

Energy Source Lifetime Optimization for a Digital System through Power Management

by

Manish Kulkarni

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
Dec 13, 2010

Keywords: Low Power Architecture, Power Source Optimization, Li-ion Battery Simulations

Copyright 2010 by Manish Kulkarni

Approved by

Vishwani Agrawal, Chair, James J. Danaher Professor, Electrical and Computer Engineering

Adit Singh, James B. Davis Professor, Electrical and Computer Engineering

Victor Nelson, Professor, Electrical and Computer Engineering

Abstract

This work analyzes a typical battery powered digital electronic system and we propose a system level voltage scaling method and a functional power management method called instruction slowdown for low power. In the first part, we examine a circuit with voltage scaling capability and observe its impact on the energy efficiency of the battery. We study the system with a power source under throughput constraints and we propose a method to find a right size of battery to satisfy given system requirements. For systems with limit on battery weight or volume, we suggest a right circuit voltage operating point. We also notice that the performance evaluation metric such as battery discharge-delay or number of cycles per recharge are more relevant when power source optimization is a primary goal. In the later part of this work, an instruction named slowdown for low power (SLOP) is introduced. Functionally, it resembles the conventional NOP but requires power-specific hardware implementation. Depending upon the power reduction requirement, adequate number of SLOPs are automatically inserted in the instruction stream by the power management hardware. A possibility also exists to allow compiler or programmer to insert SLOPs in order to create programs which would have flexibility to run in either normal mode or in low power mode. While processing a SLOP, additional power control signals are generated for various units; so they can be powered down or clock gated. Simulation of a five-stage pipelined 32-bit MIPS processor shows that the SLOP method, termed instruction slowdown (ISD), becomes more effective than a conventional clock slowdown (CSD) when leakage is high. For 32nm CMOS technology, ISD can save more than 70% power compared to about 40% by CSD. The work shows that power reduction through a judicious choice of slowdown factor and the method adopted, clock slowdown for low leakage and instruction slowdown for high leakage, can enhance the battery lifetime.

Acknowledgments

My advisor and committee were the people most directly involved with the completion of my thesis. I would like to express my appreciation and sincere thanks to my advisor Dr. Vishwani Agrawal, who patiently shaped this work as it developed through a series of false starts and dead ends. I benefited greatly from his ability to approach problems from many different directions. His advise and attitude towards life would remain a guiding light for me throughout my career. I also wish to thank my advisory committee members, Dr. Adit Singh and Dr. Victor Nelson for their guidance and advice on this work.

My work could not have been completed without a substantial support from Dr. Prathima Agrawal, for which I am grateful. I would also like to thank my advisor for providing me with an opportunity to work as a teaching assistant for CPU design projects in his Computer Architecture and Design class. This was one of the most fun and learning experiences during my master's studies. A number of people at Auburn University, including Nitin, Kim, Sree, Wei, provided help during this work, for which I am thankful. Thanks are also expressed to integration team, especially Sumeeth and Raghu, at ARM, Bangalore, for a truly memorable first industry experience. My special thanks to Ellie and Glynn O'Steen who treated me as a family member, cared for me and whose loving support kept me going.

I gratefully acknowledge financial support at Auburn University derived from a research grant received as a gift from Intel Corporation.

Finally, I would like to thank my parents, siblings and my friends Anand, Aniket, Deepti, Ameya, Saba, Indraneil, Salil for their encouragement and support during this work.

Thank you, all of you.

Manish

September 28, 2010

Table of Contents

| | |
|---|-----|
| Abstract | ii |
| Acknowledgments | iii |
| List of Figures | vii |
| List of Tables | x |
| List of Abbreviations | xi |
| 1 Introduction | 1 |
| 2 Theory and Background Work on Low Power Design | 5 |
| 2.1 Introduction | 5 |
| 2.1.1 Need for Low Power VLSI chips | 5 |
| 2.1.2 Power Vs. Energy | 7 |
| 2.2 Where Does All the Power Go? | 8 |
| 2.2.1 Dynamic Power Dissipation | 8 |
| 2.2.2 Static Power Dissipation | 9 |
| 2.2.3 The conflict between Dynamic and Static Power | 14 |
| 2.3 Low Power Design Methods | 14 |
| 2.3.1 Circuit Level Methods | 15 |
| 2.3.2 Gate Level Methods | 18 |
| 2.3.3 Architecture or System Level Methods | 21 |
| 2.4 Power Source Optimization: A System Approach | 28 |
| 2.4.1 Choice of Metric | 28 |
| 2.4.2 Classification of Power Source Optimization Methods | 29 |
| 2.4.3 A Typical Battery Powered Electronic System | 31 |
| 3 Lithium-ion Battery Background and Modelling | 33 |

| | | |
|-----|---|----|
| 3.1 | Background | 33 |
| 3.2 | Electro-chemistry | 36 |
| 3.3 | Description of Terminology | 38 |
| | 3.3.1 Capacity | 38 |
| | 3.3.2 Rate Dependent Capacity | 38 |
| | 3.3.3 Temperature Effect | 39 |
| | 3.3.4 Capacity Fading | 39 |
| 3.4 | Modelling | 40 |
| | 3.4.1 Physical Models | 40 |
| | 3.4.2 Empirical Models | 41 |
| | 3.4.3 Abstract Models | 41 |
| | 3.4.4 Analytical/Mixed Models | 42 |
| 3.5 | Model Used for This Work | 43 |
| | 3.5.1 Description | 44 |
| | 3.5.2 Battery Lifetime | 46 |
| | 3.5.3 Voltage and Current Characteristics | 46 |
| 3.6 | Summary | 47 |
| 4 | DC to DC Converter | 48 |
| | 4.1 Necessity | 48 |
| | 4.2 Topologies of Switching Regulators | 49 |
| | 4.2.1 Buck Converter | 49 |
| | 4.3 Summary | 53 |
| 5 | System Approach for Power Source Optimization | 54 |
| | 5.1 Introduction | 54 |
| | 5.2 Problem Definition | 54 |
| | 5.3 Case I: System is performance bound | 56 |
| | 5.3.1 Step 1: Determine circuit characteristics | 56 |

| | | |
|-------|--|----|
| 5.3.2 | Step 2: Determine smallest battery size | 58 |
| 5.3.3 | Step 3: Meeting the lifetime requirement | 60 |
| 5.3.4 | Step 4: Determine minimum energy modes | 61 |
| 5.4 | Case II: Battery size or weight is a primary concern | 63 |
| 5.5 | Summary | 65 |
| 6 | Instruction Slowdown Method | 66 |
| 6.1 | Problem Statement | 66 |
| 6.2 | Background on Clock Slowdown (CSD) for Power Reduction | 67 |
| 6.3 | Use of NOP for Power | 69 |
| 6.4 | Instruction Slowdown (ISD) | 70 |
| 6.5 | Hardware Implementation of SLOP | 72 |
| 6.6 | Estimating Leakage Factor, \underline{k} | 72 |
| 6.7 | Power Management for SLOP | 74 |
| 6.8 | Results | 77 |
| 7 | Conclusion | 83 |
| | Bibliography | 85 |

List of Figures

| | | |
|------|--|----|
| 2.1 | Growth in energy densities of Lithium-ion batteries | 6 |
| 2.2 | Limit on the growth of battery energy densities | 6 |
| 2.3 | A CMOS Inverter circuit | 10 |
| 2.4 | Short circuit currents of CMOS inverter during input transition | 11 |
| 2.5 | Leakage Currents for nMOS transistor | 12 |
| 2.6 | Design flow and type of tools at different levels of abstraction[22] | 16 |
| 2.7 | Two different implementations of a 4-input AND gate[22] | 18 |
| 2.8 | Various Implementations of Signal Gating [20] | 20 |
| 2.9 | Different Sleep modes supported by Intel Pentium 4 Mobile [16] | 22 |
| 2.10 | Power Dissipation of uniprocessing and parallel processing systems | 26 |
| 2.11 | Powering and Electronic System | 31 |
| 3.1 | An Electrical Model for Lithium-ion battery | 43 |
| 4.1 | Types of Converters | 50 |
| 4.2 | A Simple Buck Converter | 52 |
| 4.3 | Buck Converter output waveform | 52 |

| | | |
|-----|--|----|
| 5.1 | Circuit Delay and Current versus VDD obtained from HSPICE simulations . . . | 57 |
| 5.2 | V_{Batt} Vs Time when a battery of 1.2 AHr capacity is subjected to load current, $I_{Batt} = 3.6A$ | 59 |
| 5.3 | Battery efficiency versus battery size for various load currents | 60 |
| 5.4 | Simulation of a 400 mAHr battery for a range of supply voltages (VDD) | 61 |
| 5.5 | Battery lifetimes in clock cycles as a function of chip voltage for 400 mAHr and 1600 mAHr batteries | 62 |
| 5.6 | Battery lifetimes in clock cycles as a function of chip voltage for 400 mAHr and 1600 mAHr batteries | 63 |
| 5.7 | Battery lifetimes in number of clock cycles for CR2032 with max. $I_{battery} = 3mA$ | 64 |
| 6.1 | Clock slowdown (CSD) power and battery lifetime ratios for low and high leakage technologies. | 69 |
| 6.2 | Instruction slowdown (ISD) power and battery lifetime ratios for low and high leakage technologies. | 71 |
| 6.3 | A MIPS program used for power estimation. | 73 |
| 6.4 | Clock slowdown (CSD) power ratios for 180nm, 90nm, 65nm, 45nm and 32nm CMOS technologies. CSD is more effective for low leakage (180nm) technology. | 75 |
| 6.5 | Clock slowdown (CSD) battery lifetime ratios for 180nm, 90nm, 65nm, 45nm and 32nm CMOS technologies. Ratios greater than 1 indicate increased battery lifetime through clock slowdown for low leakage 90nm and 180nm technologies. . | 76 |

| | | |
|------|--|----|
| 6.6 | Instruction slowdown (ISD) power ratios for 180nm, 90nm, 65nm, 45nm and 32nm CMOS technologies. ISD gives greater power saving for higher leakage technologies. | 77 |
| 6.7 | Instruction slowdown (ISD) battery lifetime ratios for 180nm, 90nm, 65nm, 45nm and 32nm CMOS technologies. Ratios greater than 1 indicate increased or undegraded battery lifetime through instruction slowdown for high leakage 32nm and 45nm technologies. | 78 |
| 6.8 | Clock slowdown (CSD) vs. instruction slowdown (ISD) power ratios for 180nm, 90nm, 65nm, 45nm and 32nm CMOS technologies. <i>Ratio</i> > 1.0 indicates the advantage of ISD for 32nm and 45nm technologies. | 79 |
| 6.9 | Clock slowdown (CSD) vs. instruction slowdown (ISD) battery lifetime ratios for 180nm, 90nm, 65nm, 45nm and 32nm CMOS technologies. <i>Ratio</i> < 1.0 indicates the advantage of ISD for 32nm and 45nm technologies. | 80 |
| 6.10 | Power ratio, energy ratio and ideal battery lifetime ratio plotted against slow down factor, n , for ISD in 32nm | 80 |
| 6.11 | Circuit energy, battery lifetime and task completion time plotted against number of SLOPs, for ISD in 32nm | 82 |

List of Tables

| | | |
|-----|---|----|
| 2.1 | ITRS predictions on power dissipation of technology nodes | 7 |
| 5.1 | High performance and minimum energy modes of operation. | 61 |
| 6.1 | HSPICE simulation (32nm CMOS, 90°C). | 81 |
| 6.2 | Leakage factor (k) and SLOP power factor (β). | 82 |

List of Abbreviations

| | |
|------|---|
| CG | Clock Gating |
| CISC | Complex Instruction Set Computer |
| CSD | Clock SlowDown |
| DPCT | Dynamic Power Cut-off Technique |
| DVFS | Dynamic Voltage and Frequency Scaling |
| EPC | Energy Per Cycle |
| GILD | Gate Induced Drain Leakage |
| HDL | Hardware Description Language |
| ISA | Instruction Set Architecture |
| ISD | Instruction SlowDown |
| ITRS | International Technology Roadmap for Semiconductors |
| MIDs | Mobile Internet Devices |
| MIPS | Million Instruction Per Second |
| NiMH | Nickel Metal Hydride |
| NOP | No-Operation |
| PG | Power Gating |
| PMU | Power Management Unit |

PTM Predictive Technology Models

RISC Reduced Instruction Set Computer

SLOP Slowdown for LOw Power

Chapter 1

Introduction

Every processor chip has a physical limit on power dissipation it can support. For systems that use these processors, performance and power become opposing requirements. Modern computing systems, therefore, have built-in power control schemes. For example, thermal sensors on a processor chip may trigger a slowdown of the processor clock [35].

For mobile systems, energy consumption and the rate of consumption (power) are directly related to the battery capacity. Higher discharge rate reduces the capacity, requiring bulkier batteries with higher current rating [3] or more frequent recharging. Thus, it is important to control the power consumption. Traditional metrics like minimization of Power and Energy are not really suitable when power source (battery) optimization is a concern. For battery operated portable devices, an obvious objective is to maximize the battery lifetime. In spite of this fact, the discussions of low power design metric and methodologies have entirely focused on VLSI sub-system optimizations. The energy stored in a battery is assumed to be constant and available at any possible rate. In reality, however, the energy stored in a battery may not be used to its full extent. The delivery of energy from battery to system depends on the mean value of the current drawn from the battery. Battery lifetime does not have a simple linear relationship with power consumption of the circuit. e.g. a 2X increase in system power can cause a 3X decrease in battery lifetime. These facts motivate us to consider various approaches with design goal of power source optimization. Various approaches have been suggested in the literature and they demonstrate a potential to optimize battery energy consumption. These approaches can be classified into three broad categories.

- Voltage Management Methods
- Throughput Management Methods

- Functional Management Methods

In chapter 5, we suggest a general system level method to identify the load current on the battery and then choose battery of minimum size which can satisfy the required current. We also discuss various modes in which a system can operate in order to achieve maximum energy efficiency. The later part of the chapter focused on optimizing the battery lifetime and finding a right size of battery for a given load current. As far as the portable electronic devices are concerned, the ultimate aim is to achieve more battery lifetime or, for a rechargeable source, perform the most operations between consecutive recharges. Optimization of the circuit alone for power and energy may not always result in equivalent optimization of battery lifetime. So a study of the system consisting of battery and the circuit under consideration has been carried out in order to achieve maximum battery lifetime. In general, this lifetime should be measured in terms of the duration of the system operation. A relevant measure is the number of useful clock cycles obtained per battery life or per battery recharge. Size and weight of the batteries are major design constraints for mobile computing devices. Battery weights are generally proportional to their AHr ratings. Given an application with its load current requirement, a relevant problem is to find a battery with minimum size and weight to run the application. Since the energy drawn from the battery is not always equal to the energy consumed in the device, understanding battery discharge behaviour and its own dissipation are essential for optimal system design.

When excessive power consumption forces *clock slowdown* (CSD), the completion time of the ongoing system task increases. This increases the energy consumption. The energy penalty of the CSD method can be severe for high-leakage technologies. CSD is, therefore, not recommended without voltage scaling [8]. There is, however, another consideration. The reduced power slows the current drain from the battery. For a given battery capacity, this can increase the lifetime of the battery [21, 38]. *Lifetime* here refers to the useful life of a *primary* battery or the time between recharges for a *secondary* (rechargeable) battery. If the increase in the battery lifetime for a portable device is more than the increase in the

execution time of the task, then CSD can be beneficial [2]. Unless the efficiency aspect of the power source is properly considered, the slowing down of a computing task for power reduction would not be recommended. The lack of such consideration often results in the use of oversized batteries as well as over-design for unnecessary power dissipation, cooling, etc.

In the later sections of this thesis, we discuss a scenario where CSD may be necessary. We also find that its power saving advantage diminishes in higher leakage technologies. This leads to our motivation for finding a lower energy penalty alternative. Because clock slowdown (CSD) allows larger delay for hardware, we can further reduce power by lowering the supply voltage. Voltage reduction reduces both power and energy. However, this has limited potential in the nano-meter technologies where the voltage, already lowered due to the electric field requirement, is closer to the threshold voltage. This is particularly so for dual-threshold designs in which high-threshold devices are used to reduce leakage.

When voltage has been scaled down to some limit set by the technology, further power reduction, if necessary due to the system or operational requirements, by CSD will increase the task completion time and the leakage energy. To reduce the energy, a *dynamic power cutoff technique* (DPCT) has been proposed [27]. While DPCT can save both power and energy, it requires turning power off and on for different parts of combinational logic at different times within the clock period. Asynchronous delays for power control signals make the design complex and especially sensitive to process variation.

In this work, we address the need for a power saving method with emphasis on the energy penalty. We propose an *instruction slowdown* (ISD) method, which inserts NOP-like instructions. A new instruction named SLOP (slowdown for low power) is automatically inserted by the processor control that also generates power-down, sleep mode, or clock gating signals for various hardware units. We have analyzed several technologies ranging through 180nm to 32nm and shown that the ISD method is equally or more effective than the CSD method in higher leakage technologies.

In general, the slowdown of a computing task can consume more energy. In fact, it would always turn out to be that way if we considered the raw energy consumption from an ideal source. The conclusions differ when we consider a real source, such as the battery in a portable device. A relevant parameter is the *lifetime* or the time between consecutive recharges of a battery. A battery's capacity, usually in mA-hours, is a valid indicator of the recharge time if the battery supplies close to the rated current. At higher currents, the capacity degrades. Thus, reduction in power consumption (or current drain) can enhance the lifetime [38]. We use a battery model based on the classical Peukert's law [21] to represent the battery lifetime, which is adjusted for the increased task execution time. Alternatively, a battery efficiency model [43] can also be used. Slowdown for power reduction is considered beneficial only if the adjusted lifetime is enhanced. This advantage of ISD becomes more pronounced as the technology becomes leakier.

Instruction slowdown (ISD) can be compared to another proposed power saving method called *fetch throttling* [33, 34]. This method, when applied to multiple issue processors, slows down the rate of instruction fetch based on the lack of any parallel execution opportunity in the program being executed. Thus, the instructions that would have waited in the pipeline due to data, resource, or control conflicts are fetched after suitable delays. The reported average reduction in energy delay product is 6.7% for static throttling and could go up to 15% with dynamic throttling. These savings are due to the avoidance of incorrect speculations. We can reduce the performance penalty of instruction slowdown (ISD) by inserting the NOPs after those instructions that require speculation. However, this aspect is not discussed in this work and should be explored in the future. The objective of the present work is to reduce power with minimal energy cost and to maximize number of operations performed in a single recharge.

Chapter 2

Theory and Background Work on Low Power Design

2.1 Introduction

2.1.1 Need for Low Power VLSI chips

Higher performance and lower chip area have always been major concerns for chip designers. Low power dissipation of VLSI chips has now become one of the primary goals. In the past, the device densities were low enough that power dissipation was not a constraining factor in chips. As the scale of integration improves, more transistors, faster and smaller than their predecessors, are being packed into a chip. This leads to steady growth of operating frequency and processing capacity per chip, resulting in increased power dissipation. New generation devices are at a safe distance from reaching their fundamental physical limits so the evolution seems to continue for a while. A need for low power VLSI chip arises from such *evolution forces* of integration circuits.

Another factor that fuels need for low power chips is the increased market demand for *Mobile Internet Devices (MIDs)* powered by batteries. The craving for smaller, lighter and more durable products directly translates to low power requirements. Batteries have not experienced a similar rapid density growth compared to electronic devices. The *specific weight* (stored energy per unit weight) of batteries barely doubles in several years [61] (Figure 2.1). Also, further increase in battery specific weight will create concerns about their safety as the energy density will approach that of explosive chemicals as shown in Figure 2.2. So the battery technology is not going to solve the power demand problem in future devices but the devices, on the other hand, will have to use battery energy in a smart way.

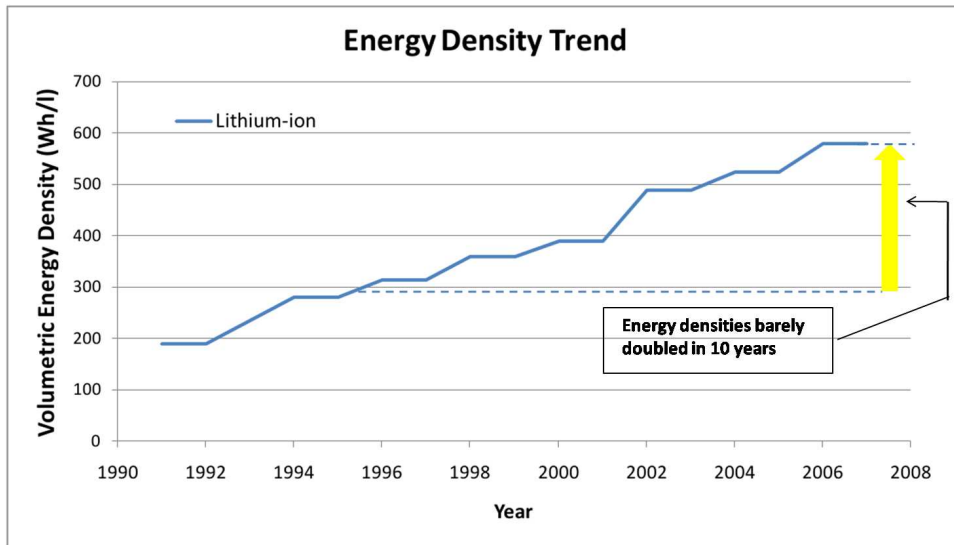


Figure 2.1: Growth in energy densities of Lithium-ion batteries

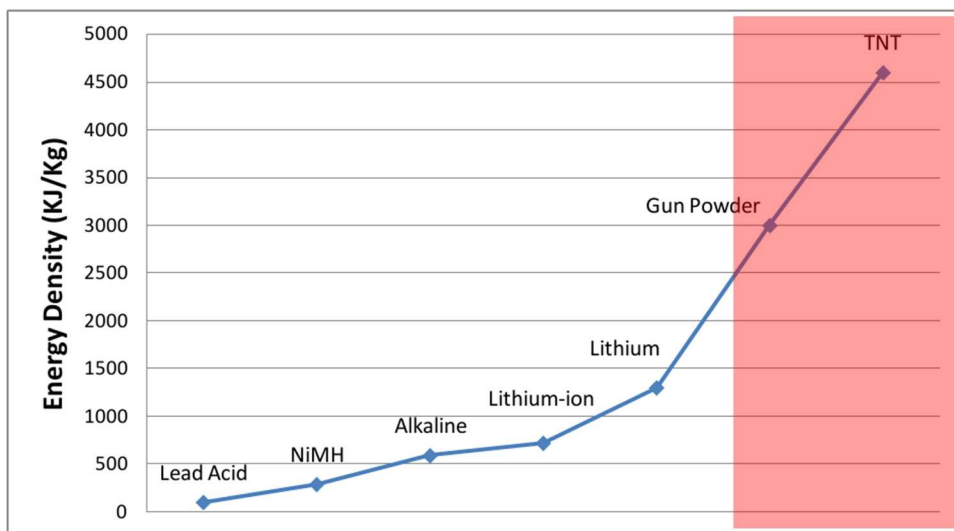


Figure 2.2: Limit on the growth of battery energy densities

Table 2.1: ITRS predictions on power dissipation of technology nodes

| Node | 90nm | 65nm | 45nm |
|--------------------------|------|------|------|
| Dynamic Power per cm^2 | 1X | 1.4X | 2X |
| Static Power per cm^2 | 1X | 2.5X | 6.5X |
| Total Power per cm^2 | 1X | 2X | 4X |

High performance computing systems characterized by large power dissipation also drive the low power needs. The power dissipation of a typical high performance microprocessor is about 150 watts with an average power density of 50-75 watts per square centimeter. Local hot spots on the die can be many times higher than the average number. This has a direct impact on packaging cost of chip and cooling cost of the system. A chip that operates at 3.3V consuming 10 watts of power means average current of 3A. Transient currents would be much higher than these. This creates problems in the design of power supply rails and poses a challenge in analysis of digital noise. This also poses a threat to reliability of the chip as mean time to failure decreases with increase in temperature. The problems are expected to get worse as we move to new technology nodes as predicted by International Technology Roadmap for Semiconductors(ITRS), shown in the Table 2.1.

