UNIVERSITY OF SOUTHAMPTON

FACULTY OF ENGINEERING AND PHYSICAL SCIENCES

Electronics and Computer Science

Ultra-Low-Power Sequential Circuit Design for Near-Threshold Voltage System

by

Cai Yunpeng

Thesis for the degree of Doctor of Philosophy

Supervisor: Dr Tom J Kazmierski
Dr Alex Weddell

March 7, 2019

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING AND PHYSICAL SCIENCES
Electronics and Computer Science

Doctor of Philosophy

ULTRA-LOW-POWER SEQUENTIAL CIRCUIT DESIGN FOR NEAR-THRESHOLD VOLTAGE SYSTEM

by Cai Yunpeng

Near-Threshold Voltage (NTV) techniques have been demonstrated to reduce energy consumption significantly by decreasing the supply voltage approaching the threshold voltage while maintaining the scaling characteristics of Super-Threshold operation. The primary challenge in applications of NTV operation is to ensure robustness, tolerance against variability and resilience against error issues at low voltage. This research focuses on addressing the design challenges in sequential logic at NTV by providing various novel circuits, for improving the SoA designs regarding power, area, robustness, and reliability at NTV.

The first contribution of this thesis is the analysis of prominent types of state-of-the-art Single-Phase Clocked (SPC) FFs and analyses their suitability for NTV operation from transistor level to system level. The yield and the design limitation issue in previously published design, TCFF, is highlighted and addressed by proposing a new circuit topology, named TCFF-NTV. The proposed TCFF-NTV improved the yield of the original TCFF by 95% and 65% power reduction compared to TCFF-based design in system level. The second contribution of the thesis is proposing the 18TSPC, a new topology of fully-static contention-free Single-Phase Clocked (SPC) Flip-Flop (FF) with only 18 transistors, the lowest number reported for this type. It achieves 20% cell area reduction compared to the conventional TGFF. Chip experimental measurements at 0.6V, 25°C show that, compared to TGFF, the proposed 18TSPC achieves reductions of 68% and 73% in overall and clock dynamic power, respectively, and 27% lower leakage. Besides the 18TSPC, 3 more ULP SPC FFs are proposed based on the 18TSPC for providing various solutions for designers to target different ULP design requirements at NTV. The third thesis is the development of a novel Single Event Upset(SEU)-resilient Double Master-Latch Transmission Gate FF (DMTGFF), which is capable of self-detection and self-correction of circuit-level SEU errors. And it can operate in SEU error-free with 0.5V supply voltage (NTV level). The result shows that, compared to the widely used Triple Modular Redundant (TMR) technique, the proposed DMTGFF achieves 15% performance improvement and 25% power reduction.

Contents

List of Figures

List of Tables

Acknowledgements

I would like to use this opportunity to give my thanks to my supervisor Dr. Tom J. Kazmierski and Dr. Alex Weddell for their technical supervision and guidance throughout my PhD. Thank them for their concentration and valuable time during my thesis writing. Much appreciate with their suggestions on this thesis and encourage during the thesis writing. Without their kindness support and encouragement, it is not possible for me to carry on my research successfully.

My special thanks go to Anand Savanth, Pranay Prabhat, James Myers from ARM for their insightful discussions and technical support of the research. It was an enjoyable research experience. My thanks also go to Dr. Ke Li, Dr. Fanfan Meng for their generous advice. I also would like to give my thanks to my colleagues for their invaluable support and the experience sharing. These people include, but is not limited to Prof. David Flynn, Graham Knight, Dr. Sheng Yang, Rohan Gaddh, Yichi Zhang, Kueh Si Mon, Dr. Domenico Balsamo, Prof. Alex Yakovlev.

Finally, I would like to give my warmest thanks to my parents for supporting me with their continuous love, patience, and understanding.

Nomenclature

A_0	Amplitude of the radiation induced transient current.
$ACFF$	Adaptive-Coupling Flip Flop.
C_L	Load Capacitance
CPU	Central Processing Unit
C_{ox}	Oxide Capacitance
$CMOS$	Complementary Metal-Oxide-Semiconductor
$C2MOS$	Clocked-CMOS flip flop
$CMLE$	Clocked Memory Logic Element
CK	the clock signal.
DET	Dual-Edge-Triggered
$DET - TGLM$	Transmission-Gate Latch-MUX
DCV	Duty Cycle Variation
$DVFS$	Dynamic Voltage and Frequency Scaling
$DMTGFF$	Double Master Latch Transmission Gate Flip Flop
$DIBL$	Drain-induced barrier
$DRAM$	Dynamic Random Access Memory
DFT	Discrete Fourier Transform
$DVFS$	Dynamic Voltage and Frequency Scaling
DVS	Dynamic Voltage Scaling
DMR	Dual Modular Redundancy
DC	Detection Clock
Din	The Input of the flip flop.
$DATA_{ML}^{present}$	The present data in Master-Latch.
$DATA_{ML}^{previous}$	The latched data from the input port Din during CK equalled 1.
$DATA_{SL}^{present}$	The present data in the Slave-Latch.
$DATA_{SL}^{previous}$	The latched data from the output of the ML during CK equalled to 1 in SL.
DTA	Dynamic Timing Analysis.
E_{min}	Minimum Energy
E_{switch}	Switching probability of the input data
E_{static}	static energy consumption
E_{static}	dynamic energy consumption
E_{total}	total energy consumption

EDA	Electronic Design Automation
$EAFF$	Error-Aware Flip Flop
FF	Flip-Flop
F_{CK}	Clock frequency
$FO4$	Fan-out-4
$FPGA$	Field Programmable Gate Array
FIT	Failure in Time.
$GDSBD$	The gate-dielectric soft breakdown
IoT	Internet-Of-Things
IoE	Internet-Of-Everything
I_{total}	Integration if the current over the given time.
$I_{leakage}$	Leakage Current
I_S	Source Current
IC	Inversion Coefficient
k	Boltzmann Constant
k_{fit}	Model fitting parameter
K_{DIBL}	Constants of proportionality capturing the impact of DIBL on threshold voltage
K_T	Constants of proportionality capturing the impact of T on threshold voltage
L_{eff}	Efficient channel length
MC	Monte-Carlo Simulation
MCU	Micro-controller
ML	Master Latch
MS	Master-Slave Structure
$MSFF$	Master-Slave Flip Flop.
$MSAFF$	Modified Sense Amplifier Flip Flop
$MUX2$	2-to-1 Multiplexer.
NTV	Near-Threshold Voltage
$NMOS$	N-type metal-oxide-semiconductor.
n	The Sub-threshold slope parameter
$NIMUX2$	Non-Inverted Selecting Pin 2-to-1 Multiplexer.
$OAI21$	OR-AND-INVERTER gate.
$PMOS$	P-type metal-oxide-semiconductor.
P_{total}	Total power
$P_{dynamic}$	Dynamic power consumption
$P_{leakage}$	Leakage power consumption
PVT	Process, Voltage and Temperature
q	Electron Charge
Q_{total}	Total charge of the radiation induced transient current.
$Q_{critical}$	Critical Charge.
RTL	Register-Transfer Level
SoA	State-of-the-Art

STV	Sub-Threshold Voltage
SEU	Single-Event Upset
$SRAM$	Static Random Access Memory
$SDFF$	Semi-Dynamic Flip Flop
SPC	Single Phase Clocked
$S2CFF$	Static contention free single phased clocked Flip Flop
SEL	Single-Event Latch-up
SEB	Single-Event Burnout
$SEGR$	Single-Event Gate Rupture
$SESB$	Single-Event Snapback
$SEFF$	Soft Edge Flip Flop
SET	Single-Event Transient
SEU	Single-Event Upset
SER	Soft Error Rate.
SN	Sensitive Node.
STA	Static Timing Analysis.
$SRLatch$	Set-Reset Latch.
SL	the Slave-Latch.
T	Room Temperature
$TGFF$	Transmission Gate Flip Flop
$TCFF$	Topologically-Compressed flip flop
$TSPC$	True Single-Phase Clock Flip Flop.
T_{CK-Q}	Clock-to-Q delay
TG	Transmission Gate
T_{hold}	Data hold time
T_{setup}	Data setup time
t_g	Gate Delay
T_0	Nominal value of the room temperature
TMR	Triple Modular Redundancy
$tri-inv$	Tristate inverter
ULP	Ultra-Low-Power
$UDSM$	Ultra-Deep-Sub-Micron
V_{cc}	Core Supply Voltage
$V_{nominal}$	nominal supply voltage
V_{GS}	Gate-to-Source Voltage
V_{th}	Threshold Voltage
V_{dd}	Power Supply Voltage
V_{dd0}	Nominal Supply Voltage
V_{th0}	Nominal threshold voltage
V_{th}	Threshold voltage
$V_{E_{min}}$	Voltage for minimum energy operation.

$VHDL$	VHSIC Hardware Description Language
WOV	Window of Vulnerability.
WNS	Wireless Sensor Network.
μ	Mobility, or mean value
Φ_t	Thermal Voltage
σ_{DIBL}	DIBL Factor
σ	standard deviation
μ	Mean value
τ_f	The time constant for fitting the charge collection process of the MOSFET
τ_r	The time constant for fitting the deposition process of the MOSFET
λ	the rail-to-rail pitch of the cell.
δ	the minimum width of the TGFF.
$18TSPC$	18-Transistor Single-Phase Clock Flip Flop.
$19TSPC$	19-Transistor Single-Phase Clock Flip Flop.
$20TSPC$	20-Transistor Single-Phase Clock Flip Flop.
$21TSPC$	21-Transistor Single-Phase Clock Flip Flop.

Chapter 1

Introduction

With the development of Wireless Sensor Network (WSN), energy harvesting, battery and related technologies, the concept of Internet-of-Things(IoT), has been considered as one of the most attractive topics in academic and industrial. The IoT concept refers to the networked interconnection of physical objects via the embedded systems [1], different from the interaction between human and digital devices, IoT enables the interaction of smart objects [2]. Moreover, the key of the IoT network is the IoT leaf nodes devices. Leaf nodes should have the characteristics of sensing, communication, low power (long battery life), low cost and secured. These devices collect the physical data signals and transfer to the cloud for further signal processing. The existed IoT devices, as the interface between the physical world and digital world (see Figure 1.1), have been used in a wide range of applications, i.e. in healthcare, smart environments, and transportation [3][4] etc., and the range ofscenarios is still increasing. Market analysis by ARM Ltd., estimates that there will be over one trillion IoT devices produced over the next 20 years [5].

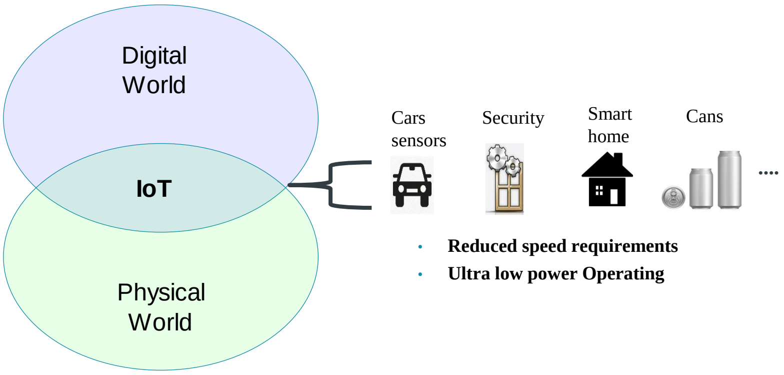


Figure 1.1: Internet-of-Things (IoT) is the extention of the digital world into the physical world [6].

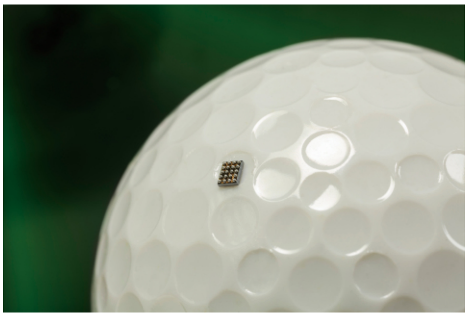


Figure 1.2: The world's smallest ARM-based microcontroller: Kinetis KL03 by Freescale(reproduced from [7]).

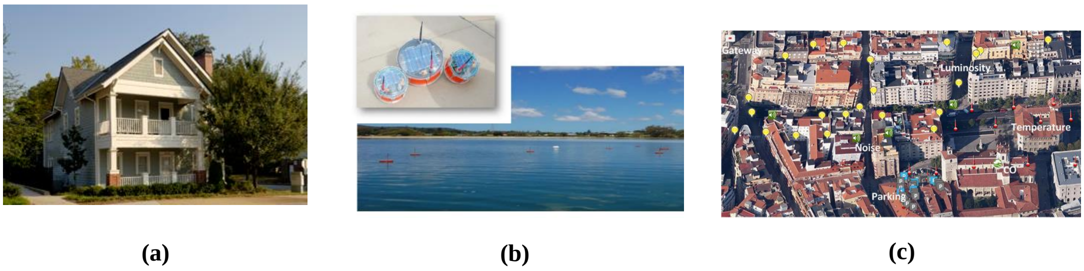
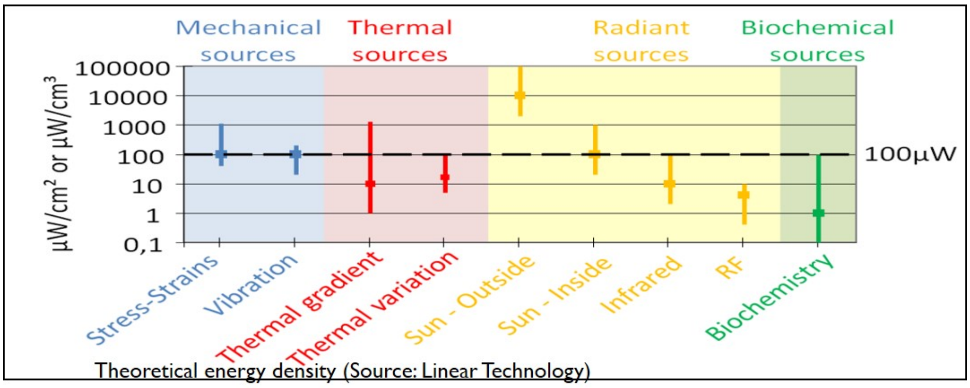


Figure 1.3: Some existing IoT platforms, (a) a small sized network, smart home/office project, Aware Home [11], (b) a large sized network, smart water project: SEMAT[12], (c) a medium sized network, smart city project: Smart Santande[13]

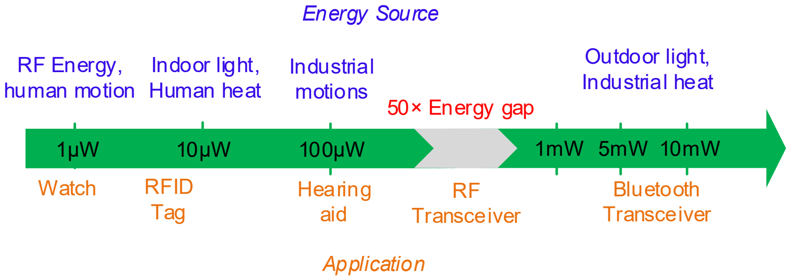
To meet the requirements of the applications they are employed in, IoT devices need to be small, low cost and (the most important criteria for such applications) have extremely low energy consumption. In contrast to speed critical applications, IoT devices are designed for long-term use which is not need to operate at high speed. Figure 1.2 shows an example of an IoT device, called Kinetis KL03 microcontroller (MCU), produced by Freescale. It has been revealed as the world's smallest ARM-based MCU [7]. The minimum Kinetis KL03 MCU (16-pin) is only 9 mm^2. In the Very-Low-Power-Run (VLPR) mode, the power consumption is as low as 50 μA/MHz*V_{cc} (Vcc: 1.71 - 3.6 V) [8]. Besides the Freescale's smallest MCU, some other low power MCUs with the bigger areas can also be found on the market. Such as the MSP430FR5733 Mixed-Signal MCU which is produced by Texas Instruments. It is 36 mm^2 and the active power in low power mode is 91.7 μA/MHz*V_{cc} (Vcc: 2 - 3.6 V) [9]. The STM32L431CB from STMicroelectronics has a slightly bigger package size of 49 mm^2 and the active power is 80 μA/MHz*V_{cc} (Vcc: 1.71 - 3.6 V) [10].

For the large number of incoming IoT leaf node, the power/energy budget is limited. Figure 1.3 shows various existing IoT platforms. The Energy sources of these platforms are either from rechargeable batteries or energy harvesting technologies [2][14]. Although rechargeable battery power is acceptable for current small or medium sized IoT networks



Theoretical energy density (Source: Linear Technology)

Figure 1.4: The power delivered from the state-of-the-art energy harvest technique is in order of 100 $\mu W/cm^2$ [6][15].



~100µW/cm²: power generation from current harvesters

~5mW : Common IoT-oriented devices power consumption

Figure 1.5: The energy source and power consumption of IoT devices map [16].

[11][13], it might not be the optimal solution for future IoT devices. With the trend in the development of WSNs, even the small sized IoT networks, such as the smart home/office, will see the number of devices increase. Also, the large sized IoT networks, such as smart water and smart transportation, suggest that the scale of the projects where these devices are employed is foreseeably larger. When the battery runs out after long-term sensing and monitoring operations, the process of recharging/replacing the batteries is predictably difficult. At the same time, applying energy harvesting techniques to IoT devices is also facing challenges. The latest research shows that the power delivered from the SoA energy harvesting technique is in the order of 100 $\mu W/cm^2$ (shown in Figure 1.4) [6][15]. Figure 1.5 shows the energy source and the power consumption of the IoT devices. It can be seen that the autonomous sensor nodes and the common IoT-oriented devices consumption would be higher than 5mW. The energy consumption of these common devices is 50× higher than the SoA energy harvest solution which lead to the issue of energy gap [16]. The amount of power budget provided by energy harvesting techniques is quite limited for powering IoT devices.

This means that, for digital circuit and system designers, with limited power/energy reservations, the most critical aim is to further minimize the power/energy consumption of the IoT devices. In order to meet the requirement of long-term operation, the demand

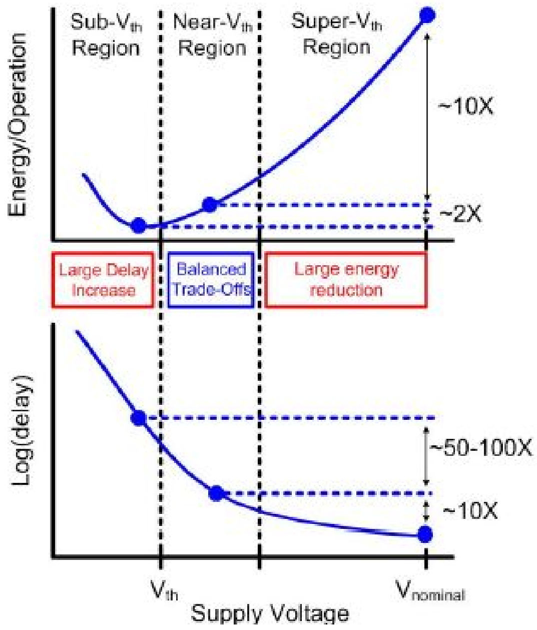


Figure 1.6: Energy consumption and delay in different supply voltage [18].

for implementing applications with Ultra Low Power (ULP) capabilities is becoming obvious.

Indeed, the low power use is not only a requirement for the duty cycled devices, but is also necessary for high performance computing devices. The excessive power density results in a rapid increase in temperature, which limits the yield and reliability of the system. And this roadblock in the semiconductor industry is known as the ”power wall” [17]. For all applications, either the limited power budget IoT leaf node devices or the speed-critical devices, the demand for low power is critical.

Driven by the demand for a further reduction in the power/energy consumption of IoT leaf nodes devices with a limited power budgets and the further development of semiconductor technology, research into power reduction and ULP techniques is essential.

1.1 Research Justification

The total power/energy consumption of the circuit is propotional to the supply voltage. The dyanmic power/energy consumption, as the subcategory of the total power, is both linear function of the supply voltage and operating frequency [19, 20]. In conventional low power/energy design technique, Dynamic Voltage and Frequency Scaling (DVFS) is widely used in today’s low power products [21]. It introduced that slightly scales down the supply voltage V_{dd} from the nominal supply voltage ($V_{nominal}$) (according to the given technology. For 65nm CMOS, $V_{nominal}$ = 1.2 V) and adjusting the clock frequency according to the demand for computing speed. The technique can usually provide a 20% to 40% reduction in energy consumption [22, 23]. The operation voltage region is introduced as the super-threshold region which is around 70% - 100%$V_{nominal}$.

A more aggressive approach, the sub-threshold voltage technique (STV), was introduced to approach the minimum energy (E_{min}) operating point (shown in Figure 1.6 in the STV region (Sub-V_{th} Region), which can provide a 12× improvement in energy efficiency compared to a $V_{nominal}$ operation, by scaling V_{dd} below the threshold voltage (V_{th}) of the transistors [24–26]. However, the large performance degradation and process variation penalty mean that the STV technique can only be used for some specific custom designed devices [18].

Unlike the STV techniques, the recently introduced Near-Threshold Voltage (NTV) approach promises to scale well with a decreasing supply voltage while yields a higher robustness and reliability in comparison to STV techniques [18], which is likely to be more adaptable for future ULP devices.

NTV techniques, a potential solution for implementing ULP applications in power critical designs, have been shown to reduce energy consumption significantly by decreasing the supply voltage to approach the V_{th} (shown in Figure 1.6), while preserving favorable variability, energy efficiency ($2E_{min}$) and performance characteristics (50-100× better) when compared to the STV operations, as a result the NTV technique may be considered as a balanced trade-offs low power technique [27–31].

However, there are some primary challenges in the applications of NTV operations. The first well-known issue which needs to be addressed is the performance loss [30, 32, 33]. Also, there is a pronounced the impact of process variations to circuits is more serve in NTV region than in the Super-Threshold Voltage region [34–36]. What is more, it has been pointed out that circuits working at NTV are more vulnerable to radiation-induced soft errors [30].

1.2 Research Questions

Driven by the opportunities and challenges which exist in NTV techniques, this research aims to investigate the method of achieving the power/energy reduction through adopting the NTV technique, while maintaining the robustness of logic circuits, tolerate process variation and being resilient against errors with the minimum performance penalty and area overhead. Based on this, the research carried out aims to solve following questions.

(b) How is the State-of-the-Art (SoA) sequential circuit design fitting with the NTV technique?

(c) How to propose a novel circuit topology to further reduce the power/energy consumption with (or without) the minimum the yield, performance and area cost penalty.

(d) How to enhance the reliability of the NTV operating circuits?

1.3 Research Contributions

For answering the research questions, the basic concept of power consumption in the digital circuit and the concept of NTV design are reviewed in in Chapter 2. Also, the design challenges and the SoA of NTV techniques are introduced in details and concept of sequential logic circuits. What is more, the literature surveyed the concept of the process variation and radiation-induced errors. The SoA error tolerant circuits are reviewed. Based on the background research, the research of this thesis led the following contributions:

1. ULP Flip-Flops evaluation and analysis and design for NTV operation

   - Prominent types of SoA Single-Phased Clock (SPC) flip-flop(FF)s are reviewed and their suitability is analysed for NTV operation in Chapter 3. Five SPC FFs are reviewed and based on a preliminary analysis, two designs, which meet all NTV circuit design requirements are further investigated. These SPC FFs are designed for NTV operation in TSMC 65LP and compared against the classic transmission gate FF (TGFF). Cell level design issues and variation are explored in the context of a 5000 gate AES encryption macro. The Key design issues are identified in previously published Topologically Compressed Flip Flop (TCFF), which erode the claimed benefits of TCFF when implemented as part of a larger design for NTV applications. Based on the research, it can be realized that aggressive reduction in FF clock loading offers benefits but can lead to functional failures when OCV is considered, especially at NTV. Given the theoretical benefits of SPC FFs for enabling IoT, the need for further work on SPC FF designs is highlighted.

   - Based on the research in evaluation and analysis of SPC FFs at NTV operation, a modified Topologically Compressed Flip Flop is proposed to enable the low voltage (NTV region) operations for addressing the design issues in TCFF, named TCFF-NTV in Chapter 3. The simulation results show the TCFF-NTV improved the yield of original TCFF by 95% and the proposed design brings 54% less hold time variation compare to the conventional TGFF at NTV. Cell-level design issues and variation are explored in the context of a 5000 gate AES encryption macro. The system synthesis results show the TCFF-NTV achieves 65% less activity register power than the TGFF based chip, with 4% area overhead and no performance penalty.

However, from the the evaluation, it can be seen that the area overhead of the new flip-flop design, TCFF-NTV, need always to be considered. This lead for the research on proposing the SPC FFs to meet all ULP FFs design requirements.

- 18TSPC, a novel master-slave topology SPC FF with only 18 transistors (the lowest reported for a fully-static contention-free SPC FF) is proposed in Chapter 4. With a simplified topology, it delivers a 20% reduction in cell area compared to TGFF. Unlike SoA designs, 18TSPC meets all ultra-low power FF design requirements. Although the performance penalty is observed, thanks to the low power characteristic of the proposed design, 18TSPC achieves about 2 times better ED product. It has been implemented in 65nm CMOS along with a TGFF in 320-bit shift-register and AES-128 encryption engine design. This proves EDA compatibility and demonstrates circuit and system-level benefits. The design was first simulated then experimentally validated, at various Data Activity Rate, showing that the proposed 18TSPC achieves reductions of 68% and 73% in overall and clock dynamic power, respectively, and 27% lower leakage compared to TGFF. Furthermore, unlike TCFF, the measurements indicate superior 18TSPC performance at NTV. Besides the 18TSPC, three extending SPC FFs are proposed for further developing. The details of these designs are also proposed in this chapter. All of these variants meet the design requirement of the ULP FFs design requirements and shows the superior power efficiency compare to the conventional TGFF and SoA designs.

2. Soft Error analysis and improvement of digital sequential circuits at NTV.

- The Critical Charge ($Q_{critical}$) and Soft Error Rate (SER) of some typical sequential logic cells, Transmission Gate Flip-Flop (TGFF) and Clocked CMOS Flip-Flop (C2MOS), were estimated at the different supply voltage (0.4V-1.2V) and the simulation results are presented in Chapter 5. The simulation results quantified the increased soft error sensitivity of different devices in the NTV region, The level of increased SER in the NTV Region are also presented. Simulation results show the evidence of devices are more vulnerable to the soft error at NTV while a comparison to the nominal supply voltage operations is provided.

- Based on the research on the soft error in FFs at NTV. Two FF designs are presented in Chapter 5 for achieving the error resilient by increasing the local Critical Charge ($Q_{critical}$) of the FF and adding redundant module blocks to the FF circuit, named Error-Aware Transmission Gate FF (EAFF) and Double Master Transmission Gate FF (DMTGFF). The simulation result shows that the EAFF increased $Q_{critical}$ of the conventional FFs by 1.9× at different V_{dd}. Consequently, the SER decreased at different levels (0.3%-3%) as well.

From the simulation results, it can be concluded that the method of increasing soft error tolerance capability by increasing $Q_{critical}$ is not efficient for NTV operation. So that, the proposed EAFF, aiming to decrease the SER by increasing the $Q_{critical}$ becomes less effective as the circuits operate at NTV.

- Based on the observations, another design named DMTGFF, was proposed. DMTGFF is capable of self-detection and self-correction of circuit-level errors due to SEU events and has superior power consumption characteristics to other state-of-the-art techniques, such as the Dual Modular Redundancy (DMR) architecture, which are based on gate-level error correction. The simulation results show the proposed DMTGFF is capable of operating in error-free at NTV ($V_{dd} = 0.5$V) with the entire range of the transient current pulse amplitude. Comparing with SoA error tolerant FF design, the DMTGFF is 34% and 15% faster than the DMR and TMR respectively. At a 20% data activity rate, the DMTGFF is 25% and 18% faster compare to the TMR and the DMR respectively.

Chapter 6 is the conclusion of the thesis. This chapter summarizes the research and findings presented in previous chapters. Potential future research directions arising from this thesis are also discussed in this chapter.

1.4 Publications

Contributions from the research presented in this thesis have been published as follows:

- **Journal Article**:

  Cai, Y., Savanth, A., Prabhat, P., Myers, J., Weddell, A.S. and Kazmierski, T. ”Ultra-Low Power 18-Transistor Fully-Static Contention-Free Single-Phase Clocked Flip-Flop in 65nm CMOS”, IEEE Journal of Solid-State Circuits, vol. 54, no. 2. pp. 550-559. Oct. 2018

- **Conference Paper**:

  Cai, Y., Savanth, A., Prabhat, P., Myers, J., Weddell, A.S. and Kazmierski, T., ”Evaluation and analysis of single-phase clock flip-flops for NTV applications”, In Power and Timing Modeling, Optimization and Simulation (PATMOS), 2017 27th International Symposium on (pp. 1-6). IEEE.

- **Granted Patent**:

  Savanth, A., Myers, J.E., Yunpeng, C.A.I., Weddell, A.S. and Kazmierski, T., University of Southampton and Arm Ltd, 2018. Flip-flop. U.S. Patent 9985613.

Chapter 2

Literature Review

As discussed in Chapter 1, this thesis aims to investigate methods that achieve power (or energy) efficiency by adopting the NTV technique, while maintaining the robustness of circuits, toleration of process variation and maintain resilience against errors. In this chapter, literature on the general concept of applying NTV techniques to provide specific potential solutions is reviewed. It also introduces the basic concept of power (or energy) consumption sources for digital circuits, the NTV computing technique definition and design challenges of the core technique are discussed. Also, the SoA research into the NTV computing technique is introduced at the end of this section. In Section 2.2, the research concentrates on sequential logic designs. After that, Section 2.3 introduces the concept and the impact of process variation errors and radiation-induced errors on modern IC circuits. Some SoA designs for robustness and error tolerance are introduced at the end of this section. Section 2.4 provides the concluding remarks for this chapter.

2.1 Near-Threshold Voltage Computing

2.1.1 Power/Energy Consumption and Dynamic Voltage Scaling

Power consumption is an instantaneous (averaged) concept. In general, Power consumption sources can be clarified as the dynamic and static consumptions. Figure 2.1(a) illustrates how the dynamic power is consumed by charging (discharging) the load capacitance of the gate. The expression of the $P_{dynamic}$ is shown in Equation 2.3, Where the V_{dd} is the supply voltage, F_{CK} is the clock frequency, C_L is the load capacitance of the circuit, α is the switching probability of the input data of the circuit. The $P_{dynamic}$ can be reduced by slowing down the clock frequency to the lowest suitable value, the C_L can be reduced by using smaller transistors or reducing the wire capacitance (i.e.,

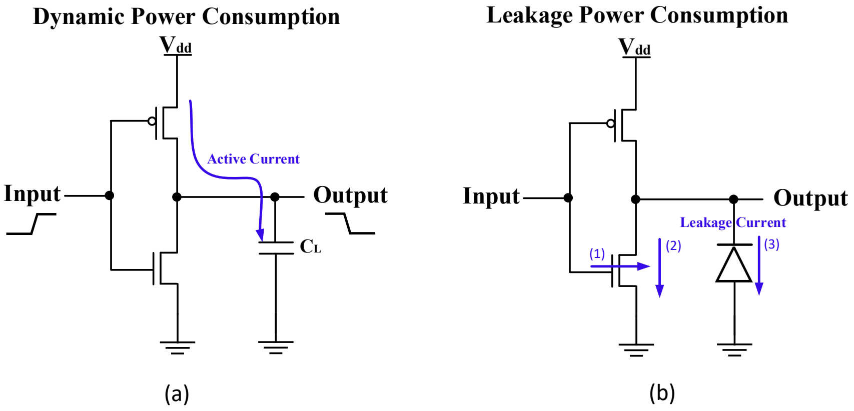


Figure 2.1: Power consumption in CMOS digital circuit, sample circuit: INV Gate. (a) Dynamic power consumption. (b) Leakage power consumption.

shorter metal wires), reduce the switching activities (e.g., applying clock gating to sequential logic circuits). However, the most efficient method is scaling down the V_{dd} since it brings a quadratic reduction in $P_{dynamic}$.

$$P_{dynamic} = V_{dd}^2 F_{CK} C_L \alpha \tag{2.1}$$

However, for experimental validation or real device measurement, the average $P_{dynamic}$ is modelled as Equation 2.2. Where the $i_{DD}(t)$ is the transient current value at time t. T is the given interval time [37].

$$P_{dynamic} = \frac{1}{T}\int_0^T i_{DD}(t)V_{DD}dt = \frac{VDD}{T}\int_0^T i_{DD}(t)dt \tag{2.2}$$

Figure 2.1(b) shows the Leakage power consumption in digital circuits and three primary sources of the leakage current is shown in digital circuit [20][21].

(1) Gate leakage current: Gate leakage current (see Figure 2.1(b−1) is the current which flows from the gate terminal to the substrate of the CMOS directly through the oxide layer of the CMOS, and it happens due to the hot carrier injection and gate oxide tunnelling.

(2) Sub-threshold leakage current: Sub-threshold leakage current (see Figure 2.1(b−2) is the current flows from the drain terminal to source terminal when the transistor is operating in the weak inversion region.

(3) Drain junction leakage current: Drain junction leakage current (see Figure 2.1(b−3) happens when a different potential is formed between the drain diffusion region and substrate of the transistor.

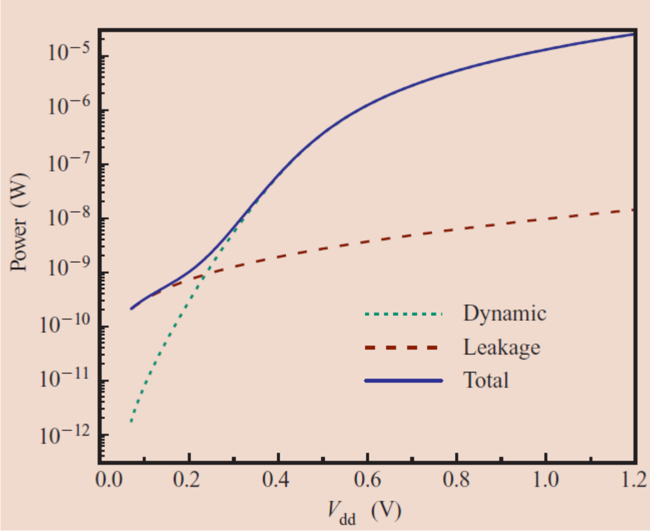


Figure 2.2: Power consumption against the supply voltage V_{dd} [45].

Generally, Some other sources of leakage current, such as Reverse bias junction leakage, punchthrough current, contention current etc.[38], will not be discussed in details due to the conditional characteristics or the negligible effects in leakage power consumption. However, for some specific circuits , these types of leakage current still need to be considered (e.g. the crobar currents in level shifter circuit) [39–41].

$$P_{leakage} = V_{dd} I_{leakage} \tag{2.3}$$

So that, in Equation 2.4, $P_{leakage}$ can be modeled as the product of Leakage current($I_{leakage}$) and V_{dd}, where can simplify all of the leakage current sources as the $I_{leakage}$. $P_{leakage}$ can be reduced by lowing the V_{dd} or reducing the $I_{leakage}$. Several approaches can by used to minimize $I_{leakage}$, such as the Multi-V_{th} techniques [42], power gating [43], body-biasing [44] etc. Alternatively, $P_{leakage}$ can be reduced by scaling down the V_{dd}.

$$P_{total} = P_{dynamic} + P_{leakage} \tag{2.4}$$

The total power consumption (P_{total}) of the digital circuits can be considered as the sum of the dynamic power consumption ($P_{dynamic}$) and the leakage power consumption ($P_{leakage}$), which is shown in Equation 2.4. From the Equation 2.1 - 2.4, it can be considered that scaling down V_{dd} can bring about a reduction in $P_{dynamic}$ and $P_{leakage}$ and so reduce the P_{total}, see Figure 2.2.

Different from the power, an instantaneous (averaged) concept, energy can be considered as the integration of the instantaneous power over a period of time, and it is proportional to the V_{dd}. It needs to be considered that, the reduction in total power consumption cannot always promise a reduction in energy consumption since the operation time need to be considered [45]. The energy consumption can be modeled as Equation 2.5, where the $t1$ is the start time of a (or serials of) operation(s), $t2$ is the deadline of the work. E_{total} is the integration of the power over the δt ($\delta t = t2 - t1$). Since the power is the product of current (I) and V_{dd}, E_{total} also can be modeled as the product of integration of the current (I_{total}) over the δt and the V_{dd}. The E_{total} is proportional to V_{dd} and scaling down the V_{dd} can brings benefit in Energy efficiency [46, 47].

$$E_{total} = \int_{t1}^{t2} P_{total}(t)dt = \int_{t1}^{t2} I_{total}(t)V_{dd}dt \tag{2.5}$$

However, it should be recognized that the speed of the circuit can be degraded when V_{dd} is scaling down. The maximum operation frequency of a voltage scaled circuit can be modeled as Equation 2.6 [48]. Where the f and $delay$ are the operation frequency and delay of the circuit respectively, k is the fitting parameter for the specific technology node.

$$f = 1/delay = k(V_{dd} - V_{th}^2)/V_{dd} \tag{2.6}$$

On the one hand, the lower V_{dd} with lower f can brings benefit in dynamic power (or energy) efficiency. On the other hand, the portion of static power (or energy) can be increased due to the longer operation time. The increased static power can even cancel the benefit of dynamic power reduction. Therefore, a limitation voltage point exists in DVS which makes the static energy dominate the E_{total} and makes the DVS technique unable to achieve a futher energy efficiencies. Figure 2.3 [49] shows the $E_{dynamic}$, E_{static} and E_{total} against the V_{dd}. In $E_{dynamic}$ dominant region, the DVS is very efficient in E_{total} reduction. The voltage point of $E_{dynamic}$ and E_{static} are balanced, the DVS achieves the minimum Energy (E_{min}) point. If the voltage is scaled down below the $V_{E_{min}}$, the E_{total}

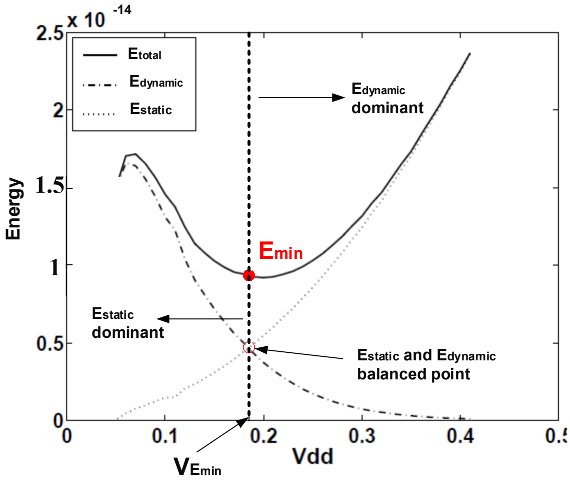


Figure 2.3: Energy consumption against the supply voltage V_{dd} [49].

increases since the E_{static} starts to became higher than the $E_{dynamic}$.So that, to achieve the E_{min}, the V_{dd} need to be scaled down to $V_{E_{min}}$ which is lower than the V_{th} of CMOS.

Therefore, several facts can be ascertained from the discussion above. Although the P_{total} can be reduced continuously with V_{dd} scaling down, E_{total} can still rise when V_{dd} is below the $V_{E_{min}}$ due to the increased operation time. So that, to achieve both power and energy reduction, V_{dd} should not be scaled lower than $V_{E_{min}}$.

2.1.2 Technique Definitions

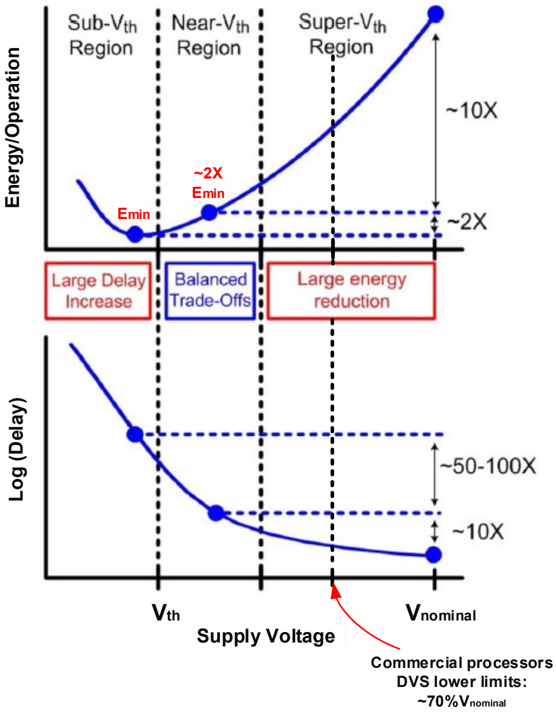


Figure 2.4: The Near-threshold computing operation region [18].

From the previous Section, it can be seen that scaling down the V_{dd} (but no less than $V_{E_{min}}$) is an efficient method to minimise the P_{total} and E_{total}. DVS technique introduces an approach to improve power efficiency by reducing the voltage slightly from the nominal value [19]. In conventional DVS techniques, commercial processors [50–52] usually set a conservative lower boundary of the voltage scaling range, which is around 70% of the nominal voltage ($V_{nominal}$) of the circuit. This boundary is derived from a trade-off among the design effort, cost, performance, and circuit robustness considerations [18, 49].

Researchers exploring the possible lower boundary of DVS and some aggressive voltage scaling techniques announced achievements in low power (or energy) consumption in CMOS design [53–55]. The ULP technique, subthreshold voltage computation (STV)

[26, 56] technique, is one of the aggressive DVS techniques. In STV, the V_{dd} of the circuit is scaled down to achieve the E_{min} [49], since the E_{min} is the balanced point of E_{static} and $E_{dynamic}$. The corresponding V_{dd} of E_{min} ($V_{E_{min}}$) is lower than the threshold voltage of CMOS [56]. Although the CMOS circuit working in STV thought to be working in the most power(or energy) efficient operating region [30], the performance degradation of this technique means it is mainly used for some specific applications instead of being used in general function applications [57].

From Figure 1.6, it can be seen that although the minimum energy operation is in the STV region, dramatic performance loss can also be observed. Instead, increasing the supply voltage so that it is slightly higher than the threshold voltage allowsthe circuit to operate in the NTV region, the energy reduction is still in the order of 10 times less and the energy consumption is just increased by 2× that of a circuit operated in the STV region. Also, the benets of operating in the NTV region can also be seen in the performance. Compared to the performance in the sub-threshold region of operation, the delay was exponentially decreased (50-100× performance degradation) when the circuit enters the NTV region. The NTV region is considered to be the balanced trade-off region which maintains an acceptable performance with dramatical energy reduction compared to the conventional nominal voltage operation [18, 30, 58]. So that, the circuit in the NTV region maintains the power (or energy) efficiency characteristics which it has in the STV region [18]. Compared to the previous sub-threshold operation, it also can be seen that the NTV operation would have improved performance allowing more opportunities for it to be developed for wide variety of uses [57].

2.1.3 NTV Design Challenges

The NTV region is regarded as the balanced energy-performance trade-off region, and the benefits of the NTV technique has been introduced in the previous section. However, for this stage, this technique although attractive has not found wide spread use especially in commercial applications because of barriers that the technique is still facing which need to be addressed.

The first barrier the NTV operation is facing is performance loss and this can be directly observed in Figure 1.6 [18, 30, 57]. It has been reported [36] that the Fan-Out-4 (FO4) inverter delay in the NTV region is 10× slower than when working at $V_{nominal}$. To investigate the delay variability of the NTV operated circuits, the Monte Carlo simulation was done to the test chips. It can be observed that there was 5 × as much variation compared to the nominal voltage operation [18]. With the worst-case 3σ variation in the PVT (process, voltage, and temperature) consideration, the simulation shows that the delay variability of a logic gate can be increased by 20 × respectively [36].

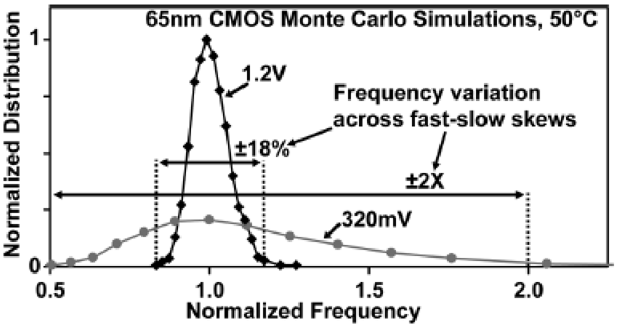


Figure 2.5: The frequency distribution in typical process at nominal supply voltage (1.2V) and the frequency distribution in fast-slow die at NTV (320mV) [58].

While the supply voltage was reduced to approach the V_{th} of the CMOS, the reliability of the circuit was found to be more susceptible to the process variations [30]. Kaul, H. et al. carried out measurement and variation analysis on the circuit at both nominal supply voltage and NTV across the fast-slow dies. The results show that the spread of frequency at nominal voltage is 18% and it also shows that frequency distribution has an increase two times higher at NTV, shown in Figure 2.5 [58].

Recent research [59] also show that the process variations also impact the circuit sensitivity to soft errors at NTV. The Monte Carlo simulation results of the SRAM which works at the nominal supply voltage shows that the probability of soft errors, specificity the SEU error, is firmly related to the parameter variations such as the injected charge variation, the threshold voltage variation and so forth. It has also been shown that the process variation has a significant impact on the soft error sensitivities of the circuit. It has been shown that the channel length variation and the V_{th} variation has the most significant impact to the $Q_{critical}$ [60] among various of the parameter variations. Therefore, the circuit with process variation can be more vulnerable to soft errors at NTV [30]. The Figure 2.6 shows the correlation between the V_{dd} and the Soft Error Rate (SER) in different technology nodes. It can be seen that the SER of the circuit is increases with reduced supply voltage and it also can be assumed that the circuit operating in the NTV region would have a higher SER. By reading the different technology node in Figure 2.6 and making a comparison, it can also be seen that the smaller technology nodes have a lower SER. Moreover, the superiority can be maintained in NTV region.

For the sequential logic, the latches and the FFs design, the NTV operation brings the data retention and hold-time variation issue [61]. Data retention is one of the fundamental metrics for the latches and FFs. If the data retention failure happens, the erratic state will be propagated to the next stage circuits which can trigger the errors. Further, same as the radiation-induced errors, the data retention failure can further corrupt the system. There are two primary sources of the data retention failures. One is the process variation, which has been introduced in the previous section. Another source of the data retention failure is the gate-dielectric soft breakdown(GDSBD) [62].

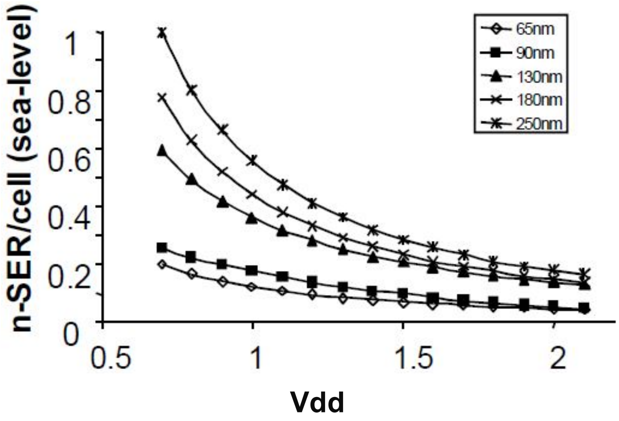


Figure 2.6: The correlation between supply voltage and soft error rate in different technology nodes [30].

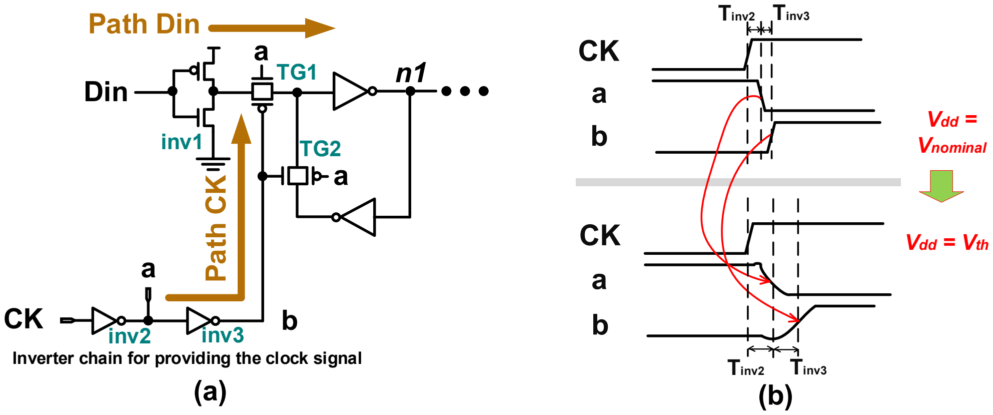


Figure 2.7: (a) Hold Time (T_{hold}) path analysis of the TGFF [65]. (b) The waveform of the internal clock signal inverter chain with $V_{dd} = V_{nominal}$ and $V_{dd} = V_{th}$.

The phenomenon of GDSBD can be observed when the gate oxide thickness is scaled under 5 nm [63]. When the electron traps are generated and formed a conductive path from the gate to the substrate, the GDSBD can occur [64]. The impact of these sources of the data retention failure can be intensified at NTV. It was proposed that an upsize in transistor size of the feedback keepers in latches and FFs could mitigate the process variation and GDSBD induced data retention failure. However, the power consumption, area, and propagation delay of the circuit are sacrificed [61].

Hold time (T_{hold}) variation is another barrier which the NTV operated sequential logic is facing [61]. Choosing the widely used conventional FF design, Transmission Gate Flip Flop(TGFF), as the case study (see Figure 2.14(a)). The CK signal is connected to an internal delay chain for generating the internal clock signal, the a and the b. The operation waveform of the internal clock signal inverter chain at $V_{dd} = V_{nominal}$ and $V_{dd} = V_{th}$ is shown in Figure 2.7(b). The T_{inv2} and T_{inv3} are the propagation delay of the inverters $inv2$ and $inv3$ in the clock path (Figure 2.7(a)). The transition delay between the b and the a makes the PMOS in TG1, which is the Data path, close later than the NMOS in TG1, which lengthens the transmission time of the TG1. If the input data in Din is 0, the transmission time increase is in its worst case. This is because the PMOS is strong in pass logic 1 while it is working as a logic switch. While the Din is 1, a 0 is latched in the ML. However, the delayed signal b will turns on the NMOS later in TG2 than the PMOS in TG2. So that, the TG2 is weakened in passing 0 which makes the feedback loop is weak in latch the 0. While considering the effect of the process variation, the hold time of the FF can be increased [66]. When the supply voltage is decreased to the NTV level, $V_{dd} = V_{th}$, it can be considered that the T_{inv2} and T_{inv3} are increased. Therefore, the on-time PMOS of TG1 in Figure 2.7 is further extended. The T_{hold} variation can be worse when the process variation is considered in NTV operation. Recent research [65] to the TGFF shows that the T_{hold} in 3σ process variation at NTV gives 10× increase compared to TGFF operated at the nominal voltage .

2.1.4 State-of-the-Art Circuit Level NTV Techniques

Since the concept of the NTV design was proposed in 2007 [28], researchers have carried out practical research and made improvements to the NTV design which address the barriers introduced in the previous section. The SoAs of the NTV techniques are briefly introduced in this section.

As introduced in the previous section, circuits that work at the NTV region are about 10 × slower than the operations at the nominal supply voltage. The research group from the University of Michigan raised the point that parallel and many-core architecture can be applied to relieve the problem of performance degradation [32].

Recent researchers [18, 28] have accepted this idea regarding NTV architecture level designs. The operation tasks are distributed to many NTV operated cores. Although these cores are working at the low clock frequency, the performance and the throughput of the overall microprocessor is maintained at a reasonable level.

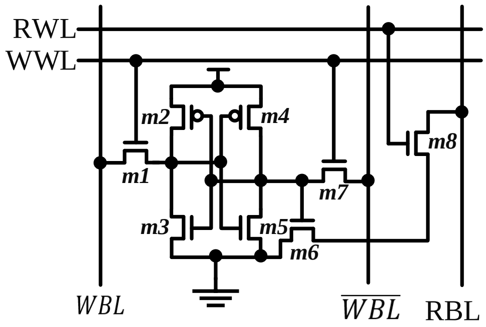


Figure 2.8: The schematic diagram of the 8T SRAM [67].

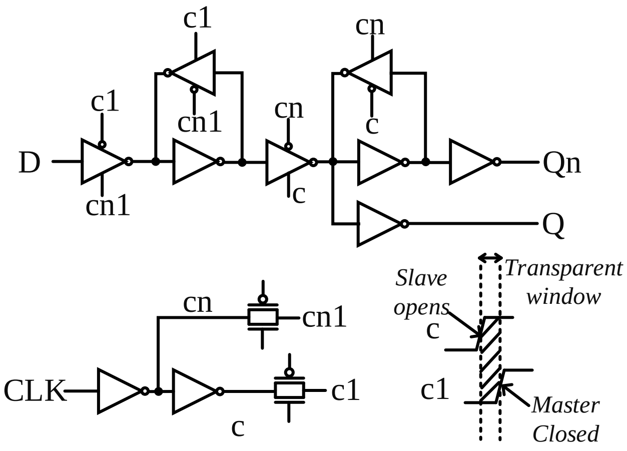


Figure 2.9: The schematic diagram of the SEFF [69].

Due to the impact of the process variation, some relevant parameters would be changed accordingly (V_{th}, L_{eff}) and these variations can lead to function failures of the circuit. One of the most critical devices in the system is the memory devices. A recent paper [30] has been pointed out that the variation at NTV might cause the function failure in conventional 6T (6 Transistors) SRAM. In order to address the variation induced function failure problem, an 8T (8 Transistors) structure SRAM [67] (shown in Figure 2.8), is proposed to provide better robustness in the NTV region. It can be seen that the extra two transistors ($m6$ and $m8$) separate the write operation and the read operations, performed on an isolated read-out buffer. The 8T SRAM design provides the designers the opportunity to arrange the size of the MOSFET which is used for the write operation which would provide higher stability.

A new SRAM paper [68] proposed a 10 transistor architecture based on the previous introduced 8 transistors SRAM to address the yield constraints in the STV region operation. However, the 10T SRAM design also brings an overhead in cell area and power consumption.

Soft Edge Flip-Flop (SEFF) [69], shown in Figure 2.9, was introduced as a potential solution to address the timing variation. SEFF accepts the latch based FF (i.e., the TGFF topology) design instead of the hard edge Clocked-CMOS (C2MOS) type FF design. The Latch style design provides half clock periods for the transparent window and any status change at the input port will be propagated to the output during the transparent time. This operation removes the time constraints which exist in the conventional FFs. The SEFF was designed based on the latch type based FF and use the logic block to delay the clock of the ML in order to generate a small period of the transparent window so that the late arriving data would be successfully captured without the hard boundary [69]. As it is shown in Figure 2.9, the clock signal for ML was delayed, so that a transparent window was created. However, the transparent window also makes

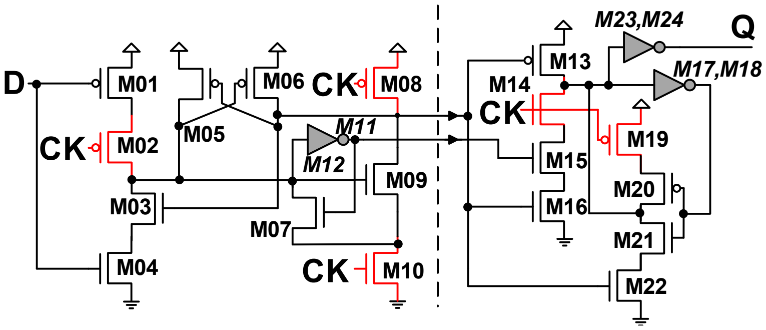


Figure 2.10: The schematic diagram of the Static Contention-free Single-phase-Clocked Flip Flop (S2CFF) [65].

it possible for error data propagating and flipping the data which lead to data error. Further research on SEFF is required to reduce the potential errors.

Figure 2.10 shows the Static Single-Phase Contention-Free Flip-Flop (S2CFF) [65]. The S2CFF is developed based on the True single phased FF (TSPC) and an additional conventional SL for addressing the T_{hold} variation issue. In SPC FF, the clock delay chain in conventional FF design is removed. So that, there are no inverters for providing the complementary clock signal. As a SPC FF, the S2CFF can be directly driven by the global clock signal. The SPC operation has the advantage in power efficiency and mitigates the hold time variation in NTV region since the inverters induced propagation delay is removed. The S2CFF is Contention-Free so that there is no contention power consumption due to the state-contention. The total device count is 24, which is the same as the conventional TGFF. As it is advocated in the S2CFF paper, in worst case 3σ Monte Carlo simulation at NTV level, the S2CFF improves the ability to resist the T_{hold} variation by 3.4× when comparing to the conventional TGFF at the same condition. The topology of S2CFF gives superiority in power efficiency in all ranges of data activity ratios compared to the TGFF. However, the complex topology results in area overhead in the layout compared to the TGFF, although the transistors count for S2CFF is same as TGFF.

2.2 Sequential Logic Circuits

In general, the digital system is categorized as a combinational logic circuit and sequential logic circuit [20]. It is different from combinational logic circuits where the output state is only dependent on the current input signals (Figure 2.11 (a)), no clock signal is involved. Sequential circuits, such as state machine (Figure 2.11 (b)), pipeline structure systems (Figure 2.12), registers ,etc.. The output of sequential logic circuits do not only according to the current inputs but also the previous state of the block. Therefore, the concept of 'temporality' is involved in circuit design. The clock signal is required to be

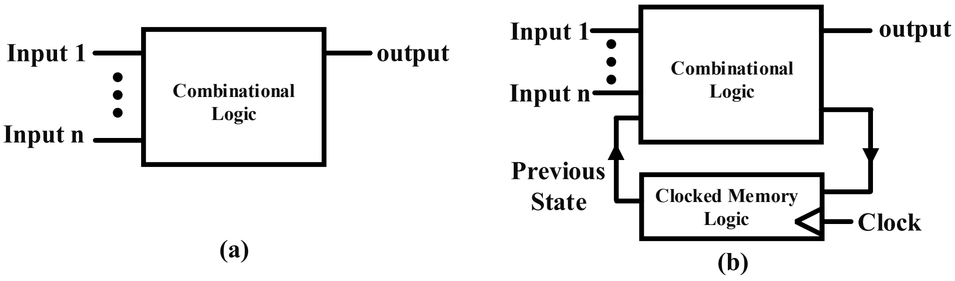


Figure 2.11: Abstract of (a) a pure combinational logic circuit. (b) Sequential logic circuit example: state machine.

adopted in the sequential system to separate the states. Otherwise, the out-of-sequence states conflict with the current input which can lead to the race-condition in logic [20].

2.2.1 Clocked Memory Logic Elements

In modern CMOS VLSI design, the design of the sequential circuits are considered to be the most important aspect of the design as the quality of the sequential logic system has a considerable impact on the overall system performance and power (or energy) consumption [70]. More specifically, the Clocked Memory Logic Elements (CMLE) is the most critical component in sequential circuits and the quality of sequential circuit is heavily dependant on the CMLE designs [71][72].

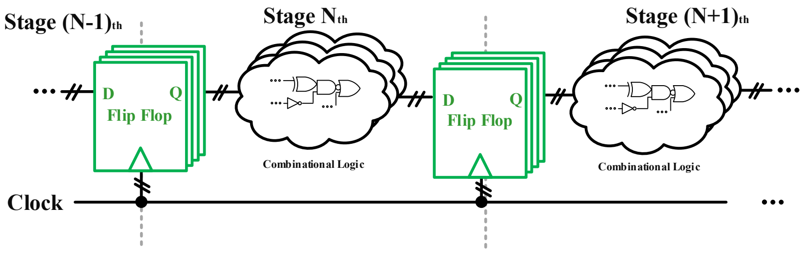


Figure 2.12: Diagram showing a pipeline structure system.

Take the pipeline structure as a case study, Flip Flops (FFs) as the CMLE is inserted between each stage of the pipelines to hold the state for one clock cycle and keep the system synchronized. Also, it prevents the early transitions between stages and FFs and so regulates the data flow of the system [70].

As the interface between each stage of combinational logic, the performance of FF affects the performance of the whole system. Timing characteristics (Clock-to-Q delay, Setup Time, Hold Time, etc.) of FF (or CLME in general) directly affects the system operating frequency configuration during the chip implementation phase. Alternatively, according to the performance requirement, CLME need to be carefully selected to meet the speed requirement. Physical implementation engineers need to put significant effort into the Static-Timing-Analysis (STA) and Dynamic-Timing-Analysis (DTA) to make sure no setup time or hold time violation appears (or with acceptable slacks) in the critical path of the system. For this reason, time constraints caused by FF timing characteristics was considered to be a sequencing overhead or clocking overhead [20][73]. The definition FF timing characteristics will be discussed in Section 2.2.2.

For today's typical SoC, up to 10 Millions of FFs as CLME can be applied to the whole system [74]. Due to the high level of activity in the clock signal, the total power (or energy) dissipation of the sequential circuits with the large number of CLME can be up to 50% of the overall power (or energy) budget of the chip [70]. As it is discussed in Chapter 1, the excessive power density can rapidly increase the temperature which limits the yield and reliability of a system. Moreover, for limited given power (or energy) budget systems (i.e., the IoT leaf nodes devices), the power (or energy) reduction for FFs (or CLME in general) is crucial.

Along with the performance and power (or energy) consumption consideration of the CLME, the robustness, and reliability of the CLME also needs attention. With the effects of the Process, voltage and temperature (PVT) variation, radiation-induced errors (discussed in Section 2.3), design limitations ,etc. state altering or functional failure can happen with CLME. These effects can further degrade the performance or induce the failure of the whole system [70–72, 75]. For a better understanding of the CLME, this section presents a general review of CLME.

