Characterizing Processors for Time and Energy Optimization

by

Harshit Goyal

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

Auburn, Alabama
August 06, 2016

Keywords: Low Power Design, Energy Per Cycle, Cycle Efficency, Peak Power, Thermal Design Power

Approved by

Vishwani D. Agrawal, Chair, James J. Danaher Professor of Electrical and Computer Engineering
Prathima Agrawal, Chair, Emeritus Professor, Formerly Smauel Ginn Distinguished Professor of Electrical and Computer Engineering
Victor P. Nelson, Professor of Electrical and Computer Engineering

Abstract

Moore's law [40] states that the number of transistors that can be most economically placed on an integrated circuit will double approximately every two years. The law has often been subjected to the following criticism: while it boldly states the blessing of technology scaling, it fails to expose its bane. A direct consequence of Moore's law is that "the power density of the integrated circuit increases exponentially with every technology generation" [45]. This implicit trend has arguably brought about some of the most important changes in electronic and computer designs. In the next two decades, diminishing transistor size, speed scaling and practical energy limit will create new challenges for continued performance scaling. As a result, the frequency of operations will increase slowly, with energy being the key limiter of performance, forcing designs to use large-scale parallelism, heterogeneous cores, and accelerators to achieve performance and energy efficiency.

Energy and performance are important aspects of microprocessors and their verification and management require, measurement, estimation and analysis, and these aspects are discussed through this research. A processor executes a computing job in a certain number of clock cycles. The clock frequency determines the time that the job will take. Another parameter, cycle efficiency or cycles per joule, determines how much energy the job will consume. The execution time measures performance and, in combination with energy dissipation, influences power, thermal behavior, power supply noise and battery life. We describe a method for power management of a processor. To show management of performance and energy, we study several Intel processors from 45 nm, 32 nm and 22 nm technology nodes for both *thermal design power* (TDP) and peak power. They are characterized for two different predictive technology models: Bulk CMOS and High-K metal Gate, which are available for

analysis in H-spice [4] simulation. Our analysis establishes correlation between the simulation data for an adder circuit and the processor data sheet, and then estimates operating frequency and cycle efficiency as functions of the supply voltage. This data is useful in managing the operational characteristics of processors, especially those used in mobile or remote systems where both execution time and energy are important. We illustrate how this information is utilized in managing the highest performance including turbo (over-clocking), lowest energy, and all in-between operating modes.

An Intel processor in 32 nm bulk CMOS technology is used as an illustrative example. First, we characterize the technology by H-spice [4] simulation of a ripple carry adder for critical path delay, dynamic energy and static power at a wide range of supply voltages. The adder data is then scaled based on the clock frequency, supply voltage, thermal design power (TDP) and other specifications of the processor. To optimize the time and energy performances, voltage and clock frequency are determined, showing 28% reduction in both execution time and energy dissipation.

Acknowledgments

There are many people to whom I would like to express my gratitude for their help during the pursuit of my master's degree. Foremost among them are Professors Prathima Agrawal and Vishwani D. Agrawal, without whose constant support and guidance this dissertation would not have been possible. I am deeply thankful to them as a very generous mentors throughout my studies. The work has been delightful and successful under their valuable advice. I would like to thank Professor Victor P. Nelson for great suggestions as my advisory committee member and through his distinguished lectures. I would also like to acknowledge Professor Narendra K. Govil and Dr. Ashutosh Mishra for their loving and caring support throughout my studies.

Every result described in this thesis was accomplished with the help and support of fellow lab-mates and collaborators. My heartfelt thanks go out to Karthik Jayaraman, Aditi, Sindhu and Muralidharan for their immense patience and guidance. Thanks to all my dearest friends: Mohit, Sarthak, Rahul, Vaibhav Agg., Amrit, Rajat, Tanuj, Gautam and Vaibhav Gupta. Without you guys it won't have been possible for me to complete this wonderful engineering journey at auburn with lots of joyful memories to cherish through my whole life.

Above all, I would like to express my deepest gratitude to my loving grandfather, parents and sister for their endless love and support during my whole life and without whom I would never be here.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Power consumption of a digital CMOS circuit is proportional to the frequency of execution and the square of the operating voltage, while energy consumption also depends on the total execution time [64]. The energy consumption has become one of the primary concerns in processor design due to the recent popularity of portable devices and cost concerns related to desktops and servers. The battery capacity has improved very slowly (a factor of 2 to 4 over the last 30 years), while the computational demands have drastically increased over the same time frame. With a number of performance oriented devices emerging with a huge demand on power from a fixed capacity battery, using the battery wisely becomes important. In energy-constrained systems, low power design is essential for extending battery and system lifetime.

Lowering voltage supply (V_{dd}) decreases dissipated energy quadratically, but also causes an increase in delay [17]. In order to satisfy the aggressive performance requirements demanded by applications, the threshold voltage (V_{th}) should also be lowered, to have both low power operation and high performance. However, there is a cost of higher static power dissipation due to large leakage currents. As the semiconductor industry has scaled into very small feature sizes, however, the need to control leakage current has prevented further threshold and supply voltage scaling. This break from Dennard scaling [22] has led to rapid increases in power density, and power consumption is now a primary constraint in all microprocessor design. Not only does power dissipation impact battery-life in embedded devices, it also constrains achievable performance in server architectures.

To optimize a microprocessor for energy efficiency, a designer must consider the energy-efficiency benefit trade-offs of all design options, choosing those features and parameter values

that offer the best return in terms of performance per unit energy. While this strategy is straightforward in theory, in practice, it has been difficult to automate since the space of processor design options is often extremely large, and one needs to consider trade-offs in the architecture, the circuit design and potentially the technology. Thus, regardless of whether one is designing low-power embedded mobile devices or high-performance servers, power consumption is now a critical factor in determining the system's overall performance.

In this new power-constrained era, the principal design objective is to achieve energy efficiency. Designers need to find ways to make the most of their power budgets, andin addition to finding new, more energy-efficient design techniquesthis requires ways of exploring existing design spaces to enable designers to tune their systems for efficient operation. This work looks into one of the most important areas of contemporary research in electrical and computer engineering: Energy Efficiency.

Power and performance are two conflicting goals a designer has to achieve [39]. In this work, we have used a recently defined parameter, cycle efficiency of processor (η) to investigate the cycles that could be run using a given amount of energy [55] [56]. Performance of a processor means how fast it can execute a task and refers to its performance in time. For given architecture, hardware and software, clock frequency (f), i.e., cycles per second (Hz), or cycle rate is the rate of computational work measured in clock cycles done per unit time. In a similar way, a recently defined new measure, cycle efficiency η as cycles per joule, or the rate of computational work per unit energy is used, that can be considered while deciding upon the working conditions of the processor for optimal energy efficiency.

In battery powered systems, the energy consumed to complete a task is often a more relevant metric than power. Circuit designers often have the ability to trade off power for performance. Thus it is possible for a high power system, which rapidly completes a task, to consume less energy than a low power system that steadily draws power for an extended period to complete the same task. Battery life for mobile systems can be extended by being energy efficient, not necessarily by being low power.

1.1 Motivation

Energy usage is increasingly a key constraint for microprocessor designs. Although once only a concern for the fastest supercomputers, energy is now important across a broad range of computer designs. For mobile systems, processor energy consumption is a limiting design factor in terms of battery weight and lifetime. High-performance microprocessors are constrained by peak power usage and the ability to supply current and dissipate the generated heat; these problems directly affect the maximum processor speed. Additionally, energy consumption is crucial in large server farms that are limited by the maximum capabilities of the power infrastructure as well as the cost of the energy and ability to keep system cool.

Traditionally, microprocessor designs have focused almost exclusively on performance, but these shifting constraints are requiring architects to consider energy, in addition to performance, when evaluating design decisions [41]. In addition, energy efficiency of the micro architecture of general-purpose microprocessors is starting to play a critical role in the performance versus power trade-offs. As processors consume the dominant amount of power in computer systems, power management of multi-core processors is extremely significant. The new goal is to develop energy-efficient designs which simultaneously have high-performance and low energy consumption [20] [24].

Electronic systems are collections of components which may be heterogeneous in nature. For example, a laptop has digital VLSI components, analog components (e.g., wireless card), mechanical parts (e.g., hard disk drive), and optical components (e.g., display). In general, peak performance is required only during some selected time intervals. As a result, the system components do not always need to be delivering peak performance. The ability to tune their performance to the workload (e.g., user's requests), is important in achieving energy efficient utilization. The aim of this research is to present new approaches for lowering energy consumption and, to serve this purpose, a method is described to optimize the performance (execution time and energy consumption) of a processor.

1.2 Problem Statement

The aims of this research are:

1. To study and obtain data on voltage, frequency and cycle efficiency of a processor to enable a methodology for power and performance management.

2. To determine the operating conditions (voltage and frequency) for optimal time and energy operation.

1.3 Organization of the Thesis

Chapter 2 presents the background material on power and energy of a microprocessor, prevailing methods of economizing those, and some metrics used to quantify the consumption. The material beyond this is original. Chapter 3 presents procedures and tools used to assess the technology of a processor using a simpler circuit. The circuit used for this purpose in this research is a ripple-carry adder (RCA). A method for finding the simulation vectors that mimic the processor in signal activity is given and sample results for 32 nanometer bulk CMOS technology are presented. Chapter 4 describes procedures for scaling the RCA results of Chapter 3 to an Intel processor. Scenarios for energy minimization under various performance requirements are presented. Chapter 5 applies the techniques of Chapters 3 and 4 to several other technologies. Chapter 6 concludes the thesis, summarizing the main findings and outlining proposals for future research.

Early results of this research were presented at the *16th International IEEE Workshop on Microprocessor/SoC Test and Verification* (MTV), Austin, Texas, December 2015 and *34th IEEE VLSI Test Symposium* (VTS), Las Vegas, Nevada, April 2016, and will soon be available in the MTV workshop proceedings [27, 28].

Chapter 2

Theory and Background

2.1 Fundamentals of Low Power Design

Low-power circuit operation is becoming an increasingly important metric for future integrated circuits. As technology continues to scale down into the sub-micron range, massively parallel architectures are becoming increasingly popular. These present serious power considerations. Low power and low energy have captivated circuit designers for the past few years in the quest for enhancing performance and extending battery lifetime. The demand for integrating more functions with faster speeds is met by a slow increase in the capacity of batteries. The increasing power dissipation for fixed-supply devices is almost as equally challenging for portable devices. As technology feature size is reduced, the number of transistors on the chip is increased and more power is dissipated.

According to Moore's law [40], the number of transistors quadruples every two to three years. Expensive packing techniques are essential for dissipating such extensive power consumption from that large number of transistors. Also, increased power dissipation has an impact on device reliability. The terms "low power" and "low energy", although, defined differently, both serve to achieve the same objective. Power is defined as the average product of the supplied voltage to a chip from the power supply and the consumed current, measured in watts. Meanwhile, the term energy refers to the energy dissipated per operation and is measured in joules. In fact, energy can be expressed in terms of *power-delay product* (PDP), which is the product of power and the duration of consumption. In general, reducing power will increase delay time and, thus, performance is affected by these two parameters. There are several techniques for power and energy reduction. Most of the techniques in low power design are not really new ideas or concepts but mainly they are revisited due to transistors

scaling, which is a source of leakage currents. In this chapter, the most significant power dissipation sources in CMOS circuits are identified and some previous and recent performance and energy metrics are discussed.

2.2 Power Consumption in CMOS Circuits

Power dissipation in CMOS digital circuits is categorized into two types: peak power and time-averaged power. Peak power is a reliability issue that can cause both hardware failure (chip lifetime) and temporary functional failure. Effects like *ground bounce* and *power droop* [57], caused by the excessive instantaneous current through the resistive and inductive power network, influence the performance of a design due to the increased gate and interconnect delays. Also noise margins are reduced, increasing the chance of chip failure due to *crosstalk* [57]. Sustained large power consumption, on the other hand, causes the device to overheat reducing the reliability and lifetime of the circuit. The time-averaged power consumption in conventional CMOS digital circuits occurs in two forms: dynamic and static. Dynamic power dissipation occurs in the logic gates that are in the process of switching from one state to another. During this process, any internal and external capacitances associated with the gate's transistors are charged or discharged, thereby consuming power. Static power dissipation is associated with inactive logic gates (i.e., not currently switching from one state to another). Dynamic power is important during normal operation, especially at high operating frequencies, whereas static power is more important during standby, especially for battery-powered devices. An overview of different power dissipation types is given in Figure 2.1

2.2.1 Dynamic Power Dissipation

Dynamic power dissipation, primarily caused by the current from the charging or discharging of parasitic capacitances, consists of three components: switching power, short-circuit power, and glitch power.

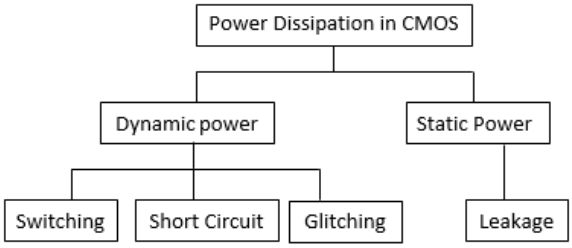


Figure 2.1: Different power dissipation types in CMOS circuits.

2.2.1.1 Switching Power Dissipation

In digital CMOS circuits, the switching power is dissipated when current is drawn from the power supply to charge up the output node capacitance. During this switching event, the output node voltage typically makes a full transition from 0 to V_{dd}, and one-half of the energy drawn from the power supply is dissipated as heat in the conducting pMOS transistors. The energy stored in the output capacitance during charge-up is dissipated as heat in the conducting nMOS transistors, when the output voltage switches from V_{dd} to 0. A CMOS inverter circuit, depicted in Figure 2.2, is presented to illustrate this dynamic power dissipation during switching. The total capacitive load C_{load} at the output of the inverter consists of the diffusion capacitance of the drains of the inverter transistors, the total interconnect capacitance, and the input gate oxide capacitance of the driven gates that are connected to the inverter's output. In most CMOS digital circuits, the switching power is the dominant component in power dissipation. Figure 2.3 exhibits the supply current waveform of the inverter circuit. The average switching power dissipation of the inverter can be calculated from the energy, required to charge up the output node to V_{dd} and discharge the total output load capacitance to ground (GND). The generalized expression for the switching power dissipation of a CMOS logic gate can be written as [21, 34],

$$P_{dyn} = \alpha \cdot C_{load} \cdot V_{dd}^2 \cdot f_{clk} \tag{2.1}$$

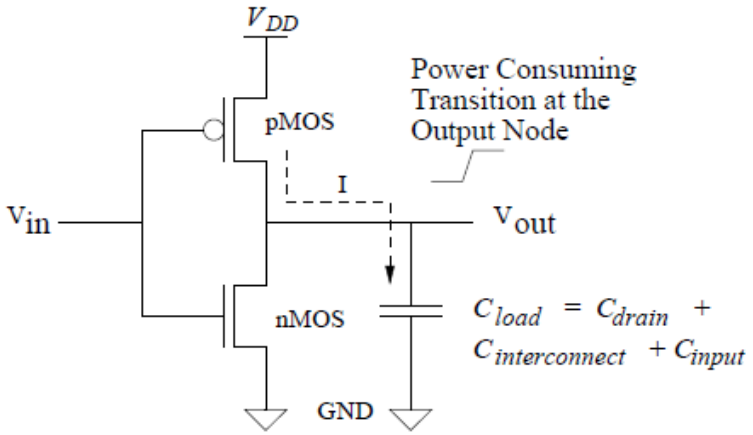


Figure 2.2: CMOS inverter for switching power calculation.

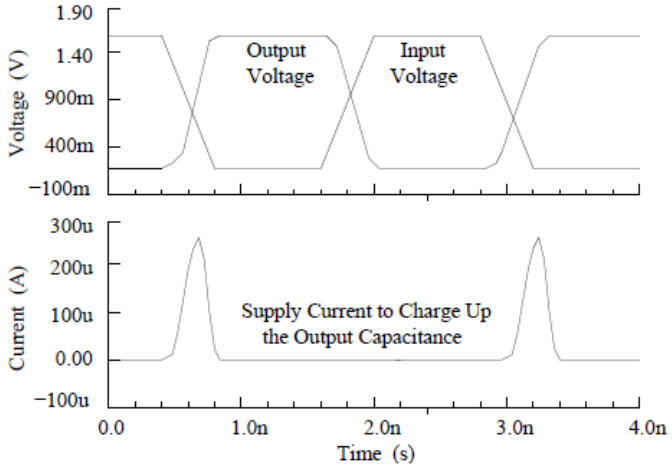


Figure 2.3: Supply current used to charge up the load capacitance.

where α is the switching activity factor of the gate (i.e., average number of $0 \rightarrow 1$ transitions in a clock period at the gate output), C_{load} represents the total load capacitance, V_{dd} is the supply voltage, and f_{clk} represents the clock frequency. The switching activity α is often computed by multiplying the probability that the output of a gate was at logic 0 by the probability that the output will rise to logic 1 [34]. This assumes that transitions are *clean*, i.e., there are no glitch or hazard pulses. The parameter α is a function of several factors, including the Boolean function performed by the gate, the logic style, and the input signal statistics. Equation 2.1 indicates that the supply voltage is the dominant factor in the switching power dissipation. Thus, reducing the supply voltage is the most effective technique to reduce the power dissipation. Other methods, such as reducing the switching

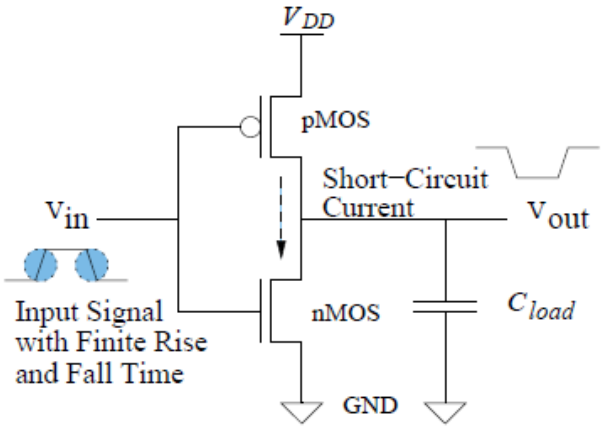


Figure 2.4: CMOS inverter for short-circuit power calculation.

activity and the load capacitance [34], for reducing the power consumption are also suggested by Equation 2.1.

2.2.1.2 Short-Circuit Power Dissipation

In static CMOS circuits, short circuit power dissipation is generated by the short circuit current flowing through both the nMOS and the pMOS transistors during switching. The short circuit current occurs if a logic gate is driven by the input voltage waveforms with the finite rise and fall times, as shown in Figure 2.4. Thus, both the nMOS and the pMOS transistors in the circuit conduct simultaneously for a short period of time during the transitions, forming a direct current path between the power supply and GND. This short circuit current does not contribute to the charging of the capacitance in the circuit. Figure 2.5 illustrates the input-output waveforms and the short circuit current of the inverter circuit with zero load capacitance in Figure 2.4. If a symmetric CMOS inverter has the same transconductance (i.e., $k_n = k_p = k$) and threshold voltage parameters (i.e., $V_{Tn} = V_{Tp} = V_t$), and if the input voltage waveform has equal rise and fall times ($\tau_{rise} = \tau_{fall} = \tau$), the average short-circuit power dissipation with a very small capacitive load is calculated as follows [34]:

$$P_{dyn} = \frac{1}{12} \cdot k \cdot \tau \cdot f_{clk} \cdot C_{load} \cdot (V_{dd} - 2V_t)^3 \tag{2.2}$$

where k is transconductance of the transistors, V_t is threshold voltage, and τ represents the

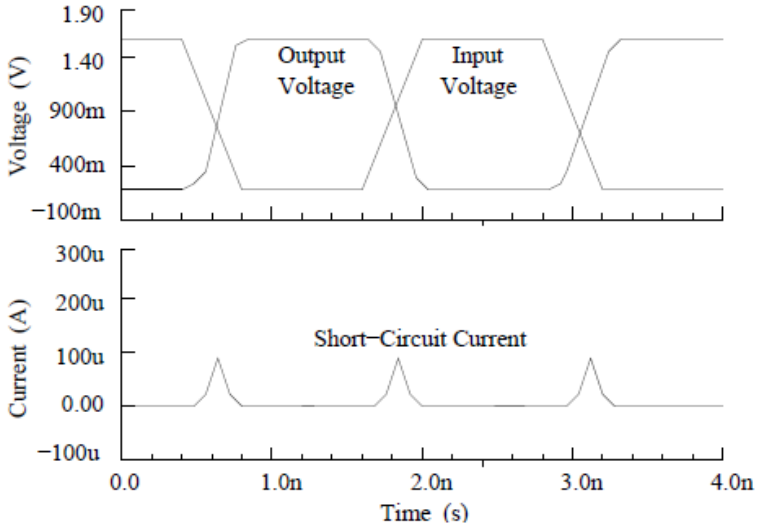


Figure 2.5: Short-circuit current during switching.

equal rise and fall times. Note that the short-circuit power dissipation is linearly proportional to the input signal's rise and fall times. Therefore, reducing the input transition times will decrease the short-circuit current component. However, the increased load capacitance (i.e., the output rise/fall time is larger than the input rise/fall time) can also lead to less short-circuit power dissipation [60]. Yet, this goal should be balanced carefully against other performance goals such as propagation delay.

2.2.1.3 Glitch Power Dissipation

Glitch power is the power dissipated in the intermediate transitions during the evaluation of the logic function of the circuit [11, 12, 49]. In multi-level logic circuits, the propagation delay from one logic block to the next can cause the input signals to the block to change at different times. Thus, a node can exhibit multiple transitions in a single clock cycle before settling to the correct logic level. These intermediate erroneous outputs lead to a power loss in charging and discharging the output load capacitance. Primarily, glitches occur due to a mismatch or imbalance in the path lengths in the logic network [11, 12, 49]. Such a mismatch in the path lengths results in a mismatch in the signal timing with respect to the primary

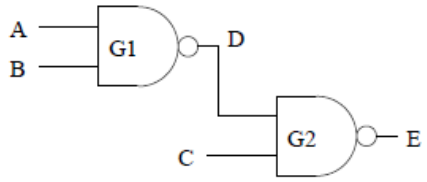


Figure 2.6: Multi-level static CMOS circuit.

