Circuit Level Techniques for Power and Reliability Optimization of CMOS Logic

A Dissertation
Presented to the
Academic Faculty

By

Abdulkadir Utku Diril

In Partial Fulfillment
Of the Requirements for the Degree
Doctor of Philosophy in
Electrical and Computer Engineering

Georgia Institute of Technology
May 2005

Circuit Level Techniques for Power and Reliability Optimization of CMOS Logic

Approved by:

Dr. Abhijit Chatterjee, Chair
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Vijay K. Madisetti
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Madhavan Swaminathan
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Adit D. Singh
Department of Electrical Engineering
Auburn University

Dr. Hsien-Hsin S. Lee
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: April 14, 2005

To

my wife Aytül, my parents Umut and Nuran Diril, and my sister Ayşegül

ACKNOWLEDGEMENTS

I am grateful to my research advisor Prof. Abhijit Chatterjee for his support and guidance. He has been a mentor to me all the time. I am also thankful to Prof. Adit Singh. The discussions we made had a profound effect on the direction of my research.

During the course of my stay at Georgia Institute of Technology, I had the chance to interact with many distinguished faculty members. I would like to thank Prof. Sean Lee for the insights he gave about the semiconductor industry. I would like to thank Prof. John Uyemura, Prof. Phillip Allen, Prof. Vincent J. Mooney, and Prof. Scott Wills for the experience they shared with me through classes and discussions.

The six months internship that I did at NVIDIA really speeded up my research. The skills that I developed while I was working there were very useful to me when I came back to complete my degree. I would like to thank David Dignam, Andrew Bell, Venkat Kommu, John Robinson, and Brian Hutsell at NVIDIA.

I would like to thank my beautiful wife Aytül for her support and love. Her love made me stronger. I would also like to thank my parents Nuran and Umut and my sister Ayşegül for always believing in me.

I want to thank Yuvraj S. Dhillon for being a great lab-mate and friend to me. Also I want to thank everyone in our lab for providing a nice environment.

I would like to thank Börteçene Terlemez for his support in almost every issue during my stay in Atlanta. I also would like to thank all the present and past members of the Atlanta Turkish Folk Dances Group and all of my friends including but not limited to Orkun Karadenizli, Doğa Özyürekli, Emre Ersin, Güçlü Onaran, Sermet Akbay, Oğuz

Ergin, Gökhan Mergen, Prof. Ahmet Erbil, Mazlum Koşma, Faik Başkaya, Özgür Çelebican, Engin Erdoğan, Hamza Kurt, Berk Gencülgen, Emrah Kotan, and İlkay Yavrucuk.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

CAD	Computer aided design
CED	Concurrent error detection
CMOS	Complementary metal oxide semiconductor
CVS	Clustered voltage scaling
DAG	Directed acyclic graph
DSM	Deep sub-micron
ECC	Error correcting code
FO4	Fanout of four
HDL	Hardware description language
MIM	Metal insulator metal
MLVS	Module level voltage scaling
MOSFET	Metal oxide semiconductor field effect transistor
NMOS	Negative channel metal oxide semiconductor
NPD	NMOS pull-up domino
PI	Primary input
PMOS	Positive channel metal oxide semiconductor
PO	Primary output
PPD	PMOS pull-up domino
RBB	Reverse body bias
SER	Soft error rate
SPICE	Simulation program with integrated circuit emphasis
SRAM	Static random access memory
STA	Static timing analysis
$\mathbf{V_{DD}}$	Supply voltage value
$\mathbf{V_{DDH}}$	High supply voltage value
$\mathbf{V_{DDL}}$	Low supply voltage value
$\mathbf{V_{Th}}$	Threshold voltage value

SUMMARY

Technology scaling trends lead to shrinking of the individual elements like transistors and wires in digital systems. The main driving force behind this is cutting the cost of the systems while the systems are filled with extra functionalities. This is the reason why a 3 GHz Intel processor now is priced less than what a 50MHz processor was priced 10 years ago. As in most cases, this comes with a price. This price is the complex design process and problems that stem from the reduction in physical dimensions.

As the transistors became smaller in size and the systems became faster, issues like power consumption, signal integrity, soft error tolerance, and testing became serious challenges. There is an increasing demand to put CAD tools in the design flow to address these issues at every step of the design process. First part of this research investigates circuit level techniques to reduce power consumption in digital systems. In second part, improving soft error tolerance of digital systems is considered as a trade off problem between power and reliability and a power-aware dynamic soft error tolerance control strategy is developed.

The objective of this research is to provide CAD tools and circuit design techniques to optimize power consumption and to increase soft error tolerance of digital circuits. Multiple supply and threshold voltages are used to reduce power consumption. Variable supply and threshold voltages are used together with variable capacitances to develop a dynamic soft error tolerance control scheme.

CHAPTER 1

INTRODUCTION

1.1 Motivation

Technology scaling brought new challenges for the designers in deep sub-micron (DSM) technologies. As transistors became smaller and more transistors are integrated into chips, issues like lowering power consumption, increasing soft error tolerance, and coping with the design complexity became serious challenges, especially for microprocessor manufacturers. New design approaches should be taken to address these problems. The focus of this research is to develop computer-aided design (CAD) tools for addressing some of the aforementioned design challenges facing the digital design community in the DSM era. Reducing power consumption and increasing soft error tolerance are the main subjects of interest.

Power consumption is one of the biggest bottlenecks for high-density chip designers. As the complexity and the transistor count increase, both dynamic and leakage power consumption became significant bottlenecks in the design process. Usage of multiple supply voltages in a digital circuit is studied in academia as an effective way of reducing dynamic power consumption resulting from the quadratic relation of supply voltage to dynamic power consumption [1]-[7]. Even though multiple power supplies can also reduce the leakage power consumption, the most effective approach for leakage power consumption reduction is using multiple threshold voltages in the circuit because of the exponential relation of threshold voltage to leakage power consumption [8]-[10].

Both approaches rely on the fact that there are many gates in a circuit that have enough slack to accommodate the increase in the gate delay resulting from a lower supply voltage and/or higher threshold voltage. Many commercial products are already employing gate-level multiple threshold voltages to decrease leakage power consumption. However, multiple supply voltage usage is still limited to applying different voltages to large portions of the chip. This is partly due to the lack of efficient gate-level CAD tools for design, optimization, verification, and layout of multiple supply voltage circuits, and partly to the overhead of generation and routing of a second supply voltage. However, it can be expected to see more chip producers employ this technique as the research on this subject matures.

Decrease in soft error tolerance in digital systems is another negative side effect of technology scaling. Soft error tolerance of digital circuits decreases significantly because of the reduction in transistor sizes. Reduction in circuit dimensions reduces the capacitance of the circuit nodes, thereby leading to an increase in the voltage magnitude of the glitches caused by a noise source such as α-particles or cosmic rays [11]. Improving the sequential elements has been the main approach taken for increasing the soft error tolerance of the chips. However, as the circuits' operating frequencies increase and the logic depths in pipeline stages decrease, soft errors seen in combinational circuits will increase and become a significant portion of the total soft errors [12]. Currently, a notebook computer with 256 MB memory which is operated in an airplane at 35000 feet altitude has a failure in every 5 hours because of the particle strikes [13]. With technology scaling, the soft errors in combinational logic will be even worse than this. For systems with variable environmental conditions, designing for the worst case of

particle flux will result in significant energy and/or area, and/or delay overhead. For such systems, adaptability will be an important design parameter. An ideal system should be able to reduce the mentioned overheads when it is not exposed to the worst case environmental conditions.

The challenges introduced by technology scaling trends to designing low power and reliable systems are the primary motivation for this work.

1.2 Thesis Organization

In this work, the problem of static dual supply voltage assignment to combinational circuits and the problem of dynamic soft error tolerance of combinational circuits are addressed. Chapter 2 discusses the effects of technology scaling on power consumption and soft error tolerance of digital systems. It is shown that both issues are posing important challenges to the design community in the deep sub-micron (DSM) era. Chapter 3 explains the circuit modeling approach used in the simulation steps of the developed optimizations. After giving standard analytical models for the energy consumption and delay of a single gate, the SPICE look-up table based approach is explained. The static timing analysis approach used to find the delay of the whole circuit using the single gate characteristics is also explained.

Chapter 4 explains the power consumption reduction techniques developed during this research. First, a review of the power reduction techniques is given. The second part of this chapter explains the low power pseudo dual supply voltage design scheme which is developed for domino logic circuits. This method reduces the power consumption of the domino logic gates using a “dual supply voltage assignment” like design approach, but without using a second supply voltage. The last part of this chapter explains the low

power CMOS circuit design technique using dual supply voltages. A novel level shifting logic gate structure is explained. Then, an algorithm for dual supply voltage assignment using the developed level shifting logic gates is given. The improvement obtained by using the developed level shifting logic gates instead of the standard level shifters is also discussed.

Chapter 5 explains the dynamic soft error tolerance control scheme that is developed for circuits that operate under changing environmental conditions. The chapter starts with the detailed examination of a single gate associated with particle strikes. This examination leads to a soft error tolerance optimization method, which can significantly improve the tolerance of a circuit when the environmental particle flux level is increased.

Finally, in Chapter 6, a brief summary of the research is presented with suggestions on future work. The conclusion also includes the discussions on the major contributions of this work.

CHAPTER 2

EFFECTS OF TECHNOLOGY SCALING TRENDS TO POWER CONSUMPTION AND SOFT ERROR TOLERANCE

2.1 Power Consumption Trends in Microprocessors

Scaling of transistors in CMOS technology is the underlying factor for the improvement of digital systems in terms of speed of execution and complexity of tasks the systems are capable of running. The industry is following closely, if not exactly, the trends anticipated by Gordon Moore [14]. Mainly, technology scaling has three goals for every new technology generation [15]:

- Reduce gate delay by 30% (increase the operating frequency by 43%).
- Double transistor density.
- Reduce energy per transistor by about 65%, power consumption by about 50%.

These goals are closely related to each other and they are the results of scaling transistor dimensions. In theory, a new technology generation with transistor width, length, and oxide thickness scaled down by 30% will accomplish all of these three goals. Real data for past Intel microprocessors show that the frequency doubled with each new technology generation, more than the anticipated 43% [15]. This is mainly due to the increased pipeline depths of the microprocessors, leading to less number of gate delays per clock period. The clock period for current Intel processors is close to 10 gate delays, which is almost equal to the optimal logic depth per pipeline stage that is given as six to eight FO4 (fan-out 4) inverter delays in [16]. It can be assumed that the increase in

frequency for future technology generations will be less than two because there is not enough room for designers to scale up frequency by increasing pipeline depth any further. Figure 1 shows a plot of processor frequencies and gate delays per pipeline stage for past Intel processors.

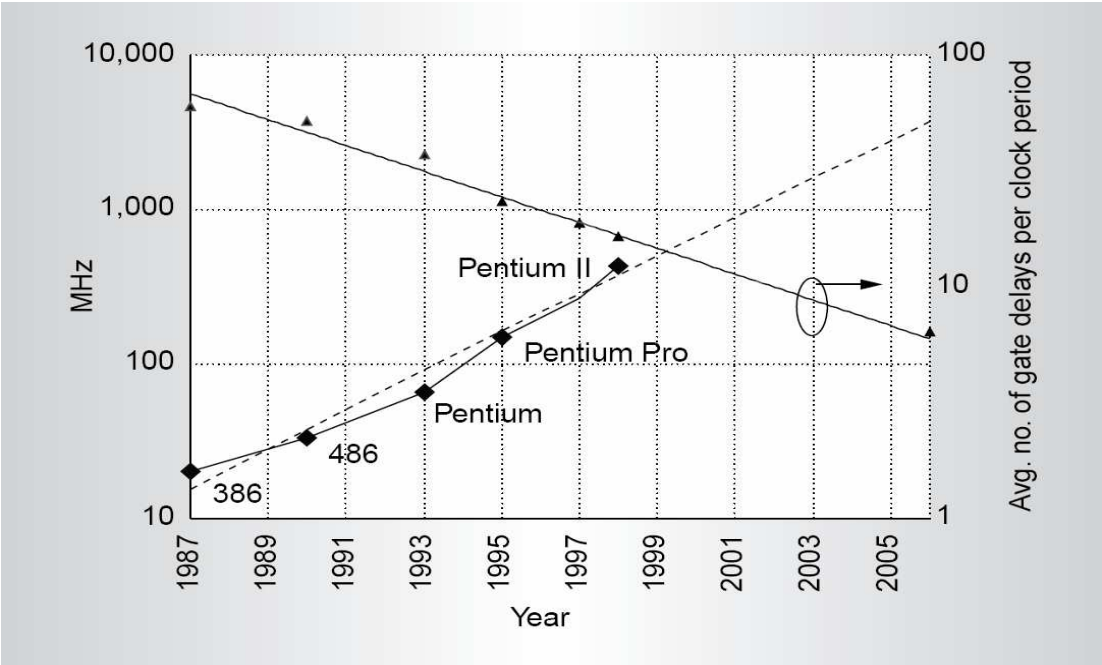

Figure 1. Processor frequency and average number of gate delays per clock period for various Intel processors.

Transistor density, however, increased less than the anticipated rate of 50% per technology generation. Even though the goal is met when an existing processor is shrunk using the next process technology, a new processor implemented in the same new process technology shows a drop in density [15]. This may be due to the increased complexity of the new microarchitecture. Figure 2 shows a plot of transistor density for past Intel processors.

Scaling was done keeping the voltage constant until reaching 0.8μ feature size. This approach keeps the dynamic power consumption resulting from

charging/discharging of capacitances per transistor the same for different technologies, as seen in Equation 1.

$$Power \propto 0.7 \cdot C \cdot V_{DD}^2 \cdot \frac{1}{0.7} \cdot f = C \cdot V_{DD}^2 \cdot f \quad (1)$$

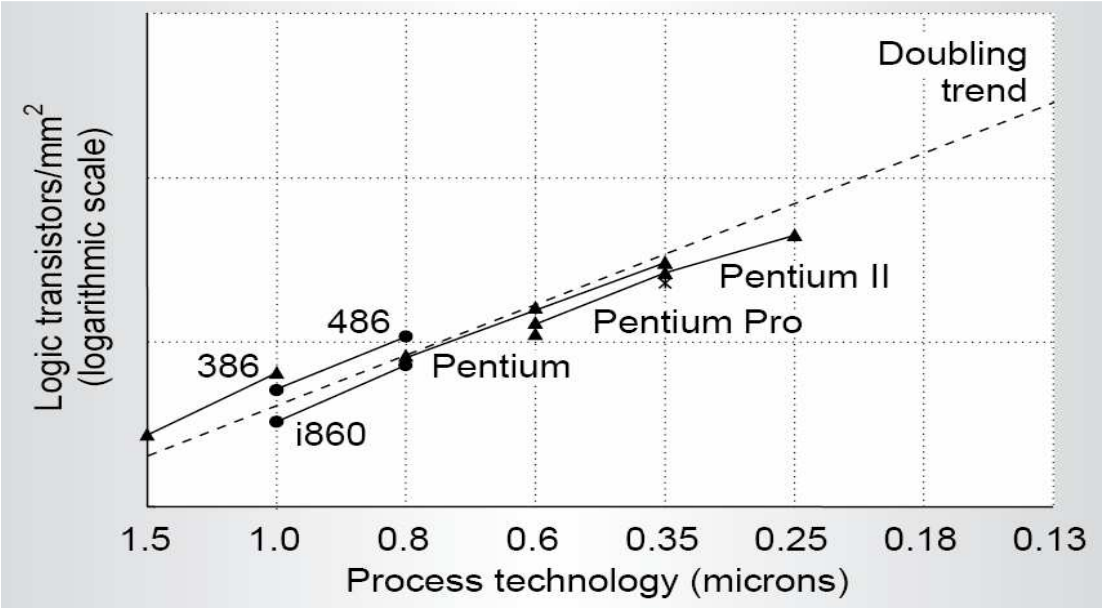

Figure 2. Transistor density for various Intel processors.

This approach led to a dramatic increase in power consumption as the transistor count and design complexity (and therefore switching activity) increased. For that reason, after 0.8µ technology, constant electric field scaling instead of constant voltage scaling was employed. In constant electric field scaling, supply voltage is scaled down by the same amount as the feature size. So, for a 0.7 scaling factor, this approach leads to a ~50% reduction in power consumption per transistor, as seen in Equation 2.

$$Power \propto 0.7 \cdot C \cdot 0.7^2 \cdot V_{DD}^2 \cdot \frac{1}{0.7} \cdot f = 0.49 \cdot C \cdot V_{DD}^2 \cdot f \quad (2)$$

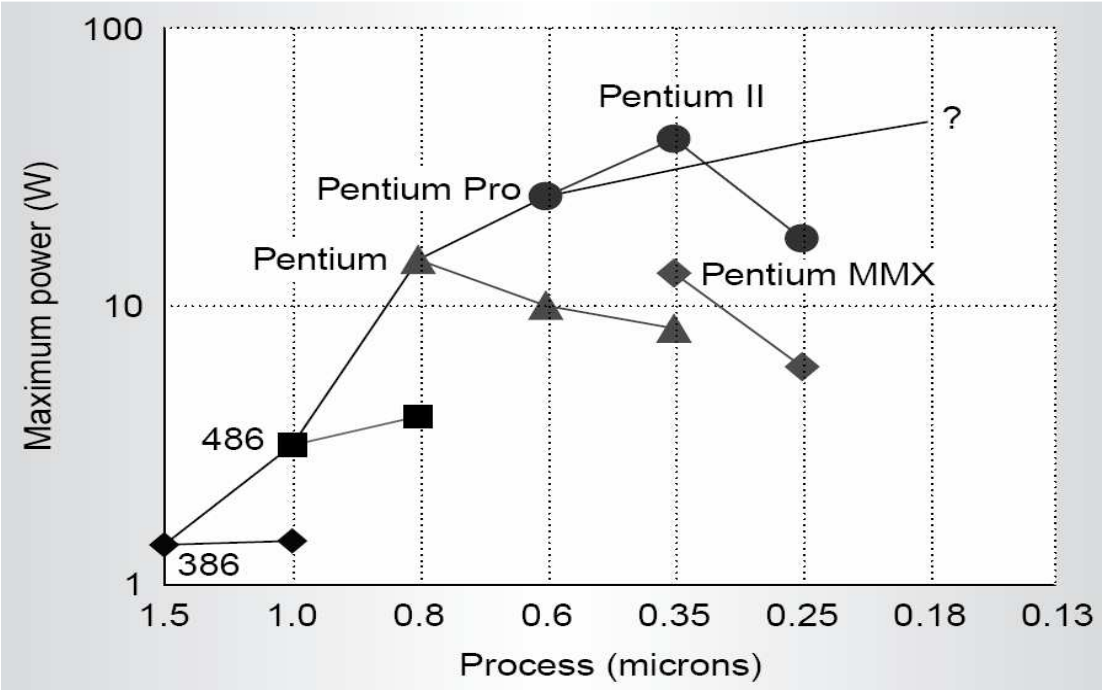

Figure 3. Maximum thermal power dissipation for various Intel processors.

Figure 3 shows the maximum thermal power dissipation for various Intel processors. Different markers represent different processors. As seen from the figure, the rate of increase in power consumption (in logarithmic scale) dropped after constant electric field scaling was applied. Also seen from the figure is the trend that processors ported to new technologies after 0.8μ show a reduction in maximum power consumption.

Even though the logarithmic rate of increase in power consumption decreased after constant electric field scaling, power consumption increase is still an alarming issue for designers. As the transistor count, design complexity, and operating frequency increase, power consumption continues to increase dramatically, which adversely affects reliability, cooling costs, and - especially for portable systems - battery life [17].

Even though reduced power supply voltages decreased dynamic power consumption per transistor, the trend of reducing power supply voltage led to a

significant increase in the leakage power consumption because of the necessity to reduce threshold voltages in order to compensate the drive loss caused by the reduced supply voltage. As a common practice, threshold voltage for a process is usually chosen to be smaller than one quarter of the supply voltage value to ensure that performance does not suffer excessively [18]. This approach, combined with the exponential relationship of leakage current to threshold voltage, led to a significant increase in the percentage of leakage power consumption in total system power consumption. Figure 4 shows this trend. If this trend continues, the leakage power consumption will be equal to the dynamic power consumption in a couple of technology generations. Since the leakage energy is in a sense "wasted" energy, half of the energy dissipation will be waste in the future technologies if significant improvements in device, circuit, architecture, and software are not introduced.

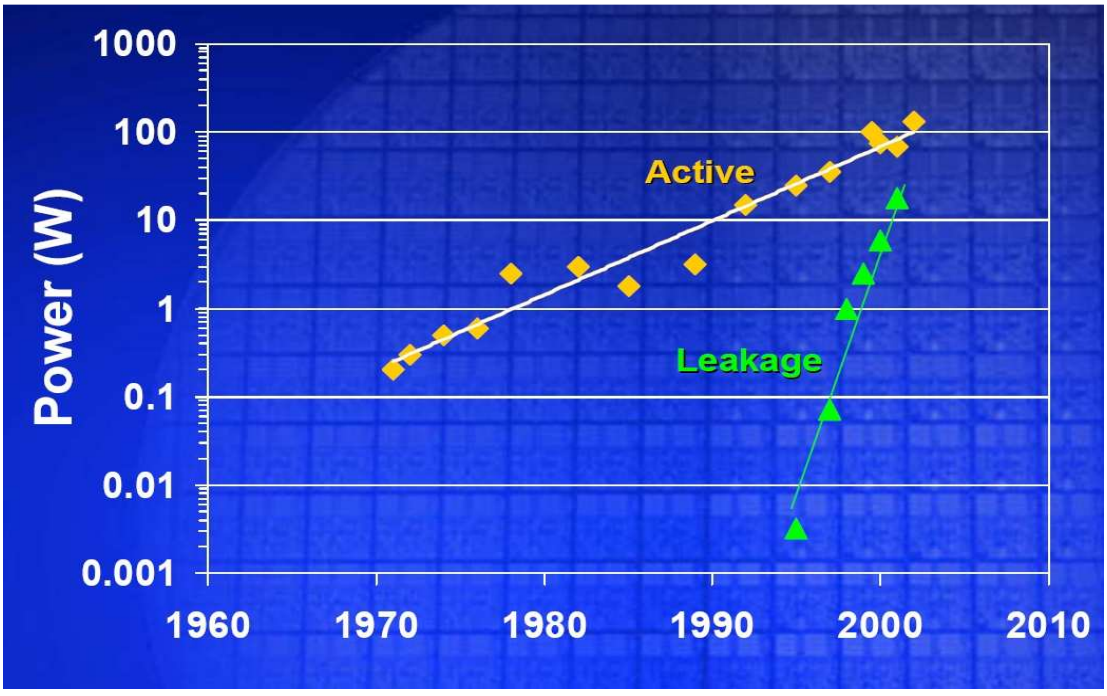

Figure 4. The change in active (dynamic) and leakage power consumptions for the past Intel processors.

Reducing leakage and dynamic power consumption constitute one of the biggest design challenges for chip makers today. Therefore, great emphasis is given to low power research both in academia and in industry.

2.2 Effects of Scaling on Soft Error Rates in Microprocessors

Soft errors (SEs) are becoming an increasing concern in electronic devices as the operating voltages and device sizes decrease. A reduction in circuit dimensions reduces the capacitance of the circuit nodes, thereby leading to an increase in the voltage magnitude of the glitches caused by a noise source such as α-particles or cosmic rays [11]. It also increases the susceptibility of circuits to hard-to-verify design/layout errors such as those that result from increased crosstalk and increased susceptibility to power supply and ground bounce. Low noise margins resulting from reduced operating voltages further aggravate the problem by allowing even small glitches to propagate.

The main contribution to the soft error rate (SER) comes from the memory cells and sequential circuits in current microprocessors. A soft error in these circuits may result in a bit flip in the saved state, which may lead to wrong execution. The SER for memories decreased dramatically after the introduction of error-correcting codes (ECC) [12][19].

The error rate in sequential circuits remains the same, while the error rate in combinational circuits increases linearly with increasing operating frequencies [20]. It is shown in [12] that this trend will lead to comparable SERs in combinational and sequential circuits for the 70 nm technology generation, as seen in Figure 5, where expected SERs are given for SRAM (without ECC), latches, and combinational logic. SER values are given for different pipeline stage delays (in terms of Fan-out four inverter

delay) for combinational logic and latches. A single SER curve is given for SRAM because the change in SER for SRAM is negligible for changing pipeline stage delay. The FIT term in Figure 5 denotes *failures in time*. An FIT of 1 means a failure is observed every 10^9 hours. The chip that is used to get the data for Figure 5 is the Alpha 21264 microprocessor. It has 15.2 million transistors on the die in 350 nm technology. Number of transistors is assumed to be doubling in every new technology generation. Combinational logic occupies 20% of the die area. The rest is occupied by memory and sequential elements. For systems in which ECC is employed for memories, the SER of combinational logic might start dominating the chip SER sooner than 70 nm technology.

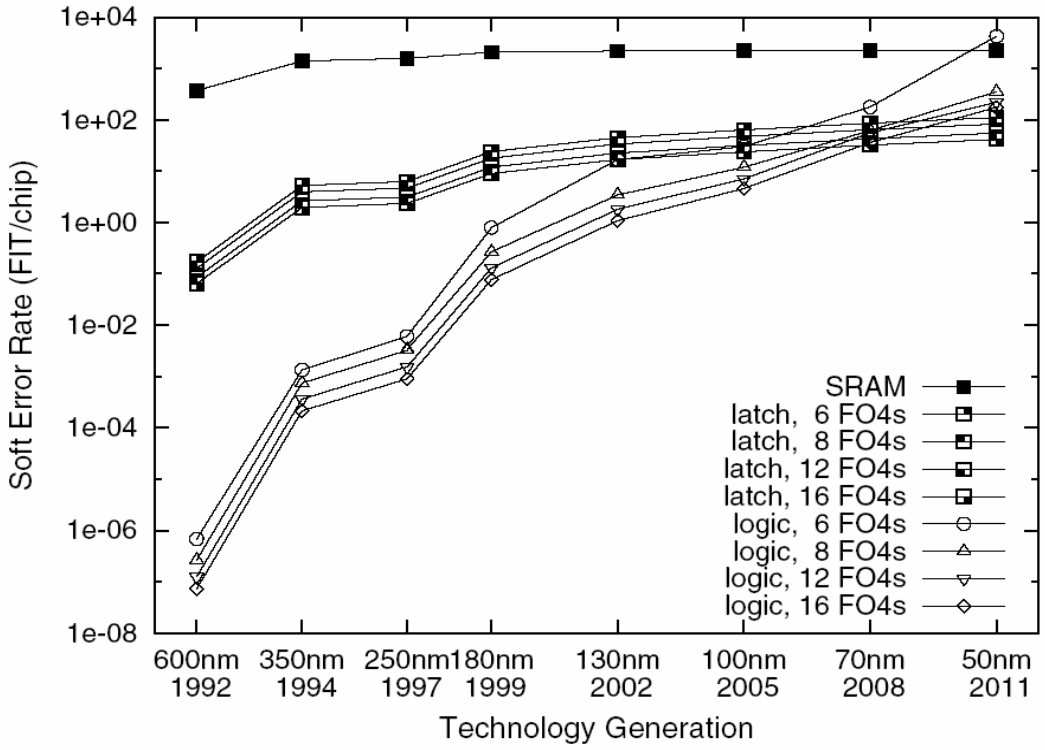

Figure 5. SER/chip for SRAM/latches/logic [12].

