Faculteit Industriële Ingenieurswetenschappen

master in de industriële wetenschappen: elektronica-ICT

Masterthesis

Low-power Dynamic logic circuits for flexible electronics

Edmond Tsampanis

Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: elektronica-ICT

PROMOTOR :
Prof. dr. ing. Kris MYNY

PROMOTOR :
ir. Mohit DANDEKAR

BEGELEIDER :
ing. Jelle BIESMANS

Gezamenlijke opleiding UHasselt en KU Leuven

UHASSELT KU LEUVEN

Universiteit Hasselt | Campus Diepenbeek | Faculteit Industriële Ingenieurswetenschappen | Agoralaan Gebouw H - Gebouw B | BE 3590 Diepenbeek

Universiteit Hasselt | Campus Diepenbeek | Agoralaan Gebouw D | BE 3590 Diepenbeek
Universiteit Hasselt | Campus Hasselt | Martelarenlaan 42 | BE 3500 Hasselt

UHASSELT KU LEUVEN

2024
2025

Faculteit Industriële Ingenieurswetenschappen

master in de industriële wetenschappen: elektronica-ICT

Masterthesis

Low-power Dynamic logic circuits for flexible electronics

Edmond Tsampanis

Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: elektronica-ICT

PROMOTOR :

Prof. dr. ing. Kris MYNY

PROMOTOR :

ir. Mohit DANDEKAR

BEGELEIDER :

ing. Jelle BIESMANS

UHASSELT KU LEUVEN

Foreword

This thesis explores the possibility of implementing efficient logic on a flexible chip composed of indium gallium zinc oxide (IGZO). Specifically, it investigates the feasibility of using dynamic logic in this context, with potential implications for the development of next-generation biomedical sensors.

While the rapid pace of technological advancement may limit the long-term applicability of this specific implementation, the process of conducting this research has been highly educational and personally rewarding. It has deepened my understanding of both the technical and conceptual challenges involved in flexible electronics and the dynamic logic style.

I would like to express my heartfelt gratitude to Prof. Dr. Ing. Kris Myny for their continuous support and guidance which made this research possible. I am also grateful to my supervisor, Ing. Jelle Biesmans and the entire Emerging Technologies, Systems & Security (ES&S) research group for their invaluable mentorship throughout this journey.

Finally, I extend my sincere thanks to my close friends and family for their unwavering support during this research. Their encouragement and feedback have been invaluable in creating memories that I will always cherish.

Contents

List of Tables

List of Figures

Explanatory terms

Dynamic logic styles	Logic families that rely on clocked operation, where outputs are precharged and then conditionally evaluated based on inputs.
Static logic styles	Logic circuits that maintain their output state indefinitely without requiring a clock, as long as power is applied, and use pull-up and pull-down networks for stable logic levels.
Diode-load logic	A static unipolar logic style where diodes or transistors in diode configuration are used as pull-up or pull-down devices instead of active transistors.
Pseudo-CMOS logic	A static unipolar logic style that mimics CMOS-like behavior. It uses two stages internally, first there is a stage in diode-load logic to generate the input for the pull-up transistor of the second stage. The first stage is typically at a higher voltage than the second stage.
Resistive-load logic	A basic logic style where passive resistors act as load devices for pull-up or pul-down functionality.
Two-stage resistive-load logic (2RLL)	A logic approach using two cascaded stages, with the first stage consisting of resistive-load logic and the second stage the same as the second stage in pseudo-cmos.
Single-ended logic	A logic style in which signals are represented by a single output, as opposed to differential logic, in which the complementary signal are also computed.
Differential logic	A logic style where each logical signal is represented by two complementary signals.
Indium gallium zinc oxide (IGZO)	A transparent metal-oxide semiconductor material used in thin-film transistors (TFTs). IGZO is well-suited for large-area, flexible, or transparent electronics, though it is typically unipolar (n-type only).
Charge sharing	A phenomenon in dynamic logic where charge stored on an intermediate node is unintentionally redistributed to other capacitively coupled nodes.
Premature node discharge	An error condition in dynamic logic where an internal node discharges too early, often before correct evaluation has occurred, leading to incorrect logic output.
Bootstrapping	A technique used in digital circuits to temporarily raise the gate voltage of a transistor above the supply voltage using a capacitor, improving switching performance or ensuring full voltage swing.

Abstract

The development of logic gates on flexible electronics is an area that has yet to be thoroughly explored. Presently, multiple different logic styles are utilised for the purpose of creating logic, which all exhibit a high static power consumption, thus rendering them impractical for certain applications. This master's thesis explores the possibility of creating dynamic, power-efficient logic on a flexible substrate.

The initial approach adopted was the creation of a delay cell for cascading dynamic logic. The purpose of the delay cell is to recreate the rippling effect observed in dynamic logic by propagating the signal at specific intervals. A secondary solution that was given consideration was the use of NAND gates in differential dynamic logic. In this configuration, the NAND gate functions as the delay cell, permitting signal propagation after the completion of the logic cell's computation.

Using these proposed logic styles and two-stage resistive-load logic (2RLL), a 4-bit ripple carry adder was designed. On this a power of 18.6 µW for the single-ended logic and 22.8 µW for the differential logic was simulated. In contradistinction to 2RLL (435 µW), this denotes a decrease of approximately 19 times. This verifies the functionality and power-efficiency of the proposed solutions. However, it is imperative to acknowledge that these cells have a negative impact on the speed of the logic circuit. Consequently, it is imperative to engineer more efficient cells for further work to increase the speed of the logic circuits.

Abstract in Dutch

De ontwikkeling van logische poorten op flexibele elektronica is een gebied dat nog grondig moet worden onderzocht. Momenteel worden verschillende logische stijlen gebruikt om logica te maken, die allemaal een hoog statisch stroomverbruik hebben. Deze masterthesis onderzoekt de mogelijkheid om dynamische, energie-efficiënte logica te maken op flexibel substraat.

De eerste benadering was het maken van een vertragende cel voor het cascaden van dynamische logica. Het doel van deze cel is om het rimpelingseffect dat wordt waargenomen in dynamische logica na te bootsen door het signaal met specifieke intervallen door te geven. Een secundaire oplossing die werd overwogen was het gebruik van NAND-poorten in differentiële logica. In deze configuratie functioneert de NAND-poort als de vertragende cel, waardoor het signaal zich kan voortplanten nadat de logische cel zijn berekening voltooid is.

Met behulp van deze voorgestelde logische stijlen en twee-trap resistieve verbruiker logica (2RLL) werd een 4-bit ripple carry-adder ontworpen. Hierop werd een vermogen van 18.6 µW gesimuleerden voor de single-ended logica en 22.8 µW voor de differentiële logica. In tegenstelling tot 2RLL (435 µW) betekent dit een afname van ongeveer 19 keer. Dit verifieert de functionaliteit en energie-efficiëntie van de voorgestelde oplossingen. Het is echter noodzakelijk om te erkennen dat deze cellen een negatieve invloed hebben op de snelheid van het logische circuit. Daarom is het noodzakelijk om efficiëntere cellen te ontwikkelen voor verder werk.

Chapter 1

Introduction

1.1 Research Context

Flexible electronics offer several advantages in areas where traditional electronics are less effective. Typical advantages include relatively cost-effective manufacturing, simple fabrication process and high mechanical stress robustness. Because of these mechanical properties [1], thin-film transistors (TFTs) could potentially be used in the medical sector as wearable and flexible sensors that are comfortable and resistant to deformation. Flexible electronics also have applications outside the medical sector, including [2]:

- displays [3];
- radio-frequency identification [4];
- flexible microprocessors [5].

TFTs are developed in different technologies, including metal oxide and low-temperature polycrystalline silicon (LTPS), both widely used for the production of integrated circuits. In the case of indium gallium zinc oxide (IGZO), the metal oxide acts only as an N-type MOSFET [2].

This research revolves around verbalising the efficiency of TFTs and is taking place at Emerging Technologies, System & Security (ES&S) research group at KU Leuven in Diepenbeek. This research group focuses on PragmatIC's IGZO technology [6].

However, the limited level of development of TFTs for integrated circuits poses a challenge for the commercialisation of this technology as wearable technologies, among others [7]. At the present time, flexible circuits based on TFTs currently consume significantly more energy than their conventional counterparts when it comes to logic computation. An illustrative example may be the 6502 microcontroller, which consumes 1mW at a clock speed of 1MHz in CMOS technology [8]. In contrast, a flexible variant of this microcontroller in IGZO operating at a clock frequency of 10 kHz, with a power consumption of 11.6 mW [9].

1.2 Problem Statement

The aim of this thesis is to optimise the power consumption of TFTs, in particular IGZO-based technologies. One of the major challenges of most flexible technology, including IGZO, is the

lack of P-type MOSFETs (unipolar technologies). This renders the conventional application of traditional CMOS logic impractical. This poses a fundamental problem in the development of efficient IGZO-based TFT circuits.

A proposed solution for logic creation in IGZO, as described in [10], is to make use of dynamic logic styles instead of static logic styles. This makes it possible to eliminate a large number of P-type MOSFETs, while the remaining MOSFETs can be replaced or minimised. In conventional CMOS technologies, dynamic logic is applied to increase speed, however, in this case it will also contribute to efficiency by reducing leakage currents. The utilisation of a dynamic logic style is accompanied by inherent complexities, including charge sharing, glitches and stringent timing requirements, which must be considered [11].

In addition, it is essential to comply with standards and good practices within chip design. These technical constraints, several of which will be discussed in the objectives, place requirements on the use of IGZO-based technology and are often difficult to achieve in designs that maximise efficiency.

1.3 Objectives

The central objective of this thesis is to reduce overall energy consumption of logic cells in IGZO. In view of the plurality of potential approaches, it is imperative to impose boundary conditions on this objective. Firstly, the area of the dynamic circuits should be limited, with the size of the logic cell being no more than double that of their CMOS counterpart. However, an exception is applied to the inverter, due to its importance and significant impact on efficiency, which may permit its use to extend to more than double the area.

In order to evaluate the effectiveness of the implementations, a comparison will be drawn between the dynamic logic and two-stage resistor-load logic (2RLL) logic [12]. In both logic styles, a 4-bit adder will be created with flip-flops at its inputs and outputs. Subsequently, a comparison will be made between the power, area and speed in order to validate the proposed circuit. It is imperative to acknowledge that the central focus of this thesis is on low-power logic circuits. Consequently, it is permissible for the other two areas to exhibit suboptimal outcomes.

Furthermore, the operating frequency is a crucial factor that remains constant when comparing different circuits. Finally, it is imperative to practically validate the proposed solutions. Initially, a 4-bit adder will be constructed in a variety of logic styles, with a view to comparing them with each other in simulation. Should the newly proposed logic styles demonstrate a satisfactory degree of performance, they may be subjected to empirical testing and measurement.

1.4 Methodology

The enhancement of the efficiency of dynamic circuits utilising unipolar technologies (IGZO) can be addressed through a number of methodologies. Firstly, a comprehensive literature review will be conducted. The present study has two principal aims. Firstly, it serves to provide a contextual framework for the thesis. Secondly, it identifies innovative solutions, with a particular focus on techniques that reduce leakage currents or enable lower voltages. The emphasis of this study is twofold: firstly, the utilisation of dynamic electronics employing bipolar technologies,

and secondly, the application of fundamental circuits comprising exclusively N-type MOSFETs (unipolar).

The subsequent stage of the process involves the design and simulation of these solutions in Cadence Virtuoso. This programme facilitates the process of verifying the performance of the developed designs in accordance with the specified parameters. Following the evaluation of the designs, dynamic and static circuits will be developed. Firstly, the simulation of 4-bit adders will be conducted across all logic styles. Should these prove to be promising results, a tape-out can be created to measure them in real life.

1.5 Prospect

The present series of topics will be pursued through the medium of a literature study. The issue of dynamic logic in IGZO-based technologies will be elucidated in the literature study. Furthermore, the discussion will proceed to address the various dynamic logic styles that are frequently employed in CMOS.

In the chapter dedicated to methodology, the implementation of the dynamic logic style in IGZO-based technologies will be explained. The proposed logic is characterised by the utilisation of a cell that generates a constant delay. The methodology will also address the cell responsible for inducing the observed delay, in addition to its utilisation within the circuit.

In the results and discussion, a comparison will be made between the various logic styles based on power, area and speed. In addition, a more detailed examination will be conducted to elucidate the characteristics of the delay cell, its operational mechanisms, and the underlying rationale for its functioning. Finally, the conclusions drawn from this thesis will be summarised.

Chapter 2

Literature Study

2.1 Contextualization and Technological Background

The following section will provide a more comprehensive introduction to the indium-gallium-zinc-oxide (IGZO) technology, offering a contextual framework to understand the current state of affairs. Subsequently, a study will be conducted between two-stage resistive-load logic (2RLL) and dynamic logic, with the objective of demonstrating the latter's potential, and to provide a rationale for the necessity of this research.

2.1.1 Indium-Gallium-Zinc-Oxide Thin-Film Technology

A variety of thin-film transistor (TFT) technologies are in existence, each of which possesses distinct properties and applications. As illustrated in Table 2.1, a concise overview of diverse TFT technologies is presented, encompassing amorphous silicon (a-Si), low-temperature polycrystalline silicon (LTPS), indium gallium zinc oxide, and organic TFTs.

Table 2.1: Comparison of different thin-film transistor technologies [2, p. 32]

Parameter	a-Si	LTPS	Oxide	Organic
μ ($cm^2 V^{-1} s^{-1}$)	0.5–1	50–100	10–40	0.1–10
Process complexity	Low	High	Low	Low
Manufacturing cost	Low	High	Low	Low
Bias and light stability	Poor	Good	Fair	Poor
Intrinsic properties on mechanical stress, without stack optimization	Poor	Poor	Good	Fair-to-good
Semiconductor	n-type	CMOS	n-type	p-type (n-type possible)
TFT-to-TFT uniformity	Good	Low	Good	Low
Large-area uniformity	Good	Low	Good	Low
L-scaling (lateral device architecture)	–	Limited due to polycrystalline semiconductor (µm-range)	Deep-submicrometre demonstrated	Limited due to contact resistance and polycrystalline semiconductor (µm-range)
Backplane applications	Low-end and large-area display and imagers	High-end display and imagers, with on-panel circuitry	Low-to-high-end display and imager applications and large-area panels	Low-end backplane and circuit applications
Circuit applications	Low-end applications	High-end digital and analogue	Low-to-high-end digital, low-to-medium-end analogue	Low-end applications

To assess the performance of semiconductor technologies, four distinct criteria are typically considered: power consumption, operational speed, physical area, and manufacturing cost. Power consumption and speed are typically application-specific parameters, whereas area and cost are more closely tied to the underlying technology. Taking this into consideration, Table 2.1 shows that IGZO offers low process complexity and manufacturing cost, while preserving its transistor performance under mechanical stress compared to other technologies.

Presently, low cost flexible electronics predominantly consist of unipolar technologies. However, unipolar technologies exhibits certain drawbacks, including diminished robustness [13] and an increase in both area and power consumption when it comes to the development of logic circuits. A variety of topologies are currently used to achieve logic circuits in unipolar technologies, including resistive load [4], pseudo-CMOS [14] and the use back gates [15]. It is important to note that all of these topologies have their own disadvantages, which may include: high static power consumption, increased area and reduced computation speed. These disadvantages add extra processing steps which increases the cost [2].

For the purpose of comparison in this thesis, 2RLL logic has been selected as the baseline because of its proven effectiveness [12]. The proposed dynamic logic will have implementations on both single-ended and differential logic. The subsequent chapters of this introduction will provide an overview of the concept of 2RLL and the potential of dynamic logic styles to increase power efficiency.

2.1.2 Examples of Flexible Integrated Circuits

Flexible electronics have applications in a variety of purposes, despite them mainly being unipolar. The following section will provide a brief overview of a number of applications in which flexible electronics could be utilised. The predominant commercial activity involving IGZO at present is the production of RFID tags [16] and displays [17]. However, a substantial amount of research has been conducted on the various applications of this technology.

One such example is that of gas sensors. These components can be fabricated using IGZO-based technologies, which are cost-effective, reliable and stable at room temperature [18]. Gas sensors require highly sensitive circuitry, due to the small differences in conductivity that must be measured, which have been proven to be achievable with IGZO technologies [19]. Due to the straightforward fabrication process of IGZO and its inherent flexibility [2], the production and deployment of gas sensors can be carried out with relative ease and efficiency [20].

Another example is [9] which created an 8-bit microcontroller. More specifically, a 6502 microcontroller was created, having a power consumption of 11.6 mW at a clock frequency of 10 kHz. This demonstrates that logic computation and a CPU can be realised, albeit with some drawbacks. This could be useful in specific cases where logic circuits, such as those used for communication protocols, are required prior to transmitting the data.

The final examples discussed are that of biosensors. These can range from ECG patches with an NFC communication protocol [21] to sensors that detect bladder cancer [22]. As previously mentioned, IGZO can be beneficial here due to its low production cost. However, the flexibility of these circuits can also provide a more comfortable experience for the patient. This is because flexible electronics can be made breathable and withstand some mechanical stress [2], thus avoiding discomfort or damage to the human body [23].

2.1.3 Logic Styles in Flexible Electronics

In the domain of unipolar technologies, there exists a multitude of design philosophies by which logic can be constructed. A concise overview of these logic styles is provided in Table 2.2. These logic styles under consideration are all static logic styles. The assessment of these styles is partially based on the metrics that have been previously discussed in Chapter 2.1.1. These metrics include area, speed, power and design complexity. Table 2.2 is derived from [24], where various circuits were evaluated across different logic styles through a comparative analysis. Table 2.2 provides an thorough look at the inverters in this study.

