

2010-01-01

# A Subthreshold Reconfigurable Architecture For Harsh Environments

Ameet O. Chavan

University of Texas at El Paso, [oachavan@miners.utep.edu](mailto:oachavan@miners.utep.edu)

Follow this and additional works at: [https://digitalcommons.utep.edu/open\\_etd](https://digitalcommons.utep.edu/open_etd)



Part of the [Electrical and Electronics Commons](#)

---

## Recommended Citation

Chavan, Ameet O., "A Subthreshold Reconfigurable Architecture For Harsh Environments" (2010). *Open Access Theses & Dissertations*. 1599.

[https://digitalcommons.utep.edu/open\\_etd/1599](https://digitalcommons.utep.edu/open_etd/1599)

This is brought to you for free and open access by DigitalCommons@UTEP. It has been accepted for inclusion in Open Access Theses & Dissertations by an authorized administrator of DigitalCommons@UTEP. For more information, please contact [lweber@utep.edu](mailto:lweber@utep.edu).

# A SUBTHRESHOLD RECONFIGURABLE ARCHITECTURE FOR HARSH ENVIRONMENTS

AMEET CHAVAN

Department of Electrical and Computer Engineering

APPROVED:

---

Eric MacDonald, Ph.D., Chair

---

Patricia A. Nava, Ph.D., P.E

---

David Zubia, Ph.D.

---

Patricia J. Teller, Ph.D.

---

Patricia D. Witherspoon, Ph.D.

Dean of Graduate School

Copyright © 2010 Ameet Chavan

All rights reserved

## **Dedication**

I dedicate this dissertation to my parents, Omprakash Chavan, father, and Chaya Chavan, mother, and to my wife, Shilpa Chavan.



# A SUBTHRESHOLD RECONFIGURABLE ARCHITECTURE FOR HARSH ENVIRONMENTS

By

AMEET CHAVAN, M.S (E.E)

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

Department of Electrical and Computer Engineering  
THE UNIVERSITY OF TEXAS AT EL PASO

May 2010

## **Acknowledgements**

The dissertation work would not have been possible without the support and guidance of many people. First I would like to express deepest gratitude to my mentor Dr. Eric MacDonald, without whom the dissertation work would not have been possible. During the course of the research work Dr. MacDonald has been an excellent source of motivation, and a reliable reference, who guided me ever path of the way. Because of his critical role, the research work has reached its final glory of being fabricated at MIT's Lincoln Lab, sponsored by DARPA.

I would like to thank Dr. Patricia Nava for providing the initial motivation to pursue the doctoral program and there after providing valuable support to persist through the program. Special thanks to my dissertation committee members Dr. Patricia Teller and Dr. David Zubia for providing constructive feedback.

With respect to the research experiments I want to acknowledge system support and tools that permitted me to accomplish my work, which are: Virgo2 Cluster – Distributed Computing Lab (DCL) at UTEP, MOSIS for providing 45nm IBM12SOI models, University of Florida for UFSOI models, UTEP Graduate School Research Grant for equipment procurement, and finally, DARPA for sponsoring chip fabrication at MITLL.

## **Abstract**

Energy harvesting and functional reconfigurability are necessary features in order to simultaneously achieve longer operating lifetimes and versatility in application for many next generation electronic systems. The presented research incorporates capabilities that not only enable applications to self-power from ambience but also permit change in functionality based on real-time application requirements. Currently, many applications are battery powered with custom hardware, which severely confines the application platform. Moreover, maintenance and upgrades are prohibitively expensive, particularly in the case of remote locations with limited accessibility. For harsh environments like Space or the battlefield, apart from features such as low power and reconfigurability, robust circuit design is also essential to withstand the effects of radiation and noise. The proposed research implements the Subthreshold Radhard Reconfigurable (SRR) architecture that collectively addresses these three design requirements:

- 1) radiation hardness with Single Event Upset (SEU) immune circuits,
- 2) self-powering with subthreshold voltage operation, and
- 3) functional adaptability with a programmable processing element.

The proposed architecture is a unique solution – well suited for the next generation systems that can simultaneously harvest energy from ambience, reconfigure based on immediate application requirements, and provide robust operation even in harsh environments.

In the course of the SRR design, that combined reconfigurability, subthreshold and robust operation novel subthreshold optimized circuits were proposed and evaluated for the first time. The proposed circuits include: (1) a Modified Sense Amplifier C<sup>2</sup>MOS Flip-Flop (MOSAC FF), which enables SEU/SET (Single Event Transient) resilience with 50% reduced power and 20%

better SET critical charge ( $Q_{crit}$ ) values compared to the best existing design, (2) Body-Tied Level Shifter, which provides 13% increased performance, and (3) Charge Pump, which provides 15% increased output voltage gain at subthreshold voltages when compared to existing designs. These proposed novel circuits, in a design competition sponsored by DARPA, were awarded fabrication at Massachusetts Institute of Technology's Lincoln Laboratory (MITLL), which provides the first ever CMOS technology specifically optimized for subthreshold operation.

The functional verification of the proposed architecture is performed by configuring the design as a 16-bit Multiply Accumulate (MAC) unit using 45nm IBM-12SOI fully depleted transistor model in HSPICE. For comparison purposes, two designs were considered: (1) the Application Specific MAC (ASMAC) design employing adaptive body-bias supply and (2) an off-the-shelf commercial Xilinx Field Programmable Gate Array (FPGA) configured to function as 16-bit MAC. The ASMAC design reported better power consumption values than the proposed design but lacked application flexibility and design robustness. Similarly, in comparison to the proposed design, the commercial FPGA provided application flexibility and substantial performance, at the expense of several orders of increased power consumption. Furthermore, the FPGA implementation lacked radiation hardness and depended on a dedicated traditional power supply. The proposed architecture was not as power efficient as the ASMAC; however, the slight increase in power is a necessary trade-off when considering the additional features provided, which include robustness, ultra-low power operation, and application adaptability. Consequently the proposed design is apt for applications requiring self-powering, versatile application, and operation in harsh environments.

## Table of Contents

	Page
List of Figures .....	xi
List of Tables .....	xiv
Chapter 1 Introduction	
1.1 Motivation.....	1
1.2 Research Objective .....	2
1.3 Research Approach and Dissertation Organization .....	3
Chapter2 Background	
2.1 Low Power Design.....	5
2.1.1 Traditional Low Power Design Techniques .....	6
2.1.2 Advanced Low Power Design Technique .....	8
2.2 Subthreshold Logic Operation .....	9
2.3 Radiation Hardening and Single Event Upsets (SEUs) .....	11
2.3.1 Fabrication Process Level SEU Mitigation Technique .....	14
2.3.2 System Level SEU Mitigation Technique .....	15
2.4 Conclusion .....	16
Chapter 3 Subthreshold Combinatorial Logic Styles	
3.1 Overview .....	17
3.2 Results.....	19
3.3 Simulation Analysis .....	20
3.4 Proposed Floating Gate Logic Style .....	23
3.5 Simulation Results .....	26
3.6 Conclusion .....	30

## Chapter 4 Subthreshold Sequential Circuits Styles

4.1 Overview .....	31
4.2 Robust Flip-Flop Design at 0.25 $\mu\text{m}$ (PDSOI) Transistor Model.....	32
4.3 Simulation Results for 0.25 $\mu\text{m}$ (PDSOI) Transistor Model .....	36
4.4 Proposed Novel Flip-Flop Design at 45 nm (FDSOI) Transistor Model.....	40
4.4.1 Proposed Flip-flop Design.....	41
4.5 Conclusion .....	46

## Chapter 5 Subthreshold Voltage Boost and Logic Interface Circuits

5.1 Overview .....	47
5.2 Charge Pump Designs.....	47
5.3. Novel SOI Subthreshold Charge Pump .....	53
5.4. Simulation Results .....	55
5.6 Level Shifters to Interface Sub and Super Threshold Logic Cells.....	59
5.7 Existing Level Shifters.....	60
5.8 Proposed Level Shifters .....	64
5.9 Simulation Results .....	66
5.10 Conclusion .....	71

## Chapter 6 Subthreshold Radhard Reconfigurable (SRR) Architecture

6.1 Programmable Processing Element (PPE).....	73
6.1.1 Subthreshold Multiplexer .....	76
6.2 Interconnection Crossbar .....	78
6.4 Level Shifters for SubT and SuperT logic Interface.....	80
6.5 Conclusion .....	82

## Chapter 7 SRR Simulation Results

7.1 SRR MAC Configuration .....	83
7.1.1 Linear Feedback Shift Register .....	84
7.1.2 Array Multiplier.....	85

7.1.3 Ripple Carry Adder and Accumulator.....	86
7.2 SubT-MAC HSPICE Functional Verification .....	87
7.3 MAC Design Comparison.....	90
7.4 Conclusion .....	95
Chapter 8 Conclusions and Future Work	
8.1 Conclusions.....	96
8.2 Future Work .....	100
Chapter 9 References .....	101
Appendix A Reconfigurable Computing	
A.1 Reconfigurable Computing.....	106
A.2 Classification of Reconfigurable Processors.....	107
A.2.1 Classification based on Granularity .....	107
A.2.3 Classification based on programmability .....	108
A.2.3 Classification based on Degree of coupling .....	110
Appendix B SPICE Code	
B.1 16-bit MAC SPICE Code .....	112
B.2 Level Shifter Interface SPICE Code .....	127
B.3 Charge Pump Interconnect SPICE Code .....	136
B.4 Flip-Flop Analysis SPICE Code .....	142
Curriculum Vitae .....	151

## List of Figures

Figure 1.1 Research Focus .....	2
Figure 2.1 Latch Free Clock Gating .....	6
Figure 2.2 Latch Based Clock Gating .....	6
Figure 2.3 Multi-Threshold Voltage Design .....	7
Figure 2.5 Subthreshold Operation .....	10
Figure 2.6 Subthreshold Operation and Reconfigurable Computing .....	11
Figure 2.7 Heavy Ions and Protons trapped in Magnetic Belts Around Earth .....	12
Figure 2.8 Radiation Effect .....	13
Figure 2.9 Silicon-On-Insulator (SOI) Process .....	14
Figure 2.10 Triple Modular Redundancy Technique .....	15
Figure 3.1 Subthreshold Inverters .....	18
Figure 3.2 SOISPICE5 Voltage Transfer Curves at Subthreshold .....	20
Figure 3.3 Proposed FGS Inverter .....	24
Figure 3.4 Proposed FGS Nand .....	25
Figure 3.5 Waveforms of FGS .....	27
Figure 3.6 Floating Gate Voltages of FGS .....	28
Figure 4.1 Traditional Flip-Flops .....	33
Figure 4.2 Radiation Hardened Flip-Flops .....	34
Figure 4.3 Dynamic Based Flip-Flops .....	35
Figure 4.4 SOISPICE5 simulation of a subthreshold flip-flop .....	36
Figure 4.5 Proposed MOSAC FF .....	42



Figure 4.6 Subthreshold (45nm) ring oscillator EDP plot .....	43
Figure 4.7 DICE FF Transient Fault Simulations at SuperT .....	45
Figure 4.7 MOSAC FF Transient Fault Simulations at SuperT .....	45
Figure 5.1 Dickson Charge Pump .....	48
Figure 5.2 DCTS 2-phase Architectures .....	50
Figure 5.3 DCTS 2-phase Architecture with cross-coupled NMOS.....	51
Figure 5.4 Four Phase Architecture .....	52
Figure 5.5 Clock Booster .....	53
Figure 5.6 Proposed Active Body Voltage Doubler .....	55
Figure 5.7 SOISPICE5 Simulations for DCP Charge Pump .....	56
Figure 5.8 Charge Pump Voltage Gain Comparisons.....	57
Figure 5.9 Charge Pump Output Voltage Vs Load Current Comparisons .....	58
Figure 5.10 Conventional Level Shifters .....	61
Figure 5.11 Voltage Doubler .....	63
Figure 5.12 Proposed Level Shifters with Body Ties .....	65
Figure 5.13 SOISPICE5 Simulation Waveforms.....	68
Figure 5.14 Bar Chart Comparisons of Level Shifters .....	71
Figure 6.1 Programmable Processing Element.....	74
Figure 6.2 Multiplexer Designs .....	77
Figure 6.3 Interconnection Switch.....	78
Figure 6.4 SubT Interconnect with Charge Pumps.....	79
Figure 6.5 HSPICE Simulation Waveform for Interconnect Routing with Charge Pumps.....	80
Figure 6.6 HSPICE Routing Simulation Waveforms Without and With Charge Pump .....	80

Figure 6.7 Interfacing SubT and SuperT with Level Shifters.....	81
Figure 6.8 HSPICE Simulation of Interface of SubT and SuperT with Level Shifters .....	82
Figure 7.1 MAC Unit Design.....	83
Figure 7.2 Linear Feedback Shift Register Design.....	85
Figure 7.3 Basic Cell of a Parallel Array Multiplier.....	86
Figure 7.4 Array Structure of Parallel Multiplier .....	86
Figure 7.5 Ripple Carry Adder .....	87
Figure 7.6 SubT-MAC Inputs – Two 8-bit Values .....	87
Figure 7.7 SubT-MAC Multiplier Output (P15-P0) .....	88
Figure 7.8 SubT-MAC Adder Output (Add15-Add0) .....	89
Figure 7.9 SubT-MAC Register Output (MAC15-MAC0) ... ..	89
Figure 7.10 Power Consumption Split for the SRR-MAC Implementation.....	93
Figure 7.11 Xilinx Spartan II laboratory Setup .....	93
Figure 7.12 Power Comparison of Measured Vs Xpower Software Tool .....	93
Figure A.2 Degree of Coupling .....	110

## List of Tables

Table 3.1 Comparison of Inverter Circuits in Subthreshold Regime.....	20
Table 3.2 FGS Metrics Comparison .....	28
Table 4.1 0.25 $\mu\text{m}$ (PDSOI) Comprehensive Comparison of Flip-flops .....	38
Table 4.2 45 nm (FDSOI) Flip-flop Simulation Analysis .....	44
Table 5.1 Charge Pump Performance Comparison.....	59
Table 5.2 Level Shifter Design Parameters .....	70
Table 6.1 Multiplexer Design Comparison.....	77
Table 7.1 Simulation Verification of SRR-MAC .....	88
Table 7.2 Power Consumption Comparison Between SRR-MAC and ASMAC.....	90
Table 7.3 Power Performance Configurations of SRR-MAC .....	91
Table 7.4 Energy Harvesting Sources .....	91
Table 7.5 Xilinx Spartan-II Analysis .....	92
Table 7.6 Architecture Merit Comparison .....	93
Table A.1 Summary of Probability Classes.....	109

# Chapter 1

## Introduction

### 1.1 Motivation

The current technology pursuit in the Integrated Circuit (IC) industry is to comply with Moore's law and will pursue to do so for at least the near future. With the reduction in feature size, to nanometer scale, operating Voltage ( $V_{dd}$ ) and the circuit node capacitance  $C_{node}$  ( $Q = V_{dd} * C_{node}$ ), have been reduced drastically [1-2]. Consequently, a significantly lower charge dose injection is sufficient to flip the logic value of the node. Therefore, maintaining consistent reliability in ICs is a greater challenge at nano metric scale. This applies not only to mission critical applications like Space but also to earth-bound electronics like PCs, servers and networks that are affected by alpha particles emitted from packaging material. Modern applications, like micro-sensor networks and implanted bio-medical devices, demand energy conservation with hardware flexibility. Research has been reported that address these issues on a stand-alone basis but none to this date has attempted a unified approach. Therefore, this research comprehensively addresses the requirements of the above mentioned applications, with a hardware design capable of providing ultra-low power consumption, hardware flexibility and resilience in radiation environment. The proposed research area, which combines three fields: subthreshold operation, radiation hardening and reconfigurable computing, is shown in Figure 1.1.

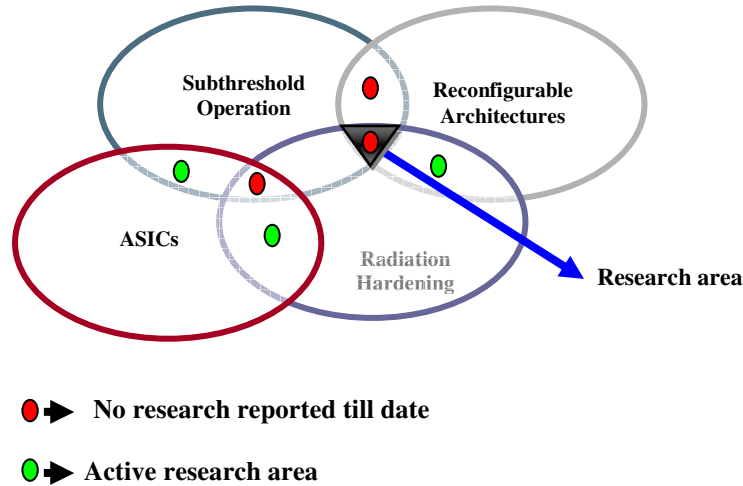


Figure 1.1 Research Focus

## 1.2 Research Objective

The objective of the research is to provide a comprehensive solution that collectively addresses three issues: 1.) energy conservation, 2.) application flexibility and, 3.) resilience to radiation effects. By combining the three fields, as shown in Figure 1.1, the proposed research provides novel circuit designs that are well suited to provide application versatility, energy harvesting and robust operation in radiation rich and noisy environment. The proposed research incorporates novel circuit modules to form the Subthreshold Radhard Reconfigurable (SRR) architecture. The design process of the SRR is divided into five stages: 1.) subthreshold logic style analysis, 2.) subthreshold sequential circuit analysis, 3.) subthreshold to super threshold logic interface with level boost, 4.) reconfigurable unit design and test and 5.) SRR design analysis with traditional FPGA design and MIT's subthreshold design. Apart from the above mentioned analysis, provisions are provided to operate the SRR in dual voltage mode. When performance is critical, operating frequency and voltage are increased, called the active-mode.

Another approach is to lower frequency and voltage for reduced performance, called the sleep-mode, to preserve system state.

### **1.3 Research Approach and Dissertation Organization**

Chapter 2 provides background information on low power consumption techniques, subthreshold operation, radiation effects and Single Event Upset (SEU) mitigation techniques. The theory related to the field of reconfigurable computing is covered in the Appendix A, which is extracted from [3]. The simulated SPICE code for the SRR architecture is provided in Appendix B. The following discussion explains the research approach for the SRR design:

#### **a. Subthreshold logic style and analysis**

As described in Chapter 3, the initial step of the SRR design process is to analyze different circuit styles for logic implementation at subthreshold voltage for performance and noise immunity. The Chapter also introduces a novel floating gate logic technique for subthreshold.

#### **b. Subthreshold sequential circuit analysis**

As memories occupy the largest block area in the modern ICs and are more sensitive to failure than combinatorial logic, optimized novel radhard flip-flop designs for operation at subthreshold voltage are proposed and analyzed in Chapter 4.

#### **c. Subthreshold to super threshold logic interface with level boost**

To enable dual mode (active- and sleep-mode) operation and to interface circuits with higher voltage designs, optimized level shifters and charge pump circuits are proposed for efficient operation at subthreshold voltage logic. Chapter 5 provides the details of power and performance analysis.

**d. Reconfigurable unit design and test**

Chapter 6 analyzes the optimized design for the reconfigurable unit and all the key components such as Look Up Table (LUT) design, multiplexers, and interconnection network.

**e. SRR design analysis with other existing designs**

Chapter 7 provides a more comprehensive analysis of the SRR design, the design is configured to work as a 16-bit Multiply and Accumulate (MAC) unit. The implementation is compared to an existing subthreshold non-hardened Application Specific Integrated Circuit (ASIC) MAC operating at 175 mv and an off the shelf commercial Xilinx FPGA operating at 2.85v. Chapter 7 also provides the comparison analysis, which is important to understand the power and performance relations for the three designs.

## Chapter 2

### Background

#### 2.1 Low Power Design

With the advancements in semi-conductor technology, the device density continues to grow, increasing system complexities. In addition, to achieve higher performance, clock frequency is increasing as well. The net side effect is increased device self-heating and higher power consumption, which is not suitable for many portable electronics. This directly affects system reliability and power costs, especially for battery-operated applications. Techniques for low power are applied at various levels of abstraction during product development. Primarily, power minimization techniques target two key factors: supply voltage ( $V_{dd}$ ) and effective switching capacitance ( $C_l$ ). The equation below demonstrates the quadratic dependence of power consumption on these two mentioned factors.

$$P_{dyn} = C_l * V_{dd}^2 * f * P \quad (1)$$

Where,  $P_{dyn}$  is the dynamic power dissipation,  $C_l$  is the capacitive load seen by the device,  $V_{dd}$  is the supply voltage,  $f$  is the frequency of logic transitions, and  $P$  is the probability of the occurrence of transition (i.e., a power-consuming transition) [4]. Static power dissipation is assumed to be significantly smaller than dynamic dissipation. For the current discussion, low power techniques are classified into two categories:

1. traditional low power design and
2. advanced low power design.



### 2.1.1 Traditional Low Power Design Techniques

Two techniques have been defined in the traditional low power design: (1) clock gating and (2) multi-threshold voltage design.

#### 1) Clock Gating

The low power consumption with clock gating technique primarily involves gating the clock to reduce the activity of flip-flops. The objective is to effectively reduce the dynamic capacitance by switching off unused clocked tree nets. Challenges for this technique are *Enable* signal timing, delay variation, power surges, and increase in testing and verification. This technique can be implemented in two ways: 1. latch free clock gating and 2. latch based clock gating, where are explained in Figures 2.1 and 2.2, respectively.

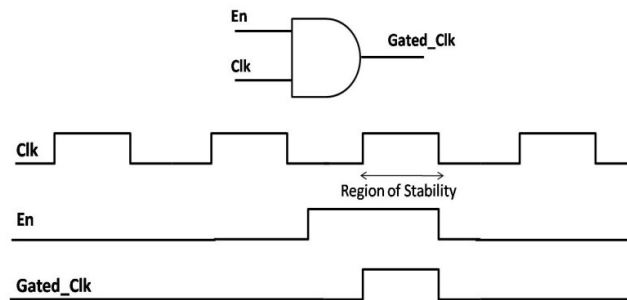


Figure 2.1 Latch-Free Clock Gating [5]

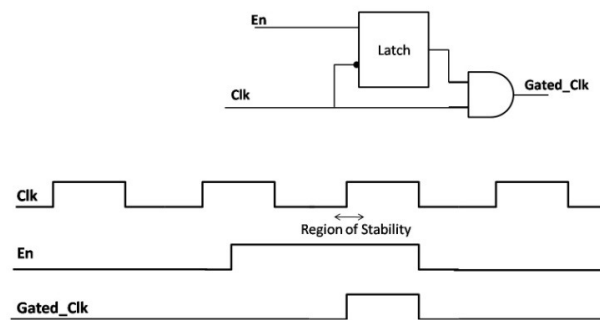
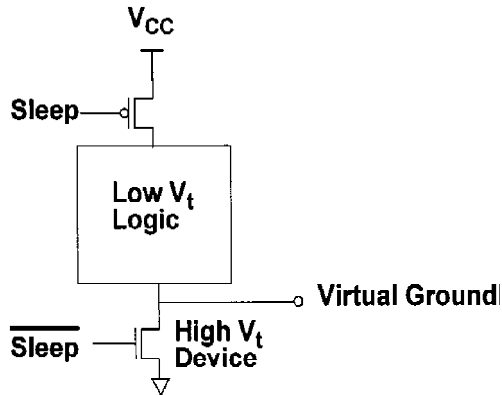
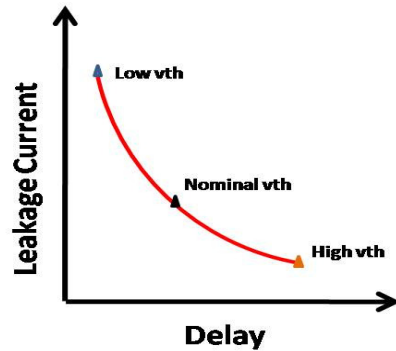


Figure 2.2 Latch Based Clock Gating [5]

## 2) Multi-Threshold Voltage Design



a. Multi-threshold CMOS circuit structure [6]

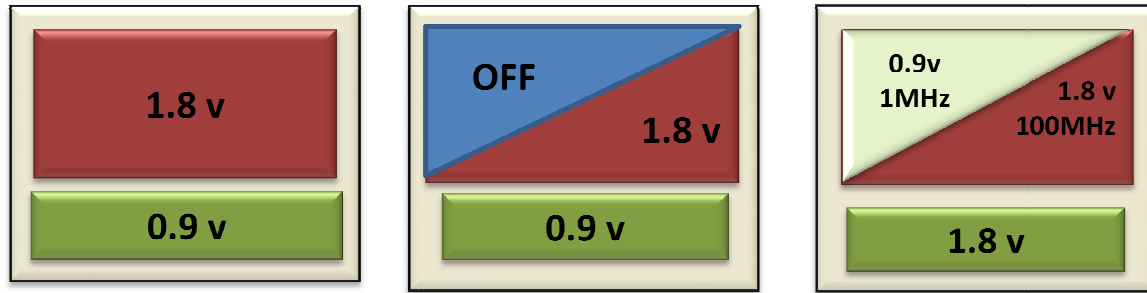


b. Leakage current Vs delay

Figure 2.3 Multi-threshold voltage design

For current generations of ICs severe  $V_{cc}$  (supply voltage) and  $V_t$  (threshold voltage) scaling has resulted in significant increases in leakage current, which causes reductions in switching energies and power dissipation. Therefore, there is a need to optimize the  $V_{cc}$  and  $V_t$  point for a given frequency operation [6]. However, the optimal minimum energy  $V_{cc}$  and  $V_t$  point is significantly below the typical threshold voltage levels of today's technologies. Multi-threshold Complementary Metal Oxide Semiconductor (MTCMOS) is a dual technology that is very effective at reducing leakage currents. High- $V_t$  devices can be used to reduce leakage currents while low- $V_t$  devices can be used whenever high performance is required. The most straight forward application of dual- $V_t$  technology is simply to partition a circuit into critical and noncritical regions, and to only use the fast low- $V_t$  devices when necessary to meet performance goals. This approach will reduce subthreshold leakage currents both in the active mode and standby mode, but may provide limited leakage reduction if the circuit contains many critical paths.

### 2.1.2 Advanced Low Power Design Technique



a. Multi-Voltage      b. Multi-Voltage with Power Gating      c. Multi-Voltage Dynamic Scaling

Figure 2.4 Advanced low power design techniques

As shown in Figure 2.4a with multi-voltage design, some blocks of the IC use lower supply voltage than others, creating “Voltage Islands” and enabling different levels of performance and optimized power consumption based on the algorithm under execution. The complexity of the design increases if dynamic voltage scaling is used to change the supply voltage level during operation. As shown in Figure 2.4b, to achieve further optimization designers use power gating to shut down blocks that are not in use. For such a design the power-down and power-on sequences can be extremely complex to design and verify. The concept in Figure 2.4c is extended to propose a complex design that employs dynamic voltage scaling to the extreme (operating between subT and SuperT voltages) for some blocks while other blocks operate with higher voltage. This design requires additional circuits like level-shifters for the interface, ring oscillators for frequency scaling, and charge pumps for boosting the logic voltage. This research implements all the modules required to implement this design. The functionality and circuit analysis for this design are explained in the following Chapters.

## 2.2 Subthreshold Logic Operation

Subthreshold circuits operate with a supply voltage that is less than the threshold of the transistor – far below traditional levels and, consequently, the transistor operates essentially on leakage. While traditional digital Complementary Metal Oxide Semiconductor (CMOS) has relied on running transistors either in the *on* state (saturation) or *off* state (subthreshold), subthreshold circuits are either in an *off* state or an *almost-on* state (still in the subthreshold regime but with weak inversion). Running at these non-standard operating points limits performance, which remains acceptable for low-to-medium cost applications like micro-sensor networks, wearable electronics, etc., given the substantial increase in the corresponding energy efficiency. Figure 2.1 illustrates the  $I_{ds}$  vs.  $V_{gs}$  for both NFET and PFET devices in logarithmic format. The Figure indicates how although the devices are never fully *on* in subthreshold mode, a sufficient difference in drive current exists between the *on* and *off* state to allow logical function. As power is related quadratically to supply voltage, reducing the voltage to these ultra low levels results in a dramatic reduction in both power and energy consumption in digital systems.

One challenge, however, is that at these low voltages, circuits become less robust to environmental and manufacturing factors, which can disrupt correct circuit operation. Temperature and across-chip parametric variation affect the transistor threshold voltage, which, in turn, has an exponential impact on the drive strength of the transistor – degrading performance or unbalancing the ratio of *on-to-off* current. Furthermore, the substantially reduced operating voltage results in a commensurate reduction in the critical charge -  $Q_{crit}$  - required to erroneously flip a bit in a memory cell or sequential element due to radiation or noise. These sensitivities require exaggerating design margins - diluting the energy benefits of subthreshold logic.

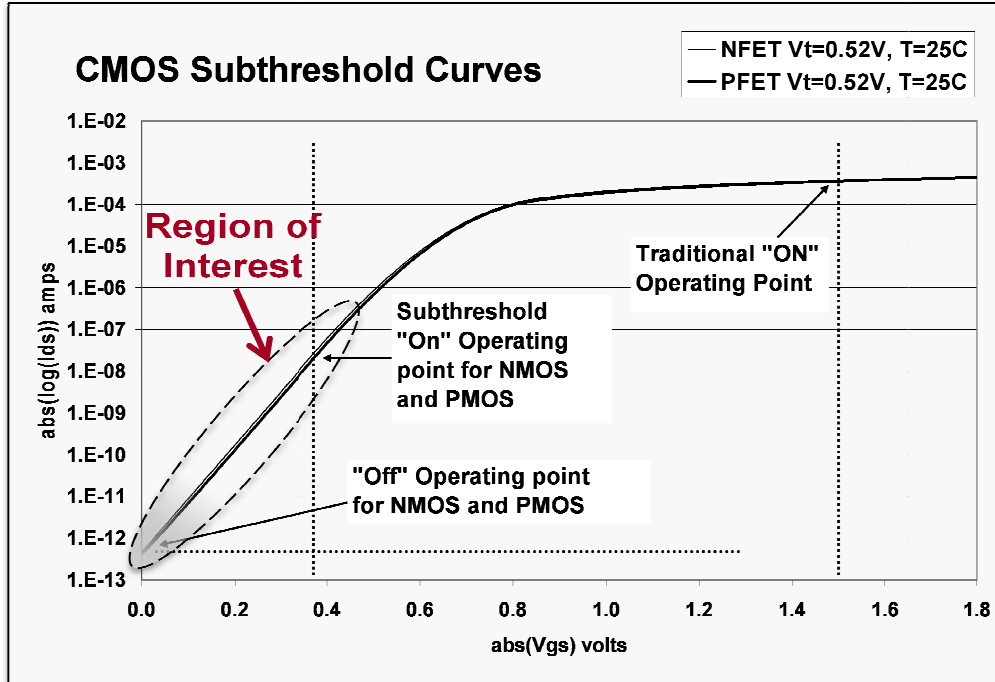


Figure 2.5 Subthreshold Operation

### 2.3 Subthreshold Operation and Reconfigurable Computing

As mentioned in the previous section, subthreshold circuits enable ultra low power operation but simultaneously suffer from reduced robustness and performance. And, as described in Appendix A, architectures enabling reconfigurable computing have the advantage of application flexibility, and by implementing parallel datapaths improve performance and concurrently, enable dynamic error correction, which in turn enhances robustness. However, due to inherent redundancy the reconfigurable architectures have large area and in turn increased power consumption overhead. Figure 2.6 lists the advantages and disadvantages of subthreshold operation and reconfigurable computing. By combining the two fields the disadvantages of each field will be ameliorated by the advantages of the other field.

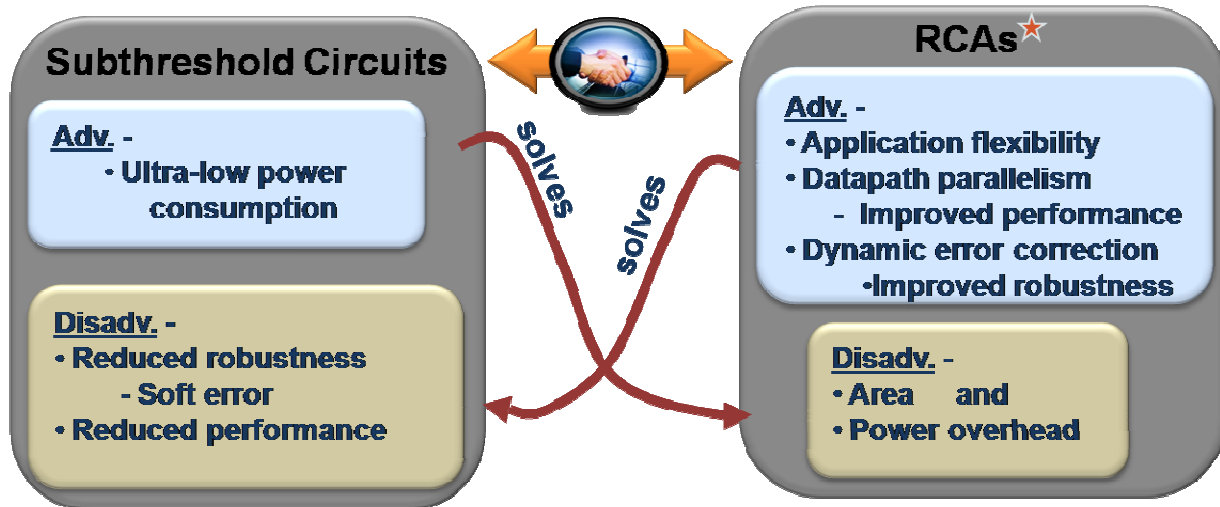


Figure 2.6 Subthreshold Operation and Reconfigurable Computing

## 2.3 Radiation Hardening and Single Event Upsets (SEUs)

Single Event Upsets (SEUs) in modern VLSI systems are a major reliability concern. These upsets originate from two primary sources: cosmic ray particles occurring in the space environment and alpha particles emitted from the radioactive decay of uranium and thorium impurities located within the chip such as the silicon die, interconnects, and ceramic packaging. Soft errors due to SEUs have been a known problem affecting semiconductor memories for quite some time [7]. However, due to faster clock rates and shrinking process technologies, SEUs are now affecting CMOS logic [8].

The most important thing to understand about the design of radiation-hardened Integrated Circuits (ICs) is that there is no “silver bullet” [9]. A common misconception in the design industry is that radiation hardening is something you *do* to a design - you switch substrates, change design rules, use circuit tricks, and/or modify fabrication steps, thereby transforming a

commercial Integrated Circuit (IC) design into a hardened design. Each of these factors influences the response of an IC in radiation environments, but the influences are not independent of one another. The complex inter-relationships between technology, design, and fabrication require that radiation-hardness objectives be considered from the outset of an IC design to ensure maximum hardness and reliability at minimum cost in terms of performance and production [9].

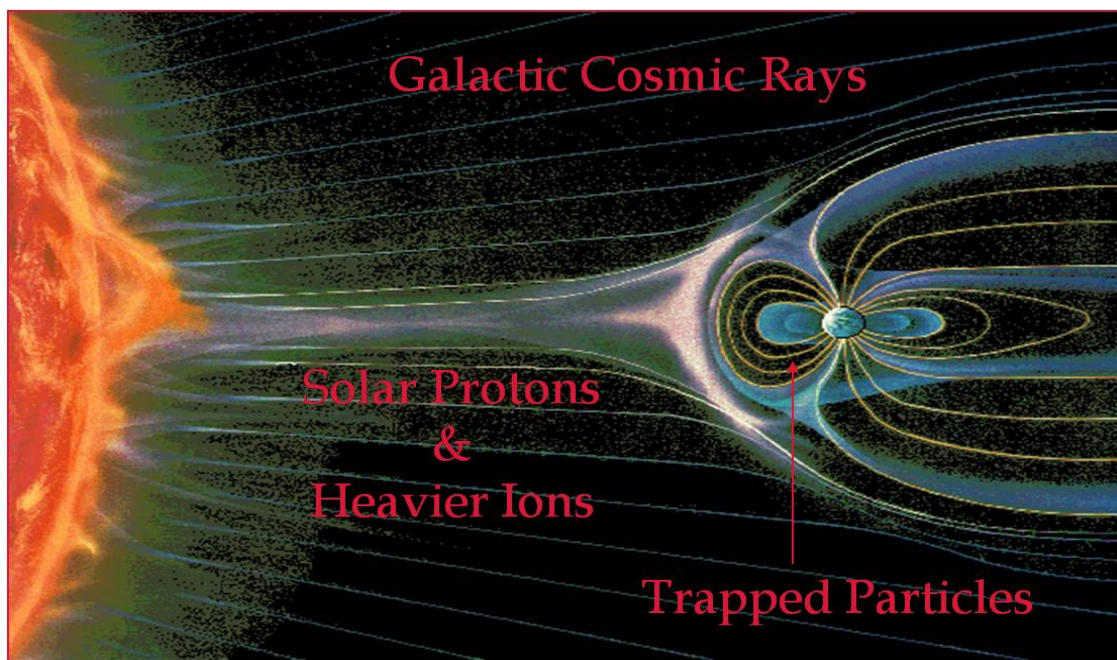


Figure 2.7 Heavy Ions and Protons trapped in magnetic belts around Earth [NASA/GSFC]

Single-event upset (SEU) is a reversible change in a digital logic state due to an energetic particle (ion, neutron, or photon) penetrating a microelectronic device, and is often referred to as a "soft error." SEUs in space-borne electronics are primarily due to the energetic ions called "cosmic" ions, and/or the protons trapped in the magnetic fields surrounding some planets,

shown in Figure 2.7. When a cosmic ion passes through a semiconductor, an ionized plasma track is created. As shown in the Figure 2.8, the holes and electrons comprising this track can be collected at device junctions, creating electrical perturbations that can result in changing the circuit response.

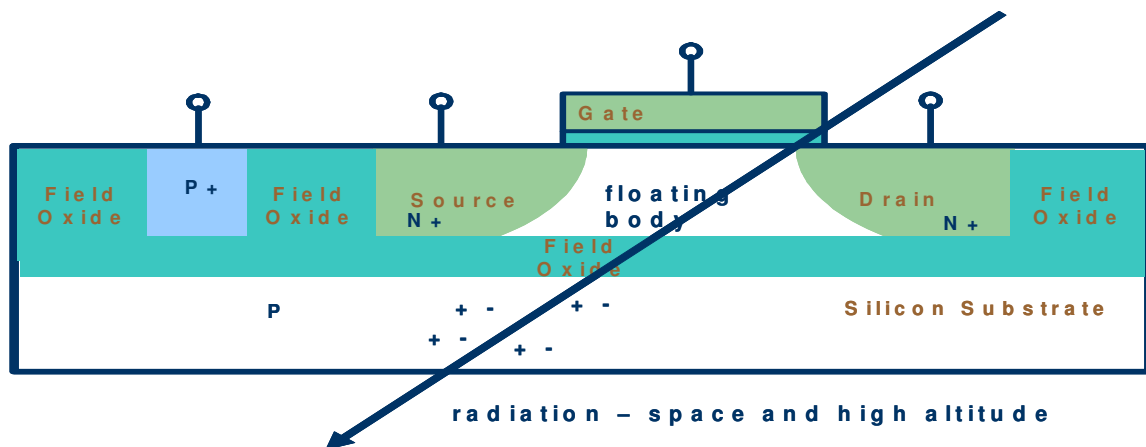


Figure 2.8 Radiation Effect

SEU in static latches and SRAMs became an important issue once feature sizes dropped below 10 microns and the critical charge for upsetting a circuit dropped below 1 pC (roughly corresponding to a particle LET of 50MeV cm<sup>2</sup>/mg and a collection depth of 2 microns) [8]. The vulnerability of a device to SEUs is determined by both circuit and technology factors, including the total charge stored on "information" nodes, the ability of the circuit to re-supply charge to perturbed nodes, and the charge-collection volumes associated with nodes sensitive to single-event perturbations. Accordingly, SEU mitigation techniques can be broadly classified into two categories:

1. Fabrication Process Level SEU Mitigation Technique (Process-Hardening)
2. System Level SEU Mitigation Technique (Design-Hardening)



### 2.3.1 Fabrication Process Level SEU Mitigation Technique

The first and most significant step that is included in this technique is the Silicon-On-Insulator (SOI) process. The major difference between SOI and normal bulk CMOS is the insertion of an insulation layer beneath the devices, which is shown in the Figure 2.9. Following are some of the important advantages of SOI are: i) diffusion capacitance reduction, ii) latch-up elimination, and iii) SEU improvement. SOI structures are available in two forms: 1) fully-depleted structure and 2) partially-depleted structure. The one big difference between them is that the silicon thickness of a fully-depleted device is thinner than the silicon thickness of a partially-depleted device. This allows the fully depleted SOI structure to eliminate any mobile body charge under all bias conditions [10]. For the current research, the partially-depleted SOI structure is chosen. Other techniques in this category include trench isolation, charge recombination, and higher  $V_t$ .