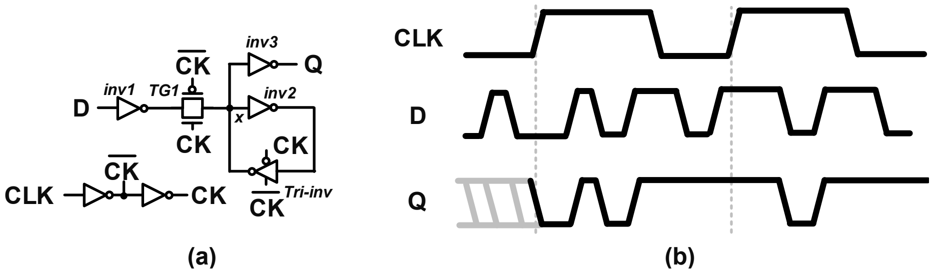


Figure 2.13: (a) Schematic diagram of a typical D Latch [20]. (b) Operation waveform of the D Latch

The Latch is defined as the level sensitive memory element and it is a critical component for building Master-Slave Flip Flop (MSFF) [76]. Taking the positive level sensitive latch as a case study. When the Enable signal is low, the latch operates in hold mode. The output data is the state what has been stored. Switching at input port will not alter the restored data. When the enable signal is high, the latch works in transparent mode. Figure 2.13(a) shows the structure of the representative D Latch design which is widely used in standard cell library [77]. It uses a transmission gate ($TG1$) for data flow control. By using a transmission gate, the V_{th} degradation issue in pass transistor can be solved. Also, a static back-to-back structure ($inv2$ and $tri\text{-}inv$) is used for data hold which

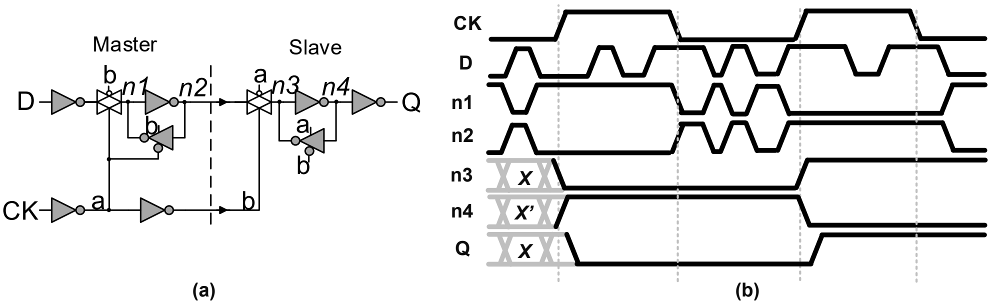


Figure 2.14: (a) Transistor level schematic diagram of Transmission Gate Flip Flop (TGFF) [76][78]. (b) Internal nodes waveform of TGFF.

ensures x is full swing (0 to V_{dd}) without charge leakage issue which can be found in dynamic logic circuits. Note that a tristate inverter (*tri-inv*) controlled by CK is used in the back-to-back structure. When $CK = 1$, D Latch is in transparent mode, the feedback loop in the back-to-back structure is clamped. The conditional operated path removed the data contention issue in conventional *inv-inv* back-to-back structure. Also, the *inv*1 and *inv*3 are placed in input and output port for noise isolation which promises the design with good robustness. The operation waveform is shown in Figure 2.13(b). The D Latch is used as the critical component in the widely used Transmission Gate Flip Flop (TGFF).

Rather than level sensitive memory element (Latch), Flip Flops (FFs) are defined as edge-triggered clocked memory device [20][79]. It samples the data at input port D at the edge of the Clock (CK), sampled data will be held for one clock cycle. For the FFs sample data at the rising edge of the CK, the FF can be considered as a positive edge-triggered FF. In contrast, negative edge-triggered FF samples data at the falling edge of the CK.

Figure 2.14(a) shows a widely used TGFF which is one of the most representative Master-Slave (MS) structure positive edge triggered Flip Flop [76][78]. It can be seen the previous introduced D-Latch structure (shown in Figure 2.13 (b)) is used as Master and Slave latch (ML and SL) in TGFF. Two inverters are required in MS structure for providing positive and complemented clock signal to control the latching activity and data flow. The waveform of the TGFF internal nodes are shown in Figure 2.14(b). The benefits from the static structure are that the TGFF appears to have better characteristics in terms of performance, power consumption, area, and robustness [75, 76, 80] when compared to other MSFFs [81–83] and the TGFF is considered as the de-facto industry standard FF cell [77, 78, 84].

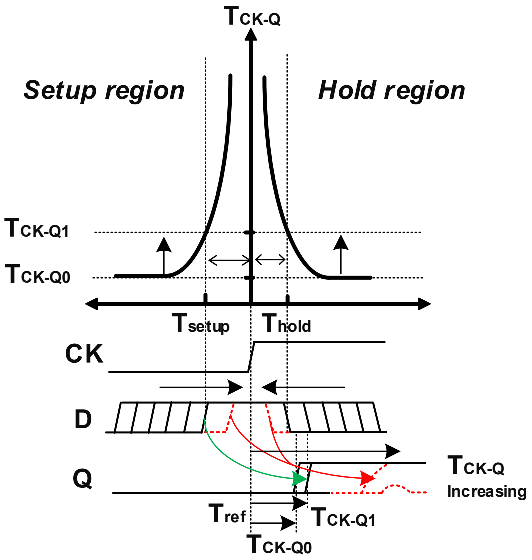


Figure 2.15: FF timing characteristics diagram [85].

2.2.2 Flip Flop Timing Characteristics

Figure 2.15 shows the timing diagram of the FF timing parameters, including Setup time (T_{setup}), Hold time (T_{hold}), Clock-to-Q Delay (T_{CK-Q}). T_{CK-Q} is measured between the 50%V_{dd} of the CK transition edge and 50%V_{dd} of the Q transition edge. From the 2.15, it can be seen that the T_{CK-Q} is steady when the D transition edge is away from the CK transition edge. The steady T_{CK-Q} is shown as T_{CK-Q0} in the Figure. When the D transition is approaching to the CK edge, T_{CK-Q} can be gradually increased and finally got metastable or failure. Normally, the T_{setup} is defined as the time before CK transition edge which increases the a 5%-10% T_{CK-Q0} (see T_{CK-Q1}). T_{hold} is defined as the time of D need to be held stable after the transition edge of the CK. Similar to the previous scenario, the T_{CK-Q} is increased when D switches too early after the CK edge. T_{CK-Q} can be gradually increased and finally got metastable or failure. Simiarly with the T_{setup}, the T_{hold} is defined as the time before CK transition edge which increases the a 5%-10% T_{CK-Q0} (see T_{CK-Q1}) [20, 84, 85].

2.2.3 State-of-the-Art Flip-Flops Design

Besides the MS structured FFs, some other classes of FFs have been proposed over the past 20 years. Back-end engineers need to carefully select the suitable FFs to meet the design requirements.

Figure 2.16 (a) shows a representative FFs of Dual-Edge-Triggered (DET) structure FFs [90], named Transmission-Gate Latch-MUX (DET-TGLM) [86]. DET FFs use half of the single edge triggered FFs' clock frequency to achieve the same throughput. So that, if the DET FFs are used in a system, a 50% power reduction can be achieved [86].

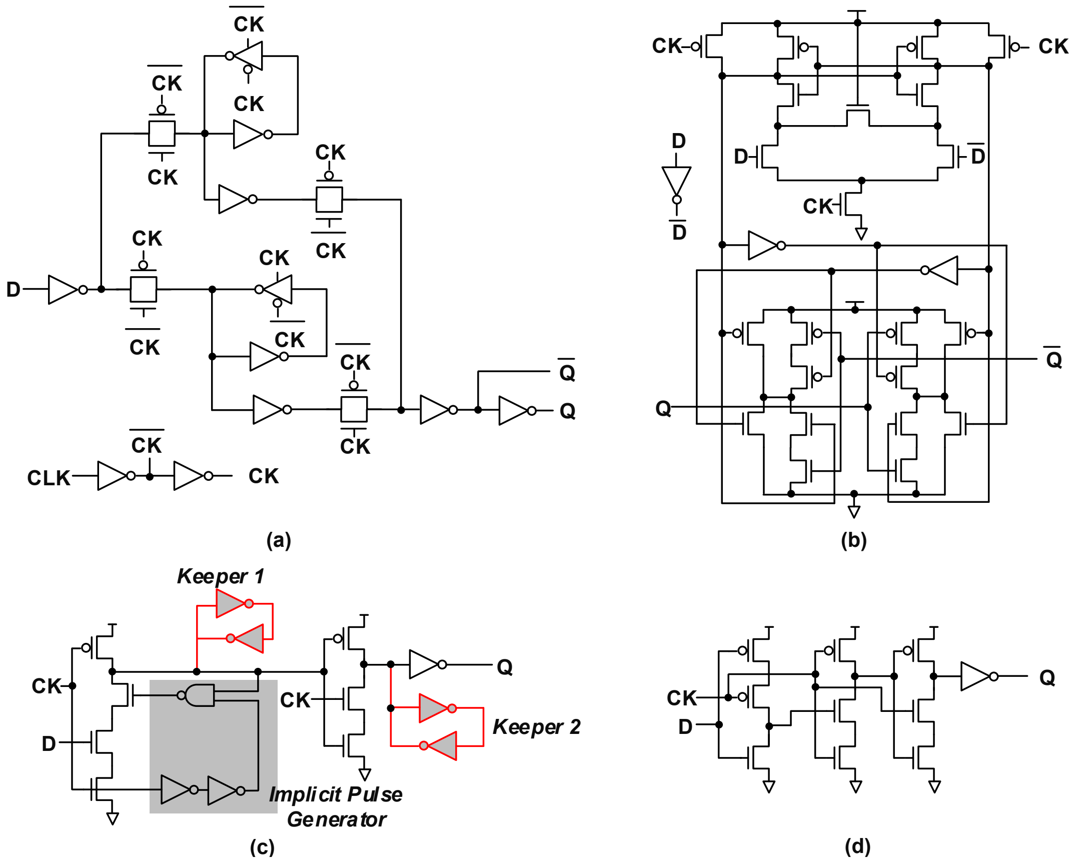


Figure 2.16: Other classes of FFs structures and its representative designs: (a)DET-TGLM [86]. (b)MSAFF [87]. (c)SDFF [88]. (d)TSPC (TSPC) [89].

However, It has a larger area (15% - 20%) when compared to conventional single edge triggered flops. Also, Since the DET FFs sample data at both CK rising and falling edge, it means the DET FFs are sensitive to the Duty Cycle Variation (DCV) [91]. To reduce the effect of DCV, the symmetrical clock buffers are required in DET FFs which puts a higher load on the clock network. So that, when estimating the overall power, including power dissipation of clock buffers, the power efficiency of the DET design can be averaged out. Extra effort is needed by designers to balance the tradeoff when considering DET-FFs [90, 92–94]. This is one of the key reasons why the DET FFs are not widely used in todays VLSI designs [20].

Figure 2.16 (b) shows a representation of the Differential Structure (DS) FFs [95–98]: Modified Sense-Amplifier FF (MSAFF) [87]. The differential structure samples the differential input (D and $\overline{D}$) by a pre-charged sense-amplifier at the rising edge of the CK. This enables the DS FF to not require the input signal achieves full swing [20][87]. The benefits of this sturcture are that the DS FFs have comparatively higher speeds than to MS FFs. For this reason, it is widely used in performance driven processors such as Alpha 21265 [99], StrongArm 110 [100] and AMD K6 [101] etc.. However, the DS structure has higher power consumption in general when compared to the MS FFs. With the same design criteria, the most energy efficient DS FF, MSAFF, still consumes about 70% higher Energy than the TGFF [92]. So that, DS FFs are more suitable for performance-driven systems [92].

Figure 2.16 (c) shows the representation of a pulsed FF, named Semi-Dynamic FF (SDFF) [88]. The pulsed FFs were designed to target the high-performance requirement [102–104], such a structure has been used in the Ultra Sparc III processor [103]. However, the high-performance characteristics has often achieved at the expense of a high power overhead. Take the SDFF for instance, the dynamic logic circuits are included in the design to yield high-speed operation. Keepers ($Keeper1$ and $Keeper2$) are used to overcome the charge dissipation issue. However, when the state alters at the dynamic node, new data overwrites the previous data in the Keepers. Since these feedback loops are not isolated, data contention occurs. Temporary short circuit paths can be built due to the racing between the keeper and the previous stage logics which lead to extra power consumption, known as contention power consumption. Also, extra power is consumed due to the high activity in the implicit pulse generator [105]. Recent comparative researches [92][106] show that the pulsed FFs are most power (or energy) efficient (50% better than MS FF) in high performance criteria. In power (or energy) critical criteria, the consumption of the pulse structure is less power (or energy) efficient (50% worse than MS FF) [92]. So that, pulse FFs are also more suitable for performance-driven systems [92].

Figure 2.16 (d) is a classic Single Phase Clocked (SPC) FF named: True Single Phase Clock FF (TSPC) [89]. In conventional TGFF, a local clock inverter chain is used to apply the positive and negative phase clock signals. However, SPC FFs remove the internal clock chain and use a single phase clock to control the circuit. The TSPC has a simple structure with only 11 transistors. The benefit of this is that the TSPC is very power and area efficient. Since the TSPC is developed with dynamic logic circuits, the design also has a speed advantage. However, the disadvantage of the TSPC is also clear. Since the TSPC is dynamic, the clock gating technique is not available to the TSPC due to the charge dissipation issue, and the dynamic logic is also less robust than static logic circuits which makes the TSPC is sensitive to process variation [20]. What is more, a non-negligible glitch can be observed on a critical internal node which can lead to state alteration or erode the next stage logics [55]. Also, the TSPC flop is very sensitive to the clock skew. The low slew rate of the CK signal transition can lead to a functional failure of the TSPC [107]. The above-described issue made the TSPC became a historical design [20].

As it was introduced in previous section, TGFF (Figure 2.14) [78] are the de-facto industry standard. However, the vast number of transistors that toggle with clock, places a large clock power overhead even at zero data activity. This problem is compounded by internal clock buffer dynamic power. The need to reduce clock switched capacitance and to eliminate clock buffer in FFs motivates SoA SPC FF designs [65, 97, 108–111] which offer power advantages over TGFF. Figure 2.17 shows claimed power benefits (up

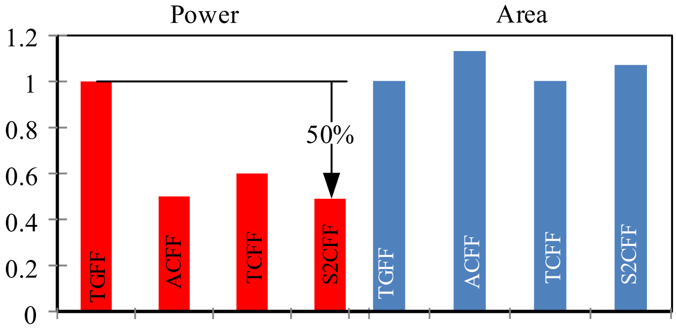


Figure 2.17: Claimed benefits of SoA Single Phase Flip-Flops.[65, 108–110]

to 50% compared to TGFF) in SoA SPC FFs for marginal area costs. Besides the SPC FF, S2CFF, introduced in Section 2.1.4, there are other four SoA SPC is published these years. The schematics of these FFs are present in Figure 2.18.

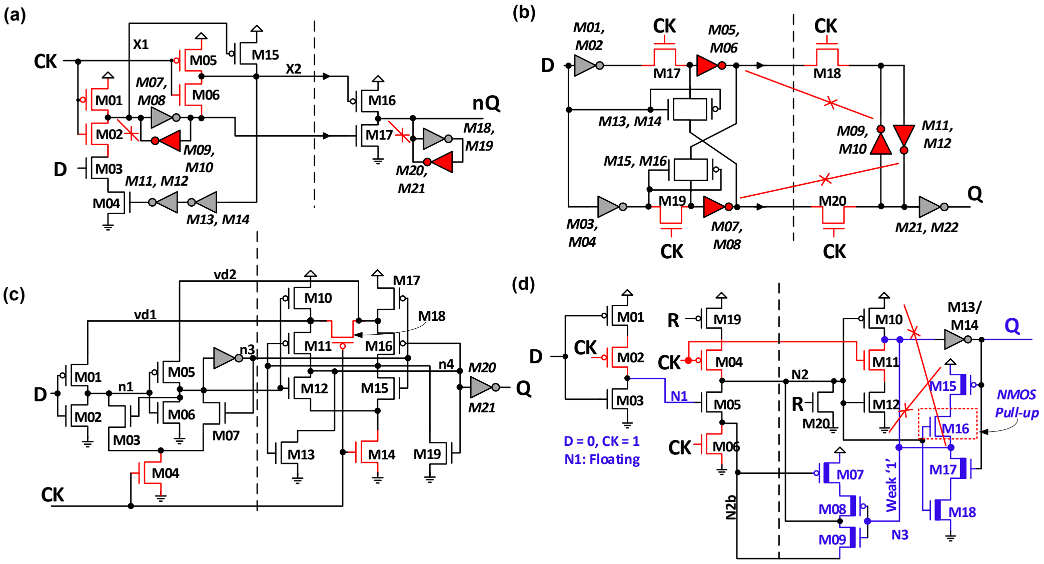


Figure 2.18: Schematic of SoA SPC FFs, highlighting master-slave isolation, contention paths and clock transistors. (a) Cross Charge-Control FF (XCFF) [108], (b) Adaptive-Coupling FF (ACFF) [109], (c) Topologically-Compressed FF (TCFF) [110]. (d) True-Single-Phase-Clock 18T FF (20T with Reset) [111].

In the Cross Charge-Control FF (XCFF) (Figure 2.18a) [108] and Adaptive-Coupling FF (ACFF) (Figure 2.18b) [109], dynamic logic nodes and contention paths are introduced in the design to improve speed. This can, however, degrade robustness when the supply voltage is decreased. Furthermore, the contention current results in extra power consumption during data transitions. In XCFF, the dynamic nodes are indicated as $X1$ and $X2$. Contending devices and nodes in XCFF and ACFF are the highlighted inverters. Although the contention issue in ACFF can be mitigated by carefully modifying the width ratio of transistors in the slave latch, or by adding devices, this results in area and power overheads.

Figure 2.18c shows the schematic diagram of the Topologically-Compressed FF (TCFF) [110]. The design is aiming for minimized total and the clock connected transistor count. It aggressively reduces the number of clock devices to (lowest reported) 3 and total device count to 21. It also successfully achieves contention free static operation. However, owing to the complex topology, there is no area reduction achieved by the reduced transistor count. Recently, a True-Single-Phase-Clock FF with 18 transistors was proposed [111], shown in Figure 2.18d. A dynamic node (N1) and contention paths (pull-up network M15, M16 contend with pull-down network M11, M12; pull-up transistor M10 contends with M17, M18) exist in the design. The FF design was implemented in 28nm FDSOI, which achieved a 40% improvement at 0.4V in energy/cycle compared with conventional MSFF. However, a non-complementary topology is used in its slave latch, i.e. the NMOS (M16) is used for pull-up, which can lead to voltage degradation in internal node N3. For mitigating the voltage drop issue, a poly-bias technique is applied to highlighted transistors. For enabling ultra-low voltage operation, a back-bias voltage is applied to lower the V_{th} of the design, requiring extra design effort. What is more, the output buffer is eliminated which makes the circuit vulnerable to noise at output port Q [37]. Eliminating the output buffer also brings the problem of decreasing fanout. For improving its robustness and increase the fanout of the FF design, an output inverter needs to be inserted. Owing to this, the total transistor count would increase to 20.

From the literature revision of the SoA SPC FFs, it can be considered that various trade-off need to be concerned when applying these designs to NTV operating system. However, there is little doubt that the SPC FF topology can achieve significant power reduction when compare to the conventional MSFFs, e.g. TGFF. Therefore, a in-depth analysis is needed for the SoA SPC FFs.

2.3 Process Variation and Radiation-Induced Errors

In order to understand the main factors which influence the reliability of the integrated circuits and tackle the reliability issue which the low voltages power saving techniques are face, it is essential to do a literature review on the process variation and the radiation-induced errors which would affect the reliability of digital ICs.

2.3.1 Process Variation

With the development in semiconductor technology, the size of each transistor needs to be scaled as small as possible to achieve a maximum integration density level. Driven by the goal of developing ever smaller scaled down transistors, the MOSFET industry has been developed. This has allowed entry into the ultra-deep-sub-micron (UDSM) dimension age, which brings the benefits of lower power and higher speed characteristics.

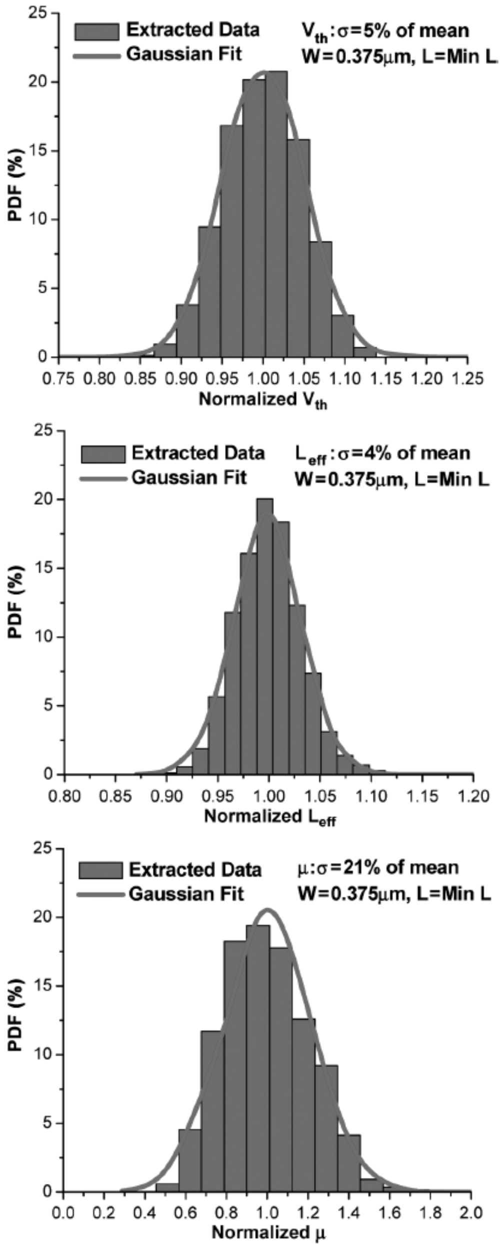


Figure 2.19: Process variation induced parameter variation in Threshold Voltage (V_{th}), Channel Length (L_{eff}) and the carrier mobility (μ) in 65nm technology node test chip with standard deviation (σ) of 5%, 4% and 21% [112].

However, the constant scaling down intransistor size leads to higher process variation [112][113]. This is because the UDSM, which the transistor channel length (L_{eff}) is smaller than 100nm, This has led to increased difficulties in the fabrication process. The size of the UDSM technology node is smaller than the wavelength of the light which is used in the lithography process during fabrication, and this makes it is difficult to control with precision the size and shape of the nano-sized devices [114]. The uneven surface of the devices and the uneven dopant atoms can change the operating characteristics of the devices. The performance, reliability, and power consumption characteristics of the devices can become unpredictable and unstable due to this variation [115].

Categorizing the source of the process variation can help with design. Variation is caused by both intrinsic and extrinsic factors. Random atom placement is classed as an intrinsic factor in process variation. On the other hand, an extrinsic factor would include the difficulties and limitations resulting from an extremely high precision manufacturing process [116].

The source of the process variation can also be categorized as random factors and the systematic factors [116]. The systematic induced variation such as the predictable variations in dielectric thickness across caused by the chemical mechanical polishing variation [117]. The variation due to the random fluctuation of atoms or the line-edge roughness can be categorized as the random variation and the randomness is considered as the primary source of the process variation [113][115].

Process variation would directly cause parameter variation of transistors and it can further change the electrical properties of transistors. The parameters which would be influenced are the thickness of the oxide (t_{ox}), the threshold voltage (V_{th}) and the channel length of transistors (L_{eff}) [118]. These parameters would directly change the electronic properties of the transistor, a chain reaction would be triggered and this would consequently make the integrated circuit cannot work in proper condition. More specificity, the variation in transistor L_{eff} and the V_{th} can lead to a variation in leakage power consumption and operation frequency. The literature [44] provides the research results about the impact of parameter variation on circuits and the results show that the 30% variation in chip frequency and an increase of 20× for variation in the leakage of power consumption which is induced by the process variation.

Figure 2.19 shows the measurement results of the process variation induced parameter variation in Threshold Voltage (V_{th}), L_{eff} and the carrier mobility (μ) in the 65nm technology node test chip. It can be seen that the V_{th}, the μ and the L_{eff} variation are all in the form of a Gaussian distribution. The standards deviation (σ) of the V_{th} is 5%, σ of L_{eff} is 4% and the μ is with the σ of 21% [112].

Also, the process variation in interconnected wires also needs to be considered. The changes in height and width of wires can lead to the delay variation between gates and blocks [116].

2.3.2 Radiation Induced Errors in Electronic Devices

The particles radiation-induced errors in electronic devices can generally be categorized as hard errors and soft errors in general.

Hard error is when there is a destructive single event effect which has been observed in the past four decades [116][119]. Compared to the recoverable soft errors, hard errors are much more harmful to ICs. These hard errors can lead to permanent degradation or the catastrophic conditions within the circuit [120]. Some typical hard errors are enumerated as below.

- ***Single-event Latch-up(SEL)***: SEL mainly occurs in bulk CMOS technology. When the trigger event is removed, the low impedance path of the PNPN four-layer structure device between the supply rail and the ground might remain. This can lead to a short-circuit current with a high magnitude which can vaporize the metal trace and the possibly melting silicon. This troublesome hard error is not only observed in ground level systems but also in space-based systems [119][121].
- ***Single-event Burnout(SEB)***: SEB failures are mainly observed in power MOSFETs, bipolar transistors and IGFETs which are used in space-based systems. It occurs during the power MOSFETs breakdown which is caused by the heavy ion passage. The high current which is induced by the SEB leads to the destructive failure of the device [119].
- ***Single-event gate rupture(SEGR)***: SEGR is a condition of MOSFET gate being isolated and the failure of the channel region. It mainly happens in analog and digital MOSFET integrated circuits, the power MOSFETs, the power transistors and non-volatile memory devices. And it often can be observed with the SEB phenomenon. Along the ion strike path, carriers will increase the local temperature which would lead to conduction increasing. This would further result in the current and heat increasing. Permanent damage of the full dielectric stack might happens as a result [119].
- ***Single-event snapback(SESB)***: The hazard of SESB is similar to the previously introduced SEB error. It is also known as the single transistor latch-up which can happen in SOI devices and the bulk MOSFETs devices. The SESB is caused by the breakdown of drain-to-source in n-FETs. A high current would be generated and lead to a device burnout[119].

Data from NASA's report [122] shows that nearly 45% of spacecraft anomalies are caused by the radiation. Among the radiation-related anomalies, the UPSET phenomena are reported as the majority failure which reaches 80%. This type of radiation-induced function failure is firstly recorded in 1978 by Intel. These errors were observed on the 2107-series 16KB DRAMs and it was accepted that these errors are caused by the α particles radiation traces. Such types of errors were categorized as a 'soft error' [120][123].

Figure 2.20 illustrates the processes where the ion strikes occur. In Figure 2.20 (a), it can be seen that the high energy ion hits the silicon substrate. Once the radiation event happens, a track of electron-hole pairs are formed with a cylindrical shape along the tracks of the ion that is hit. And this process is called the ionization process.

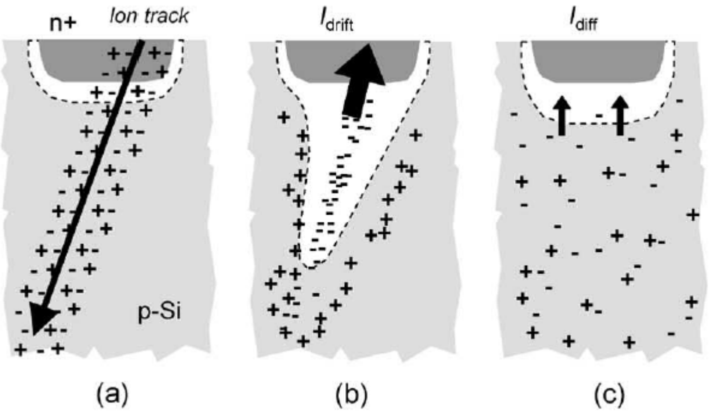


Figure 2.20: (a) Onset of the heavy ion hitting the silicon substrate; (b) Drift collection process/prompt charge collection. (c) Ion diffusion process [124].

After the ionization process, the electric field in the depletion region collects the drift, shown in Figure 2.20 (b), this process is called the drift collection process or the prompt charge collection process [124]. During the drift collection process, a transient current is generated along the track of the ion drift.

Following the drift collection, the diffusion process stars to dominate the ion drift collection which is shown in Figure 2.20 (c), and this process is called the ion diffusion process. During this process, the remaining carriers diffused into the depletion region, the internal electric field will be recovered to the same stable state that existed prior to the ion strikes, this process can last up to hundreds of nanoseconds [124].

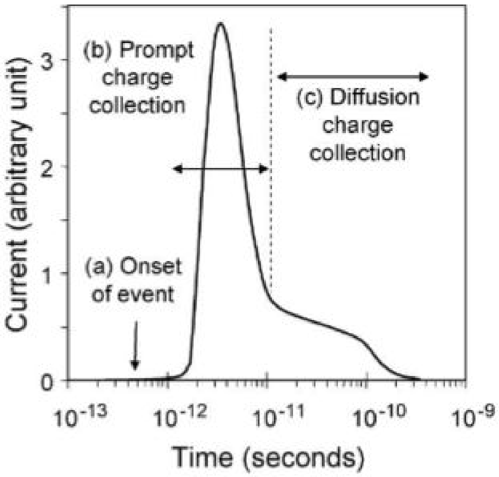


Figure 2.21: The transient current pulse is generated due to the ion strikes [124].

The Figure 2.21 illustrate the transient current pulse which is generated due to the ion strikes, the x-axis of the figure is time and the y-axis is the amplitude of the current pulse. The transient current model can be fit in the form of a double exponential function which is shown as in Equation 2.7 and Equation 2.8.

$$I(t) = A_0(e^{-\frac{t}{\tau_f}} - e^{-\frac{t}{\tau_r}}) \tag{2.7}$$

$$A_0 = \frac{Q_{total}}{\tau_f - \tau_r} \tag{2.8}$$

Where the A_0 is the amplitude of the transient current, Q_{total} is the total charge, the τ_f is the time constant for fitting the charge collection process of the MOSFET, the τ_r is the time constant for deposition process [125][126].

According to the Equation 2.7 and 2.8, it can be seen that the minimum charge at the Sensitive Node can be indirectly measured by finding the minimum amplitude of the current pulse which alters the state of the device, i.e., the $Q_{critical}$.

The phenomenon of charge deposited could happen in combinational logic and sequential logic systems. Accordingly, the errors which happen in combinational logic circuits are referred to as Single Event Transient (SET). This might corrupt the data and the corrupted data would further be captured by memory logic (e.g. a Flip-Flop). The errors which happen in sequential logic are defined as the Single Event Upsets (SEU) and such errors can alter the captured data into an unexpected state.

To quantify the minimum deposited charge on an internal circuit node which leads to the state alteration, the critical charge ($Q_{critical}$) is defined. Also, the critical charge is considered as one of the metrics that describes the soft error sensitivity of the circuit [127][35].

The Soft Error Rate (SER) is defined as the average number of bit-flip events during the particle hitting period. The value of SER can be presented in the unit of FIT (Failure In Time) where 1 FIT equal to one soft error event in 10^9 hours. The SER model for any type of circuit is shown as Equation 2.9 [128].

$$SER = \frac{IN_{flux}}{T} \sum_{n}^{nodes} A_n \sum_{i}^{Q} P(Q_{i,n}) \Delta q \sum_{t_{inject}}^{T} upset_{j,i,n} \Delta t \tag{2.9}$$

In Equation 2.9, the parameter IN_{flux} is defined as the intensity of the particle flux. T is the cycle time, A_n is the area of the drain in node n, To describe the probability of the charge (Q_i) collection which is induced by the particle hitting in node n, the parameter $P(Q_{i,n})$ is used. And the $upset_{j,i,n}$ describes if the state of the node n of the circuit was flipped by the Q_i at the specific time of t_{inject}. The parameter $upset_{j,i,n}$ would be 1 if the condition is met, otherwise the parameter $upset_{j,i,n} = 0$.

The SER can also be modelled as it is shown in Equation 2.10 [129][130].

$$SER = IN_{flux} \cdot A \cdot K \cdot exp(\frac{-Q_{critical}}{Q_s}) \tag{2.10}$$

Where the IN_{flux} is the intensity of the particle flux, and it depends on the device working environment. K is the calculation fitting parameter which is a constant number for the given technology node, and the parameter A is the area of the drain in MOSFET. Q_s is the efficiency parameter of the charge collection process. $Q_{critical}$ is the critical charge of the Sensitive Node, and this can be tested by utilizing the previously introduced method.

The soft error is considered as a temporary error, and the Error Recovery Mechanism can recover it. Soft errors also are considered to be possibly harmless because of the masking phenomena which make the error would not be propagated or latched by the following next stage FFs [131].

- ***Logic Masking***: This would happen when there is no sufficient path for propagation between the Sensitive Node on the circuit and the output node.
- ***Electrical Masking***: This would happen if the transient current is weakened after propagation through a series of logic gate, and the magnitude is not able to alter any correct data.
- ***Temporal Masking***: This would happen if the transient current pulse misses the timing constraints of the next stage FFs. Accordingly, the downstream FFs are not able to latch the error data around the CK edge in such condition.

2.3.3 The Error Tolerant Circuits

The Dynamic Voltage Scaling (DVS) is a power-aware computing technique, and it is a system level supply voltage tuning mechanism that decreases or increases the supply voltage accordingly [132]. To achieve the lowest power consumption, the voltage needs to be scaled down as low as possible. However, the reliability of the digital design is also decreases due to the circuit functional verification, the timing verification, and the power distribution integrity [55].

This section review varies the state-of-art of the error aware techniques. Firstly, one of the dominant error detection and correction technique, named Triple Modular Redundancy (TMR) will be introduced. Following that, the Dual Modular Redundancy (DMR) architecture with correction function and the Razor-type error tolerant circuits are introduced.

Triple Modular Redundancy (TMR) architecture is a logic redundancy technique which is generally used for SEU mitigating. This technique can be implemented at gate level, Register-Transfer level or higher level [116]. Figure 2.22 shows the architecture triplicates the main sequential logic block with a majority voting circuit being connected for outputting the correct data. Once the state of one module among the three modules is

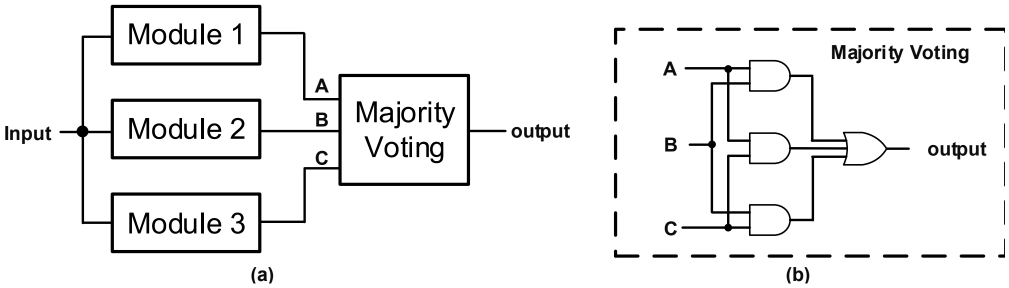


Figure 2.22: (a)The block diagram of the TMR architecture [116]. (b) The gate level schematic of the TMR Voting Block..

flipped due to the soft error, the majority voting circuit still outputs the correct data. This, however, results in an overhead over 200% to a standard sequential logic block in both power and area [116].

The Majority voting block is implemented by a four gates combinational logic block. The gate level schematic is shown in Figure 2.22 (b) and it can be easily applied to the RTL network [133]. However, TMR brings three times more power consumption and area overhead to the original block. Another issue is that the TMR cannot address the scenario of Multiple Bit Upset (MBU) [134], i.e., two modules data are flipped due to the soft error at the same time. In the scenario of multiple bit errors, the majority of the data among the three modules is the corrupted data, and the output via the Majority Voting block will be in the wrong state. The TMR architecture also faces the potential problem of error accumulation. If the present error in one of the modules has not been removed, the next error in another module might be accumulated with the previous errors which creates a multiple module error scenario. In this scenario, the Majority Voting block would be faulty and produce corrupted data.

Dual Modular Redundancy (DMR) architecture was proposed to detect the SEU error. The block diagram is shown in Figure 2.23(a). This design duplicates the main FF, and an XOR logic gate is added to detect the error when the SEU corrupts one of the FFs. The DMR naturally imposes less of a power overhead, and an area overhead compared to the TMR design. However, the DMR architecture can only detect the SEU error and is incabable of correcting it, since it cannot identify the SEU corrupted FF. Inspired by the conventional DMR architecture, a new DMR was proposed for not only detecting the SEU error but also correcting the error in FF [136]. It relies on the fact that a previousFF holds the correct state before the SEU event occurs in one of the FF to correct the SEU error in the FF and this can be observed from the timing diagram of a conventional DMR circuit which is shown in Figure 2.23(a). The DMR with an Error Recovery circuit duplicate of the main FF and uses a delay path and a latch to detect and correct the SEU error, the gate level schematic is shown in Figure 2.23(b). When there is no SEU in FFs, the states of both FFs are same and the signal *Error* is 0. Since the added D-Latch is active low, the latch during the period of $Error = 0$ is transparent. Once the SEU event corrupts one of the FFs, the signal $Error = 1$ which lead the latch

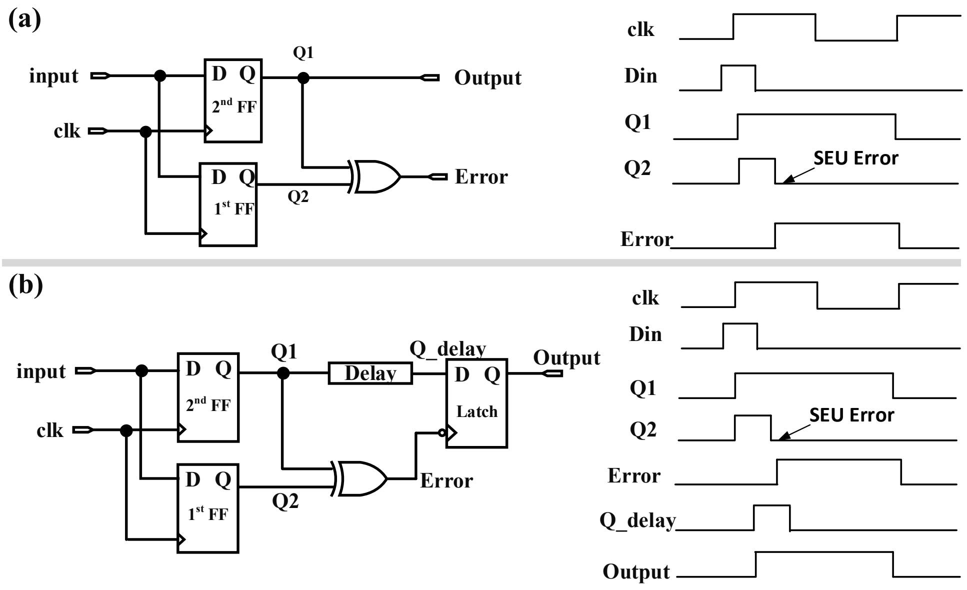


Figure 2.23: (a) Gate level schematic of the Dual Modular Redundancy (DMR) Architecture and the Timing diagram of the DMR circuit [135]. (b) The DMR circuit with an error recovery function and the Timing diagram of the DMR circuit with an error recovery function [135].

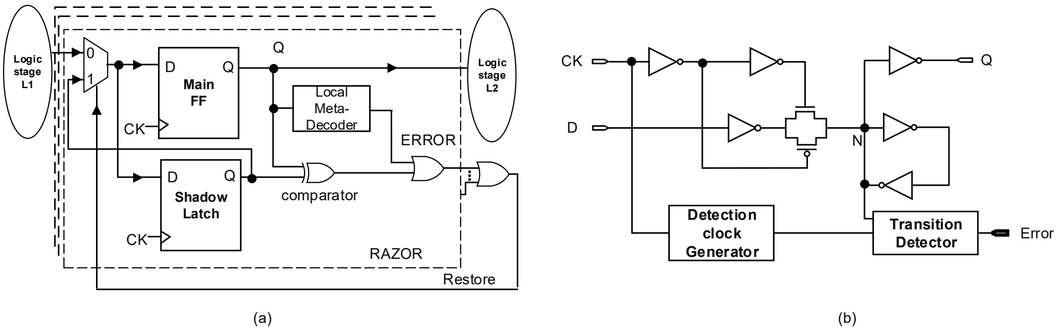


Figure 2.24: The structure of the (a) Razor FF [137], (b) RazorII FF [138].

holds the current state which is the delayed data of the second FF. According to the operation principle of the DMR technique, it can be known that the Latch will always output the delayed version of the second FF. So that, the latch always outputs the data without SEU. The timing diagram of the DMR circuit with error recovery function is shown in Figure 2.23(b). As recorded, the DMR with an error recovery function uses 70 transistors in total. Compared to the 101 transistors equivalent TMR architecture, the DMR saves 30% in total transistor count. Due to fewer transistors being used in DMR compare to the TMR, the DMR saves 38% less power overhead on average. Comparing the DMR and the TMR regarding Clock-to-Q, it was recorded that the DMR increases the delay by 10% compared to TMR. This is due to the delay block which is added into the circuit [136].

The original Razor was proposed for detecting the timing errors while the Dynamic Voltage Scaling (DVS) is applied. With several years of development, various types of Razor-type error tolerant FFs [137–140] were proposed. By employing the Razor in-situ error detection and correction mechanics, the supply voltage that causes the proper first failure point (lower than the conventional DVS margin) can be observed. Therefore, the supply voltage can be further scaled down to break through the lower limit of traditional DVS voltage to achieve further power (or energy) efficiency.

In Razor FF (see Figure 2.24(a)), the error detection is implemented by comparing the content of the main FF with the data in the 'Shadow-Latch'. If the main FF output Q is different from the 'Shadow-Latch', the Error signal will be flagged. The Error signals from different Razor FFs will be 'OR' together for outputting the 'Restore' signal to reload the correct data from the shadow latch to correct the error data. However, limitations of Razor FF have also been reported [141]. The complex topology and redundant shadow latch brings a large area overhead to conventional FF design. Also, the ERROR signals are connected to the shared OR gate and the output signal 'Restore' signal is feedback to each Razor FF which makes the 'OR' gate a high fan-in and high fan-out structure. Buffers need to be inserted to the 'Restore' signal lines otherwise the data in 'Shadow Latch' cannot be reloaded to main FF before the next CK rising edge. This 'restore' timing issue is significant when the number of critical FFs is increasing. What is more, the Razor is not suitable for soft error detection and correction.

Different from the original Razor FF circuit, which has the function of error-detection and self-correction. The RazorII [138] (see Figure 2.24b) only has the function of error detection. The error correction function is implemented by architectural level re-executed processing. RazorII accepts the positive phase level-sensitive latch to implement the function of FF which brings the benefits of a reduction in total transistor count.

From the Figure 2.24(b), it can be seen that the RazorII can generally be separated into three sub-blocks, which are the 'D-latch block', the 'detection CK Generator' and the 'Transition Detector' (see Figure 2.25(a)). The RazorII has a small period which disables the function of Transition Detector in each clock cycle. This transition detector disabled period was defined according to the valid transition time constraints and the transition delay from the input D to the output Q of the latch. The Transition Detector block samples the value at N node. During the transparent phase, the N node can be altered by the late incoming data, i.e., the timing error. Once the data transition happens after the time margin that was defined, the Error signal will be flagged by the 'Transition Detector'.

The Figure 2.26 (a) illustrates two typical scenarios in the RazorII detection procedure. The detection CK generator works as a pulse generator to generate a negative pulse according to the time margin, which means the transition event has to meet the Setup

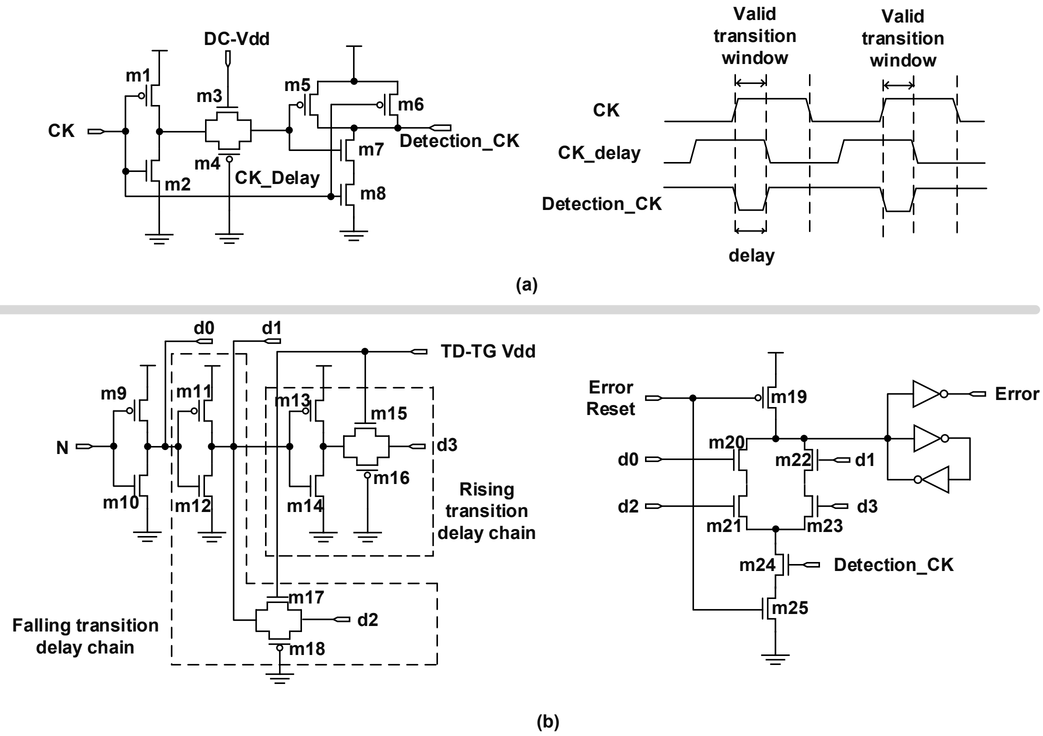


Figure 2.25: (a) Razor II detection CK generator and waveform (b) Transition Detector [138][141] .

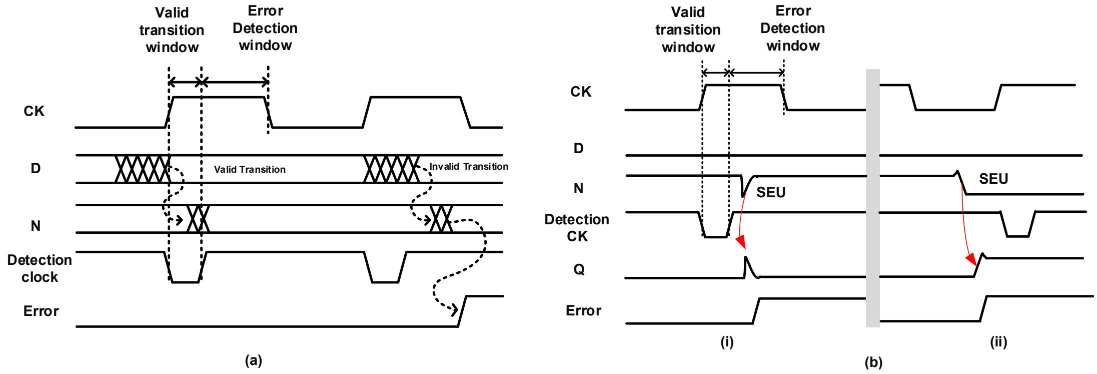


Figure 2.26: (a)Razor II error detection waveform. (b) Two scenarios of RazorII detect the SEU error [138, 141]

time and Hold time constraint. The detection CK disables the detection function to allow the valid data transition. After this, the detection CK switches to logic 1, the Transition detectors are active, any transition at during this region is invalid. If the invalid transition is detected, the Error signal will be generated. System pipeline reply mechanism will be triggered. Naturally, the transition detector of the RazorII has the function to detect the invalid transition either the error is categorized as Timing error or SEU. Figure 2.26(b) shows two scenarios of RazorII detect the SEU error. The Figure 2.26(b−i) shows the situation that the SEU error happens at the transparent phase of the RazorII FF. It can be seen that the transition detector detects the SEU error when the detection CK is logic high. The Figure 2.26(b−ii) shows a situation where the SEU error happens at the latch phase of the RazorII FF. During this phase, the transition detector is still enabled. So that, the SEU error at the negative phase of the clock is can still be detected and flagged.

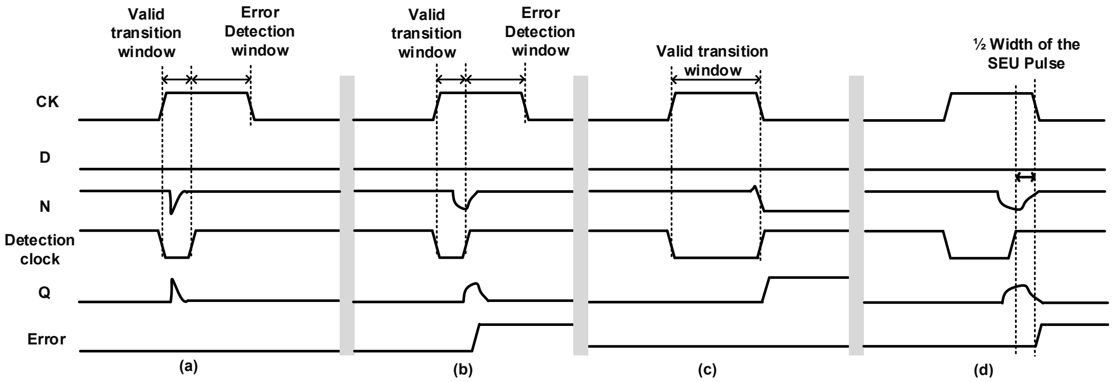


Figure 2.27: Four possible situations where the SEU error occurs during the negative phase of the detection CK [138].

Since the SEU error could happen at any time during each clock cycle, including the negative phase of the detection CK signal, i.e., the transition detector is disabled. The Figure 2.27 shows four possible situations that the SEU happens when the detection CK is low. Figure 2.27(a) shows a situation where the SEU happened at node N after the falling edge of the detection CK and recovered before the rising edge of the detection CK. In this situation, the transition detection is not able to any transitions since it is disabled and the Error signal is not flagged. This situation was considered as benign since the state of RazorII FF is not corrupted [138].

Figure 2.27(b) shows the situation where the SEU occur at node N just before the rising edge of the Detection CK and it recovers after the rising edge of the detection CK. During the recovery time of the SEU, the transition detector has been enabled. Therefore, the SEU in this situation will be detected, and the Error signal will be flagged. Figure 2.27(c) shows a special case which is the width of the valid transition window is equal to the high phase of the clock. Figure 2.27(c) shows the SEU error occurs just before the rising edge of the detection CK at the sensitive node N. While the state has been completely corrupt during at the latch phase of the Razor II. Since there are not any transitions during the transition detector enabled period, the SEU event is not detected and the Error signal is not flagged.

In order to address the detection failure of the transistor detector in such a situation, i.e., shown in Figure 2.27(c), it was suggested that the designer need to make sure the Razor II provides enough time to let the SEU error recovered before the falling edge of the clock. In order to make sure the transition detector works properly, the clock signal should be high for at least half of the SEU pulse width after the rising edge of the detection CK signal. The Figure 2.27(d) illustrates the time constraints of such a situation [138].

2.4 Concluding Remarks

There is little doubt from this literature review that the voltage scaling is one of the most efficient methods to reduce the power (or energy) consumption. However, it creates the conflicts in performance degradation and power (or energy) consumption. It also raises the trading between the static and dynamic consumptions. Researchers have proved that the DVS could not achieve further energy reduction when V_{dd} is below $V_{E_{min}}$. Static power (or energy) will be increased (higher than the dynamic power consumptions) and dominate the overall consumptions. Indeed, the E_{total} will be increasing when V_{dd} is below $V_{E_{min}}$.

Circuits that operate at $V_{E_{min}}$ - V_{th} are defined as the STV operating circuit. However, the STV circuits need to be designed with increased transistor size for tolerant the process variation and the large degradation in performance. NTV techniques, a potential solution for implementing ULP applications in IoT, have been demonstrated to reduce energy consumption significantly by decreasing the supply voltage to approach the V_{th}, while preserving the favorable variability and performance characteristics compared to the STV operation [18]. However, the NTV techniques are still facing challenges before the relevant devices can reach widespread production and use.. Two of the main barriers to NTV design are variability and performance loss. The propagation delay increases and the high sensitivity to parameter variations leads to functional errors or failure in the NTV computing. As it was shown in Figure 2.6 shows that the soft error rate significantly increases in the NTV region. To improving the NTV technique, researchers have been continually making contributions to the design over the past few years. As it was introduced in Section 2.1.4, some designs were developed to address current NTV design barriers and the recent achievements in NTV techniques provide more opportunities for developing this approach.

For a better understanding of NTV design challenges, literature related to general digital circuit designs has been reviewed. From the literature, it can be seen that sequential logic circuits which introduce data processing sequences into the system, are considered as one of the most crucial aspects of the design. For today's typical SoC, millions of FFs as CLME can be applied to the whole system [74]. The proportion of power (or energy) of FFs can be up to 50% of the overall power (or energy) budget of the chip [70]. For limited given power (or energy) budget system (i.e., the IoT leaf nodes devices), the power reduction in FFs can be considered as a crucial argument for including them in the design. Considering the design challenges in NTV, the design or selecting of FFs is even more important. In section 2.3.3, various types of structure (MS structure, DET structure, differential structured, pulsed structure and SPC structure) FFs were reviewed. It is clear that most of these designs are aimed at providing better performance (regarding speed) and most of these designs are not suitable for low power consumption operations. For low power, SPC FFs show great potential for inclusion by low power system designers. Since the SPC FFs can be clocked with single-phase clock signals, no internal clock inverters are required. The benefit of this is that a large portion of dynamic power consumed by these clock inverters can be reduced. In recent years, through intensive research and development in SPC FFs, new SPC FFs [65, 97, 108–110] are designed for the low consumption devices.

Also, for a better understanding of the main factors which influence the reliability of the circuits, literature discussing process variation and radiation-induced errors has been reviewed. Also, error tolerant circuits (the TMR technique, RazorII, the DMR with error recovery function) were viewed. As a gate level error-resilient design, the TMR circuit has various advantages over other circuit-level designs. It can be adequately implemented by employing standard logic cells which are comparatively easy to be synthesized as opposed to any circuit level or transistor level modifications. It can not only detect the SEU error but can also correct the error via a Majority voting block. However, there are also some limitations to TMR design, such as the Multiple modular errors issue, and the error acceleration issue. Also, the redundancy modular and the added Majority Voting block is reported to lead to an overhead of over 200% in both power and area compared to the standard logic cell. Different from the previous versions of Razor (which has limitation in soft error tolerance), RazorII simplifies the transistor level circuit, and it is only used in error detection and flagging the error (including soft errors). The correction process is done using a system level approach. However, some limitations exist in RazorII. One of the limitations is the power consumption required by the detection CK generator is proportional to the clock frequency. So that, the trade-off between power consumption and circuit performance needs to be considered during the design. The DMR with recovery function circuit overcomes the limitation of the previous DMR circuit in error correction. Based on the previous DMR circuit, a delay element and an $Error$ signal controlled latch is added to eliminate the output of incorrect data. This achieves 30% in the reduction of the total transistor count compared to the equivalent TMR circuit, along with an average of 38% reduction in power consumption compared to the TMR. However, the DMR was recorded to have a 10% increase in CK-to-Q delay compares to the TMR circuit which considered as performance penalty.

In light of the findings from this literature, this research carried out to answer the research questions in Chapter 1, aims to investigate a method to achieve power(or energy) reduction by adopting the NTV technique, while simultaneously maintaining the robustness of the sequential logic circuits, toleration of process variation and resilience against errors with the minimum performance penalty and area overhead The next chapter present a further in-depth analysis to the SoA SPC FFs, which answers the first research question of this thesis.

Chapter 3

Analysis and Development of SPC FFs for NTV applications

From the survey in previous chapter, it can be considered that circuit and system design for the Internet of Things (IoT) sensor nodes often benefit from reduced speed requirements. The most important criteria for such applications is minimum energy operation. Further, sensor applications are heavily duty-cycled requiring aggressive leakage mitigation techniques [142]. Voltage scaling provides the ideal knob for minimizing energy while exploiting timing slack. However scaling supply down to sub-threshold levels makes the design vulnerable to On-chip Variation (OCV), lowering yield and operational temperature range especially in memory and latch structures which inherently rely on robust pull-up and pull-down paths where aggravated mismatch induces functional failures [72]. In contrast, designs operating at NTV benefit from significant energy reductions without having to tackle serious OCV issues, making this regime of operation more suitable for industry adoption [18] [61].

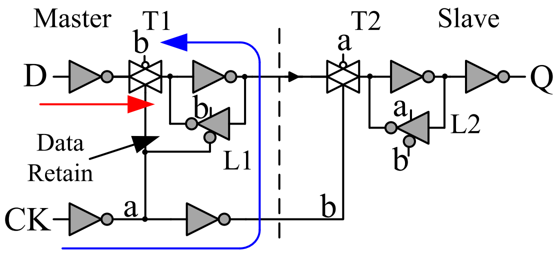


Figure 3.1: The widely used TGFF, 24 total transistors, 2 internal inverters included. The TGFF has relatively simple datapath and retention scheme.

A significant portion of system power is expended by clock-tree and sequential logic. FFs in particular are reported to have a significant impact on the power, performance, and robustness of digital systems [143]. Recent research has focused on developing lower power and reduced variability FF circuits specifically targetting low voltage operation

[110][65]. TGFF (Figure 3.1) [78] are the de-facto industry standard. However, places a large clock power overhead even at zero data activity owing to the vast number of clock toggled transistors. What is more, TGFFs are considered less optimal for low-voltage applications because transmission gates present source-drain leakage paths on nodes that are weakly held at low supply voltages [142]. On the other hand, TGFFs avoid stacked structures except in the keepers which are, by design, meant to be weak. Hence TGFFs remain open for use in NTV designs. The large clock power overhead issue is compounded by internal clock buffer dynamic power. The need to reduce clock switched capacitance and to eliminate clock buffer in FFs motivates SPC FFs designs which offer power advantages over TGFF. The brief review of these SoA designs are presented in Chapter 2.

In this Chapter, prominent types of SoA SPC FFs are in-depth analysed in suitability for NTV operation. SPC FFs are analysed and based on a preliminary analysis, two designs, which meet all NTV circuit design requirements are further investigated. These SPC FFs are designed for NTV operation in TSMC 65LP and compared against the classic transmission gate FF (TGFF). Cell-level design issues and variation are explored in the context of a 5000 gate AES encryption macro. Key design issues are identified, which erode the claimed benefits of SPC FFs when implemented as part of a larger design. We conclude that aggressive reduction in FF clock loading offers benefits but can lead to functional failures when the OCV is considered, especially at NTV. Given the theoretical benefits of SPC FFs for enabling IoT, the need for further work on SPC FF designs is highlighted. In this part, SPC FFs are evaluated in the context of NTV applications and highlights the complexity associated with these designs. Further, the results presented, show that the claimed benefit of these designs reduces significantly as yield, synthesis and system level issues are addressed.

Based on these observations, a modified version of the TCFF, called TCFF-NTV, is proposed for NTV operation. In the proposed TCFF-NTV circuit, the design limitation in the original TCFF is addressed. The result shows the TCFF-NTV improved the previous TCFF in yield by 95.4% at NTV. Compared to the conventional TGFF based chip design, the TCFF-NTV achieves 65% less register power with only 4% chip area overhead, and the performance is maintained. The details about the proposed TCFF-NTV is introduced in Section 3.5.

The key contributions of this chapter are:

1. Highlighting design issues in published SPC FFs.
2. EDA and system level issues with SPC FFs.
3. Comparison for deriving NTV SPC FF specifications.
4. Proposing a new low power Flip-Flop for NTV, named TCFF-NTV.

This chapter is organised as follows. The Section 3.1 reviews SoA SPC FFs to short-list designs appropriate for NTV. The design limitation of these SoA SPC are highlighted, simulation results are also presented in this section for illustrating the impact of the design limitations. Section 3.2 presents the in-depth discussion of the cell-level implementation of these SPC FFs highlighting three complex design issues that are not applicable to TGFFs. Section 3.3 presents an extended comparative AES-128 block-level benchmark using the two most promising SPC FFs and TGFF baseline to understand EDA level issues. The results show a progressive reduction of claimed benefits at each design stage motivating the need for further research in SPC FF designs. Based on observation refer to the SoA revision, a new SPC FF design, named TCFF-NTV, is proposed targeting the NTV operation. The design approach, analysis of the circuit and comparison results are present in Section 3.5. The discussion and the concluding remarks of this chapter are in Section 3.6.

3.1 Analysis of State-of-the-Art SPC FFs

A key feature of all SPC FFs compared to TGFF (Figure 3.1) is the significant difference between the master and slave latches. Note also that the TGFF has 12 of 24 transistors connected to nodes that toggle with clock. In contrast, the S2CFF [65] uses only 5 clock devices while also eliminating the clock buffer (Figure 2.10). This reduces the switching capacitance resulting in significant power reduction. This design further eliminates all contention paths and allows full static operation by trading-off total device count.

On the other hand, the cross charge-control FF (XCFF) [108] and adaptive-coupling FF (ACFF) [109] permit contention paths and dynamic nodes in the design for gains in speed. Conventionally, dynamic logic is introduced to achieve better speed [144]. However, dynamic logic is vulnerable to process variation, and making dynamic circuits less robust at NTV [65], so Fully-Static Flip-Flop operation is therefore desirable for ultra-low power designs. In conventional pure dynamic FF design, e.g., TSPC [89], the clock gating technique is not available due to the discharging issue. The SoA semi-dynamic design, XCFF, relief the data retention issue by adding the data retention inverters. However, the robustness of the dynamic logic is still needed to be aware. With ultra-low-power consideration, the data contention issue is also considered as a critical factor. Take the XCFF as the case study.