Table 2.2: Comparison of logic styles based on key design metrics [24]

	Diode Load	**Peudo-CMOS**	**Resistive Load**	**Two-Stage Resistive-Load**
Area	Low	Moderate	Low	Moderate
Power	High	High	Moderate	Moderate
Speed	Moderate	High	Moderate	Low
Complexity	Low	Moderate	Low	Moderate

It is important to note that the characteristics presented in Table 2.2 are defined relative to the other logic styles presented within the same table. For example, the designation of low power consumption reflects a value that is low compared to the other logic styles listed, but may not represent an absolute low level suitable for all applications.

It is also noteworthy that the measurements derived from [24] are highly device dependent. In the field of digital logic design, the utilisation of minimal sizes for devices is a common practice. This is not the case for the leaking stage of unipolar logic styles. A compromise must be made based on the qualifications that are required for that circuit. One example of this could be power efficiency, where large components are designed to ensure low power consumption, but also result in a slower working device. Furthermore, it should be noted that not all compromises made during the process of determining a particular logic style are reflected in Table 2.2. An example of this could be the suboptimal voltage swing that is present in diode-load or resistive load logic, which is solved in two-stage logic styles by typically having an extra voltage rail that is higher than the voltage supply. This is also the rationale behind the utilisation of subjective analysis for the characterisation of logic styles which should be interpreted with a degree of caution, as opposed to the use of empirical measurements.

2.1.4 Two-Stage Resistive-Load Logic

In unipolar technologies, a 2RLL logic configuration has been shown to facilitate efficient logic creation [12]. This approach is similar to that of pseudo-CMOS [14] which employs two stages to power the logic itself. An illustration of a 2RLL inverter can be observed in 2.1.

The initial stage of Figure 2.1 contains a leaking inverter, due to its pull-up network consisting of a resistor which is used to regulate the current passing through this stage. The pull-down network of the first stage is constituted by an N-MOSFET transistor, which possesses the capacity to pull the internal node down to the ground. This creates a controlled short with minimal power consumption, as the pull-up resistor restricts the current flow. With this primary stage, the inverse of the input can be computed. The resulting output serves as the input to the pull-up network of the next stage. Since N-MOSFET transistors are used in the pull-up path, an inverted

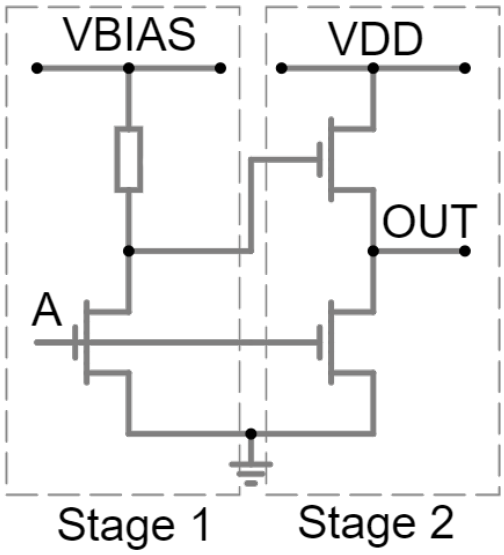

Figure 2.1: Two-stage resistive-load inverter

input is required to ensure proper signal computation within the logic gate. In addition to this, a voltage that exceeds the sum of the transistor's threshold voltage and supply voltage is required at the input of the pull-up transistor to ensure a complete voltage swing. This is accomplished by a second supply voltage, VBIAS, which is higher than that used for the logic computation. This mimics the behaviour of static CMOS logic, comprising exclusively of N-MOSFETs that form pull-up and pull-down networks. This configuration is illustrated in the second stage of the inverter.

However, it should be noted that this logic style is not without its drawbacks. Firstly, it should be acknowledged that two distinct voltage supplies are required for this logic style to function with a complete voltage swing, which can be suboptimal in terms of implementation. A secondary disadvantage is characterised by the initial stage of the logic cells, which is inherently prone to leakage. This is mitigated to a certain extent by the resistor, yet it still contributes a significant amount to the power consumption of the logic style. It is important to acknowledge that these limitations are not confined to this particular logic style. The leaking stage is applicable to all, while the addition of an extra voltage supply is only relevant to the two-stage logic styles in Table 2.2.

2.1.5 Potential of Dynamic Logic

A proposal for reducing the power of logic circuits is by implementation of a dynamic logic style. The hypothesis under consideration was proposed by [10] in the context of dynamic TFTs, wherein a comparative analysis was conducted among various inverters in different logic styles to determine the most efficient one. Table 2.3 provides an overview of the results obtained by [10].

As demonstrated in Table 2.3, utilising dynamic logic in unipolar technologies can yield substantial power efficiency gains. This is accompanied by a substantial reduction in area used, without any significant compromise to the voltage swing of the output or computation speed [10]. The findings indicate that dynamic logic has the potential to serve as a solution for low-power logic circuits in unipolar technologies. However, the implementation of dynamic logic in this technology is hindered by challenges, such as charge sharing and logic glitches.

Table 2.3: Performance comparison of different logic style inverters [10]

	DLL	P-CMOS	Differential	Dynamic
V_{OH} (V)	9.334	9.996	9.535	9.534
V_{OL} (V)	0.117	0.004	0.083	0.091
Power (µW)	261.0	248.3	0.79	0.17
T_{DLH} (µs)	1.26	2.05	1.22	1.12
T_{DLH} (µs)	0.11	0.64	3.31	1.40
Area (µm²)	30,800	17,500	11,200	5,600

2.2 Instability in Dynamic Logic

In this chapter, a range of challenges pertaining to dynamic logic shall be discussed. It is important to note that these are not technology-specific, as these are rather common problems that must be taken into consideration when creating dynamic logic. Some solutions to these problems will be discussed, although it should be noted that these solutions are used in conventional CMOS and may not directly translate to IGZO technologies.

2.2.1 Conceptual Framework of Dynamic Logic

The concept of dynamic logic is based on the utilisation of either the pull-up or pull-down network of static logic, as only either is required for logical computation, thereby creating faster logic. It is evident that computation is only possible in one direction, contingent on whether a pull-up or pull-down network is incorporated. Figure 2.2 depicts the schematic of a dynamic NAND gate comprising a pull-down network and a NOR gate utilising a pull-up network.

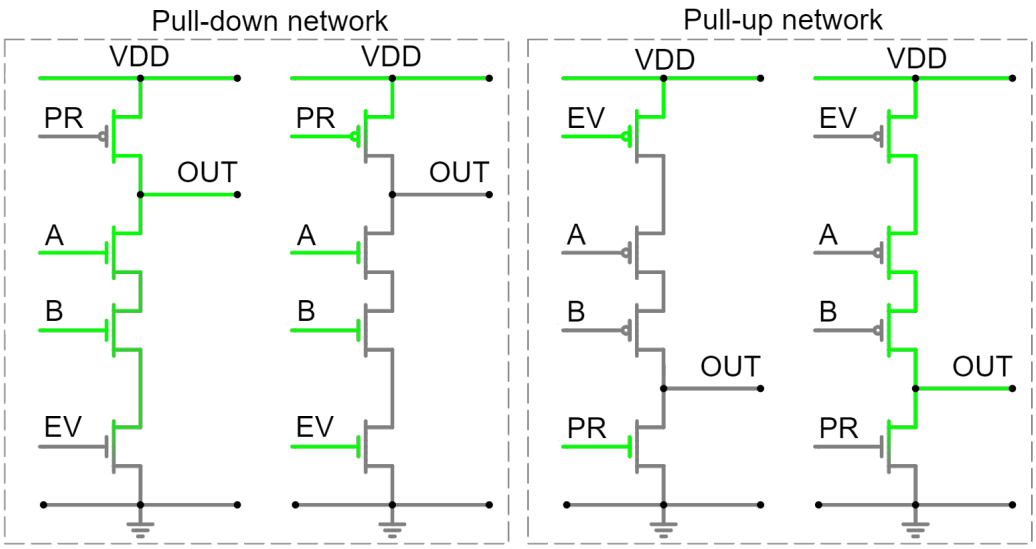

Figure 2.2: Dynamic pull-down (NAND) and pull-up (NOR) network precharging and evaluating

While the calculation of logic is impossible by only being able to pull a node up or down, dynamic logic relies on clocking the circuit, resulting in two operational phases: precharge and evaluate. In the precharge phase, the logic circuit charges a node towards ground or the supply voltage, depending on whether a pull-up or a pull-down network has been utilised. In the course of the evaluation phase, the logic inputs are capable of determining whether or not the output requires

to be raised or lowered. However, it is important to note that once the logic output changes state, it is not possible to regain its original state as the precharge phase is needed to obtain this [11], [25].

This idea behind the precharge and evaluate phase can be observed in Figure 2.2 where the initial circuit is present within either the pull-up or pull-down network during the precharge phase. Here, the output node gets charged to ground or the supply voltage. The second circuit is the evaluation phase, during which the logic is computed, potentially changing the output state.

2.2.2 Charge Sharing

A frequent issue encountered when dynamic logic styles are implemented relates to charge sharing. As depicted in the initial circuit of Figure 2.3, the output is precharged to the supply voltage. In the event of the evaluate phase being initiated in the second schematic, a considerable decline in voltage is observed (as indicated by the darker green output). The phenomenon is referred to as charge sharing.

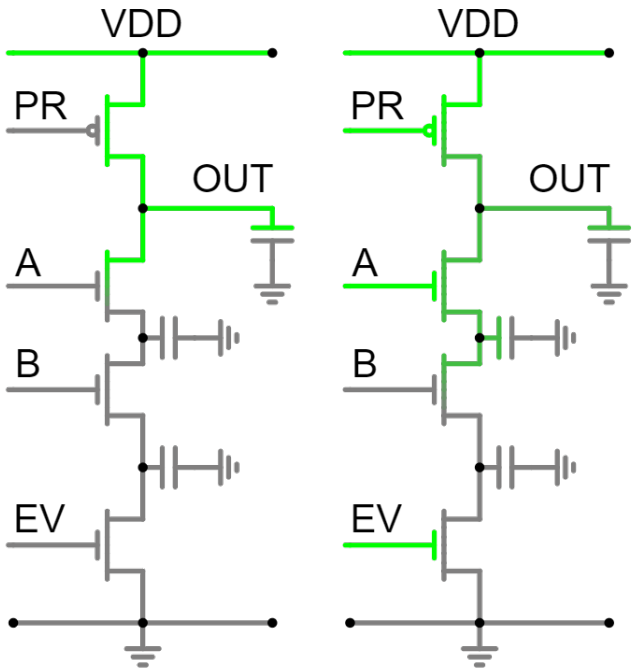

Figure 2.3: Lowered output voltage due to charge sharing

The concept of charge sharing can be attributed to the parasitic capacitance of each node in a transistor. In the process of precharging an output node to the supply voltage, it is important to consider that the output node accumulates charge to a certain extent. In the event of the evaluation phase, there is a possibility that closed transistors will open up as a result of the preceding logic. This may result in a previously discharged node becoming connected with the output node, thereby lowering the voltage of the output node (indicated by the darker green) by sharing its charge with the depleted node (Figure 2.3). This concept may also be applied to the gate of the transistors. In circumstances where an output node is is required to drive multiple gates, a larger parasitic capacitance must be charged, resulting in a reduced output voltage, as in the previous example [11], [26].

There are a number of solutions that can be utilised to address the charge sharing issue. One such

method involves the utilisation of a bleeder or keeper transistor. This is a P-MOSFET, which connects both the output and supply voltage, and is driven by the inverted output [27], [28]. An alternative solution to this issue would be to implement a precharge transistor for each individual node in order to ensure that charge is not lost upon the opening of a transistor [29].

2.2.3 Premature Node Discharge

One of the most straightforward methodologies for establishing dynamic logic is to arrange logic blocks in a cascading configuration. Nevertheless, this does not result in the creation of working logic; rather, it introduces glitches. Premature node discharge has been shown to be a contributing factor to these glitches.

The issue of premature node discharge is illustrated in Figure 2.4. In both scenarios, whether the logical input A is high or low, the output remains low. The rationale behind this phenomenon can be explained by the fact that the preceding logic stage discharges the subsequent stage before its own logic state can be computed. In more concrete terms, during the evaluation phase, the precharged output is charged to a state that enables the subsequent stage's transistor. This causes the precharged value to be lost when the evaluation phase occurs. A variety of solutions for this issue have been devised within CMOS, which are typically designated as distinct dynamic logic styles. The following subchapter will address these in greater detail [11].

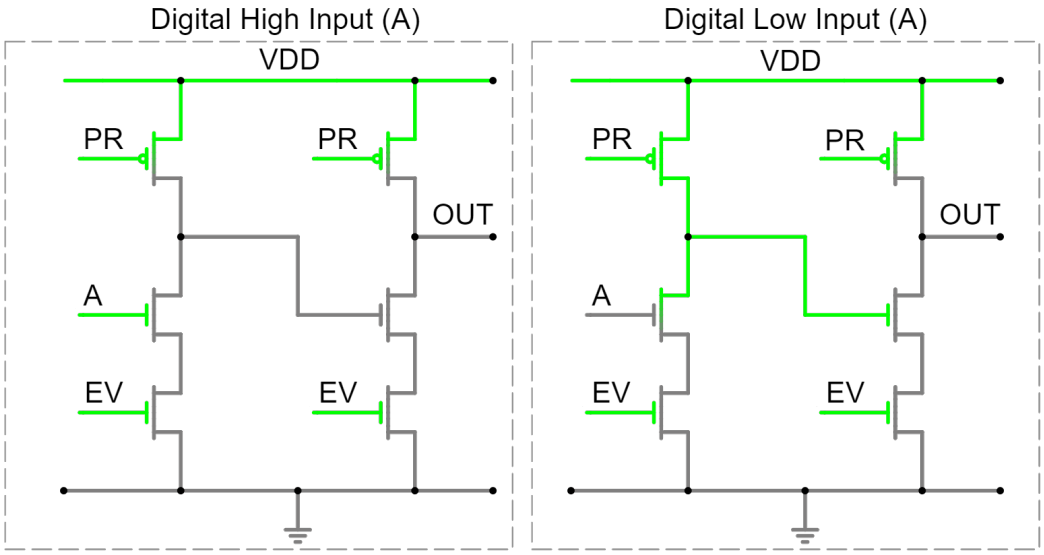

Figure 2.4: The loss of a precharged output due to premature node discharge

2.3 Transmission and Sequential Components in CMOS

2.3.1 Transmission Gates

The transmission gate represents a component utilised within conventional CMOS technologies as a switch. This component can be modulated, thereby enabling or disabling the transmission of signals. As illustrated in Figure 2.5, a transmission gate is observable, which propagates a signal when X is high, and impedes signal propagation when X is low. It is noteworthy that both N-MOSFET and P-MOSFET transistors are present in the creation of this transmission gate. The underlying reason for this can be attributed to the fact that these transistors experience difficulties respectively pulling a node up or down to the gate voltage [11].

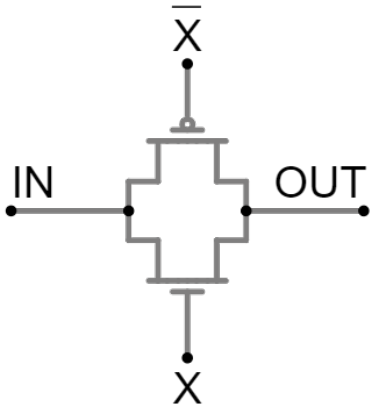

Figure 2.5: Schematic of an transmission gate

It is evident that this cannot efficiently be realised in the context of a unipolar technology, as only N-MOSFET transistors are available. In order to achieve a transmission gate implementation, it is necessary to either elevate the gate voltage to a higher level to ensure a complete transmission of the signal, or to accept the compromise of losing the threshold voltage at the output of the transmission gate. Further in this thesis, transmission gates in IGZO will be referenced to as pass gates, as only one of the two types of MOSFETs are used.

2.3.2 Flip-Flops

Flip-flops are memory elements that capture and store bits on either the rising or falling edge of a control signal, typically the clock. This enables the partitioning of circuits into discrete sections, thereby ensuring the accurate sampling of signals or stable inputs. The most straightforward method for constructing a flip-flop involves the utilisation of two latches (a master and slave). These latches facilitate the propagation of an input signal in response to a control signal. In the event of two such components being positioned in series, and provided that sampling is conducted on opposing clock signals, the creation of a rising or falling edge D flip-flop is realised.

As illustrated in Figure 2.6, two distinct methodologies for the creation of latches are presented. The first method involves the utilisation of a variety of logic cells, while the second method utilises transmission gates and inverters [11]. Transmission gate based latches are usually created in dynamic logic, as the first example requires feedback, creating the possibility for the need to recuperate the lost precharged state.

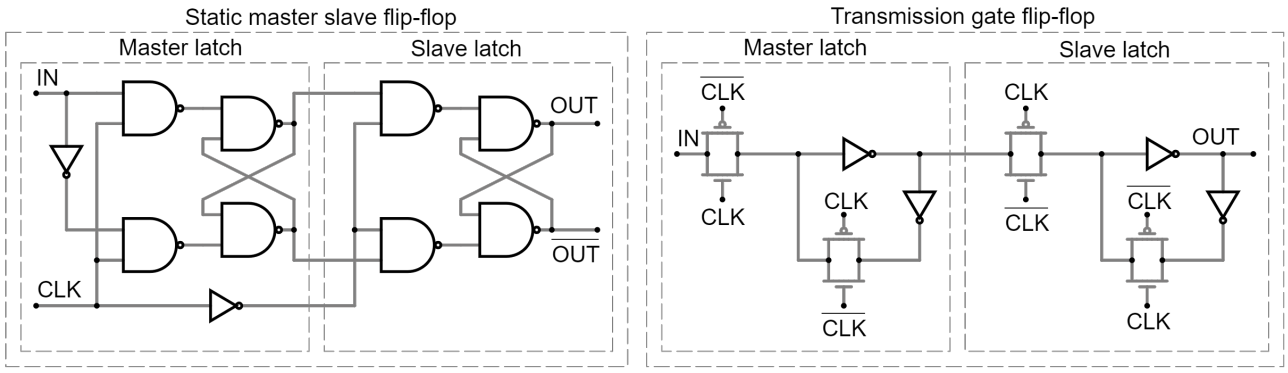

Figure 2.6: Schematic of an static and transmission gate master slave flip-flops

2.4 Dynamic Logic Styles in CMOS

2.4.1 Domino

Domino logic is an example of which can be used to create working dynamic logic in CMOS [30]. The underlying principle of this logic style is to devise a high-speed logic system by maximising the utilisation of N-MOS transistors. An example of two NAND gates in the domino logic style is presented in Figure 2.7.

