Polynomial-Time Algorithms for Designing Dual-Voltage Energy Efficient Circuits

by

Mridula Allani

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
December 12, 2011

Keywords: Energy efficient design, Dual-Vdd design, Level converter, Gate slack, Static timing analysis (STA)

Approved by

Vishwani D. Agrawal, Chair, James J. Danaher Professor of
Electrical and Computer Engineering
Victor P. Nelson, Professor of Electrical and Computer Engineering
Adit Singh, James B. Davis Professor of Electrical and Computer Engineering

Abstract

In this work, we propose a technique to use dual supply voltages in digital designs in order to get a reduction in energy consumption. Three new algorithms are proposed for finding and assigning low voltage in dual voltage designs. Given a circuit and a supply voltage, the first algorithm finds an optimal lower supply voltage and the other two algorithms assign that lower voltage to individual gates. A linear time algorithm described in the literature is used for computing slacks for all gates in a circuit for a given supply voltage.

For the computed gate slacks and the lower supply voltage, the gates in the circuit are divided into three groups. No gate in the first group can be assigned the lower supply. All gates in the second group can be simultaneously set to lower supply while maintaining positive slack for all gates. The gates in the third group are assigned low voltage in small subgroups. The gate slacks are recalculated after each such voltage assignment. Thus, the overall complexity of this reduced power dual voltage assignment procedure is $O(n^2)$. But in practice, it is observed that the computation time is close to linear in the circuit size. SPICE simulations of ISCAS'85 benchmark circuits using the PTM model for 90-nm bulk CMOS technology results show up to 60% energy savings.

Acknowledgments

This thesis would not have been possible without the support from my advisor, Dr. Vishwani Agrawal. I am thankful for his guidance and encouragement. It has been a pleasure working with him.

I thank my advisory committee members, Dr. Victor Nelson and Dr. Adit Singh. Also, I am grateful to the Auburn transition Leadership Institute (ATLI) and the Department of Electrical Engineering at Auburn University for supporting my graduate study.

I sincerely appreciate Mrs. G. Kanaka Durga and Mrs. B. Sarala for their guidance during my college. I also thank all my teachers who have inspired me through my life.

Graduate study at Auburn University has been a learning experience. I thank my colleagues Farhana, Jia, Kim, Manish, Nitin, Priya and Suraj for their valuable inputs when required. Without them my graduate experience would not have been this enriching.

I am indebted to my parents, brother, sister and all friends for their love and support. Also, I am very thankful to my husband, Abhishek Bichal, for being with me through the thick and thin.

Finally, I thank everyone who directly or indirectly helped and inspired me during the course of my graduate study.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Power and performance are the main design constraints in modern VLSI design. An effective design is obtained with the optimization of these two parameters. Decreasing the supply voltage results in decreased performance. Hence, a trade off is necessary between power consumption and circuit delay. The use of multiple supply voltages to reduce energy consumption is a very commonly used technique in CMOS circuits. This results from the fact that the dynamic power of a CMOS circuit is directly proportional to the square of its supply voltage [9, 73]. The underlying idea of this technique is to trade timing slacks for power reduction under given timing requirements. Generally, the gates in the critical path are kept at high supply voltage and the gates in non-critical paths are put to lower supply voltage, thus avoiding timing violations.

Multiple supply voltage design using the time slack is a popular technique for power reduction, [8, 10, 13, 21, 27, 41, 42, 43, 63, 78, 80, 82, 84, 86, 100]. Several references [12, 27, 62, 63] discuss dual voltage designs in FPGAs. In this work we use a linear time slack analysis algorithm [41] to calculate gate slacks.

Apart from using multiple supply voltages we can combine multiple supply voltage technique with multiple thresholds and transistor sizing to achieve additional power savings. Such work is described in articles [6, 18, 30, 37, 78, 83, 85, 86]. The trade-offs of energy savings and increase in delay when using gate sizing and multiple supplies is studied in [87].

Authors of [102] propose an algorithm that assigns voltage islands generated using a slack distribution and voltage partitioning algorithms in order to reduce the power consumption and peak temperature. Article [64] proposes a linear time approximation algorithm for

partitioning of circuit components into various voltage islands during high-level synthesis. Other works [11, 15, 26, 57, 58, 98] deal with the voltage partitioning problem.

In some instances not meeting the timing requirements of an application is not catastrophic. These are called soft deadlines. Hua and Qu [29] describe an algorithm for energy efficient designs using dual voltage to meet such soft deadlines.

1.1 Motivation

There are many algorithms for assigning a fixed low voltage value to the gates of a circuit, but relatively fewer algorithms to find the lower voltage. Many of these works related to voltage assignment have used a low voltage, V_L value of 70% of the high voltage, V_H [6, 21, 50, 56, 73]. The works described in [10, 50, 84] claims that the optimal value of V_L for minimizing total power is 50% of V_H. Authors in [101] describe an algorithm to find the lowest feasible supply voltages according to their slacks from a set of given voltages. An algorithm to find an optimum V_L value is described in [41]. The authors assign a low voltage value to a group of gates based on a modification of CVS algorithm and then calculate energy over a set of low voltages. The V_L value resulting in minimum energy is chosen to be the lower voltage. This algorithm requires the voltage assignment to be done for each voltage value and is exhaustive in nature. We understand that the lower voltage for minimum energy operation of a dual-supply design is dependent on the circuit topology and is not a fixed value for all circuits. These reasons motivate us to propose a new algorithm that finds the V_L accurately and with linear complexity.

Authors in [92, 94] propose two ways of assigning a lower voltage in a dual-voltage design. The algorithm in [92] is called Clustered Voltage Scaling (CVS). This method puts a topological restriction on the resultant dual-voltage design which does not allow a low voltage gate at the input of a high voltage gate. Following this an Enhanced Clustered voltage Scaling (ECVS) algorithm was proposed in [94]. This algorithm allows a low voltage gate to drive a high voltage gate with the inclusion of an asynchronous level converter at the

boundary. Reference [93] describes the methodology to synthesize circuits for the CVS and ECVS structures. Authors of [88] describe three algorithms for dual voltage designs based on linear programming models. Several other algorithms have been proposed which aim at power reduction using dual voltage designs. Not many aim at energy-efficient designs. Also, the complexity of these algorithms is often polynomial. Thus, we are motivated to propose two quadratic time algorithms for dual-voltage assignment one of which allows level converters and the other which does not allow level converters.

1.2 Unique Contributions

An algorithm that finds a lower supply voltage maximizing the energy savings from a specific group of gates in a given circuit is proposed. This algorithm is proportional to the number of gates in the circuit as each gate slack is compared to the difference between the low voltage delay and high voltage delay of the gate. Thus, its complexity is linear, i.e. $O(n)$, where n is the number of the gates. Two other algorithms are proposed which assign the lower supply voltage found using the gate slacks. The gate slacks are recalculated after each iteration of voltage assignment. Both the voltage assignment and the slack calculating algorithms are linear in time. Thus, the overall complexity of this method quadratic, i.e, $O(n^2)$, where n is the number of gates. But in practice, it is observed that the computation time is close to linear-time. All algorithm is written in perl programming language. Energy savings of up to 60% is seen for ISCAS'85 benchmark circuits and when the critical timing is relaxed, savings of up to 70% are observed. Such high savings have not been reported in the earlier work. Sufficient theoretical and experimental work has been done to validate these results.

1.3 Problem Statement

The aim of this work is to:

1. Develop a linear time algorithm that finds the optimal lower voltage for the dual-Vdd design for CMOS circuits.

2. Develop new algorithms for voltage assignment in dual-Vdd design for given high and low voltages using linear-time gate slack analysis to reduce computation time.

1.4 Organization

The proposed work is organized as follows. Chapter 2 provides the background information on the various existing low power/energy-efficient design techniques. The sources of energy consumption in CMOS designs have also been detailed.

Chapter 3 gives an overview of the static timing analysis algorithms and illustrates the algorithms proposed in [24, 41] with examples.

In Chapter 4, we propose a new linear time algorithm to find the lower supply voltage using gate slacks for dual-voltage designs. Relevant theorems are proved and experimental data has been discussed.

Chapter 5 proposes a new quadratic time algorithm for low voltage assignment based on gate slacks which uses certain constraints that put restrictions on circuit topology ass proposed in [92]. Relevant results have been discussed.

In Chapter 6, various level converter designs have been studied for energy and delay. The level converter with least energy consumption is selected as we are concentrating on energy-efficient designs.

In Chapter 7, we propose another new quadratic time algorithm for low voltage assignment based on gate slacks which uses the selected level converter to lift the topological constraints imposed earlier.

Finally, Chapter 8 concludes the work with suggestions for future research. The need to include level converter overheads in the algorithm is realized from the results of Chapter 7. Hence, we propose this for the future work.

Chapter 2

Background

2.1 Introduction

Power and performance are the main design constraints in modern VLSI design. An effective design is obtained with the optimization of these two parameters. This requirement for higher performance and lower power directly results from the increasing demand for portable electronic devices. Fortunately, the VLSI technology has been able to integrate millions of gates into a single chip of small area, thus contributing towards reduction of size and increase in portability of the electronic devices incorporating these chips. Also, the humongous integration on a small area has increased the speed of the devices owing to greater current drive at the transistor level and smaller propagation paths at the gate level and several other factors. But the continuous scaling has led to increased power dissipation in CMOS circuits. Hence, we need to find an optimum design where the power dissipation is acceptable and the computational speeds are not disappointing.

The power and timing efficiency can be captured by the following metrics [73]:

1. Average power
2. Power per MIPS (million instructions per second)
3. Energy
4. Energy-delay product
5. Energy-delay squared
6. Peak power

The choice of the design metric used for design optimization depends on the application and its performance specifications and power budgets. The various mechanisms causing

power dissipation in CMOS circuits and the popular power reduction techniques are discussed in the following sections of this chapter.

2.1.1 Power Dissipation in CMOS Circuits

Power dissipation in CMOS digital circuits can be classified as: static power and dynamic power. The static dissipation arises due to the leakage current through the resistive paths from the power supply to ground during steady state. The static power exists even in the inactive blocks of the digital circuits to hold the logic states between the switching events [73]. The leakage current through the transistors results from substrate injection and sub-threshold effects; and is primarily determined by fabrication technology considerations. Dynamic power is due to the switching of capacitive loads and the short circuit current resulting when both the pFET and nFET of a CMOS circuits are ON during switching transitions.

The above-mentioned sources of power consumption are summed up in the equation

$$P = CV_{DD}\Delta V f\alpha + I_{SC}V_{DD} + I_{leak}V_{DD} \tag{2.1}$$

where α is the activity factor, C is the total capacitance, V_{DD} is the supply voltage, f is the clock frequency, ΔV is the input voltage swing [9, 73], I_{SC} is the short circuit current and $I_{leakage}$is the leakage current. In equation 2.1, the first term corresponds to the switching component of power, the second is due to the short circuit current and the last term defines the power dissipated by the leakage current.

When the input voltage swing is equal to V_{DD}, the switching power will be

$$P_{Switching} = CV_{DD}^2 f\alpha \tag{2.2}$$

This switching component of power is due to the logic transition at the gates as shown in Figure 2.1. When the input goes from 1-to-0, the pMOS transistor is switched ON and

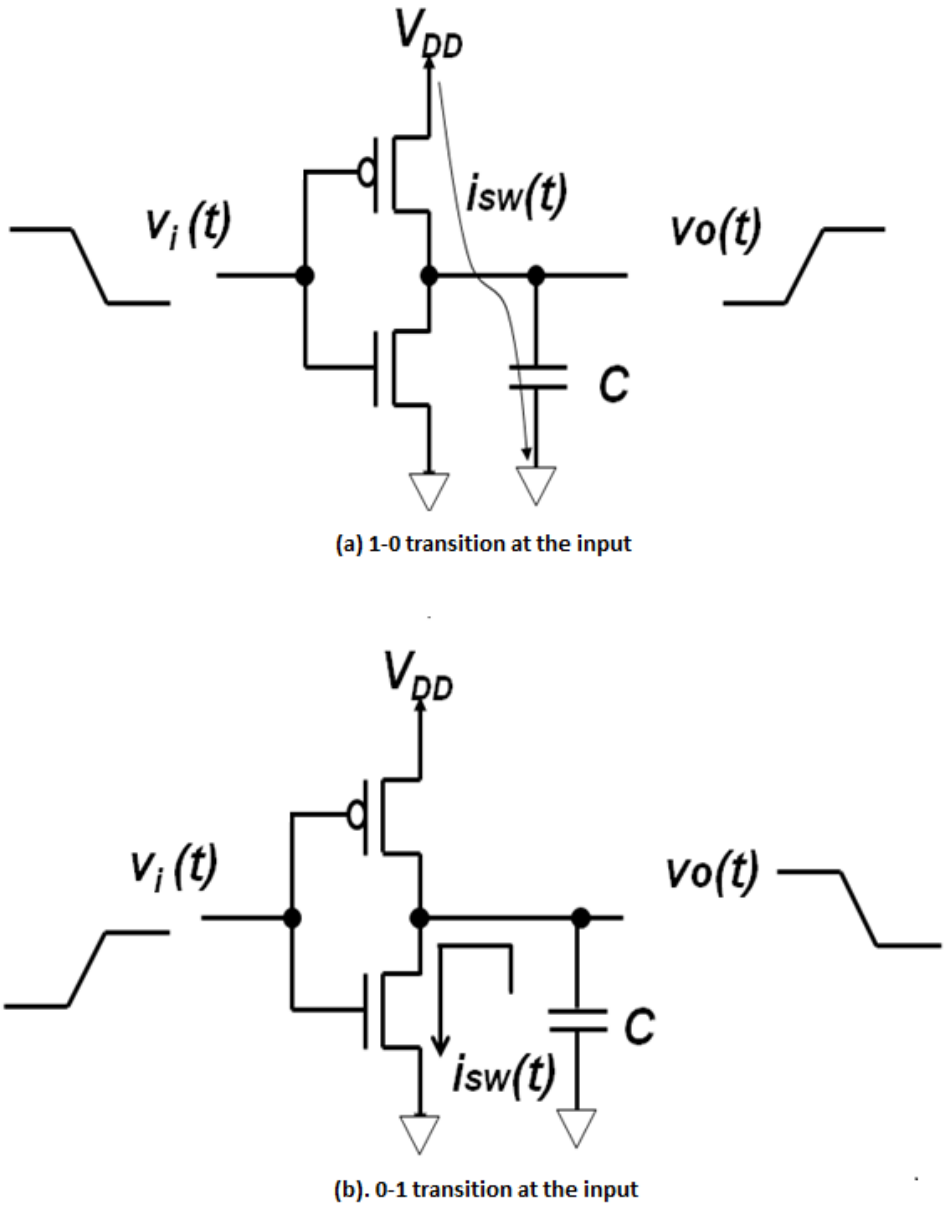

Figure 2.1: Switching current in an inverter.

the capacitance at the output node, C, is charged from the supply V_{DD}. When the input transitions from 0-to-1, the nodal capacitance is discharged into the ground through the nMOS transistor. Thus, during each transition power given by equation 2.2 is expended.

The short circuit current flows from the supply to the ground during a logic transition for a short duration for which both the nMOS and pMOS trees of the gate are turned ON. It is shown in Figure 2.2. The graphs in Figure 2.3 show the characteristics of the short circuit current. We see that during a rise transition at the input of an inverter, when V_{in} is at 0V, the pMOS transistor is ON and the nMOS transistor is OFF. Once the input becomes more than $V_{DD} - V_{tn}$ the nMOS transistor turns ON. As the input keeps rising and becomes greater than $V_{DD} - V_{tp}$ the nMOS transistor turns OFF. We notice that during the period when the input is $V_{DD} - V_{tn}$ and when the input is $V_{DD} - V_{tp}$, both the nMOS and pMOS transistors are ON. This causes a current to flow from the supply to the ground through the ON-resistances of the two transistors. Similarly, we can interpret the fall transition in the graph 2.3. Hence, the short circuit current depends on the rise and fall times of the inputs. The short circuit power of an unloaded inverter shown in figure 2.2 is found to be approximately

$$P_{SC} = \frac{\beta}{12}(V_{DD} - V_{th})^3 \tau f \tag{2.3}$$

where β is the transistor coefficient, V_{th} is the threshold voltage of the transistor assuming both pMOS and nMOS have the same threshold voltage, τ is the rise/fall time and f is the frequency [72, 77].

From the equations 2.1, 2.1 2.3 we observe that the dynamic power consumed by digital CMOS circuits can be reduced by adjusting the supply voltage, threshold voltages, node capacitance, activity factor, clock frequency, low voltage swing signaling techniques, etc.

Finally, the third component of power is the static power. This power is consumed to maintain the logic levels at the gate outputs when the inputs are in steady state. The main sources of leakage currents are the source/substrate or drain/substrate p-n junction band-to-band tunneling leakage current (I_D), gate direct tunneling current (I_G), which is due to the tunneling of electron or a hole from the substrate to the gate through the gate oxide, and the subthreshold leakage current from drain to source through the channel of an OFF transistor (I_{SUB}) [35, 72, 73, 74].

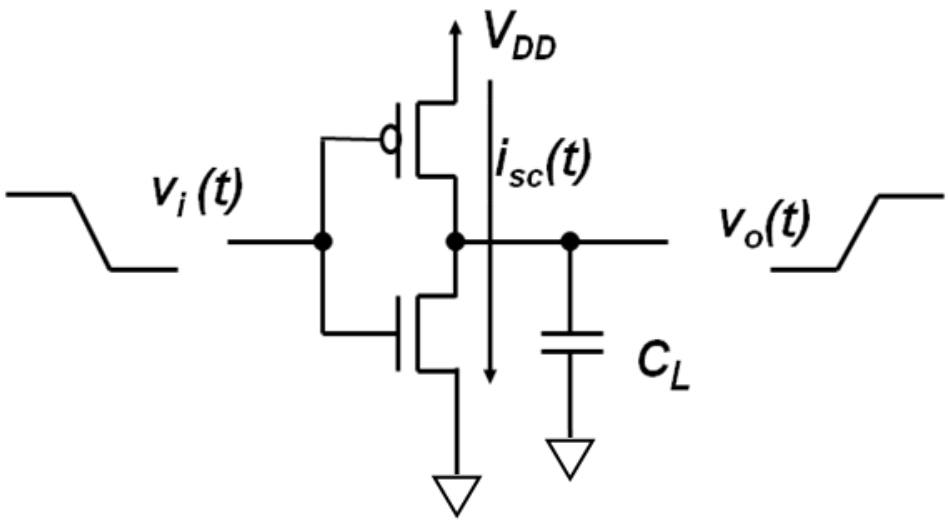

Figure 2.2: Short circuit current in an inverter.

The subthreshold leakage or weak inversion conduction occurs when the gate voltage is below V_{th}. This is due to the diffusion current of the minority carriers in the channel. Consider a CMOS inverter. When the input is low, the nMOS transistor is turned OFF and the output voltage is high. Even when V_{GS} is 0V, there is still a current flowing in the channel of the OFF nMOS device due to the V_{DD} potential between the drain and the source. This component of leakage is the dominant component due to the low threshold voltages being used. The subthreshold current is given the following equation

$$I_{SUB} = Ae^{(\frac{q}{nkT})(V_{GS}-V_{th0}-\gamma' V_{SB}+\eta V_{DS})}(1 - e^{\frac{-qV_{DS}}{kT}}) \tag{2.4}$$

$$A = \mu_0 C'_{ox} \frac{W}{L_{eff}} (\frac{kT}{q})^2 e^{1.8} \tag{2.5}$$

where γ' is the linearized body-effect coefficient, V_{SB} is the source-to-bulk voltage of the transistor, V_{GS} is the gate-to-source voltage of the transistor, η is the DIBL coefficient, V_{DS} is the drain-to-source voltage of the transistor, C_{ox} is the gate oxide capacitance, μ_0 is the zero-bias mobility, V_{th0} is the zero-body-bias threshold voltage and n is the subthreshold swing coefficient of the transistor [35, 72, 73, 74].

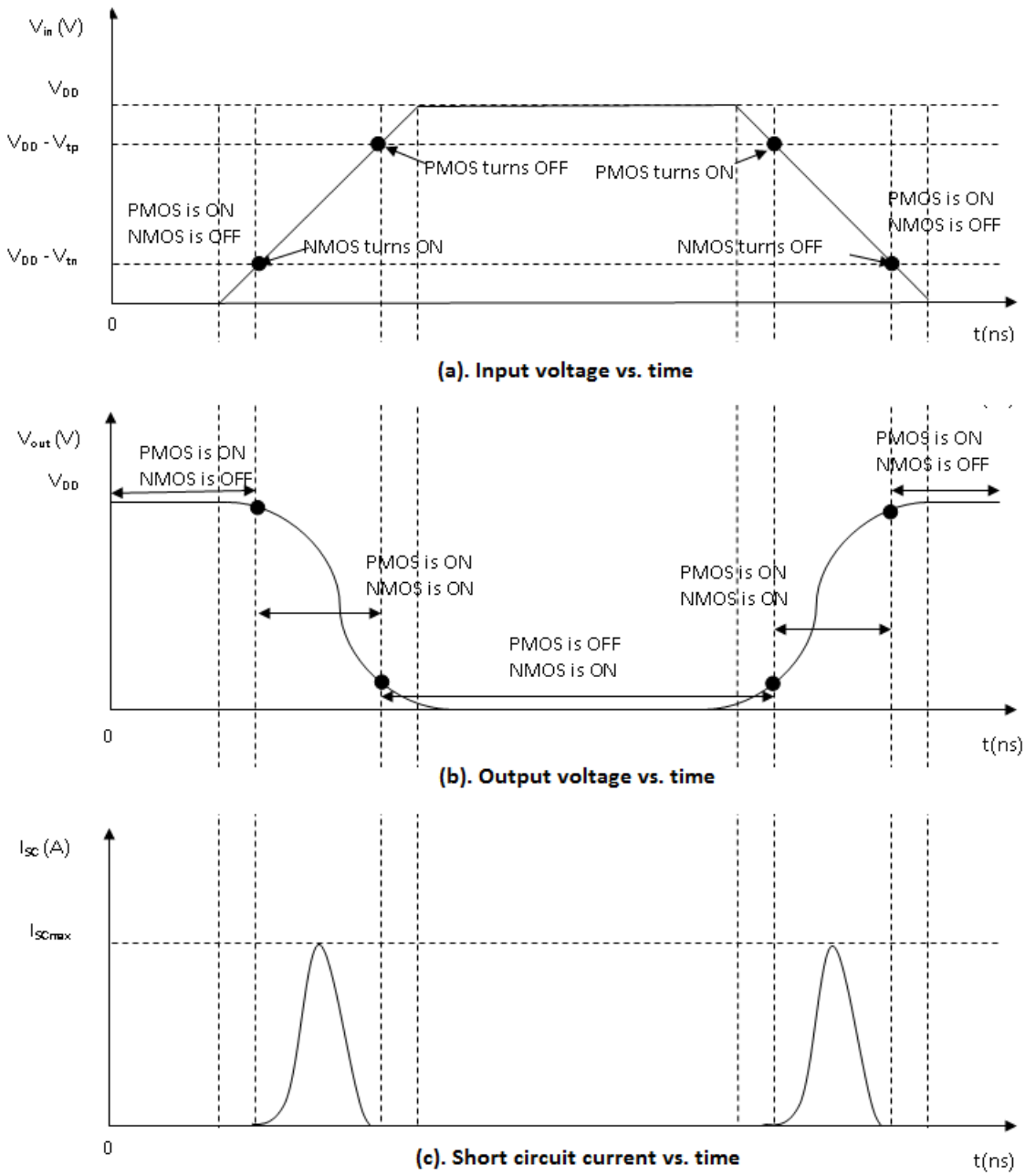

Figure 2.3: Short circuit current characteristics of an inverter.

The gate-direct tunneling current density is given by the equation

$$J_{DT} = A(\frac{V_{ox}}{t_{ox}})^2 e^{\frac{-B(1-(1-\frac{V_{ox}}{\phi_{ox}})^{\frac{3}{2}})}{\frac{V_{ox}}{t_{ox}}}} \tag{2.6}$$

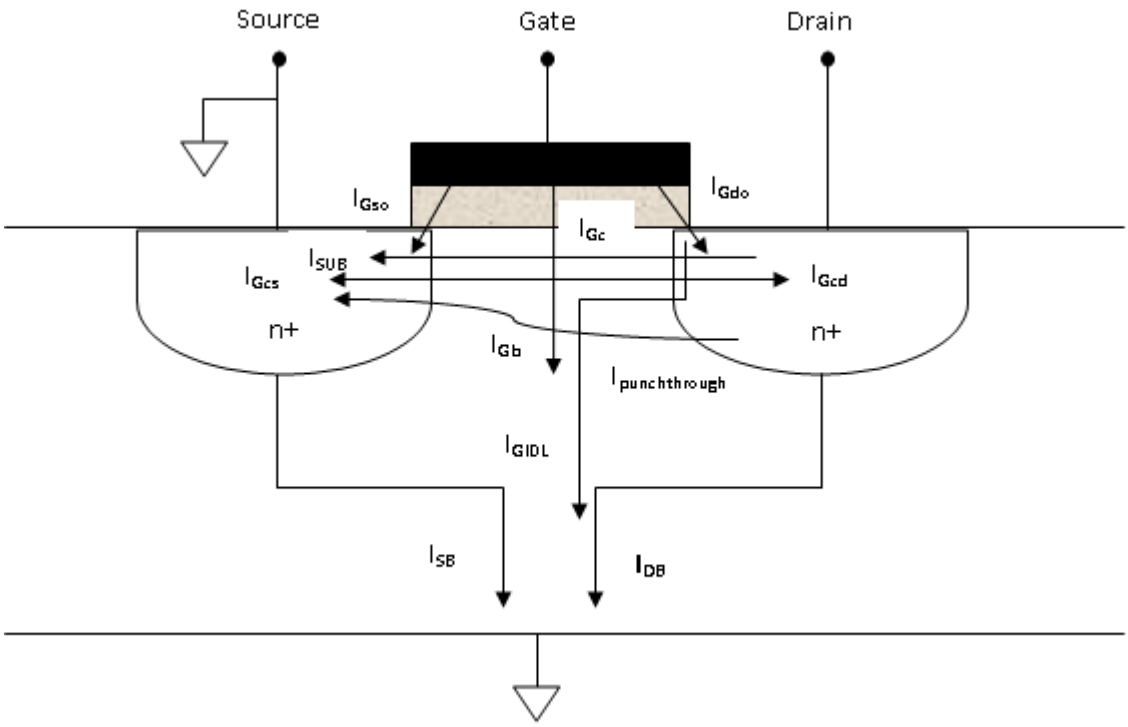

Figure 2.4: Leakage currents in an nMOS transistor.

where V_{ox} is the potential drop across the thin oxide, ϕ_{ox} is the barrier height of tunneling electron, and t_{ox} is the oxide thickness. The various components of the gate direct tunneling current are the edge direct tunneling which goes into the source and drain extension regions (I_{gso} and I_{gdo}), the gate to channel current (I_{gc}) which gets divided into the gate-to-source (I_{gsc}) and the gate-to-drain (I_{gdc}) currents and the gate-to substrate leakage current (I_{gb}) [72]. The gate direct tunneling current increases exponentially with the reduction in oxide thickness and supply voltage. This component of leakage is fast becoming the dominant part. In 90nm devices, gate leakage can be 1/3 of the subthreshold leakage. It can become equal to subthreshold leakage in 65nm devices [35]. Using a high-k dielectric for gate oxide can help reduce the gate leakage.

The band-to-band tunneling (BTBT) current from the drain and source to substrate is the reverse-biased pn-junction current consisting of the diffusion/drift near the edge of the depletion region and the current due to the electron-hole pair generation in the depletion region. In the case of an inverter, when the input is low the pMOS is ON, the nMOS is

OFF, and the output is high. This high-voltage at the output makes the drain-to-substrate voltage of the nMOS transistor equal to V_{DD}. This results in leakage from the drain to the substrate. This component of leakage current increases significantly when the wells or the substrate are highly doped or the drain and source at very high potential than the substrate. The band-to-band tunneling leakage can be reduced by forward biasing the substrate or by reducing the doping near the drain-substrate and source-substrate junctions [72, 73, 74, 76]. The BTBT current density is given by

$$J_{BTBT} = A(\frac{EV_{rev}}{E_g^{1/2}})e^{(-B\frac{E_g^{3/2}}{E})} \tag{2.7}$$

$$A = \frac{\sqrt{2m^*q^3}}{4\pi^3h^2} \tag{2.8}$$

$$B = \frac{4\sqrt{2m^*}}{3qh} \tag{2.9}$$

$$E = \sqrt{\frac{2qN_aN_d(V_{rev} + V_{bi})}{\epsilon_{si}(N_a + N_d)}} \tag{2.10}$$

where N_a and N_d are the doping in the p and n sides, ϵ_{si} is the permittivity of silicon, m^* is the effective mass of the electron, q is the charge of an electron, h is $1/2\pi$ times the Planck's constant, E is the electric field at the junction, E_g is the energy-band gap, V_{rev} is the applied reverse bias, and V_{bi} is the built-in voltage across the junction [76].

In addition to the above discussed leakage mechanisms, other secondary effects also exist. Gate induced drain leakage I_{GIDL} is the current flowing from drain to substrate because of the high electric field effect in the drain region due to the high drain-to-gate voltage [35, 76]. In short-channel devices, due to the proximity of the drain and the source, the depletion regions at the drain-substrate and source-substrate junctions extend into the channel. As the channel length is reduced, if the doping is kept constant, the separation between the

depletion region boundaries decreases. An increase in the reverse bias across the junctions also pushes the junctions nearer to each other. When the combination of channel length and reverse bias leads to the merging of the depletion regions, punchthrough currents $I_{punchthrough}$ flow from drain to source [76].

