5-2018

Asynchronous Circuit Stacking for Simplified Power Management

Andrew Lloyd Suchanek
University of Arkansas, Fayetteville

Asynchronous Circuit Stacking for Simplified Power Management

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in Engineering with a concentration in Computer Engineering

by

Andrew Suchanek
University of Arkansas
Bachelor of Science in Mathematics, 2013
University of Arkansas
Bachelor of Science in Computer Engineering, 2013
University of Arkansas
Master of Science in Computer Engineering, 2017

May 2018
University of Arkansas

This dissertation is approved for recommendation to the Graduate Council.

Jia Di, PhD
Dissertation Director

J. Patrick Parkerson, PhD
Committee Member

Dale Thompson, PhD
Committee Member

Jingxian Wu, PhD
Committee Member

ABSTRACT

As digital integrated circuits (ICs) continue to increase in complexity, new challenges arise for designers. Complex ICs are often designed by incorporating multiple power domains therefore requiring multiple voltage converters to produce the corresponding supply voltages. These converters not only take substantial on-chip layout area and/or off-chip space, but also aggregate the power loss during the voltage conversions that must occur fast enough to maintain the necessary power supplies. This dissertation work presents an asynchronous Multi-Threshold NULL Convention Logic (MTNCL) “stacked” circuit architecture that alleviates this problem by reducing the number of voltage converters needed to supply the voltage the ICs operate at. By stacking multiple MTNCL circuits between power and ground, supplying a multiple of V_{DD} to the entire stack and incorporating simple control mechanisms, the dynamic range fluctuation problem can be mitigated. A 130nm Bulk CMOS process and a 32nm Silicon-on-Insulator (SOI) CMOS process are used to evaluate the theoretical effect of stacking different circuitry while running different workloads. Post parasitic physical implementations are then carried out in the 32nm SOI process for demonstrating the feasibility and analyzing the advantages of the proposed MTNCL stacking architecture.

ACKNOWLEDGEMENTS

I would like to begin by expressing my deepest thanks and gratitude to my advisor, Dr. Jia Di, for his guidance, mentoring and support throughout my Ph.D. studies. His help was invaluable to me over the course of my time in his lab at the University of Arkansas. The standard to which he held me accountable to not only molded me into the researcher and man I am today but will undoubtedly help me continue to succeed in my professional career as well.

I am also very grateful to my committee members: Dr. J. Patrick Parkerson, Dr. Dale Thompson and Dr. Jingxian Wu for their support and assistance throughout my studies, both undergraduate and graduate. I would also like to thank Dr. Zhong Chen for his help with the physical implementation of my voltage stacking model. His semiconductor knowledge was invaluable to my research.

Over the course of my five years working at the Cato Springs Research Center (CSRC), I have worked among some very fine individuals and collaborated with them on many projects. I consider them not just colleagues, but friends and look forward to the work each of us will accomplish in the years to come. I would like to thank Dr. Landon Caley, Dr. Liang Men, Dr. Chien-Wei Lo, Dr. Nathan Kuhns, Dr. Thao Le, Mr. Brett Sparkman, Mr. John Brady, Mr. Jean Habimana, Mr. Brent Bell, and Mr. William Bouillon for their support and assistance, as well as their friendship during my graduate studies.

Last but certainly not least, the love and support of my family has been the pillar on which I have structured my life, both in and out the classroom. I offer my deepest gratitude to my father, Mark Suchanek, my mother, Deanna Suchanek, and my brother, Aaron Suchanek, for always believing in me and loving me unconditionally. I would also like to thank my in-laws, Susan and Jamie Shafer, for their love, support and help during the last several years.

DEDICATION

I would like to dedicate this work to my loving and amazing wife Brandy Suchanek and to our precious son Harrison Lloyd. Brandy, without you to guide me when I was lost or push me when I wanted to give up, I would not be writing these words today. Your strength and beauty in everything you do inspires me to be a better man and I continue to see how truly amazing you are every day I see our son grow under your parenting. Harrison, find something that you love to do and give everything you have to it and know that I will always be your biggest supporter and advocate.

TABLE OF CONTENTS

1 Introduction **1**

1.1 Power Management **1**

1.2 Previous Research **3**

1.2.1 DC-DC Converter Examples 3

1.2.2 Stacking Synchronous Circuits 4

1.3 Proposed Research and Approach **5**

1.4 Dissertation Organization **6**

2 Background **7**

2.1 Asynchronous Circuits **7**

2.2 Null Convention Logic **7**

2.3 NCL Pipelined Architecture **10**

2.4 Multi-Threshold NULL Convention Logic **11**

2.5 MTNCL Pipelined Architecture **13**

3 MTNCL Voltage Stacking **16**

4 Advanced MTNCL Voltage Stacking **24**

5 Physical Implementation and Results **40**

6 Conclusion **55**

7 Reference **57**

LIST OF TABLES

Table 1: Comparison table between LDO regulator and SI buck converter at V_{in} = 1.5V and V_{out} = 1.0 V 4

Table 2: Dual-Rail Encoding of NCL 8

Table 3: BOOLEAN Equivalents of 27 Fundamental NCL Gates 8

Table 4: Energy comparisons for MTNCL Dadda Multiplier stacked and unstacked in the 130nm Process 19

Table 5: Energy Delay Product Results from Stacked and Unstacked Multipliers in 130nm and 32nm Processes 26

Table 6: Multiplier 1 on Row 1 is Active while Multiplier 2 on Row 2 is Sleeping in the 130nm Process 27

Table 7: Multiplier 1 on Row 1 is Sleeping while Multiplier 2 on Row 2 is Active in the 130nm Process 28

Table 8: Multiplier 1 on Row 1 is Active while Multiplier 2 on Row 2 is Sleeping in the 32nm Process 29

Table 9: Multiplier 1 on Row 1 is Sleeping while Multiplier 2 on Row 2 is Active in the 32nm Process 30

Table 10: Data from Circuits Active on Row 1 while Circuit on Row 2 Sleeps in the 130nm Process 31

Table 11: Data from Circuits Active on Row 2 while Circuit on Row 1 Sleeps in the 130nm Process 32

Table 12: Data from Circuits Active on Row 1 while Circuit on Row 2 Sleeps in the 32nm Process 35

Table 13: Data from Circuits Active on Row 2 while Circuits on Row 1 Sleeps in the 32nm Process 36

Table 14: Energy Delay Product Results from Stacked and Unstacked RCA and Multiplier in the 32nm Process 48

Table 15: Data from Circuits Active on Row 1 While Circuit on Row 2 Sleeps in the 32nm Process 49

Table 16: Data from Circuits Active on Row 2 While Circuits on Row1 Sleeps in the 32nm Process 51

LIST OF FIGURES

Figure 1: Modern Power Regulation for Mixed-Signal SoCs in Portable Devices 2

Figure 2: NCL THmn Threshold Gate 9

Figure 3: NCL Static Gate Implementation 10

Figure 4: NCL Pipelined Architecture 10

Figure 5: MTNCL Static Gate Implementation 12

Figure 6: MTNCL THabm Threshold Gate 13

Figure 7: MTNCL Dual-Rail Register 13

Figure 8: MTNCL Pipelined Architecture 14

Figure 9: MTNCL Completion Detection Block 15

Figure 10: Simple MTNCL Double Stacked Implementation 16

Figure 11: MTNCL Dadda Multiplier on both rows running different workloads in the 130nm Bulk CMOS process 18

Figure 12: MTNCL Dadda Multiplier on both rows running different workloads in the 32nm SOI process 18

Figure 13: Simple MTNCL Triple Stacked Implementation 19

Figure 14: Same multiplier stacked three times running different workloads 20

Figure 15: MTNCL stacked circuits with multiplier on row 1 and RCA on row 2 in the 130nm process 21

Figure 16: Top multiplier sleeping, bottom RCA running (left) and bottom RCA sleeping, top multiplier running (right) in the 130nm process 22

Figure 17: MTNCL stacked circuits with multiplier on row 1 and RCA on row 2 in the 32nm process 23

Figure 18: Top multiplier sleeping, bottom RCA running (left) and bottom RCA sleeping, top multiplier running (right) in the 32nm process 23

Figure 19: MTNCL Double Stacked Circuits with Bypass and Awake Transistors 25

Figure 20: Energy Delay Product Curve as a Result of the Transistor Widths from Row 2 in the 130nm Process 27

Figure 21: Energy Delay Product Curve as a Result of the Transistor Widths from Row 1 in the 130nm Process 28

Figure 22: Energy Delay Product Curve as a Result of the Transistor Widths from Row 2 in the 32nm Process 29

Figure 23: Energy Delay Product Curve as a Result of the Transistor Widths from Row 1 in the 32nm Process 30

Figure 24: EDP Curve for RCA Operating on Row 1 in the 130nm Process 32

Figure 25: EDP Curve for Multiplier Operating on Row 1 in 130nm Process 32

Figure 26: EDP Curve for Multiplier Operating on Row 2 in the 130nm Process 33

Figure 27: EDP Curve for RCA Operating on Row 2 in the 130nm Process 33

Figure 28: Simulation Waveform with Increased Transistor Widths for Multiplier Active and RCA Sleeping in the 130nm Process 34

Figure 29: Simulation Waveform with Increased Transistor Widths for Multiplier Sleeping and RCA Active in the 130nm Process 34

Figure 30: EDP Curve for RCA Operating on Row 1 in the 32nm Process 35

Figure 31: EDP Curve for Multiplier Operating on Row 1 in 32nm Process 36

Figure 32: EDP Curve for Multiplier Operating on Row 2 in the 32nm Process 37

Figure 33: EDP Curve for RCA Operating on Row 2 in the 32nm Process 37

Figure 34: Simulation Waveform with Increased Transistor Widths for Multiplier Active and RCA Sleeping in the 32nm Process 38

Figure 35: Simulation Waveform with Increased Transistor Widths for Multiplier Sleeping and RCA Active in the 32nm Process 38

Figure 36: MTNCL Triple Stacked Circuits with Bypass and Awake Transistors 39

Figure 37: Cross-Section of Two Inverters Stacked in the 32nm SOI Process 40

Figure 38: MTNCL Ripple Carry Adder Placed and Routed in Cadence Innovus Tool 41

Figure 39: MTNCL Dadda Multiplier Placed and Routed in Cadence Innovus Tool 42

Figure 40: LVS-Clean RCA Design in Cadence Virtuoso 43

Figure 41: LVS-Clean Multiplier Design in Cadence Virtuoso 44

Figure 42: LVS-Clean RCA Row 1 and Multiplier Row 2 Design with Capacitors and Additional Logic in Cadence Virtuoso 45

Figure 43: LVS-Clean Multiplier Row 1 and RCA Row 2 Design with Capacitors and Additional Logic in Cadence Virtuoso 46

Figure 44: Bypass and Awake Transistors in Reference to Small Capacitor (Left) and Bypass and Awake Transistors (Right) 47

Figure 45: EDP Curve for Post-PEX RCA Operating on Row 1 in the 32nm Process 49

Figure 46: EDP Curve for Post-PEX Multiplier Operating on Row 1 in the 32nm Process 50

Figure 47: Post-PEX RCA Operating on Row 1 with Bypass and Awake Transistors' Widths Sized on Row 2 at 104 nm (left) and 936 nm (right) 50

Figure 48: EDP Curve for Post-PEX Multiplier Operating on Row 2 in the 32nm Process 52

Figure 49: EDP Curve for Post-PEX RCA Operating on Row 2 in the 32nm Process 52

Figure 50: Post-PEX Multiplier Operating on Row 2 with Bypass and Awake Transistors' Widths Sized on Row 1 at 312 nm 53

Figure 51: Post-PEX Multiplier Operating on Row 2 with Bypass and Awake Transistors' Widths Sized on Row 1 at 1,872 nm 53

1 Introduction

New challenges arise as semiconductor technology continues to advance, and newer digital integrated circuits (ICs) are being designed with smaller process nodes that run at lower voltages. Minimizing energy consumption while maintaining performance is one of the dominating factors that drives new design methodologies. One important trend to note is that as semiconductor processes continue to get smaller, more circuit components are able to fit on a single chip, but this trend results in multiple power supply rails of different voltages needed to power the different circuit components [1]. Addressing these multiple power domains is an additional concern to modulating the off-chip power supply down to the lower voltages so the circuit components can reliably operate at their desired voltage level [2]. This has traditionally been accomplished using voltage converters that were either implemented on-chip or off-chip. However, as the number of power domains continues to increase on a single chip, their tradeoffs make them less than ideal because the converters not only take up substantial on-chip layout area and off-chip space, but also aggregate the power loss during the voltage conversions.

1.1 Power Management

As design kits continue to reduce in size, the voltages needed to power core rails have continued to drop from 2.5 V to 0.9 V, while the off-chip supply has remained higher at 12 V, 5 V, and 3.3 V [1]. Therefore, for multiple cores on a single chip, dynamic voltage and frequency scaling (DVFS) with fast voltage transitions for each core and logic block is sought after because it can improve energy efficiency while reducing the overall power consumption. Implementing multiple on-chip power domains using off-chip DC-DC converters has accomplished this in the past; but more recently, there have been a lot of interests in on-chip converters that can also implement the multiple power rails [3]. There are several tradeoffs between these off-chip and on-chip conversion designs. The main issue with off-chip power supply is that there is a higher IR drop and I^2R power loss due to the board and package resistance when the current demand on-chip increases suddenly. This is also amplified by the fact that off-chip power delivery impedance realistically does not scale [4]. Linear regulators and inductor-based switching regulators are the two main types of systems implemented on-chip to supply the necessary voltages to the different components. Linear regulators are relatively low cost because they take up little area and are easier to design requiring only an internal switch along with a single input and output capacitor. The main drawback is that their efficiency is very low. On the other hand, inductor-based switching regulators have a much higher efficiency at 85% to 95% but require a lot more effort to implement because of their complexity, design time and support components that take up a lot more board area [5].

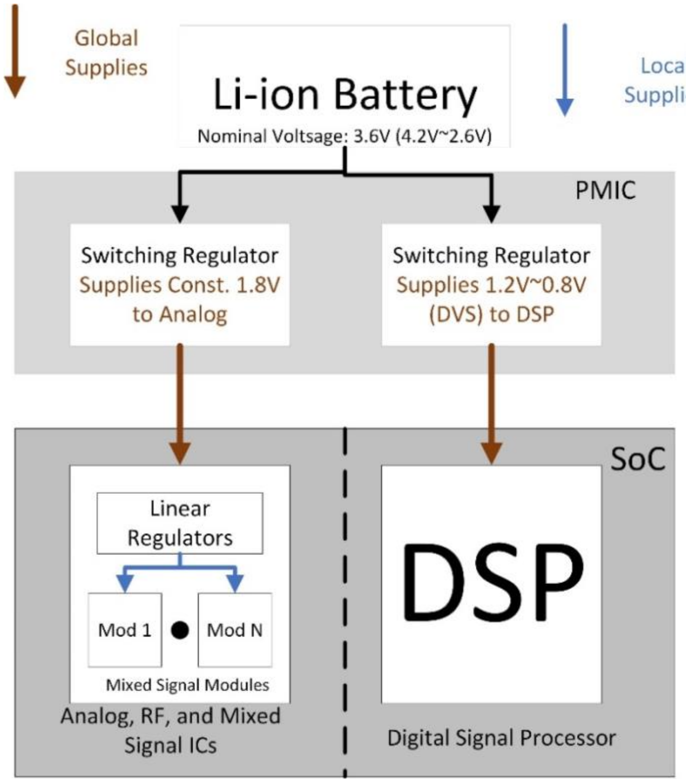

Figure 1: Modern Power Regulation for Mixed-Signal SoCs in Portable Devices [6]

Figure 1 shows that a typical mixed-signal System on Chip (SoC) is composed of two off-chip inductor-based switching regulators which operate directly off of the lithium-ion battery to generate the two global supplies. In this application, the global supplies are broken down into a digital supply used by the Digital Signal Processor (DSP) and an analog supply that is then further converted by linear regulators. The DSP uses dynamic voltage scaling (DVS) which introduces a lot of noise, so the analog supply is used as local supply. Depending on the number of mixed signal modules that are required for this portable device, the number of linear regulators needed to supply them power increases as well. Although the more cost and area effective, as the voltage drop between the main power supply and the local power required at each module increases, the collective power loss from the regulators becomes more significant [6]. Therefore, not only does power loss become a concern, but the added circuitry needed to supply the different power rails also takes up area and time for designing and debugging in order to ensure all components work together.

1.2 Previous Research

1.2.1 DC-DC Converter Examples

Prior research [2] takes two topologies of DC-DC converters and compares their performance when the input voltage is 1.5 V and the output voltage is 1.0 V. The first DC-DC converter used is an example of a linear regulator, called the low dropout (LDO) regulator. It is compared to an inductor-based switching regulator called the switched inductor (SI) buck regulator. The data in Table 1 shows clear tradeoffs between both types of DC-DC converters and such tradeoffs vary depending on the power supply input and the required output. The efficiency, speed of delivery, area used, and complexity can be major drawbacks when designing the power delivery system and should be considered from time of concept in order to plan ahead for what the various components on-chip will be required in terms of power supply.

Table 1: Comparison table between LDO regulator and SI buck converter at V_{in} = 1.5V and V_{out} = 1.0 V [2].

Parameter	LDO Regulator	SI buck converter
Efficiency at light current load=100µA	16%	13%
Efficiency at heavy current load=5mA	60%	78%
Settling Time	Less than 2µs	97µs
Area	Small	Large
Transient Response	Fast	Slow
Control Technique	Simple	Complex
Integration	On-chip	Both but mostly off-chip

1.2.2 Stacking Synchronous Circuits

An alternative method for alleviating the energy loss caused by multiple voltage converters was proposed in [4]. This research took the low-voltage blocks and incorporated them together in a "stacked" architecture. This methodology not only simplifies the off-chip and on-chip power delivery systems, but also reduces the chip current draw. It uses voltage stacking for delivering *n* times the normal operating voltage to *n* circuits that are stacked upon one another in series. By increasing the voltage level required for delivery and decreasing the chip current draw, their research could potentially achieve several benefits:

1. I^2R power loss is reduced by a factor of n^2 due to board and package resistance.
2. IR drop is reduced by a factor of n^2 because the IR drop reduction by a factor of *n* over *n-stacked* cores.
3. Voltage regulators and converters benefit from a lower step-down ratio that improves their efficiency and reduces their design complexity.

The research in [4] was carried out using synchronous logic in a 150nm Fully Depleted Silicon-on-Insulator (FDSOI) CMOS process. Although partially successful, it was only under certain strict constraints that the system would work properly. By using the clocked synchronous logic, the within-die voltage fluctuation between the different stacked layers caused a significant inner current mismatch that prevented the system from working if the cores were not the same circuitry OR if the workload between the stacked cores varied by any significant amount. In other words, this synchronous voltage stacking structure would only work if copies of the same circuit are stacked upon one another running nearly identical workloads simultaneously.

1.3 Proposed Research and Approach

In order to retain the energy efficiency benefit of voltage stacking while removing the strict constraints of stacked circuits, this dissertation research will implement the voltage stacking methodology using an asynchronous *quasi-delay-insensitive* (QDI) paradigm named Multi-Threshold NULL Convention Logic (MTNCL). Stacking MTNCL circuits not only provides the same benefits as discussed above, but also allows for a more comprehensive analysis, i.e., due to MTNCL's robustness, timing independence, and minimized effect from electromagnetic interference (EMI), not only can different workloads be run, but different stacked types and sizes of circuits can be implemented in the stacked architecture. In addition, MTNCL circuitry has the ability to be put to sleep, which in this stacked infrastructure allows other circuits to continue running while one or more sleeps. By adding simple control logic, the designer will have the option to choose whether to allow for the non-sleeping circuit to speed up while other circuits sleep or reduce energy consumption while maintaining performance.

In addition to the previous benefits of voltage stacking, the MTNCL voltage stacking methodology provides improvements in three main areas:

1. Reduced Power Loss in Converters
    a. Improves efficiency through lower step-down ratios

   b. Reduce the number of LDOs and/or switch-mode regulators

2. Improved Reliability
   a. Fewer circuit components are required
   b. Delay Insensitivity – circuits work reliably in varying conditions, temperatures, etc.
3. Size/Weight Reduction
   a. Removal of discrete passive elements such as inductors
   b. Smaller Printed Circuit Boards (PCBs) required for implementation

1.4 Dissertation Organization

Chapter 2 provides the background information on the asynchronous paradigm that is adapted by this work. Chapter 3 contains the basic voltage stacking design implementation that was developed for the MTNCL asynchronous circuitry and the schematic simulation results from it. Chapter 4 contains the advanced voltage stacking design implementation and methodology developed for the MTNCL asynchronous circuitry. In addition, chapter 4 provides schematic simulation results from various combinations of the advanced MTNCL voltage stacking model as well as optimization techniques based upon the circuits being tested. Chapter 5 provides the placed and routed design for the advanced model as well as simulation results from post parasitic extracted designs. Chapter 6 summarizes the findings and concepts discussed in this dissertation and examines future possibilities of this work.