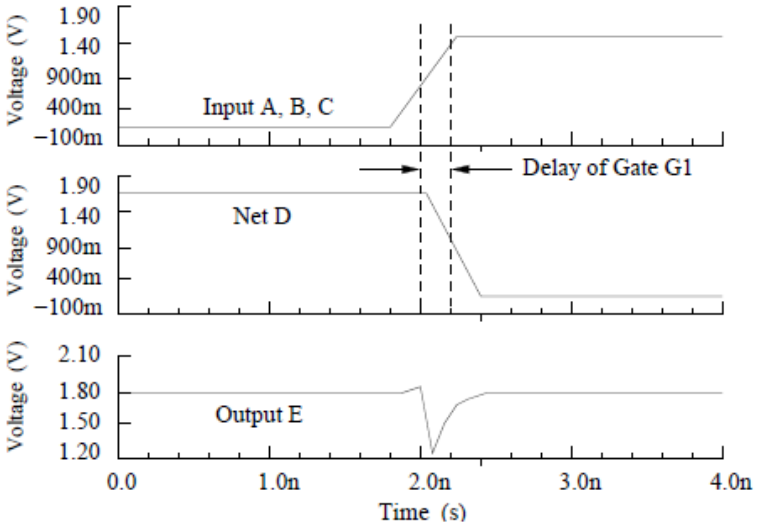


Figure 2.7: Signal glitches in multi-level CMOS circuit.

inputs. Figure 2.6 shows a simple multi-level network. If both NAND gates have the same delay and three input signals arrive at the same time, the network will suffer from glitches, as seen in Figure 2.7. To avoid such power loss, designers can use synchronous circuits in which all the outputs are either latched or gated to synchronize the inputs to the next stage. Also, dynamic circuits avoid the problem of glitch power by synchronizing the output with the clock signal. Finally, a careful layout [59] and gate delay [51] manipulation can reduce the skew among the input signals to each logic gate, leading to lower glitch activity.

2.2.2 Static Power Dissipation

Leakage power forms a significant potion of the total power dissipation in DSM technologies. The different leakage current components are shown in Figure 2.8 [35]. I_1 is the reverse-bias p-n junction leakage caused by barrier emission and minority carrier diffusion

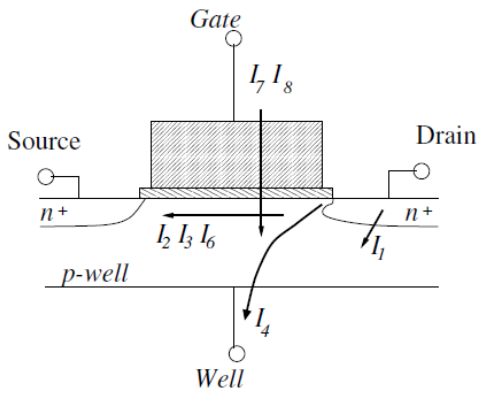


Figure 2.8: Leakage current components.

and band-to-band tunneling. I_2 is sub-threshold conduction current. I_3 results from the drain-induced barrier lowering (DIBL) effect. I_4 is gate-induced drain leakage (GIDL). I_5 is channel punch-through. I_6 is hot carrier injection current. I_7 is oxide leakage. I_8 is gate current due to hot carrier injection. I_1 through I_6 are OFF currents while I_7 and I_8 are ON and switching currents. Here, the main concern is the OFF leakage current and therefore, the focus is on the current components I_1 through I_6, which are explained below [20].

- Junction Reverse Bias Current (I_1): I_1 has two components: One is minority carrier diffusion/drift near the edge of the depletion region, and the other is due to electron hole pair generation in the depletion region of the reverse biased junction. Heavily doped junctions are also prone to Zener and band-to-band tunneling. The p-n reverse bias leakage is a function of junction area and doping concentration. I_1 is normally a minimal contributor to total OFF current.

- Sub-threshold Conduction Current (I_2): Sub-threshold conduction or weak inversion current between source and drain when supply voltage is below V_{th}. The sub-threshold current occurs due to carrier diffusion when the gate-source voltage, V_{gs}, has exceeded the weak inversion point, but still below the threshold voltage, where carrier drift is dominant. Sub-threshold conduction typically dominates modern device off-state leakage due to the low threshold devices.

- Drain-Induced Barrier Lowering, DIBL (I_3): DIBL is the effect of lowering the source potential barrier near the channel surface as a result of the applied drain voltage. Ideally, DIBL does not change the sub-threshold slope but does lower V_{th}. Higher surface and channel doping, and shallow source/drain junction depths work to reduce the DIBL mechanism.

- Gate-Induced Drain Leakage, GIDL (I_4): GIDL current arises in the high electric field under the gate/drain overlap region, causing a thinner depletion region of drain to well junction. GIDL results in an increase in leakage current when applying a negative voltage to the gate (NMOS case). GIDL is small for normal supply voltage but its effect rises at higher supply voltages (near burn-in).

- Punch-through (I_5): Punch-through occurs when source and drain depletion regions approach each other and the gate voltage loses control over the channel current in the sub-gate region. Punch through current varies quadratically with drain voltage. Punch-through is often regarded as a subsurface version of DIBL.

- Narrow width effect (I_6): Threshold voltage tends to decrease in trench-isolated small effective channel width devices. The narrow width effect causes the threshold voltage to decrease in trench isolated technologies for channel widths on the order of W $\leq 0.5\mu$m. It can be ignored for device sizes $>> 0.5\mu$m.

Subthreshold leakage current is the largest leakage current component. It increases exponentially as a result of threshold voltage reduction. In a simple form, subthreshold leakage current, I_{sub}, is given by [35] as follow:

$$I_{sub} = I_0 \cdot e^{\frac{(V_{gs}-V_t)}{(\alpha V_{th})}} \tag{2.3}$$

Where,

V_t is the device threshold voltage,

V_{th} is thermal voltage and it is 25.9 mV at room temperature (300K),

I_0 is the current when $V_{gs} = V_t$, and

α ranges from 1.0 to 2.5 and is dependent on the device fabrication process.

Sub-threshold current is becoming a limiting factor in low voltage and low power chip design. When operating voltage is reduced the device threshold voltage V_t has to be reduced accordingly to compensate for loss in switching speed.

2.2.3 The Conflict Between Dynamic Power and Static Power

Dynamic power can be reduced by reducing the supply voltage. Supply voltage reduction has been a constant phenomenon with the technology scaling [38]. Voltages for semiconductor devices have been reduced from 5 volts to 0.8 volts in the most recent technologies. But when the voltage is lowered, the transistor ON current I_{ds} reduces which makes devices switch slower. The approximate equation for I_{ds} is given by

$$I_{ds} = \mu \cdot C_{ox} \frac{W}{L} \cdot \frac{(V_{gs} - V_t)^2}{2} \tag{2.4}$$

Where,

μ is the carrier mobility,

C_{ox} is the gate capacitance,

V_t is the threshold voltage, and

V_{gs} is the gate-source voltage.

To maintain higher I_{ds} we need to lower V_{th} as we lower V_{dd} (or V_{gs}). However, lowering V_{th} results in an exponential increase in the sub-threshold leakage current as indicated by Equation 2.3. Thus the methods to lower dynamic power and leakage power in a device counteract each other. This situation has worsened for 65 nm and lower CMOS process technologies as the static power is equal to or more than dynamic power in the device. Various techniques have been developed to keep both active and leakage power under control. In the

next section, some of the effective power and energy reduction methodologies are described. The intent is to focus on these particular methodologies since the work presented in this thesis builds on these methodologies.

2.3 Techniques for Reducing Dynamic Power

The dynamic power [44] of a circuit in which all gate outputs switch exactly once per clock cycle will be $\frac{1}{2} \cdot C_{load} \cdot V_{dd}^2 \cdot f$, where C_{load} is the switched capacitance, V_{dd} is the supply voltage, and f is the clock frequency. However, most of the transistors in a circuit rarely switch from most input changes. Hence, a constant called the activity factor ($0 \leq \alpha \leq 1$) is used to model the average switching activity in the circuit. Using α, the dynamic power of a circuit composed of CMOS transistors can be estimated as [21]:

$$P_{dyn} = \alpha \cdot C_{load} \cdot V_{dd}^2 \cdot f \tag{2.5}$$

The importance of this equation lies in pointing us towards the fundamental mechanisms of reducing switching power. Figure 2.9 shows that one scheme is by reducing the activity factor α. The question here is: how to achieve the same functionality by switching only a minimal number of transistors? Techniques to do this span several design hierarchy levels, from the synthesis level, where, for example, we can encode states so that the most frequent transitions occur with minimal bit switches, to the algorithmic level, where, for example, changing the sorting algorithm from insertion sort to quick sort, will asymptotically reduce the resulting switching activity. The second fundamental scheme is to reduce the load capacitance, C_{load}. This can be done by using smaller transistors with low capacitances in non-critical parts of the circuit. Reducing the frequency of operation f will cause a linear reduction in dynamic power, but reducing the supply voltage V_{dd} will cause a quadratic reduction. In the following sections we discuss some of the established and effective mechanisms for dynamic power reduction.

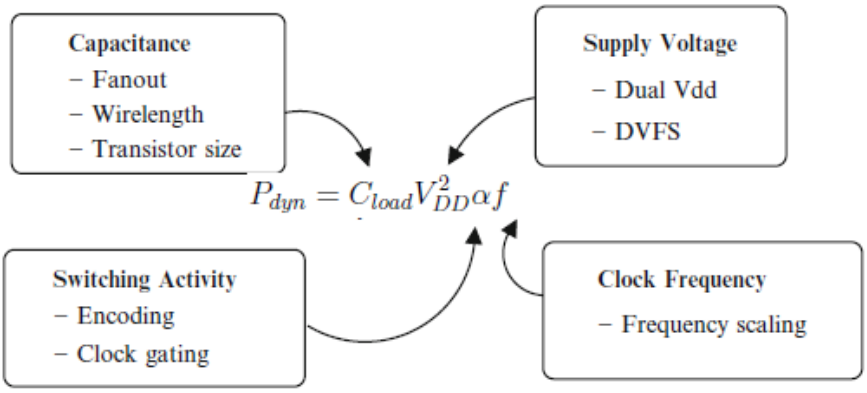


Figure 2.9: Fundamental techniques to reduce dynamic power.

2.3.1 Gate Sizing

The power dissipated by a gate is directly proportional to its capacitive load C_{load}, whose main components [44] are:

1. Output capacitance of the gate itself (due to parasitics).

2. The wire capacitance.

3. Input capacitance of the gates in its fanout.

The output and input capacitances of gates are proportional to the gate size. Reducing the gate size reduces its capacitance, but increases its delay. Therefore, in order to preserve the timing behavior of the circuit, not all gates can be made smaller; only the ones that do not belong to a critical path can be slowed down. Any gate re-sizing method to reduce the power dissipated by a circuit will heavily depend on the accuracy of the timing analysis tool in calculating the true delay of the circuit paths, and also discovering false paths. Delay calculation is relatively easier. A circuit is modeled as a directed acyclic graph. The vertices and edges of the graph represent the components and the connection respectively between the components in the design. The weight associated with a vertex (an edge) is the delay of the corresponding component (connection). The delay of a path is represented by the sum of the weights of all vertices's and edges in the path. The arrival time at the output of a

gate is computed by the length of the longest path from the primary inputs to this gate. For a given delay constraint on the primary outputs, the required time is the time at which the output of the gate is required to be stable. The time slack is defined as the difference of the required time and the arrival time of a gate. If the time slack is greater than zero, the gate can be down-sized.

2.3.2 Clock Gating

Clock signals are omnipresent in synchronous circuits. The clock signal is used in a majority of the circuit blocks, and since it switches every cycle, it has an activity factor of 1. Consequently, the clock network ends up consuming a huge fraction of the on-chip dynamic power. Clock gating has been heavily used in reducing the power consumption of the clock network by limiting its activity factor. Fundamentally, clock gating reduces the dynamic power dissipation by disconnecting the clock from an unused circuit block.

Traditionally, the system clock is connected to the clock input on every flip-flop in the design. This results in three major components of power consumption [44]:

1. Power consumed by combinatorial logic whose values are changing on each clock edge.

2. Power consumed by flip-flops has a non-zero value even if the inputs to flip-flops are steady, and the internal state of the flip-flops is constant.

3. Power consumed by the clock buffer tree in the design. Clock gating has the potential of reducing both the power consumed by flip-flops and the power consumed by the clock distribution network.

Clock gating works by identifying groups of flip-flops sharing a common *enable* signal (which indicates that a new value should be clocked into the flip-flops). This enable signal is ANDed with the clock to generate the *gated clock*, which is fed to the clock ports of all of the flip-flops that had the common enable signal. In Figure 2.10, the *sel* signal encodes whether the latch retains its earlier value, or takes a new input. This *sel* signal is ANDed with the *clk*

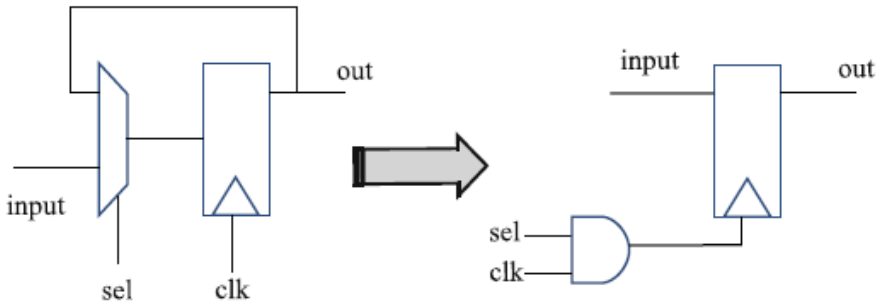


Figure 2.10: In its simplest form, clock gating can be implemented by finding out the signal that determines whether the latch will have a new data at the end of the cycle. If not, the clock is disabled using the signal.

signal to generate the gated clock for the latch. This transformation preserves the functional correctness of the circuit, and therefore does not increase the burden of verification. This simple transformation can reduce the dynamic power of a synchronous circuit by 5-10%.

There are several considerations in implementing clock gating. First, the enable signal should remain stable when the clock is high and can only switch when the clock is in its low phase. Second, in order to guarantee correct functioning of the logic implementation after the gated-clock, it should be turned on in time and glitches on the gated clock should be avoided. Third, the AND gate may result in additional clock skew. For high-performance design with short-clock cycle time, the clock skew could be significant and needs to be taken into careful consideration.

An important consideration in the implementation of clock gating for ASIC designers is the granularity of clock gating. Clock gating in its simplest form is shown in Figure 2.10. At this level, it is relatively easy to identify the enable logic. In a pipelined design, the effect of clock gating can be multiplied. If the inputs to one pipeline stage remain the same, then all the later pipeline stages can also be frozen. Figure 2.11 shows the same clock gating logic being used for gating multiple pipeline stages. This is a multi-cycle optimization with multiple implementation trade offs, and can can save significant power, typically reducing switching activity by 15-25%.

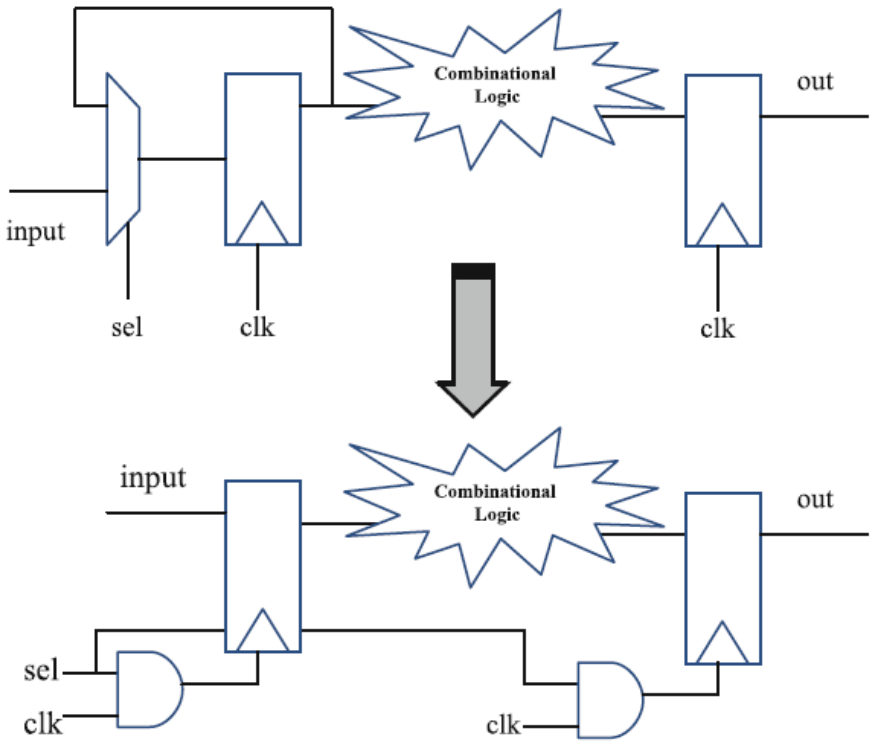
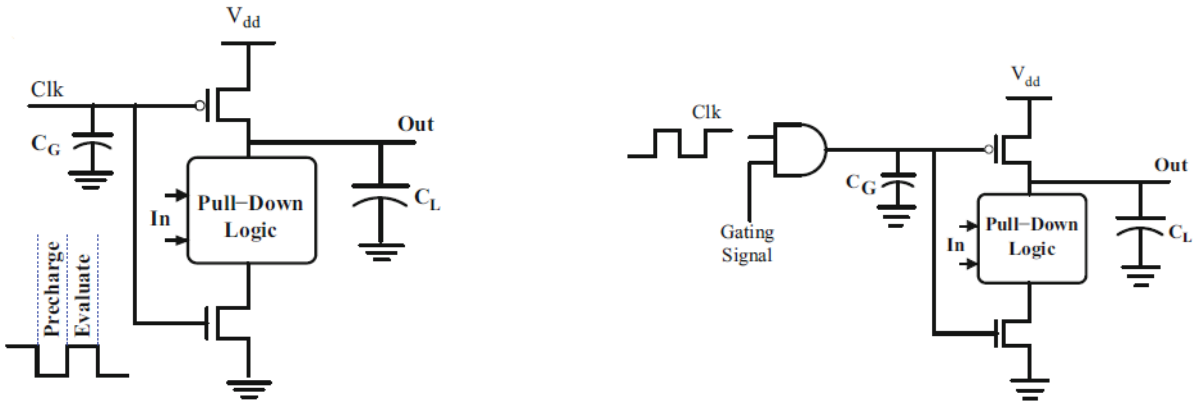


Figure 2.11: In pipelined designs, the effectiveness of clock gating can be multiplied. If the inputs to a pipeline stage remain the same, then the clock to the later stages can also be frozen.

Apart from pipeline latches, clock gating is also used for reducing power consumption in dynamic logic. Dynamic CMOS logic is sometimes preferred over static CMOS for building high speed circuitry such as execution units and address decoders. Unlike static logic, dynamic logic uses a clock to implement the combinational circuits. Dynamic logic works in two phases, precharge and evaluate. During precharge (when the clock signal is low) the load capacitance is charged. During the evaluate phase (clock is high), depending on the inputs to the pull-down logic, the capacitance is discharged.

Figure 2.12 shows the gating technique applied to a dynamic logic block. In Figure 2.12a, when the clock signal is applied, the dynamic logic undergoes precharge and evaluate phases (charging the capacitances C_G and C_{load}) to evaluate the input In, so even if the input does not change, the power is dissipated to re-evaluate the same. To avoid such redundant computation, the clock port is gated as shown in Figure 2.12b. In this case, when the input



(a) Dynamic CMOS logic

(b) Clock-gated dynamic CMOS logic

Figure 2.12: Gating technique applied to a dynamic logic block.

does not change or when the output is not used, the gating signal is enabled, which prevents the logic from evaluating the inputs and thereby reduces dynamic power dissipation. An additional AND gate is introduced to facilitate clock gating. This additional logic presents its own capacitance and hence dissipates power, but compared to the power saved by preventing the charging of capacitances C_G and C_{load} (usually large for complex execution units), the AND gate power is negligible.

Clock gating at coarse granularity or system level is much more difficult to automate, and designers have to implement it in the system functionality. For example, sleep modes in a cell phone may strategically disable the display, keyboard, or radio depending on the phone's current operational mode. System level clock-gating shuts off entire RTL blocks. Because large sections of logic are not switching for many cycles, it has the most potential to save power. However, it may result in inductive power issues due to higher $\partial i/\partial t$, since large groups of circuits are turned on/off simultaneously. In contrast, local clock gating is more effective in reducing the worst-case switching power, and also suffers less from $\partial i/\partial t$ issues. However, local clock gating may lead to frequent toggling of the clock-gated circuit between enable and disable states, as well as higher area, power, and routing overhead, especially when the clock-gating control circuitry is comparable with the clock-gated logic itself.

2.3.3 Voltage and Frequency Scaling

Dynamic power is proportional to the square of the operating voltage. Therefore, reducing the voltage significantly improves the power consumption. Furthermore, since frequency is directly proportional to supply voltage, the frequency of the circuit can also be lowered, and thereby a cubic power reduction is possible. However, the delay of a circuit also depends on the supply voltage as follows:

$$\tau = k \cdot C_{load} \cdot \frac{V_{dd}}{(V_{dd} - V_t)^2} \tag{2.6}$$

where τ is the circuit delay, k is the gain factor, C_{load} is the load capacitance, V_{dd} is the supply voltage, and V_t is the threshold voltage. Thus, by reducing the supply voltage, although we can achieve cubic power reduction, the execution time increases. The main challenge in achieving power reduction through voltage and frequency scaling is therefore to obtain power reduction while meeting all the timing constraints.

Simple analysis shows that if there is slack in execution time, executing as slow as possible, while just meeting the timing constraints is more dynamic-power-efficient than executing as fast as possible and then idling for the remaining time. This is the main idea that is used in exploiting the power reduction that arises due to the cubic relationship with power, and inverse relationship with delay, of the supply voltage.

One approach to recover the lost performance is by scaling down the threshold voltage to the same extent as the supply voltage. This allows the circuit to deliver the same performance at a lower V_{dd}. However, smaller threshold voltages lead to smaller noise margins and increased leakage current. Furthermore, this cubic relationship holds only for a limited range of V_t scaling. The quadratic relationship between energy and V_{dd} deviates as V_{dd} is scaled down into the sub-threshold voltage level. In the sub-threshold region, while the dynamic power still reduces quadratically with voltage, the sub-threshold leakage current increases exponentially with the supply voltage. Hence dynamic and leakage power become

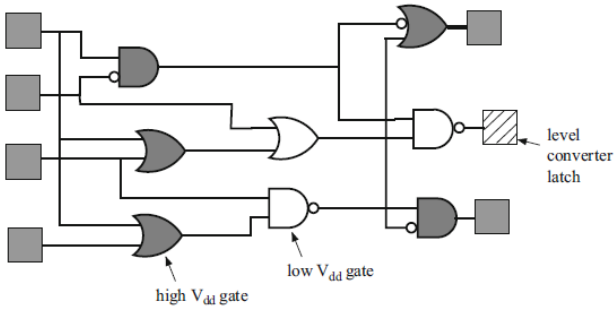
comparable in the sub-threshold voltage region, and therefore, just in time completion is not energy inefficient. In practice, extending the voltage range below half V_{dd} is effective, but extending this range to sub-threshold operations may not be beneficial.