Leakage currents are exponentially increasing with the scaling of CMOS. They increase with the decrease in threshold voltages and the length of the transistors. The operating temperature increases during device operation. This causes an exponential increase in the leakage currents.

The power dissipated by the CMOS digital circuits can be reduced while retaining the required functionality and performance. An outline of few of the techniques proposed for the power reduction in CMOS circuits is given in the following section.

2.2 Power Reduction in CMOS Circuits

Minimizing power consumption in digital CMOS circuits can be done either by slightly altering the make of the transistors at the fabrication level, or by making changes in the implementation level, or the architecture of the basic building blocks of a complex CMOS system.

At the designing level, power reduction can be achieved by making low-power dissipation the key objective of the design. This allows a simpler design with the reduction of the number of primitive gates and the resistive paths connecting them. This reduction in the number of dissipating components decreases the overall power dissipation of the system while performing the same required operations.

2.2.1 Technology Scaling

This technique proposes to scale all the voltages and linear dimensions by a constant factor γ (< 1) as shown in Figure 2.5 [73, 74, 76]. Since the electric field in the device remains constant the device current and wire capacitances are all scaled by the same factor.

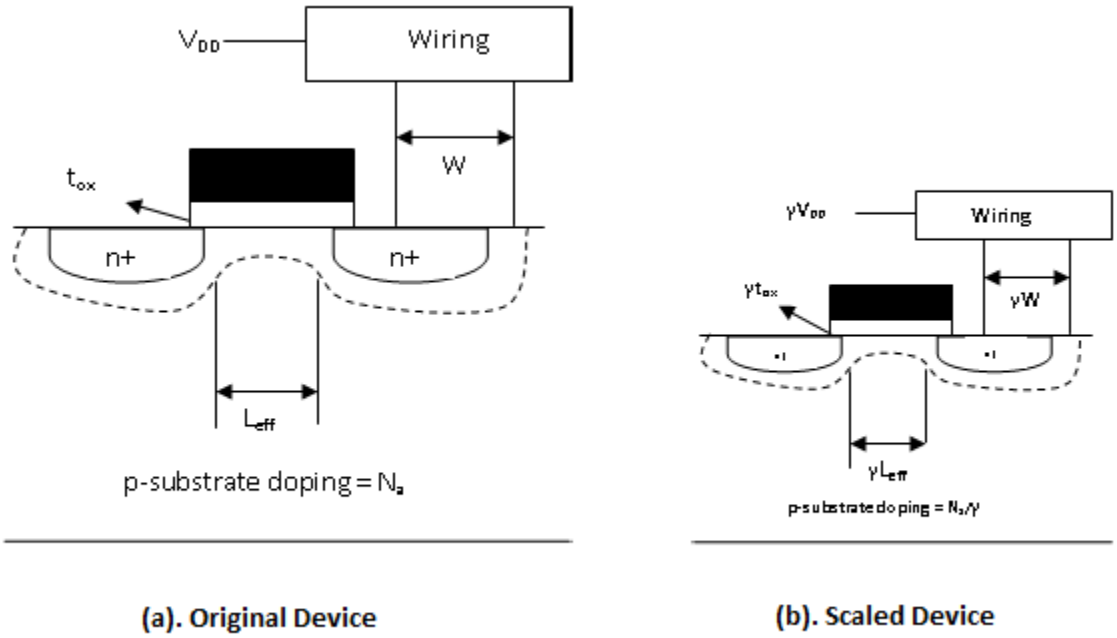

Figure 2.5: Scaling of an nMOS transistor.

The energy scales by a factor γ^3, as both voltage and current are scaled by γ. The delay is improved by a factor γ. Hence the energy-delay product decreases by a factor γ^4. But this method requires all voltages to be scaled down, including the threshold voltage. But the threshold voltage is limited by the leakage current through the OFF transistors and allowable static power [25, 73, 74].

In recent technologies, the supply voltage has reached 1V. This has imposed physical limitations to scaling. The built-in potentials of the device and Si bandgap energy do not change with scaling. They can be adjusted to required levels by increasing the doping or forward biasing the substrate. Because of thermodynamic limitations the threshold voltage of a device cannot be scaled further while keeping the leakage currents manageable. Reaching this ultimate level of scaling is stalled by increasing the electric field in the device by a factor ϵ (> 1). But this increases the power dissipation and decreases the reliability of the device. Hence, there is a limit to this scaling process due to various physical phenomena and alternative methods should be chosen to overcome this [73].

2.2.2 Transistor Sizing

We can reduce the junction capacitance and the overall gate capacitance of the transistors, which cause loading on the input, by making the transistors much smaller than the load capacitance applied [25, 72]. Reduction in size reduces the current drive of the transistors, thereby decreasing the system performance. This technique provides a trade-off between energy and speed as lower sizing results in lower energy and greater delay. Gates in the critical paths are sized according to the performance requirements and the gates in the non-critical paths are sized to minimize area to reduce power dissipation [72].

2.2.3 Supply Voltage Scaling

This method employs a reduction in supply voltage. Decreasing the supply voltage results in large savings in power because switching power is proportional to the square of the supply voltage. Supply voltage scaling also reduces the leakage power since the gate leakage and GIDL and DIBL leakage components are also reduced [72]. As the capacitance and the threshold voltage are kept constant, the performance of the system decreases. At larger voltages, reducing the supply voltage reduces energy for an acceptable change in delay. At voltages near the device threshold, small supply changes cause a large change in delay for a modest change in energy. Hence, this technique is best suitable at higher voltage ranges. For voltages comparable to the threshold voltage this method cannot be used.

When using a lower V_{DD}, we can use parallel or/and pipelined architectures to maintain decent circuit speeds [9, 25, 73]. Parallelism allows one to build N functional units, and perform N operations at the same time. This can decrease the time required for the operation of these N functions when compared with the same system having only one output.

Besides saving energy, parallelism decreases the system delay. Parallelism allows a greater number of functions to be performed simultaneously at the same time. In systems with single output function, these operations must be carried out sequentially and thus, require a greater time. Thus we can decrease the supply voltages of each of these blocks

while keeping the delays equal to that of the single block delay and eventually decrease the energy consumption of the system. Additionally, parallelism allows us to shut down some blocks when not being used. Hence, an energy-efficient system can be designed.

But all systems cannot be designed to extract parallelism. The system design, in some cases, may not give the desired level of parallelism, i.e., the number of functions being performed simultaneously might not be as large as required. Besides, designing such alternative systems requires a lot of creative insight and may also extend the time required for designing.

We can use different voltages to different functional blocks based on the performance requirements of those specific blocks. There are many algorithms proposed in the literature to do this. The various voltage scaling schemes can be classified as follows [35]:

1. Static Voltage Scaling (SVS) - different blocks and subsystems are given different, fixed supply voltages. Higher supply voltage is used for blocks falling on the critical path of the circuit and lower supply voltage on non-critical paths [72]. If two voltages are used, it is called dual-V_{DD} technique and when more than two supplies are used, it is called multi-V_{DD} technique.

2. Multi-level Voltage Scaling (MVS) - is an extension of the static voltage scaling case where a block or subsystem is switched between two or more voltage levels. Only a few, fixed, discrete levels are supported for different operating modes.

3. Dynamic Voltage and Frequency Scaling (DVFS) - an extension of MVS where a larger number of voltage levels are dynamically switched to follow changing workloads. The highest V_{DD} is used when the functional block is required to perform at the fastest frequency. When the performance demand is less, a lower V_{DD} is used for the same block [72]. The voltages must be limited to the range over which delay and voltage track monotonically [35]. A minimum clock speed that meets the workload requirements is determined and then the lowest supply voltage that will support the clock speed is determined [35]. Articles [14, 28, 45, 46, 67, 69] talk about different techniques used for employing Dual/multiple supply voltage scaling.

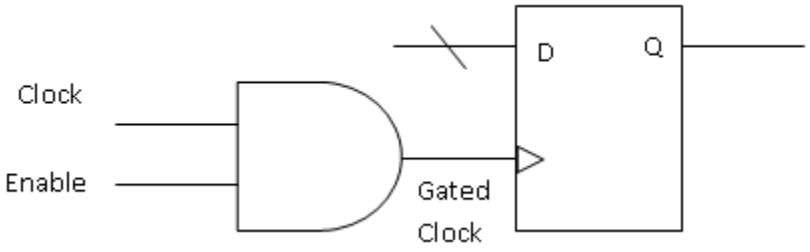

Figure 2.6: Clock gating of a flip-flop.

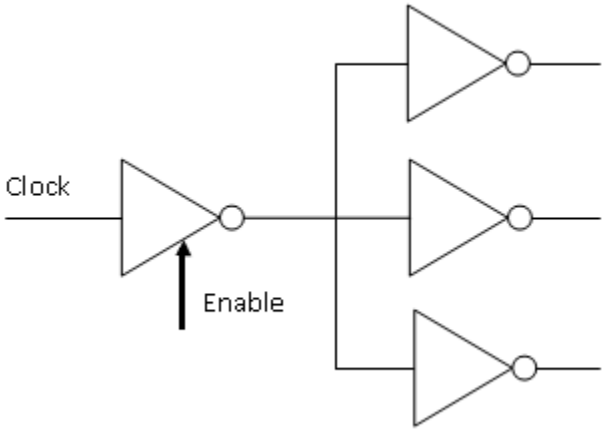

Figure 2.7: Clock gating of a clock-tree.

4. Adaptive Voltage Scaling (AVS) - an extension of DVFS where a control loop is used to adjust the voltage. A feedback system is implemented between the voltage scaling power supply and delay-sensing performance monitor [35].

Also, the supply voltage scaling used to reduce dynamic power increases the gate delay. To mitigate this performance degradation, the threshold voltages of the transistors are also lowered with the supply voltage. This increases the leakage currents in the CMOS devices.

2.2.4 Clock gating

In synchronous CMOS circuits, at the block level if the clock is gated to the functional blocks, the inactive blocks are effectively turned OFF with the stop in the clock pulse. This transition avoids the switching of the nodes in inactive blocks of the system while maintaining

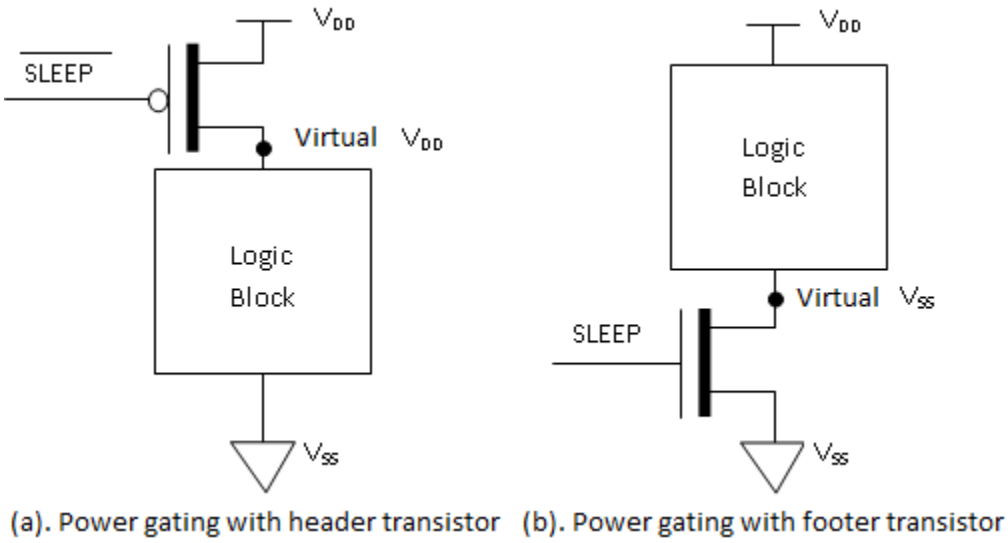

Figure 2.8: Power gating implementation using a header and a footer sleep transistors.

their logic values [72, 73]. Consider the circuit in Figures 2.6 and 2.7. When the enable is 1, the circuit works as usual. When it is 0, the clock signal is cut off from the circuit and the switching activity in the flip-flop or clock-tree is stopped. Clock gating decreases the switching activity in the flip-flops, gates in the fanout of the flip-flops, and the clock-tree and thus, decreases the power. The performance impact of clock gating is small [73].

2.2.5 Power gating

Power-gating is a technique used to reduce the subthreshold leakage power of CMOS circuits. It does not reduce gate leakage current [39]. A high-V_t pMOS transistor (header) or a high-V_t nMOS transistor (footer) controlled by a sleep bar or sleep signal is used to isolate the supply rails, V_{DD} or V_{SS}, respectively, from the rest of the logic when operating in sleep mode. This prevents the floating nodes at the logic outputs thus stopping the flow of short circuit currents into the active blocks connected to the power-gated block outputs [72, 81]. The two implementations of power gating are shown in Figure 2.8. Usually, any one of the schemes is implemented. Largely, pMOS header is used as compared to nMOS footer to reduce the leakage currents.

The sleep transistor has to be designed carefully so that the voltage drop across that is not very large during the active state. This ensures that the effective supply voltage to the logic block is maintained at V_{DD} [73]. Also, there is a dynamic power consumption associated with the turning ON/OFF of the sleep transistor. The power savings associated with the stand-by mode of a power gated logic block should be more than the power consumed while turning-ON/OFF of the sleep transistor [73]. The grouping of logic blocks for power gating implementation is also important. We should make sure to minimize the number of sleep transistors used while increasing the leakage energy savings [73]. Several algorithms have been published in the literature to address this problem.

The high-V_t sleep transistors decrease the subthreshold leakage currents. In technologies less than 60nm, gate leakage is expected to become larger than the subthreshold leakage current [39, 72]. In footer transistor implementation, once the logic block output has settled to V_{DD}, the footer induces reverse biased gate leakage current due to its bias condition. Also, nMOSFETs connected to primary inputs are other sources of gate leakage current (input gate leakage current). Assuming that the primary inputs are driven by logic blocks that are active in, the footer induces large reverse biased gate leakage current (70% of forward biased one [39]) due to the large voltage difference between its gate and source/drain. The reverse biased gate leakage current of pMOSFET is negligible; implying that a header would be less leaky. In the header implementation, the nMOS transistors driven by logic 1 inputs experience a forward biased input gate leakage current which might imply that a footer which passes a reverse biased current is superior to a header [39, 60]. Though different mechanisms of leakage exist in each of the power gating schemes and to different extents, when total leakage current is considered, using a header transistor proves to be beneficial [39].

A header is recommended when a switch is sized such that lower delay penalty can be tolerated, but a footer is superior when delay penalty is larger [39]. When a low-V_t sleep transistor is used instead of a high-V_t, a footer is always better than a header. This is because the sub-threshold leakage current of a switch now makes up large portion of total leakage

currents due to its exponential dependency on the threshold voltage, and the sub-threshold leakage current is smaller in a footer than in a header due to the smaller size of a footer with the same delay penalty [39].

The power consumption of power gating circuits can be further reduced by controlling primary inputs. By providing logic 1 to all primary inputs of a power-gated circuit with a footer (similarly logic 0 in the case of a header), input gate leakage currents can be virtually eliminated. However, gate leakage current of a footer goes up due to elevated voltage of a virtual ground. This is because the additional transistors in input control circuits induce current that flows through a footer in addition to the current from the logic block [39]. Also, when the logic is cut off from the ground using a leakier footer transistor, the difference in the potentials between this virtual ground and the V_{SS} will cause ground noise causing signal integrity problems [35, 81].

Thus, depending on the application requirements, a header or a footer sleep transistor is used. For designs with greater area constraints, an nMOS footer is a better option. Availability of less leaky transistor design in a given technology is important for power gating implementation. A thick oxide transistor can be used where an increase in area can be tolerated. A high-V_t thin oxide sleep transistor is preferred where area has higher priority; the higher leakage of the thin oxide can be reduced by longer gate and/or reverse body bias techniques [35].

2.2.6 Dual/multi-threshold designs

The threshold voltage of a transistor is given by

$$V_{th} = V_{th0} + \gamma(\sqrt{|-2\phi_F + V_{SB}|} - \sqrt{|2\phi_F|}) \tag{2.11}$$

where V_{th0} is the zero source-bulk bias threshold voltage, ϕ_F is the substrate Fermi potential, γ is the body-effect coefficient. This shows that reverse body biasing a transistor increases threshold voltage and thus decreases the leakage currents [73].

In this method, as with dual-V_{DD} technique, slower transistors with high-V_{th} are used in non-critical paths and low-V_{th} transistors are used in critical paths to achieve desired performance. Increasing the channel length or its doping density increases the threshold voltage. A thicker oxide layer can be used to increase the threshold voltage., but this reduces only subthreshold leakage and not the gate direct tunneling [72].

2.2.7 Variable threshold CMOS (VTCMOS)

In this technique a zero body bias is used in the active mode and a high reverse body bias is applied in the standby mode to increase the threshold voltage, thereby decreasing the standby leakage. This capability of reverse body biasing technique to reduce the leakage current lowers as the technology scales [36, 72]. This is because of the exponential increase in the band-to-band tunneling current at the source-substrate and drain-substrate junctions in scaled technologies [36, 72].

Authors in [91] have described a technique which reduces the standby leakage by using high-V_{th} devices and during active mode a forward body bias is applied to reduce the V_{th}. This reduces the short channel effects, limiting the reduction of the leakage current. It has been shown that forward body biasing with high-V_{th} and reverse body biasing reduces leakage 20 times while only reverse body biasing with low-V_{th} reduces leakage only 3 times [72].

2.2.8 Dynamic threshold scaling

Body biasing techniques are used to change the threshold voltage according to the performance requirements. When performance demand is low, clock frequency is lowers and the threshold voltage is increased to reduce run-time leakage. In standby mode the threshold voltage is increased to its maximum limit to reduce the standby leakage. A zero

body bias with lowest possible threshold is used when the performance requirements are the highest [72].

2.2.9 Variable supply and threshold voltages

The threshold voltage and the supply voltage can be varied simultaneously to get required power and timing efficiency. The ON current of a transistor is given by

$$I_{DS} = \mu C_{ox} \frac{W}{L} \frac{(V_{GS} - V_{th})^2}{2} \tag{2.12}$$

where μ is the mobility, C_{ox} is the gate capacitance, V_{th} is the threshold voltage, V_{GS} is the gate-to-source voltage. As we reduce the supply voltage, the input voltage swing, V_{GS}, is also reduced. To maintain the performance, the current drive, I_{DS}, has to be maintained. For this the threshold voltage also has to be decreased. If we use low-V_{DD} and low-V_{th}, the power dissipation will increase because lowering V_{th} results in exponential increase in the subthreshold leakage current. Hence, we require lower V_{DD} and higher V_{th} on non-critical paths to reduce power and maintain performance on the critical paths. We can also use dynamic supply scaling and dynamic threshold scaling together to reduce power [73, 74].

2.2.10 Transistor stacking

Using a stack of transistors in the OFF state reduces the leakage current because of negative gate-to-source biasing, body-effect induced threshold voltage increase, and increased threshold voltage due to reduced drain-to-source voltage. The time required for the leakage current in the transistor stacks to converge to its final value depends on the intermediate node capacitances and the threshold voltages of the transistors [99]. Turning off more than one transistor in the stack increases the source voltage of the stack and thus, reverse biases the source. Consider a stack of four nMOS transistors as shown in Figure 2.9 [72]. All four of these transistors are in the OFF state. If only one transistor is OFF, the voltage at the

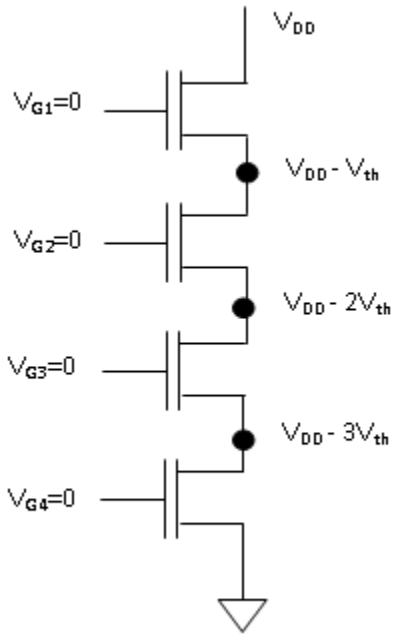

Figure 2.9: A stack of nMOS transistors.

source node of that transistor will be zero as the other transistor are ON and act as a short circuit path to ground. When all the transistors are OFF, the source voltages of the OFF transistors, except the one directly connected to the ground, will be non-zero causing the self-reverse biasing of the transistors. This reverse bias reduces the leakage currents through the transistors [72].

2.2.11 Minimum leakage vector method

The leakage current of a logic block depends on the inputs as this determines the number of ON and OFF transistors. The minimum leakage is seen when the resistance between the supply and ground through the transistors is maximum [73]. Hence, there is a combination of inputs for which the leakage power is the least. This input combination is called the minimum leakage vector (MLV). Various algorithms have been described in the literature to find the MLV of a circuit [2, 3]. After determining the MLV for a logic block, it is applied to its inputs in the standby state to reduce the standby leakage.

2.3 Conclusion

A reduction of power dissipation is important to save energy. However, it is not the only concern. Lower power dissipation decreases the heat generated within the chip, in between the individual transistors. This reduction in heat allows further increase in the level of integration and thus, contributing towards further reduction of size. Besides, the lower dissipation allows the packaging to be done using light and inexpensive plastics, rather than much costlier and heavier ceramic cases. Further, in larger devices like PCs and servers, fans are provided to cool the system during operation. This is not practical in mobile devices like laptops and personal digital assistants (PDAs). Hence, to increase the portability of the electronic devices power dissipation must be lowered. This allows their manufacturing to be done at lower costs and also reduces the chip area.

Chapter 3

Overview of Static Timing Analysis Algorithms

Static timing analysis has been first described in [4, 20, 24, 34, 55]. The author in [55] describes a linear time algorithm that finds the shortest/longest paths in order of their total delay. Static Timing Analysis (STA) is a technique to verify whether a digital design meets the given timing specifications. An alternate approach used to verify the timing is timing simulation, which can verify the functionality as well as the timing of the design. The term timing analysis is used to refer to either of these two methods - static timing analysis, or timing simulation, [7]. Timing Analysis (TA) is defined by [24] as "a design automation program which provides an alternative to the hardware debugging of timing problems".

In STA, the analysis of the design is carried out statically and does not depend upon input vectors. In simulation based timing analysis, a stimulus is applied on primary inputs, resulting behavior is observed and verified, then time is advanced with new inputs applied, and the new behavior is observed and verified, [7]. This makes STA faster than the simulation based methods for large circuits.

Any synchronous circuit can be seen as consisting of combinational and sequential blocks such that the combinational block receives the data from a clocked flip-flop and outputs data to another clocked flip-flop, Figure 3.1. The input and output flip-flops can operate at different clocks. The timing analysis program verifies whether all combinational paths meet the required timing specifications, that is, that data signals arriving at storage elements do not cause any set-up or hold-time violations. The setup time is the time taken by the the data signal to arrive at a flip-flop. This should be within the given clock period. The hold time is the time for which the data should not change after a clock transition so that there is no unexpected pass-through of data through a flip-flop. That is, the data needs to be

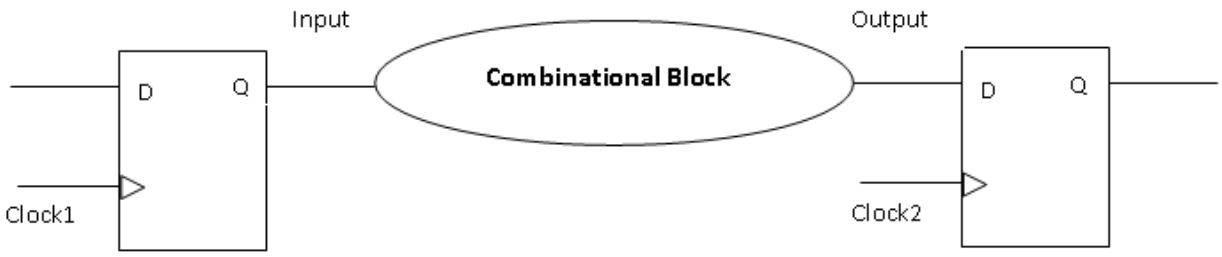

Figure 3.1: Generalization of a synchronous circuit.

held constant for at least a minimum time ensure that a flip-flop captures the intended data correctly. These checks ensure that the proper data is ready and available for capture and latched in for the new state [7].

The timing analysis of a circuit produces 'slack' at each gate to provide a measure of any timing violations. The slack is defined in [24] as the difference between the required arrival time and the actual arrival time. Negative slack indicates that the data signal takes 'slack' units longer time to arrive at the storage elements than the given specifications. A positive slack corresponds to an early signal which comes at the storage element 'slack' units earlier than the specified timing. In order to meet the specifications, the data signal should not be too late that it does not meet the set-up time requirements and not too fast that it violates the hold time of the previous signal. Thus, the entire design is analyzed and the required timing checks are performed for all possible paths of the design. Thus, STA is a complete and exhaustive method for verifying the timing of a design [7] as opposed to the simulation based timing analysis which does not verify the parts of the circuit which are not excited by input stimulus.

The basic timing analysis algorithm is analogous to the Program Evaluation and Review Techniques (PERT) [68] algorithm when run non-probabilistically. The application of PERT to logic design is described in [47]. A combinational circuit can be viewed as a Directed Acyclic Graph (DAG). In this DAG, the various nodes represent the logic blocks and edges represent the interconnects. For example, the combinational circuit shown in Figure 3.2 can be redrawn as a DAG shown in Figures 3.3 and 3.4. In these figures, each node is associated

with a weight, which represents the largest delay of the gate, and the edges do not have any weights associated with them. The delays of the gates can be specified as minimum and maximum delays or as rise and fall delays or in any other manner. Always the maximum of the delays is considered for the longest path analysis and the minimum for the shortest path analysis. When the interconnect delays are specified, then the maximum/minimum of the sum of a node and its output edge weights is considered. In the proposed work, longest path analysis is considered. The primary inputs (PI) and primary outputs (PO) as assumed to emerge from the source and sink nodes respectively. The source and sink nodes do not have any weights associated with them.

In the static timing analysis program proposed in [24], we traverse from the primary inputs to the primary outputs through each gate such that we assign the longest path delay through this gate from the primary inputs as its weight. Once the weights of all nodes are assigned, we traverse in the reverse direction, i.e., from the primary outputs to the primary inputs to assign the slack of each node. The slack of each node is calculated as the minimum difference between the required arrival time at the gate output so that the primary outputs meet the timing constraints and the actual arrival time of the gate output. This is best described by Figure 3.3.

In Figure 3.3, PI represent the primary inputs, PO the primary outputs and A through J represent logic gates. Each logic gate is characterized by a pair of delays, $(min_{delay}, max_{delay})$, which represent the minimum and maximum delays through the gate, respectively. The actual arrival times, required arrival times and the slacks of each gate are computed. The results are recorded at each node. For example, for node E, minimum delay is 1, maximum delay is 2, actual arrival time is 7, actual arrival time is 11 and the slack is 4. These values are noted at the top of node E in the DAG of Figure 3.3 as {(1,2), 7, 11, 4}.

Now, consider gate E. We have to find the longest path delay through this gate. We begin with the primary inputs and assume that the start of the algorithm represents the time 0 units. Gate E has outputs of gates A and B as its inputs. It is observed that output of gate

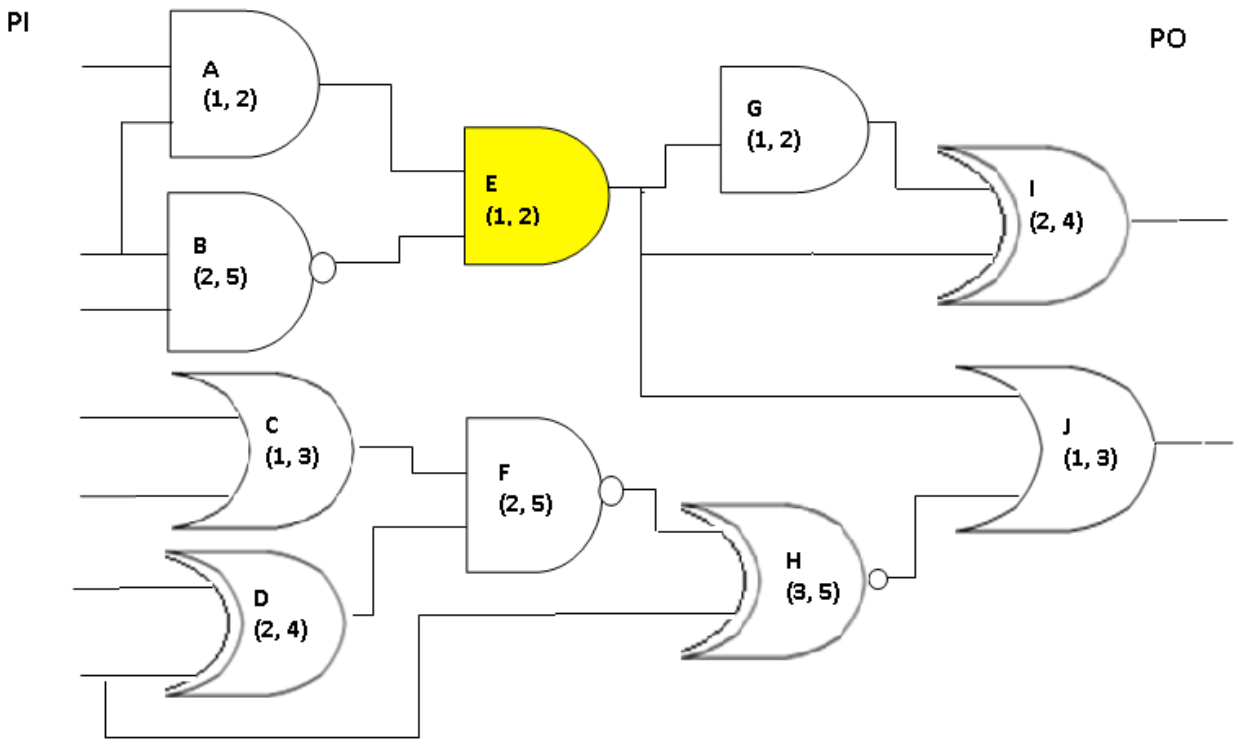

Figure 3.2: Example of a combinational circuit

A arrives 3 units earlier than gate B, which arrives at 5 time units from the start. Hence, the longest path from the primary inputs to the inputs of gate E would be 5 time units. Now, the maximum delay of gate E is 2 units. Hence, the maximum time from the start till the time that the output of E takes to stabilize is the sum of 5 and 2, i.e., 7 units. Hence, the path delay till the output of node E from the primary inputs is 7 units. The path delays of each node are calculated thus. From the outputs of E we can take two different paths to the primary outputs, shown in green and red colors. But, the path that goes through the gate G and I, green, yields the longest path with path delay of 13 units. The longest path delay of the circuit, shown in yellow, is 17 units.