CHAPTER 3

MODELING CMOS GATES

This chapter focuses on basic characteristics of CMOS gates. Certain parameters of CMOS gates such as propagation delay, dynamic energy consumption, static energy consumption, and input capacitance are needed to be modeled with respect to design variables so that the effects of the optimizations can be validated by simulations. Design variables for a single gate are size, output capacitance, input signal ramp, supply voltage, and threshold voltage.

3.1 Operation of a CMOS Gate

To understand the basic operation of a CMOS gate, let's examine the simplest CMOS gate, the inverter. An inverter, which consists of a PMOS and an NMOS transistor, generates the logical inverse of the input signal at its output. Figure 6 shows the circuit for CMOS inverter.

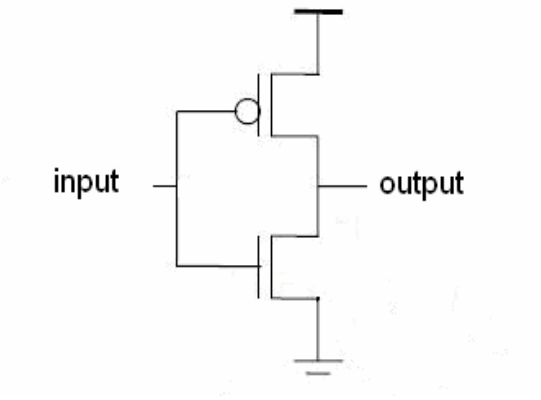

Figure 6. CMOS Inverter circuit.

The circuit operates as follows: When the input is high, the PMOS transistor is turned off and the NMOS transistor is turned on. If the output voltage is already low, circuit remains as it is. If the output voltage is greater than zero (i.e. output capacitance has charge), the charge in the output capacitance is drained to the ground through the NMOS transistor. The speed of the discharge depends on the resistance of the NMOS transistor and the initial total charge on the output capacitance. When the input is low, the opposite happens and the output capacitance is charged through the PMOS transistor. At the high level, if the delay of the gate is taken as the average of charging and discharging times, it is seen that the delay of a gate depends on the size of the transistors (influences the resistance on which the charging/discharging happens and contributes to the output capacitance), output capacitance (influences the amount of charge to be charged/discharged), threshold voltages of the transistors (influences the resistance of the transistors), and supply voltage applied to the gate (influences the resistance of the transistors and determines the amount of charge the output capacitance holds). As a first order approximation, if we model the transistors as resistances, delay of a gate can be written as $K \cdot \tau$ where K is a constant and τ is the time constant of the RC network, which is given as follows:

$$\tau = R \cdot C = \frac{V_{DD}}{k \cdot \left(\frac{W}{L}\right) \cdot (V_{DD} - V_{Th})^{\alpha}} \cdot C \tag{3}$$

where k is a process dependent constant, V_{DD} is the supply voltage value, V_{Th} is the threshold voltage value, W/L is the aspect ratio of the transistors, and α is the velocity saturation coefficient [21]. C is the total output capacitance including both the load capacitance and the parasitic capacitance of the gate. It is observed that the ramp of the

input signal also affects the delay of the gate. If a slowly changing signal is applied to the gate, the output changes slowly as well.

As in the case for delay, energy consumption of a gate also is influenced by the supply and threshold voltage values, output capacitance, transistor sizes, and input signal ramp. Two types of energy dissipation is observed during the operation of a CMOS gate. These are dynamic energy consumption and static energy consumption. Dynamic energy consumption is the energy consumed during transition of the input and output nodes. Main portion of the dynamic energy consumed is because of the charging/discharging of the capacitances. In the case of an inverter, the output capacitance is charged to hold $C \cdot V$ amount of charge when the input is at low voltage. When the input switches to high voltage, this charge is drained to the ground through the NMOS transistor and this leads to a total energy loss of $C \cdot V^2$ for a charge-discharge cycle. Half of this energy is dissipated during charging of the capacitor on the PMOS transistor, the other half is stored in the capacitor. When the capacitor is discharging, the stored half is dissipated on the NMOS transistor. In addition to the charging/discharging energy dissipation, there is also short-circuit energy dissipation, which is regarded as a part of the dynamic energy dissipation since it is observed when the input to the gate is switching. A short circuit current flows from the supply voltage to the ground during input switching when both the transistors are on. Short circuit energy consumption (E_{sc}) is about 10% of the dynamic energy consumption [18][22]. For high V_{Th} circuits ($V_{Th} > ¼ V_{DD}$), E_{sc} is negligible [22]. Equation 4 shows the dynamic energy consumption for a CMOS gate:

$$E_{dynamic} = 0.5 \cdot C \cdot V_{DD}^2 \cdot S + E_{sc} \quad (4)$$

where C is the total capacitance, V_{DD} is the supply voltage value, and S is the total number of switching observed at the gate's output during the time period of interest.

Static energy consumption is the energy loss in the gate when the gate is not operational. It is composed of three components, as seen in Figure 7: subthreshold leakage, drain junction leakage, and gate leakage. Subthreshold leakage is the dominant factor for today's technologies. Equation 5 gives the equation for energy dissipation resulting from subthreshold leakage:

$$E_{sub-vth} = V_{DD} \cdot I_S \cdot e^{\frac{V_{GS} - V_{Th}}{n \cdot v_T}} \cdot \left(1 - e^{\frac{V_{DS}}{v_T}} \right) \cdot T \qquad (5)$$

where I_S is a circuit and process dependent constant, n is the subthreshold swing coefficient, v_T is the thermal voltage ($k \cdot T/q$), V_{Th} is the threshold voltage, V_{DD} is the supply voltage, and T is the duration of idleness. Subthreshold leakage is likely to increase in the future because of the exponential relationship of subthreshold current to threshold voltage. Subthreshold current increases about 10 times for every 0.1V reduction in threshold voltage.

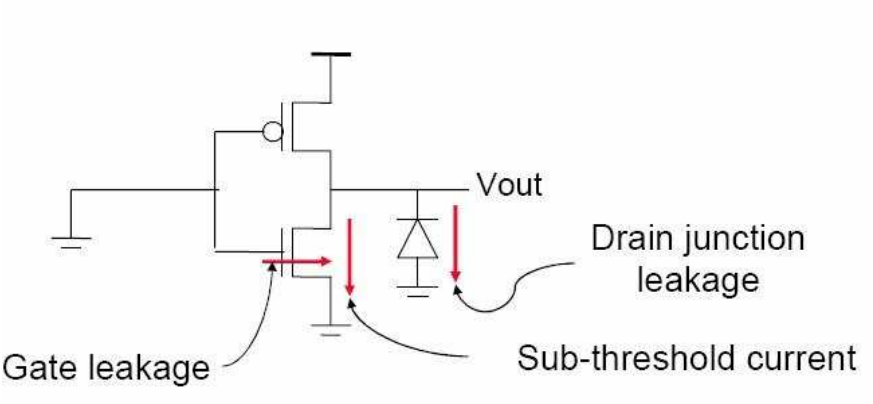

Figure 7. The Components of static energy dissipation.

Gate oxide leakage is becoming a significant contributor to the total static energy consumption because of supply voltages being scaled less than the amount necessary for maintaining a constant electric field across the gates of the transistors. Supply voltage scales down by only 15% per generation (not 30% as dictated by constant field scaling predictions) in order to sustain high transistor performance [23]. This leads to an increase in the electric field across the gate dielectric per technology generation [24], leading to an increase in the gate oxide leakage. As the physical gate oxide thickness approaches sub 10 Angstroms, gate oxide leakage becomes larger than $100A/cm^2$ because of direct band-to-band tunneling. Gate oxide leakage increases weakly with temperature, but it increases exponentially with an increase in supply voltage at a rate of two times larger leakage for every 100 mV increase in supply voltage. Since the supply voltage is not scaled down as much as the feature sizes, as gate oxide thickness decreases, gate oxide leakage becomes more important. New high-K dielectrics as gate dielectric material are needed to reduce gate oxide leakage energy.

3.2 SPICE Modeling of CMOS Gates

In order to optimize circuits for power consumption and soft error tolerance, the ability to simulate circuits fast and accurately is needed. Since the techniques used in the optimizations modify one or more of the design parameters (V_{DD}, V_{Th}, size, output capacitance) for the gates in the circuit, the effects of these modifications to the overall circuit delay and energy consumption should be considered during the optimizations. The fastest way is to use equations similar to the ones given in Section 3.1 [25]-[27]. Even though this will result in faster simulation of the circuit, the results will not be accurate enough. The main reason for the inaccuracy is the lack of compact analytical models that

include input signal ramp as a variable. Even when the input signal ramp is considered as a variable [21][28], these models mostly fall short of modeling the output signal ramp in a way that can be used to make calculations for a logic network. Also, the models in the literature are mostly generated for single supply voltage circuits. Therefore, they lose accuracy when a gate is driven by a voltage source that is different from the supply voltage of that gate. As the analytical models are made more accurate, the models become more complex and the advantage of using them diminishes.

Using SPICE simulations will give the best accuracy for approximating delay and energy consumption of the circuit. However, simulating complete circuits consisting of hundreds of gates with SPICE will be very time consuming. It is impossible to use SPICE simulations in an iterative kind of optimization because of the long simulation times.

To get both accuracy and speed, circuit simulations are done using SPICE look-up tables for individual gates. Gates are simulated using SPICE for various values of supply voltage, threshold voltage, gate size, input signal ramp, output capacitance, and input signal magnitudes. For gates with multiple inputs, only one input is changed and the others are applied the sensitizing logic values during simulations. The following parameters are extracted from SPICE simulations:

- Propagation delay: It is calculated to be the average of propagation delays for rising and falling outputs.
- Dynamic energy: The energy dissipated during output is rising and falling is calculated by multiplying the supply voltage value by the total charge drawn from the supply during rising output and falling output. These values are denoted by E^{01} and E^{10}, respectively.

- Static power: The power consumed when the gate's output is stable at logic one and logic zero is calculated by multiplying the supply voltage value by the stable current value. These values are denoted by P^1 and P^0, respectively.
- Input capacitance: The input capacitance (C_{in}) of a gate was measured by applying a voltage pulse of amplitude V with rise and fall time T to the gate and measuring the average current flowing into (I_{in}) or out of (I_{out}) the gate. C_{in} is then taken as the average of $\frac{I_{in} \cdot T}{V}$ and $\frac{I_{out} \cdot T}{V}$.
- Output ramp: Output ramp is taken to be the average of the slopes of rising and falling output signals.

Average energy dissipation of a gate per clock cycle is calculated by the following equation:

$$E = T_{clock} \cdot \left(prob^0 \cdot P^0 + prob^1 \cdot P^1 \right) + Activity \cdot \left(\frac{E^{01} + E^{10}}{2} \right) \tag{6}$$

where $prob^0$ and $prob^1$ are the static probabilities of the output of the gate being 0 and 1 respectively and Activity is the switching activity at the gate output.

Modeling domino logic gates is slightly different from the static CMOS gates. The difference comes from the fact that domino logic belongs to dynamic logic family. A domino logic gate has two phases of operation. In the precharge phase, the output node of the gate is charged to logic one. In the evaluate phase, the output node either remains at high voltage value or it is discharged to zero depending on the inputs. Because of this precharge-evaluate style of operation, the calculation of energy consumption and delay differs for domino logic gates compared to static logic gates, as follows:

- Delay: In the evaluate phase, the output of a domino gate either remains high, or it is discharged to zero through the NMOS network. The delay of a domino gate is the propagation delay when the output is falling. Instead of input signal ramps (as in the case of static CMOS gates), the driving strength of the predecessor is used to index the look-up tables. As will be explained in Section 4.2.1, two different structures of gates having different driving strengths are considered in optimizations.
- Energy consumption: Because every evaluate phase is followed by a precharge phase, the output node either does a one to zero transition followed by a zero to one transition (if the inputs are such that the output evaluates to zero) or no transition at all (if the inputs are such that the output evaluates to one). Therefore, two values are extracted from the SPICE simulations to calculate the energy consumption (dynamic + static) of the gate. These are E^{010} and E^{00}, where the superscript numbers represent the transitions observed at the output node. Average energy consumption of a domino gate per clock cycle is then calculated by the following equation:

$$\overline{E} = prob^0 \cdot E^{00} + prob^1 \cdot E^{010} \tag{7}$$

where $prob^0$ and $prob^1$ are the static probabilities of the output of the gate being evaluated to 0 and 1 respectively.

The SPICE loop-up tables are used during the circuit optimization to find the circuits' energy consumption and delay. Gates in a circuit are represented as nodes in a directed acyclic graph (DAG) and the wires are represented as edges. The DAG is then topologically sorted such that in the topologically sorted list L, for any gate G_i, all the

nodes that are driven by G_i are located after G_i in the list. As the list is traversed in order, topologically sorted list structure guarantees that when a node is visited, all of its predecessors are visited. This simplifies the delay calculation for the circuit. Topological sorting is explained in Appendix A.

3.2.1 Calculating the Delay of a Circuit

Delay of the circuit is calculated using static timing analysis (STA) [29][30][31]. To simplify the analysis, two dummy nodes are inserted to the graph, a dummy primary input node that is driving all the real primary inputs and a dummy primary output node that has all the primary outputs as inputs. Both of these nodes have zero delay. To perform STA, some parameters are defined for each node [32].

Early start time: The early start time for a node is based on the logical requirement that a node can begin only when all its predecessors have been completed.

Early Start Time = *Latest*(*Finish Time* for all predecessors)

When all predecessors are assumed to finish as early as possible then:

Early Start Time = *Latest*(*Early Finish Time* for all predecessors)

Early Start Time = *Max*(*Early Finish Time* for all predecessors)

Early finish time: The early finish time is the early start time plus the delay for the node.

Latest finish time: The latest finish time for a node is based on the logical requirement that a node must end before any of its successors may be begin.

Latest Finish Time = *Earliest*(*Start Time* for all successors)

When all successors are assumed to start as late as possible then:

Latest Finish Time = *Earliest*(*Latest Start Time* for all successors)

Latest Finish Time = *Min*(*Latest Start Time* for all successors)

Latest start time: The latest start time is simply the latest finish time less the delay for the node.

Slack: The slack time is the difference between late and early start times.

STA is performed as follows:

- Circuit is topologically sorted from primary inputs to primary outputs.
- Nodes are visited one by one starting from the dummy primary input node. This node has an early start time and early finish time of zero. For all the other nodes, early start time is calculated to be the maximum of early finish times of its predecessor. Early finish time is simply the early start time plus the delay for that node. When delay for that node is calculated, the output signal ramp of the node with the maximum early finish time is used as the input signal ramp for the node.
- When all the nodes are visited, the early finish time of the dummy primary output is equal to the circuit's delay.
- To find the slacks of all the nodes, nodes are visited in the reverse order, starting from the dummy primary output node.
- The latest start and latest finish times of the dummy primary output node are assigned to be equal to the circuit delay.
- For the rest of the nodes, latest finish time is calculated to be the minimum of latest start times of its successors. Latest start time is simply the latest finish time minus the delay for that node. Slack is the difference between late and early start times.

3.2.2 Calculating the Energy Consumption of a Circuit

Energy consumption of the circuit is obtained by adding the individual energy consumption values of all the gates using equations 6 and 7. The static probabilities and switching activities of the internal nodes are obtained for specific values of input switching activities and static probabilities using Synopsys Design Analyzer tool. Dynamic energy and static power values are obtained from the SPICE look-up tables. Only the energy spent to charge and discharge the output capacitance of the node is considered when the energy consumption is calculated for that node. The charging and discharging of the node's input is automatically taken care of when the energy consumption of its predecessor is calculated since the input capacitances of the nodes serve as output capacitances to their predecessors.

Static power consumption of a gate rather than the static energy consumption is found using SPICE. Then, depending on the circuit that this gate is used in, this value is multiplied by the clock period to obtain the static energy consumption for the gate in a clock cycle. The static power for a gate obtained by SPICE simulations is a single value, which incorporates all types of static power that is consumed in the gate, including subthreshold power and gate leakage power.

CHAPTER 4

LOW POWER DUAL SUPPLY VOLTAGE CMOS DESIGN

Power consumption increase in digital systems is one of the main bottlenecks for chip designers. Improvement to power consumption of digital systems in every level of design is needed to combat this problem. Even though software and architectural level techniques have the most dramatic effect on power consumption reduction for a specific system in consideration, circuit level techniques are very important since they provide general methods which can be applied to a spectrum of different architectures and systems. Multiple supply voltage usage and multiple threshold voltage usage in dynamic circuits are effective methods to reduce dynamic and static power consumption. These methods are general methods which can be applied to various kinds of digital systems. However, there are practical limitations for these methods especially for the usage of multiple supply voltages. In CMOS circuits, for example, the need for additional level shifting circuitry when a high voltage gate is driven by a low voltage gate is one of the most important limitations. Domino logic, however, does not require level shifting due to the lack of the PMOS tree in the domino gates. Unfortunately, the necessity to generate and route the additional supply voltages remains.

A key goal of this part of the research is to provide techniques to deal with the practical limitations of multiple supply voltage usage such as overhead of level shifters. The issues addressed in this chapter are as follows:

- Developing circuit design techniques for domino logic circuits to exploit the benefits of dual supply voltage usage without the need for a second supply voltage, to obtain pseudo dual supply voltage operation.
- Developing methods to overcome the limitations caused by the necessity of level shifter usage in CMOS circuits to increase dynamic energy consumption reduction by dual supply voltage usage.

4.1 Review of Power Reduction Techniques

4.1.1 Dynamic Power Reduction Techniques

Equation 4 in Section 3.1 gives the formula for dynamic energy consumption. As seen from the equation, dynamic energy consumption of a circuit can be reduced by decreasing switched capacitance (C x S) or decreasing supply voltage (V_{DD}).

4.1.1.1 Decreasing Switched Capacitance

The most obvious method for dynamic power reduction is to eliminate unnecessary switching activity in the circuit. Clock gating is the most commonly used technique for this purpose. By gating the clock to specific flip-flops, the activity in unused modules is eliminated. Clock gating can be applied in both fine grain and coarse grain. Some commercial tools [33][34] can automatically apply fine grain clock gating to the design where the tool finds some predetermined constructs in the system description (verilog HDL or VHDL description), but coarse grain application of clock gating needs the input from the designer. Clock gating is a circuit-level technique and it is usually applied close to the end of the design process, so the benefits obtainable by this method are limited. Techniques applied at higher levels of abstraction such as architectural changes and/or

algorithmic changes have more potential to decrease the unnecessary switching in the circuit. Therefore, power consumption reduction must be an important goal for the designers in every level of the design.

4.1.1.2 Decreasing Supply Voltage

Supply voltage is the most influential parameter for power reduction because of the quadratic relationship of power consumption to the supply voltage. This relationship is exploited by the designers in many ways. Dynamic supply voltage scaling and multiple supply voltage usage are the most popular methods targeting the V_{DD}^2 term in Equation 4.

If a circuit has delay constraints varying in time, dynamic voltage and frequency scaling is used to put the circuit in a high delay/low power mode when the constraint is relaxed and in a high power/low delay mode when the constraint is tight. There may be several other intermediate modes between these two ends. Today, most of the portable processors come with this capability such as Intel's XScale [35], AMD's K6-IIIE [36], and Transmeta's Crusoe [37]. In most of the current processors with dynamic voltage/frequency scaling, the control is done by the operating system, running software, or the users themselves. Thermal conditions of the processor or the remaining battery level can also force a change in the mode of operation. Once the control signal is generated to change the operating voltage, the voltage source is given the necessary inputs to change the voltage and the PLLs are reprogrammed to the predetermined frequency values for that operating voltage. An overhead in terms of power consumption and delay is involved with every mode change of the processor. As an example, AMD's PowerNow technology takes 200 μsec for stabilizing supply voltage and PLL frequency [36]. Apart from software-controlled voltage scaling, there are preliminary studies for

hardware controlled dynamic voltage scaling as well. RAZOR system [38] tunes the supply voltage by monitoring the error rate during circuit operation, thereby eliminating the need for voltage margins and exploiting the data dependence of the circuit delay. The feedback loop is controlled by the number of errors resulting from timing violations caused by the slowdown in the circuit. When the error rate is above a predetermined threshold, the operating voltage is increased and the voltage is reduced if the error rate is below the threshold. Up to 64.2% energy savings for a full custom multiplier and a SPICE-level Kogge-Stone adder was achieved with a 3% delay overhead.

Static multiple voltage assignment is also used to reduce power consumption. Given a timing constraint, some parts of the hardware are assigned to run with the regular supply voltage, while other parts are assigned to run with a lower supply voltage, reducing power consumption. The granularity of voltage assignment and the method for assigning the voltages vary in the literature. But all of them assign lower supply voltage to the portions of the circuit with enough slack as a gate's delay increases with decreasing supply voltage, as seen from Equation 3.

There are some important issues to deal with if dual/multiple supply voltages are to be used in a circuit. Gate-level dual supply voltage usage in CMOS circuits may suffer from excessive leakage power if low voltage gates directly drive high voltage gates. In these situations, the PMOS transistor in the high voltage gate is not turned off completely with the low voltage "logic high" input signal. This leads to the use of level shifters wherever low voltage gates drive high voltage gates. To reduce or eliminate the delay, area, and energy overhead of the level shifters, clustered voltage scaling (CVS) [39][40] and Module Level Voltage Scaling (MLVS) [6][41] were proposed. In CVS, low voltage

clusters are constructed in the circuit in such a way that there is no low voltage gate driving a high voltage gate and level shifting is only done in sequential elements. This is done by assigning low supply voltage to the gates starting from the circuit outputs depending on their slacks. MLVS assigns the dual supply voltages to relatively large partitions of the circuit. This reduces the number of level shifters needed. Both methods limit the obtainable power savings by introducing constraints to the low voltage assignment process.

There has also been research in the use of gate-level dual supply voltages where level shifters are being used whenever a low voltage gate drives a high voltage gate [42]-[44]. In [42] and [43], graph-theoretic algorithms are employed to apply dual supply voltages at the gate level while keeping the delay constant. The technique in [44] is an extension to CVS, where level shifters are not restricted to be only in sequential elements. The energy consumption overhead of the level shifters is found to be 8% in [44]. The high cost of level shifters in terms of delay and energy consumption reduces the achievable energy consumption reductions by these techniques.

4.1.2 Static Power Reduction Techniques

Static power consumption per gate can be reduced by increasing the threshold voltage (V_{Th}) of the transistors or by using transistor stacks in the pull-down and/or pull-up networks.

4.1.2.1 Increasing Threshold Voltage

Because of the exponential relationship of static power consumption to the threshold voltage, increasing the threshold voltage is an effective way of decreasing the static

power consumption. Dynamic threshold voltage adjustment and static assignment of multiple threshold voltages are methods used to decrease static power consumption.

Similar to dynamic supply voltage scaling, dynamic threshold voltage adjustment can be used for systems with performance requirements varying in time. Delay of the circuit increases and static energy consumption decreases as the threshold voltage value is increased, as seen in Equations 3 and 5. Reverse body bias (RBB) can be applied to the transistors to increase their threshold voltages when the system is idle or performance requirements are reduced [24][45]. In the sub-100nm technology generation, approximately a 2-3X reduction in leakage is achievable by the RBB technique [24]. However, effectiveness of RBB decreases as channel lengths become smaller or V_{Th} values are lowered. The V_{Th} modulation capability of RBB weakens as short-channel effects become worse or body effect diminishes because of lower channel doping [24].

A dynamic threshold voltage adjustment scheme for SRAMs is given in [46]. Using the temporal and spatial locality of cache access, the threshold voltages of active lines are lowered to meet performance requirements, while the threshold voltages of the inactive cache lines are increased by controlling the back bias, reducing static energy consumption. If a line is not accessed for a predetermined time (30-100 μsec), the threshold voltage for that line is increased. The threshold voltage is decreased back after that line is accessed. For an area overhead of 15% and a performance overhead of 1%, static power dissipation is reduced by 72% in an SRAM of size 64 KB.

Static assignment of multiple threshold voltages can save up to 80% [47] of the static power dissipation in CMOS circuits without decreasing the performance of the circuit. Low threshold voltages are assigned to the gates on the critical paths to keep the

delay unchanged, while high threshold voltages are assigned to the gates on the off-critical paths to reduce the static power dissipation. The optimum high threshold voltage is different for different circuits but is usually more than 100 mV higher than the low threshold voltage value, leading to a >10X decrease in static power consumption in the high threshold gates. Multiple threshold voltage implementation is relatively easy compared to multiple supply voltage implementations. An additional mask is added to the fabrication process to achieve a second threshold voltage. Layouts of the gates need not be changed, which simplifies the design process.

4.1.2.2 Using Transistor Stacks

The leakage of a two-transistor stack is an order of magnitude less than the leakage in a single transistor [48]. Therefore, the static current through a gate depends on the inputs to that gate. This makes the total leakage current of a circuit dependent on the states of the primary inputs [49]. The optimal input vector for a circuit for minimum static power dissipation may be determined and applied to the circuit when the circuit is idle to decrease the static power dissipation.

An additional transistor may also be used to force the stacking effect when the circuit is in standby mode. For the gates with high subthreshold leakage in non-critical paths, a leakage control transistor can be inserted in series and can be turned off during the standby mode [49]. This technique can effectively reduce the leakage current using single threshold voltage.

4.2 Level-Shifter Free Design of Dual Supply Digital Circuits

4.2.1 Pseudo Dual Supply Voltage Domino Logic Design

Domino logic is used extensively in high-speed circuit design. The main reason for the higher performance of domino logic compared to static CMOS is the reduced input capacitance seen by driver gates in domino logic. In CMOS, both PMOS and NMOS transistors are driven at the input stage whereas in domino logic, only NMOS transistors are driven. The higher performance of domino logic comes at the expense of a higher power consumption. The switching activity in domino circuits is, on the average, double that in CMOS circuits. Higher switching activity together with the clock power dissipation leads to a higher power consumption in domino gates compared to CMOS gates.

