Low Power Design Essentials
Series on Integrated Circuits and Systems

Series Editor: Anantha Chandrakasan
Massachusetts Institute of Technology
Cambridge, Massachusetts

Low Power Design Essentials
Jan Rabaey

Carbon Nanotube Electronics
Ali Javey and Jing Kong (Eds.)

Wafer Level 3-D ICs Process Technology
Chuan Seng Tan, Ronald J. Gutmann, and L. Rafael Reif (Eds.)

Adaptive Techniques for Dynamic Processor Optimization: Theory and Practice
Alice Wang and Samuel Naffziger (Eds.)

mm-Wave Silicon Technology: 60 GHz and Beyond
Ali M. Niknejad and Hossein Hashemi (Eds.)

Ultra Wideband: Circuits, Transceivers, and Systems
Ranjit Gharpurey and Peter Kinget (Eds.)

Creating Assertion-Based IP
Harry D. Foster and Adam C. Krolnik

Design for Manufacturability and Statistical Design: A Constructive Approach
Michael Orshansky, Sani R. Nassif, and Duane Boning

Low Power Methodology Manual: For System-on-Chip Design
Michael Keating, David Flynn, Rob Aitken, Alan Gibbs, and Kaijian Shi

Modern Circuit Placement: Best Practices and Results
Gi-Joon Nam and Jason Cong

CMOS Biotechnology
Hakho Lee, Donhee Ham and Robert M. Westervelt

SAT-Based Scalable Formal Verification Solutions
Malay Ganai and Aarit Gupta

Ultra-Low Voltage Nano-Scale Memories
Kiyoo Itoh, Masashi Horiguchi and Hitoshi Tanaka

Continued after index
Jan Rabaey

Low Power Design Essentials
To Kathelijn

For so many years, my true source of support and motivation.

To My Parents

While I lost you both in the past two years, you still inspire me to reach ever further.
Preface

Slide 0.1

Welcome to this book titled “Low Power Design Essentials”. (A somewhat more accurate title for the book would be “Low Power Digital Design Essentials”, as virtually all of the material is focused on the digital integrated-circuit design domain.)

In recent years, power and energy have become one of the most compelling issues in the design of digital circuits. On one end, power has put a severe limitation on how fast we can run our circuits; at the other end, energy reduction techniques have enabled us to build ubiquitous mobile devices that can run on a single battery charge for an exceedingly long time.

Slide 0.2

You may wonder why there is a need for yet another book on low-power design, as there are quite a number of those already on the market (some of them co-authored by myself). The answer is quite simple: all these books are edited volumes, and target the professional who is already somewhat versed in the main topics of design for power or energy. With these topics becoming one of the most compelling issues in design today, it is my opinion that it is time for a book with an educational approach. This means building up from the basics, and exposing the different subjects in a rigorous and methodological way with consistent use of notations and definitions. Concepts are illustrated with examples using state-of-the-art technologies (90 nm and below). The book is primarily intended for use in short-to-medium length courses on low-power design. However, the format also should work well for the working professional, who wants to update her/himself on low-power design in a self-learning manner.
This preface also presents an opportunity for me to address an issue that has been daunting low-power design for a while. Many people in the field seem to think that it is just a “bag of tricks” applied in a somewhat ad hoc fashion, that it needs a guru to get to the bottom, and that the concept of a low-power methodology is somewhat an oxymoron. In fact, in recent years researchers and developers have demonstrated that this need not be the case at all. One of the most important realizations over the past years is that minimum-energy design, though interesting, is not what we truly are pursuing. In general, we design in an energy–delay trade-off space, where we try to find design with the lowest energy for a given performance, or vice versa. A number of optimization and design exploration tools can be constructed that help us to traverse this trade-off space in an informed fashion, and this at all levels of the design hierarchy.

In addition to adhering to such a methodology throughout the text, we are also investigating the main roadblocks that we have to overcome in the coming decades if we want to keep reducing the energy per operation. This naturally leads to the question of what the physical limits of energy scaling might be. Wherever possible, we also venture some perspectives on the future.

An Innovative Format

- Pioneered in W. Sansen’s book Analog Design Essentials (Springer)
- PowerPoint slides present a quick outline of essential points and issues, and provide a graphical perspective
- Side notes provide depth, explain reasonings, link topics
- Supplemented with web-site: http://bwrc.eecs.berkeley.edu/LowPowerEssentials
- An ideal tool for focused-topic courses

Slide 0.3

Already in this preface, you observe the somewhat unorthodox approach the book is taking. Rather than choosing the traditional approach of a lengthy continuous text, occasionally interspersed with some figures, we use the reverse approach: graphics first, text as a side note. In my experience, a single figure does a lot more to convey a message than a page of text (“A picture is worth a 1000 words”). This approach was pioneered by Willy Sansen in his book Analog Design Essentials (also published by Springer). The first time I saw the book, I was immediately captivated by the idea. The more I looked at it, the more I liked it. Hence this book . . . . When browsing through it, you will notice that the slides and the notes play entirely
different roles. Another advantage of the format is that the educator has basically all the lecturing material in her/his hands right away. Besides distributing the slideware freely, we also offer additional material and tools on the web-site of the book.

Slide 0.4

The outline of the book proceeds as follows: After first establishing the basics, we proceed to address power optimization in three different operational modes: design time, standby time, and run time. The techniques used in each of these modes differ considerably. Observe that we treat dynamic and static power simultaneously throughout the text—in today’s semiconductor technology, leakage power is virtually on par with switching power.

Hence separating them does not make much sense. In fact, a better design is often obtained if the two are carefully balanced. Finally, the text concludes with a number of general topics such as design tools, limits on power, and some future projections.

Slide 0.5

Putting a book like this together without help is virtually impossible, and a couple of words of thanks and appreciation are in order. First and foremost, I am deeply indebted to Ben Calhoun, Jerry Frenkil, Dejan Marković, and Bora Nikolić for their help and co-authorship of some of the chapters. In addition, a long list of people have helped in providing the basic slideware used in the text, and in reviewing the
earlier drafts of the book. Special gratitude goes to a number of folks who have shaped the low-power design technology world in a tremendous way—and as a result have contributed enormously to this book: Bob Brodersen, Anantha Chandrakasan, Tadahiro Kuroda, Takayasu Sakurai, Shekhar Borkar, and Vivek De. Working with them over the past decade(s) has been a great pleasure and a truly exciting experience!

**Low Power Design – Reference Books**
- Chapter 6, "Low-Voltage Technologies," by Kuroda and Sakurai.
- Chapter 8, "Techniques for Leakage Power Reduction," by De, et al.

**Low Power Design – Special References**
- Proceedings of the ISLPED Conference (starting 1984)
- Proceedings of ISSCC, VLSI Symposium, ESSCIRC, A-SSCC, DAC, ASPDAC, DATE, ICCAD conferences

I personally had a wonderful and truly enlightening time putting this material together while traversing Europe during my sabbatical in the spring of 2007. I hope you will enjoy it as well.

Jan M. Rabaey, Berkeley, CA
Contents

1 Introduction ................................................................................................................. 1

2 Nanometer Transistors and Their Models .............................................................. 25

3 Power and Energy Basics ......................................................................................... 53

4 Optimizing Power @ Design Time: Circuit-Level Techniques .............................. 77

5 Optimizing Power @ Design Time – Architecture, Algorithms, and Systems. ....... 113

6 Optimizing Power @ Design Time – Interconnect and Clocks ............................... 151

7 Optimizing Power @ Design Time – Memory ......................................................... 183

8 Optimizing Power @ Standby – Circuits and Systems ............................................. 207

9 Optimizing Power @ Standby – Memory ................................................................. 233

10 Optimizing Power @ Runtime: Circuits and Systems ............................................ 249

11 Ultra Low Power/Voltage Design ......................................................................... 289

12 Low Power Design Methodologies and Flows. .................................................... 317

13 Summary and Perspectives ..................................................................................... 345

Index ............................................................................................................................. 357
Chapter 1
Introduction

Slide 1.1
In this chapter we discuss why power and energy consumption has become one of the main (if not the main) design concerns in today’s complex digital integrated circuits. We first analyze the different application domains and evaluate how each has its own specific concerns and requirements, from a power perspective soon. Most projections into the future show that these concerns most likely will not go away. In fact, everything seems to indicate that they will even aggravate. Next, we evaluate technology trends – in the idle hope that technology scaling may help to address some of these problems. Unfortunately, CMOS scaling only seems to make the problem worse. Hence, design solutions will be the primary mechanism in keeping energy/power consumption in control or within bounds. Identifying the central design themes and technologies, and finding ways to apply them in a structured and methodological fashion, is the main purpose of this book. For quite some time, low-power design consisted of a collection of ad hoc techniques. Applying those techniques successfully on a broad range of applications and without too much “manual” intervention requires close integration in the traditional design flows. Over the past decade, much progress in this direction was made. Yet, the gap between low-power design technology and methodology remains.

Slide 1.2
There are many reasons why designers and application developers worry about power dissipation. One concern that has come consistently to the foreground in recent years is the need for “green” electronics. While the power dissipation of electronic components until recently was only a small fraction of the overall electrical power budget, this picture has changed substantially in the last few decades. The pervasive use of desktops and laptops has made its mark in both the office and home environments. Standby power of electronic consumer components and set-up boxes is rising rapidly such that at the time of writing this book their power drain is becoming equivalent to...
Why Worry About Power?

The Tongue-in-Cheek Answer

- Total energy of Milky Way galaxy: $10^{59}$ J
- Minimum switching energy for digital gate (1 electron@100 mV): $1.6 \times 10^{-20}$ J (limited by thermal noise)
- Upper bound on number of digital operations: $6 \times 10^{78}$
- Operations/year performed by 1 billion 100 MIPS computers: $3 \times 10^{24}$
- Entire energy might be consumed in 180 years, assuming a doubling of computational requirements every year (Moore’s Law).

...that of a decent-size fridge. Electronics are becoming a sizable fraction of the power budget of a modern automobile. These trends will only become more pronounced in the coming decade(s).

In this slide, the growing importance of electronics as part of the power budget is brought home with a “tongue-in-cheek” extrapolation. If Moore’s law would continue unabated in the future and the computational needs would keep on doubling every year, the total energy of our galaxy would be exhausted in the relatively low time span of 180 years (even if we assume that every digital operation is performed at its lowest possible level). However, as Gordon Moore himself stated in his keynote address at the 2001 ISSCC conference, “No exponential is forever”, adding quickly thereafter, “… but forever can be delayed”.

Power: The Dominant Design Constraint (1)

Cost of large data centers solely determined by power bill …

- 800 Millions of Personal Computers worldwide (Year 2000)
- Assumed to consume 0.16 Tera (10¹⁵) kWh per year
- Equivalent to 25 nuclear power plants
- Over 1 Giga kWh per year just for cooling

(Ref. Bar-Cohen et al., 2000)

The subsequent slide sets evaluate the power need and trends for a number of dominant application areas of digital integrated circuits. First, the domains of computation and communication infrastructure are discussed. The advent of the Internet, combined with ubiquitous access to the network using both wired and wireless interfaces, has dramatically changed the nature of computing. Today massive data storage and computing centers operated by large companies at a number of centralized locations have absorbed a huge amount of the worldwide computational loads of both corporations and individuals. And this trend is not showing any signs of slowing down, as new server farms are being brought online at a staggering rate. Yet, this centralization comes at a price. The “computational density” of such a center, and hence the power usage, is substantial. To quote Luis Barosso from Google (a company which is one of the most prolific promoters of the remote-computation concept), the cost of a data center is determined
Introduction

solely by the monthly power bill, not by the cost of hardware or maintenance. This bill results from both the power dissipation in the electronic systems and the cost of removing the dissipated heat—that is, air conditioning. This explains why most data centers are now implanted at carefully chosen locations where power is easily available and effective cooling techniques are present (such as in the proximity of major rivers—in an eerie similarity to nuclear plants).

While data centers represent a major fraction of the power consumed in the computation and communication infrastructure, other components should not be ignored. The fast routers that zip the data around the world, as well as the wireless base stations (access points) which allow us to connect wirelessly to the network, offer major power challenges as well. Owing to their location, the availability of power and the effectiveness of cooling techniques are often limited. Finally, the distributed computing and communication infrastructure cannot be ignored either; the wired and wireless data routers in the office, plant, or home, the back-office computing servers and the desktop computers add up to a sizable power budget as well. A large fraction of the air conditioning bill in offices is due to the ever-growing computational infrastructure.

Slide 1.4
It is worth spending some time on the cooling issue. A typical computing server rack in a server farm can consume up to 20 kW. With the racks in a farm easily numbering over one hundred, power dissipation can top the 2 MW (all of which is transformed into heat). The design of the air conditioning system and the flow of air through the room and the racks is quite complicated and requires extensive modeling and analysis. The impact of an ill-designed system can be major (i.e., dramatic failure), or more subtle. In one such data center design, cool air is brought in from the floor and is gradually heated while it rises through the blades (boards) in the rack. This leads to a temperature gradient, which may mean that processors closer to the floor operate faster than the ones on the top! Even with the best air-cooling design practices, predicting the overall dynamics of the center can be hard and can lead to under-cooling. Sometimes some quick improvised fixes are the only rescue, as witnessed in these ironic pictures, provided by Roger Schmidt, a distinguished engineer at IBM and a leading expert in the engineering and engineering management of the thermal design of large-scale IBM computers.

Slide 1.5
While temperature gradients over racks can lead to performance variations, the same is true for the advanced high-performance processors of today. In the past die sizes were small enough, and activity over the die was quite uniform. This translated into a flat temperature profile at the surface of the die. With the advent of Systems-on-a-Chip (SoC), more and more diverse
functionality is integrated in close proximity, very often with very different workloads and activity profiles. For instance, most high-performance microprocessors (or multi-core processors) integrate multiple levels of cache memories on the die, just next to the high-performance computing engine(s). As the data path of the processor is clocked at the highest speed and is kept busy almost 100% of the time, its power dissipation is substantially higher than that of the cache memories.

This results in the creation of hot spots and temperature gradients over the die. This may impact the long-term reliability of the part and complicate the verification of the processor. Execution speed and propagation delay are indeed strongly dependent on temperature. With temperature gradients over the die (which may change dynamically depending upon the operation modes of the processors), simulation can now not be performed for a single temperature, as was the common practice.

Correct for this imbalance, a complex package has to be constructed, which allows for the heat to spread over a wider area, thus improving the heat removal process. In high-performance

**Slide 1.6**
The existence of these thermal gradients is perfectly illustrated in this slide, which plots the temperature map of the IBM PowerPC 4 (a late 1990s microprocessor). A temperature difference of over 20°C can be observed between the processor core and the cache memory. Even more staggering, the heat generation at the hot spot (the data pipeline) equals almost 140 W/cm². This is 3.6 times the heat removal capacity of the chip cooling system.
components, packaging cost has become an important (if not dominating) fraction of the total cost. Techniques that help to mitigate the packaging problems either by reducing the gradients or by reducing the power density of selected sub-systems are hence essential. Structured low-power design methodologies, as advocated in this book, do just that.

Slide 1.7
The second reason why design for low power/energy has become so important is the emergence of mobile electronics. While mobile consumer electronics has been around for a while (FM radios, portable CD players), it is the simultaneous success of portable laptops and digital cellular phones that has driven the quest for low-energy computing and communication. In a battery-operated device, the available energy is fixed, and the rate of power consumption determines the lifetime of the battery (for non-rechargeables) or the time between recharges. Size, aspect ratio, and weight are typically set by the application or the intended device. The allowable battery size of a cellular phone typically is set to at most 4-5 cm³, as dictated by user acceptance. Given a particular battery technology, the expected operational time of the device – cell phone users today expect multiple days of standby time and 4-5 h of talk time – in between recharges sets an upper bound on the power dissipation for the different operational modes. This in turn determines what functionality can be supported by the device, unless breakthroughs in low-power design can be accomplished. For instance, the average power dissipation limit of a cell phone is approximately 3 W, dictated by today’s battery technologies. This in turn dictates whether your phone will be able to support digital video broadcasting, MP3 functionality, and 3G cellular and WIFI interconnectivity.

Slide 1.8
From this perspective, it is worthwhile to classify consumer and computing devices into a number of categories, based on their energy needs and hence functionality. In the “ambient intelligent” home of the future (a term coined by Fred Boekhorst from Philips in his ISSCC keynote in 2002), we may identify three styles of components. First, we have the “Watt nodes” (P > 1 W). These are nodes connected to the power grid, offering computational capacity of around 1 GOPS and performing functions such as computing and data serving, as well as routing and wireless access. The availability of energy, and hence computational prowess, makes them the ideal home for advanced media processing, data manipulation, and user interfaces.

The second tier of devices is called the “Milliwatt nodes” (1 mW < P < 1 W). Operating at a couple of MOPS, these represent mobile, untethered devices such as PDAs, communication devices (connecting to WANs and LANs), and wireless displays. These components are battery-powered and fall into the scope of devices discussed in the previous slide.
The “Microwatt nodes” represent the final category (P < 1 mW). Their function is to add awareness to the network, providing sensing functionality (temperature, presence, motion, etc.), and to transmit that data to the more capable nodes. The 1 KOPS computational capability severely limits their functionality. Given that a typical home may contain a huge number of these nodes, they have to be energy self-contained or powered using energy scavenging. Their very low power levels enable the atter. More information

---

**Slide 1.9**

From the above discussion, obviously the question arises: “Where is battery technology heading?” As already observed in Slide 1.7, battery capacity (i.e., the amount of energy that can be stored and delivered for a given battery volume) doubles approximately every 10 years. This represents an improvement of 3–7% every year (the slope tends to vary based on the introduction of new technologies). This growth curve lags substantially behind Moore’s law, which indicates a doubling in computational complexity every 18 months. The challenge with battery technology is that chemical processes are the underlying force, and improvements in capacity are often related to new chemicals or electrode materials. These are hard to come by. Also, the manufacturing processes for every new material take a long time to develop. Yet, an analysis of the available chemicals seems to show some huge potential. The energy density

---

**Battery Storage a Limiting Factor**

- Basic technology has evolved little
  - store energy using a chemical reaction
- Battery capacity increases between 3% and 7% per year (doubled during the 1990s, relatively flat before that)
- Energy density/size and safe handling are limiting factors

<table>
<thead>
<tr>
<th>Energy density of material</th>
<th>kWh/kg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gasoline</td>
<td>14</td>
</tr>
<tr>
<td>Lead-acid</td>
<td>0.04</td>
</tr>
<tr>
<td>Li polymer</td>
<td>0.15</td>
</tr>
</tbody>
</table>

of alcohol or gasoline is approximately two orders of magnitude higher than that of lithium-polymer. Unfortunately, concerns about the effective and safe handling of these substances make it hard to exploit them in small form factors.

Slide 1.10
The historical trends in battery capacity actually vary quite a bit. Up to the 1980s, very little or even no progress was made – there was actually little incentive to do so, as the scope of application was quite limited. Flash lights were probably the driving application. In the 1990s, mobile applications took off. Intensive research combined with advanced manufacturing strategies changed the slope substantially, improving the capacity by a factor of four in almost a decade. Unfortunately, the process has stalled somewhat since the beginning of the 21st century. A major improvement in battery capacity can only be achieved by the introduction of new battery chemicals. It should also be observed that the capacity of a battery (that is the energy that can be extracted from it) also depends upon its discharge profile. Draining a battery slowly will deliver more energy than flash discharge. It is hence worthwhile to match the battery structure to the application at hand.

Slide 1.11
The fact that the energy delivery capacity of a battery is ultimately determined by the basic chemical properties of the materials involved is clearly illustrated in this slide. In the 1990s, the capacity of Lithium-ion batteries improved substantially. This was mostly due to better engineering; improved electrode structures, better charging technology, and advanced battery system design. This ultimately saturated as the intrinsic maximum potential of the material is being approached. Today, progress in Lithium-ion battery technology has stalled, and little improvement is foreseen in the future.
Slide 1.12
The bottom line from the presented trends is that only a dramatic change in chemicals will lead to substantial increase in battery capacity. The opportunity is clearly there. For instance, hydrogen has an energy density 4–8 times that of Lithium-ion. It is no surprise that hydrogen fuel cells are currently under serious consideration for the powering of electrical or hybrid cars. The oxidation of hydrogen produces water and electrical current as output. Fuels such as alcohol, methanol, or gasoline are even better. The challenge with these materials is to maintain the efficiency in small form factors while maintaining safety and reliability.

Slide 1.13
It should be of no surprise that research in this area is intensive and that major companies as well as start-ups are vying for a piece of the potentially huge cash pot. Success so far has been few and far between. Toshiba, for instance, has introduced a number of methanol fuel cells, promising to extend the operational time of your cell phone to 1000 h (i.e., 40 days!). Other companies actively exploring the fuel cell option are NEC and IBM. Yet, the technology still has to find its way into the markets. Long-term efficiency, safety, and usage models are questionable. Other candidates such as solid oxygen fuel cells (also called ceramic fuel cells) are waiting behind the curtain. If any one of these becomes successful, it could change the energy equation for mobiles substantially.
Another interesting new entry in the battery field is the “micro battery”. Using technologies inherited from thin-film and semiconductor manufacturing, battery anodes and cathodes are printed on the substrate, and micromachined encapsulations are used to contain the chemicals. In this way, it is possible to print batteries on printed circuit boards (PCBs), or even embed them into integrated circuits. While the capacity of these circuits will never be large, micro batteries can serve perfectly well as backup batteries or as energy storage devices in sensor nodes. The design of the battery involves trading off between current delivery capability (number of electrodes) and capacity (volume occupied by the chemicals). This technology is clearly still in its infancy but could occupy some interesting niche in the years to come.

As a summary of the above discussions, it is worthwhile ordering the various energy storage technologies for mobile nodes based on their capacity (expressed in J/cm³). Another useful metric is the average current that can be delivered over the time span of a year by a 1 cm³ battery (μW/cm³/year), which provides a measure of the longevity of the battery technology for a particular application.

Miniature fuel cells clearly provide the highest capacity. In their currently best incarnation, they are approximately three times more efficient than the best rechargeable (secondary) batteries. Yet, the advantage over non-rechargeables (such as alkaline) is at most 25%.

One alternative strategy for the temporary storage of energy was not discussed so far: the capacitor. The ordinary capacitor constructed from high-quality dielectrics has the advantage of simplicity, reliability, and longevity. At the same time, its energy density is limited. One technology
that attempts to bridge the gap between capacitor and battery is the so-called supercapacitor or ultracapacitor, which is an electrochemical capacitor that has an unusually high energy density when compared to common capacitors, yet substantially lower than that of rechargeable batteries. A major advantage of (ultra)capacitors is the instantaneous availability of a high discharge current, which makes them very attractive for bursty applications. It is expected that new materials such as carbon nanotubes, carbon aerogels, and conductive polymers may substantially increase the capacity of supercapacitors in the years to come.

**Slide 1.16**
The third and final motivation behind “ultra low power” design is the emergence of a new class of frontier applications, called “zero-power electronics” or “disappearing electronics” (microwatt nodes in the Boekhorst classification). The continuing miniaturization of computing and communication components, enabled by semiconductor scaling, allows for the development of tiny wireless sensor nodes, often called motes. With sizes in the range of cubic centimeters or less, these devices can be integrated into the daily-living environment, offering a wide range of sensing and monitoring capabilities. By providing spatial and temporal information about, for instance, the environmental conditions in a room, more efficient and more effective conditioning of the room is enabled. The integrated format and the low cost make it possible to deploy large or even huge numbers of these motes. These emerging “wireless sensor networks (WSN)” have made some major inroads since their inception in the late 1990s. Energy is one of the main hurdles to be overcome, if the WSN paradigm is to be successful. Given the large number of nodes in a network, regular battery replacement is economically and practically out of question. Hence, nodes should in principle be energy self-contained for the lifetime of the application (which can be tens of years). Hence, a node should be able to operate continuously on a single battery charge, or should be capable of replenishing its energy supply by energy-scavenging techniques. As both energy storage and scavenging capacities are proportional to volume and the node size is limited, ultra low-power design is absolutely essential. In the “PicoRadio” project, launched by the author in 1998, it was determined that the average power dissipation of the node could not be larger than 100 $\mu$W.

**Slide 1.17**
Since the inception of the WSN concept, much progress was made in reducing the size, cost, and power dissipation of the mote. First-generation nodes were constructed from off-the-shelf components, combining generic microcontrollers, simple wireless transceivers with little power
optimization, and standard sensors. The resulting motes were at least one or two orders of magnitude better in every aspect (size, cost, power).

Since then, research in miniature low-power electronics has blossomed, and has produced spectacular results. Advanced packaging technologies, introduction of novel devices (sensors, passives, and antennas), ultra low-voltage design, and intelligent power management have produced motes that are close to meeting all the stated goals. The impact of these innovations goes beyond the world of wireless sensor networks and can equally be felt in areas such as implantable devices for health monitoring or smart cards.

**Slide 1.18**

Even more, progress in ultra low-power design and extreme miniaturization may enable the emergence of a number of applications that otherwise would be completely impossible. A couple of examples may help to illustrate this. Dense networks of sensor nodes deployed on a broad surface may lead to "artificial skin", sensitive to touch, stress, pressure, or fatigue. Obvious applications of such networks would be intelligent plane wings, novel user interfaces, and improved robots. Embedding multiple sensors into objects may lead to smart objects such as intelligent tires that sense the condition of the road and adjust the driving behavior accordingly. The concept of "inject-able" health diagnostic, monitoring, and, eventually, surgery devices was suggested in the science fiction world in the 1960s (for instance, in the notorious "Fantastic Voyage" by Isaac Asimov), but it may not be fiction after all. Yet, bringing each of these applications into reality will require power and size reduction by another order of magnitude (if not two). The cubic-centimeter nodes of today should be reduced to true "dust" size...
(i.e., cubic millimeter). This provides a true motivation for further exploration of the absolute boundaries of ultra low-power design, which is the topic of Chapter 11 in this book.

Slide 1.19

Energy scavenging is an essential component for the success of microwatt nodes. The idea is to transform the physical energy present in various sources in the environment into electrical power. Examples of the former are temperature or pressure gradients, light, acceleration, and kinetic and electromagnetic energy. In recent years, researchers both in academics and in industry have spent substantial efforts in cataloguing and metricizing the effectiveness of the various scavenging technologies [Roundy03, Paradiso05]. The efficiency of an energy harvester is best expressed by the average power provided by a scavenger of 1 cm³, operating under various conditions. Just like with batteries, scavenging efficiency is linearly proportional to volume (or, as in the case for solar cells, to surface area).

From the table presented in the slide, it is clear that light (captured by photovoltaic cells) is by far the most efficient source of energy, especially in outdoors conditions. A power output of up to 15 mW/cm² can be obtained. Unfortunately, this drops by two or three orders of magnitudes when operated in ambient indoor conditions. Other promising sources of energy that are ubiquitously available are vibration, wind, and temperature and pressure gradients. The interested reader can refer to the above-mentioned reference works for more information. The main takeaway is that average power levels of around 100 W/cm³ are attainable in many practical situations.

The discussion so far has not included some other sources of energy, magnetic and electromagnetic, that are prime targets for scavenging. Putting moving coils in a magnetic field (or having a variable magnetic field) induces current in a coil. Similarly, an antenna can capture the energy beamed at it in the form of an electromagnetic wave. This concept is used effectively for the powering of passive RF-IDs. None of these energy sources occurs naturally though, and an “energy transmitter” has to be provided. Issues such as the impact on health should be considered, if large power levels are required. Also, the overall efficiency of these approaches is quite limited.
This introduction has so far focused on the various application domains of microelectronics and their power needs and constraints. In the subsequent slides, we will discuss the energy and power trends from a technology perspective, looking back at past evolutions and projecting future developments. Before doing so, one more side note is probably useful. To put the energy efficiency of microelectronic systems into perspective, it is worth comparing them with other “computational engines”, in this case biological machinery (i.e., the brain).

The average power consumption of an average human brain approximately equals $20 \text{ W}$, which is approximately $20\%$ of the total power dissipation of the body. This fraction is quite high, especially when considering that the brain represents only $2\%$ of the total body mass — in fact, the ratio of power to the brain versus the total body power is a telling indicator of where the being stands on the evolutionary ladder. Again considering the average brain size ($1.33 \text{ dm}^3$), this leads to a power consumption of $15 \text{ mW/cm}^3$ — similar to what could be provided by $1 \text{ cm}^2$ of solar cells. Active neurons only represent a small fraction of this volume ($4\%$) — most of the rest is occupied by blood vessels, which transport energy in and heat out of the brain, and the dense interconnecting network.

Judging the energy efficiency of the brain is a totally different matter, though. Comparing the “computational complexity” of a neuron with that of a digital gate or a processor is extremely hard, if not irrelevant. The brain contains on the average 70 million neurons per cubic centimeter, each of which performs complex non-linear processing. For the interested readers, a great analysis of and comparison between electronic and neurological computing is offered in the best-selling book by Ray Kurzweil, “The Singularity Is Near.”
Power Versus Energy

- Power in high-performance systems
  - Heat removal
  - Peak power and its impact on power delivery networks
- Energy in portable systems
  - Battery life
- Energy/power in “zero-power systems”
  - Energy-scavenging and storage capabilities
- Dynamic (energy) vs. static (power) consumption
  - Determined by operation modes

Slide 1.21
Before discussing trends, some words about useful metrics are necessary (more details to follow in Chapter 3). So far, we have used the terms power and energy quite interchangeably. Yet, each has its specific role depending upon the phenomena that are being addressed or the constraints of the application at hand. Average power dissipation is the prominent parameter when studying heat-removal and packaging concerns of high-performance processors. Peak power dissipation, on the other hand, is the parameter to watch when designing the complex power supply delivery networks for integrated circuits and systems.

When designing mobile devices or sensor network nodes, the type of energy source determines which property is the most essential. In a battery-powered system, the energy supply is finite, and hence energy minimization is crucial. On the other hand, the designer of an energy-scavenging system has to ensure that the average power consumed is smaller than the average power provided by the scavenger.

Finally, dividing power dissipation into dynamic (proportional to activity) and static (independent of activity) is crucial in the design of power management systems exploiting the operational modes of the system. We will see later that the reality here is quite complex and that a careful balancing between the two is one of the subtleties of advanced low-power design.

Slide 1.22
While concerns about power density may seem quite recent to most designers, the issue has surfaced a number of times in the design of (electrical) engineering systems before. Obviously, heat removal was and is a prime concern in many thermodynamic systems. In the electronics world, power dissipation, and consequent high temperatures, was a main cause of unreliability in vacuum-tube computers. While bipolar computer design offered prime performance, exceeding what could be delivered by MOS implementations at that time, power density and the ensuing reliability concerns limited the amount of integration that could be obtained. The same happened with pure NMOS logic — the static current inherent in non-complimentary logic families ultimately caused semiconductor manufacturers to switch to CMOS, even though this meant an increased process complexity and a loss in performance. When CMOS was adopted as the technology-of-choice in the mid 1980s, many felt that the power problem had been dealt with effectively, and that CMOS design would enjoy a relatively trouble-free run to ever higher performance. Unfortunately, it was not to be. Already in the early 1990s, the ever-increasing clock frequencies and the emergence of new application domains brought power back to the foreground.

