Energy Reduction for Asynchronous Circuits in SoC Applications

Harish Gopalakrishnan

Wright State University

Follow this and additional works at: https://corescholar.libraries.wright.edu/etd_all

Part of the Engineering Commons

Repository Citation

https://corescholar.libraries.wright.edu/etd_all/530

This Dissertation is brought to you for free and open access by the Theses and Dissertations at CORE Scholar. It has been accepted for inclusion in Browse all Theses and Dissertations by an authorized administrator of CORE Scholar. For more information, please contact library-corescholar@wright.edu.
ENERGY REDUCTION FOR ASYNCHRONOUS CIRCUITS IN SoC APPLICATIONS

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

By

Harish Gopalakrishnan
M. S., Wright State University, 2005

2011
Wright State University
I HEREBY RECOMMEND THAT THE DISSERTATION PREPARED UNDER
MY SUPERVISION BY Harish Gopalakrishnan ENTITLED Energy Reduction
for Asynchronous Circuits in SoC Applications BE ACCEPTED IN
PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE
OF Doctor of Philosophy.

J. M. Emmert, Ph.D.
Dissertation Director

Ramana V. Grandhi, Ph.D.
Director, Ph.D. in Engineering Program

Andrew Hsu, Ph.D.
Dean, Graduate School

Committee on Final Examination

J. M. Emmert, Ph.D.

Chien-In Henry Chen, Ph.D.

Gregory Creech, Ph.D.

Raymond E. Siferd, Ph.D.

Ranga Vemuri, Ph.D.
As complexity increases and gate sizes shrink for monolithic, mixed-signal integrated circuit (IC) technologies, two problems become dominant: substrate noise caused by digital clocks interfering with highly sensitive analog and radio frequency (RF) components and parametric variations that can cause circuit delays to vary in excess of 35%. Clockless logic (or asynchronous) circuits address both of these issues and more. Clockless, asynchronous circuits are by nature delay-insensitive making them immune to parametric variations. Even more important is the processing characteristics of clockless asynchronous circuits, which eliminate highly intricate clock signals that cause large power spikes every time they switch. Consequently, asynchronous design is becoming more and more attractive for low-noise, low-power applications.

In a clock free environment, energy is a more relevant metric than power. In this work, we present algorithms that attempt to minimize the energy in asynchronous integrated circuits. Our techniques are based on voltage scaling (VS) and gate sizing (GS). On average, performing a two-stage energy reduction with VS followed by GS results in 26% energy reduction.
## Contents

1 Introduction  
1.1 Motivation ......................................................... 1  
1.2 Objective .......................................................... 6  
1.3 Thesis Contributions ............................................. 6  
1.4 Thesis Organization .............................................. 7  

2 Background and Related work  
2.1 Introduction ....................................................... 9  
2.1.1 Energy Model for the Current Research .................... 10  
2.2 Related Work ....................................................... 11  
2.2.1 Previous Dynamic Energy reduction techniques ............ 11  
2.2.2 Previous Static Energy Reduction Techniques .............. 13  
2.3 Summary ............................................................ 18  

3 Introduction to Asynchronous Systems  
3.1 Introduction to NULL convention logic (NCL) .................. 20  
3.2 Communication protocols in Asynchronous Systems .......... 21  
3.3 Asynchronous design - Delay Models ............................ 24  
3.4 Data Representation or Encoding in Asynchronous Circuits .... 28  
3.5 NCL logic gates .................................................... 31  
3.6 CMOS Realization of NCL gate ................................. 35  
3.7 NCL system ........................................................ 37  
3.8 Advantages of NCL for asynchronous design .................. 37  

4 Voltage Scaling for Energy Reduction  
4.1 Introduction ......................................................... 40  
4.2 Static Voltage Scaling Design without LC ..................... 43  

iv
List of Figures

1.1 Substrate noise in (a) time domain, (b) frequency domain in a MS-environment [4].................................................. 2

1.2 $V_{dd}$ bounce in dual voltage design in MS-environment. ................. 2

2.3 General MTCMOS Architecture [93].................................................. 15

2.4 General MTCMOS-NCL Architecture [93].......................................... 16

2.5 General Architecture of Multiplexed Inputs SCCMOS (MISCCMOS) scheme [95].................................................. 18

3.6 Handshaking protocol in asynchronous systems ................................. 22

3.7 Types of Handshaking protocol (a) two-phase protocol, (b) four phase protocol .................................................. 22

3.8 Comparison of Circuit design based on delay models .......................... 27

3.9 Data Encoding Schemes .................................................................. 28

3.10 A discrete threshold gate ................................................................. 30

3.11 A discrete threshold gate with feedback ............................................ 31

3.12 NCL operator, THmn gate ............................................................... 32

3.13 (a) TH23W2 NCL operator, (b) TH54W322 NCL operator ............... 33

3.14 NCL logic gate architecture. ............................................................ 36

3.15 Transistor topology of (a) Static TH22 gate; (b) Semi-static TH22 gate 37

3.16 Basic NCL pipeline ......................................................................... 38

4.17 DVS environment: LV cell driven by HV cell. ................................. 41

4.18 Voltage Scaling by CVS. ................................................................. 43

4.19 Slack distribution for BM i8 ............................................................ 46

4.20 Voltage Scaling by ECVS. ............................................................... 46

4.21 Switching energy and timing plot of static-TH22 gate. ................ 51

4.22 Comparison of gate-LC, LC-gate and Emb-LC. ............................... 55
4.44 Comparison of percentage of LV cells after CVS, ECVS and GECVS for DIMS Architecture. .................................................. 87
4.45 Comparison of percentage of LV and LC cells after ECVS and GECVS for DIMS Architecture. .................................................. 87
4.46 Comparison of percentage of energy reduction after CVS, ECVS and GECVS for DIMS Architecture. .................................................. 88
4.47 Power and Energy plot for BM i8 before and after applying GECVS. .................................................. 88
4.48 Comparison of percentage of Energy Reduction - HSPICE Vs. Analytical Reduction (NCL-opt) style). .................................................. 89
4.49 Comparison of percentage of Energy Reduction - HSPICE Vs. Analytical Reduction (DIMS) style). .................................................. 90
5.50 GS-CVS cells on the Backward Front .................................................. 92
5.51 Cost Function plot for BM circuit t481 employing GS technique - GS-CVS. .................................................. 94
5.52 Cost Function plot for BM circuit c1908 employing VS technique - CVS. .................................................. 94
5.53 Cost Function plot for BM circuit c1908 employing GS technique - GS-CVS. .................................................. 95
5.54 Cost Function plot for BM circuit c499 employing GS technique - GS-CVS. .................................................. 95
5.55 GS-GECVS cell selection criteria. .................................................. 97
5.56 Cost Function plot for BM circuit ttt2 employing GS technique - GS-GECVS. .................................................. 98
5.57 Comparison of percentage of LV cells after GS-CVS and GECVS for NCL-opt Architecture. .................................................. 101
5.58 Comparison of percentage of Energy Reduction after GS-CVS and GECVS for NCL-opt Architecture. .................................................. 101
5.59 Comparison of percentage of LV cells after GS-CVS and GECVS for DIMS Architecture. .................................................. 102
5.60 Comparison of percentage of Energy Reduction after GS-CVS and GECVS for DIMS Architecture. .................................................. 102
5.61 Comparison of percentage of energy reduction after CVS+GS-CVS and GECVS+GS-GECVS for Ncl-opt Architecture. ...................... 106
5.62 Comparison of percentage of energy reduction after CVS+GS-CVS and GECVS+GS-GECVS for DIMS Architecture. ...................... 107
5.63 Energy savings from GS-CVS on BM i8 (DIMS) plotted from HSPICE. 108
5.64 Comparison of power consumed in BM circuit i8 (DIMS) for one DATA-cycle. .............................................................. 108
5.65 Comparison of percentage of LV cells after CVS+GS-CVS and GECVS+GS-GECVS for Ncl-opt Architecture. ...................... 109
5.66 Comparison of percentage of LV cells after CVS+GS-CVS and GECVS+GS-GECVS for DIMS Architecture. ...................... 110
5.67 Power plot of BM circuit i8 before and after GS-GECVS. ............ 110
5.68 Energy plot of BM circuit i8 before and after GS-GECVS. ............ 111
List of Tables

3.1 Data representation in dual rail encoding (NCL) [46, 47] .................. 29
3.2 Data representation in Quad rail encoding (NCL) [47, 46] ............... 30
4.3 Transistor overhead of gate-LC combination compared with Emb-LC. 60
4.4 NCL library transistor count for static and semi-static version [47] .... 73
4.5 General layout rules used by silicon generator for creating a NCL li-
   brary in 130nm process. ............................................. 73
4.6 Experimental Results: Voltage Scaling with CVS, ECVS and GECVS
   on NCL-opt Architecture. ............................................ 83
4.7 Experimental Results: Voltage Scaling with CVS, ECVS and GECVS
   on DIMS Architecture. .............................................. 83
5.8 Experimental Results: Gate Sizing with GS-CVS and GS-GECVS on
   NCL-opt Architecture. .............................................. 100
5.9 Experimental Results: Gate Sizing with GS-CVS and GS-GECVS on
   DIMS Architecture. ................................................ 100
5.10 Number of logic levels and number of levels of LV cells from PO after
    CVS. ................................................................. 100
5.11 Experimental Results: Voltage Scaling and Gate Sizing with CVS+GS-
    CVS and GECVS+GS-GECVS on NCL-opt Architecture. .......... 105
5.12 Experimental Results: Voltage Scaling and Gate Sizing with CVS+GS-
    CVS and GECVS+GS-GECVS on DIMS Architecture. ............ 105
Acknowledgement

As the journey of this dissertation work comes to an end, I am grateful and owe not just thanks but much more to many people who have shared my best and worst moments over several years.

First and foremost I thank god for keeping me hale and healthy.

I am ever grateful to have such a great advisor and mentor who has supported me all through these years. This work wouldn’t have been possible without the direction provided by my advisor Dr. J. M. Emmert. It is not often that one comes across a guru who gives the freedom to pursue and explore on your own and at the same time carefully guide you when you need him the most. His patience and assistance has helped me tide over many unavoidable obstacles during this research and complete this dissertation work on-time.

Thank you Dr. Emmert for the financial support and also providing the foundation for this exciting research topic - asynchronous circuits. You have not just taught me how to be good researcher but much more. Your time management and art of teaching is something that I admire and wish I could emulate it someday.

My thanks also go to Dr. Siferd, Dr. Chen, Dr. Creech and Dr. Vemuri for serving on the dissertation committee, guiding and providing valuable comments on this research which helped to shape this work better.

A very special thanks to the former member of Design Automation Lab - Tom Pemberton, Sowri, Mohammad, Vipul and Run. Dr. Run thank you for your encouragement and support. I’ll never forgot our conversation on wide range of interesting topics.

Graduate school days would have been really mundane, monotonous days without my friends - Lax, Zee, Shams, Mohsen, Diana, Anna, Yvette and Nadine. Thank you for the good times I had. Your friendship is much appreciated and thank you all for everything. A shout-out to my tea-club: Nisha, Andrew, Bharath, Vedu
and Dakshin. Ah, those conservations over tea/coffee are still vivid and forever will cherish it. Thank you guys, without your support and encouragement it would have been hard to get by.

I also express my gratitude to the staff at the Electrical Engineering department - Vickie and Marie. I'm also indebted to CECS computer administrator - Mike VanHorn. Mike thank you so much for justin-time processing of my computer related issues.

Most importantly, this research wouldn't have been possible without the unconditional love, constant support, unending encouragement, strength and blessings of my family. Amma and Appa I owe you a lot. Dada, Di-amma, Bhuvani-amma, Ums and Anu this is for you. I express my heart-felt gratitude to you all and I'm always indebted to you all. Anu I don't know if could have survived without your unwavering love and support. Patti and thatha I miss you everyday and I seek your blessings as I explore future endeavours. I would also acknowledge my dear brother - Vinod. Vinny, you inspire me everyday and help me recover from those low days. Stay strong Vinny! Love to all my other dear brothers.

And finally, I also thank my wife - Gayathri. She has always been there for me when I needed her and thank you for your unyielding patience, encouragement, support, care and undeniable love. Thank you for understanding and tolerating me through my occasional mood swings as I come to the end of this incredible journey.
To Amma, Appa

Patti, Thatha

&

my Guru
1 Introduction

1.1 Motivation

Abiding to the Moores’ law, today’s microelectronics has migrated to integrating a complete system on a single chip (SoC) with both analog and digital components mapped to a common substrate. Despite its advantages, arguably, the single most important issue plaguing the Mixed Signal (MS) designers is the Substrate Noise (SN). There are numerous sources of noise on MS SOCs including: capacitive coupling of MOSFETs and BJTs, signal interconnects, power supply fluctuations (power/ground bounce [fig. 1.2 shows a typical $V_{dd}$ bounce in dual voltage environment.]), and many others [4]. SN which is primarily produced by the high speed switching digital circuits and high power RF components affect the sensitive analog components through parasitic capacitive coupling of substrate. Fig. 1.1 (a) shows the substrate noise in time domain and fig. 1.1 (b) in frequency domain in a MS-environment. The pictures depicts the sensitivity of analog parts to substrate noise.

The victims (analog parts) experience delay variations, threshold voltage ($V_{TH}$) modulations, clock jitters and skews [35]. Consequently, the signal integrity and system performance becomes unreliable. MS SoC designers thus, can no longer neglect the previously ignored issues like substrate noise from design factors affecting its performance. This noise consideration thereby, considerably increases design complexity.

A number of techniques to tackle the SN in MS-SoC have been proposed over the years [38, 37, 40, 41, 39]. They can be categorized into noise mitigation, noise cancellation and noise tolerance approaches. Most of the noise mitigation techniques are employed at the physical layout (silicon level) design level building trenches (deep or shallow), guard rings around the sensitive analog components, moats and Silicon-on-Insulator have all been explored [37, 40, 41] and proven to be powerful noise mitigation techniques. Albeit, these techniques exponentially increases the silicon
Figure 1.1: Substrate noise in (a) time domain, (b) frequency domain in a MS-environment [4].

Figure 1.2: $V_{dd}$ bounce in dual voltage design in MS-environment.
design complexity [48].

At high level, when modeling SN is very complicated and unreliable, noise tolerance techniques include asynchronous circuits which have shown great prospects in the past to mitigate the SN [76, 77]. We classify these techniques that fall in the later category as active SN tolerance techniques [4] that can be used in high frequency RF design where it significantly improves the sensitivity of analog components. This dissertation research evolved from this key idea of using asynchronous circuits for noise mitigation in MS-SoC and other advantages such as low power that is inherent in asynchronous circuits is the focus of this research work.

Besides noise, power is arguably the biggest issue in clocked design. In the age of hand held portable devices, the low power and longer battery life are the new focuses even as logic complexity increases manifolds. Cooling aids for high density chips is proving to be an expensive affair which sometimes limits the functionality that can be added on the chip [36]. Invariably, this has taken the low power research in new direction with power reduction and power optimization gaining significant attention with ever shrinking power budgets.

Global clocks and buffer trees undoubtedly are the major source of power consumption in synchronous clocked design. Frequent clock switching accounts for dynamic power consumption in these designs. Peak current increases when the entire clock tree distribution, flip-flops and the their fan-out circuitry switches in synchronism with the clock transitions in clocked logic circuit contributing to significant increase in peak power in the circuit. The magnitude of peak current drawn in the circuit is proportional to the substrate noise generated in the circuit which can affect the performance of the analog components in MS circuits [44]. De-facto, synchronous clocked logic design is becoming increasingly difficult to distribute and manage the clock; the focus is now on the alternate more appealing design methodology - clockless logic or the asynchronous design methodology.
Asynchronous systems not only overcomes the drawbacks related to clock issues, but presents numerous other advantages compared to synchronous logic such as low noise by reducing the cross-talk, low emission of electromagnetic noise (EMI) [45] and including SN reduction in MS environment [76, 77]. Asynchronous systems can operate under variable frequency and the different blocks can start their next task upon completion of the current task, unlike the synchronous systems, which has to wait for clock signals to latch their next data. Furthermore, synchronous systems cannot operate at maximum specified frequency due to clock skew [47]. Thus, the operating speed of asynchronous system is determined by average case rather than the worst case that determines the speed of synchronous clocked systems.

In DSM design with synchronous logic, timing closure problems pose a serious challenge to the designers along with its voltage and temperature variations [50, 51]. It has also been projected that the parametric variations in synchronous circuits may reach as much as 35% by 2020 [52, 48].

Asynchronous technology which has shown its robustness towards temperature, voltage and process or parametric variations has also been a motivating factor for steering the design interest towards clockless logic [45]. Asynchronous circuits switches when necessary, avoiding redundant switching which drastically reduces the power consumption in clockless logic circuits, unlike synchronous circuits which switches at every clock pulse. In fact, in clocked design, a considerable portion of power is dissipated by the components that are not computationally active at the present clock edge [53].

A NCL08 processor fabricated by Theseus Logic, is NULL convention Logic (NCL) asynchronous version of STAR08 clocked microprocessor which has shown to achieve 30% lower power and 6dB less noise than the clocked STAR08 [54]. Asynchronous MIPS R3000 microprocessors developed at the asynchronous VLSI group at Caltech have shown similar results [55].
On further inspection, asynchronous logic design has proved it is advantageous to design complex circuits that are extensively data dependent. A digital compact cassette (DCC) player developed at Philips Research Laboratories exhibits 80% less power than its synchronous counterpart [56]. As a consequence of its power advantages, asynchronous NCL version of design has also found an appropriate place in medical technology by implementing implantable medical devices [58].

Power is distributed over time and area which reduces the burden on peak power demand in these systems. This peak power reduction reduces the hot-spots and noise in the system enhancing reliability [47]. In addition, asynchronous systems have shown their tolerance towards supply voltage variations and known to operate reliably at low voltages [47].

Nonetheless, asynchronous designs have some drawbacks too. It is well-known that most of the clockless design have twice (dual-rail circuits) or more area overhead. Although this factor directly affects the speed of the circuit, they avert the problem of hazard and/or race conditions (orphans in NCL) in the circuitry [52]. Orphans are unwanted DATA transitions that are unobservable at the circuit’s output [59, 51, 2]. In addition, the completion detection circuit that form a part of the handshake mechanism (control circuit) in asynchronous circuits creates bottle-neck to circuit implementation in term of area, speed and power overhead [45]. Lack of CAD tools [15, 1] and tools for testing and test vector generation pose hurdles for this otherwise promising technology [1, 15, 45].

Though the asynchronous logic has been in the research community for a while, its general acceptance in the mainstream design has been stalled by the lack of CAD tools for design and the general myth that it is more difficult to design than it synchronous counterparts. But with fresh interest in this field due to lower noise advantages for MS circuits and development of new logic like the NULL convention logic (NCL) [59], and as reported by the International Technology Roadmap for Semiconductor
(ITRS), the shift is likely towards the asynchronous design [48].

Thus, we will investigate and implement novel generic methods to reduce energy in asynchronous circuits with a focus on asynchronous threshold logic type design which can be applied to other asynchronous design versions too. The proposed methods can be implemented in the MS environment when low power and low noise design are of primary interest.

1.2 Objective

Since asynchronous design is a promising field with its numerous advantages over its synchronous counterparts, this research focuses on harnessing the low power advantages of asynchronous design methodology. The design methodology of our interest will be the threshold network design for asynchronous design using NCL. Although, NCL leverage low power advantages, it is still immature requiring a formal method for energy reduction and optimization. As a result, the goal of this research work is to address the energy reduction issues in asynchronous circuit, analyse, formulate and implement the energy reduction techniques.

Lacking industry standard CAD tool for design, asynchronous circuits are difficult by design [1, 15] and have often been neglected despite its attractive characteristics of mitigating clock related issues. This motivates this dissertation to find a potential CAD tool support for asynchronous design. This issue has been addressed in this research work with a technology independent component library generator and design flow that closely resembles the existing synchronous ASIC design flow.