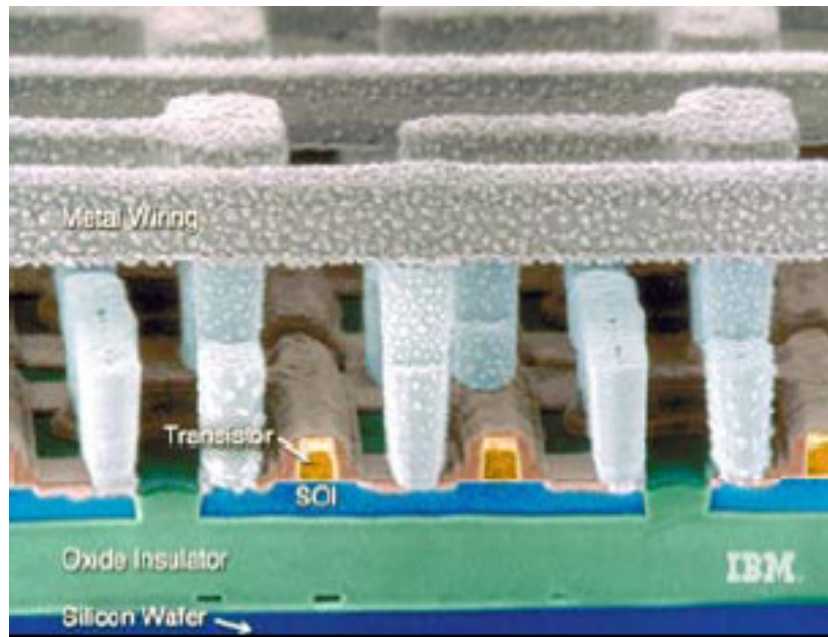


Figure 2.9 Silicon-On-Insulator (SOI) Process [6]

### 2.3.2 System Level SEU Mitigation Technique

SEU mitigation is essential to ensure absolute reliable operation of a system in a radiation environment. The primary techniques that fall under this category are: 1) Triple Modular Redundancy (TMR), 2) memory scrubbing, and 3) radiation hardened storage cells. TMR is one of the most common methods of SEU mitigation in reconfigurable architectures [11]. In TMR, the logic of the design can simply triplicate, with redundant majority voters on the output, which is shown in Figure 2.10. TMR does come with a penalty of triple power consumption and delay. Scrubbing refers to periodic read back of the memory, comparing the data to a known good copy, and writing back with (if any), corrections [12-14]. The TMR technique has been shown to greatly reduce the dynamic cross section of an FPGA design and, when combined with bitstream scrubbing, can virtually eliminate the configuration bitstream from SEU susceptibility, which is shown in the figure. Therefore, TMR can greatly decrease the downtime of a circuit in radiation environments.

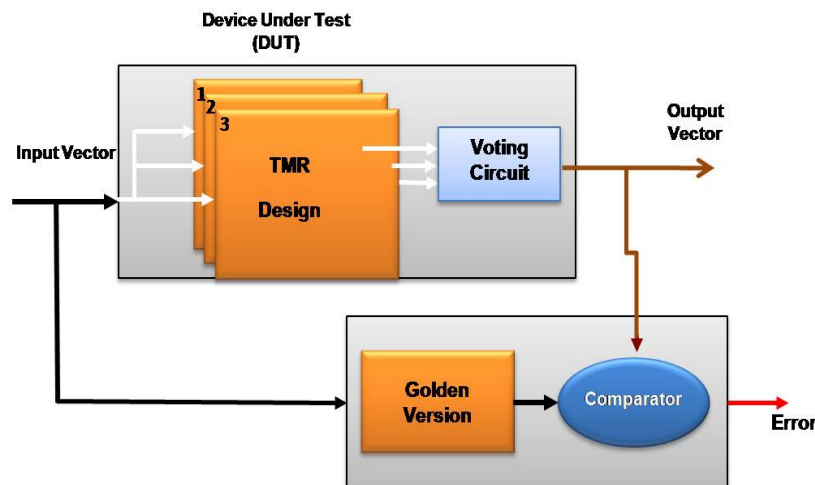


Figure 2.10 Triple Modular Redundancy technique

## **2.4 Conclusion**

This Chapter presented an overview of low power techniques, radiation effects, subthreshold operation, and popular SEU mitigation techniques. The topics covered in this Chapter form the basis for the discussion of the dissertation work. The functional modules designed for the proposed work are optimized for subthreshold voltage operation and apply advanced low power techniques discussed in this Chapter. By choosing the SOI design process, the proposed work benefits in terms of performance and in reducing radiation effects. To achieve a higher degree of robustness in subthreshold, flip-flop designs use the TMR design technique.

## Chapter 3

### Subthreshold Combinatorial Logic Styles

#### 3.1 Overview

As supply voltages continue to scale with each new generation of semiconductor technology, the obvious extrapolation is the subthreshold regime. Consequently, research activity in the subthreshold logic has steadily increased. One of the first publications to describe digital logic at these low voltages is from [15] in which the power and performance of the basic CMOS inverter (illustrated in Figure 3.1-a) was evaluated analytically. For this traditional style, the subthreshold regime was identified as the optimal point for the energy-delay product.

Several publications have focused on evaluating the effectiveness of different circuit styles in the subthreshold regime. The  $V_{dd} - V_t$  space was comprehensively mapped through simulations to obtain the optimal point for operation for traditional CMOS for a variety of performance levels [16]. Based on the results of this study, a Fast Fourier Transform was designed and fabricated in 0.18u micron technology. The chip demonstrated robust operation and provided dramatic power savings. The lowest known  $V_{dd}$ , was reported by [17], with the theoretical limit of  $\sim 2nkT/q$  ( $\sim 52$  mV at  $25^\circ\text{C}$  for ideality  $n=1$ ). Precise control of the body voltage was required to maintain threshold stability.

Pseudo-NMOS (PNMOS – shown in Figure 3.1-b) was investigated by [17]; this work was motivated by the area savings of this style based on the elimination of the dual PFET pull-up network, which is replaced by a single always-on PFET load. In the super-threshold regime this style is plagued with large static power consumption as there is often a direct path between power and ground supplies. However, this is mitigated in the subthreshold regime as the load is

left in weak inversion rather than fully driven. Consequently, PNMOS is an intriguing alternative to traditional CMOS.

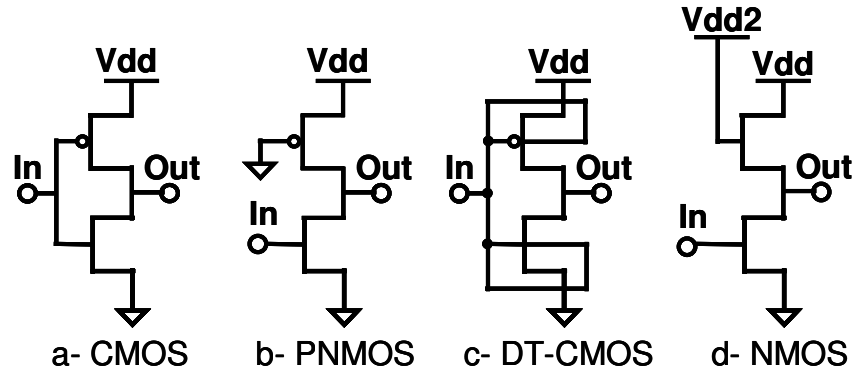


Figure 3.1 Subthreshold Inverters

One challenge with subthreshold operation is that the transistor drive currents are exponentially related to the threshold voltage, rather than linearly to quadratically related is the case in super-threshold operation and this can lead to pull-up / pull-down mismatches. An elegant solution was proposed to this sensitivity and is referred to as Self-Adjusting Threshold logic [18]. In this scheme, the bodies of PFETs and NFETs are adjusted independently through the use of simple charge pumps. By monitoring on chip leakage currents on chip in real time, the body voltages (PWEEL / NWEEL) can be manipulated to indirectly adjust the threshold voltage through the MOS body effect and, thus, maintain balance. In some cases, the approach – in some cases - tames the process variation and can be beneficial as well for applications that are subject to wide temperature variation. However, the proposed circuit can only decrease the threshold voltage below the nominal value – not increase in the case of low thresholds voltages. Additionally, the technique requires extraneous circuits for the monitoring and controlling of the well voltages - increasing die area.

Partially-Depleted Silicon-On-Insulator (SOI) became commercially available in the late nineties. This spawned reports of new circuits that took advantage of the floating body (well)

voltage. Additionally, at low supply voltages, parasitic junction capacitances are increased significantly in bulk silicon. As SOI virtually eliminates the parasitic capacitances, the fabrication technique is a natural technology choice for subthreshold circuits. Exploiting the ability to control individual body voltages in SOI, Dual Threshold CMOS (DTCMOS) was proposed where the input to the CMOS circuit is also tied to the bodies of the transistors as shown in Figure 2-c [19]. When asserted high, the input activates the NFET while simultaneously raising the body voltage. This, in turn, lowers the threshold voltage, temporarily increasing pull-down strength. At the same time, the PFET is turned off and the raised body voltage results in less leakage. Conversely, when the input is held low, the on PFET is strengthened and the off NFET weakened. This can be applied not only to CMOS but any circuit style; however, only for the transistors with gates tied to inputs (i.e., excluding always-on load transistors). Originally, this configuration was not plausible for super-threshold operation because raising the body of the NFET above the value of a diode drop caused the body-to-source junction to forward bias; however at subthreshold levels this is not possible.

### **3.2 Simulation Results**

Table 3.1 depicts the results of the six circuit styles in four separate categories: static, dynamic, energy and area. For each of the first three categories, all manufacturing corners (three for NMOS and five for all others) were simulated and the worst-case of value was reported [20]. For area, the total transistor width for the circuit was reported for the inverter. Additionally, the area of a 3-input NOR was included by extending the sizing from the inverter. Figure 3.2 shows the SOISPICE5 simulations of the voltage transfer curves for CMOS inverter at subthreshold voltage levels.

**Table 3.1** Comparison of Inverter Circuits in Subthreshold Regime

Circuit	Static Characteristics (volts)		Dynamic Characteristics (seconds)		Energy (Joules)			Area (um)	
	Vol	Voh	Tphl	Tplh	Static	Dynamic	Sleep	INV	3-input NOR
CMOS	0.00	0.35	5.2E-08	3.8E-07	2.8E-15	6.8E-15	2.8E-15	30	90
Dual Threshold CMOS	0.00	0.35	9.1E-08	8.1E-08	4.2E-15	7.7E-15	4.2E-15	30	90
Pseudo NMOS	0.11	0.33	1.1E-06	1.5E-06	1.1E-13	2.4E-15	1.7E-17	11	31
Dual Threshold Pseudo NMOS	0.03	0.35	4.0E-07	6.2E-07	3.3E-13	3.2E-15	3.6E-17	13	33
NMOS - Vdd2=0.60V	0.11	0.33	1.8E-07	4.2E-07	9.7E-13	1.3E-15	9.12E-19	10.2	30.2
Dual Threshold NMOS - Vdd2 = 0.60V	0.04	0.35	5.5E-08	5.7E-07	2.7E-12	2.6E-15	7.7E-18	10.2	30.2

### 3.3 Simulation Analysis

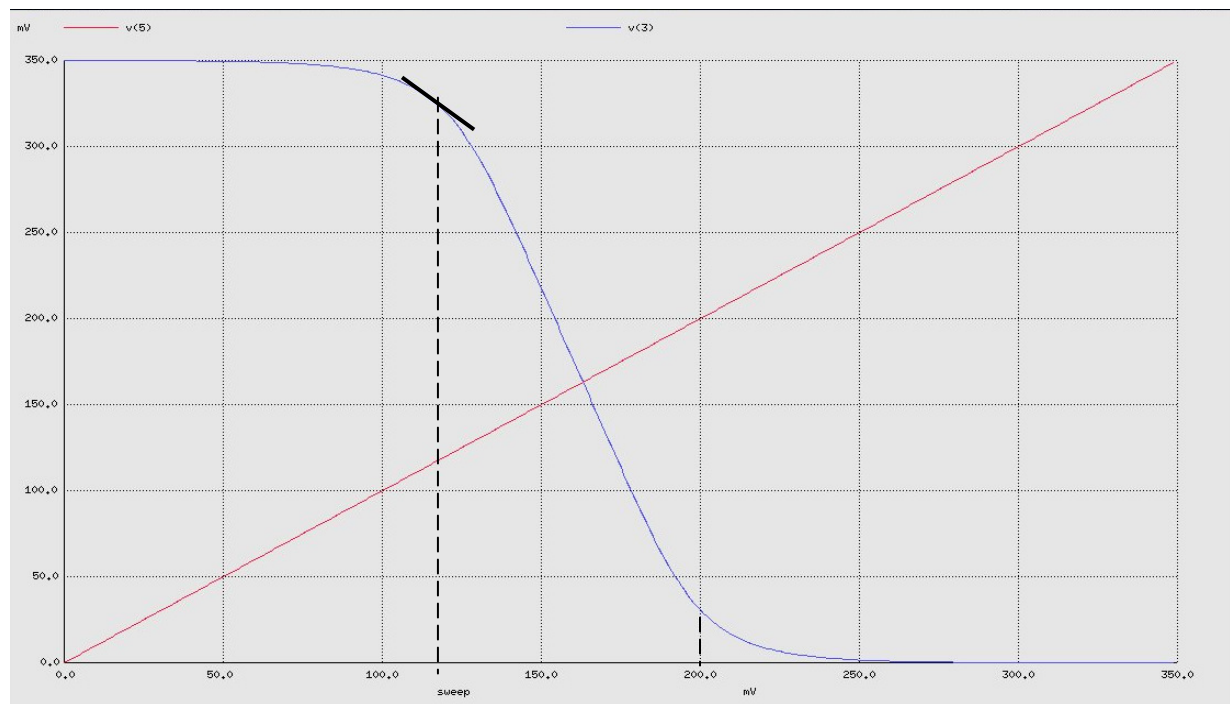


Figure 3.2 SOISPICE5 Voltage Transfer Curves at Subthreshold

CMOS circuits have been a staple in digital circuits for the past twenty years. In this experiment, traditional CMOS and DT-CMOS are analyzed and used as the standard for comparing competing circuits. Between the two styles, CMOS and DT-CMOS, dual threshold derivatives show a slight increase in performance at the expense of static and dynamic power, but generally were similar. As shown in table 3.1, the static characteristics of CMOS and DT-CMOS are the best reported providing the greatest level of noise immunity. Furthermore, static power consumption was two orders of magnitude less than the next best circuit as expected given that the other circuits implement an always-on load transistor. The major drawbacks of CMOS and DT-CMOS were dynamic power due to the large input capacitance as well as increased area – both due to the required equivalently-sized complement PFET associated with each NFET. As CMOS and DT-CMOS have no global sleep mode enable, static power was used for comparison.

Pseudo-NMOS and DT-Pseudo-NMOS provide improved area, dynamic power and sleep power relatively to CMOS and derivatives. However, the always-on PFET load results in degraded  $V_{ol}$  due to driver-to-load contention when the driver is activated. This contention does not allow the output to be driven completely to ground and furthermore, the contention results in heavy static power. Either of these two disadvantages could be improved by modifying the driver-to-load ratio, but only at the expense of the other as they have competing requirements. A larger driver would pull the output lower; however the increased driver current would further aggravates the static power consumption and vice versa.

Generally-speaking, the two NMOS circuits are similar to the PNMOS circuits given the similar always-on load configuration. These circuits also provide improved area, dynamic power and sleep power relatively to CMOS and derivatives and although the propagation delays are still not in the order of CMOS, there is a significant improvement relative to PNMOS (8% when



comparing the dual-threshold versions and 72% without). One disadvantage unique to NMOS is the required dual supply voltage; however this is less relevant today as two or more supply voltages are common in contemporary circuits.

Another substantial advantage of NMOS not captured in table 3.1 is the cost. As NMOS requires the least area, cost is minimized. However, the cost is further reduced by the simplification of manufacturing as PMOS-specific processing is no longer required. Consequently, implementing NMOS will result in increasing the number of die per wafer, while simultaneously decreasing the cost per wafer.

When considering total energy consumption, NMOS circuits are well suited for applications characterized by periodic, burst operations followed by extended periods of inactivity (examples of which include wireless network sensors). By applying a global sleep mode enable tied to all load transistors, the combinational circuits in NMOS can be shut down when idle eliminating driver contention and reducing leakages. A hypothetical example illustrates the significant reduction in total energy consumption for NMOS relative to CMOS. Consider an application that predicatively performs 50 consecutive operations every second and remains idle until the next burst occurs one second later. Assuming the frequency is set by 20 inverter delays, and then CMOS would run at 130 KHz and would require 380uS to complete the operations, while NMOS would run at approximately 110 KHz requiring 460uS. Ignoring dynamic power and considering only sleep and static power for NMOS and only static power for CMOS, the total energy would be  $4.42e-16$  and  $2.8e-15$  joules respectively. Thus, for this application, total energy per second would be reduced by 84% and the reduction would only improve for applications with longer idle times. The same concept applies to PNMOS circuits as well; however the sleep power is increased, which dilutes the power savings.

### 3.4 Proposed Floating Gate Logic Style

Floating gate technology has found wide acceptance commercially in the realm of non-volatile memories. In addition, it has been proposed as an enhancement for low voltage analog and digital circuits [15-17]. This section describes a novel approach of employing multiple input floating gate circuits enhanced by a global control signal capable of adjusting electrical performance dynamically without the delays associated with raising and lowering the power supply. This approach is well suited for applications such as wireless sensor networks in which there is a need to dynamically transform from slow, low power circuits to high performance circuits. With the goal of quickly switching between two different modes of operation (i.e., slow, low leakage mode to/from high performance mode), a standard cell library was created that included two small input capacitors for each transistor. One capacitor allows for the normal CMOS input to be capacitively coupled into the gate, while the second input was intended as a global offset that to adjust the performance of all transistors depending on system application requirements. Figure 2 illustrates an inverter with the new inputs for the true and complement values of the control signal (i.e., turbo mode). This is similar to the clever method proposed in [21], however, the biasing in this work is a global digital signal allowing for quick conversion of the energy efficiency / performance trade-off for the entire chip. Furthermore, this technique results in coupled gate voltage levels that can in low power mode rise above  $V_{dd}$  and fall below ground in order to reduce subthreshold leakage by placing *off* transistors in accumulation mode. In addition, this work differs from [21] in that non-standard (negative) voltage supplies are not required for biasing.

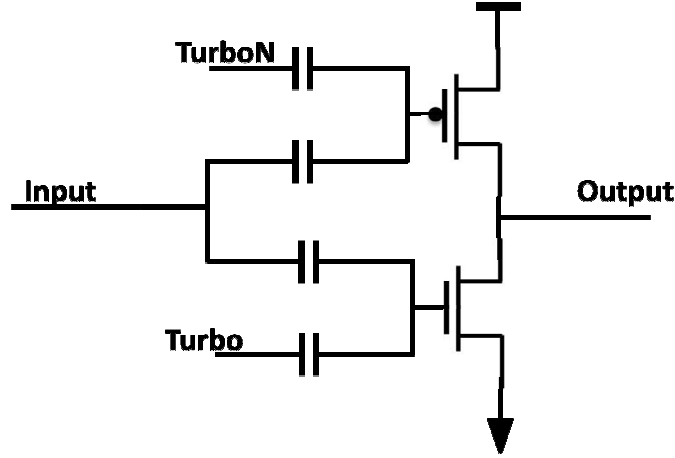


Figure 3.3 Proposed FGS inverter

In the proposed inverter of figure 3.3, for a constant supply voltage of 1 volt, any changes in the input signal result in a proportional change in the floating gate, which is then reflected at the output at rail-to-rail voltage levels. The floating gate does not necessarily swing from rail-to-rail as dictated by equation 3.1 and this may degrade performance. However, for applications in which maintaining low power consumption is a priority, such as unattended ground sensors, this degradation may be acceptable. One other impact of the less-than-rail-to-rail voltage swing is a reduction in dynamic power consumption, which helps to compensate for the increase resulting from adding the two small capacitors at each transistor.

$$\Delta V_{\text{swing}} = C_{\text{in}} / (C_{\text{turbo}} + C_{\text{in}} + C_{\text{MOS}}) * V_{\text{in}} \quad (\text{Eqn. 3.1})$$

This floating gate transistor configuration can be used to significantly reduce the static power of idle circuits significantly. Subthreshold leakage can be reduced by an exponential level given that the floating gate voltage can be coupled below ground for NFETs and above  $V_{\text{dd}}$  for PFETs and, therefore operate in accumulation mode. Approximately every 80mVs of magnitude

of the gate voltage outside of the voltage rails results in an order reduction in subthreshold leakage and this is not possible in traditional CMOS technology.

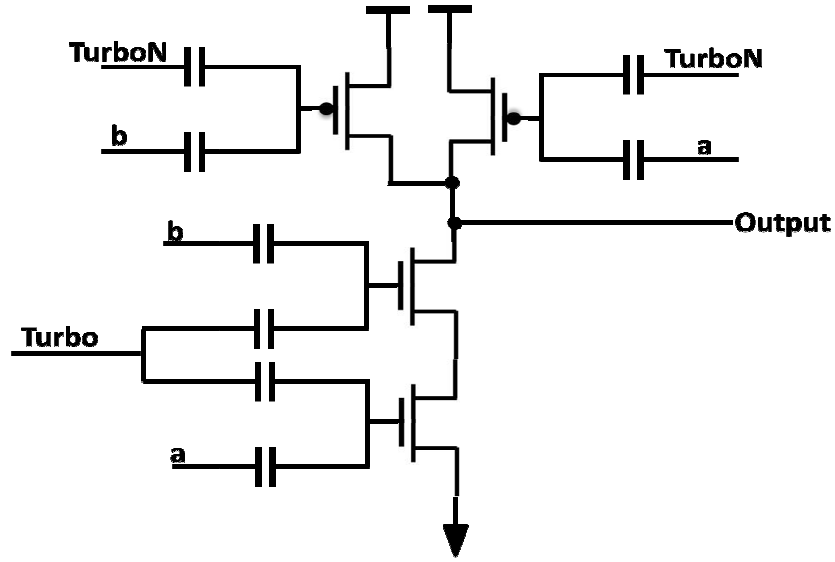


Figure 3.4 Proposed FGS NAND

In 90nm technology nodes and beyond gate oxide leakage is becoming a more dominant component in static power. The proposed configuration helps to buffer the problem with the introduction of input capacitors, which are now in series with the MOS structure. As the voltage drop across the capacitor and transistor is shared, gate oxide leakage is consequently reduced. The concept can be extended to any type of logic gate (i.e., NAND, NOR, Flip-Flop) a NAND example is illustrated in Figure 3. Furthermore, this technology can be interlaced with traditional CMOS where required. Critical paths can be maintained with the faster and leakier direct-gate CMOS and the majority of remaining logic can be implemented with the proposed circuits to minimize static power.

The second input to each transistor is a global signal that when switching states couples the NFET gate voltage by a fraction of the supply voltage. When coupled up, the NFET gate voltage swings between subthreshold (*off*) and superthreshold (*on*) levels, improving

performance. When the global turbo signal is de-asserted, the gate voltage is coupled down and the voltage swing runs from the subthreshold boundary (*on*) to below ground (*strongly off*), – providing slow performance but significantly improved static power consumption. The same is true in reverse for PFETs, which are controlled by the complement of the signal.

### 3.5 Simulation Results

To fairly assess the relative strengths and weaknesses of this circuit style, SPICE simulations using SPICE Opus 0.25u TSMC models and 0.25u models of SOISPACE5 partially-depleted SOI technology are simulated. Both simulators are branches from the original version of Berkeley SPICE 3f5. For direct comparison between traditional CMOS (at 1V), subthreshold CMOS (at 0.35V), and the proposed method, a string of ten inverters was simulated in each case with an input of four cycles of a periodic square waveform. The only difference for the floating gate simulation was that the turbo mode was initially on for the first two cycles and then de-asserted for the final two. Figure 3.5 illustrates the input (red) and output (green) waveforms for the proposed circuit style and the decrease in the performance is evident by the increase in delay, which is easily identifiable in the last two cycles. This slow-down is accompanied by an exponential decrease in leakage and illustrates that although the circuits slow down in low power mode, useful computation can continue although at a reduced level. This adjustable performance could be exploited in self-timed circuits or in ring oscillators to adjust output frequency. This work only explores the advantages of providing the benefit of dynamically switching between high performance and low power.



Figure 3.5 Waveforms of FGS

Figure 3.6 shows the two floating gate voltages for one inverter for the PFET (green) and NFET (red). Additionally, the control signal also is included (blue) and marks a single transition in the middle of the simulation. Initially, in high performance mode, the two gate voltages swing between super and subthreshold levels providing strong performance. However, in the middle of the simulation when the turbo mode de-asserts, the two gate voltages diverge with the NFET gate voltage swinging below ground when off. Similarly, the PFET gate swings above  $V_{dd}$  when idle. This illustrates how in low power mode, subthreshold leakage can be reduced by an order of 10x by running transistors in accumulation. To optimize the operation of these circuits, the initial charge on the floating gate had to be adjusted this can be accomplished as described in [18]. The ideal level was different for odd and even logic parity stages, which will require the use of two separate turbo mode signal pairs for biasing and for gate voltage programming. Table 3.2 contains the comparison of key metrics of four inverter chain simulations: 1) traditional CMOS at 1V, 2) subthreshold CMOS at 0.35V, 3) the proposed inverter chain in fast mode, and 4) the proposed inverter chain in low power mode. The first column captures the performance by the average propagation delay through 10 inverters. The performance for the floating gate inverter chain relative to traditional CMOS is reduced for the same transistor widths due to the additional

capacitor coupling of the inputs; however, when compared to the subthreshold CMOS running at 0.35V, the performance of the fast mode was over two orders of magnitude better.

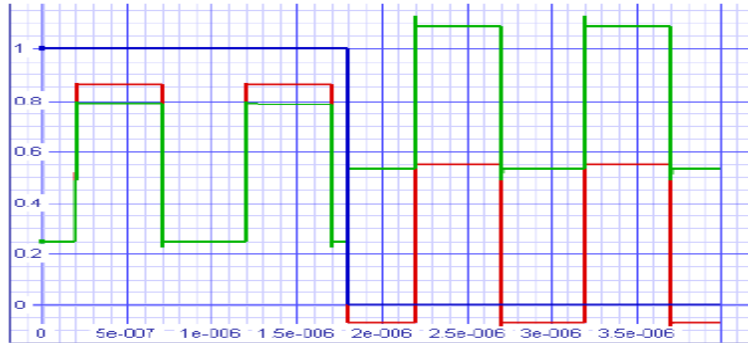


Figure 3.6 Floating Gate Voltages of FGS

Dynamic power was measured for a four-cycle input waveform. Subthreshold operation provided the lowest dynamic power but with less than an order of magnitude of improvement. If considered simultaneously with the performance degradation, the energy \* delay product of subthreshold performs poorly when compared to the other approaches.

Table 3.2 FGS metrics comparison

Design Compared	Vdd	Delay (ns)	Dynamic Power (fW)	Static Leakage (pA)
Traditional CMOS	1.0	0.21	0.81	1.89
Subthreshold CMOS	0.35	734	0.22	0.97
Proposed Fast Mode	1.0	2.3	1.42	2.3
Proposed Low Power	1.0	87	0.5	0.13

Static power was also measured and the value is worst for the high performance mode of the proposed circuit style by three orders of magnitude; however, the low power mode provided the lowest levels – beating even subthreshold operation. The improvement in the low power mode can be explained by the below ground and above  $V_{dd}$  static values that are possible with floating gate technology. Additionally, for technologies that have high gate oxide leakages, the proposed circuit would gain an additional leakage reduction. The high static power in high performance mode is due to the off transistors being only slightly in subthreshold with the gate voltage at a non-zero value. This allows for improved performance on the other end of the gate swing but with the expense of an exponential increase in leakage.

Considering all of these metrics, one can conclude that the proposed circuit style is best suited for applications with very low activity duty cycles - in which long periods of inactivity (in slow mode) are peppered with occasional, short periods of relatively high performance activity. In this scenario, the high dynamic and static power of the high performance mode can be amortized over the long durations in the ultra-efficient low power mode. Performance will be degraded by selecting this technology over traditional CMOS, but not nearly as much as with subthreshold and with a significant improvement in the overall energy budget relative to either of the two other cases. Furthermore, a hybrid approach is possible combining traditional CMOS for timing critical logic and floating gate subthreshold for the rest – obtaining the benefit of both styles. One metric not captured by the simulations in Table 3.2 is noise insensitivity. As the gate voltages are floating in the proposed circuit style and not driven strongly to a specific value, the gates can be coupled by noise and inadvertently deviate from correct logical operation. Furthermore, as the gate voltages do not swing between rail-to-rail (i.e., degraded  $V_{ol}$  and  $V_{oh}$  levels) the noise margins are compromised. This style is best suited for applications such as



wireless sensor networks in which natural redundancy in the cluster of systems provides protection against noise and soft error effects.

### **3.6 Conclusions**

The study analyzed CMOS, PNMOS, DT-CMOS and NMOS circuit styles for performance, noise immunity, and power consumption (static and dynamic) in subthreshold regime. The static characteristics of CMOS and DT-CMOS are the best reported and provide the greatest level of noise immunity. Furthermore, static power consumption was two orders of magnitude less than the next best circuit as expected given that the other circuits implement an always-on load transistor. Pseudo-NMOS and DT-Pseudo-NMOS provide improved area and dynamic power relatively to CMOS and derivatives. However, the always-on PFET load results in degraded performance and heavy static power consumption due to driver-to-load contention when the driver is activated. Considering the above analysis, the staple CMOS circuit style is determined to be better optimized for subthreshold operation. This Chapter also proposed a novel circuit style capable of dynamically (within one clock cycle) switching between two operating points: 1) fast and leaky, and 2) slow and energy efficient. Although this circuit family does not provide the same performance as pedestrian CMOS technology, the circuit can reduce the overall energy budget with an exponential impact for applications which remain in idle mode for the majority of their operational lifetime.

## Chapter 4

### Subthreshold Sequential Circuits Styles

#### 4.1 Overview

Sequential elements are the cornerstone of all digital design fabrics, especially reconfigurable architecture irrespective of their granularity (fine, medium or coarse). Sequential elements can be staged as SRAM, register files, or flip-flops. Flip-flops provide storage for intermediate logic within the processing elements and SRAM cells are available as configuration memory that defines the desired function. To keep up with the growing demand for increased functionality, the size of memories in reconfigurable architectures has grown significantly. This, in turn, has resulted in increased power consumption, which is not suitable for many battery operated applications. Although many low power designs have been proposed and investigated, but none has performed a comparison of flip-flops in the subthreshold regime. In this Chapter sequential elements are examined closely in the traditional sense. [24-25] examined a wide cross-section of commonly used flip-flops with a primary focus being on performance and energy consumption. However, in addition to confining the comparisons to only conventional operating voltages, neither publication considered the differences in radiation hardness between the cells. This Chapter also discusses two individual studies performed by the author on two different technology models – 0.25  $\mu\text{m}$  (PDSOI) [26] and 45 nm (FDSOI), respectively. The study with a 0.25  $\mu\text{m}$  (PDSOI) transistor model is more comprehensive which includes both radiation hardened and non-hardened flip-flop designs. On the other hand, the study with a 45 nm (FDSOI) model is primarily focused on radiation hardened designs comprising of a novel proposed design.

## 4.2 Robust Flip-Flop Design at 0.25 $\mu\text{m}$ (PDSOI) Transistor Model

Radiation effects have become an increasingly significant concern today more than ever where contemporary technologies contain geometries that continue to shrink in to the deep submicron. The problem is no longer just restricted to space applications but now Earth electronics as well. Given the radical nature of subthreshold operation, the following includes a wider range of circuits in a more comprehensive study of flip-flops - analyzed in terms of performance, area, power, and noise immunity. Furthermore, the study is performed at both subthreshold and traditional superthreshold levels for those cases in which a circuit may be designed to be dynamically switched between these two operating points depending on the current system load requirements.

To identify the most optimal flip-flop design for subthreshold operation in terms of energy, performance, area and radiation hardness, criteria were established to guide selection of the designs. These guidelines include: (1) restricting the selection to one design from each class of flip-flops and (2) no design was included that had un-buffered inputs (e.g., inputs could not be connected directly to source / drains of transistors). Of traditional flip-flops, the final set is shown below in Figure 4.1. From [27], two designs were chosen: 1) the traditional C2MOS Flip-flop as shown in Figure 4.1-a and 2) Sense Amp Flip-flop in Figure 4.1-d. The CPHF [27] – an improved version of the HLFF discussed in [27] was included and is shown in Figure 4.1-c. The HLFF is not shown but was included in the simulation results due to the significant departure between the two variants. Several of the cell designs explored in [27] were closely related to one another and for the purposes of our study, only the stronger design of each class was selected.

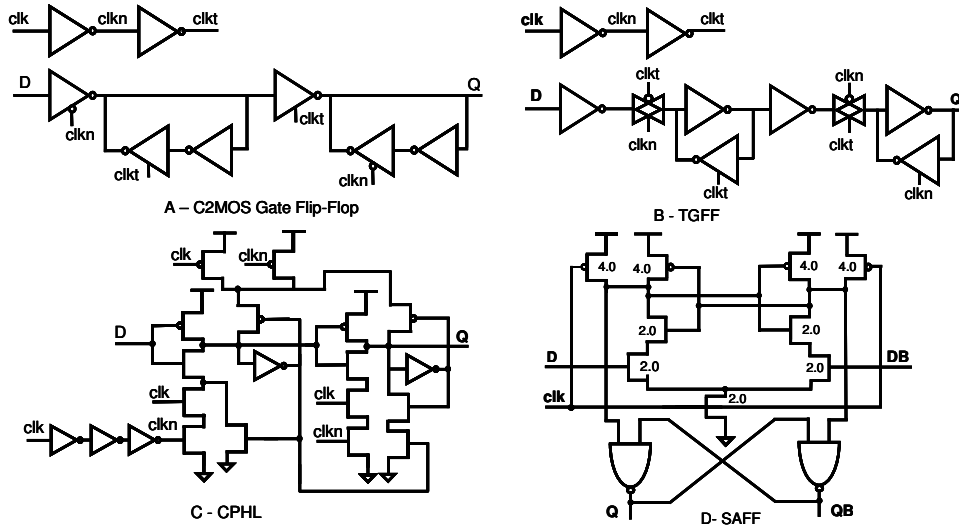


Figure 4.1 Traditional flip-flops

An example of this consolidation was the Sense Amp Flip-Flop (SAFF) and the StrongArm10 Flip-flop both of which only differed by a single transistor. The study done in [27-29] included many of the same designs but included a more appropriate Transmission Gate Flip-Flop (TGFF) in which all inputs were buffered. Consequently, this design was selected it is shown in Figure 4.1-b. Many radiation-hardened flip-flops have been described in the literature [30] and generally include redundant features. As a result, these designs tend to be larger than necessary for basic function, which results in gratuitous power consumption and performance degradation – both of which are unacceptable for traditional Earth-bound electronics. However, given the susceptibility of recent small geometry technologies to radiation effects compounded by the ultra-low voltages of subthreshold logic, these circuits were included and may eventually become mainstream sequential elements in subthreshold circuits. Figure 4.2 illustrates the two hardened designs included in the study from [31] and [32].

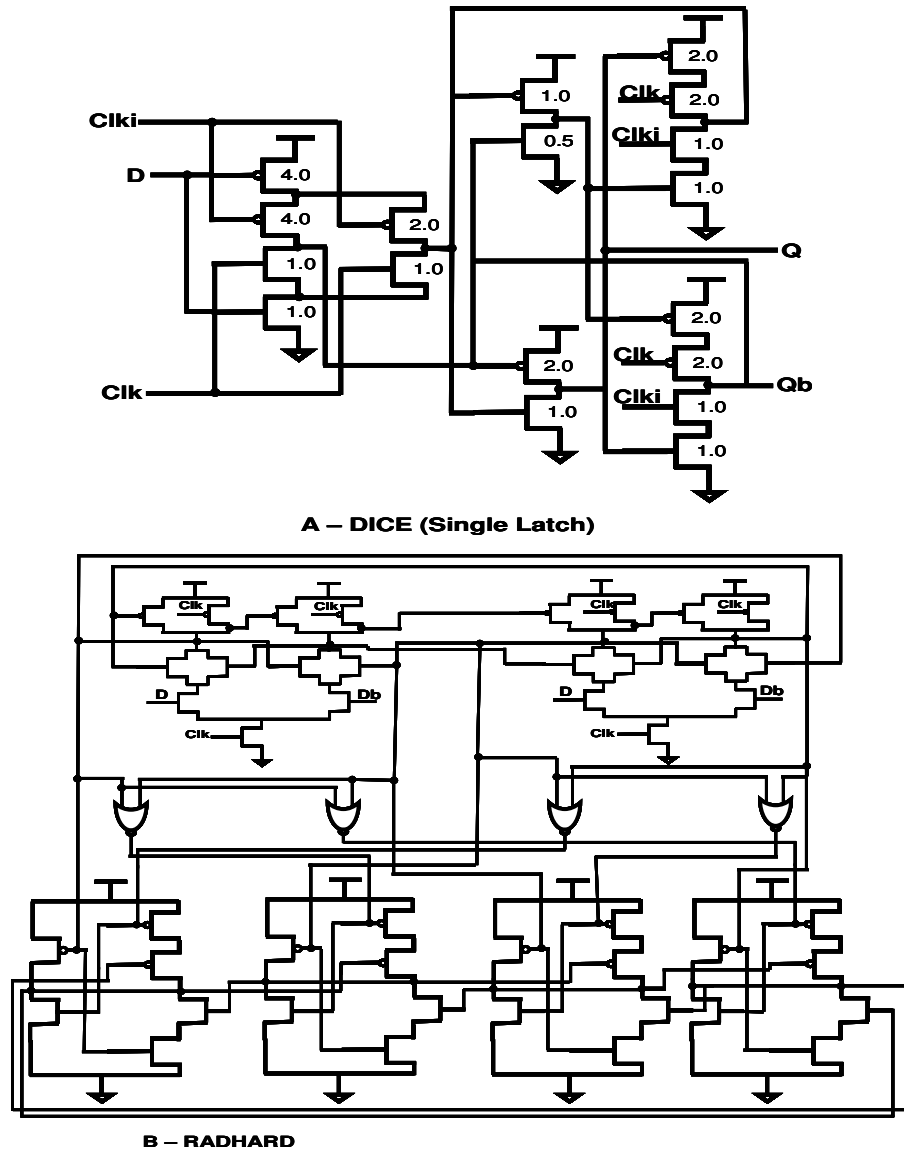


Figure 4.2 Radiation hardened flip-flops

Although the RADHARD cell is significantly larger than the DICE cell, there are inherent weaknesses in the DICE, which are described in [32]. Both were included for completeness given subthreshold logic's natural susceptibility to radiation. Yuan [29] introduced a dynamic flip-flop that provided significant area, power, and performance advantages over most other designs. However, the cell contained several floating nodes and for this reason has not enjoyed

wide use due to the lack of robustness. For the purposes of this study, keeper transistors were used on the dynamic nodes. Based on positive preliminary results of this cell with regards to energy consumption in subthreshold mode, a novel variation of this cell was included in the study in which three dynamic flip-flops were implemented as a single redundant design with a majority voting output and a muxed input. The muxed input either accepts the functional input or re-writes the majority answer back into the cells. This configuration effectively refreshes the cell during periods of inactivity and any radiation induced errors are corrected immediately. This design has been dubbed the REDDYN for REDundant DYNamic flip-flop and both the proposed cell and the base cell are illustrated in Figure 4.3.

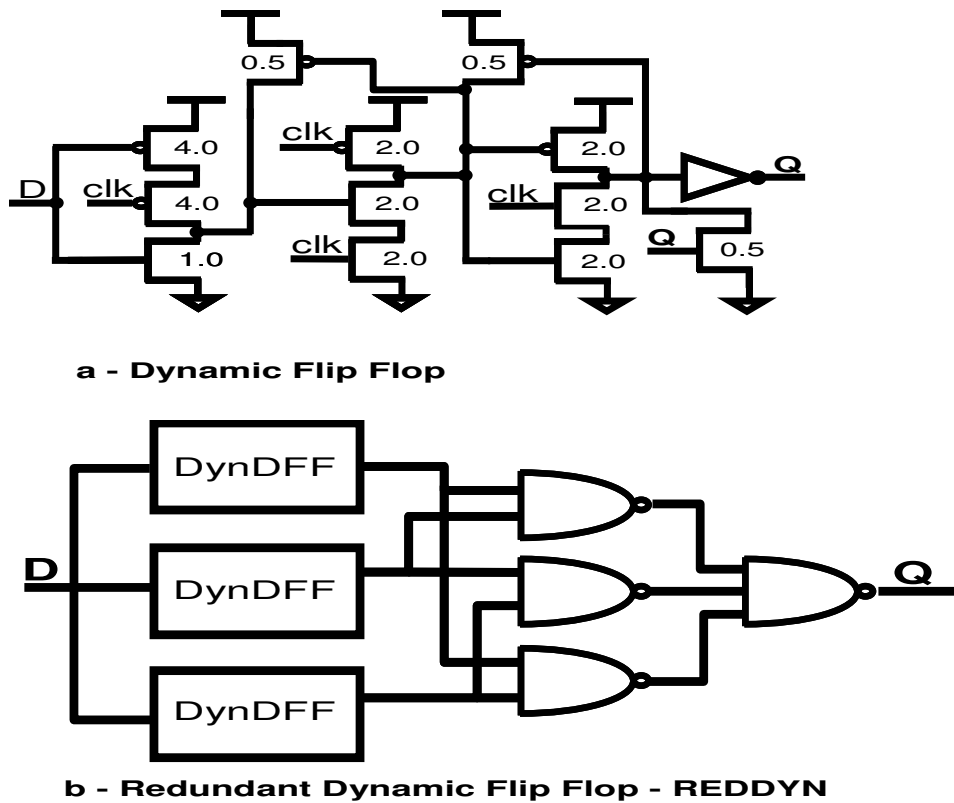


Figure 4.3 Dynamic based flip-flops

### 4.3 Simulation Results for 0.25 $\mu\text{m}$ (PDSOI) Transistor Model

To fairly assess the relative strengths and weaknesses of each flip-flop, SPICE simulations using the University of Florida's SOISPICE5 simulator were used with 0.25 $\mu\text{m}$  models of partially-depleted SOI technology. SOISPICE5 is a modified version of Berkeley SPICE 3f5 – updated for SOI technology. SOI is the natural choice for subthreshold circuits as the fabrication technique reduces parasitic junction capacitances. These capacitances increase significantly as the supply voltage is reduced and become unmanageable at subthreshold levels – both increasing the power consumption and reducing performance. Figure 4.4 illustrates the SOIPICE5 simulation of a subthreshold flip-flop; the blue plot corresponds to the input data ( $D_{\text{in}}$ ) and the red plot refers to the data out ( $D_{\text{out}}$ ) after delay with respect to the clock (gold plot).

