POWER MANAGEMENT TECHNIQUES FOR ULTRA-LOW VOLTAGE INTEGRATED CIRCUITS

By:

Md Shazzad Hossain

A Doctoral Thesis

Supervised By:

Professor Ioannis Savidis

Department of

Electrical and Computer Engineering

Drexel University

Philadelphia, Pennsylvania

2021

GRADUATE THESIS APPROVAL FORM AND SIGNATURE PAGE

Instructions: This form must be completed by all doctoral students with a thesis requirement. This form MUST be included as page 1 of your thesis via electronic submission to ProQuest.

Thesis Title: Power Management Techniques for Ultra-low Voltage Integrated Circuits

Author's Name: MD Shazzad Hossain

Submission Date: 12/15/2021

The signatures below certify that this thesis is complete and approved by the Examining Committee.

Role: Chair Name: Baris Taskin
Title: Professor
Department: Electrical & Computer Engineering
Approved: Yes Date: 12/16/2021

Role: Member Name: Ioannis Savidis
Title: Associate Professor
Department: Electrical & Computer Engineering
Approved: Yes Date: 12/17/2021

Role: Member Name: Nagarajan Kandasamy
Title: Professor
Department: Electrical & Computer Engineering
Approved: Yes Date: 12/16/2021

Role: Member Name: Bahram Nabet
Title: Professor
Department: Electrical & Computer Engineering
Approved: Yes Date: 12/16/2021

Role: Member Name: Geoffrey Mainland
Title: Associate Professor
Department: Computing
Approved: Yes Date: 12/16/2021

Contents

List of Tables

List of Figures

Abstract

Energy efficient computing is one of the primary requirements of ubiquitous battery-operated edge devices including mobile phones, wearables, active implantable medical devices, unmanned aerial, autonomous, and electric vehicles, and hardware modules used in smart IoT infrastructure and cyber-physical systems. One of the primary sources of loss in energy efficiency is the leakage current of logic and memory circuits. In addition, the emergence of on-device intelligence requires a greater number of computations, increased storage per device, and an increased number of operations per watt. Operating circuits at ultra-low voltages including near- and sub-threshold is an effective means to improve the energy efficiency. However, circuit techniques, algorithms, and power management methodologies are required that are tailored to ultra-low voltage operation. Moreover, novel techniques and architectures are needed to improve the overall energy efficiency of circuits that implement neural network models. Techniques to reduce and conserve the leakage current of an IC are, therefore, essential to further minimize the consumed power of a circuit.

In this research, novel circuits and algorithms are developed that allow for robust and energy efficient ultra-low voltage operation. Completed research tasks include the development of dynamic differential signaling based logic (DDSL) families, interface circuits that allow for signal propagation between ultra-low voltage and nominal

voltage power domains, supply and threshold voltage optimization, and clock-power co-design for sub-threshold operation. The developed DDSL families operate between 400 mV and 450 mV and reduce the total power consumption to 0.96× that of CMOS logic circuits, while improving the performance and noise margin by, respectively, 1.4× and 2.5×. The developed interface circuits include a single ended level shifter and a bidirectional input/output (I/O) circuit, which efficiently convert an input voltage of 450 mV to an output voltage of 3.3 V while consuming only 4.87 mW. The developed clock-power co-design methodology provides a robust power supply voltage of 250 mV to the clock distribution network while assuring resilience to as much as 10% noise on the power network.

A novel power management technique defined as 'leakage reuse' is developed to recycle the leakage current from idle circuits or cores and deliver the recycled energy to active circuit blocks or cores. The circuit techniques, scheduling algorithms, and the determination of the energy break-even point are completed for the leakage reuse technique. The power delivery to sub-V_t circuits through leakage reuse reduces the total power consumption to 0.41× that of a conventional power delivery system when generating a supply voltage of 380 mV from the recycled leakage current of circuits operating at 1.2 V. Two scheduling algorithms are developed that allow for the simultaneous execution of power gating and leakage reuse and reduce the total power consumption by 50.2% as compared to a baseline topology that includes neither leakage reuse or power gating.

In addition, techniques and methodologies are developed to recycle leakage energy from idle on-chip SRAM memories. Moreover, a novel multi-voltage domain heterogeneous accelerator architecture is developed for energy efficient execution of neural

network models. A near-memory computing architecture is developed for the heterogeneous accelerator that leverages the leakage reuse technique applied to an SRAM memory array, where the leakage current of idle memory banks within each processing element is utilized to deliver current to the adjacently placed multiply-and-accumulate (MAC) units. The proposed heterogeneous architecture with leakage reuse results in an energy efficiency of 3.27 tera-operations per second per watt (TOPS/W) as compared to a conventional monolithic and single voltage domain architecture that exhibits an energy efficiency of 0.0458 TOPS/W.

Chapter 1

Introduction

Energy efficient computing is the key driving force behind the development of state-of-the-art integrated circuits (ICs) constrained by a stringent power budget. Power consumed by ICs is rapidly growing as the commercial and government sectors in countries around the world are increasingly applying automation across multiple business and defense sectors as well as the larger demand for mobile devices and systems that require greater computational power. The electronic infrastructure and devices needed for the information communication technology (ICT) sector contribute 10% of global electricity consumption [1]. From the very first use of integrated circuits, improvements in transistor scaling, transistor density, and performance were the fundamental objectives targeted by both academia and industry [2–5]. In more recent years, the significant growth in battery operated or energy harvested mobile devices, ubiquitous edge devices with on-device intelligence, and internet of things

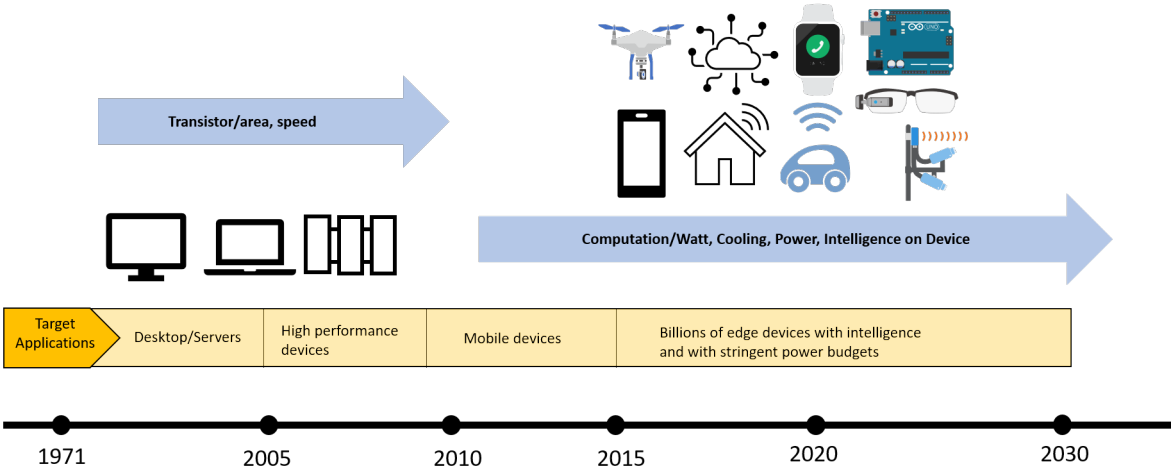


Fig. 1.1: Past, present, and future driving forces of the IC market [2, 4–10, 16, 18, 19].

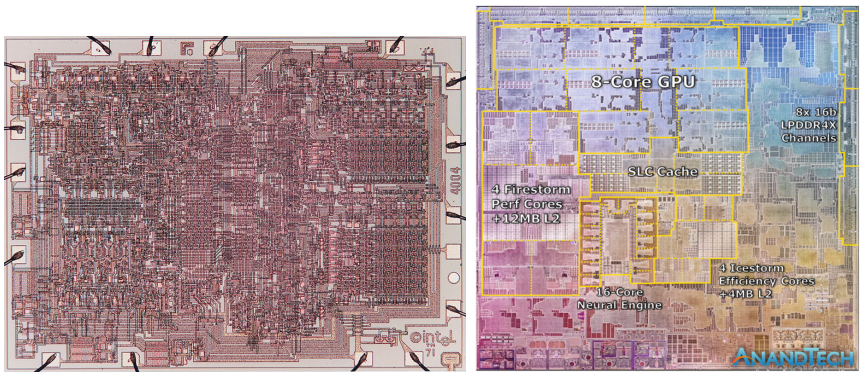
(IoT) devices for smart cities, homes, and transportation has shifted the focus to more energy efficient and reliable computing [6–10]. An overview of past, present, and future market driving forces of the semiconductor industry is provided in Fig. 1.1. Integrated circuits for desktop and server class computers dominated the semiconductor market beginning in the 1980's through the mid 2000's. From the mid 2000's and continuing to the present and beyond, growth in the IC market is dominated by ubiquitous battery operated or battery-less devices, where energy efficiency is the primary design objective [6–17].

Many techniques have been developed at the circuit, architecture, and system level to improve the energy efficiency of an IC by reducing the dynamic power, the leakage power, or both simultaneously [18, 20–29]. Since the early 2000's, dynamic supply voltage scaling has been implemented to reduce both the leakage and dynamic

power consumption of a circuit [18, 20–22, 30–33], while dynamically scaling the supply voltage to as low as the sub-threshold region of operation of a transistor has gained additional traction in the last decade [22, 24, 25]. Operating circuits with a supply voltage that is less than the threshold voltage of the transistors (sub-threshold) is an effective method to significantly reduce the energy per operation for applications where performance is not of critical importance [22, 32, 33]. Therefore, circuits operating at sub-threshold are suitable for mobile and edge devices, where a smaller energy per operation is more desired than high performance computing [22, 33–35].

Operating circuits with a supply voltage that is slightly above the threshold voltage of the transistors (near-threshold) is another effective technique that provides an optimum energy and performance point. Near-threshold operation allows for lower energy per operation without significantly impacting the performance of the circuit [18, 21, 30, 31]. In order to take advantage of operating circuits at ultra-low voltages, the development of novel optimized circuits and algorithms for logic, interfacing, clock distribution, and power management is needed.

The chapter begins with an introduction to the importance of low power design. The energy loss due to the leakage current of state-of-the-art ICs is described in Section 1.2. The energy loss due to idle memories is discussed in Section 1.3. An overview of low power techniques that are implemented at the circuit, architecture,



(a) (b)

Fig. 1.2: Microphotographs revealing the evolution of transistor density in microprocessors: (a) Intel 4004 processor that was released in 1971 with 2250 transistors [36] and (b) Apple M1 processor that was released in 2020 with 16 billion transistors [37].

and system level is provided in Section 1.4. An outline of the research proposal is provided in Section 1.5.

1.1 Low Power Design of Integrated Circuits

The primary advantages of low power circuits implemented for mobile and edge devices include improvement in a) computational throughput for a given energy budget, b) mobility by increasing battery lifetime, c) portability by requiring smaller or no battery, d) security by allowing the storing and computation of data at the edge, e) latency by avoiding interfacing with the cloud, g) cost, and h) environmental pollution by reducing the carbon footprint due to the energy use of the ICs. As mentioned, during the early stages of semiconductor development, the improvement of the computing performance of integrated circuits was the primary objective. The

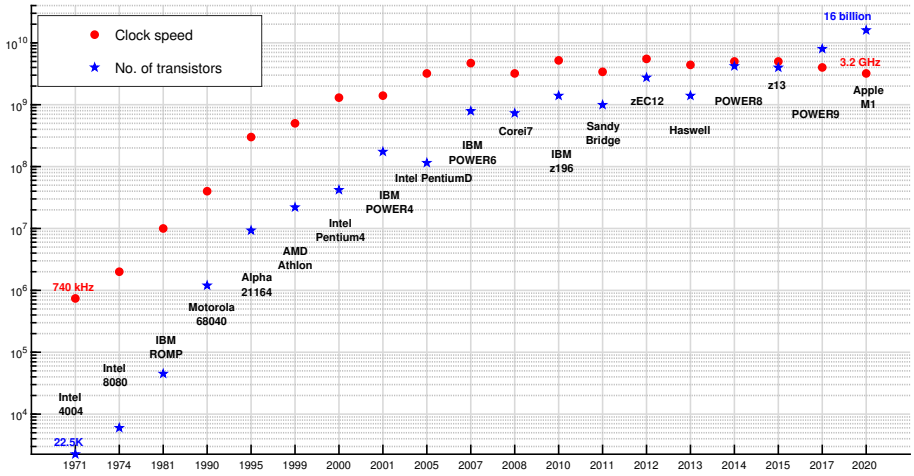


Fig. 1.3: Increase in transistor count from 2250 devices to over 15 billion, which limits the maximum clock frequency due to thermal and power budget constraints [2–4, 19, 37–40].

first commercial processor, the 4-bit Intel 4004 consisting of only 2250 transistors, was produced in 1971 and operated at a clock frequency of 740 kHz [38]. Since the introduction of the Intel 4004, the semiconductor industry has rapidly progressed, where the operating frequency of processors now exceeds 4 GHz and there are well over 5 billion transistors implemented in a single IC [2–4, 19, 37–40]. A micrograph of the first Intel processor and the latest Apple M1 processor is shown in Fig. 1.2. The Apple M1 processor consists of 16 billion transistors with a word size of 64 bits, which indicates a rapid increase in the transistor density and computational throughput over the past 50 years [37].

As shown in Fig. 1.3 the number of transistors per IC has significantly increased to well over a billion devices, The increase in transistor density has resulted in new

challenges beyond ones due solely to performance driven semiconductor growth which include a) limited cooling capacity and b) a limited power budget. The improvement in clock frequency has, therefore, flattened, as indicated by the red dots in Fig. 1.3. In addition, the scaling of the supply voltage has also stagnated beginning with the 65 nm technology node, as shown in Fig. 1.4, due to the technological limit of further reducing the transistor threshold voltage [41]. The stagnation in the reduction of the supply voltage and the increased energy loss due to leakage in state-of-the-art ICs results in a decrease in the overall energy efficiency of the circuit (more discussion on leakage loss is provided in Section 1.2) [41, 42].

While the maximum clock frequency of computing cores has saturated, the number of edge and IoT devices has grown exponentially in recent years [6, 7, 43, 44]. Edge devices operate under stringent energy efficiency requirements and cover a broad spectrum of applications that include mobile devices, smart cities, smart homes, self-driving cars, unmanned aerial vehicles, and unmanned underwater vehicles. Recent advancements in on-device artificial intelligence further increase the demand for energy efficient edge devices as improved computation is needed within a highly constrained energy budget [8–10]. The global market for edge computing is projected to rise to $15.7 billion dollars by 2025 from the current market value of $3.6 billion dollars in 2020 [45]. Therefore, the current and projected future demand of battery operated or battery less edge devices requires novel methods and circuit techniques

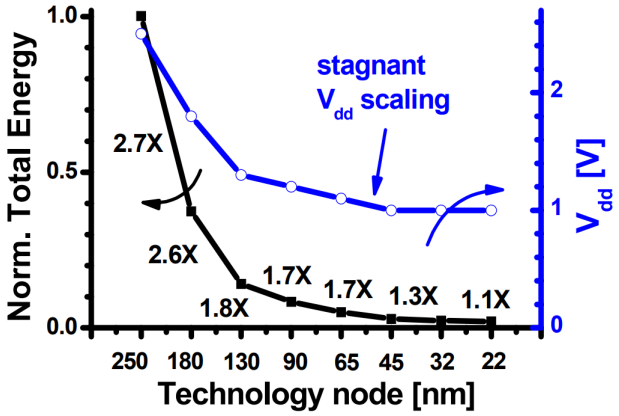


Fig. 1.4: Stagnation in the improvement of both the supply voltage and corresponding energy efficiency with technology scaling [41].

to address unique design challenges and target objectives. Additional design considerations include robustness to process variation and supply noise, in addition to robustness to ambient temperature and humidity variations as edge devices are found in multiple environments.

1.2 Power Loss due to Leakage Current in State-of-the-art Integrated Circuits

Power loss due to leakage significantly reduces the energy efficiency of mobile devices with stringent power budgets. For a given application and implementation, technology scaling provides improved computation per unit area. However, there is a greater rate of increase in the leakage power consumption than the dynamic power

consumption [42] as shown in Fig. 1.5, where the leakage power consumption as a percentage of the dynamic power consumption for different technology nodes is plotted. For older process technology nodes, the leakage power was not a significant concern as the dynamic power consumption was always the dominant component. Beginning with the 130 nm technology node, an increasing proportion of the total power consumption is contributed to leakage. The advent of FinFET technology introduced a new device structure that resulted in less leakage current than bulk CMOS technology due to better control of the transistor channel. However, the reduction in the leakage current due to FinFET technology is in comparison to previous technology node, while the amount of leakage as a percent of the total power consumption still remains high. For state-of-the-art FinFET technologies, the contribution of leakage power to the total power consumption exceeds 15% [46]. In addition, the leakage power of FinFET based circuits increases exponentially as the temperature increases [47–49]. Further, although operating CMOS and FinFET based circuits at a lower supply voltage (near-threshold and sub-threshold) improves the energy efficiency, the percentage of the total power attributed to the leakage current of the cores and system-on-chip is larger as the leakage power consumption is comparable to the dynamic power consumption at low operating voltages [24, 50].

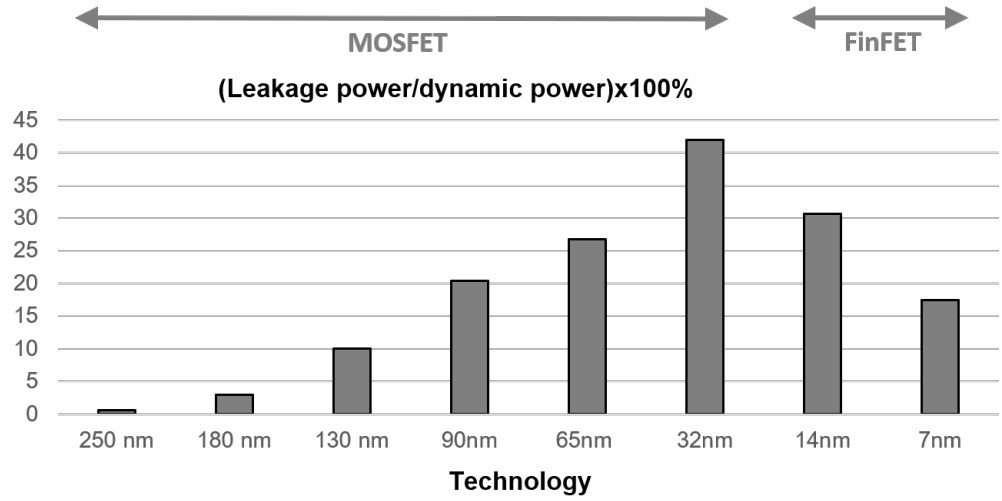


Fig. 1.5: Comparison of leakage and dynamic power for different technology nodes [17, 24, 46, 46, 50–52].

1.3 Leakage Loss in Memory

The energy consumption of an integrated circuit that is not actively switching or a memory circuit that is not reading or writing any data is considered as loss. The energy loss during inactive periods is inevitable in modern integrated circuits, where combinational circuits, sequential circuits, analog blocks, and memory circuits implemented within CPUs, GPUs, FPGA, hardware accelerators, and system-on-chips (SoCs) all contribute leakage current [53–56]. As the number of battery operated devices increasing and as for the majority of the time the full hardware resources are not utilized, the amount of wasted energy also increases.

The leakage current of a single 6T SRAM cell, a 16×16 cell array, and a 16×16

Table 1.1: Characterization of leakage power and access time of SRAM with standard-V_t transistors, high-V_t transistors, and transistors with 25% longer channel length.

	One 6T bit-cell	**16×16 cell array**	**16×16 memory bank**	**Access time of a 6T memory cell**	
				Read	**Write**
Standard-V_t and L	95.25 pW	104.5 W	113.3 W	0.179 ns	0.152 ns
High-V_t	40.51 pW	39.11 W	47.91 W	0.195 ns	0.172 ns
25% longer gate length (L)	60.08 pW	77.74 W	86.54 W	0.192 ns	0.166 ns
Total power of a 8b MAC (@0.45 V)	14.53 W				

SRAM macro are characterized, with results as listed in Table 9.1. Existing techniques that minimize the leakage current of SRAM memory cells are also considered in this work. Three scenarios are evaluated: a) an SRAM cell that includes transistors with standard-V_t and gate length, b) an SRAM cell that includes high-V_t transistors, and c) an SRAM cell that includes transistors with 25% longer gate length. Note that high-V_t transistors are utilized in both the pull-up and the pull-down networks of both inverters of the 6T SRAM cell to produce the maximum reduction in the leakage current as reported in [57, 58]. Despite the reduction in the leakage current of the SRAM cells that include transistors with high-V_t and long gate length, a significant amount of power is dissipated as leakage as indicated by the results listed in Table 9.1. Specifically, the leakage power of a 6T SRAM is 95.25 pW, 60.08 pW, and 40.51 pW when implemented with transistors of, respectively, standard-V_t and gate length, 25% longer gate length, and higher-V_t. In addition, SRAM cells that are implemented with transistors of higher threshold voltage and gate length result in greater access time

for both read and write operations as indicated by the results listed in Table 9.1, which negatively impacts the overall performance of the accelerator.

Table 1.2: On-chip memory usage in state-of-the-art inference accelerators implemented on an ASIC.

		Google TPU [59]	**MIT Eyeriss V2 [60]**	**DiaNano [61]**	**Mythic [62]**
SRAM/Global buffer size as the percentage of total on-chip area		37%	32%	38%	20%
Die power	Idle	28 W	N/A	N/A	N/A
	Active	40 W	N/A	0.485 W	5 W

A memory cell is actively utilized only when there is a memory request for either a read or write operation, while for the remainder of the time the memory cell remains idle retaining the current bit value. For example, the Eyeriss V2 includes a 246 KB on-chip SRAM and utilizes a 3.9 MB DRAM for implementation of a sparse MobileNet DNN model [63]. However, in each processing element, the utilization of 8-bit fixed point numbers for the activations and weights and 20-bit fixed-point numbers for accumulation requires a maximum length of 20 bits from memory cells per clock cycle [63]. With 192 processing elements, less than 1 KB of on-chip memory is sufficient for both read and write requests per cycle [63]. Therefore, at any given clock cycle, the majority of the off-chip and on-chip memory cells remain idle. The idle memory blocks lead to a significant amount of power consumption as seen from data listed in Table 1.2. The power consumption of the Google TPU is 28 W, which is more than 50% of the active power consumption of 40 W [59]. One of the primary reasons for the higher power consumption during idle mode is the large area occupied

by the on-chip memories (37% of the total die area) [59]. A similar trend in energy loss due to memory is present in more recent AI accelerators including Eyeriss V2 from MIT [63], DiaNano [61], and Mythic [62], where the on-chip memory occupies up to 38% of total on-chip area. Due to the large area occupied by the memory and idle state (*hold state*) of the memory cells for a large percentage of the IC lifetime, there is an opportunity to significantly reduce the in energy consumption by recycling the leakage charge from memory circuits without perturbing the state of stored bits.

1.4 Low Power Techniques at the Circuit, Architecture, and System Level

Low power techniques are applied at the circuit, architecture, and system level to improve the overall energy efficiency of the IC. A summary of different low power techniques is provided through Fig. 1.6. The techniques shown at the intersection between the circuit and architectural levels are applicable to both. Similarly, the techniques that fall within the intersection of both the dynamic and leakage bounding boxes are applied to reduce both the dynamic and leakage power consumption. The circuit level low power techniques include block level voltage scaling, voltage stacking, multi-threshold design, reverse body biasing, and adiabatic logic. In addition, clock gating, power gating, and dynamic voltage and frequency scaling (DVFS)

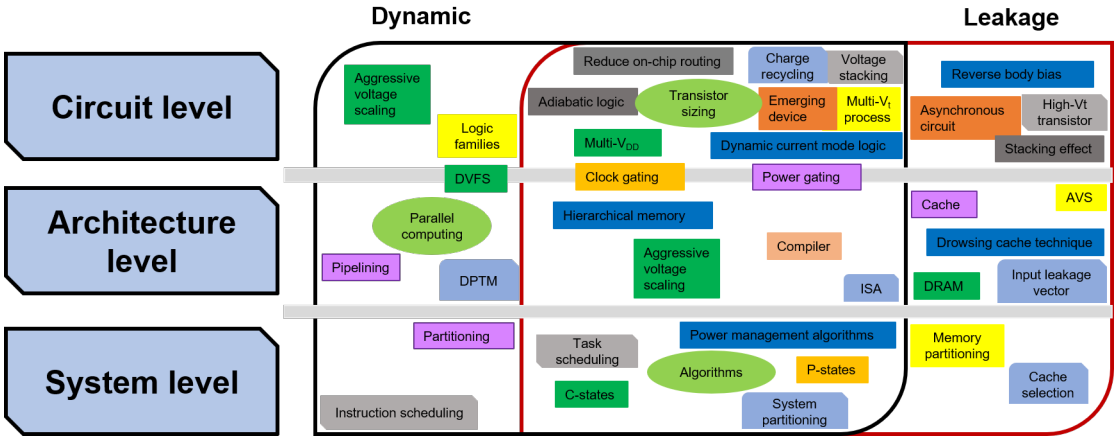


Fig. 1.6: Summary of low power techniques applied to the circuit, architecture, and system level.

are widely used techniques that are applied both at the circuit and architectural level. The system level low power techniques include optimized task scheduling, memory partitioning, and the use of different power states (i.e. P-states and C-states). Despite the application of low power techniques at the circuit, architecture, and system level, a significant amount of energy is lost as leakage in state-of-the-art ICs, where techniques and methodologies to recycle the leakage energy are not well addressed in the research community. In addition, existing leakage reduction techniques such as power gating, clock gating, and voltage stacking result in a degradation of the power supply noise while also causing an additional wake-up penalty [21, 27–29, 64–67].

1.5 Outline of Research Proposal

The primary goal of this research effort is the development of novel circuits, methodologies, and algorithms for energy-efficient and reliable computing for ubiquitous edge devices. An introduction to general power management techniques is provided in Chapter 2, where the core utilization of state-of-the-art ICs is also discussed. An overview of ultra-low voltage circuits is described in Chapter 3, where the benefits and challenges of using near-threshold and sub-threshold computing are described. In addition, the challenges of implementing ultra-low voltage circuits with CMOS logic, the importance of the robustness of sub- and near-threshold circuits to noise, and interfacing sub- and near-threshold circuits with higher voltages are also discussed in Chapter 3. Background on state-of-the-art power management techniques for ultra-low voltage circuits is provided in Chapter 4, where voltage stacking, dynamic voltage scaling, ultra-low voltage regulation, dynamic power management based on the temperature effect inversion (TEI), and error detection and correction are all discussed. In addition, the energy and power considerations of implementing neural networks on FPGAs, CPUs, GPUs, and domain specific custom accelerators including digital SoCs and architectures based on near-memory computing and in-memory computing is provided in Chapter 5.

Circuits and methodologies developed as part of this proposal are discussed in

Chapter 6, where novel ultra-low voltage logic families and an interface between high and low voltage domains is described. The chapter begins with a description of three novel differential signaling based logic families developed for robust and energy-efficient ultra-low voltage computing, which are described as dynamic current mode logic (DCML), latched dynamic current mode logic (LDCML), and dynamic feedback current mode logic (DFCML). The three logic families are developed in a 130 nm technology to operate at 400 mV and 450 mV and are compared with CMOS and current-mode logic.

In addition, novel circuits are described in Chapter 6 that are developed to interface between circuit blocks that operate in an ultra-low voltage domain with circuits that operate at higher voltages. A bi-directional input/output circuit with integrated level shifters is developed for multiple near-threshold voltage domains. The circuit provides a conversion range of 0.38 V to 1.2 V and 0.45 V to 3.3 V depending on the target output voltage.

The co-design of the clock and power delivery networks of ultra-low power edge devices operating in sub-threshold is discussed in Chapter 7. A distributed, multi-voltage domain and hierarchical power distribution network is proposed to deliver current to the clock buffers, registers, and combinational circuits of a local clock distribution network. The variation of the clock skew, setup time, hold time, and clock-to-q delay are analyzed under process and supply voltage variation. The effect

on timing due to supply and process variation is analyzed for a target operating voltage and frequency of, respectively, 250 mV and 2 MHz in a 130 nm CMOS technology.

An optimization technique developed for robust ultra-low voltage circuit operation that sets the supply voltage of a circuit for a given range of threshold voltages is also described in Chapter 7. The algorithm accounts for the variations in the maximum operating frequency f_{max}, the noise margins, and the threshold voltage of the circuit. The algorithmically determined optimal supply and threshold voltages of a CMOS circuit for the 130 nm and 45 nm technology nodes are analyzed through SPICE simulation, where a percent error of up to 14% and 8% is observed for, respectively, the average noise margins NM_{avg} and the maximum operating frequency f_{max} as compared to target circuit specifications.

Novel power management techniques developed as part of this research effort are discussed in Chapter 8. The chapter begins with a description of the leakage reuse technique and the importance of energy efficient computing for multiple-voltage domain systems. The unused leakage current during idle-mode operation of super-threshold (super-V_t) circuits is used to supply current to circuits operating in sub-threshold (sub-V_t). Super-V_t and sub-V_t circuits are simulated in a 45 nm CMOS technology node, where the super-V_t circuits operate at 1.2 V and generate a sub-V_t voltage of 380 mV. The proposed technique is compared with two conventional methods, one that includes separate power distribution networks for the super-V_t and

sub-V_t circuits (baseline) and the second that implements voltage stacking [21, 55, 67, 68].

The overhead and challenges of implementing the leakage reuse technique are also presented in Chapter 8. An analysis of the energy break-even point (BEP) is performed, which is defined as the number of clock cycles at which the total energy savings obtained from the implementation of the leakage reuse technique equals the total energy consumed by the PMOS/NMOS switches and control circuitry required to implement the technique. The analytical model of the BEP is compared with SPICE simulated results.

The application of the leakage reuse technique on an on-chip memory and the implementation of a multi-voltage domain heterogeneous AI accelerator that includes both leakage reuse and near-memory computing are discussed in Chapter 9. The leakage reuse technique described in Chapter 8 is applied to logic circuits only. The completed research includes the development of novel circuit techniques and methodologies that reuse leakage current from on-chip SRAM. The developed heterogeneous AI accelerator with leakage reuse exhibits an energy efficiency of 3.27 tera operations per seconds per watt (TOPS/W), while a conventional monolithic single voltage domain accelerator exhibits an energy efficiency of 0.0458 TOPS/W. The application of the leakage reuse technique to only half of the 0.5 KB of memory within each processing element results in a 26% (99.4 mW) reduction in the power consumption of

the accelerator as compared to an equivalent circuit that does not implement leakage reuse. In addition, techniques are developed to mitigate the effects of process, voltage, and temperature (PVT) variations on accelerators that implement the leakage reuse technique on SRAM memory banks, which ensures a stable supply voltage of 450 mV for the near-threshold power domain.

Chapter 2

Introduction to Power Management Techniques for Improved Energy Efficiency

Power management techniques involve generation and distribution of power to the circuits within an integrated circuit. Developing and executing effective power management techniques and algorithms is essential to improve the overall system energy-efficiency and reliability. Power management techniques are broadly categorized into two types: i) static power management techniques that are applied at design time, and ii) dynamic power management techniques that are applied at run-time. While static power management techniques are applied to optimize the energy efficiency at

design time, the proper implementation of dynamic power management techniques is the most effective means to improve the energy-performance trade-offs of an active IC.

This chapter provides an overview of state-of-the-art dynamic power management techniques for improved energy efficiency. A review of dynamic power management techniques is provided in Section 2.1. The components of the power delivery system and a detailed description of voltage regulators are discussed in Section 2.2. A review of the dynamic voltage and frequency scaling technique, which reduces both dynamic and leakage power, is provided in Section 2.3. The techniques to reduces leakage current through power and clock gating are discussed in, respectively, Section 2.4 and Section 2.5. In addition, an overview of core utilization as relating to the power management of modern processors and system-on-chips is provided in Section 2.6.

2.1 Dynamic Power Management

Dynamic power management (DPM) policies provide control of the generation, distribution, and regulation of power throughout the operation of an integrated circuit. The total power consumption of a CMOS based integrated circuit is given by (2.1), where I_{static} is the total current when the circuits are idle [69–72]. The dynamic power consumption is linearly dependent on the activity factor α, load capacitance C_L, and clock frequency f_{clock}, while quadratically dependent on the supply voltage

V_{dd} [70–72]. The two primary factors that determine the dynamic power consumption of a circuit at run time are the supply voltage and operating frequency. Power management techniques such as dynamic voltage and frequency scaling (DVFS) dynamically apply variable operating voltages and frequencies at run-time based on the executing workloads to achieve a target power-performance point. In addition, both power gating and clock gating are two effective dynamic power management techniques that improve the overall energy efficiency of the system [29, 73–75].

$$P_{total} = P_{dynamic} + P_{static} = \alpha \cdot f_{clock} \cdot V_{dd}^2 \cdot C_L + V_{dd} \cdot I_{static} \tag{2.1}$$

The dynamic power management techniques are applied at different hierarchical levels of the integrated circuit stack including the system level, core level, and circuit block level. A schematic representation of hierarchical power management is shown in Fig. 2.1 [73]. The on-chip power management integrated circuit (PMIC) acts as a global controller and establishes the data transfer between the system level power management algorithms and the core or circuit block power management control knobs. The PMIC initiates global controls such as chip-level power saving methods, as well as managing the local control knobs integrated within the memory hierarchy, interconnect, and cores. The local knobs provide independent control of components that implement the power management policy. For example, the core level local control knobs initiate DVFS by changing the reference voltage of on-chip voltage

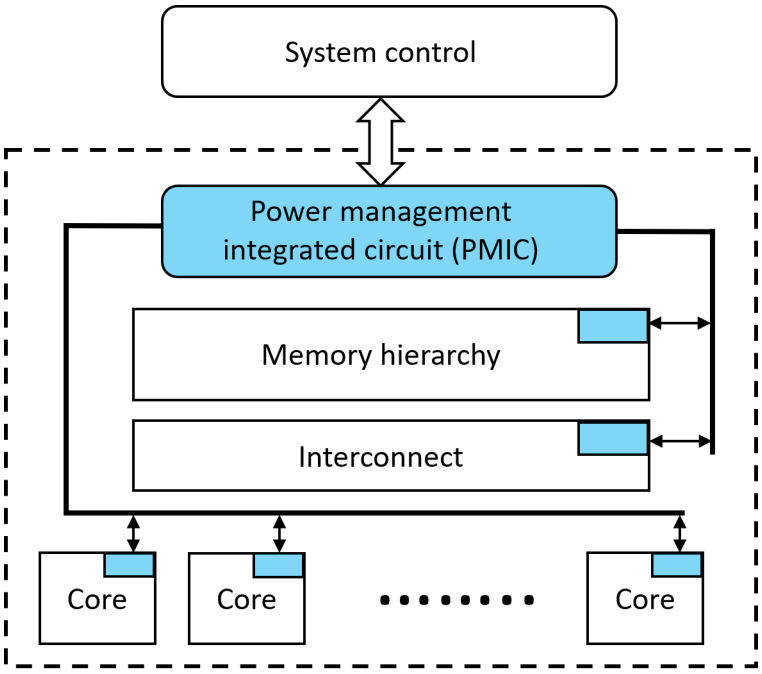


Fig. 2.1: Overview of hierarchical power management in integrated circuits [73].

regulators (OCVRs) or initiate power gating of a core or circuit block. The OCVRs are also an integral component of state-of-the-art power management methodologies as the on-chip voltage regulators reduce the transient noise on the power distribution network by being placed close to points with the greatest load current demand within the circuit [76–83].

2.2 Components of the Power Delivery System

The primary components of a power delivery system (PDS) include the voltage regulators, decoupling capacitors, and the power grid(s) [84, 85]. The components

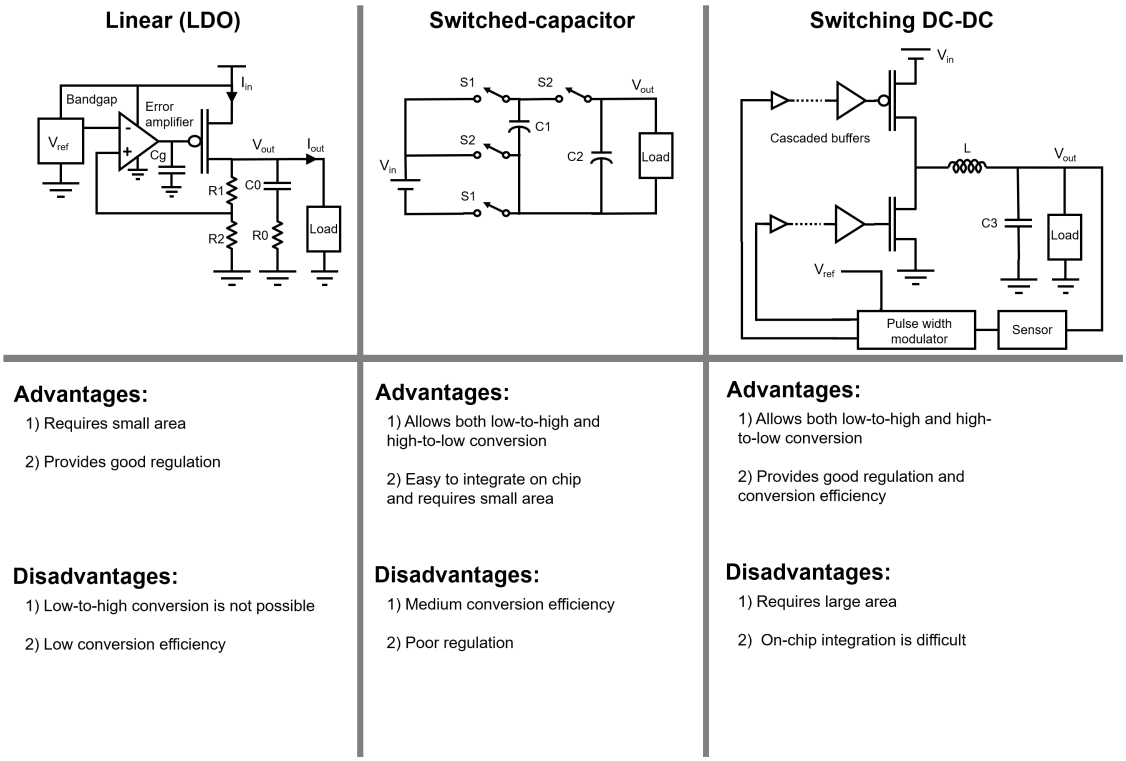


Fig. 2.2: Summary of three types of voltage regulators implemented in modern power delivery systems [84].

of the PDS are designed such that the resistive IR drop and inductive $L\frac{di}{dt}$ voltage ripple (or noise) is minimized [84, 85]. The primary goals of a voltage regulator are to improve the power conversion efficiency, the line regulation, and load regulation, while minimizing the voltage ripple on the PDN [84, 86–90]. Many prior research efforts have proposed different on-chip voltage regulator topologies to up- or down-convert and deliver supply current at multiple voltages. On-chip voltage regulators provide many advantages over off-chip voltage regulators including improved power integrity, greater power savings with per-core voltage control, a faster transient response, and reduced IR losses. A summary of the structure and characteristics of the three primary types of voltage regulators is provided through Fig. 2.2 [84]. Linear

regulators are widely used as each occupies a small area. However, linear regulators suffer from a low power conversion efficiency due to the conduction of a DC current [87, 88, 91, 92]. Low-dropout (LDO) voltage regulators, a type of linear regulator, are popular due to a highly area efficient topology and improved load regulation characteristics [87–90, 92]. However, large output capacitors are required with LDOs to maintain stability with a reasonable response time. Despite the recent improvements in the on-chip integration of capacitors that provide improved stability and performance, an off-chip output capacitor is still required [86–90].

Switching DC-DC converters are implemented due to the relatively high conversion efficiency and improved regulation, while requiring a large area [93, 94]. A switching DC-DC converter, for example, converts a higher voltage to a lower voltage (buck converters), but occupies a relatively large on-chip area due to the required passive capacitor and inductor within the low-pass filter. Therefore, a hybrid switching DC-DC converter and linear voltage regulator topology is proposed to provide distributed on-chip voltage regulation while improving both the area and power conversion efficiency [86]. In addition, many variants of on-chip fully integrated switched capacitor voltage regulators (SCVRs) were proposed in recent years, where a conversion to low voltages including near-threshold has been shown [76–83, 95]. The benefit of implementing a SCVR is the improved efficiency as compared to an LDO without the requirement of an inductor [95, 96].

All on-chip voltage regulators suffer from non-zero conversion loss as well as from area and power overhead. Therefore, in order to improve the overall energy efficiency of any integrated circuit, the energy loss due to the voltage regulators must be reduced or advanced power management techniques must be developed that require a fewer number of active voltage regulators without compromising the stability of the power delivery system.

2.3 Dynamic Voltage Scaling and Dynamic Voltage Frequency Scaling (DVFS)

In the last fifteen years, significant effort has been placed in the research community to develop voltage scaling based power management techniques that reduce the power consumption of integrated circuits [20, 22, 35, 41, 97–100]. Power management techniques are applied across the entire IC from the device and circuit level through the architecture and system level. At the lower levels of the computing stack, techniques aim to reduce the a) capacitive load, b) supply voltage, c) switching frequency, and/or d) circuit activity. Among all techniques mentioned, supply voltage scaling is the most effective technique to date to improve the energy efficiency.

The dynamic power dissipation of CMOS based circuits changes quadratically with

supply voltage as given by (2.1). As the peak performance is not continuously required throughout the operation of a processor, significant energy savings is achieved by dynamically changing the supply voltage and operating frequency based on the executing workload [101]. There are several variants of supply voltage scaling including adaptive voltage scaling (AVS) [102, 103], dynamic voltage scaling (DVS) [22, 100], dynamic voltage and frequency scaling (DVFS) [100, 104], and dynamic voltage and threshold scaling (DVTS) [98], each of which allows for the operation of a circuit from the super-V_t to sub-V_t operating region based on the performance requirements of the executing applications.

Dynamic voltage and frequency scaling is one of the most effective power management techniques developed as an optimum power-performance point is set by simultaneously varying both the supply voltage and operating frequency [101, 105]. The supply voltage of an integrated circuit is chosen to support the peak operating frequency. However, the peak performance is not always required and greatly varies according to the executing workload as shown in Fig. 2.3. Therefore, the supply voltage is dynamically adjusted such that the power consumption is minimized for the given workload. The DVS and DVFS techniques are implemented both at the circuit level and system level [101, 105]. The interaction between the hardware and software layers of a computing device when implementing DVFS based power management is shown in Fig. 2.4. System level DVFS techniques assign the cores or circuit blocks to

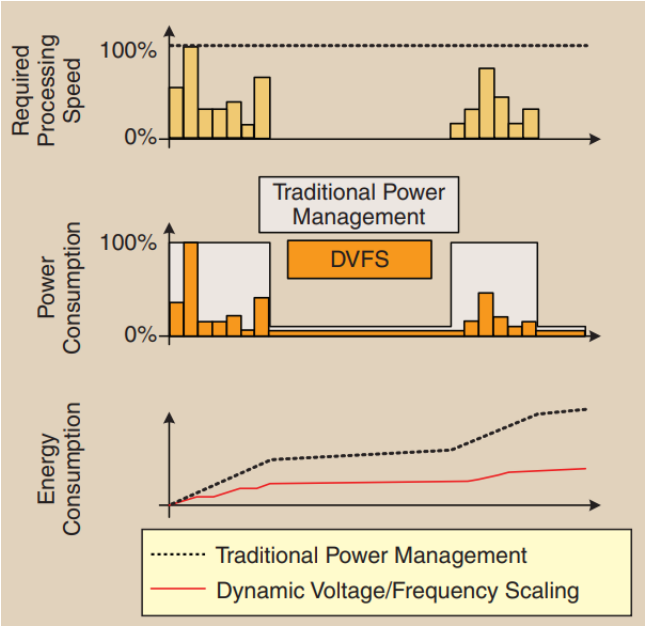


Fig. 2.3: Dynamic voltage and frequency scaling (DVFS) varies both the supply voltage and operating frequency at run-time to achieve the minimum energy cost for a target performance requirement that dynamically changes in time [105].

different performance-states (P-states) based on the executing workload. The on-chip DC-DC converter(s) and on-chip phased-locked loop(s) then dynamically modify the supply voltage and clock frequency, respectively, in response to the set P-state.

Several techniques have been developed to optimize CMOS based circuits for ultra-low supply voltage operation [32, 33, 106, 107]. In the last decade, DVFS techniques were implemented to dynamically vary the supply voltages within the near- and sub-threshold operating regions of a CMOS based system [22, 34, 35, 99, 108, 109]. However, scaling the supply voltage from nominal to near- or sub-threshold voltages does not provide robust operation as the reliability and performance of a CMOS circuit is highly sensitive to power supply noise when operating at ultra-low voltages. Despite

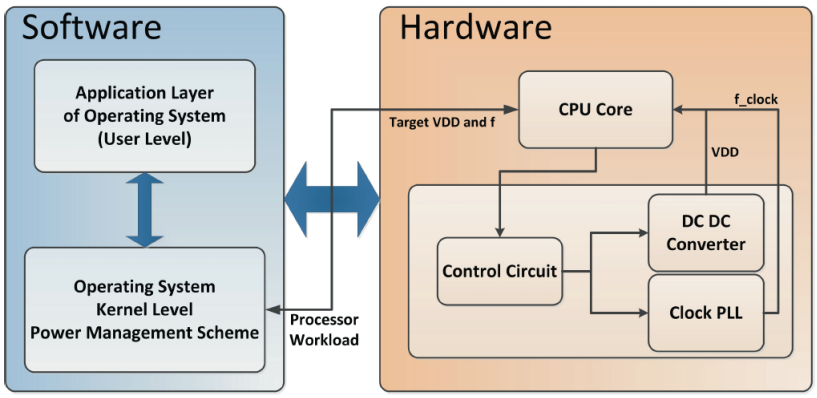


Fig. 2.4: Hardware and software layer implementation of DVFS [101].

advancements in circuit and architectural level DVFS techniques to reduce the voltage to near- and sub-threshold circuit operation, the development of energy efficient and robust logic families for near-/sub-threshold operation has not been explored as a means to address the limitations of CMOS based circuits operating at ultra-low supply voltages.

2.4 Power Gating and Multi-threshold Design

Power gating (PG), a method to disconnect idle circuits from the power and/or ground network, is a widely used power management technique to reduce leakage current [26, 28, 65, 110–116]. The conventional method of power gating is illustrated in Fig. 2.5, where gating of only the ground network is shown.

The sleep transistors shown in Fig. 2.5 are conventionally implemented with high-threshold PMOS or NMOS devices to suppress leakage current in power-gated mode [28, 64–66, 116]. The use of multi-threshold voltage CMOS (MTCMOS) devices in integrated circuits is a widely used technique that allows for high-performance operation with low-threshold transistors and relatively lower performance operation and low standby power consumption with high-threshold transistors, all while operating at the same supply voltage [117]. Power gating implemented with an MTCMOS process enables low-leakage operation by using high-threshold sleep transistors, while the logic or memory circuits are implemented with low-threshold transistors to allow for high performance operation [28, 64–66, 116]. The primary advantages of using sleep transistors in the idle mode are 1) the ground node (virtual ground) of the cores charged to a steady state value close to V_{dd} results in a reduction of the leakage current [28], and 2) the on-resistance of the transistors, assuming the *source* terminal of the NMOS transistors is connected to virtual ground, is significantly greater due to an increased threshold voltage through the reverse body-bias effect [118, 119].

Techniques developed through prior research apply cut-off and intermediate mode power gating at the core, block, memory, and network-on-chip (NoC) level in CPU, GPU, and SoC platforms [26, 28, 65, 113–115, 120]. A block level PG technique has been proposed that analyzes the correlation between RTL modules, and based on the analysis, power gates the module(s) in the case of inactivity [113]. A technique based

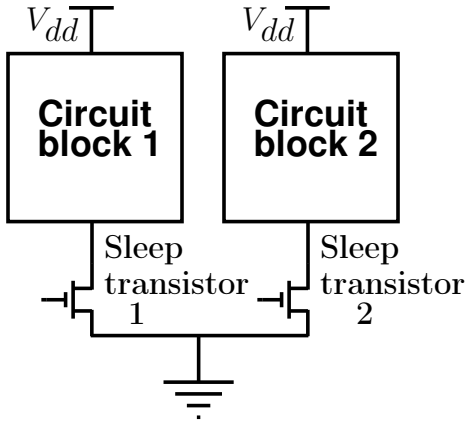


Fig. 2.5: Conventional method of power gating, where the ground network is gated through sleep transistors.

on probabilistic analysis of NoC routers has been proposed, where queuing theory is used to model the break-even point in consumed energy when applying PG to the buffers of the router [114]. In [115], a power gating technique is applied to the caches and GPUs of a multi-core platform to reduce the energy loss due to leakage current.

In addition, adaptive power gating is proposed to gate idle functional units [121]. A system level power gating technique assigns circuit blocks within a core or cores within a multi-core system to different operating modes (C-states) based on idle activity patterns to improve the performance per watt [26, 120]. Power gating with single and multiple sleep modes raises the voltage on the virtual ground node by up to VDD when applying a power-down mode and, therefore, reduces the leakage current of the circuit [28, 65, 112]. Prior work has demonstrated a 47% reduction in the leakage current when adaptive power gating is applied, while maintaining the same performance [64].

However, the reduction in the leakage current is achieved at a cost of increased current transients on the power distribution network (PDN) and a large power consumption by the sleep transistors due to the discharge of charge from the virtual ground node to the true GND node during mode transitions [111, 112]. The primary challenges of power gating include: 1) greater ground bounce due to sleep-to-active mode transitions, 2) longer wake-up (or settle) times, and 3) an increase in the power consumption due to leakage current through the sleep transistors that must be offset by savings from applying PG [28, 64–66].

2.5 Activity Based Clock Gating

In modern digital integrated circuits, more than 30% of the total dynamic power is consumed by the clock delivery network [122]. Clock gating is a widely used power management technique that disables a portion of flip-flops (FFs) in an integrated circuit due to inactivity, which minimizes the power consumed [29, 74, 75, 122, 123]. Clock gating is applied with both fine and coarse granularity at the architecture, core, block, and gate level [122, 123].

Through the simulation of a digital signal processing (DSP) core with 22K flip flops, on average, 97% of flip-flop outputs do not toggle across 240K cycles of the clock signal [122]. The high percentage of static outputs is common in many state-of-the-art digital integrated circuits. Therefore, data-driven fine grained clock gating

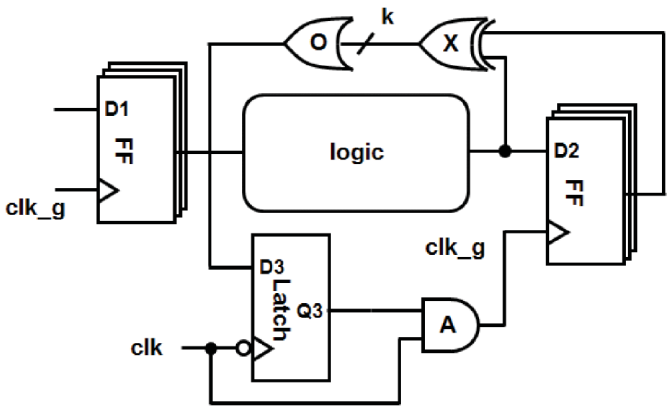


Fig. 2.6: A schematic representation of a circuit implementing data driven gate level clock gating [122].

methodologies have gained traction in the last ten years [74, 75, 122]. A schematic representation of an implementation of a clock gating technique is shown in Fig. 2.6 [75, 122]. Applying gate level clock gating allows for the fine grain control of the clock signal, which improves the overall energy efficiency of the IC. For the schematic shown in Fig. 2.6, k number of flip-flops are selected for clock gating based on gate level data, where the latch, OR, XOR, and AND gates are additional circuits required to implement the clock gating technique. The latch and AND gate are commonly provided as a single component called an integrated clock gating (ICG) cell [122]. The area and power due to the use of an ICG cell is reduced when amortizing the cost by sharing a single ICG across k FFs [122]. In state-of-the-art integrated circuits, clock gating and power gating are simultaneously implemented to further reduce the overall power consumption of the circuit [29].

2.6 Core Utilization in Modern Multi-core Platforms and System On Chips

Most modern low power circuit and power management techniques reduce the amount of energy consumed during the idle mode operation of the circuit [26, 124]. Circuit blocks within a core or cores within a multi-core system are assigned to different operating modes (C-states) based on idle activity patterns to improve the performance per watt [26, 120]. Power gating and clock gating are the most common techniques implemented to reduce the power consumption during idle mode operation of the circuits [110, 119, 123, 125]. The techniques described in Section 2.1 to 2.5 are most effective when the executing workloads exhibit long and/or predictable idle events [26].

In [120], a machine learning based prediction methodology in conjunction with an OS power management policy is implemented on a multi-core processor that assigns a core to a C-state based on the idleness of the given core, where the C-4 state places the core in power gated mode. For the analysis of system idleness, a 3 GHz quad-core processor was analyzed for 10K cycles. Within the 10K cycles, there was not a single instance when all cores were simultaneously active [120]. In addition, the SPECWeb benchmark suite was executed on a dual-core processor, and similarly, there was no instance when both cores were concurrently active [120].

Cores with long idle periods are conventionally power gated to reduce loss due to leakage current. Per core power gating (PCPG) has been proposed as an effective power management option for multi-core processors along with dynamic voltage and frequency scaling [119]. In [119], the core utilization traces of a 2.5 GHz AMD Phenom X4 9850 that includes four cores implemented in a 65 nm technology are analyzed through simulation to properly characterize the application of PCPG. The utilization traces for a commercial application server (PHARMA04), two web servers (HCOM10, ECOM3), and a desktop computer (DESKTOP) are analyzed to determine the activity of the four cores, and again, there is no instance when all four cores are simultaneously active [119]. In this proposal, to characterize the CPU utilization of individual cores across different applications, a system level simulation of a 48 core Intel Xeon CPU that operates at 3 GHz and runs Red Hat Linux is utilized.At any given time, at least one core remains idle, while all 48 cores are idle for more than 90% of the runtime.

In addition, the idleness behavior of state-of-the-art processors is characterized using both consumer and CPU-GPU benchmark applications that include *DirectX9*, *KMeans*, and *Gaussian* from the PCMark and Rodinia suites [26]. The results from the simulation of the benchmark workloads demonstrate that a minimum of 110 multi-cycle idle events per second occurred for a broad range of applications executing on a 16 nm FinFET technology [26]. The cumulative distribution function of idle events

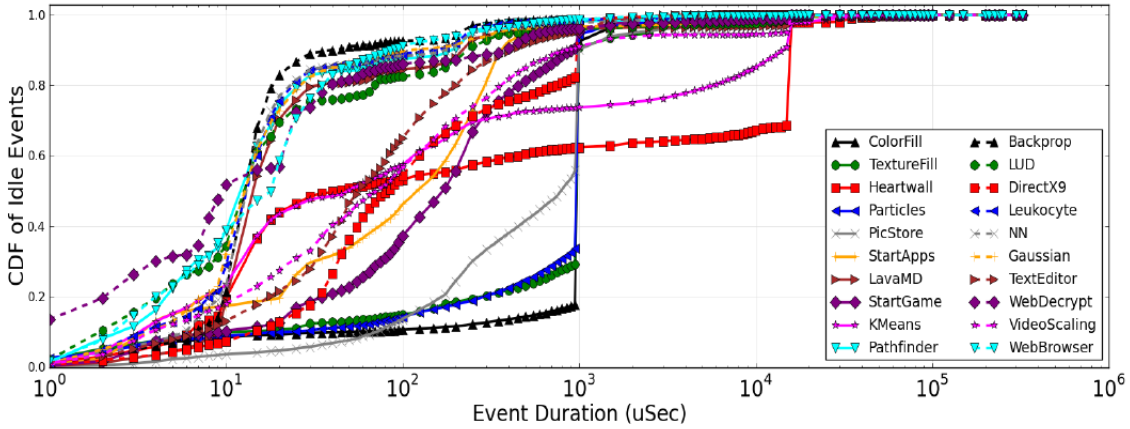


Fig. 2.7: Cumulative distribution function of idle events as a function of idle event duration for various modern CPU-GPU benchmark applications [26].

as a function of idle event duration for various CPU-GPU benchmarks is shown in Fig. 2.7. The results are obtained using an AMD A8-5600K accelerated processing unit (APU) and the Xperf tool suite [26]. All benchmark workloads shown in Fig 2.7 exhibit idle events with a minimum of 100 µs duration [26], while state-of-the-art power gating techniques require a wake-up time in the order of nanoseconds [126]. Therefore, there are enough idle circuit blocks and cores within any system that are long enough to apply various power management techniques that improve the overall energy efficiency of the system [26, 119, 120].

Battery operated devices used for edge applications also consume a significant amount of power in idleness [43, 127]. Mobile and edge devices remain idle for a large portion of the operating lifetime [127]. An idle edge device consumes 70% of the rated maximum power if no power reduction techniques are implemented [43]. In addition, there has been a significant effort to shift the computation from cloud

servers to edge devices to reduce the latency and the additional power consumed to transfer data between the servers and edge devices [43, 44]. However, edge devices are highly constrained by battery lifetime, and, therefore, any computing performed on these devices must maintain a high energy efficiency [43, 44].

The advent of on-device intelligence at the edge introduces very stringent power budget constraints with greater computational demand. Recent research applied fused tiled partitioning (FTP) on the edge devices to minimize both device idle time and the memory footprint by distributing the computation of a single convolutional neural network (CNN) into multiple runs of execution on a given hardware platform [128, 129]. However, the idle power consumption of state-of-the-art edge devices (CPU, GPU, FPGA, and custom accelerator ASIC) implementing deep neural networks remains as high as 50% of the average power consumption [130]. Therefore, the idle activity pattern in mobile and edge devices poses additional challenges due to very limited computing resources and a limited energy budget. Significant improvement in the overall energy efficiency of state-of-the-art CPUs, GPUs, and ASICs for mobile and edge devices is possible by reducing and recycling leakage energy during idle periods of circuit operation.

2.7 Summary

In this chapter, dynamic power management techniques for improved energy efficiency are discussed. Techniques including dynamic voltage and frequency scaling and clock gating are applied primarily to reduce the dynamic power consumption of a circuit, while the power gating technique is applied to reduce the leakage power of an integrated circuit. However, power gating does not disconnect all components of a circuit from the power supply that are in idle state due to the additional overhead in power consumed to implement the gating technique and the amount of time required to transition between power gating modes. In addition, components of the clock delivery network including the buffers, flip flops, and clock gating cells continue to leak current despite the implementation of the clock gating technique. Therefore, a significant amount of circuits remain idle during the operation of an integrated circuit that leads to energy loss due to leakage current. Techniques to recycle the leakage energy from idle circuits, therefore, provide further opportunities to improve the overall energy efficiency of an integrated circuit.

Chapter 3

Techniques to Operate Circuits at Ultra-Low Voltages

There are three primary operating regions of an integrated circuit broadly categorized based on the relative difference between the supply voltage and the threshold voltage. Conventional CMOS based logic circuits do not perform similarly in the three operating regions, with a significant degradation in the performance of the circuit observed when operating at ultra-low supply voltages. Therefore, alternative logic families to CMOS are proposed in prior research that are more suitable to operate at ultra-low voltages [18, 131–140]. An overview of the challenges of operating a circuit in the near- and sub-threshold operating regions is provided in this chapter. In addition, techniques developed to address these challenges are also described. An

introduction to near- and sub-threshold computing is provided in Section 3.1. Prior research on logic families that are optimized for near-threshold computing is discussed in Section 3.2. The importance of interfacing between ultra-low voltage circuits and higher voltage domains is explored in Section 3.3. A brief summary of the chapter is provided in Section 3.4.

3.1 Introduction to Near- and Sub-Threshold Computing

Supply voltage scaling has been applied in the past two decades as one of the most effective techniques to achieve improved energy efficiency. The three operating regions of a circuit are described as: 1) *Super-threshold*, where the supply voltage is considerably higher than the threshold voltage (strong inversion), 2) *near-threshold*, where the supply voltage is equal to or slightly above the threshold voltage of the transistors (moderate inversion), and 3) *sub-threshold*, where the supply voltage is below the threshold voltage of the transistors (weak inversion) [18]. The utilization of sub-threshold and near-threshold computing improves the energy-efficiency of a circuit or system, where the target applications of both differ based on the target operating frequency. In this section, an overview of near- and sub-threshold computing is provided.

3.1.1 Near-threshold Computing

Near-threshold computing (NTC) has gained traction in the research community as both the energy efficiency and the performance per watt of a circuit improve [18, 30, 71, 72, 136, 141–144]. Historically, through device scaling, the improvement in the energy efficiency of a circuit did not come at the expense of the operating frequency of the circuit. The transistor gate lengths have scaled to a few nanometers with advancements in state-of-the-art fabrication [46]. In addition, the transistor density within a given footprint has significantly increased as advanced techniques including the use of FinFET devices and 3-D integration have been implemented [18, 145, 146]. However, beginning with the 65 nm technology node, improvements in the energy efficiency have diminished as the supply voltage of a circuit has not scaled at the same ratio as the physical features of the transistors, as discussed in Section 1.1 [18]. Despite the use of smaller transistors, the dynamic power consumption is not reduced due to the quadratic dependence on the supply voltage (V_{dd}). In addition, a large number of the transistors of an IC are not operated simultaneously due to the increased power density and limitations in the cooling of the circuit [147]. Operating circuits at a near-threshold voltage, therefore, allows for a greater number of transistors to operate simultaneously within a given thermal design power (TDP) budget, resulting in increased circuit utilization and throughput [18, 144, 148, 149]. Near-threshold computing and the associated circuit, architecture, and system level

characteristics are, therefore, well suited for ultra-low power applications, where, due to a stringent power budget, energy efficiency is preferred over a high operating frequency [18, 144, 148, 149]. Operating circuits at a near-threshold voltage results in both reduced dynamic and static power consumption without a significant penalty in performance [71, 72, 136]. Despite the benefits of near-threshold circuits, most research on NTC is based on standard CMOS logic. In addition, the reliability and robustness of near-threshold circuits to noise are rarely considered in the research community [18, 71, 72, 136, 144].

3.1.2 Sub-threshold Computing

The utilization of sub-threshold computing provides high energy efficiency for ultra-low power applications with very low operating frequency requirements [18] [32, 150–152]. Unlike near-threshold computing, there exists an exponential relationship between the supply voltage and the maximum operating frequency of a circuit as the transistors operate in the sub-threshold region, as shown for voltages less than 0.3 V in Fig. 3.1 [18, 150]. Note that a significant degradation in the performance of the circuit results when operating in the deep sub-threshold region ($V_{dd,sub-V_t} \ll V_{Th}$).

For the past two decades, researchers have attempted to exploit the potential benefits of sub-threshold circuits, where 36 mV was determined as the theoretical lower limit of the supply voltage that permitted transistor activity [18, 153]. While

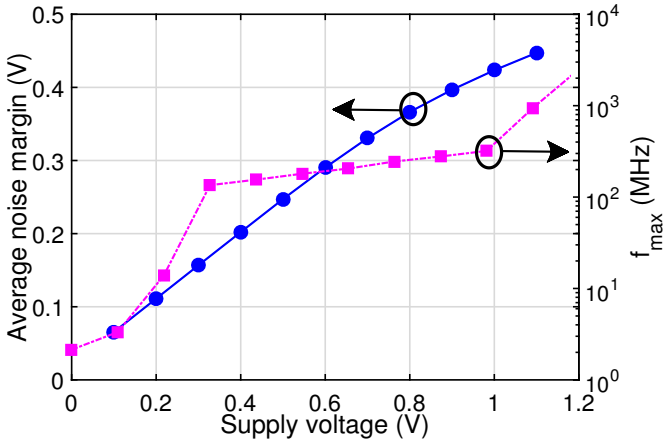


Fig. 3.1: Effect of supply voltage scaling on the maximum operating frequency and average noise margins of a CMOS inverter with a PMOS width of 5.77 µm and a NMOS width of 2 µm in a 130 nm technology.

sub-threshold circuits exhibit reduced energy per operation as compared to circuits operating in super-threshold or near-threshold, there are certain critical constraints that must be addressed to ensure reliable operation. The primary challenges of implementing sub-threshold computing include increased functional failure [18], increased sensitivity to process, voltage, and temperature (PVT) variation [150, 154], and an exponential increase in the power consumption due to leakage current at scaled supply voltages [32, 151, 152].

The circuits operating in near-threshold or super-threshold follow the same current-voltage relationship, whereas the current-voltage relationship of sub-threshold circuits differs. The equation of the current of a transistor operating in sub-threshold is given by (3.1) [155], where V_T is the thermal voltage and is given by $\frac{kT}{q}$, n is the sub-threshold slope, V_{Th} is the threshold voltage of the transistor, and K is a technology

dependent parameter. Unlike the super-threshold and near-threshold operating regions, the drain current of a transistor operating in sub-threshold is exponentially dependent on the gate voltage. Therefore, the drain current is highly sensitive to any changes in the gate voltage, in addition to the sensitivity of the current to noise in the drain voltage. For example, at room temperature, the value of the sub-threshold slope n, which represents a log linear relationship between drain current and gate voltage, is 70 mV/dec [156]. Note that for a circuit operating in sub-threshold, the leakage current I_{sub-V_t} is considered the functional current of the circuit. For bulk CMOS processes, the equation of the total energy consumed by a circuit operating in sub-threshold is given by (3.2), which also applies for a circuit operating in super-threshold or near-threshold [69, 70, 155]. The parameters α_{sub-V_t} and T_{sub-V_t} represent, respectively, the activity factor and the clock period of the circuit.

$$I_{sub-V_t} = K \cdot e^{\frac{V_{GS}-V_{Th}}{nV_T}} \cdot (1 - e^{\frac{-V_{DS}}{V_T}}) \tag{3.1}$$

$$\begin{aligned} E_{total,sub-V_t} &= E_{dynamic,sub-V_t} + E_{leakage,sub-V_t} \\ &= \alpha_{sub-V_t} \cdot V_{dd,sub-V_t}^2 \cdot C_L + V_{dd,sub-V_t} \cdot I_{sub-V_t} \cdot T_{sub-V_t} \end{aligned} \tag{3.2}$$

3.1.3 Energy-delay Tradeoffs of Near- and Sub-threshold Circuits

For both near-threshold and sub-threshold circuits, there is an optimum point in each operating region that provides the best energy-delay trade-off as shown in Fig. 3.2 [18, 157]. Depending on the energy budget and maximum allowed delay, the supply voltage is selected to provide the optimum operating point.

Circuits operating at super-threshold supply voltages consume 10× more energy as compared to near-threshold circuits, while providing improved performance of 5 to 10× [18, 157]. In addition, designing a circuit to operate at a near-threshold supply voltage increases the energy consumption by 2× while reducing the delay by up to 100× as compared to sub-threshold operation, as shown in Figs. 3.2(a) and 3.2(b). Therefore, a moderate improvement in the energy-delay trade-off is possible by operating the circuits at a near-threshold supply voltage. While sub-threshold circuits provide superior energy efficiency, the benefits are limited after a certain voltage as the circuit enters deep sub-threshold operation. Aggressive scaling of the supply voltage to deep sub-threshold significantly increases the delay and results in increased static power, which becomes the dominant component of the total power consumed by the circuit. Although the results shown in Fig. 3.2 are technology dependent (45 nm CMOS), the general tradeoffs between energy consumption and

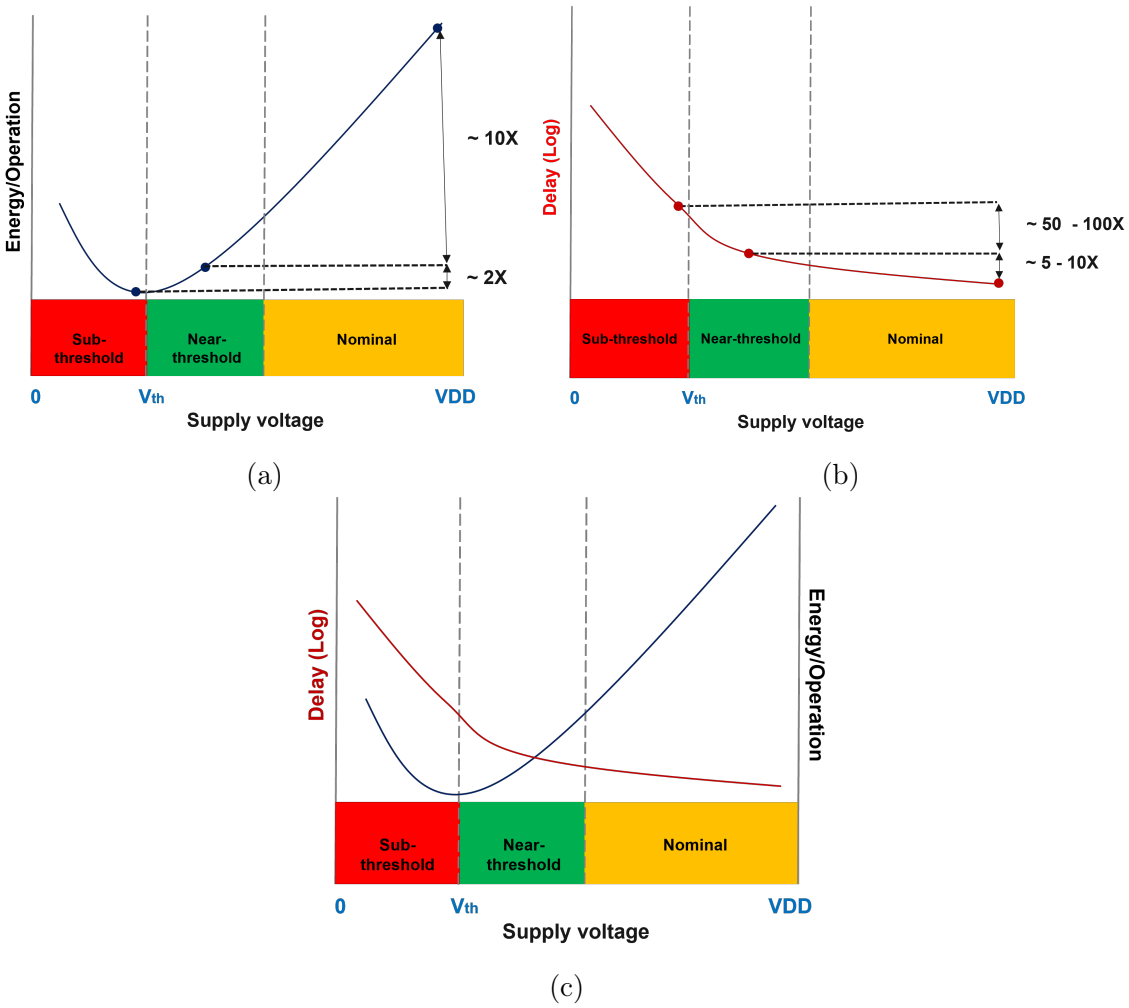


Fig. 3.2: Optimum operating points in near- and sub-threshold operation, where a) the energy per operation is minimal at near- and sub-threshold voltages, b) the delay is optimal at near- and sub-threshold voltages, and c) the combined delay and energy efficiency with respect to supply voltage [18, 157].

delay apply across technology nodes [18, 157].

3.2 Low Voltage Aware Logic Design

Circuits developed to support differential input and output signaling are well suited for high-speed and robust operation as compared to traditional single-ended

CMOS based logic families [132–134]. CMOS based logic suffers from inherent limitations at low supply voltages, which include performance degradation, reduced noise margins (as shown in Fig. 3.1), and lower voltage swing at the output of a gate [18, 30, 99, 136, 142, 144, 158]. The output voltage swing is as equally important as the supply voltage to reduce the dynamic power consumed by a circuit, as given by (3.3). In addition, the circuits designed for super-threshold operation are not well suited for efficient and robust operation at near- and sub-threshold voltages.

$$P_{dynamic} = \alpha \cdot f_{clock} \cdot V_{dd}^2 \cdot C_L = \alpha \cdot f_{clock} \cdot V_{dd} \cdot V_{switching} \cdot C_L \tag{3.3}$$

Prior research explored alternative logic families for ultra-low voltage operation, with several techniques proposed to optimize existing CMOS based logic circuits for operation in near- and sub-threshold [32, 159–163]. Several proposed techniques optimize the sizing of the transistors of a CMOS circuit for energy-efficient operation at ultra-low supply voltages [32, 159, 160]. In addition, dual mode logic (DML), a modified version of standard CMOS based combinational and sequential logic circuits that are operated in two different performance points, is proposed for ultra-low voltage operation [161]. Flip-flops modified with adaptive feedback equalization are also proposed to mitigate the greater sensitivity to process variation when operating in sub-threshold [162]. However, most of the prior techniques are based on conventional CMOS based logic circuits, which suffer from performance degradation, reduced noise

margins, and lower voltage swing at ultra-low supply voltages.

The use of differential signaling provides a larger swing at the output node of a gate as compared to single ended logic such as CMOS and dual mode logic, which is beneficial for both low voltage and high performance applications as the delay and switching power are reduced [132, 135–139, 164, 165]. For near- and sub-threshold operation, alternative logic families were proposed to mitigate the shortcomings of CMOS based logic circuits, which include current-mode logic (CML) and source coupled logic (SCL) [135, 163]. Sense amplifier based pass transistor logic (SAPTL) was also proposed for near-threshold operation [136]. Ultra-low voltage operation using FinFET devices has also been explored in prior research [155, 166]. While FinFET devices exhibit lower leakage current and a lower minimum operating voltage, the challenges of operating circuits implemented with FinFET devices at near- and sub-threshold supply voltages remain similar to those of bulk CMOS based circuits [155, 166]. In this section, an overview of current-mode logic and sense amplifier based logic families is provided.

3.2.1 Current Mode Logic (CML) for Near-threshold Operation

Current-mode logic (CML) circuits are implemented as a differential topology and are conventionally used for high speed applications. Prior research on current mode

logic circuits for near-threshold computing [135, 163] exhibited improved performance as compared to CMOS logic circuits. The primary drawback of current-mode logic, however, is the static current path through the tail transistor, which results in a significant increase in the power consumption. In addition, large transistors are required to maintain the static current through the gate, which increases the capacitive load and, therefore, the delay [135, 163]. A schematic representation of a CML inverter/buffer is shown in Fig. 3.3, which consists of three components: 1) pull-up load resistors, 2) pull-down logic network, and 3) a current source. Typically, PMOS transistors are used in the pull-up network to charge the output nodes of the gate. The NMOS transistors (N1 and N2) included in the pull-down network discharge one of the two output nodes depending on the applied differential input signal(s). Transistors with similar sizes and, therefore, properties, are used in the two branches of the differential circuits to maintain a symmetric charging/discharging path throughout the operation of the circuit. In Fig. 3.3, the NMOS transistor N0 is implemented as a current source, where a set biasing condition is applied to maintain a constant tail current. The current source is the key component of the operation of the CML circuit as the current flow through the two differential branches of the gate are dependent on a constant current set by the source. In addition, the voltage swing at the two differential output nodes is also dependent on the current source. Note that full swing is not required at the input ports of a CML circuit in order to provide a full swing at

the output ports. Therefore, CML circuits operate at a higher speed as compared to CMOS circuits. However, the constant current through the footer transistor increases the overall power consumption of current mode logic circuits as there is a continuous current drawn regardless of the activity pattern of the input signals and the operating frequency of the circuit.

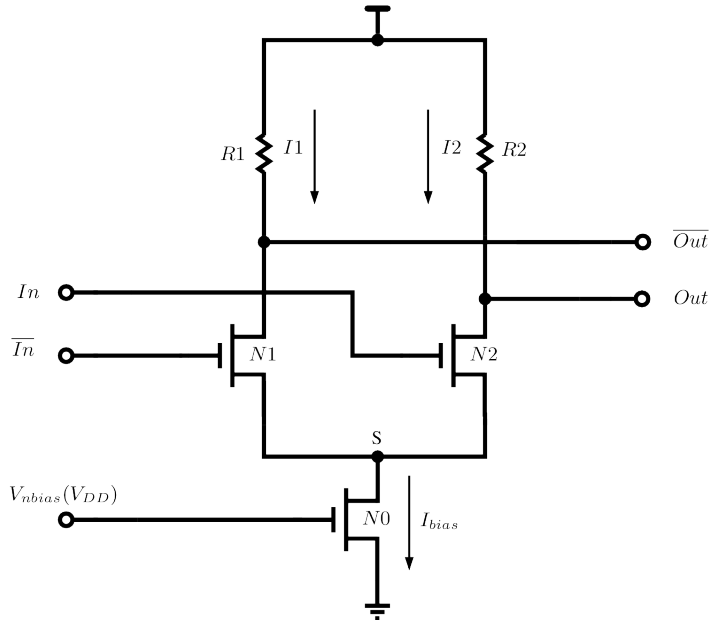
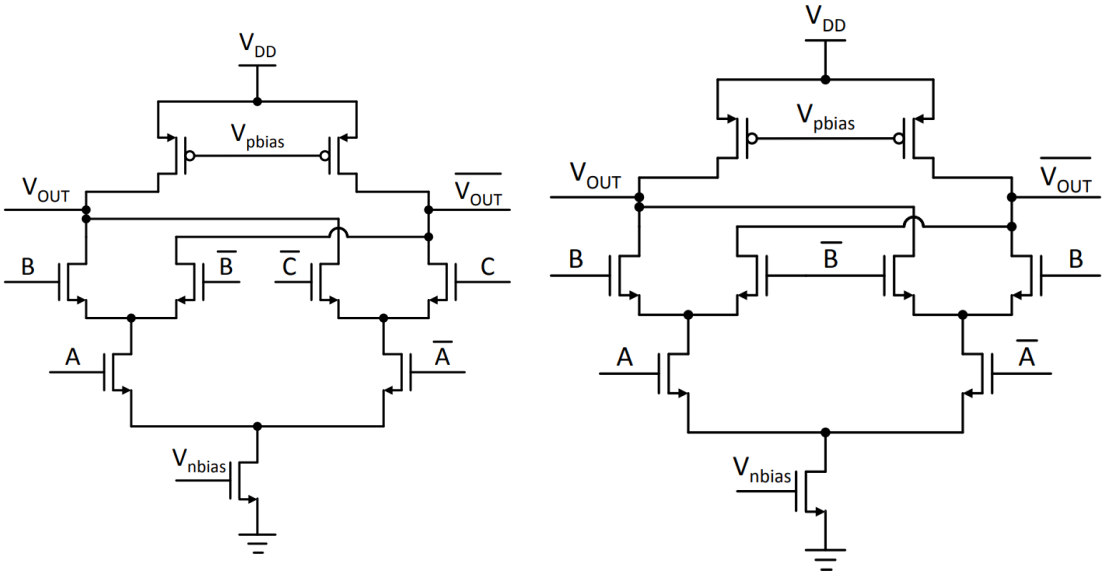


Fig. 3.3: An inverter/buffer implemented using current mode logic (CML) [135, 163].