The charts in this slide document how the increases in heat flux in bipolar and CMOS systems mirror each other, only offset by about a decade. They make the interesting point that exponentials are hard to get around. New technologies create a fixed offset, but the exponential increases in
complexity – so essential to the success of the semiconductor industry – conspire to eliminate that in the shortest possible time.

Slide 1.23
The power trends of the past are best observed by empirically sampling the leading processor designs over the years (as embodied by publications in ISSCC, the leading conference in the field) [Courtesy of T. Kuroda and T. Sakurai]. Plotting the power dissipations of microprocessors and DSPs as a function of time reveals some interesting trends. Up to the mid 1990s, the average power dissipation of a processor rose by a factor of four every three years. At that time, a discontinuity occurred. A major slowdown in the rise of power dissipation of leading-edge processors is apparent (to approximately a factor of 1.4 every three years). Simultaneously, another downward vector emerged: owing to the introduction of mobile devices, a market for lower-power lower-performance processors was materializing. One obviously wonders about the
discontinuity around 1995, the answer for which is quite simple: Owing to both power and reliability concerns, the semiconductor industry finally abandoned the idea of a supply voltage fixed at 5 V (the “fixed-voltage scaling model”), and started scaling supply voltages in correspondence with successive process nodes. Fixed-voltage scaling was an attractive proposition, as it simplified the interfacing between different components and parts, yet the power cost became unattainable. Reasoning about the precise value of the slope factors is somewhat simplified when studying power density rather than total power, as the former is independent of the actual die size.

\[
p = CV_{DD}^2 f
\]

then evolves as

\[
k_p = k \times 1 \times k^2 = k^3.
\]

Consider now the situation after 1995. Under the full-scaling mode, supply voltages were scaled in proportion to the minimum feature size of the technology. Also at that time, short-channel device effects such as velocity saturation (again see Chapter 2) were becoming important, causing the saturation current (i.e., the maximum discharge current) to scale approximately as \(k^{-0.3}\), leading to a slowdown in the clock frequency increase to \(k^{1.7}\). For the power density, this means that

\[
p = CV_{DD}^2 f
\]

now scales as

\[
k_p = k \times (1/k)^2 \times k^{1.7} = k^{0.7},
\]
which corresponds with the empirical data. Even though this means that power density is still increasing, a major slowdown is observed. This definitely is welcome news.

**Slide 1.25**

To illustrate the fact that the full scaling model was truly adopted starting around the 0.65 µm CMOS technology node, this slide plots the range of supply voltages that were (are) typically used for every generation. Up to the early 1990s, supply voltages were pretty much fixed at 5 V, dropping for the first time to 3.3 V for the 0.35 µm generation. Since then, supply voltages have by and large followed the minimum feature size.

For instance, the nominal supply voltage for the 180 nm processor equals 1.8 V; for 130 nm it is 1.3 V; and so on. Unfortunately, this trend is gradually changing for the worse again, upsetting the subtle balance between performance and power density, as will become clear in the following slides.

**Slide 1.26**

By the end of the 20th century, new storm clouds were gathering on the horizon. The then prevalent scaling model made the assumption that a certain ratio between supply voltage and threshold voltage is maintained. If not, a substantial degradation in maximum clock speed (which was generally equated to system performance) results, a penalty that the designers of that time were not willing to accept. The only plausible solution to address this challenge was to maintain a constant ratio by scaling the threshold voltages as well. This, however, posed a whole new problem. As we will discuss in detail in later chapters, the off-current of a MOS transistor (i.e., the current when the gate-source voltage is set to zero) increases exponentially with a reduction in the threshold voltage. Suddenly, static power dissipation — a problem that had gone away with the introduction of CMOS — became a forefront issue again. Projections indicated that, if left unattended, static power dissipation would overtake dynamic power sometime in the mid to late 2000s.
of techniques to keep leakage power within bounds. These will be described in detail in later chapters.
Yet, static power has become a sizable fraction of the overall power budget of today’s integrated circuits, and most indicators suggest that this problem will only get more severe with time.

### Slide 1.27

The problem was such that one was afraid that leakage might become the undoing of Moore’s law. While the International Technology Roadmap for Semiconductors (ITRS) was prescribing a further slowdown in the average power dissipation (by a factor of approximately 1.1 every three years), static power dissipation potentially was registering a very rapid increase instead.

Fortunately, designers have risen to the challenge and have developed a range

### Slide 1.28

There exist very compelling reasons why a further increase in power density should be avoided at all costs. As shown in an earlier slide for the PowerPC 4, power densities on chips can become excessive and lead to degradation or failure, unless extremely expensive packaging techniques are used. To drive the point home, power density levels of some well-known processors are compared to general-world examples, such as hot plates, nuclear reactors, rocket nozzles, or even the sun’s surface. Surprisingly, high-performance ICs are not that far off from some of these extreme heat sources! Classic wisdom dictates that power densities above 150 W/cm² should be avoided for the majority of designs, unless the highest performance is an absolute must and cost is not an issue.
Yet, they help to present the dire consequences of what would happen if we do not act and identify areas where intensive research is necessary.

The first observation is that computing density (defined as the number of computations per unit area and time) continues to increase at a rate of \( k^3 \). This assumes that clock frequencies continue to rise linearly, which is probably doubtful considering the other trends. The dynamic power density is projected to accelerate anew (from \( k^{0.7} \) to \( k^{1.9} \)). This is particularly bad news, and is mainly due to a continuing increase in clock speed combined with a slowdown in supply voltage scaling (as is plotted in the next slide). The latter is a necessity if static power dissipation is to be kept somewhat within bounds. Yet, even when accounting for a slowdown in supply- and threshold-voltage scaling, and assuming some technology and device breakthroughs such as full-depleted SOI (FD-SOI) and dual-gate transistors, static power density still grows at a rate of \( k^{2.7} \). This means that leakage power if left unattended will come to dominate the power budget of most integrated circuits.

Most probably, the above scenario will not play out. Already clock frequencies of leading processors have saturated, and architectural innovations such as multi-core processing are used to maintain the expected increase in overall performance. The obtained slack can be used to reduce either dynamic or static power, or both. In addition, the heterogeneous composition of most SoCs means that different scenarios apply to various parts of the chip.
RF, and passive components). Managing the different scaling trajectories of each of these is the task of the “power management”, which is the topic of Chapter 10.

**Slide 1.30**

To emphasize the last argument, this slide plots the power budget of a number of microprocessors and DSPs from different companies. The distribution of power over different resources, such as computation, memory, clock, and interconnect, varies wildly. Looking forward, this trend will only accelerate. Complex SoCs for communication, media processing, and computing contain a wide variety of components with vastly different performance and activity profiles (including mixed signal).

**Slide 1.31**

Leakage concerns put a lower bound on the threshold voltages. Barring the (improbable) event that a leakage-resistant logic family suddenly emerges, threshold voltages are unlikely to drop below 0.25 V. This severely impedes further scaling of the supply voltages. The ITRS (low-power scenario) optimistically projects that supply voltages will be reduced to 0.5 V. Getting there presents a severe challenge though. It is even doubtful whether reliable memories are feasible at all at these low voltage levels.

Innovations at the device and circuit level may come somewhat to the rescue. Transistors with higher mobility are currently researched at a number of institutions. Higher current drive means
that performance can be maintained even at a low $V_{DD}/V_{TH}$ ratio. Transistors with a sharp transition between the on and off states are another opportunity. In later chapters, we will also explore how we can design reliable and efficient circuits, even at very low voltages.

**A 20 nm Scenario**

Assume $V_{DD} = 1.2$ V
- FO4 delay < 5 ps
- Assuming no architectural changes, digital circuits could be run at 30 GHz
- Leading to power density of 20 kW/cm² (??)

Reduce $V_{DD}$ to 0.6 V
- FO4 delay = 10 ps
- The clock frequency is lowered to 10 GHz
- Power density reduces to 5 kW/cm² (still way too high)

[Ref. S. Borkar, Intel]

limited to 10 GHz.

**A 20 nm Scenario (contd)**

Assume optimistically that we can design FETs (Dual-Gate, FinFet, or whatever) that operate at 1 kW/cm² for FO4 = 10 ps and $V_{DD} = 0.6$ V [Frank, Proc. IEEE, 3/01]

- For a 2cm x 2cm high-performance microprocessor die, this means 4 kW power dissipation
- If die power has to be limited to 200 W, only 5% of these devices can be switching at any time, assuming that nothing else dissipates power.

[Ref. S. Borkar, Intel]

This example clearly demonstrates that a drastic review of design strategies and computational architecture is necessary.

**Slide 1.32**

A simple example is often the best way to drive the arguments home. Assume a fictitious microprocessor with an architecture that is a direct transplant of current-generation processors. In a 20nm technology, clock speeds of up to 30 GHz are theoretically plausible if the supply voltage is kept unchanged at 1.2 V. The power density however goes through the roof, even when the supply voltage is reduced to 0.6 V, and the clock frequency

**Slide 1.33**

Let us be optimistic for a while, and assume the device innovations allow us to maintain the 10 GHz clock frequency, while reducing the power density by a factor of five. Still, a 4 cm² processor would consume 4 kW. Bringing this down to an acceptable 200 W requires that most of the devices not be switching 95% of the time, and also not leaking. A formidable challenge indeed!
An Era of Power-Limited Technology Scaling

**Technology innovations offer some relief**
- Devices that perform better at low voltage without leaking too much

**But also are adding major grief**
- Impact of increasing process variations and various failure mechanisms more pronounced in low-power design regime

**Most plausible scenario**
- Circuit- and system-level solutions essential to keep power/energy dissipation in check
- Slow down growth in computational density and use the obtained slack to control power density increase
- Introduce design techniques to operate circuits at nominal, not worst-case, conditions

Slide 1.34

In summary, this introductory chapter spells out the reasons why most of the innovators involved in the semiconductor industry believe that we have entered an era of power-limited scaling. This means power considerations are the primary factors determining how process, transistor, and interconnect parameters are scaled. This is a fundamental break with the past, where technology scaling was mostly guided by performance considerations. Furthermore, we do not believe that there is a "life-saving" transition — such as the one from bipolar to MOS — on its way soon. Novel devices that are currently in the lab phase hold some great promises, but only provide a limited amount of healing. In fact, the introduction of scaled devices adds an amount of suffering to the blessings (such as decreasing reliability and increasing variability). In the end, it is new design strategies and innovative computational architectures that will set the course. The main concepts underlying those will be treated in detail in the coming chapters.

---

**Some Useful References ...**

**Selected Keynote Presentations**


**Books and Book Chapters**

<table>
<thead>
<tr>
<th>Some Useful References (cntd)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Publications</strong></td>
</tr>
<tr>
<td>- S. Berk, numerous presentations over the past decade.</td>
</tr>
</tbody>
</table>
Nanometer Transistors and Their Models

Jan M. Rabaey

Slide 2.1
As has become apparent in Chapter 1, the behavior of the MOS transistor, when scaled into the sub-100 nm regime, is having a large impact on how and where power is consumed in the next-generation integrated circuits. Hence, any discussion on low-power design should start with a good understanding of the deep submicron MOS transistor, and an analysis of its future trends. In addition, the availability of adequate models, for both manual and computer-aided analysis, is essential. As this book emphasizes optimization, simple yet accurate models that can serve in an automated (MATLAB-style) optimization framework are introduced.

Results in this and in the coming chapters are based on the Predictive MOS models, developed by UCB and the University of Arizona, as well as industrial models spanning from 180 nm down to 45 nm. Whenever possible, MATLAB code is made available on the web site of the book.

Slide 2.2
The chapter starts with a discussion of the nanometer transistor and its behavior. Special attention is devoted to the leakage behavior of the transistor. The increasing influence of variability is analyzed next. At the end of the chapter, we evaluate some innovative devices that are emerging from the research labs and discuss their potential impact on low-power design technology.

Nanometer Transistors and Their Models

- Emerging devices in the sub-100 nm regime post challenges to low-power design
  - Leakage
  - Variability
  - Reliability
- Yet also offer some opportunities
  - Increased mobility
  - Improved control (?)
- State-of-the-art low-power design should build on and exploit these properties
  - Requires clear understanding and good models

Slide 2.3
Beyond the degeneration of the on/off behavior of the MOS transistor, mentioned in Chapter 1, sub-100 nm transistors also suffer from increased variability effects, due both to manufacturing artifacts and to physical limitations. Once the feature sizes of the process technology approach the dimensions of a molecule, it is obvious that some quantum effects start to play. In addition, the reduced dimensions make the devices also prone to reliability failures such as soft errors (single-event upsets) and time-dependent degradation.

While these issues affect every MOS circuit design, their impact is more pronounced in low-power designs. Reducing power dissipation often means reducing the operational signal-to-noise margins of the circuits (for instance, by lowering the supply voltage). Effects such as variation in performance and unreliability are more apparent under these conditions. It is fair to say that today’s low-power design is closely interwoven with design for variability or reliability. In this sense, low-power design often paves the way for the introduction of novel techniques that are later adopted by the general-purpose design community.

While it may seem that scaling MOS transistors down to tens of nanometers only brings bad karma, some emerging devices may actually help to reduce power density substantially in the future. Especially transistors with higher mobility, steeper sub-threshold slopes, better threshold control, and lower off-currents are attractive.

The Sub-100 nm Transistor

- **Velocity-saturated**
  - Linear dependence between $I_D$ and $V_{GS}$
- **Threshold voltage $V_{TH}$ strongly impacted by channel length $L$ and $V_{DS}$**
  - Reduced threshold control through body biasing
- **Leaky**
  - Sub-threshold leakage
  - Gate leakage
  -> Decreasing $I_{on}$ over $I_{off}$ ratio

Slide 2.4
From an operational perspective, the main characteristics of the sub-100 nm MOS transistors can be summarized as follows: a linear dependence exists between voltage and current (in the strong-inversion region); threshold is a function of channel length and operational voltages; and leakage (both sub-threshold and gate) plays a major role. Each of these issues is discussed in more detail in the following slides.
simple models of the past are inaccurate. To address this issue, we introduce some simplified transistor models of varying complexity and accuracy. The main goal in the context of this book is to provide the necessary tools to the circuit and systems designer to predict power and performance quickly.

Another important effect to be observed from the curves is the decrease in output resistance of the device in saturation.

**Slide 2.5**
The simulated $I_D$ versus $V_{DS}$ behavior of a 65 nm NMOS transistor clearly demonstrates the linear relationship between $I_D$ and $V_{GS}$ in the saturation region. This is a result of the well-known velocity-saturation effect, which started to impact CMOS transistors around the 250 nm technology generation. The main impact is a reduced current drive for a given gate voltage. Of course, this means that the

**Slide 2.6**
Probably the most accurate model, which still allows for fast analysis and requires only a restricted set of parameters, was introduced by Taur and Ning in 1998. One important parameter in this model is the critical electrical field $E_C$, which determines the onset of velocity saturation. The problem with this model is its highly non-linear nature, which makes it hard to use in optimization programs (and

$$I_{DSat} = v_{Sat} W C_{ox} \frac{(V_{GS} - V_{TH})^2}{(V_{GS} - V_{TH}) + E_c L}$$

- Good model, could be used in hand or MATLAB analysis

$$I_{DSat} = \frac{W \mu_{Sat} C_{ox}}{L} V_{DSat} (V_{GS} - V_{TH})$$

with

$$V_{DSat} = \frac{(V_{GS} - V_{TH}) E_c L}{(V_{GS} - V_{TH}) + E_c L}$$

[Ref: Taur-Ning, '98]
Slide 2.7
The “unified model” of the MOS transistor was introduced in [Rabaey03]. A single non-linear equation suffices to describe the transistor in the saturation and linear regions. The main simplification in this model is the assumption that velocity saturation occurs at a fixed voltage \( V_{DSat} \), independent of the value of \( V_{GS} \). The main advantages of the model are its elegance and simplicity. A total of only five parameters are needed to describe the transistor: \( k' \), \( V_{TH} \), \( V_{DSat} \), \( \lambda \) and \( \gamma \). Each of these can be empirically derived using curve-fitting with respect to the actual device plots. Observe that these parameters are purely empirical, and have no or little relation to traditional physical device parameters such as the channel-length modulation \( \lambda \).

Slide 2.8
Simplicity comes at a cost however. Comparing the I–V curves produced by the model to those of the actual devices (BSIM-4 SPICE model), a large discrepancy can be observed for intermediate values of \( V_{DS} \) (around \( V_{DSat} \)). When using the model for the derivation of propagation delays (performance) of a CMOS gate, accuracy in this section of the overall operation region is not that crucial. What is most important is that the values of current at the highest values of \( V_{DS} \) and \( V_{GS} \) are predicted correctly – as these predominantly determine the charge and discharge times of the output capacitor. Hence, the propagation delay error is only a couple of percents, which is only a small penalty for a major reduction in model complexity.
Alpha Power Law Model

- Alternate approach, useful for hand analysis of propagation delay
  \[ I_{DS} = \frac{W}{2L} \mu C_{ox} (V_{GS} - V_{TH})^\alpha \]

- Parameter \( \alpha \) is between 1 and 2.
- In 65–180 nm CMOS technology \( \alpha \approx 1.2–1.3 \)

- This is not a physical model
- Simply empirical:
  - Can fit (in minimum mean squares sense) to a variety of \( \alpha \)'s, \( V_{TH} \)
  - Need to find one with minimum square error – fitted \( V_{TH} \) can be different from physical

[Ref: Sakurai, JSSC’90]

Such as the minimum-mean square (MMS) are used. Be aware that these do not yield unique solutions and that it is up to the modeler to find the ones with the best fit.

Owing to its simplicity, the alpha model is the cornerstone of the optimization framework discussed in later chapters.

Output Resistance

- Drain current keeps increasing beyond the saturation point
- Slope in I–V characteristics caused by:
  - Channel-length modulation (CLM)
  - Drain-induced barrier lowering (DIBL).

- The simulations show approximately linear dependence of \( I_{DS} \) on \( V_{DS} \) in saturation (modeled by \( \lambda \) factor)

[Ref: BSIM 3v3 Manual]

Slide 2.9

Even simpler is the alpha model, introduced by Sakurai and Newton in 1990, which does not even attempt to approximate the actual I–V curves. The values of \( \alpha \) and \( V_{TH} \) are purely empirical, chosen such that the propagation delay of a digital gate, approximated by

\[ t_p = \frac{k(V_{PD} - V_{TM})}{(V_{PD} - V_{TM})^\beta}, \]

best resembles the propagation delay curves obtained from simulation. Typically, curve-fitting techniques

function of the drain voltage. DIBL primarily impacts leakage (as discussed later), yet its effect on output resistance is quite sizable as well. SCBE (Substrate Current Body Effect) only kicks in at voltages higher than the typical operation regime, and its impact is hence not that important.

Fortunately, the relationship between drain voltage and current proves to be approximately linear, and is adequately modeled with a single parameter \( \lambda \).
empirical approach is to derive $V_{TH}$ from the $I_D - V_{GS}$ plot by linearly extrapolating the current in the saturation region (see plot). The cross-point with the zero-axis is then defined as $V_{TH}$ (also called $V_{THZ}$). Another approach is the “constant-current” (CC) technique, which defines the threshold voltage as the point where the drain–source current drops below a fixed value ($I_{D0}$), scaled appropriately with respect to the (W/L) ratio of the transistor. The choice of $I_{D0}$ is however quite arbitrary. Hence, in this book we use the extrapolation technique, unless otherwise mentioned.

Slide 2.11
With the continuing reduction of the supply voltages, scaling of the threshold voltage is a necessity as well, as illustrated at length in Chapter 1. Defining the actual threshold voltage of a transistor is not simple, as many factors play and measurements may not be that straightforward. The “physics” definition of the threshold voltage is the value of $V_{GS}$ that causes strong inversion to occur underneath the gate. This is however impossible to measure. An often-used

Slide 2.12
The threshold voltage is unfortunately not a constant parameter, but is influenced by a number of operational parameters. The foremost is the body-bias or back-bias effect, where the fourth terminal of the transistor (the bulk or well voltage) serves as an extra control knob. The relationship between $V_{TH}$ and $V_{SB}$ is well-known, and requires the introduction of one extra device parameter, the body-effect parameter $\gamma$. Observe that body-biasing can be used either to increase (reverse bias) or to decrease (forward bias) the threshold voltage. The forward-biasing effect is limited in its scope, as the
source–bulk diode must remain in reverse-bias conditions (that is $V_{SB} > -0.6$ V). If not, current is directly injected into the body from the source, effectively killing the gain of the transistor. For the 130 nm technology, a 1 V change in $V_{SB}$ changes the threshold voltage by approximately 200 mV.

The beauty of the body-biasing effect is that it allows for a dynamic adjustment of the threshold voltage during operation, thus allowing for the compensation of process variations or a dynamic trade-off between performance and leakage.

### Slide 2.13
Regrettably, scaling of device technology is gradually eroding the body-biasing effect. With the doping levels in the channel increasing, changes in the bias voltage have little effect on the onset of strong inversion. This is clearly illustrated in this slide, which plots the impact of body biasing for three technology nodes.

Emerging technologies, such as fully-depleted SOI (in which the body of the transistor is floating), even do away completely with the body biasing. This development is quite unfortunate, as this parameter is one of the few parameters a designer can use to actively control leakage effects.

### Slide 2.14
Channel length is another parameter that influences the threshold voltage. For very short channels, the depletion regions of the drain (and source) junctions themselves deplete a sizable fraction of the channel. Turning the transistor on becomes easier, thus causing a reduction in the threshold voltage. To offset this effect, device engineers add some extra “halo implants”, which cause the threshold to peak around the nominal value of the channel length. While this is beneficial in general, it also increases the sensitivity of the threshold
voltage with respect to channel-length variations. For instance, it may happen that the channel lengths of a particular wafer batch are consistently below the nominal value. This causes the thresholds to be substantially below the expected value, leading to faster, but leaky, chips.

Slide 2.15
Designers vying for large threshold values with relatively small variations often size their transistors above the nominal channel length. This obviously comes at a penalty in area. The impact can be quite substantial. In a 90nm technology, leakage currents can be reduced by an order of magnitude by staying away from the minimum channel lengths. Just a 10% increase already reaps major benefits. This observation has not escaped the attention of designers of leakage-sensitive modules, such as SRAM memories.

Slide 2.16
Drain voltage is another variable that has a sizable impact on the threshold voltage. The DIBL effect was already mentioned in the context of the output resistance of short-channel devices. As the drain voltage increases, the depletion region of the junction between the drain and the channel increases in size and extends under the gate, effectively lowering the threshold voltage (this is a hugely simplified explanation, but it catches the main tenet). The most negative feature of DIBL effect is that it turns the threshold voltage into a signal-dependent variable. For all practical purposes, it is fair to assume that $V'_{DS}$ changes the threshold in a linear fashion, with $\lambda_d$ being the proportionality factor.
Slide 2.17
Quite a number of times in the introduction, we have alluded to the increasing effects of “leakage” currents in the nanometer MOS transistor. An ideal MOS transistor (at least from a digital perspective) should not have any currents flowing into the bulk (or well), should not conduct any current between drain and source when off, and should have an infinite gate resistance. As indicated in the accompanying slide, a number of effects are causing the contemporary devices to digress from this ideal model.

Leakage currents, flowing through the reverse-biased source–bulk and drain–bulk pn junctions, have always been present. Yet, the levels are so small that their effects could generally be ignored, except in circuitry that relies on charge storage such as DRAMs and dynamic logic.

The scaling of the minimum feature sizes has introduced some other leakage effects that are far more influential and exceed junction leakage currents by 3–5 orders of magnitude.

Most important are the sub-threshold drain–source and the gate leakage effects, which we will discuss in more detail.

Slide 2.18
In earlier slides, we have alluded to a relationship between the value of the threshold voltage $V_{TH}$ and (sub-threshold) leakage. When the gate voltage of a transistor is lowered below the threshold voltage, the transistor does not turn off instantaneously. In fact, the transistor enters the so-called “sub-threshold regime” (or weak inversion). In this operation mode, the drain–source current becomes an exponential function of $V_{GS}$. This is clearly observed from the $I_D-V_{GS}$ curves, if

The current is plotted on a logarithmic scale.
The exponential dependence is best explained by the fact that under these conditions the MOS transistor behaves as a bipolar device (nnp for an NMOS) with its base coupled to the gate through a capacitive divider. We know that for an ideal bipolar transistor, the base current relates to the base-emitter voltage as \( I_{CE} = \frac{I_{S}}{e^{V_{BE}/q} - 1} \) where \( k \) is the Boltzmann constant and \( T \) the absolute temperature. The so-called thermal voltage \( (kT/q) \) equals approximately 25 mV at room temperature. For an ideal bipolar transistor, every increase in \( V_{BE} \) by 60 mV \( = 25 \text{ mV} \times \ln(10) \) increases the collector current by a factor of 10!

In the weak inversion mode of the MOS transistor, the exponential is somewhat deteriorated by the capacitive coupling between gate and channel (base). Hence, the sub-threshold current is best modeled as \( I_{DS} = \frac{V_{GS}}{e^{V_{GS}/n} - 1} \) where \( n \) is the slope factor ranging around 1.4-1.5 for modern technologies. The net effect of this degradation is that, for the current to drop by one order of magnitude in the sub-threshold region, the reduction in \( V_{GS} \) needed is not of 60 mV, but more like 70–100 mV. Obviously, for an ideal switch we would hope that the current drops immediately to zero when \( V_{GS} \) is lowered below \( V_{TH} \).

**Slide 2.19**

The growing importance of sub-threshold leakage can now be understood. If the threshold voltage is set, for example, at 400 mV, the leakage current drops by five orders of magnitude between \( V_{GS} = V_{TH} \) and \( V_{GS} = 0 \) (assuming a sub-threshold swing of approximately 80 mV/decade). Assume now that the threshold voltage is scaled to 100 mV to maintain performance under reduced supply voltage conditions.

The leakage current at \( V_{GS} = 0 \) for this low-threshold transistor will be approximately four orders of magnitude higher than that for the high-threshold device, or the leakage current goes up exponentially with a linear reduction in threshold voltage. This serves as another example of the impact of exponential relations.
Sub-threshold Current

- Sub-threshold behavior can be modeled physically

\[ I_{ds} = \frac{2n\mu C_{ox}}{L} \frac{kT}{q} \left( \frac{V_{gs} - V_{th}}{q} \right)^n \left( 1 - e^{-\frac{V_{ds}}{q}} \right) = I_s e^{-\frac{V_{th}}{q}} \left( 1 - e^{-\frac{V_{ds}}{q}} \right) \]

where \( n \) is the slope factor (\( \geq 1 \), typically around 1.5) and \( I_s = \frac{2n\mu C_{ox}}{L} \frac{kT}{q} \left( \frac{V_{gs} - V_{th}}{q} \right)^n \)

- Very often expressed in base 10

\[ I_{ds} = I_s 10^{\frac{V_{gs} - V_{th}}{n} \left( 1 - 10^{-\frac{V_{ds}}{q}} \right)} \]

where \( S = n \left( \frac{kT}{q} \right) \ln(10) \), the sub-threshold swing, ranging between 60 mV and 100 mV

- \( =1 \) for \( V_{ds} > 100 \text{ mV} \)

Slide 2.20

Since sub-threshold leakage is playing such a dominant role in the nanometer design regime, it is quite essential to have good models available. One of the (few) advantages of the sub-threshold operational regime is that physical modeling is quite possible, and that the basic expressions of the drain current as a function of \( V_{GS} \) or \( V_{DS} \) can be easily derived.

Sub-threshold Current - Revisited

- Drain-Induced Barrier Lowering (DIBL)
  - Threshold reduces approximately linearly with \( V_{DS} \)

\[ V_{TH} = V_{TH0} - \lambda_d V_{DS} \]

- Body-Biasing Effect
  - Threshold reduces approximately linearly with \( V_{BS} \)

\[ V_{TH} = V_{TH0} - \gamma_v V_{BS} \]

Leading to:

\[ I_{DS} = I_s 10^{\frac{V_{gs} - V_{th} + \lambda_d V_{ds} + \gamma_v V_{bs}}{n} \left( 1 - 10^{-\frac{V_{ds}}{q}} \right)} \]

Leakage is an exponential function of drain and bulk voltages

These effects with the addition of two parameters: \( \lambda_d \) and \( \gamma_v \).

Slide 2.21

The simple model of the previous slide does not cover two effects that dynamically modulate the threshold voltage of the transistor: DIBL and body biasing. While these effects influence the strong-inversion operational mode of the transistor (as discussed earlier), their impact is felt far more in the sub-threshold mode owing to the exponential relation between drain current and threshold voltage. The current model is easily adjusted to include these effects with the addition of two parameters: \( \lambda_d \) and \( \gamma_v \).

Slide 2.22

Especially DIBL turns out to have a huge impact on the sub-threshold leakage of the nanometer CMOS transistor. Assume, for instance, an NMOS transistor in the off-mode (\( V_{GS} = 0 \)). The sub-threshold current of the transistor is now strongly dependent on the applied \( V_{DS} \). For instance, for the device characteristics shown in the slide, raising \( V_{DS} \) from 0.1 V to 1 V increases the leakage current by a factor of 10 (while in an ideal device it should stay approximately flat). This creates
both a challenge and an opportunity, as it means that leakage becomes strongly data-dependent. Leaving this unchecked may lead to substantial problems. At the same time, it offers the innovative designer an extra parameter to play with.

**Slide 2.23**

In addition, the current flowing through the drain in the off-state is influenced by the “gate-induced drain leakage” (GIDL) effect. While one would expect the drain current to drop continuously when reducing $V_G$ below $V_{TH}$ for a given drain voltage $V_D$, the inverse is actually true. Especially at negative values of $V_G$, an increase in drain current is observed. This is the result of a combination of effects such as band-to-band tunneling and trap-assisted tunneling. A high value of the electric field under the gate/drain overlap region (as occurring for low values of $V_G$ (0 V or lower) and high $V_D$) causes deep depletion and an effective thinning of the depletion width of the drain–well junction. This effectively leads to electron–hole pair creation and an accompanying drain-to-bulk current. The effect is proportional to the applied value of $V_{DG}$. The impact of GIDL is mostly felt in the off-state of the transistor with $V_{GS} = 0$. The upward bending of the drain current curve causes an effective increase of the leakage current.

It should be noted that the GIDL effect is substantially larger in NMOS than in PMOS transistors (by about two orders of magnitude). Also observe that the impact of GIDL is quite small for typical supply voltages, which are at 1.2 V or lower.
and causes a substantial increase in leakage current. For instance, increasing $V_{DS}$ from 0.1 to 1.0 V causes the drain current to increase by a factor of almost 8.

The GIDL effect can clearly be observed for values of $V_{GS}$ smaller than 0.1 V. However, even for $V_{DS}$ at a very high value of 2.5 V, the impact at $V_{GS} = 0$ is still ignorable. GIDL hence plays a minor role in most of today’s designs.

It is worth contemplating the overall picture that emerges from this. For a minimum-sized device in a low-leakage technology with a $V_{TH}$ around 0.35 V, the drain leakage hovers around 1 nA at room temperature. This amounts to a total leakage current of approximately 0.1 A for a design with a hundred million gates (or equivalent functions). This value increases substantially at higher temperatures (which is the standard operating condition), increases linearly with the device width, and rises exponentially with a reduction in threshold voltage. Designs with standby leakage currents of multiple Amperes are hence very plausible and real, unless care is taken to stop the bleeding.