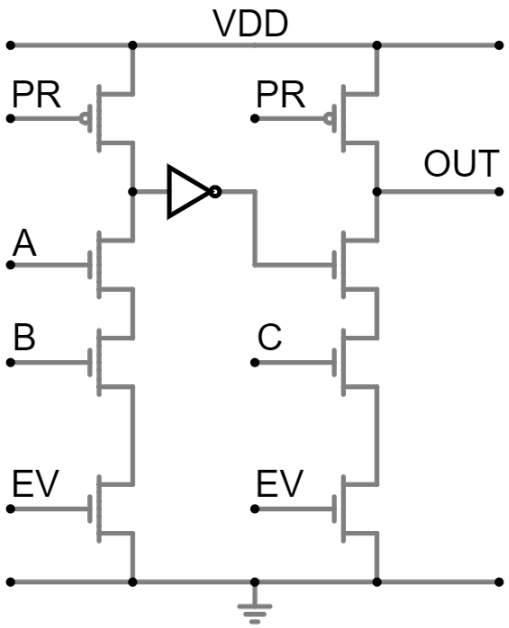

Figure 2.7: Two logic gates in domino logic style

It has been established that premature node discharge is prone to occurrence when different pull-up or pull-down stages are cascaded with each other. This issue can be resolved by incorporating a static inverter at the output of each logic stage, creating domino logic. This ensures that the subsequent stage is closed, thereby preventing early discharge of a node. Nonetheless, it must be acknowledged that this logic style is not without its drawbacks. One such example is the impossibility of creating inverting logic. A further notable disadvantage is the requirement for a strictly managed clock to ensure the functionality of this logic [30], [11].

2.4.2 No Race

Another method of creating dynamic logic while avoiding glitches is to use the No Race (NORA) logic style [31]. The concept of NORA is founded on the principle of utilising pull-up and pull-down networks interchangeably. In this instance, the precharge phase of the current node ensures the subsequent stage is closed, thereby ensuring an environment where the output nodes cannot prematurely lose their precharged state.

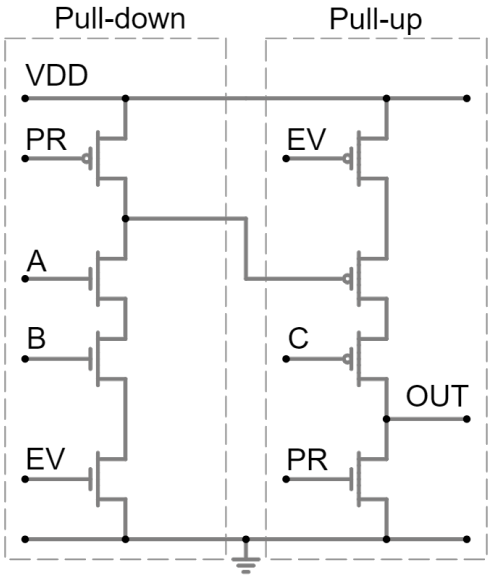

Figure 2.8: Two logic gates in NORA logic style

In comparison with the previously discussed Domino logic style, NORA is a more flexible alternative, as inversion is a possibility because of the use of pull-up networks. This degree of flexibility is still less than that exhibited by static logic. This is exemplified by the fact that it is impossible to send an inverted signal from one pull-up or pull-down stage to the same type of stage, since this is achieved in the same manner as in Domino logic, by using an inverter. In addition, this logic style facilitates the creation of pipelined structures, thereby enhancing the throughput of a logic circuit. Lastly, NORA adopts a less strict approach to clock management in comparison to Domino [31], [11].

2.4.3 Differential Logic

The fundamental principle underlying differential logic is straightforward: both the output and the inverse of the output are computed. The notion was initially put forward in order to facilitate more efficient working logic [32]. A notable property of differential logic is that both output nodes are precharged to the same value, of which one inverts when its logic has been computed [11]. In principle, this can indicate when the logic cell has completed its computation, and potentially prevent the aforementioned glitches by monitoring this.

As illustrated in Figure 2.9, a differential NAND/AND gate is represented. In this instance, the precharge phase demonstrates that both outputs are pulled up. Upon the occurrence of the evaluate phase, one of the nodes is pulled down. Additionally, all the previously mentioned issues associated with dynamic logic also apply to this style of logic.

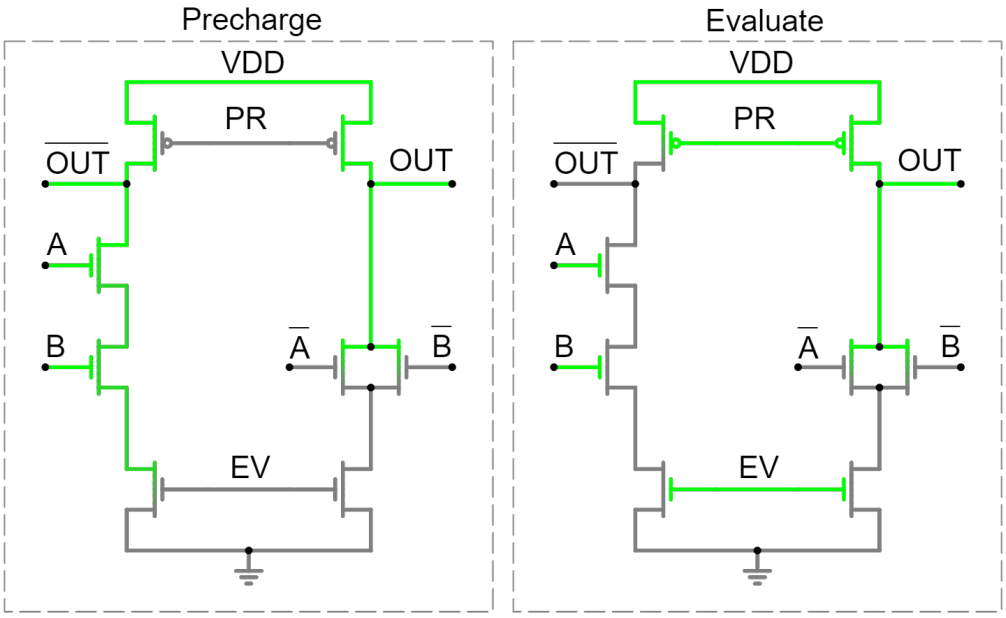

Figure 2.9: Differential AND/NAND gate during precharge and evaluate

2.5 Inverters in IGZO

2.5.1 Bootstrapped Pseudo-CMOS Inverter

The conceptual foundation of this particular inverter is rooted in the principles of bootstrapping [33] and the pseudo-CMOS logic style [14]. It has previously been documented that this inverter has been utilised by [34] for the purpose of creating a ring oscillator. The bootstrapped pseudo-CMOS inverter is illustrated in Figure 2.10.

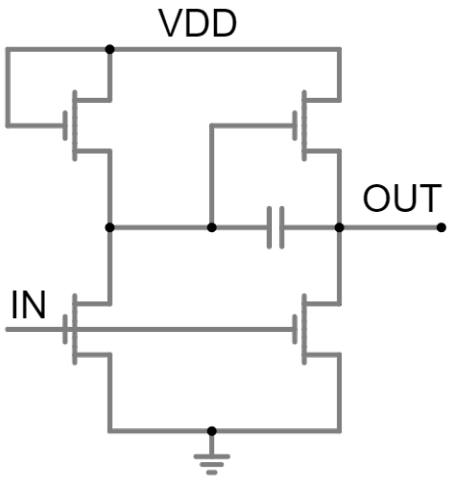

Figure 2.10: Bootstrapped pseudo-CMOS inverter

As previously outlined, the conceptual foundation of pseudo-CMOS is similar to that of 2RLL. In this particular example, the requirement for a single voltage supply is attributable to the bootstrapping functionality of the circuit. This bootstrapping circuit has been demonstrated to be capable of partially recovering the threshold voltage that has been lost in the preceding stage, with resulting speeds in the hundreds of kHz [34].

Chapter 3

Dynamic Logic in IGZO

The following chapter will provide a detailed explanation of the process of creating working dynamic logic in IGZO. The adjustments made will be predominantly attributable to the previously mentioned issues discussed in Literature Study (Chapter 2). Firstly, the concept behind how the proposed logic styles work will be explored. Further, new circuits and adjustments to current logic circuits will be made to realise this proposed logic style.

3.1 Concept of Logic in Unipolar Technologies

The conceptual idea that will be used in this thesis to achieve functional dynamic logic in unipolar technologies is by using a pass gate to allow the output signal to propagate to the subsequent stage. In order to attain this objective, it is necessary to meet two criteria. Firstly, it is necessary to precharge the output to ground in order to prevent early discharge of the output node. Secondly, it is important to ensure that the output of the logic cell is only propagated when its inputs are stable and its logic has been computed.

The fundamental principle underlying the implementation method is the utilisation of a pass gate that will allow signal propagation solely when the logic cell is prepared (Figure 3.1) The pass gate will require an external circuit (the delay cell) to function correctly. In order to ensure that the previously mentioned criteria are met, a small number of minor adjustments will be made to the logic cell.

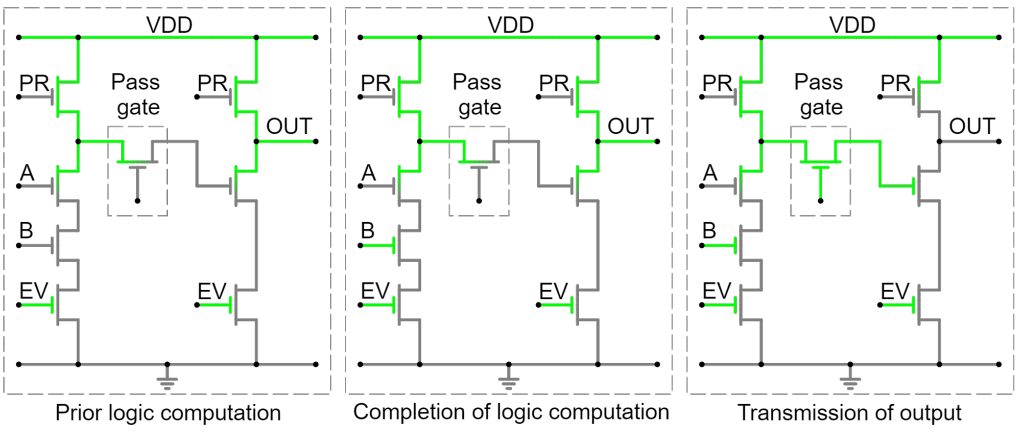

Figure 3.1: Conceptual idea of dynamic logic in unipolar technologies

3.2 Delay Cell

In this particular instance, the purpose of the delay cell is to recreate the ripple effect that is observed in dynamic logic. In this chapter, the initial focus will be on the construction of the delay cell. Next, the characteristics of the circuit will be identified and its behaviour will be demonstrated, thus allowing design rules to be imposed. Finally, implementations will be presented for both single-ended and differential cascading logic using this buffer cell.

3.2.1 Construction

The concept of the delay cell is based on a bootstrapped [33] pseudo-CMOS [14] inverter, a subject which was addressed in Chapter 2.5.1. Figure 3.2 illustrates the conversion of this inverter into the delay cell proposed.

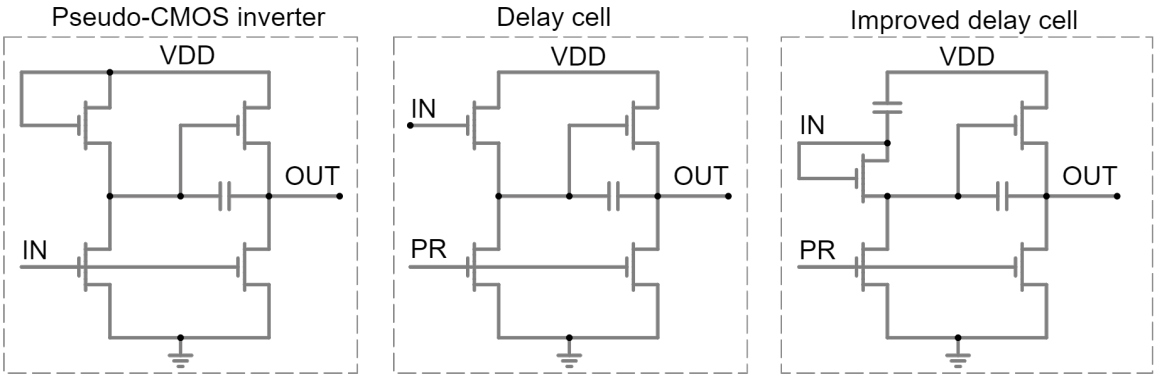

Figure 3.2: Transformation from inverter to delay cell

Transformation to Delay Cell

The initial step in the creation of this delay cell is to transform the pseudo-CMOS inverter. In order to achieve this result, it is necessary for the top-left transistor to be able to accept an input, as this would result in a non-inverted output. It is this same input that will be experiencing a delay, creating a delay cell. This is achieved by disconnecting the gate from the voltage supply, thereby creating its own input.

Secondly, because the cell is designed to be used with dynamic logic, the pull-down network can be connected to the precharge, thus creating two distinct phases. The initial phase occurs during the precharge, at which all bootstrapped nodes and outputs of the delay cells are pulled down simultaneously. The second phase is characterised by the propagation of the delay signal through every stage, with no occurrence of a voltage drop. This complete voltage swing is important as the delay cell will drive a pass gate, creating the possibility to transmitting the entire output signal.

A comparison can be drawn between this circuit and a pull-up network utilised in NORA [31] which was elaborated in Chapter 2.4.2. It should be noted that this circuit cannot be used as a pull-up network, as would be the case in NORA. This implementation would still result in logical glitches.

Improvement of Delay Cell

In order to enhance the stability and speed of the delay cell, it is necessary to connect a load to the output of the delay cell. As will be discussed later in Chapter 3.2.2, the consistency of this load is important to a certain extend. It is for this reason that the design is subject to a number of additional modifications.

The initial modification implemented was the utilisation of the output from the preceding delay cell as both the input and voltage supply for the subsequent delay cell. This is done by using a transistor in diode configuration. This configuration ensures that the current stage is only able to power the next stage, thereby preventing the formation of a single, stretched node. This approach ensures that delays are more consistent by ensuring a predetermined load on every stage.

The final adjustment implemented to marginally increase the speed of the delay cell is the incorporation of a capacitor at the top-left transistor, connecting the voltage supply and input. This facilitates the generation of currents during periods of rapid switching, thereby slightly improving the overall operating speed.

3.2.2 Characteristics

The measurement of the delay cell's characteristics will be accomplished through the construction of a chain of delay cells, where the outputs of the delay cells are connected to the inputs of the subsequent stage. This implementation will also be utilised in the forthcoming logic circuits to create working dynamic logic. As illustrated in Figure 3.3, a visual representation of the 16-stage chain is provided.

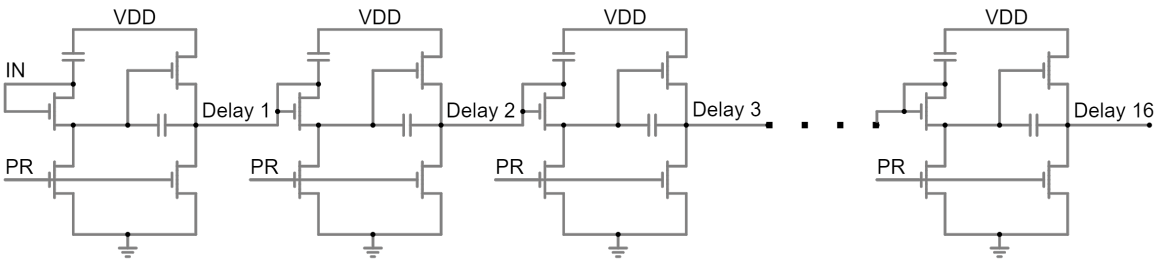

Figure 3.3: 16-stage chain of delay cells

Controlled Environment

The signals generated by this circuit are illustrated in Figure 3.4. In this particular instance, an externally generated signal is applied to the input of the primary delay cell, initiating the cascading effect. Subsequent to this point, a relatively consistent delay can be observed of the input signal. This delay is initiated by a rapid propagation process, transitioning into a slower speed over the course of the process. The occurrence of the longer output delay of stages 15 and 16, and the consequent voltage drop of stage 16 can be attributed to the presence of a floating node at the last output (Figure 3.3). In the event of a load being connected to this node, the delay and voltage drop are minimised, as should always be the case when using these buffer cells.

A fundamental consideration relates to the loads imposed on the outputs of the delay cells. The drive strength is constrained by the top-right transistor which establishes the connection between

the voltage supply and the output. In typical scenarios, this is not a significant concern, as this buffer cell is intended to drive transistor gates and should not be utilised as a voltage supply.