The dynamic nodes in XCFF are indicated as X1 and X2 and the contending devices and paths in both XCFF and ACFF (Figure 2.18a,b) are highlighted. These designs target minimum clock switching capacitance, requiring only four devices connected to the clock. The XCFF was simulated with HSPICE to illustrate the impact of contention path.

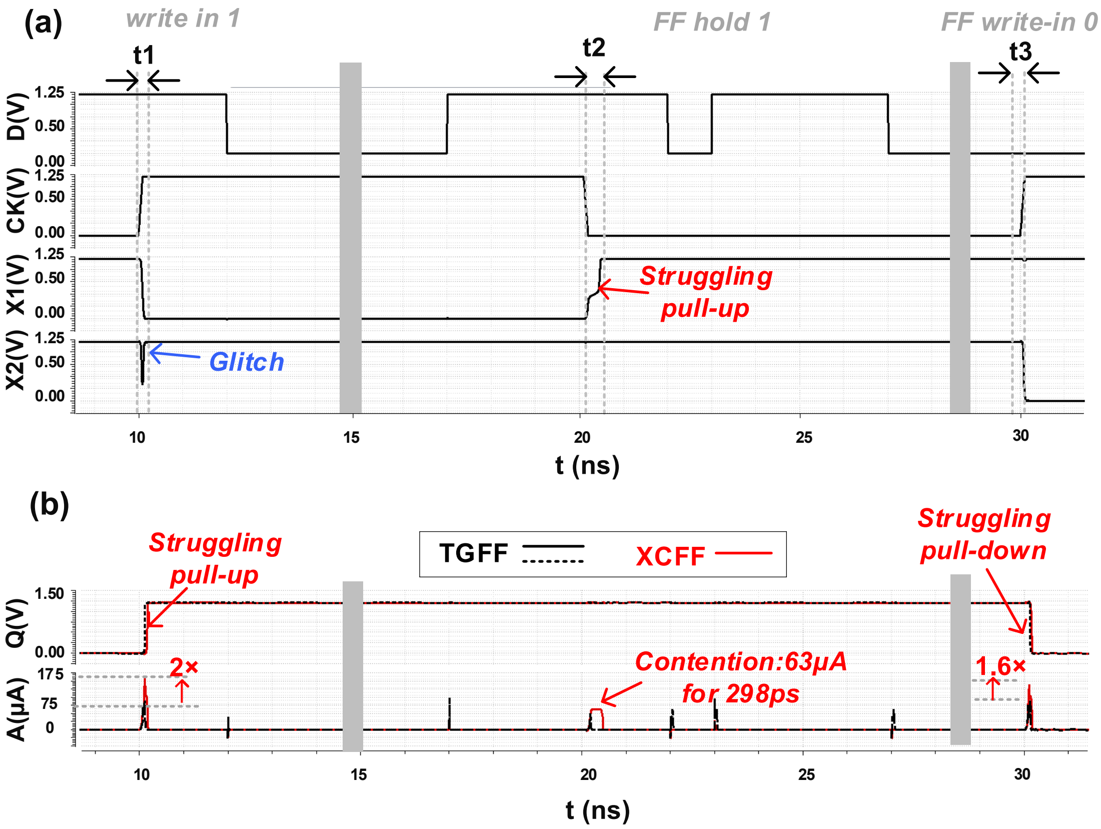


Figure 3.2: SPICE simulation results of the XCFF, illustrating the contention issue in XCFF. (a) operation waveform of the XCFF internal nodes. (b) Output and Current waveform of the XCFF and TGFF.

The Figure 3.2a shows the waveform of the internal node of the XCFF. During the phase of $t1$, a glitch is observed in node X2 and the amplitude of the glitch is higher than $V_{dd}/2$ which means the glitch is non-negligible. By analysing the operation of the XCFF, the glitch issue can be explained. When the CK = 0, the node X1 is pre-charged by the PMOS M01, the logic is retained in back-to-back inverters.

In the scenario where CK is rising and D = 1, FF write-in 1, X1 is supposed to be discharged through M02, M03, and M04. Assume the CK and D are stable, the dependency of X1 pulling down is dominated by the transition of n2 (As long as the n2 rising to 1, the X1 can be pulled down). The latency of n2 rising depends on 2 main factors. The first factor is the delay of the inverter chain (M11&M12 and M13&M14). Another factor, a more dominant factor is the stabilization of X2. For the scenario where CK is rising and D = 1, X2 is supposed to be stable at logic 1 ($V(X2) = V_{dd}$). From the schematic diagram, it can be seen that the X2 can be pulled up when CK = 0 or X1 = 0. When CK = 1, the V(X2) depends on X1 (pull-up via M15) and V(n1) (where V(n1) = X1'). However, according to the analysis of the circuit, it can be seen that there is a small period of time as X1 $\neq$ 0 when CK = 1. M06 is on and X2 is pulled down to V(n1) and V(n1) $\neq$ 1, i.e voltage degradation can be observed. From the simulation result, it can be seen that the degradation is over $V_{dd}/2$, which enable the downstream

PMOS(M16) to operate at the linear region. Later, the n1 is slowly rising. M16 and M17 are both on by non-stable voltage. Short circuit current is increased for to this reason. From the simulation result in Figure 3.2b, it can be seen that the max value of XCFF transient current during the t1 period is 2× that of the transient current pulse of the TGFF.

The worst case for the contention issue in XCFF appears when the FF is holding the 1 (see Figure 3.2a). The contention happens when D = 1 and CK is falling. In retention inverters (M07&M08 and M09&M10) of the master latch, X1 is still 0 when CK = 1. As a consequence, n1 is 1, and the NMOS (M10) is on. As the CK starts falling and M01 is turning on, M10 is not completely off. A logic contention will happen between M01 and M10 (i.e., PMOS and NMOS against each other), a contention current appears. Also, the logic contention slows down the X1 pulling up means the contention phenomenon will last for a period of time. In waveform (Figure 3.2a, t2), it can be seen that the X1 is struggling to pull-up and X1 swings at middle rail (i.e. V(X1) at $V_{dd}/2$) for about 298 ps. More details can be observed in Figure 3.2b. The contention current of XCFF achieves 63 μA for 298 ps. Since there is no contention path in TGFF, the current pulse is transient and manageable.

A similar contention phenomena can also be observed in the slave latch of the XCFF. Contention happens between the PMOS (M16) and the NMOS (M21) when nQ is switching to 1 (see Figure 3.2a, t3). Owing to the contention path, the amplitude of the transient current pulse in XCFF is 1.6 × higher than TGFF.

Owing to the contention path discussed above, the higher power consumption of the FF designs with contention is higher than the TGFF (without contention path). Therefore, the contention path should be avoided when the ultra-low power characteristic is the demand.

The findings about the SoA SPC FFs are summarized in Table 3.1. Since high speed is not critical for IoT applications, contention and dynamic nodes ought to be avoided for power and OCV sensitive NTV operation. Therefore, the remainder of this work focuses on S2CFF and TCFF and analyses relevant design and implementation aspects.

3.2 Evaluation of State-of-the-Art SPC FFs Implementation

Design reuse is particularly attractive for practical reasons, especially for large designs. The significant difference between master and slave latches in SPC FFs makes it difficult to design them when compared to TGFF where the master and slave are usually similar. This section delves into a more intricate design and implementation aspects of S2CFF

Table 3.1: Comparison of state-of-the-art SPC FFs

	TGFF[78]	S2CFF[65]	TCFF[110]	ACFF[109]	XCFF[108]	TSPC-18T[111]
Year	Std-Cell	ISSCC'14	ASSCC-13	ISSCC'11	VLSI'05	TSCAS-I'18
Type	Static	Static	Static	Static	Dyn	semi-Dyn
Contention	No	No	No	Partial	Yes	Yes
Single ϕ	No	Yes	Yes	Yes	Yes	Yes
CK/N*	12/24	5/24	3/21	4/22	4/21	4/18
Demo	N/A	40nm CTC	40nm Exp. logic	40nm CTC	100nm μCPU	28nm FD-SOI CTC

*N: Total transistor count, CTC: Custom on-die test circuit
Text in red: Design limitation
Single ϕ = Single Phase Clock

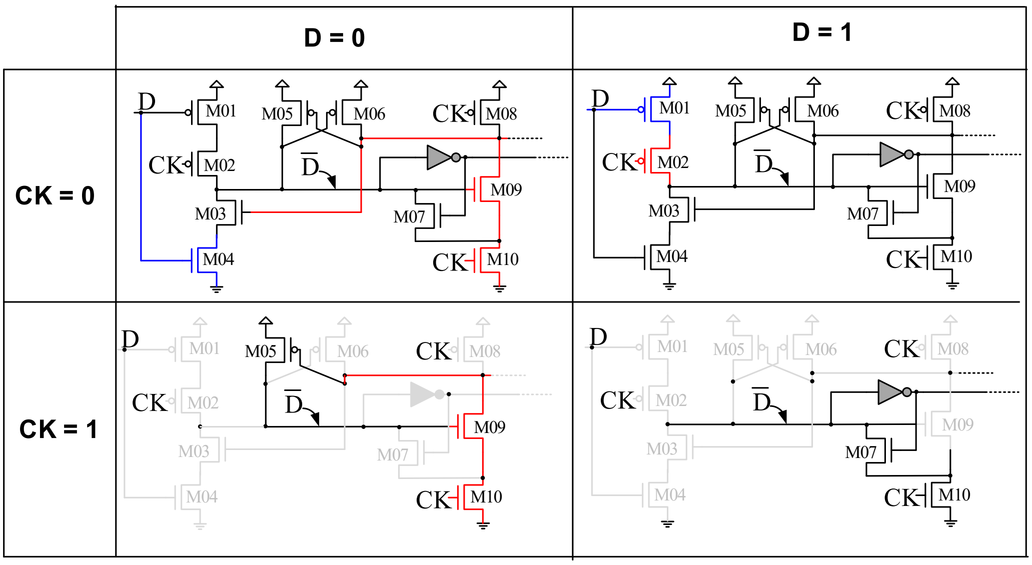


Figure 3.3: Data dependent setup/hold path and retention loops in S2CFF

and TCFF and highlights three key issues which complicate the deployability of these designs.

3.2.1 Mulitple Datapaths and Retention Loops

Unlike TGFFs, internal data path in SPC FFs depends on the state of data (D). This complicates optimizing the FF for setup and hold constraints and limits the potential

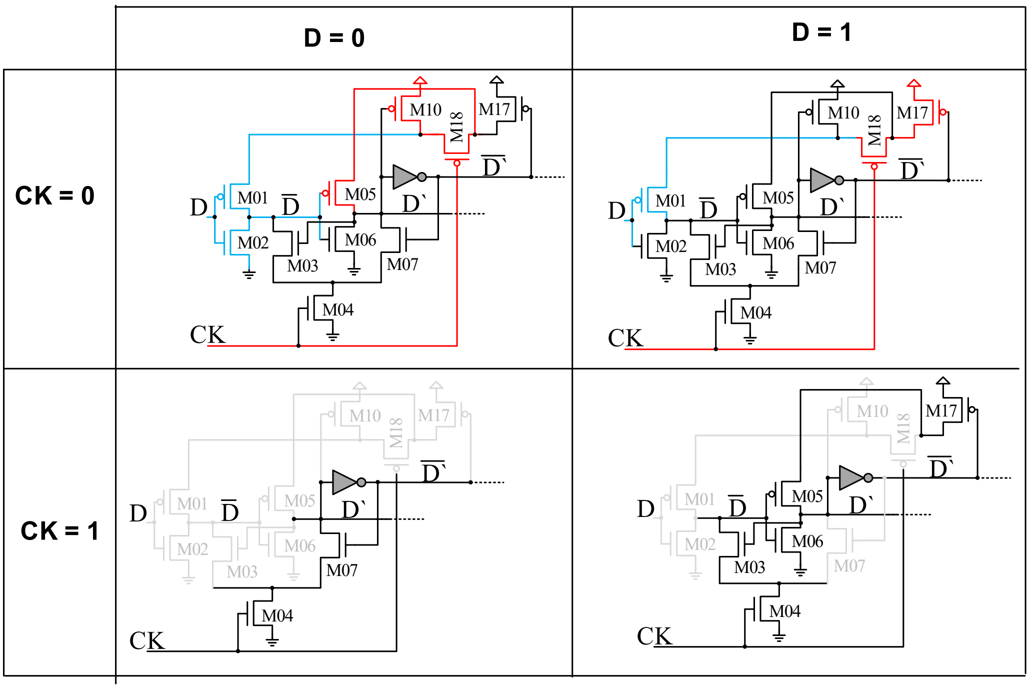


Figure 3.4: Data dependent setup/hold path and retention loops in TCFF

benefits. Figure 3.1 shows the relatively straightforward TGFF datapath and worst-case clock path. When clock (CK) is rising, the worst-case is when D and CK race to the master latch L1. If D wins, L1 captures the new data causing hold violation. If CK wins, T1 successfully blocks D from being latched setting the worst-case setup constraint. Setup issues can be resolved by slowing down the clock, but hold issues can be fatal.

In contrast, S2CFF shows strong data dependency as highlighted in Figure 3.3. When D is transitioning to 1 at about the same time as CK is rising, the race is for D to turn M04 on while CK has to turn M03 off. However, the path to M03 for CK is through M08 and M10/M09. This sets up the worst-case hold constraint. The alternate case of D falling when CK is simultaneously rising presents a relatively trivial case.

When CK is 1, previous data must remain latched. Again depending on the state of D, the latch uses different loops. When D is 0, M09 and M05 form a loop allowing data to be retained as long as CK=1 keeps M10 turned on. For D=1, the inverter in the master and M07 form the retention loop which remains active as long as CK=1. S2CFF offers good speed and substantial reduction in hold time and hold time variation over TGFF [65] which suggests there is a large potential benefit for NTC. Small hold time variation reduces the need for buffer insertion thus further reducing power and improving system yield.

Similarly, TCFF, as shown in Figure 3.4 sees different D and CK race paths depending on D. When D=0 and is rising simultaneously with CK, D can only succeed if it can turn M05 on through M01 and M02. But the clock only needs to turn M18 off which blocks M05 from charging D'. Likewise, for D falling simultaneously with CK rising, D cannot propagate if CK turns M18 off which prevents M01 from charging node $\bar{D}$. This establishes the worst case setup constraint. Further, data is retained by two different loops formed by 1) M05, M06 and M03 for D=1 and 2) master inverter and M07 for D=0.

3.2.2 Master-Slave Interface

All SPC FFs suffer from tightly coupled master-slave interaction compared to the TGFF. As shown in Figure 2.10 and Figure 2.18 the number of interacting paths increases as the total device count and clock transistors are reduced. For example:

1. TGFF: 12 CK devices, 1 data, 2 clock.
2. S2CFF: 5 CK devices, 2 data, 1 clock.
3. TCFF: 3 CK devices, 2 data, 1 clock, 2 interface.

The multiple interface paths complicate the design because any delay mismatch between these signals can result in incorrect operation. As will also be shown later in this section, this also presents layout complications which must be factored in at schematic design time. Of the selected SPC FFs, TCFF has worse interaction related design problems forcing sub-optimal transistor sizing.

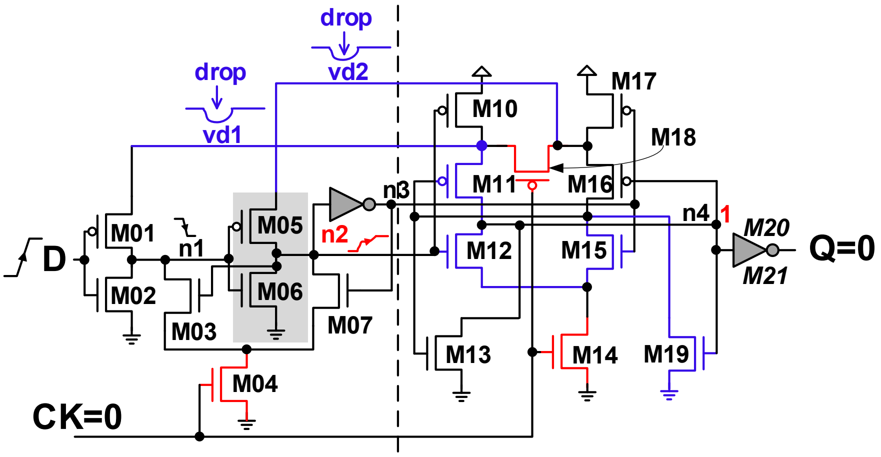
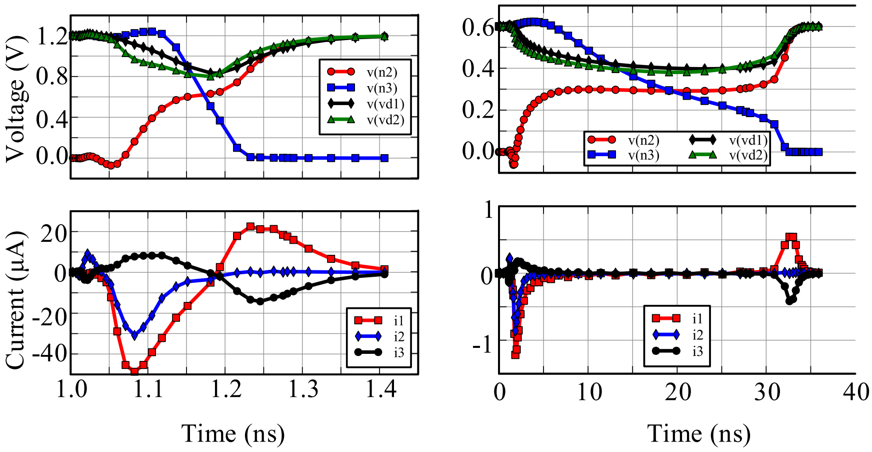


Figure 3.5: Failure mechanism in TCFF at NTV(0.6V)



(a) 1.2V **(b) 0.6V**

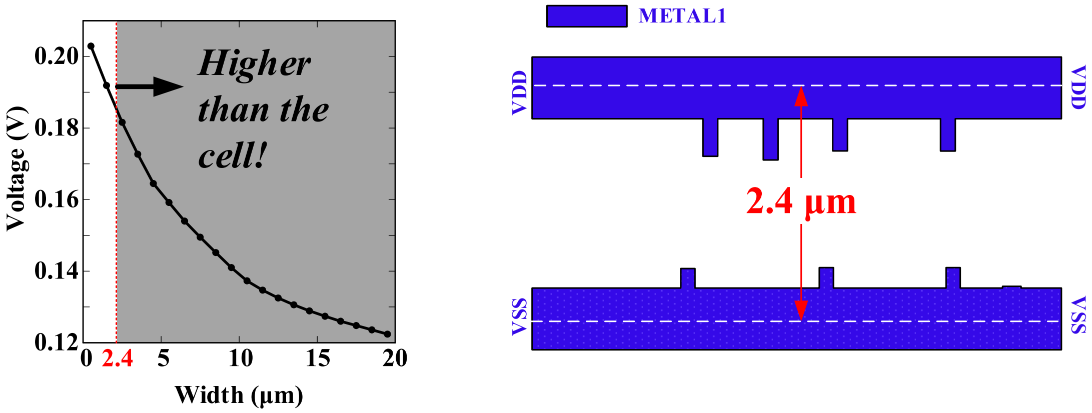
Figure 3.6: Simulation results show TCFF internal node voltages at (a)V_{dd} = 1.2V,(b) V_{dd} = 0.6V when D rising at $CK = 0$.

Consider, for example, the case when D is rising with CK=0 as shown in Figure 3.5. Since CK=0 the data transition is expected to be captured by the latch consisting of M03, M05, and M06. However, for M05 to drive node n2 which controls M03, vd2 should be equal to supply voltage (V_{dd}). M05 pull-up effort will be weakened if vd2< V_{dd}. Note that vd2 is pulled up to V_{dd} by M10 and M18. This stacked pull-up network is negated by pull-down of M06 and M07 through M04. Note that M07 is fully on because n3 does not drop until n2 has crossed $V_{dd}/2$. This causes node n2 to pause mid-rail waiting for n3 to transition. Once n3 transitions M17 turns on strengthening M05. This is supported by 65nm SPICE simulation waveforms shown in Figure 3.6 at 1.2V and also at 0.6V (NTV).

The primary drawback is that vd1 and vd2 are pulled up by stacked devices which suffer from body effect. The current i1 = i2 + i3 charges the capacitance on node n2. Note that i3 drains node n4 which experiences a non-negligible glitch. The resulting shoot-through currents in both the output buffer (driven by n4) and the master inverter (driven by n2) worsen TCFF dynamic power. This problem is exacerbated at NTV where the node n2 pauses mid-rail for as long as 25 ns.

Figure 3.7 present the hypothetical solution for TCFF failure. As it is shown in Figure 3.7a, Sizing up M10, M17 and M18 might seem to resolve this issue. However, as shown in Fig 3.7b, the design calls for unreasonably large devices (need to size the transistors higher than the rail-to-rail pinch of the std-cell) to even halve the voltage drop (VDD-vd1), which is not practical.

The resulting cell layout after factoring in these issues, for both TCFF and S2CFF, is shown in Figure 3.8 along with the corresponding sizes. Note the poly routing for CK



(a) M10,17 and 18 width vs (VDD-vd1)

(b) Rail-to-Rail Pinch of the std-cell, Cell height is fixed as 2.4 µm.

Figure 3.7: A hypothetical solution for TCFF failure.

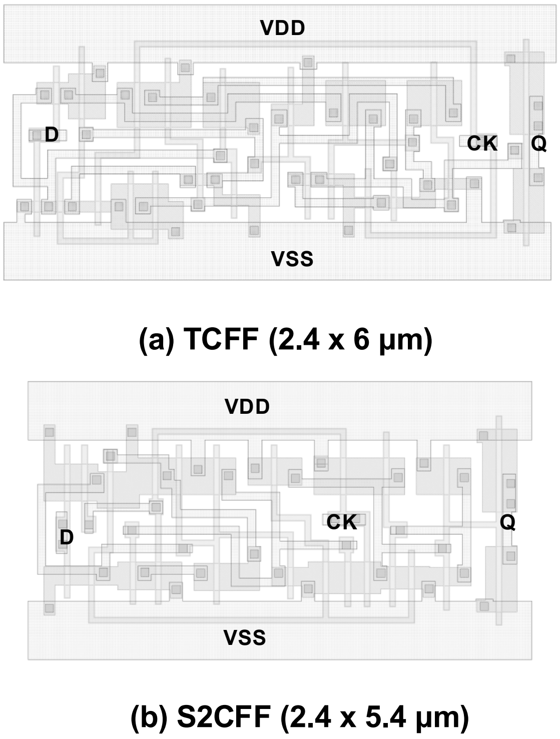
which may not be possible for smaller technology nodes [145] calling for routing in higher metal layers and in turn imposing constraints on automated place-and-route (P&R).

3.2.3 Transistor Stacks in datapath

Monte-carlo analysis was used to estimate the variability in CK-Q delay for both these designs at 1.2V and at 0.6V. As shown in Figure 3.9 S2CFF shows 34% less variation compared to TCFF over 10k runs. The variability worsens at 0.6V giving a variation co-efficient (σ/μ) of 4 for the TCFF at 0.6V, which is 32× higher than the S2CFF.

In order to investigate further, a commercial variation analysis tool was used to narrow down the cause of OCV, and the results are shown in Figure 3.10. About 80% of the sensitivity is due to two or three transistors with the remaining 18+ transistors contributing to less than 20% of the variation. The transistor labelled on the graph corresponds to the schematics in Figure 3.5 and 2.10. Note that the sensitive devices are almost always part of a multi-stack path. This exposes another issue with the design of SPC FFs - the higher the number of stacked devices in the data path, the more vulnerable the design becomes to OCV. However, note that although S2CFF has a stack consisting of M19-M22, these are weak keepers and are not part of the critical data path.

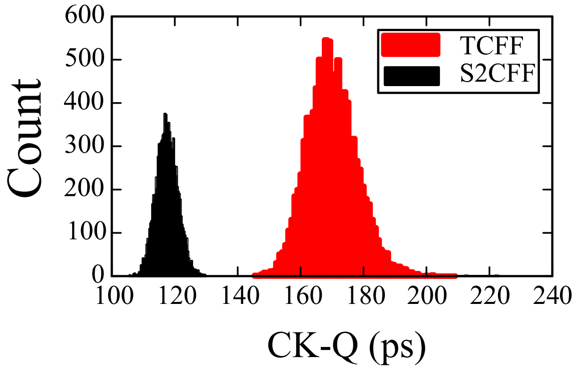
The cell-level design summary is presented in Figure 3.11. Note that SPC FFs, in comparison with TGFF, have >2× clock pin capacitance (Figure 3.11a). A significant portion of the internal clock node capacitance is now apparent at the clock pin which has adverse consequences with EDA flows as will be shown in the next section. Also, the large devices in TCFF worsen its leakage power despite having three fewer devices compared to S2CFF. Figure 3.11b shows the dynamic power of TCFF and S2CFF compared with



(a) TCFF (2.4 x 6 µm)

(b) S2CFF (2.4 x 5.4 µm)

Figure 3.8: Standard-cell layout of (a) TCFF (b) S2CFF



	σ/µ	
	1.2V	0.6V
S2CFF	0.029	0.13
TCFF	0.046	4.2

Figure 3.9: Analysis of Variability at different supply voltage.

TGFF for increasing activity rates. Considering the real-world constraints, the power benefit at 20% activity drops to 40%. TCFF achieves lower power consumption, but aggravated OCV effects from stacked paths cause the TCFF functional failures. The Monte-Carlo (10K samples) in Figure 3.12 shows that 5% runs failed at 0.54V/SS/0°C, i.e. cell level simulation shows the yield is 95%. For estimating the macro level yield, the Equation 3.1 can be used. Where the P_{fail} is the failure rate of the cell, N is the number of DUT cell in the known macro. Assume the TCFF is used for the implementation of an

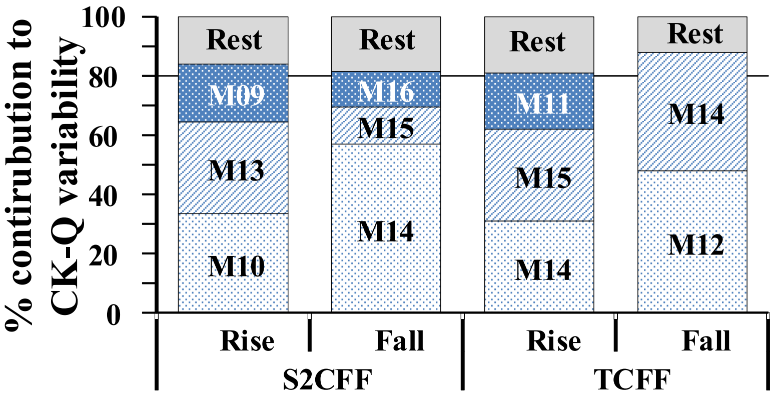


Figure 3.10: Analysis of Variability sensitivity of S2CFF and TCFF.

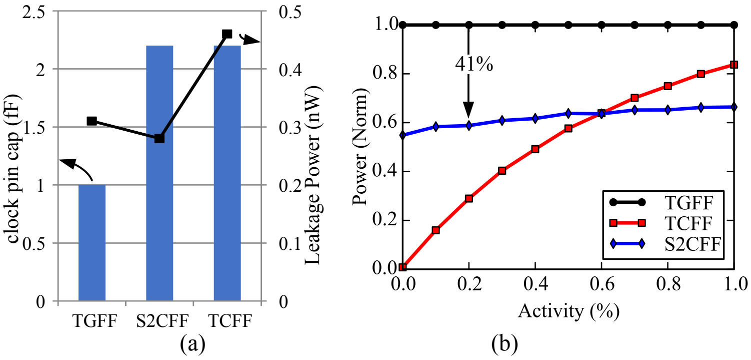


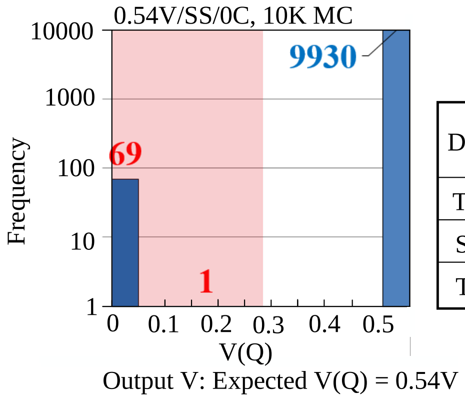
Figure 3.11: Std-cell design summary, (a) leakage and clock capacitance and (b) power vs activity.

AES-128 encryption engine, the number of the flops is 437. According to the estimate, it can be realized that a yield of the TCFF based AES-128 is only 4.6%.

$$Yield_{macro} = (1 - P_{fail})^N \tag{3.1}$$

3.3 Considerations For System Synthesis

Before carrying out the back-end implementation, all flops are tuned with the minimum transistor size for correct functionality. To evaluate the functionality of each FFs, the FF cells are layed out and extracted (equivalent Resistor and Capacitance considered) in the HSPICE model. 10K Monte-Carlo simulation were applied to each FF cell at each PVT corners (0.66V/FF/80°C, 0.6V/TT/25°C, 0.54/SS/0°C for NTV, 1.32V/FF/-40°C, 1.20V/TT/25°C, 1.08/SS/125°C). If the functionality test is pass, the extracted models (e.g. '.lib', '.DB', '.GDSII' etc) is combined with the commercial NTV standard cell library. Otherwise, the size tuning needs to be applied to the failed FF cell. The flow will be repeated until the FF cell passes all the functionality test at each PTV



Design	Fail (<0.27V)	Macro level Yield
TGFF	0	100%
S2CFF	0	100%
TCFF	70	4.6%

Figure 3.12: TCFF functionality yield estimation with 10K Monte-Carlo simulation at 0.54V.

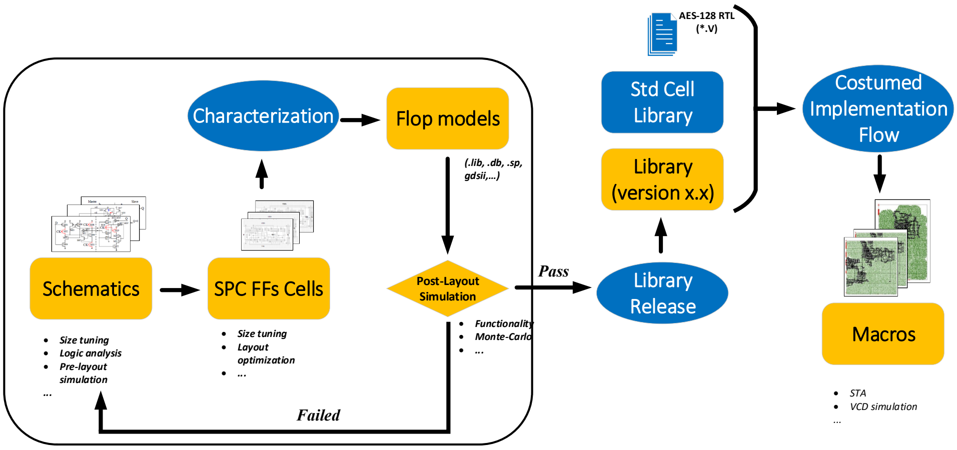
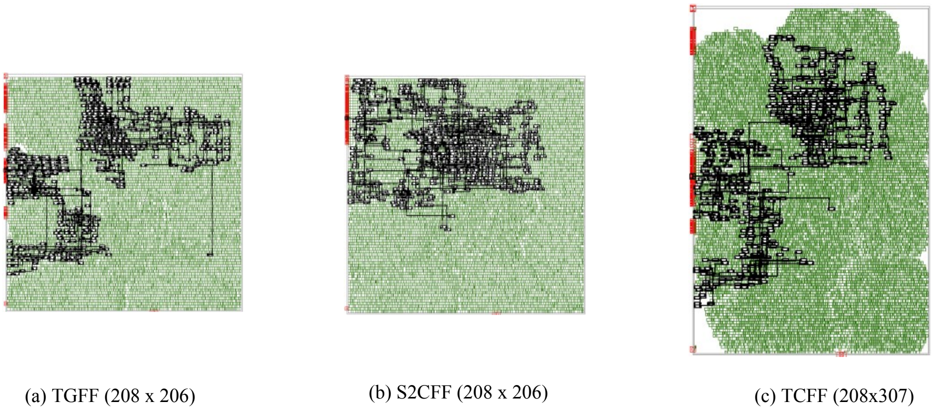


Figure 3.13: Macro implementation flow: From schematic configuration to macro implementation.

corner. Note that the TCFF is not able to pass the functionality test at PVT corners in the NTV region due to the design limitation. Therefore, the TCFF is sized with the minimum failure rate at worst corner for fair comparison. Since all the FFs are tuned with minimum functionality size, each FF is in power-optimal rather than speed-optimal form. The whole design flow is shown in Figure 3.13.

As a common block found in wireless sensor node designs [142], an approximately 5000 gate AES-128 encryption/decryption accelerator with 32b datapath was implemented as a benchmark using prototype near-threshold standard cell library and TGFF, TCFF or S2CFF.

Industry-standard synthesis and P&R tools were used with similar constraints targeting hold corners of 1.32V/FF/-40°C and 0.66V/FF/25°C, setup corners of 1.08V/SS/125°C and 0.54V/SS/25°C, and leakage corners of 1.2V/TT/25°C and 0.6V/TT/25°C. The floor-plan for the designs are shown in Figure 3.14, highlighting clock-tree and sequential elements. It was anticipated that additional clock capacitance in the TCFF and S2CFF


(a) TGFF (208 x 206)

(b) S2CFF (208 x 206)

(c) TCFF (208x307)

Figure 3.14: AES-128 implementation floorplan for (a) TGFF (b) S2CFF and (c) TCFF highlighting clock tree, clock buffers and FFs

Table 3.2: Comparison of AES-128 macro for each SPC FF

		TGFF	TCFF	S2CFF
CellArea(sq.μm)	Combinational	32579	39894	33097
	Sequential	6617	6909	6350
	Clock	183	76.8	189
Active (μW/MHz)	Combinational	1.87	2.27	1.93
	Sequential	0.58	0.28	0.36
	Clock	0.45	0.44	0.52
Leakage (nW)	0.6V,TT,25°C	277	411	279
Speed (MHz)	0.54V,SS,25°C	4	3.3	4

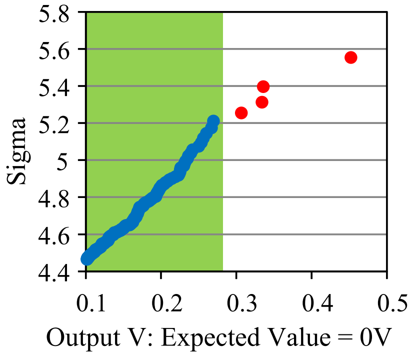
implementations would lead to deeper clock trees, but this was not observed for the 437 flip-flops in this design. It is likely that larger designs might suffer from greater clock tree buffering.

3.3.1 Power, Performance, Area

A summary of the three implementations with vector-free (10% toggle) typical power simulation is presented in Table 3.2. Although clock tree complexity is not apparent, higher clock switching power can be observed in the TCFF and S2CFF designs. However, this is made up for by the savings in both cases although not as much as 50% as claimed despite TCFF costing a 49% area increase. Also, even with relaxed timing constraints, the area of the TCFF macro is dramatically higher due to combinational logic bloating to compensate the large worst-case setup time of approximately 140ns - half of the entire 50-60 gate critical-path delay seen in the other designs - and this further erodes block level power savings.

3.3.2 Yield Analysis

Since FFs constitute the data storage elements of sequential digital circuit blocks, they are key to determining functional yield (the proportion of working chips). FFs can impact functional yield through bit flips caused by hold failure or latch breakthrough, i.e., undesired propagation from input to slave on the clock falling edge. Hold failure can be rectified during synthesis by hold buffer insertion, while latch breakthrough is internal to the FF and cannot be rectified by modifying timing outside the FF. Hold buffer insertion during synthesis may not account for hold time OCV, in which case extra hold time margin is required. FFs can also impact parametric yield (proportion of chips meeting sign-off timing) due to OCV in their setup time and propagation delay, which may not be accounted for during STA. This impact is usually small because the FF timing parameters are a small proportion of a longer timing path comprising many combinatorial stages. Therefore, the key metrics for the yield impact of FFs are: hold time variation, latch breakthrough, setup time variation and propagation delay variation. The margin required for OCV depends on the required chip yield and the number of FFs per chip. For example, the AES core implemented in this work contains 437 FFs, so each FF needs to meet an OCV margin of 4.08σ for 99% AES core yield.



FF margin for latch breakthrough

Design	ME (SS/0.54V/0°C)
TGFF	5.24σ
S2CFF	5.62σ
TCFF	5.24σ*

ME= Margin Estimate

Figure 3.15: Yield analysis for NTV operation

Assuming that hold buffer insertion and hold time margin addition mitigate the Hold Time Variation. Therefore, the latch breakthrough and excessive hold time variation are the only intrinsic failure mechanisms. The functional yield for the TGFF, S2CFF, and TCFF is simulated using a commercial high-sigma simulation tool, considering failures from either latch breakthrough or excessive hold time variation. Results shown in Figure 3.15. While TCFF appears to achieve a high sigma based on a strict failure criterion of V(Q) > $V_{dd}/2$, closer inspection (Figure 3.15) shows that many tail values are close to mid-rail, implying a much lower effective yield. S2CFF achieves the highest yield, implying significantly better robustness to data flips from latch breakthrough or excessive hold time variation.

3.4 Design Requirement of the NTV SPC FF

NTV has demonstrated a significant reduction in energy consumption and is, therefore, a potential solution for implementing ultra-low-power IoT applications. This work surveyed SoA SPC FFs to evaluate suitability for NTV operation and highlighted critical issues with incumbent designs. Contention paths and dynamic nodes in some SPC FFs impede low-power implementation and even the designs that are static and contention-free have complex design and implementation issues which necessitate sub-optimal device sizing.

The analysis was further extended to a larger design to investigate compromises to EDA flows when using SPC FFs. It is fair to conclude that while it may be possible to optimize the FF on its own to have low clock capacitance (and even eliminate clock buffers) other practical aspects cannot be overlooked. These issues lead to degraded benefits which must be understood. Another critical aspect is the logical derivation of the SPC FF scheme which must be straightforward for a better understanding of the design as it would allow designers to trade-off higher degrees of local optimization to achieve a better overall result.

The TCFF scheme has an advantage over S2CFF in this respect. However, S2CFF seems to perform better at NTV than TCFF or TGFF. Circuit and implementation complexities reduce the claimed 50% power saving for a 10% increase in area to ±3% and sometimes at the cost of 50% area increase and reduced yield. Based on this analysis, the following desirable features of SPC FFs are expected to lead to more robust and deployable designs: 1) Zero contention paths, 2) Zero dynamic nodes, 3) Minimize stacked structures on data path, 4) Minimize multiple master-slave interface paths and 5) Minimize clock pin capacitance as well as internal clock nodes.

Driven by the above discussion, a modified FF circuit is proposed for enabling the original TCFF operating at NTV in the next Section.

3.5 A modified TCFF circuit for NTV operation

The previous sections evaluate SPC FFs in the context of NTV applications and highlight the complexity associated with these designs. The analysis shows the evidence that the recently developed TCFF [110] is not suitable for NTV designs owing to the design limitations. For enabling the TCFF to operate at NTV level, a TCFF optimization strategy is proposed, the new flip-flop topology is called TCFF-NTV. The proposed topology is compared with the published SoA designs at different design phase, from the cell level to the chip implementation level. The results show that the proposed TCFF-NTV can work properly at NTV and the macro level yield is an improvement

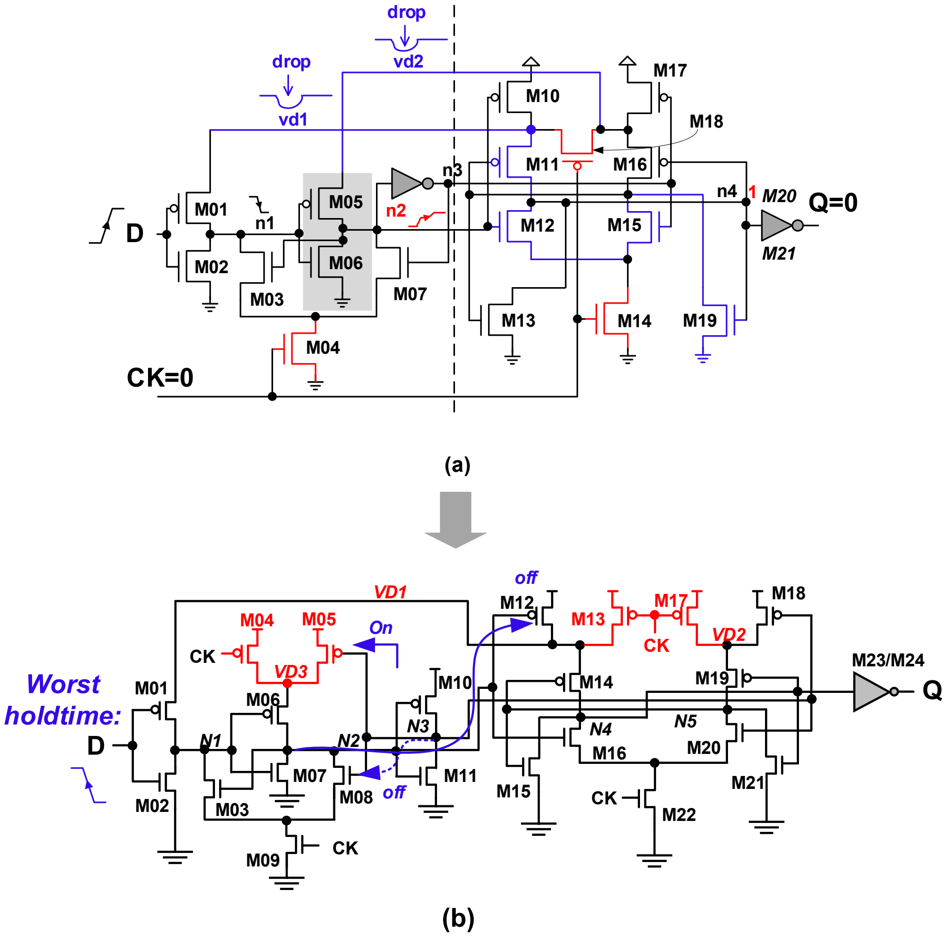


Figure 3.16: Proposed TCFF optimization strategy (a) Original TCFF schematics,(b) The Proposed TCFF-NTV schematic, the worst case hold time paths are highlighted.

over the original TCFF based design by 20×. Benefit from the simple hold time path of the proposed design, the Monte-Carlo analysis shows that the TCFF-NTV achieves better characteristics of the variability in hold time (54% less in $3\sigma_{Hold}$) compare to the conventional TGFF.

3.5.1 Proposed TCFF-NTV design for NTV operation

As it is discussed in the previous section, it can be seen that the failure of the TCFF at NTV is caused by the high stack topology in the slave-latch and the racing condition at internal node n2 due to the aggressive topology compression. To fix these issues in TCFF without aggressively up-sizing the critical transistors which are discussed in Figure 3.7, we proposed an improved version of the TCFF topology for NTV operation, named TCFF-NTV (see Figure 3.16).

In the proposed TCFF-NTV, the clocked PMOS M18 in the original TCFF (Figure 3.16(a)) is removed. Nodes VD1 and VD2 are modified as independent internal nodes

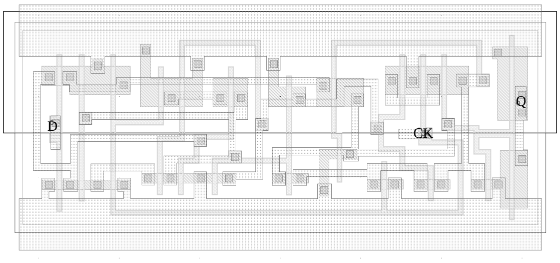
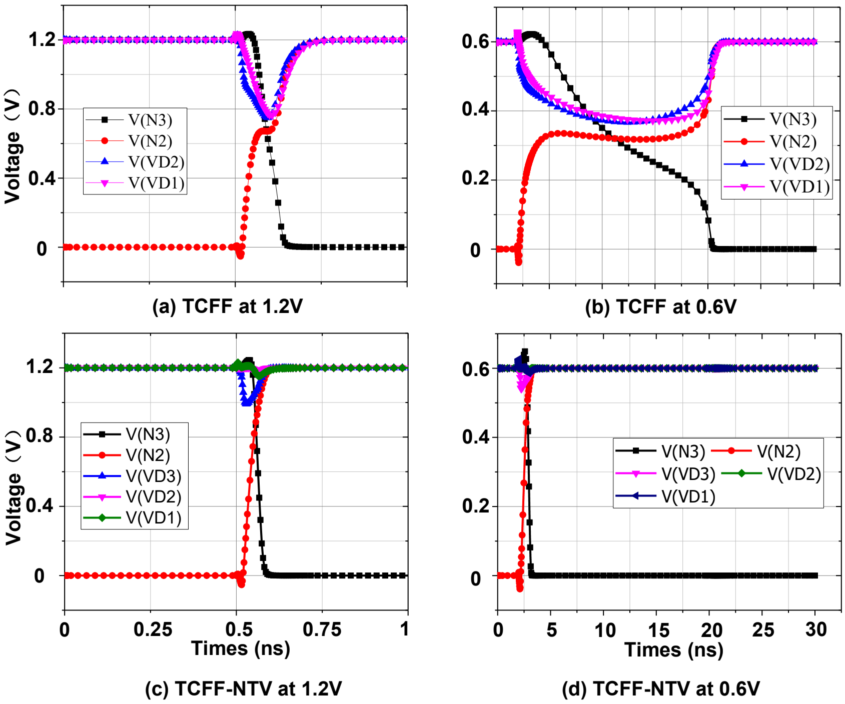


Figure 3.17: Standard-cell layout of TCFF-NTV(2.4 × 6.4).



(a) TCFF at 1.2V

(b) TCFF at 0.6V

(c) TCFF-NTV at 1.2V

(d) TCFF-NTV at 0.6V

Figure 3.18: Operation waveform of TCFF and the proposed TCFF-NTV at (a)(c) 1.2V and (b)(d) 0.6V.

by adding a clocked PMOS (see Figure 3.16(b)). The benefit from the modification are that, the PMOS stacks in the slave-latch is optimized to 2 stacks (maximum stacks) from the original three stacks PMOS in TCFF ([M17, M18, M11] and [M10, M18, M16]). Consequently, the effect of the high stack process variation can be reduced. Also, rather than sharing the PMOS, the node VD2 in Figure 3.16(a) is split by adding the PMOS transistors back to node VD3(see Figure 3.16(b)). By isolating the internal node VD2 and VD3, a full swing can be achieved at node N2 rather than being affected by internal nodes in the slave-latch. The standard-cell layout of the TCFF-NTV is shown in Figure 3.17.

Figure 3.18 shows the post-layout simulation results of internal nodes of TCFF and the proposed TCFF-NTV at the different V_{dd}. As it was discussed in the previous section,

Table 3.3: A comparison of each SPC FF

0.6V,TT,25	**TGFF**	**S2CFF**	**TCFF**	**TCFF-NTV**
Total transistors	24	24	21	24
Area	12	12.96	14.4	15.36
Setup (ns)	1.26	2.76	14.29	2.23
Hold (ns)	-0.19	1.7	-1.26	-1.34
CK-Q (ns)	2.7	2.65	2.4	3.2
D-Q (ns)	3.96	5.54	16.69	5.43
CK cap (fF)	0.83	2.44	2.4	2.7
Leakage (pW)	66	60	61	51

the voltage drop at VD2 in TCFF directly affects the speed of N2 rising and further reduce the speed of N3 falling (Figure 3.18a). And the issue is worsen at NTV (Figure 3.18b). It can be seen that the N2 is floating for 10 ns when V_{dd} = 0.6V.

With the proposed modification scheme, it can be seen that the voltage drop issue on VD2 is eliminated at both nominal voltage (Figure 3.18c, V_{dd} = 1.2V) and NTV (Figure 3.18d, V_{dd} = 0.6V). As a consequence, no floating issue is observed at N2 in TCFF-NTV .

3.5.2 Comparative analysis of the proposed TCFF-NTV

Essential leakage power, timing, and area information in the 'liberty' model are summarized in Table 3.3. The proposed TCFF-NTV has the same total transistor count as TGFF and S2CFF. The transistor usage is three more than in the TCFF. Due to the complex topology, the area of the cell is 28% bigger than the TGFF, 18% bigger than S2CFF and 7% bigger than the TCFF. As far as the timing characteristics are concerned, the setup time of TCFF-NTV is similar to the SoA design S2CFF. Also, compared to the original version of the TCFF, the TCFF-NTV reduce the Setup by 5 ×. What is more, TCFF-NTV achieves the minimum hold time due to the short CK path. More details about the hold time advantage of the proposed design are introduced in the next section. The CK-Q of the TCFF-NTV is higher than SoA FFs due to the longer Data-to-Q gate levels. According to the D-Q delay results, the TCFF-NTV improves the performance of the TCFF by 67%. Similar to the SoA SPC FFs, the CK transistors are directly connected to the clock tree, the TCFF-NTV has higher clock caps than conventional TGFF. A comparison of leakage power consumption characteristic shows the proposed TCFF-NTV achieves the minimum leakage power which is 51pW.

At the cell level, the proposed design has better power consumption compared to SoAs and the conventional TGFF. Figure 3.19 shows the normalized power consumption

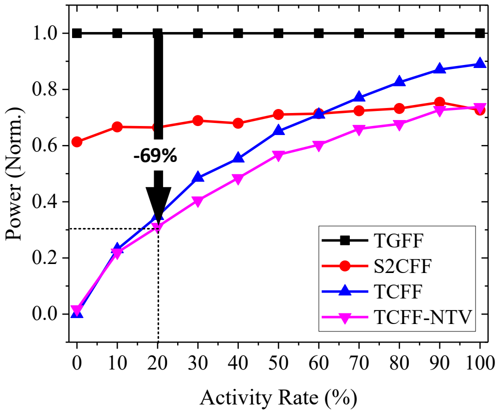
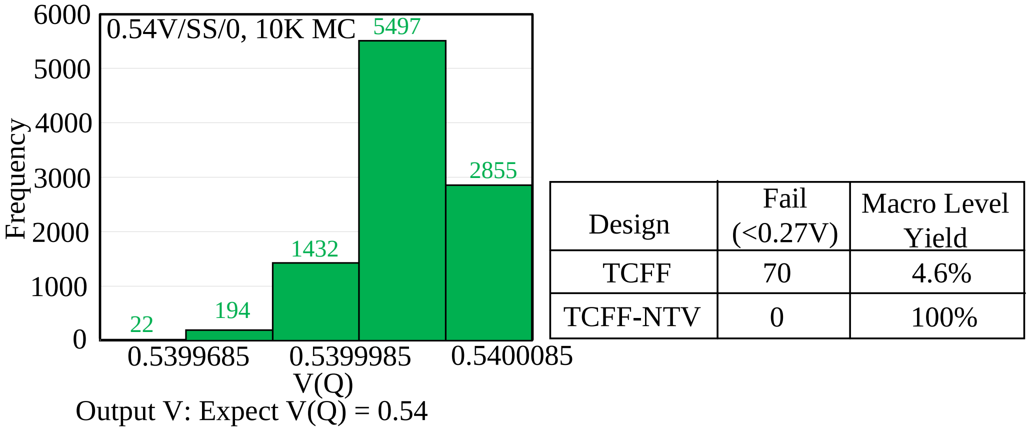


Figure 3.19: Normalized power consumption against different activity.

against different activity rates. It can be seen that the TCFF-NTV achieves minimum power consumption among the SoAs along the full range of activity rates. At the average data rate (20%), the proposed design reduced the power consumption of the TGFF by 69%. Compare to the SoA design S2CFF. The TCFF-NTV reduced the power by 30%.

3.5.3 Variability and yield analysis

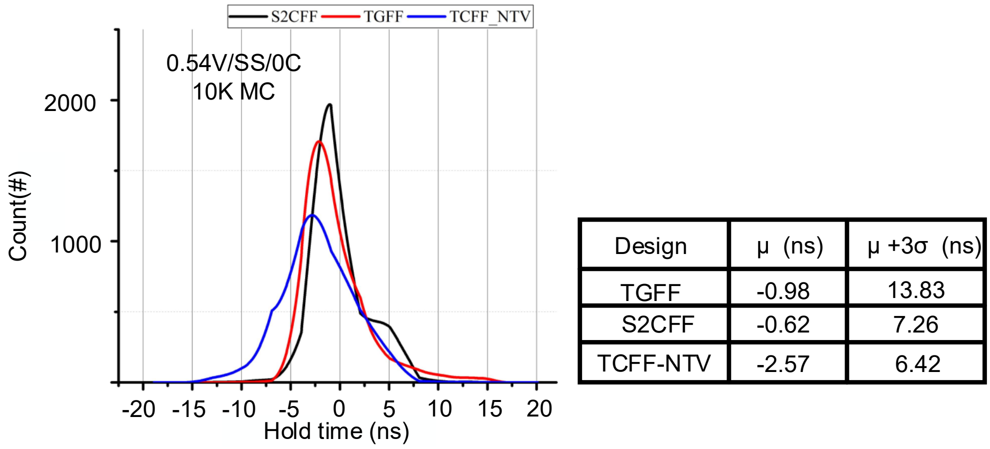
Monte-Carlo analysis was used to estimate the variability in Hold time and functionality for the proposed TCFF-NTV topology at the worst case of NTV(0.54V/SS/0°C). Since the critical topology in TCFF is modified in TCFF-NTV, the proposed TCFF-NTV is capable of operating at NTV.



Design	Fail (<0.27V)	Macro Level Yield
TCFF	70	4.6%
TCFF-NTV	0	100%

Figure 3.20: TCFF-NTV functionality yield estimation at 0.54V, SS, 0°C.

The functionality yield estimation results for the proposed TCFF-NTV are shown in Figure 3.20. It can be seen that the proposed topology can work properly at NTV with 0 failures and the macro level yield at NTV is improved by 95.4% compared to the original TCFF (see Figure 3.12), which means the previously introduced design limitation in TCFF is addressed.

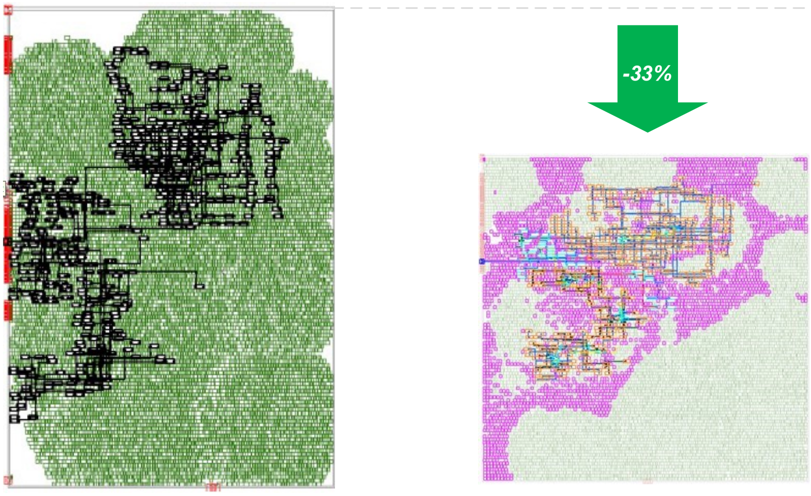


Design	μ (ns)	μ +3σ (ns)
TGFF	-0.98	13.83
S2CFF	-0.62	7.26
TCFF-NTV	-2.57	6.42

Figure 3.21: Flip-Flop Hold time variation distribution at 0.54V, SS, 0°C.

The stability of hold time is one of the most critical parameters of the flip-flop cell. Typically, the variation in hold time can be fixed by setting extra hold margin and adding buffers at the chip implementation level. However, by using such a strategy, the area penalty will be significantly increased [146]. Also, if the variation in hold time is too high to be fixed with such a strategy, the hold time variation can erode the whole system and lead to a low yield in chip implementation.

The worst case hold time in the proposed TCFF-NTV is the when D switches from 0 to 1 after the clock edge (see Figure 3.16(b)). As long as M06 discharges the N2 and the N3 is pulled up to 1, the N2 will be isolated since the M04 is shut-off by N3. So that, the worst case hold time of TCFF-NTV is dependant on the N3 discharging speed. The benefits from the original TCFF topology, the N3 discharging path is a simple one stack NMOS (M10), which is simpler than the hold time path in the S2CFF (M9/M10, two stacks) shown in Figure 3.3. Therefore, the proposed TCFF-NTV appears to have a better hold time characteristics than the SoA S2CFF. For further evaluate the variability in hold time at NTV at the worst corner(0.54V/SS/0°C), the Monte-Carlo analysis is applied to the proposed design to compare with the SoA designs and conventional TGFF in terms of mean value of Holdtime (μ_{Hold}) and 3 sigma ($\mu_{Hold}+3\sigma_{Hold}$) in hold time distribution. The Flip Flop Hold time variation distribution is shown in Figure 3.21. The result shows that the proposed TCFF-NTV has the lowest μ_{Hold} and $\mu_{Hold} + 3\sigma_{Hold}$ compare to the TGFF and the S2CFF. The benefit from the simple hold time path, the $\mu_{Hold} + 3\sigma_{Hold}$



(a) TCFF (208 × 307)

(b) TCFF-NTV (208 × 206)

Figure 3.22: AES-128 implementation floorplan for (a) TCFF and (b) TCFF-NTV. TCFF-NTV achieves a 33% reduction in block area compared to TCFF based design.

Table 3.4: Comparison of AES-128 macro for FFs

		TGFF	TCFF*	S2CFF	TCFF-NTV
CellArea(μm^2)	Combinational	30079	43264	30350	30371
	Sequential	5568	6492	5938	6862
	Clock	107	187	245	237
	Total	35647	49757	36288	37233
Active (μW)	Combinational	13	18.6	12.5	12.4
	Register	1.66	0.70	0.77	0.58
	Clock	0.40	0.73	0.78	0.78
Energy (pJ/-cycle)	0.6V, TT, 25°C	14.86	19.58	14	13.5
Speed (MHz)	0.54V, SS, 25°C	4	3.3	4	4

of the proposed TCFF-NTV is a 54% lower value than the TGFF. Therefore, the smaller hold time variation in the proposed TCFF-NTV can potentially improve the overall yield of the NTV chips.

3.5.4 System Synthesis Level Analysis

As in the previous section, the same AES-128 encryption/decryption accelerator was implemented as a benchmark using prototype near-threshold standard cell library and the proposed TCFF-NTV. Industry-standard synthesis and P&R tools were used with similar constraints targeting hold corners of 1.32V/FF/-40°C and 0.66V/FF/25°C, setup corners of 1.08V/SS/125°C and 0.54V/SS/25°C, and leakage corners of 1.2V/TT/25°C and 0.6V/TT/25°C. The floor-plan for the proposed TCFF-NTV based AES-128 macro is shown in Figure 3.22b. Compared to the original version of the TCFF based macro, the macro size is reduced by 33%.

A summary of the implementations with Value Change Dump (67% toggle) typical power simulation is presented in Table 3.4. As it was anticipated before, the additional clock capacitance in SPC FFs implementations, i.e. TCFF, S2CFF, and TCFF-NTV, lead to deeper clock trees. It can be seen that the proposed TCFF-NTV based macro increased the clock tree area by 2.2 times from the conventional TGFF based design, higher clock switching power also can be observed in SPC FF designs. Since the TCFF-NTV has bigger area than the SoAs and TGFF, the TCFF-NTV based design has the bigger sequential cell area, which is 23% bigger than TGFF, 6% bigger than the TCFF and 16% bigger than the S2CFF. However, Benefit from the simple hold path of the proposed TCFF-NTV, discussed in section 3.5.2, the chip level total area overhead is 4% compare to the TGFF, and 3% compare to S2CFF based macro. The proposed TCFF-NTV also shows the prominent superiority in power reduction. As it is shown in Table 3.4, the TCFF-NTV based design reduces the active register power by 65% from the conventional TGFF based design with maintained chip performance.

3.6 Concluding Remarks

In this chapter, the SoA SPC FFs are surveyed to evaluate suitability for NTV operation, and the design issues with incumbent designs are highlighted. The analysis covers a wide range of dierent design phases when using the SPC FFs. Based on of the observation, we concluded the requirements for designing a more robust and deployable ULP sequential circuits as follow. 1. No contention path should be accepted in design. 2. Dynamic nodes need to be eliminated in NTV operating circuit. 3. The stacked structures on data path need to be minimised. 4. Master-Slave interface path needs to be simplified. 5. Since the CK pin is directly connected to the global clock tree, the clock pin capacitance need to be minimized.

A modied FF circuit is proposed to enable the original TCFF to operate at NTV, called the TCFF-NTV. With the proposed scheme, the design issue in TCFF is eliminated. At NTV, the TCFF-NTV improved in yield compared to the previous TCFF by 95%.

The proposed TCFF-NTV also shows superiority in the hold time characteristics. At the worst case PVT corner at NTV, the $3\sigma_{Hold}$ of TCFF-NTV is 54% less than the TGFF. The advantage of TCFF-NTV in power efficient is prominent. For the average data write activity of system (20%), the power reduction of TCFF-NTV is 69% when compared with the TGFF. Compare to the SoA S2CFF, the power reduction is 30%. At system implementation level, the TCFF-NTV based AES-128 is 33% smaller than the original TCFF based design. With the VCD based simulation, it can be seen that the TCFF-NTV reduced the register power by 65% when compared to the base-line TGFF based macro.

However, the proposed design still not meet all the requirement of the ULP sequential logic circuit. To achieve the above introduced characteristic, the TCFF-NTV sacrificed the area of the cell. Due to the complexity of the proposed design, the cell area is bigger than TGFF (28%+) and SoAs. What is more, multiple master-slave paths still exist in TCFF-NTV. It brings a significant challenge in layout design. The clock pin capacitance of TCFF-NTV also needs to be aware. Five transistors are directly connected to the clock tree which makes the clock pin cap of the TCFF-NTV is 2.2× of the TGFF. The system synthesis results are affected by these factors. The overall area of the cell areas is 4% higher than the TGFF based design. The larger clock network can be observed (2× larger) and the clock network power is accordingly higher (1.95×) compare to the TGFF.