Another driving force for demand of low power chips comes from the environmental concerns. Computers are the fastest growing electricity loads in the commercial sector. Since electricity generation is major source of air pollution, inefficient energy usage in computing equipment directly contributes to environmental pollution.

2.1.2 Power Vs. Energy

For MIDs operating on batteries, the distinction between power and energy is critical. While power is decided by the instantaneous current drawn by the device, energy is decided by the duration for which the current was drawn. The power drawn by a portable device such as cell phone or a Personal Digital Assistant (PDA) varies according to what type of tasks are being performed, e.g. an active call or a web browsing task will consume a considerable

amount of current while a standby mode will not consume as much power. In both the cases, however, energy is being drawn from the battery and in many practical circumstances the standby time of the device is large enough that it consumes equal amounts of energy. For a portable equipment operating on battery, therefore, better energy management and maximizing battery life are more logical design goals than power management.

2.2 Where Does All the Power Go?

All the power consumed by a CMOS device does not produce useful activity. Part of the power is dissipated in the ON resistance of the device while charging and discharging the output capacitance. This is known as Dynamic power dissipation. Dynamic power dissipation also consists of short circuit power dissipation which is caused by a short between VDD and ground due to a momentary ON state of both the P-type and N-type network in a device. Part of the power is also dissipated in the OFF resistance of the device due to flow of leakage current from supply to ground while the device is turned OFF. This is known as Static power dissipation. The following subsections describe each of them in detail.

2.2.1 Dynamic Power Dissipation

Until 65nm CMOS technology process, the dynamic power dissipation was the dominant source of power dissipation in CMOS. It is caused by the charging and discharging of the output node capacitance. Following is the formula used for calculation of dynamic power dissipation.

$$P_D = C_L V^2 f \quad (2.1)$$

Where,

C_L = Total load capacitance of the circuit. This capacitance largely consists of the parasitic capacitance inherent in the circuit such as, CMOS gate capacitances, source to drain capacitances and interconnect capacitances. Although these capacitances can not be avoided entirely, certain measures can attempt to minimize these capacitances which is one of the methods of reducing dynamic power dissipation.

V = Supply voltage of the circuit. This is one of the important factors in controlling the power consumption, as the power reduces quadratically with change in voltage. Supply voltage also affects static power consumption as we will see in next subsection.

f = Frequency of operation. Slower circuits consume less power as compared to faster ones.

As mentioned before, short circuit power consumption also contributes to dynamic power dissipation. Figure 2.3 shows an inverter circuit and the currents associated with its operation. The circuit operates at V_{dd} with V_i as input voltage, V_{tn} as threshold for NMOS and V_{tp} as threshold for PMOS. When the input V_i changes from low (0 V) to high (V_{dd}) there is a short time duration for which the input is greater than V_{tn} and less than V_{tp} as shown in figure 2.4. This causes both PMOS and NMOS to conduct and hence a short circuit current flows from V_{dd} to ground. The shape of short circuit current curve is dependent on

- The duration and slope of input signal.
- The I-V curves of P and N transistors which depend on their sizes, process technology, temperature, etc.
- The output load capacitance of the inverter.

2.2.2 Static Power Dissipation

Ideally, CMOS circuits dissipate no static power when they are not switching. But semiconductor devices conduct or leak through reverse biased channels and provide a path from V_{DD} to ground and this constitutes to leakage power consumption. Leakage current is a

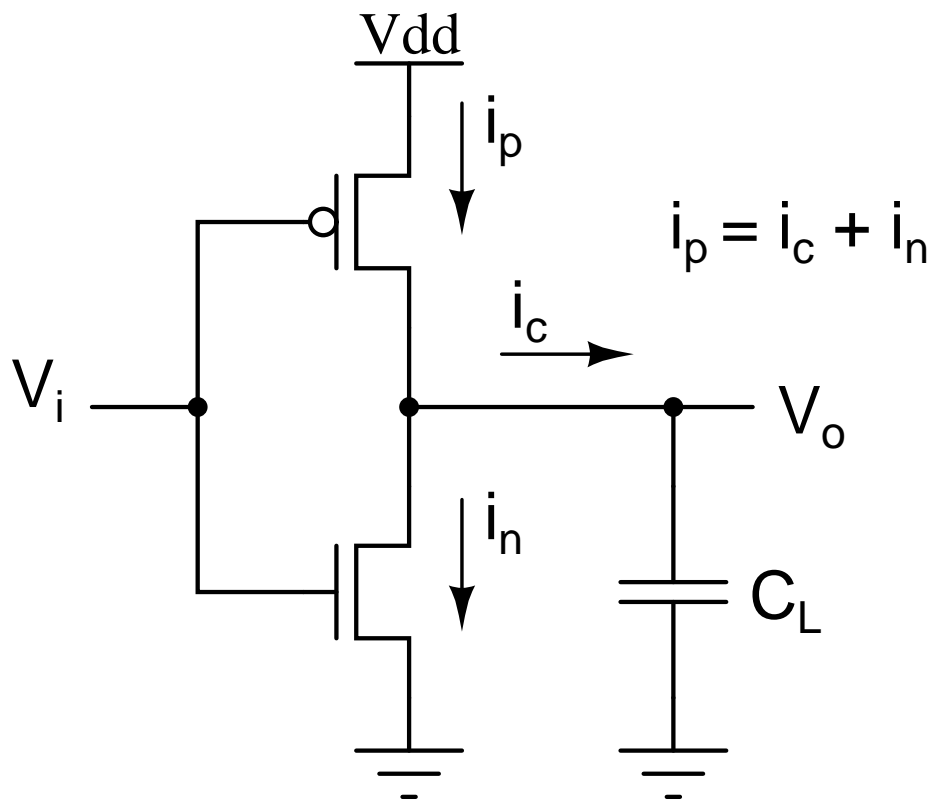


Figure 2.3: A CMOS Inverter circuit

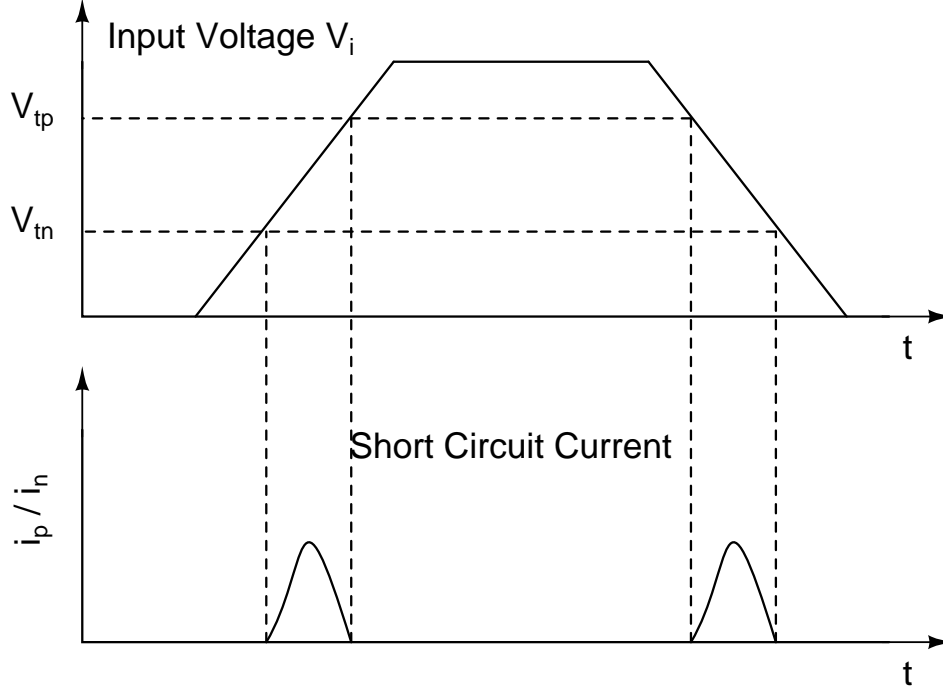


Figure 2.4: Short circuit currents of CMOS inverter during input transition

form of current which is generally not intended for normal operation of a digital circuit. This leakage current is not useful in most cases. There are various sources for leakage currents, as shown in Figure 2.5, and we will discuss three primary sources.

- Sub-threshold Channel conduction current (I_{sub})

In the OFF state, even though the transistor is logically turned off, there is a non-zero leakage current flowing through channel. This is known as sub-threshold leakage. Other than device dimensions and fabrication process, the magnitude of this current depends on threshold voltage, V_t ; gate voltage, V_{gs} ; drain voltage V_{ds} and temperature. During the OFF state, $V_{ds} \approx V_{DD}$ so the sub-threshold current essentially depends on V_{gs} . It is given by following equation [20].

$$I_{sub} = I_0 e^{(V_{gs} - V_t)/(\alpha V_{th})} \quad (2.2)$$

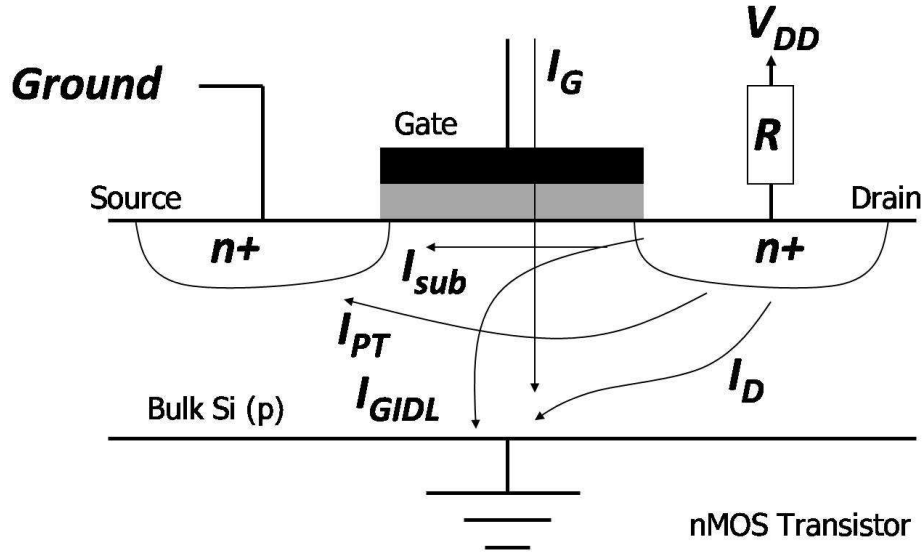


Figure 2.5: Leakage Currents for nMOS transistor

Where,

V_t is the device threshold voltage,

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

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

α ranges from 1.0 to 2.5 and is dependent on 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.

- Gate Tunnelling Current (I_G)

With scaling of the channel length, a good transistor aspect ratio can be maintained only by comparable scaling of oxide thickness, junction depth and depletion depth. Maintaining this aspect ratio is a challenge since the scaling in the vertical direction is difficult. The silicon dioxide gate dielectric thickness is approaching scaling limits and there is a rapid increase in the gate tunnelling current. The oxide thickness limit will

be reached approximately when the gate to channel tunnelling current (I_G) becomes equal to the off-state source to drain sub-threshold leakage (I_{sub}).

This limitation can be resolved by making use of different materials with high permittivity as gate dielectric. This will result in thicker and easier to fabricate dielectric with potential for significant reduction in leakage current.[19]. One such successful implementation is Hafnium based high-k dielectric in 45nm technology by Intel for their processor series code named ‘Penryn’. Hafnium silicate based dielectric materials help reduce leakage currents but they also suffer from trapped leakage currents which affects the device life.

- Reverse biased PN-Junction current (I_D)

This current flows when (for an nMOS transistor) the source is at VDD and the drain is at ground. The current flows due to a PN-junction formed at the source or drain of transistors due to parasitic effect of the bulk CMOS device structure. The junction current at the source of the transistor is picked up through bulk or a well contact. The magnitude of this current is given by following equation [20].

$$I_D = I_s(e^{V/V_{th}} - 1) \quad (2.3)$$

Where,

I_s is reverse saturation current,

V_{th} is thermal voltage which is given by $V_{th} = kT/q$ where $k = 1.38 \times 10^{-23}$ Joule/K is a Boltzmann’s constant, q is electronic charge in Coulombs and T is device operating temperature.

I_D is largely independent of operating voltage but depends, in general, on temperature, process, bias voltage and area of the PN-junction.

Other sources of leakage current such as Gate Induced Drain Leakage current (I_{GIDL}) and drain source Punch Through current (I_{PT}) also contribute to total leakage current.

2.2.3 The conflict between Dynamic 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. Voltages for semiconductor devices have been reduced from 5V to 0.8 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

$$I_{DS} = \mu C_{ox} \frac{W}{L} \cdot \frac{(V_{GS} - V_t)^2}{2} \quad (2.4)$$

Where,

μ is the carrier mobility,

C_{ox} is the gate capacitance,

V_t is the threshold voltage,

V_{GS} is the gate-source voltage

So to maintain higher I_{DS} we need to lower V_{th} as we lower VDD (or V_{GS}). However, lowering V_{th} results in an exponential increase in the sub-threshold leakage current as indicated by the I_{sub} equation (equation 2.2).

Thus the methods to lower dynamic power and leakage power in a device contradict each other. This situation has worsened for 65nm and lower CMOS process technologies as the static power is equal to or more than dynamic power in the device.

2.3 Low Power Design Methods

Low power methods for design of circuits can be classified in many different ways. One of the classic papers in this area [8] describes these techniques in three simple categories as

1. Trade area or speed for power,
2. Don't waste power and
3. Find a low power problem.

Though this functional classification is good for an insight into the subject, classification of these methods based on abstraction level is more practical from an engineer's point of view. System or architecture level techniques are most effective for managing power since often a problem can be implemented with an algorithm that consumes less power [22]. Algorithmic level changes in the solution to a problem can only be incorporated at the system or architectural level. On the other hand, estimation of power is most accurate at the transistor level and least accurate at the system level. A decision for the selection of abstraction is generally based on the overhead involved with the technique. This overhead may include area, speed, complexity and verification time. In any modern chip design flow, efforts to reduce power consumption in a circuit are incorporated at all possible stages and levels of abstractions as shown by Figure 2.6. The following subsections discuss some of the techniques of low power design at various levels of abstraction. We discuss these in a bottom up fashion.

2.3.1 Circuit Level Methods

At the circuit level, the power reduction techniques are quite limited in number and they generally don't result in more than 25% power reduction. However these techniques can have a major impact on power consumption of a design because these circuits, e.g. standard cells for most common gates and flip-flops, are repeated thousands of times on a chip. So circuit techniques with a small percentage of power savings cannot be overlooked.

Transistor sizing for Leakage Power reduction

Leakage current of a transistor increases with decrease in channel length and threshold voltage. But lower threshold voltage and channel length can provide higher saturation

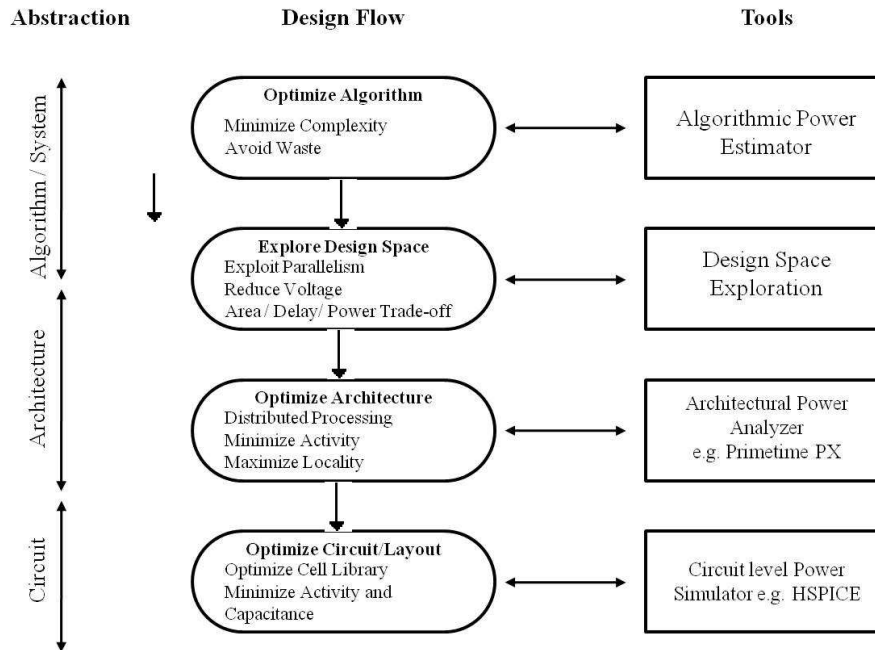


Figure 2.6: Design flow and type of tools at different levels of abstraction[22]

current resulting in faster switching frequencies. Thus there is a trade-off between leakage power and delay. One of the techniques used to reduce leakage power is to size one or more transistors in the transistor network.

Consider a simple two-transistor inverter. If the output of the inverter is logic high (P-transistor conducting) then the leakage power is determined by the N-transistor. Whereas in other case when the output is low, leakage power is determined by the P-transistor. Assuming that, in dormant mode, the inverter output is at logic high, we can reduce leakage power by increasing the channel length of the N-transistor. This also affects the switching speed of the N-transistor so the falling transition for the inverter will be affected. If the falling transition is not part of the critical path then this method can save on leakage energy without any change in the circuit speed. If the falling transition is on a critical path then we can select logic low to be the default output value during dormant mode and size the P-transistor instead. Similar effects can be observed by increasing the threshold voltage of either the P or N transistor.

Transistor Network Restructuring

Boolean functions are implemented as combinations of simple logic gates like NAND and NOR. These gates are then mapped to their equivalent transistor networks. These networks can be organized in different ways to achieve similar functionality. Choice of arrangement of transistors inside the network can be based on the leakage current minimization. Transistor stacking is a well known technique for reduction of leakage current in stand-by mode. Any implementation of a function has an input combination that results in minimum leakage current flow from VDD to ground. This input combination can be applied to the function when it is in stand-by mode. A very good summary of leakage reduction through stacking has been explained in Chapter 2 of [19].

Similarly, transistor re-organizing also plays an important role in reducing overall power consumption. Simple boolean functions can be implemented as a single complex network of transistors but as the function complexity increases the number of serial transistors in a network start to increase. This number has to be limited to ensure proper operation of the circuit. When the number of serial transistor increases the effective resistance of the serial transistor chain increases. To compensate for the increased resistance, transistor sizes have to be increased to maintain an acceptable delay. Also number of parallel transistors have a similar limit as well since each additional parallel transistor adds its own drain diffusion capacitance which increases total capacitance to the output node slowing down the circuit. These limits on the number of serial and parallel transistors are technology dependent and may also depend on operating voltage, system speed and other factors. Given an arbitrary boolean function, there can be different organizations of the circuits. Figure 2.7 indicates how a 4-input AND can be implemented in two different organizations.

Low Power Cell Libraries

Most digital designs today are designed using high level Hardware Description Languages(HDL) and synthesized using automated computer aided design (CAD) tools. The

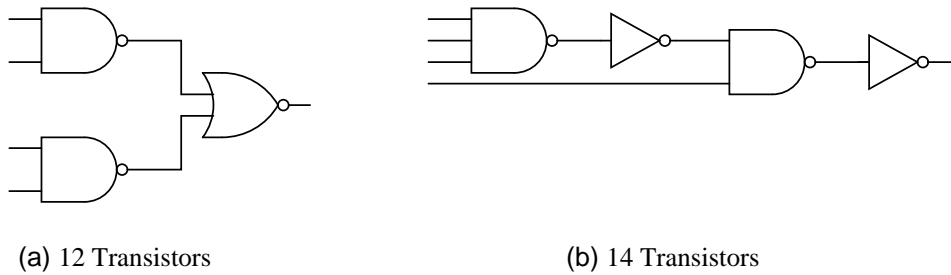


Figure 2.7: Two different implementations of a 4-input AND gate[22]

basic building blocks of these designs are customized logic gates or *Cells*. So the quality of overall design depends on the quality of these cells. Low power design is no exception to that.

The cells can be custom designed and characterized keeping power as primary constraint. The most important attribute that constitutes to a good low power design is the availability of variety of cell sizes of commonly used gates/functions. A smallest size of cell which satisfies the delay constraint can be chosen from all the available sizes. Therefore, fine granularity of the cell sizes is important. For instance, if the delay requirement demands for a cell size of 3X and the closest available size is 4X then we are unnecessarily wasting power by using an extra size cell. While deciding the range of cell sizes, the capacitance and area requirements are also important factors to be decided on. The overall circuit capacitance should be taken into consideration. Although it would be efficient for the design to have as many sizes of cells possible in the cell library, increase in the simulation and synthesis time of the design may limit the number of sizes per cell.