1.3 Thesis Contributions

Contributions of this research work are:

1. A complete design flow that closely resembles the synchronous design flow is presented. The commercially available off-the-shelf synchronous design tools are
used for synthesis, simulation and design translation to asynchronous design and improve the power/energy of circuit in the optimization process. This tackles the issue of lack of CAD tool support for asynchronous design and eliminates the need for dedicated asynchronous design CAD tool.

2. New energy optimization (specifically related to dynamic energy) methods specific for asynchronous circuits are presented here. The techniques for optimization have been adapted from synchronous logic design and tailored to suit asynchronous design methodologies. Widely accepted gate level energy reduction techniques such as combination of voltage scaling and gate sizing techniques are presented here.

3. A versatile yet sophisticated automated standard cell library generator is developed which is technology and platform independent. The cost-effective leaf cell generator offers good quality layouts along with the options a commercial tool would offer. Our tool instantly creates asynchronous standard cell library for the targeted technology along with the files required for post synthesis physical design layout.

4. Finally, this research work attempts to bridge the gap between asynchronous circuits and synchronous circuits and eventually move clockless design into mainstream. This research also aims for general acceptance of asynchronous circuit as a candidate for low power, low noise MS-SOC designs.

1.4 Thesis Organization

The dissertation is organised as follows. Chapter 2 offers a brief overview of existing energy/power reduction techniques. In Chapter 3, an introduction to asynchronous systems and focus of this research, a brief overview of asynchronous threshold network circuits - NULL Convention Logic (NCL) is provided. The voltage scaling techniques
along with it implementation and experimental results are detailed in Chapter 4. This chapter also provides information on the dual voltage scaling implementation issues at the layout level. A complete asynchronous design flow resembling the existing synchronous design flow is also presented in chapter 4. In chapter 5, gate sizing techniques implemented in this work are elaborated. The experimental details and discussions are also presented. Conclusion and future work are briefly discussed in Chapter 6.
2 Background and Related work

This chapter intends to provide a brief overview of energy metrics and review of previous energy optimization and reduction techniques intended for asynchronous circuits with the primary focus on threshold network circuits such as NULL convention logic (NCL).

2.1 Introduction

Besides speed, power dissipation is an important metric in assessing the performance of microelectronic circuits and it has gained prime importance in the ever-shrinking deep sub-micron (DSM) technologies. Power management has become crucial with the advent of portable devices and it has rapidly gained significant attention fueling a new trend in VLSI research with low power design and power/energy aware designs. To better understand the importance of power and its equivalent metric in asynchronous circuits – Energy, a quick review of power metrics are provided here.

The power dissipated in CMOS VLSI can be broken into three basic components [36]:

- Dynamic power
- Short circuit power
- Static power or Leakage power

\[ P_{avg} = P_{dynamic} + P_{short-circuit} + P_{leakage} \]  

(2.1)

where \( P_{avg} \) is the average power consumption in the circuit, \( P_{dynamic} \) is the dynamic power consumption in the circuit or the power consumed due to the switching of signals sometimes referred to as the switching power, \( P_{short-circuit} \) is the short circuit power consumed in the circuit due to the direct connection or path between the power
source \((V_{dd})\) and the sink (GND), \(P_{\text{leakage}}\) is the leakage power consumed in the circuit due to the leakage current.

### 2.1.1 Energy Model for the Current Research

For clockless asynchronous circuits, the energy term is a more meaningful metric in clock free environment than the power term \([26]\). Hence, (2.1) is reproduced as

\[
E_{\text{avg}} = E_{\text{dynamic}} + E_{\text{short\textendash}circuit} + E_{\text{leakage}}. \tag{2.2}
\]

where \(E_{\text{avg}}\) is the average energy in the circuit, \(E_{\text{dynamic}}\) is the dynamic energy, \(E_{\text{short\textendash}circuit}\) is the short-circuit energy and \(E_{\text{leakage}}\) is the leakage energy in the circuit.

In this work, for NCL, we estimate the dynamic energy consumption in the circuit. Static and short-circuit energy is ignored as they have insignificant values compared to switching energy \([34]\) (also refer fig. 4.21). Also, NCL are glitch free due to the monotonicity of signal transitions \([33]\). Therefore, the majority of the energy expended is the dynamic energy in this type of asynchronous circuits \([55]\). From (2.2) neglecting the \(E_{\text{short\textendash}circuit}\) and \(E_{\text{leakage}}\), the dynamic energy is given by \([85, 55]\)

\[
E_{\text{dynamic}} = \frac{1}{2} \times V_{dd}^2 \sum_{i=1}^{m} C_i n_i, \tag{2.3}
\]

where \(V_{dd}\) is the supply voltage, \(C_i\) is the total load capacitance at the output of \textit{gate}_i and \(n_i\) is the total number of switching on the \textit{gate}_i in the circuit.

NCL are a class of delay-insensitive asynchronous circuits \([59]\) employing dual-rail for data communication. An in-depth introduction to asynchronous circuits along with introduction to NCL type circuits are presented in chapter 3. In dual rail NCL circuits, every gate switches twice in a NULL-DATA-NULL cycle and hence, the number of switching, \(n_i\), (2.3) can be assumed as \(n_i = 2\). Therefore,

\[
E_{\text{dynamic}} = \frac{1}{2} \times V_{dd}^2 \sum_{i=1}^{m} C_i \times 2, \tag{2.4}
\]
Equation (2.5) simplifies the dynamic energy estimation and this is the energy model that has been used in this work. Equation (2.5) establishes the relationship between voltage and dynamic energy and it is also evident that reducing the voltage will have a profound effect on dynamic energy. To verify the effectiveness of lowering the voltage on NCL type circuits, voltage scaling techniques are presented in detail in chapter 4. The equation (2.5) also presents the relationship between load capacitance \((C_i)\) and dynamic energy. Load capacitance is directly related to the transistor width and hence, gate sizing is another widely accepted technique for low energy design. This technique is outlined along with implementation details in chapter 5.

2.2 Related Work

This section of the chapter is devoted to a brief discussion on previous energy reduction methods that are available up-to-date which are applicable to asynchronous circuits. This survey includes techniques that are specific to NCL type circuits and other asynchronous design methodologies.

Exploiting the LP options of the asynchronous design still requires energy reduction and optimization techniques that can be applied at various level of design abstraction. Majority of the techniques researched so far, target either dynamic energy reduction or the static energy. The following provides a brief insight on the various works currently available for asynchronous circuits.

2.2.1 Previous Dynamic Energy reduction techniques

The earliest of the power reduction technique in asynchronous systems targeting the self timed design was the adaptive power supply scaling employed in [56] and more recent one in [85]; while [56] targets the self timed design type, [85] targets the
threshold design type, specifically the NCL type circuits [59].

The rationale behind this adaptive scaling is that dynamic energy is dependent on supply voltage (square-law dependency) and scaling voltage reduces energy. But, this scaling comes at a price - speed is reduced, since the gate drive \((V_{gs} - V_{th})\) is reduced [73]. Hence, the supply voltage is adjusted automatically to the variable data rate by keeping the supply voltage at a minimum optimal level at all times to reduce the energy and at the same time satisfy the speed requirements of the circuit.

[56] employs FIFO buffers along with a state detection circuit to detect the data rate change and adjust the DC-DC buck converter for variable voltage. While kuang et al. uses handshake signal circuitry consisting of completion detection circuit, a D-FF and DC-DC converter to adjust the operating voltage of the datapath in the circuit. This method proves that the supply voltage scaling or the “justin-time processing” can exploit the robustness of the asynchronous circuits to operate at low supply voltage and significantly reduce energy consumption.

The DC-DC converter used in [56, 86] dissipates significant amount of power [56] produced by the series resistance of the package [87]. The CMOS fabrication of the converter is not only difficult involving passive parts like capacitor (C) and inductor (L), but also introduces area and delay overhead. Adoption of adaptive voltage scaling for asynchronous circuits suffers from two drawbacks [56]:

1. amount of power savings obtained is sensitive to the DC-DC converter efficiency
2. the area overhead introduced by the feedback mechanism that control the voltage has significant energy losses.

Although, adaptive voltage scaling is an innovative scheme to tackle dynamic energy issues in asynchronous circuits, the latch-up problem after a sudden \(V_{dd}\) voltage drop needs further investigation for the scheme reliability [86].
In [88], the authors propose a novel technique of signal bypassing and zero insertion targeting multi-rail design logic, NCL, for dynamic energy reduction. The proposed energy aware design technique reduces energy by controlling the amount of switching at the gate level. The technique has successfully been demonstrated in parallel adders, multipliers and rank-order filters [89, 85]. As the name suggests, signal bypassing technique involves bypassing the input to the output of a block when the other inputs of the block are all NULL.

That is, this energy aware technique reduces its switching by considering the DATA0 as NULL. And, the zero insertion technique is the compensation technique which works in pair with the bypassing technique. In this correction technique, DATA0 is inserted/replaced to obtain the correct operation of the design. This gate level technique, despite being design dependent, has proved itself to be advantageous when the input precision bits are low [85]. Along with delay, energy dissipation for NxN multiplier increases when the input precision bits are high because of the multiplexers used for implementing this technique. This design technique can be useful if the design multiplier has low precision and may not prove to be advantageous in terms of energy for higher precision bits [85].

2.2.2 Previous Static Energy Reduction Techniques

As we migrate to newer technologies with smaller nanometer gate lengths, in deep submicron technologies (DSM), the leakage current is becoming more pronounced and so is the leakage energy associated with it [90]. The transistor leakage current is exponentially dependent on threshold voltage ($V_{th}$) of the transistors and hence, circuit design with multiple threshold voltages has been proposed as a possible solution to deal with leakage energy issues. To tackle the energy reduction at DSM, depending upon the design constraints, designers have a plethora of options to choose the right technique desirable for the application intended. Some of the popular effective
techniques that have been explored for asynchronous circuits are:

- Dual threshold assignment
- Multi-threshold CMOS (MTCMOS)
- Super cut-off CMOS (SCCMOS)
- Power gating

We consolidate the above techniques in this section.

At high level of design, [91] proposes a simultaneous dual threshold voltage and supply voltage scaling process for minimizing the leakage power in asynchronous circuits. The rationale behind using the dual-threshold voltage scheme (low and high \( V_{th} \)) is, high \( V_{th} \) is used to suppress the subthreshold current \( I_{sub} \) in the circuit and low \( V_{th} \) is used to achieve high performance [92].

The authors propose an integrated framework for leakage power reduction from a high level circuit description. They believe that the synchronous method of dual-threshold assignment by finding the critical path is not possible in asynchronous circuits. This is because of the performance analysis dependency on highly concurrent events in asynchronous circuits [91]. Hence, instead of critical path analysis, they use timed petri-nets models for performance analysis and assignment of threshold values to the circuit components. A heuristics based on genetic algorithm optimally assigns the low and the high threshold voltages for gates in the circuit. The objective being to maximize the number of high \( V_{th} \) assignment as the low \( V_{th} \) components are known to consume higher standby power than the high \( V_{th} \) transistors [73]. They targeted the precharged full buffer (PCFB) type of asynchronous circuits. The effectiveness of this algorithm for NCL type of circuits’ needs detailed investigation.

Multi-threshold CMOS (MTCMOS) [93, 94] has emerged as an effective technique for reducing the leakage current in the DSMT. This state-of-art technique utilizes
sleep transistor for leakage current optimization in the standby mode. Fig. 2.3 the standard structure of Multi-threshold CMOS (MTCMOS) architecture.

During the normal operation of the gate, high $V_{th}$ transistors are turned ON connecting the low $V_{th}$ logic to power lines. During the standby mode, the low $V_{th}$ is cut-off from the power lines by the high $V_{th}$ transistors turning OFF. Thus, the sub-threshold current reduces during the standby state and ultimately, the leakage power is minimized.

Again, in this technique, low $V_{th}$ devices are used for circuit implementation to achieve good performance. This scheme is very attractive scheme as the existing gate architecture can be quickly modified by adding the extra high $V_{th}$ transistors in series with the power supply lines.

Bailey et al. [93] presents a design methodology based on MTCMOS targeting the threshold gates in NCL version of the asynchronous design. [94] uses a modified architecture proposed by [96] overcoming the drawbacks of partitioning and sizing of sleep transistor for large circuits. Fig. 2.4 shows the modified version of MTCMOS implementation of standard cells for NCL type circuits. This method incorporates a high $V_{th}$ transistor in parallel with low $V_{th}$ pull-up network (PUN) and pull-down network (PDN) respectively and an extra high $V_{th}$ PMOS between the two networks. The parallel transistors help to maintain equivalent voltage potential during the sleep

Figure 2.3: General MTCMOS Architecture [93].
The ultra low power design for the threshold logic NCL cells clearly eliminates the drawbacks of their synchronous versions [93, 94]:

- Sleep generation signals requiring extra complex logic has been eliminated in this QDI threshold logic version. The sleep signal needed for the standby mode is generated by the CD handshaking signals eliminating the extra hardware required for its generation.

- Data loss from the storage elements during the standby mode is eliminated in the NCL version as the circuit is in standby mode during NULL cycle; the cycle in which all the gates return to their NULL or “Zero” state. Hence there is absolutely no danger of data loss.

- The problem of sleep transistor sizing has been alleviated by the use of the sleep transistor in individual cells of the cell library, eliminating the transistor sizing dependency on circuit.

Clearly, MTCMOS for asynchronous NCL circuits not only eliminates the drawbacks of MTCMOS synchronous version but also presents a significant leakage power
reduction. The NCL version of the MTCMOS gates had zero or negative area overhead compared to the synchronous version. Also, eliminating the GO-TO-NULL and HOLD-DATA [14] blocks in NCL threshold gates version further helps in area reduction. However, drawbacks of this method are [92, 73]:

- High fabrication cost involved in design with dual threshold voltage.
- Variation of doping concentration in high and low Vth devices.

Recently, [80] address the issues in the current MTCMOS architecture for NCL gates and proposed an improved version of [93, 94] which eliminates the glitches and other spurious signals during the “wake-up” event in MTCMOS. For further details on the implementation of the new MTCMOS for NCL refer [80].

[97] presents power gating techniques exclusively for asynchronous circuits. They propose a state preserving and non-state preserving power gating technique. A cut-off power gating is intended for non-state preserving power gating technique whereas, zig-zag cut-off power gating technique is used for state-preserving gating technique. The techniques were applied for pseudo-static gates and its effectiveness to reduce the leakage energy was demonstrated. These gates are equivalent to semi-static version of the NCL gates and hence applicable to NCL. [97] provides in-depth these techniques along with its implementation on the AES encryption/decryption module.

Shreih et al., [95], provide ultra low power solution by technique called multiplexed input super cut-off CMOS (MISCCMOS). MISCCMOS overcomes the shortcomings of MTCMOS and super cut-off CMOS (SCCMOS). MTCMOS and SCCMOS techniques require the extra sleep transistor added in series with gate or the circuit to cut-off the supply lines during the stand-by state. This sleep transistor addition has many disadvantages [95]:

- The sizing of sleep transistors is tricky and at times turns out to be bulky, if the gates share a common sleep transistor. The leakage current reduction is
dependent on sleep transistor sizing.

- The sleep transistor also translates as extra capacitance that needs to be charged and discharged.

- Power supply scaling for that circuit is also limited because of the voltage drop across the sleep transistor.

Thus, to eliminate the disadvantages of sleep transistor, shreihi et al. proposes the MISCCMOS scheme. Fig. 2.5 shows the implemented technique. The elimination of series sleep transistor is done by a novel method of applying overdrive voltage to all or some of the PMOS transistors in the PUN of the circuit, unlike the SCCMOS method of applying over-drive voltage to the sleep transistor during the stand-by mode.

### 2.3 Summary

This chapter provided a brief insight into the research work that is currently available for asynchronous circuits’ energy reduction. Extensive research has been done in design using asynchronous design to tackle the static power issues which are accessible today. This is because as the technology scales, leakage power is expected to increase manifolds. Also, the ongoing research has proved that asynchronous design are indeed low energy circuits and are good reliable candidates for ultra-low power design [79].
The results from these research works are in the right direction for general acceptance of asynchronous design for low energy design.
3 Introduction to Asynchronous Systems

This chapter provides a brief overview of asynchronous design methodology and terminologies. The data propagation method, data communication method and control mechanisms are all elaborated with the focus on NCL threshold network methodology for asynchronous design.

3.1 Introduction to NULL convention logic (NCL)

The birth of the clock free logic can be attributed to some of the earlier attempts such as Mullers C-element [8] along with dual rail encoding schemes [45] to create a self-timed circuit [67]. The commonly used Boolean logic for the synchronous design could not be considered for the asynchronous design due to its expressional insufficiency or symbolic incompleteness [59].

“A symbolically complete expression is defined as an expression that only depends on the relationships of the symbols present in the expression without any reference to the time of evaluation” [47, 46]. In other words, a symbolically complete logic is one which has no timing relationship involved and is insensitive to propagation delay among its components [59] which are the essential characteristics of asynchronous design methodology.

In contrast, a clocked Boolean logic uses a straightforward, regular method of external control logic like clock for output validation and this gives rise to time-dependency relationships. Time-dependency of the expressions directly translates to propagation delay dependency of the components to express the validity of the output data values. For instance, a Boolean AND gate output data validity depends on the propagation delay of the gate and if the delay was say 1ns, the output of the gate is valid only after 1ns. In general, data sequencing is not an issue in clocked logic as long the output is valid at the clock edge.

In traditional Boolean logic design the control and datapath circuitry are consid-
ered independent and are designed independently which are later coordinated carefully for correct operation. But, in asynchronous circuits in the absence of clock, these two circuitries are inherently dependent [59]. The pursuit to eliminate time dependency gave rise to a logically determined logic, the NULL convention logic (NCL) [59] which is expressively sufficient [59]. Developed by Karl Fant and Scott Brandt of Theseus Logic (now Camgian Microsystems), NCL is a logically determined self timed design methodology that eliminates time references and incorporates data and control information into a multi-value encoding scheme. This brings about the symbolic completeness of NCL [59] and delegates them as candidates for asynchronous methodology.

A broad overview of asynchronous design terminologies and protocols that help build a functionally correct asynchronous system is detailed in the following sections with the emphasis on the asynchronous threshold network circuits such as NCL. Later, an inexpensive method of realizing threshold gates with hysteresis [59], which are fundamental building blocks of NCL system and its realization in CMOS are outlined.

3.2 Communication protocols in Asynchronous Systems

Though designing the systems with global clock simplifies the design process, functionality of the design can still fail because of the timing closure issues associated with the clocked design. The assumption made in designing the synchronous logic design is that the clock reaches every nook and corner of the chip uniformly, but in reality, there is clock skew associated with clock signal. Eliminating the clock is potentially the best solution that is available to this date to overcome the timing closure problem encountered in clocked logic design, apparently termed as self-timed design [67].

In the absence of global clock as the control signal in asynchronous circuits, the correct operation of the system and the data scheduling has to be ensured to avoid
potentially hazardous operation. Localized signals that communicate with adjacent block through signaling protocol of request and acknowledgement is used for data communication in asynchronous circuits. This handshaking principle eliminates the need for high-speed clocks and takes the functionality of clock in asynchronous circuits.

Fig. 3.6 shows the basic handshaking mechanism in asynchronous circuits. The ‘handshake’ package consists of the acknowledge signal, request signal and the data bus. The operation is explained as follows: the data to be processed is placed on the data bus and a request signal REQ is asserted, if the receiver block is inactive, it will accept the data and when the processing is completed, an acknowledgement signal, ACK, is sent back to the transmitter indicating the completion of computation/data processing.