**Slide 2.25**
While sub-threshold currents became an issue with the introduction of the 180 nm technology node, another leakage effect is gaining importance once technology scales below the 100 nm level – that is, gate leakage. One of the attractive properties of the MOS transistor has always been its very high (if not infinite) input resistance. In contrast, a finite base current is inherent to the structure of the bipolar transistor, making the device unattractive for usage in complex digital designs.

To maintain the current drive of the transistor while scaling its horizontal dimensions, general scaling theory prescribes that the gate oxide (SiO₂) thickness is scaled as well. Once however the oxide thickness becomes of the order of just a few molecules, some significant obstacles emerge. And this is exactly what happens with the sub-100 nm transistors, as is illustrated by the cross-section SEM picture of a 65 nm MOS transistor with an oxide thickness of 1.2 nm. It is clear that the oxide is barely a couple of molecules thick.
While posing some obvious limits on scaling, the very thin oxides also cause a reduction in the gate resistance of the transistor, as current starts to leak through the dielectric. This trend is clearly illustrated in the chart, which shows the evolution of the gate thickness and the gate leakage over various technology generations at Intel. From 180 nm to 90 nm, the gate leakage current increased by more than four orders of magnitude. The reasons behind the leveling and subsequent drop in subsequent generations will become clear in the following slides. Observe also that gate leakage increases strongly with temperature.

Unlike sub-threshold currents, which primarily cause an increase in standby power, gate currents threaten some fundamental concepts used in the design of MOS digital circuits.

**Slide 2.26**

Gate leakage finds its source in two different mechanisms: Fowler–Nordheim (FN) tunneling, and direct-oxide tunneling. FN tunneling is an effect that has been effectively used in the design of non-volatile memories, and is already quite substantial for oxide thicknesses larger than 6 nm. Its onset requires high electric-field strengths, though. With reducing oxide thicknesses, tunneling starts to occur at far lower field strengths. The dominant effect under these conditions is direct-oxide tunneling.
This trend clearly threatens the further scaling of MOS technology, unless some innovative process technology solutions emerge.

A first approach to address the challenge is to stop or slow down the scaling of the oxide thickness, while continuing the scaling of the other critical device dimensions. This negatively impacts the obtainable current density and reduces the performance benefit that typically comes with technology scaling. Yet, even considering these negative implications, this is exactly what most semiconductor companies did when moving to the 65 nm node (as is apparent in Slide 2.25). This should however be considered a temporary therapy, accompanying the mastering of some quite substantial device innovations such as high-\(k\) gate dielectrics and high-mobility transistors.

---

**Slide 2.27**

In this slide, the dependence of the direct-oxide tunneling current is plotted as a function of the applied voltage and the SiO\(_2\) thickness. The leakage current is shown to vary exponentially with respect to both of these parameters. Hence, even though we are scaling the supply voltages with successive process generations, the simultaneous scaling of the oxide thickness causes the gate leakage current density to continuously increase.

---

**Slide 2.28**

The MOS transistor current is proportional to the process transconductance parameter \(k' = \mu C_g = \mu \varepsilon / t_g\). To increase \(k'\) through scaling, one must either find a way to increase the mobility of the carriers or increase the gate capacitance (per unit area). The former requires a fundamental change in the device structure (to be discussed later). With the traditional way of increasing the gate capacitance (i.e., scaling \(t_g\)) running out of steam, the only remaining option is to look for gate dielectrics with a higher permittivity \(\varepsilon\) – the so-called high-\(k\)
dielectrics. Replacing SiO₂ with a “high-k” material yields the same effect as scaling the thickness, while keeping gate leakage under control.

Device technologists have introduced a metric to measure the effectiveness of novel dielectrics: the “equivalent oxide thickness” or EOT, which equals $T_g \times \left( \frac{\varepsilon_{ox}}{\varepsilon_g} \right)$.

Introducing new gate materials is however not a trivial process change, and requires a complete redesign of the gate stack. In fact, most dielectric materials under consideration today require a metal gate electrode, replacing the traditional polysilicon gate. Incorporating major changes of this type into a high-yield manufacturing process takes time and substantial investments. This explains why the introduction of high-k dielectrics into production processes was postponed a number of times. Major semiconductor companies such as IBM and Intel have now adopted hafnium oxide (HfO₂) as the dielectric material of choice for their 45 nm and 32 nm CMOS processes in combination with a metal gate electrode. The relative permittivity of HfO₂ equals 15–30, compared to 3.9 for SiO₂. This is equivalent to between two and three generations of technology scaling, and should help to address gate leakage at least for a while. The resulting drop in gate leakage current for the Intel 45 nm processor is apparent in the chart on Slide 2.25.

---

**Slide 2.29**
The advantages offered by high-k gate dielectrics are quite clear: faster transistors and/or reduced gate leakage.
Slide 2.30
The expected evolution of gate leakage and gate materials is best summarized by this chart, extracted from the International Technology Roadmap on Semiconductors (2005). By analyzing the maximum allowable leakage current density (obviously, this number is disputable – what is allowable depends upon the application domain), it is concluded that the step to high-k dielectrics is necessary by around 2009 (the 45 nm technology node). Combined with some other device innovations such as FD-SOI and dual-gate (more about these later in this Chapter), this may allow for the EOT to scale to around 0.7 nm (!), while keeping the gate leakage current density at approximately 100 A/cm² (or 1 µA/µm²).

Slide 2.31
The influence of temperature on the leakage behavior of a transistor was mentioned a number of times before. In general, it can be assumed that the on-current of a transistor reduces (slightly) with an increase in temperature. The decrease in threshold voltage is not sufficient to offset the decrease in carrier mobility. The threshold reduction on the other hand has an exponential impact on the leakage current. Hence, higher temperatures are detrimental for the $I_{on}$ versus $I_{off}$ ratio as demonstrated for a 90 nm NMOS transistor. Increasing the temperature from 0 to 100°C reduces the ratio by almost 25. This is mostly due to the increase in leakage current (by a factor of 22), but also to slight decrease in on-current (10%).
Variability

- Scaled device dimensions leading to increased impact of variations
  - Device physics
  - Manufacturing
  - Temporal and environmental
- Impacts performance, power (mostly leakage) and manufacturing yield
- More pronounced in low-power design due to reduced supply/threshold voltage ratios

Slide 2.32
The topic of variability rounds out the discussion of the nanometer transistor and its properties. It has always been the case that transistor parameters such as the geometric dimensions or the threshold voltage are not deterministic. When sampled between wafers, within a wafer, or even over a die, each of these parameters exhibits a statistical nature. In the past, the projection of the parameter distributions onto the performance space yielded quite a narrow distribution. This is easily understandable. When the supply voltage is 3 V and the threshold is at 0.5 V, a 25 mV variation in the threshold has only a small impact on the performance and leakage of the digital module. However, when the supply voltage is at 1 V and the threshold at 0.3 V, the same variation has a much larger impact.

So, in past generation processors it was sufficient to evaluate a design over its worst-case corners (FF, SS, FS, SF) in addition to the nominal operation point to determine the yield distributions. Today, this is not sufficient, as the performance distributions have become much wider, and a pure worst-case analysis leads to wasteful design and does not give a good yield perspective either.

Variability Impacts Leakage

Slide 2.33
While variations influence the high-performance design regime, their impact is far more pronounced in the low-power design arena. First of all, the prediction of leakage currents becomes hard. The sub-threshold current is an exponential function of the threshold voltage, and each variation in the latter is amplified in a major way in leakage fluctuations. This is illustrated very well in the performance–leakage distribution plot (for 130 nm technology). When sampled over a large number of dies (and wafers), gate performance varies over 30%, while the leakage current fluctuates by a factor of 5. Observe that the leakiest designs are also the ones with the highest performance (this should be no surprise).
Other reasons why variations play a more pronounced role in low-power design will emerge in the subsequent chapters. However, they can be summarized in the following observation: In general, low-power designs operate at lower supply voltages, lower $V_{DD}/V_{TH}$, and smaller signal-to-noise ratios; these conditions tend to amplify the importance of parameter variations.

**Slide 2.34**

Process variations are not the only cause behind the variability in the performance parameters of a design (such as delay and power dissipation). It actually originates from a broad set of causes with very different temporal characteristics. In a broad sense, we can classify them into physical, manufacturing, environmental, and operational categories. It is probably fair to state that manufacturing variations – that is, fluctuations in device and interconnect parameters caused by the manufacturing process – are dominant in today’s designs. However, with device dimensions approaching the molecular scale, statistical quantum-mechanical effects start to play a role, as the “law of large numbers” starts to be less applicable. Environmental and operational conditions are closely related. While operating a circuit, some “external” parameters such as the supply voltage, the operating temperature, and the coupling capacitance may change dynamically as a result of environmental conditions or the activity profile of the design.

**Slide 2.35**

When trying to create design solutions to address the variability concerns, it pays to understand the nature and the behavior of the sources of variation, as these will ultimately determine what design techniques can be effective in eliminating or reducing the impact. The most important statistical parameters of concern are the temporal and spatial correlations. If a parameter has a strong spatial correlation (i.e., all
devices in the neighborhood show the same trend), a solution such as global tuning proves to be effective. The same is true in the time domain. Very strong temporal correlations (i.e., a device parameter is totally predictable or may not even change over time) can again be addressed by onetime or slow adaptation.

In this slide, we have classified the different sources of variations from a temporal perspective. At the slow extreme of the spectrum are manufacturing variations, which last for the lifetime of the product. Almost similar from a lifetime perspective, but entirely different in nature, are variations caused by wear-out, which manifest themselves only after a very long time of operation (typically years). Examples of such sources are electro-migration, hot-electron degradation, and negative-bias temperature instability (NBTI). Next on the time scale are slow operational or environmental conditions. The temperature gradients on a die vary slowly (in the range of milliseconds), and changes are typically the result of alterations in the operation mode of the system. An example of such is putting a module to sleep or standby mode after a time of intensive computation. Other variations happen at a much faster time scale such as the clock period or even a single signal transition. Their very dynamic nature does not leave room for adaptive cancellation, and circuit techniques such as shielding are the only way to eliminate their impact.

**Slide 2.36**

Process and manufacturing variations are probably of the most concern. The evolutionary trend is clear: virtually all technology parameters such as transistor length, width, oxide thickness, and interconnect resistivity show an increasing variability over time (as measured by the ratio of standard deviation over the mean value). Although each of these parameters is important on its own, the resulting impact on the threshold voltage is what counts most from a digital-design perspective. As shown in the table, the threshold variability is rising from 4% to 16% while evolving from 250 nm to 45 nm CMOS technologies. One may assume that this variation primarily results from the increasing deviations in channel length, since the $V_{TH}$ is quite sensitive to variations in $L$ around the critical dimension (remember the halo implants). The resulting impact on both performance and power metrics is quite substantial.

**Slide 2.37**

Since the lengths of neighboring transistors tend to be similarly affected by deviations in the manufacturing process, one would assume that the threshold voltages of closely spaced transistors should be strongly correlated. This conclusion holds especially for > 100 nm technology nodes, where strong systematic trends in thresholds of local neighborhoods can be observed.
However, the observation becomes less true with continued scaling, when deviations in another device parameter, channel doping, start to become an issue. As shown in the graph, the number of dopant atoms, which is a discrete number, drops below 100 for transistor dimensions smaller than 100 nm. The exact number of dopants in the channel is a random variable, and can change from transistor to transistor. We may hence expect that the correlation in threshold voltages between neighboring transistors will reduce substantially in future technology generations.

The main takeaway from this discussion on process variations is that most device and design parameters will feature broader distributions over time, and that this is primarily caused by variations in the transistor threshold. While these variations tend to be mostly systematic today, we may expect larger random components in the future.

**Device and Technology Innovations**

- Power challenges introduced by nanometer MOS transistors can be partially addressed by new device structures and better materials
  - Higher mobility
  - Reduced leakage
  - Better control
- However …
  - Most of these techniques provide only a one (or two) technology generation boost
  - Need to be accompanied by circuit and system level methodologies

**Slide 2.38**

One may wonder whether these many and profound challenges may make design in the nanometer regime all but impossible. This is a very valid question indeed, which has kept many semiconductor company executives awake at night over the past years.

While reflecting, it pays to keep the following considerations in mind. Over the years, designers have proven to be quite ingenious, and they have come up over and over again with new design technologies and methodologies to address emerging challenges and roadblocks. We can be confident that this will continue to happen in the future (this is what this book is about, after all). At the same time, device engineers are not sitting still either. On the drawing board are a number of device structures that may help to address some, if not all, of the concerns raised in this chapter. For a designer, it is important to be aware of what may be coming down the device pipeline and plan accordingly.
Device and Technology Innovations

- Strained silicon
- Silicon-on-Insulator
- Dual-gated devices
- Very high mobility devices
- MEMS – transistors

Slide 2.39
The devices introduced in the coming slides present any of the following features: higher mobility, better threshold control, or faster sub-threshold current roll-off.

Slide 2.40
The concept of strained silicon was introduced by IBM to increase the mobility in traditional CMOS transistors. From the 65 nm generation onward, it is used almost universally by all semiconductor manufacturers. The generic idea is to create a layer of silicon (typically in the transistor channel), in which the silicon atoms are stretched (or strained) beyond their normal inter-atomic distance.

A generic way to create strain is to put a layer of silicon over a substrate of silicon germanium (SiGe). As the atoms in the silicon layer align with the atoms in the silicon–germanium layer, where the atoms are further apart, the links between the silicon atoms become stretched – thereby leading to strained silicon. Moving the atoms further apart reduces the atomic forces that interfere with the movement of electrons through the transistors, resulting in higher mobility.

The practical realization may differ between manufacturers. The slide illustrates one strategy, as employed by Intel. To stretch the silicon lattice, Intel deposits a film of silicon nitride over the whole transistor at a high temperature. Because silicon nitride contracts less than silicon as it cools, it locks the silicon lattice beneath it in place with a wider spacing than it would normally adopt. This improves electron conduction by 10%. For PMOS transistors, the silicon is compressed. This
is accomplished by carving trenches along opposite ends of the channel. These are filled with silicon germanium, which has a larger lattice size than silicon alone and so compresses the regions nearby. This improves hole conduction by 25%.

**Slide 2.41**

The higher mobility may be used to increase the performance. From a power perspective, a better approach is to use the higher mobility to obtain the same performance with either a higher threshold voltage (reducing leakage), or with a reduced $V_{DD}/V_{TH}$ ratio, as is illustrated in this slide.

**Slide 2.42**

Straining is only one first step toward higher mobility. Materials such as Ge and GaAs are known to have an intrinsic electron mobility that is substantially above what Si can offer. Researchers at various locations are exploring the potential of so-called hetero-devices that combine Si with other materials such as Ge, offering the potential of carriers that are 10 times as mobile, while still relying on traditional Si technology. An example of such a device is the Si-Ge-Si heterostructure developed at Stanford (this is only one example of the many structures being investigated). While these high-mobility devices will need quite some time before making it to the production line (if ever), they offer a clear glimpse at the potential for further improvement.
 Silicon-on-Insulator (SOI) is a technology that has been “on the horizon” for quite a long time, yet it never managed to really break ground, though with some exceptions here and there. An SOI MOS transistor differs from a “bulk” device in that the channel is formed in a thin layer of silicon deposited above an electrical insulator, typically silicon dioxide. Doing so offers some attractive features. First, as drain and source diffusions extend all the way down to the insulator layer, their junction capacitances are substantially reduced, which translates directly into power savings. Another advantage is the higher sub-threshold slope factor (approaching the ideal 60 mV/decade), reducing leakage. Finally, the sensitivity to soft errors is reduced owing to the smaller collection efficiency, leading to a more reliable transistor. There are some important negatives as well. The addition of the SiO₂ layer and the thin silicon layer increases the cost of the substrate material, and may impact the yield as well. In addition, some secondary effects should be noted. The SOI transistor is essentially a three-terminal device without a bulk (or body) contact, and a “body” that is floating. This effectively eliminates body biasing as a threshold-control technique. The floating transistor body also introduces some interesting (ironically speaking. . .) features such as hysteresis and state-dependency.

Device engineers differentiate between two types of SOI transistors: partially-depleted (PD-SOI) and fully-depleted (FD-SOI). In the latter, the silicon layer is so thin that it is completely depleted under nominal transistor operation, which means that the depletion/inversion layer under the gate extends all the way to the insulator. This has the advantage of suppressing some of the floating-body effects, and an ideal sub-threshold slope is theoretically achievable. From a variation perspective, the threshold voltage becomes independent of the doping in the channel, effectively eliminating a source of random variations (as discussed in Slide 2.37). FD-SOI requires the depositing of extremely thin silicon layers (3–5 times thinner than the gate length!).
Slide 2.44
The FD-SOI device architecture can be further extended with an extra feature that reinstates threshold control through a fourth terminal. A buried gate below the SiO₂ insulator layer helps to control the charge in the channel, and thus also the threshold voltage. As shown in these graphs (published by Hitachi), the buried-gate concept pretty much reinstates the idea of body biasing as a viable design option. The reduced impact of random doping variations on the threshold voltage, as is typical in FD-SOI, is also illustrated.

Slide 2.45
The FinFET (called a tri-gate transistor by Intel) is an entirely different transistor structure that actually offers some properties similar to the ones offered by the device presented in the previous slide. The term FinFET was coined by researchers at the University of California at Berkeley to describe a non-planar, double-gated transistor built on an SOI substrate. The distinguishing characteristic of the FinFET is that the controlling gate is wrapped around a thin silicon “fin”, which forms the body of the device. The dimensions of the fin determine the effective channel length of the device. The device structure has shown the potential to scale the channel length to values that are hard, if not impossible, to accomplish in traditional planar devices. In fact, operational transistors with channel lengths down to 7 nm have been demonstrated.

In addition to a suppression of deep submicron effects, a crucial advantage of the device is again increased control, as the gate wraps (almost) completely around the channel.
the threshold voltage. In a sense, this device offers similar functionality as the buried-gate FD-SOI transistor discussed earlier. Controlling the work functions of the two gates through the selection of appropriate type and quantity of the dopants helps to maximize the range and sensitivity of the control knobs.

**Slide 2.46**

This increased two-dimensional control can be exploited in a number of ways. In the dual-gated device, the fact that the gate is controlling the channel from both sides (as well as the top) leads to increased process transconductance. Another option is to remove the top part of the gate, leading to the back-gated transistor. In this structure, one of the gates acts as the standard control gate, whereas the other is used to manipulate the FinFET depends greatly upon how these changes can be translated into a scalable, low-cost and high-yield process – some formidable question, indeed! Also unclear at this time is how the adoption of such a different structure impacts variability, as critical dimensions and device parameters are dependent upon entirely different process steps.

**Slide 2.47**

The fact that the FinFET and its cousins are dramatically different devices compared to your standard bulk MOS transistor is best-illustrated with these pictures from Berkeley and Intel. The process steps that set and control the physical dimensions are entirely different. Although this creates new opportunities, it also brings challenges, as the process steps involved are vastly different. The ultimate success of
Some Futuristic Devices

FETs with sub-threshold swing $< kT/q$ (I-MOS)

Impact Ionization Region

$I^+$ I-MOS $N^+$

Buried Oxide

Zero off-current transistor
Uses MEMS technology to physically change gate control. Allows for zero-leakage sleep transistors and advanced memories
[Ref: Abele05, Kam05]

Slide 2.48
It is worth pointing out that the devices described here represent by no means the complete spectrum of new transistors and switching devices that are currently being explored. In fact, the number of options that are emerging from the research labs these days is quite extraordinary, and the excitement is palpable. Most of these will probably die with a whimper, while other ones are still decades out in terms of true applicability. Of the latter, carbon-nanotube (CNT) transistors seem to present some true potential, but the jury is still out.

When looking from a power angle, some device structures emerging from the research labs merit some special attention. The I-MOS transistor uses substantially different mechanisms, such as impact ionization, to produce a transistor with a sub-threshold slope substantially below 60 mV/decade. This opens the door for a switch with close-to-ideal characteristics. The availability of such a device would allow operation at supply voltages that are substantially lower than what we can allow today.

Another entirely new device would allow for an almost complete elimination of leakage current in standby mode: Using MEMS (Micro-electromechanical systems) technology, the suspended-gate MOSFET (SG-MOS) physically moves the actual gate up and down depending upon the applied gate voltage. In the down-position this device resembles a traditional transistor. Moving the gate into the up-position is physically equivalent to mechanically turning off the switch, effectively squelching all leakage current. The availability of such a device would come extremely handy in the design of low-standby power components.
## Summary

- Plenty of opportunity for scaling in the nanometer age
- Deep-submicron behavior of MOS transistors has substantial impact on design
- Power dissipation mostly influenced by increased leakage (SD and gate) and increasing impact of process variations
- Novel devices and materials will ensure scaling to a few nanometers

A profound awareness of the device characteristics and the ability to adapt to its varying properties will prove to be essential tenets in low-power design in the nanometer era.

## References

### Books and Book Chapters

### Articles
Chapter 3
Power and Energy Basics

Slide 3.1
The goal of this chapter is to derive clear and unambiguous definitions and models for all of the design metrics relevant in the low-power design domain. Anyone with some training and experience in digital design is probably already familiar with a majority of them. If you are one of them, you should consider this chapter as a review. However, we recommend that everyone at least browse through the material, as some new definitions, perspectives, and methodologies are offered. In addition, if one truly wants to tackle the energy problem, it is essential to have an in-depth understanding of the causes of energy dissipation in today’s advanced digital circuits.

Slide 3.2
Before discussing the various sources of power dissipation in modern digital integrated circuits, it is worth spending some time evaluating the metrics typically used to evaluate the quality of a circuit or design. Unambiguous definitions are essential if one wants to provide fair comparisons. The rest of this chapter divides the sources of power roughly along the lines of dynamic and static power. At the end of the chapter, we make the point that optimization for power or energy alone rarely makes sense. Design for low power is most often a trade-off process, performed primarily in the energy-delay space. Realizing this goes a long way in setting up the foundations for an effective power-minimization design methodology.
Slide 3.3
The basic design metrics — propagation delay, energy, and power — are well-known to anyone with a digital design experience. Yet, they may not be sufficient. In today’s design environment, where both delay and energy play on an almost equal base, optimizing for only one parameter rarely makes sense. For instance, the design with the minimum propagation delay in general takes an exorbitant amount of energy, and vice versa, the design with the minimum energy is unacceptably slow. Both represent extremes in a rich optimization space, where many other optimal operational points exist. Hence some other metrics of potential interest have been defined, such as the energy-delay product, which puts an equal weight on both parameters. In fact, the normalized energy-delay products for a number of optimized general-purpose designs fall consistently within a narrow range. While being an interesting metric, the energy-delay product of an actual design only tells us how close the design is to a perfect balance between performance and energy efficiency. In real designs, achieving that balance may not necessarily be of interest. Typically, one metric is assigned greater weight — for instance, energy is minimized for a given maximum delay or delay is minimized for a given maximum energy. For these off-balance situations, other metrics can be defined such as energy−delay”. Though interesting, these derived metrics however are rarely used, as they lead to optimization for only one target in the overall design space.

It is worth at this point to recap the definition of propagation delay: it is measured as the time difference between the 50% transition points of the input and output waveforms. For modules with multiple inputs and outputs, we typically define the propagation delay as the worst-case delay over all possible scenarios.

Slide 3.4
Power dissipation sources can be divided in two major classes: dynamic and static. The difference between the two is that the former is proportional to the activity in the network and the switching frequency, whereas the latter is independent of both. Until recently, dynamic power vastly outweighed static power. With the emergence of leakage as a major power component
though, both should now be treated on an equal footing. Biasing currents for “analog” components such as sense amplifiers or level converters strictly fall under the static power-consumption class, but originate from a design choice rather than a device deficiency.

### Active (or Dynamic) Power

Key property of active power:

\[ P_{\text{dyn}} \propto f \]

where \( f \) is the switching frequency

**Sources:**
- Charging and discharging capacitors
- Temporary glitches (dynamic hazards)
- Short-circuit currents

and should be made as small as possible.

### Slide 3.5

As mentioned, dynamic power is proportional to the switching frequency. The charging and discharging of capacitances is and should be the main source of dynamic power dissipation – as these operations are at the core of what constitutes MOS digital circuit design. The other contributions (short-circuit currents and dynamic hazards or glitches) are parasitic effects

### Charging Capacitors

Applying a voltage step

\[ E_{0 \rightarrow t} = CV^2 \]

\[ E_r = \frac{1}{2} CV^2 \]

\[ E_c = \frac{1}{2} CV^2 \]

\[ V \]

\[ R \]

\[ C \]

\[ \int_{0}^{t} V \frac{dV}{dt} dt = CV \int_{0}^{t} \frac{dV}{dt} = CV^2 \]

Value of \( R \) does not impact energy!

### Slide 3.6

The following equation is probably the most important one you will encounter in this book: to charge a capacitance \( C \) by applying a voltage step \( V \), an amount of energy equal to \( CV^2 \) is taken from the supply. Half of that energy is stored on the capacitor; the other half is dissipated as heat in the resistance of the charging network. During discharge the stored energy in turn is dissipated as heat as well. Observe that the resistance of the networks does not enter the equation.
**Slide 3.7**

This model applies directly to a digital CMOS gate, where the PMOS and NMOS transistors form the resistive charge and discharge networks. For the sake of simplicity, the total capacitance of the network is lumped into the output capacitance of the gate.

- One half of the power from the supply is consumed in the pull-up network and one half is stored on $C_L$
- Charge from $C_L$ is dumped during the $1\rightarrow 0$ transition
- Independent of resistance of charging/discharging network

**Slide 3.8**

More generically, we can compute the energy it takes to charge a capacitance from a voltage $V_1$ to a voltage $V_2$. Using similar math, we derive that this requires from the supply an amount of energy equal to $CV_2(V_2-V_1)$. This equation will come in handy for a number of special circuits. One example is the NMOS pass-transistor chain. It is well-known that the maximum voltage at the end of such a chain is one threshold voltage below the supply [Rabaey03]. Using the afore-derived equation, we find that the energy dissipation in this case equals $CV_{DD}(V_{DD}-V_{TH})$, and is proportional to the swing at the output. In general, reducing the swing in a digital network results in a linear reduction in energy consumption.
Slide 3.9
So far, we have assumed that charging a capacitor always requires an amount of energy equal to $CV^2$. This is true only when the driving waveform is a voltage step. It is actually possible to reduce the required energy by choosing other waveforms. Assume, for instance, that a current source with a fixed current $I$ is used instead. Under those circumstances, the energy consumed in the resistor is reduced to $(RC/T)CV^2$ where $T$ is the charging time, and the output voltage rises linearly with time. Observe that the resistance of the network plays a role under these circumstances. From this, it appears that the dissipation in the resistor can be reduced to very small values, if not zero, by charging the capacitor very slowly (i.e., by reducing $I$).

Slide 3.10
In fact, the current-driven scenario results in an actual energy reduction over the voltage-driven approach for $T > 2RC$. As a reference, the time it takes for the output of the voltage-driven circuit to move between 0% and 90% points equals $2.3RC$. Hence, the current-driven circuit is more energy-efficient than the voltage-driven one as long as it is slower.

For this scheme to work, the same approach should be used to discharge the capacitor, and the charge flowing through the source should be recovered. If not, the energy gained is just wasted in the source.

The idea of “energy-efficient” charging gained a lot of attention in the 1990s. However, the inferior performance and the complexity of the circuitry ensured that the ideas remained confined to the academic world. With the prospect of further voltage scaling bottoming out, these concepts may gain some traction anew (some more about this in Chapter 13).
Charging Capacitors

Driving using a sine wave (e.g., from resonant circuit)

\[ E_C = \frac{1}{2} CV^2 \]

Energy dissipated in resistor can be made arbitrarily small if frequency \( \omega \ll \frac{1}{RC} \)
(output signal in phase with input sinusoid)

evaluate the circuit in the frequency domain. The \( RC \) network is a low-pass filter with a single pole at \( \omega_p = \frac{1}{RC} \). It is well-known that for frequencies much smaller than the pole, the output sinusoid has the same amplitude and the same phase as those of the input waveform. In other words, no or negligible current is flowing through the resistor, and hence little power is dissipated. The attractive feature of the sinusoidal waveforms is that these are easily generated by resonant networks (such as LC oscillators). Again, with some notable exceptions such as power regulators, sinusoidal charging has found little industrial following.

Dynamic Power Consumption

\[
\text{Power} = \text{Energy per transition} \times \text{Transition rate}
\]

\[ = C_L V_{DD}^2 f_{0\rightarrow1} \]

\[ = C_L V_{DD}^2 f p_{0\rightarrow1} \]

\[ = C_{\text{switched}} V_{DD}^2 f \]

- Power dissipation is data dependent – depends on the switching probability, \( p_{0\rightarrow1} \)
- Switched capacitance \( C_{\text{switched}} = p_{0\rightarrow1} C_L = \alpha C_L \)
  (\( \alpha \) is called the switching activity factor)

with a clock frequency \( f \). The probability that a node will make a 0-to-1 transition at a given clock tick is given by \( \alpha f \), where \( 0 \leq \alpha \leq 1 \) is the activity factor at that node. As we discuss in the following slides, \( \alpha \) is a function of the circuit topology and the activity of the input signals. The accuracy of power estimation depends largely upon how well the activity is known – which is most often not very much.

Slide 3.12
This brings us back to the generic case of the CMOS inverter. To translate the derived energy per operation into power, it must be multiplied with the rate of power-consuming transitions \( f_{0\rightarrow1} \). The unit of the resulting metric is Watt (= Joules/sec). This translation leads right away to one of the hardest problems in power analysis and optimization: it requires knowledge of the “activity” of the circuit. Consider a circuit...
The derived expression can be expanded for a complete module by summing over all nodes. The average power is then expressed as \((\alpha C)V^2f\). Here \(\alpha C\) is called the effective capacitance of the module, and equals the average amount of capacitance that is being charged in the module every clock cycle.

### Impact of Logic Function

**Example: Static two-input NOR gate**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Assume signal probabilities

- \(p_{A=1} = 1/2\)
- \(p_{B=1} = 1/2\)

Then transition probability

\[
p_{0 \to 1} = p_{\text{out}=0} \times p_{\text{out}=1} = 3/4 \times 1/4 = 3/16
\]

If inputs switch every cycle

\[\alpha_{\text{NOR}} = 3/16\]

NAND gate yields similar result

### Slide 3.13

Let us, for instance, derive the activity of a two-input NOR gate (which defines the topology of the circuit). Assume that each input has an equal probability of being a 1 or a 0, and that the probability of a transition at a clock tick is 50–50 as well, ensuring an even distribution between states. With the aid of the truth table we derive that the probability of a 0 \(\rightarrow\) 1 transition (or the activity) equals 3/16. More generally, the activity at the output node can be expressed as a function of the 1-probabilities of the inputs \(A\) and \(B\):

\[\alpha_{\text{NOR}} = p_{A}p_{B}(1 - p_{A}p_{B})\]

### Impact of Logic Function

**Example: Static two-input XOR gate**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Assume signal probabilities

- \(p_{A=1} = 1/2\)
- \(p_{B=1} = 1/2\)

Then transition probability

\[
p_{0 \to 1} = p_{\text{out}=0} \times p_{\text{out}=1} = 1/2 \times 1/2 = 1/4
\]

If inputs switch every cycle

\[p_{0 \to 1} = 1/4\]

### Slide 3.14

A similar analysis can be performed for an XOR gate. The observed activity is a bit higher (1/4).
### Transition Probabilities for Basic Gates

As a function of the input probabilities

<table>
<thead>
<tr>
<th></th>
<th>$p_{0 \to 1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>AND</td>
<td>$(1 - p_A p_B) p_A p_B$</td>
</tr>
<tr>
<td>OR</td>
<td>$(1 - p_A)(1 - p_B)(1 - (1 - p_A)(1 - p_B))$</td>
</tr>
<tr>
<td>XOR</td>
<td>$(1 - (p_A + p_B - 2 p_A p_B))(p_A + p_B - 2 p_A p_B)$</td>
</tr>
</tbody>
</table>

Activity for static CMOS gates

$\alpha = p_0 p_1$

---

### Slide 3.15

These results can be generalized for all basic gates.

