higher domain may not turn all the way off, leading to excessive short circuit current. In order to provide a reasonable noise margin, (KEATING et al., 2007) propose that the difference between the threshold voltage of the PMOS and the 10% of the V_{DDH} rail, must be larger than the difference between domains. If the condition in 4.2 occurs, where V_{TPMOS} is the threshold voltage of the PMOS transistor, then a low-to-high LS should be inserted to shut off the receiving PMOS input transistor stage.

$$V_{DDH} - V_{DDL} > V_{TPMOS} - (0.1 \times V_{DDH}) \tag{4.2}$$

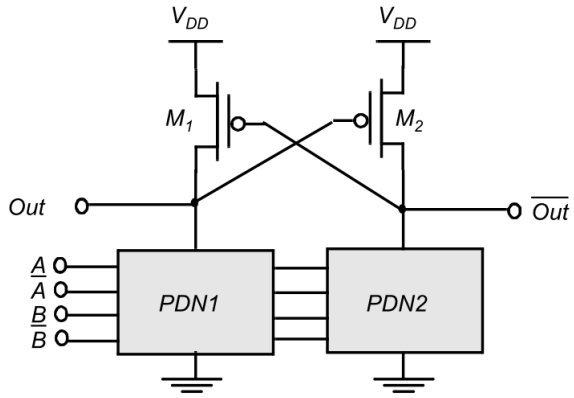
There are several design techniques for low-to-high level converters (USAMI et al., 1998; HAMADA et al., 1998; TAN; SUN, 2002; STOK et al., 2007; CHAVAN; MACDONALD, 2008; ROCHA et al., 2008). The approaches we found in the literature have particular characteristics and objectives. Some of them tackle the signal delay problem by providing fast-response level conversion (KULKARNI; SYLVESTER, 2003; TRAN; KAWAGUCHI; SAKURAI, 2005; KOO et al., 2005). Other topologies contribute to the reduction of power and area (STOK et al., 2007; BO; LIPING; XINGJUN, 2007). While some level shifters are capable of work in the sub-threshold region (ASHOUEI et al., 2010). Depending on the application needs, one approach might be more suitable than the other. With the right topology, we can control the intrinsic level shifter costs in area, energy dissipation, and delay.

We categorize the best-known level shifters into two big groups. Based on how many power rails (V_{DDH} or V_{DDL}) they need to operate, we have *Dual-rail* and *Single-rail* level shifters.

4.3 Dual-rail Level Shifters

The Dual-rail level shifters (DLS) connect to both power supplies, V_{DDH} and V_{DDL}. The conventional dual-rail LS are based on the *Differential Cascode Voltage Switch Logic* (or DCVSL) presented in (HELLER et al., 1984). Figure 4.4 shows a conceptual example of a DCVSL gate; where pull-down networks PDN1 and PDN2 use NMOS transistors, and are

Figure 4.4 – DCVSL gate



Source: (RABAEY; CHANDRAKASAN; NIKOLIC, 2003)

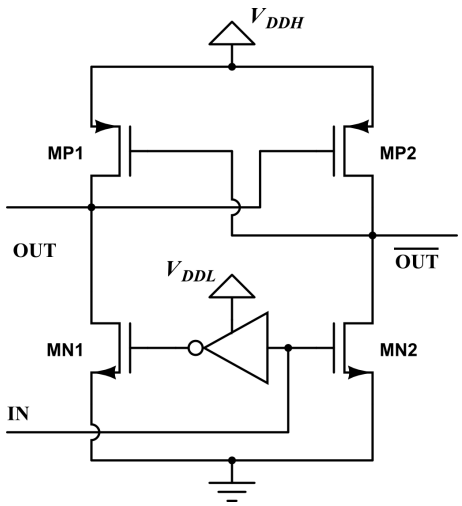
mutually exclusive. It means that when PDN1 conducts, PDN2 is off and vice-versa. If initially, the outputs Out and $\overline{Out}$ are high and low respectively; turning on PDN1 causes Out to be pulled down. Meanwhile, $\overline{Out}$ is in a high impedance state, and M_2 and PDN2 are both turned off. Therefore, PDN1 must be strong enough to bring Out below $V_{DD} - |V_{TPMOS}|$, the point at which M_2 turns on and starts charging the $\overline{Out}$ load capacitance. This in turn enables Out to fully discharge.

4.3.1 DCVS level shifter

The most traditional level shifter topology uses the cross-coupled PMOS transistors of the DCVSL to pull output to the high voltage. Hence, it is referred as *DCVS level shifter*. In Figure 4.5, the inverter connected to a low power domain (V_{DDL}) switches the input signal. When the input is *low*, the MN1 and MP2 transistors activate and raise the voltage at node OUT to V_{DDH}. Subsequently, if IN asserts, MN2 and MP1 activate; thereby lowering the output voltage to V_{DDL}. The pull-down transistors (MN1 and MN2) has to overcome the PMOS latch action of the MP1 and MP2 transistors driven with a higher supply voltage (V_{DDH}). Thus the NMOS transistors have to be larger than the PMOS transistors.

This cross-coupled level shifter has the advantage of a simple design, well suited for higher core voltages (ROCHA et al., 2008). However, the DCVS topology exhibits high short circuit during transitions because the PMOS gates experience full voltage swing from 0V to V_{DDH}. Even when the input transitions are fast.

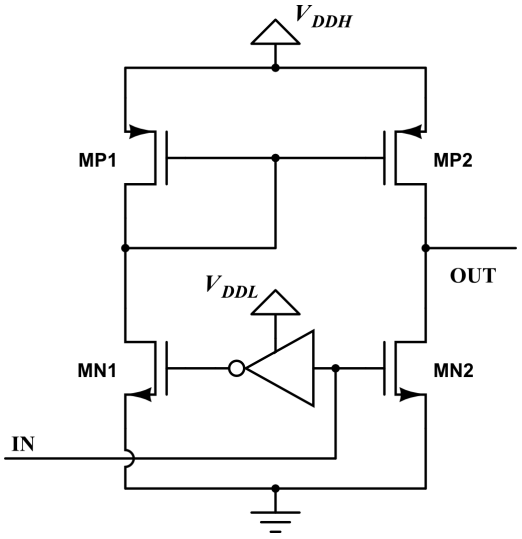
Figure 4.5 – DCVS level shifter



Source: Author. Original topology in (HELLER et al., 1984)

Kobayashi (1993) presented a modified version of the cross-couple level shifter. The circuit, shown in Figure 4.6, provides stable current driving capabilities compared with the DCVS. It replaces the PMOS half-latch with a current mirror, that is well suited for wide voltage range conversion with regards to performance. However, the *constant-current mirror* level converter suffers from increased power consumption resulting from the leakage path formed either by MP1 or MP2 in the current mirror and one of the pull down devices (CHAVAN; MACDONALD, 2008).

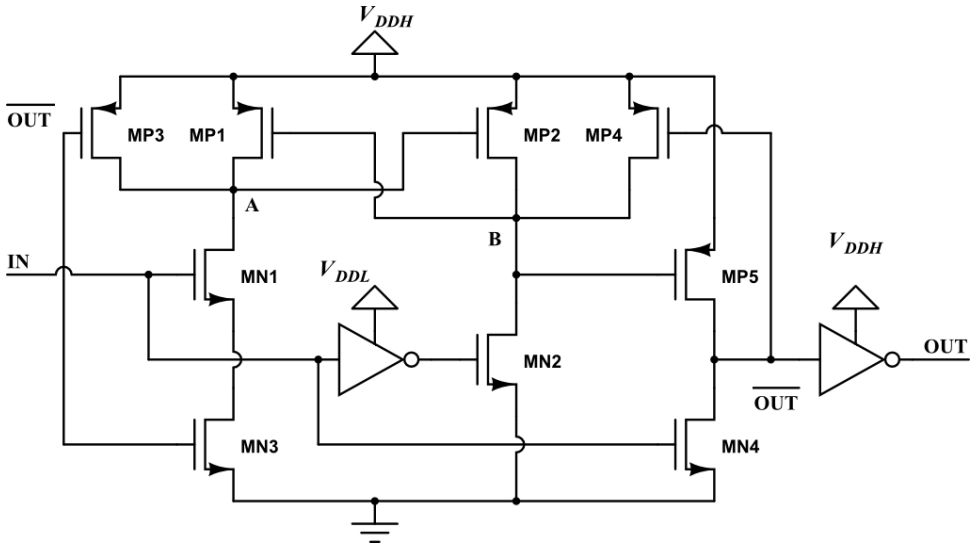
Figure 4.6 – Constant-current mirror level shifter



Source: Author. Original topology in (KOBAYASHI et al., 1993)

In (ASHOUEI et al., 2010) is presented a DCVS-based level shifter that can operate

Figure 4.7 – Near-threshold level shifter



Source: Author. Original topology in (ASHOUEI et al., 2010)

with substantially different voltage levels. The topology in Figure 4.7 supports voltages from near-threshold to full supply voltage. The level shifter was designed in a 90nm CMOS technology and uses thick-oxide transistors, along with non-minimum channel length transistors to up convert from 0.36V to 1.32V. It modifies the DCVS level shifter by adding two PMOS devices (MP3 and MP4) and a NMOS device (MN3). Also, it uses a feedback loop from the output to drive the newly added transistors and to provide *different non-conflict rise and fall paths* (ASHOUEI et al., 2010).

When IN is *low*, node $\overline{OUT}$ charges the node B towards V_{DDH} through MP4. This eventually turns MP1 on, but the "off" state of MN3 transistor avoids the discharge of node A, resulting in MP2 to be "off" as well. The static condition is different from the DCVS LS in having the MP2 transistor "off" while the node A is charged. In this moment, a '0' to '1' transition of IN involves MN2 to pull down node B competing with MP4. Since the function of the PMOS MP4 is to hold the value at the node B and has no duty at a transition, it is made weak and with high R_{on}. This implementation uses a thick oxide MP3 transistor which has high threshold voltage.

When IN is *high*, the output will be at logic '0' because MN4 pulls down node $\overline{OUT}$. Then, MP4 turns off and node B is kept at '0' by MN2. Thus, MP1 is "on" keeping node A at '1', which results in MP2 to be "off". Furthermore, MN1 disconnects the discharge path of node A. A high to low transition involves MP2 to pull up OUT, competing with the pull down path consisting of MN1 and MN3. A '1' to '0' transition involves MP2 to pull up node B, competing with the pull down path of node A (MN1 and MN3).

Among the four PMOS transistors, this level shifter employs just the MP2 to make a transition. The others PMOS are used only to hold the values at nodes A and B. Hence, MP2 can be stronger than the other transistors (MP1, MP3, and MP4 are slow). The circuit uses thick oxide IO devices for these three transistors, since thick oxide IO devices have higher threshold voltage than the standard devices.

Although the circuit of Figure 4.7 benefits from non-conflicting rise and fall transitions; it has the issue that in a '0' to '1' transition, MN2 should pull node B low enough for the output stage inverter to turn off MP4 (ASHOUEI et al., 2010). Whilst, in the classic DCVS level shifter, node B has to be pulled down only below the V_T of MP1. To address this issue, it requires an extra inverter stage that flips the input signal and also increase delay.

4.3.2 Bootstrapping level shifter

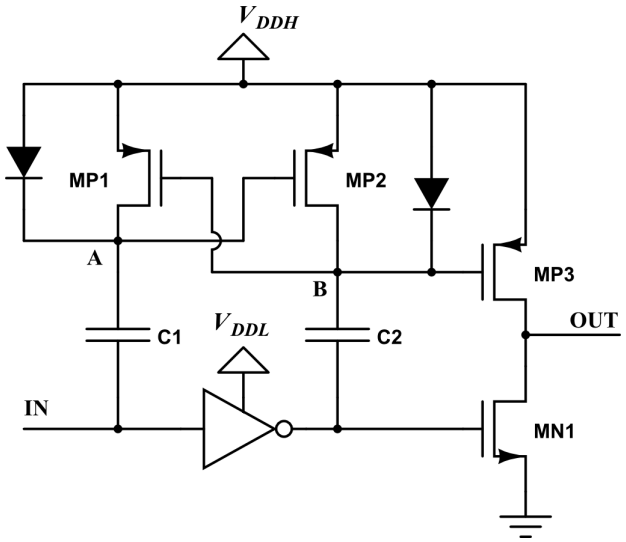
To reduce the dynamic power of the DCVS LS, produced by the excessive short-circuit current during voltage shifting, (TAN; SUN, 2002) introduced a level converter with boot-strapped gates. It uses capacitors to maintain the voltage difference between the gates of pull-up PMOS and pull-down NMOS. The power saving over conventional LS is achieved by the reduced voltage swing at specific nodes. In Figure 4.8, two boot capacitors C1 and C2 replace the pull-down NMOS transistors to maintain the voltage difference at the gate terminals of MP3 and MN1. An inverter provides the complementary signal to drive the level shifter. For simplicity, each diode represents two serially connected diodes.

The pull-down NMOS at the output stage switches between '0' and V_{DDL}; whereas, the voltage transitions of the pull-up PMOS are between $V_{DDH} - V_{DDL}$ and V_{DDH}. Initially, when the input is low, MP1 is off and MP2 activates; then C2 charges to V_{DDH} through MP2. The C1 capacitor is charged by diode-connected transistors to a voltage of $V_{DDH} - 2V_{diode}$. MN1 turns on and pulls the output to ground, while MP3 remains off. The change from '0' to V_{DDL} pushes node A to $(V_{DDL} + V_{DDH} - 2V_{diode})$, while the node B is pulled down to $(V_{DDH} - V_{DDL})$, switching the output to high. Subsequently, node A discharges to V_{DDH} through MP1, and node B discharges to $(V_{DDH} - 2V_{diode})$.

The gate voltage of MP3 transistor swings exclusively from V_{DDH} to $V_{DDH} - V_{DDL}$. Therefore, the pull-up transistor can switch off in a shorter time, minimizing the short-circuit current flow present in all previous level shifters. This topology has an intrinsic charge conservation. For instance, part of the C2 charge is transferred to the gate of the MP3 transistor. When the input of the circuit goes a high logic level, the charge on the gate of MP3 flows back to the

bootstrapping capacitor.

Figure 4.8 – Bootstrapping level shifter



Source: Author. Original topology in (TAN; SUN, 2002)

We can use the charge conservation of the level shifter to find out the capacitance of the bootstrapping capacitors, C1 and C2 (TAN; SUN, 2002). When the input switches from low to high, the node A voltage changes from $V_A = (V_{DDH} - 2V_{diode})$ to V_{DDH}, and the charge is transferred from C1 to the node A. Therefore, we can obtain an approximation of the C1 bootstrapping capacitor from equation 4.3. Where, C_A is the total capacitance at node A with respective to ground, excluding C1; and V_{diode} is the voltage drop across a single diode.

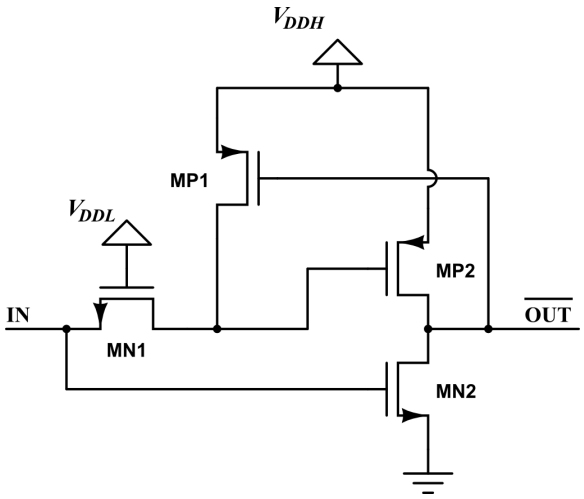
$$\begin{aligned} C1 &= C_A \frac{[V_{DDH} - V_A]}{[V_A - V_{DDH} + V_{DDL}]} \\ &= C_A \frac{2V_{diode}}{V_{DDL} - 2V_{diode}} \end{aligned} \tag{4.3}$$

The bootstrapping technique attains lower power at the expense of a significant increase in physical area due to the relatively large boot capacitors. To turn off MP3 completely, the level shifter must ensure sufficient charge transfer from C1 to C_A, C1 must be more than three times bigger than C_A. Moreover, when V_{DDL} is close enough to the voltage drop across a single diode, the boot capacitor C1 becomes considerably large. The size of the bootstrapping capacitors also defines the switching delay between low and high voltage states. If the capacitors are too small, the voltage swings at nodes A and B will be reduced owing to the charge redistribution.

4.3.3 Feedback-based level shifters

We found some topologies that use feedback paths to pull up the output or other internal stages. Hamada (1998) proposed the level shifter in Figure 4.9, that is based on a weak feedback pull-up gate MP1 and a NMOS pass gate (MN1). The purpose of the MN1 transistor is to isolate the input of the PMOS MP2 from the previous logic stage. Hence, the feedback transistor MP1 can pull-up the internal node without consequence to the prior logic that runs at V_{DDL}. This level shifter consumes less energy than the DVCS level shifter due to fewer devices and less switching activity, but it has higher static power consumption (KULKARNI; SYLVESTER, 2003).

Figure 4.9 – Pass gate level shifter

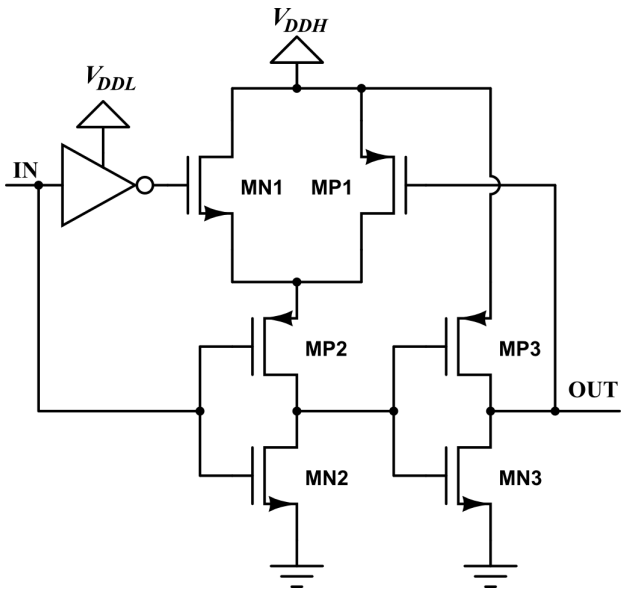


Source: Author. Original topology in (HAMADA et al., 1998)

Figure 4.10 shows the schematic diagram of a level shifter originally presented in (BO; LIPING; XINGJUN, 2007). It uses the positive and the negative states of the lower voltage input signal to simultaneously control the "on" and "off" state of the NMOS transistors (MN1 and MN2). The output of this converter is a half latch that pulls up the input of the inverter formed by MP3 and MN3. Compared to the DCVS level shifter, it presented a reduction of 36% in the leakage power dissipation, but it has an increase of 5% on delay (BO; LIPING; XINGJUN, 2007). This topology has more delay than the previously presented level shifters due to the extra inverter at the output stage.

