Politecnico di Torino

Politecnico di Torino
Master Degree in Electronic Engineering

All-Digital Noise-Tolerant Voltage-Level Detector for Dynamic Configuration of SRAM Read/Write-Assist Circuits

Candidate:
Claudia Pecorella

Supervisors:
Prof. Guido Masera
Eng. Basma Hajri
Eng. Keith Bowman
Eng. Stefano Marinaci

Academic Year 2024-2025

Politecnico di Torino
Dipartimento di Elettronica e Telecomunicazioni

Politecnico di Torino

Politecnico di Torino

Master Degree in Electronic Engineering

All-Digital Noise-Tolerant Voltage-Level Detector for Dynamic Configuration of SRAM Read/Write-Assist Circuits

Candidate:
Claudia Pecorella

Supervisors:
Prof. Guido Masera
Eng. Basma Hajri
Eng. Keith Bowman
Eng. Stefano Marinaci

Academic Year 2024-2025

A

Abstract

This thesis presents the design of a fully digital Voltage Level Detector (VLD) architecture for dynamic read and write assist configuration in SRAM arrays.

In advanced CMOS technologies, aggressive voltage scaling is essential for energy efficiency but compromises memory stability. To enhance reliability, assist circuits are commonly employed, whose activation depends on the supply voltage level. Traditionally, this level is monitored using analog voltage comparators, which suffer from poor scalability, high area and power overhead, and sensitivity to process variations.

The proposed VLD architecture replaces analog comparators with a fully digital solution composed of three key components: a Dynamic Variation Monitor (DVM), which continuously evaluates the timing margin of a tunable delay path to indirectly sense voltage changes; an Auto-Calibration Circuit, which automatically adjusts the delay configuration to align the DVM's sensitivity with a specific voltage threshold, ensuring accurate detection across varying operating conditions; a Control Block that validates voltage transitions through a dual-signal mechanism. Two VLD instances, each calibrated to a specific threshold, enable dual-threshold detection for assist activation.

The architecture is modular, synthesizable, and achieves up to 90% area and latency reduction, supporting robust and energy-efficient integration in modern System-on-Chip (SoC) platforms.

Keywords: Adaptive Circuit, Auto-Calibration, Dynamic Variations, Noise Resilience, SRAM Assist Circuit, Timing Errors, Voltage Controller, Voltage Variation.

Contents

List of Figures

List of Tables

Acronyms

ACD Adaptive Clock Distribution

ACC Access Control Configuration

ADC Analog-to-Digital Converter

ASIC Application-Specific Integrated Circuit

BIST Built-In Self-Test

BL Bit Line

BLB Bit Line Bar

BTI Bias Temperature Instability

CMOS Complementary Metal-Oxide-Semiconductor

CPU Central Processing Unit

CPUSS Central Processing Unit Subsystem

DAB Dynamic Adaptive Biasing

DDC Droop Detection Controller

DLDO Digital Low-Dropout Regulator

DNL Differential non-linearity

DPM Digital Power Meter

DSP Digital Signal Processor

DVF Dynamic Voltage and Frequency

DVFS Dynamic Voltage-Frequency Scaling

DVM Dynamic Variation Monitor

ECU Error Control Unit

EDS Embedded Error-Detection Sequential

FF Flip Flop

FPGA Field-Programmable Gate Array

FSM Finite State Machine

FTP File Transport Protocol

GAAFET Gate-All-Around Field-Effect Transistor

HCI Hot Carrier Injection

HLOS High-Level Operating System

IIR Infinite Impulse Response

LDO Low-Dropout Regulator

LSB Least Significant Bit

MIS Multiple-Input Switching

MSB Most Significant Bit

NMOS N-type Metal-Oxide-Semiconductor

NOP no-operation

PCGS Proactive Clock-Gating System

PDN Power Delivery Network

PLL Phase-Locked-Loop

PMIC Power Management Integrated Circuit

PMOS P-Channel Metal-Oxide-Semiconductor

PMU Performance Monitoring Unit

PVT Process, Voltage, and Temperature

RDD Remote Droop Detector

SAR Successive Approximation Register

SNM Static Noise Margin

SoC System-on-Chip

SRAM Static Random Access Memory

TDC Time-to-Digital Converter

TDE Tunable-Delay Elements

TLD Tunable-Lenght Delay

TRC Tunable Replica Circuits

ULL Universal Logic Line

UVFR Unified Voltage and Frequency Regulator

UVM Universal Verification Methodology

VCG Voltage Clock-Gating

VLD Voltage Level Detector

VPB VCG Performance Buffer

WL Word Line

6T 6-transistors

Introduction

In modern SoC architectures, the continuous scaling of Complementary Metal-Oxide-Semiconductor (CMOS) technologies has enabled increasingly compact and energy-efficient designs. However, this evolution has also introduced significant challenges in ensuring reliable memory operation, particularly in Static Random Access Memory (SRAM) arrays operating under aggressive voltage scaling. To maintain data integrity in such conditions, assist circuits are commonly employed to enhance read and write margins. These circuits are typically activated based on the supply voltage level at which the memory operates; in conventional designs, analog voltage comparators are used to monitor this voltage level and trigger the appropriate assist mechanisms. While effective in principle, analog comparators suffer from several drawbacks: they are difficult to scale, consume significant area and power, and are highly sensitive to Process, Voltage, and Temperature (PVT) variations. These limitations hinder their integration in advanced digital flows and motivate the need for fully digital, robust, and adaptive alternatives.

Motivation

Starting from this context, the need emerges for a new class of voltage monitoring solutions that can overcome the constraints of analog comparators while enabling precise and efficient control of assist circuits. This motivates the development of a fully digital Voltage Level Detector (VLD), designed to detect voltage threshold crossings in real time and to dynamically configure assist strategies based on actual operating conditions. Such a system must ensure low area and power overheads, high noise resilience, and compatibility with standard digital design flows. By eliminating the dependency on analog components, a digital VLD offers improved scalability, portability across technology nodes, and greater flexibility in system-level power management.

Building on this observation, attention turned to existing digital monitoring systems such as the Dynamic Variation Monitor (DVM) [2] [9], originally designed to detect timing margin violations in real time. The DVM operates by monitoring the propagation delay of a tunable path and asserting an error signal when the timing margin becomes negative, typically due to voltage droops or frequency shifts. While its primary purpose was to support adaptive clocking and error prevention mechanisms, the DVM's sensitivity to voltage-induced delay variations revealed its potential as a voltage detection primitive. This observation led to the idea of repurposing the DVM as the core sensing element in a fully digital VLD.

To transform the DVM into a reliable voltage detection system, it was integrated with two additional components: an Auto-Calibration Circuit [3] and a dedicated Control Block. The Auto-Calibration Circuit dynamically adjusts the configuration of the tunable delay elements to ensure that the timing margin is zero at a specific voltage threshold. This calibration process allows the DVM to respond accurately to voltage changes across different operating conditions, without requiring manual tuning or external references. The Control Block, on the other hand, interprets the DVM output in conjunction with a voltage-level indication signal provided by a voltage controller. The voltage controller traditionally interfaces with the Power Management Integrated Circuit (PMIC) to set the desired voltage value, providing a digital indication of the intended operating region, used by the Control Block to validate voltage transitions. This dual-signal validation suppresses false activations due to transient noise or rail oscillations, ensuring robust and energy-efficient operation. By combining these elements, the system becomes capable of detecting when the supply voltage crosses predefined thresholds and of activating assist mechanisms accordingly, enabling a fully digital, scalable, and noise-resilient solution for voltage-aware memory adaptation. By calibrating the DVM to specific voltage thresholds and integrating it with additional control logic, it becomes possible to detect when the supply voltage crosses predefined levels and to trigger assist mechanisms accordingly. This approach not only leverages the existing infrastructure but also enables a scalable and synthesizable solution for voltage-aware memory adaptation.

Design Approach

The design of the proposed Voltage Level Detector was carried out through a structured methodology that combined architectural reuse with original development. The initial phase involved a critical analysis of the state of the art, aimed at understanding existing voltage detection and assist control techniques. This study enabled the identification of limitations in conventional analog solutions and guided the definition

of design requirements for a fully digital alternative.

To build upon existing infrastructure, the project began with the simulation and review of the pre-existing DVM testbench. This included analysis of RTL components, waveform inspection and behavioral model review.

Following this, a feasibility study was conducted to define the requirements of the new architecture. Through brainstorming and iterative refinement, a new concept was proposed to repurpose the DVM as a voltage sensing element, integrating it with an auto-calibration mechanism and a control logic block for assist activation. The architectural definition was supported by block diagrams and signal flow analysis to ensure clarity and consistency.

The RTL implementation phase involved the description of a new control block in SystemVerilog, System Registers generation and definition of their programming sequence, including the modification of certain functionalities within the original system to support the extended voltage detection and assist control logic. A linear testbench was developed to verify functional correctness, which allowed for rapid functional verification and debugging in isolation. As the design matured and the blocks were integrated into a complete system, a more comprehensive simulation environment was required. To this end, a pre-existing UVM-based testbench was adapted and extended to support the new architecture, enabling more realistic and modular verification of the full system behavior.

Throughout the project, technical documentation was produced to support design decisions, and regular presentations were held with internal teams to gather feedback and ensure alignment with system-level goals. The final implementation reflects a balance between reuse of validated components and the development of new logic tailored to the specific needs of voltage-aware assist control in advanced SRAM arrays.

Thesis Structure

This thesis is structured to guide the reader through the conceptual foundations, technical implementation, and practical implications of the proposed Voltage Level Detector architecture. The content is organized as follows:

- **Chapter 1** provides a comprehensive overview of the state of the art in voltage droop detection and variation-tolerant circuit design. It outlines the limitations of existing analog and hybrid solutions, highlighting the need for scalable, low-

overhead digital alternatives. The chapter also introduces the key challenges posed by dynamic variations in supply voltage, temperature, and process parameters, which motivate the development of resilient monitoring systems.

- **Chapter 2** focuses on the DVM, a central building block of the proposed architecture. It describes its operating principles, its integration into adaptive systems, and the role of the auto-calibration circuit in ensuring accurate threshold detection across a wide range of operating conditions. The chapter also presents two representative applications, the Adaptive Clock Distribution (ACD) and a DVM-based Security Monitor, which demonstrate the versatility of the DVM in both performance optimization and system-level security.

- **Chapter 3** presents the design and implementation of the VLD, detailing its architectural components, control logic, and simulation results. It explains how the VLD extends the DVM framework to support dual-threshold voltage detection, enabling dynamic assist configuration in SRAM arrays. Particular attention is given to the VLD Control Block, which ensures robust and noise-resilient operation through a dual-signal validation mechanism. Then a critical evaluation of the architecture's strengths and limitations, discussing its potential for integration into future low-power, high-performance SoC platforms and outlining directions for further research and development.

- **Appendix A** provides a detailed review of SRAM read and write assist strategies, contextualizing their relevance to the proposed solution.

The system was developed with the support of Qualcomm Technologies, Inc., which provided access to advanced technologies, tools, software, and technical guidance for implementation. Due to confidentiality restrictions, specific technical details cannot be disclosed, but every effort has been made to present the architecture and methodology as clearly and completely as possible.

Chapter 1

State of Art

1.1 Introduction

The continuous scaling of CMOS technologies has enabled the integration of increasingly complex and high-performance SoC architectures. However, this progress has also introduced significant challenges related to power integrity, timing reliability, and energy efficiency. Among these, dynamic variations in supply voltage, temperature, and device aging have emerged as critical factors that can severely impact circuit performance and reliability.

This chapter provides a comprehensive overview of the state-of-the-art techniques developed to address these challenges. It begins by identifying the primary sources of **dynamic variations** in modern microprocessors, including voltage droops, thermal fluctuations, transistor aging, and parasitic effects such as cross-coupling capacitance and multiple-input switching. These phenomena introduce timing uncertainty and degrade energy efficiency, necessitating robust mitigation strategies.

Subsequently, the chapter explores a range of circuit-level techniques designed to tolerate or recover from timing violations induced by dynamic variations. These include sensor-based adaptive voltage and frequency control, tunable replica circuits with error recovery and embedded error-detection sequential logic. Each approach is analyzed in terms of its operating principles, benefits, and limitations.

A particular focus is placed on **voltage droop** detection and mitigation, a key area of research due to the increasing sensitivity of scaled technologies to transient supply fluctuations. The chapter categorizes existing droop detection solutions into analog, hybrid, and all-digital architectures, highlighting their respective trade-offs in terms of accuracy, integration complexity, and scalability.

This foundational review sets the context for the subsequent chapters, which delve into the design and implementation of an **all-digital Voltage Level Detector (VLD)**. The proposed solution aims to overcome the limitations of analog and hybrid approaches by leveraging a fully synthesizable, technology-portable architecture capable of robust droop detection in advanced technology nodes. A central component of this architecture is the **DVM**, a digital sensing circuit that continuously evaluates timing margins by monitoring the propagation delay of a tunable path. By detecting deviations caused by voltage droops or frequency shifts, the DVM enables real-time error signaling and supports adaptive mechanisms such as clock gating or frequency scaling. Its integration allows for fine-grained, cycle-level detection of critical timing violations without the need for analog components, making it ideal for modern low-power, high-performance SoC environments.

1.2 Dynamic Variation Sources

Dynamic variations in modern microprocessors arise from multiple physical and operational phenomena that affect timing and energy efficiency. The primary sources include:

1. **V_{dd} Droops:** These are caused by abrupt changes in switching activity, which generate large current transients in the power delivery network. The magnitude and duration of V_{dd} droops are influenced by the interaction of capacitive and inductive parasitics at the board, package, and die levels, along with the dynamic behavior of current demand. V_{dd} droops exhibit both high-frequency (fast-changing) and low-frequency (slow-changing) components and can occur both locally and globally across the die;

2. **Temperature Variations:** Temperature fluctuations in microprocessors arise from a combination of workload intensity, environmental conditions, and the thermal dissipation capabilities of the packaging and cooling solutions. These variations manifest both spatially and temporally across the die. Spatially, non-uniform power densities lead to the formation of *thermal gradients* and localized *hotspots*, where certain regions of the chip operate at significantly higher temperatures than others. Temporally, temperature changes occur over relatively slow time scales due to the high *thermal time constants* of silicon and packaging materials, which govern the rate at which heat is conducted and dissipated.

   Hotspot formation is particularly critical in high-performance cores and cache regions, where sustained activity can elevate local temperatures, exacerbating leakage currents and accelerating aging mechanisms such as *Bias Temperature*

Instability and *Electromigration*. These effects degrade transistor performance and interconnect reliability over time. Furthermore, temperature-induced variations in carrier mobility and threshold voltage directly impact circuit delay and power consumption, leading to timing uncertainty and reduced energy efficiency.

Effective thermal management and accurate modeling of temperature-dependent behavior are therefore essential for ensuring robust operation. Techniques such as dynamic thermal throttling, temperature-aware floorplanning, and on-die thermal sensors are commonly employed to mitigate the adverse effects of temperature variations in modern SoC designs.

3. **Transistor Aging:** The drive current of transistors degrades gradually over time due to prolonged gate bias and elevated temperature conditions, a phenomenon commonly referred to as transistor aging. Two primary mechanisms contribute to this degradation: *Bias Temperature Instability (BTI)* and *Hot Carrier Injection (HCI)*.

   BTI, particularly Negative BTI in PMOS transistors and Positive BTI in NMOS transistors, occurs when a constant voltage is applied to the gate terminal under elevated temperatures. This leads to the generation of interface traps and charge trapping in the gate oxide, resulting in an increase in threshold voltage (V_{th}) and a corresponding reduction in drive current. HCI, on the other hand, arises when high-energy carriers are injected into the gate oxide during switching events, causing permanent damage to the oxide-semiconductor interface.

   As aging progresses, the cumulative delay increase can lead to timing violations, particularly in critical paths with tight margins. Therefore, accurate modeling and mitigation of aging-induced delay shifts are essential for ensuring long-term reliability in scaled CMOS technologies.

4. **Cross-Coupling Capacitance:** Cross-coupling capacitance arises from the capacitive interaction between adjacent interconnect wires in densely packed digital layouts. As technology scales and metal pitches shrink, the lateral spacing between wires decreases, leading to a significant increase in parasitic coupling capacitance. This phenomenon becomes particularly pronounced in advanced CMOS nodes, where interconnect dimensions are comparable to the gate lengths of transistors.

   The presence of cross-coupling capacitance introduces *data-dependent delay variations* in signal propagation. When adjacent wires switch in opposite directions (i.e., differential switching), the effective capacitance seen by the victim

wire increases, resulting in a longer delay. Conversely, when neighboring wires switch in the same direction (i.e., common-mode switching), the coupling effect is partially canceled, leading to a shorter delay. These variations can cause timing uncertainty and degrade signal integrity.

Moreover, cross-coupling can induce transient noise, known as *crosstalk*, which may lead to spurious switching or logic errors if not properly mitigated. The impact of crosstalk is exacerbated by long interconnect lengths, high switching activity, and low supply voltages. To address these issues, physical design techniques such as shielding, wire spacing optimization, and layer assignment are employed. Additionally, timing analysis tools incorporate crosstalk-aware delay models to ensure accurate prediction of path delays under worst-case switching scenarios.

5. **Multiple-Input Switching (MIS):** MIS refers to the phenomenon where multiple inputs of a logic gate transition simultaneously or nearly simultaneously. This simultaneous switching can significantly affect the gate's propagation delay due to increased internal capacitive loading and dynamic current demands. Unlike single-input transitions, MIS events can cause a nonlinear increase in delay, as the gate must handle multiple charging and discharging paths concurrently. The delay impact of MIS is influenced by several factors, including the direction of transitions (rising or falling), the number of inputs switching, and the relative timing (skew) between input transitions. MIS-induced delay variations are highly localized and occur on fast time scales, making them difficult to capture using traditional static timing analysis. These variations can lead to timing violations in critical paths, especially in high-speed designs where timing margins are tight.

The dynamic variation sources described above, ranging from voltage droops and temperature fluctuations to aging and parasitic effects, pose significant challenges to timing reliability and energy efficiency in modern SoCs. To address these issues, a variety of circuit-level techniques have been developed that aim to detect, tolerate, or recover from timing violations induced by such variations. The following section presents an overview of these techniques, highlighting their operating principles, benefits, and limitations.

1.3 Circuit Techniques for Dynamic Variation Tolerance

This section presents a detailed overview of three key **circuit-level techniques** developed to mitigate the impact of dynamic variations in modern microprocessors. These methods aim to reduce or eliminate the conservative guardbands traditionally

used to ensure timing correctness under worst-case conditions. The discussion is based on the methodologies described in the paper *"Circuit Techniques for Dynamic Variation Tolerance"* by Bowman et al. [1], which outlines the principles, benefits, and trade-offs of each approach. The following subsections will examine the operation of adaptive sensor-based control, tunable replica circuits with recovery, and embedded error-detection sequential circuits.

1.3.1 Sensors with Adaptive Voltage and Frequency

As shown in Fig. 1.1, for a generic microprocessor with N pipeline stages, to mitigate the impact of slow-changing global variations such as temperature fluctuations, supply voltage droops, and transistor aging, modern designs incorporate *on-die sensors* coupled with Dynamic Voltage and Frequency (DVF) control circuits. These sensors monitor environmental and operational parameters in real time and provide feedback to adaptive control logic that adjusts the clock frequency, supply voltage, or body bias accordingly.

Thermal sensors are strategically placed at local hotspots on the die to detect worst-case temperature conditions. V_{dd} droop sensors monitor global supply voltage fluctuations that can affect timing across all circuit paths. Aging sensors track the degradation of critical paths due to prolonged stress and recovery cycles, enabling the system to compensate for performance loss over time.

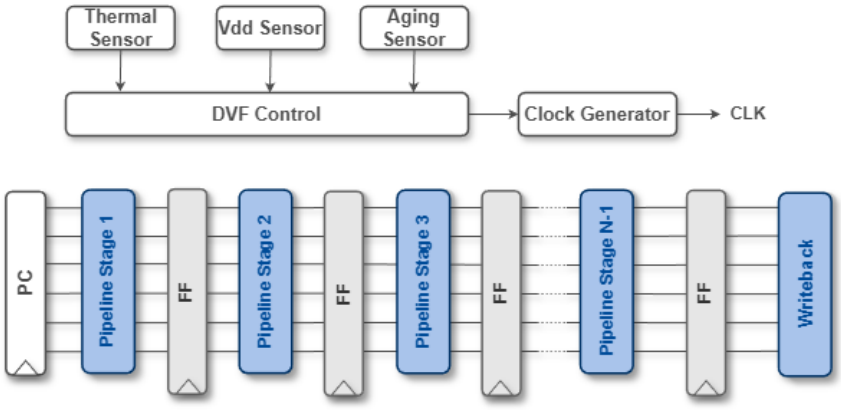

Figure 1.1: Sensors with DVF control circuits [1].

By leveraging these sensor data, adaptive control circuits can dynamically reduce guardbands traditionally used to account for worst-case scenarios. This allows the system to operate at higher average clock frequency (f_{clk}) or, alternatively, reduce energy consumption by lowering the supply voltage (V_{dd}) during favorable conditions.

The result is improved performance and energy efficiency without compromising reliability.

However, this approach has some limitations: the **response time** of the sensors and control logic restricts their effectiveness to slow-changing variations. Fast, high-frequency V_{dd} droops or localized path-level variations may not be detected or mitigated in time, potentially leading to timing violations. Moreover, both analog and digital sensors, along with adaptive circuits, necessitate considerable **calibration time** for each die, resulting in increased testing resources. Therefore, while on-die sensors and adaptive circuits are effective for global variation tolerance, they must be complemented by other techniques to ensure robust operation under all conditions.

1.3.2 Tunable Replica Circuits with Error Recovery

To further reduce the guardbands associated with dynamic variations, Tunable Replica Circuits (TRC) can be combined with error recovery mechanisms to enable aggressive frequency scaling without compromising correctness. Unlike techniques that rely solely on sensor feedback and control loops, this approach eliminates the response-time limitations by directly detecting timing errors as they occur.

TRCs are strategically placed adjacent to each pipeline stage and are calibrated to mimic the delay characteristics of local critical paths. These circuits toggle every clock cycle and are designed using a variety of logic elements, such as inverters, NANDs, NORs, pass gates, and repeated interconnects, to accurately reflect the sensitivity of real paths to variations in supply voltage and temperature. The TRCs operate under the same clock and voltage conditions as the functional logic, allowing them to capture fine-grained delay shifts due to global and local variations.

