OVERCOMING NANOSCALE VARIATIONS THROUGH STATISTICAL ERROR COMPENSATION

BY

TIANQI GAO

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2015

Urbana, Illinois

Adviser:

Professor Naresh Shanbhag

ABSTRACT

Increasingly severe parameter variations that are observed in advanced nanoscale technologies create great obstacles in designing high-performance, next-generation digital integrated circuits (ICs). Conventional design principles impose increased design margins in power supply, device sizing, and operating frequency, leading to overly conservative designs which prevent the realization of potential benefits from nanotechnology advances. In response, robust digital circuit design techniques have been developed to overcome processing non-idealities. Statistical error compensation (SEC) is a class of system-level, communication-inspired techniques for designing energy-efficient and robust systems. In this thesis, stochastic sensor network on chip (SSNOC), a known SEC technique, is applied to a computational kernel implemented with carbon nanotube field-effect transistors (CNFETs). With the aid of a well-developed CNFET delay distribution modeling method, circuit simulations show up to 90× improvement of the SSNOC-based design in the circuit yield over the conventional design. The results verify the robustness of an SEC-based design under CNFET-specific variations. The error resiliency of SEC allows CNFET circuits to operate with reduced design margins under relaxed processing requirements, while concurrently maintaining the desired application-level performance.

ACKNOWLEDGMENTS

I would like to thank my thesis advisor, Professor Naresh Shanbhag, for supporting my graduate study and for his wise and patient guidance. I would also acknowledge Gage Hills and Professor Subhasish Mitra from the Stanford Robust System Group. Gage contributed to the delay modeling used in the simulations, assisted in processing the data, and provided insightful suggestions. My special thanks also go to Eric Kim, who offered valuable suggestions and help with my research project. This project was supported in part by systems on nanoscale information fabrics (SONIC), one of the six SRC STARnet Centers, sponsored by MARCO and DARPA.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION AND BACKGROUND

Variations in nanoscale technologies create huge challenges in designing robust and power-efficient digital ICs. Fortunately, many techniques have been developed to improve the robustness of digital systems against variations. In this chapter, the problem of variations is presented, followed by a review of selected robust design techniques for very-large-scale integrated (VLSI) circuits.

1.1 Variations in Nanoscale Technologies

1.1.1 Variations in Deeply Scaled CMOS

Today, in deeply scaled CMOS technology, variations in process, voltage and temperature (PVT) create significant reliability issues for digital VLSI systems [1–3]. Even though variations were present in earlier ICs, they did not create serious issues because the transistor dimensions were large enough that the effects of PVT variations were negligible. As the transistor size keeps shrinking, however, parameter variations are becoming more problematic for the IC industry. Since 2004, when transistors with channel lengths smaller than 90 nanometers were being produced, it was realized that PVT variations created huge challenges in designing high-performance and low-cost circuits [1,2]. Even though process control has improved and better fabrication equipments have been developed in recent several decades, the problem of PVT variations is becoming more critical in designing current and future generation ICs.

In general, variations can be classified into two types: global and local. Global variations are due to inconsistencies in the manufacturing environment, resulting in variations in the electrical properties between dies. These inconsistencies usually shift the properties of all transistors on the same die in a similar way [1]; thus, global variations are referred to as die-to-die (D2D) variations. On the other hand, local variations are unpredictable differences between transistors on the same die, and are commonly known as within-die (WID) variations. There are many sources of local variations, with one of the most important sources being random dopant fluctuations [1]. In advanced CMOS processes, a transistor contains only a few hundred dopants. Thus, if the number of dopants varies slightly, the gate voltage needed to turn on a transistor can vary [1]. Local variations usually impact transistors in an independent manner, which makes the circuit performance extremely unpredictable [1]. Besides process variations, supply voltage fluctuations and non-uniform operating temperatures can also account for circuit parameter variations.

Transistors have random features due to PVT variations, making deeply scaled CMOS circuits behave in a non-deterministic manner. The impact of these variations on a digital circuit can usually be modeled as the circuit delay variations. For example, threshold voltage (V_t) variations can generate random propagation delays in logic gates. The transition time at the output node of a logic gate depends on the drive current I_{DS}, and the load capacitance. When a transistor is in the saturation mode, I_{DS} is proportional to the overdrive voltage, which is defined as the difference between the gate-to-source voltage V_{GS} and V_t. Thus, V_t variations lead to variations in I_{DS}, resulting in random propagation delays. Figure 1.1 shows that the variance of V_t increases with transistor dimension scaling [1], and it is expected that the variance of the circuit delay will also increase.

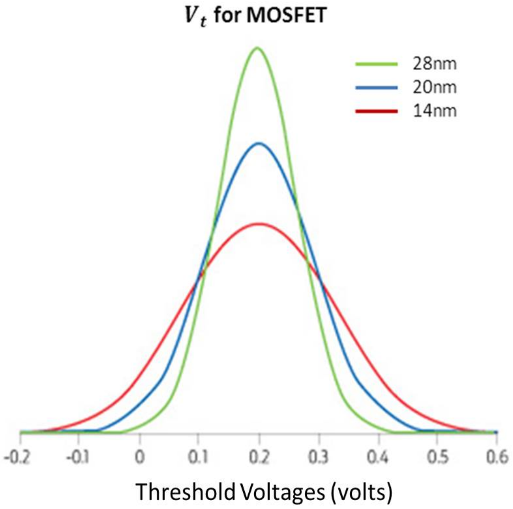


Figure 1.1: Threshold voltage distribution for scaled CMOS technologies [1].

1.1.2 Emerging Technologies

As predicted by Gordon Moore about half a century ago [4], researchers and engineers have managed to shrink technology features and made faster, smaller, and cheaper ICs with silicon-based transistors. Moore's law states that the number of transistors integrated on a chip doubles approximately every two years; and this law has been valid for the past 50 years and is the driving force behind the information age. However, the rate of scaling has slowed down during the past decade; and, as many believe that the scaling of silicon transistors will eventually reach fundamental limits imposed by physics, some researchers predict that Moore's law will end in the near future [5, 6]. Regardless of when and how Moore's law will cease, much research effort is being spent on discovering beyond CMOS devices.

Many newly developed beyond-CMOS devices have shown promising properties [7]. For example, CNFETs, compared with CMOS technologies, have been projected to provide an order of magnitude improvement in the energy-delay product (EDP), with great scalability [8]. Graphene-based transistors present high carrier mobilities and provide opportunities to design high-performance radio frequency (RF) circuits [9, 10]. Spin-based transistors have the potential of realizing ultra low-power computations and enabling novel circuit applications of non-volatile logic and reconfigurable logic [11].

Beyond-CMOS technologies present great potential in building novel, low-power and high-performance ICs, but many exhibit PVT variations as observed in deeply scaled CMOS [12–15]. Further, process variations in beyond-CMOS devices could be more severe than those in CMOS [16].

1.2 A Review of Robust Digital VLSI Design Techniques

1.2.1 The Need of Robust Digital VLSI Design Techniques

Deeply scaled CMOS and emerging beyond-CMOS processes hold great potential in designing high-performance, low-power, next-generation ICs, but they both suffer from severe variations, preventing their theoretical improvements to be realized. To overcome PVT variations, traditional design methodology requires safety margins in design parameters, such as supply voltage, operating clock or transistor size. This conservative approach leads to increased power and lower performance. For example, circuits with advanced devices may not be able to run at a higher clock frequency because of variations, even if the electrical properties of the device indicate smaller circuit delays. This effect can be illustrated in Figure 1.2, where dummy path delay histograms of a circuit built using CMOS and using a new device are plotted. The delay needed for a combinational logic circuit to respond to new inputs and produce correct outputs is a random variable, and a path delay histogram is often used by designers to view this effect. In the path delay histogram, the y-axis plots the occurrences of possible path delays, and the x-axis shows the possible delays in a circuit. A path delay histogram is equivalent to the path delay distribution if the input distribution is assumed to be uniform. The largest delay is commonly referred to as the critical path delay, which sets the clock frequency. Figure 1.2 illustrates that the improvement in circuit frequency can be reduced due to variations.

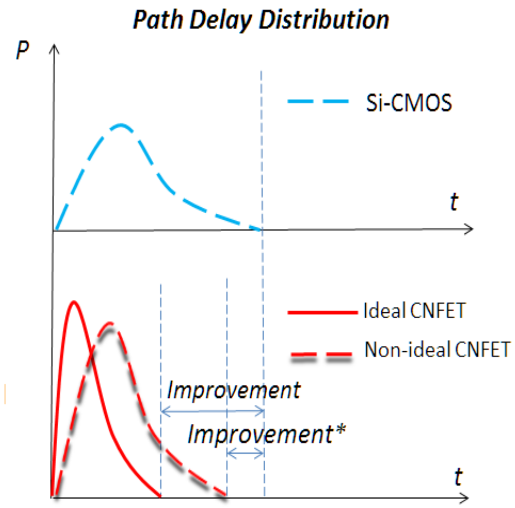


Figure 1.2: Path delay histogram (dummy plots) showing loss in throughput improvement between CNFET circuits and CMOS circuits due to variations.

Simply increasing safety margins can be overly conservative and can lead to a large design overhead in terms of area or power consumption; however, not increasing safety margins under severe variations can produce circuit errors. Thus, robust VLSI design techniques are greatly needed. Over the past few decades, many robust digital circuit design techniques have been developed. Variation tolerant techniques intentionally reduce PVT variations in order to lower the design overhead; while error-resilient techniques aim to design robust and energy-efficient circuits in the presence of hardware errors. The latter approach can significantly reduce the overhead by improving circuits error resiliency, allowing circuits to operate under PVT variations with reduced safety margins. In this chapter, several robust design techniques are reviewed due to their popularities and outstanding performance. The focus of this review is not only to reveal the design philosophy, but also to evaluate the effectiveness of these techniques in terms of robustness improvements. A brief discussion of the limitations of each technique is presented at the end of each subsection.

1.2.2 Critical Path Isolation

Critical path isolation for timing adaptiveness (CRISTA) is a recently developed, low-power, and variation-tolerant design paradigm for digital circuits [17]. A recent project reported that an average of 60% power reduction can be achieved by implementing CRISTA on a set of benchmark circuits with 18% die area overhead, compared to conventional designs [17].