Now, to find the slacks of each gate, suppose the required arrival times of both the primary outputs are 17, the longest path delay of the circuit. The output of node I arrives at 13 units, i.e., its slack is $17 - 13 = 4$ units. Similarly, the output of node J arrives at 17 units and its slack is 0 units. Since the maximum delay of node I is 4, the inputs of I should

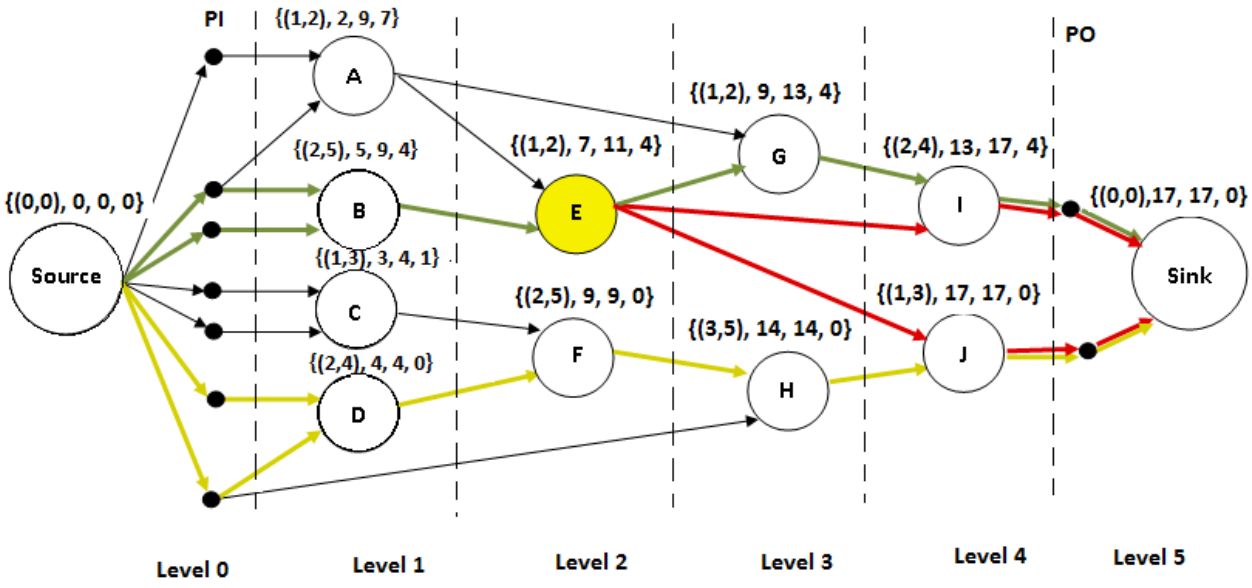

Figure 3.3: Traditional STA.

arrive at 13 units of time. The path delay of gate G is 9, i.e., its actual arrival time is 9 units. Hence, the slack of gate G is $13 - 9 = 4$ units. Now, we can reach node E from node J, or node I or node G from the sink node. Required arrival time at node J's input is 14 units, and node E's output arrives at 7 units, hence, node E's slack is 7 units. But node I's required arrival time is 13 units and this makes the slack of node E 6 units. Now, node G's required arrival time is 11 units and hence, the slack of node E is 4 units. Thus, we obtain 3 values for the slack of node E. We take the minimum of these three values because we need to meet all the three timing requirements. Hence, the actual slack of gate E is 4 units.

The slacks of each gate are found as described above. It is observed that the gates on the longest path of the circuit, also called the critical path, have 0 slack when the required time at the primary output is equal to the critical path delay. The two primary outputs in the example are assumed to have the same required arrival time. But, in practice, they can have different arrival times. This adds minor changes to the slack calculation. If the required arrival time at node I is 15 and node J remains at 17. Then the slack of gate I will be 2 units instead of 4. This algorithm requires us to traverse the graph two times, once

forward to calculate the path delays and once backwards to calculate the gate slacks. Thus, the time taken is proportional to the number of gates and the complexity of the algorithm is linear, $O(n)$ where n is the number of gates.

3.1 Algorithm 1

We use the $O(n)$ slack calculation algorithm proposed in [40, 43, 41, 42, 44] to find out the gate slacks for a given circuit. The aim is to find the find the longest path delay from the primary inputs to the primary outputs through a given gate. And then the gate slacks are calculated as the difference of critical path delay and the longest path delay through the gate. The step by step approach to implement this algorithm is described below using the example shown in Figure 3.2. The directed acyclic graph (DAG) for this circuit is shown in Figure 3.4.

A given circuit is levelized. We start from the primary inputs and traverse to the outputs of each gate such that we always choose the longest delay path. Simultaneously we traverse in the opposite direction to find the longest delay path from the primary outputs to the respective gate output. The sum of the longest delay from the primary inputs to the gate output and the longest delay from the primary outputs to the gate output will give the longest path delay through that gate. Let $T_{PI}(i)$ be the longest time for the data signal to arrive from a primary inputs to the output of gate i and $T_{PO}(i)$ be the longest time for the data signal to reach a primary outputs from the output of gate i. The delay of the longest path [41, 42, 43] through gate i is given by,

$$D_{p,i} = T_{PI}(i) + T_{PO}(i) \tag{3.1}$$

The critical path delay of the circuit is given by

$$T_c = Max\{D_{p,j}\} \; \forall \; gate \; j \tag{3.2}$$

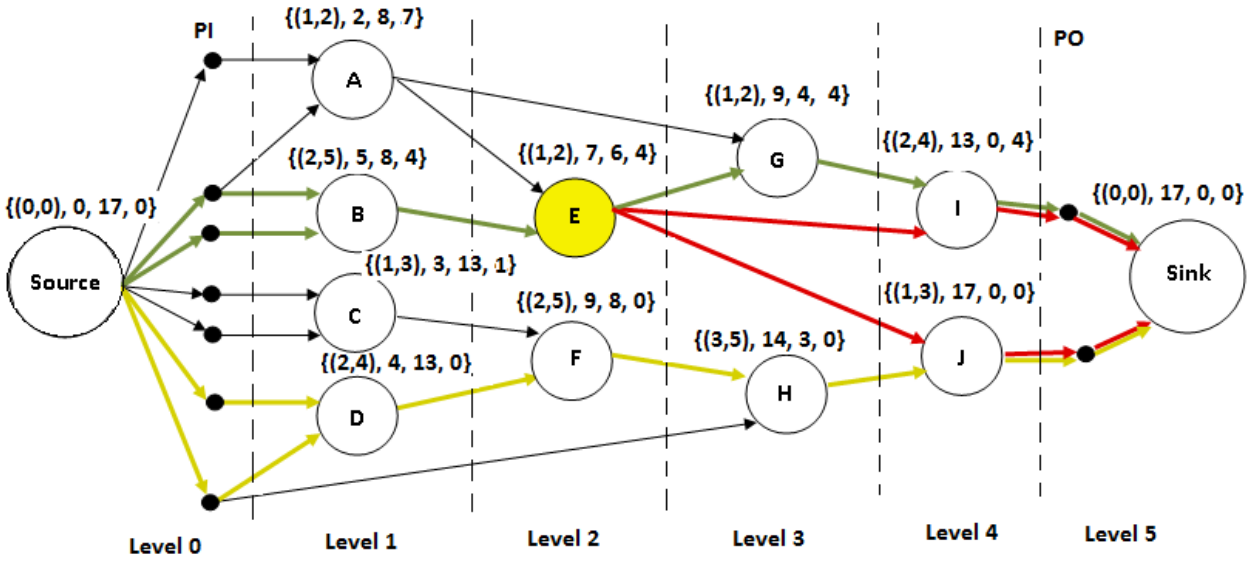

Figure 3.4: New STA.

The slack for gate i is then calculated as

$$S_i = T_c - D_{p,i} \tag{3.3}$$

Consider Figure 3.4. Here, PI represent the primary inputs, PO the primary outputs and A through J represent logic gates. As in Figure 3.3, for node E, minimum delay is 1, maximum delay is 2, T_{PI} is 7, T_{PO} is 6 and the slack is 4. These values are noted at the top of node E in the DAG of Figure 3.3 as {(1,2), 7, 6, 4}. Note that in this case, we record the path delay from the primary outputs as well. Also the slack of gate E is the same as the slack obtained from the algorithm described in [24].

To begin with we levelize the circuit. For this, the source node and primary inputs are considered to be in Level 0. Then all the nodes that have primary inputs as their inputs are grouped in Level 1. So, nodes A, B, C, D and H form Level 1. Now the gates to which the outputs of Level 1 gates are fed are considered to be in Level 2. That is, nodes E, F, and G are in Level 2. Again, the outputs of Level 2 gates feed into Level 3 gates. Thus, nodes G, H, I and J form Level 3. Note that G falls in Levels 2 and 3. We take the maximum of both.

Hence, G falls in Level 3. Similarly, H also falls in Level 3. Then Level 4 gates are gates that have Level 3 gates as their inputs. This makes node I and J as Level 4. Finally, the primary outputs and the sink node form Level 5, the highest level. These levels are separated by dashed vertical lines in Figure 3.4.

Again, as in the previous algorithm, consider gate E. To find the longest path delay through this gate, begin with the primary inputs and assume that the start of the algorithm represents the time 0 units. Gate E has outputs of gates A and B as its inputs. It is observed that output of gate A arrives 3 units earlier than gate B, which arrives at 5 time units from the start. Hence, the longest path from the primary inputs to the inputs of gate E would be 5 time units. Now, the maximum delay of gate E is 2 units. Hence, the maximum time from the start till the time that the output of E takes to stabilize is the sum of 5 and 2, i.e., 7 units. Hence, the path delay till the output of node E from the primary inputs is 7 units. The path delays of each node are calculated thus. From the outputs of E we can take two different paths to the primary outputs, shown in green and red colors. But, the path that goes through the gate G and I, green, yields the longest path with path delay of 13 units. The longest path delay of the circuit, shown in yellow, is 17 units.

As we traverse from the primary inputs to the gate,concurrently, we also traverse from the primary outputs to the gate to find the longest path delay from the primary outputs to the gate output. Thus, to find the longest path delay from the primary outputs to the output of gate E, we start from the sink node and try to reach node E. We can reach E from node J or from node I or from node G . The path from node J, red line, results in a delay of 3 units at E, the maximum delay of gate J. Similarly the path from node I, red line, results in delay of 4 units at E. The path through node G, green line, starts from the sink, goes through node I, then through node G and then end at node E. The delay of this path will be 6, the sum of maximum delays of nodes I and G. Out of these three paths leading to node E, the longest path delay will be 8 units. The longest path delay from the primary inputs to the primary outputs through gate E is thus, $7 + 6 = 13$ units.

Now, to find the slacks of each gate, we subtract the longest path delay thus obtained from the critical path delay of the circuit. The critical path delay is found similarly and is 17 units. Thus, the slack of gate E is $17 - 13 = 4$ units.

Similarly, the slacks of each gate are found. Again, it is observed that the gates on the longest path of the circuit, also called the critical path, have 0 slack when the required time at the primary output is equal to the critical path delay. Also, the slacks of all gates that have the same longest path have the same slack. This is in contrast to the slacks derived from the previous algorithms because, here, slack is calculated as the difference of critical path delay and the longest path delay through a gate. G gates having the same longest path will have the same path delay, whereas the slack defined in [24] is the difference between the required arrival time and the actual arrival time each gate output. From Figures 3.3 and 3.4, we observe that the slacks of all gates derived in both ways are the same.

The algorithm described in the previous section requires complete forward and backward traversal. But this new algorithm requires only partial traversal of the graph, up to the node output to calculate the slack of one gate. In order to find the slacks of all the gates, we require complete traversal of the graphs in both directions, but to calculate the slack of each gate we need not wait until forward traversal is complete. Traversal in both directions can be done simultaneously. Thus, the time taken for this algorithm is lower than the time taken by the previous algorithm. The complexity of the algorithm is still linear, $O(n)$ where n is the number of gates, as the time is still proportional to the number of gates.

This algorithm finds the slacks corresponding to the longest path delay through a gate. Similarly, the slacks corresponding to the shortest path delay can be calculated by considering the minimum gate delay and the paths yielding the shortest delay. The shortest delay through the gate E considered in the earlier example would then be 3 units starting from the source and 1 from the sink. The shortest path problem is considered when we are required to check for the signals arriving earlier than the required time.

The linear complexity of this algorithm makes it effective for large machines in contrast to other techniques such as delay simulation, which requires large numbers of test patterns, and path tracing.

Chapter 4

Slack-Based Algorithm For Finding an Optimal Lower Supply Voltage

As discussed in Chapter 2, using multiple supply voltages is a widely used technique to reduce power consumption in electronic circuits. In such designs, a level converter is required to boost the logic '1' level at the output of a low voltage gate, to the high voltage logic '1' level, when it has a high voltage gate at its fan-outs. If we never allow a low voltage gate to drive a high voltage gate, then a level converter is not required. Level converters are discussed in detail in Chapter 7. The authors in [56, 73] describe a method to find the lower supply voltage in a dual-voltage design. The power reduction ratio R is found as the ratio of power dissipated in a dual-voltage design and a single voltage design with the higher supply voltage and is expressed as

$$R = 1 - (\frac{C_{V_L}}{C})\{1 - (\frac{V_L}{V_H})^2\} \tag{4.1}$$

where C_{V_L} is the total capacitance of the low voltage cells and C is the total capacitance of the single voltage circuit [56]. If $p(t)$ is the number of paths whose delay is t when $V_L = V_H$ and path-delay t is the path-delay normalized to the cycle time, then the power reduction for various distribution functions of $p(t)$ is shown in Figure 4.1 [56].

Figure 4.1 shows that the minimum value of R depends on $p(t)$ and it is minimum when V_L falls between $0.6V_H$ and $0.7V_H$. This means that V_L should always be between $0.6V_H$ and $0.7V_H$ to minimize power dissipation.

Same authors in [21] derive a relationship between the optimum voltage ratios in multiple-V_{DD} circuits. For dual-supplies $\{V_1, V_2\}$, $\frac{V_2}{V_1} = 0.5 + 0.5\frac{V_{th}}{V_1}$. For three supplies $\{V_1, V_2, V_3\}$, $\frac{V_2}{V_1} = \frac{V_3}{V_2} = 0.6 + 0.4\frac{V_{th}}{V_1}$. And for four supplies $\{V_1, V_2, V_3, V_4\}$, $\frac{V_2}{V_1} = \frac{V_3}{V_2} = \frac{V_4}{V_3} = 0.7 + 0.3\frac{V_{th}}{V_1}$. The authors claim that this rule of thumb gives the optimum supplies which reduce power to a range within 1% of the optimum minimum. These results show that the

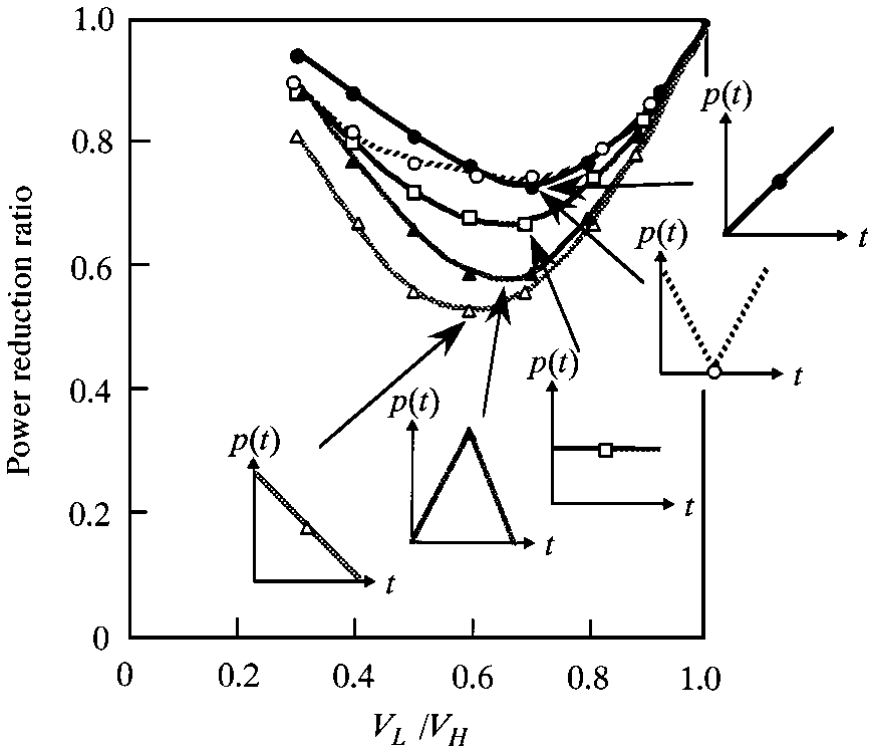

Figure 4.1: Power reduction versus ratio of low voltage and high voltage, [56, 73].

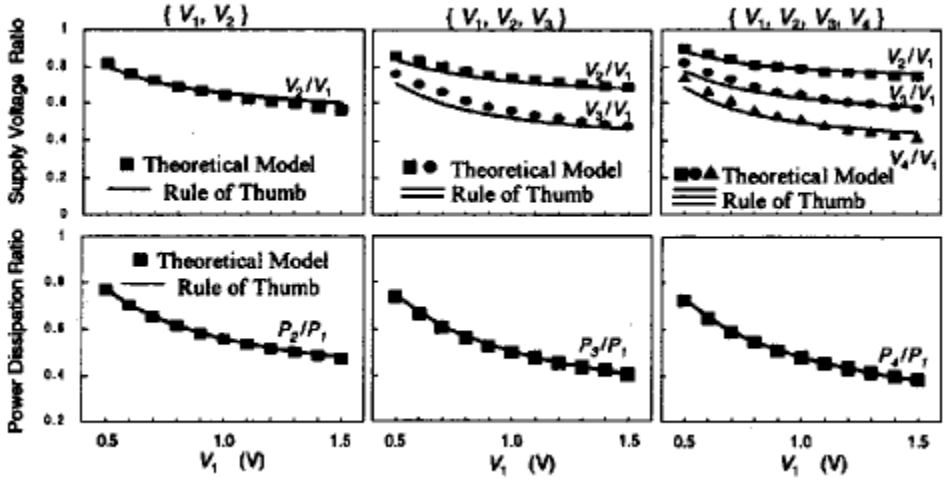

Figure 4.2: Optimum supply voltages and minimum power dissipation for multiple V_{DD} values, [56, 73].

power savings tend to saturate as the number of supplies is increased and also that the savings decrease as the supply voltage is scaled and as V_{th}/V_{DD} is higher.

An algorithm to find the optimum V_{DD} for each node is described in [10]. It is estimated by finding

$$max \sum_{v} (V_H^2 - V_L^2) \cdot C(v) \cdot E(v) \cdot k_v \tag{4.2}$$

where, $k_v = 1$, if $V_{DD}(v) \leq V_L$ or 0, if $V_{DD}(v) > V_L$.

The works described in [10, 50, 84] claims that the optimal value of V_L for minimizing total power is 50% of V_H. Several linear programming algorithms have been proposed in [8, 32, 33, 41, 79] to select the optimum value for the lower voltage. Authors in [101] describe an algorithm to find the lowest feasible supply voltages according to their slacks from a set of given voltages. Another slack-based algorithm has been proposed in [41]. The authors have proposed a method to find the optimum low voltage value based on the estimation of minimum energy state for the circuit.

In our work, we propose an algorithm to find the lower supply voltage using the gate slack. The slack of a gate is defined as the difference of the critical path delay of the circuit and the delay of the longest path through a gate. Thus, each gate has its own slack and the gates with same slack fall on the same path unless there are two paths with equal delays.

Before describing our algorithm for assigning low-voltage, we propose three theorems to categorize the gates in any given circuit based on the gate slacks. The slack of each gate in a given circuit is plotted against the difference of its respective low voltage delay and the high voltage delay, as shown in Figure 4.3.

Theorem 4.1 *The gates that fall above the* 45^o *line in the 'delay increment versus slack' plot cannot be assigned lower supply voltage without violating the positive slack constraint. These gates fall in group 1.*

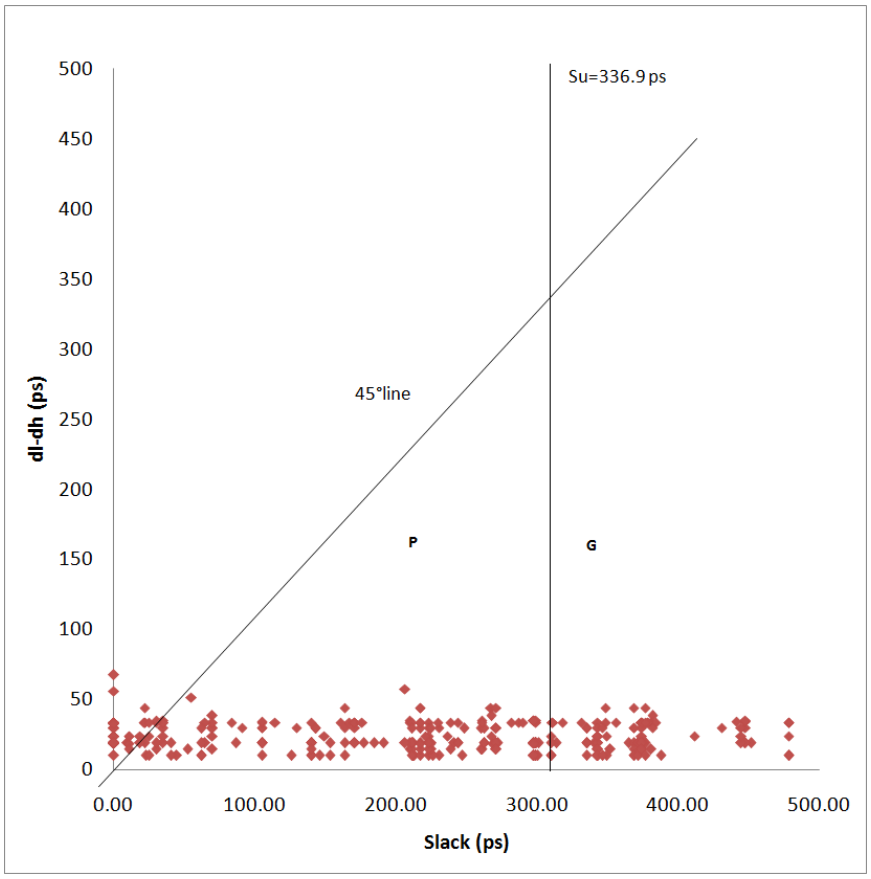

Figure 4.3: Delay increment versus slack for gates of c880 circuit, $V_H = 1.2V,\ V_L = 0.58V$.

Proof

The slacks used in the 'Delay increment versus slack' plot are the slacks when all gates are at higher supply voltage. That means that the gate delays are high voltage delays, d_{hi}, where i is the gate number.

When we put a gate to low voltage, its delay becomes the low voltage delay, d_{li}, where i is the gate number. Then the slack of the gate, is the difference between the critical path delay and the sum of gate delays through the longest path through the gate. This can be written as in the following equation

$$slack_h = T_c - \sum_{i=1}^{n} d_{hi} \tag{4.3}$$

where n is the number of gates in the circuit. When this gate is put to low voltage, its slack will become

$$slack_l = T_c - (\sum_{i=1}^{n} d_{hi} - d_{hi} + d_{li}) \tag{4.4}$$

$$\Rightarrow slack_l = T_c - \sum_{i=1}^{n} d_{hi} - (d_{li} - d_{hi}) \tag{4.5}$$

$$\Rightarrow slack_l = slack_h - (d_{li} - d_{hi}) \tag{4.6}$$

The above equation suggests that the slack is reduced by $d_{li} - d_{hi}$ amount when a gate i is put to low voltage. Now suppose,

$$slack_h < d_{li} - d_{hi} \tag{4.7}$$

When we put the gate i in low voltage, the new slack will be

$$slack_l = slack_h - (d_{li} - d_{hi}) < 0 \tag{4.8}$$

But a slack less than zero implies that the circuit fails to satisfy the timing constraints. Hence, we cannot afford to put the gates satisfying the condition represented by the inequality 4.8 to low voltage. This can be interpreted from the graph shown in Figure 4.3. When a point falling above the 45^o line is put in low voltage, its slack is lowered and goes to the left side of the y-axis if the delay increment is larger than the original slack. Thus, the gates that fall above the 45^o line in the 'Delay increment versus slack' plot cannot be assigned lower supply voltage without violating the positive slack constraint.

We define two new variables introduced in a recent paper [41], *β and S_u*. S_u, the upper slack time is the lower bound of slacks of the gates which can be unconditionally assigned low voltage without affecting the critical timing of the circuit. These include gates with a large slack, i.e., shorter paths which do not affect the critical path time even when their delay is increased as result of assigning all the gates in these paths a lower supply voltage.

Theorem 4.2

$$S_u = \frac{(\beta_{max} - 1)}{\beta_{max}} \times T_c$$

where β is the ratio of the low voltage delay and the high voltage delay of each gate and T_c is the critical path delay.

$$\beta_i = \frac{d_{li}}{d_{hi}}$$

where d_{li} is the low voltage delay and d_{hi} is the high voltage delay of gate 'i' *. The maximum value of β, β_{max}, will give us the lower bound on the gate slacks. We represent the number of gates with slacks greater than S_u in a given circuit as G and these gates are grouped under group 2.*

Proof

Consider a path with a large slack. If we put all the gates in this path to low voltage, the new slack of each gate will be

$$slack_l = slack_h - (\sum_{i=1}^{n} d_{li} - \sum_{i=1}^{n} d_{hi}) \tag{4.9}$$

$$\Rightarrow slack_l = T_c - \sum_{i=1}^{n} d_{li} \tag{4.10}$$

Now, if we consider the definition of β_i, we notice that

$$d_{li} = \beta_i d_{hi} \tag{4.11}$$

then

$$\sum_{i=1}^{n} d_{li} = \sum_{i=1}^{n} \beta_i d_{hi} \tag{4.12}$$

If β_{max} is the maximum of all β_i's, then

$$\sum_{i=1}^{n} d_{li} \leq \beta_{max} \sum_{i=1}^{n} d_{hi} \tag{4.13}$$

with equality occurring when all β_i's are equal to β_{max}. Hence, we see that

$$slack_h \geq T_c - \frac{\sum_{i=1}^{n} d_{li}}{\beta_{max}} \tag{4.14}$$

To prevent negative slacks in this path we need to prevent the new low voltage slack from becoming less than zero. Which means,

$$slack_l = T_c - \sum_{i=1}^{n} d_{li} \geq 0 \tag{4.15}$$

$$\Rightarrow \sum_{i=1}^{n} d_{li} \leq T_c \tag{4.16}$$

$$\Rightarrow -\frac{\sum_{i=1}^{n} d_{li}}{\beta_{max}} \geq -\frac{T_c}{\beta_{max}} \tag{4.17}$$

$$\Rightarrow T_c - \frac{\sum_{i=1}^{n} d_{li}}{\beta_{max}} \geq T_c - \frac{T_c}{\beta_{max}} \tag{4.18}$$

$$\Rightarrow slack_h \geq \frac{(\beta_{max} - 1)}{\beta_{max}} \times T_c \tag{4.19}$$

The slacks of the gates that can be unconditionally assigned low voltage without affecting the critical timing of the circuit should satisfy the condition specified by the inequality 4.20. Since S_u is the lower bound of these slacks,

$$S_u = \frac{(\beta_{max} - 1)}{\beta_{max}} \times T_c \tag{4.20}$$

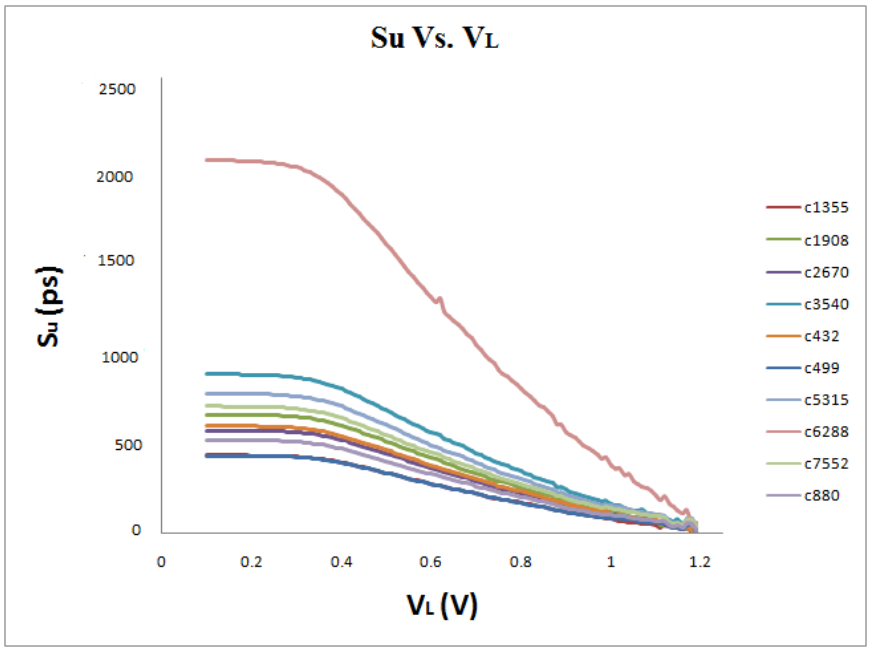

Figure 4.4: S_u Vs. V_L for ISCAS'85 benchmark circuits.