The two most widespread use of this encoded control signal are two phase signaling protocol and four phase signaling protocol [36]

**Two phase signaling protocol:** Fig. 3.7 (a) shows the two phase protocol
signaling. In this signaling, rising and falling edges of the signal are used for communication. A rising edge on the request line (REQ) indicates the receiver that the data is ready for communication and then after accepting the sent data the receiver sends an acknowledgement signal to transmitter rising the acknowledge line (ACK) high. This asserts the request line low and signals the receiver for next cycle of data. Note: here the request signal is assumed to start from its ‘low’ state, while in some cases it can be start from its ‘high’ state as well.

**Four phase signaling protocol:** Fig. 3.7(b) depicts the four phase protocol signaling. Here, all signals (request and acknowledge) starts from the low or the ‘zero’ state. As the REQ is asserted, it indicates the willingness of transmitter to send the data to the receiver. The receiver accepts the data and acknowledges the transmitter by asserting the ACK line high which returns the REQ line to its initial state or it is asserted low.

The next cycle continues with the REQ being asserted high. In all, it takes four transitions for the communication, two for the request (REQ) and two for the ACK signal as they return to the zero state. The other names given to this signaling protocol are return-to-zero (RTZ/RZ), four-cycle protocol and level signaling. On the other hand, the two phase signaling is denoted by two-cycle protocol or non-return-zero (NRZ) signaling protocol.

On the first inspection, NRZ scheme seems to have the advantage in terms of speed and power. But, this signaling scheme requires extra circuitry for its direction detection as the data can be transferred on both the edges. This translates to power dissipation by the extra circuitry that is saved by reducing the signal transitions.

On the other hand, the RZ scheme has the advantage of being robust and limitations of being complex and slower. RZ is the most preferred technique for asynchronous handshake control logic [36] and NCL uses this scheme. In some cases, RZ method has shown superior advantage in terms of power saving like the ARM2
processor which uses the RZ scheme that has better power savings than the ARM1 processor using the NRZ scheme [16].

3.3 Asynchronous design - Delay Models

The fundamental notion of timing is delay. Every digital design component used in the design has inherent delay. The entire circuit behaviour is regularly modelled with delay models that accounts for timing. These models take into account the delay of gate, wires or both. There is growing gap between the wire and the gate delay models as we migrate and move to ever shrinking DSM technologies. This has further fuelled the design methodologies without timing dependencies. In the absence of clock, asynchronous circuits are designed with no delay assumptions. The common principle is to ensure that the circuit works correctly under varying gate and wire delays. Depending on the timing assumption of wire and gates, asynchronous circuit designs are broadly classified as follows: [16]

1. Bounded delay mode: Circuits belonging to this class have a finite value of delay in the given time interval. In other words, the assumption is gate and the wire delays are known, limited or at least bounded. This is the most simple and common assumption and parallels the design assumption in synchronous design and was used in the earlier days of asynchronous research.

2. Unbounded delay model: As the name suggests, the assumption made in these delay model is the wire and the gate delays are unknown or arbitrary. The times it takes for a signal to propagate may take a finite positive value.

A further classification of the asynchronous circuits is shown below, which can be categorized based on the assumption made on gate and wire delay that fall either in, bounded delay or unbounded delay category.
**Fundamental mode circuits:** these type of circuits changes inputs only when the circuit is in the stable state. In the simplest of the cases, only one input is allowed to change at a given time (single-input-transition) and the multiple input change circuits are termed “burst-mode circuit” [16]. This simplest method of implementation of asynchronous circuits incorporates a bounded delay model.

The construction resembles very closely the synchronous circuit design with combinational design enclosed between the input and output register stages and replacing the clock signal by local hand-shake signals. These local signals are typically implemented by matched delay lines to provide the timing for circuits [59]. The delay lines are calculated such that the output would have settled to a stable state before the next set of inputs are applied to the circuit. The matched delays are usually implemented with a chain of inverters or by replicating the critical path of the circuit [57]. Following a “bundled-data” representation, they are conceptually very elegant in terms of area saving and design, though the fixed worst case completion time used for its design foils its elegance.

**Speed Independent (SI):** Circuit belonging SI assumes infinite delay on gates, but the wire delay are negligible or zero. Obviously, they fall in the unbounded delay model category.

**Delay Insensitive circuit (DI):** DI type of circuit design assumes infinite or unbounded delay on both the gate and the wire. This type of circuit design is overly restrictive, expensive and difficult to design [59]. A very few operative circuits have been designed that are fully DI. To ensure correct data transactions, DI type of design requires a completion detection (CD) circuitry that aids in handshaking protocol. Every wire fork in DI circuit will have to be acknowledged before the next set of data are presented for computation. DI implementation is becoming more attractive scheme for designers because
• It is autonomous of gate and wire delays and can adopt to variations in such delays.

• Independent of the timing interval in which the signals switch.

• Independent of number of signals switching their values.

As a result, the DI scheme exhibits robustness and reliability. Technically, this simplifies the synthesis process, since timing verification may not be necessary.

**Quasi Delay Insensitive (QDI):** This circuit is a class of DI circuits that assume unbounded delay on gates. But wires are isochronic forks [83, 84]. An isochronic fork is a forked wire or a fanout where all branches are assumed to have exactly the same delay [9]. An asymmetric isochronic fork is one in which the output transaction arrives faster in one fork than the other, while in symmetric isochronic fork, the data transition arrives at the same time. QDI though a robust “relaxed” version of DI still requires CD only from one of the wire forks. NCL falls in this category of asynchronous design with timing assumption only on wire forks [17].

Berkel et al. extended the isochronic assumption stretching it through the gates and named this relaxed version of timing assumption as quasi-quasi delay insensitive or $Q^2DI$ [12]. Though it weakens the isochronic fork assumptions and compromises it’s robustness, $Q^2DI$ provide the benefits of low cost circuit and fewer timing assumptions. The fork implementation is further simplified by circuit symmetries and easy realization by CMOS that arise due to this assumption [12].

Fig. 3.8 shows the various communication methods based on the delay model assumption and compared with the synchronous version of communication.

**NCL** is distinctive way of implementing asynchronous threshold network logic based on QDI model. Without making any assumption on timing, NCL has the ability of ensuring correct data sequencing and determining the correct arrival of data at the receiver’s end under varying gate and wire delays [59]. It has proved
itself as a possible solution for clock free logic design by eliminating the race and hazards dogging the designers in digital design. It incorporates data and control signal in a multi-rail encoding scheme. It assumes a two-phase scheme in which data communication switches between the set phase and reset phase [17, 50]. In the set phase, data changes from NULL state to the DATA state and switches back to the ‘base’ or the initial state, NULL in the reset phase. Thus, the data set of NCL can be expressed as DATA0, DATA1 [47]. A NULL simply implies ‘not data’ and used as a ‘spacer’ between the two consecutive DATA elements [47, 59, 46]. A NCL system is assumed to be in the NULL state before all computational data can be presented to the system [59].

**3NCL:** A Boolean logic with NULL (N) value incorporated in the logic, is called the 3NCL. It has 0, 1, N values where 0, 1 are the Boolean DATA set and N represents the NULL data or value. Alternating between the set and the reset phase, the operation of the 3NCL can be explained as follows. Starting from all NULL state, when
the all the inputs receive a valid DATA value, the output switches to valid DATA state. And in reset phase, when all the inputs are presented with N, the output returns to its NULL state. When one of the inputs goes to NULL, the output still maintains a DATA state until all the inputs have NULL. This enforces the property “input completeness” [59] which is the key to DI logic circuits and it acknowledges the correct operation and arrival of input data at the output. No external expression or control circuit such as clock or a delay line is required for the operation [59].

3.4 Data Representation or Encoding in Asynchronous Circuits

In synchronous circuits, the data is valid at a given time by the control signal clock. That is, the data can be resolved as logic 0 or logic 1 at every clock edge. In the case of asynchronous circuits, in the absence of time as reference, the validity of data cannot be confirmed. In other words, there is no guarantee that the data has been properly communicated to the receiver.

Asynchronous circuits communicate with adjacent blocks following the principle of handshaking. In handshaking mechanism, signal transitions are the only method of validating whether the data has been properly communicated to the receiver. For data communication in asynchronous circuits there are various choices in which the data can be transported.

The first choice is the “bundled data” mechanism, simple conventional synchronous type data encoding, employs single wire per bit. For communication, it requires a
Table 3.1: Data representation in dual rail encoding (NCL) [46, 47].

<table>
<thead>
<tr>
<th>Rail_0</th>
<th>Rail_1</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>DATA 1, Boolean ‘1’</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>DATA 0, Boolean ‘0’</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>NULL/Reset</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Illegal State</td>
</tr>
</tbody>
</table>

control signal or request line, data bus and the acknowledge signal line. Thus, for
N-bit data transportation, it requires N+2 lines or wires [16]. Fig 3.9 (a) depicts the
bundled data communication.

Fig. 3.9 (b) is the dual-rail encoding. In this method of encoding, each bit of
data is encoded in two separate lines. The main concept in these encoding schemes
is that, the wires or rails will carry its own mutually exclusive data [16]. Two rails
(rail0, rail1) can take any of the below possible states. Table 3.1 indicates the possible
combinations.

This data encoding scheme incorporates the request signal in its data lines and
eliminates the need for separate request line [16]. The scheme requires 2N+1 wires for
its N-bit data encoding [16]. Though this method is a robust method of DI encoding,
it creates significant area in terms of the data communication wires and CD circuitry.

2NCL: Although 3NCL is very simple, yet convenient representation of NCL and
is theoretically DI, it is not possible to practically realize in binary valued 2-value
logic circuit implementation [59]. The physical realization of 3NCL in its two value
logic is 2NCL realization. In the digital design world, the logic is limited to two
values. For example, the binary logic is realized with say, 0v for Boolean ‘0’, and 5v
for Boolean ‘1’. Similarly, in NCL type circuits, the NULL state is assigned a 0v and
the other two DATA values are realized with mutually exclusive dual rails (RAIL_0,
RAIL_1), meaning the rails (RAIL_0 and RAIL_1) cannot have DATA at the same
time, it is an illegal state.

The wavefront switches monotonically from a data wavefront (transition from
fully NULL state to a valid DATA state) to non data wavefront (transition from fully DATA state to a NULL state). Adhering strictly to these monotonic transitions eliminates the race, hazards and glitches in the circuit design [47], unlike synchronous clocked design that amounts for substantial glitch power.

The NULL state in NCL serves the purpose of request signal and when the data is asserted on the data lines, the correct transportation of data is validated by the acknowledgement signal that is sent back to the transmitter through the CD circuitry. This explains the data communication in NCL type systems.

The other method of data encoding scheme that is available in the research community is quad-rail encoding scheme. This scheme is very similar to dual rail encoding scheme and it uses four wires to encode the data DATA0, DATA1, DATA2, DATA3, NULL. DATA0 state corresponds to two Boolean logic signal, X = 0, Y = 0. Similarly, the DATA1, DATA2 and DATA3 are all one-hot-encoding schemes of other X, Y states. Table 3.2, shows the quad rail encoding scheme. In NCL design, a quad-rail data encoding scheme can be used for data communication and their application is demonstrated in [10, 11]. Note: The quad rails are mutually exclusive and the state (1, 1, 1, 1) is illegal and cannot occur.

Table 3.2: Data representation in Quad rail encoding (NCL) [47, 46]

<table>
<thead>
<tr>
<th>Rail_0</th>
<th>Rail_1</th>
<th>Rail_2</th>
<th>Rail_3</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X=0, Y=0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>X=0, Y=1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>X=1, Y=0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>X=1, Y=1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>NULL/Reset</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Illegal</td>
</tr>
</tbody>
</table>

Figure 3.10: A discrete threshold gate
3.5 NCL logic gates

NCL circuits are constructed from the, key component, threshold gates (TG). These discrete TG gates are used in the construction of NCL based systems. TG can be combined effectively to realize system that can resemble the traditional clocked Boolean logic systems. The example of a TG is shown in fig. 3.10 which has 5 inputs and a threshold of 3. A TG has many inputs but only one output with a number on the body representing the threshold number.

The operation of TG is as follows: when the inputs of the gate are presented with valid DATA more than the threshold number, which in this case is $\geq 3$, the output switches to DATA state indicating or satisfying the input completeness of the input criteria for DATA in relation to NULL [59]. But, even when one of the input changes to NULL, breaking its threshold, the output switches to NULL state. The NULL state at the output is an indication for the next set of data to be presented to the input, but there may be still data present in the other DATA rails which leads to an erroneous operation. In other words TG does not satisfy input completeness of the input criteria for NULL in relation to DATA [59].

To overcome the problem of TG, Thesesus logic came with a simple, yet effective, economically viable solution of adding a gate result or output feedback to the input of the TG [59]. Fig. 3.11 shows the simple feedback solution of a 2 input TG. As seen in the figure, when the input A and B are data, the data is fed-back and the threshold requirement is met. When either A or B is NULL, the Z which has a data, feeds back the value meeting the threshold and requiring both the inputs A and B to be NULL for the output to switch to NULL.
The addition of feedback in TG provides the special property of state holding and this unique functionality is provided by hysteresis, sometimes called Threshold gate with hysteresis (TGH) [50]. The rule for the TG gate to work properly as TGH is that the weighted feedback is one less than the threshold number of the gate. In addition, this makes the TGH gates with the set and the reset functions sequential gates unlike, the TG which are combinational. Fig. 3.12 (a) illustrates the conversion of TG to TGH of a 5 input threshold 3 gate. Fig. 3.12 (b) shows the feedback solution being applied to the TG gate. Fig. 3.12 (c) shows the TG with feedback, TGH, which has a rounded back where the inputs are added and a pointed output.

TGH gates have many inputs and a single output and denoted by, \( THmn \) gate, where ‘n’ denotes the number of inputs to the TG and ‘m’, where \( 1 \leq m \leq n \), denotes its threshold as shown in fig. 3.12 (d). Threshold of TH35 gate is met when the inputs (\( \geq 3 \)) of the gate are data and the output Z of the gate switches from NULL state to the DATA state satisfying the input completeness of input criteria for DATA in relation to NULL. To satisfy the input completeness of input criteria for NULL in relation to DATA, the gate output stays at DATA state until all the inputs of the gate have switched back to NULL state.

The two important properties of TGH gates are hysteresis and threshold. The operation of the TGH, \( THmn \) is as follows [59, 47]:

- **Threshold property**: The output of the threshold is asserted if the gate has valid ‘data’ on ‘m’ line of its ‘n’ inputs; i.e. its threshold is supposed to have been met and the gate output is asserted.
• **Hysteresis property**: The output will remain in its asserted state even if one of its inputs goes to NULL and will transition to NULL only after all the outputs have transitioned to NULL.

Form the above properties of TGH, it is clear that NCL operators need a state holding property that can hold the DATA values until all the inputs have switched back to NULL and hold NULL value until all DATA values have been presented at its input. This means, TGH’s state holding property is provided by adding the hysteresis [59] characteristics to these unique gates. Hysteresis also aids in monotonic transitions making NCL logic glitch free [47, 33]. This differentiates NCL gates from its C-element [47, 46] counterparts.

Another unique feature of NCL operators which optimize the NCL gates in terms of transistor count and delay is the weighted threshold gate denoted by denoted by $\text{TH}mnWw_1w_2...w_R$ [47, 46]. Weights are added on the inputs represented by $w_1, w_2...w_R$ which corresponds to each of the input1, input2...inputR and $m \geq w_R > 1$ (i.e. weights are always positive integers) [47, 46].

For example, $\text{TH}23W2$, has 3 inputs (A, B, C) with a weight of 2 on the input A. Fig. 3.13 (a), shows the NCL $\text{TH}23W2$ operator representation. The output is asserted if either input A or B and C are asserted with DATA. This completes the NULL to DATA transition and when all the inputs (A, B, C) is presented with a NULL wavefront, the output of the gate $\text{TH}23W2$ switches back to the NULL state and is ready for the next set of DATA. (Note: All gates are assumed to be in the NULL state, or the ‘base state’, before a DATA is presented to the gate.)
Similarly, for TH54W322 gate with 4 inputs (A, B, C and D), has a weight of 3 on the input A, weight 2 on B and C. Fig. 3.13 (b) shows the operator. The output of the operator is asserted DATA when the threshold of the gate is met. The threshold to be met can be expressed in a Boolean equation format as \((AB + ACD)\). Or in other words, when input in A and B or input in A, C and D are DATA, output Z is asserted DATA.

NCL can be considered as delay-insensitive (DI) circuits if it satisfies the following criteria \([47, 46, 59]\):

**Observability:** requiring every gate transition to be observable at least on one of the gate outputs. The unobservable transactions on either the gate or the wire in the circuit are called *orphans*. These signal transitions are unacknowledged by the primary output of the circuit and may cause the circuit to produce erroneous results or malfunction if the transition is too slow \([50]\). Wire *orphans* and gate *orphans* are the two types of commonly found *orphans* in asynchronous circuits. When the *orphans* are allowed to transit through the gates, it gives rise to gate *orphans*. Gate *orphans* are much more serious than the wire *orphans* as they may, sometimes, transit through a series of gates which can easily cause trouble to circuit functionality. The problem of gate *orphans* have been discussed in detail in \([50, 51]\).

**Input Completeness:** This criteria requires that the output does not transition from NULL to DATA until all the inputs have transitioned from NULL to DATA and output does not transition from DATA to NULL until all the inputs have transitioned from DATA to NULL \([59]\). The only exception being, in gates with multiple outputs, where the output can transition without a complete set of input sets as long as the output does not transition before the arrival of the inputs. This is based on Seitz’s “weak Condition” \([67]\).
3.6 CMOS Realization of NCL gate

NCL logic gates are the essential building blocks of NCL system. As discussed in the previous section, the two main properties of NCL logic gate that make them DI are, threshold property and hysteresis property [14]. NCL can be effortlessly realized in CMOS. This is because, the threshold property that determine the functionality of gate can be realized by Pull-up (PUN) and Pull-down network (PDN) of CMOS transistors whereas, the hysteresis property can be realized in CMOS with a state holding element. The state holding element is generally built with feedback elements. The NCL gates can be recognised in a variety of architecture depending upon the implementation structure of the feedback element. Some of the popular architectures are:

- Static
- Semi-static
- Dynamic

NCL gates switches when the number of input signal meets its threshold value and retains its asserted state until all the inputs have switched to NULL state. This functionality of NCL gives rise to four different building blocks:

- GO TO DATA
- GO TO NULL
- HOLD DATA
- HOLD NULL

The general architecture of NCL gates is shown in fig. 3.14 (a).
For a static structure, “GO TO DATA” block which accounts for threshold property of the NCL gate is implemented with PDN of N-transistors and PUN implements the “GO-TO-NULL” structure with a series chain of PMOS transistors that essentially resets the gate. The “HOLD NULL” and “HOLD DATA” structure are complementary blocks of “GO TO DATA” and “GO TO NULL” respectively. They implement the hysteresis property of NCL gates.

For a semi-static NCL gate architecture, the hysteresis is implemented with a cross-coupled inverter block replacing the “HOLD NULL” and “HOLD DATA” blocks in static versions. Again, the threshold property is implemented with the “GO TO DATA” and “GO TO NULL” blocks. The implementation of static and semi-TH22 gate is shown in fig. 3.15 (a) and (b) respectively.

For high speed applications, dynamic configuration is implemented with just the “GO TO DATA” and “GO TO NULL” blocks. The three different architectures are shown in fig. 3.14. For more details on implementation of all the three architectures, refer [14].
3.7 NCL system

In NCL circuits, the dual-rail asynchronous registers and the combinational circuitry consisting of NCL gates along with completion detection circuits form the basic pipeline structure as depicted in the fig. 3.16. The neighboring registers along with CD circuitry forms a part of handshake control mechanism of NCL circuits. Despite the absence of clock, NCL lends itself to pipelining by alternating through a NULL-DATA-NULL cycle and communicating with the adjacent registers with the request-acknowledge four-phase signaling protocol. This unique feature significantly differs from the pipeline structure in clocked Boolean logic. The dual rail registers are composed of the TH22 gates and the completion detection circuitry consists of tree structure of THnn gates, where n ≥ 2. For detailed explanation of the NCL system and its communication refer [59].