The success of CRISTA relies on carefully exploiting the statistical timing behavior of combinational logic circuits. As previously noted in Section 1.2.1, the critical path delay is used by designers to set the clock period. However, when the critical path is idle, which happens much more often than not, correct results can be computed before the end of a clock period, resulting in clock period overhead. Based on this effect, CRISTA manipulates the path delay distribution to create opportunities for relaxing design constraints. Figure 1.3 [17] shows an example of a desired path delay distribution favored by CRISTA. In the example, two groups of delays are separated from each other. The group with smaller delays can be referred to as non-critical delays and the group with larger delays as critical delays. There are two reasons of identifying a group of delays as critical [17]. First, for complicated digital circuits, it is hard to identify the longest path since there are probably multiple paths with very large delays. Second, if PVT variations are taken into consideration, the largest path delay may vary. In Figure 1.3, a large gap is placed between non-critical path delays and the one-cycle delay target, which sets the clock frequency; due to this gap, relaxed design parameters such as scaled voltages can be applied, in order to reduce the dynamic power consumption. When critical path delays are activated, the computation cannot be done within one clock, and the circuit will switch to the mode of two-cycle operation to allocate more computation time. If critical paths are not activated, the circuit can revert to the normal mode in the next clock cycle.

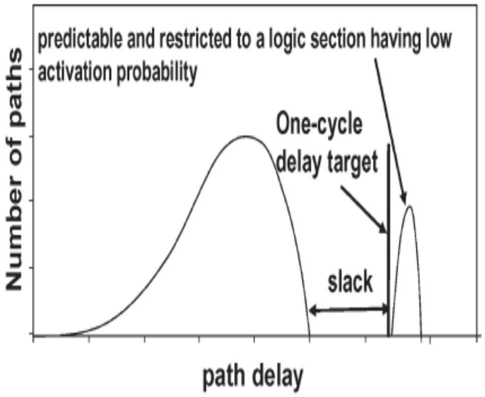


Figure 1.3: Desired path delay distributions for CRISTA [17].

According to [17], three design principles are necessary for CRISTA:

1. Critical path delays need to be isolated and a large gap needs to be created between critical delays and non-critical delays.

2. The probability of critical path excitations needs to be minimized.

3. The circuit operation needs to be switched to two-cycle mode when critical paths are excited.

The first principle is to ensure that there is enough room to relax design constraints, such as voltage and frequency; and the size of the timing slack also indicates the tolerance to variations. The second principle can help to minimize throughput penalties from switching to the two-cycle operation mode. The third principle is needed to avoid timing violations.

Based on the design principles, CRISTA design methodology includes the following steps. First, circuit synthesis with an input-based logic partition is conducted to isolate the critical paths. During this process, Shannon-expansion-based partitioning [17, 18] is applied to expand the Boolean expression f in the following way: $f(x_1, ..., x_i, ..., x_n) = x_i \cdot f(x_1, ...x_i = 1, ...x_n) + (\bar{x_i}) \cdot f(x_1, ..., x_i = 0, ..., x_n) = x_i \cdot CF_1 + (\bar{x_i}) \cdot CF_2$, where x_i is the control variable and CF is the cofactor.

More partitions are conducted on the cofactors which are associated with longer paths, and the probability of cofactor activations decreases as well. This process is repeated until the given constraints on area and delay are met. Next, gate sizing is applied to further isolate critical paths. Then, to reduce the power dissipation, supply voltage is scaled down until non-critical path delays are extended to the target clock period. Finally, to detect the excitations of critical path delays, inputs of logic networks need to be monitored by decoding logic circuits designed based on control variables. Figure 1.4 [17] shows an example of a CRISTA-based pipeline structure. As illustrated, clock gating is used to switch the circuit from the normal mode to the two-cycle mode.

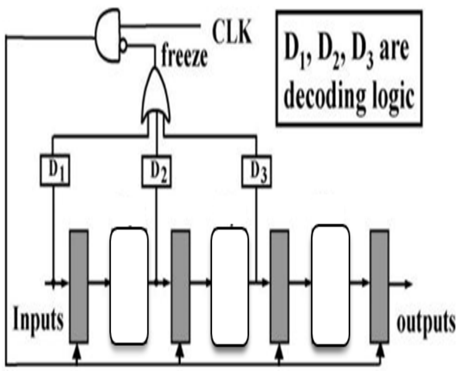


Figure 1.4: Block diagram for a CRISTA-based pipeline design [17].

One major drawback of this technique is the area penalty, and two sources account for it. First, extra circuits are needed for decoding inputs to monitor the critical path excitations. Second, Shannon-expansion-based partitioning involves logic expansions, which may convert compact logic expressions into more complicated forms.

1.2.3 Razor

Razor [19] is a low-power technique which can enable circuits to operate at lower supply voltage through detecting and correcting timing violations at the circuit-level. It was originally developed as a new dynamic voltage scaling (DVS) technique and was reported to realize more than 40% power savings compared with the traditional DVS design [19].

Like CRISTA, Razor exploits the fact that critical paths are rarely excited and overly conservative voltage margins are usually set to account for worst-case operation conditions. Razor reduces the power consumption because it allows for low supply voltage operation by detecting and correcting errors associated with timing failures. Lowering the supply voltage to reduce safety margins is effective in realizing power savings, since dynamic power consumption has an approximately quadratic relationship with the supply voltage. But when some paths with large delays are activated under a low supply voltage, circuits may make errors because of timing violations. Razor manages to capture erroneous outputs and replace them with correct values at the cost of extra hardware and more energy consumptions. Thus, the actual power reduction is the difference between the power saving from lowering the supply voltage and the power penalty from detecting and correcting errors.

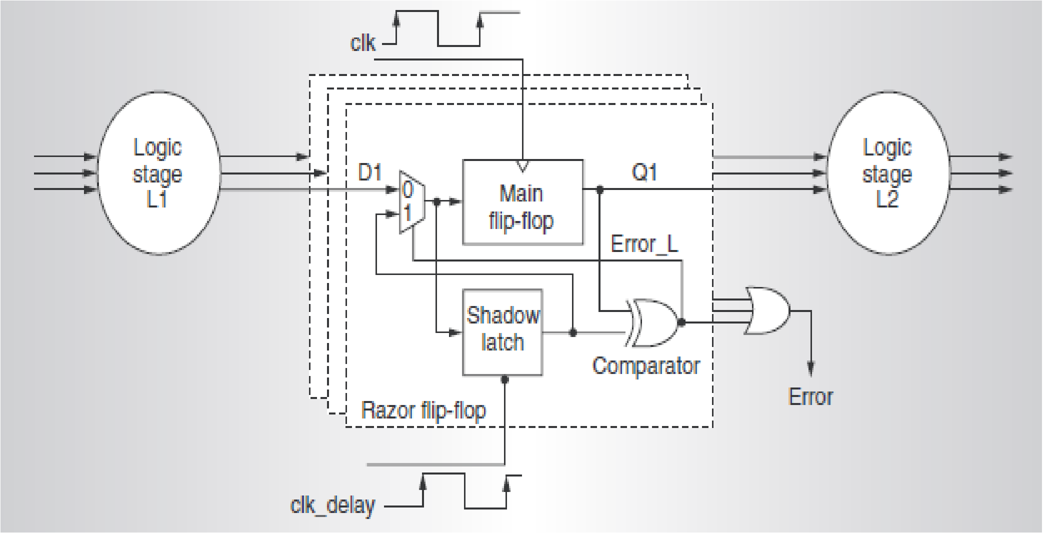


Figure 1.5: Block diagram for Razor control line [19].

Razor relies on efficient architecture-level and circuit-level error detections and corrections. Figure 1.5 [19] shows the block diagram of the control lines of Razor. In a conventional design, one flip-flop per bit is inserted between two combinational logic networks; however, a shadow latch, a comparator, and an error-controlled multiplexer are also needed for each bit in the control lines of Razor. The control line performs two tasks. The first task is to detect errors by sampling the output at a rate slower than the clock frequency, and the second is to re-execute the computations with correct values flushed back through shadow latches when errors are captured. The shadow latch, which runs at a delayed clock, serves in an additional monitoring path of timing violations. If, at one of the pipeline stages, the one-bit comparator, which is an XOR gate, detects an inconsistency between the value stored in the main flip-flop and the value in the shadow latch, indicating the output has not been settled at the clock edge, then an error is detected at this stage. The following pipeline stages after the place of the first error detected will re-compute. The inputs of the main flip-flops are selected by multiplexers between values from the previous logic stage and the values in the shadow latches. When an error is captured, the re-execution will result in a latency penalty of one clock cycle. To ensure successful error detection and correct re-computation, the following two timing constraints must be secured:

1. Slow path constraint: $T_{p,max} < T_{clk} + T_{cd}$.
2. Fast path constraint: $T_{p,min} > T_{cd} + T_{hold}$.

T_{clk} is the clock period, T_{cd} is the delay between the main clock and the clock of shadow latches, T_{hold} is the hold-up time of shadow latches and $T_{p,min}$, $T_{p,max}$ are the minimum and maximum delays of the previous logic stages respectively. When the slow path constraint is not satisfied, the delay is so large that the output cannot be settled to the correct values. Then the shadow latch will store an erroneous value as well as the main flip-flop, so re-computations with correct values are not possible. When the fast path constraint is violated, the previous logic stage operates too fast, and the output of the next clock cycle is produced before the main flip-flop manages to hold the current output. Then the comparator compares the output value of the current cycle in the shadow latch against the value of the next cycle in the main flip-flop, and a false-positive detection of an error occurs.

Razor-based DVS allows for aggressive supply voltage scaling in order to further improve power savings, because the voltage can be properly adjusted based on the error rate. An example of the relationship between supply voltage and the error rate is shown in the measured data for an 18-bit field-programmable gate array (FPGA) multiplier (Figure 1.6 [19]). Even though it was originally developed for DVS, Razor can be generally applied to other digital systems with non-dynamic voltage scaling. Moreover, the idea of adjusting voltage margins according to the error rate can be easily adapted to compensate for PVT variations.

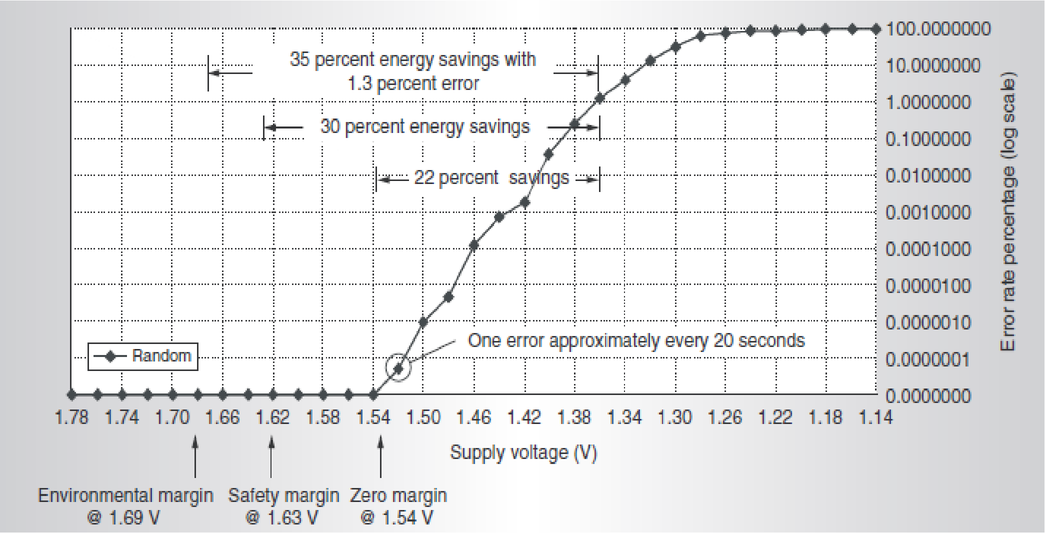


Figure 1.6: Error rate versus supply voltage scaling [19].