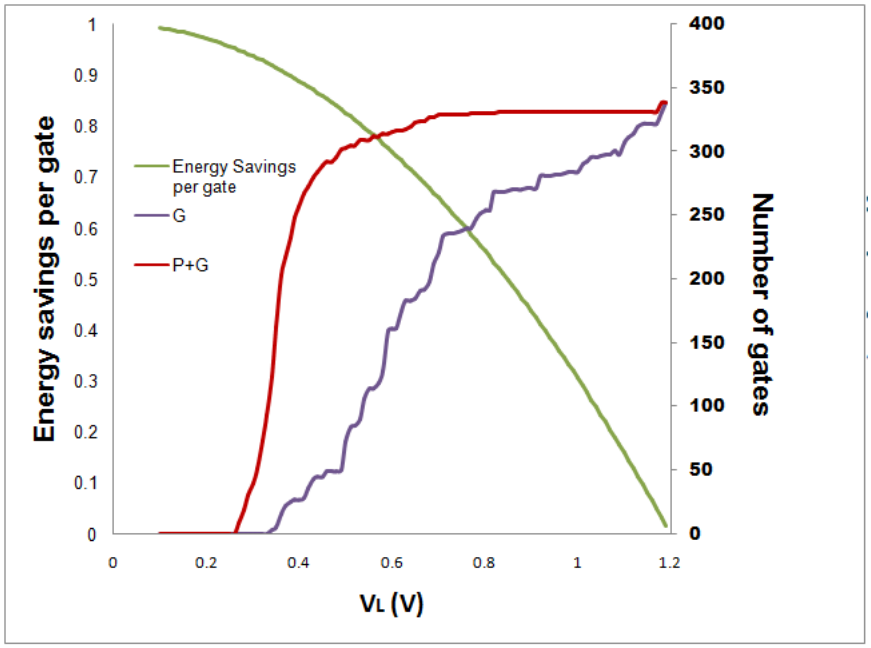

Figure 4.5: Energy savings per gate vs. V_L for c880 when level converters are not allowed. G and P+G show the number of gates. Energy savings per gate is $(V_H^2 - V_L^2)/V_H^2$.

The variation of S_u with V_L is shown in Figure 4.4. It is observed that when V_L is equal to V_H, 1.2V in this case, S_u is '0'. As V_L is decreased, S_u increases and moves rightwards

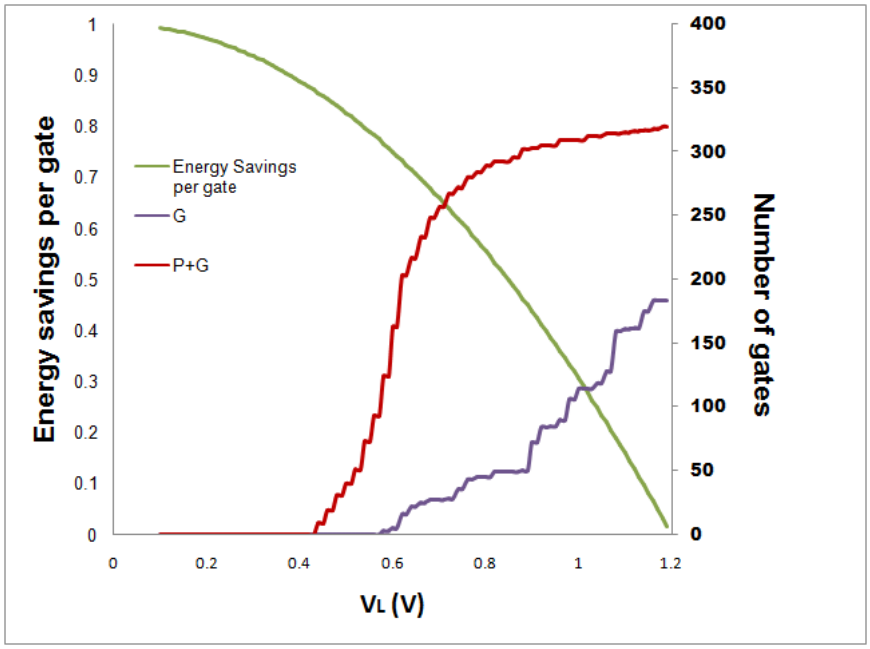

Figure 4.6: Energy savings per gate vs. V_L for c880 when level converters are allowed. G and P+G show the number of gates. Energy savings per gate is $(V_H^2 - V_L^2)/V_H^2$.

in the 'delay increment vs. Slack' graph. This decreases the number of gates whose slack is greater than S_u, this can be seen in Figures 4.5 and 4.6 as 'G'. Thus, the number of gates that can be put to low voltage decreases as V_L decreases.

The number of gates whose slack fall below the 45^o line in the 'delay increment versus slack' plot of Figure 4.3 and whose slacks are less than Su are represented by P. These gates are considered as the group 3 gates.

Theorem 4.3 *There exist a group of gates within P which can be assigned lower supply voltage simultaneously without violating the positive slack constraint satisfying the condition*

$$\sum_{i=1}^{m} y_i \leq min\{slack_{hi}\}$$

where y_i is the difference of the low voltage delay and the high voltage delay of each gate.

$$y_i = d_{li} - d_{hi}$$

where d_{li} is the low voltage delay of gate 'i' *, d_{hi} is the high voltage delay of gate* 'i' *and $slack_{hi}$ is the slack of the gate* 'i' *when it is in high voltage.*

Proof

Consider a group of m gates. When we put all of these gates to low voltage, all the new low voltage slacks should be positive to avoid timing violation. Which means,

$$\{slack_{l1}, slack_{l2}, ..., slack_{li}, ..., slack_{lm}\} \geq 0 \tag{4.21}$$

If we assume that all the gates lie on the same path, the slack of each gate will reduce by an amount $\sum_{i=1}^{m} y_i$ where $y_i = dl_i - dh_i$. This is the greatest amount by which the slack of each gate will change. If the gates lie on uncorrelated paths the amount of slack reduction for each gate will be less than $\sum_{i=1}^{m} y_i$. Hence, the least slack any gate in this group can have will be

$$\Rightarrow slack_{li} \geq slack_{hi} - (\sum_{i=1}^{m} d_{li} - \sum_{i=1}^{m} d_{hi}) \geq 0 \tag{4.22}$$

for all i belonging to the selected group of m gates.

$$\Rightarrow slack_{li} \geq slack_{hi} - \sum_{i=1}^{m} y_i \geq 0 \tag{4.23}$$

for all i belonging to the selected group of m gates.

$$\Rightarrow slack_{hi} \geq \sum_{i=1}^{m} y_i \tag{4.24}$$

for all i belonging to the selected group of m gates.

$$\Rightarrow \{slack_{h1}, slack_{h2}, ..., slack_{hi}, ..., slack_{hm}\} \geq \sum_{i=1}^{m} y_i \tag{4.25}$$

For inequality 4.25 to be satisfied by all gates in the selected group of m gates, the least slack should be greater than the sum of the low voltage delay and high voltage delay differences of each of the gates.

$$\Rightarrow min\{slack_{h1}, slack_{h2}, ..., slack_{hi}, ..., slack_{hm}\} \geq \sum_{i=1}^{m} y_i \tag{4.26}$$

Hence, we see that a group of gates satisfying the condition

$$\sum_{i=1}^{m} y_i \leq min\{slack_{hi}\}$$

can be put to lower supply voltage without violating the positive timing constraint. Consider the graph in Figure 4.3. We take a one gate falling below the 45^o line and assign it low voltage. Then the gate will have a slack reduced by its $d_l - d_h$. This means when slacks are recalculated, this point will move leftwards by $d_l - d_h$ amount. In order to make a valid low voltage assignment, this point should not go to the left side of the y-axis. Consider a group of gates being assigned low voltage simultaneously. Then each of those gates will have a slack reduced by its $d_l - d_h$. But if two gates fall on the same path, then their slack will be reduced by the sum of their $d_l - d_h$ values. In the worst case all gates falling on the same path can be assigned low voltage simultaneously and all of their slacks can be reduced by the sum of all their $d_l - d_h$ values, i.e., $\sum_{i=1}^{m} y_i$. And now, for this move to be valid, none of the slacks should become negative. That means that $\sum_{i=1}^{m} y_i$ should be less than or equal to the minimum slack of all those selected group of gates.

4.1 Algorithm 2

Step 1: We use the $O(n)$ slack calculation algorithm proposed in [41] to find out the gate slacks for a given circuit.

Step 2: The gates are divided into groups 1, 2 and 3 as described previously.

Step 3: Once this is done, we estimate the dynamic energy savings for group 2 gates and for the gates in group 2 and 3 together for the circuit.

The dynamic energy when all gates in groups 2 and 3 together are assigned a high supply voltage will be proportional to

$$E_{H1} = V_H^2 \times (G + P)$$

and for group 2 will be

$$E_{H2} = V_H^2 \times (G)$$

Similarly, the dynamic energy when all gates in groups 2 and 3 together are assigned a low supply voltage will be proportional to

$$E_{L1} = V_L^2 \times (G + P)$$

and for group 2 will be

$$E_{L2} = V_L^2 \times (G)$$

Then the energy savings for groups 2 and 3 together is estimated as

$$E_{save1} = \frac{E_{H1} - E_{L1}}{E_{H1}} \times 100 = \frac{V_H^2 - V_{L1}^2}{V_H^2} \times (G + P) \times 100 \tag{4.27}$$

and that for group 2 as

$$E_{save2} = \frac{E_{H2} - E_{L2}}{E_{H2}} \times 100 = \frac{V_H^2 - V_{L2}^2}{V_H^2} \times G \times 100 \tag{4.28}$$

Step 4: Now, repeat steps 2 and 3 for each value of V_L within the specified range of voltages

Step 5: Find the low voltages V_{L1} and V_{L2} as the V_L when E_{save1} and E_{save2} are, respectively, maximum.

Table 4.1: Optimal lower supply voltage values (V_L), V_{L1}, V_{L2}, their arithmetic mean and their geometric mean and energy savings estimate when level converters are not allowed for ISCAS'85 benchmark circuits. $V_H = 1.2V$.

Benchmark circuit	Total gates	Dual voltage assignment without level converters											
		V_{L1}			V_{L2}			$(V_{L1}+V_{L2})/2$			$\sqrt{(V_{L1})(V_{L2})}$		
		V_L V	Gates in low voltage	$E_{savg.}$ %	V_L V	Gates in low voltage	$E_{savg.}$ %	V_L V	Gates in low voltage	$E_{savg.}$ %	V_L V	Gates in low voltage	$E_{savg.}$ %
c432	154	0.8	8	2.9	0.89	8	2.34	0.84	8	2.65	0.84	8	2.65
c499	493	0.76	113	13.73	1.11	141	4.13	0.93	123	9.96	0.91	129	11.12
c880	360	0.49	213	49.3	0.71	229	41.34	0.6	229	47.7	0.58	229	48.75
c1355	469	0.77	76	9.53	1.11	108	3.35	0.94	76	6.26	0.92	76	6.68
c1908	584	0.60	221	28.38	1.00	221	11.56	0.80	221	21.93	0.77	221	22.26
c2670	901	0.48	570	53.14	0.82	570	33.73	0.65	570	44.70	0.62	570	46.38
c3540	1270	0.52	149	9.53	0.73	149	7.4	0.62	149	8.60	0.61	149	8.70
c5315	2077	0.49	1220	48.95	0.75	1226	35.96	0.62	1220	43.06	0.60	1220	44.054
c6288	2407	0.55	75	2.46	1.00	77	0.98	0.77	77	1.88	0.73	77	2.015
c7552	2823	0.54	1582	44.69	0.71	2123	48.87	0.62	1672	43.42	0.61	1672	43.42

For ISCAS'85 benchmark circuits in 90nm technology, we have used 1.2 V for V_H, and a range of 0.1V to 1.2V in steps of 0.01V for V_L values.

It is observed that the results obtained depend on the circuit topology. Circuits, like c880, which have fewer long paths can be optimized to obtain a considerably good energy savings. And circuits, like c1355, which have a large number of paths with longest path delays comparable to the critical path delay of the circuit are difficult to optimize . Empirical results, Tables 4.1 and 4.2, show that when the lower supply voltage, V_L, is taken as V_{L1} we get maximum energy savings for all ISCAS'85 benchmark circuits for voltage assignment, when level converters are not allowed, see Chapter 5. And when level converters are allowed during voltage assignment (see Chapter 7), then the geometric mean of V_{L1} and V_{L2} has to be taken to get the maximum savings.

$$V_L = V_{L1},\ \textit{when level converters are not allowed for voltage assignment} \tag{4.29}$$

Table 4.2: Optimal lower supply voltage values (V_L), V_{L1}, V_{L2}, their arithmetic mean and their geometric mean and energy savings estimate when level converters are allowed for ISCAS'85 benchmark circuits. $V_H = 1.2V$.

Benchmark circuit	Total gates	Dual voltage assignment using level converters											
		V_{L1}			V_{L2}			$(V_{L1}+V_{L2})/2$			$\sqrt{(V_{L1})(V_{L2})}$		
		V_L V	Gates in low voltage	$E_{savg.}$ %	V_L V	Gates in low voltage	$E_{savg.}$ %	V_L V	Gates in low voltage	$E_{savg.}$ %	V_L V	Gates in low voltage	$E_{savg.}$ %
c432	154	0.8	73	17.07	0.89	85	24.84	0.84	81	26.82	0.84	81	26.82
c499	493	0.76	173	21.05	1.11	359	10.52	0.93	249	20.17	0.91	247	21.30
c880	360	0.49	223	51.62	0.71	309	55.79	0.6	290	60.417	0.58	286	60.89
c1355	469	0.77	122	15.30	1.11	260	8.00	0.94	197	16.30	0.92	193	16.96
c1908	584	0.6	263	33.78	1.00	267	24.44	0.80	395	37.58	0.77	385	38.78
c2670	901	0.48	376	35.06	0.82	784	46.38	0.65	677	53.09	0.62	633	51.50
c3540	1270	0.52	647	41.38	0.73	1073	53.22	0.62	906	52.29	0.61	881	51.45
c5315	2077	0.49	1140	45.74	0.75	1777	52.14	0.62	1633	57.64	0.60	1602	57.85
c6288	2407	0.55	659	21.63	1.00	1877	23.83	0.77	1302	31.82	0.73	1189	47.34
c7552	2823	0.54	1560	44.07	0.71	2235	51.47	0.62	1998	51.88	0.61	1197	51.78

$$V_L = \sqrt{(V_{L1})(V_{L2})},\ \textit{when level converters are allowed for voltage assignment} \quad (4.30)$$

Thus, we obtain the value of the lower supply voltage giving a maximum energy savings. The gate slacks are compared to the $d_l - d_h$ values for each gate and for each value of V_L available. And then the gates are divided into respective groups based on their slacks. Then the energy savings is calculated for each value of V_L available and a minimum value is found; the corresponding V_L is the required lower voltage value. Since the number of voltages available for V_L is negligible when compared to the number of gates in the circuit, this algorithm is proportional to the number of gates in the circuit. Thus, its complexity is linear, i.e. $O(n)$, where n is the number of gates.

We define the estimated energy savings for a given circuit as

$$E_{savg.} = \frac{V_H^2 - V_L^2}{V_H^2} \times N \times 100 \quad (4.31)$$

Table 4.3: Optimal lower supply voltage values (V_L) and energy savings estimate using Algorithms 2 when level converters are not allowed for ISCAS'85 benchmark circuits. $V_H = 1.2V$.

Benchmark circuit	Total gates	Dual voltage assignment without level converters						
		Algorithm 2			V_L=0.84V		V_L=0.6V	
		V_L V	Gates in low voltage	$E_{savg.}$ %	Gates in low voltage	$E_{savg.}$ %	Gates in low voltage	$E_{savg.}$ %
c432	154	0.8	8	2.9	8	2.65	8	3.9
c499	493	0.76	113	13.73	121	12.52	56	8.52
c880	360	0.49	213	49.3	229	32.44	229	47.71
c1355	469	0.77	76	9.53	76	8.27	64	10.24
c1908	584	0.60	221	28.38	221	19.3	221	28.4
c2670	901	0.48	570	53.14	570	32.27	570	47.45
c3540	1270	0.52	149	9.53	149	5.98	149	8.8
c5315	2077	0.7	1220	48.95	1240	30.45	1220	44.054
c6288	2407	0.55	75	2.46	77	1.63	75	2.34
c7552	2823	0.54	1582	44.69	2359	42.62	1672	43.923

where N is the total number of gates to be put in low voltage once the voltage assignment is done.

$E_{savg.}$ for the ISCAS'85 benchmark circuits is calculated and reported in Tables 4.3 and 4.4. Also, we compare these with the results obtained when $V_L = 0.7 \times V_H$ and when $V_L = 0.5 \times V_H$ as done in popular literature. It is observed that the expected energy savings in both cases is large when we use V_L given by Algorithm 2.

Figures 4.8 and 4.10 show the variation of energy savings estimated when level converters are not allowed (see Chapter 5 for details) and Figures 4.7 and 4.9 when level converters are allowed (see Chapter 7), with various V_L values for circuits c880 and c1355, respectively using the following equation for energy of the dual voltage design, E_{dual},

$$E_{dual} = \sum_{i=1}^{p} \alpha_i \times C_i \times V_H^2 + \sum_{i=1}^{q} \alpha_i \times C_i \times V_L^2 \tag{4.32}$$

Table 4.4: Optimal lower supply voltage values (V_L) and energy savings estimate using Algorithms 2 when level converters are allowed for ISCAS'85 benchmark circuits. $V_H = 1.2V$.

Benchmark circuit	Total gates	Dual voltage assignment using level converters						
		Algorithm 2			V_L=0.84V		V_L=0.6V	
		V_L V	Gates in low voltage	$E_{savg.}$ %	Gates in low voltage	$E_{savg.}$ %	Gates in low voltage	$E_{savg.}$ %
c432	154	0.84	81	26.82	81	26.82	43	20.94
c499	493	0.91	247	21.30	211	21.18	99	15.06
c880	360	0.58	286	60.89	323	45.76	290	60.42
c1355	469	0.92	193	16.96	154	16.75	44	7.04
c1908	584	0.77	385	38.78	415	36.24	263	33.78
c2670	901	0.62	633	51.50	813	46.02	606	50.45
c3540	1270	0.61	881	51.45	1093	43.89	864	51.02
c5315	2077	0.60	1602	57.85	1812	44.49	1602	57.85
c6288	2407	0.73	1189	47.34	1470	31.15	780	24.31
c7552	2823	0.61	1971	51.78	2347	42.40	1943	51.62

where α_i is the average activity of each node, C_i is the node capacitance, p is the number of gates in high voltage and q is the number of gates in low voltage. The activity factors and node capacitances for each gate subjected to fanouts of 1 to 4 are obtained from SPICE simulations and tabulated.

We set the V_L to a specific value between 0.1V to 1.2V and find the energy savings in each case for c880 and c1355. From the graphs, the V_L value at the minimum energy point, when level converters are not allowed (see Chapter 5), is found to be approximately 0.5V for c880 and 0.7V for c1355 circuits. We observe that these V_L values are very close to the V_L voltages obtained using Algorithm 2 as reported in Table 4.3, which are 0.49V and 0.77V respectively. Similarly, the V_L value corresponding to the minimum energy point found, when level converters are allowed (see Chapter 7), is approximately 0.6V for c880 and 0.9V for c1355 circuits. Again these values are close to the V_L values obtained using Algorithm 2 as reported in Table 4.4, which are 0.58V and 0.91V respectively.

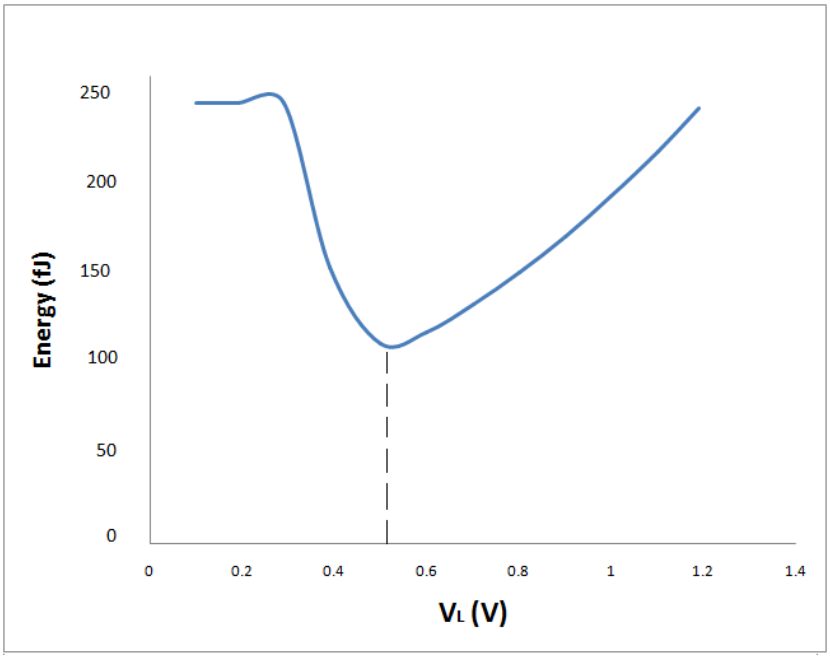

Figure 4.7: Energy versus voltage for c880 when level converters are not allowed.

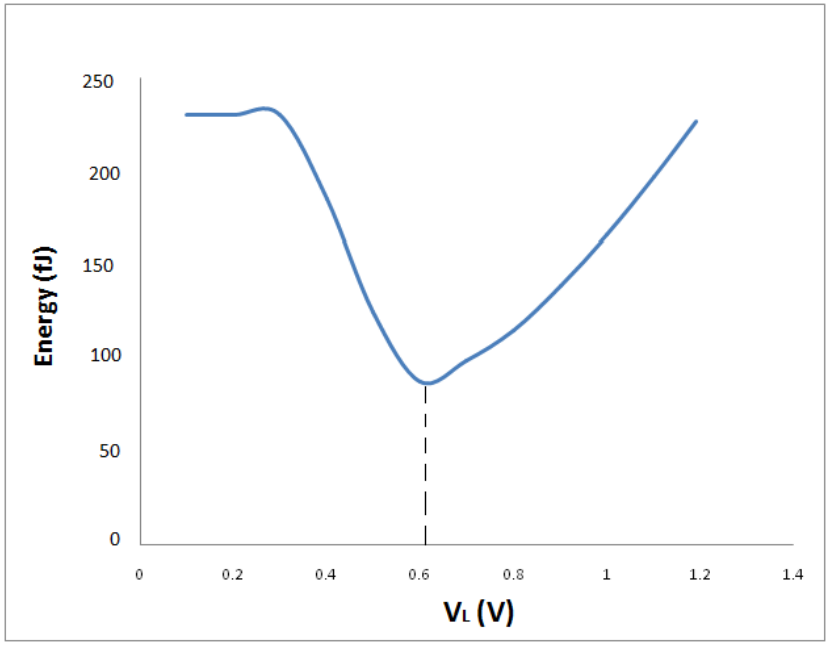

Figure 4.8: Energy versus voltage for c880 when level converters are allowed.

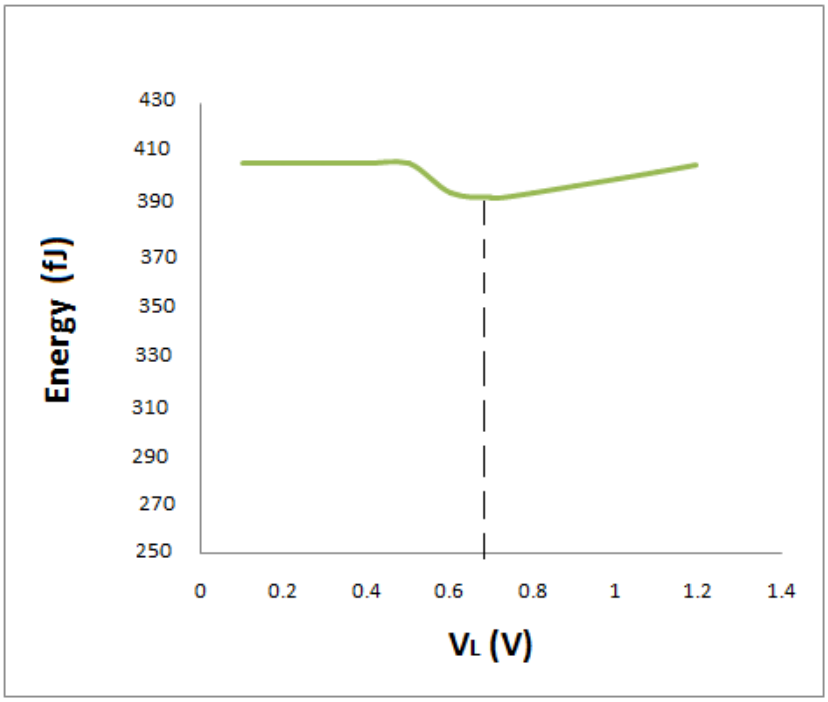

Figure 4.9: Energy versus voltage for c1355 when level converters are not allowed.

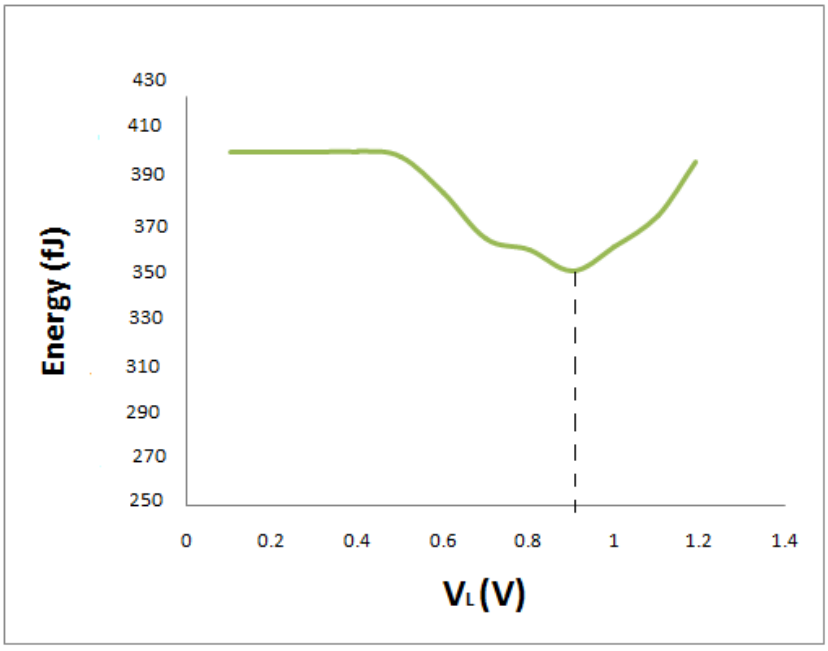

Figure 4.10: Energy versus voltage for c1355 when level converters are allowed.

Chapter 5

Slack-Based Algorithm For Dual Voltage Assignment Without Using Level Converters

5.1 Introduction

The use of multiple supply voltages to reduce energy consumption is a very commonly used technique in CMOS circuits. Generally, the gates in the critical path are kept at high supply voltage and the gates in non-critical paths are put to lower supply voltage, thus avoiding timing violation.

Usami and Horowitz [92] describe the clustered voltage scaling (CVS) technique in which the cells driven by each power supply are grouped together and use of level converter is avoided by not allowing a V_L gate to feed a V_H gate. The extended clustered voltage scaling (ECVS) proposed in [94] removes this restriction by allowing the usage of a level converter [52] at such boundaries. This is referred to as asynchronous level conversion. Both CVS and ECVS aim at utilizing the surplus timing of the non-critical paths in a circuit by applying a lower supply voltage on gates that are on these paths. This results in reduced dynamic power dissipation and hence lowers system level power dissipation. In this chapter, we do not use level converters, due to the overheads associated with them [5, 17, 41, 50].

Several other algorithms have been proposed in literature for dual/multiple-voltage assignments modifying CVS and ECVS algorithms. A greedy algorithm (GECVS) is used in [50] to group gates for V_L assignment based on a sensitivity measure derived from the gate slacks, change in total power and delay. The authors claim to improve power savings up to 28% and 13% with respect to CVS and ECVS, respectively. Authors in [88] describe three algorithms for dual voltage designs. Algorithm PROUD is essentially a linear programming model to minimize he power consumption. PRHEUDENT is a heuristically approach to reduce the computation time. The third algorithm numerically ceils a non-integral delay

to the nearest integer and uses PROUD for power minimization. All three of these algorithms allow the use of level converters. Also, a technique that optimizes the gate sizing, threshold voltage and supply voltage simultaneously using linear programming technique is discussed in [16]. Our algorithms take considerably less time when compared to these algorithms.