The TRCs in this scheme (Fig. 1.2) drives the Embedded Error-Detection Sequential (EDS), which monitor for late signal transitions indicative of timing violations. When a timing error is detected, the EDS generates an error signal that is propagated to the writeback stage and an Error Control Unit (ECU). The ECU invalidates the erroneous data, initiates a replay of the affected instruction, and temporarily reduces the clock frequency to ensure correct execution under persistent variation conditions. Once the replay completes, the ECU restores the original frequency and resumes normal operation.

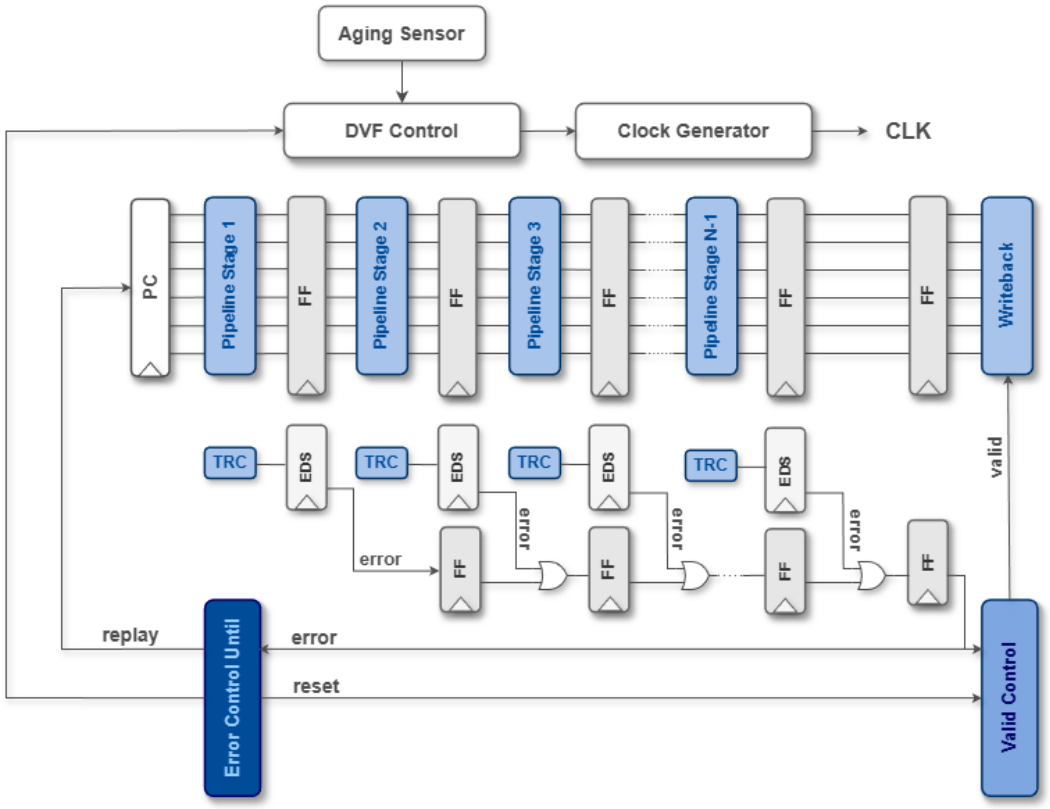

Figure 1.2: TRC for error prediction and recovery [1].

This technique enables the system to operate beyond conservative timing margins by allowing occasional timing errors and correcting them dynamically. As a result, it provides a robust and energy-efficient solution for tolerating both slow and fast dynamic variations, without the need for excessive guardbands or complex analog sensing infrastructure.

However, the TRC with recovery has several limitations, including **false error** signaling and unnecessary recovery activation. It cannot detect dynamic path-level variations and requires a delay guardband to ensure it operates slower than critical paths under varying voltage and temperature conditions. While digital TRCs are simpler than the analog sensors discussed in Section 1.3.1 , **post-silicon tuning** increases testing costs. Additionally, the recovery mechanism needs an extra **stabilization stage** in the pipeline to handle the one-cycle latency of error signal propagation, ensuring instruction validity before state commit.

1.3.3 Embedded Error-Detection with Recovery

EDS circuits represent a robust approach to eliminate f_{clk} guardbands by detecting and correcting timing errors directly within the critical paths of a microprocessor. Unlike tunable replica circuits or sensor-based techniques, EDS circuits are *embedded in the actual datapaths*, as illustrated in Fig. 1.3, enabling real-time monitoring of timing violations caused by both global and local dynamic variations, including supply voltage (V_{dd}) droops and temperature fluctuations.

Each EDS circuit monitors for late signal transitions and generates an error signal when a timing violation is detected. These error signals are aggregated through an OR-tree structure and fed into a ECU. Upon detection of an error, the ECU initiates a recovery sequence by replaying the affected instruction and temporarily reducing f_{clk} to ensure correct execution under persistent variation conditions. Once the recovery is complete, the system resumes normal operation at the target frequency. This technique not only removes the need for conservative guardbands but also allows for performance scaling beyond the worst-case critical path delay by exploiting path-activation probabilities. Since the slowest paths are often rarely activated, the system can operate at higher frequencies most of the time, correcting errors only when those paths are exercised. This results in significant throughput gains compared to conventional designs.

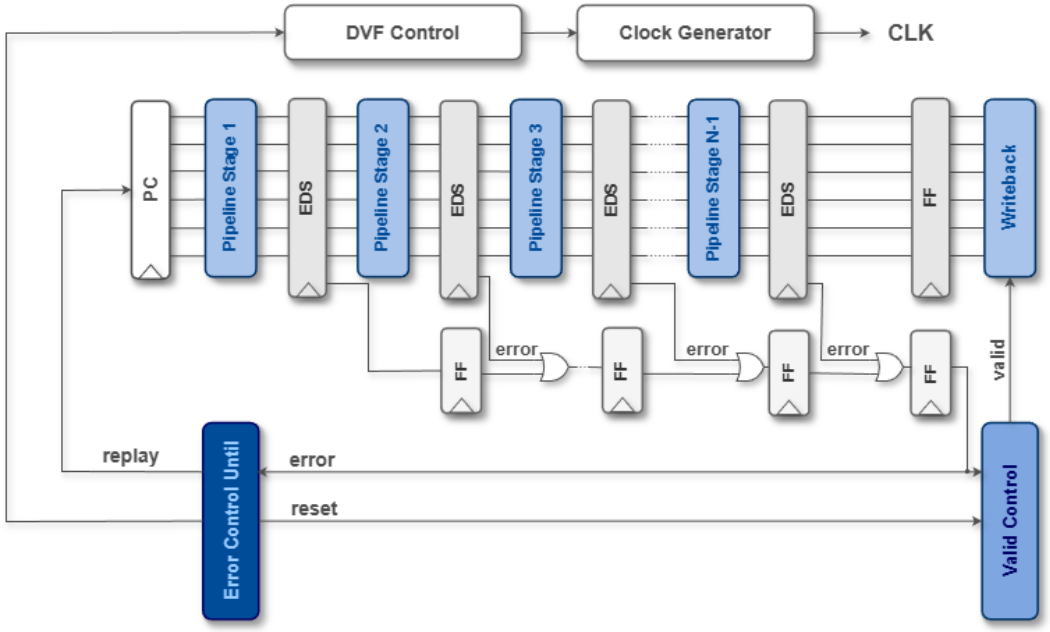

Figure 1.3: EDS circuit with error recovery [1].

However, the use of EDS circuits introduces certain trade-offs. These include additional clock energy overhead, the need for duty-cycle control to maintain a stable error-detection window, and the requirement for a stabilization pipeline stage to handle error propagation latency. Moreover, the minimum delay constraint becomes more stringent, although this is mitigated in modern microarchitectures that favor shallow pipelines for energy efficiency. Overall, embedded EDS circuits with recovery offer a powerful and scalable solution for dynamic variation tolerance, enabling both performance enhancement and energy savings.

1.4 Voltage Droop Detection and Mitigation

Among all the sources of dynamic variation previously discussed, voltage droop stands out as one of the most challenging to manage. These transient events can lead to immediate timing violations, making voltage droop one of the most critical and difficult issues to address in power-aware digital design. As a result, the development of effective droop detection and mitigation strategies has become a key area of research.

Existing droop detectors can be broadly categorized into three types: *analog*, *digital*, and *hybrid*.

- **Analog droop detectors** directly compare the supply voltage to a reference. However, their integration is often hindered by the large area of analog components, the need for dedicated analog supplies, and routing constraints between the sensing point and the comparator, which can limit the maximum operating frequency.
- **Digital droop detectors** are composed mostly of standard cells, making them highly compatible with logic synthesis flows and easily portable across process nodes. Their compact footprint, scalability, and ease of integration make them particularly attractive for modern SoC design, where flexibility and design automation are key. As a result, digital detectors are increasingly favored in advanced technology nodes.
- **Hybrid solutions** aim to combine the strengths of both analog and digital approaches, often leveraging analog sensing with digital processing to balance accuracy, speed, and integration efficiency.

The following sections provide an overview of representative solutions across all existing categories — analog, digital, and hybrid — highlighting their architectural innovations, performance trade-offs, and suitability for integration in advanced SoC environments.

1.4.1 Analog Solutions

A fundamental building block in analog voltage detection systems is the operational amplifier (Op-Amp) based comparator. Operating in open-loop configuration, the comparator evaluates the voltage difference between its non-inverting and inverting inputs and produces a rail-to-rail digital output. When the input voltage at the non-inverting terminal exceeds that at the inverting terminal, the output transitions to a high logic level; otherwise, it switches low,as shown in Fig. 1.4. This binary behavior enables the comparator to function as a threshold detector, converting continuous analog signals into discrete digital events.

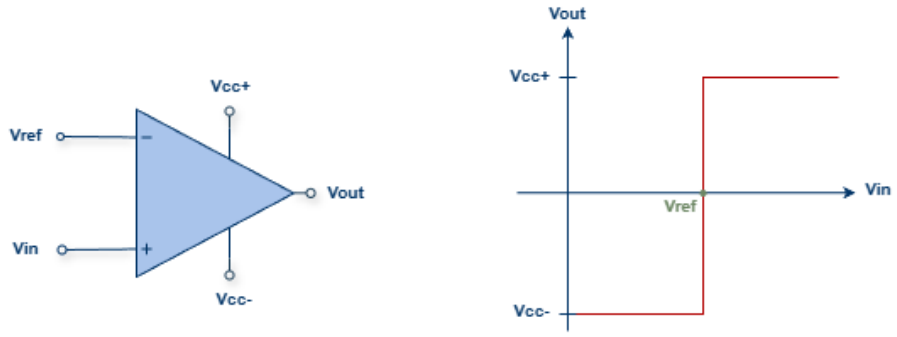

Figure 1.4: Op-Amp comparator ideal behavior.

A notable difficulty in the application of Op-Amp comparators is their heightened sensitivity to noise. Since the output can only be one of two digital values, the presence of noise may lead to an unstable output, resulting in several undesired transitions over short time frames. This instability can be problematic in numerous applications, particularly in industrial contexts. In such scenarios, employing an Op-Amp comparator with **hysteresis** can effectively mitigate the issue. This configuration is beneficial in preventing instability that arises from noise and capacitive interference between the output and input. This alternative layout is shown in Fig. 1.5: the output is connected to the non-inverting input via a resistor to restrict the current appropriately, introducing a positive feedback. This way, once a transition begins, the input must have a significant reversal in order for a reverse transition to occur.

The hysteresis can be determined as follows:

$$\Delta V = (V_{cc+} - V_{cc-}) \cdot \frac{(R_1 + R_2)}{R_1}$$

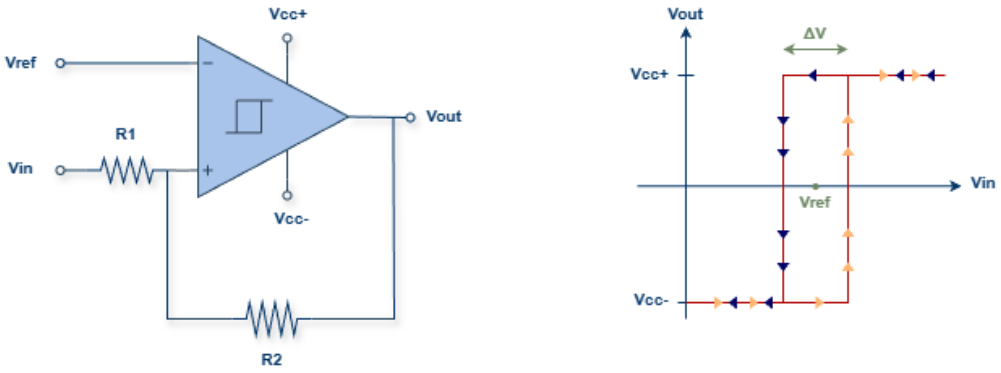

Figure 1.5: Op-Amp comparator with hysteresis ideal behavior.

Despite their simplicity and fast response, analog comparators present several drawbacks when integrated into modern SoC designs. First, they typically require dedicated analog supply rails and biasing circuits, which complicate power distribution and increase design complexity. Second, analog components are inherently more sensitive to process variations and mismatch, often requiring careful layout and post-silicon calibration to ensure consistent performance across dies and operating conditions. Moreover, their relatively large area and limited scalability make them less attractive in advanced CMOS nodes, where digital logic benefits from aggressive scaling. Finally, the analog signal routing between the sensing point and the comparator can introduce parasitic effects and delay, limiting the maximum operating frequency and reducing the effectiveness of droop detection in high-speed applications. For these reasons, analog comparators are often complemented or replaced by digital or hybrid solutions in cutting-edge voltage monitoring systems.

1.4.2 All-Digital Solutions

While analog droop detectors have traditionally offered robustness against process, voltage, and temperature variations, they suffer from several limitations that hinder their integration in advanced technology nodes.

First, they typically require dedicated, often bulky, circuit blocks that consume significant silicon area. Their reliance on analog reference voltages and biasing cir-

cuits introduces additional design complexity and necessitates separate analog power domains, which complicates integration in predominantly digital systems. Furthermore, the physical separation between the voltage sensing point and the main comparator introduces routing delays, which in turn limit the maximum operating frequency and responsiveness of the system. These constraints make analog solutions less scalable and less compatible with the design methodologies of modern system-on-chip architectures. In contrast, fully digital droop detection techniques offer a more compact, synthesizable, and technology-portable alternative.

Among the various approaches explored in the literature to mitigate voltage droops, one notable solution is presented by Kalyanam et al. [10] that propose a fully digital Proactive Clock-Gating System (PCGS) implemented in a 7 nm Hexagon™ Digital Signal Processor (DSP), designed to mitigate voltage droops by predicting power transients based on **microarchitectural activity**. The system integrates a Digital Power Meter (DPM) to estimate per-cycle power using event counters and pre-characterized weights, and a Voltage Clock-Gating (VCG) circuit that employs a configurable digital second-order Infinite Impulse Response (IIR) filter to model the Power Delivery Network (PDN) impedance and predict voltage droops. When a predicted droop exceeds a programmable threshold, the VCG triggers clock gating to reduce current transients. The system also includes a Performance Monitoring Unit (PMU) and a VCG Performance Buffer (VPB) to monitor effective clock frequency and gating activity. Entirely composed of standard digital logic, the PCGS is synthesizable, portable across technology nodes, and achieves up to 10% higher effective frequency or 5% lower minimum operating voltage V_{min} with minimal performance impact.

Among these, the work by Bowman et al. [2] stands out as a foundational contribution. Their dynamically **Adaptive Clock Distribution (ACD)** system introduces a fully digital architecture that prevents timing-margin violations by suppressing clock edges in response to high-frequency voltage droops. The design integrates a tunable-length delay, a **DVM**, and a clock-gating mechanism.

Fig. 1.6 illustrates the basic structure of the DVM circuit, placed along critical paths to detect timing margin violations, that includes two TRC (to both monitor rising and falling transitions of the input signal) with corresponding driving and receiving Flip Flop (FF)s.

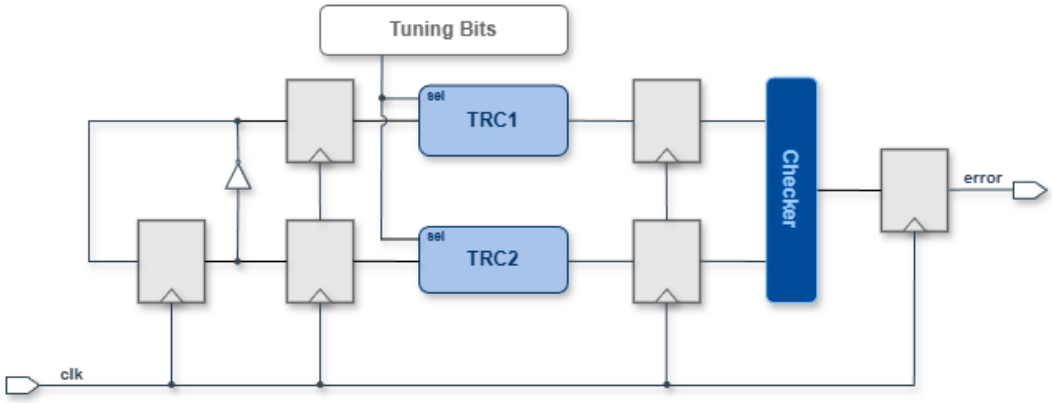

Figure 1.6: Simplified DVM architecture [2].

The tunable-length delay extends clock-data compensation, allowing a four-cycle response window during which the system can gate clocks to mitigate droop effects. This approach achieves up to 31% throughput improvement and 15% energy reduction at 0.6 V, while avoiding the complexity of analog components and error recovery logic. The system's scalability and robustness make it a compelling solution for droop tolerance in high-performance SoCs.

1.4.3 Hybrid Solutions

Beyond traditional Op-Amp comparators, more integrated hybrid solutions have been developed to address voltage droop detection in modern SoCs.

One such example is the system proposed by Dirican et al. [11], which introduces a Built-In Self-Test (BIST) architecture for **Digital Low-Dropout Regulator (DLDO)**. Traditional Low-Dropout Regulator (LDO)s are analog circuits that maintain a stable outout voltage (V_{out}) despite variations in load current or input voltage. They operate by continuously comparing the V_{out} to a reference V_{ref} and adjusting a pass transistor accordingly. While effective, analog LDOs can be challenging to scale in advanced CMOS technologies due to their reliance on analog components and limited digital configurability.

DLDOs differ from traditional analog LDOs by implementing voltage regulation through digital control logic. This makes them more scalable and easier to integrate in advanced CMOS technologies, especially in systems with multiple voltage domains or fine-grained power management. Building on this digital foundation, a hybrid architecture that combines analog sensing with digital processing is proposed to detect and quantify voltage droops. The system integrates a compact analog droop detector with a ramp generator and a 10-bit Successive Approximation Reg-

ister (SAR) Analog-to-Digital Converter (ADC), enabling accurate digitization of transient voltage variations. A leakage cancellation circuit is also included to enhance measurement reliability without relying on large analog buffers or external access. The regulation mechanism includes a clocked comparator, a bidirectional shift-register, and a power P-Channel Metal-Oxide-Semiconductor (PMOS) array. The V_{out} is continuously compared to the V_{ref}, and the shift-register adjusts the number of active PMOS transistors accordingly. This closed-loop control allows the system to maintain voltage stability while simultaneously monitoring droop events through the integrated BIST logic.

Among the various techniques developed to address voltage droop and improve power integrity in advanced SoCs, Unified Voltage and Frequency Regulator (UVFR) represents a promising approach. The concept of a UVFR was first introduced by Gangopadhyay et al. [12], who demonstrated an architecture with wide operating range, high current efficiency, and significant supply guardband reduction. This idea was later revisited and adapted by Liu et al. [13], who proposed a Universal Logic Line (ULL)-based UVFR system integrated into a self-calibrated digital framework, which tightly couples voltage and frequency regulation using a self-calibrated replica path oscillator. By tracking the critical path delay through a shared ULL structure and adjusting the supply voltage via a digitally controlled DLDO, the system eliminates conservative voltage margins and ensures timing safety during droop events.

Similarly, Jung et al. [14] propose a high-speed voltage droop detector capable of remote sensing, implemented in a 2nm Gate-All-Around Field-Effect Transistor (GAAFET) process. The proposed architecture separates the analog Droop Detection Controller (DDC) from compact **Remote Droop Detector (RDD)** embedded within Central Processing Unit (CPU) cores, enabling scalable multi-point sensing with minimal area overhead.

Tschanz et al. [15] propose a hybrid system for dynamic adaptation to temperature, voltage, and aging variations. The architecture integrates analog sensors for temperature and voltage droop detection, as well as on-die N-type Metal-Oxide-Semiconductor (NMOS) and PMOS body bias generators, with a **digital Dynamic Adaptive Biasing (DAB) controller**. The DAB controller uses a lookup table indexed by sensor outputs to determine optimal operating points for frequency, supply voltage, and body bias. A multi-Phase-Locked-Loop (PLL) clocking system enables rapid frequency switching without requiring PLL re-locking, while programmable logic ensures safe transitions between operating states. This hybrid approach allows the system to respond to fast voltage droops, thermal fluctuations, and long-term ag-

ing effects, achieving up to 22% frequency improvement through forward body biasing and maintaining performance over time. The design demonstrates the effectiveness of combining analog precision with digital flexibility for robust and energy-efficient operation in advanced CPU architectures.

These examples demonstrate how analog and digital components can be effectively combined to create robust, scalable, and self-contained solutions for on-chip power integrity monitoring in advanced SoC environments.

1.5 Conclusion

Among the circuit-level techniques discussed in Section 1.3, the *Embedded Error-Detection Sequential (EDS)* approach provides the conceptual foundation for the design of the DVM. EDS circuits enable real-time detection of timing violations by directly monitoring critical paths. This principle is inherited by the DVM, which evaluates timing margins through a tunable delay path and asserts an error signal when a violation is detected.

However, unlike traditional EDS-based resilient circuits, which rely on error recovery mechanisms such as instruction replay to maintain correct functionality, the DVM operates without any recovery logic. Classical resilient architectures embrace the occurrence of timing errors and focus on their correction, typically including complex control logic to isolate errors from the architectural state and initiate recovery procedures. While this approach is highly effective in mitigating the impact of dynamic variations, it introduces significant design complexity, area overhead, and power consumption.

In contrast, the DVM-based all-digital dynamically ACD architecture adopts a preventive strategy. By integrating a calibrated tunable replica circuit, the DVM detects impending timing margin degradation caused by high-frequency voltage droops. A tunable-length delay in the clock distribution path extends the clock-data compensation window, allowing sufficient time for the DVM to trigger proactive clock gating. This eliminates the clock edges that would otherwise lead to timing failures, thereby avoiding the need for post-error correction.

This shift from reactive to resilient enables a fully digital, lightweight, and synthesizable solution. The DVM and ACD architecture avoid the use of analog sensing components and complex recovery logic, making them highly portable across technology nodes and well-suited for integration in advanced SoC environments; the resulting design is not only scalable and robust but also energy-efficient and

performance-aware.