2 Background

2.1 Asynchronous Circuits

Asynchronous circuits are sequential digital logic circuits that operate without the use of a clock and are therefore considered clockless circuitry. The two main asynchronous design styles are the *bounded-delay* model and the *delay-insensitive* (DI) model. The former model is designed according to the worst-case propagation delay, so strenuous timing analysis is required to ensure that the delays in both the gates and the wires are bounded by that worst-case behavior in order to avoid hazards or glitches during circuit operation. The latter model is considered to be correct-by-construction because the delays in both logic elements and wires are assumed to be unbounded. As a result, the DI model needs not to be subject to much, if any, timing analysis [7]. However, based upon how a circuit is designed, there can exist arbitrary gate and wire delays that could make the timing model too constrained for some practical circuits using the DI model [8]. Therefore, *quasi-delay-insensitive* (QDI) logic evolved from the DI model in the mid-1980s by separating the wires into critical and non-critical paths. This allowed the QDI model to demonstrate that the skew from the different branches of the critical path wires were less than that of the minimum gate delay and those not in the critical path had no effect on any timing constraints [9]. These assumptions allow the QDI methodology to become a commonly used practice in asynchronous circuit design.

2.2 NULL Convention Logic

NULL Convention Logic (NCL) is a QDI asynchronous circuit design methodology that incorporates multi-rail logic [10]. NCL is therefore correct-by-construction, and there is no timing analysis required when designing the circuitry. For this work, the dual-rail encoding of

NCL incorporates two rails that provide three valid states and one invalid state as shown in Table 2 below.

Table 2: Dual-Rail Encoding of NCL [7]

	NULL	DATA0	DATA1	INVALID
RAIL^0	**0**	**1**	**0**	**1**
RAIL^1	**0**	**0**	**1**	**1**

Table 3: BOOLEAN Equivalents of 27 Fundamental NCL Gates [7]

NCL Gate	BOOLEAN Equivalent
TH12	A+B
TH22	AB
TH13	A+B+C
TH23	AB + AC + BC
TH33	ABC
TH23w2	A + BC
TH33w2	AB + AC
TH14	A+B+C+D
TH24	AB + AC + AD + BC + BD + CD
TH34	ABC + ABD + ACD + BCD
TH44	ABCD
TH24w2	A + BC + BD + CD
TH34w2	AB + AC + AD + BCD
TH44w2	ABC + ABD + ACD
TH34w3	A + BCD
TH44w3	AB + AC + AD
TH24w22	A + B + CD
TH34w22	AB + AC + AD + BC + BD
TH44w22	AB + ACD + BCD
TH54w22	ABC + ABD
TH34w32	A + BC + BD
TH54w32	AB + ACD
TH44w322	AB + AC + AD + BC
TH54w322	AB + AC + BCD
THxor0	AB + CD
THand0	AB + BC + AD
TH24comp	AC + BC + AD + BD

The two rails are mutually exclusive of each other, meaning that both rails cannot be asserted at the same time, i.e., the invalid state. The other three states depict the NULL state, and the DATA0 and DATA1 states, equivalent to logic 0 and logic 1 respectively. Table 3 shows the list of 27 fundamental NCL gates that make up the majority of their BOOLEAN equivalents. These gates are used when designing an NCL asynchronous circuit. Their naming convention follows a THmn nomenclature, where to assert the output, *m* of the *n* inputs must be asserted. Their diagram, shown in Figure 2 below, depicts an NCL THmn gate.

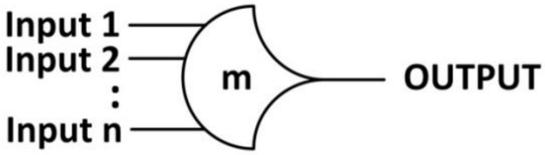

Figure 2: NCL THmn Threshold Gate [7]

The NCL architecture makes use of a method called hysteresis, where all of the inputs of a logic gate must be de-asserted before the output is able to return to a logic 0. This ensures that all of the gates in a combinational logic block propagate their data completely before the NULL wavefront occurs and the correct data is latched at the output. The static implementation of an NCL gate is seen in Figure 3, where the logic required to output a 1 is in the set logic block. From there, the logic to keep the output high with hysteresis is in the hold1 logic block. When all inputs have been returned to 0, the reset logic block turns on the NFET of the inverter connected to Z so that the output is pulled down to 0. The hold0 logic block keeps it there until all inputs have arrived for the next DATA wave.

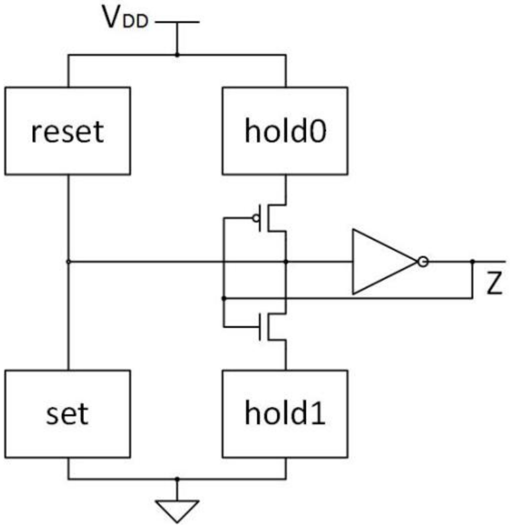

Figure 3: NCL Static Gate Implementation [7]

2.3 NCL Pipelined Architecture

At both ends of NCL combinational logic blocks are DI registers that are used for latching data in a single-stage or a pipelined design if more than one stage is required. These registers use handshaking signals, *Ko* and *Ki*, to communicate with the previous set of registers when it is ready for either a NULL or DATA wave. A NULL wave is where all rails are logic 0, therefore designating a NULL state. A DATA wave is when *rail*^{0} and *rail*^{1} have opposite values, designating either a DATA0 or DATA1 state for every instance of the dual-rails. Between every DATA wave there will be a NULL wave to ensure that the data was propagated through the pipeline correctly. Figure 4 below depicts the NCL pipelined architecture.

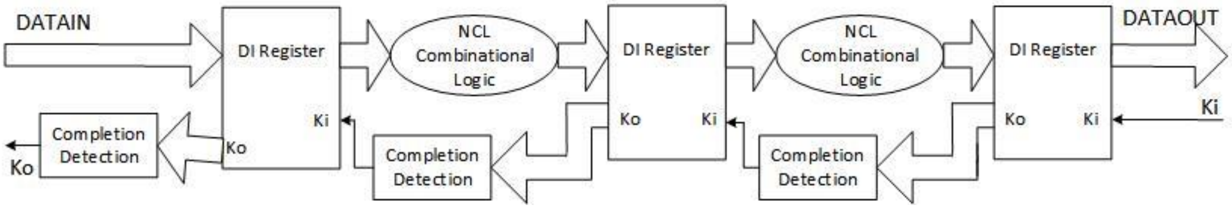

Figure 4: NCL Pipelined Architecture [7]

All registers in a stage must have the same *Ko* value before they can request the next DATA or NULL wave from the previous register stage. Therefore, if a DATA wave has just propagated through to a new stage of registers, those registers will each produce a *Ko* of 0, which will produce a *request-for-NULL (rfn)* signal that is sent to the previous register set. Conversely, when a NULL wave has propagated through a stage, all of the registers will have a *Ko* of 1, which is equivalent to producing a *request-for-DATA (rfd)* signal to the previous registers.

2.4 Multi-Threshold NULL Convention Logic

As process nodes continue to get smaller, the channel width and length for individual transistors also continues to shrink in size. Although this allows designs to run at lower voltages, one drawback is that when the transistor is "turned off", the amount of current that continues to flow is more noticeable in the smaller processes. This is referred to as leakage current and the resulting leakage power associated with it has become more of an issue to deal with when considering overall power dissipation. Multi-Threshold CMOS (MTCMOS) was created for synchronous designs to serve as an internal power-gating method. The goal was to reduce the power dissipation by incorporating two different transistors, one with a higher threshold voltage (V_t) than the other. The High-V_t transistors allow a much smaller amount of leakage current to flow when the transistor is "turned off" but do so at the cost of a much slower switching speed than their Low-V_t counterpart. Therefore, the Low-V_t transistors are mostly used when switching speed is crucial to the circuit's performance as long as every critical path from power to ground and power to the output consist of one High-V_t transistor to minimize leakage. Otherwise, when switching speed is not crucial, High-V_t transistors will be used to minimize the flow of leakage current and minimize any unwanted power loss.

Multi-Threshold NULL Convention Logic (MTNCL) was created by taking NCL and implementing MTCMOS power-gating. MTNCL uses both Low-V_t and High-Vt transistors and replaces the hysteresis function with a sleep function that is comprised of a sleep signal connected to both an NMOS transistor and an PMOS transistor in their respective pull-down and pull-up networks. Figure 5 below shows that the sleep signal controls the transistors that power-gate the circuit.

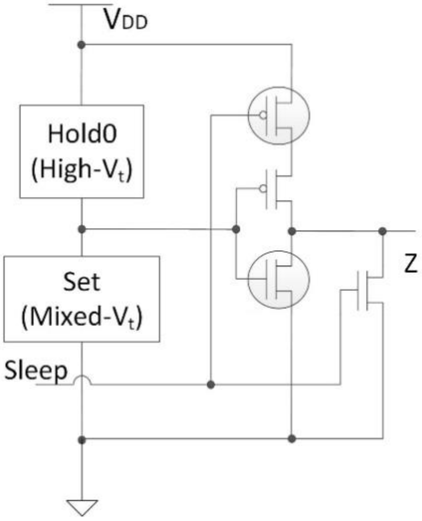

Figure 5: MTNCL Static Gate Implementation [11]

The static gate implementation shows that in addition to removal of hysteresis and inclusion of the sleep transistors, there are now the Low-V_t and High-V_t transistors being used within the circuit. The Hold0 logic block of transistors do not need to be fast switching because the sleep signal actually forces the circuit to 0 when it is asserted. At that point, the Hold0 logic holds it at 0 until the sleep signal is de-asserted. The Set logic block of transistors uses mostly Low-V_t transistors for faster switching speeds as long as the transistor whose source is directly tied to ground is a High-V_t transistor to minimize leakage.

The naming convention of MTNCL gates follows a THabm nomenclature, where *a* of *b* inputs must be asserted to have the output asserted. The output is then de-asserted when the sleep signal is enabled, which is provided by the completion logic from the stage one step ahead in the pipeline. The MTNCL gate symbol is shown in Figure 6 below.

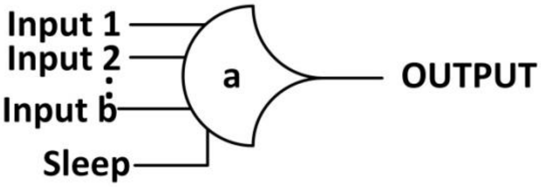

Figure 6: MTNCL THabm Threshold Gate [12]

2.5 MTNCL Pipelined Architecture

Just like in the NCL pipeline architecture, MTNCL has DI registers on both ends of any MTNCL combinational logic that latch either the DATA or NULL waves. Figure 7 below shows the MTNCL register composed of two TH12m gates. When the sleep signal is de-asserted, the inputs of *rail*^{0} and *rail*^{1} propagate to the outputs of *rail*^{0} and *rail*^{1}. Conversely, when the sleep signal is asserted, both output rails are 0, resulting in a NULL value.

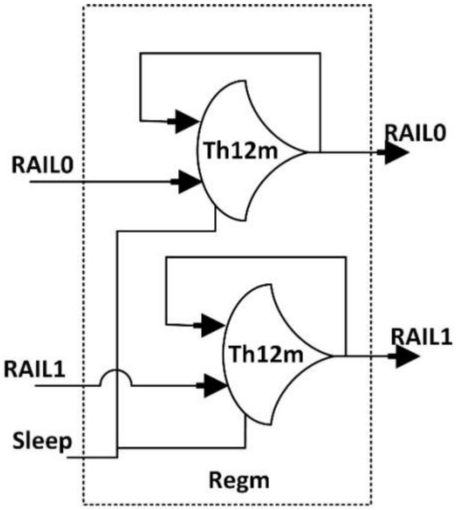

Figure 7: MTNCL Dual-Rail Register [12]

In addition to the registers, MTNCL pipeline architecture is much like that of NCL, where there is combinational logic and completion logic for each stage of the pipeline. Similarly, *Ko* and *Ki* signals are also the handshaking signals used in MTNCL pipelining to indicate when a stage is ready for the next NULL or DATA wave. Also, like NCL pipelining, each DATA wave is separated by a NULL wave in order to prevent data corruption from subsequent data sets. The main difference between the two pipelined architectures is NCL uses hysteresis to prevent the output from changing to 0 until the NULL wave is requested; whereas, MTNCL uses the sleep signal to generate the NULL wave in the pipeline. The sleep signal is created by the *Ko* from the previous stage of the pipeline, so it propagates forward to the next stage. Therefore, a NULL wave is generated when the previous stage's *Ko* is 1, thus asserting the sleep signal for the gates in the next stage of the pipeline to produce an output of 0. Conversely, when DATA is ready to propagate, the sleep signal is disabled because the previous *Ko* is now 0 and the gates are allowed to propagate the data they receive from the previous stage. Figure 8 shows the MTNCL pipelined architecture and how the handshaking signals work in conjunction with the sleep signals of each stage's components.

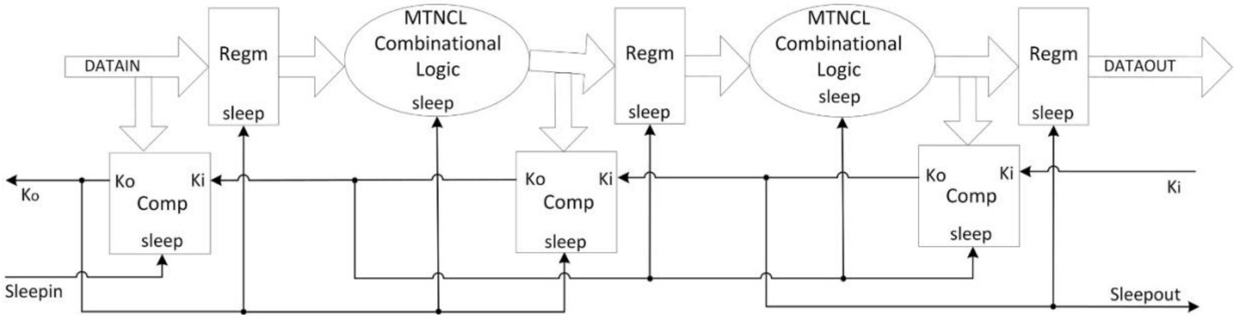

Figure 8: MTNCL Pipelined Architecture [12]

The completion logic shown in Figure 9 receives the outputs from the combinational logic along with the next stage's *Ko* which enters as this stage's *Ki*. The data from the combinational logic is also fed directly into the next stage's registers set. The combinational logic actually enables or disables the sleep signal for the subsequent stage in the pipeline based upon the value of the next stage's *Ko* signal.

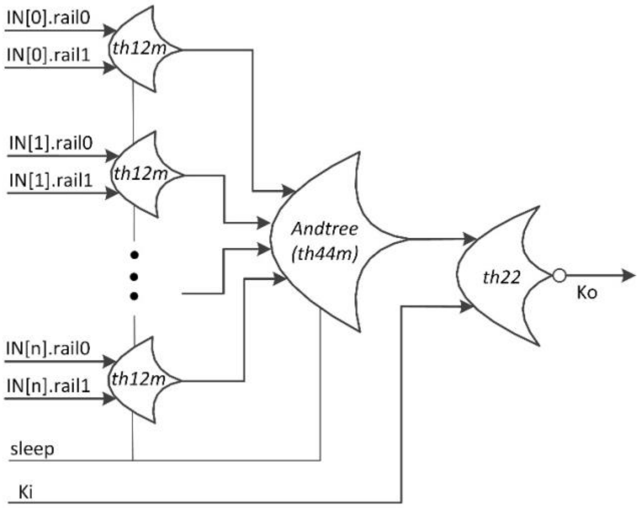

Figure 9: MTNCL Completion Detection Block [12]

When the completion logic for a stage receives a DATA wave and the subsequent stage sends a *rfd*, the completion logic disables the sleep signal for the next stage, which allows the current DATA wave to continue propagating through the register set to the next stage. Conversely, when the completion logic receives a NULL wave and the subsequent stage sends a *rfn*, the completion logic enables sleep which puts the next stage's registers, combinational logic and completion logic to sleep as well. This continues to produce a NULL wave that propagates through the pipeline one stage at a time ensuring that all previous data is erased, power gating all MTNCL gates as it goes.

3 MTNCL Voltage Stacking

This dissertation work is aimed at using the asynchronous MTNCL design paradigm to create a functional voltage stacking model. Its immediate effect comes in the reduction of the on-chip and off-chip DC-DC converters and regulators that are needed to supply the different power rails because it not only alleviates the power loss that occurs in these components, but removes the space and time needed to implement them as well. The first implementation is a simple stacked structure where two MTNCL circuits are placed in series with one another and in parallel with two capacitors as shown in Figure 10 below.

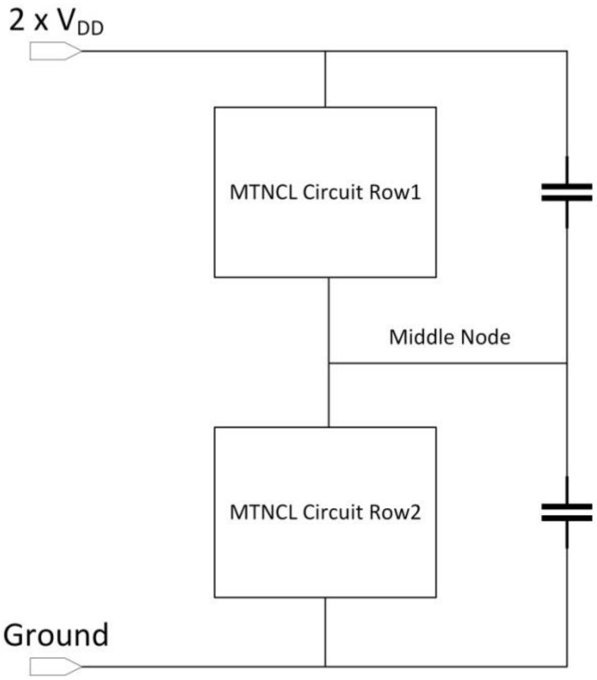

Figure 10: Simple MTNCL Double Stacked Implementation

The capacitors function as bypass capacitors and are used to ensure that the middle node voltage will remain oscillating near half of the supplied voltage, which is just the normal V_{DD} in a 2-stacked architecture. The design kits used to test all work are the IBM 130nm Bulk CMOS process and the GLOBALFOUNDRIES 32nm Silicon-on-Insulator (SOI) CMOS process. The designated voltage for each transistor is 1.2 V in the 130nm process and 0.9 V in the 32nm process, so the supply voltage used is twice that in the MTNCL Double Stacked implementation: 2.4 V and 1.8 V, respectively. The values of the capacitors are calculated by performing multiple tests on various sized circuits in both processes. It is determined that 50 fF for the top capacitor and 10 pF for the bottom capacitor are adequate for maintaining an acceptable voltage on the middle node when the same circuits are running on both rows despite the workloads. Although different sized circuits would cause the middle node voltage to fluctuate towards the larger circuit, adjusting the capacitance would not have significant effect in counterbalancing. Therefore, the capacitances remain constant throughout the design phase.

One of the first simulations instantiates two copies of the same MTNCL pipelined circuit, an 11-bit by 7-bit Dadda Multiplier, stacks them on top of one another and runs them using different workloads to see the effect they have on the middle node voltage and their individual performance. While both circuits are running their respective workloads, the middle node voltage oscillates around 1.2 V in the 130nm process and 0.9 V in the 32nm process as shown in Figures 11 and 12, respectively. Since the 130nm design kit is a Bulk CMOS process, the bases of the NFETs from the top circuit and the bases of the PFETs from the bottom circuit are tied to the voltage potential of the middle node. This causes the voltage of the middle node to fluctuate much more in the 130nm process, but the effect it has on either circuit is still negligible because the dynamic range is large enough that outputs can easily be designated as a logic 0 or logic 1. The only thing altered is the actual voltage value that is considered to be logic 0 (GND) for the upper circuit and logic 1 (V_{DD}) for the lower circuit. The middle voltages are highlighted in red for the 130nm process and blue for the 32nm process, while the remaining waveform is a single dual-rail output from the circuits running on each row.