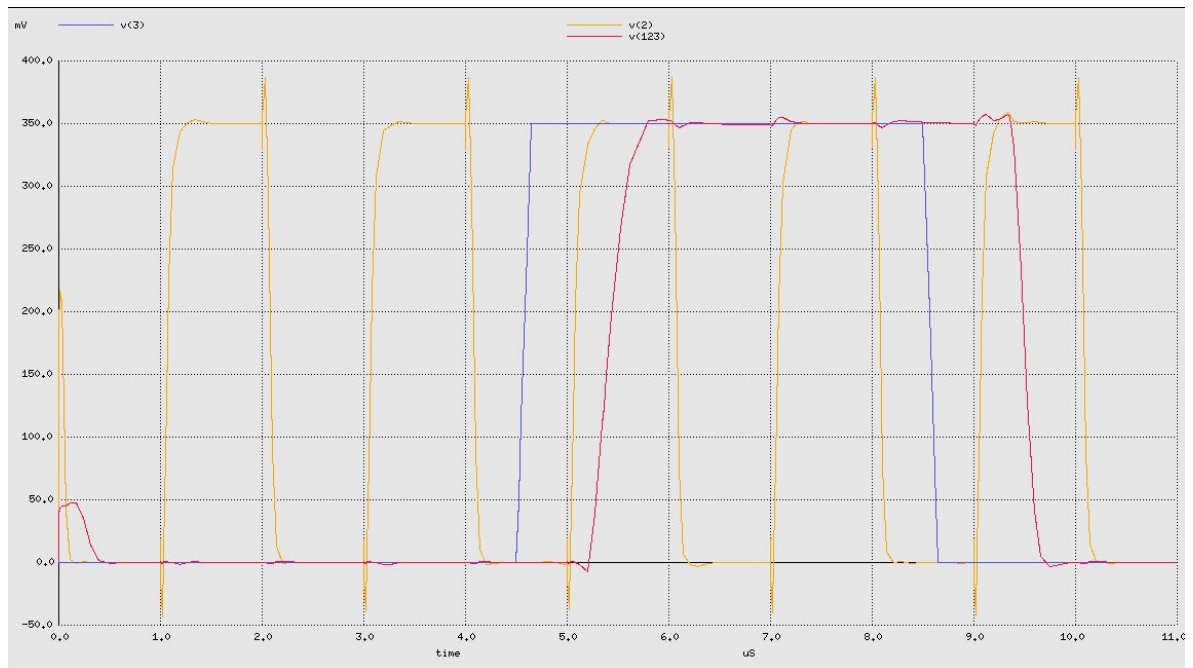


Figure 4.4 SOISPICE5 simulation of a subthreshold flip-flop

To ensure consistent and fair results, in each simulation, all inputs and outputs were buffered to provide a realistic environment and power drawn by these external components was isolated from the power measured from the flip-flop. All transistors in each of the designs were sized according to the original publication describing the flip-flop. Otherwise, simple circuit rules of thumb were employed. Further optimization is possible, but in general, the circuits worked across a wide range of voltage (0.35 to 1.8V) and were consistent for comparison purposes.

Table 4.1 describes all of the measurements made for each of the nine circuits. The first column describes the number of transistors required, which is a first order approximation of the size of the flip-flop, and the second column shows the sum of all transistor widths. This generally provides a more accurate gauge of the area. All columns from this point to the right include two sub-columns: one for the subthreshold results at 0.35V and one for the traditional operating point of 1.8V for comparison purposes. The value of the subthreshold supply voltage was chosen based on a preliminary evaluation of inverter performance and power, which is described in [16] as well the requirements for a target reconfigurable application in progress.

Launch delay was measured as the delay from the rising edge of the clock reaching 50% of  $V_{dd}$  until the switch of the data output to 50%. For launch delays (and for all of the timing parameters in general) the subthreshold values are about three orders of magnitude larger than the superthreshold values as expected, where GHz designs slow down to MHz at these low voltages. Consequently, all timing values are shown in terms of pS for 1.8V operation and nS for 0.35V operation. The setups and hold were calculated via the method described in [24], however, [29] describes this common method as overly optimistic. For the comparison purposes of this Chapter, the approach in [24] is sufficient. Regardless of the calculation method, launch



delays, setup times and hold times are not good metrics individually to compare the performance of flip-flops and composite indices were used.

**Table 4.1** 0.25  $\mu\text{m}$  (PDSOI) Comprehensive Comparison of Flip-flops

FF Type	Trans. Count	Total Width	Launch		Setup		Hold		D		R		Energy		Qcrit		E*D
			0.35v	1.8v	0.35v	1.8v	0.35v	1.8v	0.35v	1.8v	0.35v	1.8v	0.35v	1.8v	0.35v	1.8v	0.35v
			nS	pS	nS	pS	nS	pS	nS	pS	nS	pS	fJ	fJ	fC	fC	fJ*nS
TGFF	24	44	354	104	144	37	27	20	498	141	327	84	20	817	2.2	152	9960
C2MOS	24	42	263	64	138	41	35	23	401	105	228	41	17	681	9	294	6817
HLFF	20	37	319	61	23	11	286	59	342	72	33	2	28	940	4.9	200	9576
CPHL	25	40	263	58	35	30	226	54	298	88	37	4	23	857	4.9	195	6854
SAFF	23	51	482	93	130	120	429	86	612	213	53	7	20	802	2.5	505	12240
DICE	40	65	337	86	249	144	1	137	586	230	336	-51	24	1030	N/A	N/A	14064
RADHARD	64	147	739	201	132	10	521	129	871	211	218	72	74	6710	N/A	N/A	64454
DYNDF	12	50	196	32	282	90	-170	-67	478	122	366	99	11	440	2.3	480	5258
REDDYN	54	228	408	84	282	90	-170	-67	690	174	578	151	50	2010	N/A	N/A	34500

The performance of a flip-flop can be captured by combination of the setup times and launch time of the design. Both erode the time left to perform useful logical operations during a clock cycle and, thus, reduce the frequency of operations directly. As described in [24], these two values can be added to create a composite value,  $D$ , that is a more appropriate representation of the impact of the particular flip-flop on the performance of the system. Although performance is generally a secondary priority in designs that are implemented in subthreshold logic, a design with a low value of  $D$  helps to compensate for the speed penalty of ultra-low voltage operation. Our results presented in Table 4.1, show that the HLFF and CPHL flip-flops are roughly equivalent in performance as judged by the performance metric (i.e.,  $D$ ), and are approximately 20% faster than the others for both super and subthreshold. This performance can be attributed to the self-timed pulse enabling of the latches, which was effective regardless of the supply voltage.

This performance over a wide range of voltages come at the cost of increased energy consumption as these two circuits performed the worst among the non-redundant cells.

Energy was measured after a five-cycle simulation in which all data input-to-output combinations were performed (0-0, 0-1, 1-0, 1-1). All circuits were simulated with identical stimulus and dynamic power accounted for almost all energy consumption. When comparing the designs in terms of energy, the dynamic flip-flop (DYNDFF) consumed significantly less energy than any other flip-flop. This efficient operation can be attributed to low input clock capacitance and a low transistor count. Among the radiation-hardened designs, the novel REDDYN flip-flop was significantly better than the RADHARD, which is based on exploiting a collection of three of the energy-efficient dynamic cells. Radiation hardness was measured as the charge –  $Q_{crit}$  - required to flip a bit and was simulated, by inserting a current source that provided a pulse function of current for a 1 nS duration in which the magnitude could be changed to alter the total charge injected as described in [30]. In each design, susceptible nodes were identified and the current source injected sufficient charge to cause the bit to flip in both directions. The lower value in each case is reported. The results at standard voltages were as expected.

Radhard designs would not fail based on charge injected into any single node and no  $Q_{crit}$  could be established. Only in the case of two nodes being simultaneously affected could these designs become corrupted. The dynamic flip-flop did surprisingly well as a result of having added keepers on all dynamic nodes; thus, the circuit was not truly dynamic. At subthreshold levels, an interesting phenomenon was uncovered in which there was little correlation between  $Q_{crit}$  at traditional voltage levels and  $Q_{crit}$  during subthreshold operation. The dynamic and sense amp flip-flops did poorly at 0.35v, after performing well at 1.8v. The  $Q_{crit}$  level is normally linearly-related to supply voltage. However, when dropping below the threshold voltage, the

mechanisms that are relied on to maintain the voltage of the affected bit likely change order of importance. In superthreshold, circuits with strong, active transistors securing the affected node performed well, however, in subthreshold operation, the parasitic capacitance of the susceptible node played a much stronger role as the active transistors were reduced to leakage levels. The parasitic capacitances on the other hand either were voltage-level immune or increased as the voltage was reduced (i.e., diffusion capacitance). Thus, circuits with more parasitic cap on the susceptible node required relatively more charge to change the voltage sufficiently to flip the cell.

Finally, to capture energy and performance in a single metric (excluding radiation hardness), the energy delay product is included in the final column for subthreshold operation. [28] derived that minimal energy, optimal delay and supply voltage are interdependent. Therefore, once the supply voltage is chosen for a system, optimal performance and energy consumption are determined. The best combination of low power and reasonable performance among the non-radhard cells was demonstrated by the dynamic flip-flop. However, for high reliability applications, redundant cells may be required for subthreshold circuits given the degradation of  $Q_{crit}$  at these low voltages. Among these cells, the REDDYN has the best product.

#### **4.4 Proposed Novel Flip-Flop Design at 45 nm (FDSOI) Transistor Model**

This study is an extension of the previous study with 25  $\mu\text{m}$  (PDSOI) transistor models. The primary focus for this study is to analyze a proposed novel radiation hardened Flip-Flop (FF) with latest technology transistor models. For comparison purposes the study considers two previously published hardened designs: the Dual Interlocked storage Cell (DICE), an edge-triggered master-slave FF [31], and the Radhard Sense Amp Flip-Flop (RSAFF) [32]. The DICE

design operates on the principle of *dual node feedback control* to achieve upset immunity. The RSAFF consists of a sense amplifier based master stage, a middle stage with four NOR-gates and a slave stage with a modified SR-latch. RSAFF provides upset immunity due to redundant node storage and claims immunity to Transient Faults (TF) due to the delayed latching action between the SR-latch and the NOR-gate. Furthermore, [32] describes a TF Window Of Vulnerability (TFWOV) affecting the DICE cell. This study compares these two radhard FFs with the proposed FF in terms of power, performance and SEU/SET resilience at both, SuperT and SubT, voltages.

#### **4.4.1 Proposed Flip-flop Design**

The proposed SEU/SET immune FF is an edge-triggered master-slave type, consisting of two radhard latches. In turn, each latch in turn is designed as shown in Figure 4.5, by combining a MOdified Sense Amplifier half latch with a C<sup>2</sup>MOS logic switch (MOSAC). The purpose of this dual switch connection is to store redundant data at two different nodes and enable recovery after a SEU/SET. The implementation is robust at SuperT voltage levels; however, at SubT voltages – where transistors operate in the weak inversion region with reduced drive strength - the feedback path is feeble and may not overcome the charge introduced by the radiation event.

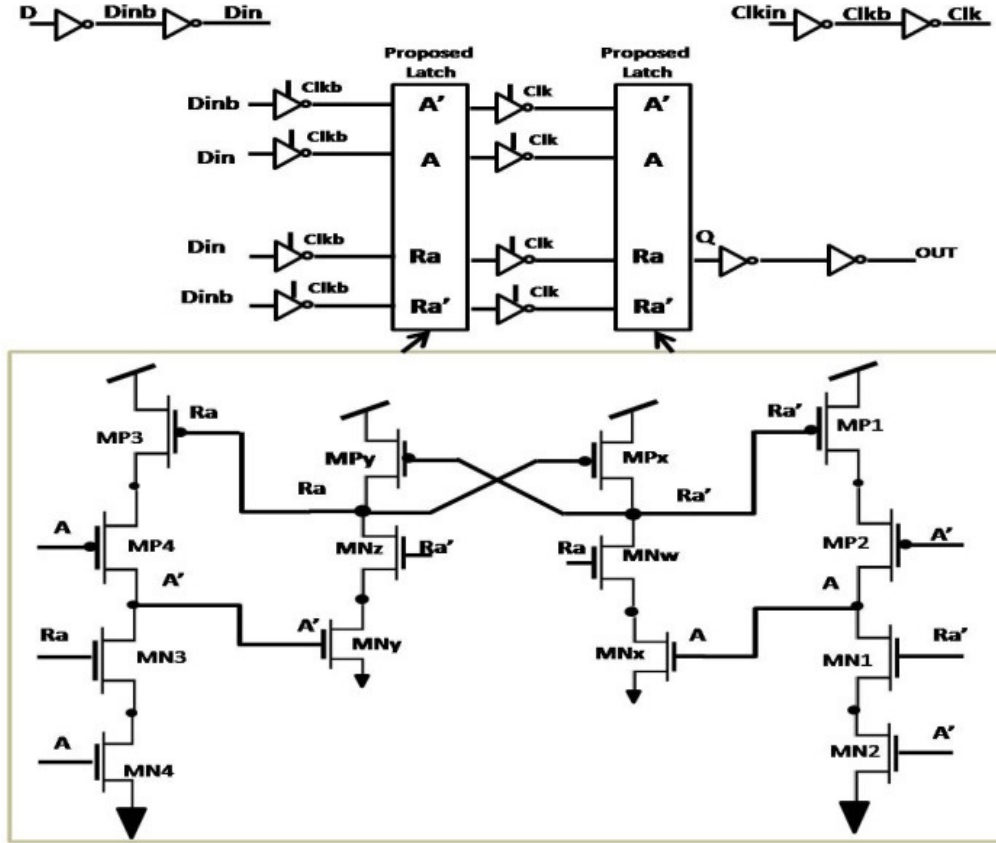


Figure 4.5 Proposed MOSAC FF

SPICE analysis was performed consistently on the DICE, RSAFF and MOSAC FFs in terms of input stimuli, duration of test, input buffering, output loading and device parameters. The SuperT voltage was selected from the device parameter reference manual for the 12SOI models. For SubT operation, a ring oscillator circuit was simulated; based on power and operational frequency, an optimized voltage of 0.35v was selected. Figure 4.6 plots the values of delay and power corresponding to their SubT voltage. In the FF analysis, for fair assessment, the power drawn from the external components was isolated from the power measurement of the FFs. Launch delay ( $T_p$ ), setup time ( $T_s$ ), and hold time ( $T_h$ ) are measured and, Delay restriction (D) and internal Race immunity (R) metrics are calculated via the method described in [24].

D and R represent traditional FF performance metrics. Radiation hardness was captured as  $Q_{crit}$  (measured as the charge required to flip a bit) and was simulated by inserting a current source between the (victim) internal node and ground. The source provided a pulse function for a 0.1nS duration with 50ps rise and 100ps fall time. In each design, all susceptible nodes were identified and a current source injected sufficient charge to test the stored charge for both polarities. For SET analysis, 25fC at 500 MHz and 2fC at 1 MHz, respectively for SuperT and SubT were injected into the master latch during the clock transition and, a TFWOV, if existing, was measured.

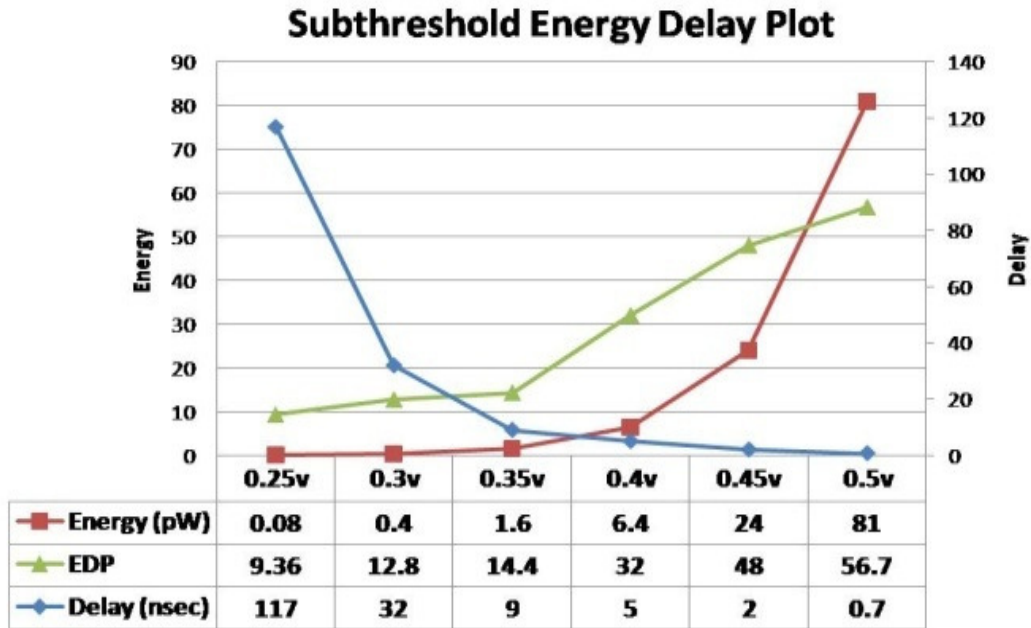


Figure 4.6 Subthreshold (45nm) ring oscillator EDP plot

**Table 4.2** 45 nm (FDSOI) Flip-flop Simulation Analysis

Clk=1MHz	SuperT Analysis (1.0v)										Clk=500MHz (25fC)
FF Type	Trans. Count	Trans. Width	Launch (ps)	Setup (ps)	Hold (ps)	D	R	Energy (pJ)	EDP	Qcrit	TFWOV (ps)
MOSAC	72	72	35	150	-	186.7	35	<b>1.99</b>	372*	NA	-
DICE	44	51	53	57.5	-	110.5	53	1.35	150	NA	<b>218</b>
RSAFF	70	69	64	6	10	73.2	54	<b>4</b>	293	NA	-
<b>* Factor increase as compared to the Sense Amp. Radhard design</b>									<b>1.26</b>		
Clk=1MHz	SubT Analysis (0.35v)										Clk=1MHz (2fC)
FF Type	Trans. Count	Tran. Width	Launch (ns)	Setup (ns)	Hold (ns)	D	R	Energy (fJ)	EDP	Qcrit (pC)	TFWOV (ns)
MOSAC	72	72	8.5	18.4	-	27.32	8.5	<b>48</b>	1312*	0.52	-
DICE	44	51	11.3	12	-	23.86	11.3	37	1036	<b>2.25</b>	<b>12</b>
RSAFF	70	69	22.5	10	1.4	33.62	21.1	<b>98</b>	3295	0.45	-
<b>* Factor decrease as compared to the Sense Amp. Radhard design</b>									<b>2.5</b>		

Table 4.2 illustrates the measured and calculated performance metrics for each FF. For SEU analysis at SuperT, with the test setup described above, all FF's were verified to be immune and, hence, indicated as Not Affected (NA). However, at SubT all FFs failed but with improved values of  $Q_{crit}$  relative to traditional, non-hardened designs. The DICE provided four times increased  $Q_{crit}$  relative to the other two hardened FFs; however, as shown in Figure 4.7 for SET analysis, the DICE cell demonstrated vulnerability as described in [32]. Conversely, Figure 4.8 depicts the proposed MOSAC with correct operation at levels of injected charge as high as 40fC. The RSAFF performed similarly well. Assuming the DICE FF is eliminated from consideration based on the TFWOV, the proposed MOSAC design offers better EDP and  $Q_{crit}$  values at SubT when compared to the RSAFF. At SuperT levels, the RSAFF had marginally better EDP and both designs were radhard. Depending on the performance requirements and the operational duty cycle of the application, one of the two designs will be optimal. Applications requiring short, intermittent periods of medium performance followed by long idle periods will potentially stand

to benefit from the proposed MOSAC FF. In applications where the  $Q_{crit}$  values in SubT are inadequate, additional measures like TMR or DR may be necessary.

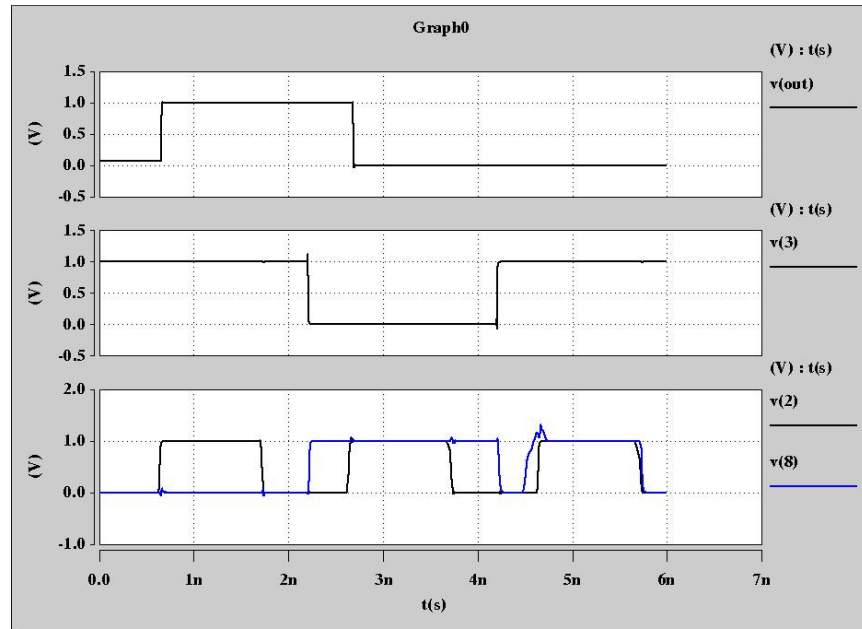


Figure 4.7 DICE FF Transient Fault Simulations at SuperT

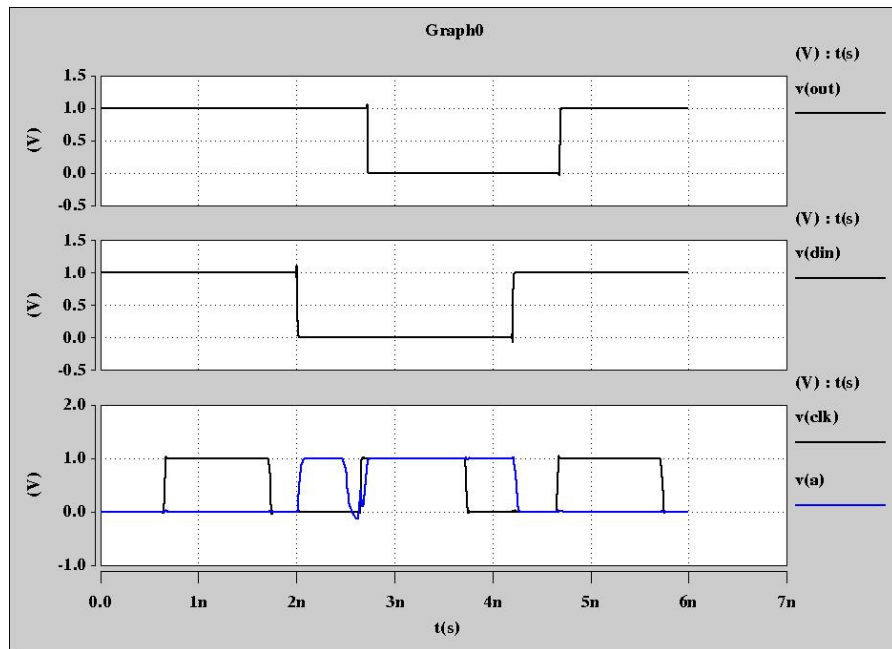


Figure 4.7 MOSAC FF Transient Fault Simulations at SuperT



## 4.5 Conclusion

This Chapter presented two sets of flip-flop analysis experiments that primarily differ in the transistor implementation technology and Simulator Program with Integrated Circuit Emphasis (SPICE) tool. The first set of flip-flop experiments are based on 0.25 $\mu$ m Partially-Depleted SOI transistor model realized using the University of Florida SPICE tool. In this analysis a group of nine flip-flops that represent a wide range of designs, traditional non-radiation hardened and radiation hardened, were compared. The study highlighted the strengths and weaknesses of each flip-flop design in the context of subthreshold operation, which has never been reported previously. The study observed that for all non-redundant cells, a susceptibility to single event upsets is an obvious weakness and the use of redundancy at some level will likely be required for applications in which data integrity is a priority. A new redundant cell was proposed that takes advantage of the reasonable performance of a base dynamic flip-flop with added keepers.

The second set of flip-flop experiments are based on 45nm Fully-Depleted SOI transistor models from IBM realized using the HSPICE tool. In this analysis radiation hardened flip-flop designs were compared for power, performance and Single Event Transient robustness. A novel flip-flop design, Modified Sense Amplifier C<sup>2</sup>MOS Flip-Flop (MOSAC FF), was proposed, it enables SEU/SET resilience with 50% reduced power and 20% better SET Qcrit values compared to other designs in the analysis.

## **Chapter 5**

### **Subthreshold Voltage Boost and Logic Interface Circuits**

#### **5.1 Overview**

One challenge at low voltages is that circuits not only operate significantly slower, but may also need modification to function correctly. For instance, circuits that rely on NFET-only pass transistors struggle to drive high voltages and are crippled in the subthreshold region. Also, reconfigurable circuits require boosted voltages for a variety of reasons including the need to overdrive pass gates for improved performance in the switching fabric, programming non-volatile configuration memory and power critical-path voltage islands to improve performance [34-35]. A unique solution to all the above-mentioned issues is a charge pump circuit, which basically pumps charge in an incremental fashion to subsequent stages to produce voltages higher than the regular supply voltages. The charge pumps examined in this Chapter are confined to staged charge pumps that allow for the output voltage to be set by the number of stages included. To optimize power and performance the SRR design can be split into dual voltage islands. Level shifters are required to interface low-voltage to high-voltage circuits. This study compares a variety of existing level shifters as well as several proposed level shifters in the context of up-converting subthreshold signals to superthreshold levels.

#### **5.2 Charge Pump Designs**

To identify the most optimal charge pump architecture, charge pumps were compared across a wide variety of criteria including: energy consumption, ramp-up time, load current, and

area. These guidelines include restricting the selection of evaluated designs to staged architectures. Staged designs allow for the output voltage to be increased incrementally and, therefore, adjusted in a static design sense. Consequently, pumps that are only capable of doubling the voltage were not considered due to the lack of flexibility. Additionally, due to our target application having very low output current requirements, output current was relegated to the lowest priority among the design criteria. The primary application includes overdriving the gates of NFET pass transistors to improve the performance of the switching fabric in a subthreshold FPGA; gate oxide leakage at a subthreshold supply voltage level provides the only current drain.

Of the staged charge pump architectures, the first circuit and the one that is the basis for those that followed is the Dickson Charge Pump (DCP) [36], which is shown in Figure 5.1. The first transistor (far left) in the figure provides an initialization voltage on  $V_1$  and clamps the voltage to a minimum of  $V_{dd} - V_t$ .

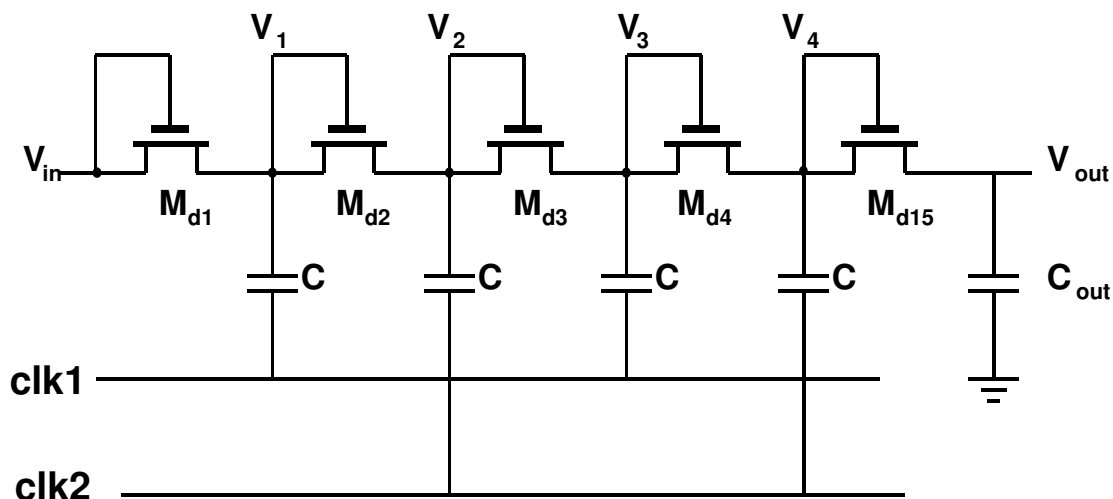


Figure 5.1 Dickson Charge Pump

The circuit runs with two non-overlapping clocks that capacitively-couple into alternating nodes between each of a line of NFET transistors. At the left side of each transistor (source/drain), as the corresponding clock rises - the node is coupled up proportionally by the amount described in Equation 1 for the no load condition. The right side of the transistor – initially at a lower voltage, is then pulled up to within a  $V_t$  drop from the drain. The rise of the second clock couples the right side above the left side and the roles of source and drain are reversed. As the gate and source of the transistor are now at the same potential ( $V_{gs} = 0$ ) the transistor no longer conducts. This process continues down the line of transistors with each stage increasing the voltage by  $\Delta V - V_t$ . Bear in mind that the  $V_t$  value at each subsequent stage increases due to the body effect. This is the result of the source of each transistor increasing relative to the body voltage in each subsequent stage.

$$\Delta V = \frac{C_c}{(C_c + C_p) * V_{clk}} \quad (\text{Equation. 1})$$

Each of the remaining architectures is based on the DCP and each is an attempt to solve the threshold voltage degradation. By doing so, higher output voltages can be obtained with fewer stages and the circuits can become more effective in terms of power and charge transfer. Moreover, the frequency of operation can be increased by eliminating the threshold degradation, and thus directly improving output load capability. However, having smaller voltage increments per stage provides more flexibility in terms of the resolution of the output voltage. Although more stages are required for a given output voltage, the DCP has fewer transistors per stage so a comparison is difficult to render. Each circuit has at least one pumping capacitor at each stage

as well and, depending on the technology (i.e. double poly process availability), the capacitor(s) may dominate the area and power efficiency.

To be comprehensive and to attempt to uncover any unobvious advantages of charge pump over the others, the study analyzed all of the selected charge pumps in the context of subthreshold operation. The Static Charge Transfer Switch charge pump (SCTS) and the improved Dynamic Charge Transfer Switch (DCTS) pump were proposed by [37]. The SCTS over-drives the NFET gate at each stage by using a subsequent stage's output voltage, thus, eliminating the threshold drop across each transistor. The DCTS (a.k.a. NCP-2) is an improved version that prevents the reverse leakage problem in the SCTS by overdriving the gates only when charge is expected to be transferred to the next stage.

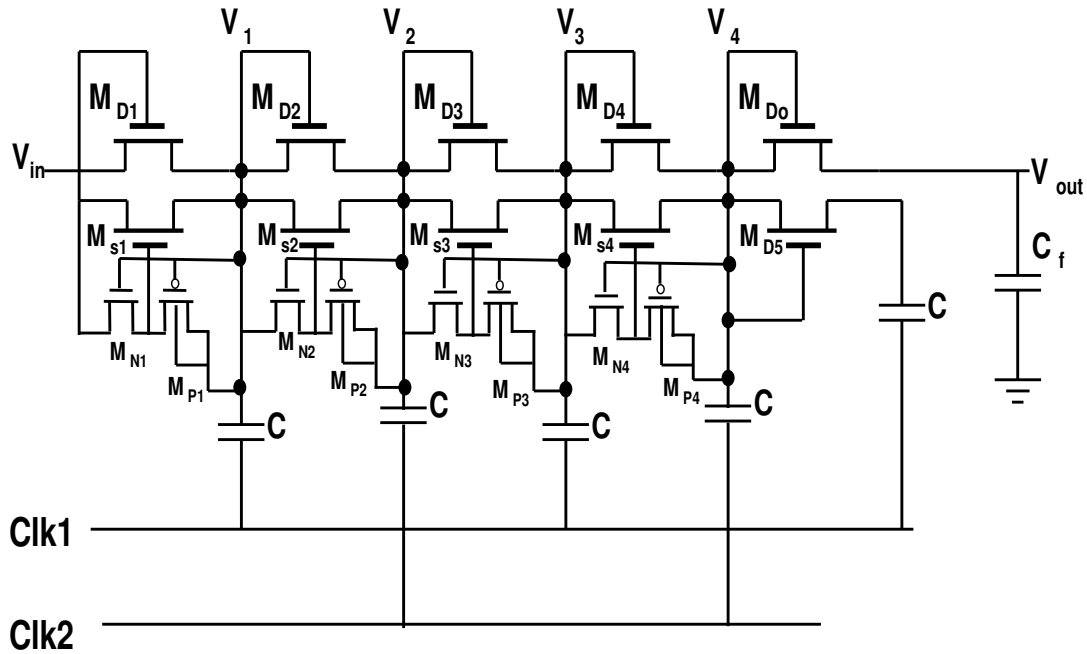


Figure 5.2 DCTS 2-phase architectures

The problem with both of these circuits is that the last stage has no access to an overdriven voltage and the voltage degradation continues to affect the output voltage. A third version was proposed using an overdriven clock on the final stage to overcome this deficiency. However, an overdriven clock generally defeats the purpose of the additional auxiliary transistors in the circuit. Figures 5.2 illustrate the charge pump simulated in this analysis, which does not include clock overdriving. [38-39] proposed a staged charge pump with cross-coupled NFETs with charge transfer through a double PFET configuration as shown in Figure 5.3.

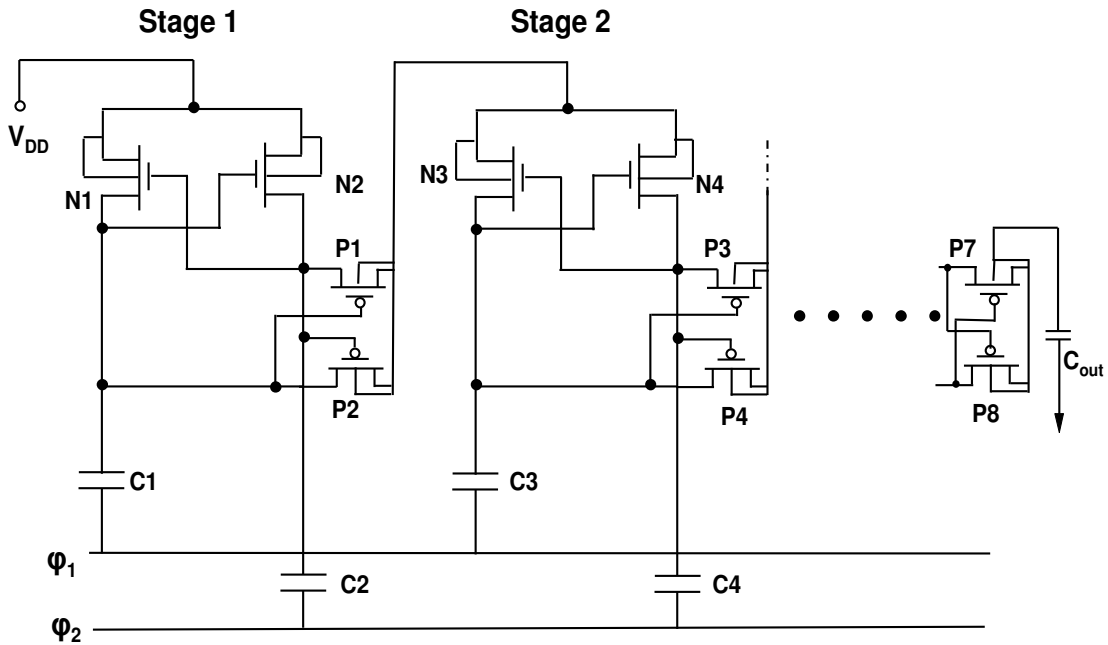


Figure 5.3 DCTS 2-phase architecture with cross-coupled NMOS

In this pump, the P and N wells for each stage separately are tied to the input voltage. This configuration maintained all voltages within limits for gate oxide reliability and also prevented the body effect from increasing the threshold voltage of each stage. This approach requires that

the wells for each stage be isolated from one another with the use of triple-well CMOS technology. The parasitic capacitance of the wells does not affect the coupling nodes and helps eliminate the influence of the body-effect. This pump requires two capacitors per stage and the dual nature of each stage makes this pump the most effective in terms of load current. Each of the previous charge pumps used two clocks to perform the ramp up.

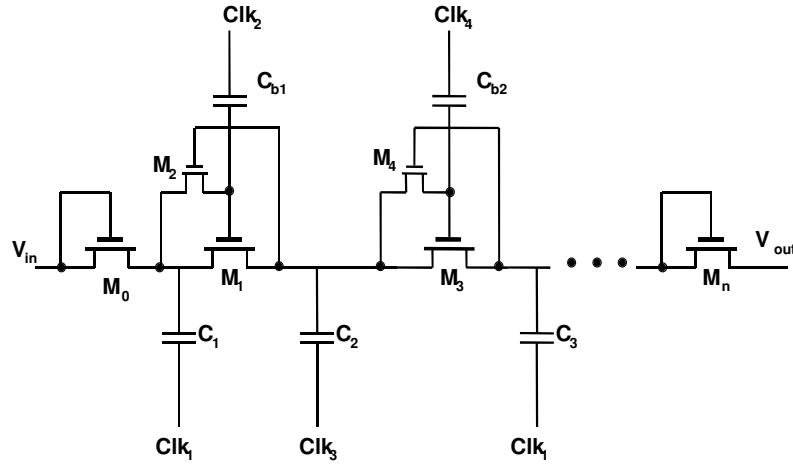


Figure 5.4 Four Phase Architecture

[40] proposed a four-phase clocking scheme to eliminate the threshold voltage degradation at each but the last stage of the pump. By adding one transistor and one capacitor to each stage, subsequent stages can feed the gates of previous stages. The architecture is shown in Figure 5.4. The four-phased clocking required for this circuit is relatively complicated and would be difficult to implement from a single clock across a wide range of supply voltages and temperatures. Furthermore, the final stage is still affected by degradation in a similar manner as the DCTS. Finally, the DCP threshold voltage degradation also can be alleviated by overdriving the clock signals, which includes the additional benefit of increased capacitive coupling. Figure 5.5

### 5.3. Novel SOI Subthreshold Charge Pump

The diagram shows a differential pair of NMOS and PMOS transistors. The left side has NMOS  $N_{1b}$  and PMOS  $P_2$  (with label  $N_{2b}$  above it). The right side has NMOS  $N_1$  and PMOS  $P_1$  (with label  $N_2$  above it). The gates of  $P_2$  and  $P_1$  are connected to a common bias voltage  $V_{ob}$ . The gates of  $N_{1b}$  and  $N_1$  are connected to a common bias voltage  $V_{o_b}$ . The drains of  $P_2$  and  $N_1$  are connected to a common supply rail. The drains of  $P_1$  and  $N_{1b}$  are connected to a common supply rail. The sources of  $P_2$  and  $N_1$  are connected to a common source rail. The sources of  $P_1$  and  $N_{1b}$  are connected to a common source rail. Two capacitors,  $C_1$  and  $C_{1b}$ , are connected between the gates of the NMOS transistors:  $C_1$  is between the gate of  $N_1$  and the gate of  $N_{1b}$ , and  $C_{1b}$  is between the gate of  $N_2$  and the gate of  $N_{2b}$ . The outputs are labeled  $Out_b$  and  $Out$ .

53



Exploiting the ability to control individual body voltages in SOI, Dual Threshold CMOS (DTCMOS) was proposed where the gate input to a CMOS combinatorial circuit is also tied to the body of the associated transistor [36]. An example of this is where the gates and bodies of the two transistors – one NFET and one PFET - in an inverter are all tied together. When asserted high, the input activates the NFET while simultaneously raising the body voltage. This, in turn, lowers the threshold voltage, temporarily increasing pull-down strength. At the same time, the PFET is turned off and the raised body voltage results in less leakage. Conversely, when the input is held low, the *on* PFET is strengthened and the *off* NFET weakened. Manipulating the body voltage of any critical transistors can be applied not only to CMOS combinatorial circuits but to any circuit style. Originally, this configuration was not plausible for super-threshold operation because raising the body of the NFET above the value of a diode drop caused the body-to-source junction to forward bias; however at subthreshold levels this concern is eliminated. The concept can also work for bulk triple-well technology as well, but it is difficult based on the much larger capacitances of the wells in comparison to the floating body of a SOI transistor.