Therefore, more in-depth research is still needed in order to meet all the ULP sequential circuit design requirements. In the next chapter, more contributions to ULP sequential circuit design are introduced.

Chapter 4

Ultra-Low Power Single Phase Clocked Flip-Flops Design

As it was introduced in Chapter 3, robustness is a primary factor in the design of ultra-low power FFs for low voltage operation. Conventionally, dynamic logic is introduced to achieve better speed [144]. However, dynamic logic is vulnerable to process variation, making dynamic circuits less robust at NTV [65], so *Fully-Static* FF operation is therefore desirable for ultra-low power designs. Additionally, for ultra-low power FFs, *single-phase clocked* (SPC) operation maximizes power efficiency in the NTV region since the inverter chain (which provides the complemented clock signal) can be eliminated [147]. Also, contention paths need to be eliminated in ultra-low power FF designs, since the contention between the pull-down network and the keepers results in increased power consumption [148]. Also, any ratioed logic is vulnerable to process variation which may be exacerbated at NTV levels [65]. Thus, ultra-low power FFs should be *contention-free*, avoiding data contention paths. Besides, reduced area helps reduce silicon real-estate costs. By analyzing the properties of the widely-used TGFF [78][149] and other SoA ultra-low power FF designs, it was found that SoA ultra-low power FFs do not meet all the above requirements, and their claimed benefits can reduce significantly as yield, EDA and system level issues are addressed.

The TCFF (Figure 3.5) [110] is the representative SoA SPC FF. It uses 21 transistors (fewer than the conventional TGFF). Its fully compressed topology improves power efficiency for all α compared to TGFF. However, a design limitation can be observed in Chapter 3. For correct operation, in the case when D is rising at $CK = 0$, data 0 is expected to be latched at node $n1$ if $n2$ is pulled up to $vd2$ (turns on $M03$). For this, $vd2$ should be at supply voltage (V_{dd}), otherwise, $n2$ can be weakened which leads to high setup time and latch failure. However, in practice, a voltage drop is observed at $vd1$ and $vd2$ in this condition. Owing to the latency of M19 turning off, a temporary short-circuit path exists, weakening $vd1$ from V_{dd} via the path $M11 \rightarrow M12 \rightarrow M15 \rightarrow M19$

$\rightarrow GND$. Since $M18$ is on, $vd2$ is lower than V_{dd}. The $M05$ pull-up effort is weakened since $vd2 < V_{dd}$. Note that $M15$ will not be off, since $n3$ will not be pulled down to zero until $n2$ crosses the mid-rail of V_{dd}. $n2$ in this scenario can be slowly rising or floating at mid-rail, due to the degraded $vd2$. This analysis is supported by the SPICE simulation results (Figure 3.6) at both supply voltages (V_{dd} = 1.2V and 0.6V). Also, the voltage drop issue cannot efficiently be resolved just by resizing, as the Monte-Carlo simulation still shows a high setup time and very low yield (approx. 5%) owing to this limitation.

In the Chapter 3, a Near-Threshold SPC FF was proposed, named TCFF-NTV (Figure 3.16). The circuit was developed based on the TCFF. The advantage of TCFF-NTV in power efficiency is prominent. At $\alpha = 20\%$, the TCFF-NTV consumes 69% less power than the conventional TGFF. However, to achieve the low power characteristic, the TCFF-NTV sacrificed the area of the cell by at least 28% when compare to TGFF. Therefore, the proposed design still not meet all the design requirement of the ULP sequential logic circuit. What is more, the multiple Master-Slave paths still exist in TCFF-NTV. Moreover, these limitations have negative impacts on the system level performance when the TCFF-NTV is applied to macro level design. The result shows the total cell area of TCFF-NTV-based AES macro is 4% higher than the TGFF-based design. Also, owing to the high input CK capacitance of the TCFF-NTV, the clock net power of the TCFF-NTV-based macro is 1.95× higher than the TGFF-based design.

This chapter proposes 18TSPC, a SPC FF with only 18 transistors (the lowest reported for a fully-static contention-free SPC FF) with a novel master-slave topology (Section 4.1). With a simplified topology, it delivers a 20% reduction in cell area compared to TGFF. Unlike SoA designs, 18TSPC meets all ultra-low power FF design requirements. It has been implemented in 65nm CMOS along with a TGFF in 320-bit shift-register and AES-128 encryption engine design. This proves EDA compatibility and demonstrates circuit and system-level benefits. The design was first simulated (Section 4.2) then experimentally validated (Section 4.3) at 0.6V, 25°C, at various Data Activity Rate (α), showing that the proposed 18TSPC achieves reductions of 68% and 73% in overall ($P_{\alpha=10\%}$) and clock dynamic power ($P_{\alpha=0\%}$), respectively, and 27% lower leakage compared to TGFF. Furthermore, unlike TCFF, the measurements indicate superior 18TSPC performance at NTV. Also, 3 extended designs (19TSPC, 20TSPC, and 21TSPC) based on 18TSPC are proposed in Section 4.5 for ameliorating the hold time and CK pin capacitance property of the baseline design, 18TSPC. Rich variants of the proposed ULP SPC FFs provide back-end engineers a comprehensive range choice for targeting different ULP objectives with the considered trade-off.

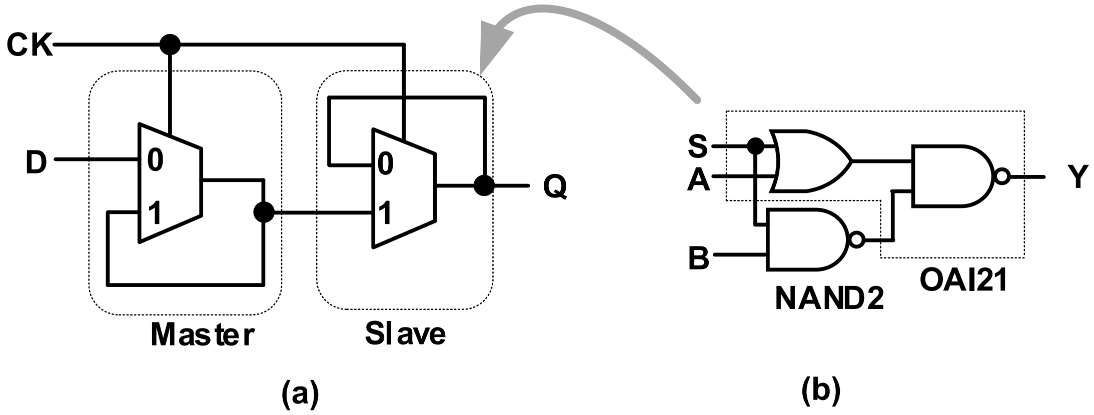


Figure 4.1: (a) The 2-to-1 Multiplexer (MUX2) based MS FF[149]. (b) Gate level of the MUX2 without inverters.

4.1 Proposed Single-Phase Clocked FF

4.1.1 SPC FF design approach

The aim of the design is to carry forward the enhancements achieved by previously-reported FFs in terms of cell area, power consumption and performance, but to overcome the limitations of these designs. To do this, the initial step is to evaluate the Boolean function of a positive-edge trigged Master-Slave FF (MSFF):

$$D_{ML}^{present} = \overline{CK} \cdot D + CK \cdot D_{ML}^{previous} \tag{4.1}$$

$$D_{SL}^{present} = \overline{CK} \cdot D_{SL}^{previous} + CK \cdot D_{ML}^{present} \tag{4.2}$$

In Equation 4.1, D is the data input, $D_{ML}^{present}$ is the present data in the master latch, and $D_{ML}^{previous}$ is the data which has been latched from D during the previous low CK. In Equation 4.2, $D_{SL}^{present}$ is the present data in the slave latch, and $D_{SL}^{previous}$ is the data which has been latched from the output of the master latch during the previous high CK in the slave latch.

Based on these equations, MSFF can be abstracted using two multiplexers [149], shown in Figure 4.1(a). However, the original MUX2-based FF requires inverters to apply a complemented clock signal. To mitigate the internal clock inverters for the select (CK) pin, a combination of a compound OR-AND-INVERTER (OAI21) gate and a NAND2 gate topology is adopted as the MUX2 circuit (Figure 4.1b). The boolean function of the the MUX2 circuit can be modelled in form of Equation 4.3.

$$Y = \overline{S} \cdot \overline{A} + S \cdot B \tag{4.3}$$

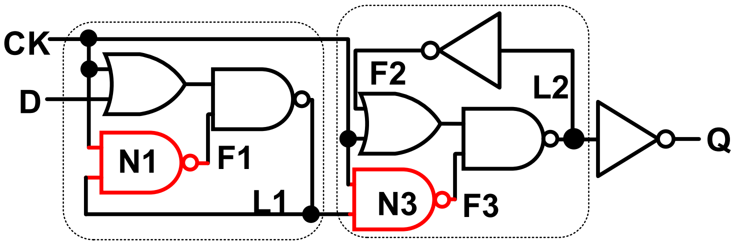
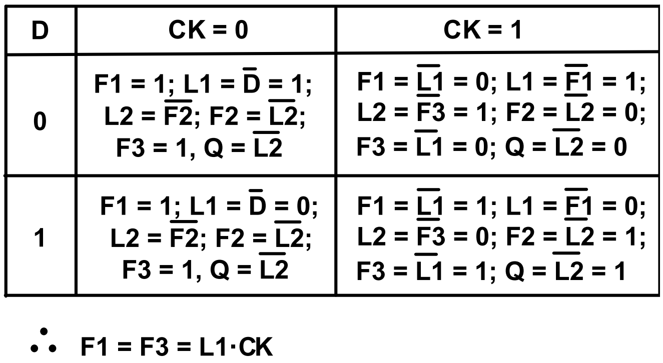


Figure 4.2: The gate level schematic of the proposed SPC FF.

By adopting the OAI21-based MUX2, the MSFF (Figure 4.2a) can be constructed in a reduced gate level topology (Figure 4.2). Assuming the combine gates are all implemented in static topology, it can be roughly estimated that the total transistor count (T) of the FF in Figure 4.2 is 32 (where OR2 is 6T, NAND2 is 4T, and Inverter is 2T).

4.1.2 SPC FF circuitry reduction



D	CK = 0	CK = 1

0	F1 = 1; L1 = $\overline{\text{D}}$ = 1;
	L2 = $\overline{\text{F2}}$; F2 = $\overline{\text{L2}}$;
	F3 = 1, Q = $\overline{\text{L2}}$
		F1 = $\overline{\text{L1}}$ = 0; L1 = $\overline{\text{F1}}$ = 1;
		L2 = $\overline{\text{F3}}$ = 1; F2 = $\overline{\text{L2}}$ = 0;
		F3 = $\overline{\text{L1}}$ = 0; Q = $\overline{\text{L2}}$ = 0

1	F1 = 1; L1 = $\overline{\text{D}}$ = 0;
	L2 = $\overline{\text{F2}}$; F2 = $\overline{\text{L2}}$;
	F3 = 1, Q = $\overline{\text{L2}}$
		F1 = $\overline{\text{L1}}$ = 1; L1 = $\overline{\text{F1}}$ = 0;
		L2 = $\overline{\text{F3}}$ = 0; F2 = $\overline{\text{L2}}$ = 1;
		F3 = $\overline{\text{L1}}$ = 1; Q = $\overline{\text{L2}}$ = 1

∴ F1 = F3 = L1·CK

Figure 4.3: Boolean function list of gate level SPC FF at different scenario.

Figure 4.3 summarized the boolean function list of gate level SPC FF (shown in Figure 4.2) in different scenarios. It can be observed that $F1$ and $F3$ are logically equivalent in each scenario. This implies that NAND gate $N3$ in the slave latch (Figure 4.2) is redundant. Therefore, NAND gate $N3$ can be merged with gate $N1$. The redundant gate reduced schematic diagram is shown in Figure 4.4a. Therefore, the total transistor count of the Flip Flop in Figure 4.4a can be estimated as 28T.

In the schematic-level design, gates $R1$-$N2$ and $R2$-$N4$ are combined as compound gates OAI21_a and OAI21_b which results in a saving of 8 transistors. The reduced NAND gate results in a saving of four transistors. The reduced topology results in a 20-transistor FF, with six transistors connected to CK (Figure 4.4b).

To further reduce the number of clocked transistors, a transistor merging process is applied to the 20-transistor SPC FF (Figure 4.5). When CK is low, the clock-connected

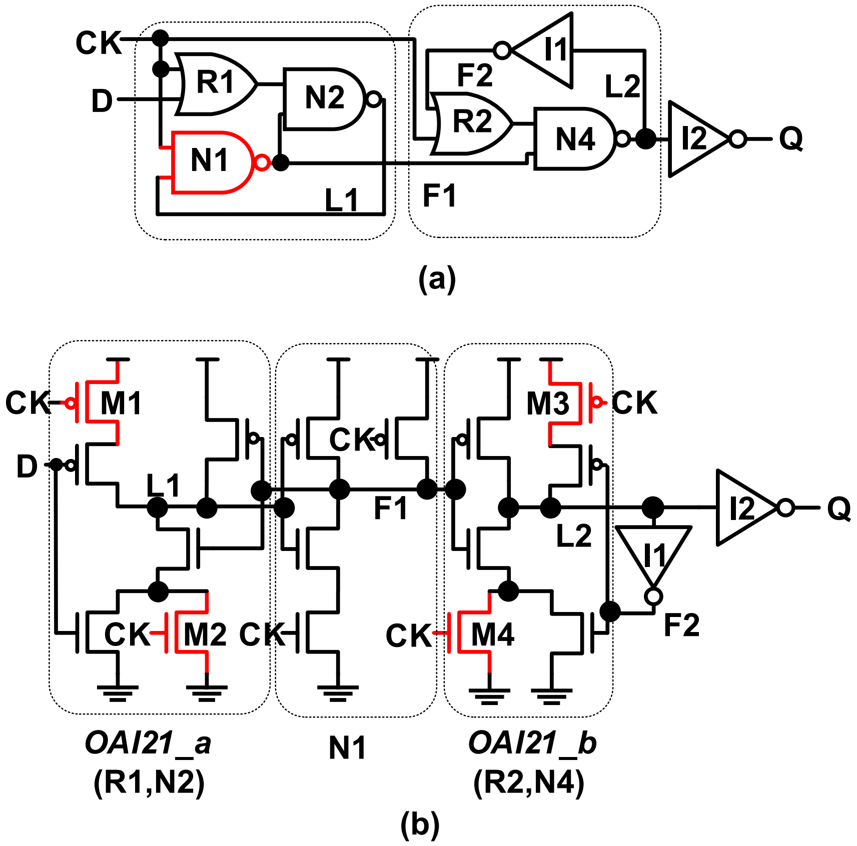


Figure 4.4: (a) The gate level schematic of the proposed SPC FF (redundant NAND gate eliminated). (b) Transistor level schematic diagram of the (a).

PMOS $M1$ and $M3$ are turned on, and nodes $X1$ and $X2$ are pulled up to V_{dd}. Otherwise, $X1$ and $X2$ are floating. Hence $M1$ and $M3$ can be merged.

Table 4.1: Net states at $Y1$ and $Y2$ at different D, $D_{SL}^{previous}$ and CK states

CK	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**
$D_{SL}^{previous}$	0	0	1	1	0	0	1	1
D	0	1	0	1	0	1	0	1
Y1	1^w	0	1^w	0	0	0	0	0
Y2	1^w	1^w	0	0	0	0	0	0

For CK $=$ 1 : Y1 $=$ Y2 $=$ 0
For CK $=$ 0: Y1 $= \overline{D}$ and Y2 $= \overline{D_{SL}^{previous}}$
$D_{SL}^{previous}$ $=$ previous data in slave-latch ($\overline{L2}$)
1^w : Weak 1 (V_{th} drop)

Further, when $CK = 1$, NMOS $M2$ and $M4$ are on, node $Y1$ and $Y2$ is pulled down to 0. When $CK = 0$, NMOS $M2$ and $M4$ are turned off, the voltage level at node $Y1$ and $Y2$ depends on the signal D and $F2$ respectively (see Table 4.1). This shows

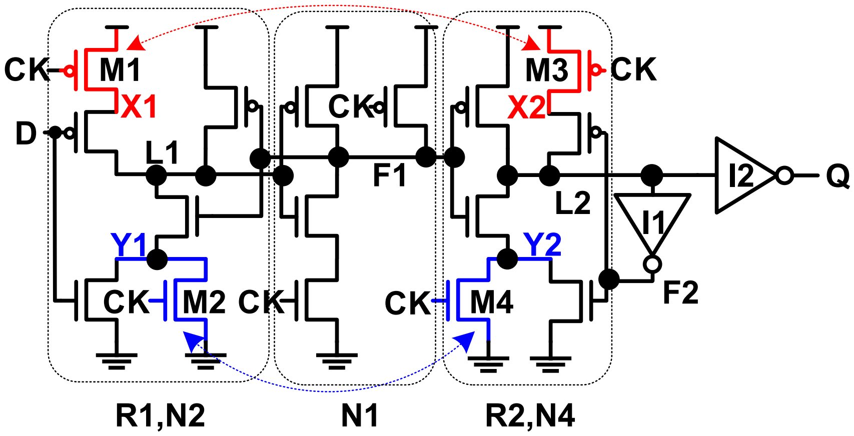


Figure 4.5: The SPC FF with 20 transistors in total. The transistor merging process can be applied to the highlighted transistors.

$M2$ and $M4$ can be replaced with a single clocked-NMOS (connected between $Y1$ and $Y2$), working as a pass transistor. When $CK = 0$, $Y1$ and $Y2$ are isolated since the clocked-NMOS is off. For $CK = 1$, the states of $Y1$ and $Y2$ are equal ($Y1 = Y2 = 0$).

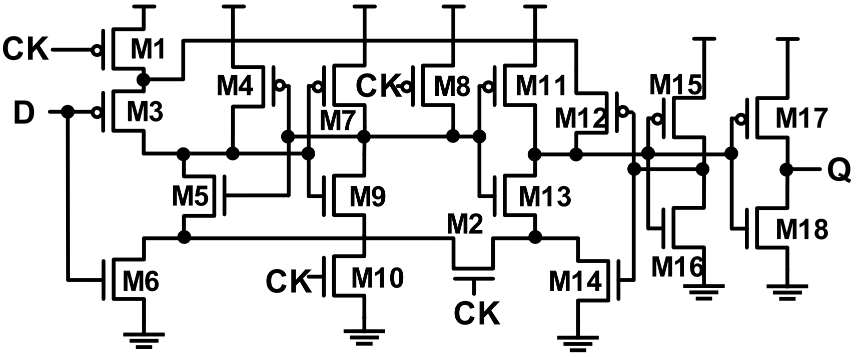


Figure 4.6: The schematic diagram of the proposed 18 transistors fully static Flip Flop (18TSPC).

Finally, by applying the above-introduced transistor merging process. The initial SPC FF with 32 transistors is optimized as the proposed 18-transistor SPC FF (18TSPC), shown in Figure 4.6.

4.1.3 18TSPC operation and timing path analysis

Figure 4.7 shows the operation of the 18TSPC at different CK and D states. When $CK = 0$, devices on D only change the state of $L1$ in the master latch. Since the slave latch remains isolated from D for $CK = 0$, the switching on $L1$ does not induce any data corruption in the slave latch. When $CK = 1$, D is isolated, and the FF outputs the

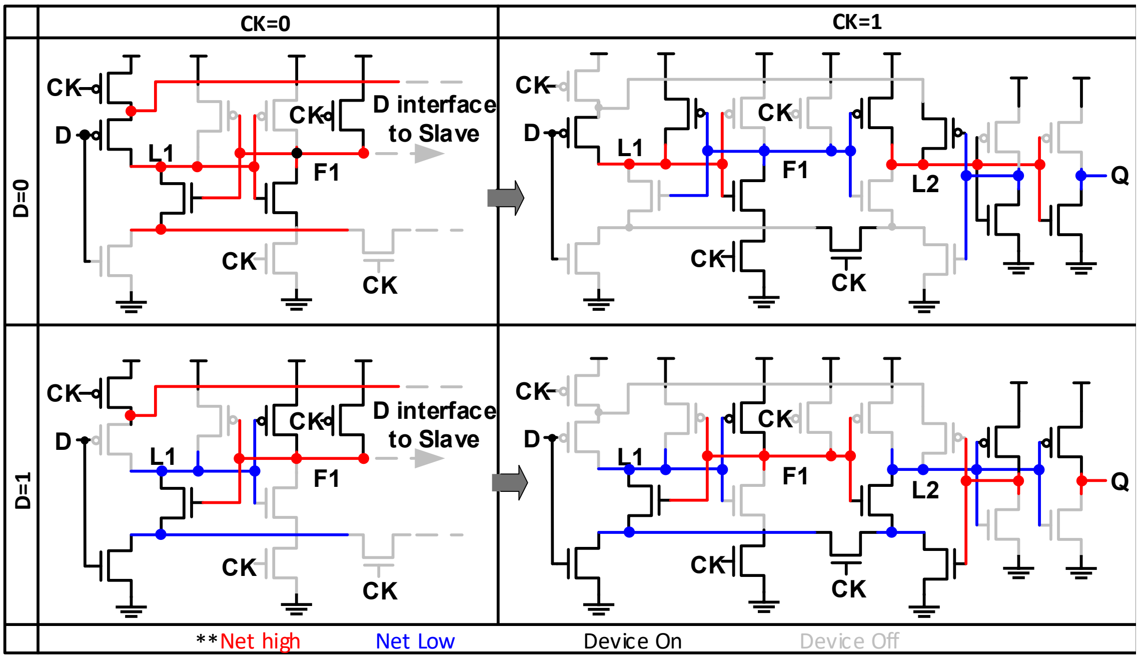


Figure 4.7: 18TSPC operation diagram at different CK and D states, highlighting the active devices, logic high nets and logic low nets.

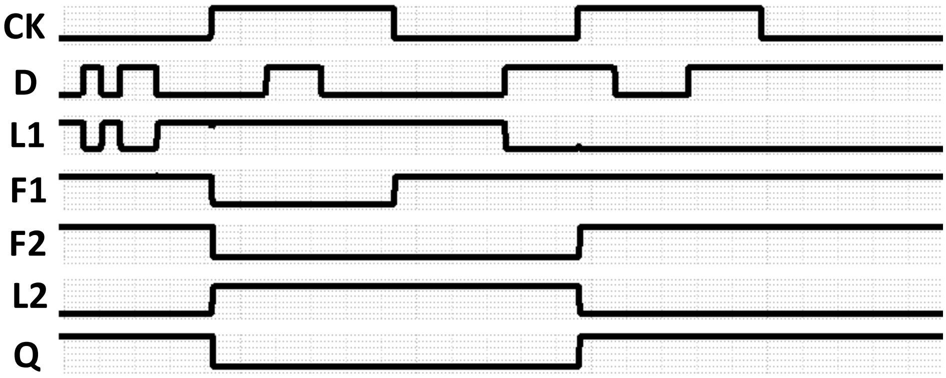


Figure 4.8: Operation Waveform of the internal nodes of 18TSPC.

previously latched data at $L1$ in the master latch. The SPICE simulation was applied to the schematic of the 18TSPC with 1.2V supply voltage, the test vector (D and CK pairs) covers all the operation of the Flop. The operation waveform of the 18TSPC is shown in Figure 4.8. From the waveform, it can be seen that there is no logic degradation, node floating or struggling rising/falling issues. No contention paths or dynamic nodes are observed in any of the scenarios in proposed 18TSPC.

The setup time of the 18TSPC is determined by the propagation delay from D to $F1$. The hold time is determined by the speed of $L2$ settling to its final value after the rising edge of CK. As shown in Figure 4.9, the worst-case hold time scenario is experienced when D falls too close to the rising edge of CK. If $M6$ is turned off by D before net $L2$ is fully discharged, a hold violation may be observed. The highlighted path in Figure 4.9a

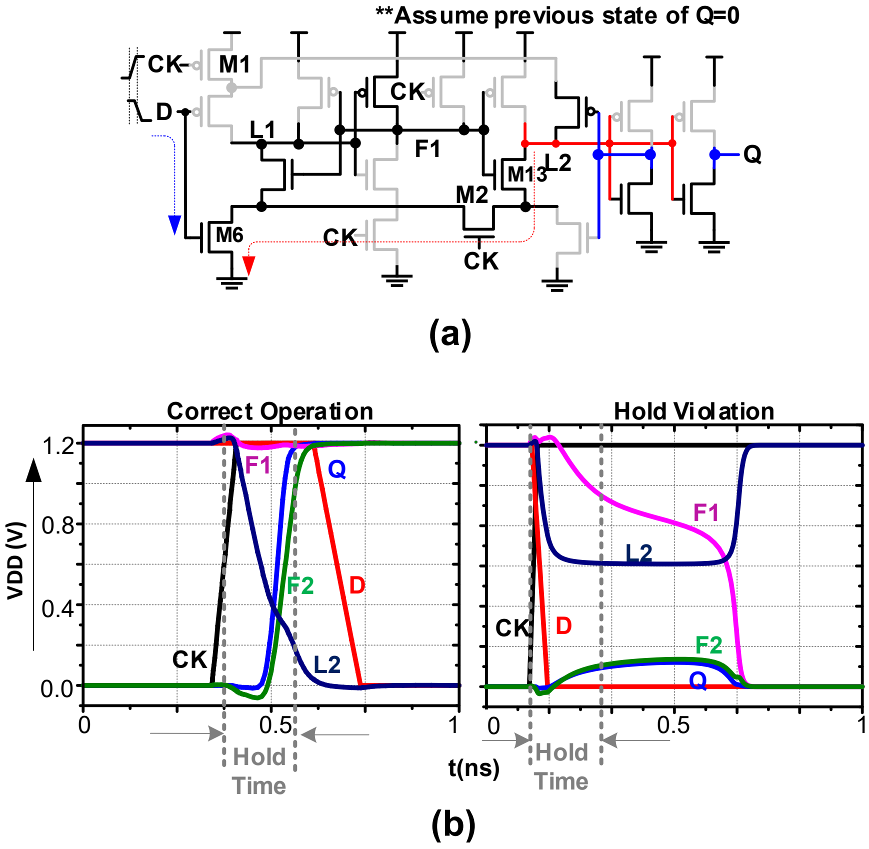


Figure 4.9: (a) Worst-case Hold time path analyse of the 18TSPC. (b) SPICE simulation results, waveform of 18TSPC with correct operation and hold violation at worst-case hold time condition.

is the critical hold time path of the design, and due to the proposed topology, the hold time is positive. The SPICE simulation waveform in Figure 4.9b illustrates both correct operation and the hold-violation scenarios.

Figure 4.10 is about the T_{CQ} analysis of 18TSPC. For Q falling, shown in Figure 4.10 (a), the speed of $Path_{CK_1}$ discharging the F1 and the speed of M11 pulls up the L2 dictate the CK to Q delay. For Q rising, shown in Figure 4.10 (b), the L2 is pulled down via $Path_{CK_2}$, the Inverter (M15,M16) can output a 1 slightly before the L2 is completely pulled down to 0 at the meaning time. The M14 is turned on by F2 accordingly. A fast path can be formed (L2 to GND) via the M13 and M14. Therefore, in this scenario, the T_{CQ} depends on the speed of M14 tuning on.

4.2 Simulation Results and Analysis

To evaluate the proposed design, 18TSPC, S2CFF, and TGFF have been laid-out and characterized by TSMC 65nm CMOS technology. For a fair comparison, the transistor

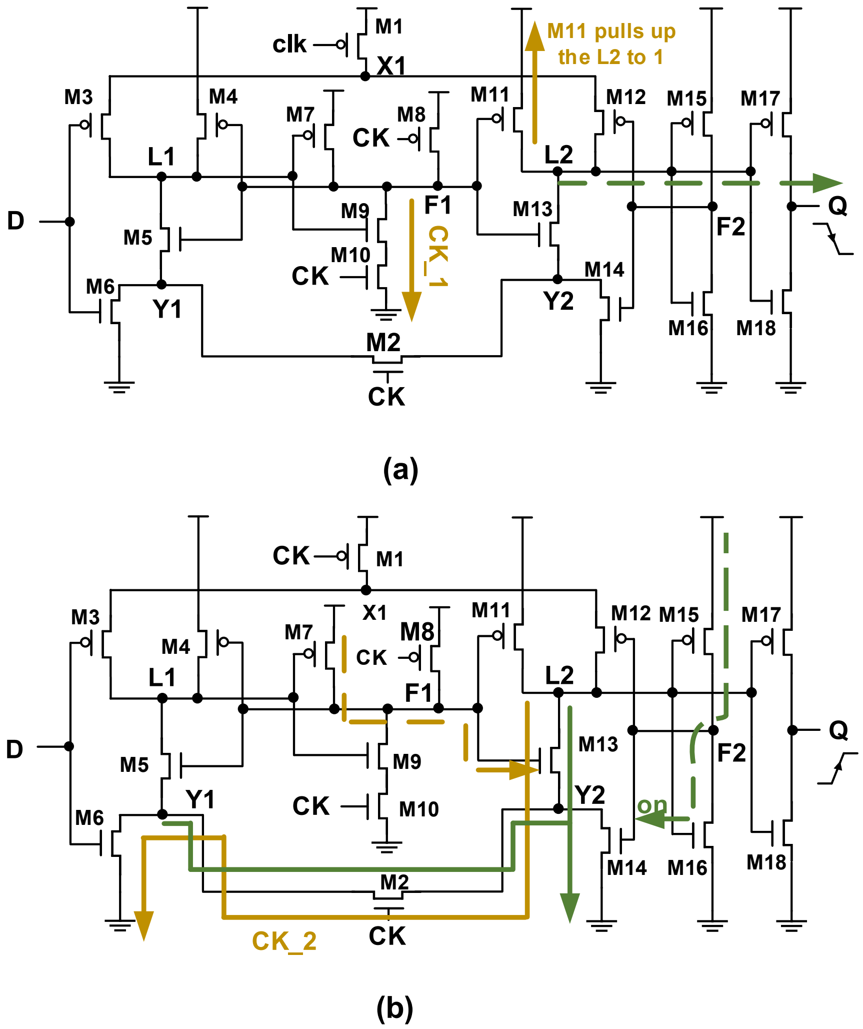
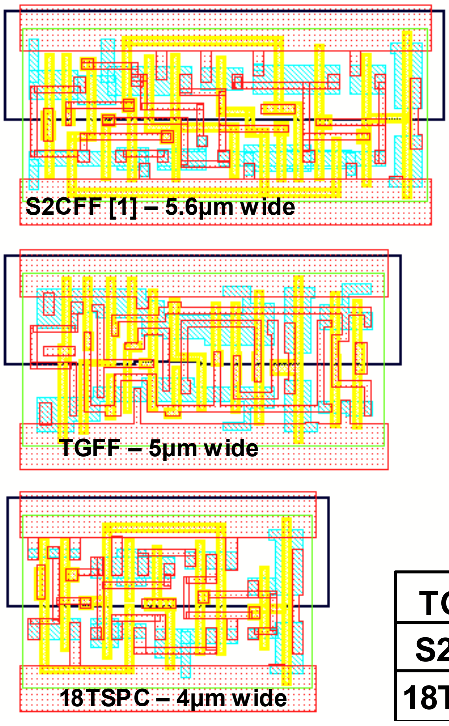


Figure 4.10: T_{CQ} path analysis of the 18TSPC. (a) Q rising at CK edge. (b) Q falling at CK edge.

sizes of each FF were tuned to achieve the minimum energy (E_0) point of the Energy-Efficient Curve (EEC), which is considered as the minimum size for correct functionality [150]. Post-layout Monte-Carlo simulations (10k runs) were performed for each FF, to evaluate functionality at different PVT corners. For EDA synthesis and further place-and-route (P&R) considerations, only the M1 metal layer is adopted in the proposed FF layouts. Figure 4.11 shows the layouts of S2CFF, TGFF, and the proposed 18TSPC, which shows a 20% and 29% reduction in cell area over TGFF and S2CFF respectively. Owing to its reduced circuitry and lower transistor count, 18TSPC achieves the lowest leakage power (104 pW at TT/1.2V/25°C) of the three FF cells, 27% less than TGFF and 32% less than S2CFF. In 18TSPC, the clocked transistor count is four, one transistor less than S2CFF. Hence, the clock pin capacitance of 18TSPC (2.16 fF) is 37% less than S2CFF. Since only two transistors are directly connected to the CK pin, TGFF also achieves the lowest clock pin capacitance (1.09 fF) of the three FFs. However, more



	Clock Pin Cap (fF)	P$_{Leak}$ 1.2V, TT, 25.C (pW)
TGFF	1.09	143
S2CFF	3.45	153
18TSPC	2.16	104

Figure 4.11: Layout of the S2CFF, TGFF and the proposed 18TSPC.

transistors (12 in total) are clock signal related in TGFF, which leads to higher dynamic power.

For reducing the area and CK network power for the conventional MSFF, one single clock inverter chain can be shared with multiple FFs, i.e. Multi-Bit FF (MBFF) topology [151]. Benefiting from the reduced topology, a multi-bit 18TSPC still has lower area compared to TGFF-based multi-bit FFs. The schematic diagram of the MBFFs based on 18TSPC and TGFF are shown in Figure 4.12. The 18TSPC based design shows 11% area reduction versus a 2-bit TGFF based MBFF. Compared with a 4-bit design, the area saving is 5%. A multi-bit 18TSPC also shows superior power efficiency compared to TGFF-based MBFF cells. At $\alpha = 0\%$, a 2-bit 18TSPC-based MBFF achieves a 66% power saving and a 4-bit 18TSPC-based design achieves a 60% power saving compared to the TGFF-based MBFFs. At $\alpha = 100\%$, the same designs show a 56% and 54% power saving, respectively. This benefits from the reduced number of CK related transistor count. For 2-bit MBFF, 20 transistors are CK (or buffered CK) signal connected in TGFF-based design. The 18TSPC has only 8 CK related transistors in the same design. For 4-bit MBFF, the TGFF-based design has 36 CK toggled transistors. In 18TSPC-based 4-bit MBFF, the CK toggled transistor is 16.

Table 4.2 shows the dynamic power and energy (per cycle) of each FF at different D and CK switching scenarios at TT/0.6V/25°C and TT/1.2V/25°C. The power data is the mean value collected from the power lookup table in generated Liberty files (.lib). In contrast with TGFF, the CK pin power is evenly distributed in each scenario.

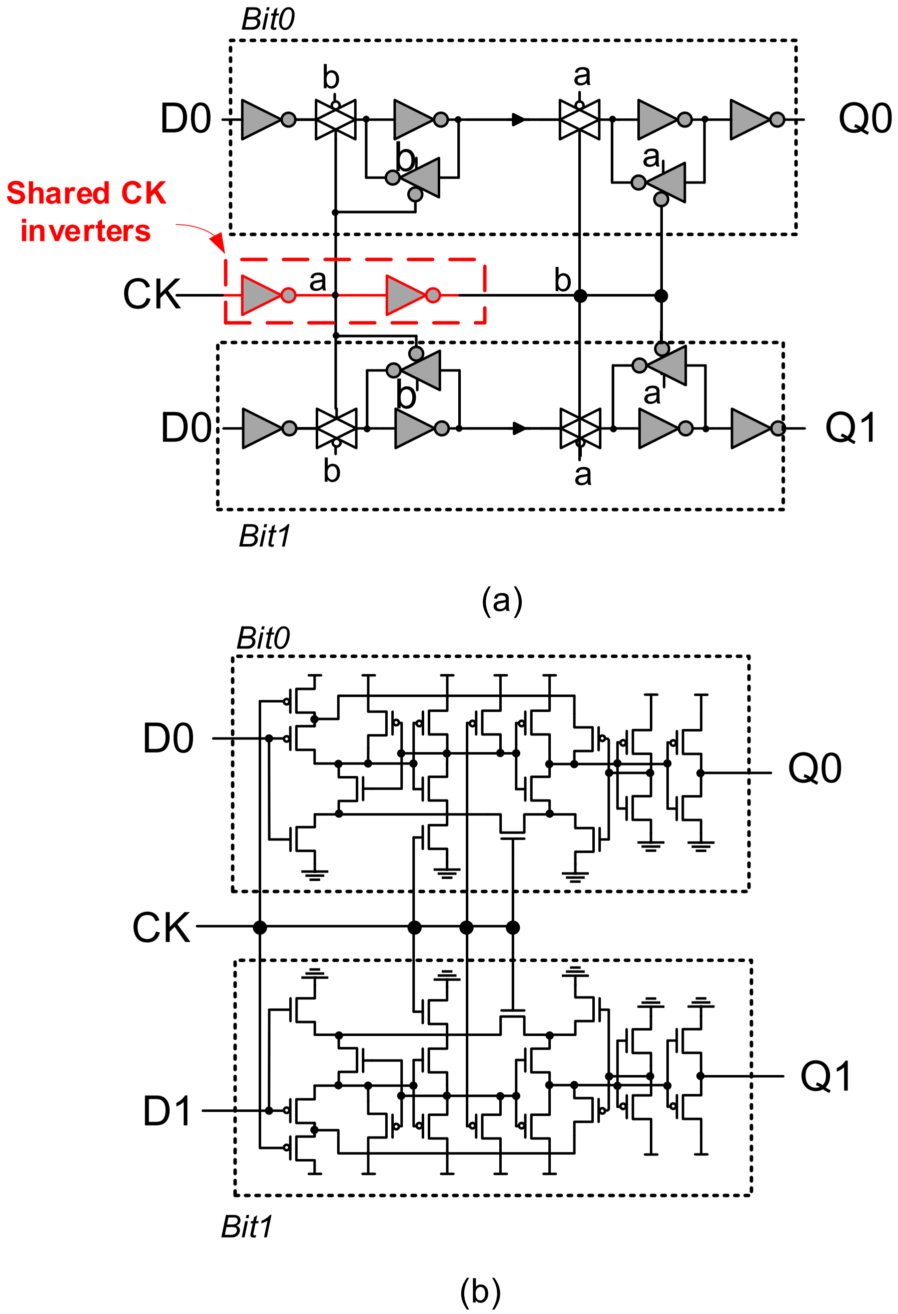


Figure 4.12: Schematic diagram of (a) TGFF-based and (b)18TSPC-based 2-bit MBFF

The dynamic power in SPC FFs is activity-dependent. In S2CFF, more CK power is consumed when $D = 0$ & CK rising and $D = 1$ & CK falling. In 18TSPC, higher CK power is reported for $D = 0$ & CK falling and $D = 1$ & CK falling. The unevenly distributed CK power in different transition scenarios is quite dependent on the topology of SPC FF. Overall, the proposed 18TSPC achieves lower dynamic power at different D and CK switching scenarios, and achieves lowest energy (2.99 fJ/cycle at TT/0.6V/25°C and 11.8 fJ/cycle at TT/1.2V/25°C) among the three FFs. The normalized results shown in Figure 4.13 highlight a 55% energy reduction versus TGFF at TT/0.6V/25°C

Table 4.2: Dynamic Power Comparison, Energy/cycle and Energy-Delay Product (ED) Comparison

PVT Corner	**FFs**	**CK pin power (nW)**				**D pin power (nW)**			
		D!		**D**		**CK!**		**CK**	
		rise	**fall**	**rise**	**fall**	**rise**	**fall**	**rise**	**fall**
TT/0.6V/25C	**TGFF**	1.33	1.68	1.48	1.52	0.93	1.64	0.06	0.62
	S2CFF	0.10	1.63	~0.0	0.50	1.24	1.49	~0.0	0.16
	18TSPC	0.28	0.80	~0.0	0.32	0.37	0.73	~0.0	0.17
TT/1.2V/25C	**TGFF**	5.51	7.04	6.03	6.56	3.82	7.22	~0.0	2.78
	S2CFF	7.37	0.22	~0.0	2.36	5.36	6.46	~0.0	0.82
	18TSPC	0.93	3.53	~0.0	1.50	1.53	3.39	~0.0	0.61

PVT Corner	**FFs**	**Q pin power (nW)**		**Energy at** $\alpha = 100\%$ **(fJ/cycle)**	**Norm. ED (i.e. Min E)**
		rise	**fall**		
TT/0.6V/25C	**TGFF**	1.86	1.93	6.64	46.81
	S2CFF	1.79	1.88	4.01	40.84
	18TSPC	1.73	1.42	2.99	21.99
TT/1.2V/25C	**TGFF**	7.79	8.74	27.10	43.97
	S2CFF	7.75	8.15	17.91	40.71
	18TSPC	7.42	6.13	11.8	23.99

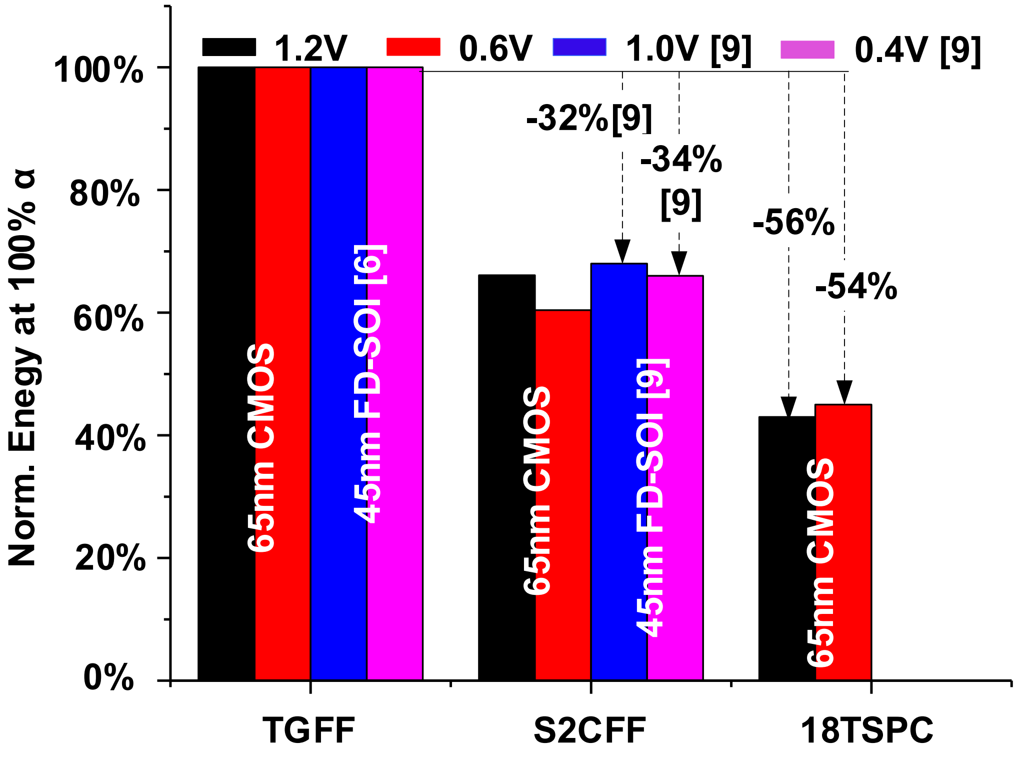
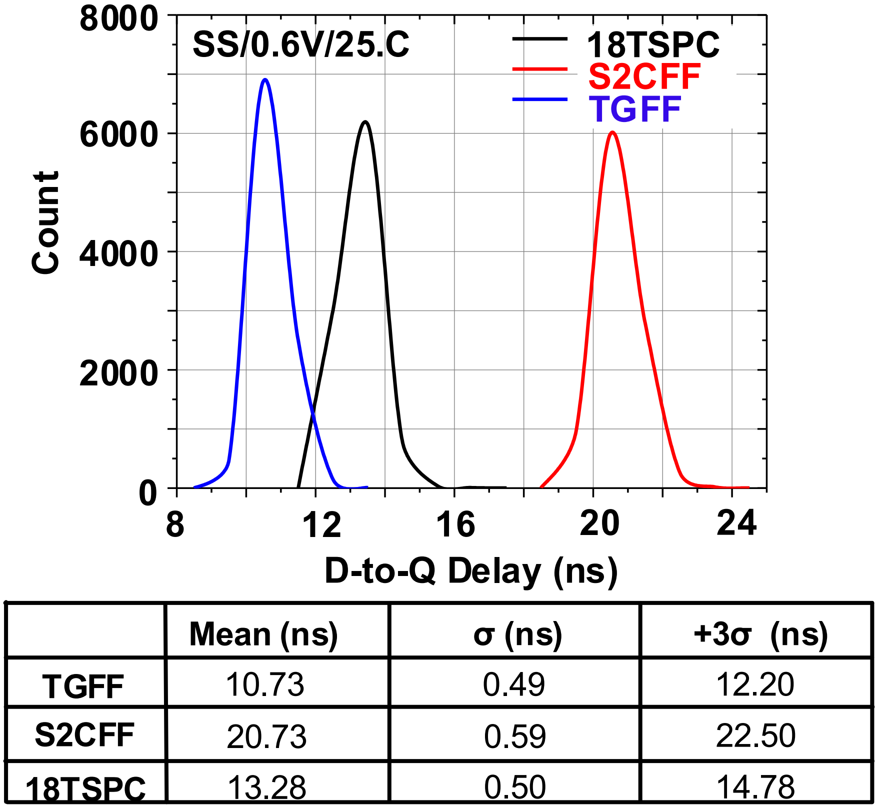


Figure 4.13: Normalized Energy/cycle with $\alpha = 100\%$ at nominal supply voltage (1.2V for 65nm CMOS, 1.0V for 45nm FDSOI [65]) and NTV (0.6V for 65nm CMOS, 0.4V for 45nm FDSOI [65].

and 56% energy saving against TGFF at TT/1.2V/25°C is achieved. Since the FFs are implemented to achieve E_0, the Energy-Delay (ED) product can be considered as the $MinE$ point on EEC [150]. The $MinE$ of the proposed 18TSPC is about 1.8× and



	Mean (ns)	σ (ns)	+3σ (ns)
TGFF	10.73	0.49	12.20
S2CFF	20.73	0.59	22.50
18TSPC	13.28	0.50	14.78

Figure 4.14: 10K Monte-Carlo simulation results of D-to-Q Delay.

1.7× better than TGFF and S2CFF, respectively, in the ED space at 1.2V. At 0.6V, the 18TSPC is about 2.1× more efficient than TGFF and 1.9× better than S2CFF in ED space. The 18TSPC shows energy efficiency in ED space at both nominal voltage and NTV operation.

Figure 4.14 shows the D-to-Q delay simulation results for the three FFs at SS/0.6V/25°. No functional failure was observed over 10K simulations. The proposed 18TSPC has a lower mean (μ) value in D-to-Q delay distribution than S2CFF (35% lower). The result shows the proposed design has a higher μ than TGFF (19% higher), considered as the performance penalty. The $\mu + 3\sigma$ value of the 18TSPC D-to-Q delay over 10K simulations is 14.78 ns, 34% lower than S2CFF, and 17% higher than TGFF.

For evaluating EDA compatibility and system-level characteristics, all three FFs were used to implement AES-128 macros using industry-standard EDA tools. Figure 4.15 shows the floor plan for each design. The clock trees are highlighted, illustrating the similar complexity of each design. In the AES-128 macro, FFs contribute 4% of standard cells, and all variants were synthesized for same area and timing constraints, as highlighted in Table 4.3.

Owing to the positive hold time characteristic of the 18TSPC, more hold buffers are inserted into the 18TSPC-based AES implementation. Because of this, the 18TSPC

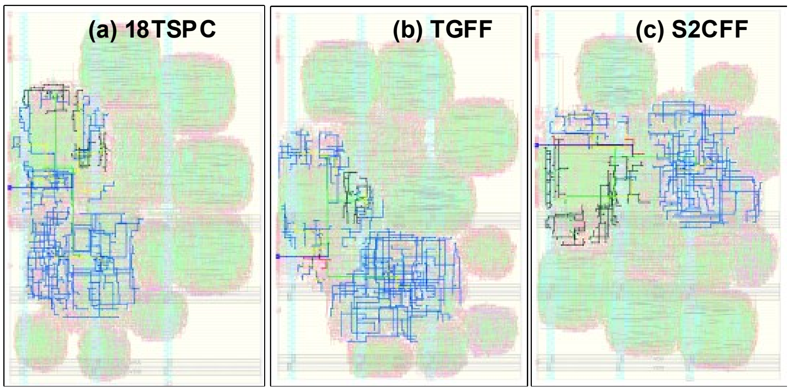


Figure 4.15: AES 128 floorplan of (a) 18TSPC, (b) TGFF and (c) S2CFF, the clock tree is highlighted.

Table 4.3: AES-128 Synthesis Results Comparison

	Unit	**18TSPC**	**TGFF**	**S2CFF**
Area	μm^2	32657	33313	33888
CK Buffers	#	17	11	15
Hold Buffers	#	72	0	0
NO. FFs	#	385	385	385
$\mathbf{P}_{REG}$	μW	2.76	5.65	5.15
$\mathbf{P}_{CK\_net}$	μW	1.62	1.38	1.68
$\mathbf{P}_{comb.}$	μW	57.4	56.3	56.97
$\mathbf{P}_{total}$	μW	61.8	63.3	63.8
WNS_SETUP	*ps*	0	0	0
WNS_HOLD	*ps*	9	0	0

Die Area : 200 μm × 299.6 μm, Target CK Frequency: 20 MHz,
Clock Uncertainty: 30 ps, Clock-Gating applied
Process Corners: 1.2V/TT/25°C, 1.08V/SS/125°C, 1.32V/FF/-40°C
WNS_HOLD: Worst Negative Hold Slack
WNS_SETUP: Worst Negative Setup Slack

based AES-128 macro consumes higher combinational power ($P_{comb.}$), 2% and 0.8% higher than the TGFF and S2CFF-based designs, respectively. Due to the better power efficiency of the proposed design, the register and clock network power ($P_{REG}+P_{CK\_net}$) of 18TSPC-based design is 37% lower than the TGFF-based macro and 36% lower than the S2CFF-based macro with clock gating applied. However, owing to the limited contribution of FFs in AES-128, the overall dynamic power (P_{total}) is 2.3% lower than the TGFF-based implementation (vector based simulation). A small negative slack in hold time is observed in the synthesis result, verified through static timing analysis after full RC extraction.

In modern SoC design, FFs are implemented with scan paths for testability. A MUX2 gate is added to the proposed 18TSPC in cell level, named S_18TSPC, The schematic diagram is shown in Figure 4.16. The S_18TSPC and the standard scan FF (S_TGFF) were also being used to implement the AES-128 macro with the same set up what was

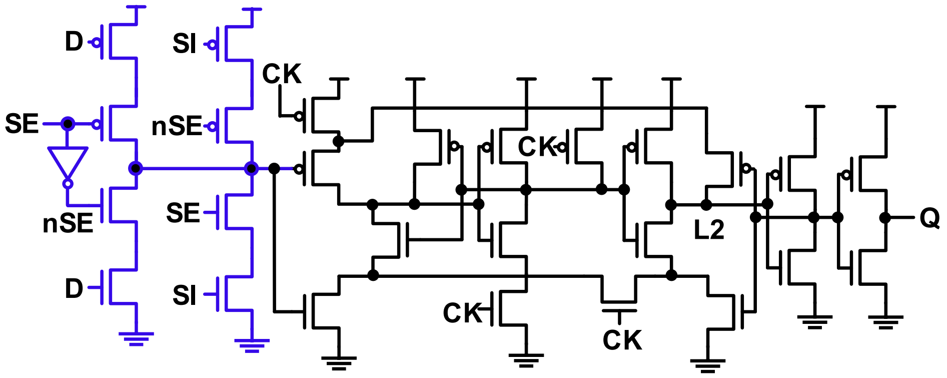


Figure 4.16: Schematic diagram of the Scan-18TSPC. Added transistors to the original 18TSPC is highlighted.

used in Table 4.3, as highlighted in Table 4.4. The S_18TPSC still has a 4% area saving at the cell level. Owing to the added MUX gate at the data in the path of the 18TSPC (one logic stage added), the data path delay is increased inducing a higher setup time compared to the S_TGFF. However, the S_18TSPC has a lower hold time than the original 18TSPC since the added MUX2 increases the data path delay.

Table 4.4: Scan FFs Implementation results

		Unit	S_18TSPC	S_TGFF
Cell Leve (1.2V/TT/25°C)	**+Transistors**	#	10	8
	Area	**norm.**	0.96	1
	Hold	**ps**	3	-6
	Setup	**ps**	170	91
AES-128	**Area**	μ**m**^{2}	36710	36514
	NO. FFs	#	385	385
	P$_{REG}$	μ**W**	4.48	5.82
	P$_{CK\_net}$	μ**W**	1.56	1.35
	P$_{comb.}$	μ**W**	5.63	5.75
	P$_{total}$	μ**W**	6.27	6.46
	WNS_SETUP	**ps**	0	0
	WNS_HOLD	**ps**	0	0

Die Area : 200 μm × 299.6 μm, Target CK Frequency: 20 MHz,
Clock Uncertainty: 30 ps, Clock-Gating applied
Process Corners: 1.2V/TT/25°C, 1.08V/SS/125°C, 1.32V/FF/-40°C
WNS_HOLD: Worst Negative Hold Slack
WNS_SETUP: Worst Negative Setup Slack

In the AES-128 macro, the S_18TSPC based design has slightly higher area overhead (0.5%) than the S_TGFF based implementation due to the higher numbers of inserted clock buffers in the clock tree. Accordingly, the P_{CK_net} of the S_18TSPC based macro is 15% higher than the S_TGFF design. Owing to this, the register and clock network

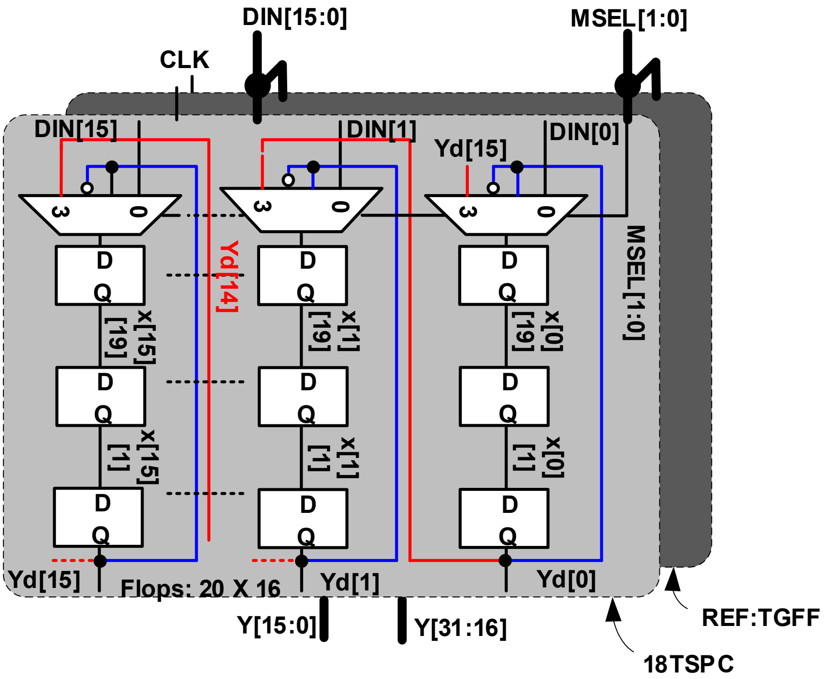


Figure 4.17: Block Diagram of the 320-bit Shift-Reg.

power (P_{REG}+P_{CK_net}) of S_18TSPC-based design is 15% lower than the S_TGFF-based macro with clock gating applied. Since the hold time of S_18TSPC is lower than 18TSPC, no hold time violation is observed. The P_{total} of the S_18TSPC based design is 3% lower than the S_TGFF-based implementation (vector based simulation).

According to the simulation results and analysis, 18TSPC shows advantages in power characteristics and cell area, and its EDA compatibility has been proved.

4.3 Experimental Validation

To validate the proposed design, the 18TSPC-based AES-128 macro, targeting nominal voltage operation (V_{dd} = 1.2V), was included in a test chip. As discussed in Section 4.2, the proportion of FF cells in the AES-128 block is limited (4%), so it can be difficult to show the power benefit of the proposed design. Therefore, to quantify the benefits of the FF in isolation, two 320-bit shift registers (18TSPC and TGFF-based) with synthesized clock trees were also implemented for nominal voltage operation (V_{dd} = 1.2V), with no hold buffers required between FF stages. Referring to the S2CFF and TGFF ED product (Table 4.2), TGFF was chosen as the reference design for comparison. The block digram is shown in Figure 4.17.

The fabricated test chip is shown in Figure 4.18a. A 32-bit Arm Cortex-M0 [152] microcontroller based test-board is shown in Figure 4.18b, which provides the state monitor,

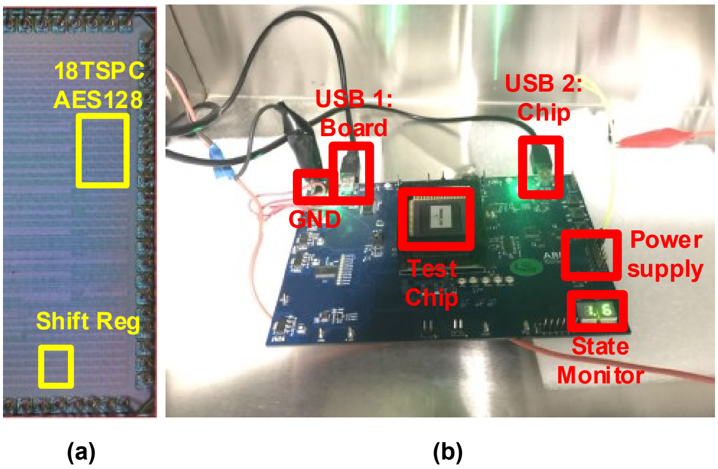


Figure 4.18: (a) Die micrograph: two blocks are built in the test chip, the AES-128 and the Shift Register (Shift-Reg). (b) Test Board.

power supply connections and USB interfaces for function monitoring, power measurement and communicating with the host computers, respectively.

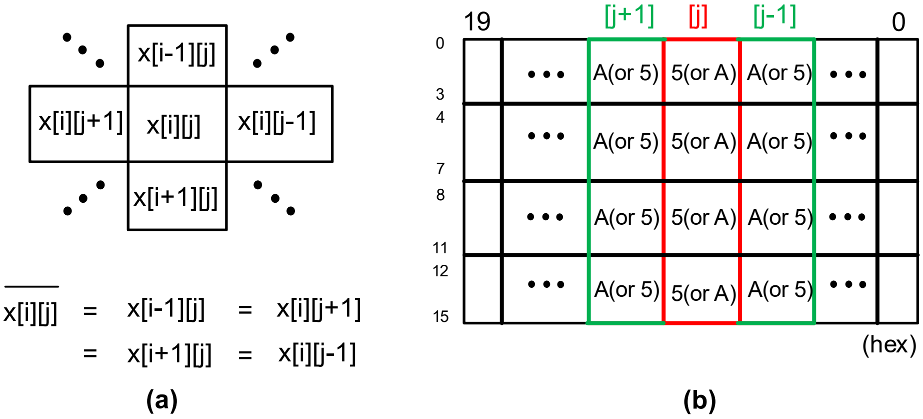
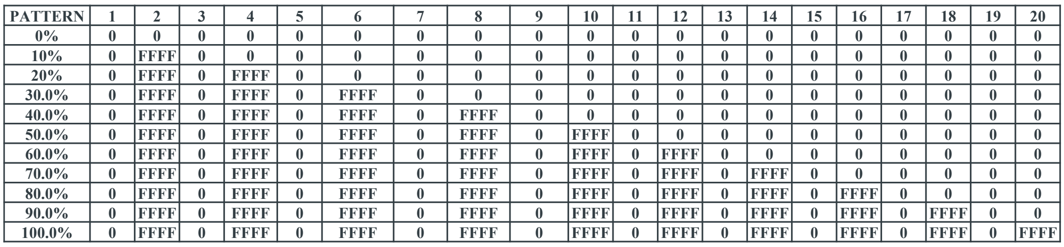


Figure 4.19: (a) model of random element of '20×16 matrix' (b) The functionality test pattern of the shift register .

The 320-bit shift register can be abstracted as a '20×16' matrix. An random element of the '20×16' matrix can be modelled as $X\left[i\right]\left[j\right]$, where $0 \leqslant i \leqslant 15$ and $0 \leqslant j \leqslant 19$. To fully test the functionality of the Shift register with full verification coverage, shift register can be initialized as the test patten in Figure 4.19. So that, the adjacent four elements $(X\left[i-1\right]\left[j\right], X\left[i+1\right]\left[j\right], X\left[i\right]\left[j-1\right] andX\left[i\right]\left[j+1\right])$ of a random element in the '20×16' matrix is equal and the logic state of the random element is always logical complement to the adjacent elements (shown in Figure 4.19a). Therefore, the content of an random column j of the matrix would be '5555(hex)' or 'AAAA(hex)'. and the adjacent column is $\overline{X[:][j]}$ ('AAAA(hex)' or '5555(hex)') (see Figure 4.19b). With the correct functionality, the shift register should maintain such correlation after a random time of data shifting in column ($\tau_{shifting}$). Therefore, after ($\tau_{shifting}$), the output of shift-reg ($REG_{out}(\tau_{shifting})$) and the next Shift-reg output ($REG_{out}(\tau_{shifting+T_{CK}})$) must still obtain the following boolean correlation (Equation 4.4 and 4.5):

$$1 = REG_{out}(\tau_{shifting}) \cdot (AAAA'h) + REG_{out}(\tau_{shifting}) \cdot (5555'h) \tag{4.4}$$

$$0 = REG_{out}(\tau_{shifting}) \cdot REG_{out}(\tau_{shifting+T_{CK}}) \tag{4.5}$$



PATTERN	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0%	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10%	0	FFFF	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20%	0	FFFF	0	FFFF	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30.0%	0	FFFF	0	FFFF	0	FFFF	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40.0%	0	FFFF	0	FFFF	0	FFFF	0	FFFF	0	0	0	0	0	0	0	0	0	0	0	0
50.0%	0	FFFF	0	FFFF	0	FFFF	0	FFFF	0	FFFF	0	0	0	0	0	0	0	0	0	0
60.0%	0	FFFF	0	FFFF	0	FFFF	0	FFFF	0	FFFF	0	FFFF	0	0	0	0	0	0	0	0
70.0%	0	FFFF	0	FFFF	0	FFFF	0	FFFF	0	FFFF	0	FFFF	0	FFFF	0	0	0	0	0	0
80.0%	0	FFFF	0	FFFF	0	FFFF	0	FFFF	0	FFFF	0	FFFF	0	FFFF	0	FFFF	0	0	0	0
90.0%	0	FFFF	0	FFFF	0	FFFF	0	FFFF	0	FFFF	0	FFFF	0	FFFF	0	FFFF	0	FFFF	0	0
100.0%	0	FFFF	0	FFFF	0	FFFF	0	FFFF	0	FFFF	0	FFFF	0	FFFF	0	FFFF	0	FFFF	0	FFFF

Figure 4.20: Shift register power measurement pattern (α = 0-100%).