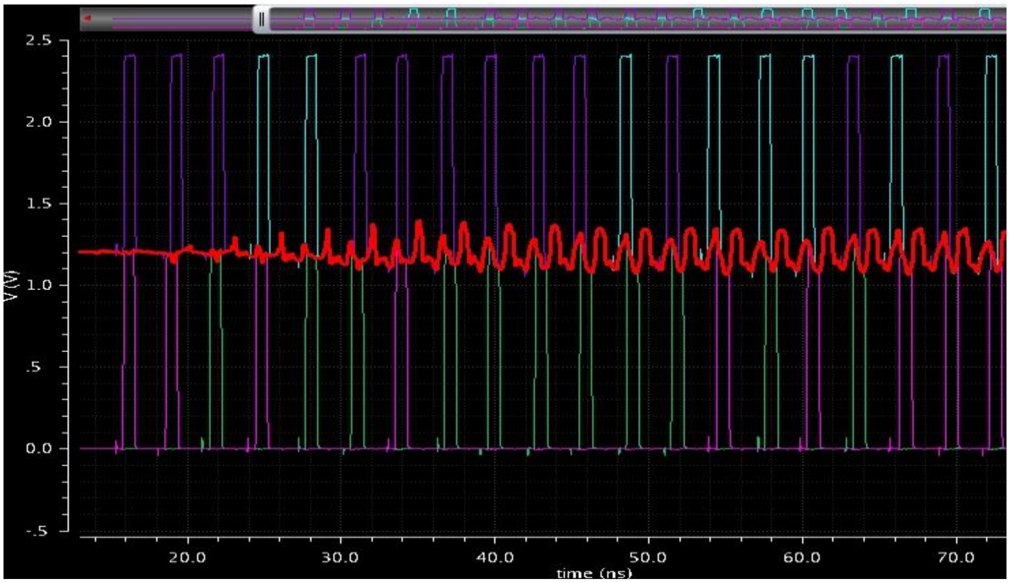

Figure 11: MTNCL Dadda Multiplier on both rows running different workloads in the 130nm Bulk CMOS process

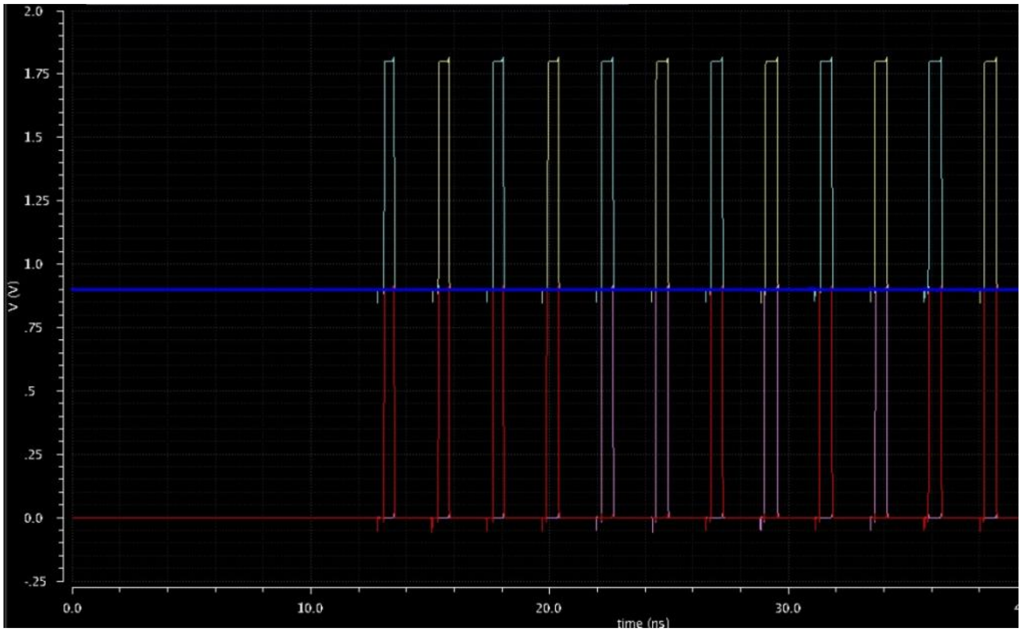

Figure 12: MTNCL Dadda Multiplier on both rows running different workloads in the 32nm SOI process

The energy comparison for the stacked MTNCL Dadda Multipliers in the 130nm process can be viewed in Table 4. They show that when running on their own, the total active energy consumption is 239 pJ, which is actually 4 pJ higher than the stacked equivalent. In addition, when the multipliers are stacked and only the top one is running, meaning the bottom multiplier is sleeping, it uses less energy than when running by itself.

Table 4: Energy comparisons for MTNCL Dadda Multiplier stacked and unstacked in the 130nm Process

Circuit	Energy Consumption (pJ)
Unstacked Single Dadda Multiplier Workload 1	122
Unstacked Single Dadda Multiplier Workload 2	117
Stacked Multipliers—both running	235
Stacked Multipliers—Workload 1 running while other is sleeping	117
Stacked Multipliers—Workload 2 running while other is sleeping	133

To show that the voltage stacking of MTNCL circuits is scalable, three identical MTNCL Dadda Multipliers designed using the 130nm process running different workloads were stacked upon one another and simulated. Figure 13 below shows the basic setup for the MTNCL Triple Stacked design.

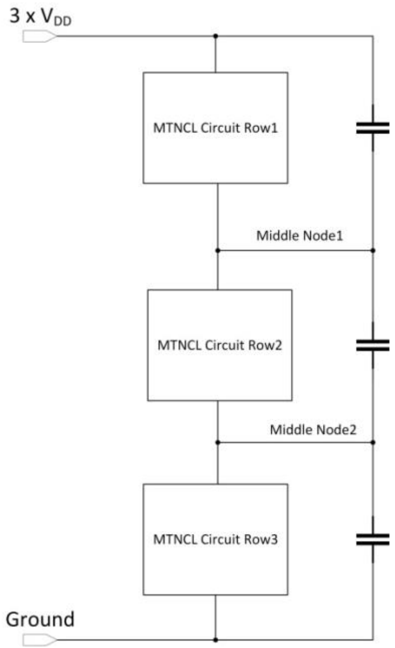

Figure 13: Simple MTNCL Triple Stacked Implementation

The capacitances used are the 50 fF for the top capacitor, 10 pF for the middle one and 20 pF for the bottom capacitor in the stack. The resulting waveform is shown in Figure 14 below with the two middle node voltage rails highlighted in blue and yellow. The blue signal is the voltage rail that oscillates around 2.4 V while the yellow one is the voltage rail oscillating around 1.2 V. The other waveforms are a single dual rail output from each of the 3 multipliers on different rows.

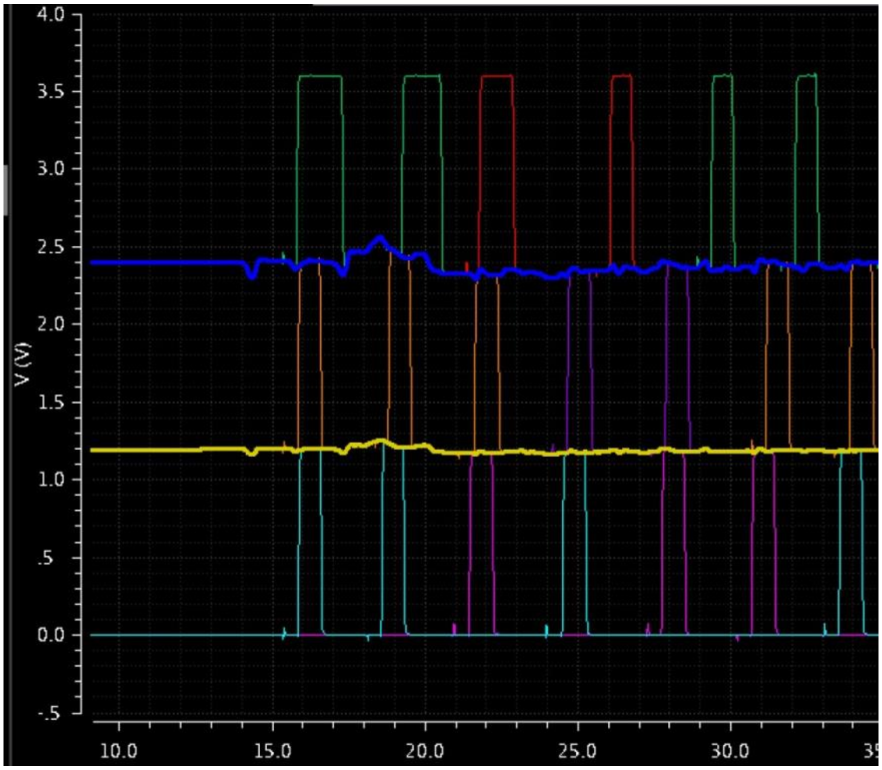

Figure 14: Same multiplier stacked three times running different workloads

After the basic testing of two and three identical MTNCL circuits are analyzed, different sized circuits are introduced into the stacking architecture for comprehensive verification purposes. The same structure is used in Figure 10, but this time with an 11-bit Ripple Carry Adder (RCA) stacked with the previously used 11-bit by 7-bit Dadda Multiplier. The multiplier is about 4 times larger than the RCA, which will demonstrate the effects of stacking different sized circuits.

There are more paths for the current to flow through in the larger circuit, so this reduces the resistance in the larger circuit compared to that of the smaller one. The effect this has on the system is that the middle node voltage shifts towards the larger circuit when both of them are operating. Figure 15 shows that in the case of the multiplier on top and the RCA on bottom, the middle node voltage highlighted in blue is oscillating around 1.5 V instead of the desired 1.2 V seen previously with the identical stacked circuits.

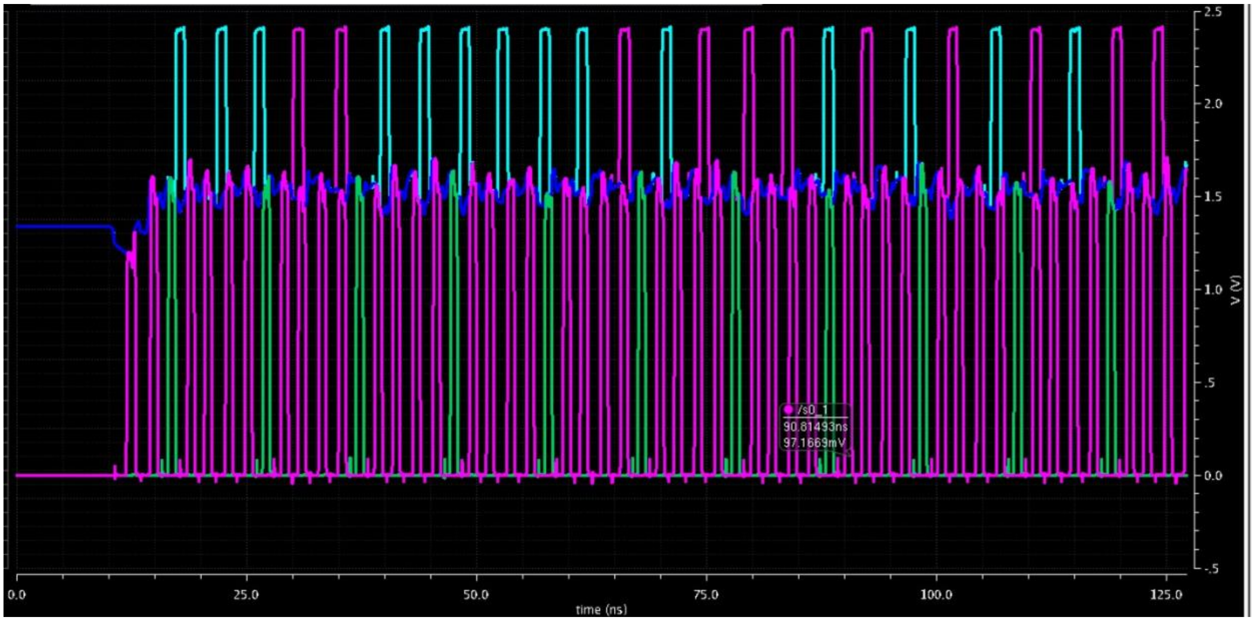

Figure 15: MTNCL stacked circuits with multiplier on row 1 and RCA on row 2 in the 130nm process

The first thing to note is that unlike the synchronous counterpart, the MTNCL stacked implementation works properly despite using different circuits in the stack. This is due to their robustness and timing independence. A second thing to address is because the middle node voltage oscillates around 1.5 V, the dynamic range for the top circuit is about 0.9 V and the bottom is 1.5 V. This has a direct effect on the performance of these two circuits because the multiplier on top actually slows down, while the RCA on bottom speeds up in reference to individual runtimes. Another benefit of using MTNCL circuitry to stack circuits is when the design has a need to sleep one or more circuits in the stack for an extended period of time. For example, when simulating an MTNCL multiplier stacked on an MTNCL RCA, then allowing one or the other to go to sleep, the middle node shifts towards the side of the circuit that is still operating. Figure 16 shows that in either situation, when either the top multiplier (left) or the bottom RCA (right) are put to sleep for an extended period, the other circuit still functions correctly despite having a smaller dynamic range of voltage.

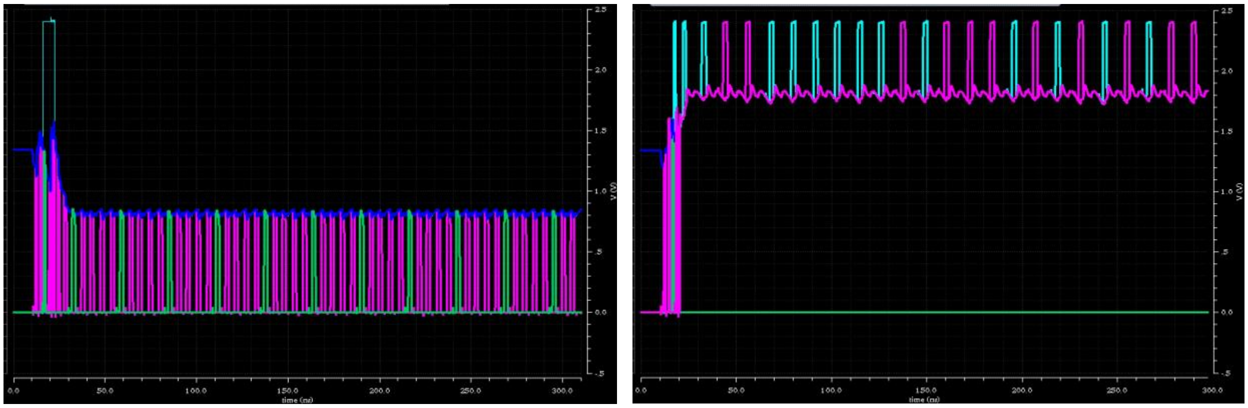

Figure 16: Top multiplier sleeping, bottom RCA running (left) and bottom RCA sleeping, top multiplier running (right) in the 130nm process

Stacking different sized circuits in the 32nm process has the same effect as seen in Figures 17 and 18. Although the decrease of the dynamic range directly affects the energy consumption and speed of the circuits, they still function properly, which is a critical improvement over the synchronous implementation where the circuits malfunction due to timing fluctuations and voltage noise. In addition, when the circuits are put to sleep (idle state) for an extended period of time, the other circuitry can still execute their desired functionalities.

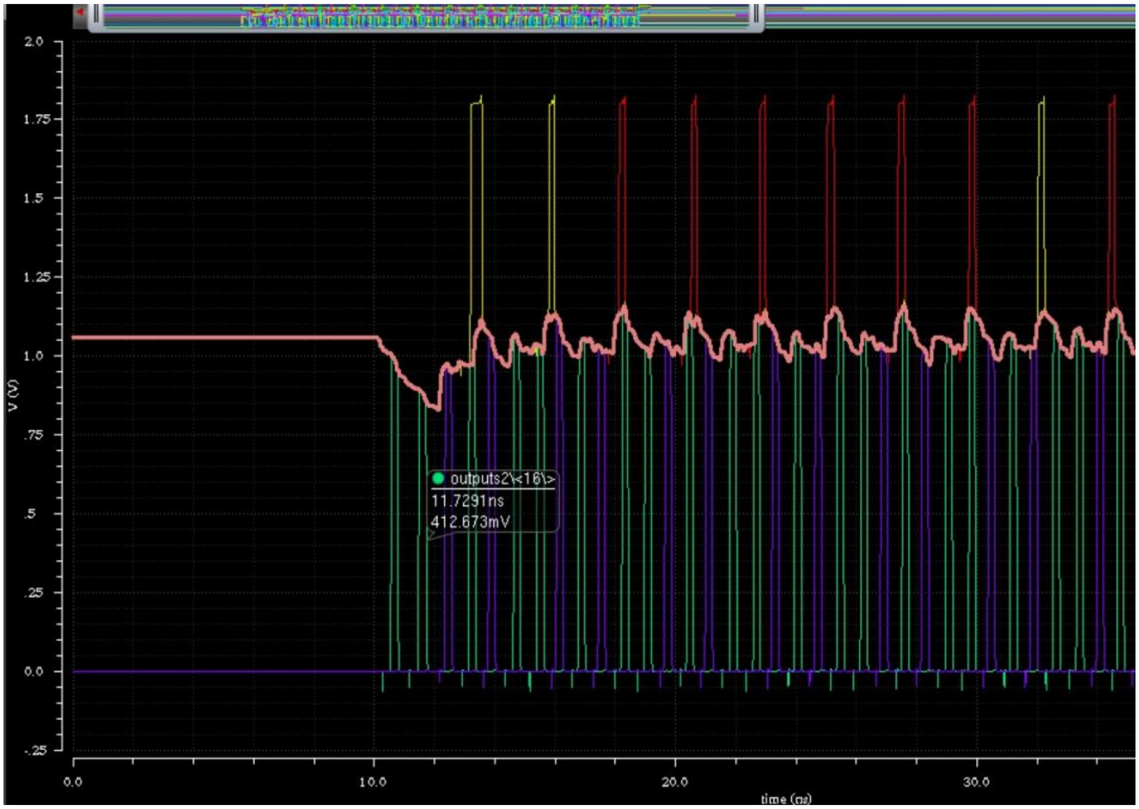

Figure 17: MTNCL stacked circuits with multiplier on row 1 and RCA on row 2 in the 32nm process

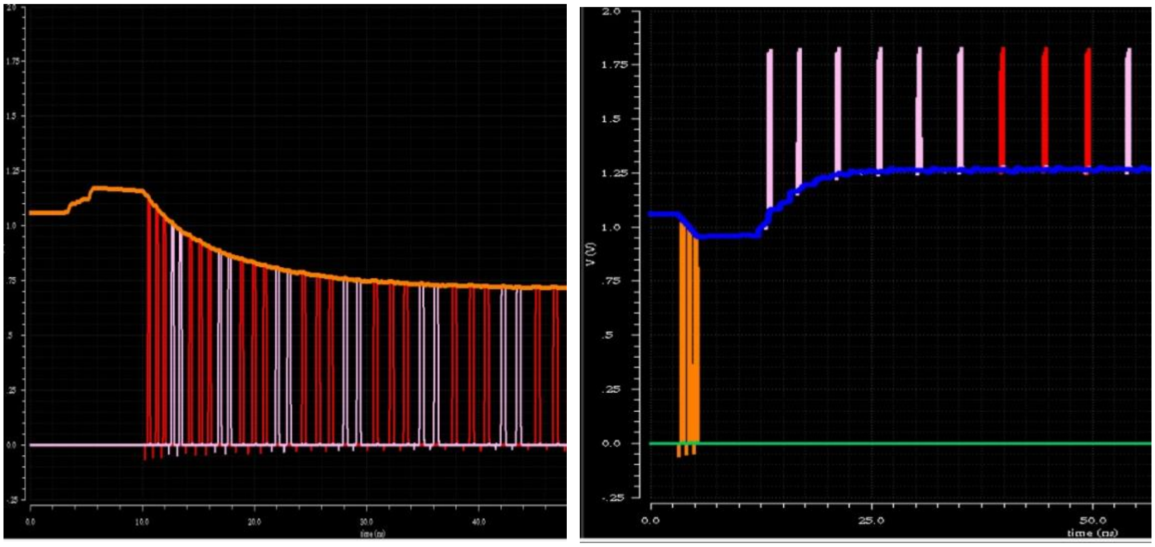

Figure 18: Top multiplier sleeping, bottom RCA running (left) and bottom RCA sleeping, top multiplier running (right) in the 32nm process

4 Advanced MTNCL Voltage Stacking

As shown in the previous chapter, when placing two similarly sized circuits on top of one another, the middle node voltage oscillates around the desired 1.2 V and 0.9 V range for the 130nm and 32nm processes, respectively, despite the workload while both circuits are active. However, when different sized circuits are stacked, the current mismatch between the two circuits causes the middle node voltage to shift drastically towards the larger circuit. In addition, when one circuit is put to sleep for an extended period of time and the other continues to run, the middle node voltage also shifts towards the operating circuit. In both situations, the dynamic ranges of the circuits change altering their speed and energy consumption. By adding some additional logic to manipulate the middle node voltage and current flow, an advanced stacking methodology has been designed to mitigate such effects and prevent the issues that arise when one or more of the circuits go to sleep. The additional logic is implemented in parallel to the MTNCL circuits in order to provide a separate path from one supply rail to the next that circumvents the slept circuit. This logic is controlled by separate signals generated at the system controller level indicating when the circuit is being put to sleep for an extended period of time. Figure 19 shows the overall diagram of how the additional logic will be implemented into the MTNCL Double Stacked architecture.