This all-digital methodology forms the conceptual basis for the work developed in this thesis, which aims to further explore and extend the potential of predictive, digitally controlled droop detection techniques. In particular, this work investigates the feasibility of reusing the DVM architecture to implement a **fully digital VLD**, capable of identifying and responding to voltage changes in real time. The proposed approach exemplifies a new paradigm in variation-aware design, where early detection and proactive mitigation replace traditional error correction strategies.

Chapter 2

Dynamic Variation Monitor

2.1 Introduction

Modern System-on-Chip (SoC) architectures are designed to deliver high performance and energy efficiency across a wide range of applications, from mobile devices to high-performance computing systems. As these systems become increasingly complex and power-constrained, ensuring reliable operation under dynamic and unpredictable workloads has become a critical design challenge.

One of the key reliability concerns in advanced SoCs is the stability of the power delivery network. As workloads shift rapidly, due to changes in instruction mix, thread activity, or peripheral interactions, the current demand of the processor cores can fluctuate significantly. These fluctuations can lead to transient disturbances in the supply voltage, commonly referred to as voltage droops. As already introduced in Section 1.2, the magnitude and duration of these droops are shaped by the interaction between capacitive and inductive parasitics at the board, package, and die levels, and the dynamic behavior of current consumption. Among these, high-frequency droops are particularly critical, as they can induce rapid and severe voltage fluctuations, leading to significant performance degradation.

While voltage droops are often the result of normal workload dynamics, they can also be intentionally induced as part of a **security attack**. Techniques such as *undervolting* or malicious *overclocking* can exploit the system's sensitivity to power fluctuations to induce faults, bypass security mechanisms, or extract sensitive information through side-channel analysis. These threats highlight the need for robust monitoring and mitigation strategies that go beyond traditional performance optimization.

This chapter presents the operating principles of the *Dynamic Variation Monitor (DVM)*, a fully digital circuit designed to detect timing margin degradation caused

by voltage droops. The DVM employs tunable replica paths to track critical path delays and assert an error signal when a timing violation occurs. The associated *Auto-Calibration circuit* is also described in detail.

Then two representative applications of the DVM are introduced. The first is the *Adaptive Clock Distribution (ACD)*, which integrates the DVM with a tunable-length delay and clock gating logic to proactively mitigate the impact of high-frequency droops by suppressing harmful clock edges before timing failures occur. The second is in the domain of system security, where the DVM can be used to continuously monitor timing margins and detect unsafe operating conditions, such as those caused by malicious voltage or frequency manipulation.

This preliminary study provides a detailed analysis of the DVM architecture and its integration into practical systems. Building on this foundation, the study introduces how the DVM can be repurposed as a voltage detection mechanism, forming the basis for the implementation of the proposed *Voltage-Level Detector (VLD)* described in the subsequent chapter.

2.2 DVM Architecture

The DVM system [3], illustrated in Fig. 2.1, is designed to operate as a real-time detector for potential timing violations caused by fluctuations in the voltage supply (V_{dd}) or variations in the clock frequency (f_{clk}). Its primary function is to monitor, at every clock cycle, the **timing margin** of a tunable-delay circuit strategically placed between a driving and a receiving Flip-Flop (FF). By doing so, the system is capable of identifying critical timing conditions that may compromise the correct operation of synchronous digital circuits. This mechanism enables early detection of marginal conditions, such as voltage droops or clock instabilities, which could otherwise lead to functional errors or data corruption.

In Tab. 2.1 the primary inputs and outputs of the system are reported, with a brief illustration of their operational roles.

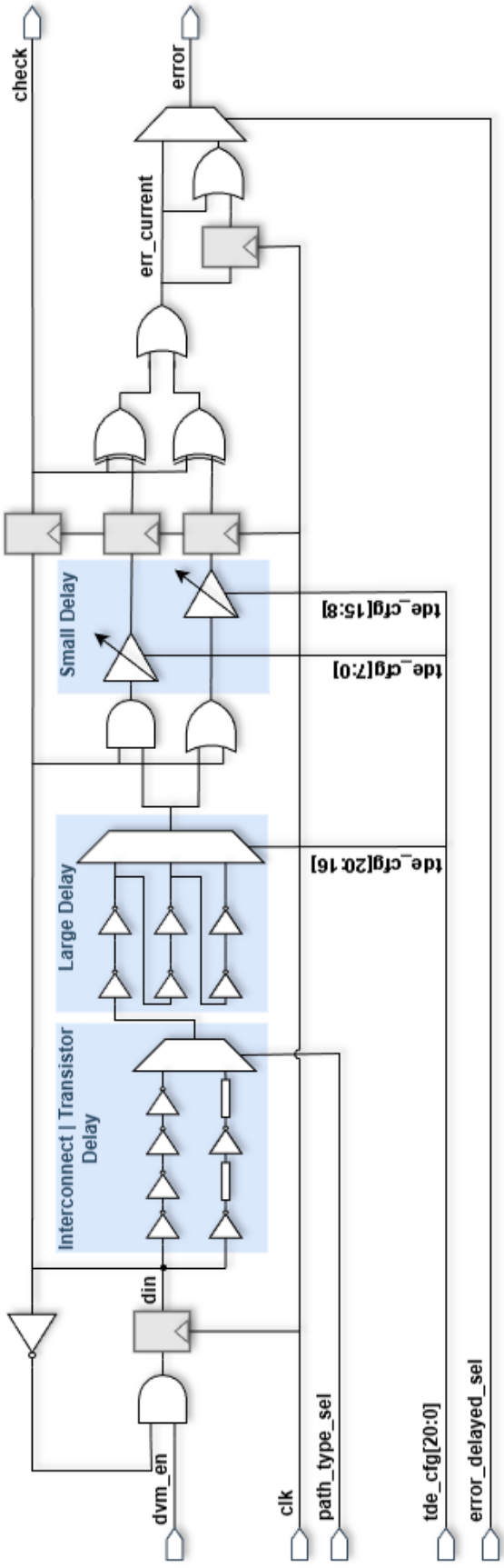

Figure 2.1: Dynamic Variation Monitor circuit schematic [3].

Signal Name	Direction	# Bits	Description
clk	Input	1	Clock signal
dvm_en	Input	1	DVM enable signal to toggle the driving FF
path_type_sel	Input	1	To select either transistor delay or interconnect delay components as the path type
tde_cfg	Input	21	Configuration bits for the tunable delay elements
error_delayed_sel	Input	1	Path selection bit for error signal generation
check	Output	1	Bit to identify the occurring transition, whether rising or falling
error	Output	1	DVM error signal

Table 2.1: DVM signals summary.

The tunable delay path consists of multiple adjustable delay branches, each calibrated to reflect the timing characteristics of the circuit's critical path. By selecting one of these branches through a programmable configuration input (*tde_cfg[20:0]*), the system can introduce a controlled delay to the propagation of the data signal from the driving FF to the receiving FF. This mechanism enables fine-grained tuning of the delay to match specific timing requirements, as described in [3].

The Tunable-Delay Elements (TDE) configuration bits consists of:

- Binary-coded large TDEs (*tde_cfg[20:16]*);
- Thermometer-coded small TDEs for separate tuning of the path for rising (*tde_cfg[7:0]*) and falling (*tde_cfg[15:8]*) input transitions.

Prior to utilization, the DVM must undergo an auto-calibration procedure, during which the configuration signals for the delay elements are set to obtain approximately zero timing margin. Additional **delay margin** is also provided in the tunable delay paths, since the critical path delay timings in the clocked circuit can vary based on process variations incurred during fabrication of the clocked circuit [3]. Once calibrated, the DVM is highly sensitive to variations in V_{dd} or f_{clk}: an unexpected drop in V_{dd} or an increase in f_{clk} can easily lead to a timing-margin failure, thereby

triggering the DVM error signal. This error signal serves as an alert to potential issues in the system's performance.

Over the course of development, several versions of the DVM have been explored, to improve its structure and usability.

Unlike the previous version (Fig. 2.1), which placed two flip-flops along the delay paths to sample intermediate signals, the updated design, shown in Fig. 2.2, simplifies the sampling stage by introducing a single flip-flop positioned after the combinational logic responsible for error detection, namely the structure composed of two XOR gates followed by an OR gate, which collectively evaluate the correctness of the delayed signal with respect to the expected transition. Although the internal detection mechanism will be described in detail later, it is worth noting that this architectural change not only centralizes the sampling process but also reduces the number of flip-flops required, leading to a more compact and efficient implementation.

Additionally, the *path_ type_ sel* signal, previously used to select between transistor and interconnect delay components, has been removed to simplify configuration.

A further enhancement is the integration of a Time-to-Digital Converter (TDC) on the *err_ pre* signal. This addition enables fine-grained quantification of timing margin violations by converting delay variations into a digital representation, improving observability and facilitating more precise analysis.

Referring to the latest DVM version, the following analysis illustrates the behavior of the circuit under different input transition conditions, specifically focusing on rising and falling edges of the *din* signal.

As shown in Fig. 2.2, the **detection logic** is composed of two stages. The first stage includes an AND and an OR gate positioned between the *Large Delay* and *Small Delay* sections. These gates process both the immediate and delayed versions of the *din* signal to generate two output signals: *rise* and *fall*, respectively associated with the monitoring of rising and falling transitions of the *din* signal.

More specifically, the *rise* signal is asserted when both the current and delayed versions of *din* are high, confirming a rising edge once the delayed signal has propagated through the tunable delay path. Conversely, the *fall* signal remains high as long as at least one of its inputs is high, and transitions to low only when both inputs are low, thus confirming a falling edge.

These two signals are then used in the second stage of detection logic, which consists of two XOR gates followed by a final OR gate. This block compares the *rise* and *fall* delayed signal with the current value of *din* to generate the signal

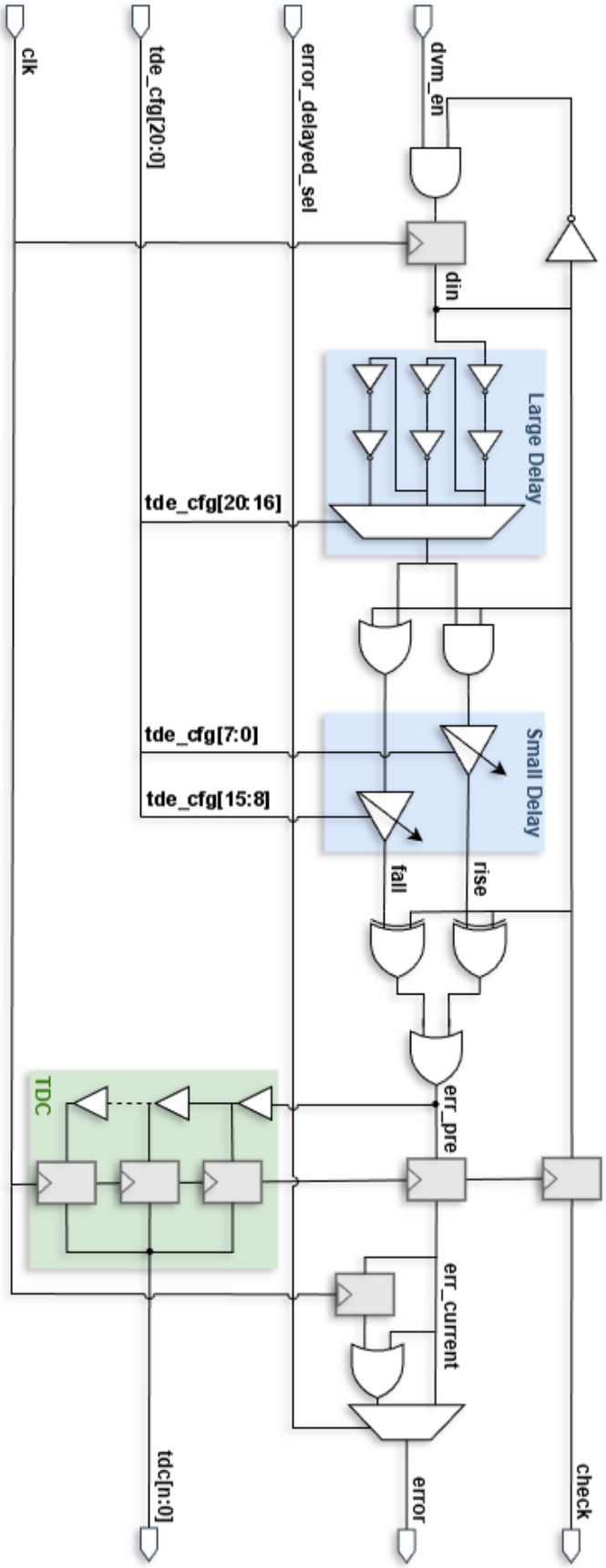

Figure 2.2: Dynamic Variation Monitor latest circuit schematic [4].

err_current. Each XOR gate checks for mismatches in rising and falling transitions, and the OR gate combines their outputs. If either XOR detects a discrepancy, *err_current* is asserted high, indicating that the delayed signal does not correctly reflect the expected transition.

Together, these two stages ensure robust detection of timing margin violations.

To illustrate the behavior of the system, we now analyze how the circuit responds when the *dvm_en* signal is asserted and the *din* input begins toggling at half the clock frequency (Fig. 2.3).

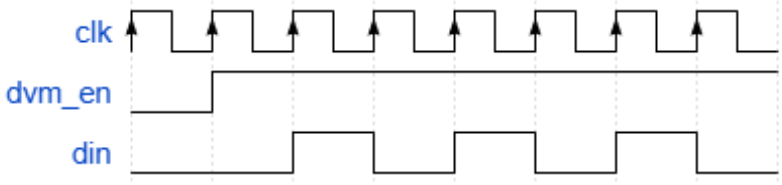

Figure 2.3: Toggling of *din* signal after DVM enabling.

For a rising *din* transition:

1. The signal will propagate quickly through the OR gate, setting its output (*fall*) to 1 regardless of the other input (coming from the tunable-delay path). On the other hand, the output of the AND gate (*rise*) will remain equal to 0 until the *din* signal propagates through the TDEs. As long as the output of the AND gate remains low, the *err_pre* signal will be held high, and it will only be reset to 0 once the delayed *din* signal reaches the AND gate through the configured delay path.

2. The second stage of the check logic, consisting in two XOR gates and an OR gate, will then compare the outputs of the first detection stage with the *din* signal: if they result to be equal (i.e., both logically high), then *err_current* is logically low.

Similarly, when having a falling *din* transition:

1. The signal will propagate rapidly through the AND gate, setting its output *rise* to 0; on the fall path, however, the OR output *fall* will remain asserted to 1 (due to the rising transition of the previous cycle) until is reset by the signal coming from the TDEs. As long as the output of the OR gate remains high, the *err_pre* signal will also be held high, and it will only be cleared once the

delayed *din* signal propagates through the TDEs and causes the OR output to transition to 0.

2. The check logic then compares the outputs of the first detection stage with *din*, and if they are both logically low, *err_current* is set to 0.

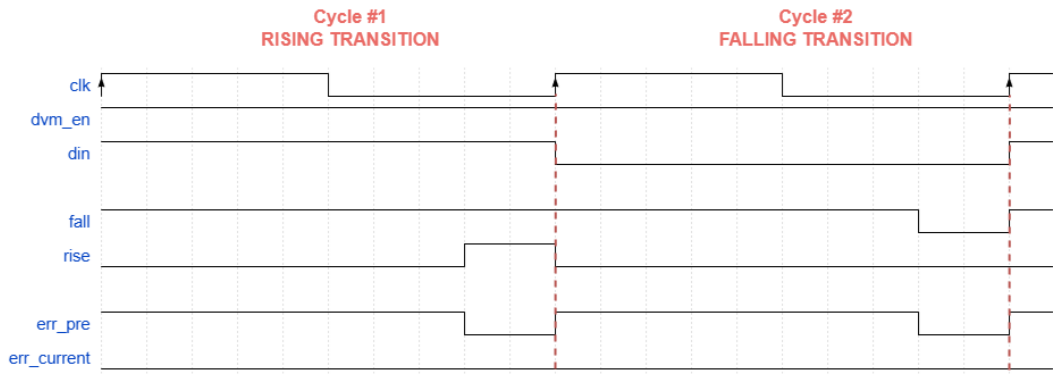

Figure 2.4: Detection of rising and falling transitions through delayed signal paths.

In both cases, if the delayed signal does not propagate fast enough through the tunable delay elements, its transition will not be completed by the next rising edge of the clock. As a result, the *rise* or *fall* signal will remain asserted, keeping the *err_pre* signal high. When this value is sampled by the receiving flip-flop at the next clock edge, the *error* output will also be set high, indicating a **timing-margin failure** due to excessive path delay relative to the clock period (T_{clk}).

To address potential **metastability** induced by late path delays, additional logic is provided to enhance error sampling accuracy during metastable conditions. Metastability can arise when a signal transition occurs too close to the clock edge of a flip-flop, violating setup or hold time constraints. In the context of the DVM, this situation may happen when the delayed *din* signal, propagating through the tunable delay elements, arrives at the receiving FF just as the clock edge triggers sampling.

Such timing violations can cause the FF to enter a metastable state, where its output is temporarily undefined or oscillates before settling to a stable logic level; this behavior can lead to incorrect or inconsistent sampling results.

To mitigate this, the design includes a mechanism that improves robustness against metastability: by asserting the *error_delayed_sel* control bit, the system modifies the generation of the final error signal: instead of relying solely on the current cycle's *err_current* signal, which may be affected by metastability, the error

output is computed as a logical-OR between *err_current* and its value from the previous clock cycle. This temporal redundancy ensures that transient metastable behavior does not cause the system to miss a genuine timing-margin failure, thereby increasing the reliability of voltage droop detection and adaptive response.

Refer to Fig. 2.5 for a complete visual representation of the described signal waveforms. These waveforms are the result of a simulation of the DVM system, conducted within a **UVM-based** verification environment. Since voltage variations cannot be directly modeled in standard UVM testbenches, timing violations are instead induced by increasing the clock frequency beyond nominal operating conditions. This approach effectively emulates the impact of voltage droops or other dynamic variations on circuit timing. As shown in the figure, the main input and output signals of the DVM are captured, clearly illustrating the system's behavior under dynamic frequency change and highlighting the occurrence of a timing margin violation:

1. **Cycle #1**: The *din* signal transitions low. The delayed signal propagates correctly, allowing *err_pre* to be reset before the next clock edge. As a result, *error* remains low, indicating no timing violation.

2. **Cycle #2**: The *din* signal transitions high. Also in this case *err_pre* briefly pulses high, but it is reset in time before being sampled. Consequently, *error* remains low.

3. **Cycle #3**: The *din* signal transitions low. Due to the increased clock frequency in the test, the timing margin degrades to mimic the behavior of a voltage droop on timing margin. As a result, the delayed signal does not propagate fast enough: *err_pre* remains high and is sampled as such, causing *error* to be asserted high, indicating a timing-margin failure.

4. **Cycle #4**: The *din* signal transitions high. The delayed signal still fails to reset *err_pre* in time, which remains high and continues to be sampled as 1. The *error* signal remains high, confirming the persistent timing violation.

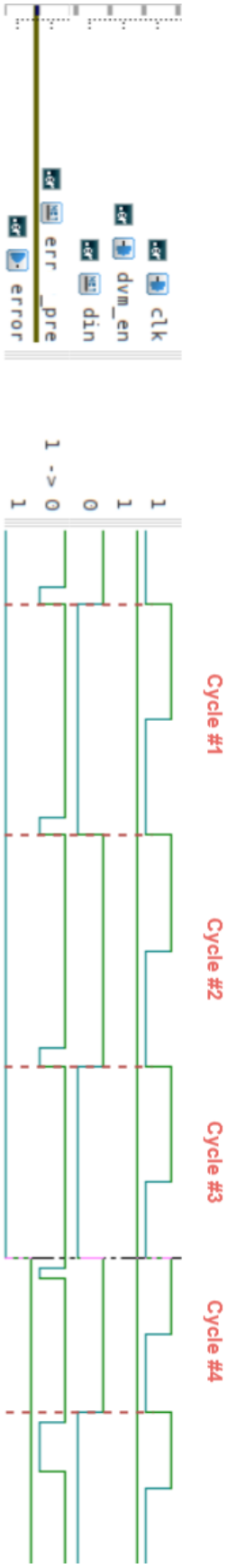

Figure 2.5: DVM UVM simulation.

2.3 Auto-Calibration Circuit

To control the tunable-path delay configuration bits (*tde_ cfg[20:0]*), an auto-calibration circuit [3] directly interfaces with the DVM, using its error signal to guide the calibration process, as shown in Fig. 2.6.

The auto-calibration is managed by a Finite State Machine (FSM), depicted in Fig. 2.7 [9]. The FSM begins in the **Idle** state and remains there until the trigger signal *cal_ start*, which is controlled by writing an on-die register, is asserted.

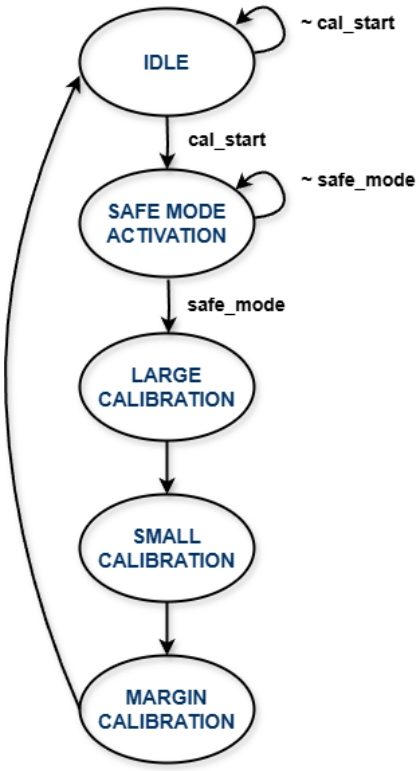

Figure 2.7: Auto-calibration circuit state diagram [3].

Once automatic calibration is initiated (*cal_ start = 1*), the FSM transitions to a state that places the clocked circuit into **safe mode**, since the calibration circuit cannot respond to power supply voltage droops during this phase. In safe mode, when dealing with ACD applications, an idle signal is driven low, forcing the clock selection signal high via a synchronizer. After two rising edges of the delayed clock signal, the synchronization error signal goes high. The delayed clock frequency is then halved and used to drive the clocked circuit, while the DVM continues to receive the high-frequency clock. Alternatively, safe mode may involve executing no-operation (NOP) instructions to prevent unintended operations.