---

### Slide 3.16

The topology of the logic network has a major impact on the activity. This is nicely illustrated by comparing the activity of NAND (NOR) and XOR gates as a function of fan-in. The output-transition probability of a NAND gate goes asymptotically to zero. The probability of the output being a 0 is indeed becoming smaller with increasing fan-in. An example of such a network is a memory-address decoder. On the other hand, the activity of an XOR network is independent of fan-in. This does not bode well for the power dissipation of modules such as large en(de)cryption and coding functions, which primarily consist of XORs.

### Slide 3.17

One obvious question is how the choice of logic family impacts activity and power dissipation. Some interesting global trends can be observed. Consider, for instance, the case of dynamic logic. The only power-consuming transitions in pre-charged logic occur when the output evaluates to 0, after which it has to be recharged to a high in the next pre-charge cycle. Hence, the activity factor $\alpha$ is equal to the probability of the output being equal to 0. This means that the activity is always higher in dynamic logic (compared to static), independent of the function. This does not mean per se that the power dissipation of dynamic logic is higher, as the effective capacitance is the product of
activity and capacitance, the latter being smaller in dynamic logic. In general though, the higher activity outweighs the capacitance gain.

E.g., \( p_{0 \rightarrow 1} \) (NAND) = \( 1/2^N \), \( p_{0 \rightarrow 1} \) (NOR) = \( (2^N - 1)/2^N \)

Activity in dynamic circuits hence always higher than in static. But ... capacitance most often smaller.

Differential Logic?

\[ \text{Static: Activity is doubled} \]
\[ \text{Dynamic: Transition probability is 1!} \]

Hence power always increases.

Slide 3.18

Another interesting logic family is differential logic, which may seem attractive for very low-voltage designs due to its increased signal-to-noise ratio. Differential implementations come unfortunately with an inherent disadvantage from a power perspective: not only is the overall capacitance higher, the activity is higher as well (for both static and dynamic implementations). The only positive argument is that differential implementation reduces the number of gates needed for a given function, and thus reduces the length of the critical path.

Slide 3.19

As activity is such an important parameter in the analysis of power dissipation, it is worthwhile spending some time on how to evaluate the activity of more complex logic networks. One may wonder whether it is possible to develop a “static power analyzer” along the lines of the “static timing analyzer”. The latter evaluates the propagation delay of a logic network analyzing only the topology of the network without any simulation (hence the name “static”). The odds for successful
straightforward indeed. However, there is a catch. For the basic gate equations to be valid, the inputs must be statistically independent. In probability theory, to say that two events are independent intuitively means that the occurrence of one event makes it neither more nor less probable that the other occurs. While this assumption is in general true for the network of the slide (assuming obviously that all the primary input signals are independent), it unfortunately rarely holds in actual circuits.

Earlier for a NAND gate is no longer applicable, and conditional probabilities need to be used. Conditional probability is the probability of some event \( A \), given the occurrence of some other event \( B \). Conditional probability is expressed as \( p(A|B) \), and is read as "the probability of \( A \), given \( B \)". More specifically, one can derive that \( p(A|B) = \frac{p(A \cap B)}{p(B)} \), assuming that \( p(B) \neq 0 \).

While propagating these conditional probabilities through the network is theoretically possible, you may guess that the complexity of doing so for complex networks rapidly becomes unmanageable – and that indeed is the case.
as well may show temporal dependence. For example, in a digitized speech signal any sample value is dependent upon the previous values.

All these arguments help to illustrate that static activity analysis is a very hard problem indeed, and actually all but impossible. Hence, power analysis tools either rely on simulations of actual signal traces to derive the signal probabilities or make simplifying assumptions – for instance, it is assumed that the input signals are independent and purely random. This is discussed in more detail in Chapter 12. In the following chapters, we will most often assume that activity of a module in its typical operation mode can be characterized by an independent parameter \( x \).

They should be considered as parasitic, and be kept to an absolute minimum.

A dynamic hazard occurs when a single input change causes multiple transitions at the output of a gate. These events, also known as "glitches", are obviously wasteful, as a capacitor is charged and/or discharged without having an impact on the final result. In the analysis of the transition
probabilities of complex logic circuits, presented in the earlier slides, glitches did not appear, as the propagation delays of the individual gates were ignored – all events were assumed to be instantaneous. To detect the occurrence of dynamic hazards a detailed timing analysis is necessary.

**Slide 3.23**
A typical example of the effect of glitching is illustrated in this slide, which shows the simulated response of a chain of NAND gates with all inputs going simultaneously from 0 to 1. Initially, all the outputs are 1, as one of the inputs was 0. For this particular transition, all the odd bits must transition to 0, while the even bits remain at the value of 1. However, owing to the finite propagation delay, the even output bits at the higher bit positions start to discharge, and the voltage drops. When the correct input ripples through the network, the output ultimately goes high. The glitch on the even bits causes extra power dissipation beyond what is required to strictly implement the logic function. Although the glitches in this example are only partial (i.e., not from rail to rail), they contribute significantly to the power dissipation. Long chains of gates often occur in important structures such as adders and multipliers, and the glitching component can easily dominate the overall power consumption.

**Slide 3.24**
The occurrence of glitching in a circuit is mainly due to a mismatch in the path lengths in the network. If all input signals of a gate change simultaneously, no glitching occurs. On the other hand, if input signals change at different times, a dynamic hazard may develop. Such a mismatch in signal timing is typically the result of different path lengths with respect to the primary inputs of the network. This is illustrated in this slide, where two equivalent, but topologically different, realizations of the function $F = A.B.C.D$ are analyzed. Assume that the AND gate
has a unit delay. The leftmost network suffers from glitching as a result of the disparity between the arrival times of the input signals for gates Y and Z. For example, for gate Z, input D settles at time 0, whereas input Y only settles at time 2. Redesigning the network so that all arrival times are identical can dramatically reduce the number of superfluous transitions, as shown in the rightmost network.

**Slide 3.25**
So far, it was assumed that the NMOS and PMOS transistors of a CMOS gate are never ON simultaneously. This assumption is not entirely correct, as the finite slope of the input signal during switching causes a direct current path between $V_{DD}$ and GND for a short period of time. The extra power dissipation due to these “short-circuit” or “crowbar” currents is proportional to the switching activity, similar to the capacitive power dissipation.

**Slide 3.26**
The peak value of the short-circuit current is also a strong function of the ratio between the slopes of the input and output signals. This relationship is best illustrated by the following simple analysis: Consider a static CMOS inverter with a 0→1 transition at the input. Assume first that the load capacitance is very large, so that the output fall time is significantly larger than the input rise time (left side). Under those circumstances, the input moves through the transient region before the output starts to change. As the source–drain voltage of the PMOS device is approximately zero during that period, the device shuts off without ever delivering any current. The short-circuit current is close to zero. Consider now the reverse case (right side), where the output capacitance is very small, and the output fall time is substantially smaller than the input rise time. The drain–source voltage of the PMOS device equals $V_{DD}$ for most of the transition time, guaranteeing a maximal
short-circuit current. This clearly represents the worst-case condition. The conclusions of this intuitive analysis are confirmed by the simulation results.

This analysis may lead to the (faulty) conclusion that the short-circuit dissipation is minimized by making the output rise/fall time substantially larger than the input rise/fall time. On the other hand, making the output rise/fall time too large slows down the circuit, and causes large short-circuit currents in the connecting gates. A more practical rule that optimizes the power consumption in a global way, can be formulated: The power dissipation due to short-circuit currents is minimized by matching the rise/fall times of the input and output signals. At the overall circuit level, this means that rise/fall times of all signals should be kept constant within a range. Equalizing the input and output transition times of a gate is not the optimum solution for the individual gate, but keeps the overall short-circuit current within bounds (maximum 10-15% of the total dynamic power dissipation). Observe also that the impact of short-circuit current is reduced when we lower the supply voltage. In the extreme case, when $V_{DD} < V_{THn} + |V_{THp}|$, the short-circuit dissipation is completely eliminated, because the devices are never ON simultaneously.

---

**Slide 3.27**

As the short-circuit power is proportional to the clock frequency, it can be modeled as an equivalent capacitor:

$$P_{sc} = C_{sc} V_{DD}^2 f$$

which then can be lumped into the output capacitance of the gate. Be aware however that $C_{sc}$ is a function of the input and output transition times.

---

**Slide 3.28**

Although dynamic power traditionally has dominated the power budget, static power has become an increasing concern when scaling below 100 nm. The main reasons behind this have been discussed at length in Chapter 2. Sub-threshold drain-source leakage, junction leakage, and gate leakage all play important roles, but in contemporary design it is the sub-threshold leakage that is the main cause of concern.
Slide 3.29
In Chapter 2, it was pointed out that the main reason behind the increase in drain–source leakage is the gradual reduction of the threshold voltage forced by the lowering of the supply voltages. Any reduction in threshold voltage causes the leakage current to grow exponentially. The chart illustrating this is repeated for the purpose of clarity.

Slide 3.30
An additional factor is the increasing impact of the DIBL effect. Combining the equations for subthreshold leakage and the influence of DIBL on $V_{TH}$, an expression for the leakage power of a gate can be derived. Observe the exponential dependence of leakage power upon both $V_{TH}$ and $V_{DD}$.

Slide 3.31
The dependence of the leakage current on the applied drain–source voltage creates some interesting side effects in complex gates. Consider, for example, the case of a two-input NAND gate where the two NMOS transistors in the pull-down network are turned off. If the off-resistance of NMOS transistors would be fixed, and not a function of the applied voltage, one would expect that the doubling of the resistance by putting two transistors in series would halve the leakage current (compared to a similar-sized inverter).

An actual analysis shows that the reduction in leakage is substantially larger. When the pull-down chain is off, node M settles to an intermediate voltage, set by balancing the leakage currents of transistors M1 and M2. This reduces the drain–source voltage of both transistors (especially of transistor M2), which translates into a substantial reduction in the leakage currents
expression for the leakage current as a function of the DIBL factor $\lambda_d$ and the sub-threshold swing $S$. The resulting equation shows that the reduction in leakage current obtained by stacking transistors is indeed larger than the linear factor one would initially expect. This is called the stacking effect.

60 mV, resulting in a ninefold reduction in leakage current. The negative $V_{GS}$ of 60 mV for transistor M1 translates into a similar reduction.

The impact of the stacking effect is further detailed in the table, which illustrates the reduction in leakage currents for various stack sizes in 90 nm technology. The leakage reductions for both NMOS and PMOS stacks are quite impressive. The impact is somewhat smaller for the PMOS chains, as the DIBL effect is smaller for those devices. The stacking effect will prove to be a powerful tool in the fight against static power dissipation.
to decrease the voltage stress over the gate dielectric – which means reducing voltage levels.

Similar to sub-threshold leakage, gate leakage is also an exponential function of the supply voltage. This is illustrated by the simulation results of a 90 nm CMOS inverter. The maximum leakage current is around 100 pA, which is an order of magnitude lower than the sub-threshold current. Yet, even for these small values, the impact can be large, especially if one wants to store a charge on a capacitor for a substantial amount of time (such as in DRAMs, charge pumps, and even dynamic logic). Remember also that the gate leakage is an exponential function of the dielectric thickness.

Slide 3.34
Finally, junction leakage, though substantially smaller than the previously mentioned leakage contributions, should not be ignored. With the decreasing thickness of the depletion regions owing to the high doping levels, some tunneling effects may become pronounced in sub-50 nm technology nodes. The strong dependence upon temperature must again be emphasized.
Other Sources of Static Power Dissipation

- Circuit with dc bias currents:
  
  sense amplifiers, voltage converters and regulators, sensors, mixed-signal components, etc.

Should be turned off when not in use, or standby current should be minimized

Slide 3.35
A majority of the state-of-the-art digital circuits contain a number of analog components. Examples of such circuits are sense amplifiers, reference voltages, voltage regulators, level converters, and temperature and leakage sensors. One property of each of these circuits is that they need a bias current for correct operation. These currents can become a sizable part of the total static power budget. To reduce their contribution, two mechanisms can be used:

1. Trade off performance for current – Reducing the bias current of an analog circuit, in general, impacts its performance. For instance, the gain and slew rate of an amplifier benefit, from a higher bias current.

2. Power management – some analog components need to operate only for a fraction of the time. For example, a sense amplifier in a DRAM or SRAM memory only needs to be ON at the end of the read cycle. Under those conditions the static power can be substantially reduced by turning off the bias when not in use. While being most effective, this technique does not always work as some bias or reference networks need to be ON all the time, or their start-up time would be too long to be practical.

In short, every analog circuit should be examined very carefully, and bias current and ON time should be minimized. The “a bias should never be on when not used” principle rules.

Summary of Power Dissipation Sources

\[ P = \alpha \cdot (C_L + C_{IC}) \cdot V_{sw} \cdot V_{DD} \cdot f + (I_{DC} + I_{LEAK}) \cdot V_{DD} \]

- \( \alpha \) – switching activity
- \( C_L \) – load capacitance
- \( C_{IC} \) – short-circuit capacitance
- \( V_{sw} \) – voltage swing
- \( f \) – frequency
- \( I_{DC} \) – static current
- \( I_{LEAK} \) – leakage current

Slide 3.36
From all the preceding discussions, a global expression for the power dissipation of a digital circuit can be derived. The two major components, the dynamic and static dissipation, are easily recognized. An interesting perspective on the relationship between the two is obtained by realizing that a given computation (such as a multiplication or the execution of an instruction on a processor) is best characterized by its energy cost. Static dissipation, on the other hand, is best captured as a power quantity. To determine the
relative balance between the two, the former must be translated into power by multiplying it with its execution rate, or, in other words, the activity. Hence, precise knowledge of the activity is essential if one wants to estimate the overall power dissipation. Note: in a similar way, multiplying the static power with the time period leads to a global expression for energy.

Slide 3.37
The growing importance of power minimization and containment is revolutionizing design as we know it. Methodologies that were long-accepted have to be adjusted, and established design flows modified. Although this trend was visible already a decade ago in the embedded design world, it was only recently that it started to
eupend a number of traditional beliefs in the high-performance design community. Ever-higher clock frequencies were the holy grail of the microprocessor designer. Though architectural optimizations played a role in the performance improvements demonstrated over the years, reducing the clock period through technology scaling was responsible for the largest fraction.

Once the architecture was selected, the major function of the design flow was to optimize the circuitry through sizing, technology mapping, and logical transformations so that the maximum performance was obtained. Supply and threshold voltages were selected in advance to guarantee top performance.

Slide 3.38
This philosophy is best-reflected in the popular “logical effort”-based design optimization methodology. The delay of a circuit is minimized if the “effective fan-out” of each stage is made equal (and set to a value of approximately 4). Though this technique is very powerful, it also guarantees that power consumption is maximal! In the coming chapters, we will reformulate the logical-effort methodology to bring power into the equation.
Model Not Appropriate Any Longer

Traditional scaling model

If \( V_{dd} = 0.7 \) and \( \text{Freq} = \frac{1}{0.7} \),
\[
\text{Power} = CV_{dd}f = \left(\frac{1}{0.7} \times 1.14^2\right) \times (0.7^2) \times \left(1\right) = 1.3
\]

Maintaining the frequency scaling model

If \( V_{dd} = 0.7 \) and \( \text{Freq} = 2 \),
\[
\text{Power} = CV_{dd}f = \left(\frac{1}{0.7} \times 1.14^2\right) \times (0.7^2) \times (2) = 1.8
\]

While slowing down voltage scaling

If \( V_{dd} = 0.85 \) and \( \text{Freq} = 2 \),
\[
\text{Power} = CV_{dd}f = \left(\frac{1}{0.7} \times 1.14^2\right) \times (0.85^2) \times (2) = 2.7
\]

Slide 3.39

That the circuit optimization philosophy of old can no longer be maintained is best illustrated by this simple example (after Sreekumar Borkar from Intel). Assume a microprocessor design implemented in a given technology. Applying a single technology scaling step reduces the critical dimensions of the chip by a factor of 0.7. General scaling, which reduces the voltage by the same factor, increases the clock frequency by a factor of 1.41. If we take into account the fact that the die size typically increases (actually, used to increase is a better wording) by a factor of 14% between generations, the total capacitance of the die increases by a factor of \((1/0.7) \times 1.14^2 = 1.86\). (This simplified analysis assumes that all the extra transistors are used to good effect.) The net effect is that the power dissipation of the chip increases by a factor of 1.3.

However, microprocessor designers tend to push harder than that. Over the past decades, processor frequency increased by a factor of 2 between technology generations. The extra performance improvement was obtained by circuit optimizations, such as a reduction in the logical depth. Maintaining this rate of improvement now pushes the power dissipation up by a factor of 1.8.

The situation gets even worse when the slowdown in supply voltage scaling is taken into account. Reducing the supply voltage even by a factor of 0.85 means that the power dissipation now rises by 270% from generation to generation. As this is clearly unacceptable, a change in design philosophy was the only option.

The New Design Philosophy

- Maximum performance (in terms of propagation delay) is too power-hungry, and/or not even practically achievable
- Many (if not most) applications either can tolerate larger latency or can live with lower-than-maximum clock speeds
- Excess performance (as offered by technology) to be used for energy/power reduction

Trading off speed for power

Slide 3.40

This revised philosophy backs off from the “maximum performance at all cost” theory, and abandons the notion that clock frequency is equivalent to performance. The “design slack” that results from a less-than-maximum clock speed in a new technology can now be used to keep dynamic and static power within bounds. Performance increase is still possible, but now comes mostly from architectural optimizations – sometimes, but not always, at the expense of extra die area. Design now becomes a trade-off exercise between speed and energy (or power).
Slide 3.41
This trade-off is wonderfully illustrated by this set of, by now legendary, charts. Originated by T. Kuroda and T. Sakurai in the mid 1990s, the graphs plot power and (propagation) delay of a CMOS module as a function of the supply and threshold voltages — two parameters that were considered to be fixed in earlier years. The opposing nature of optimization for performance and power becomes obvious — the highest performance happens to occur exactly where power dissipation peaks (high $V_{DD}$, low $V_{TH}$). Another observation is that the same performance can be obtained at a number of operational points with vastly different levels of power dissipation. The existence of these “equal-delay” and “equal-power” curves proves to be an important optimization instrument when trading off in the delay–power (or energy) space.

Slide 3.42
Contours of identical performance or energy are more evident in the two-dimensional plots of the delay and the (average) energy per operation as functions of supply and threshold voltages. The latter is obtained by multiplying the average power (as obtained using the expressions of Slide 3.36) by the length of the clock period. Similar trends as shown in the previous slide can be observed. Particularly interesting is that a point of minimum energy can be located. Lowering the voltages beyond this point makes little sense as the leakage energy dominates, and the performance deteriorates rapidly.

Be aware that this set of curves is obtained for one particular value of the activity. For other values of $\alpha$, the balance between static and dynamic power shifts, and so do the trade-off curves. Also, the curves shown here are for fixed transistor sizes.
around two times the device threshold. In fact, a better estimate is \(3V_{\text{TH}}(3-\alpha)\) (with \(\alpha\) the fit parameter in the alpha delay model – not to be confused with the activity factor). For \(\alpha = 1.4\), this translates to 1.875 \(V_{\text{TH}}\). Although this is an interesting piece of information, its meaning should not be over-estimated. As mentioned earlier, the EDP metric is only useful when equal weight is placed on delay and energy, which is rarely the case.

Slide 3.43

A further simplification of the graphs is obtained by keeping the threshold voltage constant. The opposing trends between energy and delay when reducing the supply voltage are obvious. One would expect that the product of the two (the energy–delay product or EDP) to show a minimum, which it does. In fact, it turns out that for CMOS designs, the minimum value of the EDP occurs approximately

Slide 3.44

The above charts amply demonstrate that design for low power is a trade-off process. We have found that the best way to capture the duality between performance and energy efficiency is the energy–delay curves. Given a particular design and a set of design parameters, it is possible to derive a pareto-optimal curve that for every delay value gives the minimum attainable energy and vice versa. This curve is the best characterization of the energy and performance efficiency of a design. It also helps to redefine the design problem from “generate the fastest possible design” into a two-dimensional challenge: given a maximum delay, minimize the energy, or, given the maximum energy, find the design with the minimum delay.

We will use energy–delay curves extensively in the coming chapters. In the next chapter, we provide effective techniques to derive the energy–delay curves for a contemporary CMOS design.
Slide 3.45
In summary, we have analyzed in detail the various sources of power dissipation in today’s CMOS digital design, and we have derived analytical and empirical models for all of them. Armed with this knowledge, we are ready to start exploring the many ways of reducing power dissipation and making circuits energy-efficient. One of the main lessons at the end of this story is that there is no free lunch.

Slide 3.46
Some references . . .

Optimization for energy most often comes at the expense of extra delay (unless the initial design is sub-optimal in both, obviously). Energy–delay charts are the best way to capture this duality.

---

**Summary**

- Power and energy are now primary design constraints
- Active power still dominating for most applications
  - Supply voltage, activity and capacitance the key parameters
- Leakage becomes major factor in sub-100 nm technology nodes
  - Mostly impacted by supply and threshold voltages
- Design has become energy–delay trade-off exercise!
optimizing power @ design time

Jan M. Rabaey
Dejan Marković
Borivoje Nikolić

dynamic or static power. This chapter focuses on techniques for power reduction at design time and at circuit level. Practical questions often expressed by designers are addressed: whether gate sizing or choice of supply voltage yields larger returns in terms of power–delay; how many supplies are needed; what is the preferred ratio of discrete supplies to thresholds; etc. As was made clear at the end of the previous chapter, all optimizations should be seen in the broader light of an energy–delay trade-off. To help guide this process, we introduce a unified sensitivity-based optimization framework. The availability of such a framework makes it possible to compare in an unbiased way the impact of various parameters such as gate size and supply and threshold voltages on a given design topology. The results serve as the foundation for optimization at the higher levels of abstraction, which is the focus of later chapters.
Slide 4.2
The chapter starts with the introduction of a unified energy-delay optimization framework, constructed as an extension of the powerful logical-effort approach, which was originally constructed to target performance optimization. The developed techniques are then used to evaluate the effectiveness and applicability of design-time power reduction techniques at the circuit level. Strategies to address both dynamic and static power are considered.

Slide 4.3
Before embarking on any optimization, we should recall that the power and energy metrics are related, but that they are by no means identical. The link between the two is the activity, which changes the ratio between the dynamic and static power components, and which may vary dynamically between operational states. Take, for instance, the example of an adder.

When the circuit is operated at its maximum speed and inputs are changing constantly and randomly, the dynamic power component dominates. On the other hand, when the activity is low, static power rules. In addition, the desired performance of the adder may vary over time as well, further complicating the optimization trajectory.

It will become apparent in this chapter that different design techniques apply to the minimization of dynamic and static power. Hence it is worth classifying power reduction techniques based on the activity level, which is a dynamically varying parameter as discussed before. Fortunately, there exists a broad spectrum of optimizations that can be readily applied at design time, either because they are independent of the activity level or because the module activity is fixed and known in advance. These “design-time” design techniques are the topic of the next four chapters. In general though, activity and performance requirements vary over time, and the minimization of power/energy under these circumstances requires techniques that adapt to the prevailing conditions. These are called “run-time” optimizations. Finally, one operational condition requires special attention: the case where the system is idle (or is in “standby”). Under such circumstances, the dynamic power component approaches zero, and
leakage power dominates. Keeping the static power within bounds under such conditions requires dedicated design techniques.

Slide 4.4
At the end of the previous chapter, it was argued that design optimization for power and/or energy requires trade-offs, and that energy and delay represent the major axes of the trade-off space. (Other metrics such as area or reliability play a role as well, but are only considered as secondary factors in this book.) This naturally motivates the use of energy–delay (E–D) space as the coordinate system in which designers evaluate the effectiveness of their techniques.

By changing the various independent design parameters, each design maps onto a constrained region of the energy–delay plane. Starting from a non-optimized design, we want to either speed up the system while keeping the design under the power cap (indicated by \( E_{\text{max}} \)), or minimize energy while satisfying the throughput constraint (\( D_{\text{max}} \)). The optimization space is bounded by the optimal energy–delay curve. This curve is optimal (for the given set of design parameters), because all other achievable points either consume more energy for the same delay or have a longer delay for the same energy. Although finding the optimal curve seems quite simple in this slide, in real life it is far more complex. Observe also that any optimal energy–delay curve assumes a given activity level, and that changes in activity may cause the curve to shift.

Slide 4.5
The problem is that there are many sets of parameters to adjust. Some of these variables are continuous, like transistor sizes, and supply and threshold voltages. Others are discrete, like different logic styles, topologies, and micro-architectures. In theory, it should be possible to consider all parameters at the same time, and to define a single optimization problem. In practice, we have learned that the complexity of the problem becomes overwhelming, and that the resulting designs (if the process ever converges) are very often sub-optimal.

Hence, design methodologies for integrated circuits rely on some important concepts to help manage complexity: abstraction (hiding the details) and hierarchy (building larger entities through a composition of smaller ones). The two most often go hand-in-hand. The abstraction stack of a typical digital IC design flow is shown in this slide. Most design parameters are, in general, confined to and selected in a single layer of the stack only. For instance, the choice between different instruction sets is a typical micro-architecture optimization, while the choice between devices with different threshold voltages is best performed at the circuit layer.
Layering, hence, is the preferred technique to manage complexity in the design optimization process.

**Slide 4.6**
The layered approach may give the false impression that optimizations within different layers are independent of each other. This is definitely not the case. For instance, the choice of the threshold voltages at the circuit layer changes the shape of the optimization space at the logical or architectural layers. Similarly, introducing architectural transformations such as pipelining may increase the size of the optimization space at the circuit level, thus leading to larger potential gains. Hence, optimizations may and must span the layers.

Design optimization in general follows a “meet-in-the-middle” formulation: specifications and requirements are propagated from the highest abstraction layer downward (top-down), and constraints are propagated upward from the lowest abstraction layer (bottom-up).

**Slide 4.7**
Continuous design parameters such as supply voltages and transistor sizes give rise to a continuous optimization space and a single optimal energy–delay curve. Discrete parameters, such as the choice between different adder topologies, result in a set of optimal boundary curves. The overall optimum is then defined by their composite.

For example, topology B is better in the energy-performance sense for large target delays, whereas topology A is more effective for shorter delays.
One of the goals of this chapter is to demonstrate how we can quickly search for this global optimum, and based on that, build an understanding of the scope and effectiveness of the different design parameters.

Slide 4.8
Given an appropriate formulation of the energy and delay as a function of the design parameters, any optimization program can be used to derive the optimal energy–delay curve. Most of the optimizations and design explorations in this text were performed using various modules of the MATLAB program [Mathworks]. Yet, though relying on automated optimization is very useful to address large problems or to get precise results quickly, some analytical techniques often come in handy to judge the effectiveness of a given parameter, or to come to a closed-form solution.

The energy–delay sensitivity is a tool that does just that: it presents an effective way to evaluate the effectiveness of changes in various design variables. It relies on simple gradient expressions that quantify the profitability of a design modification: how much change in energy and delay results from tuning one of the design variables. Consider, for instance, the operation point \((A_0, B_0)\), where \(A\) and \(B\) are the design variables being studied. The sensitivity to each of the variables is simply the slope of the curve obtained by a small change in that variable. Observe that the sensitivities are negative owing to the nature of energy–delay trade-off (when we compare sensitivities in the rest of
the text, we will use their absolute values – a larger absolute value indicates a higher potential for energy reduction. For example, variable $B$ has higher energy–delay sensitivity at point $(A_0, B_0)$ than the variable $A$. Changing $B$ hence yields a larger potential gain.

### Slide 4.9
The optimal energy–delay curve as defined earlier is a pareto-optimal curve (a notion borrowed from economics). An assignment or operational point in a multi-dimensional search is pareto-optimal if improving on one metric by necessity means hurting another.

An interesting property of a pareto-optimal point is that the sensitivities to all design variables must be equal. This can be understood intuitively. If the sensitivities are not equal, the difference can be exploited to generate a no-loss improvement. Consider, for instance, the example presented here, where we strive to minimize the energy for a given delay $D_0$. Using the “lower-energy-cost” variable $A$, we first create some timing slack $\Delta D$ at a small expense in energy $\Delta E$ (proportional to $A$’s E–D sensitivity). From the new operation point $(A_1, B_0)$, we can now use “higher-energy-cost” variable $B$ to achieve an overall energy reduction as indicated by the formula. The fixed point in the optimization is clearly reached when all sensitivities are equal.

### Slide 4.10
In the rest of the chapter, we primarily focus on the circuit and logic layers. Let us first focus on the active component of power dissipation, or, in light of the E–D trade-off perspective, active energy dissipation. The latter is a product of switching activity at the output of a gate, load capacitance at the output, logic swing, and supply voltage. The simple guideline for energy reduction is therefore to reduce each of the terms in the product expression. Some variables, however, are more efficient than others.

The largest impact on active energy is effected seemingly through supply voltage scaling, because of its quadratic impact on power (we assume that the logic swing scales accordingly). All other terms have
linear impact. For example, smaller transistors have less capacitance. Switching activity mostly depends on the choice of circuit topology.

For a fixed circuit topology, the most interesting trade-off exists between supply voltage and gate sizing, as these tuning knobs affect both energy and performance. Threshold voltages play a secondary role in this discussion as they impact performance without influencing dynamic energy.

**Slide 4.11**
Throughout this discussion, it is useful to keep in mind that the optimizations in the E–D space also impact other important design metrics that are not captured here, such as area or reliability. Take, for example, the relationship between transistor sizing and circuit reliability. Trimming the gates on the non-critical paths saves power without a performance penalty – and hence seems to be a win-win operation. Yet in the extreme case, this results in all paths becoming critical (unless a minimum gate size constraint is reached, of course). This effect is illustrated in the slide. The downsizing of non-critical gates narrows the delay distribution and moves the average closer to the maximum delay. This makes this design vulnerable to process-variation effects and degrades its reliability.

**Slide 4.12**
To evaluate fully the impact of the design variables in question, that of supply and threshold voltages and gate size on energy and performance, we need to construct a simple and effective, yet accurate, optimization framework. The search for a globally optimal energy–delay curve for a given circuit topology and activity level is formulated as an optimization problem:

Minimize energy subject to a delay constraint and bounds on the range of the optimization variables \((V_{DD}, V_{TH},\) and \(W)\).
Optimization is performed with respect to a reference design, sized for minimum delay at the nominal supply and threshold voltages as specified for the technology (e.g., $V_{DD} = 1.2$ V and $V_{TH} = 0.35$ V for a 90 nm process). This reference point is convenient, as it is well-defined.