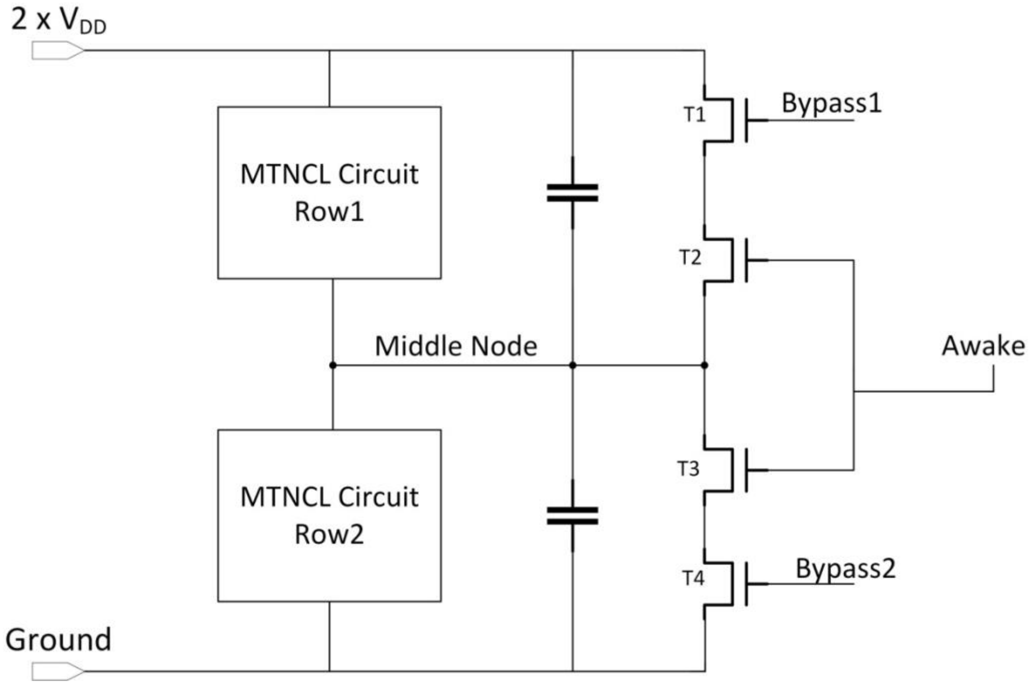

Figure 19: MTNCL Double Stacked Circuits with Bypass and Awake Transistors

When either circuit is running, the *Awake* signal stays high turning both the innermost transistors (T2 and T3) on. Now if either circuit is put to sleep for an extended period, their respective *Bypass* signal will also be enabled turning the transistor (T1 for row 1 or T4 for row 2) in the same row on and shorting either 2×V_{DD} to the middle node or the middle node to GND. By incorporating this logic, the middle node, which would normally shift drastically towards the circuit that is still running, can be pulled in the opposite direction, thereby increasing the dynamic range and speed for the working circuit. The *Awake* signal is set low (turning off transistors T2 and T3) when both circuits are put to sleep for an extended period, thereby blocking the direct path from 2×V_{DD} to GND while the two *Bypass* signals are enabled.

Before manipulating the size of the transistors, T1-T4, to show the effects they have on the performance of the circuits, simulations were carried out in both processes using the MTNCL Dadda Multipliers running different workloads. The data gathered compares the circuits running 12 input patterns separately and stacked to see how stacking the circuits affects them. Table 5 lists the execution times and the corresponding active energy and energy delay product (EDP) for the various simulations in both processes. Table 5 clearly shows that the overhead of the stacked architecture is negligible (~0.3%).

Table 5: Energy Delay Product Results from Stacked and Unstacked Multipliers in 130nm and 32nm Processes

Circuit Setup	**Execution Time (ns)**		**Energy Consumed (pJ)**		**Energy Delay Product (aJ*s)**	
Process	***32nm***	***130nm***	***32nm***	***130nm***	***32nm***	***130nm***
Individual Multiplier 1	33.65	41.62	20.64	55.62	0.694	2.32
Individual Multiplier 2	33.66	41.47	20.63	55.62	0.694	2.306
Total Unstacked	N/A	N/A	41.27	111.24	1.388	4.626
Both Running Stacked	33.66	41.5	41.33	111.8	1.391	4.64

In the two-stacked model, the sizing of the transistors (T1-T2 or T3-T4) in parallel to the circuit sleeping will have a direct effect on the circuit's functionality. That is, increasing the transistors' widths on row 1 will cause the circuit on row 2 to speed up and have a larger dynamic range. The same thing occurs to the circuit running on row 1 if the transistors' widths on row 2 are increased while the circuit on row 2 is idle. However, while the speed and dynamic range are increased, so is the active energy consumption. Therefore, a balance must be found in terms of the overall EDP, which is calculated by multiplying the execution time for a simulation by the active energy it consumes over that time. EDP can be utilized to find the optimal size for the transistors' widths that correspond to different instantiated circuits in the stacked architecture.

In the instance of the same MTNCL Dadda Multipliers stacked upon one another running at different workloads in the 130nm process, Table 6 and Figure 20, along with Table 7 and Figure 21, show that by changing the transistors' widths of the row from the slept circuit, it directly affects the speed and energy consumption of the working circuit. During the simulations in Table 6 and Figure 20, Multiplier 1 is operating on the top (row 1) while the Multiplier 2 is sleeping on bottom (row 2). Table 7 and Figure 21 comprise data from when Multiplier 1 is sleeping on row 1 while Multiplier 2 is operating on row 2.

Table 6: Multiplier 1 on Row 1 is Active while Multiplier 2 on Row 2 is Sleeping in the 130nm Process

Circuit Operating	Row 2's Transistors' Widths (µm)	Execution Time (ns)	Energy Consumption (pJ)	Energy Delayed Product (aJ·s)
Multiplier 1	1.5	58.62	82.12	4.814
Multiplier 1	2	52.18	88.83	4.635
Multiplier 1	2.5	47.75	95.89	4.579
Multiplier 1	3	44.62	103	4.596
Multiplier 1	3.5	42.39	110.1	4.668
Multiplier 1	4	40.77	117.2	4.78
Multiplier 1	4.5	39.73	124.7	4.953
Multiplier 1	5	39.22	132.9	5.213
Multiplier 1	5.5	38.9	141.4	5.502

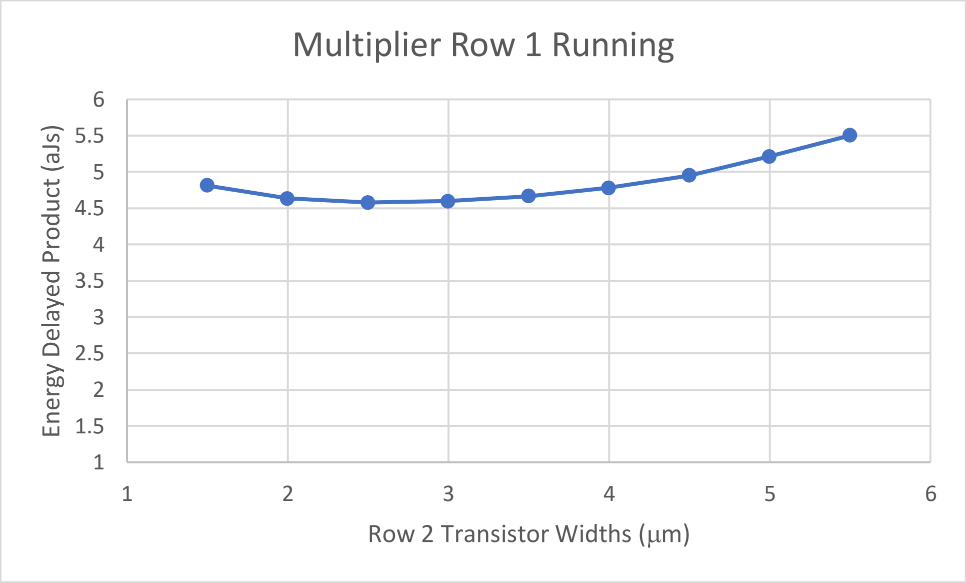

Figure 20: Energy Delay Product Curve as a Result of the Transistor Widths from Row 2 in the 130nm Process

Table 7: Multiplier 1 on Row 1 is Sleeping while Multiplier 2 on Row 2 is Active in the 130nm Process

Circuit Operating	Row 1's Transistors' Widths (µm)	Execution Time (ns)	Energy Consumption (pJ)	Energy Delayed Product (aJ·s)
Multiplier 2	0.16 (minimum)	70.5	49.19	3.47
Multiplier 2	0.5	62.24	55.62	3.46
Multiplier 2.	1.0	54.82	64.84	3.56
Multiplier 2	1.5	49.9	73.92	3.69
Multiplier 2	2	46.64	82.41	3.844
Multiplier 2	2.5	44.5	89.82	3.997
Multiplier 2	3	43.01	96.2	4.138
Multiplier 2	3.5	41.91	101.8	4.266
Multiplier 2	4	41.09	106.8	4.388
Multiplier 2	4.5	40.51	111.3	4.511
Multiplier 2	5	40.17	115.7	4.646
Multiplier 2	5.5	40	119.8	4.791

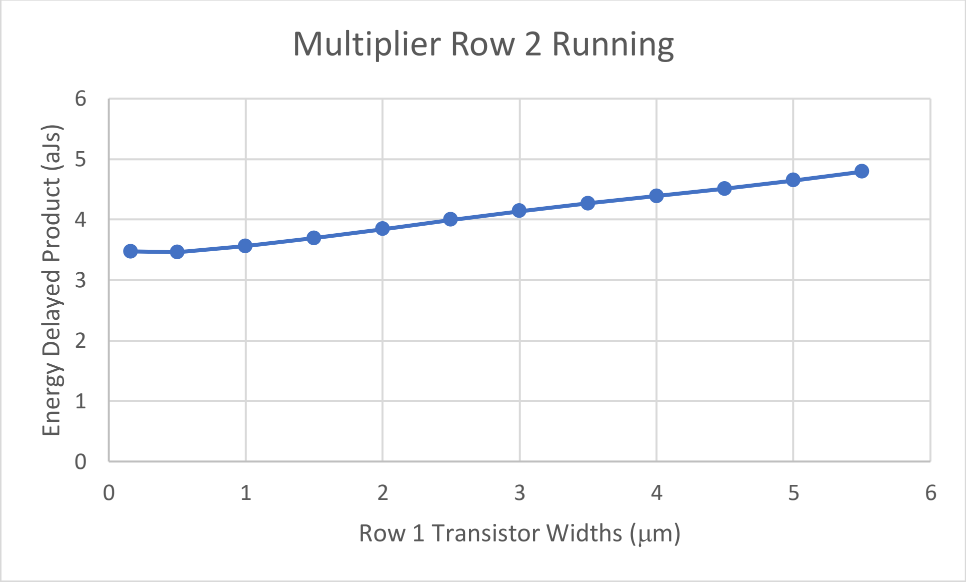

Figure 21: Energy Delay Product Curve as a Result of the Transistor Widths from Row 1 in the 130nm Process

The results from Tables 6 and 7 shows that increasing the transistors' widths shortens the execution time for the non-idle circuit, but it does so by increasing the amount of energy being consumed. Figures 20 and 21 each shows there exists a transistor width that produces a lowest point on the graph, which correlates to the optimal energy delay product. The same simulation carried out in the 32nm process produces similar results shown in Tables 8 and 9, along with their corresponding Figures, 22 and 23.

Table 8: Multiplier 1 on Row 1 is Active while Multiplier 2 on Row 2 is Sleeping in the 32nm Process

Circuit Operating	Row 2's Transistors' Widths (nm)	Execution Time (ns)	Energy Consumption (pJ)	Energy Delayed Product (aJ·s)
Multiplier 1	104	50.76	28	1.421
Multiplier 1	312	40.49	30.77	1.246
Multiplier 1	624	35.63	36.27	1.292
Multiplier 1	936	34.41	42.77	1.472
Multiplier 1	1248	33.77	49.53	1.623

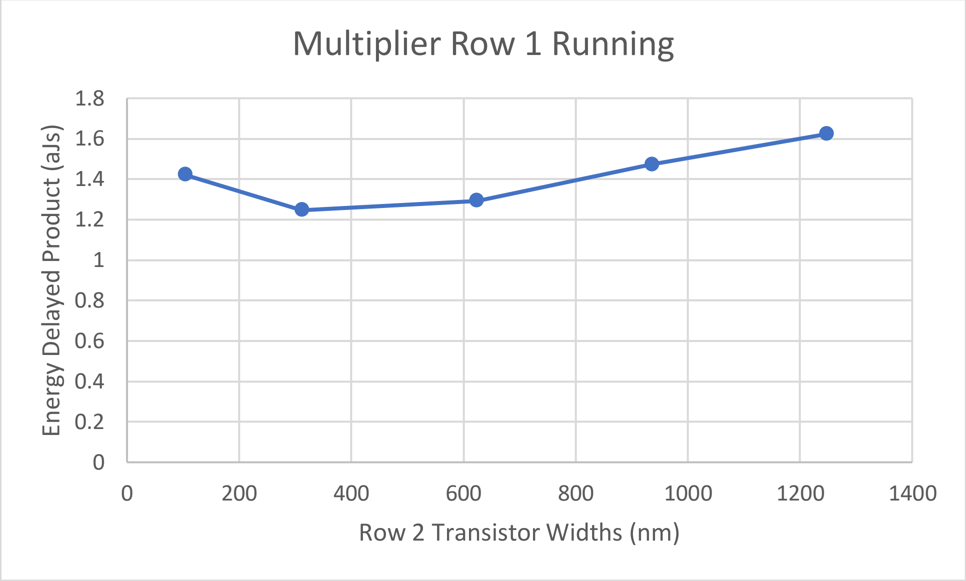

Figure 22: Energy Delay Product Curve as a Result of the Transistor Widths from Row 2 in the 32nm Process

Table 9: Multiplier 1 on Row 1 is Sleeping while Multiplier 2 on Row 2 is Active in the 32nm Process

Circuit Operating	Row 1's Transistors' Widths (nm)	Execution Time (ns)	Energy Consumption (pJ)	Energy Delayed Product (aJ·s)
Multiplier 2	104	47.78	19.87	0.9493
Multiplier 2	312	36.61	24.94	0.9131
Multiplier 2	624	34.79	33.8	1.176
Multiplier 2	936	34.31	40.43	1.387
Multiplier 2	1248	34.12	45.46	1.55

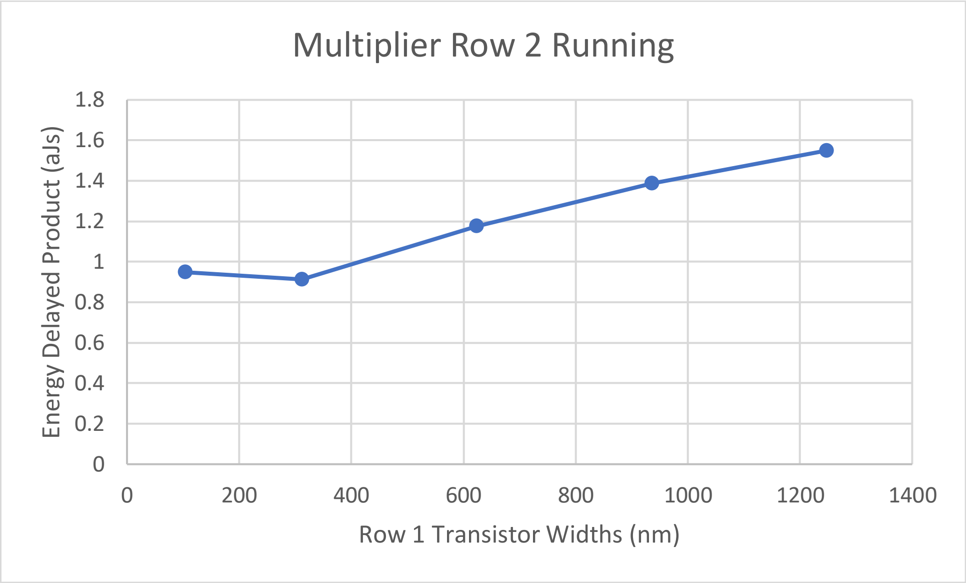

Figure 23: Energy Delay Product Curve as a Result of the Transistor Widths from Row 1 in the 32nm Process

The data from Table 5 demonstrates that by stacking the circuits, the performance is about the same as when they run individually (not stacked), so the benefit has already been gained through the simplified power management system. Furthermore, by incorporating the additional logic to manipulate the dynamic range of the circuits when others are sleeping, the data shows that improvements to the EDP can be accomplished. Therefore, the designer must decide on whether saving energy or running faster will be more crucial to their end goal when designing their combination of stacked circuitry. Other factors such as the circuit size, design process and frequency of active period are also considerations for deciding on which circuit will need to be placed where in the stack and what size of transistors will need to be implemented.

Tables 10 and 11, as well as Figures 24-27, display data taken from when stacking two different circuits on top of one another in the 130nm process. The two circuits used are the same Multiplier and RCA that were implemented in the simple stacking architecture from before. Since the multiplier is 4 times larger than the RCA and they have a different number of inputs and outputs, the number of data sets each complete in the same amount of time when they run stacked together is different. For the 130nm process, the multiplier executes 11 data sets in the same time the RCA executes 25 data sets, so the simulations will end when the respective circuits execute their target number of data sets when the other is sleeping. Each circuit is simulated on each row while different transistor widths are tested for comparisons. The data follows the same trend that while the transistors' widths get larger, the speed increases but so does the energy consumption. Therefore, the EDP will be used to find the best operating parameters.

Table 10: Data from Circuits Active on Row 1 while Circuit on Row 2 Sleeps in the 130nm Process

Circuit Operating on Row 1	Row 2's Transistors' Widths (μm)	Execution Time (ns)	Energy Consumption (pJ)	Energy Delayed Product (aJ·s)
RCA	0.5	96.7	38.93	3.76
RCA	1	68.53	50.91	3.49
RCA	1.5	58.55	62.96	3.69
RCA	2	53.92	75.32	4.06
Multiplier	2	54.98	81.36	4.47
Multiplier	3	43.98	95.42	4.196
Multiplier	4	37.39	107.2	4.01
Multiplier	5	34.42	121.1	4.168

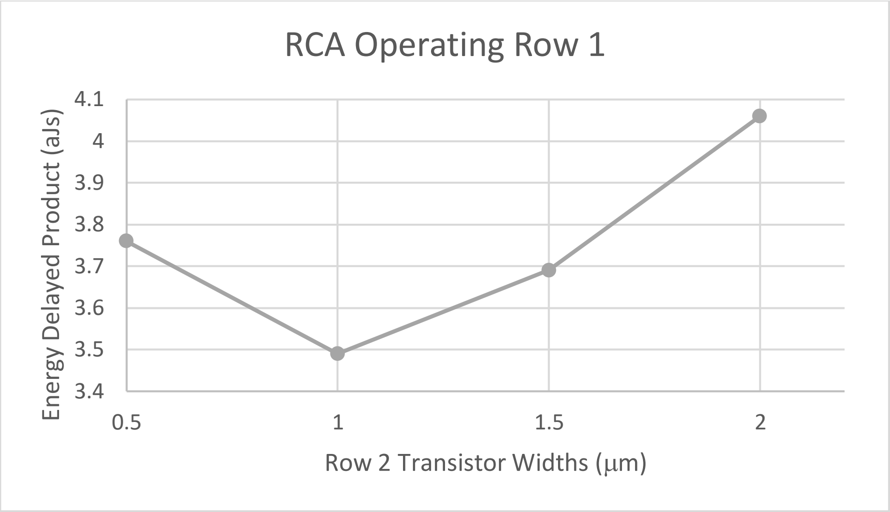

Figure 24: EDP Curve for RCA Operating on Row 1 in the 130nm Process

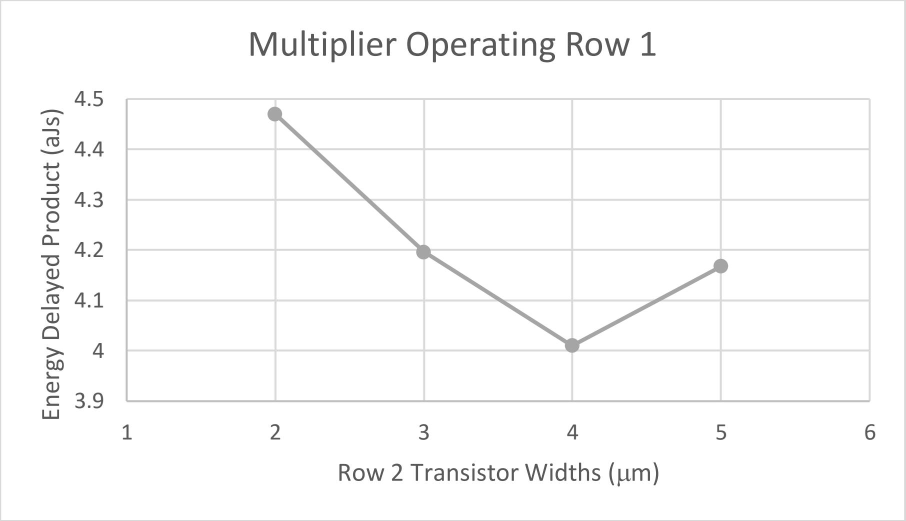

Figure 25: EDP Curve for Multiplier Operating on Row 1 in 130nm Process

Table 11: Data from Circuits Active on Row 2 while Circuit on Row 1 Sleeps in the 130nm Process