2.4 Techniques for Reducing Short Circuit Power

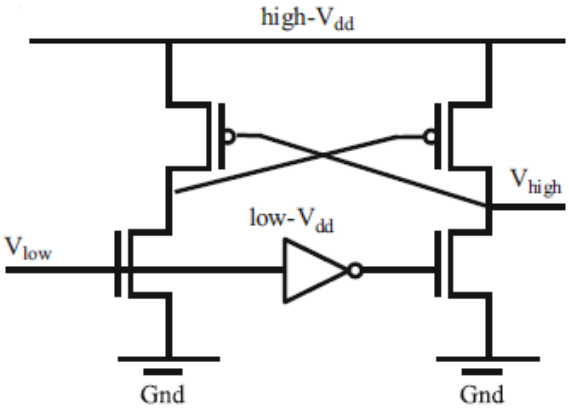
Short circuit power is directly proportional to gate rise time and fall time. Therefore, reducing the input transition times will decrease the short circuit current component. However, propagation delay requirements have to be considered while doing so. Short circuit currents are significant when the rise/fall time at the input of a gate is much larger than the output rise/fall time. This is because the short circuit path will be active for a longer period of time. To minimize the total average short circuit current, it is desirable to have equal input and output edge times. In this case, the power consumed by the short circuit current is typically less than 10% of the total dynamic power. An important point to note is that if the supply is lowered to below the sum of the thresholds of the transistors, $V_{dd}<V_{Tn}+|V_{Tp}|$, the short-circuit currents can be eliminated because both devices will never be on at the same time for any input voltage value.

2.5 Techniques for Reducing Leakage Power

In order to contain the increase in the dynamic power, the supply V_{dd} has undergone a continuous reduction in successive technology generations. Along with V_{dd}, V_t must also be scaled down, which results in an exponential increase in leakage power. Consequently, leakage power has become a significant contributor in the total chip power dissipation. Leakage power reduction techniques are especially important for handhold devices such as cell phones, which are on, but not active most of the time. Consequently, even though such devices dissipate minimal dynamic energy, leakage power becomes a significant contributor in their power equation. Some of the fundamental techniques to reduce leakage power [44] are discussed in the following sections.



(a) Multiple supply-voltage pipeline stage.



(b) Level converter latch

Figure 2.13: Using multiple V_{dd}'s essentially reduces the power consumption by exploiting the slack in the circuit. However, it requires a level converter.

2.5.1 Multiple Supply Voltages

The multiple supply system provides a high-voltage supply for high-performance circuits and a low-voltage supply for low-performance circuits. In a dual V_{dd} circuit, the reduced voltage (low-V_{dd}) is applied to the circuit on non-critical paths, while the original voltage (high-V_{dd}) is applied to the circuit on critical paths. Since the critical path of the circuit is unchanged, this transformation preserves the circuit performance. If a gate supplied with low-V_{dd} drives a gate supplied with high-V_{dd}, the pMOS may never turn off. Therefore a level converter is required whenever a module at the lower supply drives a gate at the higher supply (step-up). Level converters are not needed for a step-down change in voltage. The overhead of level converters can be mitigated by doing conversions at register boundaries and embedding the level conversion inside the latch. Figure 2.13a shows a pipeline stage in which some of the paths have low-V_{dd} gates. These are shown in a darker shade in the figure. Notice that some high-V_{dd} gates drive low-V_{dd}, but not vice versa. The transition from low to high V_{dd} is condensed into the level converter latches shown in the figure. A simple design of level converter latches is shown in Figure 2.13b [44].

Essentially, the multiple V_{dd} approach reduces power by utilizing excessive slack in a circuit. Clearly, there is an optimum voltage difference between the two V_{dd}'s. If the

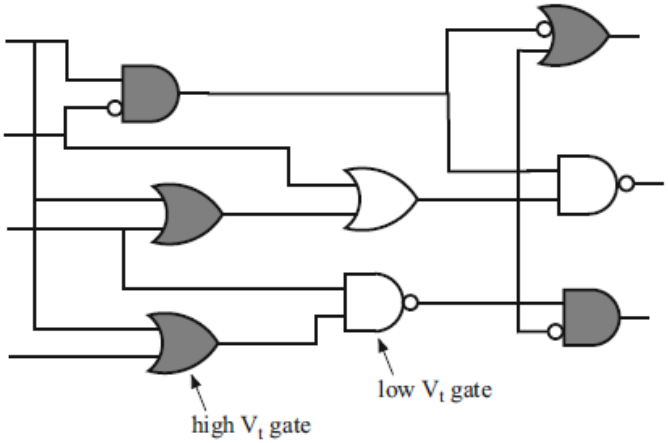


Figure 2.14: Multiple V_t technology is very effective in power reduction without the overhead of level converters. The white gates are implemented using low-V_t transistors.

difference is small, the effect of power reduction is small, while if the difference is large, there are few logic circuits that can use low-V_{dd}. Compared to circuits that operate at only high V_{dd}, the power is reduced. The latch circuit includes a level transition (DC-DC converter) if there is a path where a signal propagates from low V_{dd} logic to high V_{dd} logic.

To apply this technique, the circuit is typically designed using high-V_{dd} gates at first. If the propagation delay of a circuit path is less than the required clock period, the gates in the path are given low-V_{dd}. In an experimental setting [31], the dual V_{dd} system was applied on a media processor chip providing MPEG2 decoding and real time MPEG1 encoding. By setting high-V_{dd} at 3.3 volts and low-V_{dd} at 1.9 volts, system power reduction of 47% in one of the modules and 69% in the clock distribution was obtained.

2.5.2 Multiple Threshold Voltages

Multiple V_t MOS devices are used to reduce power while maintaining speed. High speed circuit paths are designed using low-V_t devices, while the high-V_t devices are applied to gates in other paths in order to reduce sub-threshold leakage current. Unlike the multiple–V_{dd} transformation, no level converter is required here as shown in Figure 2.14. In addition, multi-V_t optimization does not change the placement of the cells. The footprint and area of low-V_t and high-V_t cells are similar. This enables timing-critical paths to be swapped by

low-V_t cells easily. However, some additional fabrication steps are needed to support multiple V_t cells, which eventually lengthens the design time, increases fabrication complexity, and may reduce yield [10]. Furthermore, improper optimization of the design may utilize more low-V_t cells and hence could end up with increased power!

Several design approaches have been proposed for dual-V_t circuit design. One approach builds the entire device using low-V_t transistors at first. If the delay of a circuit path is less than the required clock period, the transistors in the path are replaced by high-V_t transistors. The second approach allows all the gates to be built with high-V_t transistors initially. If a circuit path cannot operate at a required clock speed, gates in the path are replaced by low-V_t versions. Finally, a third set of approaches target the replacement of groups of cells by high-V_t or low-V_t versions at one go.

In one interesting incremental scheme [48], the design is initially optimized using the higher threshold voltage library only. Then, the multi-V_t optimization computes the power-performance trade-off curve up to the maximum allowable leakage power limit for the next lower threshold voltage library. Subsequently, the optimization starts from the most critical slack end of this power-performance curve and switches the most critical gate to next equivalent low-V_t version. This may increase the leakage in the design beyond the maximum permissible leakage power. To compensate for this, the algorithm picks the least critical gate from the other end of the power-performance curve and substitutes it with its high-V_t version. If this does not bring the leakage power below the allowed limit, it traverses further from the curve (from least critical towards most critical) substituting gates with high-V_t gates, until the leakage limit is satisfied. Then the algorithm continues with the second most critical cell and switches it to the low-V_t version. The iterations continue until we can no longer replace any gate with the low-V_t version without violating the leakage power limit. The multi-V_t approach is very effective. In a 16-bit ripple-carry adder, the active-leakage current was reduced to one-third that of the all low-V_t adder [10].

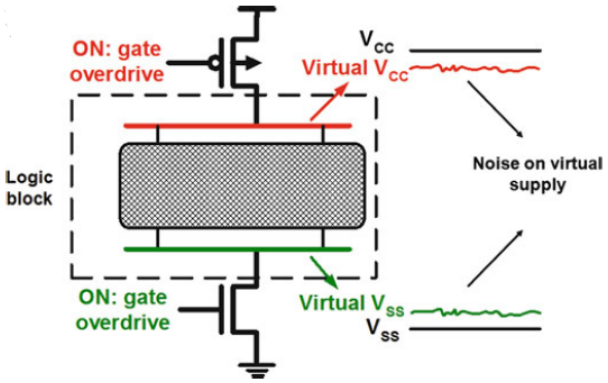
2.5.3 Adaptive Body Biasing

One efficient method for reducing power consumption is to use low supply voltage and low threshold voltage without losing performance. But increase in the lower threshold voltage devices leads to increased sub threshold leakage and hence more standby power consumption. One solution to this problem is adaptive body biasing (ABB). The substrate bias to the n-type well of a pMOS transistor is termed V_{bp} and the bias to the p-type well of an nMOS transistor is termed V_{bn}. The voltage between V_{dd} and V_{bp}, or between GND and V_{bn} is termed V_{bb}. In the active mode, the transistors are made to operate at low-V_{dd} and low-V_t for high performance. The fluctuations in V_t are reduced by an adaptive system that constantly monitors the leakage current, and modulates V_{bb} to force the leakage current to be constant. In the idle state, leakage current is blocked by raising the effective threshold voltage V_t by applying substrate bias V_{bb}.

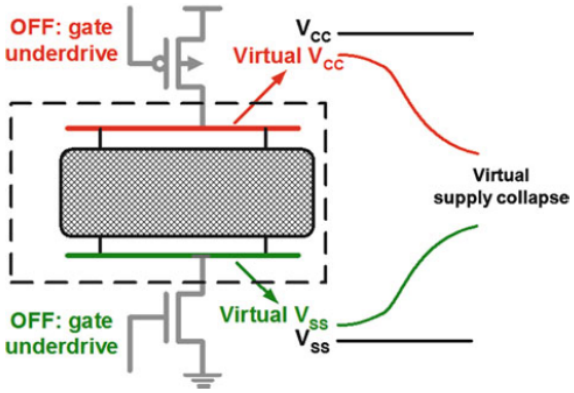
The ABB technique is very effective in reducing power consumption in the idle state, with the flexibility of even increasing the performance in the active state. While the area and power overhead of the sensing and control circuitry are shown to be negligible, there are some manufacturing-related drawbacks of these devices [58]. ABB requires either twin well or triple well technology to achieve different substrate bias voltage levels in different parts of the IC. Experiments applying ABB to a discrete cosine transform processor reported a small 5% area overhead. The substrate-bias current of V_{bb} control is less than 0.1% of the total current, a small power penalty.

2.5.4 Power Gating

Power Gating is an extremely effective scheme for reducing the leakage power of idle circuit blocks. The power (V_{dd}) to circuit blocks that are not in use is temporarily turned off to reduce the leakage power. When the circuit block is required for operation, power is supplied once again. During the temporary shutdown time, the circuit block is not operational



(a) Active mode: in the on state, the circuit sees a virtual V_{cc} and virtual V_{ss}, which are very close to the actual V_{cc}, and V_{ss} respectively.



(b) Idle mode: in the off state, both the virtual V_{cc} and virtual V_{ss} go to a floating state.

Figure 2.15: Implementation of power gating technique in pMOS transistor.

it is in *low power* or *inactive mode*. Thus, the goal of power gating is to minimize leakage power by temporarily cutting-off power to selective blocks that are not active.

As shown in Figure 2.15 [44], power gating is implemented by a pMOS transistor as a header switch to shut off power supply to parts of a design in standby or sleep mode. nMOS footer switches can also be used as sleep transistors. Inserting the sleep transistors splits the chip's power network into two parts: a *permanent power network* connected to the power supply and a *virtual power network* that drives the cells and can be turned off.

The biggest challenge in power gating is the size of the power gate transistor. The power gate size must be selected to handle the required amount of switching current at any given time. The gate must be big enough such that there is no measurable voltage (IR) drop due to it. Generally, we use 3X the switching capacitance for the gate size as a rule of thumb.

Since the power gating transistors are rather large, the slew rate is also large, and it takes more time to switch the circuit on and off. This has a direct implication on the effectiveness of power gating. Since it takes a long time for the power-gated circuit to transition in and out of the low power mode, it is not profitable to power gate large circuits for short idle durations. This implies that either we implement power gating at fine granularity, which

increases the overhead of gating, or find large idle durations for coarse-grain power gating, which are fewer and more difficult to discover. In addition, coarse-grain power gating results in a large switched capacitance, and the resulting rush current can compromise the power network integrity. The circuit needs to be switched in stages in order to prevent this. Finally, since power gates are made of active transistors, the leakage of the power gating transistor is an important consideration in maximizing power savings.

For fine-grain power-gating, adding a sleep transistor to every cell that is to be turned off imposes a large area penalty. Fine-grain power gating encapsulates the switching transistor as a part of the standard cell logic. Since switching transistors are integrated into the standard cell design, they can be easily be handled by EDA tools for implementation. Fine-grain power gating is an elegant methodology resulting in up to 10X leakage reduction.

In contrast, the coarse-grained approach implements the grid style sleep transistors which drive cells locally through shared virtual power networks. This approach is less sensitive to process variations, introduces less IR-drop variation, and imposes a smaller area overhead than the fine-grain implementations. In coarse-grain power gating, the power-gating transistor is a part of the power distribution network rather than the standard cell.

2.6 Low Power Metrics for CMOS Designs

When optimizing a design for low power it is necessary to have a metric that can be used to compare different alternatives. The most obvious choice is power, measured in watts. Power is the rate of energy use, or $P = \partial E/\partial T$. A more useful definition [25], however, is average power, or the energy spent to perform a particular operation divided by the time taken to perform the operation $P_{avg} = E_{op}/T_{op}$. How to define the operation of interest is arbitrary and depends on what is being compared. In the case of a processor, it could be the energy to run a benchmark to completion, or the energy to execute an instructionas long as all processors compared execute the same instructions.

Power is important for two reasons. The first is that it determines what kind of package can be used for the chip. For example, a small plastic package, the cheapest form of packaging, can only dissipate a few watts. A processor which dissipates more than that will have to be sold in a more expensive package. The second reason power is important is because it limits how long the system battery will last. But power as a metric of goodness of low-power designs has some drawbacks. The most important drawback is that power is proportional to the operation rate, so one can reduce the power by slowing down the system. In CMOS circuits this is very easy to do, one simply reduces the clock frequency.

Regardless of what definition of an operation one uses, the basic problem with power remains, that power decreases simply by extending the time required to complete an operation. Power, therefore, is only a good metric to compare processors that have similar performance levels. If two processors can perform computation at the same rate, then clearly whichever dissipates less power is more desirable. If the processors run at different rates the slower processor will almost always be lower power.

An alternative metric is the energy per operation, measured in J/Cycle. Energy per operation of a circuit is a key parameter for energy efficiency in ultra-low power applications. Because computing workload is characterized in terms of clock cycles, this measure directly relates to the energy consumption of workload.

From an optimization standpoint one more possible metric is also the product of energy and delay, measured in *joule-sec*. Optimizing the energy-delay product will prevent the designer from trading off a large amount of performance for a small savings in energy, or vice versa.

In this research, we characterize various Intel Processors and we use a new performance metric called cycle efficiency, η [55] to evaluate the performance and energy efficiency of the processor.

2.6.1 Power Delay Product (PDP)

The propagation delay and the power consumption of a gate are relatedthe propagation delay is mostly determined by the speed at which a given amount of energy can be stored on the gate capacitors. The faster the energy transfer (or the higher the power consumption), the faster the gate. For a given technology and gate topology, the product of power consumption and propagation delay is generally a constant. This product is called the power-delay product (or PDP) and can be considered as a quality measure for a switching device. The PDP is simply the energy consumed by the gate per switching event.

$$PDP = P_{avg} \cdot t_p \tag{2.7}$$

The PDP is a measure of energy, as is apparent from the units ($watts \times sec = joule$). Assuming that the gate is switched at its maximum possible rate of $f_{max} = 1/(2t_p)$, and ignoring the contributions of the static and direct-path currents to the power consumption, we find

$$PDP = C_{Load} \cdot V_{dd}^2 \cdot f_{max} \cdot t_p = \frac{C_{Load} \cdot V_{dd}^2}{2} \tag{2.8}$$

The PDP stands for the average energy consumed per switching event (that is, for a 0→ 1, or a 1→0 transition). Remember that earlier we had defined E_{av} as the average energy per switching cycle (or per energy-consuming event). As each inverter cycle contains a 0→1, and a 1→0 transition, E_{av} hence is twice the PDP.

2.6.2 Energy Delay Product

The validity of the PDP as a quality metric for a process technology or gate topology is questionable. It measures the energy needed to switch the gate, which is an important property for sure. Yet for a given structure, this number can be made arbitrarily low by

reducing the supply voltage. From this perspective, the optimum voltage to run the circuit would be the lowest possible value that still ensures functionality. This comes at the major expense in performance, at discussed earlier. A more relevant metric should combine a measure of performance and energy. The energy-delay product (EDP) does exactly that.

$$EDP = PDP \cdot t_p = P_{avg} \cdot t_p^2 = \frac{C_{load} \cdot V_{dd}^2}{2} \cdot t_p \tag{2.9}$$

It is worth analyzing the voltage dependence of the EDP. Higher supply voltages reduce delay, but harm the energy, and the opposite is true for low voltages. An optimum operation point should hence exist. Assuming that nMOS and pMOS transistors have comparable threshold and saturation voltages, we can define the propagation delay expression as [25]:

$$t_p = \frac{\alpha \cdot C_{load} \cdot V_{dd}}{V_{dd} - V_{Te}} \tag{2.10}$$

where $V_{Te} = V_T + V_{DSAT}/2$, and α is a technology parameter. Combining Equation 2.9 and Equation 2.10,

$$EDP = \frac{\alpha \cdot C_{load}^2 \cdot V_{dd}^3}{2(V_{dd} - V_{Te})} \tag{2.11}$$

This equation is only accurate as long as the devices remain in velocity saturation, which is probably not the case for the lower supply voltages. This introduces some inaccuracy in the analysis, but will not distort the overall result.

The optimum supply voltage can be obtained by taking the derivative of Equation 2.11 with respect to V_{dd}, and equating the result to 0.

$$V_{ddopt} = \frac{3}{2} \cdot V_{Te} \tag{2.12}$$

The remarkable outcome from this analysis is the low value of the supply voltage that simultaneously optimizes performance and energy. For sub-micron technologies with thresholds in the range of 0.5 volts, the optimum supply is situated around 1 volts.

2.6.3 Cycle Efficiency

Cycle efficiency is defined as performance per unit of energy. To increase this efficiency it is required that the fundamental energy of operations be reduced. Further, power is defined as the rate of energy consumption (watts $\equiv$ J/second) and is directly affected by the performance. This distinction between power and energy is important because what may seem like a trade-off may just be a modulation in performance resulting in changes in power consumption.

The performance (inverse of time) can be called *time efficiency* just as cycle efficiency (inverse of energy per cycle) is *energy efficiency*. If we regard the clock cycle as a unit of work that a processor performs, then it means work done in a time period $1/f$, where f is the frequency in cycles per second or hertz (Hz). A clock cycle also means certain amount of energy or energy per cycle (EPC). We define cycle efficiency, $\eta = 1/EPC$, its unit being cycles per joule [55], [56]. Thus, a clock cycle means $1/f$ second in time and $1/\eta$ joule in energy. Consider a program being run on a processor and suppose it takes c clock cycles to execute. Then we have,

$$\text{Execution time} = \frac{c}{f} \tag{2.13}$$

$$\text{Energy consumed} = \frac{c}{\eta} \tag{2.14}$$

where, η is cycle efficiency of the processor in cycles per joule. Equation 2.13 gives the time performance of the processor as,

$$\text{Performance in time} = \frac{1}{\text{Execution time}} = \frac{f}{c} \tag{2.15}$$

Similarly, Equation 2.5 gives the energy performance as,

$$\text{Performance in energy} = \frac{1}{\text{Energy consumed}} = \frac{c}{\eta} \tag{2.16}$$

Clearly, cycle efficiency (η) characterizes the energy performance in a similar way as frequency (f) characterizes the time performance. These two performance parameters are related to each other by the power being consumed, as follows:

$$\text{Power} = \frac{f}{\eta} \tag{2.17}$$

For a computing task, f is the rate of execution in time and η is the rate of execution in energy. Consider the analogy of automobiles; f is analogous to speed in miles per hour (mph) and η is analogous to miles per gallon (mpg). A practical way to see the cycle efficiency is: $f \rightarrow$ mph, $\eta \rightarrow$mpg. These two parameters allow the designer to effectively manage time and energy of the system.

Chapter 3

Technology Assessment Methodology

To show our proposed power management method, a certain set of procedures was carried out which are described in various sections of this chapter. The reason for selecting the micro-benchmark adder circuit is described in the next section followed by introduction of various tools and techniques used for circuit modeling, netlist generation, simulation, process variation, and result analysis. There is a wide variety of CMOS predictive technology models therefore what models are selected to conduct the experiment and why they are important are explained further in this chapter.

3.1 Ripple Carry Adder Benchmark Circuit

A ripple carry adder [37] is a digital circuit that produces the arithmetic sum of two binary numbers. It can be constructed with full adders (Figure 3.1) connected in cascade, with the carry output from each full adder connected to the carry input of the next full adder in the chain. Figure 3.2 shows the interconnection of n-bit full adder (FA) circuits to provide a n-bit ripple carry adder. Notice from Figure 3.2 that the input is from the right side because the first cell traditionally represents the least significant bit (LSB). Bits a_0 and b_0 in the figure represent the least significant bits of the numbers to be added. The sum output is represented by the bits s_n-s_0.

The ripple carry adder circuit in this work is used to learn the energy and delay characteristics of the technology of the processor [46]. Usually, a simple replicable circuit or a benchmark circuit where performance and working can be easily monitored is chosen. For this thesis, a 16-bit ripple carry adder was chosen for its simple design yet it has a sufficient logic depth for the proper utilization of the design technique. The design methodology emphasizes the

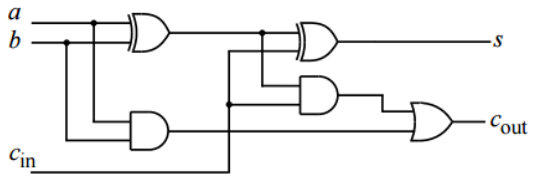


Figure 3.1: Gate implementation of full adder.