The timing constraints create major limitations of using Razor. To fulfill the slow path constraint, the critical delay has to be smaller than the total time between the clock edge of main flip-flops and the sample edge of shadow latches. To secure the fast path constraint, buffers sometimes need to be inserted before the main flip-flop. The difference between the main clock and the clock of the shadow latches is the timing margin of the system. The supply voltage cannot be lowered without limitations, because the timing margin needs to be confined within a certain range; otherwise, penalties from buffer insertions can be overwhelmed. Other drawbacks of Razor are the hardware overhead and the latency penalty.

1.2.4 Body Bias Adjustment

Body bias adjustment is a technique used to compensate for parameter variations by calibrating the transistor source-to-body voltage, V_{SB} [20,21]. It was originally developed to compensate for D2D variations only [20]. In 2002, body bias adjustment was modified to handle WID variations as well [21]. Body bias adjustment was demonstrated to improve the frequency bin splitting, and to increase the die acceptance rate [21]. In [21], the acceptance rate for 62 dies was 50%, with all dies in the lowest frequency bin before the application of body bias adjustment; however 99% of the dies are accepted in the highest frequency bin after body bias adjustment is adaptively applied.

The key idea of body bias adjustment is to use the threshold voltage V_{th} of transistors as a control knob to realize the best trade-off between the frequency and the leakage power. D2D and WID variations, taken together, result in circuit frequency variations and leakage power variations. Figure 1.7 [21] shows the distribution of 62 circuits fabricated in 150 nm CMOS technology. The frequency range was divided into several bins, and dies were placed into the highest possible frequency bin while the power constrains were still met; however, some of the dies could not be accepted because they either were too slow or leaked too much. Adjusting V_{th} is an effective method to improve the acceptance rate of dies under variations for the following two reasons:

1. Lowering the threshold voltages can increase the drain currents of transistors, and thus increase the circuit speed, because the drain current can be

roughly expressed as $I_{on} \propto (V_{GS} - V_{th})^2$ where V_{GS} is the voltage across the gate and the source of a transistor.

2. Increasing the threshold voltage can lower the leakage current and therefore can reduce the standby power consumption, since $I_{leakage} \propto e^{(-V_{th})}$.

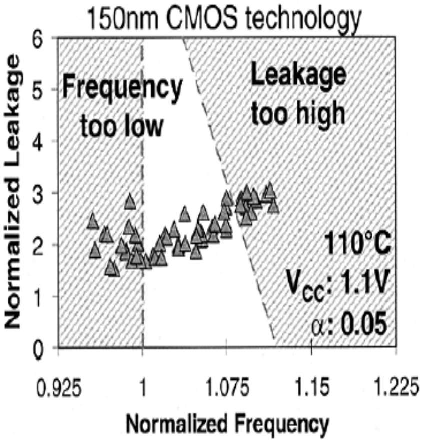


Figure 1.7: Frequency and leakage distributions for 62 test chips [21].

Body effect, which can be exploited to adjust the threshold voltage, is expressed as $\Delta V_{th} \propto \sqrt{(\phi_f + V_{sb})}$, where ϕ_f is a process-dependent physical parameter. Designers can intentionally employ either reverse body bias (raise the substrate voltage of PMOS or lower the substrate voltage of NMOS) or forward body bias (lower the substrate voltage of PMOS or raise the substrate voltage of NMOS) to increase or decrease the threshold voltages accordingly. By setting body bias voltages of different dies, the distribution of frequency and leakage power can be narrowed into the middle range, where the constraints on speed and power can be satisfied simultaneously and the acceptance rate can also be improved.

Extra circuitry is needed to calibrate the body bias level. Figure 1.8 [21] shows the block diagram of a circuit with adaptive body bias. As shown, a circuit block, which copies the critical path of the circuit under test, is built on chip to model the actual circuit operation conditions under variations. The target frequency Φ is applied externally and is compared against the actual frequency of the critical-path-replica by a phase detector. The output of the phase detector is then converted to a 5-bit code, which represents the proper level of body bias voltage. Taking the 5-bit code and a reference voltage as inputs, a digital-to-analog converter, which consists of a resistor network and an operational amplifier, generates the bias voltage. Alternatively, body bias voltage can be set using software control with phase detector outputs instead of using on-chip circuits.

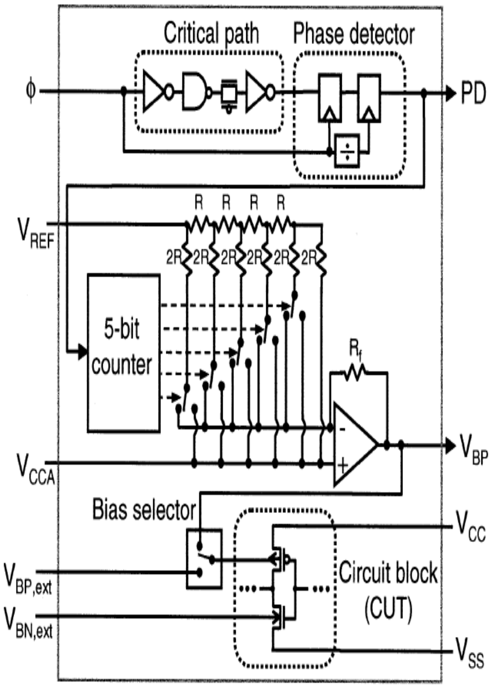


Figure 1.8: Block diagram for adjusting body bias voltage [21].

By applying the same combination of PMOS and NMOS body bias voltages to all the transistors on a die, D2D variations can be compensated for and the acceptance rate can be improved. Yet, this method does not address WID variations. To account for the parameter variations of the transistors on the same die, a method was proposed in [21] to divide the circuit into blocks and apply the adaptive body bias adjustment technique to each of them. Compared with the former method, WID body bias adjustment can further increase the number of dies placed in the highest frequency bin [21].

The disadvantages of using this technique are the extra circuitry and power consumption needed to monitor the critical path delay and to generate the bias voltages. WID body bias adjustment needs more hardware because a number of sub-blocks require more phase detectors and more bias voltage generators. The overheads of hardware and power depend on variances of process parameters and resolutions of bias voltages.

1.2.5 Approximate Computation

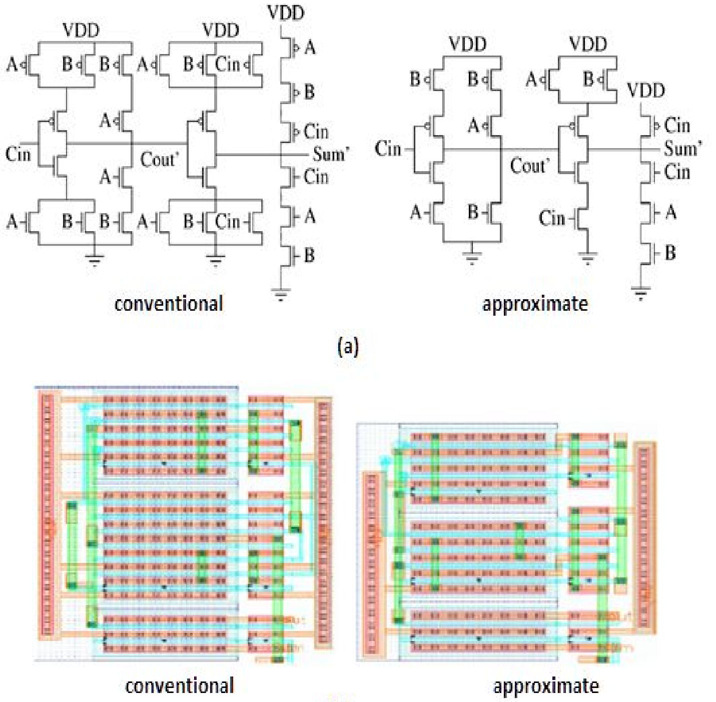
Approximate computation is a recently developed technique for digital signal processing (DSP) applications, and it trades computation accuracy for energy through transistor-level complexity reductions [22]. Gupta et al. [22] demonstrated the benefits of approximate computation by implementing DSP blocks with approximated versions of mirror adders. They reported that approximate computation achieved a greater than 60% power reduction based on simulations, compared with conventional implementations with accurate adders.

Approximate computation takes advantage of the fact that for many DSP applications, it is not necessary to produce numerically exact results, because approximations usually have adequate accuracy for human users [22]. In fact, researchers have long been developing techniques for low-power circuits through simplifications at different levels of design abstractions. Simplifications at logic, architecture, and algorithm levels are widely used in applications where accuracy requirements are statistical in nature. A simple example of these simplifications is the word length truncation, which is commonly applied in many digital systems. The project reported in [22] was the first to reduce the design complexity at the transistor-level and to build computational systems with erroneous arithmetic units.

Figure 1.9 [22] shows the schematics, layouts and truth tables of an accurate mirror adder and of an approximate adders used in [22]. As shown in Figure 1.9, the approximate adder is created by removing some transistors from the original adder; and as a result, the total capacitance in the circuit is reduced. There are two benefits of using approximate adders. First, the dynamic power is reduced since the power is mostly consumed by charging and discharging the capacitances. Second, the critical path delay is reduced, which enables lower supply voltage to further reduce the dynamic power. Most of the DSP circuits are built using adders and multipliers, so substantial power reductions are feasible by using approximate adders. To meet application-level specifications, the performance degeneration induced from logic-level errors needs to be controlled to some extent, which prevents replacing all of the conventional adders in a circuit with approximate adders. Thus, the actual power savings depends on the number of least significant bits (LSBs) that can be approximated, which is determined by the accuracy requirements. Approximate computation takes advantage of the inherent system robustness, but does not improve it.