2.3.2 Gate Level Methods

Gate level design, or logic design in general is the most basic form of design where the logic synthesis starts. Due to the complexity of the designs today, the synthesis process is not done manually. Although the design process at logic level is done by HDLs, power optimization at logic level can still be performed by modifying the synthesis algorithms. The

most common theme in power optimization at logic level is reduction of switching activity. Switching activity directly contributes towards dynamic power and hence elimination of unnecessary switching activity should be a primary goal.

Gate Reorganization

Gate reorganization is a technique similar to transistor restructuring that was described in last section. In general, this reorganization is an operation to transform one logic circuit to another that is functionally equivalent. Since there are many possible combinations, it is important to choose an organization which does not differ drastically from the existing one in terms of area and delay while consuming lower power. A logic synthesis produces an initial logic network of gates from the HDLs. Then, depending on power constraints, some local transformations are applied to optimize the circuit. Some of the local transformations are:

1. Combine several gates into a single gate.
2. Decompose a single gate into several gates.
3. Duplicate a gate and redistribute its output connections.
4. Delete a wire.
5. Add a wire.
6. Eliminate unconnected gates.

This reorganization can be targeted towards low power design of functions. The transformation from a function to a gate structure is called as *Technology Mapping*. An excellent discussion of technology mapping for low power design has been carried out by Tiwari et. al.[30]

Signal Gating

Signal gating is a technique to mask unwanted switching activity from propagating forward causing unnecessary power dissipation. Since signal activities can be monitored

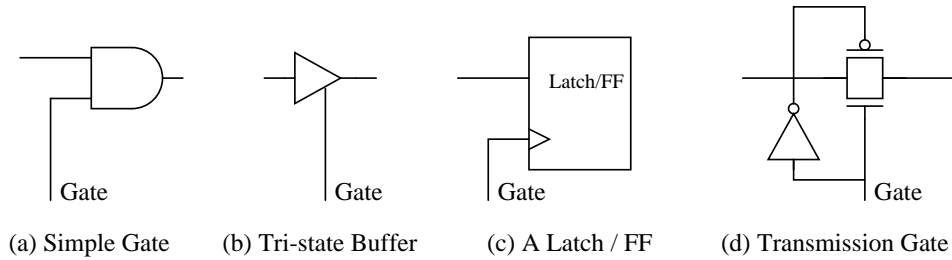


Figure 2.8: Various Implementations of Signal Gating [20]

and analysed better at gate level, these techniques are generally applied at gate level. There are many different methods to implement a signal gating. Figure 2.8 shows some of the implementations of signal gating

All signal gating methods require control signals to stop the propagation of switching activities. These control signals are generated by additional logic in the controller. This can add to area and cause additional leakage power. So a designer must take this fact into account and see if the design leads to overall power saving. The identification of signals to be gated is application dependent and is subjected to feasibility of implementation. Potential candidates for signal gating are clock signals, address buses and signals with high activity or glitches.

Logic and State Machine Encoding

Reduction in logic activity of the signals can also be achieved by changing the encoding of the combinational or sequential circuits. For instance, a 3-bit counter can be implemented in both binary and *Gray encoding*. In binary encoding the number of transitions for the counter is 14, whereas for a Gray encoding they are 8. For a 6-bit counter this difference is 126 for binary coding against 64 for Gray coding. So dynamic power can be greatly reduced by using Gray encoding.

Another example is *Bus Invert* encoding in which the signals transmitted over a parallel bus are examined and they are sent in normal form or in complemented form. This decision logic inspects two consecutive signal vectors for the activities and decides whether

to complement the next vector or not. A polarity signal is transmitted along with the vector so that the vector can be converted to its original form at the receiving end.

State machines perform transitions from one state to other depending on present state and input. To define this behaviour, a state transition graph/diagram is prepared first and then a synthesis tool will convert this graph (generally a HDL description) to combination of flip-flops and logic gates. Allocating the binary codes to the states in a state transition graph is called as state machine encoding. This encoding is one of the important factors that decides area, power and speed of the state machine. One goal is to reduce number of states in the machine so as to minimize number of flip-flops. Another key decision to make is which state encoding method to use. One hot or one cold methods have least number of transitions but they also use more number of flip flops. Binary encoding, on the other hand, has very few flip flops but may have many transitions if not properly designed. Gray encoding achieves a balance between number of flip flops and number of transitions for a state machine.

2.3.3 Architecture or System Level Methods

As we move up in abstraction level, the optimization problems become less exact and obscured due to more freedom in design configuration and decision. Due to this fact, higher level techniques rely more on human intuition and the art of chip design.

System Power Management

Low power Standby or Sleep modes: The system level power management ensures that we do not waste power by designing hardware that has more performance than necessary. Also, when the system throughput requirement is low, a low power oriented system should be able to adapt to the change and consume less power. Low power standby modes, or sleep modes, for a microprocessor are examples of such power management schemes. The best way to achieve a better power efficiency is to shut down functional units which are not being used or

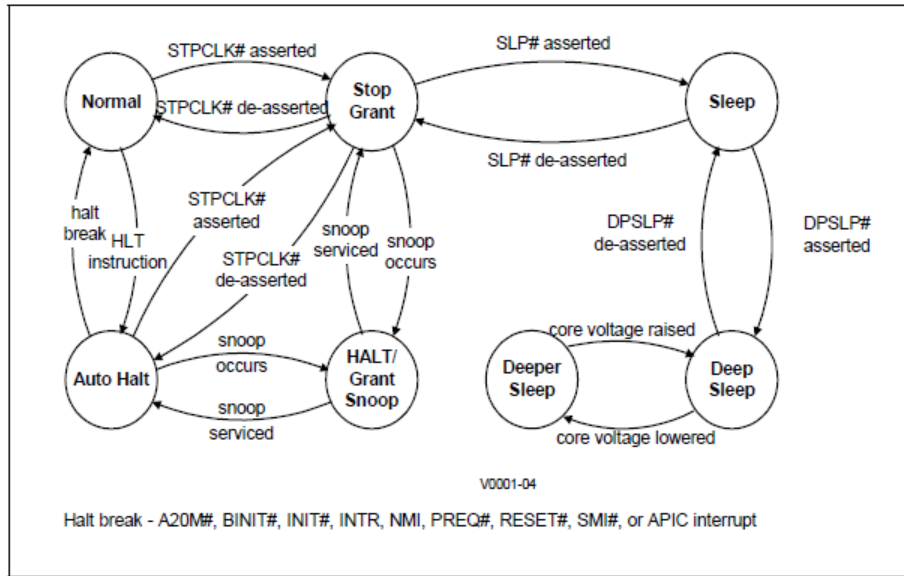


Figure 2.9: Different Sleep modes supported by Intel Pentium 4 Mobile [16]

gate the clock to these units in order to suppress the activity. In modern day microprocessor design, there is a variety of sleep modes available which can be activated depending on the state of the processor and performance requirements. These modes can be extremely effective, reducing standby power to a small fraction of the power consumed during normal operation. Figure 2.9 shows the state diagram of a processor for its transition from one mode of operation to other in order to achieve maximum power efficiency. If the processor clock frequency is reduced, functionality will be maintained during the low power mode and the processor can still service low priority tasks that do not require full frequency performance. Processor clocks can only be stopped if the machine state is maintained statically. Clocks can also be gated in only a portion of the design so that some functions are still active. Examples of a functions which need clocking even during a sleep mode are bus snooping controllers or control logic which actually provides sleep and clock gating signals.

Low power modes may be implemented with software or hardware control. Software control requires specific instructions to enter a sleep mode when the processor is idle. This code can be part of operating system (OS) code. The OS enters this state of sleep mode when system has been idle for pre-decided period of time. The system returns to normal mode of operation when a high priority interrupt is detected. To provide this support in an OS, certain provisions in the hardware are necessary. The power management unit on the chip needs to clock gate or power gate functional units depending on how deep a sleep mode has been requested by the OS.

Supply voltage selection for Standby mode: Reduction in supply voltage affects both dynamic and static power dissipation but it also increases delay, so the throughput is low. During a sleep mode, the system throughput requirements can be very low and there is a possibility of reduction in the voltage to a level which satisfies the throughput requirement. There is also a possibility of turning off power to a chip if all the memory states can be saved off chip and reloaded when the activity resumes. This approach can only be considered if the overhead (in time and power) of storing the data off chip and reloading it back justify the overall saving achieved by turning off the chip. Modern chips also have different voltage domains on a chip that can be turned off independently in order to achieve maximum efficiency.

Architectural Methods

Architectural methods are quite commonly used to develop microprocessors that are more power efficient and have equal or nearly equal performance as their power hungry counterparts. Architectural modifications can save power either with no compromise on speed and area or with trade-offs. These architectural decisions are made depending on the application for which the processor is being designed. A processor designed for a portable computer, PDA or a smart phone can use power saving techniques that trade performance off for power but may have strict area constraints. On the other hand, a processor designed

for a server can not use techniques that sacrifice speed for power. Support from compiler, operating system or an application are also important factors to be considered while making architectural modifications in order to reduce power. Following are some of the areas in architecture design where decisions and modifications for power efficiency can be made.

Instruction Set Architecture: Instruction fetch is performed for every single instruction. So a large portion of energy is spent in fetching operation. Instructions can be designed in a way that programs will have higher code density, smaller instruction lengths and reduction in code size. This will allow on-chip cache memories to hold the complete program which saves greatly on fetch energy. Decisions regarding choice of CISC or RISC type of ISA can also affect the energy efficiency. CISC has greater code density and smaller program length but CISC also needs complex decoding hardware. So CISC can be energy efficient if the ISA contains only few types of instructions. RISC on the other hand can use a simpler decoding logic but also has longer program lengths so it can be only energy efficient if wide variety of small instructions are required. Number of instructions accessing memory variables directly should be limited. Such instructions reduce code length but also need more energy due to their longer execution times. A proper combination of memory-to-memory accesses and register operations in an ISA can obtain maximum energy efficiency. A fixed reduced instruction length or a variable instruction length are other decisions that a designer needs to make while designing ISA.

Datapath: Pipelining is common way of implementing the ISA due to its inherent throughput advantage. Two important parameters to be considered while designing the pipeline are the number of pipeline stages and the number of execution pipelines. Two pipelining strategies which emphasize the two factors are described below. [22]

Superscalar

Performance: increased throughput by providing multiple execution units so that parallel

execution may be implemented.

Power: increase in design complexity and area, data dependency check requirements increase dispatch logic area.

Superpipelined

Performance: increased number of simple pipeline stages, perform faster and higher clock frequency can be achieved.

Power: increased number of clocked elements, inherent increase in dynamic power due to increased frequency.

Microprocessors chosen for low power typically have five pipeline stages or less. Use of register files can save a lot of energy by reducing traffic to memory. But register files themselves can also be made power efficient by power/clock gating them during pipeline stalls and disabling read ports when data is being provided from other sources.

Parallel Architecture with Voltage Reduction: Parallelism has traditionally been used to boost system throughput. It does so without increasing the operating frequency but requires additional hardware to perform multiple functions at the same time. In short, parallelism trades area for performance. This trade-off can also be used to reduce the power. Voltage scaling has a quadratic effect on dynamic power reduction and linearly reduces leakage power. So scaling down the voltage is an attractive solution for a power efficient design. But since circuit delay is inversely proportional to the voltage, reduction in voltage increases delay and hence there is a performance penalty. This problem can be overcome by using a parallel architecture which allows lowering the voltage while still maintaining the throughput. Consider a signal processing system whose throughput requirement is satisfied by a frequency f . Let V be the system voltage and C is the total amount of capacitance being switched, then the power consumption is given by

$$P = (CV^2f) \tag{2.5}$$

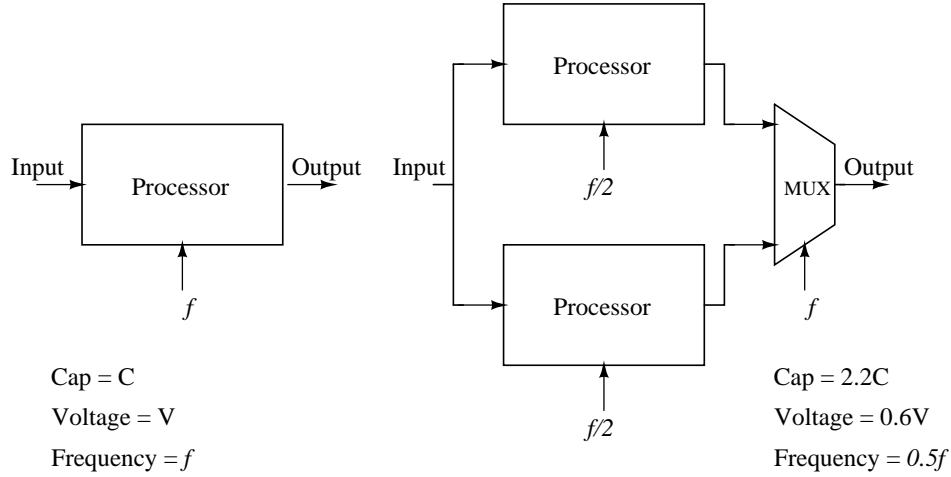


Figure 2.10: Power Dissipation of uniprocessing and parallel processing systems

If the number of processors is doubled as shown in figure 2.10, each of the processors can be operated at half the frequency $f/2$ and the output is multiplexed at the desired frequency f . Now assuming that due to increase in components the total capacitance switched is $2.2C$ and the voltage can be scaled down to $0.6V$, the new power dissipation is given by

$$P' = (2.2C)(0.6V)^2(0.5f) = 0.396P \quad (2.6)$$

So in the best case, we get about 60% power reduction compared to the single processor system. But there are other factors which limit this technique to achieve higher power reduction. One important factor is *leakage power*. Since we have additional components in the system the leakage current will be at least twice that of the single processor configuration. So according to formula $P_{leakage} = V \times I_{leakage}$ the leakage power is 1.2 times its original value. Another factor is the *availability of inputs in the parallelizable form*. When considering the system with two processors, we assumed that the input can be split into two equal length parts and that these parts are independent. But, in practice, only very few types of inputs such as images, certain matrix operations, etc., have such properties. Most other problems are sequential and have inter-dependability of variables on each other. This realization has

changed direction of new research towards making applications, programs and basic algorithms more parallelizable [60].

Dynamic Voltage and Frequency Scaling (DVFS) : The total power at each node of CMOS circuit can be represented by

$$P = (C_L V_{dd}^2 f) + (I_{SC} V_{dd}) + (I_{leakage} V_{dd}) \quad (2.7)$$

It is apparent from the above equation that each of the contributors to total power can be reduced by reducing the supply voltage V_{dd} . Also, the first term, which represents dynamic power, reduces quadratically with the voltage. Voltage reduction has been one of the most common techniques of power reduction. Low voltage modes are used in conjunction with lowered clock frequencies to minimize power consumption associated with components such as CPUs and DSPs; only when significant computational power is needed will the voltage and frequency be raised. Many modern chips also contain multi-voltage domains that can be operated on different voltages depending on their critical delay requirements and can also have multiple voltage assignments (including 0V) for each domain.

Dynamic frequency scaling (also known as CPU throttling) is a technique in computer architecture whereby the frequency of a microprocessor can be automatically adjusted at run time, either to conserve power or to reduce the amount of heat generated by the chip. Dynamic frequency scaling is commonly used in laptops and other mobile devices, where energy comes from a battery and thus is limited. It is also used in quiet computing settings and to decrease energy and cooling costs for lightly loaded machines. Less heat output, in turn, allows the system cooling fans to be throttled down or turned off, reducing noise levels and further decreasing power consumption. Dynamic frequency scaling reduces the number of instructions a processor can issue in a given amount of time, thus reducing performance. Hence, it is generally used when the performance requirements are not critical. Dynamic

frequency scaling by itself is rarely worthwhile as a way to conserve switching power. Saving the most power requires dynamic voltage scaling too, because of the V^2 component and the fact that modern CPUs are strongly optimized for low power idle states. In most constant-voltage cases it is more efficient to run briefly at peak speed and stay in a deep idle state for longer (called “race to idle”), than it is to run at a reduced clock rate for a long time and only stay briefly in a light idle state. However, reducing voltage along with clock rate can change those trade-offs.

Both dynamic voltage and frequency scaling (DVFS) can be used to prevent computer system overheating, that can result in program or operating system crashes, and possibly hardware damage. Some of the examples of DVFS implementation are Intel’s CPU throttling technology, SpeedStep, which is used in its mobile CPU processors and AMD’s two different CPU throttling technologies- Cool’n’Quiet, which is used on its desktop and server processor lines, and PowerNow, which is used in its mobile processor line.

2.4 Power Source Optimization: A System Approach

2.4.1 Choice of Metric

Traditional metrics like minimization of Power and Energy are not really suitable when power source (battery) optimization is a concern. For battery operated portable devices, an obvious objective is to maximize the battery lifetime. In spite of this fact, the discussions of low power design metric and methodologies have entirely focused on VLSI sub-system optimizations. The energy stored in a battery is assumed to be constant and available at any possible rate. In reality, however, the energy stored in a battery may not be used to its full extent. The delivery of energy from battery to system depends on the mean value of the current drawn from the battery. Battery lifetime does not have a simple linear relationship with power consumption of the circuit. e.g. a 2X increase in system power can cause a 3X decrease in battery lifetime. These facts motivate us to consider other metrics for design goal of power source optimization.

Weiser et al. [56] present Millions of Instructions Per Joule(MIPJ) as a quality metric for dynamic voltage scaling (DVS). The key idea is to eliminate idle time by reducing the processor voltage and clock for a given segment of computation. To predict processor utilization, either a fixed-size window of future events or a fixed-size window of past events is analyzed, and the corresponding DVS decisions are evaluated using trace-based simulations. This method has limited practicality since measurement and tracking of battery energy in terms of joules is difficult.

Rakhmatov et al. [44, 45] use an analytical model of the battery to minimize a cost function $\sigma(t)$. This cost is function of load current $i(t)$ and sum of $l(t)$ and $u(t)$, where, $l(t)$ is the charge lost in load and $u(t)$ is the charge unavailable. Evaluation of this cost function is in the context of DVS for task scheduling and battery optimization. Minimization of this cost function is subjected to constraints such as task dependencies, task deadlines etc.

Pedram et al. [43] propose battery discharge-delay product as the metric. This metric is similar to the energy-delay product while accounting for the battery characteristics and the DC/DC conversion efficiency. The BD-delay product states that the design goal should be to minimize delay and maximize battery lifetime at the same time.

2.4.2 Classification of Power Source Optimization Methods

Since the primary aim is to optimize the energy of power source, the methods normally used for low power design are only a part of power source optimization methods. Various methods have already been proposed [56, 43, 38, 46] and these can, in general, be classified in three following categories.

Voltage Management Methods

Most common of voltage management methods is dynamic voltage management. Here the system has a capability of statically or dynamically varying VDD. A relevant problem is to find an optimum value of supply voltage which would minimize the energy consumption

of the battery and still maintain the throughput requirements. [43] propose a method to find optimum operating voltage for minimization of battery discharge-delay product. First of our two proposed techniques falls into this category. In chapter 5, we discuss this technique in detail.

Throughput Management Methods

Dynamic frequency scaling is one of the most used methods in this category. CPU frequency scaling for battery powered computers is examined in [48] in terms of its impact on battery life, system performance, and power consumption. Frequency scaling approaches use information from a battery model to vary the clock frequency of system components dynamically at run time. They also use workload characteristics such as run-time and idle-time percentages dynamically, and models of system power and performance. These approaches can be used to ensure efficient use of the battery without significantly compromising system performance. [46]

Functional Management Methods

These methods include most of the methods discussed in the chapter 2 above. Most of these methods focus on power management of the system in order to reduce the average current drawn from the battery. Battery aware dynamic task scheduling is one such technique [45]. Second of our two proposed methods, which exploits idleness in a pipeline processor to dynamically manage power to different units, falls under functional management category.

Dynamic voltage and frequency scaling (DVFS) is combination of voltage and throughput management methods and architecture level parallelism is a combination of all the three methods mentioned above.

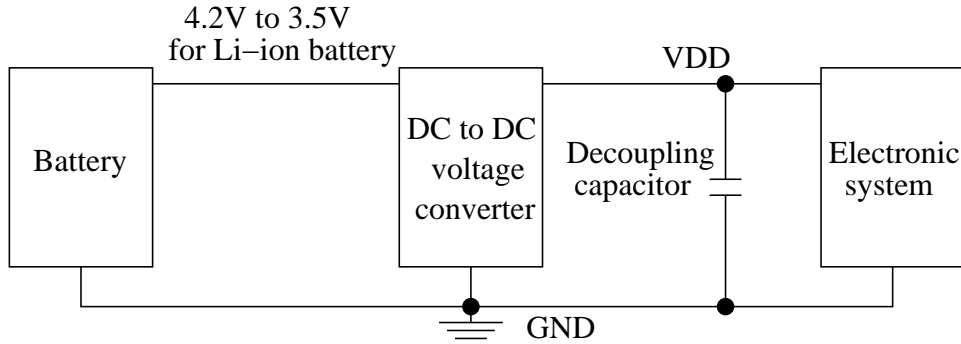


Figure 2.11: Powering and Electronic System

2.4.3 A Typical Battery Powered Electronic System

A typical power supply for an electronic system is shown in Figure 2.11. The primary source of energy is a battery, normally an electrochemical device [21]. The battery can be a primary type that is discarded after it is discharged, or a rechargeable type. As shown in Figure 2.11, a fully charged Lithium-ion battery supplies 4.2 volts and when the voltage drops below 3.0 volts it is recharged. The electronic system is supplied a voltage VDD that is close to 1 volt or lower for modern nanometer technologies. A DC-to-DC converter [55, 43] provides the voltage transformation as well as the capability to vary VDD for power management. Because the current requirement of the electronic system is often pulsed and time varying, decoupling capacitors are used to smooth the transient ripples. The decoupling capacitors is, in general, distributed in the power grid of the system.