3.8 Advantages of NCL for asynchronous design

To summarize here are some of the advantages of using NCL for asynchronous design [59]:

Figure 3.15: Transistor topology of (a) Static TH22 gate; (b) Semi-static TH22 gate
Ease of design: NCL circuits are symbolically complete and they are inherently DI, which eliminates the need for external control circuits like clock. They can conveniently be designed without any timing assumption and still guarantee to work correctly. They can be easily expressed in high level languages (VHDL/Verilog), synthesized and compiled in silicon just like their synchronous counterparts.

Cost effective and low risk: In synchronous circuits, clock tree with its clock buffers and other attendants not only consume large area on the chip, but also drain the major budget time and cost allocated for the design. Apart from creating this clock which is proving to be difficult to the on-chip process variations, they are a burden in the form of clock skew and jitter. Throwing away the clock is not only cost effective but also reduces the risk of malfunctioning of the circuit due to clock skew, race and hazards.

Low power design: In the absence of clock and its associated components, an asynchronous circuit switches when “active” and remains dormant otherwise. This offers the advantage of distributing the power over the chip. A glitch-free NCL circuit is an ideal candidate for low power design.

Low noise and EMI: Relinquishing the simple clock in NCL design eradicates
the simultaneous switching of the large number of transistors and this translates to straightforward advantage of crosstalk and substrate noise mitigation.

**Immune to hostile environment:** In DSM, the designs performance is affected by process, temperature and voltage variations. Along with other manufacturing variations, the design is subject to delay variations in these hostile environments. Since NCL is inherently DI, they can easily adapt themselves to these unforgiving environments and still operate correctly under power and voltage variations.

**Ease of technology migration:** Emphasizing again on the DI property of the NCL type circuits, they are insensitive to the physical property variations and its behavior. They are easily portable to different technologies proving to be promising for asynchronous design as they show no sign of propagation delay changes due to ageing or other manufacturing variations [59]. With no detailed timing analysis and changes in scale, a design rule change is sufficient for technology migration.

**Reliability:** A circuit is reliable when it can operate correctly under all varying conditions and has no failure modes. Purging the clock in NCL design eradicates the failure modes like race, hazards and skews. A risk-free design not only avails reliability but also cost-effectiveness.
4 Voltage Scaling for Energy Reduction

Equation 2.5 models the relationship between energy within the circuit and supply voltage with a quadratic equation. From (2.5), scaling the voltage should lead to energy reduction/savings. Historically, supply voltage scaling has been shown to be the most effective power reduction technique in synchronous circuits [63, 62] at the cost of delay penalty. Dual voltage techniques have also been explored for synchronous circuits power reduction [63, 62, 64, 60, 61]. Typically these techniques use the regular high supply voltage (HV) for critical paths and lower supply voltage (LV) for non-critical paths where LV < HV. In this chapter we apply this technique to asynchronous circuits and we present energy savings due to dual voltage (DV) scaling.

4.1 Introduction

Dynamic energy reduces quadratically when the supply voltage is scaled down with penalty to be paid in terms of performance. Instead of reducing the supply voltage of the entire circuit, the gates in the non-critical path can be operated at a reduced supply voltage by trading slack available in each gate for energy as gate delay increases due to decrease in voltage. Timing critical gates are operated at HV to maintain the timing requirements of the design. This simple, yet effective technique is implemented at the gate level without sacrificing the performance of the system.

Depending on the low power requirements of the design, multi-LV power supplies or dual voltages can be implemented. However, design with multi voltages comes with two main issues: (i) Multi-voltage scaling calls for multi-supplies on board thereby increasing the supply voltage footprint and interface between these multi-voltage blocks requires complicated and expensive interfaces [29, 5]. (ii) Multi-Vdd technique require level converters which directly increase energy, area and delay overhead [60, 61], coupled with increase in wiring congestion [31] and the signal integrity issues that arise when signals switches to and from multi-power domain. Therefore, this work
focuses only on dual voltage for energy reduction in asynchronous circuits.

To sum it up, DV technique offers the following advantages [62]:

1. The existing gate netlist and the arrangement after design synthesis can be used with supply voltage changes in gates with extra slack.

2. The threshold voltage need not be changed offering the advantage of using the same fabrication process.

3. Existing commercially available off-the-shelf (COTS) CAD tools can be used to identify the excess slack in the circuit and those “potential” slack can be converted to power/energy savings.

The problem of energy reduction of asynchronous circuit with dual voltage technique is described as follows. For every gate in the circuit, apply either HV or LV with the objective of maximizing the LV cell assignment such that the timing constraint is not violated and the voltage assignment ultimately leads to energy reduction.

In DV designs, often signals cross over one energy domain to another and there are two cases that arise. A LV signal driving a HV cell is shown in the fig. 4.17. In this case, when the node ‘X’ is at logic ‘1’, the PMOS side of the HV cells is not completely shut off since \((VddL < VddH - |V_{THp}|)\), leading to high static current flowing from Vdd to ground which leads to substantial static power loss. Also, problem of reduced
noise margin of the LV signal may cause erroneous results at the output of the HV gate due to the low drive capability of LV cell [32]. To counter these problems, a level changing (LC) or level shifter which acts as an isolator at the boundary of two power domains is required. LC’s are simple buffer type circuits that provide the necessary isolation between the source (LV) and destination (HV) power domain. It reduces the leakage current and also provides the necessary drive by scaling up the input voltage to HV. Despite its advantages, the LC cell causes area, energy and delay overhead.

In the second case of a signal travelling from HV to LV domain, the LV cells are “overdriven” by the HV signals leading to minor changes in rise/fall time and/or propagation delay [32]. This may be acceptable in many cases and the design can be implemented with no LC cells when going from HV to LV cells.

A voltage assignment demanding a large number of LC cells may not provide significant energy improvements because the LC cells themselves dissipate energy and also introduce area and delay overhead in the design. This design limiting factor calls for LC optimization along with LV cell assignment. In other words, the DV assignment algorithm must account for the delay and power/energy penalties in the optimization function while minimizing energy/power. In this work, two approaches for DV assignment to reduce energy dissipation in circuits are under-taken along with a timing constraint

- DV assignment technique without LC.

- DV assignment technique with LC.

This research work demonstrates the VS technique with a widely accepted LC-free design and shows how a design with LC can achieve better energy savings with two of the popular static VS techniques used in synchronous design for asynchronous design.
4.2 Static Voltage Scaling Design without LC

Horowitz et al. [63], first proposed this dual voltage technique without LC. In this technique, a rigid topology is followed. A HV cell always drives a LV cell and this, results in a LC free design. Fig. 4.18 shows the CVS technique. The goal of the CVS technique is described as follows:

Aim: To maximize the assignment of LV cells in the design thereby reducing energy without violating [63]:

1. The topological constraint: HV cell always drives a LV cell.

2. The design time constraint.

The CVS technique is implemented as follows:

1. The design starting point is a gate level netlist with all gates assigned HV and the circuit netlist is now represented as directed acyclic graph (DAG) $G(V, E)$, where each gate is represented by a node in $V$, and edges in $E$ between nodes represent wires.

2. The LV cell assignment starts from the primary outputs (PO), and traverses the circuits to primary inputs (PI). Circuit traversing can be done by simple and fast traversing techniques such as breadth-first-search (BFS) or depth-first-search (DFS) technique.
3. As the circuit is traversed from PO to PI (reverse topological traversal), the slack available at the each node is examined for LV cell replacement.

4. A cell is assigned to LV, only IF all its fan-outs (FO) are also LV cells and the assignment doesn’t violate the timing constraint. The timing violation can be monitored by static timing analysis (STA) that gives a quick, vectorless and gate functionality free timing analysis of the circuits.

5. After a cell assignment is accepted, the slack timing on all nodes are updated before proceeding with the next assignment.

6. As the algorithm proceeds, the slack at each node reduces. The LV assignment stops, when there isn’t enough slack left (or in other words, the LV assignment leads to timing violation) at the node that can be assigned to LV or all its FO’s are not LV or when all the gate have been considered for LV assignment.

The advantages of CVS are summarized as follows [63]:

- LC free design directly translates to area, delay and energy overhead elimination. Although, a LC is needed at the PO nodes to restore the output voltage level to HV.

- Low layout overhead because of its clustered structure, where LV cells and HV cells can be grouped and placed and routed in separate rows.

- This simple yet effective technique offers a runtime complexity of $O(V^2)$ [32, 25].

- They do not modify the topological structure of the design. Only the supply voltage at each gate is modified.

Disadvantages of CVS technique are as follows:
• CVS technique has a tendency to break down under tight timing constraints as the number of nodes operating at LV is limited by the slack available in the circuit [60].

• Also, cells on the PI side may have “potential” slack but LV assignment may not be possible since it will violate the topological constraint of the CVS algorithm.

• Energy saving numbers reported by CVS is sub-optimal as it may get stuck in local minima.

• CVS technique is very sensitive to its starting point. Often, heuristics are used to decide the best starting point. [63] proposes to prioritize starting point by using either gate load capacitance or the slack in descending order.

4.3 Static Voltage Scaling Design with Level Converters

In design with LC, this research focuses on the two main techniques (Extended-CVS (ECVS) and Greedy-ECVS (GECVS)) amongst many others that are available today [27, 26, 60]. These two techniques are related to CVS and have shown huge improvements over the conservative CVS. The techniques are explained in detail in this section and their implementation for asynchronous circuits is discussed in the next section along with the results.

4.3.1 Technique 1: ECVS

A CVS technique uses the slack available at the node to convert the HV cell to a LV cell. Due to its topological constraint the slack available in the circuit is under utilized which otherwise might lead to better energy savings. Fig. 4.19 shows the slack distribution in the circuit before and after applying the CVS technique. The plot clearly indicates that the “potential” slack available in the circuit is under-utilized because of the rigid topological constraint of CVS.
A more flexible version of CVS, called the Extended-CVS (ECVS) was introduced by Usami et al. [62]. In this method, the rigid topological of HV cell driving a LV cell is relaxed, where a LV cell is allowed to drive a HV cell by introducing a LC at the boundary. ECVS proceeds in the similar fashion as CVS, i.e, traversing the circuit in a levelized manner by applying the reverse graph traversal methods - BFS/DFS.

The algorithm replaces the HV cell with LV provided the assignment obeys the timing constraint. A LV cell assignment meeting the timing is accepted despite having HV cells in its fan-out (FO). Regardless of the fact that a LC introduces delay and consumes energy, the key idea of adding LC in the design is to discover more LV cells deeper in the circuit that can lead to incremental energy savings. This technique is continued until all the nodes in the circuit have been considered for potential LV
cell assignment. Fig. 4.20 shows an example ECVS arrangement. More detailed description of ECVS algorithm can be found in [62, 32].

Advantages and drawbacks of ECVS:

1. Achievable energy savings from ECVS are higher compared to CVS since ECVS has a more relaxed approach to voltage assignment compared to CVS. Also, ECVS incorporates CVS theoretically leading to considerable energy savings.

2. LV cell assignment is similar to CVS and thus the runtime complexity for ECVS is also $O(V^2)$ [32, 25].

3. Both CVS and ECVS approaches the voltage assignment problem in a levelized manner. This heuristic approach merely assigns LV in a constrained manner thus, resulting in non-optimal solution.

4. ECVS energy achievements are largely governed by the delay and energy of the standalone-asynchronous LC used in the design. The power/energy savings have been shown to be sensitive to LC delay in [32, 64].

5. In addition to using dual voltages, ECVS uses standalone LC that presents an area overhead at the physical layout level.

4.3.2 Technique 2: GECVS

The major drawback of CVS and ECVS style of LV assignment is levelized method of approach. The timing slack available in the circuit may be drained quickly due to its first-seen-first-assign approach. Since static VS techniques (CVS and ECVS) discretely assign supply voltage (HV or LV) without considering the sensitivity of energy to delay, they tend to leave more slack unutilized, impeding further energy savings [61]. Furthermore, as [64] explains, a LV assignment can lead to increase in energy dissipation. This is because unconstrained LV arrangement may call for a
large number of LC assignment eliminating the primary objective of energy reduction by lowering gate voltage. At times, these approaches may not lead to large cluster or good number of LV cell clusters consequently resulting in sub-optimal energy savings. A large cluster or groups of clusters of LV cell reduces the usage of LC in the design and in that process mitigates the overhead associated with LC usage. Lacking a global view, CVS and ECVS may not be a powerful energy reduction technique especially when the timing constraints are very tight [60]. All these facts motivates one to look for a more efficient approach that would depart from the constrained LV appointment and exploit the slack available at each gate in the circuit for maximum energy savings.

It’s imperative from the above discussion that we choose gates wisely, that is, prioritize the LV cell selection such that the slack on the each gate can be utilized to maximum extend and thus, maximize the LV cell assignment in the design to achieve superior energy savings.

In any static VS technique at gate level, scaling down the voltage, increases gate delay and obviously reduces energy. This implies a system implementing a VS technique is sensitive to two key parameters: delay and energy. Hence, a system would provide the best energy savings to the sensitivity factor - energy per unit delay [28]. To identify the probable candidate for VS, the gate with largest energy reduction along with the least delay increment is given higher priority. Therefore, a sensitivity measure of change in energy due to voltage scaling per unit delay penalty can be used for LV assignment. To meet the timing constraint, the slack available at each gate could be used as a guiding factor to map the cells to LV.

Kulkarni et al. in [64, 25] proposed a greedy optimization technique that makes use of such a sensitivity measure that identifies potential LV candidates for energy minimization. Greedy algorithm works on the principle that iteratively choosing local optimal solutions will eventually lead to global optimum. The advantages of using greedy optimization techniques are:
• Cheaper than exhaustive search.

• Greedy algorithm assumes local optimal solution as part of global optimum.

• Always makes a next available solution best solution, usually requires sorting through all the choices.

• Fast, simple replacement for exhaustive search.

The sensitivity factor used in [64] greedy-ECVS (GECVS) is as follows:

\[
sensitivity(gate'x') = \frac{-\Delta\text{Energy}}{\Delta\text{Delay}} \times \text{slack} \tag{4.6}
\]

where \(\Delta\text{Energy}\) represents change in energy when gate ‘\(x'\) is assigned LV, and \(\Delta\text{Delay}\) is the delay change due to the change in voltage. The GECVS algorithm is described as follows:

1. Starting from all HV configuration, the sensitivity factor for all the gates are calculated and the gate with the maximum sensitivity factor that doesn’t violate the timing constraint is chosen for replacement.

2. Since GECVS doesn’t follow the rigid topological constraint as CVS, every LV cell replacement may call for a removal or insertion of LC.

3. GECVS accounts for LC expenses by considering the delay incurred when LC is added in the design along with its energy penalty.

4. Once a gate is assigned LV, the current state is saved and the slack timing for the nodes are updated before re-calculating the sensitivities of rest of the gates.

5. GECVS allows negative moves, or moves that increase the energy in the circuit. The hill-climbing capability of GECVS is key to its success of producing good clusters of LV cells.
6. The algorithm continues from its current state to explore more feasible moves that can be assigned LV without violating timing.

7. The algorithm stops when any more assignment of LV cells in the design leads to timing violation.

More details about GECVS along with pseudo-code are found in [64, 32]. To summarize, the advantages and disadvantages of GECVS are:

1. GECVS prioritizes the LV selection process leading to better solution than CVS and ECVS.

2. It produces better LV clusters along with a low LC count consequently producing incremental energy savings even in a design with LC. On average, GECVS has proven its effectiveness with over 2x power savings or more in some of the designs over CVS and ECVS [64].

3. Nevertheless, GECVS flexibility of LV arrangement comes at a price of having polynomial runtime. [32, 31, 25] shows that runtime complexity of this algorithm is $O(V^3)$. GECVS produces substantial energy savings at the cost of large runtime.

4. GECVS is also sensitive to LC delay and energy dissipation and hence requires a LC with low delay and energy overheads.

5. GECVS is based on greedy type optimization and power/energy optimization is a unimodal optimization problem, and thus it may lead to non-optimal energy savings.
4.4 Assumptions in Implementation of Dual Voltage Scaling (DVS) for Asynchronous circuits (NCL type)

The focus of this research is on dynamic energy reduction of the asynchronous circuits especially on the threshold network circuits such as NCL. This will be the first ever attempt aimed at reducing the dynamic energy at gate level for NCL type circuit. The above discussed methods CVS (design without LC), ECVS and GECVS (design with LC) are implemented and their implementation is discussed in detail in this section.

In implementing the VS schemes for NCL, certain assumptions were made that made the popular synchronous DVS techniques such as CVS, ECVS and GECVS applicable to NCL [34, 33] and they are discussed as follows:

- In NCL circuits, the input waveform alternates between a valid DATA wavefront followed by a NULL wavefront. In the NULL wavefront, the wires that were asserted in the DATA wavefront are reset. In our work, the input data rate is assumed to be constant and fixed. (i.e.) time duration for DATA and NULL
are equal. \( T_{\text{NULL}} = T_{\text{DATA}} \).

- The input and the output registration stages are neglected. That is, the inputs to the circuits are applied directly at the input pins rather than an input register loading the DATA to the input pins of circuit.

- No rise, fall time or transition time/signal slew are included in the delay calculation.

- Since NCL static gates exhibit delay robustness, parametric variations such as gate length and oxide thickness that impact the delay of the gate are ignored.

- As assumed in [34] and as shown in fig. 4.21 which shows the timing and the switching energy plot of a static-TH22 gate, this experiment also considers only the switching energy and the non switching energy are insignificant and thus, neglected.

- Upon close examination, [34] states that the NCL gates consumes significant energy only when the input switches and the static energy is constant and insignificant 4.21. Compared to the critical dynamic energy consumed in NCL circuits, leakage energy/static energy represents a meagre amount and therefore neglected from the energy calculation parameter.

- In NCL circuits the outputs are asserted during the DATA cycle and de-asserted during the NULL cycle. Thus, a NCL gate makes exactly two transitions in every DATA-NULL cycle and in general, the switching activity calculations have been simplified by assuming the number of switching at every gate in the circuit to be equal to two [55, 68].

- From the above assumption, the circuits are assumed to be immune to input vector that is applied to the circuit and the energy that is calculated is directly
proportional to the voltage and load capacitance instead of 2.5.

- NCL circuits are glitch-free circuits because of its monotonic switching between DATA and NULL. The glitch power which accounts for significant energy dissipation in synchronous circuits can be conveniently neglected in NCL circuits and this simplifies the energy optimization problem in NCL type circuits.

- The current work targets the IBM 130nm process with a nominal HV supply of 1.2V and the LV value chosen is 70% of HV. This LV voltage was shown to be ideal voltage for DV design that offers optimal energy saving in DV environment by [29, 30].

- In many of the energy-delay optimization problem, often dual $V_{TH}$ are also employed to maintain or reduce the delay penalty and account for leakage issues. For this work, dual $V_{TH}$ is not considered in the design that addresses these issues. Moreover, each additional $V_{TH}$ requires an additional mask and hence is beyond this dissertation work [31]. Throughout this work the threshold voltage $V_{TH}$ is assumed to be constant.

- Although, this work includes the effects of wire parasitic on delay and energy dissipation in the circuit, the $C_{wire}$ and $R_{wire}$ was considered to be constant, even when GS technique is employed.

- Static version of NCL gates are favoured over the semi-static and dynamic logic gates. Dynamic logic circuits are not considered, because they have higher dynamic switching and leakage energy and display higher sensitivity to noise compared to standard static cells. In addition, the semi-static gates were observed to have unreliable operation when the supply voltage was reduced and even with some careful sizing, the circuits failed quickly than the static NCL version because the PUN and PDN of the semi-static version could not fight
the contention current from weak feedback inverter [24].