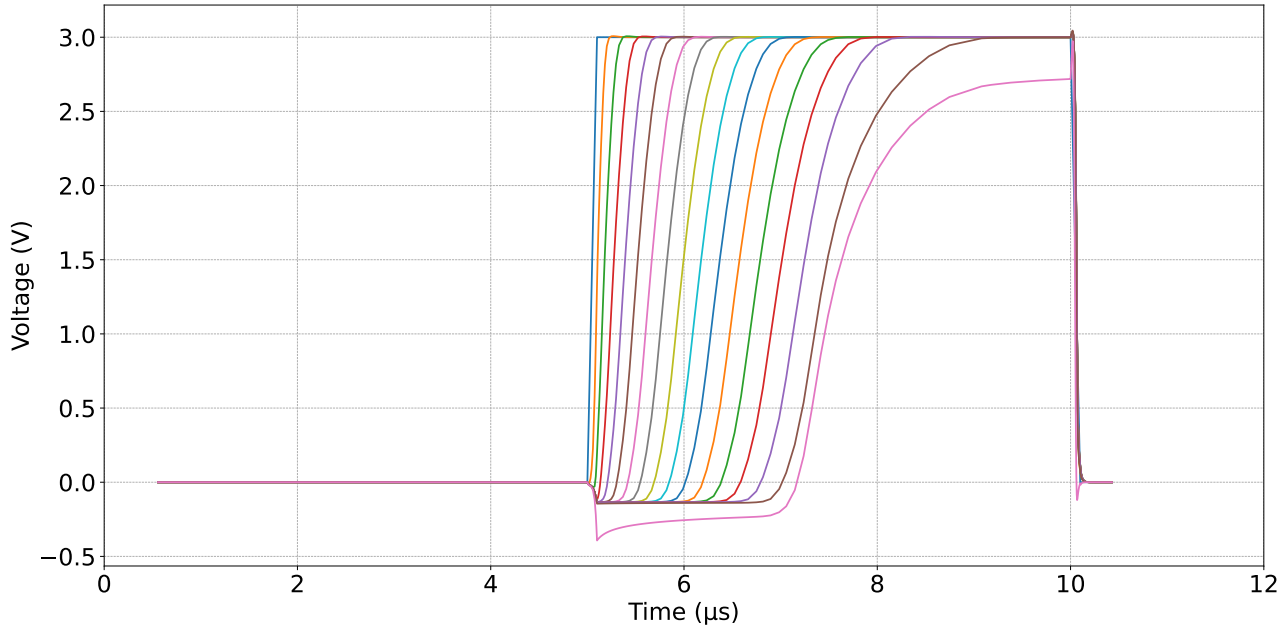

Figure 3.4: Delayed signals produced by the delay cell

Realistic Environment

It is imperative to ensure that the loads applied to the outputs of the delay cells are balanced to a certain extent. In the absence of proper load matching, inconsistencies such as those illustrated in Figure 3.5 may emerge. The underlying reason for this phenomenon can be attributed due to the fact that not all stages have an equal number of logic cells that need to be driven. In general, such inconsistencies should not result in faulty logic; rather, they can result in logic that is slower in operation, owing to inconsistent computation timings. Consequently, it is not worthwhile to optimise the loads of every stage in order to ensure consistency on each output. However, this factor should be taken into consideration during the design process of a logic circuit, as this has the potential to influence the performance of the circuit.

In instances where it is necessary to ensure a uniform delay between distinct stages, the utilisation of capacitors at the outputs of each delay cell can serve to equalise the load across each stage. The utilisation of these capacitors should be exclusively for the purpose of equalising the capacitance of each stage.

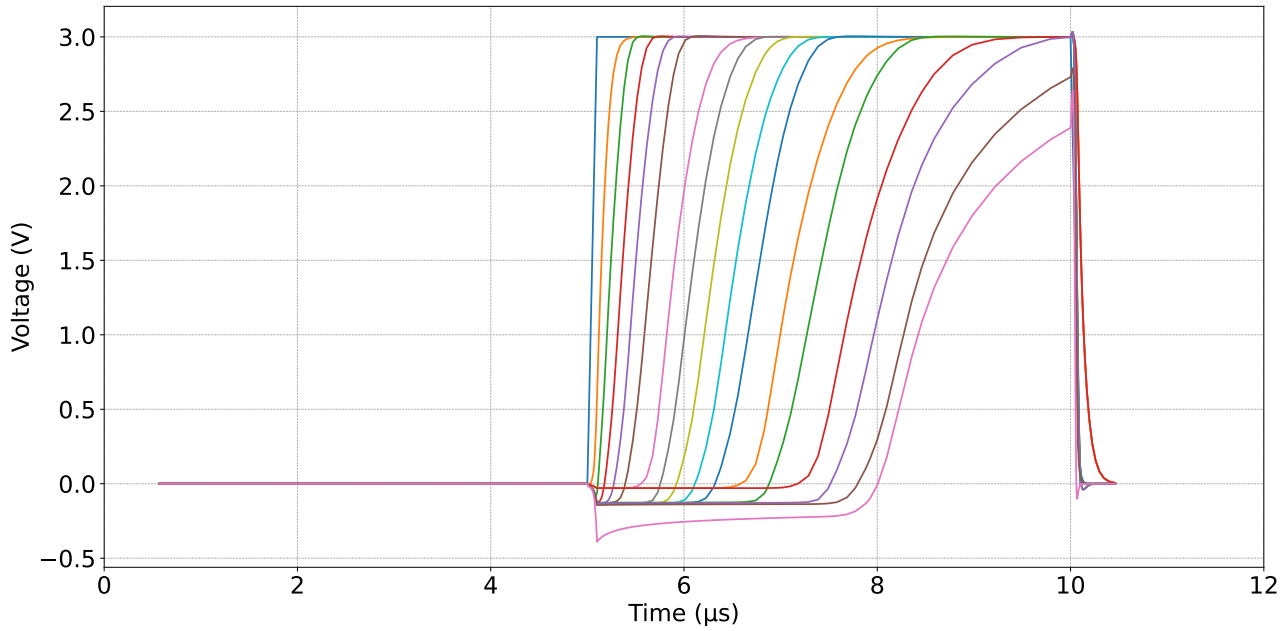

Figure 3.5: Inconsistent loads on delay cell

Operational Speed

A further point of interest regarding the characteristics of the delay cell is that its operational speed is significantly influenced by the voltage supply. As illustrated in Figure 3.6, the fourth delayed signal is represented from the 16-stage chain using a 2 V, 3 V and 4 V supply voltage. The calculation of the time between two consecutive signals enables the determination of the speed of a delay cell. This procedure was repeated for each voltage level, resulting in delays of 800 ns, 250 ns and 100 ns for 2 V, 3 V and 4 V respectively. This indicates an increase in speed of 3.2 times from 2 V to 3 V and 2.5 times from 3 V to 4 V. It is important to acknowledge that the 4 V measurement may be less accurate, as this voltage is higher then the model is characterized for.

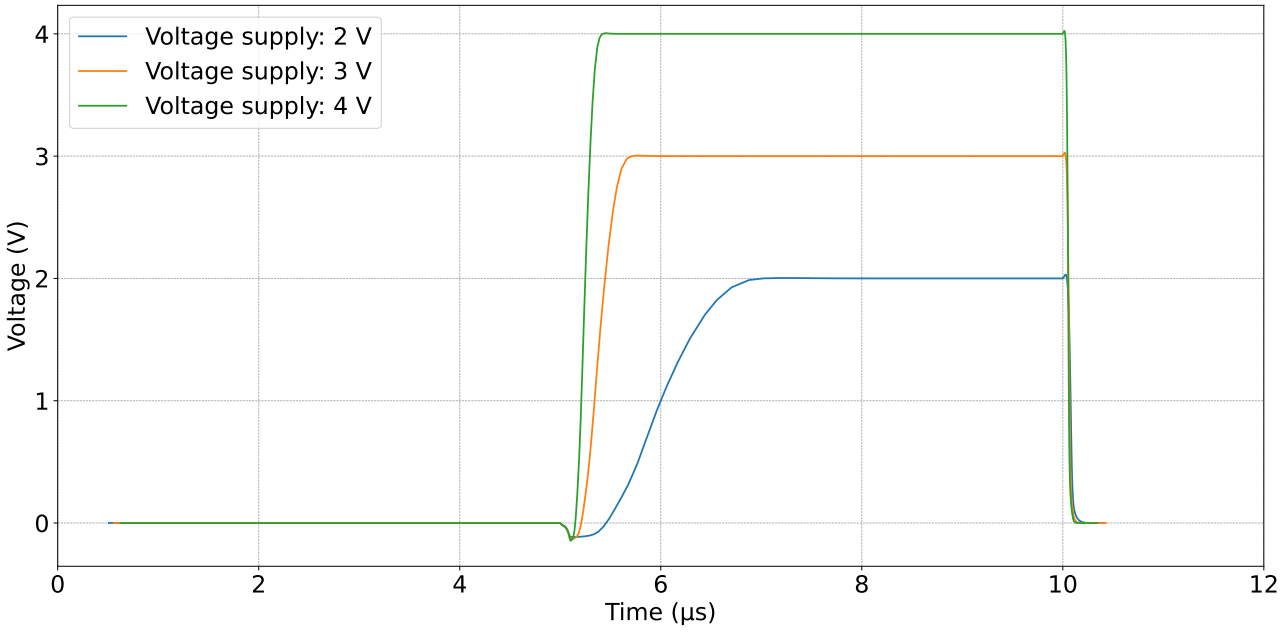

Figure 3.6: Speed of delay cell based on supply voltage

Process Variation

Finally, it should be noted that delay circuits are particularly vulnerable to process variations. This is illustrated in Figure 3.7, which presents a simulation of the slow, typical and fast corners of the N-MOSFET transistor. It is important to note that the simulated situations are improbable, yet a considerable difference in speed is observed depending on the corner of the simulation. In consideration of this, it may be necessary to minimise process variation, particularly in instances where timing is of high significance.

In the context of typical applications, this should not be a significant consideration. The hypothesis is that if the mean of all delays experienced by each stage is taken, it will be equivalent to that of an delay cell with typical transistors. A notable consideration is the potential mismatch between the speed of a fast delay cell and that of a slow logic cell, where the former may be too rapid for the latter. No metrics were collected to substantiate this hypothesis. As will be discussed in greater detail at a later point, the delay cell is considerably slower than the logic itself. This might suggest that a faster delay cell will not result in logical glitches. This should nevertheless be taken into consideration during the use of these buffer cells to avoid glitches.

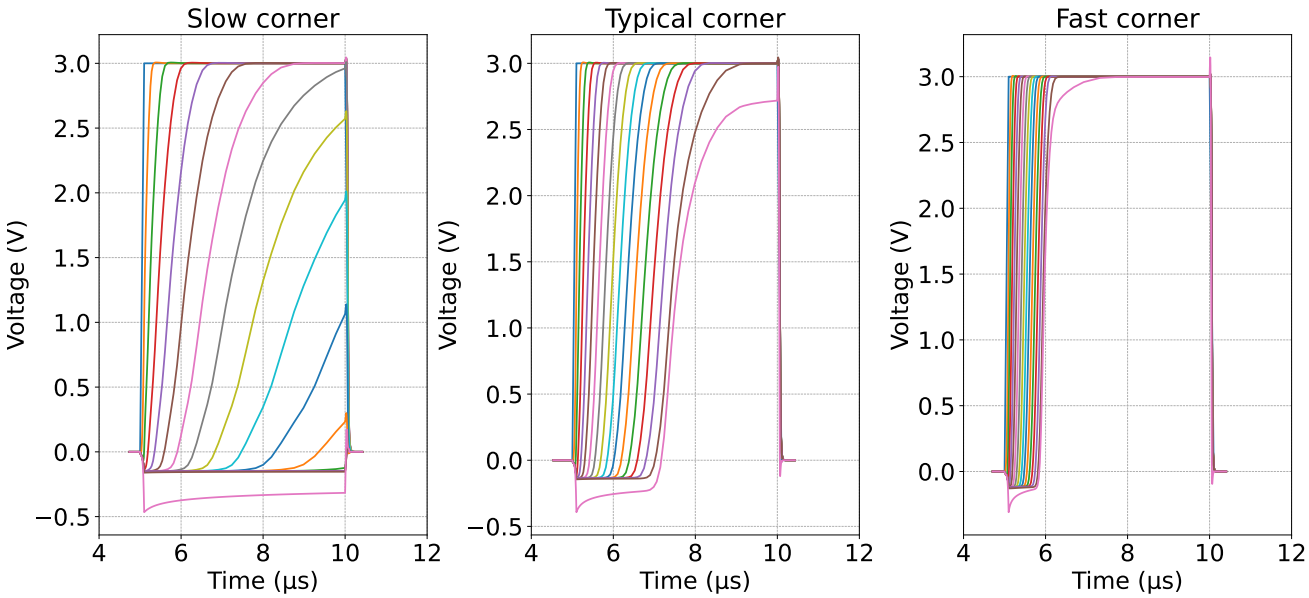

Figure 3.7: Simulated corners of delay cell

3.2.3 Component Sizing

In the design of this delay cell, efforts were made to ensure that almost no leakage current flows to ensure a power efficient implementation. Subsequently, the delay cell was optimized to maximise operational speed. The underlying reason for this is that the logic performs its computations at a significantly faster rate than the delay cell itself. As will be seen later, in Figure 4.5 the slow 2RLL version is seven times faster than the delay cell itself. The initial components examined to increase the speed of the delay cell are the precharge transistors. The dimensions of these precharge transistors are 1 µm in width and 1.2 µm in length. The implementation of this procedure is intended to guarantee minimal leakage, improving the output and bootstrapped nodes' capacity to be pulled up, thereby enhancing the speed of the cell.

The capacities' sizes utilised are found to have no effect on the speed or stability of the circuit. It is imperative to acknowledge that the capacity of the bootstrap capacitance must be sufficiently large to drive its load. This can be achieved with a capacity of 20 fF, which is sufficient to drive most, if not all loads in typical circumstances. In order to determine the capacity at the input of the delay cell, the smallest possible value of 5 fF is selected, as this does not have any effect on the cell.

The final components to be examined are the two pull-up transistors. It is fundamental to establish an optimal size that balances both current flow and parasitic capacitances of the transistors. The input transistor exerts a marginal influence on the delay created by the cell, but more significantly impacts the slope. A reduction in the size of the transistor results in an slight increase in the delay, whilst also resulting in a steeper slope. It is apparent that the dimensions of 2 µm for the width and 600 nm for the length appear to be a suitable compromise in terms of both speed and steepness. The transistor driving the output is selected to be of the same size. The determination of the optimal size for this transistor is dependent on the driving strength of the input transistor, given the lack of feedback in the output stage.

3.3 Circuit Design with Delay Cell

3.3.1 Logic Cell's Components

The initial phase of this section involved modifications to the logic cell. Figure 3.8 presents a visual representation of the various components that are present in all logic cells. The additional components added to the logic cell comprise out of an output controller and a keeper.

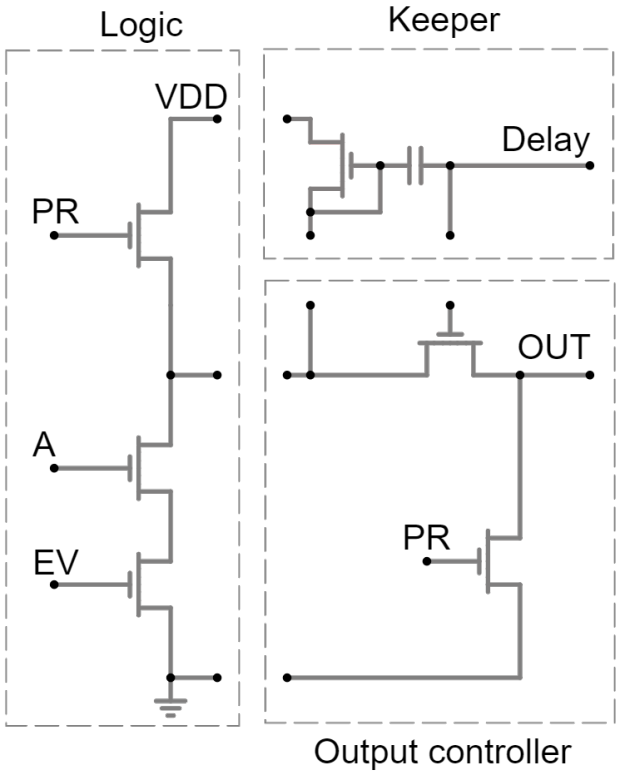

Figure 3.8: The different parts of a logic cell

The operational mechanisms of the logic cell will not be elaborated upon in this chapter, as they have been previously discussed in Chapter 2.2.1. It is important to note that the output of this circuit is always lower than the supply voltage, due to the use of a N-MOSFET as precharge transistor, which is not the case in previously explained examples.

Output Controller

The output controller is composed of two sections: a precharge transistor to the ground and a pass gate (Figure 3.9). As was outlined in the Literature Study, premature node discharge (Chapter 2.2.3) can result in logical glitches. The output controller is utilised to address this issues. More specifically, the output controller makes use of a pass gate to propagate the output signal once it has been computed, and a precharge gate to precharge the output to the ground.

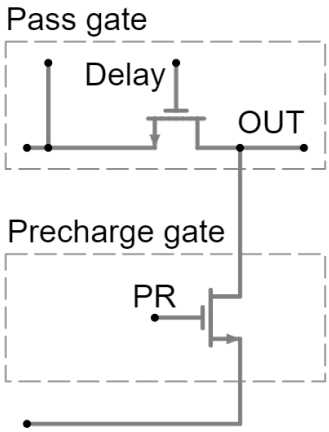

Figure 3.9: The output controller in logic cells

Domino Logic [30] (Chapter 2.4.1) serves as a notable comparison in this regard, utilising an inverter to precharge its output to the ground. The same outcome is attained in this instance through the connection of a transistor from the output to the ground, which is powered by the precharge phase. However, this in itself is inadequate, as the creation of a short and an unknown precharged state are both observed when both the pull-up and pull-down transistors opened and closed simultaneously.