Also, the activity rate (α) of the shift register can be initialized as specific patten. The Figure 4.20 shows the data pattern need to be initialized in shift register for $\alpha = 0\%$ to $\alpha = 100\%$.

The Python programming for shift Register chip measurement is attached in Appendix.

Figure 4.21a shows the measured normalized power vs α at 1.2V with maximum clock frequency of the board (F_{Board_MAX} = 66 MHz). At $\alpha = 0\%$, the total power is reduced by 68.5%. The average α of FFs in systems is typically 5% to 15% [109]. Measurement results show a 62.5% power saving at $\alpha = 10\%$. The benefits are retained at 0°C and 85°C. Figure 4.21b shows the measured power vs α at 0.6V at 25°C, measured results show that at $\alpha = 0\%$ the total power saving is increased to 73% and at $\alpha = 10\%$ the power saving is increased to 68%.

Figure 4.22a shows the measured power with $\alpha = 100\%$ at different V_{dd}. The clock frequency is set as 66 MHz (F_{Board_MAX}). As V_{dd} decreases to 0.85V, the 18TSPC-based Shift-Reg ceased to work at 66 MHz. Due to the performance penalty (Figure 4.14), 18TSPC needs to work at lower frequency when $V_{dd} < 0.85V$. For TGFF, with better D-to-Q delay characteristic, functionality was maintained with a 66 MHz clock frequency down to 0.65V. From the result, it can be seen that the proposed 18TSPC saves 39% power at 1.2V and the power benefit is maintained as V_{dd} is decreased until the TGFF register fails at 0.65V. The power benefit with $\alpha = 0\%$ ($P_{\alpha=0\%}$) (CK pin dynamic power dominant) is shown in Figure 4.22b. At 1.2V, the $P_{\alpha=0\%}$ of 18TSPC is 68.7% less than the reference TGFF. At the minimum V_{dd} of 18TSPC for 66 MHz clock frequency operation (V_{dd} = 0.85V), the $P_{\alpha=0\%}$ saving increased to 69.4%. Although Figure 4.22a shows that total power is equivalent for both designs at their minimum operating voltage

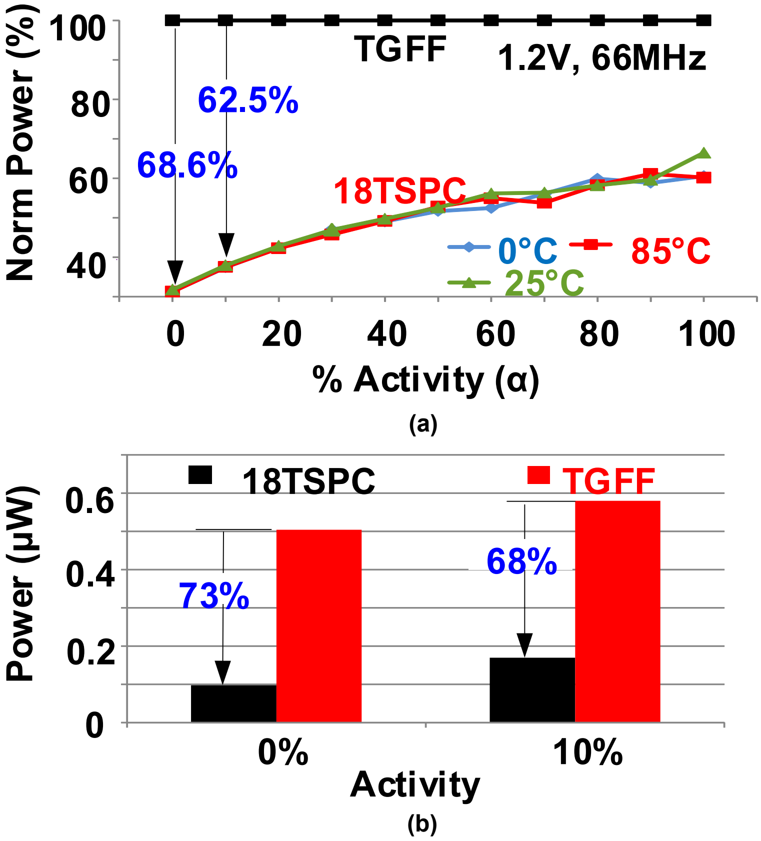


Figure 4.21: Measured power of 320-bit Shift-Reg against α at (a) 1.2V (with $0°C$, $25°C$ and $85°C$) (b) 0.6V with $25°C$

(TGFF = 0.65V and 18TSPC = 0.85V) with fixed frequency (F_{Board_MAX}), note that the $P_{\alpha=0\%}$ of 18TSPC at 0.85V is still 54.3% less than TGFF at 0.65V.

Logic Built-In Self-Test (LBIST) is applied to 18TSPC-based AES-128 for functional test, active power and maximum frequency measurements. Figure 4.23 shows the total power of the AES-128 macro at different supply voltages with the respective maximum clock frequency. Leakage power is also measured at various supply voltages. Although the AES-128 is functionally correct at 0.6V, the results with acceptable clock frequencies ($F_{CK} > 0.1$ MHz) are shown. The minimum operating voltage and respective maximum F_{CK} of the 18TSPC AES-128 macro is 0.7V with 0.81 MHz. The leakage power at 0.7V is 62 nW. For 1.2V operation, the test macro shows a maximum F_{CK} of 56 MHz with 2.3 mW active power consumption and the leakage power at 1.2V is 390 nW.

Figure 4.24 shows the measurement on minimum functional voltage (V_{min}) of the Shift-Reg over 92 test chips. Note that the on-chip macro design was targeted for 1.0-1.2V operation, but the measurement results show a mean V_{min} of 0.63V. The functionality at low voltage is mainly limited by the increased hold time of the FFs. To enable lower voltage operation, hold buffers should be inserted between stages during macro implementation.

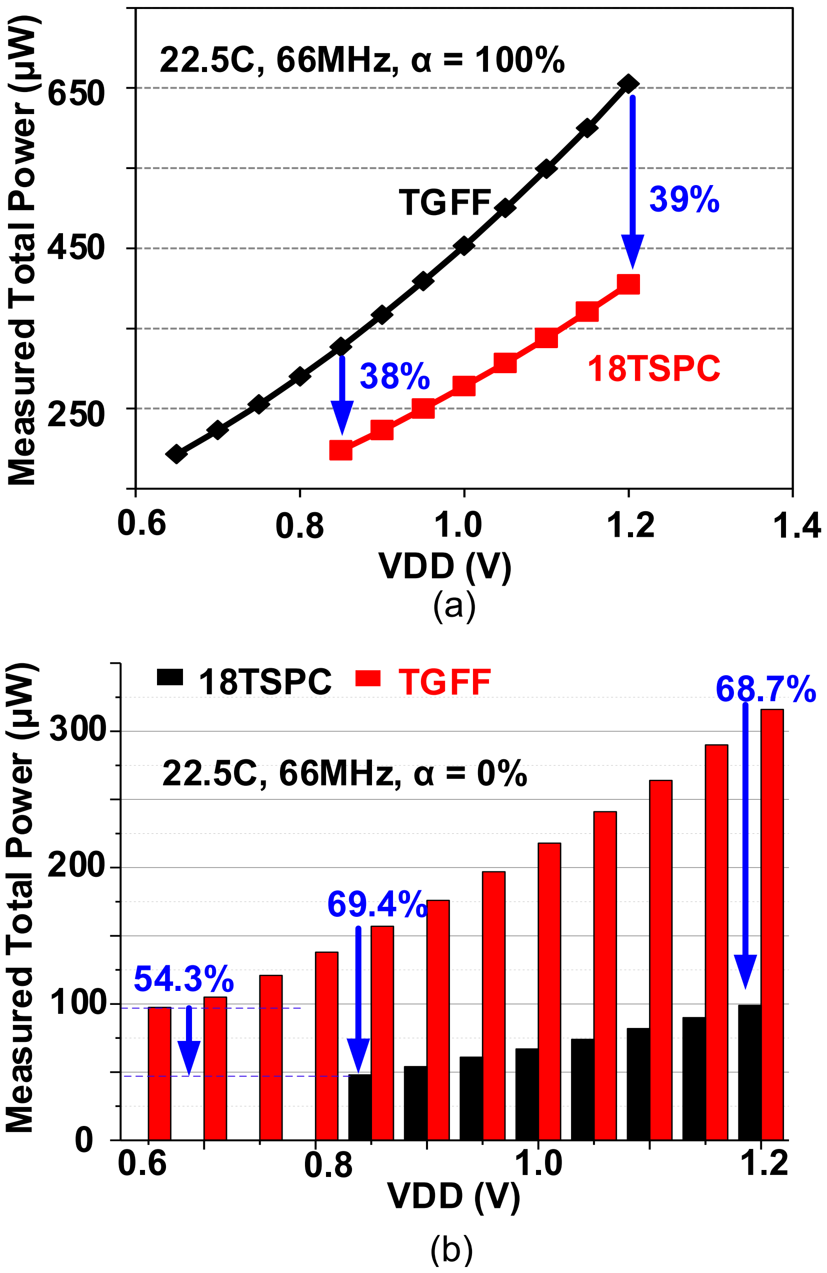


Figure 4.22: Measured total power of 320-bit Shift-Reg with (a) $\alpha = 100\%$ (b) $\alpha = 0\%$ with fixed clock frequency ($F_{Board\_MAX} = 66$ MHz) at different supply voltage.

For the temperature-related measurements, the chip was placed in a temperature chamber. The temperature effects on the functional V_{min} of AES-128 and Shift-Reg with 0.1 MHz clock frequency are shown in Figure 4.25. Owing to their higher sensitivity to the decreased threshold voltage induced by higher temperature at low V_{dd} (leading to decreased gate delay and stronger temperature inversion effects [21]), for both blocks the functional V_{min} is decreased as the temperature increases. The AES block is a combinational logic-dominant circuit which brings a variety of hold paths with some containing

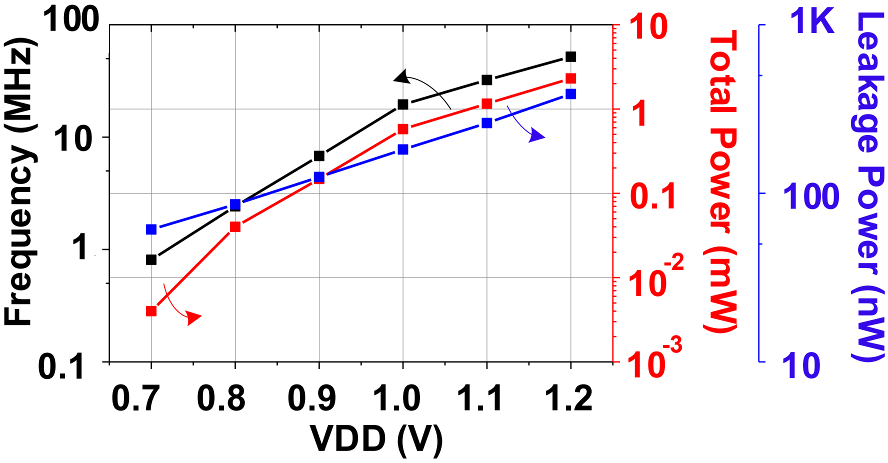


Figure 4.23: Measured results of the 18TSPC AES-128 block (Typical Die).

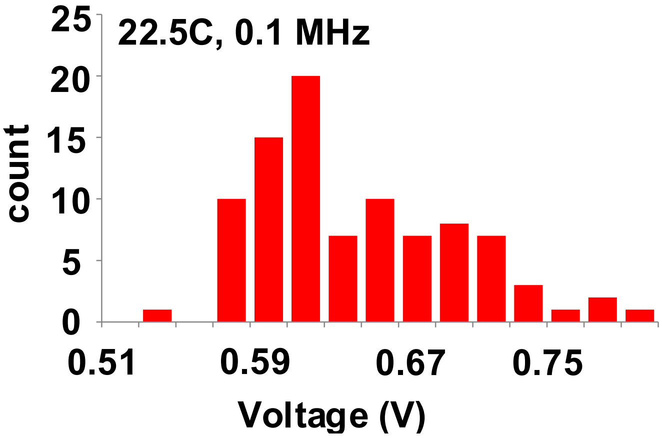


Figure 4.24: V_{min} distribution of Shift-Reg over 92 test chips.

multiple 2-stack gates, which makes the V_{min} of the AES is more sensitive to temperature. Only one type of hold path exists in the shift register, meaning that temperature has less effect on its V_{min} ($\Delta V_{min} = 70mV$ over 0 °C - 80 °C) compared to the AES-128 ($\Delta V_{min} = 120mV$ over 0 °C - 80 °C).

4.4 Discussion of 18TSPC

Previous sections proposed 18TSPC, a fully-static and contention-free SPC FF with the lowest reported number of transistors (18), demonstrating a 20% cell area reduction with respect to the conventional TGFF. Fewer devices also result in 27% lower leakage. With a $MinE$ driven circuit implementation, the proposed design has a higher D-to-Q delay and hold time than TGFF. Although a performance penalty is observed, thanks to the low power characteristic of the proposed design, 18TSPC achieves 1.8× better ED product. Chip measurement results show a 62.5% reduction in overall power at $\alpha = 10\%$, and a 68% reduction in $P_{\alpha=0\%}$ at 1.2V, 25°C. When V_{dd} scales down to NTV

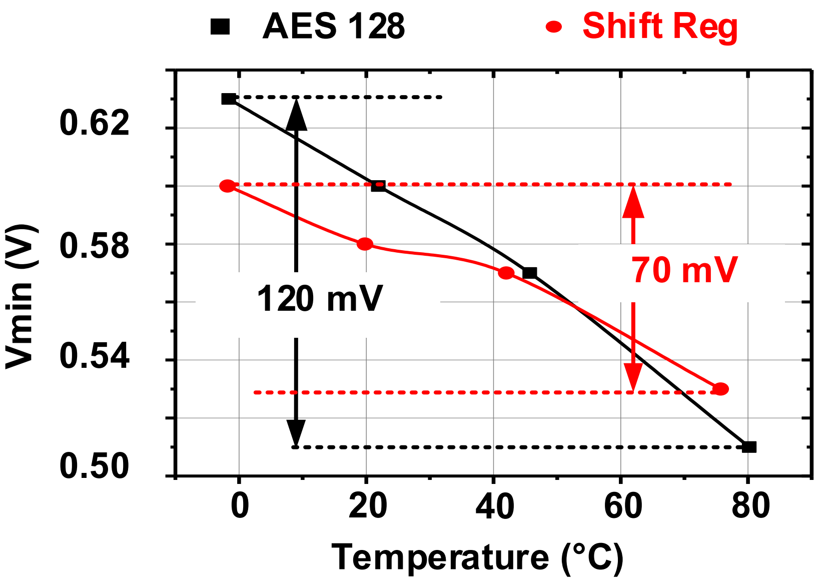


Figure 4.25: Functional V_{min} of AES-128 block and Shift-Reg with 0.1 MHz clock frequency at different temperature condition.

level ($V_{dd} = 0.6$ V), the overall power benefit at $\alpha = 10\%$ increases to 68% and the $P_{\alpha=0\%}$ benefit increased to 73% compare to the conventional TGFF. Also, the chip test with an AES-128 macro proves the compatibility of the proposed 18TSPC for automatic EDA implementation based on standard cells. A brief summary of the proposed 18TSPC and comparison with prior-works is shown in Table 4.5. The proposed 18TSPC has better power characteristics than the SoA S2CFF design.

As it was introduced in the previous section, some facts about the proposed 18TSPC need to be aware. The proposed 18TSPC has four clock transistors directly connect to the global clock tree. Although such scheme shows the advantage in power efficiency, the clock pin capacitance of the 18TSPC is double of the TGFF at the cell level (consider the FF as a built-in block). So that, the design could potentially degrade clock slew rates when the flops are employed as part of a very large design, resulting in degraded performance. It can be observed from the results present in Table 4.3 and 4.4. The number of clock buffers in SPC FFs (18TSPC and S2CFF)-based is higher than the conventional TGFF-based design.

It also needs to be aware that the 18TSPC has the positive Hold time value due to the topology (introduced in Section 4.1.3. This made the proposed design might be less suitable for further aggressive voltage scaling to STV region (where V_{dd} < V_{th}). This can potentially cause higher stress for system level timing closure in STV region, i.e. More Hold buffers need to be inserted to fix the hold time violation.

In the next session, some extension SPC FF schemes based on 18TSPC are proposed

Table 4.5: summary of Comparison with prior-works

FF Design	**18TSPC**	**TGFF**	**S2CFF**	**TCFF**
Year	This work	std-cell	ISSCC'14 [65]	ASSCC'13 [110]
Technology (Reported)	65nm	-	45nm SOI	40nm
Type	static	static	static	static
Contention	No	No	No	No
Single-Phase	Yes	No	Yes	Yes
Complementary Topology	Yes	Yes	Yes	Yes
Output Inverter	Yes	Yes	Yes	Yes
Poly Biasing	No	No	No	No
FBB/RBB	No	No	No	No
Transitors CK/Total	4/18	12/24	5/24	3/21
Norm. Power @10% α	0.32	1	0.6	0.34
Setup (ns)**	9.2	4.66	14.7	137
Hold (ns)**	11	-2.9	-10.2	-8
CK-to-Q (ns)**	14.6	14.8	14.5	13.4

FF Design	**ACFF***	**XCFF***	**TSPC-18T***
Year	ISSCC'11 [109]	VLSI'05 [109]	TCASI'18 [111]
Technology (Reported)	40nm	100nm	28nm FDSOI
Type	static	dynamic	semi-dynamic
Contention	partial	yes	yes
Single-Phase	yes	yes	yes
Complementary Topology	yes	yes	no
Output Inverter	yes	no	no
Poly Biasing	no	no	Yes
FBB/RBB	no	no	Yes
Transitors CK/Total	4/22	4/21	4/18
Norm. Power @10% α	0.4	1.2	0.42

* Not Implemented
** 18TSPC, TGFF, S2CFF, TCFF are characterised with 65nm SS/0.54V/25°C
Text in red: Design limitation

for addressing the specific limitations in 18TSPC and targeting different design requirements.

4.5 Extending SPC FFs design based on 18TSPC

4.5.1 The 20-Transistors SPC FF Design

As it is discussed in the previous section, SPC FFs has higher clock pin capacitance which could potentially degrade clock slew rates when the SPC FFs is employed as part of a larger design resulting in the stress on CK tree. A 20-Transistors (20T) SPC FF is proposed in Figure 4.26. In the proposed flop, the clock is inverted internally to lower capacitance as seen from the clock driver.

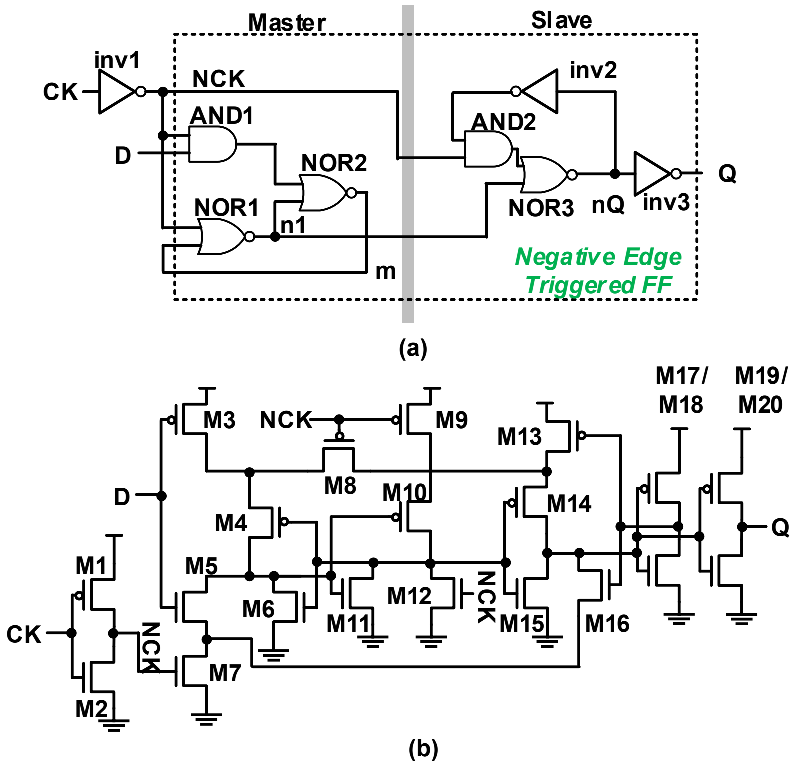


Figure 4.26: (a) Gate level schematic diagram (b) The Transistor level schematic diagram of the proposed 20TSPC FF.

Figure 4.26(a) shows the gate level schematic of the proposed design. Same as the basic abstract of 18TSPC, the 20TSPC is also based on the MUX2-based MS FF. However, the 20TSPC can be viewed as a negative edge triggered SPC FF clocked by 1 stage buffered (invl) clock (NCK). The negative edge triggered SPC FF is the complement topology of the 18TPSC gate level schematic shown in Figure 4.4(a). Transistor level schematic diagram of 20TSPC is shown in Figure 4.26(b).

Figure 4.27 illustrate the layout of the 20TSPC. Comparing to the TGFF, the 20TSPC shows 8% area saving. The post-layout analysis shows the 20TSPC achieves 15% leakage power reduction. With the clock buffer, the CK pin capacitance achieves 1.5 times reduction from the baseline design 18TSPC.

Table 4.6 shows the comparison of the proposed 20TSPC with other FFs (18TSPC, S2CFF and the conventional TGFF). At 10% α, Post-Layout simulation result shows the proposed 20TSPC reduce the power by 58% from TGFF and 18% from S2CFF. However, the proposed 20TSPC increased the power by 10% compared to the baseline design 18TSPC. The Setup time of 20TSPC is reduced by 65% and 52% from S2CFF and 18TSPC respectively which is benefit from the extended CK signal path in proposed 20TSPC. However, 20TSPC topology has a longer D to Q path (1 stage longer than

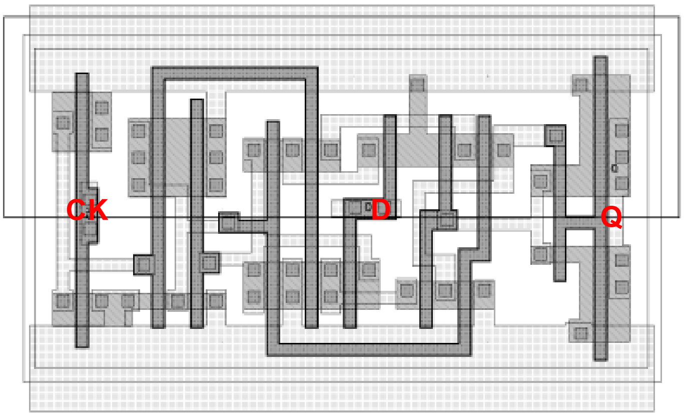


Figure 4.27: The layout of the proposed 20TSPC; Clock pin capacitance: 1.45 fF; P_{Leak} at 1.2V/TT/25°C: 122 pW; Area: 2.4×4.6 μm^2.

Table 4.6: Comparison of SPC FFs in terms of power and timing characteristics (I)

FF Design	TGFF	S2CFF	18TSPC	20TSPC
Transistors CK/Total	12/24	5/24	4/18	6/20
Norm. Power at 10% α	1	0.6	0.32	0.42
Setup (ns)**	4.66	14.7	9.2	4.4
Hold (ns)**	-2.9	-10.2	11	18
CK-to-Q (ns)**	14.8	14.5	14.6	25.8
D-to-Q (ns)**	19.46	29.2	23.8	30.2

** Data are extracted with 65nm SS/0.54V/25°C

18TSPC). Therefore, the 20TSPC has the highest D-to-Q delay in 4 designs. It also needs to be aware that the 20TSPC has higher hold time at NTV which is also owing to the extended CK signal path (the added CK buffer).

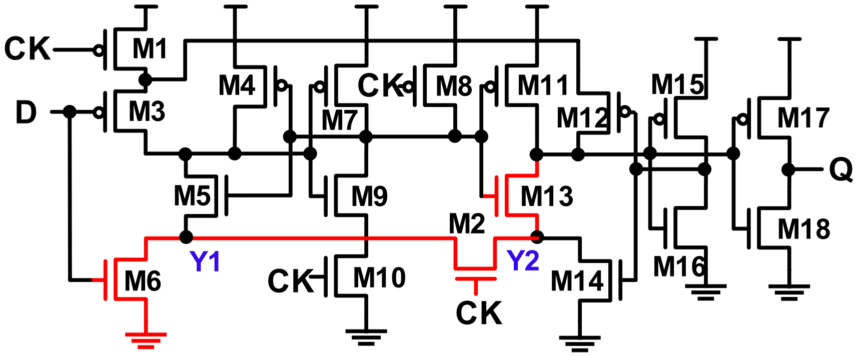
To achieve lower CK pin capacitance with single-phase clock operation, 20TSPC is one of the potential solutions. However, employing the proposed 20TSPC in the system means a trade-off between the performance and power efficiency requirements need to be considered.

4.5.2 The 19 and 21-Transistor SPC FF Design

The hold time, as one of the most critical parameters of FF designs, has a significant impact on the system regarding power, performance, and area. This can be exacerbated at NTV due to the increased logic transition delay. As it was mentioned before, both proposed 18TSPC has the positive hold time value owing to the topology. The 20TSPC also have such timing characteristic since the topology design just complementary logic of 18TSPC. With V_{dd} scaling down, the hold time of 18TSPC and 20TSPC is tending to shift to higher positive value. Due to such timing characteristic, hold buffers required to be asserted into the during macro implementation phase to fix the hold violation at

NTV. Consequently, although it is controllable, unexpected power and area resource is sacrificed by the hold buffers. In this section, a 19-Transistor (19T) SPC FF is proposed to address the limitation of baseline design 18TSPC. However, a non-negligible glitch can be observed in 19TSPC owing to the modification. To eliminate the glitch and improve the robustness of the low power design, a 21-Transistor (21T) SPC FF is proposed in this section.

For improving the hold time characteristics of the FF design, the research direction should focus on optimizing the worst-case hold time scenario. In Section 4.1.3, it was introduced that the worst-case hold time is experienced when D falls too close to the rising edge of CK. If the M6 in Figure. 4.9(a) is turned off by D before net L2 is fully pulled down to 0, a hold violation , and the speed of the L2 been pulling down is the determinant of the worst-case hold time of 18TSPC.



***3 stack topology exists: M13, M2 and M6 path.**

Figure 4.28: 3 stacks pull down path exists in 18TSPC.

As it was introduced in Figure 4.9(a), the L2 is pulled down to 0 via the path M13, M2, and M6. Figure 4.28 highlight these transistors and it can be considered as a critical three stack (NMOS) topology. Although the three stack transistors can be sized up to improve the strength of L2 pulling down, a more efficient method is breaking the bridge transistor (M2) between Y1 and Y2. One transistor needs to be added to the topology to ensure the Y1 and Y2 be pull down by separate CK transistor. Therefore, the total transistor count of the design becomes 19, named as 19TSPC. The pull-down path of L2 is reduced to 2 stacks from 3 stacks. The transistor schematic diagram is presented in Figure 4.29

The layout of the 19TSPC FF is presented in Figure 4.30. Comparing to the TGFF, the 19TSPC shows a 16% area saving. The post-layout analysis shows the 19TSPC achieves 21% leakage power reduction. Comparing to the 18TSPC, one more clocked transistor is added resulting five transistors are clocked in 19TSPC. Therefore, the CK pin capacitance of 19TSPC is 2.7 times of TGFF.

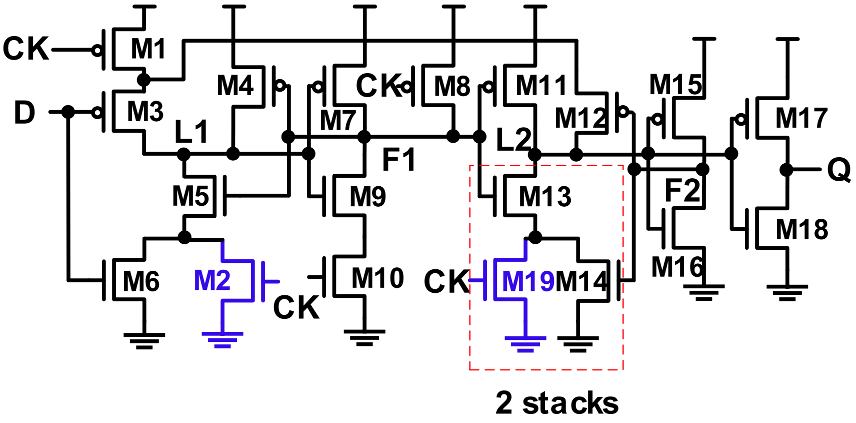


Figure 4.29: The transistor level schematic diagram of the 19 Transistor (19T) SPC FF, maximum stack is optimised to 2.

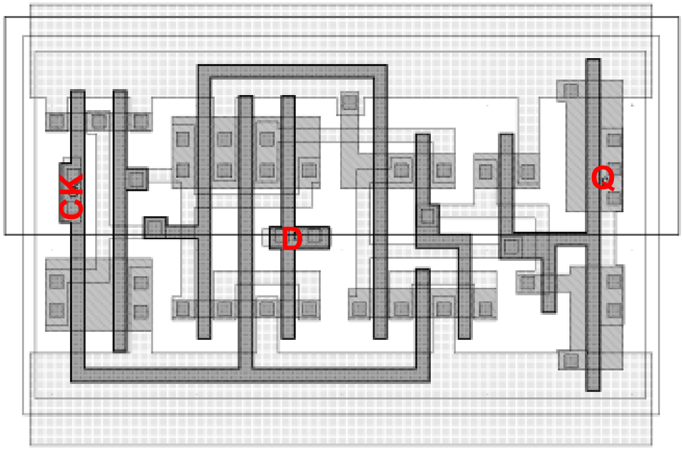


Figure 4.30: The layout diagram of the 19TSPC. Clock pin capacitance: 2.99 fF; P_{Leak} at 1.2V/TT/25°C: 113 pW; Area: 2.4×4.2 μm^2.

Table 4.7 shows the comparison of the proposed 19TSPC with other FFs (18TSPC, 20TSPC, S2CFF and the conventional TGFF). At 10% α, Post-Layout simulation result shows the proposed 19TSPC reduce the power by 63% from TGFF and 23% from S2CFF. With the topology modification, the Setup time of 19TSPC is reduced by 39% from S2CFF. Also, 19TSPC topology modification shortens the D to Q path from the baseline design 18TSPC. Therefore, the 19TSPC has a shorter D-to-Q delay in 4 SPC FFs, which is 32%, 16% and 34% faster than S2CFF, 18TSPC and 20TSPC respectively. More important, with the proposed modification, the hold time became negative at NTV level. The observed potential hold time issues in 18TSPC and 20TSPC are fixed in 19TSPC.

However, with the PVT considered simulation, a non-negligible glitch can be observed on internal node L1 of the 19TSPC when FF write-in 0 (Initial Q =0, D = 0 at CK rising). The internal node operation waveform and the glitch on L1 are illustrated in Figure 4.31. At 0.6V/SS/0°C, a 150 mV voltage degradation can be observed at L1 just

Table 4.7: Comparison of SPC FFs in terms of power and timing characteristics (II)

FF Design	TGFF	S2CFF	18TSPC	20TSPC	19TSPC
Transistors CK/Total	12/24	5/24	4/18	6/20	5/19
Norm. Power at 10% α	1	0.6	0.32	0.42	0.37
Setup (ns)**	4.66	14.7	9.2	4.4	9.01
Hold (ns)**	-2.9	-10.2	11	18	-2.11
CK-to-Q (ns)**	14.8	14.5	14.6	25.8	10.98
D-to-Q (ns)**	19.46	29.2	23.8	30.2	19.99

** Data are extracted with 65nm SS/0.54V/25°C

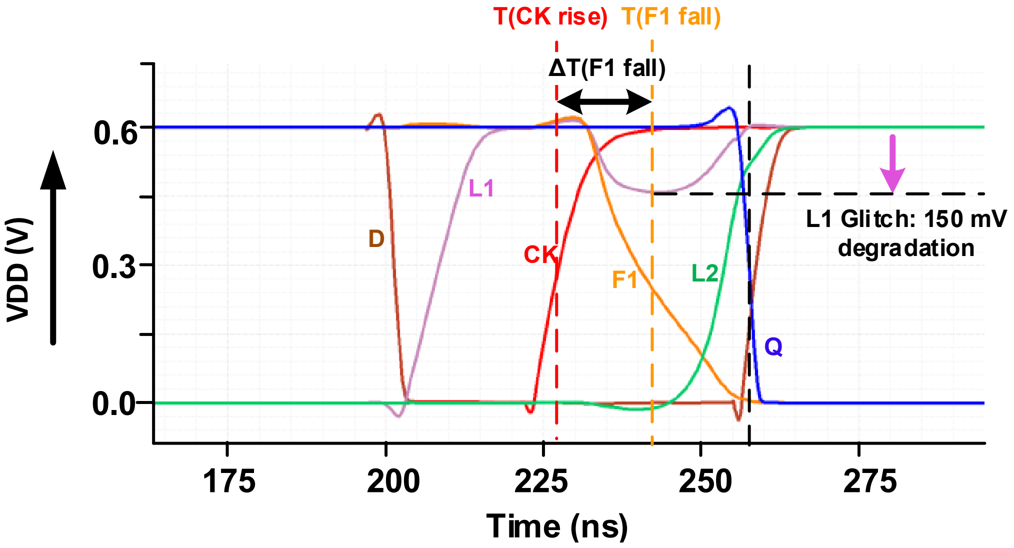


Figure 4.31: Operation waveform of 19TSPC at 0.6V/SS/0°C, an non-negligible glitch is observed in 19TSPC internal node L1 (see Figure 4.29).

after the CK switches to 1. With the fixed transistor electrical parameters, the glitch is recovered in 10ns. When 10K MC simulation at 0.6V/SS/0°C is applied to the circuit, the failure rate is 0.05%. Therefore, for a macro with 435 FFs, the block yield is 80.4%.

The analysis of the non-negligible glitch is presented in Figure 4.32. Assume the previous F1 is 1 (Q = 1), and D = 0 wrote in FF at CK rising edge. When CK = 0 and D falling, L1 is pulled up to 1 and M9 is on, F1 is 1 (M8 on and M10 off). When CK rising at $T(CKrise)$, F1 turns M5 off and the M4 is on at $T(F1fall)$ after the transition time of F1 pulled down ($\Delta F1\_fall$). The F1 transition timing can be modelled as Equation 4.6:

$$T(F1fall) = T(CKrise) + \Delta F1\_fall \tag{4.6}$$

From the model, it can be realised that the F1 falling always happens after the CK rising with $\Delta F1\_fall$ lagging. Therefore, F1 is weak 1 for $\Delta F1\_fall$ which keeps M5 partially on. L1 is temporarily pulled down via path M5 and M2 which lead to a 150

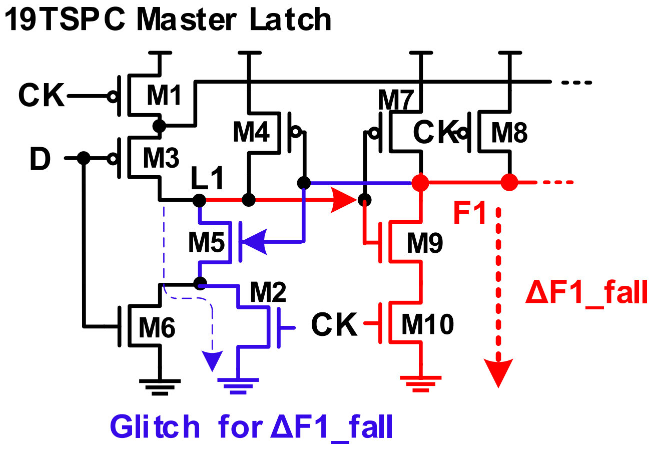


Figure 4.32: 19TSPC master latch operation path analysis.

mV voltage degradation (see waveform Figure 4.31). After $\Delta F1\_fall$, M5 off and M4 is on, the glitch is recovered to 1, F1 and L1 is stable.

One possible solution to minimizing the L1 glitch is speeding up the F1 falling speed by upsizing the M9 and M10. Accordingly, the F1 falling time $\Delta F1\_fall$ can be reduced. Although the $\Delta F1\_fall$ is minimized, it can not change the fact that F1 falls after CK rising edge and a weak F1 can be generated during $\Delta F1\_fall$. Rather than optimize the design in transistor size tuning, a better solution is modifying the master latch topology. The objective of the modification is breaking the path of the L1 pulling down via M5, M2 and increase the feedback speed from F1 to L1.

To achieve the objective, the boolean function of $\overline{L1}$ is modelled in Equation 4.7

$$\overline{L1} = F1 \cdot (D + CK) \tag{4.7}$$

According to the principle of of duality, the boolean function can be written in the form of Equation 4.8:

$$\overline{L1} = F1 \cdot D + F1 \cdot CK \tag{4.8}$$

So that, the schematic of the L1 can be modified as Figure 4.33.

The boolean function of F1 is $\overline{CK} + \overline{L1}$. So that, the Equation 4.8 can be re-write in form of Equation 4.9. And the Equation 4.9 can further deriving to Equation 4.10:

$$\overline{L1} = F1 \cdot D + (\overline{CK} + \overline{L1_{previous}}) \cdot CK \tag{4.9}$$

$$\overline{L1} = F1 \cdot D + \overline{L1_{previous}} \cdot CK \tag{4.10}$$

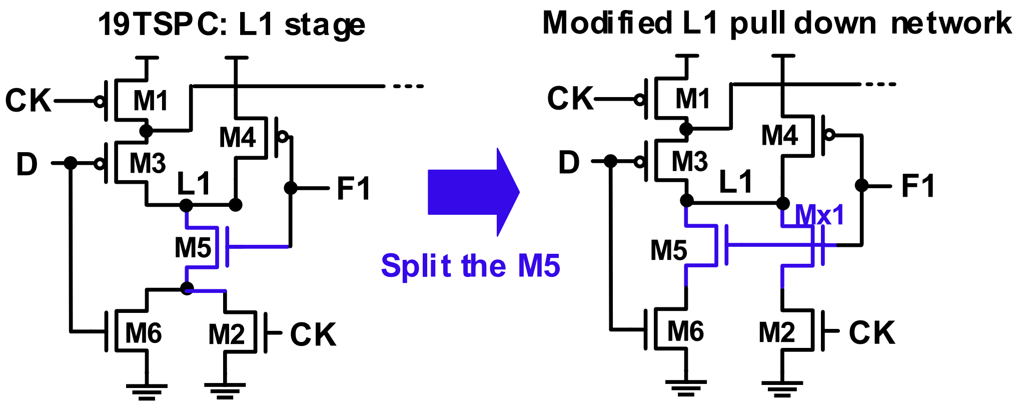


Figure 4.33: L1 node pull-down network re-built.

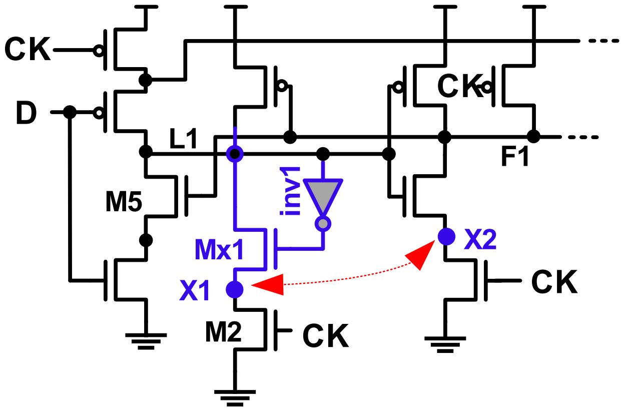


Figure 4.34: Master latch of the 22-Transistors SPC FF.

According to the boolean function, the schematic of the L1 pull-down network can be reconstructed as it is shown in Figure 4.34. One NMOS and one inverter are added to the master latch, which increases the total transistor count of the modified SPC FF to 22. To further reduce the number of clocked transistors, a transistor merging process is applied to the 22-Transistor SPC FF (4.34). When CK = 0, NMOS M2 and M10 is off, F1 is always 1. So that, L1 = $\overline{D}$. X1 always equal to 0. X2 depends on L1 and it only equal to weak 1 when L1 is 1. For CK = 1, X1 and X2 always equal to 0 (see Table 4.8). Based on the observation, the clocked transistor M2 and M10 can be merged as a single clock transistor. Although X2 is 1^w when CK = 0, D = 0. The data on F1 cannot be diffused back to L1 since the NMOS Mx1 is off. The schematic diagram of the optimized design is present in Figure 4.35. The total transistor count is 21.

The layout of the proposed 21TSPC is shown in Figure 4.36. Comparing to the TGFF, the 21TSPC shows a 4% area saving. The post-layout analysis shows the 21TSPC saves

Table 4.8: Net states at $X1$ and $X2$ at different D and CK states

CK	**0**	**0**	**1**	**1**	**1**	**1**
D	0	1	0	1	0	1
F1	1	1	0	0	1	1
L1	1	0	1	1	0	0
Mx1	off	on	off	off	on	on
M9	on	off	on	on	off	off
X1	0	0	0	0	0	0
X2	1^w	0	0	0	0	0

For CK = 1 : X1 = X2 = 0
For CK = 0: L1 = $\overline{D}$, X1 = $\overline{L1} \cdot \overline{D} = 0$
X2 depends on L1 (X2 = $L1 \cdot 1^w$)
1^w : Weak 1 (V_{th} drop)

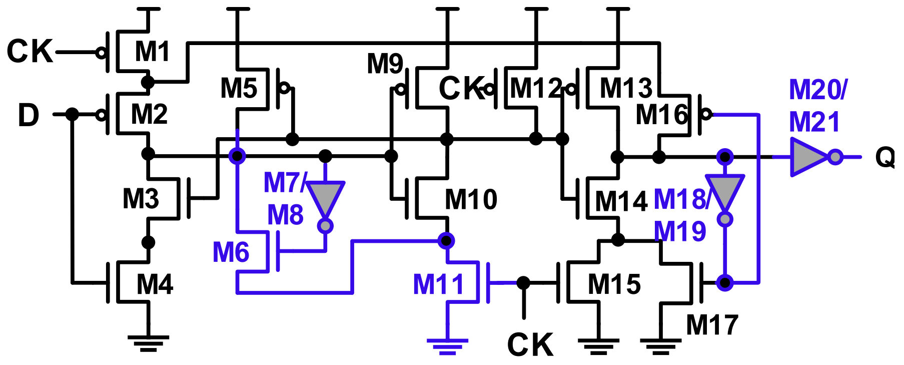


Figure 4.35: The Transistor level schematic diagram of the proposed 21TSPC FF.

19% leakage power from TGFF. The CK pin capacitance is two times of TGFF since four transistors are connected to global CK tree.

Figure 4.37 shows the operation waveform of the 21TSPC FF at 0.6V/SS/0°C (same test environment as launched to 19TSPC). From the post-layout simulation, it can be seen that the non-negligible glitch is Effective relieved. The magnitude of the glitch is reduced to 38 mV (6% V_{dd}) from 150 mV (25% V_{dd}), which makes the glitch negligible. The proposed 21TSPC passed all 10K MC simulation.

The timing and power data are compared with other unbuffered CK SPC FFs (S2CFF, 18TSPC, and 19TSPC) and TGFF. At 10% α, the Post-layout simulation results shows the 21TSPC reduced the power by 55% from TGFF and 15% from S2CFF. Owing to the topology modification in the master latch, the setup time of 21TSPC is 12.4 ns. Accordingly, the D-to-Q delay of the 21TSPC is longer than the 18TSPC and 19TSPC. Since the 21TSPC is developed based on the 19TSPC, the hold time is negative.

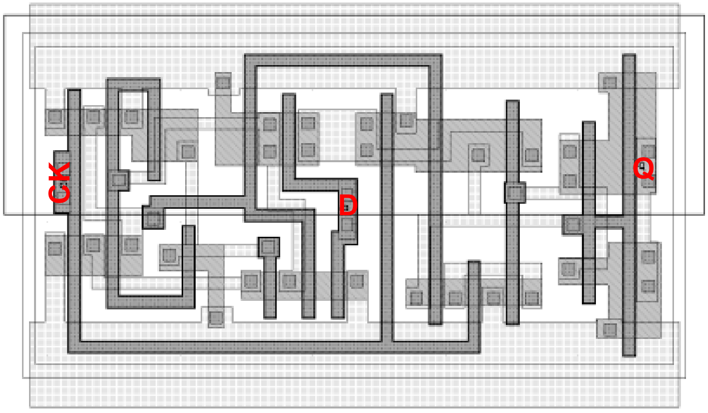


Figure 4.36: Layout diagram of the 21TSPC FF. Clock pin capacitance: 2.68 fF; P_{Leak} at 1.2V/TT/25°C: 116 pW; Area: 2.4×4.8 μm^2.

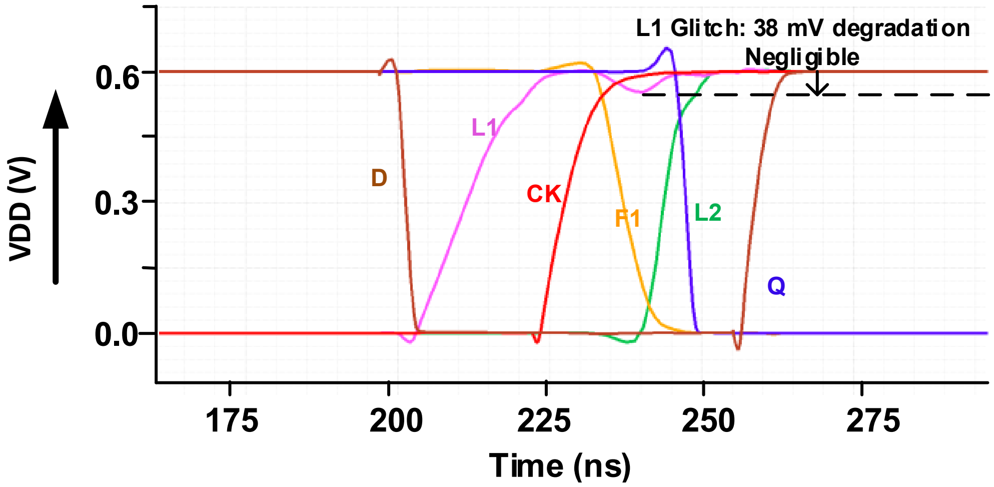


Figure 4.37: Operation waveform of the 21TSPC FF at 0.6V/SS/0°C, the glitch in 19TSPC internal node L1 is negligible

To further evaluate the robustness of the 21TSPC in term of timing, a 10K MC simulation is applied to the reference design TGFF and unbuffered SPC FFs. Figure 4.38 shows the hold time standard deviation (σ_{Hold}) comparison at 0.54V/SS/0°C. From the results, it can be seen that the 21TSPC shows the best hold time robustness at worst case process corner among all proposed SPC FFs. The 21TSPC shows a 13% and 19% less hold time variation compare to S2CFF and TGFF respectively.

All of the proposed SPC FFs were used to implement AES-128 macros target to NTV region (0.6V/TT/25°C). Figure 4.39 shows the Energy/cycle of AES-128 based on different FFs. As it was introduced before, AES-128 is not a FF heavy circuit and the combinational logic power/energy consumption dominate the overall power/energy consumption. So that, the overall power/energy saving of our proposed design from the conventional TGFF based design is only 1%-10%. The 18TSPC achieves the minimum energy consumption among 6 FFs which saves 10% from a TGFF-based macro. The buffered CK SPC FF, 20TSPC, is the second energy efficient SPC FF. However, sharp

Table 4.9: Comparison of SPC FFs in terms of power and timing characteristics (III)

FF Design	TGFF	S2CFF	18TSPC	19TSPC	21TSPC
Transistors CK/Total	12/24	5/24	4/18	5/19	4/21
Norm. Power at 10% α	1	0.6	0.32	0.37	0.45
Setup (ns)**	4.66	14.7	9.2	9.01	12.4
Hold (ns)**	-2.9	-10.2	11	-2.11	-0.81
CK-to-Q (ns)**	14.8	14.5	14.6	10.98	11.95
D-to-Q (ns)**	19.46	29.2	23.8	19.99	24.35

** Data are extracted with 65nm SS/0.54V/25°C

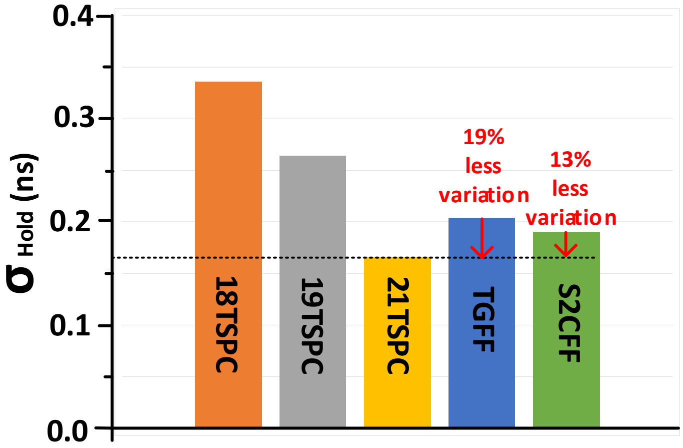


Figure 4.38: Hold time standard deviation (σ_{Hold}) comparison at 0.54V/SS/0°C.

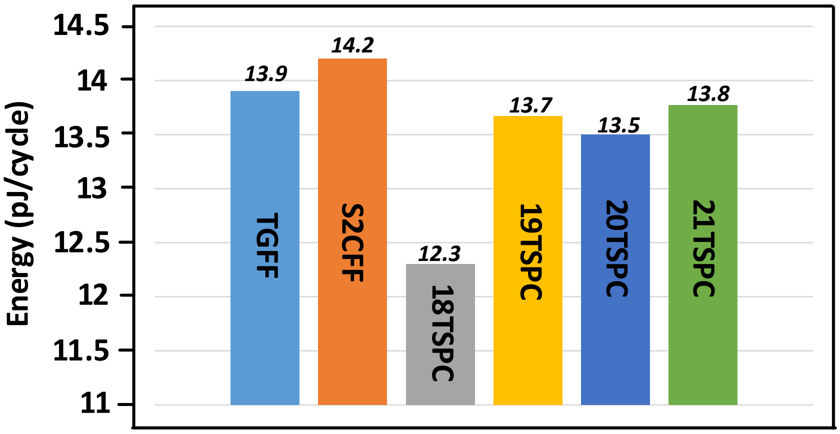


Figure 4.39: Energy per cycle of AES-128 block based on different FF at 0.6V/TT/25°C.

energy consumption increased from 18TSPC-based design can be observed in 20TSPC. This is owing to the high setup time of the 20TSPC. More CK buffers are inserted to the block which leads to an increase in combinational logic power/energy consumption. So that, the 20TSPC saves the energy from TGFF-based design is 3%. 19TSPC-based AES consumes higher energy than the 20TSPC-based design because the overall CK pin capacitance is highest among 6 FFs. Higher stress on global clock tree and higher CK network power consumption can be observed in 20TSPC-based which makes the 20TSPC-based design has higher energy consumption and the energy saving from TGFF of the 20TSPC-based design is 2%. 21TSPC has highest transistor counts, biggest in area among 4 proposed SPC FFs. Although 21TSPC has the same CK transistor count as 18TSPC, the CK pin capacitance of 21TSPC is 20% higher than 18TSPC. Because the 21TSPC has longer poly routing path in layout (see Figure 4.36) owing to the more complex topology than 18TSPC.

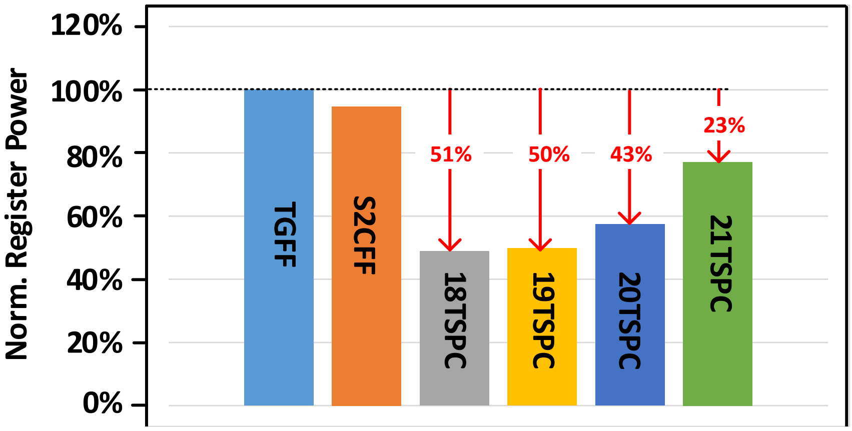


Figure 4.40: Parallel comparison of normalized AES register power consumption.

Figure 4.40 shows VCD simulation results of the AES-128 register power consumption based on different SPC FFS. It can be considered that our proposed FFs have superior power efficiency characteristic at the system level. 18TSPC is the most power efficient design among four flops which saves 51% power from TGFF. Although the 21TSPC has the highest power consumption among four SPC design, it still capable of saving 23% power from TGFF.

4.6 Concluding Remarks

FFs are considered as the key component in model sequential logic circuits. Research shows that the FFs have a great impact on performance, robustness, area and total power efficiency of the system. When the NTV operation low power technique is applied to the system, the requirement of the ultra-low power FF design with high quality becomes more obvious. As it was introduced in this chapter, the ultra-low power FFs should be fully-static, contention-free and area reduced. However, none of the SoA and existing topology meet all the above requirements and their claimed benefits can reduce significantly as yield, EDA and system level issue are addressed.

In this chapter, I proposed the 18-Transistor Fully Static Contention Free Single Phase Clocked Flip Flop which firstly meets all of the requirement. And the proposed design has the lowers reported transistor count (only 18) for a fully-static contention-free SPC FF. Benefit from the simple topology, it demonstrates a 20% cell area reduction with respect to the conventional TGFF. From the post-layout characterization, it also shows a 27% lower leakage than TGFF. With the fair sizing strategy ($MinE$ driven), the proposed design achieves 1.8× better ED production than the reference design, TGFF. Chip measurement results show a 62.5% reduction in overall power at system typical data activity rate and 68% power reduction when data activity rate is 0% at 1.2V with ambient temperature environment. When V_{dd} enters the NTV region (V_{dd} = 0.6V), the benefit is further extended (68% reduction when α = 10% and 73% reduction when α = 10%. Also, the chip test with an AES-128 macro proves the compatibility of the proposed 18TSPC for on-die test circuit implementations. Based on our experimental results, the proposed 18TSPC shows better power characteristics compare with the SoA S2CFF design.

However, it also needs to be aware that the proposed 18TSPC has higher pin capacitance than the TGFF. To improve the 18TSPC, a 20TSPC is proposed in this chapter. The proposed 20TSPC has a built-in CK buffer which reduces the CK pin capacitance by 1.5× from the baseline design 18TSPC. Only two more transistors than 18TSPC, the 20TSPC still shows an 8% area reduction compare to the TGFF. Post-layout simulation the proposed 20TSPC shows a 15% leakage power reduction from TGFF. At typical system activity rate, the 20TSPC saves 58% power from TGFF and 18% power from S2CFF. The power overhead compare to 18TSPC (10%) is acceptable. However, it needs to be aware that the 20TSPC has higher D-to-Q delay compare to other SoA SPC designs which mean a trade-off between performance and power efficiency are need to be considered.

The 18TSPC and 20TSPC have a positive hold time value due to the topology. This made the proposed design might be less suitable for further aggressive voltage scaling. To modulating the positive hold time characteristic, a 19TSPC is proposed. The 19TSPC shows a negative hold time at NTV level. 19TSPC maintain the area efficiency of 18TSPC (16% area saving from TGFF). Compare to the TGFF, the 19TSPC shows 21% leakage power reduction, and 63% power reduction at typical system activity rate. The benefit of the shorter D-to-Q delay makes the 19TSPC is 32%, 16% and 34% faster than S2CFF, 18TSPC and 20TSPC respectively. However, at the extreme worst-case at NTV, a non-negligible glitch can be observed on an internal node which can reduce the yield.

To mitigating the non-negligible glitch in 19TSPC, a SPC FF with the 21-transistor topology is proposed (21TSPC) with the improved yield at NTV. The proposed 21TSPC has 21 transistors in total include 4 CK-connected transistors and it saves 4% area compared to TGFF. The post-layout result shows the 21TSPC saves 19% leakage power from TGFF. For system typical activity rate, the power is 55% less compare to TGFF. Owing to the improved topology, the setup time is 2 ns less than the SoA S2CFF.

Consequently, the D-to-Q delay of the 21TSPC is 5 ns less than S2CFF. The proposed 21TSPC maintain the negative hold time characteristic and the 21TSPC has better robustness in hold time compare to TGFF and S2CFF.

All of the above-introduced design meets the requirement of ultra-low power FFs and show the advantage in power efficiency compare to SoA design S2CFF and TGFF. The second and third research objectives of this thesis are met.

Chapter 5

Flip Flop Soft Error Estimation and SEU resilient Design for NTV

From the literature review, it can be seen that the NTV region is the balanced energy-performance trade-off region. However, this technique, although attractive, has not wide spread take up for use in commercial applications because there are some barriers that still require to be research and need to be addressed. The recent research into SRAM operating at $V_{nominal}$ shows that the probability of soft errors, specially the logic upset error, is firmly related to parameter variations. Therefore, the NTV operated circuits are potentially more vulnerable to radiation-induced soft errors.

This chapter aims to answer the last research question of this thesis. Regarding to the reliability concern, the relevant research was carried out. Section 5.1 presents the $Q_{critical}$ and SER estimation for the widely used conventional FFs (TGFF and C2MOS) at different supply voltages, i.e., from $V_{nominal}$ to NTV. The simulation results show that the $Q_{critical}$ level is positively correlated to the V_{dd}. Compared to the nominal voltage operating circuit, the $Q_{critical}$ degradation can be up to 30 × higher. To improve the soft error tolerance capability of the FFs, an error-aware FF design is proposed in Section 5.2. Compared to conventional TGFF and C2MOS, the proposed Error-Aware Flip Flop(EAFF) increased the $Q_{critical}$ by 1.9× at both $V_{nominal}$ and the NTV level. In accordance with this, the SER of the proposed EAFF reduced the SER by 10% at $V_{dd} = 1.2V$ compare to conventional FFs. However, at the NTV, the SER reduction is not as obvious owing to the degradation of the $Q_{critical}$. To improve the EAFF, a new soft error resilient FF, named Dual-Master Latch TGFF (DMTGFF), is proposed in Section 5.3. Instead of tolerating soft error by increasing $Q_{critical}$, the logic error masking technique is used in the proposed DMTGFF. The proposed DMTGFF has the function of both SEU detection and correction. The details of the DMTGFF design are explained in Section 5.3.2. Simulation results in Section 5.3.3 show the proposed design is capable of detecting and correcting the SEU error at NTV. Compared to the SoA soft

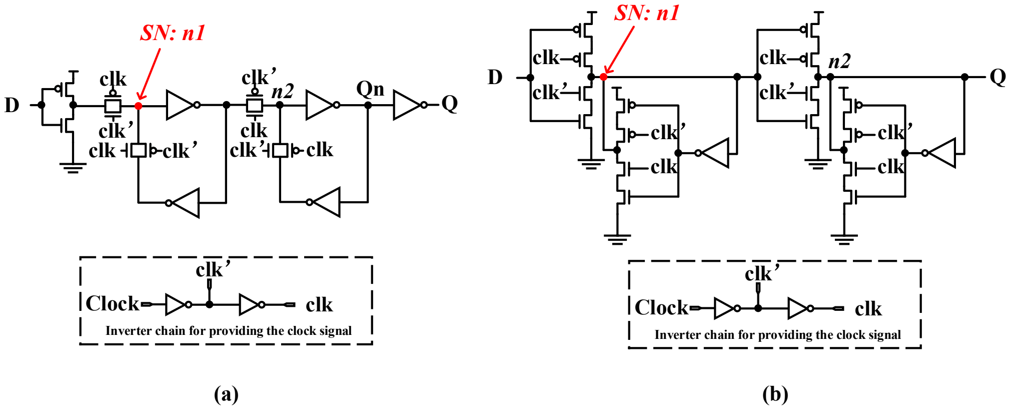


Figure 5.1: The SN are the error injection point. (a) The schematic of TGFF and the sensitive node (SN): n1.(b)The schematic of C2MOS FF and the SN: n1.

error tolerant designs, the DMTGFF consumes 50% less power compare the RazorII and a 25% power saving compared to the DMR (with the error correction function). Section 5.4 concludes with the discussion and summary on the proposed designs.

5.1 Critical Charge and Soft Error Rate Estimation

As introduced in the Chapter 2, the $Q_{critical}$ is a key metric to indicate sensitivity to soft errors. From the literature review on radiation-induced errors, it can be seen that the transient current model can be fit into the double exponential function form. So that, the model (Equation 2.72.8) which is introduced in Chapter 2 Section 2.3 is used for soft error injection in simulation.

In order to inject the transient current to the circuit, a double exponential current source is connected to the Sensitive Node (SN). The SN is defined as the node that is the most sensitive node to soft error, and the position might be vary because of the circuits topology. We chose two typical types of FFs, the Transmission Gate FF (TGFF) and the Clocked CMOS FF (C2MOS) as a case study for measuring the $Q_{critical}$ from the VDD at NTV Region to the VDD = $V_{nominal}$. The schematics of chosen devices and the double exponential current source injection points are shown in Figure 5.1. The SN in devices are set based on previous research which is provided in the paper on $Q_{critical}$ research [129][153].