In [42], the authors use a mixed integer linear programming (MILP) technique to find the V_L value for circuits operating in the subthreshold region. Then an ECVS kind of method is used with multiple logic gates between the V_L and V_H boundaries instead of level converters.

5.2 An Example of a Chain of Inverters

Consider the chain of inverters shown in Figure 5.1. We simulated this circuit using Synopsys HSPICE program [1], with voltages V_1 and V_2 as 0.4V, 0.6V, 0.8V, 1.0V and 1.2V. A 1GHz 50% duty-cycle clock is applied at the input and a capacitance of 6fF, equivalent to four inverters, was used as the load at the output. The results for 90nm technology are presented in Figure 5.2. It reports the total energy consumption and delay for the circuit at various values of V_1 and V_2. The energy values shown in the green squares are when V_1 and V_2 are equal, corresponding to a single voltage operation. The values reported in blue squares below the $V_1 = V_2$ diagonal are the values when V_1 is greater than V_2, i.e., when a high voltage gate is feeding a low voltage gate. The squares above this diagonal represent the operation when V_2 is greater than V_1, i.e., when a low voltage gate feeds a high voltage gate. We observe that the delay measurement in two of the top cells fails, represented as infinite delay, when the voltage difference is large. For all cases above the diagonal, although logic 1 level matched V_{DD}, logic 0 levels for the five inverters near the output were higher than ground. That produced significantly higher leakage. This indicates the necessity for level conversion at the voltage boundary. The use of level converters has been studied in the literature [5, 17]. However, the design of such devices is still evolving and as problems with their performance have been reported. Especially, their performance in terms of power and

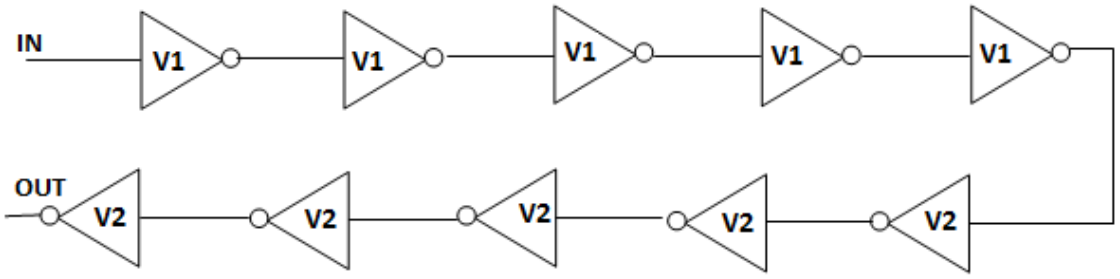

Figure 5.1: A chain of ten inverters.

V2(V) \ V1(V)	0.4	0.6	0.8	1.0	1.2
1.2	6.344fJ ∞	31.23fJ 282.9ps	12.31fJ 123.3ps	7.008fJ 95.66ps	7.863fJ 84.11ps
1.0	4.279fJ ∞	7.861fJ 203.7ps	4.442fJ 123.3ps	5.199fJ 99.46ps	6.606fJ 91.05ps
0.8	4.749fJ 1179ps	2.568fJ 203.1ps	3.233fJ 132.1ps	4.283fJ 115ps	5.656fJ 107.6ps
0.6	1.261fJ 796.1ps	1.757fJ 234ps	2.532fJ 179.3ps	3.566fJ 164.4ps	4.932fJ 155.8ps
0.4	0.753fJ 1065ps	1.280fJ 614.1ps	2.036fJ 565ps	3.074fJ 561ps	4.443fJ 557.6ps

Figure 5.2: Energy and delay measurements at various values of V_1 and V_2 for the inverter chain of Figure 5.1.

delay overheads deteriorates as the difference between the two voltages increases, further limiting the capability for power reduction.

For all cases where a high voltage gate feeds a low voltage gate, energy savings are seen. These results demonstrate the effectiveness of using topological constraints in dual-V_{DD} designs. In the following sections we describe a new algorithm that assigns low voltage to the gates of a given circuit such that no low voltage gate drives a high voltage gate.

5.3 Algorithm 3

Step 1: Initially we assume that all gates are at high voltage, i.e., are connected to V_H supply voltage.

Step 2: Once we obtain the V_L value as described in the previous section, we assign the gates in group 2 the lower supply voltage.

Step 3: Then we recalculate the slack using Algorithm 1. Theorem 4.2 mandates that no negative slack occurs during this voltage assignment.

Step 4: The gates are again divided into new groups 1, 2 and 3 once the slack is calculated.

Step 5: The circuit is levelized and, starting from the primary outputs, we take a small group of high-voltage gates out of group 3 satisfying the condition stated in Theorem 4.3 and assign them the low voltage and recalculate the slack.

Step 6: Once we assign low voltage to is group of gates, we check whether there are any low voltage gates driving high voltage gates anywhere in the circuit.

Step 7: If this occurs, the supply of that low voltage gate is changed back to the high voltage.

Step 8: Gate slacks are calculated again.

Step 9: The gates are redivided into groups 1, 2 and 3.

Step 10: Steps 5 to 9 repeated on the remaining gates in a reverse levelized manner until we reach the primary inputs.

Table 5.1: Optimal lower supply voltage values (V_L) and energy savings using Algorithms 2 and 3 for ISCAS'85 benchmark circuits, $V_H = 1.2V$.

Benchmark circuit	Algorithm 2 V_L V	Total gates	Algorithm 3 Gates in low voltage	Algorithm 3 $E_{savg.-expc.}$ %	SPICE Results $E_{single\ VDD}$ fJ	SPICE Results $E_{dualVDD}$ fJ	SPICE Results $E_{savg.-obs.}$ %	CPU time to run Algorithms 2 and 3 s*
c432	0.8	154	8	2.9	161.3	155.4	3.66	1.78
c499	0.76	493	113	13.73	463	427	7.8	9.41
c880	0.49	360	213	49.3	277.6	115.8	58.29	5.39
c1355	0.77	469	76	9.53	455.2	433.1	4.86	8.75
c1908	0.60	584	221	28.38	496.5	378.3	23.81	11.43
c2670	0.48	901	570	53.14	660.3	251.5	61.9	23.49
c3540	0.52	1270	149	9.53	1843	1620	12.23	45.44
c5315	0.49	2077	1220	48.95	2320	1272	45.17	109.47
c6288	0.55	2407	75	2.46	1932	1869	3.26	154.94
c7552	0.54	2823	1582	44.69	2465	1562	36.63	191.04

* Intel Core i5 2.30GHz, 4GB RAM

Table 5.2: Optimal lower supply voltage values (V_L) and energy savings using Algorithms 2 and 3 for ISCAS'85 benchmark circuits, when T_C is increased by 5%, $V_H = 1.2V$.

Benchmark circuit	Algorithm 2 V_L V	Total gates	Algorithm 3 Gates in low voltage	Algorithm 3 $E_{savg.-expc.}$ %	SPICE Results $E_{single\ VDD}$ fJ	SPICE Results $E_{dualVDD}$ fJ	SPICE Results $E_{savg.-obs.}$ %	CPU time to run Algorithms 2 and 3 s*
c432	1.08	154	154	19.00	161.3	123.9	23.19	1.70
c499	1.03	493	493	26.33	463	321.9	30.48	9.18
c880	0.67	360	344	65.77	277.6	83.86	69.79	4.32
c1355	1.06	469	469	21.97	455.2	339.9	12.15	8.52
c1908	1.00	584	584	30.56	496.5	445	10.37	8.56
c2670	0.81	901	899	54.32	660.3	257.3	61.03	15.81
c3540	0.90	1270	1270	43.75	1843	949.5	48.48	28.22
c5315	0.72	2077	2077	64.00	2320	716.8	69.11	61.77
c6288	1.07	2407	2407	20.49	1932	1464	24.22	108.39
c7552	0.68	2823	2816	67.72	2465	677.2	72.28	175.07

* Intel Core i5 2.30GHz, 4GB RAM

5.4 Results

We ran our algorithm on ISCAS'85 benchmark circuits. These benchmark circuits are synthesized using a small set of 90nm standard cells consisting of an inverter, INV, a two-input NAND gate, NAND2, a three-input NAND gate, NAND3, and a two-input NOR gate NOR2. Each circuit is simulated with a logic simulator with randomly generated input vectors to determine the circuit's average activity. The gate delays, average activity of each node and node capacitances of each circuit are obtained from SPICE simulations done for supply voltages ranging from 0.1V to 1.2V in 0.01V steps. These values are also tabulated for each standard cell with fanout of one to four. For all SPICE simulations, 90nm bulk CMOS predictive technology models are used at room temperature and the higher supply voltage used is $V_H = 1.2$V. A hundred random input vectors are used for simulations and energy per vector is found. In the Table 5.1, $E_{single\ VDD}$ and $E_{dualVDD}$ are the SPICE results for energy per vector consumed in single-voltage design and that in dual-voltage design, respectively. Also, $E_{savg.-expc.}$ is the energy savings estimated by the equation 4.31 using the lower supply voltage value given by equation 4.29. The actual energy savings reported by SPICE for dual-voltage design is $E_{savg.-obs.}$.

Algorithms 2 and 3 are used on each of the ISCAS'85 benchmark circuits to obtain the lower supply voltage, V_L, the number of gates which can be assigned low voltage using our algorithms, the final energy savings and the CPU time to run the algorithms. Results for each of the ISCAS'85 circuits are tabulated in Table 5.1.

Energy for the single supply design, E_{single}, with V_H supply voltage and for dual voltage design, $E_{savg.-obs.}$ are obtained from SPICE simulations. The results are also tabulated in Table 5.1. The actual energy savings reported by $E_{savg.-obs.}$ is observed to be very close to the estimated values. Also, the CPU time is very low.

Figures 5.3, 5.4, 5.7 and 5.8 show the "Delay increment versus slack" graphs for the initial slacks and final slacks for the circuits c880 and c499, respectively. The brown markers indicate gates in high voltage and blue markers indicate gates in low voltage. It is seen that

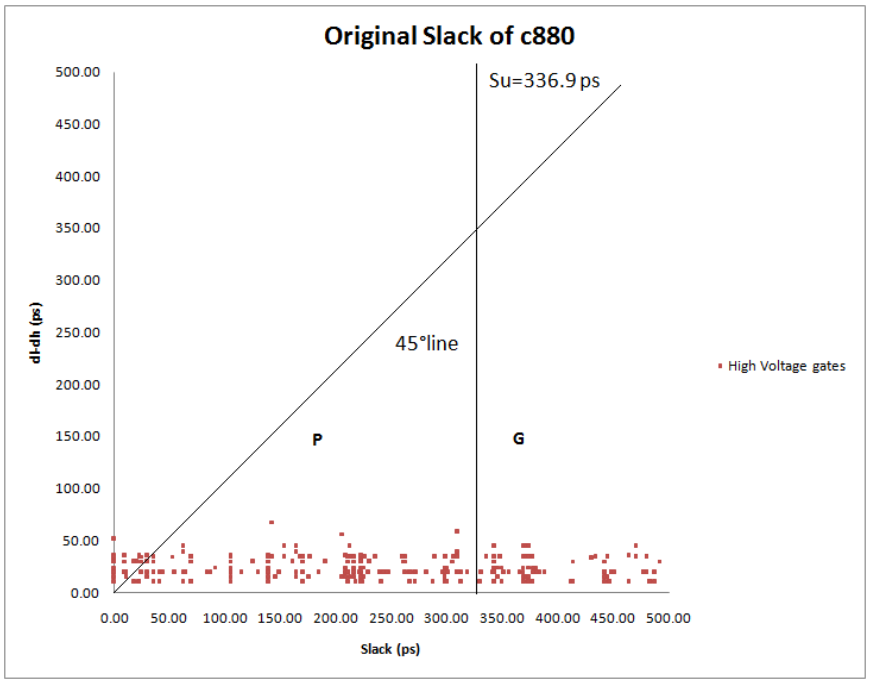

Figure 5.3: Delay increment versus initial slack for gates of c880 circuit, $V_H = 1.2V$, $V_L = 0.49V$.

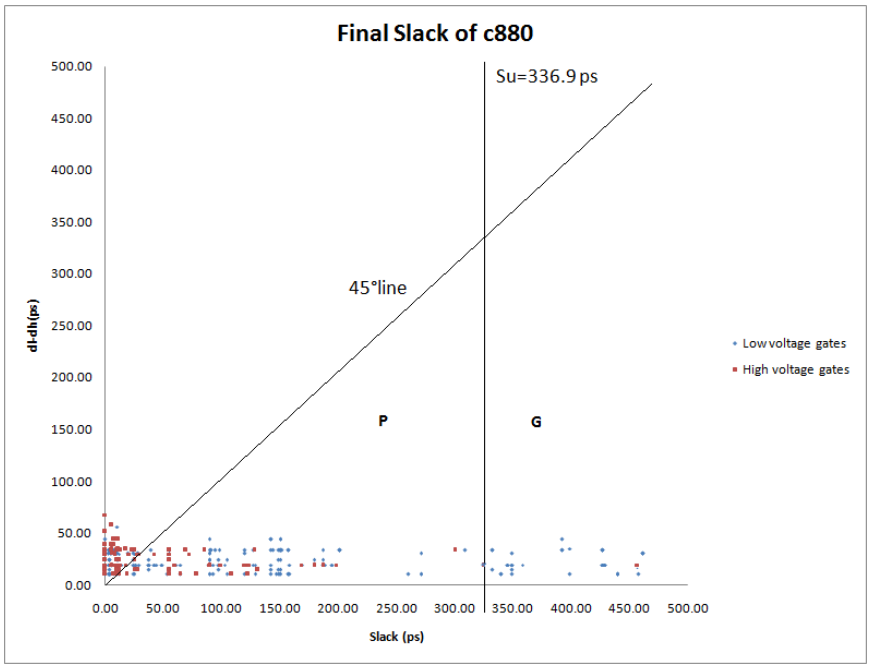

Figure 5.4: Delay increment versus final slack for gates of c880 circuit, $V_H = 1.2V$, $V_L = 0.49V$.

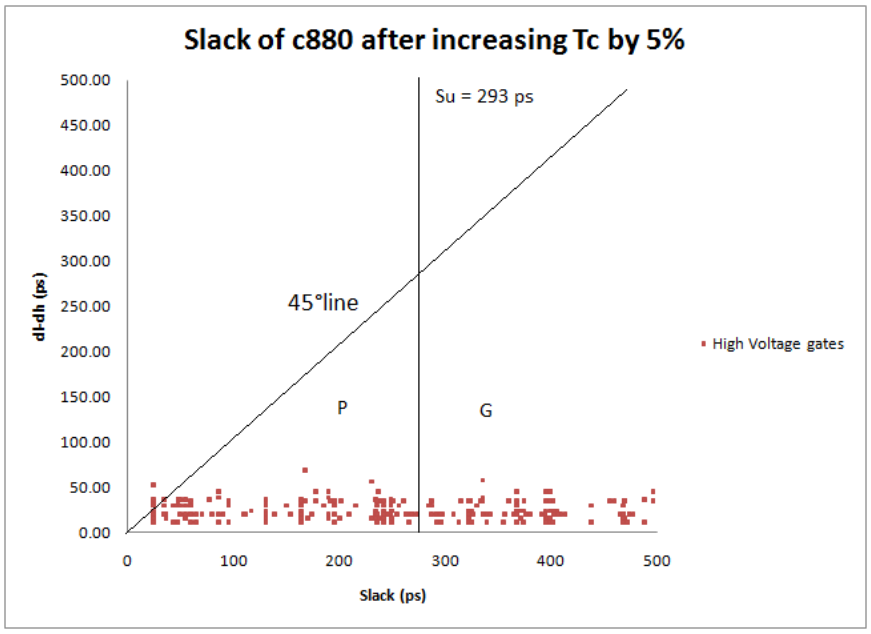

Figure 5.5: Delay increment versus initial slack for gates of c880 circuit when Tc is increased by 5%, $V_H = 1.2V,\ V_L = 0.67V$.

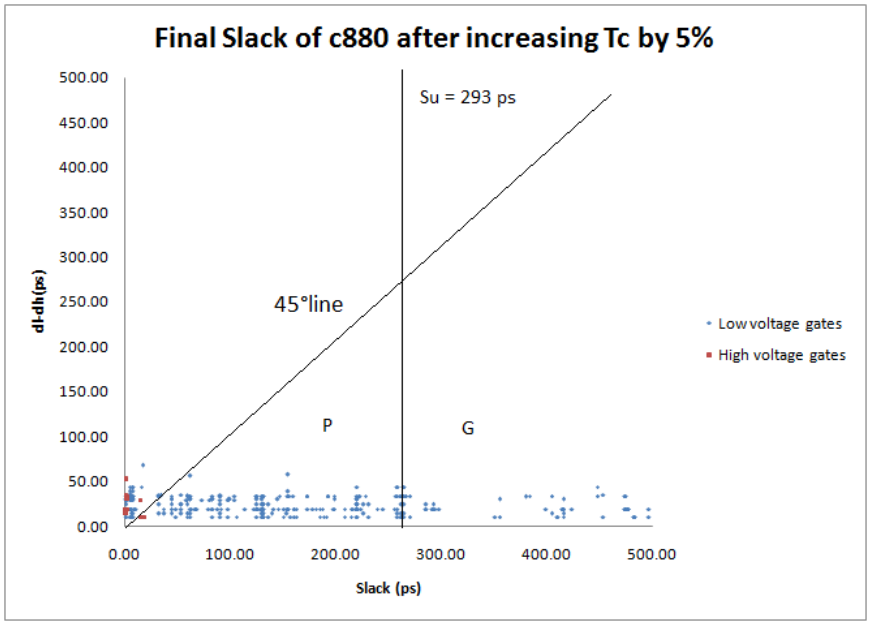

Figure 5.6: Delay increment versus final slack for gates of c880 circuit when Tc is increased by 5%, $V_H = 1.2V,\ V_L = 0.67$.

initially all gates are in low voltage. Then once Algorithms 2 and 3 are used and slacks are recalculated using Algorithm 1, all gate slacks are reduced. All the high voltage gates tend

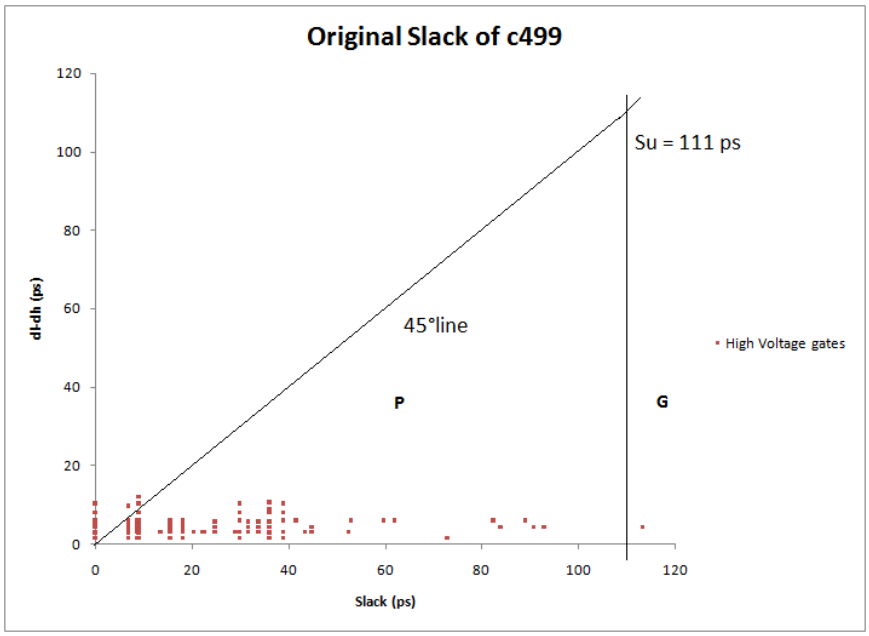

Figure 5.7: Delay increment versus initial slack for gates of c499 circuit, $V_H = 1.2V$, $V_L = 0.91V$.

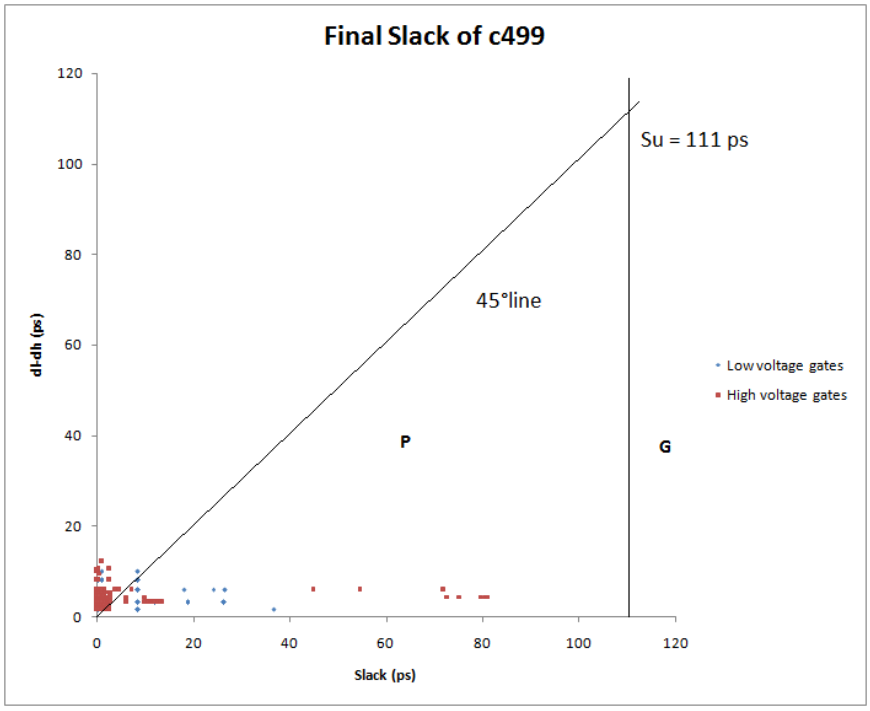

Figure 5.8: Delay increment versus final slack for gates of c499 circuit, $V_H = 1.2V$, $V_L = 0.91V$.

to concentrate at lower slack values and also, many gates have moved above the 45^o line. The low voltage gates still below the 45^o line are gates with very large slacks and the high voltage gates still lying below the line indicate the gates which cannot be put in low voltage due to topological constraints imposed by Algorithm 3.

The results for ISCAS'85 benchmark circuits when their critical timings are increased by 5% are tabulated in Table 5.2. Figures 5.5 and 5.6 show the "Delay increment versus slack" graphs for the initial slacks and final slacks for the circuits c880 when its critical timing is allowed to increase by 5%. From the graphs it can be seen that the slacks of the gates have moved towards the right due to increased critical path delay, which in turn increases the gate slacks. Also, the final number of gates in high voltage is less, which can be seen by the reduced density of brown dots.

In many cases, it is easy to assume that the group of gates G with slacks greater than S_u, can always be assigned the lower supply voltage in a dual voltage design such that a low voltage gate never feeds into a high voltage gate.

Proof of Theorem 4.2 indicates that the gates with slacks greater S_u, represented by G, might consist of full paths from primary inputs to the primary outputs that can be assigned a lower supply without violating timing constraints. This means that all gates in this group fall on a set of paths which have all low voltage gates. However, a possibility of a low voltage gate feeding a high voltage gate occurs when any of these paths branch into another path which has a high voltage gate.

Consider the example shown in Figure 5.9. PI represents the primary input and PO1 and PO2 represent primary outputs. The various colored rectangles are logic gates. Suppose that the slack of the yellow gate is greater than S_u. Then, all the gates in the longest path through this gate will have the same slack, from Chapter 3. Thus, all these gates can be assigned low voltage without timing violations, according to Theorem 4.2. Now since the path from PI to PO2 consists of five gates, it is okay to assume that this is the longest path through the yellow gate. Thus, the blue gates and the yellow gate are assigned low voltage.

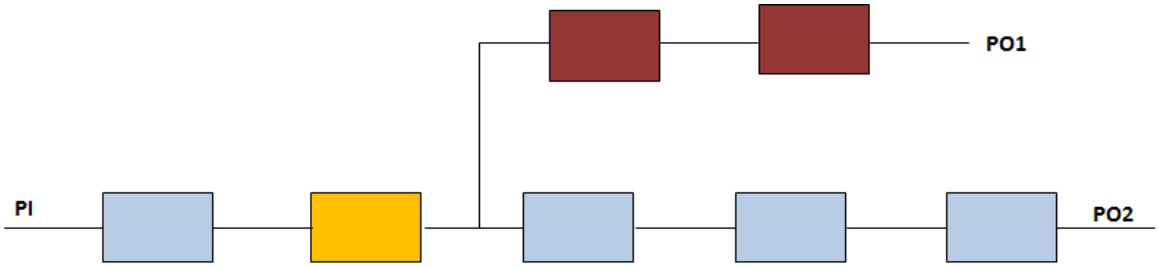

Figure 5.9: An illustration conforming to the topological constraints on gates with slacks greater than S_u.

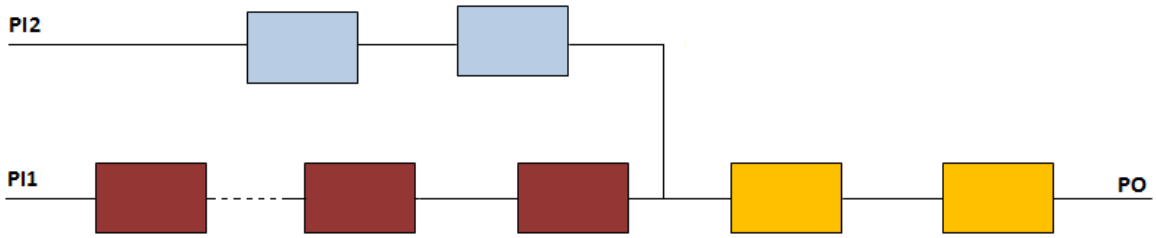

Figure 5.10: An illustration violating the topological constraints on gates with slacks greater than S_u.

Logically, this makes the path from PI to PO1, smaller than the path from PI to PO2. Since the slack is the difference of critical path delay and the longest path delay through the gate, the path delays of the brown gates will be less than the path delay of the rest of the gates. Hence, their slacks will be greater than the slack of the yellow gate. But the slack of the yellow gate is greater than S_u. Thus, the slack of the brown gates is also greater than S_u. From Theorem 4.2, then the brown gates should be low voltage gates.

Alternately, if the path from PI to PO1 is in low voltage, then this path should be the longest path through the yellow gate because low voltage gates have higher delay than high voltage gates. This makes the slacks of the gates falling on the other path greater than S_u, as they fall on a shorter path.

Next, consider the group of gates shown in Figure 5.10. PI1 and PI2 are the primary inputs and PO is the primary output. The colored rectangles are logic gates and the dots in between the two brown gates on the path from PI1 and PO represent a large number of gates in between. Assume that the path from PI1 to PO is a critical or near-critical path. Then these gates have small slacks and none of them can be assigned low voltage. Now

consider the gates on the path from PI2 and PO.This is a small path from primary inputs to the primary outputs and the longest path through the blue gates goes through the two yellow gates which lie on some other longer path. For the two blue gates, the slack will be the difference between the critical path delay and the longest path delay through these gates which includes the delays of the yellow gates. This slack can be larger than S_u as this is a short path. Hence, we can put these gates in low voltage. The yellow gates cannot be put in low voltage because the longest path through these gates is from PI1 and not from PI2. Hence, their slacks are not greater than S_u. This causes a low voltage gate driving a high voltage gate at the interface between blue and yellow gates. Thus, causing a violation of topological constraint. This might be a rare occurrence, but we cannot assume that all gates which have slacks greater than S_u do no violate the topological constraint. These gates are put back to high voltage by Algorithm 3 when it checks for the topological constraints over entire circuit.

It is worth noting that if the gates with slacks greater than S_u are not assigned V_L at the same time, we can assign V_L starting from the primary outputs in a reverse levelized manner. This will prevent us from assigning low voltage to the two blue gates in Figure 5.10. Also, the brown gates in Figure 5.9 will be assigned low voltage before the yellow gate.

In this chapter, we have described an algorithm to assign low voltage to the gates which puts restrictions on the circuit structure. This restriction can be lifted by using a level converter at the interface where a low voltage gate has a high voltage gate at its fan-outs. The level converter designs is described in Chapter 6 and the algorithm to assign low voltage to gates which allows the use of level converters is described in Chapter 7.

Chapter 6

Asynchronous Level Converters in Dual-Voltage Design

In a dual voltage design level converters are required whenever a low voltage gate feeds a high voltage gate. In this chapter, we study various level converter designs available in the literature and determine the design to be used in this work. The specific details of the designs are beyond the scope of this work. Interested readers can refer to [5, 17, 23, 31, 49, 51, 53, 89, 97].

Figures 6.6, 6.7, 6.8, 6.9, 6.10, and 6.11 show various designs of level converters proposed in the literature. The most commonly used design is the Dual Cascode Voltage Switch [75, 96], Figure 6.6. It is also called a conventional level converter (shifter) in many references. Another widely used level converter is the passgate level converter [31, 50, 51], Figure 6.7. 'Conventional Type II' level converter, Figure 6.8, is described in [49, 97]. The conventional single supply voltage level shifter, Figure 6.9, has been referred to in [5, 31, 53]. The contention mitigated level converter, Figure 6.11, was proposed in [89]. Also, a recent design using single supply voltage [5], Figure 6.10, is considered.