Despite the increased complexity of CML logic circuits, there are benefits as compared to CMOS logic gates (NAND/NOR). CML circuits operated at near-threshold are evaluated for high frequency applications at low and high activity factors [135]. The simulation results are compared with standard CMOS. Simulation results indicate that CML circuits operating at near-threshold are more power efficient than



(a) (b)

Fig. 3.4: Universal gates based on current-mode logic implementing (a) NAND/AND gate, and (b) NOR/OR gate [135].

CMOS circuits only when operating at higher frequencies and with higher activity factors [135].

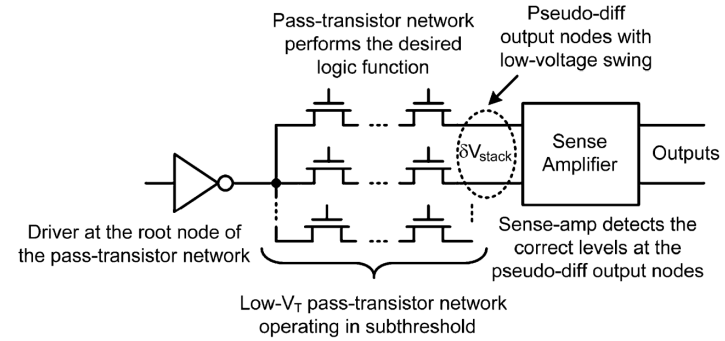
CML universal logic gates implemented for near-threshold computing are shown in Fig. 3.4, where V_{pbias} is connected to ground and the gate voltage V_{nbias} drives the NMOS footer transistor that sets the current through the gate [135]. The two input universal gate shown in Fig. 3.4(a) implements the NAND and AND logic functions depending on the configurations of the inputs and outputs of the gate [135]. Similarly, the NOR and OR logic functions are implemented as shown in Fig. 3.4(b), based again on the configurations of the input and output connections [135].

Simulations were performed on gates implemented in a 14 nm technology with a

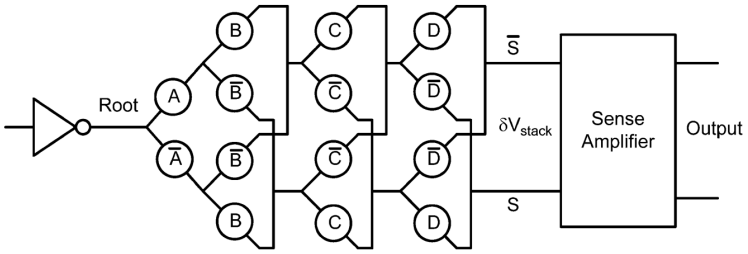
Table 3.1: Characterization of power consumption and delay for CMOS and current-mode logic circuits at a near-threshold voltage of 400 mV [135].

Gate type	Technology	Delay (ps)	Static power (nW)	Dynamic power (nW)	Supply voltage (mV)
NAND gate	CMOS with NTC	120	0.150	2270	400
	MCML with NTC	99	800	800	400
NOR gate	CMOS with NTC	112	0.090	1600	400
	MCML with NTC	89	1200	1200	400
XOR gate	CMOS with NTC	267	1.225	2600	400
	MCML with NTC	147	800	800	400

threshold voltage V_{Th} of 350 mV. The supply voltage was set to 400 mV, which is considered a near-threshold voltage for the 14 nm node. Gate level characterization of the power consumption and delay of NAND, NOR, and XOR gates implemented in both standard CMOS and current-mode logic was performed with results as listed in Table 3.1 [135] m,l. Except for the NOR gate, the other two standard CMOS gates consume more power as compared to the equivalent current-mode logic gate. The delay of all three current-mode logic gates is less than that of the corresponding CMOS gates. In addition, a 32-bit Kogge Stone adder is simulated to evaluate the high frequency performance of current-mode logic. Simulation results of the adder set to a near-threshold voltage indicate that CML circuits are more power efficient than CMOS circuits when operated above 2 GHz with a 100% activity factor or 9 GHz with a 20% activity factor. However, CML circuits consume more power than CMOS for low activity factors (10% and 20%) and when operating at low frequencies. Therefore, due to the static current path, CML circuits provide benefits only at higher frequencies and activity factors.



(a)



(b)

Fig. 3.5: Schematic representation of a) a generic gate implemented with sense amplifier based pass transistor logic (SAPTL) that consists of a driver, network of pass transistors, and a sense amplifier and b) a 4-input XOR gate implemented with SAPTL [136].

3.2.2 Sense Amplifier Based Logic for Near-threshold Operation

Sense amplifier based pass transistor logic (SAPTL) was developed as a means to operate circuits at ultra-low voltages ranging from near-threshold to deep subthreshold [136]. A schematic representation of a generic SAPTL gate is shown in Fig. 3.5(a). The SAPTL gate consists of three components: 1) a driver, 2) a pass transistor network, and 3) a sense amplifier [136]. Note that only transistors within

the pass transistor network operate at near- or sub-threshold, while the driver and sense amplifier operate at a super-threshold supply voltage. Due to the sense amplifier, the SAPTL circuits are capable of discerning the logic value of signals operating at ultra-low voltages. However, there is significant overhead in power consumption and delay due to the driver and the sense amplifier. A schematic representation of a 4-input XOR gate implemented using SAPTL is shown in Fig. 3.5(b), where each circle within the network represents a pass transistor. Similar to current-mode logic, the overhead in the power consumption of SAPTL logic is significant due to the constant current path through the tail transistor of the sense amplifier. The overhead due to the driver and the sense amplifier is only amortized if the number of inputs to a SAPTL gate is greater than five [136], which is not practical in most integrated circuits. Typically, most digital circuits are implemented with two or three input gates. In addition, SAPTL gates are not suitable for circuits with a single power domain as the pass transistor network, the driver, and the sense amplifier require separate voltages.

3.3 Interfacing of Ultra-Low Voltage Circuits With Higher Voltage Domains

State-of-the-art system-on-chips (SoCs) and systems with multi-voltage domains include circuit sub-blocks that operate at disparate voltages that range from super-threshold down to ultra-low voltages (near-threshold to sub-threshold operation) [22, 109]. In addition, circuit blocks that provide differing functionality within a core or SoC require different minimum operating voltages (V_{min}) for energy efficient operation as shown in Fig. 3.6 [22]. The analog blocks, the input/output (I/O) circuitry, and the memory circuits operate at a higher supply voltage as compared to sequential and combinational logic. Therefore, an efficient and robust interface is required between different types of circuit blocks regardless of the operating voltage of each block.

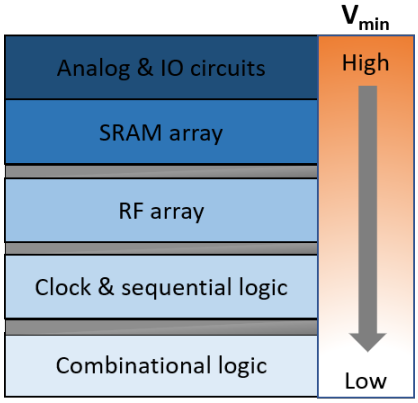


Fig. 3.6: Differing minimum operating voltage (V_{min}) of circuit blocks of an SoC [24].

In addition, power management techniques such as dynamic voltage scaling, adaptive voltage scaling, and dynamic voltage and frequency scaling are applied to dynamically scale the supply voltage from the super-threshold to the near- and sub-threshold operating regions [22, 25, 34, 109]. For example, a GPU core implemented in a 14 nm technology executes DVFS to scale the supply voltage from 1.2 V to 0.3 V [25]. Similarly, a DSP core developed in a 90 nm technology provides dynamic scaling of the supply voltage from 1.2 V to 0.25 V through DVS [22]. Although sequential and logic circuits are dynamically scaled to an ultra-low voltage through DVS and DVFS, the input and output data from these circuits require transmission to memory, I/O, and analog circuits operating at higher voltages. Therefore, robust and energy efficient interfacing techniques including uni-directional and bi-directional level converters and I/O circuits are critical for ultra-low voltage systems that implement dynamic power management techniques.

3.4 Summary

Conventional single-ended CMOS based logic families and logic circuits developed for super-threshold supply voltages are not suitable for robust and energy-efficient operation at near- and sub-threshold voltages. Circuits based on current-mode logic and sense amplifier based logic are proposed for near- and sub- threshold computing. The primary drawback of current-mode and sense amplifier based logic, however, is

the static current path through the tail transistor, which results in a significant increase in the power consumption. In addition, the robustness of both circuit families to noise and energy efficiency at low operating frequencies is not analyzed at scaled supply voltages. Also, robust and energy efficient interfacing methodologies are required for ultra-low voltage operation of circuit blocks within a system-on-chip that includes dynamic power management techniques.

Chapter 4

Power Management Techniques for Ultra-Low Voltage Circuits

Conventional power management techniques are *applicable* to ultra-low voltage circuits operating at near- and sub-threshold supply voltages. However, such techniques are *not sufficient* for the power management of ultra-low voltage circuits as a) fine-grain voltage regulation at near- and sub-threshold voltage levels is challenging to implement, b) a limited number of discrete voltage levels dose not allow for the setting of the optimum power-performance operating point, c) as noted in Fig. 3.2, the delay characteristics of transistors operating at near- and sub-threshold supply voltages differs from that of transistors operating at a super-threshold supply voltage [18, 157], and d) there is a higher chance of timing failure at ultra-low voltages due to

increased delay caused by PVT variation and aging. An overview of techniques that address the challenges of power management for ultralow voltage circuits is provided in this chapter. The dynamic voltage scaling at ultra-low voltages is discussed in Section 4.1. The power management techniques based on voltage stacking and temperature effect inversion are provided in, respectively, Section 4.2 and 4.3. Techniques to dynamically adjust the supply voltage based on error detection and correction are discussed in Section 4.4.

4.1 Dynamic Voltage Scaling at Near- and Sub-threshold Operation

In this section, fine-grain dynamic voltage scaling at near- and sub-threshold supply voltages is discussed. The panoptic dynamic voltage scaling technique [22, 35] that provides control of the voltage source from nominal voltages down to ultra-low voltages is discussed in Section 4.1.1. The voltage dithering technique [22, 102, 167–170] that allows for near-continous fine-grain supply voltage scaling is discussed in Section 4.1.2

4.1.1 Panoptic Dynamic Voltage Scaling

At lower operating voltages, the implementation of fast and efficient on-chip voltage regulators poses both spatial and temporal challenges [22]. In addition, traditional DVS techniques are not applied at a fine granularity as DVS is executed at the core or IC level [35, 98–100, 105, 109, 168]. Recent research has led to significant advancement in the implementation of DVS techniques, where the techniques are now executed at the circuit block level [22]. However, to implement DVS at a fine granularity within a component, circuit block, and/or datapath, an increased number of voltage domains is required, which results in a corresponding increase in the number of expensive voltage regulators that are required (spatial challenge) [22]. The panoptic dynamic voltage scaling (PDVS) technique addresses the temporal and spatial challenges of operating voltage regulators at ultra-low voltages. Note that PDVS is defined as an "all-inclusive" dynamic voltage scaling method that scales the voltage from sub-threshold to super-threshold [22].

A schematic representation of the PDVS technique implemented on a 32 bit data flow processor executing a set of DSP algorithms and fabricated in a 90 nm technology is shown in Fig.4.1 [22, 35]. Each component of the datapath of the processor, including the multipliers and adders, are connected to one of three local VDD levels (high-VDDH, medium-VDDM, and low-VDDL) through dedicated power switches that allow for fast switching between voltage levels. The level converters are utilized

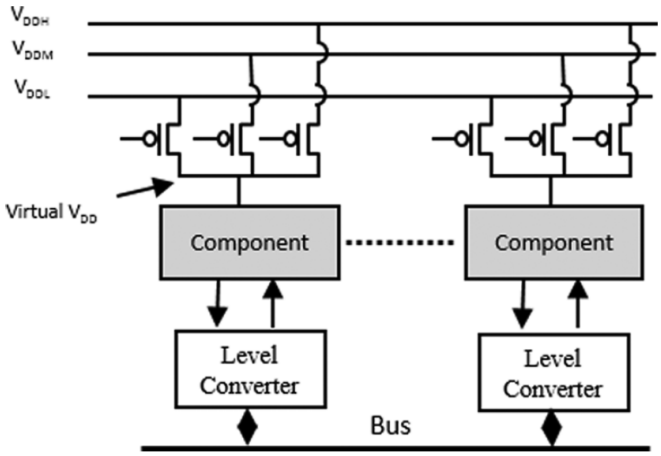


Fig. 4.1: A circuit block implementation of panoptic dynamic voltage scaling (PDVS) that allows for execution of DVS at a fine temporal and spatial granularity [22].

to up convert the output of the block or circuit to a super-threshold voltage.

4.1.2 Voltage Dithering

Dynamic voltage and frequency scaling techniques are dependent on discrete output voltage levels provided by on-chip or off-chip voltage regulators. Specifically, the voltage-frequency operating points stored in a look up table are limited by 1) the maximum and minimum voltage produced by the voltage regulator and 2) the number of discrete voltage levels produced by the voltage regulator [167]. However, the ability to set the voltage to any value between the maximum and minimum permitted voltages produced by the voltage regulators is needed to select the optimal energy-delay operating point of the core or circuit. In addition, circuits operating at ultra-low supply voltages such as near- or sub-threshold require fine grain supply voltage scaling

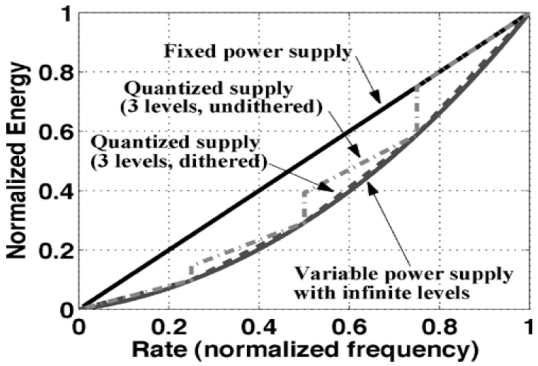


Fig. 4.2: Characterization of the energy consumption as a function of operating frequency for four power management scenarios [168].

[22, 168]. Voltage dithering is a technique that provides a near-continuous fine-grain assignment of the supply voltage within the range of voltages provided by the voltage regulators [22, 102, 167–170]. Through voltage dithering, an executing workload utilizes more than one voltage-frequency operating point [167, 168]. Therefore, through averaging of the discrete voltage-frequency points applied during execution of the entire workload, an intermediate average voltage-frequency operating point is achieved, which allows for more energy savings as compared to the discrete voltage-frequency pairs provided by the voltage regulators [167, 168]. To enable fine-grained voltage scaling in the sub-threshold operating region, a localized implementation of the voltage dithering technique is implemented, where smaller sub-circuits are supplied current through distributed on-chip voltage regulators and PMOS header switches [168].

A characterization of the energy consumption as a function of the operating frequency for four power management scenarios is shown in Fig. 4.2, where the execution

of a generic workload is characterized for different voltage-frequency pairs [167, 168]. The baseline scenario is labeled as "Fixed power supply," where the entire workload is completed with a fixed supply voltage regardless of the computational requirements of the executing workload. Therefore, the normalized energy is linearly related to the voltage-frequency pair. However, operating all workloads on a set supply voltage does result in the optimum energy efficiency. The ideal scenario is denoted as "Variable power supply with infinite levels" and assumes a continuous voltage assignment provided by the voltage regulators. However, the theoretical analysis of the normalized energy with an infinite number of voltage levels is not practical as regulators only provide a set number of discrete voltages. For practical implementations that provide improved energy efficiency, the supply voltage is quantized into three levels, which is denoted as "Quantized supply (3 levels, undithered)" in Fig. 4.2. The technique based on an undithered supply selects the minimum voltage that meets the target frequency of a generic workload and executes the entire workload with the selected voltage-frequency pair. Therefore, the energy consumption is reduced as compared to the baseline scenario that includes a fixed power supply voltage. In addition, within each quantized voltage level of the undithered supply, the energy consumption changes linearly with the set voltage-frequency pair, where the slope of the normalized energy curve is sharper at the higher quantized levels. The voltage dithering technique, which is indicated by the line labeled as "Quantized supply (3

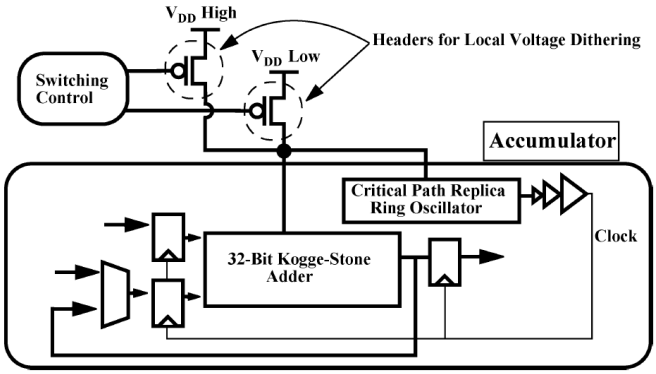


Fig. 4.3: Circuit level implementation of the voltage dithering technique [168].

levels, dithered)" in Fig. 4.2, provides further improvement in the energy efficiency by providing a near-continuous voltage assignment when utilizing three quantized voltage levels. Instead of operating the entire workload on a fixed quntized voltage, the dithering technique executes the workload at two or more voltage-frequency pairs around the average target frequency. Therefore, the overall voltage-frequency operating point of the executing workload reaches an intermediate average value, which provides further reduction in the energy consumption and provides a close to ideal behavior in energy consumption as a function of the set voltage and frequency point.

The application of the voltage dithering technique at the core or IC level requires many microseconds of execution to modify the supply voltage, which does not allow for fine-grained application of the voltage dithering technique. Therefore, a localized voltage dithering technique is proposed to provide fast voltage transitions and to achieve a greater energy efficiency [168]. A circuit level implementation of the local voltage dithering technique is shown in Fig. 4.3, where two quantized voltage levels

are averaged to provide a supply voltage for an adder circuit [168]. The two header PMOS switches provide a rapid transition between the nominal supply voltage and the sub-threshold supply voltage in the order of a few nanoseconds. The same PMOS switches implement power gating when needed [168].

4.2 Voltage Stacking

Voltage stacking is a technique where the power distribution networks of multiple circuit blocks within a core or multiple cores within a many-core system are vertically stacked, sharing a common path from the supply node (V_{DD}) to ground (V_{SS}). In a voltage stacked system, a single high voltage is applied across multiple cores connected in series, in contrast to non-stacked topologies where current is delivered to cores connected in parallel at a lower supply voltage [171]. Therefore, the overall current through the stacked path decreases as the stacked cores provide a larger effective resistance that results in reduced leakage current [171].

With the advancement of on-chip voltage regulators (OCVRs) and dynamic voltage and frequency scaling (DVFS), delivering power to multi-core systems and SoCs with multiple power domains is an effective means to improve the energy efficiency of the circuit [25, 47, 169, 173]. As the number of cores in an integrated circuit increases, the implementation of DVFS at a fine granularity becomes a challenge due to the inefficiency of the step-down on-chip voltage regulators, particularly at lower

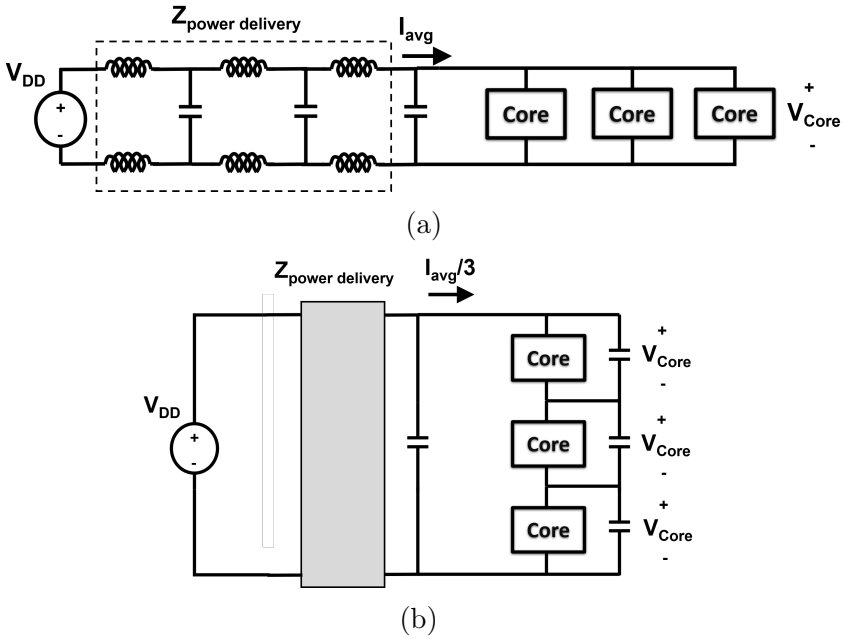


Fig. 4.4: Power delivery using a) a conventional non-stacked method and b) the voltage stacking technique, where three cores are vertically stacked [172].

operating voltages including near-threshold [172]. Voltage stacking has been applied as an alternative means to provide partitioned voltage domains across a stacked topology [172]. However, recent research has shown that the voltage stacked system still requires dedicated voltage regulators at the intermediate nodes due to an imbalance in the current of the stacked power domains [68].

An implementation of voltage stacking is shown in Fig. 4.4 (b), where three cores are vertically stacked as opposed to a conventional power delivery network that delivers current to the three cores in parallel as shown in Fig. 4.4 (a). Therefore, the average current through the vertical stack is reduced by a factor of 3×. In addition, the voltage swing across each core is also reduced by factor of 3× due to the stacking

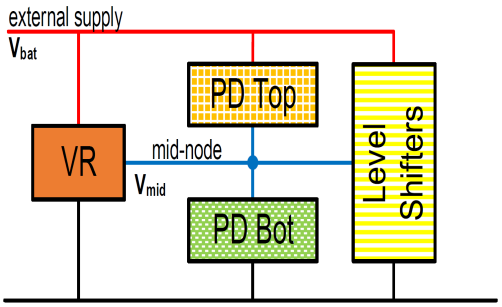


Fig. 4.5: Dedicated on-chip voltage regulators are required at the intermediate nodes of a voltage stacked system to avoid adverse effects due to the imbalance of current between stacked domains [68].

effect. The power supply voltage, however, is increased by a factor of 3× to maintain the full voltage swing across each core. For a system with n number of stacked units, the average current through the stacked path is reduced by n [172]. Each stacked voltage domain is assigned a similar voltage level (V_{CORE}), which is analogous to a resistive voltage divider with equal resistance (or equal current demand) on each stacked core.

The advantage of implementing the voltage stacking technique is primarily due to the reduction of the current in the power distribution network [21, 68, 172, 174, 175]. Voltage stacking has been recently used for logic and memory circuits in both 2-D and 3-D integrated circuits to 1) minimize the total power consumption, 2) minimize the peak rush current [55, 67, 68, 175–177], which reduces inductive noise ($L{\cdot}di/dt$), 3) reduce IR drop across the power delivery network and, therefore, reduce I^2R power loss due to the package, board, and on-chip resistances [172], and 4) limit interconnect wear-out due to electromigration (EM) as the current density is reduced [116]. Prior

work has shown a 60% reduction in transient noise when voltage stacking is applied to a 3-D IC as compared to a non-stacked 3-D IC [67]. In addition, the stacking of SRAM banks during idle mode, as proposed in [55], reduces the leakage power consumed by 93%.

There are limitations of implementing the voltage stacking technique as the advantages are only achieved when a) all cores are homogeneous and executing identical workloads, b) there is no imbalance in the drawn current between stacked cores, and c) no voltage regulators are required between the intermediate nodes. However, none of the three conditions are practical as the stacked cores recycle current regardless of both the executing workload on each core and the current state of activity (active versus idle) of each core [21, 67, 68, 171, 174, 175]. The primary limitations of the voltage stacking technique include 1) the need for regulation of the mid-node voltage (V_{mid} in Fig. 4.5) due to variation in load current [67, 68, 171], 2) variation in voltage due to imbalances in both the workload and the drawn current [67], 3) the need for a boosted supply voltage, which increases the power overhead [67, 68], and 4) a reduction in the energy efficiency of voltage stacked circuit blocks due to an increasing imbalance in the current load between the stacked voltage domains [176]. Level shifters are utilized between power domains to adjust the voltage of signals propagating between stacked cores or circuits [68].

4.3 Power Management Techniques Based on Temperature Effect Inversion

Dynamic power management techniques such as DVFS and dynamic thermal management are widely used to efficiently distribute power to the integrated circuits. The larger number of transistors per unit area in a single circuit results in an increasing power density and a greater rate of heat generation [49, 99, 115, 178]. Therefore, dynamic thermal management plays a pivotal role in advanced technology nodes and is an integral part of dynamic power management for both high performance and low power integrated circuits [48, 179].

In bulk CMOS process technology, changes in temperature directly affect the on-current of the transistors, which impacts the delay of an integrated circuit. An increase in the temperature of a circuit operating at a super-threshold supply voltage reduces both the mobility of charge carriers and the threshold voltage of the transistors [48]. However, from the two, the variation in the current due to the mobility of charge carriers dominates when operating in super-threshold [48]. The result is a reduction in the on-current of the transistors and, therefore, a corresponding increase in the delay of the circuit.

The delay versus temperature characteristics in ultra-low voltage CMOS circuits operating at sub- and near-threshold is fundamentally different from CMOS circuits

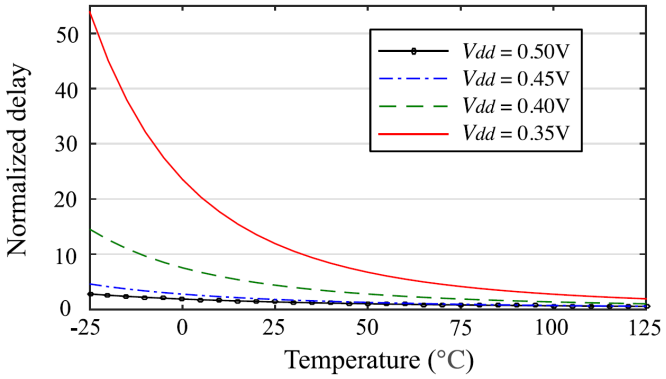


Fig. 4.6: Effect of increasing temperature on the delay of circuits operating at ultra-low voltages in a bulk CMOS process [48].

operating in super-threshold [48]. Unlike the super-threshold circuits, the delay of circuits operating in near- or sub-threshold decreases as the temperature increases [48, 180–182]. This phenomenon is called temperature effect inversion (TEI) [48, 49, 179, 180, 182]. Unlike the circuits that operate at a super-threshold voltage, the variations in the transistor on-current for circuits operating at ultra-low voltages are dominated by the changes in the threshold voltage, which is temperature dependent [48]. Therefore, for circuits operating in sub- and near-threshold, the on-current increases as the temperature increases, which results in a reduction in the delay of the circuit.

The effect of increasing temperature on the delay of circuits operating at ultra-low voltages is shown in Fig. 4.6. The results are obtained through SPICE simulation of standard cells in a TSMC 40 nm process [48]. There is a reduction in the delay of the standard cells as the temperature increases from -25°C to 125°C, where an

approximately 10× reduction in the delay is observed for a supply voltage of 0.35 V. Conventionally, the worst-case timing corners occur at high temperatures for circuits operating at super-threshold supply voltages [48, 180, 182]. However, the worst-case timing corner for circuits operating at sub- and near-threshold voltages occurs at the lower temperatures due to the TEI effect [48, 180, 182]. Power management techniques based on TEI have recently been developed [47, 48]. For example, TEI based power management techniques are applied to a network-on-chip (NoC) that is implemented in a bulk CMOS process [48]. The power savings is obtained by a) reducing the supply voltage once the temperature of the circuit increases above a certain threshold, which results in a reduction in the power consumption while maintaining the operating frequency of the circuit, and b) increasing the operating frequency of a given router to allow for increased traffic as the delay is reduced due to the elevated temperatures, which permits the turning off of one or more surrounding routers to further reduce the overall power consumption [48]. Therefore, TEI-aware power management reduces the power consumption of a circuit without impacting the operating speed [48].

The TEI effect is also present in FinFET based circuits [47, 48, 182]. However, unlike bulk CMOS technology, the TEI effect in FinFET based circuits is observed for super-threshold voltages in addition to near- and sub-threshold operating voltages [47, 48, 182]. Due to the 3-D channel structure of FinFET transistors, the bandgap is

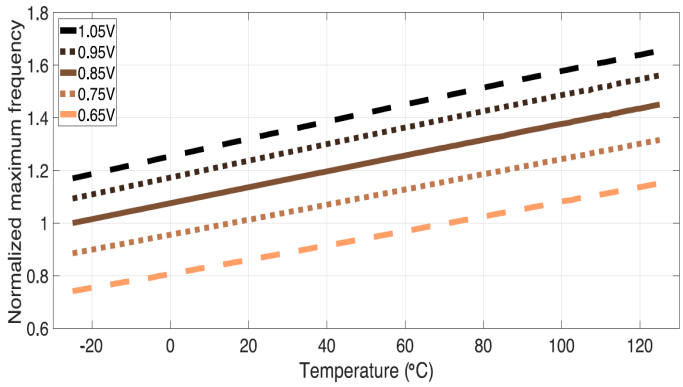


Fig. 4.7: Effects of increasing temperature on the maximum frequency of a fan-out-of-four (FO4) ring oscillator designed in a 16 nm FinFET technology [47].

reduced by the tensile effect of the thin insulator layer under the fins, which results in a reduction in the threshold voltage [47]. In addition, carrier mobility increases as the temperature of the FinFET transistors increases, which corresponds to a reduction in the delay of the circuit [47]. The normalized maximum operating frequency of a fan-out-of-four (FOR) ring oscillator implemented in a 16 nm FinFET technology is characterized with results as shown in Fig. 4.7 [47]. The TEI effect is observed for all supply voltages between 0.65 V and 1.05 V as the temperature is increased from -25°C to 125°C. Therefore, similar to bulk CMOS operating in sub- or near-threshold, the worst-case timing corner is observed at the lowest temperature of -25°C [47].

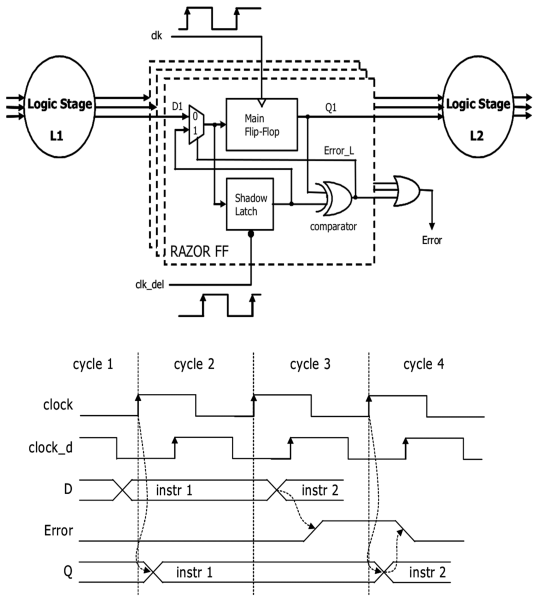


Fig. 4.8: Razor flip-flop utilized to implement an error detection and correction methodology. The associative timing diagram is also provided [183].

4.4 Error Detection and Correction (EDAC) Techniques

Power management techniques based on error detection and correction (EDAC) are effective by providing timing failure, while also providing a reduction in the power consumption of ultra-low voltage circuits [183–185]. As the supply voltage is scaled, the adverse effects of delay variation increase [183–185]. Bias temperature instability, hot carrier injection, static and transient noise on the power supply network, and variations due to process and temperature all adversely affect circuit parameters, which eventually results in variation in the delay of a circuit [186, 187]. Error detection

and correction techniques include circuits to monitor the timing of critical paths, and correct erroneous data by either pipeline flushing or architecture replay [183–185].

Various methods of implementing EDAC have been proposed by the research community [183–185]. The first proposed scheme includes the implementation of the Razor flip-flop, where the supply voltage is dynamically scaled based on the measured error rates in timing [183, 188]. The circuit and timing diagram of a Razor flip-flop is shown in Fig. 4.8, where a shadow latch is implemented within a conventional flip-flop circuit [183]. The shadow latch is enabled by a delayed clock signal. The comparator circuit generates an error signal if there is a discrepancy between the signal captured by the regular clock and the delayed clock. Once a timing error is detected, the Razor-based EDAC technique requires n number of cycles to correct the error for a design with n pipeline stages. The block level representation of the circuit that flushes the pipeline and re-executes a given instruction is shown in Fig. 4.9(a).

The implementation of the technique allows for the reduction of the voltage guard band, which results in a significant savings in the total power consumed. The closed-loop voltage regulation of the circuit based on the EDAC technique is shown in Fig. 4.9(b). More recently, variants of the Razor flip-flop have been proposed to reduce the area and power overhead of the error detection and correction circuitry [185, 189–191]. For example, a current based error detection and correction technique is proposed in [185] that requires three additional transistors in each flop flop instead

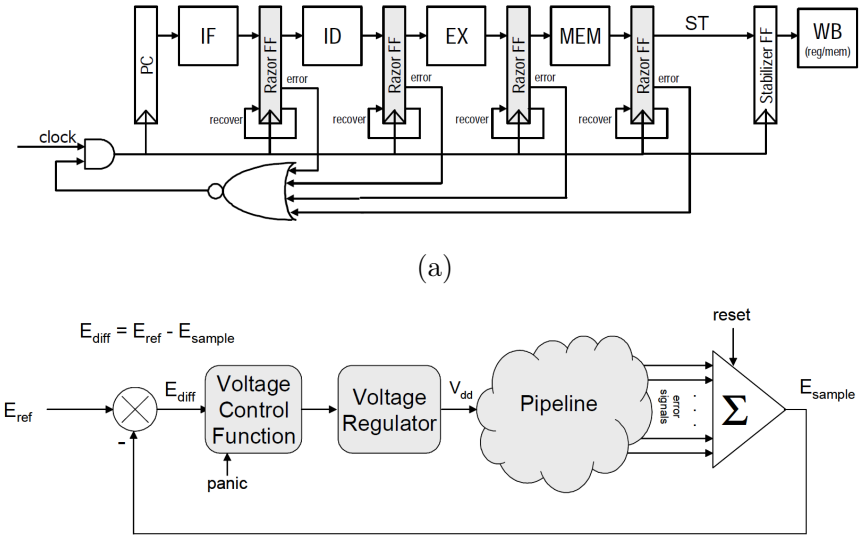


Fig. 4.9: Error correction through architectural and power management techniques: (a) pipeline flushing and re-execution of instructions and (b) a closed-loop voltage regulation technique based on the EDAC technique [183].

of the more complex Razor flip-flop, which reduces the overhead in area and power [185]. However, both techniques require either flushing the pipeline or architectural re-execution [183, 185]. Most prior research on EDAC techniques implement circuits to monitor the delay or timing slack within the flip-flop at the end of each critical path. Recently, a technique called "SlackProbe" has been proposed that places circuits to monitor the timing of data paths at an intermediate point in the datapath between two flip-flops [184]. In addition, stretching the clock period within a given cycle is proposed to allow for the detection and correction of errors [186]. Similar to the technique described in [184], the technique places delay monitors in the middle of critical paths. Due to the increased variation in the delay of a circuit operating at

ultra-low voltages, the placement of monitors in the middle of critical paths is not effective. For example, the delay of clock buffers becomes comparable to the delay of interconnect in the sub-threshold operating region [192]. In addition, the significant increase in the variation of the setup time, hold time, clock-to-q delay, and clock skew at ultra-low voltages increases the chances of timing failure at the leaf nodes even if a given signal arrives on time at an intermediate point of the datapath [192]. Furthermore, for techniques that stretch the clock period to detect and correct errors, there is a corresponding degradation in the operating clock frequency of the circuit. The EDAC techniques are also implemented on DNN hardware accelerators, where the supply voltage of each DNN layer is scaled based on the determined timing error rates of the given layer [193]. Despite the reduction in the power consumption by dynamically scaling the supply voltage based on the measured error rates in timing, the established EDAC techniques require a) clock stretching or placing delay monitors at the middle of critical paths, and b) architectural support such as flushing the pipeline or re-execution of the instructions.

4.5 Summary

An overview of power management techniques optimized for circuits operating at near- and sub-threshold voltages is provided in this chapter. The panoptic dynamic voltage scaling technique provides fine-grain control for circuits operating in the near-

and sub-threshold regions. Voltage dithering provides a near-continuous voltage in a system with discrete voltage levels. The power management techniques based on the TEI phenomenon and EDAC are also described, which are developed to address the variation in circuit parameters due to process, voltage, and temperature (PVT) and transistor aging. In addition, power delivery through voltage stacking is discussed. Despite the reduction in peak rush current, the voltage stacking technique is only beneficial under certain ideal conditions and requires separate voltage regulators for intermediate nodes of the stacked circuits.

Chapter 5

Near-/In-Memory Computing and Custom ASICs for Deep Neural Networks

The hardware implementation of deep neural networks (DNNs) for training and inference has led to the development of artificial intelligence (AI) accelerators tailored for ASICs and FPGAs [194, 195]. Unlike accelerators for training, accelerators for inference to work effectively with low-precision, which allows for the integration of inference accelerators within IoT edge devices and wearables for tasks that include real-time object detection, keyword spotting, computer vision, augmented reality, face recognition, image processing, speech applications, and robot motion planning [53,

196, 197]. Among several types of hardware implementations, deep neural networks implemented on systolic array based custom accelerators have become an effective alternative to CPU and GPU based implementations due to an inherent capability of performing convolution operations, parallel computation, and allowing for data reuse. However, AI accelerators require a large amount of data for both training and inference. As a result, a significant portion of the total power consumption of any AI accelerator is due to the storage and movement of data. More recently, in-memory and near-memory computing were proposed to address the overhead due to the movement of large quantities of data [198–210]. In this chapter, an overview of both Von Neumann and non-Von Neumann based AI accelerator architecture is provided. An introduction to deep neural networks is provided in Section 5.1. An overview of prior research that implement DNN accelerators on CPUs and GPUs is provided in Section 5.2. The opportunities and challenges of using custom ASICs for AI acceleration is described in Section 5.3. The requirements of large on- and off-chip memory in DNN accelerators are discussed in Section 5.4. The dataflows of DNN accelerators are discussed in Section 5.5. An introduction to in- and near-memory computing is provided in Section 5.6. Some concluding remarks are provided in Section 5.7.

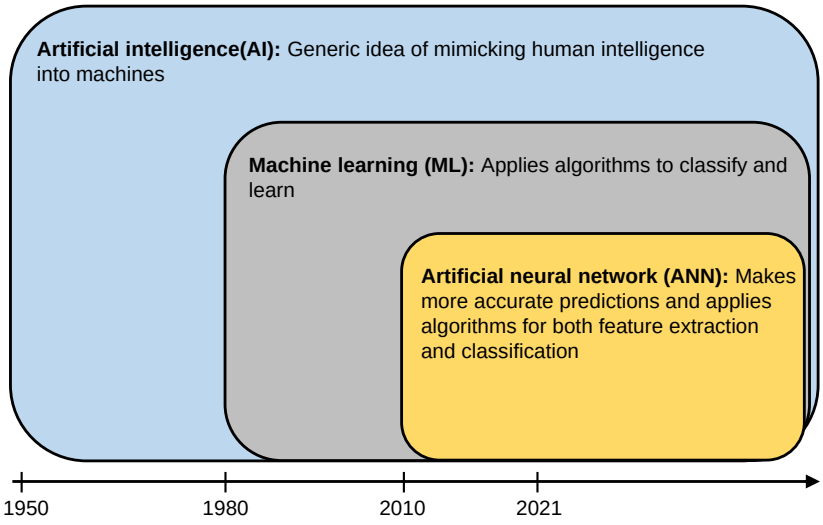


Fig. 5.1: Historical advancement of artificial intelligence [211–213].

5.1 Introduction to Deep Neural Networks (DNNs)

There has been significant growth in the last two decades in the development of techniques, models, and algorithms for artificial intelligence (AI) that attempts to mimic human intelligence [53, 61, 63, 193, 198–210, 214–225]. Machine learning (ML), a subset of AI, applies mathematical models to perform accurate prediction, which involves training an existing algorithm on a large data set to predict an outcome while also actively improving the prediction through learning [212, 213, 226]. Artificial neural networks (ANNs) are an implementation of specialized algorithms that are developed based on the biological neural network and are considered as a subset of ML [212, 213, 226]. Unlike machine learning, ANNs are capable of self-learning features during the learning process. However, ANNs require a greater amount of data and computing power than traditional ML algorithms [212, 213, 226]. Learning with

ANNs is often referred to as deep learning (DL) as any neural network consisting of more than three hidden layers is considered a deep neural network [212, 213]. A progression from AI to ML and final ANNs is described through Fig. 5.1. In addition, an overview of the training and inference of an ANN is provided through Fig. 5.2 [211–213, 226]. The training of an ANN consists of both forward pass and backward pass process. At the end of each forward pass, the resultant prediction is compared with the labels and consequently an error value is generated. Based on the error value, the weight parameters of the network are updated by executing a backward pass through the neural network, a process referred to as 'backpropagation' [211, 227]. The inference process consists of only a forward pass as no update to the weights is performed. Note that the term 'ANN' refers to a generic representation of a convolutional neural network (CNN), a deep neural network, and a recurrent neural network (RNN) [212, 213, 227]. Deep neural networks (DNNs) have become more prominent in recent years due to the availability of greater computing power and the better predictive accuracy provided. More discussion on the operation of neural networks is provided in Section 9.1.2 of Chapter 9.

The general representation of a deep neural network with only one hidden layer is shown in Fig. 5.1. The DNNs consist of inputs, outputs, neurons, weighted connections, and bias parameters. The neurons are represented by nodes as shown in

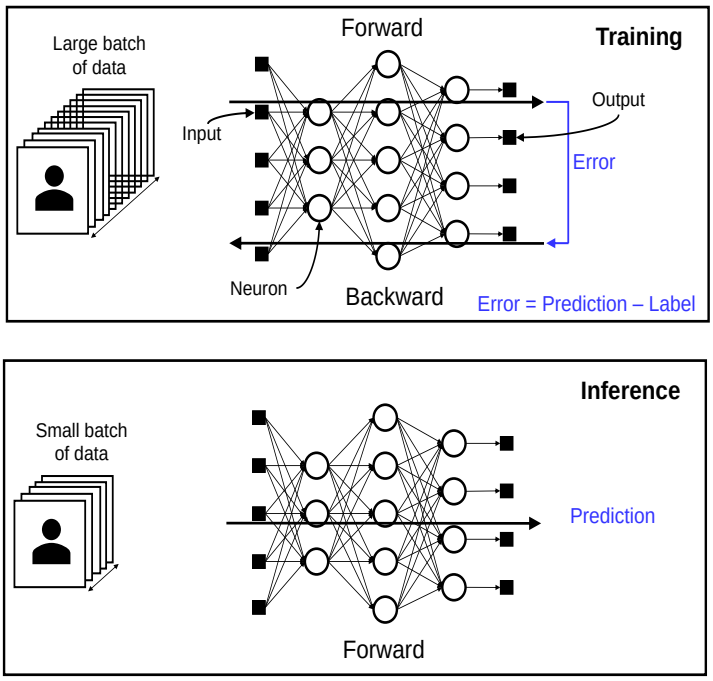


Fig. 5.2: An overview of training and inference in artificial neural networks and the growth of artificial intelligence over the years [211–213].

Fig. 5.1. An abstract representation of a neuron is shown in Fig. 5.3 and the corresponding mathematical computation model of neuron is given by (5.1). A neuron takes n number of inputs $I_0, I_1, ..., I_n$ and provides a single output OUT. The inputs are multiplied by corresponding weights $W_0, W_1, ..., W_n$ and the resulting sum of all $I \times W$ terms is modified by a bias factor β. The bias factor and the summed product terms are applied to a non-linear activation function α that generates the final output OUT. The most commonly used activation functions include the rectified linear unit (ReLU) [228] and the Sigmoid [229].

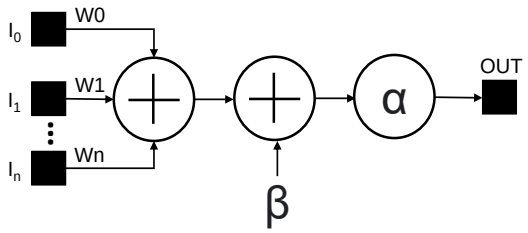


Fig. 5.3: An abstract representation of a neuron [211].

$$OUT(I) = \alpha\left(\sum_{n=0}^{N-1} I[n] \cdot W[n] + \beta\right) \tag{5.1}$$

5.2 Hardware Implementation of AI Workloads

Computing systems based on conventional Von Neumann architectures consist of separate processing and memory units, where a continuous transfer of data is required between both units [63, 211, 218, 230]. The overall latency and energy consumption of the system is highly correlated to the data traffic [218]. With the recent growth in big data and AI applications, the overhead in the overall energy budget and latency of any system-on-chip due to the movement of data has become significant [63, 218, 230]. Prior research that analyzed the execution of Google workloads on consumer devices indicates that more than 60% of the total energy consumed by a system is due to the movement of data [231].

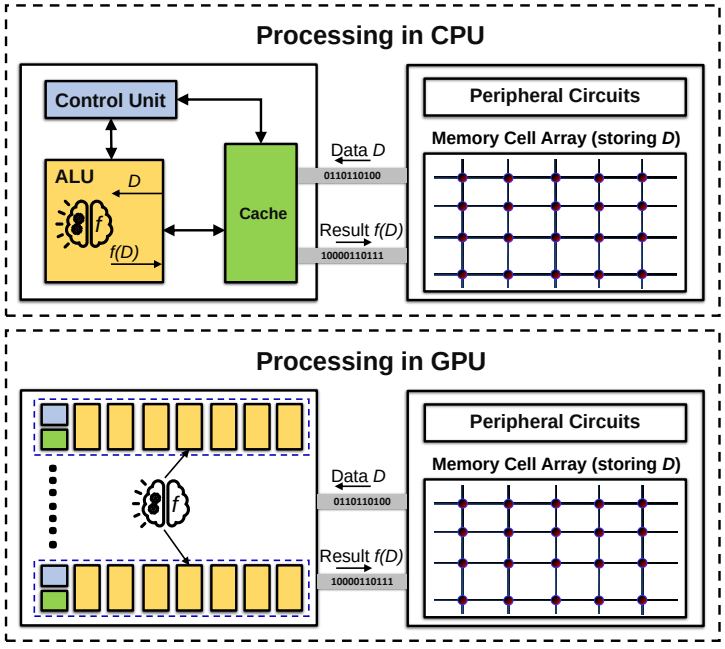


Fig. 5.4: An overview of processing in central processing unit (CPU) and graphic processing unit (GPU) [230]

An overview of conventional Von Neumann based CPU and GPU architectures is provided through Fig. 5.4. The CPU based architectures are suitable for executing generic applications, while the computing power of the CPU is limited due to a smaller number of cores or computing units [53, 61, 63, 193, 211, 214–225, 230]. Conversely, GPU based architectures consists of a large number of cores. Therefore, GPUs are comparatively better suited for executing a large number of specific and repetitive types of workloads including matrix-matrix or matrix-vector multiplications that are common in DNN applications [232–234]. The maximum number of cores in an AMD Ryzen Threadripper CPU and an Nvidia A100 GPU is, respectively, 64 and 6912 [211, 232]. Although GPU based systems provide greater computing power as

compared to CPUs due to the ability of parallel execution on multiple processing units, the energy consumption remains similar in both CPUs and GPUs due to the movement of data. Therefore, both CPU and GPU architectures are not well suited for the execution of DNN workloads due to the limited energy efficiency provided [53, 61, 63, 193, 211, 214–225, 230].

Early DNN models included only five total layers when LeNet was first introduced in 1998 [235]. However, DNNs developed in recent years consist of hundreds of hidden layers. For example, the implementation of the ResNet and SeNet154 models consist of 152 and 154 hidden layers, respectively [236, 237]. In addition, the total number of parameters in the state-of-the-art neural networks has increased to hundreds of millions as compared to early neural networks implemented with less than a million parameters [235–237]. Despite the ability of executing data intensive DNN workloads on CPUs and GPUs, an improved energy efficiency is achieved when DNN workloads are executed on custom ASICs. For example, an AMD Ryzen CPU and a Nvidia A100 GPU perform, respectively, 6.64 giga operations per second per watt (GOPS/Watt) and 195 GOPS/Watt when both are implemented on 7 nm technology [211, 232]. The energy efficiency of custom DNN accelerators implemented as an ASIC has improved significantly as compared to both CPUs and GPUs. For example, the FlexFlow DNN accelerator implemented in a 65 nm technology, the SIGMA implemented in a 28 nm technology, the NullHop implemented in a 28 nm technology, and

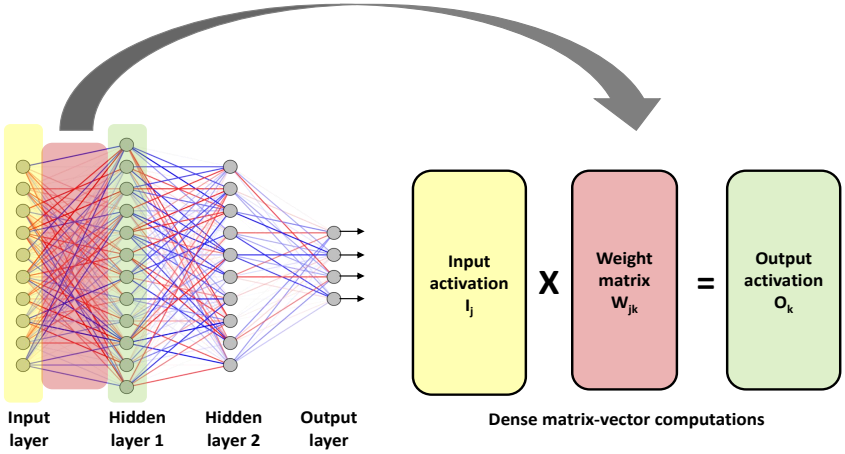


Fig. 5.5: DNN model for inference with two hidden layers. The computation in each layer is decomposed into matrix-vector operations.

the EIE implemented in a 45 nm technology provide, respectively, 420 GOPS/Watt, 484 GOPS/Watt, 2903 GOPS/Watt,and 5000 GOPS/Watt [223–225, 238]. Therefore, research and development in custom ASICs that execute DNN workloads has gained traction in the last five years [53, 61, 63, 193, 198–208, 214–217, 219–225, 239].

5.3 Custom ASICs for AI Acceleration

Early development of hardware based AI accelerators began in the 1990s. The first neural processor developed by a commercial company was by Intel in 1989, where an electronically trainable spiking neural networks IC was implemented [240]. Around the same time, Bell Labs developed a convolutional neural network IC and demonstrated a successful recognition of handwritten digits [241, 242]. As more efficient

software solutions and neural network models emerged, hardware based solutions were not actively pursued. Beginning in the early 2010s, interest has grown in both academia and industry to efficiently implement various neural network models in hardware for improved power-performance-accuracy-throughput trade-offs. The recent advancement of hardware implementations of spiking neural networks on application specific integrated circuits began with the development of *TrueNorth* by IBM in the 2010s, where a brain inspired neuromorphic IC was developed under the DARPA SyNAPSE program [243]. During the same decade, European researchers developed a spiking neural network based architecture labelled as *SpiNNaker* through the Human Brain Project [244]. More recently, Intel developed a spiking neural network based neuromorphic IC referred to as *Loihi* [245].

The hardware implementation of DNNs on ASICs started to get attention when Google launched the first tensor processing unit (TPU) in 2017 [59]. The primary innovation of the TPU is the matrix multiply unit (MXU), which includes a 256×256 2-D array of 8-bit MAC units [59]. However, the Google TPU is tailored for server class computation at datacenters and is not suited for deep learning models executed for mobile and embedded applications including MobileNet and SquezeNet [63]. A more recent implementation of an AI accelerator developed by MIT researchers addressed the limitations of the TPU, where an AI accelerator tailored for mobile and embedded applications was developed [63]. The MIT accelerator introduced the SIMD

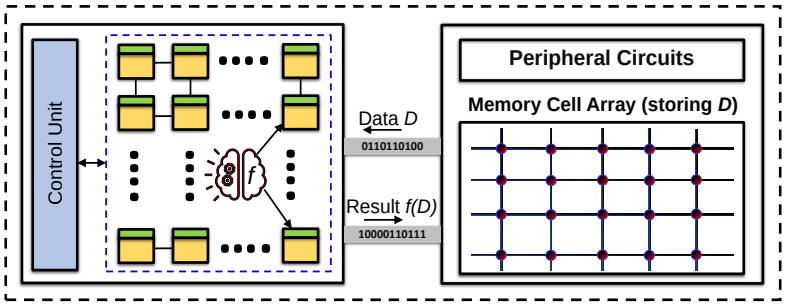


Fig. 5.6: Block representation of a systolic array based architecture [59, 63].

architecture within the systolic array. In addition, the processing elements and global buffers are clustered to support sparse DNNs [63].

The state-of-the-art hardware implementations of DNNs on ASICs primarily consist of three types of architectures: a) a systolic architecture and various modified versions [63, 195], b) a SIMD based architecture [246, 247], and c) a MIMD based architecture [248, 249]. Among the three, the systolic architecture is the most energy efficient due to the inherent capability of data reuse [63, 219, 250]. An overview of processing in a systolic array based custom DNN ASIC architecture is shown in Fig. 5.6. The general structure of a systolic array is a 2-D grid of computational units called processing elements (PEs), where the primary tasks of the PEs include the computation of both dense and sparse matrix-matrix and matrix-vector operations, communication with neighbor PEs, and passing data in both the vertical and horizontal directions of the array [193, 194]. The salient feature of the systolic array is that information (both input and results) flows between PEs in a pipeline fashion.

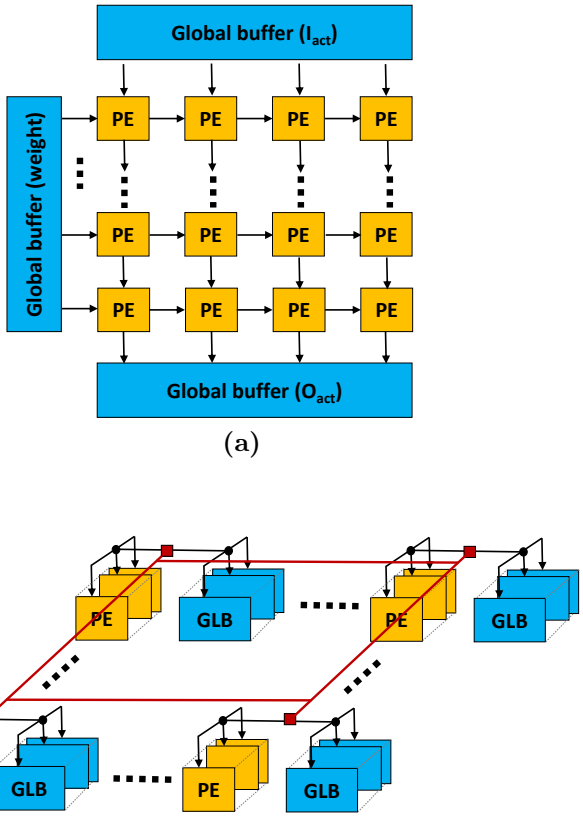


Fig. 5.7: Two versions of a systolic array architecture, which include a) conventional and b) clustered PEs with global buffers (GLBs).

Therefore, there is no need to store the intermediate results and any exchange of data with components outside of the array occurs only at the PEs that are on the boundary of the systolic array [63, 214]. The key motivation behind the utilization of systolic arrays is the capability of parallel computation to accelerate any special purpose computation as shown in Fig. 5.5. The most commonly executed computations in DNNs including convolutions and matrix-matrix, matrix-vector, and non-linear functions

are decomposed into a sequence of simpler calculations such as a multiply and accumulate and are executed in parallel in the systolic array [53, 63, 214, 216, 217]. In general, compute bound applications are ideal for hardware acceleration [59]. A more detailed depiction of the computation unit and memory architecture of a systolic array is shown in Fig. 5.7, where two configurations of the systolic array are provided. However, the fundamental properties of the systolic array remain the same for both configurations. Note the PEs shown in Fig. 5.7 are generic and represent different implementations that include MAC units implemented with conventional CMOS and FinFET based circuits (such as TPU and Eyeriss) [59, 63], in-memory computing units (processing-in-memory (PIM) accelerators) [251, 252], and analog computation units such as the Mythic AI accelerator [62].

5.4 Memory Requirements of AI Accelerators

Regardless of the model, application, and hardware architecture, all DNN accelerators require a sufficiently large data set, where the data is primarily categorized into three types: 1) input activation, 2) output activation, and 3) weight (or filter) as shown in Fig. 5.5. The DNN accelerators are efficient in performing convolution operations on an array of processing elements (PEs), where each PE is composed of single or multiple multiply-and-accumulate (MAC) units and local memories [63]. However, storing and moving a large amount of data poses a key challenge to improving the

energy efficiency of a DNN accelerator [63, 239]. The large data sets required for DNN inference are stored in a combination of off-chip and on-chip memory, while the choice between off-chip and on-chip memory is based on the fundamental trade-off between latency and energy consumption [53, 239, 253].

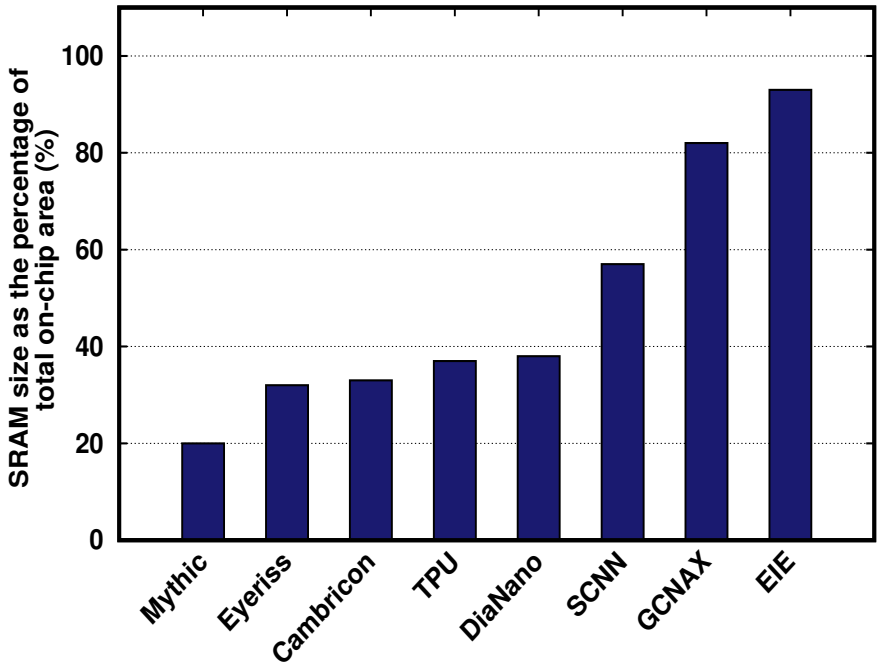


Fig. 5.8: The size of on-chip memory utilized in AI accelerators that were developed in the last five years.

The use of on-chip memory such as SRAM or embedded DRAM (eDRAM) reduces the latency of read and write operations at the cost of increased circuit area. In contrast, off-chip memory such as DRAM does not occupy on-chip area, while resulting in a significant increase in the latency and energy consumption of the overall

system [53, 239, 253, 254]. The use of off-chip memory for edge devices with stringent energy budgets is not optimal as off-chip memory consumes orders of magnitude more energy than on-chip memory [53, 239, 253]. For example, a single off-chip DRAM access consumes 200× more energy than a MAC operation, while a single on-chip SRAM access consumes approximately 6× more energy than a MAC operation [194, 239]. Therefore, many current DNN accelerators store a significant portion of the active data set in on-chip SRAM to avoid expensive (increased latency and energy per access) off-chip data traffic and to improve the overall power-performance tradeoff [53, 194, 239, 253, 255].

For example, a 10-bit DNN accelerator implemented to operate at a near-threshold voltage (NTV) of 0.4 V utilizes separate memories of 400 KB each for the activation and weight parameters [255]. In addition, a 16-bit Eyeriss accelerator includes 181.5 KB of on-chip memory, where 0.5 KB of local memory is dedicated to each PE [256]. Furthermore, the DaDianNao accelerator utilizes separate on-chip memory spaces of 4 MB and 32 MB for, respectively, the input/output activations and weights [257]. Similarly, the Efficient Inference Engine (EIE) accelerator includes separate on-chip memory locations for activations (128 KB) and weights (10.2MB) [238]. The Sparse Convolutional Neural Network (SCNN) accelerator stores all activations in on-chip memory (1.2 MB), while the weights are fetched from off-chip memory through a 32 KB FIFO [215]. The size of the on-chip SRAM as a percentage of the total area of

the AI accelerators that were developed in the last five years is shown in Fig. 5.8 [59, 61–63, 215, 238, 258, 259]. The on-chip SRAM occupies 20%, 32%, 33%, 37%, 38%, 57%, 82%, and 93% of the total circuit area of, respectively, the Mythic [62], Eyeriss V2 [63], Cambricon-X [258], TPU [59], DiaNano [61], SCNN [215], GCNAX [259], and EIE [238] accelerators.

A large on-chip memory, however, results in a significant amount of leakage energy, where the leakage loss is further increased with technology scaling [54–56]. For example, the power consumption of an idle TPU that consists of 29.3 MB of on-chip memory is 28 W, which is equivalent to 70% of the 40 W of total power consumed while in active mode [59]. In addition, the power consumed by the on-chip memory corresponds to 86.64% of the total system power consumption of the NTV accelerator [255]. More recently, architectural techniques were proposed to address the power overhead due to data traffic including 1) in-memory computing, where computing is performed within the memory array [205] and 2) near-memory computing, where computing units are placed adjacent to the memory array [199].

5.5 Dataflows for AI Accelerator

Dataflow is one of the primary attributes of state-of-the-art AI accelerators that impacts the power, performance, and throughput. Due to the simultaneous processing of a large number of filters and channels when executing high-dimensional convolution

operations, a massive amount of data transfer is required in an AI accelerator [60]. Therefore, the energy consumption to store and move large amounts of data exceeds that of the energy consumed for computation [60]. The fundamental objective of parallel processing with DNN models is achieved due to the reuse of data within the target hardware. Therefore, an appropriate mapping is required based on the shape and size of the implemented DNN model as well as the configuration of the target hardware, which is a complex optimization problem. The computation performed by each layer of a DNN model is decomposed into many simple matrix-matrix and matrix-vector operations. The dataflow algorithm optimally maps the computations to the available hardware resources such that: 1) data reuse is maximized, 2) maximum parallelism is achieved, and 3) the memory at each level of the hierarchy is utilized such that both the latency and energy consumption is minimized [218].

The implementation of effective dataflow for deep neural networks is an emerging research area. The primary and most widely utilized dataflows to date are weight stationary (WS), output stationary (OS), row stationary (RS), and no-local-reuse (NLR). The WS dataflow provides stationary filter weights such that filter reuse is improved, while input feature maps are broadcasted to all PEs simultaneously [59, 260, 261]. In addition, the partial sums (psums) are accumulated spatially across the PE array. Implementations of the WS dataflow include a CNN architecture developed by NEC Laboratory [260], a low power coprocessor for mobile applications developed

by Purdue University [261], and the TPU developed by Google [59]. In contrast, the OS dataflow keeps the psums static in the PEs to reduce the energy consumed due to the movement of psums. The filter weights are broadcasted to all PEs simultaneously and input feature maps are passed spatially. Implementations of the OS dataflow include a CNN accelerator for vision processing developed by EPFL [262] and a low-precision DNN accelerator for learning with stochastic rounding developed by IBM [263]. The RS dataflow utilizes a temporal reuse of hardware resources in a spatial architecture, where a subset of MAC operations are grouped and executed temporally on the same PE in a specific order [218]. The Eyeriss accelerator developed by MIT is implemented as a RS dataflow [63]. The NLR dataflow keeps nothing stationary within the PE, which allows removal of the memory space (register file) withing each PE. The input feature maps and filter weights are directly swarmed from the global buffer and the psums are spatially accumulated. NLR implementations include a CNN accelerator called DianNao developed by the Chinese Academy of Sciences [61] and a FPGA based CNN accelerator developed by UCLA [222].

Table 5.1: AlexNet model simulated using two popular AI accelerator architectures.

Architecture	Array height	Array width	Ifmap SRAM Size (KB)	Filter SRAM size (KB)	Ofmap SRAM size (KB)	Dataflow	Number of cycles per layer ($\times 10^4$)					Current per layer (mA)				
							CONV1	CONV2	CONV3	CONV4	CONV5	CONV1	CONV2	CONV3	CONV4	CONV5
TPU	256	256	8192	8192	8192	OS	0.46	38	0.47	0.7	0.38	45.21984	3735.552	46.20288	68.8128	37.35552
TPU	128	128	8192	8192	8192	OS	0.85	155	0.7	1.05	0.7	20.8896	3809.28	17.2032	25.8048	17.2032
TPU	128	128	8192	8192	8192	WS	0.98	158	0.2	0.4	0.2	28.90138	4659.6096	5.89824	11.79648	5.89824
Eyeriss	12	14	108	108	108	OS	63	15663	71	106	72	15.876	3947.076	17.892	26.712	18.144
Eyeriss	8	8	108	108	108	OS	160	39567	177	265	177	15.36	3798.432	16.992	25.44	16.992
Eyeriss	8	8	108	108	108	WS	162	39583	200	300	200	18.6624	4559.962	23.04	34.56	23.04

Each dataflow includes certain benefits and drawbacks. The dataflows are generally optimized based on the underlying DNN models implemented, the available

hardware resources, and the design objectives. In this research defense, the two most popular DNN accelerators (TPU and Eyeriss) are simulated using a cycle-accurate DNN simulator [216]. A server class computational workload is represented by the TPU, while edge computation for mobile devices is represented by the Eyeriss architecture. The simulation is carried out using the AlexNet model with two different dataflows (OS and WS). There are two reads and one write operation executed on each PE per clock cycle. Assume the total current drawn for each read operation is X nA, each write operation is $2X$ nA, and each MAC operation is $3X$ nA, where each PE executes only one MAC operation per cycle. The energy consumed for read/write and MAC operations is not known for the TPU and Eyeriss ICs. The value of X is assumed as 300 nA to match the total power consumption of a TPU with an array of 256×256 PE. The simulated result for two DNN accelerators with output stationary and weight stationary dataflows are listed in Table 5.1, where the number of clock cycles and total current drawn for each convolution layer are listed. There is a significant difference in the total number of cycles required to complete computation for each layer despite running on the same model (AlexNet), which is directly proportional to total current drawn per layer. In addition, the total current varies when the model is executed on the same hardware configuration with a different dataflow as shown in Table 5.1.

5.6 Near- and In-Memory Computing for Reduced Latency and Improved Energy Efficiency

More recently, in- and near-memory computing has gained traction in the research community to address the challenges of data movement in hardware accelerators [198–210, 230, 264–266]. In-memory computing encompasses techniques that perform computation where the data resides, specifically within the memory array [198–208]. Therefore, large data movement is not required with in-memory computing architectures. As a result, a significant improvement in both latency and energy efficiency is achieved with in-memory based computer architectures [198–210, 230]. The concept of in-memory computing was first proposed in the 1970s [267]. However, in-memory computing has gained popularity in the last five years due to the significant benefits provided with data driven workloads [198–210, 230].

An overview of an in-memory computing (IMC) architecture is provided through Fig. 5.9, which is labeled as Processing in Memory. The computation of data D is performed in the memory array storing data D. The computation is performed on the bitline (BL) of the cell array. The weight parameters are stored in the cell array and inputs are encoded in either the wordline (WL) or the pre-charge circuit as a direct binary signal or a binary weighted analog signal. For example, a binary weighted analog signal applied to the WL represents different binary signals based on

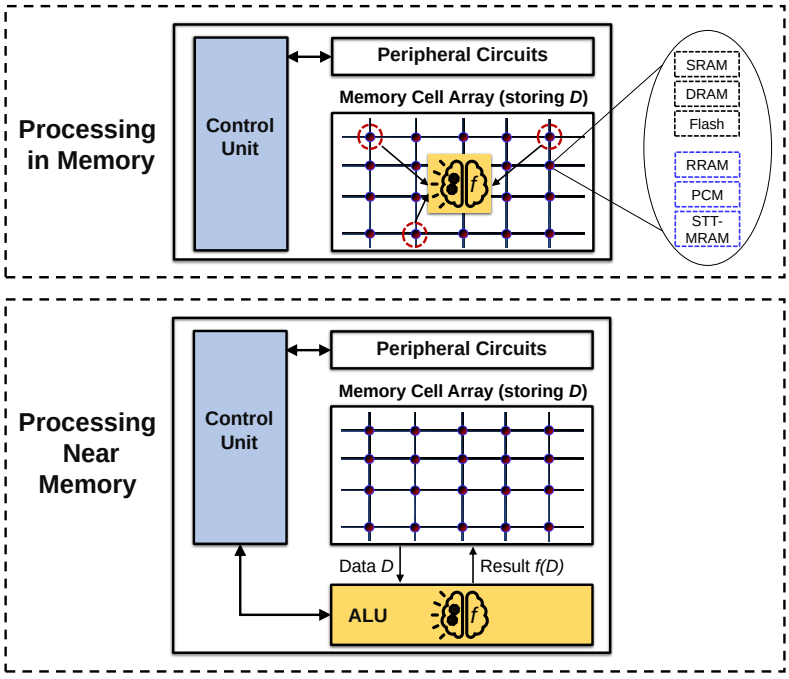


Fig. 5.9: Block representation of in-memory computing (IMC) and near-memory computing (NMC) circuits [199, 206, 207, 209, 210, 230, 264–266].

the applied voltage [198–206]. The bit cells discharge voltage on the BL based on the applied WL voltage and pre-charge circuit. The resulting voltage on the BL is applied to an analog-to-digital converter (ADC), which decodes the result of the bitline computation. Computation on the bitline supports several operations including multiply and accumulate (MAC), multiply and average (MAV), XNOR and accumulate, various boolean functions, and arithmetic operations [198–206]. Both volatile and non-volatile memory elements have been used as the bit cells of in-memory computing architectures [230]. In addition, both charge-based and resistance-based memory cells are used as the computing elements of in-memory systems in prior research,

where charge-based memory includes static random access memory (SRAM), dynamic random access memory (DRAM), and flash memory, while resistance-based memory includes phase change memory (PCM), resistive random access memory (RRAM), and spin-transfer torque magnetoresistive random-access memory (STT-MRAM) [230]. Additionally, implementing two modes of operation in in-memory computing architectures is possible, where 1) memory mode and 2) compute mode are utilized [198–206]. Modifications are required in the peripheral and array circuits to implement the two modes of operation.

Near-memory computing (NMC) has also been pursued to address the costs due to the large movement of data, where computing units are placed adjacent to the memory units [199, 203, 206, 207, 209, 210, 230]. While near-memory computing does not eliminate the need to move data, both the latency and energy required to transfer data are significantly reduced [199, 203, 206, 207, 209, 210, 230]. Near-memory computing was first introduced in the 1990s, where monolithic processing units were placed close to monolithic memory units [268]. An overview of near-memory computing is provided through Fig. 5.9, which is labeled as Processing Near Memory in the figure. Unlike in-memory computing, the computation unit of near-memory systems is capable of implementing any operation regardless of the utilized memory unit. In addition, no modification is required in the the peripheral circuits and cell arrays of the memory unit to implement near-memory computing.

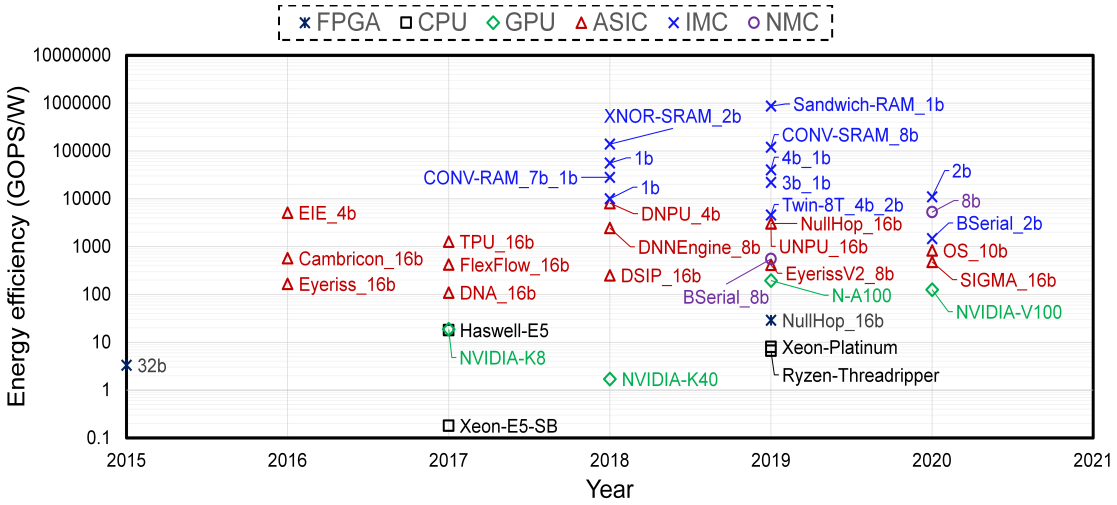


Fig. 5.10: Summary of the energy efficiency of AI accelerators developed in the last six years that are implemented on an FPGA, CPU, GPU, digital Von Neumann architecture based custom SoC (represented as an ASIC), an architecture based on near-memory computing (NMC), and an architecture based on in-memory computing (IMC).

Among all AI accelerator architectures, custom ASICs, in-memory computing, and near-memory computing are the most efficient. A summary of the energy efficiency of state-of-the-art AI accelerators developed in the last six years is provided in Fig. 5.10 [53, 61, 63, 193, 199–208, 214–217, 219–225, 239]. The energy efficiency is characterized in giga operations per second per watt (GOPS/W). In addition, the bit precision is appended with the label of each domain specific custom accelerator. For example, the EIE accelerator is labeled as 'EIE_4b' in Fig. 5.10, which implies that EIE is implemented to perform 4-bit computation. Only the bit precision is provided as a label for accelerators without a given name. The AI accelerators developed in FPGAs allow for quick prototyping and incur lower cost, while providing

lower energy efficiency and operating speeds [195, 222, 223]. For example, the FPGAs developed in 2015 and 2019 exhibit a peak energy efficiency of, respectively, 3.3 GOPS/W and 28.8 GOPS/W [222, 223]. Although CPU based systems consume less power than GPUs, the AI accelerators developed on the GPUs exhibit higher energy efficiency as compared to accelerators implemented on CPUs due to a) the availability of computational units that are optimized for a subset of functions and b) the higher parallelism provided by GPUs. Domain specific custom accelerators provide a superior energy efficiency. The maximum energy efficiency of DNPU_4b is 8100 GOPS/W, of Sandwich-RAM_1b is 866000 GOPS/W, and of 8b is 5270 GOPS/W as shown in Fig. 5.10, which represent the peak efficiencies for, respectively, an ASIC, IMC, and NMC architecture. Note that the higher energy efficiency of the IMC and NMC based accelerators is due to a) a higher memory bandwidth, b) lower data movement, and c) lower computational precision, where the precision of a majority of in-memory computing based accelerators is limited to 4-bits or less.

Both in-memory and near-memory computing based accelerators are well suited for data intensive neural network applications. The increase in computing power of a processing element in traditional Von Neumann based architectures is not beneficial if the memory bandwidth is not proportionally increased. Therefore, the performance of a majority of Von Neumann architecture based custom AI accelerators is determined by the maximum memory bandwidth [53, 61, 63, 193, 214–217, 219–225, 239]. The

Table 5.2: Advantages and challenges of in-memory and near-memory computing architectures for AI acceleration.

Computation platform	Advantages	Challenges
In-memory computing	1) Access to higher memory bandwidth 2) Higher energy efficiency as compared to NMC, CPU, GPU, and FPGA 3) Allows distributed and parallel computation 4) Least amount of data transfer is required (less than NMC)	1) Fixed and limited functionality 2) Lower precision due to a) quantization error during computation in analog domain and b) limited ADC precision 3) At any given time, a memory bank is used in either a) compute mode or b) memory mode 4) Challenges due to computation in the analog domain include a) non-linearity, b) higher sensitivity to PVT variation, and c) dependence on threshold voltage of transistors 5) Ultra-low voltage computation is a challenge 6) Requires ADC with higher precision and energy efficiency 7) Requires modification of memory banks or arrays, which increases a) design complexity and b) access latency 8) Allows only a subset of functions for computation 9) Leakage power increases with the increasing size of memory
Near-memory computing	1) Access to higher memory bandwidth 2) Higher energy efficiency as compared to CPU, GPU, and FPGA 3) Allows distributed and parallel computation 4) Requirement of data transfer is lower than CPUs, GPUs, FPGAs, and ASICs 5) Ultra-low voltage computation is possible 6) Modification to memory array or bank is not required 7) Allows any function for computation	1) Lower energy efficiency as compared to IMC 2) Leakage power increases with the increasing size of memory

in-memory and near-memory based accelerators allow for hierarchical and distributed computing within or near the memory, which is beneficial for highly data intensive and parallel computations [269]. The advantages and disadvantages of in-memory and near-memory based computing architectures for AI acceleration are listed in

Table 5.2 [198, 199, 201–210, 269]. Despite the greater energy efficiency provided by accelerators implemented with in-memory computing, challenges remain that must be addressed as listed in Table 5.2.