Given that the clock booster as well as the first stage in the DCP are operating at subthreshold levels with a subthreshold input voltage, we focused on improving these sections. Inspired by the concepts in [41], the clock booster circuit was improved by dynamically adjusting the threshold voltage in the top-stack NFET (N2b and N2) as well as the PFETs (P1 and P2) [42]. When *on*, the transistors have a reduced  $V_t$  to improve the boost and when *off*, the transistors have an increased  $V_t$  for lower leakage. The circuit, Active Body Voltage Doubler (ABVD), which is included in the comparison, is illustrated in Figure 5.6 with four additional capacitors.

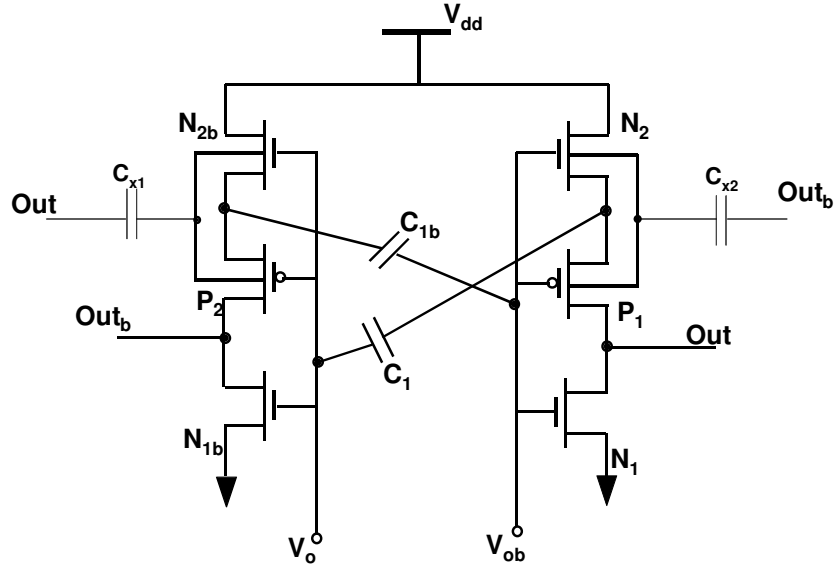


Figure 5.6 Proposed Active Body Voltage Doubler

#### 5.4. Simulation Results

To fairly assess the relative strengths and weaknesses of each charge pumps, SPICE simulations using the University of Florida's SOISPICE5 simulator were performed with  $0.25\mu$  models of partially-depleted SOI technology. SOISPICE5 is a modified version of Berkeley SPICE 3f5 – updated for SOI technology. As described in the earlier section, SOI is the natural choice for subthreshold circuits.

As capacitors will likely dominate the floor plan for direct comparison purposes, all pumps were limited to 1 pF per pump stage. This may result in an advantage for the clock-booster circuits which have additional capacitors in the voltage doubler. Figure 5.7 illustrates the SOISPICE5 simulation results for the DCP circuit for subthreshold voltage level (350mV). The

red plot is for the input voltage ( $V_{dd}=350\text{mv}$ ) and the blue plot is for the boosted output voltage ( $V_{out}\approx 1.42\text{v}$ ).

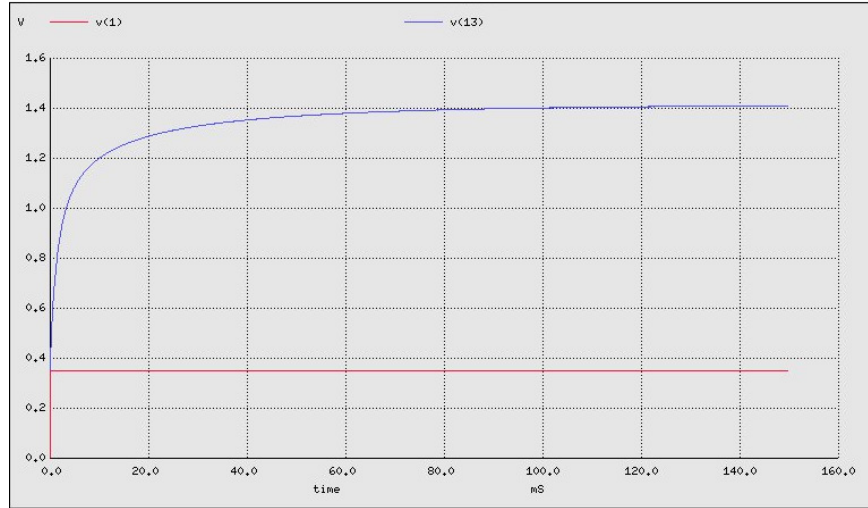


Figure 5.7 SOISPICE5 Simulations for DCP Charge Pump

Each of the pumps described in the previous sections were simulated; the results are shown in Figure 5.8. In this figure, the output voltage was measured for a range of input voltages – including both sub and super threshold levels. The Figure 5.8 excludes the proposed ABVD design because the body control can only be possible at subthreshold levels. The load was assumed to be wholly capacitive with no DC current; this most closely matches our application requirement of overdriving gates on NFET pass transistors, which is necessary for improved performance in the switching fabric of a reconfigurable architecture. Of all of the existing designs, the clock over-driven Dickson charge pump provided the highest voltage level for a given input voltage in the context of a subthreshold operation and our proposed pump provided a significant additional improvement of 310mV. In subthreshold operation, the clock booster and

first stage operate with supply voltages exclusively below the threshold of the transistor and subsequent stages ascend to superthreshold levels. Consequently, our improvement bolsters the performance of one low voltage section, the clock booster, by temporarily lowering the threshold voltage of the pull-up NFET transistor. This results in an increase in the voltage of the boosted clock.

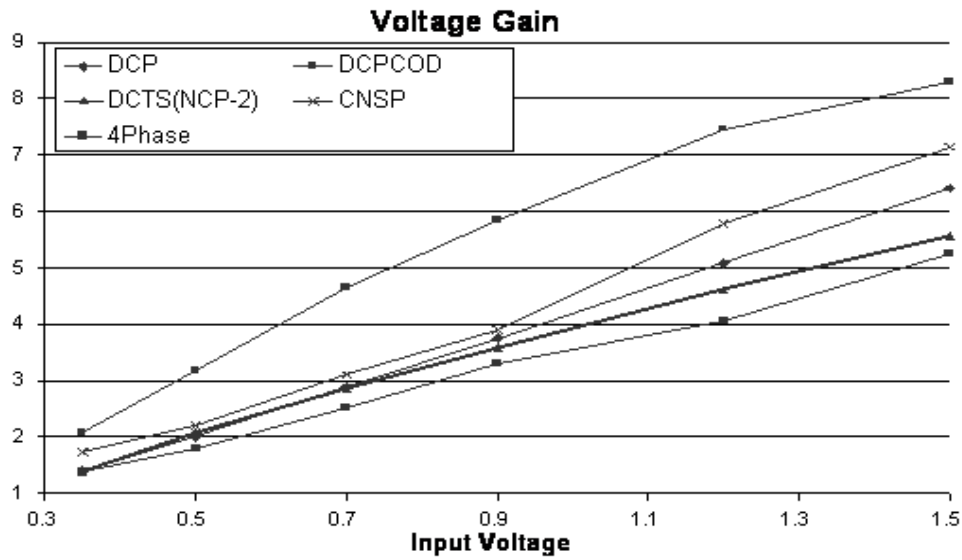


Figure 5.8 Charge pump voltage gain comparisons

Additionally, the larger clock swing results in larger  $\Delta V$ s at each stage of the pump and thus compounds the improvement in the output voltage. To broaden the analysis, several current loads were included at the subthreshold level (0.35v) and the results are shown in Figure 5.8. In the case of a DC current load, the proposed DCP with ABVD pump provides the most drive capability in subthreshold and therefore is well suited for a wide variety of ultra-low power applications.

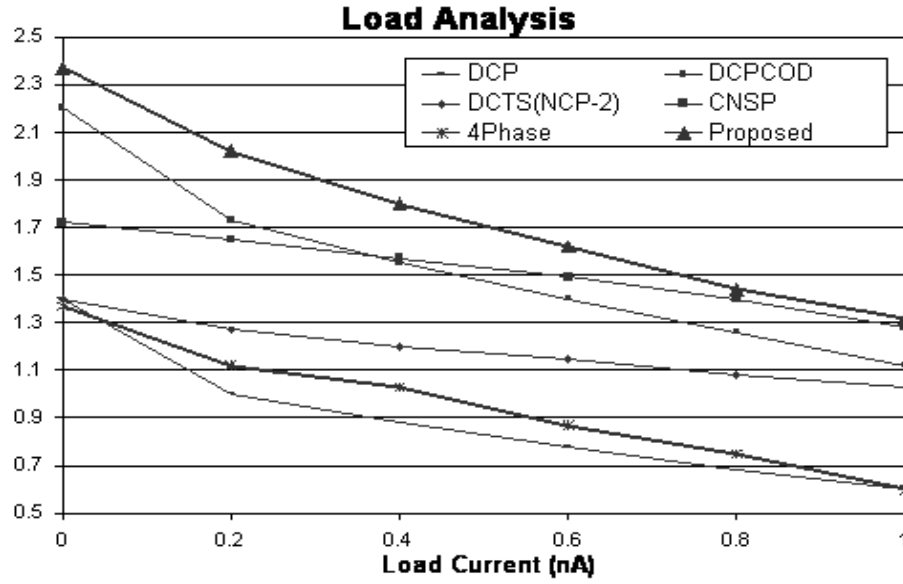


Figure 5.9 Charge pump output voltage Vs load current comparisons

Other aspects of the pumps that were characterized in this study include the energy required to initialize the output voltage, the total time for ramp-up and the output voltage ripple. Table 5.1 shows these values as well as the output voltage while running at subthreshold levels for the no load condition. The Table 5.1 illustrates how the DCP provides the lowest power while the clock-boosted circuits are the most power-hungry. The clock overdriven circuits included the voltage doublers shown in Figures 5.5 and 5.6, which involve high frequency activity with a double voltage swing, all of which contribute heavily to power consumption. With regard to initialization time, all pumps operating in the subthreshold regime required milliseconds to establish a final voltage and any application that employs a charge pump at these ultra-low levels will need to accommodate this constraint. Finally, the ripple at the output voltage was insignificant in all cases of this study, however, considering the reduction in noise margin while running at subthreshold levels, noise and ripple must be considered carefully in designs.

Table 5.1 Charge pump performance comparison

	Output Voltage (V)	Startup Energy (pJ)	Ripple (mV)	Ramp-up Time (mS)
<b>DCP</b>	1.40	0.5	0.3	30
<b>DCPCOD</b>	2.06	272.0	1.0	9
<b>DCTS(NCP-2)</b>	1.40	0.3	0.3	7
<b>CNSP</b>	1.72	0.2	0.1	3
<b>4Phase</b>	1.37	0.4	0.2	31
<b>ABVD-DCP<sub>proposed</sub></b>	2.37	462.0	1.0	21

## 5.6 Level Shifters to Interface Sub and Super Threshold Logic Cells

Traditionally, level shifters were employed exclusively to allow chip core signals to be transmitted to the outside world through the pad ring, which often operated at a different voltage to maintain compatibility with older technology used at the system level. More recently, with the increased use of voltage islands within chips, functional units are being operated at different voltages allowing the core processor to execute the critical algorithm while running at a higher voltage ( $V_{ddH}$ ), thus, maximizing the performance. Simultaneously, all other non-critical circuits operate at a lower voltage ( $V_{ddL}$ ) to improve power efficiency. [43] reported that optimized multi- $V_{dd}$  with multi- $V_{th}$  designs provide a dramatic dynamic power reduction of 40-50% as compared to the original single  $V_{dd}$  design. In order to effectively interface critical cells at higher voltage, with non-critical cells at lower voltage, level shifters are required to fully turn off the PFET (P channel Field Effect Transistor) of the driven gate and, in some cases, to ensure that no gate oxide voltage exceeds the reliability limits set by the technology node. To date, no report

has been published that examines the use of level shifters to convert ultra-low subthreshold signals to higher, traditional super-threshold voltages.

This report compares a wide variety of level shifters for the extreme case of shifting between subT and superT levels and proposes several novel level shifters that are optimized for subthreshold operation. The final aim of this research is to provide an ultra-low-power radhard, reconfigurable chip, which is optimized by virtue of a hybrid subT and superT power supply scheme. The next section presents three previously reported level shifters, while the study proposes three novel level shifters optimized for subthreshold input levels.

## **5.7 Existing Level Shifters**

In this study the existing CMOS level shifters are broadly classified into two main categories: 1) Dual Supply Level Shifters (DSLS) and 2) Single Supply Level Shifters (SSLS). The advantages of SSLS over DSLS has been illustrated in [44] on the grounds of pin count, congestion in supply routing, complexity and overall system cost. SSLS circuits do not require access to the lower supply voltage other than the signal to be converted. However, in our target applications, all level shifters have unhindered access to both supply voltages without increasing routing congestion. In the spirit of maintaining comprehensiveness, one SSLS circuit was included in this comparison. Figures 5.10-a and 5.10-b illustrate the two traditional DSLS circuits that were evaluated in this analysis [45].

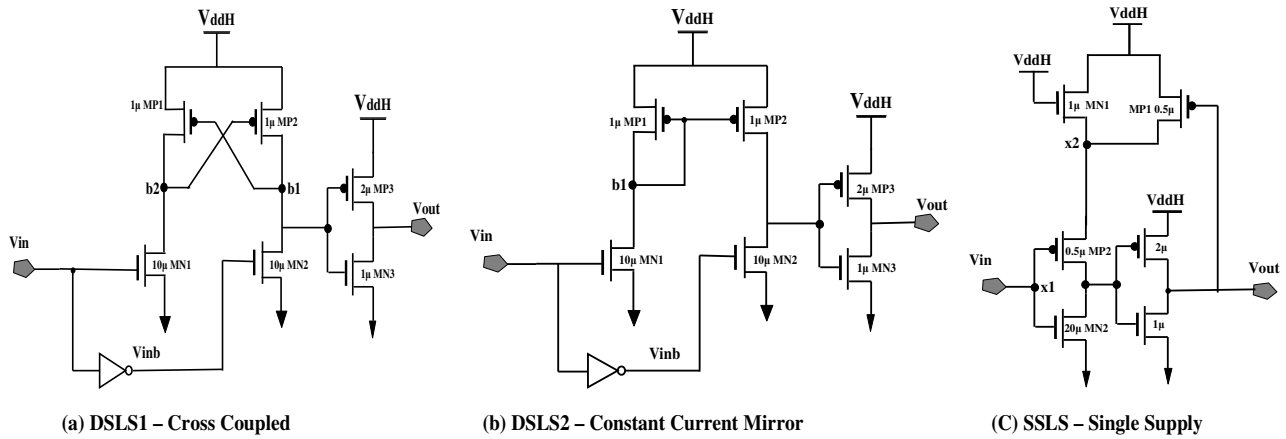


Figure 5.10 Conventional level shifters

Figure 1-a describes the traditional DSLS1, which is a differential cascode voltage switched logic gate, which uses a cross-coupled PMOS half latch operating at the higher supply voltage. The low input voltage,  $V_{in}$ , is shifted-up due to the positive feedback action of the cross-coupled transistors MP1 and MP2 to  $V_{ddH}$ . When  $V_{in}$  is high, MN1 and MP2 are activated and, thus, raise the voltage at node b1 to  $V_{ddH}$ , which results in the output being driven to low. Subsequently, if  $V_{in}$  asserts low, MN2 and MP1 are activated thereby raising the output voltage to  $V_{ddH}$ . The pull down transistors MN1 and MN2 are required to be much larger size than MP1 and MP2 because each has to overcome the PMOS latch action driven with a higher supply voltage. DSLS1 has the advantage of a simple design and is well suited for higher core voltages. However, for low input supply voltages, the performance degrades as the NFET devices (operating essentially with leakage in subthreshold operation) are incapable of flipping the superT PFET half-latch. In fact, for typical transistor sizing, this circuit may not operate at all when converting subthreshold signals.

The DSLS2 in Figure 5.10-b provides improved performance and stable current-driving capability compared to DSLS1 by replacing the PFET half-latch with a PFET current mirror.



This design is well suited for wide voltage range conversion with regards to performance; however, DSLS2 suffers from increased power consumption resulting from the leakage path formed by either PFET in the current mirror and one of the NFET pull-down devices, one of which is always on in a static sense. The final traditional level shifter evaluated in this comparison is the SSLS, which is shown in Figure 5.10-c and it only requires access to the higher supply voltage, thus potentially easing routing congestion for cross supply signals. This circuit has high performance but can have significant leakage currents if there is a wide difference in supply voltages. Other level shifters have been reported, which include thick gate oxide transistors, ideal capacitors and low-threshold transistors [46-47] – all of which are not necessary in the context of subthreshold operation and increase the cost of the standard CMOS process. Consequently, these circuits have been excluded from this analysis. Furthermore, other level shifters [48] have been reported that have pass-gate inputs; these also were excluded as they are susceptible to above- $V_{dd}$  and below-ground noise as well as being a source of potential reverse stage leakage.

For each of the conventional level shifters described above, an incremental improvement is possible by adding a voltage doubler at the input of the level shifter as shown in Figure 5.11. Voltage doublers have been proposed in [49] that bootstrap the true and complement signal to almost double the low supply voltage. For subthreshold circuits, this means that the outputs are typically raised to superthreshold levels (i.e., 0.35V doubled to 0.70V) and this has an exponential impact on the drive strengths of the NFET devices used subsequently in the level shifter, – significantly improving performance.

Unfortunately, this improvement comes with the price of a large area increase due to the addition of the voltage doubler and, which includes two large transistors used as MOS capacitors necessary for boot strapping the input. Bootstrapping circuits are generally not used in industrial designs because nodes are at times left floating and are therefore susceptible to noise and are difficult to test. The susceptibility to noise arises because one of the two capacitors always will have one node floating above the supply voltage and as the node is not driven directly and can be discharged by a variety of sources (i.e. radiation, capacitive coupling, leakage, etc). Furthermore, the capacitors lose charge over extended periods of inactivity and consequently, can exhibit varying propagation delay through the circuits based on the recent switching history of the input. Although these boosters have significant drawbacks, their advantages in the context of subthreshold circuits warranted their inclusion in the study.

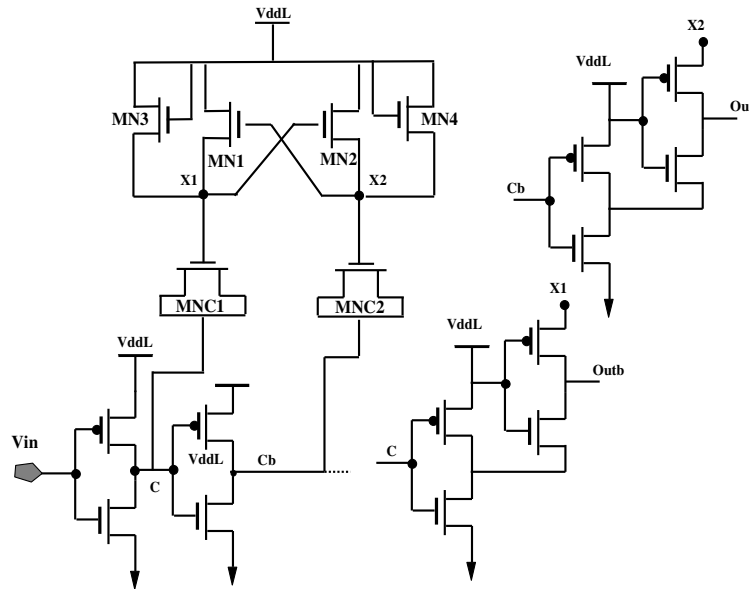


Figure 5.11 Voltage Doubler

## 5.8 Proposed Level Shifters

The major obstacle for the existing level shifters operating in subthreshold is that the high supply voltage pull-up network (i.e., current mirror or half-latch) must be overcome by the pull-down network, which is under driven with anemic subthreshold level signals [52]. To improve the performance of these transistors, either the gate voltage must be increased, or the body voltage must be increased to reduce the threshold of the transistor (via the body effect) – either case results in an improvement in the drive strength of the pull-down network relative to the pull-up network. By implementing the subthreshold circuits in SOI CMOS (Silicon-On-Insulator Complementary Metal Oxide Semiconductor) technology, the large diffusion capacitances associated with the low-voltage reverse biasing of the source and drain junctions are virtually eliminated. As a result of this natural technology selection, interesting transistor configurations are now, possible particularly in light of subthreshold operation. [20, 50-51] describe an interesting circuit style called Dynamic Threshold CMOS, which involves tying the gate of the transistor to the body and, thus, dynamically adjusting the threshold of the transistor depending on if the transistor was in cut-off or saturation. The proposed connection is not normally possible in the superthreshold regime as the source-to-body junction would forward bias and result in significant leakage current. However, in subthreshold, this connection is not only possible but can improve performance by reducing the threshold of the transistor when on or reduce leakage by increasing the threshold when off.

In each of the three existing circuits in this study, a novel version has been created in which key transistors were body-tied in order to improve the ratio of NFET-to-PFET networks. Again, these proposed configurations are not suitable for normal level shifting (e.g. 0.7V to

1.8V) because of the possibility of forward biasing the body to source diode exists. Note however, that this topology stand to provide significant improvement for those designs operating below the threshold of the transistors on the low side supply. All other transistors have floating bodies and the bodies are inaccessible. Body ties add some area for the additional connection and will result in some additional energy consumption through the body-to-source diode even though the input will never exceed the forward bias potential for the diode.

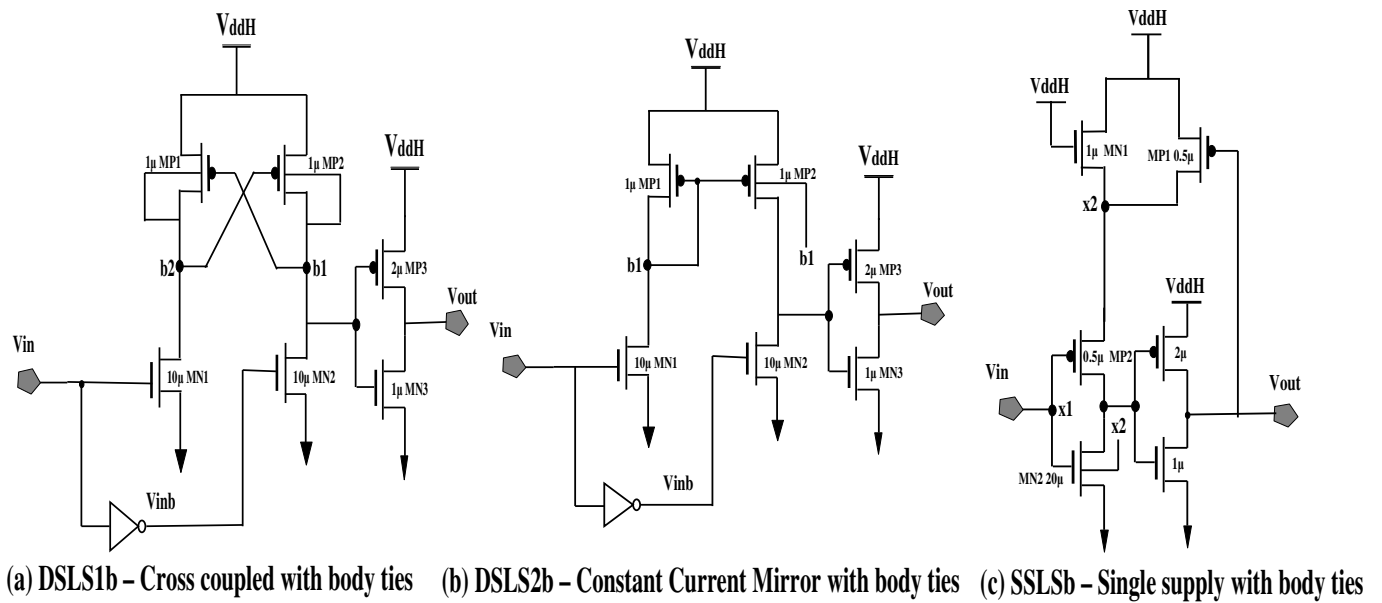


Figure 5.12 Proposed level shifters with body ties

Figure 5.12.a illustrates the enhanced version of the DSLS1 in which the superthreshold PFET devices have been body contacted to reduce the drive strength of the half latch and, thus, improve performance. This configuration has an indirect pathway between power and ground as the source-to-body diode of either PFET can be forward biased; however, the subthreshold NFETs in the path mitigate the potential leakage. Figure 5.12-b illustrates the improved version of DSLS2 in which the output-side PFET of the current mirror is body-tied with the gate voltage.

This improves the drive strength of this transistor when the device is expected to be in saturation, but reduces the strength when the device is turned off, allowing the subthreshold NFET to overcome the PFET and switch the output. Only a single transistor in this configuration requires a body contact; accordingly less leakage should occur as the current mirror gate voltage (used for the body connection) does not swing rail-to-rail in this analog circuit, thus, never forward biasing the source-to-body diode of the PFET. Finally, Figure 5.12-c illustrates the single supply level shifter with a subthreshold-optimized improvement. In this circuit the single subthreshold-driven NFET is body-tied to a superthreshold level that is either at the high  $V_{dd}$  level or a threshold drop below the value. In this case, the threshold voltage of the NFET is dramatically reduced (i.e., from 0.5V to 0.1V) and, thus, the subthreshold input is capable of driving the device into full saturation. Consequently, the subthreshold input can switch the latching mechanism. In this configuration there is always a direct short between power and ground through the source-to-body diode of the NFET. This leakage can be slightly mitigated by the body-tie resistance in the transistor, but for SOI technologies that have low resistance values, this configuration may not be sustainable.

## 5.9 Simulation Results

To compare the original three-level shifters, the three proposed dynamic threshold level shifters and the voltage doubler, simulations were run with the University of Florida SOI UFSOI SPICE tool using  $0.25\mu$  partially-depleted SOI CMOS models. All outputs were loaded with 20 fF of capacitance. Input stimuli included four subthreshold pulses driven into the input and a total of 18  $\mu$ S of simulation time in order to capture rising and falling delays as well as dynamic power. Simulations were run with no input activity for the same duration to capture static

power, and, finally static simulations were also run in which a radiation event also was included to identify sensitivity to noise in each circuit.

For the dynamic case, four pulses were used to ensure that delays were consistent and were not affected by the switching history. Example waveforms of one of the simulations are shown in Figure 4 and this figure illustrates how the input signal is subthreshold at the value of 0.35V. This subthreshold level was chosen simply due to the ease of producing this voltage using energy-scavenging techniques (i.e., piezo-electric, temperature gradient, solar, etc). All circuits were sized with a total of 25 microns of transistor width to maintain roughly equivalent area while comparing other aspects of the design in terms of performance, power and radiation insensitivity.

Major problems for the voltage doubler resulted in excluding this promising circuit from the analysis. First, depending on the recent switching history of the input signal, the delay through the circuit is changed by as much as 15%. In the case where the circuit had been idle for extended periods, the two dynamic nodes (the top node on both capacitors) would leak to an equivalent value, one threshold voltage drop below the supply voltage. This would result in a worst-case propagation delay immediately following long periods of inactivity. First transitions would be long and second and subsequent transitions had delays that would drop and settle at an equilibrium level. This behavior would be generally unacceptable in most applications in which an additional timing margin would need to be added to accommodate the worst case delays for setup analysis. Additionally, the input timing of the true and complement signals had a thin window of operation. By simply modifying the supply voltage, temperature or process (within specifications), the doubler performance would significantly degrade. Furthermore, the dynamic nature of the circuit, in which one of the two nodes is always floating at a level above the supply

voltage, results in the circuit being susceptible to noise as other signals can capacitively couple the node down and, thus, introduce additional unexpected delay, – reducing circuit robustness further. Consequently, we do not consider this approach feasible for typical industrial applications.

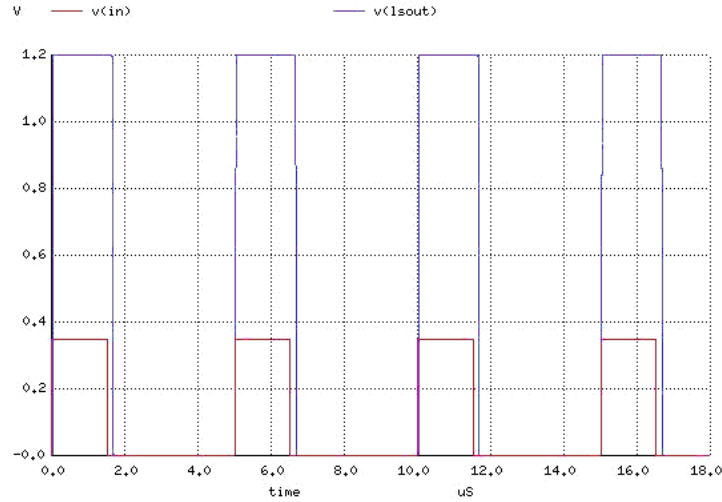


Figure 5.13 SOISPICE5 simulation waveforms

Of the traditional circuits, the DSLS1 and SSLS circuits failed to function with a subthreshold input and the output was statically held low in both cases, unable to switch regardless of time. By significantly increasing the size of the NFET pull-down transistors, the circuits could be designed to work but only at the expense of unreasonable area increase (i.e., 500 micron transistor widths, 100 times larger). Consequently, Table 5.2 shows only the measured values of the four remaining circuits, DSLS2 and the three improved versions (DSLS1b, DSLS2b, and SSLSb).

When considering performance and power, the traditional DSLS2 worked well and only by the improved version (DSLS2b) out performed it by approximately 12%. Both of the DSLS circuits had less static power consumption and significantly less dynamic power. This is a result

of intermediate nodes for the DSLS not swinging rail-to-rail. The performance gained by DSLS2b over DSLS2, comes at the mild expense of a slight increase in dynamic power. The DSLS1b and SSLSb circuits both worked but were slower and consumed more energy, making both of them, generally as poor choices for level shifting in subthreshold. However, the SSLSb circuit does provide extra flexibility in terms of power routing and in some cases may be the appropriate circuit.

In terms of radiation and noise sensitivity, each of the circuits was held in one of two states (i.e., input high and input low) and charge was injected into the critical node for a total of 1 nS, simulating a soft error transient. The injected charge was increased until the circuit was deemed to have failed, where failure was defined as the output spiking above  $V_{il}$  or below  $V_{ih}$  for any period of time.  $V_{ih}$  and  $V_{il}$  were determined based on an inverter running at 1.2V and were 0.69V and 0.49V, respectively. Typical soft errors are defined as data being corrupted, but in the case of level shifters – where the inputs are statically held and the circuit is essentially combinatorial – this failure type is not possible. By defining failure as the charge required to spike the output outside of the range of a known logic level, the noise event could propagate through the logic path and increase in magnitude through each subsequent gate. However, eventually, the event will subside and the level shifter will recover, unlike a latch or flip-flop. The  $Q_{crit}$  value in Table 5.2 provides insight into the insensitivity of the circuit to any type of charge injection due to noise.



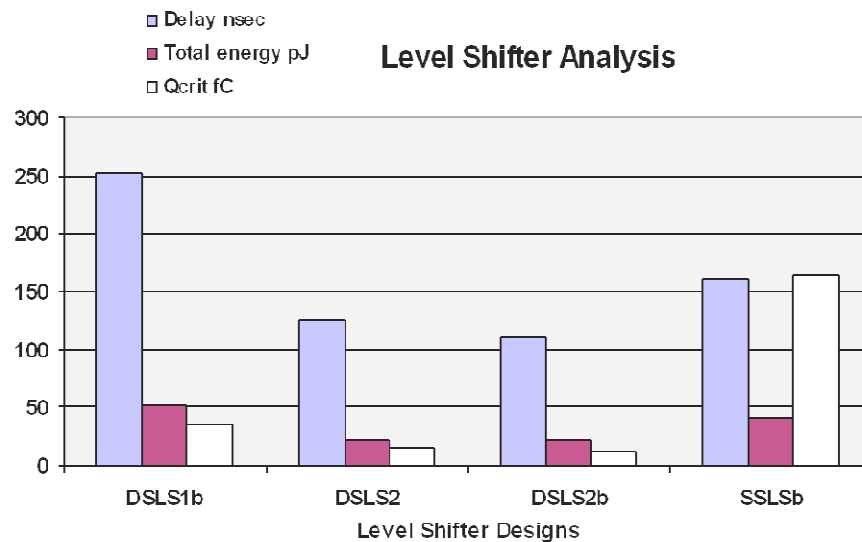
Table 5.2 Level shifter design parameters

	Level Shifter Designs			
Design Parameter	DSLS1b	DSLS2	DSLS2b	SSLSb
Delay (nsec)	252	125	110	161
Total Energy (pJ)	52	21.4	21.8	40.5
Static Energy (pJ)	35	21	21	37
Dynamic energy (pJ)	17	0.4	0.8	3.5
Qcrit (fC)	35	14.5	11.8	165

Referring to the Table 5.2, we observe that the least sensitive circuit is the SSLS; and this is based on the fact that even the input NFET in this circuit is operating in superthreshold (subthreshold signal driving a gate with a very low threshold of around 0.1V). Consequently, every node in the circuit is driven robustly and more charge from a noise event can be accommodated. The DSLS2b circuit on the other hand had the worst performance due to the fact that the subthreshold transistors were not improved in this circuit, but rather the superthreshold PFET was weakened to improve the NFET- to-PFET drive ratio. Following from this, the circuit was the least capable of sinking injected charge without impacting the voltage of the output. Even in light of this significant difference in noise performance, the DSLS2 is likely to be the best candidate circuit, because unrecoverable soft errors are not technically possible in level shifters as described previously.

Power, performance and noise insensitivity are all captured in Figure 5.14, which illustrates a bar graph representation for all four operational circuits. From this graph and with an emphasis

on the performance and power consumption, we consider the DSLSb to be the superior circuit for subthreshold level shifting.



## 5.10 Conclusion

A charge pump is necessary for implementation of the SRR design, which is run from a single subthreshold supply voltage. Thus, this Chapter has compared a variety of charge pump. The main concern identified with existing charge pumps is the operation of the first stages beneath the threshold voltage and Dickson-based charge pump with a proposed clock booster was introduced that alleviates some of the performance degradation. Furthermore, of current load applications, the CNSP pump remains the most effective whether operating in sub- or super-threshold. The CNSP design was proposed to offer enhanced output voltage from the use of the proposed clock booster in the subthreshold operation.

Also in this Chapter several level shifters were been evaluated in the context of translating signals from subthreshold levels to traditional CMOS levels. Three conventional circuits were included in the evaluation and along with three novel level shifters, which were proposed for the first time in this study. All six circuits were evaluated in terms of power, performance and radiation hardness for a constant area. Voltage doublers were eliminated from consideration based on issues with noise and operating range. The proposed current mirror level shifter with dynamically-adjusted threshold was the best, it provides 13% increased performance.

## Chapter 6

### Subthreshold Radhard Reconfigurable (SRR) Architecture

An overview of the Subthreshold Radhard Reconfigurable (SRR) architecture is presented in this Chapter. In the previous Chapters the design and performance analysis of the flip-flop, level shifter and charge pump blocks of the SRR design are provided. The below discussion introduces the processing block and interconnection that complete the SRR design. The architecture of SRR is a fine grain island style and SRAM (Static Random Access Memory) based. The definitions of granularity and implementation style classifications are provided in Appendix A. For the SRAM based design, as the name suggests, the configuration bits defining the datapath and interconnection are stored in the SRAM cells within a reconfigurable architecture. Other popular styles of reconfigurable architectures are Flash based, Hardwired or ROM based. The architecture for SRR is divided primarily into three sub-designs:

- 1.) Programmable Processing Element (PPE)
- 2.) Interconnection Cross-bar, and
- 3.) Configuration SRAM Cells

#### 6.1 Programmable Processing Element (PPE)

The PPE forms the basic functional unit of the SRR design for algorithm execution. As shown in Figure 6.1, the PPE internal architecture consists of a 4-input Look Up Table (LUT), carry and control logic block, conditional routing elements and a radhard storage unit. The design is consistent with most commercially available Field Programmable Gate Array (FPGA)

architecture design. The concept of integrating an LUT and storage element is to provide the flexibility of implementing either a combinatorial or sequential design. Compared to commercial FPGAs, the difference mainly lies in the design of optimized modules incorporating radiation tolerance and functionality at ultra-low voltage. For special-grade FPGAs, the popular radiation mitigation technique is to have redundant designs running in parallel. The output from the individual designs is cross referenced using a majority voting circuit. The majority logic value is passed out as the result for that circuit. As discussed in Chapter 2, the technique is popularly known as Triple Modular Redundancy (TMR). TMR is effective for only Single Event Upset (SEU) but fails for a Multiple Bit Upset (MBU). However, the storage element within PPEs is a novel flip-flop design, MOSACFF, which is radiation hardened by design and not by applying redundant copies, thus, enabling the advantage of power conservation and area efficiency compared to the TMR technique.

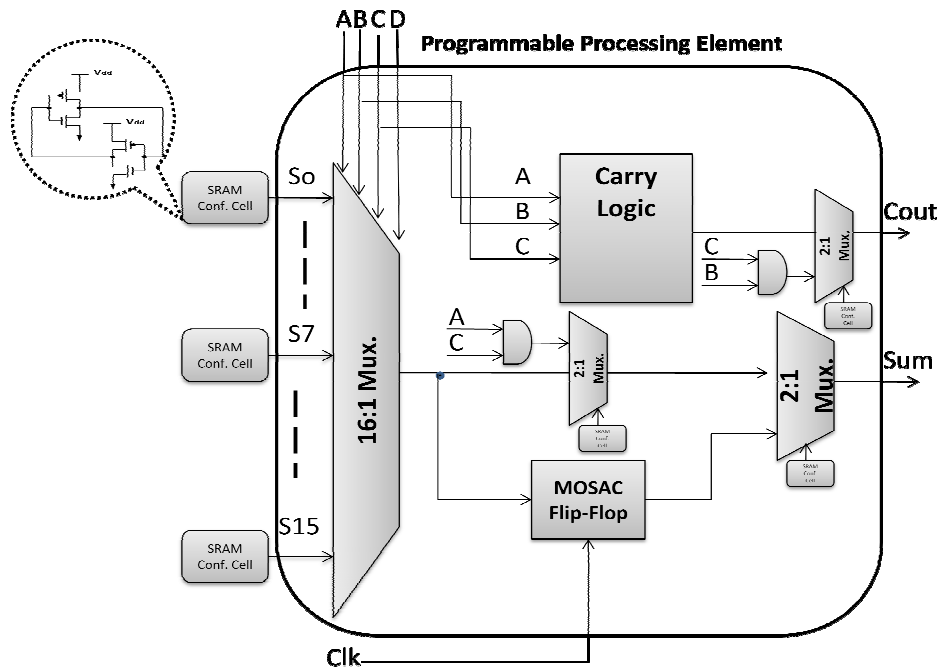


Figure 6.1 Programmable Processing Element

A k-input LUT implements a k-input logic function. As shown in the Figure 6.1, the possible values for an output are stored in  $2^k$  SRAM cells connected as inputs to a  $2^k:1$  multiplexer. The choice of a 4-input LUT is based on the analysis given in [53], which primarily compares the area efficiency, routing complexity and performance for different inputs. To reduce the computation time, designers devised faster ways to add two binary numbers by using carry look ahead adders. They work by creating two signals ( $P$  and  $G$ ) for each bit position, based on whether a carry is propagated through from a less significant bit position (at least one input is a '1'), a carry is generated in that bit position (both inputs are '1'), or a carry is killed in that bit position (both inputs are '0'). In most cases,  $P$  is simply the sum output of a half-adder and  $G$  is the carry output of the same adder. After  $P$  and  $G$  are generated the carries for every bit position are created. As illustrated below Carry logic within a PPE implements the carry ahead logic.

#### **Example - 4 bit Carry Look Ahead Logic**

$$\begin{aligned}
 g_i &= A_i \cdot B_i && \text{-- Generate} \\
 p_i &= (A_i + B_i) \cdot C_i && \text{-- Propagate} \\
 C_{i+1} &= g_i + p_i \cdot C_i && \text{-- Carry next stage} \\
 C_1 &= g_0 + p_0 \cdot C_0 && \text{(2-Level Product-of-Sums)} \\
 \\ 
 C_2 &= g_1 + p_1 \cdot C_1 \\
 C_2 &= g_1 + p_1 \cdot (g_0 + p_0 \cdot C_0) \\
 C_2 &= g_1 + p_1 \cdot g_0 + p_1 \cdot p_0 \cdot C_0 && \text{(2-Level Product-of-Sums)} \\
 \\ 
 C_3 &= g_2 + p_2 \cdot C_2 \\
 C_3 &= g_2 + p_2 \cdot (g_1 + p_1 \cdot g_0 + p_1 \cdot p_0 \cdot C_0) \\
 C_3 &= g_2 + p_2 \cdot g_1 + p_2 \cdot p_1 \cdot g_0 + p_2 \cdot p_1 \cdot p_0 \cdot C_0 && \text{(2-Level Product-of-Sums)} \\
 \\ 
 C_4 &= g_3 + p_3 \cdot C_3 \\
 C_4 &= g_3 + p_3 \cdot (g_2 + p_2 \cdot g_1 + p_2 \cdot p_1 \cdot g_0 + p_2 \cdot p_1 \cdot p_0 \cdot C_0) \\
 C_4 &= g_3 + p_3 \cdot g_2 + p_3 \cdot p_2 \cdot g_1 + p_3 \cdot p_2 \cdot p_1 \cdot g_0 + p_3 \cdot p_2 \cdot p_1 \cdot p_0 \cdot C_0 && \text{(2-Level Product-of-Sums)}
 \end{aligned}$$

Another significant difference, compared to popular commercial FPGAs is the implementation of two AND gates, used to efficiently calculate the partial products in the configuration of a multiplier. This direct computation saves the SRR from any additional configuration SRAM cells and control logic required to implement the operation using a LUT. Three 2:1 routing multiplexers (A, B and C) within PPE are used to select data as shown:-

Mux A - AND1 operation or carry logic,

Mux B - AND2 operation or LUT output, and

Mux C – Storage element or Mux B.