The function of the level shifter in Figure 4.10 is as follows: when the input signal (IN) is high, the MN2 transistor turns on. Regardless of the state of MP2, there is no leakage power path between the V_{DDH} and the ground because the MN1 is "off". Meanwhile, for a low input,

Figure 4.10 – Half latch level shifter



Source: Author. Original topology in (BO; LIPING; XINGJUN, 2007)

Table 4.1 – Simulation results for the Half-latch level shifter under two operation conditions at 100MHz input

Parameter	First Condition			Second Condition		
	DCVS	**SLS**	**HL-LS**	**DCVS**	**SLS**	**HL-LS**
Leakage power (W)	259.648p	8.794u	158.133p	277.901p	238.667p	177.137p
Delay (ns)	0.242	0.1667	0.263	0.188	0.148	0.198
Leakage current (A)	77.316p	2.665u	46.552p	84.212p	72.232p	48.377p

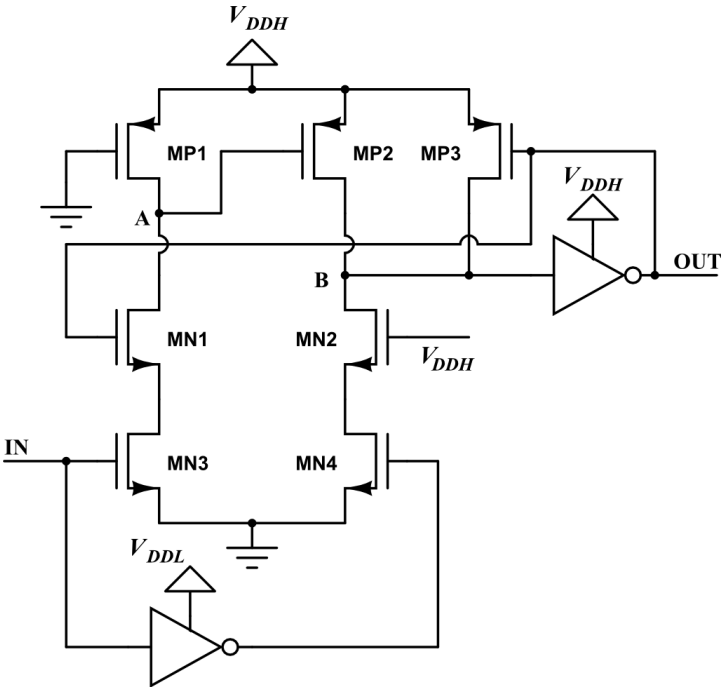
First condition: High-to-low transition (1.8V to 0V). Second Condition: Low-to-High transition (0V to 3.3V).

Source: (BO; LIPING; XINGJUN, 2007)

the MN1 and MP2 transistors are both “on”, and the MN2 is “off”. Although there is a threshold drop across the MN1, the feedback transistor MP1 can charge its node till V_{DDH}.

Table 4.1 shows the simulation results for this level shifter (HL-LS) compared with the DCVS and the single-rail level shifter (SLS) in (STOK et al., 2007). The circuit was designed in 180nm CMOS technology and simulated with SPICE on leakage power dissipation, delay and leakage current of V_{DDH}, under two conditions: 1) a 100MHz input logic signal swinging between 1.8V and 0V, with level shifted signals between 3.3V and 0V; 2) a 100MHz input logic signal swinging between 0V and 2.6V and the level shifted signals between 0V and 3.3V. As we can see from the results, this proposed circuit increases the delay time. Also, the three level shifters presented a relatively high leakage current.

Figure 4.11 – Fedback-based level shifter



Source: Author. Original topology in (GUPTA et al., 2008)

(GUPTA et al., 2008) proposed the converter shown in Figure 4.11. The circuit uses a feedback mechanism to shutoff the static current path that is created during low-to-high transitions. It shows more robust operation than the DCVS level shifter in terms of consumption, but it presents higher time delays under certain conditions (GUPTA et al., 2008). For fast transitions, the proposed level shifter draws less switching current than the DCVS.

This feedback based level shifter uses thin oxide, low voltage MOSFETs (MN3 and MN4) and high voltage transistors (MN1, MN2, MP1, MP2, and MP3). The thin oxide transistors operate at lower voltage level and a higher voltage (V_{DDH}) supplies the thick oxide transistors. The MN1 and MN2 transistors protect the thin oxide transistors MN3 and MN4 from high voltage level at their drains. The level shifter receives a low voltage level input and level shifts it to the high voltage level at output OUT.

For a steady state logic high at IN, the output node OUT is “low”. The nodes A and B are held at V_{DDH} level through the “on” PMOS transistors MP1 and MP3 respectively. When the input goes from logic high to low, transistor MN3 turns-off and MN4 turns on, pulling down the node B through the activated NMOS MN2. This in turn switches OUT to logic high, deactivating MP3 and turning-on MN1.

The MP3 transistor just holds the logic state and does not affect the signal transition, hence it is minimum sized. While pulling the node B down to ground, the cascade of MN2 and

MN4 easily overcomes its small size.

As the input transitions from logic zero to V_{DDH}, MN3 and MN4 turn on and off respectively. The activation of MN3 pulls down the node A through the already "on" NMOS MN1. This creates a current path (MP1→MN1→MN3) between V_{DDH} and ground. Then MP2 turns on, which pulls the node B to V_{DDH} and eventually switching the output OUT to low. The decreasing voltage at the output deactivates MN1, shutting off the current path and pulling the node A back to V_{DDH}. Finally, the MP2 transistor turns off and logic "high" at node B is held by MP3 again.

The needs of the DLS for two voltage supply limit the physical placement of such level shifters to the boundary of high and low voltage regions. It also restricts the physical design flexibility and the operation range of DCVS-based level converters. In the literature, several approaches require only one voltage rail to operate and to address almost every drawback of the dual-rail level shifters.

4.4 Single-rail Level Shifters

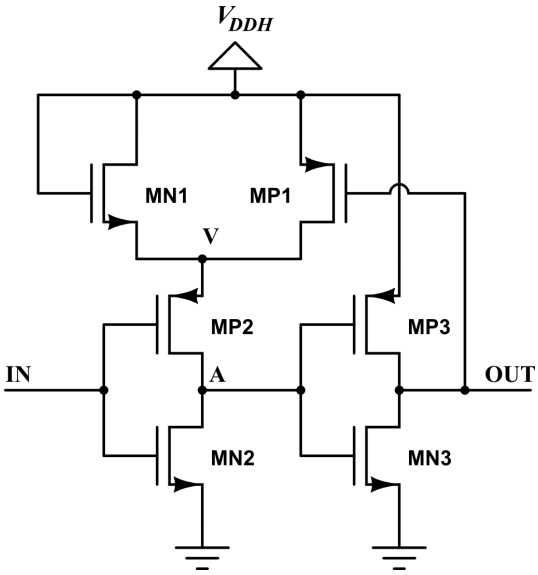
The advantage of single-rail level shifters (SLS) over dual-rail level shifters has been illustrated in (KHAN; WADHWA; MISRI, 2006). SLS afford fewer pin count; reduced congestion in supply routing; complexity; and overall system cost. These topologies bypass the access to the lower power supply voltage; instead, they just need a supply from the signal to be converted. The biggest advantage of these level shifters is their flexible placement which enables efficient physical design of voltage islands. As disadvantage, a single-rail level shifter can suffer from higher leakage currents if input signal supply level is lower (or V_{DDH} is higher) than the input supply level by more than V_T.

4.4.1 Puri's level shifter

The most common SLS have some feedback scheme to convert the input low voltage signal to the higher voltage. Also, the threshold drop across specific transistors generate a virtual lower V_{DD}. (STOK et al., 2007) proposed the level shifter in Figure 4.12. The threshold drop across the NMOS MN1 (V_{Tn}) provides a virtual V_{DDL} to the input inverter formed by MP2 and MN2. The output stage is a half latch that pulls up the input of the inverter (MP3, MN3) to V_{DDH} in order to avoid leakage.

When the input signal (IN) is *high*, the voltage at node V is $V_{DDH} - V_{Tn}$ with the purpose of reducing gate to source voltage of MP2, and hence, turn it off. For a low-to-high conversion, the feedback transistor MP1 turns on, so that charges node V to V_{DDH} to compensate the threshold drop of MN1. Therefore, the supply voltage of the inverter at the input stage dynamically switches between $V_{DDH} - V_{Tn}$ and V_{DDH} depending upon the input state.

Figure 4.12 – Puri's level shifter

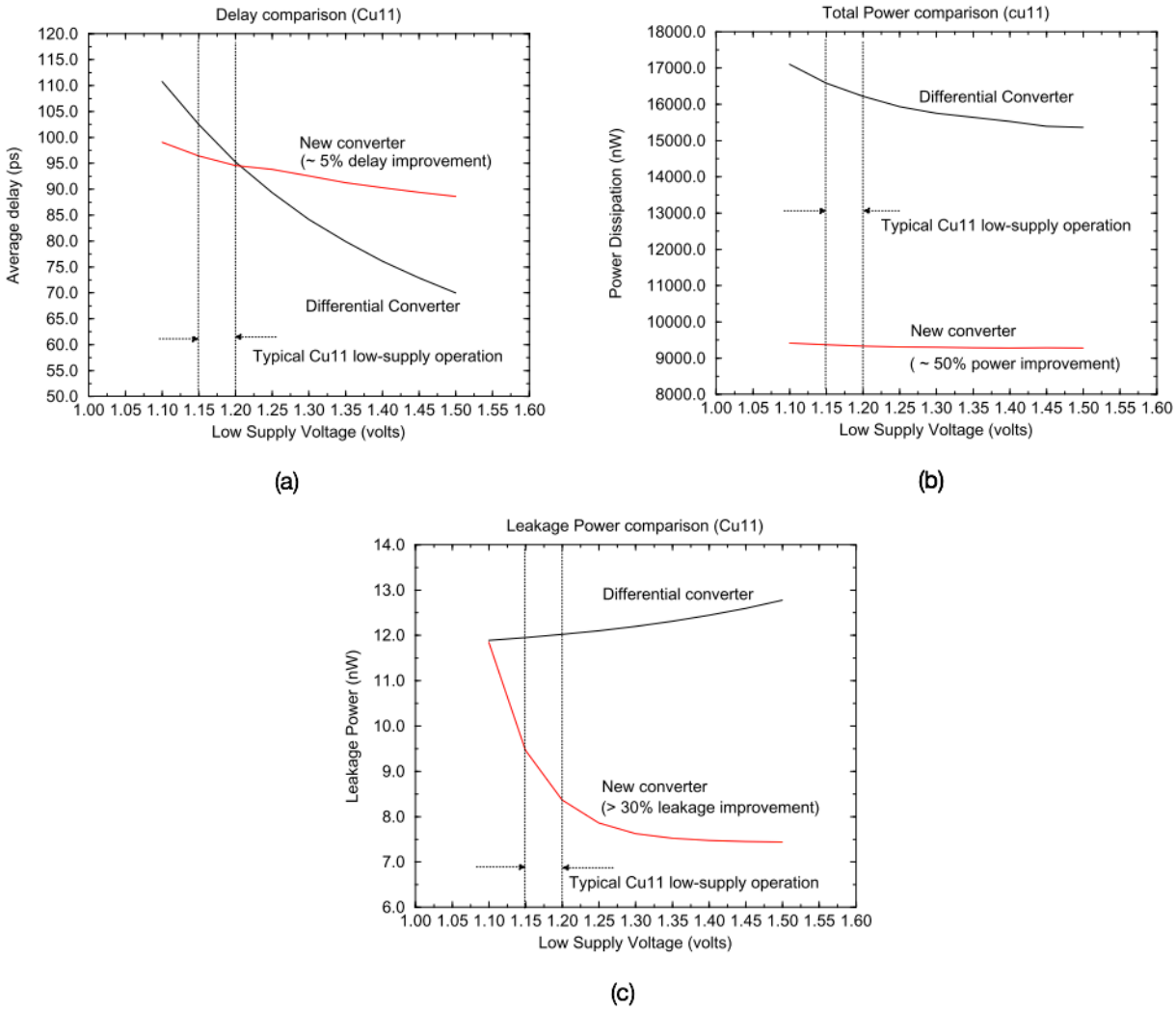


Source: Author. Original topology in (STOK et al., 2007)

This topology can suffer from higher leakage currents if the input signal level is lower (or V_{DDH} is higher) than the input supply level by more than V_{Tn}. This contribute to the power consumption on a chip with large number of level shifters. In order to maintain good CMOS performance characteristics, it is desirable to have the ratio of V_T/V_{DD} below 0.3 (TAUR, 2002). Thus typically, the low supply in sub-100nm designs will be limited to 25-30% below V_{DDH} (STOK et al., 2007). Moreover, the diode connected transistor NM1 limits the operation speed of the circuit.

Authors compared the circuit in Figure 4.12 to the DCVS level shifter, designed in 130nm Cu11 technology with nominal V_{DD} =1.5V. The results are reproduced in Figure 4.13. It shows that the Puri's level shifter (*New converter*) achieved up to 5% better delay (Figure 4.12a), consumes 50% less total power (Figure 4.12b), and approximately 30% less leakage power (Figure 4.12c).

Figure 4.13 – Comparison of DCVS converter and Puri's level shifter

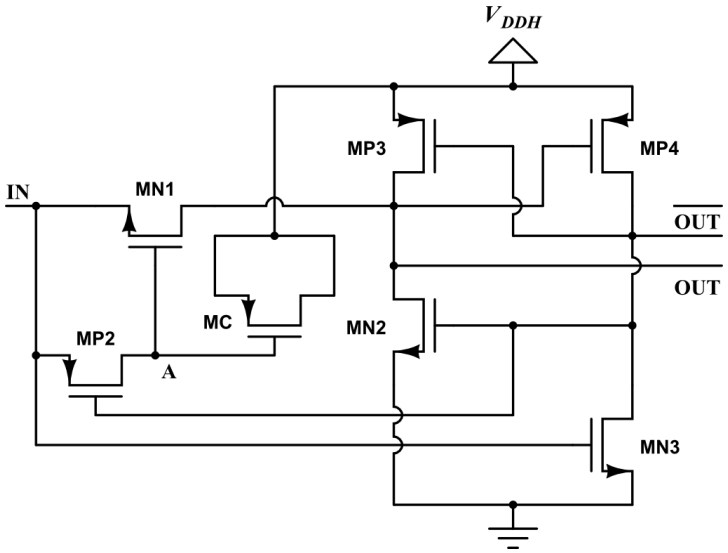


Source: (STOK et al., 2007)

4.4.2 Cross-coupled with single supply

(KHAN; WADHWA; MISRI, 2006) presented a level shifter based on the DCVS. It uses a CMOS capacitor (MOSCAP) to *pre-charge* the output value when the input is at logic zero. The range at which the level shifting can be done is limited only by the technology and not by the design. Figure 4.14 depicts the schematic diagram of this single supply level shifter. We call the input stage as the *pre-charge phase*, it contains a CMOS capacitor (MC) and a node (A), formed by MN1 and MP2 transistors, to control its charging and discharging. The pull up network of the output stage is formed by cross-coupled PMOS transistors (MP3 and MP4) as in the DCVS level shifter.

Figure 4.14 – Cross-coupled SLS



Source: Author. Original topology in (KHAN; WADHWA; MISRI, 2006)

The MC capacitor acts as a start-up when the circuit is powered up. Along with the MP2 transistor, it guarantees the correct value at the output OUT. When V_{DDH} supply is turning on and IN is *low*, there is a possibility that the node OUT starts following V_{DDH} and causes a wrong output. Thanks to MC, node A follows V_{DDH}, thus activating MN1. This provides a discharge path for OUT if it starts to pull up to V_{DDH}, and eventually pulls it down to ground.

When IN is *high*, MN3 turns on pulling down $\overline{OUT}$ to ground. This activates MP3, which pulls the output node OUT to V_{DDH}. With the complementary output at logic *low*, the MN2 and MP2 transistors will be "off" and "on" respectively. The current flow through MP2 charges node A until the voltage level of IN. MN1 will remain deactivated because the V_{DDH} level is higher than the IN level (V_{DDL}).

If a high-to-low transition occurs, initially node A will remain at V_{DDL}, turning on MN1 due to the voltage difference. To prevent the node A from getting discharged before the output OUT, MP2 is weak compared to MN1. The latter should be strong enough to charge node A at V_{DDL} level of IN within "on" period of the input signal. Hence, the size of both transistors depends upon the design frequencies and the voltage levels of the low and high power domains (KHAN; WADHWA; MISRI, 2006). As the node A starts to discharge through MN1, the MP4 transistor charges node $\overline{OUT}$ towards V_{DDH}. When node $\overline{OUT}$ charges above the threshold voltage of MN2, it turns on and begins to discharge node OUT, providing a positive feedback to discharge the node OUT quickly.

The value of the MC capacitance generates a sizing problem because MP2 should be

able to charge node A within the "on" period of the input signal IN. At lower frequency, if the input remains in *low* state for long time, node A might discharges completely through MP2 leakage path, turning it off. Despite node OUT being disconnected from IN, the output signal may not change because the previous state gets latched within the cross-coupled transistors and MN2.

The major problem of this level shifter is that the pre-charge phase conditions the power sequence of the design. The body of the MP2 transistor *must connect with V_{DDH} in case of power sequencing is defined in such a way that V_{DDH} ramps up before the input signal supply* (KHAN; WADHWA; MISRI, 2006). Otherwise, the body of MP2 should be connected to node A to prevent any leakage through body diode in case input is *high* while supply voltage is off. This increase the complexity of the design as well as the routing of the voltage scaling interfaces.

4.5 DLS versus SLS

As Table 4.2 shows, the advantage of SLS over DLS goes from the design to the implementation. In some cases, single-rail level shifters outperform the dual-rail level shifters. It obviously depends on the topology, circuit optimizations, and final application. For instance, in a complex multi-voltage system, the placement and routing of single-rail level shifters could be much easier than if we use dual-rail level shifters.

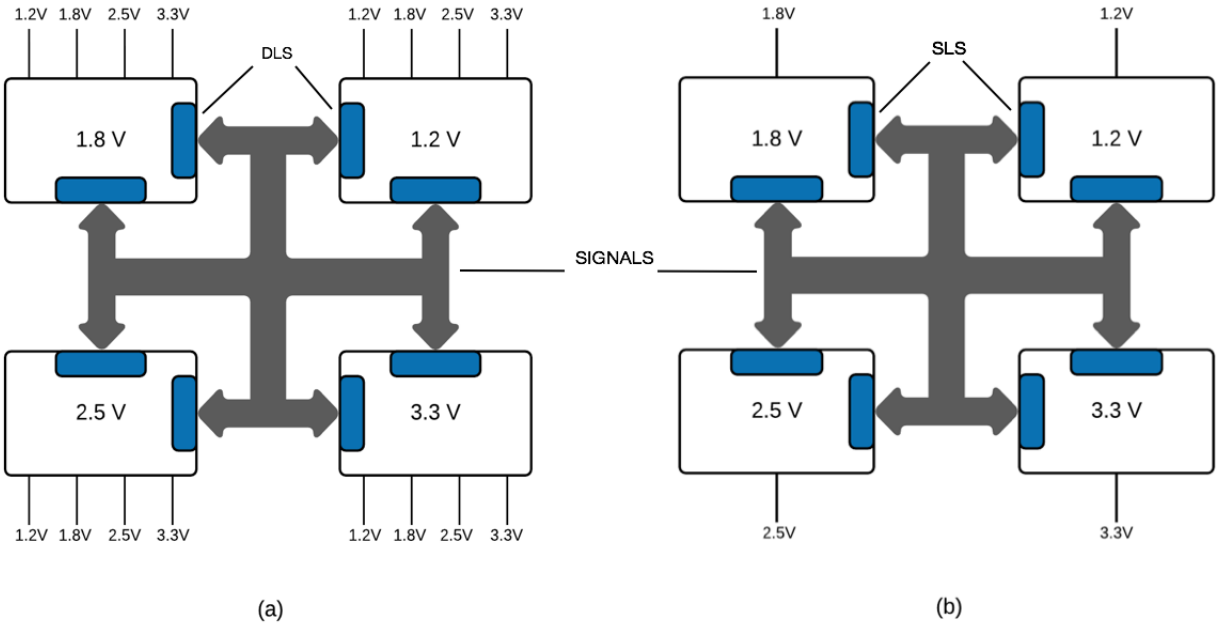
Table 4.2 – Impact of DLS and SLS in the design of a multi-voltage system.