In order to resolve this issue, the implementation of a pass gate is required, thereby ensuring the above mentioned requirements. The precise timing of this process is dictated by the operation the delay cells (Chapter 3.2). The precise manner of this process will be discussed at a later point.

Keeper Transistor

As stated in the literature study, charge sharing is identified as a second problem (Chapter 2.2.2). The issue of charge sharing was found to be a contributing factor to the suboptimal driving strength of the logic cell, which in turn may resulted in the occurrence of logical glitches.

One of the proposed solutions involved the utilisation of a keeper transistor [27, 28], which is powered by the output of the logic cell. It is apparent that this will not be sufficient, as an N-MOSFET transistor is incapable of elevating a node to the same voltage as the gate voltage that is driving it. The solution to this issue involves the implementation of capacitive boosting,

a technique that is utilised to temporarily elevate a node to a higher voltage than the supply voltage, as shown in Figure 3.10.

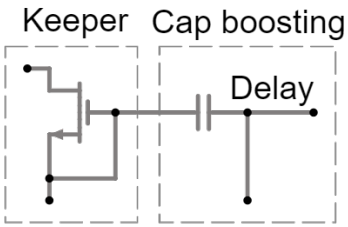

Figure 3.10: A keeper transistor with capacitive boosting

It is essential to ensure that the keeper transistor is powered at the appropriate time, as capacitive boosting is a phenomenon that is only present for a brief period. Consequently, this capacity is connected to the delay cell, which increases its voltage output at the moment the pass gate opens. This is also the point in time at which uncharged nodes may need to be charged.

3.3.2 Single-ended Cascading Logic

Following the clarification of the adjustments of the logic and delay cells, the next step is to provide an example of its implementation. The initial implementation that will be examined is that of singe-ended logic. An illustration of this is provided in Figure 3.11, which depicts an inverter chain comprising of three stages. It is noteworthy that each delay cell in this schematic is responsible for its own stage. It is therefore possible to connect multiple logic gates to a delay cell, provided that they belong to the same computation stage, thus not causing any logical glitches.

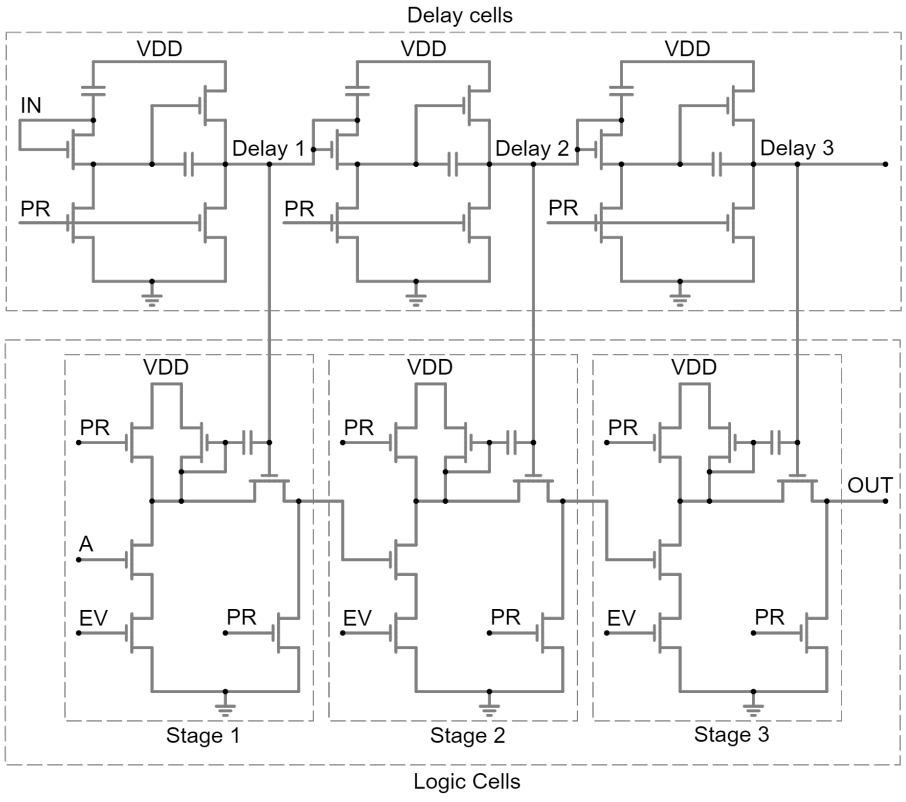

Figure 3.11: Implementation of single-ended cascaded logic with delay cell

A more in-depth visual illustration of the functionality of the delay cells is presented in the Appendix (Figure A.1). The fundamental concept underlying this design is that the delay induced by the buffer cells is sufficiently long to allow the logic cells sufficient time to compute their logic. The most important thing is that the delay cell is significantly slower than the logic for these circuits, this guarantees correct functionality.

3.3.3 Differential Cascading Logic

The operation of the differential delay cell is marginally different from the delay cell previously discussed in Chapter 3.2. The normal delay cell fulfils the role of a buffer, thereby preserving the input signal. In turn, the differential delay cell functions as a leaking NAND gate that uses the previous stage as a voltage supply. This is achieved by replacing the precharge transistor at the output stage of the delay cell with the two differential precharged outputs of the logic cell (Figure 3.12).

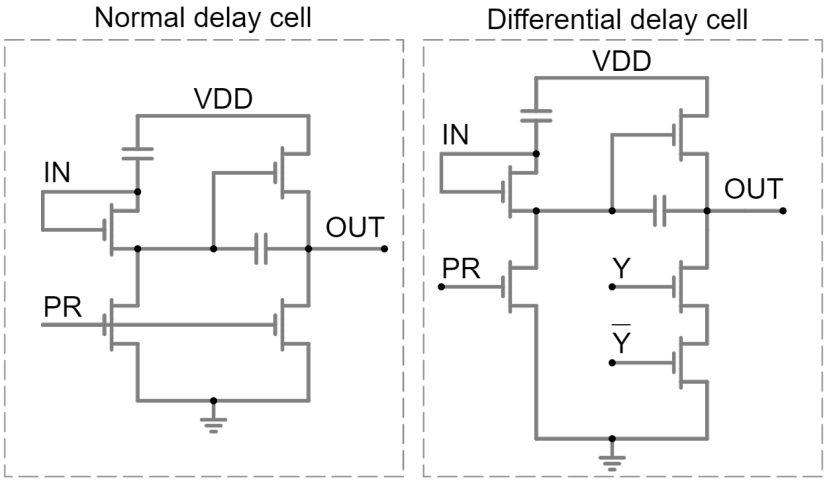

Figure 3.12: Comparison of normal and differential delay cell

As a consequence of both outputs of the differential logic cells being precharged to the supply voltage, the delay cell will begin to leak. It is worth noting that this is only the case for the current stage, since the following stages are powered by the preceding ones. It is only upon the completion of the computation by the logic cell that one of the two pull-down transistors of the delay cell will close. This, in turn, will power the next stage's delay cell and open the pass gate of the logic cell. A concrete example of this is provided in the appendix (Figure A.2).

Figure 3.13 provides an illustrative representation of an differential inverters with a delay cell that has been connected according to the previously discussed principle. A notable distinction between this schematic and the typical cascading logic is that each logic cell incorporates its own delay cell, as opposed to each stage sharing a delay cell.

Consequently, it is possible for a single delay cell to be responsible for powering multiple logic cells. To elaborate further, a logic cell also drives all other logic cells in which its own output is routed to. It is imperative to note that in circumstances where a logic cell receives multiple outputs from other logic cells, the selection of a delay signal from the preceding logic cells must be made. In this particular implementation, it is not possible to make faulty logic, since the propagation of the output will only happen after the logic computation of the cell has been

completed. In principle, however, it is recommended to utilise the slowest logic cell, as this will minimise leakage and optimise power efficiency. A demonstration of routing in this logic style can be found in Figure A.3 of the appendix, where the bottom cell is utilised to propagate its delay to subsequent cells.

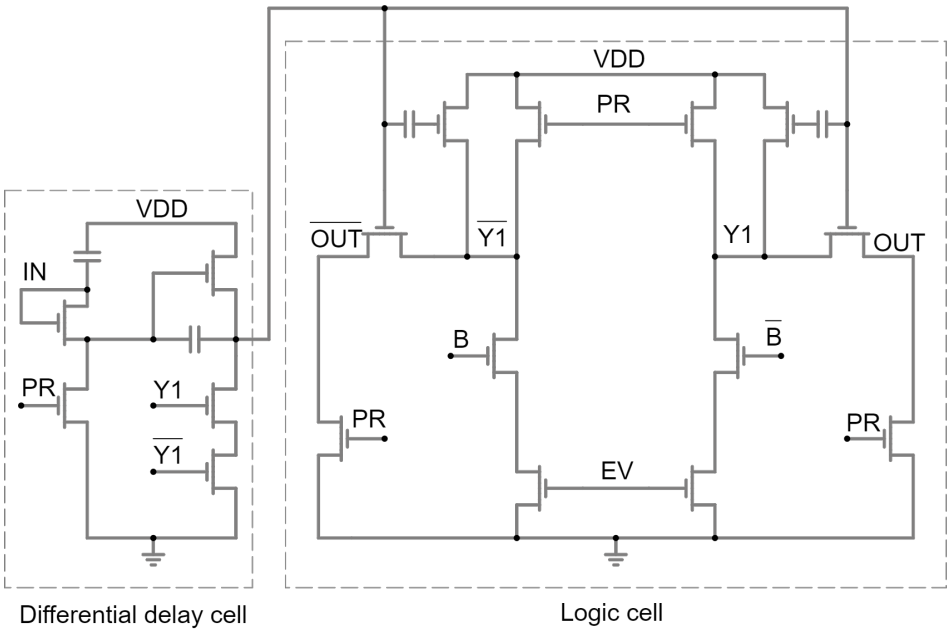

Figure 3.13: Implementation of differential cascaded logic with NAND delay cell

Chapter 4

Circuit Design and Validation

The validation circuit utilised was a 4-bit adder with flip-flops at both the input and output of the adders. Due to the size of these schematics, a written explanation will be provided alongside the relevant schematics in the Appendix.

4.1 Circuit Design

4.1.1 Adder

The initial circuit to be examined is the adder itself. In this particular instance, exclusively full-adders are implemented. It could be argued that this arrangement is suboptimal, given that only a half-adder is required for the initial stage. The use of only full adders is due to the fact that it is a more efficient approach of data collection from the validation circuit, thus reducing the time required for the design process. The adders under consideration are constituted by two XORs in series in order to calculate the sum, and three AND gates connected to an OR gate in order to calculate the carry. As non-inverting gates cannot be implemented using only a single-stage pull-down network in dynamic logic, their inverting counterparts are used instead, followed by an additional inverting stage to restore the intended non-inverting functionality.

Single-Ended Cascading Adder

The schematic of a single-ended adder can be found in Figure 4.1. The adder is comprised of five distinct stages. This example illustrates the use of a single delay cell to power an entire stage of logic cells. Furthermore, it is demonstrated that not all delay cells experience equivalent loads; this is evidenced by the requirement for different amounts of logic cells per stage to be driven.

Differential Cascading Adder

The schematic for the differential logic can be found in Figure 4.2. It is important to note that the visual representation is limited to the logic gates, due to the fact that the delay cell are incorporated within the logic cell, as previously outlined in Chapter 3.3.3. Because of the nature of how the differential logic works, a decision must be made regarding the propagation of the delay. In an ideal scenario, the slowest path would be selected to ensure that the delay cell does not leak. However, if a faster path is selected, no logical glitches will occur.

Two-Stage Resistive-Load Adder

The 2RLL adder can be found in Figure 4.3. Two different implementations are made with exact the same schematic, with different electrical components to make a fast (higher leakage) and slow (lower leakage) version. The only thing that needs to be kept in mind here is the fact that two voltage supplies are used. This can not be seen in the schematic itself but is still important to consider when comparing the implementations later on.