The primary disadvantages of in-memory computing include 1) limited precision for computation, 2) non-linear effects of analog computation, 3) higher sensitivity to process, voltage, and temperature (PVT) variations, 4) limited opportunity of supply voltage scaling due to higher sensitivity to PVT variation and non-linearity, and 5) higher leakage power consumption of the memory [198–208]. In comparison, near-memory computing provides an optimal solution for data intensive neural network applications, since NMC provides greater energy efficiency as compared to CPUs, GPUs, and FPGAs without the additional challenges associated with in-memory computing. Note that the loss due to the leakage current of the large on-chip SRAM memory remains similar in both in-memory and near-memory accelerators.

Table 5.3: Leakage power consumption of SRAM cells [17, 270, 271].

SRAM cell	6T (65 nm)	6T (7 nm)	8T (32 nm)
Leakage power (pW)	95.25	27.5	114.44

The need for larger on-chip memory is common to all AI architectures including for accelerators implemented in CPUs, GPUs, custom ASICs, near-memory computing, and in-memory computing due to the execution of data-intensive workloads [53, 61, 63, 193, 198–206, 214–217, 219–225, 239]. A summary of the size of on-chip SRAM utilized in state-of-the-art domain specific custom accelerators is provided through

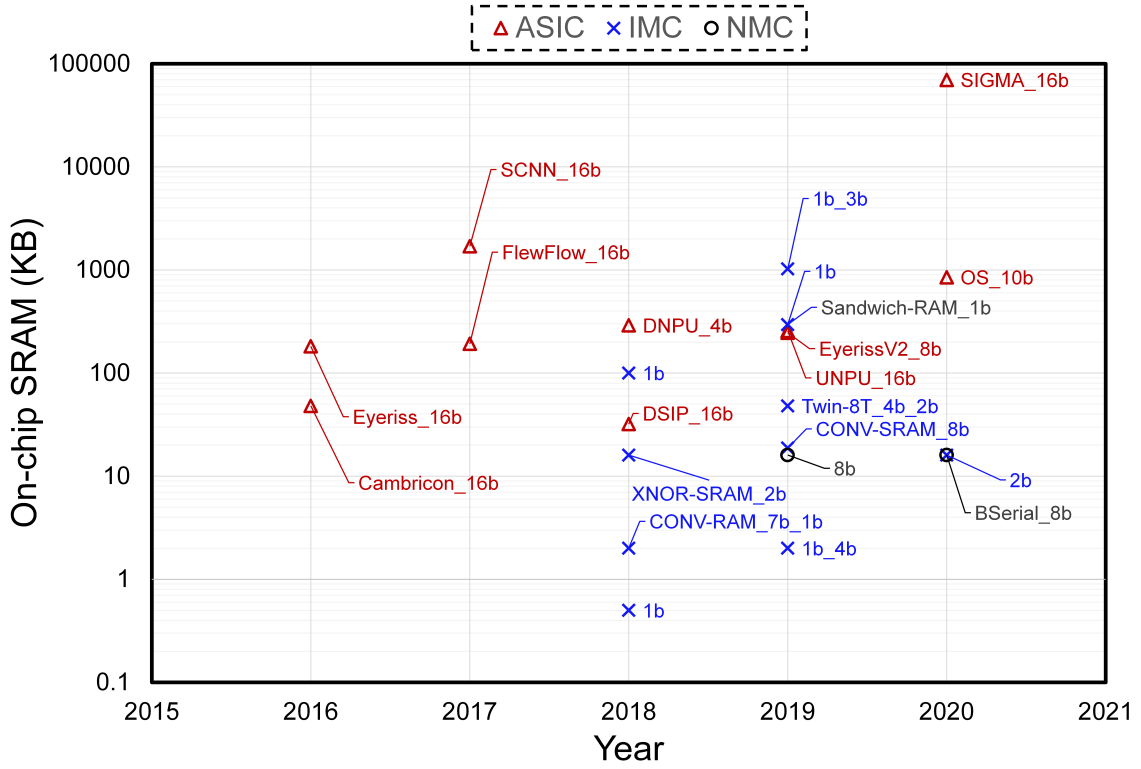


Fig. 5.11: Summary of on-chip memory utilized in state-of-the-art custom AI accelerators including Von Neumann based digital SoCs, which are represented as ASICs, in-memory computing (IMC), and near-memory computing (NMC) in the figure.

Fig. 5.11 [53, 63, 193, 198–208, 217, 219–225, 239]. Regardless of the computing platform, a large on-chip SRAM is utilized in the majority of the accelerators. For example, the size of the on-chip SRAM is more than 10KB for a majority of the accelerators shown in Fig. 5.11. A large on-chip memory, however, incurs a large loss in leakage power. The leakage power consumption of 6T and 8T SRAM bit cells are listed in Table 5.3, where a single bit is stored per cell [17, 270, 271]. As more transistors are added to a bit cell, the leakage power increases [17, 270, 271]. Despite the inherent reduction in leakage current due to FinFET technology, the leakage power

consumption of a 6T SRAM bit cell implemented in the 7nm FinFET node is 27.5 pW [17, 271]. Note that the loss in leakage power linearly increases with the size of the SRAM bit cell array. For example, the leakage power of an idle 1 KB 6T SRAM array is at least 0.78 mW when implemented with 6T bit cells in a 65 nm CMOS technology, where the leakage power of the peripheral circuits is not considered [270].

5.7 Summary

In this chapter, an overview of AI acceleration and deep neural networks is provided. The hardware implementation of deep neural networks in CPU, GPU, custom ASICs, and near-/in-memory computing based architectures is discussed. Due to significant degradation in the latency and energy efficiency of data driven workloads, the implementation of AI applications on custom ASICs is preferred over implementations on CPU and GPU based architectures. However, the high cost of data transfer remains in Von Neumann based architectures. Near- and in-memory computing architectures are developed to reduce the amount of data transfer and to consequently reduce the latency and power consumption. Despite the increased energy efficiency and performance with near- and in-memory computing, the increase in on-chip memory sizes required for AI workloads significantly increase the energy loss due to leakage current due to the idle memories of AI accelerators.

Chapter 6

Circuits and Techniques for Ultra-Low Voltage Operation

Circuits developed with differential input and output signaling are well suited for high-speed and robust operation as compared to traditional singled-ended logic such as CMOS and dual mode based logic families [132–134]. In this chapter, three novel types of dynamic differential signaling based logic (DDSL) families are developed for ultra-low supply voltages; specifically, 1) dynamic current mode logic (DCML), 2) latched DCML (LDCML), and 3) dynamic feedback current-mode logic (DFCML) [164, 165, 272]. In addition, the interfacing of ultra-low voltage circuits, such as near- and sub-threshold circuits, with circuits operating at higher operating voltages is developed for overall system-on-chip integration [273]. An overview of the DDSL

logic families is provided in Section 6.1. The implementation of boolean functions and a multiplier implemented with the DDSL logic family is discussed in Section 6.2. The characterization of the DDSL families is provided in Section 6.3. In addition, a level shifter circuit and a bi-directional input/output (I/O) circuit developed for ultra-low voltage computing are discussed in Section 6.4.

6.1 Dynamic Differential Signaling Based Logic Families

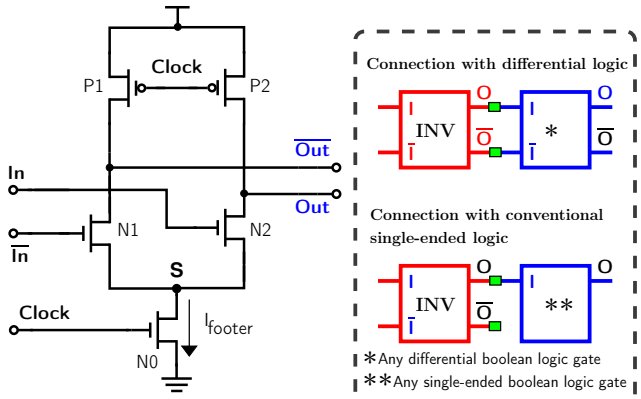
While supply voltage scaling improves energy efficiency, the challenge is to achieve low voltage operation without significantly impacting the performance and robustness of the circuit. In this work, robust logic families are proposed for near-threshold computing (NTC) that function at low voltage while improving the performance and robustness to noise as compared to CMOS and CML. The DDSL circuits mitigate energy loss due to the static current path through the footer transistor of CML gates by operating in two phases: ***1) pre-charge***, and ***2) evaluate***. As the charging and discharging paths never turn on simultaneously, when assuming that no clock skew exists in the clock network, the two phase operation of DDSL circuits also reduces the short-circuit current, which is more pronounced in conventional CMOS circuits operating at near-threshold. Both the delay and power consumption of DDSL circuits

are, therefore, reduced at near-V_t operating voltages [274].

In order to benefit from the increased operating frequency, reduced power consumption, and greater robustness to noise provided by DDSL, a family of three logic circuits are developed and characterized for NTC: 1) dynamic current mode logic (DCML) described in Section 6.1.1, 2) latched DCML (LDCML) described in Section 6.1.2, and 3) dynamic feedback current-mode logic (DFCML) described in Section 6.1.3. The implemented DCML and LDCML families are based on a DCVSL gate [137–139], while the DFCML family is based on a static FCML gate [275]. The implemented logic families are described as *dynamic* and *current-mode* since functional principles of both dynamic and current-mode logic circuits apply [140]. The DDSL families differ in the structure of the pull-up network (PMOS devices) and the two differential branches. However, the fundamental operation of all DDSL circuits is similar.

6.1.1 Dynamic Current Mode Logic (DCML)

The schematic of a DCML inverter is shown in Fig. 6.1. Unlike conventional CMOS gates, the circuit operation is controlled by two identical differential current paths. In addition, the current I_{footer} through the gate is regulated by an applied voltage on the footer transistor *N0*, which effects the input voltage and operating characteristics of the next stage. During the pre-charge phase, both the Out and $\overline{Out}$



	In	$\overline{\text{In}}$	Input	Out	$\overline{\text{Out}}$	Output
State 1	0	1	0	1	0	1
State 2	1	0	1	0	1	0

Fig. 6.1: Implementation of an inverter with dynamic current-mode logic (DCML). The differential logical operation of the DCML inverter is provided in the table. The output of a DCML inverter is connected to inputs of either a single-ended or differential signal based logic gate as shown by the sub-figure in the dotted box.

nodes are charged to V_{dd} and the entire circuit is disconnected from the V_{gnd} node. During the evaluate phase, depending on the signals on In and $\overline{In}$, Out or $\overline{Out}$ is discharged through, respectively, transistor *N2* or *N1*.

The logical evaluation of the inverter is listed in the table included as part of Fig. 6.1. The voltage difference between In and $\overline{In}$ is the ***Input*** (V_{In} - $\overline{V_{In}}$) of the logic gate, and the difference between Out and $\overline{Out}$ is the ***Output*** (V_{Out} - $\overline{V_{Out}}$). In order to evaluate the functionality of a DCML inverter, V_{dd} is applied to In (V_{In}) and V_{gnd} to $\overline{In}$ ($\overline{V_{In}}$). The difference between In and $\overline{In}$ is $V_{dd} - V_{gnd} = V_{dd}$, which is evaluated as logic high. During the evaluation phase, Out discharges through *N2* and the $\overline{Out}$ node remains high. If the output is taken as the difference between Out and $\overline{Out}$, the resulting -V_{dd} is evaluated as logic low, and the input signal is

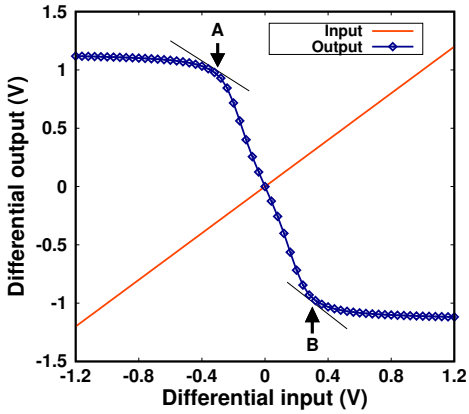


Fig. 6.2: Voltage transfer curve of a differential inverter for a V_{dd} of 1.2 V.

inverted. If, however, the output is evaluated as the difference between $\overline{Out}$ and Out, the circuit behaves as a buffer. Explicit inversion is, therefore, not required for dynamic current-mode logic.

The input-output characteristics of the DCML inverter are described by the voltage transfer curve (VTC) shown in Fig. 6.2. The points where the derivative of the VTC curve is -1 are labeled as A and B. The differential voltage of the output to the left of point A represents logic high, and the voltage of the output to the right of B represents logic low. As In is increased with respect to $\overline{In}$, more current flows through branch 2 as compared to branch 1, which results in a decrease in the voltage at Out, while $\overline{Out}$ remains unchanged. As a result, the differential ***Input*** and ***Output*** voltage of the logic circuit begins to increase and decrease, respectively.

A chain of inverters is used to evaluate the self restoring property and drive capability of the DCML inverter and to characterize the delay for a rising and falling output voltage of the logic families. The overdrive voltage V_{od} given by (6.1) [84] is an

important parameter used to regulate the output voltage swing. Depending on the voltages at the terminals and the transistor dimensions, the overdrive voltage varies between 20 to 120 mV and 0 to 100 mV for, respectively, the PMOS and NMOS transistors of a 130 nm CMOS process. Assuming a set supply voltage, and therefore gate voltage, modifying the width is the most effective means to set the overdrive voltage of the transistors. However, increasing the width results in a larger intrinsic load capacitance, which produces a longer propagation delay. There is, therefore, a trade-off and upper limit between reducing the power consumption and preserving the performance when modifying the transistor width.

$$V_{od} = (V_{GS} - V_T) = \sqrt{\frac{2I_{footer}}{\mu_n C_{ox} \frac{W}{L}}} \tag{6.1}$$

Assuming that a single clock source is used throughout the circuit, a chain of DCML gates lacks the self-restoring property due to a floating state in the evaluation phase. As a result, within a given clock cycle, the circuit does not evaluate correctly after a certain number of sequentially connected logical stages as the stored charge at the differential output node gradually leaks. For the chain of CMOS inverters operating at a near-V_t supply voltage of 400 mV, including more than four logical stages of CMOS gates is a challenge even with transistors that are sized 5× larger as the full voltage swing is not obtained at the output within the user specified limit of

85% of the input signal period, which is used to determine the maximum operating frequency of the inverter chain. However, at a 400 mV supply voltage and a 1 MHz operating frequency, a chain of DCML inverters functions properly with up to eight sequentially connected logical stages. The logical output is discernible with the use of differential signals, even with a 20 to 30 mV reduction in the voltage swing. At near-V_t operating voltages, a larger number of stages is, therefore, feasible with DCML logic families at a cost of reduced frequency.

6.1.1.a Gain of DCML Inverters in Near-Threshold

The gain of a DCML inverter is a function of the supply voltage, threshold voltage, and process parameters. The conventional CMOS dynamic logic inverter exhibits the maximum gain when the voltage of the output node is close to the threshold voltage of the inverter [276]. The voltage at the output nodes of DCML (shown in Fig. 6.1) is V_{dd} at the beginning of the evaluate phase. Therefore, the current of an NMOS transistor operating in the linear region, as given by (6.2), is considered. The current through transistors N1 and N2 is, respectively, I_1 and I_2. The total current through the footer transistor I_{footer} is the sum of the currents I_1 and I_2.

$$I_D(linear) = \mu C_{ox} \cdot \frac{W}{L} \cdot \left[(V_{GS} - V_T) \cdot V_{DS} - \frac{V_{DS}^2}{2} \right] \tag{6.2}$$

$$\begin{aligned} V_{diff,out} &= |V_{Out} - V_{\overline{Out}}| && (6.3) \\ &= \Big(V_{dd} - I_2 \cdot (R_{N2} + R_{N0}) \Big) - V_{dd} \\ &= I_2 \cdot (R_{N2} + R_{N0}) \\ &= I_{footer} \cdot (R_{N2} + R_{N0}) && (6.4) \\ V_{diff,in} &= V_{In} - V_{\overline{In}} \\ &= (V_{dd} - V_{gnd}) \\ &= V_{dd} \\ &= V_{od} + V_T && (6.5) \end{aligned}$$

The differential output voltage is given by (6.3), where at the end of the pre-charge phase the voltages at the nodes Out and $\overline{Out}$ are, respectively, $V_{dd} - I_2(R_{N2} + R_{N0})$ and V_{dd}. When applying a differential input of V_{dd} and V_{gnd} for, respectively, In and $\overline{In}$, the charge on Out begins to discharge through N2 and N0, resulting in a $V_{diff,out}$ as given by (6.4). The channel ON resistance of NMOS transistors $N1$ and $N2$ is, respectively, R_{N1} and R_{N2}. In addition, $R_{N0} = y \cdot R_{N1} = y \cdot R_{N2}$ as $R_{N1} = R_{N2}$ since both $N1$ and $N2$ are set to the same width and length. The variable y represents the ratio of the width of either transistor $N1$ or $N2$ to the width of the footer transistor

$N0$ ($y = \frac{W_{N1}}{W_{N0}} = \frac{W_{N2}}{W_{N0}}$). For In and $\overline{In}$ set to, respectively, V_{dd} and V_{gnd}, the resulting differential input is, therefore, given by (6.5).

The maximum gain of the DCML inverter, as given by (6.6), is calculated using (6.4) and (6.5). The gain is further expanded by substituting into (6.6) the equation of the channel ON resistance $R_N = \frac{1}{\mu_n C_{ox}(V_{GSn}-V_{Tn})\frac{W_n}{L_n}}$ and the equation of the current I_{footer} through $N0$, which results in (6.7). The factor m in (6.7) is defined as the ratio of drain voltage V_{DS} to the overdrive voltage $V_{od} = \big(V_{GS} - V_T\big)$, where $0 \leq m \leq 1$. Note that the gain is primarily dependent on the threshold voltage, supply voltage, the ratio of the ON resistance of $N0$ to $N2$ or $N1$, and the ratio of drain voltage to the overdrive voltage.

$$\begin{aligned}
\text{gain} &= \frac{I_{footer} \cdot (R_{N2} + R_{N0})}{V_{dd}} \\
&= \frac{R_{N2}(1+y)\mu_n C_{ox} W\left[(V_{GS} - V_T)V_{DS} - \frac{V_{DS}^2}{2}\right]}{L \cdot V_{dd}} && (6.6) \\
&= \frac{(V_{GS} - V_T) \cdot (1+y) \cdot \left(m - \frac{m^2}{2}\right)}{V_{dd}} && (6.7) \\
&= \frac{V_{od} \cdot (1+y) \cdot \left(m - \frac{m^2}{2}\right)}{V_{dd}} && (6.8)
\end{aligned}$$

6.1.2 Latched Dynamic Current Mode Logic (LDCML)

The LDCML circuits operate similar to the DCML circuits, as both operate in two phases. However, the LDCML inverter includes two minimum sized PMOS transistors

(P3 and P4) used as keepers, as shown in Fig. 6.3 [134]. The keeper transistors preserve the output voltage during the evaluation phase if charge leaks at either of the two output nodes. The LDCML logic family, therefore, provides the self-restoring property for a larger number of logical stages, as compared to DCML circuits. As the keepers prevent voltage drop or charge loss at the output nodes, the circuit is robust to external noise. Aside from leakage current, the P3 and P4 transistors do not consume additional power as there is no direct current path through P3/N1 or P4/N2 (either N1 or N2 is off). In addition, the area overhead is negligible as minimum sized transistors are used.

For the LDCML inverter, once the input signals are applied, there is a hysteresis in the voltage transfer curve due to the keeper transistors $P3$ and $P4$. For example, the voltage difference between In and $\overline{In}$ has to be sufficiently large (approximately 180 mV) to flip the potentials of Out and $\overline{Out}$ from the previously held values. Since Out was at 450 mV for the previous input conditions, $N2$ must sink enough current to overcome the current supplied by the keeper ($P4$) and discharge the Out terminal. As a result, the potential at the gates of $P3$ and $P4$ are also flipped. The hysteresis response of LDCML gates results in improved noise margins for a given differential input.

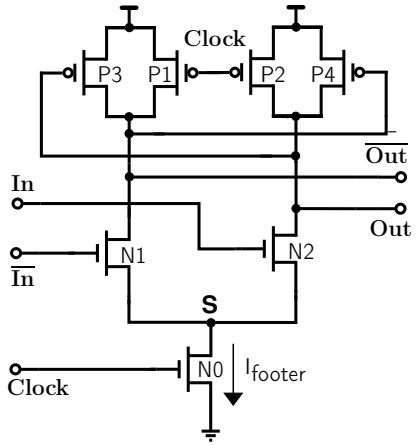


Fig. 6.3: Implementation of an inverter with latched DCML.

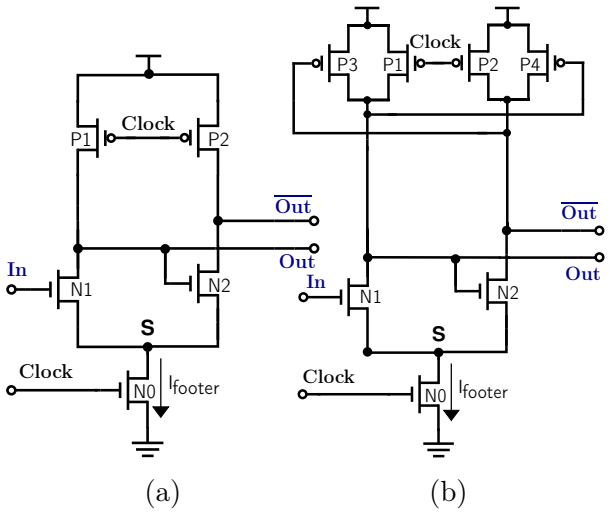


Fig. 6.4: Implementation of an inverter with a) dynamic feedback current-mode logic (DFCML) and b) latched DFCML (LDFCML).

6.1.3 Dynamic Feedback Current Mode Logic (DFCML)

The DFCML family performs pseudo differential operation, as a single ended input In results in a differential output Out and $\overline{Out}$, as shown in Fig. 6.4(a). The transistors $N1$ and $N2$ are controlled by, respectively, the input signal and the feedback signal from one of the differential output nodes.

The DFCML circuits follow the operating principles of both dynamic logic and feedback current mode logic (FCML), which was introduced for circuits operating at super-V_t supply voltages [275]. Despite having higher noise immunity and less sensitivity to both parasitic impedances and to process, voltage, and temperature variations, FCML circuits sink a significant amount of current due to the static current path through the tail transistor. The energy efficiency is, therefore, improved with DFCML circuits as there is no longer a static current path through the footer transistor, while also providing the improved performance, low voltage of operation, and higher noise immunity offered by FCML. The operation of DFCML circuits is similar to DCML and LDCML circuits as there is both a pre-charge and evaluate phase. However, the proper sizing of the input ($N1$) and feedback ($N2$) transistors is critical for the proper operation of the DFCML circuits. As both the output nodes are precharged to V_{dd}, $N1$ must be sufficiently large to discharge the Out terminal quickly enough to assure that $N2$ turns off and prevents charge sinking from $\overline{Out}$. For the DCML inverter, the input transistor $N1$ is approximately nine times larger than the feedback transistor $N2$.

The latched DFCML (LDFCML) circuit is also characterized, with a topological structure as shown in Fig. 6.4(b). With LDFCML, the transistor count is less than LDCML when the gate includes more than one input, which results in lower area and power consumption. Despite the pseudo differential operation, LDFCML benefits

from a higher immunity to external noise, less sensitivity to parametric variation, and low voltage operation, similar to a fully differential inverter.

6.2 Implementation of Boolean Functions and a Multiplier

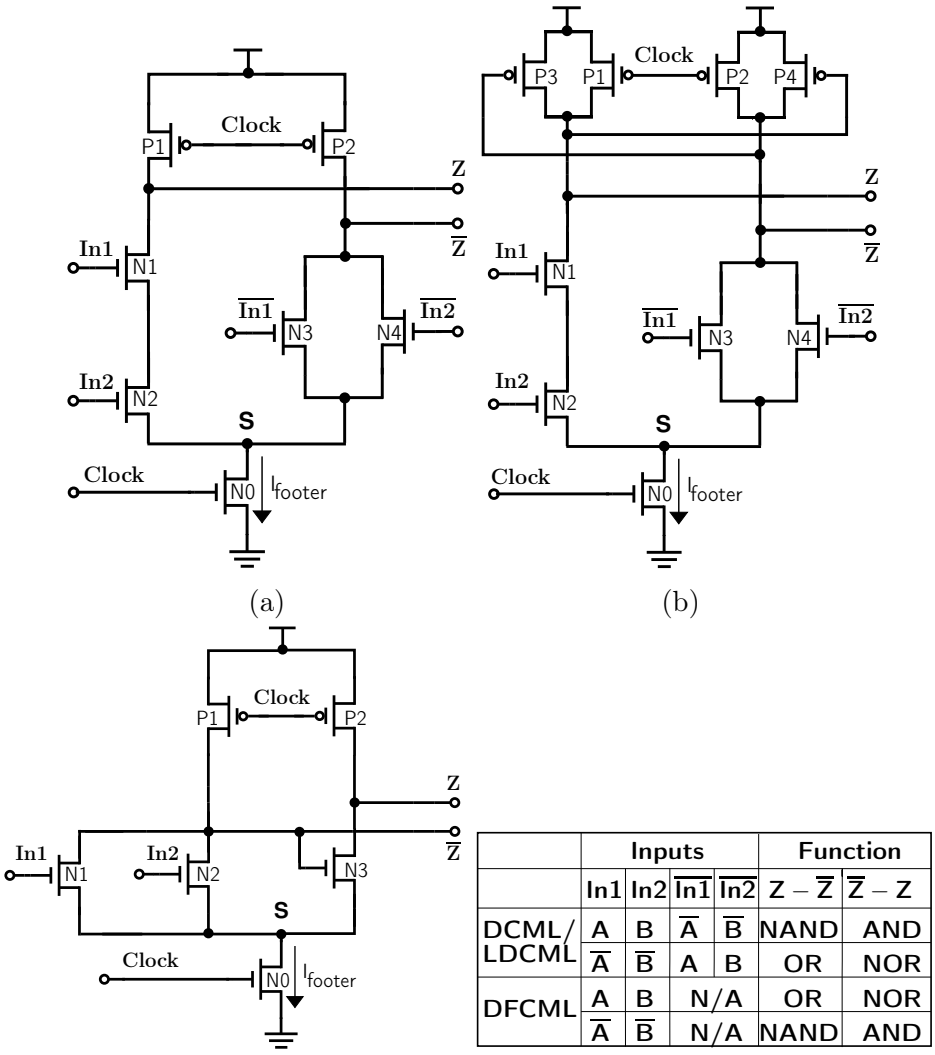
In addition to inverters, boolean functions including the AND, NAND, NOR, and OR and a 4×4 bit array multiplier are implemented in each DDSL circuit family for operation at near-V_t supply voltages. The implementation of universal gates is presented in Section 6.2.1, while the multiplier is described in Section 6.2.2.

6.2.1 Universal Gates

DDSL circuits allow for a single topology to function as an AND, NAND, OR, or NOR based on the configuration of both the inputs and outputs of the gate. The circuit is, therefore, described as a universal gate (UG). Characterization of the proposed logic families, including when implemented as universal gates, provides an analysis of the benefits and challenges of realizing DDSL circuits. In this work, two input universal gates are implemented using the DCML, LDCML, and DFCML families as shown in Fig. 6.5, where each gate realizes up to four boolean functions. The circuit operation of the DCML, LDCML, and DFCML universal gates is similar to

the operation of, respectively, the DCML, LDCML, and DFCML inverters discussed in Section 6.1. The proposed gate topologies are compared with CMOS NAND and NOR gates and CML universal gates. As a single DCML circuit implements four logical functions, the design complexity, design time, and cost of a digital logic circuit is reduced. In addition, unlike standard CMOS, it is not necessary to invert the output of a NAND or NOR gate to implement, respectively, an AND or OR gate. Therefore, the use of the DCML families results in a reduction in the area, power consumption, and delay as compared to CMOS gates.

Additional inverters are not required in the charging path (the pull-up network) when increasing the number of inputs. However, like CMOS NAND and NOR gates, two MOS transistors are required for each additional input included in DCML and LDCML gates. Therefore, there is a limit in the maximum number of inputs permitted within a gate for a target current and performance requirement as the voltage headroom is reduced when increasing the number of NMOS transistors in series. In order to maintain proper functionality, a minimum DC voltage difference of approximately 200 mV (worst case among all differential logic families) is required between A and $\overline{A}$ and B and $\overline{B}$. Since the DCML circuits are operating at near-V_t voltages, the voltage headroom is an important parameter that must be met to maintain a proper drain to source voltage (V_{DS}) bias. In this work, no significant drop in voltage headroom is observed for gates with *two inputs*. Unlike DCML and LDCML gates,



	Inputs				Function	
	In1	In2	$\overline{In1}$	$\overline{In2}$	$Z-\overline{Z}$	$\overline{Z}-Z$
DCML/ LDCML	A	B	$\overline{A}$	$\overline{B}$	NAND	AND
	$\overline{A}$	$\overline{B}$	A	B	OR	NOR
DFCML	A	B	N/A		OR	NOR
	$\overline{A}$	$\overline{B}$	N/A		NAND	AND

(d)

Fig. 6.5: Implementation of universal gates using a) dynamic current-mode logic (DCML), b) latched DCML (LDCML), c) dynamic feedback current-mode logic (DFCML), and d) the implementation of four logical functions using two different input and output combinations.

the DFCML universal gate does not experience a reduction in the voltage headroom as the number of input signals increases, since each additional input is added in parallel to existing inputs that are directly connected to the tail transistor as shown in Fig. 6.5(c).

6.2.2 Multiplier Implementation

A 4×4 bit array multiplier, which includes 16 AND gates, 4 half adders, and 8 full adders, operating at a supply voltage of 450 mV is implemented in each logic family to characterize the worst-case delay and power consumption. The circuit diagram of a 4×4 bit array multiplier is shown in Fig. 6.6. An array multiplier incurs a longer critical path delay and consumes a greater amount of power as compared to other multipliers such as the carry-save multiplier, the wallace-tree multiplier, and the booth multiplier [274]. The propagation delay is measured on the critical path, which is between the least significant bit of one of the inputs and the most significant bit of the multiplier output [274]. The multipliers are implemented using the previously described inverters and universal gates as well as half adders and full adders designed in each logic family. The interconnect between differential signal based logic gates is similar to the method shown for a DCML inverter in Fig. 6.1.

For the CMOS multiplier, 24 additional buffers are implemented to assure full-swing at the outputs. An additional 24 and 14 buffers are used between the logic gates

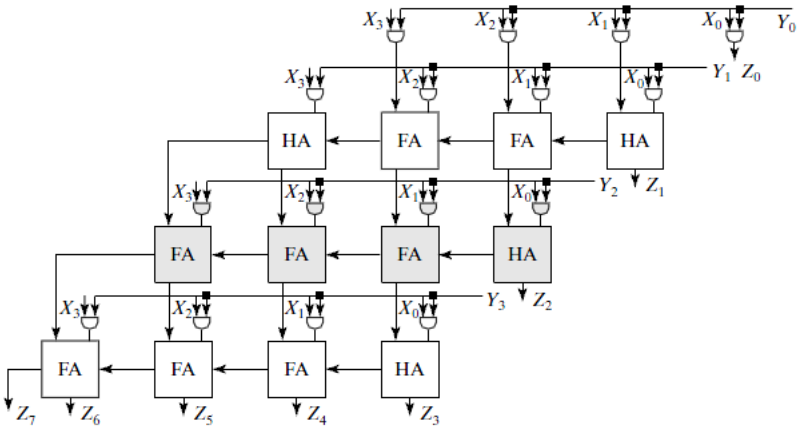


Fig. 6.6: Circuit diagram of a 4×4 bit array multiplier [274].

of the multipliers implemented with, respectively, DCML and DFCML to restore full logical swing as discussed in Section 6.1.1. However, no additional buffers are needed for the CML and LDCML multipliers. The 14 buffers required for the DFCML multiplier account for the larger resistance on one of the discharging paths of each gate, which include transistors of smaller size.

6.3 Evaluation of DDSL Families Through SPICE Simulation

The power consumption, maximum operating frequency, and area are characterized through implementation of boolean functions and a 4×4 bit array multiplier. A chain of four inverters is used to characterize the energy-delay product and the power consumption at three activity factors (25%, 50%, and 100%). In addition, the robustness of the logic families to noise and variability is evaluated by analyzing the

noise margins of a chain of inverters and the threshold voltage variation of universal gates. The 4×4 array multiplier is used to characterize process, voltage, and temperature variation through corner analysis, and characterization of statistical local and global variation through Monte Carlo analysis of the propagation delay and total power consumption. The SPICE simulations are performed using Cadence Spectre on a 130 nm CMOS technology, where the near-V_t voltages of 400 mV and 450 mV are considered. For the 130 nm fabrication process, the threshold voltage of the NMOS and PMOS transistors is, respectively, 0.35 ± 0.05 V and -0.33 ± 0.05 V. The maximum operating frequency of the logic families is determined as the time required for the output node to reach the full voltage swing within 70% of the active high clock for DCML, LDCML, and DFCML or 70% of the signal pulse width for CML and CMOS. In other words, the maximum allowed propagation delay is set to 70% of the evaluation phase (equivalent to 85% of the signal period for CML/CMOS), assuming a 50% duty cycle. At a near-V_t voltage of 400 mV, for an input signal with a period of 16.66 ns (60 MHz operating frequency), the output is determined such that a full voltage swing is obtained within 14.161 ns.

Table 6.1: Comparison of the performance, area, and power consumption of an inverter chain operating at a 400 mV supply voltage.

	CMOS	CML	DCML	LDCML	DFCML
f_{max} with 4 stages (MHz)	63	70	115	100	70
Area $(\mu m)^2$	2.88	10.56	4.56	4.85	3.0576
Static power (nW)	0.54	4550	0.2	0.14	0.12862
Dynamic power (nW)	544.46	300	368.8	393.86	240.97
Total power (nW)	545	4850	369	394	241.1

6.3.1 Characterization of the Maximum Operating Frequency, Power Consumption, and Area

Three types of circuit topologies are implemented to analyze the total area, power consumption, and maximum operating frequency of each developed logic family. The topologies include 1) a chain of four inverters, 2) a universal gate (AND, OR, NAND, and NOR), and 3) a 4×4 bit array multiplier.

6.3.1.a Characterization of Logic Families Using a Chain of Four Inverters

The maximum operating frequency, power consumption (static, dynamic, and total), and area of a chain of inverters implemented with the CMOS, CML, DCML, LDCML, and DFCML logic families at 400 mV and 450 mV are listed in Tables 6.1 and 6.2, respectively. Each logic family is designed for minimum area and full voltage swing at the end of the fourth stage.

At all supply voltages and among all logic families, DCML operates at the highest frequency and CMOS at the lowest, where the maximum operating frequency of a

Table 6.2: Comparison of the performance, area, and power consumption of an inverter chain operating at a 450 mV supply voltage.

	CMOS	CML	DCML	LDCML	DFCML
f_{max} with 4 stages (MHz)	143	157	170	158	161
Area $(\mu m)^2$	2.88	10.56	4.56	4.85	3.0576
Static power (nW)	0.65	11797	0.22	0.15	0.14
Dynamic power (nW)	1634.35	1243	1077.8	1155.85	727.56
Total power (nW)	1635	13040	1078	1156	727.7

CMOS inverter chain at 400 mV and 450 mV is, respectively, 63 MHz and 143 MHz. Therefore, for a fair analysis and comparison of the power and robustness to noise and variation, the inverter chains implemented in all logic families are set to an operating frequency of 60 MHz at 400 mV and 140 MHz at 450 mV. Among all differential signal based logic families, DFCML occupies the smallest area, with an increase of 1.06× as compared to CMOS. Both LDCML and DFCML exhibit up to a 1.68× increase in area as compared to CMOS, while the inverter chain implemented with CML resulted in an increase in area of 3.7× that of CMOS. In addition, the results listed in Tables 6.1 and 6.2 indicate a significant change in the optimum performance-power point when the supply voltage is increased from 400 mV to 450 mV. The 50 mV increase in the supply voltage results in, on average, an f_{max} that improves by approximately 2× and a total power consumption that increases by 2.5×.

For both near-V_t operating voltages, the inverter chain implemented with DFCML dissipates the least total power with a consumption of 0.45× that of a CMOS inverter chain while providing a maximum operating frequency 1.13× that of CMOS. In comparison, at both near-V_t voltages, the DCML inverter chain consumes a total power

of 0.68× and improves the maximum operating frequency by at least 1.19× that of a CMOS inverter chain. The total power consumption is reduced to 0.08× with at least a 1.08× improvement in the maximum operating frequency when using DCML over CML for both supply voltages. The comparison of power and performance, therefore, varies with a reduction of 50 mV in the near-V_t supply voltage, as a greater drop in the maximum operating frequency is observed for CML (0.45×) and CMOS (0.44×) circuits as compared to DCML (0.68×) and LDCML (0.63×) circuits, where the CMOS inverter chain operating at 450 mV provides the baseline for comparison.

The transient behavior of a chain of four inverters is shown in Fig. 6.7. The LDCML family exhibits the maximum power consumption and area among the three DDSL families. Therefore, only the LDCML chain of inverters is evaluated through transient analysis and compared with both the CML and CMOS chain of inverters. An iso-area comparison is performed, where the area of the CML and CMOS chain of inverters are resized to match the area of LDCML chain of inverters (4.85 μm^2). The SPICE simulation is performed at a 450 mV supply voltage and 140 MHz frequency. The average power consumption per cycle of the CMOS, CML, and LDCML chain of inverters is, respectively, 1.408 μW, 7.3 μW, and 0.74 μW. Despite the same area, operating frequency, and supply voltage, the LDCML chain of inverters consumes significantly less average power.

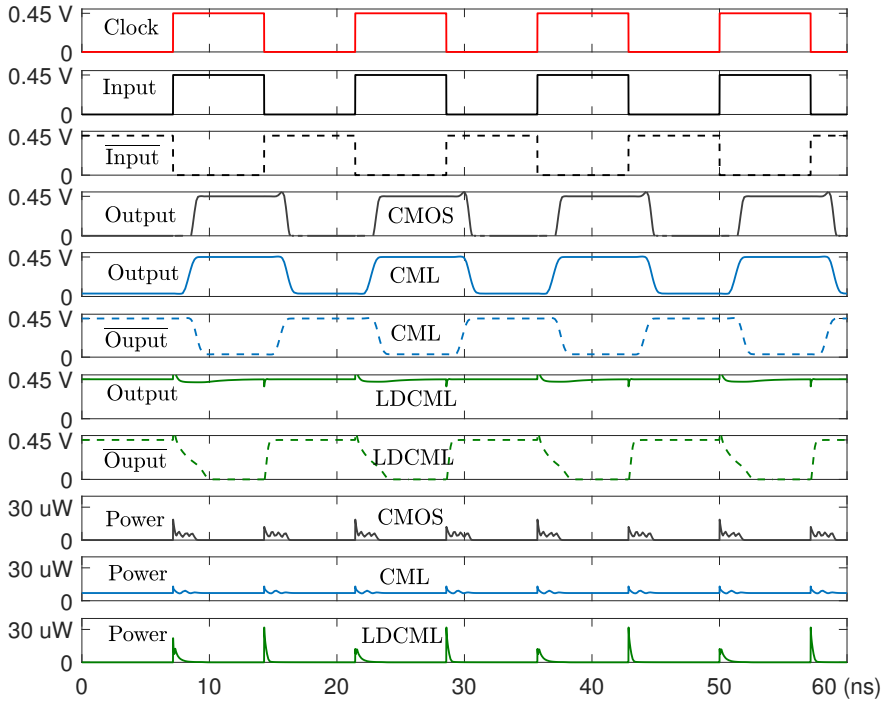


Fig. 6.7: Transient simulation for iso-area comparison of a CMOS, CML, and LDCML chain of four inverters operating at 450 mV and 140 MHz.

6.3.1.b Characterization of Logic Families Using Universal Gates

A characterization of the maximum operating frequency and active area is provided for universal gates implemented using all logic families at a near-V_t voltage of 450 mV, with results shown in Fig. 6.8. The DCML universal gate exhibits a maximum operating frequency of 280 MHz, which is an improvement in the maximum operating frequency of 1.83× and 1.56× that of, respectively, CMOS NAND/NOR gates and a CML universal gate. In addition, both DFCML and LDFCML exhibit at least a 1.63× improvement in the maximum operating frequency as compared to

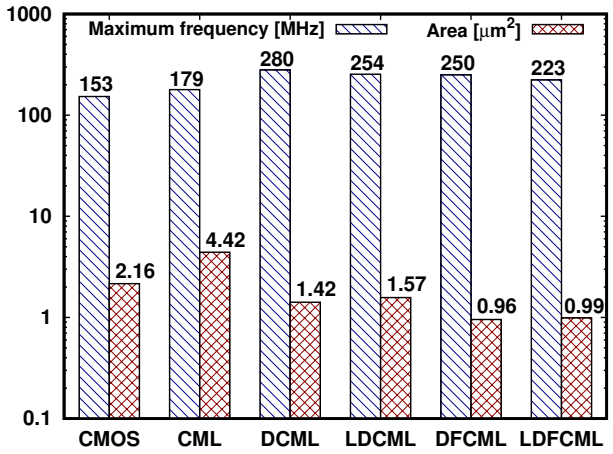


Fig. 6.8: Characterization of the maximum operating frequency and area of universal gates at 450 mV. CMOS NAND/NOR gates were implemented for comparison.

CMOS NANDs and NORs. Among all logic families, the DFCML universal gates occupy the smallest area, where the area overhead of CMOS NAND/NOR and CML universal gates are, respectively, 2.26× and 4.63× greater than the DFCML universal gates. The opposite behavior is observed for the inverter chain, where the CMOS implementation occupied the least area. Unlike standard CMOS circuits, the pull up network and voltage headroom are not significantly impacted when switching from an inverter to a NAND or NOR gate when using DFCML. As an example, the implementation of two-input DCML NAND and NOR gates, as compared to an inverter, require an additional series connected NMOS transistor in one of the two discharging paths (see Fig. 6.5 (a)), while the pull up network remains unchanged. In contrast, the CMOS NAND or NOR gate, as compared to an inverter, requires an additional transistor in both the pull up and pull-down network for each additional input.

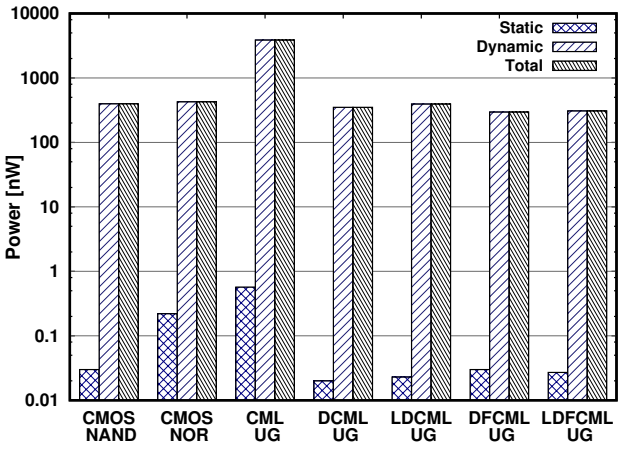


Fig. 6.9: Characterization of the power consumption of universal gates (UG) at a supply voltage of 450 mV.

The universal gates implemented using all logic families are characterized for static, dynamic, and total power consumption at a near-V_t voltage of 450 mV, with the results provided in Fig. 6.9. The static power of the CMOS NOR and CML universal gate is, respectively, 11× and 28.5× greater than that of a DCML universal gate. In addition, the use of DCML and DFCML universal gates reduces the total power consumption to, respectively, 0.82× and 0.7× that of a CMOS NOR gate. The total power consumption of CML universal gates is at least 9.8× greater than any of the DCML universal gates. The analysis of the static, dynamic, and total power consumption of the universal gates implemented with all logic families indicates similar behavior as the four-stage inverter chain.

6.3.1.c Characterization of Logic Families with a 4×4 Bit Array Multiplier

The maximum operating frequency, total area, and total power consumption of a 4×4 bit array multiplier implemented in each logic family is characterized at 450 mV. The results are normalized to the respective values of a CMOS multiplier and are shown in Fig. 6.10. The total area, maximum operating frequency, static, dynamic, and total power consumption of a 4×4 array multiplier implemented in CMOS are, respectively, 150.904 μm^2, 20 MHz, 0.01185 μW, 3.60315 μW, and 3.615 μW. Similar to the inverter chain and universal gates, multipliers implemented using DCML, LDCML, and DFCML exhibit improved performance as compared to CMOS and CML. The maximum operating frequency of the DCML families is at least 1.4× and 1.12× greater as compared to, respectively, CMOS and CML multipliers. The total area of the DCML multipliers is 1.5× and 0.57× the area of, respectively, the CMOS and CML multipliers. The total power consumption of LDCML and DFCML multipliers, which is approximately equal, is 0.9× and 0.009× the total power of, respectively, a CMOS and CML multiplier. However, the total power of a DCML multiplier is 1.48× the total power of a CMOS multiplier. The increase in the area and power consumption of a DCML multiplier is a result of the added buffers used to restore the full voltage swing of the signal at the output.

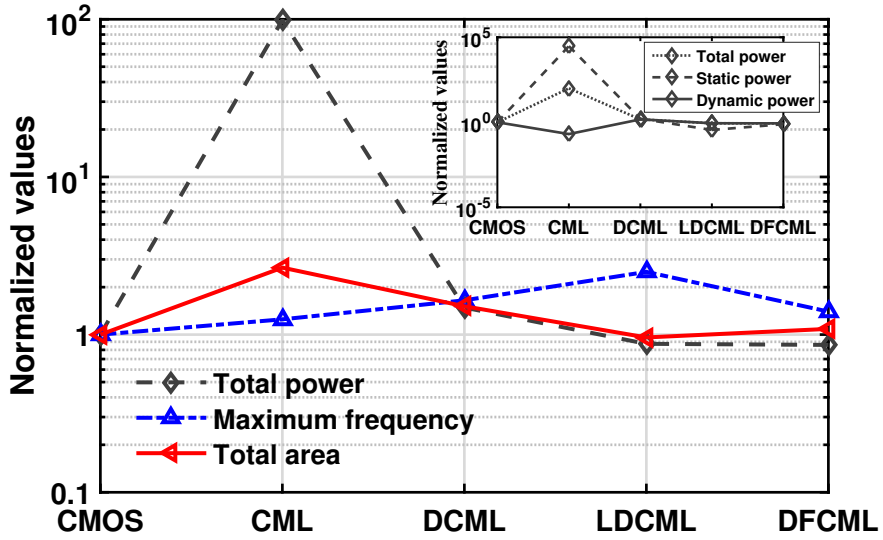


Fig. 6.10: Power, area, and maximum operating frequency of a 4×4 bit array multiplier implemented in all logic families.

6.3.2 Characterization of Static, Dynamic, and Total Power Consumption for Different Activity Factors

A chain of four inverters is implemented in each logic family to analyze the effect of activity factor on the power consumption. The dynamic power consumption is reduced with lower activity factor for CMOS circuits. However, for DCML, the dynamic power does not change with activity factor as the output terminals of the DCML circuit are always charged and discharged during, respectively, the pre-charge and evaluate phases. The power consumption is, therefore, relatively constant regardless of input activity. CML circuits exhibit minor changes in power consumption with

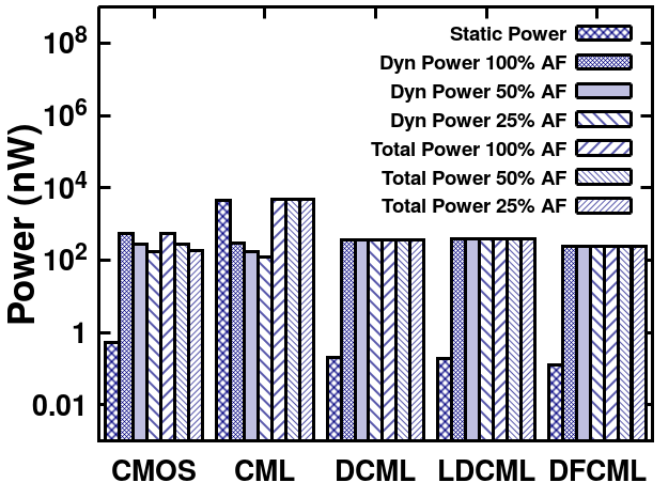


Fig. 6.11: Power consumption of a chain of four inverters implemented with all logic families for activity factors of 25%, 50%, and 100% and for a supply voltage of 400 mV.

a change in activity factor as the dominant component is the static power dissipation. The power consumption for activity factors of 25%, 50%, and 100% at a supply voltage of 400 mV is shown in Fig. 6.11. The total power consumption (dynamic and static) is calculated as the average power per cycle over 60 cycles at each activity factor. For all logic families, the operating frequency of the inverter chain is set to 60 MHz.

Among all logic families, the DCML, LDCML, and DFCML inverter chains consume the minimum static power, where the static power consumption of the three families is approximately 0.05% of the total power consumption. The majority of the total power consumed by CML is static since, regardless of input activity or switching events, there is a constant static current through the tail transistor. The majority

of the total power consumption of the DCML families, however, is dynamic as there is a periodic charging and discharging of the output nodes. The use of DCML over CML results in additional power savings as the static current path through the tail transistor is no longer present.

6.3.3 Characterization of Energy-Delay Product (EDP)

An analysis of the energy-delay product (EDP) for all logic families is performed using a four-stage inverter chain. The supply voltage is set to 450 mV and the operating frequency to 60 MHz. The two middle stages are considered when determining the EDP to more accurately represent the input and output capacitive loads and intrinsic delay of the inverters. The characterization of EDP for all logic families is provided through the results shown in Fig. 6.12, where all values are normalized to the EDP of a CMOS inverter chain (0.02×10^{-20} J·s). All DCML families exhibit improved EDP as compared to both the CMOS and CML inverter chains. LDCML exhibited the minimum EDP, which is approximately 0.07× and 0.004× the EDP of, respectively, CMOS and CML.

6.3.4 Characterization of Circuit Robustness to Noise

One of the key challenges of NTC is reduced robustness to external noise due to a scaled supply voltage. The robustness of the dynamic current mode logic families is,

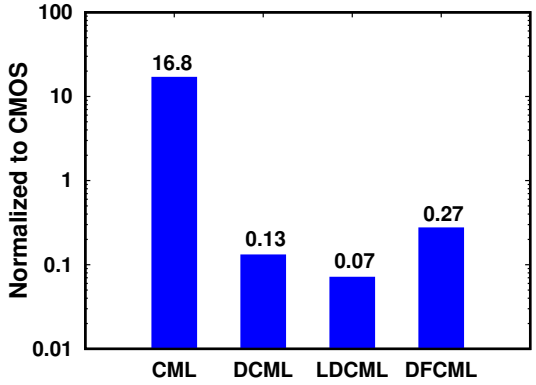


Fig. 6.12: Energy delay product of a chain of inverters operating at a supply voltage of 450 mV. All values are normalized to the EDP of a CMOS inverter chain (0.02×10^{-20} J·s).

therefore, analyzed and compared with CML and CMOS logic circuits. The robustness is evaluated through 1) an analysis of the noise margin, where a novel method to identify the noise margins of differential logic circuits is introduced, and 2) an analysis of the sensitivity of the logic families to threshold voltage variation. The effect of process, voltage, and temperature (PVT) variations on the propagation delay and total power consumption is also analyzed. External sources of noise, including power supply noise and crosstalk due to inductive, capacitive, and conductive coupling, are considered through noise margin analysis. In addition, both global and local process variation and mismatch variation is characterized through Monte-Carlo simulation.

6.3.4.a Analysis of Noise Margins

The analysis of the worst case variation of the noise margin is performed for a single inverter in all logic families at a supply voltage of 400 mV and with a set

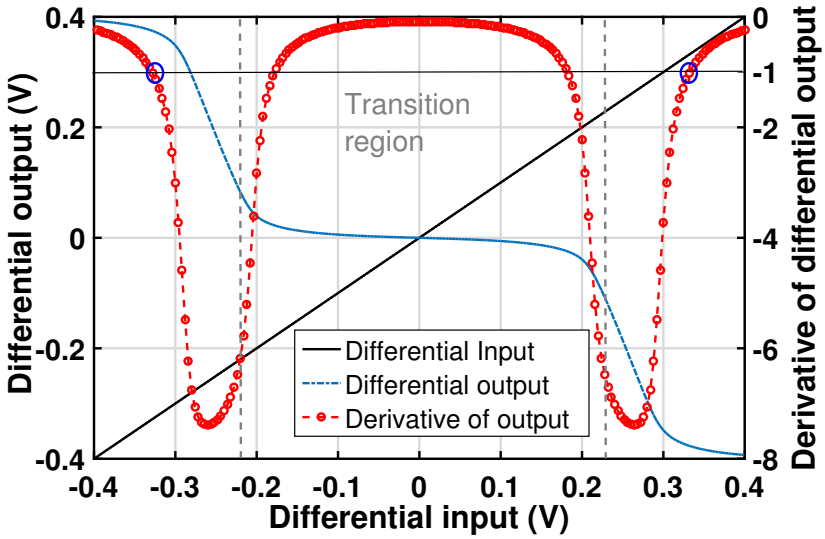


Fig. 6.13: VTC characterization to determine the noise margins of a DCML inverter at a supply voltage of 400 mV.

frequency of 60 MHz. For the CMOS inverter, a conventional voltage transfer curve (VTC) is analyzed [84]. Both CML and DCML inverters are, however, evaluated by differential inputs and outputs. The inverter VTC curve, therefore, represents the input and output as the difference between, respectively, the two input and two output signals. The method of measuring noise margins for differential logic families is illustrated in Fig. 6.13, where the solid line and dotted line represent, respectively, the differential input and differential output. The noise margins are determined for values when the derivative of the differential output voltage is -1. For differential logic circuits, the noise margins are evaluated using (6.9), where x is the differential input voltage between In and $\overline{In}$, y is the differential output voltage between Out and $\overline{Out}$, and N is the value of x for which $|\frac{dy}{dx}|$ is equal to 1. The two points of the

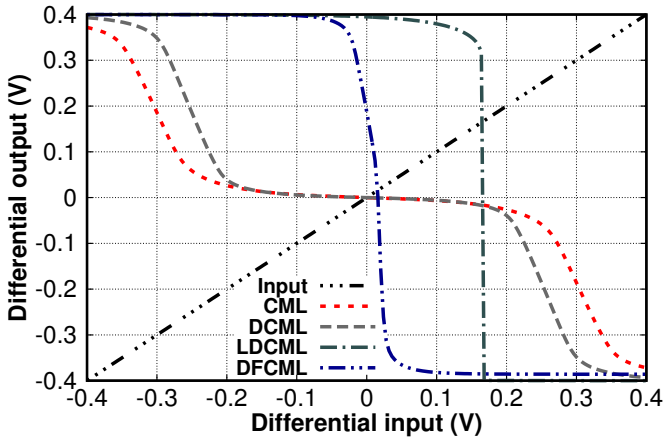


Fig. 6.14: VTC of all differential inverters at a 400 mV supply voltage.

derivative curve that are equal to -1 and fall within the transition region of the VTC are ignored. The transition points A and B shown in Fig. 6.2 are determined using the proposed technique.

$$NM = V_{dd} - N, \tag{6.9}$$

$$N = \text{value of } x \text{ for which } |\frac{dy}{dx}| = 1,$$

$$x = V_{In} - V_{\overline{In}}, \tag{6.10}$$

$$y = V_{Out} - V_{\overline{Out}} \tag{6.11}$$

The VTCs of the differential logic families are shown in Fig. 6.14 for a supply voltage of 400 mV. The LDCML and DFCML inverters exhibit a sharp transition from logic high to logic low. The CML and DCML inverters include an undefined region (logically indiscernible between logic high and logic low) for differential input

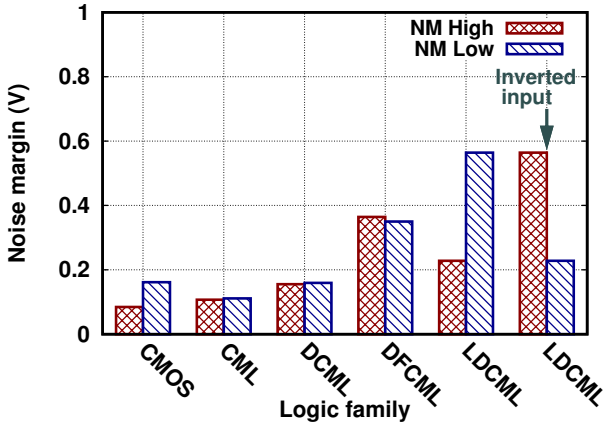


Fig. 6.15: Comparison of the noise margins of the logic families at a near-threshold voltage of 400 mV.

signals between -0.2 V and 0.2 V, where the differential output remains close to 0 V. The result is a reduction in the noise margins of the CML and DCML circuits as compared to the LDCML circuits. The noise margins of all logic families are shown in Fig. 6.15. The NM_H is always a positive voltage while the NM_L is always negative for the differential inverters. However, for ease of analysis and comparison, the absolute value of the NM_L is provided in Fig. 6.15 for all differential inverters.

$$V_{Tn} = V_{Tp} = V_T \tag{6.12}$$

$$k_R = \frac{k_n}{k_p} = 1 \tag{6.13}$$

$$k_n = \mu_n C_{ox} (\frac{W}{L})_n \tag{6.14}$$

$$k_p = \mu_p C_{ox} (\frac{W}{L})_p \tag{6.15}$$

The noise margins of the LDCML and DFCML inverter are larger than the DCML, CML, and CMOS inverters, where the DFCML exhibits mean noise margins of 350 mV. The mean noise margin of LDCML and DFCML inverters is at least 2.25× greater than the mean noise margin of a CMOS inverter. At near-V_t, the NM_H and NM_L of a LDCML inverter are, respectively, 228 mV and 564 mV when an initial differential voltage of 400 mV is applied at the input. Applying the opposite differential input of -400 mV, results in a swap of the NM_H and NM_L to, respectively, 564 mV and 228 mV (inverted input in Fig. 6.15). The NM_H and NM_L of a CMOS inverter are, respectively, 161 mV and 90 mV. The symmetric VTC behavior of a CMOS inverter is not maintained at near-V_t to optimize the operating frequency of the gate, while minimizing the increase in area and power consumption. Equal noise margins are achieved when satisfying (6.12) and (6.13), where μ_n is the electron surface mobility, μ_p is the hole surface mobility, and W the width and L the length of the transistor [84, 274]. The threshold voltages and, therefore, noise margins, also vary with changes in PVT and substrate bias. For a 130 nm technology, the variations in PVT and non-zero substrate bias result in variations in the threshold voltage of up to 100 mV from the nominal value of 350 mV [151]. Disparate NMOS and PMOS threshold voltages and drive strengths, therefore, lead to differences in the values of the NM_H and NM_L.

Due to the PMOS keeper transistors, the input-output characteristics of LDCML circuits are sensitive to the polarity and magnitude of the initial input voltages, which

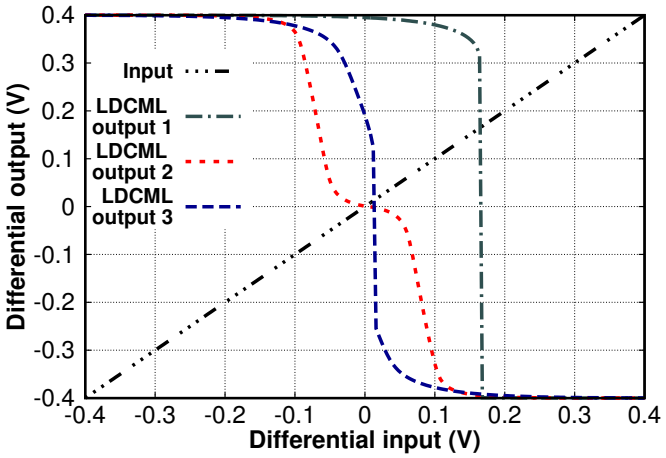


Fig. 6.16: VTC of a LDCML inverter at a 400 mV supply voltage. An initial -400 mV differential input results in LDCML output 1, whereas an initial 0 V differential input results in LDCML output 2. LDCML output 3 exhibits a symmetric VTC.

results in disparate high and low noise margins. Inputs with equal applied voltages result in a similar voltage at the two output nodes unless one of the inputs changes with respect to the other. The VTC for a differential input voltage of 0 V when both input nodes are initially equal is shown in Fig. 6.16 as LDCML output 2. However, for the analysis of the noise margins, a non-zero initial differential input voltage is applied, corresponding to LDCML output 1 in Fig. 6.16, as equal voltages at the inputs results in the inverter not functioning as intended. The asymmetric behavior of the noise margins is not present when the size of the NMOS transistors is increased by 20×, resulting in a symmetric response as shown by LDCML output 3 in Fig. 6.16. The large NMOS transistors sink current from the output node with a moderate gate voltage, even as the holding PMOS keeper is supplying additional charge. As a result, the NM_L and NM_H are equal regardless of the input polarities. However,

the increased size of the NMOS transistors results in a larger capacitive load and, therefore, higher power consumption, while also negating the charge restoration at the output nodes provided by the PMOS keepers. In addition, the asymmetric behavior of the VTC only occurs when the differential input voltage is between -0.15 V and 0.15 V, which are voltages not within the permitted operating point of the circuit. The asymmetric behavior of the noise margins of the LDCML inverter, therefore, does not effect the circuit operation, while providing additional power savings and performance benefits as compared to the symmetric implementation. Depending on the application and specific circuit biasing conditions, the differential inputs are set to either enhance the NM_H or NM_L of a LDCML circuit.

6.3.4.b Characterization of Logic Families Under Threshold Voltage Variation

Near-threshold circuits suffer from inherent sensitivity to variation as a small change in threshold or supply voltage significantly impacts the optimum operating point. In addition, advanced technology nodes are more susceptible to the effects of process variation including fluctuations in the doping profile and the oxide thickness [84, 274], which significantly impact the threshold voltage and, therefore, the performance and power consumption of the circuits.

The threshold voltage is dependent on biasing conditions and on process parameters including silicon and oxide materials, physical dimensions, and doping concentrations [84, 274]. The non-zero substrate bias threshold voltage V_T of a MOSFET is given by (6.16), and includes the zero substrate-bias threshold voltage V_{T0} given by (6.17), where Φ_{GC} is the difference in the work function between the gate and channel, Q_{B0} is the charge density of the depletion region at a zero substrate-bias, Q_{ox} is the charge density at the interface between the oxide and silicon, ϕ is the fermi potential, and γ is the substrate bias (body effect) coefficient. The substrate bias coefficient given by (6.18) is dependent on the electron charge q, the doping concentration N_A, the dielectric constant of silicon ϵ_{si}, and the gate oxide capacitance per unit area C_{ox} [84, 274].

$$V_T = V_{T0} + \gamma(\sqrt{|-2\phi + V_{SB}|} + \sqrt{|2\phi|}) \tag{6.16}$$

$$V_{T0} = \Phi_{GC} - 2\phi - \frac{Q_{B0}}{C_{ox}} - \frac{Q_{ox}}{C_{ox}} \tag{6.17}$$

$$\gamma = \frac{\sqrt{2qN_A\epsilon_{si}}}{\sqrt{C_{ox}}} \tag{6.18}$$

Changes in the threshold voltage due to PVT variation and/or substrate body bias modify the drive current of the transistors. For a given gate to source voltage V_{GS} and drain to source voltage V_{DS}, a positive source to body voltage V_{SB} for an NMOS transistor increases the width of the depletion layer and reduces the drive current. The maximum operating frequency of the transistor is, therefore, reduced. Variations

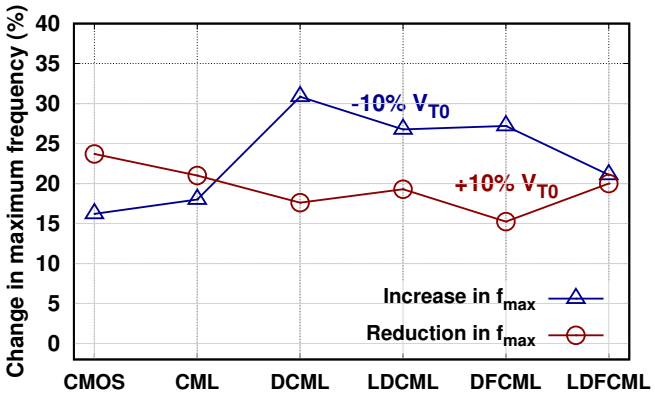


Fig. 6.17: Characterization of the maximum operating frequency f_{max} with $\pm 10\%$ variation in the threshold voltage of universal gates implemented in all logic families, where the maximum operating frequency at a supply voltage of 450 mV for all logic families is provided in Fig. 6.8.

in process parameters impact both V_{T0} and V_T as indicated through evaluation of (6.16) and (6.17). For example, a larger Q_{B0}/C_{ox} ratio or Q_{ox}/C_{ox} ratio results in a reduction in the zero bias threshold voltage V_{T0}. In this case, for a given V_{GS} and V_{DS}, the reduced V_{T0} results in an increase in the drive current of the transistors and, therefore, an increase in the maximum operating frequency.

The variation of the threshold voltage is analyzed to determine the effect on the maximum operating frequency of the universal gates implemented with all logic families. The percentage change in the maximum operating frequency of all logic families for a $\pm 10\%$ variation in the nominal zero-bias threshold voltage V_{T0} is shown in Fig. 6.17. The variation in V_{T0} resulted in up to an approximately 5% fluctuation in the non-zero substrate-bias threshold voltage V_T. The analysis of threshold voltage variation is performed at a near-V_t voltage of 450 mV for all logic families, and the

Table 6.3: Characterization of the propagation delay and total power consumption of a 4×4 bit array multiplier implemented with all logic families in three process corners at a supply voltage of 450 mV and temperature of 27°C. All delay and power results are normalized to, respectively, the delay and power of a CMOS multiplier operating in the *tt* corner and a temperature of 27°C.

Logic families	**Propagation delay**			**Power consumption**		
	tt	***ff***	***ss***	***tt***	***ff***	***ss***
CMOS	1	0.637	1.608	1	0.997	0.936
CML	0.589	0.387	0.897	98.5	131	74.5
DCML	0.157	0.082	0.318	1.479	1.423	1.615
LDCML	0.191	0.095	0.396	0.951	0.962	0.982
DFCML	0.206	0.085	0.634	0.952	0.943	1.0003

results are compared with CMOS and CML gates.

The results indicate that there is a larger increase in the maximum operating frequency of the DCML families as compared to CMOS and CML circuits with a reduction in the threshold voltage, as shown in Fig. 6.17. With a 10% reduction in V_{T0}, the DCML universal gate exhibits a 30.85% increase in the operating frequency, the largest amongst all logic families. In contrast, CMOS and CML circuits exhibit an increase in the maximum operating frequency of no more than 16.2% and 18%, respectively. Therefore, a 10% reduction in the threshold voltage results in up to a 1.7× increase in the frequency of a DCML circuit as compared to a CML or CMOS circuit. With a 10% increase in V_{T0}, DFCML exhibits the smallest reduction in the operating frequency of 15.24% as compared to reductions of 23.7% and 21% for, respectively, CMOS and CML.

Table 6.4: Characterization of propagation delay and total power consumption of a 4×4 bit array multiplier implemented with all logic families for supply voltages between 0.35 V and 0.5 V. All delay and power results are normalized to, respectively, the delay (14.3 ns) and power consumption (3.615 μW) of a CMOS multiplier operating at a supply voltage of 0.45 V with the *tt* corner and a temperature of 27°C.

Logic families	**Propagation delay**				**Power consumption**			
	0.35 V	**0.4 V**	**0.45 V**	**0.5 V**	**0.35 V**	**0.4 V**	**0.45 V**	**0.5 V**
CMOS	5.406	2.189	1	0.531	0.428	0.674	1	1.265
CML	3.182	1.270	0.589	0.318	12.704	38.204	98.481	217
DCML	1.087	0.391	0.157	0.076	0.843	1.130	1.479	1.872
LDCML	1.770	0.585	0.191	0.081	0.477	0.701	0.952	1.225
DFCML	1.307	0.475	0.206	0.106	0.415	0.570	0.951	1.206

6.3.4.c Characterization of Logic Families Under Process, Voltage, and Temperature Variation

A characterization of process (*tt*, *ff*, and *ss*), voltage (from 0.35 V to 0.5 V), and temperature (from 27°C to 100°C) variations on a 4×4 bit array multiplier is performed. All logic families are analyzed at 20 MHz for iso-performance analysis as the CMOS multiplier exhibits a maximum operating frequency of 20 MHz at 450 mV. The change in the propagation delay and total power consumption of a 4×4 bit array multiplier due to variations in the process, voltage, and temperature are provided in, respectively, Tables 6.3, 6.4, and 6.5.

The propagation delay and total power consumption are characterized for the *tt*, *ff*, and *ss* corners as listed in Table 6.3 with the supply voltage set to 450 mV and the temperature to 27°C. All delay and power results are normalized to the delay and power of a CMOS multiplier operating in the *tt* corner, which are 14.3 ns and 3.615 μW, respectively. For the *ss* corner, the delay of the CMOS, CML, DCML,

Table 6.5: Characterization of propagation delay and total power consumption of a 4×4 bit array multiplier implemented with all logic families at 27°C, 50°C, and 100°C, a supply voltage of 450 mV, and a process corner of *tt*. All delay and power results are normalized to, respectively, the delay (14.3 ns) and power consumption (3.615 μW) of a CMOS multiplier operating in the *tt* corner and at a temperature of 27°C.

Logic families	**Propagation delay**			**Power consumption**		
	27°C	**50°C**	**100°C**	**27°C**	**50°C**	**100°C**
CMOS	1	0.790	0.536	1	1.041	1.182
CML	0.589	0.477	0.338	98.5	122.8	178.9
DCML	0.157	0.120	0.080	1.479	1.559	1.824
LDCML	0.191	0.159	0.133	0.951	1.020	1.227
DFCML	0.206	0.163	0.114	0.952	1.021	1.201

LDCML, and DFCML multipliers is, respectively, 1.6×, 0.9×, 0.3×, 0.4×, and 0.6× the delay of a CMOS multiplier operating in the *tt* corner. All DCML multipliers exhibit greater variation among the three process corners (*tt*, *ss*, and *ff*) due to a larger number of transistors. For example, the delay of a CMOS inverter is effected by only two transistors, whereas the delay of a DCML inverter is impacted by four transistors. The total power consumption of the CMOS, CML, DCML, LDCML, and DFCML multipliers operating at 20 MHz in the *ff* corner is, respectively, 0.997×, 131×, 1.42×, 0.96×, and 0.94× that of the total power consumption of a CMOS multiplier operating in the *tt* corner.

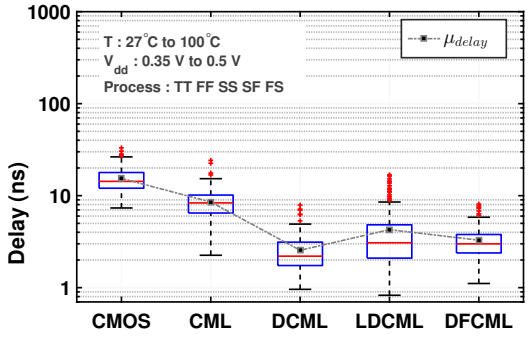
Despite the lower maximum operating frequency of a DCML multiplier as compared to a LDCML multiplier, the 4×4 DCML multiplier incurs the shortest propagation delay. The shorter delay is due to: 1) as the number of logical stages implemented with DCML increases, the ability to provide full swing at the output node of the last

stage decreases due to charge loss from the preceding logical stages during the evaluate phase, 2) although the propagation delay is calculated as the time difference between when an input and output signal change by 50%, full-swing at the output of DCML is not guaranteed, and 3) the condition set to determine the maximum operating frequency requires the full swing within 70% of the active pulse width of the output. As a result, DCML and DFCML do not exhibit a higher f_{max} due to charge leakage at the output nodes.

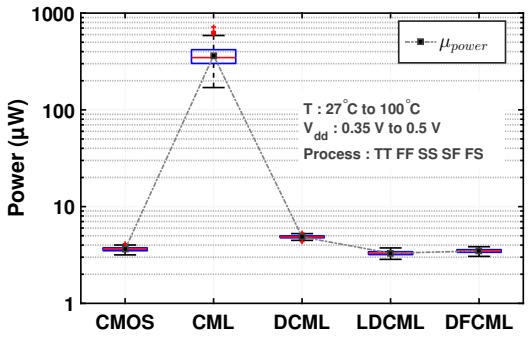
The variation in the propagation delay and total power consumption of the multipliers implemented with CML, DCML, LDCML, and DFCML for supply voltages ranging from sub-V_t (0.35 V) to near-V_t (0.5 V) is listed in Table. 6.4. The delay and power consumption of all multipliers operating at 27°C and with the *tt* process corner is determined and normalized to, respectively, the delay and power of a CMOS multiplier operating at a supply voltage of 0.45 V. The delay (power consumption) of a CMOS multiplier is 77.3 ns (1.55 μW), 31.3 ns (2.44 μW), 14.3 ns (3.615 μW), and 7.59 ns (4.58 μW) at a supply voltage of, respectively, 0.35 V, 0.4 V, 0.45 V, and 0.5 V. All DCML multipliers exhibit up to a 3× increase in the normalized total power consumption as the supply voltage is scaled from a sub-V_t voltage of 0.35 V to a near-V_t voltage of 0.5 V, much less than the 17× increase seen by the CML multiplier. In addition, multipliers implemented with all logic families exhibit at least a 10× increase in the normalized propagation delay as the supply voltage is scaled

from a near-V_t voltage of 0.5 V to a sub-V_t voltage of 0.35 V. The normalized propagation delay and total power consumption of the LDCML and DFCML multipliers are smaller as compared to a CMOS multiplier for all supply voltages between 0.35 V and 0.5 V.

The effect of temperature variation (from 27°C to 100°C) on the propagation delay and total power consumption of the logic families is analyzed, with results listed in Table 6.5. All delay and power results are normalized to, respectively, the delay and power consumption of a CMOS multiplier operating at 27°C, in the *tt* corner, and at 450 mV. The normalized delay and power varies less than 1.82× among all logic families when the temperature is increased from 27°C to 100°C. From all logic families, LDCML exhibits the greatest reduction in the normalized delay (0.7×) when the temperature increases to 100°C from 27°C. The delay of the multiplier at a near-V_t voltage of 450 mV decreases with increasing temperature due to temperature effect inversion, while at super-V_t, higher temperatures result in larger delays [48, 49, 179]. All differential logic families exhibit higher variation in normalized power consumption as compared to a CMOS multiplier, where CML exhibits the greatest variation (1.82×) when the temperature is increased to 100°C from 27°C.



(a)



(b)

Fig. 6.18: Monte Carlo simulation with 200 runs of a 4×4 bit array multiplier to characterize a) delay and b) power.

6.3.5 Analysis of Local and Global Statistical Variation

Process and mismatch are characterized using Monte-Carlo simulation with 200 runs to statistically analyze the variation in the propagation delay and total power consumption of the multipliers. The Latin-hypercube sampling method is applied, which allows for accurate process coverage with fewer runs [277]. Die-to-die and within-die variation are considered to characterize, respectively, global and local variation [278]. Five process corners (tt, ff, ss, sf, fs) and a wide temperature range (27°C to 100°C) are used for Monte-Carlo analysis of the logic families across supply

voltages that fall between sub-V_t (0.35 V) and near-V_t (0.5 V).

The variation in delay and power is represented with a box-and-whisker plot as shown in, respectively, Figs. 6.18(a) and 6.18(b), to accurately analyze the population distribution. The mean value μ of both the delay and power is also provided in the figures for each logic family. The delay of all DCML multipliers is less than that of a CMOS multiplier. The mean delay μ_{delay} of the CML, DCML, LDCML, and DFCML multipliers is, respectively, 0.55× (8.53 ns), 0.17× (2.55 ns), 0.28× (4.27 ns), and 0.21× (3.28 ns) the μ_{delay} of a CMOS multiplier (15.4 ns). The μ_{power} of the CML, DCML, LDCML, and DFCML multipliers is, respectively, 100× (365 µW), 1.34× (4.86 µW), 0.91× (3.30 µW), and 0.96× (3.48 µW) the μ_{power} of a CMOS multiplier (3.62 µW). In addition, the μ_{power} (μ_{delay}) of the DCML, LDCML, and DFCML multipliers is, respectively, 0.013× (0.3×), 0.009× (0.5×), and 0.009× (0.4×) the μ_{power} (μ_{delay}) of a CML multiplier. Therefore, both the delay and total power consumption is reduced with LDCML and DFCML multipliers as compared to CMOS and CML multipliers.

6.3.6 Comparison Between Near- and Super-V_t Operation

A comparison between circuits operating at a near-V_t voltage and at a super-V_t voltage is provided through results listed in Table 6.6, which includes data from prior research in NTC [18, 131, 144]. As the supply voltage is scaled from 1.2 V to 0.45 V, the reduction in energy/cycle, the reduction in f_{max}, the change in NM_{avg}/V_{DD}, and

Table 6.6: Figures of merit at near-threshold voltage of 0.45 V (except NM_{avg}/V_{DD}, which is analyzed at 0.4 V). All results are normalized to CMOS logic operating at 1.2 V. Results for CML are from simulations by the authors. The topology of the CML universal gate described in [131] is implemented and characterized.