Design Phase	**Single-rail LS (SLS)**	**Dual-rail LS (DLS)**
Placement	*low*	*medium*
Routing	*medium*	*high*
Power planning	*low*	*medium*

Figure 4.15 better illustrates the previous situation for a circuit with four power domains that communicate between them. Each rectangle represents modules at equal voltage level. Figure 4.15a depicts the block diagram of a multi-voltage system using conventional level shifters with duple supply voltages. The pin count of the interfaces increases significantly because each level shifter must connect to the respective power rails. In a multi-voltage SoC, there might be thousands of level shifters for transferring the signals between modules. This leads to congestion in the routing of supplies and muddles the placement of the voltage scaling interfaces.

The single-rail level shifter overcomes this problem by using only one supply, hence reducing the routing congestion and relaxing the placement constraints. Figure 4.15b shows the system of the previous example, but this time with SLS. By using single-rail level shifters to interface the modules, we require fewer pins to set up the power grid. In fact, SLS allows a correct communication between modules without adding any extra supply.

Figure 4.15 – DLS versus SLS on a multi-voltage design



Source: Author

Having in mind these advantages and the few options we found in the literature for a level shifter that uses only one power rail, we propose a modified topology and compare it to level converters we found in the literature.

5 ENERGY-EFFICIENT LEVEL SHIFTER

Multi-Voltage approaches take advantage of the existence of different blocks inside the IC, with particular target performance, objectives, and constraints. Basically, these techniques partition the internal logic into multiple voltage regions or power domains compounded of logic blocks with same supplies. The energy demand of those blocks or cells varies under certain conditions and, sometimes, some of them do not operate at all when the system is in standby mode, enabling greater power savings than the low level optimization strategies.

The CVS algorithm (USAMI; HOROWITZ, 1995) achieves an economy of 10-20% in power (with reference of initial design; all cells at V_{DDH}) by assigning lower supply voltage to the cells that are off the critical paths and do not drive cells supplied with nominal voltage (V_{DDH}). Hence, the signal level conversion functionality is restricted to the flip-flops that delimit a logic cell, the *Level Converting Flip-Flop* (LCFF). The ECVS algorithm (USAMI et al., 1998) bypasses the topological constraint of the CVS and allows V_{DDL} driven cell to be part of the fan-in of a V_{DDH} cell. It is accomplished with the aid of *Asynchronous Level Converters* (ALC), here referred to as *Level Shifters* (LS). ECVS leads to 20-30% higher energy savings because bigger portions of the circuit can be clustered into low voltage regions. The GECVS algorithm (KULKARNI; SRIVASTAVA; SYLVESTER, 2004) improves the performance and power savings of the CVS by reducing the group of V_{DDL} cells driving V_{DDH}. It creates bigger V_{DDL} cluster cells and removes LS in particular non-critical logical paths with enough slack. GECVS achieves 25-35-% power savings with reference of the initial design.

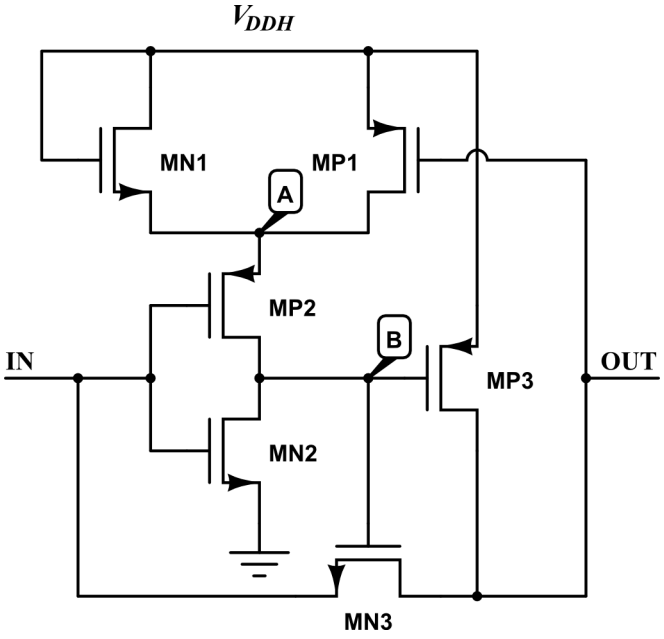
Projecting a system with different supplies adds some complexity to the design. For instance, the *Power Planning* phase requires a more careful and detailed floor-planning to support multiple power domains, turning the power grid a more elaborated structure. Moreover, most of the complexity of using multiple voltages shows up on the boundaries of the power domains. Often, the signals that go between blocks of different power domains demand additional voltage scaling interfaces. These level shifters can be classified into two groups: one containing all the level converters that involve a single power rail to operate (SLS) and the other group of dual-rails level shifters (DLS). In some complex multi-voltage systems, SLS are a better option for voltage scaling interfaces. Mainly, because the placement and routing of single-rail level shifters could be much easier than the DLS.

(STOK et al., 2007) accomplished a low-to-high voltage level conversion with only one power rail by creating a virtual V_{DDL}. In Figure 4.12, the threshold drop across the NMOS transistor (MN1) generates a lower voltage that is used to supply the input stage of the level

shifter. The output stage of the circuit regenerates the signal, raising its voltage level to V_{DDH}. We propose a simple modification to this level shifter and compare the resulting circuit with the traditional DCVS and the Puri level shifters. The resulting topology reduces power consumption, has a low cost (small number of transistors and low area), and works efficiently converting a wide range of voltage levels. It also achieves good results in terms of signal delay and Power-Delay Product (PDP).

5.1 Formal overview and characterization

Figure 5.1 – Proposed level shifter *EF-LS*



Source: Author

The circuit in Figure 5.1 (level shifter *EF-LS*) replaces the inverter in the output stage of the Puri's converter (STOK et al., 2007) by a NMOS pass transistor, which is used as a switch to pass logic levels from the input to the output. With this modification we try to reduce the delay of the topology and the leakage current for low input signals that the original level shifter presents. The Puri's topology can suffer from higher leakage currents if the input signal level is lower (or V_{DDH} is higher) than the input supply level by more than V_{Tn}, contributing to excessive power consumption. The proposed level shifter can speed up the switching time for input levels under 50% of the nominal V_{DDL} supply voltage. It achieves lower mean power-delay product and, hence, greater energy savings.

The voltage of node A becomes a virtual lower V_{DD} that supplies the inverter of the

input stage. This value can be calculated as: $V_A = V_{DDH} - V_N$; where V_N is the threshold drop of the MN1 transistor. Node B controls the gate of the pass transistor MN3 that receives the input signal. The inversion of IN guarantees the correct logic level at the output and the control voltage in node B.

If IN is at logic *high*, node B will be at low voltage level, turning on the MP3 transistor. This creates a path to charge OUT at V_{DDH} level. In the other case, when the input is *low*, the high voltage level at node B activates the NMOS MN3 transistor passing the logic *low* to the output. The feedback transistor turns on the charging node A to V_{DDH} to compensate the threshold drop of MN1.

5.1.1 Simulations and Results

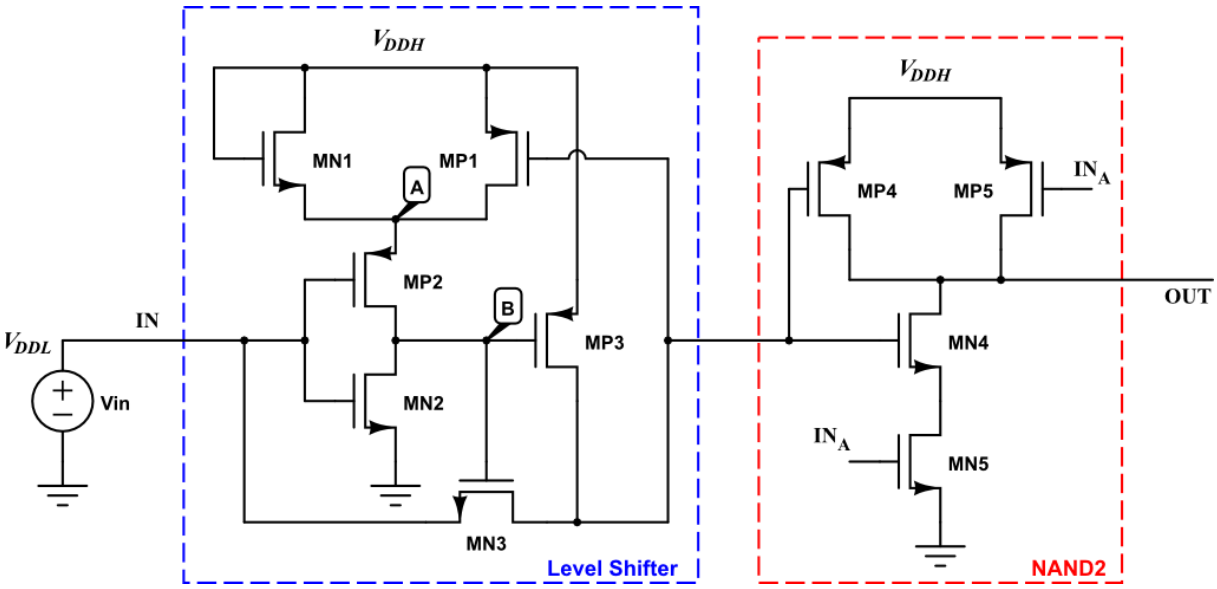
We performed simulations using the set of tools from Cadence ® with the International Business Machines Corp. (IBM) ® CMOS 130nm technology. Two different scenarios compare the three level shifters (DCVS, Puri and EF-LS), in the first scenario the circuits drive a NAND and their inputs come directly from a supply voltage, in the second scenario a NAND gate is also connected to input to evaluate a more real situation in which the level converters have the same *fan-in*. All level shifter transistors were scaled prioritizing the area factor, they all are dimensioned with minimum size (L = 130nm and W = 160nm). Then compared in terms of time delay and power consumption with variable input supply. For reference purposes, this delay is the time it takes the signal to travel from the input to the output OUT. *Rise time* delay and *Fall time* delay are computed when the output stabilizes and reaches 50% of its level nominal value.

5.2 First Scenario

In the first simulation scenario, each level shifter drives a two-input CMOS NAND (*NAND2*) and all the data is measured at the output of the gate. Figure 5.2 illustrates the simulation scheme for the proposed circuit, where V_{DDH} (1.5V) is a nominal voltage that belongs to the same power domain of OUT and V_{in} represents a low power domain (V_{DDL}) that varies from $[V_{DDH} \times 0.95]$ to $[V_{DDH} \times 0.35]$ with a 5% pace. The dotted box in blue contains the level shifter under test and the red dotted box the output NAND. The size of the NAND transistors (MP4, MP5, MN4, MN5) is calculated based on the logical effort (g) of the gate (SUTHERLAND;

SPROULL; HARRIS, 1999). The typical logical effort of a NAND2 is $4/3$. It is defined as *the ratio of the gate input capacitance to the input capacitance of an inverter* that can deliver the same output current. Hence, each transistor has twice the minimum length. IN_A controls the logic of the gate; when the input of the level shifter goes from low to high, IN_A connects to ground; if the input transitions from high to low, IN_A is set to high.

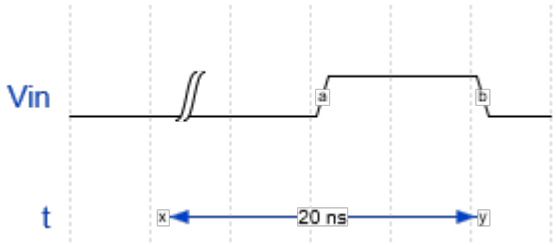
Figure 5.2 – Simulation Scheme of First Scenario



Source: Author

We simulated the three level shifters (DCVS, Puri, and EF-LS) using the set of tools from Cadence ® with the IBM ® CMOS 130nm physical design kit. All transistors were scaled prioritizing area, they are dimensioned with minimum size. Figure 5.3 represents the applied input signal with a slew rate of $0.03ns$ and a $20ns$ period.

Figure 5.3 – Input Signal, slew rate of $0.03ns$



a: Rise time = $0.03ns$; **b**: Fall time = $0.03ns$

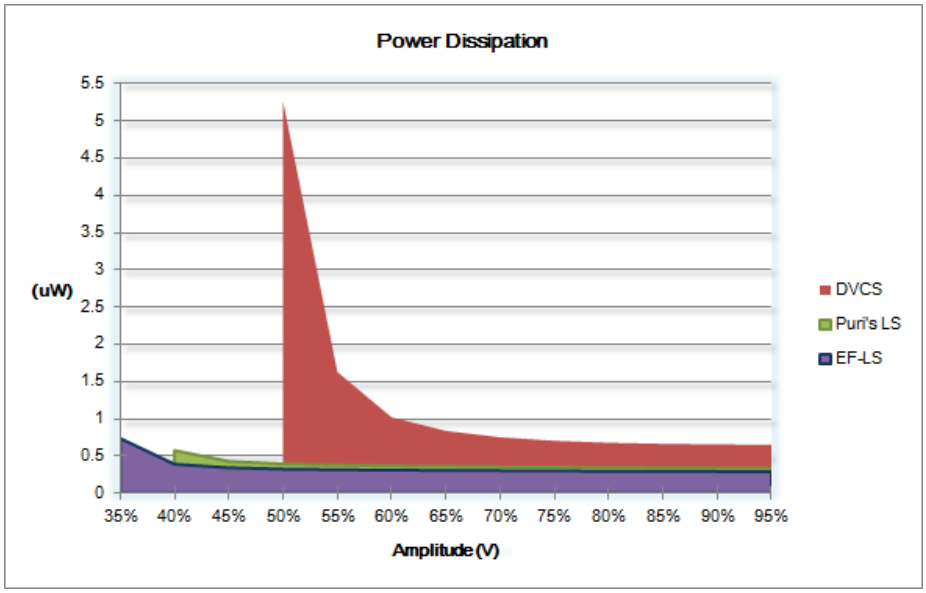
Source: Author

5.2.1 Power Consumption

Power consumption considers that the LS drive a nominal voltage island and their inputs come from a variable power domain (V_{DDL}). With this condition we can determine the input range at which the topology can operate. To measure the power consumption of each topology, it was necessary to obtain the function of the current supplied by V_{DDH} through a charge of $50fF$ connected at the NAND output. Then with the resulting graph, we calculate the *mean* and *RMS* power by multiplying the wave for the corresponding value of V_{DDH}; for this we use the integrated calculator of the *Virtuoso* tool. The mean energy consumption of a pair of logic transitions (low-to-high and high-to-low) is calculated by:

$$Mean\ energy\ (J) = Mean\ power \times Period \tag{5.1}$$

Figure 5.4 – Comparison of Power Dissipation



Source: Author

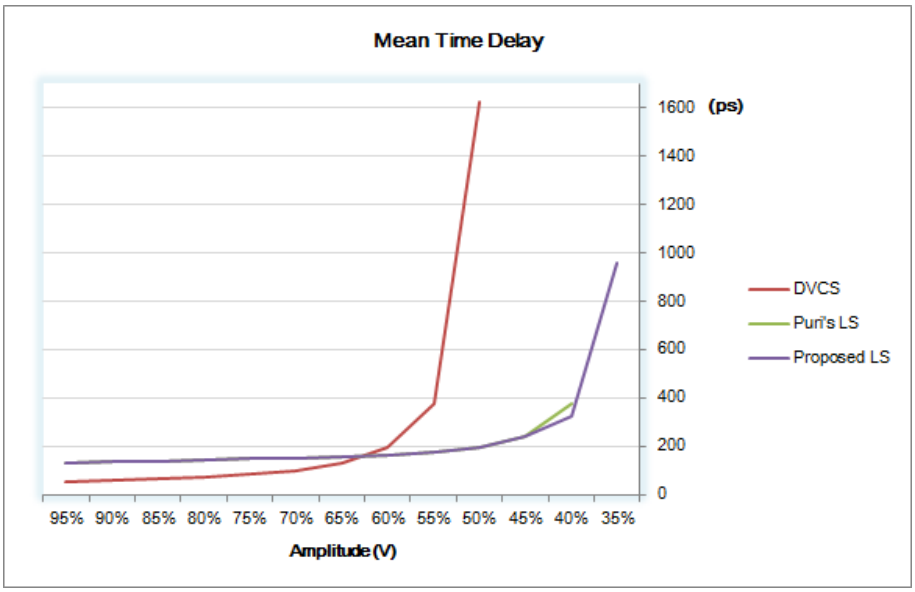
When the input is at 50% of the nominal power supply (0.75V), our topology presented 93.79% and 17.32% lower consumption than DCVS and Puri's LS respectively. The simulations show that the proposed level shifter was the only one capable to operate at 35% of the power supply's amplitude (0.525V). Which opens the possibility to work at near-threshold voltage. Figure 5.4 compares the power dissipation results. When the input voltage is higher than 50% of the nominal supply, we achieved on average 64.2% and 14.9% less power dissipation than the DCVS and Puri LS respectively. The results obtained by the three level shifters are presented

in Table A.1 of Appendix A, where the column *Amplitude* represents the input voltage as a percentage of V_{DDH}. For each input level, the mean power is calculated as explained before. We can observe that the DCVS configuration does not operate for input voltage levels under 50% of nominal supply.

5.2.2 Time Delays

The EF-LS level shifter presented 13.26% smaller total propagation delay (mean time) than the Puri for an input with 40% of the nominal value. For an input at 50% amplitude of the nominal power supply, we obtained 88.03% smaller mean time delay than the DCVS and 1.27% lower delay than the Puri level shifter. The DCVS presented the best results for the input range between 65% and 95% of the nominal supply, as Figure 5.5 depicts.