**Slide 4.13**
The core of the framework consists of effective models of delay and energy as a function of the design parameters. To develop the expressions, we assume a generic circuit configuration as illustrated in the slide. The gate under study is at the $i$-th stage of a logical network, and is loaded by a number of gates in stage $i + 1$, which we have lumped into a single equivalent gate. $C_{w}$ represents the capacitance of the wire, which we will assume to be proportional to the fan-out (this is a reasonable assumption for a first-order model).

$$t_p = \frac{K_d V_{DD}}{(V_{DD} - V_{on})^{\alpha_d}} \left( \frac{C_i + C_{w} + C_{i+1}}{C_i} \right) = \tau_{base} \left( 1 + \frac{1}{\gamma} \frac{C_{i+1}}{C_i} \right)$$

Fit parameters: $V_{on}, \alpha_d, K_d, \gamma$

---

**Slide 4.14**
The delay modeling of the complex gate $i$ proceeds in two steps. First, we derive the delay of an inverter as a function of supply voltage, threshold, and fan-out; Next, we expand this to more complex gates.

The delay of an inverter is expressed using a simple linear delay model, based on the alpha-power law for the drain current (see Chapter 2). Note that this model is based on curve-fitting. The parameters $V_{on}$ and $\alpha_d$ are intrinsically related, yet not equal, to the transistor threshold and the velocity saturation index. $K_d$ is another fit parameter and relates to the transconductance of the process (amongst others). The model fits SPICE simulated data quite nicely, across a range of supply voltages, normalized to the nominal supply voltage (which is 1.2 V for our 90 nm CMOS technology). Observe that this model is only valid if the supply voltage exceeds the threshold voltage by a reasonable amount. (This constraint will be removed in Chapter 11, where we present a modified model that extends into the sub-threshold region.)
The fan-out $f = C_{i-1}/C_i$ represents the ratio of the load capacitance divided by the gate capacitance. A small modification allows for the inclusion of the wire capacitance ($f'$). $\gamma$ is another technology-dependent parameter, representing the ratio between the output and input capacitance of a minimum-sized unloaded inverter.

\begin{align*}
    t_p &= \tau_{\text{nom}} \left( p_i + \frac{f_i g_i}{\gamma} \right) \\
    \text{- Parasitic delay } p_i \text{ - depends upon gate topology} \\
    \text{- Electrical effort } f_i = S_{i+1}/S_i \\
    \text{- Logical effort } g_i \text{ - depends upon gate topology} \\
    \text{- Effective fan-out } h_i = f_i g_i
\end{align*}

[Ref: I. Sutherland, Morgan-Kaufman'99]

\textbf{Slide 4.15}

The other part of the model is based on the logical-effort formulation, which extends the notion to complex gates. Using the logical-effort notation, the delay can be expressed simply as a product of the process-dependent time constant $\tau_{\text{nom}}$ and a unitless delay, $p_i + f_i g_i/\gamma$, in which $g$ is the logical effort that quantifies the relative ability of a gate to deliver current, $f$ is the ratio of the total output to input capacitance of the gate, and $p$ represents the delay component due to the self-loading of the gate. The product of the logical effort and the electrical effort is called the effective fan-out $h$. Gate sizing enters the equation through the fan-out factor $f = S_{i+1}/S_i$.

\textbf{Slide 4.16}

For the time being, we only consider the switching energy of the gate. In this model, $f'_{i}C_{i}$ is the total load at the output, including wire and gate loads, and $\gamma C_{i}$ is the self-loading of the gate. The total energy stored on these capacitances is the energy taken out of the supply voltage in stage $i$.

Now, if we change the size of the gate in stage $i$, it affects only the energy stored on the input capacitance and parasitic capacitance of that gate.

\[ E_i = K_i S_i (V_{DD,i+1}^2 + \gamma V_{DD,i}^2) \]

$E_i$ hence is defined as the energy that the gate at stage $i$ contributes to the overall energy dissipation.
Optimizing Return on Investment (ROI)

**Depends on Sensitivity \( \frac{\partial E}{\partial D} \)**

- **Gate Sizing**
  \[
  \frac{\partial E}{\partial S_i} = \frac{E_i}{\tau_{na} (h_i - h_d)}
  \]

  \( \approx \) for equal \( h \)

  \( (D_{\text{min}}) \)

- **Supply Voltage**
  \[
  \frac{\partial E}{\partial V_{bb}} = \frac{E}{D} \left( \frac{2 \cdot (1 - \frac{V_{cm}}{V_{bb}})}{\alpha_d - 1 + \frac{V_{cm}}{V_{bb}}} \right)
  \]

  max at \( V_{bb}(\text{max}) \)

  \( (D_{\text{min}}) \)

Slide 4.17
As mentioned, sensitivity analysis provides intuition about the profitability of optimization. Using the models developed in the previous slides, we can now derive expressions for the sensitivities to some of the key design parameters.

The formulas indicate that the largest potential for energy savings is at the minimum delay, \( D_{\text{min}} \), which is obtained by equalizing the effective fan-out of all stages, and setting the supply voltage at the maximum allowable value. This observation intuitively makes sense: at minimum delay, the delay cannot be reduced beyond the minimum achievable value, regardless of how much energy is spent. At the same time, the potential of energy savings through voltage scaling decreases with reducing supply voltages: \( E \) decreases, while \( D \) and the ratio \( V_{\text{in}}/V_{\text{DD}} \) increase.

The key point to realize is that optimization primarily exploits the tuning variable with the largest sensitivity, which ultimately leads to the solution where all sensitivities are equal. You will see this concept at work in a number of examples.

Example: Inverter Chain

- **Properties of inverter chain**
  - Single path topology
  - Energy increases geometrically from input to output

Slide 4.18
We use a number of well-known circuit topologies to illustrate the concepts of circuit optimization for energy. The examples differ in the amount of off-path loading and path reconvergence. By analyzing how these properties affect the energy profile, we may come to some general principles related to the impact of the various design parameters. More precisely, we study the (well-understood)

inverter chain and the tree adder - as these examples differ widely in the number of paths and path reconvergence.

Let us begin with the inverter chain. The goal is to find the optimal sizing, the supply voltages, and the number of stages that result in the best energy–delay trade-off.
In a first step, we consider solely transistor sizing. For a given delay increment, the optimum size of each stage, which minimizes energy, can be derived. The sensitivities derived in Slide 4.17 already give a first idea on what may unfold: the sensitivity to gate sizing is proportional to the energy stored on the gate, and is inversely proportional to the difference in effective fan-outs. What this means is that, for equal sensitivity in all stages, the difference in the effective fan-outs of a gate must increase in proportion to the energy stored on the gate, indicating that the difference in the effective fan-outs should increase exponentially toward the output.

This result was already analytically derived by Ma and Franzon [Ma, JSSC’94], who showed that a tapered staging is the best way to combine performance and energy efficiency. One caveat: At large delay increments, a more efficient solution can be found by reducing the number of stages — this was not included as a design parameter in this first-order optimization, in which the topology was kept unchanged.

This source of energy consumption first, by reducing the supply voltage of the gate that drives the load. As (dis)charging $C_L$ is the largest source of energy consumption, the impact of this is quite profound.
energy dissipation by more than 70%. Again, it is shown that for any value of the delay increment, the parameter with the largest sensitivity has the largest potential for energy reduction. For example, at small delay increments sizing has the largest sensitivity (initially infinity), so it offers the largest energy reduction. Its potential however quickly falls off. At large delay increments, it pays to scale the supply voltage of the entire circuit, achieving the sensitivity equal to that of sizing at around 25% excess delay. The largest reductions can be obtained by custom voltage scaling. Yet, two discrete voltages are almost as good, and are a lot simpler from an implementation perspective.

**Slide 4.21**
Now, how good can all this be in terms of energy reduction? In the graphs, we present the results of various optimizations performed on the inverter chain: sizing, reducing the global $V_{DD}$, two discrete $V_{DD}$, and a customizable $V_{DD}$ per stage. For each of these cases, the sensitivity and the energy reduction are plotted as functions of the delay increment (over $D_{min}$). The prime observation is that increasing the delay by 50% reduces the

**Example: Kogge–Stone Tree Adder**

- Tree adder
  - Long wires
  - Reconvergent paths
  - Multiple active outputs

[Ref: P. Kogge, Trans. Comp’73]
(circles). The final-sum outputs are generated through XOR functions (diamonds). To balance the delay paths, buffers (triangles) are inserted in many of the paths.

**Slide 4.23**
The adder topology is best understood in a two-dimensional plane. One axis is formed by the different bit slices \( N \) (we are considering a 64-bit adder for this example), whereas the other is formed by the consecutive adder stages. As a tree adder, the number of stages equals \( \log_2(N) + M \), where \( M \) is the extra stages for propagate/generate and the final XOR functionality. The energy of an internal node is best understood when plotted with respect to this two-dimensional topology.

As always, we start from a reference design that is optimized for minimum delay, and we explore how we can trade off energy and delay starting from that point. The initial sizing makes all paths in the adder equal to the critical path. The first figure shows the energy map for the minimum delay. Though the output nodes are responsible for a sizable fraction of the energy consumption, a number of internal nodes (around stage 5) dominate.

The large internal energy increases the potential for energy reduction through gate sizing. This is illustrated by the case where we allow for a 10% delay increase. We have plotted the energy distribution resulting from sizing, as well as from the introduction of two discrete supply voltages. The former results in 54% reduction in overall energy, whereas the latter only (!) saves 27%.

This result can be explained as follows. Given the fact that the dominant energy nodes are internal, sizing allows each of these nodes to be attacked individually without too much of a global impact. In the case of dual supplies, one must be aware that driving a high-voltage node from a low-voltage node is hard. Hence the preferable assignment of low-voltage nodes is to start from the output nodes and to work one’s way toward the input nodes. Under these conditions, we have already sacrificed a lot of delay slack on low-energy intermediate nodes before we reach the internal high-energy nodes. In summary, supply voltages cannot be randomly assigned to nodes. This makes the usage of discrete supply voltages less effective in modules with high internal energy.

**Slide 4.24**
We can now put it all together, and explore the tree adder in the energy–delay space. Each of the design parameters \( (V_{DD}, V_{TH}, S) \) is analyzed separately and in combination with the others. (Observe that inclusion of the threshold voltage as a design parameter only makes sense when the leakage energy is considered as well – how this is done is discussed later in the chapter).

A couple of interesting conclusions can be drawn:

- Through circuit optimization, we can reduce the energy consumption of the adder by a factor of 10 by doubling the delay.
Exploiting only two out of the three variables yields close to the optimal gain. For the adder, the most effective parameters are sizing and threshold selection. At the reference design point, sizing and threshold reduction feature the largest and the smallest sensitivities, respectively. Hence, this combination has the largest potential for energy reduction along the lines demonstrated in Slide 4.8.

Finally, circuit optimization is most effective in a small region around the reference point. Expanding beyond that region typically becomes too expensive in terms of energy or delay cost for small gains, yielding a reduced return on investment.

---

**Slide 4.25**

So far, we have studied the theoretical impact of circuit optimization on energy and delay. In reality, the design space is more constrained. Choosing a different supply or threshold voltage for every gate is not a practical option. Transistor sizes come in discrete values, as determined by the available design library. One of the fortunate conclusions emerging from the preceding studies is that a couple of well-chosen discrete values for each of the design parameters can get us quite close to the optimum.

Let us first consider the practical issues related to the use of multiple supply voltages – a practice that until recent was not common in digital integrated circuit design at all. It impacts the layout strategy and complicates the verification process (as will be discussed in Chapter 12). In addition, generating, regulating, and distributing multiple supplies are non-trivial tasks.

A number of different design strategies exist with respect to the usage of multiple supply voltages. The first is to assign the voltage at the block/macro level (the so-called voltage island
approach). This makes particular sense in case some modules have higher performance/activity requirements than others (for instance, a processor’s data path versus its memory). The second and more general approach is to allow for voltage assignment all the way down to the gate level (“custom voltage assignment”). In general, this means that gates on the non-critical paths are assigned a lower supply voltage. Be aware that having signals at different voltage levels requires the insertion of level converters. It is preferable if these are limited in number (as they consume extra energy) and occur only at the boundaries of the modules.

Slide 4.26
With respect to multiple supply voltages, one cannot help wondering about the following question: If multiple supply voltages are employed, how many discrete levels are sufficient, and what are their values? This slide illustrates the potential of using three discrete voltage levels, as was studied by Tadahiro Kuroda [Kuroda, ICCAD’02]. Supply assignment to the individual logic gates is performed by an optimization routine that minimizes energy for a given clock period. With the main supply fixed at 1.5 V, providing a second and third supply yields a nearly twofold power reduction ratio.

A number of useful observations can be drawn from the graphs:

- The power minimum occurs for $V_2 \approx 1$ V and $V_3 \approx 0.7$ V.
- The minimum is quite shallow. This is good news, as this means that small deviations around this minimum (as caused, for instance, by $IR$ drops) will not have a big impact.

The question now is how much impact on power each additional supply carries.
Slide 4.27
In fact, the marginal benefits of adding extra supplies quickly bottom out. Although adding a second supply yields big savings, the extra reductions obtainable by adding a third or a fourth are marginal. This makes sense, as the number of (non-critical) gates that can benefit from the additional supply shrinks with each iteration. For example, the fourth supply works only with non-critical path gates close to the tail of the delay distribution. Another observation is that the power savings obtainable from using multiple supplies reduce with the scaling of the main supply voltage (for a fixed threshold).

Slide 4.28
Our discussion on multiple discrete supply voltages can be summarized with a number of rules-of-thumb:

- The largest benefit is obtained by adding a second supply.
- The optimal ratio between the discrete supplies is approximately 0.7.
- Adding a third supply provides an additional 5–10% incremental savings. Going beyond that does not make much sense.
Slide 4.29
Distribution of multiple supply voltages requires careful examination of the floorplanning strategy. The conventional way to support multiple $V_{DD}$’s (two in this case) is to place gates with different supplies in different wells (e.g., low-$V_{DD}$ and high-$V_{DD}$). This approach does not require a redesign of the standard cells, but comes with an area overhead owing to the necessary spacing between $n$-wells at different voltages. Another way to introduce the second supply is to provide two $V_{DDL}$ rails for every standard cell, and selectively route the cells to the appropriate supply. This “shared $n$-well” approach also comes with an area overhead owing to the extra voltage rail. Let us further analyze both techniques to see what kind of system-level trade-offs they introduce.

Slide 4.30
In the conventional dual-voltage approach, the most straightforward method is to cluster gates with the same supply (scheme b). This scheme works well for the “voltage island” model, where a single supply is chosen for a complete module. It does not very well fit the “custom voltage assignment” mode, though. Logic paths consisting of both high-$V_{DD}$ and low-$V_{DD}$ cells incur additional overhead in wire delay due to long wires between the voltage clusters. The extra wire capacitance also reduces the power savings. Maintaining spatial locality of connected combinational logic gates is essential.

Another approach is to assign voltages per row of cells (scheme a). Both $V_{DDL}$ and $V_{DDH}$ are routed only to the edge of the rows, and a special standard cell is added that selects between the two voltages (obviously, this applies only to the standard-cell methodology). This approach suits the “custom voltage assignment” approach better, as the per-row assignment provides a smaller granularity and the overhead of moving between voltage domains is smaller.
Slide 4.31
The most versatile approach is to redesign standard cells, and have both $V_{DDL}$ and $V_{DDH}$ fail inside the cell ("shared n-well"). This approach is quite attractive, because we do not have to worry about area partitioning – both low-$V_{DD}$ and high-$V_{DD}$ cells can be abutted to each other. This approach was demonstrated on a high-speed adder/ALU circuit by Shimazaki et al.

Slide 4.32
Level conversion is another important issue in designing with multiple discrete supply voltages. It is easy to drive a low-voltage gate from a high-voltage one, but the opposite transition is hard owing to extra leakage, degraded signal slopes, and performance penalty. It is hence worthwhile minimizing the occurrence of low-to-high connections.

As we will see in the next few slides, low-to-high level conversion is best accomplished using positive feedback – which is naturally present in flip-flops and registers. This leads to the following strategy: Every logical path starts at the high-voltage level. Once a path transitions to the low voltage, it never switches back. The next up-conversion happens in flip-flops. Supply voltage assignment starts from critical paths and works backward to find non-critical paths where the supply voltage can be reduced. This strategy is illustrated in the slide. The conventional design on the left has all gates operating at the nominal supply (the critical path is highlighted). Working backward from the flip-flops, non-critical paths are gradually converted to the low voltage until they become critical (gray-shaded gates operate at $V_{DDL}$). This technique of grouping is called “clustered voltage scaling” (CVS).
Slide 4.33
As the level-converting flip-flops play a crucial role in the CVS scheme, we present a number of flip-flops that can do level conversion and maintain good speed.

The first circuit is based on the traditional master–slave scheme, with the master and slave stages operating at the low and high voltages, respectively. The positive feedback action in the slave latch ensures efficient low-to-high level conversion. The high voltage node \( st \) is isolated from low-voltage node \( mo \) by the pass-transistor, gated by the low-voltage signal \( ck \).

The same concept can also be applied in an edge-triggered flip-flop, as shown in the second circuit (called the pulse-based half-latch). A pulse generator derives a short pulse from the clock edge, ensuring that the latch is enabled only for a very short time. This circuit has the advantage of being simpler.

Slide 4.34
Dynamic gates with NMOS-only evaluation transistors are naturally suited for operation with reduced logic swing, as the input signal does not need to develop a full high-\( V_{DD} \) swing to drive the output node to logic zero. The reduced swing only results in a somewhat longer delay. A dynamic structure with implicit level conversion is shown in the figure.

Observe that level conversion is also possible in an asynchronous fashion. A number of such non-clocked converters will be presented in a later chapter on Interconnect (Chapter 6). Clocked circuits tend to be more reliable, however.
implementation of this 64-bit ALU module, which is composed of the ALU, the loop-back bus driver, the input operand selectors, and the register files. For performance reasons, a domino circuit-style was adopted. As the carry generation is the most critical operation, circuits in the carry tree are assigned to the $V_{ddh}$ domain. On the other hand, the partial-sum generator and the logical unit are assigned to the $V_{ddl}$ domain. In addition, the bus driver, as the gate with the largest load, is also supplied from $V_{ddl}$. The level conversion from the $V_{ddl}$ signal to the $V_{ddh}$ signal is performed by the sum selector and the 9:1 multiplexer.

Slide 4.35
A real-life example of a high-performance Itanium-class (©Intel) data path helps to demonstrate the effective use of dual-$V_{dd}$. From the block diagram, it is apparent that the critical component from an energy perspective is the very large output capacitance of the ALU, which is due to its high fan-out. Hence, lowering the supply voltage on the output bus yields the largest potential for power reduction.

The shared-well technique was chosen for the

Slide 4.36
This schematic shows the low-swing loop-back bus and the domino-style level converter. Since the loop-back bus $sumb$ has a large capacitive load, low-voltage implementation is quite attractive. Some issues deserve special attention:

- One of the concerns of the shared-well approach is the reverse biasing on the PMOS transistor. As $sum$ is a monotonically rising signal (output of a domino stage), this does not impact the performance of the important gate $INV1$. 

- $INV2$ is placed near 9:1 MUX to increase noise immunity
- Level conversion is done by a domino 9:1 MUX
• In dynamic-logic designs, noise is one of the critical issues. To eliminate the effects of disturbances on the loop-back bus, the receiver INV2 is placed near the 9:1 multiplexer to increase noise immunity.
• The output of INV2, which is a $V_{DDL}$ signal, is converted $V_{DDH}$ by the 9:1 multiplexer. The level conversion is fast, as the precharge levels are independent of the level of the input signal.

**Slide 4.37**
This figure plots the familiar energy–delay plots of the ALU (as measured). The energy–delay curve for single-supply operation is drawn as a reference. At the nominal supply voltage of 1.8 V (for a 180 nm CMOS technology), the chip operates at 1.16 GHz. Introducing a second supply yields an energy saving of 33% at the small cost of 8% in delay increase. This example demonstrates that the theoretical results derived in the earlier slides of this chapter are actually for real.

**Slide 4.38**
Transistor sizing is the other high-impact design parameter we have explored at the circuit level so far. The theoretical analysis assumes a continuous sizing model, which is only a possibility in purely custom design. In ASIC design flows, transistor sizes are predetermined in the cell library. In the early days of application-specific integrated circuit (ASIC) design and automated synthesis, libraries used to be quite small, counting between 50 and 100 cells. Energy considerations have changed the picture substantially. With the need for various sizing options for each logical cell,
industrial libraries now count close to 1000 cells. As with supply voltages, it is necessary to move from a continuous model to a discrete one. Similarly, the overall impact on energy efficiency of doing so can be quite small.

**Slide 4.39**

In the ASIC design flow, it is in the “technology mapping” phase that the actual library cells are selected for the implementation of a given logical function. The logic network, resulting from “technology-independent” optimizations, is mapped onto the library cells such that performance constraints are met and energy is minimized. Hence, this is where the transistor (gate) sizing actually happens. Beyond choosing between identical cells with different sizes, technology mapping also gets to choose between different gate mappings: simple cells with small fan-in, or more complex cells with large fan-in. Over the last decade(s), it has been common understanding that simple gates are good from a performance perspective — delay is a quadratic function of fan-in. From an energy perspective, complex gates are more attractive, as the intrinsic capacitance of these is substantially smaller than the inter-gate routing capacitances of a network of simple gates. Hence, it makes sense for complex gates to be preferentially used on non-critical paths.
Slide 4.40
This argument is illustrated with an example. In this slide, we have summarized the area, delay, and energy properties of four cells (INV, NAND2, NOR2, NAND4) implemented in a 90 nm CMOS technology. Two different libraries are considered: a low-power and a high-performance version.

<table>
<thead>
<tr>
<th>Gate type</th>
<th>Area (cell unit)</th>
<th>Input cap. (fF)</th>
<th>Average delay (ps)</th>
<th>Average delay (ps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>INV</td>
<td>3</td>
<td>1.8</td>
<td>7.0 + 3.8C_L</td>
<td>12.0 + 6.0C_L</td>
</tr>
<tr>
<td>NAND2</td>
<td>4</td>
<td>2.0</td>
<td>10.3 + 5.3C_L</td>
<td>16.3 + 8.8C_L</td>
</tr>
<tr>
<td>NAND4</td>
<td>5</td>
<td>2.0</td>
<td>13.6 + 5.8C_L</td>
<td>22.7 + 10.2C_L</td>
</tr>
<tr>
<td>NOR2</td>
<td>3</td>
<td>2.2</td>
<td>10.7 + 5.4C_L</td>
<td>16.7 + 8.9C_L</td>
</tr>
</tbody>
</table>

(delay formula: \( C_L \) in fF)
(numbers calibrated for 90 nm)

Slide 4.41
These libraries are used to map the same function, an AND4, using either two-input or four-input gates (NAND4 + INV or NAND2 + NOR2). The resulting metrics show that the complex gate implementation yields a substantial reduction in energy and also reduces area. For this simple example, the complex-gate version is just as fast, if not faster. However this is due to the somewhat simplistic nature of the example. The situation becomes even more pronounced if the library would contain very complex gates (e.g., fan-in of 5 or 6).

<table>
<thead>
<tr>
<th></th>
<th>four-input AND</th>
<th>(a) NAND4 + INV</th>
<th>(b) NAND2 + NOR2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area</td>
<td>8</td>
<td>31.0 + 3.8C_L</td>
<td>32.7 + 5.4C_L</td>
</tr>
<tr>
<td>HS: Delay (ps)</td>
<td>11</td>
<td>53.1 + 6.0C_L</td>
<td>52.4 + 8.9C_L</td>
</tr>
<tr>
<td>LP: Delay (ps)</td>
<td>0.1</td>
<td>0.06C_L</td>
<td>0.83 + 0.06C_L</td>
</tr>
<tr>
<td>Sw Energy (fF)</td>
<td>0.06</td>
<td>0.06C_L</td>
<td>0.83 + 0.06C_L</td>
</tr>
</tbody>
</table>

- Area
  - Four-input more compact than two-input (two gates vs three gates)
- Timing
  - Both implementations are two-stage realizations
  - Second-stage INV (a) is better driver than NOR2 (b)
  - For more complex blocks, simpler gates will show better performance
- Energy
  - Internal switching increases energy in the two-input case
  - Low-power library has worse delay, but lower leakage (see later)

Slide 4.42
Technology mapping has brought us almost seamlessly to the next abstraction level in the design process – the logic level. Transistor sizes, voltage levels, and circuit style are the main optimization knobs at the circuit level. At the logic level, the gate–network topology to implement a given
Gate-Level Trade-offs for Power

- **Technology mapping**
  - Gate selection
  - Sizing
  - Pin assignment

- **Logical Optimizations**
  - Factoring
  - Restructuring
  - Buffer insertion/deletion
  - Don’t-care optimization

function is chosen and fine-tuned. The link between the two is the already discussed technology-mapping process. Beyond gate selection and transistor sizing, technology mapping also performs pin assignment. It is well known that, from a performance perspective, it is a good idea to connect the most critical signal to the input pin “closest” to the output node. For a CMOS NAND gate, for instance, this would be the top transistor of the NMOS pull-down chain. From a power reduction point of view, on the other hand, it is wise to connect the most active signal to that node, as this minimizes the switching capacitance.

The technology-independent part of the logic-synthesis process consists of a sequence of optimizations that manipulate the network topology to minimize delay, power, or area. As we have become used to, each such optimization represents a careful trade-off, not only between power and delay, but sometimes also between the different components of power such as activity and capacitance. This is illustrated with a couple of examples in the following slides.

**Slide 4.43**
In Chapter 3, we have established that the occurrence of dynamic hazards in a logic network is minimized when the network is balanced from a timing perspective – that is, most timing paths are of similar lengths. Paths of unequal length can always be equalized with respect to time in a number of ways: (1) through the restructuring of the network, such that an equivalent network with balanced paths is obtained; (2) through the introduction of non-inverting buffers on the fastest paths. The attentive reader realizes that although the latter helps to minimize glitching, the buffers themselves add extra switching capacitance. Hence, as always, buffer insertion is a careful trade-off process. Analysis of circuits
generated by state-of-the-art synthesis tools have shown that simple buffers are responsible for a considerable part of the overall power budget of the combinatorial modules.

Slide 4.44

**Factoring** is another transformation that may introduce unintended consequences. From a capacitance perspective, it seems obvious that a simpler logical expression would require less power as well. For instance, translating the function \( f = a \cdot b + a \cdot c \) into its equivalent \( f = a \cdot (b + c) \) seems a no-brainer, as it requires one less gate. However, it may also introduce an internal node with substantially higher transition probabilities, as annotated on the slide.

This may actually increase the net power. The lesson to be drawn is that power-aware logical synthesis must not only be aware of network topology and timing, but should – to the best possible extent – incorporate parameters such as capacitance, activity, and glitching. In the end, the goal is again to derive the pareto-optimal energy-delay curves, which we are now so familiar with, or to reformulate the synthesis process along the following lines: choose the network that minimizes power for a given maximum delay or minimizes the delay for a maximum power.

Slide 4.45

Based on the preceding discussions, we can now draw a clear set of guidelines for energy-delay optimization at the circuit and logical levels. An attempt of doing so is presented in this slide.

Yet, so far we have only addressed dynamic power. In the rest of the chapter we tackle the other important contributor of power in contemporary networks: leakage.
Considering Leakage at Design Time

- Considering leakage as well as dynamic power is essential in sub-100 nm technologies
- Leakage is not essentially a bad thing
  - Increased leakage leads to improved performance, allowing for lower supply voltages
  - Again a trade-off issue …

Leakage has so far been presented as an evil side effect of nanometer-size technology scaling, something that should be avoided by all cost. However, given an actual technology node, this may not necessarily be the case. For instance, a lower threshold (and increased leakage) allows for a lower supply voltage for the same delay – effectively trading off dynamic power for static power. This was already illustrated graphically in Slide 3.41, where power and delay of a logical function were plotted as a function of supply and threshold voltages. Once one realizes that allowing for an amount of static power may actually be a good thing, the next question inevitably arises: is there an optimal balance between dynamic and static power, and if so, what is the “golden” ratio?

Leakage – Not Necessarily a Bad Thing

\[
\frac{E_{lk}}{E_{sw}} = \frac{2}{\ln \left( \frac{L_k}{\alpha_{avg}} \right) - K}
\]

\(\text{Topo} | \text{Inv} | \text{Add} | \text{Dec} \)

\(E_{lk}/E_{sw}\)\(\text{opt} \) 0.8 0.5 0.2

Optimal designs have high leakage \((E_{lk}/E_{sw} = 0.5)\)
Must adapt to process and activity variations

[Ref: D. Markovic, JSSC’04]

Slide 4.47

The answer is an unequivocal yes. This is best illustrated by the graph in this slide, which plots the normalized minimum energy per operation for a given function and a given delay as a function of the ratio between static and dynamic power. The same curve is also plotted for a modified version of the same function.

A number of interesting observations can be drawn from this set of graphs:

- The most energy-efficient designs have a considerable amount of leakage energy.
  - For both the designs, the static energy is approximately 50% of the dynamic energy (or one-third of the total energy), and does not vary very much between the different circuit topologies.
  - The curves are fairly flat around the minimum, making the minimum energy somewhat insensitive to the precise ratio.

This ratio does not change much for different topologies except if activity changes by orders of magnitude, as the optimal ratio is a logarithmic function of activity and logic depth. Still, looking into significantly different circuit topologies in the last few slides, we found that optimal
ratio of the leakage-to-switching energy did not change much. Moreover, in the range defined by these extreme cases, energy of adder-based implementations is still very close to minimum, from 0.2 to 0.8 leakage-to-switching ratio, as shown in this graph. A similar situation occurs if we analyze inverter chain and memory decoder circuits assuming an optimal leakage-to-switching ratio of 0.5.

From this analysis, we can derive a very simple general result: energy is minimized when the leakage-to-switching ratio is about 0.5, regardless of logic topology or function. This is an important practical result. We can use this knowledge to determine the optimal $V_{\text{DD}}$ and $V_{\text{TH}}$ in a broad range of designs.

---

**Slide 4.48**

The effect of leakage is easily introduced in our earlier-defined optimization framework. Remember that the leakage current of a module is a function of the state of its inputs. However, it is often acceptable to use the average leakage over the different states. Another observation is that the ratio between dynamic and static energy is a function of the cycle time and the average activity per cycle.

---

**Slide 4.49**

When trying to manipulate the leakage current, the designer has a number of knobs at her disposition — In fact, they are quite similar to the ones we used for optimizing the dynamic power: transistor sizes, and threshold and supply voltages. How they influence leakage current is substantially different though. The choice of the threshold voltage is especially important.
Slide 4.50
While wider transistors obviously leak more, the chosen transistor length has an impact as well. As already shown in Slide 2.15, very short transistors suffer from a sharp reduction in threshold voltage, and hence an exponential increase in leakage current. In leakage-critical designs such as memory cells, for instance, it makes sense to consider the use of transistors with longer channel lengths rather than the ones prescribed by the nominal process parameters. This comes at a penalty in dynamic power though, but that increase is relatively small. For a 90 nm CMOS technology, it was shown that increasing the channel length by 10% reduces the leakage current by 50%, while raising the dynamic power by 18%. It may seem strange to deliberately forgo one of the key benefits of technology scaling – that is, smaller transistors – yet sometimes the penalty in area and performance is inconsequential, whereas the gain in overall power consumption is substantial.