Logic Family	**Reduction in Energy/Cycle**	**Reduction in f_{max}**	**Increase in NM_{avg}/V_{DD}**	**Area**
CMOS [18, 144]	≈10	≈14	Not given	Not given
CMOS	8.5	19.4	1.06	1
CML [131]	1.01	16.6	0.94	1.75
DCML	13.23	10.6	2.72	0.54
LDCML	10.9	11.7	3.42	0.63
DFCML	12.6	11.9	3.08	0.42

the change in area are characterized using universal gates. In addition, the effects of voltage scaling (1.2 V to 0.4 V) on noise margins is analyzed using a chain of inverters. For CMOS logic, the scaling of the supply voltage from 1.2 V to 0.45 V reduces the energy/cycle to 8.5× at a cost of a 19.4× reduction in f_{max}, while prior research reported that scaling from super-V_t to near-V_t reduces the energy/cycle by approximately 10× and reduces the f_{max} by approximately 14× [18, 144]. Therefore, the DCML, LDCML, and DFCML circuits consume less energy/cycle with a smaller reduction in f_{max}, which indicates that dynamic current mode logic families are more suitable for near-V_t operation over CML and CMOS. In addition, the NM_{avg}/V_{DD} remains unchanged for CMOS when the supply voltage is scaled from 1.2 V to 0.4 V, while dynamic current mode logic families exhibit at least a 2.72× increase in NM_{avg}/V_{DD}.

6.3.7 Summary of Results

Simulated results are summarized in Table 6.7 for a supply voltage of 450 mV, where an inverter chain, universal gates, multi-level logical functions, and a 4×4 bit array multiplier are used to characterize the DCML families for area, power, operating frequency, noise margin, and EDP. The characteristics of all logic families are normalized to the respective results of CMOS. The dynamic current mode logic families are superior to CMOS and CML in total power consumption, maximum operating frequency, noise margin, EDP, and propagation delay. The DCML and DFCML families provide benefit for circuits with a fewer number of stages. However, LDCML is beneficial for any circuit size including a 4×4 bit array multiplier, as the total power, area, and maximum operating frequency are improved. The area of the LDCML chain of inverters is larger than an inverter chain implemented with CMOS, however, the opposite is observed for more complex logical gates including universal gates and a 4×4 bit array multiplier, where the CMOS implementation requires greater area. The improvement in total power consumption and maximum operating frequency is, therefore, scalable to larger circuit blocks when dynamic current mode logic families are used.

Table 6.7: Summary of results for all logic families normalized to CMOS for a supply voltage of 450 mV.

		CMOS (Baseline)	CML	DCML	LDCML	DFCML
Chain of inverters	Total power (at 60 MHz)	1 (1.635 µW)	8	0.66	0.7	0.45
	Maximum operating frequency	1 (143 MHz)	1.09	1.19	1.15	1.13
	Total area	1 (2.88 μm^2)	3.7	1.58	1.68	1.06
	Noise margin	1 (123 mV)	0.89	1.28	3.22	2.9
	EDP	1 (0.02×10^{-20} J·s)	16.84	0.13	0.071	0.27
Universal gates	Total power (at 140 MHz)	1 (0.427 µW)	9.07	0.82	0.92	0.7
	Maximum operating frequency	1 (153 MHz)	1.17	1.83	1.66	1.63
	Total area	1 (2.16 μm^2)	2.05	0.66	0.73	0.44
4×4 array multiplier	Total power (at 20 MHz)	1 (3.62 µW)	98	1.48	0.95	0.95
	Maximum operating frequency	1 (20 MHz)	1.25	1.65	2.5	1.4
	Total area	1 (151 μm^2)	2.7	1.5	0.96	1.09
	Propagation delay	1 (14.3 ns)	0.59	0.16	0.19	0.21

6.4 Interfacing of Ultra-Low Voltage Circuits

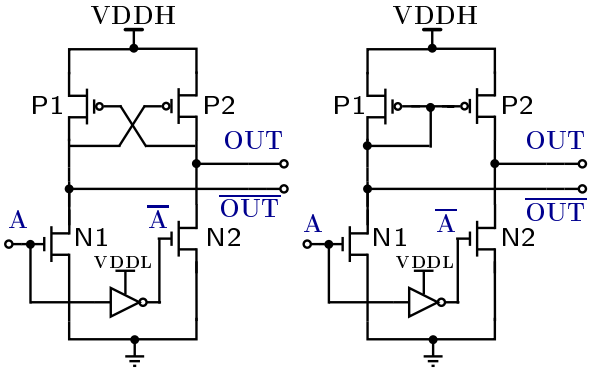
The supply voltage for optimal power-performance point varies across logic, memory, input/output, and peripheral analog and mixed-signal circuit blocks [18]. For example, the optimum energy-delay point for a memory circuit requires a higher V_{min} as compared to the V_{min} for the optimum point of core logic [18]. In addition, systems on chip (SoC) and multi-core systems contain multiple power domains for improved energy efficiency while meeting targetted performance requirements [18]. Novel interface circuits are, therefore, required when data is transferred between near-threshold circuits and circuits operating on a separate power domain as interfacing techniques designed for nominal voltage operation are not optimal at near-threshold. In this research, the interfacing of circuits operating at near-threshold supply voltage with the circuits operating at super-threshold and I/O voltage level is developed.

Techniques to interface between voltage domains of an IC implement either input/output (I/O) circuits or level shifters. The primary advantages of a bidirectional I/O circuit over a level shifter are the inclusion of an input receiver mode and a tristate output driver mode. Prior work on bidirectional I/O circuits examined the interface between a nominal supply voltage for the IC and a separate I/O voltage with integrated level shifters [279, 280]. However, the interface (I/O circuit) between near- and nominal-threshold circuits is rarely addressed.

Bidirectional I/O circuits with integrated level shifters are proposed in prior work for nominal supply voltages, where cross coupled level shifters are used for low power consumption [279, 280]. However, cross coupled level shifters are not optimal when supply voltages are set close to or below the transistor threshold voltage. For example, minimum input voltages of 0.55 V and 0.625 V are required to up-convert to output voltages of, respectively, 1.2 V and 3.3 V. In addition, the overall delay and power consumption of an I/O circuit increases for a longer level shifter delay.

In this research, a novel bidirectional I/O circuit that implements a combination of integrated level shifters is discussed. In addition, standard implementations of the level shifters are evaluated for comparison with the proposed configurations of the I/O circuit. Prior work has explored level shifter topologies for ultra low power and low voltage operation. Level shifters based on two types of circuit topologies; 1) cross-coupled [281], and 2) current mirror [282], are implemented as shown in Fig. 6.19. At

a near-threshold supply voltage, the primary drawbacks of cross coupled based level shifters are: 1) a very weak pull down network, even if the NMOS transistors are significantly larger than the PMOS transistors, 2) the inability to change the set values at the outputs OUT and $\overline{OUT}$ as the NMOS transistors do not drain enough current to overcome the supplied current from the pull-up keeper at the lower supply voltages, and 3) increased delay. The primary drawback of current mirror based level shifters is an increased power consumption due to a static current path. In this research, a single ended level shifter is developed to mitigate the drawbacks of conventional cross coupled and current mirror level shifters. The proposed single ended level shifter and bidirectional I/O circuits are discussed in, respectively, Section 6.5 and Section 6.6. The simulation setup and characterization results are provided in Section 6.7.



(a) (b)

Fig. 6.19: Conventional level shifter topologies. a) Cross coupled, and b) current mirror.

6.5 Single Ended Level Shifter for NTC

A single ended input topology of the level shifter is proposed for near-threshold operation. The proposed level shifter is shown in Fig. 6.20, where input A operates at a near-threshold supply voltage. The single ended level shifter is compared with standard implementations of the cross coupled and current mirror based level shifters. A comparison of the conversion range, total power consumption, and propagation delay is performed. In addition, the proposed level shifter is compared with state-of-the-art low voltage level shifters. The operation of the single ended level shifter is as follows: Assume a nominal voltage of $V_{DD,H}$ and near-threshold voltage of $V_{DD,L}$, each set to, respectively, 1.2 V and 0.45 V. For an applied input of 0.45 V on N1, the $\overline{OUT}$ node discharges through N1 and N2 turns off, which results in a logic high at the OUT node. Note that a higher resistive path is required through N2 as compared to N1 to minimize charge leakage at OUT while $\overline{OUT}$ is transitioning to logic low. In addition, for an applied input of 0 V at the gate of N1, a nominal supply voltage of 1.2 V is applied to the gate of N2. The OUT node, therefore, discharges faster than $\overline{OUT}$, which reduces the propagation delay at a cost of increased power consumption. Note that for a set $V_{DD,H}$ of 1.2 V, $\overline{OUT}$ discharges to a minimum voltage level of 80 mV and 312 mV for a $V_{DD,L}$ of, respectively, 0.45 V and 0.4 V as there is a direct path from $V_{DD,H}$ to ground through P1 and N1. For input voltages below

0.4 V, the single ended level shifter does not operate as intended, since the voltage on $\overline{OUT}$ exceeds the threshold voltage of N2 (when 0 V is expected). Applying a voltage comparable to the threshold voltage of transistor N1 is sufficient to change the voltage at OUT, which results in an increased conversion range for the single ended level shifter as compared to the cross-coupled level shifter. The circuit techniques for improved energy efficiency implemented for cross coupled and current mirror based level shifters such as the use of a revised wilson current mirror [282] or a split input inverting buffer [281] also apply to the single ended level shifter.

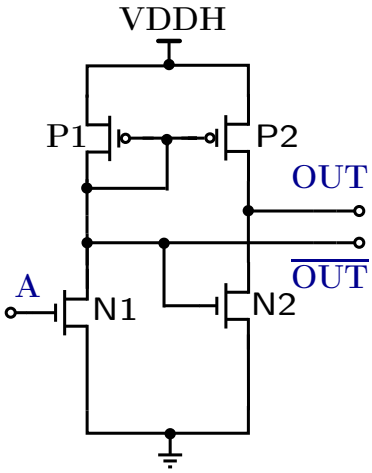


Fig. 6.20: Proposed level shifter with single ended input and differential output.

6.6 Bi-directional Input/Output Circuits

The proposed bi-directional I/O circuit for near-threshold operation is shown in Fig. 6.21, where integrated level shifters are labeled as *Level Shifter 1 (LS1)* and *Level Shifter 2 (LS2)*. The application of the near-threshold voltage CVDD to the core and

the nominal or I/O voltage NVDD are also shown in Fig. 6.21. The control signals IN_IDLE and EN set the operation of the I/O circuit to, respectively, input and output mode. For example, a logic high EN signal forces the outputs of the NAND1 and NOR to, respectively, logic high and logic low. The output driver, therefore, remains in a high impedance state. In addition, a logic low at the NAND2 gate forces the inputs DIN1 and DIN2 to logic low, irrespective of the signal on the PAD terminal. Additional level restorers are implemented using PMOS transistors P2 and P3 to prevent an unwanted switching at the input terminals DIN1 and DIN2 of the core.

Depending on the specific application, the optimum power-performance point requires multiple near-threshold power domains with a difference of tens of millivolts. Therefore, multiple input buffers (DIN1 and DIN2) are proposed within a single I/O circuit, where the implemented power domains are labeled as CVDD1 (0.4 V) and CVDD2 (0.45 V). The properties of the proposed I/O circuit are: 1) an optimized level shifter topology, where a fast conversion rate is required for *LS1* and an energy efficient topology is required for *LS2*, 2) isolation of the input circuits from the transients of the external signals by implementing level restorers P2 and P3 and applying IN_IDLE on NAND2, which reduce both the noise and power consumption, and 3) multiple near-threshold input modes for fine grain energy-efficiency in an integrated circuit with multiple voltage domains.

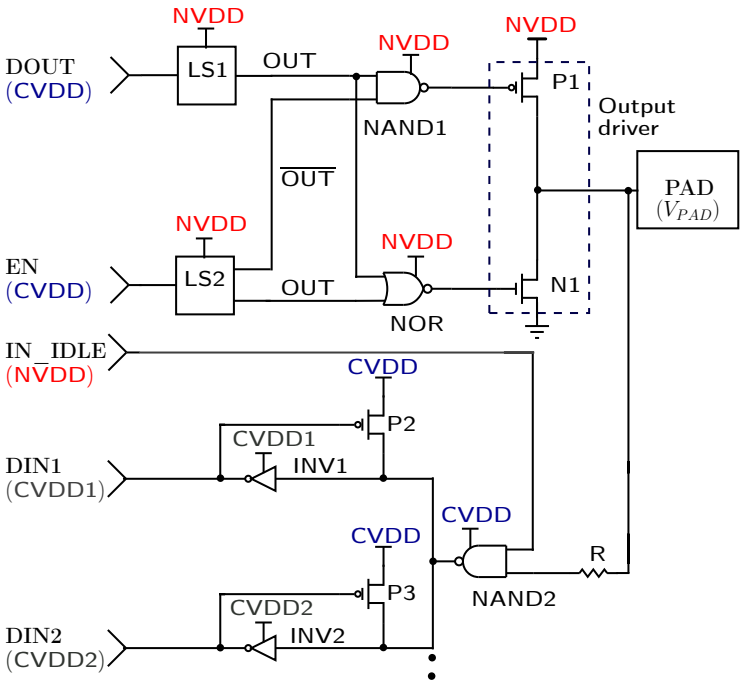


Fig. 6.21: Proposed bidirectional input/output circuit.

6.7 Simulated Results of Interface Circuit

Characterization of the interface circuit is described in this section. The simulation setup is provided in Section IV-A. The level shifters and I/O circuits are characterized in Sections IV-B and IV-C, respectively.

6.7.1 Simulation Setup

The simulation of the level shifters and I/O circuits is performed using an IBM 130 nm technology. The area of both the cross coupled and current mirror level shifters is 3.64 µm^2, while the area of the single ended level shifter is 2.47 µm^2 (calculated from the listed transistor sizes in Table 6.8). The level shifters and I/O circuits are

Table 6.8: Transistor size of the level shifters.

Cross coupled	**Current mirror**	**Single ended**
P1, P2 = 2 µm/0.13 µm × 2 N1, N2 = 8 µm/0.13 µm × 2 PMOS (Inverter) = 6 µm/0.13 µm NMOS (Inverter) = 2 µm/0.13 µm		P1, P2 = 0.6 µm/0.13 µm × 2 N1 = 18 µm/0.13 µm N2 = 0.4 µm/0.13 µm

simulated at, respectively, 17 MHz and 100 MHz operating frequencies. An output load of 5 fF is used for all data paths, which is more than 10x larger than the calculated gate capacitance of a standard CMOS inverter in the 130 nm technology node. The low to high voltage level shifters are simulated for multiple input ($V_{DD,L}$) and output ($V_{DD,H}$) voltages for comparison of the conversion range, total power consumption, and propagation delay. The nominal core and I/O voltage levels are set to, respectively, 1.2 V and 3.3 V. Supply voltages of 0.4 V and 0.45 V are considered for near-threshold as the transistor threshold voltages range from 0.35 V to 0.42 V. The proposed I/O circuit is evaluated through eight different configurations using the current mirror, cross-coupled, and the proposed single ended level shifter, as listed in Table 6.9. SPICE simulation to characterize the conversion range, total power, and propagation delay is performed for a set value of EN and IN_IDLE. EN is set to 0 V when the I/O circuit is in output mode, and IN_IDLE is set to 0 V when in input mode.

6.7.2 Characterization of Level Shifters

The conversion range of all level shifters is analyzed for output voltages of 1.2 V and 3.3 V. Note that the conversion range shifts with a change in the output

Table 6.9: Configurations of the I/O circuits based on the type of implemented level shifters.

Configuration	**Level shifter 1**	**Level shifter 2**
A	cross coupled	cross coupled
B	cross coupled	current mirror
C	single ended	cross coupled
D	single ended	single ended
E	single ended	current mirror
F	current mirror	single ended
G	current mirror	current mirror
H	current mirror	cross coupled

voltage $V_{DD,H}$. The current mirror level shifter and single ended level shifter have an approximately equal conversion range of 0.4 V to 1.2 V and 0.5 V to 3.3 V for output voltages of, respectively, 1.2 V and 3.3 V. However, the cross coupled level shifter exhibits a reduced conversion range of 0.5 V to 1.2 V and 0.625 V to 3.3 V for output voltages of, respectively, 1.2 V and 3.3 V. The propagation delay of the level shifters is compared with an inverter FO4 delay over a range of input voltages $V_{DD,L}$ and for a set output voltage of 1.2 V, as shown in Fig. 6.22. The single ended level shifter exhibits the minimum delay for all values of $V_{DD,L}$. For example, for a set $V_{DD,L}$ of 0.55 V and $V_{DD,H}$ of 1.2 V, the propagation delay of a FO4 inverter (for process characterization), cross coupled level shifter, and current mirror level shifter is, respectively, 4.62x, 2.72x, and 1.42x longer than the single ended level shifter.

The comparison of the propagation delay and total power consumption of the level shifters is performed for a set $V_{DD,L}$ of 0.625 V as $V_{DD,H}$ is swept from 0.7 V to 3.3 V. The delay of the cross coupled level shifter increases with increasing output voltage, while the delay of the current mirror and single ended level shifters

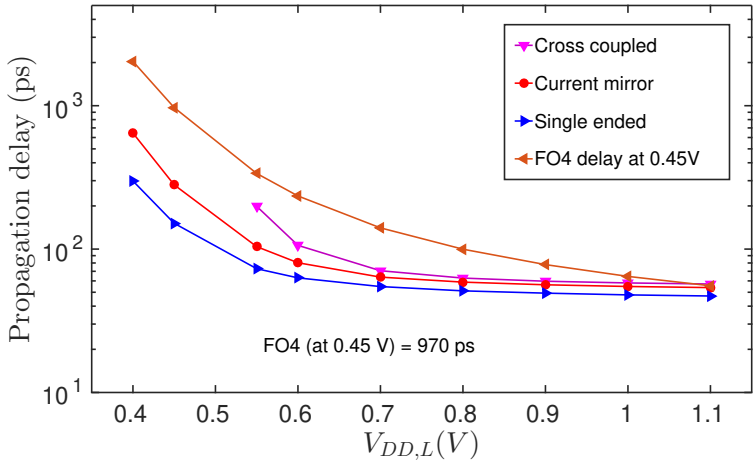


Fig. 6.22: Comparison of the delay of the level shifters at an output voltage $V_{DD,H}$ of 1.2 V.

is reduced for higher output voltages, which is shown in Fig. 6.23(a). The reduction in delay is more significant at higher output voltages for the single ended level shifter as compared to the current mirror topology since a higher potential is applied to the gate of N2. The cross coupled level shifter consumes the least power of all topologies over the range of output voltages examined, as shown in Fig. 6.23(b). The power consumption of the single ended level shifter is, however, less than the current mirror topology over the entire range of output voltages. For a $V_{DD,L}$ of 0.625 V and $V_{DD,H}$ of 1.5 V, the propagation delay of the single ended level shifter is 0.53x and 0.74x the delay of, respectively, the cross coupled and current mirror level shifters. In addition, the total power consumption of the single ended level shifter is 33x and 0.4x the power consumed by, respectively, the cross coupled and current mirror level shifters. Although the cross coupled level shifter consumes much less power, the delay is significantly larger (4.62x) than the proposed single ended topology.

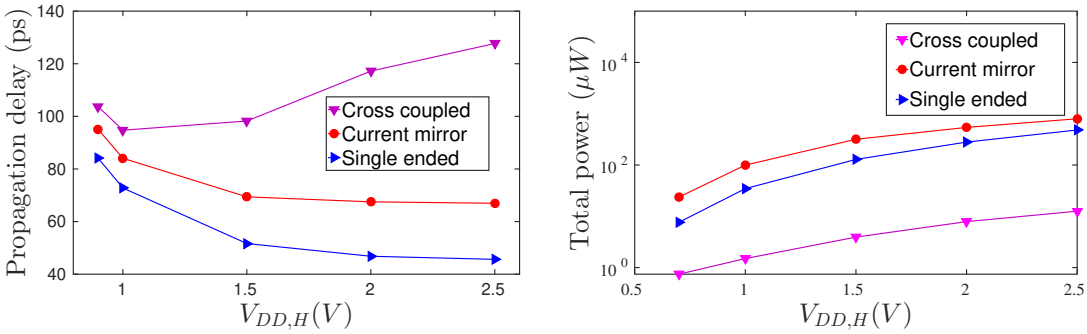


Fig. 6.23: Comparison of the a) propagation delay, and b) total power of the level shifters for an input voltage $V_{DD,L}$ of 0.625 V.

The single ended level shifter exhibits reduced conversion delay with a significant power overhead as compared to the cross coupled level shifter, while exhibiting reduced delay and power consumption as compared to the current mirror level shifter. In addition, the cross coupled level shifter suffers from a fundamental limitation in the minimum voltage that can be converted for a given output voltage, where, for the 130 nm process, input voltages of 0.55 V and 0.625 V are required for output voltages of, respectively, 1.2 V and 3.3 V. The cross coupled level shifters are, however, beneficial for relatively large input voltages, where the propagation delay is not significantly impacted. A single ended level shifter therefore exhibits the best characteristics when operating at near-threshold voltages, as 0.4 V and 0.45 V are up-converted to, respectively, a nominal IC voltage of 1.2 V and an I/O voltage of 3.3 V with a faster conversion time as compared to cross coupled and current mirror topologies.

The conversion range, performance, energy per transition, and area of proposed level shifters are compared with state-of-the-art level shifters for near/sub-threshold

Table 6.10: Comparison with state-of-the-art low voltage level shifters.

Parameters	**VLSIC 2011****	**JSSC 2012***	**TCASI 2015****	**TCASI 2017***	**JLPEA 2016****	**TCASII 2017****	**This work ***
Process	130 nm	350 nm	180 nm	180 nm	130 nm	180 nm	130 nm
Conversion range	0.3 V to 2.5 V	0.23 V to 3 V	0.21 V to 3.3 V	0.3 V to 1.8 V	0.145 V to 1.2 V	0.1 V to 1.8 V	0.4 V to 1.2 V, 0.5 V to 3.3 V
Delay (FO4)	2.38 (0.3 V to 2.5 V)	Not provided	~1.05 (0.3 V to 1.2 V)	Not provided	Not provided	2.2 (0.3 V to 1.2 V)	0.146 (0.4 V to 1.2 V)
Delay (ns)	41.51 (0.3 V to 2.5 V)	~8500 (0.4 V to 1.2 V)	~170 (0.3 V to 1.2 V)	17.3 (0.4 V to 1.8 V)	>200	31.7 (0.4 V to 1.8 V)	0.2986 (0.4 V to 1.2 V)
Area (μ^2)	102.6	1880 **	153.01	229.5 **	466	108.8	2.28^
Energy/transition (fJ)	229 (0.3 V to 2.5 V)	~4250 (0.4 V to 1.2 V)	~8 (0.3 V to 1.2 V)	56 (0.4 V to 1.8 V)	1200 (0.145 V to 1.2 V)	173 (0.4 V to 1.8 V)	570 (0.4 V to 1 V), 1600 (0.45 V to 1.2 V)
* Simulated ** Measured ^Active channel area Note: FO4 delays are measured at VDDL							

operation, as listed in Table 6.10. The proposed level shifter exhibits the smallest delay among the compared level shifter topologies, which is 0.146 FO4 delay (0.2986 ns) when converting from 0.4 V to 1.2 V. The energy overhead of the proposed level shifter is compensated by the energy savings in the I/O circuit, where short circuit power is significantly reduced due to the faster response time of *Level Shifter 1 (LS1)*.

6.7.3 Characterization of I/O Circuit Configurations

The conversion range of each type of I/O circuit configuration is listed in Table 6.11 for PAD voltages V_{PAD} of 1.2 V and 3.3 V. Configurations A and B of the I/O circuit exhibit a limited conversion range for both PAD voltages since the cross coupled level shifters are used for *LS1* in Fig. 6.21. A comparison of the propagation delay and total power consumption of the different I/O circuit configurations is shown in, respectively, Fig. 6.24(a) and 6.24(b) across a range of V_{PAD} voltages and for a DOUT voltage V_{DOUT} of 0.55 V. Configurations E and F are implemented with, respectively, a single ended and current mirror level shifter for *LS1*. Although there is only a minor increase in the propagation delay of the current mirror level shifter as compared to the single ended topology, the overall power consumption of the I/O

circuit (configuration F) increases significantly with the use of a current mirror level shifter, where configurations G and H also exhibit similar characteristics. The power consumption of the pre-driver NAND1 and NOR gates increases significantly for a larger rise or fall time of the inputs due to increased short circuit current, which negatively impacts the overall power consumption of the I/O circuit. However, a set value of EN is used in this work, which avoids the periodic switching of the NOR gate. Configurations A and B exhibit the minimum power consumption with a significant increase in delay. However, A and B are not feasible for near-threshold operation unless the input voltage V_{DOUT} to *LS1* is at least 0.625 V.

Among the eight configurations, C, D, and E provide the optimum power-delay points as shown in Fig. 6.24, where the single ended level shifter is used in place of *LS1*. An improvement in the delay and power is seen for configuration C as compared to D and E for both a V_{PAD} of 1.5 V and 2.5 V (simulation was performed sweeping V_{PAD} from 0.7 V to 3.3 V), since the low power cross coupled level shifter is used in place of *LS2*. However, the use of the cross coupled level shifter limits the operating range of the I/O circuit. In addition, D and E exhibit almost similar power and delay results for both V_{PAD} voltages of 1.5 V and 2.5 V since *LS2* is not switching (EN is set to 0). Note that E is not affected by the increased delay of *LS2* as compared to *LS1*, since the switching frequency of the control signal EN is less likely to change than V_{DOUT}. Therefore, the use of a single ended level shifter in place of

Table 6.11: Conversion range of I/O circuit configurations for V_{PAD} of 1.2 V and 3.3 V.

V_{PAD}	**A, B**	**C, D, E, F, G, H**
1.2 V	0.53 V to 1.2 V	0.38 V to 1.2 V
3.3 V	0.55 V to 3.3 V	0.45 V to 3.3 V

LS1 and either a single ended or current mirror level shifter in place of *LS2* provides the optimum power-delay operating point with the maximum conversion range. The optimal configuration, *E*, at a supply voltage of 0.45 V (CVDD), consumes a total power of 4.87 mW and has a FO4 delay of 0.46 for a V_{PAD} voltage of 3.3 V.

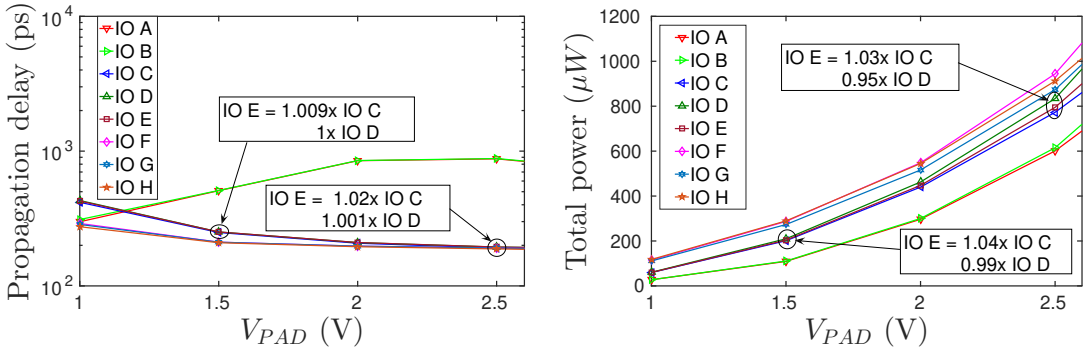


Fig. 6.24: Comparison of different I/O circuit configurations for a V_{DOUT} of 0.55 V. a) Propagation delay, and b) total power consumption.

6.8 Summary

In this chapter, novel logic families and interfacing circuits are discussed that are capable of ultra-low voltage operation, including circuits that operate at near-threshold supply voltages. Three novel dynamic differential logic based families (DCML, LDCML, and DFCML) are developed to operate at ultra-low voltages that

outperform CMOS and CML based circuits in power consumption, performance, and robustness. In addition, circuits are developed to interface between circuit blocks that operate at near-threshold voltages with those that operate at nominal or I/O voltage levels. A single ended level shifter and bi-directional I/O circuit are developed for near-threshold computing. The single ended level shifter provides conversion ranges of 0.4 V to 1.2 V and 0.5 V to 3.3 V. The optimized configuration of the I/O circuit provides a conversion range of 0.38 V to 1.2 V for core voltages and 0.45 V to 3.3 V for I/O voltages.

Chapter 7

Clock-Power Co-Deign and Optimum Voltage Selection

Ultra-low voltage logic and memory circuits have shown proper functionality and high energy-efficiency at voltages less than 350 mV [144]. For example, an FIR filter has been designed to operate at 85 mV and 240 Hz in a 130 nm technology [283]. A sub-V_t processor was implemented for sensor networks in a 130 nm technology with a 130 mV supply voltage, while consuming 11 nW of power [284]. A sub-V_t SoC was implemented in a 130 nm technology for wireless electrocardiogram (ECG) monitoring with a 280 mV supply, while consuming 2.6 μW of power [285]. In addition, an adaptive 32b sub-V_t processor was implemented that dissipates 27.2 pJ per instruction, while operating at 325 mV [286]. In the work described in this chapter, through

SPICE analysis of an inverter, 140 mV is determined as the minimum functional supply voltage that permits 100 KHz operation in a 130 nm CMOS technology. Despite extensive research on developing cores and memories for sub-threshold computing, the analysis of power and clock delivery and the interaction between the power and clock networks at voltages less than 350 mV has not been addressed by the research community.

The energy efficiency of integrated circuits is determined by the appropriate selection of supply and threshold voltages for a given technology and performance requirement. Supply voltage scaling provides an optimum power-delay and a minimum energy point at the expense of performance, while threshold voltage scaling improves the performance at the expense of reduced noise margins [164, 273, 287–289]. Prior research addressed supply and threshold voltage tuning considering switching speed, power consumption, total energy, and circuit characteristics [290–293]. However, the noise margins and f_{max} are rarely considered when optimizing the supply and threshold voltages. Aggressive supply voltage scaling in the range of the near- and sub-threshold voltage of a transistor severely degrades the noise margins and stability of the circuit [287, 294]. Threshold voltage scaling and deviation due to process, voltage, and temperature (PVT) variation reduce the robustness of the circuit to noise [287, 289, 294]. In addition, dynamic voltage and frequency scaling (DVFS) does not account for the noise margins of the circuit when the supply voltage is scaled to

reduce the operating frequency of the circuit [295].

An exponential relationship exists between the supply voltage and the f_{max} of the circuit when operating in sub-threshold (0.1 V to 0.4 V), as shown in Fig. 3.1. The NM_{avg} (mean of noise margins high and low) decrease linearly with supply voltage scaling, which is also shown in Fig. 3.1. In addition, variation in threshold voltage inversely affects the noise margins and f_{max} [296]. Therefore, an optimization of the supply and threshold voltages is required for different circuit constraints as there are trade-offs between operating frequency, energy efficiency, and robustness to noise.

In this chapter, a clock-power co-design methodology is developed for deep sub-threshold circuits operating at 250 mV. A distributed, multi-voltage domain, and hierarchical power distribution network (PDN) is developed to deliver power to the components of the sub-threshold clock distribution network. The skew variation and timing conditions of the circuit are evaluated for both a conventional single domain power distribution network and the proposed distributed, multi-domain, and hierarchical power distribution network. In addition, the negative impacts of power supply noise on skew variation and timing are characterized for a sub-threshold voltage of 250 mV by using both conventional and optimized clock buffers.

In addition, an optimization technique for the selection of the supply voltage is developed for a given minimum and maximum range of threshold voltages. The optimization model is developed based on the variations in noise margins and f_{max}

due to supply and threshold voltage variations. The change in threshold voltage due to supply voltage variation is also considered in the proposed model. For a given f_{max} and NM_{avg}, the proposed algorithm implicitly improves the energy-efficiency of the circuit by reducing the supply and threshold voltages.

The rest of the chapter is organized as follows. The challenges of sub-threshold power and clock distribution are discussed in Section 7.1. The proposed clock-power co-design methodology is presented in Section 7.2.1. The selection of the voltage regulator topology to optimize the energy efficiency of the circuit is discussed in Section 7.2.2. Simulation results for clock-power co-design are described in Section 7.3. The description and evaluation of the proposed algorithm for the optimum selection of supply voltage is discussed in Section 7.4. The simulation results from executing the optimization techniques are described in Section 7.5.

7.1 Challenges of Sub-Threshold Power and Clock Distribution

Sub-threshold operation imposes some fundamental challenges, including a reduced I_{on}/I_{off} ratio of the transistors and increased sensitivity to random dopant fluctuation, which degrades robustness to noise and increases threshold voltage variation. Due to an exponential relationship between the transistor gate to source voltage V_{GS}, drain to source voltage V_{DS}, and threshold voltage V_t with the drain current in sub-threshold, a significant degradation in the delay is possible for a 5% variation

in the supply voltage (assuming a 250 mV supply). Therefore, the impact of IR drop, $L \cdot di/dt$ switching noise, and process variation is much more severe for circuits operating in sub-threshold as compared to circuits operating at a near-threshold or nominal voltage, even though the magnitude of transient switching noise (di/dt) in sub-threshold circuits is smaller.

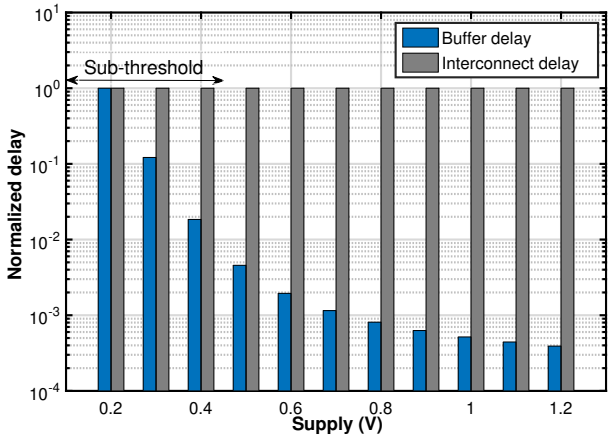


Fig. 7.1: Effect of supply voltage scaling on buffer and interconnect delay.

Conventionally, in super-threshold clock distribution networks (CDN), clock buffers are placed to mitigate clock skew and satisfy timing constraints as the interconnect delay is the dominant component of the total delay of a clock signal [84, 297]. However, in a sub-threshold clock distribution network, the logic or buffer delay dominates the total delay of a clock signal due to the significant increase in the delay of the transistors. The primary challenges of low voltage clock distribution networks are clock uncertainty and power overhead [298]. The various sources of clock uncertainty include clock generation circuits, device variation, interconnect variation, and power

supply noise [298]. An exponential increase in transistor delay and sensitivity to process, voltage, and temperature (PVT) variation in sub-threshold operation imposes new challenges to mitigate clock skew, slew, and jitter, particularly for circuits operating at a voltage of less than 300 mV [299, 300]. In addition, the drive strengths of clock buffers are reduced by an order of magnitude in sub-threshold, which significantly affects the skew and increases the delay. The delay of a clock buffer and a 200 μm interconnect are characterized with SPICE simulation in a 130 nm CMOS technology between 200 mV and 1.2 V and the results are shown in Fig. 7.1. The propagation delay of the buffer and the interconnect at a 200 mV supply voltage is, respectively, 229 ns and 1.53 ps, which are used to normalize all other results at higher supply voltages. Unlike the buffer delay, the interconnect delay does not change with supply voltage, as shown in Fig. 7.1. Therefore, implementing a deeper clock network and meeting timing constraints becomes more challenging when the clock buffers are operating in sub-threshold [300].

Prior work has explored clock distribution networks operating in sub-threshold. A clock network designed with multiple supply voltages was proposed in [300] to reduce clock skew, where the supply voltage was reduced to 340 mV. An unbuffered clock tree operating at 300 mV was proposed to minimize skew, slew, and energy consumption [299, 301]. In addition, a slew aware clock tree was developed that operates at 300 mV, where a larger and smaller slew was implemented in, respectively, the root and

leaf nodes of the tree by dynamically controlling nodal capacitances [302, 303]. A capacitive boosting based interconnect technique was proposed in [304] to reduce clock skew for a 400 mV supply voltage. The use of separate power distribution networks for buffers at different levels of the clock network was proposed to reduce the clock jitter of high performance systems[305]. In addition, a DC-DC converter was implemented for sub-threshold circuits to deliver power at 250 mV [306]. However, no prior work addressed the impact of power supply noise on the skew and timing of the clock distribution network in sub-threshold. In addition, clock distribution below 300 mV is not addressed in prior work.

7.2 Clock-Power Co-Design for Ultra-Low Voltage Circuits

The timing of digital circuits is dependent on the implemented combinational logic and the clock distribution network of the circuit, which includes clock buffers, global interconnect, and flip-flops. A subsection of a clock distribution network is shown in Fig. 7.2, where all components are supplied current through a single power distribution network. The static and transient noise induced on the power distribution network affect all components of the clock distribution network. Prior research used clock data compensation techniques, where voltage noise on the clock network is allowed to match the peak voltage noise on the datapath [307–309]. However, the technique is not applicable to sub-V_t clock networks due to the significant increase in

the sensitivity of the circuit to variation.

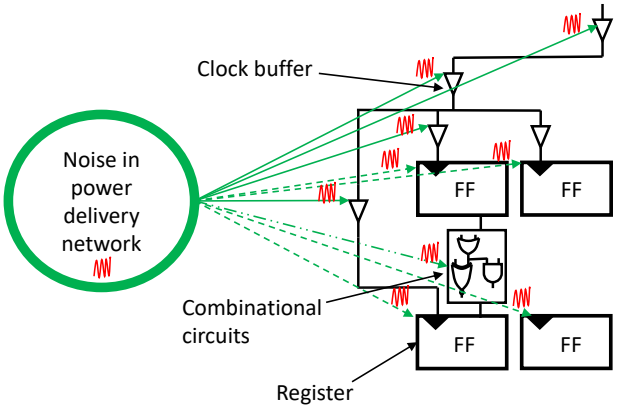


Fig. 7.2: Noise propagation from the power supply network to a subsection of a clock network operating in sub-V_t.

In a sub-threshold CDN, the propagation delay of the clock buffers, combinational circuits, and flip-flops significantly increase due to a minor voltage drop in the power network, which affects the timing profile of the sub-threshold CDN. Therefore, there is a greater and direct effect of power supply noise on the performance and stability of the sub-threshold clock network. In addition, local and global process variation disproportionally affect the delay of the clock buffers, combinational logic, and flip-flops across the circuit. At sub-threshold voltages, process variation results in increased skew variation and slew degradation, which causes timing failure. Therefore, robust circuit and system level techniques are required for sub-threshold clock distribution networks to meet the timing requirements of the circuit under the combined effects of power supply noise and process variation.

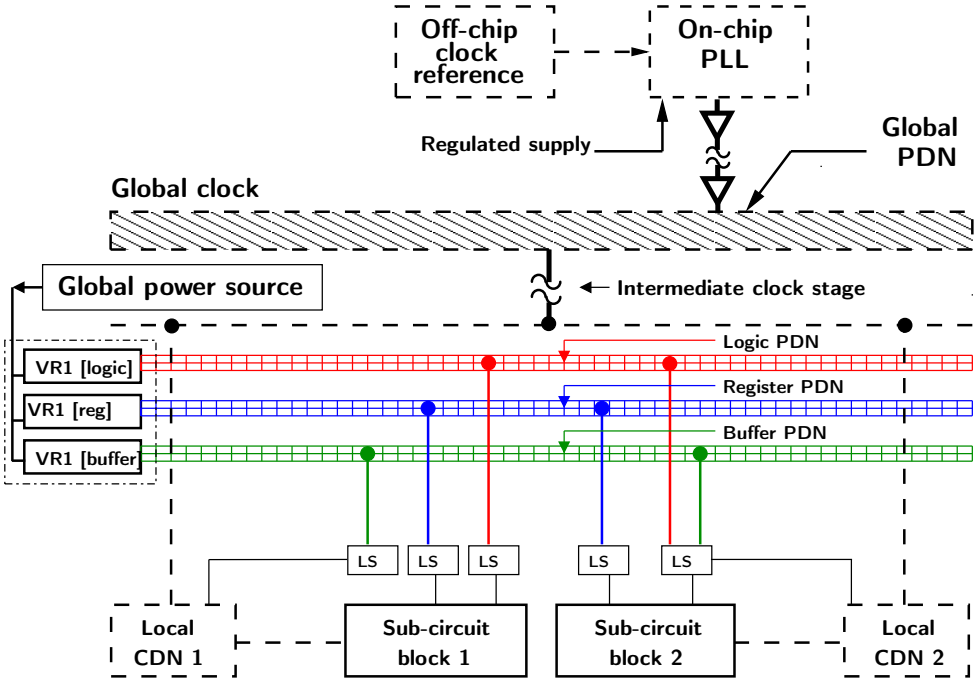


Fig. 7.3: Circuit model of the co-designed clock-power networks.

7.2.1 Multi-Voltage Domains and Distributed Power Delivery for Local Clock Distribution Networks

A clock-power co-design methodology is proposed for ultra-low voltage circuits operating in the range of 200 mV to 300 mV. For this paper, only 250 mV operation is described. Instead of using a single power domain to deliver current to combinational logic, registers, and clock buffers, a design time reconfiguration of the sub-V_t power distribution network is proposed to ensure timing constraints are met at sub-threshold voltages. The sub-threshold PDN is implemented as a multi-voltage

domain, distributed, and hierarchical network. The primary objectives of the co-design methodology are to 1) improve the stability and performance of the CDN in sub-threshold, and 2) mitigate the effects of supply noise on the clock network.

The circuit model that includes the co-designed clock and power networks developed to analyze the effect of power supply noise on skew variation is shown in Fig. 7.3. The clock and power networks are represented by, respectively, dashed and solid lines. An off-chip crystal oscillator is assumed, which drives an on-chip phase-locked loop (PLL). The PLL drives a global clock distribution network, which is then connected to local CDNs. The PLL is supplied current through either off- or on-chip voltage regulators. The global clock network is supplied current through a global PDN, which isolates noise propagation from the switching activities of the local clock domains, registers, and logic circuits. The synchronous circuit blocks (e.g. flip flops), local clock buffers, and combinational logic circuits are supplied current through separate PDNs labeled as, respectively, Register PDN, Buffer PDN, and Logic PDN in Fig. 7.3.

A DC-DC buck converter is considered for the first-stage conversion as a switching regulator drives higher current loads with higher efficiency as compared to linear and switched-capacitor voltage regulators [84]. The three first-stage on-chip voltage regulators (OCVR) supplying current to the logic, register, and buffer circuits are labeled as, respectively, VR1 [logic], VR1 [reg], and VR1 [buffer] in Fig. 7.3 and

convert a battery voltage (3.3 V to 20 V) to an intermediate voltage between 0.6 V to 1.2 V. Further details regarding the on-chip voltage regulators is provided in Section 7.2.2.

High-to-low level shifters (LS) are used to produce output voltages between 0.2 V and 0.4 V [310]. The use of LS circuits reduces the area and power overhead as a fewer number of voltage regulators are now required to generate output voltages between 0.2 V and 0.4 V. The input and output voltage of the regulators and the output supply voltage of the level shifters are controlled by the power control unit (PCU).

A distributed and multi-voltage domain power delivery system to deliver current to two local clock distribution networks and sub-circuits labeled as *Local CDN 1* and *Local CDN 2* is shown in Fig. 7.3. Separate power distribution networks are assumed for each local clock network. The local CDNs consist of only sub-V_t clock buffers, while sub-circuit blocks 1 and 2 consist of combinational circuits, registers, gates used for clock gating, and level shifters. However, in this paper, the sub-circuits connected with each local CDN include only registers and combinational logic. Each local clock domain includes a dedicated PDN for buffers. The clock depth and the number of implemented buffers is limited when delivering the clock signal in sub-threshold due to more stringent timing constraints. In addition, the sub-circuit block connected with each local CDN includes one or two PDNs, each providing current

to combinational logic and registers through level shifters (see Fig. 7.3). The output voltage of each LS is set individually to any value between 0.2 V and 0.4 V, which enables distributed and multi-voltage domain operation in sub-threshold. Note that when a power domain is isolated through a separate PDN, the respective ground network is also isolated to ensure perfect noise isolation.

The proposed characteristics of the clock-power co-designed circuit are 1) distributed and multi-voltage domain power delivery to the circuits close to the clock leafs (local PDN), where current to each local PDN is individually supplied by an LS, 2) isolated logic, buffer, and register PDNs to reduce the power supply noise seen by the clock buffers and registers, where the isolated PDNs are operated with independent sub-threshold supply voltages, 3) operating the logic, register, and combinational logic within a local clock distribution network at a similar sub-threshold voltage if the timing constraints and target frequency requirements are satisfied for the maximum noise in the supply voltage rail, 4) operating the logic, buffer, and register PDNs associated with a local CDN in different sub-threshold voltages for fine grained tuning between delay, power, and noise margins, and 5) merging any two PDNs among the three if the effect of power supply noise on the clock network is minimal for the given circuit block, which also reduces the design complexity.

Two scenarios are illustrated in Fig. 7.3: 1) local CDN 1 and sub-circuit block 1 with three separate PDNs for buffers, logic, and registers, and 2) local CDN 2

and sub-circuit block 2, where a single PDN is used for buffers and combinational logic and all registers are supplied current through a second PDN. The circuit model includes both global and local on-chip clock distribution networks. However, in this paper, the analysis is limited to the local clock distribution network.

7.2.2 Voltage Regulator Selection for the Proposed Clock-Power Network

In the clock-power co-designed circuit shown in Fig. 7.3, three on-chip buck converters are used to generate an output voltage of 0.6 V to isolated PDNs that supply current to the logic, register, and buffers. The three regulators are not adding additional power loss as the power dissipation of the switching regulators is directly proportional to the output current for a given input and output voltage as described by (7.6).

The basic circuit diagram of a switching DC-DC buck converter is shown in Fig. 7.4, where MOSFETs are used as switches $Q1$ and $Q2$ [84]. The primary components of the power dissipation of a buck converter are the conduction loss of the inductor given by (7.1), the MOSFET conduction loss on the high-side MOSFET $Q1$ and low-side MOSFET $Q2$ given by, respectively, (7.2) and (7.3), and the MOSFET switching loss, which is not provided in this paper as switching losses are highly dependent on switching frequency [311, 312]. I_o, I_L, I_{Q1}, and I_{Q2} represent the current through, respectively, the output node of the buck converter, the inductor L, the $Q1$

switch, and the $Q2$ switch.

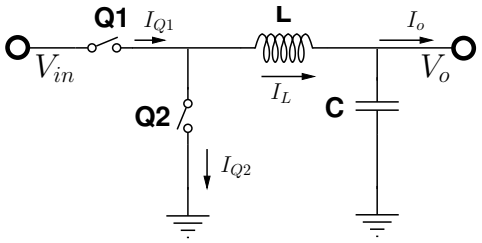


Fig. 7.4: Basic circuit representation of a switching DC-DC buck converter.

$$\begin{aligned} P_L &= I_L^2 \cdot R_L \\ &\approx I_o^2 \cdot R_L \end{aligned} \tag{7.1}$$

$$\begin{aligned} P_{Q1} &= I_{Q1}^2 \cdot R_{Q1} \\ &= \frac{V_o}{V_{in}} \cdot I_L^2 \cdot R_{Q1} \end{aligned} \tag{7.2}$$

$$\begin{aligned} P_{Q2} &= I_{Q2}^2 \cdot R_{Q2} \\ &= (1 - \frac{V_o}{V_{in}}) \cdot I_L^2 \cdot R_{Q2} \end{aligned} \tag{7.3}$$

The total power dissipated by the MOSFET switches is given by

$$P_{MOS} = P_{Q1} + P_{Q2}$$

$$P_{MOS} = I_L^2 \cdot \left(\frac{V_o}{V_{in}} \cdot R_{Q1} + (1 - \frac{V_o}{V_{in}}) \cdot R_{Q2} \right)$$

$$P_{MOS} = I_o^2 \cdot M, \tag{7.4}$$

$$\text{where } M = \left(\frac{V_o}{V_{in}} \cdot R_{Q1} + (1 - \frac{V_o}{V_{in}}) \cdot R_{Q2} \right).$$

The total power dissipation of the buck converter is, therefore, given by

$$P_{Buck} = P_L + P_{MOS} + P_{other} \tag{7.5}$$

$$\approx I_o^2 \cdot R_L + I_o^2 \cdot M$$

$$\approx I_o^2 (R_L + M), \tag{7.6}$$

where V_o/V_{in} is the output to input voltage ratio of the buck converter, R_L is the DC resistance of the inductor L, I_o is the output load current of the regulator, and R_{Q1} and R_{Q2} are the on-time drain-to-source resistances of MOSFET $Q1$ and $Q2$, respectively. The switching loss of the MOSFETs and the power loss due to quiescent current is represented by P_{other}, which is not considered in this paper as quiescent current is independent of output current and, as previously mentioned, switching losses are highly dependent on switching frequency. Ignoring P_{other}, the total power consumed by the buck converter is given by (7.6).

To further analyze the regulator loss and efficiency for variation in current demand, an industry standard switching DC-DC buck converter (Analog Devices ADP1851)

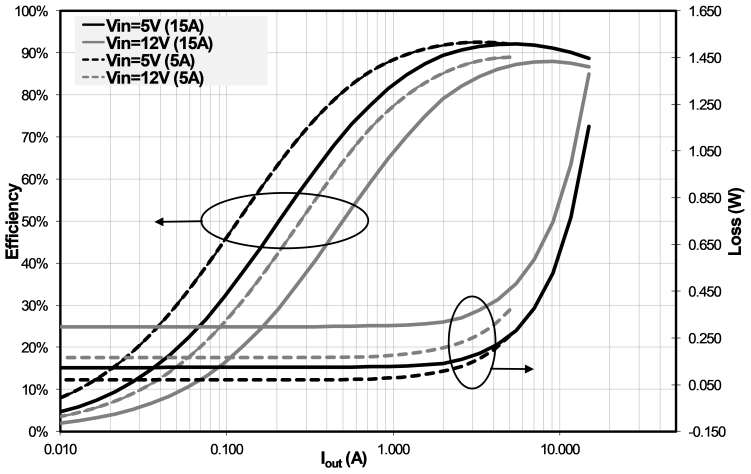


Fig. 7.5: Conversion efficiency and regulator loss of a DC-DC buck converter that generates an output voltage of 0.6 V from input voltages of 5 V and 12 V for current loads of 5 A and 15 A.

is simulated using ADIsimPE. The converter takes an input voltage V_{in} in the range of 2.75 V to 20 V and generates an output voltage between 0.6 V and 90% of V_{in}, for a maximum output current load of 25 A [313]. The buck converter is simulated with a 5 A output current load to emulate the current demand of logic, registers, and buffers supplied current through isolated PDNs, while a 15 A output current is used to emulate a single bulk regulator delivering current through a single PDN to the logic, registers, and buffers. The converter efficiency and total power loss of the regulator is shown in Fig. 7.5 for output currents between 0.01 A and 15 A, while generating an output voltage of 0.6 V from input voltages of 5 V and 12 V. The total power loss of the buck converter for an input voltage of 5 V (12 V) is 0.26 W (0.37 W) and 1.15 W (1.4 W) for, respectively, a 5 A and 15 A output current load. Therefore, for similar input and output voltages, the regulator power loss P_{Buck} given by (7.5) increases by

3.8× when the output current increases to 15 A from 5 A. However, no significant change in conversion efficiency is observed between a regulator designed to supply an output load current of 5 A and a regulator supplying 15 A as the power loss of the buck converter minimally impacts the efficiency. Therefore, using three separate buck converters for logic, registers, and buffers does not incur additional power loss, while maintaining a high conversion efficiency.

7.3 Simulation Results on Clock-Power Co-Design

All SPICE simulations are performed in a 130 nm CMOS technology. Characterization of the variation in the delay of flip-flops due to process variation at different supply voltages is described in Section 7.3.1. The timing characteristics of a flip-flop operating in sub-threshold are discussed in Section 7.3.2. The effect of power supply noise on the clock skew and circuit timing is analyzed in Section 7.3.3.

7.3.1 Delay Variation of Flip Flops Under Supply Variation

The variation in flip-flop (FF) delay at sub-threshold voltages is analyzed for three different process corners (*tt*, *ss*, and *ff*). A flip-flop topology based on transmission gates is implemented and simulated in SPICE. The variation in flip-flop parameters including setup time (t_s), hold time (t_h), and clock-to-q delay ($t_{clk-to-q}$) is analyzed for supply voltages between 200 mV and 1.2 V, although simulation results are provided in

Fig. 7.6 for up to 450 mV. The average variation of the flip-flop parameters for process corners in deep sub-threshold (250 mV) is more than 500× greater than the variation of the parameters in super-threshold operation (1.2 V). The average variation in flip-flop parameters is determined by 1) taking the difference in delay between the *tt* and *ss* case and setting the value to x, 2) taking the difference between the *tt* and *ff* case and setting the value to y, and 3) calculating the average variation in flip-flop parameters as $(x + y)/2$. As a result, power supply noise of a few millivolts drastically changes the timing characteristics of flip-flops operating in sub-threshold, which, therefore, requires a power delivery system that is robust to noise.

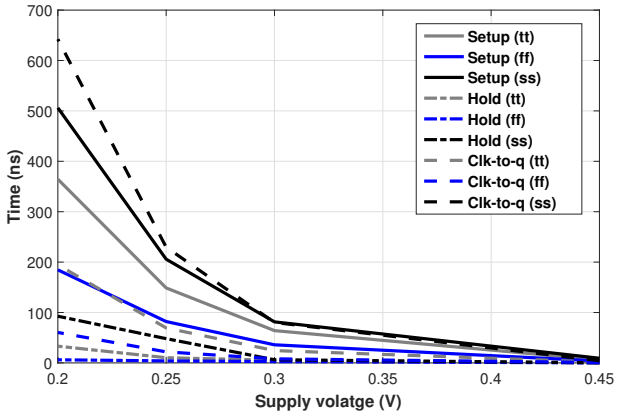


Fig. 7.6: Delay variation of a flip-flop operating in sub-threshold.

7.3.2 Timing Constraints and Skew Variation in Sub-Threshold Circuits

A local clock branch that includes two sequentially adjacent flip-flops is used to model a single node of a clock distribution network for SPICE simulation performed at 250 mV, as shown in Fig. 7.7. The circuit is used to characterize the clock skew

and the effects of power supply voltage variation on the timing of sequential circuits.

The clock insertion point is labeled as *M*. The interconnect (*wire* in Fig. 7.7) impedance is represented as an equivalent π-model to accurately represent the clock network. A 200 μm long interconnect is used in each path. Two series connected clock buffers (*B* in Fig. 7.7) that operate at 250 mV are inserted midway along each path, which splits the interconnect into two 100 μm long segments, one before and the other after the buffers. Four π-segments each representing 25 μm of length are used to model each 100 μm wire segment. The flip-flops are optimally resized to operate at 250 mV. The sheet resistance R_S of a $M3$ interconnect in a 130 nm technology is 0.0584 ± 0.0217 Ω/square. In addition, the capacitance per unit length C of the interconnect is 1.8 to 2.2 pF/cm as given in the 2013 ITRS [314]. The wire width and C considered for the analysis are, respectively, 0.2 μm and 2 pF/cm, giving a resistance R_π and capacitance C_π of, respectively, 7.3 Ω/π-segment and 4.125 fF/π-segment.

The effects of power supply noise on the clock skew and the overall timing margins are analyzed for a supply voltage of 250 mV. If the launching and capturing registers are defined as, respectively, m and s, then the insertion delays of the m (t_{mi}) and s (t_{si}) registers are used to calculate the clock skew (δ), as given by (7.7) and (7.8). For the analysis in this paper, $\delta_{adjusted}$ is used, as δ_{worst} is overly pessimistic.

$$\delta_{worst} = t_{si}(ss) - t_{mi}(ff) \tag{7.7}$$

$$\delta_{adjusted} = [t_{si}(ss) - t_{mi}(ff) + t_{si}(ss) - t_{mi}(tt)]/2 \tag{7.8}$$

The longest (or minimum clock period) and shortest (or race condition) path delay constraints must be met to assure the proper timing of the circuit. The constraints to determine the minimum clock period and to avoid a race condition are given by, respectively, [298]

$$T \geq t_{clk-to-q,max} + t_{logic,max} + t_s - \delta, \text{ and} \tag{7.9}$$

$$\delta < t_{clk-to-q,min} + t_{logic,min} - t_h. \tag{7.10}$$

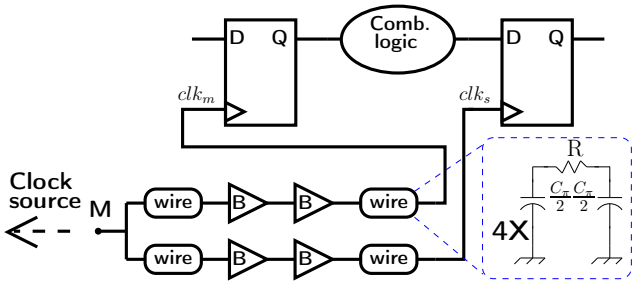


Fig. 7.7: Circuit model used to analyze the effect of supply noise on timing.

A PDN with a single voltage domain and the proposed PDN with up to three voltage domains are simulated to analyze the effects of supply voltage variation on the delay of the components of the sub-threshold clock distribution network [298]. In addition, the typical corner is used to characterize the setup time $t_{s,tt}$, hold time $t_{h,tt}$, and clock-to-q delay $t_{clk-to-q,tt}$ of the register, as the primary objectives of this

work are to analyze 1) the effects of power supply noise on the skew, and 2) the effect on timing due to noise induced skew variation. The buffers are sized to produce symmetric rise and fall times in the operating mode each is optimized for. Therefore, two different sizing ratios are used for the nominal and sub-threshold buffers: 1) nominal buffers include two inverters with a PMOS width W_p of 3.6 μm and an NMOS width W_n of 1.2 μm, and 2) sub-V_t buffers include two inverters resized to optimize the sub-threshold delay with a PMOS width W_p of 2.4 μm and an NMOS width W_n of 2.4 μm. The total area for both the nominal and sub-threshold buffers is the same for iso-area comparison. Note that the P/N ratio of the sub-V_t buffers is not the ideal ratio for different process corners and voltages, but is kept constant across all simulations, providing results and intuition on the effect non-optimized buffers have on the timing characteristics of the circuit. The DC behavior of the nominal and sub-V_t inverters is shown in Fig. 7.8. The sub-V_t inverter exhibits symmetric behavior at a supply voltage of 250 mV for the typical process corner, while the voltage transfer curve (VTC) of the nominal inverter (non-optimized for sub-threshold operation) is shifted to the right as a stronger PMOS response is exhibited. The 25 mV ($V_{dd}/10$) shift from the sub-V_t to nominal VTC implies an increase in the fall time of the inverter. A chain of four inverters is simulated with nominal and sub-Vt inverters to further characterize the delay variation at a sub-threshold voltage of 250 mV, with results indicating a delay of 290 ns and 204 ns, respectively.

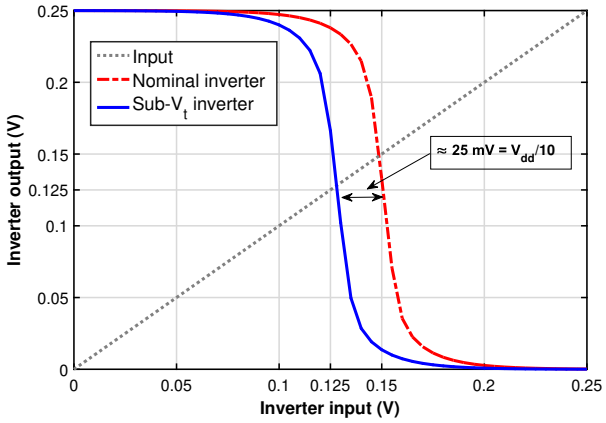


Fig. 7.8: DC behavior of the nominal and sub-V_t inverters.

The optimal logic depth for circuits operating at a nominal supply voltage is 8 FO4 delays at 3.6 GHz in a 100 nm technology [315]. However, the logic paths of sub-threshold circuits have much higher delay and operate at much lower maximum frequencies. An analysis of the minimum clock period, skew, and the timing properties of the register is used to determine the upper bound (critical path) and lower bound (shortest path) of the allowed delay. Therefore, the timing requirements given by (7.9) and (7.10) are re-written as, respectively, (7.11) and (7.12), where T_{min} is the minimum allowed clock period and t_{logic} is the propagation delay through the combinational logic. The longest and shortest delay using the nominal buffers operating in sub-threshold is determined as, respectively, 12 FO4 and 6 FO4 for supply voltages ranging from 200 mV to 250 mV, and for the optimized sub-V_t buffers as 5 FO4 and 2 FO4 for the same voltage range. For the 130 nm CMOS process used in this paper, the FO4 delay at a sub-threshold voltage of 200 mV and 250 mV is, respectively, 103.2

ns and 36.25 ns for the optimized inverter.

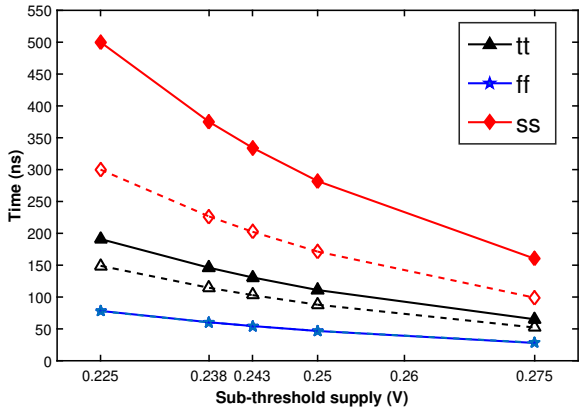
$$t_{logic,critical} \leq T_{min} + \delta_{adjusted} - t_{clk-to-q,tt} - t_{s,tt} \tag{7.11}$$

$$t_{logic,shortest} > \delta_{adjusted} - t_{clk-to-q,tt} + t_{h,tt} \tag{7.12}$$

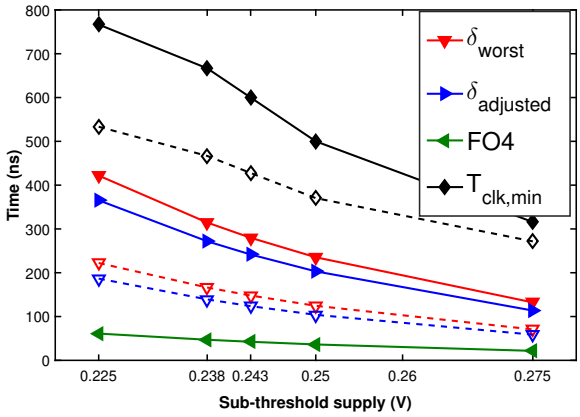
7.3.3 Sensitivity of Clock Network to Power Supply Variation

The effect of up to ±10% power supply variation on the clock network (schematic shown in Fig. 7.7) is analyzed for a sub-threshold supply voltage of 250 mV, with results provided in Fig. 7.9. The variation in insertion delay, FO4 delay, skew, and minimum clock period is characterized using unoptimized nominal and optimized sub-V_t buffers, both operating in sub-threshold. Improvement in the insertion delay, clock skew, and clock period is observed when sub-V_t buffers are used as compared to a clock distribution network using buffers designed for nominal operation. The insertion delay is reduced to 0.79× when sub-V_t buffers are used (for the typical corner) as compared and normalized to the insertion delay of nominal buffers operating at 250 mV. The insertion delay using nominal and sub-V_t buffers increases to, respectively, 1.72× and 1.34×, normalized to the insertion delay of a nominal buffer operating at 250 mV, when the supply voltage is reduced to 225 mV. In addition, the minimum clock period is reduced to 0.74× when optimized sub-V_t buffers are used as compared to non-optimized nominal buffers for a 250 mV supply. The use of optimized buffers

reduces the skew to 0.52× as compared to the skew with non-optimized buffers. At a supply voltage of 250 mV, the skew of the clock-path with nominal and sub-V_t buffers is, respectively, 5.6 FO4 and 2.9 FO4.



(a)



(b)

Fig. 7.9: Variation in (a) insertion delay, and (b) skew, minimum clock period, and FO4 delay. The clock network with nominal and sub-V_t buffers is represented as, respectively, solid and dashed lines.

A minimum clock period of 500 ns is required to meet the timing constraints of the flip-flops at a 250 mV supply voltage. The requirements of 1) a minimum permitted clock period and 2) preventing race conditions, are used to analyze the maximum

allowed variation in power supply voltage. In addition, power supply variation is analyzed by considering three different PDN configurations: *Case 1*) a single power distribution network that delivers current to the clock buffers, registers, and combinational logic circuits (baseline), *Case 2*) two power domains, where one power network is dedicated to the combinational logic circuits and the other power network supplies current to the clock buffers and registers, and *Case 3*) two power domains, where one power network supplies current to the combinational logic circuits and clock buffers and the other power network supplies current to the registers. The separate PDNs for Case 2 and 3 are either operated at two different sub-threshold voltages with a minimum voltage of 250 mV or the circuit blocks within the PDNs are more robust to noise at 250 mV and, therefore, the voltage is kept the same for both PDNs, as done for the analysis in this section.

Table 7.1: Maximum tolerable noise of clock network for three PDN configurations using nominal and sub-V_t buffers at 250 mV.

	Maximum tolerable noise (per cent of 250 mV)		
	Case 1	Case 2	Case 3
Nominal buffer	2%	4%	4%
Sub-V_t buffer	7%	8%	10%

The simulation results are provided in Table. 7.1, where the maximum tolerable noise is listed for the three different PDN configurations with both nominal and sub-V_t buffers. The power overhead of the VR is not considered as analyzing the effect of power supply noise on the clock network is the primary objective. Based on

simulation results, the use of a single PDN with nominal buffers tolerates up to 2% supply noise variation, while the use of sub-V_t buffers with a single PDN increases the noise tolerance to 7%. In addition, when using nominal buffers, the tolerance to noise increases by up to 4% when separate PDNs are used for either combinational logic only (Case 2) or nominal buffers with combinational logic together (Case 3). The use of sub-V_t buffers with a separate PDN for combinational logic provides the maximum noise tolerance of 10%, which indicates a benefit of including multiple power domains when considering sub-threshold clock delivery. Therefore, the robustness and performance of the clock distribution network is improved when buffers optimized for sub-V_t are used for a supply voltage of 250 mV. In addition, for ultra-low voltage circuits, a multi-domain power distribution network is beneficial to meet the timing constraints of the circuit when in the presence of supply and process variation. The proposed technique is also applicable to multi-core systems, where each core includes dedicated OCVRs.

7.4 Selection of Optimum Supply and Threshold Voltage

The procedure to optimize the supply and threshold voltages of an inverter is described by Algorithm 1. The problem formulation and description of the parameters

required for the algorithm are provided in Section 7.4.1. The expansion of the algorithm for more complex circuits that include a larger number of gates is described in Section 7.4.2.

7.4.1 Problem Formulation

The operating characteristics of an integrated circuit are affected by changes in the supply and threshold voltages. A reduction in supply voltage reduces both the noise margins and the f_{max}. However, a reduction in threshold voltage reduces the noise margins, but increases the f_{max}. The changes in the operating characteristics of the circuit due to variation in supply and threshold voltage are accounted for by the proposed optimization model described in this paper. Note that the supply voltage, threshold voltage, noise margin, and operating frequency are all of different units. Therefore, a common unitless representation of all variables is considered for the optimization model. One unit of supply voltage (or threshold voltage or noise margin) and one unit of operating frequency are set to, respectively, 100 mV and 100 MHz. In addition, the parameters implemented in the algorithm are considered based on a single CMOS inverter.

7.4.1.a Definition 1 - NPP

A design metric is introduced to optimize the supply voltage algorithmically, which is described as the noise performance product (NPP). The NPP is the product of the

Algorithm 1 Supply and Threshold Voltage Optimization

$$
\begin{aligned}
\min \quad & x \cdot NPP_{Vdd} + y \cdot NPP_{Vt} \\
\text{s.t.} \quad & N_{Vdd} \cdot x + N_{Vt} \cdot y \geq N_m \\
& P_{Vdd} \cdot x - P_{Vt} \cdot y \geq P_m \\
& V_{t,min} = V_{t,min} + \Gamma \cdot V_{dd,min} \\
& V_{dd,max} \geq x \geq V_{dd,min} \\
& V_{t,max} \geq y \geq V_{t,min} \\
& x, y \geq 0
\end{aligned}
$$

where,
x = number of units of supply voltage,
y = number of units of threshold voltage,
N_m= minimum required noise margin,
P_m= minimum required operating frequency,
$V_{dd,min}$ and $V_{dd,max}$= minimum and maximum allowed supply voltage,
$V_{t,min}$ and $V_{t,max}$= minimum and maximum allowed threshold voltage, and
Γ= fitting parameter that accounts for the change in threshold voltage for variation in supply voltage

NM_{avg} and the f_{max}. The change (reduction or increase) in the noise performance product as the supply and threshold voltages are modified is given by, respectively, NPP_{Vdd} and NPP_{Vt}. For a given technology and supply voltage, a higher value of NPP is desired to meet the noise constraints and frequency requirements of the circuit.

7.4.1.b Definition 2 - N_{Vdd}

A change (reduction or increase) in NM_{avg} for each unit of scaling of the supply voltage V_{dd}, where smaller noise margins are expected at lower V_{dd}. For example,

an N_{Vdd} of 0.5 implies the NM_{avg} is reduced by 50 mV when the supply voltage is reduced by 100 mV.

7.4.1.c Definition 3 - N_{Vt}

A change in NM_{avg} for each unit of scaling of the threshold voltage V_t, where a reduction is expected with lower V_t.

7.4.1.d Definition 4 - P_{Vdd}

A change in the f_{max} for each unit of scaling of the supply voltage V_{dd}, where a reduction is expected at lower V_{dd}. For example, a P_{Vdd} of 0.5 implies the maximum operating frequency is reduced by 50 MHz when the supply voltage is scaled by 100 mV.

7.4.1.e Definition 5 - P_{Vt}

A change in f_{max} for each unit of scaling of the threshold voltage V_t, where an increase is expected for lower V_t.

The objective function accounts for the reduction in the noise performance product due to the scaling of the supply and threshold voltages. The first constraint limits the NM_{avg} to at least equal or greater than N_m, while subject to the reduction in noise margins due to supply and threshold voltage scaling. The algorithm will, therefore, not return any determined supply and threshold voltage values that result

in an NM_{avg} that is less than the minimum required noise margin for the circuit. The second constraint limits the f_{max} to at least equal or greater than P_m, while subject to a reduction in operating frequency due to supply voltage scaling. Note that a negative value for P_{Vt} is assumed since f_{max} increases as the threshold voltage is reduced. The third and fourth constraint define the ranges of, respectively, supply and threshold voltage. Note that the minimum and maximum values of allowed supply and threshold voltages are determined based on the implemented fabrication technology and application. The reduction in supply and threshold voltages reduce the value of, respectively, NPP_{Vdd} and NPP_{Vt}. The algorithm minimizes the reduction in the NPP metric and provides the optimum supply voltage for a target operating frequency and noise margin for a target range of threshold voltages. In addition, the change in the threshold voltage due to variation in the supply voltage is also included in the model.

7.4.2 Expansion of the Algorithm for Larger Circuit Blocks

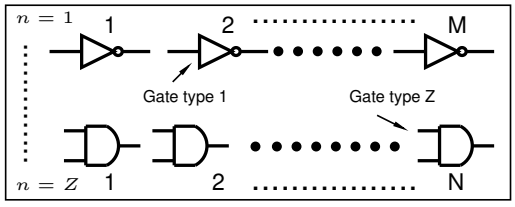


Fig. 7.10: Expanding Algorithm 1 for multi-gate circuits.

The algorithm is applied to a circuit block with a large number of gates as shown in Fig.7.10. There are a total of M and N number of, respectively, NOT and AND gates

in the circuit. Note that only two types of gates are implemented in this particular circuit block. The objective function is rewritten for the two types of gates as given by (7.13), where the first constraint for noise margins (N_{Vdd} and N_{Vt}) and the second constraint for maximum frequency (P_{Vdd} and P_{Vt}) are modified.

$$\begin{aligned} \min \quad & x \cdot [NPP_{Vdd}(\text{NOT}) \cdot M + NPP_{Vdd}(\text{AND}) \cdot N] \\ & + y \cdot [NPP_{Vt}(\text{NOT}) \cdot M + NPP_{Vt}(\text{AND}) \cdot N] \end{aligned} \tag{7.13}$$

s.t.

$$\begin{aligned} [N_{Vdd}(\text{NOT}) \cdot M + N_{Vdd}(\text{AND}) \cdot N] \cdot x + [N_{Vt}(\text{NOT}) \cdot M + \\ N_{Vt}(\text{AND}) \cdot N] \cdot y \geq [N_m(\text{NOT}) \cdot M + N_m(\text{AND}) \cdot N] \\ [P_{Vdd}(\text{NOT}) \cdot M + P_{Vdd}(\text{AND}) \cdot N] \cdot x - [P_{Vt}(\text{NOT}) \cdot M + \\ P_{Vt}(\text{AND}) \cdot N] \cdot y \geq [P_m(\text{NOT}) \cdot M + P_m(\text{AND}) \cdot N] \end{aligned}$$

The supply and threshold voltage selection procedure is also generalized for Z types of gate in a circuit as described by Algorithm 2, where the number of gate types n ranges from 1 to Z. Note that an equal N number of gates for each type of logic is assumed. In addition, a similar supply voltage is assumed for all gates, and a similar threshold voltage is assumed for all transistors.

Algorithm 2 Modification of Algorithm 1 for multi-gate circuits

$$\begin{aligned}
\min \quad & x \cdot \sum_{n=1}^{Z} NPP_{Vdd}(n) \cdot N(n) + y \cdot \sum_{n=1}^{Z} NPP_{Vt}(n) \cdot N(n) \\
\text{s.t.} \quad & x \cdot \sum_{n=1}^{Z} N_{Vdd} \cdot N(n) + y \cdot \sum_{n=1}^{Z} N_{Vt} \cdot N(n) \geq \sum_{n=1}^{Z} N_m \cdot N(n) \\
& x \cdot \sum_{n=1}^{Z} P_{Vdd} \cdot N(n) - y \cdot \sum_{n=1}^{Z} P_{Vt} \cdot N(n) \geq \sum_{n=1}^{Z} P_m \cdot N(n) \\
& V_{t,min} = V_{t,min} + \Gamma \cdot V_{dd,min} \\
& V_{dd,max} \geq x \geq V_{dd,min} \\
& V_{t,max} \geq y \geq V_{t,min} \\
& x, y \geq 0
\end{aligned}$$

7.5 Simulation Results and Analysis

7.5.1 Method of Evaluation

The proposed algorithm is evaluated using SPICE simulation in both a 130 nm and 45 nm technology node. The parameters determined through execution of Algorithm 1 for a 130 nm technology are used in simulation to compare NM_{avg} and f_{max} for both a 130 nm and 45 nm technology to analyze the effect of scaling on voltage selection. The supply and threshold voltages for both nodes are optimized by Algorithm 1 for the target noise margins and frequencies listed in Table 7.2. The $V_{dd,min}$, $V_{dd,max}$, $V_{t,min}$, and $V_{t,max}$ are chosen approximately based on the minimum and maximum allowable voltages in a 130 nm technology. Note that without the application of body

biasing, the range of threshold voltages in a 130 nm technology for PMOS and NMOS transistors is, respectively, $-0.3V \geq V_{t,p} \geq -0.435V$ and $0.45V \geq V_{t,n} \geq 0.35V$. The NM_{avg} and f_{max}, which are shown in Fig. 3.1, are used to calculate the noise performance product for the range of supply voltages between 0.2 V and 1.2 V. The average change in noise margins and f_{max} due to supply and threshold voltage variations is used to calculate the values of NPP_{Vdd}, NPP_{Vt}, N_{Vdd}, N_{vt}, P_{Vdd}, and P_{Vt}. The N_m and P_m are design and technology dependent parameters. Note that the use of the average value of P_{Vdd} reduces computational complexity at the cost of accuracy for the given range of supply voltages (0.2 V to 1.2 V), while the use of a smaller supply voltage range improves the accuracy (see f_{max} in Fig. 3.1).

7.5.2 Comparison of the Algorithm with Simulation Results

The required parameters to evaluate the algorithm are extracted using SPICE simulation. The average variation in NPP_{Vdd} (NPP_{Vt}) for a reduction in V_{dd} (V_t) by one unit is 3.484 (1.02073). The reduction (increase) in P_{Vdd} (P_{Vt}) for scaling of V_{dd} (V_t) by one unit is 2.49667 (2.91767). The reduction in N_{Vdd} (N_{Vt}) for scaling of V_{dd} (V_t) by one unit is 0.23 (0.40). In addition, Γ is determined as 0.01862 by characterizing the change in threshold voltage for variation in supply voltage for the 130 nm technology. The parameters for the algorithm described in Section 7.4 are used to solve for x and y for the six different test cases listed in Table 7.2, where the optimum supply voltage is obtained for a given maximum and minimum range

of threshold voltages and for the required noise margins and operating frequency. A similar range of possible supply voltages (200 mV $\leq V_{dd} \leq$ 1200 mV) is used for all test cases, while different ranges of possible threshold voltages are used to determine the optimum supply and threshold voltages.

The optimum V_{dd} and V_t listed in Table 7.2 are used to determine the f_{max} and NM_{avg} through SPICE simulation, which are listed in Table 7.3. The noise margins are determined using the voltage transfer curve of a CMOS inverter through SPICE simulation. Body biasing is used to tune the threshold voltages to the values obtained from execution of Algorithm 1. The noise margin and maximum frequency obtained from SPICE simulation are compared with the constraints used in Algorithm 1. The optimization of V_{dd} and V_t results in up to 14% error for the six test cases considered (see Tables 7.2 through 7.4), while maintaining the NM_{avg} provided as a constraint, and exhibits an error from the targeted operating frequency of up to 8%.