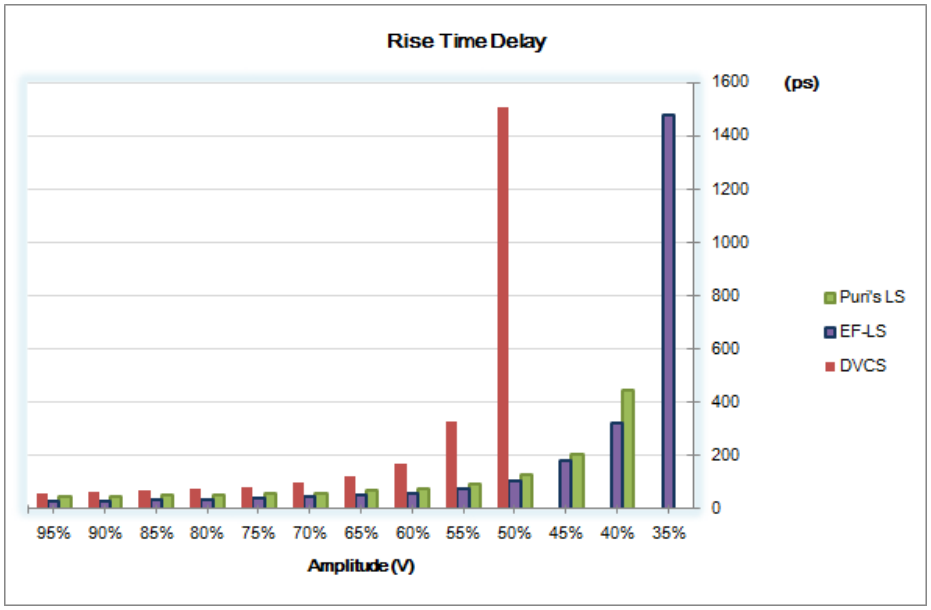
Figure 5.5 – Mean Time Delay



Source: Author

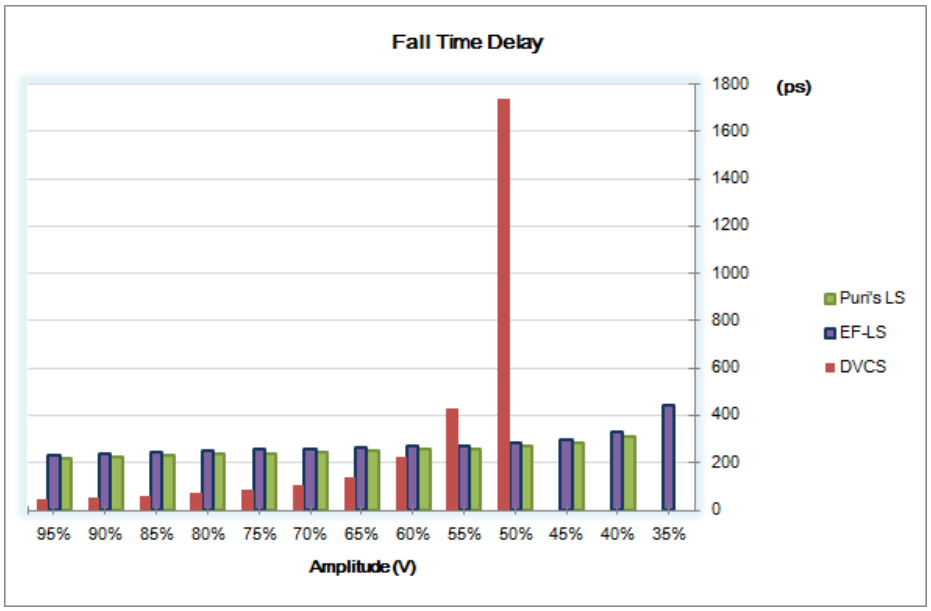
With the modified topology we achieved smaller delays for the *Rise time* and similar results for *Fall time* delays if compared to the Puri's level shifter, as Figure 5.6 and Figure 5.7 illustrate. Again, among the three topologies, the DCVS level converter was the fastest for *Fall time* delays with V_{DDL} input at 60-95% of the nominal supply. Table A.2 presents the data of propagation delays for the level shifters under comparison. The EF-LS obtained the best results for *Rise time* but was considerable slower for *Fall time*. This delay could be caused by a weak pull-down network.

Figure 5.6 – Rise Time Delay



Source: Author

Figure 5.7 – Fall Time Delay



Source: Author

5.2.3 Power-Delay Product

The factor of merit *Power-Delay Product* (PDP) takes into account both, the energy consumed by the circuit and the delay penalty resulting from the logic-level transitions. The

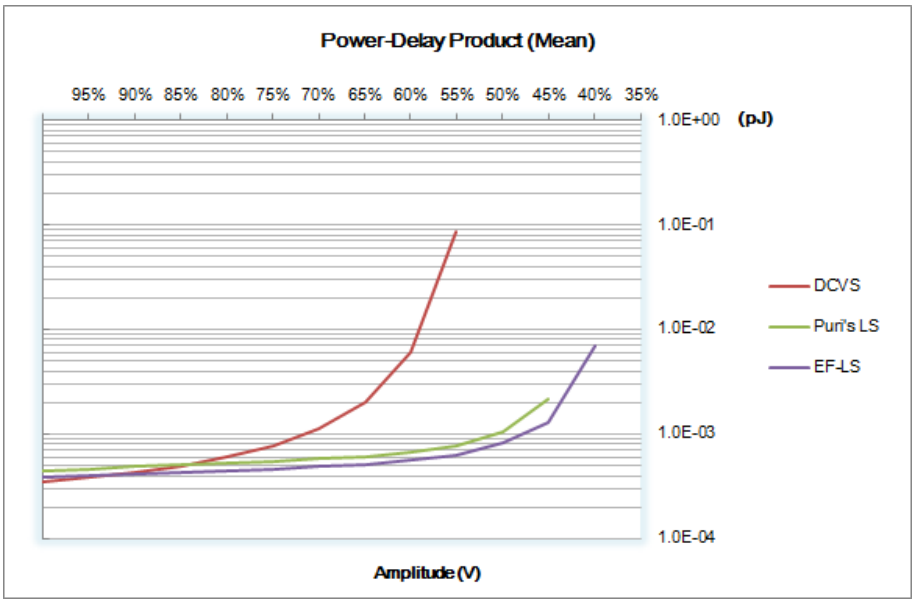
PDP for mean time is calculated as follows:

$$PDP_{Mean} = \left[\frac{Rise\ time + Fall\ time}{2}\right] P_{(\mu W)} \tag{5.2}$$

Figure 5.8 presents the mean Power-Delay product of the three topologies in logarithmic scale. The data shows the DCVS with the biggest PDP; the Puri and EF-LS present similar results with exception of the input range between 50-35% of amplitude. The EF-LS achieved slightly better results for mean time (average of *Rise* and *Fall time*) than the Puri level shifter. Despite the low mean time of the DCVS, the PDP of this topology is bigger because of its high dynamic power consumption.

The proposed circuit presented the best results for PDP when considered the low-to-high logic transitions. On average, it shows 42.31% and 96.8% lower PDP than the Puri and the DCVS respectively as Figure 5.9 depicts. When the input is at 50% of the nominal voltage, the economy in PDP is of 30.44% and 99.56% with reference to the Puri and DCVS LS respectively.

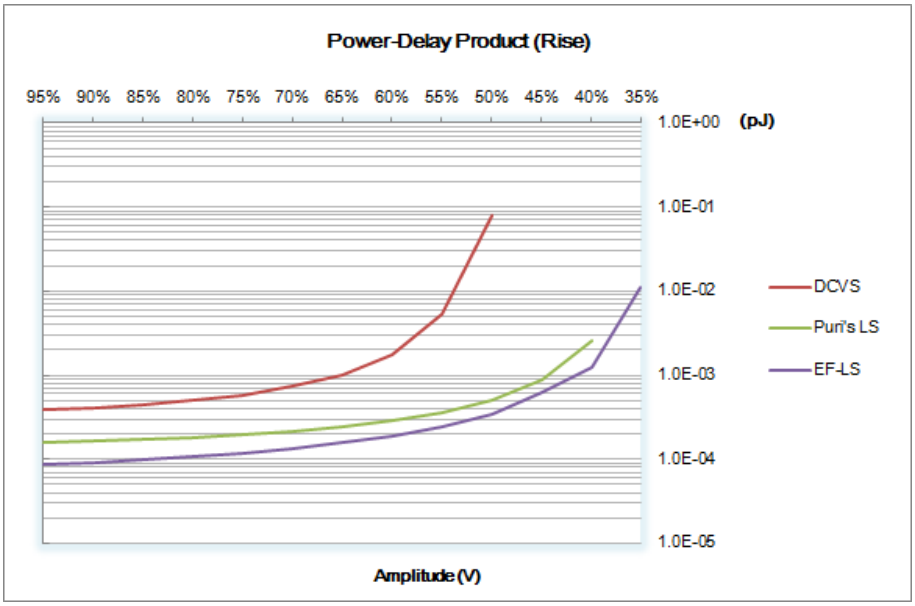
Figure 5.8 – Mean Power-Delay Product



Source: Author

Due to the DCVS fast high-to-low logic transitions, it produces the lowest PDP for the fall condition when the input voltage is bigger than 65% of the nominal supply (Figure 5.10). The DCVS spent on average $49.8fJ$, meanwhile, the EF-LS consumes $73.89fJ$ under these input configurations. Still, it represents an economy of 8.65% for the EF-LS over the Puri level shifter. When the input is at 50% of V_{DDL}, the DCVS spent from 90% to 98% more energy than the other LS. Furthermore, the modified topology presents a reduction of 12.66% in PDP

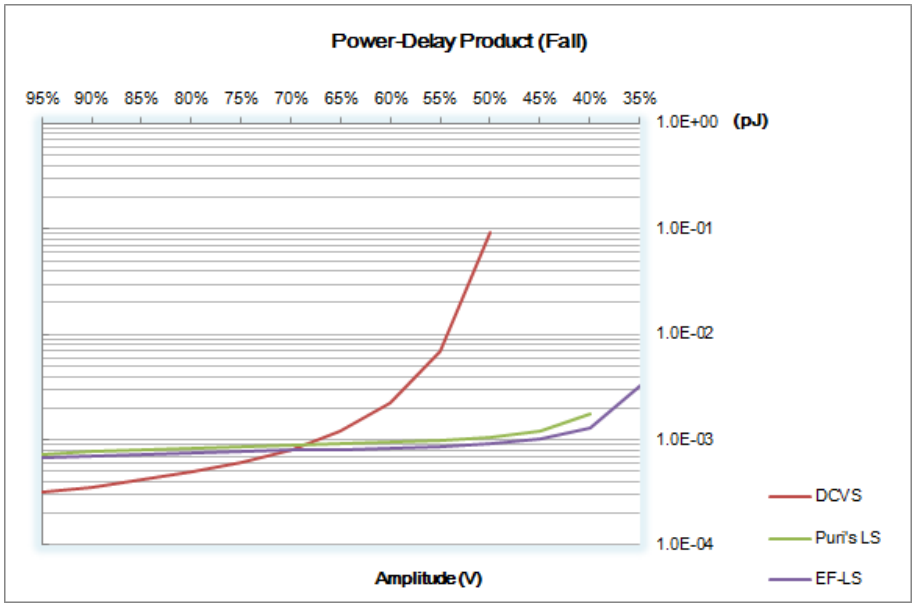
Figure 5.9 – Power-Delay Rise



Source: Author

with reference to the original Puri level converter. Table A.3 and Table A.4 present the resulting data, power is expressed in Joules (J).

Figure 5.10 – Power-Delay Fall

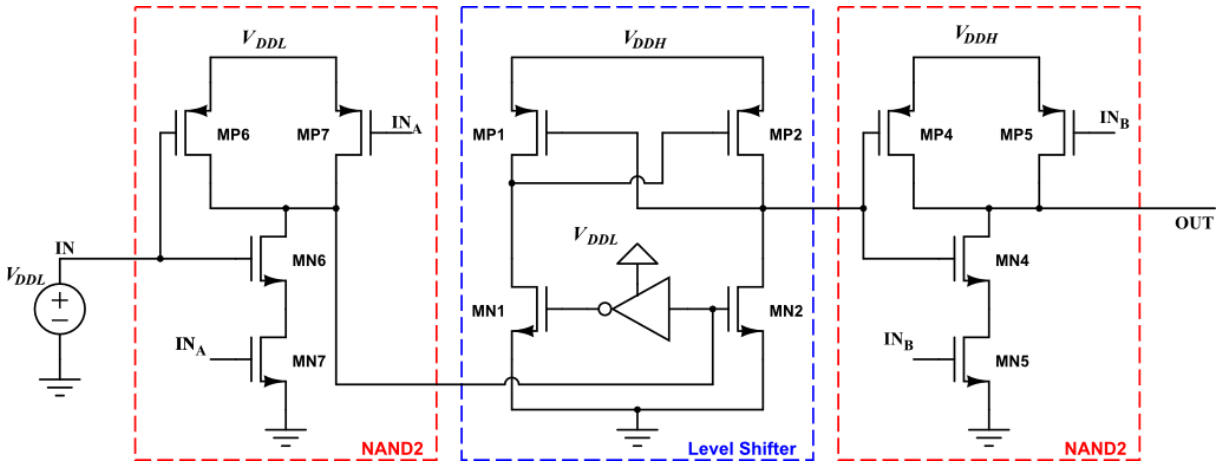


Source: Author

5.3 Second Scenario

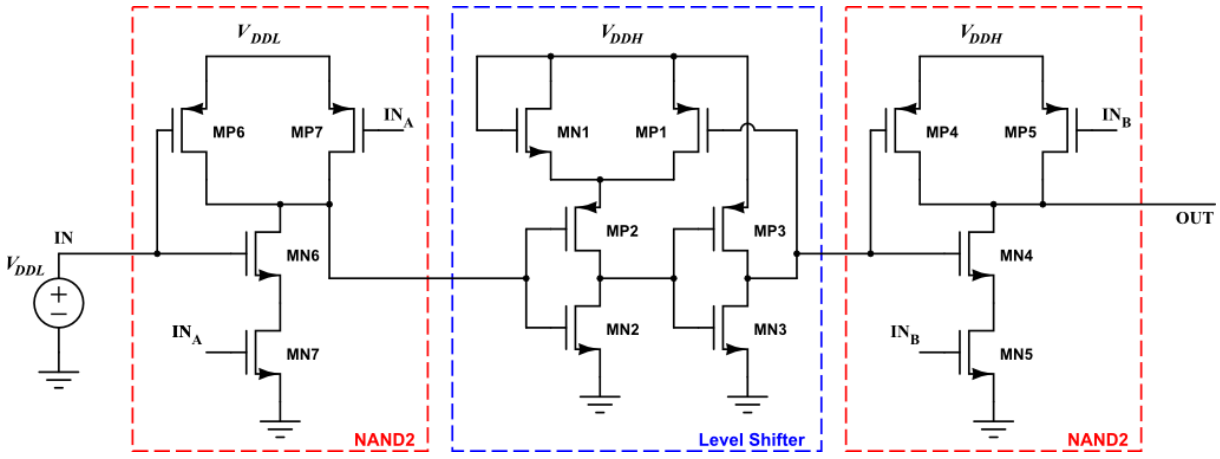
Figures 5.11, 5.12 and 5.13 show the second scenario setup for the three level shifters, this time they are driven by the NAND2 previously introduced. The input NAND belongs to a lower power domain (V_{DDL}) and at the output power domain there is another NAND with the same characteristics.

Figure 5.11 – DCVS's simulation scheme of Second Scenario



Source: Author

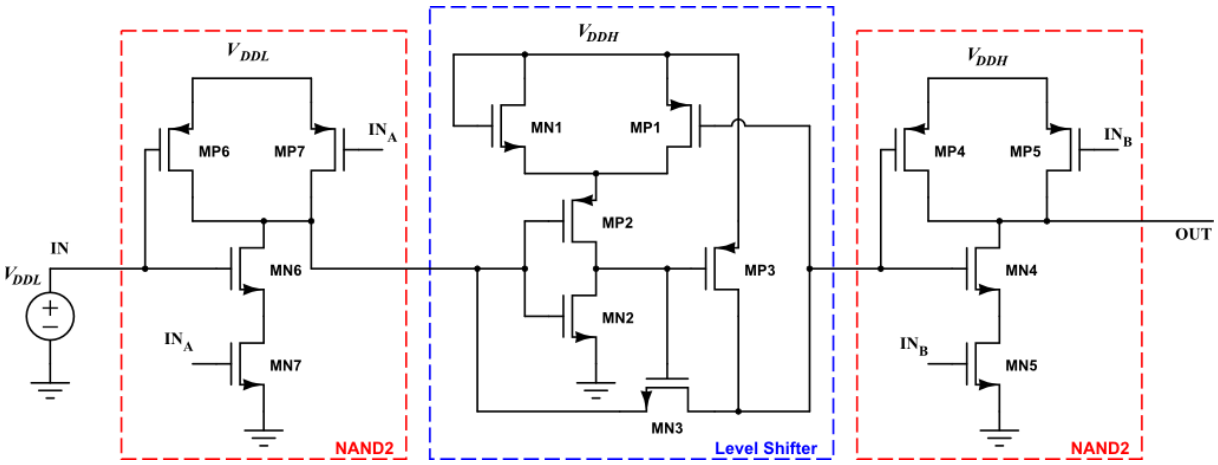
Figure 5.12 – Puri's simulation scheme of Second Scenario



Source: Author

The transistors of the LS are again dimensioned with minimum size (L = 130nm; W = 160nm). Each circuit must shift the input voltage level to the nominal voltage at OUT (1.5V). The input signal V_{in} varies in the range of $[V_{DDH} \times 0.95]$ to $[V_{DDH} \times 0.35]$ at a 5% pace. In the case of the DCVS level shifter, the V_{DDL} that supplies the inverter is adjusted according to

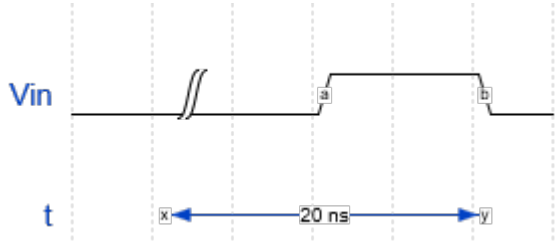
Figure 5.13 – EF-LS's simulation scheme of Second Scenario



Source: Author

the input level to avoid leakage. Initially, the input signal has the same form of the Figure 5.3 (slew rate of $0.03ns$ and a period of $20ns$). Later, the input signal is a modified waveform of it (Figure 5.14), with bigger rise and fall delay times in order to evaluate the circuits response for a different input slew rate. Propagation delays and power consumption are measured and compared for both input situations.