In the consequent chapters, we discuss these components of a system in detail. Chapter 3 describes Lithium-ion batteries in detail along with background, electro-chemistry and terminology used for lithium ion batteries. This chapter also discussed various models that have been proposed and the model used for this work. Chapter 4 summarizes theory and background work on DC-to-DC converters. Chapter 5 describes the proposed technique for power source optimization. This technique falls into the first class of methods i.e. voltage

management. Chapter 6 describes a proposed functional method of power source optimization where we demonstrate savings in battery lifetime. Chapter 7 makes concluding remarks on the methods.

Chapter 3

Lithium-ion Battery Background and Modelling

3.1 Background

For many years, nickel-cadmium had been the only suitable battery for portable equipment from wireless communications to mobile computing. Nickel-metal-hydride(NiMH) and lithium-ion emerged in the early 1990s and today, lithium-ion is the fastest growing and most promising battery chemistry.

Lithium is the *lightest* of all metals, has the greatest electrochemical potential and provides the largest energy density per weight. Attempts to develop rechargeable lithium batteries failed due to safety problems. Because of the inherent instability of lithium metal, especially during charging, research shifted to a non-metallic lithium battery using lithium ions. Although slightly lower in energy density than lithium metal, lithium-ion is safe, provided certain precautions are met when charging and discharging. In 1991, the Sony Corporation commercialized the first lithium-ion battery. Other manufacturers like Hitachi, Panasonic, and LG followed suit.

The *energy density* of lithium-ion is typically twice that of the standard nickel-cadmium. There is potential for higher energy densities for lithium-ion batteries. The load characteristics are reasonably good and behave similarly to nickel-cadmium in terms of discharge. The high cell voltage of 3.6 volts allows battery pack designs with only one cell. Most of today's mobile phones run on a single cell. A nickel-based pack would require three 1.2-volt cells connected in series.

Lithium-ion is a *low maintenance* battery. There is no memory and no scheduled cycling is required to prolong the battery's life. In addition, the self-discharge is less than half

compared to nickel-cadmium, making lithium-ion well suited for modern portable computing applications. Lithium-ion cells cause little harm when disposed.

Despite its overall advantages, lithium-ion has its drawbacks. It is *fragile and requires a protection circuit* to maintain safe operation. Built into each pack, the protection circuit limits the peak voltage of each cell during charge and prevents the cell voltage from dropping too low on discharge. In addition, the cell temperature is monitored to prevent temperature extremes. The maximum charge and discharge current on most packs are limited to between 1C and 3C. With these precautions in place, the possibility of metallic lithium plating occurring due to overcharge is virtually eliminated.

Ageing is a concern with most lithium-ion batteries. Some capacity deterioration is noticeable after one year, whether the battery is in use or not. The battery frequently fails after two or three years. It should be noted that other chemistries also have age-related degenerative effects. This is especially true for nickel-metal-hydride if exposed to high ambient temperatures. Storage in a cool place slows the ageing process of lithium-ion (and other chemistries). Manufacturers recommend storage temperatures of $15^{\circ}C$ ($59^{\circ}F$). In addition, the battery should be partially charged during storage. The manufacturer recommends a 40% charge.

The most economical lithium-ion battery in terms of cost-to-energy ratio is the cylindrical 18650 (18 is the diameter and 650 the length in mm). This cell is used for mobile computing and other applications that do not demand ultra-thin geometry. If a slim pack is required, the prismatic lithium-ion cell is the best choice. These cells come at a higher cost in terms of stored energy.

Advantages of lithium-ion batteries

- High energy density - potential for yet higher capacities.
- Does not need prolonged priming when new. One regular charge is all that's needed.

- Relatively low self-discharge - self-discharge is less than half that of nickel-based batteries.
- Low Maintenance - no periodic discharge is needed; there is no memory.
- Speciality cells can provide very high current to applications such as power tools.

Limitations of lithium-ion batteries

- Requires protection circuit to maintain voltage and current within safe limits.
- Subject to ageing, even if not in use - storage in a cool place at 40% charge reduces the ageing effect.
- Transportation restrictions - shipment of larger quantities may be subject to regulatory control. This restriction does not apply to personal carry-on batteries.
- Expensive to manufacture - about 40 percent higher in cost than nickel-cadmium.
- Expensive to manufacture - about 40 percent higher in cost than nickel-cadmium.
- Not fully mature - metals and chemicals are changing on a continuing basis.

The Lithium Polymer battery

The *lithium-polymer* battery differentiates itself from conventional battery systems in the type of electrolyte used. The original design, dating back to the 1970s, uses a dry solid polymer electrolyte. This electrolyte resembles a plastic-like film that does not conduct electricity but allows ions exchange (electrically charged atoms or groups of atoms). The polymer electrolyte replaces the traditional porous separator, which is soaked with electrolyte.

The dry polymer design offers simplifications with respect to fabrication, ruggedness, safety and thin-profile geometry. With a cell thickness as little as one millimeter (0.039 inches), equipment designers are left to their own imagination in terms of form, shape and size.

Unfortunately, the dry lithium-polymer suffers from poor conductivity. The internal resistance is too high and cannot deliver the current bursts needed to power modern communication devices and spin up the hard drives of mobile computing equipment. Heating the cell to $60^{\circ}C$ ($140^{\circ}F$) and higher increases the conductivity, a requirement that is unsuitable for portable applications.

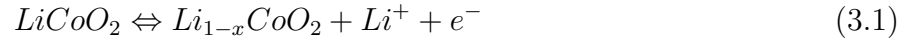
To compromise, some gelled electrolyte has been added. The commercial cells use a separator, or electrolyte membrane, prepared from the same traditional porous polyethylene or polypropylene separator filled with a polymer, which gels upon filling with the liquid electrolyte. Thus the commercial lithium-ion polymer cells are very similar in chemistry and materials to their liquid electrolyte counter parts.

Lithium-ion-polymer has not caught on as quickly as some analysts had expected. Its superiority to other systems and low manufacturing costs has not been realized. No improvements in capacity gains are achieved - in fact, the capacity is slightly less than that of the standard lithium-ion battery. Lithium-ion-polymer finds its market niche in wafer-thin geometries, such as batteries for credit cards and other such applications.

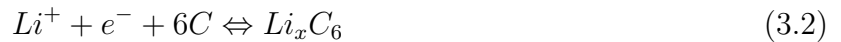
3.2 Electro-chemistry

The three participants in the electrochemical reactions in a lithium-ion battery are the anode, cathode, and electrolyte. Both the anode and cathode are materials into which, and from which, lithium can migrate. The process of lithium moving into the anode or cathode is referred to as insertion (or intercalation), and the reverse process, in which lithium moves out of the anode or cathode is referred to as extraction (or de-intercalation). When a lithium-based cell is discharging, the lithium is extracted from the anode and inserted into the cathode. When the cell is charging, the reverse process occurs: lithium is extracted from the cathode and inserted into the anode. During discharge, the anode of a conventional Li-ion cell is made from carbon, the cathode is a metal oxide, and the electrolyte is a lithium salt in an organic solvent.

Useful work can only be extracted if electrons flow through a (closed) external circuit. The following equations are written in units of moles, making it possible to use the coefficient x . The cathode half-reaction (with charging being forward) is:



The anode half reaction is:



Overcharge up to 5.2V leads to the synthesis of cobalt(IV) oxide, as evidenced by x-ray diffraction



The overall reaction has its limits. Over discharge will supersaturate lithium cobalt oxide, leading to the production of lithium oxide, possibly by the following irreversible reaction:



In a lithium-ion battery the lithium ions are transported to and from the cathode or anode, with the transition metal, Co, in Li_xCoO_2 being oxidized from Co_3^+ to Co_4^+ during charging, and reduced from Co_4^+ to Co_3^+ during discharge.

3.3 Description of Terminology

3.3.1 Capacity

Capacity of the battery is its ability to hold and supply charge. For practical purposes, this capacity is defined in units of Ampere Hour(Ahr). So a 1 Ahr battery is able to provide current of 1A for an hour. The capacity for modelling purposes can be categorized in different types. Full charge capacity is the remaining capacity of a fully charged battery at the beginning of a discharge cycle, and full design capacity is the remaining capacity of a newly manufactured battery. Further, theoretical capacity is the maximum amount of charge that can be extracted from a battery based on the amount of active material it contains, standard capacity is the amount of charge that can be extracted from a battery when discharged under standard load and temperature conditions, and actual capacity is the amount of charge a battery delivers under given load and temperature conditions.

3.3.2 Rate Dependent Capacity

Battery capacity decreases as the discharge rate increases. In a fully charged cell, the electrode surface contains the maximum concentration of active ions. When the cell is connected to a load, a current flows through the external circuit; active ions are consumed at the electrode surface and replenished by diffusion from the bulk of the electrolyte. However, this diffusion process cannot keep up with the reaction process, and a concentration gradient builds up across the electrolyte. A higher load current results in a higher concentration gradient and thus a lower concentration of active ions at the electrode surface. When this concentration falls below a certain threshold, which corresponds to the voltage cut-off, the electrochemical reaction can no longer be sustained at the electrode surface. At this point, the charge that was unavailable at the electrode surface due to the gradient remains unusable and is responsible for the reduction in capacity.

However, the unused charge is not physically lost, but simply unavailable due to the lag between reaction and diffusion rates. Decreasing the discharge rate effectively reduces this lag as well as the concentration gradient. If the battery load goes to zero, the concentration gradient flattens out after a sufficiently long time, reaching equilibrium again. The concentration of active ions near the electrode surface following this rest period makes some unused charge available for extraction. This charge can be used for recovery to control the discharge rate to maximize battery lifetime under performance constraints. However, at sufficiently low discharge rates, the battery will behave like an ideal energy source.

3.3.3 Temperature Effect

Temperature strongly affects battery capacity and its shelf life. Temperatures much lower than room temperature lowers the internal activity of the battery resulting in higher internal resistance and hence increasing slope of discharge curve. On the other hand, temperatures much above room temperature causes less internal resistance and hence the battery can deliver full rate of discharge and voltage. However, this results in a quicker self-discharge and the battery has less capacity to start with. Temperature effects on battery in a device are rather difficult to manage.

3.3.4 Capacity Fading

Because of their high energy density and capacity, lithium-ion batteries are the popular choice for many portable applications. However, these batteries lose a portion of their capacity with each discharge-charge cycle. This capacity fading results from unwanted side reactions including electrolyte decomposition, active material dissolution, and passive film formation. These irreversible reactions increase cell internal resistance, ultimately causing battery failure. To deal with this problem, system users can attempt to control the depth of discharge before recharging. Typically, a battery subjected to shallow discharge state, that

is, voltage is still relatively high when recharging occurs, will be good for more cycles than a battery subjected to deep discharge state for example, until the cut-off voltage is reached.

3.4 Modelling

Battery modelling, a mathematical description of batteries, is an important part of battery design and battery related system design. Several types of battery models have been reported in the literature. Use of any particular model is decided by its suitability in the application. For instance, a physical model may be suitable to construct a battery whereas an abstract or analytical model is suitable for designing a system containing batteries and optimization of battery parameters for the system. The following subsections briefly describe different types of models [38].

3.4.1 Physical Models

Physical models are the most accurate and have great utility for battery designers as a tool to optimize battery's physical parameters. However, they are also the slowest to produce predictions and the hardest to configure. These models may need as many as 50 parameters such as structure, chemical composition, temperature etc. for their configuration. They also provide a very limited analytical insight for system designers. Doyle et al [39, 40] developed an isothermal electrochemical model which describes charging and discharging cycles of a lithium ion polymer battery for one cycle. The model uses concentrated solution theory to derive set of differential equations which when solved can provide battery voltage as function of time. Dualfoil [41] is a Fortran program written to model lifetime of the battery. The program reads a sequence of constant current steps and compares the output voltage to cut-off voltage. This program has been widely used by many researchers for lifetime computation.

3.4.2 Empirical Models

Empirical models are the easiest to configure, and they quickly produce predictions, but they generally are the least accurate. Although they work well in certain special cases, the constants used have no physical significance, which seriously limits their analytical insight. Peukert's law [38] attempts to capture non-ideal discharge behavior using relatively simple equations. While an ideal battery with capacity C , discharged at a constant current I would be expected to have a lifetime L given by $C = LI$, Peukert's law expresses this as a power law relationship, $C = LI^k$. The exponent provides a simple way to account for rate dependence. However, the values for different temperatures must be obtained empirically, and the fit is not always accurate. Though easy to configure and use, Peukert's law does not account for time-varying loads. Most batteries in portable devices experience widely varying loads, for example, an iPhone user may run a movie player application followed by a text editor, which yields a profile with two very different loads for the battery.

Massoud Pedram and Qing Wu [43] model battery efficiency, the ratio of actual capacity to theoretical capacity, as a linear quadratic function of the load current. They derive bounds on the actual power consumed for different current distributions with the same average current and show that these bounds depend on maximum and minimum values of the current. Among all distributions with the same mean, a constant current (least variance) would give the longest battery lifetime, and a uniformly distributed current (highest variance) would give the shortest. This model accounts for rate dependence and can handle variable loads. Researchers have used it, with slight modifications, to maximize the lifetime of multi-battery systems, to minimize the discharge delay product in an interleaved dual-battery system design and in static task scheduling for real-time embedded systems.

3.4.3 Abstract Models

Instead of modelling discharge behaviour either by describing the electrochemical processes in the cell or by empirical approximation, abstract models attempt to provide an

equivalent representation of a battery. Although the number of parameters is not large, such models employ lookup tables that require considerable effort to configure. In addition, despite acceptable accuracy and computational complexity, these models have limited utility for design exploration because they lack analytical expressions for many variables of interest. Electrical-circuit and discrete-time models are particularly useful when compatible models of other system components, circuit models or VHSIC Hardware Description Language (VHDL) models, are available to simulate the entire system in a single continuous-time or discrete-time environment.

Gold [53] proposed a PSpice model which uses linear passive elements along with voltage sources and lookup tables to model the battery behaviour. This model can represent capacity fading, effect of temperature on internal resistance. It is a continuous time model. Benini [54] proposed a discrete time model which makes use of high level hardware description languages such as VHDL. Besides modelling basic parameters, the advantage of using this model is its compatibility with system level power management designs. Some of the other models include Hageman's PSpice model [52] for NiMH batteries, Bergveld's electrical circuit model [50] for NiCd batteries and more recently Chen's accurate electric model [49] for run time lifetime prediction which we use for this work and we will be discussing it in next section.

3.4.4 Analytical/Mixed Models

Some mixed models based on mathematical analysis have also been proposed. They use results obtained from a series of experiments to create system level models.

[44] proposes one such model, which describes a battery using two variables, derived from the lifetime values for a series of constant load tests. The parameter is a measure of the battery's theoretical capacity, which models the rate at which the active charge carriers are replenished at the electrode surface. Accuracy of battery lifetime predictions with this model has been verified with the Dualfoil model.

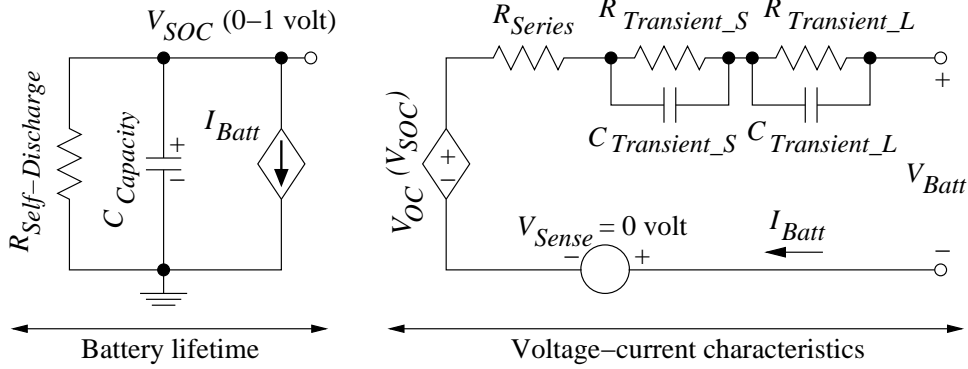


Figure 3.1: An Electrical Model for Lithium-ion battery

Peng Rong and Pedram [47] proposed a high level battery model to estimate remaining capacity that considers both the temperature effect and capacity fading with successive cycles. They derived an expression for cell terminal voltage as a function of time and, using the Arrhenius dependence on temperature of cell kinetics and transport phenomena, obtained an expression for the bulk properties of the active material as a function of the temperature. They also derived an expression for film thickness as a function of the temperature, discharge rate, and number of cycles.

3.5 Model Used for This Work

As mentioned before, we use an electrical model provided by [49]. This model is shown in Figure 3.1. One of the reasons behind choosing this model is its capability of predicting lifetime and I-V performance. Besides load current, it considers effects of temperature, number of cycles and storage time dependence of capacity on battery lifetime. This model is also scalable as it models batteries of varying AHr ratings and predicts runtime for different load current profiles. This model can be used for Lithium-ion, polymer Lithium-ion and NiMH batteries.

3.5.1 Description

On the left side of figure 3.1, a capacitor $C_{Capacity}$ represents the present state of charge(SOC) of the battery and a current source I_{Batt} models the discharge. The right side of the circuit models the voltage and current characteristics of the battery based on the current drawn from the battery. These two parts are connected to each other by a voltage controlled voltage source V_{SOC} whose value depends on the open circuit voltage(V_{OC}) of the capacitor $C_{Capacity}$

Assuming a battery is discharged from an equally charged state to the same end-of-discharge voltage, the extracted energy, called usable capacity, declines as cycle number, discharge current, and/or storage time (self-discharge) increases, and/or as temperature decreases[49]. The usable capacity can be modelled by a full-capacity capacitor ($C_{Capacity}$), a self-discharge resistor ($R_{Self-Discharge}$), and an equivalent series resistor (the sum of R_{Series} , $R_{Transient_S}$, and $R_{Transient_L}$). The full-capacity capacitor $C_{Capacity}$ represents the whole charge stored in the battery, i.e., SOC, by converting nominal battery capacity in Ahr to charge in coulomb and its value is defined as

$$C_{Capacity} = 3600 \times Capacity \times f_1(Cycles) \times f_2(Temp) \quad (3.5)$$

Where,

$Capacity$ is the nominal capacity in AHr,

$f_1 (Cycle)$ is a correction factor for number of cycles,

$f_2 (Temp)$ is a temperature-dependent correction factor

A fully charged battery can be initialised by setting the initial voltage across $C_{Capacity}$ (V_{SOC}) equal to 1 V or fully discharged by setting V_{SOC} to 0 V. In other words, V_{SOC} represents the SOC of the battery quantitatively and $0 \leq V_{SOC} \leq 1$.

As seen from equation 3.5, $C_{Capacity}$ will not change with current variation, which is reasonable for the battery's full capacity because energy is conserved. The variation of current-dependent usable capacity comes from different SOC values at the end of discharge for different currents owing to different voltage drops across internal resistor (the sum of R_{Series} , $R_{Transient_S}$, and $R_{Transient_L}$) and the same end of discharge voltage. When the battery is being charged or discharged, current-controlled current source I_{Batt} is used to charge or discharge $C_{Capacity}$ so that the SOC, represented by V_{SOC} , will change dynamically. Therefore, the battery runtime is obtained when battery voltage reaches the end-of-discharge voltage.

Self-discharge resistor $R_{Self-Discharge}$ is used to characterize the self-discharge energy loss when batteries are stored for a long time. Theoretically, $R_{Self-Discharge}$ is a function of SOC, temperature, and, frequently, cycle number. Practically, it can be simplified as a large resistor, or even ignored, which shows that usable capacity decreases slowly with time when no load is connected to the battery. In our implementation of the model we set its value to a very large resistance of about 1 GigaOhm.

Open-circuit voltage (V_{OC}) is changed to different capacity levels, i.e., SOC. The non-linear relation between the open-circuit voltage (V_{OC}) and SOC is important to be included in the model. Thus, voltage-controlled voltage source $V_{OC}(V_{SOC})$ is used to represent this relation.

In a step load current event, the battery voltage responds slowly. Its response curve usually includes instantaneous and curve-dependant voltage drops. Therefore, the transient response is characterized by the shaded RC network in figure 3.1. The electrical network consists of series resistor R_{Series} and two RC parallel networks composed of $R_{Transient_S}$, $C_{Transient_S}$, $R_{Transient_L}$, and $C_{Transient_L}$. Series resistor R_{Series} is responsible for the instantaneous voltage drop of the step response. $R_{Transient_S}$, $C_{Transient_S}$, $R_{Transient_L}$, and $C_{Transient_L}$ are responsible for short- and long-time constants of the step response. Theoretically, all the parameters in the proposed model are multi-variable functions of SOC, current, temperature, and cycle number.

3.5.2 Battery Lifetime

The state of charge (SOC) is defined as 1.0 for a fully charged battery. It is represented by a voltage V_{SOC} , which ranges between 0 and 1 volt. The charge of the battery is stored in a capacitor $C_{Capacity}$ whose value is determined as follows.

$$C_{Capacity} = 3600 \times Capacity \times f1(cycles) \times f2(Temperature) \quad (3.6)$$

Where, Capacity is the AHr rating of the battery.

Thus, 1 AHr \times 3600 seconds is the total amount of charge in coulombs. As the battery goes through cycles of charging and discharging its capacity to hold charge is affected, reducing the usable capacity. That is represented by $f1(Cycles)$. Similarly, temperature affects the usable capacity and that is represented by $f2(Temp)$. For simplicity, we have assumed both factors to be unity in the present discussion. The resistance $R_{SelfDischarge}$ represents leakage when the battery is stored over a long period. For reasonable time between recharge, this can be considered to be large or practically infinite. The current source I_{Batt} represents a source when the battery is being charged or a load when the battery is powering a circuit. In the latter case, it is the current being supplied to the DC-to-DC converter and to the circuit after conversion. When the model is used to simulate the behavior of a battery that is fully charged, V_{SOC} is initialized to 1 volt.