Test Cases 2 and 6 have similar constraints, but with differences in the range of possible threshold voltages of, respectively, 400 mV $\leq V_t \leq$ 420 mV and 300 mV $\leq V_t \leq$ 700 mV. However, Test 2 requires a higher supply voltage as compared to Test 6, as the minimum threshold voltage for Test 2 is larger than the minimum threshold voltage for Test 6. In addition, despite the use of different supply and threshold voltages for Test 2 and 6, the SPICE simulation for each case remains within 0.86× to 1.13× and 1.05× to 1.08× (normalized to results from the model for each case)

of, respectively, the expected NM_{avg} and expected f_{max}. A similar constraint for the noise margins is applied in Tests 1 and 5. However, a larger value for the performance constraint is applied in Test 1 as compared to Test 5, which results in a larger supply voltage for Test Case 1. Additionally, Tests 3 and 4 are constrained by similar values of operating frequency and threshold voltage. However, Test 4 provides a higher optimum voltage as compared to Test 3 due to an increased noise margin constraint.

Table 7.2: Optimization of V_{dd} and V_t for constraints N_m, P_m, V_{dd} range, and V_t range.

Test	N_m **(mV)**	P_m **(MHz)**	V_{dd} **range (mV)**	V_t **range (mV)**	**Optimum** V_{dd} **(mV)**	**Optimum** V_t **(mV)**
1	≥ 400	≥ 350	$200 \leq V_{dd} \leq 1200$	$350 \leq V_t \leq 420$	1009	420
2	≥ 190	≥ 150	$200 \leq V_{dd} \leq 1200$	$400 \leq V_t \leq 420$	527	406
3	≥ 350	≥ 250	$200 \leq V_{dd} \leq 1200$	$350 \leq V_t \leq 420$	791	420
4	≥ 370	≥ 250	$200 \leq V_{dd} \leq 1200$	$350 \leq V_t \leq 420$	878	420
5	≥ 400	≥ 300	$200 \leq V_{dd} \leq 1200$	$250 \leq V_t \leq 500$	870	500
6	≥ 190	≥ 150	$200 \leq V_{dd} \leq 1200$	$300 \leq V_t \leq 700$	418	306

Table 7.3: Characterization of NM_{avg} and f_{max} through SPICE simulation with optimally selected V_{dd} and V_t.

Test	V_{dd} **(mV)**		V_t **(mV)**		NM_{avg} **(mV)**		f_{max} **(MHz)**		**% error** (NM_{avg})	**% error** (f_{max})
	Model	SPICE (applied)	Model	SPICE (applied)	Model	SPICE (determined)	Model	SPICE (determined)		
1	1009	1009	420	419.85	≥ 400	404.71	≥ 350	322	-1.178	8
2	527	527	406	406.05	≥ 190	214.8	≥ 150	162.4	-13.05	-8.267
3	791	791	420	420.9	≥ 350	328.66	≥ 250	245.7	6.09	1.72
4	878	878	420	421.8	≥ 370	362.38	≥ 250	260	2.06	-4
5	870	870	500	501.85	≥ 400	367.355	≥ 300	288	8.16	4
7	418	418	306	306.9	≥ 190	163.4	≥ 150	157.1	14	-4.73

The SPICE results obtained for a 130 nm technology node are compared with the SPICE results for a 45 nm technology, as listed in Table 7.4. Note that similar supply voltage ranges are used in 45 nm, while the threshold voltages (-384 mV for PMOS and 410 mV for NMOS) are kept constant to characterize the difference in simulation results between 130 nm and 45 nm while controlling for changes to the threshold voltage due to increased sensitivity to PVT variation for scaled technologies

[287, 289, 294].

The f_{max} increases when transistor gate length is reduced [316]. The propagation delay is given by (7.14). Assuming the transistor is in saturation, the delay is proportional to the square of the gate length. The μ and L in (7.14) are, respectively, the electron mobility and transistor gate length. Assuming a constant V_{dd}, V_t, and μ, the delay for a 130 nm technology is approximately 8.3× that of a 45 nm inverter. Therefore, the f_{max} in 45 nm is up to 4.3× greater than the maximum frequency of an inverter in 130 nm (for Test Case 4). However, the noise margin in 45 nm is reduced to 0.89× the noise margin of the 130 nm inverter (for Test Case 1).

$$\text{Propagation delay, } t_p \propto \frac{L^2 \times V_{dd}}{\mu(V_{dd} - V_t)^2} \tag{7.14}$$

Table 7.4: Comparison of average noise margin and maximum frequency between 130 nm and 45 nm technologies.

Test	NM_{avg} **(mV)**			**Max. f (MHz)**		
	Model (constraint)	130 nm (determined)	45 nm (determined)	Model (constraint)	130 nm (determined)	45 nm (determined)
1	≥ 400	404.712	359	≥ 350	322	1123.6
2	≥ 190	214.8	205.3	≥ 150	162.4	666.7
3	≥ 350	328.66	304	≥ 250	245.7	1010.1
4	≥ 370	362.38	329	≥ 250	260	1111
5	≥ 400	367.4	326.9	≥ 300	288	1091.7
6	≥ 190	163.4	158.7	≥ 150	157.1	621.1

7.5.3 Feasible Region of the Optimal Solution

The feasible region and optimum solution for selection of the supply and threshold voltages are discussed in this section as a means to analyze the solution obtained from the algorithm. The feasible region and the contour plot representing different

optimum solutions are shown in Fig. 7.11, where the constraints of Test Case 5 are used. The feasible region is bounded by the constraints for noise margins (0.23x + 0.4y = 4), supply voltage (200 mV $\leq V_{dd} \leq$ 1200 mV), and threshold voltage (250 mV $\leq V_t \leq$ 500 mV), which is marked by the shaded region (blue) in Fig. 7.11. Note that x and y corresponds to, respectively, the supply and threshold voltages. The three vertices of the feasible region are marked by A, B, and C, where the numerical values for A, B, and C describe the reduction in the NPP for a reduction in supply and threshold voltage. The optimal solutions from execution of Algorithm 1 for points A, B, and C are approximately equal to, respectively, 35.4, 46.91, and 44.97, where point A exhibits the minimum reduction in NPP. The optimum supply voltage and threshold voltage at point A is, respectively, 870 mV and 500 mV. Therefore, the optimum solution obtained from the algorithm provides the supply and threshold voltages that correspond to a minimal reduction in the noise performance product.

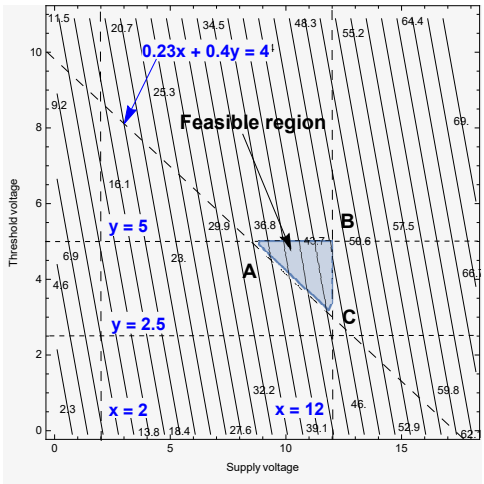


Fig. 7.11: Feasible region of optimum solution for Test 5. The unit value of x and y correspond to, respectively, 100 mV of supply voltage and 100 mV of threshold voltage.

7.6 Summary

In this chapter, a methodology for clock-power co-design is developed for sub-threshold computing. A multi-voltage domain power distribution network is used to deliver current to the components of a clock distribution network operating in sub-threshold. The effect of process and supply voltage variation on the clock network at an operating voltage of 250 mV is analyzed. In addition, noise and performance constrained optimization of the supply voltage is proposed for a given range of threshold voltages, where the variation in noise margins, maximum operating frequency, and threshold voltage are considered.

Chapter 8

Leakage Reuse for Energy Efficient Computing

Energy loss due to leakage current is inevitable in modern processors and system-on-chips (SoCs), which results in energy waste as no computation or data storage is performed with leakage current. As the transistor is scaled and the number of transistors in a circuit increases, a greater amount of leakage current is lost to the ground and substrate nodes of the circuit. The energy loss due to leakage current is more significant in state-of-the-art multi-core systems as cores idle for longer periods of time [26, 317]. The state-of-the-art CPU, GPU, DNN accelerators, SoCs, and battery operated devices consume a significant portion of the total energy as leakage due to an increased number of on-chip computation and memory units as well as

increased idle times while executing a diverse set of applications [26, 110, 120, 317]. A technique to reuse (or recycle) the leakage current of idle core(s) or circuit block(s), consequently, significantly reduces the total energy dissipation of an integrated circuit. In this chapter, circuits, methodologies, and algorithms are developed that permit the reuse of leakage current from idle cores or blocks that operate at a higher voltage (super-threshold supply voltage) to drive the circuits of core(s) or circuit block(s) that operate at an ultra-low supply voltages, including near- or sub-threshold supply voltages.

The remainder of the chapter is as follows. A circuit level description of the proposed leakage reuse technique is provided in Section 8.1, where the rationale and feasibility of the leakage reuse technique are also discussed. The analysis of the energy break-even point of leakage reuse technique is described in Section 8.2. The algorithm and corresponding circuitry that implements the optimum scheduling of idle donor cores for leakage reuse is provided in Section 8.3. The methodology and algorithms for simultaneous implementation of leakage reuse and power gating are discussed in Section 8.4.

8.1 Introduction to Leakage Reuse Technique

The leakage current of the idle cores or circuit blocks operating at a nominal supply voltage is used to drive the circuits operating at a lower supply voltages such as

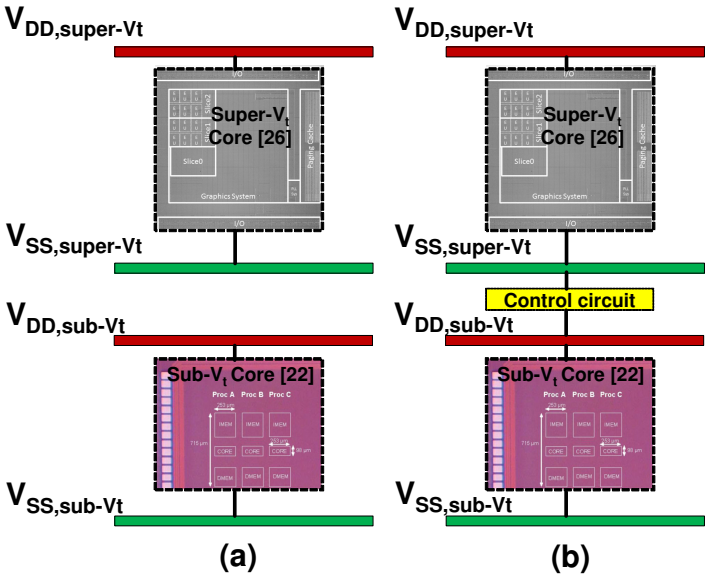


Fig. 8.1: Comparison of a) conventional PDNs with two independent voltage domains, and b) the proposed PDN implementing the leakage reuse technique.

near-threshold or sub-V_t. A simple representation of a conventional power delivery network (PDN) with two independent voltage domains [25, 284] and the proposed PDN implementing the leakage reuse technique is shown in Fig. 8.1, where the current from the super-V_t core reused by the sub-V_t core is regulated by the *Control circuit* block. In addition, the operating flow diagram of the proposed leakage reuse technique is shown in Fig. 8.2, where the super-V_t cores are switched to either normal or leakage reuse mode depending on the workload activity of the super-V_t cores. The developed technique assigns an active super-V_t core to normal operating mode. An idle super-V_t core is, however, switched to leakage reuse mode to supply current to an energy efficient sub-V_t core. Given the growing interest in efficient power management techniques for multi-core systems, the proposed leakage reuse technique provides

improved overall energy efficiency of multi-voltage domain SoCs consisting of both super-V_t (high performance) and sub-V_t (low performance) cores.

The primary advantages of the leakage reuse technique are:

1) A novel leakage reuse method that reduces the total power consumption of a multi-voltage system by utilizing the leakage current of idle super-V_t cores for computation and storage in sub-V_t cores. Separate voltage regulators and power delivery networks (PDNs) are, therefore, not required for the sub-V_t circuits, unlike conventional power delivery to sub-V_t circuits that require an independent and dedicated PDN and expensive voltage regulators [25, 318], and

2) a reduction in the total leakage current of the super-V_t circuits due to power network stacking during idle mode operation without significantly affecting the performance of the super-V_t core(s) operating in active mode.

The proposed method is applicable to any heterogeneous multi-core system implementing dynamic voltage scaling as well as for circuit that integrate disparate technologies with multiple voltag e domains including 3-D integrated systems, multi-core systems composed of hybrid technologies, neuromorphic systems, system-on-chips for deep neural networks, and processing elements implementing multiple clock and voltage domains through globally asynchronous and locally synchronous technique [192, 319].

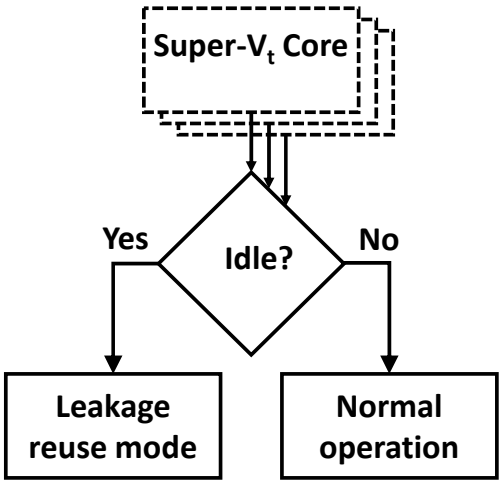


Fig. 8.2: Operating flow diagram of the proposed leakage reuse technique.

8.1.1 Rationale for leakage reuse

The contribution of leakage current to the total on-chip power consumption of microprocessors and system on chips (SoCs) continues to increase due to reductions in the physical dimensions of the transistor, as well as an increase in the transistor density per unit area [28]. In addition, the long idle periods of most battery-operated mobile devices results in leakage current becoming a dominant component of the total power consumption [65], which is given by (8.1).

$$P_{total} = \alpha \cdot V^2 \cdot C_L \cdot f + V \cdot I_{Leakage} \tag{8.1}$$

$$I_{Leakage} = I_{gate} + I_{sub-V_t} + I_{junction} \tag{8.2}$$

The primary components of the leakage current of CMOS devices include 1) gate

leakage, 2) sub-V_t leakage, and 3) junction leakage, as given by (8.2) [320]. The gate leakage as a percent of the total leakage power has significantly increased as the gate length and oxide thickness are reduced, where a 30× increase in gate leakage has been reported when advancing to the next scaled fabrication node [28, 50]. In addition, despite the reduced leakage current as compared to CMOS devices, FinFET devices also consume a significant amount of leakage power as a percentage of the total power [321]. In [321], the ISCAS85 benchmark circuits are implemented with shorted-gate and low-power mode FinFET logic cells, and on average, 13% of the total power consumption is attributed to leakage current. Therefore, as the transistor is scaled and the number of transistors in a circuit increases, a greater amount of leakage current is lost to the ground and substrate nodes of the circuit. The energy loss due to leakage current is more significant in state-of-the-art multi-core systems as cores idle for longer periods of time [26, 317]. A technique to reuse (or recycle) leakage current of idle core(s) or circuit block(s), consequently, significantly reduces the total energy dissipation of an integrated circuit.

As the performance requirements of executing applications varies, the use of both low-performance energy-efficient cores and high-performance cores within a multi-core system is needed to improve the overall energy-efficiency of the SoC. Current heterogeneous multi-core systems integrate high-performance super-V_t cores with energy-efficient cores operating at near- and sub-V_t voltages [104, 273, 322–324]. In addition,

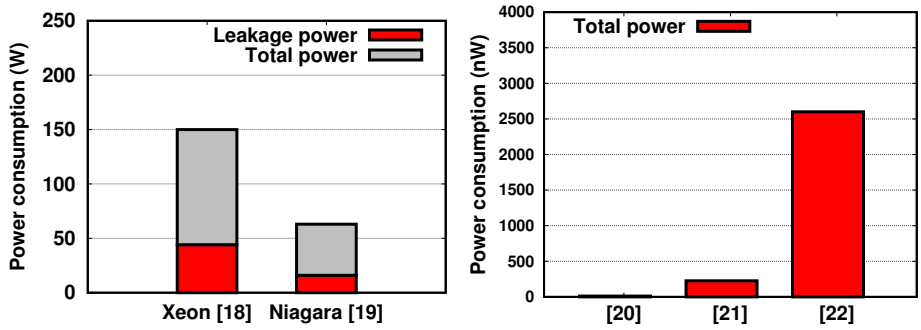


Fig. 8.3: Comparison of (a) total and leakage power consumption of super-V_t cores, and (b) total power consumption of sub-V_t cores.

different voltage and frequency scaling techniques are implemented to improve the energy efficiency of the circuit including adaptive voltage scaling (AVS) [102, 103], dynamic frequency scaling (DFS) [103], dynamic voltage scaling (DVS) [22, 100], dynamic voltage and frequency scaling (DVFS) [100, 104], and dynamic voltage and threshold scaling (DVTS) [98] to operate circuits from the super-V_t to sub-V_t region based on the performance requirements of the executing applications. In addition, microprocessors and processing elements that include a globally asynchronous and locally synchronous (GALS) clocking system achieve greater energy efficiency by operating sub-modules in multiple clock and voltage domains [319, 325]. The leakage current of state-of-the-art microprocessors operating at super-V_t supply voltages provides no computational and storage benefit. In contrast, the computation and storage in sub-V_t circuits is performed by consuming only leakage current.

The total and leakage power consumption of cores and SoCs implemented for both super-V_t and sub-V_t operation are shown in Fig. 8.3(a) and 8.3(b), respectively. The

leakage power consumption of a Xeon Tulsa processor operating at 3.4 GHz with a power supply voltage of 1.25 V is 44.1 W, which is equivalent to 31% of the total power consumption of 150 W as shown in Fig. 8.3 [326]. Similarly, the leakage power consumption of the UltraSPARC T1 Niagara processor is 16 W, 26% of the total power of 63 W when operating at 1.2 GHz and with a 1.2 V supply voltage [327]. Prior work implementing sub-V_t processors indicate ultra low power operation and very low energy dissipation per instruction. A sub-V_t processor implemented in a 130 nm technology for sensory network applications operates at 66 KHz and at a 160 mV supply voltage, resulting in a power consumption of 11 nW [284]. A sub-V_t Phoenix processor implemented in a 180 nm technology for sensing applications operates at 106 KHz and at a 500 mV supply voltage, resulting in a power consumption of 35.4 pW and 226 nW in, respectively, idle and active mode [328]. In addition, a sub-V_t SoC implemented in a 130 nm technology for wireless electrocardiogram (ECG) monitoring, operating at 475 KHz and at a 280 mV supply voltage, consumes 2.6 μW of power [285]. A tiny fraction of the total leakage power dissipation from the processors operating at a nominal supply voltage is, therefore, sufficient to drive an entire sub-V_t processor. Note that the terms nominal and super-V_t are used interchangeably.

8.1.2 Feasibility of Leakage Reuse Technique

Idle core(s) or circuit block(s) are continuously available as varying workloads executing on a multi-core system do not use all hardware resources at any given time. A discussion of the availability of idle cores is provided in Section 8.1.2.a. In addition, the proposed technique recycles the leakage current from idle super-V_t core(s) through voltage stacking, which is described in Section 8.1.2.b.

8.1.2.a Availability of Idle Core(s) and Circuit Block(s)

Most modern low power circuit and power management techniques reduce the amount of energy consumed during idle mode operation of the circuit [26, 124]. Circuit blocks within a core or cores within a multi-core system are assigned to different operating modes (C-states) based on idle activity patterns to improve the performance per watt [26, 120]. Power and clock gating are the most common techniques implemented to reduce the power consumption during idle mode operation of the circuit [110, 119, 123, 125]. In [120], a machine learning based prediction method in conjunction with an OS power management policy is implemented in state-of-the-art multi-core processors to assign cores to a C-state based on the idleness of a given core, where the C-4 state places the core in power gated mode. For the analysis of system idleness, a 3 GHz quad-core processor ran for 10K cycles. Within the 10K cycles, there was not a single instance when all cores were simultaneously active. In

addition, the SPECWeb benchmark suite was executed on a dual-core processor, and similarly, there was no instance when both cores were concurrently active [120].

Per core power gating (PCPG) has been proposed as an effective power management option for multi-core processors along with dynamic voltage and frequency scaling [119]. In [119], core utilization traces are simulated to analyze the use of PCPG on a 2.5 GHz AMD Phenom X4 9850, which includes four cores implemented in a 65 nm technology. The utilization traces for a commercial application server (PHARMA04), two web servers (HCOM10, ECOM3), and a desktop computer (DESKTOP) are used to monitor the activity of the four cores, and again, there is no instance when all four cores are simultaneously active [119]. In addition, the idleness behavior of state-of-the-art processors is characterized using both consumer and CPU-GPU benchmarks including *DirectX9*, *KMeans*, and *Gaussian* from the PCMark and Rodinia suites [26]. The results from the simulation of the benchmark workloads demonstrate that a minimum of 110 multi-cycle idle events per second occurred for a broad range of applications executing on a 16 nm FinFET technology [26]. In this paper, to characterize the CPU utilization of individual cores, simulation of a system running Red Hat Linux and with 48 Intel Xeon CPU cores each operating at 3 GHz is used to execute different applications. At any given time, at least one core remains idle, while all 48 cores are idle for more than 90% of the runtime. Therefore, there are enough idle circuit blocks and cores within any system to allow for the reuse of unused leakage

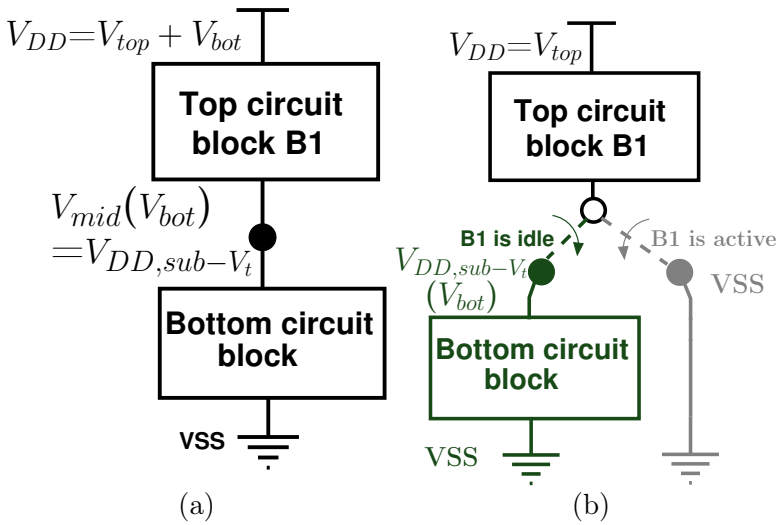


Fig. 8.4: Delivering current to a sub-threshold circuit block through a) voltage stacking technique, where two circuit blocks within a core are vertically stacked regardless of the circuit activity of the super-V_t (top) circuit block, and b) the proposed technique of delivering current to sub-V_t (bottom) circuit blocks, where *only* an idle super-V_t circuit block is stacked.

current [26, 119, 120].

8.1.2.b Charge recycling During Idle Mode

Charge recycling (or voltage stacking) is a technique where the power distribution networks of two circuit blocks within a core or two cores within a multi-core system are vertically stacked, sharing a common path from the supply (V_{DD}) to ground (V_{SS}). The conventional method of voltage stacking is shown in Fig. 8.4 (a), where the circuit block set to a lower absolute voltage is vertically stacked with a circuit block set to a higher absolute voltage regardless of the executing workloads of the higher voltage circuit block. In other words, the two circuit blocks are vertically stacked for the entire operation of the device [21, 68, 174, 175].

Charge recycling has been recently used for logic and memory circuits in 2-D and 3-D integrated circuits to 1) minimize the total power consumption and the peak rush current [55, 67, 68, 175–177], which reduces inductive noise ($L{\cdot}di/dt$), and 2) limit interconnect wear-out due to electromigration (EM) as the current density is reduced [116]. Prior work has shown a 60% reduction in transient noise when voltage stacking is applied to a 3-D IC as compared to a non-stacked 3-D IC [67]. In addition, the stacking of SRAM banks during idle mode, as proposed in [55], reduces the leakage power by 93%. However, there are limitations of implementing the voltage stacking technique as circuit blocks are stacked during both active and idle mode operation. Limitations include 1) the need for regulation of the mid-node voltage (V_{mid} in Fig. 8.4) due to variation in load current [67], 2) variation in voltage due to workload and current imbalances [67], 3) the need for a boosted supply voltage, which increases the power overhead [67, 68], and 4) a reduction in the energy efficiency of voltage stacked circuit blocks as the imbalance in current load between the stacked voltage domains increases [176].

In this research, however, charge recycling is applied to only idle super-V_t cores or circuit blocks. The proposed method increases the effective resistance of the stacked path only during idle mode operation of the super-V_t core and, therefore, reduces the total leakage current through the stacked path. Due to the stacking of only idle super-V_t cores, the proposed approach is not adversely effected like conventional voltage

stacking techniques, where careful regulation is required to compensate for variations in the workload and current imbalances between stacked voltage domains. During the idle mode operation of the super-V_t circuit, the leakage current from the super-V_t circuit block is delivered to the sub-V_t circuit block for computation and storage as shown in Fig. 8.4 (b). Therefore, the super-V_t circuit block remains unaffected during active mode operation while the leakage current from an unused super-V_t circuit block is recycled to deliver power to the sub-V_t circuit block.

8.1.3 Reusing Leakage Current of Super-V_t Cores to Drive Sub-V_t Cores

There are three primary advantages of the proposed leakage reuse technique: 1) A reduction in the total leakage current of the super-V_t cores, 2) a reduction in the total power consumption of the system by leveraging the leakage current of the super-V_t core(s) to supply current to circuits in the sub-V_t core(s), and 3) no separate voltage source is required for the sub-V_t core(s). The system level and circuit level model of the leakage reuse technique is presented in Section 8.1.3.a and 8.1.3.b, respectively. In addition, the control circuit of a super-V_t core consisting of two switches is described in Section 8.1.3.c.

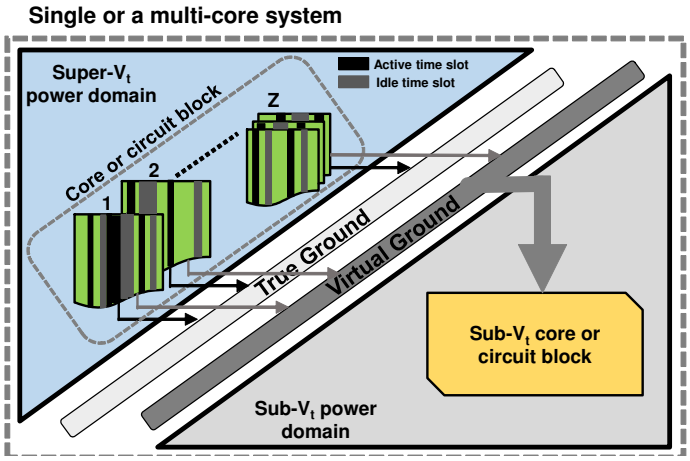


Fig. 8.5: System level model of reusing the leakage current of super-threshold circuits to drive sub-threshold circuits.

8.1.3.a System Level Model of Leakage Reuse Technique

The proposed system level model that accounts for the reuse of the leakage current from idle super-V_t cores or circuit blocks to supply circuits operating in a sub-V_t voltage domain is shown in Fig. 8.5. Due to significant idle states in existing state-of-the-art circuits, as discussed in Section 8.1.2.a, at any given time, at least one super-V_t core (or circuit block) is assumed to operate in idle mode.

The system level model is applicable to a) a single core that includes both a super-V_t and sub-V_t voltage domain such as a system that implements globally asynchronous and locally synchronous clocking and b) a multi-core system with cores operating at both super-V_t and sub-V_t voltages. Therefore, the proposed leakage reuse technique is categorized as either 1) inter-core, where two cores that operate at a super-V_t supply voltage drive a core operating at a sub-V_t supply voltage, or 2) intra-core, where two

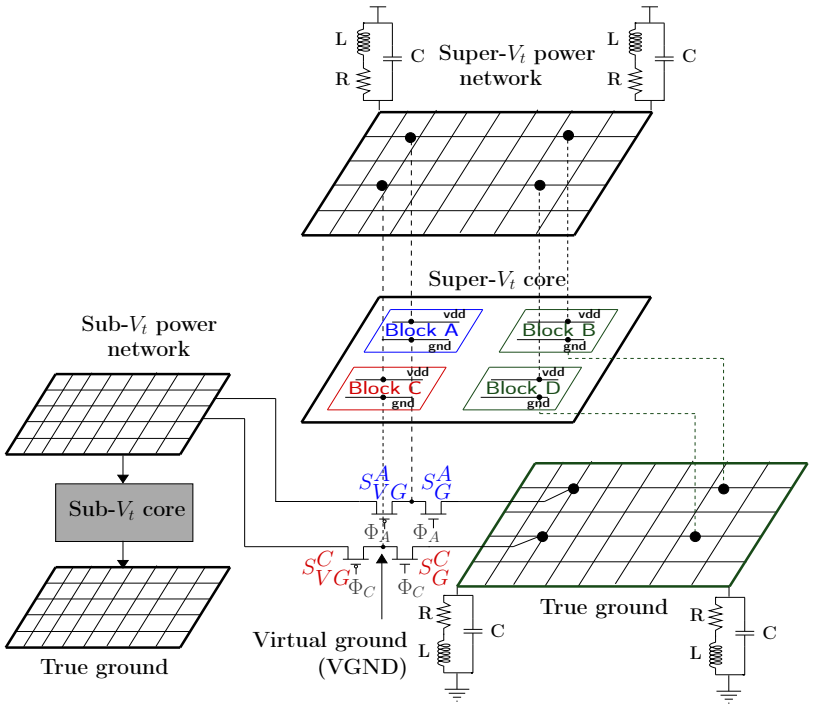


Fig. 8.6: Circuit model of the proposed leakage reuse technique, where the ground distribution network is modified to allow for voltage stacking during idle mode operation of the super-V_t core(s). Blocks A and C supply leakage current to one sub-V_t core.

circuit blocks within a core operate at a super-V_t supply voltage and drive a circuit block operating at a sub-V_t voltage.

8.1.3.b Circuit Model Accounting for Leakage Reuse

Conventionally, the circuits within a core operating in a single voltage domain receive current through a hierarchical power delivery system. In general, a 10% activity factor implies that 10% of all gates switch at any given time. Therefore, both dynamic and leakage power is consumed for 10% of the gates of a core, while 90% of the gates consume only leakage power.

The circuit model used to analyze the leakage reuse technique is shown in Fig. 8.6.

The super-V_t core consists of four functional blocks that are supplied by a conventional hierarchical power delivery system. The ground network is, however, modified to allow for the reuse of the leakage current from the super-V_t circuits. The functional blocks B and D are assumed to be executing a high activity workload and are highly sensitive to ground bounce and are, therefore, directly connected to true ground. The functional blocks A and C are, however, assumed to be performing low activity tasks and are less sensitive to ground bounce noise. In this case, the leakage current from blocks A and C is used to supply the sub-V_t core. The cumulative total width of all transistors within a functional block provides an estimate of the total leakage current of the circuit during idle mode [320]. A continuous power supply to the sub-V_t core exists, since at any given time at least one of the functional blocks (either A or C) is operating in idle mode. Both super-V_t blocks A and C are connected to ground through either of two switches, implemented as one PMOS and one NMOS transistor, where A is connected through S^A_{VG} and S^A_G, and C through S^C_{VG} and S^C_G as shown in Fig. 8.6. Transistor S^A_{VG} (S^C_{VG}) sinks current from block A (C) to the sub-V_t power network through a virtual ground (VGND) during idle mode operation of the blocks, while transistor S^A_G (S^C_G) sinks current from block A (C) to true ground during active mode.

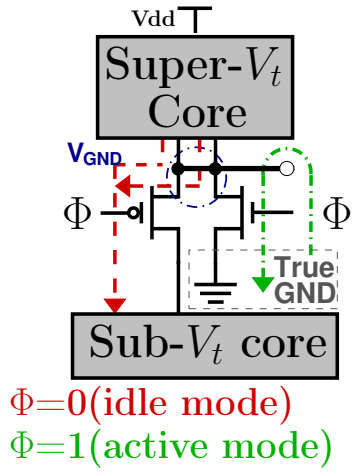


Fig. 8.7: MOS switches to connect the super-V_t core to true ground when active or to the sub-V_t core when idle.

8.1.3.c Control Circuit of the Leakage Reuse Technique

The control circuit block of the proposed leakage reuse technique behaves similar to sleep transistors used for power gating as the two footer MOS transistors are implemented to switch the operating mode of a super-V_t core between normal activity and leakage reuse [28, 64, 116, 118, 119]. Note that during idle mode operation of the circuit, the ground node of the super-V_t cores becomes a virtual ground.

The circuit implementation of the MOS switches is shown in Fig. 8.7. The NMOS transistor connects the virtual ground to the true ground when either circuit block A or C is in active mode. The PMOS transistor connects the virtual ground to the power network of the sub-V_t core when either A or C is idle. The gates of transistors S^A_{VG} and S^A_G in Fig. 8.6 are connected to the control signal Φ_A, while S^C_{VG} and S^C_G are connected to Φ_C. Φ_A and Φ_C are each set to 0 or 1 when the corresponding super-V_t

circuit is operating in either idle mode or active mode, respectively. The transistors S_{VG}^{A} and S_{VG}^{C} must be large enough to sufficiently conduct the leakage current through the virtual ground (sub-V_t power network) within a given clock period and without causing significant resistive drop, while the threshold voltages of transistors S_{G}^{A} and S_{G}^{C} must be large enough to prevent leakage loss during the idle mode operation of the super-V_t cores.

The proposed leakage reuse technique does not replace power gating. Simultaneous implementation of both power gating and leakage reuse is, therefore, possible. A multi-core system that implements both power gating and the leakage reuse technique is more energy efficient than a system that only applies power gating as current is delivered to an energy efficient sub-V_t core without requiring a dedicated PDN with expensive voltage regulators.

8.1.3.d Characterization and Analysis

The proposed leakage reuse technique is evaluated through SPICE simulation in a 45 nm CMOS process. The super-V_t supply voltage is set to 1.2 V, while the target sub-V_t supply voltage is 380 mV. The 45 nm fabrication process includes low-threshold (low-V_t), nominal threshold (nominal-V_t), and high-threshold devices (high-V_t). Low-V_t transistors are used for the circuit blocks operating at a super-V_t supply voltage to improve performance, while nominal-V_t transistors are used for the circuit blocks

operating at a sub-V_t supply voltage to reduce power consumption. In addition, high-V_t NMOS devices are used as sleep transistors S_G^A and S_G^C to reduce the idle power consumption, while low-V_t PMOS transistors are used for the switches connecting to the sub-V_t core. Note that the threshold voltage of a nominal-V_t, low-V_t, and high-V_t NMOS transistor in a 45 nm process is, respectively, 410 mV, 322 mV, and 608 mV.

The leakage reuse technique is evaluated using a chain of inverters (`COI`) and two ISCAS89 benchmarks circuit (`s27` with 19 gates and `s208` with 112 gates). The ISCAS89 benchmark circuits represent super-V_t cores, while separate `COIs` are used to represent both the super-V_t and sub-V_t cores. The super-V_t chain consists of six equally sized inverters, whereas the sub-V_t chain consists of six tapered inverters that are optimized for sub-V_t operation. The ground networks of both the `COI` and the ISCAS89 benchmark circuits are modified to enable a connection with either true ground or the sub-V_t power network through the virtual ground of the super-V_t core.

Three circuit topologies are simulated, which are an implementation of the system level model shown in Figure 8.5 applicable to both inter- and intra-core current reuse. The circuit topologies include:

Topology 1 - Baseline: Two isolated super-V_t circuit blocks representing two individual super-V_t cores and one isolated sub-V_t circuit block (chain of tapered inverters) representing an individual sub-V_t core. Note that there is no connection between the super-V_t and sub-V_t circuit blocks as each includes an independent power supply,

which is the conventional practice.

Topology 2 - Proposed leakage reuse (L.R.): Two isolated super-V_t circuit blocks representing two individual super-V_t cores. The ground network of both the super-V_t circuit blocks is connected through the switching circuit shown in Fig. 8.6 to either true ground or the power network of a sub-V_t circuit block consisting of a chain of tapered inverters. The leakage current of the super-V_t circuit blocks is used to supply power to the sub-V_t circuit block.

Topology 3 - Voltage stacking (V.S.): Two isolated super-V_t circuit blocks representing two individual super-V_t cores, where the ground network of both the super-V_t circuit blocks is directly connected with the power network of a sub-V_t circuit block consisting of a chain of tapered inverters. In [176], stacked voltage domains within the same core are implemented for implicit down conversion of the voltage of the power supply of the near-threshold circuits, which are placed at the bottom of the stacked topology. In this work, the technique proposed in [176] is extended to deliver power to sub-V_t circuits as shown in Fig. 8.4 (a), which is then compared with the proposed leakage reuse technique.

Similar input signals and output capacitive loads are applied to all three topologies, which ensures that the active and idle mode transitions are equally applied to the three configurations. The power supply and ground networks of all three circuit topologies are represented with equivalent electrical parameters obtained from the V_{cc}

Table 8.1: Active area of the chain of inverters (`COI`), `s27`, and `s208` circuits for the three analyzed topologies.

		Area of two super-V_t circuit blocks (μm²)	**Switch area (μm²)**	**Area of sub-V_t circuit block (μm²)**	**Peak noise on V_{SS} as a percent of rail-to-rail voltage (1.2 V)**
COI	Baseline	2.331	N/A	7.56	< 5%
	Leakage reuse	2.331	0.35 (15% of Super-V_t COI) S_{VG}:S_G of 1.5 μm: 2 μm	7.56	< 5%
	Voltage stacking	2.331	N/A	7.56	< 5%
s27	Baseline	6.73	N/A	21	< 5%
	Leakage reuse	6.73	0.31 (4.6% of Super-V_t s27) S_{VG}:S_G of 3 μm: 0.1 μm	21	< 5%
	Voltage stacking	6.73	N/A	21	< 5%
s208	Baseline	21.9	N/A	21	< 5%
	Leakage reuse	21.9	0.6 (2.7% of Super-V_t s208) S_{VG}:S_G of 5 μm: 1 μm	21	< 5%
	Voltage stacking	21.9	N/A	21	< 5%

and V_{SS} pins of a model of the DIP-40 package to analyze the transient noise induced on the power and ground networks. The resistance R, inductance L, and capacitance C of the pins are, respectively, 0.217 Ω, 8.18 nH, and 5.32 pF.

The voltage stacking technique is not directly comparable to the leakage reuse technique as the circuit blocks in a voltage stacked system are continuously stacked for the entire duration of the operation of the circuit. In contrast, the leakage reuse technique stacks two circuit blocks within a core or in two different cores within a multi-core system only during the idle mode operation of the circuits operating at a super-V_t supply voltage.

The three power delivery topologies are implemented on a `COI` and the `s27` and `s208` benchmark circuits operating at 5 MHz, where the active area of the super-V_t and sub-V_t circuit blocks is optimized for each of the three benchmarks circuits to obtain a sub-V_t voltage of 380 mV. The total active area for the super-V_t circuit

blocks, sub-V_t circuit blocks, and switches is listed in Table 8.1. The same circuit area is occupied by each topology, except for the leakage reuse technique, where an additional area is required for the switches. Therefore, iso-frequency and iso-area analysis is performed across the three topologies.

Due to the large reduction in the drive strength of the transistors when in sub-V_t operation, the transistor widths are increased to provide sufficient drive current to charge and discharge loads, where the sub-V_t circuit block is 3.24× and 3.12× the total area of the two super-V_t circuit blocks for, respectively, a chain of inverters and the s27 benchmark circuit. However, the area of the sub-V_t circuit block is 0.96× the size of the two s208 circuit blocks as the increased area of s208 (21.9 μm^2) provides sufficient current to drive the sub-V_t circuit block (21 μm^2) without further increasing the size of the transistors.

The voltage on the virtual ground node V_{GND}, which is the supply voltage of the sub-V_t block generated by the leakage current of the super-V_t core, is dependent upon several circuit parameters including 1) the ratio of the area of the super-V_t circuit block to the area of the sub-V_t circuit block, which is directly correlated to the current ratio, 2) the total leakage current of the super-V_t circuit block, 3) the on and off current through the switches, which are partly dependent on the threshold voltage, gate voltage, and the dimensions of the transistors, and 4) the total current required by the sub-V_t circuit block. For a set size of the super-V_t and sub-V_t cores,

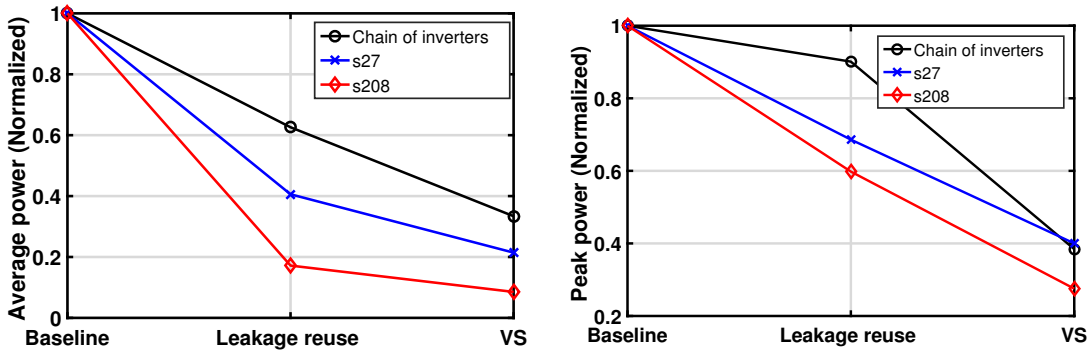
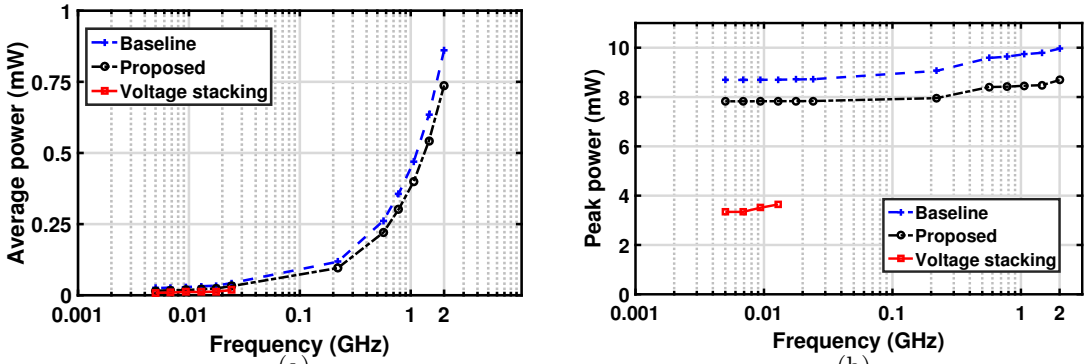


Fig. 8.8: Characterization of a) average power consumption per cycle, and b) peak power consumption of the `COI`, `s27`, and `s208` benchmark circuits at 5 MHz.

the width of the switches acts as a controlling knob that results in a trade-off between the voltage level at the VGND node and the total area and power consumption. An analysis of the effect of switch size is, therefore, performed when the super-V_t core is active, as described in Section 8.1.3.e.

Characterization of Power Consumption

The average power consumption per cycle and peak instantaneous power consumption of the multi-voltage system, which includes the super-V_t and sub-V_t circuit blocks, for the three circuit topologies described in Section 8.1.3.d are shown in, respectively, Figs. 8.8(a) and 8.8(b). The average (instantaneous peak) power consumption of the baseline is 25.24 µW (8.69 mW), 15.47 µW (4.154 mW), and 112.4 µW (7.636 mW) for, respectively, the `COI`, `s27`, and `s208` benchmark circuits. The average (peak) power consumption of the leakage reuse technique is 0.63× (0.9×), 0.41× (0.7×), and 0.17× (0.6×) that of the baseline for, respectively, `COI`, `s27`, and `s208`. The reduced



(a) (b)

Fig. 8.9: Characterization of a) average power consumption per cycle and b) peak power consumption for 5 MHz to 2 GHz operation of the super-V_t circuit blocks.

power consumption due to implementing the leakage reuse technique is, therefore, greater as the circuit size increases. However, the leakage reuse technique consumes more power than the voltage stacking technique as the super-V_t and sub-V_t circuits are stacked for the entire duration of circuit operation when voltage stacking is implemented. The leakage reuse technique applied to `s208` consumes an average and peak power of, respectively, 2.02× and 2.17× that of the voltage stacking technique.

The circuit blocks operating at a super-V_t supply voltage are characterized for both average and peak power consumption at different operating frequencies up to 2 GHz. The chain of inverters operating at a super-V_t supply voltage of 1.2 V is characterized between 5 MHz and 2 GHz using the baseline, leakage reuse, and voltage stacking techniques. The results from simulation of the average and peak power consumption at different operating frequencies is shown in, respectively, Figs. 8.9(a) and 8.9(b). At all frequencies, the average and peak power consumption of the leakage reuse technique is less than the average and peak power consumption of the

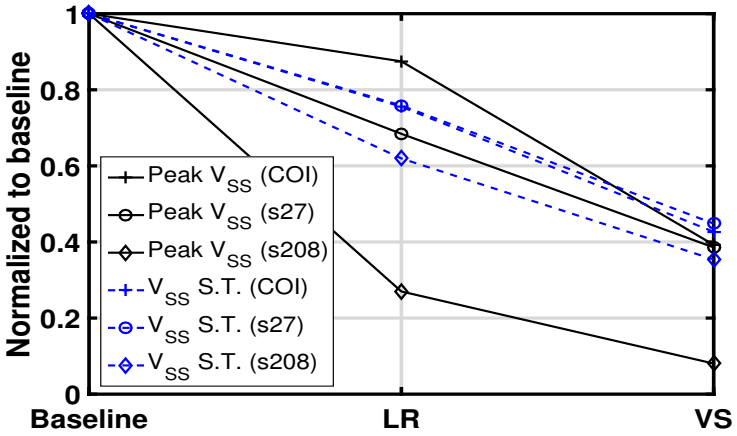


Fig. 8.10: Characterization of the peak V_{SS} bounce and the settling time (S.T.) of the peak V_{SS}.

baseline. In addition, the leakage reuse technique provides a greater reduction in the average and peak power consumption at higher frequencies as compared to the baseline technique. Therefore, a sufficient amount of leakage current is supplied to the sub-V_t core(s) during the idle mode operation of the nominal cores for a range of operating frequencies. Although voltage stacking consumes less power as compared to both leakage reuse and the baseline, the voltage stacking technique is limited to frequencies no greater than 25 MHz as the super-V_t circuit does not provide correct output at higher frequencies due to significant noise on the VGND node.

Characterization of Noise on True and Virtual Ground

The voltage drop on the super-V_t power network is within ±5% of the nominal voltage of 1.2 V for the three circuit topologies described in Section 8.1.3.d. However, the voltage noise on the ground network exceeds ±5% of the ideal ground voltage of 0 V. The results from the characterization of the peak ground voltage bounce on the

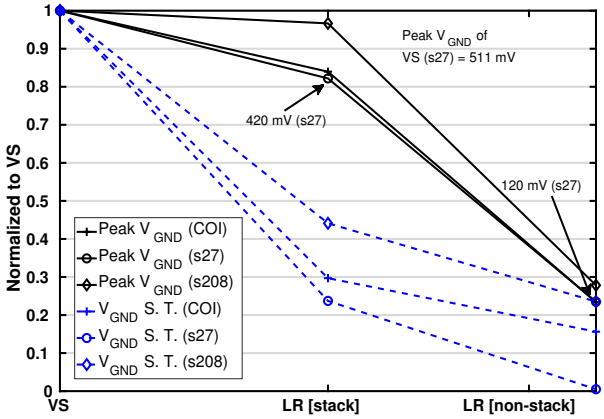


Fig. 8.11: Characterization of the peak V_{GND} bounce and settling time of the V_{GND}, where switches for LR are sized for 10% noise on the stacked and non-stacked V_{GND}.

VSS node and the settling time of V_{SS} (the time required to settle within ±5% of the ideal ground voltage) are shown in Fig. 8.10, where the results obtained for the leakage reuse and voltage stacking techniques are normalized to the results obtained for the baseline. The leakage reuse technique exhibits a reduced peak noise on the VSS node and a reduced settling time of V_{SS} as compared to the baseline for the chain of inverters (`COI`), `s27`, and `s208` circuits. The implementation of the leakage reuse technique on `s27` (`s208`) reduces the peak noise on VSS and the V_{SS} settling time to, respectively, 0.68× (0.29×) and 0.44× (0.37×) that of the baseline. The peak voltage noise on the true ground node VSS (settling time of V_{SS}) for the baseline is 30.07 mV (56 ns), 17.66 mV (110 ns), and 85.82 mV (56.5 ns) for, respectively, `COI`, `s27`, and `s208`.

Unlike the voltage stacking method, the leakage reuse technique allows the stacking of only idle super-V_t circuit blocks with the sub-V_t circuit block, while connecting

the active super-V_t circuit blocks to true ground. However, the noise transients propagate between the true ground of the active circuit blocks and the virtual ground of the idle circuit blocks. For the leakage reuse technique, the super-V_t active and idle circuit blocks are described as, respectively, non-stacked and stacked. The voltage bounce on the virtual ground node of the super-V_t core is analyzed by characterizing the peak voltage noise and settling time (the time required to settle within $\pm 5\%$ of the steady state V_{GND}) of the `COI`, `s27`, and `s208` circuits each implemented with the leakage reuse (stacked and non-stacked) and voltage stacking techniques. The results from characterization of the peak V_{GND} noise and the V_{GND} settling time are shown in Fig. 8.11, where the leakage reuse technique is normalized to the voltage stacking technique. The peak V_{GND} (V_{GND} settling time) of the voltage stacking technique is 513.3 mV (64 ns), 511 mV (105 ns), and 431.4 mV (34 ns) for, respectively, the `COI`, `s27`, and `s208` circuits.

The steady-state voltage V_{GND} on the VGND node is the supply voltage of the sub-V_t circuit, which is set to 380 mV. The peak V_{GND} bounce and V_{GND} settling time of the active circuit blocks when implementing the leakage reuse technique (non-stacked) are less than both the voltage stacking technique and the idle circuit blocks implemented with the leakage reuse technique (stacked). In all cases, the voltage bounce on VGND for both the active and idle circuit blocks does not exceed 10% of the rail-to-rail voltage of 1.2 V for the super-V_t V_{DD} and 380 mV for the sub-V_t V_{DD}.

The peak voltage noise on VGND (V_{GND} settling time) for the leakage reuse technique implemented on the `s208` circuit is reduced to 0.97× (0.4×) and 0.28× (0.23×) in, respectively, the stacked and non-stack mode, as compared and normalized to the voltage stacking technique.

The peak voltage noise on the virtual ground node VGND, peak voltage noise on the true ground node VSS, and the $V_{DD,sub-V_t}$ are characterized in the *tt*, *ff*, and *ss* process corners for both the leakage reuse and voltage stacking techniques. The worst-case process variation, taken as the larger of the two values of $\frac{|tt-ff|}{tt}\times 100\%$ and $\frac{|tt-ss|}{tt}\times 100\%$, is determined for the `s208` benchmark circuit. The leakage reuse technique exhibits a worst-case variation in the peak V_{GND} voltage noise for the stacked topology, peak V_{GND} voltage noise for the non-stacked topology, peak V_{SS} voltage noise, and sub-V_t V_{DD} noise of, respectively, 0.82%, 5.82%, 5.05%, and 1.54%. The voltage stacking technique exhibits a worst-case variation of 9.6%, 56.33%, and 2.05% in, respectively, peak V_{GND} noise, peak V_{SS} noise, and sub-V_t V_{DD}.

8.1.3.e Characterization of Switch Size Used for Implementation of the Leakage Reuse Technique

A circuit implementing the leakage reuse technique is sensitive to the dimensions of the transistors S_G and S_{VG} used as switches. The size of the switches impacts both the virtual and true ground nodes and, therefore, the power consumption and performance as a large instantaneous transient current is induced when the switches

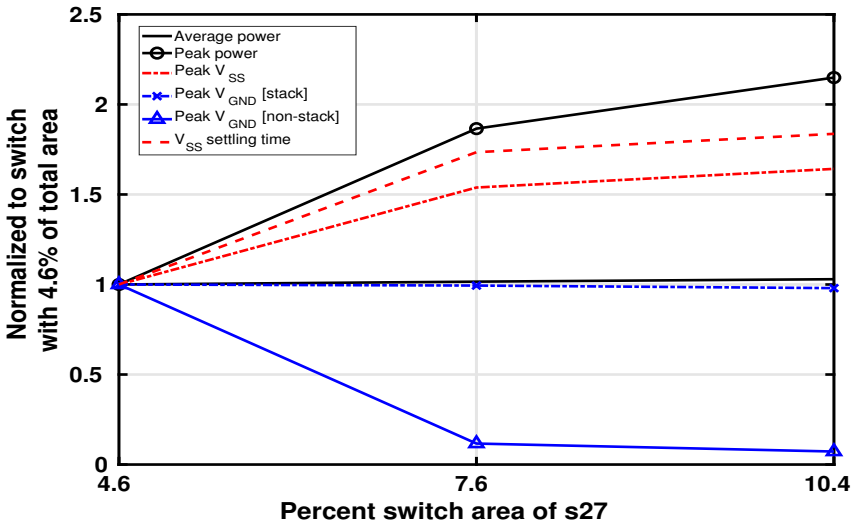


Fig. 8.12: Variation in peak power, average power, and settling time for s27 as a function of switch size.

are turned on. The leakage reuse technique implemented on the s27 benchmark circuit is characterized for three different switch sizes, with results as shown in Fig. 8.12. The three implemented switches occupy 4.6%, 7.6%, and 10.4% of the total area of the two s27 circuit blocks operating at a super-V_t supply voltage. As indicated from the results of the analysis, both the total and peak power consumption increase with larger switches. The switch occupying 10.4% of the area consumes an average and peak power of, respectively, 1.03× and 2.15× that of the switch occupying 4.6% of the total area. However, switches occupying 10.4% of the total area reduce the peak noise on the virtual ground node VGND to 0.07× and 0.98× in, respectively, the non-stacked and stacked mode, as compared to switches occupying 4.6% of the total area.

Due to the increased influx current (initial current) through the switches, the peak

Table 8.2: Trade-off between the percentage noise permitted on the VGND node, the total power consumption, and the area in the non-stacked mode. Power consumption and area are normalized to the baseline topology (independent power delivery networks for the super and sub-V_t cores) for the `COI`, `s27`, `s208` circuits.

		Permitted noise on VGND [non-stack] as compared to ideal ground		
		5%	10%	15%
Total power normalized to baseline	COI	0.642	0.635	0.616
	s27	0.410	0.409	0.408
	s208	0.174	0.168	0.167
Total area normalized to baseline	COI	1.0480	1.0261	1.0182
	s27	1.0083	1.0079	1.0072
	s208	1.0096	1.0081	1.0075

bounce on the true ground node VSS and the settling time of V_{SS} for the switches occupying 10.4% of the total area increase by, respectively, 1.64× and 1.83× that of switches occupying 4.6% of the total area. Therefore, there is a trade-off between the power consumption and ground bounce when sizing the switches, where for the non-stacked mode, the peak noise on the VGND node is reduced to 0.07× at a cost of increasing the total power consumption and the switch area by 1.03× and 2.26×, respectively, when increasing the percentage of the total area occupied by the switches from 4.6% to 10.4%.

In addition, the noise on the VGND node (non-stacked), when the super-V_t circuits are operating in active mode, is characterized for 5%, 10%, and 15% allowed voltage bounce from the nominal value of V_{SS} (0 V). The reduction in noise on the VGND node (non-stacked) results in an increase in the area and power consumption due to the required larger switches, as shown from results listed in Table 8.2. There is an

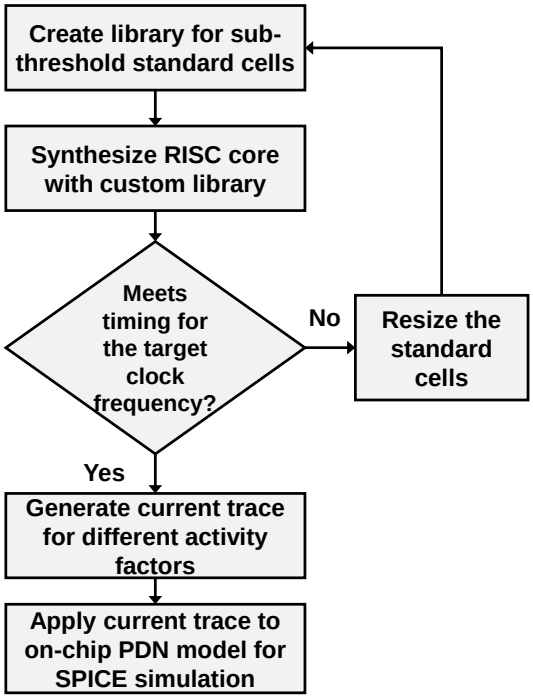


Fig. 8.13: Methodology to characterize workload variation of a sub-threshold core with SPICE simulation.

average increase in the power consumption of 1.03× when the voltage noise on the VGND node is constrained to 5% rather than 15% of the rail-to-rail voltage of V_{DD} (1.2 V for the super-V_t core).

8.1.3.f Regulation of Sub-threshold Power Supply Voltage Considering Workload Variation in the Sub-threshold Core

The super-threshold cores or circuit blocks are idle when applying the leakage reuse technique. Therefore, there are no executing workloads on the super-threshold cores that directly effect the generated sub-threshold voltage. However, the variation in the workload of the sub-threshold core does affect the generated sub-threshold

power supply voltage. Specifically, the generated voltage at the VGND node increases when a sub-threshold core transitions from a high activity state to a low activity state and decreases when transitioning from a low to high activity state. For the transition from a high activity state to a low activity state and for a given sequence of tasks of an executing application *A1*, there is a reduction in the current demand of the sub-threshold core. Therefore, the recycled leakage current from the idle super-threshold core is greater than the required current of the sub-threshold core when executing the application *A1*. The effects of workload variation of a sub-threshold core on the generated sub-threshold supply voltage is, therefore, analyzed. A 32-bit RISC-V core is synthesized in a 45 nm CMOS technology with an operating voltage and frequency of respectively, 380 mV and 1 MHz. Custom logic cells are designed for a 380 mV supply voltage as standard cells and libraries are not properly optimized for sub-threshold operation in standard process development kits (PDKs) [33]. The methodology of synthesizing the sub-threshold 32-bit RISC-V core is shown in Fig. 8.13. The custom cells are used to estimate the total current of the RISC-V core, which is obtained for different activity factors that emulate variations in the workload of the core. The sensitivity of sub-V_t supply voltage to variation in the workloads of the sub-V_t core is described in Section 8.1.3.g. In addition, a dynamic workload-aware idle core assignment is discussed in Section 8.1.3.h to provide a stable sub-V_t supply voltage.

Table 8.3: Off-chip and on-chip electrical parameters of the power delivery network [329–331].

Resistance	**Value (Ω)**	**Inductance**	**Value (H)**	**Capacitance**	**Value (F)**
R_{vrm}	100u	L_{vrm}	5n	C_{bulk}	132u
R_{pcb}	485u	L_{bulk}	510p	C_{pcb}	10u
R_{bulk}	4.365m	L_{pcb}	48p	C_{pkg}	440n
R_{ppcb}	10m	L_{pkg}	19p	C_{odc}	1n
R_{pkg}	123u	L_{ppkg}	104p	$C_{odc,subvt}$	1.5n
R_{ppkg}	15.72m	L_{bump}	38p		
R_{bump}	400u	L_{grid}	5.6f		
R_{grid}	50m				

8.1.3.g Sensitivity of Sub-V_t Supply to Workload Variations

The circuit used to evaluate the voltage noise on the VGND node due to variations in the workload of the RISC-V core is shown in Fig. 8.14. Similar to Fig. 8.6, four super-threshold cores supply current to a sub-threshold core through the proposed leakage reuse technique. The super-threshold cores operate at 1.2 V and with a clock frequency of 1 GHz, while a 380 mV supply voltage is generated for the RISC-V core operating at a frequency of 1 MHz. A current source is placed between the sub-threshold power and ground grids to emulate the total current load of the synthesized RISC-V core for different activity factors.

To accurately characterize the voltage noise on the sub-threshold supply voltage with variations in the workload of a RISC-V core operating at 380 mV, the four super-threshold cores are supplied current through a distributed power delivery network (PDN) modeled with both off-chip and on-chip power and ground networks as shown in Fig. 8.14. The electrical parameters of the off-chip and on-chip power and ground

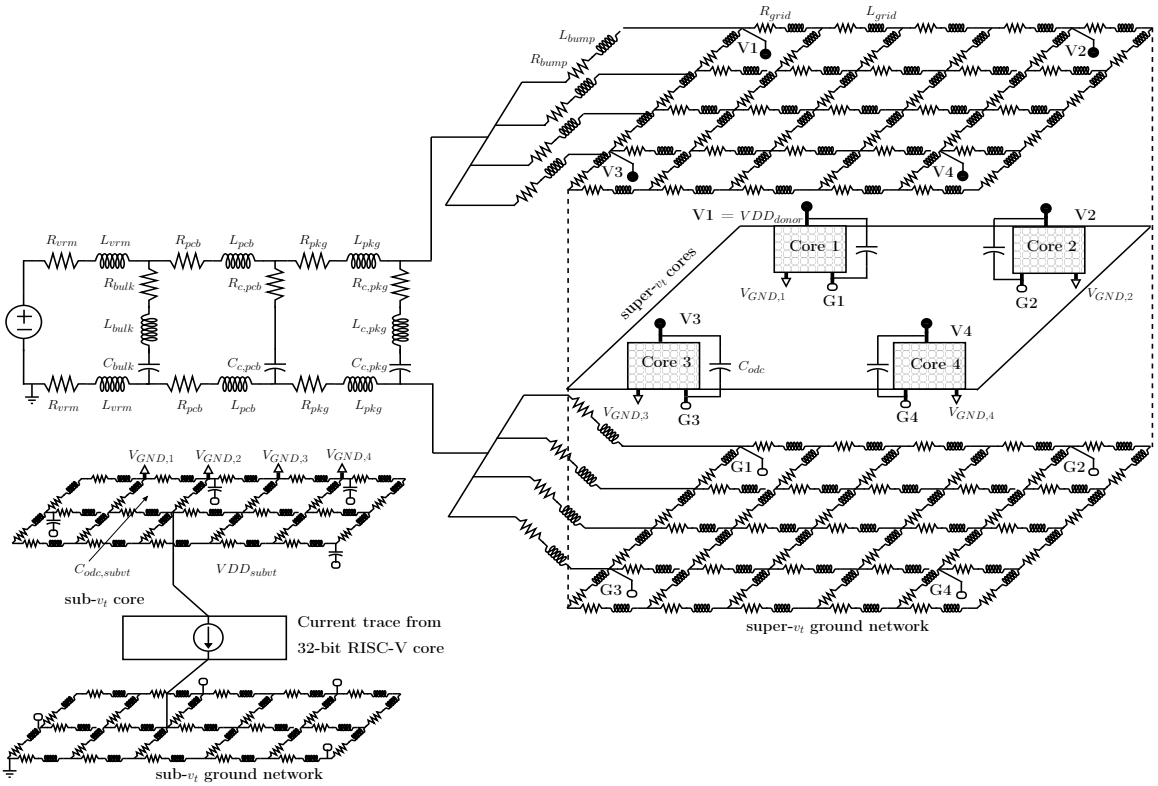


Fig. 8.14: Circuit model of the off-chip and on-chip PDN used to evaluate the voltage noise due to variation in the workload of a 32-bit RISC-V core operating at a sub-threshold voltage of 380 mV.

grids are listed in Table 8.3. A 6×5 on-chip grid is implemented for simulation to accurately model the impedance of the PDN, where the electrical parameters of the on-chip grid (R_{grid} and L_{grid}) and the C-4 bumps (R_{bump} and L_{bump}) are based on the impedance of a PDN of a four core SoC described in [329]. The electrical impedance of the off-chip voltage regulator module (VRM), board (PCB), and package are determined based on the PDN models presented in [330, 331]. The output of the VRM is maintained at 1.2 V to emulate the closed-loop response of a VRM.

The PDN is optimized to maintain the voltage ripple to within 10% of the V_{DD}

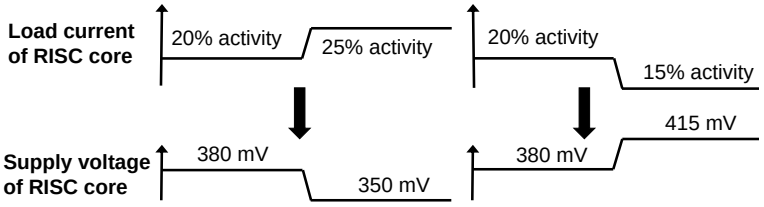


Fig. 8.15: A 380 mV supply voltage is generated for the RISC core through the leakage reuse technique, where the target activity factor is 20%. The variation in the supply voltage of the RISC core is characterized for circuit activity factors of 15% to 25%.

voltage and V_{SS} voltage. On-chip decoupling capacitors C_{odc} are placed close to each load to compensate for the voltage ripples on the PDN, where a total of 4 nF of decoupling capacitance is included for the four super-threshold cores. In addition, four on-chip decoupling capacitors, $C_{odc,subvt}$, each providing a capacitance of 1.5 nF are placed between the power ($V_{GND,1}$, $V_{GND,2}$, $V_{GND,3}$, and $V_{GND,4}$) and ground nodes of the sub-threshold core to minimize the voltage ripple on the power distribution network of the sub-threshold core during the highest circuit activity.

Each of the four super-threshold cores is composed of a s27 ISCAS89 benchmark circuit and a chain of buffers. The chain of buffers are added to accurately model the ratio of the logic area of the super-threshold core to the sub-threshold core. The circuit model shown in Fig. 8.14 is used for SPICE simulation to characterize the workloads executing on a 32-bit RISC-V core, where the circuit activity is restricted between 15% and 25% of all logical gates. The generated current profile for activity factors ranging from 15% to 25% is obtained from analysis of a synthesized RISC-V core. The total logical area of each super-V_t core is 1 mm^2, while the total logical

area of the RISC-V core is 10× larger (10 mm^2) than each super-threshold core. The 1 to 10 ratio of the area of the super-threshold to sub-threshold cores results in a 380 mV supply voltage for the sub-threshold RISC-V core executing workloads that activate 20% of all devices. In addition, for the given area ratio, the supply voltage is maintained within ±10% of 380 mV when the activity factor of the RISC-V core varies between 15% and 25% as indicated by results shown in Fig. 8.15. Therefore, a stable supply voltage with a bounded variation in voltage is available to the sub-threshold RISC-V core as long as the executing workloads are bound to a utilization of 15% to 25% of all transistors at any given time. However, for the case when the activity of the RISC-V core exceeds 25% or drops below 15%, a sub-threshold workload-aware idle core assignment technique is implemented to maintain a stable sub-threshold supply voltage.

8.1.3.h Dynamic Super-threshold Idle Core Assignment Based on Sub-threshold Workloads

In the case when the circuit activity factor of the RISC-V core exceeds 25%, a novel sub-threshold workload aware idle core assignment is applied. Depending on the executing workloads and, therefore, activity factor of the sub-threshold RISC-V core, the number of super-threshold idle cores or circuit blocks providing current to the RISC-V core is either increased or reduced at run time to maintain the required

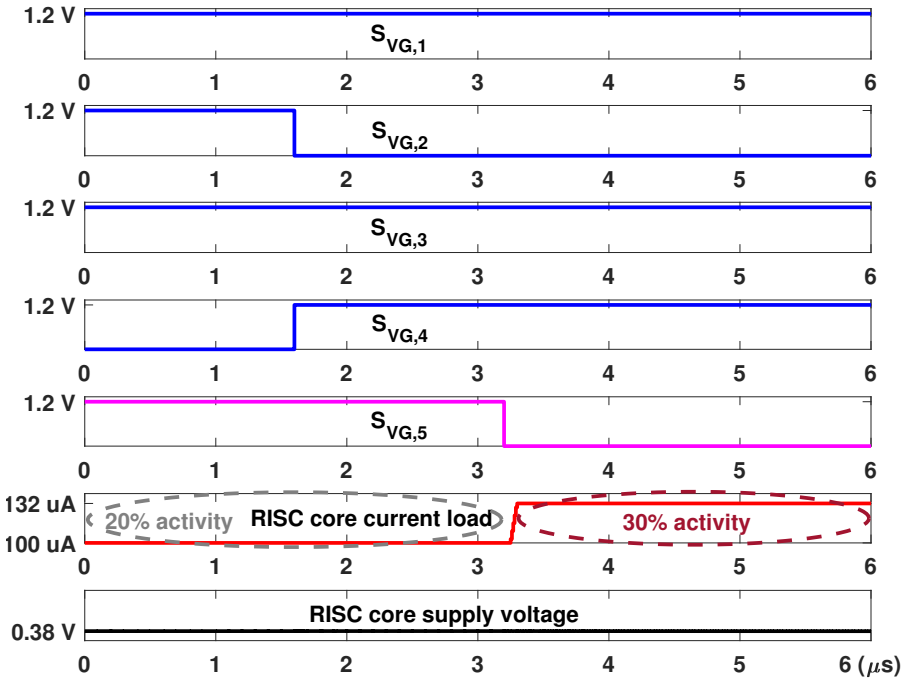


Fig. 8.16: SPICE simulation results indicating a stable sub-threshold supply voltage when the activity of the RISC-V core increases from 20% to 30%.

supply voltage at the VGND node. Therefore, close-loop voltage regulation of the power network is not required.

The proposed workload-aware idle core assignment is evaluated through SPICE simulation with the results shown in Fig. 8.16, where the noise on the grid of the sub-threshold power supply is characterized when the circuit activity of the RISC-V core changes from 20% to 30% transistor utilization. The total current of the RISC-V core operating at a clock frequency of 1 MHz with a 380 mV supply voltage is 82 μA, 100 μA, 115 μA, and 132 μA for, respectively, 15%, 20%, 25%, and 30% activity factors. The execution of the leakage reuse technique on the four super-threshold cores is managed by four control signals, $S_{VG,1}$, $S_{VG,2}$, $S_{VG,3}$, and $S_{VG,4}$, which are

applied to respectively, core 1, core 2, core 3, and core 4 as shown in Fig. 8.16. The circuit shown in Fig. 8.14 is used to obtain the results shown in Fig. 8.16. The current load of the sub-threshold RISC-V core is changed from 100 μA (20% activity) to 132 μA (30% activity) to evaluate the voltage noise generated on the VGND node when leakage reuse is applied. A sub-block of core 4 is added to the existing circuit configuration through a fifth control signal $S_{VG,5}$ to provide the additional charge needed to maintain the power supply voltage of the sub-threshold core within ±10% of 380 mV. The sub-block is sized such that the provided leakage current is sufficient to meet the additional current demand of the RISC-V core as the load increases to 132 μA from 100 μA. Therefore, a stable supply voltage of 380 mV is maintained on the power supply network of the sub-threshold core despite the increase in the circuit activity of the RISC-V core from 20% to 30%.

8.1.4 Trade-off Between Voltage Stacking and Leakage Reuse Technique

The transient analysis of both the leakage reuse and voltage stacking techniques is shown in Fig. 8.17, where the voltage bounce on the VGND node and the voltage levels of the logical output from the super-V_t and sub-V_t cores are compared. Similar clock and input switching patterns are applied to both implementations of the circuit.

The transient simulation is performed using two super-V_t cores each consisting

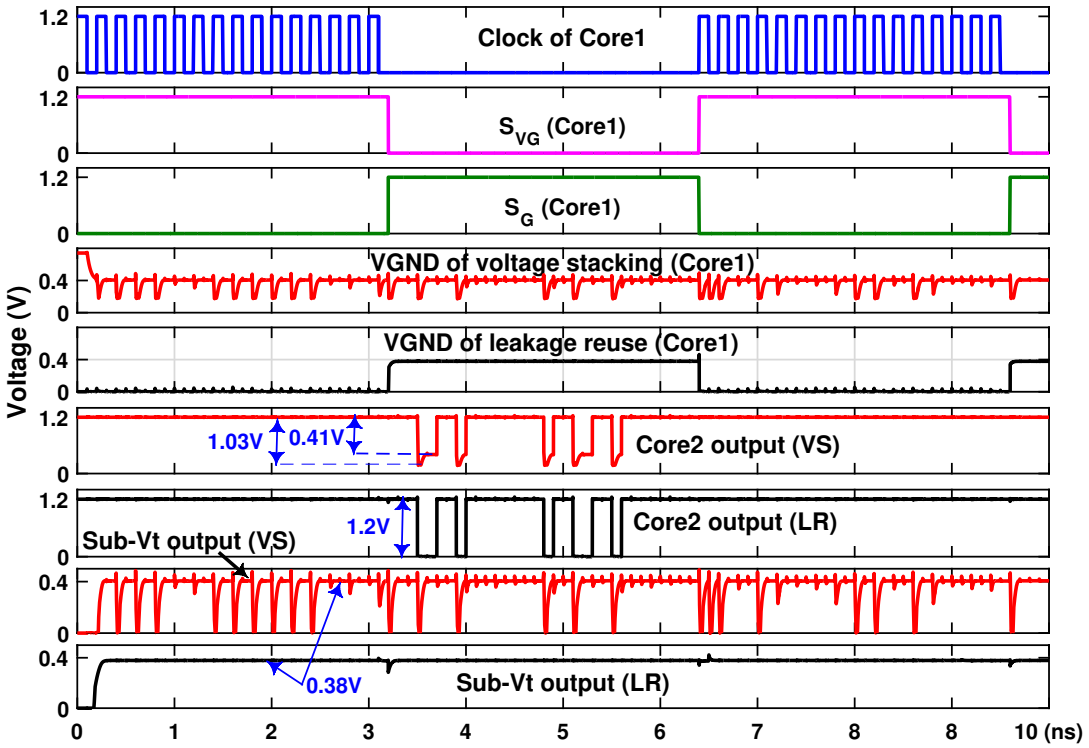


Fig. 8.17: Transient simulation of the leakage reuse and voltage stacking techniques comparing the noise on the virtual ground node and the voltage of the output signals of the super-V_t and sub-V_t circuits.

of an ISCAS `s27` benchmark circuit and one sub-V_t core consisting of a chain of six inverters. The clock, S_{VG}, S_G, V_{GND}, and output signals of one core are shown in Fig. 8.17. In addition, clock-gating is applied when *Core* 1 is inactive. In the case of voltage stacking, although *Core* 1 is idle, the switching transients ($L \cdot di/dt$) of the super-V_t *Core* 2 directly couple into the virtual ground node, where the charging and discharging of internal nodes results in significant voltage noise due to a large amount of internal capacitance. The noise induced on the VGND node directly impacts the logical output of *Core* 2, where the voltage for a logic 1 varies from 0.17 V to

Table 8.4: Characterization through SPICE simulation of s27 implemented with the baseline, leakage reuse, and voltage stacking techniques.

	Total power (μW)	Peak power (mW)	Peak V_{dd} (V)	Peak V_{ss} (mV)	Peak V_{GND} [non-stack] (mV)	Peak V_{GND} [stack] (mV)	Switching speed	Super-V_t transistors	Switch transistors	Sub-V_t transistors
Baseline	15.47	4.154	1.219	17.66	N/A	N/A	4×clock speed	Low V_t	N/A	Nominal V_t
Leakage reuse	6.448	3.822	1.211	13.41	120	420	4×clock speed	Low V_t	NMOS: 0.9 μm (high-V_t), PMOS: 3 μm (low V_t)	Nominal V_t
Voltage stacking	3.315	1.662	1.206	6.803	N/A	511	4×clock speed	Low V_t	N/A	Nominal V_t

0.41 V, which causes unwanted delay and logical failure for the circuits connected to the logical outputs of the super-V_t cores. In addition, the noise induced on the virtual ground node of the voltage stacked super-V_t core directly couple to the supply node of the sub-V_t circuits as the VGND node and sub-V_t supply node are shorted when voltage stacking is implemented. Therefore, additional voltage regulators are required to generate a stable supply voltage for the bottom core in the stack when voltage stacking is implemented, which results in a significant overhead in energy consumption [68, 176, 177].

In contrast, for the leakage reuse technique, the VGND node of $Core$ 1 is not affected by the switching of $Core$ 2. A stable voltage is, therefore, present at the VGND node used to supply current to the sub-V_t core. The logical outputs of $Core$ 2 are full-swing voltage signals, even as $Core$ 1 is stacked with the sub-V_t circuits. In addition, as the V_{GND} voltage is stable, the output signals from the sub-V_t core are also stable, as shown in Fig. 8.17.

Simulated results of the power consumption and peak voltage on the VGND and true ground nodes obtained using s27, which represents each super-V_t circuit block,

and a chain of inverters, which represents the sub-V_t circuit block, are listed in Table 8.4. The super-V_t and sub-V_t circuit blocks are sized to generate a sub-V_t supply voltage of 380 mV.

The voltage stacking technique reduces the leakage current for the entire duration of operation through the stacking effect, which increases the effective resistance in a given charging or discharging path. Therefore, the total power, peak power, and peak V_{SS} are reduced as compared to the baseline and leakage reuse technique at the cost of increased voltage bounce on the VGND node, as indicated by results shown in Figs. 8.8, 8.10, and 8.11. For `s27`, the resulting peak noise on VGND due to the implementation of the voltage stacking technique is 2.07× greater than the leakage reuse (non-stacked) technique, which imposes severe limitations for circuits operating in super-V_t as the drive strength of the NMOS transistors is significantly reduced. Therefore, the propagation delay of the circuits connected to the output of the super-V_t circuit block increases. The fanout-of-four (FO4) delay is characterized at 1.2 V in a 45 nm CMOS technology for ideal ground and for voltages of 120 mV, 420 mV, and 511 mV on the VGND node obtained from the results shown in Fig. 8.11 for, respectively, the non-stacked LR, the stacked LR, and the VS technique. The implementation of the voltage stacking and the leakage reuse techniques result in an increase in the FO4 delay of, respectively, 4.17× (V_{GND} voltage of 511 mV) and 1.24× (V_{GND} voltage of 120 mV) as compared to the baseline technique (ideal ground).