Circuit Operating on Row 2	Row 1's Transistors' Widths (μm)	Execution Time (ns)	Energy Consumption (pJ)	Energy Delayed Product (aJ·s)
Multiplier	2	45.535	85.91	3.91
Multiplier	3	39.42	96.14	3.79
Multiplier	4	36.15	104.2	3.77
Multiplier	5	34.17	110.7	3.78
Multiplier	6	32.81	116.2	3.81
RCA	0.5	82.94	41.75	3.46
RCA	1	66.64	49.1	3.27
RCA	1.5	60.31	54.07	3.26
RCA	2	56.88	57.83	3.29

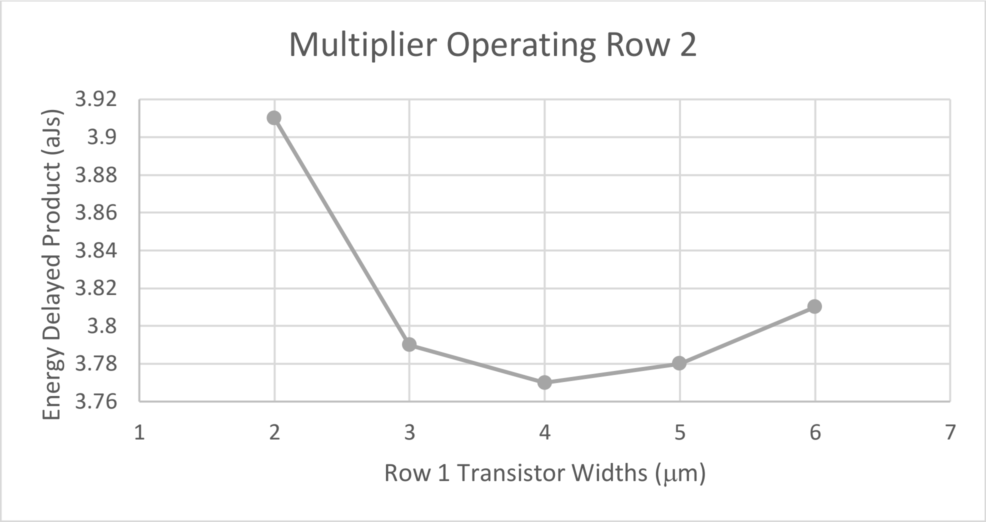

Figure 26: EDP Curve for Multiplier Operating on Row 2 in the 130nm Process

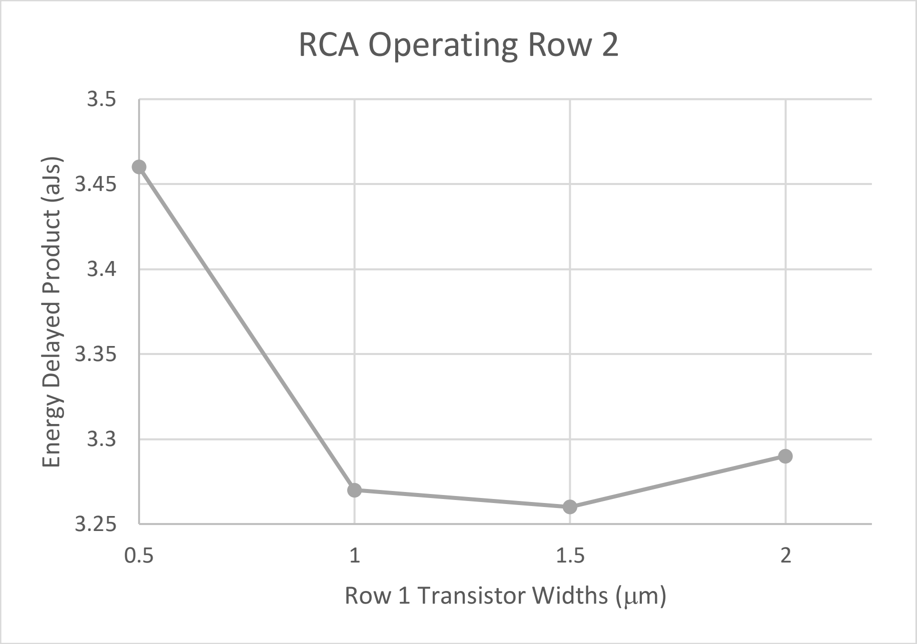

Figure 27: EDP Curve for RCA Operating on Row 2 in the 130nm Process

To illustrate this better, recall the waveforms from Figure 16 in the previous chapter of the simple voltage stacking model. When the multiplier is stacked on top of the RCA in the 130nm process and one of the circuits is put to sleep while the other continues to run, the dynamic range decreases for the running circuit while also decreasing its speed. Now, with the new logic implemented and a transistor width chosen to achieve the desired performance from the circuits, Figures 28 and 29 below depict the running circuit's increased dynamic range that corresponds to a much faster overall speed, more than double in these simulations.

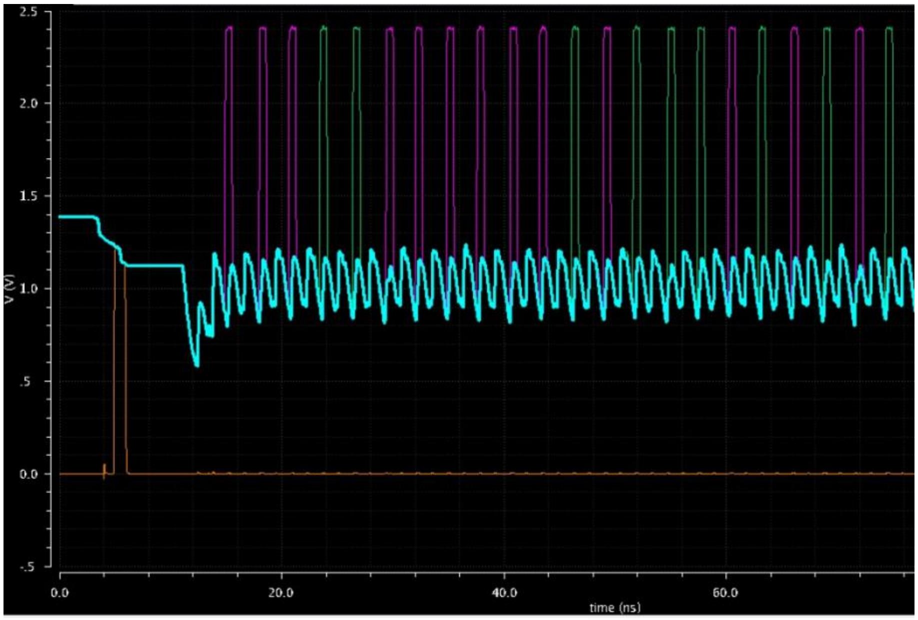

Figure 28: Simulation Waveform with Increased Transistor Widths for Multiplier Active and RCA Sleeping in the 130nm Process

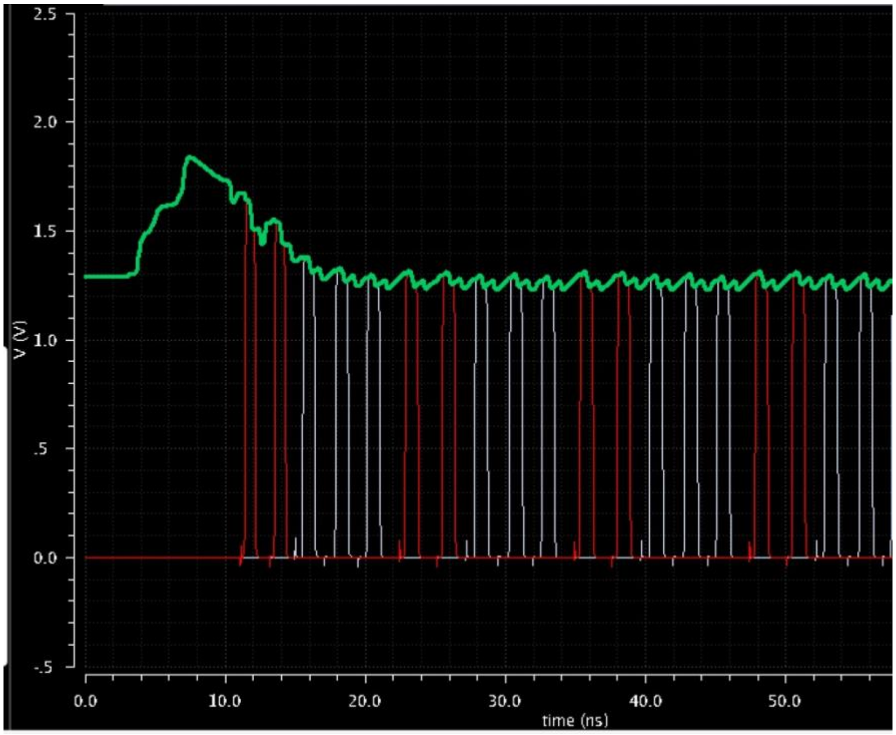

Figure 29: Simulation Waveform with Increased Transistor Widths for Multiplier Sleeping and RCA Active in the 130nm Process

The same simulations are carried out in the 32nm process and although the optimal EDP is different, the same trends emerge. One difference is the number of data sets each circuit executes when stacked together in this process is 12 for the multiplier and 40 for the RCA, so these are used when comparing the circuits of one row while the other is sleeping. Tables 12 and 13, along with Figures 30-35, display the data taken from the 32nm process.

Table 12: Data from Circuits Active on Row 1 while Circuit on Row 2 Sleeps in the 32nm Process

Circuit Operating on Row 1	Row 2's Transistors' Widths (nm)	Execution Time (ns)	Energy Consumption (pJ)	Energy Delayed Product (aJ·s)
RCA	104	52.81	19.62	1.036
RCA	312	38.83	22.78	0.8846
RCA	624	32.16	27.54	0.8857
RCA	936	29.2	32.41	0.9462
RCA	1248	27.53	37.45	1.031
Multiplier	104	54.09	21.82	1.181
Multiplier	312	42.4	23.84	1.011
Multiplier	624	32.41	27.29	0.8846
Multiplier	936	29.86	33.77	1.008
Multiplier	1248	29.27	41.71	1.221

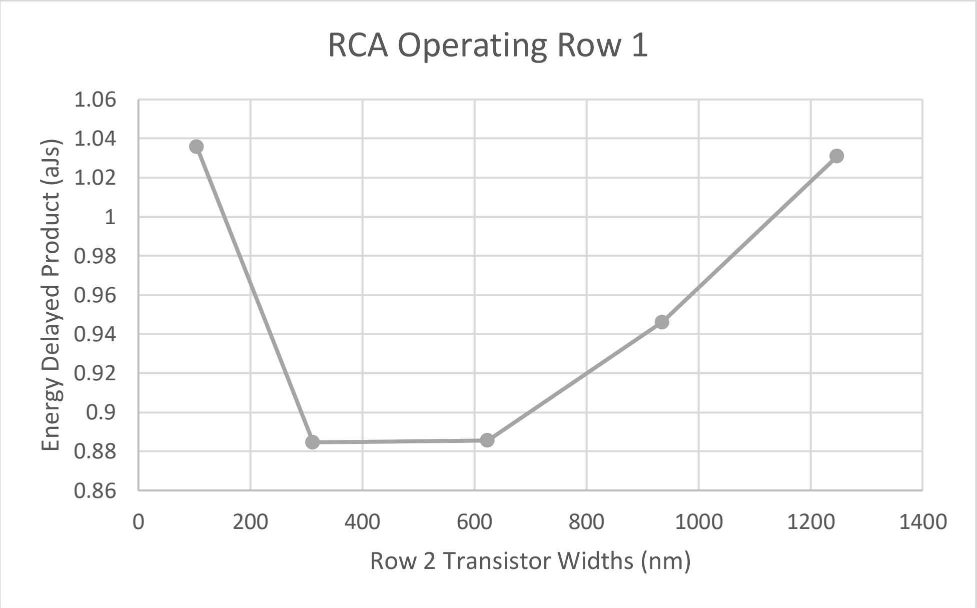

Figure 30: EDP Curve for RCA Operating on Row 1 in the 32nm Process

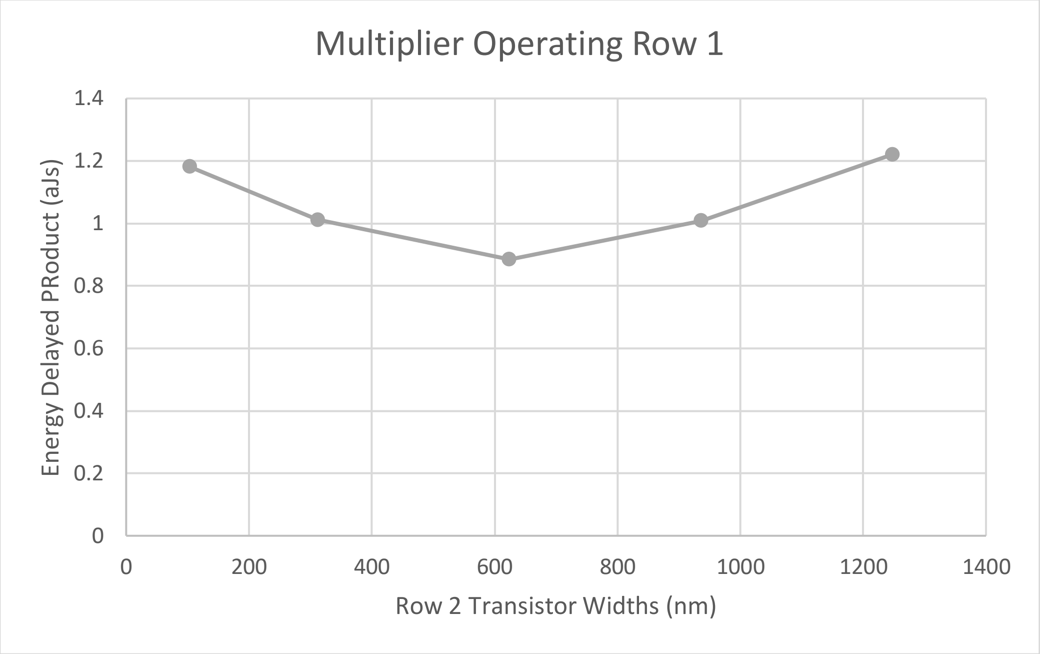

Figure 31: EDP Curve for Multiplier Operating on Row 1 in 32nm Process

Table 13: Data from Circuits Active on Row 2 while Circuits on Row 1 Sleeps in the 32nm Process

Circuit Operating on Row 2	Row 1’s Transistors’ Widths (nm)	Execution Time (ns)	Energy Consumption (pJ)	Energy Delayed Product (aJ·s)
Multiplier	104	47.62	18.25	0.8697
Multiplier	312	33.84	22.61	0.7652
Multiplier	624	30.15	30.47	0.9186
Multiplier	936	29.65	37.12	1.101
Multiplier	1248	29.44	42.09	1.239
RCA	104	57.97	16.81	0.9744
RCA	312	40.47	20.86	0.8442
RCA	624	33.89	25.01	0.8477
RCA	936	31.17	28.08	0.8752
RCA	1248	29.64	30.53	0.905

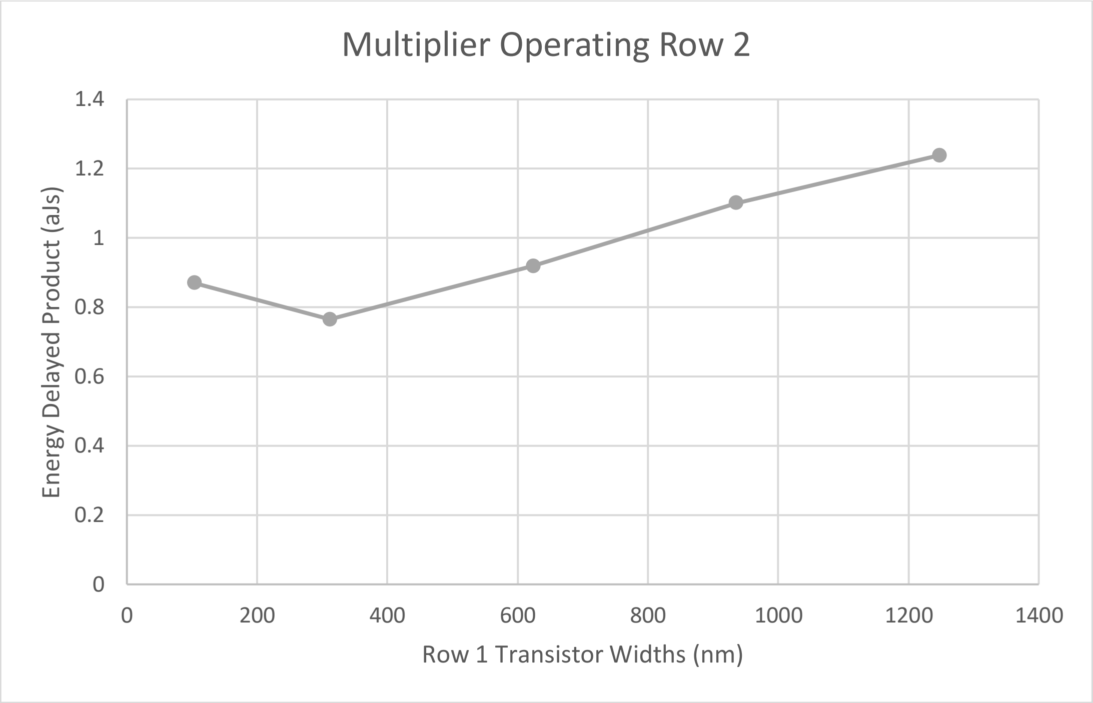

Figure 32: EDP Curve for Multiplier Operating on Row 2 in the 32nm Process

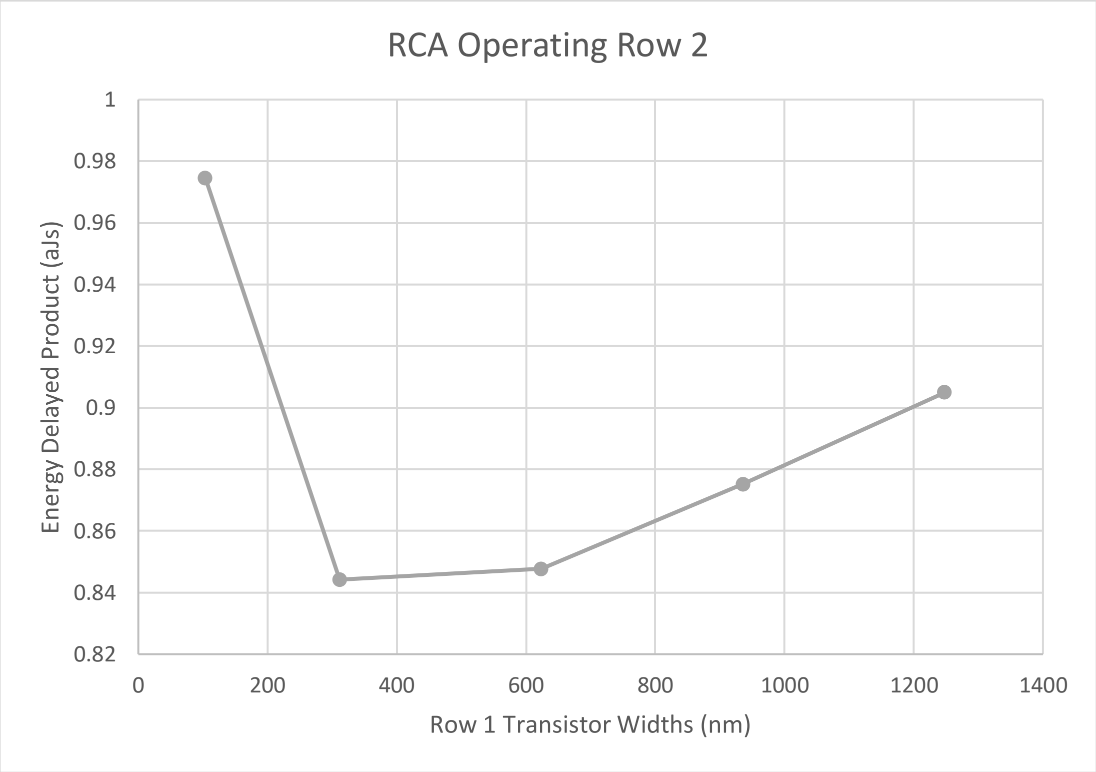

Figure 33: EDP Curve for RCA Operating on Row 2 in the 32nm Process

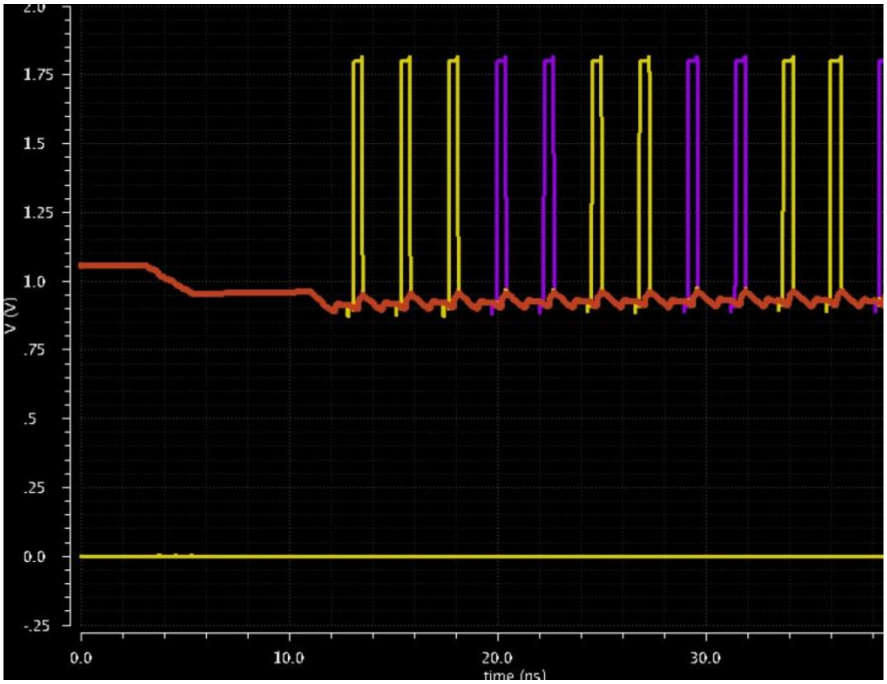

Figure 34: Simulation Waveform with Increased Transistor Widths for Multiplier Active and RCA Sleeping in the 32nm Process