An error model of the approximate adder was developed in [22]. Error statistics of approximate adders and of data truncation were examined in [22]. In Figure 1.10 [22], the y-axis shows the error mean and variance of a ripple carry adder, with some LSBs replaced with different versions of approximate adders, and the x-axis shows the number of approximated bits or the number of truncated bits. Computations with data truncations discard all the information in several LSBs; however, approximate computation produces hardware errors at LSBs with an error distribution. Statistically, approximate adders should outperform data truncation if the same number of LSBs are discarded or approximated.

Approximate computation and its various modifications have been widely stud-



Inputs			Accurate Adder		Approximate Adder	
A	B	C_{in}	Sum	C_{out}	Sum	C_{out}
0	0	0	0	0	1	0
0	0	1	1	0	1	0
0	1	0	1	0	1	1
0	1	1	0	1	0	1
1	0	0	1	0	1	0
1	0	1	0	1	0	1
1	1	0	0	1	0	1
1	1	1	1	1	0	1

(c)

Figure 1.9: Approximate computation: (a) Circuit schematic, (b) layout, and (c) truth table for both conventional and approximate designs [22].

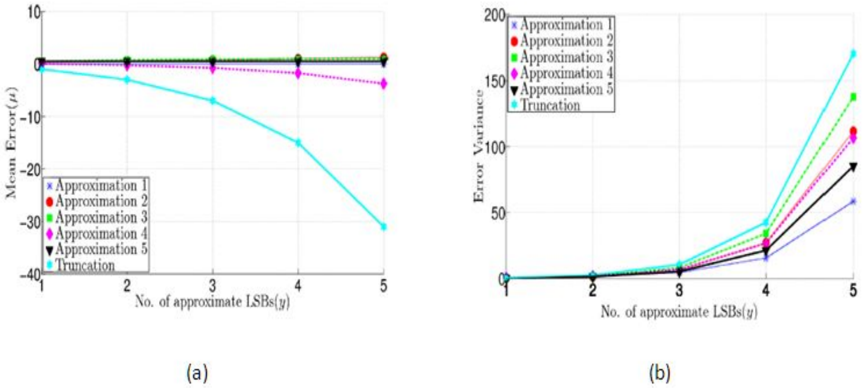


Figure 1.10: Error statistics for approximate adders: (a) mean and (b) variance [22].

ied. Reconfigurable versions of approximate adders that can adapt to variable accuracy requirements were built [23, 24]; custom synthesis methods for approximate circuits were developed [25–27]; and analytic models were created which can be used to realize optimal trade-offs between the output quality and the power savings of approximate circuits [27, 28].

The major limitations of approximate computation are noticeable. It is not applicable to general purpose computing where imprecise computations are not acceptable. What is also notable is that this method only explores the inherent error-resiliency of the applications, but it does not compensate for errors.

1.2.6 N-Modular Redundancy

N-modular redundancy (NMR) is a traditional fault tolerant technique in which one circuit block is replicated by N times, and a majority voter selects the majority result among the outputs of all the replicas. The underlying principle of NMR is similar to the idea of using repetition codes in communication systems; however, the redundancy being introduced through simple repetition is not efficient in compensating for the unreliability, and the overhead is large. NMR can be applied at different abstraction levels, such as gate-level or system-level. NMR improves the robustness of the hardware implementation at the cost of an N× power overhead and an N× area penalty.

1.2.7 Statistical Error Compensation

SEC [29] is a class of promising robust circuit design techniques. Inspired by communication techniques, SEC treats unreliable substrates as noisy communication links and compensates for hardware errors at the system-level with detection and estimation techniques [29]. The error resiliency of SEC enables normal-designed or even under-designed circuits to operate in a statistically acceptable manner. In SEC-based design, computation efforts can be dramatically lowered, so that variations can be overcomed. SEC is reviewed in detail in Chapter 3.

To demonstrate that SEC has an outstanding tolerance against process variations, a specific SEC technique, known as SSNOC, is leveraged to design a signal-detection kernel built on CNFETs, and the application performance is evaluated in the presence of CNFET variations. Simulation results show that a 90× improvement in circuit yield can be achieved and 90% of power efficiency of ideal CNFETs with no variations can be retained.

The rest of this thesis is organized as follows. Chapter 2 presents CNFET-specific variations and existing CNFET circuit design techniques. Chapter 3 reviews SEC techniques in detail. Chapter 4 describes conventional and SSNOC-based designs of a signal-detection kernel, and presents simulation results. Chapter 5 concludes this thesis.

CHAPTER 2

ROBUSTNESS CHALLENGES FROM CNFET-SPECIFIC VARIATIONS

CNFETs are excellent candidates for implementing energy-efficient next-generation digital systems in the sub-10 nm region, and are projected to provide an order of magnitude improvement in EDP over CMOS technology [30]. Despite the great potential in the energy efficiency of ideal CNFETs, significant process imperfections and CNFET-specific variations have prevented the realization of CNFET-based VLSI circuits. For the past several years, research breakthroughs have been made and many CNFET circuit specific design techniques have been developed [31–33]. However, those techniques, most of which are on the circuit-level, are not enough to fully realize the EDP benefits of ideal CNFETs [34].

2.1 CNFET-Specific Variations

In addition to variations presented in silicon devices, CNFETs are also subject to CNFET-specific variations (Figure 2.1), including variations in carbon nanotube (CNT) type, diameter, density, alignment, and doping [31]. Single-CNT CNFETs present significant variations [35], and they are not able to provide sufficient drain currents for practical VLSI applications. In contrast, CNFETs with multiple CNTs have large on-current I_{on}, and the impact of variations is mitigated because of statistically averaging effect [36]. Thus, multiple-CNT CNFETs are used in building VLSI circuits.

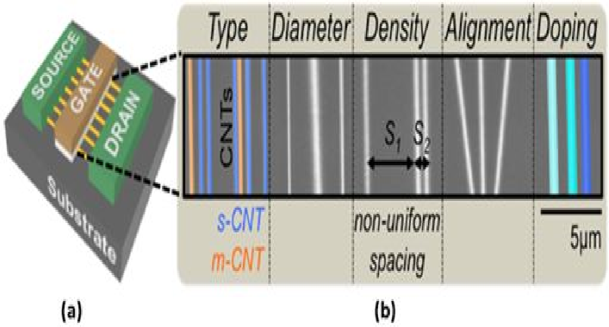


Figure 2.1: The CNFET: (a) CNFET and (b) scanning electron microscopy (SEM) visualization of CNT variations in the CNFET channel [30].

From [31], I_{on} variations are dominated by CNT count variations. CNT count is defined as the number of CNTs left in a CNFET after removing metallic type CNTs, referred to as m-CNTs, using a technique called VLSI-compatible metallic carbon nanotube removal [33]. There are multiple sources of CNT count variations, and the most significant two sources are CNT density variations and m-CNT-induced variations. Figure 2.2 illustrates the quantified impact of each source of CNFET variations on I_{on} variations at the 5 nm node [34] (other details of processing can be found in [34]).

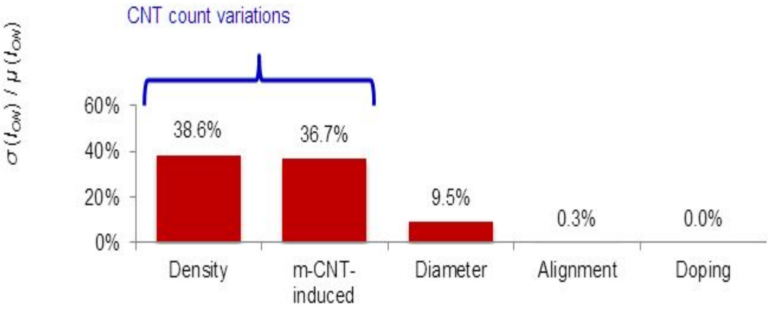


Figure 2.2: I_{on} variations as a result of CNFET variations (details of processing parameters in [34]).

To parametrize CNT count variations, Hills et al. [34] used the parameters defined below, referred to as the processing parameters:

- IDC (index of dispersion for CNT count [37]): The square of the ratio between the mean and variance of the grown-CNT spacing distribution.
- p_m: Probability that a given CNT is an m-CNT.
- p_{rs}: Conditional probability that a CNT is removed, given that it is a semiconductor type CNT (s-CNT).
- p_{Rm}: Conditional probability that a CNT is removed, given that it is an m-CNT.

2.2 Circuit Impact of CNFET Variations

The delay penalty is the metric used in [31] to quantify the impact of CNFET variations on CNFET circuit delays, and is defined as the increase in the 95-percentile delay (T_{95}: the clock period at which the probability of error-free computation reaches 95%) relative to the nominal delay (T_{Nom}: the critical circuit delay when no variations are present) [37]. In [34], a Monte-Carlo statistical static timing analysis (MC SSTA), which is based on a variation-aware gate timing model, is employed to compute the delay penalty.

In addition to the delay variations, CNFET circuits also exhibit logic-level failures. The probability of not having any working CNTs in a given CNFET is greater than zero because of CNT count variations, leading to opens or shorts in CNFET circuits [37]. The probability of count failure, p_{cf}, is referred to as the probability that at least one CNFET experiences this type of functional failure in a given circuit [37]. In [37], the most effective method to minimize p_{cf} is to set a pre-defined minimal width, W_{MIN}, and all CNFETs smaller than W_{MIN} will be upsized to it. This type of upsizing, referred to as min-width upsizing [37], reduces p_{cf} with the penalties of increased power and area.

Upsizing CNFETs and reducing the frequency both can help to compensate for variations; but, as discussed in Section 1.2.1, they are still based on conventional conservative design principles, and they sacrifice the potential benefits from improved electrical properties of CNTs. Some other techniques developed for CNFET variations are also described in [31]. Layout design optimization exploits the special correlation property of CNT placements, in order to lower the variance of logic gate delay distributions and to reduce delay penalties. Selective transistor upsizing optimizes CNFET sizing, with the aid of logic effort analysis, to reduce energy overhead from evenly upsizing all CNFETs.

Even if all the techniques previously mentioned are applied together, CNFET circuits would not present their full EDP potential. According to a recent project reported in [31], in order to realize CNFET circuit energy-efficiency as predicted, applying layout optimization and selective upsizing would not be enough; however, a set of advanced processing parameters would also be required [31]. But, those processing parameters have not been experimentally demonstrated [31]. With current processing techniques and existing circuit-level design techniques, CNFET digital circuits cannot be energy efficient and reliable simultaneously.