In addition, the noise margins of the circuits connected to the output of `s27` are significantly degraded as the voltage on the VGND node increases. The noise margins of a CMOS inverter with a PMOS width of 3.6 μm and NMOS width of 1.2 μm are characterized for an operating frequency of 5 MHz and with a 5 fF capacitive load. The noise margin low of the CMOS inverter is 483 mV, which is less than the 511 mV voltage on the VGND node when implementing the voltage stacking technique. Therefore, logic low is not discernible with the voltage stacking technique if an inverter is connected to the output of `s27`. In contrast, the leakage reuse technique implemented on `s27` provides a discerning logic low as the voltage on the VGND node is 120 mV, while also reducing the total and peak power consumption to, respectively, 0.41× and 0.7× that of the baseline. However, the FO4 delay increases by 1.24× as compared to the baseline. The peak voltage of 420 mV on the VGND node during the stacking mode of the leakage reuse technique does not effect the circuit delay and noise margin as the stacked super-V_t circuit blocks are operating in idle mode.

8.2 Analysis of Energy Break-Even Point

The PMOS and NMOS transistors shown in Figs. 8.6 and 8.7 and the circuitry that provides the control signal Φ consume additional energy. An analysis of the energy break-even point (BEP) is, therefore, performed, which is defined as the number

Table 8.5: Circuit parameters used in the analysis of the energy break-even point of the leakage reuse technique.

Parameter	*Definition*
$V_{DD,super-V_t}$	Supply voltage of super-V_t circuit
$V_{DD,sub-V_t}$	Supply voltage of sub-V_t circuit
V_{GND}	Virtual ground voltage
S_G	NMOS transistor connected to real ground
S_{VG}	PMOS transistor connected to sub-V_t core
$\alpha_{super-V_t}$	Switching activity of super-V_t core
α_{sub-V_t}	Switching activity of sub-V_t core
$C_{S,super-V_t}$	Total switching capacitance of super-V_t core including drain, source, and gate capacitance as well as interconnects
$C_{int,super-V_t}$	Total source capacitance of all transistors connected to virtual ground
$C_{S,sub-V_t}$	Total switching capacitance of sub-V_t core including drain, source, and gate capacitance of all transistors
C_{VGND}	Total capacitance of virtual ground due to S_G and S_{VG}
W_r	Ratio of switch size (sum of S_G and S_{VG}) to the size of super-V_t core

of clock cycles at which the total energy savings obtained from the implementation of the leakage reuse technique equals the total energy consumed by the PMOS/NMOS switches and the control circuitry [332]. The leakage current model changes significantly from one technology to another [27, 46]. Instead of performing a technology dependent analysis, heuristic based analytical expressions are derived using standard circuit and device parameters, which are listed in Table 8.5, that are applicable to any CMOS technology. In order to reduce the complexity of the analysis, a few assumptions are made: 1) The total energy required to operate a sub-V_t core is supplied by one super-V_t core and, therefore, analytical expressions are derived for $Core$ 1 and S_{Core} 1 only; a similar analysis can be performed for more complex systems, and 2) separate ground networks are implemented for the active super-V_t core and sub-V_t core to prevent noise coupling.

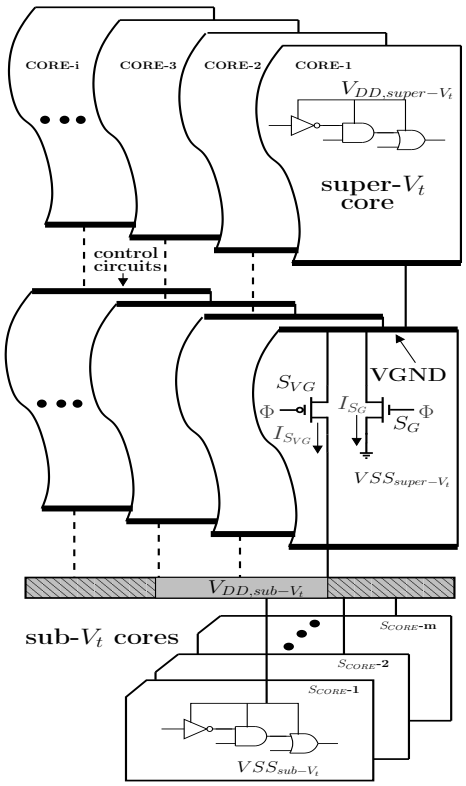


Fig. 8.18: Schematic representation of the proposed technique to reuse the leakage current from i number of super-V_t cores supplying current to one sub-V_t core.

A schematic representation of an implementation of the leakage reuse technique is shown in Fig. 8.18, where i number of super-V_t cores supply unused leakage current to m number of sub-threshold cores. The two footer transistors S_{VG} (PMOS) and S_G (NMOS) are utilized as control switches for each super-V_t core, where additional switches are added for a more distributed supply of current to the sub-V_t core. The control signal Φ is set to *logic low* when an onset of a long idle period of $Core$ 1 is detected, which results in S_{VG} turning on and S_G turning off as shown in Fig. 8.19. The control signal Φ applied to the gate of S_G transitions to a *logic high* when $Core$ 1 switches to active mode.

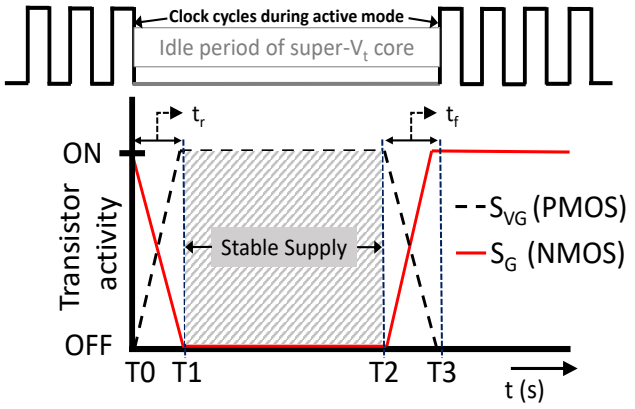


Fig. 8.19: Switching transitions of S_{VG} and S_G corresponding to the time intervals when leakage reuse is active ($\Phi = 0$).

The onset of the switching transitions of S_{VG} and S_G occur at times $T0$ and $T2$ as indicated by the transient waveforms shown in Fig. 8.19, while $T1$ and $T3$ represent the beginning of a period of steady-state voltage for the sub-V_t core. Assume that the idle activity of $Core$ 1 begins at time $t = T0$ and Φ, shown in Fig. 8.18, is set to logic 0. As transistor S_G begins to turn off, $Core$ 1 begins to supply unused leakage current to the sub-V_t core. In addition, at $t = T0$ $Core$ 1 switches to an idle state, which results in the charging of $C_{V_{GND}}$ and $C_{int,super-V_t}$. Note that the virtual ground node includes capacitances $C_{V_{GND}}$, $C_{int,super-V_t}$, and $C_{S,super-V_t}$, where half of the internal nodes of the super-V_t core contribute to the capacitance of the virtual ground node as described in [27]. At $t = T1$, the capacitance of the virtual ground node $C_{V_{GND}}$ is fully charged to the target V_{GND} voltage, and the leakage current through S_{VG} saturates. The voltage drop across the S_{VG} transistor is negligible, and V_{GND} is approximately

equal to the sub-V_t supply voltage $V_{DD,sub-Vt}$ of 380 mV. As the potential of the VGND node reaches a steady-state of 380 mV, the overall leakage current of $Core$ 1 is significantly reduced due to 1) an increase in the threshold voltage attributed to the reverse body-bias effect and 2) an increase in the effective resistance when stacked with the sub-V_t core. However, there is an additional energy loss E_{loss1} during time interval $t = T1 - T0$ due to the switching of both the S_{VG} and S_G transistors, where E_{loss1} per cycle is given by

$$E_{loss1} = \frac{1}{2}\alpha_{super-V_t}V_{DD}^2\left(C_G + C_{VG} + C_{int,super-V_t} + \frac{1}{2}C_{S,super-V_t}\right). \tag{8.3}$$

Similarly, the energy overhead per cycle during the time interval $T3 - T2$ is given by

$$E_{loss2} = \frac{1}{2}\alpha_{super-V_t}V_{DD}^2\left(C_G + C_{VG} + C_{int,super-V_t} + \frac{1}{2}C_{S,super-V_t}\right), \tag{8.4}$$

where at time $t = T2$ $Core$ 1 switches into an active mode and S_G and S_{VG} are turned on and off, respectively, which results in the discharge of the VGND node through S_G. During $t = T2 - T1$ the leakage energy of $Core$ 1 is used for computation by the sub-V_t core. At $t = T3$ the VGND node of $Core$ 1 discharges to true ground as $Core$ 1 begins to operate in active mode. The total energy overhead per cycle between time interval $T0$ and $T3$ due to the implementation of the leakage reuse technique is given by

$$E_{loss,total} = E_{loss1} + E_{loss2}$$
$$= \alpha_{super-V_t} V_{DD}^2 \left(C_G + C_{VG} + C_{int,super-V_t} + \frac{1}{2} C_{S,super-V_t} \right). \quad (8.5)$$

The reduction in the energy consumption due to the utilization of the leakage reuse technique over N number of cycles is given by (8.6). The E_L^{T1} term describes the total energy savings between time interval $T0$ and $T1$ and $E_{sub-V_t}^{cyc}$ is the total energy dissipation per cycle of the sub-V_t core. In addition, the average leakage energy dissipation per cycle when $Core$ 1 is in active mode and idle mode is given by $E_{L,active}^{cyc}$ and $E_{L,idle}^{cyc}$, respectively.

$$E_{saved}^{total} = E_L^{T1} + \sum_{i=0}^{N} \left(E_{sub-V_t}^{cyc} + E_{L,active}^{cyc} - E_{L,idle}^{cyc} \right) \quad (8.6)$$

The total energy dissipation of the sub-V_t core per cycle, as given by (8.7), is calculated as the sum of the static and dynamic energy dissipated per cycle, where $I_{S_{VG}}$ is the current through transistor S_{VG} and T is the clock period. The energy consumption of the circuit is reduced due to the implementation of the footer transistors S_{VG} and S_G, which behave similar to a switch in power-gated circuits. The energy

savings between the onset of an idle event ($T0$) and the time when V_{GND} begins to maintain a steady state ($T1$) is given by (8.8) and (8.9) [23, 27, 332]. The number of completed clock cycles during the time interval $T0$ to $T1$ is given as N_x and is empirically formulated as (8.10), where f is the operating frequency of the super-V_t core, W_r is the ratio of the size of the switch to the size of the super-V_t core, and a is the wake up latency, which is 5.27 ns for a 1 GHz clock frequency [333].

$$E^{cyc}_{sub-V_t} = \alpha_{sub-V_t} C_{sub-V_t} V^2_{DD,sub-V_t} + I_{S_{VG}} V_{DD,sub-V_t} T \tag{8.7}$$

$$E^{T1}_L = \frac{F_{DIBL}}{nV_T} N_x T E^{cyc}_L \tag{8.8}$$

$$= \frac{F_{DIBL}}{nV_T} \frac{\alpha_{super-V_t} \delta V_{DD}}{4\left(\frac{1}{2} + \frac{C_{V_{GND}}}{C_{S,super-V_t}}\right)} N_x T \tag{8.9}$$

$$N_x = \frac{fa}{W_r} \tag{8.10}$$

The average leakage energy for time interval $T0$ to $T1$, given by E^{T1}_L, is dependent on the drain-induced barrier lowering factor F_{DIBL} (≈ 0.1), the sub-threshold slope factor n (≈ 1.3), and the thermal voltage V_T ($= kT/q \approx 25mV$). The leakage factor δ, described in [27], is given as the ratio of the average leakage energy dissipation to the switching energy dissipation per cycle ($\delta = E^{cyc}_L / E^{cyc}_S$).

The switching energy dissipation per cycle of the super-V_t core is given by

$$E_S^{cyc} = \frac{1}{2}\alpha_{super-V_t} C_{S,super-V_t} V_{DD}^2, \tag{8.11}$$

where $C_{S,super-V_t}$ represents the total switching capacitance of the core including the drain, source, and gate capacitances as well as the capacitance due to the interconnects. The average leakage energy dissipated per cycle when $Core$ 1 is in active mode (not stacked) is given by

$$\begin{aligned} E_{L,active}^{cyc} &= TI_L V_{DD} \\ &= \delta E_S^{cyc} \\ &= \frac{1}{2}\alpha_{super-V_t}\delta C_{S,super-V_t} V_{DD}^2, \end{aligned} \tag{8.12}$$

where I_L is the average leakage current through all leakage paths in $Core$ 1 [27]. The leakage current through the PMOS transistor S_{VG} is negligible since the *drain* of S_{VG} is connected to $V_{DD,sub-V_t}$. Therefore, all of I_L is assumed to pass through S_G during the active mode operation of $Core$ 1.

The average leakage energy dissipated per cycle when $Core$ 1 is in idle mode (stacked) is given by

$$E_{L,idle}^{cyc} = TI_{S_G} V_{GND}, \tag{8.13}$$

where I_{S_G} is the current through S_G when operating in the idle mode. If an average leakage energy dissipation per cycle is assumed, the leakage energy dissipated over N

cycles is given by

$$\sum_{i=0}^{N} E_{L,idle}^{cyc} = NTI_{S_G}V_{GND}. \tag{8.14}$$

At the energy break-even point, the reduction in the total energy consumption equals the energy loss due to implementing the leakage reuse technique. Therefore, (8.5) and (8.6) are modified to

$$E_{saved}^{total} = E_{loss,total}, \quad \text{and} \tag{8.15}$$

$$E_L^{T1} + \sum_{i=0}^{N} \left(E_{sub-V_t}^{cyc} + E_{L,active}^{cyc} - E_{L,idle}^{cyc} \right)$$

$$= \sum_{i=0}^{N} \alpha_{super-V_t} V_{DD}^2 \left(C_G + C_{VG} + C_{int,super-V_t} + \frac{1}{2} C_{S,super-V_t} \right). \tag{8.16}$$

The energy consumption per cycle remains relatively constant except for a few clock cycles after a power down or power up event. For the provided analysis, a constant energy consumption is assumed for each cycle, where (8.7) is simplified by assuming $E_{sub-V_t}^0 \approx E_{sub-V_t}^1 \approx E_{sub-V_t}^2 \approx ... \approx E_{sub-V_t}^N$ for N cycles. A similar assumption is made for $E_{L,active}^{cyc}$ and $E_{L,idle}^{cyc}$. Substituting for E_L^{T1}, $E_{L,active}^{cyc}$, $E_{L,idle}^{cyc}$, and $E_{sub-V_t}^{cyc}$ in (8.16) results in

$$\left(E_L^{T1} + NE_{sub-V_t}^{cyc} + NE_{L,active}^{cyc} - NE_{L,idle}^{cyc} \right) = \left(C_G + C_{VG} + C_{int,super-V_t} + \frac{1}{2} C_{S,super-V_t} \right) N\alpha_{super-V_t} V_{DD}^2, \tag{8.17}$$

where

$$N_{break-even} = |N|$$

$$= \frac{E_L^{T1}}{\left(\left(C_G + C_{VG} + C_{int,super-V_t} + \frac{1}{2} C_{S,super-V_t} \right) \alpha_{super-V_t} V_{DD}^2 - E_{sub-V_t}^{cyc} + E_{L,active}^{cyc} - E_{L,idle}^{cyc} \right)}. \tag{8.18}$$

Advanced power management and task scheduling algorithms are used to identify the idle intervals in a multi-core system [317]. The reuse of the leakage current of the super-V_t core is activated through the control signal Φ only when the number of idle cycles in the super-V_t core exceeds $N_{break-even}$ and the voltage on the VGND node meets the target sub-V_t supply voltage. Therefore, the activation of the control signal Φ is formulated as

$$\forall_i, \Phi_{core,i} = \begin{cases} 0 \text{ (LR on)}, & \text{when } N > N_{break-even} \ \& \\ & |V_{k,i} - V_{DD,sub-V_t}| \leq 0.1 V_{DD,sub-V_t}^{target}, \\ 1 \text{ (LR off)}, & \text{otherwise} \end{cases} \tag{8.19}$$

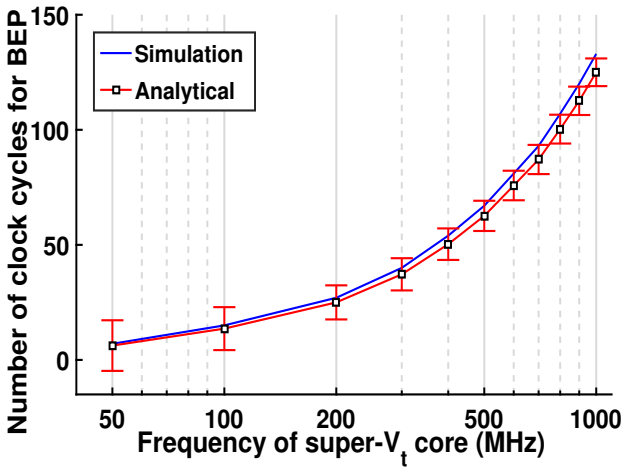


Fig. 8.20: Comparison between analytical and simulation results when determining the number of idle clock cycles of the super-V_t core corresponding to the break-even point in energy consumption for frequencies of 50 MHz to 1 GHz.

where i is the super-V_t core identifier for n cores $= \left\{1,\, 2,\, 3,\, \ldots n\right\}$. The voltage at the VGND node after k number of cycles is given as $V_{k,i}$. The target supply voltage of the sub-V_t core is represented by $V_{DD,sub-V_t}^{target}$. The minimum number of cycles N_{min} that the super-V_t core must be idle is given by (8.20), which ensures an overall gain in the energy efficiency of the system and a supply voltage for the sub-threshold core that is within 10% of $V_{DD,sub-V_t}^{target}$. The number of cycles required to settle within 10% of $V_{DD,sub-V_t}^{target}$ is defined as $N_{DD,sub-V_t}^{target}$.

$$N_{min} = \max\left\{N_{break-even}, N_{DD,sub-V_t}^{target}\right\} \tag{8.20}$$

Simulation is performed to determine the total energy consumption of a multi-core system where the required energy to operate the sub-V_t core is supplied by one super-V_t core. The results of the characterization of the number of cycles obtained from simulation are compared with the number of cycles determined through the analytical expressions. The super-V_t $Core$ 1 and the sub-V_t core are each implemented with a chain of six inverters. To analytically determine the $N_{break-even}$, the technology parameters listed in Table 8.5 are applied to (8.18) for both the super-V_t and the sub-V_t cores, where $C_{V_{GND}}/C_{S,super-V_t} = 0.1$, $\delta = 0.1$, $V_{GND} \approx V_{DD,sub-V_t} = 0.38V$, $T = 1$ ns, $\alpha_{super-V_t} = 1$, $\alpha_{sub-V_t} = 0.05$, $I_{S_G} = I_{S_{VG}} = 0.15I_{total}$, $I_{total} = 10$ µA, $W_r = 0.125$, and $C_{S,super-V_t} = 70$ fF. The total area of both the super-V_t and sub-V_t chain of inverters is listed in Table 8.1. The analytical and simulated results are plotted in Fig. 8.20, where the number of idle clock cycles corresponding to the break-even point in energy consumption is evaluated for 50 MHz to 1 GHz operating frequencies of the super-V_t core. For example, the number of cycles N when operating at a 1 GHz frequency that results in the energy break-even point ($N_{break-even}$) is 125 and 133 for the analytical and simulated analysis, respectively. The N_{min} that satisfies (8.20) is taken as 133 cycles. $Core$ 1 must, therefore, be stacked for at least 133 ns when operating at 1 GHz for the leakage reuse technique to provide an improvement in the overall energy efficiency of the circuit. The percent difference between the analytical and simulated results in Fig. 8.20 is calculated as $\frac{|Analytical-Simulation|}{Simulation} \times 100$. The maximum and

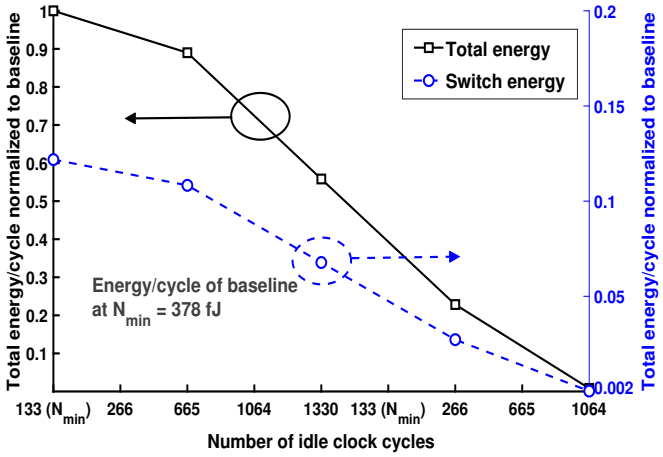


Fig. 8.21: Characterization of the energy per cycle normalized to the baseline for a range of idle cycles (133 to 1330) when the super-V_t core is operating at 1 GHz and for 1.33 µs. The left Y-axis represents the energy consumed by both the super-V_t and sub-V_t core per cycle, while the right Y-axis represents energy consumed by switches S_G and S_{VG} per cycle.

minimum percent error in the calculated number of idle cycles needed for the energy break-even point using the analytical expressions is, respectively, 11% (at 50 MHz) and 6% (at 1 GHz) as compared to simulated results. Note that the percent error decreases as the operating frequency increases since, for a given difference in the number of cycles between the analytical and simulated results, the relative difference is smaller at higher frequencies due to the larger number of cycles.

The total energy consumed by the super-V_t and sub-V_t cores per cycle and the energy consumed by the switches (S_{VG} and S_G) per cycle are characterized for 1 GHz operation of the super-V_t core starting from N_{min} (133 idle cycles at 1 GHz) to $10\times N_{min}$ (1330 idle cycles), with results shown in Fig. 8.21. The analysis of the leakage reuse technique is normalized to the baseline, where independent power

delivery networks are implemented for the super-V_t and sub-V_t cores as described by Topology 1 in Section 8.1.3.d. For the leakage reuse technique, the size of the super-V_t core is increased by 1.2× as compared to the baseline to account for the overhead due to the supporting circuits required to generate and propagate the control signal Φ. The overhead in area due to the supporting circuits as a percentage of the total area is smaller as the size of the super-V_t core increases, since the area of the supporting circuits remains about constant. Similar switching activity is applied to both the leakage reuse and the baseline implementations. The savings in the energy per cycle due to the leakage reuse technique increases as the leakage current from the super-V_t core is supplied for an increasing number of idle cycles. For example, the total energy per cycle is 0.23× the baseline for 1064 idle cycles, while consuming 0.89× the baseline for 266 idle cycles. In addition, as the number of continuous idle cycles of the super-V_t core increases, the fewer the number of triggered switching events needed to switch to an idle super-V_t core, which reduces the energy overhead due to the switches as a percentage of the total energy consumed. As an example, the energy per cycle of two switches is 0.12× the total energy when charge is reused for 133 cycles, while the switches consume 0.027× of the total energy per cycle when charge is recycled for 1064 cycles.

8.3 Algorithms for Leakage Reuse

Reusing leakage current from idle circuits provides significant improvement in the overall energy efficiency while also generating intrinsic voltage source(s) for ICs with multiple voltage domains [334, 335]. A controlled reuse (recycling) of the leakage current of an idle core generates a virtual ground voltage *VGND* that is used as a supply voltage for an active core. In addition, the leakage reuse technique reduces the leakage current through the idle cores. In this research, both circuit and algorithmic techniques are proposed to manage and reuse the leakage current of a circuit such that the overall energy consumption in a multi-processor system-on-chip (MPSoC) is reduced. In addition, the simultaneous implementation of the leakage reuse (LR) and power gating techniques is explored to further improve the energy efficiency.

In this section, idle cores from which leakage current is reused are described as *donor* cores and active cores to which reused charge is delivered as *receiver* cores. The proposed power management techniques are evaluated through a circuit model simulated in SPICE to accurately characterize the improved energy efficiency and the dynamic transients on the power delivery network (PDN) while accounting for process variation.

The algorithmic development of leakage reuse technique includes

- a novel dynamic idle core management technique and scheduling algorithm,

longest idle time - leakage current reuse (LIT-LR), to optimally assign the idle donor circuit block(s) or core(s) for leakage current reuse, and

- a novel technique and algorithm, *longest idle time - simultaneous leakage reuse and power gating (LIT-LRPG)*, to optimally assign idle cores for simultaneous execution of power gating and leakage reuse.

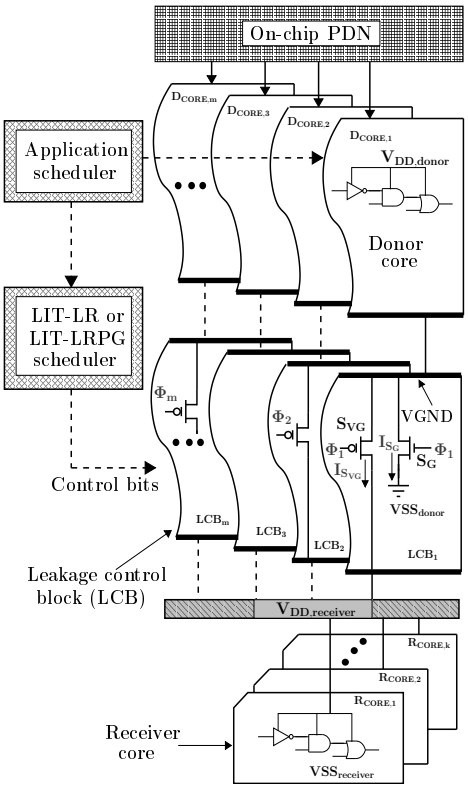


Fig. 8.22: System level model of the leakage reuse technique with dynamic idle core assignment.

The proposed power management technique is applied to only idle cores such that the circuit is unaffected when operating in active mode. The leakage current from idle cores is reused and delivered to an active core within a MPSoC through voltage stacking [68].

8.3.1 Circuit and System Model of Leakage Reuse

The proposed circuit model that accounts for the reuse of leakage current is shown in Fig. 8.22. In a MPSoC platform, m number of donor cores are connected to k number of receiver cores through m number of *leakage control blocks (LCBs)*, which consists of two transistors behaving as switches. Each of m number of LCB is implemented with an n-bit (n is 1 for Algorithm 3 and 2 for Algorithm 4) control signal Φ such that Φ_1 is applied to LCB_1, Φ_2 to LCB_2, and Φ_m to LCB_m. Each control signal Φ_1, Φ_2, ... Φ_m is connected to the gate terminals of the S_{VG} and S_G transistor of the corresponding LCB_m. Note that both the number of donor cores and receiver cores are scalable at design time. Under normal circuit activity, when a donor core $D_{CORE,m}$ is executing a workload, Φ_m is set to logic *high* and is applied to the LCB_m such that the ground node of all transistors within the donor core $D_{CORE,m}$ is connected to true ground (GND). The normal activity of the donor cores, therefore, remains unaffected. At the onset of an idle period, where the energy conserved exceeds the break-even point as defined by $N_{BE,LR,m}$ in Algorithm 3, Φ_m is set to logic *low* and is applied to LCB_m such that the donor core $D_{CORE,m}$ and the receiver core $R_{CORE,k}$ are stacked. The stacking serves two primary purposes: 1) to produce a virtual ground voltage that is used as the supply voltage of the receiver core $R_{CORE,k}$ and 2) to reduce the total leakage current of the idle donor core $D_{CORE,m}$. The conventional voltage stacking technique stacks two cores (a top PDN at a higher potential and a bottom PDN

at a lower potential) regardless of the circuit activity during both active and idle periods, where the executing workloads of the top core affects the supply voltage of the bottom core [68]. Therefore, expensive on-chip voltage regulators and closed-loop power management techniques are required to maintain a steady supply voltage for the bottom core [68]. However, the proposed technique only stacks idle donor cores (top core(s)) with no workload activity with receiver core(s) (bottom core(s)). Expensive voltage regulators are, therefore, not required to maintain a stable supply for the receiver cores. The simulated results presented in Section 8.4.3 indicate that the use of only on-chip decoupling capacitors is sufficient to negate the voltage transients on the PDN and, therefore, maintain a stable supply voltage for the receiver core.

The number of donor cores required to generate a desired supply voltage for a given receiver core is determined at design time. The desired supply voltage of a receiver core is set by either adjusting the ratio of the area of the donor and receiver cores, modifying the size of the switch in the LCB, or applying a charge booster such as an on-chip switch capacitor based voltage doubler to increase the supply level of a receiver core. In this paper, the supply voltage is set through the ratio of the areas of the donor and receiver cores.

8.3.2 Scheduling of Idle Donor Cores for Leakage Current Reuse

Conventional hardware and software based power management techniques do not implement leakage reuse despite the opportunities of efficiently conserving energy from idle donor cores [113, 317, 336]. In this paper, idle donor cores or circuit blocks are selected for leakage reuse to minimize the number of transistors switching in the *leakage control block* and to maximize the energy-efficiency of a MPSoC system. The proposed algorithm is equally applicable to core or circuit block level assignment for leakage reuse.

The proposed technique requires the scheduling information of donor cores to effectively select the idle donor cores for leakage reuse. In order to evaluate the effectiveness of the proposed leakage reuse technique, in the worst case, the application scheduling algorithm executes a task with the shortest duration possible and with a minimal idle time in a multi-core platform. The heterogeneous earliest-finish time (HEFT) algorithm meets both requirements as 1) the overall completion time is minimized by providing an execution rank for each task in an application for a set number of cores in a multi-core system and 2) tasks are scheduled based on an insertion-based policy that assigns tasks in idle time slots between two already-scheduled tasks on a core, which reduces the total idle time of the system [337].

Recently, dynamic voltage and frequency scaling (DVFS) is applied in conjunction with the execution of the HEFT algorithm, which provides an estimate of the total number of processor cycles executed by each core for a given task within a frequency range of 1 GHz to 2 GHz [317]. In this paper, the DVFS enabled HEFT algorithm is used as the application scheduler for the donor cores, which sends the scheduling information to the proposed *longest idle time - leakage current reuse (LIT-LR)* algorithm as shown in Fig 8.22. The *LIT-LR* algorithm generates and sends the control bit(s) to the LCB when scheduling the idle donor cores for leakage current reuse. In addition, a directed acyclic graph (DAG) is used to represent the application workload of the donor cores and is utilized by the HEFT algorithm.

Algorithm 3 Idle core assignment for leakage current reuse

Input: m, $Task_{num}$, $Order_{core}$, $Pcycles_{task}$, $N_{BE,LR}$
Output: Available idle core assignment to be used for leakage reuse with usage-based ordering (LR_{cores})

while $i \leq Task_{num}$ **do**
  Calculate the total number of cycles until the application is assigned to the next processor for execution
  if $Pcycles_{task}[i] > N_{BE,LR}$ **then** ▷ $N_{BE,LR}$ is the number of cycles corresponding to the energy break-even point for LR
    while j $\leq (Task_{num} - i)$ **do**
      Track the earliest execution start time of all of $Task_{num}$ - i tasks in m number of cores.
      Store a ranking of each core in $Core_{m,ranking}$ based on the order in which each task is executed.
    end while
    if $i \neq (Task_{num}$ - 1) **then**
      $LR_{cores\_index}[i] \Leftarrow max(Core_{1,ranking}, Core_{2,ranking}, Core_{3,ranking}, \ldots Core_{m,ranking})$ ▷ Assigning the donor core with longest idle time for LR
      $LR_{cores}[i] \Leftarrow Order_{core}[LR_{cores\_index}[i]]$
    else if $i == (Task_{num}$ - 1) **then**
      $LR_{cores}[i] \Leftarrow LR_{cores}[0]$ ▷ Tracking the donor core with longest idle time at the beginning of last execution and assigning for current state
    end if
  else
    $LR_{cores}[i] \Leftarrow Order_{core}[LR_{cores\_index}[i-1]]$ ▷ Keep the last idle core assignment for leakage reuse
  end if
end while
return LR_{cores}

8.3.3 Longest Idle Time - Leakage Current Reuse Algorithm

The pseudo-code of the proposed *LIT-LR* algorithm that efficiently schedules idle donor cores for leakage current reuse is provided as Algorithm 3. First, the execution order of each core $Order_{core}$ is calculated for the given task graph using the HEFT algorithm [337]. Then, the number of processor cycles required for each task $Pcycles_{task}$ is calculated in a DVFS enabled MPSoC platform [317]. The calculated values of $Order_{core}$ and $Pcycles_{task}$ are applied as inputs to both Algorithms 3 and 4. Based on the number of cores m, number of tasks in each task graph $Task_{num}$, $Order_{core}$, $Pcycles_{task}$, and number of cycles corresponding to the break-even point $N_{BE,LR}$, the *LIT-LR* algorithm determines the earliest execution time of each task on the m core system. Depending on the task execution order and idle intervals, each core is dynamically assigned a ranking, where the core with the longest idle time is given the highest ranking and the core with shortest idle time is assigned the lowest ranking. The scheduler gives priority to the highest ranking idle donor cores when assigning cores for leakage current reuse and sends the control bit Φ to the corresponding LCB. In this work, a single receiver core is assumed, and one donor core provides sufficient leakage current for the receiver core.

8.4 Simultaneous Leakage Reuse and Power Gating

Power gating (PG), a method to disconnect idle circuits from the power network, is a widely used power management technique to reduce the leakage current [26, 65, 110–115]. Techniques developed through prior research apply cut-off and intermediate mode power gating at the core, block, memory, and network-on-chip (NoC) level [26, 28, 65, 113–115, 120]. A block level PG technique has been proposed that analyzes the correlation between RTL modules and, therefore, power gates the module(s) in the case of inactivity [113]. A technique based on probabilistic analysis of NoC routers has been proposed, where queuing theory is used to model the break-even point in consumed energy when applying PG to the buffers of the router [114]. In addition, PG is used to gate caches and GPUs of a multi-core platform to reduce energy loss due to leakage current [115].

In addition, adaptive power gating is proposed to gate idle functional units [121]. Circuit blocks within a core or cores within a multi-core system are assigned to different operating modes (C-states) based on idle activity patterns to improve the performance per watt [26, 120]. Power gating with single and multiple sleep modes raises the voltage at the virtual ground node by up to VDD when applying a power-down mode and, therefore, reduces the leakage current of the circuit [28, 65, 112]. However,

the reduction in the leakage current is achieved at a cost of increased current transients on the power distribution network (PDN) and a large power consumption by the footer transistors (MOS switches) due to the discharging of charge on the virtual ground node to true GND during mode transitions [111, 112].

Current state-of-the-art power management techniques and algorithms do not include leakage current reuse with power gating despite the opportunity of significant improvement in energy-efficiency. However, despite the potential benefits, the overall system energy efficiency is not improved unless all idle donor cores are scheduled for leakage current reuse at any given time. In a MPSoC platform, the idle donor cores that are not scheduled for leakage current reuse continue to leak current. In this paper, the simultaneous implementation of power gating and leakage reuse is considered, which is applicable at the block, core, memory, and NoC level. Only donor cores are considered for power gating.

The opportunity of applying both power gating and leakage reuse introduces new challenges as both techniques target idle circuits or cores in a multi-core system. Power gating is best utilized when the core with the longest idle time is placed in deep sleep mode and the core with shortest ($\geq$ break-even point $N_{BE,PG}$) idle time is placed in light sleep mode [28]. Note that deep sleep and light sleep correspond to the largest and smallest wake-up penalty, respectively. Similarly, the leakage reuse technique is best utilized when the core with the longest idle time is placed in leakage

reuse mode to minimize the energy consumption of switching the LCBs. Therefore, the maximum benefit in energy efficiency is achieved when the cores with the longest idle times are assigned for either power gating or leakage reuse. Assigning the longest idle core for leakage reuse is beneficial only when the donor core provides just the sufficient amount of current to the receiver core. If the amount of supplied leakage current from a particular donor core is more than a receiver core requires, than the energy efficiency is reduced as an undesired increase in the *VGND* voltage occurs. Power gating improves the overall energy efficiency for a given donor core only when the reduction in the leakage current from applying power gating is greater than savings provided by leakage current reuse (reduction in leakage current plus the total energy consumed by the receiver core). Therefore, the assignment of idle cores for either power gating or leakage reuse is an optimization problem.

8.4.1 Longest Idle Time - Simultaneous Leakage Reuse and Power Gating Algorithm

A power management algorithm, *longest idle time - simultaneous leakage reuse and power gating (LIT-LRPG)*, is developed that dynamically assigns each idle donor core in a MPSoC for either power gating or leakage current reuse. The pseduo code describing *LIT-LRPG* is provided as Algorithm 4. The *LIT-LRPG* dynamically applies a core idleness ranking to each core, which is similar to Algorithm 3.

Algorithm 4 Simultaneous implementation of LR and PG

Input: m, $\delta[m]$, $N_{BE,PG}$, $N_{BE,LR}$, $Pcycles_{task}$, two-bits core tracker for each of m cores $core_t[m]$, which keeps track of each core for possible use in LR and PG mode
Output: core assignment for leakage reuse (LR_{cores}) and power gating (PG_{cores})
Procedure: Assign idle cores to both LR and PG

1: ▷ The following section (lines 2 to 36) replaces lines 8 to 15 in Algorithm 3
2: **for** $core = 1, 2, \ldots, m$ **do**
3:
$$\begin{aligned} \max \quad & X[m] \cdot E_{saved,LR}[m] + Y[m] \cdot E_{saved,PG}[m] \\ \text{s.t.} \quad & LR_{wake-up}[m] \cdot X[m] + PG_{wake-up}[m] \cdot Y[m] \leq L_{max}[m] \\ & E_{loss,LR}[m] \cdot X[m] + E_{loss,PG}[m] \cdot Y[m] \leq E_{loss,max}[m] \\ & X[m], Y[m] \geq 0 \end{aligned}$$
4: **if** $X[m] > Y[m]$ **then**
5: $\delta[m] \Leftarrow 1$
6: **else**
7: $\delta[m] \Leftarrow 0$
8: **end if**
9: **if** $core_t[m] == 01$ **then** ▷ Core m is idle and not used for either LR or PG
10: **if** $\delta[m] == 1$ **then** ▷ LR is selected
11: **if** $i \neq (Task_{num}$ - 1) **then**
12: $LR_{cores\_index}[i] \Leftarrow max(Core_{1,ranking}, Core_{2,ranking}, Core_{3,ranking}, \ldots Core_{m,ranking})$ ▷ Assigning the core with longest idle time for LR
13: $LR_{cores}[i] \Leftarrow Order_{core}[LR_{cores\_index}[i]]$
14: $PG_{second,max} \Leftarrow$ sort and assign core index with second longest idle time
15: $PG_{cores}[i] \Leftarrow Order_{core}[PG_{second,max}]$
16: $core_t[m] \Leftarrow 10$
17: $core_t[PG_{second,max}] \Leftarrow 11$
18: **else if** $i == (Task_{num}$ - 1) **then**
19: $LR_{cores}[i] \Leftarrow LR_{cores}[0]$ ▷ Tracking the core with longest idle time from the execution of last application and assigning for current state
20: $PG_{cores}[i] \Leftarrow LR_{cores}[1]$ ▷ Tracking the core with second longest idle time from the execution of last application and assigning for current state
21: **end if**
22: **else if** $\delta[m] == 0$ **then** ▷ PG is selected
23: **if** $i \neq (Task_{num}$ - 1) **then**
24: $PG_{cores\_index}[i] \Leftarrow max(Core_{1,ranking}, Core_{2,ranking}, Core_{3,ranking}, \ldots Core_{m,ranking})$ ▷ Assigning the core with longest idle time for power gating
25: $PG_{cores}[i] \Leftarrow Order_{core}[PG_{cores\_index}[i]]$
26: $LR_{second,max} \Leftarrow$ sort and assign core index with second longest idle time
27: $LR_{cores}[i] \Leftarrow Order_{core}[LR_{second,max}]$
28: $core_t[m] \Leftarrow 11$
29: $core_t[PG_{second,max}] \Leftarrow 10$
30: **else if** $i == (Task_{num}$ - 1) **then**
31: $PG_{cores}[i] \Leftarrow PG_{cores}[0]$ ▷ Tracking the core with longest idle time from the execution of last application and assigning for current state
32: $LR_{cores}[i] \Leftarrow PG_{cores}[1]$ ▷ Tracking the core with second longest idle time from the execution of last application and assigning for current state
33: **end if**
34: **end if**
35: **end if**
36: **end for**

A two-bit core tracker $core_t[m]$ is utilized in Algorithm 4 to represent four core activity scenarios: 1) 00 = $D_{CORE,m}$ is active, 2) 01 = $D_{CORE,m}$ is idle (not currently used for LR or PG), 3) 10 = $D_{CORE,m}$ is in use with LR, and 4) 11 = $D_{CORE,m}$ is in use with PG. The priority variable $\delta[m]$ defines which technique (LR or PG) provides greater energy savings for an idle donor core and, therefore, assigns the core to either power gating or leakage current reuse based on the value of the variable. For a $\delta[m]$ of 1, the idle core m is assigned for leakage current reuse, while the remaining idle cores, starting from the core with the second longest idle time, are assigned for power gating. For a $\delta[m]$ of 0, the idle core m is assigned for power gating, while the core with the second longest idle time is assigned for leakage current reuse. All remaining idle cores are assigned for power gating beginning with the core with the third longest idle time.

Assigning $\delta[m]$ at design time such that a fixed priority is set for LR and PG is possible, where one is assigned with a core with the longest idle time and another with a core with the second longest idle time. However, a dynamic assignment of $\delta[m]$ is implemented through linear programming as described by lines 3 through 8 in Algorithm 4. The variables X and Y define the impact of, respectively, LR and PG on the overall energy efficiency of the circuits in a MPSoC, where a higher value indicates greater energy efficiency and results in the assignment of δ to logic high. The variables $LR_{wake-up}$ and $PG_{wake-up}$ are the wake up latencies normalized

to the cycle time for, respectively, the leakage reuse and power gating techniques. The energy loss normalized to the total energy consumed by the donor core during the idle state is given by $E_{loss,LR}$ and $E_{loss,PG}$ for, respectively, the leakage reuse and power gating techniques. The energy loss is a function of the energy consumption of an idle donor core $E_{donor,idle}$, the technology dependent fitting parameter η, the energy consumed by the LR LCBs $E_{LCB,LR}$, and the energy consumed by the PG LCBs $E_{LCB,PG}$ and is given by $E_{loss,LR} = \frac{\eta \cdot E_{LCB,LR}}{E_{donor,idle}}$ and $E_{loss,PG} = \frac{\eta \cdot E_{LCB,PG}}{E_{donor,idle}}$ for, respectively, the LR and PG techniques. The savings in energy normalized to the total energy consumed for the leakage reuse and power gating techniques is given by $E_{saved,LR} = \frac{E_{donor,idle} - E_{LCB,LR} + E_{receiver,LR}}{E_{total}}$ and $E_{saved,PG} = \frac{E_{donor,idle} - E_{LCB,PG}}{E_{total}}$, respectively. In addition, L_{max} describes the maximum allowed latency normalized to the cycle time, and $E_{loss,max} = \frac{E_{donor,idle}}{E_{total}}$ defines the maximum allowed energy loss normalized to the total energy.

8.4.2 Simulation Framework

The activation and deactivation of a power gating mode results in voltage transients on the power distribution network (PDN) due to the switching of the control transistors implementing the power gating [111, 338]. The proposed technique that reuses leakage current from idle cores results in similar switching voltage transients. Therefore, the voltage noise due to the implementation of the leakage reuse and power gating techniques is characterized while accounting for the impedance of the power

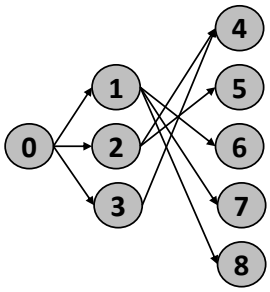


Fig. 8.23: DAG with nine tasks representing the executing application used to characterize *LIT-LR* and *LIT-LRPG* algorithms.

network. In this work, an optimized PDN with off-chip and on-chip impedance parameters is implemented that allows for a more accurate analysis of the transient noise.

Both the *LIT-LR* and *LIT-LRPG* algorithms are evaluated through SPICE simulation using a 45 nm CMOS process. A MPSoC platform with five cores is utilized, where four homogeneous cores are implemented as donors and one core is implemented as a receiver. The donor cores operate at a supply voltage of 1.2 V with a 1 GHz clock frequency, while a sub-threshold (Sub-V_t) supply voltage of 340 mV is generated for the receiver core from the reused leakage current of one of the four donor cores. The receiver core operates at 100 MHz. Note that the supply voltage of the receiver core is adjustable based on the ratio of the areas between the donor and receiver cores as mentioned in Section 8.3.1. The ISCAS89 benchmark circuits and boolean logic circuits are implemented within each of the four donor cores, while the receiver core is implemented as either the ISCAS89 s27 benchmark circuit or a 32-bit ARM Cortex M0 core.

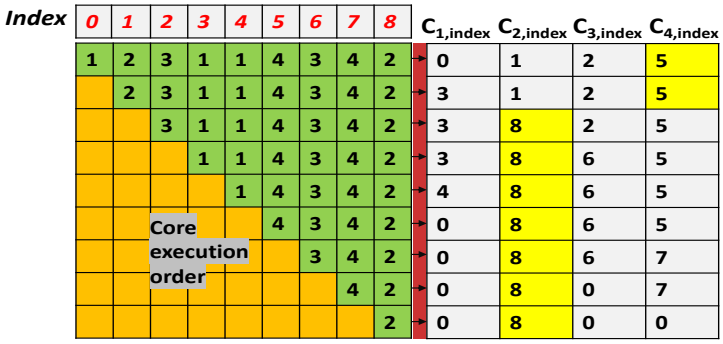


Fig. 8.24: Identification of donor cores with longest idle time that are assigned for leakage current reuse using the *LIT-LR* algorithm.

An application with nine periodic tasks is used to characterize non-preemptive scheduling of donor cores, where the task graph is as shown in Fig. 8.23. The execution order of four donor cores is shown in Fig 8.24. During the execution of a task on a given donor core, the remaining donor cores are idle and, therefore, available for *LIT-LR* and *LIT-LRPG* scheduling. The longest idle period for each core when executing the nine tasks is determined through the *LIT-LR* algorithm. For example, Task 0 is executed on core 1 as shown in Fig 8.24, while core 4 provides the longest idle time before starting execution of a task and, therefore, is assigned with the largest index value ($C_{4,index} = 5$). In this case, core 4 is assigned for leakage current reuse when executing Task 0 on core 1. Similarly, core 2 provides the longest idle time when executing third task on core 3 and is, therefore, assigned for leakage current reuse. Note that the index number and task number are not the same.

An accurate estimation of the number of cycles corresponding to the break-event points ($N_{BE,LR}$ and $N_{BE,PG}$) requires a separate analysis. In this paper, $Pcycles_{task}$

is assumed as larger than both $N_{BE,LR}$ and $N_{BE,PG}$ for the entire task graph since only one receiver core implements leakage current reuse and four homogeneous donor cores are used for both LR and PG. To the best of our knowledge, this is the first work that addresses the scheduling of idle cores for leakage current reuse. Therefore, the proposed *LIT-LR* algorithm is compared against random assignment of available idle donor cores for leakage current reuse. Based on the idle core assignment performed by the *LIT-LR* and *LIT-LRPG* algorithms, control bits are set and provided to the LCB for SPICE simulation. For the *LIT-LR* algorithm, a one bit control signal Φ is generated and supplied, while a two-bit control signal is generated and supplied to the LCB after executing the *LIT-LRPG* algorithm. Note that the LCB circuit is similar to the control circuit used in [334] and [335], and the same number of transistors are used in each LCB for both the *LIT-LR* and *LIT-LRPG* algorithms.

Optimized PDN with Off-Chip and On-Chip Impedance Model:

The four donor cores are supplied current through an optimized PDN that includes both off-chip impedance components and on-chip distributed power and ground networks as shown in Fig. 8.25. The off-chip and on-chip impedance parameters are listed in Table 8.3 [335]. A 4×5 on-chip grid is used to improve the simulation accuracy without significantly increasing the computational cost. The output of the off-chip voltage regulator module (VRM) is maintained at 1.2 V to emulate the closed-loop

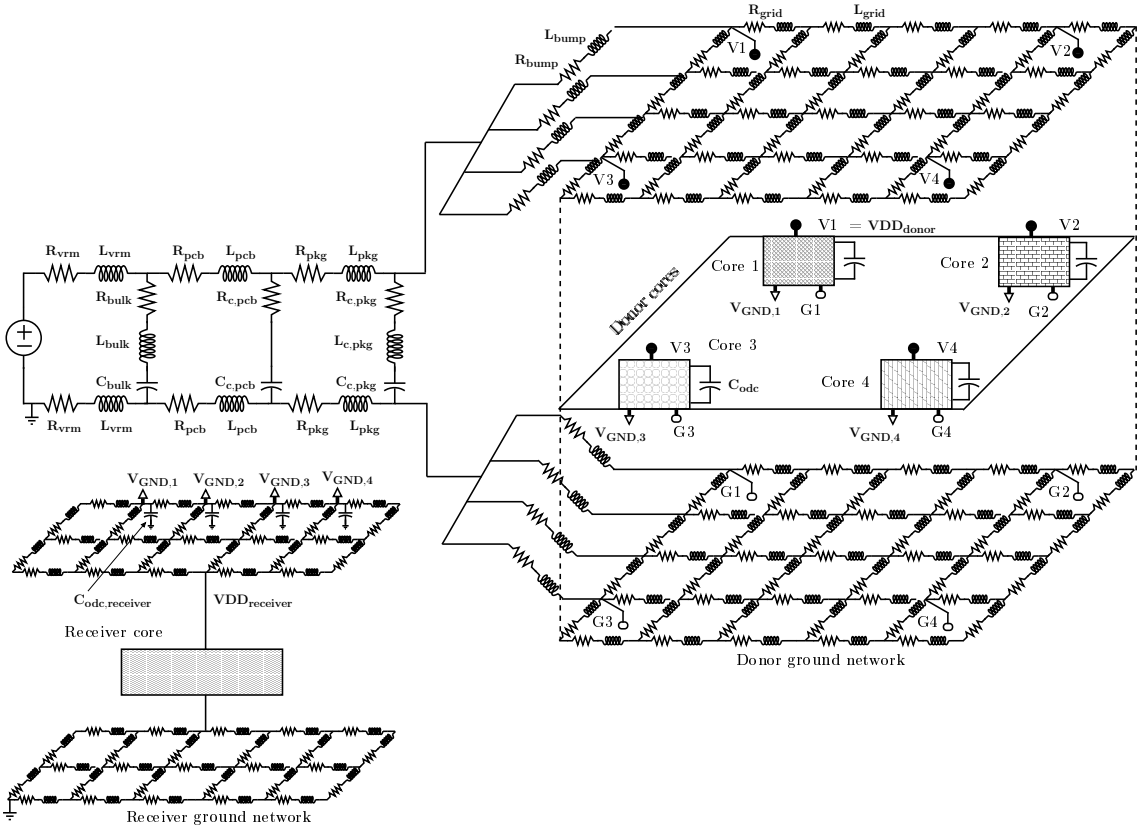


Fig. 8.25: Optimized PDN modeled with off-chip and on-chip impedance to evaluate the *LIT-LR* and *LIT-LRPG* algorithms.

response of a VRM.

On-chip decoupling capacitors (C_{odc}) are placed at each load point to compensate for any voltage ripples, where a total of 4 nF of decoupling capacitance is implemented for the four donor cores. In addition, an on-chip decoupling capacitor $C_{odc,receiver}$ of 50 pF is implemented on the supply node of the receiver core to minimize any voltage ripple on the virtual ground node. The frequency response of the PDN is evaluated. A maximum impedance of 320 mΩ occurs at 700 MHz, which is much lower than

the target impedance of the PDN given by $\frac{V_{DD} \times \%ripple}{I_{max} - I_{min}}$. The PDN is optimized to maintain the voltage ripple within 10% of the target voltage on both the VDD and GND networks.

8.4.3 Results and Discussion

The *LIT-LR* algorithm is evaluated through SPICE simulation using the circuit shown in Fig. 8.25 for the `ss, tt, and ff process corners`. The voltage transients due to the leakage reuse technique are characterized for three different sizes of the switches within each LCB, where switches with 7.5%, 15%, and 30% of the area of a donor core are used.

Table 8.6: Execution order of four *donor* cores and the corresponding assignments by the *LIT-LR* and *LIT-LRPG* algorithms for leakage current reuse and power gating.

Execution order of donor cores		1	2	3	1	1	4	3	4	2
Idle core assignment	Leakage current recycling: LIT-LCR	4	4	2	2	2	2	2	2	4
	Leakage current recycling: Random	2	1	4	2	3	1	2	3	1
	Power gating (LIT-LCRPG)	3 2	1 3	4 1	3 4	3 4	3 1	4 1	1 3	3 1

The execution order generated by the HEFT algorithm for the DAG shown in Fig.8.23, the idle donor core assignments produced by the *LIT-LR* algorithm, and a random allocation of cores is listed in Table 8.6. The control bits generated by both *LIT-LR* and a random assignment are provided to the LCB for SPICE simulation.

The transient behavior of the leakage current reuse technique when executing the

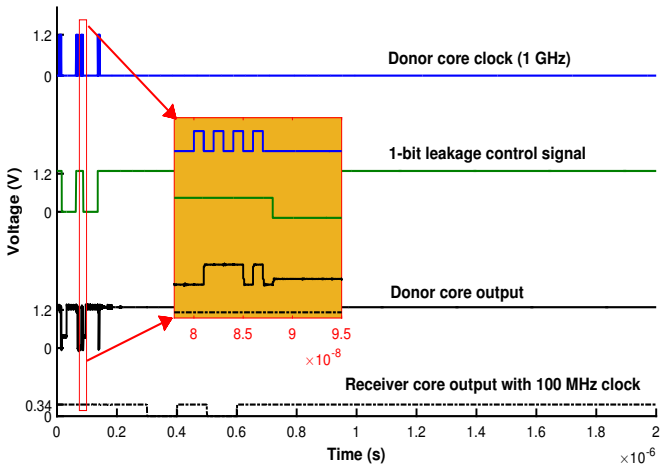


Fig. 8.26: SPICE simulation results characterizing the activity of both the donor and receiver cores when executing the leakage reuse technique when applying the *LIT-LR* algorithm.

LIT-LR algorithm is shown in Fig. 8.26, where the area of the LCB is 7.5% that of a donor core. The outputs of a donor and a receiver core are plotted in Fig. 8.26, which operate at a clock frequency of, respectively, 1 GHz and 100 MHz. Despite the reuse of leakage current, a full voltage swing is maintained at every clock cycle at the outputs of all four donor cores (the output of only one donor core is shown). In addition, a steady voltage of 340 mV is maintained at the output of the receiver core. Additional characterization of the maximum and minimum voltage noise on the supply node of the receiver core (the *VGND* node) is performed while considering process variation and for three different sized LCBs. The results of the characterization are listed in Table 8.7. Even with large MOS switches (up to 30% of the donor core) in each LCB and for the extreme process corners (`ss` and `ff`), the maximum and minimum supply voltage of the receiver core remains within ±3% of 340 mV. Therefore, the

proposed leakage current reuse technique and scheduling algorithm provide a stable supply voltage for the receiver core with the use of only on-chip decoupling capacitors.

Table 8.7: Voltage generated on the receiver core from reused charge from idle donor cores for different switch sizes and process corners.

Switch size (%)	7.5			15			30		
Corner	*SS*	*TT*	*FF*	*SS*	*TT*	*FF*	*SS*	*TT*	*FF*
Max. $Sub-V_t$ **VDD (mV)**	336.5	340.6	345.4	337.2	341.2	345.9	338.6	342.4	346.9
Min. $Sub-V_t$ **VDD (mV)**	336.2	340.2	344.8	336.9	340.8	345.4	338.2	342	346.4

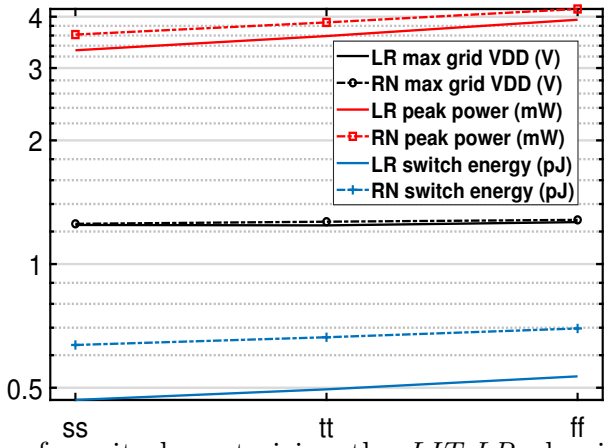


Fig. 8.27: Figures of merit characterizing the *LIT-LR* algorithm (solid lines) and random core assignment (dotted lines).

The maximum noise on the supply voltage *VDD*, the peak power consumption of the MPSoC system (the sum of the power consumed by the donor cores, receiver core, and LCBs), and the total energy consumed by the LCBs due to the execution of the *LIT-LR* assignment and the random core assignment are characterized for the `ss`, `tt`, and `ff` process corners with the results shown in Fig 8.27. The assignment of idle donor cores using the *LIT-LR* algorithm reduces the energy consumed by the switches by 27%, 25%, and 24% in, respectively, the `ss`, `tt`, and `ff` corners as

compared to the random assignment of cores. The peak power consumption when assigning cores with *LIT-LR* is reduced by 8.4%, 7.4%, and 5.8% as compared to the random assignment of cores in, respectively, the `ss`, `tt`, and `ff` corners. In addition, the maximum voltage noise on the on-chip distributed PDN is reduced by 0.64%, 2.1%, and 1.3% when *LIT-LR* assignment is performed in, respectively, the `ss`, `tt`, and `ff` corners as compared to random assignment.

Similar to the *LIT-LR* algorithm, the control bits generated from execution of the *LIT-LRPG* algorithm are supplied to the LCBs shown in Fig. 8.22 to evaluate the simultaneous implementation of LR and PG. SPICE simulation is performed at the `tt` corner and with a LCB area of 7.5% that of a donor core. However, for the simultaneous implementation of both techniques, two-bit control signals are passed to each LCBs: one bit to S_{VG} and another to S_G.

The simulated results obtained from a 45 nm CMOS technology are used to solve the optimization problem described in line 3 of Algorithm 4. The $LR_{wake-up}$ and $PG_{wake-up}$ parameters, which are extracted from simulation, are set to, respectively, 1.1 and 1.2 for a 1 GHz clock frequency. The $E_{donor,idle}$, $E_{LCB,LR}$, $E_{LCB,PG}$, and E_{total} parameters are determined as, respectively, 4.75 pJ, 0.16 pJ, 0.63 pJ, and 29.2 pJ. Consequently, an optimum solution is found with a larger value of X by setting $E_{loss,LR}$, $E_{loss,PG}$, $E_{saved,LR}$, $E_{saved,PG}$, L_{max}, $E_{loss,max}$, and η to, respectively, 0.33, 1.33, 0.162, 0.14, 5, and 0.17, and 10. As homogeneous cores are used, the value of

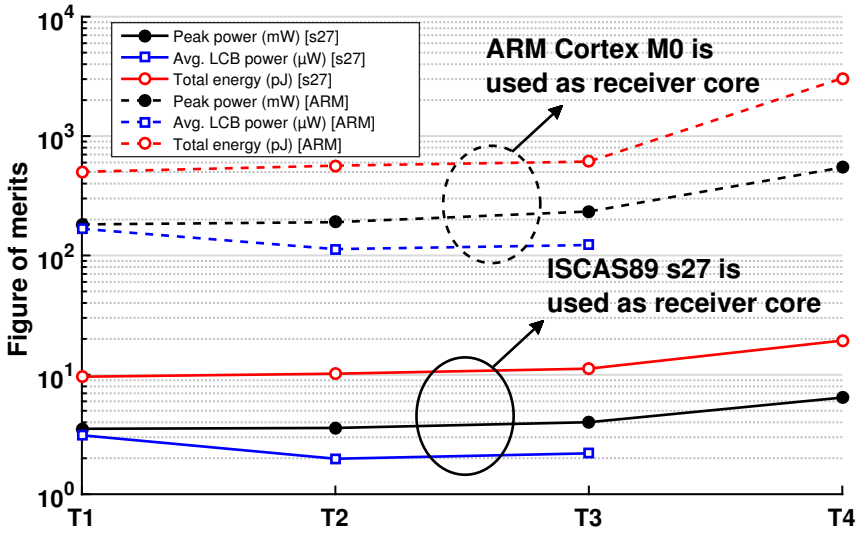


Fig. 8.28: Simulated figure of merits using proposed Algorithm 4, where T1, T2, T3, T4 represents, respectively, simultaneous implementation of LR and PG, LR only, PG only, and baseline without LR and PG.

$\delta\,[m]$ is the same for all donor cores. Therefore, in this case, the idle donor core with the longest idle period is always assigned for leakage current reuse.

The peak power, average power consumption per cycle, and total energy consumption are characterized for four power management scenarios: T1) simultaneous implementation of leakage current reuse and power gating, T2) power gating only, T3) leakage current reuse only, and T4) without either leakage current reuse or power gating (baseline). The simulation is performed with two sizes of receiver cores: one that contains only an s27 benchmark circuit (3.36 μm^2) and another that consists of a 32-bit ARM Cortex M0 core (98930 μm^2). Both s27 and the synthesized ARM core operate with a supply voltage of 340 mV. The current trace for 10% activity of the ARM core is applied to the PDN shown in Fig. 8.25 for SPICE simulation. The size

of each donor core is increased by 1750× (equivalent to 1750 s27 blocks) to provide sufficient leakage current to the ARM core operating at 340 mV.

The simulation results for the four scenarios are shown in Fig. 8.28. The simultaneous implementation of LR and PG reduces the total energy consumption of the donor cores by 50.2% (75.5%), 14.4% (18%), and 5.7% (11.2%) for receiver cores consisting of s27 (ARM Cortex M0) as compared to, respectively, the baseline with no PG or LR, power gating only, and leakage current reuse only. The use of leakage current reuse also reduces the average power consumption per cycle of the LCB by 9.9% (7.6%) as compared to power gating when applied to receiver core consisting of s27 (ARM core). In addition, the simultaneous implementation of PG and LR on s27 (ARM core) reduces the peak power consumption by 45.2% (66.7%), 11.9% (22%), and 1.62% (4.2%) as compared to, respectively, the baseline, PG only, and LR only. Although the simultaneous implementation of LR and PG results in a greater power consumption by the LCB than power gating or leakage current reuse implemented separately, the energy consumption and peak power consumption of the overall MPSoC is reduced.

8.5 Summary

In this chapter, a novel leakage reuse technique is developed to deliver unused current to low voltage circuits, where the leakage current from the idle donor circuits

is recycled to deliver current to receiver circuits that are active and operated at sub- and near-threshold supply voltages. An additional voltage source is, therefore, not required for the receiver circuits, while the leakage overhead of the donor circuits is also reduced. An analysis of the energy break-even point is provided, which describes the minimum idle number of cycles of a donor core required to achieve an overall improvement in the energy-efficiency of the multi-core system. In addition, a novel method of dynamically assigning idle donor cores or circuits for leakage reuse is implemented to maintain a stable receiver supply voltage for workloads executed on the receiver circuits with variable current load demands. Circuit and algorithmic techniques for dynamic idle core management are proposed to reduce the leakage current and improve the overall system energy efficiency of a MPSoC platform. Algorithms are developed to optimally schedule workloads when implementing the leakage reuse technique with and without implementing power gating.

Chapter 9

Leakage Reuse for Energy Efficient Near-memory Computing of Heterogeneous DNN Accelerators

The implementation of artificial neural network (ANN) based models in existing CPU, GPU, and FPGA based architectures is computationally costly [59–61, 242, 339, 340]. Therefore, active research to accelerate the computation of ANNs with custom ASICs is being pursued that leverages parallelism and data reuse [59–62]. However, the challenges of storing and moving large quantities of data still exist in state-of-the-art AI accelerator architectures [242, 339, 340]. The large number of input activations, output activations, weights, and addresses are stored in a combination of

off-chip and on-chip memory depending on the latency and bandwidth requirements of the application [242, 339, 340]. However, at any given clock cycle, only a tiny fraction of both the off-chip and on-chip memory is actively used [59–61]. In this chapter, the application of the leakage reuse technique on an on-chip SRAM array is discussed, where the array implements a heterogeneous multi-voltage domain DNN accelerator for near-memory computing.

The primary contributions described in this chapter include

- a design space exploration analyzing the computational demand and execution time of the DNN models, which includes the characterization of multiple sub-layers of the models,
- a novel leakage reuse (LR) technique applied to on-chip SRAM, where the leakage current from idle memory banks (memory banks that are in a hold state) are recycled to deliver power to the adjacent processing elements of a DNN accelerator for near-memory computing,
- a novel multi-voltage domain heterogeneous DNN accelerator architecture that executes multiple models simultaneously with different power-performance operating points, where each sub-array of PEs is tied to an independent power domain, and
- a close-loop adaptive voltage control technique to mitigate variations due to process, voltage, temperature (PVT), and aging.

This chapter is organized as follows. An introduction to custom DNN accelerators is provided in Section 9.1. The application of the leakage reuse technique to an on-chip SRAM is described in Section 9.2. The power management of multi-voltage domain DNN accelerators that reuse the leakage current from SRAM memory is described in Section 9.3.

9.1 Custom DNN Accelerators for Edge Applications

The exploration of custom deep neural network (DNN) accelerators for highly energy constrained edge devices with on-device intelligence is gaining traction in the research community. Several custom architectures were proposed in the past five years that implement DNN accelerators with improved performance, throughput, and energy efficiency as compared to CPU, GPU, and FPGA based implementations [53, 59, 62, 194, 199, 205, 239, 253, 255]. Despite the superior throughput and performance of custom accelerators, the energy efficiency and versatility of state-of-the-art DNN accelerators is constrained due to the storage and movement of a large quantity of data. Recent advances in DNN models and DNN accelerators (customized

hardware architectures optimized for DNN inference) have provided significant improvement in incorporating intelligence into ubiquitous edge devices, which are implemented with highly stringent energy efficiency requirements. The use of edge devices for applications including computer vision, augmented reality (AR), face recognition, image processing, and speech applications require DNNs with varying specifications [53, 214]. The state-of-the-art DNN models customized for resource constrained edge devices are capable of *inference* with as few as 2-bits of arithmetic operation [341], while *training* is demonstrated with as few as 4-bits of arithmetic operation [342]. As the bit precision is reduced, the execution of both inference and training is more feasible on edge devices. However, hardware level implementations of optimized DNNs has not sufficiently progressed in the research community as compared to the breakthroughs made in model and algorithm development due to a) the lack of efficient circuits and architectures implementing the DNNs, and b) the higher power consumption of the DNN accelerators as a result of executing a large number of computations and the inclusion of large off- and on-chip memories.

The development of an optimized DNN accelerator is an area of on-going research effort [53, 59, 62, 194, 199, 205, 239, 253, 255]. As neural networks are implemented with a greater number of layers and with increasingly diverse layer structures, new challenges emerge due to a) the storing and management of a large volume of data sets, and b) the requirement of efficient dataflows for layer by layer processing of

DNN models [53, 253]. In addition, the implementation of a DNN accelerator that offers both optimized energy efficiency and optimized performance across the various DNN models is a challenge to achieve as the architecture of the DNN accelerators is often fixed and only provides optimized results for a sub-set of DNN models [53, 196]. The primary challenges of implementing optimized DNN accelerators are discussed in this section. The challenges due to the management of the large number of network parameters are discussed in Section 9.1.1. The challenges of executing DNN models on a target hardware platform in a sequential layer-by-layer flow is discussed in Section 9.1.2. The limited scope of monolithic DNN accelerators is discussed in Section 9.1.3.

9.1.1 Storing a Large Number of Network Parameters On-chip

Most prior research has focused on the development of efficient hardware that computes target DNN models, where an accurate characterization of the on-chip and off-chip memory requirements is not satisfactorily performed [53, 63, 196, 253]. Recent research proposed maximizing the size of the on-chip memory of DNN accelerators as a means to improve the performance and energy efficiency by avoiding expensive off-chip data transfers [53, 253]. In addition, several techniques are explored to reduce the memory footprint of deep neural networks [53]. The memory requirements of DNN accelerators for several neural network models are characterized with an emphasis on

the size of the on-chip memory and the off-chip memory bandwidth, where results indicate that increasing the size of the on-chip memory improves the performance, bandwidth, and energy efficiency [53]. Recently, a memory management technique for DNN accelerators was proposed, where the activation memory of two sequentially adjacent layers are intentionally overlapped, as opposed to ping-pong buffering where the input and output activations of two consecutive layers are temporarily stored into two dedicated memory blocks for layer-by-layer processing [253]. By overlapping the activation memory, the energy overhead due to the on-chip memory of the DNN accelerators is reduced [253].

As the DNNs are implemented with deeper and larger networks to improve accuracy, the storage of a greater number of parameters in a combination of on-chip and off-chip memory is required. The increasing need for additional on-chip memory poses challenges on power and resource constrained DNN accelerator SoCs, where all or a majority of the network data (activations and weights) are stored in on-chip memory as discussed in Section 5.4 [53, 253]. Both the energy per access and the latency are reduced by storing the activations (inputs and outputs) and weights in on-chip memory [53]. The performance and energy efficiency of state-of-the-art DNN accelerators are, therefore, constrained by the on-chip memory, where the energy consumed by a unit operation of the on-chip memory is $6\times$ greater than the energy required by a unit operation of computation [239, 253]. Activations and weights are conventionally

stored in either a global on-chip memory and/or within each PE in separate local memory [53]. As recent research has targeted improving the breadth of models executed on a DNN accelerator, the minimum and maximum memory requirements vary by orders of magnitude for different neural network models [53, 196, 253]. Therefore, as the number of DNN models and the size of the neural networks increases, a greater portion of the total power consumption of a DNN accelerator is consumed by the on-chip memory [59, 63, 196]. In addition, as the size of the on-chip memory increases, the amount of unused memory for the implementation of smaller models at any given time results in a significant amount of energy loss due to leakage current.

Algorithm 5 An algorithm that implements convolution for n number of CNN layers [253].

```
1: for Layer = 0 to n
2:     for y_out = 0 to Y_out
3:         for x_out = 0 to X_out
4:             for c_out = 0 to C_out
5:                 for k_y = 0 to K_y
6:                     for k_x = 0 to K_x
7:                         for c_in = 0 to C_in
8:                             x_in = x_out · S_x − P_x
9:                             y_in = y_out · S_y − P_y
10:                            out(x_out, y_out, c_out) += in(x_in+k_x, y_in+k_y, c_in)·k(k_x, k_y, c_in)
11:                        end
12:                    end
13:                end
14:            end
15:        end
16:    end
17: end
```

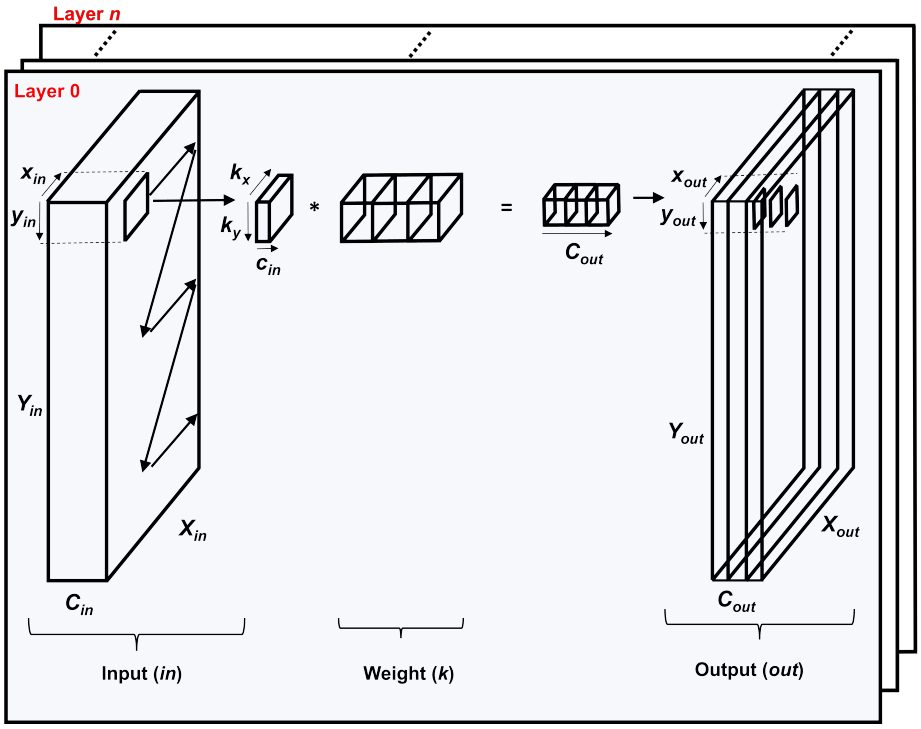


Fig. 9.1: Layer-by-layer processing of the convolution operation, where the algorithm implementing convolution for n number of CNN layers is given by Algorithm 5. A two-dimensional convolution operation of a CNN layer is shown. [253].

9.1.2 Layer-by-Layer Processing of Neural Network Models

DNN models are composed of several convolution layers, fully connected layers, and activation layers, where the convolution layers are the most computation and memory intensive [343]. The algorithm executing the convolution operation and a depiction of an implementation of a convolutional neural network (CNN) with n number of layers are provided as, respectively, Algorithm 5 and Fig. 9.1 [253]. In each layer, the input feature map $in(X_{in},\ Y_{in},\ C_{in})$ is convolved with C_{out} number of weight kernels $k(K_x,\ K_y,\ C_{in})$ to produce an output feature map $out(X_{out},\ Y_{out},$

C_{out}). The parameters X_{in}, X_{out}, Y_{in}, Y_{out}, C_{in}, C_{out}, K_x, and K_y represent, respectively, the width of the input, the width of the output, the height of the input, the height of the output, the number of input channels, the number of output channels, the width of each filter, and the height of each filter. For Algorithm 5, the parameters S_x, S_y, P_x, and P_y represent, respectively, the stride in the x direction, the stride in the y direction, the padding of zeros in the x direction, and the padding of zeros in the y direction. For the n-layer convolution operation shown in Fig. 9.1 [253], each weight kernel is convolved with the input feature map by sliding in the x and y directions. Once the computation of the first layer is complete, the output is provided to the second layer, where the same computation is repeated. Therefore, the convolutional neural network performs layer-by-layer processing, where the output from a given layer is applied to the input of the subsequent layer [53, 253]. The layer-by-layer computation and data access pattern is common to all CNNs regardless of the implemented model, executing application, and implemented hardware architecture. However, the layer-by-layer computation poses challenges when optimizing the utilization of the processing elements and the on-chip memory.

For a given CNN model, multiple layers exhibit different shapes (stride, padding, number of channels, and dimension of inputs, outputs, and weights), which results in disparate configurations of the PE array and memory. Therefore, the execution of models on an accelerator SoC must be dynamically assigned through an efficient

dataflow [194]. Several optimized dataflows including row stationary (RO), output stationary (OS), and weight stationary (WS) were proposed to maximize the utilization of PEs across all layers [194]. However, a given dataflow optimizes the target hardware only for a sub-set of layers and models. Therefore, the challenge of efficiently utilizing on-chip memory resources and PE arrays remains. Specifically, available PEs are not best utilized for all layers, which results in the need for dynamic memory allocation for each layer.

In addition, the on-chip memory utilization at any given time changes when executing layer-by-layer processing. As noted in Section 5.4 and Section 9.1.1, the size of the on-chip memory of current hardware accelerators is larger [53, 194, 239, 253, 255]. If the weights and/or activations of all layers are stored in on-chip memory, only a fraction of the total on-chip memory is utilized when executing a particular layer [253]. Therefore, a significant amount of energy is lost due to the leakage current through the idle memory banks, as leakage energy increases proportionally to the size of the circuits.

9.1.3 Limited Scope of Monolithic DNN Accelerators

The DNN accelerators are typically composed of an array of monolithic processing elements (homogeneous accelerator architecture where all PEs are tasked with executing a single model at any given time), and all PEs are tied to a single power

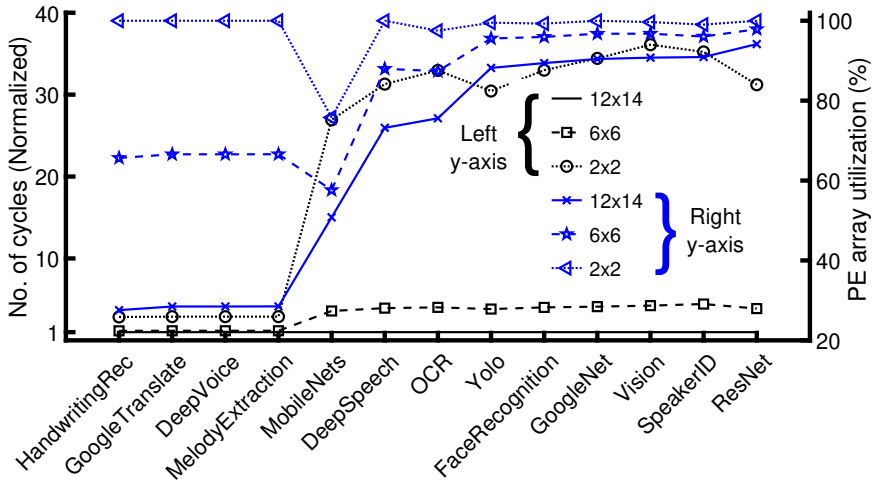


Fig. 9.2: Characterization of the utilization of the PEs and the average number of cycles to execute a set of DNN models for array sizes of 12×14, 6×6, and 2×2.

domain [53, 59, 61–63, 194, 239, 253, 255]. The implementation of a monolithic DNN architecture limits any improvement in the energy efficiency and the performance of the system as i) all PEs are allocated for a particular model regardless of the actual hardware requirements of the executing model and ii) the same supply voltage is applied to all PEs regardless of the specific latency and throughput requirements of the executed application.