The SRR design is not cluster based, where a group of logic elements, typically two, are interconnected as a single unit. The motivation behind this is to efficiently utilize the logic elements and uncomplicated the routing.

### **6.1.1 Subthreshold Multiplexer**

The multiplexer circuit in the SRR architecture plays an important role in not only routing the data with the PPE, but also implementing the LUT functionality. Three multiplexer designs were analyzed for subthreshold operation with primary focus on power consumption. The design styles considered are:

1. Transmission Gate Based Mux,
2. NAND Gate Based mux, and
3. Tri-State Based Mux.

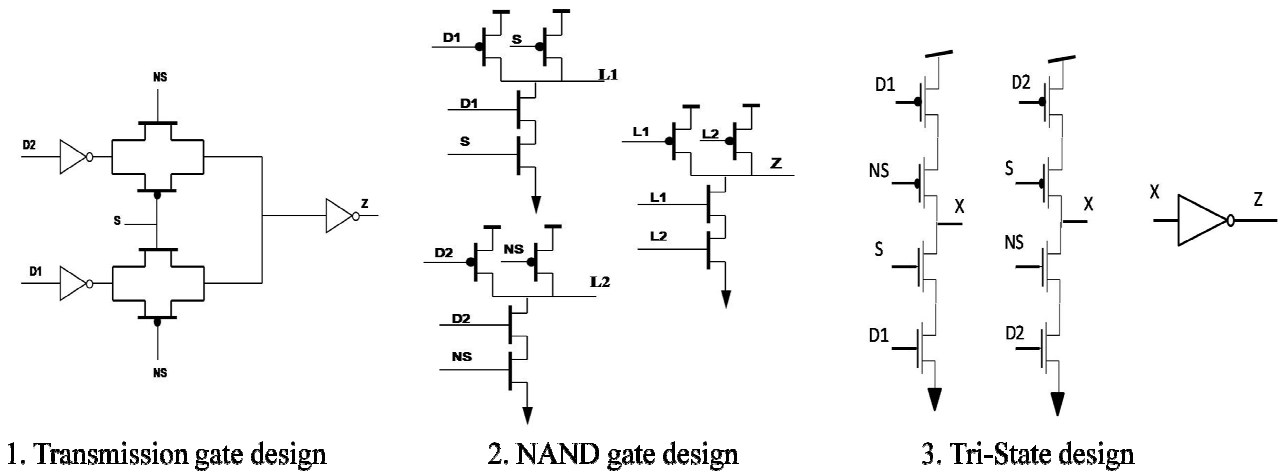


Figure 6.2 Multiplexer designs

Figure 6.2 illustrates the transistor-level implementation of the multiplexer circuits. Table 6.1 provides performance and power consumption analysis for the multiplexer designs in superT, 1.0V ( $V_{dd} > V_t$ ) and subT, 175mV ( $V_{dd} < V_t$ ), voltages. As shown in Table 6.1, the transmission gate design suffers in performance at subT but excels in superT, however, the power consumption values are on the higher compared to tri-state inverter design. The NAND based design offered better performance in subT but adds a power consumption penalty. As power consumption is critical, tri-state based design is ideal for operation at both, subT and superT, voltages.

Table 6.1 Multiplexer Design Comparison

Multiplexer Design	16:1 Delay 175mv(nsec)	16:1 Power 175mv(nW)	16:1 Delay 1.0v(psec)	16:1 Power 1.0v( $\mu$ W)
Transmission gate based	128.1	10.2	44.5	3.45
NAND based	126.12	10.53	48.3p	3.2
Ti-Sate inverter based	133	<b>7.5</b>	52.5	<b>2.52</b>



## 6.2 Interconnection Crossbar

Interconnect crossbars are used to route data between the processing elements of the SRR design. Figure 6.3 illustrates a simple interconnection crossbar comprising pass transistors (NFETs). In this design the routing path is selected by programming the configuration cells connected to the pass transistor gates as shown in Figure 6.4.

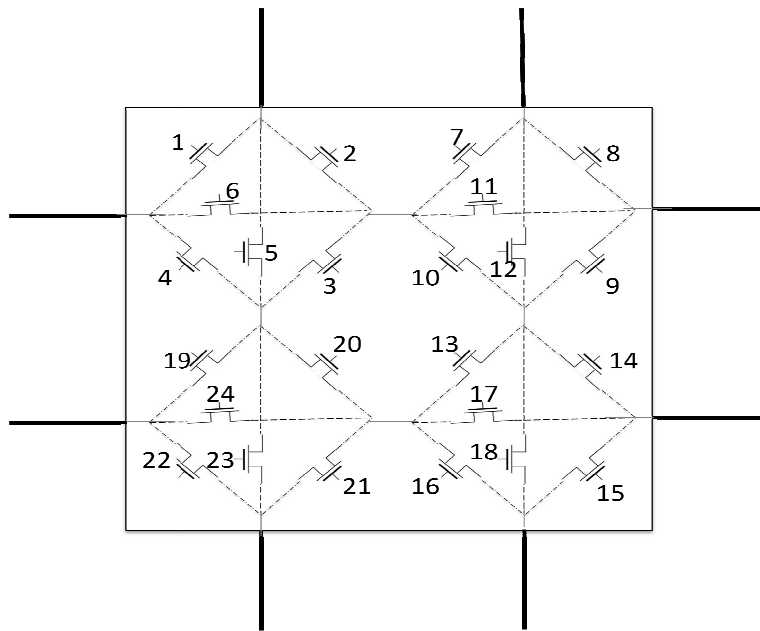


Figure 6.3 Interconnection switch

As shown in Figure 6.6, for the SRR SubT operation, an interconnection crossbar with pass transistor severely degrades as the signal passes through the NFET. For each pass through the NFET the transmitted signal voltage is reduced by  $V_t$ , the threshold voltage of the transistor ( $V_g - V_t$ ). Therefore, as shown in Figure 6.4, the SRR design utilizes charge pumps circuits proposed in Chapter 4 to boost the static voltage of the configuration cell. The boosted cell voltage, X1, X2, X3 and X4 as shown in Figure 6.4, is well above the transmitted logic value.

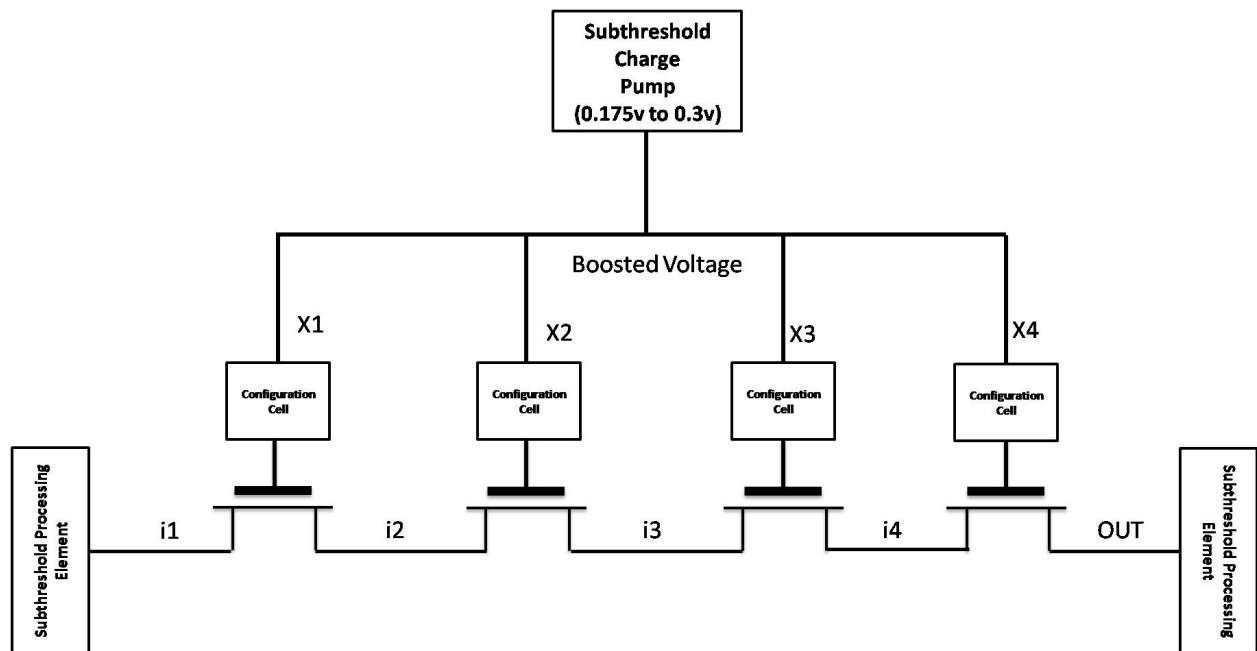


Figure 6.4 SubT Interconnect with Charge Pumps

Figure 6.5 illustrates the HSPICE simulation of the setup shown in Figure 6.4, for interconnecting logic with pass gates. Routing with pass gates is preferred as it reduces area overhead and, in turn, power. Furthermore, the implementation simplifies the design. At time interval 'A' for Figure 6.5 the configuration cells are connected to the charge pumps and followed by an increment of their respective stored voltages (X1, X2, X3, and X4). At time 100  $\mu$ secs, when the configuration cell output potential has reached sufficient level ( $>175\text{mv}$ ), the setup is ready to transmit data. Figure 6.6 illustrates the routing verification for multiple transitions of input over time with the charge pump. The simulated total power consumed for the setup shown in Figure 6.4 is 0.4nW.

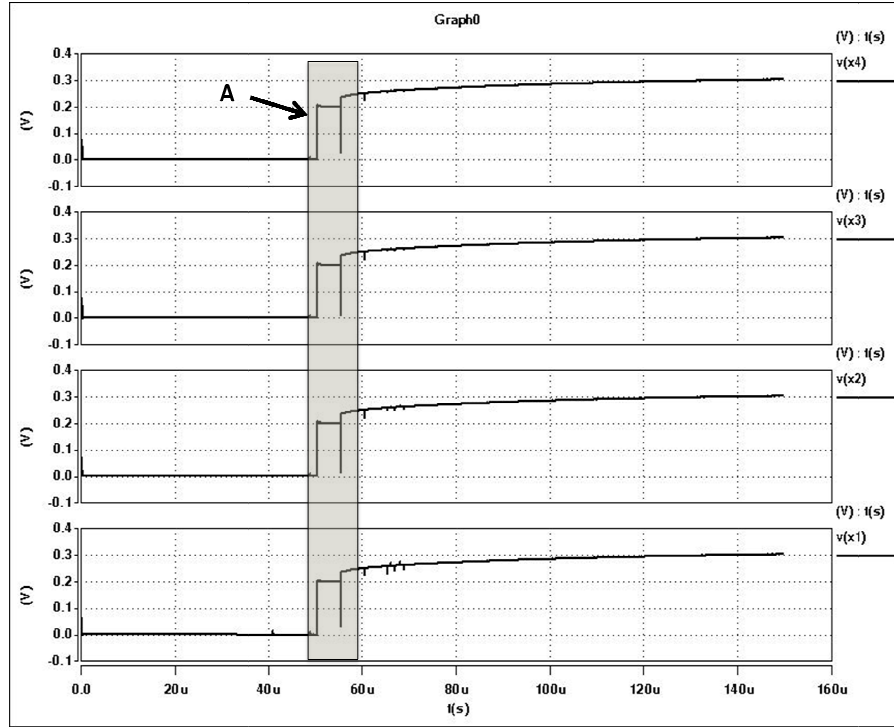


Figure 6.5 HSPICE simulation waveform for interconnect routing with charge pumps

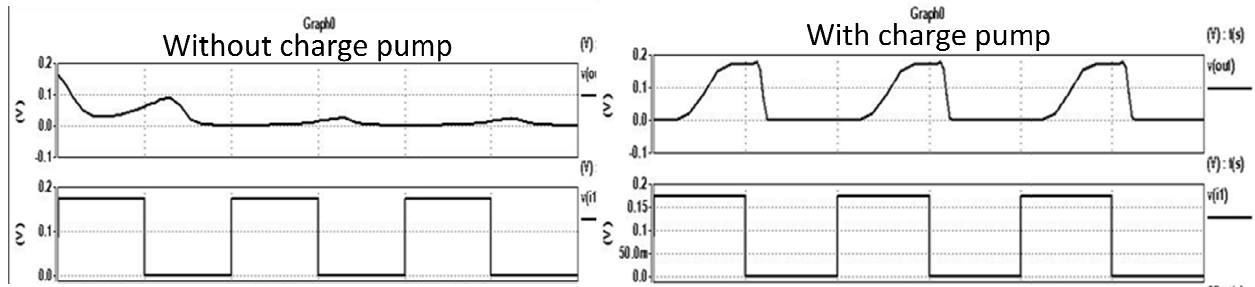


Figure 6.6 HSPICE routing simulation waveforms without and with charge pump

#### 6.4 Level Shifters for SubT and SuperT Logic Interface

To optimize power and performance, the processing elements of the SRR design can be operated in parallel with SubT and SuperT voltages. This dual-voltage operation in turn forms voltage islands of SubT and SuperT processing elements within the SRR design. To interface the

logic computed from SubT to SuperT level shifters proposed in Chapter 4 are utilized. Figure 6.7 shows the setup of the interface, the SubT logic value stored in the proposed MOSAC FF is connected as input to the level shifter and the output of the level shifter is connected as input to the MOSAC FF operating at superT. Figure 6.8 illustrates the HSPICE simulation for the level shifter interface setup. The data input to the SubT FF is *din*, which is latched with respect to the clock, *clk*, operating at 166 KHz and the output is *subout* from the SubT MOSAC FF. The signal *subout* is passed as input to the level shifter and the output of level shifter *lsout* is connected as input to the SuperT MOSAC FF. The value *lsout* is latched with respect to *clk2* clock signal operating at 10M Hz.

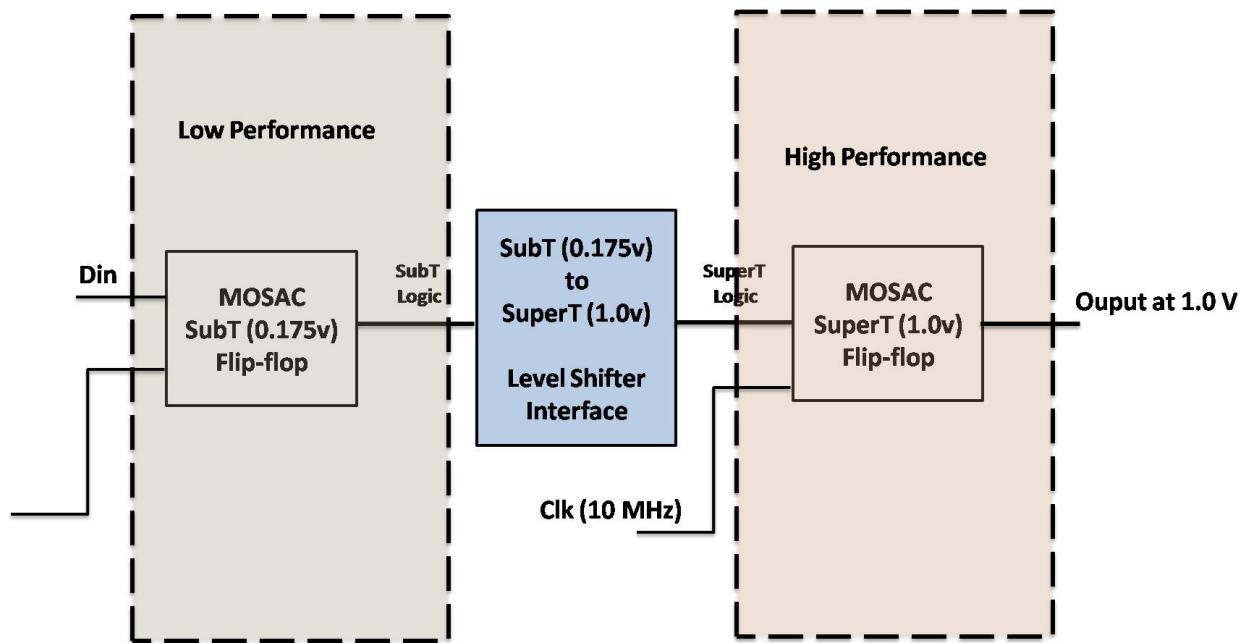


Figure 6.7 Interfacing SubT and SuperT with level shifters

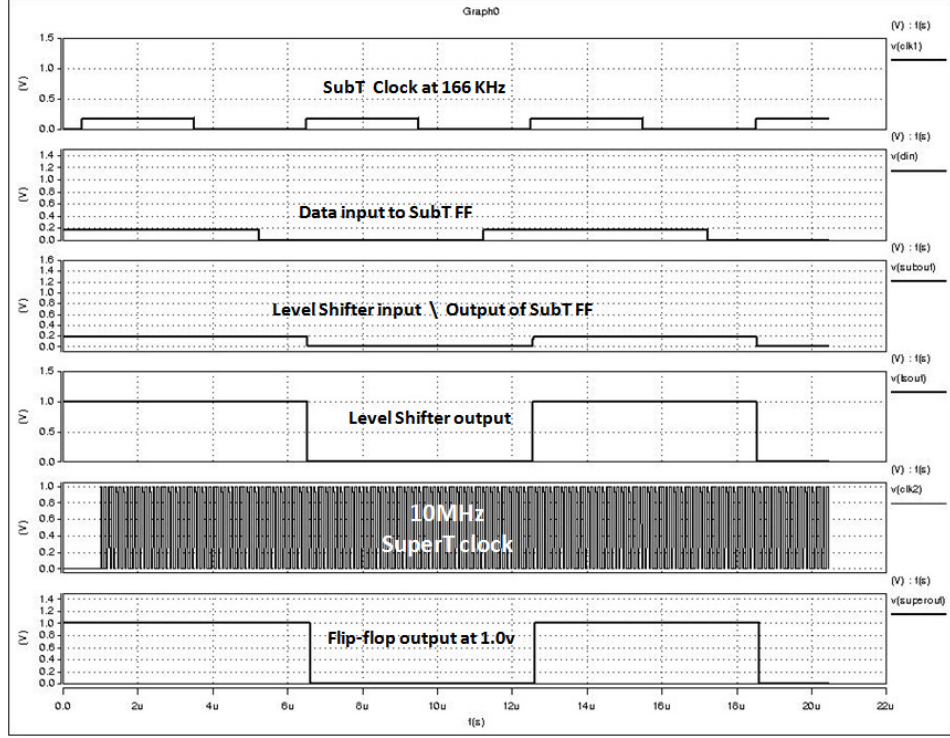


Figure 6.8 HSPICE simulation of interface of SubT and SuperT with level shifters

## 6.5 Conclusion

The Chapter describes the implementation and analysis of all the essential design modules for the PPE of the SRR architecture, which include: a 4-input Look Up Table (LUT), carry and control logic block, conditional routing elements and a radhard storage unit. Various circuit style implementations for Look Up Table (LUT) multiplexors are evaluated for subthreshold operation. Tri-state-inverter based multiplexors provided better power consumption values than transmission-gate based and NAND-based multiplexers at subT, 175mv. To improve routing at subT, without signal degradation, interconnect crossbar configuration cells are boosted with charge pumps. Furthermore, to optimize power and performance, level shifter interfaces are used to connect subT and superT processing elements.

## Chapter 7

### SRR Simulation Results

This Chapter describes the Multiply and Accumulate (MAC) design implemented to verify the functionality of the SRR architecture. All the simulations of the SRR-MAC are performed using 45nm IBM12SOI fully depleted transistor models in HSPICE. For comparison purposes, the power consumption values of an existing ASIC design and commercial FPGA conFIGured as MAC are matched with the proposed SRR-MAC. Finally, Xpower software estimated power consumption values for the FPGA-MAC are used to extrapolate the measured value.

#### 7.1 SRR MAC Configuration

To verify the functionality of all the designed modules for the SRR a basic Digital Signal Processing (DSP) algorithm is implemented. The DSP algorithm chosen is a Multiply-Accumulate (MAC) unit, because MAC forms the core computational block for most DSP algorithms. As shown in Figure 7.1, the implemented MAC unit comprises an 8-bit multiplier followed by a 16-bit ripple carry adder and a 16-bit accumulator. Pseudo-Random values for the MAC unit input are generated by a Linear Feedback Shift Register (LFSR).

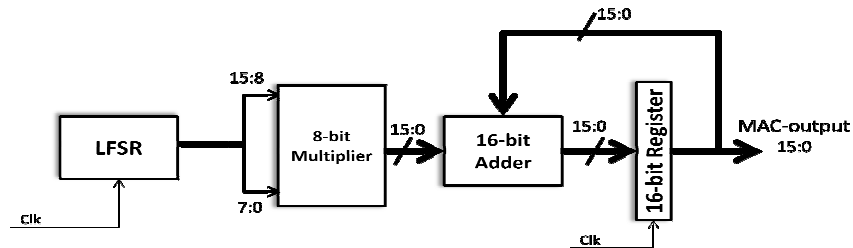


Figure 7.1 MAC Unit Design

The simulated power consumption values of the SRR-MAC design are compared with an existing Application Specific MAC (ASMAC) design that employs an adaptive body-bias supply and an off-the-shelf commercial Xilinx Field Programmable Gate Array (FPGA) programmed to function as an 16-bit MAC. The following sections describes the design units comprising the MAC unit along with a comparative analysis for the MACs implemented in SubT and SuperT platforms.

### **7.1.1 Linear Feedback Shift Register**

A LFSR is a shift register in which the input bits are a linear function of the previous state. The only linear functions of single bits are exclusive-or (XOR) and inverse-Xor (XNOR); thus, an LFSR is a shift register with an input bit that is driven by the XOR of some bits of the overall shift register value. The initial value of the LFSR is called the seed, and because the operation of the register is deterministic, the stream of values produced by the register is completely determined by the current (or previous) state. Likewise, because the register has a finite number of possible states, the register must eventually enter a repeating cycle. However, an LFSR with a well-chosen feedback function can produce a sequence of bits that appear random and that have a very long cycle. Applications of LFSRs include generating pseudo-random numbers, pseudo-noise sequences, fast digital counters, and whitening sequences. Both hardware and software implementations of LFSRs are common.

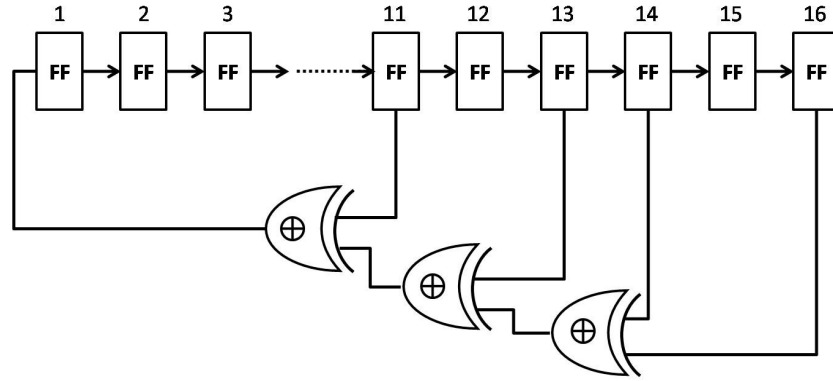


Figure 7.2 Linear Feedback Shift Register Design

The arrangement of taps for feedback in an LFSR can be expressed in finite field arithmetic as a polynomial mod 2. This means that the coefficients of the polynomial must be 1's or 0's. This is called the feedback polynomial or characteristic polynomial. For example, if the taps are at the 16th, 14th, 13th and 11th bits (as shown), the feedback polynomial is

$$X^{16} + X^{14} + X^{13} + X^{11} + 1 \quad (\text{Equation 7.1})$$

### 7.1.2 Array Multiplier

After addition, multiplication is the most common operation for DSP algorithms. There are many multiplication algorithms proposed for hardware implementation like the Baugh-Wooley, Booth Encoding, Wallace Tree, Serial and Array methods. In the current research for comparison purposes with ASMAC, an array-based multiplication algorithm is implemented. The technique is simple and easier; the first partial products are independently computed in parallel. For example, consider two binary numbers  $A$  and  $B$ , of  $m$  and  $n$  bits, respectively. Then, there are  $mn$  summands that are produced in parallel by a set of  $mn$  AND gates. The  $n \times n$



multiplier requires  $n(n-2)$  full adders,  $n$  half-adders and  $2n$  AND gates–Worst case delay would be  $(2n+1)td$ .

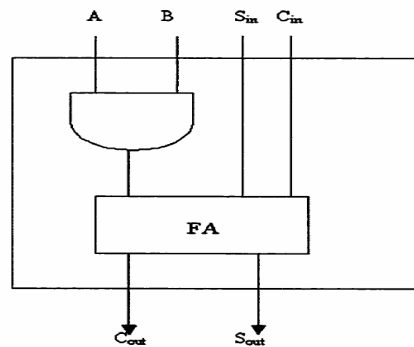


Figure 7.3 Basic Cell of a Parallel Array Multiplier

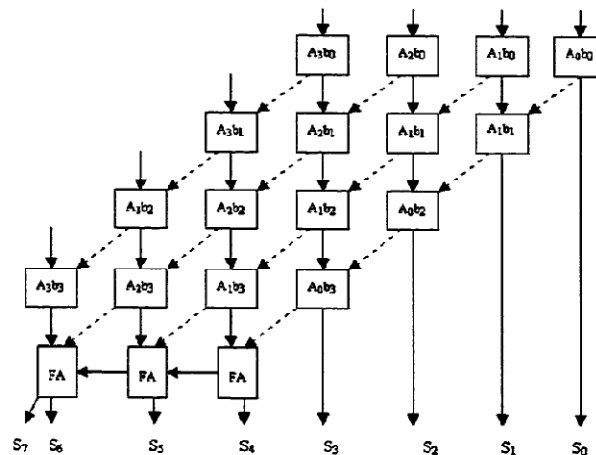


Figure 7.4 - Array Structure of Parallel Multiplier

### 7.1.3 Ripple Carry Adder and Accumulator

16-bit ripple carry adder is implemented by interconnecting 16 processing elements; each processing element is in turn, programmed as a 1-bit full adder. The logic for a 1-bit full adder is stored in the LUT of the processing element via a configuration cell. Shown in Figure 7.5 is simple setup example for the ripple carry adder interconnection.

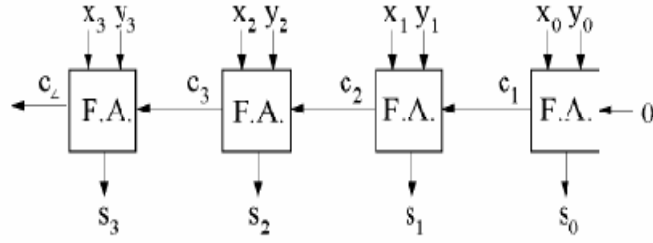


Figure 7.5 Ripple Carry Adder

## 7.2 SubT-MAC HSPICE Functional Verification

The SRR SubT-MAC implementation, was simulated, at 175mv, for functional verification in HSPICE using 45nm IBM12SOI Fully-Depleted transistor models. The pseudo-random 16-bit values, A7-A0 and B7-B0, generated from the LFSR are connected as inputs to the MAC unit. The SubT input values supplied are shown in Figure 7.6, with A7-A0 = 56<sub>H</sub> and B7-B0=DB<sub>H</sub>.

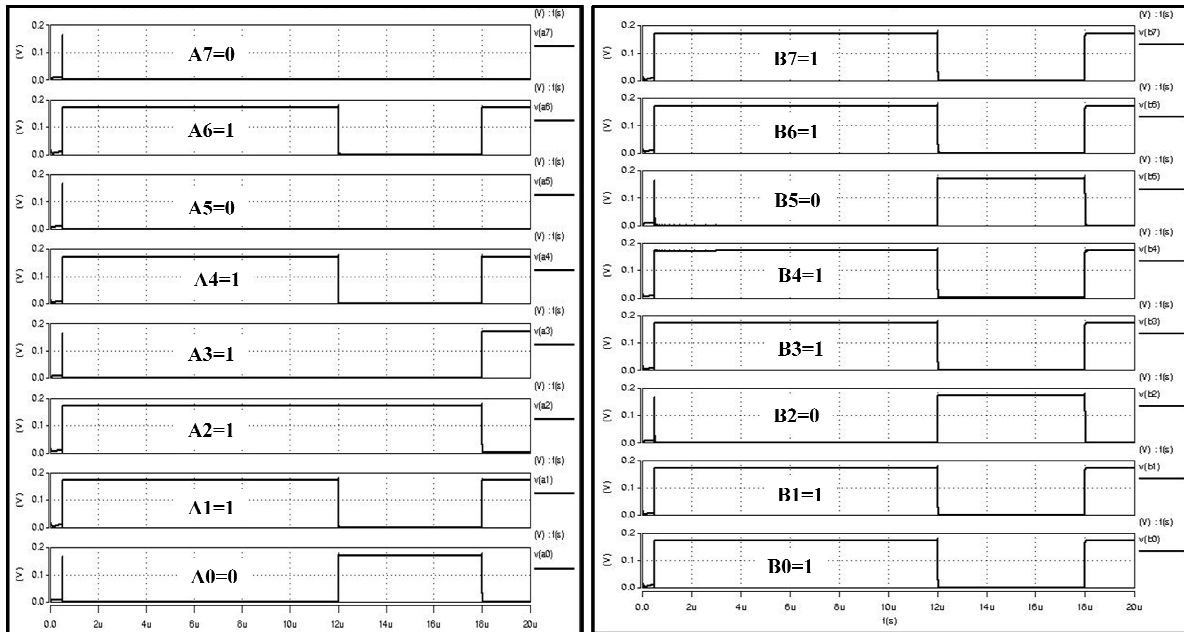


Figure 7.6 SubT-MAC Inputs – Two 8-bit Values

The first computation performed on the input values, A and B, is the 8-bit multiplication using the array multiplier conFigured block of SRR. As shown in Figure 7.7, for the supplied input values ( $56_H \times DB_H$ ), the product generated is  $P15-P0 = 4992_H$ .

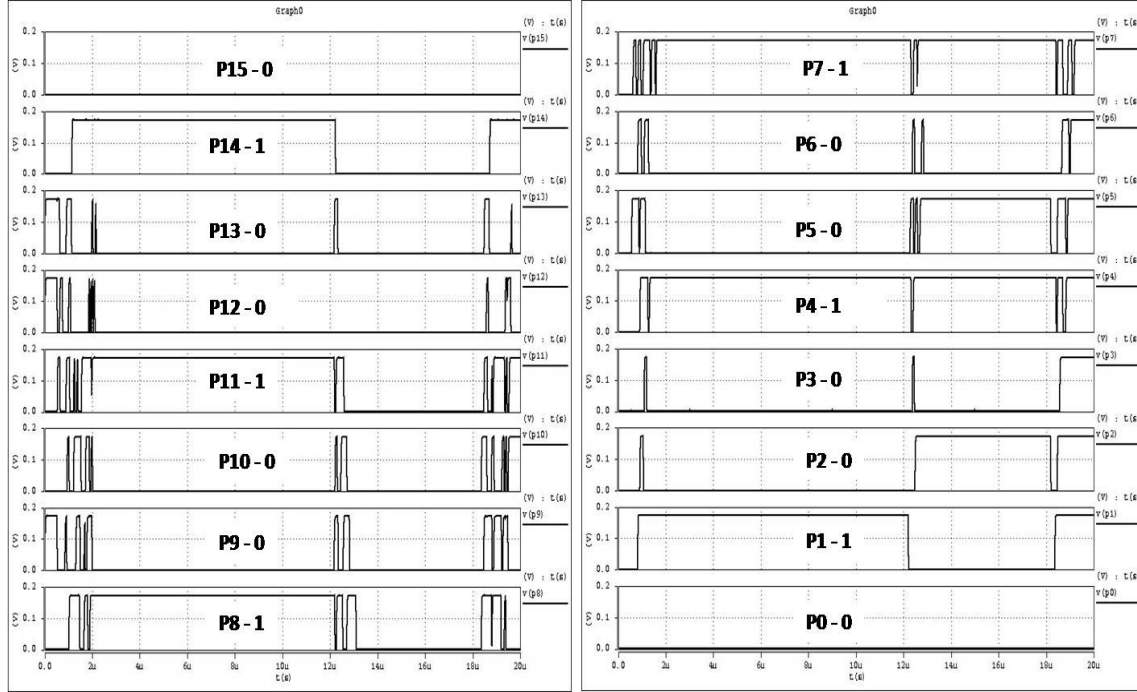


Figure 7.7 SubT-MAC multiplier output (P15-P0)

The 16-bit unsigned adder block within the MAC implementation adds the contents of the multiplier (P0-P15) and the output of the 16-bit accumulator register (MAC0- MAC15). For the sample computation, the initial value stored in the accumulator register is  $MAC0-MAC15 = 0001_H$  and the value generated after addition is  $4993_H$ , as shown in Figure 7.8, which is stored in the accumulator during the next clock cycle. Table 7.1 shows the functional computation at different time steps for the SRR-MAC simulation.

Table 7.1 Simulation verification of SRR-MAC

Time	Inputs (A:B)	Multiplier Output	Adder Output	MAC Reg. Output
2 $\mu$ sec	56 <sub>H</sub> : DB <sub>H</sub>	4992 <sub>H</sub>	4992 <sub>H</sub> + 0001 <sub>H</sub> = 4993 <sub>H</sub>	0001 <sub>H</sub>
4 $\mu$ sec	56 <sub>H</sub> : DB <sub>H</sub>	4992 <sub>H</sub>	4992 <sub>H</sub> + 4993 <sub>H</sub> = 9325 <sub>H</sub>	4993 <sub>H</sub>
9 $\mu$ sec	56 <sub>H</sub> : DB <sub>H</sub>	4992 <sub>H</sub>	4992 <sub>H</sub> + 9325 <sub>H</sub> = DCB7 <sub>H</sub>	9325 <sub>H</sub>

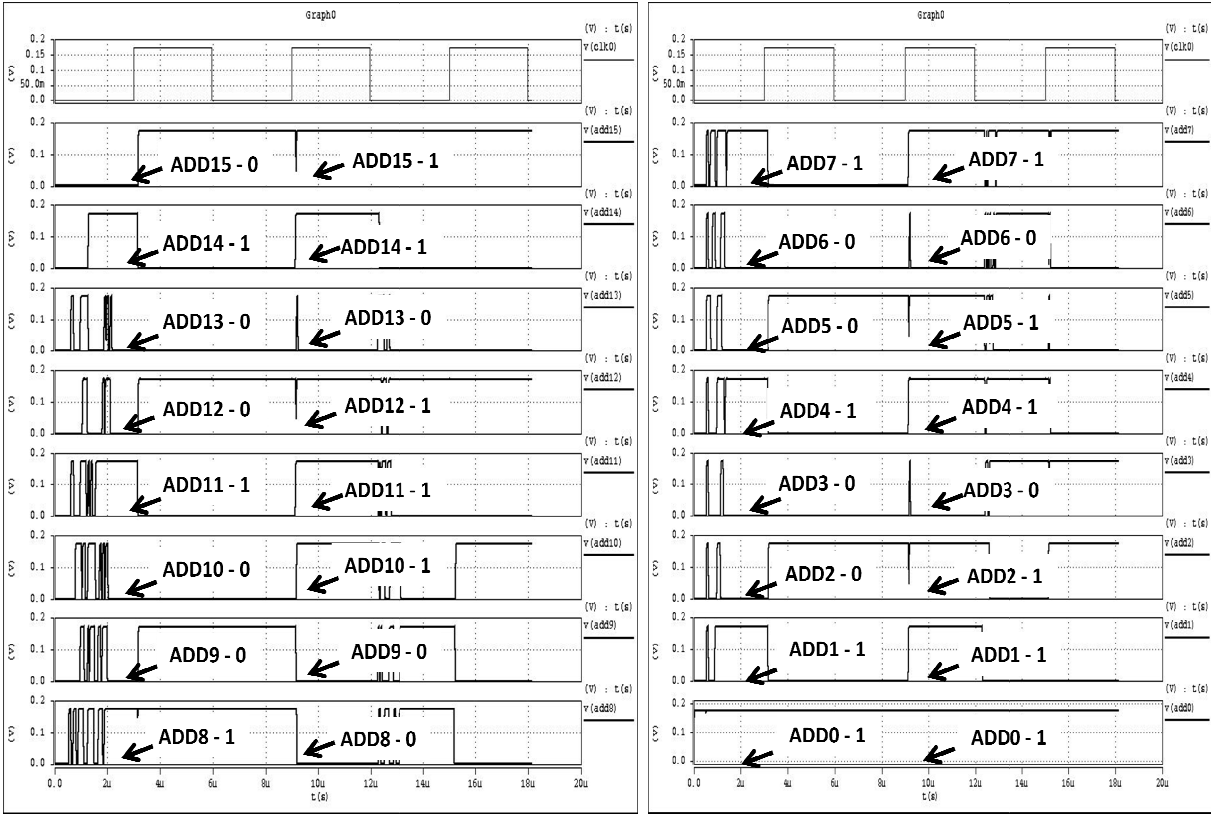


Figure 7.8 SubT-MAC adder output (Add15-Add0)



Figure 7.9 SubT-MAC register output (MAC15-MAC0)

### 7.3 MAC Design Comparison

The following section compares, the SRR architecture for power consumption with two other architectures setup: (1) the ASMAC with adaptive body bias [54] and (2) a Commercial Xilinx Spartan-II XCS15 FPGA [55]. All the architectures implement a 16-bit MAC unit. Table 7.2 illustrates the power consumption comparison values for the SRR-MAC, ASMAC and traditional FPGA. The ASMAC hardware is specific for MAC operation and applies adaptive body bias to reduce the leakage current to obtain optimized subT operation at 175mv. Therefore, the power consumption of the ASMAC design is several orders less than the other designs.

For comparison purposes, the operating frequency and supply voltage are consistent with the ASMAC, however, the 45nm technology of the SRR architecture permits other power and performance configurations, as illustrated in Table 7.3. Additionally, The SRR design

incorporates several important features like functional adaptability, improved noise immunity, robustness to radiation and dynamic voltage scaling capability. Figure 7.10 shows the power distribution of the SRR-MAC design for the circuits integrated to work as a MAC unit. The most power consuming unit is the Interconnection Network (IC N/W) because the circuit employs charge pumps, which consume high power as they constantly switch internal nodes with respect to the system clock. As show in table 7.4, even though the SRR-MAC design consumes higher power than the ASMAC, the design is suitable for energy harvesting, which can be easily powered from solar cells, piezoelectric or thermoelectric devices as energy source.

Table 7.2 Power consumption comparison between SRR-MAC and ASMAC

All designs are analyzed at 166KHz operating frequency						
Design	Technology	Supply Voltage	Power Consumption	Reconfiguration	Radiation Hardness	Energy Harvesting
ASIC -MAC	0.14 $\mu\text{m}$	175mv	14 nW	Not Available	Not Available	Applicable
SubT SRR MAC	45 nm	175 mv	3.75 $\mu\text{W}$	Available	Available	Applicable
SuperT FPGA MAC	0.18 $\mu\text{m}$	2.85 v	3.32 mW	Available	Not Available	Not Applicable

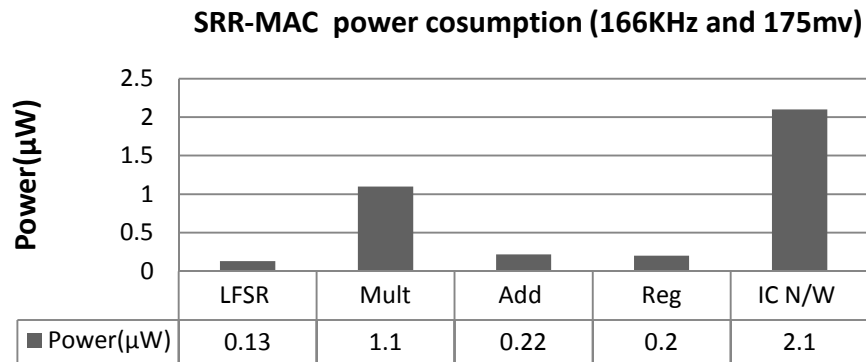


Figure 7.10 Power distribution of the SRR-MAC implementation

Table 7.3 Power performance configurations of SRR-MAC

<b>SRR-MAC (45nm Technology)</b>		
<b>Supply Voltage</b>	<b>Freq</b>	<b>Power consumption</b>
125 mV	166kHz	2.4 $\mu$ W
175 mV	166kHz	3.75 $\mu$ W
175 mv	500kHz	6.2 $\mu$ W

Table 7.4 Energy harvesting sources [56]

<b>Harvesting Technique</b>	<b>Power Density</b>
Solar Cells	15mW/cm <sup>3</sup>
Piezoelectric	330 $\mu$ W/cm <sup>3</sup>
Vibration	116 $\mu$ W/cm <sup>3</sup>
Thermoelectric	40 $\mu$ W/cm <sup>3</sup>

As shown in table 7.2, the MAC implementation on traditional FPGA stands on the extreme end of the power spectrum comparisons. The FPGA-MAC design relies completely on an external dedicated power supply for operation and is unsuitable for energy harvesting applications. However, the traditional FPGA is suited for applications requiring high performance and reconfigurability. As shown in Figure 7.11, for the power measurement, instead of using an off the shelf FPGA development board for analysis, a stand-alone FPGA pin extension socket was designed. This stand-alone FPGA socket permits precise power consumption measurement as it avoids auxiliary devices and components from the development board.