Due to the lack of PMOS network, gate-level multiple supply voltage assignment can be done to domino circuits without the need for level shifters [50][51]. However, the overhead of generating and routing an additional power supply voltage remains. Shieh et. al. [50] use dual supply voltages, gate sizing, and a contention-alleviated static keeper (CASK) to reduce power consumption in domino circuits while keeping the delay fixed. This approach needs two separate supply voltages for the gates and a bias voltage for the CASK circuitry, which is used to speed up V_{DDL} to V_{DDH} interfaces. Jung et. al. [51] use dual supply voltages and dual threshold voltages together with a low voltage swing clock in order to reduce power consumption of domino circuits. A separate back-biasing voltage is also used to turn off the pull-up PMOS completely when low voltage swing clock is applied to a gate with high supply voltage. Both these approaches suffer from the complexity of generating and routing additional voltages therefore it is not feasible to

implement these approaches to a circuit which is fabricated without a second supply voltage.

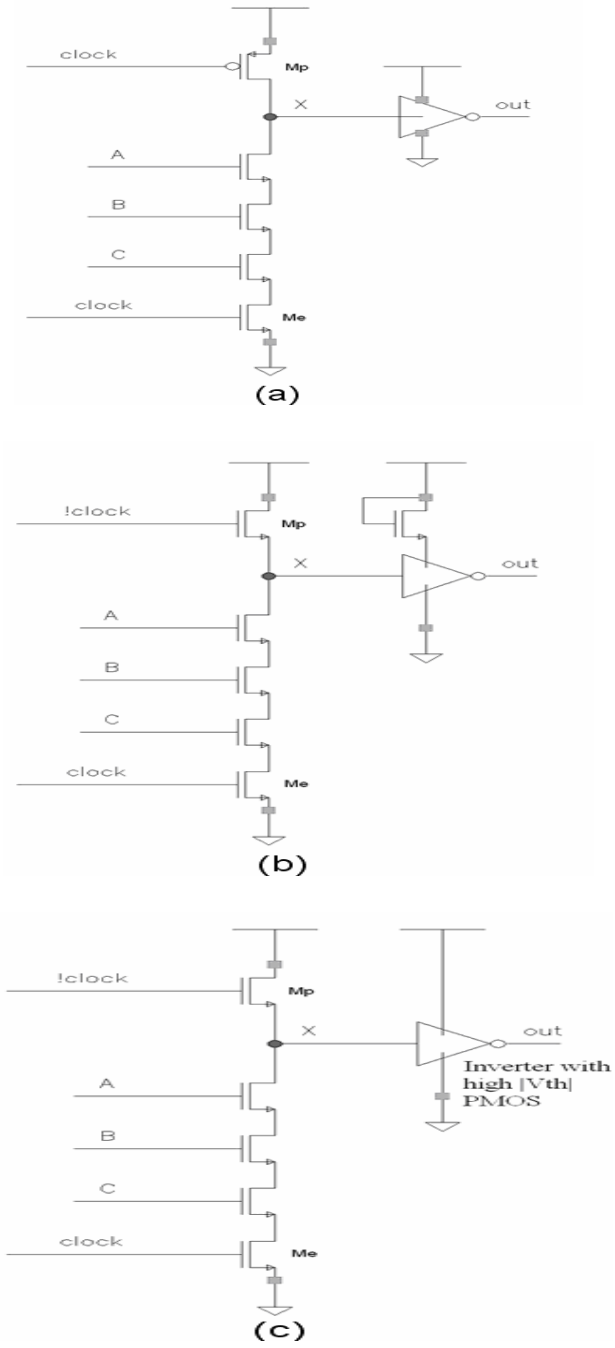

Figure 8. Domino Logic 3-input AND gate
(a) PPD (PMOS pull-up Domino), (b) NPD1 (NMOS Pull-up Domino with extra NMOS between source of PMOS in inverter and power supply), (c) NPD2 (NMOS Pull-up Domino with high |Vth| PMOS in the inverter).

A pseudo dual supply voltage assignment scheme without the need for an additional power supply to lower the power consumption of combinational domino logic blocks while maintaining the performance is developed [52][53]. The basic idea is to replace the standard domino gates [referred to as PPD (PMOS Pull-up Domino) gates from now on] on the off-critical paths by low-power (but higher delay) domino gates. The low-power domino gates use a novel technique to effectively operate at a lower supply voltage. The PMOS pull-up transistor in Figure 8a is replaced by an NMOS transistor (Figures 8b and 8c), which leads to a reduced voltage swing at the input of the output inverter. This node (Node X in Figure 8) has a high capacitance in domino logic gates to eliminate problems resulting from charge sharing. So, reducing the voltage applied to this node reduces the energy consumption when that node is charged and discharged. If the energy consumed for one charging-discharging of the node capacitance at Node X is assumed to be $K \cdot V_{DD}^2$ for a regular domino gate, it is $K \cdot V_{DD} \cdot V_{swing}$ for an NMOS Pull-up domino gate, where V_{swing} is $V_{DD} - V_{Thn} - \Delta V_{Thn} < V_{DD}$ because of the limitation of the NMOS transistor when passing a "high" voltage. The ΔV_{Thn} term is the threshold voltage modification seen on the NMOS pull-up transistor due to the body-bias effect. Note that an inverted clock signal is applied as the precharge signal to the NMOS pull-up transistor in NPD gates.

However, reducing the voltage at the input of the CMOS inverter in the domino gate leads to high static current in the inverter due to the non-zero voltage between PMOS gate and source when the inverter input is high. Two ways to reduce this static current are developed. One way is to use an always on NMOS, whose gate is tied to power supply, between power supply and the source of the PMOS in the inverter. This

brings the source voltage of the PMOS in the inverter down because of the NMOS pass transistor, making the PMOS gate to source voltage close to zero. This type of NMOS Pull-up Domino gates will be referred to as NPD1 from now on. The second way is using a high threshold PMOS in the inverter. This way, the current flow through the PMOS transistor when its gate to source voltage is not zero is reduced, reducing the static current flow when input of the inverter is high. This type of NMOS Pull-up Domino gates will be referred to as NPD2 from now on. Both NPD1 and NPD2 types of gates have lower energy consumption but higher delay compared to PPD type gates.

4.2.1.1 Design Strategy with NPD1 Type Gates

The primary design parameter is the size of the diode connected NMOS transistor when NPD1 type domino gates are used in a circuit. Figures 9, 10 and 11 show the SPICE simulation results of the variation of the delays and energy consumption values respectively for a 3 fan-in, 1 fan-out AND NPD1 gate with different values of NMOS transistor width and with different drivers (NPD1 and PPD). The delay and energy of the corresponding PPD gate are also shown. As seen from the plots, increased size improves both the energy consumption and delay of an NPD1 gate. The price is the increased area. Also seen from the plots is the effect of the driving gate on the delay and energy consumption characteristics of the driven gate. A gate operates faster if it is driven by a PPD gate because a PPD gate provides a higher voltage value at its output which reduces the resistive effects of the NMOS transistors in the driven gate compared to the NMOS transistor driven by an NPD1 gate. The output of an NPD1 gate swings from zero to $V_{DD} - V_{Thn} - \Delta V_{Thn}$ leading to reduced driving strength. The simulations are done using Level 49 SPICE models for 0.18μ technology generation [54] using HSPICE.

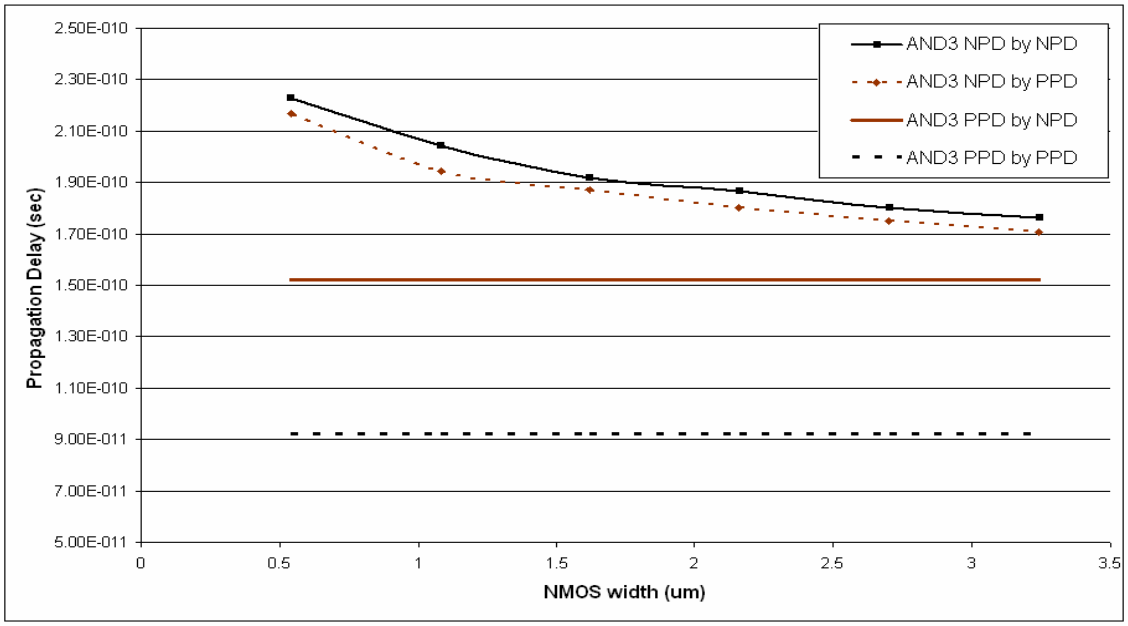

Figure 9. Variation of propagation delay for NPD1 AND gate with width of extra NMOS transistor.

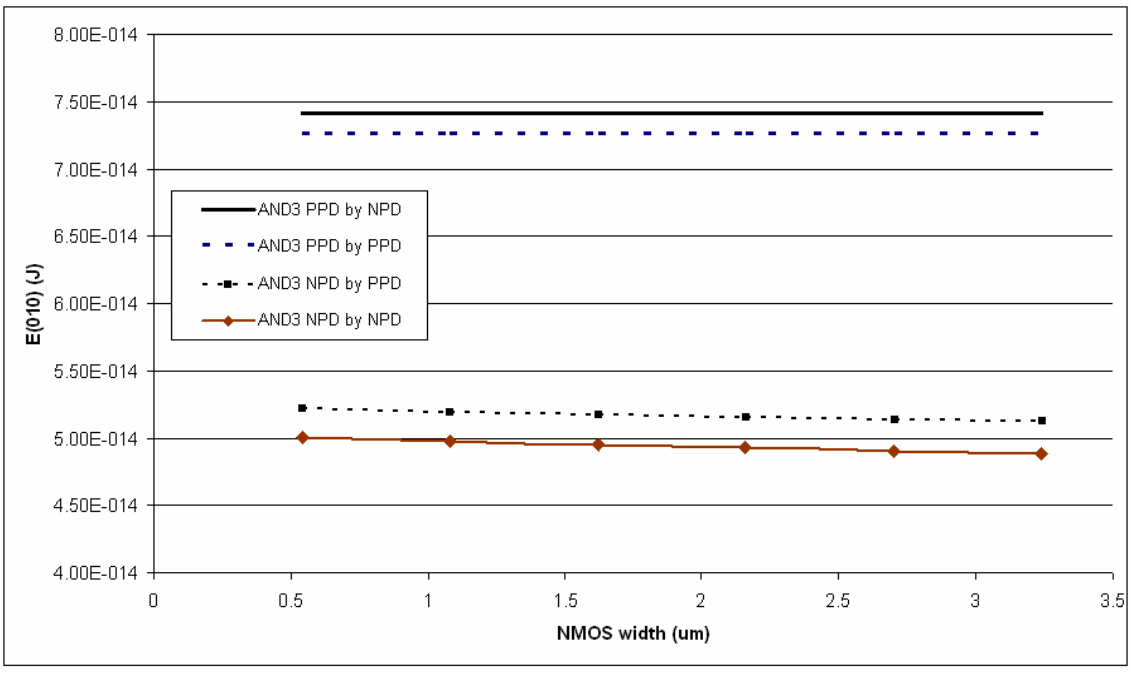

Figure 10. Variation of E^{010} for NPD1 AND gate with width of extra NMOS transistor.

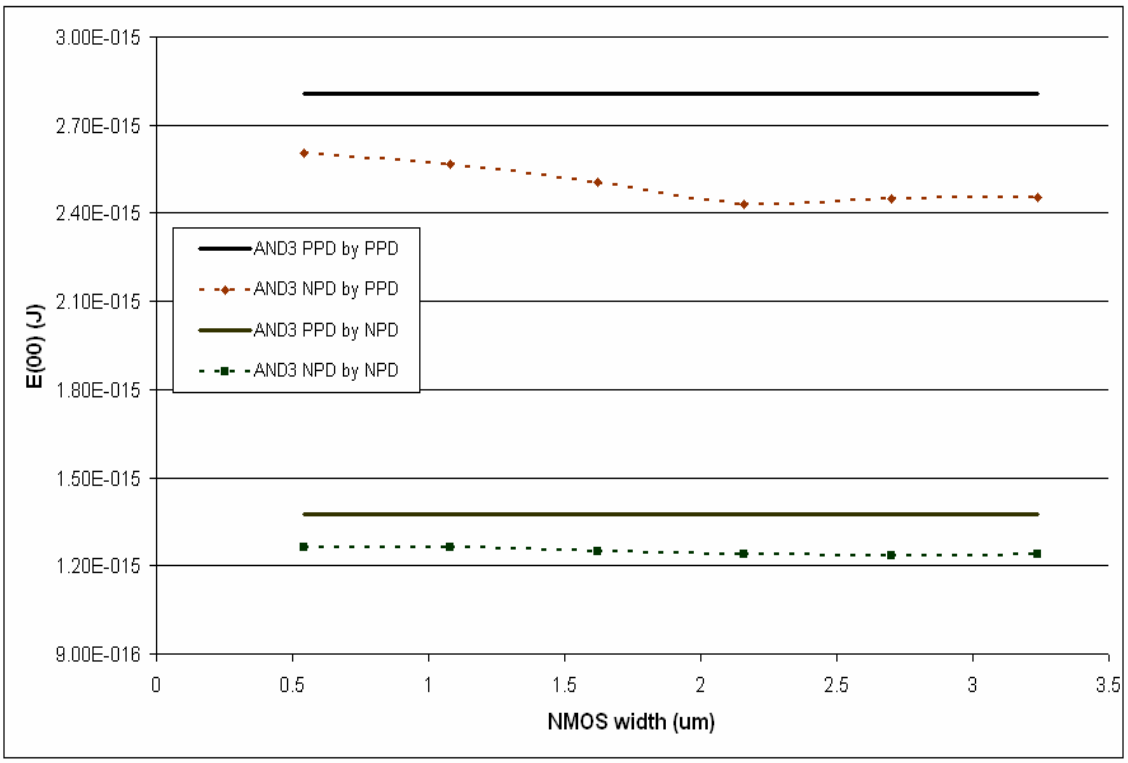

Figure 11. Variation of E^{00} for NPD1 AND gate with width of extra NMOS transistor.

4.2.1.2 Design Strategy with NPD2 Type Gates

The primary design parameter is the threshold voltage value to be used in the inverter PMOS when NPD2 type domino gates are used in a circuit. If the magnitude of V_{thp} is small, the leakage current is large and if the magnitude of V_{thp} is large, gate delay becomes large. Figures 12, 13, and 14 show SPICE simulation results of the variation of delay and energy consumption values respectively for a 3 input AND NPD2 gate with different values of threshold voltage of inverter PMOS transistor and with different drivers (NPD2 and PPD). The simulations are done using Level 49 SPICE models for 0.18µ technology generation [54] using HSPICE. The delay and energy consumption of the corresponding PPD gate are also shown. The supply voltage is 1.8V and the load capacitance is 3fF for all the cases. The regular threshold voltage value is 0.4 V. The

trends seen in the plots are similar for the other gate types and loads, but the intersection points of the curves vary. Note that for V_{thp} values of ~0.6V to ~0.75V, the NPD2 AND3 gate is both faster and lower energy than the PPD AND3 for a load of 3fF. This threshold voltage range for which an NPD gate is better than its PPD counterpart differs for different gate types and different loads (for example, delay of an NPD gate increases more rapidly with increasing load capacitance compared to a PPD gate). Therefore, replacing all the PPD gates in a circuit by NPD gates with a specific value of V_{thp} will not, in general, make the entire circuit both faster and lower energy. When choosing the second threshold voltage, the circuit should be considered as a whole. As it will be shown in the results, choice of the second threshold voltage significantly effects the achieved energy consumption reduction.

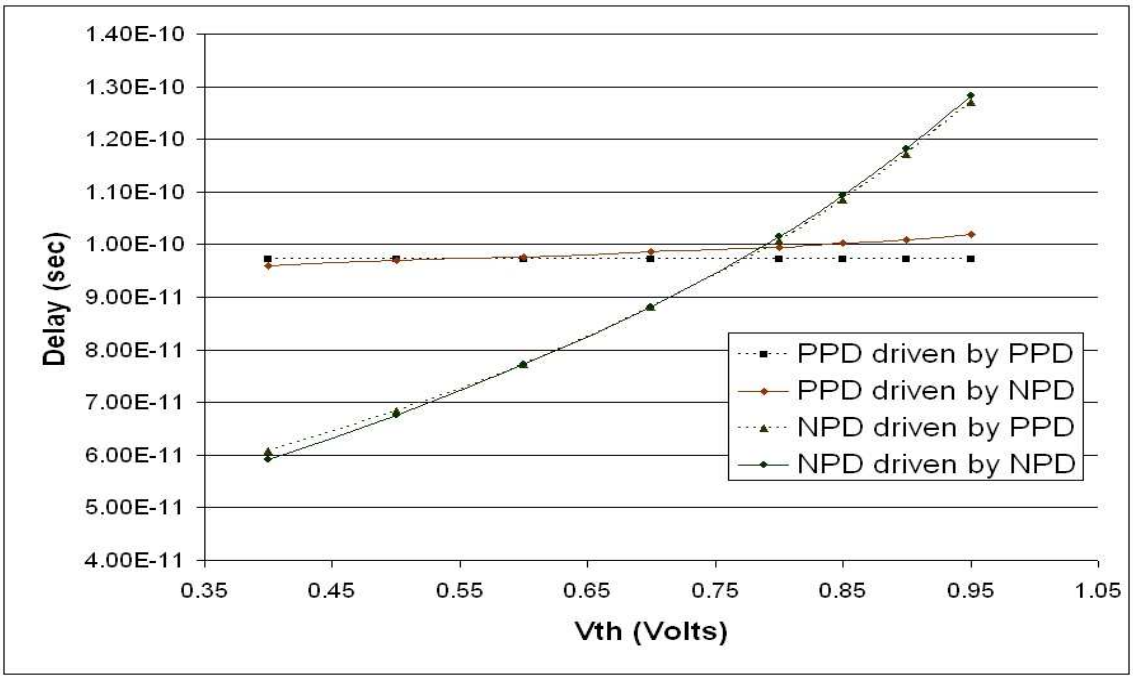

Figure 12. Variation of propagation delay for AND3 gates with threshold voltage of inverter PMOS transistor.

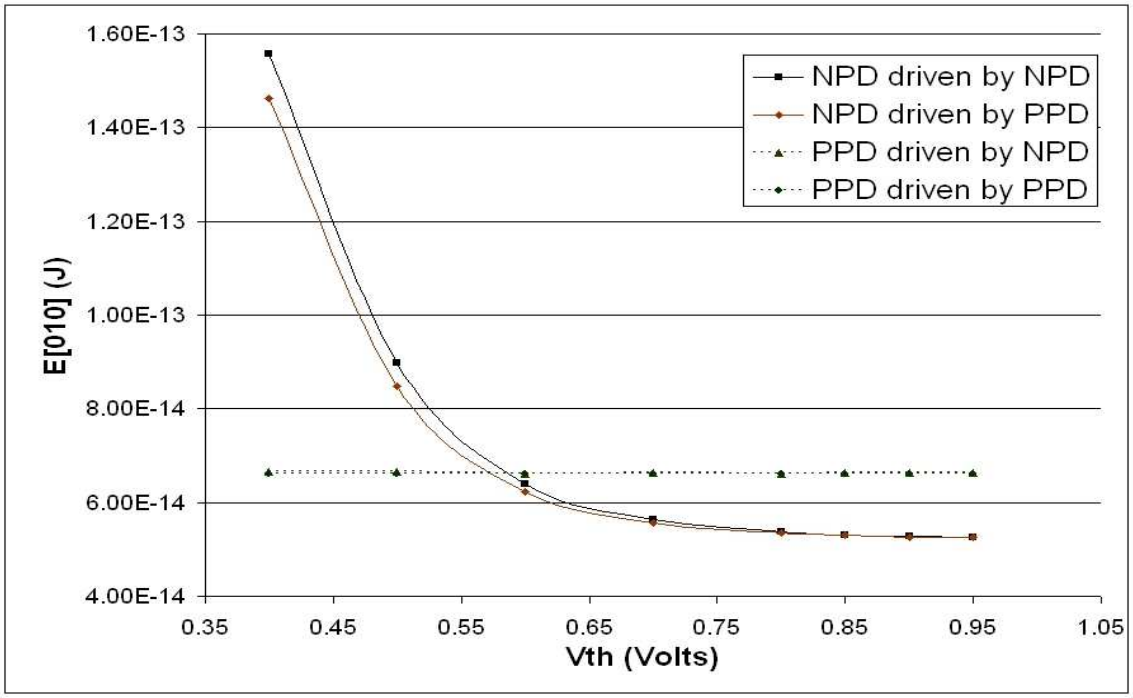

Figure 13. Variation of E^{010} for AND3 gates with threshold voltage of inverter PMOS transistor.

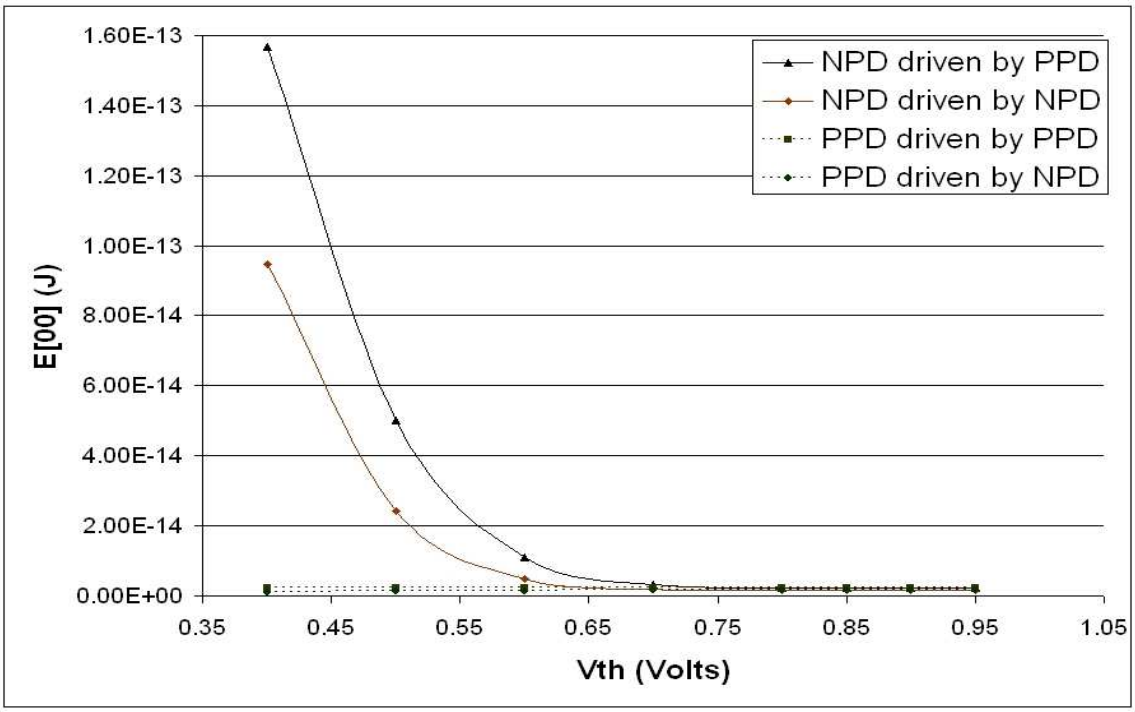

Figure 14. Variation of E^{00} for AND3 gates with threshold voltage of inverter PMOS transistor.

4.2.1.3 Algorithm for Power Optimization using NPD Gates

To reduce the energy consumption of domino logic circuits, PPD gates on the non-critical paths are replaced with slower, lower energy NPD gates. The total delay of the circuit remains the same as the original circuit which has only PPD gates. The algorithm explained in this section applies to both PPD-NPD1 and PPD-NPD2 replacement.

The circuit is represented as a directed acyclic graph (DAG), G(V,E). If the circuit has multiple primary inputs (PIs), a dummy PI vertex, PI_d, is created which fans-out to the original PIs. Underlying this is the assumption that all inputs arrive simultaneously. Similarly for POs, a dummy PO vertex, PO_d, is created which has fan-ins from the original POs. Each vertex 'v' of the DAG has associated with it the following information:

1. The logic function computed by the gate corresponding to the vertex.
2. Structure of the gate (NPD or PPD).
3. The current delay (v.delay), energy (v.energy), time slack (v.ts), early start time (v.es), early finish time (v.ef), latest start time (v.ls) and latest finish time (v.lf) of the vertex at any stage of the replacement process. The delay and energy consumption values for PPD and NPD gates of different fan-ins, fan-outs, gate types, and sizes when driven by a PPD gate or an NPD gate are obtained from SPICE simulations to form look-up tables. These look-up tables are used to obtain delay and energy consumption values of the gates. These values are updated for a gate whenever the gate or its driver gates change from type PPD to NPD. The dummy vertices have zero delay and energy consumption.

Algorithm PPD-NPD replacement
Inputs: Topologically sorted list of circuit vertices, V^s;
Output: Circuit with off-critical path PPD gates replaced with NPD gates.

```
for as many times as number of vertices {
    Let v be the vertex with maximum metric value.
    Map v to be of structure NPD.
    low_energy←0
    for every predecessor p of v {
        Depending on whether p is mapped to PPD or NPD, look-up "pin_energy" as the
        energy for this gate when it is driven by p, using SPICE look-up tables and
        switching activities.
        low_energy← low_energy+pin_energy
    }
    high_delay←delay of v when it driven by the predecessor p with maximum early
    finish time. Found using SPICE look-up tables.
    low_energy←low_energy/(No. of predecessors p of v)
    if((high_delay-v.delay)≤v.ts) { // Check if PPD-NPD replacement violated time
                                    // slack of v.
        flag← 0
        for every successor p of v {
            Compute largest delay of p with v mapped to a NPD gate. Let this be
            "p_delay".
            if((v.es+high_delay)≥p.es) { // Check if replacing v with its NPD equivalent
                                         // violates time slack of any of its successors.
                if((v.es+high_delay+p_delay)>p.lf) flag← 1,break.
            }
            else if((p_delay-p.delay)>p.ts) flag← 1,break.
        }
        if(flag=0) {
            Map v to a NPD gate.
            v.delay← high_delay
            v.energy← low_energy
            v.metric←-2
            Update_Time_Slacks(V^s)
        }
        else v.metric←-1
        Map v back to PPD
    }
    else v.metric←-1
    Map v back to PPD
}
```

Figure 15. Algorithm for PPD-NPD replacement.