For each level converter, energy consumption per cycle, average power dissipation, rise delays and fall delays have been recorded in Tables 6.1, 6.2, 6.3, 6.4, 6.5, and 6.6 for different designs at various low voltage values. The simulations are done in HSPICE [1]. For the simulations, the level converter is loaded with a capacitor of 6fF (Figure 6.1), which is equivalent to the nodal capacitance of four inverters (Figure 6.2), at the output. The transistor sizes are optimized for lowest energy using HSPICE's internal optimizing function. A clock of 1.5ns period is applied at the input of the level converter with 0V for logic 0 and VL for logic 1. VH is 1.2V in all cases. The simulation is done for one hundred cycles.

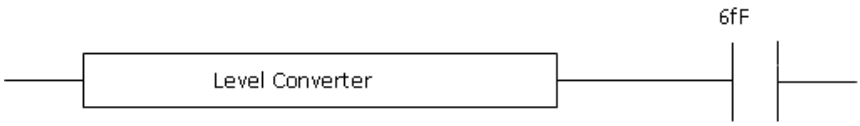

Figure 6.1: Experimental setup for level converter.

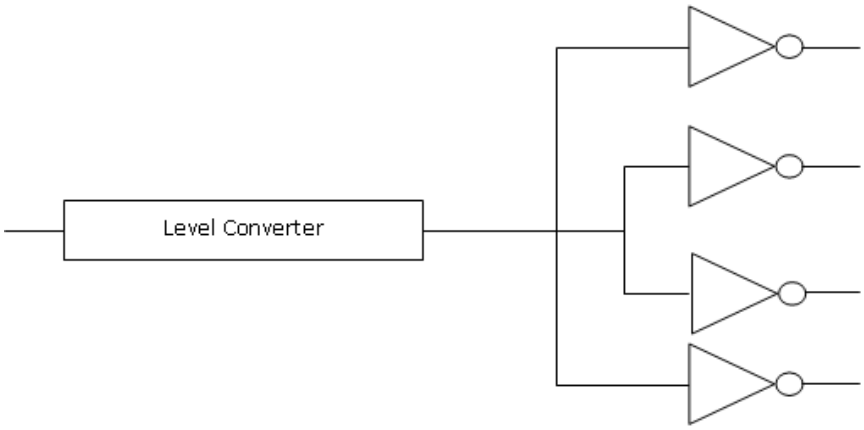

Figure 6.2: Equivalent setup for level converter experiments.

In Tables 6.1, 6.2, 6.3, 6.4, 6.5, and 6.6, the energy consumption per cycle and average power of the level converter over the simulation period is decreasing up to a certain voltage and then starting to increase as the low voltage gets closer to the higher voltage. The difference between the rise delay and fall delay is decreasing as the low voltage is increased. Delay measurements failed for some V_L values for the chosen clock period. The cells that are supposed to report delays are left blank in these cases.

We observe that the energy consumption per cycle, Figure 6.3, of the contention mitigated level converter is the least of all level converters. The delay of the contention mitigated level converter is not the least, but lies in the middle of the range of the delays of all level converters. Since we are concerned about minimum energy, we choose contention mitigated level converter for our designs.

A level converter can be used in two different ways between a low voltage gate and a high voltage gate. We can use a level converter at each fan-in of the high voltage gate that comes from a low voltage gate, as shown in Figure 6.4. Another way is to put a level

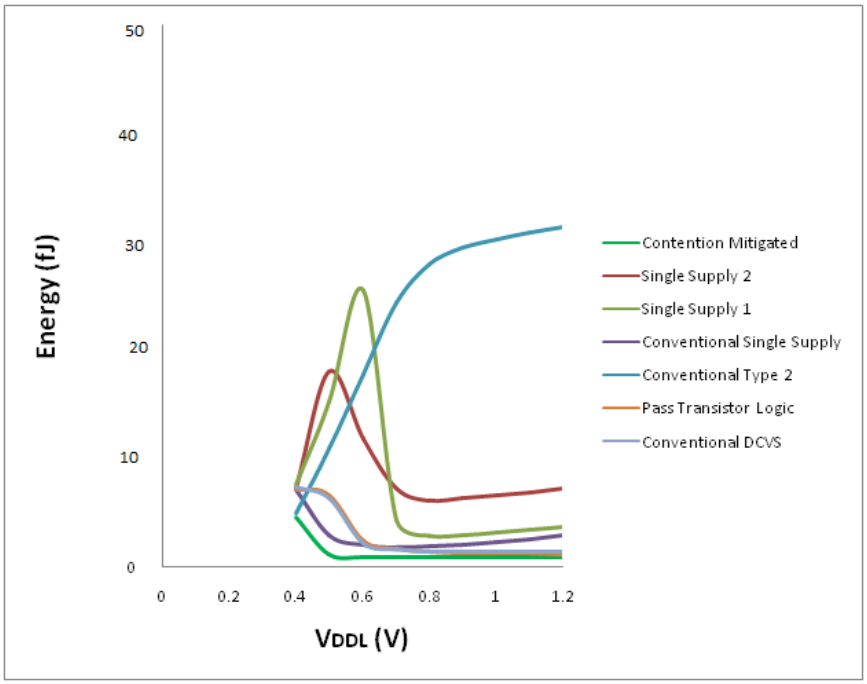

Figure 6.3: Energy per cycle versus V_L. $V_H = 1.2$V.

converter at the output of each low voltage gate that feeds into at least one high voltage gate. This is shown in Figure 6.5. Authors in [50] claim that the level converters should be directly at the inputs of the high voltage fanout gates, rather than at the output of the low voltage driving gate, in order to reduce the dynamic power consumed in switching the wire. This is particularly important because a substantial portion of the total capacitance on a chip is due to wiring and not devices [50, 65]. Hence, we use the level converters at the fan-in of each high voltage gate that is fed by a low voltage gate.

The authors in [19] modify the threshold voltage of the pMOS transistors in the high voltage gates that are driven by low voltage gates in order to obtain the level shifting operation together with the logic operation, as seen in Figure 6.12. The authors claim an average of 20% energy savings, when compared to the single voltage design with the higher supply voltage, for ISCAS85 benchmark circuits using 180-nm technology and 17% with 70-nm. It is shown that as the threshold voltage of the high voltage gate driven by low voltage gate is increased, The dynamic energy of the high voltage gate reduces. For example, for

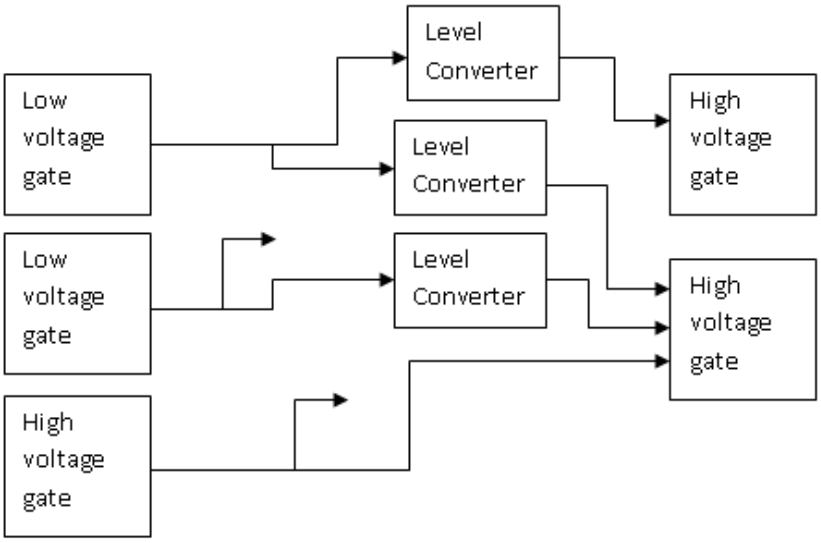

Figure 6.4: Dual voltage with level converter inserted before each fan-in of a high-voltage gate.

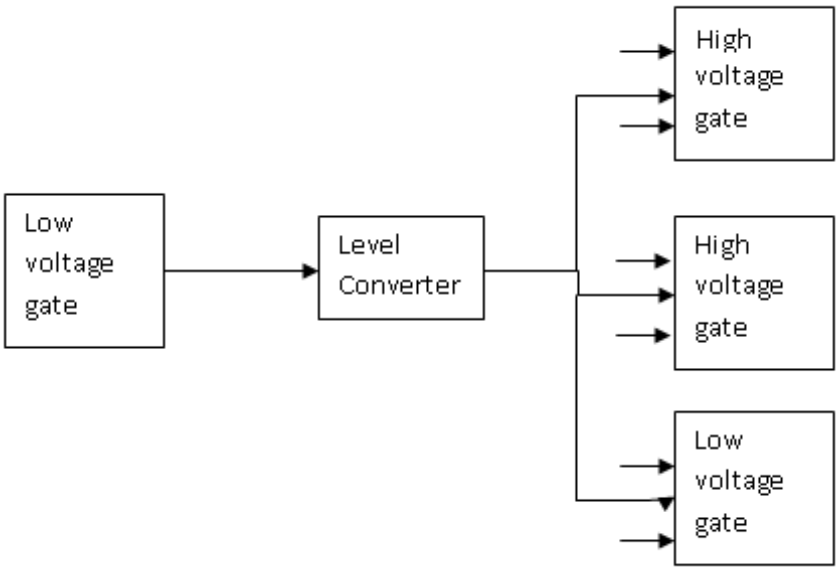

Figure 6.5: Dual voltage with level converter inserted after the output of a low-voltage gate.

70nm technology NOT gate operating at a higher supply voltage of 1V, the energy consumption is 1.33fJ. A level-shifting NOT gate consumes 1.43fJ of dynamic energy at $V_{ddh} = 1V$, $V_{ddl} = 0.8V$ and $V_{thp} = 0.2V$. When the threshold voltage of the pMOS transistor is increased to 0.4V, the energy consumption reduces to 1.3fJ. But we have to trade this reduction

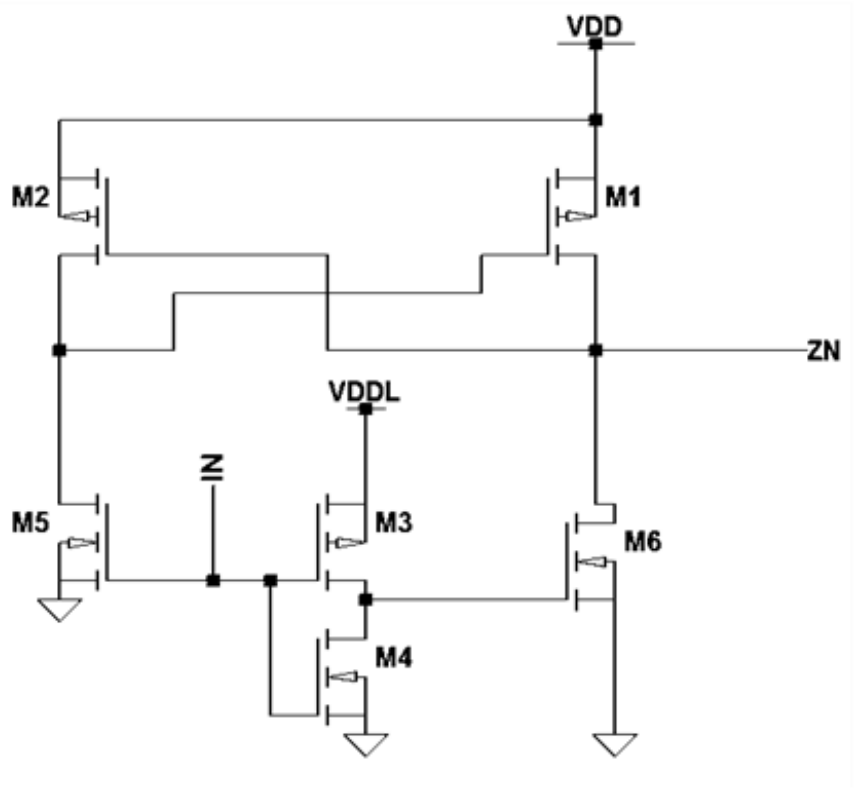

Figure 6.6: Dual Cascode Voltage Switch (DCVS) level converter [75, 96].

Table 6.1: DCVS level converter, $V_H = 1.2V$.

V_L V	Energy per cycle fJ	Average power μW	Rise delay ps	Fall delay ps
0.4	7.375	4.925		
0.5	6.370	4.216	305.5	348.2
0.6	2.224	1.479e	62.04	95.85
0.7	1.651	1.113	50.41	35.53
0.8	1.463	0.99	26.01	32.95
0.9	1.397	0.93	20.89	24.55
1.0	1.377	0.92	18.32	20.04
1.1	1.414	0.94	16.50	17.09
1.2	1.477	0.98	15.39	15.38

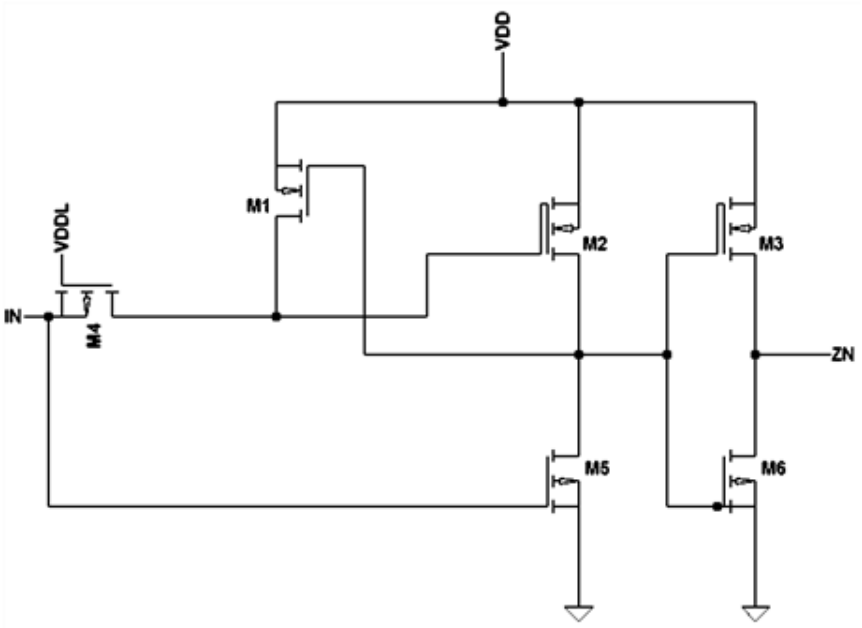

Figure 6.7: Pass transistor logic level converter [31, 50, 51].

Table 6.2: Pass transistor logic level converter, $V_H = 1.2V$.

V_L V	Energy per cycle fJ	Average power μW	Rise delay ps	Fall delay ps
0.4	7.170	4.790		
0.5	6.674	4.387	129.7	498.5
0.6	2.403	1.615	47.62	87.25
0.7	1.736	1.160	29.44	47.10
0.8	1.484	0.975	21.76	34.53
0.9	1.366	0.896	17.84	29.39
1.0	1.287	0.840	15.28	26.50
1.1	1.259	0.797	13.90	24.39
1.2	1.214	0.782	13.13	23.34

in energy with an increase in delay of the gate. for a high voltage NOT gate, the delay is 26.2ps and for a level-shifting NOT gate, the delay is 25.5ps when $V_{thp} = 0.2V$ and 33.5ps when $V_{thp} = 0.4V$.

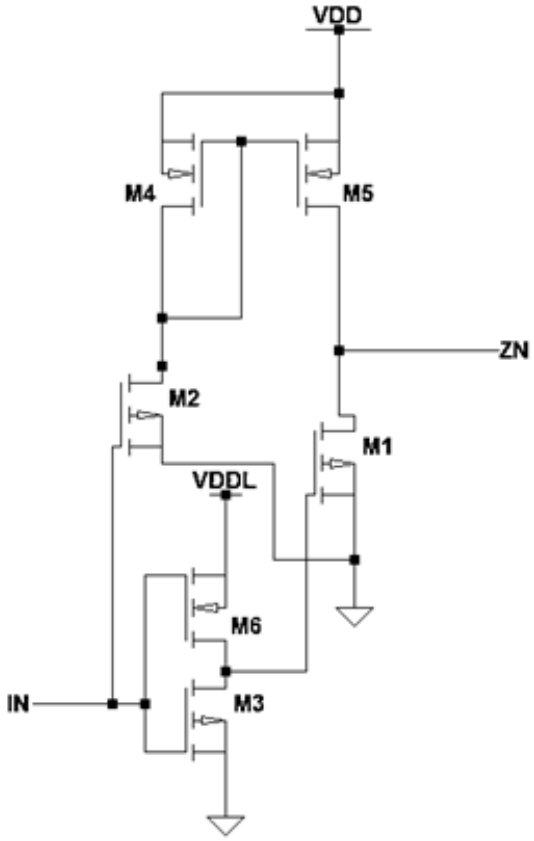

Figure 6.8: Conventional Type II level converter [49, 97]

Table 6.3: Conventional Type II level converter, $V_H = 1.2V$.

V_L V	Energy per cycle fJ	Average power μW	Rise delay ps	Fall delay ps
0.4	5.029	3.365	37.63	137.7
0.5	10.89	7.278	24.00	55.27
0.6	17.84	11.90	18.43	33.26
0.7	24.36	16.24	15.44	23.87
0.8	27.92	18.63	13.37	19.06
0.9	29.42	19.61	12.02	16.19
1.0	30.29	20.19	11.25	14.33
1.1	30.89	20.58	10.79	13.03
1.2	31.34	20.89	10.48	12.08

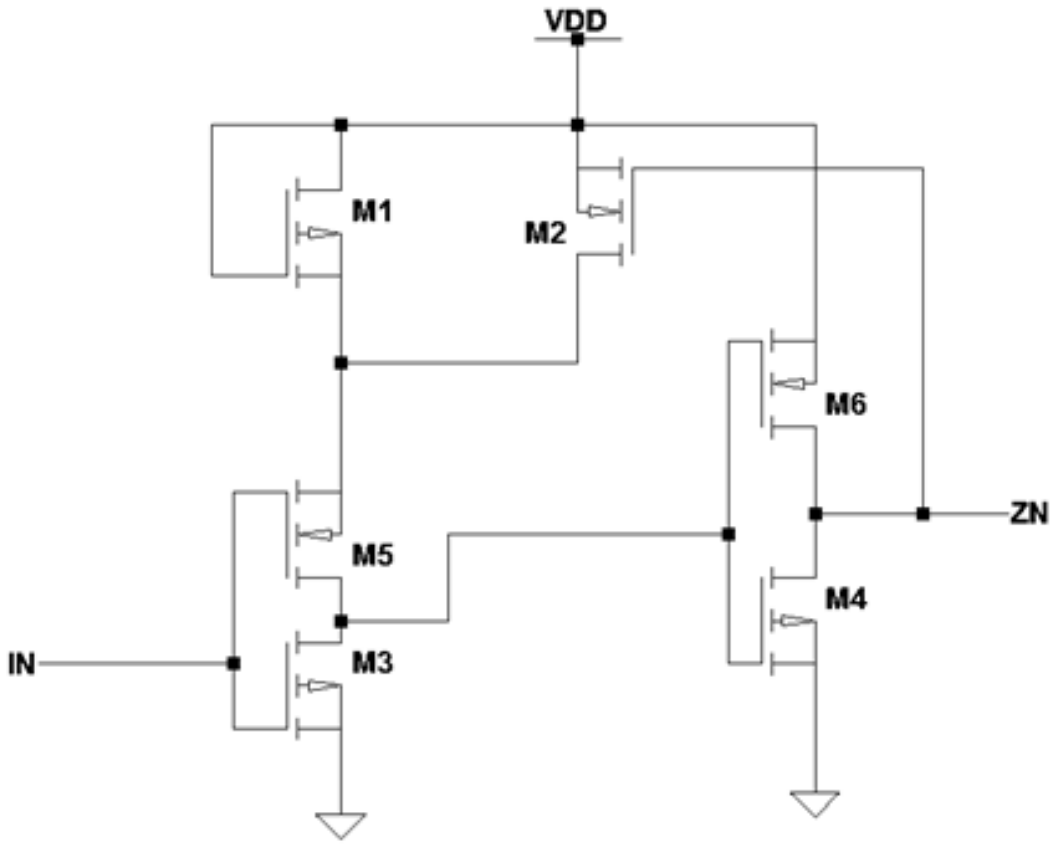

Figure 6.9: Conventional single supply level converter [5, 31, 53].

Table 6.4: Conventional single supply level converter, $V_H = 1.2V$.

V_L V	Energy per cycle fJ	Average power μW	Rise delay ps	Fall delay ps
0.4	7.183	4.799		
0.5	2.902	1.948	64.82	31.40
0.6	2.039	1.377	31.23	33.83
0.7	1.821	1.238	23.34	35.19
0.8	1.878	1.270	19.63	36.18
0.9	2.060	1.399	17.13	36.64
1.0	2.319	1.568	15.84	37.23
1.1	2.627	1.775	14.84	36.84
1.2	2.944	1.988	14.18	36.97

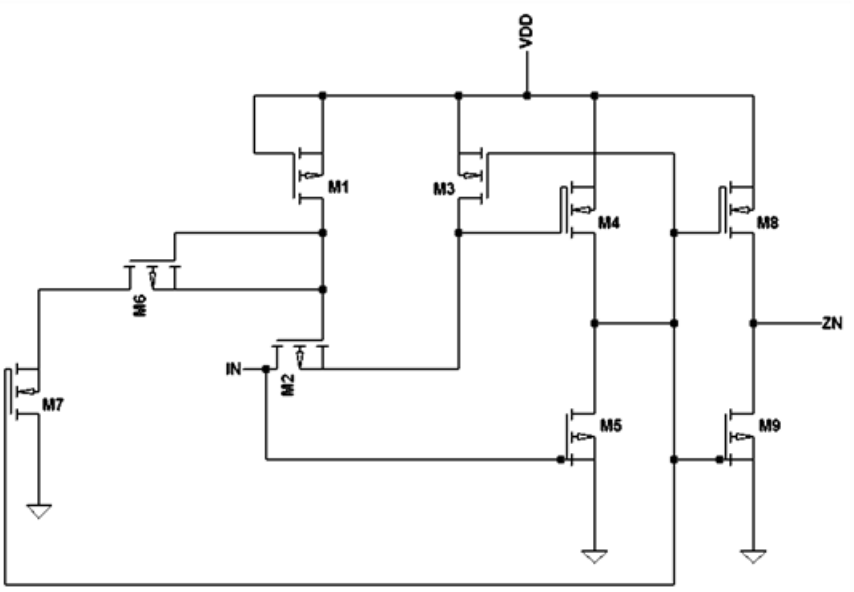

Figure 6.10: Recently published single supply level converter [5].

Table 6.5: Recently published single supply level converter, $V_H = 1.2V$.

V_L V	Energy per cycle fJ	Average power μW	Rise delay ps	Fall delay ps
0.4	7.313	4.880		
0.5	18.03	8.781	127.0	23.34
0.6	11.79	5.841	37.33	25.18
0.7	7.235	4.288	25.40	26.36
0.8	6.142	4.060	20.86	27.00
0.9	6.322	4.221	17.99	27.56
1.0	6.599	4.403	16.08	27.87
1.1	6.914	4.615	14.57	28.08
1.2	7.248	4.840	13.70	28.18

We have discussed Algorithm 3 for a level-converter free dual-voltage design in Chapter 5. When such combinational blocks have to be interfaced with other combinational blocks operating at higher supply voltages, level converting flip-flops are used at the inputs and outputs of the dual-voltage block to account for the changed logic levels.

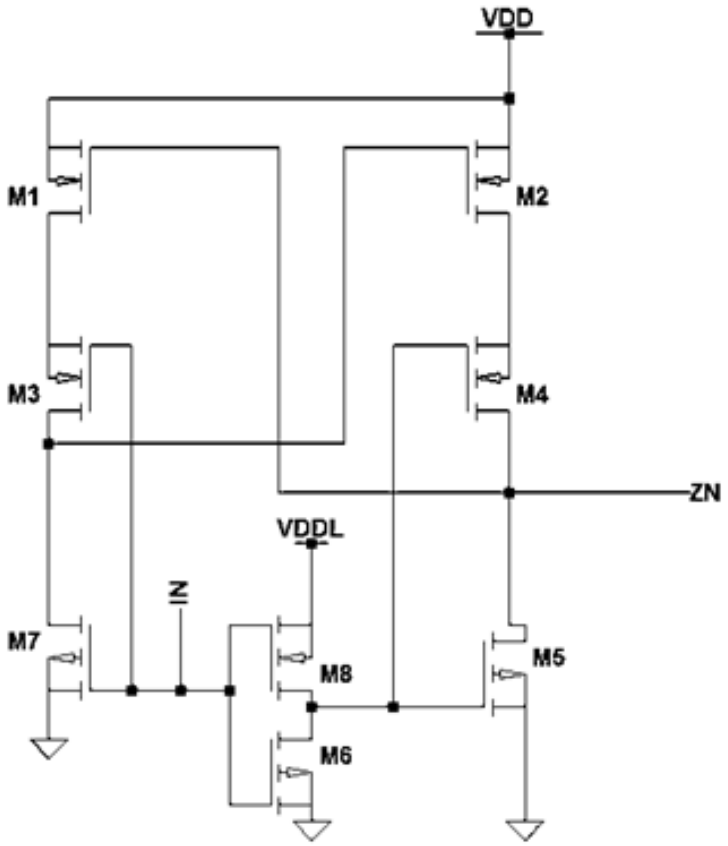

Figure 6.11: Contention mitigated level converter [23, 53, 89]

Table 6.6: Contention mitigated logic level converter, $V_H = 1.2V$.

V_L V	Energy per cycle fJ	Average power μW	Rise delay ps	Fall delay ps
0.4	4.573	3.067	441.4	832.6
0.5	1.204	0.837	47.80	131.2
0.6	0.9860	0.7.12	31.98	60.47
0.7	0.9255	0.656	26.75	38.97
0.8	0.910	0.6377	24.07	28.77
0.9	0.906	0.6304	22.54	23.40
1.0	0.918	0.6273	21.67	19.83
1.1	0.933	0.6440	21.29	17.48
1.2	0.976	0.6648	21.06	15.75

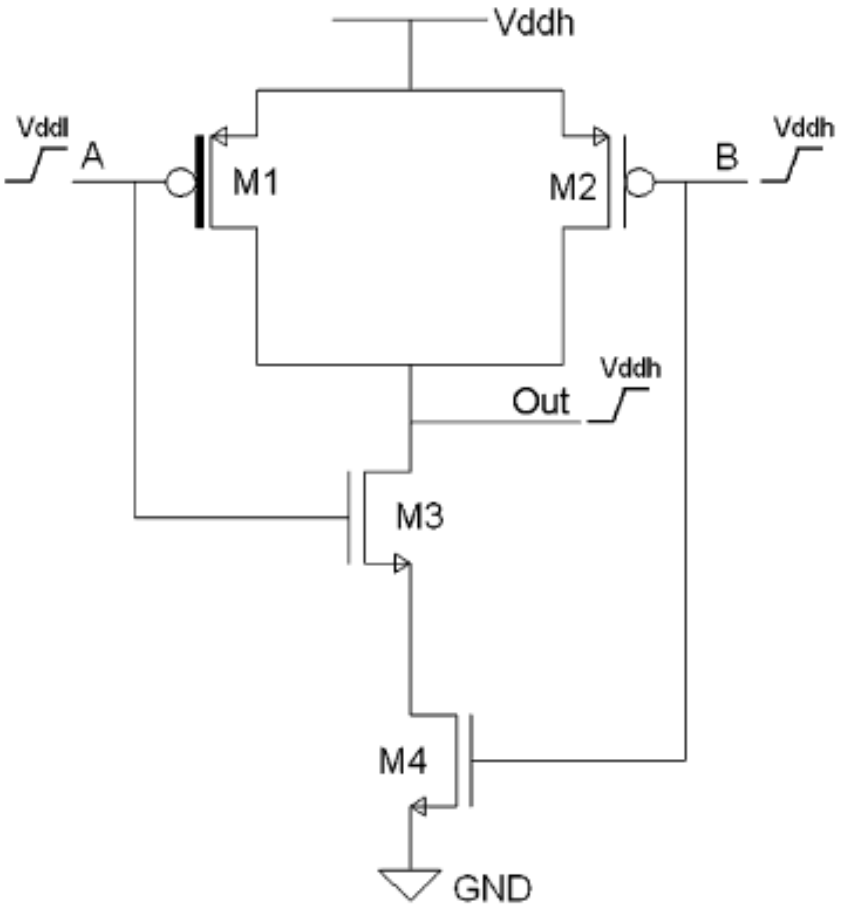

Figure 6.12: Level shifting NAND2 with one high voltage and one low voltage inputs. M1 has higher threshold voltage than M2 [19].

The design of level converting flip-flops (LCFF) is studied widely. The data inputs and clock of an LCFF have low voltage swing and the output has a high voltage swing [66]. In the conventional LCFF [22], the level converting feature is incorporated in the slave latch of a master-slave flip-flop. Hence, it is called a Slave Latch Level Shifting (SLLS) flip-flop. This structure has a large delay especially when the V_L is much lower than the V_H [66]. Another LCFF proposed in [22] is called the Clock-level Shifted Sense Amplifier (CSSA) flip-flop. The low voltage clock and data signals cannot drive the high voltage pMOS transistors in this flip-flop structure [66]. Authors in [71] address this issue by using a high voltage swing clock signal to be applied to the pMOS precharged transistors of a sense amplifier flip-flop to correct the precharging operation. Data is applied to the nMOS and not pMOS, hence,

level shifting is not required for the data signal [66]. Also, the continuous level-shifting of the low-voltage swing clock signal will result in higher energy consumption per cycle [66].