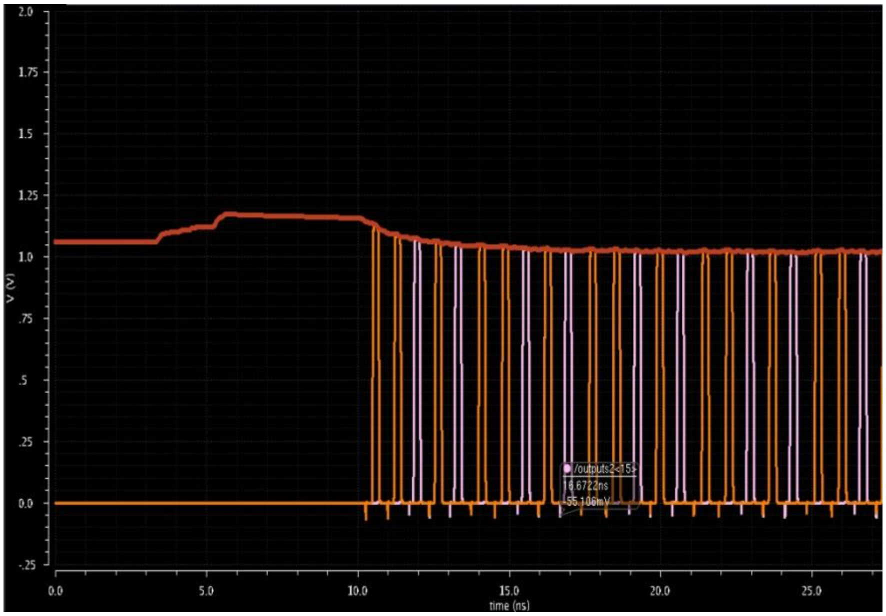

Figure 35: Simulation Waveform with Increased Transistor Widths for Multiplier Sleeping and RCA Active in the 32nm Process

Although an optimal transistor width for each stacked implementation exists where the EDP is the lowest, this may not be the best option depending on the desired application the circuits are being implemented for. Therefore, designers will be able to fine-tune the circuit parameters for their needs since this voltage stacking methodology takes advantages of the robustness of MTNCL. For example, a circuit that is put to sleep for an extended period of time and then called to work for just short period may want to complete the computation very quickly while consuming more energy in that short span. Conversely, a circuit that will need to continue to run for an extended period of time could do so at a slower speed in order to consume much less energy over its duty cycle. Either option is available by manipulating the size of the Bypass and Awake (T1-T4) Transistors in the additional logic presented in the advanced MTNCL voltage stacking model. The same can be done for implementing the MTNCL Triple Stacked model as shown in Figure 36. The designer will just need to specify the (T1-T6) transistor widths needed for each of the 3 rows based upon the desired performance of the circuits running in the alternate rows.

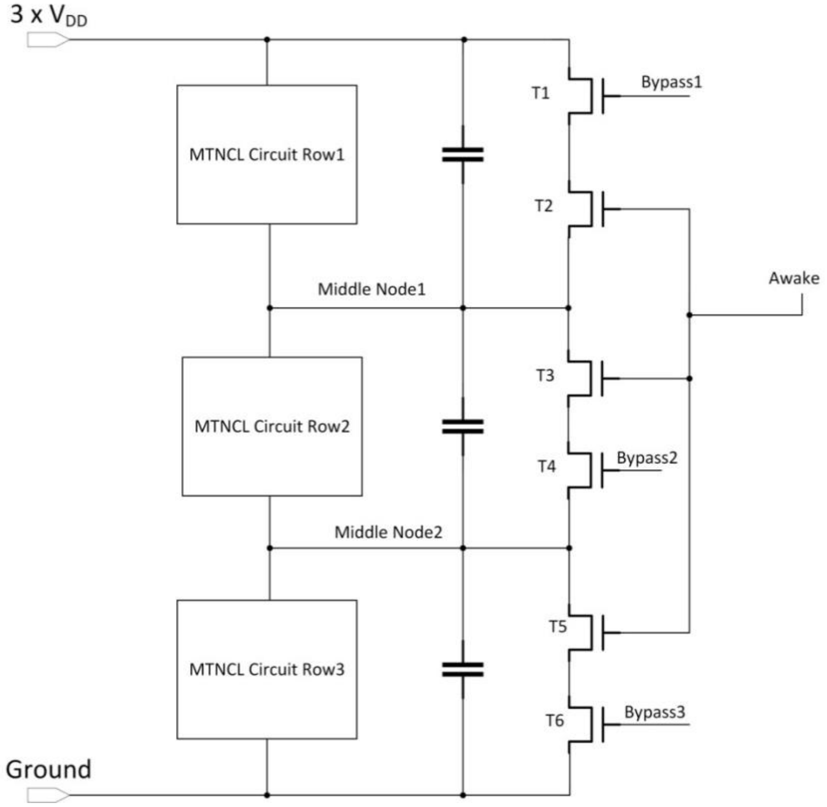

Figure 36: MTNCL Triple Stacked Circuits with Bypass and Awake Transistors

5 Physical Implementation and Results

While schematic simulation results of MTNCL voltage stacking from the previous chapter show that by adding the additional logic and manipulating the sizing can be promising, it needs to be physically laid out to demonstrate its feasibility. With the semiconductor processes continuing to get smaller in feature size, it is more meaningful to place and route the voltage stacked MTNCL implementation in a more advanced process node, e.g., the GLOBALFOUNDRIES 32nm SOI process, using the Cadence Innovus tool. Since the 32nm process is a fully-depleted SOI process, there are no body contacts for any of the transistors, which makes it ideal for stacking multiple circuits in series. The cross-section view of stacking two simple inverters in the 32nm process can be seen in Figure 37, which can be scaled to work with much larger circuits and with more circuitry implemented into the stacked architecture.

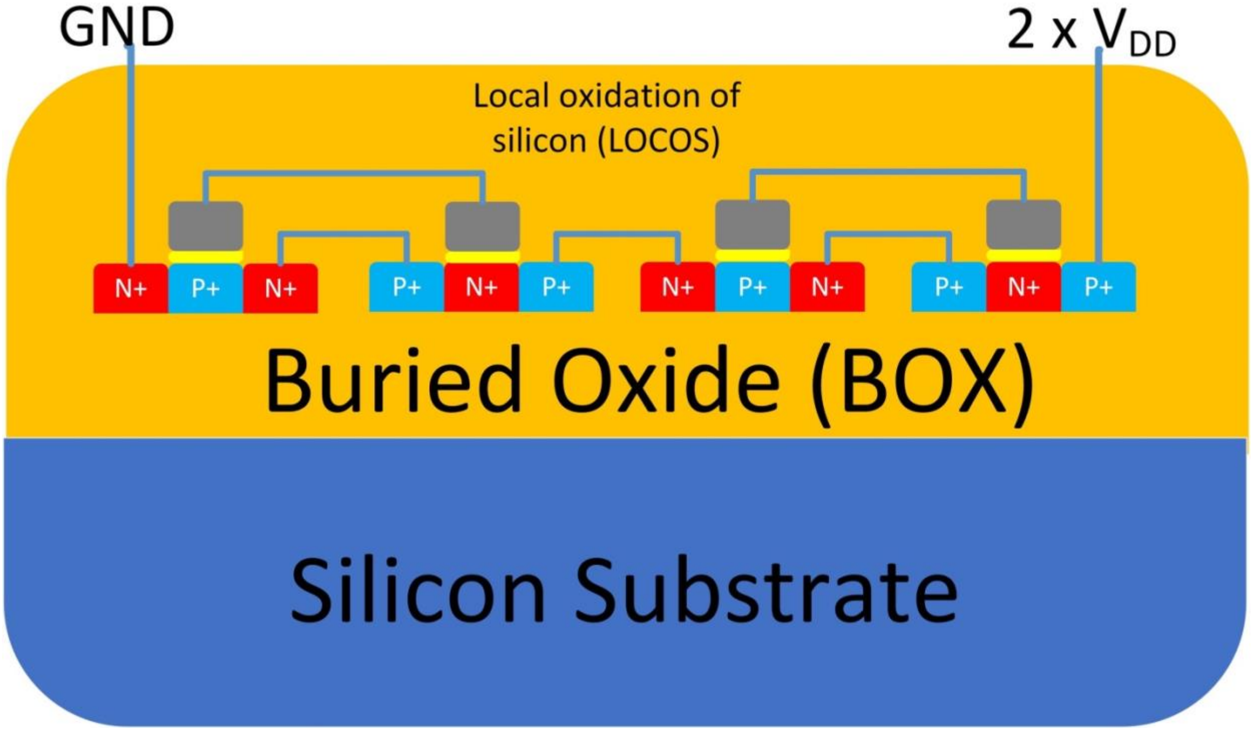

Figure 37: Cross-Section of Two Inverters Stacked in the 32nm SOI Process

Both the MTNCL Ripple Carry Adder and MTNCL Dadda Multiplier are placed and routed individually using the Cadence Innovus tool. The resulting designs are shown in Figures 38 and 39, respectively.

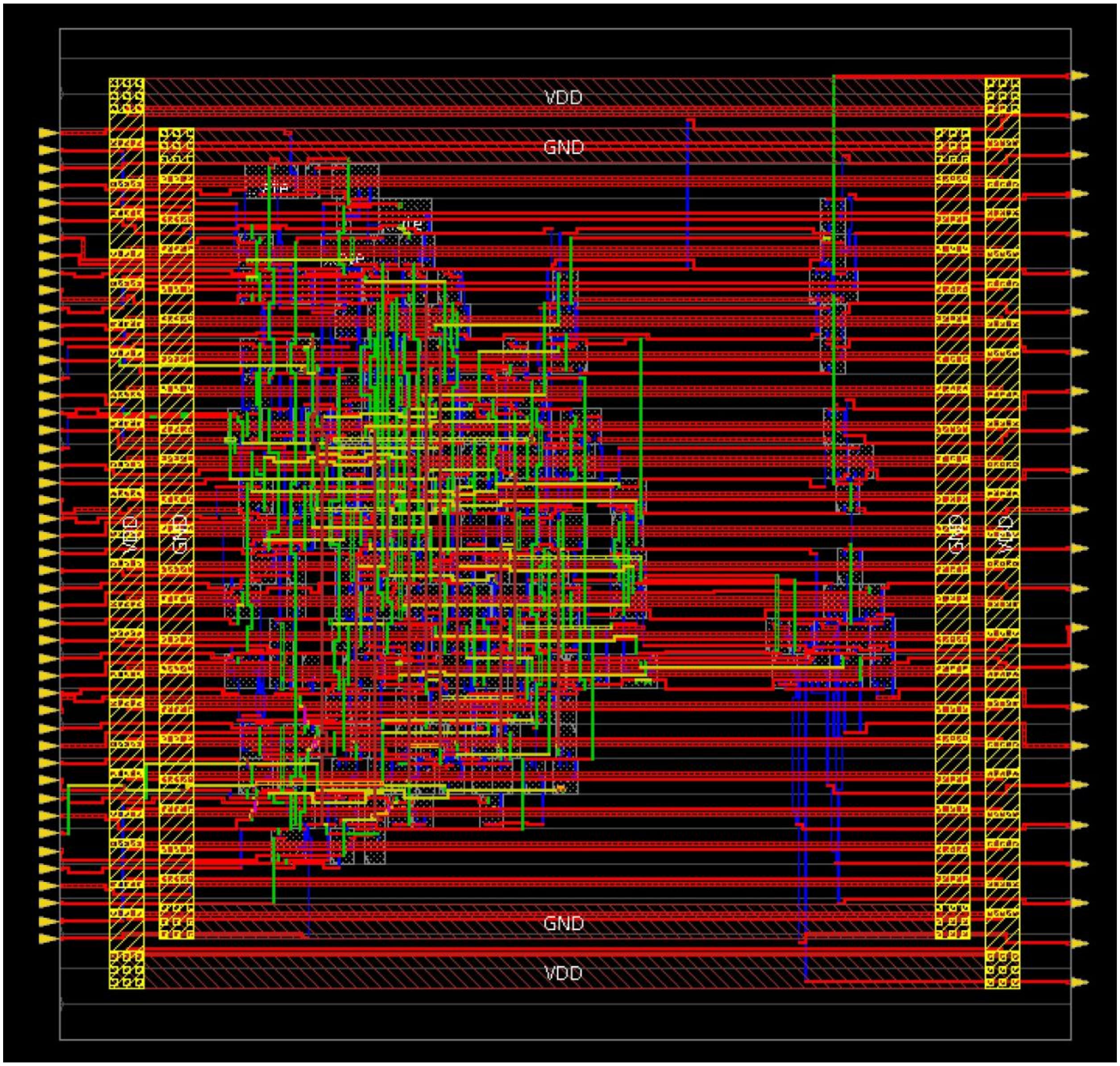

Figure 38: MTNCL Ripple Carry Adder Placed and Routed in Cadence Innovus Tool

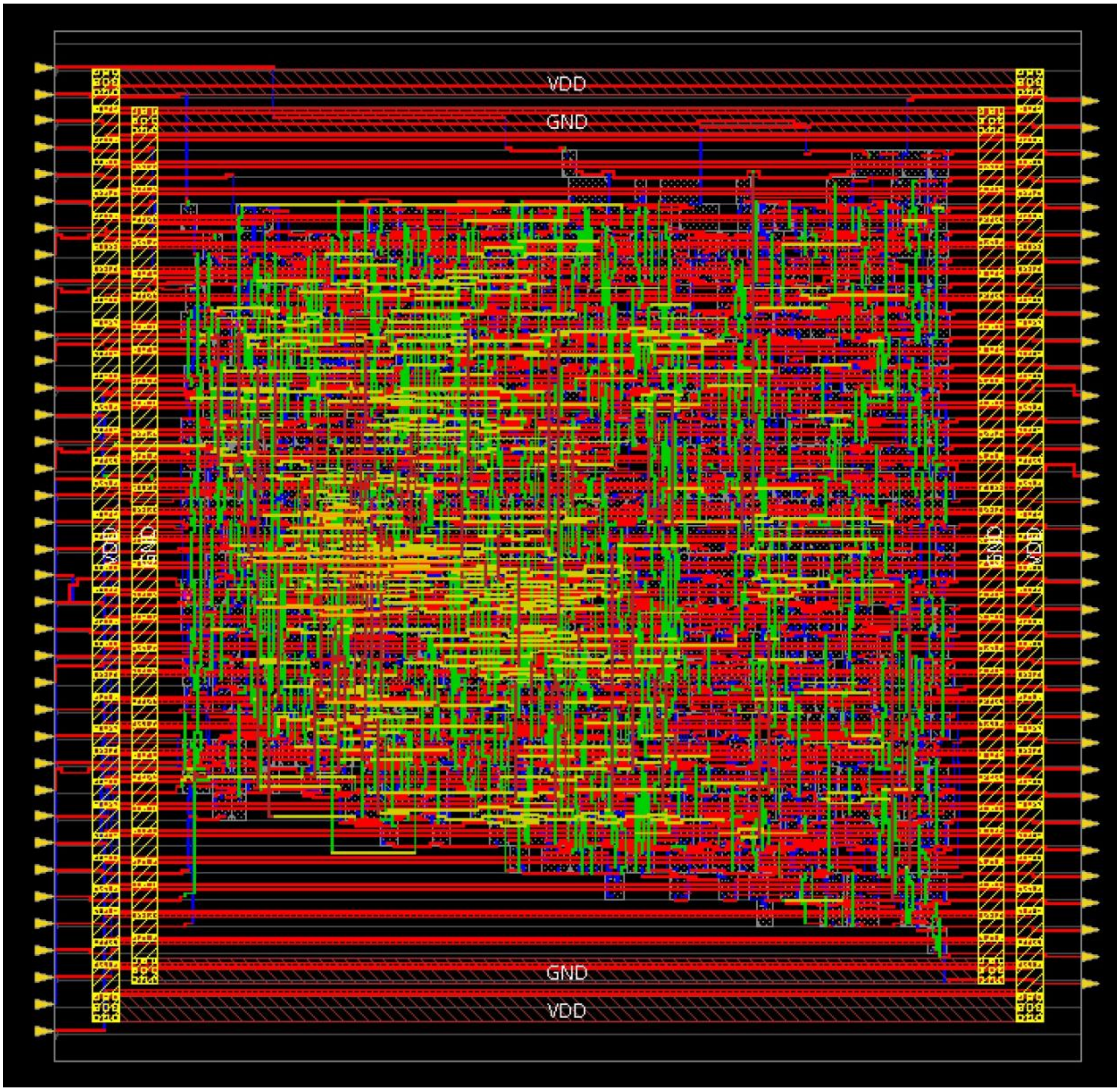

Figure 39: MTNCL Dadda Multiplier Placed and Routed in Cadence Innovus Tool

To compare the sizes of the two different circuits, core utilization area and number of gates are examined. The MTNCL RCA is laid out on a 60 μm × 60 μm grid, so the area is 3,600 μm^2. The core utilization of the RCA is only 16.08%, so the actual area that the circuit covers is approximately 579 μm^2. The MTNCL Dadda Multiplier is laid out on an 80 μm × 80 μm grid, so the area is 6,400 μm^2. The core utilization of the multiplier is 40.77%, so the actual area that the circuit covers is approximately 2,609 μm^2. Therefore, the RCA covers roughly 22% of the same area as the multiplier. Similarly, the multiplier is comprised of 714 gates including buffers while the RCA only consists of 161 gates including buffers, so the RCA has approximately 23% the number of gates as the multiplier has. Considering both comparisons, the multiplier is about four times larger than the RCA.

Since both designs are placed and routed individually, they are imported into Cadence Virtuoso separately. This allows them to pass the layout versus schematic (LVS) tests individually before being stacked. In addition, this allows parasitic extraction (PEX) to be run on them individually so they can be compared to the stacked architecture PEX results. The LVS-clean placed and routed designs in Cadence Virtuoso are shown in Figures 40 and 41, respectively.