3.5.3 Voltage and Current Characteristics

The circuit on the right in figure 3.1 emulates the terminal voltage of the battery as it supplies current. This part is linked to the part on the left by state of charge (SOC), a quantity in the (0.0, 1.0) range. $V_{OC}(SOC)$ is the open circuit voltage. For Lithium-ion batteries, Chen and Rincon-Mora [49] empirically derive expressions for the circuit components, which all depend on SOC.

$$V_{OC}(SOC) = -1.031e^{-35 \times SOC} + 3.685 + 0.2156 \times SOC - 0.1178 \times SOC^2 + 0.3201 \times SOC^3 \quad (3.7)$$

$$R_{Series}(SOC) = 0.1562e^{-24.37 \times SOC} + 0.07446 \quad (3.8)$$

$$R_{Transient_S}(SOC) = 0.3208e^{-29.14 \times SOC} + 0.04669 \quad (3.9)$$

$$C_{Transient_S}(SOC) = -752.9e^{-13.51 \times SOC} + 703.6 \quad (3.10)$$

$$R_{Transient_L}(SOC) = 6.6038e^{-155.2 \times SOC} + 0.04984 \quad (3.11)$$

$$C_{Transient_L}(SOC) = -6056e^{-27.12 \times SOC} + 4475 \quad (3.12)$$

3.6 Summary

Many unique properties of Lithium ion batteries, including high energy density and quick recharging, have made Lithium ion batteries a popular choice of power source for portable computing devices. Since the focus of computing community has shifted to mobile, multi-function, wireless communication devices, study of batteries and research in power source optimization techniques has become an important part of product design. Various proposed battery models help designers understand impact of design decisions on battery energy and create better designs without having to set up time consuming experiments.

Chapter 4

DC to DC Converter

4.1 Necessity

There are various reasons to convert a DC voltage of one magnitude to another. Firstly, most of the commercially used lithium ion batteries have rated voltages in the range of 3.7 V to 4.2 V. But modern VLSI chips run at much smaller voltages of 1 V to 1.5 V. Secondly, battery operated portable systems have several different chips working together to provide the varied functionality. These are analog, digital and mixed signal chips and they may operate on different supply voltages. A single chip may also contain multiple voltage domains. Third, as a fully recharged battery is being used, the battery voltage drops as the stored charge from the battery drains. So regulation of output voltage is required in order to maintain a steady supply to the chips. DC to DC converters are switching regulators in general. Switching regulators are more efficient than linear regulators. Linear regulators are cheap and have simple structures but they can only convert from high voltage level to low voltage level. The excess voltage appears across a resistor and produces heating in the resistor. This heat has to be dissipated. So switching regulators are useful only for conversions with low output current ratings and low difference in the voltage levels as this limits the power dissipation. Switching regulators on the other hand are highly efficient. Typical range of efficiency being 75% to 98%. This efficiency of conversion is necessary to make efficient use of limited battery energy. Switching regulators store the input energy temporarily in magnetic (inductor) or electric (capacitor) storage elements in one phase of operation and release it to the output in the next phase at a different voltage. They can convert voltages from low to high and high to low levels. They can also be designed to produce negative voltages. The drawbacks of

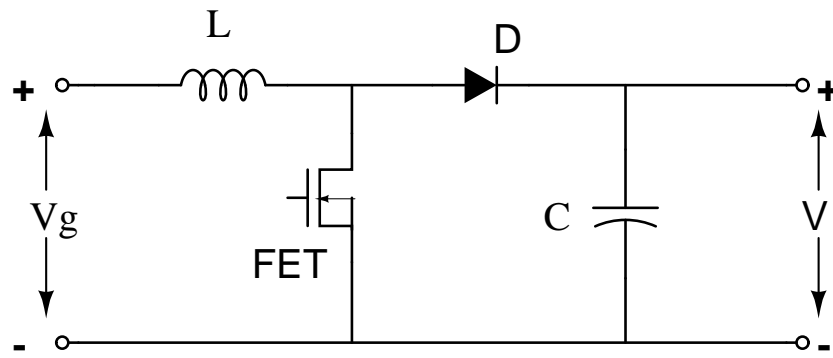
switching regulators include design complexity, high switching noise and higher cost. They also require energy management in the form of a control loop.

4.2 Topologies of Switching Regulators

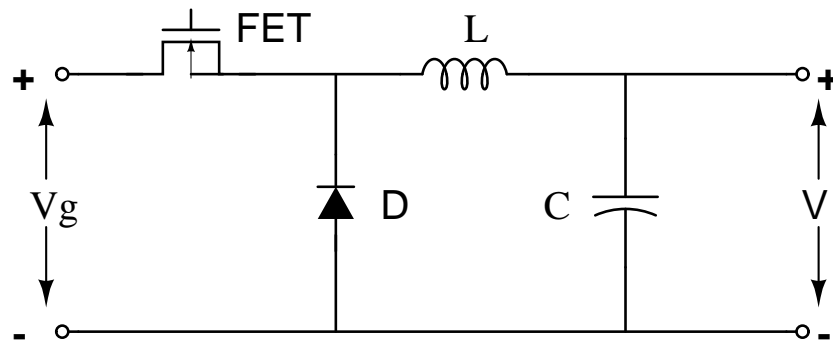
Switching regulators can be up converters (boost), down converter (buck) and inverter(flyback) as shown by (a), (b) and (c) in Fig 4.1, respectively. Some regulators also provide isolation between input and output. A power switch is a key to switching regulators. Vertical Diffused MOS (VDMOS), aka Double Diffused MOS (DMOS) is used as a power switching transistor. These transistors have high switching frequencies and low power dissipation. An inductor is used to control the DC current through the switch, thus reducing the heating. The inductor also serves as a storage element in the charge cycle and provides the energy to the load in the discharge cycle. This makes switching regulators very efficient. The following subsection describes the operation of a buck converter.

4.2.1 Buck Converter

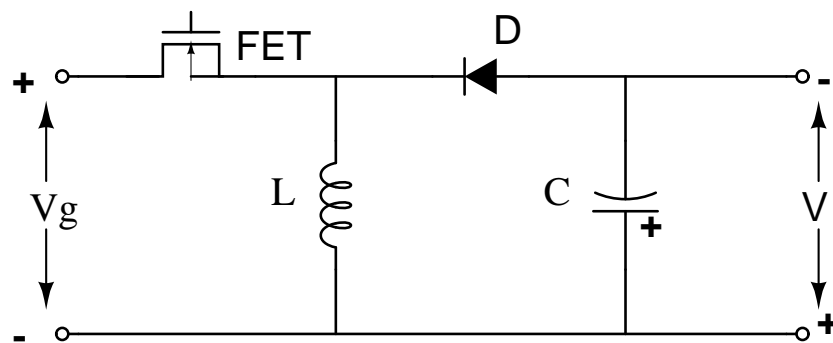
A basic buck converter is shown in Figure 4.2 [51]. A Single Pole Double Throw (SPDT) switch is connected to input DC voltage V_g . When the switch is on position 1, output DC voltage V is equal to V_g and V is equal to 0 when the switch is at position 2. The switch position varies periodically, such that $V_s(t)$ is a rectangular waveform having period T_s and duty cycle D as shown in Figure 4.3. The duty cycle is equal to the fraction of time that the switch is connected in position 1, and hence $0 \leq D \leq 1$. The switching frequency f_s is equal to $1/T_s$. In practice, the SPDT switch is realized using semiconductor devices such as diodes, power MOSFETs, IGBTs, BJTs, or thyristors. Typical switching frequencies lie in the range 1 kHz to 1 MHz, depending on the speed of the semiconductor devices. The switch network changes the DC component of the output waveform. This component is given by the average value of the waveform obtained given by Equation 4.1



(a) Boost Converter



(b) Buck Converter



(c) Inverter (Flyback)

Figure 4.1: Types of Converters

$$V = \frac{1}{T_s} \int_0^{T_s} V_s(t) dt = DV_g \quad (4.1)$$

The integral is equal to the area under the waveform, or the height V_g multiplied by the time DT_s . The switch network reduces the DC component of the voltage by a factor equal to the duty cycle D . Since $0 \leq D \leq 1$, the DC component of V_s is less than or equal to V_g . In addition to the desired DC voltage component V_s , the switch waveform $V_s(t)$ also contains undesired harmonics of the switching frequency. In most applications, these harmonics must be removed, such that the converter output voltage $v(t)$ is essentially equal to the DC component $V = V_s$. A low-pass filter is employed for this purpose. The converter of figure contains a single-section L-C low-pass filter. The filter has corner frequency f_0 given by equation 4.2

$$f_0 = \frac{1}{2\pi\sqrt{LC}} \quad (4.2)$$

The corner frequency f_0 is chosen to be sufficiently less than the switching frequency f_s , so that the filter passes only the DC component of $V_s(t)$.

Ideally, the power dissipated by the converter is zero. For the switch network, when the switch contacts are closed, the voltage across the contacts is equal to zero and hence the power dissipation is zero. When the switch contacts are open, then there is zero current and the power dissipation is again equal to zero. Therefore, the ideal switch network is able to change the DC component of the voltage without dissipation of power. In practice however, since the switch is realized by a MOSFET device it has finite resistance in ON mode and a leakage current in OFF mode which will result in power dissipation. Similarly an ideal filter removes the switching harmonics without dissipation of power but practical inductor and capacitors have finite DC resistances which cause power dissipation. Thus, the converter

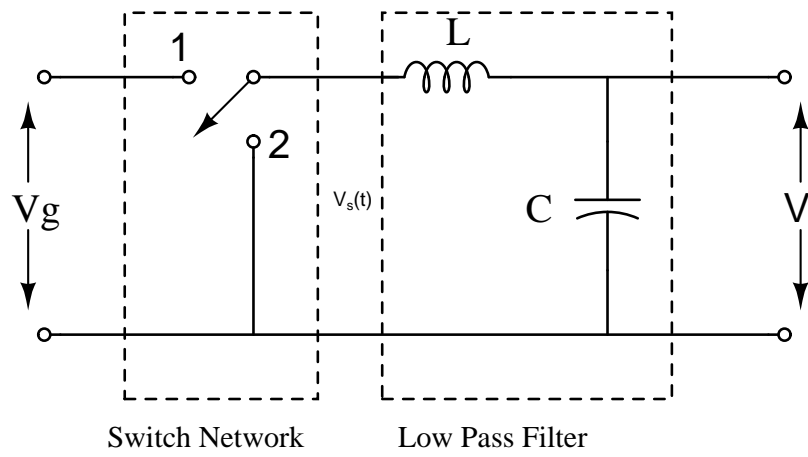


Figure 4.2: A Simple Buck Converter

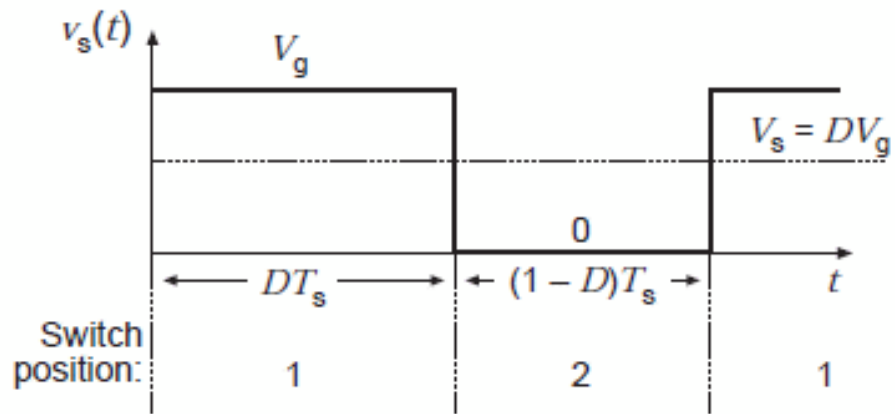


Figure 4.3: Buck Converter output waveform

produces a DC output voltage whose magnitude is controllable via the duty cycle D , using circuit elements that (ideally) do not dissipate power. The conversion ratio, $M(D)$, is given by the ratio of output DC voltage to input DC voltage i.e. For a Buck Converter,

$$M(D) = \frac{V}{V_g} = D \quad (4.3)$$

Efficiency, η of a DC to DC converter is defined by the ration of output DC power to the input DC power.

$$\eta = \frac{P_{out}}{P_{in}} \quad (4.4)$$

4.3 Summary

Although switching techniques are more difficult to implement, switching circuits have almost completely replaced linear power supplies in a wide range of portable and stationary designs. MOSFET power switches are now integrated with controllers to form single-chip solutions. With switching frequencies in MHz range, the output inductor and filter capacitors can be reduced in size, further saving valuable space and component count. As MOSFET power-switch technologies continue to improve, so will switch-mode performance, further reducing cost, size, and thermal management problems.

Chapter 5

System Approach for Power Source Optimization

5.1 Introduction

Most of the work on low power design is focused on designing circuits which consume lower energy and power. As far as the portable electronic devices are concerned, the ultimate aim is to achieve more battery lifetime or, for rechargeable source, perform most operations between consecutive recharges. Optimization of the circuit alone for power and energy may not always result in equivalent optimization of battery lifetime. So a study of the system consisting of battery and the circuit under consideration is required in order to achieve maximum battery lifetime. In general, this lifetime should be measured in terms of the duration of the system operation. A relevant measure is the number of useful clock cycles obtained per battery life or per battery recharge.

Size and weight of the batteries are major design constraints for mobile computing devices. Battery weights are generally proportional to their AHr ratings. Given an application with its load current requirement, a relevant problem is to find a battery with minimum size and weight to run the application. Since the energy drawn from the battery is not always equal to the energy consumed in the device, understanding battery discharge behaviour and its own dissipation are essential for optimal system design. Finding and using a suitable model for a battery is an important part of the problem.

5.2 Problem Definition

Consider a typical battery powered system mentioned in section 2.4.3. The size of a battery is specified in terms of the electrical charge it can supply. A Lithium-ion battery

of 400mAHr can supply 400mA for one hour. It will supply 200mA for two hours. While 400mA is the rated current for this battery, up to three times the rated current or 1.2A can be drawn for a duration of 20 minutes. However, a discharge rate higher than this can cause noticeable loss in the internal impedance of the battery resulting in heating. This results in a loss of efficiency as defined below.

The time for which a fully charged battery can supply current before requiring recharge is called its lifetime. Thus,

$$Ideal\ Lifetime = \frac{AHr\ rating}{Load\ Current\ in\ Amperes} \quad (5.1)$$

The end of lifetime is indicated by significant drop in the terminal voltage. Thus, the end of lifetime for a 4.2 volt Lithium-ion battery is indicated by a drop in terminal voltage below 3 volts. In practice, a battery can maintain an ideal lifetime for load currents smaller than three times the rated current. Thus, a 400mAHr battery can supply up to 1.2A current. For higher currents, there is generally a reduction in actual lifetime due to internal losses. Therefore,

$$Efficiency = \frac{Actual\ lifetime}{Ideal\ Lifetime} \quad (5.2)$$

To avoid loss in efficiency, we must use larger battery. For lithium-ion battery 400mAHr is considered a unit cell. Using multiple cells in parallel enhances the current capacity and lifetime. Thus, a battery size N means a battery consisting of N unit cells. For example, a battery of size N = 5 will be rated at 2Ahr. The problems we address here are [59]:

1. Determine the minimum voltage supply VDD for a synchronous clocked digital system that will meet the performance (critical path delay) requirement. Obtain the load current

for the battery.

2. Determine the minimum battery size (efficiency $\geq 85\%$) for the required load current. The lifetime of the minimum size battery will be 20 minutes. Determine the battery size for given recharge interval. For example, if the minimum battery size is $N = 2$ and the system recharge time is one hour, then we select a battery of size $N = 6$ or 2.4Ahr.

3. For the selected size of the battery, we determine a low performance energy saving supply voltage VDD for which the lifetime of the battery in clock cycles is maximized.

We examine these problems under various system constraints as described by following cases:

- *Case I:* System is performance bound
- *Case II:* Battery size or weight is a primary concern

5.3 Case I: System is performance bound

We analyse the above mentioned problem statements for a case where the system has to meet a certain throughput requirement. We analyze these problems and propose a step wise solution to find a matching battery for an electronic system [59]:

5.3.1 Step 1: Determine circuit characteristics

For understanding the effects of voltage scaling on battery efficiency, we consider a 70 million gate hypothetical system. We assume that the critical path consists of a 32-bit ripple-carry adder consisting of 352 NAND gates. The technology assumed is 45 nanometer bulk CMOS. For simulation, the predictive technology model (PTM) is used [1, 37]. The 32-bit adder was simulated using the HSPICE simulator [42]. The description of this circuit follows:

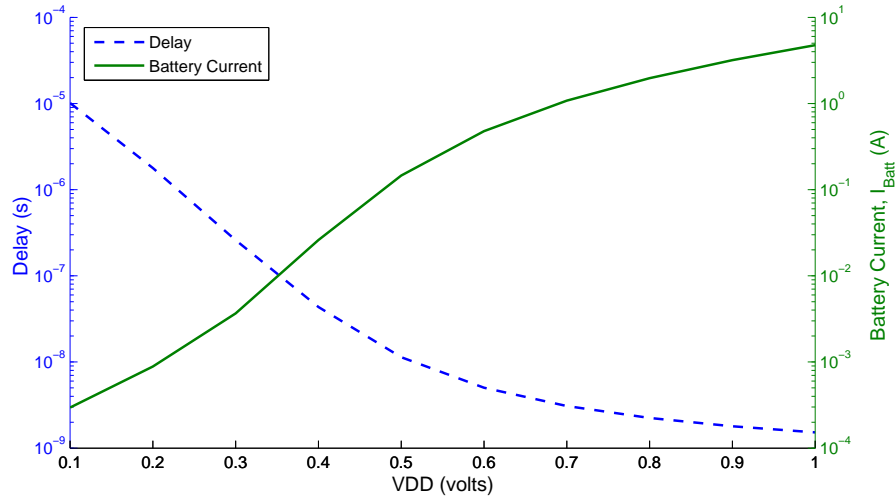


Figure 5.1: Circuit Delay and Current versus VDD obtained from HSPICE simulations

- Function: 32-bit ripple-carry adder
- Inputs: Operand A (32-bit), Operand B (32-bits), Carry-in (1-bit)
- Outputs: Sum (32-bits), Carry-out (1-bit)
- Transistors: 1,472 (352 two and three input NAND gates)
- Technology: 45nm bulk CMOS.
- Critical path: B(0) to Carry-out. Sensitizing vectors (3): A = 8hFFFF FFFF, B = 8h0000 000x, where x changes 0-1-0, Carry-in = 0.

Using the Hspice simulator [42] and the 45nm PTM [1, 37], we determined the critical path delay of the 32-bit adder for VDD ranging from 1.0V to 0.1V at interval of 0.1V. This is shown in Figure 5.1.

We found that the although the circuit slows down by more than three orders of magnitude, it works correctly upto $VDD = 0.1V$, which is below the threshold voltage of 0.292V for the 45nm PTM devices [37]. Next, to determine the average current we simulated the circuit using 100 random vectors. The simulation was repeated for all the same values of

VDD as before. In each case, vectors were applied at an interval equal to the corresponding critical path delay. Assuming a similar activity for the entire 70 million gate system, the average current measured for the 352-gate adder from Hspice simulation was multiplied by 200,000. Considering a 100% efficiency DC-to-DC converter that translates VDD to the 4.2V rated terminal voltage of Lithium-ion battery, we determine the battery load current I_{Batt} by multiplying the circuit current by $VDD/4.2$. That I_{Batt} as a function of VDD is shown in Figure 5.1.

Now, as mentioned in the problem statement, we determine the operating voltage of the circuit based on the throughput requirements. e.g if the circuit needs to work at 200MHz, then from Figure 5.1, the operating voltage is 0.6 V and the corresponding current drawn from the battery is 477mA.

5.3.2 Step 2: Determine smallest battery size

The model of the selected battery type is simulated for various current loads obtained in the previous step. Every battery type has its terminal voltages corresponding to fully charged state and fully discharge state. Using the load current, scaled for the ratio of battery voltage to circuit VDD, the battery model is simulated to determine the terminal voltage as a function of time. In practice this scaling is achieved by a DC-to-DC converter that is known to have high conversion efficiency (greater than 90%) [54, 43]. Alternatively, the circuit of DC-to-DC converter can be attached to the battery model. The time between the fully charged state to the fully discharged state gives the battery lifetime in time units (seconds). This is repeated for increasing battery sizes, normalized with respect to the smallest unit. A lower bound on battery size is determined for a minimum of 85% efficiency. While the selected battery should not be smaller, its actual size is determined by the recharge interval requirement of the system.

We assume the use of Lithium-ion batteries with a unit battery ($N = 1$) of 400mAHr rating. As an example, consider the battery load current $I_{Batt} = 3.6A$ for $VDD = 0.9V$ in

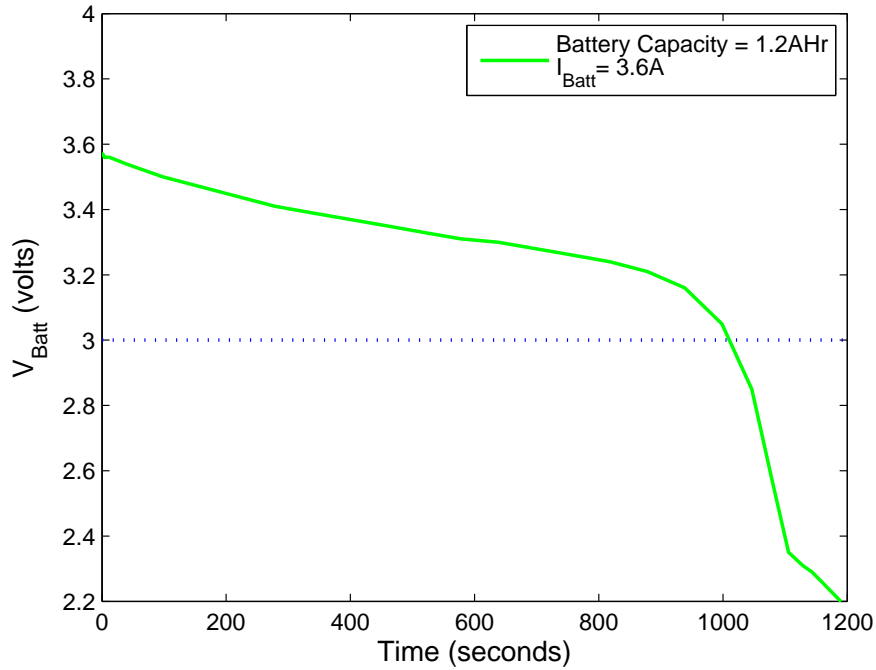


Figure 5.2: V_{Batt} Vs Time when a battery of 1.2 Ahr capacity is subjected to load current, $I_{Batt} = 3.6A$

Figure 5.1. Figure 5.2 shows the battery terminal voltage V_{Batt} obtained from HSPICE [42] simulation of the battery model of Figure 3.1. In this figure, the battery size is $N = 3$, i.e., Capacity = 1.2Ahr. The leakage resistance, usually very large, was taken as 1 gigaohms. All other parameters of the battery model have been described in Section 3.5.