- The transistor topology of the static gates remains the same immaterial of whether GS or VS is applied. Transistor re-ordering which is intended to reduce energy consumption by reducing the average number of signal transitions at the gate doesn’t have a profound effect on dynamic energy reduction [61] and is strictly out of bounds from this work.

- Netlist topology of the design is un-altered. Meaning, No extra logic cells or duplication and remapping of logic are used that can provide additional slack which can affect the performance or energy savings in the circuit.

- Input pin capacitances of the cell are used to calculate the load capacitances measured from HSPICE and this capacitance is assumed to be fixed and constant when the voltage at the gate is reduced.

- Asynchronous circuits are intended for applications where area is not a major concern and in this current work too, there are no area constraints. The inherent low noise and low power advantages combined with significant energy savings obtained by applying VS-GS techniques to asynchronous circuits clearly justifies the area penalty.

4.5 DVS implementation of Asynchronous circuits

In this section, the above discussed static voltage scaling techniques that are commonly applied at gate level is adopted for asynchronous circuits from synchronous design methodology. The energy of the asynchronous circuits is reduced by performing a dual voltage assignment to each asynchronous standard cell such that the overall delay from input to output is balanced. The CVS, ECVS and GECVS techniques that are targeted for NCL type circuits are described in detail and the results are compared across all the three methods.
4.5.1 Embedded Level Converters NCL version (Emb-LC)

In a multi-Vdd or DVS circuit using LC with its design, the LC cell sensitivity plays a major role in steering the energy-delay optimization problem towards global optimum. Although LC’s are inevitable in DVS design, penalties (delay and energy) imposed by LC’s needs to be curtailed. In an ASIC design environment, LC can be added at either the input of the HV cell at the boundary between HV and LV cell (we will call it LC-gate) or, at the output of LV cell (called gate-LC) as shown in the fig. 4.22. [23] proves that both these typical scenarios that arise are delay and energy non-optimal. In fact, the gate-LC combination is much slower than the LC-gate because of the two-level logic represented by these combinations. A better choice would be the one that work as level converting gate which reduces the delay and energy penalty imposed by the gate-LC and LC-gate combination. We call such a gate Embedded-Level-Converters (EmbLC). EmbLC is shown in fig. 4.22(c).

For NCL-type circuits, EmbLC cells can be easily realized with Differential logic-NCL (DNCL) [24] type gates. DNCL gate architecture basically resembles the Differential Cascode Voltage Switch Logic (DCVSL). DCVSL logic produces both the function and its complement making it desirable for dual-rail logic such as NCL. The basic DVSL logic is represented in the fig. 4.23. The two branches of the DCVSL logic implement the PDN with NMOS logic of its function and complement, while the
Figure 4.23: Basic structure of DCVS logic gates.

Figure 4.24: General architecture of DNCL.

PUN is implemented with two cross-coupled PMOS transistors. This has the merits of reduced area and faster gate than a CMOS realization.

NCL circuits are special type of threshold gates with state holding element providing the hysteresis. With a slight modification, the DCVSL logic can be easily molded to represent a DNCL gate that can be used as an embedded-level converter. This requires the addition of two extra NMOS transistors along with the cross coupled PMOS network. The addition of NMOS represents a cross coupled inverter providing the necessary state holding characteristics of NCL gates. Either branches of DNCL
can never be asserted at the same time thereby providing the necessary isolation between the two voltage domains which are the necessary characteristics of a LC. Thus, in this design, the DNCL gates are used as embedded-LC gate that perform the required functionality of the NCL gate and at the same time working as a level shifter at the periphery between dual voltages. The general architecture of modified DCVSL - DNCL type gate is shown in fig. 4.24.

The set branch can be implemented by the uncomplemented set function of the NCL gate while, the reset branch can be implemented by complemented NMOS series chain. In addition, our outputs are taken from the \( \overline{Z} \) branch by adding an inverter at the output. In addition to providing the drive needed for the FO gate it has an added advantage of isolating the level shifter stage from the high fanout output that can load the level shifter stage and considerably increase the propagation delay [25]. The DNCL-thand0 or Emb-thand0 NCL gate, as it is called in this work is shown in fig. 4.25.

The following delay-energy plot 4.26 compares the gate-LC version of thand0 gate with the LV-thand0 gate being connected with a conventional level converter [23] at its output against the EmbLC-thand0 gate. The LC in the thand0-LC gate uses
Figure 4.26: Plot of Energy Vs Delay for THand0 with CCLC and THand0-EmbLC.

Figure 4.27: Conventional level shifter.
the standalone traditional level shifter, shown in fig. 4.27, often used in the DV environment.

The plot 4.26 clearly shows that the embedded level converter outperforms the thand0-LC gate in both the delay and energy metric. For 40fJ energy expended by both the gates, the EmbLC cell is roughly 70% faster than gate-LC arrangement. This is because the gate-LC arrangement makes them a two-stage gate operation increasing the delay [25]. The appreciable gains in both energy and delay make them ideal candidates for DVS design with LC. Further, the Table 4.3 compares the gate-LC configuration with an EmbLC configuration in terms of transistor count and shows the superiority of EmbLC over gate-LC in terms of area thereby mitigating the cost of level converter usage in DVS [25].

In this research work, instead of a standalone LC, an embedded type level converter is used when the DVS algorithms such as ECVS and GECVS calls for a LC.

4.5.2 DVS design CAD implementation flow

The proposed DVS algorithms were implemented and the design implementation using the standard commercial-off-the-shelf synchronous tools for asynchronous design is discussed in detail here along with experiments performed to evaluate the effectiveness of DVS on asynchronous threshold network circuits like NCL.

Objective: Even though there has been significant advancement in asynchronous design like the DI microprocessors [55] or the NCL ADC [18], it turns out that this promising design methodology is stalled by the lack of CAD tools [22]. Though there are various CAD tools developed [21, 20, 19] they still suffer from a few shortcomings and very specific to some asynchronous design methodology (there are no specific CAD tool for NCL design methodology) and this may sometimes require designers to get acquainted with the tool prior to design. Some tools may not be as optimized or efficient as the commercially available tool for synchronous design [22]. To overcome
Table 4.3: Transistor overhead of gate-LC combination compared with Emb-LC.

<table>
<thead>
<tr>
<th>NCL gates</th>
<th>Boolean Function</th>
<th>Emb-cells</th>
<th>gates+CCLC</th>
<th>transistor savings</th>
</tr>
</thead>
<tbody>
<tr>
<td>TH12</td>
<td>A+B</td>
<td>12</td>
<td>14</td>
<td>2</td>
</tr>
<tr>
<td>TH22</td>
<td>AB</td>
<td>14</td>
<td>20</td>
<td>6</td>
</tr>
<tr>
<td>TH13</td>
<td>A+B+C</td>
<td>14</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>TH23</td>
<td>AB+AC+BC</td>
<td>20</td>
<td>26</td>
<td>6</td>
</tr>
<tr>
<td>TH33</td>
<td>ABC</td>
<td>18</td>
<td>24</td>
<td>6</td>
</tr>
<tr>
<td>TH23w2</td>
<td>A+BC</td>
<td>18</td>
<td>22</td>
<td>4</td>
</tr>
<tr>
<td>TH33w2</td>
<td>AB+AC</td>
<td>18</td>
<td>22</td>
<td>4</td>
</tr>
<tr>
<td>TH14</td>
<td>A+B+C+D</td>
<td>22</td>
<td>18</td>
<td>-4</td>
</tr>
<tr>
<td>TH24</td>
<td>AB+AC+AD+BC+BD+CD</td>
<td>26</td>
<td>34</td>
<td>8</td>
</tr>
<tr>
<td>TH34</td>
<td>ABC+ABD+ACD+BCD</td>
<td>26</td>
<td>34</td>
<td>8</td>
</tr>
<tr>
<td>TH44</td>
<td>ABCD</td>
<td>22</td>
<td>28</td>
<td>6</td>
</tr>
<tr>
<td>TH24w2</td>
<td>A+BC+BD+CD</td>
<td>24</td>
<td>28</td>
<td>4</td>
</tr>
<tr>
<td>TH34w2</td>
<td>AB+AC+AD+BCD</td>
<td>25</td>
<td>30</td>
<td>5</td>
</tr>
<tr>
<td>TH44w2</td>
<td>ABC+ABD+ACD</td>
<td>25</td>
<td>32</td>
<td>7</td>
</tr>
<tr>
<td>TH34w3</td>
<td>A+BCD</td>
<td>22</td>
<td>26</td>
<td>4</td>
</tr>
<tr>
<td>TH44w3</td>
<td>AB+AC+AD</td>
<td>22</td>
<td>24</td>
<td>2</td>
</tr>
<tr>
<td>TH24w22</td>
<td>A+B+CD</td>
<td>22</td>
<td>24</td>
<td>2</td>
</tr>
<tr>
<td>TH34w22</td>
<td>AB+AC+AD+BC+BD</td>
<td>24</td>
<td>24</td>
<td>0</td>
</tr>
<tr>
<td>TH44w22</td>
<td>AB+ACD+BCD</td>
<td>25</td>
<td>30</td>
<td>5</td>
</tr>
<tr>
<td>TH54w22</td>
<td>ABC+ABD</td>
<td>22</td>
<td>26</td>
<td>4</td>
</tr>
<tr>
<td>TH54w32</td>
<td>A+BC+BD</td>
<td>22</td>
<td>25</td>
<td>3</td>
</tr>
<tr>
<td>TH54w32</td>
<td>AB+ACD</td>
<td>22</td>
<td>28</td>
<td>6</td>
</tr>
<tr>
<td>TH44w32</td>
<td>AB+AC+AD+BC</td>
<td>24</td>
<td>28</td>
<td>4</td>
</tr>
<tr>
<td>TH54w322</td>
<td>AB+AC+BCD</td>
<td>24</td>
<td>29</td>
<td>5</td>
</tr>
<tr>
<td>THxor0</td>
<td>AB+CD</td>
<td>22</td>
<td>28</td>
<td>6</td>
</tr>
<tr>
<td>THand0</td>
<td>AB+BC+AD</td>
<td>23</td>
<td>30</td>
<td>7</td>
</tr>
<tr>
<td>TH24comp</td>
<td>AC+BC+AD+BD</td>
<td>22</td>
<td>26</td>
<td>4</td>
</tr>
</tbody>
</table>
Figure 4.28: Existing design flow with commercially available synchronous tools [15, 1].

these problems, the objective of this research work is to have a complete design flow for asynchronous threshold network circuits replicating the traditional design flow available for synchronous ASIC design. The entire design flow completely automated from design entry in high level language like VHDL/verilog to layout incorporating the energy optimization techniques like DVS and GS consists of commercially available off-the-shelf (COTS) synchronous design tools from Cadence™ and synopsys™.

4.5.3 Existing design flow

The currently available design flow is shown in fig. 4.28. The design entry is done in high level language such as VHDL or Verilog. Since the asynchronous design incorporates both the data and the control circuit together, minor changes have to be incorporated in the HDL description. The HDL description has to account for [15]:

- NULL-DATA behavior which is accounted for with a creation of special package ‘ncl_logic’ similar to ‘std_logic’ package. The ‘ncl_logic’ package is IEEE 1076/1164 compliant and accounts for the multi-value logic DATA (0, 1) and NULL (N). Similarly, other packages such as ‘ncl_signed’, ‘ncl_unsigned’ and ‘ncl_arith’ are created.
- Hysteresis behavior of NCL gates - it is inherent property of every NCL gate and cannot be synthesized. It is written as procedure in ‘ncl\_logic’ and used only for simulation.

- Completion detection circuits with request and acknowledge signals - In NCL circuits, the registration and the CD circuits are implemented with NCL gates and hence, CD circuits can be component instantiated in separate file or manually instantiated in a process statement [15].

Before the design synthesis, the design is verified for its functionality with any of the synchronous simulators such as Cadence\textsuperscript{TM} NC-verilog or NC-VHDL or Mentor Graphics\textsuperscript{TM} Modelsim. With CD, hysteresis and multi-value logic accounted for in the ‘ncl\_logic’ package, the Boolean gates in 3NCL can be simulated with traditional synchronous simulation tools.

Following the functional simulation, the next step is the translation of the behavioral VHDL/Verilog code in ‘ncl\_logic’ to map to the components in the library. The commercial tool - synopsys DC\textsuperscript{TM} or DC shell is used for the translation. The NULL value during synthesis is treated as ‘don’t-care’ and this enables the commercial tool to treat the DATA values (0, 1) as single wire.

Hysteresis can be conveniently neglected as it is inherent property of every NCL gate. The initial library targeted is the synopsys GTECH (Generic TECHNOlogy) library. The code is optimized and mapped to the generic components in the GTECH library. The result of the mapping is the translation of the behavioral code to generic Boolean components that are still in 3-value logic and hence the netlist obtained is called the 3NCL netlist. The wires though are single wires they carry 3-valued logic (‘0’, ‘1’, ‘N’) [15].

**3NCL to 2NCL translation:** This step is the conversion of the 3NCL to the dual rail equivalents, the wire are dual-rail and the gates are replaced by their NCL
<table>
<thead>
<tr>
<th>Basic gates</th>
<th>DIMS Expansion</th>
<th>NCL Expansion</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Diagram" /></td>
<td><img src="image2" alt="Diagram" /></td>
<td><img src="image3" alt="Diagram" /></td>
</tr>
</tbody>
</table>

Figure 4.29: Macro Expansion of 3NCL to 2NCL in DIMS and NCL-opt Style.

threshold gate equivalents [15, 1]. The translation done in the first step is a single rail structural netlist obtained from high level synthesis by commercially available tools like synopsys DC™ compiler. These single rail netlist are expanded into its dual rail equivalents and mapped to the cells in the NCL threshold library cells. The .db library (ncl.db) is identical to any .lib library in synopsys™ liberty format that can be used with DC shell, which contains the gate functionality, timing parameters and the wire load models [15].

The dual rail expansion from 3NCL to 2NCL or the translation can be done in various methods; each method of expansion affects the area in some way. The wires are expanded into dual rail with the dual rail package that is read in with the intermediate 3NCL netlist in the synopsys™ DC. The cells in 3NCL can be macro expanded tile-by-tile with its dual rail counterparts. The complete list of two input gates along with its dual rail counterparts are shown in fig. 4.29. Note, there are two common styles of dual rail expansion of the 3NCL. The Delay-Insensitive Minterm Synthesis (DIMS)
[51, 46, 47] substitutes every minterm of the 3NCL synthesis with its 2NCL macros with a network of TH22 and TH12 gates. In contrast, the NCL-optimized style TCR optimization [47] is applied and each rail is mapped to a complex NCL gate that optimizes the minterm. Both the styles are timing robust NCL implementation that are gate and wire orphan free [51, 2].

4.5.4 Proposed Post Synthesis CAD design flow

This step is involved with providing a link between the physical domain and logical domain. For the post synthesis physical design approach, standard cell methodology is adopted for NCL IC design layout. After the 2NCL synthesis of the design obtained from the existing CAD tool flow with COTS for NCL, we obtain a structural gate level netlist that is ready for physical synthesis. The post synthesis design flow involves the CAD design flow to produce mask-ready, design rule error free NCL IC design layout in GDSII format. A semi-custom approach is undertaken in this work to produce good quality layout of standard cells required for NCL design library. Again, commercially available off-the-shelf synchronous CAD tools will be used for the physical layout of the asynchronous design. Preferably, Cadence™ Encounter will be used for Place and Route (P&R).

Although there are numerous VS and GS algorithms [61, 60, 63, 62, 64, 7] described in the literature claiming huge energy saving at the gate level, DV design at the physical design level faces numerous complications over the single voltage design and thus, requires design changes[3]. Particularly, there are physical design issues during P&R relating to the bulk of the LV and HV cells. For a DV design, a partitioning algorithm would produce two clusters of HV and LV cells and the clusters can be P&R in voltage island style[3]. The other types of layout architecture can be found in [62].

When a traditionally used row-based placement of standard cells with automatic
Figure 4.30: DV realization at physical design level (a) HV and LV well voltages at different potential, (b) HV and LV well cell voltages at HV voltage.

P&R tools are used, we are faced with two situations as shown in fig. 4.30. A N-well process is considered here. In the first case, fig. 4.30 (a), HV and LV cell well voltages and PMOS transistors operate at same voltage level, the issue with this type of arrangement is that the N-well at different potential requires bulk-separation which leads to a poor layout density and increasing the interconnect lengths in some cases [3, 5]. In the second case, fig. 4.30 (b), the HV and LV cell share the well, i.e the LV cell N-well voltage is at HV and this produces an area-efficient layout compared to the previous case.

In this work, we use shared well approach [5], in which the LV NCL standard cell’s N-well is connected to HV voltage. The LV power rail runs all along the width of the standard cell with a minimum metal spacing between the HV and LV power rail. The cell shares a common ground rail fixed at the bottom of the cell.

The complete list of all the 27 NCL cells (static, semi-static and embedded-LC versions) that constitute the NCL standard cell library was redesigned with our silicon generator to suit the DV physical design environment. The silicon generator and the list of all the NCL cells are described in the next section. Fig. 4.31 shows a static
Figure 4.31: LV static-TH22 gate for 130nm process.

Figure 4.32: BM circuit dalu P&R using Cadence™ Encounter.
TH22 LV standard cell for 130nm process, generated by the silicon generator tool, with its N-well connection and the extra HV power rail. This arrangement aids the HV and LV cell to be placed abutting each other sharing the common well and the power rails running all along the rows. Now, any commercially available P&R tool can be used to place and route the design leading to design automation at the physical level. Fig. 4.32 shows the BM circuit  

placed and routed with Cadence™Encounter tool in a DV environment. Observe, the dual power rail running along the rows in alternate fashion and the power rings of HV, LV and GND around the core, with HV and LV cells placed in the same row.

The advantages of using such a design is:

1. The commercially existing automatic P&R tools can be used to place the HV and LV clusters, leading to design automation with a slight cell modification.

2. It produces an area efficient good layout with minimum wire congestion [6].

3. A cost effective solution which address the major DV design concern at physical layout level.

Drawbacks are addressed as follows:

1. The cell alteration leads to area overhead, where every cell has to be modified to accommodate the extra power rail. The HV and LV power rails are separated by minimum metal spacing rule and in our design; the area overhead is approximately 11%.

2. Connecting the N-well of the LV to HV, body effect [66] comes into play and hence increases the $V_{TH}$ of the PMOS, thereby increasing the delay of the LV cell [5]. The increase in delay for the modified dual-rail NCL cells over the single rail NCL cell is about 22% in a 130nm process.
3. [5] notes that the drive current reduces, resulting in decreased drive strength of the cell.

4.5.5 Asynchronous Standard Cell Generator – Silicon Generator

The most prevalent physical design approach for ASIC’s still relies on standard cell place and route approach to produce reliable good quality layouts. But, unlike its synchronous counterparts, NCL design library cells for the targeted technology are not available from vendors. To generate good quality layout which are design-error-free standard cell layouts, it is imperative to automate the layout generation. In addition, to improve the design cycle time for creating standard cell library for rapidly changing technology processes, an automated cell library generator reduces the burden of design time at physical layout level. Automatic cell library generator has the added advantage of producing a flexible standard cell library with different drive strength providing the designer with good number of options in a performance driven or area constrained physical design environment. To lend support for automatic P&R and create error free asynchronous NCL standard cell library an automated standard cell generator -silicon generator (SG) is developed.

Our design flow involves a novel semi-custom method for the post synthesis design flow. An automated asynchronous standard leaf cell generator and cell library generator was developed that are ready to be placed and routed.