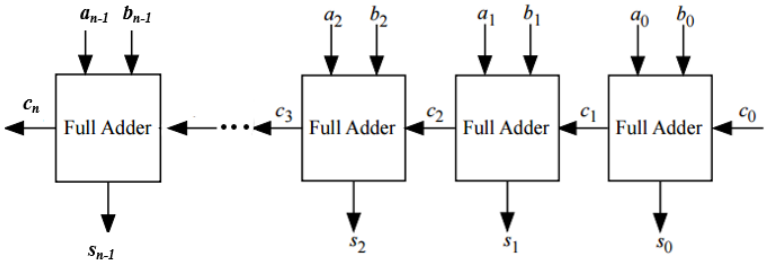


Figure 3.2: Interconnection of n-bit full adder (FA) circuits to provide a n-bit ripple carry adder (RCA).

operation of the adder in 32 nm bulk PTM CMOS technology and the results are shown along with other *predictive technology models* (PTM) [19, 65].

3.2 IC Design and Simulation Tools

In the initial phase of a CMOS product chip architecture and design, an assessment of power and performance at the technology of interest is made from the compact models provided by the silicon foundry. In the design implementation phase, circuits and physical layouts are optimized by incorporating these models in the EDA tools.

In migrating a design from one technology node to the next, or when substituting a different model for the one already in place, it is important to compare circuit behaviors from the two sets of models. Differences in device properties, parameter distributions, physical layout ground rules, and reliability models beyond those expected from pure scaling provide an early assessment on what aspects of the design will be affected the most.

Essential to the success of this approach is that the compact models do accurately capture the physical behavior of devices and circuits over the range of application conditions. It is therefore prudent to evaluate the device models after incorporating them in the chip

design environment and in EDA tools. This evaluation should be conducted over the expected range of operation for the specific chip and product design.

This section gives an introduction to the various tools and techniques that are used to conduct the experiments with the test circuit in this research. There are different tools for circuit modeling, netlist generation, simulation, process variation, and result analysis.

3.2.1 QuestaSim

QuestaSim [6] is a hardware simulation and debug environment primarily targeted at smaller ASIC and FPGA design. It is a Simulator with additional Debug capabilities targeted at complex FPGA's and SoC's. QuestaSim can be used by users who have experience with ModelSim as it shares most of the common debug features and capabilities. One of the main differences between QuestaSim and Modelsim (besides performance/capacity) is that QuestaSim is the simulation engine for the Questa Platform which includes integration of Verification Management, Formal based technologies, Questa Verification IP, Low Power Simulation and Accelerated Coverage Closure technologies. QuestaSim natively supports SystemVerilog for Testbench, UPF, UCIS, OVM/UVM.

3.2.2 Leonardo Spectrum

Leonardo Spectrum [5] is a logic synthesis tool from Mentor Graphics Corp. Logic synthesis is the process of translating a Hardware Description Language (HDL) model into a technology-specific gate-level description. Leonardo Spectrum offers design capture, VHDL and Verilog entry, register transfer level debugging for logic synthesis, constraint based optimization, timing analysis, encapsulated place and route, and schematic viewing for *complex programmable logic devices* (CPLD), *field programmable gate arrays* (FPGA), and *application specific integrated circuits* (ASIC).

3.2.3 Design Architect

Design Architect [3] is more than a computer-aided schematic capture application. It is a multi-level design environment that includes: a Schematic Editor, a Symbol Editor, and the VHDL Editor. In a multi-level design environment you can:

- Implement top-down and bottom-up design methodology
- Specify a design at different levels of abstraction, from high-level specifications to gate-level implementation
- Specify a design with different modeling techniques
- Configure and manage different design descriptions to explore alternate design implementations

Design Architect lets you create and edit logical designs that are used by downstream processes such as: board design, IC and PCB layout, and analog and digital simulation.

3.2.4 H-spice [4]

Simulation program with integrated circuit emphasis (SPICE) [42, 43] is a general purpose electronic circuit simulator used to check the integrity of circuit design and predict circuit behavior. H-spice is a circuit simulator tool derived from SPICE and designed by Synopsys Incorporated [4] in order to predict the timing, functionality, power consumption, and yield of their designs. H-spice uses a netlist file design.sp, where design is the name of your circuit, as a source file. This text file contains the circuit netlist, element models, analysis commands and output commands. Execution of H-spice [4] produces a number of files depending on user-specified options. By use of the appropriate options, files are produced which act as the input files for meta waves for displaying, analyzing, and printing results from H-spice.

3.2.5 EZwave

EZwave H-spice [4] is a high-capacity, high-performance graphical waveform environment for displaying and analyzing complex analog, digital, RF, and mixed-signal simulation results. EZwave can analyze time or frequency domain waveform of any type: analog, digital, eye diagram, smith chart, polar or complex chart, and histogram.

3.3 Predictive Technology Models (PTM) of Conventional CMOS Devices

3.3.1 Why Predictive Technology Models (PTM) are Important?

The scaling of CMOS technology has been the driving force of the semiconductor industry during past five decades, with the minimum feature size expected to reach 10 nm in coming years [1]. Beyond that benchmark, the present scaling approach may have to take a different route, in order to overcome dramatic barriers in transistor performance degradation, power consumption, process and environmental variations, and reliability issues. For instance, Figure 3.3 illustrates the scaling trends of the maximum on-state current (I_{on}) and the off-state leakage current (I_{off}), from a comprehensive set of published data [36] [14]. From the 0.5 μm node to the 32 nm node, the increase in I_{on} is smaller than 3x; meanwhile, I_{off} increases by more than six orders. Such a dramatic reduction in the ratio of I_{on}/I_{off} significantly affects the drivability of the device, and further influences all aspects of circuit performance, such as data stability of on-chip memory.

To continue the success of integrated circuit (IC) design, the grand challenge to IC community is to identify unconventional materials and structures, such as carbon-based electronics, integrate them into the large-scale circuit architecture, and enable continuous growth of chip scale and performance [1]. Different from previous design paradigm, today's competitive circuit design and research must begin before a future generation of CMOS technology is fully developed, in order to successfully manage the development cost and guarantee the time to market. Figure 3.4 highlights the paradigm shift toward concurrent technology

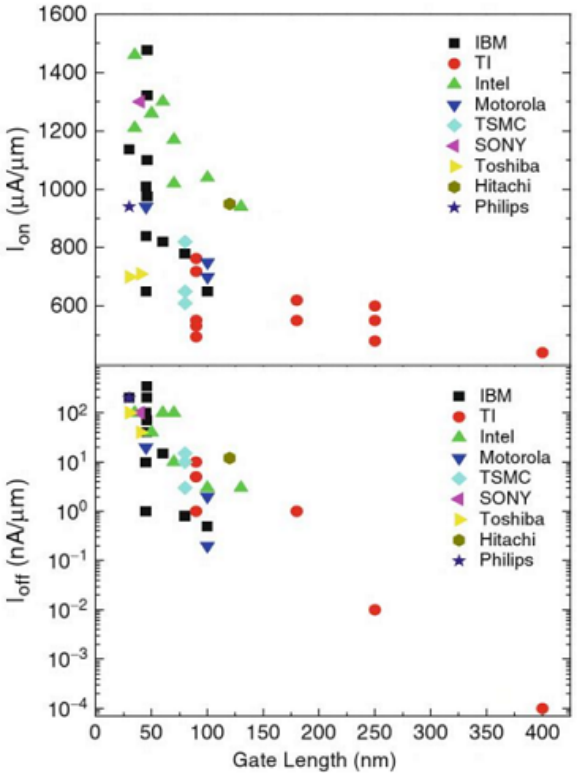


Figure 3.3: The scaling trends of I_{on} and I_{off} [19].

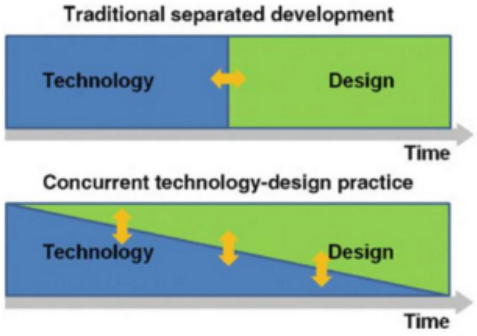


Figure 3.4: The new paradigm of joint technology-design research [19].

and design research [33]. In this context, the *predictive technology model* (PTM) [19, 65], which bridges the process/material development and circuit simulation through device modeling, is essential to assessing the potential and limits of new technology and to supporting early design prototyping. PTM is the critical interface between technology innovation and IC design exploration, as shown in Figure 3.5. Coupled with circuit simulation tools, they significantly improve design productivity, providing the insight into the relationship between technology/design choices and circuit performance. In order to guarantee the quality of the prediction, PTM should be scalable with the latest technology advances, accurate across a wide range of process uncertainties and operation conditions, and efficient for large-scale computation.

As semiconductor technology scales into the nanoscale regime, these modeling demands are tremendously challenged, especially by the introduction of alternative device materials and structures, as well as the ever-increasing amount of process variations. Driven by the

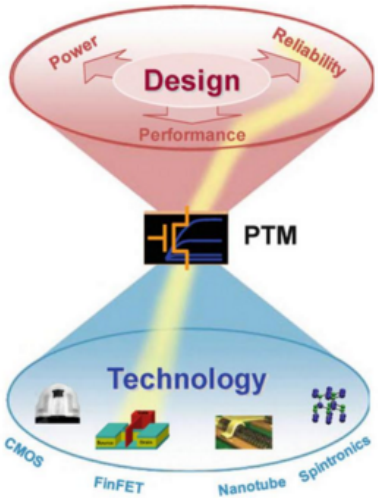


Figure 3.5: PTM [19, 65]: a bridge between technological prediction and early stage design exploration.

increasingly complex and diverse nature of the underlying technology, the overarching goal of PTM is to provide early comprehension of process choices and design opportunities, as well as to address key design needs, such as variability and reliability, for robust system integration.

Since we are managing performance of various processors in different technology nodes such as 45 nm, 32 nm and 22 nm, to carry out the characterization, two versions of *predictive technology models* (PTM) [19, 65] were chosen that are available from an Arizona State University site [7] which are widely used to carry out research experiments: Bulk MOSFET with conventional SiON/Polysilicon gate and Secondly high-K dielectric with metal gate technology, a combination known as HKMG (High-K, metal gate).

3.3.2 Benefits of High-K Metal Gate CMOS over Bulk MOSFET With Conventional SiON/Polysilicon Gate

The relentless scaling of CMOS technology has accelerated in recent years and will arguably continue toward the 10 nm regime. In the nanometer era, physical factors that previously had little or no impact on circuit performance are now becoming increasingly significant. Particular examples include process variations, transistor mobility degradation,

and power consumption. These new effects pose dramatic challenges to robust circuit design and system integration. To continue the design success and make an impact on leading products, advanced circuit design exploration must start in parallel with, or even earlier than silicon development. This new design paradigm demands predictive MOSFET models that are reasonably accurate, scalable with main process and design knobs, and correctly capture those emerging physical effects.

A predictive MOSFET model is critical for early circuit design research. To accurately predict the characteristics of nano-scale CMOS, emerging physical effects, such as process variations and correlations among model parameters, must be included. The planar bulk-silicon MOSFET has been the workhorse of the semiconductor industry over the last 40 years. However, the scaling of bulk MOSFETs becomes increasingly difficult for gate lengths below 20 nm (sub-45 nm half-pitch technology node) expected by the year 2016. As the gate length is reduced, the capacitive coupling of the channel potential to the source and drain increases relative to the gate, leading to significantly degraded short-channel effect (SCE). This manifests itself as:

- Increased off-state leakage,
- Threshold voltage (V_{th}) roll-off, i.e. smaller V_{th} at shorter gate lengths, and
- Reduction of V_{th} with increasing drain bias due to a modulation of the source-channel potential barrier by the drain voltage, also called drain-induced barrier lowering (DIBL).

To overcome these difficulties and continue the path projected by Moore's law [40], new materials need to be incorporated into the bulk CMOS structure, including high-permittivity (high-K) gate dielectric, metal gate electrodes, low- resistance source/drain, and strained Si channel for high mobility. Furthermore, more flexible process choices, such as multiple- V_{th} are required in today's integrated circuit design, in order to satisfy various design needs (e.g., low power vs. high performance). In order to maintain the relatively strong gate control of the channel potential in bulk devices, various technological improvements such as ultra-thin

gate dielectric have been necessary. Insulators that have a larger dielectric constant than silicon dioxide (referred to as high-K dielectric), such as group IV b metal silicates, e.g., hafnium and zirconium silicates and oxides are being used to reduce the gate leakage from the 45 nanometer technology node onwards. Present high performance CPUs use metal gate technology, together with high-K dielectric.

3.4 RCA Benchmark Circuit Modeling

A 16 bit adder is designed using VHDL model and its compilation and simulation is carried out using Questa Sim [6] to ensure correct operation before it is synthesized. The VHDL model was then imported into Leonardo Spectrum tool [5], which can create a simulatable netlist for the VHDL model. A circuit netlist can be created for any technology. For this thesis, the circuit was modeled in TSMC 0.18 micron technology. Spectrum translated our RTL model into a technology-specific gate-level circuit and optimized for time as a design constraint. Leonardo Spectrum generated a Verilog file which contained the properly synthesized netlist. This synthesized verilog file was then imported into the Design Architect tool [3], which gave the schematic of the 16-bit ripple carry adder using the standard TSMC cell libraries. The Design Architect tool has an internal SPICE simulator which can internally generate a SPICE netlist. This netlist is further verified and simulated with modified length and width keeping the ratios constant to match the predictive technology model files specifications. Instead of using the TSMC libraries as used by the Design Architect, we used the bulk CMOS and high-K metal gate predictive technology model (PTM) in 45 nm, 32 nm and 22 nm technology nodes [7, 19, 65]. This was done because Design Architect did not provide these technology node libraries, and the research required us to simulate circuits in the latest transistor technologies. The EZwave tool is used to view the waveforms of the probed signals after SPICE simulations.

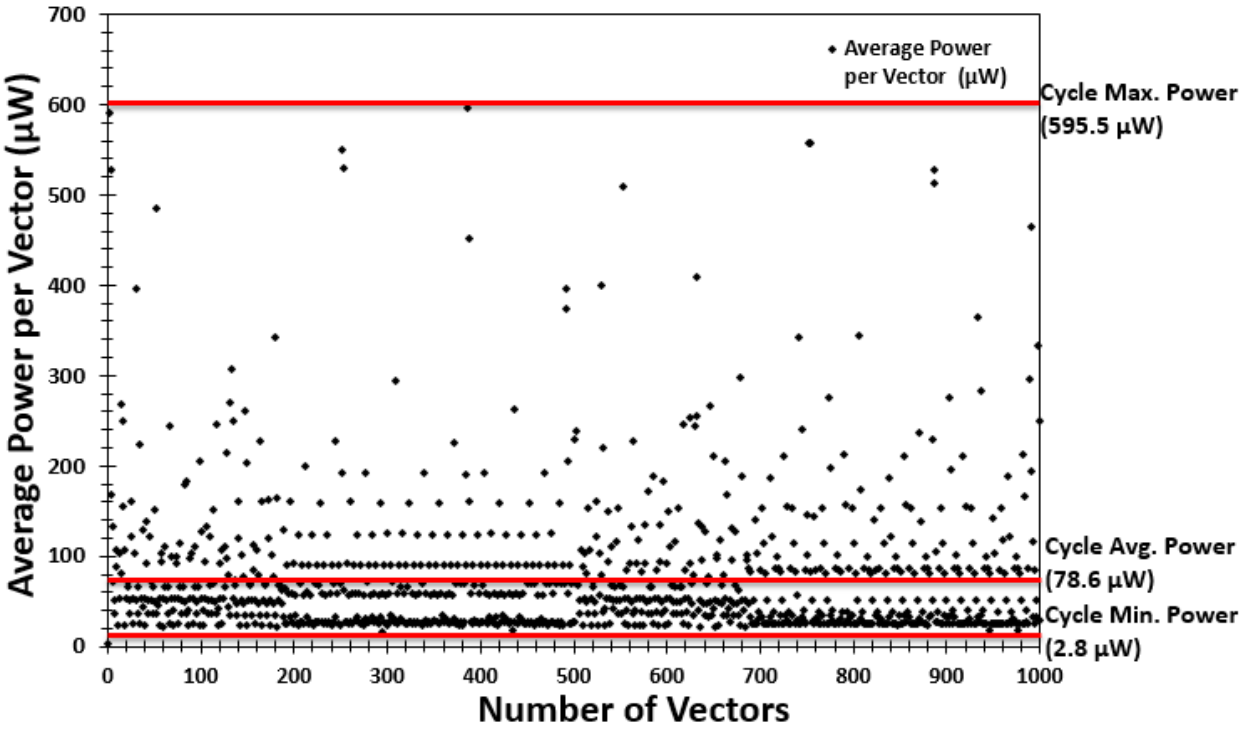


Figure 3.6: H-spice [4] simulation of 16-bit ripple carry adder in 90 nm bulk CMOS PTM [7] at V_{dd}=1.4 volts and 1.45 GHz clock frequency.

3.5 Technology Characterization of Adder

Our power management method for microprocessors entirely depends on characterization of our chosen micro benchmark adder circuit. Technology characterization is done by choosing an appropriate set of vectors and simulating the adder circuit to measure its critical path delay, power consumption and minimum energy point. We assume that the processor being characterized is large and a full scale gate-level or transistor-level model may not be readily available. Even if such a model was available, a detailed simulation at various voltages would be impractical due to high complexity. However, operational data about the processor, such as voltage, maximum clock frequency and power consumption, is available. Also, the technology of the device is specified. We, therefore, characterize the technology using known and easily analyzable adder benchmark. Then, we scale this characterization to the processor.

3.5.1 Vector Selection

Initially, 1,000 random vectors were generated using a MATLAB program [2] which resulted in some average power per vector when simulated using H-SPICE [4] as shown in Figure 3.6. Input vector selection is an important step in determining the average power and critical path delay. For our adder circuit a selected set of vectors is applied as inputs at the fastest clock possible. Below is the MATLAB program [2] that generated 1,000 random vectors:

```
function [inp,dout]=parity(n,b)
[inp,dout]=parity4uni(16) %% parity-n for unipolar; set b=1 for bipolar
    if nargin==2 && b==1,
    inp(inp==0)=-1;
    dout(dout==0)=-1;
end;
%-----------------------------
function [inp,dout]=parity4uni(n)
    if n==1,
        inp=[0;1];
        dout=[0;1];
    else
        [inpp,doutp]=parity4uni(n-1);
        [a,b]=size(inpp);
        inp=[zeros(a,1),inpp;ones(a,1),inpp];
        dout=[doutp;1-doutp];
end;
more on
```

To calculate the delay at each voltage, the critical path needs to be activated.Therefore, the first vector pair is chosen for the adder circuit such that they activate the critical path for the circuit.

The critical vector pair used for our 16 bit adder circuit is:

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Out of those 1000 random vectors simulated, 50 vector pairs were selected such that 16 consume average power, 17 consume above average power, including a peak power vector pair and 17 consume below average power including, a minimum power vector pair. Figure 3.7 shows the power profiles of 50 selected vector pairs when they were embedded in the set of 1,000 vectors and then re-simulated as 100 standalone vectors. We observe that out of 100 vectors 50 have the same power consumption while power for the other 50 changes because they are now preceded by different vectors. The reason to select vectors through such a rigorous procedure is that use of the adder circuit and random vectors assumes an overall imitation of processor characteristics. On conducting the SPICE simulations using H-spice [4], the average current consumed by the circuit was measured. It was then multiplied by voltage to give the average power dissipated by the test circuit. To determine the average energy per cycle, the average power was multiplied by the delay of the circuit. The average energy per cycle for each voltage step was calculated, tabulated and graphed.

3.5.2 Simulation Results for Ripple-Carry Adder (RCA) Circuit

In the previous section we discussed the ripple carry adder modeling using different simulation tools. This section discusses experimental results for our benchmark circuit that was designed and analyzed in 45 nm, 32 nm and 22 nm, using the H-SPICE [4] simulator with two versions of *predictive technology models* (PTM) that are available from an Arizona State University site: Bulk MOSFET with conventional SiON/Polysilicon gate and Secondly high-K dielectric with metal gate technology, a combination known as HKMG (high-K metal gate).

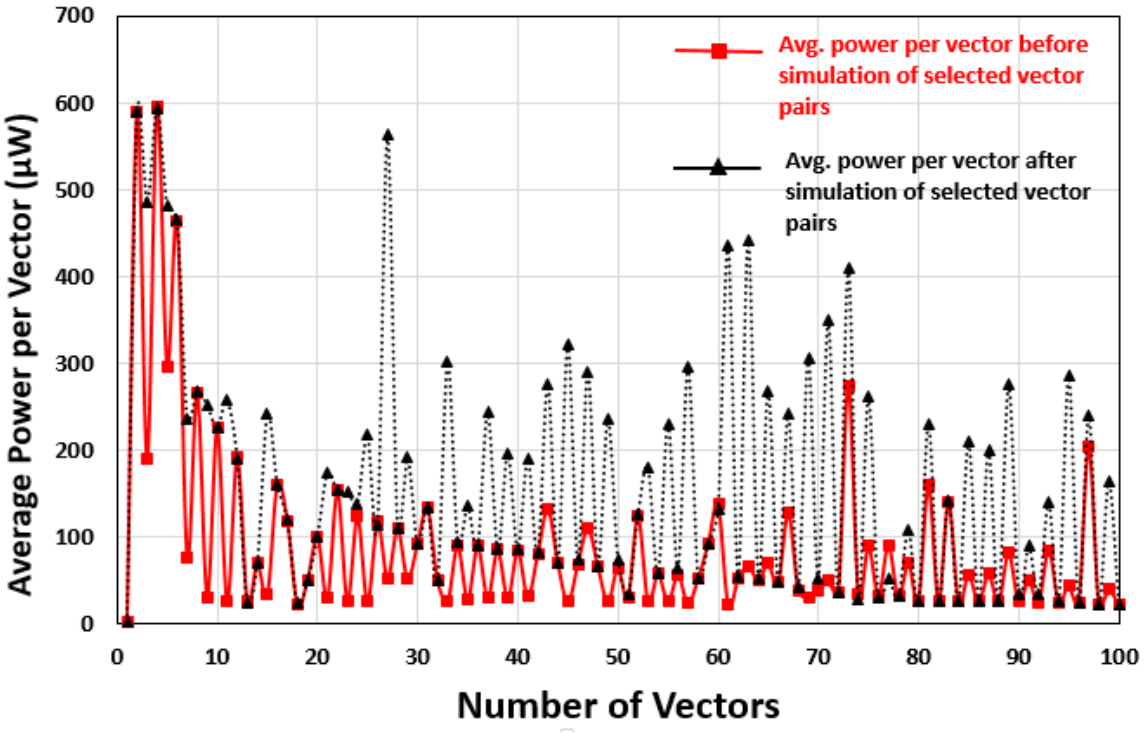


Figure 3.7: H-spice [4] simulation of 16-bit ripple carry adder with 50 input vector pairs in 90 nm bulk CMOS PTM [7] at V_{dd}=1.4 volts and 1.45 GHz clock frequency.