The algorithm for replacing PPD gates with NPD equivalent gates consists of two steps:

1. Initialization: In this step, the DAG is topologically sorted to get the sorted vertex list, V^s. V^s is used to compute the total circuit delay, T, of the baseline circuit in which each vertex is mapped to a PPD gate. In this step, the initial total energy of the baseline circuit with all PPD gates is also computed. Delay and energy consumption calculations are done as explained in Section 3.2.
2. PPD to NPD replacement: First, with each vertex mapped to a PPD gate, an energy metric for each vertex is computed. The metric is the energy saving obtainable if the vertex gate is changed from PPD to NPD. This metric is -1 for a vertex if either (i) the delay increase of the vertex due to the change is greater than the vertex's time slack or (ii) the delay increase of any of the driven gates (because of the reduction in drive strength) is greater than its time slack. In case all the delay increases are less than the corresponding time slacks, the change in energy of the driven gates is also included in the metric for the driver gate. The metric value might still be negative (if energy is increased due to change from PPD to NPD), but it will be greater than -1 and hence the vertex will have higher priority for changing from PPD to NPD over vertices which have metric -1.

   Next, every vertex is visited in decreasing order of the energy metric, and it is attempted to replace the gate corresponding to the vertex with the NPD equivalent. This might not always be possible, even for a vertex with non-negative metric value, because the metric for the vertex was computed under the assumption that all other vertices are mapped to PPD gates and hence the vertex had a lot of slack. As the

replacement proceeds, the slack available to a vertex keeps on reducing and might not be sufficiently large to allow PPD-NPD replacement. After every replacement, the slacks for all gates are recomputed using the function Update_Time_Slacks(V^s). The whole procedure is repeated till all vertices have been visited. Figure 15 gives the details of the algorithm used in this step.

After step 2 has been carried out, some PPD gates on off-critical paths have been replaced by NPD gates. This step does not change the total circuit delay since only those gates which have sufficient time-slack are replaced by NPD gates.

4.2.1.4 Results of PPD-NPD Replacement

The low power/high delay domino gates are swapped with the regular domino gates in the off-critical paths to reduce total energy consumption while keeping the delay same. An inverted clock signal is applied to the precharge NMOS transistors in NPD1 and NPD2 gates. Using NPD1 gates leads to an area overhead and using NPD2 gates leads to the need for a second threshold voltage. Depending on the application, one of these gate types can be selected. Once the gate type is selected, HSPICE simulation driven look-up tables are generated for delay and energy values for the standard gates. These values are found for all the gates for various load capacitance values and drivers by running HSPICE simulations using 0.18μ SPICE Level 49 MOSFET models, which has a standard supply voltage of 1.8 Volts and a standard threshold voltage of 0.4 Volts. Delay and energy values for the gates change depending on the driving strength of the driving gate. Therefore, two simulations were run for every NPD and PPD gate: one for the case when the gate is driven by an NPD gate and the other for the case when it is driven by a

PPD gate. Then, the algorithm in Figure 15 is run to swap the PPD gates in the non-critical paths by their NPD equivalents.

The developed scheme is tested on the ISCAS'85 benchmark circuits. The circuits are synthesized using Synopsys Design Compiler to a target library that is reduced to have only two to four input 'AND' and 'OR' gates, and 'INVERTER' gates for simplicity. Circuits are optimized for minimum delay. Since domino logic can only implement non-inverting functions, the resulting circuit was not suitable for mapping to domino logic gates. Bubble pushing and duplication algorithm [55] was implemented to turn the CMOS logic style mapping to domino logic style, where inverters are present only at the primary inputs. Bubble pushing and duplication algorithm is explained in Appendix B.

Switching activity of 0.1 and static probability of 0.5 were applied to the primary inputs. Synopsys Design Compiler was used to get the static probabilities of the internal nodes. The average energy, $\overline{E}_i$, for gate 'i' was calculated using Equation 7.

The delay and energy characteristics of the NPD2 gates can be varied by changing the threshold voltage of the inverter PMOS transistor, and NPD1 gates by changing the size of the NMOS pull-up transistor connected in series with the CMOS inverter. This variation in the delay and energy characteristics leads to a variation in the number of PPD gates that are replaced with NPD gates by the replacement algorithm and hence a variation in the overall energy savings can be obtained. Figures 16 and 17 shows the variation in average energy savings with respect to changing NMOS pull-up width for NPD1 replacement, and to changing PMOS threshold voltage for NPD2 replacement schemes. As shown in Figure 16, average savings go up with the increasing NMOS pull-

up size for NPD1 replacement. The designer should decide how much area is to be sacrificed for energy reduction. Tables 1 and 2 show the energy savings for various ISCAS'85 benchmark circuits for NPD1 and NPD2 replacement schemes.

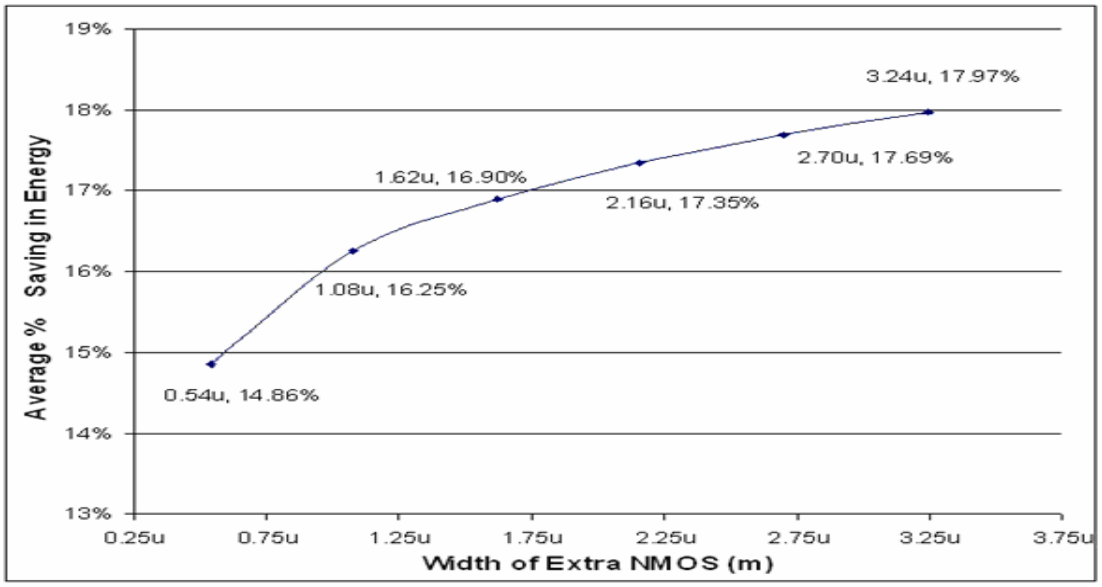

Figure 16. Variation of average energy savings with width of extra NMOS transistor for ISCAS'85 benchmark circuits for NPD1 replacement.

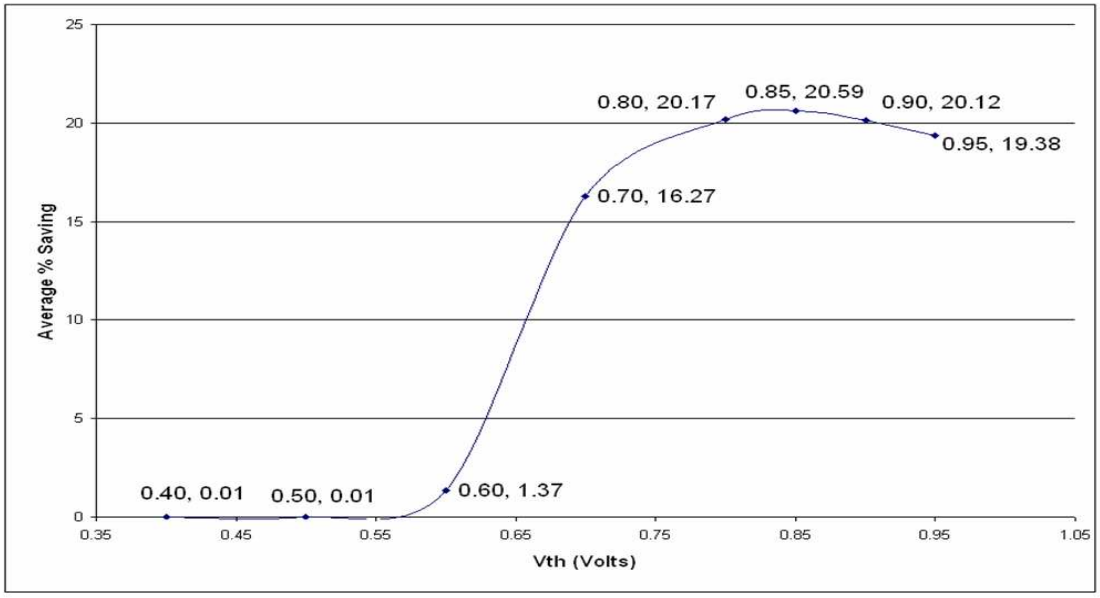

Figure 17. Variation of average energy savings with threshold voltage of inverter PMOS transistor for ISCAS'85 benchmark circuits for NPD2 replacement.

Table 1. Results for NPD1 replacement scheme when the extra NMOS transistor is sized to be 1.08μ.

Circuit	# Gates	Delay (nsec)	Initial Energy (pJ)	Final Energy (pJ)	Fraction of NPD Gates	% Saving
C432	318	1.91	9.05	8.62	0.14	4.74
C499	937	2.11	24.3	22.4	0.18	8.08
C1908	794	2.76	21.0	19.1	0.23	9.13
C2670	1253	3.60	33.8	24.7	0.69	27.03
C3540	1987	3.40	57.3	45.9	0.47	19.98
C5315	2861	2.72	75.7	60.4	0.51	20.24
C7552	3582	3.76	105	79.5	0.65	24.55
Average	1676				0.41	16.25

Table 2. Results for NPD2 replacement scheme when the second threshold voltage is chosen to be 0.85 V.

Circuit	# Gates	Initial Delay (nsec)	Final Delay (nsec)	Initial Energy (pJ)	Final Energy (pJ)	Fraction of NPD Gates	% Saving
C432	318	1.86	1.85	7.57	6.45	0.73	14.8
C499	937	2.02	2.01	19.64	15.43	0.90	21.5
C1908	794	2.61	2.61	17.03	13.76	0.78	19.2
C2670	1253	3.23	3.22	27.82	22.13	0.93	20.4
C3540	1987	3.21	3.21	46.77	36.36	0.96	22.3
C5315	2861	2.55	2.55	61.19	47.52	0.95	22.3
C7552	3582	3.49	3.49	84.65	64.66	0.98	23.6
Average	1676					0.89	20.6

It is seen from the results that NPD2 replacement is more effective than NPD1 replacement. However, there can be cases where NPD1 replacement may be preferred, such as not having a second threshold voltage. There are several reasons for NPD2 replacement to be more effective:

- NPD2 gates are faster than NPD1 gates. The NMOS transistor added between the power line and the circuit acts as a series resistor, which increases the time constant of the NPD1 gates. This leads to more PPD-NPD2 replacement compared to PPD-NPD1 replacement as seen in the "fraction of NPD gates" column of Tables 1 and 2.
- The output of NPD1 gates has a voltage swing less than the rail to rail voltage because of the NMOS transistor connected between the inverter and the power rail. This results in reduced driving strength for the NPD1 gates. An NPD1 gate has a more negative effect on its successors' delays than an NPD2 gate.

4.2.2 Dual Supply Voltage CMOS Design

Because of the quadratic relation of the supply voltage to the dynamic energy consumption of a gate, changing the supply voltage of a gate impacts its energy consumption significantly. Lowering supply voltage reduces the gate's dynamic energy consumption at the expense of increased delay. In a CMOS circuit, critical paths are a small fraction of the total number of paths [3][56]-[58]. This observation suggests that there are some gates that have time slacks greater than zero. The time slacks of such gates are exploited by applying a lower supply voltage to such gates. There are many studies in academia about multiple supply voltage CMOS design [1]-[9][25][26][39]-[44][56]-[58]. It is shown in [56] and [58] that the improvement of power consumption saturates with

the increased number power supplies used. The most significant improvement in power consumption is obtained when a single supply design is optimized to use dual supply voltages. Therefore, this work focuses on dual supply voltage usage. However, the same technique can be used in more than two supply voltage designs as well.

The main problem of designing dual supply voltage CMOS circuits is the increased leakage current in the high voltage gates when a low voltage gate is driving a high voltage gate. Figure 18 shows the case when a low voltage inverter is driving a high voltage inverter. Assume that the circuit is designed in 70 nm technology [59]. The regular supply voltage for this technology is 1 Volt. Threshold voltages are 0.2 and -0.2 V for NMOS and PMOS transistors, respectively. Assume that a low voltage value of 0.7 is used as the second supply voltage. If logic zero (0 V) is applied to input, Node X will be stabilized at a voltage of 0.7 V. A voltage of 0.7 V is not enough to put M2 into cut-off mode (because $|V_{GS}| > |V_{Th}|$). As a result, a large current passes through M2 when the input to the gate is logic high. Even when the low voltage is selected high enough to be able to put M2 into cut-off stage when the input is at logic high, the sub-threshold leakage will still be significantly larger than the case when the gate is driven by a high voltage gate. This is because of the $(|V_{GS}| - |V_{Th}|)$ term in the exponent in Equation 5. Low voltage and high voltage gates can not be used in a dual supply voltage circuit without considering its effect on the leakage current. Voltage level converter circuits are designed to solve the increased leakage problem [60]-[63]. Level converting circuits (level shifters) convert a low voltage signal to a high voltage signal without having an increased leakage current. Level conversion can be performed by combinational circuits [60]-[62] or by flip-flops [63].

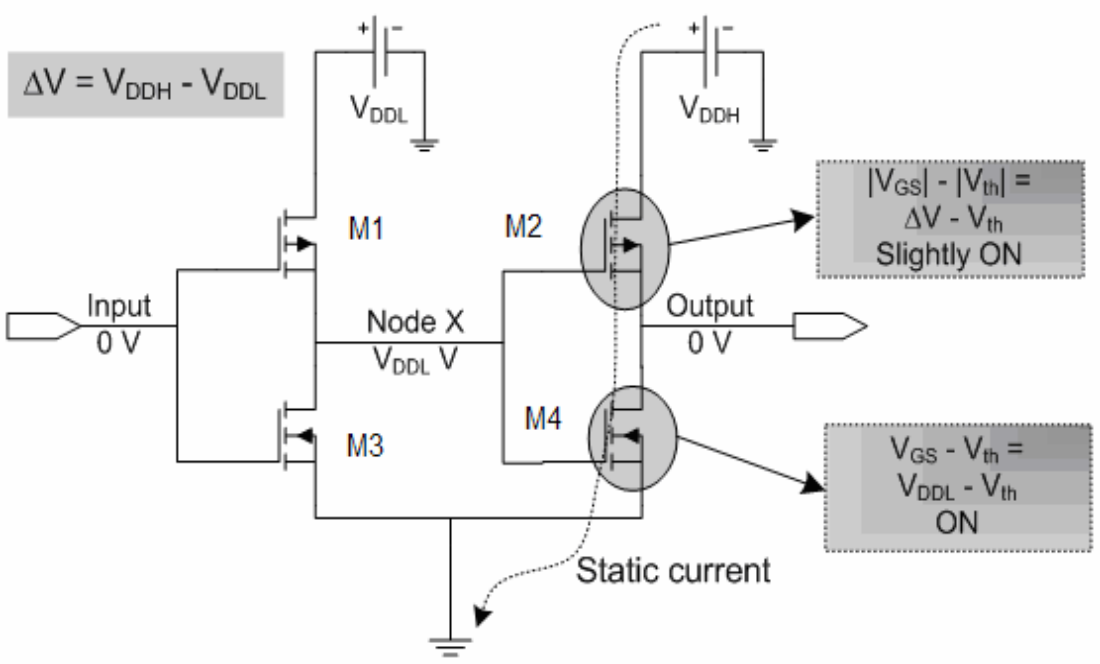

Figure 18. A low voltage inverter driving a high voltage inverter.

The level converter circuitry, which is added to the circuit to solve the problem of increased leakage current, unfortunately introduces area, delay, and energy overhead. To reduce overhead of the level shifters, researchers have proposed Clustered Voltage Scaling (CVS) [39][40] and Module Level Voltage Scaling (MLVS) [6][41]. In CVS, low voltage clusters are constructed in the circuit in such a way that there is no low voltage gate driving a high voltage gate. This is done by assigning low supply voltage to the gates starting from the circuit outputs depending on their slacks, eliminating the use of level shifters in the combinational logic. However, there is still need for level shifters in the flip-flops. MLVS assigns the dual supply voltages to partitions of the circuit, reducing the number of level shifters needed. Clearly both of these methods introduce additional constraints to the dual supply voltage assignment process, reducing the obtainable energy savings. There has also been research in gate-level dual supply voltage assignment

[42]-[44]. These techniques require the use of level shifters when a high voltage gate is driven by a low voltage gate. The level shifting circuitry in the optimized circuits constitutes 8% of the total energy consumption [44].

4.2.2.1 CMOS Gate Design with Built-in Level Shifting Capability

The main need for level shifters in dual supply voltage CMOS circuits is to reduce the static leakage current in the high voltage gates when they are driven by low voltage gates. The low voltage output applied to the gate of the PMOS transistor in a high voltage gate is not enough to turn the PMOS transistor completely off. The slightly on PMOS transistor causes static current to flow from the power supply to the ground wasting significant energy. Level shifters are able to shift the voltage from a lower level to a higher one but since they do not perform any logic function, they cause area, delay and energy overhead.

To eliminate the overhead of additional level shifters, a second threshold voltage is used for the PMOS transistors in the high voltage gates which are driven by low voltage gates [64]. By increasing the magnitude of the threshold voltage of the PMOS transistor, the static current flowing through the transistor when the gate voltage is V_{DDL} (lower supply voltage) is decreased. This increases the rise time for that gate slightly. Depending on the second threshold voltage value (the magnitude of the original PMOS threshold voltage will be referred to as V_{thp1} and the magnitude of the second PMOS threshold voltage will be referred to as V_{thp2} from now on. $V_{thp2} > V_{thp1}$), the static leakage current can be decreased substantially.

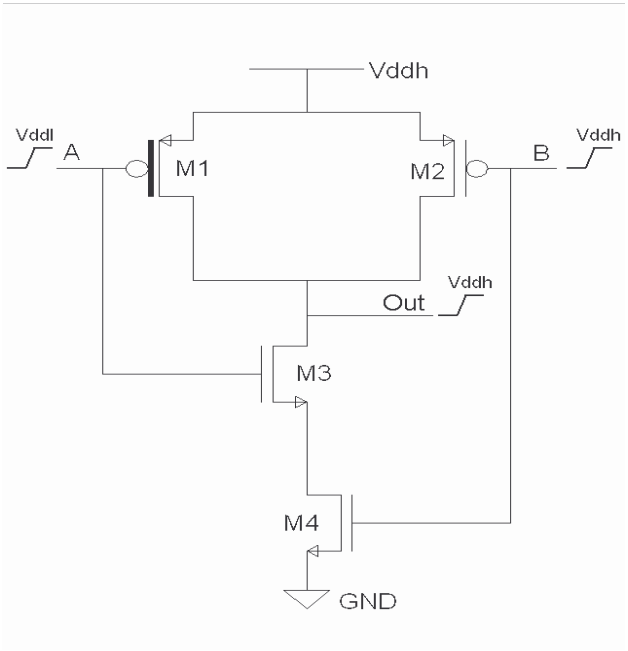

Figure 19. Level shifting NAND2 gate with one high voltage and one low voltage inputs. M1 has higher threshold voltage magnitude than M2.

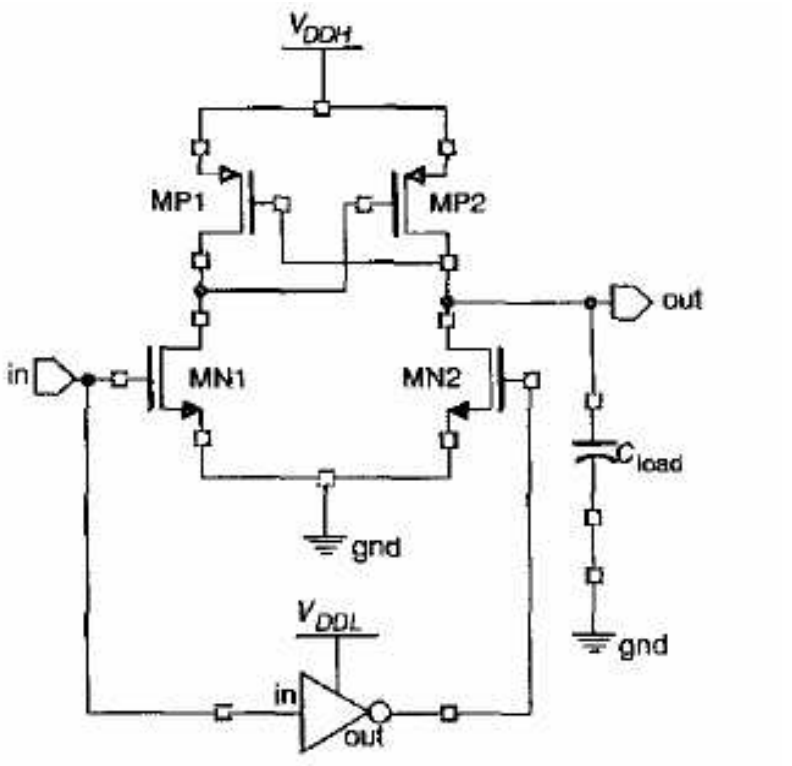

Figure 20. A regular level shifter.

Figure 19 shows the schematic for a NAND2 gate with built-in level shifting capability. The darker line at the gate of M1 depicts the higher threshold voltage magnitude used for it. Such gates will be referred to as "level shifters" even though they are different from conventional level shifters. Figure 20 shows the schematic for a regular level shifter. Figure 21 shows SPICE simulation results for static power dissipation of a level shifting NOT gate for different values of V_{thp2} when the input voltage is at 1.4 Volts and the power supply voltage is at 1.8 Volts. 0.18µ SPICE Level 49 MOSFET models [54] were used for the simulations. The static power dissipation values for a high voltage NOT gate (driven by another high voltage gate) and a low voltage NOT gate (driven by another low voltage gate) are also given for comparison. It is seen that when V_{thp2} is the same as V_{thp1} (=0.4V), there is significant static power dissipation in the level shifting gate due to the inverter PMOS transistor not being OFF. Figure 22 and Figure 23 compare the propagation delay (average of delays for rising and falling output) and switching energy (the energy spent for 0 to 1 transition at the output) respectively for a high voltage, low voltage, and level shifting inverter for different values of V_{thp2}. The high supply voltage is again 1.8 Volts and the low supply voltage is 1.4 Volts. The threshold voltage magnitude for a regular PMOS (V_{thp1}) is 0.4 Volts. Given that conventional level shifters have a delay close to two serially connected FO4 inverter delays of that technology [60], Figure 22 shows that a level shifting NOT gate will be much faster than a conventional level shifter followed by a high voltage NOT gate even when a high V_{thp2} is chosen. This trend is similar for NAND and NOR gates as well.

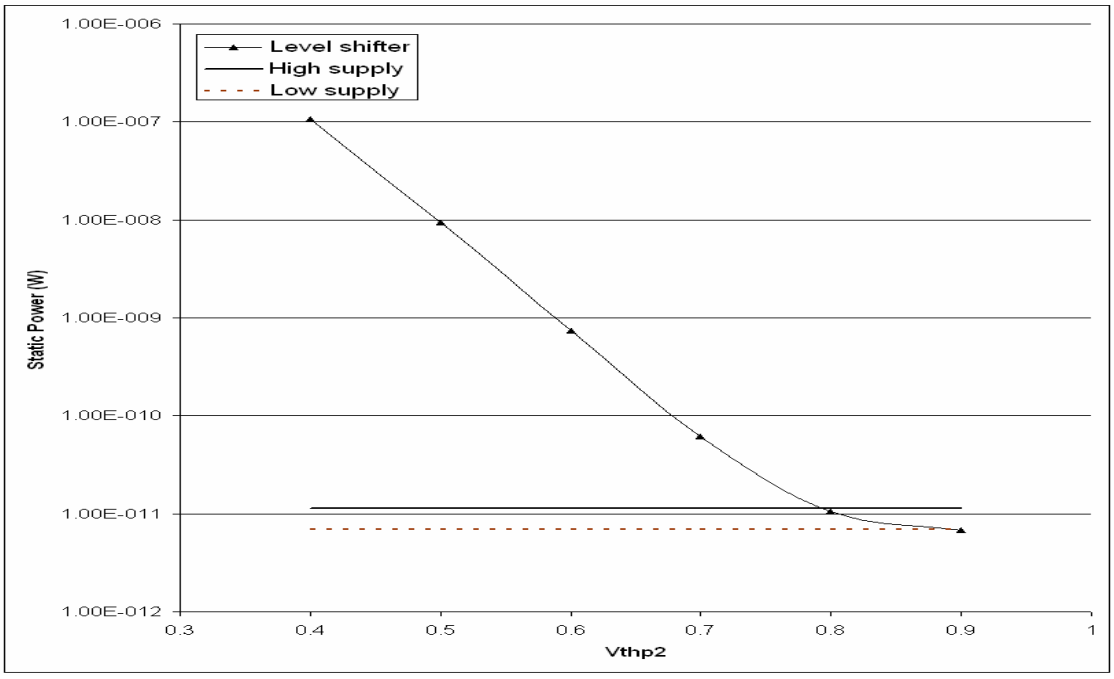

Figure 21. Static power dissipation of high voltage, low voltage, and level shifting NOT gates for different V_{thp2} values.