Take the $Q_{critical}$ estimation of TGFF for instance, shown in Figure 5.2, a transient current source is connected to the SN. The SEU error can be considered in two scenarios, i.e., latched data is flipped from 0 to 1, or the latched data is flipped from 1 to 0. As shown in Figure 5.2 (a) and (b). The similar simulation setup can be also applied to the C2MOS FF.

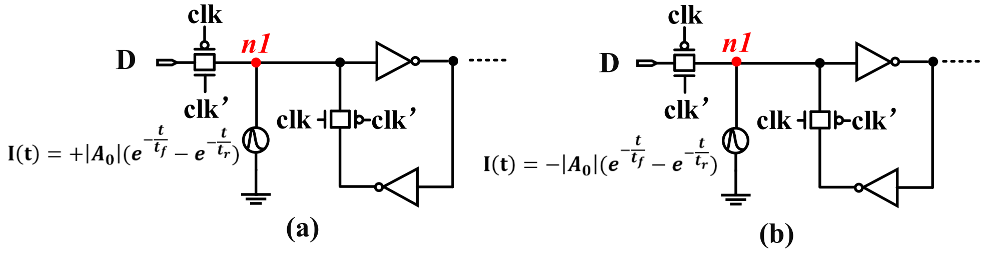

clk
D
n1
clk'
clk
clk'
$\mathbf{I(t)} = +|A_0|(e^{-\frac{t}{t_f}} - e^{-\frac{t}{t_r}})$
(a)
clk
D
n1
clk'
clk
clk'
$\mathbf{I(t)} = -|A_0|(e^{-\frac{t}{t_f}} - e^{-\frac{t}{t_r}})$
(b)

Figure 5.2: SEU error injection to the sensitivity node of the TGFF, the figure shows the master latch of the TGFF. (a) The 0 to 1 SEU error injection. (b) The 1 to 0 SEU error injection.

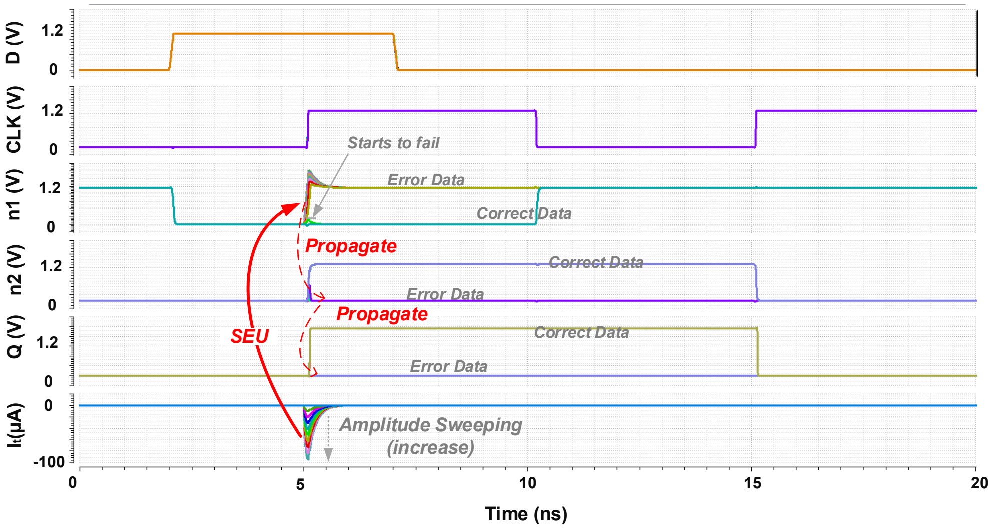


Figure 5.3: Simulation results of the 1 → 0 SEU error injection (TGFF).

The circuits were built by using the Synopsys SAED PDK 90nm technology typical speed NMOS/PMOS SPICE Models with standard Threshold Voltage which is 0.397V. The supply voltage for the test circuits was swept from 1.2V down to 0.4V with steps of 0.1V. The double exponential current source was injected into the Sensitive node in the circuits with positive and negative amplitude values to test the $Q_{critical}$ during particle strikes on PMOS and NMOS. The τ_r is the time constant for fitting the charge deposition process of the MOSFET is set to 33 ps and the τ_f is the time constant for fitting the charge collection process of the MOSFET which is set to 161 ps in accordance with the experimental research paper [154]. The current pulse was generated at the rising edge of the CK signal since the circuits which are being tested are at their most sensitive at this specific time in C2MOS and TGFF [129]. The amplitude of the current pulse A_0 is swept from $0\mu A$ to 100 μA, until the minimum amplitude of the current pulse, which alters the state of FF, was found at a different V_{dd}.

Figure 5.3 shows the simulation results of the SEU soft error injection. The 1 → 0 SEU error injection to TGFF was illustrated here as a case study. The n1 is initialized to 1, n2 is initialized to 0, and the Q is initialized to 0. A 1 at D is supposed to be latched in FF. For the correct operation, at the rising edge of the CK, n1 is stable at 0, n2 is the complement data of n1 which is 1. Q should be rising at CK rising edge and held for 1 clock cycle. Once the SEU error is injected into the SN and the A_0 is increasing, the internal node n1 starts to fail. From the simulation waveform shown in Figure 5.3, it can be seen that the data in n1 is ipped to 1. The error data at n1 then propagated to next stage n2, so that the n2 is still 0. Owing to this, there is no altering at Q which means FF is corrupted by the SEU. The first failure point at n1 with the corresponding A_0 of I_t was recorded, named A_{fe}. For calculating the $Q_{critical}$, the Equation 2.8 can be rewritten in the form of Equation 5.1.

$$Q_{critical} = A_{fe} \cdot (\tau_f - \tau_r) \tag{5.1}$$

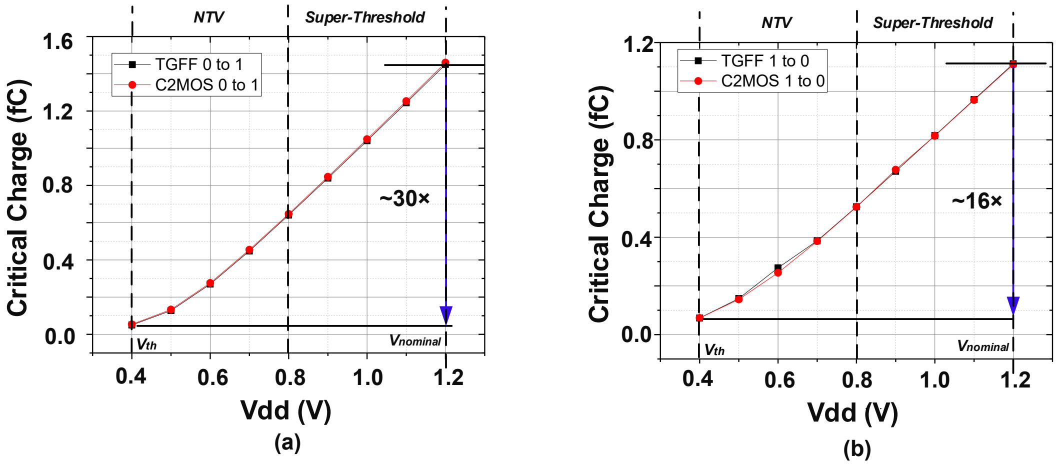


Figure 5.4: The Critical Charge (fC) of NMOS/PMOS particle strikes on TGFF and C2MOS against the V_{dd} (V), (a) Critical charges of 0 to 1 SEU errors (b) Critical charges of 0 to 1 SEU errors

Figure 5.4 shows the HSpice simulation results of the $Q_{critical}$ of TGFF and C2MOS while the particle strikes happened on NMOS and PMOS (i.e. the 0 → 1 or 1 → 0 SEU error). From the results, it can be seen that the $Q_{critical}$ is decreased while the supply voltage was dropped to NTV supply voltage level. These can be observed in both scenarios of SEU error scenarios, i.e. PMOS particle hitting and NMOS particle hitting (flipped from 0 to 1, flipped from 1 to 0). For both types of FFs, the $Q_{critical}$ for the situation 0→1 SEU error at NTV has approximately 30× more degradation when compared to the $V_{nominal}$ ones, For 1→0 SEU error, the degradation of $Q_{critical}$ is about 16 ×. These facts reect that the FFs working in the Near-threshold region are more vulnerable to iron strikes induced transient current pulse. It is can also be concluded from Figure 5.4 that the FFs are more vulnerable to the 1 → 0 SEU since the $Q_{critical}$ is lower in the super-threshold region. When the $V_{dd} < 0.6V$, the $Q_{critical}$ of FFs in both scenarios are at approximately the same level.

While calculating the SER for two types of FFs, the IN_{flux} was set as $0.00565(n/cm^2/s)$, fitting parameter K was set at $2.2 \times 10^{-5}(fC)$, the efficiency parameter of charge collection Q_s was set at 13 [129]. It is assumed 10^7 cells are being tested for 10^9 hours. Therefore, the Equation 2.10 in Chapter 2 Section 2.3 can be written in the form of Equation 5.2.

$$SER = (0.006 \times (2.2 \times 10^{-5}) \times 0.00565 \times e^{\frac{-Q_{critical}}{13}} \times 10^9) \tag{5.2}$$

From the calculation results shown in Figure 5.5, it can be observed that the number of failures increased as the V_{dd} was reduced from $V_{nominal}$ to the NTV. In the scenario of 0→1 SEU (see Figure 5.5(a)), an increase of about 8% in SER increased can be observed when V_{dd} is decreased from 1.2V to 0.4V and an increase of approximately 11% in SER can be observed when V_{dd} is decreased from 1.2V to 0.4V in the scenario where 0→1 SEU (see Figure 5.5(b)).

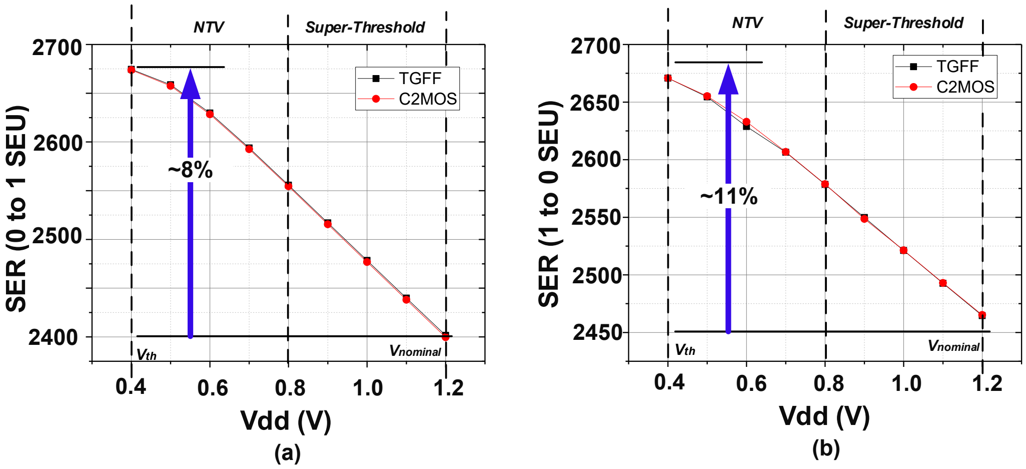


Figure 5.5: SER (FIT) of NMOS/PMOS particle strikes on TGFF and C2MOS against the V_{dd}, (a) 0→1 SEU SER of both FFs, (b) 1→0 SEU SER of both FFs

From the simulation and calculation results, several facts are apparent. First, the $Q_{critical}$ is proportional to V_{dd}. Second, the SER is proportional to $Q_{critical}$. Therefore, one of the most efficient methods to against the soft error at low V_{dd} is increasing the $Q_{critical}$ of the SN in FF.

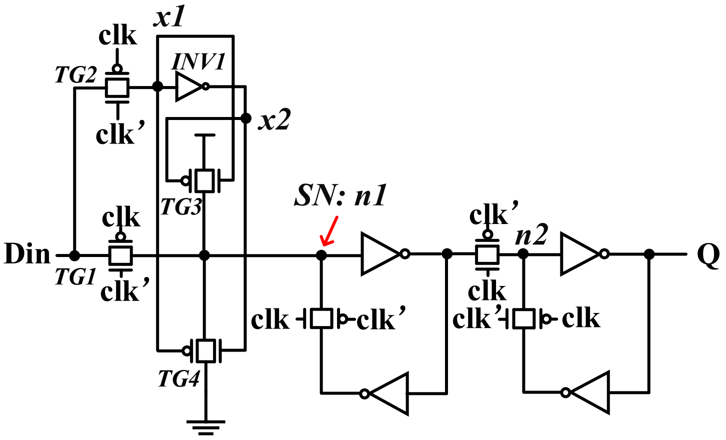


Figure 5.6: Circuit schematic of the proposed Error-Aware FF.

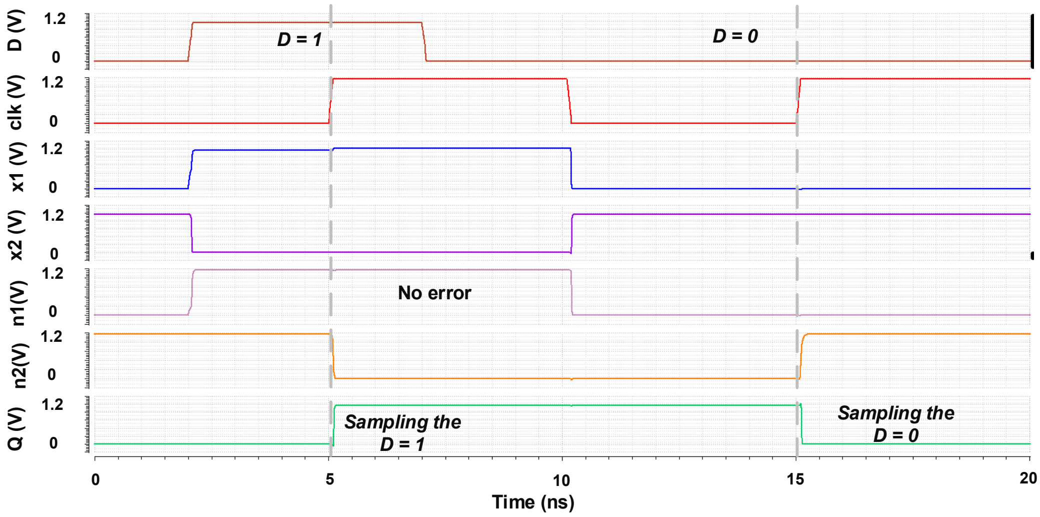


Figure 5.7: Operation waveform of the EAFF without SEU error injection.

5.2 The Proposed Error-Aware Flip Flop Design

Taking the TGFF as a case study. The circuit schematic for the proposed Error-Aware FF (EAFF) is shown in Figure 5.6. This EAFF is proposed based on the principal of increasing the $Q_{critical}$ of the SN in TGFF. The redundancy sample network (TG2, INV1, TG3, TG4) are added to the circuit to increase the soft error tolerant capability.

The proposed FF double samples the input data from port D. The TG2 is the transmission gate which is same as the TG1, these two Transmission gates sample the input data at the rising edge of the clock. The input data which is sampled by the TG1 is propagated to the Master latch and Slave latch via the sensitive node SN. In conventional TGFF, once the SN is affected by the particles strikes induced transient current, a non-negligible glitch can be observed. If the magnitude of the glitch is high enough,

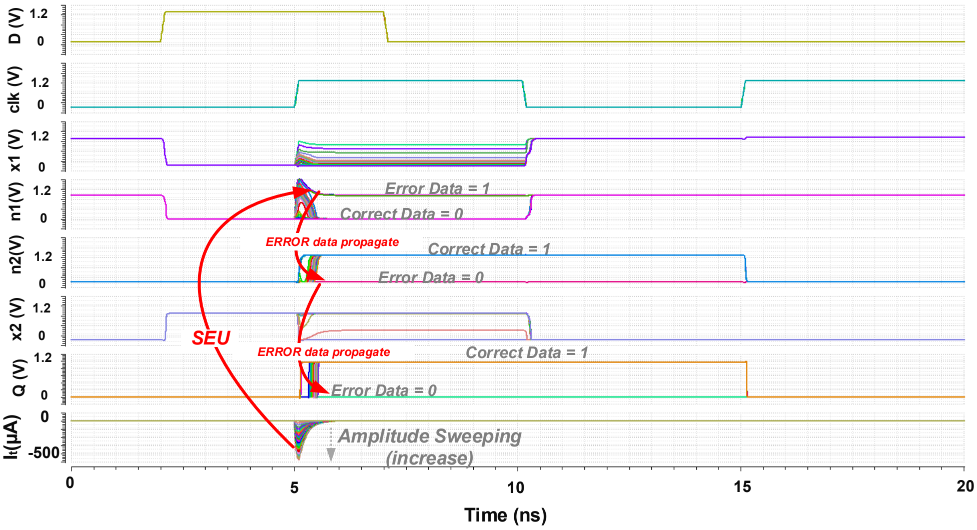


Figure 5.8: Operation waveform of the EAFF with 1 → 0 SEU error injection.

the data in the latch can be flipped. To stabilize the logic values at the Sensitive Node, the redundancy network is employed. The input data which is sampled by TG2 is used to drive the transmission gate TG3 and TG4. The transmission gate TG3 is used to provide a more stable logic 1 while $D = 1$. The transmission gate TG4 is the pull-down network which would provide a more stable logic 0 while $D = logic0$. The operation waveform of the proposed EAFF is shown in Figure 5.7.

5.2.1 Simulation Results of the proposed EAFF

The simulation was performed using an HSpice simulator. The device model, transient current model and simulation set-up are the same as set out in Section 5.1. The supply voltage V_{dd} is swept from V_{dd} which is 1.2V down to the V_{th} of the 90nm CMOS model which is 0.4V. The simulation results of $Q_{critical}$ are shown in Figure 5.9.

The simulation result (Figure 5.9 (a) and (b)) show that the proposed FF EAFF have higher $Q_{critical}$ values compared to the TGFF and C2MOS at different V_{dd}. As shown in Figure 5.9 (a), while the circuit is affected by the particles PMOS hitting (0 →1 error), at nominal voltage, the $Q_{critical}$ of the SN in EAFF is increased by 1.9× from when it was in C2MOS and TGFF. The improvement is also been maintained at NTV (1.9× $Q_{critical}$ improvement compared to conventional FFs at 0.4V). However, However, it should be noted that there is about a 30 × degradation in $Q_{crtical}$ when compared to $V_{dd} = 1.2$V and $V_{dd} = 0.4$V. From the previous results (Figure 5.4), it can be found that the $Q_{critical}$ in conventional FFs are also about 30× degradation. As it is shown in Figure 5.9 (b), while the circuit is affected by the particles hitting the NMOS (0 →1 error), $V_{dd} = 1.2$V, the $Q_{critical}$ of SN in EAFF is increased by 1.4 × when compared to conventional FFs. At NTV, the improvement remains at 1.4 ×. This is similar to

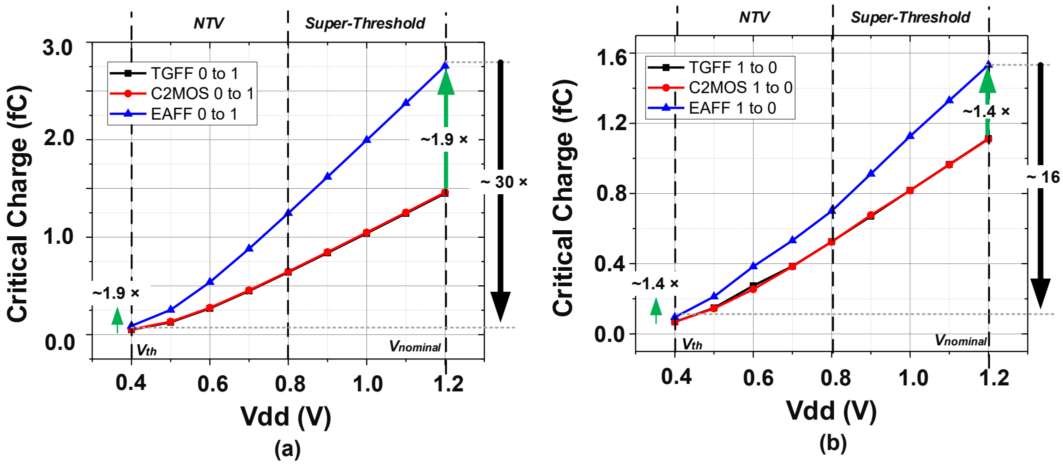


Figure 5.9: The Critical Charge (fC) of NMOS/PMOS particle strikes on the proposed FF EAFF, TGFF and C2MOS against the V_{dd}, (a) The 0→1 SEU critical charge. (b) The 1→0 SEU critical charge.

0→1 scenario, about 16 × degradation in $Q_{critical}$ can be observed when comparing the the values at V_{dd} at 1.2V and V_{dd} at 0.4V. And the level of degradation is similar to the conventional FFs. Therefore, from the results of both error scenarios, it can be seen that the proposed circuit is not able to tolerate the effect of V_{dd}.

In order to calculate the SER and compare the results with the previous tests, the parameter conguration is the same as described in Section 5.1. The calculation results are shown in Figure 5.10.

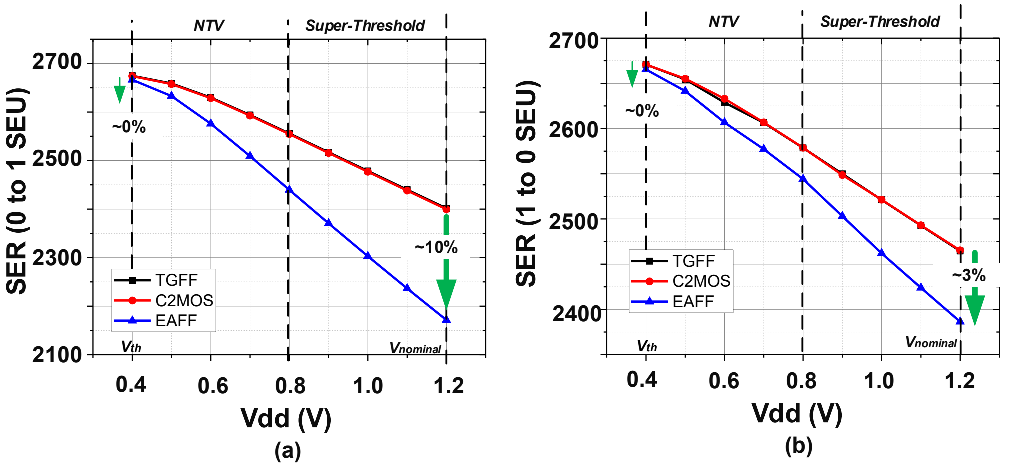


Figure 5.10: SER (FIT) of NMOS/PMOS particle strikes on the proposed EAFF, TGFF and C2MOS vs. V_{dd}. (a) The 0→1 SEU SER of three FFs. (b) The 1→0 SEU SER of three FFs.

When the circuit is under the PMOS particle hitting at nominal voltage supply, the calculation results show that the proposed FF decreases the SER by 10% compared to C2MOS and TGFF. When the circuit is under the PMOS particle hitting at NTV region, the calculation results show that the proposed FF decrease the SER to nearly 0% compared to C2MOS and TGFF. For NMOS particle hitting at nominal voltage, the EAFF decrease the SER by about 3% compared to C2MOS and TGFF. At 0.4V supply voltage, the calculation results show that the proposed FF decreases the SER to nearly 0% compared to C2MOS and TGFF.

It can be seen that the proposed FF can decrease the SER super-threshold region and the nominal voltage point. However, when the supply voltage enters the NTV region, the advantage of the proposed model is not apparent. This is because of the strong impact of the V_{dd} in $Q_{critical}$.

Based on the observations in this section, it can be concluded that error correction techniques are able to protect circuits from soft errors especially when the circuits are at NTV levels. In the next section, a proposed soft error tolerant FF model with error detection and correction is assessed.

5.3 The Proposed Soft Error Resilient Flip Flop Design

From the literature review in Chapter 2 Section 2.3.3, it can be seen that one of the dominant soft error tolerant techniques is TMR [155], it is a gate level technique which adds two extra modules copied from the main device, the output of three modules are passed to the majority voting block. TMR employs 101 transistors. This, however, results in at least a 200% overhead in the area and power. Later, a more efficient technique at the gate level, the Double Modular Redundancy (DMR) architecture, was proposed to detect and correct SEU errors [135]. The design duplicates the main FF and uses a delay path and a latch to detect and correct the SEU error. Although this design saves power and area overhead compared to the previously introduced TMR, it still causes more than 100% overhead in the area and power. An ultra-low-power SEU masking latch which is designed for Sub-threshold voltage operation is proposed [156] by increase the local $Q_{critical}$ in transistor level, while the $Q_{critical}$ is still relatively low, would mean that SEU would still occur while the amplitude radiation induced transient current is high [30]. RazorII is designed with the capability of detecting timing errors and soft errors due to its efficient latch based architecture [138], while it does not have the ability to correct the soft-error of the circuit in transistor level. The SETTOFF circuit is designed based on RazorII to implement transient error detection and correction at transistor level, however, the SETTOFF leads to a 150% area overhead [157].

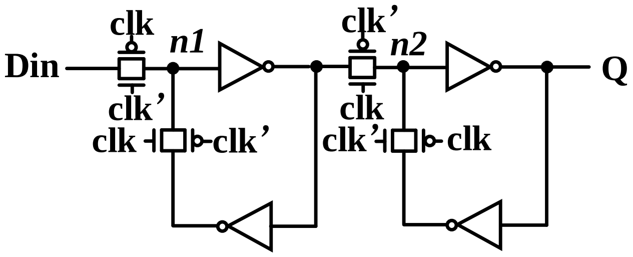


Figure 5.11: Transistor schematic of Transmission Gate FF, soft error is injected to the node $n1$ or $n2$.

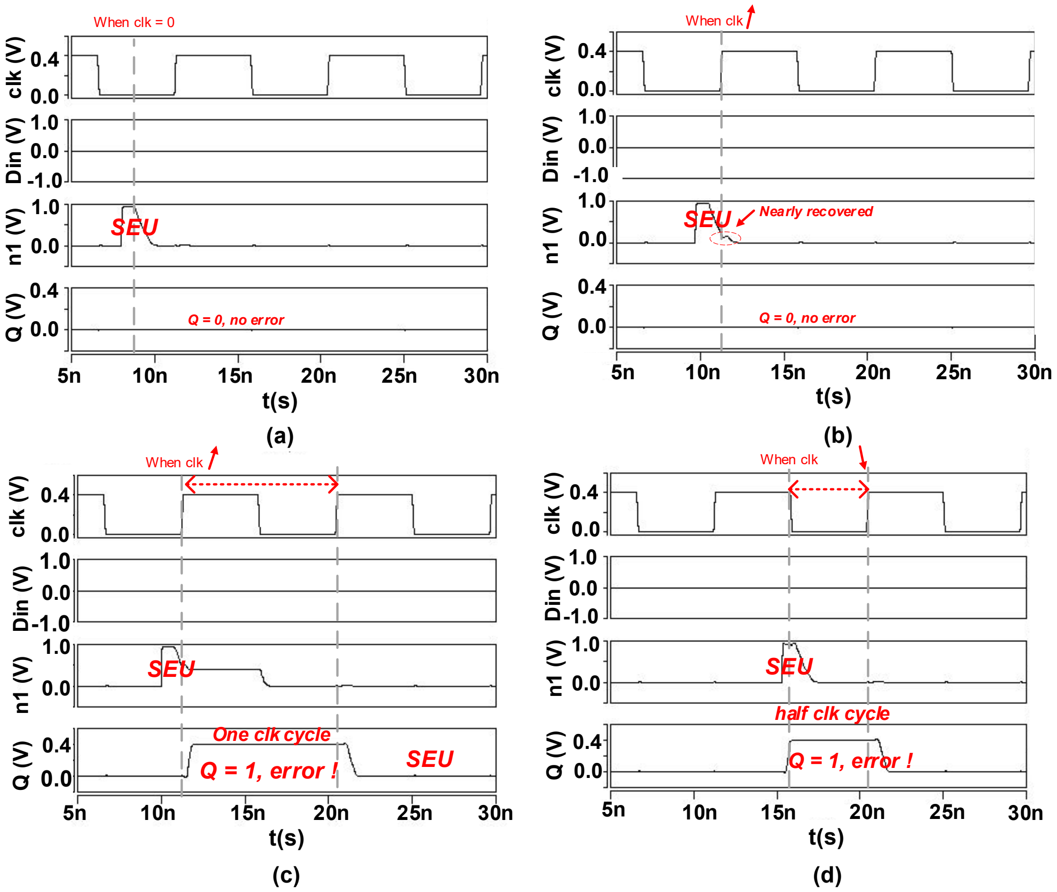


Figure 5.12: 4 scenarios showing SEU events in TGFF.(a)SEU event is injected during the negative clock phase. (b)Injection point just before the rising edge of the clock. (c)Injection point is at the rising edge of the clock. (d)Injection point is at the falling edge of the clock.

5.3.1 Analysis of the $Q_{critical}$ Level of the TGFF

Since the $Q_{critical}$ can be affected by the V_{dd}, the $Q_{critical}$ of the TGFF master latch with that of the slave latch are compared at NTV, $V_{dd} = 0.4V$. The TGFF (see Figure 5.11) was simulated with SYNOPSYS SAED PDK 90nm technology in HSPICE. Soft error injection due to SEU events is modeled by a current source connected to nodes $n1$ or $n2$ (Figure 5.11), to obtain the $Q_{critical}$ in the master and slave latch respectively. Soft errors are injected at various times points to find the worst timing for error tolerance.

The simulation results show that there are four possible scenarios for SEU events, depending on the injection time (see Figure 5.12). Figure 5.12 (a) shows that the SEU event is injected during the negative phase of the clock signal and it recovers before the next rising edge of the clock. The output signal q is not flipped since the slave latch masks the error. Figure 5.12 (b) shows that the SEU event is injected just before the rising edge of the clock signal. Although part of the falling phase of the SEU enters the positive phase of the clock, the output q is not flipped since the voltage of node $n1$ is not strong enough to flip the state while the slave latch is enabled. Figure 5.12 (c) shows that the SEU event is injected at the rising edge of the clock. It can be seen that the output q is flipped for one clock cycle. Finally, Figure 5.12 (d) shows that the SEU event is injected at the falling edge of the clock signal, the output q is ipped for half a clock cycle. Based on these four scenarios, it may be observed that the worst case scenario is when an SEU event is injected at the rising edge of the clock (Scenario 3). Therefore, the effects of soft errors injected at the rising edge of the clock were analyzed as a worst case study and $Q_{critical}$ level comparisons were made.

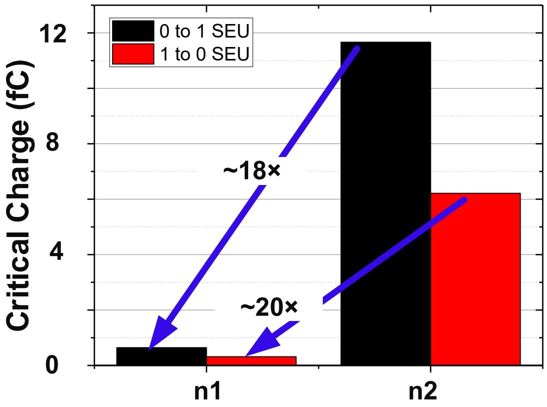


Figure 5.13: Critical Charge of internal node in master (n1) and slave (n2) latch at 0.4V.

For the 0 → 1 type SEU, the simulation results (see Figure 5.13) show that the $Q_{critical}$ at node $n2$ in the TGFF slave latch is $11.667fC$, and at the master latch internal node $n1$ it is $0.644fC$. For the 1 → 0 type SEU, $Q_{critical}$ at node $n2$ is $6.212fC$ and at node $n1$ it is $0.318fC$. Hence, the $Q_{critical}$ of the TGFF slave latch is about 18 - 20× greater than that of the master latch at NTV supply levels. This is confirmation that the master latch of the TGFF is much more likely to be affected by SEU events.

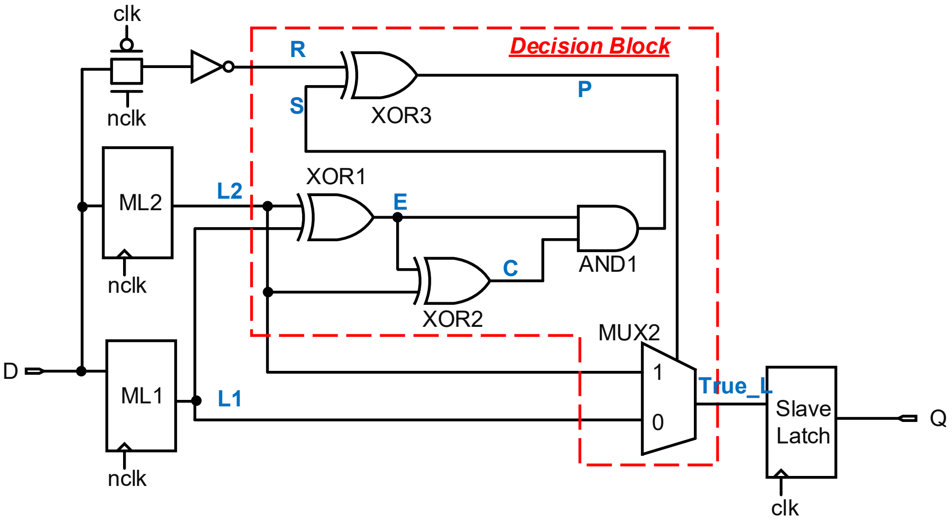


Figure 5.14: Gate Level abstract of the proposed SEU resilient FF deisng, Dual-Master Latch Transmission Gate FF (DMTGFF).

5.3.2 Design of the Dual-Master Latch Transmission Gate Flip-Flop

Instead of replicating the entire FF to provide SEU resilience, the idea of the proposed DMTGFF design, shown in Figure 5.14, is to add a redundant master latch in a Transmission Gate FF (TGFF). Taking advantage of the fact that it is not necessary to replicate the slave latch of a TGFF because of its high $Q_{critical}$ level. Because the node capacitances in a TGFF slave latch are higher than those in the master latch, and due to the strong drive from the master latch at the positive clock edge, an SEU-deposited charge is unlikely to exceed the critical level [129]. In DMTGFF, a 'Decision Block' (Shown in Figure 5.14) is added between the MLs and the SL. It was designed with the function of identifying the ML with SEU and passing the non-corrupted data to SL.

In the situation of no SEUs occurring in the proposed device, the data in both master latches (MLs) are the same and the error monitoring signal E and the data path selection signal P are at logic 0 (see Figure 5.14). In that case, at a positive edge of the CK the slave latch samples the data of the first master-latch (ML1). If there is an SEU in one of the master latches, signal E becomes 1. There are two possible cases when an SEU occurs in a master latch (see Figure 5.15):

- ***Master latch data flipped from 0 to 1***: Once the data is altered in the affected master latch and E changes to 1, gates $XOR2$ and $AND1$ identify the affected master latch. If the SEU occurs in the second master latch (ML2), the error location signal S is 0, otherwise S is 1. At the next positive edge of the clock, the input data is sampled again at the transmission gate acting in parallel to the master latches, and the reference signal R is set to the value of the inverted

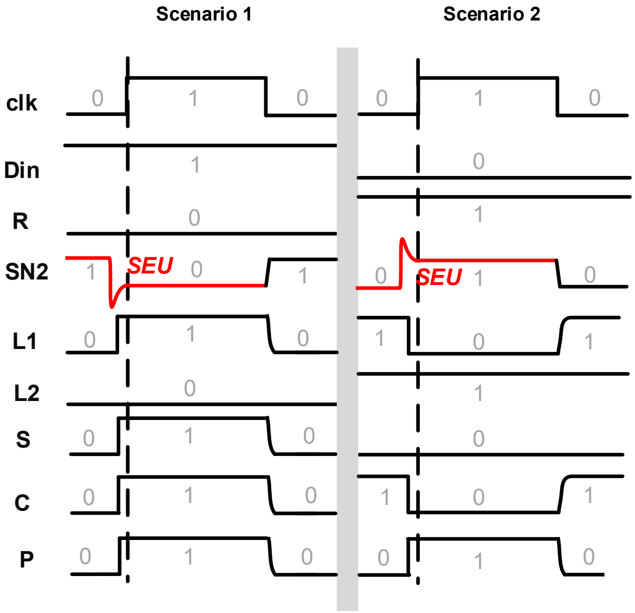


Figure 5.15: The decision block operation diagram for two SEU scenarios in DMTGFF. In both cases, the SEU is assumed to occur at a positive edge of the clock. Case 1: ML1 data is flipped from 0 to 1; Case 2: ML1 data is flipped from 1 to 0.

sampled input, i.e. $R = 0$. In this scenario, Gate $XOR3$ compares the inverted sampled input with the error location signal S to set the data path selection signal P.

- ***Master latch data flipped from 1 to 0***: Here also E changes to 1 once the data is altered in one of the master latches. The affected master latch is identified via the gates $XOR2$ and $AND1$. If the SEU occurs in ML2, the error location signal S is 1; otherwise, S is 0. Here, the reference signal $R = 1$, i.e. it is equal to the inverted sampled input data. The data path selection signal P selects the non-corrupted master latch.

The truth table shown in table. 5.1 provides the logical proof of the 'Decision Block' . However, it needs to be noted that the proposed model is not able to address the Multiple Bit Upsets (i.e. L1 and L2 are both flipped to 1 from 0, or L1 and L2 are both flipped to 0 from 1 simultaneously). From Table 5.1, it can be seen that E remains at 0 when L1 and L2 are equal. So that, the proposed design is not suitable for MBU.

In the *Decision Block*, three XOR gate structures are used for error detection and data path selection. In conventional designs, the XOR2 are implemented with 12 transistors (Figure 5.16 (a)). A transistor reduced XOR gate shown in Figure 5.16 (b) only uses 6-Transistors (6T) [158]. The 6T XOR gate is designed based on a transmission gate

Table 5.1: Truth table of signals in decision block

L1	L2	C	E	S	R	P
0	0	0	0	0	0	0
0	1	0	1	0	0	0
0	0	0	0	0	0	0
1	0	1	1	1	0	1
1	1	1	0	0	1	0
1	0	1	1	1	1	0
1	1	1	0	0	1	0
0	1	0	1	0	1	1

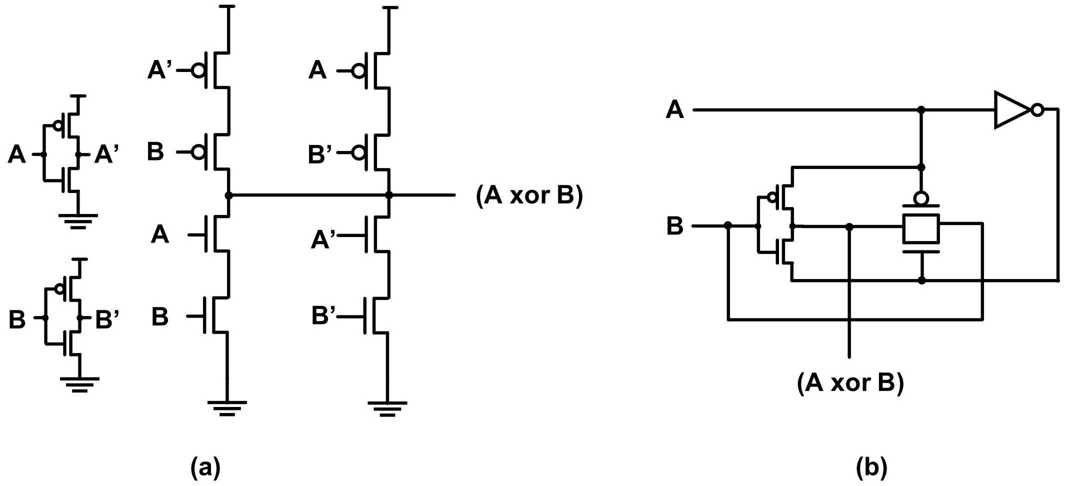


Figure 5.16: XOR gate implementations (a) a 12 transistors conventional XOR gate. (b) a 6 transistors transmission gate based XOR gate [158].

that alleviates the problems of the V_{th} degradation, and it can provide the full swing (i.e., output swing from 0V to V_{dd}).

Figure 5.18 shows the simulation results of the Decision Block at 1.2V. The test bench is developed based on the truth table shown in Table 5.1. From 0 → 10 ns, assuming the correct data in L1 and L2 are 0. 0 → 1 SEU are simulated in both L1 and L2, the data path selection signal 'P' shows the correct output. From 10 → 20 ns, Assuming the correct data in L1 and L2 are 1. 1 → 0 SEU are simulated in both L1 and L2, the response of the data path selection signal P is same to the expecting value. The Decision Block works with the correct functionality. However, a delay on P can be observed, i.e. the propagation delay from $L1$ to P ($T_{L1 \to P}$) and $L2$ to P ($T_{L2 \to P}$). The delay on P is non-negligible since it can lead to error correction failure. The late arrived P signal can pass the error data from the corrupted latch, and a temporary glitch can be observed on $True\_L$.

Figure 5.19 shows the critical path of the Decision Block that is the propagation delay of L1 to P ($T_{L1 \to P}$). The worst case critical path delay can also be identified from the

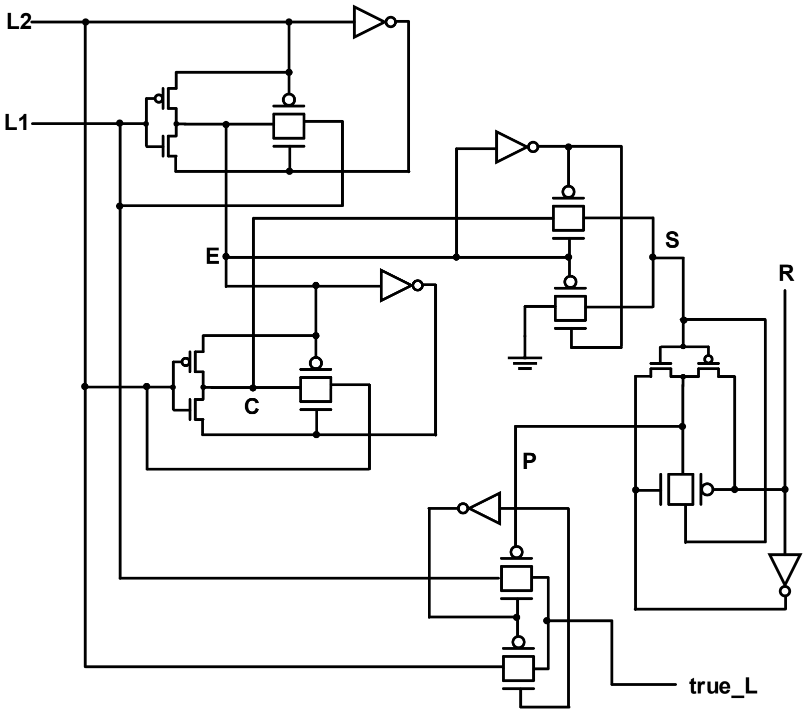


Figure 5.17: Transistor level schematics of the decision block

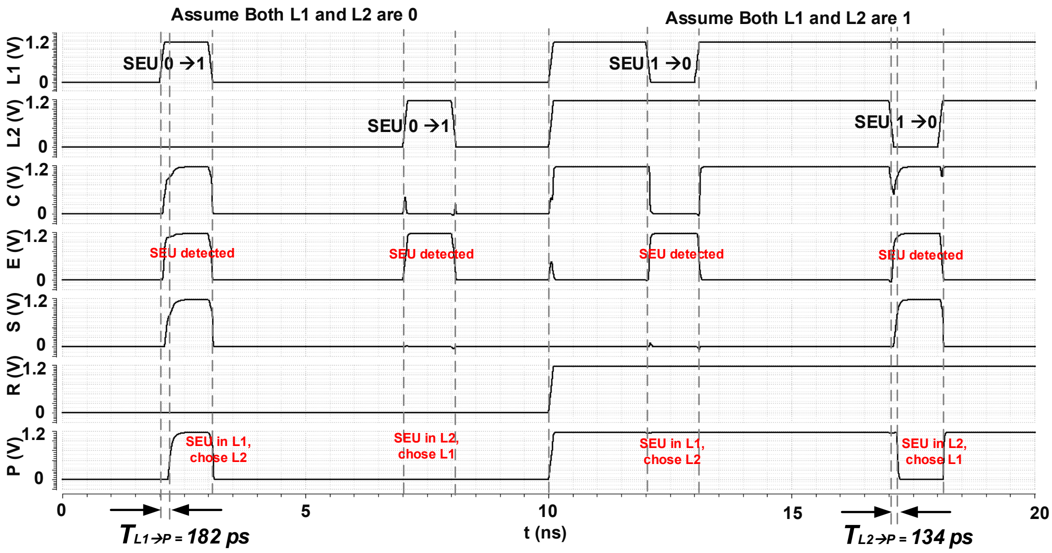


Figure 5.18: Decision Block SPICE simulation results.

SPICE simulation results shown in Figure 5.18. From the analysis, it can be seen that the $T_{L1\to P}$ can be modelled in the form of Equation 5.3.

$$T_{L1\to P} = \Delta t1 + \Delta t2 + \Delta t3 + \Delta t4 \tag{5.3}$$

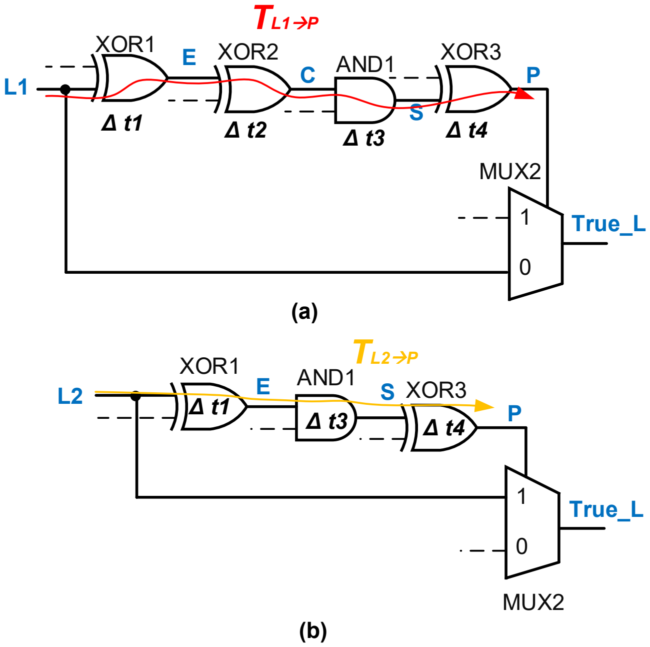


Figure 5.19: Critical path of the Decision Block, the mechanics of delay mismatching.

Therefore, a delay in mismatching appears in the MUX2 gate. There are no gates between the '0' terminal of the MUX2, which means L1 arrive the MUX2 input much earlier than the response of signal P. Note that the propagation delay from L2 to P ($T_{L2\to P}$) is also non-negligible due to the same mechanism which can be found in the path of L1 to P. $T_{L2\to P}$ can be modelled in Equation 5.4.

$$T_{L2\to P} = \Delta t1 + \Delta t3 + \Delta t4 \tag{5.4}$$

To mitigate the Delay induced glitch at $True\_L$, two approaches need to be considered. Firstly, the $T_{L1\to P}$ and $T_{L2\to P}$ need to be minimized. The second approach is adding Delay Elements (DE) between L1 (L2) and MUX2 inputs, a schematic diagram is shown in Figure 5.20. The propagation delay of the DEs ($\Delta De1$ and $\Delta De2$) need to meet following conditions (Equation 5.5 and 5.6).

$$\Delta De1 \geq (T_{L1\to P})_{MAX} \tag{5.5}$$

$$\Delta De2 \geq (T_{L2\to P})_{MAX} \tag{5.6}$$

To determine the $(T_{L1\to P})_{MAX}$ and $(T_{L2\to P})_{MAX}$ at different supply voltages, a Monte Carlo simulation was launched to simulate the decision block at different V_{dd} (from $V_{norminal}$ to V_{th}). At each V_{dd}, 5000 samples are simulated. Delay histograms with

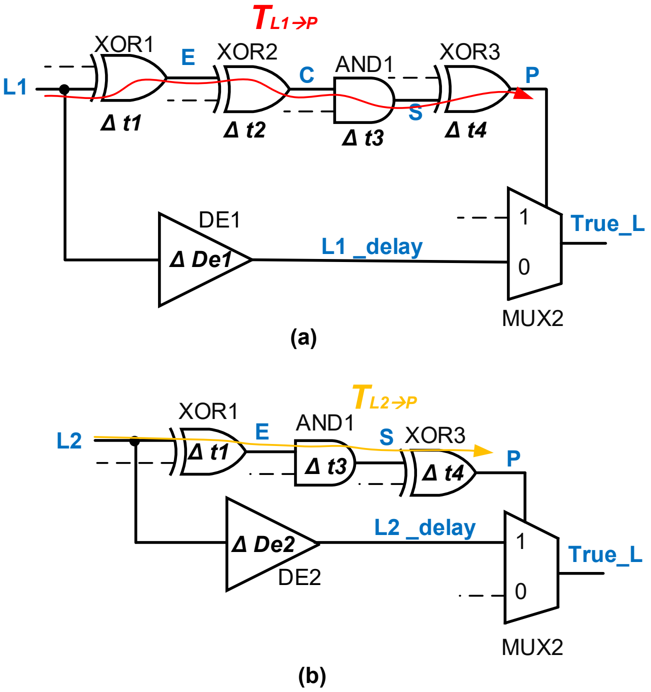


Figure 5.20: Delay Elements are inserted between (a) L1 (b) L2 and MUX2 inputs.

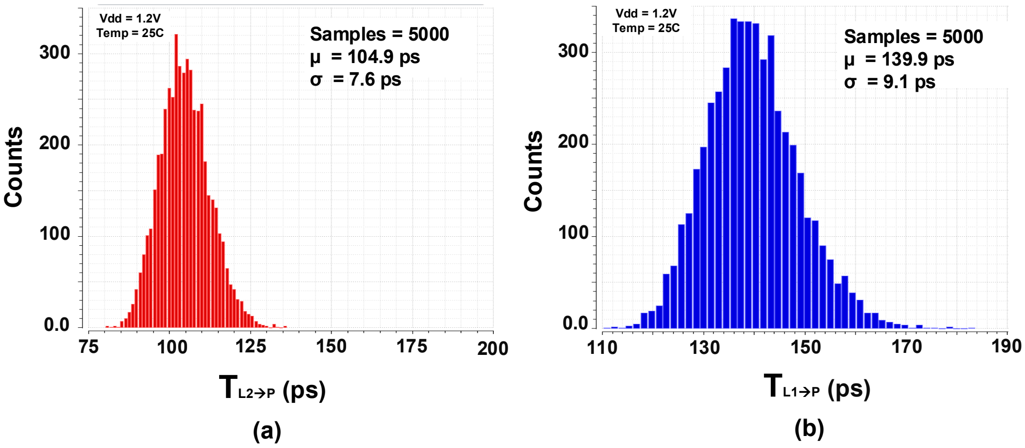


Figure 5.21: (a) $T_{L2\to P}$ and (b) $T_{L1\to P}$ and Monte Carlo simulation at 1.2V.

Gaussian Distribution (e.g. Figure 5.21 shows the delay distribution at 1.2V) were recorded. Also, the mean value (μ) and the standard deviation value (σ) of each histogram are recorded. The worst case (maximum) delay is defined as the Equations 5.7 and 5.8.

$$(T_{L1\to P})_{MAX} = \mu_{(T_{L1\to P})} + 3\sigma_{(T_{L1\to P})} \tag{5.7}$$

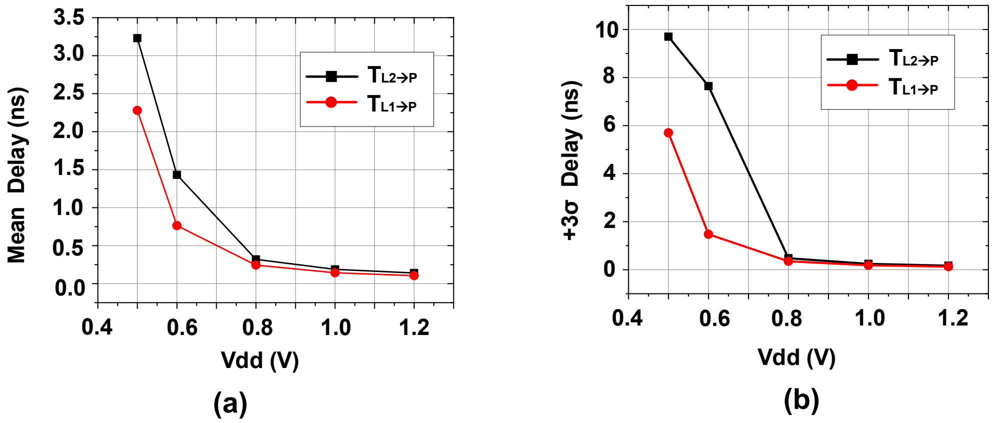


Figure 5.22: The mean (a) and worst case scenario (b) $T_{L1\to P}$ and $T_{L2\to P}$ at different V_{dd} (1.2 - 0.5).

$$(T_{L2\to P})_{MAX} = \mu_{(T_{L2\to P})} + 3\sigma_{(T_{L2\to P})} \tag{5.8}$$

From the simulation results, it can be seen that the Decision Block is not able to pass all the tests at 0.4V due to the effect of the process variation. According to the target (100% pass in Monte Carlo), the minimum operation V_{dd} of the Decision Block is 0.5V. Figure 5.22 shows the worst case $T_{L1\to P}$ and $T_{L2\to P}$ at different V_{dd} (1.2 - 0.5).

To meet the delay balancing condition (Equation 5.5 and 5.6), the delay of DE1 and DE2 need to be larger than the worst-case L1 to P delay. Conventionally, Buffers are designed with series inverters. In this design, inspired by the detection clock signal generator in RazorII [138], the DE1 and DE2 are designed with inverters and transmission gates combination. The schematic diagram of the Delay Element (DE) is shown in Figure 5.23. The DE consists three part: input inverter (Inv_d1), Transmission Gate Chain and the output inverter (Inv_d2). Inv_d1 and Inv_d2 are designed to mitigate the possible voltage degradation in the previous stage of logic output and pull the voltage to rail ($0 - V_{dd}$). The Transmission Gate Chain is equiped with a tunable supply voltage V_{d2}. By scaling the V_{d2}, the delay can be tuned to the target value under different condition, instead of inserting more inverters or selecting a suitable data path from the multiple inverter delay chain.

Figure 5.24 shows the simulation results of the propagation delay of the DE at different V_{dd}, The corresponding value of V_{d2} is shown in both scenarios(Figure 5.24 (a) for balancing the delay in mean value, Figure 5.24 (b) for balancing the delay in the worst-case scenario) . Figure 5.24 (a) shows the DE delay can be turned to specific values (a 5% increase in $T_{L1\to P}$) to balance the mean value (μ) of the $T_{L1\to P}$ and $T_{L2\to P}$ at different V_{dd}. Figure 5.24 (b) shows the DE delay can be turned to specific values (a 5% increase in $T_{L1\to P}$) to balance the Worst-Case ($\mu + 3\sigma$) value of the $T_{L1\to P}$ and $T_{L2\to P}$ at different V_{dd}, which proves the proposed DE is capable to met the delay balancing

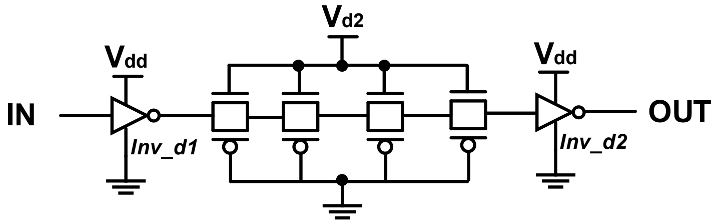


Figure 5.23: Schematic of the Delay Element(DE).

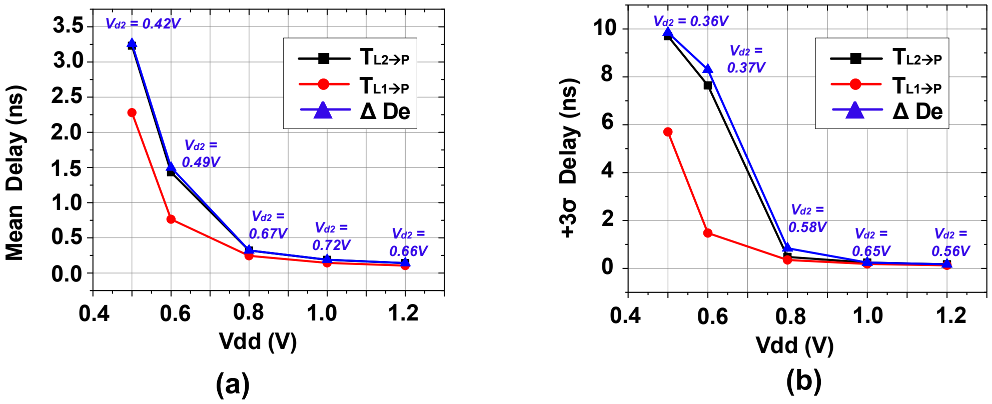


Figure 5.24: The propagation delay of the DE at different V_{dd} and corresponded tuning voltage V_{d2} to balance the $T_{L1\rightarrow P}$ and $T_{L2\rightarrow P}$. (a) Mean(μ) delay balancing. (b) Worst-case ($\mu + 3\sigma$) delay balancing.

conditions (Equation 5.5 and 5.6). The whole schematic of the DMTGFF is shown in Figure 5.25.

5.3.3 Simulation Results and Analysis

The functionality test results of the proposed design are shown in Figure 5.26. The input test vector covers all the scenarios of CK, D and Q combinations. V_{dd} is swept from 1.2V to the minimum operation voltage of the Decision Block (i.e. $V_{dd} = 0.5$V). From the simulation results, it can be considered that proposed design is capable of operating, in the NTV region, where the V_{dd_min} is 0.1V higher than V_{th} (0.4V) of the CMOS model.

To evaluate the error tolerance capability of the proposed DMTGFF at NTV ($V_{dd} =$ 0.5V). A current source (same model as shown in Figure 5.2) was connected to the nodes $SN1$ or $SN2$ (see Figure 5.25) to inject the SEUs with positive and negative amplitude values to generate $0 \rightarrow 1$ and $1 \rightarrow 0$ SEUs. To analyse the amplitude range of the radiation-induced transient currents which the DMTGFF is able to tolerate, the amplitude A_0 of the current source was set as $500\mu A$, what is the upper limit of A_0

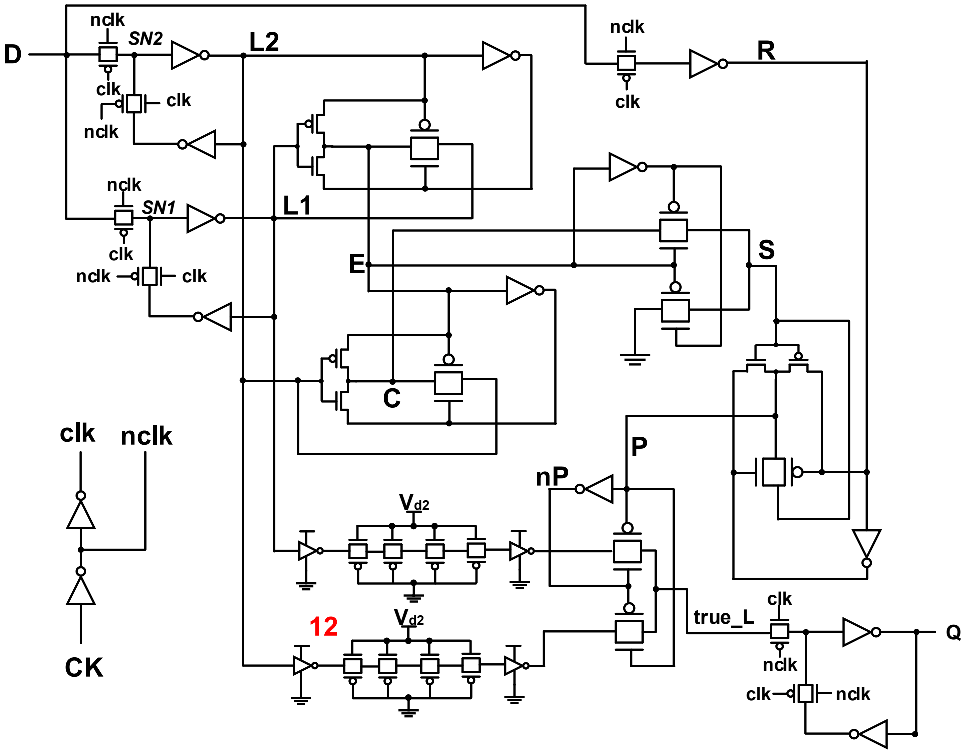


Figure 5.25: A full transistor schematic diagram of the proposed DMTGFF.