Figure 5.14 – Input Signal, slew rate of $0.3ns$



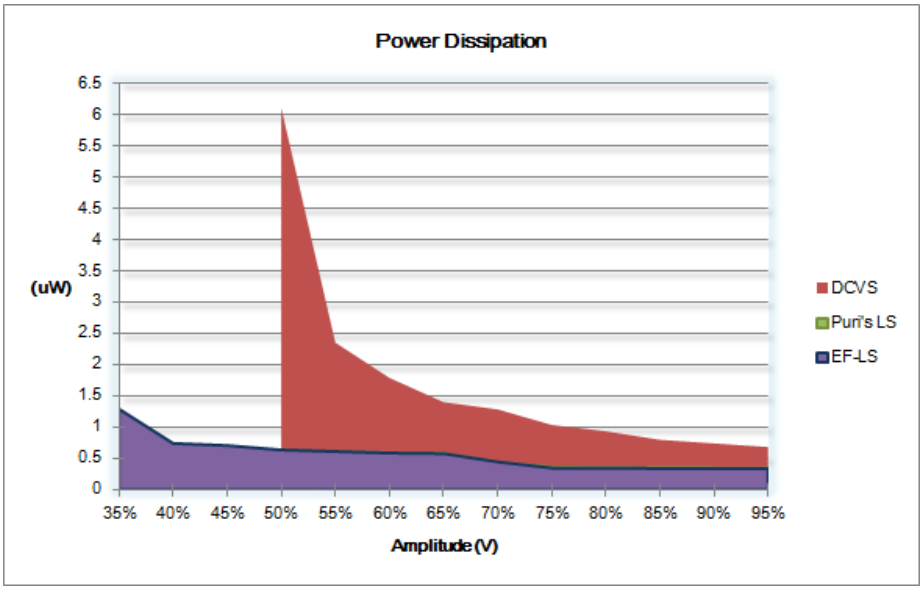
a: Rise time = $0.3ns$; **b**: Fall time = $0.3ns$

Source: Author

5.3.1 Power Consumption

To measure the power consumption of each topology, it was necessary to obtain the function of the current supplied by V_{DDH} through a charge of $50fF$ connected at the NAND output. Then with the resulting graph, we calculate the *mean* and *RMS* power by multiplying the wave for the corresponding value of V_{DDH}; for this we use the integrated calculator of the

Figure 5.15 – Power Dissipation for $0.03ns$ slew-rate input signal



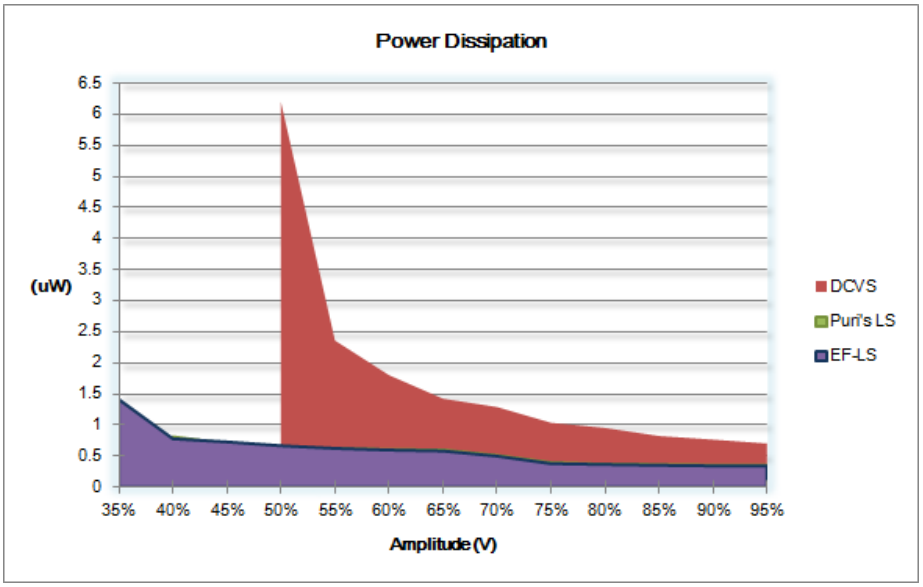
Source: Author

Virtuoso tool. The mean energy consumption of a pair of logic transitions (low-to-high and high-to-low) is calculated in the same way as in the First Scenario.

Figure 5.15 compares the power consumption of the three level shifters for the input signal with a $0.03ns$ slew rate. The results are similar between Puri's and the proposed level shifter for an input voltage level higher than 35%. When the input is under its 60%, our level shifter consumes more than the Puri with a 3.80% higher power dissipation, despite the fact it could work at 35% of the input signal with a power consumption of 1.293μW. For values beyond the 60% of the input, the proposed circuit presented 1.51% lower consumption than the Puri's level shifter and an economy in power of 67.17% with respect of the DCVS. Again, the DCVS was unable to operate for input voltage levels under 50% of nominal supply, presenting 73.32% more power dissipation than our circuit in its range of operation. Table A.5 contains all the obtained data.

The results are similar when the input signal has a slew rate of $0.3ns$, as expected, each level shifter spent more power to complete the transitions due to the increased time delays of the input. For voltage levels under 60% of the nominal supply, the proposed circuit presents 1.378% more power dissipation than the Puri without take into account the 1.410μW spent by our circuit at 35% amplitude. Meanwhile, the economy for higher input levels was of 2.59% with respect of the Puri. Compared to the DCVS, in total, our circuit spent 66.07% less power and consumes 79% less in the DCVS's operation range, as Figure 5.16 shows. The results for these simulations are presented in Table A.9.

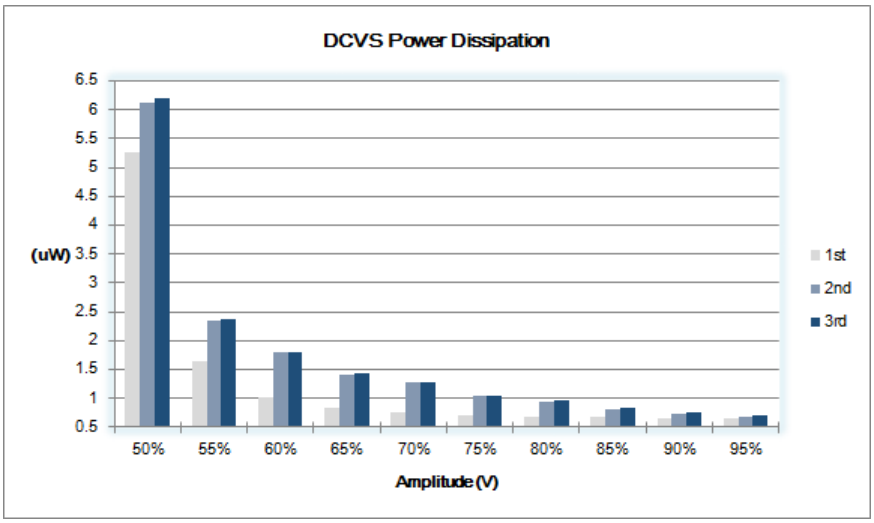
Figure 5.16 – Power Dissipation for $0.3ns$ slew-rate input signal



Source: Author

For comparison purposes, Figures 5.17, 5.18 and 5.19 illustrate the power consumption of the DCVS, Puri and EF-LS level shifters respectively for three different situations: the simulations of the first scenario (1st), the second scenario with $0.03ns$ slew-rate input (2nd) and the same scenario with a $0.3ns$ slew-rate input signal (3rd). Each figure only presents the operation range of the respective level shifter.

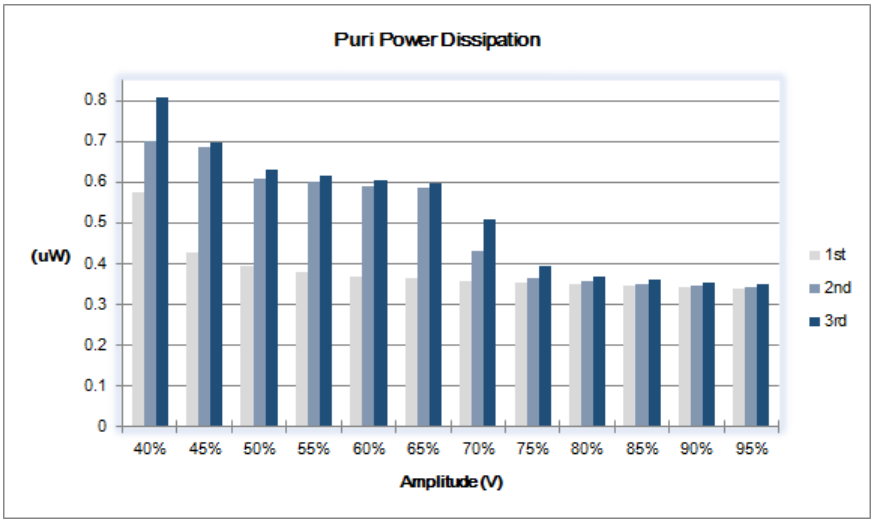
Figure 5.17 – Power Dissipation of DCVS



Source: Author

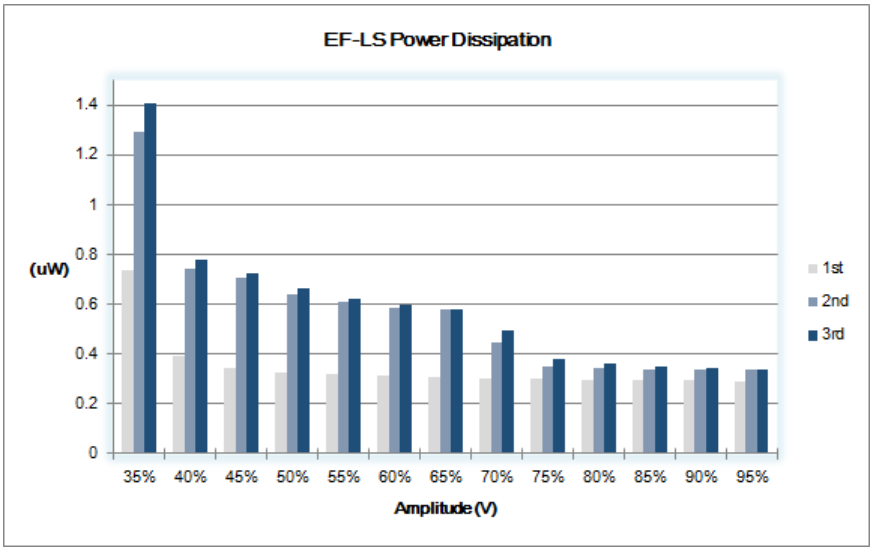
In the case of the DCVS, the change between the two simulation situations of the second scenario is unrepresentative, but in average, there is an increment of 33% of power from the first

Figure 5.18 – Power Dissipation of Puri's level shifter



Source: Author

Figure 5.19 – Power Dissipation of proposed level shifter



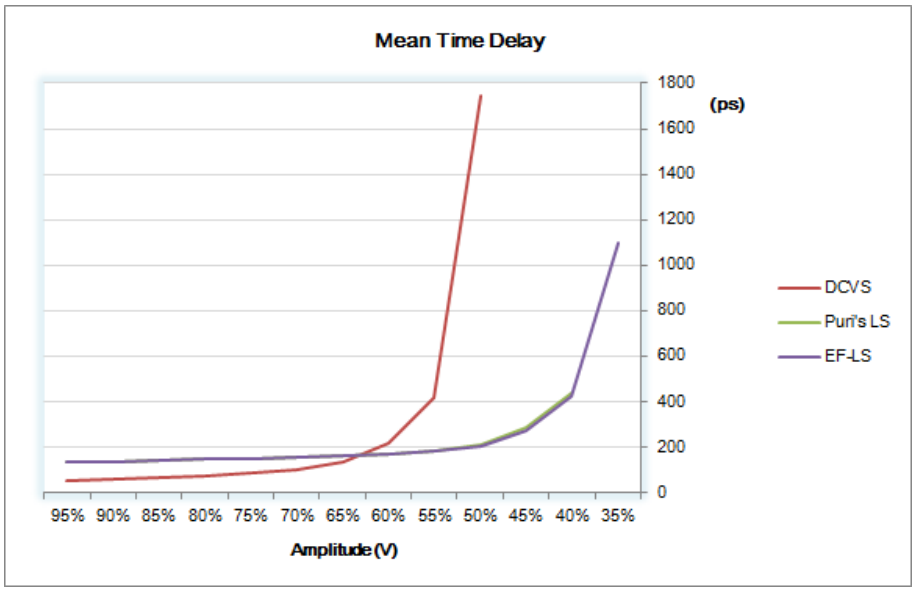
Source: Author

scenario. The greatest increment in power for the Puri's circuit is when the input signal is at 40% of the nominal amplitude, it is a difference of 108nW between the 2nd and 3rd situations. In average, it spent 29% more power than the first scenario. The proposed level shifter spent twice the amount of power consumed in the first scenario for input signals between the range of 35% and 65% amplitude. In average, it dissipates 62% more power than the spent in the first scenario.

5.3.2 Time Delays

When the input signal has a slew rate of $0.03ns$, the EF-LS level shifter presented 1.362% smaller total propagation delay than the Puri, without accounting the operation at 35% of the nominal value. The proposed level shifter presented lower values for input voltage under 70% the amplitude, but has an increment when the input signal is bigger than 1.05V. The EF-LS level shifter showed 22.5% minor mean time delay when compared with the DCVS circuit. It presents lower propagation delay for all input values, achieving the biggest reduction when the input signal is 50% of the nominal amplitude, it is 88.2% smaller mean time than the DCVS. Figure 5.20 shows the comparison of the mean time propagation for the three level shifter.

Figure 5.20 – Mean time for $0.03ns$ slew-rate input signal

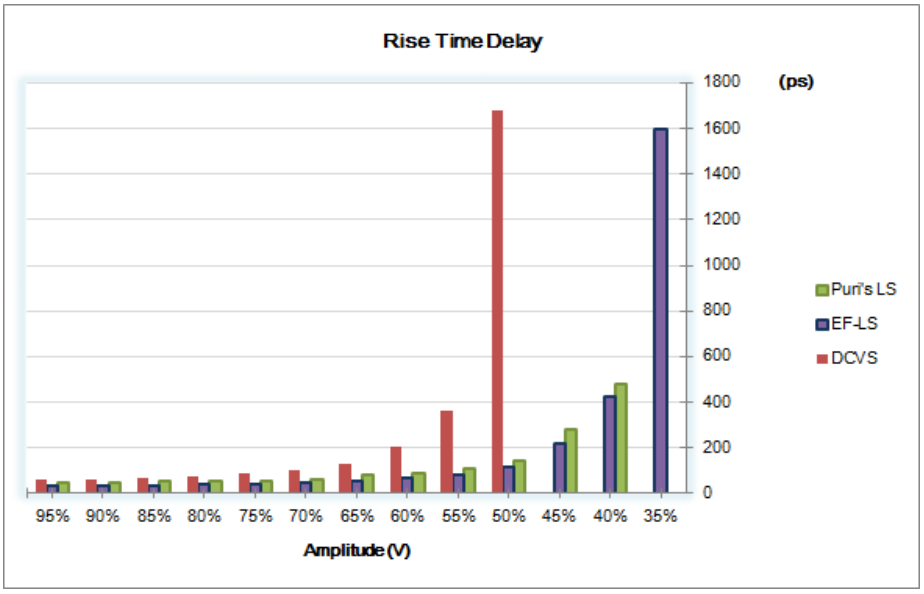


Source: Author

Under this scenario, the modified level shifter achieved smaller delays than the Puri for the *Rise time* and similar results for *Fall time* delays, as Figure 5.21 and Figure 5.22 illustrate respectively. Again, among the three topologies, the DCVS level converter was the fastest for *Fall time* delays with V_{DDL} input range of 60-95% of the nominal supply. Table A.6 presents the obtained data for propagation delays of the three level shifters. The EF-LS obtained the best results for *Rise time* but was considerable slower for *Fall time* despite the narrow difference. This delay could be caused by a weak pull-down network.

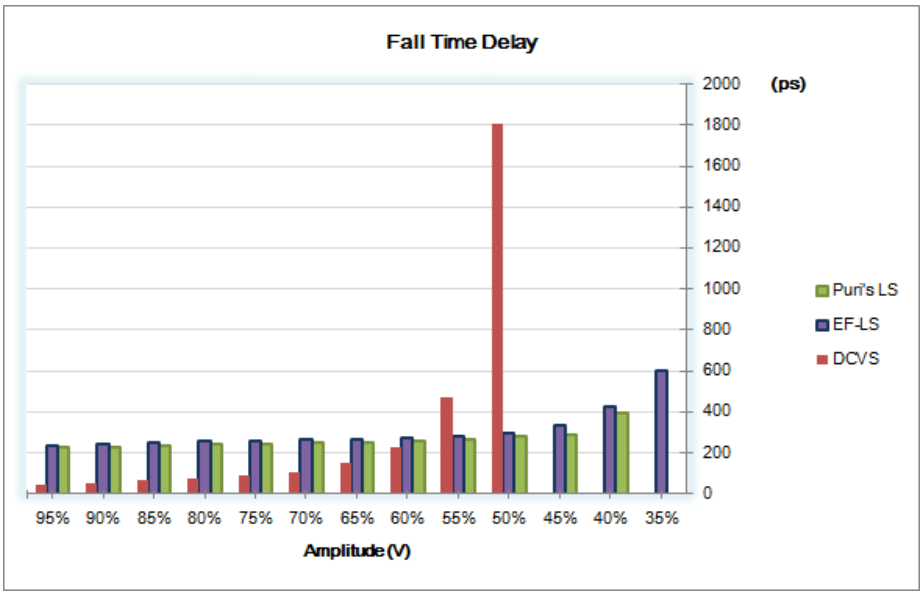
For a slower input signal, with a slew rate of $0.3ns$, the EF-LS level shifter presented bigger mean time delays than the Puri, showing an increase of 0.86% in average. The worst result happens when the input signal is 35% of the nominal supply, with a difference of $39ns$ it

Figure 5.21 – Rise Time Delay for 0.03ns slew-rate input signal



Source: Author

Figure 5.22 – Fall Time Delay for 0.03ns slew-rate input signal

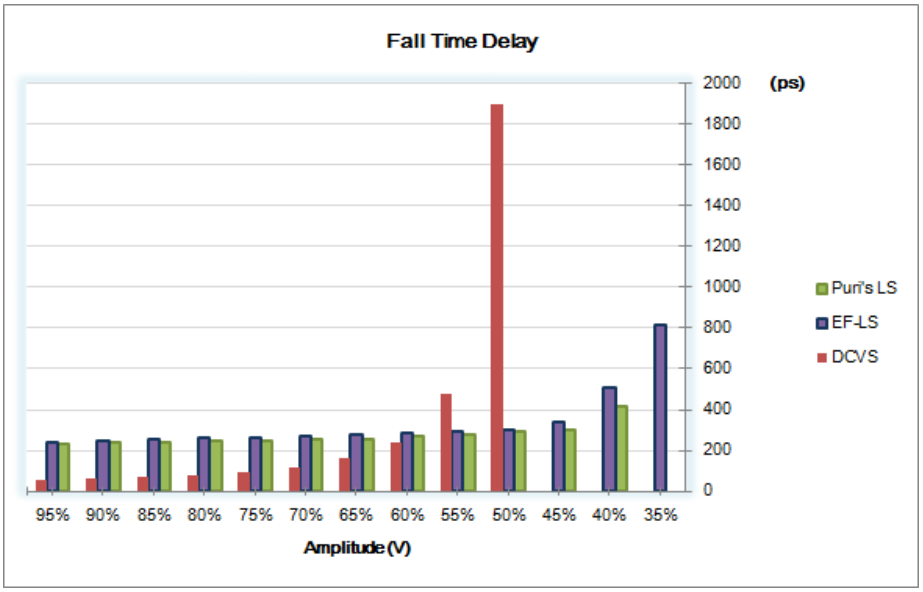


Source: Author

represents an increase of 8.32% with respect to the Puri's response. These deteriorated results may be justified by the increase in the fall-time response of the proposed level shifter. Figure 5.23 compares the *fall time* of the three level shifters, again, the DCVS level converter presents the best results for the input range between 60-95% of the nominal supply, with 56% and 58.5% lower time delay than the Puri and EF-LS respectively.

Figure 5.24 shows the *rise-time* delays of the three level shifters, the EF-LS presented

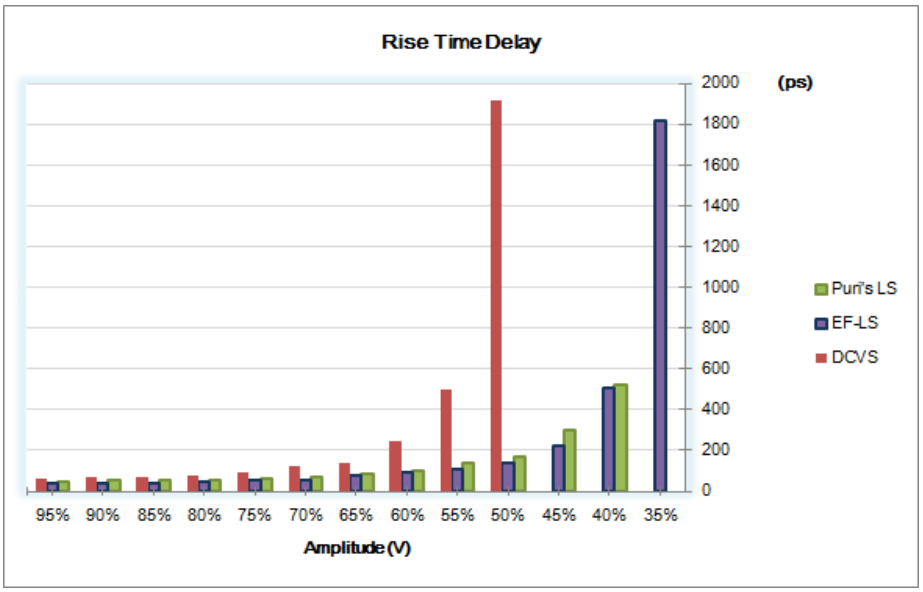
Figure 5.23 – Fall Time Delay for $0.3ns$ slew-rate input signal



Source: Author

14.63% and 57.2% less time than the Puri and DCVS topologies respectively. Table A.10 contains the obtained data in this simulation situation.