Table 7.5 illustrates the data from the experiment, wherein the FPGA conFigured as MAC was subjected to operate under various frequencies, ranging from 166 KHz to 20 MHz. For each operating frequency, corresponding power consumption was measured. As expected, with the increase in the operating frequency, there is a rise in power consumption for the FPGA-MAC.

The study was extended to evaluate the accuracy of the power estimating software tool from Xilinx called Xpower. The Xpower tool power estimates are based on operating frequency, device utilization and toggle rate. As observed, the Xpower software tool moderately overshoots in estimating the power consumption values. Figure 7.12 shows how the estimated values of the Xpower software can be utilized to extrapolate the measured power consumption of the FPGAs.

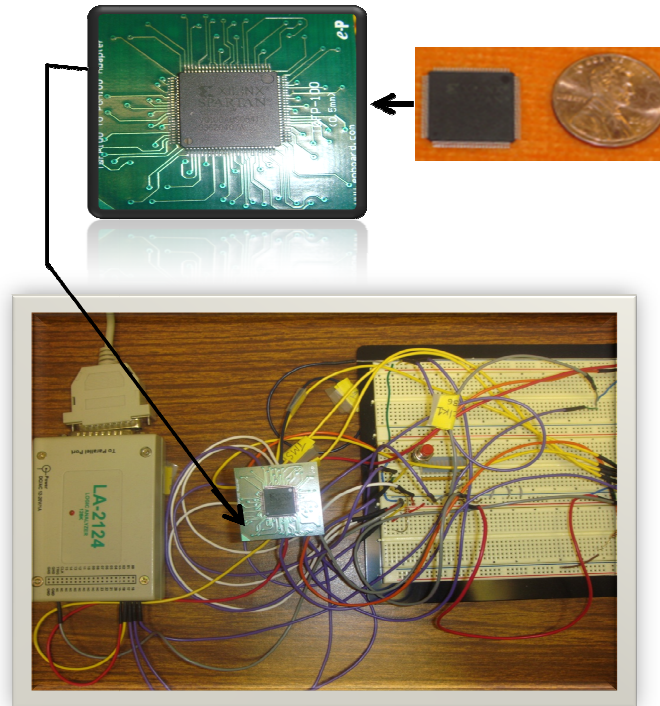






Figure 7.11 Xilinx Spartan II laboratory setup

Table 7.5 Xilinx Spartan-II Analysis

Measured		Xpower Software Estimate	
Clock Freq (MHz)	FPGA-MAC mW	Clock Freq (MHz)	FPGA-MAC mW
166 KHz	3.32	20 MHz	29.37
1 MHz	9.8	30 MHz	33.9
2 MHz	10.36	40 MHz	41.78
5 MHz	11.8	50 MHz	49.66
10 MHz	14.56	60 MHz	57.54
15 MHz	17.0	80 MHz	75.2
20 MHz	20.16	100 MHz	87.0



Table 7.6 Architecture merit comparison

Symbol	Description	Merits
	SRR-MAC	Radhard and reconfigurable
	Traditional FPGA-MAC measured	Reconfigurable, not radhard
	ASMAC	Static design, not radhard
	Traditional FPGA-MAC estimated	Quick power estimate

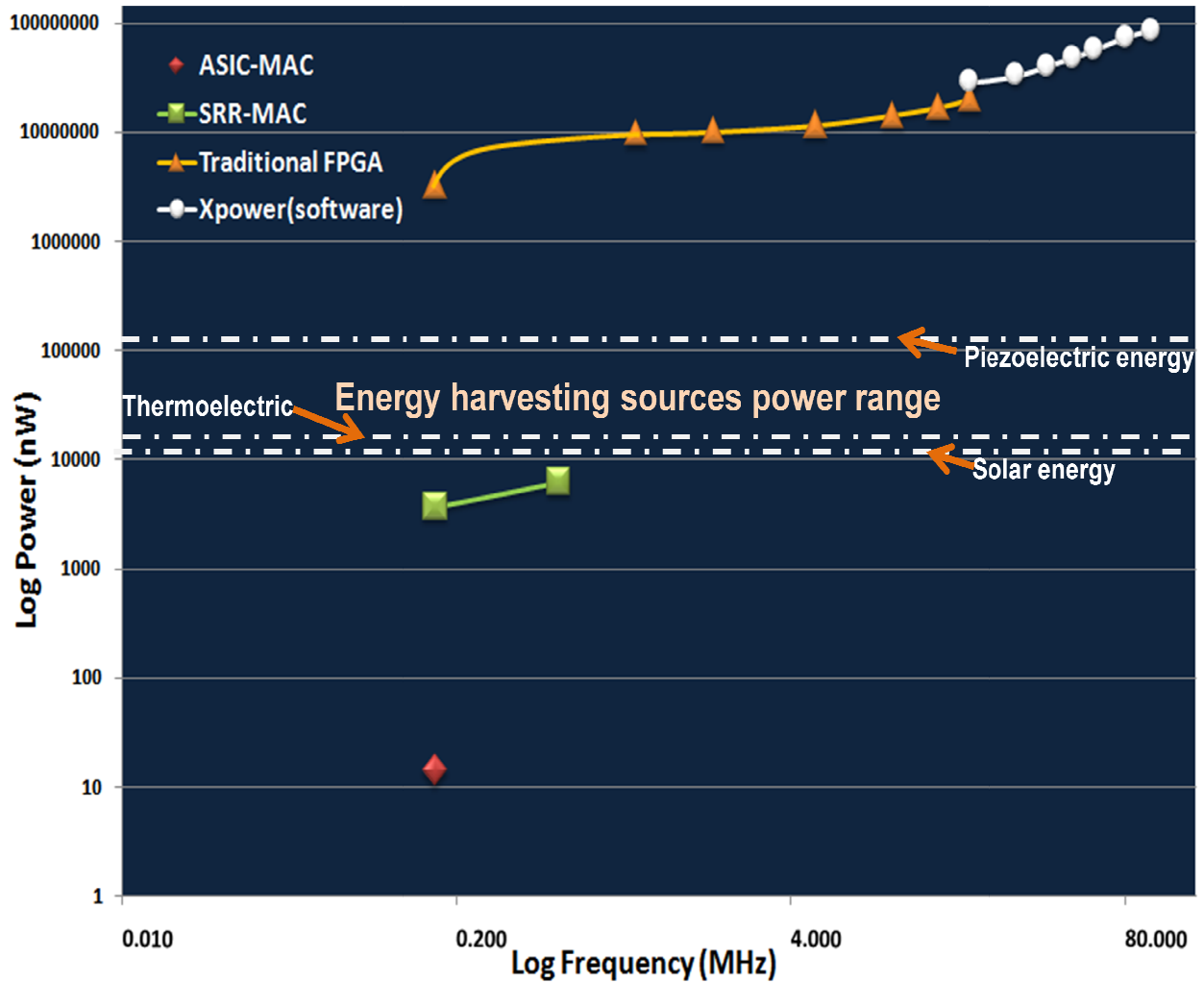


Figure 7.12 is a composite plot of the analysis provided in Tables 7.2, 7.3 and 75. The figure captures the merits and power consumption difference between the ASMAC, SRR-MAC and FPGA-MAC. Table 7.6 lists the description of the merits inherently available in each of the architectures in comparison. As illustrated in Figure 7.12 the SRR design leads in features that are favored by many modern energy-conscious systems running critical applications. On the other hand, other architectures with limited merits are restricted in application base and working environments.

## **7.4 Conclusion**

This Chapter presented the MAC modules implementation by programming SRR processing elements. The SRR-MAC modules are interconnected and verified for functionality in subT (175mv) at 166 KHz frequency. The implemented SRR-MAC unit design was compared for power consumption with ASMAC and FPGA-MAC. The SRR-MAC design showed four orders of power saving compared to that of the off-the-shelf Xilinx FPGA operating at 2.85v. Also in this Chapter the accuracy of Xpower software tool for power measurement was verified. The tool provided conservatively high power estimates and is beneficial for system designers to acquire an initial ball-park power consumption value.

Finally, the SRR architecture was not only shown to be functional, by reconfiguring the design as a MAC but also efficient, by operating under varied power and performance configurations. Moreover, the SRR architecture provides additional benefits of robust operation and dynamic voltage scaling capability. Therefore, the architecture is not only suitable for harsh environment applications like space electronics but also for monitoring and data acquisition applications like biomedical and wireless sensors.

## **Chapter 8**

### **Conclusions and Future Work**

#### **8.1 Conclusions**

This dissertation research implemented an energy efficient robust design that is capable of self-powering, reconfigurable to requirements and resilient in operation to the effects of radiation. The goal is accomplished with the design of the SRR architecture, built with modules of novel designs. The integration of the SRR design is proven operational by configuring a MAC design. The glory of the research lies in the novel circuits presented as building blocks in the course of SRR architecture design, referenced by other researchers to advance the field of subthreshold logic design. The research covered significant background in understanding concepts related to low power techniques, radiation effects, subthreshold operation and SEU mitigation techniques.

Towards this end, an initial study explored an optimized circuit style for subthreshold operation. The study analyzed CMOS, PNMOS, DT-CMOS and NMOS circuit styles for performance, noise immunity and power consumption (static and dynamic) in the subthreshold regime. The study showed, that the static characteristics of CMOS and DT-CMOS circuit styles are the best reported providing the greatest level of noise immunity. Furthermore, the static power consumption was two orders of magnitude less than the next best circuit as expected given that the other circuits implement an always-on load transistor. Pseudo-NMOS and DT-Pseudo-NMOS circuit provided improved area and dynamic power relative to CMOS and derivatives. However, the always-on PFET load resulted in degraded performance and heavy

static power consumption due to driver-to-load contention when the driver is activated. Considering the above analysis, the staple CMOS circuit style was determined to be better optimized for subthreshold operation.

As a next step towards the SRR architecture design, a study was conducted to analyze various flip-flop designs, traditional and radiation hardened, for subthreshold operation. The study performed two sets of experiments which primarily differ in the transistor implementation technology and Simulator Program with Integrated Circuit Emphasis (SPICE) tool. The first set of flip-flop experiments are based on 0.25 $\mu$ m Partially-Depleted SOI transistor models with University of Florida SPICE tool. This simulation study compares a group of nine flip-flops that represent a wide range of designs, traditional non-radiation hardened and radiation hardened, were compared. The study, first of its kind, highlighted the strengths and weaknesses of each flip-flop design in the context of subthreshold operation. The study observed that for all non-redundant cells, a susceptibility to single event upsets is an obvious weakness and the use of redundancy at some level will likely be required for applications in which data integrity is a priority. A new redundant cell was proposed that takes advantage of the reasonable performance of a base dynamic flip-flop with added keepers. Accordingly, the second set of flip-flop experiments are based on 45nm Fully-Depleted SOI transistor models from IBM and using the HSPICE tool. In this analysis radiation hardened flip-flop designs were compared for power, performance and Single Event Transient robustness. A novel flip-flop design, Modified Sense Amplifier C<sup>2</sup>MOS Flip-Flop (MOSAC FF), is proposed which enables SEU/SET resilience with 50% reduced power and 20% better SET Qcrit values compared to other designs in the analysis.

For efficient signal transmission and to support multi-voltage (SubT and SuperT) operation circuits such as charge pumps and level shifters were analyzed in subthreshold regime. To

simplify routing and reduce area overhead, the SRR architecture relies on NFET only pass transistors, but this implementation struggles to drive high voltages and is crippled in the subthreshold region. A unique solution is to overdrive the pass-transistors by applying boosted gate voltages by using a charge pump circuit. A study conducted analyzed a variety of charge pumps, that include: Dickson Charge Pump (DCP), Static and Dynamic Charge Transfer Switch based charge pump (SCTS and DCTS), four-phase charge pump and the proposed voltage doubler charge pump. The main concern identified with existing charge pumps is the operation of the first stages beneath the threshold voltage and Dickson-based charge pump with a proposed clock booster was introduced that alleviates some of the performance degradation. Furthermore, for current load applications the CNSP pump remains the most effective whether operating in sub- or super- threshold. The CNSP design is proposed to offer enhanced output voltage due to the use of the proposed clock booster in the subthreshold operation. In order to effectively interface critical cells at higher voltage, with non-critical cells at lower voltage, several level shifters have been evaluated. Three conventional circuits were included in the evaluation as well as three novel level shifters proposed for the first time in the level shifter study. All six circuits were evaluated in terms of power, performance, and radiation hardness for a constant area. Voltage doublers were eliminated from consideration based on issues with noise and operating range. The proposed current mirror level shifter with dynamically-adjusted threshold was the best, which provided 13% increased performance.

The next phase of the SRR architecture design was to design a Programmable Processing Element (PPE). The PPE internal architecture consists of a 4-input LUT, carry and control logic block, conditional routing elements and a radhard storage unit. Various circuit style implementations for LUT multiplexers were tested and analyzed for subthreshold operation. Tri-

state-inverter-based multiplexers provided two times better performance values than transmission gate based multiplexer in SubT. Further, to optimize power and performance level shifter interface are used to connect SubT and SuperT processing elements. Finally, the designed modules are interconnected and verified for functionality as a MAC unit in SubT using the SRR architecture. The implemented MAC unit design is compared for power consumption with other proposed ASIC architecture. As the SRR-MAC design contains redundant designs the power consumption values are higher. However this, enables features like- radiation hardening, dynamic voltage scaling capability and reconfigurability. To illustrate the power saving achieved with SRR architecture the MAC unit is also implemented using an off-the-shelf Xilinx FPGA operating at 2.85v. The SRR-MAC design consumed three order of less power than compared to the superT Xilinx implementation. Finally, based on the features and ultra power saving operation the SRR architecture is an ideal candidate for applications needing to substitute batteries to enable self-powering and permit versatile application base including harsh environments.

## 8.2 Future Work

While the dissertation implemented a SPICE simulated version of the SRR architecture design, a comprehensive evaluation with silicon fabrication is essential especially to test the radiation hardness aspect of the design. To accomplish the self-assigned criteria, the novel modules of the SRR design were submitted for an ultra-low power design competition co-sponsored by the Defense Advanced Research Projects Agency (DARPA) and Massachusetts Institute of Technology Lincoln Laboratory (MITLL). The sponsors awarded free fabrication in a multi-run project, for the submitted SRR module proposal, with 180nm fully-depleted SOI transistor technology optimized by DARPA for subthreshold operation. Apart from implementing the proposed modules of the SRR design the project also implements other leading designs. For example, apart from implementing the design of the proposed radhard MOSAC FF the project also implements other radhard FFs such as DICE and Radhard Sense Amplifier FF. This permits a fair and comprehensive comparison with respect to implementation technology and operating conditions. Currently, the work is under progress to meet the tape-out deadline.

Further, the future work upon receiving the silicon, the post-silicon analysis is divided into two phases:

- (1) Run performance and power analysis for all the circuit modules (flip-flops, level shifters, and charge pumps) and carry out a comprehensive comparison study. Consecutively, publish the findings in a reputable related journal.
- (2) Perform radiation tolerance testing on the radhard flip-flop designs.

## 9. References

- [1] Vikas Chandra, Robert Aitken, “***Impact of Technology and Voltage Scaling on the Soft Error***”, IEEE International Symposium on Defect and Fault Tolerance of VLSI Systems, 1550-5774/08 2008.
- [2] Mark Horowitz, Thomas Indermaur, and Ricardo Gonzalez M, “***Low Power Digital Design***”, IEEE International Symposium on Low Power Electronics, pp. 8-11, 1994.
- [3] Ameet Chavan, “***Design and Test of Reconfigurable Data Path Processor using Xilinx Development Tools***”, MSEE thesis August 2003.
- [4] ***Digital Integrated Circuits: A Design Perspective***, Second Ed., Jan Rabaey, Anantha Chandrakasan, Borivoje Nikolic.
- [5] ***Low Power Design Methodologies***, Rabaey, Jan M.; Pedram, Massoud (Eds.) 1995, 388 p., ISBN: 978-0-7923-9630-7
- [6] James T. Kao, Anantha Chandrakasan, “***Dual-Threshold Voltage Techniques for Low Power Digital Circuits***”, IEEE Journal of Solid State Circuits, Vol.35., July 2000.
- [7] Granlund, T., Granbom, B., and Olsson, N, “***Soft error rate increase for new generations of SRAMS***”, IEEE Transactions on Nuclear Science 2003, 50(6):2065 2068.
- [8] Mohanram, K. and Touba, N. A., “***Cost-effective approach for reducing soft error failure rate in logic circuits***. International Test Conference 2003, pages 893-901.
- [9] Shera E. Kerns, B.D. Shafer, “***The Design of Radiation-Hradened ICs for Space: A compendium of Approaches***”, Invited paper, IEEE Processdings Vol. 76, Nov. 1988
- [10] Kerry Bernstein , Norman J. Rohrer (both from IBM Microelectronics) “***SOI Circuit Design Concepts***” Springer 1<sup>st</sup> edition ISBN-10: 0792377621
- [11] Nicolaidis, M. “***Design for soft error mitigation***”, Device and Materials Reliability, IEEE Transactions on Volume 5, Issue 3, Sept. 2005 Page(s):405 – 418 Digital Object Identifier 10.1109/TDMR.2005.855790.
- [12] Baloch, S. Arslan, T. Stoica, A ., “***Design of a single event upset (SEU) mitigation technique for programmable device***”, Inst. for Syst. Level Integration, Alba Centre, Livingston, UK; Quality Electronic Design, 2006. ISQED'06 7th International Symposium on Publication Date: 27-29 March 2006.



- [13] C, Carmichael, Xilinx, 11/01, ***“Triple module redundancy design techniques for Virtex FPGAs”*** (xAPP197, v1.0).
- [14] ***“Virtex architecture guide”***, Xilinx, San Jose, CA, 9/00.
- [15] Bryant, A.; et. al, ***“Low-Power CMOS at  $V_{dd}=4kT/q$ ”***, Device Research Conf., pp. 25-27, 2001.
- [16] Wang, A.; Chandrakasan A, ***“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, January, 2005
- [17] Soeleman, H.; Roy, K.; Paul, B. ***Robust Ultra-Low-Power Subthreshold DTMOS Logic.*** Intl. Symp. Low-Power Electronics Design, pp. 25-30, 2000.
- [18] Kobayashi, T.; Sakurai, T. ***Self-Adjusting Threshold-Voltage Scheme (SATS) for Low Voltage High-Speed Operation.*** Custom Integrated Circuits Conf, pp. 271-274, May 1994
- [19] Assaderaghi, F.; Sinitsky, D.; Parke, S.; Bokor, J.; Ko, P. ***Dynamic Threshold-Voltage MOSFET (DTMOS) for Ultra-Low Voltage VLSI.*** IEEE Trans. on Electron Devices. Vol. 44, No.3 March 1993.
- [20] B. Graniello, A. Chavan, B. Rodriguez, E. MacDonald, ***“Optimized Circuit Styles for Subthreshold Logic,”*** International Electro Conf., Oct. 2005.
- [21] Y. Berg, D. Wisland, T. Lande, ***“Ultra-Low-Voltage/ Low-Power Digital Floating Circuits,”*** IEEE Transactions on Circuits and SystemsII, pp. 930-936, 1999.
- [22] B. Calhoun, A. Chandrakasan, ***“Ultra-Dynamic Voltage Scaling (UDVS) using sub-threshold operation and local voltage dithering,”*** IEEE Journal of Solid-State Circuits, Vol. 41, No. 1, pp. 238 – 245, Jan 2006.
- [23] Nowka, K.; Carpenter, G.; MacDonald, E.; Ngo, H.; Brock, B.; Nguyen. T. ***A 32-bit PowerPC System-on-a-Chip with Support for Dynamic Voltage and Frequency Scaling.*** IEEE Journal of Solid-State Circuits, Vol. 37, No. 11, pp. 1441-147, Nov. 2002.
- [24] D. Markovic, B. Nikolic, R. Brodersen, ***“Analysis and Design of Low-Energy Flip-Flops,”*** International Symposium on Low Power Electronics and Design, pp. 52-55, Aug. 2001.
- [25] V. Stojanovic, V. Oklobdzija, R. Bajwa, ***“A Unified Approach in the Analysis of Latches and Flip-Flops for Low-Power Systems,”*** International Symposium on Low Power Electronics and Design, pp. 227-232, Aug, 1998.

- [26] A. Chavan, G. Dukle, B. Graniello, E. MacDonald, “**Robust Ultra-Low Power Subthreshold Logic Flip-flop Design for Reconfigurable Architectures**”, IEEE International Conference on Reconfigurable Computing And FPGA’s 2006, ISBN:1-4244-0689-7.
- [27] N. Nedovic and V.G. Oklobdzija.”**Hybrid Latch Flip-Flop with Improved Power Efficiency**”, 13th Symposium on Integrated Circuits and Systems Design proceedings 18-24 Sept. 2000 Page(s):211 – 215
- [28] V. Oklobdzija, “**Clocking and Clocked Storage Elements for a Multi-Gigahertz Environment**,”, *IBM Journal of Research*, Vol 47, No. 5, pp. 567-583, Dec 2004.
- [29] J. Yuan and C. Svensson, “**High-Speed CMOS Circuit Technique**,” IEEE Journal of Solid-State Circuits, Vol. 24, No. 1, February 1989.
- [30] T. Monnier, F. Roche, G. Cathebras, “**Flip-Flop Hardening for Space Applications**,” *IEEE International Workshop on Memory Technology, Design and Testing*, p 104-107
- [31] T. Calin, M. Nicolaidis, R. Velazco, “**Upset Hardened memory Design for Submicron CMOS Technology**,” *IEEE Transactions on Nuclear Science*, Vol. 43, No. 6, pp. 2874-2878, Dec 1996.
- [32] W. Wang, H. Gong, “**Sense Amplifier Based RADHARD Flip-Flop Design**,” *IEEE Transaction on Nuclear Science*, Vol 51, No 6, pp. 3811-3815, Dec. 2004.
- [33] Hoang Q. Dao, BartR. Zeydel, and Vojin g. Oklobzija, “**Energy Optimization of Pielined Digital Systems Using Circuit Sizing and Supply Scaling**”. *IEEE VLSI Systems*, VOL. 14, No.2, Feb 2006.
- [34] T. Tanzawa, T Tanaka, T. Takeuchi, K. Nakamura, “**Circuit Techiques for a 1.8 V NAND Flash Memory**”, *IEEE J. Solid State Circuits*, Vol. 37, No. 1, pp. 84-89, Jan 2002.
- [35] V. George, H. Zhang, J. Rabaey, “**Design of a Low Power FPGA**”, 1999 *International Symposium on Low Power Electronics and Design*, pp. 183-193.
- [36] J. Dickson, “**On-Chip High-Voltage Generation in NMOS Integrated Circuits Using an Improved Voltage Multiplier Technique**”, *IEEE J. Solid State Circuits* Vol. 11, pp. 374-378, June 1976.
- [37] J. Wu, K. Chang, “**MOS Charge Pump for Low-Voltage Operaiton**,” *IEEE J. Solid State Circuits* Vol. 29, No. 4, pp. 461-469, April 1994.
- [38] M. Ker, S. Chen, C. Tsai, “**A New Charge Pump Circuit Dealing With Gate-Oxide Reliability Issue in Low-Voltage Processes**,” *International Conference on Circuits and Systems* , pp. 321-324, 2004.

- [39] Bhalerao, S. Chaudhary, A. Patrikar, R. “**A CMOS Low Voltage Charge Pump**,” *International Conference on VLSI Design* pp. 941-946, Jan 2007.
- [40] F. Pan, T. Samaddar, **Charge Pump Circuit Design**, McGraw-Hill Electronic Engineering, ISBN 0-07-147045-X, 2006.
- [41] F. Assaderaghi, D. Sinitsky, S. Parke, J. Bokor, P. Ko, “**Dynamic Threshold Voltage MOSFET for Ultra-Low Voltage VLSI**”, *IEEE Trans. on Electron Devices*, Vol. 44, No 3, Mar 1993.
- [42] A. Chavan, E. MacDonald, B. Graniello, “**Optimized Charge Pumps for Subthreshold Operation**”, 13<sup>th</sup> NASA symposium on VLSI Design, Post Falls, Idaho, 2007.
- [43] R. Puri, L. Stok, J. Cohn, D. S. Kung, D. Z. Pan, D. Sylvester, A. Srivastava, S. Kulkarni, “**Pushing ASIC Performance in a Power Envelope**,” *DAC-40:ACM/IEEE Design Automation Conference*, pp. 788-793, Anaheim, CA, June 2003.
- [44] Qadeer. A. Khan, Sanjay. K. Wadhwa, Kulbhushan Misri, “**A Single Supply Level Shifter for Multi-Voltage Systems**”, 19<sup>th</sup> *International Conference on VLSI Design* 2006: 557-560.
- [45] Kyoung-Hoi Koo , Jin-Ho Seo, Myeong-Lyong Ko , Jae-Whui Kim : “**A New Level-up Shifter for High Speed and Wide Range Interface in Ultra Deep Sub-Micron**”, *IEEE International symposium on Circuits and Systems, 2005. ISCAS 2005*, 1063-1065 Vol. 2.
- [46] Kyoung-Hoi Koo Jin-Ho Seo Myeong-Lyong Ko Jae- Whui Kim, “**A new level shifter in ultra deep sub-micron for low to wide range voltage applications**”, *IEEE International SOC Conference, 2004. Proceedings*, pages 155- 156.
- [47] Tan, S.C. Sun, X.W, “**Low power CMOS level shifters by bootstrapping technique**”, *IEEE electronics Letters*, Aug 2002, Volume:38, On page(s): 876- 878.
- [48] Sarvesh H. Kulkarni and Dennis Sylvester, “**High performance level conversion for dual VDD design**”, *IEEE Trans. VLSI Syst.* 12(9): 926-936 (2004)
- [49] Kanno, Y. Mizuno, H. Tanaka, K. Watanabe, T., Central Res. Lab., Hitachi Ltd., Tokyo; “**Level Converters With High Immunity to Power-Supply Bouncing for High-Speed Sub-1V LSIs**”, 2000 symposium on VLSI Circuits Digest of Technical Papers, *Publication, date: 2000* On page(s): 202-203.
- [50] Kobayashi, T.; Sakurai, T., “**Self-Adjusting Threshold-Voltage Scheme (SATS) for Low Voltage High-Speed Operation.**” *Custom Integrated Circuits Conf*, pp. 271-274, May 1994.

- [51] H. Soeleman, K. Roy, B. Paul, “***Robust Ultra Low-Power Subthreshold DTMOS Logic***,” Proc. Int. Symp. Low-Power Electronics Design, 2000, pp. 25-30.
- [52] A. Chavan, E. MacDonald, “***Ultra Low Voltage Level Shifter to Interface Sub and Super Threshold Reconfigurable Logic Cells***”, IEEE Aerospace Conference 2008
- [53] J. Rose, R. Francis, D. Lewis and P.Chow, “ ***Architecture of Field-Programmable Gate Arrays: The Effect of Logic Block Functionality on Area Efficiency***,” JSSC, Oct. 1990, pp. 127-1225
- [54] Kao, J.; Miyazaki, M.; Chandrakasan, A. ***A 175 mV Multiply-Accumulate Unit Using an Adaptive Supply Voltage and Body Biasing Architecture***. IEEE Journal of Solid-State Circuits, Vol. 37, No. 11, pp. 1545-1554, Nov. 2002
- [55] Xilinx Spartan-II FPGA family data sheet, Xilinx.com, DS001 Jun 2008.
- [56] Chalasani, S., Conard, J.M, “***A Survey of Energy Harvesting Sources for Embedded Systems***”, Proceedings of IEEE International. Conference on Wearable Computing, April 2008.

## Appendix A

### A.1 Reconfigurable Computing

There are primarily three main branches in computing, for the execution of algorithms: *General Purpose Computing*, *Application Specific Computing* and *Reconfigurable Computing*. Reconfigurable computing is intended to fill the gap between hardware design (ASICs) and software design (programs for general purpose processors) . Reconfigurable computing has the potential of achieving much higher performance than general computing. That is, algorithms written in a programming language, while maintaining a higher level of flexibility than hardware design, carry with them the overhead of the general purpose architecture and instruction set. While ASIC design remains constant once committed to physical implementation, reconfigurable computing involves manipulation of the logic within the Programmable Logic Device (PLD) at run-time. That is, the hardware configuration can be changed depending on the system's algorithm, and downloaded as needed. The typical PLD used for reconfigurable computing is an FPGA. Reconfigurable computing has several advantages as compared to general purpose computing and application specific computing: (1) greater functionality than with simpler hardware design; (2) lower system cost (much lower when compared to the expense of ASIC design and fabrication); (3) systems based on reconfigurable computing can be upgraded in the field (which extends the useful life of the system, thus reducing lifetime costs); and (4) reconfigurable computing reduces the development cycle and time to market, as there is no chip design and no prototyping cycles.

## A.2 Classification of Reconfigurable Processors

Reconfigurable architectures are broadly classified based on their granularity, programmability, and degree of coupling.

Granularity is defined as the size of the computational units within the reconfigurable logic blocks. Reconfigurable architectures normally fall into three categories: fine-grained, medium-grained, and coarse-grained.

### A.2.1 Classification based on Granularity

***Fine-grained architecture:*** The fine-grained architectures perform bit-level manipulation. They are represented by traditional FPGAs and CPLDs (Complex Programmable Logic Devices). There are many different types of FPGAs based on their gate capacity and other available features. One example is the Xilinx XC 4000 series, which has gate capacity above 1,600 gates.

***Medium-grained architecture:*** Medium-grained logic blocks are used to implement data path circuits of varying bit widths. They can efficiently execute more complex operations, such as finite state machines. As an example the Reconfigurable Data Path Processor (RDPP) architecture is discussed in detail.

***Coarse-grained architecture:*** Coarse-grained architectures are still an emerging technology, combining features of processors and FPGAs. Since coarse-grained architectures have the advantages of clock speed, power, cost, etc. over traditional FPGAs, they show great potential in embedded systems which have requirements for high performance and low power. Coarse-

grained blocks are better optimized for standard large datapath applications. Example: PipeRench reconfigurable architecture.

### **A.2.3 Classification based on programmability**

Reconfigurable architectures can be divided into three programmability classes based on the strategies by which programs may be embedded in them. This type of classification considers the number of configurations and the time at which the reconfiguration takes places [3]. They are Static Design (SD), Statically Reconfigurable Design (SRD), and Dynamically Reconfigurable Design (DRD). The main characteristics of all programmability classes are presented in Table A.1.

***Static Design (SD):*** Architecture in the SD class has a single configuration that is never changed, neither before nor after system reset. The programmable device is programmed to perform only one function that remains unchanged during the system's lifetime. This class of computer architecture does not exploit reconfiguration flexibility. The advantages are ease of implementation and short development time by the use of programmable devices.

***Statically Reconfigurable Design (SRD):*** SRD architecture has several configurations, and reconfiguration occurs only at the end of each processing task. This class utilizes the reconfigurability feature, and the programmable devices are better used. SRD hardware shows better performance compared to general purpose processors (GPPs).

***Dynamically Reconfigurable Design (DRD):*** The DRD architecture also has several configurations, but reconfiguration takes place at runtime, i.e., during the execution of a task. This class also is known as Run-Time-Reconfiguration (RTR). DRD utilizes the reconfigurability feature more efficiently than SRD. This class is implemented by using either partially programmable devices or a set of conventional programmable devices. In the latter method, while one programmable device performs the task, the others are reconfigured.

Table A.1 Summary of probability classes

Design	Number of Configurations	Reconfiguration at
SD	Single	Design time
SRD	Multiple	End of task
DRD	Multiple	Reconfiguration time

The next classification method focuses on the degree of coupling (if any) between the reconfigurable architecture and a host microprocessor. In order to use the reconfigurable computing system efficiently, the areas of program that cannot be executed by the reconfigurable logic are carried out by the host processor. The coupling can be categorized as follows: Functional unit coupling, Coprocessor coupling, Multiprocessor coupling, and Loose coupling (represented in Figure A.2).



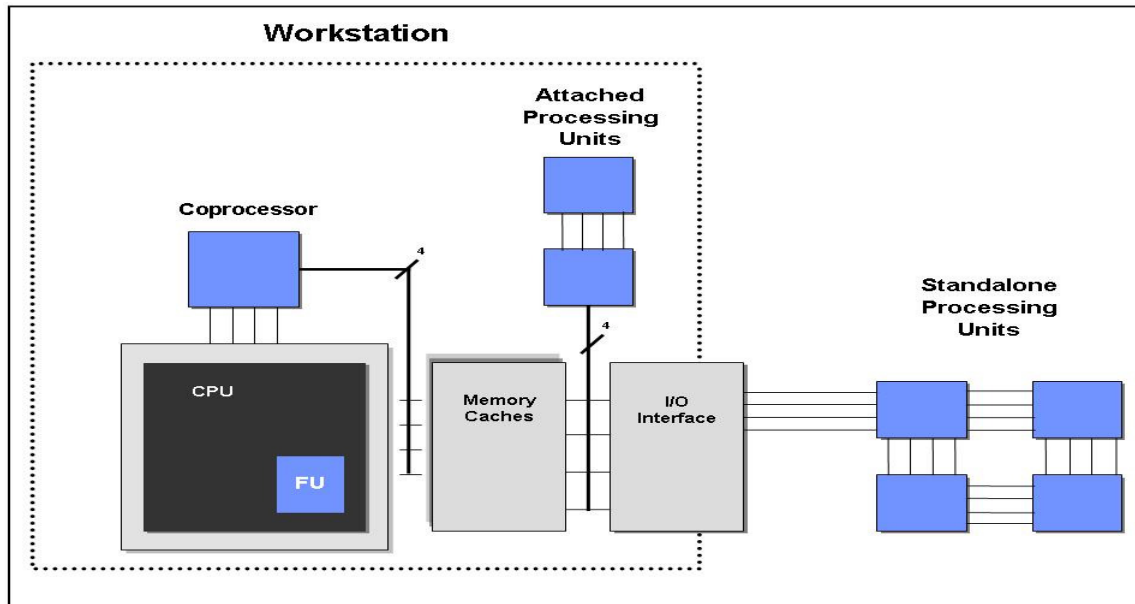


Figure A.2 Degree of coupling

### A.2.3 Classification based on Degree of coupling

**Functional Unit Coupling:** Reconfigurable hardware operates as functional multiple units on the main microprocessor data path with registers used to hold input and output operands. They are used to execute custom instructions that may change over time.

**Coprocessor coupling:** As the name suggests, a reconfigurable unit is used as a coprocessor. In general, a coprocessor is larger than a functional unit, and is able performing computations without the constant supervision of the host processor, as compared to the functional-unit coupling method discussed above. In this type of coupling, the processor initializes the reconfigurable hardware and sends the necessary data to the logic, or provides information on where the data might be found in memory.

***Multiprocessor Coupling:*** A reconfigurable unit acts as an additional processor, for example, in a multiprocessor system. The host processor's data cache is not visible to the attached reconfigurable processor. Therefore, there is a higher communication delay between the host processor and reconfigurable hardware. In this model, the reconfigurable hardware has less computational independence or workload as compared to the above two methods.

***Loosely Coupled:*** The reconfigurable hardware in the loosely coupled category is an external standalone-processing unit. This method requires the reconfigurable hardware to communicate with the host CPU through an I/O interface. Because the communication through I/O interfaces is relatively slow, this integration is useful only on systems where the communication occurs with a very low frequency.