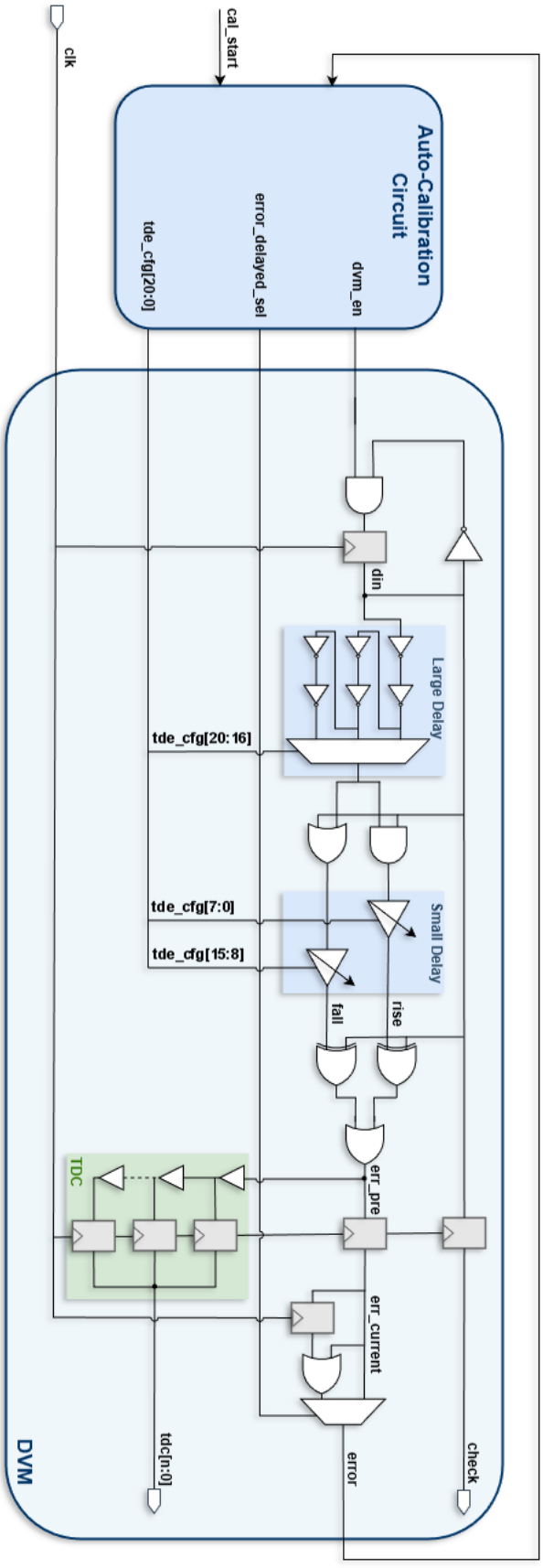

Figure 2.6: Auto-calibration circuit integrated with the DVM [3].

The FSM then enters the **Large Calibration** state, where a *binary search* algorithm is employed to configure the most significant bits (*tde_cfg[20:16]*), as in the example of Fig. 2.8. The process begins by setting the Most Significant Bit (MSB) to 1 and enabling the DVM. If an error is detected, the bit is reset to 0; otherwise, it remains 1. This procedure continues down to the Least Significant Bit (LSB), refining the delay to achieve a timing margin close to zero, as illustrated by the simulation waveforms in Fig. 2.9.

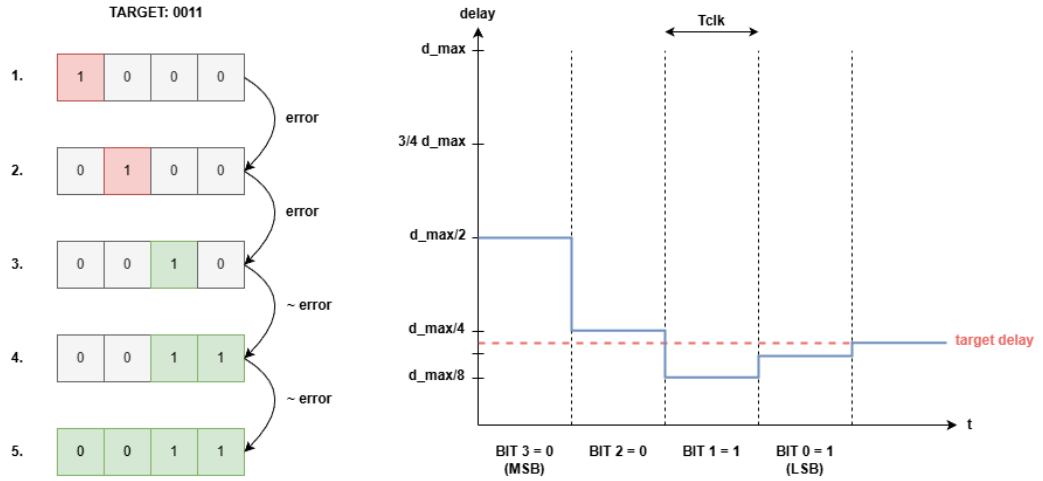

Figure 2.8: Binary search algorithm example.

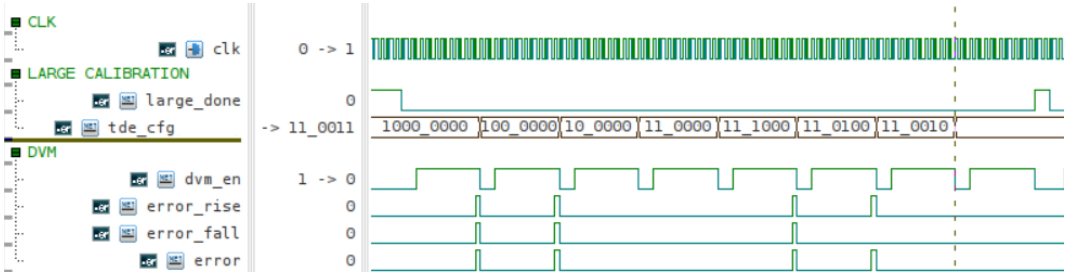

Figure 2.9: Large calibration simulation.

Once the large calibration is completed, the FSM enters the **Small Calibration** state, where a *unidirectional sweep* of *tde_cfg[7:0]* and *tde_cfg[15:8]* is performed to configure the individual timing margin for rising and falling input transitions: this step, besides improving the delay resolution, generates a nearly equal margin for both paths. The algorithm starts by asserting to 1 the LSB and checking if a violation occurs during the DVM activation; if not, also the LSB + 1 bit is set, repeating the check, and so on, as shown in the example of Fig. 2.10. In this case each bit corresponds to the same capacitance value, resulting in a **thermometer**

coding.

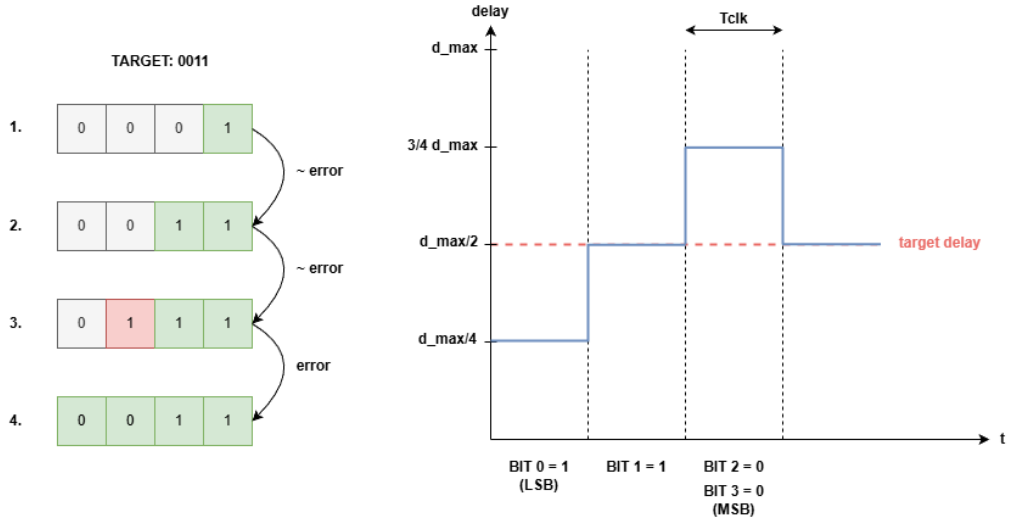

Figure 2.10: Unidirectional sweep algorithm example.

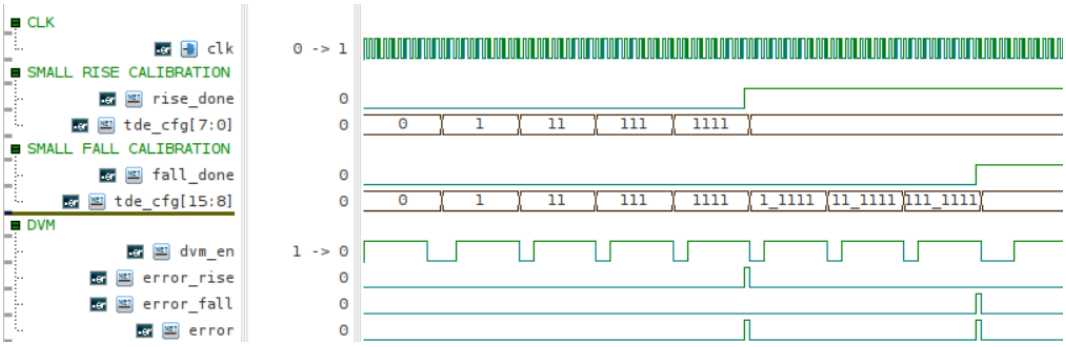

Figure 2.11: Small calibration simulation.

In the **Margin Calibration** state, a small positive timing margin is introduced by removing a programmable number of large TDEs. This ensures the system does not react to minor V_{dd} fluctuations. After this last stage:

$$T_{clk} - T_{path} \approx 0$$

where:

- T_{clk} is the clock period.
- T_{path} is the delay of the tunable path, including the clock-to-output delay of the driving flip-flop and the setup time of the receiving flip-flop.

Finally, the FSM returns to the Idle state and holds the *tde_ cfg[20:0]* configuration until the next calibration cycle is triggered.

2.4 Time-to-Digital Converter

In more advanced DVM-based systems, a **Time-to-Digital Converter** (TDC) is incorporated to measure the timing margin at each clock cycle. This integration allows for precise monitoring and adjustment of timing parameters, ensuring optimal performance and reliability of the system.

TDCs are devices that can be easily integrated on-chip in CMOS technology, used to measure a time interval and convert it into a digital (binary) code. The time interval T to be measured is generally defined with a *Start* and a *Stop* signal [5].

Many parameters are used in order to evaluate the TDC's performance:

- **Time resolution**
  Defines the precision with which the TDC can measure a time interval. This resolution represents the smallest time increment that the TDC can accurately quantify, often referred to as the LSB. This parameter is influenced by various circuit characteristics, including the technology used and the noise performance.

  An ideal TDC is characterized by its ability to produce the first code at a time equal to one LSB, with the subsequent codes shifted by this value along the time axis. In such a scenario, the resolution remains constant. In absence of any distorsion, the transfer function results in a stepped curve, as shown in Fig. 2.12, with each step corresponding to one LSB.

- **Dynamic Range**
  Expresses the maximum time interval that can be correctly measured.

- **Conversion Time**
  Minimum amount of time required to translate a time input into a digital code output, counting from the *Start* event. Within the conversion time is included also the **latency**, that indicates the time needed after the *Stop* event to produce a valid output.

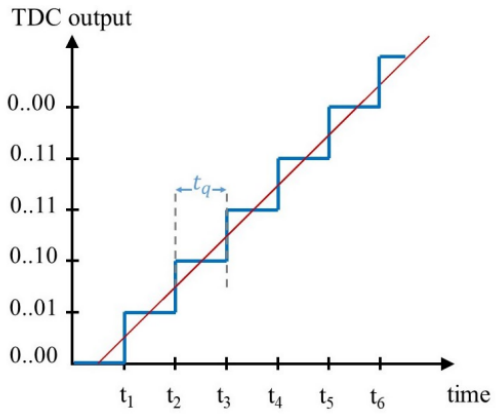

Figure 2.12: Ideal TDC transfer function [5].

In real-world devices, it is essential to account for non-idealities, that can be summarized in two categories: **linear** and **non-linear** imperfections. While the first type is relatively straightforward to detect and correct, the second type presents greater challenges, necessitating advanced calibrations and often proving difficult to completely eliminate. If non-idealities become substantial, they can lead to missing codes, which is a critical issue for TDCs, as shown in Fig. 2.13 (where code i is missing).

Apart from these imperfections, other factors like jitter and thermal noise can also negatively impact the performance of TDCs.

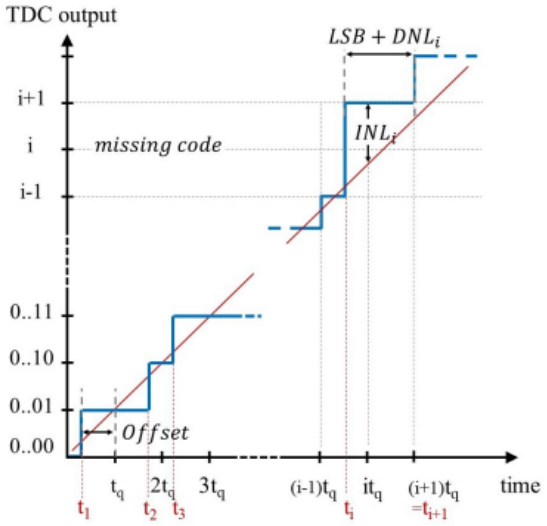

Figure 2.13: TDC transfer function non-linearities [5].

In the context of TDCs, it is crucial to understand the typical non-linearities and linear errors that can affect their performance. The following list outlines these key issues:

- **Offset error**
  Represents a deviation of the TDC from the ideal first measurement, which should be equal to one LSB. For a time interval $0 \leq t < LSB$, the output code should be 0 until reaching a time equal to one LSB. The offset error is defined as the time difference between the actual time t_1, at which the first code 0..01 appears and the ideal time corresponding to one LSB.

  The normalized offset error is expressed as:

  $$E_{off} = \frac{t_1 - t_q}{t_q}$$

- **Quantization error**
  Occurs when the continuous time interval is converted into discrete digital values. Since the TDC can only measure time intervals in fixed increments (LSBs), any time interval that falls between these increments will be rounded to the nearest LSB. This rounding process introduces a small error, typical of any digital converter.

- **Differential non-linearity (DNL)**
  For a specific code, corresponds to the amplitude deviation of the quantization interval with respect to the ideal one. As a result, the real step width is LSB + DNL. If the DNL is too large (DNL $>$ 1 LSB), certain output codes may never appear, resulting in gaps in the output range: very wide intervals can suppress adjacent ones.

- **Integral non-linearity**
  Is defined as the maximum vertical difference between the actual and the ideal curve. It is a measure of the amount of deviation of the real curve from the ideal transfer curve.

The first proposed architectures of TDCs were based on analog techniques, suitable for short measurement ranges, since they provide high precision and good linearity, reaching a resolution of few picoseconds. The traditional structure is based on the conversion of a time interval into a voltage (similarly to ADCs). Traditionally, one possible way to generate the corresponding voltage is to use an **integrator**, in which a capacitor is charged with a constant current source, as presented in Fig. 2.14. The integrator transforms the pulse into a voltage proportional to the time interval. Then, this voltage is converted thanks to an ADC, which provides the digital output.

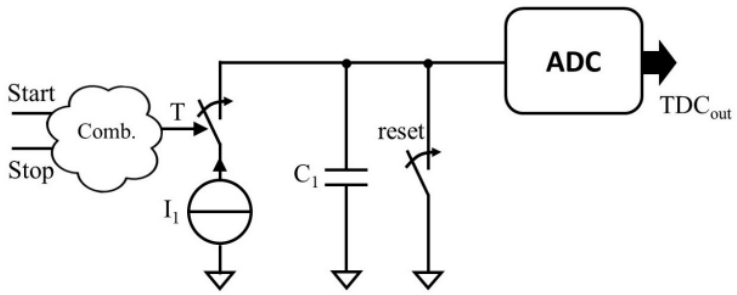

Figure 2.14: Basic architecture of analog TDC [5].

However, analog TDCs have many limitations: firstly, the current sources generate a nonlinear output and suffer from a finite output resistance (non ideal source). Thus, the linearity of the TDC is degraded especially for large dynamic ranges since the analog TDCs are known to be sensitive to the temperature. Furthermore, they are not suitable for fast applications due to the limitations imposed by dead time. Additionally, these TDCs require a large area because of the capacitors involved, leading to higher power consumption compared to their digital counterparts. Consequently, they are less suitable for advanced CMOS technologies.

Today, fully digital techniques have been developed to address these issues. Digital TDCs can be implemented using very small, low-power, and simple circuits. Typically, the architecture of a digital TDC is based on counting the number of edges of a reference clock signal.

2.4.1 Delay-Line based TDC

In the analyzed DVM circuit, the type of TDC used to quantify the timing margin is fully digital and based on delay lines. The basic architecture for this topology is shown in Fig. 2.15: the *error* signal provides an event propagating along the delay line, while the *clk* rising edge acts as a stop signal. The generated output code is represented on n bits.

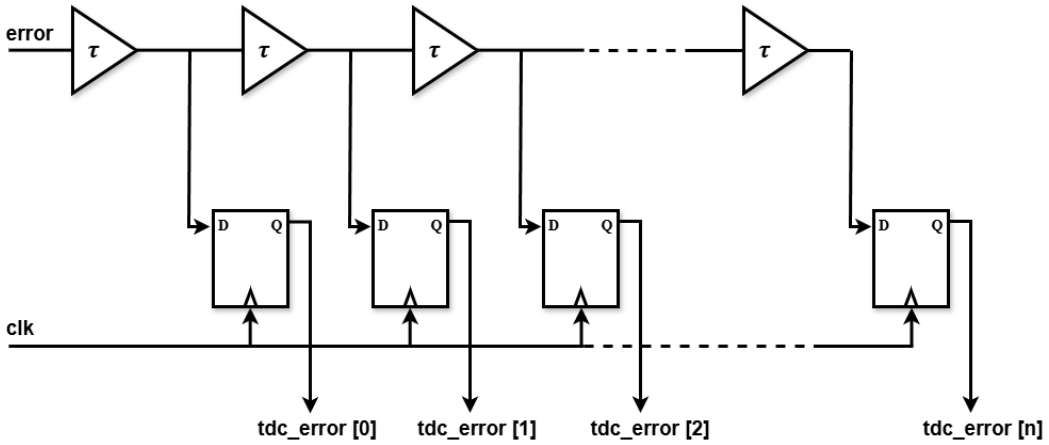

Figure 2.15: Basic implementation of a delay-line based TDC.

This TDC implementation can be easily integrated into Application-Specific Integrated Circuit (ASIC) or Field-Programmable Gate Array (FPGA) but with moderate performances. It presents a low latency because the output can be quickly available after the arrival of the stop event. However, its resolution is bounded by the CMOS gate delays; it cannot be lower than the propagation delay of a logical gate. In addition, to cover a larger time interval, the number of stages must be duplicated as needed. Thus, the cost and power consumption will increase.

In the simulation shown in Fig. 2.16 it can be seen how the TDC input signal *error_in* propagates through all of the eight stages, each one of them introducing a known delay. At each rising edge of the clock the FFs sample the input signal, providing a numeric representation of the timing margin, proportional to the distance between the rising and falling error pulses.

Whenever a switch to higher frequency occurs (in order to simulate a voltage droop), it can be seen that the *error_in* takes longer time to be reset to zero, until on a rising transition the timing margin is violated and the sampled *error* output is equal to 1. Consistently, the *tdc_error* signal assumes its maximum value (i.e. all the FFs are sampling a 1).

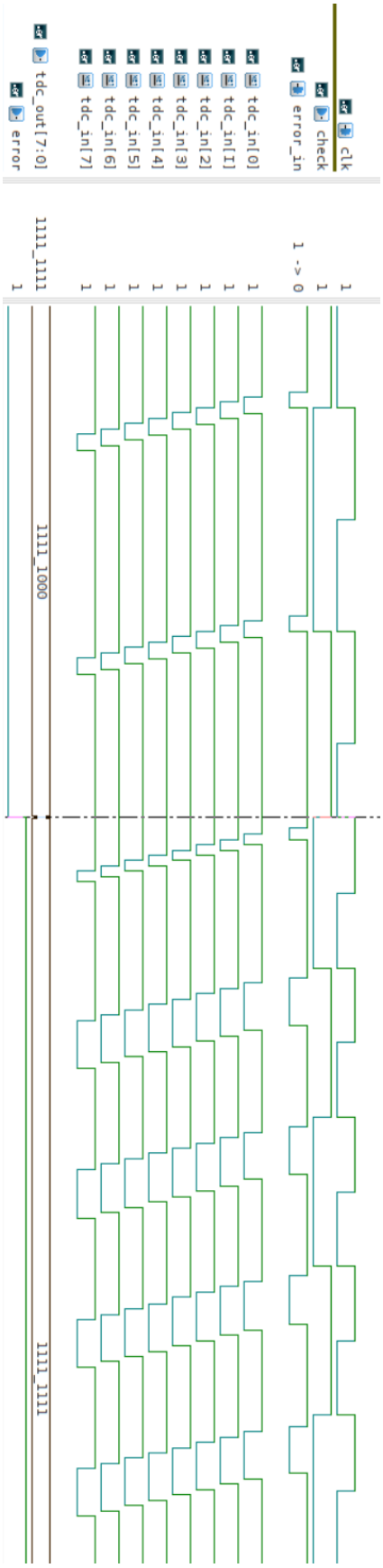

Figure 2.16: TDC simulation.

2.5 Dynamic Variation Monitor Applications

As modern microprocessor systems continue to scale in complexity and performance, the ability to manage dynamic variations in voltage and frequency has become increasingly critical. The DVM plays a central role in this context, enabling real-time detection of timing margin degradation caused by supply voltage fluctuations or frequency shifts. This section presents two representative applications of the DVM, each addressing a distinct class of challenges: *performance resilience* and *system security*.

The first application, **Adaptive Clock Distribution (ACD)**, leverages the DVM in conjunction with a tunable-length delay and clock-gating logic to mitigate the effects of high-frequency voltage droops. By exploiting the temporary alignment between delayed clock and data paths, the ACD system proactively suppresses clock edges that could lead to timing violations, thereby improving energy efficiency and performance without relying on complex error recovery mechanisms.

The second application, addresses the security risks associated with software-level control over voltage and frequency settings. Specifically, it investigates how timing faults, triggered by techniques such as **overclocking** or **undervolting**, can be leveraged to undermine system reliability and integrity. This section begins with a technical overview of timing violations in synchronous digital circuits, providing the necessary background to introduce the DVM-based security monitor as a hardware-level defense mechanism. The monitor continuously tracks timing margins and responds to malicious alterations in voltage and frequency parameters, ensuring safe operational conditions are maintained in real time.