The computational requirements, size of the on-chip memory, and the memory bandwidth vary by multiple orders of magnitude for different neural network models as well as across layers within a given model [53, 253]. Maintaining a high energy efficiency when implementing a monolithic DNN accelerator is a challenge as a given dataflow does not map diverse layers and models optimally to the available hardware resources. In this work, an Eyeriss-like architecture is characterized using a cycle

accurate neural processing unit (NPU) simulator [194, 216]. The architecture is analyzed across a set of DNN application models and for three PE array sizes (12×14, 6×6, and 2×2). A weight stationary (WS) dataflow is applied to all array sizes and models [239]. The average utilization of the PE array and the average number of cycles required to complete execution of a diverse set of models used for applications including vision, object detection, and speech recognition are characterized, with results as shown in Fig 9.2. The number of cycles of execution is normalized to the number of cycles required by the 12×14 array. For the 12×14 PE array, the average number of cycles required to complete execution of HandwritingRec, GoogleTranslate, DeepVoice, MelodyExtraction, MobileNet, DeepSpeech, OCR, Yolo, FaceRecognition, GoogleNet, Vision, SpeakerID, and ResNet is, respectively, 0.245K, 2.84K, 2.48K, 3.627K, 206K, 1639K, 127K, 1735K, 423K, 174K, 1823K, 6007K, and 462K. The average PE utilization is less than 90% for most of the models when the PE array size is 12×14, while the PE utilization significantly increases for the smaller array sizes of 6×6 and 2×2. Underutilization of PE arrays across hundreds to thousands of cycles results in a significant reduction in the energy efficiency of the accelerator due to increased leakage current. Conversely, an increase in the utilization of PEs results in a significant improvement of the total energy efficiency of the DNN accelerator. However, the reduction in the size of the array results in an increase in the average number of cycles needed to complete execution of the models by 1.16× to

4.43× and 2.86× to 36× for, respectively, an array of size 6×6 and 2×2. The overall power-performance trade-off is, therefore, improved when the different models and layers are optimally mapped to a heterogeneous PE sub-array.

State-of-the-art edge devices execute multiple applications that concurrently run in the background. As an example, edge devices performing augmented reality (AR) require concurrent execution of object detection, speech recognition, pose estimation, and hand tracking [196, 214]. In addition, due to the increasing complexity and the greater variety of DNN based workloads executed on edge devices, dynamic allocation of resources and the computational loads is required [196, 214, 217, 343]. Traditional DNN accelerators with monolithic architectures optimized to efficiently execute only a sub-set of models are not suitable for current trends in applications that require the use of a diverse set of DNN models. Recent research proposed flexible and heterogeneous DNN accelerators to improve the performance and energy efficiency of edge devices simultaneously running a diverse set of DNN models [196, 214, 217]. The heterogeneous DNN accelerators are composed of multiple sub-arrays of PEs each optimized for different layer shapes and operations [196, 214, 217]. Each sub-array of PEs is mapped to a dataflow that optimizes the utilization of resources and improves the overall power-performance trade-off.

In addition, monolithic DNN accelerators, where all PEs share a common power domain, limit any improvement in the energy efficiency of the circuit provided by

techniques such as fine-grain dynamic voltage scaling, adaptive voltage scaling, and power gating [59, 215, 238, 257]. For example, more recently, an inference processor was implemented for improved energy efficiency, where all of the PEs operate at a near-threshold voltage of 0.4 V for the entire execution of the DNN [255]. While the power consumption is lower at 0.4 V as compared to 1.2 V, the operating frequency of 60 MHz is also significantly lower, which limits the inference processor to the implementation of only a sub-set of DNN models [255]. The highly constrained monolithic architecture is, therefore, not well suited for the execution of a diverse set of DNN models as the throughput, energy efficiency, and execution time of the accelerator are not dynamically adjustable.

The operating frequency of most state-of-the-art DNN accelerators is limited by the memory bandwidth despite the opportunity of running the computational units at much higher frequencies [63, 194, 255, 256]. For example, the Eyeriss accelerator implemented in a 65 nm CMOS fabrication process operates at a clock frequency of 200 MHz, where each PE consists of either a 16-bit MAC [194, 256] or two 8-bit MACs [63]. In addition, an inference processor implementing 848 KB of SRAM memory operates at a 120 MHz frequency for a supply voltage of 0.7 V in a 65 nm CMOS technology [255]. In both cases, the 65 nm technology permits much higher operating frequency. Therefore, there are opportunities to improve the overall system energy efficiency without significantly impacting the performance of the accelerator

SoC by operating the computational units (MACs) at a lower supply voltage, while operating memory at a higher supply voltage.

9.2 Leakage Reuse in Memory

Leakage reuse is a technique where the leakage current from an idle circuit block or core in a system-on-chip (SoC) is reused to deliver power to an active circuit block or core within the same SoC [272, 334, 335]. Prior research applied the leakage reuse technique to logic circuits [334, 335]. Recently, circuits and algorithms for simultaneous implementation of leakage reuse and power gating were proposed [272]. In this paper, however, the leakage reuse technique is applied to on-chip memory (SRAM). Idle memory banks from which leakage current is reused are described as *donors* and computing units (PEs or MACs) to which reused charge is delivered as *receivers*. Unlike the prior techniques, in this paper the leakage current from memories (donors) is reused to supply power to computing units (receivers) that implement a near-memory computing architecture. The stored data is not affected when leakage reuse is applied, which is validated through SPICE simulation (more discussion is provided in Section 9.2.2). The primary advantages of applying the leakage reuse technique include a) a reduction in the leakage current of the donors, which improves the overall energy-efficiency of the system, b) the leakage energy, which is otherwise considered wasted energy, is recycled to deliver power to computing units, and c)

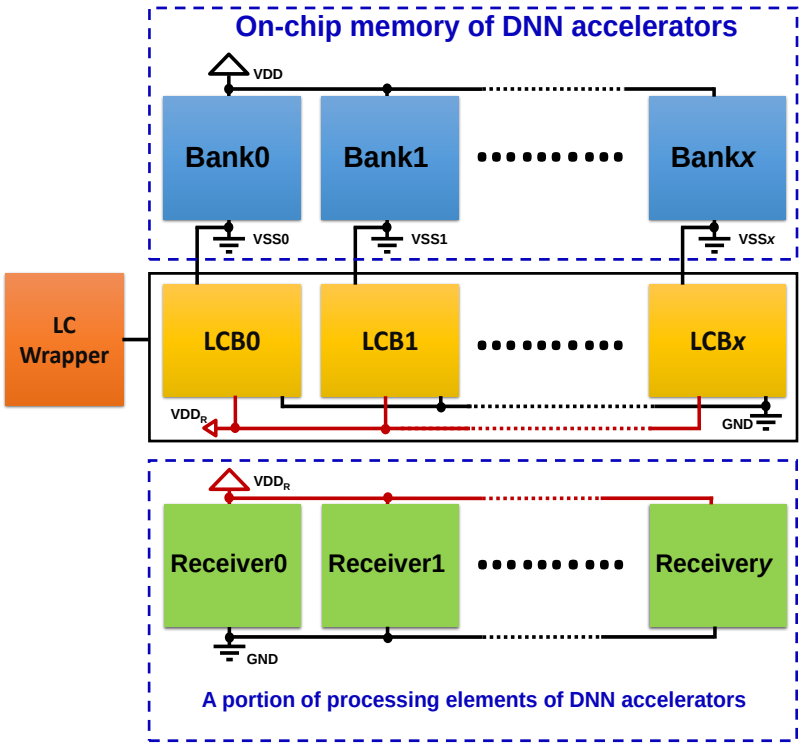


Fig. 9.3: An overview of applying the leakage reuse technique at the memory bank level, where leakage current from x number of SRAM banks (donors) provide power to y number of computing elements (receivers).

voltage regulators are not required to generate and regulate the supply voltages of the receivers, which reduces the area and power overhead of the IC.

9.2.1 Bank-level Leakage Reuse

Large state-of-the-art on-chip SRAM memories are composed of many smaller banks. The smaller distributed memory blocks allow for selective activation of banks that are required for specific workloads, which results in a significant reduction in the overall power consumption of the SoC [254, 344]. In addition, for conventional

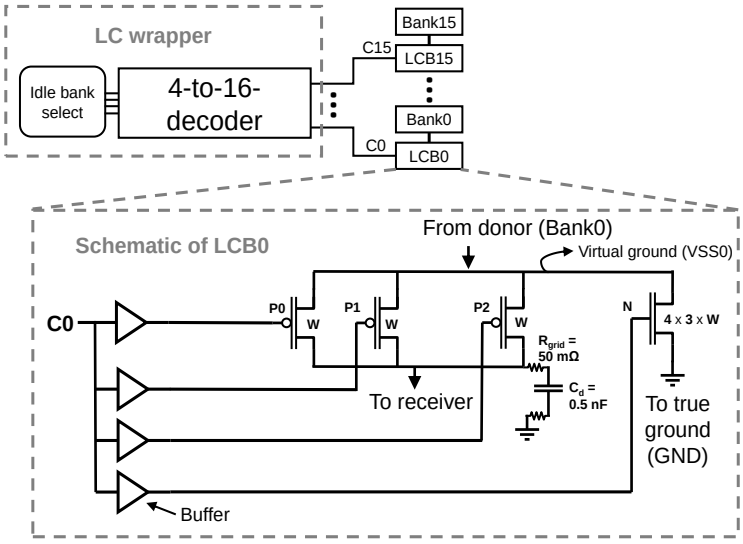


Fig. 9.4: Schematic representation of the leakage control blocks and the LC wrapper.

topologies, the maximum allowed bit cells per row or column in an SRAM array is limited to no more than 300 due to the increased RC delays on the wordlines and bitlines [344]. Furthermore, the size of the decoders is increased correspondingly, which results in greater delay as the size of the wordline or array size is increased. The greater decoder delay is in addition to large distributed RC load on the wordlines and bitlines. The use of banks also increases the throughput as *memory interleaving* is permitted, which is a technique where the addresses of large on-chip or off-chip memory are evenly distributed across many smaller memory banks [254]. The benefit of memory interleaving is that the execution of read/write operations and the waiting time to re-execute read/write operations on a specific memory address is reduced, which increases the data throughput [254].

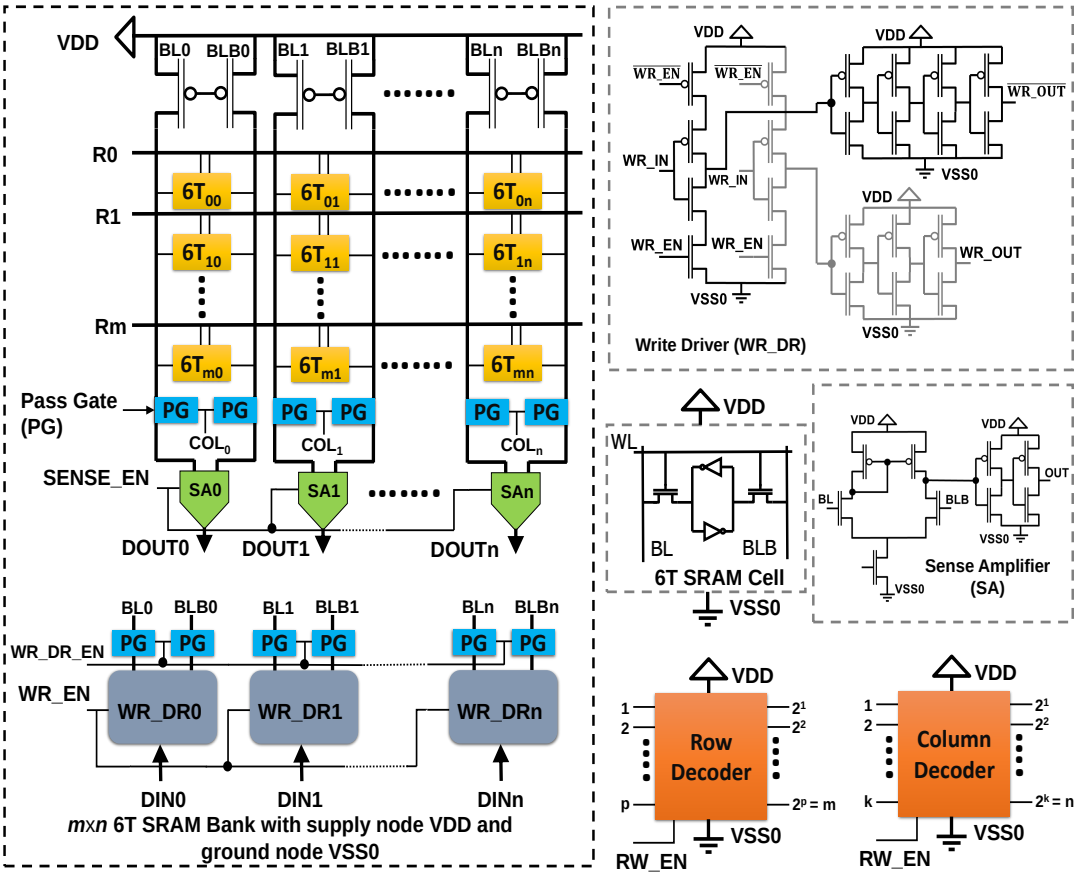


Fig. 9.5: Circuit representation of an implemented SRAM bank (Bank0) with a cell array of m rows and n columns, where the supply and ground node of Bank0 is, respectively, *VDD* and *VSS0*.

Given the significant benefits of using SRAM banks to improve power consumption, latency, and throughput, in this paper the leakage reuse technique is applied to the SRAM banks, which are the smallest units the technique is implemented on. An overview of applying the leakage reuse technique to SRAM banks is provided through the block level representation shown in Fig. 9.3, where the leakage current from x number of donors (memory banks) is recycled to generate supply voltages for

y number of receivers (computing units). Each donor consists of a switching fabric described as a leakage control block (LCB) that controls the leakage flow between donor and receiver(s), while the leakage control wrapper (LC wrapper) provides the control signals to the LCBs.

The circuit implementation of the LCB and the LC wrapper is shown in Fig. 9.4. Each LCB consists of PMOS and NMOS switches that allow current in, respectively, leakage reuse and active mode. Multiple PMOS switches are utilized to allow for finer control of the generated voltage for the receiver. The LC wrapper sets the control bits based on the activity of the donor(s) and receiver(s) and the amount of current required for the receiver to generate a given supply voltage. A binary 4-to-16 decoder is implemented that sets the idle donor memory bank(s) for leakage reuse. The NMOS transistor N of the LCB connects the virtual ground node *VSS0* of Bank0 to the true ground node *GND* when Bank0 is in active mode. The PMOS transistors $P0$ through $P2$ tie the *VSS0* node to the power network of the receiver when Bank0 is idle (hold state). The control signal $C0$, which is connected to the gates of the PMOS and NMOS transistors in Fig. 9.4, is set to 0 or 1 when Bank0 is operating in either idle mode or active mode, respectively. The transistors $P0$ to $P2$ must be large enough to sufficiently conduct the leakage current through the virtual ground (receiver power network) within a given clock period and without resulting in significant resistive voltage drop, while the threshold voltages of transistor N must be large enough to

prevent leakage loss during the idle mode operation of the memory banks.

The circuit representation of Bank0 is shown in Fig. 9.5, where each bank has a separate virtual ground node. Note that the circuit structure of all remaining banks is identical to that of Bank0. Each bank consists of an $m \times n$ SRAM cell array, row/column decoders, sense amplifiers, and driver circuits as shown in Fig. 9.5, where in this work 16 rows ($m = 16$) and 16 columns ($n = 16$) are implemented in each bank.

Table 9.1: Characterization of leakage power and access time of SRAM with standard-V_t transistors, high-V_t transistors, and transistors with 25% longer channel length.

	One 6T bit-cell	**16×16 cell array**	**16×16 memory bank**	**Access time of a 6T memory cell**	
				Read	**Write**
Standard-V_t and L	95.25 pW	104.5 W	113.3 W	0.179 ns	0.152 ns
High-V_t	40.51 pW	39.11 W	47.91 W	0.195 ns	0.172 ns
25% longer gate length (L)	60.08 pW	77.74 W	86.54 W	0.192 ns	0.166 ns
Total power of a 8b MAC (@0.45 V)	14.53 W				

Existing techniques that minimize the leakage current of SRAM memory cells are also considered in this research. The leakage current of a single 6T SRAM cell, a 16×16 cell array, and a 16×16 SRAM macro are characterized, with results as listed in Table 9.1. Three scenarios are evaluated: a) an SRAM cell that includes transistors with standard threshold voltage V_t and standard gate length, b) an SRAM cell that includes high-V_t transistors, and c) an SRAM cell that includes transistors with 25% longer gate length. Note that high-V_t transistors are utilized in both the pull-up and

the pull-down networks of both inverters of the 6T SRAM cell to produce the maximum reduction in the leakage current as reported in [57, 58]. Despite the reduction in the leakage current of the SRAM cells that include transistors with high-V_t and long gate length, a significant amount of power is dissipated as leakage, as indicated by the results listed in Table 9.1. Specifically, the leakage power of a 6T SRAM is 95.25 pW, 60.08 pW, and 40.51 pW when implemented with transistors of, respectively, standard threshold voltage V_t and gate length, 25% longer gate length, and higher-V_t. In addition, SRAM cells that are implemented with transistors of higher threshold voltage and longer gate length result in greater access time for both read and write operations as indicated by the results listed in Table 9.1, which negatively impacts the overall performance of the accelerator. Therefore, SRAM cells with both standard V_t and gate length are used in this work. However, the leakage reuse technique is applicable to the SRAM banks that are implemented with transistors of higher threshold voltage and longer channel length by adjusting the ratio of the donor area to receiver area at design time. Note that the leakage power of the SRAM banks is significantly larger than the total power of an 8-bit MAC even with transistors of higher-V_t and longer channel length.

Despite the reduction in the leakage current in more advanced process nodes, leakage current loss is not eliminated, and a significant amount of power is still dissipated as leakage. Specifically, the leakage power of an 8T SRAM cell is 114.44 pW and 27.5

pW for a 6T SRAM cell in, respectively, a 32 nm tri-gate CMOS and 7 nm FinFET process [17, 271]. The proposed technique generates the receiver supply voltages by tuning the size and configuration of the leakage control blocks, the LC wrappers, and the ratio of the donor area to the receiver area at design time. Therefore, even at advanced technology nodes, the leakage reuse technique remains applicable. Moreover, the complex circuit techniques available in state-of-the-art on-chip SRAM architectures including the sharing of decoders, the sharing of sense amplifiers, and the use of higher order address bits to apply chip select and bank addressing are not considered in this paper as 1) the design of the SRAM architecture is not a primary objective of this work and 2) the use of the SRAM architecture shown in Fig. 9.5 provides sufficient circuit details for the proof of concept described, where the inclusion of more complex techniques only increases the area and leakage current.

9.2.2 Functionality of Donors During Leakage Reuse

The functionality of the SRAM banks (donors) and the computing units (receivers) when implementing the leakage reuse technique is analyzed. Four banks are implemented to characterize the functionality of the donors and receivers when implementing the proposed technique. The 6T SRAM bit cell array and the peripheral circuits of the memory bank are developed in a 65 nm CMOS technology. The results from transient analysis of Bank0, which include the implementation of the leakage

reuse technique, are obtained through SPICE simulation and are shown in Fig. 9.6. A 4-bit address is used to access a smaller sub-array (4×4) of the memory, as shown in Fig. 9.6. At any given time, two banks are assumed active, while the remaining two are assumed idle, where Bank0 and Bank1 (Bank2 and Bank3) share the same control signals. A 4-bit carry look-ahead adder is implemented as the receiver. A 0.45 V supply voltage is generated for the receiver from the leakage current of the two idle SRAM banks. The control signals for the SRAM banks are listed in Table 9.2, where signals for only Bank0 are provided. At the beginning of the control cycle, Bank0 and Bank1 are active (C0 = C1 = 1) and the leakage current from the two idle banks (Bank2 and Bank3) is provided to the adder. A logic high is written into a memory cell located at address 0100, which is followed by a control pulse that changes the state of Bank0 to idle (hold state). Therefore, Bank0 and Bank1 are now available for leakage reuse. During the third control pulse, where Bank0 is in an active state again, the stored data is retrieved correctly. Similarly, a logic low is written to memory address 0000 and is retrieved correctly. Therefore, the proposed technique recycles leakage current from memory banks without causing functional errors and data loss.

The implementation of the leakage reuse technique does not result in a perturbation of the stored bits of the SRAM cell arrays. As an example, a bit cell from Bank0 is analyzed as shown in Fig. 9.7. As the leakage reuse technique is only applied

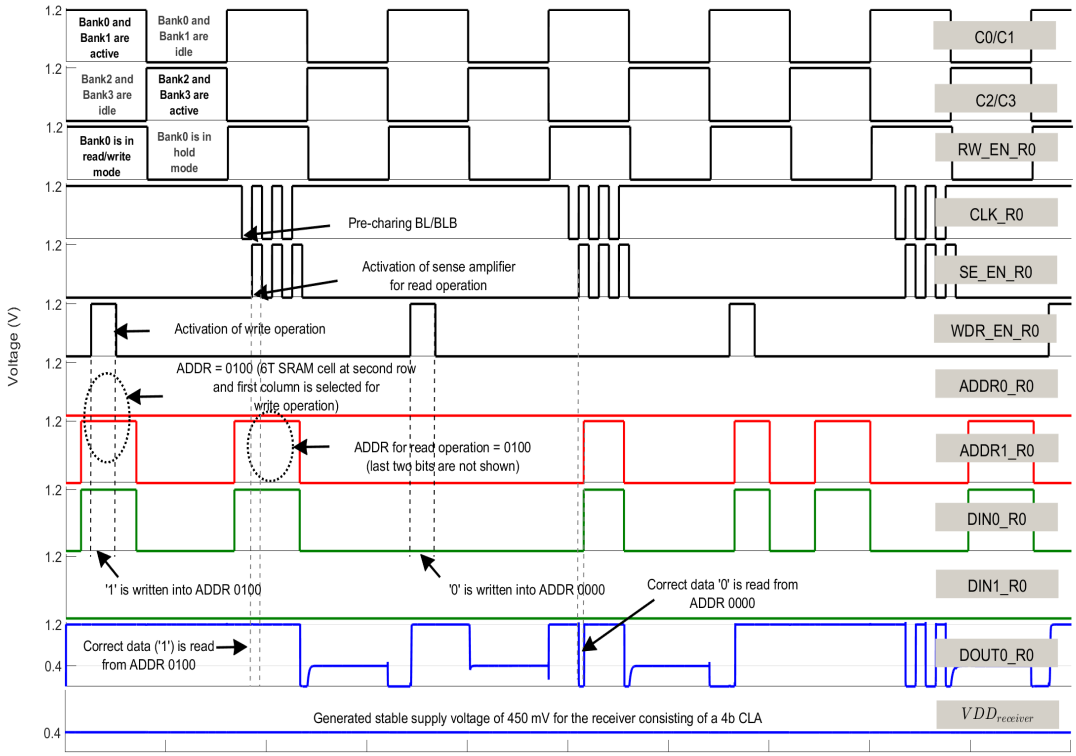


Fig. 9.6: Analysis of the functionality of SRAM Bank0 (donor) as a 4×4 cell array before and after implementing leakage reuse.

during the hold state of the memory bank, data must be retained when the bank is returned to the active operating modes (read/write). When the bit cell stores a logic '1', the input and output of INV2 is, respectively, 1.2 V and 0 V. As the bit cell enters the leakage reuse mode, the output node of INV2 rises to 0.45 V, which results in a corresponding change to the input of INV1 to 0.45 V from 0 V. However, the rise in the input voltage of INV1 does not perturb the output state of INV1 as the NMOS transistor M3 is in cut-off mode ($V_{GS,M3} = 0V$). Therefore, the 1.2 V input to INV2 remains unchanged. Similarly, when a logic '0' is stored on the bit cell, the

Table 9.2: Control signals corresponding to SRAM Bank0.

Signal	Operation
CLK_R0	= 0 : pre-charge = 1 : evaluate
RW_EN_R0 (read/write enable)	= 0 : hold mode = 1: read/write mode
SENSE_EN_R0 (enable sense amplifier)	= 0 : write mode = 1 : read mode
WR_DR_EN_R0 (write driver enable)	= 0 : read/hold mode = 1 : write mode
C0 (leakage control)	= 0 : LR mode = 1 : donor is active

output node of INV1 rises to 0.45 V from 0 V. In this case, the NMOS transistor M5 remains in cut-off mode, which prevents any perturbation of the output node of INV2. Therefore, stored data on the SRAM cells is retained even with implementation of the leakage reuse technique.

9.2.3 Functionality of Receivers During Leakage Reuse

The functionality of the computing units (receivers) when implementing the leakage reuse technique is analyzed. A 4-bit carry look-ahead (CLA) adder that operates at a supply voltage of 0.45 V is implemented as the receiver. The supply voltage of the adder is generated from the leakage current of two memory banks. The results from transient analysis obtained through SPICE simulation are shown in Fig. 9.8, where two 4-bit numbers (A and B) are added to produce the output ADD_OUT. Despite the need to change donors while applying leakage reuse, a stable voltage of 0.45 V is maintained at the receiver.

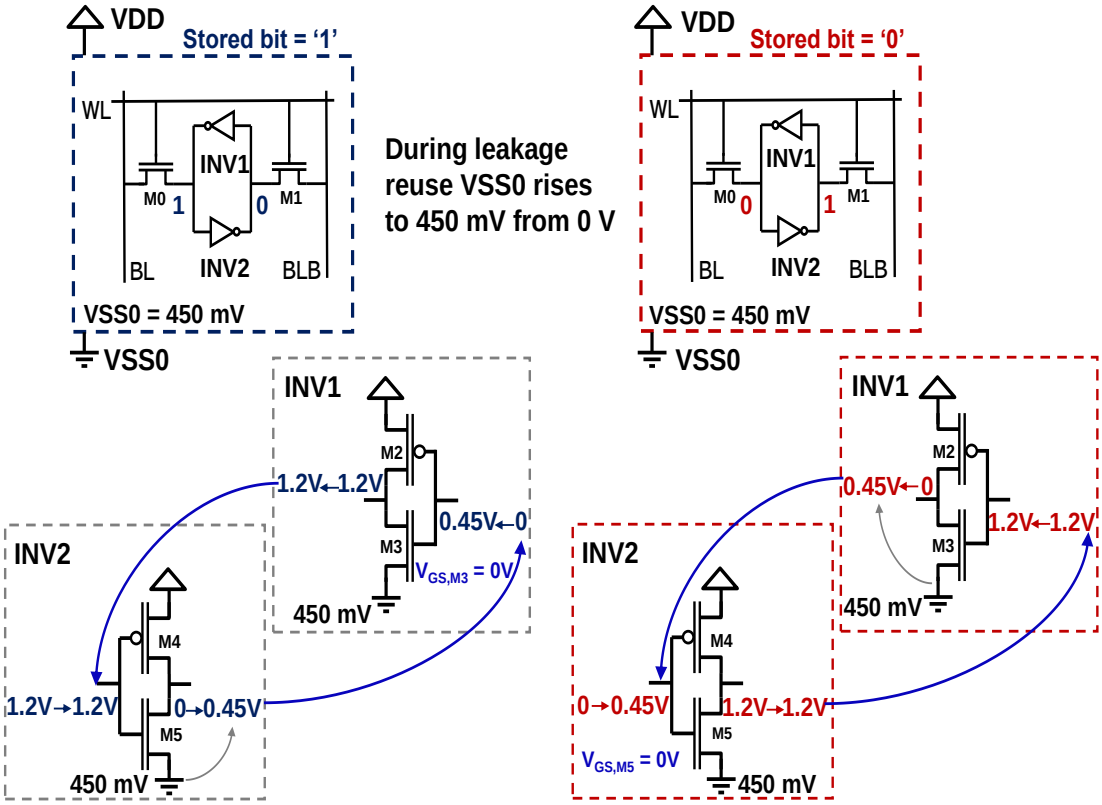


Fig. 9.7: Intrinsic data retention of SRAM cells when accounting for the implementation of the leakage reuse technique.

In addition, the changes in adder inputs, which represents workload variation, incur minor voltage drops, which is shown through the magnified view of the $VDD_{receiver}$ signal. However, the $VDD_{receiver}$ signal returns to it's target value due to the added on-chip decoupling capacitor of size 0.5 nF as shown in Fig. 9.4. The adder is implemented with a single-stage pipeline and, therefore, requires a one clock cycle delay to provide a logical output.

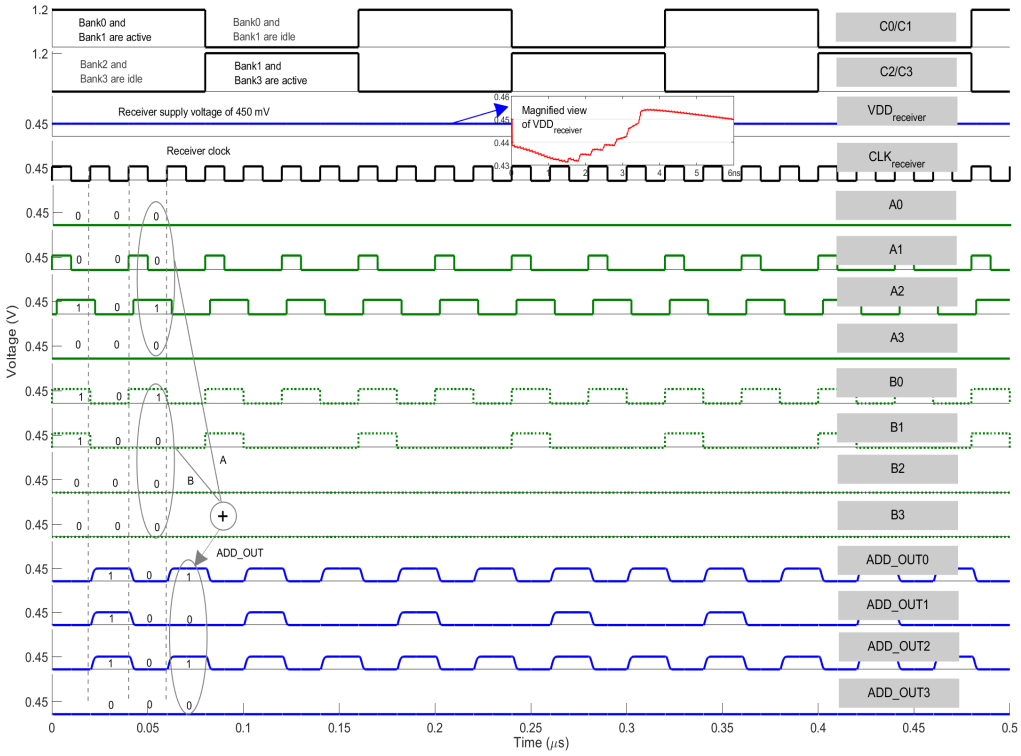


Fig. 9.8: Functional verification of a 4-bit carry look-ahead adder (receiver) that is supplied current from the implementation of the leakage reuse technique.

9.3 Power Management of DNN Accelerator by Reusing SRAM Leakage

In this research, a novel power management technique is developed through controlled charge recycling from on-chip SRAM memory cells during the *hold state* to supply current to processing elements of the DNN accelerators. The proposed technique implements a near-memory computing architecture by bringing the computations units closer to the on-chip memories. In addition, the proposed technique

implements a heterogeneous and multi-voltage domain DNN accelerator, where multiple PEs are grouped to form N_{SA} number of sub-arrays that perform simultaneous execution of N_{SA} number of models.

The leakage current of idle SRAM banks or blocks is reused to supply current to the computing units of the DNN accelerator. Recently, many variants of custom DNN ASICs have been proposed, where regardless of the underlying architecture, the generic structure of each accelerator is composed of 1) an array of processing elements, 2) on-chip memory, 3) off-chip memory, and 4) a communication network such as a network on chip (NoC) [61, 194]. The proposed DNN accelerator that implements near-memory computing with leakage reuse is shown in Fig. 9.9, where the leakage current from idle on-chip memory (SRAM) is reused to deliver power to the computing units within the processing elements. A conventional power delivery system is assumed for the memory banks, where the supply voltage *VDD* is generated and distributed through integrated voltage regulators (IVRs) and/or distributed on-chip voltage regulators (OCVRs) controlled by a power management unit (PMU) [345].

The detailed description of the proposed heterogeneous multi-voltage domain DNN accelerator architecture is discussed in Section 9.3.1. The framework to evaluate the proposed technique is described in Section 9.3.2, which is followed by a discussion of simulation results in Sections 9.3.3, 9.3.4, and 9.3.5. In addition, a novel technique

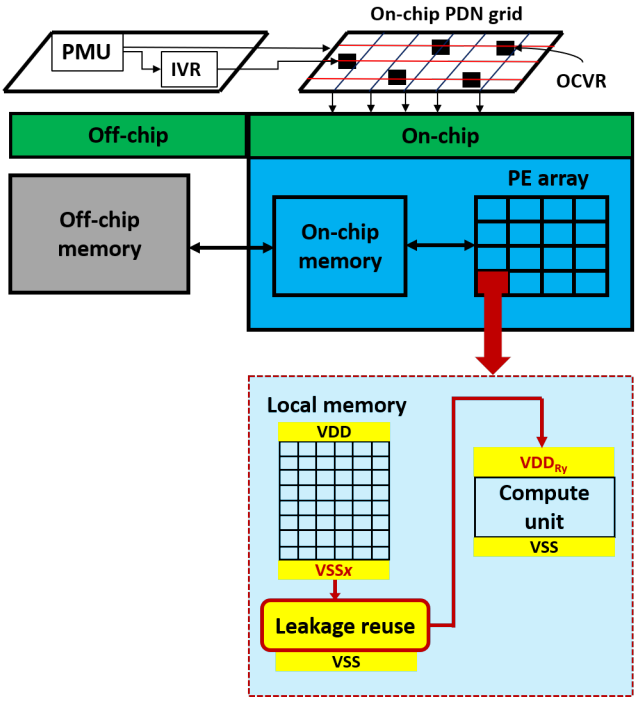


Fig. 9.9: Architecture of the proposed multi-voltage domain DNN accelerator implementing both near-memory computing and leakage reuse.

to maintain a stable supply voltage from the recycled leakage current of the memory cells is discussed in Section 9.3.6.

9.3.1 Proposed Heterogeneous DNN Accelerator Architecture Implementing Near-Memory Computing through Leakage Reuse

A multi-voltage domain heterogeneous DNN accelerator architecture is proposed to address the limitations of monolithic DNN accelerators and the energy loss due to the leakage current of on-chip memory. The proposed architecture implements both

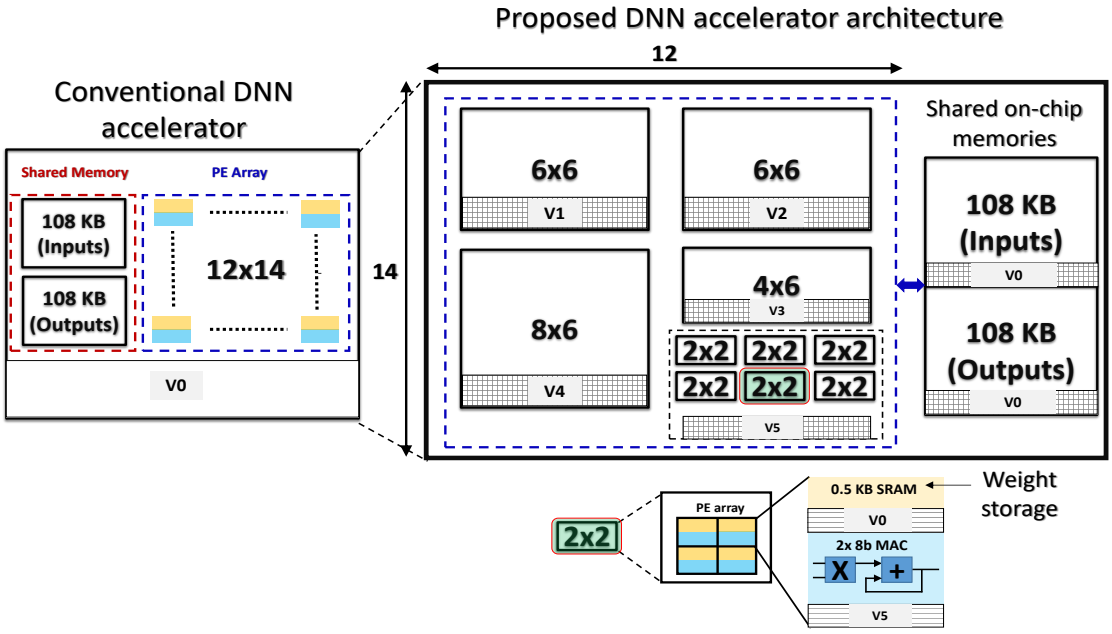


Fig. 9.10: Block level representation of the proposed multi-voltage domain heterogeneous DNN accelerator based on the Eyeriss architecture.

near-memory computing and leakage reuse. The architecture of the DNN accelerator is shown in Fig. 9.10, where the accelerator is composed of multiple sub-arrays each operating in a separate voltage domain, as opposed to established architectures that implement one large PE array with a single voltage domain V0 as shown for the conventional DNN accelerator of Fig. 9.10. The input and output activations are stored in separate global on-chip memory, where the total size of each memory block is 108 KB. The weights required by the multiple layers are stored in an on-chip memory of 0.5 KB within each PE. The total memory to store weights for 168 PEs is 84 KB.

The proposed technique implements near-memory computing within a given PE by generating the supply voltage of the MAC units from the leakage current of the adjacent idle memory blocks that store the weight parameters. Within a processing element, the idle memory banks from all non-active layers of the DNN accelerator are utilized for leakage reuse, where at any given time a single layer and the corresponding weights are active. When any of the sub-arrays execute a given layer L, the weights associated with only layer L are in use, while the remaining memory cells storing weights are idle within each PE of a given sub-array.

The total number of PEs, the number of MACs per PE, the size of the memory storing the input and output activation values, and the size of the on-chip memory per PE are listed in Table 9.3, where the hardware specifications are similar to that required by the Eyeriss architecture [63, 256]. However, unlike the Eyeriss architecture, the PEs for the proposed heterogeneous DNN accelerator are clustered into multiple sub-arrays with independent voltage domains. Therefore, each sub-array of PEs executes a separate DNN model with an optimized performance-energy operating point. The block level overview of a single PE is shown in Fig 9.11, where each PE includes 0.5 KB of on-chip SRAM and two fixed-point 8-bit MACs with two-stage pipelines that produce two multiply-and-accumulate results per cycle.

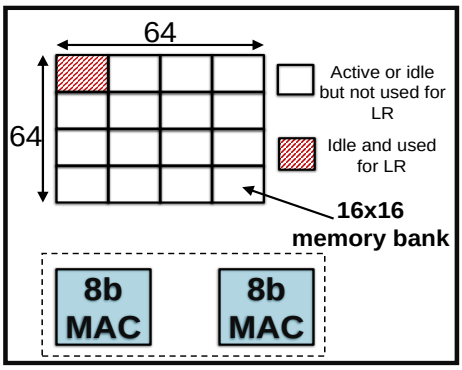


Fig. 9.11: A single processing element with a 0.5 KB on-chip memory and two 8-bit MACs. The 0.5 KB memory is implemented as 16 16×16 SRAM banks, where at any given time one bank is utilized for leakage reuse.

Table 9.3: Hardware specification of the proposed heterogeneous multi-voltage domain accelerator.

Process technology	TSMC 65 nm LP
Total on-chip SRAM	300 KB
Total number of PEs	168
Number of MAC units per PE	2
Number of PE sub-array	10
Number of voltage domain	6
Shared memroy	216 KB
On-chip SRAM per PE	0.5 KB
Clock frequency	200 MHz
Combined peak throughput when all PE sub-arrays operate at 0.45 V	14.686 GOPS
Arithmetic precision	8-bit fixed point
Operating voltage	Memory: 1.2 V MAC: 0.45 V

9.3.2 Evaluation Framework

The proposed multi-voltage domain heterogeneous DNN accelerator consists of a total of 168 PEs and 300 KB of on-chip SRAM memory. The accelerator is implemented and characterized through SPICE simulation in a 65 nm CMOS technology. Each 8-bit MAC unit consists of a radix-4 booth multiplier and a carry look-ahead

adder. Six voltage domains (V0 to V5) are implemented as shown in Fig. 9.10 for the selection of the optimal power-performance point. However, two voltages are applied for the characterization of the heterogeneous accelerator, where voltage domain V0 is set to 1.2 V, and voltage domains V1, V2, V3, V4, and V5 are all set to 0.45 V. Therefore, ultra-low voltage scaling is not applied to the on-chip SRAM as 1) both the performance and throughput of DNN accelerators are already limited due to the on-chip memory and any further reductions in the operating voltage of the SRAM leads to significant degradation in the performance [54, 63, 255] and 2) operating the SRAM cells at ultra-low voltages requires specialized techniques that include write assist circuits or dual-supply rail architectures [346, 347], which are not objectives of this work.

The characterization of the proposed heterogeneous DNN accelerator is performed with 168 PEs that are clustered into ten sub-arrays consisting of a) one 8×6 sub-array with a throughput of 3.67 giga operations per second (GOPS), b) two 6×6 sub-arrays each with a throughput of 2.75 GOPS, c) one 4×6 sub-array with a throughput of 1.84 GOPS, and d) six 2×2 sub-arrays each with a throughput of 0.306 GOPS. The number of sub-arrays and the size of each sub-array are chosen such that the different throughput requirements of the DNN models and corresponding layers are met.

Three scenarios are considered to characterize the throughput and energy efficiency of the proposed accelerator, which are the 1) baseline, 2) proposed architecture without leakage reuse, and 3) proposed architecture with leakage reuse. The three scenarios differ based on the implemented power management technique. The baseline includes a single voltage domain V0 set to 1.2 V that provides current to both the memory and MAC units. For the second scenario, the heterogeneous DNN architecture includes two independent voltage domains, one for the on-chip memory and the other for the MAC units, which are set to, respectively, 1.2 V and 0.45 V. For the third scenario, the memory banks are supplied by a 1.2 V source, while the 0.45 V voltage for the MAC units is generated from the leakage current of idle memory banks within each PE. The 0.5 KB of SRAM memory in each PE that stores the weights consists of 16 banks of 32 bytes each in a 16×16 cell array. The on-chip SRAM of each PE is considered as the donor blocks that supply current, while the MAC units are considered as the receivers. Therefore, for the third scenario, the 0.45 V supply voltage for domain V5 is generated from the implementation of the leakage reuse technique, while OCVRs provide the 1.2 V supply to the on-chip SRAM as shown in Fig. 9.9. Through simulation, the leakage current from an idle 16×16 SRAM bank (donor) is determined to sufficiently generate a stable supply voltage of 0.45 V for two 8-bit MAC units (receiver). Therefore, at any given time, only one idle memory bank (16×16) of a PE is utilized for leakage reuse. Note that the leakage reuse technique is

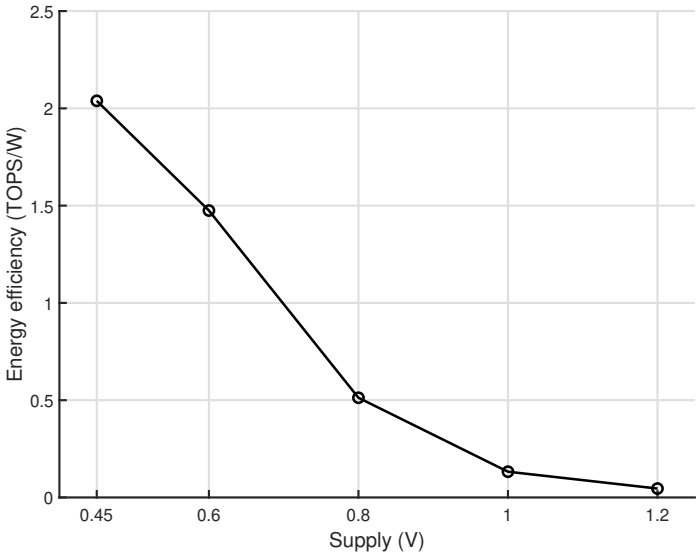


Fig. 9.12: Energy efficiency of the accelerator with a monolithic 12×14 PE array across five voltages.

capable of generating supply voltages that are either larger or smaller than 0.45 V for domains V1 to V4 [272, 334]. In addition, the ratio of the donor area to the receiver area is determined at design time as opposed to run-time due to the significant energy overhead of reconfiguration needed at run-time. Recent research has demonstrated that heterogeneous DNN accelerators exhibit superior energy efficiency as compared to reconfigurable DNN accelerators due to the requirement of per layer reconfiguration and the energy overhead of the switches, interconnect, and controllers needed to implement the reconfiguration [196].

9.3.3 Characterization of a Monolithic Accelerator Architecture Using the MobileNets Model

An accelerator that consists of 336 MAC units within 168 PEs is characterized for energy efficiency, which is represented as tera operations per second per watt (TOPS/W), across the five voltages of 1.2 V, 1 V, 0.8 V, 0.6 V, and 0.45 V with results as shown in Fig. 9.12. The energy efficiency increases as the supply voltage is reduced, with the PE array being 44.5× more efficient at 0.45 V (2.04 TOPS/W) as compared to 1.2 V (0.0458 TOPS/W). Therefore, the leakage reuse technique is evaluated at a supply voltage of 0.45 V.

In addition, the monolithic accelerator architecture is characterized through the implementation of the MobileNets model using a cycle accurate neural processing unit simulator to obtain the number of MAC operations required to execute each of 27 convolutional layers [216, 348]. The simulation provides an analysis of the energy consumed by each layer at voltages of 1.2 V, 1 V, 0.8 V, 0.6 V, and 0.45 V, with results as shown in Fig. 9.13. The number of MAC operations is directly proportional to the completion time of each layer as shown in Fig. 9.13, where the completion time at each voltage is calculated based on the minimum cycle time of the two 8-bit MACs at a given voltage. For example, the Conv27 layer is the most computationally expensive layer of the MobileNets model and requires 3282 MAC operations when executed

on the 12×14 PE array. The energy consumption (completion time) of Conv27 is 15.96 J (0.5 ms), 4.37 J (0.63 ms), 0.73 J (0.97 ms), 0.07 J (3.6 ms), and 0.01 J (14.4 ms) when operating at a supply voltage of, respectively, 1.2 V, 1 V, 0.8 V, 0.6 V, and 0.45 V. Conv24 is the least computationally expensive layer and, therefore, requires only 139 MAC operations to execute, resulting in an energy consumption and a completion time of, respectively, 0.68 J and 0.021 ms at a supply voltage of 1.2 V. Among the 27 convolutional layers, the standard deviation (SD) of the number of MAC operations required is 845. The completion time and energy consumed per layer is also characterized for additional models including GoogleNet, ResNet50, Yolo, and AlexNet. The average mean and average standard deviation across all layers of the five models evaluated in this paper are provided through Fig. 9.14. Note that the results for the completion time and energy consumption are obtained for a supply voltage of 0.45 V. Similar to the MobileNets model, large variation in the MAC operations across all layers is observed for the other models, where the standard deviation of the number of MAC operations required across all layers of Googlenet, Resnet50, Yolo, and Alexnet is, respectively, 1782, 1236, 8813, and 414809. There is significant variation in the required computational resources and completion time across the multiple layers of a single model. The variation further increases when implementing multiple models as discussed in Section 9.1.3. Therefore, there is benefit in computing multiple models on a heterogeneous PE array that accounts for the required number

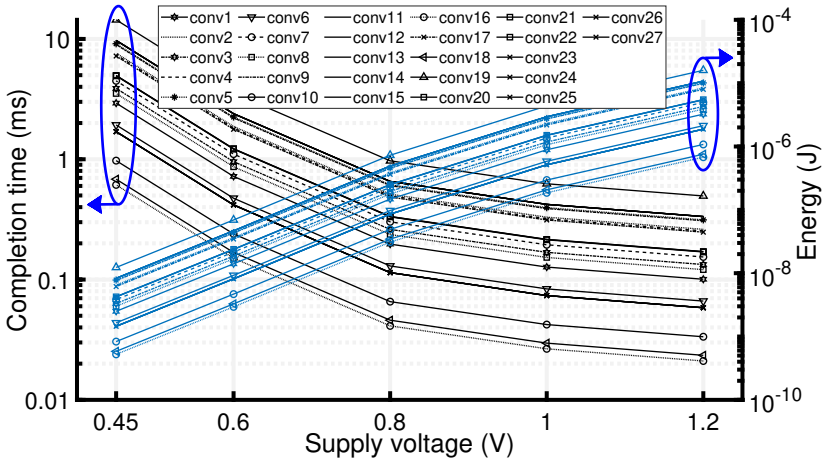


Fig. 9.13: A monolithic 12×14 PE array is characterized for energy consumption and completion time of each layer when executing the 27 convolutional layers of the MobileNets model and operating at each of five different supply voltages.

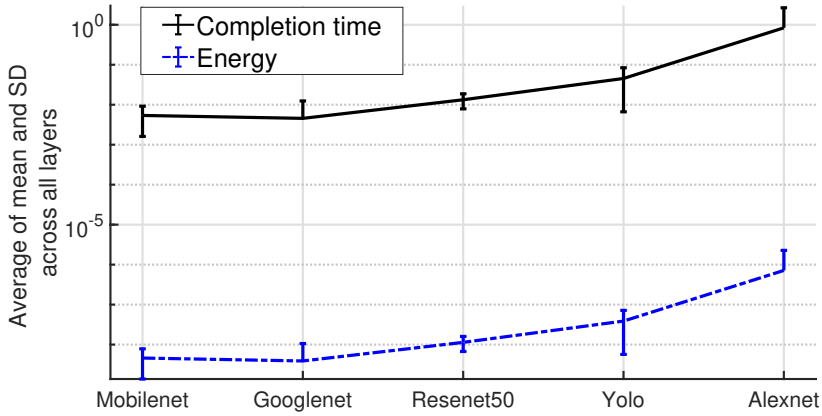


Fig. 9.14: Characterization of the mean and standard deviation of the completion time and energy consumed per layer, which are averaged across all layers of each respective model.

of MAC operations and the latency of each layer of the neural network.

9.3.4 Characterization of Energy Efficiency and Throughput of the Proposed Heterogeneous Architecture that Includes Leakage Reuse

The total power consumption and the throughput of the MAC arrays are characterized for the baseline and the proposed multi-voltage domain heterogeneous DNN accelerator architecture with and without applying leakage reuse, with results as shown in Fig. 9.15. The supply voltage is set to 1.2 V and 0.45 V for, respectively, the baseline and the proposed technique (with and without leakage reuse) across the four sub-array sizes of 2×2, 4×6, 6×6, and 8×6. The relative difference between the three configurations in total power consumption and energy efficiency scales with sub-array size. The power consumption of the baseline is 605.5× and 487.2× that of, respectively, the proposed technique with leakage reuse and without leakage reuse. The throughput of the baseline MAC units operating at 1.2 V is 35× the throughput of the proposed technique with or without leakage reuse as the supply voltage in both cases is set to 0.45 V.

The energy efficiency of the proposed architecture, with and without leakage reuse, is compared with a baseline homogeneous implementation, with results as shown in Fig. 9.16. Note that 1) the total power consumption and delay of the MAC arrays is considered when calculating the energy efficiency for each topology and 2) the leakage

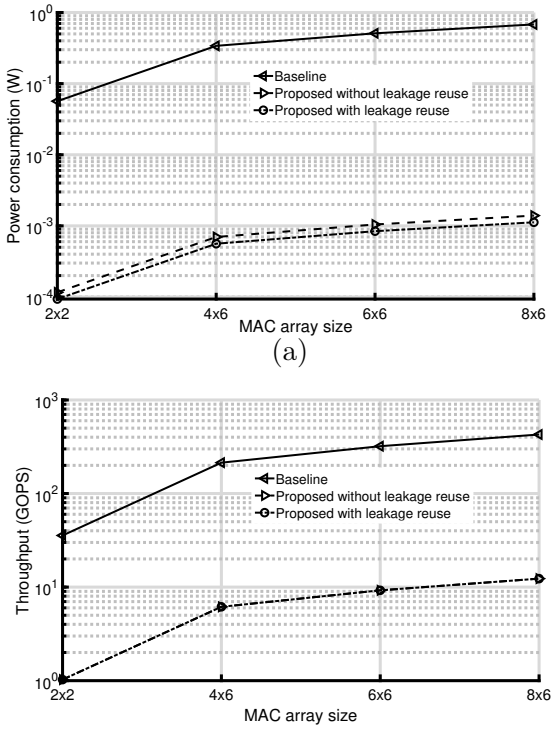


Fig. 9.15: Comparison of a) total power consumption and b) throughput for the baseline, proposed technique without leakage reuse, and proposed technique with leakage reuse.

reuse technique is applied to 6.25% of a processing element. The implementation of leakage reuse on the proposed heterogeneous architecture exhibits a maximum energy efficiency of 3.27 TOPS/W, which is 71.44× and 1.60× greater than the baseline and the proposed architecture without leakage reuse, respectively.

9.3.5 Characterization of Total Power Consumption

The total power consumption of a 2×2 sub-array is characterized, where each processing element within the sub-array consists of two 8-bit fixed-point MACs and

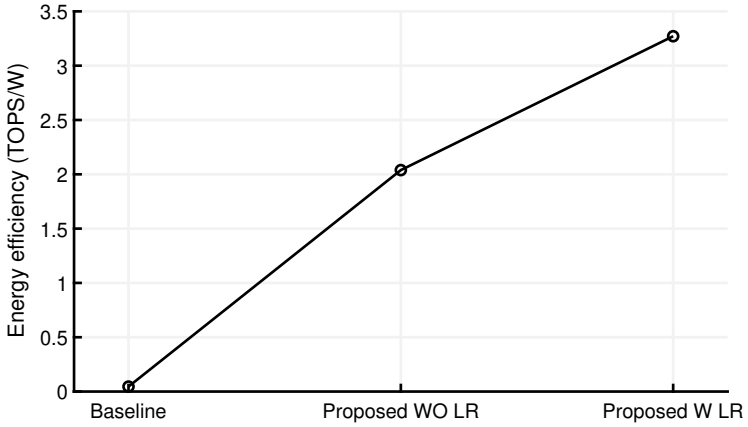


Fig. 9.16: Characterization of the energy efficiency of all MAC sub-array sizes in TOPS/W when assuming all processing elements in each topology are equally loaded and at the same voltage.

a 0.5 KB memory as shown in Fig 9.11. In addition, the total power consumption of a group of six 2×2 sub-arrays (shown in Fig. 9.10) is also characterized. The simulation results of both a single 2×2 sub-array and six 2×2 sub-arrays for the baseline, proposed architecture without leakage reuse, and proposed architecture with leakage reuse are shown in Fig. 9.17. The total power consumption of one 2×2 sub-array is 65.57 mW, 9.04 mW, and 8.73 mW for, respectively, the baseline, proposed architecture without leakage reuse, and proposed architecture with leakage reuse. Note that energy loss due to the voltage regulators is not considered when characterizing the power consumption of the accelerator without leakage reuse. The relative ratio of the power consumption across the three topologies, which are described in Section 9.3.2, is maintained when characterizing the six 2×2 PE sub-arrays. Note that application of the leakage reuse technique to only six 2x2 sub-arrays, which includes a total of 24

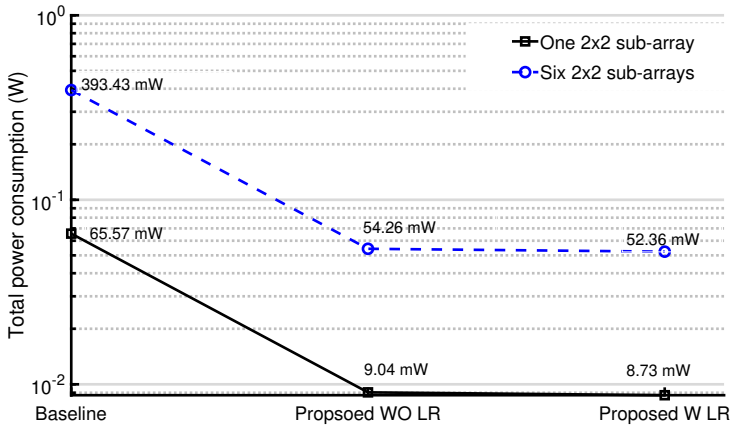


Fig. 9.17: Characterization of the total power consumption of a 2×2 sub-array, where a single PE within each 2×2 sub-array is composed of 0.5 KB memory and two 8-bit MACs.

PEs, results in a 1.9 mW (3.5%) reduction in the total power consumption. For the 24 out of 168 PEs that the leakage reuse technique is applied to, only 24 16×16 memory banks (one bank from each PE) are used for leakage reuse from a total of 2688 (16×168) memory banks in the entire accelerator. Therefore, 6.25% of the memory storing the weights in 24 PEs (24 banks out of 384 banks) is used for leakage reuse, where the 24 memory banks correspond to only 0.89% of the entire 2688 memory banks of the accelerator.

The total power consumption of all PE sub-arrays is characterized for seven different percentages (1%, 5%, 10%, 20%, 30%, 40%, and 50%) of idle on-chip memory used for leakage reuse as shown in Fig. 9.18, where all MACs operate at 0.45 V. The total power consumption includes the on-chip memory (84 KB) and MAC units within each of the 168 PE. The power savings increases as more of the memory used for the

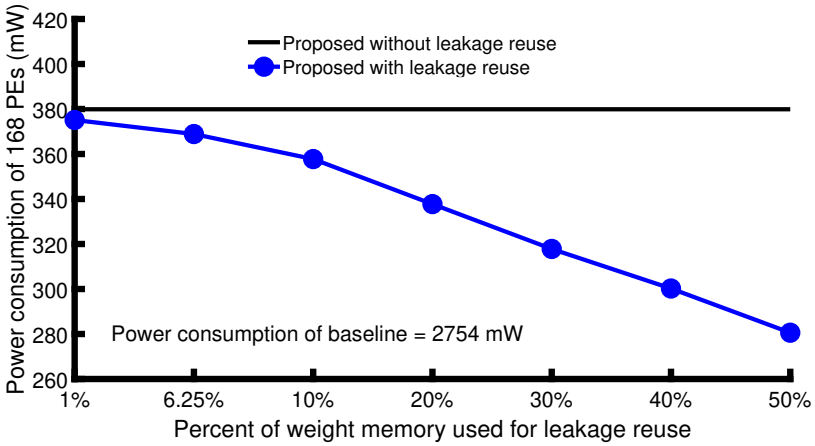


Fig. 9.18: Characterization of total power consumption of the accelerator SoC for four different percentage of weight memory utilization for leakage reuse.

weights is included for leakage reuse. For example, the total power consumption of the accelerator is reduced by 6 mW and 99.4 mW when, respectively, 1% and 50% of the memory is used for leakage reuse.

The components of the total power consumption of the baseline and the proposed accelerators is listed in Table 9.4, where the power is characterized for 50% utilization of the weight memory when leakage reuse is implemented. The proposed accelerator implemented with the leakage reuse technique reduces the power consumption by 26% (99.4 mW) as compared to an accelerator that does not apply leakage reuse. The simulated results are obtained from the execution of the accelerators across 8000 clock cycles and for a clock frequency of 200 MHz. Note that the active area of one 16×16 memory bank is 345.3 μm^2. The total area of the LC wrapper is equivalent to 4.75% of the area of a 16×16 memory bank. The total area of the leakage control

block (LCB) when process, voltage, and temperature (PVT) mitigation circuits are included, as discussed in Section 9.3.6, is equivalent to 2.93% of the area of a 16×16 memory bank.

Table 9.4: Power consumption of the baseline and the proposed accelerators.

Breakdown of total power consumption	**Baseline**	**Proposed without leakage reuse**	**Proposed with leakage reuse**
One PE including 0.5 KB of weight memory and two 8-bit MACs	16.39 mW	2.26 mW	1.21 mW
168 PEs	2754 mW	380 mW	203.35 mW
Overhead due to leakage reuse including 2688 LCBs, 168 LC wrappers, and 168 PVT mitigation circuit blocks	N/A	N/A	75.71 mW + 0.63 mW + 0.88 mW
Total power consumption	2754 mW	380 mW	280.57 mW

9.3.6 Circuit Techniques for Robust Leakage Reuse Under Process, Voltage, and Temperature Variations

The results described in Sections 9.3.4 and 9.3.5 are obtained for the typical-typical (TT) process corner and a temperature of 27°C. As noted, for the leakage reuse technique proposed in this paper, the supply voltage of the MAC units is generated from the leakage current of the SRAM banks. However, the leakage current is highly sensitive to process and temperature variations, which results in a deviation in the generated supply voltage of the MAC units from the target of 0.45 V.

In this paper, a novel circuit technique is developed to generate a stable supply

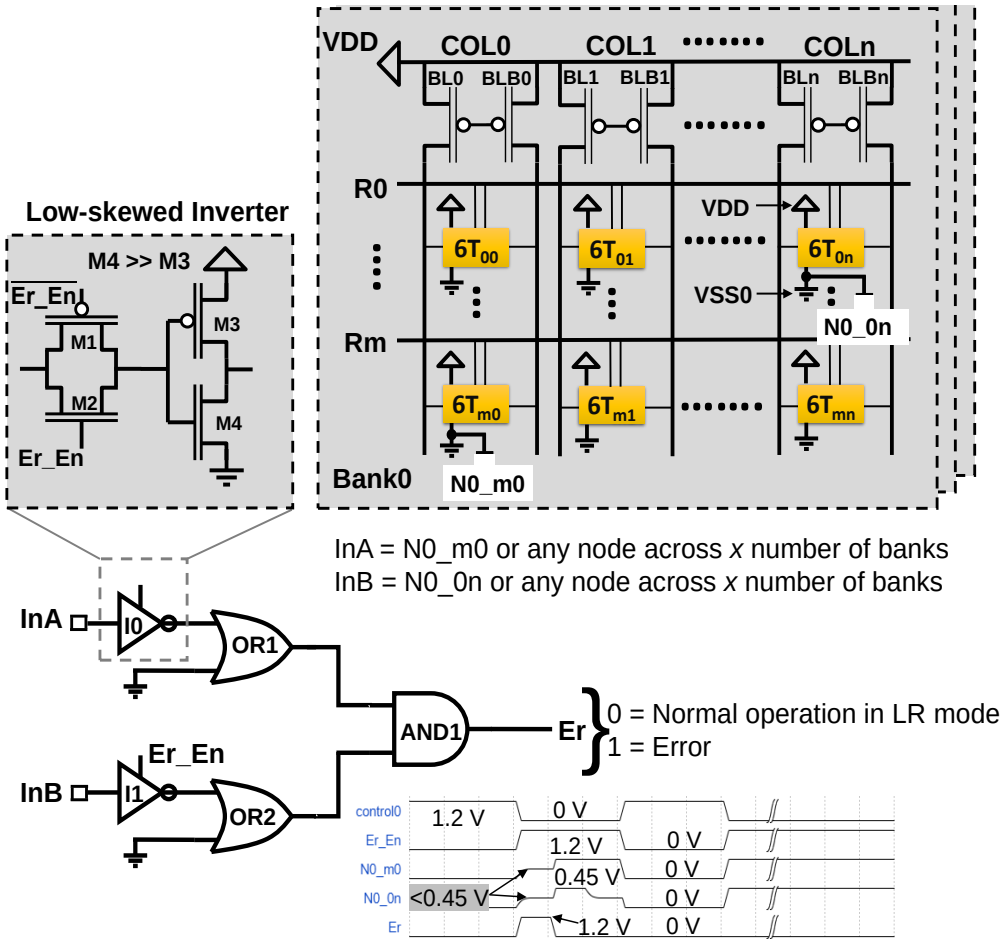


Fig. 9.19: Proposed circuit that maintains a stable supply voltage for the receiver in the presence of process and/or temperature variation.

voltage when applying leakage reuse in the presence of process, voltage, and temperature (PVT) variations. The proposed circuit is shown in Fig. 9.19, which dynamically detects any variation in the generated supply voltage from the leakage current of Bank0 and maintains a stable supply voltage of 0.45 V. The error detection circuit consists of two tri-state low-skewed inverters, two OR gates, and one AND gate. The term “error" refers to any increase or decrease in the supply voltage from the target

value of 0.45 V. The two error detection circuits are placed on the virtual ground node of two memory cells of Bank0 to avoid any invalid activation of the error signal. When the voltage fluctuation on the virtual ground node of both cells *m0* (row *m*, column *0*) and *0n* (row *0*, column *n*) exceeds a set threshold, the error signal is asserted. The threshold for generating the error signal is tuned by transistor widths of the error detection circuit, which is set to 0.43 V in this paper. The leakage control block (LCB), which is shown in Fig. 9.20, is modified to implement the proposed error detection and correction technique. Once the supply voltage deviates from the target value of 0.45 V, the error signal E_r is asserted as logic high. The error signal is connected to the modified leakage control block as shown in Fig. 9.20. The modified LCB consists of n number of PMOS transistors and one NMOS transistor that allow for the flow of current in both the idle and active mode of operation of the donor SRAM, respectively. However, the PMOS transistors are placed in parallel to enable fine-grain control of the generated supply voltage for the receiver. In this paper, three PMOS transistors (P0, P1, and P2) are used to recycle the leakage current of the idle memory units with a 1-bit control signal C0 that is applied to all three transistors. The gate of transistor P0 is also controlled by the E_r signal. Transistor P0 is forced to operate in the cut-off mode when E_r is set to logic high, which increases the supply voltage of the receiver. Note that the tuning of the supply voltage of the receiver to a different voltage than 0.45 V is possible at run-time by applying the multi-bit control

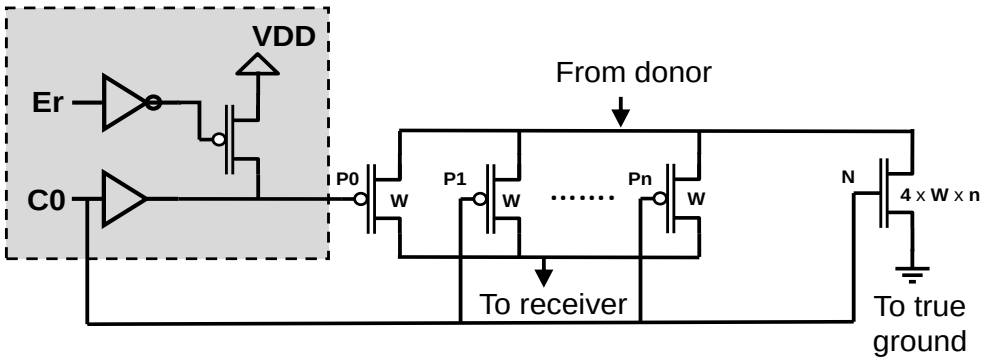


Fig. 9.20: Modified leakage control block that dynamically increases the supply voltage for the receiver.

signal C_n to the n number of PMOS transistors.

The supply voltage of the MAC units, which is generated through leakage reuse, is characterized for the typical-typical (TT), the fast-fast (FF), and the slow-slow (SS) process corners and at temperatures of 27°C, 50°C, and 100°C. The simulation results are shown in Fig. 9.21. The generated supply voltage significantly deviates from the target voltage of 0.45 V across process corners and temperatures. Note that in all cases, the supply voltage is lower than the target voltage due to the sizing of the PMOS and NMOS transistors within the leakage control block. The size of the NMOS transistor is 4× the size of the PMOS transistors to minimize the resistance along the path to ground during the active mode operation of the SRAM. The current through the NMOS transistor is the primary cause of any variation that results from operating the circuit in the SS corner, FF corner, and at higher temperatures (50°C and 100°C). Therefore, tri-state low-skewed inverters are used, which produce a logic high for an

input voltage less than 430 mV and a logic low at the output when the input voltage is greater than 430 mV. Note that a logic high is required on the *Er_En* input signal to enable the inverters. The results from transient simulation of the mitigation circuit across process and temperature corners are shown in Fig. 9.21(b). At all process corners and temperatures, the proposed technique generates a stable supply voltage of 0.45 V for the MAC units.

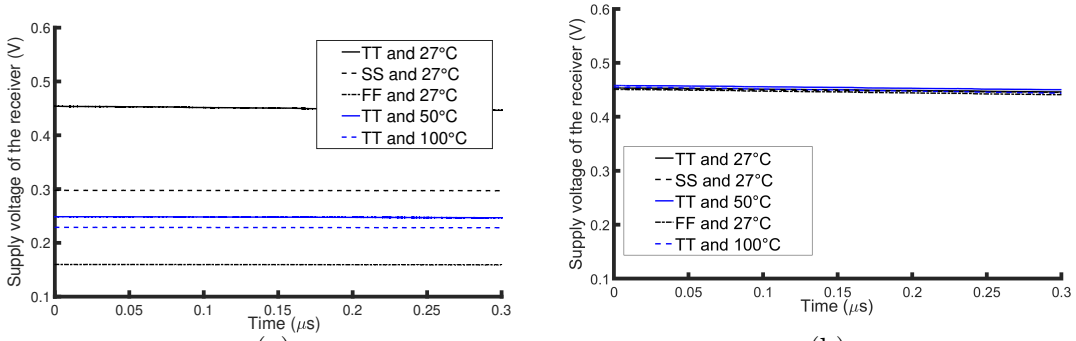


Fig. 9.21: Characterization of the supply voltage of the receiver with TT, FF, and SS process corners and at the temperatures of 27°C, 50°C, and 100°C, where a) includes the results without variation mitigation and b) shows the results with the proposed process and temperature variation mitigation technique.

9.4 Summary

A heterogeneous multi-voltage domain DNN accelerator architecture is proposed that implements near-memory computing within a given PE. The leakage current from memory banks operating at 1.2 V is recycled to generate a supply voltage of 0.45 V for the adjacent MAC units. Therefore, a separate voltage source is not required for the

MAC units operating at 0.45 V. In addition, a novel circuit to mitigate process and temperature variation effects is proposed that maintains a stable supply voltage of 0.45 V for the MAC units. The proposed architecture improves the energy efficiency by 71.4× (3.27 TOPS/W) as compared to a baseline topology that includes both memory and MAC units operating within a single voltage domain of 1.2 V. However, the throughput is reduced by 35×. Applying the leakage reuse technique to only half of all memory cells within each PE reduces the total power consumption by 26% (99.4 mW) as compared to an accelerator that does not apply leakage reuse. Therefore, the proposed architecture and techniques allow for more energy efficient means of inference for edge devices.

Conclusions

The importance of energy efficient computing is increasing with the rapid growth of battery-operated devices, especially with the emergence of on-device intelligence. The market for intelligent and efficient battery-operated devices is expected to grow in the foreseeable future. Ultra-low voltage circuits are one of the most effective ways of achieving energy efficient computing. However, novel circuit and power management methodologies are needed to fully benefit from ultra-low voltage operation. In this research defense, circuit techniques, methodologies, algorithms, architectural techniques, and power management techniques are developed in conjunction with a novel leakage reuse technique for energy efficient, robust, and ultra-low voltage computing. Circuit and algorithmic techniques developed include 1) robust and energy efficient logic families for near-threshold computing, 2) interfacing of ultra-low voltage circuits with circuits operating at higher voltages, 3) clock-power co-design for sub-threshold computing, and 4) optimization of the supply and threshold voltage.

Three novel differential signaling based logic families are developed for near-threshold computing that outperform the equivalent CMOS and current-mode logic (CML) circuits with respect to power, performance, and noise margins. The three logic families, which are described as dynamic current mode logic (DCML), latched

dynamic current mode logic (LDCML), and dynamic feedback current mode logic (DFCML), are developed in a 130 nm technology to operate at 400 mV and 450 mV. The circuits implemented with dynamic differential signaling logic (DDSL) families improve the noise margin, increase the maximum operating speed, and reduce the total power consumption by, respectively, 2.5×, 1.4×, and 0.96× as compared to circuits developed with CMOS logic. Techniques are developed to interface near-threshold circuits with circuits operating at nominal and I/O voltages. A novel single-ended level shifter and a bi-directional input/output circuit are developed that provide a conversion range of 0.38 V to 1.2 V and 0.45 V to 3.3 V depending on the target output voltage. The interface circuit that allows for signal propagation between the near-threshold voltage domain of 0.45 V and I/O ports operating at a voltage of 3.3 V consumes 4.87 mW and produces a fanout of four (FO4) delay of 0.46 in a 130 nm process. The single ended level shifter reduces the delay and power consumption to, respectively, 0.74× and 0.4× that of a current mirror level shifter when converting an input voltage of 0.625 V to an output voltage of 1.5 V. A robust clock-power co-design methodology is developed for sub-threshold operation at 250 mV, where a distributed, multi-voltage domain, and hierarchical power distribution network exhibits increased tolerance to power supply noise. The multi-voltage domain power distribution network delivers current to the clock buffers, the registers, and the combinational circuits of a local clock network while providing noise immunity to up to 10% variation in a target supply voltage of 250 mV. In addition, the use of optimized sub-Vt buffers results in a reduction in the minimum clock period, skew, and insertion delay to, respectively, 0.74×, 0.52×, and 0.79× that of a clock network that includes

non-optimized buffers operating at 250 mV. A linear programming-based optimization technique is developed and applied to optimally determine the supply voltage of circuits operating at ultra-low voltages for a given range of threshold voltages. The variations in the maximum operating frequency, the noise margins, and the threshold voltage of the transistors are considered in the optimization algorithm, where the proposed technique optimally selects a supply voltage with up to 14% and 8% error in respectively, the average noise margins and the maximum operating frequency as compared to target circuit specifications.

An energy efficient multi-voltage domain heterogeneous architecture is developed to accelerate neural network models, which is evaluated in an Eyeriss-like architecture with 168 processing elements and a total of 300 KB of on-chip memory. A design space exploration is performed that indicates that the proposed heterogeneous accelerator is better suited to execute various state-of-the-art DNN models as compared to conventional monolithic accelerators. The proposed accelerator implements multiple PE sub-arrays for the concurrent execution of different neural network models.

A novel power management technique referred to as 'leakage reuse' is developed as part of this research effort. Work completed based on leakage reuse includes 1) circuits and methodologies to reduce power consumption through the reuse of leakage current from idle computing units of integrated circuits, 2) algorithms for the implementation of leakage reuse as both a stand alone technique as well as in conjunction with existing power gating methodologies, and 3) circuits and methods to reduce power consumption by applying leakage reuse to idle on-chip memories of AI accelerators. The leakage reuse technique utilizes leakage current from idle donor circuits and provides current to receiver circuits by utilizing the recycled leakage energy.

Therefore, an additional voltage source is not required for receivers that are operating at an ultra-low voltage. Circuit methodologies are developed for the implementation of the leakage reuse technique on the computing units of ISCAS benchmark circuits and the ARM Cortex M0 core. In addition, an analysis of the energy break-even point is performed. The total power consumption of a power management system that implements the leakage reuse technique is reduced to 0.41x that of a conventional power delivery system. Moreover, novel scheduling algorithms are developed for a) the optimal assignment of idle donor cores for leakage reuse, which is described as 'longest idle time – leakage reuse (LIT-LR),' and b) the optimal and simultaneous execution of leakage reuse and power gating techniques, which is described as 'longest idle time- simultaneous execution of leakage reuse and power gating (LIT-LRPG).' The implementation of the LIT-LR algorithm on a MPSoC with five cores resulted in a reduction of the energy consumption by the leakage control blocks and a reduction in the peak power consumption of, respectively, 25% and 7.4% as compared to a random assignment of cores for leakage reuse. The execution of the LIT-LRPG algorithm on the MPSoC system reduces the total energy consumption by 50.2%, 14.4%, and 5.7% as compared to, respectively, the baseline scenario that includes neither leakage reuse or power gating, only power gating, and only leakage reuse.

The leakage reuse technique is also applied to idle on-chip memories of multi-voltage domain heterogeneous DNN accelerators. Circuit technique and methodologies are developed to deliver current to the multiply-accumulate (MAC) units of processing elements from the recycled leakage current of the adjacent on-chip SRAM, which are utilized for energy efficient near-memory computing. In addition, a novel

circuit is developed to maintain a stable supply voltage for the MAC units implemented with the leakage reuse technique in the presence of variations in process, voltage, and temperature. The proposed accelerator that operates memory banks at 1.2 V and generates a supply voltage of 0.45 V for the MAC units when leakage reuse is applied, improves the energy efficiency by 71.4× (3.27 TOPS/W) as compared to a baseline topology that includes both memory and MAC units operating within a single voltage domain of 1.2 V. In addition, applying the leakage reuse technique to only half of all memory cells within each processing element reduces the total power consumption by 26% (99.4 mW) as compared to an accelerator that does not apply leakage reuse. The techniques and methodologies developed as part of this research effort allow for the implementation of reliable and energy efficient circuit solutions for artificial intelligence computation of ubiquitous edge, IoT, and embedded applications.

Bibliography

[1] M. P. Mills, "The cloud begins with coal-an overview of the electricity used by the global digital ecosystem," *Digital Power Group*, pp. 1–48, August 2013.

[2] F. Faggin, M. E. Hoff, S. Mazor, and M. Shima, "The history of the 4004," *IEEE Micro*, Vol. 16, No. 6, pp. 10–20, December 1996.

[3] S. H. M. of Japan. (2020, November) 12-bit engine-control microprocessor (Toshiba). [Online]. Available: https://www.shmj.or.jp/english/pdf/ic/exhibi739E.pdf

[4] P. Hester, R. O. Simpson, and A. Chang, "The IBM RT PC ROMP and memory management unit architecture," *IBM RT Personal Computer Technology*, pp. 48–56, 1986.

[5] R. K. Montoye, E. Hokenek, and S. L. Runyon, "Design of the IBM RISC system/6000 floating-point execution unit," *IBM Journal of Research and Development*, Vol. 34, No. 1, pp. 59–70, January 1990.

[6] H. El-Sayed, S. Sankar, M. Prasad, D. Puthal, A. Gupta, M. Mohanty, and C.-T. Lin, "Edge of things: The big picture on the integration of edge, IoT and the cloud in a distributed computing environment," *IEEE Access*, Vol. 6, pp. 1706–1717, December 2017.

[7] X. Sun and N. Ansari, "EdgeIoT: Mobile edge computing for the internet of things," *IEEE Communications Magazine*, Vol. 54, No. 12, pp. 22–29, December 2016.