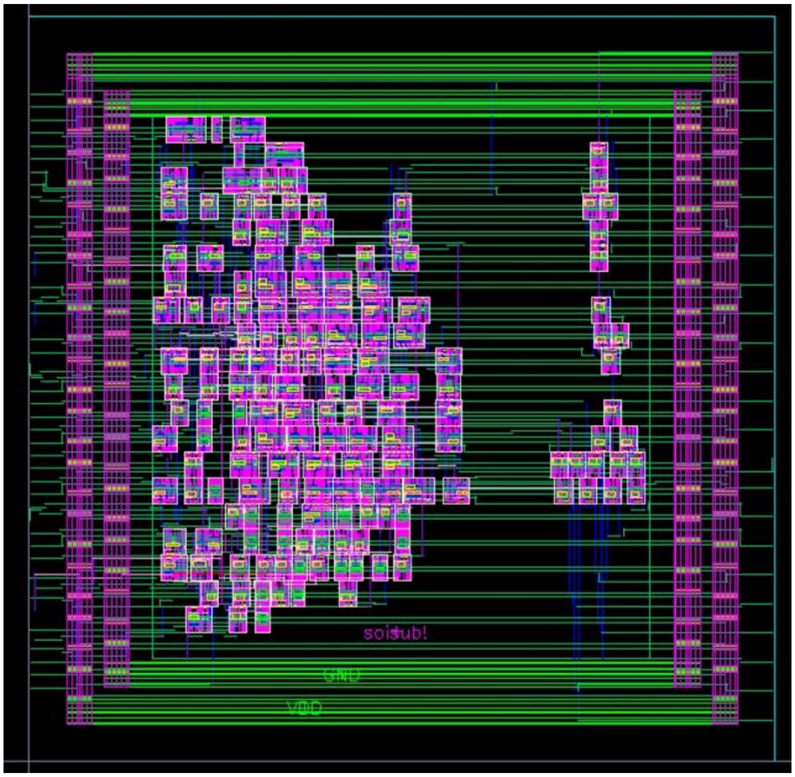

Figure 40: LVS-Clean RCA Design in Cadence Virtuoso

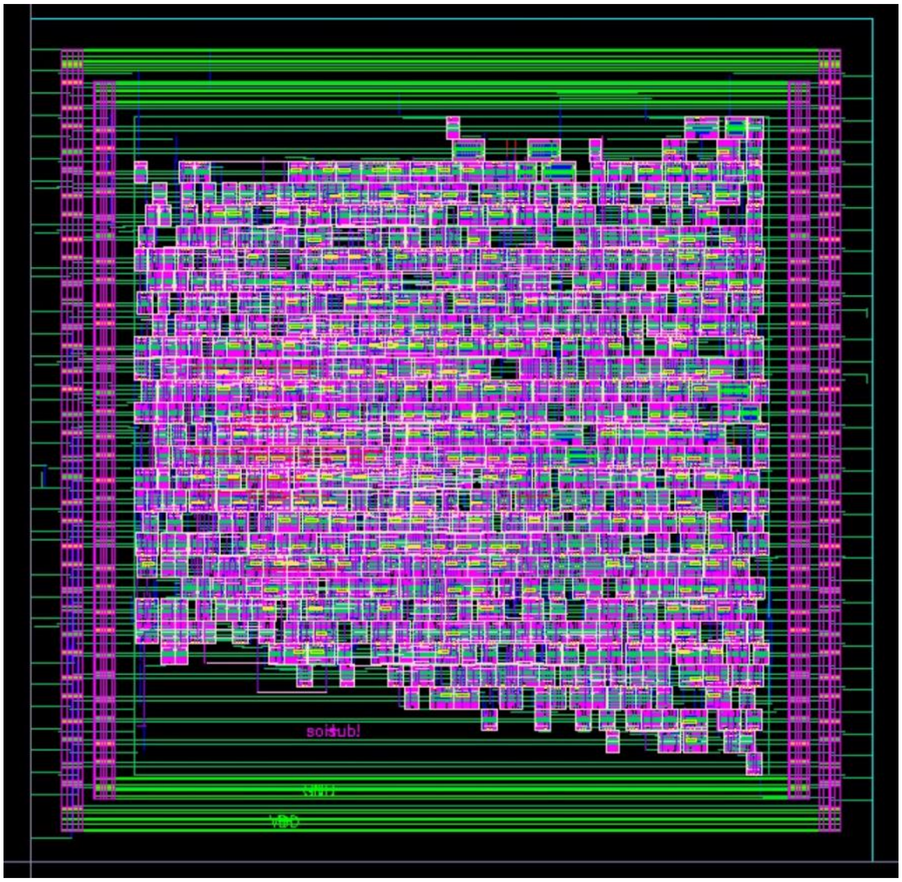

Figure 41: LVS-Clean Multiplier Design in Cadence Virtuoso

With both designs LVS clean, they are manually placed in the same layout window with the capacitors so they all fit as close together as possible without creating any design rule errors. The capacitors that are used are the vertical natural capacitors (vncap) since they are easily implemented in the 32nm SOI process using the given parameterized cell and can be sized to meet the 50 fF and 10 pF requirements from the schematic simulations. The actual sizes for the vncaps are a calculated 50.24116 fF from a width of 5.038 μm and a length of 5.2 μm, and a calculated 10.05664 pF from a width of 69.802 μm and a length of 72.0 μm. The two designs and two capacitors are manually routed together with the two Bypass Transistors and two Awake Transistors. Figures 42 and 43 show the complete LVS clean Double Stacked implementation for the MTNCL RCA and Dadda Multiplier in both configurations with capacitors and additional logic.

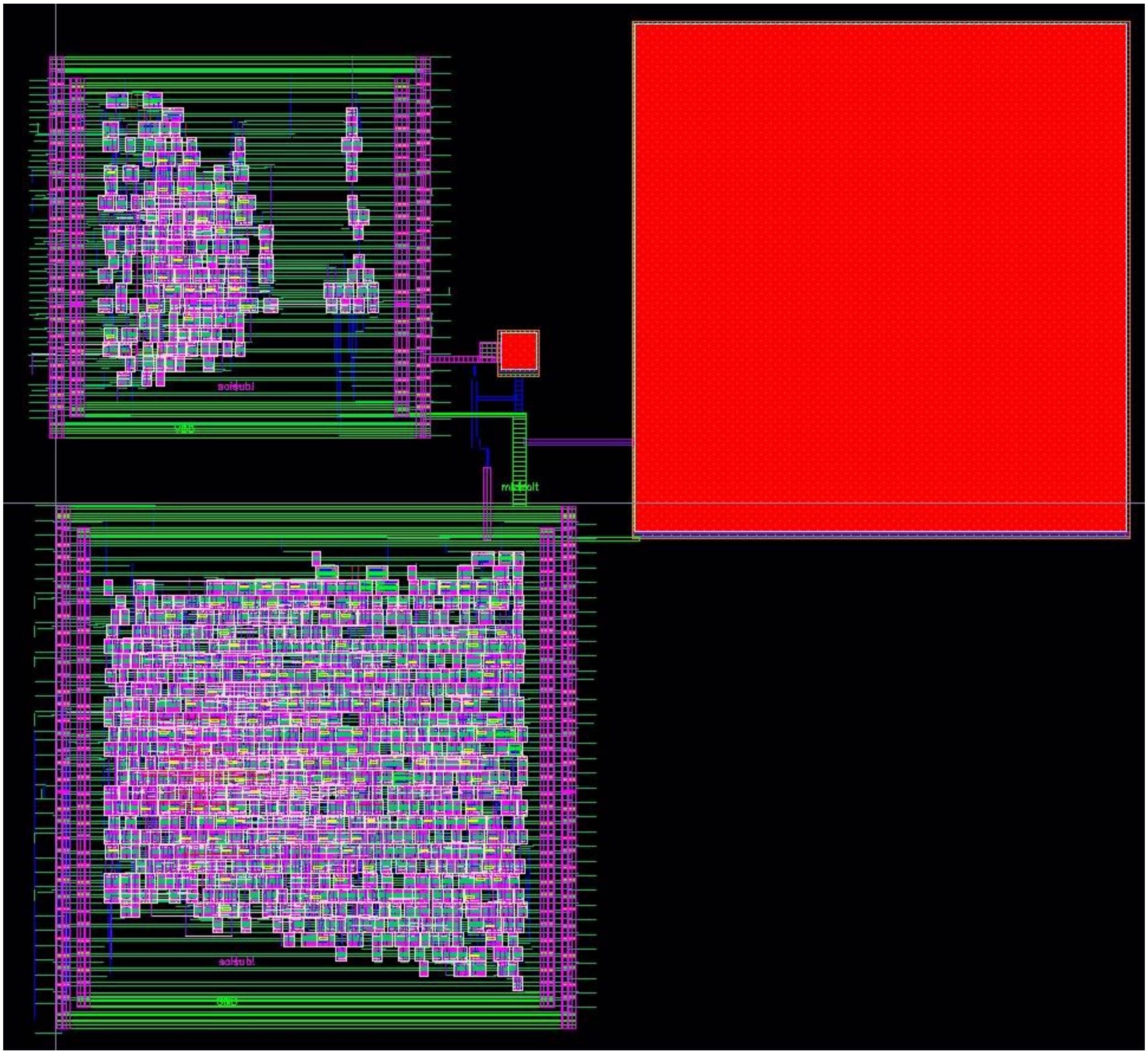

Figure 42: LVS-Clean RCA Row 1 and Multiplier Row 2 Design with Capacitors and Additional Logic in Cadence Virtuoso

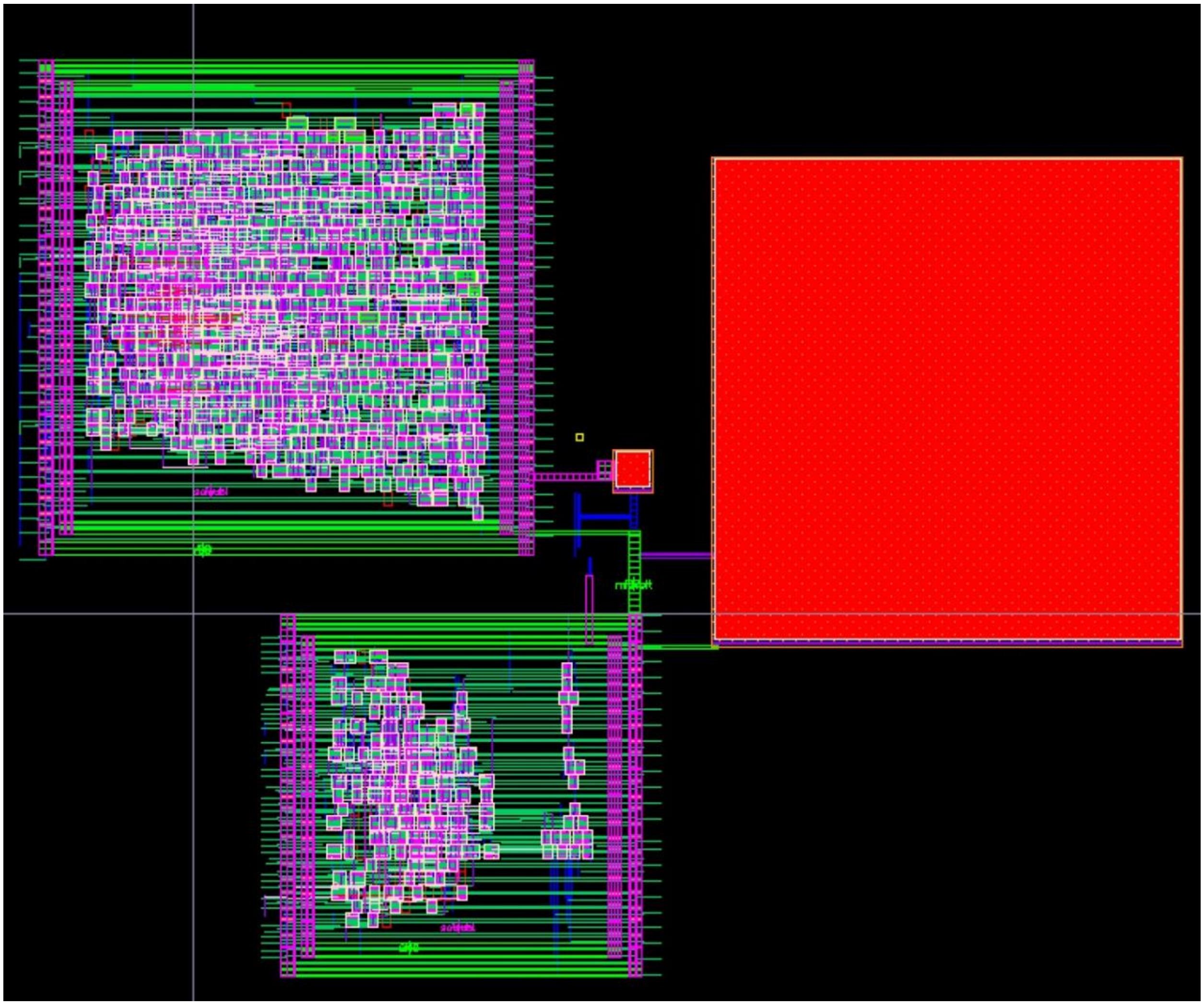

Figure 43: LVS-Clean Multiplier Row 1 and RCA Row 2 Design with Capacitors and Additional Logic in Cadence Virtuoso

Since the additional logic is hard to view in the Figures 42 and 43, Figure 44 shows two clearer images. Both the left and right screenshots are zoomed in versions of the four NFETs that comprise the Bypass and Awake Transistors to give a better idea of their locations and sizes compared with the overall designs. PEX is then performed on both designs, the multiplier on row 1 with the RCA on row 2 and the RCA on row 1 with the multiplier on row 2. A Spectre netlist is created with all of the parasitic information included, i.e., wire resistance and capacitance, coupling capacitance, etc. This Spectre netlist is used to simulate each of the designs with these physical attributes emulated in Cadence Analog Design Environment.

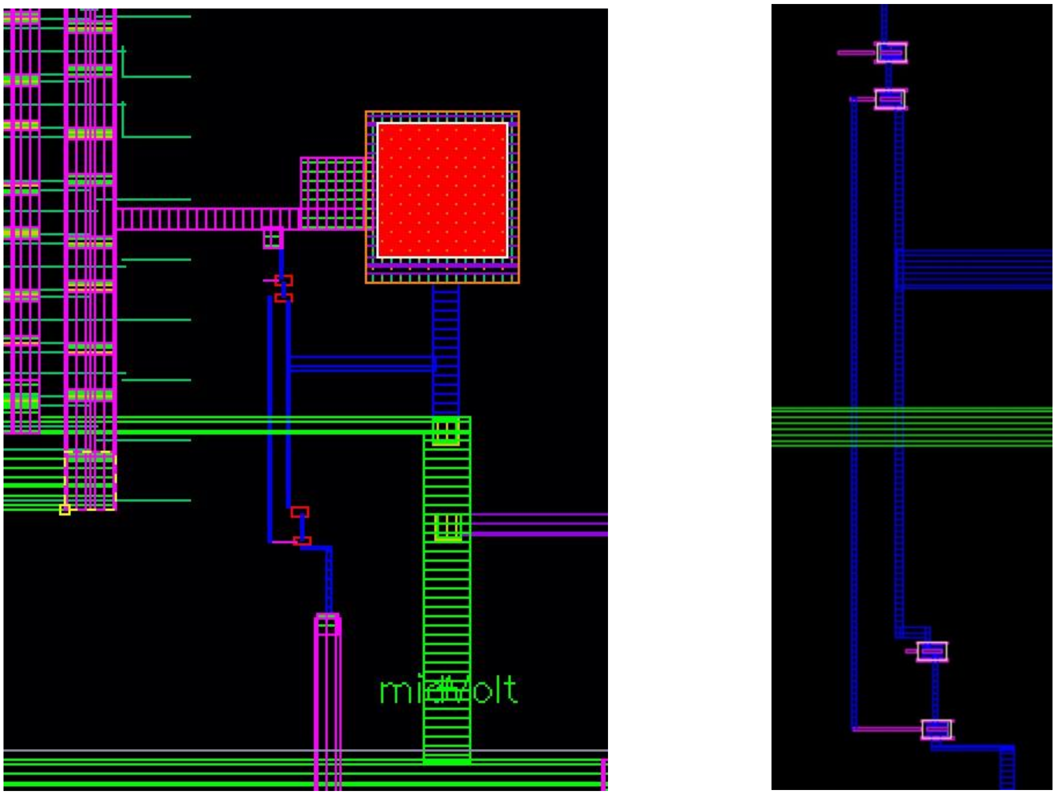

Figure 44: Bypass and Awake Transistors in Reference to Small Capacitor (Left) and Bypass and Awake Transistors (Right)

These simulations are performed on both of the post-PEX Double Stacked designs and then compared with the results from simulating the RCA and multiplier by themselves after they are individually PEXed. Although the trend is the same as the pre-PEX simulations, due to the added parasitic capacitances and resistances caused by wires and devices that PEX adds to the overall simulation, the numbers of data patterns that the RCA and the multiplier perform in the same amount of time are now different. That is, while in the 32nm process, the pre-PEX Double Stacked implementation performs 12 data patterns through the multiplier in the same amount of time that 40 data patterns pass through the RCA. Now that the design has gone through PEX, the multiplier performs 10 data patterns in the same amount of time the RCA performs 24 data patterns while in the Double Stacked model. Therefore, the data shown in Table 14 displays the execution time, energy consumption, and EDP for each of the Double Stacked designs when the multiplier performs 10 data patterns and the RCA performs 24 data patterns.

Table 14: Energy Delay Product Results from Stacked and Unstacked RCA and Multiplier in the 32nm Process

Circuit Setup	Execution Time (ns)	Energy Consumption (pJ)	Energy Delay Product (aJ*s)
Individual Multiplier	27.3	31.03	0.8472
Individual RCA	44.1	15.4	0.6792
Total Unstacked	N/A	N/A	1.5264
Both Running Stacked Multiplier Row 1-RCA Row 2	33.73	47.41	1.599
Both Running Stacked RCA Row 1-Multiplier Row 2	34.6	47.49	1.643

Table 14 clearly shows that the EDP increase from the post-PEX Double Stacked implementation is still quite small (~4.5% to ~7%), but the execution times and energy vary based upon setup. There is an immediate benefit to area and design time introduced by reducing the number of converters and regulators by stacking the two circuits. There is also minimal loss that occurs in performance when doing so. Therefore, the Bypass and Awake Transistors' sizes can be manipulated in order to see the effect they have on the post-PEX simulations.

Table 15 is comprised of simulations from the two post-PEX designs while the circuit on row 1 is active and the circuit on row 2 is sleeping. The transistor widths on row 2 are changed to show the trend in execution time, energy consumption and overall EDP. Figures 45 and 46 show the trend of the EDP when the RCA is active on row 1 and the multiplier is active on row 1, respectively. When analyzed and compared to the pre-PEX simulation results, the numbers will obviously be larger due to the added parasitics, but the trend of the EDPs for the RCA and multiplier are about the same. Since the multiplier is roughly four times the size of the RCA, the accumulative current flow through the multiplier is larger. Thus, the Bypass and Awake Transistors parallel to the RCA need to be sized larger to allow more current to flow through, which corresponds to an increase in the dynamic range of the multiplier while it is active.

Table 15: Data from Circuits Active on Row 1 While Circuit on Row 2 Sleeps in the 32nm Process

Circuit Operating on Row 1	Row 2's Transistors' Widths (nm)	Execution Time (ns)	Energy Consumption (pJ)	Energy Delayed Product (aJ·s)
RCA	104	63.55	26.16	1.662
RCA	312	46.6	31.1	1.45
RCA	624	38.55	38.05	1.47
RCA	936	35.2	45.11	1.588
RCA	1248	33.4	52.36	1.75
Multiplier	104	71.1	36.71	2.61
Multiplier	312	54.9	38.76	2.128
Multiplier	624	42.2	41.84	1.766
Multiplier	936	35	45.16	1.581
Multiplier	1248	30.87	48.54	1.498
Multiplier	1872	27.6	31.5	1.565

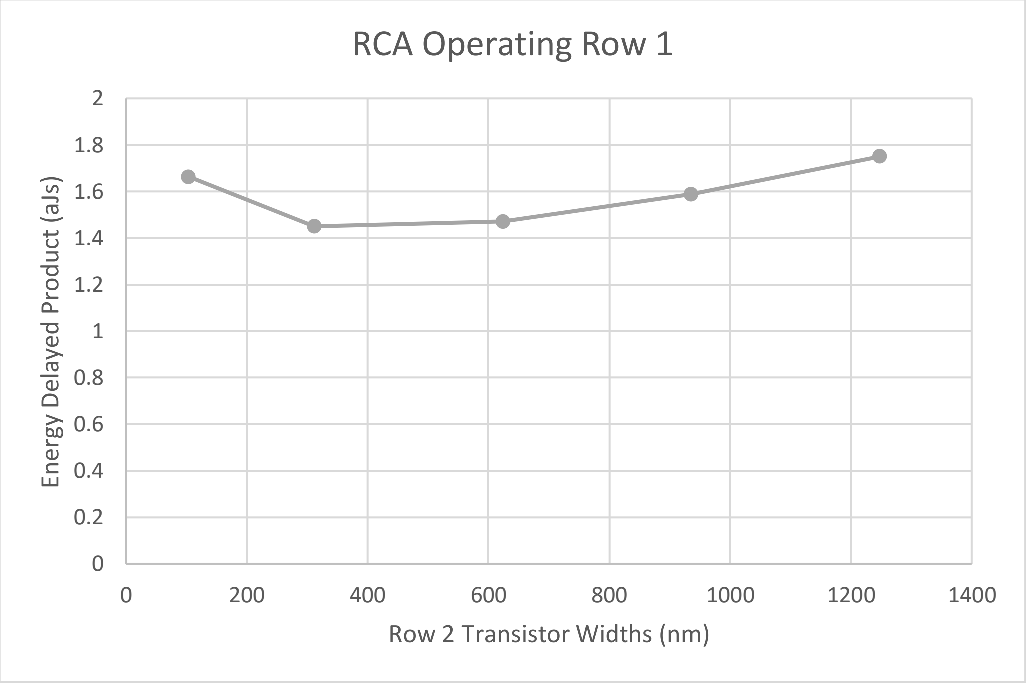

Figure 45: EDP Curve for Post-PEX RCA Operating on Row 1 in the 32nm Process

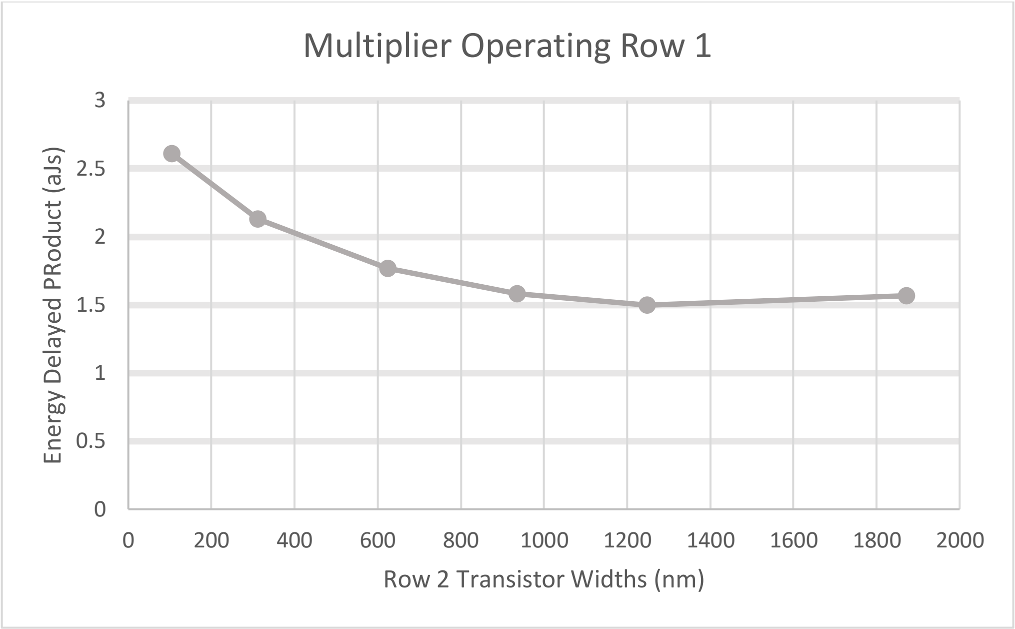

Figure 46: EDP Curve for Post-PEX Multiplier Operating on Row 1 in the 32nm Process

To see how the dynamic range and speed of the RCA circuit operating on row 1 alters when the Bypass and Awake Transistors are sized differently on row 2, Figure 47 shows it operating on row 1 with the transistors' widths sized at 104 nm (left) and 936 nm (right).