Beyond its original role, the DVM is then repurposed as a digital voltage detector, enabling threshold-based monitoring without the need for analog components.

Together, these applications demonstrate the versatility of the DVM architecture in addressing both performance and security challenges in advanced SoC environments.

2.5.1 Adaptive Clock Distribution

Adaptive Clock Distribution [6] is a design technique aimed at mitigating the impact of high-frequency supply voltage droops on the performance and energy efficiency of microprocessors.

Previous adaptive circuit methodologies focused on mitigating the impact of V_{dd} droops by actively monitoring supply voltage fluctuations with an on-die sensor and

modifying the operating frequency conditions accordingly, as described in 1.3.1. Such circuits necessitate a detection period for the V_{dd} variation and a response interval to *prevent critical-path failures*. While these designs can substantially alleviate performance degradation at lower frequencies, issues with high-frequency droops persist.

As an alternative, resilient timing-error detection and recovery circuits relax the response time constraint and avoid the limitations of analog circuitry, by allowing a V_{dd} droop to *induce a timing failure*.

To comprehend the effects of a droop event, it is essential to analyze its impact on both the clock distribution delay and the datapath delay within a microprocessor pipeline. Figure 2.17 illustrates a standard clock distribution system.

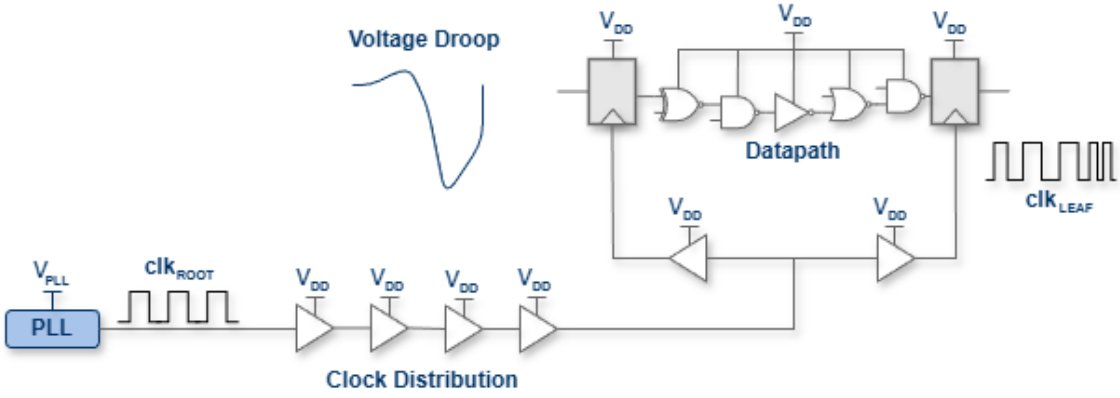

Figure 2.17: Conventional Clock Distribution [6].

Fig. 2.18 presents a concrete example that highlights standard clock distributions limitations under droop conditions:

1. In Cycle #1, both the root and leaf clock signals (clk_{ROOT} and clk_{LEAF}) operate at a nominal period of 1.00 ns, and the datapath delay (T_{DATA}) is also 1.00 ns, resulting in a balanced timing scenario;

2. However, in Cycle #2, a voltage droop increases the delay of the clock distribution path, causing clk_{LEAF} to stretch to 1.25 ns, while clk_{ROOT} remains unaffected at 1.00 ns due to its separate voltage domain (V_{PLL}). This extended clock period at the leaf node compensates for the slower datapath delay, which also increases to 1.25 ns, resulting in a temporary alignment, an effect known as **clock-data delay compensation**;

3. In Cycle #3, although the droop persists and both the clock and data paths

remain slow at 1.25 ns, no further compensation occurs because the new clock edge is now referenced to the previous delayed edge. This leads to a compressed effective clock period and a negative timing margin, potentially causing a timing failure.

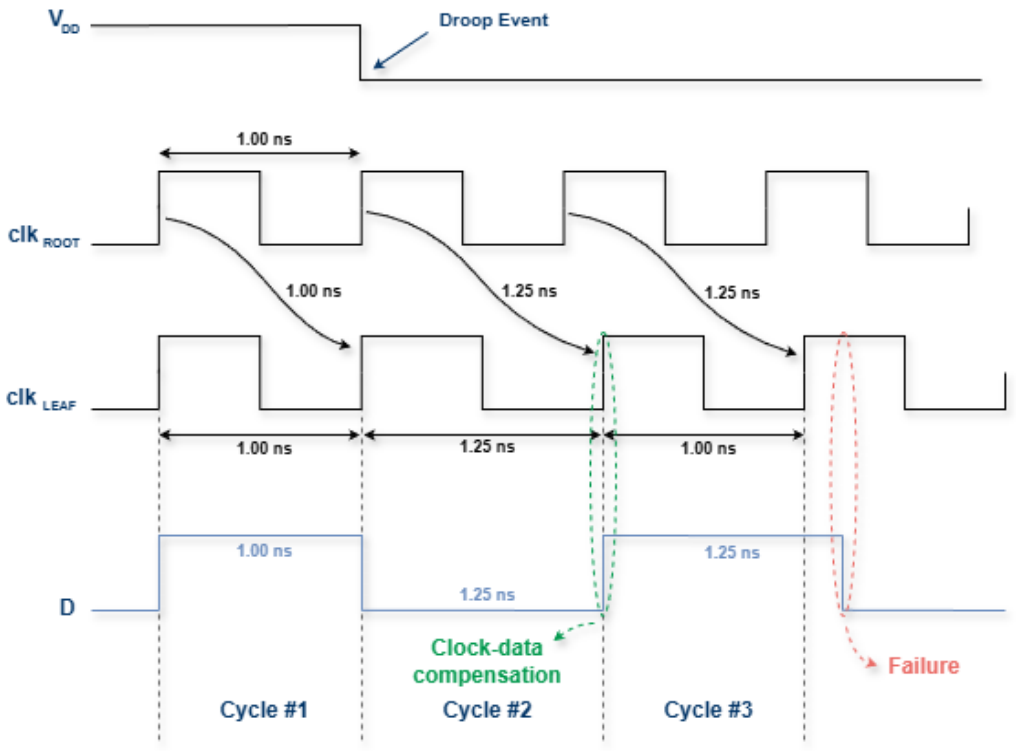

Figure 2.18: Timing diagram of temporary clock-data delay compensation [6].

An all-digital, dynamically Adaptive Clock Distribution system can significantly improve performance and energy efficiency under voltage droop conditions. The design leverages the clock-data delay compensation effect, which temporarily aligns delayed clock and data paths during a droop, providing enough response time to proactively gate the clock. This approach helps mitigate the impact of droops without relying on complex error recovery mechanisms typically required in resilient designs.

At the core of the Adaptive Clock Distribution system (Fig. 2.19) is a **Tunable-Lenght Delay (TLD)** inserted before the global clock distribution. This delay, implemented using scan-programmable transistors and interconnects, temporarily compensates for increased datapath delays during voltage droops by adjusting the delay sensitivity of the clock path. A dedicated DVM detects the onset of a droop

and activates a **clock-gating circuit**, selectively suppressing clock edges that could violate critical path timing constraints. This coordinated response helps preserve timing margins for several cycles after the droop begins [6].

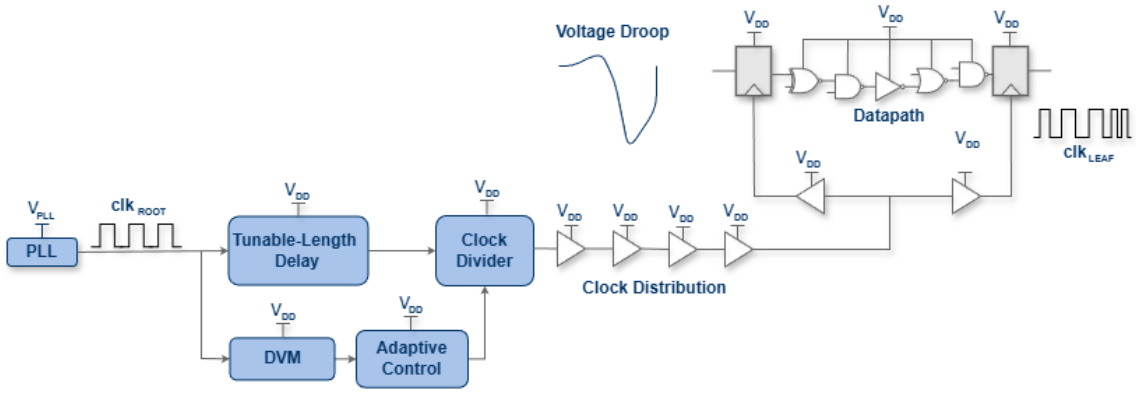

Figure 2.19: Adaptive Clock Distribution [6].

Post-silicon calibration is aimed at achieving a specified tunable-length delay and **delay sensitivity** (how much the delay changes in response to variations in supply voltage) for the clock distribution. Given that adaptive clock gating occurs subsequent to the tunable-length delay circuit, the target TLD is determined by the product of the clock period and the number of cycles required for adaptive response. For instance, with a clock frequency of 1.0 GHz (1 ns period), a 4-cycle response necessitates a tunable-length delay of 4 ns. The target delay is entirely independent of droop frequency.

The target clock-distribution-delay sensitivity equals the datapath-delay sensitivity for the critical paths. This condition maximizes the clock-data delay compensation during a droop or overshoot. Otherwise, non-ideal clock-data delay compensation occurs, and consequently, reduces the potential benefits of the ACD to mitigate the impact of V_{dd} droops on performance and energy efficiency. As an example, if the clock distribution delay is more sensitive to the power supply than the datapath delay for the critical paths, then the leaf clock period stretches further than the datapath delay increases during a droop, resulting in a positive critical-path timing margin. During a overshoot, however, the leaf clock period compresses more than the datapath delay reduces, resulting in a negative critical-path timing margin. As a contrasting example, if the clock distribution delay is less sensitive to V_{dd} than the datapath delay for the critical paths, then the positive and negative critical-path timing margins occur for overshoots and droops, respectively. Since clock distribution

and datapath differences in delay sensitivity to limit the clock-data delay compensation, the **TLD calibration** with transistor and interconnect delay components aims to minimize these differences.

The calibration of the target clock-distribution-delay sensitivity to V_{dd} requires a measurement of the datapath-delay sensitivity for the critical paths: it is necessary to perform multiple tests at the maximum frequency f_{max}, at V_{dd} values above and below the nominal one, to generate measurements of the worst-case datapath delay versus supply voltage; in this way, the target sensitivity is derived.

This value is then used to adjust the delay sensitivity of the clock distribution network. After enabling and configuring the TLD to drive the clock at nominal V_{dd}, on-die measurements of the entire clock distribution delay are taken at the same V_{dd} levels used in the $f_{\max}$ tests. These measurements allow for the calculation of the clock distribution delay's sensitivity to V_{dd}. Since transistor delays are more sensitive to V_{dd} variations than interconnect delays, the TLD includes both transistor and interconnect delay components to accurately tune the overall clock distribution delay sensitivity to V_{dd}.

2.5.2 Additional Applications

The demand for computing that is both power and energy efficient has led to the development of robust cooperative hardware-software energy management strategies in contemporary commodity devices. For instance, many current systems enable software to manage the frequency and voltage of the hardware components with a high degree of precision to prolong battery lifespan. A prominent example of such a technique is **Dynamic Voltage-Frequency Scaling (DVFS)**, a ubiquitous energy management technique that saves energy by regulating the frequency and voltage of the processor cores according to runtime computing demands. However, despite the advantages they offer, these energy management mechanisms that are accessible to software present significant security risks that have not been previously examined.

In this context, the **$\mathbf{CLK_{screw}}$ attack** represents a novel category of fault attacks that take advantage of the security unawareness inherent in energy management systems to compromise security. A significant advantage for attackers is that these fault attacks have become increasingly attainable, as they can now be executed without requiring physical access to the devices or specialized fault injection tools [7]. Essentially, a $\mathrm{CLK_{screw}}$ attack takes advantage of unrestricted software access to energy management hardware, thereby pushing the operational limits of processors to the extent of causing erroneous computations. This poses a significant risk, particularly when such faults can be triggered from lower-privileged software across hardware-

enforced boundaries, where computations that are sensitive to security are executed.

To comprehend the risks associated with unrestricted access to hardware regulators, it is essential to grasp the general reasons why exceeding frequency limits (commonly referred to as overclocking) or insufficiently supplying voltage (known as undervolting) may lead to unintended behaviors in digital circuits.

Synchronous digital circuits consist of memory components known as Flip-Flops (FF). These FFs are responsible for storing stateful data necessary for digital computation. A typical FF has an input D, and an output Q, and only changes the output to the value of the input upon the receipt of the rising edge of the clock.

For a single flip-flop to correctly propagate the input to the output, three key **timing constraints**, illustrated in Fig. 2.20, must be satisfied:

1. **Setup Time**: the incoming data signal must remain stable for a duration of T_{setup} before the arrival of the next clock edge;

2. **Hold Time**: the input signal must also be held stable within the flip-flop for a duration of T_{FF} after the clock edge;

3. **Propagation Delay**: a minimum propagation delay of $T_{\text{max\_path}}$ is required for the output Q_{src} of the source flip-flop (FF_{src}) to reach the input D_{dst} of the destination flip-flop (FF_{dst}).

Therefore, for the circuit to reliably propagate the input D_{src} to the output Q_{dst}, the minimum required clock cycle period T_{clk} must satisfy the following timing constraint:

$$T_{\text{clk}} \geq T_{\text{setup}} + T_{\text{FF}} + T_{\text{max\_path}} + K$$

where K is a microarchitectural constant accounting for additional timing margins.

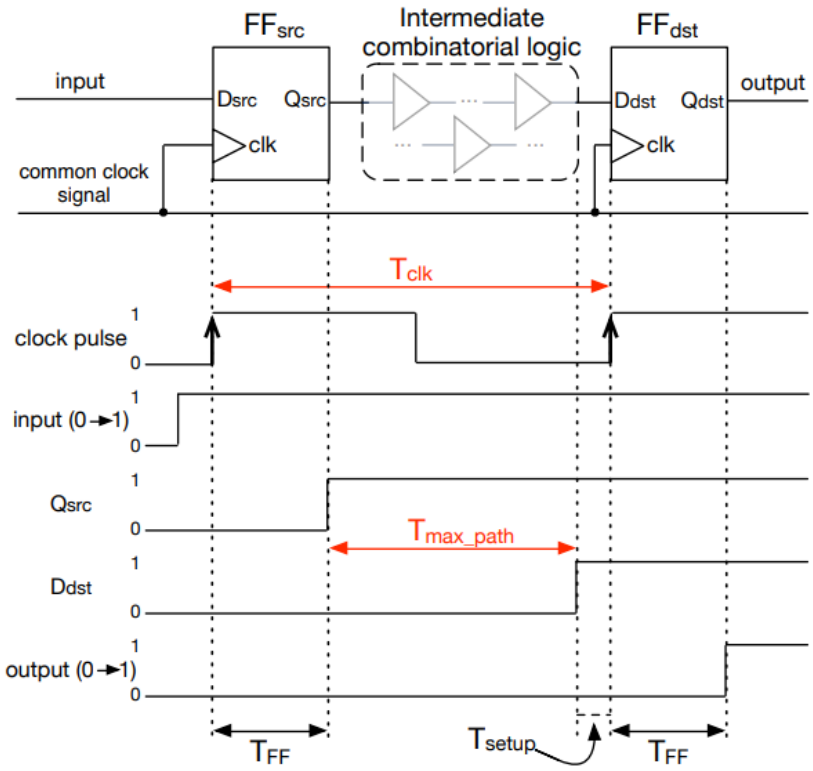

Figure 2.20: Timing constraint for error-free data propagation [7].

When a timing constraint is violated across two consecutive rising edges of the clock signal, the output from the source flip-flop (FF_{src}) fails to propagate correctly to the input of the destination flip-flop (FF_{dst}) within the required time window. As a result, FF_{dst} continues to operate on stale or incorrect data.

There are two primary scenarios in which this timing constraint can be violated:

1. **Overclocking**, which reduces the clock period T_{clk} below the minimum required threshold (Fig. 2.21);

2. **Undervolting**, which increases the propagation delay of the circuit, thereby increasing T_{max_path} and potentially violating the setup time.

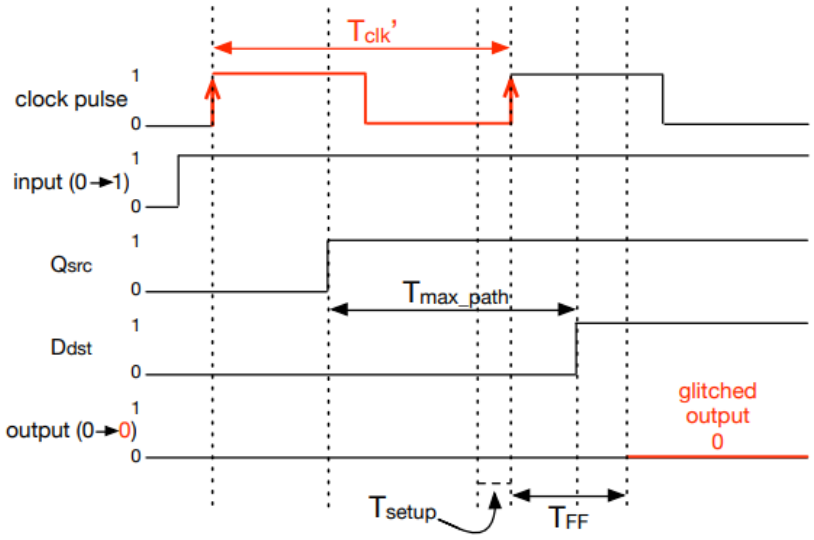

Figure 2.21: Bit-level fault due to overclocking [7].

In summary, the integrity of synchronous digital circuits critically depends on satisfying strict timing constraints that govern the interaction between clock signals and data propagation. When these constraints are violated, either through overclocking or undervolting, faults can be introduced at the hardware level, leading to incorrect or stale data being processed. These faults, once considered rare or requiring physical access, can now be deliberately induced via software-level manipulation of energy management interfaces. This highlights the urgent need to reassess the security implications of exposing low-level power and frequency controls to software, especially in systems where sensitive computations are expected to be isolated and protected.

Beyond timing faults, modern systems must also contend with increasingly sophisticated software-based threats. In the past ten years, harmful software has emerged as a widespread issue for computer users. Specifically, a category of malware may demonstrate actions such as infecting, encrypting, deleting, and/or stealing files (hereafter referred to as *file altering malware*). This type of malware specifically aims at computer systems for:

- Restricting access to portions of a computer system and demanding payment for the removal of the restriction;
- Infecting computer systems with information theft routines, which may seek to steal information such as login credentials to applications, system information, File Transport Protocol (FTP) credentials.

Infecting malware may target a computer architecture, such as High-Level Operating System (HLOS) of the computer architecture. Unfortunately, malware detection systems for preventing infecting malware have difficulty detecting that a computer architecture is affected by the infecting malware. This difficulty in detection occurs because non-malicious applications may affect the computer architecture in a manner similar to the infecting malware. Therefore, current malware detection systems may fail to prevent the infecting malware from using the HLOS to negatively alter the computer architecture and cause device failure.

An on-die DVM-based Security Monitor [16] can be employed within the system to safeguard against malicious attacks targeting the Central Processing Unit Subsystem (CPUSS). Its primary components consist of the DVM and the associated Auto-Calibration circuit, which adjusts the tunable delay elements of the Dynamic Variation Monitor according to the current operating frequency f_{clk} and supply voltage V_{dd}, ensuring that the timing margin is approximately zero.

By continuously monitoring timing margins and adapting its delay configuration through auto-calibration, the system can detect anomalous conditions such as voltage droops or frequency overshoots. The core idea is to maintain a calibrated timing margin under nominal conditions, so that any unexpected variation, such as a voltage droop or frequency overshoot, produces a detectable timing error. These anomalies may indicate unsafe or unintended operating states, potentially caused by external disturbances or malicious manipulation.

When unsafe conditions are identified, the monitor activates a protective mechanism that adjusts the clock frequency to a secure level. Simultaneously, a secure interrupt is triggered to initiate appropriate countermeasures, thereby preserving system stability and preventing fault-induced vulnerabilities.

In addition to its role in detecting timing anomalies, the DVM can be repurposed as a **voltage detection** mechanism [4]. By calibrating its sensitivity to specific voltage thresholds, the monitor becomes capable of identifying when the supply voltage crosses predefined limits. This is achieved by tuning the delay path such that the timing margin becomes zero at a target voltage level; any deviation from this calibrated point results in a detectable timing error. This principle allows the DVM to act as a fully digital voltage comparator, eliminating the need for analog components and improving scalability across technology nodes. The next chapter builds upon this concept to present the design of a Voltage-Level Detector (VLD), which leverages the DVM for dynamic assist configuration in SRAM.

2.6 Conclusion

This chapter presented the architecture, calibration mechanisms, and practical applications of the DVM, a fully digital circuit designed to detect timing margin violations caused by dynamic variations in voltage and frequency. Through the integration of tunable-delay elements and auto-calibration logic, the DVM enables fine-grained, cycle-level monitoring of critical path delays, offering a scalable and synthesizable solution for modern SoC environments.

Two representative applications were explored in detail. The first, ACD, leverages the DVM to proactively suppress clock edges during high-frequency voltage droops, thereby preserving timing margins without relying on complex error recovery mechanisms. The second demonstrates how the DVM can be employed in security-sensitive contexts to detect anomalous operating conditions, such as those caused by malicious voltage or frequency manipulation, and to support adaptive countermeasures.

Building upon the architectural foundations and operational principles of the DVM, a novel design was conceived to implement an all-digital VLD. This system repurposes the DVM as a voltage detection mechanism, enabling precise threshold monitoring in real time. By reusing the existing infrastructure, including the auto-calibration logic, and integrating additional digital control blocks, the proposed architecture achieves a lightweight and fully synthesizable solution for voltage-aware assist activation.

The next chapter will detail the design and implementation of this voltage detection system, and demonstrate how it can be used to dynamically trigger assist strategies in SRAM arrays under aggressive voltage scaling.

Chapter 3

All-Digital Voltage Level Detector

3.1 Introduction

In the previous chapter, the Dynamic Variation Monitor (DVM) was introduced as a robust mechanism for detecting timing violations caused by voltage droops or frequency overshoots. Building upon this foundation, this chapter presents the design and implementation of an all-digital Voltage Level Detector (VLD). The proposed system leverages the architecture and operational principles of the DVM to enable a fully synthesizable and technology-portable solution for voltage monitoring.