The SG is a technology independent standard cell generator tool. As new technology emerges every year, the technology parameters also changes along with it. For a TSMC 0.18µm technology with a gate length of 0.18µm the minimum spacing between the two metal1 (M1) rails is 270µm, whereas in IBM 0.13µm technology it is 160µm. Then, the standard cells would have to be redesigned to suit that particular technology to save design time and cost involved with redesign, we have developed an automated standard cell generator that is fully technology independent. The tool
was written in C++, which makes it completely CAD tool independent and the parameters could be quickly changed and the standard cell can be created to suit the targeted technology. The following summarizes the key advantages of our generator tool:

- Technology independent
- CAD tool independent
- Reduces design time
- Quickly generate standard cells layouts for the target technology with different drive strength.
- Quickly adapts to emerging technologies
- Cost effective

The fully automated CMOS standard cell generator creates not only 27 standard asynchronous leaf cells in the NCL library [47], but can also create filler cells, substrate taps and some complex cells like NCL-XOR, FA, HA which are area efficient. The tool has the flexibility of adapting and generating any style of asynchronous NCL cells to suit the designers need. At present it can accommodate and instantly generate static, semi-static, dynamic and DNCL varieties of NCL standard cells. Additionally, the tool offers components required for DV scheme such as NCL library with extra voltage rail and level converters (LC) in different topologies.

The design rules which impose the geometric constraints on the layout ensures the correct fabrication and functionality of the cell. Often, these rules can either be vendor and process specific called the “vendor rules” or it could be more generic, process and metric independent rules like the Mead and Conway scalable rules expressed in the abstract metric of lambda, $\lambda$ [66].
Our tool has the ability to support all the three set of MOSIS’ [49] rules from a conservative DEEP rules to aggressive ones like SUBM and SCMOS [49]. The process feature sizes, separation and overlaps are all expressed in $\lambda$ [66]. This generator uses $\lambda$ -rule for its geometric constraints.

Our tool follows a conservative 2D “matrix layout” style, where the layout area is gridded into rows (Y-direction) and columns (X-direction) and the transistors, I/O ports and the intra-routing connections are made by specifying the grid co-ordinates (X, Y). The initial user-defined control parameters that dictate the cell layout area are:

1. Y-pitch (YP) which defines the cell height, this parameter is multiple of the underlying process technology. This input parameter defines the cell height from the ground rail to $V_{dd}$ rail. The row spacings (Y-grid spacing) are then automatically calculated to be multiples of $\lambda$. The cell height has to accommodate the (P and N) transistors’ width, the routing area between them and the power rail widths.

2. The next input parameter is the height of the p-substrate (PSUBH). Assuming, N-well process, the height of the N-well is automatically calculated from YP and PSUBH. N-well height (NSUBH = YP - PSUBH).

3. Power width (PW), this variable defines the width of the power rails, (VDD and GND). The supply rails run at the top and the bottom of the cells which also hosts the substrate contacts for the standard cells. Although, the rails run horizontally all along the cell width, the tool has the ability to run the rails vertically depending on the design needs. The tool also has the ability to modify the leaf cells for dual-voltage cells in variety of arrangements where the cell can have two VDD’s or two GNDs [13]. In this work with DVS, we have DV cells created with two voltage rails (HV and LV) running all along the width of
standard cell and the common GND line at the bottom of the cell.

The other input files consists of targeted technology process file which consists of complete description of all the widths and/or spacing rules for all the mask layers, description of the cell layout in different topologies in a high level language like C++. The file contains details about the cell design parameter variables such as transistors’ width W, the length L, cell pitch, power routing metal widths and N/P-well widths which are completely parameterizable depending upon the leaf cell requirements.

The fig. 4.33 shows the layout generated by the automated silicon generator following the above strategy which is design-rule correct and without human intervention.

The tool has the ability to individually size the transistor in the leaf cell, although for static version of standard cells a P-N ratio of 2:1 is followed. For the semi-static versions, the PUN and PDN are carefully sized because of the weak-feedback inverter.

The versatile tool also allows a designer to rotate the transistor according to the design requirements. 90, 180 and 270 degree rotations are supported by the tool. A 90 degree rotated transistor is used in the design of our semi-static cell layouts. The tool offers the designers the option of creating the standard cells for either N-well or P-well process.
The versatile, yet sophisticated tool has the features that a commercially available tool can offer. It is furnished with transistor folding techniques to reduce the cell height and renders all angle rotation of transistors, contacts and cells. Granting the designers a variety of other advantages like well or substrate tap contact insertion, this tool provides good tie coverage. The IO port are placed on the routing grid for compatibility with other P&R tools. All these features makes this tool even more powerful.

4.5.6 Generator Implementation

The proposed tool was implemented in C++ and the cells were generated on SUN SPARC workstation with 1152MB RAM running SunOS 5.1. The tool takes in the layout topology along with the technology process design rules and outputs a compacted design rule correct cell layouts in standard mask layout interchange format (GDSII).

It also generates timing and power parameters in the industry standard Synopsys\textsuperscript{TM} Liberty format-LIB file. The LEF describes the cells physical attributes including cell area, port location and type. The other process LEF files which usually has the mask layer and via definitions are regularly provided by the foundry.

A complete list of the NCL leaf cells which forms the NCL library has 27 cells [47] with distinct cell functions and variety of drive strength is created by our tool. Currently, for GS, our tool generates 2x, 3x, and 4x drive strength cells, although higher drive strength cells can be generated depending on the design requirement. Table 4.4 lists the complete list of NCL cell library in both the static and semi-static topology along with number of transistors. Note that the T1n cells are simple Boolean OR gates and do not have the sequential functionality unlike the other cells. The general layout rules that were followed for producing the cell layouts using our tool is listed in Table 4.5.
Table 4.4: NCL library transistor count for static and semi-static version [47].

<table>
<thead>
<tr>
<th>NCL gates</th>
<th>Boolean Function</th>
<th>Transistor Count</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Static</td>
</tr>
<tr>
<td>TH12</td>
<td>A+B</td>
<td>6</td>
</tr>
<tr>
<td>TH22</td>
<td>AB</td>
<td>12</td>
</tr>
<tr>
<td>TH13</td>
<td>A+B+C</td>
<td>8</td>
</tr>
<tr>
<td>TH23</td>
<td>AB+AC+BC</td>
<td>18</td>
</tr>
<tr>
<td>TH33</td>
<td>ABC</td>
<td>16</td>
</tr>
<tr>
<td>TH23w2</td>
<td>A+BC</td>
<td>14</td>
</tr>
<tr>
<td>TH33w2</td>
<td>AB+AC</td>
<td>14</td>
</tr>
<tr>
<td>TH14</td>
<td>A+B+C+D</td>
<td>10</td>
</tr>
<tr>
<td>TH24</td>
<td>AB+AC+AD+BC+BD+CD</td>
<td>26</td>
</tr>
<tr>
<td>TH34</td>
<td>ABC+ABD+ACD+BCD</td>
<td>26</td>
</tr>
<tr>
<td>TH44</td>
<td>ABCD</td>
<td>20</td>
</tr>
<tr>
<td>TH24w2</td>
<td>A+BC+BD+CD</td>
<td>20</td>
</tr>
<tr>
<td>TH34w2</td>
<td>AB+AC+AD+BC+BD</td>
<td>22</td>
</tr>
<tr>
<td>TH44w2</td>
<td>ABC+ABD+ACD</td>
<td>24</td>
</tr>
<tr>
<td>TH34w3</td>
<td>A+BCD</td>
<td>18</td>
</tr>
<tr>
<td>TH44w3</td>
<td>AB+AC+AD</td>
<td>16</td>
</tr>
<tr>
<td>TH24w22</td>
<td>A+B+CD</td>
<td>16</td>
</tr>
<tr>
<td>TH34w22</td>
<td>AB+AC+AD+BC+BD</td>
<td>16</td>
</tr>
<tr>
<td>TH44w22</td>
<td>AB+ACD+BCD</td>
<td>22</td>
</tr>
<tr>
<td>TH54w22</td>
<td>ABC+ABD</td>
<td>18</td>
</tr>
<tr>
<td>TH34w32</td>
<td>A+BC+BD</td>
<td>17</td>
</tr>
<tr>
<td>TH54w32</td>
<td>AB+ACD</td>
<td>20</td>
</tr>
<tr>
<td>TH44w322</td>
<td>AB+AC+AD+BC</td>
<td>20</td>
</tr>
<tr>
<td>TH54w322</td>
<td>AB+AC+BCD</td>
<td>21</td>
</tr>
<tr>
<td>THxor0</td>
<td>AB+CD</td>
<td>20</td>
</tr>
<tr>
<td>THand0</td>
<td>AB+BC+AD</td>
<td>22</td>
</tr>
<tr>
<td>TH24comp</td>
<td>AC+BC+AD+BD</td>
<td>18</td>
</tr>
</tbody>
</table>

Table 4.5: General layout rules used by silicon generator for creating a NCL library in 130nm process.

<table>
<thead>
<tr>
<th>Cell layout Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cell Height</td>
<td>$110\lambda$</td>
</tr>
<tr>
<td>Cell Width</td>
<td>Multiple of $\lambda$</td>
</tr>
<tr>
<td>Gate Length</td>
<td>$2\lambda, 8\lambda$ for Weak Inverters</td>
</tr>
<tr>
<td>Power Width</td>
<td>$8\lambda$</td>
</tr>
</tbody>
</table>
A 130-nm twin-well CMOS technology was chosen for demonstration to produce the cell library. The transistor widths used for static version have a ratio of 2:1 for PUN/PDN to have almost equal rise time and fall time for the cell. The semi-static gates have been carefully sized for same condition with a weak-inverter having a gate length of $8\lambda$. The cells were imported into Cadence™Virtuoso for DRC; DIVA was used for parasitic extraction and simulated using HSPICE.

The .LIB file contains the cells library characterization information. We used a non-linear delay model and made use of lookup table format. Each cell’s timing information comprises of rise and fall times along with the signal transition times for each input-to-output path. The cell timing information is provided in a two-dimensional array.

In the LEF file, every cell has its cell physical attributes such as area and pin locations described which are easily derived from the cell layout. Our tool IO ports have M1-M2 contacts which enable cell routing with M2 layer and above. The OBS layer completely blocks the usage of M1 in the area between the cell power rails.

In conclusion, although SG creates threshold network asynchronous circuits cell library, it can easily adapt to and create cell libraries for any new emerging or existing asynchronous methodologies. It is flexible enough to mould itself to any new process technologies and wide range of layout topologies can be generated thereby demon-
Figure 4.35: NCL - static TH23 gate generated by Silicon Generator.

Figure 4.36: TH22 semi-static standard cell layout from silicon generator.
strating the effectiveness of the tool. Our tool is completely technology and platform independent tool thereby achieving foundry independence for cell library creation. The tool provides the designer with variety of options to choose while creating the library, proving its worthiness with quick turn-around time.

Its generated cells can be easily placed and routed using standard commercial synchronous tools eliminating the need for separate asynchronous tools proving to be a cost-effective method for physical synthesis. Having unrestricted circuit structure it proves its robustness by creating complex NCL type cells such as NCL-FA, HA and more with quick runtime in an area efficient manner. One such complex cell, a NCL-FA created is shown in fig. 4.34. The tool also has the capability of assisting the designers with creating guard-rings and can be tailored to one’s need.

Fig. 4.35 shows the TH23 static cell in 130nm and 180nm and fig. 4.36 (a) shows the semi-static version of TH22 cell in 180nm process and (b) shows the dual voltage TH22 static cell in 130nm. This tool has also demonstrated that it is invaluable for creating a design-rule-error-free new library quickly as we migrate to new process technology.

And finally, this work which concentrates on the physical synthesis (layout generation) of the NCL design which fits perfectly well with the existing NCL design flow [15]. This completes the design flow turning the design entry in high level language such as VHDL/Verilog into a manufacturable database.

The key challenge in the design flow has been addressed with a typical synchronous ASIC physical synthesis design flow utilizing industry standard synchronous CAD tools at every step of the design flow. The complete design flow is shown in fig. 4.37. The 2NCL structural netlist along with the LEF files that is generated by the silicon generator are inputs to the standard P&R tool. The placed and routed design result is written to Design Exchange Format (DEF) file that is ready for parasitic extraction and post P&R simulation.
Figure 4.37: A complete design using COTS.

Figure 4.38: Snapshot of 8-bit Asynchronous transceiver chip.
Recently, we designed a 16-point 8-bit Synchronous and Asynchronous FFT processor in 130nm process using the cell generated by our tool and physical synthesis done with Cadence Encounter. The snapshot of the P&R design is shown in fig. 4.38.

### 4.6 DVS - Experiments and Results

In this section, the DV algorithm implementation details and experimental setup are described. Finally, the results of applying the DVS scheme for MCNC benchmark (BM) are discussed. The experiments were conducted on the combinational MCNC BM circuits.

The process technology targeted in this research work is a 130nm twin well CMOS technology. The HV voltage (VDDH) considered is 1.2v and the LV voltage (VDDL) is 0.8v. Throughout this work, the threshold voltage is assumed to be constant, despite dual threshold voltages scheme being commonly used along with VS or GS techniques.

With technology scaling, the routing density increases and the previously neglected wire parasitics can no longer be ignored. Wire parasitics in the form of wire capacitances (C) and resistances (R) contribute to delay in the circuit. The parasitic RC values were obtained by back-annotation from the initial placement of the NCL netlist in Cadence™ Encounter. The command `extractRC` extracts the RC values of interconnects in the design. Using a worst-case tree model for wire delays, the delay of wire \(x\) is given as,

\[
\text{wireDelay}_x = R_{\text{wire}} \times \left( \frac{C_{\text{wire}}}{2} + \sum_{F_{O_x}} C_{\text{pin\_cap}} \right) \quad (4.7)
\]

where \(R_{\text{wire}}\) is resistance of the entire net along with net capacitance \(C_{\text{wire}}\) and gate input pin capacitance \(C_{\text{pin\_cap}}\). It is also assumed that the \(C_{\text{wire}}\) remains unaffected when GS is employed. The input pin capacitance of each gate was measured from HSPICE while the output load capacitance was calculated as the sum of all input
capacitances of a gate's FO.

In order to model the delay of the gates which accounts for the delay penalty, a standard cell library consisting of NCL threshold gates (including the embedded NCL gates) with different drive strengths was characterized for delay using HSPICE. Delay modelling of the gates using HSPICE was preferred over the usual analytical method of accounting for delay. The major advantages that HSPICE pose over the analytical method are:

1. Quick and accurate delay model.

2. The commonly used analytical expression, delay being inversely proportional to supply voltage, is mostly valid for single voltage supply design and it loses its accuracy when used in dual voltage environment.[31, 7].

The complex NCL gates were simulated using HSPICE to obtain a comprehensive (.lib) file that contained all the timing information along with the gate input capacitance. The timing information characterized includes cell\_rise and cell\_fall that accounts for propagation delay of the cell and rise\_transition and fall\_transition provides the rise and fall time at the cell output. Following a synopsys\textsuperscript{TM} liberty format and accounting for a non-linear-delay-model, a look-up table of two-dimensional array with five different output load capacitance at the gate with five different input slew rate was created to provide information on propagation delay and transition time parameters. Note: the .lib file that was characterized included all the HV and LV NCL cells along with the embedded-type NCL gates. While characterizing the LV cells, a LV signal was used which reports the worst case delay, although in voltage-scaled environment often, the LV cells may be driven by a HV cell.

A CAD tool was built that incorporated the DVS algorithms. The entire implementation of the DVS algorithms was done in C++. To evaluate the effectiveness of the DVS algorithms, the CVS, ECVS and GECVS were applied to set of MCNC
BM circuits, which were randomly chosen from a set of suggested larger circuits. The MCNC BM circuits were in the BLIF format and needed to be preprocessed to suit our work for asynchronous NCL design. A two-step synthesis of MCNC benchmark circuit was performed converting the BLIF format circuit to its equivalent dual-rail NCL circuit in structural format.

1. **Synthesis to 3NCL:** In this step, a multilevel technology independent optimization of the MCNC BM circuits was performed by the SIS tool using the script file - *script_rugged* [2]. This step was followed by the *map* command that mapped the resulting circuit to what is considered as 3NCL Boolean netlist. This 3NCL netlist consists only primitive two-input Boolean gates (AND, OR, XOR, NOT, NAND and NOR).

2. **3NCL to 2NCL synthesis:** The second step involved converting the 3NCL netlist to 2NCL dual rail netlist in Verilog or VHDL format that is ready to be processed by the optimization CAD tool. For the conversion, a simple dual-rail converter tool written in C++ was built that macro expands every 3NCL gate and its corresponding wires to its equivalent dual-rail gates and wires in either the DIMS or NCL-optimized format. The final 2NCL output format is in either Verilog or VHDL structural netlist to be processed by our CAD tool. The synthesized design has all its cells mapped to smallest size (x0) from the NCL library and the gate voltages are all assigned HV. The timing information or the speed of the circuit can be extracted by performing a STA. Since DIMS and NCL-opt styles are *orphan*-free by construction [2] the dual-rail BM circuits can be assured that they are completely gate and wire-*orphan* free.

Before proceeding further with the DVS or GS implementations, the 2NCL structural netlist is verified for its functionality with any of the commercially available synchronous simulators. In this work, Cadence\textsuperscript{TM} NC-sim simulator was used for
Figure 4.39: Cadence\textsuperscript{TM} NC-sim graphical window showing simulation waveform of BM c3540.

functional verification. All the BM circuits (both for NCL-opt and DIMS styles) was tested with a set of random input vectors and cross verified for correct functionality with its corresponding single rail synchronous version of the BM circuit. The fig. 4.39 shows the waveform window of the NC-sim tool that was used to simulate a dual-rail NCL-optimized verilog netlist of c3540 BM circuit.

The synthesized and simulated designs are then ready to be processed by the CAD tool to apply a variety of VS and/or GS optimization techniques (discussed in previous sections) depending on the users request. The CAD tools takes in a complete structural verilog netlist along with the targeted technology library (.lib) file that contains the timing information of the gates used in the netlist and outputs a structural netlist in verilog format that has low energy and obeys timing requirements of the design. This netlist is now ready to be P&R by any commercial tool.

Two experiments were conducted. In the first experiment impact of the VS on asynchronous NCL circuits on both the NCL-opt style and DIMS style were studied
when LC was strictly prohibited in this design. That is, effectiveness of the CVS technique at gate level of the threshold network circuits was examined. In the second experiment a relaxed approach of using level converters in the design was studied with two techniques ECVS and GECVS. These algorithms used the embedded-LC when a LC was required in the design instead of using a stand-alone conventional LC. All the three techniques were applied to the BM circuits in both the DIMS and NCL-opt style architectures and the energy savings are tabulated in Table 4.6 for NCL-opt version and Table 4.7 for DIMS style. Columns 4, 6 and 9 represents the percentage of LV cells allocated using the three VS algorithms; the percentage of energy reduction obtained by applying CVS, ECVS and GECVS techniques are in columns 5, 8 and 11. Columns 7 and 10 indicates the percentage of LC usage in designs with LC (ECVS and GECVS).

The GECVS technique turned out to be a better VS option than the CVS or the ECVS and thus, only the GECVS technique was P&R using the Cadence Encounter, extracted and simulated with HSPICE to verify the energy reduction when GECVS was applied to NCL type circuits.

A quick observation reveals the fact that GECVS surpasses the performance of CVS and ECVS for all the BM circuits in both the NCL-opt and DIMS style of architecture. This is because; GECVS due its iterative nature of LV assignment forms good clusters of LV cells and effectively swaps excess slack in circuit for energy. In fact, CVS fails to assign any LC cells for c499 and c1355 in the both the NCL-opt and DIMS architectures. Further, it discovers only a single LV cell for symml9 and t481 designs for both the varieties demonstrating its poor performance due to its overly restrictive nature, despite having “potential” slack deeper in the circuit that can be harnessed for energy savings. Furthermore, CVS technique achieves a meagre 3% savings on average for NCL-opt style and 4% for DIMS style.