Authors in [90] proposed a pulsed flip-flop which is faster than the master-slave or static flip-flop. The work described in [48] employs internal clock gating to reduce the power consumption at low switching data signals. These flip-flops require high-voltage clock and data signals as they are used to drive the pMOS transistors for the precharge operation [66]. Applying level shifting for these signals is not energy-efficient [66]. Authors in [66, 95] propose a pulsed flip-flop that uses a self-precharging technique and eliminates the need for the clock signal to drive the pMOS transistors. Hence, it is called the Self-Precharging Flip-Flop (SPFF). It also includes the internal clock gating technique to reduce power. For 250nm technology, at $V_H = 2.5V$, $V_L = 1.75V$, the CSSA flip-flop consumes $86.4\mu W$ power and 430ps D-to-Q delay [66]. The SLLS flip-flop consumes $38.9\mu W$ power and 938ps D-to-Q delay [66]. The SPFF single-ended flip-flop consumes $63.1\mu W$ power and 372ps D-to-Q delay [66].

A recent paper [59] proposes a pre-discharge flip-flop (PDFF). The authors also study the unsymmetrical 0-1 and 1-0 output delays of various LCFF discussed in the literature. This is called the metastability of the flip-flops. As the supply voltage is reduced on the clock and/or the data signal, the LCFFs experience various degrees of metastability degradation. They achieve the least amount of degradation by pre-charging the critical nodes back to high voltage [59]. The authors in [31] propose three more LCFF designs.

Chapter 7

Slack-Based Algorithm For Dual Voltage Assignment Using Level Converters

In this chapter we describe an algorithm to assign lower supply voltage without any constraints imposed on the circuit topology. This method is called Enhanced Clustered Voltage Scaling (ECVS) in [92]. This algorithm removes the condition that requires only high voltage gates feeding the low voltage gates and allows the low voltage gate feeding the high voltage gate with an asynchronous level converter (ALC) [50] in between to shift the logic level low to high. This allows more gates to be assigned a lower voltage and higher energy savings are expected. However, the level shifters added at low voltage gate to high voltage gate junctions contribute to additional energy consumption, which needs to be taken into account while calculating the final energy savings. An additional delay penalty is also associated with these level converters. Thus, fast and low power level converters are important in mitigating these penalties.

In both CVS and ECVS algorithms described in [92, 94], we start with all gates at high voltage and then assign the lower power supply to gates by traversing the circuit from the primary outputs to the primary inputs in a levelized order. However, since ECVS performs this assignment simply by visiting gates one at a time in a reverse levelized manner, it still assigns supply voltages in a fundamentally constrained manner [50]. Noting these drawbacks, we do not use levelization in our algorithm for voltage assignment. Our algorithm is related to the existing ECVS approach in that it allows the use of asynchronous level converters. We first describe an example of a chain of ten inverters in the next section.

7.1 An Example of a Chain of Inverters

As in Chapter 5, consider the chain of inverters with a level converter in between shown in Figure 7.2. We use a contention mitigated level-up converter to convert logic level V_1 to V_2. The circuit is simulated with Synopsys HSPICE program [1], with voltages V_1 and V_2 as 0.4V, 0.6V, 0.8V, 1.0V and 1.2V. A 50% duty-cycle clock with 1.5 ns period is applied at the input and a capacitance of 6fF, equivalent to four inverters, used as the load at the output. The results for 90nm technology are presented in Figure 7.1.

As in Figure 5.1, the total energy consumption and delay for the circuit at various values of V_1 and V_2 are reported in Figure 7.1. The energy values shown in the green squares are when V_1 and V_2 are equal, corresponding to a single voltage operation. But these values also include the energy consumption of the level converter, unlike the values in Figure 5.1. The values reported in blue squares below the $V_1 = V_2$ diagonal are the values when V_1 is greater than V_2, i.e., when a high voltage gate is feeding a low voltage gate. The squares above this diagonal represent the operation when V_2 is greater than V_1, i.e., when a low voltage gate feeds a high voltage gate. We observe that the delay measurement in extreme cells fails, represented as infinite delay, when the voltage difference is large. For all cases, although logic 1 level at the output matched V_{DD}, logic 0 levels were at ground.

Energy savings are expected when a low voltage gate feeds a high voltage gate. If we consider the middle square, it represents the energy and delay when $V_1 = 0.8V$ and $V_2 = 0.8V$. This consumes 3.56fJ of energy. Now consider the orange square to the left of this green square. It represents the parameters when $V_1 = 0.6V$ and $V_2 = 0.8V$ and consumes 2.83fJ of energy which gives us a savings of 20.5%. Now, when we go further left, V_1 is 0.4V and energy consumption is 2.74fJ, which gives us savings of 23.03%. Hence, we see that as we reduce the lower voltage, we get more energy savings. But also note that the delay is increasing. Hence, the use of a level converter in dual voltage designs is viable to get energy savings.

V2(V) \ V1(V)	0.4	0.6	0.8	1.0	1.2
1.2	10.44fJ ∞	7.18fJ 249.1ps	7.18fJ 184.0ps	7.98fJ 161.7ps	9.316fJ 153.4ps
1.0	7.13fJ 1198ps	4.39fJ 268.5ps	4.96fJ 203.3ps	5.94fJ 182.8ps	8.05fJ 174.8ps
0.8	2.74fJ 952.5ps	2.83fJ 309.4ps	3.56fJ 251.4ps	4.93fJ 231.8ps	16.14fJ 225.8ps
0.6	1.408fJ 948.8ps	1.91fJ 470.7ps	2.82fJ 418.9ps	10.34fJ 405.7ps	45.31fJ 387.8ps
0.4	0.81fJ 2188ps	1.4fJ 1757ps	7.08fJ 1733ps	6.46fJ ∞	9.75fJ ∞

Figure 7.1: Energy and delay measurements at various values of V_1 and V_2 for the inverter chain of Figure 7.2.

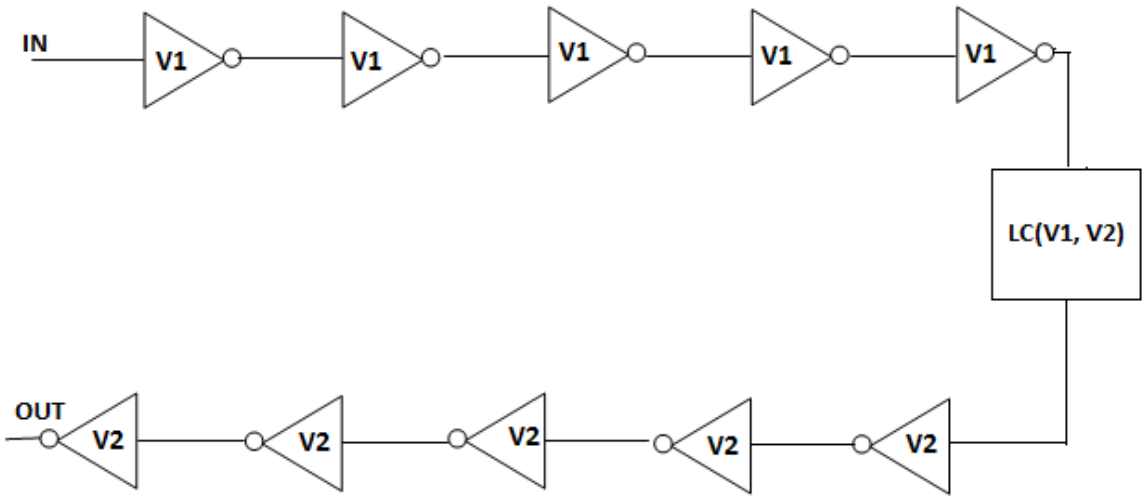

Figure 7.2: A chain of ten inverters with a level converter.

Also, it is worth noting that as we increase the V_H for a given V_L, the energy is increasing and delay is decreasing. This is the effect of using higher voltage gates.

7.2 Algorithm 4

Step 1: Similar to Algorithm 3, initially we assume that all gates are at high voltage, i.e., are connected to V_H supply voltage.

Step 2: Once we obtain the V_L value from Algorithm 2, we assign the gates in group 2 the lower supply voltage.

Step 3: Then we recalculate the slack using Algorithm 1. Theorem 4.2 mandates that no negative slack occurs during slack recalculation.

Step 4: The gates are divided into new groups 1, 2 and 3.

Step 5: We take a small group of high-voltage gates out of group 3 satisfying the condition stated in Theorem 4.3 and assign them the low voltage.

Step 6: The need of a level converter for each of these gates is verified.

Step 7: If a level converter is needed for any of the gates, its delay is changed to the sum of the gate delay and the level converter delay, and then slack is recalculated.

Step 8: If any negative slack is encountered, then the gate is put back to high voltage.

Step 9: Gate slacks are calculated again.

Step 10: The gates are redivided into groups 1, 2 and 3.

Step 11: Steps 5 to 10 are repeated on the remaining gates until there are no high-voltage gates left in group 3.

The only difference between Algorithm 3 and 4 is that Algorithm 3 requires an additional constraint that forbids the low voltage gates from being the input of the high voltage gates. Algorithm 4 thus gives us the gates which can be put in low voltage to get maximum energy savings.

7.3 Results

All simulations described in this section are done as explained in Chapter 5. A contention mitigated level converter design is used in the simulations. The results for 90nm ISCAS'85

Table 7.1: Optimal lower supply voltage values (V_L) and energy savings using Algorithms 2 and 4 for ISCAS'85 benchmark circuits using Contention Mitigated Level Converter, $V_H = 1.2V$.

Benchmark circuit	Algorithm 2 V_L V	Total gates	Algorithm 4 Gates in low voltage	Algorithm 4 Number of Level Converters	Algorithm 4 $E_{savg.-expc.}$ %	SPICE Results $E_{single\ VDD}$ fJ	SPICE Results $E_{dualVDD}$ fJ	SPICE Results $E_{savg.-obs.}$ %	CPU time to run Algorithms 2 and 4 s*
c432	0.84	154	81	43	26.82	161.3	151.7	5.95	3.11
c499	0.91	493	247	67	21.30	463	444.9	3.91	64.13
c880	0.58	360	286	47	60.89	277.6	143.1	48.45	14.11
c1355	0.92	469	193	61	16.96	455.2	426.5	6.31	47.87
c1908	0.77	584	385	95	38.78	496.5	1237	-149.14	93.70
c2670	0.62	901	633	126	51.50	660.3	2787	-322.08	229.97
c3540	0.61	1270	881	232	51.45	1843	3727	-102.23	1047.33
c5315	0.60	2077	1602	234	57.85	2320	3214	-38.54	2320.81
c6288	0.73	2407	1189	552	47.34	1932	1781	7.82	9016.82
c7552	0.61	2823	1971	380	51.78	2465	5328	-116.15	10890.49

* Intel Core i5 2.30GHz, 4GB RAM

Table 7.2: Optimal lower supply voltage values (V_L) and energy savings using Algorithms 2 and 4 for ISCAS'85 benchmark circuits using Contention Mitigated Level Converter, when T_c increased by 5%, $V_H = 1.2V$.

Benchmark circuit	Algorithm 2 V_L V	Total gates	Algorithm 4 Gates in low voltage	Algorithm 4 Number of Level Converters	Algorithm 4 $E_{savg.-expc.}$ %	SPICE Results $E_{single\ VDD}$ fJ	SPICE Results $E_{dualVDD}$ fJ	SPICE Results $E_{savg.-obs.}$ %	CPU time to run Algorithms 2 and 4 s*
c432	0.80	154	80	43	28.86	161.3	149.1	7.56	3.83
c499	0.78	493	238	67	27.87	463	415.2	10.33	63.13
c880	0.60	360	294	47	61.25	277.6	144.3	48.02	13.92
c1355	0.80	469	211	61	24.99	455.2	442.6	2.77	59.89
c1908	0.72	584	387	95	42.41	496.5	1393	-180.56	96.46
c2670	0.63	901	683	126	54.91	660.3	1463	-127.57	184.26
c3540	0.67	1270	983	232	53.27	1843	4143	-124.80	772.88
c5315	0.59	2077	1610	238	58.78	2320	3696	-59.31	2307.67
c6288	0.67	2407	1025	503	29.31	1932	1491	22.83	9265.48
c7552	0.57	2823	1897	415	53.04	2465	7804	-216.59	10523.00

* Intel Core i5 2.30GHz, 4GB RAM

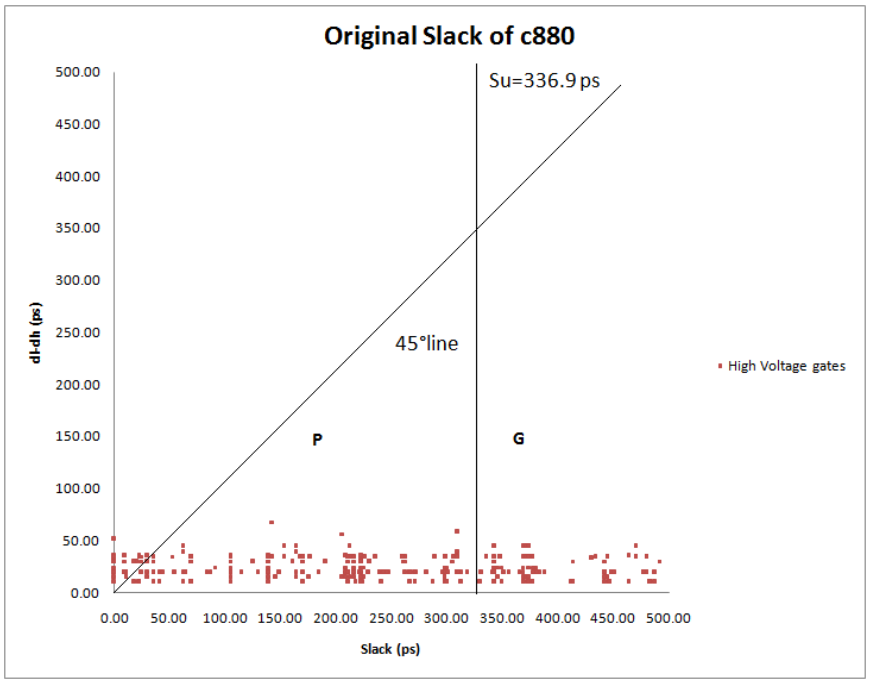

Figure 7.3: Delay increment versus initial slack for gates of c880 circuit, $V_H = 1.2V$, $V_L = 0.58V$.

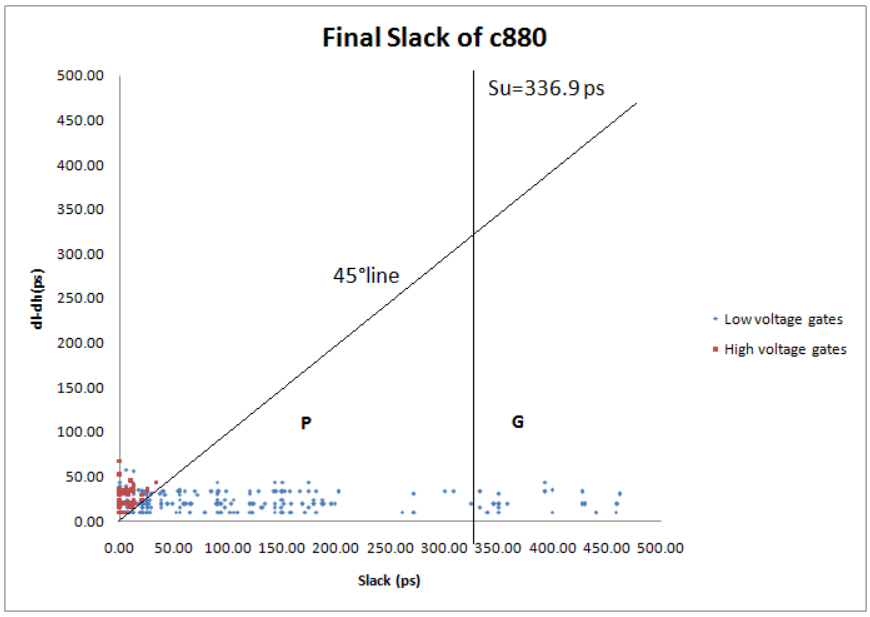

Figure 7.4: Delay increment versus final slack for gates of c880 circuit, $V_H = 1.2V$, $V_L = 0.58V$.

benchmark circuits are tabulated in Table 7.1 and when their critical timings are increased by 5% are tabulated in Table 7.2.

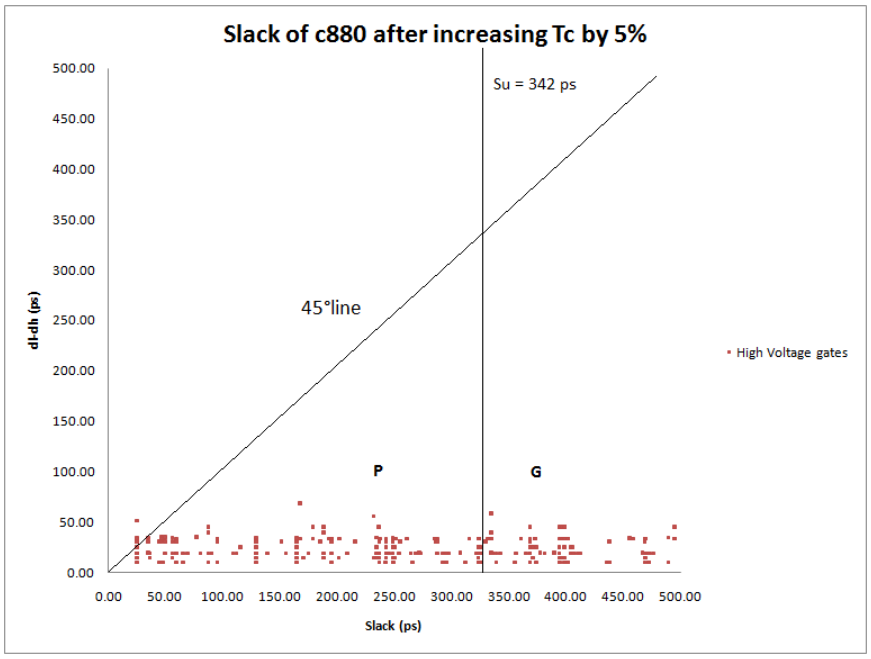

Figure 7.5: Delay increment versus initial slack for gates of c880 circuit when Tc is increased by 5%, $V_H = 1.2V,\ V_L = 0.6V$.

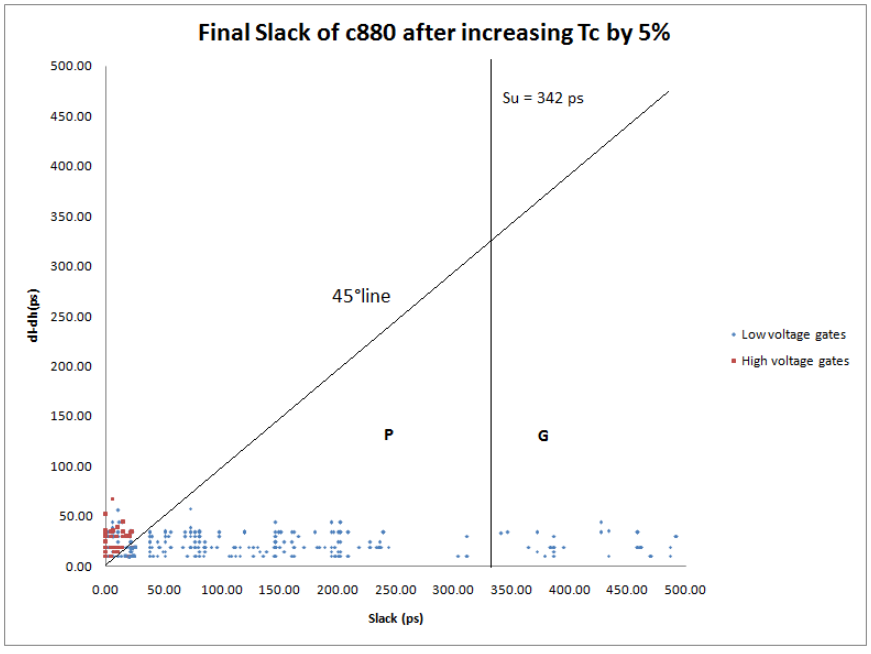

Figure 7.6: Delay increment versus final slack for gates of c880 circuit when Tc is increased by 5%, $V_H = 1.2V,\ V_L = 0.6V$.

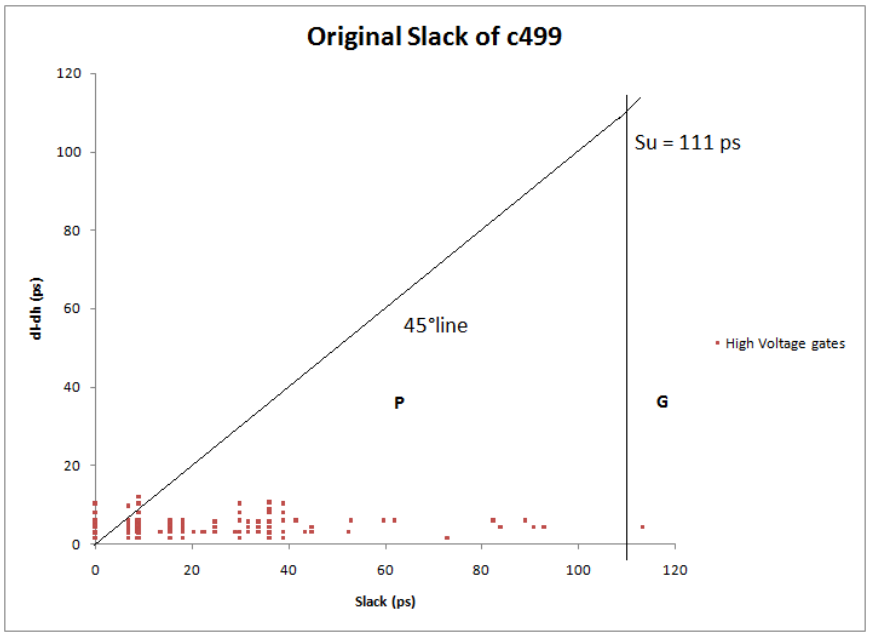

Figure 7.7: Delay increment versus initial slack for gates of c499 circuit, $V_H = 1.2V$, $V_L = 0.91V$.

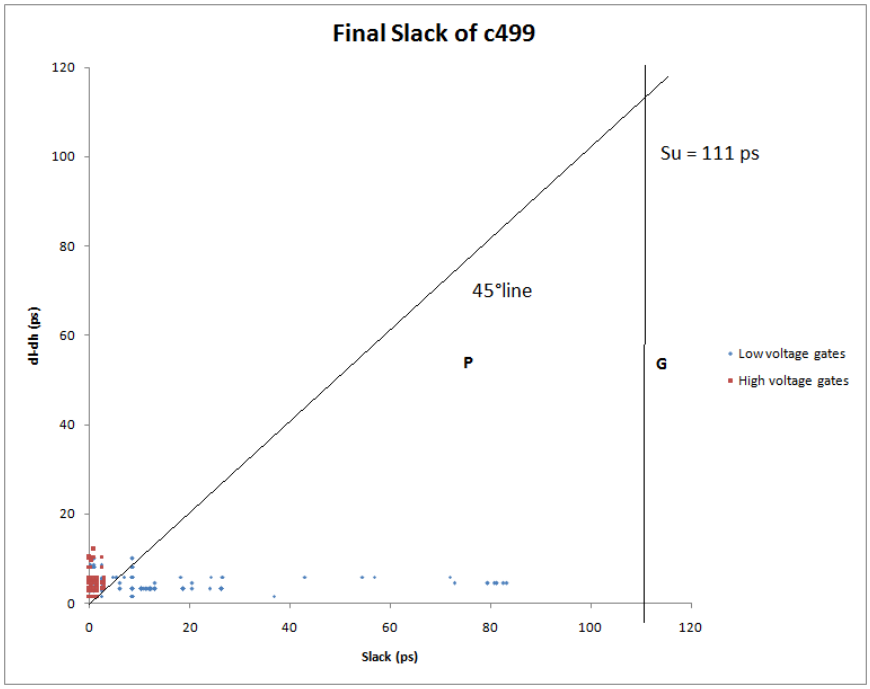

Figure 7.8: Delay increment versus final slack for gates of c499 circuit, $V_H = 1.2V$, $V_L = 0.91V$.

As in Chapter 5, Figures 7.3, 7.4, 7.7 and 7.8 show "delay increment versus slack" graphs for the initial slacks and final slacks for the circuits c880 and c499, respectively. The brown markers indicate gates in high voltage and blue markers indicate gates in low voltage. It is seen that initially all gates are in low voltage. Then, once Algorithms 2 and 4 are used and slacks are recalculated using Algorithm 1, all gates slacks are reduced. And all the high voltage gates have moved above the 45^o line. The low voltage gates still below the 45^o line are gates with very large slacks and there are no high voltage gates below the line, indicating that no gate is left unassigned due to topological constraints, as in the case of Algorithm 3.

Figures 5.5 and 5.6, show "delay increment versus slack" graphs for the initial slacks and final slacks for the circuits c880 when its critical timing is allowed to increase by 5%. From the graphs it can be seen that the slacks of the gates have moved towards the right due to increased critical path delay, which in turn increases the gate slacks. Also, that the final number of gates in high voltage is less which can be seen by the reduced density of brown dots.

It is seen from Table 7.1 that up to 60% energy savings are expected from Algorithm 4. These expectations are met in SPICE simulations of some of the benchmark circuits. We observe higher energy savings in circuits like c6288 using Algorithm4 than using Algorithm3. This is because in circuits where there are several long paths that have delays comparable to the critical path delay, c6288 is a multiplier, it is difficult to put the gates in low voltage with the topological constraints imposed by Algorithm 3. Since, Algorithm 4 lifts those conditions, we can put more gates in low voltage. Hence, even though we use a large number of level converters in c6288, we still get energy savings.

It is observed that energy savings are not good for the circuits in which you have both higher gates in low voltage and the low supply voltage is very low. For example see c7552, which has 1897 gates in low voltage at a supply of only 0.57V. Leakage is higher at lower supply voltages. Now, when we assign a big fraction of gates to low supply voltage the leakage energy consumption shoots up and any gains in the dynamic energy reduction are

more than offset by this leakage energy increase. To prove this, we simulated c7552 at 0.8V and got energy consumption of 5105fJ, which is still larger than 2465fJ, the single voltage case, but smaller than 7804fJ, as given in Table 7.2. The number of gates in low voltage was not changed. Perhaps we need to increase the voltage further or reduce the number of gates in low voltage by using Algorithm 4 at that voltage to get energy savings in these cases. Hence, leakage energy needs to be considered when predicting estimated energy savings.

We simulate c1908 using V_{L2}, obtained from Algorithm 2. It is found to be 1V and puts 437 gates in low voltage using 81 level converters. But still, it consumes 928.3fJ of energy, which is almost double the single voltage energy given by Table 7.1. This is due to the level converter overheads. We need to accommodate the level converter energy overheads to get a reasonably good energy savings.

We see that our Algorithm 4 gives up to 48% savings in some cases, such as c880, and no savings in some cases, whether due to high leakage or due to level converter overheads.

The energy savings we got from Algorithms 3 and 4 are the lower bounds. The benchmark circuits have been simulated with a clock period which equals the respective critical path delays. If the required critical timing is relaxed, then more gates can be assigned low voltage which will result in more energy savings. This fact is demonstrated by Tables 7.2 and 5.2 which present the results when the critical path delay is relaxed by 5%.

When the critical timing is relaxed, the gate slacks increase and more gates fall below the 45^o line in the 'Delay increment versus Slack' plot. This means we have more gates in sets 'P' and 'G', and hence, more gates can be assigned low voltage. This trend can be seen in Figures 7.5 and 7.6.

Chapter 8

Conclusion and Future Work

This works describes new algorithms for dual voltage design. We find the optimum V_L value using a new $O(n)$ algorithm. Using this value, a new algorithm determines a set of gates that can be assigned a lower supply voltage without violating the positive slack constraint. The gates are divided into groups based on their slacks and the difference of the low voltage delays and the high voltage delays. An energy savings of up to 60% is expected from this method. Also, the results are obtained at lower CPU times than the previously published results [41, 50]. We use the $O(n)$ complexity slack calculation algorithm iteratively to put the gates in low voltage. If we put one gate at a time to low voltage the complexity of this algorithm will be $O(n^2)$. In practice, it is observed to be close to linear time. This is because we take a group of gates at a time for low-voltage assignment and hence the effective n is reduced after each iteration.

8.1 Using the Proposed Algorithms at Higher Abstraction Levels

The algorithms proposed in Chapters 5 and 7 work for the gate-level circuit. We can extend these algorithms to the register-transfer level (RTL) or higher levels of abstraction. The Algorithm 4 will be very effective at higher level of abstraction than at the gate level because the number of asynchronous level converters used at the gate-level can be large and thus, delay and energy overheads might form a significant portion of the total energy consumed by the dual-voltage design. At the RTL level, the register files can be implemented using level converting flip-flops discussed in Chapter 6 and the combinational blocks in between the register files can be designed using Algorithm 3 (at the gate-level, in this case) to get a level-converter free dual-voltage design. Further, the various combinational blocks

in the circuit can be assigned lower voltage using Algorithm 4 with level converting flip-flops at the interface where a low voltage block drives a high voltage block. At the system on chip (SOC) level, the various IP-cores can be assigned lower voltage by using Algorithm 4 and suitable level converting elements at the interfaces. For using the Algorithm 2 in order to determine the lower supply voltage at higher abstraction, we have to characterize each unique block/core for delay, activity, capacitance and power at various voltages. once this is done, we can use Algorithm 2 to obtain the optimum lower supply voltage.