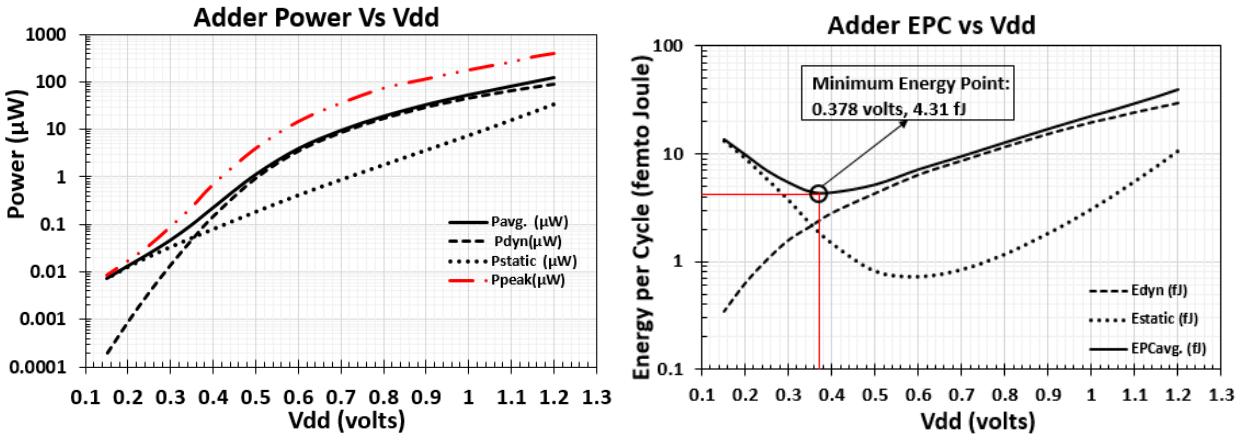
We have chosen 32 nm bulk CMOS technology as an illustrative example to show our proposed method for power management of microprocessor, therefore we are describing different aspects of CMOS design in Table 3.1 showing power from simulation: average power, dynamic power, static power and peak power; timing from simulation: critical path delay and its inverse f_{max}; and energy per cycle (dynamic, static and total) for the 16-bit ripple carry adder circuit estimated by the H-SPICE tool [4] using 32 nm bulk CMOS predictive technology model (PTM) when subjected to varying V_{dd} voltages ranging from 1.2 volts (nominal voltage) to 0.15 volts (sub-threshold region). We have several components of power for the digital CMOS circuit:

$$\begin{aligned} P_{avg} &= P_{dyn} + P_{stat} \\ &= (P_{short} + P_{switch}) + P_{stat} \qquad (3.1) \\ &= \alpha I_{sc} V_{dd} + \alpha C_L V_{dd}^2 f + I_{leak} V_{dd} \qquad (3.2) \end{aligned}$$

Table 3.1: H-SPICE [4] simulation of 16 bit ripple carry adder (RCA) for 32 nm technology node in bulk CMOS PTM [7] at different voltages (V_{dd}).

V_{dd}	Power from simulation				Timing from simulation		Energy per cycle		
volts	p_{avg} μW	p_{dyn} μW	p_{static} μW	p_{peak} μW	Critical path delay, ps	f_{max} GHz	e_{dyn} fJ	e_{static} fJ	e_{total} fJ
1.20	124.03	91.37	32.66	397.71	320.85	3.12	29.31	10.48	39.80
1.15	100.50	78.31	22.19	335.74	338.91	2.95	26.54	7.52	34.06
1.10	81.93	66.72	15.21	261.90	360.46	2.77	24.05	5.48	29.53
1.05	66.21	55.74	10.47	217.46	386.50	2.59	21.54	4.05	25.59
1.00	53.77	46.51	7.26	178.20	418.72	2.39	19.47	3.04	22.51
0.95	42.65	37.58	5.07	144.77	459.03	2.18	17.25	2.33	19.58
0.90	33.40	29.83	3.57	115.34	509.72	1.96	15.21	1.82	17.03
0.80	19.08	17.32	1.75	73.71	666.65	1.50	11.55	1.17	12.72
0.70	9.59	8.73	0.86	35.76	986.51	1.01	8.62	0.84	9.46
0.60	3.97	3.57	0.41	14.71	1792.1	0.56	6.39	0.73	7.12
0.50	1.14	0.96	0.18	4.01	4511.7	0.22	4.31	0.82	5.13
0.40	0.229	0.150	0.079	0.695	18928	0.053	2.84	1.49	4.33
0.35	0.099	0.048	0.051	0.233	44168	0.023	2.13	2.27	4.40
0.30	0.047	0.014	0.033	0.090	112760	0.009	1.60	3.75	5.35
0.25	0.025	0.004	0.021	0.036	279310	0.004	1.06	5.85	6.91
0.20	0.0136	0.0009	0.0127	0.0172	716150	0.0014	0.65	9.08	9.73
0.15	0.0074	0.0002	0.0072	0.0086	1851700	0.0005	0.35	13.27	13.62

The term P_{short} is the power consumed during gate voltage transient time, which in CMOS technology, is related to the direct path short circuit current (I_{sc}) that flows when both the NMOS and PMOS transistors are simultaneously on (or partially on), flowing directly from supply V_{dd} to ground or V_{ss}. The term, P_{switch} refers to the dynamic component of switching power due to charging and discharging of load capacitance, C_L, f is the clock frequency and α is the average switching activity factor. Imperfect cutoff of transistors leads to leakage (I_{leak}) and power dissipation (P_{static}) without any switching activity. With increasing number of gates both the total capacitance and the channel width are relevant to increased power. Figures 3.8a and 3.8b show the dynamic and leakage power and energy per cycle as functions of V_{dd}, as explained in [18, 50]. It can be observed that for the lowest power, the optimal V_{dd} is the lowest V_{dd} that the system can operate on and is limited by performance and/or robustness requirements. The power exponentially reduces as the supply voltage reduces.



(a) Average, peak, dynamic and static power for 16 bit adder

(b) Energy per cycle for 16 bit adder

Figure 3.8: Power and energy plots for 16 bit adder in 32 nm bulk CMOS from H-spice [4] simulation.

Therefore, the total average energy, that is calculated by the power delay product (PDP), also reduces when the supply voltage reduces, as shown in Figure 3.8b. This is because the power reduction is much faster than the increase of the circuit delay. For minimum energy, optimal V_{dd} is in the near-threshold region. However, interestingly, when supply voltage is less than 0.4 volts, the average energy per cycle begins to increase. This is a result of the dynamic energy per cycle decreasing with V_{dd} and leakage energy per cycle increasing as we get close to the sub-threshold region. With down scaling of V_{dd} below the threshold voltage, there is an exponential increase in circuit delay that increases the time per operation (clock cycle) over which the circuit leaks. As a result, a minimum energy per cycle point occurs at around 0.378 volts which is about 4.31 fJ per addition.Thus, depending on the minimum power or minimum energy requirements of the system, the choice of optimal V_{dd} may be different. This result is attractive for energy-efficient design of portable devices that require to dissipate low energy while operating with limited battery sources.

Chapter 4

Characterizing Processor for Energy and Performance Management

With the power consumption of recent desktop microprocessors having reached 130 watts, power has emerged at the forefront of challenges facing the microprocessor designer [13, 29]. The goal of modern microprocessors is to deliver as much performance as possible while keeping power consumption within reasonable limits. To carry out our experiment, there is a need to investigate the characteristics of the processor's voltage at certain frequencies. In this chapter, to compute the relative performance and power for chosen microprocessor we apply the adder data measured in previous sections to an Intel processor [32] assuming its technology to be the same as that used for adder simulation. For correlating adder data to processor we rely on specifications of the latter.

4.1 Intel Processor Specifications

In this section the Intel i5-2500K processor will be discussed. This processor is part of intel's new 32 nm Sandy bridge product line. The 32 nm process technology with second generation high-K + metal gate transistors enable designers to optimize for size, performance and power, simultaneously. The decreased oxide thickness and reduced gate length enable a greater than 22% transistor performance gain in terms of drive current. These transistors provide the highest drive currents and tightest gate pitch reported in the industry. Leakage current can also be optimized for a more than 5X reduction in leakage over 45 nm for nMOS transistors, and more than 10X reduction in leakage for pMOS transistors. These improvements combine to enable circuits to be designed that are both smaller and have improved energy efficiency.

Table 4.1: Intel i5 Sandy Bridge 2500K processor specifications.

Technology node	32 nm
Voltage range, V_{dd}	1.2-1.5 volts
Nominal base frequency, f_{TDP}	3.3 GHz
Overclock frequency, f_{max}	5.01 GHz
Thermal Design Power, TDP	95 watts

The Intel Core i5 2500K is an amazing core from Intel, sporting four powerful cores, each clocking at an impressive speed of 3.3 GHz, including a 6MB L3 cache and also having the option to reach up to 3.7 GHz with *Turbo Boost*. Intel's cutting-edge 32 nm micro-architecture with the second generation Hi-K metal gate process delivers higher performance at lower power, and a better overclocking capability. When Intel launched the Sandy Bridge architecture in 2011, it changed the nature of CPU overclocking (OC) by releasing specific OC-capable processors - all of which have a K (or X) suffix. CPU speed is defined by two factors - base clock (typically 100 MHz) and a multiplier, which is set to 33 on the 2500K. The 2500K typically runs at 3.3 GHz (100 MHz base clock, multiplied by 33). The K chips allow users to adjust the multiplier to whatever value they want. The specifications of the selected processor are defined in Table 4.1 as per Intel data sheet [32].

4.1.1 Important Definitions by Intel

Thermal design power (TDP) is the average maximum power in watts the processor dissipates when operating at base frequency with all cores active under a manufacturer defined, high complexity workload. TDP is not the maximum power the CPU may consume - there may be periods of time when the CPU dissipates more power than that allowed by its thermal design. TDP is usually 20%-30% lower than the CPU maximum power dissipation.

Peak power is the maximum power dissipated by the processor under the worst case conditions - at the maximum core voltage, maximum temperature and maximum signal loading conditions.

Table 4.2: Scale factors (Adder to Processor).

Scale factors	Calculated values
Voltage factor, σ	1
Area factor, β	7.3414×10^5
f_{nom} factor, δ	1.0588
f_{max} factor, γ	1.6075

Processor base frequency describes the rate at which the processor's transistors open and close. The processor base frequency is the operating point where TDP is defined. *Turbo boost* and *overclocking* are both essentially the same thing although they may work a little differently. Turbo boost is a feature of Intel processors created to dynamically overclock a CPU, meaning the more you use your CPU, the faster the CPU moves up to a certain point which is determined by the manufacturer. Overclocking is a similar concept except that it is not dynamic and is implemented manually, either through software or through BIOS on newer motherboards.

4.1.2 Scale Factors and Their Values for Processor

Characterization of a processor can be very complex and expensive. Therefore, we simulated a reasonable-size adder circuit as a technology benchmark and now we determine scale factors to scale that data to obtain processor power, energy per cycle (cycle efficiency) and clock frequency. Our scale factors, as shown in Table 4.2, are obtained using processor's specifications given at the rated voltage of 1.2 volts, assuming that voltage was not raised for overclocking.

Figure 4.1a shows scaled energy per cycle data for an intel i5 2500K processor. We know that cycle efficiency, $\eta = 1/EPC$, and Figure 4.1b shows cycle efficiency for the chosen processor. This will be used as a parameter for the given processor power management. Figure 4.1c shows scaled plots of TDP, dynamic and static power for the chosen intel processor.

Next, total power for adder and processor can be written as,

$$p = (e_{dyn} \times f_{max}) + p_{stat} \quad \text{(Adder}'\text{s Total Power)} \tag{4.1}$$

$$TDP = (E_{dyn} \times f_{TDP}) + P_{stat} \quad \text{(Processor}'\text{s Total Power)} \tag{4.2}$$

Since we selected our vectors in specific way as described in Section 3.5.1, the activity produced in both the circuits is assumed to be same and hence the activity scale factor in this case is 1. Now, if β is the scale factor representing the relative size of processor to adder circuit and σ is the voltage factor accounting for voltage at which adder is simulated being different from the processor supply voltage, then Equation 4.1 modifies Equation 4.2 as:

$$TDP = \beta.\sigma[(e_{dyn} \times f_{TDP}) + p_{stat}] \tag{4.3}$$

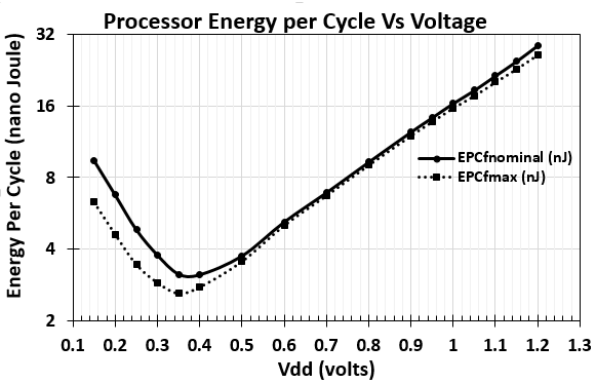
Solving for area factor β gives us,

$$\beta = \frac{TDP}{\sigma[(e_{dyn} \times f_{TDP}) + p_{stat}]} \tag{4.4}$$

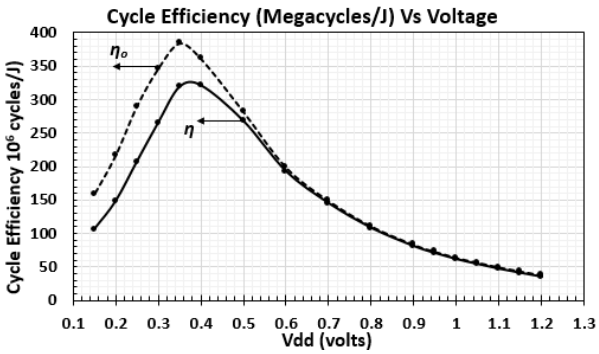
where, σ is defined as:

$$\sigma = \frac{V_{dd}\ (Processor)}{v_{dd}\ (Adder)} \tag{4.5}$$

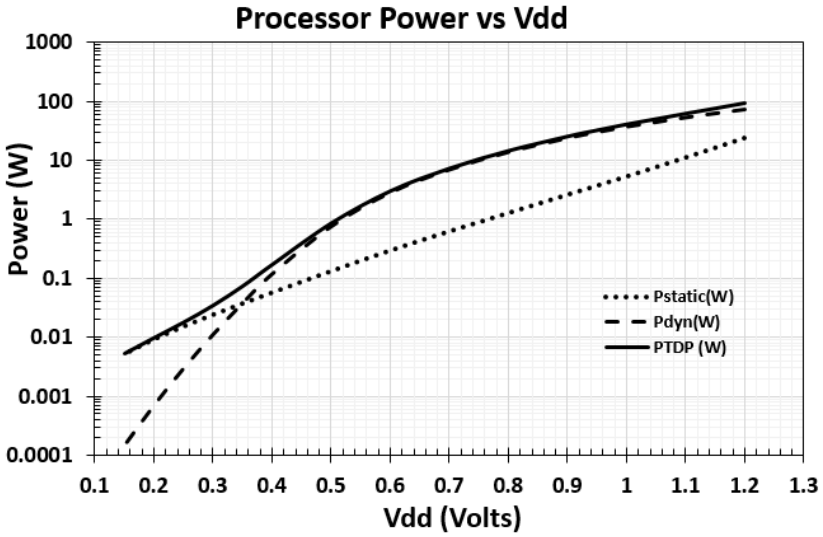
Equation 4.4 provides the area scale factor β based on processor thermal design power, $TDP = 95$ watts, adders dynamic energy e_{dyn}, adder's static power p_{stat} and the power constrained frequency $f_{TDP} = 3.3$ GHz at the rated voltage of 1.2 volts. Equation 4.5 provides voltage factor σ and is defined as ratio of rated supply voltage V_{dd} of the processor and supply voltage v_{dd} of the adder circuit. In this particular case, the adder circuit is simulated at same voltage at which the processor is rated which is 1.2 v, therefore the scale factor σ is 1. Table 4.3 shows all the values for the Intel i5-2500k processor obtained from



(a) Energy per cycle (EPC) for processor.



(b) Cycle efficiency $\eta = 1/EPC$ for processor.



(c) Thermal Design Power, dynamic and static power for processor.

Figure 4.1: Power consumption, energy per cycle and cycle efficiency plots for intel Sandy Bridge i5-2500k processor obtained by scaling adder data in 32 nm bulk CMOS technology.

scaling the adder data using scale factors from Table 4.2. Thermal design power (TDP) for chosen processor at any given voltage is defined below:

$$TDP = P_{dyn} + P_{static} = \beta \times (p_{dyn} + p_{stat}) \tag{4.6}$$

$$\text{or } TDP = \beta \times [(e_{dyn} \times f_{TDP}) + p_{stat}] \tag{4.7}$$

where TDP is thermal design power, P_{dyn} is dynamic power of the processor, P_{static} is static power for the processor, β is an area scale factor, whereas p_{dyn} is adjusted dynamic power of

Table 4.3: Scaled values for intel i5 2500K processor for 32 nm technology node in bulk CMOS PTM at different voltages (V_{dd}).

V_{dd}	Scaled Power			Scaled Frequency		Energy per cycle		Cycle efficiency	
	TDP	P_{dyn}	P_{static}	f_{nom}	f_{max}	E_{fnom}	E_{fmax}	10^6 cycles/J	
volts	W	W	W	GHz	GHz	nJ	nJ	η	η_0
1.2	95	71.02	23.98	3.3	5.01	28.79	26.31	34.74	38.01
1.15	77.16	60.87	16.29	3.12	4.74	24.7	22.92	40.49	43.63
1.1	63.03	51.86	11.17	2.94	4.46	21.46	20.16	46.6	49.6
1.05	51.01	43.33	7.69	2.74	4.16	18.62	17.66	53.7	56.61
1	41.48	36.15	5.33	2.53	3.84	16.4	15.68	60.96	63.76
0.95	32.93	29.21	3.72	2.31	3.5	14.28	13.73	70.04	72.85
0.9	25.81	23.19	2.62	2.08	3.15	12.43	11.99	80.48	83.37
0.8	14.75	13.47	1.29	1.59	2.41	9.29	9.01	107.66	110.96
0.7	7.42	6.79	0.63	1.07	1.63	6.91	6.71	144.71	149.02
0.6	3.07	2.77	0.3	0.59	0.9	5.2	5.02	192.43	199.02
0.5	0.8767	0.7434	0.1333	0.2347	0.3563	3.74	3.54	267.7	282.35
0.4	0.174	0.1166	0.0577	0.0559	0.0849	3.12	2.76	321.02	361.92
0.35	0.0752	0.0375	0.0377	0.024	0.0364	3.14	2.6	318.66	384.45
0.3	0.0354	0.011	0.0244	0.0094	0.0143	3.77	2.89	265.04	346.44
0.25	0.0183	0.002	0.0154	0.0038	0.0058	4.83	3.45	206.93	290.03
0.2	0.012	0.0007	0.0093	0.0015	0.0022	6.77	4.62	147.71	216.41
0.15	0.0054	0.0001	0.0053	0.0006	0.0009	9.46	6.32	105.74	158.31

the adder circuit for the frequency of processor at chosen voltage and is defined as product of e_{dyn} and f_{TDP}, i.e., dynamic energy of the adder circuit times frequency of the processor at that chosen voltage.

4.1.3 Nominal, Structure Constrained and Power Constrained Frequencies

Three different frequencies, f_{nom} (nominal frequency/base frequency), f_{max} (structure constrained or maximum frequency) and f_{TDP} (power constrained frequency) are also measured by scaling adder data which also results in energy per cycle and cycle efficiency for the defined frequencies.

Processor *base* or *nominal* clock frequency describes the rate at which the processor's transistors open and close. The processor base frequency is the operating point where TDP is defined. Frequency is measured in gigahertz (GHz), or billion cycles per second. We

calculated nominal frequency, f_{nom} as:

$$f_{nom} = \delta \times f_{max}(Adder) \tag{4.8}$$

where δ is a scale factor for f_{nom} and is given by,

$$\delta = \frac{f_{nomVdd}(Processor)}{f_{maxVdd}(Adder)} \tag{4.9}$$

In the equation defined above, f_{nomVdd} is nominal frequency of processor and f_{maxVdd} is maximum frequency of an adder circuit at a rated voltage $V_{dd} = 1.2$ volts.

In a structure constrained system, the frequency f_{max} is limited by the critical path delay of the circuit as follows:

$$f_{max} = \gamma \times f_{max}(Adder) \tag{4.10}$$

where γ is a scale factor for f_{max} and is given by,

$$\gamma = \frac{f_{maxVdd}\ (Processor)}{f_{maxVdd}\ (Adder)} \tag{4.11}$$

Similarly, in the equation defined above f_{maxVdd} is maximum frequency of processor and f_{maxVdd} is maximum frequency of an adder circuit at a rated voltage $V_{dd} = 1.2$ volts.

In a power constrained system [61–63], the frequency f_{TDP} is limited by the maximum allowable power of the circuit. In general it can be represented as,

$$f_{TDP} = \frac{TDP - \sigma\beta p_{stat}}{\sigma\beta e_{dyn}} \tag{4.12}$$

where TDP is thermal design power of processor at given power constrained frequency f_{TDP} and rated voltage, σ is voltage factor, β is the area scale factor (adder-benchmark circuit to

processor), p_{stat} is the static power of the adder circuit, and e_{dyn} is the dynamic energy of the adder circuit.

The energy per cycle for the processor for the nominal frequency and overclock/maximum frequency for a any given V_{dd} is defined by:

$$EPC_{nom} = \frac{TDP}{f_{nom}} \tag{4.13}$$

$$EPC_{F_0} = \frac{P_{dyn}}{f_{nom}} + \frac{P_{static}}{F_0} \tag{4.14}$$

Equation 4.14 defines the energy per cycle EPC_{F_0} for any given frequency F_0 of processor where F_0 value ranges from $f_{nom} \leq F_0 \leq f_{max}$. In this case, $F_0 = f_{max} = 5.01$ GHz. Therefore, we call EPC_{F_0} as EPC_{fmax}, i.e., energy per cycle for maximum frequency allowed to run the system at a given voltage. As we know, cycle efficiency $\eta = 1/EPC$, therefore, from Equations 4.13 and 4.14 we can define cycle efficiencies for the given processor as:

$$\eta = \frac{1}{EPC_{nom}} \tag{4.15}$$

$$\eta_0 = \frac{1}{EPC_{F_0}} \tag{4.16}$$

where η is defined as nominal cycle efficiency and η_0 as cycle efficiency for any given frequency F_0 ranging between $f_{nom} \leq F_0 \leq f_{max}$. Here, $EPC_{F_0} = EPC_{f_{max}}$, therefore, we call η_0 as peak cycle efficiency.