CVS reduces energy by approx. 7% for DIMS and NCL-opt versions of c1908
Table 4.6: Experimental Results: Voltage Scaling with CVS, ECVS and GECVS on NCL-opt Architecture.

<table>
<thead>
<tr>
<th>BM circuits</th>
<th>CVS</th>
<th>ECVS</th>
<th>GECVS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#i/#o</td>
<td>% LV cells</td>
<td>% Energy</td>
</tr>
<tr>
<td>9symml</td>
<td>18/2</td>
<td>0.27</td>
<td>0.03</td>
</tr>
<tr>
<td>ttt2</td>
<td>48/42</td>
<td>18.49</td>
<td>5.77</td>
</tr>
<tr>
<td>C499</td>
<td>82/64</td>
<td>16.13</td>
<td>7.14</td>
</tr>
<tr>
<td>C1908</td>
<td>66/50</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>C1355</td>
<td>82/64</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>t481</td>
<td>32/2</td>
<td>0.1</td>
<td>0.01</td>
</tr>
<tr>
<td>i8</td>
<td>266/162</td>
<td>13.99</td>
<td>5.14</td>
</tr>
<tr>
<td>dalu</td>
<td>150/32</td>
<td>13.87</td>
<td>5.19</td>
</tr>
<tr>
<td>C3540</td>
<td>100/44</td>
<td>9.76</td>
<td>4.18</td>
</tr>
<tr>
<td>Average percentage</td>
<td>8.07</td>
<td>3.05</td>
<td>32.82</td>
</tr>
</tbody>
</table>

Table 4.7: Experimental Results: Voltage Scaling with CVS, ECVS and GECVS on DIMS Architecture.

<table>
<thead>
<tr>
<th>BM circuits</th>
<th>CVS</th>
<th>ECVS</th>
<th>GECVS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#i/#o</td>
<td>% LV cells</td>
<td>% Energy</td>
</tr>
<tr>
<td>9symml</td>
<td>18/2</td>
<td>0.11</td>
<td>0.02</td>
</tr>
<tr>
<td>ttt2</td>
<td>48/42</td>
<td>20.56</td>
<td>6.83</td>
</tr>
<tr>
<td>C499</td>
<td>82/64</td>
<td>15.87</td>
<td>7.19</td>
</tr>
<tr>
<td>C1908</td>
<td>66/50</td>
<td>14.21</td>
<td>8.96</td>
</tr>
<tr>
<td>C1355</td>
<td>82/64</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>t481</td>
<td>32/2</td>
<td>0.04</td>
<td>0.01</td>
</tr>
<tr>
<td>i8</td>
<td>266/162</td>
<td>18.65</td>
<td>10.46</td>
</tr>
<tr>
<td>dalu</td>
<td>150/32</td>
<td>15.37</td>
<td>5.19</td>
</tr>
<tr>
<td>C3540</td>
<td>100/44</td>
<td>10.08</td>
<td>4.24</td>
</tr>
<tr>
<td>Average percentage</td>
<td>9.26</td>
<td>4.02</td>
<td>34.67</td>
</tr>
</tbody>
</table>
circuit and impressive 10% for BM i8 with quick runtime. On the other hand, the circuit in which CVS failed ECVS has notable performance with reducing the energy to almost 7% for c499 and c1355 NCL-opt versions. Allocating 45% of cells to LV, along with low 9% embLC reservation, ECVS reduces energy as much as 22% for c3540 NCL-opt circuit. Having the same run-time complexity as CVS, ECVS on average attains a good 15% energy reduction for NCL-opt style and 18% in DIMS version.

![Slack distribution for BM dalu](image)

Figure 4.40: Slack distribution for BM dalu

The plot of the slack distribution before and after applying the GECVS technique is shown in the figure 4.40 for the BM circuit dalu. The plot confirms the superiority of GECVS over the other two VS techniques where the slack in drastically reduced by prioritizing the assignment of LV to gates. In fact, the LV assignment in some circuit such as NCL-opt BM dalu has almost double the LV assignment over ECVS. In addition, GECVS almost achieved 25% better savings than CVS and 16% lower energy than ECVS for NCL-opt style. Moreover, for DIMS architecture GECVS outperformed CVS and ECVS by approximately 26% and 14% respectively. Some of the bigger circuits such as c3540, i8 and dalu achieved impressive savings of almost 40% for both the style of architecture when voltage was reduced to save energy at the
cost of polynomial runtime. Nevertheless, all the three techniques could not lead to significant power savings in BM c499 and c1355 circuits due to the balanced nature of the circuit [64].

To better appreciate the LV cell distribution amongst the three VS techniques, a plot of percentage of LV cells allocation is shown in fig. 4.41 for NCL-opt version of the BM circuit, whereas, the plot 4.44 compares the LV assignment for DIMS architecture. This plot reveals GECVS’s ability to discover more LV cells and it explains why GECVS outperforms CVS and ECVS. The plots 4.43 and 4.46 compares the energy savings achieved by applying the corresponding VS techniques. We also compared the usage of emb-cells for ECVS and GECVS techniques. GECVS consumed on average 17% of the total cells as Embedded-LC. A plot comparing the embedded-LC usage of ECVS and GECVS is shown in fig. 4.42 for NCL-opt version and in fig. 4.45 for the DIMS style.

The BM circuit layouts were simulated in HSPICE for set of random input vectors and the energy savings were compared against the calculated energy from our CAD tool. The simulated power and energy plot shown in fig. 4.47 for the BM circuit i8
Figure 4.42: Comparison of percentage of LV and LC cells after ECVS and GECVS for NCL-opt Architecture.

Figure 4.43: Comparison of percentage of energy reduction after CVS, ECVS and GECVS for NCL-opt Architecture.
Figure 4.44: Comparison of percentage of LV cells after CVS, ECVS and GECVS for DIMS Architecture.

Figure 4.45: Comparison of percentage of LV and LC cells after ECVS and GECVS for DIMS Architecture.
Figure 4.46: Comparison of percentage of energy reduction after CVS, ECVS and GECVS for DIMS Architecture.

Figure 4.47: Power and Energy plot for BM i8 before and after applying GECVS.
Figure 4.48: Comparison of percentage of Energy Reduction - HSPICE Vs. Analytical Reduction (NCL-opt) style).

(NCL-opt) indicates the energy savings obtained at the layout level when GECVS was applied.

The plot 4.48 and 4.49 compares the energy reduction at the gate level versus the energy reduction at the physical layout level obtained from HSPICE for NCL-opt and DIMS style respectively.

While the energy reduction as measured by HSPICE closely tracked that of our cost function, on average we obtained approx. 33% lower than those calculated by our cost function. This indicates that a more accurate model could be developed for these asynchronous circuits in order to reduce the error percentage between the design at gate level and the layout level. In the current model we assume that each async. gate switches two time for each DATA wave passing through it (i.e. NULL-DATA-NUL cycle). In reality, a dual-rail circuit having one gate per rail, only the rail that has been activated switches, indicating that the switching model has been over-estimated. This is in addition to the assumption that the gate input capacitance does not change when the gate voltage is changed. Together, they could be leveraged to improve the cost model used to drive the algorithm.
Figure 4.49: Comparison of percentage of Energy Reduction - HSPICE Vs. Analytical Reduction (DIMS) style.)
5 Gate Sizing for Energy Reduction in Asynchronous Circuits

In the previous chapter we discussed the techniques of achieving energy reduction by simply lowering the voltage at every gate, as long as the timing was preserved. It is widely accepted that the energy reduction in circuits can be accomplished by gate sizing (GS) techniques by reducing load capacitance [61, 28].

GS is another widely popular synchronous energy reduction technique that has been adopted for asynchronous design in this work. VS is a discrete optimization problem which means there are still pockets of slack which are un-utilized [60, 61]. In such cases a GS technique can be used to further reduce the energy in the circuit.

Gate resizing by choosing the discrete gates available in the cell library is a well prevalent technique in a standard cell based ASIC design methodology to compensate for the delay in delay constrained energy optimization environment. Often, in these situations, just applying VS would not be sufficient to meet the energy requirements of the design. In such cases, combinations of VS, GS or even a simultaneous VS+GS technique are exploited to further reduce the energy consumption.

After the VS step has exhausted all options to assign LV cell in the design obeying the timing constraint, a gate upsizing technique can be deployed to reduce energy in the circuit. when assigning any more LV cells to the LV cluster violates timing constraint, cells with different drive strengths from the ASIC library compensates for the delay penalty. As upsizing the gates increases the energy consumption in the circuit, the GS techniques would have to prioritize and choose the cell that would compensate for maximum delay with minimum energy penalty due to upsizing. Thus, these GS techniques tend be iterative in nature using a sensitivity factor to choose the right cell to compensate for delay without paying a huge energy price. This is the rationale behind employing GS after VS in a two-step energy reduction approach.
In this research work, GS techniques that are extension of CVS and GECVS are used. This chapter describes methods for reducing energy consumption in asynchronous circuit by applying these GS techniques after VS techniques have been applied.

5.1 CVS based Gate Sizing technique - GS-CVS

As discussed in the previous chapter, CVS technique is the simplest way to reduce energy at the gate level, but it fails to improve energy reduction in some circuits or performs poorly with suboptimal results. In designs with stringent rules on circuit topology to produce a LC free design, a HV cell always drives a LV cell thereby eliminating the need for a LC. The LV cells are clustered and restricted to cells near the PO. In such circumstances to achieve the energy goal of the circuit, techniques other than VS would have to be deployed. One such technique that is applied after the CVS failure is gate resizing. It works to create extra slack in the circuit and can be swapped for energy by continuing with CVS from the failure point. The details of GS-CVS algorithm is described below.

It is activated at the point where the CVS technique fails. In other words, the GS-CVS is started after the application of CVS breaks down. At this point, adding
any more cells to the LV cluster leads to a timing violation due to insufficient slack at the node. The cells that are on the boundary of the HV and the LV cells are now, said to be on a backward front [25]. The cells in the backward front have all the FO cells assigned to LV and converting any cell to LV causes a timing violation. An example of the backward front is shown in the fig. 5.50.

GS-CVS reduces energy in the circuit by pushing the envelope of LV cells towards the PI by assigning the cells in the backward pass to the LV and compensates for the delay penalty by sizing up the gates. Upsizing the gates reduces the delay and creates slack in the circuit. To explain in detail, a node on the backward front is first selected and assigned to LV. This selection process is done by a heuristic method which calculates the sum of the product of slack and the load capacitance available at each node in the FI cone. This predictive measure helps to identify the cell whose voltage can be scaled down and consequently leads to minimum timing violation. It also steers the GS-CVS in the right direction [25].

The timing violation can be fixed by upsizing a small percentage of the cells in the circuit. The selection for upsizing is done such that it would produce maximum compensation for the delay with low energy penalty as the gates are upsized. Therefore, a sensitivity measure which is the ratio of delay improvement to change in energy is calculated for every gate in the circuit and the gate with maximum sensitivity to this factor is resized to the next available size in the library. The sensitivity factor is given as follows [25]

\[
Sensitivity_x = \frac{1}{\Delta E} \sum \frac{\Delta D_x}{\text{slack}_x - \text{slack}_{min} + K}
\]

(5.8)

where \(\Delta E\) is the change in energy due to upsizing the node ‘x’ and the corresponding change in delay is represented by \(\Delta D_x\). The slack at the node ‘x’ is given by \(\text{slack}_x\) and the minimum slack in the circuit is \(\text{slack}_{min}\). To ensure stability \(K\) is required to be a small positive number.
Figure 5.51: Cost Function plot for BM circuit t481 employing GS technique - GS-CVS.

This sensitivity factor iteratively selects the gate that improves the performance of the circuit the most. The selected gate is replaced by the next higher size available in the cell library. The selection process and the cell replacement will continue until the timing is satisfied. Once the timing is satisfied, the current state is saved and the timing is updated for the circuit and the process is repeated with selection of another node from the backward front. The front is pushed deep in the circuit to grow the LV cluster thereby reducing energy.

It is important to note that there is limit on the number of such moves that can be performed in the circuit to avoid large run times. The cap on moves is set to 10% [25] of the total number of gates in the circuit. Additionally, this restriction prevents the circuit from pursuing bad moves that can increase the area of circuit and has negligible effect on power reduction.

The algorithm fails when the backward front cannot be pushed any further without violating timing even with exhaustive cell resizing moves. The algorithm also
Figure 5.52: Cost Function plot for BM circuit c1908 employing VS technique - CVS.

Figure 5.53: Cost Function plot for BM circuit c1908 employing GS technique - GS-CVS.
Figure 5.54: Cost Function plot for BM circuit c499 employing GS technique - GS-CVS.

Since this algorithm has hill-climbing property, the best state is saved and restored at the end of GS-CVS failure. This hill-climbing characteristic of GS-CVS allows some negative moves, which guides the algorithms out of the local minima to find better solution in the long run [25]. This is also the reason why applying GS after applying CVS produces huge energy savings in the design. The hill-climbing property of GS-CVS is demonstrated by tracking its cost function for BM circuit t481 implemented in NCL-opt style in the fig. 5.51. Figs. 5.52 and 5.53 compares the VS and GS-CVS respectively for a BM c1908 circuit. It clearly confirms the superior performance of GS-CVS over CVS and also shows the steepest-descent approach taken by CVS which is highly liable to get stuck in the local minima. The fig. 5.54 showcases the case when GS-CVS fails to produce any significant energy reduction in the circuit for the BM c499.
5.2 GECVS based Gate Sizing approach - GS-GECVS

This algorithm works on the premise that in a technology mapped circuit, after the VS technique such as GECVS failure to add LV cells to its LV cluster, more LV cells can be discovered by upsizing the cells in the circuit that compensates for performance degradation. GS-GECVS is a GS technique that is an extension of the VS technique GECVS. An overview of GS-approach is provided here and more detailed explanation along with the pseudo code can be found in [25].

The GS-GECVS technique is activated after GECVS fails to add any more cells to the LV cluster. At the end of GECVS, the nodes on the circuit have insufficient slack which cannot be exchanged for LV cells that causes the energy reduction. The plot of slack available in the circuit before and after the application of GECVS is shown in fig. 4.40. The plot explains why at the end of GECVS, any additional LV assignment leads to timing violations. The gates with slack almost zero have increased at the end of GECVS. With the exhaustion of slack in the circuit, a heuristic approach as described in [25] is used to increase the LV count in the circuit by applying gate resizing to recreate slack that accounts for extra delay in the circuit.

Extra LC, which is a regular feature of GECVS technique, is avoided to reduce the extra energy in addition to the increase in energy due to upsizing. In that case, a HV cell that has all of its FO as LV is chosen for LV assignment with the idea that the LV cluster can be grown and hence, reduces the energy in the circuit.
Figure 5.56: Cost Function plot for BM circuit ttt2 employing GS technique - GS-GECVS.

The candidates for LV assignment are then given preference with the GECVS selection criteria given in equation 4.6. To compensate for increase in delay, the sensitivity measure used in equation 5.8 is used to iteratively select the best gate for upsizing that steers the algorithm to quick convergence. Again, the number of upsizing moves that are allowed in GS-GECVS are restricted to 10% of the total number of gates in the circuit in order to avert the algorithm from producing bad moves that may lead to large area overhead and ultimately, eliminate the primary objective of energy reduction. When the number of upsizing moves to meet the timing exceeds the limit, the cell that was assigned LV is converted back to HV and all the gate sizes are reverted back to the starting size. Then the algorithm continues with the remaining candidates until the list is emptied.

As LV assignments are accepted without timing violation, more cells are added to the list and GS-GECVS continues until no more cell can be assigned to LV without
violating the timing even after reaching the upsizing limit. The fig. 5.56 shows the hill-climbing property of GS-GECVS technique which avoids the local minima and produces better results with large runtime. The complexity of GS-GECVS which incorporates GECVS is $O(V^3)$ [25].

### 5.3 Experiments and Results

To verify the effectiveness and capabilities of GS algorithms in energy reduction for asynchronous circuits, the GS-CVS and GS-GECVS algorithms were implemented under the same environment as that of VS algorithms. Again, two experiments were performed. The experiments were carried out on a set of MCNC benchmark circuits applying both the GS-CVS and GS-GECVS. The CAD tool optimizer written in C++ that performs these GS techniques takes in the technology mapped gate level netlist that have been mapped to the minimum size available in the library. The minimum size available in our NCL library was size - x0, that was appropriately sized for equal rise and fall delay characteristics.

The starting point of these GS algorithms is after the failure of their corresponding VS technique to discover LV cells without violating timing. Hence, the netlist to the CAD tool has dual voltage gates. Along with the gate netlist, the CAD tool is also fed a fully characterized dual voltage cell library with different drive strength that aids in gate resizing. The NCL library (.lib) file has 27 dual voltage cells (HV and LV) with drive strengths of x0, x1, x2 and x3 which are used in this work for GS and automatically generated by our versatile silicon generator. Although the drive strength of the cells have been limited to max. size of x4, the silicon generator can instantly create cells with higher drive strength and also with finer granularity depending upon the design and designers demands.

The GS technique, GS-CVS, is applied after the CVS failure and the GS-GECVS technique is activated at the end of GECVS. Both the GS techniques described in
Table 5.8: Experimental Results: Gate Sizing with GS-CVS and GS-GECVS on NCL-opt Architecture.

<table>
<thead>
<tr>
<th>BM circuits</th>
<th>GS-CVS</th>
<th>GS-GECVS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>#i/#o</td>
<td>% LV cells</td>
</tr>
<tr>
<td>9symml</td>
<td>18/2</td>
<td>48.36</td>
</tr>
<tr>
<td>ttt2</td>
<td>48/42</td>
<td>41.89</td>
</tr>
<tr>
<td>C499</td>
<td>82/64</td>
<td>0.0</td>
</tr>
<tr>
<td>C1908</td>
<td>66/50</td>
<td>19.33</td>
</tr>
<tr>
<td>C1355</td>
<td>82/64</td>
<td>12.36</td>
</tr>
<tr>
<td>t481</td>
<td>32/2</td>
<td>47.61</td>
</tr>
<tr>
<td>i8</td>
<td>266/162</td>
<td>75.8</td>
</tr>
<tr>
<td>dalu</td>
<td>150/32</td>
<td>49.45</td>
</tr>
<tr>
<td>C3540</td>
<td>100/44</td>
<td>58.31</td>
</tr>
</tbody>
</table>

Average percentage: 39.23 | 29.96 | 11.88 | 4.07 | 12.51 | 2.78

Table 5.9: Experimental Results: Gate Sizing with GS-CVS and GS-GECVS on DIMS Architecture.

<table>
<thead>
<tr>
<th>BM circuits</th>
<th>GS-CVS</th>
<th>GS-GECVS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>#i/#o</td>
<td>% LV cells</td>
</tr>
<tr>
<td>9symml</td>
<td>18/2</td>
<td>54.95</td>
</tr>
<tr>
<td>ttt2</td>
<td>48/42</td>
<td>66.49</td>
</tr>
<tr>
<td>C499</td>
<td>82/64</td>
<td>25.45</td>
</tr>
<tr>
<td>C1908</td>
<td>66/50</td>
<td>39.94</td>
</tr>
<tr>
<td>C1355</td>
<td>82/64</td>
<td>38.03</td>
</tr>
<tr>
<td>t481</td>
<td>32/2</td>
<td>79.12</td>
</tr>
<tr>
<td>i8</td>
<td>266/162</td>
<td>77.16</td>
</tr>
<tr>
<td>dalu</td>
<td>150/32</td>
<td>52.08</td>
</tr>
<tr>
<td>C3540</td>
<td>100/44</td>
<td>58.61</td>
</tr>
</tbody>
</table>

Average percentage: 54.65 | 37.55 | 23.68 | 15.32 | 21.4   | 6.41

the previous section, were analyzed for further energy savings on NCL-opt and DIMS style of NCL architectures. Tables 5.8 and 5.9 show the results of applying the GS-CVS technique after CVS for the NCL-opt and DIMS architecture respectively. These tables report on the percentage of LV cells that were obtained after applying the GS technique (columns 4 and 7), along with the corresponding energy reduction (columns 6 and 9) for chosen set of BM circuits. The percentage of gates that were upsized when GS techniques were applied is tabulated in columns 5 and 8 respectively.