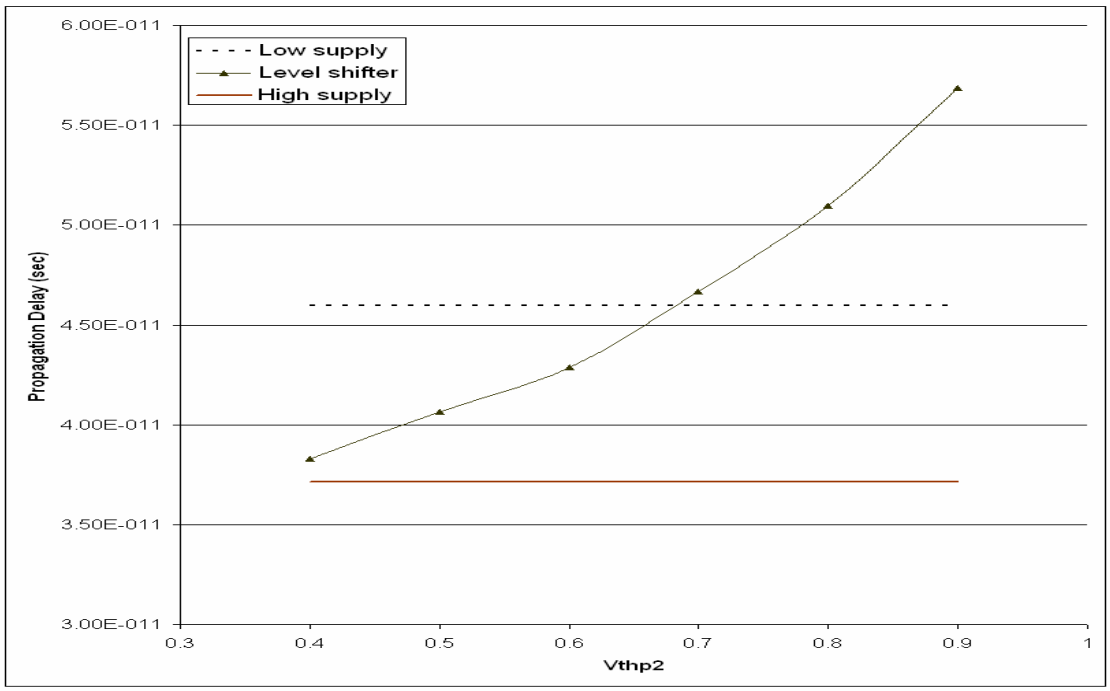

Figure 22. Propagation delay of high voltage, low voltage, and level shifting NOT gates for different V_{thp2} values.

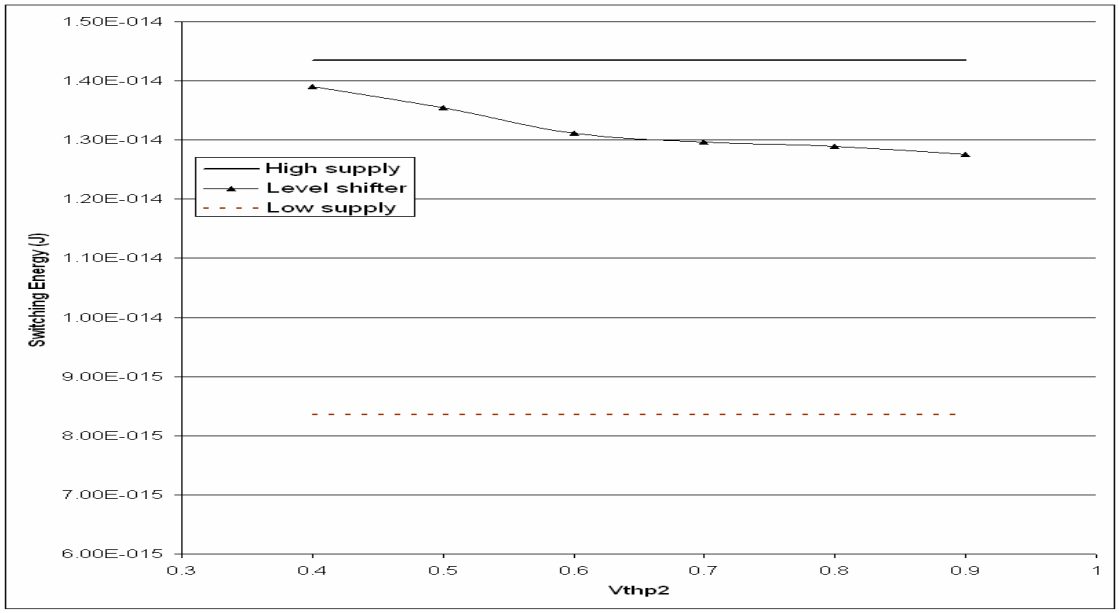

Figure 23. Switching energy of high voltage, low voltage, and level shifting NOT gates for different V_{thp2} values.

4.2.2.2 Algorithm for Dual Supply Voltage Assignment Using Level Shifting CMOS Logic Gates

The developed level shifting gates can be used in any application which uses more than one supply voltages. Because of its advantages over the regular level shifters, using them will increase the energy savings. The algorithm that was used in this research is explained in this section.

To reduce the energy consumption of combinational CMOS logic circuits, high supply voltage (V_{DDH}) gates on the non-critical paths are replaced with slower, lower energy low supply voltage (V_{DDL}) gates. The total delay of the circuit remains the same as the original circuit which has only V_{DDH} gates.

The combinational circuit is represented as a directed acyclic graph (DAG), G(V,E). If the circuit has multiple primary inputs (PIs), a dummy PI vertex, PI_d, is created which fans-out to the original PIs. Underlying this is the assumption that all

inputs arrive simultaneously. Similarly for POs, a dummy PO vertex, PO_d, is created which has fan-ins from the original POs. Each vertex 'v' of the DAG has associated with it the following information:

1. The logic function computed by the gate corresponding to the vertex.
2. The supply voltage type of the gate (V_{DDH} or V_{DDL}).
3. The current delay (v.delay), energy (v.energy), time slack (v.ts), early start time (v.es), early finish time (v.ef), late start time (v.ls) and late finish time (v.lf) of the vertex at any stage of the replacement process. The delay, dynamic energy consumption, and static energy consumption values for V_{DDH} and V_{DDL} gates of different fan-ins, load capacitances, types, input signal ramps, and gate sizes are obtained from SPICE look-up tables. These values for a gate are updated whenever it or its driver gates change from type V_{DDH} to V_{DDL}. The dummy vertices have zero delays and zero energy consumptions. Delay and energy consumption calculations are done as explained in Section 3.2.

The algorithm for replacing V_{DDH} gates with V_{DDL} equivalent gates consists of two steps:

1. Initialization: In this step, the DAG is topologically sorted to get the sorted vertex list, V^s. V^s is used to compute the total circuit delay, T, of the baseline circuit in which each vertex is mapped to a V_{DDH} gate. In this step, the initial total energy of the baseline circuit with all V_{DDH} gates is also computed. Delay and energy consumption calculations are done as explained in Section 3.2.
2. V_{DDH} to V_{DDL} replacement: First, with each vertex mapped to a V_{DDH} gate, an energy metric for each vertex is computed. The metric is the energy saving obtainable if the

vertex gate is changed from V_{DDH} to V_{DDL}. This metric is -1 for a vertex if either (i) the delay increase of the vertex due to the change is greater than the vertex's time slack or (ii) the delay increase of any of the driven gates (i.e. if a gate is assigned a low supply voltage, that gate's successors that are assigned to high supply voltage are turned into level shifting gates, which increases their delay) or any of the gates following the driven gates (a level shifting gate has less driving strength compared to a high voltage gate, therefore the gates driven by them slows down) is greater than its time slack. In case all the delay increases are less than the corresponding time slacks, the change in energy of the driven gates is also included in the metric for the driver gate. The metric value might still be negative (if energy is increased due to change from V_{DDH} to V_{DDL}), but it will be greater than -1 and hence the vertex will have higher priority for changing from V_{DDH} to V_{DDL} over vertices which have metric -1.

Next, every vertex is visited in decreasing order of the energy metric, and it is attempted to replace the gate corresponding to the vertex with the V_{DDL} equivalent. This might not always be possible, even for a vertex with non-negative metric value, because the metric for the vertex was computed under the assumption that all other vertices are mapped to V_{DDH} gates and hence the vertex had a lot of slack. As the replacement proceeds, the slack available to a vertex keeps on reducing and might not be sufficiently large to allow V_{DDH}-V_{DDL} replacement. After every replacement, the slacks for all gates are recomputed using the function Update_Time_Slacks(V^S). The whole procedure is repeated till all vertices have been visited. Figure 24 gives the details of the algorithm used in this step. Section 4.2.2.5 gives the implementation details of Update_Time_Slacks procedure and complexity analysis of the algorithm.

Algorithm V_{DDH}-V_{DDL}

Inputs: Topologically sorted list of circuit vertices, V^s;

Output: Circuit with off-critical path V_{DDH} gates replaced with V_{DDL} gates.

```
for as many times as number of vertices {
    Let v be the vertex with maximum metric value.
    Map v to low supply voltage
    low_energy←0
    for every predecessor p of v {
        Depending on whether p is mapped to V_DDH or V_DDL, look-up "pin_energy" as the energy
        consumption of this gate using SPICE look-up tables and switching activity and static
        probability values for this gate .
        low_energy← low_energy+pin_energy
    }
    high_delay←delay of v when it is driven by the predecessor p with maximum early finish
    time. Found using SPICE look-up tables.
    low_energy←low_energy/(No. of predecessors p of v)
    if((high_delay-v.delay)≤v.ts) { // Check if assigning low supply voltage violates timing slack
                                    // for this gate
        flag← 0
        for every successor p of v {
          Compute largest delay of p with v mapped to a V_DDL gate. Let this be "p_delay".
          if((v.es+high_delay)≥p.es) {
              if((v.es+high_delay+p_delay)>p.lf)
                  flag← 1,break.
          }
          else if((p_delay-p.delay)>p.ts)
              flag← 1,break.
        }
        Update_Time_Slacks(V^s)
        if(time slack for any gate < 0)
          flag← 1,break.
        if(flag=0) {
          Map v to a V_DDL gate.
          v.delay← high_delay
          v.energy← low_energy
          v.metric←-2
          Map the successors of v that are assigned to high supply voltage to be level shifters
          Update_Time_Slacks(V^s)
        }
        else v.metric←-1
        Map v back to high supply voltage
    }
    else v.metric←-1
    Map v back to high supply voltage
}
```

Figure 24. Algorithm for V_{DDH}-V_{DDL} replacement.

After step 2 has been carried out, some V_{DDH} gates on off-critical paths have been replaced by V_{DDL} gates. This step does not change the total circuit delay since only those V_{DDH} gates which have sufficient time-slack are replaced by V_{DDL} gates. The input PMOS transistors of the V_{DDH} gates that are driven by V_{DDL} gates are modified to use the higher threshold voltage, V_{thp2} to reduce their static energy dissipation.

4.2.2.3 Results for Gate-Level Dual Supply Voltage CMOS Implementation

The algorithm in Figure 24 is implemented using C++. TSMC 0.18μ technology parameters [54][52] are used to generate SPICE look-up tables. ISCAS'85 benchmark circuits are synthesized using Synopsys Design Compiler. Only 2 to 4 input NAND and NOR gates and inverters are used in the synthesis. Then the dual supply voltage assignment algorithm is run on the synthesized circuits. Table 3 shows the optimization results for ISCAS'85 benchmark circuits. Initial energy is the average energy per clock cycle for the baseline circuit which has 1.8 Volt supply voltage for all gates. Final energy is the average energy per clock cycle for the optimized circuit which has some fraction of V_{DDL} gates. The fraction of V_{DDH} gates driven by V_{DDL} gates are also given in Table 3. For small circuits, best energy reduction is obtained when V_{thp2} is 0.4 Volts (i.e. level shifters are identical to high voltage gates). This is due to the fact that static energy dissipation is negligible in small circuits compared to the dynamic energy consumption. Even the high static energy dissipated in the high voltage gates driven by low voltage gates does not increase the total energy dissipation of the circuit substantially. As the circuit size increases (number of gates > 1000), the optimum V_{thp2} value increases to 0.5. It can be expected to see the V_{thp2} increase more for even larger circuits.

The algorithm is run for a range of V_{DDL} and V_{thp2} values. Figure 25 shows the variation of average energy savings for the ISCAS'85 benchmark circuits with V_{DDL} varied from 1.1 to 1.6 Volts and V_{Thp2} varied from 0.4 to 0.9 Volts. Saving of "0" means that the algorithm couldn't find any V_{DDH} gates to swap with a V_{DDL} equivalent. The best average energy saving of 19.81% was obtained for V_{DDL} of 1.4 Volts and V_{Thp2} of 0.5 Volts.

Table 3. Results of dual supply voltage assignment for input switching activity of 0.1.

Ckt	# Gates	Ckt Delay (nsec)	V_{DDL} (V)	V_{Thp2} (V)	Initial Energy (pJ/cycle)	Final Energy (pJ/cycle)	Fraction of Low V_{DD} Gates	Fraction of High V_{Thp2} gates	% Saving
C432	332	1.22	1.5	0.4	0.49	0.42	0.48	0.19	14.56
C499	772	1.17	1.5	0.4	1.40	1.21	0.34	0.17	13.91
C1908	899	1.60	1.5	0.4	1.58	1.30	0.51	0.18	17.88
C2670	934	0.92	1.4	0.4	1.52	1.18	0.59	0.15	22.29
C3540	1395	1.79	1.4	0.5	2.28	1.82	0.59	0.14	20.41
C5315	2106	1.50	1.3	0.5	3.64	2.61	0.64	0.12	28.21
C7552	2846	1.30	1.4	0.5	5.48	4.05	0.62	0.13	26.06

The optimum V_{DDL}, V_{thp2} and the low voltage assignment depends on the circuit size, structure and input switching activity. It is desirable to reduce the energy dissipation for a given circuit for a range of input switching activities. The circuits which were optimized for an input switching activity of 0.1 are simulated with different input switching activities. Figure 26 shows the average energy savings obtained for the benchmark circuits, which are optimized with input switching activity of 0.1, when different switching activities are applied to them. As seen in the figure, the circuits optimized for switching activity of 0.1 gives similar savings for different input switching activities as well. Even though the input switching activity is an important factor in the

optimization, a circuit which is optimized for a particular switching activity still saves energy for other switching activities.

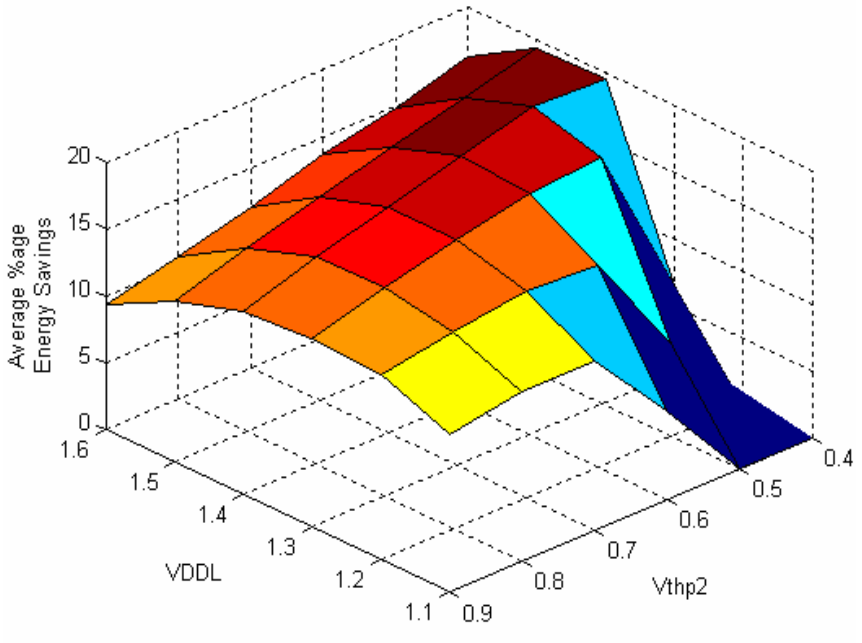

Figure 25. Variation of average energy savings with varying V_{thp2} and V_{DDL} for switching activity = 0.1.

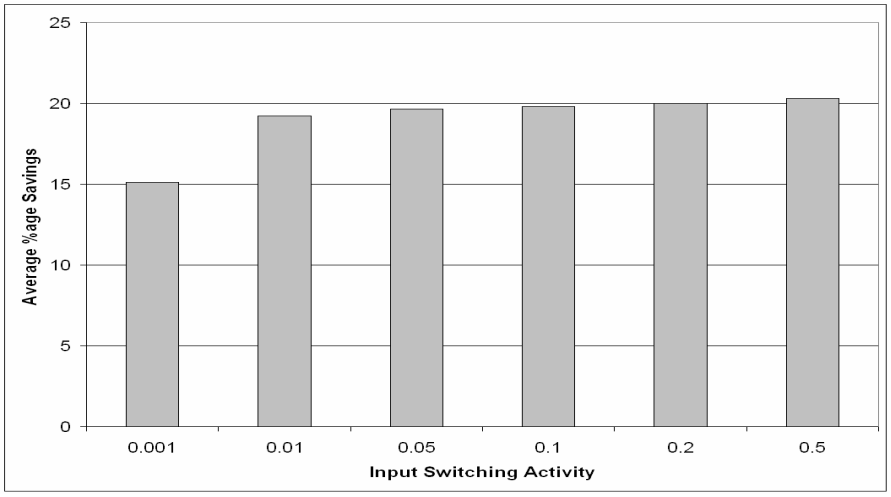

Figure 26. Average energy savings obtained for different switching activities with circuits optimized for switching activity of 0.1.

The gates used in the library are designed as balanced gates i.e. the PMOS transistors are sized up to compensate for the smaller mobility. Therefore, the rise times and fall times of the gates are comparable. If minimum sized gates are used, instead of a lower supply voltage, a higher ground can be used to design the low power gates. Then, the NMOS transistor's threshold voltage should be modified in the level shifting gates instead of the PMOS transistor. This will prevent a significant delay increase in level shifting circuits which will happen if the threshold voltages of the minimum sized PMOS transistors are increased.

4.2.2.4 Improvement Obtained by Using Level Shifting Logic Gates over Regular Level Shifters

A regular level converter (Figure 20) has a delay close to two FO4 inverter delays of that technology [60]. For a zero to one transition at the input, MN1 turns ON, reducing the gate voltage of MP2. Then MP2 turns ON and the output rises to V_{DDH}. For a one to zero transition at the input, the inverter pulls the gate voltage of MN2 up to V_{DDL}, turning MN2 ON. Then MN2 pulls the output voltage down to zero. Both transitions have two stages between the input and the output. This results in both increased delay and increased energy dissipation (caused by charging/discharging of the internal nodes) compared to a simple CMOS gate. The main disadvantage of regular level shifters is the delay overhead. A level shifter added to a path reduces the slacks of the other gates in that path, reducing the number of gates that can operate with low supply voltage.

Improvements to the level shifting circuit in Figure 20 have been developed by the research community [60][61][62]. A level shifter with a delay slightly less than 2 FO4 inverter delays is presented in [60]. This circuit however has significant energy penalty. The algorithm in Figure 24 is modified to use level shifting circuitry when a low voltage

Table 4. Comparison of energy savings when dual supply voltage assignment is done (i) using level shifting logic gates and (ii) using dedicated level shifters.

Circuit	Number of Gates	Delay (psec)	V_{Thp2}	V_{DDL}	Number of V_{DDL} Gates	Number of Level Converters	% Energy Saving
c432 (i)	267	518	0.2	0.9	196	0	16.29
c432 (ii)	301	518	-	0.8	107	34	13.31
c499 (i)	835	401	0.2	0.85	233	0	7.68
c499 (ii)	845	401	-	0.8	54	10	1.45
c1908 (i)	680	605	0.2	0.9	485	0	14.78
c1908 (ii)	685	605	-	0.7	99	5	4.21
c2670 (i)	875	375	0.2	0.9	759	0	19.95
c2670 (ii)	881	375	-	0.85	234	6	8.03
c3540 (i)	1319	736	0.3	0.85	343	84	9.19
c3540 (ii)	1323	736	-	0.7	36	4	1.18
c5315 (i)	1994	582	0.25	0.85	1270	146	18.75
c5315 (ii)	2033	582	-	0.7	493	39	13.72
c7552 (i)	2538	515	0.3	0.9	1715	214	15.85
c7552 (ii)	2541	515	-	0.8	199	3	4.00
Average							14.64
							6.56

gate is driving a high voltage gate. The level shifters are assumed to have zero energy penalties and 2X FO4 inverter delays. ISCAS'85 benchmark circuits are synthesized using Synopsys Design Compiler for the maximum speed. 70 nm technology parameters [59] are used in SPICE simulations to generate the look-up tables. Results of the optimizations are given in Table 4. For every circuit, first line in the column gives the optimization results when level converting logic gates are used. The second line gives the results when level converters are used. Circuit inputs are assumed to have 0.1 switching

activity and 0.5 static probability. Similar to the simulations run in 180 nm technology, circuits with fewer gates do not need level shifting gates with increased PMOS threshold voltage magnitude. However, circuits with a large number of gates (number of gates > 1000) need the level shifting gates to reduce leakage energy dissipation. Results show that, even when the energy consumption of the level shifters is neglected, the delay overhead of them causes the energy consumption reduction to go down. For circuits that are optimized for speed, a delay overhead of 2 FO4 inverter delays for the level converters reduces the low supply voltage usage dramatically.

4.2.2.5 Complexity Analysis of Dual Supply Voltage Assignment Algorithm

Before analyzing the complexity of the algorithm, the Update_Time_Slacks procedure should be analyzed. This procedure is used to update the delay, ramp, and energy consumption values for all the gates in the circuit. The procedure takes the topologically sorted node list, V^S, as the input. The straightforward way to update all of this information is to start from the primary inputs and to get the necessary information using the SPICE look-up tables and the output ramp and voltage magnitude values of the predecessors for all the gates. In this manner, the procedure will have a complexity of N, where N is the number of nodes in the circuit. However, changing the supply voltage of a single gate does not necessarily change the input ramps for all the gates in the circuit. Therefore, to achieve faster run times, only the gates those are in the output cone of the modified gate are updated. This approach reduces the complexity of the procedure to a fraction of what it was before, but the complexity is still O(N). To achieve even faster run times by compromising the accuracy, only the gates which are the immediate fanouts of the modified gates are updated when Update_Time_Slacks procedure is run after a gate is

replaced with its low voltage equivalent. This approach reduces the complexity of the procedure to O(1) since the number of gates to be updated is not a function of N. This may lead to errors in the delay values of some gates. After the optimization is run, Update_Time_Slacks procedure is run to update all of the gates. If the timing is violated, the gates with the most negative time slacks are replaced back to operate with high voltage until the timing requirement is met. The error caused by this simplification was less than 1% of the circuit's deadline for the optimized circuits and backtracking was limited to converting a small number of gates back to high supply voltage.

When the complexity of Update_Time_Slacks procedure is reduced to O(1), the algorithm in Figure 24 has a complexity of O(N) because every gate is visited once to check if it can be applied low supply voltage, and checking has a complexity of O(1). Run times for the ISCAS'85 benchmark circuits are given in Table 5. When the code is profiled, it is seen that most of the time is spent in finding the gate delay and energy consumption using the look-up tables. Look-up tables are formed by performing many simulations for various values of supply voltage, threshold voltage, input signal ramp, input signal magnitude, and output capacitance. Then linear interpolation is used to find the delay and energy values for a given set of values. An improvement in the efficiency

Table 5. Run time of the low supply voltage assignment algorithm for ISCAS'85 benchmark circuits.

Circuit	c432	c499	c1908	c2670	c3540	c5315	c7552
Number of gates	267	835	680	875	1319	1994	2538
Run time (sec)	519	4182	3013	4219	11543	17191	24549

of the table look-up scheme will directly effect the run time of the optimization algorithm.

Run times are plotted with respect to the number of gates in Figure 27. The optimizations are run on a Sun-Blade-2500 machine.

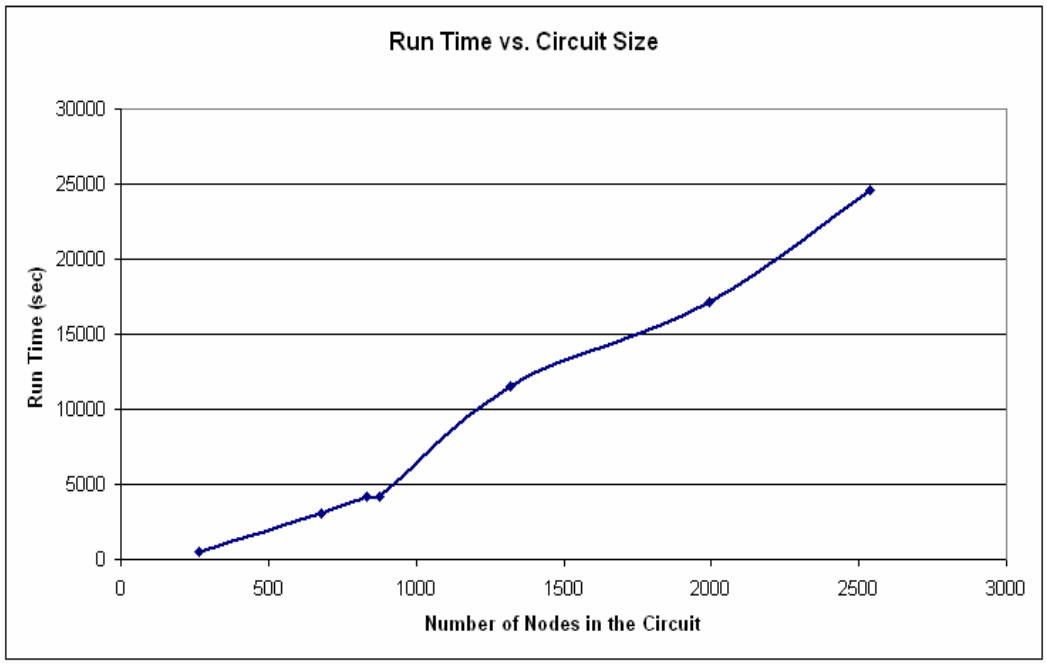

Figure 27. Run time of the low supply voltage assignment algorithm plotted against the number of nodes.

CHAPTER 5

IMPROVING SOFT ERROR TOLERANCE OF COMBINATIONAL CMOS CIRCUITS

Feature size reduction is the main driving force behind the increase in the performance of digital circuits. Feature sizes are reduced by roughly 30% in every technology generation (about 18 month cycle). Supply voltages are also reduced to limit the increase in dynamic energy consumption, leading to weakening of the drive strengths of transistors. This reduction in strength is compensated by reducing the threshold voltages of the transistors, leading to dramatic increase in static energy consumption. All these trends (reduced feature sizes, reduced supply and threshold voltages) have a negative effect on the circuit soft error tolerance (the term "soft error" refers to a bit-flip in a circuit node caused by a highly energetic particle, such as an alpha-particle or a neutron, striking that node). Soft error susceptibility of a circuit increases with feature size reduction because of the reduced average node capacitances. A same-energy particle will generate a larger voltage fluctuation at a node with less capacitance. The reduced noise margins caused by supply and threshold voltage reduction also aggravate the problem.