All the parameters defined above are used in the next section to show our proposed power management method. With these parameters we have shown how one can optimize time and energy of a processor based on performance requirements by the user.

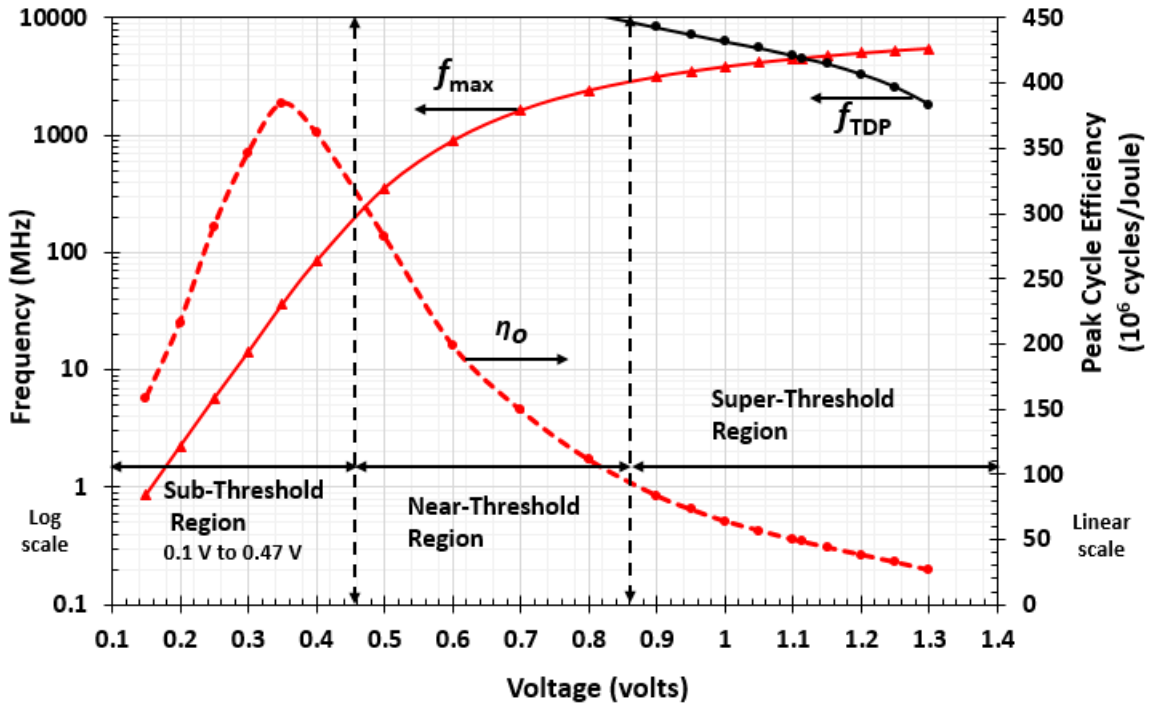


Figure 4.2: Plot showing proposed "Power Management Method" for three different regions.

4.2 Power Management Methodology

Power management provides a system solution to boost the processor frequency to values higher than the nominal value whenever required, as per performance criteria. For workloads that are not operating at the cooling/power supply limits this can often result in real performance increase. The focus of this experiment is to evaluate the benefits of proposed method and establish the relationship between the workload and related system characteristics, which determine the benefits. Some of the newer works that look at power management and its impact on performance in a non-embedded-systems context can be found through these references [16, 26, 30, 47].

In Figure 4.2, we see three different regions of operation for a processor, shown as: super-threshold region, near threshold region and sub-threshold region. Energy and time optimization for processor that runs on higher performance is explained in the super-threshold region (0.85 volts to 1.3 volts) in Figure 4.3 whereas processors or low power devices that

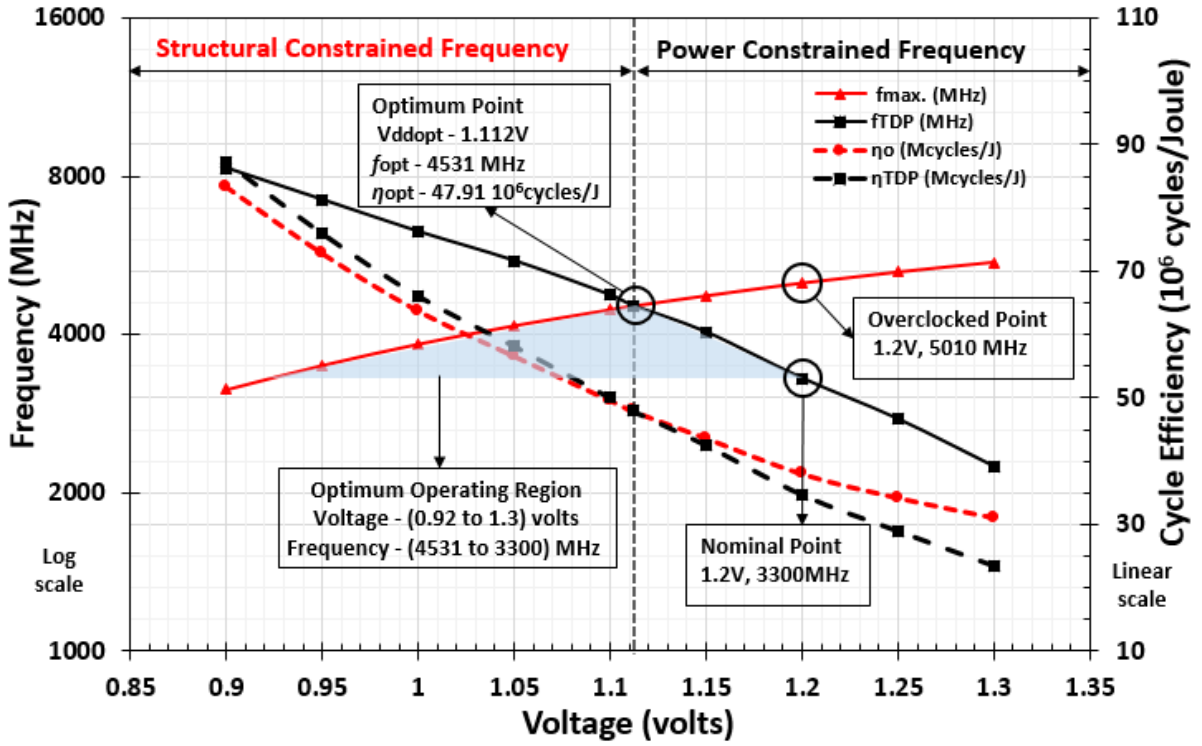


Figure 4.3: Processor's calculated scaled curves of f_{max} and f_{TDP} at various voltages. The cross point exact value (V_{ddopt}, f_{opt}) is obtained by curve fitting the data with polynomial equations of degree 3.

do not demand high clock-speed performance may operate in near-threshold (0.45 volts to 0.85 volts) or sub-threshold (0.15 volts to 0.45 volts) region as shown in Figure 4.4.

This method discusses all the aspects necessary for time and energy optimization, such as: (a) when is it possible to run a processor at a higher clock speed without exceeding the power limits explained through Figure 4.3 and Table 4.4, (b) what will be the most energy-efficient point for the processors that requires low power and rules out high performance as a main criteria explained through Figure 4.4, and (c) the value of doing so explained in Section 4.2.2. Using the processor performance counters to measure execution events of the applications, we identify the characteristics that determine the extent of performance benefits in terms of time and energy from higher as well as lower clock frequencies and those characteristics that cause the application to become power-limited.

Table 4.4: Structure constrained and power constrained clock frequencies for processor with their corresponding cycle efficiency.

Voltage V_{dd} (volts)	Clock frequency (MHz)		Cycle efficiency (10^6 cycles/J)	
	Structure constrained f_{max}	Power constrained f_{TDP}	Peak η_0 at f_{max}	η_{TDP} at f_{TDP}
1.30	5486	2243	31.09	23.57
1.25	5257	2761	34.22	29.04
1.20	5010	3300	38.01	34.74
1.15	4740	4040	43.63	42.52
1.112	**4531**	**4531**	**47.91**	**47.91**
1.10	4460	4750	49.6	49.98
1.05	4160	5520	56.61	58.11
1.00	3840	6270	63.76	66.02
0.95	3500	7210	72.85	75.87
0.90	3150	8280	83.37	87.11

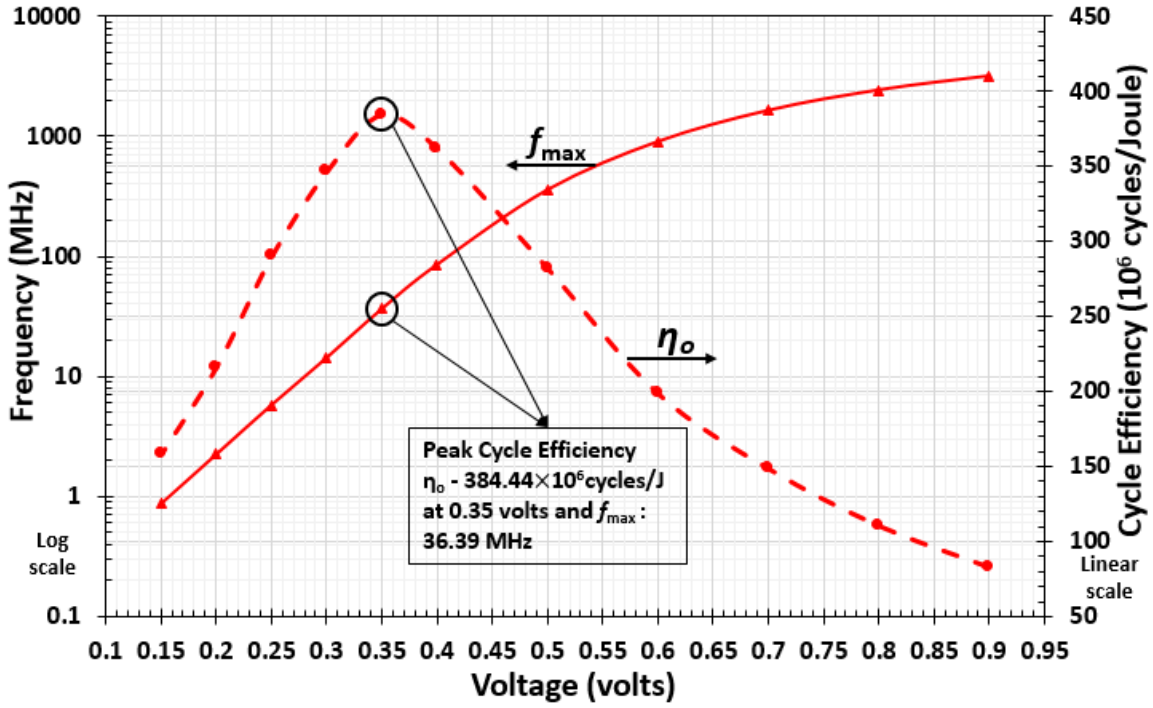


Figure 4.4: Minimum energy operation point.

4.2.1 Optimum Voltage, Frequency and Cycle Efficiency

One method to reduce the power dissipation in CMOS circuits is to reduce the supply voltage V_{dd}. However, reducing supply voltage has an inverse relation with gate delay, i.e. the gate delay increases as the voltage is reduced. Sakurai and Newton [53] proposed a delay model that characterizes the delay based on the velocity saturation index α (*Not to be confused with the activity factor α used in the previous discussion for estimating the dynamic power*). An approximation of this model was stated in [54] called the alpha-power-law delay model and is re-written below

$$t_d \propto \frac{V_{dd}}{(V_{dd} - V_{th})^{\alpha}} \tag{4.17}$$

where α is the velocity saturation index, V_{th} is the threshold voltage of the device and V_{dd} is the supply voltage. In this section we aim to find the best voltage and frequency at which a power-constrained system can run with maximum frequency without exceeding the peak power or violating the critical path delay constraint of the circuit. As mentioned above, the frequency can be increased while limiting the power by reducing the supply voltage. However, there exists a point where the voltage will not be enough to charge the output load capacitance within the right amount of time. Thus the value at the output will be wrong. At this point the circuit is considered structure constrained and the frequency is now dependent on the critical path delay of the circuit and is defined by Equation 4.10 in the previous section.

To maximize the performance we find the highest frequency, f_{opt} that would exceed neither the power constraint of Equation (4.12) nor the critical path constraint of Equation (4.10) satisfying the expression given below:

$$f_{opt} = f_{TDP} = f_{max} \tag{4.18}$$

At any given voltage the optimum frequency is obtained as,

$$f_{opt} = \min\{max f_{TDP}, max f_{max}\} \tag{4.19}$$

Using Equations 4.12 and 4.10 we measure the two frequencies for different supply voltages. From Equation 4.12, we observe that as voltage is reduced f_{TDP} increases, but at the same time from Equation 4.10, f_{max} reduces. This is also evident from Table 4.4. Similarly, we measure cycle efficiencies for the above two frequencies, f_{TDP} and f_{max}, using Equation 4.16 and call it as η_{TDP} (TDP cycle efficiency) and η_0 (Peak cycle efficiency).

Thus, if we plot these frequencies from Table 4.4, the two functions will intersect at a point which we define as an optimum point (V_{ddopt}, f_{opt}). This is well explained through Figure 4.3. Using a curve fitting tool in Microsoft Excel we get two polynomial equations of degree 3 for these two functions, producing the expressions,

$$f_{TDP} = -9730.6V_{dd}^3 + 45254V_{dd}^2 - 78922V_{dd} + 49719 \tag{4.20}$$

and

$$f_{max} = -168.35V_{dd}^3 - 2991.2V_{dd}^2 + 13042V_{dd} - 6043 \tag{4.21}$$

Similarly, if we plot for the two cycle efficiencies from Table 4.4, the two functions will intersect at a point defined as η_{opt} and we fit two polynomial equations of degree 3 for these two functions as well, producing the expressions,

$$\eta_0 = -66.649V_{dd}^3 + 412.23V_{dd}^2 - 792.82V_{dd} + 511.39 \tag{4.22}$$

and

$$\eta_{TDP} = -100.33V_{dd}^3 - 468.29V_{dd}^2 + 820.67V_{dd} - 519.25 \tag{4.23}$$

Solving Equations 4.20 and 4.21 by a numerical solver in MATLAB, we obtain two complex roots and one real root. Discarding the complex roots, the real root gives $V_{dd} = 1.112$ volts. This is the optimum voltage V_{ddopt} at which the processor will run fastest without exceeding the TDP. We can calculate f_{opt} from Equation 4.20 or 4.21 by substituting $V_{dd} = V_{ddopt}$, which gives $f_{opt} = 4531$ MHz. Similarly, we can calculate η_{opt} from Equation 4.22 or 4.23 by substituting $V_{dd} = V_{ddopt}$, which gives $\eta_{opt} = 47.91{\times}10^6$ cycles/J.

In Figure 4.3 we also observe that an optimum operation region is defined where a processor can be operated at any voltage ranging from 0.92 volts to 1.2 volts (rated voltage) and any frequency ranging from 3300 MHz (rated frequency) to 4531 MHz without exceeding 95 watts which is the processor's rated power (TDP). In this region, a processor can be run at the rated frequency but with higher cycle efficiency $\eta_0 = 79.01{\times}10^6$ cycles/J by reducing the voltage from 1.2 volts to 0.92 volts, whereas if high performace is required within this operating range, a processor can be made to run at highest frequency $f_{opt} = 4531$ MHz at $V_{ddopt} = 1.112$ volts. In the next section we will illustrate the proposed method through an application and will describe five senarios for time and energy optimization.

4.2.2 Power Management Application

Consider a program that executes in two billion clock cycles ($c = 2 \times 10^9$). Five scenarios are presented in Table 4.5 and are explained in detail below:

1. **Nominal Operating Point**: Our first scenario is a power-constrained operation because we operate the processor at the nominal operating voltage $V_{dd} = 1.2$ volts and clock frequency $f = 3300$ MHz, which are the rated voltage and frequency. We see from Figure 4.1b and Table 4.4 that cycle efficiency $\eta TDP = 34.74{\times}10^6$ cycles/J, power consumption is 95 watts and is available from processor specification shown in Table 4.1 also can be calculated by using Equation 2.17, execution time is 0.61 seconds, and is measured using Equation 2.13, and total energy consumed by program is 57.57 J, calculated using Equation 2.14.

Table 4.5: Managing the processor operation for time and energy used by a program requiring two billion clock cycles ($c = 2 \times 10^9$).

Operating Modes	V_{dd} volts	Clock frequency f (MHz)	Cycle efficiency η 10^6cycles/J	Average power $\frac{f}{\eta}$ watts	Execution time $\frac{c}{f}$ seconds	Total energy $\frac{c}{\eta}$ (J)
Nominal Operating Point	1.2	3300	34.74	95	0.61	57.57
Overclocked Operating Point 20% Overclk	1.2 20%	3300 (80%) 5010 (20%)	34.74 38.01	95 132	0.485 +0.0798 = 0.57	46.06 +10.52 = 56.58
Optimum Operating Point	1.112	4531	47.91	95	0.44 (−28%)	41.75 (−28%)
Dynamic Voltage Scaled Point	0.92	3300	79.01	41.77 (−56%)	0.61 (0%)	25.31 (−56%)
Energy Efficient Operating Point	0.35	36.39	384.45	0.0946	54.96	5.20

2. **Overclocked Operating Point**: The second scenario also uses 1.2 volts and 80% of the program is executed at 3300 MHz clock, but for higher performance the remaining 20% of the program is executed at an overclock frequency of 5010 MHz, which is the highest frequency the critical path will allow at this voltage. The power is allowed to exceed TDP for 20% of time. Note that power increase from 95 watts to 132 watts is not proportional to the frequency ratio, because only dynamic power increases, leaving static power unchanged. Cycle efficiency η_{TDP} at 3330 MHz is 34.74×10^6 cycles/J and η_0 at 5010 MHz is 38.01×10^6 cycles/J. The execution time is reduced to 0.57 seconds and total energy consumption is also slightly lower at 56.58 J. We do not observe a significant reduction in execution time or total energy with this kind of operation despite higher power consumption.

3. **Optimum Operating Point**: Using our proposed method, in the third scenario, we find optimum voltage, frequency and cycle efficiency ($V_{ddopt}, f_{opt}, \eta_{opt}$) from Figure 4.3

and allow the program to run at $V_{dd} = 1.112$ volts and clock frequency $f = 4531$ MHz. η_{opt} is calculated from Equation 4.22 or 4.23 by substituting $V_{dd} = V_{ddopt}$, which gives $\eta_{opt} = 47.91 \times 10^6$ cycles/J. The power consumption is no more than 95 watts (TDP) but the program execution time is reduced to 0.44 seconds and total energy consumed is 41.75 J. Thus, we observe improved performance with 28% reduction in both energy consumption and execution time.

4. **Dynamic Voltage Scaled Point**: There have been a number of efforts over the years examining the implementation and effectiveness of dynamic voltage and frequency scaling for saving power in embedded systems [52]. Performance-oriented explorations include attempts to quantify and/or reduce the performance loss encountered in an energy-saving adoption of DVS. In contrast, our fourth scenario targets performance increase from DVS in a power-constrained environment. Here the program can execute at the rated frequency, which is 3300 MHz, by decreasing the voltage to 0.92 volts. Again, η_0 is calculated from Equation 4.22 by substituting $V_{dd} = 0.92$ volts, which gives $\eta_0 = 79.01 \times 10^6$ cycles/J. The power consumption is 41.77 watts but the program execution time is same as rated voltage of 0.61 seconds whereas total energy consumed is reduced to 25.31 J. Here, we can see the performance in terms of energy and not the time, therefore, when low energy is the criteria and not the speed, this type of operation is successful.

5. **Energy Efficient Point**: The fifth scenario is for very low power devices. This type of operation is highly energy efficient and is used only for circuits with low switching activity or that do not requires high clock speed to operate. Figure 4.4 shows minimum energy operation at $V_{dd} = 0.35$ volts and frequency 36.39 MHz. When a program executes at this low voltage, it gives cycle efficiency $\eta_0 = 384.45 \times 10^6$ cycles/J, which is the peak cycle efficiency for this processor. The power consumption for this type of

execution is 0.0946 watts, or 94.6 μwatts, but the program execution time is increased to 54.96 seconds and the energy consumption is lowest, i.e., 5.20 J.

In the next chapter we will apply the introduced method to other predictive technology models in different technology nodes and discuss the results briefly. PTM models discussed in next section are 32 nm High-K, 45 nm Bulk and High-K and 22 nm Bulk and High-K.

Chapter 5

Simulation Results for Other PTM Technologies

In this chapter we present simulation results for other predictive technology models (PTM) described in an earlier section on technology nodes. The proposed power management method is used to determine optimum voltage and frequency for Intel processors in 45 nm, 32 nm and 22 nm technologies.

Table 5.1 shows power: average, dynamic, static and peak; timing: critical path delay and its inverse f_{max}; and energy per cycle : dynamic, static and total, for the 16-bit ripple carry adder (RCA) circuit estimated by H-Spice tool [4] using 45 nm Bulk CMOS Predictive Technology Model when subjected to varying V_{dd} ranging from 1.3 volts (nominal voltage) to 0.15 volt (sub-threshold region).

These simulation results of adder circuit are for 45 nm technology therefore we chose an intel Core 2 Duo T9500 processor [9] in same technology node with specifications listed in table 5.2. The Intel Core 2 Duo T9500 was an upper middle class dual core CPU for laptops at the time of its introduction. It was intended for use in the Santa Rosa platform due to the 800 MHz FSB (front-side bus). The T9550 has a slightly higher clock than FSB1066. Due to the relatively high clock speed and 6MB Level 2 cache, the T9500 offers enough performance demanded by games (in 2009) and other applications. The T9500 uses a Penryn (Montevina Update) core that features 2 integer units, 1 floating point unit, 1 load unit, and 1 store unit in a 14-stage pipeline. Due to the Wide Dynamic Execution Technology, the core is able to simultaneously execute up to four instructions. The integrated *enhanced speedstep* is able to down clock the core dynamically to save power (in idle mode).

5.1 45 nm Bulk CMOS PTM

Table 5.1: H-spice [4] simulation of 16 bit ripple carry adder for 45 nm technology node in bulk CMOS PTM [7] at different voltages (V_{dd}).