From Figure 5.2, the terminal voltage drops to 3.0V, i.e., battery needs recharge, after it supplies current for 1008 seconds. This is the actual lifetime for this battery. From equation 1, the ideal lifetime is $3600 \cdot 1.2 / 3.6 = 1200$ seconds. This, according to equation 2, gives an 84% efficiency.

Figure 5.3 shows the battery efficiencies obtained in this way for various battery sizes and for varying load currents. We observe,

1. When the load current is small compared to the Ahr rating, the efficiency is 100% or higher. For example, for a battery of size $N = 5$ (2Ahr) the efficiency for $I_{Batt} = 0.6A$ is

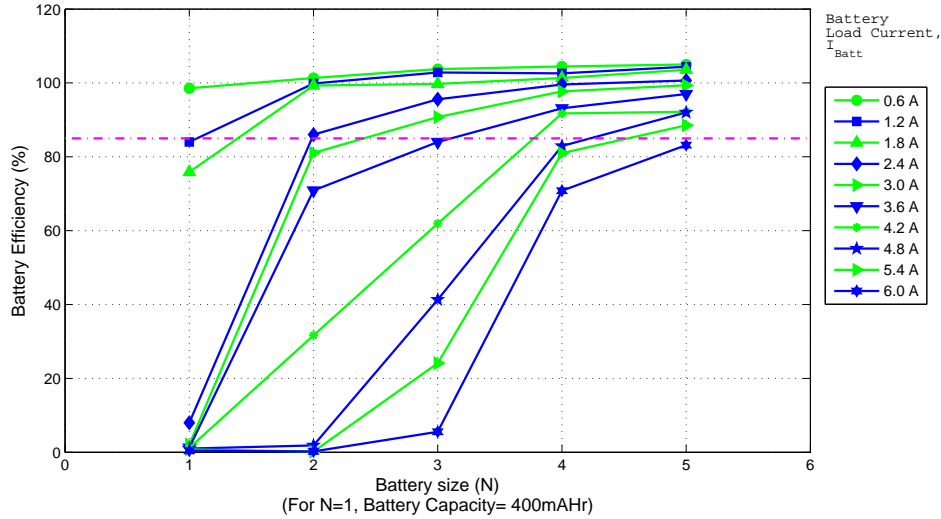


Figure 5.3: Battery efficiency versus battery size for various load currents

107%.

2. When the load current is large compared to the AHr rating, the efficiency can be significantly lower. The 85% line is shown to indicate that a power source with lower efficiency may be considered unacceptable. For any given load current this 85% line allows us to determine the smallest battery that can be used.

Continuing further with our example from previous subsection, with a current of 477 mA and an efficiency of $\geq 85\%$, a battery of size 400 mAHr is chosen. Now this battery is simulated for entire range of voltages and then a graph of supply voltage versus number of cycles per recharge is plotted as shown in figure 5.4. This graph also indicates that as we move towards right from the dotted line the circuit throughput increases and battery efficiency decreases, while moving towards left increases the battery lifetime decreasing throughput.

5.3.3 Step 3: Meeting the lifetime requirement

While the smallest size battery has advantages of weight and cost, it can provide a lifetime (time between recharges) which may not be sufficient. Figure 5.1 is used to determine the battery current I_{Batt} for given performance requirement.

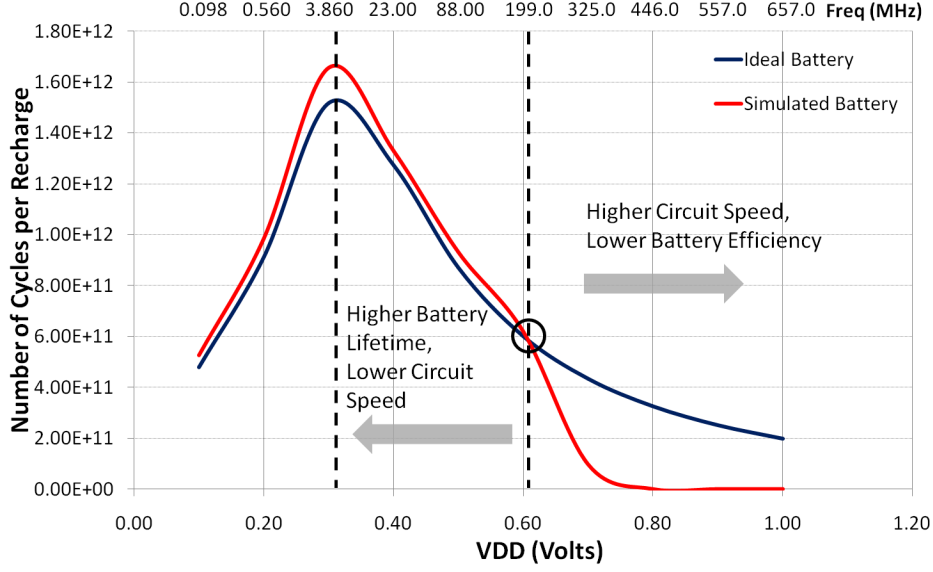


Figure 5.4: Simulation of a 400 mAHr battery for a range of supply voltages (VDD)

Table 5.1: High performance and minimum energy modes of operation.

| Battery size | | 200MHz, $VDD = 0.6V$ | | | 5MHz, $VDD = 0.3V$ | | |
|--------------|-----|----------------------|----------|--------------------|--------------------|--------------------|--------------------|
| | | Effici. | Lifetime | | Effici. | Lifetime | |
| N | AHr | % | sec. | cycles | % | sec. | cycles |
| 1 | 0.4 | 98 | 3000 | 619×10^9 | > 100 | 414×10^3 | 1660×10^9 |
| 4 | 1.6 | 103 | 12300 | 2540×10^9 | > 100 | 1364×10^3 | 6630×10^9 |

Again, continuing with our previous example, consider the system has a battery lifetime requirement of 3 hours. From figure 5.3, the minimum size battery i.e. 400 mAHr ($N=1$) gives 98% efficiency and hence the lifetime is $3600 \times 0.98 \times 0.4/0.477 = 2952$ seconds. To meet the requirement of 3 hours, i.e 10800 seconds, We, therefore, use the battery size of $N = 10800/2952 = 3.658 \approx 4$. So we select a battery of 1600 mAHr. Number of cycles obtained per recharge with these batteries is as shown in the figure 5.5

5.3.4 Step 4: Determine minimum energy modes

The previous step determines two battery sizes, namely, the smallest usable battery that meets the performance requirement and another size that can meet both performance and recharge interval requirements. We now determine maximum lifetime modes for each

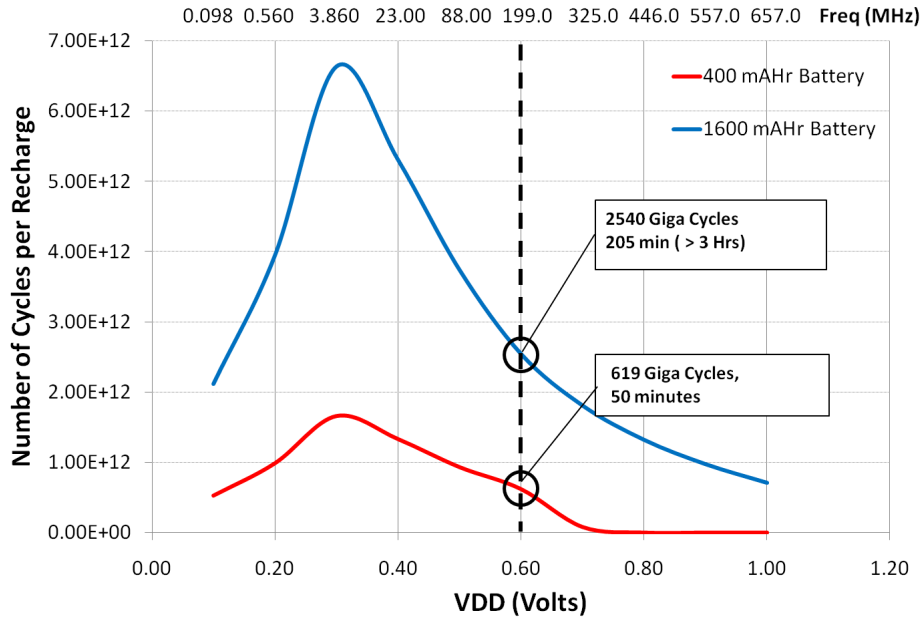


Figure 5.5: Battery lifetimes in clock cycles as a function of chip voltage for 400 mAHr and 1600 mAHr batteries

battery. In this mode the performance requirement is completely relaxed and the supply voltage (VDD) is determined for maximum lifetime in clock cycles. For some nanometer technologies, this VDD can be below the sub-threshold voltage [57].

Most electronic systems have performance and uninterrupted operation requirements that determines the battery size as discussed above. But, a system does not always operate in the maximum performance environment. Lowering VDD that can be easily done by the DC-to-DC converter reduces I_{Batt} and hence extends the battery lifetime. Critical path delay, however, increases and clock frequency must be reduced. A relevant measure of lifetime, therefore, is the lifetime in number of clock cycles. Thus, instead of expressing the lifetime in raw seconds, we express it in terms of computational work units.

Figure 5.6 shows the lifetime in clock cycles as a function of VDD for the two batteries of Table 5.1. According to Figure 5.1, the critical path delay for $VDD = 0.3V$ is 0.2s, giving a clock frequency of 5MHz. The high performance mode and the minimum energy modes are summarized in Table 5.1. The minimum energy mode increases the time between

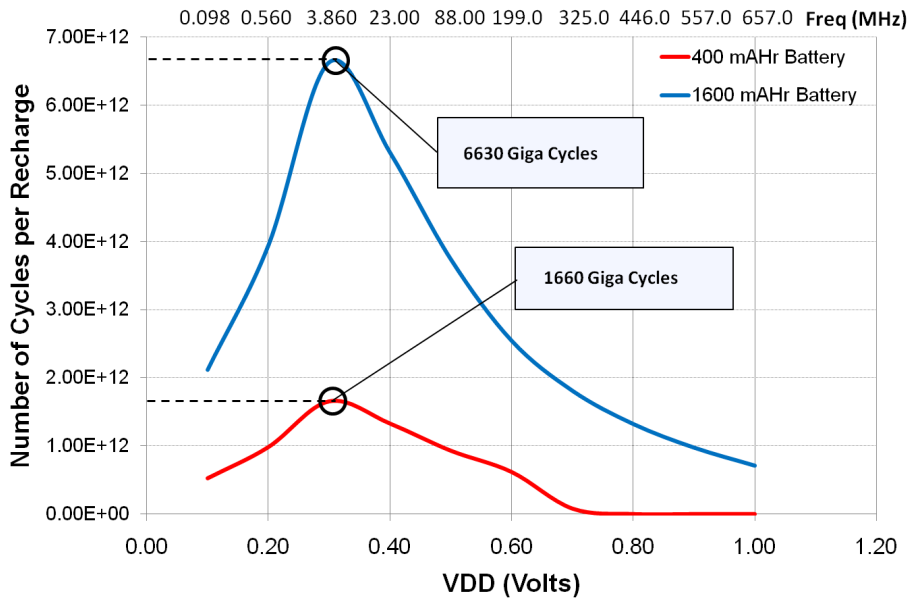


Figure 5.6: Battery lifetimes in clock cycles as a function of chip voltage for 400 mAHr and 1600 mAHr batteries

recharges by thousand fold. That is misleading because the clock frequency is reduced 100 times. However, it does provide more than two fold increase in the number of clock cycles per battery recharge.

5.4 Case II: Battery size or weight is a primary concern

Some applications call for a special set of requirements from the circuit due to a stringent limit on battery size and weight. Applications such as bio-implantable devices, wearable computing devices, hearing aid cannot exceed a certain volume or weight of the battery. Such devices often do not have very high performance requirements. These devices make use of lithium ion batteries which are light weight, have high energy density and are less bulky. One such popular battery is CR2032(CR) and its properties are as described below. Note that even though the battery rating is 225 mAHr, the maximum current that the batter can

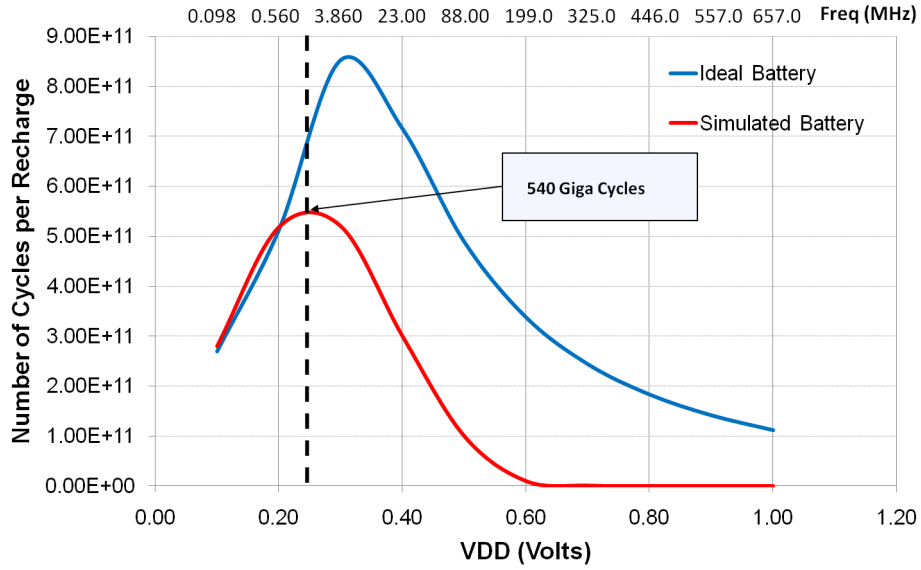


Figure 5.7: Battery lifetimes in number of clock cycles for CR2032 with max. $I_{battery} = 3\text{mA}$ provide is only 3 mA.

CR2032 Lithium ion battery:

- Nominal Voltage: 3V
- Capacity: 225mAHr
- Nominal Current: 0.3 mA
- Maximum Current: 3 mA

A four step analysis, similar to that explained for the previous case, can be carried out for this case. Simulation of the above mentioned CR2032(CR) battery is shown in Figure 5.7. It is clear from Figure 5.7 that though ideal battery can keep providing higher number of cycles for voltage $\geq 0.3\text{ V}$, practically it would have lower efficiency since the maximum current battery can supply is only 3 mA.

5.5 Summary

This chapter shows how a power source is selected to economically satisfy the operational requirements of a system. An electrical model of a battery allows the determination of its lifetime and efficiency. Lifetime measured in terms of clock cycles is shown to be a useful measure. Simulation of the battery as well as that of the circuit being powered allows determination of high performance and minimum energy operational modes. Other applications of battery analysis may be in assessing and optimizing the power management techniques. Given the size of the battery, its efficiency reduces for higher currents. While power reduction is necessary from temperature and other environmental requirements of semiconductor chips, the influence of power reduction on battery lifetime is important for portable devices.

6.1 Problem Statement

Consider a processor built in certain semiconductor technology. If we reduce the supply voltage V , the critical path delay will increase and hence the maximum clock frequency f will have to be decreased. This will reduce the dynamic power in proportion to $V^2 f$. Static power will also decrease as V^2 . However, a measure of energy a computing task will use is the total energy per cycle (EPC), consisting of dynamic EPC and static EPC. Dynamic EPC is proportional to V^2 and static EPC is proportional to V^2/f . We notice that dynamic EPC always reduces with voltage scale down. However, static EPC is proportional to $1/f$, which will increase rapidly as V approaches the threshold voltage.

Thus, for a given technology (i.e., given threshold voltage), there is an optimum supply voltage and a corresponding clock frequency that minimize the total EPC. Any further power reduction by voltage scaling beyond this optimum value will incur an increase in the total EPC, although power will reduce. As the supply voltage gets closer to the threshold voltage, the performance also becomes sensitive to process variation that is common in nano-scale technologies. In practice, therefore, the supply voltage has a lower bound [61]. If further power reduction is required, say, due to battery characteristics, thermal factors or other operational considerations, then clock frequency alone would have to be reduced. This will reduce power but increase energy per cycle (EPC). Dynamic voltage control within a clock period [27] can reduce the EPC but, as pointed out earlier, requires complex control circuitry.

We assume a situation where voltage is at its lowest permissible limit and power must be reduced. Traditionally, we would slow down the clock and let EPC increase. This will be a performance-power trade off that involves an essential energy penalty. We explore an

alternative solution in which clock is not slowed down but performance is degraded, similar to clock slowdown, for power reduction while energy penalty is reduced, especially for high leakage technologies.

6.2 Background on Clock Slowdown (CSD) for Power Reduction

Clock slowdown (CSD) is a known technique for power reduction and we use it as a reference for evaluating the proposed method. When we slow down the clock, dynamic power is reduced in proportion to the clock rate, while leakage power remains unchanged. The computing task now takes longer to complete. This results in the same dynamic energy consumption whereas the leakage energy consumed is more. We will use a processor slowdown factor n . Without loss of generality, n is assumed to be an integer. Thus, $n = 1$ is the normal (rated-clock) operation. Let us define:

$$n = \text{processor slowdown factor} \quad (6.1)$$

$$f = \text{rated clock frequency in Hz} \quad (6.2)$$

$$P_d = \text{dynamic power with rated clock} \quad (6.3)$$

$$P_s = \text{static power with rated clock} \quad (6.4)$$

$$k = P_s/P_d = \text{static power ratio} \quad (6.5)$$

$$T = \text{time duration of a computing task} \quad (6.6)$$

When the processor is slowed down by a factor of n , its power consumption is given by,

$$P_{CSD}(n) = \frac{P_d}{n} + P_s = P_d \frac{1 + kn}{n} \quad (6.7)$$

We notice that a computing task of original duration T is now completed in duration nT . However, we may expect that a reduced current from the battery will result in an enhanced capacity to supply energy and increase the lifetime, L . However, we may expect

that a reduced current from the battery will result in an enhanced capacity to supply energy and increase the lifetime, L . This is often represented by Peukert's law [21, 38]:

$$L = C_1/I^\alpha = C_2/P^\alpha \quad (6.8)$$

where C_1 and C_2 are constants related to the battery capacity, I is the current, and P is power assumed to be drawn at a constant rated voltage. In reality, this condition assumes a steady current. Though not a reality for digital circuits, this condition can be maintained by using a *supercapacitor* and battery combination [31]. In this case, the current fluctuations are smoothed by a large capacitor of several farads capacity. The exponent, α , in equation 6.8 can take different values depending on the type of battery, for the present illustration we use $\alpha = 1.3$.

Next, we denote the power and energy savings by the following ratios:

$$P_{CSDratio} = \frac{P_{CSD}(n)}{P_{CSD}(1)} = \frac{1 + kn}{n(1 + k)} \quad (6.9)$$

$$L_{CSDratio} = \frac{1}{n} \times \frac{1}{(P_{CSDratio})^\alpha} \quad (6.10)$$

and,

$$E_{CSDratio} = nP_{CSDratio} \quad (6.11)$$

We observe that for very low leakage, $k \approx 0$, $P_{CSDratio} = 1/n$ and $L_{CSDratio} = n^{0.3}/(1 + n)$, which show power saving with lifetime enhancement at least for small values of n . To consider very high leakage technologies, let us assume $k = 1$. Then $P_{CSDratio} = (1 + n)/(2n)$. CSD now cannot reduce the power ratio below 0.5 and there is battery lifetime degradation for any clock slowdown factor n . These trends are illustrated in Figure 6.1.

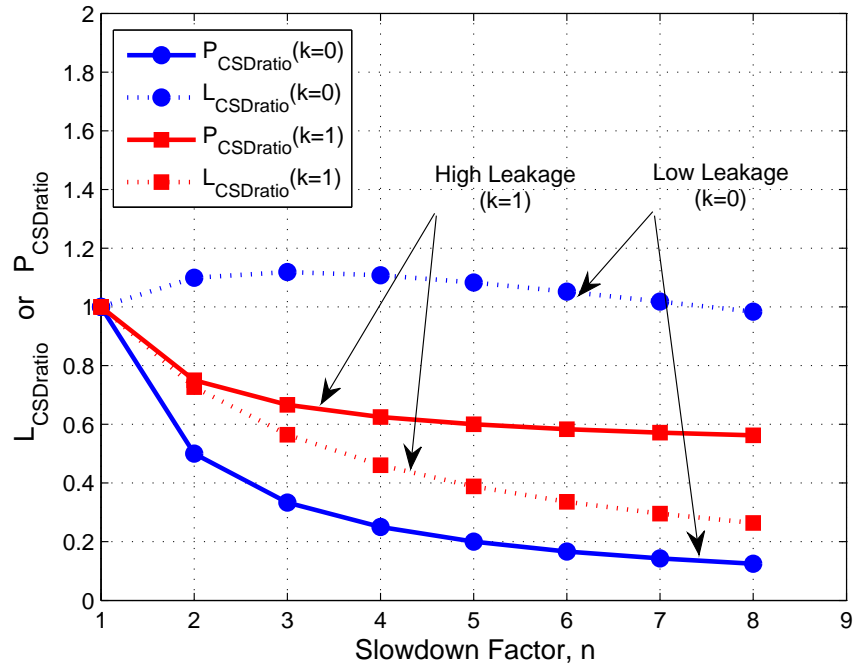


Figure 6.1: Clock slowdown (CSD) power and battery lifetime ratios for low and high leakage technologies.