CHAPTER 3

A REVIEW OF STATISTICAL ERROR COMPENSATION TECHNIQUES

SEC is a class of techniques used to design energy efficient and robust systems on nanoscale processing technologies. SEC is based on the communication-inspired idea of treating unreliable substrates as noisy communication links, and it corrects or compensates for errors at the system-level or algorithmic-level by applying detection and estimation techniques [29]. The error resiliency of SEC enables unreliable circuits to recover from the performance degeneration due to variations, by allowing nominal-case or average-case designs instead of worst-case conservative designs. Figure 3.1 shows a general setting of SEC-based designs, where the input x is fed into an unreliable computational kernel and several observations of the correct output are processed in an estimator or a detector to produce the final output. The goal of the estimator/detector is to recover the correct computational results from errors and to minimize the difference between the corrected output and the desired outputs. SEC realizes power savings while meeting statistical system-level or application-level requirements, such as signal to noise ratio (SNR), bit error rate (BER), and probability of correct decisions, etc. SEC has been demonstrated to tolerate errors and to achieve significant power savings in CMOS designs. In this chapter, some well-developed SEC techniques are presented, followed by a brief comparison between SEC and before-mentioned robust circuit design techniques.

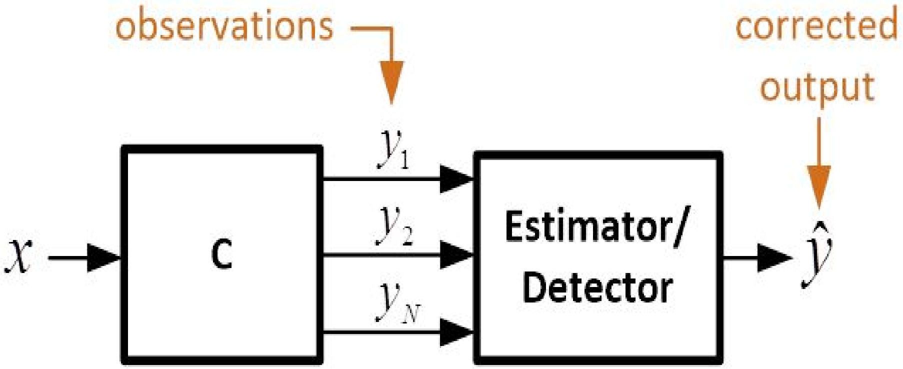


Figure 3.1: General setting for SEC.

3.1 Algorithmic Noise-Tolerance

Algorithmic noise-tolerance (ANT) [38] is the first developed SEC technique. As shown in Figure 3.2(a) [29], an ANT-based design incorporates the main block with an estimator. The estimator is the low-complexity, usually reduced-precision, version of the main block. The main block is operated under relaxed design constraints, such as overly scaled voltages, and it exhibits hardware errors from worst-case operation scenarios; the estimator, on the other hand, performs exact computations, but it produces estimation errors. The outputs of the main block and the estimator can be expressed as:

$$y_a = y_o + \eta,$$
$$y_e = y_o + e,$$

where y_a is the output of the main block, y_e the output of the estimator, y_o is the correct output, and η and e are hardware errors and estimation errors respectively, as shown in Figure 3.2(b) [29]. ANT design methodology leverages the fact that the statistical behaviors of η and e are distinctly different, and ANT based designs incorporate a simple but efficient detection mechanism to select between main block output and the estimator output to be the final output. ANT ensures that the optimal quality output will be selected every time, thus it maintains high application performance while relaxing design constraints. An ANT-based finite impulse response (FIR) filter has been reported to achieve a 3× improvement in power savings [39], and an ANT-based Viterbi decoder has been presented to have an 800× improvement in BER and a 3× improvement in energy savings [39]. ANT prefers LSB-first computations, because this type of computations tend to generate large magnitude hardware errors, and such errors are easily separated from estimation errors, as shown in Figure 3.2(b). The drawback of ANT is the area overhead from the estimation block and the decision circuit, which account for approximately 20% of the main block [29].

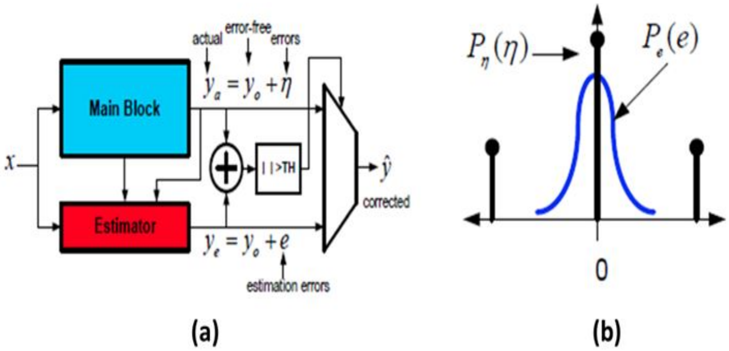


Figure 3.2: Algorithmic noise-tolerance: (a) block diagram for ANT-based design, and (b) desired error distributions for ANT [29].

3.2 Stochastic Sensor Network on Chip

SSNOC takes multiple erroneous outputs as observations or estimations of the desired output and fuses them to generate a corrected output, as illustrated in Figure 3.3 [29]. In an SSNOC-based design, a main computation block is decomposed into reduced complexity sub-blocks, referred to as sensors, in a statistically similar manner. If the sensor output is expressed as $y_{ei} = y_o + e_i + \eta_i$, then the statistical similarity ensures that $E[y_{ei}] = E[y_o]$. The errors of the sensors are from two sources: η_i is the hardware error resulting from intentional under-design and e_i is the estimation error resulting from the decomposition. Hardware error η_i yields to an irregular distribution, while estimation error e_i yields to a Gaussian distribution. Together, they can be modeled as an ϵ-contaminated Gaussian distribution, assuming the errors are independent. An ϵ-contaminated Gaussian distribution is a distribution with a probability of $1-\epsilon$ being a Gaussian, and with a probability of ϵ being an unknown distribution. The sensor outputs with such distributions can be fused to produce worst-case optimal estimation of the correct output, according to the theory of robust statistics developed by Huber [40].

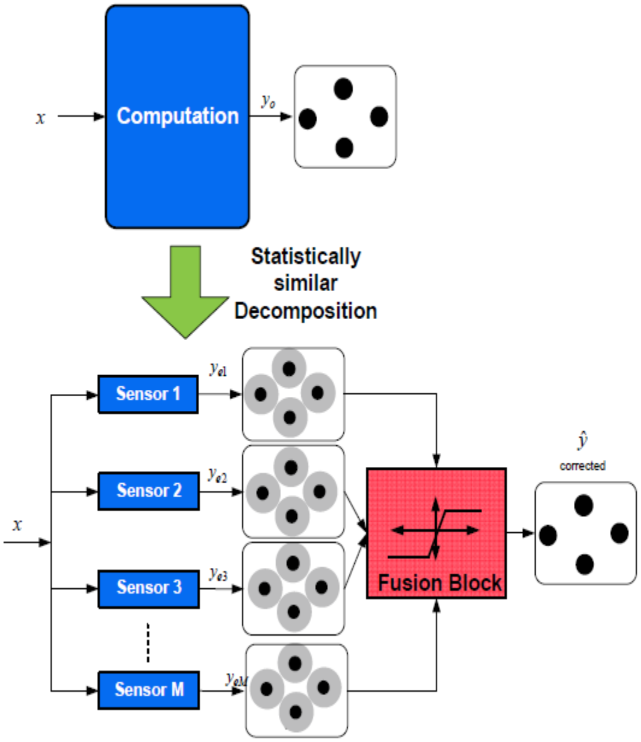


Figure 3.3: SSNOC-based design [29].

In a previously published project, two fusion algorithms were examined [41]. The first, a simplified version of Huber's method [41], is named the one-step Huber algorithm, and it can offer great robustness improvement [41]. The second algorithm takes the median of all the sensor outputs (which is the first step in the one-step Huber algorithm). The latter has been demonstrated to provide acceptable robustness as a fusion method; however, the median filter used in the latter needs much simpler hardware compared to the fusion block used in the one-step Huber's algorithm [41, 42]. An SSNOC-based pseudo-random noise code (PN-code) acquisition filter was built using a 180 nm CMOS process [42]. Multiple dies were tested, and the energy savings of the proposed design ranged from 2.4× up to 5.8× (3.86× on average) compared with the conventional design [42]. One drawback of SSNOC is the area penalty from using fusion block which is not needed in conventional designs; however, in some special implementations, the area penalty can be negligible if the fusion block is simple enough (explained further in Section 4.1). The major limitation of SSNOC is that not all computations can be easily decomposed in a statistically similar manner.

3.3 Soft N-Modular Redundancy and Likelihood Processing

Soft n-modular redundancy (soft NMR), Figure 3.4(a), and likelihood processing (LP), Figure 3.4(b), are two other SEC techniques. Traditional fault tolerant technique NMR forcibly uses hardware redundancy for error compensations; but soft NMR leverages the statistical error behavior into conventional designs, exploiting the redundancy in a statistically efficient manner at the word level [43]. In a discrete cosine transform (DCT) image coder design, soft triple-MR (TMR) provided a 10× improvement in robustness and 13% power savings and soft dual-MR (DMR) provided a 2× improvement in robustness and 35% power savings, compared with regular TMR [43]. LP also exploits the hardware redundancy for error compensations, and it utilizes error statistics at the bit level instead of at the word level [44]. LP uses the bit-level confidence, or likelihood, to generate the corrected output. For a 2D DCT coder design, LP improves error tolerance by 100× compared with conventional designs, and by 5× compared with regular TMR [44].

Previously, SEC was leveraged to compensate for errors resulting from voltage over scaling (VOS) for ultra low-power circuits. In an SEC-based design, erroneous outputs of a computational kernel are treated as the observations of a

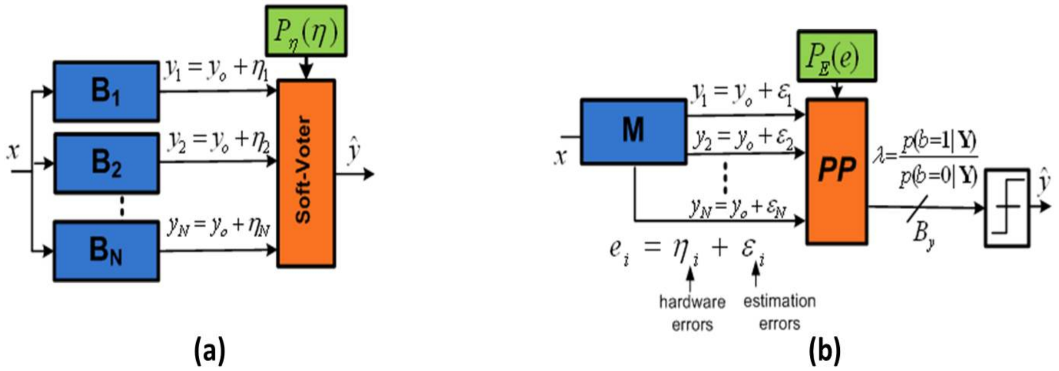


Figure 3.4: Block diagrams for (a) soft NMR and (b) likelihood processing [29].

stochastic process, and certain knowledge of the error behavior is required to apply error compensation techniques. But the error sources are not limited to VOS or device variations. Digital systems designed with SEC have been implemented on inference-based applications where accuracy requirements are statistical in nature. For applications that require precise data processing, SEC cannot be applied in its current form [29]. Area penalties are common in SEC techniques, because extra circuitries are needed either to generate multiple observations of desired outputs or to make corrections on these observations.