In order to comprehend the tables better, a plot comparing the percentage of LV
Figure 5.57: Comparison of percentage of LV cells after GS-CVS and GECVS for NCL-opt Architecture.

Figure 5.58: Comparison of percentage of Energy Reduction after GS-CVS and GECVS for NCL-opt Architecture.
Figure 5.59: Comparison of percentage of LV cells after GS-CVS and GECVS for DIMS Architecture.

Figure 5.60: Comparison of percentage of Energy Reduction after GS-CVS and GECVS for DIMS Architecture.

assignment for the two techniques are drawn in fig. 5.57 for NCL-opt and in fig. 5.59 for DIMS architecture. Their corresponding energy reduction obtained by applying GS techniques are plotted in graphs 5.58 and 5.60. From these plots, it is apparent that the GS-CVS technique performs better than the GS-GECVS technique for both the NCL architecture versions. This is because, at the end of CVS technique, there are more cells than can be assigned to LV.

CVS fairs poorly compared to GECVS and thus has a larger room for optimization.
Table 5.10: Number of logic levels and number of levels of LV cells from PO after CVS.

<table>
<thead>
<tr>
<th>BM circuits</th>
<th># gates</th>
<th># logic levels</th>
<th>Max CVS level</th>
<th># gates</th>
<th># logic levels</th>
<th>Max CVS level</th>
</tr>
</thead>
<tbody>
<tr>
<td>9symml</td>
<td>919</td>
<td>43</td>
<td>0</td>
<td>366</td>
<td>22</td>
<td>0</td>
</tr>
<tr>
<td>ttt2</td>
<td>1328</td>
<td>19</td>
<td>7</td>
<td>530</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>c499</td>
<td>1446</td>
<td>35</td>
<td>0</td>
<td>556</td>
<td>18</td>
<td>0</td>
</tr>
<tr>
<td>c1908</td>
<td>1745</td>
<td>59</td>
<td>6</td>
<td>688</td>
<td>30</td>
<td>3</td>
</tr>
<tr>
<td>c1355</td>
<td>2590</td>
<td>49</td>
<td>0</td>
<td>1036</td>
<td>25</td>
<td>0</td>
</tr>
<tr>
<td>t481</td>
<td>2620</td>
<td>47</td>
<td>0</td>
<td>1048</td>
<td>24</td>
<td>0</td>
</tr>
<tr>
<td>i8</td>
<td>4360</td>
<td>29</td>
<td>11</td>
<td>1744</td>
<td>15</td>
<td>5</td>
</tr>
<tr>
<td>dalu</td>
<td>5962</td>
<td>43</td>
<td>20</td>
<td>2372</td>
<td>22</td>
<td>9</td>
</tr>
<tr>
<td>c3540</td>
<td>6350</td>
<td>85</td>
<td>18</td>
<td>2540</td>
<td>43</td>
<td>9</td>
</tr>
</tbody>
</table>

This fact is reiterated in the Table 5.10 which shows how deep the LV cells have travelled from PO. In contrast, GECVS technique has large cluster of LV cells and the percentage of LV cells that can be obtained by GECVS is as much as 60% in some circuit leaving little room for solution space when GS-GECVS is applied.

Nevertheless, GS-GECVS technique is able to obtain approx. 2.5% energy reduction on average for NCL-opt style and almost 7% for the DIMS style. Energy reduction of 6% or higher were attained in BM circuits dalu, i8 and ttt2 (NCL-opt version) with GS-GECVS.

Gate sizing almost 40% of its cells to gain 77% LV cells in DIMS-i8 circuit, GS-CVS offered a notable 43% improvement. This justifies that a GS technique can be used to create extra slack in the circuit than can enhance energy savings. It also reiterates the fact that as the backward front in GS-CVS is advanced towards the PI’s, significant energy savings can be obtained.

BM circuits such as 9symml and t481(NCL-opt version) moved the LV envelope just a single level from PO and hence leaves a larger room for optimization. Altogether, these BM circuits were able to achieve 10-14% energy improvement when GS-CVS was applied with 40% of its gate voltages lowered to LV. It should also be noted that GS-CVS and GS-GECVS techniques when applied to c499 (NCL-opt)
produced no energy reduction while the DIMS version produced 10% and 4% energy savings respectively. This is because, c499 (NCL-opt) circuit are mapped to some complex NCL gates that have large delays as their gate voltage is scaled. In contrast, DIMS architecture maps its gates to simpler TH22 and TH12 gates with LV delays significantly lower than complex NCL gates.

The energy savings reported after applying the GS-CVS technique for DIMS style of NCL circuits also has a notable 32% energy savings on average whereas, the GS-GECVS reported with 6% energy reduction. BM circuit i8 performed the best of the lot with allocating as much as 75% of its cell to LV for NCL-opt version and a close 77% for DIMS style achieving on average 40% savings.

In conclusion, the approaches taken here manifest the benefits of applying GS techniques to asynchronous circuits in low energy applications. The experiments conducted clearly demonstrate improvements after a VS technique was applied at the cost of large runtime and area penalty. On average the GS-CVS technique needed to resize 27% of gates for NCL-opt and 31% for DIMS version. GS-GECVS, on the other hand sized 11% and 26% of the cells to assign 36% and 47% of cells to LV to reduce energy for NCL-opt and DIMS version, respectively. GS-GECVS improved energy savings by 2.5% to 6% for NCL-opt and DIMS version respectively, employing a greedy technique with cubic runtime.

5.4 Summary of VS-GS technique:

Our approach to reduce energy of asynchronous circuits involved a two-step energy reduction process by applying VS technique followed by GS. This approach tends to produce effective energy reduction instead of applying just the VS or GS technique alone. This is because of the discrete nature of VS and GS which tends to leave the slack on some gates unutilized [78]. In this work, VS technique which drains the slack in circuit is energized by GS technique with the premise that more gates can
be assigned to LV, effectively reducing energy. This section summarizes the results of applying both VS and GS techniques on the MCNC BM circuits.

Two VS+GS experiments were performed. First, CVS followed by GS-CVS was applied to the set of chosen BM circuits and the effect of energy reduction was studied. In the second experiment, powerful voltage scaling technique − GECVS was followed by GS-GECVS gate sizing technique and its performance on the BM circuit was analyzed. Both the experiments were conducted on two varieties of NCL architecture − NCL-opt and DIMS. Table 5.11 compares the application of VS and GS techniques on NCL-opt type of BM circuits. The results of conducting the VS+GS algorithms on DIMS are tabulated in Table 5.12.

Columns 4 and 8 denotes the % of LV cells obtained after VS+GS whereas columns 6 and 10 represents the % of energy obtained during the two step process. The % of energy cells upsized during the GS process is reported in columns 5 and 9.

The two VS+GS techniques were compared for each BM circuit for energy reduction under a delay constrained environment. The percentage of energy reduction from each VS and its corresponding GS technique was plotted as a bar chart in the fig. 5.61 for a NCL-opt version and DIMS version is plotted in graph 5.62. The Bar 1 of each circuit represents the energy reduction for a combination of CVS+GS-CVS and the Bar 2 corresponds to the energy reduction when GECVS+GS-GECVS were applied.

A quick observation of the plot reveals that in a CVS environment where a strict topological constraint is followed, significant energy reduction happens in the GS phase. On average, as much as 30% of the total energy reduction occurs during the GS-CVS step for NCL-opt type circuits. In DIMS circuits, GS-CVS has an impressive 37% energy reduction.

Although GS-CVS and GECVS get much higher energy reduction when applied individually, the overall energy reduction by GECVS+GS-GECVS has superior per-
Table 5.11: Experimental Results: Voltage Scaling and Gate Sizing with CVS+GS-CVS and GECVS+GS-GECVS on NCL-opt Architecture.

| BM circuits | GS-CVS | | GS-GECVS | |
|-------------|--------|----------------|----------------|
| Name        | #i/#o  | #g    | % LV Cells | % GS cells | % Energy | % HSPICE | % LV Cells | % GS cells | % Energy | % HSPICE |
| 9symml      | 18/2   | 366   | 48.63      | 24.04      | 14.57    | 10.93    | 68.85      | 2.46       | 37.7     | 27.01    |
| ttt2        | 48/42  | 530   | 60.38      | 37.36      | 15.8     | 5.55     | 61.32      | 19.02      | 35.27    | 1.82     |
| C499        | 82/64  | 556   | 0.0        | 0.0        | 0.0      | 0.0      | 15.47      | 0.0        | 8.06     | 3.54     |
| C1908       | 66/50  | 688   | 35.47      | 12.21      | 10.3     | 11.23    | 43.02      | 0.0        | 22.12    | 6.24     |
| C1355       | 82/64  | 1036  | 12.36      | 9.46       | 0.47     | 0.61     | 20.75      | 4.92       | 11.46    | 6.37     |
| i481        | 32/2   | 1048  | 47.71      | 21.37      | 10.82    | 12.34    | 66.79      | 9.92       | 39.37    | 26.81    |
| i8          | 266/162| 1744  | 89.79      | 40.83      | 38.63    | 38.06    | 47.02      | 16.4       | 33.67    | 23.75    |
| dalu        | 150/32 | 2372  | 63.32      | 53.37      | 21.19    | 0.25     | 69.01      | 50.55      | 39.84    | 1.67     |
| C3540       | 100/44 | 2540  | 68.07      | 50.98      | 18.43    | 13.03    | 67.83      | 8.39       | 40.37    | 27.07    |
| Average percentage | 47.3 | 29.96 | 14.47 | 10.22 | 50.79 | 12.47 | 29.76 | 13.81 |

Table 5.12: Experimental Results: Voltage Scaling and Gate Sizing with CVS+GS-CVS and GECVS+GS-GECVS on DIMS Architecture.

| BM circuits | GS-CVS | | GS-GECVS | |
|-------------|--------|----------------|----------------|
| Name        | #i/#o  | #g    | % LV Cells | % GS cells | % Energy | % HSPICE | % LV Cells | % GS cells | % Energy | % HSPICE |
| 9symml      | 18/2   | 919   | 55.06      | 28.84      | 20.87    | 14.92    | 71.6       | 9.58       | 41.13    | 27.45    |
| ttt2        | 48/42  | 1328  | 87.05      | 79.52      | 36.65    | 6.22     | 67.85      | 40.59      | 38.7     | 19.13    |
| C499        | 82/64  | 1446  | 25.45      | 16.87      | 10.42    | 3.14     | 31.95      | 14.45      | 14.21    | 5.05     |
| C1908       | 66/50  | 1745  | 55.82      | 29.34      | 19.24    | 14.04    | 49.91      | 10.14      | 25.8     | 7.19     |
| C1355       | 82/64  | 2590  | 38.03      | 30.77      | 13.88    | 2.38     | 44.98      | 30.81      | 18.44    | 6.18     |
| i481        | 32/2   | 2620  | 79.16      | 26.95      | 30.41    | 25.15    | 76.56      | 18.47      | 40.9     | 49.1     |
| i8          | 266/162| 4360  | 95.8       | 39.22      | 49.45    | 46.76    | 69.13      | 41.42      | 38.17    | 20.9     |
| dalu        | 150/32 | 5962  | 70.08      | 57.58      | 28.21    | 12.03    | 59.11      | 18.9       | 44.04    | 15.25    |
| C3540       | 100/44 | 6350  | 68.69      | 28.83      | 29.7     | 15.37    | 70.87      | 8.27       | 42.51    | 21.78    |
| Average percentage | 63.9 | 37.55 | 26.54 | 15.56 | 60.22 | 21.4 | 33.77 | 19.11 |
Figure 5.61: Comparison of percentage of energy reduction after CVS+GS-CVS and GECVS+GS-GECVS for Ncl-opt Architecture.

Figure 5.62: Comparison of percentage of energy reduction after CVS+GS-CVS and GECVS+GS-GECVS for DIMS Architecture.

formance compared to CVS+GS-CVS for the style of BM circuit. The only exception being the BM circuit i8, which had slightly better performance when CVS and GS-CVS were employed. The Energy savings obtained by applying GS-CVS on BM circuit i8 from HSPICE is plotted in fig. 5.63. and fig. 5.64 compares the power curves before and after applying the GS-CVS for a single NULL-DATA-NULL cycle.

Circuits such as t481, dalu, and c3540 each see an energy improvement of about
Figure 5.63: Energy savings from GS-CVS on BM i8 (DIMS) plotted from HSPICE.

Figure 5.64: Comparison of power consumed in BM circuit i8 (DIMS) for one DATA-cycle.
40% or more when GECVS and GS-GECVS were applied for the DIMS style. In all the cases (for both NCL-opt and DIMS), the improvement in energy savings was significant when the corresponding algorithms where able to allocate large number of its cells to LV. This is confirmed in the plot comparing the percentage of LV cell allocation in each of the VS and GS techniques which is shown in fig. 5.65 for the NCL-opt version. The corresponding DIMS plot is shown in fig. 5.66.

It is also important to note that DIMS style circuits such as c3540, dalu and ttt2 despite having good percentage of LV when CVS+GS-CVS was applied has comparatively lower energy reduction than their GECVS counterparts. The reason for this variation can be associated with the percentage of upsized cells. The percentage of upsized cells that were allocated during GS-CVS process nearly defeats the purpose of LV assignment to reduce energy. In the case of BM circuit ttt2 (DIMS style), although at the end of VS+GS 87% of cells were assigned LV, nearly three-fourths of the cells were upsized to compensate for timing and hence, reduces the margin of energy savings.

To conclude, one can infer that applying VS alone cannot produce an energy efficient circuit and combination of VS and GS are more effective. The power plot
Figure 5.66: Comparison of percentage of LV cells after CVS+GS-CVS and GECVS+GS-GECVS for DIMS Architecture.

of applying GECVS followed by GS-GECVS for the BM circuit i8 is shown in fig. 5.67. The power plot clearly illustrates the reduction in power after each stage of VS and GS. The corresponding energy plot showing the energy savings after each stage is shown in fig. 5.68.
Figure 5.67: Power plot of BM circuit i8 before and after GS-GECVS.

Figure 5.68: Energy plot of BM circuit i8 before and after GS-GECVS.
6 Conclusions

This chapter presents the conclusions of this dissertation research work along with a section on scope for future work.

6.1 Summarizing the dissertation work

This research work addressed the noise mitigation in MS environment with a promising asynchronous design methodology especially the threshold network circuits such as NCL. The primary objective of this work was to exploit and harness the inherent low power advantages of asynchronous circuits that make them good candidates for low power and low noise environment. New methods that tackle the dynamic energy reduction in asynchronous circuits particularly focused on asynchronous threshold network circuits were implemented.

Widely popular synchronous techniques such as voltage scaling and gate sizing were proposed and implemented to reduce dynamic energy. Our energy reduction techniques involved both the VS and GS techniques. They were applied in a two-step process, VS first followed by GS, one technique complementing the other.

In the chapter 4 we presented an overview of voltage scaling techniques that have been extensively used in synchronous design at gate level which were extended for asynchronous design. The impact of voltage scaling on asynchronous threshold network circuits were studied in dual voltage environment with three voltage assignment algorithms that attempted to produce a low energy circuit without sacrificing the speed of the circuit. The approaches undertaken to reduce energy involved a heuristic method and sensitivity based techniques that were applied to asynchronous NCL type circuits. The widely used MCNC benchmark circuits were mapped to two popular NCL architectures - NCL-opt and DIMS version and the energy savings were compared and analyzed in detail.

Since, VS techniques exploits slack for energy savings, at the end of VS, a gate
upsizing technique was used to further reduce the energy in the circuits based on a sensitivity measure. The GS approaches applied to asynchronous NCL type circuits were analyzed for its effectiveness. The results of applying GS techniques on both NCL-opt and DIMS architectures are detailed in chapter 5. A novel embedded level converter for NCL gates was also proposed and used in the dual voltage design demanding LCs. Significant improvements in delay and energy were obtained by employing embedded-LC over the gate-LC version.

At the end of applying both the VS and GS, the energy reduction techniques produced energy improvements averaging 26%. VS+GS techniques were found to be effective on both the asynchronous NCL architectures and it was verified by simulating the BM circuits at physical layout level with Industry standard tool HSPICE to measure energy.

These energy efficient, yet low noise circuits have often been blamed for its design complexity and deficiency of industry standard CAD tools and shunned from adoption for the same reason. This issue was acknowledged with the use of commercial synchronous tools that attempts to bridge the gap between synchronous and asynchronous design domains. NCL has the flexibility of employing synchronous tools for asynchronous design and a complete design flow from high level design to layout was equipped, with commercially available off-the-shell industry standard tools.

Issues related to dual voltage P&R was addressed with modifying the NCL standard cell to accommodate an extra rail for DV design using commercial P&R tools. Other important considerations such as standard cells for ASIC physical design of NCL circuit were addressed in this work by creating a novel automated standard cell generator.
6.2 Future work

There are several possible directions for future work. The current work only addressed the dynamic power reduction in asynchronous circuits and neglected the static energy effects. As design technologies reach nanometer regime, threshold voltage is scaled along with the supply voltage. As a consequence, the sub-threshold current increases exponentially [80]. This implies that the leakage power can no longer be ignored and this opens new areas of research interest especially in the asynchronous design.

As supply voltage is scaled, to compensate for speed penalty, in synchronous designs, threshold voltage scaling is also employed [82, 81] with dual voltage scaling. These dual threshold voltage scaling techniques together with VS or GS can be extended to asynchronous design. Simultaneous VS, GS approach or applying VS and GS concurrently with threshold voltage scaling for asynchronous design are additional options that could potentially lead to better energy savings.

In this work, we proposed and implemented two-stage energy reduction algorithms. However, due to the discrete nature of using dual voltage and gate re-sizing, VS and GS algorithms tend to perform differently and hence, extending the research to analyze the effects of applying GS first, followed by VS and vice versa could lead to effective energy reduction techniques.

Further, the drive strength of the cell was limited to max size of X4 and the sensitivity of GS algorithms to cell size available in the library needs further investigation. Our versatile silicon generator tool is capable of generating cells of higher drive strengths and finer granularity and it could aid in detailed investigation with the creation of cell libraries of variable drive strength.

The timing models that were used in this work, did not consider the input signal slew effects or the transition time. A better timing model which considers all these effects and also incorporates path delays has the prospects of leading to better results.
Another assumption that was undertaken in this research work was, every gate in the design switches twice in a NULL-DATA-NUL cycle and to improve this model, a possible solution as suggested in [68] could be incorporated and studied.

The gate synthesis of BM circuits was done by replacing every 3NCL gate with its 2NCL counterparts. Although, the resulting netlist was timing robust and orphan-free, it has a huge area overhead almost 2x more than its synchronous counterparts. An optimal technology mapping as prescribed in [51] or the overly conservative synthesis could be approached in a relaxed manner without sacrificing the robustness of the circuit as described in [2]. This work relinquished completion detection (CD) circuitry which are known to present area, delay and power overhead, an energy reduction algorithm would be effective if CD circuitry contribution to energy is also considered.

It should also be noted that this work considered only the static architecture implementation of NCL gates. Semi-static NCL gates have less area and offer better speed advantages than its static counterpart. Analyzing the effects of the proposed VS and GS algorithms on semi-static NCL architecture is another potentially interesting research direction that can be pursued.
References