Figure 5.24 – Rise Time Delay for $0.3ns$ slew-rate input signal

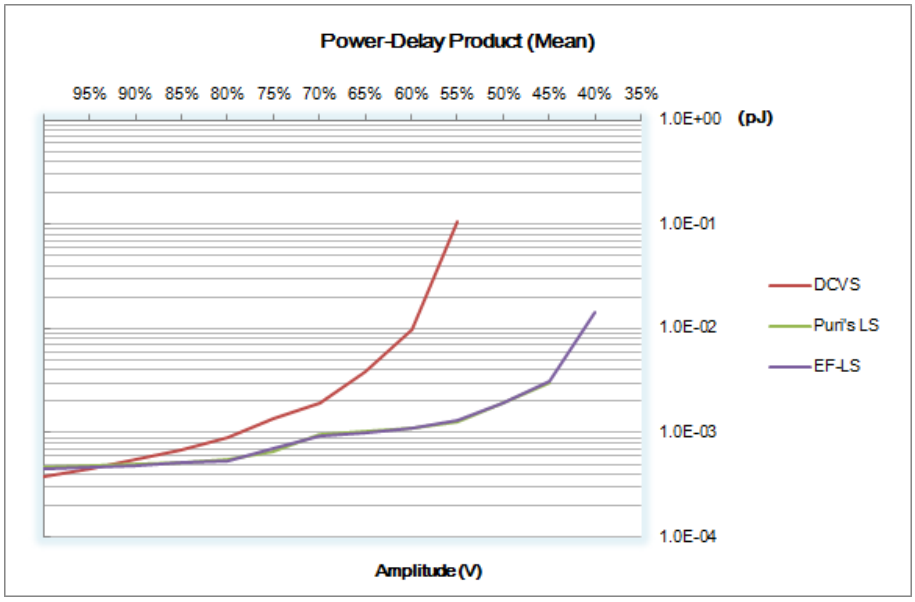


Source: Author

5.3.3 Power-Delay Product

Figure 5.25 and 5.26 present the mean Power-Delay product of the three topologies for the first input condition (slew rate=0.03ns) and second input condition (slew rate=0.3ns) respectively. The data shows the DCVS with the biggest mean PDP (average of *Rise* and *Fall time*), except when the input's amplitude is between 90-95% of V_{DDH}. The Puri and EF-LS present similar results, in average there is just a difference of 0.1% between both. The EF-LS achieved better results for mean PDP than the Puri in the input range of 70% and 95% of the nominal supply, it achieves 3.4% lower PDP. When compared to the DCVS, the proposed circuit presented 9.4% less PDP in average. For the curious reader, the obtained data for both input signals is presented in Table A.8 and Table A.12 of Appendix A.

Figure 5.25 – Mean Power-Delay Product for $0.03ns$ slew-rate input signal

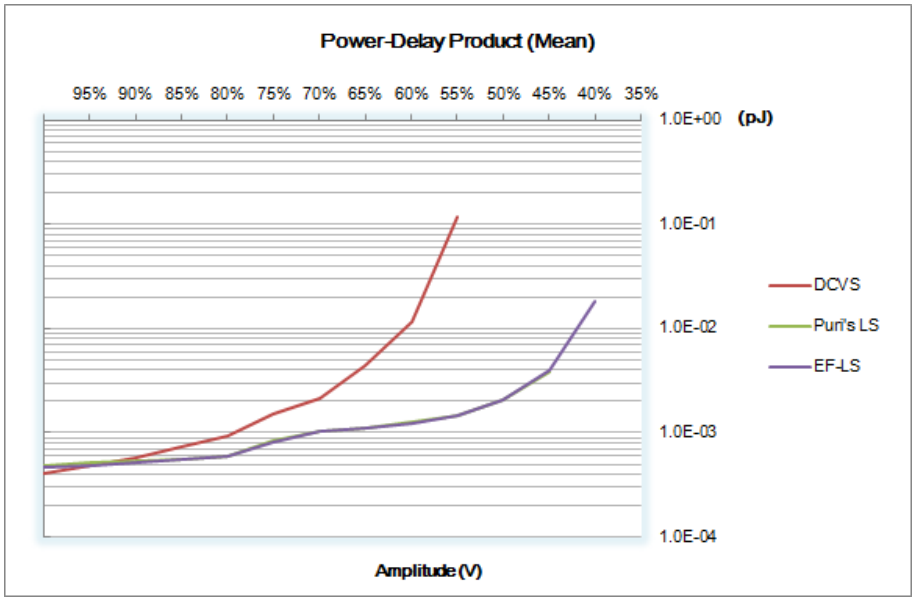


Source: Author

In both input situations, the proposed circuit presented the best results of power-delay product when considered the low-to-high logic transitions. On average, it shows 14-16% lower PDP than the Puri and around 97% lower than the DCVS, as Figure 5.27 and Figure 5.28 depict. When the input is at 50% of the nominal voltage, the economy in PDP is of 16-17.5% and 93-97% with reference to the Puri and DCVS LS respectively.

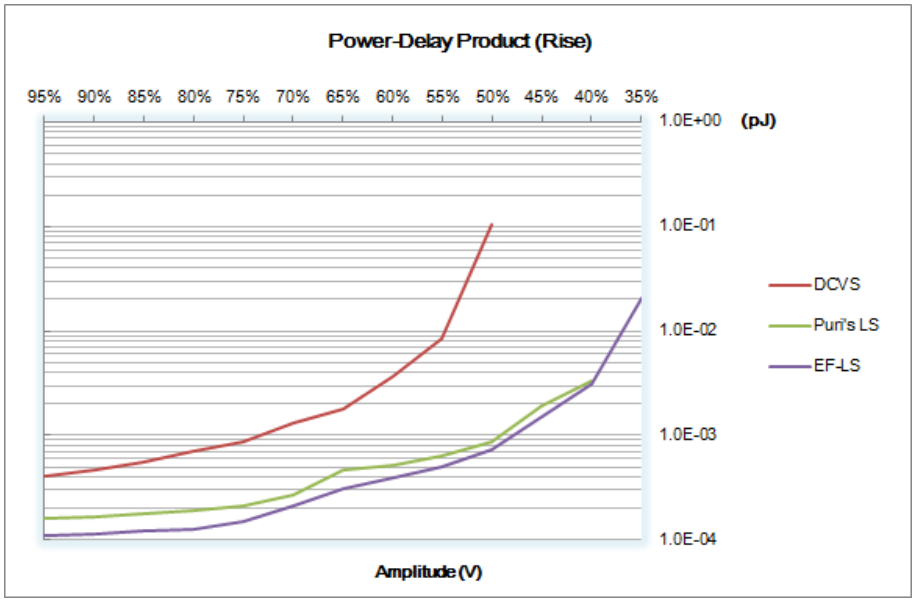
Consistent with the results obtained in the first simulation scenario, the DCVS produces the lowest PDP for the fall condition when the input voltage is bigger than 65% of the nominal supply (Figures 5.29 and 5.30). The DCVS spent on average 71fJ, meanwhile, the EF-LS consumes 91fJ and the Puri 85fJ under the same input configurations. On average, the EF-LS

Figure 5.26 – Mean Power-Delay Product for $0.3ns$ slew-rate input signal



Source: Author

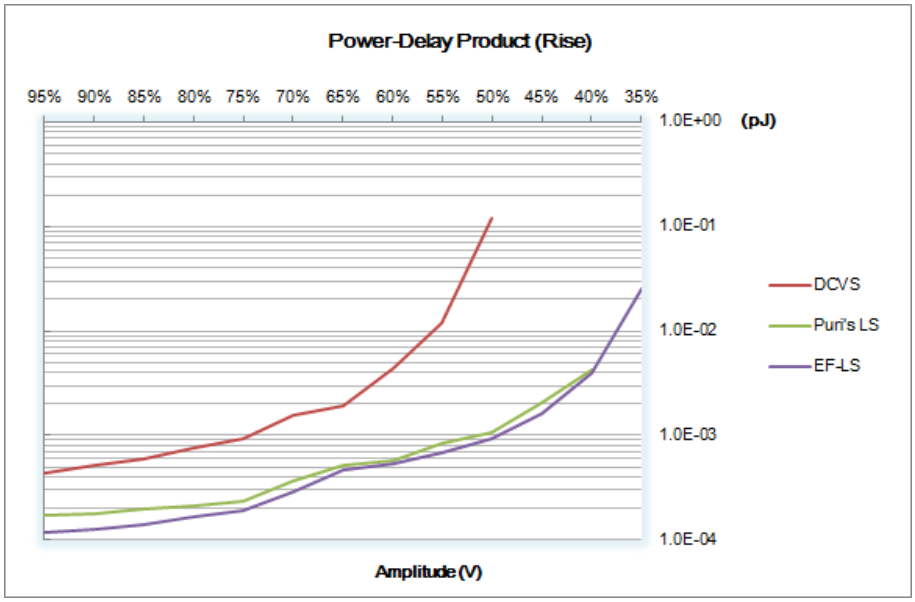
Figure 5.27 – Power-Delay Rise for $0.03ns$ slew-rate input signal



Source: Author

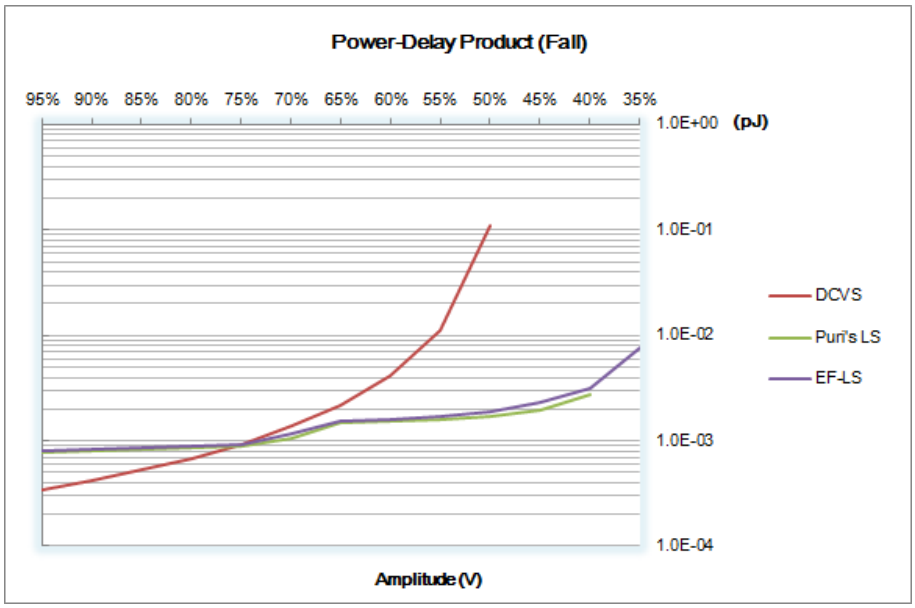
showed 9-9.7% higher PDP-Rise for all the input voltage levels. Table A.7 and A.11 present the resulting data for both input signals.

Figure 5.28 – Power-Delay Rise for $0.3ns$ slew-rate input signal



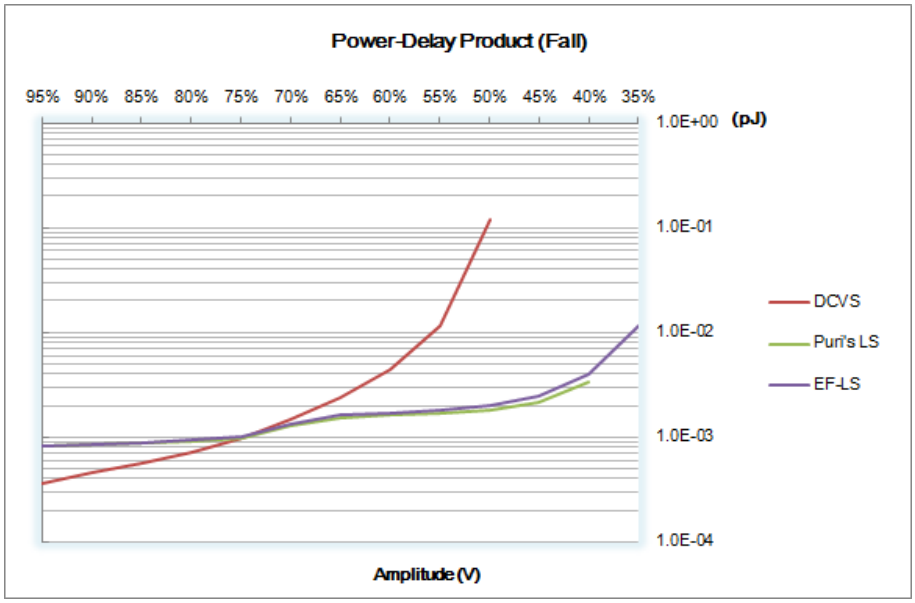
Source: Author

Figure 5.29 – Power-Delay Fall for $0.03ns$ slew-rate input signal



Source: Author

Figure 5.30 – Power-Delay Fall for $0.3ns$ slew-rate input signal



Source: Author

6 CONCLUSIONS

There is a broad range of approaches for optimizing power consumption of VLSI systems, here we have presented some of the most common. From designs with multiple voltages to power-awareness designs, optimizations at the architecture or system level can enable more effective power minimization and maintain performance at the same time. While circuit techniques may yield improvements in the 10-50% range, architecture and algorithm optimizations have reported power reduction by orders of magnitude (RABAEY, 2009). This is true because, at higher abstraction levels, design modifications tend to have a greater potential impact, allowing circuit level techniques such as voltage scaling or sizing to be more efficient. Moreover, it is well known that the *Return-on-Investment* (ROI) increases at higher levels of the design process. Hence, the relevance of architectural approaches like Multi-Voltage in the design of IC.

Power Gating and Voltage Scaling optimizations rely on moving away from the traditional approach of using a single, fixed supply rail for all of the gates in the design. Accordingly, Multi-Voltage design exploits these concepts (explained in Chapter 2) to reduce power consumption of modern digital systems. In Multi-Voltage approaches, the cells that belong to critical paths may be driven by nominal voltage (V_{DDH}) and the cells that are not part of the critical paths will have a lower supply voltage (V_{DDL}). In the same way, cells that are not being used can be associated with the standby mode caused by applying a neutral voltage (V_{off}).

We found that the well-known techniques for V_{DD} assignment are based on the *Clustered Voltage Scaling* (CVS) algorithm (USAMI; HOROWITZ, 1995). The CVS algorithm achieves an economy of 10-20% in power by assigning lower supply voltage to the cells that are off the critical paths and do not drive cells supplied with nominal voltage. Hence, the signal level conversion functionality is restricted to the *Level Converting Flip-Flops* (LCFFs). Employing *Asynchronous Level Converters* (ALC), the *Extended Clustered Voltage Scaling* (ECVS) algorithm (USAMI et al., 1998) bypasses the topological constraint of the CVS and allows V_{DDL} driven cell to be part of the fan-in of a V_{DDH} cell. ECVS leads to 20-30% higher energy savings because bigger portions of the circuit can be clustered into low voltage regions. The *Greedy*-ECVS algorithm (KULKARNI; SRIVASTAVA; SYLVESTER, 2004) improves the performance and power savings of the CVS by reducing the group of V_{DDL} cells driving V_{DDH}. It creates bigger V_{DDL} cluster cells and removes level converters on particular non-critical logical paths with enough slack. GECVS achieves 25-35% power savings with reference to the initial design.

Multi-voltage designs implement *Dynamic Voltage Scaling* by using different supplies in specific regions of the chip. Each region belongs to a power domain and may have two or more supply voltage configurations. Regardless of distinct energy levels on different power domains, the logical blocks shall process signals with coherent logic levels. When driving signals between power domains with radically different power rails, the level converters are essentials. These voltage scaling interfaces (or *Level Shifters*) are at the power domain boundaries, and they must ensure the correct communication between two regions by providing the corresponding voltage levels. The level shifters also guarantee the voltage swings that the particular domain expects, avoiding *rise* or *fall*-time degradation that could lead to timing closure problems. The acceptable high and low voltages are delimited by the V_{IH} and V_{IL} voltage regions. For proper circuit operation, the region between V_{IH} and V_{IL} (*transition width*) should be avoided. Therefore, to interface two power domains successfully, we must meet the following requirements:

- The V_{OH} of the driver cell must be greater than the V_{IH} of the receiver power domain.
- The V_{OL} of the driver must be less than the V_{IL} of the receiver.
- The output voltage from the driver must not exceed the I/O voltage tolerance of the receiver.

Many EDA tools allow the designer to specify a level shifter placement strategy. It is, to place the level shifter in the lower power domain, in the higher domain, or between them. Besides, the designer can specify which blocks require voltage level conversion or establish a minimum voltage difference that requires level shifter insertion. These tools must perform level shifter or buffer insertion in two different situations: one in which the signals travel from a domain with high voltage levels to a low power domain (*high-to-low*); and a second case in which V_{DDL} cells drive blocks in a higher power domain (*low-to-high*). *Low-to-High* level conversion is a more critical problem because under-driven signals degrade the rise and fall times at the receiving inputs. It can lead to higher switching currents and reduced noise margins. A slow transition time means that the signal spends more time near the threshold voltage, causing the short circuit current to last longer than necessary.

We reviewed several level shifter topologies primarily designed for *low-to-high* voltage conversion. We categorize them into *Single-rail* and *Dual-rail* groups (SLS and DLS respectively), depending on the number of power lines required. We found that the vast majority are based on the *Differential Cascode Voltage Switch Logic* (DCVS logic). The DCVS level shifter has the advantage of a simple design, well suited for higher core voltages. However, this topology exhibits high short circuit current during transitions because the PMOS gates experience

full voltage swing from 0V to V_{DDH}. Also, it can only operate at a restricted range of voltage inputs.

In (KOBAYASHI et al., 1993), the voltage range conversion of the DCVS is extended by replacing the PMOS half-latch with a current mirror. But this increases the power consumption because the leakage path created by the PMOS transistors and one of the pull-down devices. Ashouei et al., (2010) proposed a DCVS-based topology that can also operate with substantially different voltage levels. Although this level shifter benefits from non-conflicting rise and fall transitions, it has the issue of a '*weak*' pull-down network. To address this issue, it requires an extra inverter stage that flips the input signal and also increase delay.(KHAN; WADHWA; MISRI, 2006) introduces a DCVS-based level shifter that only requires one power rail. It uses an MOS capacitor (MOSCAP) to pre-charge the output value when the input is "low". The MOSCAP acts as a start-up when the circuit is powered up, becoming a limitation because it conditions the power sequence of the design.