The primary objective of the VLD is to dynamically detect when the supply voltage crosses predefined thresholds and to generate a trigger signal accordingly. This trigger is then used to activate specific *read and write assist strategies* within SRAM arrays, thereby enhancing their robustness under aggressive voltage scaling. To achieve this, the system dynamically sets the **Access Control Configuration (ACC)** bits of the SRAM based on the detected voltage level. These ACC bits serve as programmable control flags that define the access behavior of each SRAM cell. Depending on the operating conditions, they can enable or disable assist circuits, adjust wordline pulse widths, or modify bitline precharge conditions. This dynamic configuration allows the memory to adapt in real time, optimizing both performance and energy efficiency while maintaining data integrity.

This approach is particularly beneficial in advanced CMOS technologies, where reduced supply voltages and increased variability pose significant challenges to SRAM stability. At low V_{dd} levels, the Static Noise Margin (SNM) of the memory cell degrades, making it more susceptible to read and write failures. To counteract this, assist techniques are commonly employed to temporarily modify the cell's electrical conditions during access operations.

Techniques like *wordline underdrive*, *supply voltage collapse*, and *negative bitline*

are widely used to reinforce cell stability under these conditions. However, when operating at higher voltages, such aggressive methods may be unnecessary or even counterproductive. In such cases, more refined strategies can offer better efficiency. The proposed VLD architecture addresses this challenge by enabling voltage-aware assist activation: by dynamically detecting the operating voltage region, the system can selectively apply the most appropriate assist strategy, thereby optimizing memory behavior across a wide range of supply conditions.

This chapter is organized to guide the reader through the design and implementation of the proposed VLD, starting from its architectural foundations and progressing toward its control logic and operational behavior.

It begins by introducing the overall architecture of the system, explaining how the existing infrastructure has been extended to support dual-threshold voltage detection. This includes a discussion of the modifications introduced to enable the system to distinguish between high and low voltage regions, and to respond accordingly.

The chapter then focuses on the **VLD Control Block**, which plays a central role in interpreting voltage-level and timing margin signals. This block generates intermediate detection flags that are used to drive the internal FSM, ensuring that assist triggers are only asserted under valid and stable conditions, avoiding undesired oscillations in ACC settings.

Following this, the FSM operation is described in detail; the FSM governs the dynamic behavior of the VLD, managing state transitions based on real-time voltage and calibration signals. Its design ensures robustness against noise and transient fluctuations, while also enabling power-efficient operation by selectively disabling the DVM when not required.

Finally, the chapter concludes with a summary of the key contributions of the VLD design, highlighting its advantages in terms of scalability, energy efficiency, and integration into modern digital design flows.

For a comprehensive overview of the SRAM assist techniques referenced throughout this chapter, the reader is encouraged to consult Appendix A, which provides detailed descriptions of both read and write assist strategies, including their operating principles, benefits, and limitations.

3.2 Design Architecture and Operation

Building upon the DVM framework introduced in the previous chapter, this section presents the architecture of the proposed all-digital Voltage Level Detector. The system is designed to detect when the supply voltage crosses predefined thresholds

and to generate corresponding trigger signals to dynamically configure SRAM assist strategies.

The DVM achieves voltage change detection by continuously monitoring the timing margin of a tunable delay path. Since the propagation delay of digital circuits is strongly dependent on the supply voltage, any deviation from the nominal voltage, either a droop or an overshoot, manifests as a variation in the timing margin. When this margin becomes negative, the DVM asserts an error signal, effectively flagging a voltage transition. This mechanism enables the DVM to detect voltage changes indirectly but with high sensitivity and digital precision, without relying on analog components [4].

However, relying solely on a DVM-based detection system is not sufficient to ensure robust and reliable dynamic assist configuration in SRAMs. Although this approach can detect voltage undershoots, it suffers from several critical limitations that undermine its effectiveness in practical scenarios.

First, it lacks the ability to differentiate between high and low voltage levels, which is essential for making accurate and context-aware assist decisions. This limitation arises from the intrinsic behavior of the DVM, which detects voltage changes indirectly by monitoring timing margin violations: when the supply voltage drops, the propagation delay of the monitored path increases, potentially leading to a negative timing margin and triggering an error signal. However, in the case of a voltage overshoot (i.e., an increase in supply voltage) the propagation delay decreases, resulting in a larger positive timing margin. Since no timing violation occurs, the DVM does not raise any error, and the voltage transition remains undetected.

To enable the detection of both undervoltage and overvoltage events, an additional configuration mechanism has been introduced. Specifically, a configuration bit (dvm_err_cfg) allows the interpretation of the DVM output to be inverted when operating in high-threshold detection mode. This enables the system to treat the absence of a timing violation as an indication of a voltage overshoot, thereby extending the DVM's functionality to support dual-threshold voltage detection.

The second limitation, instead, concerns the system's sensitivity to transient noise and voltage rail oscillations, which can cause repeated threshold crossings. These false triggers lead to unstable assist behavior, such as unnecessary toggling of assist mechanisms, ultimately degrading both performance and energy efficiency. Furthermore, the absence of a mechanism to validate whether a detected voltage change is intentional or spurious makes the system vulnerable to unpredictable behavior under dynamic operating conditions.

To overcome this, a more structured triggering mechanism has been adopted, inspired by the architecture described in *US Patent No. 10,317,968 B2*, titled "Power Multiplexing with an Active Load" by Pant et al. [8], that introduces a power-multiplexing scheme that leverages two key signals: a *Relative Voltage Signal*, generated by an analog voltage comparator that monitors the voltage difference between two supply rails, and a *Voltage-Level Indication Signal*, issued by a voltage controller to explicitly validate the voltage change. This dual-signal mechanism is designed to prevent spurious transitions caused by transient noise or rail oscillations, ensuring that switching between power rails occurs only when the voltage change is intentional and stable.

The triggering circuit proposed in the patent, shown in Fig. 3.1, served as the conceptual foundation for the development of a more scalable and digitally integrated solution. The original architecture introduces the idea of combining analog voltage comparison with a digital control signal to ensure that switching between power rails occurs only when explicitly validated. However, it is primarily intended for coarse-grained power domain management, such as dynamic voltage scaling in processors, where entire cores or subsystems are transitioned between supply rails to optimize energy consumption.

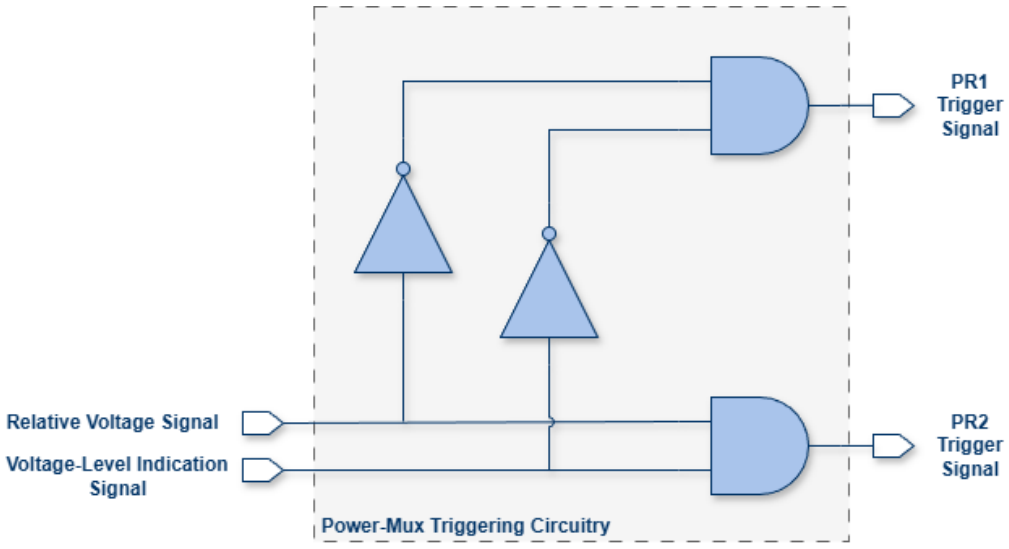

Figure 3.1: Power-Mux Triggering Circuitry [8].

The proposed design leverages the same dual-signal validation principle, combining a relative voltage signal with a voltage-level indication signal, but in a fully digital context: while the original power-multiplexing scheme [8] relies on an analog voltage comparator to generate a relative voltage signal, the presented architecture

adopts a similar control logic but replaces the analog comparator with a fully digital DVM.

In light of these considerations, the proposed system extends the dual-signal validation approach to enable robust and energy-efficient SRAM operation across varying supply voltages. As illustrated in Fig. 3.2, the system is designed to assert one of two trigger signals depending on the supply voltage region:

- *trig_high* is asserted when the supply voltage exceeds the high threshold Vth_{high}, enabling assist techniques optimized for high-voltage operation;
- *trig_low* is asserted when the supply voltage drops below the low threshold Vth_{low}, activating assist mechanisms tailored for low-voltage conditions.

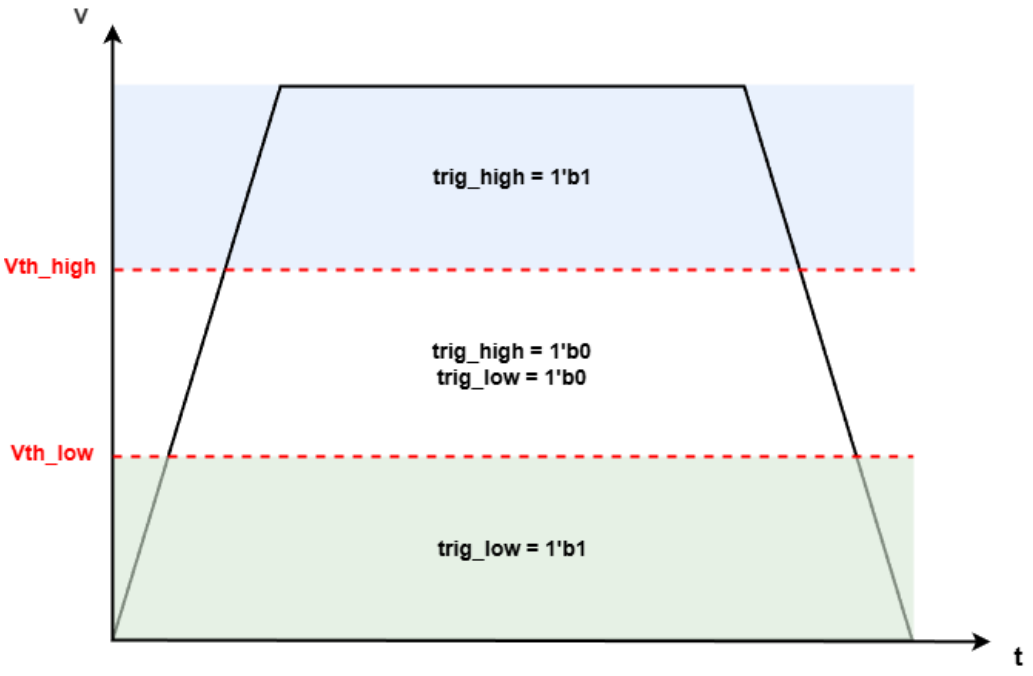

Figure 3.2: Expected behavior of *trig_high* and *trig_low* signals across voltage regions.

To implement this **dual-threshold** detection mechanism, the proposed architecture employs two independent VLD instances, each tuned to a specific reference voltage, as shown in Fig. 3.3. These detectors operate in parallel, enabling the system to respond to both undervoltage and overvoltage conditions with minimal latency.

In this application, it is assumed that the system operates at a **fixed clock frequency**. This constraint is essential because the timing margin evaluated by the DVM is inherently dependent on the operating frequency: any variation in frequency would alter the timing margin. In such cases, it would become unclear whether a detected timing violation is caused by a change in the supply voltage or by a shift in the operating frequency, compromising the reliability of voltage detection. Therefore, to maintain reliable threshold detection, each DVM must be calibrated specifically for the frequency at which the system is intended to operate; if the frequency changes, the calibration process must be repeated to realign the delay elements with the new timing conditions.

The architecture described below represents the general framework adopted for dynamic ACC setting:

- Each VLD integrates a DVM block that continuously monitors the timing margin of a tunable-delay path, raising an error whenever a violation occurs;
- The error signal is processed by the VLD Control block, which also receives a voltage-level indication signal from a dedicated Voltage Controller;
- A Finite State Machine (FSM) within the control logic validates both inputs before asserting a trigger signal. This dual-signal validation suppresses false activations due to transient noise or rail oscillations;
- Trigger signals from each VLD (*trig_high* and *trig_low*), along with their corresponding validity indicators (*trig_valid_high* and *trig_valid_low*), are routed to the Memory Control units;
- These signals are jointly evaluated to dynamically configure the SRAM assist circuits, ensuring that assist mechanisms are only activated when genuinely required.

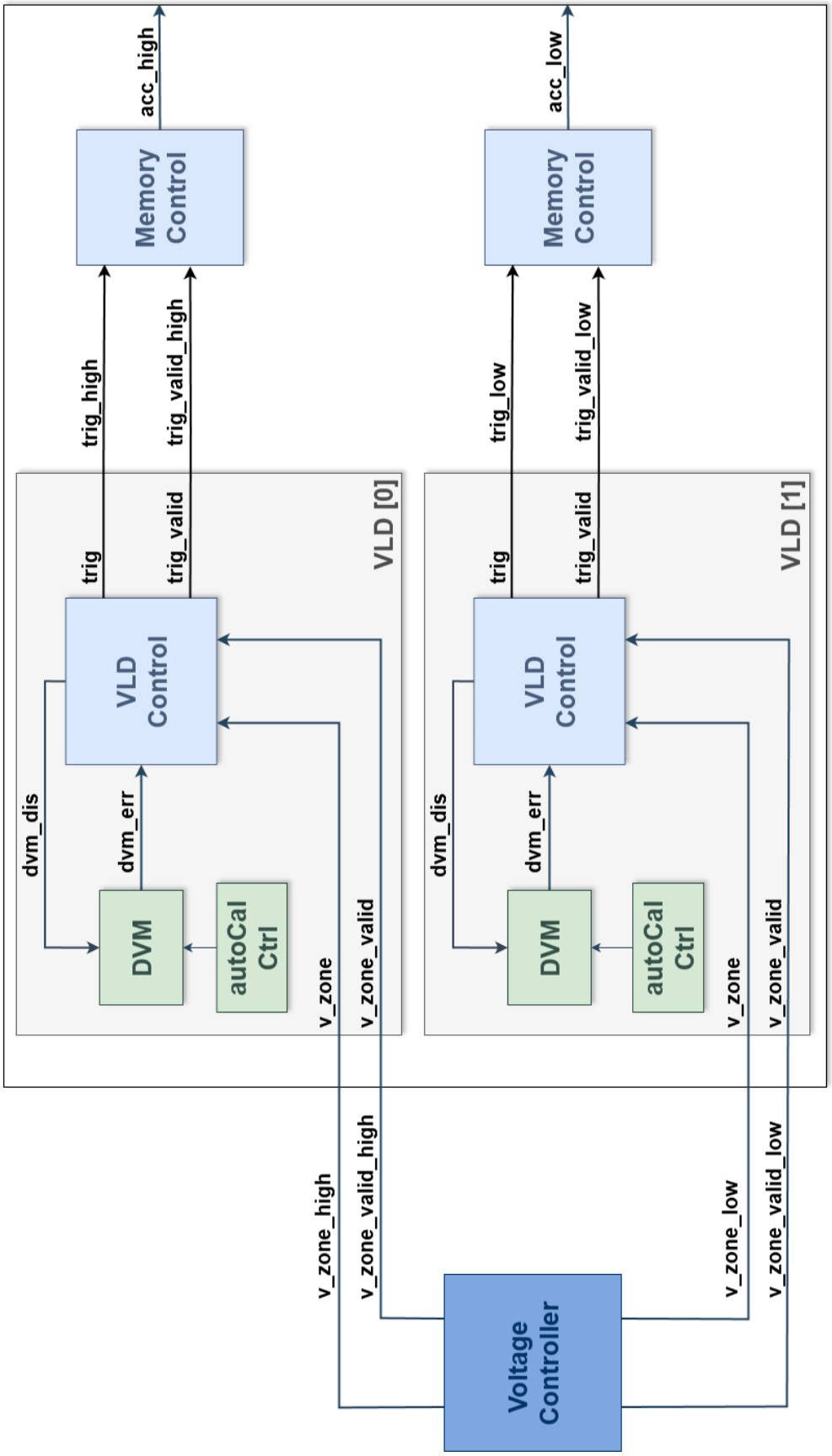

Figure 3.3: Block diagram of the proposed dual-threshold architecture.

More in detail, each VLD (Fig. 3.4) is mainly composed of three tightly integrated functional blocks:

1. **Dynamic Variation Monitor (DVM)**: as discussed in 2.2, this block continuously evaluates the timing margin of a tunable-delay path. It is calibrated to its respective voltage threshold, such that the timing margin is zero at the target voltage, allowing the DVM to reliably detect deviations from the calibrated point. It raises an error signal when the margin becomes negative, indicating a voltage droop. The DVM includes both large and small delay elements to provide a wide tuning range and high resolution, respectively. It is clocked at high frequency and operates cycle-by-cycle;
2. **Auto-Calibration Circuit**: to ensure accurate detection across different frequency and voltage conditions, the DVM is calibrated using an automatic calibration loop. This circuit, already analyzed in detail in 2.3, adjusts the delay elements such that the timing margin is zero at the target threshold voltage;
3. **VLD Control**: this logic block processes the outputs of the DVM and the voltage-level indication signal from a Voltage Controller. It implements a FSM that ensures triggers are only issued when both the DVM indication and the controller signal are consistent, thereby suppressing false activations due to transient noise or rail oscillations.

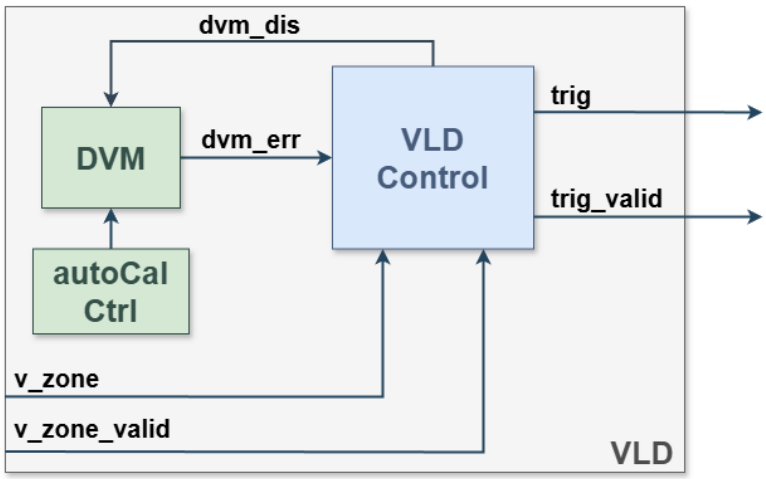

Figure 3.4: Close-up view of Voltage Level Detector instance.

Since the operation of the DVM and the auto-calibration circuit has already been thoroughly discussed in Chapter 2, the following section focuses on a detailed analysis of the additional control logic implemented in the *VLD Control* block.

3.3 VLD Control Block

The VLD Control Block, illustrated in detail in Fig. 3.5, is a central component in the dynamic configuration of SRAM assist circuitry. Its primary role is to interpret Voltage Controller indication signals and DVM's timing margin violations detection, and to generate appropriate trigger signals for activating or deactivating assist mechanisms. This logic ensures that only valid and meaningful voltage transitions result in configuration changes, thereby avoiding false triggers caused by noise or transient oscillations.

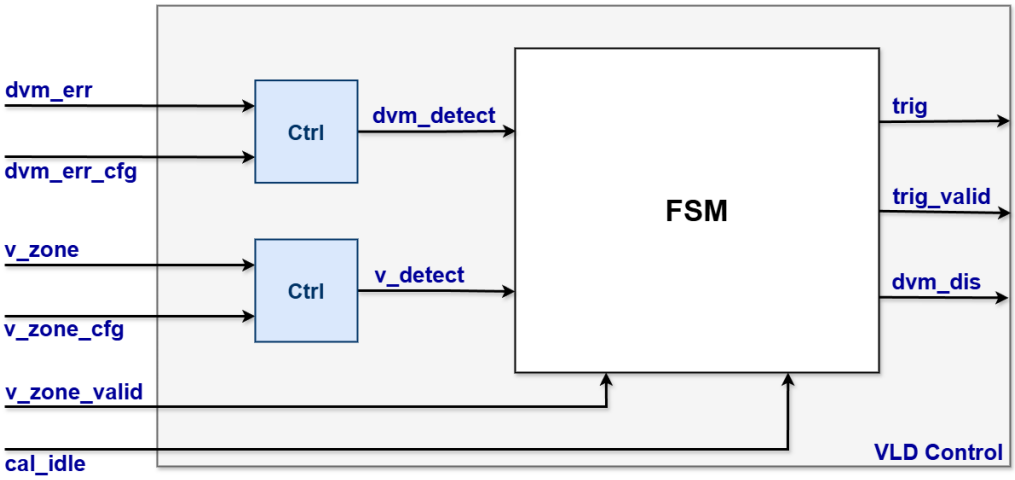

Figure 3.5: VLD Control block.

The controller receives the following input signals:

- ***v_zone***: indicates whether the system intends to operate above or below the configured voltage threshold, as determined by the Voltage Controller. A value of 1 means the system targets $V > V_{th}$, while 0 means it targets $V < V_{th}$;

- ***v_zone_valid***: indicates whether the *v_zone* signal is valid. When this signal is low, the voltage region is undefined and no action is taken. When high, the system can safely interpret the *v_zone* signal;

Table 3.1: Voltage level interpretation logic.

v_zone_valid	v_zone	Voltage Region
0	X	Undefined
1	1	$V > V_{th}$
1	0	$V < V_{th}$

- ***v_ zone_ cfg***: configuration bit that selects whether the controller operates in high-threshold or low-threshold mode. In the following descriptions, we assume that a value of 1 corresponds to high-threshold detection, while 0 corresponds to low-threshold detection;

- ***dvm_ err***: error signal generated by the DVM, indicating whether a timing margin violation has occurred. A high value typically corresponds to a voltage droop (i.e., $V < V_{th}$);

- ***dvm_ err_ cfg***: configuration bit that determines how the *dvm_ err* signal should be interpreted. For high-threshold detection, that is intended to recognize voltage overshoots, the timing indication logic must be inverted, since the DVM is typically designed to detect undershoots. This bit allows the same DVM logic to be reused for both voltage overshoot and undershoot detection;

- ***cal_ idle***: signal from the Auto-Calibration circuit indicating whether a calibration process is currently active. When calibration is in progress (*cal_ idle = 0*), no action should be taken and the ACC bits must remain unchanged to avoid misconfiguration during unstable conditions.