## **Appendix B**

### **SPICE Simulation Code**

```

*****
*****SRR-MAC Proposed design
*****
.param Vsply=0.175

Vdd1      1      0 DC Vsply
Vdls      ls1    0 DC Vsply
Vtcl      itc1   0 DC Vsply
Vddmu1    mu1    0 DC Vsply
Vadd1     add1   0 DC Vsply
Vreg1     reg1   0 DC Vsply
Vddg      100    0 DC Vsply
*****

*Clock input
Vclk1  clk0  0 PULSE(0 Vsply 1.0u 200P 200P 2.96u 6u)
Vclk2  clkn  0 PULSE(0 Vsply 0.1u 200P 200P 2.96u 6u)
*****

* Input for LUT SRAxmpd
Vb1  b1  0  PWL(0n Vsply )
Vb0  blb 0  PWL(0n 0.0 )

* Boosted Config. Input for Pass Gate
VBb1  Bbl 0  PWL(0U 0  0.5u 0  0.5002u 0.50)
VBb0  Bblb 0  PWL(0U 0.50 0.5u 0.50 0.5002u 0)

*****

* Input Signal -LFSR
*Pseudo-Random inputs generated by LFSR
XLFSR0 ls1 clk ax0 ax1 ax2 ax3 ax4 ax5 ax6 ax7 bx0 bx1 bx2 bx3 bx4 bx5 bx6 bx7 LFSR
.IC v(ax0)=Vsply v(ax1)=0.0 v(ax2)=Vsply v(ax3)=Vsply v(ax4)=Vsply v(ax5)=Vsply v(ax6)=0
v(ax7)=0
+ v(bx0)=Vsply v(bx1)=Vsply (bx2)=0 v(bx3)=Vsply v(bx4)=Vsply v(bx5)=Vsply v(bx6)=0
v(bx7)=0
*****
*****

* Dummy signal for routing testing
Vduxmxmy1 56 0 PWL(0 0.0 3.5u 0.0 20.0u 0.0 20.002u Vsply)
Vduxmxmy2 57 0 PWL(0 Vsply 4.5u Vsply 20.0u Vsply 20.002u 0.0)
Vduxmxmy3 58 0 PWL(0 0.0 8.7u 0.0 20.0u 0.0 20.002u 0)

***Global Delay***
xm100 clkka clk0 1 0 hvtpfet L=40N W=6U
xm101 clkka clk0 0 0 hvtnfet L=40N W=7U
xm102 clk clkka 1 0 hvtpfet L=40N W=6U
xm103 clk clkka 0 0 hvtnfet L=40N W=7U
*Clk clk 0 0.10pf
*.IC v(clk)=Vsply v(clka)=0.0

XDELAY1 100 ax0 pma0 DELAY

```

XDELAY2 100 ax1 pma1 DELAY  
XDELAY3 100 ax2 pma2 DELAY  
XDELAY4 100 ax3 pma3 DELAY  
XDELAY5 100 ax4 pma4 DELAY  
XDELAY6 100 ax5 pma5 DELAY  
XDELAY7 100 ax6 pma6 DELAY  
XDELAY8 100 ax7 pma7 DELAY

XDELAY9 100 bx0 pmb0 DELAY  
XDELAY10 100 bx1 pmb1 DELAY  
XDELAY11 100 bx2 pmb2 DELAY  
XDELAY12 100 bx3 pmb3 DELAY  
XDELAY13 100 bx4 pmb4 DELAY  
XDELAY14 100 bx5 pmb5 DELAY  
XDELAY15 100 bx6 pmb6 DELAY  
XDELAY16 100 bx7 pmb7 DELAY

XDELAY17 100 56 dx1 DELAY  
XDELAY18 100 57 dx2 DELAY  
XDELAY19 100 58 dx3 DELAY

\*\*\*\*\*

\* \*  
\* IMPLEMENTATION \*  
\* \*

\*\*\*\*\*

\*\*\*\* Switch xmpdatrix Models Instantiation

XS0 itc1 Bbl Bblb pma0 a0 ITCMODEL  
XS1 itc1 Bbl Bblb pma1 a1 ITCMODEL  
XS2 itc1 Bbl Bblb pma2 a2 ITCMODEL  
XS3 itc1 Bbl Bblb pma3 a3 ITCMODEL  
XS4 itc1 Bbl Bblb pma4 a4 ITCMODEL  
XS5 itc1 Bbl Bblb pma5 a5 ITCMODEL  
XS6 itc1 Bbl Bblb pma6 a6 ITCMODEL  
XS7 itc1 Bbl Bblb pma7 a7 ITCMODEL

.IC v(a0)=0.0 v(a1)=0.0 v(a2)=0.0 v(a3)=0.0 v(a4)=0.0 v(a5)=0.0 v(a6)=0.0 v(a7)=0.0

XS8 itc1 Bbl Bblb pmb0 b0 ITCMODEL  
XS9 itc1 Bbl Bblb pmb1 b1 ITCMODEL  
XS10 itc1 Bbl Bblb pmb2 b2 ITCMODEL  
XS11 itc1 Bbl Bblb pmb3 b3 ITCMODEL  
XS12 itc1 Bbl Bblb pmb4 b4 ITCMODEL  
XS13 itc1 Bbl Bblb pmb5 b5 ITCMODEL  
XS14 itc1 Bbl Bblb pmb6 b6 ITCMODEL  
XS15 itc1 Bbl Bblb pmb7 b7 ITCMODEL

.IC v(b0)=0.0 v(b1)=0.0 v(b2)=0.0 v(b3)=0.0 v(b4)=0.0 v(b5)=0.0 v(b6)=0.0 v(b7)=0.0

\*\*\*\*\*

\*\*\*\*\*

\*\*\* LUT A \*\*\*

\*\*\*\*\*  
\*\*\*\*\*

\* Following LUTS perform AND operation

XLUTA mu1 clk bl blb Bbl Bblb Bbl Bbl a0 a1 b0 dx1 dx2 dx2 Aout2 P0 MLUT  
XLUTB mu1 clk bl blb Bbl Bblb Bbl Bbl a2 a3 b0 dx1 dx2 dx2 Bout2 Bout1 MLUT  
XLUTC mu1 clk bl blb Bbl Bblb Bbl Bbl a4 a5 b0 dx1 dx2 dx2 Cout2 Cout1 MLUT  
XLUTD mu1 clk bl blb Bbl Bblb Bbl Bbl a6 a7 b0 dx1 dx2 dx2 Dout2 Dout1 MLUT

\*  
.IC v(P0)=0.0 v(Aout2)=0.0 v(Bout2)=0.0 v(Bout1)=0.0 v(Cout1)=0.0 v(Cout2)=0.0 v(Dout1)=0.0  
v(Dout2)=0.0

XLUTE mu1 clk bl blb Bbl Bblb Bbl Bbl a0 a1 b1 dx1 dx2 dx2 Eout2 Eout1 MLUT  
XLUTF mu1 clk bl blb Bbl Bblb Bbl Bbl a2 a3 b1 dx1 dx2 dx2 Fout2 Fout1 MLUT  
XLUTG mu1 clk bl blb Bbl Bblb Bbl Bbl a4 a5 b1 dx1 dx2 dx2 Gout2 Gout1 MLUT  
XLUTH mu1 clk bl blb Bbl Bblb Bbl Bbl a6 a7 b1 dx1 dx2 dx2 Hout2 Hout1 MLUT

.IC v(Eout2)=0.0 v(Eout1)=0.0 v(Fout1)=0.0 v(Fout2)=0.0 v(Gout2)=0.0 v(Gout1)=0.0 v(Hout2)=0.0  
v(Hout1)=0.0

XLUTI mu1 clk bl blb Bbl Bblb Bbl Bbl a0 a1 b2 dx1 dx2 dx2 Iout2 Iout1 MLUT  
XLUTJ mu1 clk bl blb Bbl Bblb Bbl Bbl a2 a3 b2 dx1 dx2 dx2 Jout2 Jout1 MLUT  
XLUTK mu1 clk bl blb Bbl Bblb Bbl Bbl a4 a5 b2 dx1 dx2 dx2 Kout2 Kout1 MLUT  
XLUTL mu1 clk bl blb Bbl Bblb Bbl Bbl a6 a7 b2 dx1 dx2 dx2 Lout2 Lout1 MLUT

\*\*  
.IC v(Iout2)=0.0 v(Iout1)=0.0 v(Jout1)=0.0 v(Jout2)=0.0 v(Kout2)=0.0 v(Kout1)=0.0 v(Lout2)=0.0  
v(Lout1)=0.0  
\*

XLUTM mu1 clk bl blb Bbl Bblb Bbl Bbl a0 a1 b3 dx1 dx2 dx2 Mout2 Mout1 MLUT  
XLUTN mu1 clk bl blb Bbl Bblb Bbl Bbl a2 a3 b3 dx1 dx2 dx2 Nout2 Nout1 MLUT  
XLUTO mu1 clk bl blb Bbl Bblb Bbl Bbl a4 a5 b3 dx1 dx2 dx2 Oout2 Oout1 MLUT  
XLUTP mu1 clk bl blb Bbl Bblb Bbl Bbl a6 a7 b3 dx1 dx2 dx2 Pout2 Pout1 MLUT

.IC v(Mout2)=0.0 v(Mout1)=0.0 v(Nout1)=0.0 v(Nout2)=0.0 v(Oout2)=0.0 v(Oout1)=0.0  
v(Pout2)=0.0 v(Pout1)=0.0  
\*

XLUTQ mu1 clk bl blb Bbl Bblb Bbl Bbl a0 a1 b4 dx1 dx2 dx2 Qout2 Qout1 MLUT  
XLUTR mu1 clk bl blb Bbl Bblb Bbl Bbl a2 a3 b4 dx1 dx2 dx2 Rout2 Rout1 MLUT  
XLUTS mu1 clk bl blb Bbl Bblb Bbl Bbl a4 a5 b4 dx1 dx2 dx2 Sout2 Sout1 MLUT  
XLUTT mu1 clk bl blb Bbl Bblb Bbl Bbl a6 a7 b4 dx1 dx2 dx2 Tout2 Tout1 MLUT

\*  
.IC v(Qout2)=0.0 v(Qout1)=0.0 v(Rout1)=0.0 v(Rout2)=0.0 v(Sout2)=0.0 v(Sout1)=0.0 v(Tout2)=0.0  
v(Tout1)=0.0  
\*

XLUTU mu1 clk bl blb Bbl Bblb Bbl Bbl a0 a1 b5 dx1 dx2 dx2 Uout2 Uout1 MLUT  
XLUTV mu1 clk bl blb Bbl Bblb Bbl Bbl a2 a3 b5 dx1 dx2 dx2 Vout2 Vout1 MLUT  
XLUTW mu1 clk bl blb Bbl Bblb Bbl Bbl a4 a5 b5 dx1 dx2 dx2 Wout2 Wout1 MLUT  
XLUTX mu1 clk bl blb Bbl Bblb Bbl Bbl a6 a7 b5 dx1 dx2 dx2 Xout2 Xout1 MLUT

```
.IC v(Uout2)=0.0 v(Uout1)=0.0 v(Vout1)=0.0 v(Vout2)=0.0 v(Wout2)=0.0 v(Wout1)=0.0
v(Xout2)=0.0 v(Xout1)=0.0
```

\*

```
XLUTY mu1 clk bl blb Bbl Bblb Bbl Bbl a0 a1 b6 dx1 dx2 dx2 Yout2 Yout1 MLUT
XLUTZ mu1 clk bl blb Bbl Bblb Bbl Bbl a2 a3 b6 dx1 dx2 dx2 Zout2 Zout1 MLUT
XLUT11 mu1 clk bl blb Bbl Bblb Bbl Bbl a4 a5 b6 dx1 dx2 dx2 aaout2 aaout1 MLUT
XLUT12 mu1 clk bl blb Bbl Bblb Bbl Bbl a6 a7 b6 dx1 dx2 dx2 bbout2 bbout1 MLUT
```

```
.IC v(Yout2)=0.0 v(Yout1)=0.0 v(Zout1)=0.0 v(Zout2)=0.0 v(aaout2)=0.0 v(aaout1)=0.0
v(bbout2)=0.0 v(bbout1)=0.0
```

\*

```
XLUT13 mu1 clk bl blb Bbl Bblb Bbl Bbl a0 a1 b7 dx1 dx2 dx2 ccout2 ccout1 MLUT
XLUT14 mu1 clk bl blb Bbl Bblb Bbl Bbl a2 a3 b7 dx1 dx2 dx2 ddout2 ddout1 MLUT
XLUT15 mu1 clk bl blb Bbl Bblb Bbl Bbl a4 a5 b7 dx1 dx2 dx2 eeout2 eeout1 MLUT
XLUT16 mu1 clk bl blb Bbl Bblb Bbl Bbl a6 a7 b7 dx1 dx2 dx2 ffout2 ffout1 MLUT
```

\*

```
.IC v(ccout2)=0.0 v(ccout1)=0.0 v(ddout1)=0.0 v(ddout2)=0.0 v(eeout2)=0.0 v(eeout1)=0.0
v(ffout2)=0.0 v(ffout1)=0.0
```

\*\*\* Follwing LUTS perforxm 1-bit addition for xmultiplier

```
XADD0 mu1 clk bl blb Bbl Bblb Bblb Bbl Eout1 Aout2 dx3 dx3 dx3 dx3 P1 Ca01 MLUT
.IC v(P1)=0 v(Ca01)=0
```

```
XADD1 mu1 clk bl blb Bbl Bblb Bblb Bbl Bout1 Eout2 Ca01 dx3 dx3 dx3 Pa0 Ca02 MLUT
XADD2 mu1 clk bl blb Bbl Bblb Bblb Bbl Pa0 Iout1 dx3 dx3 dx3 dx3 P2 Cb00 MLUT
.IC v(Pa0)=0 v(P2)=0 v(Ca02)=0 v(Ca03)=0
```

\*

```
XADD3 mu1 clk bl blb Bbl Bblb Bblb Bbl Bout2 Fout1 Ca02 dx3 dx3 dx3 Pb0 Ca03 MLUT
XADD4 mu1 clk bl blb Bbl Bblb Bblb Bbl Pb0 Iout2 Cb00 dx3 dx3 dx3 Pb1 Cb01 MLUT
XADD5 mu1 clk bl blb Bbl Bblb Bblb Bbl Pb1 Mout1 dx3 dx3 dx3 dx3 P3 Cc00 MLUT
.IC v(Pb0)=0 v(Pb1)=0 v(P3)=0 v(Ca04)=0 v(Ca05)=0 v(Ca06)=0
```

\*\*

```
XADD6 mu1 clk bl blb Bbl Bblb Bblb Bbl Cout1 Fout2 Ca03 dx3 dx3 dx3 Pc0 Ca04 MLUT
XADD7 mu1 clk bl blb Bbl Bblb Bblb Bbl Pc0 Jout1 Cb01 dx3 dx3 dx3 Pc1 Cb02 MLUT
XADD8 mu1 clk bl blb Bbl Bblb Bblb Bbl Pc1 Mout2 Cc00 dx3 dx3 dx3 Pc2 Cc01 MLUT
XADD9 mu1 clk bl blb Bbl Bblb Bblb Bbl Pc2 Qout1 dx3 dx3 dx3 dx3 P4 Cd00 MLUT
.IC v(Pc0)=0 v(Pc1)=0 v(P4)=0 v(Ca07)=0 v(Ca08)=0 v(Ca09)=0
```

\*\*

```
XADD10 mu1 clk bl blb Bbl Bblb Bblb Bbl Cout2 Gout1 Ca04 dx3 dx3 dx3 Pd0 Ca05 MLUT
XADD11 mu1 clk bl blb Bbl Bblb Bblb Bbl Pd0 Jout2 Cb02 dx3 dx3 dx3 Pd1 Cb03 MLUT
XADD12 mu1 clk bl blb Bbl Bblb Bblb Bbl Pd1 Nout1 Cc01 dx3 dx3 dx3 Pd2 Cc02 MLUT
XADD13 mu1 clk bl blb Bbl Bblb Bblb Bbl Pd2 Qout2 Cd00 dx3 dx3 dx3 Pd3 Cd01 MLUT
XADD14 mu1 clk bl blb Bbl Bblb Bblb Bbl Pd3 Uout1 dx3 dx3 dx3 dx3 P5 Ce00 MLUT
.IC v(Pd0)=0 v(Pd1)=0 v(P5)=0 v(Ca11)=0 v(Ca12)=0 v(Ca13)=0
```

\*

```
XADD15 mu1 clk bl blb Bbl Bblb Bblb Bbl Dout1 Gout2 Ca05 dx3 dx3 dx3 Pe0 Ca06 MLUT
XADD16 mu1 clk bl blb Bbl Bblb Bblb Bbl Pe0 Kout1 Cb03 dx3 dx3 dx3 Pe1 Cb04 MLUT
XADD17 mu1 clk bl blb Bbl Bblb Bblb Bbl Pe1 Nout2 Cc02 dx3 dx3 dx3 Pe2 Cc03 MLUT
```

XADD18 mu1 clk bl blb Bbl Bblb Bblb Bbl Pe2 Rout1 Cd01 dx3 dx3 dx3 Pe3 Cd02 MLUT  
XADD19 mu1 clk bl blb Bbl Bblb Bblb Bbl Pe3 Uout2 Ce00 dx3 dx3 dx3 Pe4 Ce01 MLUT  
XADD20 mu1 clk bl blb Bbl Bblb Bblb Bbl Pe4 Yout1 dx3 dx3 dx3 dx3 P6 Cf00 MLUT  
.IC v(Pe0)=0 v(Pe1)=0 v(P6)=0 v(Ca16)=0 v(Ca17)=0 v(Ca18)=0

\*\*

XADD21 mu1 clk bl blb Bbl Bblb Bblb Bbl Dout2 Hout1 Ca06 dx3 dx3 dx3 Pf0 Ca07 MLUT  
XADD22 mu1 clk bl blb Bbl Bblb Bblb Bbl Pf0 Kout2 Cb04 dx3 dx3 dx3 Pf1 Cb05 MLUT  
XADD23 mu1 clk bl blb Bbl Bblb Bblb Bbl Pf1 Oout1 Cc03 dx3 dx3 dx3 Pf2 Cc04 MLUT  
XADD24 mu1 clk bl blb Bbl Bblb Bblb Bbl Pf2 Rout2 Cd02 dx3 dx3 dx3 Pf3 Cd03 MLUT  
XADD25 mu1 clk bl blb Bbl Bblb Bblb Bbl Pf3 Vout1 Ce01 dx3 dx3 dx3 Pf4 Ce02 MLUT  
XADD26 mu1 clk bl blb Bbl Bblb Bblb Bbl Pf4 Yout2 Cf00 dx3 dx3 dx3 Pf5 Cf01 MLUT  
XADD27 mu1 clk bl blb Bbl Bblb Bblb Bbl Pf5 ccout1 dx3 dx3 dx3 dx3 P7 Cg00 MLUT  
.IC v(Pf0)=0 v(Pf1)=0 v(P7)=0 v(Ca22)=0 v(Ca23)=0 v(Ca24)=0

\*\*

XADDx1 mu1 clk bl blb Bbl Bblb Bblb Bbl Hout2 Cb05 Ca07 dx3 dx3 dx3 Pgx Ca08 MLUT  
XADD28 mu1 clk bl blb Bbl Bblb Bblb Bbl Pgx Lout1 Cc04 dx3 dx3 dx3 Pg0 Cc05 MLUT  
XADD29 mu1 clk bl blb Bbl Bblb Bblb Bbl Pg0 Oout2 Cd03 dx3 dx3 dx3 Pg1 Cd04 MLUT  
XADD30 mu1 clk bl blb Bbl Bblb Bblb Bbl Pg1 Sout1 Ce02 dx3 dx3 dx3 Pg2 Ce03 MLUT  
XADD31 mu1 clk bl blb Bbl Bblb Bblb Bbl Pg2 Vout2 Cf01 dx3 dx3 dx3 Pg3 Cf02 MLUT  
XADD32 mu1 clk bl blb Bbl Bblb Bblb Bbl Pg3 Zout1 Cg00 dx3 dx3 dx3 Pg4 Cg01 MLUT  
XADD33 mu1 clk bl blb Bbl Bblb Bblb Bbl Pg4 ccout2 dx3 dx3 dx3 dx3 P8 Cgxx MLUT  
.IC v(Pg0)=0 v(P1)=0 v(P8)=0 v(Ca29)=0 v(Ca30)=0 v(Ca31)=0

\*

XADDx2 mu1 clk bl blb Bbl Bblb Bblb Bbl Lout2 Cc05 Ca08 dx3 dx3 dx3 Phx Ca09 MLUT  
XADD34 mu1 clk bl blb Bbl Bblb Bblb Bbl Phx Pout1 Cd04 dx3 dx3 dx3 Ph0 Cd05 MLUT  
XADD35 mu1 clk bl blb Bbl Bblb Bblb Bbl Ph0 Sout2 Ce03 dx3 dx3 dx3 Ph1 Ce04 MLUT  
XADD36 mu1 clk bl blb Bbl Bblb Bblb Bbl Ph1 Wout1 Cf02 dx3 dx3 dx3 Ph2 Cf03 MLUT  
XADD37 mu1 clk bl blb Bbl Bblb Bblb Bbl Ph2 Zout2 Cg01 dx3 dx3 dx3 Ph3 Cg02 MLUT  
XADD38 mu1 clk bl blb Bbl Bblb Bblb Bbl Ph3 ddout1 Cgxx dx3 dx3 dx3 P9 Cexx MLUT  
.IC v(Ph0)=0 v(Ph1)=0 v(P9)=0 v(Ca35)=0 v(Ca36)=0 v(Ca37)=0

\*

XADDx3 mu1 clk bl blb Bbl Bblb Bblb Bbl Pout2 Cd05 Ca09 dx3 dx3 dx3 Pix Ca10 MLUT  
XADD39 mu1 clk bl blb Bbl Bblb Bblb Bbl Pix Tout1 Ce04 dx3 dx3 dx3 Pi0 Ce05 MLUT  
XADD40 mu1 clk bl blb Bbl Bblb Bblb Bbl Pi0 Wout2 Cf03 dx3 dx3 dx3 Pi1 Cf04 MLUT  
XADD41 mu1 clk bl blb Bbl Bblb Bblb Bbl Pi1 aaout1 Cg02 dx3 dx3 dx3 Pi2 Cg03 MLUT  
XADD42 mu1 clk bl blb Bbl Bblb Bblb Bbl Pi2 ddout2 Cexx dx3 dx3 dx3 P10 Cdxx MLUT  
.IC v(Pi0)=0 v(Pi1)=0 v(P10)=0 v(Ca40)=0 v(Ca41)=0 v(Ca42)=0

XADDx4 mu1 clk bl blb Bbl Bblb Bblb Bbl Tout2 Ce05 Ca10 dx3 dx3 dx3 Pjx Ca11 MLUT  
XADD43 mu1 clk bl blb Bbl Bblb Bblb Bbl Pjx Xout1 Cf04 dx3 dx3 dx3 Pj0 Cf05 MLUT  
XADD44 mu1 clk bl blb Bbl Bblb Bblb Bbl Pj0 aaout2 Cg03 dx3 dx3 dx3 Pj1 Cg04 MLUT  
XADD45 mu1 clk bl blb Bbl Bblb Bblb Bbl Pj1 eeout1 Cdxx dx3 dx3 dx3 P11 Ccxx MLUT  
.IC v(Pj0)=0 v(Pj1)=0 v(P11)=0 v(Ca44)=0 v(Ca45)=0 v(Ca46)=0

XADDx5 mu1 clk bl blb Bbl Bblb Bblb Bbl Xout2 Cf05 Ca11 dx3 dx3 dx3 Pkx Ca12 MLUT  
XADD46 mu1 clk bl blb Bbl Bblb Bblb Bbl Pkx bbout1 Cg04 dx3 dx3 dx3 Pk0 Cg05 MLUT  
XADD47 mu1 clk bl blb Bbl Bblb Bblb Bbl Pk0 eeout2 Ccxx dx3 dx3 dx3 P12 Cbxx MLUT  
.IC v(Pk0)=0 v(P12)=0 v(Ca47)=0 v(Ca48)=0

\*\*

XADDx6 mu1 clk bl blb Bbl Bblb Bblb Bbl bbout2 Cg05 Ca12 dx3 dx3 dx3 PL0 Ca13 MLUT



XADD48 mu1 clk bl blb Bbl Bblb Bblb Bbl PL0 ffout1 Cbxx dx3 dx3 dx3 P13 Caxx MLUT  
.IC v(P13)=0 v(Ca49)=0  
\*\*

XADDx7 mu1 clk bl blb Bbl Bblb Bblb Bbl ffout2 Ca13 Caxx dx3 dx3 dx3 P14 P15 MLUT  
.IC v(P14)=0 v(P15)=0  
\*

xm403 clke clk0 100 0 hvtpfet L=40N W=3U  
xm404 clke clk0 0 0 hvtnfet L=40N W=3.5U  
xm405 clkf clke 100 0 hvtpfet L=40N W=3U  
xm406 clkf clke 0 0 hvtnfet L=40N W=3.5U

\*\*\*\*\* 16 bit Adder

XADD50 add1 clkf bl blb Bbl Bblb Bblb Bbl P0 MAC0 dx3 dx3 dx3 dx3 ADD0 Ct1 MLUT  
XADD51 add1 clkf bl blb Bbl Bblb Bblb Bbl P1 MAC1 Ct1 dx3 dx3 dx3 dx3 ADD1 Ct2 MLUT  
XADD52 add1 clkf bl blb Bbl Bblb Bblb Bbl P2 MAC2 Ct2 dx3 dx3 dx3 dx3 ADD2 Ct3 MLUT  
XADD53 add1 clkf bl blb Bbl Bblb Bblb Bbl P3 MAC3 Ct3 dx3 dx3 dx3 dx3 ADD3 Ct4 MLUT  
.IC v(ADD0)=0 v(ADD1)=0 v(ADD2)=0 v(ADD3)=0 v(Ct1)=0 v(ct2)=0 v(ct3)=0 v(ct4)=0  
\*

XADD54 add1 clkf bl blb Bbl Bblb Bblb Bbl P4 MAC4 Ct4 dx3 dx3 dx3 dx3 ADD4 Ct5 MLUT  
XADD55 add1 clkf bl blb Bbl Bblb Bblb Bbl P5 MAC5 Ct5 dx3 dx3 dx3 dx3 ADD5 Ct6 MLUT  
XADD56 add1 clkf bl blb Bbl Bblb Bblb Bbl P6 MAC6 Ct6 dx3 dx3 dx3 dx3 ADD6 Ct7 MLUT  
XADD57 add1 clkf bl blb Bbl Bblb Bblb Bbl P7 MAC7 Ct7 dx3 dx3 dx3 dx3 ADD7 Ct8 MLUT  
.IC v(ADD4)=0 v(ADD5)=0 v(ADD6)=0 v(ADD7)=0 v(Ct5)=0 v(ct6)=0 v(ct7)=0 v(ct8)=0  
\*

XADD60 add1 clkf bl blb Bbl Bblb Bblb Bbl P8 MAC8 Ct8 dx3 dx3 dx3 dx3 ADD8 Ct9 MLUT  
XADD61 add1 clkf bl blb Bbl Bblb Bblb Bbl P9 MAC9 Ct9 dx3 dx3 dx3 dx3 ADD9 Ct10 MLUT  
XADD62 add1 clkf bl blb Bbl Bblb Bblb Bbl P10 MAC10 Ct10 dx3 dx3 dx3 dx3 ADD10 Ct11 MLUT  
XADD63 add1 clkf bl blb Bbl Bblb Bblb Bbl P11 MAC11 Ct11 dx3 dx3 dx3 dx3 ADD11 Ct12 MLUT  
.IC v(ADD8)=0 v(ADD9)=0 v(ADD10)=0 v(ADD11)=0 v(Ct9)=0 v(ct10)=0 v(ct11)=0 v(ct12)=0

XADD64 add1 clkf bl blb Bbl Bblb Bblb Bbl P12 MAC12 Ct12 dx3 dx3 dx3 dx3 ADD12 Ct13 MLUT  
XADD65 add1 clkf bl blb Bbl Bblb Bblb Bbl P13 MAC13 Ct13 dx3 dx3 dx3 dx3 ADD13 Ct14 MLUT  
XADD66 add1 clkf bl blb Bbl Bblb Bblb Bbl P14 MAC14 Ct14 dx3 dx3 dx3 dx3 ADD14 Ct15 MLUT  
XADD67 add1 clkf bl blb Bbl Bblb Bblb Bbl P15 MAC15 Ct15 dx3 dx3 dx3 dx3 ADD15 Ct16 MLUT  
.IC v(ADD12)=0 v(ADD13)=0 v(ADD14)=0 v(ADD15)=0 v(Ct13)=0 v(ct14)=0 v(ct15)=0 v(ct16)=0  
\*  
\*

\*\*\*\*\* 4-bit MAC register

XREG0 reg1 clkf bl blb Bbl Bblb Bblb Bblb dx1 MAC0 dx2 dx3 dx3 dx3 MAC0 rx0 MLUT  
XREG1 reg1 clkf bl blb Bbl Bblb Bblb Bblb dx1 MAC1 dx2 dx3 dx3 dx3 MAC1 rx1 MLUT  
XREG2 reg1 clkf bl blb Bbl Bblb Bblb Bblb dx1 MAC2 dx2 dx3 dx3 dx3 MAC2 rx2 MLUT  
XREG3 reg1 clkf bl blb Bbl Bblb Bblb Bblb dx1 MAC3 dx2 dx3 dx3 dx3 MAC3 rx3 MLUT  
.IC v(MAC0)=0 v(MAC1)=0 v(MAC2)=0 v(MAC3)=0

```

*
XREG4 reg1 clkf bl blb Bbl Bblb Bblb Bblb dx1 MAC4 dx2 dx3 dx3 dx3 MAC4 rx4 MLUT
XREG5 reg1 clkf bl blb Bbl Bblb Bblb Bblb dx1 MAC5 dx2 dx3 dx3 dx3 MAC5 rx5 MLUT
XREG6 reg1 clkf bl blb Bbl Bblb Bblb Bblb dx1 MAC6 dx2 dx3 dx3 dx3 MAC6 rx6 MLUT
XREG7 reg1 clkf bl blb Bbl Bblb Bblb Bblb dx1 MAC7 dx2 dx3 dx3 dx3 MAC7 rx7 MLUT
.IC v(MAC4)=0 v(MAC5)=0 v(MAC6)=0 v(MAC7)=0
**
XREG8 reg1 clkf bl blb Bbl Bblb Bblb Bblb dx1 MAC8 dx2 dx3 dx3 dx3 MAC8 rx0 MLUT
XREG9 reg1 clkf bl blb Bbl Bblb Bblb Bblb dx1 MAC9 dx2 dx3 dx3 dx3 MAC9 rx1 MLUT
XREG10 reg1 clkf bl blb Bbl Bblb Bblb Bblb dx1 MAC10 dx2 dx3 dx3 dx3 MAC10 rx2 MLUT
XREG11 reg1 clkf bl blb Bbl Bblb Bblb Bblb dx1 MAC11 dx2 dx3 dx3 dx3 MAC11 rx3 MLUT
*.IC v(MAC8)=0 v(MAC9)=0 v(MAC10)=0 v(MAC11)=0
*
XREG12 reg1 clkf bl blb Bbl Bblb Bblb Bblb dx1 MAC12 dx2 dx3 dx3 dx3 MAC12 rx4 MLUT
XREG13 reg1 clkf bl blb Bbl Bblb Bblb Bblb dx1 MAC13 dx2 dx3 dx3 dx3 MAC13 rx5 MLUT
XREG14 reg1 clkf bl blb Bbl Bblb Bblb Bblb dx1 MAC14 dx2 dx3 dx3 dx3 MAC14 rx6 MLUT
XREG15 reg1 clkf bl blb Bbl Bblb Bblb Bblb dx1 MAC15 dx2 dx3 dx3 dx3 MAC15 rx7 MLUT
.IC v(MAC12)=0 v(MAC13)=0 v(MAC14)=0 v(MAC15)=0
*

*****
*****
*
* SUB-CIRCUITS
*
*****
*****
*** Linear Feedback Shift Register circuit
.SUBCKT LFSR 1 clk x0 x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 x13 x14 x15

XFF1 1 clk zout x0 PROFF
XFF2 1 clk x0 x1 PROFF
XFF3 1 clk x1 x2 PROFF
XFF4 1 clk x2 x3 PROFF
XFF5 1 clk x3 x4 PROFF
XFF6 1 clk x4 x5 PROFF
XFF7 1 clk x5 x6 PROFF
XFF8 1 clk x6 x7 PROFF
XFF9 1 clk x7 x8 PROFF
XFF10 1 clk x8 x9 PROFF
XFF11 1 clk x9 x10 PROFF
XFF12 1 clk x10 x11 PROFF
XFF13 1 clk x11 x12 PROFF
XFF14 1 clk x12 x13 PROFF
XFF15 1 clk x13 x14 PROFF
XFF16 1 clk x14 x15 PROFF

XEXOR0 1 x13 x15 z1 EXOR
XEXOR1 1 z1 x12 z2 EXOR
XEXOR2 1 z2 x10 zout EXOR

```

```

.ENDS
*****

** EXOR circuit

.SUBCKT EXOR 1 2 3 OUT
XNAND0 1 2 3 X1 NAND
*** A ex-or B
XNAND1 1 x1 2 x2 NAND
XNAND2 1 x1 3 x3 NAND
XNAND3 1 x2 x3 OUT NAND
.ENDS

*****

.SUBCKT MLUT 1 clk bl blb Bbl Bblb w1 w2 2 3 4 5 6 7 lout Carout
XUXLA 1 blb bl bl blb bl blb blb bl blb blb blb blb blb blb 2 3 4 5 t2 t3 yout MUX161
XCLA 1 2 3 4 Ca CRLG

***AND GAtE 1**
XNDLUT1 1 2 4 X1 NAND
xm1pd Px1 X1 1 0 hvtpfet L=40N W=2U *.IC = 0 0 0
xm2pd Px1 X1 0 0 hvtnfet L=40N W=2U *.IC = 0 0 0
*****
XU1A 1 Ca Px1 w1 Ca2 TI21
XDLT1 1 Ca2 Carout DELAY
***AND GATE 2***
XNDLUT2 1 3 4 Y1 NAND
xm3pd Px2 Y1 1 0 hvtpfet L=40N W=2U *.IC = 0 0 0
xm4pd Px2 Y1 0 0 hvtnfet L=40N W=2U *.IC = 0 0 0
*****
XU1B 1 yout Px2 w1 Intout TI21
*****
XDFFA 1 clk Px2 Adfout PROFF *Px2 instead of yout
XU1C 1 Adfout Intout w2 fout TI21
XDLT2 1 fout lout DELAY
.ENDS
*****

*****MAC Redundant Proposed FF design
*.SUBCKT REDNFF 1 clk Din OUT

*XP1FF 1 clk Din OUTA PROFF
*XP2FF 1 clk Din OUTB PROFF

*xMa1 x1a OUTA 1 0 hvtpfet L=40N W=4U
*xMa2 OUTC OUTB x1a 0 hvtpfet L=40N W=4U
*xMa3 OUTC OUTA y1a 0 hvtnfet L=40N W=2U
*xMa4 y1a OUTB 0 0 hvtnfet L=40N W=2U

```

```

*xMa5  OUT  OUTC  1  0  hvtpfet L=40N W=0.5U
*xMa6  OUT  OUTC  0  0  hvtnfet L=40N W=0.2U
*xMa7  OUTC OUT  1  0  hvtpfet L=40N W=0.5U
*xMa8  OUTC OUT  0  0  hvtnfet L=40N W=0.2U

```

\*.ENDS

\*\*\*\*\*

.SUBCKT PROFF 1 clk Din OUT

\*\* Inv Din

```

xMx1  Dinb  Din  1  0  hvtpfet L=40N W=2U
xMy1  Dinb  Din  0  0  hvtnfet L=40N W=1U

```

\*\* Inv clk

```

xMx4  clk  clk  1  0  hvtpfet L=40N W=2U
xMy4  clk  clk  0  0  hvtnfet L=40N W=1U

```

\*CIRCUIT ELEMENT

\* d g s e p b

\*LATCH 1 with tranmission gate

```

xMa1  x1a  Din  1  0  hvtpfet L=40N W=1U
xMa2  A    clk  x1a 0  hvtpfet L=40N W=1U
xMa3  A    clk  y1a 0  hvtnfet L=40N W=0.5U
xMa4  y1a  Din  0  0  hvtnfet L=40N W=0.5U

xMa5  w1a  Dinb  1  0  hvtpfet L=40N W=1U
xMa6  B    clk  w1a 0  hvtpfet L=40N W=1U
xMa7  B    clk  z1a 0  hvtnfet L=40N W=0.5U
xMa8  z1a  Dinb  0  0  hvtnfet L=40N W=0.5U

xMa9  x2a  Din  1  0  hvtpfet L=40N W=1U
xMa10 E    clk  x2a 0  hvtpfet L=40N W=1U
xMa11 E    clk  y2a 0  hvtnfet L=40N W=0.5U
xMa12 y2a  Din  0  0  hvtnfet L=40N W=0.5U

xMa13 w2a  Dinb  1  0  hvtpfet L=40N W=1U
xMa14 F    clk  w2a 0  hvtpfet L=40N W=1U
xMa15 F    clk  z2a 0  hvtnfet L=40N W=0.5U
xMa16 z2a  Dinb  0  0  hvtnfet L=40N W=0.5U

```

XLATCH1 1 A B F E PROLATCH

\*LATCH 2 with tri-state gate at input

```

xMb1  x1b  A    1  0  hvtpfet L=40N W=1U
xMb2  Qa   clk  x1b 0  hvtpfet L=40N W=1U
xMb3  Qa   clk  y1b 0  hvtnfet L=40N W=0.5U
xMb4  y1b  A    0  0  hvtnfet L=40N W=0.5U

xMb5  w1b  B    1  0  hvtpfet L=40N W=1U
xMb6  Qb   clk  w1b 0  hvtpfet L=40N W=1U

```

```

xMb7   Qb   clk  z1b  0   hvtnfet L=40N W=0.5U
xMb8   z1b   B   0    0   hvtnfet L=40N W=0.5U

xMb9   x2b   E   1    0   hvtpfet L=40N W=1U
xMb10  Q     clkb x2b  0   hvtpfet L=40N W=1U
xMb11  Q     clk  y2b  0   hvtnfet L=40N W=0.5U
xMb12  y2b   E   0    0   hvtnfet L=40N W=0.5U

xMb13  w2b   F    1    0   hvtpfet L=40N W=1U
xMb14  QN    clkb w2b  0   hvtpfet L=40N W=1U
xMb15  QN    clk  z2b  0   hvtnfet L=40N W=0.5U
xMb16  z2b   F    0    0   hvtnfet L=40N W=0.5U

```

\*

```
XLATCH2 1 Qa Qb QN Q PROLATCH
```

```
XBUFA1 1 Qa OUT DELAY
```

```
.ENDS
```

```
*****
```

```
.SUBCKT PROLATCH 1 A B E F
```

```
*Latch
```

```

xM1   aw   F    1    0   hvtpfet L=40N W=1U
xM2   B    A    aw   0   hvtpfet L=40N W=1U
xM3   B    F    cw   0   hvtnfet L=40N W=1.5U
xM4   cw   A    0    0   hvtnfet L=40N W=1.5U

```

```

xM5   ax   E    1    0   hvtpfet L=40N W=1U
xM6   A    B    ax   0   hvtpfet L=40N W=1U
xM7   A    E    cx   0   hvtnfet L=40N W=1.5U
xM8   cx   B    0    0   hvtnfet L=40N W=1.5U

```

```
*Cross coupled PFETS
```

```

xM11  E    F    1    0   hvtpfet L=40N W=1U
xM12  F    E    1    0   hvtpfet L=40N W=1U

```

```

xM13  E    F    X    0   hvtnfet L=40N W=2U
xM14  F    E    Y    0   hvtnfet L=40N W=2U
xM15  X    A    0    0   hvtnfet L=40N W=2U
xM16  Y    B    0    0   hvtnfet L=40N W=2U

```

```
.ENDS
```

```
*****
```

```
*****
```

```
* Interconnect xmpdodel
```

```
*****
```

```
.SUBCKT ITCMODEL 1 Bbl Bblb 2 OUT
```

```

xm1pd 2 Bbl x1 0 hvtnfet L=40N W=1U
xm2pd x1 Bbl x2 0 hvtnfet L=40N W=1U

```

```

xm3pd x2 Bbl x3 0 hvtnfet L=40N W=1U
xm4pd x3 Bbl OUT 0 hvtnfet L=40N W=1U
*SRAXmpd conf. cells for Matrix A
*xm1pd x1 2 1 0 hvtpfet L=40N W=1U
*xm2pd x1 Bblb 3 0 hvtpfet L=40N W=1U
*xm3pd y1 2 3 0 hvtnfet L=40N W=1U
*xm4pd y1 Bbl 0 0 hvtnfet L=40N W=1U

*xm5pd x2 3 1 0 hvtpfet L=40N W=1U
*xm6pd x2 Bblb OUT 0 hvtpfet L=40N W=1U
*xm7pd y2 3 OUT 0 hvtnfet L=40N W=1U
*xm8pd y2 Bbl 0 0 hvtnfet L=40N W=1U
.ENDS
*****
***** CARRY LOIC
*****
.SUBCKT CRLG 1 2 3 4 5
*** A and B
XNAND0 1 2 3 X1 NAND
*** A ex-or B
XNAND1 1 x1 2 x2 NAND
XNAND2 1 x1 3 x3 NAND
XNAND3 1 x2 x3 x4 NAND

****
XNAND4 1 x4 4 x5 NAND
**** OR gate
XNAND5 1 x1 x5 5 NAND
.ENDS

*****
***** 16:1 xmpdux ixmplementation
*****
.SUBCKT MUX161 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
XTI810 1 2 3 4 5 6 7 8 9 18 19 20 22 TI81
XTI811 1 10 11 12 13 14 15 16 17 18 19 20 23 TI81
XTI214 1 22 23 21 24 TI21
.ENDS
*****
.SUBCKT TI81 1 2 3 4 5 6 7 8 9 10 11 12 13
XTI410 1 2 3 4 5 10 11 14 TI41
XTI411 1 6 7 8 9 10 11 15 TI41
XTI213 1 14 15 12 13 TI21
.ENDS
*****
.SUBCKT TI41 1 2 3 4 5 6 7 8
XTI210 1 2 3 6 9 TI21
XTI211 1 4 5 6 10 TI21
XTI212 1 9 10 7 8 TI21
.ENDS

```

```

*****
.SUBCKT TI21 1 2 3 4 5
* Inverter for Select signal
xm1pd 7 4 1 0 hvtpfet L=40N W=1U
xm2pd 7 4 0 0 hvtnfet L=40N W=0.5U
* Inverter at output
xm3pd 5 16 1 0 hvtpfet L=40N W=1U *.IC = 0 0 0
xm4pd 5 16 0 0 hvtnfet L=40N W=0.5U *.IC = 0 Vsply 0
* Tri-Inverter 1
xm5pd 12 2 1 0 hvtpfet L=40N W=1U *.IC = 0 0 0
xm6pd 16 4 12 0 hvtpfet L=40N W=1U *.IC = 0 0 0
xm7pd 16 7 13 0 hvtnfet L=40N W=0.5U *.IC = 0 Vsply 0
xm8pd 13 2 0 0 hvtnfet L=40N W=0.5U *.IC = 0 Vsply 0
* Tri-inverter 2
xm9pd9 14 3 1 0 hvtpfet L=40N W=1U *.IC = 0 0 0
xm10pd 16 7 14 0 hvtpfet L=40N W=1U *.IC = 0 0 0
xm11pd 16 4 15 0 hvtnfet L=40N W=0.5U *.IC = 0 Vsply 0
xm12pd 15 3 0 0 hvtnfet L=40N W=0.5U *.IC = 0 Vsply 0
.ENDS

*****
***** NAND GATE *****
*****
.SUBCKT NAND 1 2 3 4
xm1pd 4 3 1 0 hvtpfet L=40N W=1U
xm2pd 4 2 1 0 hvtpfet L=40N W=1U
xm3pd 4 3 5 0 hvtnfet L=40N W=2U
xm4pd 5 2 0 0 hvtnfet L=40N W=2U
.ENDS

*****
*** DELAY
*****
.SUBCKT DELAY 1 2 3
xm1pd 4 2 1 0 hvtpfet L=40N W=1U
xm2pd 4 2 0 0 hvtnfet L=40N W=1.5U
xm3pd 3 4 1 0 hvtpfet L=40N W=1U
xm4pd 3 4 0 0 hvtnfet L=40N W=1.5U
.ENDS

*****
*** BUFFER
*****
.SUBCKT BUFFER 1 2 3
xm1pd 4 2 1 0 hvtpfet L=40N W=1U *.IC = 0 0 0
xm2pd 4 2 0 0 hvtnfet L=40N W=1.5U *.IC = 0 Vsply 0
xm3pd 3 4 1 0 hvtpfet L=40N W=1U *.IC = 0 Vsply 0
xm4pd 3 4 0 0 hvtnfet L=40N W=1.5U *.IC = 0 0 0
.ENDS
*****
* Load Capacitance

```

```

.param sigma=0
.inc "../design.inc"
.option symb=1
*.OPTION CONVERGE=5
.options post measout
*.probe v(clk) v(out) v(din)
.tran 0.1n 20u 0n
.print tran v(a0) v(a1)

.measure avg_power AVG power from=1u to=19u

.meas tran Itot INTEGRAL i(Vdd1) from=1.0U to=19U
.meas tran Energy param='Itot*Vsply'
.meas tran AVG_P1 param='Energy/18u'

.meas tran Itls INTEGRAL i(Vdls) from=1.0U to=19U
.meas tran En_ls param='Itls*Vsply'
.meas tran AVG_Pls param='En_ls/18u'

.meas tran Itot1 INTEGRAL i(Vitc1) from=1.0U to=19U
.meas tran Energy1 param='Itot1*Vsply'
.meas tran AVG_Pitc param='Energy1/18u'

.meas tran Itot2 INTEGRAL i(Vddmu1) from=1.0U to=19U
.meas tran Energy2 param='Itot2*Vsply'
.meas tran AVG_Pmult param='Energy2/18u'

.meas tran Itot3 INTEGRAL i(Vadd1) from=1.0U to=19U
.meas tran Energy3 param='Itot3*Vsply'
.meas tran AVG_Padd param='Energy3/18u'

.meas tran Itot_reg INTEGRAL i(Vreg1) from=1.0U to=19U
.meas tran En_reg param='Itot_reg*Vsply'
.meas tran AVG_Preg param='En_reg/18u'
.end