6.3 Use of NOP for Power

In the next section, we will introduce a new power reduction method called instruction slowdown (ISD) [10]. The processor is slowed down not by clock slowdown but by inserting NOP cycles. The NOP instruction has been used for power optimization. Najeeb *et al.* [25] mix NOP instructions in an instruction sequence to produce a maximum power consuming cycle, which they term as *power virus*. Such an instruction sequence is useful for the design and test of the processor. Lotfi-Kamran *et al.* [23] suggest freezing certain data bits in a pipeline processor whenever a NOP, either contained in the instruction stream or generated due to hazards, is executed. They report about 10% power saving with a modest hardware overhead of 0.1%. Hurd [13] describes a technique of manipulating the positions of NOP instructions in a multiple instruction word architecture so that certain instructions need not be fetched. In another technique, also due to Hurd [12], a NOP instruction is replaced by another instruction called “proxy NOP”. This instruction uses the data patterns of its

neighboring instruction but executes like NOP. It thus reduces activity in the datapath. None of these techniques perform the power management as discussed in the following section.

6.4 Instruction Slowdown (ISD)

In this new methodology [10], the operation of a processor is slowed down for power reduction by inserting non-functional cycles while the rated clock frequency (f) is maintained. This is similar to inserting instruction we call SLOP (slowdown for low power). Although it is described as a purely hardware induced operation, SLOP can be included in the software instruction set.

In a typical implementation, a power management unit (PMU) monitors the system and, if necessary, determines an appropriate slowdown factor (n), which is supplied to the control. The control then inserts the required number of SLOPs in the pipeline. The factor n is assumed to be an integer here but, in general, can be any number that determines the percentage of SLOPs to inserted in the instruction stream.

Hardware execution of SLOP resembles a conventional NOP, stall or bubble [26] with a few differences. First, its execution in a pipeline requires no “fetch” because the control generates it locally. Second, the control generates low power mode signals for various hardware units. To analyze the power and energy relations, we will use the same symbol definitions as in the previous section. We also define a SLOP power factor:

$$\beta = \frac{\text{power consumed by SLOP}}{\text{av. power consumed by non NOP instr.}} \quad (6.12)$$

where $0 \leq \beta \leq 1$. For a slowdown factor n , we insert $n - 1$ SLOPs after each instruction. Consider a period of 1 second, containing f clock cycles. The energy consumed during a regular instruction (assumed to be non-NOP) cycle is $P_d(1 + k)/f$ and that during a SLOP cycle is $\beta P_d(1 + k)/f$. Of those f cycles, f/n are regular instruction cycles and $(n - 1)f/n$ are SLOP cycles. Thus, total power consumption, or energy dissipated per second, is obtained as,

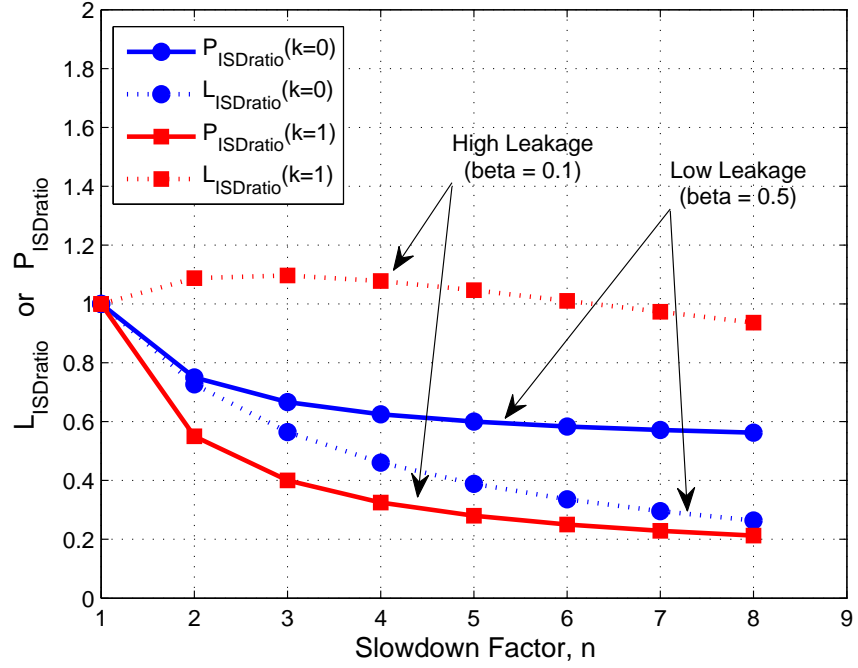


Figure 6.2: Instruction slowdown (ISD) power and battery lifetime ratios for low and high leakage technologies.

$$\begin{aligned}
 P_{ISD}(n) &= \frac{P_d(1+k)}{f} \times \frac{f}{n} + \frac{\beta P_d(1+k)}{f} \times \frac{(n-1)f}{n} \\
 &= P_d(1+k) \frac{\beta n - \beta + 1}{n}
 \end{aligned} \tag{6.13}$$

Similar to the CSD, now also a computing task of original duration T will require nT time. We find the power and battery lifetime ratios as follows:

$$P_{ISDratio} = \frac{P_{ISD}(n)}{P_{ISD}(1)} = \frac{\beta n - \beta + 1}{n} \tag{6.14}$$

$$L_{ISDratio} = \frac{1}{n} \times \frac{1}{(P_{ISDratio})^\alpha} \tag{6.15}$$

These lifetime and power ratios as functions of slowdown factor n are shown in Figure 6.2. The ratios below 1 indicate both power reduction (desirable) and lifetime reduction (undesirable). Notice that power (solid line) is always reduced. More reduction is achieved for higher leakage ($\beta = 0.1$) technology. Lifetime (dotted line) for high leakage improves for small n and then degrades because the NOP cycles consume non-zero energy. However, the lifetime degrades for low leakage technology in a similar way as it did for CSD with high leakage.

6.5 Hardware Implementation of SLOP

We used a 32-bit MIPS pipelined processor for evaluation of the ISD and CSD methods. It has a conventional five-stage pipeline containing the fetch (IF), decode (ID), execute (EX), memory (DM) and write-back (WB) stages [26]. It also contains hazard and forwarding units. We obtained an available VHDL model [9] and synthesized using Mentor Graphics Leonardo Spectrum. This provided us a gate-level model for power analysis.

Various blocks of the processor were extracted as transistor-level netlists using Mentor Graphics Design Architect. Each block was simulated in HSPICE for 1,000 random input vectors with 10ns clock rate ($f = 100\text{MHz}$) to determine the average per cycle dynamic and static energy dissipation. This evaluation was repeated for five CMOS technologies, 180nm, 90nm, 65nm, 45nm and 32 nm, using the predictive technology models (PTM) [1, 4, 37]. The simulation assumed 90°C temperature. A sample result for 32nm is shown in Table 6.1. The last three columns of this table are discussed in a later subsection. Communication buses are not considered separately because all drivers and buffers are included as parts of various hardware blocks.

6.6 Estimating Leakage Factor, k

We wrote a MIPS program that multiplies hexadecimal integers FFFF and 0004 by repeated additions. Our processor has separately addressable instruction (IM) and data

```

0000 LW    $1, X:0002($0)
0001 ADD   $4, $1, $0
0002 ADD   $1, $0, $0
0003 LW    $3, X:0004($0)
0004 LW    $2, X:0003($0)
0005 BEQ   $2, $0, X:0003
0006 SUB   $2, $2, $3
0007 ADD   $1, $1, $4
0008 J     X:0000005
0009 SW    $1, X:0004($3)
000A #J    X:000000A(HALT)

```

Figure 6.3: A MIPS program used for power estimation.

(DM) memories. Initially, $DM(2) = \text{FFFF}$, $DM(3) = 4$, $DM(4) = 1$. Final result is $DM(5) = 0003\text{FFFC}$. The MIPS code is given in Figure 6.3.

This program completes in 34 cycles. The number of times pipeline stages are activated are: 34 IF, 29 ID, 18 EX, 4 DM and 14 WB. The execution statistics of hardware stages and the instruction mix as well as the number of cycles can be easily changed by varying the parameters in the program. It was assembled by hand and the gate-level model was simulated using Mentor Graphics ModelSim. The final result was verified. For power, active blocks in a pipeline stage were identified. Total energy of the pipeline stage was computed by adding the dynamic and static energies of its active blocks. After characterizing each pipeline stage for its energy, the total energy of the program was computed by adding energies of pipeline stages as per the numbers obtained above. The dynamic energy was added up for active stages while the static energy was added up for all blocks for 34 cycles, using the technology-specific data (e.g., Table 6.1 for 32nm). The ratio of total static energy to dynamic energy for each technology gives the respective value of the leakage factor k shown in Table 6.2.

6.7 Power Management for SLOP

Table 6.1 quantitatively shows how power was reduced by clock gating (CG), power gating (PG) and drowsy memories.

Power gating (PG) focuses on leakage. Circuit level approaches for leakage reduction include body bias control [6], dual threshold domino logic [5, 17], input vector control [15] and power gating [11, 18, 29]. We adopt power gating for combinational blocks. It is assumed that the supply line will be gated by pull-up or a pull-down devices that will be put in the cutoff mode during SLOP cycles. This will almost completely eliminate both static and dynamic power during those cycles [14]. We must, however, realize that power gating at clock cycle level represents a design challenge. Studies [6, 32] show that improvements will be needed both in the speed and energy cost of power control and implemented in the present-day design. The basic strategy in power gating is to provide two modes: a low power mode idle stage and an Active mode. The goal is to switch between these modes at appropriate time and in appropriate manner so as to maximize power savings while minimizing the effect on performance. Power gating can be done at the system level which includes a software (OS) controlled power gating of entire CPU or core when the OS detects an idle loop of sufficient duration. Dynamically power gating selected units within a pipeline of a processor is another technique which exploits workload phases and characteristics [11]. Power gating can be implemented in fine grained or coarse grained manner. In fine grained approach, the gating switch is placed in the standard cell library which increases cell area. In coarse grain approach, a component or a set of gates is switched by a collection of switches [18]. Coarse grained approach has less area overhead but involves design complexity to control the switches.

Drowsy mode for caches: Cache memories represent significant fraction of chip area in modern microprocessors. These include multiple levels of instruction caches and data caches. The dynamic and leakage power consumed by instruction and data caches is a sizeable portion of total power consumed by the processor. In the instruction slowdown approach we have

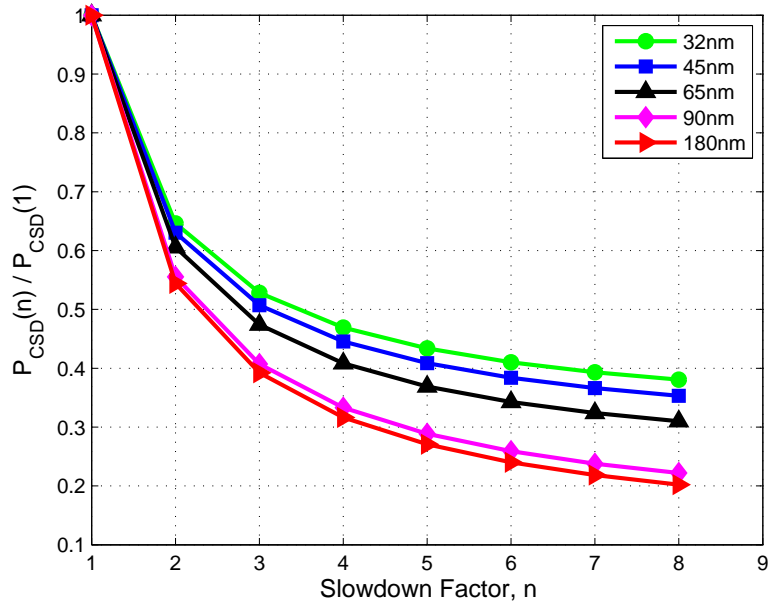


Figure 6.4: Clock slowdown (CSD) power ratios for 180nm, 90nm, 65nm, 45nm and 32nm CMOS technologies. CSD is more effective for low leakage (180nm) technology.

considered clock gating in order to reduce the dynamic power consumption but the leakage power remains the same. There are techniques to reduce this leakage power consumption so as to achieve additional saving. For a given period of time, cache memories generally have their active operations centered to a small number of cells and hence the other cells are not in active state. During SLOP cycles, the memory cells are put into low voltage “drowsy mode”, which can allow up to 75% energy reduction with no more than 1% performance overhead [7]. In addition, decoder and sense amplifier can be power gated. Another technique identifies an application’s cache requirements dynamically, and uses a circuit-level mechanism, “*gated-Vdd*”, to gate the supply voltage to the SRAM cells of the cache’s unused sections to reduce leakage [29].

Clock gating (CG) is applied to registers. Their power is not gated because the state must be preserved. A significant fraction of the dynamic power in a processors is consumed by the clock network and flip-flops. It’s a major component because the clock is fed to most of the circuit blocks and it changes every cycle. The clock buffers can consume 50% or more of total dynamic power [18, 36]. Clock gating turns off the clocks when they are not

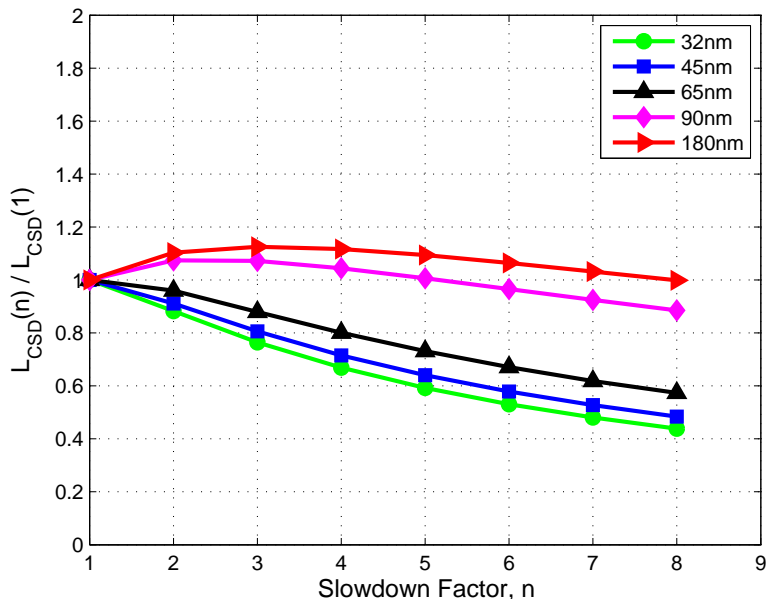


Figure 6.5: Clock slowdown (CSD) battery lifetime ratios for 180nm, 90nm, 65nm, 45nm and 32nm CMOS technologies. Ratios greater than 1 indicate increased battery lifetime through clock slowdown for low leakage 90nm and 180nm technologies.

required or stop them from feeding to the components which are not being used. Results show that up to 43% power saving can be achieved with a possible 20% reduction in area when clock gating replaces the state-retention feedback logic of flip-flops [28]. The clock gating employed in the register file with high switching activity of about 0.25 shows that power saving of about 70% can be achieved [24].

At the time of this writing, we have not completed an evaluation of these techniques. The data in the last two columns of Table 6.1 is based on the references cited here. To compute the SLOP power factor (β) we first weight columns 2 and 3 by columns 5 and 6, respectively. The dynamic and static power of a SLOP cycle is then calculated in a similar way as described before for a regular instruction. The ratio of the power of SLOP cycle to that of the regular instruction cycle is β given in Table 6.2.

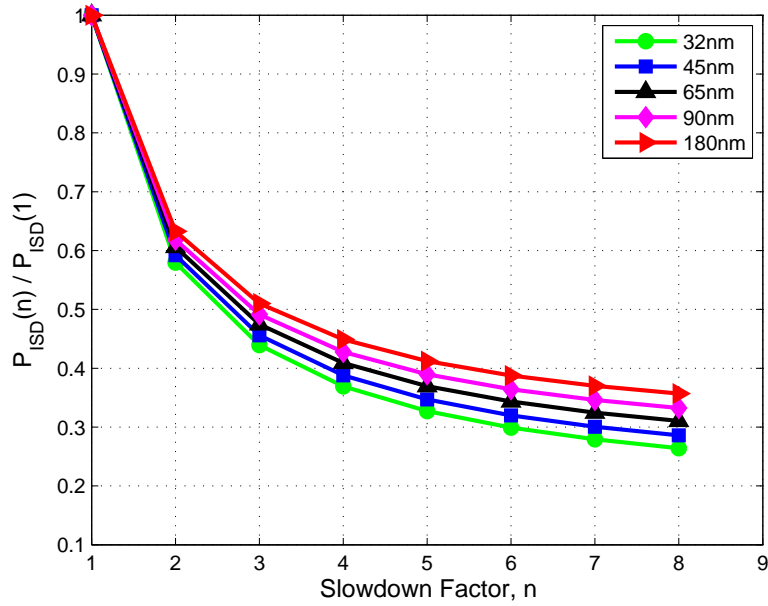


Figure 6.6: Instruction slowdown (ISD) power ratios for 180nm, 90nm, 65nm, 45nm and 32nm CMOS technologies. ISD gives greater power saving for higher leakage technologies.

6.8 Results

Figures 6.4 and 6.5 display power and battery lifetime ratios as functions of the clock slowdown (CSD) factor n for five CMOS technologies. These graphs were computed from equations 6.9 and 6.10, respectively, using values of leakage factor k taken from Table 6.2. We observe that the CSD method degrades for technologies that are finer than 65nm. This is because as n increases, leakage power becomes a dominant factor in the total power. Besides, saving of dynamic energy is compensated for by increase of leakage energy.

Figures 6.6 and 6.7 display power and battery lifetime ratios as functions of the instruction slowdown (ISD) factor n for five CMOS technologies. These graphs were computed from equations 6.14 and 6.15, respectively, using values of SLOP power factor β taken from Table 6.2. Because ISD is assisted by hardware in reducing leakage for the SLOP cycles, we see greater savings of power for high leakage 32nm technology. To compare the two methods directly, we use equations 6.7 and 3.11 to obtain the following ratio:

$$\frac{P_{CSD}}{P_{ISD}} = \frac{1 + kn}{(1 + k)(\beta n - \beta + 1)} \quad (6.16)$$

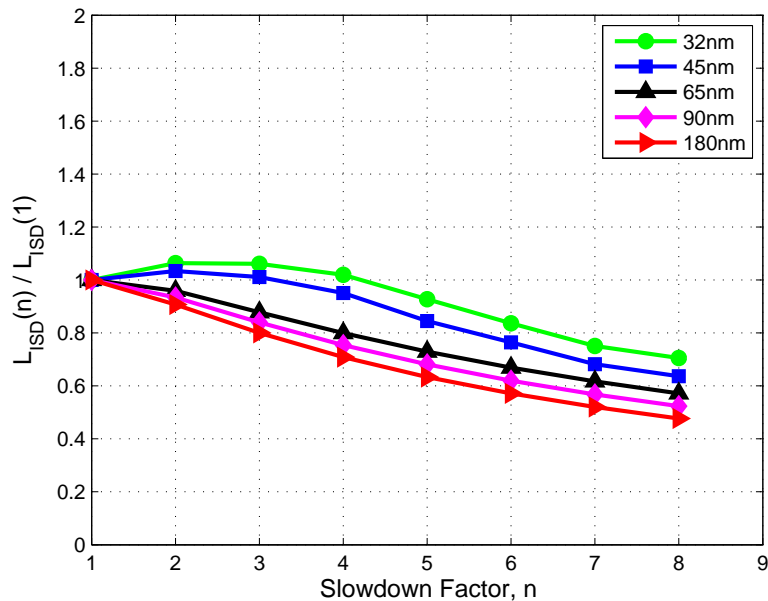


Figure 6.7: Instruction slowdown (ISD) battery lifetime ratios for 180nm, 90nm, 65nm, 45nm and 32nm CMOS technologies. Ratios greater than 1 indicate increased or undegraded battery lifetime through instruction slowdown for high leakage 32nm and 45nm technologies.

The graph in Figure 6.8 shows this ratio as a function of the slowdown factor n for five technologies in the range 180nm through 32nm. The ratio = 1 horizontal line divides this graph in two parts. Points above this line favor ISD and those below favor CSD. The curves will shift upward with improved dynamic power management in high leakage technologies. Results for battery lifetime are shown in Figure 6.9.

Since Peukert's law models only limited properties of a battery. We simulated a representative case of ISD for 32 nm with the battery model [49] mentioned in section 3.5. For such a model, we define Ideal lifetime as,

$$Ideal\ Lifetime = \frac{Ahr\ rating}{Load\ Current\ in\ Amperes} \quad (6.17)$$

A graph of power ratios, energy ratios and ideal battery lifetime ratios against slow down factor, n , is plotted and is as shown in Figure 6.10. From this graph, it is clear that

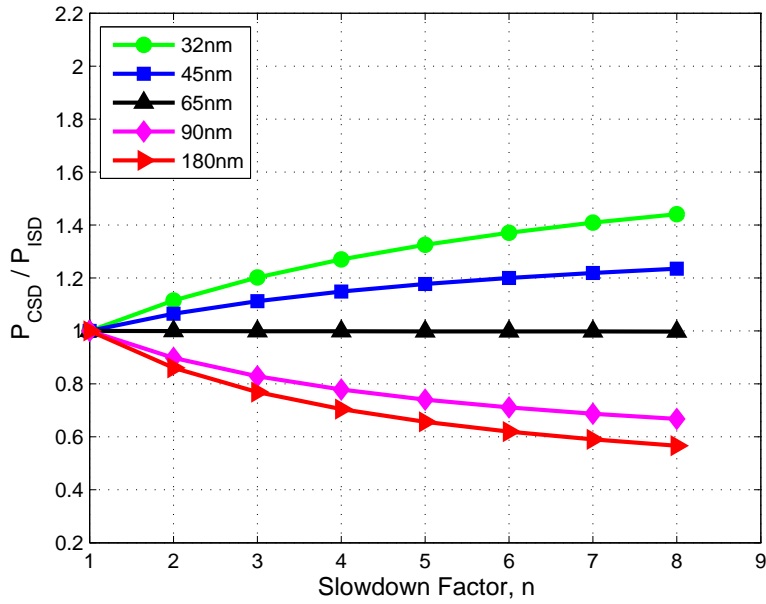


Figure 6.8: Clock slowdown (CSD) vs. instruction slowdown (ISD) power ratios for 180nm, 90nm, 65nm, 45nm and 32nm CMOS technologies. *Ratio* > 1.0 indicates the advantage of ISD for 32nm and 45nm technologies.

with increasing slow down factor, power reduces, energy increases and ideal battery lifetime also reduces due to increase in energy. Ideal battery, however, does not consider the increase in efficiency of the battery due to reduced power (and hence the current drawn from the battery). When the ideal battery was replaced with a practical battery as represented by the model mentioned in section 3.5, we see different results as shown in Figure 6.11

Here zero number of SLOPs correspond to slow down factor (n) of 1, one number of SLOP corresponds to slow down factor (n) of 2 and so on. As we can observe in Figure 6.11, the lifetime saving achieved using ISD exceeds the task completion time for 1, 2 and 3 SLOPs with peak saving at 2 SLOPs. This indicates that for these cases, we gain in terms of battery lifetime with slow down.