3.4 A Comparison of SEC and Other Robust Digital VLSI Design Techniques

SEC techniques operate at the system-level to compensate for errors, which distinguish them from the techniques described in Chapter 2. Critical path isolation [17] and Razor [19] are architecture-level techniques that can prevent hardware errors resulting from timing violations to reduce safety margins; body bias adjustment [21] reduces the variances of the frequency and the leakage power by exploiting the second-order effect of transistors; approximate computation [22] trades output qualities for the power consumption with the help of transistor-level complexity reductions. In terms of error handling, critical path isolation and body bias adjustment explore the statistical nature of PVT variations and prevent errors from occurring; Razor allows the occurrence of errors and applies additional error detection and correction mechanisms to ensure exact computation; approximate computation takes advantage of inherent error resiliencies of applications and treat the error rate as a measure of the output quality. These techniques do not intentionally compensate for errors. In contrast, SEC exploits the inherent robustness of applications in a manner similar to approximate computation, but it also elegantly compensates for errors in a stochastic manner.

CHAPTER 4

CIRCUIT DESIGN AND SIMULATION RESULTS

To demonstrate that SEC can carry forward the recovery from imperfections and variations of CNFET circuits, an SSNOC-based computational kernel is implemented and its performance is examined. It is important to note that the fast modeling method developed in [34] for CNFET circuits delay variations is the key enabler for the methodology used in simulations. In this chapter, the design of an SSNOC-based signal detection circuit is presented, the simulation flow adapted to generate the results is described, and simulation results are shown.

4.1 SSNOC-Based Signal Detection Kernel

To illustrate the error-resilience of SEC, a computational kernel is implemented to detect a signal of interest within a noise-contaminated input. The circuit output is a 1-bit binary decision corresponding to whether or not the signal of interest is detected. At the heart of this detection is a correlation kernel that calculates the inner product of two vectors. The inner product is then compared against a threshold to determine the decision bit. The correlation kernel, shown in Figure 4.1, is commonly used in a variety of applications, including classifications, pattern recognitions, multi-media signal processings, and communication receivers [41]. In this project, the kernel is used in a PN-code acquisition system for wireless communications, as previously mentioned. As shown in Figure 4.1, H is the locally stored PN-code, X is the noisy input signal from a communication channel, Y is the inner product of H and X, and T is the threshold. Threshold T is set at the level that keeps the probability of false alarm at 5% [41], though it can be set according to other application-level requirements.

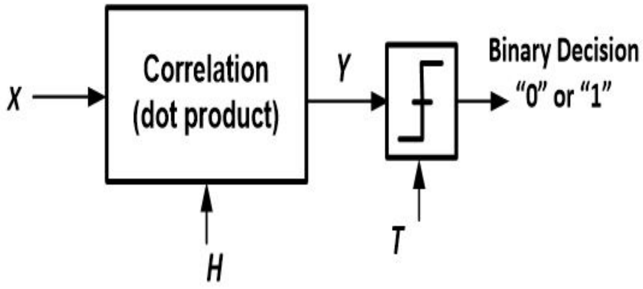


Figure 4.1: Signal detection kernel. Vector H is the signal of interest, vector X is the noise-contaminated input, Y is the correlation between X and H, and T is the threshold for decision-making.

The inner product is commonly implemented using an N-tap FIR filter, shown in Figure 4.2(a), with the input-output relationship defined as:

$$y[n] = \textstyle\sum_{j=0}^{N-1} h[j]x[n-j].$$

In the above equation, $h[j]$ is the filter coefficient and $x[n]$ is the noisy input with 8-bit precision. By using the delay modeling method developed in [34], it is found that to ensure almost error-free computation ($p_{cf} < 1\%$) for a conventional CNFET-based implementation, the circuit has to suffer from unacceptable penalties in both frequency and power. The circuit EDP would be reduced significantly.

To retrieve ideal CNFET benefits, design parameters need to be relaxed. Reducing the clock period or decreasing W_{MIN} can help to fully realize the CNFET EDP improvement, but errors would be produced and the application-level performance would be degraded. Even though the inherent robustness would help to maintain an acceptable signal detection rate for slightly smaller transistors and slightly increased frequency when the error rate is relatively low, the conventional design would still fail when T_{CLK} is 5% above T_{NOM} (5% delay penalty) and W_{MIN} is set to achieve a 5% energy penalty, given state-of-the-art CNT processing parameter values.

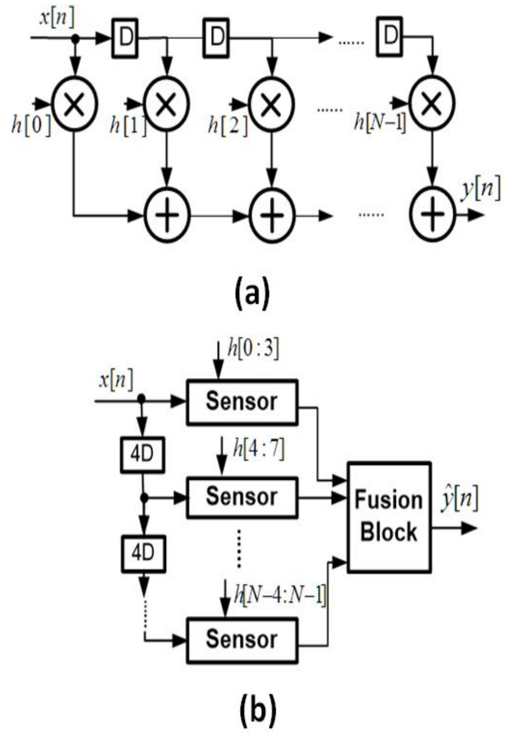


Figure 4.2: (a) Conventional design of N bit FIR filter; (b) the SSNOC-based design of N bit FIR filter.

The SSNOC-based design decomposes the N-tap FIR filter into M identical sub-blocks called sensors ($M < N$), and then fuses the estimated outputs to produce a corrected output (Figure 4.2(b)). In this project, $N = 256$, $M = 64$ and each sensor is a 4-tap FIR filter. The output of sensor i is expressed as:

$$y_i[n] = \sum_{j=0}^{3} h[4i+j]x[n-4i-j].$$

The fusion output $\hat{y}[n]$ (Figure 4.2(b)) is used for decision making. As previously discussed, the one-step Huber algorithm is more robust; however, a median-based fusion algorithm is implemented, because it performs reasonably well and requires considerably less circuitry [41].

Figure 4.3 depicts the block diagrams of the conventional and SSNOC-based designs. The conventional design sums the outputs of all sensors with an adder tree and compares this sum with the threshold T to produce the decision bit. The SSNOC-based design first subtracts T from each sensor output. The sign bit of the subtraction is defined to be 1 if the result is positive, and 0 if the result is zero or negative. The decision bit is then produced as the result of a majority vote among the sign bits of all subtractions. Note that this is equivalent to taking the median of all sensor outputs and then comparing it with T with less circuitry. To prove the two methods are equivalent, suppose that the sensor outputs $y_1, y_2, ..., y_n$ are ranked in an ascend order as $y_{r,1}, y_{r,2}, ..., y_{r,n}$, and the median output is y_{median}. When n is an even number, $y_{median} = \frac{1}{2} \times (y_{r,\frac{n}{2}} + y_{r,\frac{n}{2}+1})$; however when n is an odd number, $y_{median} = y_{r,\frac{n+1}{2}}$. If $y_{median} > T$, then at least n or $\frac{n+1}{2}$ outputs are also greater than T, which means that $sign[median(y_1, y_2, ..., y_n) - T] = majority[sign(y_1 - T), sign(y_2 - T), ..., sign(y_n - T)] = 1$. The case of $y_{median} <= T$ can be proved by following the same procedure. Each design is pipelined so that the critical path delay of the system is equal to the maximum sensor delay (i.e., the critical path delay is not in the adder tree or fusion block). The 64 sensors are common to each design; the adder tree and the fusion block are referred to as the post-processing blocks.

Table 4.1 provides the synthesis results of both designs in the nominal case. SSNOC requires less combinational logic, as well as fewer registers, so it is expected to offer energy savings over the conventional design. It is important to note that the simplicity of the joint design of the fusion block (a digital median filter) and the threshold comparing block provides the hardware efficiency. For other applications and for other designs of the fusion block, the SSNOC-based design does not guarantee less circuitry than the conventional design in general.

Table 4.1: Conventional vs. SSNOC

	Conventional Adder Tree	SSNOC Fusion Block
Combinational Cells	11170 (total),4943 (INV / BUF)	6364 (total),1858 (INV / BUF)
Registers	1390	1011

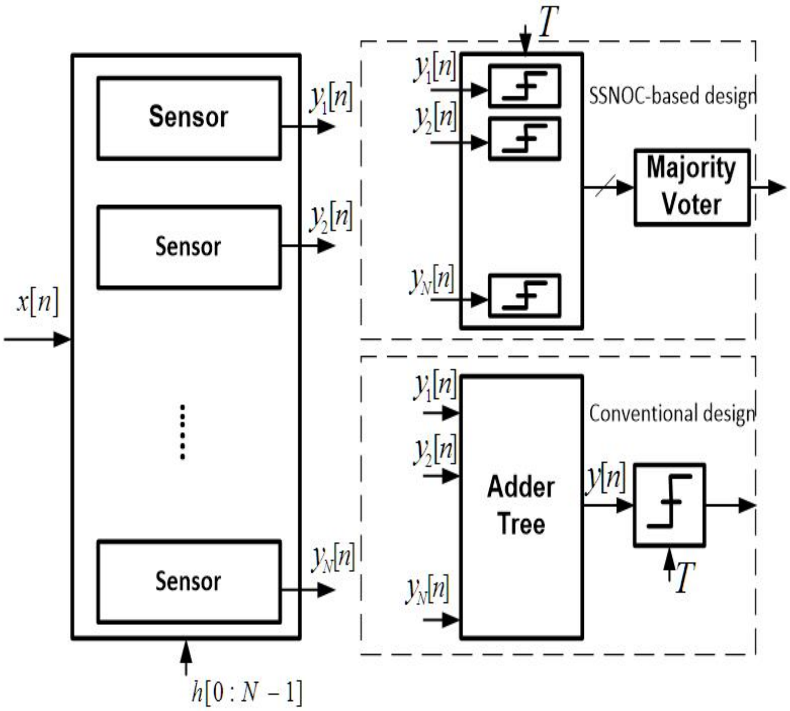


Figure 4.3: Block diagrams of the SSNOC-based design (top-right) and conventional design (bottom-right).