[8] C. Gatenberg. (2020, February) ARM's new edge AI chips promise IoT devices that won't need the cloud.

[Online]. Available: https://www.theverge.com/2020/2/10/21130800/arm-new-edge-ai-chips-processing-npu-cortex-m55-u55-iot

[9] Bitfury. (2020, July) How to choose the best AI processor for the edge. [Online]. Available: https://www.edge-ai-vision.com/2020/07/how-to-choose-the-best-ai-processor-for-the-edge/

[10] Edgify.ai. (2019, August) The Rise of Edge Devices. [Online]. Available: https://edgify.medium.com/the-rise-of-the-edge-devices-953f96908a84

[11] E. Shih, P. Bahl, and M. J. Sinclair, "Wake on wireless: An event driven energy saving strategy for battery operated devices," *Proceedings of the Annual International Conference on Mobile Computing and Networking*, pp. 160–171, September 2002.

[12] T. Austin, D. Blaauw, S. Mahlke, T. Mudge, C. Chakrabarti, and W. Wolf, "Mobile supercomputers," *Computer*, Vol. 37, No. 5, pp. 81–83, May 2004.

[13] M. Halpern, Y. Zhu, and V. J. Reddi, "Mobile CPU's rise to power: Quantifying the impact of generational mobile CPU design trends on performance, energy, and user satisfaction," *Proceedings of the IEEE International Symposium on High Performance Computer Architecture*, pp. 64–76, March 2016.

[14] D. Brunelli, C. Moser, L. Thiele, and L. Benini, "Design of a solar-harvesting circuit for batteryless embedded systems," *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 56, No. 11, pp. 2519–2528, February 2009.

[15] Q. Ju and Y. Zhang, "Predictive power management for internet of battery-less things," *IEEE Transactions on Power Electronics*, Vol. 33, No. 1, pp. 299–312, February 2017.

[16] H. Sutter, "The free lunch is over: A fundamental turn toward concurrency in software," *Dr. Dobb's Journal*, Vol. 30, No. 3, pp. 202–210, March 2005.

[17] S. Paul, *et al.*, "A sub-cm^3 energy-harvesting stacked wireless sensor node featuring a near-threshold voltage IA-32 microcontroller in 14-nm tri-gate CMOS for always-ON always-sensing applications," *IEEE Journal of Solid-State Circuits*, Vol. 52, No. 4, pp. 961–971, March 2017.

[18] R. G. Dreslinski, M. Wieckowski, D. Blaauw, D. Sylvester, and T. Mudge, "Near-threshold computing: Reclaiming Moore's law through energy efficient integrated circuits," *Proceedings of the IEEE*, Vol. 98, No. 2, pp. 253–266, January 2010.

[19] Intel. (2020 (Accessed), December) Intel Comparison Table of Haswell Celeron, Pentium, i3, i5, and i7 models. [Online]. Available: https://ark.intel.com/content/www/us/en/ark/compare.html?productIds=/77773,77775,77777,77480,77769,77771,75036,75037,75043,76640,75044,75045,75047,75048,76641,75049,75050,75121,75122,75123,76642,75124,75125

[20] B. Zhai, D. Blaauw, D. Sylvester, and K. Flautner, "Theoretical and practical limits of dynamic voltage scaling," *Proceedings of the 41st annual Design Automation Conference*, pp. 868–873, July 2004.

[21] A. Zou, J. Leng, X. He, Y. Zu, C. D. Gill, V. J. Reddi, and X. Zhang, "Voltage-stacked power delivery systems: Reliability, efficiency, and power management," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 39, No. 12, pp. 5142–5155, January 2020.

[22] K. Craig, Y. Shakhsheer, S. Arrabi, S. Khanna, J. Lach, and B. H. Calhoun, "A 32b 90 nm processor implementing panoptic DVS achieving energy efficient operation from sub-threshold to high performance," *IEEE Journal of Solid-State Circuits*, Vol. 49, No. 2, pp. 545–552, February 2014.

[23] K. Usami and N. Ohkubo, "A design approach for fine-grained run-time power gating using locally extracted sleep signals," *Proceedings of the IEEE International Conference on Computer Design*, pp. 155–161, October 2007.

[24] V. De, "Fine-grain power management in manycore processor and System-on-Chip (SoC) designs," *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 159–164, November 2015.

[25] P. Meinerzhagen, *et al.*, "An energy-efficient graphics processor featuring fine-grain DVFS with integrated voltage regulators, execution-unit turbo, and retentive sleep in 14nm tri-gate CMOS," *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 38–40, February 2018.

[26] M. Arora, S. Manne, I. Paul, N. Jayasena, and D. M. Tullsen, "Understanding idle behavior and power gating mechanisms in the context of modern benchmarks on CPU-GPU integrated systems," *Proceedings of the IEEE International Symposium on High Performance Computer Architecture*, pp. 366–377, March 2015.

[27] Z. Hu, A. Buyuktosunoglu, V. Srinivasan, V. Zyuban, H. Jacobson, and P. Bose, "Microarchitectural techniques for power gating of execution units," *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 32–37, August 2004.

[28] H. Singh, K. Agarwal, D. Sylvester, and K. J. Nowka, "Enhanced leakage reduction techniques using intermediate strength power gating," *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 15, No. 11, pp. 1215–1224, November 2007.

[29] L. Bolzani, A. Calimera, A. Macii, E. Macii, and M. Poncino, "Enabling concurrent clock and power gating in an industrial design flow," *Proceedings of the Conference on Design, Automation and Test in Europe*, pp. 334–339, April 2009.

[30] X. Zhang, T. Tong, S. Kanev, S. K. Lee, G. Wei, and D. Brooks, "Characterizing and evaluating voltage noise in multi-core near-threshold processors," *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 82–87, September 2013.

[31] S. K. Samal, G. Chen, and S. K. Lim, "Improving performance under process and voltage variations in near-threshold computing using 3D ICs," *ACM Journal on Emerging Technologies in Computing Systems*, Vol. 13, No. 4, p. 59, August 2017.

[32] J. Kwong and A. P. Chandrakasan, "Variation-driven device sizing for minimum energy sub-threshold circuits," *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 8–13, October 2006.

[33] M. Li, C. Ieong, M. Law, P. Mak, M. Vai, and R. P. Martins, "Sub-threshold

standard cell library design for ultra-low power biomedical applications," *Proceedings of the IEEE International Conference on Engineering in Medicine and Biology Society*, pp. 1454–1457, July 2013.

[34] J. Myers, A. Savanth, P. Prabhat, S. Yang, R. Gaddh, S. O. Toh, and D. Flynn, "A 12.4 pJ/cycle sub-threshold, 16pJ/cycle near-threshold ARM Cortex-M0+ MCU with autonomous SRPG/DVFS and temperature tracking clocks," *Proceedings of the IEEE Symposium on VLSI Circuits*, pp. 332–333, June 2017.

[35] K. Craig, Y. Shakhsheer, and B. H. Calhoun, "Optimal power switch design for dynamic voltage scaling from high performance to subthreshold operation," *Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design*, pp. 221–224, July 2012.

[36] S. Cass. (2018, May) Chip Hall of Fame: Intel 4004 Microprocessor. [Online]. Available: https://spectrum.ieee.org/tech-history/silicon-revolution/chip-hall-of-fame-intel-4004-microprocessor

[37] Apple. (2020, November) Apple unleashes M1. [Online]. Available: https://www.apple.com/newsroom/2020/11/apple-unleashes-m1/

[38] Wikipedia. (2020, November) Microprocessor chronology. [Online]. Available: https://en.wikipedia.org/wiki/Microprocessor_chronology

[39] Wikipedia. (2020, November) Transistor count. [Online]. Available: https://en.wikipedia.org/wiki/Transistor_count

[40] Intel. (2018, May) Intel at 50: The 8080 Microprocessor [Online]. [Online]. Available: https://newsroom.intel.com/articles/intel-50-8080-microprocessor/

[41] R. Dreslinski, M. Wieckowski, D. S. Blaauw, and T. Mudge, "Near threshold computing: Overcoming performance degradation from aggressive voltage scaling," *Proc. Workshop Energy-Efficient Design*, pp. 44–49, June 2009.

[42] C.-M. Kyung and S. Yoo, *Energy-aware system design: Algorithms and architectures*. Springer, 2011.

[43] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys & Tutorials*, Vol. 19, No. 4, pp. 2322–2358, August 2017.

[44] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE Journal on Selected Areas in Communications*, Vol. 34, No. 12, pp. 3590–3605, September 2016.

[45] G. view research. (2020, March) Edge Computing Market Size, Share and Trends Analysis Report. [Online]. Available: https://www.grandviewresearch.com/industry-analysis/edge-computing-market

[46] Q. Xie, X. Lin, Y. Wang, S. Chen, M. J. Dousti, and M. Pedram, "Performance comparisons between 7-nm FinFET and conventional bulk CMOS standard cell libraries," *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol. 62, No. 8, pp. 761–765, August 2015.

[47] E. Cai and D. Marculescu, "Temperature effect inversion-aware power-performance optimization for FinFET-based multicore systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 36, No. 11, pp. 1897–1910, February 2017.

[48] K. Han, J. Lee, J. Lee, W. Lee, and M. Pedram, "TEI-NoC: Optimizing ultralow power NoCs exploiting the temperature effect inversion," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 37, No. 2, pp. 458–471, February 2018.

[49] W. Lee, Y. Wang, T. Cui, S. Nazarian, and M. Pedram, "Dynamic thermal management for FinFET-based circuits exploiting the temperature effect inversion phenomenon," *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 105–110, August 2014.

[50] S. Jain, *et al.*, "A 280mV-to-1.2 V wide-operating-range IA-32 processor in 32nm CMOS," *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 66–68, February 2012.

[51] B. Doyle, R. Arghavani, D. Barlage, S. Datta, M. Doczy, J. Kavalieros, A. Murthy, and R. Chau, "Transistor elements for 30nm physical gate lengths and beyond," *Intel Technology Journal*, Vol. 6, No. 2, May 2002.

[52] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi, "The McPAT framework for multicore and manycore architectures: Simultaneously modeling power, area, and timing," *ACM Transactions on Architecture and Code Optimization*, Vol. 10, No. 1, pp. 1–29, April 2013.

[53] K. Siu, D. M. Stuart, M. Mahmoud, and A. Moshovos, "Memory requirements for convolutional neural network hardware accelerators," *Proceedings of the IEEE International Symposium on Workload Characterization*, pp. 111–121, October 2018.

[54] M. Qazi, M. Sinangil, and A. Chandrakasan, "Challenges and directions for low-voltage SRAM," *IEEE Design and Test of Computers*, Vol. 28, No. 1, pp. 32–43, October 2010.

[55] A. C. Cabe, Z. Qi, and M. R. Stan, "Stacking SRAM banks for ultra low power standby mode operation," *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 699–704, June 2010.

[56] H. Qin, Y. Cao, D. Markovic, A. Vladimirescu, and J. Rabaey, "SRAM leakage suppression by minimizing standby supply voltage," *Proceedings of the IEEE International Symposium on Signals, Circuits and Systems*, pp. 55–60, March 2004.

[57] G. Pasandi, R. Mehta, M. Pedram, and S. Nazarian, "Hybrid cell assignment and sizing for power, area, delay-product optimization of SRAM arrays," *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol. 66, No. 12, pp. 2047–2051, December 2019.

[58] B. Amelifard, F. Fallah, and M. Pedram, "Leakage minimization of SRAM cells in a dual-V_t and dual-T_{ox} technology," *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 16, No. 7, pp. 851–860, June 2008.

[59] N. P. Jouppi, *et al.*, "In-datacenter performance analysis of a tensor processing unit," *Proceedings of the ACM/IEEE Annual International Symposium on Computer Architecture*, pp. 1–12, May 2017.

[60] Y. Chen, T. Yang, J. Emer, and V. Sze, "Eyeriss v2: A flexible accelerator for emerging deep neural networks on mobile devices," *arXiv preprint arXiv:1807.07928*, 2018.

[61] T. Chen, Z. Du, N. Sun, J. Wang, C. Wu, Y. Chen, and O. Temam, "Diannao: A small-footprint high-throughput accelerator for ubiquitous machine-learning," *Proceedings of the ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, Vol. 49, No. 4, pp. 269–284, March 2014.

[62] D. Fick and M. Henry. (2018, August) Analog computation in flash memory for datacenter-scale AI inference in a small chip. [Online]. Available: https://www.hotchips.org

[63] Y.-H. Chen, T.-J. Yang, J. Emer, and V. Sze, "Eyeriss v2: A flexible accelerator for emerging deep neural networks on mobile devices," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, Vol. 9, No. 2, pp. 292–308, June 2019.

[64] H. Jiang, M. Marek-Sadowska, and S. R. Nassif, "Benefits and costs of power-gating technique," *Proceedings of the IEEE International Conference on Computer Design*, pp. 559–566, October 2005.

[65] S. Kim, S. V. Kosonocky, D. R. Knebel, and K. Stawiasz, "Experimental measurement of a novel power gating structure with intermediate power saving mode," *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 20–25, August 2004.

[66] S. Kim, S. V. Kosonocky, and D. R. Knebel, "Understanding and minimizing ground bounce during mode transition of power gating structures," *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 22–25, August 2003.

[67] R. Zhang, K. Mazumdar, B. H. Meyer, K. Wang, K. Skadron, and M. R. Stan, "Transient voltage noise in charge-recycled power delivery networks for many-layer 3D-IC," *Proceedings of the IEEE/ACM International Symposium on Low Power Electronics and Design*, pp. 152–158, September 2015.

[68] K. Blutman, A. Kapoor, J. G. Martinez, H. Fatemi, and J. P. Gyvez, "Lower power by voltage stacking: A fine-grained system design approach," *Proceedings of the ACM/IEEE Annual Design Automation Conference*, pp. 78:1–78:5, June 2016.

[69] N. H. Weste and D. Harris, *CMOS VLSI design: a circuits and systems perspective*. Pearson Education India, 2015.

[70] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-power CMOS digital design," *IEEE Journal of Solid-State Circuits*, Vol. 27, No. 4, pp. 473–484, April 1992.

[71] R. G. Dreslinkski, B. Zhai, T. Mudge, D. Blaauw, and D. Sylvester, "An energy efficient parallel architecture using near threshold operation," *Proceedings of the IEEE International Conference on Parallel Architecture and Compilation Techniques*, pp. 175–188, September 2007.

[72] B. Zhai, R. G. Dreslinski, D. Blaauw, T. Mudge, and D. Sylvester, "Energy efficient near-threshold chip multi-processing," *Proceedings of the IEEE/ACM International Symposium on Low Power Electronics and Design*, pp. 32–37, August 2007.

[73] P. Bose, *et al.*, "Power management of multi-core chips: Challenges and pitfalls," *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition*, pp. 977–982, March 2012.

[74] Y. Liu, Y. Jin, and P. Li, "Exploring sparsity of firing activities and clock gating for energy-efficient recurrent spiking neural processors," *Proceedings of the IEEE/ACM International Symposium on Low Power Electronics and Design*, pp. 1–6, July 2017.

[75] G. Yang and T. Kim, "Design and algorithm for clock gating and flip-flop co-optimization," *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 1–6, November 2018.

[76] Y. K. Ramadass, A. Fayed, A. P. Chandrakasan, *et al.*, "A fully-integrated switched-capacitor step-down DC-DC converter with digital capacitance modulation in 45 nm CMOS," *IEEE Journal of Solid-State Circuits*, Vol. 45, No. 12, pp. 2557–2565, December 2010.

[77] S. Bang, A. Wang, B. Giridhar, D. Blaauw, and D. Sylvester, "A fully integrated successive-approximation switched-capacitor DC-DC converter with 31mV output voltage resolution," *Proceedings of 2013 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, pp. 370–371, February 2013.

[78] H. Meyvaert, T. V. Breussegem, and M. Steyaert, "A 1.65 W fully integrated 90nm Bulk CMOS intrinsic charge recycling capacitive DC-DC converter: Design & techniques for high power density," *Proceedings of IEEE Energy Conversion Congress and Exposition*, pp. 3234–3241, September 2011.

[79] T. M. Andersen, *et al.*, "A sub-ns response on-chip switched-capacitor DC-DC voltage regulator delivering 3.7 W/mm^2 at 90% efficiency using deep-trench capacitors in 32nm SOI CMOS," *Proceedings of the IEEE International Solid-State Circuits Conference Digest of Technical Papers*, pp. 90–91, February 2014.

[80] H. Le, S. R. Sanders, and E. Alon, "Design techniques for fully integrated switched-capacitor DC-DC converters," *IEEE Journal of Solid-State Circuits*, Vol. 46, No. 9, pp. 2120–2131, August 2011.

[81] R. Jain, B. M. Geuskens, S. T. Kim, M. M. Khellah, J. Kulkarni, J. W. Tschanz, and V. De, "A 0.45–1 V fully-integrated distributed switched capacitor DC-DC converter with high density MIM capacitor in 22 nm tri-gate CMOS," *IEEE Journal of Solid-State Circuits*, Vol. 49, No. 4, pp. 917–927, February 2014.

[82] S. R. Sanders, E. Alon, H. Le, M. D. Seeman, M. John, and V. W. Ng, "The road to fully integrated DC–DC conversion via the switched-capacitor approach," *IEEE Transactions on Power Electronics*, Vol. 28, No. 9, pp. 4146–4155, February 2013.

[83] M. D. Seeman and S. R. Sanders, "Analysis and optimization of switched-capacitor DC–DC converters," *IEEE Transactions on Power Electronics*, Vol. 23, No. 2, pp. 841–851, March 2008.

[84] E. Salman and E. G. Friedman, *High performance integrated circuit design.* McGraw Hill Professional, 2012.

[85] A. Shayan, X. Hu, H. Peng, M. Popovich, W. Zhang, C.-K. Cheng, L. Chua-Eoan, and X. Chen, "3D power distribution network co-design for nanoscale stacked silicon ICs," *Proceedings of the IEEE International Conference on Electrical Performance of Electronic Packaging*, pp. 11–14, October 2008.

[86] S. Kose, S. Tam, S. Pinzon, B. McDermott, and E. G. Friedman, "Active filter-based hybrid on-chip DC–DC converter for point-of-load voltage regulation," *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 21, No. 4, pp. 680–691, April 2013.

[87] P. Zarkesh-Ha and J. D. Meindl, "Optimum on-chip power distribution networks for gigascale integration (GSI)," *Proceedings of the IEEE International Interconnect Technology Conference*, pp. 125–127, June 2001.

[88] C. L. Fai and P. Mok, "A monolithic current-mode CMOS DC-DC converter with on-chip current-sensing technique," *IEEE Journal of Solid State Circuits*, Vol. 39, pp. 3–14, January 2004.

[89] T. Kadoyama, N. Suzuki, N. Sasho, H. Iizuka, I. Nagase, H. Usukubo, and M. Katakura, "A complete single-chip GPS receiver with 1.6-V 24-mW radio in 0.18-μm CMOS," *IEEE Journal of Solid-State Circuits*, Vol. 39, No. 4, pp. 562–568, April 2004.

[90] E. Salman, E. G. Friedman, R. M. Secareanu, and O. L. Hartin, "Identification of dominant noise source and parameter sensitivity for substrate coupling," *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 17, No. 10, pp. 1559–1564, October 2009.

[91] A. V. Mezhiba and E. G. Friedman, "Scaling trends of on-chip power distribution noise," *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 12, No. 4, pp. 386–394, April 2004.

[92] A. V. Mezhiba and E. G. Friedman, "Frequency characteristics of high speed power distribution grids," *Analog Integrated Circuits and Signal Processing*, Vol. 35, No. 2-3, pp. 207–214, June 2003.

[93] C. Guiducci, A. Schmid, F. K. Gürkaynak, and Y. Leblebici, "Novel front-end circuit architectures for integrated bio-electronic interfaces," *Proceedings of the Conference on Design, automation and test in Europe*, pp. 1328–1333, March 2008.

[94] M. Popovich, E. G. Friedman, M. Sotman, and A. Kolodny, "On-chip power distribution grids with multiple supply voltages for high-performance integrated circuits," *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 16, No. 7, pp. 908–921, July 2008.

[95] M. Abdelfattah, B. Dupaix, S. Naqvi, and W. Khalil, "A fully-integrated switched capacitor voltage regulator for near-threshold applications," *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 201–204, May 2015.

[96] Y. K. Ramadass, A. A. Fayed, and A. P. Chandrakasan, "A fully-integrated switched-capacitor step-down DC-DC converter with digital capacitance modulation in 45 nm CMOS," *IEEE Journal of Solid-State Circuits*, Vol. 45, No. 12, pp. 2557–2565, December 2010.

[97] V. Mohan, A. Iyer, and J. Sartori, "Enhancing workload-dependent voltage scaling for energy-efficient ultra-low-power embedded systems," *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 1–6, June 2018.

[98] N. Mehta and B. Amrutur, "Dynamic supply and threshold voltage scaling for CMOS digital circuits using in-situ power monitor," *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 20, No. 5, pp. 892–901, May 2012.

[99] S. Kiamehr, M. Ebrahimi, M. S. Golanbari, and M. B. Tahoori, "Temperature-aware dynamic voltage scaling to improve energy efficiency of near-threshold computing," *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 25, No. 7, pp. 2017–2026, July 2017.

[100] M. Konijnenburg, Y. Cho, M. Ashouei, T. Gemmeke, C. Kim, J. Hulzink, J. Stuyt, M. Jung, J. Huisken, and S. Ryu, "Reliable and energy efficient 1 MHz 0.4 V dynamically reconfigurable SoC for ExG applications in 40 nm LP CMOS," *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 430–431, February 2013.

[101] S. Sinha, *Neuromorphic Controller for Low Power Systems From Devices to Circuits.* Arizona State University, 2011.

[102] B. Keller, M. Cochet, B. Zimmer, J. Kwak, A. Puggelli, Y. Lee, M. Blagojević, S. Bailey, P. Chiu, and P. Dabbelt, "A RISC-V processor SoC with integrated power management at submicrosecond timescales in 28 nm FD-SOI," *IEEE Journal of Solid-State Circuits*, Vol. 52, No. 7, pp. 1863–1875, July 2017.

[103] M. Igarashi, T. Uemura, R. Mori, H. Kishibe, M. Nagayama, M. Taniguchi, K. Wakahara, T. Saito, M. Fujigaya, and K. Fukuoka, "A 28 nm High-K/MG heterogeneous multi-core mobile application processor with 2 GHz cores and low-power 1 GHz cores," *IEEE Journal of Solid-State Circuits*, Vol. 50, No. 1, pp. 92–101, January 2015.

[104] Z. Wang, Y. Liu, Y. Sun, Y. Li, D. Zhang, and H. Yang, "An energy-efficient heterogeneous dual-core processor for internet of things," *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 2301–2304, July 2015.

[105] D. Ma and R. Bondade, "Enabling power-efficient DVFS operations on silicon," *IEEE Circuits and Systems Magazine*, Vol. 10, No. 1, pp. 14–30, March 2010.

[106] B. C. Paul, A. Raychowdhury, and K. Roy, "Device optimization for ultra-low power digital sub-threshold operation," *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 96–101, August 2004.

[107] N. Reynders and W. Dehaene, "A 190mV supply, 10MHz, 90nm CMOS, pipelined sub-threshold adder using variation-resilient circuit techniques," *Proceedings of the IEEE Asian Solid-State Circuits Conference*, pp. 113–116, November 2011.

[108] C.-Y. Hsieh, M.-H. Chang, and W. Hwang, "Programmable clock generator used in dynamic-voltage-and-frequency-scaling (DVFS) operated in sub-and near-threshold region," August 2012, US Patent 8,237,477.

[109] P.-C. Wu, Y.-P. Kuo, C.-S. Wu, C.-T. Chuang, Y.-H. Chu, and W. Hwang, "PVT-aware digital controlled voltage regulator design for ultra-low-power (ULP) DVFS systems," *Proceedings of the IEEE International System-on-Chip Conference*, pp. 136–139, September 2014.

[110] R. Ye and Q. Xu, "Learning-based power management for multicore processors via idle period manipulation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 33, No. 7, pp. 1043–1055, July 2014.

[111] Z. Wang, X. Wang, J. Xu, H. Li, R. K. Maeda, Z. Wang, P. Yang, L. H. Duong, and Z. Wang, "An adaptive process-variation-aware technique for power-gating-induced power/ground noise mitigation in MPSoC," *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 24, No. 12, pp. 3373–3386, December 2016.

[112] K. Agarwal, K. Nowka, H. Deogun, and D. Sylvester, "Power gating with multiple sleep modes," *Proceedings of the IEEE/ACM International Symposium on Quality Electronic Design*, pp. 633–637, March 2006.

[113] H. Cherupalli, H. Duwe, W. Ye, R. Kumar, and J. Sartori, "Enabling effective module-oblivious power gating for embedded processors," *Proceedings of the IEEE International Symposium on High Performance Computer Architecture*, pp. 157–168, February 2017.

[114] N. Nasirian, R. Soosahabi, and M. A. Bayoumi, "Probabilistic analysis of power-gating in network-on-chip routers," *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol. 66, No. 2, pp. 242–246, February 2018.

[115] D. Zoni, L. Colombo, and W. Fornaciari, "Darkcache: Energy-performance optimization of tiled multi-cores by adaptively power-gating LLC banks," *ACM Transactions on Architecture and Code Optimization*, Vol. 15, No. 2, p. 21, June 2018.

[116] E. Pakbaznia, F. Fallah, and M. Pedram, "Charge recycling in power-gated CMOS circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 27, No. 10, pp. 1798–1811, October 2008.

[117] S. Mutoh, T. Douseki, Y. Matsuya, T. Aoki, S. Shigematsu, and J. Yamada, "1-V power supply high-speed digital circuit technology with multithreshold-voltage CMOS," *IEEE Journal of Solid-state circuits*, Vol. 30, No. 8, pp. 847–854, August 1995.

[118] S. Roy, N. Ranganathan, and S. Katkoori, "A framework for power-gating functional units in embedded microprocessors," *IEEE transactions on Very Large Scale Integration Systems*, Vol. 17, No. 11, pp. 1640–1649, March 2009.

[119] J. Leverich, M. Monchiero, V. Talwar, P. Ranganathan, and C. Kozyrakis, "Power management of datacenter workloads using per-core power gating," *IEEE Computer Architecture Letters*, Vol. 8, No. 2, pp. 48–51, August 2009.

[120] Q. Diao and J. Song, "Prediction of CPU idle-busy activity pattern," *Proceedings of the IEEE International Symposium on High Performance Computer Architecture*, pp. 27–36, October 2008.

[121] A. B. Kahng, S. Kang, T. S. Rosing, and R. Strong, "Many-core token-based adaptive power gating," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 32, No. 8, pp. 1288–1292, July 2013.

[122] S. Wimer and I. Koren, "Design flow for flip-flop grouping in data-driven clock gating," *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 22, No. 4, pp. 771–778, May 2013.

[123] L. Benini, A. Bogliolo, and G. D. Micheli, "A survey of design techniques for system-level dynamic power management," *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 8, No. 3, pp. 299–316, June 2000.

[124] B. Pandey, J. Yadav, M. Pattanaik, and N. Rajoria, "Clock gating based energy efficient ALU design and implementation on FPGA," *Proceedings of the International Conference on Energy Efficient Technologies for Sustainability*, pp. 93–97, June 2013.

[125] A. Silva, M. Liu, and M. Moghaddam, "Power-management techniques for wireless sensor networks and similar low-power communication devices based on nonrechargeable batteries," *Journal of Computer Networks and Communications*, Vol. 2012, pp. 1–10, August 2012.

[126] K.-W. Chiu, Y.-G. Chen, and C. Lin, "An efficient NBTI-aware wake-up strategy for power-gated designs," *2018 Design, Automation & Test in Europe Conference & Exhibition*, pp. 901–904, March 2018.

[127] K. Toczé, J. Lindqvist, and S. Nadjm-Tehrani, "Characterization and modeling of an edge computing mixed reality workload," *Journal of Cloud Computing*, Vol. 9, No. 1, pp. 1–24, December 2020.

[128] X. Wang, Y. Han, V. C. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, Vol. 22, No. 2, pp. 869–904, January 2020.

[129] Z. Zhao, K. M. Barijough, and A. Gerstlauer, "DeepThings: Distributed adaptive deep learning inference on resource-constrained IoT edge clusters," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 37, No. 11, pp. 2348–2359, October 2018.

[130] R. Hadidi, J. Cao, Y. Xie, B. Asgari, T. Krishna, and H. Kim, "Characterizing the deployment of deep neural networks on commercial edge devices," *Proceedings of the IEEE International Symposium on Workload Characterization*, pp. 35–48, November 2019.

[131] Y. Bai, Y. Song, M. N. Bojnordi, A. Shapiro, E. Ipek, and E. G. Friedman, "Architecting a MOS current mode logic (MCML) processor for fast, low noise and energy-efficient computing in the near-threshold regime," *Proceedings of the IEEE International Conference on Computer Design*, pp. 527–534, December 2015.

[132] R. Taco, I. Levi, M. Lanuzza, and A. Fish, "An 88-fJ/40-MHz [0.4 V]–0.61-pJ/1-GHz [0.9 V] Dual-Mode Logic 8x8 Bit Multiplier Accumulator With a

Self-adjustment Mechanism in 28-nm FD-SOI," *IEEE Journal of Solid-State Circuits*, Vol. 54, No. 2, pp. 560–568, 2019.

[133] L. G. H. W. R. G. J. W. Davis and N. G. Thoma, "Cascode voltage switch logic: A differential CMOS logic family," *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 16–17, February 1984.

[134] M. W. Allam and M. Elmasry, "Dynamic current mode logic (DyCML): A new low-power high-performance logic style," *IEEE Journal of Solid-State Circuits*, Vol. 36, No. 3, pp. 550–558, October 2001.

[135] A. Shapiro and E. G. Friedman, "MOS current mode logic near threshold circuits," *Journal of Low Power Electronics and Applications*, Vol. 4, No. 2, pp. 138–152, June 2014.

[136] D. Marković, C. C. Wang, L. P. Alarcon, T. Liu, and J. M. Rabaey, "Ultralow-power design in near-threshold region," *Proceedings of the IEEE*, Vol. 98, No. 2, pp. 237–252, February 2010.

[137] L. Heller, W. Griffin, J. Davis, and N. Thoma, "Cascode voltage switch logic: A differential CMOS logic family," *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 16–17, February 1984.

[138] P. Ng, P. T. Balsara, and D. Steiss, "Performance of CMOS differential circuits," *IEEE Journal of Solid-State Circuits*, Vol. 31, No. 6, pp. 841–846, June 1996.

[139] J. M. Musicer and J. Rabaey, "MOS current mode logic for low Power, low noise CORDIC computation in mixed-signal environments," *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 102–107, July 2000.

[140] M. Yamashina and H. Yamada, "An MOS current mode logic (MCML) circuit for low-power sub-GHz processors," *Transactions on Electronics*, Vol. 75, No. 10, pp. 1181–1187, October 1992.

[141] Y. Yang, H. Jeong, S. C. Song, J. Wang, G. Yeap, and S. Jung, "Single bit-line 7T SRAM cell for near-threshold voltage operation with enhanced performance

and energy in 14 nm FinFET technology," *IEEE Transactions on Circuits and Systems*, Vol. 63, No. 7, pp. 1023–1032, July 2016.

[142] A. Gebregiorgis, S. Kiamehr, F. Oboril, R. Bishnoi, and M. B. Tahoori, "A cross-layer analysis of soft error, aging and process variation in near threshold computing," *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*, pp. 205–210, March 2016.

[143] S. Keller, D. M. Harris, and A. J. Martin, "A compact transregional model for digital CMOS circuits operating near threshold," *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 22, No. 10, pp. 2041–2053, October 2014.

[144] H. Kaul, M. Anders, S. Hsu, A. Agarwal, R. Krishnamurthy, and S. Borkar, "Near-threshold voltage (NTV) design: Opportunities and challenges," *Proceedings of the IEEE/ACM Design Automation Conference*, pp. 1153–1158, June 2012.

[145] X. Huang, *et al.*, "Sub-50 nm P-channel FinFET," *IEEE Transactions on Electron Devices*, Vol. 48, No. 5, pp. 880–886, May 2001.

[146] A. W. Topol, *et al.*, "Three-dimensional integrated circuits," *IBM Journal of Research and Development*, Vol. 50, No. 4.5, pp. 491–506, July 2006.

[147] U. R. Karpuzcu, N. S. Kim, and J. Torrellas, "Coping with parametric variation at near-threshold voltages," *IEEE Micro*, Vol. 33, No. 4, pp. 6–14, June 2013.

[148] S. Hanson, *et al.*, "Performance and variability optimization strategies in a sub-200mV, 3.5 pJ/inst, 11nW subthreshold processor," *Proceedings of the IEEE Symposium on VLSI Circuits*, pp. 152–153, June 2007.

[149] M. Seok, S. Hanson, Y. Lin, Z. Foo, D. Kim, Y. Lee, N. Liu, D. Sylvester, and D. Blaauw, "The Phoenix processor: A 30pW platform for sensor applications," *Proceedings of the IEEE Symposium on VLSI Circuits*, pp. 188–189, June 2008.

[150] B. H. Calhoun and A. Chandrakasan, "Characterizing and modeling minimum energy operation for subthreshold circuits," *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 90–95, August 2004.

[151] N. S. Kim, T. Austin, D. Blaauw, T. Mudge, K. Flautner, J. S. Hu, M. J. Irwin, M. Kandemir, and V. Narayanan, "Leakage current: Moore's law meets static power," *Computer*, Vol. 36, No. 12, pp. 68–75, December 2003.

[152] J. Kao, S. Narendra, and A. Chandrakasan, "Subthreshold leakage modeling and reduction techniques," *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 141–148, November 2002.

[153] R. M. Swanson and J. D. Meindl, "Ion-implanted complementary MOS transistors in low-voltage circuits," *IEEE Journal of Solid-State Circuits*, Vol. 7, No. 2, pp. 146–153, April 1972.

[154] C. Wilkerson, H. Gao, A. R. Alameldeen, Z. Chishti, M. Khellah, and S. Lu, "Trading off cache capacity for reliability to enable low voltage operation," *Proceedings of the IEEE/ACM International Symposium on Computer Architecture*, pp. 203–214, July 2008.

[155] X. Wu, F. Wang, and Y. Xie, "Analysis of subthreshold FinFET circuits for ultra-low power design," *Proceedings of the IEEE International SOC Conference*, pp. 91–92, September 2006.

[156] C. Auth, *et al.*, "A 22nm high performance and low-power CMOS technology featuring fully-depleted tri-gate transistors, self-aligned contacts and high density MIM capacitors," *Proceedings of the IEEE Symposium on VLSI Technology*, pp. 131–132, June 2012.

[157] U. R. Karpuzcu, I. Akturk, and N. S. Kim, "Accordion: Toward soft near-threshold voltage computing," *Proceedings of the IEEE International Symposium on High Performance Computer Architecture*, pp. 72–83, February 2014.

[158] V. M. V. Santen, J. Martin-Martinez, H. Amrouch, M. M. Nafria, and J. Henkel, "Reliability in super-and near-threshold computing: A unified model of rtn, bti, and pv," *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 65, No. 1, pp. 293–306, January 2018.

[159] H. Soeleman, K. Roy, and B. C. Paul, "Robust subthreshold logic for ultra-low power operation," *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 9, No. 1, pp. 90–99, February 2001.

[160] B. H. Calhoun, A. Wang, and A. Chandrakasan, "Modeling and sizing for minimum energy operation in subthreshold circuits," *IEEE Journal of Solid-State Circuits*, Vol. 40, No. 9, pp. 1778–1786, August 2005.

[161] A. Kaizerman, S. Fisher, and A. Fish, "Subthreshold dual mode logic," *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 21, No. 5, pp. 979–983, June 2012.

[162] M. Zangeneh and A. Joshi, "Designing tunable subthreshold logic circuits using adaptive feedback equalization," *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 24, No. 3, pp. 884–896, April 2015.

[163] A. Tajalli, E. J. Brauer, Y. Leblebici, and E. Vittoz, "Subthreshold source-coupled logic circuits for ultra-low-power applications," *IEEE Journal of Solid-State Circuits*, Vol. 43, No. 7, pp. 1699–1710, June 2008.

[164] M. S. Hossain and I. Savidis, "Robust near-threshold inverter with improved performance for ultra-low power applications," *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 738–741, May 2016.

[165] M. S. Hossain and I. Savidis, "Dynamic current mode inverter for ultra-low power near-threshold computing," *Proceedings of the Government Microcircuit Applications and Critical Technology Conference*, pp. 409–412, March 2016.

[166] F. Crupi, M. Alioto, J. Franco, P. Magnone, M. Togo, N. Horiguchi, and G. Groeseneken, "Understanding the basic advantages of bulk FinFETs for sub- and near-threshold logic circuits from device measurements," *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol. 59, No. 7, pp. 439–442, June 2012.

[167] V. Gutnik and A. P. Chandrakasan, "Embedded power supply for low-power DSP," *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 5, No. 4, pp. 425–435, December 1997.

[168] B. H. Calhoun and A. P. Chandrakasan, "Ultra-dynamic voltage scaling (UDVS) using sub-threshold operation and local voltage dithering," *IEEE Journal of Solid-State Circuits*, Vol. 41, No. 1, pp. 238–245, 2005.

[169] M. K. Yadav, M. R. Casu, and M. Zamboni, "DVFS based on voltage dithering and clock scheduling for GALS systems," *Proceedings of the IEEE International Symposium on Asynchronous Circuits and Systems*, pp. 118–125, May 2012.

[170] C.-H. Huang and W.-J. Chen, "A spatial–temporal error spreading technique based on voltage dithering demonstrates a power savings of 35% in a real-time video processing datapath without timing-error detection and correction," *IEEE Solid-State Circuits Letters*, Vol. 3, pp. 378–381, September 2020.

[171] S. K. Lee, T. Tong, X. Zhang, D. Brooks, and G.-Y. Wei, "A 16-core voltage-stacked system with an integrated switched-capacitor DC-DC converter," *Proceedings of the IEEE Symposium on VLSI Circuits Circuits*, pp. C318–C319, June 2015.

[172] S. K. Lee, D. Brooks, and G.-Y. Wei, "Evaluation of voltage stacking for near-threshold multicore computing," *Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design*, pp. 373–378, July 2012.

[173] P. Mantovani, E. G. Cota, K. Tien, C. Pilato, G. Di Guglielmo, K. Shepard, and L. P. Carloni, "An FPGA-based infrastructure for fine-grained DVFS analysis in high-performance embedded systems," *Proceedings of the 53rd Annual Design Automation Conference*, p. 157, June 2016.

[174] J. E. Myers, "Unregulated voltage stacked memory," August 2019, US Patent 10,395,724.

[175] R. T. Possignolo, E. Ebrahimi, E. K. Ardestani, A. Sankaranarayanan, J. L. Briz, and J. Renau, "GPU NTC process variation compensation with voltage stacking," *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 26, No. 9, pp. 1713–1726, September 2018.

[176] K. Mazumdar and M. R. Stan, "Charge recycling on-chip DC-DC conversion for near-threshold operation," *Proceedings of the IEEE Subthreshold Microelectronics Conference*, pp. 1–3, October 2012.

[177] R. Zhang, K. Mazumdar, B. H. Meyer, K. Wang, K. Skadron, and M. R. Stan,

"A cross-layer design exploration of charge-recycled power-delivery in many-layer 3D-IC," *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 1–6, June 2015.

[178] H. Esmaeilzadeh, E. Blem, R. S. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," *Proceedings of the IEEE Annual international symposium on computer architecture*, pp. 365–376, June 2011.

[179] W. Lee, K. Han, Y. Wang, T. Cui, S. Nazarian, and M. Pedram, "TEI-Power: Temperature effect inversion–aware dynamic thermal management," *ACM Transactions on Design Automation of Electronic Systems*, Vol. 22, No. 3, p. 51, April 2017.

[180] Y. Pu, *et al.*, "Misleading energy and performance claims in sub/near threshold digital systems," *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 625–631, November 2010.

[181] M. Ashouei, H. Luijmes, J. Stuijt, and J. Huisken, "Novel wide voltage range level shifter for near-threshold designs," *Proceedings of the IEEE International Conference on Electronics, Circuits and Systems*, pp. 285–288, December 2010.

[182] E. Cai and D. Marculescu, "TEI-turbo: Temperature effect inversion-aware turbo boost for FinFET-based multi-core systems," *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 500–507, November 2015.

[183] D. Ernst, *et al.*, "Razor: A low-power pipeline based on circuit-level timing speculation," *Proceedings of the IEEE/ACM International Symposium on Microarchitecture*, pp. 7–18, December 2003.

[184] L. Lai, V. Chandra, R. C. Aitken, and P. Gupta, "SlackProbe: A flexible and efficient in situ timing slack monitoring methodology," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 33, No. 8, pp. 1168–1179, July 2014.

[185] Y. Zhang, M. Khayatzadeh, K. Yang, M. Saligane, N. Pinckney, M. Alioto,

D. Blaauw, and D. Sylvester, "iRazor: Current-based error detection and correction scheme for PVT variation in 40-nm ARM Cortex-r4 processor," *IEEE Journal of Solid-State Circuits*, Vol. 53, No. 2, pp. 619–631, October 2017.

[186] H. A. Balef, H. Fatemi, K. Goossens, and J. P. De Gyvez, "Timing speculation with optimal in situ monitoring placement and within-cycle error prevention," *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 27, No. 5, pp. 1206–1217, May 2019.

[187] F. A. Kuentzer, L. R. Juracy, and A. M. Amory, "On the reuse of timing resilient architecture for testing path delay faults in critical paths," *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition*, pp. 379–384, March 2018.

[188] D. Ernst, S. Das, S. Lee, D. Blaauw, T. Austin, T. Mudge, N. S. Kim, and K. Flautner, "Razor: Circuit-level correction of timing errors for low-power operation," *IEEE Micro*, Vol. 24, No. 6, pp. 10–20, November 2004.

[189] I. Kwon, S. Kim, D. Fick, M. Kim, Y.-P. Chen, and D. Sylvester, "Razor-lite: A light-weight register for error detection by observing virtual supply rails," *IEEE Journal of Solid-State Circuits*, Vol. 49, No. 9, pp. 2054–2066, 2014.

[190] D. Fick, N. Liu, Z. Foo, M. Fojtik, J.-s. Seo, D. Sylvester, and D. Blaauw, "In situ delay-slack monitor for high-performance processors using an all-digital self-calibrating 5ps resolution time-to-digital converter," *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 188–189, February 2010.

[191] S. Das, C. Tokunaga, S. Pant, W.-H. Ma, S. Kalaiselvan, K. Lai, D. M. Bull, and D. T. Blaauw, "RazorII: In situ error detection and correction for PVT and SER tolerance," *IEEE Journal of Solid-State Circuits*, Vol. 44, No. 1, pp. 32–48, December 2008.

[192] M. S. Hossain and I. Savidis, "Multi-voltage domain power distribution network for optimized ultra-low voltage clock delivery," *Proceedings of the IEEE International Green and Sustainable Computing Conference*, pp. 1–8, October 2018.

[193] J. Zhang, K. Rangineni, Z. Ghodsi, and S. Garg, "Thundervolt: Enabling aggressive voltage underscaling and timing error resilience for energy efficient deep learning accelerators," *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 1–6, June 2018.

[194] Y. Chen, J. Emer, and V. Sze, "Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks," *ACM SIGARCH Computer Architecture News*, Vol. 44, No. 3, pp. 367–379, June 2016.

[195] X. Wei, C. H. Yu, P. Zhang, Y. Chen, Y. Wang, H. Hu, Y. Liang, and J. Cong, "Automated systolic array architecture synthesis for high throughput CNN inference on FPGAs," *Proceedings of the Annual Design Automation Conference*, pp. 1-6, June 2017.

[196] H. Kwon, L. Lai, M. Pellauer, T. Krishna, Y. Chen, and V. Chandra, "Heterogeneous dataflow accelerators for multi-DNN workloads," *Proceedings of the IEEE International Symposium on High-Performance Computer Architecture*, pp. 71–83, March 2021.

[197] N. D. Lane, S. Bhattacharya, P. Georgiev, C. Forlivesi, and F. Kawsar, "An early resource characterization of deep learning on wearables, smartphones and internet-of-things devices," *Proceedings of the International Workshop on Internet of Things Towards Applications*, pp. 7–12, November 2015.

[198] Z. Jiang, S. Yin, J.-S. Seo, and M. Seok, "C3SRAM: In-Memory-Computing SRAM Macro Based on Capacitive-Coupling Computing," *IEEE Solid-State Circuits Letters*, Vol. 2, No. 9, pp. 131–134, September 2019.

[199] J. Wang, X. Wang, C. Eckert, A. Subramaniyan, R. Das, D. Blaauw, and D. Sylvester, "A 28-nm compute SRAM with bit-serial logic/arithmetic operations for programmable in-memory vector computing," *IEEE Journal of Solid-State Circuits*, Vol. 55, No. 1, pp. 76–86, January 2020.

[200] Z. Jiang, S. Yin, J.-S. Seo, and M. Seok, "C3SRAM: An in-memory-computing SRAM macro based on robust capacitive coupling computing mechanism," *IEEE Journal of Solid-State Circuits*, Vol. 55, No. 7, pp. 1888–1897, July 2020.

[201] X. Si, *et al.*, "A 28nm 64Kb 6T SRAM computing-in-memory macro with 8b MAC operation for AI edge chips," *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 246–248, February 2020.

[202] S. Yin, Z. Jiang, J.-S. Seo, and M. Seok, "XNOR-SRAM: In-memory computing SRAM macro for binary/ternary deep neural networks," *IEEE Journal of Solid-State Circuits*, Vol. 55, No. 6, pp. 1733–1743, June 2020.

[203] M. Ali, A. Jaiswal, S. Kodge, A. Agrawal, I. Chakraborty, and K. Roy, "IMAC: In-memory multi-bit multiplication and accumulation in 6T SRAM array," *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 67, No. 8, pp. 2521–2531, August 2020.

[204] J. Yue, *et al.*, "A 65nm computing-in-memory-based CNN processor with 2.9-to-35.8 TOPS/W system energy efficiency using dynamic-sparsity performance-scaling architecture and energy-efficient inter/intra-macro data reuse," *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 234–236, February 2020.

[205] Q. Dong, M. E. Sinangil, B. Erbagci, D. Sun, W. Khwa, H. Liao, Y. Wang, and J. Chang, "A 351 TOPS/W and 372.4 GOPS compute-in-memory SRAM macro in 7nm FinFET CMOS for machine-learning applications," *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 242–244, February 2020.

[206] J. Wang, X. Wang, C. Eckert, A. Subramaniyan, R. Das, D. Blaauw, and D. Sylvester, "A compute SRAM with bit-serial integer/floating-point operations for programmable in-memory vector acceleration," *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 224–226, February 2019.

[207] Q. Dong, S. Jeloka, M. Saligane, Y. Kim, M. Kawaminami, A. Harada, S. Miyoshi, M. Yasuda, D. Blaauw, and D. Sylvester, "A 4+ 2T SRAM for Searching and In-Memory Computing With 0.3-V V_{DDmin}," *IEEE Journal of Solid-State Circuits*, Vol. 53, No. 4, pp. 1006–1015, April 2018.

[208] J. Zhang, Z. Wang, and N. Verma, "A machine-learning classifier implemented in a standard 6T SRAM array," *Proceedings of the IEEE Symposium on VLSI Circuits*, pp. 1–2, June 2016.

[209] X. Si, *et al.*, "A twin-8T SRAM computation-in-memory macro for multiple-bit CNN-based machine learning," *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 396–398, February 2019.

[210] A. Biswas and A. P. Chandrakasan, "CONV-SRAM: An energy-efficient SRAM with in-memory dot-product computation for low-power convolutional neural networks," *IEEE Journal of Solid-State Circuits*, Vol. 54, No. 1, pp. 217–230, 2018.

[211] M. Capra, B. Bussolino, A. Marchisio, M. Shafique, G. Masera, and M. Martina, "An updated survey of efficient hardware architectures for accelerating deep convolutional neural networks," *Future Internet*, Vol. 12, No. 7, pp. 113 (1-22), July 2020.

[212] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," *Proceedings of the Neural Information Processing Systems*, pp. 1–9, December 2013.

[213] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," *Digital Signal Processing*, Vol. 73, pp. 1–15, October 2018.

[214] H. Kwon, L. Lai, T. Krishna, and V. Chandra, "HERALD: Optimizing heterogeneous DNN accelerators for edge devices," *arXiv preprint arXiv:1909.07437*, pp. 1–13, December 2020.

[215] A. Parashar, M. Rhu, A. Mukkara, A. Puglielli, R. Venkatesan, B. Khailany, J. Emer, S. W. Keckler, and W. J. Dally, "Scnn: An accelerator for compressed-sparse convolutional neural networks," *ACM SIGARCH Computer Architecture News*, Vol. 45, No. 2, pp. 27–40, May 2017.

[216] A. Samajdar, Y. Zhu, P. Whatmough, M. Mattina, and T. Krishna, "Scale-sim: Systolic CNN accelerator," *arXiv preprint arXiv:1811.02883*, 2018.

[217] L. Yang, Z. Yan, M. Li, H. Kwon, L. Lai, T. Krishna, V. Chandra, W. Jiang, and Y. Shi, "Co-exploration of neural architectures and heterogeneous ASIC accelerator designs targeting multiple tasks," *Proceedings of the ACM/EDAC/IEEE Design Automation Conference*, pp. 1-6, July 2020.

[218] Y. Chen, J. Emer, and V. Sze, "Using dataflow to optimize energy efficiency of deep neural network accelerators," *IEEE Micro*, Vol. 37, No. 3, pp. 12–21, June 2017.

[219] J. Lee, C. Kim, S. Kang, D. Shin, S. Kim, and H. Yoo, "Unpu: An energy-efficient deep neural network accelerator with fully variable weight bit precision," *IEEE Journal of Solid-State Circuits*, Vol. 54, No. 1, pp. 173–185, January 2019.

[220] A. Page, A. Jafari, C. Shea, and T. Mohsenin, "Sparcnet: A hardware accelerator for efficient deployment of sparse convolutional networks," *ACM Journal on Emerging Technologies in Computing Systems*, Vol. 13, No. 3, p. 31, May 2017.

[221] M. Peemen, S. A. A. Arnaud, B. Mesman, and H. Corporaal, "Memory-centric accelerator design for convolutional neural networks," *Proceedings of the IEEE International Conference on Computer Design*, pp. 13–19, October 2013.

[222] C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao, and J. Cong, "Optimizing FPGA-based accelerator design for deep convolutional neural networks," *Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pp. 161–170, February 2015.

[223] A. Aimar, *et al.*, "Nullhop: A flexible convolutional neural network accelerator based on sparse representations of feature maps," *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 30, No. 3, pp. 644–656, March 2019.

[224] W. Lu, G. Yan, J. Li, S. Gong, Y. Han, and X. Li, "Flexflow: A flexible dataflow accelerator architecture for convolutional neural networks," *Proceedings of the IEEE International Symposium on High Performance Computer Architecture*, pp. 553–564, February 2017.

[225] E. Qin, A. Samajdar, H. Kwon, V. Nadella, S. Srinivasan, D. Das, B. Kaul, and T. Krishna, "Sigma: A sparse and irregular gemm accelerator with flexible interconnects for dnn training," *Proceedings of the IEEE International Symposium on High Performance Computer Architecture*, pp. 58–70, February 2020.

[226] I. C. Foundation. Artificial Intelligence (AI). [Online]. Available: https://www.ibm.com/cloud/learn/what-is-artificial-intelligence

[227] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems.* O'Reilly Media, 2019.

[228] G. E. Dahl, T. N. Sainath, and G. E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 8609–8613, October 2013.

[229] A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial neural networks: A tutorial," *Computer*, Vol. 29, No. 3, pp. 31–44, March 1996.

[230] A. Sebastian, M. Le Gallo, R. Khaddam-Aljameh, and E. Eleftheriou, "Memory devices and applications for in-memory computing," *Nature Nanotechnology*, Vol. 15, No. 7, pp. 529–544, July 2020.

[231] A. Boroumand, *et al.*, "Google workloads for consumer devices: Mitigating data movement bottlenecks," *Proceedings of the ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, pp. 316–331, March 2018.

[232] Nvidia. NVIDIA A100 Tensor Core GPU Architecture. [Online]. Available: https://images.nvidia.com/aem-dam/en-zz/Solutions/data-center/nvidia-ampere-architecture-whitepaper.pdf

[233] Z. Yao, S. Cao, W. Xiao, C. Zhang, and L. Nie, "Balanced sparsity for efficient DNN inference on GPU," *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, pp. 5676–5683, December 2018.

[234] Nvidia. NVIDIA CUDA-X: GPU-Accelerated Libraries. [Online]. Available: https://developer.nvidia.com/gpu-accelerated-libraries

[235] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, Vol. 86, No. 11, pp. 2278–2324, November 1998.

[236] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, June 2016.

[237] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, June 2018.

[238] S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M. A. Horowitz, and W. J. Dally, "EIE: Efficient inference engine on compressed deep neural network," *ACM SIGARCH Computer Architecture News*, Vol. 44, No. 3, pp. 243–254, June 2016.

[239] Y. Chen, J. Emer, and V. Sze, "Using dataflow to optimize energy efficiency of deep neural network accelerators," *IEEE Micro*, Vol. 37, No. 3, pp. 12–21, June 2017.

[240] M. Holler, S. Tam, H. Castro, and R. Benson, "An electrically trainable artificial neural network (ETANN) with 10240 floating gate synapses," *Proceedings of the International Joint Conference on Neural Networks*, Vol. 2, pp. 191–196, August 1989.

[241] E. Säckinger, B. E. Boser, J. M. Bromley, Y. LeCun, and L. D. Jackel, "Application of the ANNA neural network chip to high-speed character recognition," *IEEE Transactions on Neural Networks*, Vol. 3, No. 3, pp. 498–505, March 1992.

[242] Y. LeCun, "Deep learning hardware: Past, present, and future," *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 12–19, February 2019.

[243] P. A. Merolla, *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, Vol. 345, No. 6197, pp. 668–673, August 2014.

[244] Y. Yan, D. Kappel, F. Neumärker, J. Partzsch, B. Vogginger, S. Höppner, S. Furber, W. Maass, R. Legenstein, and C. Mayr, "Efficient reward-based structural plasticity on a SpiNNaker 2 prototype," *IEEE Transactions on Biomedical Circuits and Systems*, Vol. 13, No. 3, pp. 579–591, June 2019.

[245] M. Davies. (2017, March) Loihi-A brief introduction. [Online]. Available: https://niceworkshop.org/wp-content/uploads/2018/05/Mike-Davies-NICE-Loihi-Intro-Talk-2018.pdf

[246] W. A. Simon, Y. M. Qureshi, A. Levisse, M. Zapater, and D. Atienza, "BLADE: A bitline accelerator for devices on the edge," *Proceedings of the IEEE Great Lakes Symposium on VLSI*, pp. 207–212, May 2019.

[247] E. Park, D. Kim, and S. Yoo, "Energy-efficient neural network accelerator based on outlier-aware low-precision computation," *Proceedings of the ACM/IEEE Annual International Symposium on Computer Architecture*, pp. 688–698, July 2018.

[248] H. Yoo, S. Park, K. Bong, D. Shin, J. Lee, and S. Choi, "A 1.93 TOPS/W scalable deep learning/inference processor with tetra-parallel mimd architecture for big data applications," *IEEE International Solid-State Circuits Conference*, pp. 80–81, February 2015.

[249] A. Yazdanbakhsh, K. Samadi, N. S. Kim, and H. Esmaeilzadeh, "Ganax: A unified mimd-simd acceleration for generative adversarial networks," *Proceedings of the Annual International Symposium on Computer Architecture*, pp. 650–661, June 2018.

[250] S. Sen, S. Jain, S. Venkataramani, and A. Raghunathan, "SparCE: Sparsity aware general-purpose core extensions to accelerate deep neural networks," *IEEE Transactions on Computers*, Vol. 68, No. 6, pp. 912–925, June 2019.

[251] M. Imani, S. Gupta, Y. Kim, and T. Rosing, "Floatpim: In-memory acceleration of deep neural network training with high precision," *Proceedings of the International Symposium on Computer Architecture*, pp. 802–815, June 2019.

[252] Z. Jiang, S. Yin, M. Seok, and J. Seo, "XNOR-SRAM: In-memory computing SRAM macro for binary/ternary deep neural networks," *Proceedings of the IEEE Symposium on VLSI Technology*, pp. 173–174, June 2018.

[253] P. Jokic, S. Emery, and L. Benini, "Improving memory utilization in convolutional neural network accelerators," *IEEE Embedded Systems Letters*, pp. 1–4, July 2020.

[254] S. Thoziyoor, J. H. Ahn, M. Monchiero, J. B. Brockman, and N. P. Jouppi, "A comprehensive memory modeling tool and its application to the design and analysis of future memory hierarchies," *ACM SIGARCH Computer Architecture News*, Vol. 36, No. 3, pp. 51–62, June 2008.

[255] J. Sim, S. Lee, and L. Kim, "An energy-efficient deep convolutional neural network inference processor with enhanced output stationary dataflow in 65-nm CMOS," *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 28, No. 1, pp. 87–100, September 2019.

[256] Y. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE Journal of Solid-State Circuits*, Vol. 52, No. 1, pp. 127–138, November 2016.

[257] Y. Chen, *et al.*, "Dadiannao: A machine-learning supercomputer," *Proceedings of the IEEE/ACM International Symposium on Microarchitecture*, pp. 609–622, December 2014.

[258] S. Zhang, Z. Du, L. Zhang, H. Lan, S. Liu, L. Li, Q. Guo, T. Chen, and Y. Chen, "Cambricon-X: An accelerator for sparse neural networks," *Proceedings of the IEEE/ACM International Symposium on Microarchitecture*, pp. 1–12, December 2016.

[259] J. Li, A. Louri, A. Karanth, and R. Bunescu, "GCNAX: A flexible and energy-efficient accelerator for graph convolutional neural networks," *Proceedings of the*

IEEE International Symposium on High-Performance Computer Architecture, pp. 775–788, March 2021.

[260] S. Chakradhar, M. Sankaradas, V. Jakkula, and S. Cadambi, "A dynamically configurable coprocessor for convolutional neural networks," *ACM SIGARCH Computer Architecture News*, Vol. 38, No. 3, pp. 247–257, June 2010.

[261] V. Gokhale, J. Jin, A. Dundar, B. Martini, and E. Culurciello, "A 240 g-ops/s mobile coprocessor for deep neural networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 682–687, June 2014.

[262] Z. Du, R. Fasthuber, T. Chen, P. Ienne, L. Li, T. Luo, X. Feng, Y. Chen, and O. Temam, "ShiDianNao: Shifting vision processing closer to the sensor," *ACM SIGARCH Computer Architecture News*, Vol. 43, No. 3, pp. 92–104, June 2015.

[263] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, "Deep learning with limited numerical precision," *International Conference on Machine Learning*, pp. 1737–1746, 2015.

[264] S. Yu, X. Sun, X. Peng, and S. Huang, "Compute-in-memory with emerging nonvolatile-memories: Challenges and prospects," *Proceedings of the IEEE Custom Integrated Circuits Conference*, pp. 1–4, March 2020.

[265] S. Bavikadi, P. R. Sutradhar, K. N. Khasawneh, A. Ganguly, and S. M. Pudukotai Dinakarrao, "A review of in-memory computing architectures for machine learning applications," *Proceedings of the IEEE Great Lakes Symposium on VLSI*, pp. 89–94, September 2020.

[266] F. Staudigl, F. Merchant, and R. Leupers, "A survey of neuromorphic computing-in-memory: Architectures, simulators and security," *IEEE Design & Test*, August 2021.

[267] H. S. Stone, "A logic-in-memory computer," *IEEE Transactions on Computers*, Vol. 100, No. 1, pp. 73–78, January 1970.

[268] D. Patterson, T. Anderson, N. Cardwell, R. Fromm, K. Keeton, C. Kozyrakis, R. Thomas, and K. Yelick, "A case for intelligent RAM," *IEEE Micro*, Vol. 17, No. 2, pp. 34–44, April 1997.

[269] S. Srinivasa, A. K. Ramanathan, J. Sundaram, D. Kurian, S. Gopal, N. Jain, A. Srinivasan, R. Iyer, V. Narayanan, and T. Karnik, "Trends and opportunities for SRAM based in-memory and near-memory computation," *Proceedings of the IEEE International Symposium on Quality Electronic Design*, pp. 547–552, April 2021.

[270] M. S. Hossain and I. Savidis, "Leakage reuse for energy efficient near-memory computing of heterogeneous DNN accelerators," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, Vol. 11, No. 4, pp. 762–775, December 2021.

[271] M. U. Mohammed, A. Nizam, and M. H. Chowdhury, "Performance stability analysis of sram cells based on different FinFET devices in 7nm technology," *Proceedings of the IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference*, pp. 1–3, October 2018.

[272] M. S. Hossain and I. Savidis, "Dynamic differential signaling based logic families for robust ultra-low power near-threshold computing," *Microelectronics Journal*, Vol. 102, pp. 1–14, August 2020.

[273] M. S. Hossain and I. Savidis, "Bi-directional input/output circuits with integrated level shifters for near-threshold computing," *Proceedings of the IEEE International Midwest Symposium on Circuits and Systems*, pp. 1240–1243, August 2017.

[274] J. M. Rabaey, A. P. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits*. 2nd Edition, Prentice Hall Englewood Cliffs, 2002.

[275] A. Bellaouar, "Current mode logic gates for low-voltage high speed applications," July 24 2001, US Patent 6,265,898.

[276] G. Palumbo, M. Pennisi, and M. Alioto, "A simple circuit approach to reduce delay variations in domino logic gates," *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 59, No. 10, pp. 2292–2300, October 2012.

[277] A. A. Mutlu and M. Rahman, "Statistical methods for the estimation of process variation effects on circuit operation," *IEEE Transactions on Electronics Packaging Manufacturing*, Vol. 28, No. 4, pp. 364–375, October 2005.

[278] K. A. Bowman, A. R. Alameldeen, S. T. Srinivasan, and C. B. Wilkerson, "Impact of die-to-die and within-die parameter variations on the throughput distribution of multi-core processors," *Proceedings of the IEEE International Symposium on Low Power Electronics and Design*, pp. 50–55, August 2007.

[279] M. Assar, P. C. Agarwal, and V. Bril, "CMOS low power mixed voltage bidirectional I/O buffer," April 1994, uS Patent 5300835.

[280] D. G. Kim, E. G. Kim, and J. Y. Kim, "Semiconductor integrated circuit device with a fail-safe IO circuit and electronic device including the same," February 2010, uS Patent 7656185.

[281] M. Lanuzza, F. Crupi, S. Rao, R. D. Rose, S. Strangio, and G. Iannaccone, "An ultralow-voltage energy-efficient level shifter," *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol. 64, No. 1, pp. 61–65, January 2017.

[282] J. Zhou, C. Wang, X. Liu, X. Zhang, and M. Je, "An ultra-low voltage level shifter using revised wilson current mirror for fast and energy-efficient wide-range voltage conversion from sub-threshold to I/O voltage," *IEEE Transactions on Circuits and Systems*, Vol. 62, No. 3, pp. 697–706, January 2015.

[283] M. Hwang, A. Raychowdhury, K. Kim, and K. Roy, "A 85mV 40nW process-tolerant subthreshold 8×8 FIR filter in 130nm technology," *Proceedings of the IEEE Symposium on VLSI Circuits*, pp. 154–155, October 2007.

[284] S. Hanson, *et al.*, "Performance and variability optimization strategies in a sub-200mV, 3.5 pJ/Inst, 11nW subthreshold processor," *Proceedings of the IEEE Symposium on VLSI Circuits*, pp. 152–153, June 2007.

[285] S. C. Jocke, J. F. Bolus, S. N. Wooters, A. D. Jurik, A. C. Weaver, T. N. Blalock, and B. H. Calhoun, "A 2.6-μW sub-threshold mixed-signal ECG SoC," *Proceedings of the IEEE Symposium on VLSI Circuits*, pp. 60–61, June 2009.

[286] S. Luetkemeier, T. Jungeblut, M. Porrmann, and U. Rueckert, "A 200mV 32b subthreshold processor with adaptive supply voltage control," *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 484–486, February 2012.

[287] A. P. Chandrakasan, D. C. Daly, D. F. Finchelstein, J. Kwong, Y. K. Ramadass, M. E. Sinangil, V. Sze, and N. Verma, "Technologies for ultra-dynamic voltage scaling," *Proceedings of the IEEE*, Vol. 98, No. 2, pp. 191–214, February 2010.

[288] M. Alioto, "Understanding DC behavior of subthreshold CMOS logic through closed-form analysis," *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 57, No. 7, pp. 1597–1607, July 2010.

[289] D. Marković, C. C. Wang, L. P. Alarcon, T. Liu, and J. M. Rabaey, "Ultralow-power design in near-threshold region," *Proceedings of the IEEE*, Vol. 98, No. 2, pp. 237–252, February 2010.

[290] J. Zhu, L. Pan, Y. Yan, D. Wu, and H. He, "A fast application-based supply voltage optimization method for dual voltage FPGA," *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 22, No. 12, pp. 2629–2634, January 2014.

[291] B. H. Calhoun and A. Chandrakasan, "Characterizing and modeling minimum energy operation for subthreshold circuits," *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 90–95, August 2004.

[292] D. Markovic, V. Stojanovic, B. Nikolic, M. A. Horowitz, and R. W. Brodersen, "Methods for true energy-performance optimization," *IEEE Journal of Solid-State Circuits*, Vol. 39, No. 8, pp. 1282–1293, August 2004.

[293] E. Morifuji, T. Yoshida, M. Kanda, S. Matsuda, S. Yamada, and F. Matsuoka, "Supply and threshold-voltage trends for scaled logic and SRAM MOSFETs," *IEEE Transactions on Electron Devices*, Vol. 53, No. 6, pp. 1427–1432, May 2006.

[294] R. Vaddi, S. Dasgupta, and R. P. Agarwal, "Device and circuit co-design robustness studies in the subthreshold logic for ultralow-power applications for

32 nm CMOS," *IEEE Transactions on Electron Devices*, Vol. 57, No. 3, pp. 654–664, February 2010.

[295] K. Choi, R. Soma, and M. Pedram, "Dynamic voltage and frequency scaling based on workload decomposition," *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 174–179, August 2004.

[296] M. S. Hossain and I. Savidis, "Noise constrained optimum selection of supply voltage for IoT applications," *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 1–5, May 2018.

[297] K. Han, J. Li, A. B. Kahng, S. Nath, and J. Lee, "A global-local optimization framework for simultaneous multi-mode multi-corner clock skew variation reduction," *Proceedings of the 52nd Annual Design Automation Conference*, p. 26, June 2015.

[298] J. M. Rabaey, A. P. Chandrakasan, and B. Nikolic, *Digital integrated circuits*, 2nd *Edition*. Prentice Hall, Englewood Cliffs, 2002.

[299] M. Seok, D. Blaauw, and D. Sylvester, "Robust clock network design methodology for ultra-low voltage operations," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, Vol. 1, No. 2, pp. 120–130, August 2011.

[300] L. Lin, S. Jain, and M. Alioto, "Reconfigurable clock networks for random skew mitigation from subthreshold to nominal voltage," *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 440–441, February 2017.

[301] M. Seok, D. Blaauw, and D. Sylvester, "Clock network design for ultra-low power applications," *Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design*, pp. 271–276, August 2010.

[302] J. R. Tolbert, X. Zhao, S. K. Lim, and S. Mukhopadhyay, "Analysis and design of energy and slew aware subthreshold clock systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 30, No. 9, pp. 1349–1358, August 2011.

[303] J. R. Tolbert, X. Zhao, S. K. Lim, and S. Mukhopadhyay, "Slew-aware clock tree design for reliable subthreshold circuits," *Proceedings of the ACM/IEEE*

International Symposium on Low Power Electronics and Design, pp. 15–20, August 2009.

[304] J. Kil, J. Gu, and C. H. Kim, "A high-speed variation-tolerant interconnect technique for sub-threshold circuits using capacitive boosting," *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 16, No. 4, pp. 456–465, April 2008.

[305] R. Kumar, "Power source for clock distribution network," August 2016, US Patent 9419589.

[306] Y. K. Ramadass and A. P. Chandrakasan, "Minimum energy tracking loop with embedded DC–DC converter enabling ultra-low-voltage operation down to 250 mV in 65 nm CMOS," *IEEE Journal of Solid-State Circuits*, Vol. 43, No. 1, pp. 256–265, January 2008.

[307] T. Na, J. H. Ko, and S. Mukhopadhyay, "Clock data compensation aware digital circuits design for voltage margin reduction," *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 64, No. 9, pp. 2401–2413, June 2017.

[308] K. A. Bowman, S. Raina, J. T. Bridges, D. J. Yingling, H. H. Nguyen, B. R. Appel, Y. N. Kolla, J. Jeong, F. I. Atallah, and D. 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*, Vol. 51, No. 1, pp. 8–17, January 2016.

[309] K. L. Wong, T. Rahal-Arabi, M. Ma, and G. Taylor, "Enhancing microprocessor immunity to power supply noise with clock-data compensation," *IEEE Journal of Solid-State Circuits*, Vol. 41, No. 4, pp. 749–758, March 2006.

[310] P. Chiu and M. Chen, "High-to-low level shifter," September 2005.

[311] R. Nowakowski and N. Tang, "Efficiency of synchronous versus nonsynchronous buck converters," *Texas Instruments Incorporated*, 2009 [Online], Avaiable: http://www.ti.com/lit/an/slyt358/slyt358.pdf.

[312] D. Jauregui, B. Wang, and R. Chen, "Power loss calculation with common source inductance Consideration for Synchronous Buck Converters," *Texas Instruments Incorporated*, June 2011.

[313] A. D. Sheet, "Analog Devices," 2018, available from http://www.analog.com/media/en/technical-documentation/data-sheets/ADP1851.pdf.

[314] I. Team, "International Technology Roadmap for Semiconductors," 2013, available from http://www.itrs.net/ITRS

[315] M. Hrishikesh, D. Burger, N. P. Jouppi, S. W. Keckler, K. I. Farkas, and P. Shivakumar, "The optimal logic depth per pipeline stage is 6 to 8 FO4 inverter delays," *Proceedings of the IEEE International Symposium of Computer Architecture*, pp. 14–24, May 2002.

[316] A. Tajalli and Y. Leblebici, "Design trade-offs in ultra-low-power digital nanoscale CMOS," *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 58, No. 9, pp. 2189–2200, September 2011.

[317] A. Esmaili, M. Nazemi, and M. Pedram, "Modeling processor idle times in MP-SoC platforms to enable integrated DPM, DVFS, and task scheduling subject to a hard deadline," *Proceedings of the Asia and South Pacific Design Automation Conference*, pp. 532–537, January 2019.

[318] T. Tong, S. K. Lee, X. Zhang, D. Brooks, and G. Wei, "A fully integrated reconfigurable switched-capacitor DC-DC converter with four stacked output channels for voltage stacking applications," *IEEE Journal of Solid-State Circuits*, Vol. 51, No. 9, pp. 2142–2152, July 2016.

[319] K. Chong, K. Chang, B. Gwee, and J. S. Chang, "Synchronous-logic and globally-asynchronous-locally-synchronous (GALS) acoustic digital signal processors," *IEEE Journal of Solid-State Circuits*, Vol. 47, No. 3, pp. 769–780, March 2012.

[320] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, "Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits," *Proceedings of the IEEE*, Vol. 91, No. 2, pp. 305–327, February 2003.

[321] A. Muttreja, N. Agarwal, and N. K. Jha, "CMOS logic design with independent-gate FinFETs," *Proceedings of the International Conference on Computer Design*, pp. 560–567, August 2007.

[322] E. Kultursay, K. Swaminathan, V. Saripalli, V. Narayanan, M. T. Kandemir, and S. Datta, "Performance enhancement under power constraints using heterogeneous CMOS-TFET multicores," *Proceedings of the IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*, pp. 245–254, October 2012.

[323] S. Moradi, N. Qiao, F. Stefanini, and G. Indiveri, "A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (DYNAPs)," *IEEE Transactions on Biomedical Circuits and Systems*, Vol. 12, No. 1, pp. 106–122, August 2017.

[324] H. Mair, G. Gammie, A. Wang, S. Gururajarao, I. Lin, H. Chen, W. Kuo, A. Rajagopalan, W. Ge, and R. Lagerquist, "A highly integrated smartphone SoC featuring a 2.5 GHz octa-core CPU with advanced high-performance and low-power techniques," *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 1–3, March 2015.

[325] A. Iyer and D. Marculescu, "Power efficiency of voltage scaling in multiple clock, multiple voltage cores," *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 379–386, November 2002.

[326] S. Rusu, S. Tam, H. Muljono, *et al.*, "A dual-core multi-threaded Xeon processor with 16 MB L3 cache," *Proceedings of the IEEE International Solid State Circuits Conference*, pp. 315–324, February 2006.

[327] A. S. Leon, K. W. Tam, J. L. Shin, D. Weisner, and F. Schumacher, "A power-efficient high-throughput 32-thread SPARC processor," *IEEE Journal of Solid State Circuits*, Vol. 42, No. 1, pp. 7–16, January 2007.

[328] S. Hanson, M. Seok, Y. Lin, Z. Foo, D. Kim, Y. Lee, N. Liu, D. Sylvester, and D. Blaauw, "A low-voltage processor for sensing applications with picowatt standby mode," *IEEE Journal of Solid-State Circuits*, Vol. 44, No. 4, pp. 1145–1155, April 2009.

[329] M. S. Gupta, J. L. Oatley, R. Joseph, G. Wei, and D. M. Brooks, "Understanding voltage variations in chip multiprocessors using a distributed power-delivery network," *Proceedings of the Conference on Design, Automation and Test in Europe*, pp. 624–629, April 2007.

[330] L. D. Smith and E. Bogatin, *Principles of power integrity for PDN design: Robust and cost effective design for high speed digital products*. Prentice Hall, 2017.

[331] D. Pathak and I. Savidis, "On-chip power supply noise suppression through hyperabrupt junction varactors," *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 26, No. 11, pp. 2230–2240, November 2018.

[332] A. Lungu, P. Bose, A. Buyuktosunoglu, and D. J. Sorin, "Dynamic power gating with quality guarantees," *Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design*, pp. 377–382, August 2009.

[333] Y. Çakmak, W. Toms, J. Navaridas, and M. Luján, "Cyclic power-gating as an alternative to voltage and frequency scaling," *IEEE Computer Architecture Letters*, Vol. 15, No. 2, pp. 77–80, September 2015.

[334] M. S. Hossain and I. Savidis, "Reusing leakage current for improved energy efficiency of multi-voltage systems," *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 1–5, May 2019.

[335] M. S. Hossain and I. Savidis, "Recycling of unused leakage current for energy efficient multi-voltage systems," *Microelectronics Journal*, Vol. 101, No. 7, pp. 1–16, July 2020.

[336] T. Nakada, H. Yanagihashi, H. Nakamura, K. Imai, H. Ueki, T. Tsuchiya, and M. Hayashikoshi, "Energy-aware task scheduling for near real-time periodic tasks on heterogeneous multicore processors," *IEEE International Conference on Very Large Scale Integration*, pp. 1–6, December 2017.

[337] H. Topcuoglu, S. Hariri, and M. Wu, "Performance-effective and low-complexity task scheduling for heterogeneous computing," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 13, No. 3, pp. 260–274, August 2002.

[338] H. Jiang and M. Marek-Sadowska, "Power gating scheduling for power/ground noise reduction," *Proceedings of the IEEE/ACM Design Automation Conference*, pp. 980–985, June 2008.

[339] J. Chen and X. Ran, "Deep learning with edge computing: A review," *Proceedings of the IEEE*, Vol. 107, No. 8, pp. 1655–1674, August 2019.

[340] A. Reuther, P. Michaleas, M. Jones, V. Gadepally, S. Samsi, and J. Kepner, "Survey and benchmarking of machine learning accelerators," *arXiv preprint arXiv:1908.11348*, August 2019.

[341] J. Choi, S. Venkataramani, V. Srinivasan, K. Gopalakrishnan, Z. Wang, and P. Chuang, "Accurate and efficient 2-bit quantized neural networks," *Proceedings of the Systems and Machine Learning Conference*, pp. 1–12, April 2019.

[342] X. Sun, N. Wang, C. Chen, J. Ni, A. Agrawal, X. Cui, S. Venkataramani, K. E. Maghraoui, V. V. Srinivasan, and K. Gopalakrishnan, "Ultra-low precision 4-bit training of deep neural networks," *Advances in Neural Information Processing Systems*, Vol. 33, pp. 1–12, December 2020.

[343] J. Lee, S. Kang, J. Lee, D. Shin, D. Han, and H. Yoo, "The hardware and algorithm co-design for energy-efficient DNN processor on edge/mobile devices," *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 67, No. 10, pp. 3458–3470, October 2020.

[344] E. Karl, Y. Wang, Y. Ng, Z. Guo, F. Hamzaoglu, U. Bhattacharya, K. Zhang, K. Mistry, and M. Bohr, "A 4.6 GHz 162Mb SRAM design in 22nm tri-gate CMOS technology with integrated active V_{MIN}-enhancing assist circuitry," *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 230–232, February 2012.

[345] E. A. Burton, G. Schrom, F. Paillet, J. Douglas, W. J. Lambert, K. Radhakrishnan, and M. J. Hill, "FIVR–fully integrated voltage regulators on 4th generation intel core SoCs," *Proceedings of the IEEE Applied Power Electronics Conference and Exposition*, pp. 432–439, March 2014.

[346] M. Clinton, *et al.*, "A low-power and high-performance 10 nm SRAM architecture for mobile applications," *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 210–211, February 2017.

[347] A. Banerjee, M. E. Sinangil, J. Poulton, C. T. Gray, and B. H. Calhoun, "A reverse write assist circuit for SRAM dynamic write V_{MIN} tracking using canary SRAMs," *Proceedings of the ACM/IEEE International Symposium on Quality Electronic Design*, pp. 1–8, March 2014.

[348] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, pp. 1–9, April 2017.