The Bootstrapping technique appears as a way to reduce the dynamic power of DCVS LS. (TAN; SUN, 2002) presented a level converter with bootstrapped gates, that uses capacitors to maintain the voltage difference between the gates of pull-up PMOS and pull-down NMOS. The power saving over conventional LS is achieved by the reduced voltage swing at specific nodes. The bootstrapping level shifter attains lower power at the expense of a significant increase in physical area due to the relatively large bootstrap capacitors. Moreover, the size of these capacitors defines the switching delay between low and high voltage states. If they are too small, the voltage swings will be reduced owing to the charge redistribution.

Feedback-based approaches take advantage of the V_{DDH} voltages to increase the current flow and speed up logic transitions. The circuit in (HAMADA et al., 1998) consumes less energy than the DVCS level shifter due to fewer devices and less switching activity, but it has higher static power consumption. (BO; LIPING; XINGJUN, 2007) presented a novel topology that uses the *high* and *low* logic levels of the V_{DDL} signal to control simultaneously the "on" and "off" state of the NMOS transistors at the input stage. The output of this converter is a half latch that pulls up the input of the inverter at the output stage. Compared to the DCVS, this level shifter presented a reduction of 36% in the leakage power dissipation, but it has an increase of 5% in delay. (GUPTA et al., 2008) proposed a level converter that uses a feedback mechanism to shutoff the static current path that is created during *low-to-high* transitions. It shows more robust operation than the DCVS level shifter in terms of consumption but has higher time delays under certain conditions.

The needs of the DLS for two voltage supply rails limit the physical placement of such

circuits to the boundaries of V_{DDL} and V_{DDH} voltage regions. It also restricts the physical design flexibility and the operation range of DCVS-based level converters. We explained the advantage of SLS over DLS. With the former we can afford fewer pin count; reduced congestion in supply routing, complexity and overall system cost. The impact in the Power Planning, Placement, and Routing phases lessens when preferred SLS over DLS. For instance, in a complex multi-voltage system, the placement and routing of single-rail level shifters could be much easier than if we use dual-rail level shifters. Moreover, in some cases, single-rail level shifters outperform the dual-rail level shifters. It obviously depends on the topology, circuit optimizations, and final application.

(STOK et al., 2007) proposed one of the first single-rail level shifters that is not based on the DCVS. In this topology, the threshold drop across a NMOS transistor provides a virtual V_{DDL} to the input stage of the circuit. The output stage of the Puri's level shifter is a half latch that pulls up the input to V_{DDH} in order to avoid leakage. It can suffer from higher leakage currents when the input signal level is lower, or V_{DDH} is higher, than the supply voltage by more than V_T. It can increase power consumption on a chip with a large number of level shifters. Also, this topology has bigger delay times for input levels near to nominal voltage. Mainly because of the diode connected transistor (MN1 in Figure 4.12) limits the operation speed of the circuit.

We proposed a modification of the Puri level shifter and compared it with the traditional DCVS and the Puri level shifters, under two different simulation scenarios: the first without fain and the second scenario with a NAND2 at the fan-in. With this modification, we try to reduce the delay of the original topology and the leakage current for low input signals that the Puri's level shifter presents. From the simulations, we observe a reduction in power consumption and a wide range of operational voltage levels of the proposed circuit (EF-LS). The obtained data shows that the proposed level shifter was the only one capable of operating at 35% of the power supply's amplitude.

When the input is at 50% of the nominal power supply, and there is not a fan-in (first scenario), our topology presented 93.79% and 17.32% lower consumption than DCVS and Puri's LS respectively. For an input higher than 50% of the nominal supply, the modified level circuit achieved on average 64.2% and 14.9% less power dissipation than the DCVS and Puri level shifters respectively. In the second scenario, the EF-LS spent in average 0.8% more power than the Puri's level shifter when the input has a slew rate of $0.03ns$, but for values beyond the 60% of the input, the proposed circuit presented 1.51% lower consumption. When compared to the DCVS, the proposed circuit presents an economy of 70% in average. For an input signal

with slower rise and fall time (slew rate of $0.3ns$), our circuit spent 0.85% less power than the Puri and presents an mean economy of 66.07% with respect of the DCVS, 79% less power in its operation range.

In terms of propagation delay, the modification affects the performance of the original circuit for *Fall time* but shows better results for *Rise time*. For an input at 50% amplitude of the nominal power supply, the EF-LS obtained 88.03% smaller mean time delay than the DCVS and 1.27% lower delay than the Puri level shifter in the first scenario. The DCVS presented the best results for the input range between 65% and 95% of the nominal supply. For the scenario with fan-in, when the input signal has a slew rate of $0.03ns$, the EF-LS level shifter presented 1.362% smaller total propagation delay than the Puri, without taking into account the operation at 35% of the nominal value. The proposed level shifter obtained the lowest values for input voltages under 70% the amplitude, but has an increment when the input signal is bigger than 1.05V. The EF-LS level shifter showed 22.5% minor mean time delay when compared with the DCVS circuit. When the input signal has a slew rate of $0.3ns$, the EF-LS level shifter presented bigger mean time delays than the Puri, showing an increase of 0.86% in average. The worst result happens when the input signal is 35% of the nominal supply, with a difference of $39ns$ it represents an increase of 8.32% with respect to the Puri's propagation delay. These deteriorated results may be justified by the increase in the fall-time response of the proposed topology.

The EF-LS circuit presented the best results for Power-Delay Product when considered the *low-to-high* logic transitions. On average, it shows 42.31% and 96.8% lower PDP than the Puri and the DCVS respectively. When the input is at 50% of the nominal voltage, the economy in PDP is of 30.44% and 99.56% with reference to the Puri and DCVS level shifters respectively. When the input voltage is bigger than 65% of the nominal supply, the DCVS produces the lowest PDP for the fall condition.

In general, the results show that the DCVS has the fastest responses but consumes more energy. The EF-LS presents similar results for propagation delay when compared to the Puri's circuit, but outperforms both (DCVS and Puri) in energy efficiency under some circumstances.

As future work, we can cite the following:

- Test the circuit in a more recent technology node and compare it with other level shifters presented in Chapter 4 (currently under development).
- Perform further tests to evaluate the frequency performance of the proposed topology (currently under development).
- Generate the layout of the circuit and extract parasitic to analyze its impact on performance and power consumption.

- Test the proposed topology inside a Multi-Voltage system and evaluate its performance under several simulation corners.

REFERENCES

ABDOLLAHI, A.; FALLAH, F.; PEDRAM, M. Leakage current reduction in CMOS VLSI circuits by input vector control. **IEEE Transactions on Very Large Scale Integration (VLSI) Systems**, v. 12, n. 2, 2004. ISSN 1063-8210.

ASHOUEI, M. et al. Novel wide voltage range level shifter for near-threshold designs. In: **2010 IEEE International Conference on Electronics, Circuits, and Systems, ICECS 2010 - Proceedings**. [S.l.]: IEEE, 2010. p. 285–288. ISBN 9781424481576.

ATHAS, W. Low-power VLSI techniques for applications in embedded computing. In: **Proceedings IEEE Alessandro Volta Memorial Workshop on Low-Power Design**. [S.l.]: IEEE, 1999. ISBN 0-7695-0019-6.

BENINI, L.; DEMICHELI, G. **Dynamic Power Management: Design Techniques and CAD Tools**. [S.l.]: Springer Science & Business Media, 1998. 231 p. ISBN 9780792380863.

BO, Z.; LIPING, L.; XINGJUN, W. A new level shifter with low power in multi-voltage system. In: **ICSICT-2006: 2006 8th International Conference on Solid-State and Integrated Circuit Technology, Proceedings**. [S.l.: s.n.], 2007. p. 1857–1859. ISBN 1424401615.

BORAH, M.; IRWIN, M.; OWENS, R. Minimizing power consumption of static CMOS circuits by transistor sizing and input reordering. In: **Proceedings of the 8th International Conference on VLSI Design**. [S.l.: s.n.], 1995. ISBN 0-8186-6905-5. ISSN 1063-9667.

BORAH, M.; OWENS, R.; IRWIN, M. Transistor sizing for low power CMOS circuits. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, v. 15, n. 6, p. 665–671, 1996. ISSN 0278-0070.

BURD, T.; BRODERSEN, R. Design issues for Dynamic Voltage Scaling. In: **ISLPED'00: Proceedings of the 2000 International Symposium on Low Power Electronics and Design (Cat. No.00TH8514)**. [S.l.: s.n.], 2000. ISBN 1-58113-190-9.

BURD, T. D. et al. Dynamic voltage scaled microprocessor system. **IEEE Journal of Solid-State Circuits**, v. 35, n. 11, p. 1571–1580, 2000. ISSN 00189200.

CHANDRAKASAN, A. P.; SHENG, S.; BRODERSEN, R. W. Low-power CMOS digital design. **IEEE Journal of Solid-State Circuits**, v. 27, n. 4, p. 473–484, 1992. ISSN 00189200.

CHAVAN, A.; MACDONALD, E. Ultra low voltage level shifters to interface sub and super threshold reconfigurable logic cells. In: **IEEE Aerospace Conference Proceedings**. [S.l.]: IEEE, 2008. ISBN 1424414881. ISSN 1095323X.

CHINNERY, D.; KEUTZER, K. **Closing the gap between ASIC & custom: tools and techniques for high-performance ASIC design**. [S.l.]: Springer Science & Business Media, 2007. ISBN 9780387257631.

DHAR, S.; MAKSIRNOVI, D.; KRANZEN, B. Closed-loop adaptive voltage scaling controller for standard-cell ASICs. In: **Proceedings of the International Symposium on Low Power Electronics and Design**. [S.l.: s.n.], 2002. ISBN 1-5811-3475-4.

ELGEBALY, M.; SACHDEV, M. Variation-aware adaptive voltage scaling system. **IEEE Transactions on Very Large Scale Integration (VLSI) Systems**, v. 15, n. 5, p. 560–571, 2007. ISSN 10638210.

FLACH, G. et al. Simultaneous gate sizing and Vth assignment using Lagrangian Relaxation and delay sensitivities. In: **2013 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)**. [S.l.]: IEEE, 2013. p. 84–89. ISBN 978-1-4799-1331-2.

FLACH, G. et al. Effective Method for Simultaneous Gate Sizing and Vth Assignment Using Lagrangian Relaxation. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, v. 33, n. 4, p. 546–557, abr. 2014. ISSN 0278-0070.

GONZALEZ, R.; GORDON, B.; HOROWITZ, M. Supply and threshold voltage scaling for low power CMOS. **IEEE Journal of Solid-State Circuits**, v. 32, n. 8, p. 1210–1216, 1997. ISSN 0018-9200.

GUPTA, A. et al. A robust level-shifter design for adaptive voltage scaling. In: **Proceedings of the IEEE International Frequency Control Symposium and Exposition**. [S.l.: s.n.], 2008. p. 383–388. ISBN 0769530834. ISSN 1063-9667.

HAMADA, M. et al. A top-down low power design technique using clustered voltage scaling with variable supply-voltage scheme. In: **Proceedings of the IEEE 1998 Custom Integrated Circuits Conference (Cat. No.98CH36143)**. [S.l.: s.n.], 1998. p. 495–498. ISBN 0-7803-4292-5. ISSN 08865930.

HEDENSTIERNA, N.; JEPPSON, K. CMOS Circuit Speed and Buffer Optimization. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, v. 6, n. 2, 1987. ISSN 0278-0070.

HELLER, L. et al. Cascode voltage switch logic: A differential CMOS logic family. In: **1984 IEEE International Solid-State Circuits Conference. Digest of Technical Papers**. [S.l.]: IEEE, 1984. XXVII, n. 3, p. 3–4.

HOPPE, B. et al. Optimization of high-speed CMOS logic circuits with analytical models for signal delay, chip area, and dynamic power dissipation. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, v. 9, n. 3, p. 236–247, 1990. ISSN 02780070.

ITRS. International technology roadmap for semiconductors. **Available from: http://www.itrs.net/**, 2013.

KEATING, M. et al. **Low Power Methodology Manual: For System-on-Chip Design**. [S.l.]: Springer Publishing Company, Incorporated, 2007. 303 p. ISBN 9780387718187.

KESHAVARZI, A.; KAO, J. Techniques for Leakage Power Reduction. In: **Design of High-Performance Microprocessor Circuits**. [S.l.: s.n.], 2002. p. 329–329. ISBN 9780470544365.

KHAN, Q. A.; WADHWA, S. K.; MISRI, K. A single supply level shifter for multi-voltage systems. In: **Proceedings of the IEEE International Conference on VLSI Design**. [S.l.]: IEEE, 2006. v. 2006, p. 557–560. ISBN 0769525024. ISSN 10639667.

KO, U.; BALSARA, P. T. Short-circuit power driven gate sizing technique for reducing power dissipation. **IEEE Transactions on Very Large Scale Integration (VLSI) Systems**, v. 3, n. 3, p. 450–455, 1995. ISSN 10638210.

KOBAYASHI, T. et al. A current-controlled latch sense amplifier and a static power-saving input buffer for low-power architecture. **IEICE transactions on electronics**, The Institute of Electronics, Information and Communication Engineers, v. 76, n. 5, p. 863–867, 1993.

KOO, K. H. et al. A new level-up shifter for high speed and wide range interface in ultra deep sub-micron. In: **Proceedings - IEEE International Symposium on Circuits and Systems**. [S.l.: s.n.], 2005. p. 1063–1065. ISBN 0-7803-8834-8. ISSN 02714310.

KULKARNI, S.; SRIVASTAVA, A.; SYLVESTER, D. A New Algorithm for Improved VDD Assignment in Low Power Dual VDD Systems. In: **Proceedings of the 2004 International Symposium on Low Power Electronics and Design**. [S.l.]: IEEE, 2004. p. 200–205. ISBN 1-58113-929-2.

KULKARNI, S.; SYLVESTER, D. Fast and energy-efficient asynchronous level converters for multi-VDD design [CMOS ICs]. In: **IEEE International [Systems-on-Chip] SOC Conference, 2003. Proceedings.** [S.l.: s.n.], 2003. p. 169–172. ISBN 0-7803-8182-3.

LAVAGNO, L.; KEUTZER, K.; SANGICIVANNI-VINCENTELLI, A. Algorithms for synthesis of hazard-free asynchronous circuits. In: **28th ACM/IEEE Design Automation Conference**. [S.l.: s.n.], 1991. ISBN 0-89791-395-7. ISSN 01467123.

LI, H. et al. Combined circuit and architectural level variable supply-voltage scaling for low power. **IEEE Transactions on Very Large Scale Integration (VLSI) Systems**, v. 13, n. 5, p. 564–575, 2005. ISSN 10638210.

MERMET, J. P. **Low power design in deep submicron electronics**. [S.l.]: Springer Science & Business Media, 1997. 580 p. ISBN 9780792345695.

NARENDRA, S. G.; CHANDRAKASAN, A. P. **Leakage in nanometer CMOS technologies**. [S.l.]: Springer Science & Business Media, 2006.

OLUKOTUN, K. et al. The case for a single-chip multiprocessor. In: **Proceedings of the seventh international conference on Architectural support for programming languages and operating systems - ASPLOS-VII**. [S.l.: s.n.], 1996. p. 2–11. ISBN 0897917677.

OZDAL, M. M.; BURNS, S.; HU, J. Algorithms for Gate Sizing and Device Parameter Selection for High-Performance Designs. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, v. 31, n. 10, p. 1558–1571, oct. 2012. ISSN 0278-0070.

POUWELSE, J.; LANGENDOEN, K.; SIPS, H. Dynamic Voltage Scaling on a Low-Power Microprocessor. In: **Proceedings of the 7th Annual International Conference on Mobile Computing and Networking**. [S.l.: s.n.], 2001. p. 251–259. ISBN 1581134223.

RABAEY, J. **Low power design essentials**. [S.l.]: Springer Science & Business Media, 2009. 371 p. ISBN 9780387717128.

RABAEY, J. M.; CHANDRAKASAN, A.; NIKOLIC, B. **Digital integrated circuits: A design perspective**. 2nd. ed. [S.l.]: Prentice Hall, 2003. 761 p. ISBN 978-0130909961.

REIMANN, T.; SZE, C. C. N.; REIS, R. Gate sizing and threshold voltage assignment for high performance microprocessor designs. In: **The 20th Asia and South Pacific Design Automation Conference**. [S.l.]: IEEE, 2015. p. 214–219. ISBN 978-1-4799-7792-5.

ROCHA, J. F. da et al. Level shifters and DCVSL for a low-voltage CMOS 4.2-V buck converter. **IEEE Transactions on Industrial Electronics**, v. 55, n. 9, p. 3315–3323, 2008. ISSN 02780046.

ROY, K.; MUKHOPADHYAY, S.; MAHMOODI-MEIMAND, H. Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits. In: **Proceedings of the IEEE**. [S.l.]: IEEE, 2003. v. 91, n. 2, p. 305–327. ISBN 0018-9219. ISSN 00189219.

SAPATNEKAR, S. S. et al. Exact solution of the transistor sizing problem for CMOS circuits using convex optimization. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, v. 12, n. 11, p. 1621–1634, 1993. ISSN 02780070.

SCARTEZZINI, G.; REIS, R. Power consumption in transistor networks versus in standard cells. In: **2011 18th IEEE International Conference on Electronics, Circuits, and Systems**. [S.l.]: IEEE, 2011. p. 740–743. ISBN 978-1-4577-1846-5.

SHEN, W.-Z. S. W.-Z.; LIN, J.-Y. L. J.-Y.; WANG, F.-W. W. F.-W. Transistor reordering rules for power reduction in CMOS gates. In: **Proceedings of ASP-DAC'95/CHDL'95/VLSI'95 with EDA Technofair**. [S.l.: s.n.], 1995. ISBN 4-930813-67-0.

SHIN, D.; KIM, J.; LEE, S. Intra-task voltage scheduling for low-energy, hard real-time applications. **IEEE Design & Test of Computers**, IEEE Computer Society, Cosponsored by the IEEE Circuits and Systems Society, v. 18, n. 2, p. 20–30, 2001.

SIRICHOTIYAKUL, S. et al. Stand-by power minimization through simultaneous threshold voltage selection and circuit sizing. In: **Proceedings 1999 Design Automation Conference (Cat. No. 99CH36361)**. [S.l.: s.n.], 1999. ISBN 1-58113-092-9. ISSN 0738100X.

STOK, L. et al. Pushing ASIC performance in a power envelope. **Closing the Power Gap Between ASIC and Custom: Tools and Techniques for Low Power Design**, p. 323–356, 2007. ISSN 0738100X.

SUTHERLAND, I. E.; SPROULL, R. F.; HARRIS, D. F. **Logical effort: designing fast CMOS circuits**. [S.l.]: Morgan Kaufmann, 1999.

TAN, S.; SUN, X. Low power CMOS level shifters by bootstrapping technique. **Electronics Letters**, v. 38, n. 16, p. 876, 2002. ISSN 00135194.

TAUR, Y. CMOS design near the limit of scaling. **IBM Journal of Research and Development**, v. 46, n. 2.3, p. 213–222, 2002. ISSN 0018-8646.

TERRES, M. et al. A novel approach to reduce power consumption in level shifter for Multiple Dynamic Supply Voltage. In: **Proceedings of the IEEE International Conference on Electronics, Circuits, and Systems**. [S.l.: s.n.], 2013. p. 715–718. ISBN 9781479924523.