Slide 4.51
Using multiple threshold voltages is an effective tool in the static-power optimization portfolio. In contrast to the usage of multiple supply voltages, introducing multiple thresholds has relatively little impact on the design flow. No level converters are needed, and no special layout strategies are required. The real burden is the added cost to the manufacturing process. From a design perspective, the challenge is on the technology mapping process, which is where the choice between cells with different thresholds is really made.
norm in the sub-100 nm technologies.

**Slide 4.52**
The immediate question is how many threshold voltages are truly desirable. As with supply voltages, the addition of more levels comes at a substantial cost, and most likely yields a diminishing return. A number of studies have shown that although there is still some benefit in having three discrete threshold voltages for both NMOS and PMOS transistors, it is quite marginal. Hence, two thresholds for both devices have become the

**Slide 4.53**
As was the case with dynamic power reduction, the strategy is to increase the threshold voltages in timing paths that are not critical, leading to static leakage power reduction at no performance and dynamic power costs. The appealing factor is that high-threshold cells can be introduced anywhere in the logic structure without major side effects. The burden is clearly on the tools, as timing slack can be used in a number of ways: reducing transistor sizes, supply voltages, or threshold voltages. The former two reduce both dynamic and static power, whereas the latter only influences the static component. Remember however that an optimal design carefully balances both components.
module. Yet even with these options, it is becoming increasingly apparent that dynamic logic is facing serious challenges in the extreme-scaling regimes.

### Slide 4.54
Most of the discussion on leakage so far has concentrated on static logic. I reckon that dynamic-circuit designers are even more worried: for them, leakage means not only power dissipation but also a serious degradation in noise margin. Again, a careful selection between low- and high-threshold devices can go a long way. Low-threshold transistors are used in the timing-critical paths, such as the pull-down logic

### Slide 4.55
Repeating what was stated earlier, the concept of multiple thresholds is introduced quite easily in the existing commercial design flows. In hindsight, this is clearly a no-brainer. The major impact is the size of the cell library doubles (at least), which increases the cost of the characterization process. This, combined with the introduction of a range of size options for each cell, has led to an explosion in the size of a typical library. Libraries with more than 1000 cells are not an exception.
**Slide 4.56**
In this experiment, performed jointly by Toshiba and Synopsys, the impact of the introduction of cells with multiple thresholds in a high-performance design is analyzed. The dual-threshold strategy leaves timing and dynamic power unchanged, while reducing the leakage power by half.

<table>
<thead>
<tr>
<th></th>
<th>High- $V_{TH}$ Only</th>
<th>Low- $V_{TH}$ Only</th>
<th>Dual- $V_{TH}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Slack</td>
<td>-53 ps</td>
<td>0 ps</td>
<td>0 ps</td>
</tr>
<tr>
<td>Dynamic Power</td>
<td>3.2 mW</td>
<td>3.3 mW</td>
<td>3.2 mW</td>
</tr>
<tr>
<td>Static Power</td>
<td>914 nW</td>
<td>3873 nW</td>
<td>1519 nW</td>
</tr>
</tbody>
</table>

All designs synthesized automatically using Synopsys Flows

[Courtesy: Synopsys, Toshiba, 2004]

---

**Slide 4.57**
A more detailed analysis is shown in this slide, which also illustrates the impact of the chosen design flow over a set of six benchmarks with varying complexity. It compares the high-$V_{TH}$ and low-$V_{TH}$ designs (the extremes) with a design starting from low-$V_{TH}$ transistors only followed by a gradual introduction of high-$V_{TH}$ devices, and vice-versa. It shows that the latter strategy – that is, starting exclusively with high-$V_{TH}$ transistors and introducing low-$V_{TH}$ transistors only in the critical paths to meet the timing constraints – yields better results from a leakage perspective.

[Selected combinational tests 130 nm CMOS]

[Courtesy: Synopsys 2004]
Slide 4.58
In earlier chapters, we have already introduced the notion that stacking transistors reduces the leakage current super-linearly primarily due to the DIBL effect. The stacking effect is an effective means of managing leakage current at design time. As illustrated in the graphs, the combination of stacking and transistor sizing allows us to maintain the on-current, while keeping the off-current in check, even for higher supply voltages.

Slide 4.59
This combined effect is put in a clear perspective in this graph, which plots the $I_{on}/I_{off}$ ratio of a transistor stack of 10 versus a single transistor as a function of $V_{DD}$. For a supply voltage of 1 V, the stacked transistor chain features an on-versus-off current ratio that is 10 times higher. This enables us to lower thresholds to values that would be prohibitive in simple gates. Overall, it also indicates that the usage of complex gates, already beneficial in the reduction of dynamic power, helps to reduce static power as well. From a power perspective, this is a win–win situation.
Complex Gates Increase $I_{on}/I_{off}$ Ratio

- Example: four-input NAND

With transistors sized for similar performance:
Leakage of Fan-in(2) = Leakage of Fan-in(4) x 3
(Averaged over all possible input patterns)

Slide 4.60
The advantage of using complex gates is illustrated with a simple example: a fan-in(4) NAND versus a fan-in(2) NAND/NOR implementation of the same function. The leakage current is analyzed over all 16 input combinations (remember that leakage is state-dependent). On the average, the complex-gate topology has a leakage current that is three times smaller than that of the implementation employing simple gates. One way of looking at this is that, for the same functionality, complex gates come with fewer leakage paths. However, they also carry a performance penalty. For high-performance designs, simple gates are a necessity in the critical-timing paths.

Example: 32-bit Kogge–Stone Adder

Reducing the threshold by 150 mV increases leakage of single NMOS transistor by a factor of 60

[Ref. S. Narendra, ISLPED’01]

Slide 4.61
The complex-versus-simple gate trade-off is illustrated with the example of a complex Kogge–Stone adder (from [Narendra, ISLPED’01]). This is the same circuit we studied earlier in this chapter. The histogram of the leakage currents over a large range of random input signals is plotted. It can be observed that the average leakage current of the low-$V_{TH}$ version is only 18 times larger than that of the high-$V_{TH}$ version, which is substantially smaller than what would be predicted by the threshold ratios. For a single NMOS transistor, reducing the threshold by 150 mV would cause the leakage current to go up by a factor of 60 (for the slope factor $n = 1.4$).
Summary

- Circuit optimization can lead to substantial energy reduction at limited performance loss
- Energy–delay plots are the perfect mechanisms for analyzing energy–delay trade-offs
- Well-defined optimization problem over $W$, $V_{DD}$, and $V_{TH}$ parameters
- Increasingly better support by today’s CAD flows
- Observe: leakage is not necessarily bad – if appropriately managed

Slide 4.62
In summary, the energy–delay trade-off challenge can be redefined into a perfectly manageable optimization problem. Transistor sizing, multiple supply and threshold voltages, and circuit topology are the main knobs available to a designer. Also worth remembering is that energy-efficient designs carefully balance the dynamic and static power components, subject to the predicted activity level of the modules. The burden is now on the EDA companies to translate these concepts into generally applicable tool flows.

References

Books:
- I. Sutherland, B. Sorel and D. Harris, Logical Effort: Designing Fast CMOS Circuits, Morgan-Kaufmann, 1st ed. 1999.

Articles:
References (cont.):

Chapter 5
Optimizing Power @ Design Time – Architecture, Algorithms, and Systems

The complexity of global optimization involving variables from all layers of the design-abstraction chain can be quite high. Fortunately, it turns out that many of the variables can be independently tuned, so a designer can partition optimization routines into smaller tractable problems. This modular approach helps gain insight into individual variables and provides a way to navigate top-level optimization through inter-layer interactions.

Slide 5.1
This chapter presents power-area-performance optimization at the higher levels of the design hierarchy – this includes joint optimization efforts at the circuit, architecture, and algorithm levels. The common goal in all these optimizations is to reach a global optimum in the power-area-performance space for a given design.

Slide 5.2
The goal of system-level power (energy) optimizations is to transform the energy-delay space such that a broader range of options becomes available at the logic or circuit levels. In this chapter, we classify these transformations into a number of classes: the usage of concurrency, considering alternative topologies for the same
function, and eliminating waste. The latter deserves some special attention. To reduce the non-recurring expenses and to encourage re-use, programmable architectures are becoming the implementation platform of choice. Yet, this comes at a huge expense in energy efficiency. The exploration of architectures that combine flexibility and efficiency is the topic of the last part of this chapter.

**Slide 5.3**

The main challenge in hierarchical optimization is the interaction between the layers. One way to look at this is that optimizations at the higher abstraction layers enlarge the optimization space, and allow circuit-level techniques such as supply voltage or sizing to be more effective. Other optimizations may help to increase the computational efficiency for a given function.

**Slide 5.4**

Consider the energy–delay design space exploration exploiting size as well as supply and threshold voltages as parameters, as discussed in Chapter 4. For a 64-bit tree adder and a given technology, a pareto-optimal energy–delay curve is obtained showing some nice energy or delay improvements over the reference design. Yet the overall optimization space is restricted by the topology of the adder. Larger energy savings could be obtained by choosing a different adder topology such as a ripple adder. To accomplish these larger gains (both in delay and energy), accompanying transformations at the micro-architecture or system architecture level are needed. Over the past decades, it has been shown that this can lead to orders of magnitude in energy-efficiency improvement — quite impressive compared to the 30% range that is typically obtained at the circuit level. In this chapter, we present a methodological approach to extend the techniques introduced so far to the higher abstraction layers.
Slide 5.5
The digital design abstraction stack is pictured in this slide. So far, we have mostly covered the device, circuit, and logic levels. Now, we will explore the micro-architecture and architecture levels. While software- and system-level optimizations may have a huge impact as well, they are somewhat out of the scope of this text, and are discussed only in passing.
To make higher-level exploration effective however, it is essential that information from the lower-level layers percolates upward and is available as information to the architecture or system designer. For example, the energy–delay curve of a given adder in a given technology determines the design space that can be offered by that adder (overall, its design parameters). Here the information propagates “bottom-up”. At the same time, the application at hand imposes constraints on the implementation (such as, for instance, the minimum performance or maximum energy). These constraints propagate in the “top-down” fashion. Exploration at a given level of design abstraction hence becomes an exercise in marrying the top-down constraints to the bottom-up information. This process is called “meet-in-the-middle”.

Slide 5.6
The design parameters at the circuit level were mostly continuous (sizing, choice of threshold and/or supply voltage). At the higher abstraction levels, the choices are rather more discrete: which adder topology to use, how much pipelining to introduce, etc. From an exploration perspective, these discrete choices help to expand the energy–delay space. For instance, when two adder topologies are available, each of them comes with its own optimal energy–delay curve. The design space is now the combination of the two, and a new optimal E–D curve emerges as shown in trade-off plot 2. In some cases, one version of a
function is always superior to another, which makes the selection process very simple (trade-off plot 1). This may not be obvious at a first glance, and may only be revealed after a rigorous analytical inspection. Using an informed design-space exploration, we will demonstrate in this chapter that some adder topologies are inferior under all circumstances and should never be used (at least, not in modern CMOS processes). A third scenario is where the exploration space consists of many discrete options (such as the sizes and the number of register files). In this case, we can derive an optimal composite E–D curve by selecting the best available option for each performance level (trade-off plot 3).

Although the E–D space represents one very interesting projection of the overall design space, the designer should be aware that the overall design space is far more complex, and that other metrics such as area and design cost are relevant as well. On the first two scenarios, the gray arrows points towards implementations with smaller area. In most cases, implementations with lower energy also are smaller in size, but that is not necessarily always the case. It holds however for most instances of the first two exploration scenarios. This is indicated on the graphs by the gray arrows, which point toward smaller implementations.

Complex systems are built through by composing a number of simpler modules in a hierarchical fashion. Deriving the E–D curves of the composition can be quite involved. The energy component is purely additive, and hence quite straightforward. Delay analysis may be more complex, yet is very well understood.

**Slide 5.7**

A first (and famous) example of architectural energy–delay trade-off is the exploitation of concurrency to enable aggressive supply voltage scaling [Chandrakasan, JSSC’92], or equivalently, to provide improved performance at a fixed energy per operation (EOP). To demonstrate the concept, we start from a simple reference design that operates at a nominal (reference) supply voltage $V_{dd,ref}$ and a frequency $f_{ref}$. The average switched capacitance of this design is $C_{ref}$.

**Slide 5.8**

A parallel implementation of the same design essentially replicates the design such that parallel branches process interleaved input samples. Therefore, the inputs coming into each parallel branch are effectively down-sampled. An output multiplexer is needed to recombine the outputs, and produce a single data stream.
Owing to the parallelism, branches now can operate at half the speed, hence $f_{\text{par}} = \frac{f_{\text{ref}}}{2}$. This reduced delay requirement enables a reduction of the supply voltage by a factor $\epsilon_{\text{par}}$. It is the squared effect of that reduction that makes this technique so effective. The multiplexing overhead is typically small, especially when parallelism is applied to large blocks. Notice that though the overhead in switching capacitance is minimal, the area overhead is substantial (effectively larger than the amount of concurrency introduced).

**Example: Parallelism in 90 nm Technology**

![Graph showing the relationship between $f_P$ and $V_{DD}$](image)

Assuming $\epsilon_{\text{par}} = 7.5\%$

$$P_{\text{par}} = \frac{0.003 \cdot 2.15}{2} \cdot P_{\text{ref}} = 0.47P_{\text{ref}}$$

$$P_{\text{par}} = \frac{0.52^2 \cdot 4.3}{4} \cdot P_{\text{ref}} = 0.29P_{\text{ref}}$$

The impact of introducing concurrency to reduce EOP for a fixed performance hinges on the delay–supply voltage relationship. For a 90 nm technology, increasing the delay by a factor of 2 is equivalent to reducing the supply voltage by a factor of 0.66. Squaring this translates into an energy reduction by a factor of more than one half (including the overhead). Increasing the parallelism by another factor of 2 reduces the required supply voltage to 0.52 V, resulting in even higher power savings (71%).

**Slide 5.10**

Other forms of introducing concurrency can be equally effective in reducing the supply voltage, and hence reducing power dissipation. An example of such is pipelining, which improves throughput at the cost of latency by inserting extra registers between logic gates. The area overhead of pipelining is much smaller than that of parallelism – the only cost being the extra registers, compared to replicating the design and adding multiplexers. However, owing to the
A Pipelined Implementation

\[ f_{\text{pipe}} = f_{\text{ref}} \]
\[ C_{\text{pipe}} = (1 + \alpha V_{\text{pipe}}) C_{\text{ref}} \]
\[ V_{D\text{Dpipe}} = \varepsilon_{\text{pipe}} \cdot V_{\text{Dref}} \]

Shallower logic reduces required supply voltage
(this example assumes equal \( V_{\text{DD}} \) for parallel/pipeline designs)

Assuming \( \alpha V_{\text{pipe}} = 10\% \)

\[ P_{\text{pipe}} = \varepsilon_{\text{pipe}}^2 \cdot (1 + \alpha V_{\text{pipe}}) \cdot P_{\text{ref}} \]
\[ P_{\text{pipe}4} = 0.52^2 \cdot 1.1 \cdot P_{\text{ref}} = 0.29 P_{\text{ref}} \]

extra switching capacitance introduced by the registers (and the extra clock load), pipelined implementations typically come with a higher switched capacitance than parallel designs. Assuming a 10\% pipelining overhead, power savings are similar to those obtained with parallelism. The area cost is substantially lower, though.

Increasing Use of Concurrency Saturates

- Can combine parallelism and pipelining to drive \( V_{\text{DD}} \) down
- But close to process threshold, overhead of excessive concurrency starts to dominate

Slide 5.11

As we have learned from our earlier discussions on circuit-level optimization, the effect of reducing the supply voltage quickly saturates – especially when the \( V_{\text{DD}}/V_{\text{TH}} \) ratio gets small. Under those conditions, a small incremental reduction in \( V_{\text{DD}} \) translates into a large increase in delay, which must be compensated by even more concurrency. As shown for a typical 90\( \text{nm} \) technology, concurrency levels higher than eight do little to further improve the power dissipation.

Slide 5.12

The reality is even worse. The introduction of concurrency comes with an overhead. At low voltage levels (and hence high levels of concurrency), that overhead starts to dominate the gain made by the further reduction in supply voltage, and the power dissipation actually increases anew. Leakage also has a negative impact, as parallelism decreases the activity factor. The
The presence of a large number of gates with long delays tends to emphasize static over dynamic power.

The only way to keep increasing concurrency levels (and hence the EOP) is to reduce the threshold as well allowing for a further reduction in voltage without a major performance penalty. However, this requires a careful management of the leakage currents, which is non-trivial (as you must be convinced about by now).

**Slide 5.13**

The overall impact of the introduction of concurrency and its potential benefits are best understood in our familiar energy–delay space. In this plot, we have plotted the optimal energy–delay curves for an ALU design implemented with varying degrees of concurrency (for each implementation, the pareto-optimal curve is obtained using the techniques described in Chapter 4). Again, the curves can be combined to yield a single optimal E–D curve.

Two different optimization scenarios using concurrency can be considered:

- Fixed Performance: Adding concurrency reduces the EOP until a given point at which the overhead starts to dominate. Hence, for every performance level there exists an optimum level of concurrency that minimizes the energy.
- Fixed EOP: Introducing concurrency helps to improve performance at no EOP cost. This is in contrast to the traditional approach, where an increase in performance is equivalent to higher clock frequencies, and hence larger dynamic power.

It is interesting to observe that each design instance is optimal over a limited delay range. For instance, if the requested throughput is small, using a high level of parallelism is an inferior option,
as the overhead dominates. The attraction of the energy–delay curve representation is that it allows
the designer to make architectural decisions in an informed way.

**Slide 5.14**

The question now arises as to what to do when the requested throughput is really low (for instance, in the case of the microwatt nodes described in Chapter 1). This is especially valid with scaling of technology, where the speed of the transistors may be more than what is needed for a given application, and the nominal—that is, no concurrency—implementation is still too fast. In such cases, the solution is to introduce the inverse of concurrency, which is time-multiplexing, to trade off the excess speed for reduced area.

**Slide 5.15**

Reverting to energy–delay space, we observe that with parallelism and time-multiplexing we can span a very wide range on the performance axis. Relaxed-delay (low-throughput) targets prefer time-multiplexed solutions, whereas increased concurrency is the right option when high throughput is needed. One additional factor to bring to the game is the area cost, which we would like to minimize for a given set of design constraints. Let us consider different scenarios:

- For a given maximum delay ($D_{\text{target}}$ in the figure): if the goal is to minimize the EOP, then there exists an optimum amount of concurrency (= 1/2); on the other hand, if the goal is to minimize the area for a given EOP, a smaller amount of concurrency is desirable (= 1/5).
- For a given maximum of EOP, we choose the amount of concurrency that meets the minimum performance and minimizes the area, as indicated by the red and blue curves on
the plot. In this scenario, concurrency and time-multiplexing provide an efficient way to trade off throughput for area.

Slide 5.16
To summarize, maximum-performance designs request the maximum possible concurrency at the expense of increased area. For a given performance, however, one should optimize the amount of concurrency to minimize energy.

Equivalently, for a given energy budget, the least amount of concurrency that meets the performance goals should be used. For the absolute minimum energy, a direct mapping architecture (concurrency = 1) should be used because it has no overhead in switched capacitance, provided of course that this architecture meets the design constraints.

Slide 5.17
The ideas that were put forward in the previous slides originated in the early 1990s. Yet it took some time for them to be fully embraced by the computing community. That happened only when people started to realize that the traditional performance improvement strategies for processors – that is, technology scaling combined with increasing the clock frequency – started to yield less, as is shown in this graph. It was mostly power constraints that slowed down the increases in clock frequency, and ultimately conspired to halt it altogether. Remember how clock frequency was the main differentiator in the advertisements of new processors in the 1980s and 1990s? With stalling clock frequencies, the only way to maintain the scaling of the performance of the single processor was to increase the instructions per cycle (IPC) through the introduction of extra architectural performance-enhancing techniques such as multi-threading and
speculation. All of these add to the complexity of the processor, and come at the expense of energy efficiency. (Caveat: this is an in-a-nutshell summary – the real equation is a lot more complex).

**Slide 5.18**
The reality of these concerns is quite clearly illustrated in this three-dimensional chart, which plots three Intel processor families in the power–clock frequency–Spec2K/MHz space. The latter metric measures the effective performance of a processor, independent of the clock frequency. It shows that ultimately the effective performance of the processors within a single family increased little over time, and that clock frequency was the primary tuning knob. Unfortunately this translated directly into massive increases in power dissipation.

**Slide 5.19**
After a slow adoption, the idea of using concurrency as the way out of the quagmire gathered full steam in the 2000s, when all major microprocessor vendors agreed that the only way to improve performance within a given power envelope is to adopt concurrency, leading to the multitude of multi-core architectures we are seeing today.
Slide 5.20

This slide just shows a sample of the many multi-core architectures that were introduced starting 2005. Initially adopted in application-specific processors (telecommunications, media processing, graphics, gaming), the multi-core idea spread to general-purpose processing starting with the dual-core architecture, expanding rapidly to four and more cores on a die thereafter.

Slide 5.21

There is one important caveat though. Conceptually, we can keep on improving the performance (for a given technology) at a fixed EOP cost by providing more concurrency. This requires however that the concurrency is available in the application(s) at hand. From the (in)famous Amdahl’s law, we know that the amount of speed-up attainable through concurrency is limited by the amount of serial (non-parallelizable) code. Even if only 20% of the code is sequential, the maximum amount of performance gain is limited to just a little more than 4.

Slide 5.22

The impact of this is clearly illustrated in the following case study performed at Intel. A set of benchmarks is mapped on three different multi-core architectures with different granularity and different amount of concurrency (12 large, 48 medium, or 144 small processors). Each realization is such that it occupies the same area (13 mm on the side), and dissipates the same maximum power (100 W) in a speculative 22 nm CMOS technology. The large processors are more powerful from a computational throughput perspective, but operate at a higher EOP cost.
When comparing the overall performance of the three alternatives over a set of benchmarks with different levels of concurrency, it shows that large-processor version outperforms the others when little parallelism is available. However, when plenty of concurrency is present, the “many small cores” option outperforms the others. This is especially the case for the TPT (totally parallel) benchmark.

In summary, it becomes clear that “massively parallel” architectures only pay off for applications where there is sufficient concurrency. There is no clear answer about the right granularity of the computational elements. It is fair to assume that architectures of the future will combine a variety of processing elements with different granularities (such as, for instance, the IBM cell processor\textsuperscript{TM}, or the Xilinx Vertex\textsuperscript{TM}).

**Slide 5.23**

One option to improve the odds is to make the applications more parallel with the aid of optimizing transformations. Loop transformations such as loop unrolling and loop retiming/pipelining are the most effective. Algebraic transformations (commutativity, associativity, distributivity) come in very handy as well. Automating this process is not easy though. The (dormant, but reviving) field of high-level synthesis actually created some real breakthroughs in this area, especially in the domain of signal processing. In the latter, an infinite loop (i.e., time) is always present, allowing for the creation of almost limitless amounts of concurrency [Chandrakasan, TCAD’95].

For example, the loop-unfolding transformation translates a sequential execution (as captured in a recursive loop) into a concurrent one by unrolling the loop a number of times and introducing pipelining. The reverse is obviously also possible.
Slide 5.24
The effectiveness of today’s optimizing compilers in exposing concurrency is illustrated with the well-known example of MPEG-4 video encoder. The hierarchical composition of the computational core of MPEG-4, that is the motion compensation, is shown on the left. On the right, the amount of available parallelism in the reference design is visualized. White represents fully concurrent loops, whereas black stands for fully sequential ones. Gray represents a loop that is only partially concurrent. Also important are the “spurious dependencies”, which prevent the code from being concurrent, but only occur intermittently, or even may be false. It is quite obvious that the reference design, as it is, is not very friendly for concurrent architectures.

Slide 5.25
The application of a single transformation (such as pointer analysis or disambiguation) can make some major inroads, but it is only the combined effect of multiple transformations executed in concert that can make the code almost fully concurrent (which is an amazing accomplishment in itself). The reader should realize that the latter cannot be performed automatically right now, but needs the manual intervention of the software developer. It is not clear when, if ever, fully automatic parallelization of sequential code will become feasible. Very often, a high-level perspective is needed, which is hard to accomplish through localized transformations. User intervention guided by the appropriate visualization may be a necessity. Even better would be to train software engineers to write
concurrent code from scratch. This is best accomplished through the use of programming environments that make concurrency explicit (such as the Mathworks Simulink™ environment).

In summary, concurrency is a great tool to keep energy in check, or to even reduce it. It requires a rethinking however on how complex design is done.

**Slide 5.26**

Beyond the introduction of concurrency, other architectural strategies can be explored, such as considering alternative implementation topologies for a given function. For each topology, an energy–delay curve can be obtained in a bottom-up fashion. The architectural exploration process then selects the most appropriate implementation for a given set of performance or energy constraints.

The E–D curves of the alternative topologies can be combined to define a composite, globally optimum, trade-off curve for the function (bottom graph). The boundary line is optimal, because all other points consume more energy for the same delay, or have longer delay for the same energy.

**Slide 5.27**

Consider, for instance, the case of the adder module, which is often the most performance- and energy-critical component in a design. The quest for the ultimate adder topology has filled many conference rooms, journal articles, and book chapters. The choices are between ripple, carry-select, carry-bypass, and various styles of look-ahead adders (amongst others). Within each of these categories, an extra number of design choices must be made, such as the exact topology (e.g., the radix in a look-ahead adder) or the circuit style to be used. Although it seems that the number of options is overwhelming, using some hierarchy in
the decision-making process makes the selection process a lot more amenable, and may even lead to the establishment of some fundamental selection rules and some definitive truths.

Let us consider the case of a 64-bit carry look-ahead adder. Design parameters include the radix number (fan-out in the tree) as well as the circuit style. Optimizing transistor sizes and supply voltages (using the methodology introduced in Chapter 4), optimal energy–delay curves can be provided for each option, as shown in the slide. As a result, the decision process is substantially simplified: For high performance, use higher-radix solutions and dynamic logic, whereas static logic and lower-radix numbers are preferred for low-energy solutions. The beauty of this E–D exploration is that designers can rely on objective comparisons to make their decisions.

**Slide 5.28**
The same approach equally applies to higher levels of the decision hierarchy. Even with the carry look-ahead topology, many options exist, an example of which is the Ling adder that uses some logic manipulations to get even higher performance. Again, some ground truths can be established. Radix-2 CLA adders rarely seem to be attractive. If performance is the primary goal, then radix-4 Ling adder is the preferred choice.

The bottom line of this discussion is that energy–delay trade-off can be turned into an engineering science, and should not be a black art.
Improving Computational Efficiency

Implementations for a given function maybe inefficient and can often be replaced with more efficient versions without penalty in energy or delay

Inefficiencies arise from:
- Over-dimensioning or over-design
- Generality of function
- Design methodologies
- Limited design time
- Need for flexibility, re-use, and programmability

Knowledge or lack of exploration capabilities and most often because of adherence to generally accepted design paradigms or methodologies.

Slide 5.29
The plots in the previous slides already indicate that some architectural options are truly inferior in all aspects, and hence should be discarded from the designer’s consideration. Such inefficiencies may arise from a number of reasons including legacy or historical issues, over-design, use of inferior logic style, inferior design methodology, etc. Although this may seem obvious, it is surprising that inefficient options are still in frequent use, often because of lack of

Slide 5.30
Some simple and general guidelines are valid when it comes to improving computational efficiency. Whereas the slide enumerates a number of concepts that are worked out in more detail in subsequent pages, it is worth condensing this to an even smaller number of ground truths.

- Generality comes with a major penalty in efficiency.
- It pays to have the architecture match the intent of the computation.
- Never have anything consume power when it is not in use.

Slide 5.31
Consider first the issue of matching computation to architecture. To illustrate this, let us consider a simple example of computing a second-order polynomial. In a traditional Von–Neumann style processor, the computation is taken apart into a set of sequential instructions, multiplexed on a generic ALU. The architecture of the computational engine and the topology of the algorithm are
Matching Computation and Architecture

- Choice of computational architecture can have major impact on energy efficiency (see further)

Example: Compute $y = A \cdot x^2 + B \cdot x + C$

or

$t_1 \leftarrow x$
$t_2 \leftarrow A \cdot t_1$
$t_3 \leftarrow t_2 \times B$
$t_4 \leftarrow t_3 \times t_1$
$y \leftarrow t_4 \times C$

programming”. As we will show later, the difference in energy efficiency between the two is huge.

Slide 5.32
Another example of the matching between algorithm and architecture is the choice of the word length. Most programmable processors come with a fixed word length (16, 32, or 64 bit), although the actual computation may need far less. During execution, this leads to a sizable amount of switching energy (as well as leakage) being wasted. This is avoided if the word length of the computational engine can either be matched or adjusted to the algorithmic needs.

Slide 5.33
To illustrate the impact of the architectural optimizations discussed so far and their interaction, we use a case study of a singular-value decomposition (SVD) processor for multiple-input and multiple-output (MIMO) communications [D. Marković, JSSC’07]. Multi-antenna techniques are used to improve robustness or increase capacity of a wireless link. Link robustness is improved by averaging the signal over multiple propagation paths as shown in the illustration. The number of averaging paths can be artificially increased by sending the same signal over multiple antennas. Even more aggressively, the
This algorithm is however quite complex, and involves hundreds of additions and multiplications, as well as divisions and square roots, all of which have to be executed real-time at the data rate (which is in the range of 100s of MHz). This far exceeds the complexity of standard communication blocks such as FFT or Viterbi en(de)coding. The challenge is to come up with an architecture that is both energy- and area-efficient. In this particular case, we study a multi-antenna algorithm that can achieve around 250 Mbps over 16 frequency sub-channels using a 4×4 antenna system.

Slide 5.34
This slide illustrates how various optimization techniques are used to reach the target speed with minimal power and area. The process starts with a fully parallel implementation, which is both very large and too fast. The excess performance is traded for area and energy reduction. Qualitatively, word-length optimization reduces both area and energy; interleaving and folding mainly impact area and have a small impact on energy (neglected in this simplified diagram); gate sizing primarily affects the energy (small change in area of standard-cell based design); and, finally, voltage scaling has a major impact on energy.
synthesis, which includes gate sizing and supply voltage optimizations. From prior discussions, we know that sizing is most effective at small incremental delays compared to the minimum delay, so we synthesize the design with 20% slack and perform incremental compilation to utilize benefits of sizing for a 40% reduction in energy and a 20% reduction in area. Standard cells are characterized for 1V supply, so we translate timing specifications to that voltage. At the optimal $V_{DD}$ and $W$, energy–delay curves of sizing and $V_{DD}$ are tangential, which means that the sensitivities are equal. The final design is 64 times smaller and consumes 16 times less energy than the original 16-bit direct-mapped parallel realization.

**Slide 5.35**

The Energy–Delay–Area diagram is a convenient way to look at the combined effect of all optimization steps. As the slide shows, the major impact on energy comes from supply voltage scaling and gate sizing, whereas area is primarily reduced by interleaving and folding.