V_{dd}	Power from simulation				Timing from simulation		Energy per cycle		
volts	p_{avg} μW	p_{dyn} μW	p_{static} μW	p_{peak} μW	Critical path delay, ps	f_{max} GHz	e_{dyn} fJ	e_{static} fJ	e_{total} fJ
1.3	155.19	139.82	15.37	531.66	383	2.61	53.55	5.89	59.44
1.25	132.91	121.28	11.63	462.8	394	2.54	47.79	4.58	52.37
1.2	115.09	106.3	8.79	438.34	416	2.4	44.22	3.66	47.88
1.15	97.86	91.16	6.7	364.69	435	2.3	39.65	2.91	42.57
1.1	82.99	77.87	5.12	322.34	461	2.17	35.9	2.36	38.26
1.05	69.91	65.98	3.92	250.1	494.32	2.02	32.62	1.94	34.56
1	58.2	55.18	3.03	218.01	536.86	1.86	29.62	1.63	31.25
0.95	47.21	44.92	2.29	171.01	578.87	1.73	26	1.32	27.33
0.9	37.96	36.21	1.75	138.88	637.97	1.57	23.1	1.12	24.22
0.85	29.94	28.6	1.34	118.95	713.56	1.4	20.41	0.957	21.37
0.8	22.96	21.94	1.02	92.14	813.39	1.23	17.85	0.83	18.68
0.7	11.89	11.32	0.576	48.15	1165.8	0.858	13.19	0.671	13.87
0.6	5.17	4.86	0.307	20.28	1945.4	0.514	9.46	0.597	10.06
0.5	1.64	1.49	0.152	6.41	4360.9	0.229	6.49	0.664	7.15
0.4	0.331	0.259	0.072	1.15	16191	0.062	4.19	1.17	5.36
0.35	0.136	0.086	0.05	0.405	36932	0.027	3.19	1.85	5.04
0.3	0.06	0.026	0.034	0.139	93800	0.011	2.4	3.23	5.64
0.25	0.03	0.007	0.023	0.049	234060	0.004	1.62	5.44	7.06
0.2	0.017	0.002	0.015	0.023	614330	0.002	1.01	9.26	10.26
0.15	0.009	0	0.009	0.011	1624700	0.001	0.574	14.83	15.41

In Table 5.1, as we reduce the supply voltage, power reduces quadratically. Therefore, the total average energy, calculated as power delay product (PDP), also reduces when the supply voltage reduces. With down scaling of V_{dd} below the threshold voltage, there is an exponential increase in circuit delay that increases the time per operation (clock period) over which the circuit leaks. As a result, the minimum energy point occurs at around 0.35 volts that is about 5.04 fJ per addition for this technology.

In the Table 5.4, scaled values are shown for Intel core 2 Duo T9500, which are obtained using scale factors given in Table 5.3. The minimum energy point for the processor occurs at sub-threshold voltage 0.35 volts and energy per cycle for nominal frequency is measured as, $EPC_{fnom} = 1.2858$ nJ. For maximum frequency f_{max}, $EPC_{fmax} = 1.2058$ nJ.

Table 5.2: Intel Core2 Duo T9500 processor specifications [9].

Technology node	45 nm
Voltage range	1-1.25 volts
Nominal base frequency, f_{TDP}	2.6 GHz
Overclock frequency, f_{max}	3.14 GHz
Thermal Design Power, TDP	35 watts

Table 5.3: Scale factors (Adder to Processor).

Scale factors	Calculated values
Voltage factor, σ	1
Area factor, β	2.5760×10^5
f_{nom} factor, δ	1.0244
f_{max} factor, γ	1.23716

Table 5.4: Scaled values for intel Core2 Duo T9500 processor [9] for 45 nm technology node in bulk CMOS PTM [7] at different voltages (V_{dd}).

V_{dd}	Scaled Power			Scaled Frequency		Energy per cycle		Cycle efficiency 10^6 cycles/J	
	TDP	P_{dyn}	P_{static}	f_{nom}	f_{max}	E_{fnom}	E_{fmax}		
volts	W	W	W	GHz	GHz	nJ	nJ	η	η_0
1.25	35	32.01	2.99	2.6	3.14	13.46	13.26	74.29	75.4
1.2	30.32	28.05	2.26	2.46	2.97	12.31	12.15	81.23	82.29
1.15	25.78	24.06	1.7248	2.35	2.84	10.95	10.82	91.35	92.41
1.1	21.87	20.55	1.3186	2.22	2.68	9.84	9.74	101.62	102.68
1.05	18.42	17.41	1.0109	2.07	2.5	8.89	8.81	112.49	113.56
1	15.34	14.56	0.7799	1.9081	2.3	8.04	7.97	124.39	125.49
0.95	12.44	11.85	0.5893	1.7697	2.14	7.03	6.97	142.22	143.38
0.9	10.01	9.56	0.4517	1.6057	1.9392	6.23	6.18	160.46	161.71
0.85	7.89	7.55	0.3454	1.4356	1.7338	5.5	5.46	181.88	183.26
0.8	6.05	5.79	0.263	1.2594	1.521	4.81	4.77	208.07	209.63
0.7	3.14	2.99	0.1483	0.8787	1.0612	3.57	3.54	280.29	282.59
0.6	1.3625	1.2835	0.079	0.5266	0.6359	2.59	2.56	386.47	390.37
0.5	0.4317	0.3924	0.0392	0.2349	0.2837	1.8376	1.8089	544.19	552.83
0.4	0.0869	0.0683	0.0187	0.0633	0.0764	1.3738	1.323	727.93	755.85
0.35	0.0357	0.0228	0.0129	0.0277	0.0335	1.2858	1.2058	777.7	829.29
0.3	0.0156	0.0068	0.0089	0.0109	0.0132	1.4325	1.2926	698.07	773.61
0.25	0.0078	0.0018	0.006	0.0044	0.0053	1.7858	1.5504	559.98	645
0.2	0.0043	0.0004	0.0039	0.0017	0.002	2.5873	2.1869	386.51	457.26
0.15	0.0024	0.0001	0.0024	0.0006	0.0008	3.8776	3.2362	257.89	309

5.2 45 nm High-K PTM

Table 5.5: H-spice [4] simulation of 16 bit ripple carry adder for 45 nm technology node in high-K CMOS PTM [7] at different voltages (V_{dd}).

V_{dd}	Power from simulation				Timing from simulation		Energy per cycle		
volts	p_{avg} μW	p_{dyn} μW	p_{static} μW	p_{peak} μW	Critical path delay, ps	f_{max} GHz	e_{dyn} fJ	e_{static} fJ	e_{total} fJ
1.3	371.69	225.47	146.22	1033.37	196.66	5.08	44.34	28.76	73.1
1.25	306.1	202.55	103.55	903.61	198.71	5.03	40.25	20.58	60.83
1.2	256.9	179.48	77.42	788.78	200.95	4.98	36.07	15.56	51.62
1.15	216.97	157.22	59.75	687.82	203.79	4.91	32.04	12.18	44.22
1.1	186.59	139.58	47.01	600.77	206.72	4.84	28.85	9.72	38.57
1.05	159.64	122.4	37.24	527.64	209.47	4.77	25.64	7.8	33.44
1	135.03	105.28	29.75	449.71	214.84	4.65	22.62	6.39	29.01
0.95	113.8	90.03	23.77	384.62	221.03	4.52	19.9	5.25	25.15
0.9	94.49	75.62	18.87	323.82	228.47	4.38	17.28	4.31	21.59
0.85	77.77	62.82	14.95	267.42	238.24	4.2	14.97	3.56	18.53
0.8	63.07	51.29	11.78	219.48	249.81	4	12.81	2.94	15.75
0.7	39.96	32.81	7.15	140.13	283.56	3.53	9.3	2.03	11.33
0.6	23.05	18.88	4.17	79.8	340.91	2.93	6.44	1.42	7.86
0.5	11.82	9.49	2.33	39.6	458.61	2.18	4.35	1.068	5.42
0.4	4.6	3.38	1.229	15.37	768.19	1.302	2.59	0.944	3.54
0.35	2.46	1.584	0.873	7.52	1173	0.853	1.858	1.025	2.88
0.3	1.247	0.633	0.614	3.2	2049.2	0.488	1.298	1.258	2.56
0.25	0.636	0.208	0.428	1.309	4133.4	0.242	0.861	1.768	2.63
0.2	0.347	0.055	0.292	0.547	9888.9	0.101	0.544	2.89	3.43
0.15	0.201	0.01	0.191	0.258	24973	0.04	0.259	4.76	5.02

To illustrate the optimization framework we are assuming the same processor for both the technology models bulk and High-K in 45 nm, because limited information is available from the Intel data sheet [9]. For 45 nm High-K, the minimum energy point occurs at around 0.3 volts, which is about 2.56 fJ per addition shown in Table 5.5. As it is a High-K model, therefore, the scale factors for intel core 2 Duo T9500 processors changes to those given in Table 5.7 and the scaled values for the processor are shown in Table 4.3. The minimum energy point for the processor occurs at sub-threshold voltage 0.30 volts and energy per cycle is measured as, $EPC_{fnom} = 0.6275$ nJ for nominal frequency and $EPC_{fmax} = 0.5571$ nJ for the maximum frequency.

Table 5.6: Intel Core2 Duo T9500 processor specifications [9].

Technology node	45 nm
Voltage range	1-1.25 volts
Nominal base frequency, f_{TDP}	2.6 GHz
Overclock frequency, f_{max}	3.14 GHz
Thermal Design Power, TDP	35 watts

Table 5.7: Scale factors (Adder to Processor).

Scale factors	Calculated values
Voltage factor, σ	1
Area factor, β	1.681×10^5
f_{nom} factor, δ	0.517
f_{max} factor, γ	0.624

Table 5.8: Scaled values for intel Core2 Duo T9500 processor [9] for 45 nm technology node in high-K CMOS PTM [7] at different voltages (V_{dd}).

V_{dd}	Scaled Power			Scaled Frequency		Energy per cycle		Cycle efficiency 10^6 cycles/J	
	TDP	P_{dyn}	P_{static}	f_{nom}	f_{max}	E_{fnom}	E_{fmax}		
volts	W	W	W	GHz	GHz	nJ	nJ	η	η_0
1.25	35	17.59	17.41	2.6	3.14	13.46	12.31	74.29	81.23
1.2	28.6	15.59	13.01	2.57	3.1	11.13	10.25	89.89	97.52
1.15	23.7	13.66	10.04	2.54	3.06	9.35	8.67	106.97	115.38
1.1	20.03	12.12	7.9	2.5	3.02	8.01	7.47	124.8	133.89
1.05	16.89	10.63	6.26	2.47	2.98	6.85	6.41	146.02	155.96
1	14.15	9.14	5	2.4	2.9	5.88	5.52	170	181.01
0.95	11.82	7.82	4	2.34	2.82	5.05	4.76	197.84	210.05
0.9	9.74	6.57	3.17	2.26	2.73	4.31	4.07	232.16	245.94
0.85	7.97	5.46	2.51	2.17	2.62	3.67	3.48	272.11	287.72
0.8	6.43	4.45	1.9798	2.07	2.5	3.11	2.95	321.42	339.37
0.7	4.05	2.85	1.2027	1.822	2.2	2.22	2.11	449.62	473.8
0.6	2.34	1.6402	0.7004	1.5155	1.8302	1.5444	1.465	647.48	682.61
0.5	1.2157	0.8243	0.3915	1.1265	1.3605	1.0792	1.0194	926.64	980.96
0.4	0.4997	0.2932	0.2066	0.6725	0.8122	0.7431	0.6902	1345.8	1448.78
0.35	0.2844	0.1376	0.1468	0.4404	0.5319	0.6457	0.5884	1548.61	1699.5
0.3	0.1582	0.055	0.1032	0.2521	0.3045	0.6275	0.5571	1593.6	1794.93
0.25	0.09	0.0181	0.0719	0.125	0.151	0.7201	0.6212	1388.71	1609.9
0.2	0.0539	0.0048	0.0491	0.0522	0.0631	1.032	0.8703	968.95	1149.05
0.15	0.033	0.0009	0.0321	0.0207	0.025	1.5935	1.3269	627.57	753.63

5.3 32 nm High-K PTM

Table 5.9: H-spice [4] simulation of 16 bit ripple carry adder for 32 nm technology node in high-K CMOS PTM [7] at different voltages (V_{dd}).

V_{dd}	Power from simulation				Timing from simulation		Energy per cycle		
volts	p_{avg} μW	p_{dyn} μW	p_{static} μW	p_{peak} μW	Critical path delay, ps	f_{max} GHz	e_{dyn} fJ	e_{static} fJ	e_{total} fJ
1.20	270.29	161.55	108.74	703.12	156.73	6.38	25.32	17.04	42.36
1.15	219.54	141.9	77.64	604.45	159.05	6.29	22.57	12.35	34.92
1.10	181.26	122.84	58.42	518.78	162	6.17	19.9	9.46	29.36
1.05	151.9	106.64	45.26	454.99	165.06	6.06	17.6	7.47	25.07
1.00	127.44	91.86	35.58	389.63	168.5	5.93	15.48	5.99	21.47
0.95	106.14	77.95	28.19	328.94	173.18	5.77	13.5	4.88	18.38
0.90	87.11	64.83	22.29	264.8	180.64	5.54	11.71	4.03	15.74
0.80	57.38	43.65	13.73	179.8	197.63	5.06	8.63	2.71	11.34
0.70	34.93	26.77	8.16	119.51	227.32	4.4	6.09	1.85	7.94
0.60	20.05	15.44	4.61	68	276.24	3.62	4.26	1.27	5.54
0.50	9.79	7.36	2.43	32.35	376.01	2.66	2.77	0.915	3.68
0.40	3.79	2.61	1.17	11.45	655.61	1.53	1.71	0.77	2.48
0.35	1.97	1.18	0.79	5.59	1050.1	0.95	1.24	0.83	2.07
0.30	0.98	0.46	0.52	2.4	1895.6	0.53	0.86	0.99	1.85
0.25	0.48	0.14	0.34	0.95	4038.1	0.25	0.56	1.38	1.94
0.20	0.255	0.035	0.22	0.393	9930.8	0.101	0.348	2.184	2.53
0.15	0.142	0.006	0.136	0.181	26306	0.038	0.153	3.59	3.74

The Table 5.9 shows simulation results for adder circuit in 32 nm High-K model. The minimum energy point for adder circuit occurs at around 0.3 volts, which is about 1.85 fJ. The scale factors for Intel i5-2500K processors are defined in Table 5.11 and the scaled values for the processor are shown in Table 5.12. The minimum energy point for the processor occurs at sub-threshold voltage 0.30 volts and energy per cycle is measured as, $EPC_{fnom} = 1.371$ nJ for nominal frequency and $EPC_{fmax} = 1.048$ nJ for maximum frequency.

Table 5.10: Intel i5 Sandy Bridge 2500K processor specifications [32].

Technology node	32 nm
Voltage range	1.2-1.5 volts
Nominal base frequency, f_{TDP}	3.3 GHz
Overclock frequency, f_{max}	5.01 GHz
Thermal Design Power,TDP	95 watts

Table 5.11: Scale factors (Adder to Processor).

Scale factors	Calculated values
Voltage factor, σ	1
Area factor, β	4.940×10^5
f_{nom} factor, δ	0.5172
f_{max} factor, γ	0.7852

Table 5.12: Scaled values for intel i5-2500K processor [32] for 32 nm technology node in high-K CMOS PTM [7] at different voltages (V_{dd}).

V_{dd}	Scaled Power			Scaled Frequency		Energy per cycle		Cycle efficiency 10^6 cycles/J	
	TDP	P_{dyn}	P_{static}	f_{nom}	f_{max}	E_{fnom}	E_{fmax}		
volts	W	W	W	GHz	GHz	nJ	nJ	η	η_0
1.2	95	41.28	53.72	3.3	5.01	28.79	23.23	34.74	43.04
1.15	74.61	36.26	38.36	3.25	4.94	22.95	18.92	43.58	52.86
1.1	60.25	31.39	28.86	3.19	4.85	18.87	15.79	52.99	63.35
1.05	49.61	27.25	22.36	3.13	4.76	15.83	13.4	63.16	74.65
1	41.05	23.47	17.58	3.07	4.66	13.37	11.42	74.78	87.57
0.95	33.84	19.92	13.93	2.99	4.53	11.33	9.74	88.24	102.66
0.9	27.57	16.56	11.01	2.86	4.35	9.63	8.32	103.84	120.22
0.8	17.94	11.15	6.78	2.62	3.97	6.85	5.97	145.9	167.52
0.7	10.87	6.84	4.03	2.28	3.45	4.78	4.17	209.27	239.59
0.6	6.22	3.94	2.28	1.87	2.84	3.32	2.91	300.92	343.89
0.5	3.08	1.88	1.202	1.376	2.09	2.24	1.942	446.31	514.87
0.4	1.248	0.668	0.58	0.789	1.198	1.582	1.331	631.97	751.18
0.35	0.691	0.301	0.39	0.493	0.748	1.403	1.133	712.61	882.55
0.3	0.374	0.116	0.258	0.273	0.414	1.371	1.048	729.48	953.81
0.25	0.204	0.035	0.168	0.128	0.194	1.592	1.143	628.25	875.11
0.2	0.118	0.009	0.109	0.052	0.079	2.258	1.546	442.79	646.72
0.15	0.069	0.001	0.067	0.02	0.03	3.505	2.334	285.34	428.41

5.4 22 nm Bulk CMOS PTM

Table 5.13: H-spice [4] simulation of 16 bit ripple carry adder for 22 nm technology node in bulk CMOS PTM [7] at different voltages (V_{dd}).

V_{dd}	Power from simulation				Timing from simulation		Energy per cycle		
volts	p_{avg} μW	p_{dyn} μW	p_{static} μW	p_{peak} μW	Critical path delay, ps	f_{max} GHz	e_{dyn} fJ	e_{static} fJ	e_{total} fJ
0.8	32.75	18.46	14.29	90.24	413.37	2.42	7.63	5.91	13.54
0.7	16.18	10.04	6.14	46.5	577.69	1.731	5.8	3.55	9.35
0.6	6.98	4.35	2.63	20.19	972.36	1.028	4.23	2.56	6.78
0.5	2.45	1.346	1.103	6.52	2088.8	0.479	2.81	2.3	5.12
0.4	0.692	0.242	0.45	1.508	6820.2	0.147	1.65	3.07	4.72
0.35	0.357	0.088	0.27	0.66	14164	0.071	1.242	3.82	5.06
0.3	0.19	0.029	0.16	0.303	30658	0.033	0.89	4.92	5.81
0.25	0.099	0.007	0.092	0.138	69901	0.014	0.507	6.41	6.92
0.2	0.051	0.001	0.05	0.063	165410	0.006	0.189	8.19	8.38
0.15	0.025	0.0016	0.024	0.029	394250	0.003	0.065	9.63	9.69

Table 5.13 shows simulated values of the adder circuit in 22 nm bulk CMOS technology. As it scales down to 22 nm technology, optimum supply voltage scales down to near threshold voltage of 0.8 volts because of the increased leakage at higher supply voltage. The minimum energy point for adder circuit occurs at around 0.4 volts, which is about 4.72 fJ. To optimize for energy and performance in 22 nm, we chose intel core i7 3820QM processor [8] with specification available in Table 5.14.

The Intel Core i7-3820QM [8] is a fast quad-core processor for laptops based on the Ivy Bridge architecture. Due to Hyper-threading, the four cores can handle up to eight threads in parallel leading to better utilization of the CPU. Each core offers a base speed of 2.7 GHz but can dynamically increase clock rates with Turbo Boost up to 3.5 GHz (for 4 active cores), 3.6 GHz (for 2 active cores) and 3.7 GHz (for 1 active core). The CPUs are produced in 22nm (versus 32nm Sandy Bridge CPUs) and are the first to introduce 3D transistors for increased energy efficiency when compared to similarly clocked Sandy Bridge processors. The performance of the Core i7-3820QM is slightly above a similarly clocked Sandy Bridge processor due to architectural improvements.

Table 5.14: Intel Core i7 3820QM processor specifications [8].

Technology node	22 nm
Voltage range	0.8-1.25 volts
Nominal base frequency, f_{TDP}	2.7 GHz
Overclock frequency, f_{max}	3.8 GHz
Thermal Design Power, TDP	45 watts

Table 5.15: Scale factors (Adder to Processor).

Scale factors	Calculated values
Voltage factor, σ	1
Area factor, β	1.2896×10^5
f_{nom} factor, δ	1.1161
f_{max} factor, γ	1.571

Table 5.16: Scaled values for intel Core i7 3820QM processor [8] for 22 nm technology node in bulk CMOS PTM [7] at different voltages (V_{dd}).

V_{dd}	Scaled Power			Scaled Frequency		Energy per cycle		Cycle efficiency	
	TDP	P_{dyn}	P_{static}	f_{nom}	f_{max}	E_{fnom}	E_{fmax}	10^6 cycles/J	
volts	W	W	W	GHz	GHz	nJ	nJ	η	η_0
0.8	45	26.57	18.43	2.7	3.8	16.67	14.69	60	68.07
0.7	22.37	14.46	7.92	1.9320	2.72	11.58	10.39	86.36	96.21
0.6	9.65	6.25	3.39	1.1478	1.6155	8.4	7.55	118.98	132.47
0.5	3.36	1.94	1.4229	0.5343	0.752	6.29	5.52	159.04	181.26
0.4	0.9288	0.3482	0.5806	0.1636	0.2303	5.68	4.65	176.19	215.12
0.35	0.4739	0.1262	0.3477	0.0788	0.1109	6.01	4.74	166.27	211.1
0.3	0.2488	0.0418	0.207	0.0364	0.0512	6.83	5.19	146.35	192.78
0.25	0.1288	0.0104	0.1184	0.016	0.0225	8.07	5.92	123.97	168.9
0.2	0.0655	0.0016	0.0639	0.0067	0.0095	9.71	6.97	102.96	143.44
0.15	0.0317	0.0002	0.0315	0.0028	0.004	11.21	7.99	89.21	125.17

The scale factors for intel i7-3820QM processors are given in Table 5.15 and the scaled values for the processor are shown in Table 5.16. The minimum energy point for the processor occurs at sub-threshold voltage 0.38 volts and energy per cycle is measured as, $EPC_{fnom} =$ 5.882 nJ for nominal frequency and $EPC_{fmax} = 4.673$ nJ for maximum frequency. As we know that cycle efficiency, η is defined as 1/EPC, therefore cycle efficiency, η corresponding to EPC_{fnom} is 170×10^6 cycles/J and peak cycle efficiency, η_0 corresponding to EPC_{fmax} is 213.99×10^6 cycles/J.

5.5 22 nm High-K PTM

Table 5.17: H-spice [4] simulation of 16 bit ripple carry adder for 22 nm technology node in high-K CMOS PTM [7] at different voltages (V_{dd}).