8.2 Accommodate Level Converter Overheads

Similar to Algorithms 3 and 4, initially we assume that all gates are at high voltage, i.e., are connected to V_H supply voltage. Once we obtain the V_L value from Algorithm 2, we assign the gates in group 2 the lower supply voltage. Then we recalculate the slack using Algorithm 1. Theorem 4.2 mandates that no negative slack occurs during slack recalculation. Also, the gates are divided into new groups 1, 2 and 3 once we calculate the slack again.

We take a small group of high-voltage gates out of group 3, satisfying the condition stated in Theorem 4.3, and assign them the low voltage. The need of a level converter for each of these gates is verified. If a level converter is needed for any of the gates, its delay is changed to the sum of the gate delay and the level converter delay, and then slack is recalculated. If any negative slack is encountered, then the gate is put back to high voltage. If no timing violation occurs, then we check for energy. Energy is calculated using equation 4.32. The level converter overhead times the number of level converters is added to this energy. If this energy exceeds the energy consumption of a CVS circuit, the gate is put back to high voltage. This process is repeated on the remaining gates. The gates are redivided into groups 1, 2 and 3 after each iteration. We stop these iterations when there are no valid high-voltage gates left in group 3. Similar to Algorithm 3, we will have some high voltage gates still left in groups 2 and 3, which is unavoidable.

This algorithm is very simplistic in nature. This makes a decision to use a level converter by putting one gate in low voltage and checking energy consumption. It ignores the fact that when we put one gate in low voltage, we may not achieve significant energy savings. But this will allow us to a number of gates at the fan-in of this gate to low voltage and thus, derive a larger savings. Hence, a good heuristic needs to be developed to accommodate this. Similar work has been proposed in [50], where the authors use a greedy heuristic to group the possible low voltage gates.

8.3 Dual Threshold Design

The leakage currents are exponentially dependent on the threshold voltage of the device. As seen in Chapter 2, we can assign higher threshold devices in non-critical paths to reduce leakage power. Slack-based algorithms similar to the ones proposed in this work can be developed to determine and assign the high threshold devices in dual-threshold design for minimum-energy.

8.4 Multiple-Supply Voltage Design

Using more than two supply voltages might be advantageous in few designs in terms of energy savings. The algorithms presented in this work can be modified to include more than two supply voltages. In such a case, level converter designs should be redone for all the different logic level interfaces.

8.5 Simultaneous Multiple V_{DD} and Multiple V_{th} Designs

As described in Chapter 2, many works describe these kind of techniques. It would be a worthwhile experiment to use multiple supply voltage and multiple threshold devices in order to get the timing optimized minimum energy designs. At some point the advantage of these techniques will reduce. In [73], the authors claim that multiple supplies and multiple

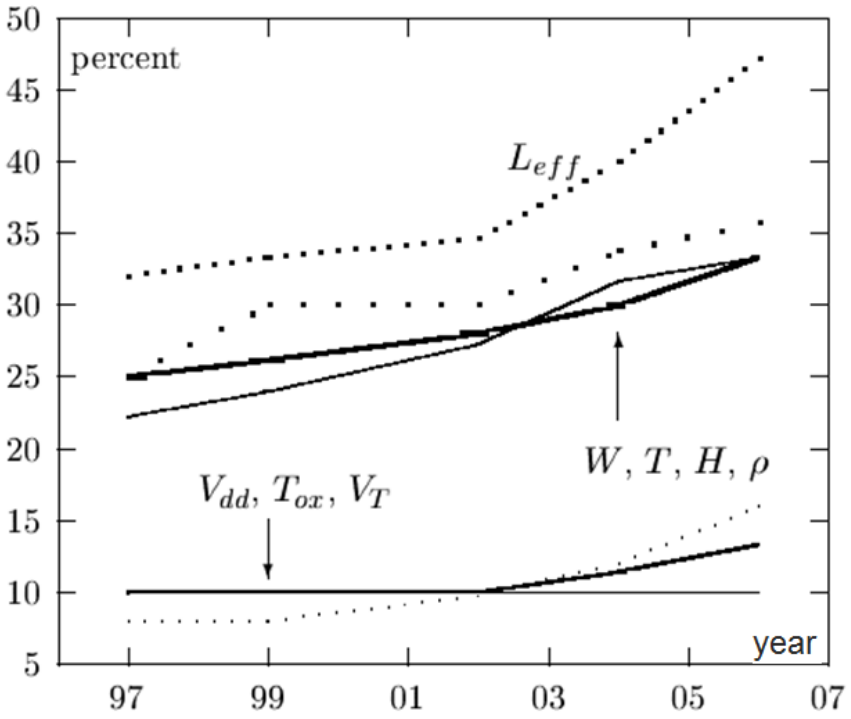

Figure 8.1: Trends in parameter variations [70].

thresholds might not be advantageous when the number exceeds four. Also, authors in [33] claim that the benefit of going from two to three voltages never exceeded 15%.

8.6 Leakage Energy Reduction

Leakage is higher in sub-90nm technology and at lower supply voltages which are closer to the device threshold. Efficient algorithms need to be designed to minimize this leakage energy consumption. This becomes more important because of the faster pace of technology scaling.

8.7 Process Variations

With the reduction of MOSFET dimensions, manufacturing process variations have been increasing in the CMOS technology. According to authors of [61], the variation in threshold voltage for a 45nm CMOS process is about 42% and the variation in channel length is about

10%. Authors of [54] state that the intra-die variations in channel length were observed to be 35% of total variations in 130nm CMOS and 60% in 70nm CMOS. It is observed that there is an increasing trend in the variations with decreasing technology node. The trends in various parameter variations are discussed in [70] and shown in Figure 8.1.

Authors in [38] have worked on supply voltage selection for reducing the impact of process variations on timing. Similarly, we can find ways of voltage assignment that can reduce the effect of process variations.

Hence, the proposed algorithms should be modified to include the effects of these variations in the proposed design in order to extend the usability of the algorithms in aggressively scaled devices.

8.8 Short Paths

In the proposed work, the longest path problem is solved, i.e., we make sure that data signals through all paths in the circuit do not arrive late at the outputs of the circuit. It will be interesting to work on the short path problem. As discussed in Chapter 3, we propose an algorithm to make sure that a data signal does not reach the output before the required time. Then we can start from all low voltage design and start assigning high voltages to make the circuit faster, but not fast enough to reach the outputs before the specified time.

Bibliography

[1] "HSPICE®Reference Manual: Commands and Control Options, Version D-2010.03-SP1," June 2010. http://www.synopsys.com/Tools/Verification/AMS Verification/ CircuitSimulation/HSPICE/Pages/default.aspx (accessed on October 14, 2011).

[2] A. Abdollahi, F. Fallah, and M. Pedram, "Leakage Current Reduction in CMOS VLSI Circuits by Input Vector Control," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 2, pp. 140–154, February 2004.

[3] A. Abdollahi, M. Pedram, and F. Fallah, "Runtime Mechanisms for Leakage Current Reduction in CMOS VLSI Circuits," in *Proceedings of the International Symposium on Low Power Design*, 2002, pp. 213–218.

[4] V. D. Agrawal, "Synchronous Path Analysis in MOS Circuit Simulator," in *Proceedings of the 19th Design Automation Conference*, 1982, pp. 629–635.

[5] J.-Y. An, H. S. Park, and Y. H. Kim, "Level Up/Down Converter with Single Power Supply Voltage for Multi V_{DD} Systems," *Journal of Semiconductor Technology and Science*, vol. 10, no. 1, pp. 55–60, March 2010.

[6] S. Augsburger and B. Nikolic, "Combining Dual-Supply, Dual-Threshold and Transistor Sizing for Power Reduction," in *Proceedings of the IEEE International Conference on Computer Design*, 2002, pp. 316–321.

[7] J. Bhasker and R. Chadha, *Static Timing Analysis for Nanometer Designs - A Practical Approach*. Springer, 2009.

[8] N. Chabini, I. Chabini, E. Aboulhamid, and Y. Savaria, "Methods for Minimizing Dynamic Power Consumption in Synchronous Designs with Multiple Supply Voltages," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 3, pp. 346–351, March 2003.

[9] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-Power CMOS Digital Design," *IEEE Journal of Solid-State Circuits*, vol. 27, no. 4, pp. 473–484, April 1992.

[10] C. Chen, A. Srivastava, and M. Sarrafzadeh, "On Gate Level Power Optimization Using Dual-Supply Voltages," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 9, no. 5, pp. 616–629, October 2001.

[11] D. Chen, J. Cong, Y. Fan, and J. Xu, "Optimality Study of Resource Binding With Multi Vdds," in *Proceedings of the 43rd annual Design Automation Conference*, 2006, pp. 580–585.

[12] D. Chen, J. Cong, F. Li, and L. He, "Low-Power Technology Mapping for FPGA Architectures with Dual Supply Voltages," in *Proceedings of the 12th International Symposium on Field Programmable Gate Arrays*, 2004, pp. 109–117.

[13] D. Chen, J. Cong, and J. Xu, "Optimal Module and Voltage Assignment for Low-Power," in *Proceedings of the Asia and South Pacific Design Automation Conference*, 2005, pp. 850–855.

[14] J. C. Chi, H. H. Lee, S. H. Tsai, and M. C. Chi, "Gate Level Multiple Supply Voltage Assignment Algorithm for Power Optimization Under Timing Constraint," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 6, pp. 637 –648, June 2007.

[15] R. L. S. Ching, E. F. Y. Young, K. C. K. Leung, and C. Chu, "Post Placement Voltage Island Generation," in *Proceedings IEEE/ACM International Conference on Computer-Aided Design*, November 2006, pp. 641–646.

[16] D. G. Chinnery and K. Keutzer, "Linear Programming for Sizing, Vth and Vdd Aassignment," in *Proceedings of the International Symposium on Low Power Electronics and Design*, 2005, pp. 149–154.

[17] D. G. Chinnery and K. Keutzer, *Closing the Power Gap Between ASIC and Custom - Tools and Techniques for Low Power Design*. Springer, 2007.

[18] Y. S. Dhillon, A. U. Diril, A. Chatterjee, and A. D. Singh, "Low-Power Dual Vth Pseudo Dual Vdd Domino Circuits," in *Proceedings of the 17th symposium on Integrated Circuits and System Design*, 2004, pp. 273–277.

[19] A. Diril, Y. Dhillon, A. Chatterjee, and A. Singh, "Level-Shifter Free Design of Low Power Dual Supply Voltage CMOS Circuits Using Dual Threshold Voltages," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 13, no. 9, pp. 1103 –1107, September 2005.

[20] C. Forzan and D. Pandini, "Statistical Static Timing Analysis: A Survey," *Integration, the VLSI Journal*, vol. 42, no. 3, pp. 409–435, 2009.

[21] M. Hamada, Y. Ootaguro, and T. Kuroda, "Utilizing Surplus Timing for Power Reduction," in *Proceedings of the IEEE Custom Integrated Circuits Conference*, 2001, pp. 89–92.

[22] M. Hamada, M. Takahashi, H. Arakida, A. Chiba, T. Terazawa, T. Ishikawa, M. Kanazawa, M. Igarashi, K. Usami, and T. Kuroda, "A Top-down Low Power Design Technique Using Clustered Voltage Scaling With Variable Supply-voltage Scheme," in *Proceedings of the IEEE Custom Integrated Circuits Conference*, May 1998, pp. 495 –498.

[23] S.-K. Han, K. Park, B.-S. Kong, and Y.-H. Jun, "High Speed Low Power Bootstrapped Level Converter for Dual Supply Systems," in *Proc. IEEE Asia Pacific Conference onCircuits and Systems*, December 2010, pp. 871–874.

[24] R. B. Hitchcock, G. L. Smith, and D. D. Cheng, "Timing Analysis of Computer Hardware," *IBM Journal of Research and Development*, vol. 26, no. 1, pp. 100–105, January 1982.

[25] M. Horowitz, T. Indermaur, and R. Gonzalez, "Low-Power Digital Design," in *Digest of Technical Papers, IEEE Symposium Low Power Electronics*, October 1994, pp. 8–11.

[26] J. Hu, Y. Shin, N. Dhanwada, and R. Marculescu, "Architecting Voltage Islands in Core-Based System-on-a-Chip ," in *Proceedings of the International Symposium on Low Power Electronics and Design*, August 2004, pp. 180–185.

[27] Y. Hu, Y. Lin, L. He, and T. Tuan, "Simultaneous Time Slack Budgeting and Retiming for Dual-V_{dd} FPGA Power Reduction," in *Proceedings of the 43rd Annual Design Automation Conference*, 2006, pp. 478–483.

[28] S. Hua and G. Qu, "Approaching the Maximum Energy Saving on Embedded Systems with Multiple Voltages," in *Proc. International Conference on Computer Aided Design*, November 2003, pp. 26–29.

[29] S. Hua and G. Qu, "Energy-Efficient Dual-Voltage Soft Real-Time System with (m,k)-firm Deadline Guarantee," in *Proceedings of the International Conference on Compilers, Architecture, and Synthesis for Embedded Systems*, 2004, pp. 116–123.

[30] W. Hung, Y. Xie, N. Vijaykrishnan, M. Kandemir, M. Irwin, and Y. Tsai, "Total Power Optimization through Simultaneously Multiple-VDD Multiple-VTH Assignment and Device Sizing with Stack Forcing," in *Proceedings of the International Symposium on Low Power Electronics and Design*, August 2004, pp. 144–149.

[31] F. Ishihara, F. Sheikh, and B. Nikolic, "Level Conversion for Dual Supply Systems," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 2, pp. 185–195, February 2004.

[32] M. C. Johnson and K. Roy, "Optimal Selection of Supply Voltages and Level Conversions During Data Path Scheduling Under Resource Constraints," in *Proceedings of the International Conference on Computer Design*, 1996, pp. 72–77.

[33] M. C. Johnson and K. Roy, "Datapath Scheduling With Multiple Supply Voltages and Level Converters," *ACM Transactions on Design Automation of Electronic Systems*, vol. 2, pp. 227–248, July 1997.

[34] N. P. Jouppi, "Timing Analysis for nMOS VLSI," in *Proceedings of the 20th Design Automation Conference*, 1983, pp. 411–418.

[35] M. Keating, D. Flynn, R. Aitken, A. Gibbons, and K. Shi, *Low Power Methodology Manual: For System-on-Chip Design*. Springer, 2007.

[36] A. Keshavarzi, S. Ma, S. Narendra, B. Bloechel, K. Mistry, T. Ghani, S. Borkar, and V. De, "Effectiveness of Reverse Body Bias for Leakage Control in Scaled Dual V_t CMOS ICs," in *Proceedings of the International Symposium on Low Power Design*, 2001, pp. 207–212.

[37] M. M. Khellah and M. I. Elmasry, "Power Minimization of High-Performance Submicron CMOS Circuits Using a Dual-Vdd Dual-Vth (DVDV) Approach," in *Proceedings of the International Symposium on Low power Electronics and Design*, 1999, pp. 106–108.

[38] B. Kheradmand-Boroujeni, C. Piguet, and Y. Leblebici, "Logic Architecture and VDD Selection for Reducing the Impact of Intra-Die Random VT Variations on Timing," in *Proceedings of the 20th International Conference on Integrated Circuit and System Design: Power and Timing Modeling, Optimization and Simulation*, 2011, pp. 170–179.

[39] H.-O. Kim and Y. Shin, "Analysis and Optimization of Gate Leakage Current of Power Gating Circuits," in *Proceedings of the Asia and South Pacific Design Automation Conference*, 2006, pp. 565–569.

[40] K. Kim, *Ultra Low Power CMOS Design*. PhD thesis, Auburn University, ECE Dept., Auburn, AL, May 2011.

[41] K. Kim and V. D. Agrawal, "Dual Voltage Design for Minimum Energy Using Gate Slack," in *Proceedings of the IEEE International Conference on Industrial Technology*, March 2011, pp. 419–424.

[42] K. Kim and V. D. Agrawal, "Minimum Energy CMOS Design With Dual Subthreshold Supply and Multiple Logic-Level Gates," in *Proceedings of the 12th International Symposium on Quality Electronic Design*, March 2011, pp. 689–694.

[43] K. Kim and V. D. Agrawal, "True Minimum Energy Design Using Dual Below-Threshold Supply Voltages," in *Proceedings of the 24th International Conference on VLSI Design*, January 2011, pp. 292–297.

[44] K. Kim and V. D. Agrawal, "Ultra Low Energy CMOS Logic Using Below Threshold Dual Voltage Supply," *Journal of Low Power Electronics*, vol. 7, no. 4, October 2011.

[45] W. Kim, J. Kim, and S. Lyul-Min, "Dynamic Voltage Scaling Algorithm for Fixed-Priority Real-Time Systems Using Work-Demand Analysis," in *Proceedings of the International Symposium on Low Power Electronics and Design*, August 2003, pp. 396–401.

[46] W. Kim, J. Kim, and S. Min, "A Dynamic Voltage Scaling Algorithm for Dynamic-Priority Hard Real-Time Systems Using Slack Time Analysis," in *Proceedings of the Conference on Design, Automation and Test in Europe*, 2002, pp. 788–795.

[47] T. I. Kirkpatrick and N. R. Clark, "PERT as an Aid to Logic Design," *IBM Journal of Research and Development*, vol. 10, no. 2, pp. 135–141, March 1966.

[48] B.-S. Kong, S.-S. Kim, and Y.-H. Jun, "Conditional-capture Flip-flop for Statistical Power Reduction," *IEEE Journal of Solid-State Circuits*, vol. 36, no. 8, pp. 1263 –1271, August 2001.

[49] K.-H. Koo, J.-H. Seo, M.-L. Ko, and J.-W. Kim, "A New Level Up Shifter for High Speed and Wide Range Interface in Ultra Deep Sub Micron," in *Proc. IEEE International Symposium on Circuits and Systems*, volume 2, May 2005, pp. 1063–1065.

[50] S. H. Kulkarni, A. N. Srivastava, and D. Sylvester, "A New Algorithm for Improved V_{DD} Assignment in Low Power Dual V_{DD} Systems," in *Proceedings of the International Symposium on Low Power Design*, 2004, pp. 200–205.

[51] S. H. Kulkarni and D. Sylvester, "Fast and Energy Efficient Asynchronous Level Converters for Multi-V_{DD} Design [CMOS ICs]," in *Proceedings of IEEE International Systems on Chip (SOC) Conference* , September 2003, pp. 169–172.

[52] S. H. Kulkarni and D. Sylvester, "High Performance Level Conversion for Dual V_{DD} Design," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 9, pp. 926–936, September 2004.

[53] M. Kumar, S. K. Arya, and S. Pandey, "Level Shifter Design for Low Power Applications," *International Journal of Computer Science & Information Technology*, vol. 2, no. 5, October 2010.

[54] Y. Kumar and H. C. Chien-In, "Process Variation Aware Transistor Sizing for Load Balance of Multiple Paths in Dynamic CMOS for Timing Optimization," *Journal of Computers*, vol. 3, no. 2, pp. 21–28, 2008.

[55] S. Kundu, "An Incremental Algorithm for Identification of Longest (Shortest) Paths," *Integration, the VLSI Journal*, vol. 17, pp. 25–31, August 1994.

[56] T. Kuroda and M. Hamada, "Low-Power CMOS Digital Design with Dual Embedded Adaptive Power Supplies," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 4, pp. 652–655, April 2000.

[57] D. Lackey, P. Zuchowski, T. Bednar, D. Stout, S. Gould, and J. Cohn, "Managing Power and Performance for System-on-Chip Designs Using Voltage Islands," in *Proc. International Conference on Computer Aided Design*, November 2002, pp. 195 – 202.

[58] W.-P. Lee, H.-Y. Liu, and Y.-W. Chang, "Voltage Island Aware Floorplanning for Power and Timing Optimization," in *International Conference on Computer Aided Design*, November 2006, pp. 389–394.

[59] D. Li, P. Chuang, and M. Sachdev, "Comparative Analysis and Study of Metastability on High Performance fFlip-flops," in *11th International Symposium on Quality Electronic Design*, March 2010, pp. 853 –860.

[60] Y. Li, D. Parikh, Y. Zhang, K. Sankaranarayanan, M. Stan, and K. Skadron, "State-Preserving vs. Non-State-Preserving Leakage Control in Caches," in *Proceedings of the conference on Design, Automation and Test in Europe*, 2004, pp. 10022–10027.

[61] L. Lin and W. Burleson, "Analysis and Mitigation of Process Variation Impacts on Power-Attack Tolerance," in *Proceedings of the 46th Annual Design Automation Conference*, 2009, pp. 238–243.

[62] Y. Lin and L. He, "Leakage Efficient Chip-Level Dual-Vdd Assignment with Time Slack Allocation for FPGA Power Reduction," in *Proceedings of the 42nd Annual Design Automation Conference*, 2005, pp. 720–725.

[63] Y. Lin and L. He, "Dual-V_{dd} Interconnect with Chip-Level Time Slack Allocation for FPGA Power Reduction," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 10, pp. 2023–2034, October 2006.

[64] H.-Y. Liu, W.-P. Lee, and Y.-W. Chang, "A Provably Good Approximation Algorithm for Power Optimization Using Multiple Supply Voltages," in *Proceedings of the 44th Annual Design Automation Conference*, 2007, pp. 887–890.

[65] N. Magen, A. Kolodny, U. Weiser, and N. Shamir, "Interconnect-Power Dissipation in a Microprocessor," in *Proceedings of the International Workshop on System Level Interconnect Prediction*, 2004, pp. 7–13.

[66] H. Mahmoodi-Meimand and K. Roy, "Self-Precharging Flip-flop (SPFF): A New Level Converting Flip-flop," in *Proceedings of the 28th European Solid-State Circuits Conference*, September 2002, pp. 407 –410.

[67] T. Mahnke, S. Panenka, M. Embacher, W. Stechele, and W. Hoeld, "Efficiency of Dual Supply Voltage Logic Synthesis for Low Power in Consideration of Varying Delay Constraint Strictness," in *International Conference on Electronics, Circuits and Systems*, volume 2, 2002, pp. 701 – 704.

[68] D. G. Malcolm, J. H. Roseboom, C. E. Clark, and W. Fazar, "Application of a Technique for Research and Development Program Evaluation," *Operations Research*, vol. 7, no. 5, 1959.

[69] A. Milutinovic, A. Molnos, K. Goossens, and G. J. M. Smit, "Dynamic Voltage and Frequency Scaling and Adaptive Body Biasing for Active and Leakage Power Reduction in MPSoC: A Literature Overview," in *Proceedings of the Program for Research on Ingrated Systems and Circuits*, Technology Foundation / SAFE &ProRISC, November 2009, pp. 488–495.

[70] S. R. Nassif, "Modeling and Forecasting of Manufacturing Variations (Embedded Tutorial)," in *Proceedings of the Asia and South Pacific Design Automation Conference*, 2001, pp. 145–150.

[71] B. Nikolic, V. Oklobdzija, V. Stojanovic, W. Jia, J. K.-S. Chiu, and M. Ming-Tak Leung, "Improved Sense-amplifier-based Flip-flop: Design and Measurements," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 6, pp. 876 –884, June 2000.

[72] B. C. Paul, A. Agarwal, and K. Roy, "Low-Power Design Techniques for Scaled Technologies," *Integration, The VLSI Journal*, vol. 39, no. 2, pp. 64–89, 2006.

[73] M. Pedram and J. M. Rabaey, *Power Aware Design Methodologies*. Kluwer Academic Publishers, 2002.

[74] J. Rabaey, *Low Power Design Essentials*. Springer, 2009.

[75] J. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuit - A Design Perspective*. Pearson Education, 2003.

[76] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, "Leakage Current Mechanisms and Leakage Reduction Techniques in Deep-Submicrometer CMOS Circuits," *Proceedings of the IEEE*, vol. 91, no. 2, pp. 305–327, February 2003.

[77] K. Roy and S. Prasad, *Low-Power CMOS VLSI Circuit Design*. Wiley India Pvt. Ltd., 2009.

[78] K. Roy, L. Wei, and Z. Chen, "Multiple V_{dd} Multiple V_{th} CMOS (MVCMOS) for Low Power Applications," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, volume 1, July 1999, pp. 366–370.

[79] V. Sandararajan and K. Parhi, "Synthesis of Low Power CMOS VLSI Circuits Using Dual Supply Voltages," in *Proceedings of the 36th Design Automation Conference*, 1999, pp. 72–75.

[80] F. Sheikh, A. Kuehlmann, and K. Keutzer, "Minimum-Power Retiming for Dual-Supply CMOS Circuits," in *Proceedings of the 8th ACM/IEEE International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, 2002, pp. 43–49.

[81] K. Shi and D. Howard, "Sleep Transistor Design and Implementation - Simple Concepts Yet Challenges To Be Optimum," in *Proceedings of the International Symposium on VLSI Design, Automation and Test*, April 2006, pp. 1–4.

[82] D. Shin, J. Kim, and S. Lee, "Low-Energy Intra-Task Voltage Scheduling Using Static Timing Analysis," in *Proceedings of the 38th Annual Design Automation Conference*, 2001, pp. 438–443.

[83] A. Srivastava and D. Sylvester, "Minimizing Total Power by Simultaneous Vdd/Vth Assignment," in *Proceedings of the Asia and South Pacific Design Automation Conference*, 2003, pp. 400–403.

[84] A. Srivastava, D. Sylvester, and D. Blaauw, "Concurrent Sizing, V_{dd} and V_{th} Assignment for Low-Power Design," *Proceedings of the Design, Automation and Test in Europe Conference*, pp. 10718–10719, 2004.

[85] A. Srivastava, D. Sylvester, and D. Blaauw, "Concurrent Sizing, Vdd and Vth Assignment for Low Power Design," in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*, volume 1, February 2004, pp. 718–719.

[86] A. Srivastava, D. Sylvester, and D. Blaauw, "Power Minimization Using Simultaneous Gate Sizing, Dual-V_{dd} and Dual-V_{th} Assignment," in *Proceedings of the Design Automation Conference*, July 2004, pp. 783–787.

[87] V. Stojanovic, D. Markovic, B. Nikolic, M. A. Horowitz, and R. W. Brodersen, "Energy-Delay Tradeoffs in Combinational Logic Using Gate Sizing and Supply Voltage Optimization," in *Proceedings of the 28th European Solid-State Circuits Conference*, Septmeber 2002, pp. 211–214.

[88] V. Sundararajan and K. K. Parhi, "Synthesis of Low Power CMOS VLSI Circuits Using Dual Supply Voltages," in *Proceedings of the 36th Annual Design Automation Conference*, 1999, pp. 72–75.

[89] C. Tran, H. Kawaguchi, and T. Sakurai, "Low Power High Speed Level Shifter Design for Block Level Dynamic Voltage Scaling Environment," in *International Conference on Integrated Circuit Design and Technology*, May 2005, pp. 229–232.

[90] J. Tschanz, S. Narendra, Z. Chen, S. Borkar, M. Sachdev, and V. De, "Comparative Delay and Energy of Single Edge-triggered and Dual Edge-triggered Pulsed Flip-flops for High-performance Microprocessors," in *International Symposium on Low Power Electronics and Design*, 2001, pp. 147 –152.

[91] J. Tschanz, S. Narendra, Y. Ye, B. Bloechel, S. Borkar, and V. De, "Dynamic Sleep Transistor and Body Bias for Active Leakage Power Control of Microprocessors," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 11, pp. 1838–1845, November 2003.

[92] K. Usami and M. Horowitz, "Clustered Voltage Scaling Technique for Low-Power Design," in *Proceedings of the International Symposium on Low Power Design*, 1995, pp. 23–26.

[93] K. Usami and M. Igarashi, "Low-power Design Methodology and Applications Utilizing Dual Supply Voltages," *Proceedings of the Asia and South Pacific Design Automation Conference*, pp. 123–128, 2000.

[94] K. Usami, M. Igarashi, F. Minami, T. Ishikawa, M. Kanzawa, M. Ichida, and K. Nogami, "Automated Low-Power Technique Exploiting Multiple Supply Voltages Applied to a Media Processor," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 3, pp. 463–472, March 1998.

[95] Y. Wang, H. Mahmoodi, L.-Y. Chiou, H. Choo, J. Park, W. Jeong, and K. Roy, "Energy-efficient Hardware Architecture and VLSI Implementation of a Polyphase Channelizer with Applications to Subband Adaptive Filtering," *Journal of Signal Processing Systems*, vol. 58, pp. 125–137, 2010.

[96] N. H. Weste and D. Harris, *CMOS VLSI Design - A Circuits and Systems Perspective*. Pearson Education, 2005.

[97] S. Wooters, B. Calhoun, and T. Blalock, "An Energy Efficient Subthreshold Level Converter in 130nm CMOS," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 57, no. 4, pp. 290–294, April 2010.

[98] H. Wu, M. Wona, and I.-M. Liu, "Timing Constrained and Voltage Island Aware Voltage Assignment," in *Proc. Design Automation Conference*, 2006, pp. 429–432.

[99] Y. Ye, S. Borkar, and V. De, "A New Technique for Standby Leakage Reduction in High-Performance Circuits," in *Digest of Technical Papers, Symposium on VLSI Circuits*, June 1998, pp. 40–41.

[100] C. Yeh, M.-C. Chang, S.-C. Chang, and W.-B. Jone, "Gate-Level Design Exploiting Dual Supply Voltages for Power-Driven Applications," in *Proceedings of the Design Automation Conference*, 1999, pp. 68–71.

[101] Y.-J. Yeh and S.-Y. Kuo, "An Optimization-Based Low-Power Voltage Scaling Technique Using Multiple Supply Voltages," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, volume 5, 2001, pp. 535–538.

[102] Zhenyu, Y. Yang, J. Wang, R. P. Dick, and L. Shang, "TAPHS: Thermal Aware Unified Physical Level and High Level Synthesis," in *Proc. Asia and South Pacific Conference on Design Automation*, January 2006.