In this project, it is assumed that both post-processing blocks are error-free. To ensure robustness against functional failures, minimal width upsizing is applied to both post-processing blocks.

4.2 Simulation Setup

The following three-step approach is used to simulate the performance of the conventional and SSNOC-based designs in the presence of CNFET variations (illustrated in Figure 4.4):

1. CNFET circuit delay modeling: The MC SSTA methodology [34] (conducted by Gage Hills from the Stanford Robust System Group) is used to generate 100,000 delay samples for each logic gate, using the synthesized netlist and placement information to model correlations among logic gate

delays. CNFETs that do not contain any s-CNTs for a given trial are modeled as an open- or short-circuit.

2. Hardware description language (HDL) simulation: Delay samples and functional failures (from Step 1) are injected into an HDL model and simulated to produce (possibly erroneous) outputs of all the sensors.

3. Probability of detection calculation: For each trial, the sensors' outputs are used to compute decision sequences for both designs, which are error-free. Finally, both binary decision sequences are compared against correct detections to calculate detection probability, as probability of false alarm is fixed at 5%.

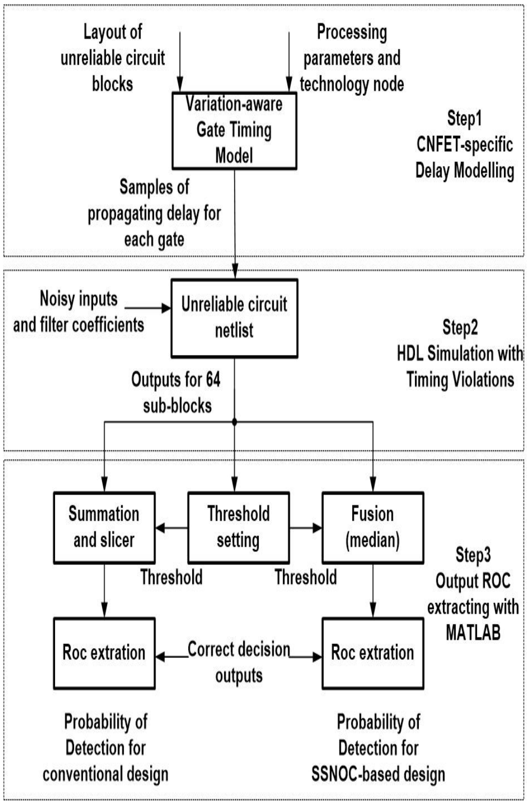

Layout of unreliable circuit blocks
Processing parameters and technology node
Variation-aware Gate Timing Model
Step1 CNFET-specific Delay Modelling
Samples of propagating delay for each gate
Noisy inputs and filter coefficients
Unreliable circuit netlist
Step2 HDL Simulation with Timing Violations
Outputs for 64 sub-blocks
Summation and slicer
Threshold setting
Fusion (median)
Threshold
Threshold
Step3 Output ROC extracting with MATLAB
Roc extration
Correct decision outputs
Roc extration
Probability of Detection for conventional design
Probability of Detection for SSNOC-based design

Figure 4.4: Three-step statistical simulation flow.

For this simulation methodology, each instance of the system would exhibit a different circuit delay distribution and different error statistics, under the exact same set of processing parameters and operation conditions. This methodology examines the error resiliency under the effects of CNFETs D2D and WID variations, which were not the main focus of previous SEC-related projects. The results (detailed in Section 4.3) strongly indicate the power of SEC to compensate for parameter variations.

4.3 Simulation Results

Circuit yield is used as a system-level metric to compare the performances of the conventional design with the SSNOC-based design in the presence of CNTFET variations. Results are generated with different sets of processing parameters and multiple combinations of operating conditions.

Before analyzing SEC resilience of variations, the performances of the conventional design and the SSNOC-based design under variation-free conditions are compared first. The probability of detection of the conventional design reaches 95.1%; and in the SSNOC-based design, the probability of detection decreases to 92.5%. The false alarm rate is fixed at 5% by adjusting the threshold in order to conduct a fair comparison. The conventional design outperforms the SSNOC-based design because the sensors have estimation errors and simply taking the median of estimations is not the optimal detection strategy under such a condition [40]. On the other hand, the detection rule in the conventional design is based on the Euclidean distance, which is the optimal solution for Gaussian noise contaminated signals when there are no computation errors.

Simulations were conducted using the methodology described in Section 4.2. CNFET delay distributions were generated for a set of processing parameters (Table 4.2) and for a range of W_{MIN} (from 10 nm to 50 nm). Circuit delays were then sampled and these samples were used in HDL simulations. Simulations were conducted at three different clock periods, which were set at 5%, 10% and 20%

above nominal (variation-free) critical delay.

Table 4.2: CNT Processing Parameters Used in Simulations

Parameter Set	1	2	3	4	5
IDC	0.500	0.409	0.309	0.207	0.106
p_m	10.0%	9.3%	8.2%	6.5%	4.2%
p_{Rs}	5.0%	4.8%	4.5%	4.0%	3.0%

To quantify the robustness of the designs under CNFET variations, circuit yield is defined as a function of a desired probability of detection, p_d. In the experiments, the yield is estimated as the percentage of emulated systems whose performance meets the p_d requirement. It is expressed as:

$$yield = \frac{1}{n}\sum_{i=1}^{n} I\{p_{d,i} \geq p_d\},$$

where $I\{\}$ is the indicator function that returns 1 if the argument is true and 0 if false.

Figure 4.5 plots yield vs. p_d as the processing parameters degrade (Figure 4.5(a)) and as the transistor minimal size W_{MIN} reduces (Figure 4.5(b)), with the clock being set at 5% above the nominal critical delay for both the conventional and SSNOC-based designs.

In Figure 4.5, the conventional design presents higher yield for a given p_d, when the processing parameters are good (IDC below 0.02) and the transistors are still wide (W_{MIN} above 30 nm). However, as processing conditions are worsened and transistor sizes are reduced, the yield of conventional design drops dramatically and fast. In contrast, the SSNOC-based design is significantly more robust against variations and transistor downsizing. In extreme cases, a 90× improvement in yield for a given probability of detection and a 2.2× improvement in the probability of detection for a given yield were observed. The plots indicate that

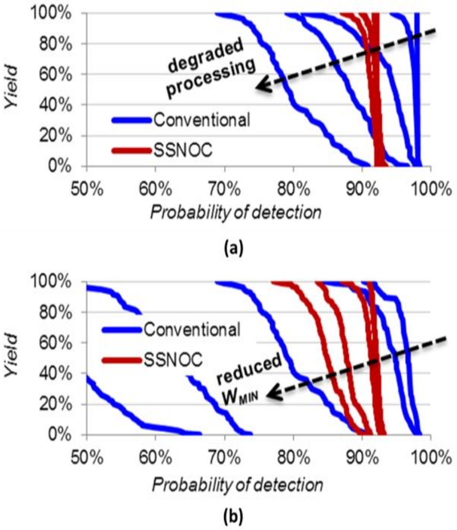


Figure 4.5: (a) Yield vs. p_d, for a range of processing parameters with $W_{MIN} =$ 20 nm. (b) Yield vs. p_d, for a range of W_{MIN} (10 nm to 40 nm) with $IDC = 0.5$.

the SSNOC-based design not only relaxes design margins of the frequency and the transistor size, but also reduces the EDP overhead. The results shown verify SSNOC's superior robustness over the the conventional design.

To visualize that the SSNOC-based design relaxes CNFET processing requirements, Figure 4.6 plots the parameters (IDC, p_m, p_{Rs}) needed for both designs to exhibit 90% yield of 90% detection probability at 5 nm node (with 5% above nominal critical delay and 5% transistor minimal upsizing).

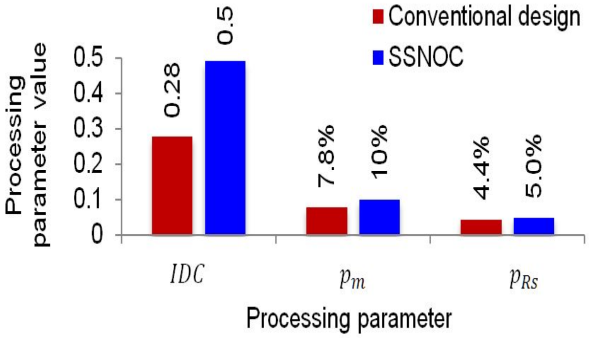


Figure 4.6: Processing parameters need to meet 5% delay penalty with 5% energy increase at the 5 nm node, with 90% p_d and 90% yield.

CHAPTER 5

CONCLUSION

Increasingly severe variations in electric devices, as observed both in deeply scaled CMOS as well as in emerging beyond-CMOS technologies, pose substantial challenges in designing high-performance digital VLSI circuits with low power and high reliability. This thesis exploits system-level innovations to overcome nanoscale variations. As indicated in this thesis, the unreliable behavior of basic circuit blocks can be modeled and exploited in a statistical manner to aid system-level design. Further, this thesis shows that SEC techniques can maintain the desired application-level requirements without the conservative over-design. The simulation results verify the benefits to apply SEC as an energy-efficient and robust design technique for advanced nanoscale technologies with significant process variations. Particularly, SEC greatly contributes to the realization of EDP benefits of CNFETs over CMOS.

The results presented clearly demonstrate the robustness improvement of an SSNOC-based design against CNFET-specific variations. For the application selected in this thesis, the SSNOC-based design and the conventional design both can present a higher than 90% probability of detection at a 5% false alarm rate, when there are no variations; but, the SSNOC-based design exhibits a 90× increase in circuit yield at a 90% detection rate under extremely degraded processing with greatly reduced safety margins, compared with the conventional design. The relaxed processing parameters and the reduced design margins can enable the realization of the ideal CNFET energy efficiency in VLSI applications. Thus, the projected EDP benefits of ideal CNFETs can be recovered from CNFET processing variations through SEC.

REFERENCES

[1] M. Miranda, “When every atom counts,” *Spectrum, IEEE*, vol. 49, no. 7, pp. 32–32, July 2012.