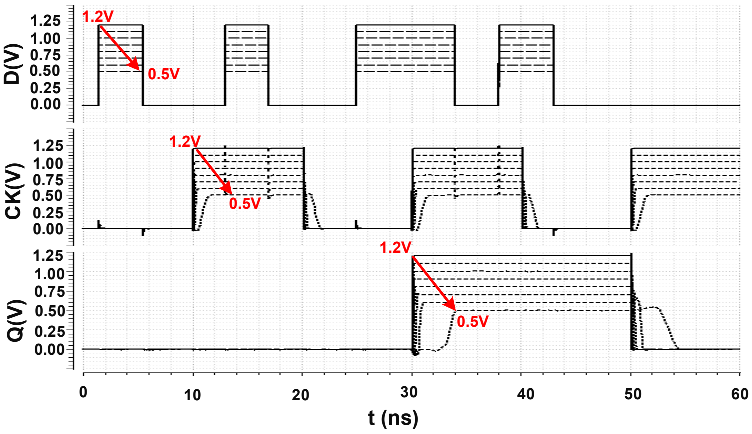


Figure 5.26: Functionality tests results at different V_{dd} (1.2V - 0.5V), without error injection.

corresponds with the maximum reported value of photo-current pulse amplitudes due to heavy ion radiation [154]. The current source (Equation 2.7 and 2.8) time constants are $\tau_r = 33ps$ for the charge deposition process, and $\tau_f = 161ps$ for the charge collection process [154]. Soft errors are injected at the rising edge of the clock which is the most

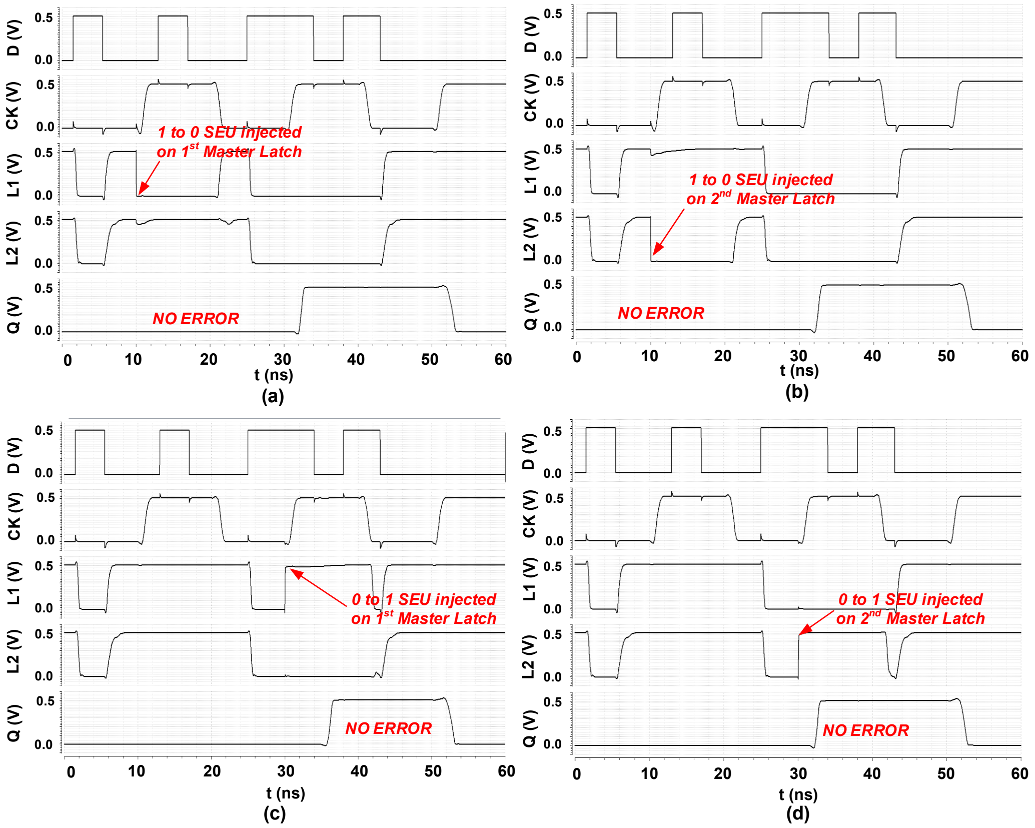


Figure 5.27: (a) DMTGFF output recovered from (a) 1 → 0 SEU error which happens in ML1. (b) 1 → 0 SEU error which happens in ML2. (c) 0 → 1 SEU error which happens in ML1. (d) 0 → 1 SEU error which happens in ML2 at NTV (V_{dd} = 0.5V).

sensitive point of time from the error tolerance point of view as discussed before. The clock frequency is 50MHz. Simulation results are shown in Figure 5.27. The results show the 0 → 1 and 1 → 0 SEU events injected into ML1 or ML2. In all the cases the Decision Block (Figure 5.25) selects the error-free ML to propagate the uncorrupted data into the slave latch. It has also been verified that the proposed DMTGFF is capable of tolerating SEUs across the entire range of the transient current amplitudes from $10\mu A$ to $500\mu A$.

The power consumption and speed of the DMTGFF, DMR and TMR have been analyzed at different supply voltages with a $50MHz$ clock frequency. Figure 5.28 (a) shows the CK-Q delay of the proposed DMTGFF, DMR, and TMR. The proposed DMTGFF has a similar Clock-to-Q delay to that of TGFF at different supply voltages. At the target operation V_{dd} (0.5V), the proposed DMTGFF is 34% faster than the DMR (with error correction). Figure 5.28 (b) shows that the DMTGFF consumes less power than the DMR and TMR with the whole range of the flop activity rate (defined as the write-in activity rate over ten clock cycles) at 0.5V. For the average activity rate of the flip-flop

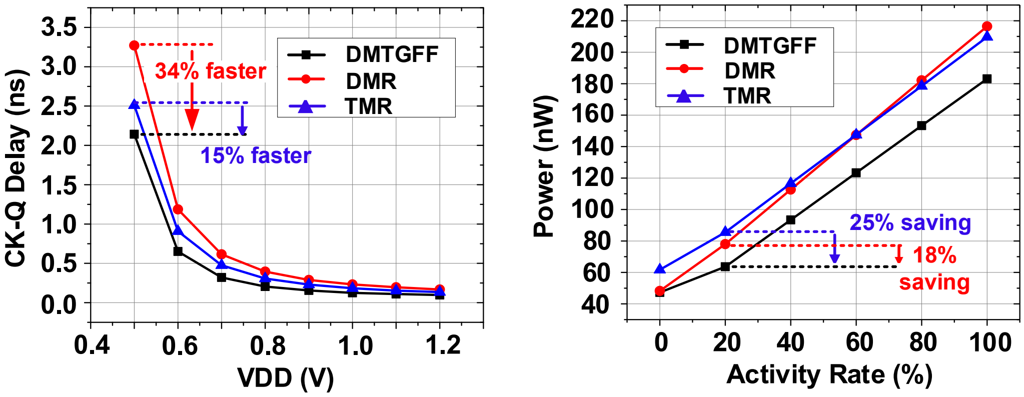


Figure 5.28: (a) CK-Q delay of DMTGFF, DMR(with error correction) and TMR at different V_{dd} (0.5V - 1.2V) (b)Power consumption of the proposed DMTGFF, DMR (with error correction) and TMR against the activity rate at V_{dd} = 0.5V

Table 5.2: Comparison of Error-tolerant Device Performance

	DMTGFF	**DMR⧫**	**TMR**	**Razor**	**RazorII**
Transistor Count	86	80	96	76	47
SEU Detection	Yes	Yes	Yes	No	Yes
SEU Correction	Yes	Yes	Yes	No	No
Power (nW)*	63.6	78	85.5	75.4	127.6
CK-Q Delay (ns)**	2.1	3.3	2.5	2.8	1.3

DMR⧫ : DMR with error correction function.

* V_{dd} = 0.5 V, F_{CK} = 50 MHz, Temp = 25 ^oC, Activity Rate = 20%

** V_{dd} = 0.5 V, Temp = 25 ^oC,

(20%) DMTGFF consumes 25% less power compares to the TMR and 18% less power compare to TMR.

It can be seen that the TMR uses the highest transistor count, which is 96. It requires 86 transistors to implement the proposed DMTGFF design, which is higher than the DMR (with error correction), Razor and RazorII. This is due to the complex topology of the built-in block, i.e. the decision block, and the delay elements in the decision block of the DMTGFF. Note that the Razor is neither able to detect the SEU error nor able to correct the SEU. RazorII is capable of SEU detection. However, the SEU correction is not able to be done in the cell. The DMTGFF has the lowest power consumption at NTV. Compared to the RazorII, which uses minimum transistor count, the DMTGFF uses 50.2% less power at NTV (V_{dd} = 0.5 V, F_{CK} = 50 MHz, Temp = 25 ^oC, Activity Rate = 20%). Also, the proposed DMTGFF has the advantage of the CK-Q delay at NTV. The RazorII has the lowest delay. However, the RazorII is a latch instead of a flip-flop. The lower CK-Q delay benefits from shorter logic stages in RazorII when compared to Flip-flop designs.

5.4 Concluding Remarks

From the literature review in Chapter 2 Section 2.3, it can be observed that the circuit works in Ultra-low voltage region would be more vulnerable to radiation induced soft errors with a decreased voltage supply [30], the particle radiation induced errors on MOSFET would cause a transient current to be generated at the SN. The SN has the lowest $Q_{critical}$ among all internal nodes of the circuit. This can flip the data in memory devices randomly and further corrupt the system, this phenomenon is the SEU. And this phenomena is reported as the main cause of failure [122] among the soft errors.

To quantify the impact of the supply voltage on the $Q_{critical}$ and SER of the sequential logics, two widely used FF designs, i.e. TGFF and C2MOS, are chosen for the case study. The $Q_{critical}$ and SER of the conventional FFs are measured from the $V_{nominal}$ to the NTV. The simulation results show that the $Q_{critical}$ of both FFs has a positive correlation to the V_{dd}. Comparing to the $V_{nominal}$ operation, approximately 30× degradation in $Q_{critical}$ has been observed at NTV level. Furthermore, the SER at NTV is 11% higher than the $V_{nominal}$ operation. The initial design, EAFF, was developed to increase the $Q_{critical}$ of the TGFF. In the proposed EAFF, eight transistors were added to the TGFF to increase the $Q_{critical}$ thereby decreasing the SER. Although the proposed design increased the $Q_{critical}$ at different voltages by 1.9×, the degradation of the $Q_{critical}$ (approximately 30×) at NTV showed no improvemnt when compared to the $Q_{critical}$ at $V_{dd} = 1.2$V. The advantage of the proposed model is counteracted by the substantial impact of $Q_{critical}$ degradation at NTV. At nominal voltage ($V_{dd} = 1.2$V), the proposed EAFF shows a 10% reduction in SER when compared to conventional FFs. However, the decrease in SER is less efficient when V_{dd} is scaled down. Based on the observed results, increasing $Q_{critical}$ might not be the best solution for ultra-low voltage operating system. For eliminating the SEU error in FFs at the NTV, the FFs with both the functions of error detection and correction is needed.

Therefore, a new SEU-resilient Double Master-latch Transmission Gate FF (DMTGFF) for NTV operation is proposed. The DMTGFF is capable of self-detection and self-correction of circuit-level errors due to SEU events. Unlike the existing DMR and TMR techniques, DMTGFF only duplicates the master latch rather than the whole device. From the simulation results, it can be seen the DMTGFF is capable of operating in error- free at NTV with the entire range of the transient current pulse amplitudes. The proposed DMTGFF is compared with the SoA error-tolerant FF designs regarding transistors count, power, SEU tolerant capability, and performance. The result shows that the DMTGFF is 34% faster than the DMR (for error correction) and 15% faster than the TMR. The DMTGFF uses more transistors compared to the SoA designs. However, the DMTGFF shows the best power characteristics among all the SoAs. For a typical FFs activity rate (20%), the DMTGFF require 25% less power than the TMR and has a 18% power saving compared to the DMR (for error correction).

Chapter 6

Conclusion and Future Work

6.1 Summary of the work

With the development of WSN, energy harvesting, innovations in better and related technologies, the concept of IoT, has become a popular research topic among academics and researchers in industry. IoT devices requires designs that are small, low cost and power/energy efficient. This is not only a requirement for slow duty cycled devices that require power/energy efficiency but also a requirement for high-performance computing devices. The excessive power density results in a rapid increase in temperature, which limits the yield and reliability of the system. This is known as the ”power wall”. Driven by the demand for power/energy efficiency and further developments in semiconductor technology, research into power reduction and ULP techniques is essential. NTV techniques, a potential solution for implementing ULP IoT devices, have demonstrated significant reductions in energy consumption by decreasing the supply voltage to approach the V_{th}, while preserving favorable variability and performance characteristics compared to STV operations. This research focuses on addressing challenges in digital circuit design. More specifically, the research is focused on the methodology applied to address design challenges in e sequential logic design. The background, SoA technology and challenges in the NTV design are reviewed and discussed in Chapter 2.

Based on the observations in Chapter 2, the work carried out under this project started with surveying SoA SPC FFs and TGFF to evaluate suitability for the NTV operations and highlighted critical issues with incumbent designs from circuit level to system level in Chapter 3. The existing design issues in SoA SPC FFs lead to degraded benefits which must be considered. More importantly, with a in-depth analysis of the TCFF design issue. A more detailed NTV operated SPC FF design requirement was derived, which are expected to lead to the better robust and deployable NTV design: (1) Zero contention path, (2) zero dynamic nodes, (3) Minimize stacked structures on data path (4) Minimized multiple master-slave interface paths and (5) minimized clock pin capacitance as well as internal clock nodes.

By obtaining the above design requirement, a modified TCFF circuit, named TCFF-NTV, is proposed for enabling the original TCFF operating at NTV. The proposed TCFF-NTV maintains the benefit of power efficiency and timing characteristics. However, the TCFF-NTV still not meet all the requirement of ULP SPC FFs owing to the area overhead and high CK pin capacitance. This motivate us to carry out the research to design, developing and evaluating the novel ULP FFs rather than optimizing the existing SoA ULP FFs.

In Chapter 4, 18TSPC, a SPC FF with only 18 transistors (the lowest reported for a fully-static contention-free SPC FF) with a novel topology is proposed. It is the first design for this type which achieve smaller area compared to the TGFF cell. Unlike the other SoA designs, 18TSPC meets all ULP FF design requirements. The design has been implemented in 65nm CMOS along with the TGFF in sub system designs and later being fabricated in silicon. The silicon experimental results show that the proposed 18TSPC achieves up to 75% power consumption compared to the conventional TGFF design at both the $V_{nominal}$ and the NTV. Although such scheme shows the advantage in power efficiency, it worth to concern that the CK pin capacitance of the 18TSPC is higher than TGFF at the cell level. It also need to be aware that the 18TSPC has the positive hold time value due to the topology. Therefore, three extended ULP SPC FF designs (19TSPC, 20TSPC, and 21TSPC) based on 18TSPC are proposed in Chapter 4 section 4.5 for ameliorating the hold time and CK pin capacitance property of the baseline design, 18TSPC. All of these designs achieves the area saving (compared to the TGFF) and the power efficiency as the 18TSPC. These designs provide the designers a wide choice for targeting different design scenarios with different Power/Performance/Area trade-offs.

As it previously introduced, reliability issue also need to be considered at the NTV operating system. In Chapter 5, the sensitive level ($Q_{critical}$) estimation have been done to TGFF and C2MOS from $V_{normial}$ to the NTV. Results show that the $Q_{critical}$ of two FF circuit can be reduced by 30× when V_{dd} is scaled down to the NTV compared to the $V_{norminal}$ operation. Accordingly, SER of two FFs are increased by 11% comparing to the nominal operated circuits. An Error-Aware flip-flops (EAFF) circuit model was proposed to against the soft error by increasing the $Q_{critical}$. However, results show the reduction in SER is less efficient at the NTV owing to the strong impact of voltage to the $Q_{critical}$. It also can be inferred that against the SEU by just increasing $Q_{critical}$ is not the optimal solution for the NTV systems. Base on this observation, a new SEU-resilient DMTGFF for NTV operation is proposed. The DMTGFF circuit is capable of self-detection and self-correction of the SEU error. Results proves that the proposed DMTGFF is capable of operating in error-free at NTV) through the entire range of the transient current pulse amplitude. Compared to the SoA error tolerant FF designs, including the Razor II, the DMTGFF shows a better power efficiency than all existing SoA designs.

The conclusions drawn from this thesis are supported by an in-depth analysis using SoA industrial standard EDA tools and technology libraries. The simulation of post-layout transistor level designs using high accuracy simulators including the HSpice and PrimePower etc., as well as experimental validation provided by fabricated test chip designs were used to meet the objectives of this research. Some of the designs proposed in this thesis has been granted patents and have been considered by industry for further product level evaluation. It is hoped that the techniques proposed in this thesis will make further contributions to the development of future ultra-low power IoT devices.

6.2 Future Work Direction

Based on the research presented in this thesis, some possible future directions are outlined in this section.

6.2.1 Live-Slave Flip Flop for Further Power Reduction

As previously introduced, the proposed designs in Chapter 4 are raw circuits without any additional power reducing techniques. So that, further power reduction techniques (e.g., FBB/RBB, Multi-V_{th}, Power Gating, etc.) can be added to these raw low power circuits to enable further power efficient operation.

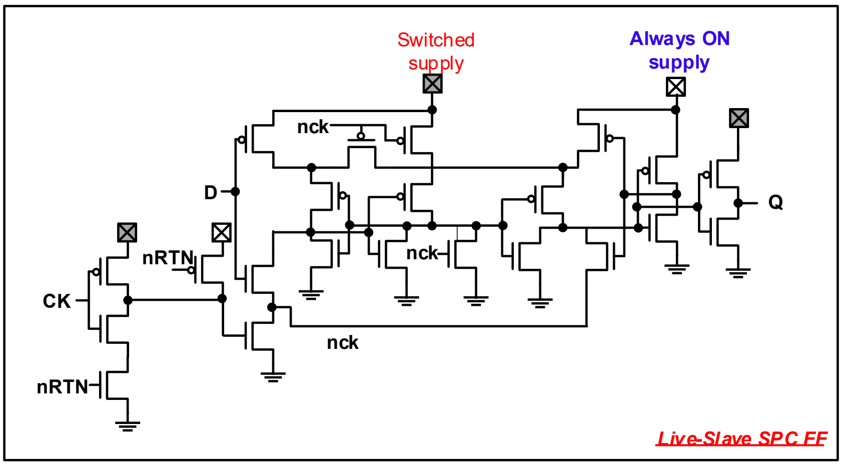


Figure 6.1: Initial design of Live-Slave Retention FF based on 20TSPC.

To further reduce the power consumption of the proposed FF design, a slave-retention functionality can be added to the circuit. Figure 6.1 shows the schematic diagram of a possible live-slave retention FF design, named Live-Slave 22TSPC (22TSPC-LS).

The 22TSPC-LS is developed based on the 20TSPC. The existing internal clock buffer (M01, M02) provide the designer with the opportunity to implement the local clock gating by adding the minimum number of transistors (2) which are M21 and M22. Combined with M01 and M02, the functionality of NAND is provided. The retention nRTN signal controls the gate. Three voltage sources are used in 22TSPC-LS, including the Switched Supply (blank triangle), Always-ON Supply (shade triangle) and ground. The Switched Supply comprises of a switchable voltage source. The Always-ON Supply is with the fixed voltage value.

When nRTN = 1, nck is the complement signal of CK. Switched Supply is on, with the same voltage value as the Always-ON. The 22TSPC-LS performs as the same function as the 20TSPC with the characteristic of single-phase, full-static and contention-free. When nRTN = 0, nck is held to 1. Which means the nck signal clocked FF is clock-gated. After this, the Switched Supply can be turned off. Leading to the transistor M01, the whole master latch, and the output inverter being powered off. The Always-ON Supply is fixed so that the only powered block is the slave latch.

With this functionality, both the dynamic and leakage power can be further reduced during the data retention phase. Because of the constrained research time, the proposed 22TSPC-LS design has not been layed out and thoroughly evaluated. However, it is worth carrying on with a full evaluation and a more in-depth analysis of the 22TSPC-LS, since it has strong potential to be used to construct the register file of IoT devices for power reduction.

6.2.2 Reliability Enhancement for SPC FFs

The proposed SPC FFs in Chapter 4 provides the designer with the options to achieve lower power with less cost (area). For further investigation into the potential application scenarios, a deeper reliability evaluation needs to be done to the proposed SPC FFs, including the capability of resisting noise and soft errors.

Take the 18TSPC as a case study, the critical charge needs to be measured, with the setup what was introduced in Chapter 5, at the sensitive node f1 and f2 (see Figure 4.6). Based on these findings, the soft error hardened SPC FF design based on 18TSPC can be considered as a possible future direction for further research.

Appendix A

Appendix: Shift Register Verilog Model

The verilog model of the Dual-Shift register is present as bellow:

```
module SOFTSHIFTSLICE #(parameter depth = 20)
(
output          Y,
input           DIN,
input           DSERIAL,
input   [1:0]   MSEL,
input           CK
);

reg [depth-1:0] chain;

reg next_msb;

assign Y = chain[0];

always @(*)
        case(MSEL)
                2’d0 : next_msb <= DIN;
                2’d1 : next_msb <= chain[0];
                2’d2 : next_msb <= !chain[0];
                2’d3 : next_msb <= DSERIAL;
        endcase

always @(posedge CK) begin
        chain <= {next_msb, chain[depth-1:1]};
end

endmodule

module SINGLESHIFTREG
(
output  [15:0]  Y,
input   [15:0]  DIN,
input   [1:0]   MSEL,
input           CK
```

```
);

wire [15:0] yint;
assign Y = yint;

SOFTSHIFTSLICE u0  (.Y(yint[0] ), .DIN(DIN[0] ),
        .DSERIAL(yint[15]), .MSEL(MSEL), .CK(CK));
SOFTSHIFTSLICE u1  (.Y(yint[1] ), .DIN(DIN[1] ),
        .DSERIAL(yint[0] ), .MSEL(MSEL), .CK(CK));
SOFTSHIFTSLICE u2  (.Y(yint[2] ), .DIN(DIN[2] ),
        .DSERIAL(yint[1] ), .MSEL(MSEL), .CK(CK));
SOFTSHIFTSLICE u3  (.Y(yint[3] ), .DIN(DIN[3] ),
        .DSERIAL(yint[2] ), .MSEL(MSEL), .CK(CK));
SOFTSHIFTSLICE u4  (.Y(yint[4] ), .DIN(DIN[4] ),
        .DSERIAL(yint[3] ), .MSEL(MSEL), .CK(CK));
SOFTSHIFTSLICE u5  (.Y(yint[5] ), .DIN(DIN[5] ),
        .DSERIAL(yint[4] ), .MSEL(MSEL), .CK(CK));
SOFTSHIFTSLICE u6  (.Y(yint[6] ), .DIN(DIN[6] ),
        .DSERIAL(yint[5] ), .MSEL(MSEL), .CK(CK));
SOFTSHIFTSLICE u7  (.Y(yint[7] ), .DIN(DIN[7] ),
        .DSERIAL(yint[6] ), .MSEL(MSEL), .CK(CK));
SOFTSHIFTSLICE u8  (.Y(yint[8] ), .DIN(DIN[8] ),
        .DSERIAL(yint[7] ), .MSEL(MSEL), .CK(CK));
SOFTSHIFTSLICE u9  (.Y(yint[9] ), .DIN(DIN[9] ),
        .DSERIAL(yint[8] ), .MSEL(MSEL), .CK(CK));
SOFTSHIFTSLICE u10 (.Y(yint[10]), .DIN(DIN[10]),
        .DSERIAL(yint[9] ), .MSEL(MSEL), .CK(CK));
SOFTSHIFTSLICE u11 (.Y(yint[11]), .DIN(DIN[11]),
        .DSERIAL(yint[10]), .MSEL(MSEL), .CK(CK));
SOFTSHIFTSLICE u12 (.Y(yint[12]), .DIN(DIN[12]),
        .DSERIAL(yint[11]), .MSEL(MSEL), .CK(CK));
SOFTSHIFTSLICE u13 (.Y(yint[13]), .DIN(DIN[13]),
        .DSERIAL(yint[12]), .MSEL(MSEL), .CK(CK));
SOFTSHIFTSLICE u14 (.Y(yint[14]), .DIN(DIN[14]),
        .DSERIAL(yint[13]), .MSEL(MSEL), .CK(CK));
SOFTSHIFTSLICE u15 (.Y(yint[15]), .DIN(DIN[15]),
        .DSERIAL(yint[14]), .MSEL(MSEL), .CK(CK));

endmodule

module DUALSHIFTREG
(
output  [31:0]  Y,
input   [15:0]  DIN,
input   [1:0]   MSEL,
input           CK0, CK1
);

SINGLESHIFTREG uSHIFT18T(
.Y (Y[15:0]),
.DIN (DIN),
.MSEL (MSEL),
.CK (CK0)
);
SINGLESHIFTREG uSHIFTTG(
.Y (Y[31:16]),
.DIN (DIN),
.MSEL (MSEL),
.CK (CK1)
```

```
);

endmodule
```

Appendix B

Appendix: Test Board Information

B.1 Address assignment

The address assignment of the test board is presented as follow:

0xF0C0_0004	AHBLAYER1_CTRL			32'hFFFF0000	
		1	AES-LBIST-CAI enable		
		2	confidential block		
		3	confidential block		
		4	confidential block		
		5	confidential block		
		6	confidential block		
		7	confidential block		
		[11:8]	SCANIN pattern: common to all 4 macros		
		[31:18]	LBIST pattern: expanded for 128b key/data		
0xF0C0_0024	AHBLAYER1_CTRL_SET				
0xF0C0_0034	AHBLAYER1_CTRL_CLR				
0xF0D0_0064	AHBLAYER2_DIAG21		CAI_LBIST_OUT	RO	
		0	**confidential block**		
		1	**confidential block**		
		2	**confidential block**		
		3	**confidential block**		
		4	**CAI LBIST running now**		
		5	**CAI LBIST completed (at least once)**		
		6	**CAI LBIST pass (last loop only)**		
		7	**AHBLAYER1_CTRL**		
		8	confidential block		
		9	confidential block		
		10	confidential block		
		11	confidential block		
		12	confidential block		
		13	confidential block		
		14	confidential block		
		15	confidential block		
		[19:16]	confidential block		
		[23:20]	CAI SCANOUT		
		[27:24]	confidential block		
		[31:28]	confidential block		
0xF0D0_1064	AHBLAYER2_DIAG21_SET				
0xF0D0_2064	AHBLAYER2_DIAG21_CLR				

Figure B.1: Test chip address assignment. (continue...)

0xF0D0_0004	AHBLAYER1_MISC		SHIFTREG_CTRL	32'h0	
		0	18TFF clock enable		
		1	TGFF clock enable		
		2	Clock select:0=IVR-clock;1=register(bit3)		
		3	Clock source for register-mode clocking		
		[5:4]	input mux: 0=external;1=parallel;2=parallel-inverted;3=serial		
		6	output mux: 0=18TFF; 1=TGFF		
		[15:7]	unused		
		[31:16]	input data (sequential load)		
0xF0D0_1004	AHBLAYER1_MISC_SET				
0xF0D0_2004	AHBLAYER1_MISC_CLR				
0xF0D0_0034	RAND_MISC		RNG_OUTPUT	RO	
		[15:0]	SHIFTREG output (muxed)		
		[19:16]	unused		
		[27:20]	VIRO Q		
		[31:28]	VIRO STM (bit 4 stripped)		

Figure B.2: Test chip address assignment.

Appendix C

Appendix: Chip Measurement Programming

The Chip measurement programs are present in this part, *Phyton* (programming language) is used for board controlling.

C.1 Shift register general functionality test

```
def cai_dualshiftreg_test():
        Vmin = 0.6              #set initail supply voltage
        power_state =1  #power state of power supply: 1 = power on,  0 = power off
        set_vdd_tti(pldcport,2, Vmin) #set voltage value of the power supply
        turn_x_tti(pldcport, 2, power_state) #on/off the power supply
        freq = 500      #initial operating frequency
        set_supply(’VBAT’, 0) #vbat off
        set_supply(’VREGI’, 0) #vregi off
        set_supply(’VESSI’, ’VESSI_XT’) #external supply for voltage
        set_supply(’VRNG’,1) #dualshiftreg is on VRNG power domain
        set_clk(’CKAXI’, ’PGFREQ2’, freq) #feed the frequency to the circuit clock port
        reset_dut() #reset dut
        m_u34410_conf(mrange=’0.0001’, nplc=’10’, samples=10)   #setting up multimeter
###########################################Writing Testing pattern###############
        print "####Specitic Pattern Testing start####"
######clean up the reg by shifting 0s#######################
        pattern = 0x0000
        for aa in range(20):
                Tokachi.memory_write(0xF0D00004, pattern<<16|3|1<<2)
                #enable dualshiftreg and set pattern
                Tokachi.memory_write(0xF0D00004, pattern<<16|3|1<<2|1<<3)
                #enable dualshiftreg and set pattern
        print "Reg_out:", Tokachi.memory_read(0xF0D00034) #read dualshiftreg’s output
        Tokachi.memory_write(0xF0D00004, pattern<<16|0) #disable clock
        time.sleep(0.1)
############################################################
##########################Writing in test pattern###########################
        loop_num = 21
        while (loop_num > 0):
                loop_num -= 1
                for test1 in range (1):
                        pattern = 0xAAAA
#                       print "Column 20","Reg_out:", Tokachi.memory_read(0xF0D00034)
                        #read dualshiftreg's output
                        Tokachi.memory_write(0xF0D00004, pattern<<16|3|1<<2)
                        #enable dualshiftreg and set pattern
                        Tokachi.memory_write(0xF0D00004, pattern<<16|3|1<<2|1<<3)
                        #enable dualshiftreg and set pattern
                        Tokachi.memory_write(0xF0D00004, pattern<<16|0)
                        #enable dualshiftreg and set pattern
                        time.sleep(0.1)
                for test2 in range (1):
                        pattern = 0x5555
#                       print "Column 20","Reg_out:", Tokachi.memory_read(0xF0D00034)
                        #read dualshiftreg's output
                        Tokachi.memory_write(0xF0D00004, pattern<<16|3|1<<2)
                        #enable dualshiftreg and set pattern
                        Tokachi.memory_write(0xF0D00004, pattern<<16|3|1<<2|1<<3)
                        #enable dualshiftreg and set pattern
                        Tokachi.memory_write(0xF0D00004, pattern<<16|0)
                        #enable dualshiftreg and set pattern
                        time.sleep(0.1)
        print "All set"
        Tokachi.memory_write(0xF0D00004, pattern<<16|0)
         #enable dualshiftreg and set pattern
        time.sleep(0.1)
        #####################TEST  MODE###################################
#       print " Parallel shifting: 18T TEST mode (Manual CK)"
#       for test in range (20):
#               print "Column 20","Reg_out:", Tokachi.memory_read(0xF0D00034)
#read dualshiftreg's output
#               Tokachi.memory_write(0xF0D00004, 1<<4|0<<6|1|1<<2)
#enable dualshiftreg and set pattern
#               Tokachi.memory_write(0xF0D00004, 1<<4|0<<6|1|1<<2|1<<3)
#enable dualshiftreg and set pattern
#       Tokachi.memory_write(0xF0D00004, pattern<<16|0)
#enable dualshiftreg and set pattern
#       print "Parallel shifting: TG TEST mode (Manual CK)"
#       for test in range (20):
#               print "Column 20","Reg_out:", Tokachi.memory_read(0xF0D00034)
#read dualshiftreg's output
#               Tokachi.memory_write(0xF0D00004, 1<<4|1<<6|2|1<<2)
#enable dualshiftreg and set pattern
#               Tokachi.memory_write(0xF0D00004, 1<<4|1<<6|2|1<<2|1<<3)
#enable dualshiftreg and set pattern
#       Tokachi.memory_write(0xF0D00004, pattern<<16|0)
#enable dualshiftreg and set pattern
#       time.sleep(0.1)
        ######################################################################
        Tokachi.memory_write(0xF0D00004, pattern<<16|3|1<<4)
        #parallel shifting start, both regs run
        #clock source now: IVR clock
        print "Rolling..."
        time.sleep(10)
        #vdd_cai = meas_v()
        itot = m_u34410_geti() #get current reading from multimeter
        #print 'voltage = ', vdd_cai
        print 'current = ', itot
        Tokachi.memory_write(0xF0D00004, pattern<<16|0) #disable dualshiftreg
        time.sleep(0.5)
        c_num = 20
        fail_flag =0
        print "Let's see the matrix:"
        for scan_out_D in range (20):
         time.sleep(0.5)
         c = int(Tokachi.memory_read(0xF0D00034),16)
         if (c == 0xAAAA or c == 0x5555):
                fail_flag = 0
         else:
                fail_flag = 1
                print "18T Shift-Reg, Column", c_num, "Fail", ':', "0x%x" %c
                power_state =0
                break
        Tokachi.memory_write(0xF0D00004, 1<<4|1|1<<2)
        #enable dualshiftreg and set pattern
        Tokachi.memory_write(0xF0D00004,1<<4|1|1<<2|1<<3)
        #enable dualshiftreg and set pattern
        e = int(Tokachi.memory_read(0xF0D00034),16)
##########the column x and column x+1 should not same at any bit#######################
          if (c&e != 0):
                print "18T Shift-Reg: FAIL", "0x%x" %c, "0x%x" %e
                fail_flag = 1
                power_state =0
                break
          c_num -= 1
        if(fail_flag == 0): print "18T Shift-Reg: PASS"
        Tokachi.memory_write(0xF0D00004, pattern<<16|0)
        #enable dualshiftreg and set pattern
#       c_num = 20
#       for scan_out_D in range (20):
#               d = int(Tokachi.memory_read(0xF0D00034),16)
#               if (c == 0xAAAA or c == 0x5555):
#               print "TGFF Shift-Reg, Column", c_num,"Reg_out:", "0x%x" %d,", correct"
#               else:
#               print "TGFF Shift-Reg, Column", c_num,"Reg_out:", "0x%x" %d,", wrong"
#               Tokachi.memory_write(0xF0D00004, 1<<4|1<<6|2|1<<2)
                #enable dualshiftreg and set pattern
#               Tokachi.memory_write(0xF0D00004, 1<<4|11<<6|2|1<<2|1<<3)
                #enable dualshiftreg and set pattern
#               c_num -= 1
#       Tokachi.memory_write(0xF0D00004, pattern<<16|0)
                #enable dualshiftreg and set pattern
#       time.sleep(0.1)
```

C.2 Shift register V_{min} measurement

```
def cai_dualshiftreg__18TVmin_test():
        power_state = 1
        Vmin = 0.5
        set_vdd_tti(pldcport,2, Vmin)
        turn_x_tti(pldcport, 2, power_state)
#       set_supply('VESSI', 'VDDSOC')
        freq = 19
        set_supply('VESSI', 'VESSI_XT') #external supply for voltage
        set_supply('VRNG',1) #dualshiftreg is on VRNG power domain
        set_clk('CKAXI', 'PGFREQ2', freq)
##Writing Testing pattern
        while (1):
                reset_dut()
                print "####Specitic Pattern Testing start####"
                pattern = 0x0000
                for aa in range(20):
                        Tokachi.memory_write(0xF0D00004, pattern<<16|3|1<<2)
                        #enable dualshiftreg and set pattern
                        Tokachi.memory_write(0xF0D00004, pattern<<16|3|1<<2|1<<3)
                        #enable dualshiftreg and set pattern
                print "Reg_out:", Tokachi.memory_read(0xF0D00034)
                #read dualshiftreg's output
                Tokachi.memory_write(0xF0D00004, pattern<<16|0)
                #disable clock
                time.sleep(0.1)
                loop_num = 21
                while (loop_num > 0):
                  loop_num -= 1
                  for test1 in range (1):
                  attern = 0xAAAA
        #         print "Column 20","Reg_out:", Tokachi.memory_read(0xF0D00034)
        #read dualshiftreg's output
                Tokachi.memory_write(0xF0D00004, pattern<<16|3|1<<2)
                #enable dualshiftreg and set pattern
                Tokachi.memory_write(0xF0D00004, pattern<<16|3|1<<2|1<<3)
                #enable dualshiftreg and set pattern
                Tokachi.memory_write(0xF0D00004, pattern<<16|0)
                #enable dualshiftreg and set pattern
                time.sleep(0.1)
        for test2 in range (1):
                pattern = 0x5555
        #       print "Column 20","Reg_out:", Tokachi.memory_read(0xF0D00034)
        #read dualshiftreg's output
                Tokachi.memory_write(0xF0D00004, pattern<<16|3|1<<2)
                #enable dualshiftreg and set pattern
                Tokachi.memory_write(0xF0D00004, pattern<<16|3|1<<2|1<<3)
                #enable dualshiftreg and set pattern
                Tokachi.memory_write(0xF0D00004, pattern<<16|0)
                #enable dualshiftreg and set pattern
                time.sleep(0.1)
print "All set"
                Tokachi.memory_write(0xF0D00004, pattern<<16|0)
                #enable dualshiftreg and set pattern
                time.sleep(0.1)
                ####################TEST  MODE###################################
        #       print " Parallel shifting: 18T TEST mode (Manual CK)"
        #       for test in range (20):
        #               print "Column 20","Reg_out:", Tokachi.memory_read(0xF0D00034)
        #read dualshiftreg's output
        #               Tokachi.memory_write(0xF0D00004, 1<<4|0<<6|1|1<<2)
        #enable dualshiftreg and set pattern
        #               Tokachi.memory_write(0xF0D00004, 1<<4|0<<6|1|1<<2|1<<3)
        #enable dualshiftreg and set pattern
        #       Tokachi.memory_write(0xF0D00004, pattern<<16|0)
        #enable dualshiftreg and set pattern
        #       print "Parallel shifting: TG TEST mode (Manual CK)"
        #       for test in range (20):
```

```
        #               print "Column 20","Reg_out:", Tokachi.memory_read(0xF0D00034)
        #read dualshiftreg's output
        #               Tokachi.memory_write(0xF0D00004, 1<<4|1<<6|2|1<<2)
        #enable dualshiftreg and set pattern
        #               Tokachi.memory_write(0xF0D00004, 1<<4|1<<6|2|1<<2|1<<3)
        #enable dualshiftreg and set pattern
        #       Tokachi.memory_write(0xF0D00004, pattern<<16|0)
        #enable dualshiftreg and set pattern
        #       time.sleep(0.1)
        ####################################################################
        Tokachi.memory_write(0xF0D00004, pattern<<16|3|1<<4)
        #parallel shifting start, both regs run
        #clock source now: IVR clock
        print "Rolling..."
        time.sleep(1)
        Tokachi.memory_write(0xF0D00004, pattern<<16|0)
        #disable dualshiftreg
        time.sleep(0.5)
        #dummy = 0x0000#dummy data
        fail_flag =0


#########Find the target from low to high, finding the first passing point#####

                c_num =20
                for scan_out_D in range (20):
                        time.sleep(0.5)
                        c = int(Tokachi.memory_read(0xF0D00034),16)
                        if (c == 0xAAAA or c == 0x5555):
                                        fail_flag = 0
                        else:
                          print "18T Shift-Reg, Column", c_num, "Fail", ':', "0x%x" %c
                          fail_flag = 1
#                               power_state = 1
                          print "18T Shift-Reg fail at", Vmin,'V.
                          at Temperature:', get_temp()
                          break
                        Tokachi.memory_write(0xF0D00004, 1<<4|1|1<<2)
                        #enable dualshiftreg and set pattern
                        Tokachi.memory_write(0xF0D00004,1<<4|1|1<<2|1<<3)
                        #enable dualshiftreg and set pattern
                        e = int(Tokachi.memory_read(0xF0D00034),16)
                        if (c&e != 0):
                                fail_flag = 1
#                               power_state = 0
                                print "18T Shift-Reg: FAIL", "0x%x" %c, "0x%x" %e
                                print "18T Shift-Reg fail at", Vmin,'V.'
                                break
                        c_num -= 1
                if(fail_flag == 0):
                        print "18T Shift-Reg: PASS at", Vmin,
                        'V at Temperature:', get_temp()
                        power_state = 0
                        turn_x_tti(pldcport,2, power_state)
                        break
                if (Vmin > 1.2):
                        power_state = 0
                        turn_x_tti(pldcport,2, power_state)
                        break
                if(fail_flag): Vmin += 0.01
                set_vdd_tti(pldcport,2,Vmin)
                turn_x_tti(pldcport,2, power_state)
                Tokachi.memory_write(0xF0D00004, pattern<<16|0)
                #enable dualshiftreg and set pattern
```

C.3 18TSPC Shift Register Power VS VDD measurement

```
def cai_18Tshiftreg_power_vs_Vdd_test():
        power_state = 1
        Vmin = 1.2
############
        set_supply(’VBAT’, 0)
        set_supply(’VREGI’, 0)
        set_supply(’VESSI’, ’VESSI_XT’) #external supply for voltage
        set_supply(’VRNG’,1) #dualshiftreg is on VRNG power domain
        set_vdd_tti(pldcport,2, Vmin)
        turn_x_tti(pldcport, 2, power_state)
        set_clk(’CKAXI’, ’PGFREQ2’, 66000)
        reset_dut()
        m_u34410_conf(mrange=’0.01’, nplc=’10’, samples=10)
############
##Writing Testing pattern
        while (1):
                print "####Specitic Pattern Testing start####"
                pattern = 0x0000
                for aa in range(20):
                        Tokachi.memory_write(0xF0D00004, pattern<<16|1|1<<2)
                        #enable dualshiftreg and set pattern
                        Tokachi.memory_write(0xF0D00004, pattern<<16|1|1<<2|1<<3)
                        #enable dualshiftreg and set pattern
                print "Reg_out:", Tokachi.memory_read(0xF0D00034)
                #read dualshiftreg’s output
                Tokachi.memory_write(0xF0D00004, pattern<<16|0)
                #disable clock
                time.sleep(0.1)
                loop_num = 21
                while (loop_num > 0):
                loop_num -= 1
                for test1 in range (1):
                        pattern = 0xAAAA
        #               print "Column 20","Reg_out:", Tokachi.memory_read(0xF0D00034)
        #read dualshiftreg’s output
                        Tokachi.memory_write(0xF0D00004, pattern<<16|1|1<<2)
                        #enable dualshiftreg and set pattern
                        Tokachi.memory_write(0xF0D00004, pattern<<16|1|1<<2|1<<3)
                        #enable dualshiftreg and set pattern
                        Tokachi.memory_write(0xF0D00004, pattern<<16|0)
                        #enable dualshiftreg and set pattern
                        time.sleep(0.1)
                for test2 in range (1):
                        pattern = 0x5555
        #               print "Column 20","Reg_out:", Tokachi.memory_read(0xF0D00034)
        #read dualshiftreg’s output
                        Tokachi.memory_write(0xF0D00004, pattern<<16|1|1<<2)
                        #enable dualshiftreg and set pattern
                        Tokachi.memory_write(0xF0D00004, pattern<<16|1|1<<2|1<<3)
                        #enable dualshiftreg and set pattern
                        Tokachi.memory_write(0xF0D00004, pattern<<16|0)
                        #enable dualshiftreg and set pattern
                        time.sleep(0.1)
                print "All set"
                Tokachi.memory_write(0xF0D00004, pattern<<16|0)
                #enable dualshiftreg and set pattern
                time.sleep(0.1)
                ####################TEST  MODE###################################
        #       print " Parallel shifting: 18T TEST mode (Manual CK)"
        #       for test in range (20):
        #       print "Column 20","Reg_out:", Tokachi.memory_read(0xF0D00034)
        #read dualshiftreg's output
        #       Tokachi.memory_write(0xF0D00004, 1<<4|0<<6|1|1<<2)
        #enable dualshiftreg and set pattern
        #       Tokachi.memory_write(0xF0D00004, 1<<4|0<<6|1|1<<2|1<<3)
        #enable dualshiftreg and set pattern
        #       Tokachi.memory_write(0xF0D00004, pattern<<16|0)
        #enable dualshiftreg and set pattern
        #       print "Parallel shifting: TG TEST mode (Manual CK)"
        #       for test in range (20):
        #       print "Column 20","Reg_out:", Tokachi.memory_read(0xF0D00034)
        #read dualshiftreg's output
        #       Tokachi.memory_write(0xF0D00004, 1<<4|1<<6|2|1<<2)
        #enable dualshiftreg and set pattern
        #       Tokachi.memory_write(0xF0D00004, 1<<4|1<<6|2|1<<2|1<<3)
        #enable dualshiftreg and set pattern
        #       Tokachi.memory_write(0xF0D00004, pattern<<16|0)
        #enable dualshiftreg and set pattern
        #       time.sleep(0.1)
                ######################################################################
                Tokachi.memory_write(0xF0D00004, pattern<<16|1|1<<4)
                #parallel shifting start, both regs run
                #clock source now: IVR clock
                print "Rolling..."
                time.sleep(1)
#               itot = float(m_u34410_geti())#*1e6#A to uA
#               Power_reg = Vmin*itot
#               print 'voltage = ', Vmin, 'V'
#               print 'current = ', itot, 'A'
                Tokachi.memory_write(0xF0D00004, pattern<<16|0)
                #disable dualshiftreg
                time.sleep(0.5)
                #dummy = 0x0000#dummy data
                fail_flag =0
                c_num =20
                for scan_out_D in range (20):
                        time.sleep(0.5)
                        c = int(Tokachi.memory_read(0xF0D00034),16)
                        if (c == 0xAAAA or c == 0x5555):
                                        fail_flag = 0
                        else:
                                print "18T Shift-Reg, Column", c_num, "Fail", ':'
                                        , "0x%x" %c
                                fail_flag = 1
        #                               power_state = 1
                                print "18T Shift-Reg fail at", Vmin,'V.'
                                break
                        Tokachi.memory_write(0xF0D00004, 1<<4|1|1<<2)
                        #enable dualshiftreg and set pattern
                        Tokachi.memory_write(0xF0D00004,1<<4|1|1<<2|1<<3)
                        #enable dualshiftreg and set pattern
                        e = int(Tokachi.memory_read(0xF0D00034),16)
                        if (c&e != 0):
                                fail_flag = 1
        #                               power_state = 0
                                print "18T Shift-Reg: FAIL", "0x%x" %c, "0x%x" %e
                                print "18T Shift-Reg fail at", Vmin,’V.’
                                break
                        c_num -= 1
                if (fail_flag == 0):
                        print "18T Shift-Reg: PASS at", Vmin, ’V.’
                        Tokachi.memory_write(0xF0D00004, pattern<<16|1|1<<4)
                        #parallel shifting start, 18T regs run
                        #clock source now: IVR clock
                        print "Rolling..."
                        time.sleep(10)
                        #read multimeter
                        itot = float(m_u34410_geti())#*1e6#A to uA
                        Power_reg = Vmin*itot
                        print ’voltage = ’, Vmin, ’V’
                        print ’current = ’, itot, ’A’
                        print ’18T shift-Reg at 100% activity, ’
                        print ’Power = ’, Power_reg,’W’
                        Vmin -= 0.05
                        if (Vmin < 0.6):
                                power_state = 0
                                turn_x_tti(pldcport,2, power_state)
                                break
                        set_vdd_tti(pldcport,2,Vmin)
                        turn_x_tti(pldcport,2, power_state)
        #       if (Vmin > 1.2):
        #               power_state = 0
        #               turn_x_tti(pldcport,2, power_state)
        #               break
                if(fail_flag): break
                Tokachi.memory_write(0xF0D00004, pattern<<16|0)
                #enable dualshiftreg and set pattern
```

C.4 Reference TGFF Shift Register Power VS VDD measurement

```
def cai_Refshiftreg_power_vs_Vdd_test():
        power_state = 1
        Vmin = 1.2
############
        set_supply(’VBAT’, 0)
        set_supply(’VREGI’, 0)
        set_supply(’VESSI’, ’VESSI_XT’) #external supply for voltage
        set_supply(’VRNG’,1) #dualshiftreg is on VRNG power domain
        set_vdd_tti(pldcport,2, Vmin)
        turn_x_tti(pldcport, 2, power_state)
        set_clk(’CKAXI’, ’PGFREQ2’, 66000)
        reset_dut()
        m_u34410_conf(mrange='0.01', nplc='10', samples=10)
############
        pattern = 0x0000
        for aa in range(20):
                Tokachi.memory_write(0xF0D00004, pattern<<16|1<<6|2|1<<2)
                #enable dualshiftreg and set pattern
                Tokachi.memory_write(0xF0D00004, pattern<<16|1<<6|2|1<<2|1<<3)
                #enable dualshiftreg and set pattern
        print "Reg_out:", Tokachi.memory_read(0xF0D00034) #read dualshiftreg's output
        Tokachi.memory_write(0xF0D00004, pattern<<16|0) #disable clock
##Writing Testing pattern
        while (1):
                print "####Specitic Pattern Testing start####"
                pattern = 0x0000
                for aa in range(20):
                        Tokachi.memory_write(0xF0D00004, pattern<<16|1<<6|2|1<<2)
                        #enable dualshiftreg and set pattern
                        Tokachi.memory_write(0xF0D00004, pattern<<16|1<<6|2|1<<2|1<<3)
                        #enable dualshiftreg and set pattern
                print "Reg_out:", Tokachi.memory_read(0xF0D00034)
                #read dualshiftreg's output
                Tokachi.memory_write(0xF0D00004, pattern<<16|0)
                #disable clock
                time.sleep(0.1)
                loop_num = 21
                while (loop_num > 0):
                        loop_num -= 1
                        for test1 in range (1):
                                pattern = 0x0000
        #                       print "Column 20","Reg_out:",
                                Tokachi.memory_read(0xF0D00034)
        #read dualshiftreg's output
                                Tokachi.memory_write(0xF0D00004,
                                pattern<<16|1<<6|2|1<<2)
                                #enable dualshiftreg and set pattern
                                Tokachi.memory_write(0xF0D00004,
                                pattern<<16|1<<6|2|1<<2|1<<3)
                                #enable dualshiftreg and set pattern
                                Tokachi.memory_write(0xF0D00004,
                                pattern<<16|1<<6|0)
                                #enable dualshiftreg and set pattern
                                time.sleep(0.1)
                        for test2 in range (1):
                                pattern = 0xFFFF
        #                       print "Column 20","Reg_out:",
        #                       Tokachi.memory_read(0xF0D00034)
        #read dualshiftreg's output
                                Tokachi.memory_write(0xF0D00004,
                                pattern<<16|1<<6|2|1<<2)
                                #enable dualshiftreg and set pattern
                                Tokachi.memory_write(0xF0D00004,
                                pattern<<16|1<<6|2|1<<2|1<<3)
                                #enable dualshiftreg and set pattern
                                Tokachi.memory_write(0xF0D00004,
                                pattern<<16|1<<6|0)
                                #enable dualshiftreg and set pattern
                                time.sleep(0.1)
                print "All set"
                Tokachi.memory_write(0xF0D00004, pattern<<16|0)
                #enable dualshiftreg and set pattern
                time.sleep(0.1)
                #####################TEST  MODE###################################
#               print " Parallel shifting: 18T TEST mode (Manual CK)"
#               for test in range (20):
#                       print "Column 20","Reg_out:", Tokachi.memory_read(0xF0D00034)
#read dualshiftreg’s output
#                       Tokachi.memory_write(0xF0D00004, 1<<4|0<<6|1|1<<2)
#enable dualshiftreg and set pattern
#                       Tokachi.memory_write(0xF0D00004, 1<<4|0<<6|1|1<<2|1<<3)
#enable dualshiftreg and set pattern
#               Tokachi.memory_write(0xF0D00004, pattern<<16|0)
#enable dualshiftreg and set pattern
#               print "Parallel shifting: TG TEST mode (Manual CK)"
#               for test in range (20):
#                       print "Column 20","Reg_out:", Tokachi.memory_read(0xF0D00034)
#read dualshiftreg’s output
#                       Tokachi.memory_write(0xF0D00004, 1<<4|1<<6|2|1<<2)
                        #enable dualshiftreg and set pattern
#                       Tokachi.memory_write(0xF0D00004, 1<<4|1<<6|2|1<<2|1<<3)
#enable dualshiftreg and set pattern
#               Tokachi.memory_write(0xF0D00004, pattern<<16|0)
#enable dualshiftreg and set pattern
#               time.sleep(0.1)
                ######################################################################
                Tokachi.memory_write(0xF0D00004, pattern<<16|1<<6|2|1<<4)
                #parallel shifting start, both regs run
                #clock source now: IVR clock
                print "Rolling..."
                time.sleep(1)
                Tokachi.memory_write(0xF0D00004, pattern<<16|1<<6|0)
                #disable dualshiftreg
                time.sleep(0.5)
                fail_flag =0
                c_num =20
                for scan_out_D in range (20):
                        time.sleep(0.5)
                        c = int(Tokachi.memory_read(0xF0D00034),16)
                        if (c == 0xAAAA or c == 0x5555):
                                        fail_flag = 0
                        else:
                                print "REF Shift-Reg, Column",
                                c_num, "Fail", ’:’, "0x%x" %c
                                fail_flag = 1
        #                               power_state = 1
                                print "REF Shift-Reg fail at", Vmin,’V.’
                                break
                        Tokachi.memory_write(0xF0D00004, 1<<4|1<<6|2|1<<2)
                        #enable dualshiftreg and set pattern
                        Tokachi.memory_write(0xF0D00004,1<<4|1<<6|2|1<<2|1<<3)
                        #enable dualshiftreg and set pattern
                        e = int(Tokachi.memory_read(0xF0D00034),16)
                        if (c&e != 0):
                                fail_flag = 1
        #                               power_state = 0
                                print "REF Shift-Reg: FAIL", "0x%x" %c, "0x%x" %e
                                print "REF Shift-Reg fail at", Vmin,’V.’
                                break
                        c_num -= 1
                if (fail_flag == 0):
                        print "REF Shift-Reg: PASS at", Vmin, 'V.'
                        Tokachi.memory_write(0xF0D00004, pattern<<16|1<<6|2|1<<4)
                        #parallel shifting start, REF regs run
                        #clock source now: IVR clock
                        print "Rolling..."
                        time.sleep(10)
                        #read multimeter
                        itot = float(m_u34410_geti())#*1e6#A to uA
                        Power_reg = Vmin*itot
                        print 'voltage = ', Vmin, 'V'
                        print 'current = ', itot, 'A'
                        print 'REF shift-Reg at 100% activity,
                        Power = ', Power_reg,'W'
                        Vmin -= 0.05
                        if (Vmin < 0.6):
                                power_state = 0
                                turn_x_tti(pldcport,2, power_state)
                                break
                        set_vdd_tti(pldcport,2,Vmin)
                        turn_x_tti(pldcport,2, power_state)
                if(fail_flag):
                        power_state = 0
                        turn_x_tti(pldcport,2, power_state)
                        break
                Tokachi.memory_write(0xF0D00004, pattern<<16|0)
                #enable dualshiftreg and set pattern
```

C.5 18TSPC CK Power VS VDD measurement

```
def cai_18Tshiftreg_CKpower_vs_Vdd_test():
        power_state = 1
        Vmin = 1.2
############
        set_supply('VBAT', 0)
        set_supply('VREGI', 0)
        set_supply('VESSI', 'VESSI_XT') #external supply for voltage
        set_supply('VRNG',1) #dualshiftreg is on VRNG power domain
        set_vdd_tti(pldcport,2, Vmin)
        turn_x_tti(pldcport, 2, power_state)
        set_clk('CKAXI', 'PGFREQ2', 66000)
        reset_dut()
        m_u34410_conf(mrange='0.01', nplc='10', samples=10)
############
##Writing Testing pattern
        while (1):
                print "####Specitic Pattern Testing start####"
                pattern = 0x0000
                for aa in range(20):
                        Tokachi.memory_write(0xF0D00004, pattern<<16|1|1<<2)
                        #enable dualshiftreg and set pattern
                        Tokachi.memory_write(0xF0D00004, pattern<<16|1|1<<2|1<<3)
                        #enable dualshiftreg and set pattern
                print "Reg_out:", Tokachi.memory_read(0xF0D00034)
                #read dualshiftreg's output
                Tokachi.memory_write(0xF0D00004, pattern<<16|0)
                #disable clock
                time.sleep(0.1)
                print "All set"
                fail_flag =0
                c_num =20
                for scan_out_D in range (20):
                        time.sleep(0.5)
                        c = int(Tokachi.memory_read(0xF0D00034),16)
                        if (c == 0x0000):
                                        fail_flag = 0
                        else:
                                print "18T Shift-Reg, Column",
                                c_num, "Fail", ':', "0x%x" %c
                                fail_flag = 1
        #                               power_state = 1
                                print "18T Shift-Reg fail at", Vmin,'V.'
                                break
                        Tokachi.memory_write(0xF0D00004, 1<<4|1|1<<2)
                        #enable dualshiftreg and set pattern
                        Tokachi.memory_write(0xF0D00004,1<<4|1|1<<2|1<<3)
                        #enable dualshiftreg and set pattern
                        e = int(Tokachi.memory_read(0xF0D00034),16)
                        if (c&e != 0x0000):
                                fail_flag = 1
        #                               power_state = 0
                                print "18T Shift-Reg: FAIL", "0x%x" %c, "0x%x" %e
                                print "18T Shift-Reg fail at", Vmin,'V.'
                                break
                        c_num -= 1
                if (fail_flag == 0):
                        print "18T Shift-Reg: PASS at", Vmin, 'V.'
                        Tokachi.memory_write(0xF0D00004, pattern<<16|1|1<<4)
                        #parallel shifting start, 18T regs run
                        #clock source now: IVR clock
                        print "Rolling..."
                        time.sleep(10)
                        #read multimeter
                        itot = float(m_u34410_geti())#*1e6#A to uA
                        Power_reg = Vmin*itot
                        print 'voltage = ', Vmin, 'V'
                        print 'current = ', itot, 'A'
                        print '18T shift-Reg at 0% activity,
                        Power = ', Power_reg,'W'
                        Vmin -= 0.05
                        if (Vmin < 0.6):
                                power_state = 0
                                turn_x_tti(pldcport,2, power_state)
                                break
                        set_vdd_tti(pldcport,2,Vmin)
                        turn_x_tti(pldcport,2, power_state)
                if(fail_flag): break
                Tokachi.memory_write(0xF0D00004, pattern<<16|0)
                #enable dualshiftreg and set pattern
```

C.6 Ref TGFF CK Power VS VDD measurement

```
def cai_Refshiftreg_CKpower_vs_Vdd_test():
        power_state = 1
        Vmin = 1.2
############
        set_supply('VBAT', 0)
        set_supply('VREGI', 0)
        set_supply('VESSI', 'VESSI_XT') #external supply for voltage
        set_supply('VRNG',1) #dualshiftreg is on VRNG power domain
        set_vdd_tti(pldcport,2, Vmin)
        turn_x_tti(pldcport, 2, power_state)
        set_clk('CKAXI', 'PGFREQ2', 66000)
        reset_dut()
        m_u34410_conf(mrange='0.01', nplc='10', samples=10)
############
        pattern = 0xFFFF
        for aa in range(20):
                Tokachi.memory_write(0xF0D00004, pattern<<16|1<<6|2|1<<2)
                #enable dualshiftreg and set pattern
                Tokachi.memory_write(0xF0D00004, pattern<<16|1<<6|2|1<<2|1<<3)
                #enable dualshiftreg and set pattern
        print "Reg_out:", Tokachi.memory_read(0xF0D00034)
        #read dualshiftreg's output
        Tokachi.memory_write(0xF0D00004, pattern<<16|0)
        #disable clock
        print "All set"
##Writing Testing pattern
        while (1):
                Tokachi.memory_write(0xF0D00004, pattern<<16|1<<6|0)
                #enable dualshiftreg and set pattern
                time.sleep(0.1)
                fail_flag =0
                c_num =20
                for scan_out_D in range (20):
                        time.sleep(0.5)
                        c = int(Tokachi.memory_read(0xF0D00034),16)
                        if (c == 0xFFFF):
                                        fail_flag = 0
                        else:
                                print "REF Shift-Reg, Column",
                                c_num, "Fail", ':', "0x%x" %c
                                fail_flag = 1
        #                               power_state = 1
                                print "REF Shift-Reg fail at", Vmin,'V.'
                                break
                        Tokachi.memory_write(0xF0D00004, 1<<4|1<<6|2|1<<2)
                        #enable dualshiftreg and set pattern
                        Tokachi.memory_write(0xF0D00004,1<<4|1<<6|2|1<<2|1<<3)
                        #enable dualshiftreg and set pattern
                        e = int(Tokachi.memory_read(0xF0D00034),16)
                        if (c&e != 0xFFFF):
                                fail_flag = 1
        #                               power_state = 0
                                print "REF Shift-Reg: FAIL", "0x%x" %c, "0x%x" %e
                                print "REF Shift-Reg fail at", Vmin,'V.'
                                break
                        c_num -= 1
                if (fail_flag == 0):
                        print "REF Shift-Reg: PASS at", Vmin, ’V.’
                        Tokachi.memory_write(0xF0D00004, pattern<<16|1<<6|2|1<<4)
                        #parallel shifting start, REF regs run
                        #clock source now: IVR clock
                        print "Rolling..."
                        time.sleep(10)
                        #read multimeter
                        itot = float(m_u34410_geti())#*1e6#A to uA
                        Power_reg = Vmin*itot
                        print ’voltage = ’, Vmin, ’V’
                        print ’current = ’, itot, ’A’
                        print ’REF shift-Reg at 0% activity,
                        Power = ’, Power_reg,’W’
                        Vmin -= 0.05
                        if (Vmin < 0.6):
                                power_state = 0
                                turn_x_tti(pldcport,2, power_state)
                                break
                        set_vdd_tti(pldcport,2,Vmin)
                        turn_x_tti(pldcport,2, power_state)
                if(fail_flag):
                        power_state = 0
                        turn_x_tti(pldcport,2, power_state)
                        break
                Tokachi.memory_write(0xF0D00004, pattern<<16|0)
                #enable dualshiftreg and set pattern
```