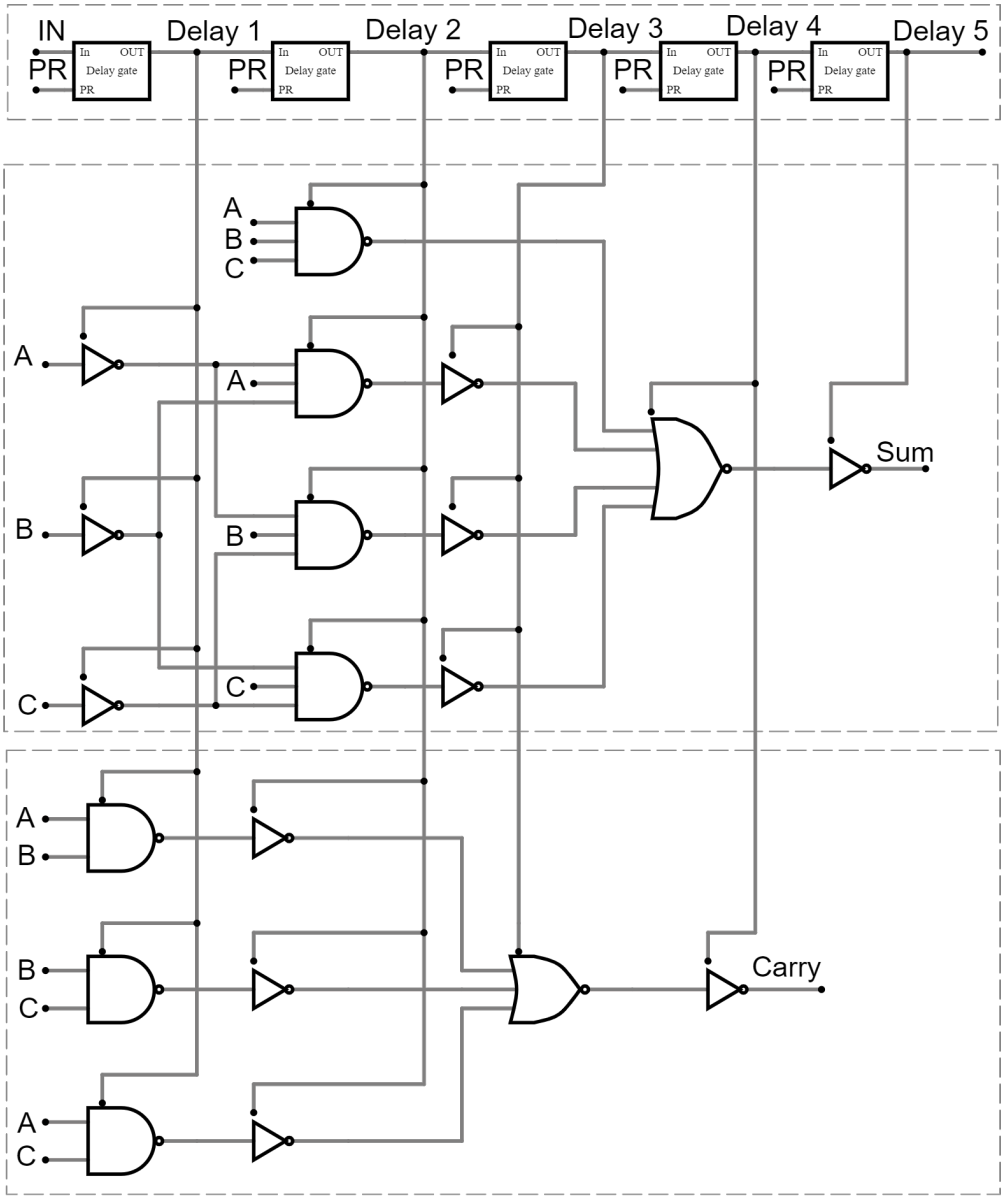

Figure 4.1: Schematic of a dynamic single-ended adder with delay cells

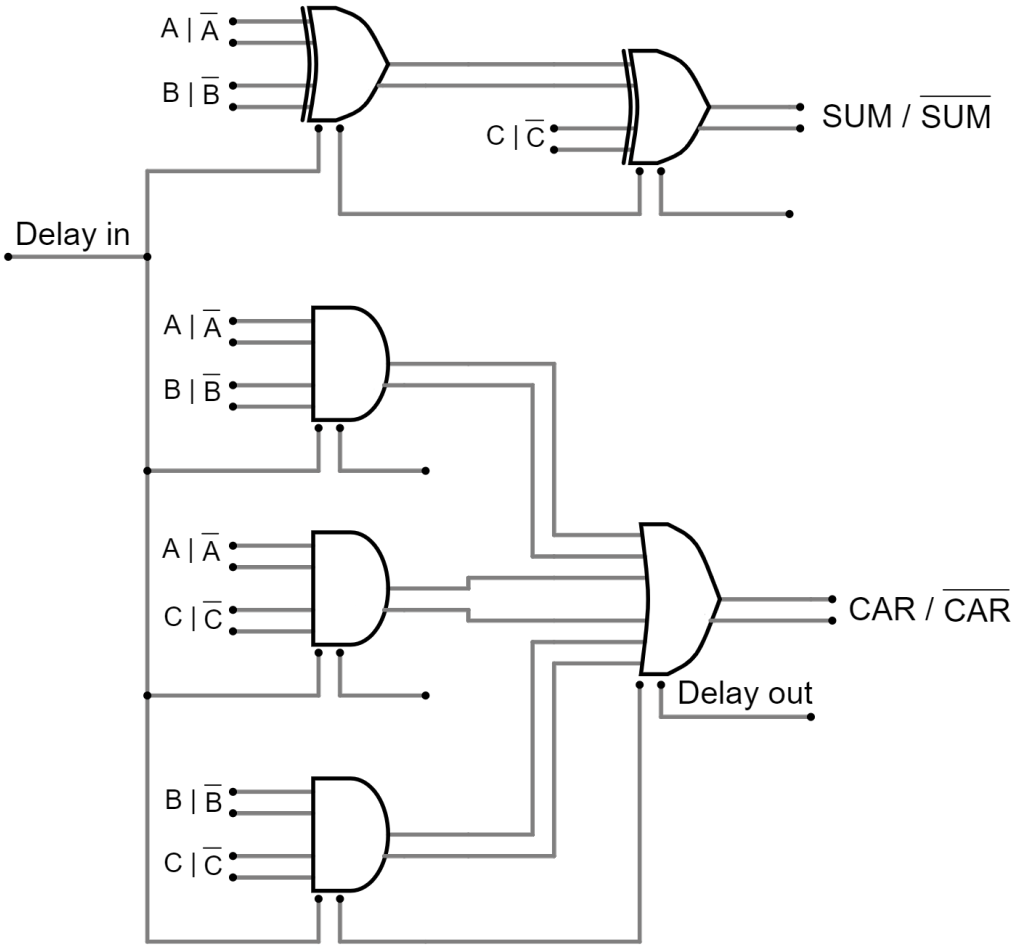

Figure 4.2: Schematic of a dynamic differential adder with delay cells

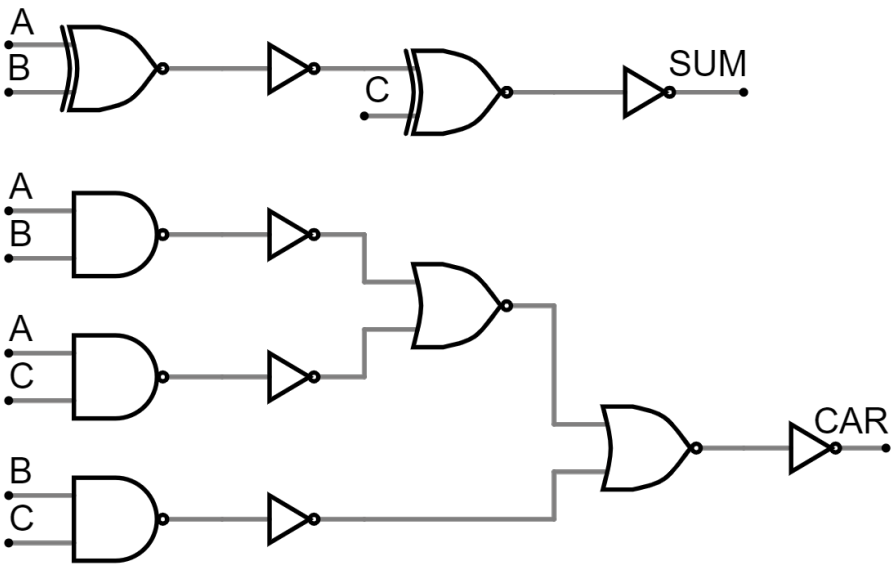

Figure 4.3: Schematic of a static 2RLL adder with delay cells

4.1.2 Flip-flops

Dynamic Flip-Flops

The dynamic flip-flops created are based on transmission gate flip-flops. In order to realise the aforementioned concepts, it was necessary to make minor adjustments to the logic cells and the design of transmission gate flip-flops (Figure 4.4).

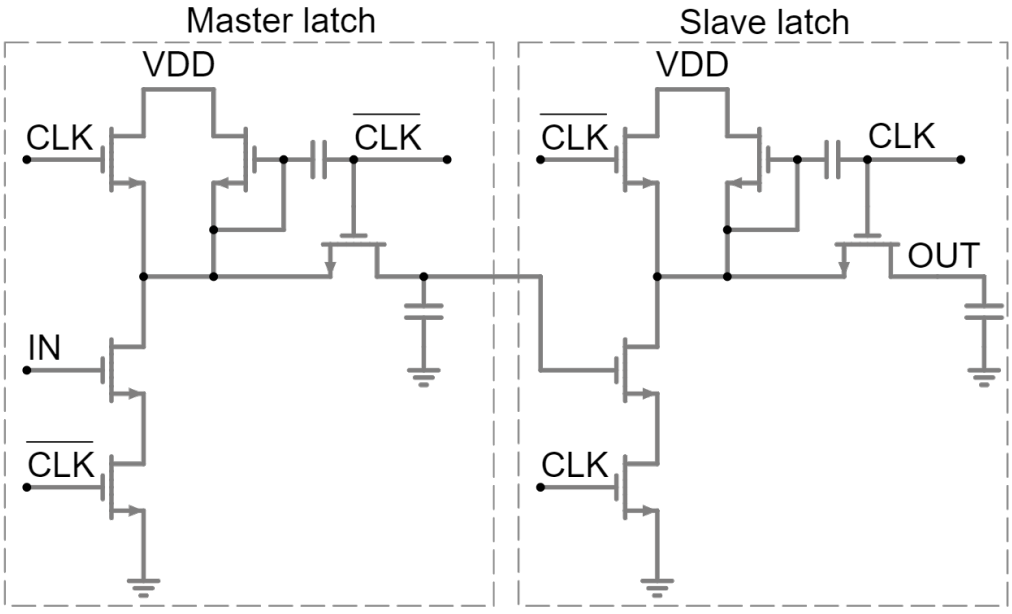

Figure 4.4: Dynamic flip-flop

The initial modification required to the logic cells involved the control element. This particular element is equipped with a transistor that precharges the output to ground. It is evident that this is no longer required, given that a latch functions as a memory cell which retains the last input when it becomes deactivated. This component is substituted by a capacitor behind the pass gate, with the objective of guaranteeing that the most recent state sampled is stored. It is important to note that the pass gate is powered by the evaluate signal as opposed to a delay signal. This is acceptable as the subsequent stage is precharging during this period, making the subsequent stage Insusceptible to the occurrence of previously discussed logical errors. This also ensures that the latch remains active throughout the entire phase of logic computation until completion. Because the latches used in the flip-flop function in opposite phases, the clock is utilised to denote the phases of the precharge and evaluate transistor to avoid confusion. In order to ensure the correct functionality of the flip-flop, it is essential that the first latch is aligned with the phases of the logic powering it. In this particular instance, CLK would be the precharge phase and $\overline{CLK}$ the evaluate phase.

The second adjustment made is to the transmission gate used by the latches. As previously explained, the logic cells themselves already make use of pass gates, and therefore they have been relocated from the input of the inverter to the output. Secondly, no feedback is provided via a second inverter. IGZO transistors typically have extremely low leakage, which is the main reason why an unstable state is not a significant concern when running at a normal clock frequency of higher than 1 kHz [35]. Secondly, the utilisation of dynamic logic results in the absence of the possibility for feedback. This is due to the fact that, once the precharge is lost, it cannot be recovered until the subsequent precharge phase.

Two-Stage Resistive-Load Flip-flops

In the case of the 2RLL implementation of a flip-flop, the static version is utilised. This one is based one eight NAND gates and shown in figure 2.6

4.2 Circuit validation

This chapter will present the findings derived from the construction of the pipelined 4-bit adders in the static and dynamic logic styles. Initially, a power consumption comparison will be conducted, with the analysis being based on varying clock frequencies. Subsequent to this, the power consumption in relation to different voltage sources will be displayed for each dynamic logic style. Finally, an estimated area of each logic style.

4.2.1 Average Power Consumption of 4-Bit Adders and Flip-Flops

Results

The average power consumption of the two 2RLL and proposed dynamic logic implementations is illustrated in Figure 4.5. It should be noted that the voltage supply used for the dynamic logic is 3 V, while the voltage supply for the 2RLL is 2 V with a bias voltage of 3 V.

This graph (4.5) illustrates that the power of the proposed dynamic logic styles can achieve less than 1 µW power consumption at very low operating frequencies whilst the power consumption of the 2RLL implementations exceeds the 100 µW. While the maximum operational speed of the differential 4-bit adders (140 kHz) is approximately seven times slower than that of the slow version of the 2RLL 4-bit adder (1 MHz), the power efficiency of the circuit is enhanced by approximately 19 times, from 435 µW to 22.8 µW at its fastest frequency. This graph also demonstrates that power efficiency is directly proportional to clock frequency, with lower frequencies resulting in increased of power efficiency. This is not applicable to 2RLL implementations, which demonstrate a constant power consumption mostly independent of the clock frequency.

It is noteworthy that two non-linearities are visible in the graph (4.5). The first one is found at the final measuring point of the slow variant of the 2RLL logic. The second non-linearity is located in the differential dynamic logic, a property that becomes more pronounced at lower frequencies.

Discussion

The following discussion will commence with an explanation as to why the differential logic exhibits a superior power efficiency to the static logic. This phenomenon can be attributed to the absence of leakage currents in the dynamic logic in comparison to its static counterpart, which are independent of clock frequency. Its optimal performance is therefore only achieved when operating at maximum frequency. This is not the case for the dynamic logic styles. The dynamic logic styles possess the capacity to remain inactive and not consume any power. This property enables them to operate at lower frequencies, thereby conserving energy over time.

It must be noted, however, that this does not imply that the dynamic logic styles are more efficient in terms of logic computation. When the measuring points are extended linearly beyond their maximum operating frequency, which is limited by the delay cell, the power consumption

of the dynamic logic style approaches that of the static logic style. This indicates that the logic/buffer cells themselves do not necessarily consume less power. The capacity to remain idle and not leak any current is the primary factor contributing to enhanced efficiency in this regard.

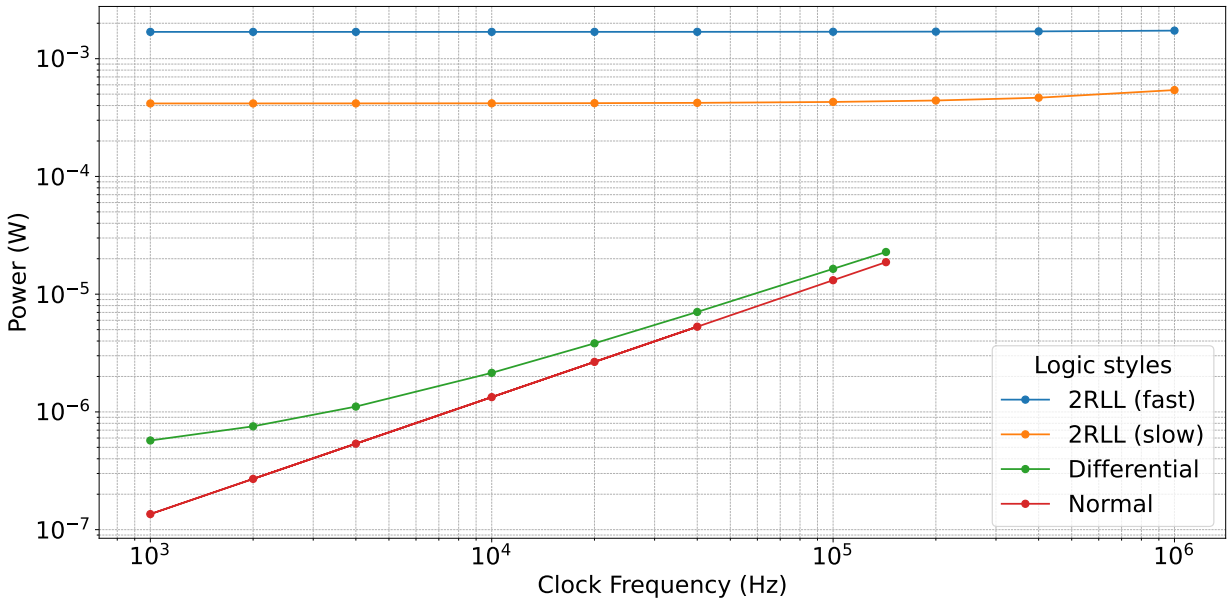

Figure 4.5: Average power consumption of all logic style pipelined 4-bit adders

Finally, the discussion will turn to the abnormalities observed in the slow 2RLL and dynamic differential implementation, which can be seen in Figure 4.5. Firstly, it was anticipated that the 2RLL logic would exhibit a linear trajectory leading to its maximum operating frequency. The rationale behind this deviation is attributable to the fact that, at elevated operating frequencies, the power consumption associated with switching becomes large enough to be observed on the graph. The second anomaly that can be observed is in the differential dynamic logic. In this instance, a linear trend was anticipated, in accordance with the single-ended dynamic logic. The phenomenon of diminished power efficiency at reduced operating frequencies can be attributed to the operation of the delay cell. The differential delay cell exhibits a leaking state until the logic cell has completed its computation. This demonstrates that the incorrect logic cells were designated as the slowest path, resulting in leakage in the delay cells during the wait for other logic to be computed.

4.2.2 Average Power Different Voltages

Results

As demonstrated in Figures 4.6 and 4.7, the average power consumption of single-ended and differential dynamic logic is shown to depend on varying supply voltages. Evidently, the diverse power consumption exhibited by each logic style adheres to a linear trend, characterised by a constant offset between them. It is vital to acknowledge the nature of the data points displayed on a logarithmic scale. Despite the uniformity in appearance across the entirety of the graph, the actual numerical value of the difference between each data point increases exponentially in proportion to the growth of the y-axis.

As previously discussed, Figure 4.7 also displays the non-linearities found in Figure 4.5. Furthermore, these non-linearities diverge at varying rates in accordance with the supply voltage; a greater divergence is observed at higher supply voltages and vice versa.

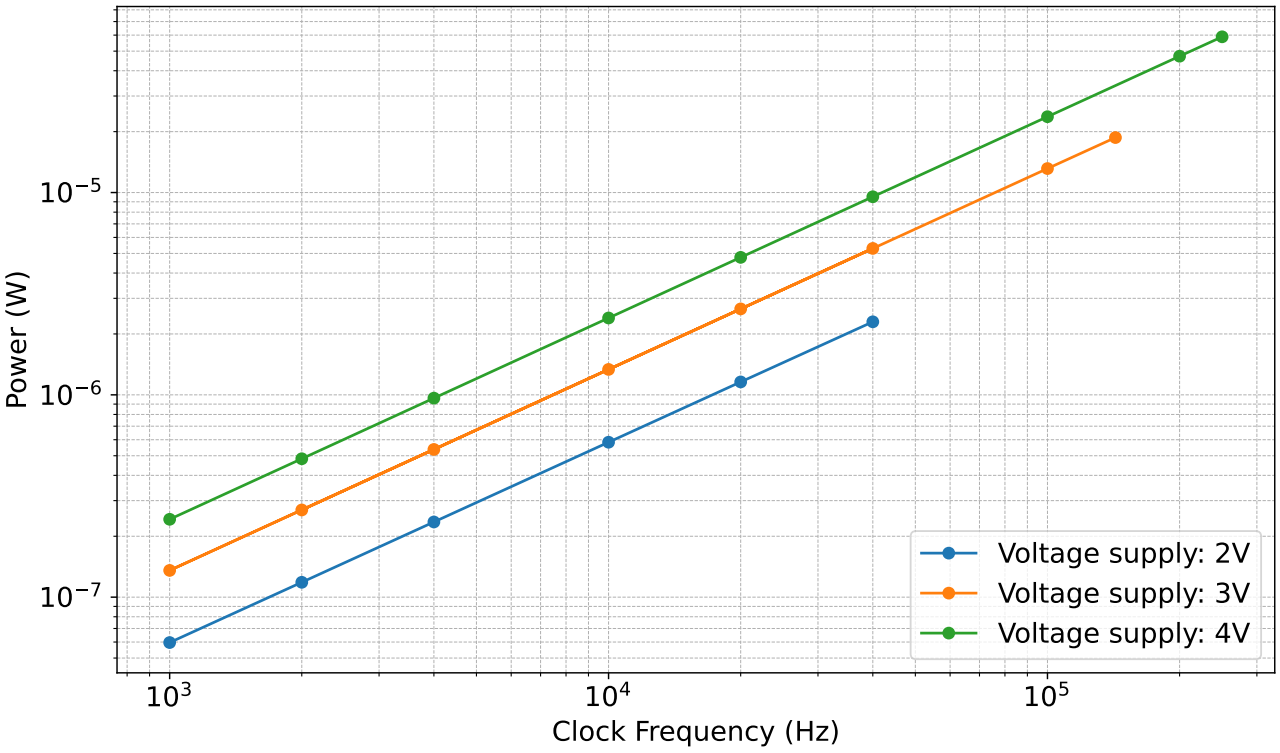

Figure 4.6: Power consumption of single-ended logic based on supply voltage

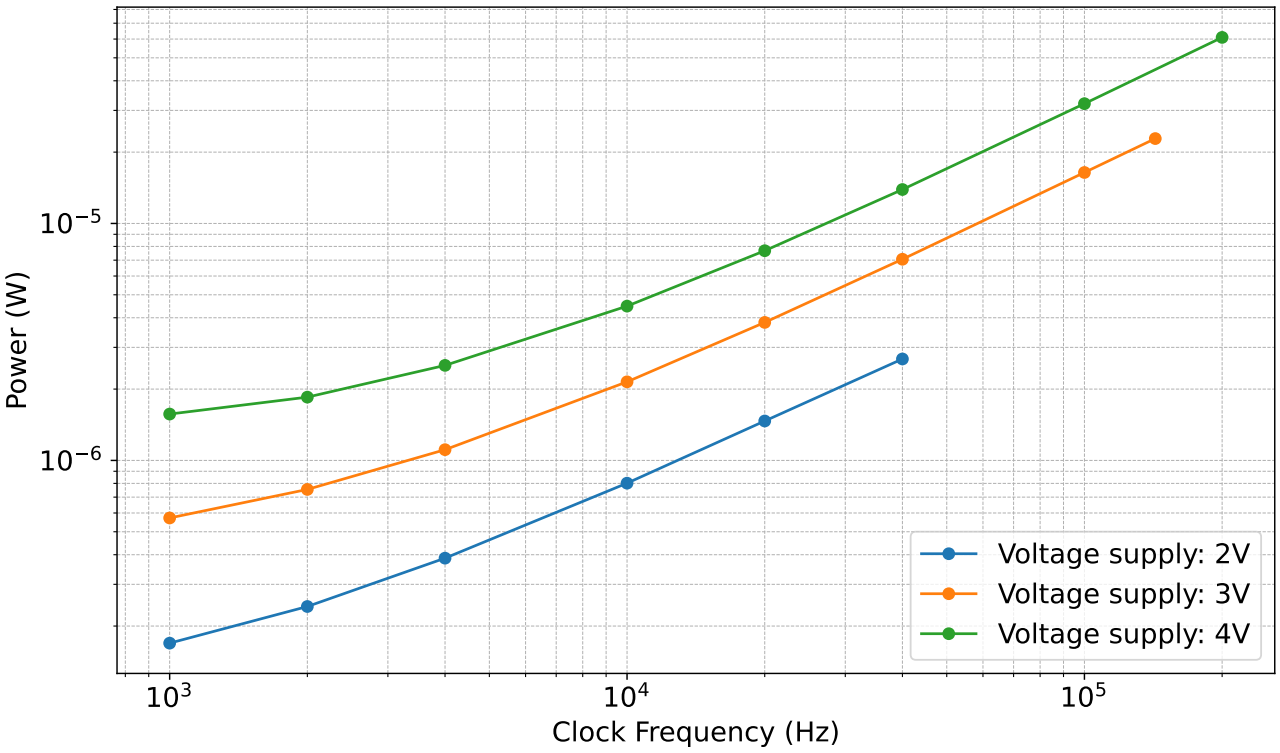

Figure 4.7: Power consumption of differential logic based on supply voltage

Discussion

On initial observation, there appears to be negligible variance in power consumption in relation to differing voltage supplies. This would be a misinterpretation, however, as both axes are on a logarithmic scale. Should the axes be transformed back to a normal scale, the different voltages would start at the same point and fan out, displaying a drastically different angle of incline. This makes it imperative to utilise a voltage source that is as low as possible in order to create as power-efficient logic as possible. However, it should be noted that certain trade-offs must be taken into consideration, as this logic style does not demonstrate the highest level of operational speed when compared to other conventional dynamic logic styles used in.