V_{dd}	Power from simulation				Timing from simulation		Energy per cycle		
volts	p_{avg} μW	p_{dyn} μW	p_{static} μW	p_{peak} μW	Critical path delay, ps	f_{max} GHz	e_{dyn} fJ	e_{static} fJ	e_{total} fJ
0.8	59	43	16	109	162	6.17	6.97	2.59	9.56
0.7	36	26.93	9.07	67	186	5.38	5.01	1.69	6.7
0.6	20	15.03	4.97	34	231	4.33	3.47	1.15	4.62
0.5	9.5	7	2.5	15	323	3.1	2.26	0.81	3.07
0.4	3.47	2.36	1.11	5.17	592	1.689	1.397	0.657	2.05
0.35	1.84	1.135	0.705	2.76	966	1.035	1.096	0.681	1.777
0.3	0.838	0.399	0.439	1.31	1851	0.54	0.739	0.813	1.551
0.25	0.382	0.115	0.267	0.564	4145	0.241	0.477	1.107	1.583
0.2	0.186	0.025	0.161	0.24	10406	0.096	0.26	1.675	1.936
0.15	0.097	0.004	0.093	0.115	28050	0.036	0.112	2.609	2.721

Table 5.17 shows simulated values of the adder circuit in 22 nm High-K technology model. The minimum energy point for adder circuit occurs at around 0.3 volts, which is about 1.55 fJ. The scale factors for intel i7-3820QM processors are given in Table 5.19 and the scaled values for the processor are shown in Table 5.21. The minimum energy point for the processor occurs at sub-threshold voltage 0.30 volts and energy per cycle is measured as, EPC_{fnom} = 3.36 nJ for nominal frequency and EPC_{fmax} = 2.66 nJ for maximum frequency. As we know that cycle efficiency, η is defined as 1/EPC, therefore cycle efficiency, η corresponding to EPC_{fnom} is 297.93$\times 10^6$ cycles/J and peak cycle efficiency, η_0 corresponding to EPC_{fmax} is 375.76$\times 10^6$ cycles/J.

Table 5.18: Intel Core i7 3820QM processor specifications [8].

Technology node	22 nm
Voltage range	0.8-1.25 volts
Nominal base frequency, f_{TDP}	2.7 GHz
Overclock frequency, f_{max}	3.8 GHz
Thermal Design Power, TDP	45 watts

Table 5.19: Scale factors (Adder to Processor).

Scale factors	Calculated values
Voltage factor, σ	1
Area factor, β	1.2928×10^5
f_{nom} factor, δ	0.4374
f_{max} factor, γ	0.6156

Table 5.20: Scaled values for intel Core i7 3820QM processor [8] for 22 nm technology node in high-K CMOS PTM [7] at different voltages (V_{dd}).

V_{dd}	Scaled Power			Scaled Frequency		Energy per cycle		Cycle efficiency	
	TDP	P_{dyn}	P_{static}	f_{nom}	f_{max}	E_{fnom}	E_{fmax}	10^6 cycles/J	
volts	W	W	W	GHz	GHz	nJ	nJ	η	η_0
0.8	45	24.32	20.68	2.7	3.8	16.67	14.45	60	69.21
0.7	26.95	15.23	11.73	2.35	3.31	11.46	10.02	87.25	99.82
0.6	14.92	8.5	6.43	1.8935	2.66	7.88	6.9	126.87	144.94
0.5	7.19	3.96	3.23	1.3542	1.9059	5.31	4.62	188.33	216.51
0.4	2.77	1.3345	1.435	0.7389	1.0399	3.75	3.19	266.78	313.85
0.35	1.5532	0.6418	0.9114	0.4528	0.6373	3.43	2.85	291.52	351.17
0.3	0.7932	0.2256	0.5675	0.2363	0.3326	3.36	2.66	297.93	375.76
0.25	0.4102	0.065	0.3452	0.1055	0.1485	3.89	2.94	257.25	340.09
0.2	0.2223	0.0141	0.2081	0.042	0.0592	5.29	3.85	189.1	259.42
0.15	0.1225	0.0023	0.1202	0.0156	0.0219	7.86	5.62	127.3	177.83

5.6 Summary

Using our power management methodology on the data obtained in Sections 5.1 through 5.5, performance and energy optimization for both bulk and High-K technologies in 45 nm, 32 nm and 22 nm transistor size are summarized in Table 5.21. These results lead us to following observations:

Table 5.21: Performance and energy optimization for Intel processors characterized using various PTM [7] models.

PTM Model	Intel Chip	Nominal Operation					Performance Optimization			Energy Optimization		
		Rated Specifications			Optimized							
		f_{TDP} MHz	V_{dd} V	η_{TDP} Mc/J	V_{dd} V	η_0 Mc/J	V_{ddopt} V	f_{opt} MHz	η_{opt} Mc/J	V_{dd} V	f_{η_0} MHz	η_0 Mc/J
45 nm Bulk	Core2 Duo T9500	2600	1.25	74.29	1.07	108.58	1.2	2920	82.28	0.35	33.51	829.29
45 nm High-K	Core2 Duo T9500	2600	1.25	74.29	0.79	350.91	1.226	3120	89.08	0.30	304.48	1795
32 nm Bulk	Core i5 2500K	3300	1.2	34.74	0.92	79.01	1.112	4531	47.91	0.35	36.39	384.45
32 nm High-K	Core i5 2500K	3300	1.2	34.74	0.67	267.57	1.155	4940	51.77	0.30	414.2	953.81
22 nm Bulk	Core i7 3820QM	2700	0.8	60	0.7	96.22	0.771	3494	75.46	0.38	177.3	213.99
22 nm High-K	Core i7 3820QM	2700	0.8	60	0.61	137.65	0.76	3626	80.38	0.30	332.6	375.76

1. **Optimizing Nominal Operation (columns 5 and 7):** For nominal clock frequency, optimized efficiency is always higher than the efficiency for the specified operation. This is accomplished by lowering the supply voltage.

2. **Bulk vs High K:** When we compare the two PTM models, we observe that High-K consistently has higher frequency as well cycle efficiency. This is perhaps due to the reduced leakage.

3. **Performance Optimization (columns 8-10):** Clock rate can be increased by suitably lowering the voltage, but efficiency drops (compare columns 7 and 10), Still, this efficiency is marginally superior to the rated specification (compare columns 5 and 10).

4. **Energy Optimization (columns 11-13):** Efficiency increases almost by an order of magnitude over that for the rated specification (compare columns 5 and 13), even though the performance in the sub-threshold voltage region (column 11) is reduced almost by an order of magnitude (compare columns 3 and 12).

In interpreting the available information on the specifications and structure of these processors, we have made several assumptions. Hence the data presented and the observations made here may not exactly represent the behavior of Intel processors. Nevertheless, the purpose of this investigation is to present a methodology for evaluation of processors for performance and energy optimization.

Chapter 6

Conclusion

Since the late 1980s higher performance has been the most important driving force behind processor evolution. For the past 10 or 15 years designers have doubled processor performance every 18 to 24 months. Unfortunately, designers paid little or no attention to power. The result is large and growing power levels in processors. This thesis has explored how power management affects the energy and performance of a processor. This research shows that performance (execution time and energy consumption) of a processor is optimized when operated at a voltage such that the highest clock frequency allowed by the critical path will consume the thermal design power (TDP).

6.1 Achievements

The proposed power management method was entirely a simulation based evaluation and by introducing such method, we accomplished the goal of performance and energy optimization with the observations concluded below:

1. Highest performance mode has better sustained clock rate than the rated (nominal or specified) clock rate.

2. Highest efficiency at the rated clock requires lowering of voltage.

3. Performance in both of these modes can be further increased by over-clocking, which essentially requires increasing voltage whenever frequency is increased.

4. Highest efficiency with no performance bound is a sub-threshold operation with clock in mega-hertz range.

6.2 Future Work

It has been realized that energy efficiency will continue to be a major issue in processor design and applications [15]. This focus of this dissertation has been to create an optimization framework, but it is important to clearly define the scope of this work and acknowledge areas where work still needs to be done. Although this work presents solutions to existing problems, it has also opened the door for other research venues and some of them are briefly discussed below:

1. Analysis and simulation in this work did not consider process variability that is especially important in nanometer technologies.

2. With the proposed optimized operation, *over-clocking* (raising performance by short bursts of power) is possible but has not been discussed. It will, however, essentially require voltage boost. This is because the optimum clock is fastest that critical path would support at the selected voltage.

3. Highest efficiency operation is in the sub-threshold voltage region, which may be sensitive to the thermal as well as other types of noise. Reliability of sub-threshold operation requires study.

4. We notice that energy efficiency increases as voltage is reduced. For a given performance, operating voltage should be lowest that will allow that frequency. This suggests further exploration of the near (but above) threshold range [23] of V_{dd} where significant increase in energy efficiency may be possible with only minor loss of performance.

5. In this work the signal activity of the ripple carry adder (RCA) was assumed to be the same as that of the processor. Here any differences in the activity are implicitly compensated for by adjustment of the area scale factor. Alternatively, a separate scale factor can be defined as the ratio of activity factors of the two circuits.

Bibliography

[1] "International Technology Road-map for Semiconductors (ITRS), 2010 Update, Emerging Research Devices (ERD)." Prepared by Semiconductor Industry Association (SIA) and Others.

[2] "MATLAB." MathWorks. www.mathworks.com/products/matlab/.

[3] *Design Architect User Guide*. Wilsonville, OR: Mentor Graphics Corp., 1991-1995.

[4] *HSPICE Signal Integrity User Guide*. 700 East Middlefield Road, Mountain View, CA 94043: Synopsys, Inc., 2010.

[5] *Leonardo Spectrum User Guide*. Wilsonville, OR: Mentor Graphics Corp., 2011.

[6] *Questa Sim User Guide*. Wilsonville, OR: Mentor Graphics Corp., 2011.

[7] "Predictive Technology Model." Nanoscale Simulation and Modeling (NIMO) Group, Arizona State University, 2012. website: ptm.asu.edu.

[8] "Intel Core i-7-3820QM Processor Specifications." `http://ark.intel.com/products/64889/Intel-Core-i7-3820QM-Processor-8M-Cache-up-to-3_70-GHz`, 2016. [Online; accessed 24-Feb-2016].

[9] "Intel Core2 Duo T9500 Processor Specifications." `http://ark.intel.com/products/33918/intel-core2-duo-processor-t9500-6m-cache-2_60-ghz-800-mhz-fsb`, 2016. [Online; accessed 22-Feb-2016].

[10] A. Agarwal, K. Kang, S. K. Bhunia, J. D. Gallagher, and K. Roy, "Effectiveness of Low Power Dual-V_t Designs in Nanoscale Technologies Under Process Parameter Variations," in *Proc. International Symp. on Low Power Electronics and Design*, IEEE, 2005, pp. 14–19.

[11] V. D. Agrawal, "Low Power Design by Hazard Filtering," in *Proc. 10th International Conf. VLSI Design*, Jan. 1997, pp. 193–197.

[12] V. D. Agrawal, M. L. Bushnell, G. Parthasarathy, and R. Ramadoss, "Digital Circuit Design for Minimum Transient Energy and a Linear Programming Method," in *Proc. 12th International Conf. VLSI Design*, Jan. 1999, pp. 434–439.

[13] M. Annavaram, E. Grochowski, and J. Shen, "Mitigating Amdahl's Law Through EPI Throttling," in *Proc. 32nd IEEE International Symposium on Computer Architecture (ISCA)*, 2005, pp. 298–309.

[14] P. Bai, C. Auth, S. Balakrishnan, M. Bost, R. Brain, V. Chikarmane, R. Heussner, M. Hussein, J. Hwang, D. Ingerly, *et al.*, "A 65nm Logic Technology Featuring 35nm Gate Lengths, Enhanced Channel Strain, 8 Cu Interconnect Layers, Low-k ILD and 0.57 μm 2 SRAM Cell," in *Technical Digest, IEEE International Electron Devices Meeting (IEDM)*, 2004, pp. 657–660.

[15] S. Borkar and A. A. Chien, "The Future of Microprocessors," *Communications of ACM*, vol. 54, no. 5, pp. 67–77, May 2011.

[16] B. Brock and K. Rajamani, "Dynamic Power Management for Embedded Systems," in *Proc. IEEE International SOC Conference*, (Portland, OR), 2003, pp. 1–4.

[17] T. D. Burd, T. A. Pering, A. J. Stratakos, and R. W. Brodersen, "A Dynamic Voltage Scaled Microprocessor System," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 11, pp. 1571–1580, 2000.

[18] B. H. Calhoun and A. P. Chandrakasan, "Characterizing and Modeling Minimum Energy Operation for Sub-threshold Circuits," in *Proc. International Symposium on Low Power Electronics and Design*, (Newport Beach, CA), 2004, pp. 90–95.

[19] Y. Cao, *Predictive Technology Model for Robust Nanoelectronic Design*. Springer, 2011.

[20] A. P. Chandrakasan, W. J. Bowhill, and F. Fox, *Design of High-Performance Microprocessor Circuits*. Wiley-IEEE press, 2000.

[21] A. P. Chandrakasan and R. W. Brodersen, *Low Power Digital CMOS Design*. Springer, 1995.

[22] R. H. Dennard, F. H. Gaensslen, H.-N. Yu, V. Leo Rideovt, E. Bassous, and A. R. Leblanc, "Design of Ion-Implanted MOSFET's With Very Small Physical Dimensions," *IEEE Solid-State Circuits Society Newsletter*, vol. 12, no. 1, pp. 38–50, 2007.

[23] R. G. Dreslinski, M. Wieckowski, D. Blaauw, D. Sylvester, and T. Mudge, "Near-Threshold Computing: Reclaiming Moore's Law Through Energy Efficient Integrated Circuits," *Proc. IEEE*, vol. 98, pp. 253–266, Feb. 2010.

[24] R. Gonzalez and M. Horowitz, "Energy Dissipation in General Purpose Microprocessors," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 9, pp. 1277–1284, 1996.

[25] R. E. Gonzalez, *Low Power Processor Design*. PhD thesis, Citeseer, 1997.

[26] K. Govil, E. Chan, and H. Wasserman, "Comparing Algorithm for Dynamic Speed-Setting of a Low-Power CPU," in *Proc. 1st Annual International ACM Conference on Mobile Computing and Networking*, 1995, pp. 13–25.

[27] H. Goyal and V. D. Agrawal, "Characterizing Processors for Energy and Performance Management," in *Proc. 16th International Workshop on Microprocessor/SoC Test and Verification (MTV)*, (Austin, TX), Dec. 2015.

[28] H. Goyal and V. D. Agrawal, "Characterizing Processors for Energy and Performance Management," in *Proc. 34th IEEE VLSI Test Symp. (VTS)*, (Las Vegas, NV), Apr. 2016. Poster.

[29] E. Grochowski, R. Ronen, J. Shen, and H. Wang, "Best of Both Latency and Throughput," in *Proc. International Conf. on Computer Design*, 2004, pp. 236–243.

[30] D. Grunwald, C. B. Morrey III, P. Levis, M. Neufeld, and K. I. Farkas, "Policies for Dynamic Clock Scheduling," in *Proc. 4th Symposium on Operating System Design & Implementation, Volume 4*, USENIX Association, 2000, p. 6.

[31] F. Ichiba, K. Suzuki, S. Mita, T. Kuroda, and T. Furuyama, "Variable Supply-Voltage Scheme With 95%-Efficiency DC-DC Converter for MPEG-4 Codec," in *Proc. International Symp. on Low Power Electronics and Design*, ACM, 1999, pp. 54–59.

[32] Intel, "Intel Core i5-2500K Processor Specifications." `http://ark.intel.com/products/52210/Intel-Core-i5-2500K-Processor-6M-Cache-up-to-3\_70-GHz`, 2016. [Online; accessed 20-Feb-2016].

[33] S. K. Jha, "Challenges on Design Complexities for Advanced Wireless Silicon Systems," in *Proc. ACM/IEEE Design Automation Conference*, IEEE, 2008, pp. xii–xii.

[34] S.-M. Kang and Y. Leblebici, *CMOS Digital Integrated Circuits*. Tata McGraw-Hill Education, 2003.

[35] A. Keshavarzi, K. Roy, and C. F. Hawkins, "Intrinsic Leakage in Low Power Peep Submicron CMOS ICs," in *Proc. International Test Conference*, IEEE, 1997, pp. 146–155.

[36] M. Khare, S. H. Ku, R. A. Donaton, S. Greco, C. Brodsky, X. Chen, A. Chou, R. DellaGuardia, S. Deshpande, B. Doris, *et al.*, "A High Performance 90nm SOI Technology With 0.992/spl mu/m 2 6T-SRAM Cell," in *Proc. International Electron Devices Meeting (IEDM)*, IEEE, 2002, pp. 407–410.

[37] K. Knauer, "Ripple-Carry Adder," June 13 1989. US Patent 4,839,849.

[38] M. Kulkarni, *Energy source lifetime optimization for a digital system through power management.* PhD thesis, Auburn University, 2010.

[39] J. B. Kuo and S.-C. Lin, *Low-Voltage SOI CMOS VLSI Devices and Circuits.* John Wiley & Sons, 2004.

[40] G. E. Moore, "Cramming More Components onto Integrated Circuits," *IEEE Solid-State Circuits Newsletter*, vol. 3, no. 20, pp. 33–35, 2006. Reprinted from *Electronics*, vol. 38, no. 8, April 19, 1965.

[41] T. Mudge, "Power: A First Class Design Constraint for Future Architectures," in *High Performance Computing*, pp. 215–224, Springer, 2000.

[42] L. W. Nagel, *A Computer Program to Simulate Semiconductor Circuits.* PhD thesis, University of California, Berkeley, CA, May 1975.

[43] L. W. Nagel and D. O. Pederson, "SPICE - Simulation Program with Integrated Circuit Emphasis," Memo ERL-M382, University of California, Berkeley, CA, Apr. 1973.

[44] P. R. Panda, A. Shrivastava, B. V. N. Silpa, and K. Gummidipudi, "Basic Low Power Digital Design," in *Power-Efficient System Design*, chapter 2, pp. 11–39, Springer, 2010.

[45] P. R. Panda, B. V. N. Silpa, A. Shrivastava, and K. Gummidipudi, *Power-Efficient System Design.* Springer, 2010. Chapter 2: "Basic Low Power Digital Design," pp. 11-39.

[46] D. Patil, O. Azizi, M. Horowitz, R. Ho, and R. Ananthraman, "Robust Energy-Efficient Adder Topologies," in *Proc. 18th IEEE Symposium on Computer Arithmetic*, IEEE, 2007, pp. 16–28.

[47] T. Pering, T. Burd, and R. Brodersen, "Dynamic Voltage Scaling and the Design of a Low-Power Microprocessor System," in *Proc. Power Driven Micro-architecture Workshop (Held with ISCA98)*, 1998, pp. 96–101.

[48] R. Puri, "Minimizing Power Under Performance Constraint," in *Proc. IEEE International Conf. on Integrated Circuit Design and Technology*, 2004, pp. 159–163.

[49] A. Raghunathan, S. Dey, and N. K. Jha, "Glitch Analysis and Reduction in Register Transfer Level Power Optimization," in *Proceedings of 33rd Design Automation Conference*, 1996, pp. 331–336.

[50] S. Raghunathan, S. K. Gupta, H. S. Markandeya, P. P. Irazoqui, and K. Roy, "Ultra Low-Power Algorithm Design for Implantable Devices: Application to Epilepsy Prostheses," *Jour. Low Power Electronics and Design*, vol. 1, no. 1, pp. 175–203, 2011.

[51] T. Raja, V. D. Agrawal, and M. L. Bushnell, "Transistor Sizing of Logic Gates to Maximize Input Delay Variability," *Jour. Low Power Electronics*, vol. 2, no. 3, pp. 121–128, Dec. 2006.

[52] J. Rubio, K. Rajamani, F. Rawson, H. Hanson, S. Ghiasi, and T. Keller, "Dynamic Processor Over-Clocking for Improving Performance of Power-Constrained Systems," Technical Report RC23666 (W0507-124), IBM Research Division, Austin Research Laboratory, 11501 Burnet Road, Austin, TX 78758, July 2005.

[53] T. Sakurai, "CMOS Inverter Delay and Other Formulas Using Alpha-Power Law MOS Model," in *Proc. International Conf. Computer-Aided Design*, 1988, pp. 74–77.

[54] T. Sakurai and A. R. Newton, "Alpha-Power Law MOSFET Model and Its Applications to CMOS Inverter Delay and Other Formulas," *IEEE Journal of Solid-State Circuits*, vol. 25, no. 2, pp. 584–594, 1990.

[55] A. Shinde and V. D. Agrawal, "Managing Performance and Efficiency of a Processor," in *Proc. 45th IEEE Southeastern Symposium on System Theory (SSST)*, 2013, pp. 59–62.

[56] A. J. Shinde, "Managing Performance and Efficiency of a Processor," Master's thesis, Auburn University, Auburn, AL, Dec. 2012.

[57] M. Tehranipoor and K. M. Butler, "Power Supply Noise: A Survey on Effects and Research," *IEEE Design & Test of Computers*, vol. 27, no. 2, pp. 51–67, March/April 2010.

[58] Y. Tsividis, *Operation and Modeling of the MOS Transistor*. Oxford University Press, 2003.

[59] S. Uppalapati, "Low Power Design of Standard Cell Digital VLSI Circuits," Master's thesis, Rutgers University, New Brunswick, NJ, 2004.

[60] H. J. M. Veendrick, "Short-Circuit Dissipation of Static CMOS Circuitry and its Impact on the Design of Buffer Circuits," *IEEE Journal of Solid-State Circuits*, vol. 19, no. 4, pp. 468–473, 1984.

[61] P. Venkataramani, *Reducing ATE Test Time by Voltage and Frequency Scaling.* PhD thesis, Auburn University, Auburn, AL, May 2014.

[62] P. Venkataramani and V. D. Agrawal, "Reducing Test Time of Power Constrained Test by Optimal Selection of Supply Voltage," in *Proc. 26th International Conf. VLSI Design*, Jan. 2013, pp. 273–278.

[63] P. Venkataramani, S. Sindia, and V. D. Agrawal, "A Test Time Theorem and its Applications," *Journal of Electronic Testing: Theory and Applications*, vol. 30, no. 2, pp. 229–236, 2014.

[64] A. Wang and A. P. Chandrakasan, "A 180-mV Sub-threshold FFT Processor Using a Minimum Energy Design Methodology," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 1, pp. 310–319, 2005.

[65] W. Zhao and Y. Cao, "New Generation of Predictive Technology Model for Sub-45 nm Early Design Exploration," *IEEE Transactions on Electron Devices*, vol. 53, no. 11, pp. 2816–2823, Nov. 2006.