C.7 18TSPC Shift Register power VS activity rate measurement

```
def cai_18Tshiftreg_P_vs_D_rate_test():
        power_state = 1
        Vmin = 0.6
############
        set_supply(’VBAT’, 0)
        set_supply(’VREGI’, 0)
        set_supply(’VESSI’, ’VESSI_XT’) #external supply for voltage
        set_supply(’VRNG’,1) #dualshiftreg is on VRNG power domain
        set_vdd_tti(pldcport,2, Vmin)
        turn_x_tti(pldcport, 2, power_state)
        set_clk(’CKAXI’, ’PGFREQ2’, 500)
        m_u34410_conf(mrange=’0.01’, nplc=’10’, samples=10)
        reset_dut()
############
        loop_reduction = 0
##Writing Testing pattern
        while (1):
                print "####Specitic Pattern Testing start####"
                pattern = 0x0000
                for aa in range(20):
                        Tokachi.memory_write(0xF0D00004, pattern<<16|1|1<<2)
                        #enable dualshiftreg and set pattern
                        Tokachi.memory_write(0xF0D00004, pattern<<16|1|1<<2|1<<3)
                        #enable dualshiftreg and set pattern
                print "Reg_out:", Tokachi.memory_read(0xF0D00034)
                #read dualshiftreg's output
                Tokachi.memory_write(0xF0D00004, pattern<<16|0)
                #disable clock
                time.sleep(0.1)
                loop_num = 10 - loop_reduction
                print loop_num
                print loop_reduction
                for loop in range (loop_num):
#                       loop_num -= 1
                        for test1 in range (1):
                          pattern = 0xFFFF
                                #change to 0x0000 for complemented initial data
        #                 print "Column 20","Reg_out:",
        #Tokachi.memory_read(0xF0D00034)
        #read dualshiftreg's output
                        Tokachi.memory_write(0xF0D00004, pattern<<16|1|1<<2)
                        #enable dualshiftreg and set pattern
                        Tokachi.memory_write(0xF0D00004, pattern<<16|1|1<<2|1<<3)
                        #enable dualshiftreg and set pattern
                        Tokachi.memory_write(0xF0D00004, pattern<<16|0)
                        #enable dualshiftreg and set pattern
                        time.sleep(0.1)
                        for test2 in range (1):
                                pattern = 0x0000
                                #change to 0xFFFF for complemented initial data
        #                       print "Column 20","Reg_out:",
        #Tokachi.memory_read(0xF0D00034)
        #read dualshiftreg's output
                                Tokachi.memory_write(0xF0D00004,
                                pattern<<16|1|1<<2)
                                 #enable dualshiftreg and set pattern
                                Tokachi.memory_write(0xF0D00004,
                                pattern<<16|1|1<<2|1<<3)
                                #enable dualshiftreg and set pattern
                                Tokachi.memory_write(0xF0D00004,
                                pattern<<16|0)
                                #enable dualshiftreg and set pattern
                                time.sleep(0.1)
                print "All set"
                Tokachi.memory_write(0xF0D00004, pattern<<16|0)
                #enable dualshiftreg and set pattern
                time.sleep(0.1)
                #####################TEST  MODE####################################
                print " Parallel shifting: 18T TEST mode (Manual CK)"
                for test in range (20):
                        print "Matrix:", Tokachi.memory_read(0xF0D00034)
                        #read dualshiftreg's output
                        Tokachi.memory_write(0xF0D00004, 1<<4|1|1<<2)
                        #enable dualshiftreg and set pattern
                        Tokachi.memory_write(0xF0D00004, 1<<4|1|1<<2|1<<3)
                        #enable dualshiftreg and set pattern
                Tokachi.memory_write(0xF0D00004, pattern<<16|0)
                #enable dualshiftreg and set pattern
        #       print "Parallel shifting: TG TEST mode (Manual CK)"
        #       for test in range (20):
        #               print "Column 20","Reg_out:", Tokachi.memory_read(0xF0D00034)
        #read dualshiftreg's output
        #               Tokachi.memory_write(0xF0D00004, 1<<4|1<<6|2|1<<2)
        #enable dualshiftreg and set pattern
```

```
#               Tokachi.memory_write(0xF0D00004, 1<<4|1<<6|2|1<<2|1<<3)
#enable dualshiftreg and set pattern
#       Tokachi.memory_write(0xF0D00004, pattern<<16|0)
#enable dualshiftreg and set pattern
#       time.sleep(0.1)
        #####################################################################
        Tokachi.memory_write(0xF0D00004, pattern<<16|0)
        #disable dualshiftreg
        time.sleep(0.1)
        #dummy = 0x0000#dummy data
        fail_flag =0
        c_num =20
        for scan_out_D in range (20):
                time.sleep(0.5)
                c = int(Tokachi.memory_read(0xF0D00034),16)
                if (c == 0xFFFF or c == 0x0000):
                                fail_flag = 0
                else:
                        print "18T Shift-Reg, Column",
                        c_num, "Fail", ':', "0x%x" %c
                        fail_flag = 1
                        print "18T Shift-Reg fail at", Vmin,'V.'
                        break
                Tokachi.memory_write(0xF0D00004, 1<<4|1|1<<2)
                #enable dualshiftreg and set pattern
                Tokachi.memory_write(0xF0D00004,1<<4|1|1<<2|1<<3)
                #enable dualshiftreg and set pattern
                c_num -= 1
        if (fail_flag == 0):
                print "18T Shift-Reg: PASS at", Vmin, 'V.'
                Tokachi.memory_write(0xF0D00004, pattern<<16|1|1<<4)
                #parallel shifting start, 18T regs run
                #clock source now: IVR clock
                print "Rolling..."
                time.sleep(10)
                #read multimeter
                itot = float(m_u34410_geti())#*1e6#A to uA
                Power_reg = Vmin*itot*1e6
                print 'voltage = ', Vmin, 'V'
                print 'current = ', itot, 'A'
                print '18T shift-Reg at 100% activity,
                Power = ', Power_reg,'uW'
                loop_reduction += 1
                if (loop_reduction > 10):
                        power_state = 0
                        turn_x_tti(pldcport,2, power_state)
                        break
        if(fail_flag):
                power_state = 0
                turn_x_tti(pldcport,2, power_state)
                break
        Tokachi.memory_write(0xF0D00004, pattern<<16|0)
        #enable dualshiftreg and set pattern
```

C.8 Ref TGFF Shift Register power vs. activity rate measurement

```
def cai_Refshiftreg_P_vs_D_rate_test():
        power_state = 1
        vdd = 0.6
        freq = 500
############
        set_supply('VBAT', 0)
        set_supply('VREGI', 0)
        set_supply('VESSI', 'VESSI_XT') #external supply for voltage
        set_supply('VRNG',1) #dualshiftreg is on VRNG power domain
        set_vdd_tti(pldcport,2, vdd)
        turn_x_tti(pldcport, 2, power_state)
        set_clk('CKAXI', 'PGFREQ2', freq)
        m_u34410_conf(mrange='0.01', nplc='10', samples=10)
        reset_dut()
############
        loop_reduction = 0
##Writing Testing pattern
        while (1):
                print "####Specitic Pattern Testing start####"
                pattern = 0x0000
                for aa in range(20):
                        Tokachi.memory_write(0xF0D00004,
                        pattern<<16|1<<6|2|1<<2)
                        #enable dualshiftreg and set pattern
                        Tokachi.memory_write(0xF0D00004,
                        pattern<<16|1<<6|2|1<<2|1<<3)
                        #enable dualshiftreg and set pattern
                print "Reg_out:", Tokachi.memory_read(0xF0D00034)
                #read dualshiftreg's output
                Tokachi.memory_write(0xF0D00004, pattern<<16|1<<6|0)
                #disable clock
                time.sleep(0.1)
                loop_num = 10 - loop_reduction
                print loop_num
                print loop_reduction
                for loop in range (loop_num):
#                       loop_num -= 1
                        for test1 in range (1):
                                pattern = 0xFFFF
        #                       print "Column 20","Reg_out:",
        #Tokachi.memory_read(0xF0D00034)
        #read dualshiftreg's output
                                Tokachi.memory_write(0xF0D00004,
                                pattern<<16|1<<6|2|1<<2)
                                #enable dualshiftreg and set pattern
                                Tokachi.memory_write(0xF0D00004,
                                pattern<<16|1<<6|2|1<<2|1<<3)
                                #enable dualshiftreg and set pattern
                                Tokachi.memory_write(0xF0D00004,
                                pattern<<16|1<<6|0)
                                #enable dualshiftreg and set pattern
                                time.sleep(0.1)
                        for test2 in range (1):
                                pattern = 0x0000
        #                       print "Column 20","Reg_out:",
        #Tokachi.memory_read(0xF0D00034)
        #read dualshiftreg's output
                                Tokachi.memory_write(0xF0D00004,
                                pattern<<16|1<<6|2|1<<2)
                                #enable dualshiftreg and set pattern
                                Tokachi.memory_write(0xF0D00004,
                                pattern<<16|1<<6|2|1<<2|1<<3)
                                #enable dualshiftreg and set pattern
                                Tokachi.memory_write(0xF0D00004,
                                pattern<<16|1<<6|0)
                                #enable dualshiftreg and set pattern
                                time.sleep(0.1)
                print "All set"
                #####################TEST  MODE###################################
#               print " Parallel shifting: 18T TEST mode (Manual CK)"
#               for test in range (20):
#                       print "Matrix:", Tokachi.memory_read(0xF0D00034)
#read dualshiftreg's output
#                       Tokachi.memory_write(0xF0D00004, 1<<4|2|1<<2)
#enable dualshiftreg and set pattern
#                       Tokachi.memory_write(0xF0D00004, 1<<4|2|1<<2|1<<3)
#enable dualshiftreg and set pattern
#               Tokachi.memory_write(0xF0D00004, pattern<<16|1<<6|0)
#enable dualshiftreg and set pattern
                Tokachi.memory_write(0xF0D00004, pattern<<16|1<<6|0)
                #enable dualshiftreg and set pattern
                time.sleep(0.1)
                print "Parallel shifting: TG TEST mode (Manual CK)"
                for test in range (20):
                        print "REF matrix:", Tokachi.memory_read(0xF0D00034)
                        #read dualshiftreg's output
                        Tokachi.memory_write(0xF0D00004, 1<<4|1<<6|2|1<<2)
                        #enable dualshiftreg and set pattern
                        Tokachi.memory_write(0xF0D00004, 1<<4|1<<6|2|1<<2|1<<3)
                        #enable dualshiftreg and set pattern
                Tokachi.memory_write(0xF0D00004, pattern<<16|1<<6|0)
                #enable dualshiftreg and set pattern
                time.sleep(0.1)
                ####################################################################
                #dummy = 0x0000#dummy data
                fail_flag =0
                c_num =20
                for scan_out_D in range (20):
                        time.sleep(0.5)
                        c = int(Tokachi.memory_read(0xF0D00034),16)
                        if (c == 0xFFFF or c == 0x0000):
                                        fail_flag = 0
                        else:
                                print "18T Shift-Reg, Column",
                                c_num, "Fail", ':', "0x%x" %c
                                fail_flag = 1
                                print "18T Shift-Reg fail at", vdd,'V.'
                                break
                        Tokachi.memory_write(0xF0D00004, 1<<4|1<<6|2|1<<2)
                        #enable dualshiftreg and set pattern
                        Tokachi.memory_write(0xF0D00004,1<<4|1<<6|2|1<<2|1<<3)
                        #enable dualshiftreg and set pattern
                        c_num -= 1
                if (fail_flag == 0):
                        print "18T Shift-Reg: PASS at", vdd, ’V.’
                        Tokachi.memory_write(0xF0D00004, pattern<<16|1<<6|2|1<<4)
                        #parallel shifting start, 18T regs run
                        #clock source now: IVR clock
                        print "Rolling..."
                        time.sleep(10)
                        #read multimeter
                        itot = float(m_u34410_geti())#*1e6#A to uA
                        Power_reg = vdd*itot*1e6
                        print ’voltage = ’, vdd, ’V’
                        print ’current = ’, itot, ’A’
                        print ’18T shift-Reg at 100% activity,
                        Power = ’, Power_reg,’uW’
                        loop_reduction += 1
                if(fail_flag):
                        power_state = 0
                        turn_x_tti(pldcport,2, power_state)
                        break
                turn_x_tti(pldcport,2, power_state)
                if (loop_reduction > 10):
                        power_state = 0
                        turn_x_tti(pldcport,2, power_state)
                        break
        Tokachi.memory_write(0xF0D00004, pattern<<16|1<<6|0)
        #enable dualshiftreg and set pattern
        time.sleep(0.1)
```

C.9 AES scanin/out at different VDD

```
def AES_scantest():
        init_vdd = 1.2
        final_vdd = 0.6
        vstep = -0.1
        freq = 66000
        vesso= 0
        reset_dut()
        power_state = 1
        vdds = np.arange(init_vdd, final_vdd, vstep)
        x = np.zeros(len(vdds),
        dtype={’names’:[’vdd’, ’freq’, ’ileak’, ’itot’],
        ’formats’:[’f4’,’f4’,’f4’,’f4’]})
        x[’vdd’] = vdds
############set supply voltage###############################
        set_supply(’VBAT’, 0)
        set_supply(’VREGI’, 0)
        set_supply(’VESSI’, ’VESSI_XT’)
        #external supply for voltage
        set_supply(’VRNG’,1)
        #dualshiftreg is on VRNG power domain
############################################################
#       m_u34410_conf(mrange=’0.0001’, triggers=1, samples=100)
        m_u34410_conf(mrange=’0.01’, nplc=’10’, samples=10)
        for d in x:
                vdd = d[’vdd’]
                print "\n=============================\n"
                print "Test running at VDD = \n", vdd
                set_clk('CKAXI', 'PGFREQ2', freq)
                MbedTx('rstdut\n')
                Tokachi.get_prompt()
#               set_vdac(SAFE_VDD)
                set_vdd_tti(pldcport,2, vdd)
                turn_x_tti(pldcport, 2, power_state)
###########SCANIN-SCANOUT TEST###################################
#               Tokachi.memory_write(AHBLAYER1_CTRL,0xA<<8|1<<1|1<<7)
#               reference test
                Tokachi.memory_write(AHBLAYER1_CTRL,0xF<<8)
                time.sleep(0.5)
                Tokachi.memory_write
                (AHBLAYER1_CTRL,0xF<<8|AES_LBIST_SCAN_ENABLE|AES_LBIST_CK_ENABLE)
                time.sleep(1)
                scan_pattern_cai =
                (int(Tokachi.memory_read(AHBLAYER2_DIAG21),16)>>20)&0xf
                scan_pattern_ref =
                (int(Tokachi.memory_read(AHBLAYER2_DIAG21),16)>>16)&0xf
                #scan out results
                print hex(scan_pattern_cai)
        #       print hex(scan_pattern_ref)
#               print bin(Tokachi.memory_read(AHBLAYER2_DIAG21))
#REF-scanout results

#               print bin(scan_pattern)

def do_CAI_aes_lbist
(vdd=0, freq=0, run_once=0, verbose=0):
        Tokachi.memory_write(AHBLAYER1_CTRL,
        AES_LBIST_RESET|AES_LBIST_CK_ENABLE)
        time.sleep(0.1)
        Tokachi.memory_write(AHBLAYER1_CTRL,
        nAES_LBIST_RESET|AES_LBIST_CK_ENABLE)
        time.sleep(0.1)
        Tokachi.memory_write(AHBLAYER1_CTRL,
        AES_LBIST_CK_DISABLE)
        time.sleep(0.5)
        if run_once:
                Tokachi.memory_write(AHBLAYER1_CTRL_SET,
                ENABLE_CAI_AES_LBIST|AES_RUN_ONCE)
                #ENABLE LBIST - run once
#reference
#               Tokachi.memory_write(AHBLAYER1_CTRL_SET,
                ENABLE_REF_AES_LBIST|AES_RUN_ONCE)
                print 'RUN=ONCE'
        else:
                Tokachi.memory_write(AHBLAYER1_CTRL_SET,
                ENABLE_CAI_AES_LBIST)
                #ENABLE LBIST - loop
                print 'RUN=LOOP'
        time.sleep(0.5)
        Tokachi.memory_write(AHBLAYER1_CTRL_SET,
        AES_LBIST_CK_ENABLE)
        #START CLOCK
        time.sleep(5)
        #change to 10 for current measurement
        itot = 0
#       itot = float(m_u34410_geti())
        Tokachi.memory_write(AHBLAYER1_CTRL,
        AES_LBIST_CK_DISABLE)
        #STOP CLOCK
        time.sleep(1)
        if verbose: print
        Tokachi.memory_read(AHBLAYER2_DIAG21)
        lbistresult =
        (int(Tokachi.memory_read(AHBLAYER2_DIAG21),16)>>4)&0xf
        #read CAI_AES states
#       lbistresult =
#(int(Tokachi.memory_read(AHBLAYER2_DIAG21),16))&0xf
#read REF_AES states
        time.sleep(0.5)
        print ’lbistresult is:’, lbistresult
        if ((lbistresult == 7) or (run_once and lbistresult == 6)) :
                fail_flag = 0
        else:
                fail_flag = 1
        Tokachi.memory_write(AHBLAYER1_CTRL, 0XFFFF0000)
         #DISABLE LBIST
        return [fail_flag, vdd, freq, itot, lbistresult]
```

C.10 AES LBIST based F_{max} measurement

```
def AES_lbist_Fmax_sweep():
        init_vdd = 0.8
        final_vdd = 0.9
        vstep = 0.01
        init_freq = 5363
        itot=0
        vesso= 0
        reset_dut()
        vdds = np.arange(init_vdd, final_vdd, vstep)
        x = np.zeros(len(vdds),
        dtype={’names’:[’vdd’, ’freq’, ’itot’],
        ’formats’:[’f4’,’f4’,’f4’]})
        x[’vdd’] = vdds
############set supply voltage###############################
        set_supply(’VBAT’, 0)
        set_supply(’VREGI’, 0)
        set_supply(’VESSI’, ’VESSI_XT’)
        #external supply for voltage
        set_supply(’VRNG’,1)
        #dualshiftreg is on VRNG power domain
###########################################################
#       m_u34410_conf(mrange=’0.001’, nplc=’10’, samples=10)
        for d in x:
                freq = init_freq
                vdd = d[’vdd’]
                print "\n============================\n"
                print "Test running at VDD = \n", vdd
                MbedTx(’rstdut\n’)
                Tokachi.get_prompt()
                power_state = 1
                set_vdd_tti(pldcport,2, vdd)
                turn_x_tti(pldcport, 2, power_state)
                list_all = [0,0,0,0,0]
                while(1):
                        reset_dut()
                        set_clk('CKAXI', 'PGFREQ2', freq)
                        time.sleep(1)
        ###########SCANIN-SCANOUT TEST###################################
                        Tokachi.memory_write(AHBLAYER1_CTRL,0xA<<8)
                        time.sleep(0.5)
                        Tokachi.memory_write
                        (AHBLAYER1_CTRL,0xA<<8|
                        AES_LBIST_SCAN_ENABLE|AES_LBIST_CK_ENABLE)
                        time.sleep(1)
                        scan_pattern_cai =
                        (int(Tokachi.memory_read(AHBLAYER2_DIAG21),16)>>20)&0xf
                        #scan out CAI_AES results
        #               scan_pattern_ref =
        #               (int(Tokachi.memory_read(AHBLAYER2_DIAG21),16)>>16)&0xf
                        #scan out ref aes results
                        if (scan_pattern_cai == 0xa):
                                print 'CAI_AES, scan out:',
                                hex(scan_pattern_cai),', PASS!'
                                Tokachi.memory_write(AHBLAYER1_CTRL,
                                AES_LBIST_SCAN_DISABLE)
                                list_all = do_CAI_aes_lbist(vdd, freq,0,0)
                                print list_all
                                fail_flag = list_all[0]
#                               itot = float(list_all[3])
                                lbistresult = int(list_all[4])
                                print 'lbistresult is:', bin(lbistresult)
                                if(not fail_flag):
#                                       x['itot'] = itot
                                        x['freq'] = freq
                                if (fail_flag):
                                        print 'AES_LBIST_FAILs at', vdd,'V', freq,'kHz'
                                        power_state = 0
                                        turn_x_tti(pldcport, 2, power_state)
                                        break
#                               if(itot>1e-5):
#                                       m_u34410_conf(mrange='0.01', samples=100)
                                print  'VDD =', vdd,'Current =',
                                itot,'at', freq, 'kHz.'
#                               power = vdd*itot*1e6
#                               print 'power' ,power,'uW'
                        if (scan_pattern_cai != 0xa):
                                print 'CAI_AES, scan out:',
                                hex(scan_pattern_cai),', FAIL!'
                                print 'Fmax of CAI_AES at',
                                vdd,'V is', freq,'kHz'
                                power_state = 0
                                turn_x_tti(pldcport, 2, power_state)
                                break
                        if(freq==66000):
                                break
                        freq += max(1, int(freq*0.1))
                        freq = min(66000, freq)
        turn_x_tti(pldcport, 2, 0)
        return x
```

C.11 AES LBIST based V_{min} measurement

```
def cai_AES128_18TVmin_test():
        power_state = 1
        vdd = 0.5
        set_vdd_tti(pldcport,2, vdd)
        turn_x_tti(pldcport, 2, power_state)
        freq = 2
        set_supply(’VESSI’, ’VESSI_XT’)
        #external supply for voltage
        set_supply(’VRNG’,1)
        #dualshiftreg is on VRNG power domain
        set_clk(’CKAXI’, ’PGFREQ2’, freq)
##Writing Testing pattern
        while (1):
                reset_dut()
                fail_flag = 0
                print "####Specitic Pattern Testing start####"
                Tokachi.memory_write(AHBLAYER1_CTRL,0xA<<8)
                time.sleep(0.5)
                Tokachi.memory_write(AHBLAYER1_CTRL,0xA<<8|
                AES_LBIST_SCAN_ENABLE|
                AES_LBIST_CK_ENABLE)
                time.sleep(1)
                scan_pattern_cai =
                (int(Tokachi.memory_read(AHBLAYER2_DIAG21)
                ,16)>>20)&0xf
                #scan out CAI_AES results
#               scan_pattern_ref =
#(int(Tokachi.memory_read(AHBLAYER2_DIAG21),16)
>>16)&0xf
#scan out ref aes results
                if (scan_pattern_cai != 0xa):
                        print ’CAI_AES, scan out:’,
                        hex(scan_pattern_cai),’, FAIL!’
                        fail_flag = 1
#                       break
                if (scan_pattern_cai == 0xa):
                        print ’CAI_AES, scan out:’,
                         hex(scan_pattern_cai),’, PASS!’
                        Tokachi.memory_write
                        (AHBLAYER1_CTRL,AES_LBIST_SCAN_DISABLE)
                        list_all =
                        do_CAI_aes_lbist(vdd, freq,0,0)
                        print list_all
                        fail_flag = list_all[0]
                        itot = float(list_all[3])
                        lbistresult = int(list_all[4])
                        print ’lbistresult is:’, bin(lbistresult)
                if(not fail_flag):
                        print ’AES_LBIST Vmin is’, vdd,’V’,
                        freq,’kHz at Temperature:’, get_temp()
                        power_state = 0
                        turn_x_tti(pldcport, 2, power_state)
                        break
                if (fail_flag):
                        print ’AES_LBIST_FAILs at’,
                        vdd,’V’, freq,’kHz’
                        vdd += 0.01
                if (vdd>1.2):
                        power_state = 0
                        turn_x_tti(pldcport, 2, power_state)
                        break
                set_vdd_tti(pldcport,2, vdd)
        power_state = 0
        turn_x_tti(pldcport, 2, power_state)
```

C.12 AES Leakage power VS VDD measurement

```
def AES_leakage_v_sweep():
        init_vdd = 1.2
        final_vdd = 0.2
        vstep = -0.05
        freq = 1
        vesso= 0
        reset_dut()
        power_state = 1
        vdds = np.arange(init_vdd, final_vdd, vstep)
        x = np.zeros(len(vdds),
        dtype={'names':['vdd', 'freq', 'itot'],
        'formats':['f4','f4','f4']})
        x['vdd'] = vdds
############set supply voltage###############################
        set_supply('VBAT', 0)
        set_supply('VREGI', 0)
        set_supply('VESSI', 'VESSI_XT')
        #external supply for voltage
        set_supply('VRNG',1)
        #dualshiftreg is on VRNG power domain
############################################################
        m_u34410_conf(mrange='0.001', nplc='10', samples=100)
        for d in x:
                vdd = d['vdd']
                print "\n=============================\n"
                print "Test running at VDD = \n", vdd
                set_clk('CKAXI', 'PGFREQ2', freq)
                MbedTx('rstdut\n')
                Tokachi.get_prompt()
                set_vdd_tti(pldcport,2, vdd)
                turn_x_tti(pldcport, 2, power_state)
###########SCANIN-SCANOUT TEST###################################
                Tokachi.memory_write(AHBLAYER1_CTRL,0xA<<8)
                time.sleep(0.5)
                Tokachi.memory_write(AHBLAYER1_CTRL,0xA
                <<8|AES_LBIST_SCAN_ENABLE|AES_LBIST_CK_ENABLE)
                time.sleep(1)
                scan_pattern_cai =
                (int(Tokachi.memory_read(AHBLAYER2_DIAG21),16)
                >>20)&0xf
#               scan_pattern_ref =
#               (int(Tokachi.memory_read(AHBLAYER2_DIAG21),16)
>>16)&0xf
#scan out results
                if (scan_pattern_cai == 0xa):
                        print 'CAI_AES, scan out:',
                        hex(scan_pattern_cai),', PASS!'
                        Tokachi.memory_write
                        (AHBLAYER1_CTRL,AES_LBIST_SCAN_DISABLE)
                else:
                        print 'CAI_AES, scan out:',
                        hex(scan_pattern_cai),', FAIL!'
                        power_state = 0
                        turn_x_tti(pldcport, 2, power_state)
                        break
                list_all = do_CAI_aes_lbist(vdd, freq,0,0)
                print list_all
                fail_flag = list_all[0]
                itot = float(list_all[3])
                lbistresult = int(list_all[4])
                print 'lbistresult is:', bin(lbistresult)
                print 'leakage current is:', itot
                power = itot*vdd*1e6
                print 'leakage power is:', power, 'uW'
                if(not fail_flag):
                        x['itot'] = itot
                        x['freq'] = freq
                        Power_reg = vdd*itot
                if (fail_flag):
                        print 'AES_LBIST_FAILED!'
                        power_state = 0
                        turn_x_tti(pldcport, 2, power_state)
                        break
                if(itot>1e-5):
                        m_u34410_conf(mrange='0.01', samples=100)
                print  'VDD =', vdd,'Current =', itot,'at', freq, 'kHz.'
        turn_x_tti(pldcport, 2, 0)
        return x
```

References

[1] F. Xia, L. T. Yang, L. Wang, and A. Vinel, "Internet of things," *International Journal of Communication Systems*, vol. 25, no. 9, p. 1101, 2012.

[2] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future generation computer systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[3] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[4] K. L. Chang, J. S. Chang, B. H. Gwee, and K. S. Chong, "Synchronous-logic and asynchronous-logic 8051 microcontroller cores for realizing the internet of things: A comparative study on dynamic voltage scaling and variation effects," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 3, pp. 23–34, March 2013.

[5] P. Sparks, "The route to a trillion devices, the outlook for iot investment to 2035," *White paper, ARM*, 2017.

[6] Y. Greg, "Vlsi implications of the internet of things," June 2015.

[7] "Freescale shrinks worlds smallest arm-based mcu by an additional 15 percent." http://www.businesswire.com/news/home/20140225005317/en/Freescale-Shrinks-World%E2%80%99s-Smallest-ARM-Based-MCU-Additional. Accessed: 2014-02-25.

[8] Freescale Semiconductor, *KL03 Product Brief, Supports all KL03 devices*, 7 2014. Rev. 3.

[9] TEXAS INSTRUMENTS, *MSP430FR573x Mixed-Signal Microcontrollers*, 7 2011. Rev. APRIL 2016.

[10] STMicroelectronics, *STM32L431xx*, 6 2017. Rev. 2.

[11] C. Kidd, R. Orr, G. Abowd, C. Atkeson, I. Essa, B. MacIntyre, E. Mynatt, T. Starner, and W. Newstetter, "The aware home: A living laboratory for ubiquitous computing research," *Cooperative buildings. Integrating information, organizations, and architecture*, pp. 191–198, 1999.

[12] R. Johnstone, D. Caputo, U. Cella, A. Gandelli, C. Alippi, F. Grimaccia, N. Haritos, and R. E. Zich, "Smart environmental measurement & analysis technologies (semat): Wireless sensor networks in the marine environment," *Proceedings of the Wireless Sensor and Actuator Network Research on Opposite Sides of the Globe (SENSEI)*, 2008.

[13] J. M. Hernández-Muñoz, J. B. Vercher, L. Muñoz, J. A. Galache, M. Presser, L. A. H. Gómez, and J. Pettersson, "Smart cities at the forefront of the future internet," in *The Future Internet Assembly*, pp. 447–462, Springer, 2011.

[14] P. Kamalinejad, C. Mahapatra, Z. Sheng, S. Mirabbasi, V. C. Leung, and Y. L. Guan, "Wireless energy harvesting for the internet of things," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 102–108, 2015.

[15] O. Cetinkaya and O. B. Akan, "Electric-field energy harvesting from lighting elements for battery-less internet of things," *IEEE Access*, vol. 5, pp. 7423–7434, 2017.

[16] G. Lallement, F. Abouzeid, M. Cochet, J. Daveau, P. Roche, and J. Autran, "A 2.7 pj/cycle 16 mhz, 0.7μWdeep sleep power arm cortex-m0+ core soc in 28 nm fd-soi," *IEEE Journal of Solid-State Circuits*, vol. 53, pp. 2088–2100, July 2018.

[17] T. Kuroda, "cmos design challenges to power wall," in *Microprocesses and Nanotechnology Conference, 2001 International*, pp. 6–7, Oct 2001.

[18] R. 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, pp. 253–266, Feb 2010.

[19] T. Burd, T. Pering, A. Stratakos, and R. Brodersen, "A dynamic voltage scaled microprocessor system," *Solid-State Circuits, IEEE Journal of*, vol. 35, pp. 1571–1580, Nov 2000.

[20] N. H. Weste and D. M. Harris, *Integrated circuit design.* Pearson, 2011.

[21] D. Flynn, R. Aitken, A. Gibbons, and K. Shi, *Low power methodology manual: for system-on-chip design.* Springer Science & Business Media, 2007.

[22] P. Pillai and K. G. Shin, "Real-time dynamic voltage scaling for low-power embedded operating systems," in *ACM SIGOPS Operating Systems Review*, vol. 35, pp. 89–102, ACM, 2001.

[23] G. Semeraro, G. Magklis, R. Balasubramonian, D. H. Albonesi, S. Dwarkadas, and M. L. Scott, "Energy-efficient processor design using multiple clock domains with dynamic voltage and frequency scaling," in *High-Performance Computer Architecture, 2002. Proceedings. Eighth International Symposium on*, pp. 29–40, IEEE, 2002.

[24] A. Wang and A. Chandrakasan, "A 180-mv subthreshold fft processor using a minimum energy design methodology," *IEEE Journal of solid-state circuits*, vol. 40, no. 1, pp. 310–319, 2005.

[25] A. Wang, B. H. Calhoun, and A. P. Chandrakasan, *Sub-threshold design for ultra low-power systems*, vol. 95. Springer, 2006.

[26] B. H. Calhoun and A. P. Chandrakasan, "A 256-kb 65-nm sub-threshold sram design for ultra-low-voltage operation," *Solid-State Circuits, IEEE Journal of*, vol. 42, no. 3, pp. 680–688, 2007.

[27] S. Khare and S. Jain, "Prospects of near-threshold voltage design for green computing," in *2013 26th International Conference on VLSI Design and 2013 12th International Conference on Embedded Systems*, pp. 120–124, Jan 2013.

[28] B. Zhai, R. G. Dreslinski, D. Blaauw, T. Mudge, and D. Sylvester, "Energy efficient near-threshold chip multi-processing," in *Low Power Electronics and Design (ISLPED), 2007 ACM/IEEE International Symposium on*, pp. 32–37, Aug 2007.

[29] S. Jain, S. Khare, S. Yada, V. Ambili, P. Salihundam, S. Ramani, S. Muthukumar, M. Srinivasan, A. Kumar, S. K. Gb, R. Ramanarayanan, V. Erraguntla, J. Howard, S. Vangal, S. Dighe, G. Ruhl, P. Aseron, H. Wilson, N. Borkar, V. De, and S. Borkar, "A 280mv-to-1.2v wide-operating-range ia-32 processor in 32nm cmos," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2012 IEEE International*, pp. 66–68, Feb 2012.

[30] H. Kaul, M. Anders, S. Hsu, A. Agarwal, R. Krishnamurthy, and S. Borkar, "Near-threshold voltage (ntv) design: opportunities and challenges," in *Proceedings of the 49th Annual Design Automation Conference*, pp. 1153–1158, ACM, 2012.

[31] D. Markovic, C. Wang, L. Alarcon, T.-T. Liu, and J. Rabaey, "Ultralow-power design in near-threshold region," *Proceedings of the IEEE*, vol. 98, pp. 237–252, Feb 2010.

[32] R. Dreslinski, M. Wieckowski, D. S. Blaauw, and T. Mudge, "Near threshold computing: Overcoming performance degradation from aggressive voltage scaling," in *Proc. Workshop Energy-Efficient Design*, pp. 44–49, 2009.

[33] U. R. Karpuzcu, K. B. Kolluru, N. S. Kim, and J. Torrellas, "Varius-ntv: A microarchitectural model to capture the increased sensitivity of manycores to process variations at near-threshold voltages," in *Dependable Systems and Networks (DSN), 2012 42nd Annual IEEE/IFIP International Conference on*, pp. 1–11, IEEE, 2012.

[34] U. R. Karpuzcu, N. S. Kim, and J. Torrellas, "Coping with parametric variation at near-threshold voltages," *IEEE Micro*, vol. 33, pp. 6–14, July 2013.

[35] L. Kou, *Impact of process variations on soft error sensitivity of 32-nm VLSI circuits in near-threshold region.* PhD thesis, Vanderbilt University, 2014.

[36] M. Seok, G. Chen, S. Hanson, M. Wieckowski, D. Blaauw, and D. Sylvester, "Cas-fest 2010: Mitigating variability in near-threshold computing," *Emerging and Selected Topics in Circuits and Systems, IEEE Journal on*, vol. 1, no. 1, pp. 42–49, 2011.

[37] N. H. Weste and D. Harris, *CMOS VLSI design: a circuits and systems perspective.* Pearson Education India, 2015.

[38] 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, pp. 305–327, Feb 2003.

[39] A. Keshavarzi, S. Ma, S. Narendra, B. Bloechel, K. Mistry, T. Ghani, S. Borkar, and V. De, "Effectiveness of reverse body bias for leakage control in scaled dual vt cmos ics," in *Proceedings of the 2001 international symposium on Low power electronics and design*, pp. 207–212, ACM, 2001.

[40] S. Hareland, A. Tasch, and C. Maziar, "New structural approach for reducing punchthrough current in deep submicrometre mosfets and extending mosfet scaling," *Electronics Letters*, vol. 29, no. 21, pp. 1894–1896, 1993.

[41] V. Kursun and E. G. Friedman, "Variable threshold voltage keeper for contention reduction in dynamic circuits," in *ASIC/SOC Conference, 2002. 15th Annual IEEE International*, pp. 314–318, IEEE, 2002.

[42] J. Kao, A. Chandrakasan, and D. Antoniadis, "Transistor sizing issues and tool for multi-threshold cmos technology," in *Proceedings of the 34th annual Design Automation Conference*, pp. 409–414, ACM, 1997.

[43] Z. Hu, A. Buyuktosunoglu, V. Srinivasan, V. Zyuban, H. Jacobson, and P. Bose, "Microarchitectural techniques for power gating of execution units," in *Proceedings of the 2004 international symposium on Low power electronics and design*, pp. 32–37, ACM, 2004.

[44] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De, "Parameter variations and impact on circuits and microarchitecture," in *Proceedings of the 40th annual Design Automation Conference*, pp. 338–342, ACM, 2003.

[45] S. Hanson, B. Zhai, K. Bernstein, D. Blaauw, A. Bryant, L. Chang, K. K. Das, W. Haensch, E. J. Nowak, and D. M. Sylvester, "Ultralow-voltage, minimum-energy cmos," *IBM journal of research and development*, vol. 50, no. 4.5, pp. 469–490, 2006.

[46] K. Flautner, S. Reinhardt, and T. Mudge, “Automatic performance setting for dynamic voltage scaling,” in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, MobiCom ’01, (New York, NY, USA), pp. 260–271, ACM, 2001.

[47] K. Flautner and T. Mudge, “Vertigo: Automatic performance-setting for linux,” in *Proceedings of the 5th Symposium on Operating Systems Design and implementationCopyright Restrictions Prevent ACM from Being Able to Make the PDFs for This Conference Available for Downloading*, OSDI ’02, (Berkeley, CA, USA), pp. 105–116, USENIX Association, 2002.

[48] M. T. Schmitz, B. M. Al-Hashimi, and P. Eles, “Energy-efficient mapping and scheduling for dvs enabled distributed embedded systems,” in *Proceedings 2002 Design, Automation and Test in Europe Conference and Exhibition*, pp. 514–521, 2002.

[49] B. Zhai, D. Blaauw, D. Sylvester, and K. Flautner, “Theoretical and practical limits of dynamic voltage scaling,” in *Proceedings. 41st Design Automation Conference, 2004.*, pp. 868–873, July 2004.

[50] “Transmeta Crusoe.” http://www.transmeta.com/.

[51] “Intel Xscale.” http://www.intel.com/design/intelxscale/.

[52] “IBM PowerPC.” http://www.chips.ibm.com/products/powerpc/.

[53] A. Dancy and A. Chandrakasan, “Techniques for aggressive supply voltage scaling and efficient regulation [cmos digital circuits],” in *Custom Integrated Circuits Conference, 1997., Proceedings of the IEEE 1997*, pp. 579–586, IEEE, 1997.

[54] L. Shang, L.-S. Peh, and N. K. Jha, “Dynamic voltage scaling with links for power optimization of interconnection networks,” in *High-Performance Computer Architecture, 2003. HPCA-9 2003. Proceedings. The Ninth International Symposium on*, pp. 91–102, IEEE, 2003.

[55] J. Rabaey, *Low power design essentials*. Springer Science & Business Media, 2009.

[56] B. H. Calhoun, A. Wang, and A. Chandrakasan, “Modeling and sizing for minimum energy operation in subthreshold circuits,” *Solid-State Circuits, IEEE Journal of*, vol. 40, no. 9, pp. 1778–1786, 2005.

[57] R. Dreslinski Jr, *Near threshold computing: From single core to many-core energy efficient architectures*. PhD thesis, The University of Michigan, 2011.

[58] H. Kaul, M. A. Anders, S. K. Mathew, S. K. Hsu, A. Agarwal, R. K. Krishnamurthy, and S. Borkar, “A 320 mv 56 μw 411 gops/watt ultra-low voltage motion estimation accelerator in 65 nm cmos,” *Solid-State Circuits, IEEE Journal of*, vol. 44, no. 1, pp. 107–114, 2009.

[59] A. Kauppila, B. Bhuva, J. Kauppila, L. Massengill, and W. Holman, “Impact of process variations on sram single event upsets,” *Nuclear Science, IEEE Transactions on*, vol. 58, pp. 834–839, June 2011.

[60] Q. Ding, R. Luo, and Y. Xie, “Impact of process variation on soft error vulnerability for nanometer vlsi circuits,” in *ASIC, 2005. ASICON 2005. 6th International Conference On*, vol. 2, pp. 1117–1121, Oct 2005.

[61] N. Pinckney, D. Blaauw, and D. Sylvester, “Low-power near-threshold design: Techniques to improve energy efficiency energy-efficient near-threshold design has been proposed to increase energy efficiency across a wid,” *IEEE Solid-State Circuits Magazine*, vol. 7, pp. 49–57, Spring 2015.

[62] M. Agostinelli, J. Hicks, J. Xu, B. Woolery, K. Mistry, K. Zhang, S. Jacobs, J. Jopling, W. Yang, B. Lee, T. Raz, M. Mehalel, P. Kolar, Y. Wang, J. Sandford, D. Pivin, C. Peterson, M. DiBattista, S. Pae, M. Jones, S. Johnson, and G. Subramanian, “Erratic fluctuations of sram cache vmin at the 90nm process technology node,” in *IEEE InternationalElectron Devices Meeting, 2005. IEDM Technical Digest.*, pp. 655–658, Dec 2005.

[63] T. Pompl, C. Engel, H. Wurzer, and M. Kerber, “Soft breakdown and hard breakdown in ultra-thin oxides,” *Microelectronics reliability*, vol. 41, no. 4, pp. 543–551, 2001.

[64] R. Degraeve, G. Groeseneken, R. Bellens, M. Depas, and H. E. Maes, “A consistent model for the thickness dependence of intrinsic breakdown in ultra-thin oxides,” in *Electron Devices Meeting, 1995. IEDM ’95., International*, pp. 863–866, Dec 1995.

[65] Y. Kim, W. Jung, I. Lee, Q. Dong, M. Henry, D. Sylvester, and D. Blaauw, “27.8 a static contention-free single-phase-clocked 24t flip-flop in 45nm for low-power applications,” in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International*, pp. 466–467, IEEE, 2014.

[66] C. H. Chen, K. Bowman, C. Augustine, Z. Zhang, and J. Tschanz, “Minimum supply voltage for sequential logic circuits in a 22nm technology,” in *Low Power Electronics and Design (ISLPED), 2013 IEEE International Symposium on*, pp. 181–186, Sept 2013.

[67] L. Chang, Y. Nakamura, R. Montoye, J. Sawada, A. Martin, K. Kinoshita, F. Gebara, K. Agarwal, D. Acharyya, W. Haensch, K. Hosokawa, and D. Jamsek, “A 5.3ghz 8t-sram with operation down to 0.41v in 65nm cmos,” in *VLSI Circuits, 2007 IEEE Symposium on*, pp. 252–253, June 2007.

[68] B. Calhoun and A. Chandrakasan, “A 256kb sub-threshold sram in 65nm cmos,” in *Solid-State Circuits Conference, 2006. ISSCC 2006. Digest of Technical Papers. IEEE International*, pp. 2592–2601, Feb 2006.

[69] V. Joshi, D. Blaauw, and D. Sylvester, "Soft-edge flip-flops for improved timing yield: design and optimization," in *Computer-Aided Design, 2007. ICCAD 2007. IEEE/ACM International Conference on*, pp. 667–673, Nov 2007.

[70] M. Alioto, E. Consoli, and G. Palumbo, *Flip-Flop Design in Nanometer CMOS*. Springer, 2016.

[71] M. Alioto, E. Consoli, and G. Palumbo, "Variations in nanometer cmos flip-flops: Part i ;impact of process variations on timing," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, pp. 2035–2043, Aug 2015.

[72] M. Alioto, E. Consoli, and G. Palumbo, "Variations in Nanometer CMOS Flip-Flops: Impact of Process Variations on Timing," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, pp. 2035–2043, Aug 2015.

[73] W. P. Burleson, M. Ciesielski, F. Klass, and W. Liu, "Wave-pipelining: a tutorial and research survey," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 6, pp. 464–474, Sept 1998.

[74] M. LaPedus, "Isscc: Toshiba rolls flip-flop," February 2011. [Online; posted 20-Februrary-2011].

[75] M. Alioto, E. Consoli, and G. Palumbo, "Flip-flop energy/performance versus clock slope and impact on the clock network design," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, pp. 1273–1286, June 2010.

[76] J. M. Rabaey, A. P. Chandrakasan, and B. Nikolic, *Digital integrated circuits*, vol. 2. Prentice hall Englewood Cliffs, 2002.

[77] A. Components, "Tsmc 0.18 um process 1.8-volt sage-x standard cell library databook," 2002.

[78] G. Gerosa, S. Gary, C. Dietz, D. Pham, K. Hoover, J. Alvarez, H. Sanchez, P. Ippolito, T. Ngo, S. Litch, J. Eno, J. Golab, N. Vanderschaaf, and J. Kahle, "A 2.2 w, 80 mhz superscalar risc microprocessor," *IEEE Journal of Solid-State Circuits*, vol. 29, pp. 1440–1454, Dec 1994.

[79] M. Zwolinski, *Digital system design with SystemVerilog*. Pearson Education, 2009.

[80] M. Alioto, E. Consoli, and G. Palumbo, "Comparative analysis of the robustness of master-slave flip-flops against variations," in *Electronics, Circuits, and Systems (ICECS), 2015 IEEE International Conference on*, pp. 117–120, IEEE, 2015.

[81] D. Markovic, J. W. Tschanz, and V. K. De, "Transmission-gate based flip-flop," Nov. 4 2003. US Patent 6,642,765.

[82] D. Markovic, B. Nikolic, and R. Brodersen, “Analysis and design of low-energy flip-flops,” in *Proceedings of the 2001 international symposium on Low power electronics and design*, pp. 52–55, ACM, 2001.

[83] M. Nogawa and Y. Ohtomo, “A data-transition look-ahead dff circuit for statistical reduction in power consumption,” *IEEE Journal of Solid-State Circuits*, vol. 33, no. 5, pp. 702–706, 1998.

[84] V. Stojanovic and V. G. Oklobdzija, “Comparative analysis of master-slave latches and flip-flops for high-performance and low-power systems,” *IEEE Journal of Solid-State Circuits*, vol. 34, pp. 536–548, Apr 1999.

[85] C. Foley, “Characterizing metastability,” in *Proceedings Second International Symposium on Advanced Research in Asynchronous Circuits and Systems*, pp. 175–184, Mar 1996.

[86] R. Llopis and M. Sachdev, “Low power, testable dual edge triggered flip-flops,” in *Proceedings of the 1996 international symposium on Low power electronics and design*, pp. 341–345, IEEE Press, 1996.

[87] B. Nikolic, V. G. Oklobdzija, V. Stojanovic, W. Jia, J. K.-S. Chiu, and M. M.-T. Leung, “Improved sense-amplifier-based flip-flop: Design and measurements,” *IEEE Journal of Solid-State Circuits*, vol. 35, no. 6, pp. 876–884, 2000.

[88] F. Klass, C. Amir, A. Das, K. Aingaran, C. Truong, R. Wang, A. Mehta, R. Heald, and G. Yee, “A new family of semidynamic and dynamic flip-flops with embedded logic for high-performance processors,” *IEEE Journal of Solid-State Circuits*, vol. 34, no. 5, pp. 712–716, 1999.

[89] J. Yuan and C. Svensson, “High-speed cmos circuit technique,” *Solid-State Circuits, IEEE Journal of*, vol. 24, pp. 62–70, Feb 1989.

[90] M. Alioto, E. Consoli, and G. Palumbo, “Det ff topologies: A detailed investigation in the energy-delay-area domain,” in *2011 IEEE International Symposium of Circuits and Systems (ISCAS)*, pp. 563–566, May 2011.

[91] W. Chung, T. Lo, and M. Sachdev, “A comparative analysis of low-power low-voltage dual-edge-triggered flip-flops,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 10, pp. 913–918, Dec 2002.

[92] M. Alioto, E. Consoli, and G. Palumbo, “Analysis and comparison in the energy-delay-area domain of nanometer cmos flip-flops: Part ii ;results and figures of merit,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, pp. 737–750, May 2011.

[93] J. Tschanz, S. Narendra, Z. Chen, S. Borkar, M. Sachdev, and V. De, “Comparative delay and energy of single edge-triggered & dual edge-triggered pulsed

flip-flops for high-performance microprocessors," in *Proceedings of the 2001 international symposium on Low power electronics and design*, pp. 147–152, ACM, 2001.

[94] P. Zhao, T. K. Darwish, and M. A. Bayoumi, "High-performance and low-power conditional discharge flip-flop," *IEEE transactions on very large scale integration (VLSI) systems*, vol. 12, no. 5, pp. 477–484, 2004.

[95] B. Nikolic, V. Stojanovic, V. G. Oklobdzija, W. Jia, J. Chiu, and M. Leung, "Sense amplifier-based flip-flop," in *Solid-State Circuits Conference, 1999. Digest of Technical Papers. ISSCC. 1999 IEEE International*, pp. 282–283, IEEE, 1999.

[96] N. Nedovic, V. G. Oklobdzija, and W. W. Walker, "A clock skew absorbing flip-flop," in *2003 IEEE International Solid-State Circuits Conference, 2003. Digest of Technical Papers. ISSCC.*, pp. 342–497 vol.1, Feb 2003.

[97] B.-S. Kong, S.-S. Kim, and Y.-H. Jun, "Conditional-capture flip-flop for statistical power reduction," *IEEE Journal of Solid-State Circuits*, vol. 36, pp. 1263–1271, Aug 2001.

[98] S. D. Shin and B. S. Kong, "Variable sampling window flip-flops for low-power high-speed vlsi," *IEE Proceedings - Circuits, Devices and Systems*, vol. 152, pp. 266–271, June 2005.

[99] P. E. Gronowski, W. J. Bowhill, R. P. Preston, M. K. Gowan, and R. L. Allmon, "High-performance microprocessor design," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 5, pp. 676–686, 1998.

[100] J. Montanaro, R. T. Witek, K. Anne, A. J. Black, E. M. Cooper, D. W. Dobberpuhl, P. M. Donahue, J. Eno, W. Hoeppner, D. Kruckemyer, *et al.*, "A 160-mhz, 32-b, 0.5-w cmos risc microprocessor," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 11, pp. 1703–1714, 1996.

[101] D. Draper, M. Crowley, J. Holst, G. Favor, A. Schoy, J. Trull, A. Ben-Meir, R. Khanna, D. Wendell, R. Krishna, *et al.*, "Circuit techniques in a 266-mhz mmx-enabled processor," *IEEE Journal of Solid-State Circuits*, vol. 32, no. 11, pp. 1650–1664, 1997.

[102] H. Partovi, R. Burd, U. Salim, F. Weber, L. DiGregorio, and D. Draper, "Flow-through latch and edge-triggered flip-flop hybrid elements," in *1996 IEEE International Solid-State Circuits Conference. Digest of TEchnical Papers, ISSCC*, pp. 138–139, Feb 1996.

[103] R. Heald, K. Aingaran, C. Amir, M. Ang, M. Boland, P. Dixit, G. Gouldsberry, D. Greenley, J. Grinberg, J. Hart, *et al.*, "A third-generation sparc v9 64-b microprocessor," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 11, pp. 1526–1538, 2000.

[104] N. Nedovic and V. G. Oklobdzija, "Hybrid latch flip-flop with improved power efficiency," in *Proceedings 13th Symposium on Integrated Circuits and Systems Design (Cat. No.PR00843)*, pp. 211–215, 2000.

[105] N. Nedovic and V. G. Oklobdzija, "Dynamic flip-flop with improved power," in *Proceedings of the 26th European Solid-State Circuits Conference*, pp. 376–379, Sept 2000.

[106] D. Markovic, B. Nikolic, and R. Brodersen, "Analysis and design of low-energy flip-flops," in *Proceedings of the 2001 International Symposium on Low Power Electronics and Design*, ISLPED '01, (New York, NY, USA), pp. 52–55, ACM, 2001.

[107] P. Larsson and C. Svensson, "Impact of clock slope on true single phase clocked (tspc) cmos circuits," *IEEE Journal of Solid-State Circuits*, vol. 29, pp. 723–726, Jun 1994.

[108] A. Hirata, K. Nakanishi, M. Nozoe, and A. Miyoshi, "The cross charge-control flip-flop: A low-power and high-speed flip-flop suitable for mobile application socs," in *VLSI Circuits, 2005. Digest of Technical Papers. 2005 Symposium on*, pp. 306–307, IEEE, 2005.

[109] C. K. Teh, T. Fujita, H. Hara, and M. Hamada, "A 77% energy-saving 22-transistor single-phase-clocking d-flip-flop with adaptive-coupling configuration in 40nm cmos," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*, pp. 338–340, Feb 2011.

[110] N. Kawai, S. Takayama, J. Masumi, N. Kikuchi, Y. Itoh, K. Ogawa, A. Ugawa, H. Suzuki, and Y. Tanaka, "A fully static topologically-compressed 21-transistor flip-flop with 75% power saving," *Solid-State Circuits, IEEE Journal of*, vol. 49, no. 11, pp. 2526–2533, 2014.

[111] F. Stas and D. Bol, "A 0.4-v 0.66-fJ/cycle Retentive True-Single-Phase-Clock 18T Flip-Flop in 28-nm Fully-Depleted SOI CMOS," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, pp. 935–945, March 2018.

[112] W. Zhao, F. Liu, K. Agarwal, D. Acharyya, S. R. Nassif, K. J. Nowka, and Y. Cao, "Rigorous extraction of process variations for 65-nm cmos design," *IEEE Transactions on Semiconductor Manufacturing*, vol. 22, pp. 196–203, Feb 2009.

[113] K. Agarwal and S. Nassif, "Characterizing process variation in nanometer cmos," in *Design Automation Conference, 2007. DAC'07. 44th ACM/IEEE*, pp. 396–399, IEEE, 2007.

[114] L. R. Harriott, "Limits of lithography," *Proceedings of the IEEE*, vol. 89, no. 3, pp. 366–374, 2001.

[115] K. A. Bowman, S. G. Duvall, and J. D. Meindl, “Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration,” *Solid-State Circuits, IEEE Journal of*, vol. 37, no. 2, pp. 183–190, 2002.

[116] M. Ghahroodi, *Variation and reliability in digital CMOS circuit design.* PhD thesis, University of Southampton, 2014.

[117] L. He, A. Kahng, K. H. Tam, and J. Xiong, “Simultaneous buffer insertion and wire sizing considering systematic cmp variation and random leff variation,” *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 26, pp. 845–857, May 2007.

[118] K. Bernstein, D. J. Frank, A. E. Gattiker, W. Haensch, B. L. Ji, S. R. Nassif, E. J. Nowak, D. J. Pearson, and N. J. Rohrer, “High-performance cmos variability in the 65-nm regime and beyond,” *IBM journal of research and development*, vol. 50, no. 4.5, pp. 433–449, 2006.

[119] F. Sexton, “Destructive single-event effects in semiconductor devices and ics,” *Nuclear Science, IEEE Transactions on*, vol. 50, pp. 603–621, June 2003.

[120] J. S. JESD89A, “Measurement and reporting of alpha particle and terrestrial cosmic ray-induced soft errors in semiconductor devices,” *JEDEC solid state technology association*, 2006.

[121] R. Gaillard, “Single event effects: Mechanisms and classification,” in *Soft Errors in Modern Electronic Systems*, pp. 27–54, Springer, 2011.

[122] R. Leach, “Spacecraft system failures and anomalies attributed to the natural space environment,” in *Space Programs and Technologies Conference*, p. 3564, 1995.

[123] T. C. May and M. H. Woods, “A new physical mechanism for soft errors in dynamic memories,” in *Reliability Physics Symposium, 1978. 16th Annual*, pp. 33–40, April 1978.

[124] R. Baumann, “Radiation-induced soft errors in advanced semiconductor technologies,” *Device and Materials Reliability, IEEE Transactions on*, vol. 5, pp. 305–316, Sept 2005.

[125] G. Messenger, “Collection of charge on junction nodes from ion tracks,” *Nuclear Science, IEEE Transactions on*, vol. 29, pp. 2024–2031, Dec 1982.

[126] G. Srinivasan, H. Tang, and P. Murley, “Parameter-free, predictive modeling of single event upsets due to protons, neutrons, and pions in terrestrial cosmic rays,” *Nuclear Science, IEEE Transactions on*, vol. 41, no. 6, pp. 2063–2070, 1994.

[127] T. Heijmen and B. Kruseman, "Alpha-particle-induced ser of embedded srams affected by variations in process parameters and by the use of process options," *Solid-state electronics*, vol. 49, no. 11, pp. 1783–1790, 2005.

[128] N. Seifert, X. Zhu, D. Moyer, R. Mueller, R. Hokinson, N. Leland, M. Shade, and L. Massengill, "Frequency dependence of soft error rates for sub-micron cmos technologies," in *Electron Devices Meeting, 2001. IEDM'01. Technical Digest. International*, pp. 14–4, IEEE, 2001.

[129] R. Ramanarayanan, V. Degalahal, N. Vijaykrishnan, M. Irwin, and D. Duarte, "Analysis of soft error rate in flip-flops and scannable latches," in *SOC Conference, 2003. Proceedings. IEEE International [Systems-on-Chip]*, pp. 231–234, IEEE, 2003.

[130] P. Hazucha and C. Svensson, "Impact of cmos technology scaling on the atmospheric neutron soft error rate," *Nuclear Science, IEEE Transactions on*, vol. 47, no. 6, pp. 2586–2594, 2000.

[131] T. Karnik and P. Hazucha, "Characterization of soft errors caused by single event upsets in cmos processes," *Dependable and Secure Computing, IEEE Transactions on*, vol. 1, no. 2, pp. 128–143, 2004.

[132] T. Kehl, "Hardware self-tuning and circuit performance monitoring," in *Computer Design: VLSI in Computers and Processors, 1993. ICCD '93. Proceedings., 1993 IEEE International Conference on*, pp. 188–192, Oct 1993.

[133] L. Entrena, C. Lopez, and E. Olias, "Automatic insertion of fault-tolerant structures at the rt level," in *On-Line Testing Workshop, 2001. Proceedings. Seventh International*, pp. 48–50, 2001.

[134] E. Ibe, H. Taniguchi, Y. Yahagi, K. i. Shimbo, and T. Toba, "Impact of scaling on neutron-induced soft error in srams from a 250 nm to a 22 nm design rule," *IEEE Transactions on Electron Devices*, vol. 57, pp. 1527–1538, July 2010.

[135] M. Ghahroodi, M. Zwolinski, and E. Ozer, "Radiation hardening by design: A novel gate level approach," in *Adaptive Hardware and Systems (AHS), 2011 NASA/ESA Conference on*, pp. 74–79, June 2011.

[136] M. Ghahroodi, M. Zwolinski, and E. Ozer, "Radiation hardening by design: A novel gate level approach," in *Adaptive Hardware and Systems (AHS), 2011 NASA/ESA Conference on*, pp. 74–79, June 2011.

[137] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, "Razor: a low-power pipeline based on circuit-level timing speculation," in *Microarchitecture, 2003. MICRO-36. Proceedings. 36th Annual IEEE/ACM International Symposium on*, pp. 7–18, Dec 2003.

[138] S. Das, C. Tokunaga, S. Pant, W.-H. Ma, S. Kalaiselvan, K. Lai, D. Bull, and D. Blaauw, "Razorii: In situ error detection and correction for pvt and ser tolerance," *Solid-State Circuits, IEEE Journal of*, vol. 44, pp. 32–48, Jan 2009.

[139] M. Fojtik, D. Fick, Y. Kim, N. Pinckney, D. Harris, D. Blaauw, and D. Sylvester, "Bubble razor: An architecture-independent approach to timing-error detection and correction," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2012 IEEE International*, pp. 488–490, Feb 2012.

[140] 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, pp. 619–631, Feb 2018.

[141] S. Das, *Razor: A variability-tolerant design methodology for low-power and robust computing.* ProQuest, 2009.

[142] J. Myers, A. Savanth, R. Gaddh, D. Howard, P. Prabhat, and D. Flynn, "A Subthreshold ARM Cortex-M0+ Subsystem in 65 nm CMOS for WSN Applications with 14 Power Domains, 10T SRAM, and Integrated Voltage Regulator," *IEEE Journal of Solid-State Circuits*, vol. 51, pp. 31–44, Jan 2016.

[143] T. Fischer, S. Arekapudi, E. Busta, C. Dietz, M. Golden, S. Hilker, A. Horiuchi, K. A. Hurd, D. Johnson, H. McIntyre, *et al.*, "Design solutions for the bulldozer 32nm soi 2-core processor module in an 8-core cpu," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*, pp. 78–80, IEEE, 2011.

[144] V. Stojanovic and V. G. Oklobdzija, "Comparative analysis of master-slave latches and flip-flops for high-performance and low-power systems," *IEEE J. Solid-State Circuits*, vol. 34, no. 4, pp. 536–548, 1999.

[145] G. Yeric, B. Cline, S. Sinha, D. Pietromonaco, V. Chandra, and R. Aitken, "The past present and future of design-technology co-optimization," in *Proceedings of the IEEE 2013 Custom Integrated Circuits Conference*, pp. 1–8, Sept 2013.

[146] N. Pinckney, D. Blaauw, and D. Sylvester, "Low-power near-threshold design: Techniques to improve energy efficiency energy-efficient near-threshold design has been proposed to increase energy efficiency across a wid," *IEEE Solid-State Circuits Magazine*, vol. 7, pp. 49–57, Spring 2015.

[147] N. Pinckney, D. Blaauw, and D. Sylvester, "Low-Power Near-Threshold Design: Techniques to Improve Energy Efficiency," *IEEE Solid-State Circuits Mag.*, vol. 7, pp. 49–57, Spring 2015.

[148] V. Kursun and E. G. Friedman, "Variable threshold voltage keeper for contention reduction in dynamic circuits," in *ASIC/SOC Conference, 2002. 15th Annual IEEE International*, pp. 314–318, IEEE, 2002.

[149] M. R. Jan, C. Anantha, and N. Borivoje, *Digital Integrated Circuits–A Design Perspective*. Prentice-Hall, 2002.

[150] M. Alioto, E. Consoli, and G. Palumbo, "General strategies to design nanometer flip-flops in the energy-delay space," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, pp. 1583–1596, July 2010.

[151] Y. T. Shyu, J. M. Lin, C. P. Huang, C. W. Lin, Y. Z. Lin, and S. J. Chang, "Effective and efficient approach for power reduction by using multi-bit flip-flops," *IEEE Trans. Very Large Scale Integration (VLSI) Syst*, vol. 21, pp. 624–635, April 2013.

[152] "Arm, Cortex-M0 Revision: r0p0 Technical Reference Manual." `http://www.arm.com/products/processors/cortex-m/cortex-m0.php`. Accessed: 2017-09-30.

[153] L. Zeng and P. Beckett, "Soft error rate estimation in deep sub-micron cmos," in *Dependable Computing, 2007. PRDC 2007. 13th Pacific Rim International Symposium on*, pp. 210–216, IEEE, 2007.

[154] R. Naseer, Y. Boulghassoul, J. Draper, S. DasGupta, and A. Witulski, "Critical charge characterization for soft error rate modeling in 90nm sram," in *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, pp. 1879–1882, IEEE, 2007.

[155] W. Van Gils, "A triple modular redundancy technique providing multiple-bit error protection without using extra redundancy," *Computers, IEEE Transactions on*, vol. C-35, pp. 623–631, July 1986.

[156] Y. Choi, Y.-B. Kim, and F. Lombardi, "Soft error masking latch for sub-threshold voltage operation," in *Circuits and Systems (MWSCAS), 2012 IEEE 55th International Midwest Symposium on*, pp. 25–28, Aug 2012.

[157] Y. Lin and M. Zwolinski, "Settoff: A fault tolerant flip-flop for building cost-efficient reliable systems," in *On-Line Testing Symposium (IOLTS), 2012 IEEE 18th International*, pp. 7–12, June 2012.

[158] S. S. Mishra, A. K. Agrawal, and R. Nagaria, "A comparative performance analysis of various cmos design techniques for xor and xnor circuits," *International Journal on Emerging Technologies*, vol. 1, no. 1, pp. 1–10, 2010.