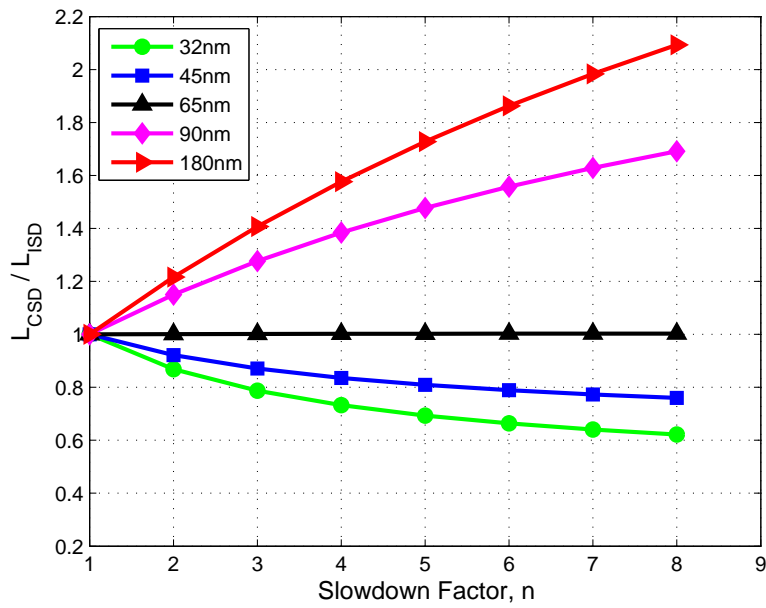


Figure 6.9: Clock slowdown (CSD) vs. instruction slowdown (ISD) battery lifetime ratios for 180nm, 90nm, 65nm, 45nm and 32nm CMOS technologies. *Ratio* < 1.0 indicates the advantage of ISD for 32nm and 45nm technologies.

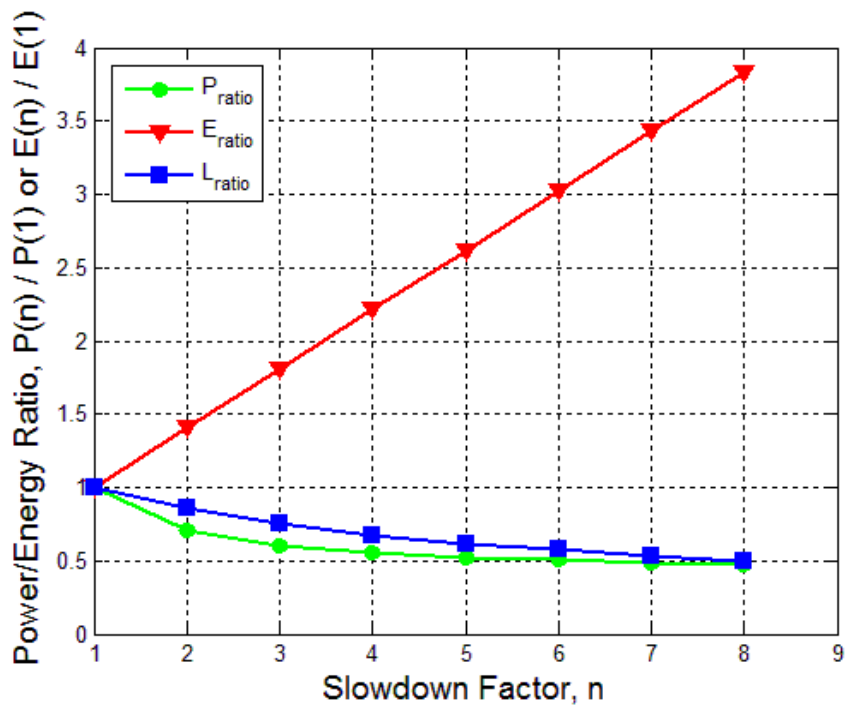


Figure 6.10: Power ratio, energy ratio and ideal battery lifetime ratio plotted against slow down factor, n, for ISD in 32nm

Table 6.1: HSPICE simulation (32nm CMOS, 90°C).

| Hardware block | Energy/cycle | | SLOP power | | |
|-------------------|--------------|-------------|---------------|-----------|------------|
| | Dyn. nJ | Stat. nJ | Power mode | Dyn. % | Stat. % |
| PC | 85114 | 17742 | CG | 25 | 100 |
| PC+1 adder | 28947 | 6536 | PG | 0 | 0 |
| IM | 6780 | 3209 | Drowsy | 25 | 25 |
| Regfile | 98262 | 192375 | CG | 30 | 100 |
| Forwarding | 31297 | 4090 | PG | 0 | 0 |
| Hazard | 25421 | 3744 | PG | 0 | 0 |
| Controller | 14338 | 2973 | None | 100 | 100 |
| 32-b ALU | 263815 | 22346 | PG | 0 | 0 |
| 32-b comp | 39710 | 5695 | PG | 0 | 0 |
| DM | 64343 | 50699 | Drowsy | 25 | 25 |
| 3-1 mux | 392374 | 56299 | PG | 0 | 0 |
| 2-1 mux | 204456 | 44106 | PG | 0 | 0 |
| BrnchAddrCal | 181878 | 13680 | PG | 0 | 0 |
| IF/ID reg | 156027 | 32048 | CG | 50 | 100 |
| ID/EX reg | 213447 | 58412 | CG | 50 | 100 |
| EX/DM reg | 131033 | 34324 | CG | 50 | 100 |
| DM/WB reg | 127885 | 33481 | CG | 50 | 100 |
| ForwDM/WB | 5820 | 1009 | PG | 0 | 0 |

Table 6.2: Leakage factor (k) and SLOP power factor (β).

| Technology | Leakage factor k | SLOP power factor β |
|------------|--------------------|---------------------------|
| 180nm | 0.097 | 0.265081 |
| 90nm | 0.124 | 0.23699 |
| 65nm | 0.268 | 0.212003 |
| 45nm | 0.353 | 0.183881 |
| 32nm | 0.413 | 0.159012 |

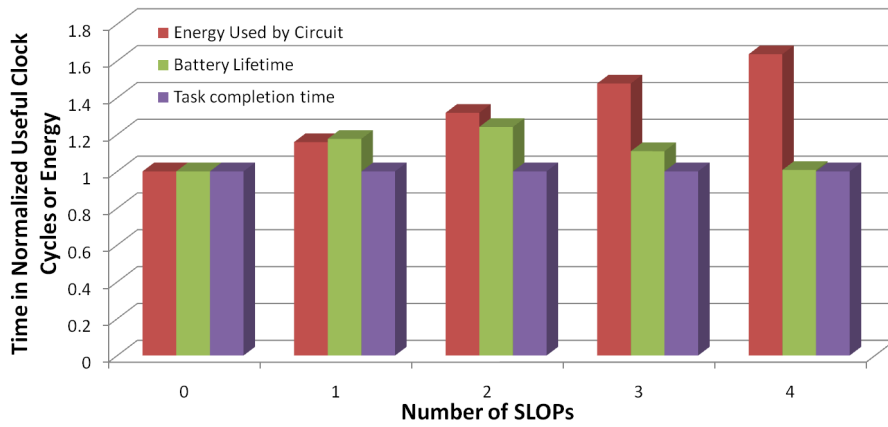


Figure 6.11: Circuit energy, battery lifetime and task completion time plotted against number of SLOPs, for ISD in 32nm

Chapter 7

Conclusion

This work provides an insight into the power source optimization techniques. We present a broad categorization of optimization techniques and propose two methods which fall in voltage management and functional management categories.

First method demonstrates how a power source is selected to economically satisfy the operational requirements of a system. An electrical model of a battery allows the determination of its lifetime and efficiency. Lifetime measured in terms of clock cycles is shown to be a useful measure. Simulation of the battery as well as that of the circuit being powered allows determination of high performance and minimum energy operational modes. Other applications of battery analysis may be in assessing and optimizing the power management techniques. Given the size of the battery, its efficiency reduces for higher currents. While power reduction is necessary from temperature and other environmental requirements of semiconductor chips, the influence of power reduction on battery lifetime is important for portable devices.

The other proposed method of instruction slowdown (ISD) has advantages in power saving for high leakage technologies. We suggest combining the slowdown methods with overall supply voltage scaling. Voltage reduction will save dynamic and static power as well as energy. But the increased hardware delay will necessitate a clock slowdown. Thus, for $n = 2$, CSD may be used. Thereafter, $n > 2$ slowdown should use ISD. The throughput aspect of slowdown methods is not studied. CSD preserves all hazard penalties and throughput drops as $1/n$. ISD will eliminate hazards progressively as n increases. SLOP is presented purely as an internal mechanism supported by power management and control hardware.

Its inclusion in the instruction set will allow compilers to explore creative ways to use the power management hardware.

Bibliography

- [1] <http://www.eas.asu.edu/ptm>.
- [2] L. Benini and G. D. Micheli, “Dynamic Power Management, Design Techniques and CAD Tools”, Springer, 1998.
- [3] I. Buchmann, “Batteries in a PortableWorld: A Handbook on Rechargeable Batteries for Non-Engineers”, Richmond, British Columbia: Cedex Electronics, Inc., second edition, 2001.
- [4] Y. Cao, T. Sato, D. Sylvester, M. Orshansky, and C. Hu, “New Paradigm of Predictive MOSFET and Interconnect Modeling for Early Circuit Design”, in Proc. Custom Integrated Circuits Conference, 2000, pp.201-204.
- [5] S. Dropsho, V. Kursun, D. H. Albonesi, S. Dwarkadas, and E. G. Friedman, “Managing Static Leakage Energy in Microprocessor Functional Units”, in Proc. 35th Annual International Symp. Microarchitecture, MICRO, 2002, pp. 321-332.
- [6] D. Duarte, Y. F. Tsai, N. Vijaykrishnan, and M. J. Irwin, “Evaluating Run-Time Techniques for Leakage Power Reduction”, in Proc. 15th International Conf. VLSI Design, 2002.
- [7] K. Flautner, N. S. Kim, S. Martin, D. Blaauw, and T. Mudge, “Drowsy Caches: Simple Techniques for Reducing Leakage Power”, in Proc. International Symposium on Computer Architecture, 2002, pp.148-157.
- [8] M. Horowitz, T. Indermaur, and R. Gonzalez, “Low-Power Digital Design”, in Proc. International Symp. Low Power Electronics and Design, 1994, pp. 8-11.
- [9] A. Arthurs and L. Ngo, “Analysis of the MIPS 32-Bit, Pipelined Processor Using Synthesized VHDL,” Technical report, University of Arkansas, Department of Computer Science and Engineering. www.csce.uark.edu/~ajarthu/papers/mips_vhdl.pdf.
- [10] Khushaboo Sheth, “A Hardware-Software Processor Architecture using Pipeline Stalls for Leakage Power Management”, Master’s Thesis, Auburn University, December 2008
- [11] Z. Hu, A. Buyuktosunoglu, V. Srinivasan, V. Zyuban, H. Jacobson, and P. Bose, “Microarchitectural Techniques for Power Gating of Execution Units”, in Proc. International Symp. Low Power Electronics and Design, 2004, pp. 32-37.
- [12] L. L. Hurd, “Power Reduction for Multiple-Instruction-Word Processors with Proxy NOP Instructions”, U.S. Patent 6535984, March 18, 2003.

- [13] L. L. Hurd, "Power Saving by Disabling Memory Block Access for Aligned NOP Slots During Fetch of Multiple Instruction Words" U.S. Patent 6442701, August 27, 2002.
- [14] J. Frenkil and S. Venkatraman, "Power Gating Design Automation", in D. Chinnery and K. Keutzer, "Closing the Power Gap Between ASIC and Custom Tools and Techniques for Low-Power Design", chapter 10, pp.251-280, Springer, 2007.
- [15] M. C. Johnson, D. Somasekhar, L.-Y. Chiou, and K. Roy, "Leakage Control with Efficient Use of Transistor Stacks in Single Threshold CMOS", IEEE Trans. Very Large Scale Integration (VLSI) Systems, vol. 10, no. 1, pp.1-5, Feb. 2002.
- [16] "Mobile Intel Pentium 4 Processor with 533 MHz Front Side Bus", Intel Incorporation, January 2004.
- [17] J. T. Kao and A. P. Chandrakasan, "Dual-Threshold Voltage Techniques for Low-Power Digital Circuits", IEEE Journal of Solid-State Circuits, vol. 35, no. 7, pp. 1009-1018, July 2000.
- [18] M. Keating, D. Flynn, R. Aitken, A. Gibbons, and K. Shi, "Low Power Methodology Manual for System On Chip Design", Boston: Springer, 2008.
- [19] S. Narendra, A. Chandrakasan, "Leakage in Nanometer CMOS Technologies", Springer, 2006
- [20] Gary Yeap, "Practical Low Power Digital VLSI Design", Boston: Kluwer Academic Publishers, 1998
- [21] D. Linden and T. Reddy, "Handbook of Batteries", 3rd Edition. McGraw-Hill, 2001.
- [22] J. M. Rabaey, M. Pedram, "Low Power Design Methodologies", Kluwer Academic Publishers, 1996.
- [23] P. Lotfi-Kamran, A. Rahmani, A. Salehpour, A. Afzali-Kusha, and Z. Navabi, "Stall Power Reduction in Pipelined Architecture Processors", in Proc. of 21st International Conference on VLSI Design, 2008, pp. 541546.
- [24] M. Mueller, A. Wortmann, S. Simon, M. Kugel, and T. Schoenauer, "The Impact of Clock Gating Schemes on the Power Dissipation of Synthesizable Register Files", in Proc. International Symp. Circuits and Systems, volume 2, 2004, pp. 609-612.
- [25] K. Najeeb, V. V. R. Konda, S. S. Hari, V. Kamakoti, and V. M. Vedula, "Power Virus Generation Using Behavioral Models of Circuits, in Proc. 25th IEEE VLSI Test Symposium", 2007, pp. 35-40.
- [26] D. A. Patterson and J. L. Hennessy, "Computer Organization and Design: The Hardware/Software Interface", Fourth Edition. Morgan Kaufmann, 2009.
- [27] B. Yu and M. L. Bushnell, "A Novel Dynamic Power Cutoff Technique (DPCT) for Active Leakage Reduction in Deep Submicron CMOS Circuits", Proc. International Symp. Low Power Electronics and Design, pp.214-219, 2006.

- [28] K. C. Pokhrel, “Physical and Silicon Measures of Low Power Clock Gating Success: An Apple to Apple Case Study”, Synopsys Users Group (SNUG), 2007.
- [29] M. Powell, S.-H. Yang, B. Falsafi, K. Roy, and T. N. Vijaykumar, “Gated-Vdd: A Circuit Technique to Reduce Leakage in Deep-Submicron Cache Memories”, in Proc. International Symp. Low Power Electronics and Design, 2000, pp. 90-95.
- [30] V. Tiwari, P. Ashar, S. Malik, “Technology Mapping for Low Power”, 30th Design Automation Conference, 1993, pp. 74-79
- [31] R. F. Service, “New Supercapacitor Promises to Pack More Electrical Punch”, Science, vol. 313, p.902, 18 Aug. 2006.
- [32] J. W. Tschanz, S. G. Narendra, Y. Ye, B. A. Bloechel, S. Borkar, and V. De, “Dynamic Sleep Transistor and Body Bias for Active Leakage Power Control of Microprocessors”, IEEE Jour. Solid-State Circuits, vol. 38, no. 11, pp. 1838-1845, Nov. 2003.
- [33] O. S. Unsal, I. Koren, C. M. Krishna, and C. A. Moritz, “Cool-Fetch: Compiler-Enabled Power-Aware Fetch Throttling”, IEEE Computer Architecture Letters, vol. 1, Apr. 2002.
- [34] H.Wang, Y. Guo, I. Koren, and C. M. Krishna, “Compiler-Based Adaptive Fetch Throttling for Energy Efficiency”, in IEEE International Symp. on Performance Analysis of Systems and Software, Mar. 2006, pp. 112119.
- [35] W. Wolf, “Cyber-physical Systems”, Computer, vol. 42, no. 3, pp. 8889, Mar. 2009.
- [36] K.-S. Yeo and K. Roy, “Low-Voltage, Low-Power VLSI Subsystems”, McGraw-Hill, 2005.
- [37] W. Zhao and Y. Cao, “New Generation of Predictive Technology Model for Sub-45nm Early Design Exploration”, IEEE Transactions on Electron Devices, vol. 53, pp.2816-2823, Nov. 2006.
- [38] R. Rao, S. Vrudhula, and D. N. Rakhmatov, “Battery Modeling for Energy-Aware System Design”, Computer, vol. 36, no. 12, pp. 77-87, Dec. 2003.
- [39] M. Doyle, T.F. Fuller, and J. Newman, “Modeling of Galvanostatic Charge and Discharge of the Lithium/Polymer/Insertion Cell”, J. Electrochemical Soc., vol.140, no. 6, 1993, pp. 1526-1533.
- [40] T.F. Fuller, M. Doyle, and J. Newman, “Simulation and Optimization of the Dual Lithium Ion Insertion Cell”, J. Electrochemical Soc., vol. 141, no. 1, 1994, pp. 1-10.
- [41] J.S. Newman, “FORTRAN Programs for Simulation of Electrochemical Systems, Dualfoil.f Program for Lithium Battery Simulation”; www.cchem.berkeley.edu/jsngrp/fortran.html.

- [42] Synopsys, Inc., “HSPICE The Gold Standard for Accurate Circuit Simulation”, www.synopsys.com/Tools/Verification/AMSVerification/CircuitSimulation/HSPICE/Documents/hspice_ds.pdf.
- [43] M. Pedram and Q. Wu, “Design Considerations for Battery-Powered Electronics”, Proc. 36th ACM/IEEE Design Automation Conference, ACM Press, 1999, pp. 861-866.
- [44] D.N. Rakhmatov and S.B.K. Vrudhula, “An Analytical High-Level Battery Model for Use in Energy Management of Portable Electronic Systems”, Proc. 2001 IEEE/ACM Intl Conf. Computer-Aided Design, IEEE Press, 2001, pp. 488-493.
- [45] D. Rakhmatov, S. Vrudhula, and C. Chakrabarti, “Battery-Conscious Task Sequencing for Portable Devices Including Voltage/Clock Scaling, Proc. 39th Design Automation Conf., ACM Press, 2002, pp.189-194.
- [46] Kanishka Lahiri , Sujit Dey , Debashis Panigrahi , Anand Raghunathan, “Battery-Driven System Design: A New Frontier in Low Power Design”, Proceedings of the 2002 conference on Asia South Pacific design automation/VLSI Design, p.261, January 07-11, 2002
- [47] P. Rong and M. Pedram, “An Analytical Model for Predicting the Remaining Battery Capacity of Lithium-Ion Batteries”, Proc. 2003 Design, Automation and Test in Europe Conf. and Exposition, IEEE CS Press, 2003, pp. 1148-1149.
- [48] T. L. Martin, “Balancing Batteries, Power and Performance: System Issues in CPU Speed-Setting for Mobile Computing”, PhD thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, 1999.
- [49] M. Chen and G. A. Rincon-Mora, “Accurate Electrical Battery Model Capable of Predicting Runtime and I-V Performance”, IEEE Transactions on Energy Conversion, vol. 21, no. 2, pp. 504-511, June 2006.
- [50] H.J. Bergveld, W.S. Kruijt, and P.H.L. Notten, “Electronic- Network Modeling of Rechargeable NiCd Cells and Its Application to the Design of Battery Management Systems”, J. Power Sources, vol. 77, no. 2, 1999, pp. 143-158
- [51] R. W. Erickson, ”DC-DC power converters”, Wiley Encyclopedia of Electrical and Electronics Engineering, pp. 1988:Wiley
- [52] S.C. Hageman, “Pspice Models Nickel-Metal-Hydride Cells”, EDN Access, 2 Feb. 1995; www.reedelectronics.com/ednmag/archives/1995/020295/03di1.htm.
- [53] S. Gold, A PSPICE Macromodel for Lithium-Ion Batteries, Proc. 12th Ann. Battery Conf. Applications and Advances, IEEE Press, 1997, pp. 215-222.
- [54] L. Benini, G. Castelli, A. Macci, E. Macci, M. Poncino, and R. Scarsi, “Discrete-time battery models for system-level low-power design”, IEEE Trans. VLSI Systems, vol. 9, no. 5, pp. 630640, Oct. 2001.

- [55] L. Benini, G. Castelli, A. Macii, E. Macii, M. Poncino, and R. Scarsi, “A Discrete-Time Battery Model for High-Level Power Estimation”, in Proceedings Conference on Design, Automation and Test in Europe, Mar. 2000, pp. 3541.
- [56] Weiser, M., Welch, B., Demers, A., AND Shenker, S. “Scheduling for reduced CPU energy”, Proceedings of OS Design and Implementation, 1994.
- [57] A. Wang, B. H. Calhoun, and A. P. Chandrakasan, “Sub-Threshold Design for Ultra Low-Power Systems”, Springer, 2006.
- [58] H. Wang and Y. Guo and I. Koren and C. M. Krishna, “Compiler-Based Adaptive Fetch Throttling for Energy-Efficiency”, *IEEE International Symp. on Performance Analysis of Systems and Software*, pp.112-119, Mar, 2006
- [59] Kulkarni, M., Agrawal, V., “Matching Power Source to Electronic System: A tutorial on battery simulation”, VLSI Design and Test Symposium, July, 2010
- [60] D. A. Patterson, “The Trouble with Multi-Cores”, *IEEE Spectrum*, vol. 47, no. 7, pp. 28-32 and 52-53, July 2010.
- [61] Jan Rabaey, “Low Power Design Essentials”, Springer, 2009