Soft error tolerance of combinational logic circuits is affected more than memory elements and flip-flops by technology scaling and architectural advances. Due to super-pipelining, the number of gates in pipeline stages is reduced, which reduces the average number of gates a particle induced glitch passes through before reaching a latch. In addition, higher clock frequencies increase the chance of a glitch being captured by a latch. Even though the soft error rate (SER) in combinational circuits is currently smaller

than that of sequential and memory elements, it is expected to rise 9 orders of magnitude between 1992 to 2011, when it will equal the SER of unprotected memory elements [12]. Given that memory elements used in critical missions are already being protected by techniques such as error correcting codes (ECC), the SER of combinational logic circuits may dominate the system SER in the near future.

There are three mechanisms in combinational logic circuits which mask the glitches generated by particle strikes [65]:

- Because of logical masking, a glitch might not propagate to a latch because of a gate on the path not being sensitized to facilitate glitch propagation.
- Because of electrical masking, a generated glitch might get attenuated because of the delays of the gates on the path to the output.
- Because of latching-window masking, a glitch that reaches the primary output might not cause an error because of the latch not being "open."

All of these three masking mechanisms diminish in effectiveness as technology scales down, resulting in an increase in SER in combinational logic circuits. Because of the decreasing number of gates in a pipeline stage, logical masking as well as electrical masking has been decreasing for new technology generations. Electrical masking has also been decreasing because of the reduction in node capacitances and supply voltages in every generation. Furthermore, increasing clock frequencies have reduced the time window in which latches are not accepting data, thereby reducing latching-window masking.

Generally, in mission-critical space applications combinational circuits are protected by using duplication/triplication and concurrent-error detection (CED) [66].

However, these methods have too high delay, area and power overheads. Recently, low-cost methods for increasing soft-error tolerance of commodity applications using time-redundancy [67] and partial duplication [68] have been proposed. However, these methods still add additional delay overhead to the original circuit because of the use of "checking" circuitry. Also, these methods have system level overheads (such as pipeline flushes) when an error is detected, either to correct the error or to perform the computation again.

A novel dynamic soft error tolerance control method is developed. This technique has negligible delay and energy overhead in the normal mode of operation, and it can increase the soft error tolerance of a circuit dramatically when the on-chip SER sensors detect an increased flux of energetic particles. The use of the developed control method allows nanometer CMOS logic to adapt to changing environmental radiation conditions. The method utilizes dynamic supply and threshold voltage (via back body bias voltage) modulation and variable capacitance banks. Various operating modes for a circuit can be implemented depending on the condition of the environment and the performance needs of the system. The effects of a particle strike on a single gate should be examined carefully to understand the effects of gate characteristics to the global soft error tolerance of the circuit.

5.1 Glitch Tolerance Characteristics of Individual Gates

There are two characteristics of interest for a single gate in terms of soft error tolerance: These are glitch generation characteristics and glitch propagation characteristics.

- The glitch generation characteristics of a logic gate determine the shape and magnitude of the voltage glitch generated at the output of the gate because of a particle strike on the gate.
- The glitch propagation characteristics of a logic gate determine how the gate attenuates a glitch that is generated at some prior circuit node as it passes through the logic gate.

When a particle strikes a circuit node, the voltage magnitude of the corresponding glitch is dependent on the total capacitance of the node. The duration of the generated glitch is dependent on the delay of the gate that is driving the node. If the gate driving the node is fast, it will quickly discharge (or charge) the node back to its original value. Therefore, faster gates have better glitch generation characteristics in terms of the generated glitch width if the output capacitance is kept constant.

However, the behaviour is opposite for glitch propagation. Assuming a linear ramp at the output of the gate, for a gate propagation delay of d and glitch duration of w_i at the gate input, glitch duration at the output of the gate, w_o, can be approximated as follows:

$$\begin{aligned} w_o &= 0 \ \ if \ \ w_i < d \\ w_o &= 2 \cdot \left(w_i - d \right) \ if \ \ d < w_i < 2 \cdot d \\ w_o &= w_i \ \ if \ \ w_i > 2 \cdot d \end{aligned} \tag{8}$$

This model is similar to the glitch amplitude attenuation model used in [69]. As seen from Equation 8, a slow gate will attenuate a glitch at its output more compared to a fast gate. Therefore, slow gates have better glitch attenuation characteristics.

Increasing a gate's output capacitance increases the delay of that gate. This makes the glitch attenuation characteristics of that gate better. Furthermore, if the capacitance is

large enough, the particle may not have enough energy to create enough voltage fluctuation for an error. So, a large enough output capacitance may improve both glitch generation and glitch propagation characteristics of a gate at a cost of significantly increased gate delay.

Figures 28 and 29 show SPICE simulation results for generated glitch width and propagated glitch width, respectively, for an inverter with different values of gate supply voltage (V_{DD}), gate threshold voltage (V_{Th}), and gate's load capacitance (C_{Load}). In these plots, only one parameter is changing and the other parameters are kept constant. The SPICE models are for 70nm technology node [59]. The minimum and maximum values of the variables are indicated on the x-axis. It is seen that, if the output capacitance is kept constant, factors that slow down a gate (reduction in V_{DD}, and increase in V_{Th}) increase generated glitch width but also increase the attenuation of propagating glitches. The generated glitch width first increases with output capacitance, then it starts to decrease. This behaviour is explained as follows: If the capacitance is small, the voltage generated at the gate's output is clipped by the diode between the source and the body of the transistor. For this case, smaller capacitance will hold less charge for the same voltage (Q = CV), making the discharge (recharge) time faster. This initially results in larger glitch widths for increasing values of output capacitance. However, if the output capacitance is large enough, the magnitude of the generated voltage glitch will reduce, and eventually become too small to cause an error. The glitch width is taken as the duration between two $0.5 \times V_{DD}$ crossings of the gate's output

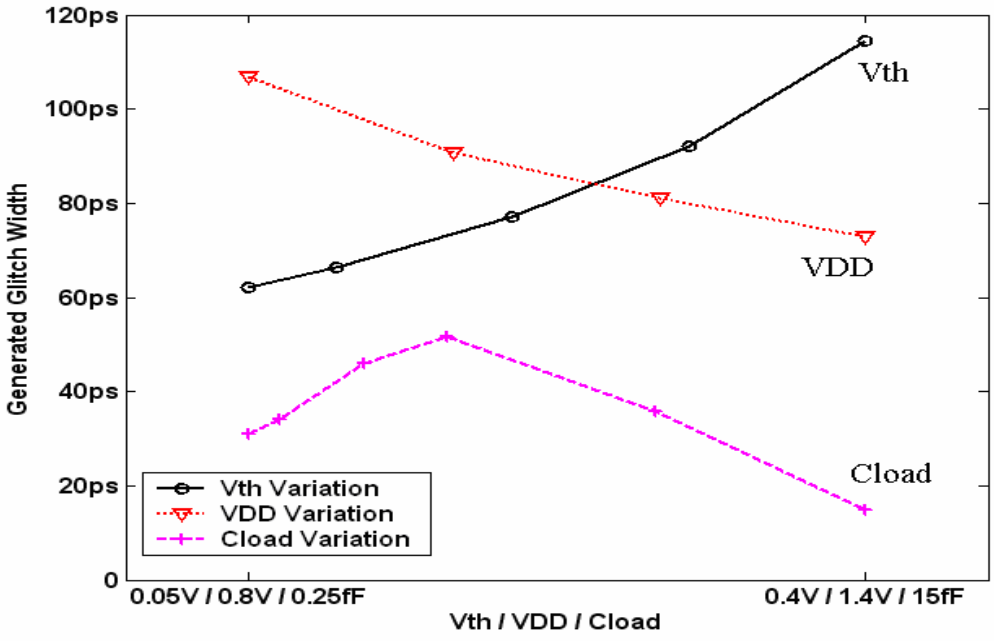

Figure 28. Glitch generation characteristics for an inverter for an injected charge of 16fC at its output.

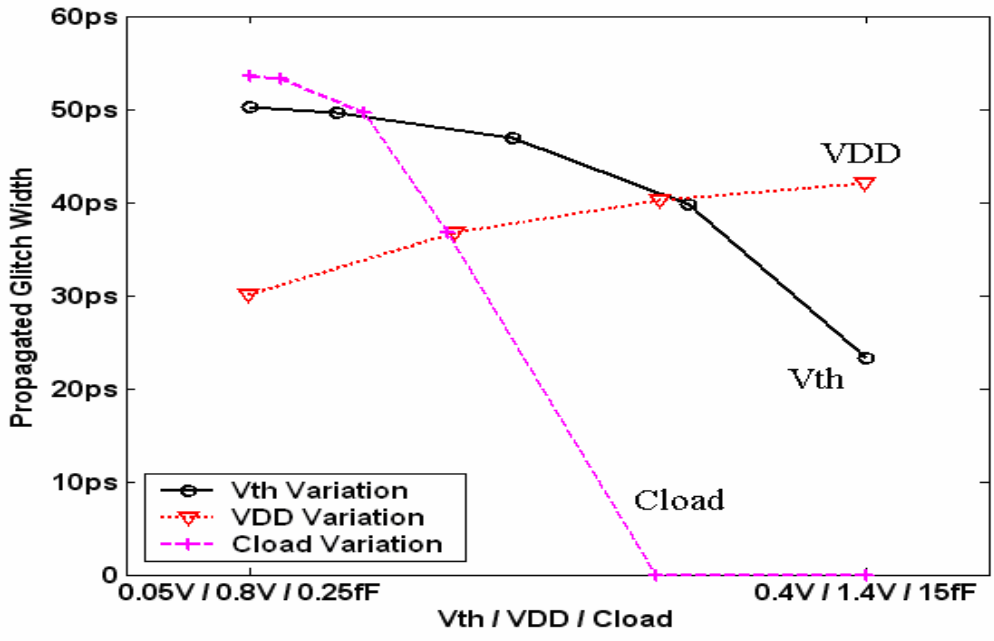

Figure 29. Glitch propagation characteristics of an inverter for an input glitch of duration 50ps.

There are two insights gained from the SPICE simulations. First, only generated glitch width or propagated glitch width are not enough to characterize the "softness" of a gate as this might lead to erroneous conclusions. If only glitch propagation characteristics are considered as a measure of the "softness" of a gate (as in [70]), slowing down a gate would apparently always reduce the softness of the circuit; however, a slower gate will produce a bigger glitch at its output when it is subjected to a particle strike. Such a glitch can easily propagate to the output and cause an error. The circuit must be considered as a whole and any soft error tolerance enhancement scheme should consider both glitch generation and glitch propagation characteristics of the gates as well as their location in the circuit.

The second insight is that the soft-error tolerance of a combinational circuit can be increased by increasing the capacitive loads of the gates at the primary outputs (POs) as this will attenuate all glitches reaching the POs (see propagated glitch width variation with C_{Load} in Figure 29). The capacitive load should be increased beyond the critical point (peak in C_{Load} curve in Figure 28) so that the generated glitch width at the POs is also small. Furthermore, the delay penalty incurred due to the increased load at the POs can be offset by increasing the supply voltage of the whole circuit which will have the additional advantage of reducing the generated glitch width at all interior nodes in the circuit (see generated glitch width variation with V_{DD} in Figure 28).

5.2 Soft Error Rate Monitoring

To control the soft error rate (SER) of the circuit under changing environmental conditions, the circuit SER has to be monitored by soft error monitoring circuitry. Two types of monitors can be used to estimate circuit SER:

5.2.1 Independent SER Sensors

Figure 30 shows an SER sensor design using inverter chains. If a glitch is generated at one of the nodes in the chain, the glitch will clock the D flip-flop, giving a rising edge at the Q output. Because the flip-flop resets itself after a delay, the glitch inside the chain results in a pulse at the Q output of the flip-flop of width equal to the delay of the delay element. The advantage of this setup over just using an inverter chain as the sensor is that the bigger pulse generated at the output guarantees that the signal will propagate to the OR gate without attenuation (so the glitch will be counted). There are two constraints for this structure to work. There should be even number of inverters in the chain and the delay of the delay element should be larger than possible glitch durations.

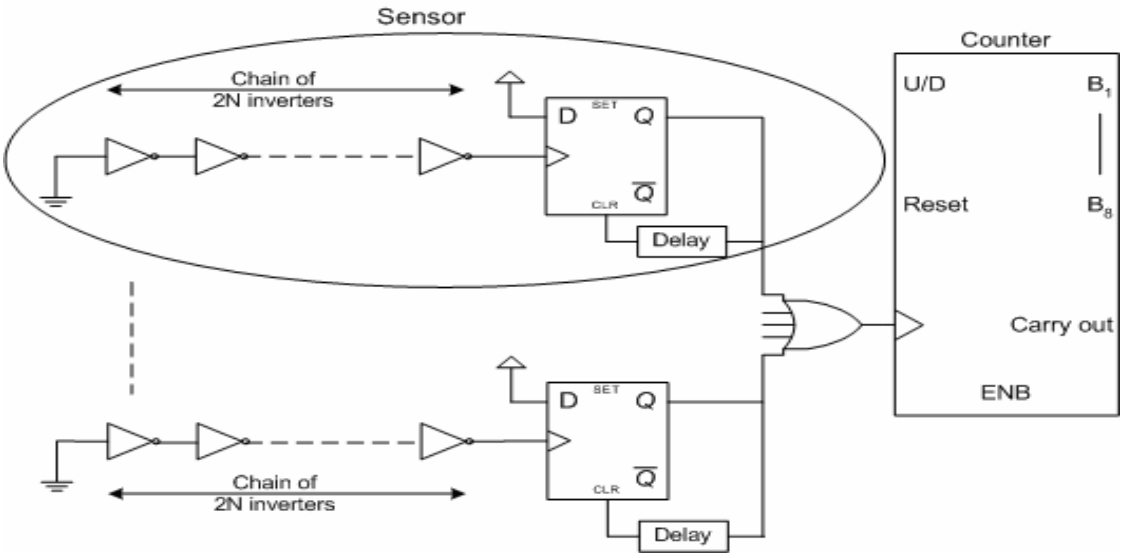

Figure 30. Inverter chain setup for monitoring SER.

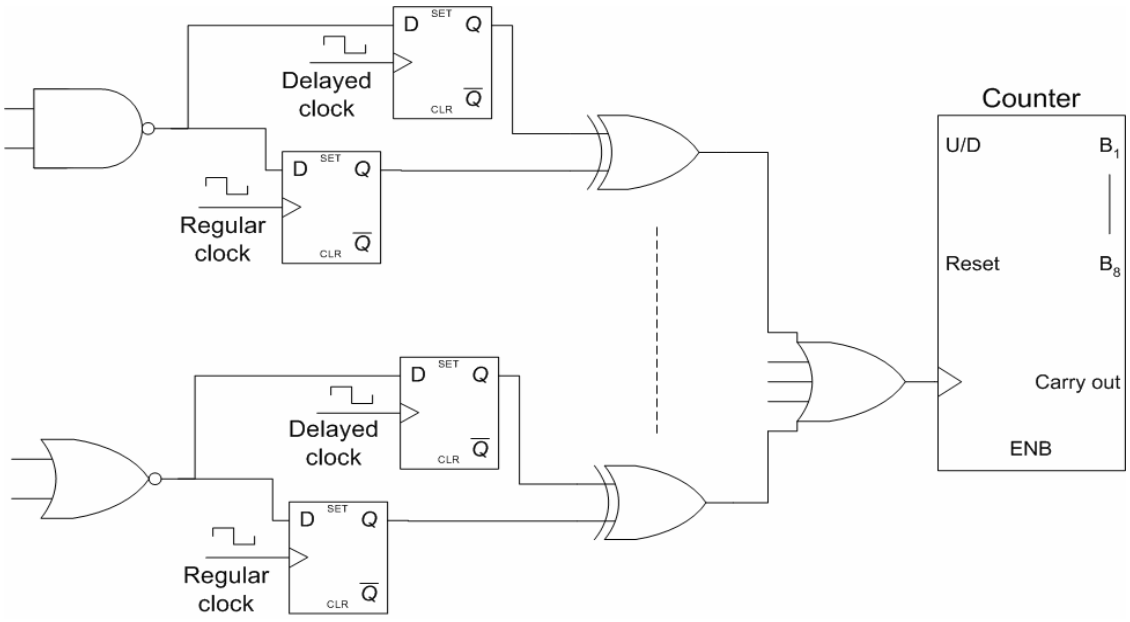

Figure 31. Shadow latch setup for monitoring SER.

An inverter chain does not have any logical masking. So, a glitch at any internal node of the chain will reach the clock input of the flip-flop (if the glitch is wide enough). These sensors have little area and static energy overheads. There is negligible dynamic energy overhead because the node voltages of the chains will not change unless there is a particle strike. The number of sensors to be used depends on the system area and the SER estimation precision desired. More chains should be used for larger circuits to track the system SER. Different inverter chains may have different glitch generation and propagation characteristics to sense different environmental conditions.

5.2.2 Embedded Concurrent Error Detectors

The circuit's primary outputs are sampled at two different times using shadow latches as done in [38][67]. These two values are XORed to determine if there is an error at that particular primary output. The XOR outputs are ORed and the number of soft errors in an

interval of time is counted to determine the circuit SER. The percentage of the outputs to put shadow latches will determine the precision of the estimated SER. If all of the outputs are sampled twice, SER obtained from this method will be equal to the actual circuit SER. This circuitry is shown in Figure 31.

Inverter chain sensors will give a measure of the amount of radiation the circuit is currently exposed to. They are easier than shadow latches to implement. Embedded sensors (shadow latches) give a measure of the soft errors observed in the circuit. Their design is more complicated than independent sensors. Short paths should be considered when determining the duration between two samples. If this duration is selected to be δ, the minimum path delay from the primary inputs to the primary outputs with shadow latches should be greater than δ. Both types of sensors can be used together to get a measure of both the radiation level of the environment and soft errors observed in the circuit. Shadow latch usage for error detection is studied in [71] extensively. The critical design parameter for designing systems with shadow latches is the delay between the clocks driving the regular latch and the shadow latch, δ. This delay determines the maximum duration of erroneous signal that can be detected by the shadow latch. As δ increases, the area/power overhead of the error detection system increases [71]. The dynamic soft error tolerance control strategy can be used with a fully checked (where every output is checked with a shadow latch) system by allowing the system to be designed with a small δ, reducing the overhead (especially power overhead) for normal mode of operation. Or the proposed system can be used with a partially checked system where the shadow latches are used just to sense the environmental flux and error detection/correction is done in software level. In this case, the control methodology will

reduce the runtime of the error detection/correction routine when the system enters high-flux environments by making the circuit more tolerant to particle strikes.

5.3 Circuit Soft Error Tolerance Estimation

The soft error tolerance of the circuit is found by considering the effects of a particle strike on the circuit's outputs for all the possible strike points. The possible strike points are taken to be the output nodes of all the gates in the circuit. A particle strike at a node will have different consequences depending on the input signal applied to the circuit. A SPICE level simulation for various input combinations will be very time consuming. A tool is generated to do the circuit soft error tolerance estimation in much less time then SPICE simulations and in reasonable accuracy. This tool is called Accurate Soft Error Tolerance Analyzer (ASERTA) [72].

ASERTA models a particle strike at a node as a current source injecting (or removing) a fixed amount of charge into (or from) that node. If the node is at low voltage, charge is injected into the node and if the node is at high voltage, charge is removed by the current source. The opposites of these two cases do can not cause a voltage glitch to be generated and are neglected. A SPICE look-up table is constructed for generated glitch width (because of charge injected at gate output) for different types of gates, fan-ins, sizes, V_{DD}s, V_{Th}s and load capacitances.

SPICE look-up tables are also constructed for delays, static energies, dynamic energies, output ramp and gate input capacitances for different types of gates, fan-ins, sizes, V_{DD}s, V_{Th}s, input ramps and load capacitances. ASERTA uses linear-interpolation inside the look-up tables to compute output values for arbitrary values of input parameters. Using look-up tables allows ASERTA to have better accuracy than analytical

models while still being much faster than SPICE. To estimate the soft-error tolerance of a circuit, ASERTA injects charge into every gate output, looks-up the generated glitch width from the table and then propagates the generated glitch to the primary outputs (POs) taking into account the effects of logical and electrical masking. The sum total of the widths of the glitches reaching the POs is taken as a measure of the "unreliability" of the circuit. How ASERTA models logical, electrical, and latching-window masking is described next.

5.3.1 Estimating the Logical Masking

Since actual signal values are not known, for every node ASERTA calculates the probability that there is at least one sensitized path from that node to a primary output. Calculation of the sensitization probability values from the input signal statistics is easy for circuits which do not have reconvergent fan-out. Sensitization probabilities for such circuits can be calculated as in [70]. However, finding the values for circuits with reconvergent fan-out is an NP-complete problem [73]. ASERTA uses zero delay simulation of the circuit with 10000 random inputs applied (as in [68]) to compute the probability, P_{ij}, that there is at least one path sensitized from output of gate i to primary output j. For primary output j, P_{jj} is 1. The static probability, p_i, of a node i being at logic 1 is obtained for all nodes using a commercially available tool, Synopsys Design Compiler, given a static probability of 0.5 at the primary inputs.

For all successor gates s of gate i, the probability that a glitch at i will be able to propagate through gate s to primary output j is calculated as follows:

$$\pi_{isj} = \frac{S_{is} \cdot P_{ij}}{\sum_{k \in \Psi} S_{ik} \cdot P_{kj}} \quad (9)$$

where Ψ is the set of successors of gate i and S_{is} is the probability that gate s is sensitized to gate i (i.e. all other inputs of gate s have non-controlling values). S_{is} can be obtained by multiplying together the static probabilities of the other inputs being 1/0 for a AND/OR gate. Note that π_{isj} is not taken to be just $S_{is} \cdot P_{sj}$ since π_{isj} should have the property that $\sum_{k \in \Psi} \pi_{ikj} \cdot P_{kj} = P_{ij}$. Also note that π_{isj} is an approximation to the actual probability value since in circuits with reconvergent fan-out, the probability that gate s is sensitized to gate i conditions the probability of gate s having a path sensitized to a primary output.

5.3.2 Estimating the Electrical Masking

ASERTA computes the expected output glitch width, W_{ij}, at primary output j for generated glitch width, w_i, at gate i. To do this efficiently in one pass over the circuit, for every gate, the expected output glitch widths, WS_{ijk}, for 10 sample glitch widths, ws_k (k between 1 and 10) are computed.

The output glitch widths are computed for all gates in reverse topological order (i.e. from POs to PIs) as follows:

(i) Let current gate be i.

(ii) If gate i is a primary output, set $WS_{iik}=ws_k$ for all k.

Set $WS_{ijk}=0$ for all other primary outputs j.

Also, since gate is primary output, it will propagate generate glitch width, w_i, directly. Hence, set $W_{ii}=w_i$ and $W_{ij}=0$ for all other primary outputs j.

(iii) If gate i is not a primary output, for all sample glitch widths, ws_k:

For all successors s of gate i:

Let d_s be the delay of gate s looked up from the SPICE tables.

Calculate the glitch width, wo_{sk}, propagated to the output of gate s for input width of ws_k using Equation 8.

For each primary output j, look up the expected output glitch width, WE_{sjk}, for generated glitch width of wo_{sk} from the table of expected output glitch widths for gate s, linearly interpolating if necessary.

Finally, Let $WS_{ijk} = \sum_{s \in \Psi} \pi_{isj} \cdot WE_{sjk}$

(iv) Compute W_{ij} by looking up the table of expected output glitch widths, WS_{ijk}, computed in step (iii), for a generated glitch width of w_i, again linearly interpolating if necessary. Now process the next gate.

At the end of this procedure, expected output glitch widths, W_{ij}, at primary output j for generated glitch width, w_i, for every gate i are known. The complexity of the procedure is O(V+E), where V is the number of gates and E is the number of circuit edges.

Lemma 1: For a very wide glitch ww_i generated at output of gate i, the above procedure correctly computes the expected output glitch width at primary output j as $WW_{ij} = ww_i \cdot P_{ij}$, if it is assumed that ww_i is one of the sample glitch widths used.

Proof: Since the generated glitch is very wide, it will pass through all gates on any path from i to j without attenuation. WS_{jj1} is correctly computed as ww_i at primary output j. Assume that WS_{rj1} is correctly computed for all successor gates r of a gate p between i and j as $ww_i \cdot P_{rj}$. Then, the expected width WS_{pj1} will be computed as:

$$WS_{pj1} = \sum_{r \in \Psi} \pi_{prj} \cdot WS_{rj1} = \sum_{r \in \Psi} \pi_{prj} \cdot ww_i \cdot P_{rj}$$
$$= ww_i \cdot \sum_{r \in \Psi} \pi_{prj} \cdot P_{rj} = ww_i \cdot P_{pj}$$

where WS_{rj1} can be used instead of WE_{rj1} because ww_i is wide enough to propagate through gate r without attenuation. By induction, WS_{ij1} is also computed as $ww_i \cdot P_{ij}$. Since ww_i is the first sample glitch width, WS_{ij1} is WW_{ij}. □

5.3.3 Estimating the Latching-Window Masking

A glitch must arrive within the setup and hold times of the latch at the primary output to be captured. Since the exact time of the particle strike is unknown, it can be assumed to be uniformly distributed within the clock cycle. The probability of a glitch being captured by a latch is directly proportional to its duration. Hence, by summing up the expected output glitch widths, W_{ij}, for all primary outputs j, the total contribution of gate i to the circuit unreliability is obtained.

Figure 32 shows the unreliability numbers, U_i, for the gates in ISCAS'85 benchmark circuit "c432" calculated by ASERTA plotted along with values calculated by SPICE for 70nm technology node. In SPICE, the unreliability was computed by applying 50 random input vectors, injecting charge at every gate output i and summing the glitch widths at the primary outputs. Only the nodes that were at most five levels deep from the POs are plotted. It is seen that there is close matching. The correlation between the two series was computed to be 0.96. For the ISCAS'85 benchmark circuits, an average correlation of 0.9 was obtained. The run time of ASERTA was less than ten seconds for all the ISCAS'85 benchmark circuits simulated in this work.