[2] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De, “Parameter variations and impact on circuits and microarchitecture,” *Design Automation Conference, 2003. Proceedings*, pp. 338–342, June 2003.

[3] C. Hu, “Future CMOS scaling and reliability,” *Proceedings of the IEEE*, vol. 81, no. 5, pp. 682–689, May 1993.

[4] G. E. Moore, “Cramming more components onto integrated circuits, reprinted from *Electronics*, volume 38, number 8, April 19, 1965, pp. 114 ff.” *Solid-State Circuits Society Newsletter, IEEE*, vol. 11, no. 5, pp. 33–35, Sept 2006.

[5] C. Mack, “Fifty years of Moore’s law,” *Semiconductor Manufacturing, IEEE Transactions on*, vol. 24, no. 2, pp. 202–207, May 2011.

[6] A. Chien and V. Karamcheti, “Moore’s law: The first ending and a new beginning,” *Computer*, vol. 46, no. 12, pp. 48–53, Dec 2013.

[7] J. Esch, “Overview of beyond-CMOS devices and a uniform methodology for their benchmarking,” *Proceedings of the IEEE*, vol. 101, no. 12, pp. 2495–2497, Dec 2013.

[8] A. Franklin, S.-J. Han, G. Tulevski, M. Luisier, C. Breslin, L. Gignac, M. Lundstrom, and W. Haensch, “Sub-10 nm carbon nanotube transistor,” *Electron Devices Meeting (IEDM), 2011 IEEE International*, pp. 23.7.1–23.7.3, Dec 2011.

[9] S. K. Banerjee, L. Register, E. Tutuc, D. Basu, S. Kim, D. Reddy, and A. MacDonald, “Graphene for CMOS and beyond CMOS applications,” *Proceedings of the IEEE*, vol. 98, no. 12, pp. 2032–2046, Dec 2010.

[10] J.-S. Moon, M. Antcliffe, H. C. Seo, S. C. Lin, A. Schmitz, I. Milosavljevic, K. McCalla, D. Wong, D. K. Gaskill, P. Campbell, K. M. Lee, and P. Asbeck, “Graphene transistors for RF applications: Opportunities and challenges,” *Semiconductor Device Research Symposium (ISDRS), 2011 International*, pp. 1–2, Dec 2011.

[11] S. Sugahara and J. Nitta, "Spin-transistor electronics: An overview and outlook," *Proceedings of the IEEE*, vol. 98, no. 12, pp. 2124–2154, Dec 2010.

[12] G. Liang, N. Neophytou, D. Nikonov, and M. Lundstrom, "Performance projections for ballistic graphene nanoribbon field-effect transistors," *Electron Devices, IEEE Transactions on*, vol. 54, no. 4, pp. 677–682, April 2007.

[13] Y.-Y. Chen, A. Sangai, M. Gholipour, and D. Chen, "Schottky-barrier-type graphene nano-ribbon field-effect transistors: A study on compact modeling, process variation, and circuit performance," *Nanoscale Architectures (NANOARCH), 2013 IEEE/ACM International Symposium on*, pp. 82–88, July 2013.

[14] A. Shahi and P. Zarkesh-Ha, "Prediction of gate delay variation for CNFET under CNT density variation," *Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2012 IEEE International Symposium on*, pp. 140–145, Oct 2012.

[15] Z. Zhang, Y. Liu, J. Nyathi, and J. Delgado-Frias, "Performance of CNFET SRAM cells under diameter variation corners," *Circuits and Systems, 2009. MWSCAS '09. 52nd IEEE International Midwest Symposium on*, pp. 547–550, Aug 2009.

[16] A. Shahi, P. Zarkesh-Ha, and M. Elahi, "Comparison of variations in MOSFET versus CNFET in gigascale integrated systems," *Quality Electronic Design (ISQED), 2012 13th International Symposium on*, pp. 378–383, March 2012.

[17] S. Ghosh, S. Bhunia, and K. Roy, "CRISTA: A new naradigm for low-power, variation-tolerant, and adaptive circuit synthesis using critical path isolation," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 26, no. 11, pp. 1947–1956, Nov 2007.

[18] X. Bai, C. Visweswariah, P. Strenski, and D. Hathaway, "Uncertainty-aware circuit optimization," *Design Automation Conference, 2002. Proceedings. 39th*, pp. 58–63, 2002.

[19] 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," *Micro, IEEE*, vol. 24, no. 6, pp. 10–20, Nov 2004.

[20] M. Miyazaki, G. Ono, T. Hattori, K. Shiozawa, K. Uchiyama, and K. Ishibashi, "A 1000-MIPS/W microprocessor using speed adaptive threshold-voltage CMOS with forward bias," in *Solid-State Circuits Conference, 2000. Digest of Technical Papers. ISSCC. 2000 IEEE International*, Feb 2000, pp. 420–421.

[21] J. Tschanz, J. Kao, S. Narendra, R. Nair, D. Antoniadis, A. Chandrakasan, and V. De, "Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage," *Solid-State Circuits Conference, 2002. Digest of Technical Papers. ISSCC. 2002 IEEE International*, vol. 1, pp. 422–478, Feb 2002.

[22] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power digital signal processing using approximate adders," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 32, no. 1, pp. 124–137, Jan 2013.

[23] R. Ye, T. Wang, F. Yuan, R. Kumar, and Q. Xu, "On reconfiguration-oriented approximate adder design and its application," *Computer-Aided Design (ICCAD), 2013 IEEE/ACM International Conference on*, pp. 48–54, Nov 2013.

[24] A. Kahng and S. Kang, "Accuracy-configurable adder for approximate arithmetic designs," *Design Automation Conference (DAC), 2012 49th ACM/EDAC/IEEE*, pp. 820–825, June 2012.

[25] A. Ranjan, A. Raha, S. Venkataramani, K. Roy, and A. Raghunathan, "ASLAN: Synthesis of approximate sequential circuits," *Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014*, pp. 1–6, March 2014.

[26] S. Venkataramani, A. Sabne, V. Kozhikkottu, K. Roy, and A. Raghunathan, "SALSA: Systematic logic synthesis of approximate circuits," *Design Automation Conference (DAC), 2012 49th ACM/EDAC/IEEE*, pp. 796–801, June 2012.

[27] J. Miao, K. He, A. Gerstlauer, and M. Orshansky, "Modeling and synthesis of quality-energy optimal approximate adders," *Computer-Aided Design (ICCAD), 2012 IEEE/ACM International Conference on*, pp. 728–735, Nov 2012.

[28] C. Liu, J. Han, and F. Lombardi, "An analytical framework for evaluating the error characteristics of approximate adders," *Computers, IEEE Transactions on*, vol. PP, no. 99, p. 1, 2014.

[29] N. Shanbhag, R. Abdallah, R. Kumar, and D. Jones, "Stochastic computation," *Design Automation Conference (DAC), 2010 47th ACM/IEEE*, pp. 859–864, June 2010.

[30] L. Wei, D. Frank, L. Chang, and H. S. P. Wong, "A non-iterative compact model for carbon nanotube FETs incorporating source exhaustion effects," *Electron Devices Meeting (IEDM), 2009 IEEE International*, pp. 1–4, Dec 2009.

[31] J. Zhang, A. Lin, N. Patil, H. Wei, L. Wei, H. S. P. Wong, and S. Mitra, "Carbon nanotube robust digital VLSI," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 31, no. 4, pp. 453–471, April 2012.

[32] N. Patil, A. Lin, E. Myers, K. Ryu, A. Badmaev, C. Zhou, H. S. P. Wong, and S. Mitra, "Wafer-scale growth and transfer of aligned single-walled carbon nanotubes," *Nanotechnology, IEEE Transactions on*, vol. 8, no. 4, pp. 498–504, July 2009.

[33] N. Patil, A. Lin, J. Zhang, H. Wei, K. Anderson, H. S. P. Wong, and S. Mitra, "VMR: VLSI-compatible metallic carbon nanotube removal for imperfection-immune cascaded multi-stage digital logic circuits using carbon nanotube FETs," *Electron Devices Meeting (IEDM), 2009 IEEE International*, pp. 1–4, Dec 2009.

[34] G. Hills, J. Zhang, C. Mackin, M. Shulaker, H. Wei, H.-S. Wong, and S. Mitra, "Rapid exploration of processing and design guidelines to overcome carbon nanotube variations," *Design Automation Conference (DAC), 2013 50th ACM / EDAC / IEEE*, pp. 1–10, May 2013.

[35] B. Paul, S. Fujita, M. Okajima, T. Lee, H. S. P. Wong, and Y. Nishi, "Impact of a process variation on nanowire and nanotube device performance," *Electron Devices, IEEE Transactions on*, vol. 54, no. 9, pp. 2369–2376, Sept 2007.

[36] A. Raychowdhury, V. De, J. Kurtin, S. Borkar, K. Roy, and A. Keshavarzi, "Variation tolerance in a multichannel carbon-nanotube transistor for high-speed digital circuits," *Electron Devices, IEEE Transactions on*, vol. 56, no. 3, pp. 383–392, March 2009.

[37] J. Zhang, S. Bobba, N. Patil, A. Lin, H. S. P. Wong, G. De Micheli, and S. Mitra, "Carbon nanotube correlation: Promising opportunity for CNFET circuit yield enhancement," *Design Automation Conference (DAC), 2010 47th ACM/IEEE*, pp. 889–892, June 2010.

[38] R. Hegde and N. Shanbhag, "Soft digital signal processing," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 9, no. 6, pp. 813–823, Dec 2001.

[39] R. Hegde and N. Shanbhag, "A voltage overscaled low-power digital filter IC," *Solid-State Circuits, IEEE Journal of*, vol. 39, no. 2, pp. 388–391, Feb 2004.

[40] P. Huber, *Robust Statistics.* John Wiley and Sons, 1981.

[41] G. Varatkar, S. Narayanan, N. Shanbhag, and D. Jones, “Stochastic networked computation,” *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 18, no. 10, pp. 1421–1432, Oct 2010.

[42] E. Kim, D. Baker, S. Narayanan, N. Shanbhag, and D. Jones, “A 3.6-mW 50-MHz PN code acquisition filter via statistical error compensation in 180-nm CMOS,” *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. PP, no. 99, p. 1, 2014.

[43] E. Kim, R. Abdallah, and N. Shanbhag, “Soft NMR: Exploiting statistics for energy-efficiency,” *System-on-Chip, 2009. SOC 2009. International Symposium on*, pp. 052–055, Oct 2009.

[44] R. Abdallah and N. Shanbhag, “Robust and energy efficient multimedia systems via likelihood processing,” *Multimedia, IEEE Transactions on*, vol. 15, no. 2, pp. 257–267, Feb 2013.