The process proceeds as follows: Starting from a 16-bit realization of the algorithm, word-length optimization yields a 30% reduction in energy and area. The next step is logic synthesis, which includes gate sizing and supply voltage optimizations. From prior discussions, we know that sizing is most effective at small incremental delays compared to the minimum delay, so we synthesize the design with 20% slack and perform incremental compilation to utilize benefits of sizing for a 40% reduction in energy and a 20% reduction in area. Standard cells are characterized for 1V supply, so we translate timing specifications to that voltage. At the optimal $V_{DD}$ and $W$, energy–delay curves of sizing and $V_{DD}$ are tangential, which means that the sensitivities are equal. The final design is 64 times smaller and consumes 16 times less energy than the original 16-bit direct-mapped parallel realization.

**Slide 5.36**

The measured performance data of a chip implementation of the SVD algorithm are shown in this slide. Implemented in a 90nm CMOS technology, the optimal supply voltage is 0.4V for a 100 MHz clock. The chip is actually functional all the way down to 255mV, running with a 10 MHz clock. The leakage power is 12% of the total power in the worst case, and clocking power is 14mW, including leakage.

A comparison against a number of custom chips
from the multimedia and wireless tracks of the ISSCC conference shows how this combined set of optimizations leads to a design that simultaneously excels in area and energy efficiency. Publication year and paper number are indicated; figures are normalized to a 90 nm, 1 V process.

Slide 5.37
Maintaining locality of reference is another mechanism to increase the efficiency of an architecture. This in fact not only is true for power, but also helps performance and area (in other words, a win–win). Anytime a piece of data or an instruction has to be fetched from a long distance, it comes at a cost in energy and delay. Hence, keeping relevant or often-used data and instructions close to the location where they are processed is a good idea. This is, for instance, the main motivation behind the construction of memory hierarchies, such as multi-level caches.

Slide 5.38
The instruction loop buffer (ILB) is an example of how locality of reference is effectively used in Digital Signal Processors (DSPs). Many DSP algorithms such as FIR filters, correlators, and FFTs can be described as short loops with only a few instructions. Rather than fetching the instructions from a large instruction memory or cache, it is far more energy-efficient to load these few instructions into a small buffer memory on the first execution of the loop, and fetch them from there on the subsequent iterations.

Slide 5.39
Similar considerations are true for data locality. Sometimes a careful reorganization of the algorithm or code is sufficient to realize major benefits – without needing any extra hardware
Intermediate storage requirements are reduced substantially (that is, to a single line), leading to both energy reduction and performance improvement.

**Slide 5.40**

Even on generic architectures such as the Pentium, the impact of optimizations for data locality can be huge. Starting from generic reference code for MPEG-4, memory accesses can be reduced by a factor of 12 through a sequence of software transformations. This translates almost directly into energy savings, simultaneously improving the performance by a factor of almost 30.

**Slide 5.41**

Architectural optimizations can also be used to minimize activity – an important component of the dynamic power component. In fact, when taking a close look at many integrated circuits, we can discover a large amount of spurious activity which little or even zero computational meaning, and some of it being a direct consequence of architectural choices.

A simple example can help to illustrate this. Many data streams exhibit temporal correlations. Examples are speech, video, images, sensor data, etc. Under such conditions, the probability of a...
bit to undergo a transition from sample to sample is substantially smaller than when the data is purely random. In the latter case, the transition probability per bit would be exactly 1/2. Consider now the case of an N-bit counter. The average transition probability (per bit) equals 2/N (for large N), which is substantially lower than the random case for N>4.

Time multiplexing two or more unrelated streams over the same bus destroys these correlations, and turns every signal into a random one, hence maximizing the activity. When data is very strongly correlated (and the load capacitance is large), it is often advisable to avoid multiplexing.

30% reduction in signal activity

Another example on how data correlations can help to reduce activity is shown in this slide. In signal processing applications, multiplication with a fixed number is often replaced by a sequence of adds and shifts. This avoids the use of expensive multipliers, is faster, and saves energy. The order in which we add those numbers (which is totally arbitrary owing to the associativity of the add function) impacts the activity in the network. In general, it is advisable to combine closely correlated signals first (such as \( x >> 8 \) and \( x >> 7 \)), as this preserves the correlation and minimizes spurious activity.
Slide 5.43
One dominant source of energy inefficiency in contemporary integrated circuits and systems is the provision for “flexibility” or “programmability”. Although programmability is a very attractive proposition from an economic and business perspective, it comes at a huge efficiency cost. The design challenge is to derive effective architectures that combine flexibility and programmability with computational efficiency.

Slide 5.44
Quantifying flexibility is not simple. The only real way to evaluate the flexibility of a given architecture is to analyze its performance metrics over a set of representative applications, or benchmark sets. A fitting framework is our energy–delay graph, extended with an extra “application axis”. A dedicated or custom architecture only executes a single application, and yields a single energy–delay curve. For a more flexible architecture, energy–delay curves can be generated for each member of the test bench. One (preferred) option is then to have the average of these curves represent the architecture. Another option is to use the circumference of the E–D plots over all individual applications.

Slide 5.45
Whatever representation is chosen, it is important to realize that the energy–delay curves of the same application implemented in custom style and on a programmable processor are generally quite far apart. It has been reported a number of times that for the same performance the EOP (measured in MIPS/mW or billion operations/Joule) for different implementation styles can be as much as three orders of magnitude apart, with custom implementation and general-purpose
programmable logic at the extreme ends. To fill in the gap between the two, designers have come up with some intermediate approaches such as application-specific instruction processors (ASIPs) and DSPs, which trade generality for energy efficiency. For instance, by adding dedicated hardware and instructions, a processor can become more efficient for a specific class of applications at the expense of others. Even closer to dedicated hardware are the configurable spatial programming approaches, in which dedicated functional units are reconfigured to perform a given function or task.

Slide 5.46
T. Claasen and H. De Man observed similar ratios in their ISSCC keynotes. Observe that these graphs project that the extremes grow even further apart with the scaling of technology.

Slide 5.47
The trade-off between flexibility and energy efficiency is beautifully illustrated in this example, which explores a variety of implementations of correlators for CDMA (Code-division multiple access), as used in 3G cellular telephony. All implementations are normalized to the same technology node. Implementing this compute-intensive function as a custom module reduces the energy per execution by a factor of approximately 150 over a software implementation on a DSP. Observe that the area cost is reduced substantially as well. A reconfigurable solution gets closer to the custom one, but still needs six times more energy.
An energy-conscious person may wonder who would ever use such inefficient implementations. However, the design trade-off space is much broader than just performance and energy. The addition of software programming adds an abstraction layer that allows a much wider range of developers to map applications, while reducing the time-to-market and risk. In fact, with nanometer design and fabrication becoming ever more complex and challenging, the trend is clearly toward ever more flexibility and programmability. The challenge for the micro-architecture and system-on-a-chip designer now is how to effectively combine the two. We have already mentioned the use of concurrent architectures. However, the range of options and choices is much broader than that. Without a structured exploration strategy, solutions are mostly chosen in an ad hoc fashion, and decisions are made based on rules of thumb or pure intuition.

### Slide 5.48
To illustrate the process, let us first consider the case of a single-instruction processor to be used in a system-on-a-chip. The first option is to use a generic processor, which can be obtained either for free, or from an intellectual-property (IP) company. Even within that constrained framework, a range of options exist, such as choosing the width of the data path or the structure of the memory architecture. If energy efficiency is crucial, further options should be considered as well, some of which are indicated on this slide and are elaborated on in the subsequent ones.

Given the potential benefits, industry has been intensely exploring all these options (and many others). Companies in the networking, communications, and signal processing domains all have their own (and different) processor recipes. Numerous start-ups have had high hopes that their
unique concept presents the perfect solution (and the road to richness). Only a few have succeeded, though.

**Slide 5.49**
Consider first the case of a single generic processor designed to cover a set of representative benchmarks. To determine whether a simple or a more complex processor is the best choice for a given performance or EOP specification, a structured-exploration approach can be used. Given the relevant design parameters, a set of meaningful instances are generated, synthesized, and extracted (courtesy of David Blaauw). The energy-delay metrics are found by simulating the obtained instances over the benchmark set, and averaging the results.

**Slide 5.50**
Each of the processor instances is now a single point in the architecture’s energy-delay space. With sufficient representative points, we see the common hockey-stick pareto-optimal curve emerge. A couple of quick observations: for the absolute lowest energy, choose the simplest processor; for the highest performance, go for the complex one. Also, a large number of processor options are inferior from any perspective, and should be rejected right off the bat.

The results shown in this slide suggest again that the energy-delay trade-off game takes on identical formats at all levels of the design hierarchy.
**Application-Specific Processors**

- Tailored processor to be efficient for a sub-set of applications
  - Memory architecture, interconnect structure, computational units, instructions

- Digital signal processors best known example
  - Special memory architecture provides locality
  - Datapath optimized for vector-multiplication (originally)

- Examples now available in many other areas (graphics, security, control, etc.)

---

**Example 1: DSPs**

- The first type of application-specific processor to become popular
- Initially mostly for performance, but energy benefit also recognized now
- Key properties: dedicated memory architecture (multiple data memories), data path specialized for specific functions such as vector multiplies and FFTs
- Over time: introduction of more and more concurrency (VLIW)

---

**Slide 5.51**

To further improve energy and extend the energy-delay space, the processor can be fine-tuned to better fit the scope of a sub-set of applications (such as communications, graphics, multimedia, or networking). Often, recurring operations or functions are captured in dedicated hardware in the data path, and special instructions are introduced. In addition, register files, interconnect structure, and memory architecture can be modified. This leads to the so-called ASIP.

**Slide 5.52**

The premier example of an ASIP is the DSP. Introduced first by Intel and AT&T in the late 1970s, the concept was ultimately popularized by Texas Instruments. Whereas performance was the initial driving force, it was soon realized that focusing on a specific application domain (such as FIR filtering, FFTs, modems, and cellular processors) ultimately leads to energy efficiency as well.
Slide 5.53
It was Gene Frantz of TI fame who was the first one to observe that the energy efficiency of DSPs doubles every 18 months as a result of both technology scaling and architectural improvements, pretty much along the same lines as performance in general-purpose processors (GPP). It is interesting to observe that the efficiency improvements started to saturate in the early 2000s, in close resemblance to the performance in GPPs.

Slide 5.54
Extrapolating from the past predicts that DSPs may require only 1 µW per MIPS by 2012. Given the technological trends, this is highly unlikely. In fact, in recent years the DSP concept has been severely challenged, and other ideas such as hardware accelerators and co-processors have firmly gained ground.
**Slide 5.55**
With the growing importance of energy efficiency, the ASIP concept has gained considerable ground. One option that is very attractive is to start from a generic processor core, and to extend the instruction set by adding dedicated hardware units to the data path—based on the application domain at hand. The advantage of this approach is that it is incremental. The compilers and software tools can be automatically updated to embrace the extended architecture. In a sense, this approach combines temporal (through the generic processor core) and spatial processing (in the dedicated concurrent hardware extensions).

**Slide 5.56**
Consider, for example, a processor for security applications, in which the DES encryption algorithm prominently features. The basic processor core is extended by adding a dedicated hardware module, which efficiently executes the permutations and the S-boxes that form the core of the DES algorithm. The extra hardware only takes 1700 additional gates. From a software perspective, these additions are translated into just 4 extra instructions. The impact on performance and energy efficiency is huge though.

**Slide 5.57**
The second and more recent example of the attractiveness of the extensible-processor approach is in the domain of video processing, particularly in realizing a decoder for the popular H.264 standard. One interesting option of that standard is the highly effective, but compute-intensive, CABAC coder, which is based on arithmetic coding. Arithmetic coding requires a large number of bit-level operations, which are not well-supported in a general-purpose core. Adding some dedicated instructions improves the CABAC performance by a factor of over 50, and the energy
loops with large iteration counts. The advantage is even better energy efficiency at the expense of area overhead. Typical examples of co-processors are correlators for wideband CDMA and FFT units for wireless OFDM (as used in WiFi implementations).

As an example, we show the computational core (all periphery omitted) of the Texas Instruments OMAP 2420 platform™, targeting mobile wireless applications. In addition to an ARM general-purpose core and a TI C55 DSP processor, the system-on-a-chip contains a number of accelerator processors for graphics, video, and security functions.

Slide 5.58
One step that goes a bit further is to spawn off complete functions as co-processors (often called accelerators). The difference with the extended ISA is that the co-processors not only perform data operations but also feature their own sequencers, and hence operate independently from the core processor. Once started, the co-processor operates until completion upon which control is passed back to the main core. Most often, these accelerators embody small efficiencies by a factor of 30, for the extra cost of just 20K extra gates.

Hardware Accelerators

Often-executed functions implemented as dedicated modules and executed as co-processors

- Opportunities: Network processing, MPEG Encode/Decode, Speech, Wireless Interfaces
- Advantage: Energy efficiency of custom implementation
- Disadvantage: Area overhead

Examples: Computational core of Texas Instruments OMAP 2420 Platform™
Slide 5.59
The effectiveness of the accelerator approach is demonstrated by this example (provided by Intel). It features an accelerator for the prevalent TCP networking stack. Given a power budget of 2 W, the TCP accelerator processor outperforms a general-purpose processor (with a 75 W budget). In fact, the gap between the two is shown to be increasing over time (as we have already observed in Slide 5.46).

Slide 5.60
One of the main negatives of the accelerator approach is the hardware overhead – Though energy-efficient, the dedicated accelerators are only used a fraction of the time, and their area efficiency is quite low. Reconfigurable spatial programming presents a means to provide simultaneously energy and area efficiencies. The idea is to create accelerators on a temporary basis by assembling a number of functional units into a dedicated computational engine. When the task at hand is finished, the structure is de-assembled, and the computational units can be reused in other functions. The underlying tenet is that reconfiguration is performed on a per-task basis. Reconfiguration events hence are rare enough that their overhead is small compared to the energy benefits of spatial computing and the area benefits of re-use.
Slide 5.61

An example again goes a long way in illustrating the concept. A number of signal-processing applications, such as speech compression, require the computation of a co-variance matrix. The sequential “C” language code implementing this function is shown on the left. However, rather than implementing it on a sequential instruction set processor, the same function can be more efficiently implemented in a spatial fashion by connecting a number of functional units such as ALUs, multipliers, memories, and most importantly address generators. The latter replace, in a sense, the indexes of the nested loops. Observe that all the units are quite generic. Creating a dedicated function requires two actions: setting the parameters of the computational modules and programming the interconnections.

Slide 5.62

There are many ways of translating this idea into silicon. This slide shows one particular incarnation. The intended application area was speech processing for cellular telephony (by now, this application only represents a small fraction of the computational needs of a mobile communicator, but in the late 1990s it was a big thing). The system-on-a-chip consists of an embedded core (ARM) and an array of computational units, address generators, and memories of various stripes. To provide even more spatial programmability for small-granularity functions, two FPGA modules are included as well. The key element linking all these modules is the reconfigurable network.
Slide 5.63
When mapping a VCELP coder on this architecture, it turns out that approximately 80% of the computational cycles can be performed on the reconfigurable fabric, whereas 20% remains on the ARM core (remember Amdahl’s law mentioned earlier in the chapter). The net effect is a gain of almost 20 over equivalent implementations on ASIP processors.

Slide 5.64
Reconfigurable accelerators are now used in a wide range of applications, including high-volume components such as CD, DVD, and MP3 players. For example, Sony has been using an architecture called the Virtual Mobile Engine (VME) in quite a number of its consumer applications. The VME architecture bears close resemblance to the reconfigurable processor structure presented in the previous slides. A quick scan of today’s system-on-a-chip architectures reveals many cases that closely fit this model.

Slide 5.65
Yet, the effectiveness of the ASIP and accelerator ideas is limited by just the same factor that hampers the multi-core idea, presented earlier in the chapter: Amdahl’s law. In other words, the potential gains in energy efficiency are bounded by the fraction of the application or algorithm that is purely sequential. Augmenting concurrency through input languages with explicit concurrency semantics, automated transformations, and/or algorithmic innovations is absolutely essential.
Remember: Amdahl’s Law Still Holds

- Effectiveness of alternative architectures (ASIP, Accelerator, Reconfigurable) determined by the amount of code spawned from GP
- Mostly effective for repetitive kernels
- 80%-20% rule typically seems to apply
- Transformations can help to improve effectiveness
- Most important: code development and algorithm selection that encourage concurrency

Although this is not a new idea at all, it has gained substantially more urgency over the past years. We recall a quote by John Tukey, one of the co-inventors of the popular FFT algorithm in the 1980s: “In the past we had to concentrate on minimizing the number of computations, now it is more important for an algorithm to be parallel and regular.”

Slide 5.66
To bring all the aforementioned concepts together, it is worth analyzing some of the integrated solutions that industry is using today for embedded applications such as multimedia and communications, which are energy- and cost-constrained. To reduce design time and time-to-market, the industry has embraced the so-called platform-based design strategy [K. Keutzer, TCAD’00]. A platform is a structured approach to programmable architectures. Built around one or more general-purpose processors and/or DSPs and a fixed interconnect structure, one can choose to add a variety of special-purpose modules and accelerators to target the component for a particular product line and to ensure that the necessary performance and energy efficiency is obtained. Re-use dramatically reduces the cost and the time for a new design.

An example is the NXP Nexperia™ platform, targeting multimedia applications. The core of the platform is an interconnect structure and two cores (either of which can be omitted): a MIPS GPP and a TriMedia™ DSP. Nexperia provides a large library of I/O modules, memory structures, and fixed and reconfigurable accelerators. Various versions of the platform are now in use in HDTVs, DVD players, portable video players, etc.
Slide 5.67
One instance of the Nexperia platform, a media processor for HDTV, is shown here. This incarnation only contains a DSP, no GPP. Most interesting are the wide range of input–output processing modules and the MPEG-2 and HDVO accelerator units. The latter is a reconfigurable co-processor for image filtering, combining flexibility, performance, and energy efficiency at a low area overhead.

Slide 5.68
Another example of the platform approach to energy-efficient programmable systems-on-a-chip is the already mentioned OMAP platform™ of Texas Instruments. The application target of this platform is the wireless-communications arena. With wireless portable devices embracing a wider range of functionality (ranging from MP3 player over TV and video playback to gaming), as well as a broad spectrum of interfaces (3G cellular, WiFi, Bluetooth, WiMAX, etc.), programmability and flexibility are essential. At the same time, the form factor limits the amount of energy available. Power is capped at 3 W, as we discussed in Chapter 1. This slide shows one instance of the OMAP platform (OMAP3430), focused in particular on providing graphics and multimedia functionality.
Summary and Perspectives

- Architectural and algorithmic optimization can lead to drastic improvements in energy efficiency
- Concurrency is an effective means to improve throughput at fixed energy or reduce energy for fixed throughput
- Energy-efficient architectures specialize the implementation of often-recurring instructions or functions

process mostly used today). Understanding the trade-offs in the energy–delay–(area, flexibility) space goes a long way in establishing such a methodology.

References

Theses:

Articles:

Slides 5.70 and 5.71

Some references . . .
References

Articles (contd.)

Chapter 6
Optimizing Power @ Design Time – Interconnect and Clocks

Slide 6.1
So far we have focused our discussion mostly on the energy efficiency of logic. However, interconnect and communication constitute a major component of the overall power budget, as we will demonstrate. They hence deserve some special attention, especially in light of the fact that the physics of interconnect scale somewhat differently than those of logic. As with logic, power optimization can again be considered at multiple levels of the design hierarchy.

Slide 6.2
The chapter commences with an analysis of the scaling behavior of interconnect wires. Some fundamental bounds on the energy dissipation of interconnect are established. One particular aspect of this chapter is that it treats on-chip communication as a generic networking problem, and hence classifies the low-energy design techniques along the lines of the standard OSI layering (just as we would do for large-scale networking). The chapter concludes with a discussion of one class of wires that need special attention: the clock distribution network.

Slide 6.3
If we consult the ITRS predictions on how interconnect will evolve in the coming decade, we observe that scaling is projected to go forward at the same pace as it does today. This leads to some staggering numbers. By 2020, we may have 14–18 (!) layers of interconnect with the lowest levels of the interconnect stack at a half pitch of only 14 nm. Clocks speeds could be at multiple tens of GHz, and the number of input and output signals may be larger than 3000. A simple analysis of what it means to switch this huge interconnect volume leads to some incredible power numbers. Even aggressive voltage scaling may not be sufficient to keep the dissipation within bounds. Hence, novel approaches on how to distribute signals on a chip are required.

Slide 6.4
In fact, the problem is already with us today. If we evaluate today’s most advanced 65 nm devices with up to 8 interconnect layers, multiple hundreds of I/O pins, and clock frequencies (at least locally) of up to 5 GHz, we see that providing connectivity between the components poses an essential limitation on the latency levels that can be achieved. It also dominates the power dissipation — at least, if we also take the clock distribution network into account. Manufacturing the multiple layers of metal (mostly Cu and Al) and dielectric material in a reliable and predictable fashion is already a challenge in itself.

Slide 6.5
To drive the point home, this slide shows the power distribution over the different resources for a number of typical classes of integrated circuits. If I/O, interconnect, and clocks are lumped together, they constitute 50% or more of the budget for each class of devices. The worst case is the FPGA,
where interconnect power takes more than 80% of the power budget [Kusse’98]. Observe that these numbers represent designs of the late 1990s and that the pendulum has swung even more in the direction of interconnect in recent years.

**Slide 6.6**

To understand why this shift is happening, it is worthwhile examining the overall scaling behavior of wires. The ideal scaling model assumes that the two dimensions of the cross-section of the wire ($W$ and $H$) are reduced with the same scaling factor $S$ between process nodes (with $S$ the same as the scaling factor of the critical dimensions of the process). How wire delay and energy dissipation per transition evolve depends upon how the wire length evolves with scaling. The length of local wires (e.g., those between gates) typically evolves the same way as the logic, whereas global wires (such as busses and clock networks) tend to track the chip dimensions (as is illustrated in the next slide). With the values of $S$ and $S_C$ (the chip scaling factor) typically being at 1.4 and 0.88, respectively, between subsequent processor nodes, we can derive the following scaling behavior:

- The delay of a local wire remains constant (in contrast to the gate delay which reduces by 1.4), whereas long wire gets 2.5 times slower!
- From an energy perspective, the picture does not seem too bad, and depends strongly on how the supply voltage scales ($U$). In the ideal model ($U = S$), things look quite good as the energy dissipation of a transition of a local and global wire reduces by 2.7 and 1.7, respectively. If the voltage is kept constant, the scaling factors are 1.4 and 0.88, respectively. Gate and wire energy exhibit approximately the same scaling behavior.
Unfortunately, the ideal model does not reflect reality. To address the wiring delay challenge, wire dimensions have not been scaled equally. For the layers at the bottom of the interconnect stack, where reducing the wire pitch is essential, wire heights have been kept almost constant between technology generations. This increases the cross-section, and hence decreases resistance – which is good for delay reduction. On the other hand, it increases capacitance (and hence energy) due to the increased contributions of the sidewalls of the wires.

Wires on the top of the stack are not scaled at all. These “fat” wires are used mostly for global interconnect. Their capacitance and energy now scales with the chip dimensions – which means that they are going up.

An important divergence between logic and interconnect is worth mentioning: though leakage has become an important component in the energy budget of logic, the same is not true (yet) in interconnect. The dielectrics used so far have been good enough to keep their leakage under control. This picture may change in the future though.

**Slide 6.7**
This slide plots a histogram showing the distribution of wire lengths in an actual microprocessor design, which contains approximately 90,000 gates. Though most of the wires are only a couple of gate pitches long, a substantial number of them are much longer, reaching lengths of up to 500 gate pitches, which is approximately the size of the die.

**Slide 6.8**
The obvious question is what technology innovations can do to address the problem. Research in novel interconnect strategies has been intense and is ongoing. In a nutshell, they can be summarized as follows:

- **Interconnect materials with lower resistance** – This only indirectly impacts energy. For the same delay, wires can be made thinner, thus reducing capacitance and in turn switching-energy. However, this avenue has led to a dead end. With copper in general use, there are no other materials in sight that can provide a next step.

- **Dielectrics with lower permittivity (so-called low-k materials)** – These directly reduce capacitance and hence energy. Advanced process technologies already use organic materials such as polyimides, which reduce the permittivity compared to the traditional SiO₂. The next step would be to move to aerogels ($\varepsilon_r \approx 1.5$). This is probably as close as we will ever get to free space. A number of companies are currently researching ways to effectively deposit “air bubbles”, by using self-assembly, for instance.
• **Shorter wire lengths** – One way to effectively reduce the wire lengths (at least those of the global wires) is to go the three-dimensional route. Stacking components vertically has been shown to have a substantial effect on energy and performance. The concept has been around for a long time, but recently has gathered a lot of renewed interest (especially in light of the perceived limits to horizontal scaling). The challenges still remain formidable – with yield and heat removal being the foremost ones.

• **Novel interconnect media** – optical interconnect strategies have long been touted as offering major performance and energy benefits. Although it is questionable if optical signaling ever will become competitive for on-chip interconnect due to the optical-electrical conversion overhead, recent advances have made off-chip optical interconnect a definitive possibility. Over the long term, carbon nanotubes and graphene offer other interesting opportunities. On an even longer time scale, we can only wish that we would be able to exploit the concept of quantum entanglement one day (tongue-in-check).

---

**Slide 6.9**

It is worthwhile to spend some time reflecting on the fundamental scaling differences between logic and wires. Under ideal scaling conditions, the Power-Delay product (i.e., the energy) of a digital gate scales as $1/S^3$. Hence gates get more effective with scaling.
Slide 6.10

On the other hand, wires tend to become less effective. For a given technology, the product of wire delay and $L^{-2}$ is a constant, assuming that the delay is dominated by the $rc$ effect. It can hence be considered to be a figure of merit. Again assuming ideal scaling rules (i.e., all dimensions scale equally with the exception of the wire length), $\tau L^{-2}$ scales as $S^2$.

$$\frac{\tau}{L^2} = \frac{rc}{HT} \propto S^2$$

In other words, the figure of merit of a wire gets worse with technology scaling, at least from a performance perspective. The only ways to change the scenario is to modify the material characteristics ($\rho e$), or the propagation mechanism (for instance, by moving from $rc$-dominated diffusion to wave propagation).

Slide 6.11

Though we have established bounds on performance, it is valuable to know if there are bounds on the energy efficiency as well. And indeed there are—and they can be obtained by using nothing less than the Shannon theorem, famous in the communication and information theory communities. The theorem relates the available capacity of a link (in bits/sec) to the bandwidth and the average signal power. Using some manipulations and assuming that a link can take an infinite time to transmit a bit, we derive that the minimum energy for transmitting a bit over a wire equals $kT\ln(2)$ (where $T$ is the absolute temperature, and $k$ the Boltzmann constant) – a remarkable result as we will see in later chapters. At room temperature, this evaluates to 4 zeptoJoules (or $10^{-21}$ J – a unit worth remembering). As a reference, sending a 1 V signal over a
1 mm intermediate-layer copper wire implemented in a 90 nm technology takes approximately 200 fl, or eight orders of magnitude more than the theoretical minimum.

**Slide 6.12**
The techniques to make interconnect more energy efficient are in many ways similar to what we do for logic. In a way, they are somewhat simpler, as they relate directly to what people have learned for a long time in the world of (large-scale) communications and networking. Hence, it hence pays off to consider carefully what designers have come up with in those areas. We should keep the following caveat in mind however: what works on the macro scale, does not always scale well to the micro scale. Not all physical parameters scale equally. For example, at shorter wire lengths and lower energy levels, the cost of signal shaping and detection becomes more important (and often even dominant). Yet, over time we have seen more and more of what once was system- or board-level architecture migrate to the die.

**Slide 6.13**
We had introduced the logical abstraction layers in Chapter 5. A similar approach can be taken for interconnect. Here, the layers are well understood, and have long been standardized as the OSI protocol stack (check [http://en.wikipedia.org/wiki/OSI_model](http://en.wikipedia.org/wiki/OSI_model) if you are not familiar with the concept). The top layers of the stack (such as the session and the presentation layers) are currently not really relevant for chip interconnects, and are more appropriate for the seamless communication between various applications over the internet. Yet, this picture may change over time when 100s to 1000s of processors get integrated on a single die. Today
though, the relevant layers are the physical, link/MAC, and network layers. We organize the rest of the chapter along those lines. Before embarking on the discussion of the various techniques, it is worth pointing out that, just as for logic, optimizations at the higher layers of the abstraction chain often have more impact. At the same time, some problems are more easily and more cheaply addressed at the physical level.

Slide 6.14
The physical layer of the interconnect stack addresses how the information to be transmitted is represented in the interconnect medium (in this case, the wire). Almost without exception, we are using voltage levels as the data representation today. Other options would be to use either currents, pulses (exploiting a wider bandwidth), or modulated sinusoids (as used in most wireless communication systems). These schemes increase the complexity of the transmitter and/or receiver, and hence have not been very attractive for integrated circuits. Yet, this may change in the future, as we discuss briefly at the end of the chapter.

Slide 6.15
The majority of the wires on a chip can be considered either as being purely capacitive (for very short connections), or as distributed $rc$-lines. With the availability of thick copper lines at the top of the chip, on-chip transmission lines have become an option as well. They form an interesting option for the distribution of signals over longer distances.

Given their prominence, we focus most of our attention on the $rc$ lines in this chapter. It is well-known that the delay of the wire increases quadratically with
its length, whereas the energy dissipation rises linearly. The common technique to get around the delay concern is to insert repeaters at carefully selected intervals, which makes it possible to make the delay proportional to the length of the wire. The optimal insertion rate (from a performance perspective) depends upon the intrinsic delays of both the driver and the interconnect material.

The introduction of repeaters adds active components to an otherwise passive structure, and hence adds extra energy dissipation.

---

**Slide 6.16**

The cost of optimal performance is very high (this should be of no surprise by now). Consider, for instance, a 1 cm copper line implemented in a 90 nm technology. The energy cost of the receiver is six times higher than what it takes to just charge the wire with a single driver. Again, just backing off a bit from the absolute minimum delay goes a long way in making the design more energy-efficient.

---

**Slide 6.17**

As always, the trade-off opportunities are best captured by the energy–delay curves. Doubling the allowable delay reduces the required energy by a factor of 5.5! Even just backing off 10% already buys a 30% energy reduction.
Slide 6.18

It is worth spending some time contemplating on how this pareto-optimal E-D curve was obtained. The design parameters involved include the supply (signal) voltage, the number of stages, and the transistor sizes (in the buffer/repeaters). From the results of the multi-dimensional optimization, it can be seen that the supply voltage has the biggest impact, followed by insertion rate of the repeaters. Observe that the width of the wire only has a secondary impact on the wire delay. Once the wire is wide enough to make the contribution of the fringing capacitance or sidewall capacitance ignorable, further increases in the wire width do nothing more than raising the energy dissipation, as is illustrated by the equations below.

\[ \begin{align*}
    c_w &= w \cdot c_{pp} + c_f \\
    r_w &= r_{sq}/w \\
    \tau_w &= c_{pp}r_{sq} + c_f r_{sq}/w
\end{align*} \]

Slide 6.19

With reduction of the supply voltage (or more precisely, the signal swing) proven to be the most effective technique to save energy, some contemplation on how to accomplish this effectively is at hand. As we have observed earlier, sending a signal along a wire is a communication problem, and it is worth considering as such. A communication link consists of a transmitter (TX), a communication medium, and a receiver (RX). The generic configuration in CMOS is to have a driver (inverter) as TX, a stretch of aluminum or copper wire in between, and another inverter as a receiver.
This changes once we reduce the signal swing. The TX acts as a driver as well as a level down-converter, whereas the RX performs the up-conversion. Though the energy savings are either linear or quadratic, depending upon the operational voltage of the TX, reducing the swing comes with an overhead in delay (maybe) and complexity (for sure). In addition, it reduces the noise margins and makes the design more susceptible to interference, noise, and variations. Yet, as we have learned from the communications community, the benefits of properly conditioning the signal can be quite substantial.