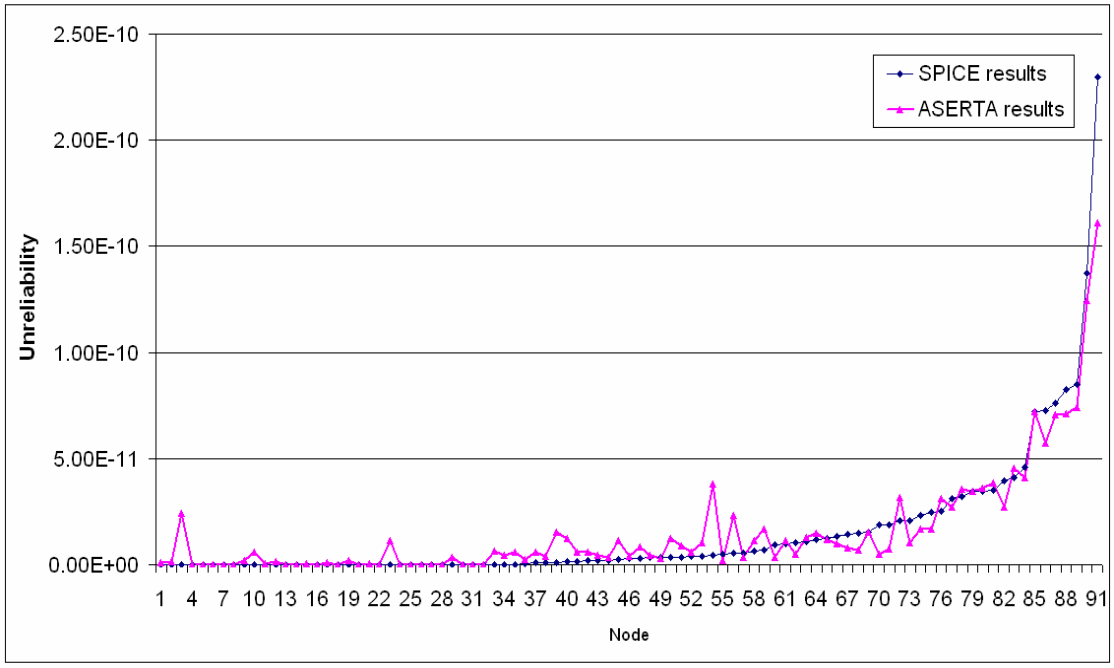

Figure 32. Unreliability values obtained by SPICE and ASERTA for nodes in c432.

5.4 Dynamic Soft Error Tolerance Control

Depending on the measured SER and the power/performance needs of the circuit at that time, circuit supply voltage and threshold voltages and output capacitances of select nodes are adjusted. At the core of the dynamic SER control is the dynamic output capacitance modulation via capacitor banks controlled by transistor switches as shown in Figure 33. As explained in Section 5.1, increasing the output capacitance of a node will improve both its glitch generation and propagation characteristics. If the output capacitances of all the primary outputs are increased, the glitches generated in prior levels will be attenuated and also the voltage magnitude of glitches generated at the primary outputs will reduce, dramatically improving the system soft error tolerance. The cost is significant increase in system delay if supply voltage is not increased to compensate for the delay increase, so this configuration is not appropriate for normal mode of operation.

Figure 33 shows the schematic of the proposed structure. When the switch is "ON", node capacitance is increased by C. If the parasitic capacitance of C (to the ground) is small enough, this structure will add an additional capacitance of only $\left(C \cdot C_{par}\right) / \left(C + C_{par}\right)$ (two capacitances in series) to the node N, which is smaller than the parasitic capacitance of a single transistor. Metal-insulator-metal (MIM) capacitors can be used for this purpose because of their high reliability and low parasitic capacitances [74]. However, it is possible to get much higher capacitance per unit area by using MOS capacitors [75]. The penalty will be increased parasitic capacitance in the normal mode of operation. If MOS capacitor is used, the order of the capacitor and the MOS switch should be reversed in Figure 33 (i.e. the switch should be connected to the node) to limit the parasitic capacitance effect on the node.

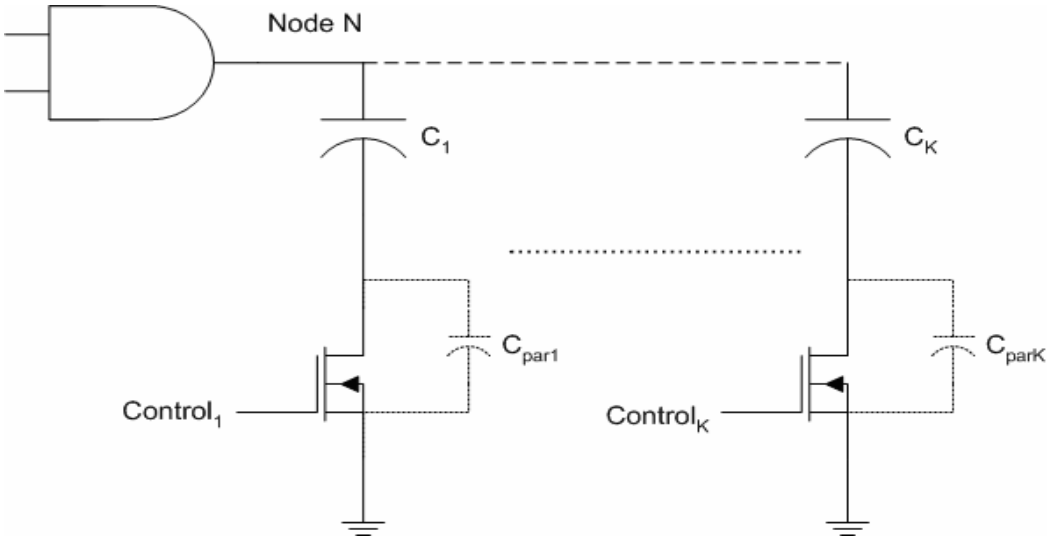

Figure 33. Node capacitance control using NMOS switches (if MOSFET capacitors will be used, switch and capacitor should be swapped).

Typically, the SER of the system is not expected to change significantly in a short amount of time during its operation. It is possible to use either discrete supply voltages in the control scheme or use continuous V_{DD} modulation. The developed control technique uses the former approach.

In addition to modulating supply voltage, it is possible to also decrease threshold voltages to speed up some of the gates. Such gates will be fabricated with low threshold voltage and reverse body bias will be applied during normal operation. This reverse bias is "released" to lower their threshold voltages when necessary. The number of gates to be speeded up is chosen to be at most 20% of the total gate count to limit the increase in static energy consumption. Low threshold voltage assignment is done as follows:

1- Sort the nodes according to their slacks.

2- Assign low threshold voltage to the node with the lowest slack

3- If 20% of the gates are assigned low threshold voltage, stop. Else go back to Step 1.

This algorithm is used to test the potential improvement of SER with variable threshold voltage. As it is seen in Section 5.6, the improvement of SER with variable threshold voltage is very small and the energy consumption overhead is intolerable. Therefore, dynamic threshold voltage modulation is not used in the following feedback control strategy.

Feedback Control Strategy: For a system where no delay increase is allowed, a simple control strategy can be implemented as follows. If the output from the embedded concurrent error detectors (ECEDs) indicates increased SER, increase the supply voltage until enough delay slack is created to be able to switch in an additional capacitor from the capacitor bank. If the output from the ECEDs indicates decreased or constant SER,

switch-out the capacitance first, and then reduce the supply voltage to take up the delay slack created. To avoid oscillation of the system supply voltage, switching-out the capacitance and reducing the supply voltage is done after a predetermined duration. This scheme guarantees that the circuit always operates within the delay constraint, while adapting to the radiation environment. Also, it guarantees that the system will return to its normal mode of operation when the environmental conditions go back to normal.

5.5 Implementation of the Control Technique

To verify the effectiveness of the dynamic SER control methodology, SPICE look-up tables for delay, input capacitance, dynamic energy consumption, static energy consumption, and output signal ramp were generated for 2 to 4 input NAND and NOR gates and inverters of various sizes, output capacitances, threshold voltages, supply voltages, input signal ramps, and body bias voltages. These models were used to evaluate system performance and energy consumption for various operating modes. Average energy consumption of a gate during a clock cycle (=delay of the circuit) is obtained by using Equation 6.

ISCAS'85 benchmark circuits were synthesized using a generic library with only 2 to 4 input NAND and NOR gates and inverters using Synopsys Design Compiler. Also, static probabilities and switching activities of the internal nodes were obtained by using Synopsys Design Compiler for a switching activity of 0.1 and static probability of 0.5 at the primary inputs.

Soft-error tolerances of the circuits were obtained by using ASERTA. In a real environment, the injected charge values differ from strike to strike but to simplify reporting, only data for a single value of charge (12fC) is reported. For simplicity, the

current waveform is selected to be of a trapezoid shape with duration of 24ps and with 4ps rise time, 16ps of fall time. This captures the fast rising and slow falling behavior of collection waveform given in [76] by the following equation:

$$I(t) = \frac{2}{T \cdot \sqrt{\pi}} \cdot \sqrt{\frac{t}{T}} \cdot \exp\left(\frac{-t}{T}\right) \tag{10}$$

where T is a process dependent parameter.

For every node, the unreliability value is obtained using ASERTA. Unreliability is the expected value of the total erroneous signal duration at the outputs if a particular node is struck by a particle. The "unreliability" value for a node incorporates the effects of logical masking and electrical masking. Latching-window masking is also incorporated in the "unreliability" value for a node because if the total duration of erroneous logic value at the primary outputs is longer, the probability of a glitch being captured by a latch increases. The "unreliability" of the circuit is the sum of "unreliability" values of all nodes. The expected erroneous signal duration at the outputs if a single gate is struck by a particle is found by dividing the circuit unreliability by the total number of gates in the circuit.

5.6 Experimental Results for Dynamic Soft Error Tolerance Control Scheme

ISCAS'85 benchmark circuits were optimized for static energy consumption by dual threshold voltage assignment (0.2V and 0.3V) using the algorithm in [47]. These static energy optimized circuits were used as the base case to get more realistic results. Without this step, the improvements will be boosted as the improvements depend on the slacks of the nodes in the circuits, but the results would not reflect real scenarios since most of the

digital systems today are manufactured with dual threshold voltages to reduce static energy consumption. As explained in Section 5.5, the gates which are selected for speeding up have low threshold voltage (0.1V) and their body bias voltages are modified so that they have effective threshold voltage of 0.2V in the normal mode of operation. Primary outputs are assumed to have 3fF of load capacitance in the normal mode. A minimum size inverter in 70nm technology [59] has a delay of ~15ps and an input capacitance of 0.6fF.

Since some primary outputs have huge slacks, the maximum amount of capacitance to be added to a single node is limited to a maximum amount to limit the area increase. As this limit is increased, the soft error tolerance usually increases at the expense of more area overhead. It is observed for a few example circuits that more extra capacitance may reduce the soft error tolerance. This is because of the first rising then falling generated glitch width characteristics of gates with respect to output capacitance (Figure 28). If MOS capacitances are used, total capacitance of the circuit can be used as a measure of area. The results are reported using this approach.

Figure 34 shows the energy/unreliability trade-offs for benchmark circuit "c432" where the capacitance increase is limited to 20fF. Area overhead in this case is 8%. Three different trade-offs are shown in the figure:

1. C_{load} scaling only: The effect of using load capacitances giving 0%, 10% and 20% delay penalty is shown. 1A is the case where only 2 V_{th}s (0.2 and 0.3) are used and 1B is the case where 3 V_{th}s (0.1, 0.2, and 0.3) are used.
2. V_{DD} and C_{load} scaling: The effect of scaling supply voltage and C_{load} simultaneously keeping circuit delay constant is shown (2A and 2B).

V_{DD} scaling only: The effect of using supply voltages of 1, 1.2 and 1.4 volts is shown (3A and 3B).

It is seen that the low V_{Th} assignment has no significant effect on the circuit unreliability while causing energy to go up drastically. Hence it can be dropped as an optimization variable. It is seen from Figure 34 that the combination of V_{DD} and C_{load} scaling gives much better soft error tolerance/energy trade-off compared to only V_{DD} or only C_{load} scaling. Unreliability values were obtained as explained in Section 5.5.

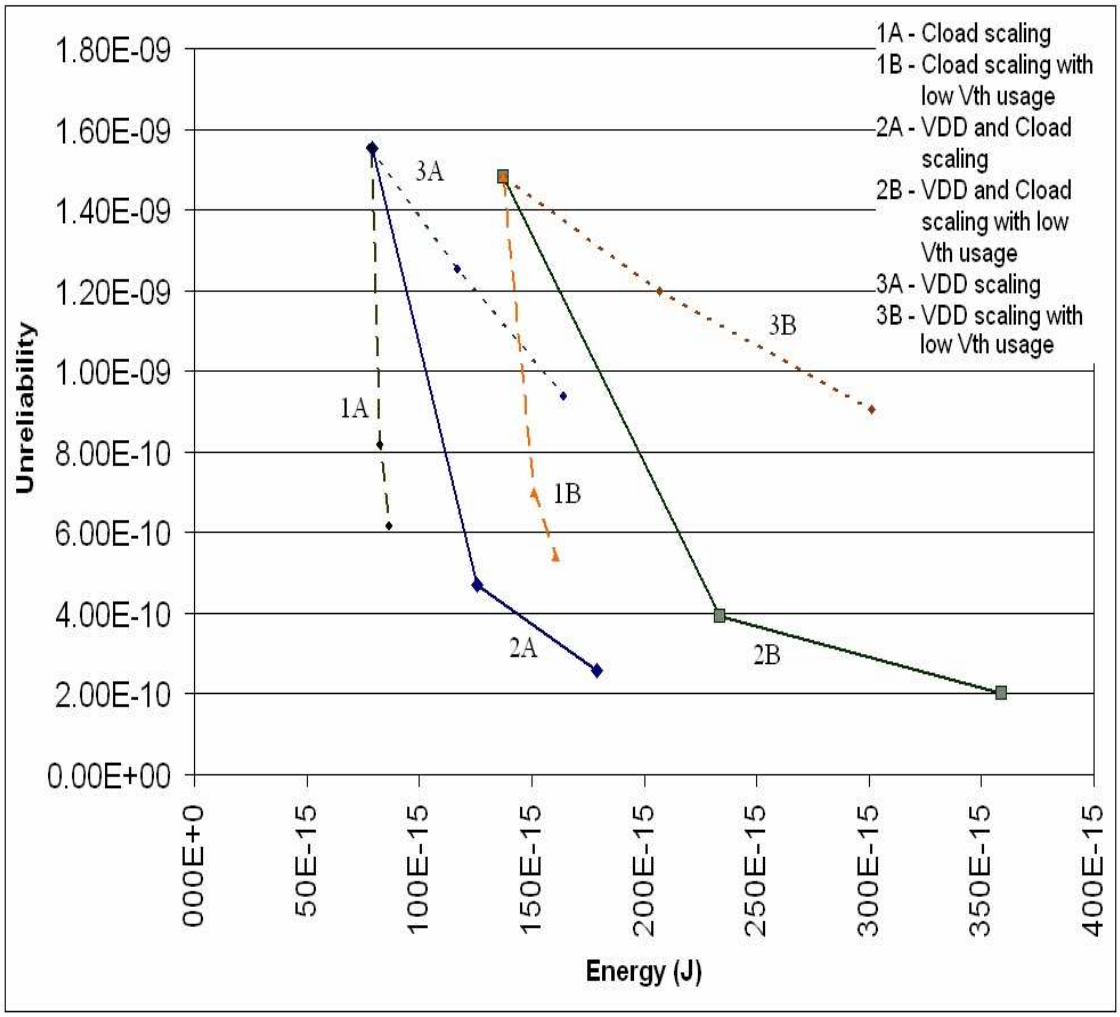

Figure 34. Effect of V_{DD} scaling, C_{load} scaling and V_{th} scaling on unreliability and energy of c432.

Table 6 lists the unreliability, energy, and delay values for various ISCAS'85 benchmark circuits for (1) normal mode, (2) 1.2V supply voltage, no delay overhead, and (3) 1.4V supply voltage, no delay overhead. Optimization is run for (a) 30fF limit on capacitance increase and for (b) the case where capacitance increase is not limited. Area overheads are shown in the table. Circuit unreliability values are obtained using ASERTA by injecting 12fC to all the nodes. A TSMC wire load model was used for net length estimation for different fan-outs. Coupling capacitances and ground plane capacitances in a typical VLSI layout are taken into account when computing the distributed capacitance per unit length [77]. The width and spacing between interconnects was taken to be 0.1 micron, and the thickness of interconnect and dielectric was taken to be 0.2 micron respectively. The dielectric constant was taken to be 3.9 (SiO_2).

The control strategy is simulated for c432 for a hypothetical scenario using Matlab. The circuit is assumed to be located in an airplane which takes off from ground and ascends to 10000 feet linearly in half an hour. Only neutrons are considered to be causing soft errors. The circuit has three modes of operation. These are: (1) normal mode, (2) 1.2V supply voltage, no delay overhead, and (3) 1.4V supply voltage, no delay overhead. Control strategy is as explained in Section 5.4. In [78], the neutron flux is given to be two orders of magnitude larger at aircraft altitudes compared to ground level. For simulation purposes it is assumed that a neutron hits a circuit node 5 times in every 10 million cycles at ground level and 500 times in every 10 million cycles at 10000 feet. Flux is scaled linearly with altitude during the flight. Clock period is taken to be 0.5 ns and the error threshold at which the system will go into more tolerant mode is selected to be 1 error in 10 million cycles. Simulation of a particle strike is done using the average

Table 6. Results of dynamic soft error tolerance control scheme for (a) 30fF limit and (b) no limit on added capacitance.

Circuit / Mode		Unreliability (a,b) (ns)		Energy (a,b) (fJ)		Area (a,b)	Delay (ns)
c432	(1)	1.53	1.00X	91.3	1.00X	1.00X	0.53
	(2)	0.42, 0.43	0.27X, 0.28X	150, 150	1.64X, 1.64X	1.08X, 1.09X	
	(3)	0.17, 0.17	0.11X, 0.11X	222, 224	2.43X, 2.45X	1.11X, 1.12X	
c499	(1)	5.04	1.00X	353	1.00X	1.00X	0.41
	(2)	2.00, 2.00	0.40X, 0.40X	557, 557	1.58X, 1.58X	1.13X, 1.13X	
	(3)	0.34, 0.35	0.07X, 0.07X	815, 815	2.31X, 2.31X	1.20X, 1.20X	
c1908	(1)	4.52	1.00X	256	1.00X	1.00X	0.62
	(2)	0.75, 0.68	0.17X, 0.15X	441, 445	1.72X, 1.74X	1.23X, 1.26X	
	(3)	0.20, 0.05	0.04X, 0.01X	654, 681	2.55X, 2.66X	1.26X, 1.37X	
c2670	(1)	6.13	1.00X	310	1.00X	1.00X	0.38
	(2)	1.71, 1.66	0.28X, 0.27X	530, 610	1.71X, 1.97X	1.18X, 1.46X	
	(3)	0.71, 0.71	0.12X, 0.12X	767, 896	2.47X, 2.89X	1.20X, 1.53X	
c3520	(1)	4.47	1.00X	480	1.00X	1.00X	0.75
	(2)	0.99, 1.07	0.22X, 0.24X	794, 835	1.65X, 1.74X	1.09X, 1.17X	
	(3)	0.33, 0.40	0.07X, 0.09X	1185, 1282	2.47X, 2.67X	1.12X, 1.22X	
c5315	(1)	17.05	1.00X	714	1.00X	1.00X	0.60
	(2)	2.97, 2.05	0.17X, 0.12X	1254, 1449	1.76X, 2.03X	1.28X, 1.68X	
	(3)	0.95, 0.52	0.06X, 0.03X	1822, 2171	2.55X, 3.04X	1.32X, 1.81X	
c7552	(1)	11.22	1.00X	1037	1.00X	1.00X	0.52
	(2)	2.46, 2.24	0.22X, 0.20X	1712, 1753	1.65X, 1.69X	1.12X, 1.19X	
	(3)	0.54, 0.45	0.05X, 0.04X	2521, 2603	2.43X, 2.51X	1.16X, 1.24X	
ARM inst decoder	(1)	0.92	1.00X	5.57	1.00X	1.00X	0.32
	(2)	0.52,0.52	0.57X, 0.57X	10, 10	1.8X, 1.8X	1.23X, 1.23X	
	(3)	0.26,0.26	0.28X, 0.28X	15.7, 15.7	2.75X, 2.75X	1.36X, 1.36X	

unreliability value per node for the circuit. This value represents the expected value of

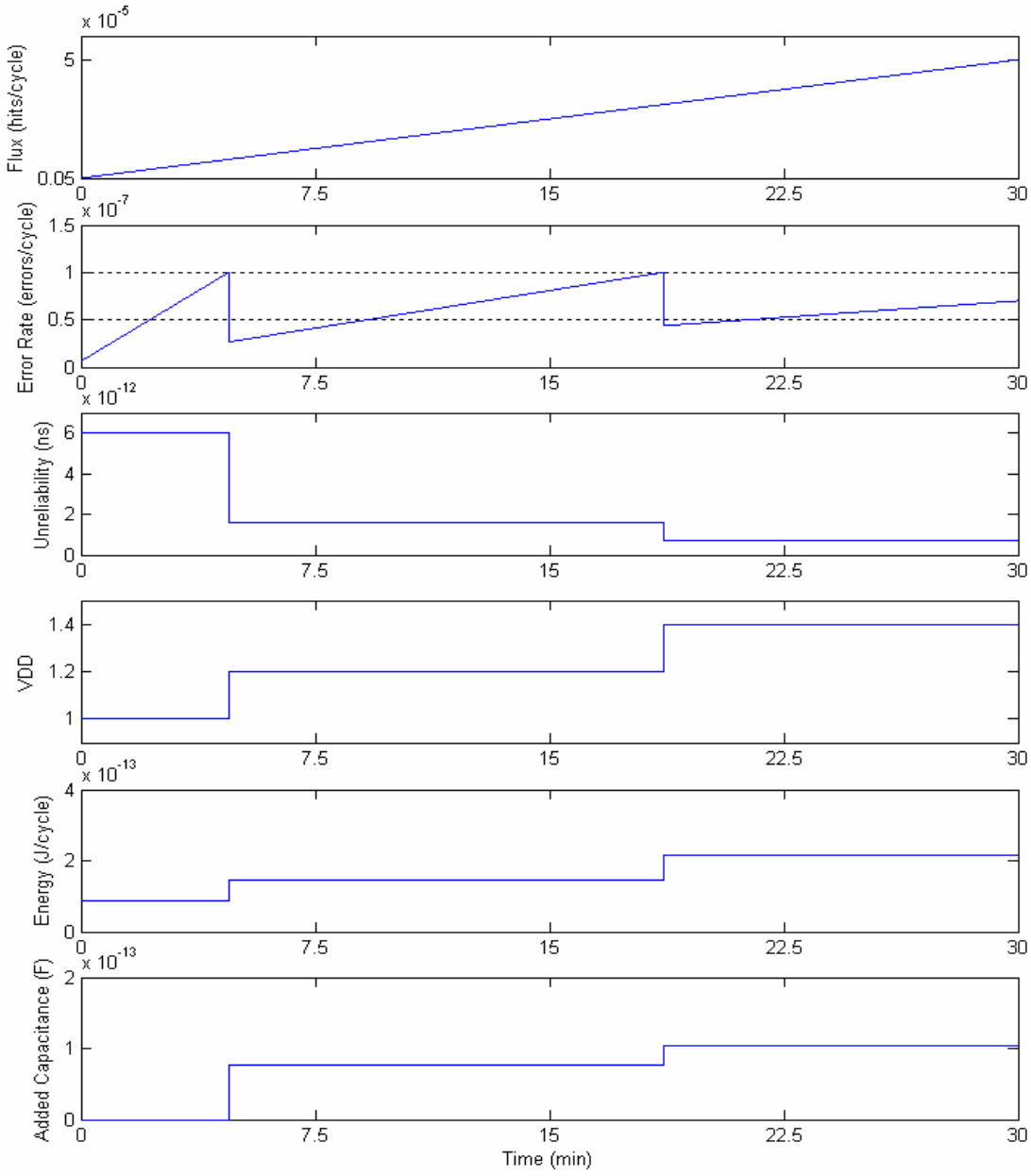

Figure 35. Matlab simulation of control methodology on c432 assuming the system is ascending to 10000 feet from ground level in 30 minutes.

total duration of the erroneous signal at the circuit outputs when a particle hits a node. This value divided by the clock period gives the probability of an error being captured by the latch. The extra capacitances are limited to be 30fF per output. The values for energy

consumption and unreliability for c432 are given in Table 6. The circuit has 267 nodes so the average unreliability values for operating modes for a single node are as follows: (1) Normal mode: 0.006ns, (2) 1.2V: 0.0016ns, and (3) 1.4V: 0.0007ns. This system has an area overhead of 11%. Figure 35 shows the Matlab simulation of this system.

CHAPTER 6

CONCLUSION AND FUTURE RESEARCH

In the closing chapter of this thesis, the major contributions of the work are summarized along with the inquiry of some prospective directions in which this research can proceed.

6.1 Conclusions

Technology scaling trends introduces new challenges to the design community. Power management, managing design complexity, improving soft error tolerance, and managing process variations are a few of the most important problems that the chip designers should address. This work focused on two of these problems: low power design and soft error tolerance. Multiple supply and threshold voltages are utilized to reduce the power consumption of a digital block without changing the delay of that block. Variable supply and threshold voltages together with adjustable capacitances are used to provide digital systems a dynamic soft error tolerance control scheme. The contributions of this research can be summarized as follows:

- Two domino logic gate design styles that use NMOS transistors to do precharging are introduced. The use of NMOS transistors reduces the voltage swing of the internal nodes, which in turn reduces the energy consumed per switching in the domino gates. These gates are used together with the regular domino gates to design pseudo dual supply voltage domino logic blocks. The fast/high energy regular domino gates are used on the critical paths and the slow/low energy NMOD pull-up domino gates are used on the off-critical paths. The algorithm for

power optimization guarantees that the resulting circuit has the same delay as the initial circuit, but it has in average 20% less energy consumption. This scheme does not need as second supply voltage.

- A level shifting logic gate design is introduced to be used in multiple supply voltage static CMOS circuit designs. The high leakage caused by the positive gate to source voltage for the PMOS transistor when a high voltage gate is driven by a low voltage gate is reduced by using a higher magnitude threshold voltage in those PMOS transistors. Doing this reduces the (V_{GS} - $|V_{Th}|$) term in the exponent in the leakage current equation, reducing the leakage current. This structure is much faster and more efficient than a gate followed by a regular level shifter, which does not perform any logic operation. Using the level shifting logic gates in a multiple supply system increases the energy consumption reduction of the used scheme by reducing the energy, delay, and area overheads of the regular level shifters.
- The developed level shifting logic gates are used in a dual supply voltage assignment algorithm. The magnitude of the threshold voltage of the PMOS transistors in the high voltage gates is increased if they are driven by a low voltage gate. The algorithm assigns low supply voltage to the gates on the off-critical paths. This reduces the total energy consumption of the block while not changing its delay. 20% average energy consumption reduction is achieved for the optimized benchmark circuits.
- The characteristics of single gates associated with particle strikes are examined in great detail. It is shown that a gate has two important characteristics that affect the

soft error tolerance of the circuit the gate is in. These are named glitch generation and glitch propagation characteristics of the gate. Glitch generation characteristics are associated with the effects of the gate's properties (supply voltage, threshold voltage, size, output capacitance …etc) to the duration of the generated glitch when a particle hits the gate's output. Glitch propagation characteristics are associated with their effects to the duration of the glitch propagated to the output of the gate, when an already generated glitch signal is present in its input. It is observed that these characteristics usually work against each other. To improve one, the other one is usually compromised. The only exception to that is the effect of the output capacitance of the gate. It is observed that, after a certain value, rising output capacitance improves both of these characteristics for a gate.