Within the VLD Control Block, two key intermediate signals, *v_ detect* and *dvm_ detect*, are generated by dedicated combinational logic. These signals are not final outputs of the system, their purpose is to provide the FSM with real-time information about the voltage domain and timing margin status, which ultimately manages the configuration of the assist circuitry.

The generation of these signals is based on the interpretation of the voltage controller indicators and timing margin violations:

- ***v_ detect***: asserted when the voltage enters a region that requires assist activation. Its value depends on the combination of *v_ zone* and *v_ zone_ cfg*, as shown in Table 3.2.

Table 3.2: Voltage controller detection logic based on configuration.

Threshold Type	*v_zone_cfg*	*v_zone*	*v_detect*
Low threshold	0	0	1
		1	0
High threshold	1	0	0
		1	1

- ***dvm_detect***: indicates whether the DVM has detected a threshold crossing. Its logic is controlled by *dvm_err* and *dvm_err_cfg*, as shown in Table 3.3.

Table 3.3: DVM detection logic based on configuration.

Threshold Type	*dvm_err_cfg*	*dvm_err*	*dvm_detect*
Low threshold	0	0	0
		1	1
High threshold	1	0	1
		1	0

Now that all the input signals of the block have been thoroughly described, the next section analyzes how the internal Finite State Machine manages these signals to ensure robust and stable assist configuration.

3.3.1 Finite State Machine Operation

The internal behavior of the VLD Control Block is governed by a FSM, which ensures that assist configuration signals are asserted only when the voltage level has stably crossed a predefined threshold. This mechanism avoids false triggers due to noise, voltage oscillations, or calibration transients.

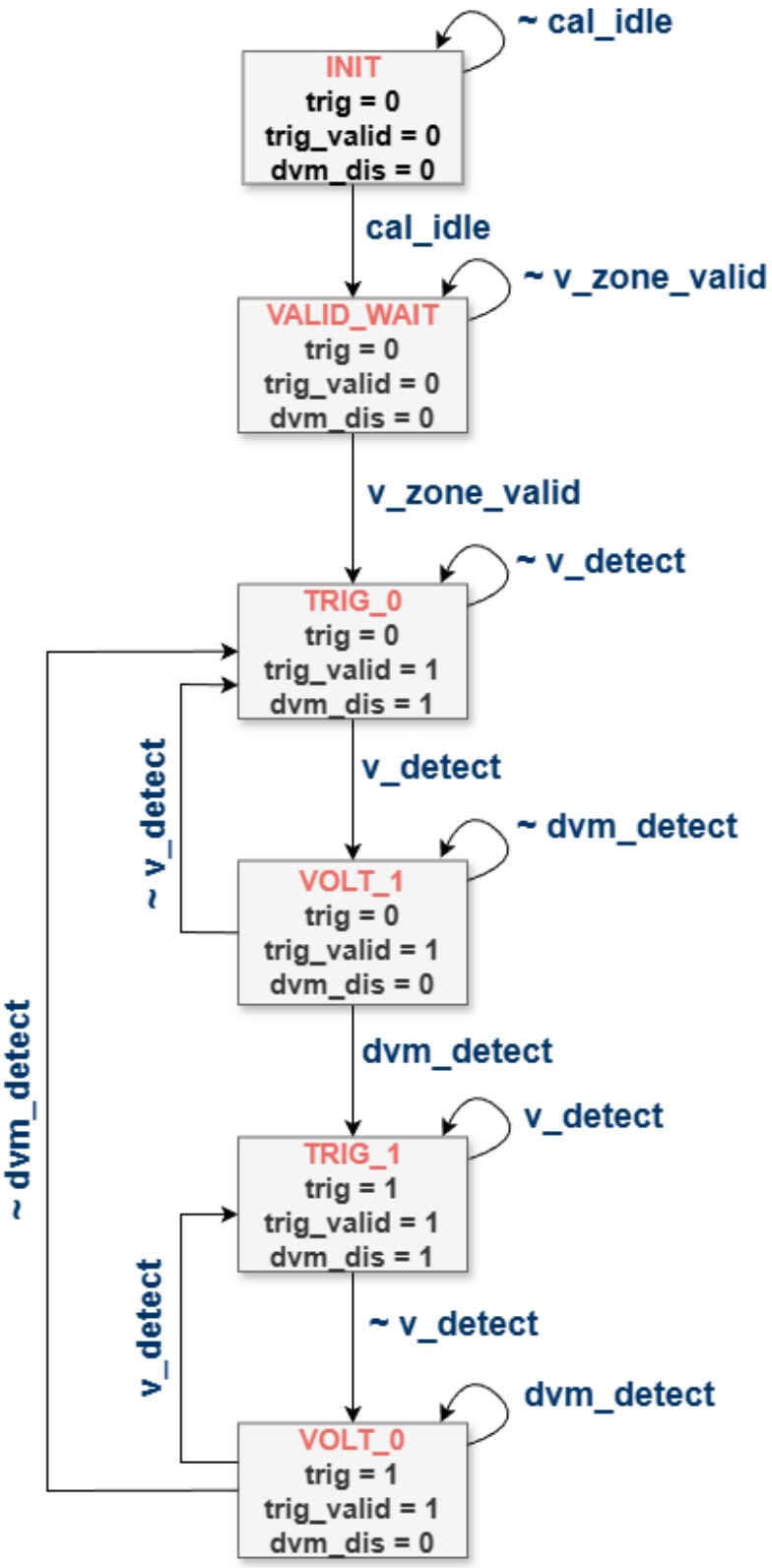

Figure 3.6: VLD Control block Finite State Machine.

I. Initialization

The FSM begins in the `INIT` state. In this state, the system remains idle and no assist trigger is generated. This ensures that no configuration changes occur while the system is undergoing calibration. The FSM remains in `INIT` until the *cal_idle* signal is asserted, indicating that the auto-calibration process has completed and the system is ready to operate.

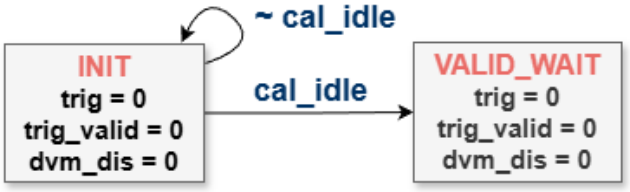

Figure 3.7: Initialization state.

II. Voltage Controller Validation

Once *cal_idle* = *1*, the FSM transitions to the `VALID_WAIT` state. Here, it waits for the *v_zone_valid* signal, issued by the Voltage Controller, to be asserted. This signal confirms that the voltage-level indication (*v_zone*) is valid and can be safely interpreted. Until this signal is high, the FSM does not proceed, ensuring that no decisions are made based on undefined or unstable voltage readings.

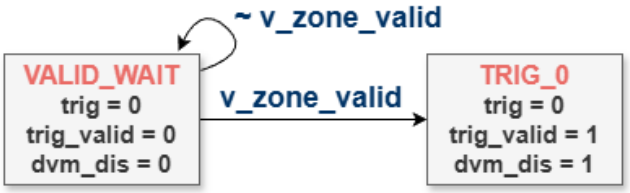

Figure 3.8: Voltage controller validation.

III. Trigger Zero

Once the voltage region is validated, the FSM enters the `TRIG_0` state. In this state, the *trig_valid* output is asserted to indicate that the system is ready to evaluate whether assist activation is required. However, the actual trigger signal *trig* remains low until further confirmation is received. The transition from `TRIG_0` to `VOLT_1` will happen only if the voltage controller signal *v_detect* transitions high, indicating the intention of the system to enter the voltage zone where the ACC bits need to be set. If, for any reason, the operating voltage does not follow the expected transition and *v_detect* returns to zero, the FSM reverts to the `TRIG_0` state. This mechanism

ensures that the trigger is asserted only when both the intention and the actual voltage condition are aligned.

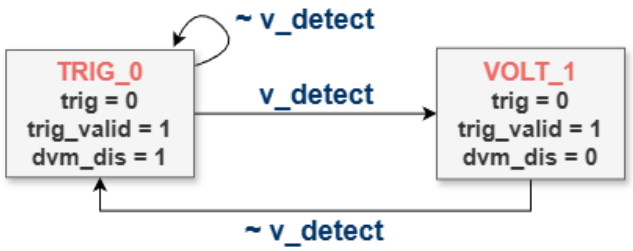

Figure 3.9: Trigger evaluation.

The transition from `TRIG_0` to `VOLT_1` is determined solely by the value of the *v_ detect* signal, which is derived from the voltage controller. This design choice ensures that the FSM reacts only to stable and intentional voltage transitions, rather than to transient or noisy DVM outputs. In this phase, the DVM can be disabled to save power, as its output is not required for the transition decision, by asserting *dvm_ dis* to zero.

IV. Trigger One

Upon entering `VOLT_1`, the FSM evaluates the *dvm_ detect* signal. This check ensures that the voltage transition is confirmed by a valid DVM output before asserting the assist trigger. If *dvm_ detect* is high, indicating the system is operating in the voltage region where ACC bits need to be set, then the FSM transitions to the `TRIG_1` state. The *trig* output is set high and the DVM is once again gated, waiting for another change in voltage controller indication.

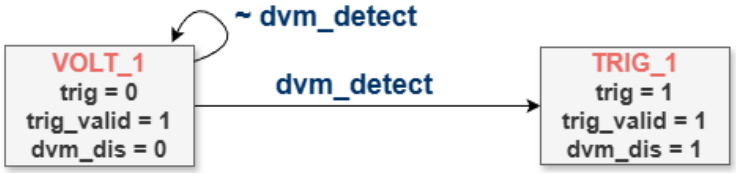

Figure 3.10: Final evaluation and trigger assertion.

The same logic applies in reverse when transitioning from `TRIG_1` to `TRIG_0`. If the voltage controller indicates that the system should exit the assist region (*v_ detect* goes low), the FSM transitions to `VOLT_0`, where it checks the *dvm_ detect* signal. If the DVM confirms that the voltage has exited the assist region, the FSM returns to `TRIG_0` and deasserts the *trig* signal.

At any point during operation, if the *cal_ idle* signal is deasserted, indicating that a new calibration cycle has started, the FSM immediately returns to the `INIT`

state. Similarly, if the *v_ zone_ valid* signal is lost, the FSM transitions back to the `VALID_WAIT` state, ensuring that no decisions are made based on invalid or undefined voltage information.

This FSM structure provides two key advantages:

1. **Robustness**: By prioritizing *v_ detect* over *dvm_ detect* in state transitions, the FSM avoids reacting to spurious DVM errors caused by voltage noise or oscillations;

2. **Power Efficiency**: During the TRIG_0 and TRIG_1 states, the DVM can be disabled to save power, since the FSM relies solely on the voltage controller signal for transitions.

This state machine-based control strategy ensures that assist configuration is only applied under well-defined and validated voltage conditions, significantly reducing the risk of false activations due to noise, glitches, or calibration transients. By decoupling the decision logic from raw DVM outputs and introducing a staged validation process, the FSM enhances the robustness and reliability of the system. Furthermore, the ability to selectively disable the DVM during stable phases contributes to overall power efficiency. The effectiveness of this approach is further demonstrated through simulation results, which are presented in the following section.

3.4 Simulation

To validate the proposed architecture, the additional control block of the VLD controller was first simulated in isolation using a linear SystemVerilog testbench. This allowed for early functional verification and debugging of the FSM logic and signal interactions. Once the block was verified, it was integrated into the pre-existing system, with appropriate modifications and interconnections between modules to ensure compatibility and correct operation.

Following integration, the entire architecture was simulated within a UVM-based verification environment. This setup enabled modular and realistic testing of the full system behavior under dynamic operating conditions, including voltage threshold crossings and assist trigger generation.

Figure 3.11 shows a waveform simulation of the VLD Control Block, illustrating the behavior of the FSM in response to various input conditions. Specific markers

highlight key transitions throughout the simulation. Below is a step-by-step explanation of each marked point:

(A) The FSM is initially in the `INIT` state, as the asynchronous reset *ares* is active. The signal *cal_ idle* is asserted, indicating that the calibration process has completed. Once the reset goes low, the FSM transitions to the `VALID_WAIT` state, where the FSM waits for a valid voltage indication.

(B) The signal *v_ zone_ valid* is asserted, confirming that the voltage controller has provided a valid indication. The FSM transitions to the `TRIG_0` state, preparing to evaluate whether assist activation is required. At this point, *trig_ valid* is set high, and *dvm_ dis* is asserted to temporarily disable the DVM. This prevents unnecessary toggling during stable operation and ensures that the system waits for a new voltage indication before re-enabling the monitor.

(C) The signal *v_ detect* is asserted, indicating that the system intends to operate in a voltage region that requires assist activation. The FSM transitions to the `VOLT_1` state, where it waits for confirmation from the DVM ($dvm\_\ dis = 0$).

(D) The signal *dvm_ detect* is asserted, confirming that the DVM has detected the voltage threshold crossing. The FSM transitions to the `TRIG_1` state, asserting the *trig* output to activate the assist mechanism. At this stage the DVM can be disabled.

(E) A new voltage indication is received from the controller, causing *v_ detect* to deassert. The FSM transitions to `VOLT_0`, preparing to deactivate the assist trigger once the DVM confirms the exit from the voltage region.

(F) The signal *dvm_ detect* is deasserted, indicating that the DVM no longer detects the voltage threshold condition. The FSM transitions back to the `TRIG_0` state, deasserting the *trig* output and disabling the assist mechanism. The *dvm_ dis* is set high.

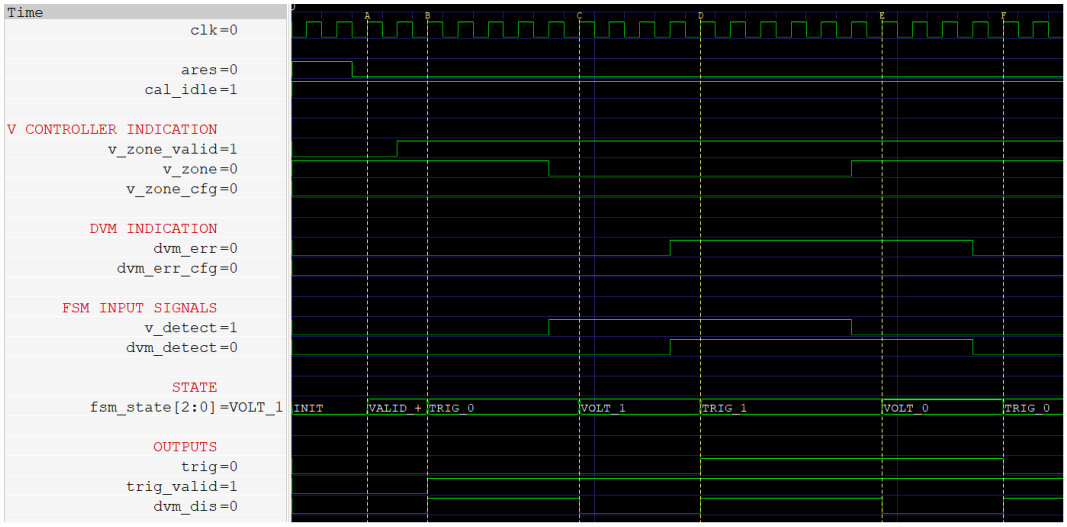

Figure 3.11: Simulation.

In Fig. 3.12, additional simulations are included to demonstrate alternative operating scenarios and validate the system's response under varying conditions.

(G) The FSM is in the `TRIG_0` state. The voltage controller indicates that the system intends to operate within a region where assist configuration is required ($v\_zone = 1$), so the FSM transitions to the `VOLT_1` state, where it waits for confirmation from the DVM.

(H) The FSM is in the `VOLT_1` state, and the *dvm_detect* signal is transitions high, confirming the voltage threshold crossing. As a result, the *trig* output is set high, activating the assist mechanism. Although *dvm_detect* subsequently oscillates due to transient noise or rail fluctuations, the *trig* signal remains stable. This behavior reflects the robustness of the control block, which is designed to suppress spurious transitions and prevent undesired toggling of the ACC bits, ensuring consistent assist configuration.

(I) A calibration cycle is simulated by deasserting the *cal_idle* signal. As a result, the FSM immediately transitions back to the `INIT` state, ensuring that all outputs are deasserted. This behavior guarantees that no assist configuration is applied during calibration, preserving system stability and preventing misconfiguration under potentially unstable conditions.

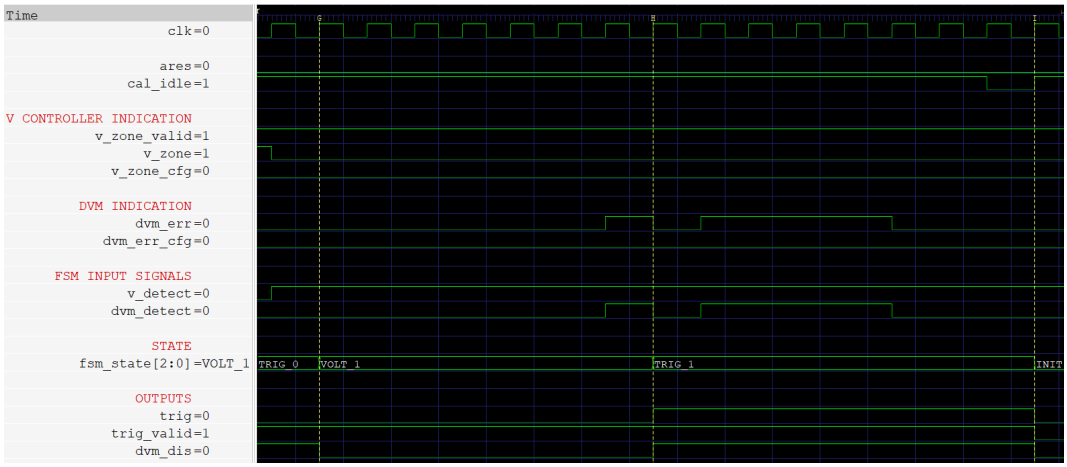

Figure 3.12: Simulation.

In addition to functional simulations, a set of **Design Rule Checks (DRC)** was performed to ensure the structural and architectural integrity of the design. These checks are aimed at identifying potential issues related to connectivity, logic synthesis, simulation behavior, and testability. They include analyses to detect structural inconsistencies, such as unconnected signals or logic loops, as well as verification of scan and test-related features to ensure proper controllability and observability of internal states. Furthermore, specific attention was given to low-power design aspects, including the validation of clock gating logic and its integration within the system. These analyses are essential to guarantee that the design adheres to best practices and is suitable for integration in complex SoC environments.

Furthermore, a dedicated analysis was conducted to validate the correctness of **Clock Domain Crossings (CDC)**. This verification step ensures that signals transferred between different clock domains are properly synchronized, avoiding metastability and timing violations. CDC checks are essential in complex digital systems where multiple asynchronous clock domains coexist, and they help guarantee reliable data communication and system stability across the entire architecture.

These checks confirmed that the architecture meets the required design constraints and is suitable for synthesis and integration in advanced SoC platforms.

3.5 Conclusion

This thesis has explored the design, implementation, and application of an all-digital Voltage Level Detector architecture for dynamic assist configuration in SRAM arrays.

The work is motivated by the increasing challenges posed by aggressive voltage scaling in modern CMOS technologies, where maintaining timing reliability and memory stability becomes critical.

The first chapter provided a comprehensive overview of the state of the art in dynamic variation tolerance, analyzing existing analog, hybrid, and digital techniques for voltage droop detection and mitigation. Particular attention was given to the limitations of analog comparators in terms of area, power, and scalability, which motivated the shift toward fully digital solutions.

Chapter 2 introduced the Dynamic Variation Monitor (DVM), a key building block of the proposed architecture. The chapter also detailed the auto-calibration circuit, which ensures accurate detection across different voltage, frequency, and process conditions. Applications such as Adaptive Clock Distribution and the DVM-based Security Monitor were presented to demonstrate the versatility of the DVM in both performance and security contexts.

Chapter 3 focused on the proposed VLD architecture, which extends the DVM framework to enable dual-threshold voltage detection for adaptive SRAM assist configuration. The system integrates two DVMs, each calibrated to a specific voltage threshold, and a control block that implements a Finite State Machine to manage assist triggers. A dual-signal validation mechanism, combining the DVM output with a voltage-level indication from a controller, ensures robust and noise-resilient operation. The architecture is fully digital, modular, and scalable, offering significant improvements in area, latency, and power consumption compared to analog alternatives.

The proposed VLD demonstrates a wide range of **technical benefits**, which make it a compelling solution for modern low-power, high-performance SoC environments:

- **All-Digital Implementation:** The VLD is entirely digital, eliminating the need for analog comparators or voltage references. This allows for seamless integration into standard digital design flows and avoids the complexity and area overhead associated with analog circuitry;
- **Dynamic Adaptability:** The system supports both high and low-threshold detection through configurable control bits, enabling real-time adaptation of SRAM assist strategies based on the current operating voltage;
- **Low Power Consumption:** The FSM-based control logic allows the DVM

to be selectively disabled when not needed, reducing dynamic power. Moreover, assist mechanisms are triggered only under validated voltage conditions, avoiding unnecessary toggling and energy waste;

- **Significant Area and Latency Reduction:** Compared to traditional analog voltage comparators, the DVM-based VLD is projected to achieve up to 90% reduction in both area and latency, based on internal estimations. This would be particularly impactful in designs where multiple voltage detectors are required;
- **Robustness to Noise and Oscillations:** By combining the DVM output with a voltage-level indication signal from a controller, the system avoids false triggers caused by transient voltage noise or rail oscillations. This dual-signal validation mechanism ensures that assist configuration changes are intentional and stable;
- **Scalability and Technology Portability:** The all-digital nature of the design ensures compatibility with advanced CMOS nodes and future technology generations. The architecture is modular and can be replicated to support multiple voltage thresholds, making it suitable for a wide range of applications and product tiers;
- **PVT Tolerance:** The auto-calibration loop dynamically adjusts the delay elements to maintain accurate detection across variations in process, voltage, and temperature. This guarantees reliable operation even under aggressive voltage scaling and environmental fluctuations;
- **Technology-Independent and Low-Overhead:** The VLD does not require additional voltage rails or analog biasing, making it a low-overhead solution that can be easily integrated into any processor architecture (CPU, GPU, NSP, modem) and across markets (datacenter, compute, mobile, IoT, automotive).