```



```
*****
*Constant-Current LEVEL SHIFTER CKT. for logic interface
*****
```

```
Vdd1 1 0 0.175
Vddbuf 100 0 0.175
Vddsuf 500 0 1.00
Vdd2 2 0 1.0
Vdd3 3 0 1.0
```

```
.param VsubT = 0.175
.param VsuperT = 1.0
```

```
*Clock input
Vclk clk0 0 PULSE(0 0.175 0.5u 200P 200P 2.996u 6u)
Vclksup clk2 0 PULSE(0 1.0 1.0u 200P 200P 0.0496u 0.1u)
Ven En 0 PWL (0u 0 0.5u 0 0.5002u 1.0)
*****
```

```
*IR1 8 0 PULSE (0 -0.4m 2.2u 50p 100p 1n 10u)
```

```
*D input
VdI0 in 0 PWL( 0.0u 0.175
+ 5.2350u 0.175
+ 5.2352u 0.0
+ 11.2200u 0.0
+ 11.2202u 0.175
+ 17.2200u 0.175
+ 17.2202u 0)
```

```
XDELAY1 100 clk0 clk1 DELAY
XDELAY2 100 in Din DELAY
```

```
XTESTSUB 1 Din clk1 SUBOUT TESTFF
```

```
*****
xMc7 6 SUBOUT 1 0 hvtpfet L=40n W=2U
xMc8 6 SUBOUT 0 0 hvtnfet L=40n W=1U
```

```
*****
```

```
*Current meters
V5 a5 5 0
V8 a8 8 0
*CIRCUIT ELEMENT
* d g s
```

```
xM1 a5 5 3 0 hvtpfet L=40n W=1U
xM2 a8 5 3 0 hvtpfet L=40n W=1U
```

```
*xM3 8 6 x1 0 hvtpfet L=40N W=1U
```

```

*xM4  5  SUBOUT  x2  0  hvtpfet L=40N W=1U
xM5   5  6      0  0  hvtnfet L=40N W=2U
xM6   8  SUBOUT  0  0  hvtnfet L=40n W=2U

***INV
xM7  LSOUT  8      3  0  hvtpfet L=40n W=2U
xM8  LSOUT  8      0  0  hvtnfet L=40n W=1U
*xM9  LSOUT  LSOUT0  500  0  hvtpfet L=40n W=2U
*xM10 LSOUT  LSOUT0  0  0  hvtnfet L=40n W=1U

```

```

XTESTSUPER 2 LSOUT clk2 SUPEROUT TESTFF

```

```

.SUBCKT TESTFF 1 Din clk1 OUT

```

```

** Inv Din

```

```

xMx1  Dinb  Din   1  0  hvtpfet L=40N W=2U
xMy1  Dinb  Din   0  0  hvtnfet L=40N W=1U

```

```

** Inv clk

```

```

xMx4  clk  clk1  1  0  hvtpfet L=40N W=2U
xMy4  clk  clk1  0  0  hvtnfet L=40N W=1U

```

```

xMx5  clk  clk1  1  0  hvtpfet L=40N W=2U
xMy5  clk  clk1  0  0  hvtnfet L=40N W=1U

```

```

*CIRCUIT ELEMENT

```

```

* d g s e p b

```

```

*LATCH 1 with tranmission gate

```

```

xMa1  x1a  Din   1  0  hvtpfet L=40N W=1U
xMa2  A    clk   x1a  0  hvtpfet L=40N W=1U
xMa3  A    clk  y1a  0  hvtnfet L=40N W=0.5U
xMa4  y1a  Din   0  0  hvtnfet L=40N W=0.5U

```

```

xMa5  w1a  Dinb  1  0  hvtpfet L=40N W=1U
xMa6  B    clk   w1a  0  hvtpfet L=40N W=1U
xMa7  B    clk  z1a  0  hvtnfet L=40N W=0.5U
xMa8  z1a  Dinb  0  0  hvtnfet L=40N W=0.5U

```

```

xMa9  x2a  Din   1  0  hvtpfet L=40N W=1U
xMa10 E    clk   x2a  0  hvtpfet L=40N W=1U
xMa11 E    clk  y2a  0  hvtnfet L=40N W=0.5U
xMa12 y2a  Din   0  0  hvtnfet L=40N W=0.5U

```

```

xMa13 w2a  Dinb  1  0  hvtpfet L=40N W=1U
xMa14 F    clk   w2a  0  hvtpfet L=40N W=1U
xMa15 F    clk  z2a  0  hvtnfet L=40N W=0.5U
xMa16 z2a  Dinb  0  0  hvtnfet L=40N W=0.5U

```

```

XLATCH1 1 A B F E PROLATCH

```

```

*LATCH 2 with tri-state gate at input

```

```

xMb1  x1b  A   1   0  hvtpfet L=40N W=1U
xMb2  Qa   clkb x1b  0  hvtpfet L=40N W=1U
xMb3  Qa   clk  y1b  0  hvtnfet L=40N W=0.5U
xMb4  y1b  A   0   0  hvtnfet L=40N W=0.5U

xMb5  w1b   B   1   0  hvtpfet L=40N W=1U
xMb6  Qb   clkb w1b  0  hvtpfet L=40N W=1U
xMb7  Qb   clk  z1b  0  hvtnfet L=40N W=0.5U
xMb8  z1b   B   0   0  hvtnfet L=40N W=0.5U

xMb9  x2b   E   1   0  hvtpfet L=40N W=1U
xMb10 Q   clkb x2b  0  hvtpfet L=40N W=1U
xMb11 Q   clk  y2b  0  hvtnfet L=40N W=0.5U
xMb12 y2b   E   0   0  hvtnfet L=40N W=0.5U

xMb13 w2b   F   1   0  hvtpfet L=40N W=1U
xMb14 QN   clkb w2b  0  hvtpfet L=40N W=1U
xMb15 QN   clk  z2b  0  hvtnfet L=40N W=0.5U
xMb16 z2b   F   0   0  hvtnfet L=40N W=0.5U
*
```

```
XLATCH2 1 Qa Qb QN Q PROLATCH
```

```

XBUF1  1 Qa   OUT BUFFER
.ENDS
```

```
*****
```

```

.SUBCKT PROLATCH 1 A B E F
*Latch
```

```

xM1  aw  F   1   0  hvtpfet L=40N W=1U
xM2  B   A   aw  0  hvtpfet L=40N W=1U
xM3  B   F   cw  0  hvtnfet L=40N W=0.5U
xM4  cw  A   0   0  hvtnfet L=40N W=0.5U
```

```

xM5  ax  E   1   0  hvtpfet L=40N W=1U
xM6  A   B   ax  0  hvtpfet L=40N W=1U
xM7  A   E   cx  0  hvtnfet L=40N W=0.5U
xM8  cx  B   0   0  hvtnfet L=40N W=0.5U
```

```
*Cross coupled PFETS
```

```

xM11 E   F   1   0  hvtpfet L=40N W=2U
xM12 F   E   1   0  hvtpfet L=40N W=2U
```

```

xM13 E   F   X   0  hvtnfet L=40N W=0.5U
xM14 F   E   Y   0  hvtnfet L=40N W=0.5U
xM15 X   A   0   0  hvtnfet L=40N W=2U
xM16 Y   B   0   0  hvtnfet L=40N W=2U
```

```
.ENDS
```

```
*****
```

```

*****
*** DELAY
*****

.SUBCKT DELAY 1 2 3
xm1pd  4  2  1  0  hvtpfet L=40N W=2U
xm2pd  4  2  0  0  hvtnfet L=40N W=1U
xm3pd  3  4  1  0  hvtpfet L=40N W=2U
xm4pd  3  4  0  0  hvtnfet L=40N W=1U
.ENDS
*****

*** BUFFER
*****

.SUBCKT BUFFER 1 2 3
xm1pd  4  2  1  0  hvtpfet L=40N W=2U
xm2pd  4  2  0  0  hvtnfet L=40N W=1U
xm3pd  3  4  1  0  hvtpfet L=40N W=2U
xm4pd  3  4  0  0  hvtnfet L=40N W=1U
.ENDS
*****

.inc "../skewParams.inc"
.param sigma=0
.inc "../design.inc"

.options post measout
*.probe v(clk) v(out) v(din)
.tran 0.1n 20.5u 0n
.print tran i(va) v(out) v(a) v(b) v(e) v(f) v(q) v(qn) v(qa) v(qb)
*.print dc I(V(a))

.meas tran I_5 INTEGRAL i(V5) from=0.55U to=20.5U
.meas tran I_8 INTEGRAL i(V8) from=0.55U to=20.5U

.meas tran Isubt INTEGRAL i(Vdd1) from=0.5U to=20.5U
.meas tran EsubT param = 'Isubt*Vsubt/20u'

.meas tran Isupert INTEGRAL i(Vdd2) from=0.5U to=20.5U
.meas tran Esupert param = 'Isupert*Vsupt/20u'

.meas tran ILS INTEGRAL i(Vdd3) from=0.5U to=20.5U
.meas tran E_LS param = 'Isupert*Vsupt/20u'

.meas tran LS_f TRIG V(subout) VAL='VsubT/2' TD=1.3u FALL=1
+ TARG V(lout) VAL='Vsupt/2' TD=1.3u FALL=1

.meas tran LS_r TRIG V(subout) VAL='0.175/2' TD=3.3u RISE=1
+ TARG V(lout) VAL='1.0/2' TD=3.3u RISE=1

.end

```

\*Single Supply LEVEL SHIFTER CKT. IBM without booster 1.2v  
 \*\*\*\*\*

Vdd1 1 0 0.35  
 vdd2 2 0 1.2

Vin2 i 0 PWL ( 0U 0.350  
 + 1.50U 0.350  
 + 1.55U 0.0  
 + 3.0U 0.0  
 + 3.05U 0.350  
 + 5.0U 0.350 )

\*\*Global Delay for the input

Mc1 3 i 1 P1 L=200n W=20U  
 Mc2 3 i 0 N1 L=200n W=10U  
 Mc3 6 3 1 P1 L=200n W=20U  
 Mc4 6 3 0 N1 L=200n W=10U

\*\*\*\*\*

\*\*\*Level Shifter

\*CIRCUIT ELEMENT

\* d g s

M1 9 6 5 P1 L=200n W=2U  
 M2 9 6 0 N1 L=200n W=50U

\*\*\*\*\*

M3 5 o 2 P1 L=200n W=0.125U  
 M4 5 2 2 N1 L=200n W=0.125U

\*\*\*\*\*

M5 o 9 2 P1 L=200n W=2U  
 M6 o 9 0 N1 L=200n W=1U

\*Stimulus

.IC v(9)=0.0 v(o)=1.2 v(5)=0.6

.tran 1N 6U 0N 1N UIC

.print tran V(8) i(vdd2)

.option reltol=3e-4 numdgt=7 itl1=1e3 nopage  
 + gmin=1e-18 abstol=1e-15 vntol=1e-6 pivtol=1e-24  
 + method=gear maxord=2 acct list node

.include ufsoi\_nom.mod

.end

\*CONSTANT CURRENT LEVEL SHIFTER CKT. with voltage booster 0.35v to 1.2v

\*\*\*\*\*

vdd1 1 0 0.35

vdd2 2 0 1.2

Vin2 i 0 PWL ( 0U 0.350

+ 1.500U 0.350

+ 1.505U 0.0

+ 3.000U 0.0

+ 3.005U 0.350

+ 5.000U 0.350 )

\*\*Global Delay for the input

Mc1 x1 i 1 P1 L=200n W=200U

Mc2 x1 i 0 N1 L=200n W=100U

Mc3 7 x1 1 P1 L=200n W=40U

Mc4 7 x1 0 N1 L=200n W=2U

Mc5 y x1 1 P1 L=200n W=40U

Mc6 y x1 0 N1 L=200n W=20U

\*\*\*\*\*

Mc7 6 y 1 P1 L=200n W=30U

Mc8 6 y 0 N1 L=200n W=10U

\*\*\*\*\*Voltage doubler\*\*\*\*\*

Mb1 1 6 x N1 L=200n W=2U

Mb2 9 6 x P1 L=200n W=2U

Mb3 9 6 0 N1 L=200n W=4U

Mb4 1 7 xb N1 L=200n W=2U

Mb5 11 7 xb P1 L=200n W=2U

Mb6 11 7 0 N1 L=200n W=4U

Cd1b 6 xb 1.0p

Cd1 7 x 1.0p

\*\*\*\*\*

\*\*\*Level Shifter

\*CIRCUIT ELEMENT

\* d g s

M1 5 8 2 P1 L=200n W=0.125U

M2 8 8 2 P1 L=200n W=0.125U

M4 8 9 0 N1 L=200n W=0.125U

M5 5 11 0 N1 L=200n W=0.125U

\*\*\*INV

M8 o 5 2 P1 L=200n W=0.25U

M9 o 5 0 N1 L=200n W=0.125U

\*\*\*OUTPUT CAPACITANCE

Cout o 0 20f

\*Stimulus

.IC v(x)=0.35 v(xb)=0.35

.tran 1N 6U 0N 1N UIC

```
.print tran V(o) i(vdd1)

.option reltol=3e-4 numdgt=7 itl1=1e3 nopage
+ gmin=1e-18 abstol=1e-15 vntol=1e-6 pivtol=1e-24
+ method=gear maxord=2 acct list node

.include ufsoi_nom.mod
.end
```

\*LEVEL SHIFTER CKT. without booster//Current Mirror// ckt. 0.35v to 1.2v

\*\*\*\*\*

Vdd1 1 0 0.35

vdd2 2 0 1.2

Vin1 i 0 PWL ( 0N 0.0

+ 350.000N 0.0

+ 350.150N 0.350

+ 650.00N 0.350

+ 650.150N 0.0

+ 700.00N 0.0 )

Vin2 4 0 PWL ( 0N 0.350

+ 350.000N 0.35

+ 350.150N 0.0

+ 650.00N 0.0

+ 650.150N 0.350

+ 700.00N 0.350 )

\* d g s

\*Mi1 4 3 1 P1 L=200n W=2U

\*Mi2 4 3 0 N1 L=200n W=1U

\*\*\*\*\*Voltage doubler\*\*\*\*\*

Mb1 1 4 x 0 72 N1 L=200n W=2U

Mb2 6 4 x 0 72 P1 L=200n W=2U

Mb3 6 4 0 0 N1 L=200n W=6U

Mb4 1 i xb 0 62 N1 L=200n W=2U

Mb5 7 i xb 0 62 P1 L=200n W=2U

Mb6 7 i 0 0 N1 L=200n W=6U

Cd1b 4 xb 5.0p

Cd1 i x 5.0p

Cd2b 72 7 5.0p

Cd2 62 6 5.0p

\*\*\*\*\*

\*\*\*Level Shifter

\*CIRCUIT ELEMENT

\* d g s

M1 8 5 2 P1 L=200n W=0.25U

M2 5 8 2 P1 L=200n W=20U

M4 7 1 8 N1 L=200n W=0.125U

M5 5 7 0 N1 L=200n W=40U



```

***INV
M6  o  5  2    P1 L=200n W=20U
M7  o  5  0    N1 L=200n W=10U
*Stimulus
.IC v(x)=0.35 v(xb)=0.35
.tran 1N 1U 0N 1N UIC
.print tran V(o) i(vdd1)

.option reltol=3e-4 numdgt=7 itl1=1e3 nopage
+ gmin=1e-18 abstol=1e-15 vntol=1e-6 pivtol=1e-24
+ method=gear maxord=2 acct list node

.include ufsoi_nom.mod
.end

```

\*\*\*\*\*

\*Charge Pump for the interconnect

\*\*\*\*\*

\*VOLTAGE SOURCES

Vdd 1 0 0.175

Vddbuf 100 0 0.175

Vclk1 6 0 PULSE(0.175 0.00 6U 200P 200P 1U 2U)

Vclk2 7 0 PULSE(0.00 0.175 6U 200P 200P 1U 2U)

vbitline bl 0 PWL (0u 0 49u 0 49.002u 0.2 60.500u 0.2 60.5002u 0)

vbitbar blb 0 PWL (0u 0 49u 0 49.002u 0 60.500u 0.2 60.5002u 0.2)

Vwl wl 0 PWL (0u 0.0 50.5u 0.0 50.5002u 0.5 55.5u 0.5 55.5002u 0)

Vtest i1 0 PWL (0u 0 41.0u 0 41.002u 0.175 65.5u 0.175 65.5002u 0

+ 66.0u 0 66.002u 0.175 67.0u 0.175 67.002u 0

+ 68.0u 0 68.002u 0.175 69.0u 0.175 69.002u 0

+ 70.0u 0 70.002u 0.175 71.0u 0.175 71.002u 0)

\*CIRCUIT ELEMENT

\* d g s e p b

xmn1 1 9 8 0 uvtnfet L=40N W=2U

xmn2 1 8 9 0 uvtnfet L=40N W=2U

xmp1 9 8 s1 0 uvtpfet L=40N W=2U

xmp2 8 9 s1 0 uvtpfet L=40N W=2U

xmn3 s1 11 10 0 uvtnfet L=40N W=2U

xmn4 s1 10 11 0 uvtnfet L=40N W=2U

xmp3 11 10 n2 0 uvtpfet L=40N W=2U

xmp4 10 11 n2 0 uvtpfet L=40N W=2U

xmn5 n2 13 12 0 uvtnfet L=40N W=2U

xmn6 n2 12 13 0 uvtnfet L=40N W=2U

xmp5 13 12 n3 0 uvtpfet L=40N W=2U

xmp6 12 13 n3 0 uvtpfet L=40N W=2U

xmn7 n3 15 14 0 uvtnfet L=40N W=2U

xmn8 n3 14 15 0 uvtnfet L=40N W=2U

xmp7 15 14 n4 0 uvtpfet L=40N W=2U

xmp8 14 15 n4 0 uvtpfet L=40N W=2U

xmn9 n4 17 16 0 uvtnfet L=40N W=2U

xmn10 n4 16 17 0 uvtnfet L=40N W=2U

xmp9 17 16 99 0 uvtpfet L=40N W=2U

xmp10 16 17 99 0 uvtpfet L=40N W=2U

XSRAM1 99 wl bl blb x1 y1 SRAM

```

XSRAM2 99 wl bl blb x2 y2 SRAM
XSRAM3 99 wl bl blb x3 y3 SRAM
XSRAM4 99 wl bl blb x4 y4 SRAM

```

```

XSRAM5 99 wl bl blb x5 y5 SRAM
XSRAM6 99 wl bl blb x6 y6 SRAM
XSRAM7 99 wl bl blb x7 y7 SRAM
XSRAM8 99 wl bl blb x8 y8 SRAM

```

```

xmn11 i1 x1 i2 0 uvtnfet L=40N W=0.5U
xmn12 i2 x2 i3 0 uvtnfet L=40N W=0.5U
xmn13 i3 x3 i4 0 uvtnfet L=40N W=0.5U
xmn14 i4 x4 i5 0 uvtnfet L=40N W=0.5U
xmn15 i5 x5 i6 0 uvtnfet L=40N W=0.5U
xmn16 i6 x6 i7 0 uvtnfet L=40N W=0.5U
xmn17 i7 x7 i8 0 uvtnfet L=40N W=0.5U
xmn18 i8 x8 i9 0 uvtnfet L=40N W=0.5U

```

```

xmp201 i10 i9 100 0 uvtpfet L=40N W=0.5U
xmn201 i10 i9 0 0 uvtnfet L=40N W=0.5U

```

```

xmp202 OUT i10 100 0 uvtpfet L=40N W=0.5U
xmn202 OUT i10 0 0 uvtnfet L=40N W=0.5U

```

\*\*\*\*Charge pump capacitance

```

C1 8 6 0.5p
C2 9 7 0.5p
C3 10 6 0.5p
C4 11 7 0.5p
C5 12 6 0.5p
C6 13 7 0.5p
C7 14 6 0.5p
C8 15 7 0.5p
C9 16 6 0.5p
C10 17 7 0.5p
C11 99 0 1.0p

```

```

.SUBCKT SRAM 99 wl bl blb x y
xmp100 y x 99 0 uvtpfet L=40N W=0.2U
xmn100 y x 0 0 uvtnfet L=40N W=0.2U
xmp101 x y 99 0 uvtpfet L=40N W=0.2U
xmn101 x y 0 0 uvtnfet L=40N W=0.2U

```

```

xmn102 bl wl x 0 uvtnfet L=40N W=1U
xmn103 blb wl y 0 uvtnfet L=40N W=1U
.ends

```

\*Stimulus

```

.IC v(8)= 0.175 v(9)= 0.175 v(10)= 0.175 v(11)= 0.175 v(12)= 0.175 v(13)= 0.175 v(14)= 0.175 v(15)=
0.175 v(99)= 0.175

```

```
.param sigma=0
.inc "../design.inc"
.options post measout

.tran 1n 150U 0N UIC
.print tran V(99) i(vdd)
.meas tran Itot INTEGRAL i(Vdd) from=0u to=100u
.meas tran Energy param = 'Itot*0.175'
.meas tran Power param = 'Itot*0.175/120u'

.end
```

\*\*\*\*\*DICE- cell latch based Flip-Flop design\*\*\*\*\*

\*\*-----NOTE-----

\*For all design simulate the following nodes

\* Clock-----is ----- v(2)

\* Data i/p---is----- v(3)

\* Capacitance o/p is-- v(123)

\*Voltage Sources

Vdd 1 0 DC 0.35

vdclk 100 0 DC 0.35

\*Clock input

Vin 80 0 PULSE(0 0.35 1U 200P 200P 0.9996U 2U)

\*Data input

Vda 3 0 PWL( 0U 0

+ 4.50U 0

+ 4.650U 0.35

+ 8.50U 0.35

+ 8.650U 0 )

\*\*\*\*\*Single Event Upset Modeled Strike\*\*\*\*\*

\*IR1 8 0 PULSE(0 2.40u 2.4u 50p 50p 1n 10u)

\*\*\*\*\*

\*Global clock delay

M100 81 80 100 0 p1 L=200N W=2U

M101 81 80 0 0 n1 L=200N W=1U

M102 2 81 100 0 p1 L=200N W=2U

M103 2 81 0 0 n1 L=200N W=1U

\*\*\*\*\*

\*Circuit Elements

\*Internal clock inversion and clock pulse

M104 40 2 1 0 p1 L=200N W=2U

M105 40 2 0 0 n1 L=200N W=1U

M106 50 40 1 0 p1 L=200N W=2U

M107 50 40 0 0 n1 L=200N W=1U

\*\*\*\*\*

\*\*\*\*\*

\* d g s e p

\*\*\*Circuit design

\*\*\* First latch

M1 5 3 1 0 p1 L=200N W=4u

M2 6 50 5 0 p1 L=200N W=4u

M3 6 40 7 0 n1 L=200N W=1u

M4 7 3 0 0 n1 L=200N W=1u

M5 8 50 5 0 p1 L=200N W=2u  
M6 8 40 7 0 n1 L=200N W=1u

M7 9 8 1 0 p1 L=200N W=1u  
M8 9 6 0 0 n1 L=200N W=0.5u

M9 10 6 1 0 p1 L=200N W=2u  
M10 10 8 0 0 n1 L=200N W=1u

M11 11 10 1 0 p1 L=200N W=2u  
M12 8 40 11 0 p1 L=200N W=2u  
M13 8 50 12 0 n1 L=200N W=1u  
M14 12 9 0 0 n1 L=200N W=1u

M15 13 9 1 0 p1 L=200N W=2u  
M16 6 40 13 0 p1 L=200N W=2u  
M17 6 50 14 0 n1 L=200N W=1u  
M18 14 10 0 0 n1 L=200N W=1u

\*\*\*\* Second Latch

M21 15 9 1 0 p1 L=200N W=4u  
M22 16 40 15 0 p1 L=200N W=4u  
M23 16 2 17 0 n1 L=200N W=1u  
M24 17 9 0 0 n1 L=200N W=1u

M25 18 40 15 0 p1 L=200N W=2u  
M26 18 2 17 0 n1 L=200N W=1u

M27 123 18 1 0 p1 L=200N W=1u  
M28 123 16 0 0 n1 L=200N W=0.5u

M29 20 16 1 0 p1 L=200N W=2u  
M30 20 18 0 0 n1 L=200N W=1u

M31 21 20 1 0 p1 L=200N W=2u  
M32 28 2 21 0 p1 L=200N W=2u  
M33 28 40 22 0 n1 L=200N W=1u  
M34 22 123 0 0 n1 L=200N W=1u

M35 23 123 1 0 p1 L=200N W=2u  
M36 16 2 23 0 p1 L=200N W=2u  
M37 16 40 24 0 n1 L=200N W=1u  
M38 24 20 0 0 n1 L=200N W=1u

\*\*\*Load Capacitance

CL 123 0 10F

\*INITIAL NODE CONDITIONS

.IC v(81)=0.35 v(9) = 0.00 v(123) = 0.00

```
*Stimulus
.tran 0.1N 11U 0U UIC
.print tran i(vdd)

*OPTION INCLUDE
.option reltol=3e-4 numdgt=7 itl1=1e3 nopage
+ gmin=1e-18 abstol=1e-15 vntol=1e-6 pivtol=1e-24
+ method=gear maxord=2 acct list node

*MODEL INCLUDE
.include ufsoi_nom.mod

.end
```

# \*\*\*\*\*D-Flip-Flop (Tristatable) Subthreshold Circuit

\*\*-----NOTE-----

\*For all design simulate the following nodes

\* Clock-----is ----- v(2)

\* Data i/p---is----- v(3)

\* Capacitance o/p is-- v(123)

\*Voltage Sources

Vdd 1 0 DC 0.35

vdclk 100 0 DC 0.35

\*Clock input

Vin 80 0 PULSE(0 0.35 1U 200P 200P 0.9996U 2U)

\*Data input

Vda 3 0 PWL( 0U 0

+ 4.50U 0

+ 4.650U 0.35

+ 8.50U 0.35

+ 8.650U 0 )

\*\*\*\*\*Single Event Upset Modeled Strike\*\*\*\*\*

\*IR1 8 0 PULSE(0 2.40u 2.4u 50p 50p 1n 10u)

\*\*\*\*\*

\*Global clock delay

M100 81 80 100 0 p1 L=200N W=2U

M101 81 80 0 0 n1 L=200N W=1U

M102 2 81 100 0 p1 L=200N W=2U

M103 2 81 0 0 n1 L=200N W=1U

\*\*\*\*\*

\* Rise and Fall delay of 150Ns

\*Circuit Elements

\* d g s e p

\*First Dynamic inverter

M1 4 3 1 0 p1 L=200N W=4U

M2 5 2 4 0 p1 L=200N W=2U

M3 5 3 0 0 n1 L=200N W=2U

\*Second Dynamic inverter

M4 60 2 1 0 p1 L=200N W=8U

M44 6 5 60 0 p1 L=200N W=8U

M5 6 5 7 0 n1 L=200N W=4U

M6 7 2 0 0 n1 L=200N W=4U



```

*Third Dynamic inverter
M7 8 6 1 0 p1 L=200N W=4U
M8 8 2 9 0 n1 L=200N W=4U
M9 9 6 0 0 n1 L=200N W=4U

M10 123 8 1 0 p1 L=200N W=4U
M11 123 8 0 0 n1 L=200N W=2U

CL 123 0 10F

*INTIAL NODE CONDITIONS
.IC v(81)=0.00 v(5)=0.35 v(123)=0.0

*stimulus
.tran 0.1N 11U 0U UIC
.print tran i(vdd)

*OPTION INCLUDE
.option reltol=3e-4 numdgt=7 itl1=1e3 nopage
+ gmin=1e-18 abstol=1e-15 vntol=1e-6 pivtol=1e-24
+ method=gear maxord=2 acct list node

*MODEL INCLUDE
.include ufsoi_nom.mod

.end

```

\*HL Flip Flop Subthreshold Circuit

\*\*-----NOTE-----

\*For all design simulate the following nodes

\* Clock-----is ----- v(2)

\* Data i/p---is----- v(3)

\* Capacitance o/p is-- v(123)

\*Voltage Sources

Vdd 1 0 DC 0.35

vdclk 100 0 DC 0.35

\*Clock input

Vin 80 0 PULSE(0.0 0.35 1U 200P 200P 1U 2U)

\*Data input

Vda 3 0 PWL( 0U 0

+ 4.500U 0

+ 4.650U 0.35

+ 8.500U 0.35

+ 8.650U 0)

\*\*\*\*\*Single Event Upset Modeled Strike\*\*\*\*\*

\*IR1 8 0 PULSE(0 2.40u 2.4u 50p 50p 1n 10u)

\*\*\*\*\*

\*Global clock delay

M100 81 80 100 0 p1 L=200N W=2U

M101 81 80 0 0 n1 L=200N W=1U

M102 2 81 100 0 p1 L=200N W=2U

M103 2 81 0 0 n1 L=200N W=1U

\*\*\*\*\*

\*Circuit Elements

\*Internal clock inversion and clock pulse

M104 40 2 1 0 p1 L=200N W=2U

M105 40 2 0 0 n1 L=200N W=1U

M106 50 40 1 0 p1 L=200N W=2U

M107 50 40 0 0 n1 L=200N W=1U

M108 60 50 1 0 p1 L=200N W=2U

M109 60 50 0 0 n1 L=200N W=1U

\*\*\*\*\*

\*\*\*\*\*Single Event Upset Modeled Strike\*\*\*\*\*

\*IR1 8 0 PULSE(0 2.40u 2.4u 50p 50p 1n 10u)

\*\*\*\*\*

\* d g s e p

\*

M1 4 2 1 0 p1 L=200N W=2U

M2 4 2 5 0 n1 L=200N W=4U

M3 5 3 6 0 n1 L=200N W=4U

```

M4  6 60  0 0  n1 L=200N W=8U

M5  4 3  1 0  p1 L=200N W=0.125U
M6  4 60  1 0  p1 L=200N W=0.125U

*
M12 123 4  1 0  p1 L=200N W=4U
M13 123 2 11 0  n1 L=200N W=8U
M14 11  4 10 0  n1 L=200N W=8U
M15 10 60  0 0  n1 L=200N W=10U

* Inverter
M17 13 123 1  0  p1 L=200N W=1U
M18 13 123 0  0  n1 L=200N W=0.5U

* Inverter
M19 123 13 1  0  p1 L=200N W=0.5U
M20 123 13 0  0  n1 L=200N W=0.25U

* Load
CL 123 0 10F

*INITIAL NODE CONDITIONS
.IC v(81)=0.35 v(4) = 0.35 v(123) = 0.00

*Stimulus
.tran 0.1N 11U 0U UIC
.print tran i(vdd)

*OPTION INCLUDE
.option reltol=3e-4 numdgt=7 itl1=1e3 nopage
+ gmin=1e-18 abstol=1e-15 vntol=1e-6 pivtol=1e-24
+ method=gear maxord=2 acct list node

*MODEL INCLUDE
.include ufsoi_nom.mod

.end

```

# \*\*\*\*\*D-Flip-Flop (Tristatable) Subthreshold Circuit

\*\*-----NOTE-----

\*For all design simulate the following nodes

\* Clock-----is ----- v(2)

\* Data i/p---is----- v(3)

\* Capacitance o/p is-- v(123)

\*Voltage Sources

Vdd 1 0 DC 0.35

vdclk 100 0 DC 0.35

\*Clock input

Vin 80 0 PULSE(0 0.35 1U 200P 200P 0.9996U 2U)

\*Data input

Vda 3 0 PWL( 0U 0

+ 4.350U 0

+ 4.50U 0.35

+ 8.350U 0.35

+ 8.50U 0 )

\*HOLD input

Vdb 4 0 PWL( 0U 0

+ 1.5U 0

+ 1.650U 0

+ 10.50U 0

+ 10.650U 0

+ 12.500U 0

+ 12.650U 0 )

\*\*\*\*\*Single Event Upset Modeled Strike\*\*\*\*\*

\*IR1 8 0 PULSE(0 800.00u 7.4u 50p 50p 1n 10u)

\*\*\*\*\*

\*Global clock delay

M100 81 80 100 0 p1 L=200N W=2U

M101 81 80 0 0 n1 L=200N W=1U

M102 2 81 100 0 p1 L=200N W=2U

M103 2 81 0 0 n1 L=200N W=1U

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*INPUT MUX DESIGN - Selects D or Feedback data

XINV0 1 4 8 INV

XAND0 1 8 3 5 AND

XAND1 1 4 123 6 AND

XOR0 1 5 6 33 OR

\*\*\*\*\*

\*\*\*\*\*REDUNDANT DYDFF\*\*\*\*\*

XDY0 1 2 33 50 DYDFF

XDY1 1 2 33 51 DYDFF

XDY2 1 2 33 52 DYDFF

\*\*\*\*\*MAJORITY CIRCUIT\*\*\*

XAND2 1 50 51 61 AND

XAND3 1 51 52 62 AND

XAND4 1 50 52 63 AND

XORI0 1 61 62 63 123 OR3I

\*\*\*\*\*

.SUBCKT AND 1 2 3 4

\*TWO INPUT AND

M20 6 2 1 0 p1 L=200N W=4U

M21 6 3 1 0 p1 L=200N W=4U

M22 6 2 5 0 n1 L=200N W=4U

M23 5 3 0 0 n1 L=200N W=4U

M24 4 6 1 0 p1 L=200N W=4U

M25 4 6 1 0 p1 L=200N W=4U

M26 4 6 7 0 n1 L=200N W=4U

M27 7 6 0 0 n1 L=200N W=4U

.ENDS

.SUBCKT OR 1 2 3 4

\*TWO INPUT OR

M30 8 2 1 0 p1 L=200N W=4U

M31 8 2 1 0 p1 L=200N W=4U

M32 8 2 10 0 n1 L=200N W=4U

M33 10 2 0 0 n1 L=200N W=4U

M34 9 3 1 0 p1 L=200N W=4U

M35 9 3 1 0 p1 L=200N W=4U

M36 9 3 11 0 n1 L=200N W=4U

M37 11 3 0 0 n1 L=200N W=4U

M38 4 8 1 0 p1 L=200N W=4U

M39 4 9 1 0 p1 L=200N W=4U

M40 4 8 5 0 n1 L=200N W=4U

M41 5 9 0 0 n1 L=200N W=4U

.ENDS

.SUBCKT OR3I 1 2 3 4 5

\*THREE INPUT OR

M50 8 2 1 0 p1 L=200N W=4U  
M51 8 2 1 0 p1 L=200N W=4U  
M52 8 2 12 0 n1 L=200N W=4U  
M53 12 2 0 0 n1 L=200N W=4U

M54 9 3 1 0 p1 L=200N W=4U  
M55 9 3 1 0 p1 L=200N W=4U  
M56 9 3 11 0 n1 L=200N W=4U  
M57 11 3 0 0 n1 L=200N W=4U

M58 10 4 1 0 p1 L=200N W=4U  
M59 10 4 1 0 p1 L=200N W=4U  
M60 10 4 11 0 n1 L=200N W=4U  
M61 11 4 0 0 n1 L=200N W=4U

M62 5 8 1 0 p1 L=200N W=4U  
M63 5 9 1 0 p1 L=200N W=4U  
M64 5 10 1 0 p1 L=200N W=4U  
M65 5 8 6 0 n1 L=200N W=6U  
M66 6 9 7 0 n1 L=200N W=6U  
M67 7 10 0 0 n1 L=200N W=6U

.ENDS

.SUBCKT INV 1 2 3

M70 3 2 1 0 p1 L=200N W=2U  
M71 3 2 0 0 n1 L=200N W=1U

.ENDS

.SUBCKT DYDFF 1 2 3 4

\*First Dynamic inverter

M1 40 3 1 0 p1 L=200N W=4U  
M2 5 2 40 0 p1 L=200N W=2U  
M3 5 3 0 0 n1 L=200N W=2U

\*Second Dynamic inverter

M4 60 2 1 0 p1 L=200N W=8U  
M5 6 5 60 0 p1 L=200N W=8U  
M6 6 5 7 0 n1 L=200N W=4U  
M7 7 2 0 0 n1 L=200N W=4U

```

*Third Dynamic inverter
M8 8 6 1 0 p1 L=200N W=4U
M9 8 2 9 0 n1 L=200N W=4U
M10 9 6 0 0 n1 L=200N W=4U

M11 4 8 1 0 p1 L=200N W=4U
M12 4 8 0 0 n1 L=200N W=2U

.ENDS

CL 123 0 10F

*INTIAL NODE CONDITIONS
.IC v(81)=0.00 v(5)=0.0 v(123)=0.0

*stimulus
.tran 0.1N 15U 0U UIC
.print tran i(vdd)

*OPTION INCLUDE
.option reltol=3e-4 numdgt=7 itl1=1e3 nopage
+ gmin=1e-18 abstol=1e-15 vntol=1e-6 pivtol=1e-24
+ method=gear maxord=2 acct list node

*MODEL INCLUDE
.include ufsoi_nom.mod

.end

```

\*C2MOS Flip-Flop Input Subthreshold Circuit

\*\*-----NOTE-----

\*For all design simulate the following nodes

\* Clock-----is -----v(2)

\* Data i/p---is-----v(3)

\* Capacitance o/p is v(123)

\*Voltage Sources

Vdd 1 0 DC 0.35

vdclk 100 0 DC 0.35

\*Clock input

Vin 80 0 PULSE(0 0.35 1U 200P 200P 0.9996U 2U)

\*Data input

Vda 3 0 PWL( 0U 0

+ 4.50U 0

+ 4.650U 0.35

+ 6.500U 0.35

+ 6.650U 0 )

\*\*\*\*\*Single Event Upset Modeled Strike\*\*\*\*\*

\*IR1 8 0 PULSE(0 2.40u 2.4u 50p 50p 1n 10u)

\*\*\*\*\*

\*Global clock delay

M100 81 80 100 0 p1 L=200N W=2U

M101 81 80 0 0 n1 L=200N W=1U

M102 2 81 100 0 p1 L=200N W=2U

M103 2 81 0 0 n1 L=200N W=1U

\*\*\*\*\*

\*Circuit Elements

\*Internal clock inversion and clock pulse

M104 40 2 1 0 p1 L=200N W=2U

M105 40 2 0 0 n1 L=200N W=1U

M106 50 40 1 0 p1 L=200N W=2U

M107 50 40 0 0 n1 L=200N W=1U

\*\*\*\*\*

\*Circuit Elements

\* Phase 1

\* d g s e p

\* Tri-State Inverter 1

M1 4 3 1 0 p1 L=200N W=4U

M2 5 50 4 0 p1 L=200N W=4U

M3 5 40 6 0 n1 L=200N W=2U



M4 6 3 0 0 n1 L=200N W=2U

\* Tri-State Inverter 2

M5 9 7 1 0 p1 L=200N W=1U

M6 5 40 9 0 p1 L=200N W=1U

M7 5 50 10 0 n1 L=200N W=1U

M8 10 7 0 0 n1 L=200N W=1U

\* Inverter 1

M9 7 5 1 0 p1 L=200N W=1U

M10 7 5 0 0 n1 L=200N W=1U

\* Phase 2

\* d g s e p

\* Tri-State Inverter 3

M11 11 5 1 0 p1 L=200N W=4U

M12 123 40 11 0 p1 L=200N W=4U

M13 123 50 31 0 n1 L=200N W=2U

M14 31 5 0 0 n1 L=200N W=2U

\* Tri-State Inverter 4

M15 15 13 1 0 p1 L=200N W=1U

M16 123 50 15 0 p1 L=200N W=1U

M17 123 40 14 0 n1 L=200N W=1U

M18 14 13 0 0 n1 L=200N W=1U

\* Inverter 2

M19 13 123 1 0 p1 L=200N W=1U

M20 13 123 0 0 n1 L=200N W=1U

CL 123 0 10F

\*INITIAL NODE CONDITIONS

.IC v(5)=0 v(81)=0

\*Stimulus

.tran 0.1N 11U 0U 1N UIC

.print tran i(vdd)

\*OPTION INCLUDE

.option reltol=3e-4 numdgt=7 itl1=1e3 nopage

+ gmin=1e-18 abstol=1e-15 vntol=1e-6 pivtol=1e-24

+ method=gear maxord=2 acct list node

\*MODEL INCLUDE

.include ufsoi\_nom.mod

.end

## **Curriculum Vitae**

Ameet Chavan received his Bachelor of Science degree in Electrical and Computer Engineering from Pune University in India in 1998. He received a Master of Science degree in Electrical Engineering from University of Texas at El Paso in 2003 and began his doctoral studies that same year.

While completing his studies Dr. Chavan worked as an assistant instructor for the Department of Electrical and Computer Engineering. He has interned with Intel Corporation Inc., Advanced Micro Devices (AMD) Inc., and with ECHOSTAR Technologies Ltd. While in India, he worked as a test engineer with Bull Power Systems.

Dr. Chavan dissertation entitled, "A Subthreshold Radhard Reconfigurable Architecture for Harsh Environments," was supervised by Dr. Eric MacDonald. Dr. Chavan has worked on a chip design project sponsored by the Defense Advanced Research Projects Agency (DARPA) at MIT Lincoln Labs, which incorporates his dissertation work.

Dr. Chavan has presented and published his research at various national and international conference meetings of the Institute of Electrical and Electronics Engineers (IEEE) and a NASA Symposium on VLSI Design. To further support his research, Dr. Chavan recieved research awards from NASA and The University of Texas at El Paso Graduate School. Additionally he was awarded a Texas Instruments Foundation Scholarship, a Texas Public Education Scholarship, and a National Science Foundation Distributed Computing Laboratory Scholarship.