4.2.3 Area

Results

The area designated for static and dynamic logic can respectively be observed in Tables 4.1 and 4.2. The tables presented below contain the logic gates used to create the pipelined 4-bit adder. The calculation of the area deviates from the conventional method, which is typically based on a layout. A more time-efficient method was utilised in order to obtain an rough approximation of the area in question. The method is based on calculating the sum of the area of each electrical component utilised.

As illustrated in Tables 4.1 and 4.2, a clear difference emerges in the area used in the dynamic and static logic styles implementations, with the former exhibiting a significantly reduced in area. A notable exception to this is the observation that the single-ended dynamic logic is marginally larger than the differential logic. This can be attributed to the suboptimal method of calculating the sum. Should the implementation of XORs be used for the purpose of calculating the sum, it is estimated that the area used will decrease significantly.

Discussion

It is imperative to note that the tables presented offer estimates of the area utilised, rather than the actual area used. This is due to the fact that the total size would increase as a result to factors such as routing. In dynamic logic, it is necessary for the data signal and the two clock phases (precharge and evaluate) to be routed to each logic cell, in contrast static logic, where only the data signal needs to be routed. In this particular instance, the presence of a third signal, designated as the delay signal, introduces a significant challenge in the creation of compact logic when only two distinct metal layers are utilised. It is therefore apparent that the dynamic logic styles appear to be quite optimistic in size, despite this not being the case. This is due to the fact that their routing is far more difficult than that of the static logic styles, particularly in combination with the delay cells.

The most significant contribution to the size of the 2RLL logic can be attributed to the resistor size. It is imperative that this size is considered in conjunction with power consumption and operating speed. In this example, two distinct instances of 2RLL logic were created (a fast and slow version), which demonstrate a substantial difference in surface area utilised (682 μm^2). The rationale behind this occurrence is mostly attributable to the implementation of different resistors in the 2RLL instances.

It is imperative to devise a layout in order to determine whether routing in dynamic logic will exert a considerable influence on the area utilised. It is hypothesised that the area of the proposed dynamic logic will be comparable in size, and might even surpass that of 2RLL. This is particularly relevant in the context of single-ended dynamic logic, where the delay signals run perpendicular to the output signals, in contrast to the differential logic, where they are in parallel with each other.

Table 4.1: Component sizes for 2RLL configurations

2RLL (fast)				**2RLL (slow)**			
Component	**Size**	**Amount**	**Total**	**Component**	**Size**	**Amount**	**Total**
Adder	271.6	4	1086.4	Adder	384	4	1536
Inverter	10.9	7	76.3	Inverter	13.8	7	96.6
XNOR	50.6	2	101.2	XNOR	90.6	2	181.2
NAND	21.7	3	65.1	NAND	24.6	3	73.8
NOR	14.5	2	29	NOR	16.2	2	32.4
Flip-flop	195.4	8	1563.2	Flip-flop	224.4	8	1795.2
NAND	21.7	8	173.6	NAND	24.6	8	196.8
Inverter	10.9	2	21.8	Inverter	13.8	2	27.6
Total = 2649.6 μm^2				**Total = 3331.2** μm^2			

Table 4.2: Component sizes for Dynamic logic configurations

Dynamic Single-ended				**Dynamic differential**			
Component	**Size**	**Amount**	**Total**	**Component**	**Size**	**Amount**	**Total**
Adder	169.8	8	1358.4	Adder	203.4	4	813.6
Inverter	7.6	12	91.2	XOR/XNOR	28.8	2	57.6
NAND (3 in)	8.8	4	35.2	AND/NAND	16.4	3	49.2
NAND (2 in)	8.2	3	24.6	OR/NOR	27.6	1	27.6
NOR (4 in)	9.4	1	9.4	Delay Cell	11.5	6	69
NOR (3 in)	9.4	1	9.4				
Flip-flop	28	8	224	Flip-flop	107	8	856
Inverter	28	2	56	Inverter/Buffer	42	2	84
				Delay Cell	11.5	2	23
Delay Cell	10.3	22	226.6				
Total = 1809 μm^2				**Total = 1669.6** μm^2			

Chapter 5

Discussion

The purpose of this chapter is to synthesise the various elements of the thesis. Firstly, a overview will be given discussing the goal of this thesis. The subsequent section will present and discuss the results obtained. In addition, the limitations of the present study will be outlined, with a view to identifying avenues for future research on the subject. In conclusion, the preceding discussion will be synthesised in order to draw all relevant points together.

5.1 Overview of Research Objectives

The primary research objective was to develop logic that was more energy efficient in unipolar technologies, with a particular focus on indium gallium zinc oxide. It should be noted, however, that this was not the sole objective. Speed and area were also considered to be two other factors in the validation process. It is important to note that these could be compromised within a realistic expectation, given that the objective of the main goal was to enhance power efficiency. The premise underlying the notion that dynamic logic is potentially more efficient than static logic is based on the premise that dynamic logic is capable of avoiding leakage when compared to static logic styles.

5.2 Summary of the Results

As demonstrated in Figure 4.5, a substantial enhancement in power efficiency of more than 100 times at 10 kHz can be achieved through the utilisation of dynamic logic styles. This increase in power efficiency is less pronounced in scenarios where the frequency clock frequency of the logic system is increased. However, it must be noted that this approach of logic creation is not without its drawbacks. It is evident that the maximum operating frequency of dynamic logic is lower in comparison to that of static equivalents.

A secondary disadvantage may be associated with the area (Tables 4.1 and 4.2). While this preliminary result appears to be promising, a closer inspection reveals that only component size is specified. It is evident that the routing of logic cells has not been factored into the size calculation, as no layout has been created. This will drastically increase the area of dynamic logic cells due to the fact that dynamic logic encounters significantly more difficulties with regard to routing than static logic. It is hypothesised that this area will approach that of static logic

and potentially even surpass it.

When comparing the speed of the proposed dynamic logic the delay cell has a delay of 250 ns on 3 V. This is less then 7 times slower then the simulated speed of 2 stage resistive load in [12], and less then 4 times slower if compared on the same voltage of 3 V. The 4-bit adder achieving a frequency of 140 kHz, which is an acceptable speed for specific applications that require more energy efficiency and less speed.

5.3 Drawbacks of the Dynamic Logic Style

It is important to note that this particular style of dynamic logic is not without its drawbacks. One such example is the fact that the full potential speed of the dynamic logic gates is not utilised. This is attributable to the fact that these are controlled by the delay cells, which is considerably slower than the logic itself.

The subsequent section is more specifically related to the differential logic implementation. This implementation utilises a delay cell that continuously leaks current until the corresponding logic cell completes its computation. Should the signal of a drastically faster path be utilised to power subsequent cells, this may result in a considerable increase in power consumption through the leakage of the delay cells. The present issue may be resolved by the fabrication of a cell that is not leaking, resulting in the fact that the slowest past does not need to be used to connect logic gates.

This does make differential logic a safe alternative to create logic in unipolar technologies. It is important to note that this is not the case in the single-ended implementation, since the delay cell does not verify whether the logic computation has been completed or not. In order to ensure the absence of any logical errors, it is important that in the course of simulation, all the delays of each stage are compared with the slowest path. However, this does not guarantee a circuit that is entirely free of glitches, as process variation has the capacity to alter the timings of the delay and logic cells, with the potential to introduce new errors.

Finally, it is important to note that a strict management of the clock is required for both dynamic logic styles proposed in this thesis. The attainment of low-power results is only possible under the condition that the precharge and evaluate phases do not overlap, which is not a simple task. Consequently, there is an absolute absence of leakage currents.

5.4 Limitations of the Study

The main constraint that will be discussed pertains to the absence of a layout. This issue gives rise to a plethora of problems. Firstly, it is not possible to make a meaningful area comparison. The rough area estimation provided is derived from the electrical components area used to create the logic circuits. As has been previously discussed, dynamic logic styles require a significantly greater area for the routing of a precharge, evaluation, and delay signal. It is necessary for these to be connected to every logic cell. This is a particularly problematic aspect in the researched IGZO-based technologies, where only two routing metal layers are provided, usually resulting in a lower density designs compared to conventional technologies.

A further potential outcome of the layout that has not been considered in this thesis is an

application in physical context. This makes it possible to perform measurements on a variety of circuit types in order to determine the viability of the proposed logic styles in real-life scenarios. It is important that this is designed in conjunction with a Monte Carlo simulation, which is also absent from this thesis. These simulations facilitate the creation of a more detailed view of the delay cells and potentially provide extra layers of validation in a glitch-free circuit during the taping out process.

5.5 Future Work

It is evident that a considerable amount of work remains to be completed in the context of future tasks. Thus far, only a limited selection of logic gates and flip-flops have been created. It is recommended that a more comprehensive library of cells be designed. Furthermore, an alternative design for the delay cells should be considered. This design has been developed to be power-efficient; however, its processing speed is not optimal. It is to be expected that a faster equivalent will result in higher performance.

Furthermore, it is important to ensure the robustness of the design created. It is therefore essential to analyse the noise margin and robustness of the created cells. This is particularly relevant for the delay cell, given that the design of robust dynamic logic cells has been extensively documented.

A further consideration is the flow of using this logic. Typical standard logic and most dynamic logic styles can be automaticity synthesized and routed using industry standard tools. While this styles with delay cells is more custom and is recommended to be designed and verified manually with analogue design software, it might be possible to create software to design circuits faster.

Chapter 6

Conclusion

The objective of this thesis was to devise a more energy-efficient alternative to logic computation on flexible substrates. Presently, the utilisation of static logic styles is predominant, and these styles have been observed to exhibit a considerable degree of leakage power, thereby contributing to elevated power consumption. The proposal set out a methodology for overcoming this problem, which involved the implementation of dynamic logic. This logic comprises two distinct phases, the implementation of which is shown to result in a reduction of leakage. The implementation of this process is not without its challenges, particularly in the context of unipolar technologies. Issues such as premature node discharge and charge sharing can impede progress, as their mitigation requires complex solutions.

Two distinct solutions have been proposed for the implementation of dynamic logic in unipolar technologies. These solutions are characterised as single-ended and differential logic. The implementations under consideration made use of an buffer cell and a NAND gate for the single-ended and differential logic respectively. These delay cells designs were based on a bootstrapped pseudo-CMOS inverter. In addition, certain adjustments were made to the logic cells themself to remove the leaking current. The dynamic logic is based on standard CMOS dynamic logic where an additional control unit and a keeper transistor was added. The control unit is equipped with a precharge gate, which precharges the output to ground. In addition, it contains a pass gate that enables the propagation of signals. In the case of the keeper transistor, the addition of a capacitor is used to temporarily assist the transistor in enabling itself with the delay signal, thereby increasing the driving power of the output.

The proposed delay cell in question demonstrated it generates consistent delays, with certain exceptions. Firstly, it was necessary to apply relatively consistent loads to each output. If inconsistent loads are applied, less optimal delay patterns may occur. Secondly, it is important to consider the potential impact of process variation on the delay, which is created by the buffer cells. This was not generally regarded as an issue; however, the possibility of a too-rapid delay was taken into consideration. The speed of the delay cell was also analysed in function of the voltage, which can be increased by increasing the supply voltage of the delay cells.

The proposed dynamic logic demonstrates the potential for significantly enhanced energy efficiency compared to static alternatives. However, it should be noted that this is not without its limitations. The primary limitation observed is the fact that the maximum operational speed of the dynamic logic is considerably lower than that of static implementations. This issue can

be mitigated to a certain extent by utilising a higher voltage supply; however, this approach also results in an increase in the power consumption. A second drawback is the area utilised to achieve this logic style. The area of logic circuits depend on the size of the electrical component and the number of electrical component used. In this instance, dynamic logic uses significantly less area and components compared to 2RLL logic. However, if the routing is also taken into consideration, it is hypothesised that the area of the dynamic logic style will significantly increase and potentially exceed that of the static logic styles.

Nevertheless, it must be acknowledged that the proposed thesis and logic styles can be used for certain applications where the main limitation is power consumption and not speed. Certain applications do not require a high computational speed but need to be optimized for power consumption because they are battery powered. These applications can benefit by using the proposed dynamic logic to significantly extend their battery life.

Bibliography

[1] S. Wagner and S. Bauer, "Materials for stretchable electronics," *MRS Bulletin*, vol. 37, no. 3, pp. 207–213, 3 2012. [Online]. Available: https://www.cambridge.org/core/journals/mrs-bulletin/article/materials-for-stretchable-electronics/2A6311118C9C5B9D6B0DD94C92822344

[2] K. Myny, "The development of flexible integrated circuits based on thin-film transistors," *Nature Electronics 2017 1:1*, vol. 1, no. 1, pp. 30–39, 1 2018. [Online]. Available: https://www.nature.com/articles/s41928-017-0008-6

[3] G. H. Gelinck, H. E. A. Huitema, E. V. Veenendaal, E. Cantatore, L. Schrijnemakers, J. B. Van Der Putten, T. C. Geuns, M. Beenhakkers, J. B. Giesbers, B. H. Huisman, E. J. Meijer, E. M. Benito, F. J. Touwslager, A. W. Marsman, B. J. Van Rens, and D. M. De Leeuw, "Flexible active-matrix displays and shift registers based on solution-processed organic transistors," *Nature Materials*, vol. 3, no. 2, pp. 106–110, 1 2004. [Online]. Available: https://www.nature.com/articles/nmat1061

[4] E. Cantatore, T. C. Geuns, G. H. Gelinck, E. Van Veenendaal, A. F. Gruijthuijsen, L. Schrijnemakers, S. Drews, and D. M. De Leeuw, "A 13.56-MHz RFID system based on organic transponders," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 1, pp. 84–92, 1 2007.

[5] N. Karaki, T. Nammoto, H. Ebihara, S. Inoue, and T. Shimoda, "43.1: A Flexible 8-bit Asynchronous Microprocessor Based on Low-Temperature PolySilicon (LTPS) TFT Technology," *Sid Symposium Digest of Technical Papers*, vol. 36, 5 2005.

[6] "EUROPRACTICE — Flexible Electronics." [Online]. Available: https://europractice-ic.com/technologies/flexible-electronics/

[7] T. Zhang, N. Liu, J. Xu, Z. Liu, Y. Zhou, Y. Yang, S. Li, Y. Huang, and S. Jiang, "Flexible electronics for cardiovascular healthcare monitoring," *The Innovation*, vol. 4, no. 5, p. 100485, 9 2023. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC10440597/

[8] I. The Western Design Center, "W65C02S 8-bit Microprocessor," The Western Design Center, Inc., Tech. Rep., 4 2022. [Online]. Available: www.WDC65xx.com

[9] H. Celiker, A. Sou, B. Cobb, W. Dehaene, and K. Myny, "Flex6502: A Flexible 8b Microprocessor in 0.8m Metal-Oxide Thin-Film Transistor Technology Implemented with a Complete Digital Design Flow Running Complex Assembly Code," *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, vol. 2022-February, pp. 272–274, 2022.

[10] J. S. Kim, J. H. Jang, Y. D. Kim, J. W. Byun, K. Han, J. S. Park, and B. D. Choi, "Dynamic Logic Circuits Using a-IGZO TFTs," *IEEE Transactions on Electron Devices*, vol. 64, no. 10, pp. 4123–4130, 10 2017.

[11] J. Uyemura P., *CMOS LOGIC CIRCUIT DESIGN*. New York: Kluwer Academic Publishers, 2001. [Online]. Available: www.circuitmix.com

[12] H. Celiker, A. Sou, W. Dehaene, and K. Myny, "Two-Stage Resistor-Load Logic for Digital Applications on Flexible Substrates," in *FLEPS 2021 - IEEE International Conference on Flexible and Printable Sensors and Systems*. Institute of Electrical and Electronics Engineers Inc., 6 2021.