In conclusion, the DVM-based VLD architecture marks a significant step forward in the design of low-power embedded memory systems. Its fully digital nature, combined with a compact and modular structure, makes it particularly well-suited for modern SoCs where efficiency, scalability, and robustness are essential.

What makes this solution especially relevant is its direct applicability to SRAM assist circuits. These circuits are fundamental to ensuring reliable memory operation, particularly as technology nodes continue to scale down and supply voltages become more constrained. In such scenarios, maintaining read and write stability is increasingly difficult, and static assist configurations often lead to inefficiencies or overdesign.

The VLD addresses this challenge by enabling real-time, adaptive control of assist techniques, such as wordline underdrive or supply voltage collapse, based on the actual operating conditions. This dynamic approach ensures that assist mechanisms are activated only when needed, optimizing both performance and energy consumption. Thanks to its low overhead and configurability, the proposed architecture can be easily integrated into a wide range of systems, making it a valuable asset for next-generation SRAM-based designs.

Looking forward, several directions for **future research** could further enhance the flexibility and efficiency of the proposed architecture. One promising avenue is the redesign of the DVM structure to support the detection of multiple voltage thresholds within a single instance. This would eliminate the need for duplicating VLD blocks for each threshold, significantly reducing area and resource usage. At the same time, expanding the frequency adaptability of the system would allow the VLD to operate reliably across a broader spectrum of clock domains. The current implementation is optimized for high frequency, which limits its applicability in systems with low-frequency operation; by increasing the tuning range of the tunable delay elements, the DVM could maintain accurate timing margin detection across a wide range of frequencies, from ultra-low-power IoT devices to high-performance compute cores.

These enhancements would further consolidate the VLD as a key enabler for adaptive SRAM assist strategies in next-generation SoCs, combining configurability, compactness, and robustness in a single, fully-digital solution.

Appendix A

SRAM Read and Write Assist Strategies

With the continuous scaling of semiconductor technologies into the nanometer regime, both operating voltages and device dimensions have been significantly reduced. While this trend enhances energy efficiency, it also introduces considerable design challenges. One of the most critical issues is the reduced overdrive voltage, which intensifies the inherent trade-off between **read stability** and **write margin** in SRAM cells. To address these limitations, modern large-scale SRAM arrays increasingly incorporate **assist techniques**, circuit-level strategies aimed at improving the operational robustness of the 6-transistors (6T) SRAM cell by enhancing its read and write margins [17].

SRAM plays a pivotal role as the primary embedded memory in advanced CMOS technologies due to its simplicity, high speed, and compatibility with standard logic processes. The fundamental 6T SRAM cell comprises two cross-coupled inverters forming a **bi-stable latch**, which maintains one of two stable logic states. This latch is accessed via two NMOS pass-gate transistors that connect the internal nodes to the bitlines under the control of a wordline signal. The pull-up devices (typically PMOS transistors) and pull-down devices (NMOS transistors) constitute the core of the latch, while the pass transistors facilitate read and write operations. As a **volatile** memory, it loses stored data when power is removed. This compact and effective design supports high-density integration without requiring additional fabrication steps, making SRAM a cost-efficient and reliable solution for on-chip memory in digital systems.

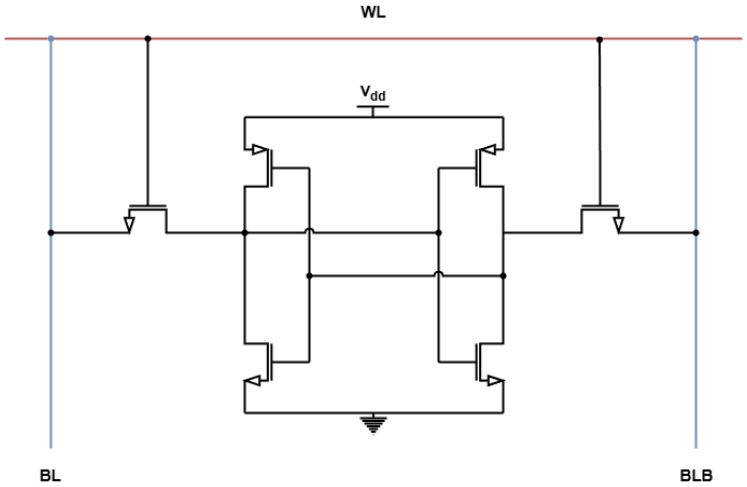

Figure A.1: 6T SRAM cell.

Several factors can contribute to read and write failures in SRAM cells, which are generally categorized into two main types: **hard failures**, typically caused by manufacturing defects, and **soft failures**, which are associated with operational issues such as the inability to read, write, retain data, or maintain stability during a read operation.

Among the various causes of soft failures, a key concern is the **stability** of the SRAM cell under noise and voltage fluctuations. In this context, a widely used metric to quantify stability is the **SNM**, which measures the maximum tolerable noise voltage that does not alter the stored data. A low SNM increases the likelihood of read or write failures, especially under aggressive voltage scaling or process variations.

The SNM is defined as the amount of voltage noise required at output nodes to flip the state of a cell [18]. It can be evaluated by plotting the voltage transfer characteristc of the two cross-coupled inverters. The SNM is the side of the larger square that can be fitted between the inverters' characteristic, as shown in Fig. A.2.

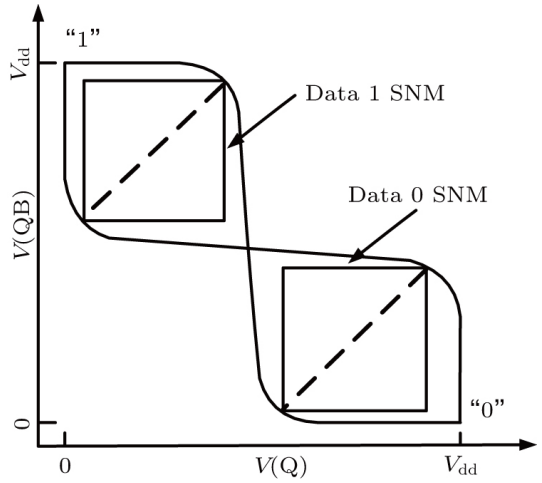

Figure A.2: Voltage transfer characteristics of cross-coupled inverters in SRAM.

During a read operation, the stability of an SRAM cell is quantified by the *read SNM*. This represents the maximum noise voltage that the cell can tolerate without altering the stored bit. The SNM is evaluated under conditions where both bit lines are precharged to a logic high level and the word line is driven to the supply voltage (V_{dd}). If the cell stores a logic '0', the corresponding internal node may be pulled upward through the access transistor, potentially degrading the SNM. This degradation can be severe enough to cause a *destructive read*, where the stored data is unintentionally flipped.

The ability of the cell to accept new data during a write operation is measured by the *write SNM*, defined as the minimum voltage disturbance required to successfully flip the state of the cell. If the write margin is smaller than the noise present in the system, a *write failure* may occur, preventing the correct data from being stored.

A.1 Read Operation

The 6T SRAM cell performs read operations through access transistors controlled by a Word Line (WL) and connected to complementary bitlines (Bit Line (BL) and Bit Line Bar (BLB)). Before a read access, both bitlines are precharged to the supply voltage (V_{dd}) using a precharge and equalization circuit, ensuring a known and balanced initial condition.

When the wordline is asserted, the access transistors connect the internal storage nodes of the cell to the bitlines. Depending on the stored value, one of the bitlines begins to discharge slightly through the pull-down path of the cell, creating a small voltage differential between BL and BLB. This differential is then detected and amplified by a **sense amplifier**, which produces a full-swing digital output.

Read performance is highly sensitive to process variations, device mismatches and parasitic effects. These factors can degrade the development of the bitline differential and compromise read reliability. In particular, read failures may occur under the following conditions:

- The NMOS pull-down transistor is too weak to discharge the bitline effectively, resulting in a small or slow voltage differential;

- The NMOS pass-gate (access transistor) is too strong relative to the pull - down device, potentially injecting charge into the internal node and causing a destructive read.

Additionally, parasitic effects such as increased bitline capacitance or resistance in the path to the sense amplifier can further degrade signal integrity. Sense amplifiers themselves are also subject to mismatch due to layout and process variations. Given the strict area constraints in high-density memory arrays, these amplifiers are typically compact, which limits their robustness against mismatch and noise. As a result, read failures may occur when the voltage differential between the bitlines is insufficient by the time the sense amplifier is triggered, leading to incorrect data being latched and propagated.

To ensure reliable operation and prevent data corruption, careful transistor sizing is essential. In particular, the ratio between the pull-down and access transistors must be optimized. As shown in Fig. A.3, if the access transistor is too strong relative to the pull-down device, the internal node storing a logic '0' may be inadvertently pulled high during a read due to charge injection from the BL to the internal nodes, resulting in a *destructive read* [18, 19].

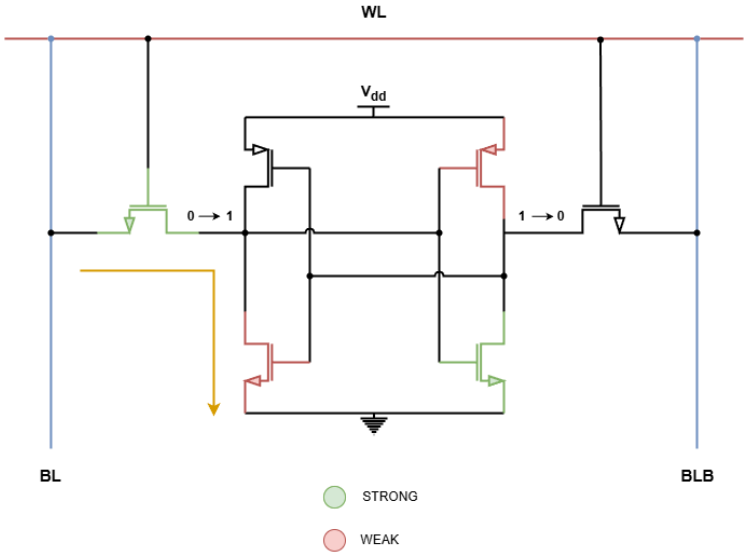

Figure A.3: Destructive read.

A.2 Write Operation

During a write operation, the BLs are driven with the data to be written, and the WL is asserted. The access transistor must overpower the pull-up PMOS to flip the internal node storing the opposite value. Write-ability depends on the relative strength of the access and pull-up transistors.

The primary mechanism during a write operation involves discharging the internal storage node that is initially at a high logic level. This task is predominantly carried out by the NMOS pass-gate transistor. In a successful write operation, the pass-gate must be sufficiently strong to overcome the pull-up PMOS transistor of the inverter. This ensures that the high node is pulled low enough to initiate a state flip in the cross-coupled inverter pair. The write operation may fail under the following conditions:

- The NMOS pass-gate is too weak to discharge the node effectively;
- The PMOS pull-up transistor is too strong, resisting the discharge;
- The opposing NMOS pull-down transistor is excessively strong, making it difficult to flip the cell state.

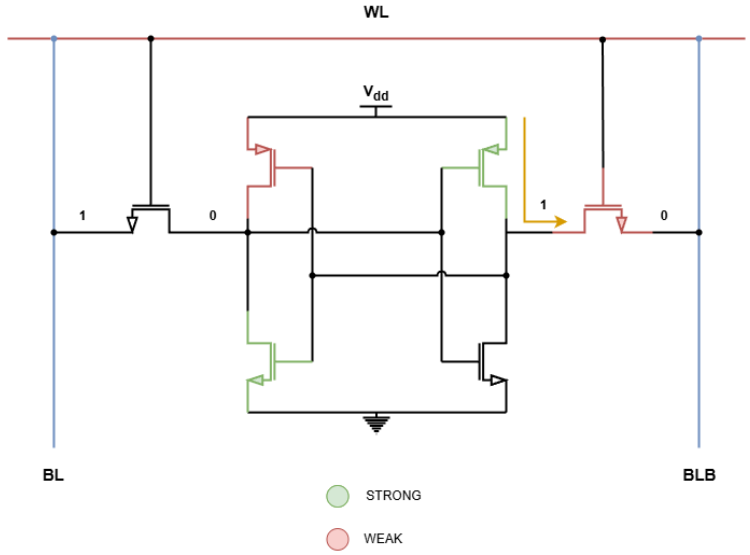

Figure A.4: Write failure.

When the write operation functions correctly, the wordline is asserted, causing the internal node to discharge, while the complementary node begins to rise. However, in marginal or failing cases, that discharge may not be sufficient to trigger the bistable inverter pair to switch states. This incomplete transition results in a *write failure*, as in Fig. A.4.

A.3 Assist Circuits

After analyzing the fundamental read and write operations of the 6T SRAM cell, as well as the various failure mechanisms associated with the bitcell and its peripheral circuitry, it becomes evident that ensuring robust operation requires careful margin management. Memory designers face the challenge of balancing conflicting requirements, such as read stability, write-ability, and data retention, under PVT variations.

To address these challenges, a variety of circuit-level techniques have been developed over the years. These techniques, commonly referred to as **assist circuits**, are specifically designed to enhance the reliability and performance of SRAM bitcells. The term 'assist' reflects their role in supporting the bitcell during critical operations, particularly under aggressive scaling and low-voltage conditions.

In essence, assist circuits enable the SRAM to operate with improved margins, often allowing for lower supply voltages without compromising functionality [20].

A.3.1 Read Assist Techniques

Read assist circuits are critical in addressing read stability failures in SRAM cells, which can occur due to the delicate balance between performance and robustness at low voltages. Tab. A.1 provides a summary of the main techniques.

Assist	**Increase Global V_{dd}**	**Negative V_{ss} at Bitcell**	**Boosted V_{dd} at Bitcell**	**Wordline Underdrive**
Description	Coupling of low bitline below ground to increase pass-gate drive	Collapse of active column cell supply to weaken latch strength	Increase of active column ground supply to weaken latch strength	Decrease in wordline voltage during read to reduce pass-gate strength
Type	Strengthens Latch	Strengthens Latch	Strengthens Latch	Reduce Noise into Latch
Adoption	Uncommon	Uncommon	Uncommon	Common

Table A.1: Comparison of SRAM read assist techniques.

Among the various techniques, **Wordline Underdrive** is the most widely adopted across the industry. This method involves reducing the maximum voltage applied to the wordline during a read operation. By trimming the wordline voltage, the strength of the pass-gate transistor is moderated, thereby minimizing the risk of charge injection into the storage node and improving read stability.

Other less common techniques include *Global V_{dd} Increase*, *Negative V_{ss} at Bitcell*, and *Boosted V_{dd} at Bitcell*. Increasing the global V_{dd} can improve read margins but is generally avoided due to the high power cost and the complexity of dynamically adjusting a large power net. Similarly, applying a negative reference voltage V_{ss} to the bitcell or boosting the local V_{dd} can enhance read stability, but these approaches often require complex circuitry and can negatively impact read performance due to the introduction of stacked devices in critical paths. Consequently, while these methods are theoretically viable, they are not widely implemented in practice. Wordline underdrive remains the preferred solution due to its effectiveness and relative simplicity in integration.

A.3.2 Write Assist Techniques

Among the various write-assist techniques employed in modern SRAM design, shown in Tab. A.2, the two most commonly adopted are **Negative Bitline** and **Supply Voltage Collapse**.

Assist	**Negative Bitline**	**Supply Voltage Collapse**	**Ground Voltage Increase**	**Wordline Boost**
Description	Coupling of low bitline below ground to increase pass-gate drive	Collapse of cell supply voltage to weaken PMOS pull-up	Increase of ground voltage to weaken latch strength	Boost wordline voltage to enhance pass-gate drive
Type	Strengthens pass-gate	Reduces latch strength	Reduces latch strength	Strengthens pass-gate
Adoption	Common	Common	Uncommon	Uncommon

Table A.2: Comparison of SRAM write assist techniques.

The Negative Bitline technique involves temporarily coupling the bitline to a voltage level below ground: this results in a significantly increased gate-to-source voltage across the pass-gate transistor, thereby enhancing its drive strength. This method is particularly effective in improving write success rates, even in marginal bitcells that may arise due to manufacturing variations. As such, it is widely implemented in contemporary SRAM architectures.

On the other hand, the Supply Voltage Collapse technique operates by momentarily reducing the supply voltage to the SRAM cell. This action weakens the PMOS pull-up transistor, thereby reducing the internal latch strength of the bi-stable inverter pair. The reduced contention between the pass-gate and the pull-up device facilitates easier bitcell overwriting during write operations. This approach is especially beneficial for lowering the minimum operating voltage of the memory array, thereby improving energy efficiency.

Other techniques, such as *Ground Voltage Increase* and *Wordline Boost*, are less commonly used. Ground Voltage Increase functions similarly to supply collapse but introduces additional stacked circuitry in the discharge path, which complicates the

design and limits its adoption. Wordline Boost, while capable of enhancing pass-gate drive by increasing the wordline voltage, poses a risk of read stability failures in non-targeted cells along the same row. This trade-off has limited its widespread use.

Overall, Negative Bitline and Supply Voltage Collapse remain the dominant strategies in state-of-the-art SRAM write-assist circuit design.

Bibliography

[1] Keith Bowman, James Tschanz, Chris Wilkerson, Shih-Lien Lu, Tanay Karnik, Vivek De, and Shekhar Borkar. Circuit techniques for dynamic variation tolerance. In *Proceedings of the 46th Annual Design Automation Conference (DAC)*, pages 4–7. ACM, 2009.

[2] Keith A. Bowman, Carlos Tokunaga, Tanay Karnik, Vivek K. De, and Jim W. Tschanz. A 22nm dynamically adaptive clock distribution for voltage droop tolerance. In *2012 Symposium on VLSI Circuits (VLSIC)*, pages 94–95. IEEE, 2012.

[3] K. Bowman, J. Bridges, S. Raina, Y. Kolla, J. Jeong, F. Atallah, W. Flederbach, and J. Fischer. Automatic calibration circuits for operational calibration of critical-path time delays in adaptive clock distribution systems, and related methods, and systems. US Patent 9,413,344, August 2016.

[4] D. R. Pal, H. Pant, A. Roy, S.-H. Hu, and K. Bowman. Adaptive voltage controller. US Patent 11,249,530, February 2022.

[5] Assia El-Hadbi, Oussama Elissati, and Laurent Fesquet. Time-to-digital converters: A literature review and new perspectives. *IEEE*, 2019.

[6] Keith A. Bowman, Carlos Tokunaga, Tanay Karnik, Vivek K. De, and James W. Tschanz. A 22 nm all-digital dynamically adaptive clock distribution for supply voltage droop tolerance. *IEEE Journal of Solid-State Circuits*, 48(4):907–916, April 2013.

[7] Adrian Tang, Simha Sethumadhavan, and Salvatore Stolfo. CLKSCREW: Exposing the Perils of Security-Oblivious Energy Management. In *26th USENIX Security Symposium (USENIX Security 17)*, Vancouver, BC, Canada, 2017. USENIX Association.

[8] H. Pant, R. Jain, S. Shahrokhinia, and L. Ho. Power multiplexing with an active load. US Patent 10,317,968 B2, June 2019.

[9] Keith A Bowman, Sarthak Raina, J Todd Bridges, Daniel J Yingling, Hoan H Nguyen, Brad R Appel, Yesh N Kolla, Jihoon Jeong, Francois I Atallah, and David W Hansquine. A 16 nm all-digital auto-calibrating adaptive clock distribution for supply voltage droop tolerance across a wide operating range. *IEEE Journal of Solid-State Circuits*, 51(1):8–20, 2016.

[10] Vijay Kiran Kalyanam, Eric Mahurin, Keith Bowman, and Jacob Abraham. A proactive voltage-droop-mitigation system in a 7nm hexagon processor. In *2020 IEEE Symposium on VLSI Circuits (VLSI)*, pages 1–2. IEEE, 2020.

[11] Aydin Dirican, Cagatay Ozmen, and Martin Margala. Leakage-aware droop measurement built-in self-test circuit for digital low-dropout regulators. *Journal of Electronic Testing*, 34(4):405–415, 2018.

[12] Samantak Gangopadhyay, Saad B. Nasir, A. Subramanian, Visvesh Sathe, and Arijit Raychowdhury. Uvfr: A unified voltage and frequency regulator with 500mhz/0.84v to 100khz/0.27v operating range, 99.4 In *ESSCIRC Conference 2016: 42nd European Solid-State Circuits Conference*, pages 321–324, 2016.

[13] Jiliang Liu, Huidong Zhao, Zhi Li, Kangning Wang, and Shushan Qiao. A self-calibrated unified voltage-and-frequency regulator system design based on universal logic line circuit. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 33(2):593–597, 2025.

[14] DongHoon Jung, Dongha Lee, Seki Kim, Susie Kim, Min Young Kang, Takahiro Nomiyama, Dongsu Kim, and Jongwoo Lee. A 4ghz, 0.69%-accuracy voltage-droop detector with multiple remote sensing and under 2-cycle detection latency in 2nm gaafet. In *2025 IEEE International Solid-State Circuits Conference (ISSCC)*, pages 164–165. IEEE, 2025.

[15] James Tschanz, Nam Sung Kim, Saurabh Dighe, Jason Howard, Gregory Ruhl, Sriram Vangal, Siva Narendra, Yatin Hoskote, Howard Wilson, Carol Lam, Matthew Shuman, Carlos Tokunaga, Dinesh Somasekhar, Stephen Tang, David Finan, Tanay Karnik, Nitin Borkar, Nasser Kurd, and Vivek De. Adaptive frequency and biasing techniques for tolerance to dynamic temperature-voltage variations and aging. In *2007 IEEE International Solid-State Circuits Conference (ISSCC)*, pages 292–293. IEEE, 2007.

[16] B. K. Rangarajan, D. R. Pal, K. A. Bowman, S. Turaga, A. D. De, J. Hu, and C. Agarwalla. On-die voltage-frequency security monitor. US Patent 11,880,454, January 2024.

[17] Divya Suneja, Nitin Chaturvedi, and S. Gurunarayanan. A comparative analysis of read/write assist techniques on performance & margin in 6t sram cell

design. In *2017 International Conference on Computer, Communications and Electronics (Comptelix)*, pages 659–664. IEEE, 2017.

[18] Christiensen D.C. Arandilla, Anastacia B. Alvarez, and Christian Raymund K. Roque. Static noise margin of 6t sram cell in 90-nm cmos. In *2011 UKSim 13th International Conference on Modelling and Simulation*, pages 534–539, University of the Philippines – Diliman, Quezon City, Philippines, 2011. IEEE.

[19] QuocDat Tai Nguyen. Read/write assist circuits and sram design. Master of science in engineering report, The University of Texas at Austin, Austin, TX, December 2009. Supervised by Abraham A. Jacob.

[20] Eric Karl. Fundamentals of ultra-low voltage embedded memory design. In *2023 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE, 2023. Live Q&A Session: February 20, 2022.