Slide 6.20
In Chapter 4, we had already touched on the topic of level conversion and multiple supply voltages (Slides 4.32, 4.33 and 4.34). It was concluded that down-conversion is relatively easy if multiple supply voltages are available. The challenge is in the up-conversion. In the logic domain, where the overhead penalty easily offsets the energy gains, we concluded that the level-conversion is best confined to the boundaries of the combinational logics (i.e., the flip-flops), where the presence of a clock helps to time when to perform the energy-hungry amplification. The availability of positive feedback in most of the latches/registers is another big plus.

Yet, synchronous or clocked conversion is not always an option in the interconnect space, and asynchronous techniques are worth examining. In this slide, a conventional reduced-swing interconnect scheme is presented. To reduce the signal swing at the transmit site, we simply use an inverter with a reduced supply voltage. The receiver resembles a differential cascade voltage switch logic (DCVSL) gate [Rabaey03], which consists of complementary pull-down networks and a cross-coupled PMOS load. The only difference is that the input signals are at a reduced voltage level, and that a low-swing inverter is needed to generate the complementary signal. The disadvantage of this approach is that it effectively needs two supply voltages.

Slide 6.21
One interesting way to create a voltage drop is to exploit implicit voltage references such as threshold voltages. Consider, for instance, the circuit presented in this slide. By swapping the NMOS and PMOS transistors in the drive, the logic levels on the wire are now set to $V_{THp}$ and $V_{DD} - V_{THn}$. For a supply voltage of 1 V and threshold voltages for NMOS and PMOS transistors around 0.35 V, this translates into a signal swing of only 0.3 V!

The receiver consists of dual cross-coupled pairs. The transistors N2 and P2 ensure full logic swing at the outputs, whereas N3 and P3 isolate the full-swing output nodes from the low-swing interconnect wires. A number of alternative level-conversion circuits, and a comparison of their effectiveness can be found in [Zhang’00]. From this, we learn that to be effective the
TX–RX overhead should be no more than 10% of the overall energy budget of the communication link.

Slide 6.22
One of the concerns of the reduced-swing circuits is their increased sensitivity to interference and supply noise. Using a differential scheme not only offers a major increase in common-mode rejection, but also helps to reduce the influence of interference by 6 dB. Signaling schemes with reliable swing levels of as low as 200 mV have been reported and used. At the same time, differential interconnect networks come with a substantial overhead, as the overall wire capacitance is doubled – translating directly into extra energy dissipation. In addition, the differential detection scheme at the receiver consumes continuous power, and should be turned off when not in use. This most often (but not necessarily always) means that a clocked synchronous approach is required. Differential techniques are most effective when the wiring capacitance is huge, and the benefits of the extra small swing outweigh the overhead of the doubled capacitance and the extra clocking.
Lower Bound on Signal Swing?

- Reduction of signal swing translates into higher power dissipation in receiver—trade-off between wire and receiver energy dissipation
- Reduced SNR impacts reliability—current on-chip interconnect strategies require Bit Error Rate (BER) of zero (in contrast to communication and network links)
  - Noise sources: power supply noise, crosstalk
- Swings as low as 200 mV have been reported [Ref: Burd’00], 100 mV definitely possible
- Further reduction requires crosstalk suppression

Slide 6.23
At this point, it is worth wondering if there exists a lower bound on the signal swing that can be used in practice. A number of issues should be considered:

- Reducing the swing negatively impacts the delay, as it substantially increases the time it takes to reconstruct the signal level at the receiver end. In general, we may assume that the receiver delay is proportional to the swing at its input. This again leads to a trade-off. In general, the longer the wire the more the reduced swing makes sense.
  - The smaller the signal, the larger the influence of parasitic effects such as noise, crosstalk, and receiver offset (if differential schemes are used). All of these may cause the receiver to make erroneous decisions. Besides the power supply noise of sender and receiver, the primary noise source in interconnect networks is the capacitive (and today even inductive) coupling between neighboring wires. This is especially a problem in busses, in which wires may run alongside each other for long distances, and crosstalk becomes substantial. This problem can be reduced by crosstalk-repressing techniques such as proper shielding—which comes at the expense of area and wire folding.

So far, signaling swings that have been reported on busses hover around 200 mV. There is however no compelling reason to assume that this number can no further be reduced, and 100 mV swings have been considered in a number of designs.

Caveat: The usage of reduced signal swings on-chip definitely is incompatible with the standard design methodologies and flows, and hence falls into the realm of custom design. This means that the designer is fully responsible for the establishment of verification and test strategies—clearly not for the fainthearted . . . Fortunately, a number of companies have brought modular intellectual-property (IP) solutions for energy-efficient on-chip communication on the market in recent years, thus hiding the complexity from the SoC designer.

Slide 6.24
A wide range of other energy-reducing on-chip data communication schemes have been published—few of which have made it onto industrial designs, though. A couple of those ideas are too compelling to omit from this text. The first one is based on the adiabatic charging approach we briefly touched on in Chapter 3. If delay is not of primary importance, we can extend the energy–delay space by using alternative charging techniques. Yet, the implementation of a truly adiabatic circuit requires the implementation of an energy-recovering clock generator, which typically requires a resonating network including inductors [L. Svensson, CRC’05]. The latter are expensive and low-quality when implemented on-chip. Off-chip inductors, on the other hand, increase the system cost.
In this slide, a quasi-adiabatic driver for large capacitive busses is presented. A stepwise approximation of a ramp is produced by connecting the output in sequence to a number of evenly distributed voltages, starting from the bottom. From each reference, it receives a charge $C_L V/N$ before eventually being connected to the supply. To discharge the capacitance, the reverse sequence is followed. For each cycle, a charge equal to $C_L V/N$ is drawn from the supply, a reduction by a factor of $N$ over the single-step charging. The total energy is reduced by the same factor.

The $N-1$ intermediate voltage references are realized using a capacitor tank $C_{T1}$ (where $C_{T1} >> C_L$). During each charge-and-discharge cycle, each capacitor $C_{T1}$ provides and receives the same amount of charge, so the tank capacitor voltages are self-sustaining. Even more, it can be shown that during the start-up, the tank voltages automatically converge to an equal distribution.

In essence, this driver is not truly adiabatic. It rather belongs to the class of the “charge-redistribution” circuits: each cycle, a charge packet is injected from the supply, which then gradually makes its way from level to level during subsequent cycles, and is finally dumped into ground after $N$ cycles. The following slide shows another circuit of the same class.

**Slide 6.25**

“Charge recycling” is another idea that is very intriguing, but has rarely been used. In a traditional CMOS scheme, charge is used only a single time: it is transferred from the supply to the load capacitor in a first phase, and dumped to the ground in a second. From an energy perspective, it would be great if we could use charge a couple of times before dumping it. This by necessity requires the use of multiple voltage levels. A simplified example of a chargerecycling bus with two levels is shown in this slide. Each bit $i$ is present in differential form...
During the precharge phase, the two differential lines for each bit are equalized by closing the switches P. During evaluation, one of the lines is connected to a line of a “higher-order” bit (representing a 1), whereas the other is equalized with a “lower-order” bit (representing a 0) using the switches E. This produces a differential voltage at each line pair, the polarity of which depending upon the logic value to be transmitted. Differential amplifiers at the end of the bus (one for each pair) reproduce the full swing signals.

Assuming that the capacitances of all lines are equal, we can see that the precharge voltage levels divide equally between $V_{DD}$ and GND. The signal swing on each bus pair equals $V_{DD}/N$. The principle is quite similar to that of the quasi-adiabatic driver of the previous slide—a charge packet is injected from the supply, which sequentially drives every bit of the bus in descending order, until eventually being dumped on the ground.

The challenge resides in adequately detecting the various output levels in the presence of process variations and noise. Yet, the idea has enough potential that it is bound to be useful in a number of special cases.

**Slide 6.26**

At the end of this discussion on physical-layer signaling, it is worth pointing out some other signaling strategies that may become attractive. In this slide, we show only one of them. Rather than connecting resistively into the interconnect network, drivers could also couple capacitively. The net effect is that the swing on the interconnect wire is reduced automatically without needing any extra supplies. In addition, driver sizes can be reduced, and signal transitions are sharper. The approach comes with a lot of challenges as well (one still needs a level-restoring receiver, for instance), but is definitely worth keeping an eye on.

**Slide 6.27**

So far, we have concentrated on the data representations of our signaling protocol, and have ignored timing. Yet, the interconnect network plays an important role in the overall timing strategy of a complex system-on-a-chip (SoC). To clarify this statement, let us consider the following simple observation: It takes an electronic signal moving at its fastest possible speed (assuming transmission-line conditions) approximately 66 ps to move from one side to the other of a 1 cm chip. When $rc$-effects dominate, the reality is a lot worse, as shown in Slide 6.16, where the minimum delay was determined to be 500 ps. This means that for clock speeds faster than 2 GHz, it takes more than one clock cycle for a signal to propagate across the chip! The situation is even worse, when the interconnect wire is loaded with a large distributed fan-out – as is always the case with busses, for instance.

There are a number of ways to deal with this. One commonly used option is to pipeline the wire by inserting a number of clocked buffer elements. This happens quite naturally in the
network-on-a-chip (NoC) paradigm, which we will discuss shortly. Yet, all this complicates the overall timing of the chip, and intrinsically links the timing of global interconnect and localized computation. This hampers the introduction of a number of power reduction techniques we have discussed earlier (such as multiple supply voltages and timing relaxation), or to be discussed in coming chapters (such as dynamic voltage and frequency scaling).

Hence it makes sense to decouple global interconnect and local compute timing through the use of asynchronous signaling. Along these lines of thinking, one methodology called GALS (Globally Asynchronous Locally Synchronous) has attracted a following in recent years [Chapiro’84]. The idea is to use a synchronous approach for the local modules (called synchronous islands), while communication between them is performed asynchronously. This approach dramatically relaxes the clock distribution and interconnect timing requirements, and enables various power-saving techniques for the processor modules.

The generation and termination of the control signals is most effectively performed by a standardized wrapper around the computational modules, which serves as the boundary between the synchronous and asynchronous domains. One of the very first designs that followed that concept is presented in [Zhang’00].
The Data Link/Media Access Layer

- Reliable transmission over physical link and sharing interconnect medium between multiple sources and destinations (MAC)
  - Bundling, serialization, packetizing
  - Error detection and correction
  - Coding
  - Multiple-access schemes

If the link connects to multiple sources and destinations, the media access control (MAC) protocol ensures that all sources can share the media in a fair and reliable fashion. Bus arbitration is a great example of a MAC protocol used extensively in SoCs.

Most designers still consider interconnect purely as a set of wires. Yet, thinking about them as a communication network opens the door for a broad range of opportunities, the scope of which will only increase with further scaling. As a starter, the link layer offers a great number of means of introducing energy-saving techniques for global interconnects. For example, adding error-correcting coding allows for a more aggressive scaling of the voltage levels used in a bus.

Coding

Adding redundancy to communication link (extra bits) to:
- Reduce transitions (activity encoding)
- Reduce energy/bit (error-correcting coding)

Slide 6.29
Whereas the physical layer deals with the various aspects of how to represent data on the interconnect medium, the function of the link layer is to ensure that data is reliably transmitted in the appropriate formats between the source and the destination. For example, in the wired and wireless networking world, packets of data are extended with some extra error-control bits, which help the destination to determine if the packet was not corrupted during its travel.

Slide 6.30
Coding is a powerful technique, which is extensively used in most wired and wireless communication systems. So far, its overhead has been too high to be useful in on-chip interconnects. With the growing complexity of integrated circuits, this may rapidly be changing. A number of coding strategies can be considered:
- Channel-coding techniques, which modify the
data to be transmitted so that it better deals with imperfections of the channel.
- Error-correcting codes, which add redundancy to the data so that eventual transmission errors can be detected and/or corrected.
• Source codes, which reduce the communication overhead by compressing the data.

As the last one is application-dependent, we focus on the former two in this section. While coding may yield with a substantial energy benefit, it also comes with an overhead:

• Both channel and error-correcting codes require a redundancy in the representation, which most often translates into extra bits.
• Implementation of the coding requires an encoder (at the TX side) and a decoder (at the RX side).

As a result, coding is only beneficial today for interconnect wires with substantive capacitive loads.

<table>
<thead>
<tr>
<th>Activity Reduction Through Coding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example: Bus-Invert Coding</td>
</tr>
<tr>
<td><img src="Image" alt="Diagram" /></td>
</tr>
</tbody>
</table>

- Data word $D$ inverted if Hamming distance from previous word is larger than $N/2$.

<table>
<thead>
<tr>
<th>$D$</th>
<th># $T$</th>
<th>$D_{enc}$</th>
<th>$p$</th>
<th># $T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>00101010</td>
<td>0</td>
<td>00101010</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>00111011</td>
<td>2</td>
<td>00111011</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>11010100</td>
<td>7</td>
<td>00101011</td>
<td>1</td>
<td>1+1</td>
</tr>
<tr>
<td>00001101</td>
<td>5</td>
<td>00001101</td>
<td>0</td>
<td>3+1</td>
</tr>
<tr>
<td>01110110</td>
<td>6</td>
<td>10001001</td>
<td>1</td>
<td>2+1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

[Ref: M. Stan, TVLSI’95]

Slide 6.31

As stated in the beginning of the chapter, reducing activity on the interconnect network is an effective way of reducing the energy dissipation. Coding can be an effective means of doing exactly that. To demonstrate the concept, we introduce a simple coding scheme called “bus-invert coding” (BIC). If the number of bit transitions between two consecutive data words is high, it is advantageous to invert the second word, as is shown in the example. The BIC encoder computes the Hamming distance between the previous data transmission $D_{enc(t-1)}$ and the current $D(t)$. If the Hamming distance is smaller than $N/2$, we just transmit $D(t)$ and set the extra code bit $p$ to 0. In the reverse case, $D_{enc(t)} = D(t)$, and $p$ is set to 1.
Bus-Invert Coding

Gain:
- 25% (at best — for random data)
- Overhead:
  - Extra wire (and activity)
  - Encoder, decoder
  - Not effective for correlated data

[Ref: M. Stan, TVLSI'95]

Under the best possible conditions, the bus-invert code may result in a 25% power reduction. This occurs when there is very little correlation between subsequent data words (in other words, data is pretty much random, and transitions are plentiful). When the data exhibits a lot of correlations, other schemes can be more effective. Also, the code is less effective for larger values of $N$ (> 16).

Other Transition Coding Schemes

- Advanced bus-invert coding (e.g. partition bus into sub-components)
- Coding for address busses (which often display sequentiality)
  [e.g. B. Benini, DATE’98]
- Full-fledged channel coding, borrowed from communication links
  [e.g. B. Ramprasad, TVLSI’99]

<table>
<thead>
<tr>
<th>bit #1</th>
<th>bit k-1</th>
<th>Delay factor $g$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$+$</td>
<td>$+$</td>
<td>$1 + r$</td>
</tr>
<tr>
<td>$+$</td>
<td>$-$</td>
<td>$1 + 2r$</td>
</tr>
<tr>
<td>$-$</td>
<td>$+$</td>
<td>$1 + 2r$</td>
</tr>
<tr>
<td>$+$</td>
<td>$+$</td>
<td>$1 + 4r$</td>
</tr>
</tbody>
</table>

Codiing to reduce impact of Miller capacitance between neighboring wires
[Ref: Solfiedis, ASPDAC’01]

Maximum capacitance transition—can be avoided by coding

Slide 6.32

One possible realization of the BIC scheme is shown in this slide. The Encode module computes the cardinality of the number of bit transitions between $D_{enc}(t-1)$ and $D(t)$. If the result is larger than $N/2$, the input word is inverted by a bank of XORs, otherwise it is passed along unchanged. Decoding requires no more than another bank of XORs and a register. Observe that the BIC scheme comes with the overhead of one extra bit ($p$).

Slide 6.33

The idea of transition coding has gained quite some traction since the BIC scheme was introduced. For the sake of brevity, we just provide a number of references. One class of schemes further optimizes the BIC scheme by, for instance, partitioning the bus if the word length $N$ gets too large. More generic channel-coding schemes have been considered as well.

In case the data exhibits a lot of temporal correlation, a totally different class of codes comes into play. For example, memory is often accessed sequentially. Using address representations that exploit the correlations, such as Gray coding, can help to reduce the number of transitions substantially.

Most transition-coding techniques focus on temporal effects. Yet spatial artifacts should not be ignored either. As we have observed before, bus wires tend to run alongside one
another for long distances. The intra-wire capacitance can hence be as or even more important than the capacitance to ground. Under unfavorable conditions, as the table on the slide indicates, a wire can experience a capacitance that is many times larger than the most favorable case. Codes can be engineered to minimize the occurrence of “aggressions” between neighboring bits. Their overhead is manageable as well, as was established in [Sotiriades'01].

**Slide 6.34**

Error-correction codes present another interesting opportunity. Under the motto of “better-than-worst-case” design – a concept that receives a lot more attention in Chapter 10 – it often serves well to purposely violate a design constraint, such as the minimum supply voltage or clock period. If this only results in errors on an occasional basis, the energy savings can be substantial. Again, this requires that the overhead of encoding/decoding and transmitting a couple of extra bits is lower than the projected savings. This concept, which is instrumental to the operation of virtually all wireless communications, has been used extensively in various memory products such as DRAM and Flash over the last decades.

It is not too hard to envision that error-correcting coding (ECC) will play a major role in on-chip interconnect in the very near future as well. Our colleagues in the information theory community have done a wonderful job in coming up with a very broad range of codes, ranging from the simple and fast to the very complex and effective. A code is classified by the number of initial data bits, the number of parity bits, and the number of errors it can detect and/or correct. At the current time instance on the technology roadmap, only the simplest codes, such as Hamming, truly make sense. Most other schemes come with too much latency to be useful. An example of a (4,3,1) Hamming code is shown in the slide.

As we had established earlier, scaling causes the cost of communication and computation to increase and decrease, respectively. This will make coding techniques more and more attractive as time progresses.

**Slide 6.35**

Another aspect of the link layer is the management of media access control (MAC) – in case multiple senders and receivers share the media. A bus is a perfect example of such a shared medium. To avoid collisions between data streams from different sources, time-division multiplexing (TDM) is used. The overhead for sharing the media (in addition to the increased capacitance) is the scheduling of the traffic.
Media Access

- Sharing of physical media over multiple data streams increases capacitance and activity (see Chapter 5), but reduces area.
- Many multi-access schemes known from communications
  - Time domain: Time-Division Multiple Access (TDMA)
  - Frequency domain: narrow band, code division multiplexing
- Buses based on Arbitration-based TDMA most common in today's ICs

With an eye on the future, it is again worth pondering the possibility of borrowing ideas from the wireless- and optical-communication communities that utilize the diversity offered by the frequency domain. Today, all IC communications are situated in the baseband (i.e., from 0 Hz to 100s of MHz), similar to where optical communications used to be for a long time (before the introduction of wave-division multiplexing or WDM). Modulation of signal streams to a number of higher-frequency channels allow for the same wire to be used simultaneously by a number of streams. The overhead of modulation/demodulation is quite substantial – yet, it is plausible that frequency-division multiplexing (FDM) or code-division multiplexing (CDM) techniques may be used in the foreseeable future for high-capacity energy-efficient communication backplanes between chips, where the link capacitance is large. A great example of such a strategy is given in [Chang'08].

In light of this, TDM is the only realistic on-chip media-access protocol today. Even here, many different options exist, especially regarding the granting of access to the channel. The simplest option is for a source to just start transmitting when data is available. The chance for a collision with other sources is extremely high though, and the overhead of retransmission attempts dominates the energy and latency budgets. The other extreme is to assign each stream its own time slot, granted in advance. This works well for streams that need guaranteed throughput, but may leave some channels heavily under-utilized. Bus arbitration is the most common scheme: a source with data available requests the channel, and starts transmitting when access is granted. The overhead of this scheme is in the execution of the “arbitration” protocol.
with a well-defined protocol (which is orthogonal to the actual interconnect implementation). Based on throughput and latency requirements, links are either granted a number of fixed slots, or compete for the others using a round-robin arbitration protocol.

**Slide 6.37**
The network is the next layer in the OSI stack. With the number of independent processing modules on a chip growing at a fast rate, this layer – which was missing on chips until very recently – is rapidly gaining attention. The networking and parallel computing communities have provided us with an overwhelming array of options, a large number of which is not really relevant to the “network-on-a-chip” (NoC) concept. The choices can be classified into two major groups: (1) the network topology; and (2) the time of configuration.
than make sense. In addition, the inserted switches/routers act as repeaters and help to control the interconnect delay.

Slide 6.38
One may wonder if a network-on-a-chip approach truly makes sense. We are convinced that it is an absolute necessity. With a large number of modules, point-to-point connections rapidly become unwieldy, and occupy a disproportionate amount of area. A shared time-multiplexed resource, such as a bus, saturates, if the number of connected components becomes too high. Hence, breaking up the connection into multiple segments does more

Slide 6.39
The architectural exploration of NoCs follows the lines of everything else in this text: it involves a trade-off between delay and energy, and in addition, flexibility and area. From an energy perspective, common themes re-emerge:

- Preserving locality – The advantage of a partitioned network is that communications between components that are near to each other – and which make up a lot of the overall traffic – are more energy-efficient.

- Building hierarchy – This creates a separation between local and global communications. Networks that work well for one do not work well for the other.

- Optimal re-use of resources – depending upon the energy and delay constraints there exists an optimum amount of concurrency and/or multiplexing that minimizes the area.

Sound familiar?
Networking Topologies

- Homogeneous
  - Crossbar, Butterfly, Torus, Mesh, Tree, ...
- Heterogeneous
  - Hierarchy

Slide 6.40
We could spend a lot of valuable book “real estate” on an overview of all the possible interconnect topologies that are known to mankind, but much of it would be wasted. For instance, a number of structures that excel in high-performance parallel computers do not map well on the two-dimensional surface of an integrated circuit; an example of such is the hypercube. We therefore restrict our discussion to the topologies that are commonly used on-chip.

- The crossbar presents a latency-efficient way of connecting \( n \) sources to \( m \) destinations. However, it is expensive from both an area and energy perspective.
- The mesh is the most popular NoC architecture today. The FPGA was the first chip family to adopt this topology. The advantage of the mesh is that it uses only nearest-neighbor connections, thus preserving locality when necessary. For long-distance connections, the multi-hop nature of the mesh leads to large latencies. To combat this effect, FPGAs overlay meshes with different granularity.
- A binary tree network realizes a \( \log_2(N) \) latency network (where \( N \) is the number of elements in the network) with relatively low wiring and capacitance costs. Other versions of this network vary the cardinality of the tree. In a fat tree, the cardinality is gradually increased for the higher levels. Trees offer an interesting counterpart to meshes as they are more effective in establishing long-distance connectivity.

Given that each network topology has its strengths and weaknesses, it comes as no surprise that many of the deployed NoCs pragmatically combine a number of schemes in a hierarchical fashion, presenting one solution for local wiring supplemented by another for the global connections. In addition, point-to-point connections are used whenever needed.

Slide 6.41
Navigating the myriad choices in an educated fashion once again requires an exploration environment that allows a study of the trade-off between the relevant metrics over the parameter set.

This is illustrated by the example in this slide, which compares the mesh and binary tree networks in the energy–delay space. As can be expected, the mesh network is the most effective solution for short connections, whereas the tree is the preferred choice for the longer ones. A solution that combines the two networks leads to a network with a merged pareto-optimal curve.

This combination is not entirely effective. If the goal is to make the latency between any two modules approximately uniform, straightforwardly combining the tree and the mesh topologies
Network Topology Exploration

- Short connections in tree are redundant
- Inverse clustering complements mesh

[Ref: V. George, Springer'01]

Circuit-Switched Versus Packet-Based

- **On-chip reality**: Wires (bandwidth) are relatively cheap; buffering and routing expensive
- Packet-switched approach versatile
  - Preferred approach in large networks
  - But... routers come with large overhead
  - Case study Intel: 18% of power in link, 82% in router
- Circuit-switched approach attractive for high-data-rate quasi-static links
- Hierarchical combination often preferred choice

Slide 6.42

The other important dimension in the network exploration is the choice of the routing strategy, and the time at which it is established. The *static routing* is the simplest option. In this case, network routes are set up at design time. This is, for instance, the case in FPGAs, where the switches in the interconnect network are set at design time. A simple modification of this is to enable reconfiguration. This allows for a route to be set up for a time (for instance, for the duration of a computational task), and then be ripped up and rerouted. This approach resembles the *circuit-switched* approach of the traditional telephone networks. The advantage of both static- and circuit-switched routing is that the overhead is reasonable and solely attributable to the additional switches in the routing paths. (Note: for performance reasons, these switches are often made quite large and add a sizable amount of capacitance).

Packet-switched networks present a more flexible solution, where routes are chosen on a per-packet basis (as is done in some of the Internet routers). The overhead for this is large, as each router element has to support buffering as well as dynamic route selection. A major improvement is
inspired by the realization that most data communications consist of a train of sequential packets. Under those conditions, the routing decision can be made once (for the first packet), with the other packets in the train just following. This approach is called “flit-routing” [Ref: Dally’01].

Experiments have shown that flit routing in NoCs is still quite expensive and that the energy cost of the dynamic routing is multiple times higher than the cost of the link. This realization is quite essential: the energy cost of transmitting a bit over a wire on a chip is still quite reasonable compared to the implementation cost of routers and buffers. As long as this is the case, purely dynamic network strategies may not be that attractive. Heterogeneous topologies, such as combining busses for short connections with mesh- or tree-based circuit- or packet-switched networks for the long-distance ones, most probably offer a better solution.

**Example: The Pleiades Network-on-a-Chip**

- Configurable platform for low-energy communication and signal-processing applications (see Chapter 5)
- Allows for dynamic task-level reconfiguration of process networks
- Energy-efficient flexible network essential to the concept

[Ref: H. Zhang, JSSC’00]

**Slide 6.43**

One of the earlier NoCs, with particular focus on energy efficiency, is found in the Pleiades reconfigurable architecture, already discussed in Chapter 5. In this platform, modules are wired together to form a dedicated computational engine on a per-task basis. Once the task is completed, the routes are ripped up and new ones established, reusing interconnect and compute modules for different functions. This approach hence falls into the “circuit-switched” class of networks.
traffic requirements at this level allow for the use of a simpler and more restrictive switchbox. This topology, which was produced by an automated exploration tool, reduces the interconnect energy by a factor of seven over a straightforward crossbar, and is also substantially more area-efficient.

Slide 6.45

Traditional OSI network stacks support even more abstraction layers, such as transport, session, presentation, and application. Each of these layers serves to further abstract away the intricacies of setting up, maintaining, and removing a reliable link between two nodes. It will take some time before these abstractions truly make sense for a NoC. Yet, some elements are already present in today’s on-chip networks. For example, the concept of a session is clearly present in the circuit-switched Pleiades network.

Although the impact on energy may not immediately be obvious, it is the existence of these higher abstraction levels that allow for a consistent, scalable, and manageable realization of energy-aware networks.
What about Clock Distribution?

- Clock easily the most energy-consuming signal of a chip
  - Largest length
  - Largest fan-out
  - Most activity (α = 1)

- Skew control adding major overhead
  - Intermediate clock repeaters
  - De-skewing elements

- Opportunities
  - Reduced swing
  - Alternative clock distribution schemes
  - Avoiding a global clock altogether

Slide 6.46
At the end of this chapter on energy-efficient interconnect, it is worth spending some time on the interconnection that consumes the most: the clock. The clock network and its fan-out have been shown to consume as much as 50% of the total power budget in some high-performance processors.

When performance was the only thing that mattered, the designers of clock distribution networks spent the majority of their time on “skew management”, and power dissipation was an afterthought. This explains, for instance, the usage of power-hungry clock meshes [Rabaey’03, Chapter 10]. A lot has changed since then. The clock distribution networks of today are complex hierarchical and heterogeneous networks, combining trees and meshes. In addition, clock gating is used to disable inactive parts of the network (more about this in Chapter 8). Mayhaps, it is better to avoid using a clock altogether (Chapter 13). Detailed discussions on the design of clock networks and the philosophy of time synchronization are unfortunately out of the scope of this text.

Some interesting approaches at the physical level, however, are worth mentioning. Similar to what we had discussed with respect to data communications, it may be worthwhile to consider alternative clock signaling schemes, such as reduced signal swings.

Slide 6.47
Reducing the clock swing is an attractive proposition. With the clock being the largest switching capacitance on the chip, reducing its swing translates directly into major energy savings. This is why a number of ideas on how to do this effectively popped up right away when power became an issue in the mid 1990s. An example of a “half-swing” clock generation circuit is shown in this slide. The clock generator uses charge redistribution over two equal capacitors to
generate the mid voltage. The clock is distributed in two phases for driving the NMOS and PMOS transistors, respectively, in the connecting flip-flops.

The reduction in clock swing also limits the driving voltage at the fan-out flip-flops, which translates into an increase in clock-to-output delay. As this directly impacts the timing budget, reduced-swing clock distribution comes with a performance hit. Another challenge with the reduced clock swing is the implementation of the repeaters and buffers that are part of a typical clock distribution network. These need to operate from the reduced voltage as well. Because of these and other concerns, reduced-swing clock networks have been rarely used in complex ICs.

### Slide 6.48
Another option is to consider alternative clock-distribution approaches. Over the years, researchers have explored a broad range of ideas on how to accurately synchronize a large number of distributed components on a chip. Ideas have ranged from coupled oscillator networks to generating standing waves in distributed resonant circuit elements (e.g., [Sathe’07]). Others have considered the idea of optical clock distribution. Given the importance of this topic, research is ongoing and a workable alternative to the paradigm of a centralized clock distributed with a “skewfree” network may emerge.

Given the limited space, we have chosen to present one single option in this slide (this does not presume a judgment on any of the other schemes). It is based on the assumption that virtually lossfree transmission lines can be implemented in the thick copper metal layers, which are available in all advanced CMOS processes. The transmission lines without a doubt present the fastest interconnect medium. Assume now that a pulse is transmitted over a folded transmission line (the contour of which is not important at all). At any point along the trajectory, the average between the early and late arrivals of the pulse is a constant—hence skewfree. By strategically positioning a number of “clock-extracting circuits” (which could be an analog multiplier) over the chip, a skewfree clock distribution network can be envisioned. The power dissipation of this network is very low as well. Though this scheme comes with some caveats, it is this form of disruptive technology that the energy-minded designer has to keep an eye on.
Summary

- Interconnect important component of overall power dissipation
- Structured approach with exploration at different abstraction layers most effective
- Lot to be learned from communications and networking community – yet, techniques must be applied judiciously
  - Cost relationship between active and passive components different
- Some exciting possibilities for the future: 3D integration, novel interconnect materials, optical or wireless I/O

References

Books and Book Chapters

Articles

Slides 6.50–6.52

Some references...


V. Pradhan and M. Banu, "GHz serial passive clock distribution in VLSI using bidirectional signaling," Proceedings ICCG 06.