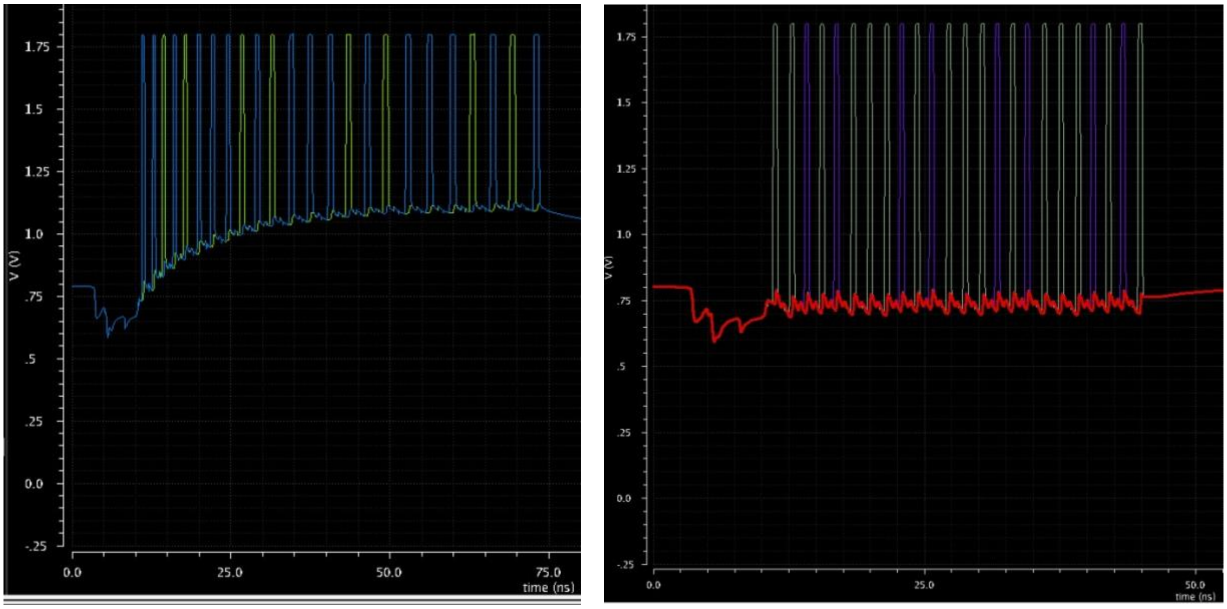

Figure 47: Post-PEX RCA Operating on Row 1 with Bypass and Awake Transistors' Widths Sized on Row 2 at 104 nm (left) and 936 nm (right)

The waveforms in Figure 47, which only display the middle node and the same single dual-rail output, show that while both simulations run correctly, the simulation on the left has a smaller dynamic range and slower execution time than the one on the right. This is confirmed by the execution times from Table 15, which are 63.55 ns (simulation begins at 10 ns) when the transistors' widths are 104 nm, and 35.2 ns when the transistors' widths are 936 nm.

Table 16 is populated with the simulations from the two post-PEX designs while the circuit on row 2 is active and the circuit on row 1 is sleeping. The transistor widths on row 1 are changed to show the trend in execution time, energy consumption and overall EDP.

Table 16: Data from Circuits Active on Row 2 While Circuits on Row 1 Sleeps in the 32nm Process

Circuit Operating on Row 2	Row 1's Transistors' Widths (nm)	Execution Time (ns)	Energy Consumption (pJ)	Energy Delayed Product (aJ·s)
Multiplier	104	67.9	28.77	1.954
Multiplier	312	46.6	33.63	1.567
Multiplier	624	35.2	39.9	1.404
Multiplier	936	30.44	45.05	1.371
Multiplier	1248	28.15	49.48	1.393
Multiplier	1872	27.15	58.14	1.579
RCA	104	87.6	10.41	0.9119
RCA	312	84.9	10.74	0.9118
RCA	624	82.5	11.04	0.9108
RCA	936	80.7	11.3	0.912
RCA	1248	79.3	11.55	0.9155

Figures 48 and 49 show the trend of the EDP when the multiplier and the RCA are active on row 2, respectively. The same patterns emerge when the circuits operating on row 2 are analyzed and compared to the pre-PEX simulation results, i.e., the energy consumption increases as the execution time gets shorter.

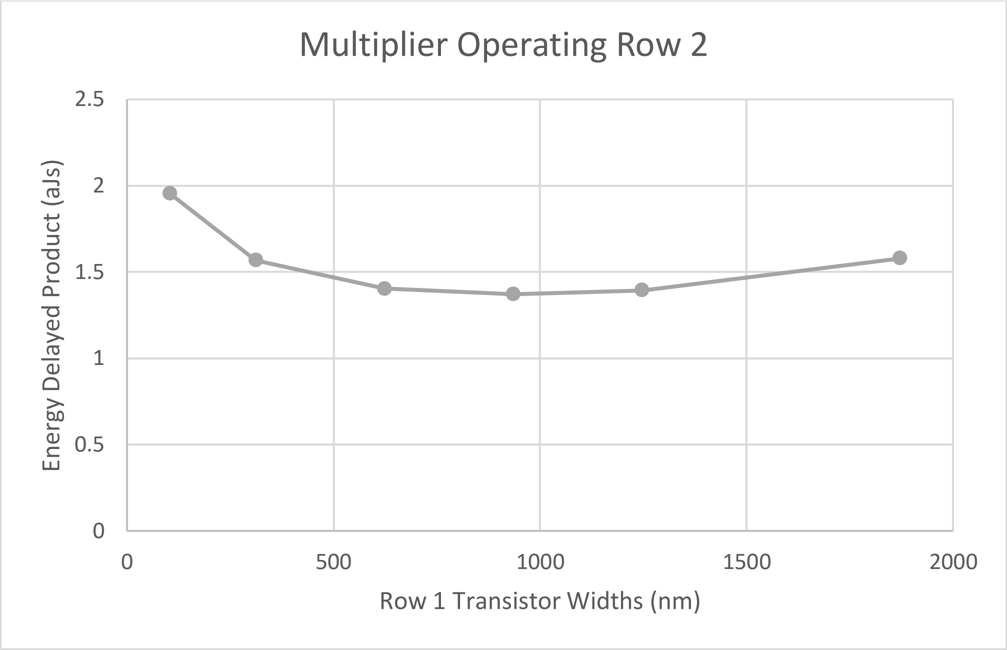

Figure 48: EDP Curve for Post-PEX Multiplier Operating on Row 2 in the 32nm Process

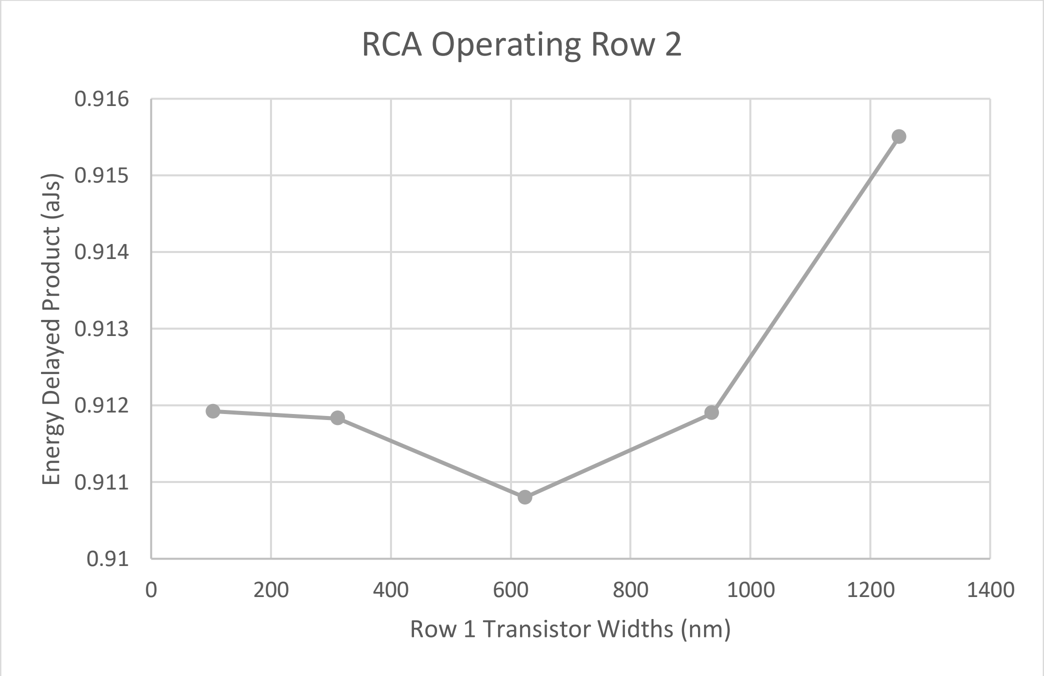

Figure 49: EDP Curve for Post-PEX RCA Operating on Row 2 in the 32nm Process

To see how the dynamic range and speed of the multiplier circuit operating on row 2 is affected when the Bypass and Awake Transistors are sized differently on row 1, Figure 50 shows it operating on row 2 with the transistors' widths on row 1 sized at 312 nm and Figure 51 shows it operating on row 2 with the transistors' widths on row 1 sized at 1,872 nm.

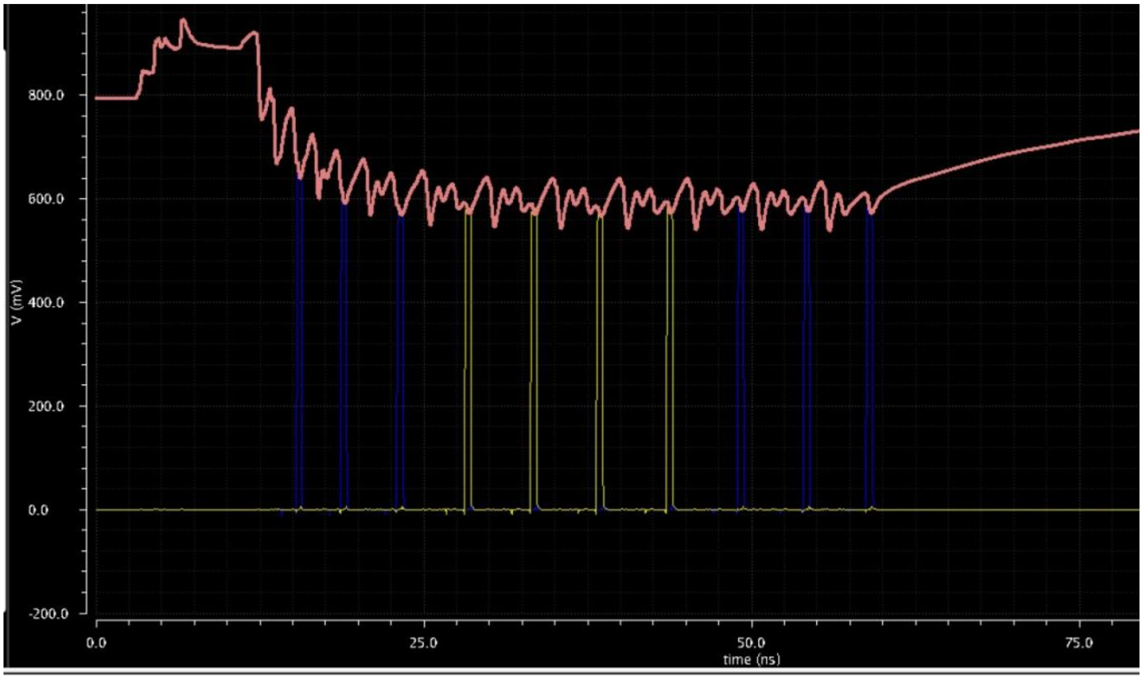

Figure 50: Post-PEX Multiplier Operating on Row 2 with Bypass and Awake Transistors' Widths Sized on Row 1 at 312 nm

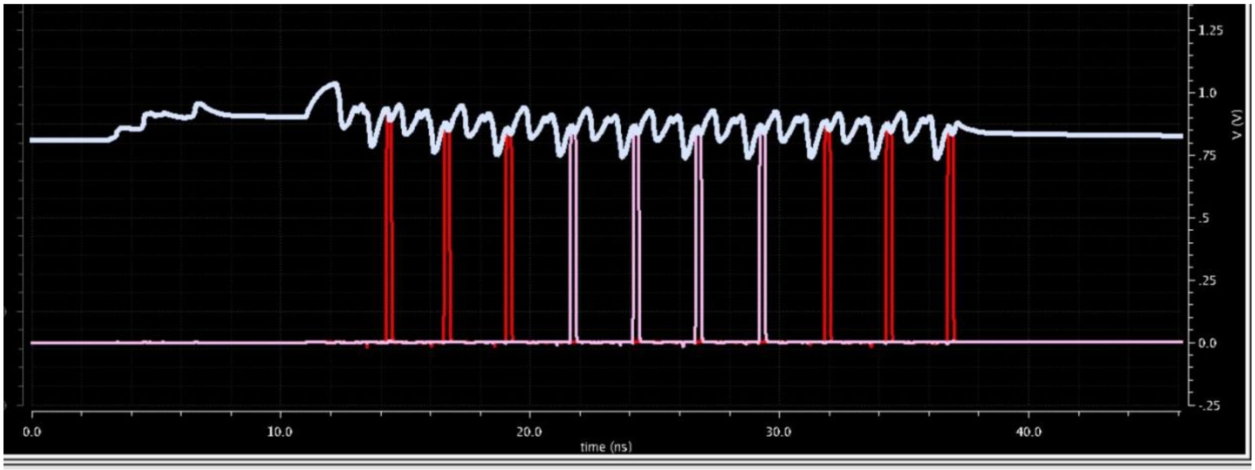

Figure 51: Post-PEX Multiplier Operating on Row 2 with Bypass and Awake Transistors' Widths Sized on Row 1 at 1,872 nm

The waveforms in Figures 50 and 51, which only display the middle node and the same single dual rail output, show that while both simulations run correctly, the one with the smaller sized Bypass and Awake Transistors has a smaller dynamic range and slower execution time than the other. This is confirmed by the execution times from Table 16, which are 46.6 ns (simulation begins at 10 ns) when the transistors' widths are 312 nm, and 27.15 ns when the transistors' widths are 1,872 nm.

In addition to demonstrating the physical design feasibility of the MTNCL circuit stacking architecture, the overall analysis shows that simulating the MTNCL Double Stacked architecture after implementing PEX follows the same trends that the schematic simulation models do. That is, when running either the multiplier stacked on top of the RCA or vice versa, the middle node will fluctuate towards the larger circuit. In addition, when putting either of the circuits to sleep for an extended period of time, the middle node, which would normally shift drastically towards the active circuit, can be manipulated using the additional logic. This ensures that the dynamic range of the active circuit remains at an acceptable level based upon the energy and performance requirements of the overall system.

6 Conclusion

In this dissertation research, a stacked architecture for MTNCL circuits is developed and verified in both schematic and post-PEX simulations. The energy consumption and speed of the circuits in the MTNCL Stacked architecture are comparable to them running individually. The stacked model also immediately reduces chip area as well as the overall energy and design complexity of the host system by removing extra DC-DC converters. Unlike the synchronous stacked architecture, using the asynchronous MTNCL paradigm allows different combinations of circuits to be stacked running different workloads with different operating cycles, while maintaining reliable operations without external adjustments. Moreover, the MTNCL Stacked architecture has the potential to be incorporated into mixed-signal systems to raise the supply voltage of digital components to match or nearly match the supply voltages of analog/RF components. This would simplify the overall power management system design and save energy from the power sources, e.g., batteries.

In order to stabilize the middle node voltage if one or more circuits are slept, the additional logic incorporated into the stacked architecture provides the circuit designer with the flexibility to optimize the design based upon their individual needs. There is an optimal EDP that the circuit designer can find based upon various design parameters, such as circuit size, number of inputs/outputs, process node, operating constraints (active time vs. idle time), etc. Or the designer can simply choose to implement a design that operates faster or saves more energy while operating.

One thing to note is that the additional logic, which is comprised solely of NFETs, needs to be evaluated during design phase to see the amount of current that flows through the transistors, in order to size the transistors properly. As the transistors' widths increase, so does the current flow through them when they are turned on to act as the bypass route around the idle circuit. In the 32nm process, the current flowing through the transistors of minimum size (i.e., 104 nm), while the RCA or multiplier is operating, is roughly 80 to 110 µA. The current flowing through the transistors of 936 nm in width, while the RCA or multiplier is operating, is roughly 490 to 640 µA. When the transistors are sized to 1,872 nm and the multiplier is active on either row, the current flowing through the transistors is 1 to 1.2 mA. Although the simulations completed correctly, in the event that a device is not able to handle the current needed for a larger circuit, multiple NFETs would be placed in parallel with one another and tied to the same signal in order to share the current.

Future work includes verifying the Double Stacked and Tripled Stacked models in other process nodes, as well as incorporating larger stacked models. In addition, although the Awake Transistors prevent any supplementary current leakage to occur through the Bypass Transistors when they are both on, power gating techniques can be incorporated to minimize the current leakage through the actual MTNCL circuitry.

7 Reference

[1] Jeff Falin. “Powering today’s multi-rail FPGAs and DSPs, Part 1,” *Analog Applications Journal* (1Q 2006).

[2] D. Kilani, B. Mohammad, H. Saleh and M. Ismail, "LDO regulator versus switched inductor DC-DC converter," *2014 21st IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, Marseille, 2014, pp. 638-641.

[3] W. Kim, D. M. Brooks and G. Y. Wei, "A fully-integrated 3-level DC/DC converter for nanosecond-scale DVS with fast shunt regulation," *2011 IEEE International Solid-State Circuits Conference*, San Francisco, CA, 2011, pp. 268-270.

[4] Sae Kyu Lee, David Brooks, and Gu-Yeon Wei. 2012. Evaluation of voltage stacking for near-threshold multicore computing. In *Proceedings of the 2012 ACM/IEEE international symposium on Low power electronics and design* (ISLPED '12). ACM, New York, NY, USA, 373-378.

[5] Jeff Falin. “Powering today’s multi-rail FPGAs and DSPs, Part 2,” *Analog Applications Journal* (2Q 2006).

[6] Heungjun Jeon, Fully integrated on-chip switched capacitor DC-DC converters for battery-powered mixed-signal SoCs (Doctoral Dissertation). Northeastern University, Boston, Massachusetts, 2012. http://hdl.handle.net/2047/d20002887. 8/15/2017.

[7] J. Di and S. Smith, Designing Asynchronous Circuits using NULL Convention Logic (NCL). Morgan & Claypool Publishers, (2009).

[8] Martin, Alain J. “The limitations to delay-insensitivity in asynchronous circuits.” Springer New York, 1990.

[9] Nowick, Steven M., and Charles W. O'Donnell. "On the existence of hazard-free multi-level logic." In *Asynchronous Circuits and Systems, 2003. Proceedings. Ninth International Symposium on*, pp. 109-120. IEEE, 2003.

[10] Karl M. Fant and Scott A. Brantd. “NULL Convention Logic: a complete and consistent logic for asynchronous digital circuit synthesis,” in ASAP 96, Chicago, IL, Aug. 1996.

[11] Liang Zhou, Ravi Parameswaran, Farhad A. Parsan, Scott C. Smith and Jia Di. “Multi-Threshold NULL Convention Logic (MTNCL): An Ultra-Low Power Asynchronous Circuit Design Methodology,” *Journal of Low Power Electronics*, 2015, pp. 81-100.

[12] L. Men and J. Di, "An asynchronous finite impulse response filter design for Digital Signal Processing circuit," *2014 IEEE 57th International Midwest Symposium on Circuits and Systems (MWSCAS)*, College Station, TX, 2014, pp. 25-28.