TERRES, M. et al. Exploring more efficient architectures for Multiple Dynamic Supply Voltage designs. In: **2014 IEEE 5th Latin American Symposium on Circuits and Systems, LASCAS 2014 - Conference Proceedings**. [S.l.]: IEEE, 2014. p. 2–5. ISBN 9781479925070.

THOMPSON, S. et al. An enhanced 130 nm generation logic technology featuring 60 nm transistors optimized for high performance and low power at 0.7 - 1.4 V. In: **International Electron Devices Meeting. Technical Digest (Cat. No.01CH37224)**. [S.l.: s.n.], 2001. ISBN 0-7803-7050-3. ISSN 01631918.

TRAN, C.; KAWAGUCHI, H.; SAKURAI, T. Low-power high-speed level shifter design for block-level dynamic voltage scaling environment. In: **2005 International Conference on Integrated Circuit Design and Technology, 2005. ICICDT 2005.** [S.l.: s.n.], 2005. v. 2, p. 6–9. ISBN 0-7803-9081-4.

USAMI, K.; HOROWITZ, M. Clustered voltage scaling technique for low-power design. In: **ACM/IEEE International Symposium on Low Power Design**. [S.l.: s.n.], 1995. p. 3–8. ISBN 0897917448. ISSN 10844309.

USAMI, K. et al. Automated low-power technique exploiting multiple supply voltages applied to a media processor. **IEEE Journal of Solid-State Circuits**, v. 33, n. 3, p. 463–471, 1998. ISSN 00189200.

VEENDRICK, H. **Short-circuit dissipation of static CMOS circuitry and its impact on the design of buffer circuits**. 1984. 468–473 p.

VIERI, C. et al. SOIAS: dynamically variable threshold SOI with active substrate. In: **1995 IEEE Symposium on Low Power Electronics. Digest of Technical Papers**. [S.l.: s.n.], 1995. ISBN 0-7803-3036-6.

WU, Q. W. Q.; PEDRAM, M.; WU, X. W. X. Clock-gating and its application to low power design of sequential circuits. **IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications**, v. 47, n. 3, 2000. ISSN 1057-7122.

YAO, F.; DEMERS, A.; SHENKER, S. A scheduling model for reduced cpu energy. In: IEEE. **Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on**. [S.l.], 1995. p. 374–382.

YUAN, J.; DI, J. D. J. Teaching low-power electronic design in electrical and computer engineering. **IEEE Transactions on Education**, v. 48, n. 1, p. 169–182, 2005. ISSN 0018-9359.

ZHAI, B. Z. B. et al. Theoretical and practical limits of dynamic voltage scaling. In: **Proceedings. 41st Design Automation Conference, 2004.** [S.l.: s.n.], 2004. ISBN 1-51183-828-8. ISSN 0738-100X.

APPENDIX A — TABLES

Table A.1 – Power dissipation of DCVS, Puri, and EF-LS level shifters for first simulation scenario

Amplitude	**Power Dissipation (uW)**		
	DCVS	**Puri**	**EF-LS**
35%	-	-	0.7037
40%	-	0.576	0.391
45%	-	0.430	0.344
50%	5.260	0.395	0.326
55%	1.630	0.379	0.318
60%	1.022	0.370	0.312
65%	0.839	0.364	0.308
70%	0.753	0.358	0.305
75%	0.707	0.353	0.301
80%	0.680	0.349	0.299
85%	0.666	0.346	0.296
90%	0.657	0.342	0.294
95%	0.654	0.340	0.292

No fan-in; Rise-time and fall-time of input signal = $0.03ns$; Amplitude is in percentages of the nominal voltage supply (1.5V).

Source: Author

Table A.2 – Time delays of DCVS, Puri, and EF-LS level shifters for first simulation scenario

Amplitude	Rise Time Delay (ps)			Fall Time Delay (ps)			Mean Time (ps)		
	DCVS	**Puri**	**EF-LS**	**DCVS**	**Puri**	**EF-LS**	**DCVS**	**Puri**	**EF-LS**
35%	-	-	1480.0	-		441.0	-	-	960.5
40%	-	444.0	321.0	-	310.0	333.0	-	377.0	327.0
45%	-	205.0	179.0	-	281.0	298.0	-	243.0	238.5
50%	1510.0	126.0	106.0	1740.0	268.0	283.0	1625.0	197.0	194.5
55%	327.0	95.0	77.5	429.0	260.0	274.0	378.0	177.5	175.8
60%	172.0	77.8	61.0	222.0	255.0	269.0	197.0	166.4	165.0
65%	122.0	67.5	51.3	142.0	250.0	264.0	132.0	158.8	157.7
70%	97.0	60.7	44.5	107.0	246.0	260.0	102.0	153.4	152.3
75%	82.8	56.0	39.7	86.1	241.0	256.0	84.5	148.5	147.9
80%	73.5	52.5	36.2	72.1	236.0	251.0	72.8	144.3	143.6
85%	67.2	49.8	33.6	62.0	231.0	247.0	64.6	140.4	140.3
90%	62.5	47.8	31.6	54.4	225.0	241.0	58.5	136.4	136.3
95%	59.0	46.1	30.0	48.5	216.0	233.0	53.8	131.1	131.5

No fan-in; Rise-time and fall-time of input signal = $0.03ns$; Amplitude is in percentages of the nominal voltage supply (1.5V).

Source: Author

Table A.3 – Comparison chart of PDP for DCVS, Puri, and EF-LS level shifters in the first simulation scenario

Amplitude	PDP Rise (pJ)			PDP Fall (pJ)		
	DCVS	**Puri**	**EF-LS**	**DCVS**	**Puri**	**EF-LS**
35%	-	-	1.090E-02	-	-	3.249E-03
40%	-	2.557E-03	1.256E-03	-	1.786E-03	1.303E-03
45%	-	8.807E-04	6.165E-04	-	1.207E-03	1.026E-03
50%	7.943E-02	4.974E-04	3.460E-04	9.152E-02	1.058E-03	9.237E-04
55%	5.330E-03	3.602E-04	2.465E-04	6.993E-03	9.859E-04	8.713E-04
60%	1.758E-03	2.875E-04	1.903E-04	2.269E-03	9.425E-04	8.393E-04
65%	1.023E-03	2.454E-04	1.582E-04	1.191E-03	9.090E-04	8.142E-04
70%	7.308E-04	2.171E-04	1.356E-04	8.061E-04	8.797E-04	7.925E-04
75%	5.852E-04	1.979E-04	1.196E-04	6.086E-04	8.517E-04	7.711E-04
80%	5.001E-04	1.833E-04	1.082E-04	4.906E-04	8.241E-04	7.500E-04
85%	4.474E-04	1.721E-04	9.959E-05	4.127E-04	7.983E-04	7.321E-04
90%	4.108E-04	1.635E-04	9.290E-05	3.576E-04	7.695E-04	7.085E-04
95%	3.857E-04	1.566E-04	8.748E-05	3.170E-04	7.335E-04	6.794E-04

No fan-in; Rise-time and fall-time of input signal = 0.03ns; Amplitude is in percentages of the nominal voltage supply (1.5V).

Source: Author

Table A.4 – Mean PDP of DCVS, Puri, and EF-LS level shifters for first simulation scenario

Amplitude	**PDP Mean (pJ)**		
	DCVS	**Puri**	**EF-LS**
35%	-	-	7.077E-03
40%	-	2.172E-03	1.279E-03
45%	-	1.044E-03	8.214E-04
50%	8.548E-02	7.778E-04	6.348E-04
55%	6.161E-03	6.731E-04	5.589E-04
60%	2.013E-03	6.150E-04	5.148E-04
65%	1.107E-03	5.772E-04	4.862E-04
70%	7.685E-04	5.484E-04	4.641E-04
75%	5.969E-04	5.248E-04	4.453E-04
80%	4.953E-04	5.037E-04	4.291E-04
85%	4.300E-04	4.852E-04	4.158E-04
90%	3.842E-04	4.665E-04	4.007E-04
95%	3.514E-04	4.450E-04	3.835E-04

No fan-in; Rise-time and fall-time of input signal = 0.03ns; Amplitude is in percentages of the nominal voltage supply (1.5V).

Source: Author

Table A.5 – Power dissipation of DCVS, Puri, and EF-LS level shifters for second simulation scenario

Amplitude	**Power Dissipation (uW)**		
	DCVS	**Puri**	**EF-LS**
35%	-	-	1.2935
40%	-	0.7020	0.7422
45%	-	0.6880	0.7083
50%	6.1091	0.6102	0.6381
55%	2.3550	0.6022	0.6136
60%	1.7905	0.5910	0.5875
65%	1.4032	0.5861	0.5792
70%	1.2873	0.4313	0.4461
75%	1.0382	0.3666	0.3492
80%	0.9371	0.3582	0.3474
85%	0.8015	0.3512	0.3417
90%	0.7436	0.3485	0.3383
95%	0.6850	0.3436	0.3365

With NAND2 fan-in; Rise-time and fall-time of input signal = 0.03ns; Amplitude is in percentages of the nominal voltage supply (1.5V).

Source: Author

Table A.6 – Time delays of DCVS, Puri, and EF-LS level shifters for second simulation scenario

Amplitude	Rise Time Delay (ps)			Fall Time Delay (ps)			Mean Time (ps)		
	DCVS	**Puri**	**EF-LS**	**DCVS**	**Puri**	**EF-LS**	**DCVS**	**Puri**	**EF-LS**
35%	-	-	1597.2	-		602.0	-	-	1099.6
40%	-	476.0	422.5	-	396.0	427.2	-	436.0	424.9
45%	-	282.3	215.6	-	286.0	331.5	-	284.2	273.6
50%	1680.0	140.5	112.8	1803.0	279.0	298.0	1741.5	209.8	205.4
55%	359.2	107.2	80.4	473.2	266.2	282.0	416.2	186.7	181.2
60%	203.0	88.0	72.0	231.0	257.8	272.3	217.0	172.9	172.2
65%	127.0	79.4	65.0	151.6	251.1	267.0	139.3	165.3	166.0
70%	103.0	62.3	47.8	107.0	247.4	265.9	105.0	154.9	156.9
75%	85.2	57.1	42.5	88.1	242.3	260.0	86.7	149.7	151.3
80%	74.6	53.8	37.0	73.3	240.0	257.0	74.0	146.9	147.0
85%	68.9	51.0	35.2	67.5	237.0	249.0	68.2	144.0	142.1
90%	63.2	48.6	33.2	56.2	230.0	245.2	59.7	139.3	139.2
95%	60.5	47.2	32.7	49.1	226.0	237.0	54.8	136.6	134.9

With NAND2 fan-in; Rise-time and fall-time of input signal = $0.03ns$; Amplitude is in percentages of the nominal voltage supply (1.5V).

Source: Author

Table A.7 – Comparison chart of PDP for DCVS, Puri, and EF-LS level shifters in the second simulation scenario

Amplitude	PDP Rise (pJ)			PDP Fall (pJ)		
	DCVS	**Puri**	**EF-LS**	**DCVS**	**Puri**	**EF-LS**
35%	-	-	2.065E-02	-	-	3.249E-03
40%	-	3.342E-03	3.135E-03	-	2.780E-03	1.303E-03
45%	-	1.942E-03	1.526E-03	-	1.968E-03	1.026E-03
50%	1.026E-01	8.571E-04	7.197E-04	1.101E-01	1.702E-03	9.237E-04
55%	8.459E-03	6.453E-04	4.929E-04	1.114E-02	1.603E-03	8.713E-04
60%	3.634E-03	5.201E-04	4.226E-04	4.135E-03	1.524E-03	8.393E-04
65%	1.782E-03	4.653E-04	3.764E-04	2.127E-03	1.471E-03	8.142E-04
70%	1.326E-03	2.685E-04	2.132E-04	1.377E-03	1.066E-03	1.186E-03
75%	8.844E-04	2.090E-04	1.483E-04	9.145E-04	8.868E-04	9.074E-04
80%	6.990E-04	1.926E-04	1.284E-04	6.868E-04	8.592E-04	8.918E-04
85%	5.519E-04	1.790E-04	1.200E-04	5.407E-04	8.319E-04	8.491E-04
90%	4.696E-04	1.691E-04	1.122E-04	4.176E-04	8.004E-04	8.288E-04
95%	4.144E-04	1.619E-04	1.099E-04	3.363E-04	7.752E-04	7.963E-04

With NAND2 fan-in; Rise-time and fall-time of input signal = 0.03ns; Amplitude is in percentages of the nominal voltage supply (1.5V).

Source: Author

Table A.8 – Mean PDP of DCVS, Puri, and EF-LS level shifters for second simulation scenario

Amplitude	**PDP Mean (pJ)**		
	DCVS	**Puri**	**EF-LS**
35%	-	-	1.422E-02
40%	-	3.061E-03	3.152E-03
45%	-	1.955E-03	1.937E-03
50%	1.064E-01	1.279E-03	1.310E-03
55%	9.802E-03	1.124E-03	1.111E-03
60%	3.884E-03	1.022E-03	1.011E-03
65%	1.954E-03	9.684E-04	9.611E-04
70%	1.351E-03	6.674E-04	6.996E-04
75%	8.994E-04	5.479E-04	5.279E-04
80%	6.929E-04	5.259E-04	5.101E-04
85%	5.463E-04	5.054E-04	4.846E-04
90%	4.436E-04	4.848E-04	4.705E-04
95%	3.754E-04	4.685E-04	4.531E-04

With NAND2 fan-in; Rise-time and fall-time of input signal = 0.03ns; Amplitude is in percentages of the nominal voltage supply (1.5V).

Source: Author

Table A.9 – Power dissipation of DCVS, Puri, and EF-LS level shifters for second simulation scenario

Amplitude	**Power Dissipation (uW)**		
	DCVS	**Puri**	**EF-LS**
35%	-	-	1.410
40%	-	0.810	0.778
45%	-	0.699	0.727
50%	6.198	0.630	0.666
55%	2.360	0.617	0.623
60%	1.802	0.605	0.597
65%	1.423	0.598	0.581
70%	1.290	0.510	0.498
75%	1.036	0.394	0.378
80%	0.953	0.368	0.363
85%	0.823	0.360	0.351
90%	0.765	0.354	0.342
95%	0.702	0.351	0.338

With NAND2 fan-in; Rise-time and fall-time of input signal = $0.3ns$; Amplitude is in percentages of the nominal voltage supply (1.5V).

Source: Author

Table A.10 – Time delays for DCVS, Puri, and EF-LS level shifters for second simulation scenario

Amplitude	Rise Time Delay (ps)			Fall Time Delay (ps)			Mean Time (ps)		
	DCVS	**Puri**	**EF-LS**	**DCVS**	**Puri**	**EF-LS**	**DCVS**	**Puri**	**EF-LS**
35%	-	-	1816.0	-	-	813.0	-	-	1314.5
40%	-	523.0	504.0	-	413.5	510.5	-	468.25	507.25
45%	-	297.0	220.0	-	301.5	341.0	-	299.25	280.5
50%	1920.0	168.0	140.3	1895.0	290.0	302.0	1907.5	229.0	221.15
55%	502.0	135.3	109.0	480.0	275.0	290.0	491.0	205.15	199.5
60%	245.1	96.5	90.2	241.0	267.5	284.1	243.05	182.0	187.15
65%	137.0	86.2	79.6	165.0	258.0	278.0	151.0	172.1	178.8
70%	120.0	73.0	57.0	113.0	253.1	270.0	116.5	163.05	163.5
75%	89.5	60.0	50.5	92.5	247.0	264.5	91.0	153.5	157.5
80%	79.2	57.1	46.4	76.0	243.0	260.0	77.6	150.05	153.2
85%	72.4	55.5	40.1	69.1	240.5	251.4	70.75	148.0	145.75
90%	67.0	50.6	37.5	60.5	237.6	248.0	63.75	144.1	142.75
95%	62.0	49.0	35.0	52.0	230.0	241.0	57.0	139.5	138.0

With NAND2 fan-in; Rise-time and fall-time of input signal = $0.3ns$; Amplitude is in percentages of the nominal voltage supply (1.5V).

Source: Author

Table A.11 – Comparison chart of PDP for DCVS, Puri, and EF-LS level shifters in the second simulation scenario

Amplitude	PDP Rise (pJ)			PDP Fall (pJ)		
	DCVS	**Puri**	**EF-LS**	**DCVS**	**Puri**	**EF-LS**
35%	-	-	2.561E-02	-		1.146E-02
40%	-	4.236E-03	3.921E-03	-	3.349E-03	3.972E-03
45%	-	2.076E-03	1.599E-03	-	2.107E-03	2.479E-03
50%	1.190E-01	1.058E-03	9.344E-04	1.175E-01	1.827E-03	2.011E-03
55%	1.185E-02	8.348E-04	6.791E-04	1.133E-02	1.697E-03	1.807E-03
60%	4.417E-03	5.838E-04	5.385E-04	4.343E-03	1.618E-03	1.696E-03
65%	1.950E-03	5.155E-04	4.625E-04	2.348E-03	1.543E-03	1.615E-03
70%	1.548E-03	3.723E-04	2.839E-04	1.458E-03	1.291E-03	1.345E-03
75%	9.272E-04	2.364E-04	1.909E-04	9.583E-04	9.732E-04	9.998E-04
80%	7.548E-04	2.101E-04	1.684E-04	7.243E-04	8.942E-04	9.438E-04
85%	5.959E-04	1.998E-04	1.408E-04	5.687E-04	8.658E-04	8.824E-04
90%	5.126E-04	1.791E-04	1.283E-04	4.628E-04	8.411E-04	8.482E-04
95%	4.352E-04	1.720E-04	1.183E-04	3.650E-04	8.073E-04	8.146E-04

With NAND2 fan-in; Rise-time and fall-time of input signal = $0.3ns$; Amplitude is in percentages of the nominal voltage supply (1.5V).

Source: Author

Table A.12 – Mean PDP of DCVS, Puri, and EF-LS level shifters for second simulation scenario

Amplitude	**PDP Mean (pJ)**		
	DCVS	**Puri**	**EF-LS**
35%	-	-	1.853E-02
40%	-	3.793E-03	3.946E-03
45%	-	2.092E-03	2.039E-03
50%	1.182E-01	1.443E-03	1.473E-03
55%	1.159E-02	1.266E-03	1.243E-03
60%	4.380E-03	1.101E-03	1.117E-03
65%	2.149E-03	1.029E-03	1.039E-03
70%	1.503E-03	8.316E-04	8.142E-04
75%	9.428E-04	6.048E-04	5.954E-04
80%	7.395E-04	5.522E-04	5.561E-04
85%	5.823E-04	5.328E-04	5.116E-04
90%	4.877E-04	5.101E-04	4.882E-04
95%	4.001E-04	4.896E-04	4.664E-04

With NAND2 fan-in; Rise-time and fall-time of input signal = $0.3ns$; Amplitude is in percentages of the nominal voltage supply (1.5V).

Source: Author

APPENDIX B — PUBLICATIONS

- **SIM 2013** - *Global Routing and Parallelism*. Roger C. Llanos, Diego Tumelero, Marcelo Johann and Ricardo Reis.
- **Iberchip 2014** - *Design Automation for Digital Microfluidic Biochips*. Roger C. Llanos, Guilherme Bontorin, Marcelo de Oliveira Johann and Ricardo Augusto da Luz Reis.
- **PATMOS 2015** - *Energy-Efficient Level Shifter Topology*. Roger C. Llanos, Diego Sousa, Marco Terres, Guilherme Bontorin, Ricardo Reis, and Marcelo Johann.