[13] D. Bode, C. Rolin, S. Schols, M. Debucquoy, S. Steudel, J. Genoe, and P. Heremans, "Noise-Margin Analysis for Organic Thin-Film Complementary Technology," *IEEE Transactions on Electron Devices*, vol. 57, p. 201, 5 2010.

[14] T. C. Huang, K. Fukuda, C. M. Lo, Y. H. Yeh, T. Sekitani, T. Someya, and K. T. Cheng, "Pseudo-CMOS: A design style for low-cost and robust flexible electronics," *IEEE Transactions on Electron Devices*, vol. 58, no. 1, pp. 141–150, 1 2011.

[15] M.-J. Spijkman, K. Myny, E. Smits, P. Heremans, P. Blom, and D. Leeuw, "Dual-Gate Thin-Film Transistors, Integrated Circuits and Sensors," *Advanced materials (Deerfield Beach, Fla.)*, vol. 23, pp. 3231–3242, 5 2011.

[16] PragmatIC, "PragmatIC RFID." [Online]. Available: https://www.pragmaticsemi.com/nfc-connect/pr1301/

[17] Y. Hara, T. Kikuchi, H. Kitagawa, J. Morinaga, H. Ohgami, H. Imai, T. Daitoh, and T. Matsuo, "IGZO-TFT technology for large-screen 8K display," *Journal of the Society for Information Display*, vol. 26, no. 3, pp. 169–177, 3 2018.

[18] M. Naqi, S. C. Jang, Y. Cho, J. M. Park, J. O. Oh, H. Y. Rho, H.-S. Kim, and S. Kim, "Low temperature processed, highly stable CMOS inverter by integrating Zn-ON and tellurium thin-film transistors," *Journal of Information Display*, vol. 24, no. 3, pp. 199–204, 2023. [Online]. Available: https://doi.org/10.1080/15980316.2023.2174195

[19] A. Singh, S. Sikarwar, A. Verma, and B. Chandra Yadav, "The recent development of metal oxide heterostructures based gas sensor, their future opportunities and challenges: A review," *Sensors and Actuators A: Physical*, vol. 332, p. 113127, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924424721005926

[20] Y. Han, J. Seo, D. Lee, and H. Yoo, "IGZO-Based Electronic Device Application: Advancements in Gas Sensor, Logic Circuit, Biosensor, Neuromorphic Device, and Photodetector Technologies," *Micromachines*, vol. 16, p. 118, 5 2025.

[21] Z. Mohammad, S. Stefano, R. Ganesh, S. Smout, W. Myriam, M. Kris, and C. Eugenio, "A flexible ECG patch compatible with NFC RF communication," *NPJ Flexible Electronics*, vol. 4, no. 1, 2020.

[22] Y. Li, B. Zeng, Y. Yang, H. Liang, Y. Yang, and Q. Yuan, "Design of high stability thin-film transistor biosensor for the diagnosis of bladder cancer,"

Chinese Chemical Letters, vol. 31, no. 6, pp. 1387–1391, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1001841720301625

[23] A. Gefen, "How medical engineering has changed our understanding of chronic wounds and future prospects," *Medical Engineering & Physics*, vol. 72, pp. 13–18, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1350453319301663

[24] H. Çeliker, K. Myny, and W. Dehaene, "Thin-Film Logic Families for VLSI Circuits on Plastic," Ph.D. dissertation, IMEC, 2025.

[25] V. Friedman and S. Liu, "Dynamic Logic CMOS Circuits," *IEE Journal of Solid-State Circuits*, vol. SC-19, no. 2, pp. 263–266, 4 1984.

[26] C. H. Cheng, S. C. Chang, J. S. Wang, and W. B. Jone, "Charge sharing fault detection for CMOS domino logic circuits," in *Proceedings 1999 IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (EFT'99)*, 1999, pp. 77–85.

[27] R. K. Montoye, "Special Correspondence Design-Performartce Trade-Offs in CMOS-Domino Logic," *IEEE JOURNAL OF SOLID-STATECIRCUITS*, no. 2, p. 21, 1986.

[28] V. Kursun and E. G. Friedman, "Domino logic with variable threshold voltage keeper," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 11, no. 6, pp. 1080–1093, 12 2003.

[29] P. Srivastava, A. Pua, and L. Welch, "Issues in the design of domino logic circuits," in *Proceedings of the 8th Great Lakes Symposium on VLSI (Cat. No.98TB100222)*, 1998, pp. 108–112.

[30] R. H. Krambeck, C. M. Lee, and H.-F. Law, "High-speed compact circuits with CMOS," *IEEE Journal of Solid-State Circuits*, vol. 17, no. 3, pp. 614–619, 1982.

[31] N. F. Goncalves and H. De Man, "NORA: a racefree dynamic CMOS technique for pipelined logic structures," *IEEE Journal of Solid-State Circuits*, vol. 18, no. 3, pp. 261–266, 1983.

[32] K. Chu and D. Pulfrey, "A comparison of CMOS circuit techniques: Differential Cascode Voltage Switch logic versus conventional logic," *Solid-State Circuits, IEEE Journal of*, vol. SC-22, pp. 528 – 532, 5 1987.

[33] D. Geng, Y. F. Chen, M. Mativenga, and J. Jang, "30 μm -Pitch Oxide TFT-Based Gate Driver Design for Small-Size, High-Resolution, and Narrow-Bezel Displays," *IEEE Electron Device Letters*, vol. 36, no. 8, pp. 805–807, 2015.

[34] B. Tiwari, P. Bahubalindruni, A. Santos, A. Santa, C. Figueiredo, M. Pereira, R. Martins, E. Fortunato, and P. Barquinha, "Low-Voltage High-Speed Ring Oscillator with a-InGaZnO TFTs," *IEEE Journal of the Electron Devices Society*, vol. PP, p. 1, 5 2020.

[35] M. Velazquez Lopez, B. Linares-Barranco, J. Lee, H. Erfanijazi, A. Patino-Saucedo, M. Sifalakis, F. Catthoor, and K. Myny, "A tunable multi-timescale Indium-Gallium-Zinc-Oxide thin-film transistor neuron towards hybrid solutions for spiking neuromorphic applications," *Communications Engineering*, vol. 3, no. 1, p. 102, 2024. [Online]. Available: https://doi.org/10.1038/s44172-024-00248-7

Appendix A

Clarifying Schematics

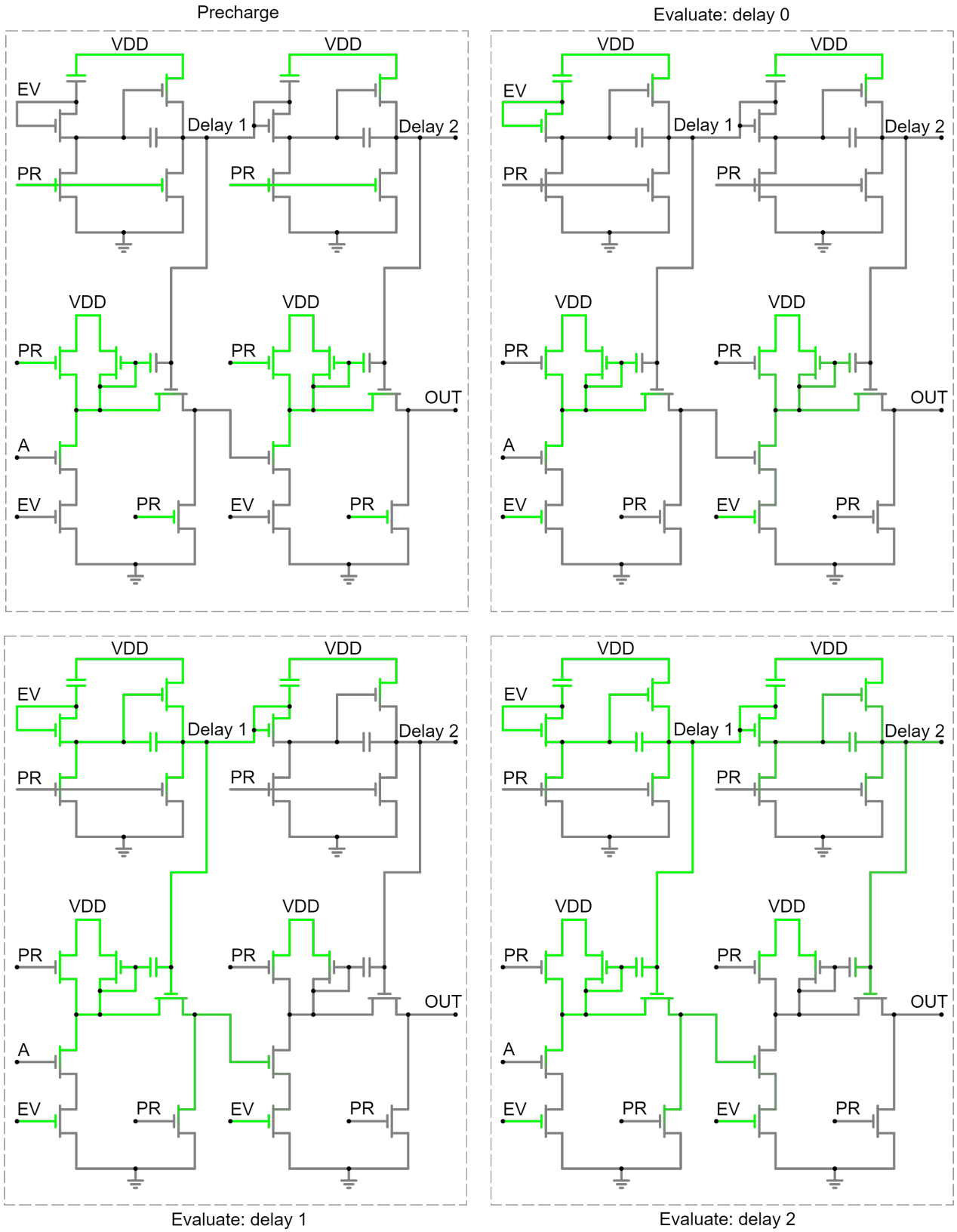

Figure A.1: Example functioning of single-ended cascading logic with delay cells

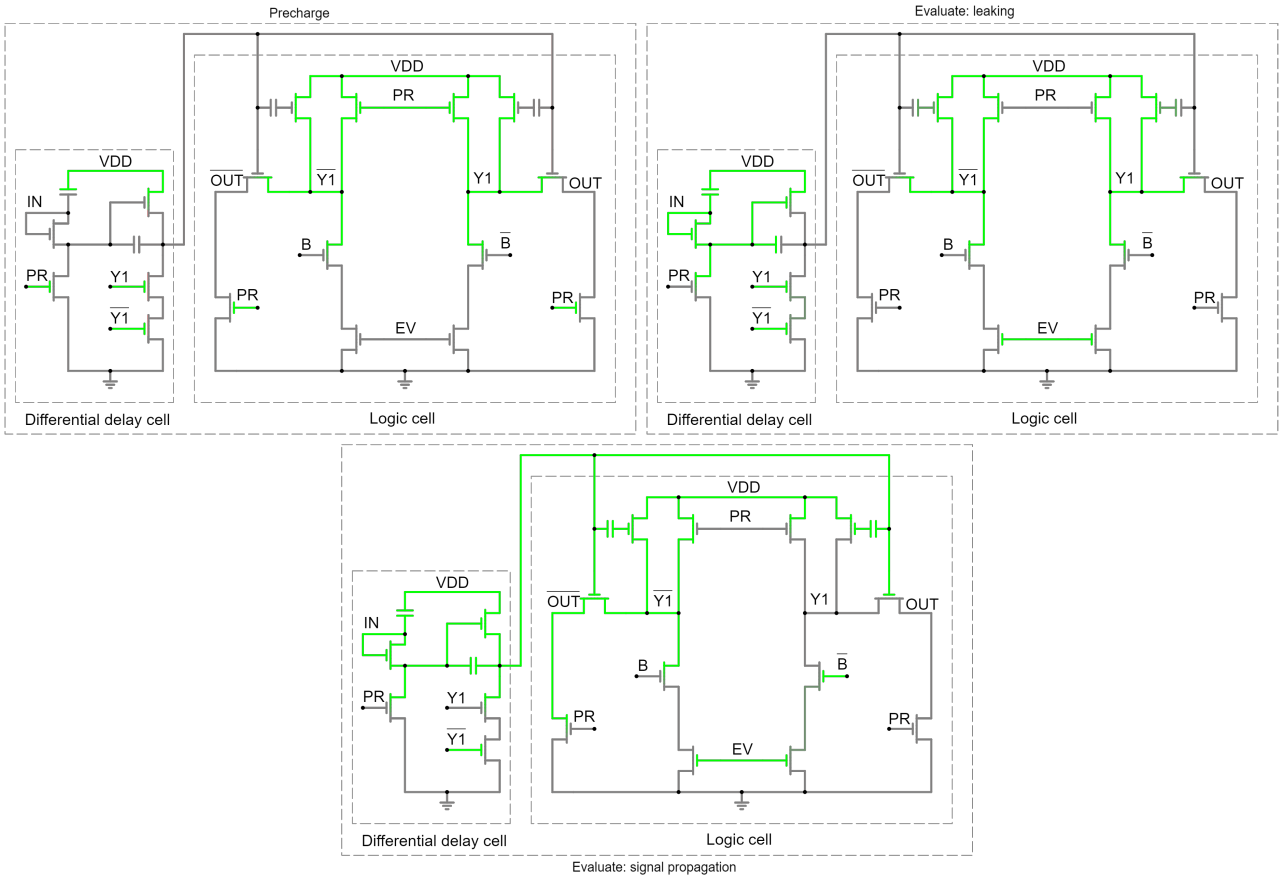

Figure A.2: Example functioning of differential cascading logic with delay cells

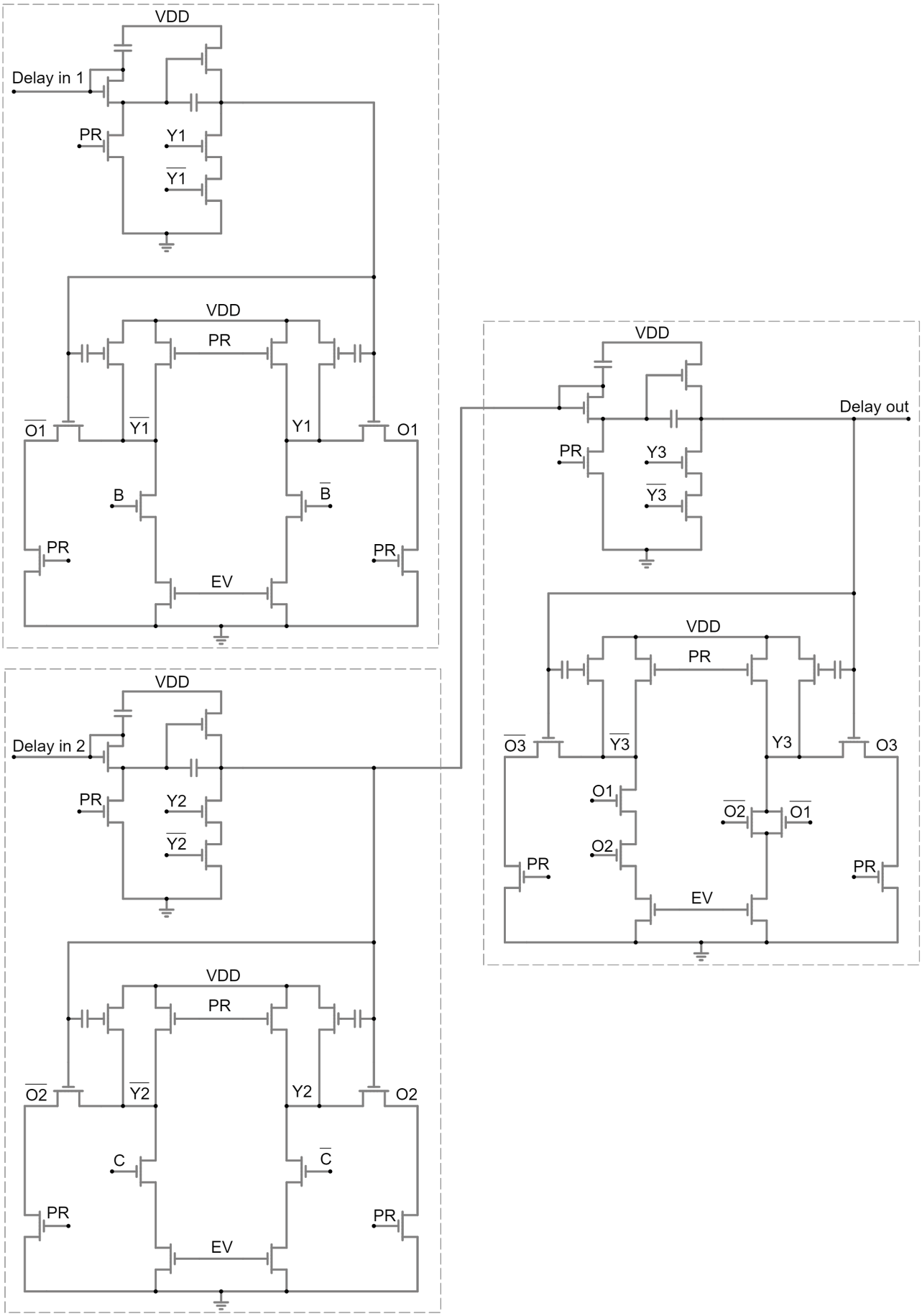

Figure A.3: Example of connecting differential logic cells