- The observations made for a single gate are used to develop a dynamic soft error tolerance control scheme for digital circuits. This scheme uses soft error sensors to gather information about the environmental conditions. Depending on the required tolerance level, the output capacitances of the gates driving the flip-flops and the system supply voltage is varied. If the particle flux is increasing, the supply voltage of the system is increased. The slack generated by this increase is taken by increasing the output capacitances of the gates driving the flip-flops. By improving the glitch generation and propagation characteristics of the output gates, the soft error tolerance of the whole circuit is improved significantly. When the environmental conditions go back to normal, first the additional capacitances are switched out, and then the supply voltage level is reduced. This scheme is especially beneficial for systems those change environment during their operation.

The energy consumption and delay overheads of this scheme are negligible during normal mode of operation. Up to 100X improvement in soft error tolerance of the circuits is obtained with only 2.5X increase in energy consumption.

- A fast circuit soft error tolerance estimator (ASERTA) is developed. This tool runs orders of magnitude faster than SPICE and it estimates the soft error tolerance of the circuit accurately. It is suitable to be used for comparing design alternatives in terms of soft error tolerance.

6.2 Future Directions

Several possible future research directions based on this work are summarized below:

- A static dual supply voltage assignment scheme is developed. A novel level shifting logic gate is utilized between the power supply boundaries when a low supply gate is driving a high supply gate. This level shifter strategy can be used in a dynamic supply voltage modulation when supply voltages of power islands are changed depending on the circuit's performance and power needs. By applying variable body back-biasing, the threshold voltages of the boundary gates can be changed depending on the voltages assigned to the power domains bordering these gates.
- A dynamic soft error tolerance improvement scheme is developed. This scheme uses dynamic supply voltage modulation. This scheme can be combined with a dynamic power management scheme to utilize dynamic supply voltage modulation capability for power management as well.

APPENDIX A

TOPOLOGICAL SORTING

Topological sort of a directed acyclic graph (DAG) is a linear ordering of nodes in that graph such that for any gate G_i, all the nodes that are driven by G_i are located after G_i. The algorithm for topological sorting uses depth-first search. The run time is linear with the number of nodes plus the number of edges. The algorithm is as follows [79]:

```
Insert the dummy primary input, PI_d, to a queue, Q.
while Q is nonempty
   remove a node n from Q
   insert n into topologically sorted list, TSL.
   for each node m with an edge e from n to m
      remove edge e from the graph
      if m has no other incoming edges
         insert m into Q
```

Figure 36. Algorithm for topological sorting.

TSL will have the topologically sorted ordering of the input DAG after the algorithm finishes. There can be more than one orderings for a DAG that satisfies the conditions for being topologically sorted orderings.

APPENDIX B

BUBBLE PUSHING AND DUPLICATION[1]

Domino logic offers improvements over static logic in circuit area, and speed. However, domino logic can not implement inverting functions. The inherently non-inverting nature of domino gates requires the implementation of logic network without inverters. This inverter-free logic constraint is a fundamental constraint for implementing logic functions with domino gates.

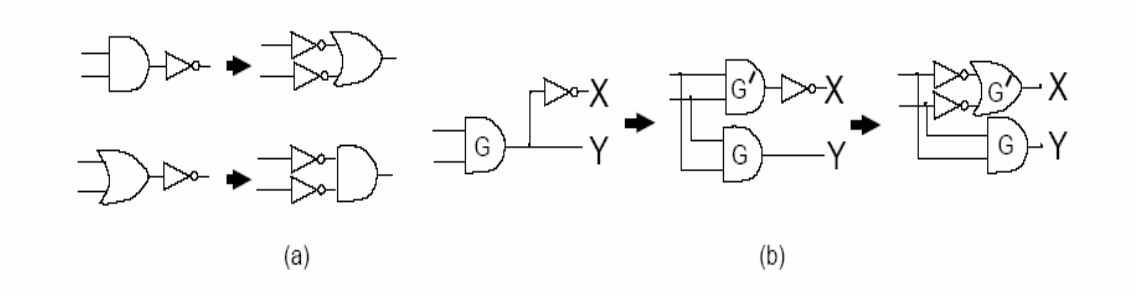

Figure 37. Propagating an inverter through logic gate (a) without fanout (b) with fanout.

Because of the non-inverting nature of the domino gates, a domino logic block should be designed where all the necessary inversions take place at the primary inputs or at the primary outputs of the block, i.e., at the clock phase boundaries. This way, the inverters can be absorbed in registers. Thus the first step in domino logic synthesis is to make the logic inverter-free. The standard approach is to convert the technology independent logic into AND, OR, and NOT gates only. Subsequently, the inverters can

[1] [80] is used as a reference for this appendix.

be propagated towards the inputs by applying simple De Morgan's laws as shown in Figure 37(a) starting at the primary outputs. If an inverter is trapped (An inverter is said to be trapped at a fanout net N_i, if it cannot be propagated back towards primary inputs without duplicating the logic gate that is feeding the fanout net N_i.) at the fanout of a gate G, then gate G is duplicated for implementing both positive and negative signals and the inverter is pushed backward as shown in Figure 37(b). This procedure does not increase the number of logic levels in the circuit and the area is at most doubled [81]. This procedure transforms the given logic network into an inverter-free logic network with inverters at its primary inputs only. Figure 38 shows the steps of this process on an example circuit.

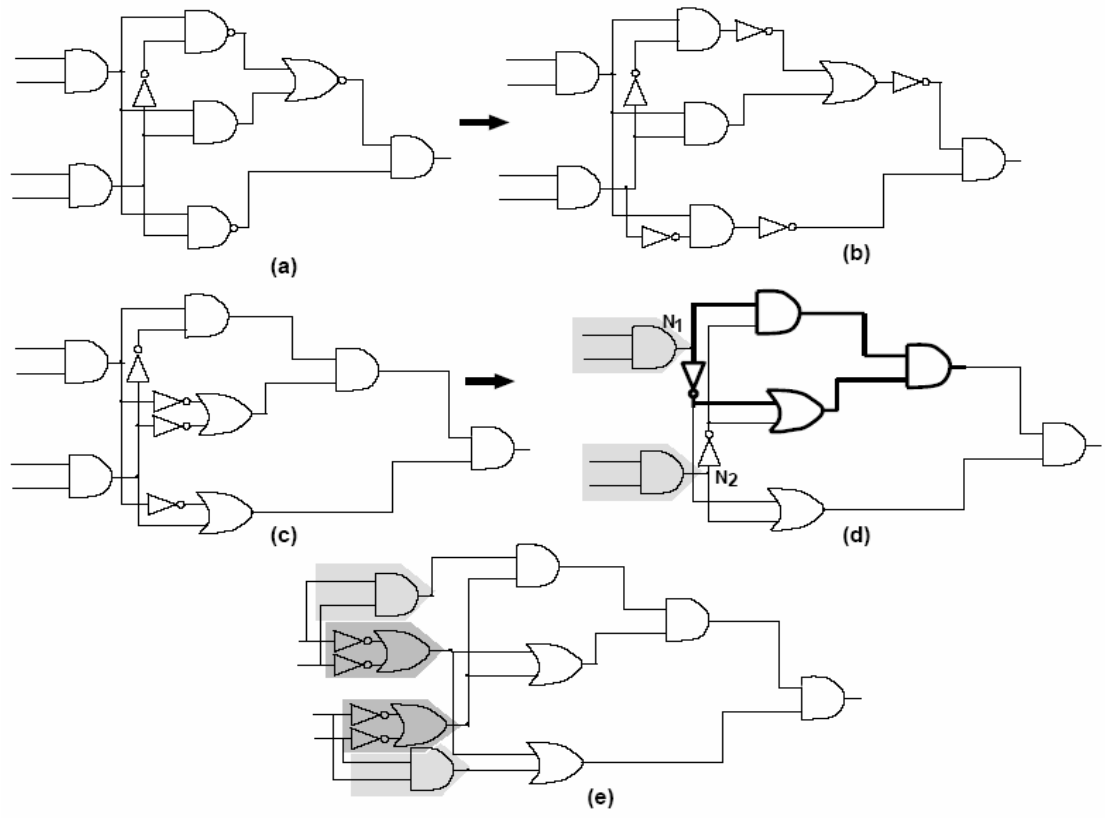

Figure 38. Backward propagation of inverters to obtain inverter-free logic.

Explanation of the steps in Figure 38 is as follows:

(a) Logic with NAND/NOR/AND/OR/NOT gates

(b) Logic with AND/OR/NOT gates

(c) Propagating inverters back towards primary inputs without any logic duplication

(d) Combining the inverters trapped at intermediate fanouts

(e) Inverter-free logic after propagating the inverters back towards primary inputs with logic duplication.

This procedure is implemented in C++. Benchmark circuits are first compiled by Synopsys Design Compiler to contain only AND, OR, and NOT gates, then this procedure is run to obtain the domino implementation for the circuit.

REFERENCES

[1] 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, pp. 616-29, 2001.

[2] V. Sundararajan and K. K. Parhi, "Synthesis of Low-Power CMOS VLSI Circuits using Dual Supply Voltages," *ACM Design Automation Conference*, New Orleans, 1999. pp. 72-75.

[3] K. Usami, K. Nogami, M. Igarashi, F. Minami, Y. Kawasaki, T. Ishikawa, M. Kanazawa, T. Aoki, M. Takano, C. Mizuno, M. Ichida, S. Sonoda, M. Takahashi, and N. Hatanaka, "Automated low-power technique exploiting multiple supply voltages applied to a media processor," *Proceedings of the 1997 IEEE Custom Integrated Circuits Conference*, May 5-8 1997, Santa Clara, CA, USA, 1997. pp. 131-134.

[4] K. Usami and M. Horowitz, "Clustered voltage scaling technique for low-power design," *Low Power Design Symposium*, 23-26 April 1995, Dana Point, CA, USA, 1995. pp. 3-8.

[5] M. Donno, L. Macchiarulo, A. Macii, E. Macii, and M. Poncino, "Enhanced clustered voltage scaling for low power," *GLSVLSI '02. Proceedings of the 12th ACM Great Lakes Symposium on VLSI*, 18-20 April 2002, New York City, NY, USA, 2002. pp. 18-23.

[6] Y. S. Dhillon, A. U. Diril, A. Chatterjee, and H. S. Lee "Algorithm for achieving minimum energy consumption in CMOS circuits using multiple supply and threshold voltages at the module level," *International Conference on Computer Aided Design*, 2003. pp. 693 - 700.

[7] J.-M. Chang and M. Pedram, "Energy minimization using multiple supply voltages," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 5, pp. 436-443, 1997.

[8] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, "Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits," *Proceedings of the IEEE*, vol. 91, pp. 305-327, 2003.

[9] M. Hirabayashi, K. Nose, and T. Sakurai, "Design methodology and optimization strategy for dual-V/sub TH/ scheme using commercially available tools," *ISLPED'01: Proceedings of the 2001 International Symposium on Low Power Electronics and Design*, 6-7 Aug. 2001, Huntington Beach, CA, USA, 2001. pp. 283-6.

[10] N. Tripathi, A. Bhosle, D. Samanta, and A. Pal, "Optimal assignment of high threshold voltage for synthesizing dual threshold CMOS circuits," *Proceedings of*

14th International Conference on VLSI Design, 3-7 Jan. 2001, Bangalore, India, 2001. pp. 227-32.

[11] T. Karnik, B. Bloechel, K. Soumyanath, V. De, and S. Borkar, "Scaling trends of cosmic ray induced soft errors in static latches beyond 0.18u," *2001 Symposium on VLSI Circuits. Digest of Technical Papers*, 14-16 June 2001, Kyoto, Japan, 2001. pp. 61-2.

[12] P. Shivakumar, M. Kistler, S. W. Keckler, D. Burger, and L. Alvisi, "Modeling the effect of technology trends on the soft error rate of combinational logic," *Proceedings International Conference on Dependable Systems and Networks*, 23-26 June 2002, Washington, DC, USA, 2002. pp. 389-98.

[13] Ritesh Mastipuram and Edwin C Wee, "Soft errors' impact on system reliability," http://www.edn.com.

[14] G. E. Moore, "Cramming more components onto integrated circuits," *Electronics*, 1965.

[15] S. Borkar, "Design challenges of technology scaling," *IEEE Micro*, vol. 19, pp. 23-29, 1999.

[16] M. S. Hrishikesh, N. P. Jouppi, K. I. Farkas, D. Burger, S. W. Keckler, and P. Shivakumar, "The optimal logic depth per pipeline stage is 6 to 8 FO4 inverter delays," *29th Annual International Symposium on Computer Architecture*, May 25-29 2002, Anchorage, AK, 2002. pp. 14-24.

[17] H. Yu and J.-D. Cho, "Low-power design and architecture," *Potentials*, IEEE, vol. 20, pp. 18-22, 2001.

[18] C. Hu, "Device and Technology Impact on Low Power Electronics," *in Low Power Design Methodologies*, J. M. Rabaey and M. Pedram, Eds.: Kluwer Academic Publishers, 1996, pp. 21-35.

[19] N. Seifert, X. Zhu, and L. W. Massengill, "Impact of scaling on soft-error rates in commercial microprocessors," *Nuclear Science, IEEE Transactions on*, vol. 49, pp. 3100-3106, 2002.

[20] S. Buchner, M. Baze, D. Brown, D. McMorrow, and J. Melinger, "Comparison of error rates in combinational and sequential logic," *IEEE Transactions on Nuclear Science Proceedings of the 1997 IEEE Nuclear and Space Radiation Effects Conference*, NSREC. Part 1 (of 3), Jul 21-25 1997, vol. 44, pp. 2209-2216, 1997.

[21] T. Sakurai and A. R. Newton, "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas," *Solid-State Circuits*, IEEE Journal of, vol. 25, pp. 584-594, 1990.

[22] B. L. Austin, K. A. Bowman, X. Tang, and J. D. Meindl, "A low power transregional MOSFET model for complete power-delay analysis of CMOS gigascale integration (GSI)," *ASIC Conference 1998 Proceedings. Eleventh Annual IEEE International*, 1998. pp. 125-129.

[23] R. K. Krishnarnurthy, A. Alvandpour, V. De, and S. Borkar, "High-performance and low-power challenges for sub-70 nm microprocessor circuits," *Custom Integrated Circuits Conference*, 2002. Proceedings of the IEEE 2002, 2002. pp. 125-128.

[24] G. Sery, S. Borkar, and V. De, "Life is CMOS: why chase the life after?," *Design Automation Conference*, 2002. Proceedings. 39th, 2002. pp. 78-83.

[25] A. U. Diril, Y. S. Dhillon, K. Choi, and A. Chatterjee, "An O(N) supply voltage assignment algorithm for low-energy serially connected CMOS modules and a heuristic extension to acyclic data flow graphs," *Proceedings IEEE Computer Society Annual Symposium on VLSI. New Trends and Technologies for VLSI Systems Design. ISVLSI 2003*, 20-21 Feb. 2003, Tampa, FL, USA, 2003. pp. 173-9.

[26] P. Pant, V. K. De, and A. Chatterjee, "Simultaneous power supply, threshold voltage, and transistor size optimization for low-power operation of CMOS circuits," *Very Large Scale Integration (VLSI) Systems*, IEEE Transactions on, vol. 6, pp. 538-545, 1998.

[27] P. Pant, R. K. Roy, and A. Chattejee, "Dual-threshold voltage assignment with transistor sizing for low power CMOS circuits," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 9, pp. 390-394, 2001.

[28] T.-Y. Choi, W.-I. Cho, and D.-W. Kim, "A simple CMOS delay model for wide applications," *IEEE Asia Pacific Conference on Circuits and Systems*, 1996. pp. 77-80.

[29] R. B. Hitchcock, “Timing verification and the timing analysis program,” in *Proc. IEEE/ACM Design Automation Conf.*, 1982, pp. 594–604.

[30] N. P. Jouppi, “Timing analysis for nMOS VLSI,” in *Proc. IEEE/ACM Design Automation Conf.*, 1983, pp. 411–418.

[31] J. D. Wiest, F. K. Levy, “A Management Guide to PERT/CPM,” Prentice-Hall, 1977.

[32] P. Jensen, “Operations Management / Industrial Engineering,” http://www.me.utexas.edu/~jensen/ORMM/omie/design/unit/project/crit_path.html, University of Texas at Austin (as of 02/19/2005).

[33] Synopsys, "Power Compiler," http://www.synopsys.com.

[34] Sequence, "Power Theater," http://www.sequencedesign.com.

[35] “Introducing Intel XScale Microarchitecture,” http://www.intel.com/update/departments/applied/ac09003.pdf.

[36] “AMD Power Now Technology,” http://www.amd.com/epd/processors/6.32bitproc/8.amdk6fami/x24404/24404a.pdf.

[37] “Crusoe: Features and Benefits,” http://www.transmeta.com/crusoe/features.html.

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

[39] K. Usami and M. Horowitz, "Clustered voltage scaling technique for low-power design," *Low Power Design Symposium*, 23-26 April 1995, Dana Point, CA, USA, 1995. pp. 3-8.

[40] M. Donno, L. Macchiarulo, A. Macii, E. Macii, and M. Poncino, "Enhanced clustered voltage scaling for low power," *GLSVLSI '02. Proceedings of the 12th ACM Great Lakes Symposium on VLSI*, 18-20 April 2002, New York City, NY, USA, 2002. pp. 18-23.

[41] J.-M. Chang and M. Pedram, "Energy minimization using multiple supply voltages," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 5, pp. 436-443, 1997.

[42] 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, pp. 616-29, 2001.

[43] V. Sundararajan and K. K. Parhi, "Synthesis of Low-Power CMOS VLSI Circuits using Dual Supply Voltages," *ACM Design Automation Conference*, New Orleans, 1999. pp. 72-75.

[44] K. Usami, K. Nogami, M. Igarashi, F. Minami, Y. Kawasaki, T. Ishikawa, M. Kanazawa, T. Aoki, M. Takano, C. Mizuno, M. Ichida, S. Sonoda, M. Takahashi, and N. Hatanaka, "Automated low-power technique exploiting multiple supply voltages applied to a media processor," *Proceedings of the 1997 IEEE Custom Integrated Circuits Conference*, May 5-8 1997, Santa Clara, CA, USA, 1997. pp. 131-134.

[45] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, "Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits," *Proceedings of the IEEE*, vol. 91, pp. 305-327, 2003.

[46] C. H. Kim and K. Roy, "Dynamic V_t SRAM: a leakage tolerant cache memory for low voltage microprocessors," *Low Power Electronics and Design, 2002. ISLPED '02. Proceedings of the 2002 International Symposium on*, 2002. pp. 251-254.

[47] L. Wei, Z. Chen, K. Roy, M. C. Johnson, Y. Ye, and V. K. De, "Design and optimization of dual-threshold circuits for low-voltage low-power applications," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 7, pp. 16-24, 1999.

[48] Y. Ye, S. Borkar, and V. De, "A new technique for standby leakage reduction in high-performance circuits," *VLSI Circuits*, 1998. Digest of Technical Papers. 1998 Symposium on, 1998. pp. 40-41.

[49] M. C. Johnson, D. Somasekhar, L.-Y. Chiou, and K. Roy, "Leakage control with efficient use of transistor stacks in single threshold CMOS," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 10, pp. 1-5, 2002.

[50] S. J. Shieh, J. S. Wang, "Design of low-power domino circuits using multiple supply voltages," *IEEE International Conference on Electronics, Circuits and Systems*, Sept. 2001, pp. 711 - 714.

[51] S. O. Jung, K. W. Kim, S. M. Kang, "Low-swing clock domino logic incorporating dual supply and dual threshold voltages," *Design Automation Conference*, June 2002, pp. 467 - 472.

[52] A. U. Diril, Y. S. Dhillon, A. Chatterjee, A. D. Singh, "Low-Power Domino Circuits using NMOS Pull-up on Off critical Paths," *Proceedings of Asia and South Pacific Design Automation Conference (ASP-DAC'2005), January 21-23, 2005. Shanghai, China*

[53] Y. S. Dhillon, A. U. Diril, A. Chatterjee, and A. D. Singh, "Low-power dual Vth pseudo dual Vdd domino circuits," *Integrated Circuits and Systems Design, 2004. SBCCI 2004. 17th Symposium on*, 2004. pp. 273-277.

[54] http://www.tsmc.com, Level 49 Spice parameters for 0.18µ TSMC process.

[55] M. R. Prasad, D. Kirkpatrick, R. K. Brayton, "Domino logic synthesis and technology mapping," *Int. Workshop on Logic Synthesis*, 1997.

[56] A. Srivastava and D. Sylvester, "Minimizing total power by simultaneous Vdd Vth assignment," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, pp. 665-77, 2004.

[57] A. Srivastava, D. Sylvester, and D. Blaauw, "Power minimization using simultaneous gate sizing, dual-Vdd and dual-Vth assignment," *Design Automation Conference, 2004. Proceedings. 41st*, 2004. pp. 783-787.

[58] M. Hamada, Y. Ootaguro, and T. Kuroda, "Utilizing surplus timing for power reduction," *IEEE 2001 Custom Integrated Circuits Conference, May 6-9 2001*, San Diego, CA, 2001. pp. 89-92.

[59] Y. Cao, T. Sato, M. Orshansky, D. Sylvester, and C. Hu, "New paradigm of predictive MOSFET and interconnect modeling for early circuit simulation," *IEEE Custom Integrated Circuits Conference Proceedings (CICC' 2000)*, pp. 201-204.

[60] S. H. Kulkarni and D. Sylvester, "Fast and Energy-Efficient Asynchronous Level Converters for Multi-VDD Design," *IEEE International SOC Conference*, 2003. pp. 169-172.

[61] V. Kursun, R. M. Secareanu, and E. G. Friedman, "CMOS voltage interface circuit for low power systems," *2002 IEEE International Symposium on Circuits and Systems, May 26-29 2002*, Phoenix, AZ, 2002. pp. 667-670.

[62] C.-C. Yu, W.-P. Wang, and B.-D. Liu, "A new level converter for low-power applications," *IEEE International Symposium on Circuits and Systems (ISCAS 2001), May 6-9 2001*, Sydney, NSW, 2001. pp. 113-116.

[63] F. Ishihara, F. Sheikh, and B. Nikolic, "Level Conversion for Dual-Supply Systems," *Proceedings of the 2003 International Symposium on Low Power Electronics and Design, (ISLPED'03), Aug 25-27 2003*, Seoul, South Korea, 2003. pp. 164-167.

[64] A. U. Diril, Y. S. Dhillon, A. Chatterjee, and A. D. Singh, "Level-Shifter Free Design of Low Power Dual Supply Voltage CMOS Circuits Using Dual Threshold Voltages," *VLSI Design, 2005. 18th International Conference on*, 2005. pp. 159-164.

[65] P. E. Dodd and L. W. Massengill, "Basic mechanisms and modeling of single-event upset in digital microelectronics," *IEEE Transactions on Nuclear Science*, vol. 50, pp. 583-602, 2003.

[66] M. Nicolaidis and Y. Zorian, "On-line testing for VLSI - a compendium of approaches," *Journal of Electronic Testing: Theory and Applications (JETTA)*, vol. 12, pp. 7-20, 1998.

[67] M. Nicolaidis, "Time redundancy based soft-error tolerance to rescue nanometer technologies," *Proceedings of the IEEE VLSI Test Symposium (VTS'99), Apr 1999*, pp. 86-94, 1999.

[68] K. Mohanram and N. A. Touba, "Cost-effective approach for reducing soft error failure rate in logic circuits," *Proceedings of the International Test Conference (ITC)*, 2003. pp. 893-901.

[69] M. Oman, G. Papasso, D. Rossi, and C. Metra, "A model for transient fault propagation in combinatorial logic," *9th International IEEE On-Line Testing Symposium, 7-9 July 2003*, Kos Island, Greece, 2003. pp. 111-15.

[70] C. Zhao, X. Bai, and S. Dey, "A scalable soft spot analysis methodology for compound noise effects in nano-meter circuits," *Proceedings of 41st Design Automation Conference,* pp. 894-899, 2004.

[71] L. Anghel and M. Nicolaidis, "Cost reduction and evaluation of a temporary faults detecting technique," *Proceedings of Design, Automation and Test in Europe Conference and Exhibition*, pp. 591-598, 2000.

[72] Y. S. Dhillon, A. U. Diril, A. Chatterjee, "Soft-Error Tolerance Analysis and Optimization of Nanometer Circuits," *Design, Automation and Test in Europe Conference, DATE 2005,* pp. 288-293, 2005.

[73] F. N. Najm and I. N. Hajj, "The complexity of fault detection in MOS VLSI circuits," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 9, pp. 995-1001, 1990.

[74] C. S. Chang, "Applications of Metal-Insulator-Metal (MIM) Capacitors," www.sematech.org: International SEMATECH Technology Transfer # 00083985A-ENG, 2000.

[75] J. N. Burghartz, M. Soyuer, K. A. Jenkins, M. Kies, M. Dolan, K. J. Stein, J. Malinowski, and D. L. Harame, "Integrated RF components in a SiGe bipolar technology," *IEEE Journal of Solid-State Circuits,* vol. 32, pp. 1440-1445, 1997.

[76] L. B. Freeman, "Critical charge calculations for a bipolar SRAM array," *IBM J. Res. Develop.*, vol. 40, pp. 119-129, 1996.

[77] S.C. Wong, G.Y. Lee, D.J. Ma, “Modeling of interconnect capacitance, delay, and crosstalk in VLSI,”, *IEEE Trans. on Semiconductor Manufacturing*, vol. 13, no. 1, pp. 108 – 111, 2000.

[78] J. F. Ziegler, "Terrestrial cosmic rays," *IBM J. Res. Develop.*, vol. 40, pp. 19-39, 1996.

[79] Anon. “Topological Sorting,” http://en.wikipedia.org/wiki/Topological_sorting (as of 02/23/2005).

[80] R. Puri, A. Bjorksten, and T. E. Rosser, "Logic optimization by output phase assignment in dynamic logic synthesis," *Proceedings of the 1996 IEEE/ACM International Conference on Computer-Aided Design, Nov 10-14 1996*, San Jose, CA, USA, 1996. pp. 2-8.

[81] S. Kundu, "Efficient technique for obtaining unate implementation of functions through input encoding," *Integration, the VLSI Journal*, vol. 17, pp. 265-270, 1994.