Low Power Methodology Manual

For System-on-Chip Design

Michael Keating • David Flynn •
Robert Aitken • Alan Gibbons • Kaijian Shi

Low Power Methodology Manual

For System-on-Chip Design

Springer

Michael Keating
Synopsys, Inc.
Palo Alto, CA
USA

David Flynn
ARM Limited
Cambridge
United Kingdom

Robert Aitken
ARM, Inc.
Almaden, CA
USA

Alan Gibbons
Synopsys, Inc.
Northampton
United Kingdom

Kaijian Shi
Synopsys, Inc.
Dallas, TX
USA

Library of Congress Control Number: 2007928355

ISBN 978-0-387-71818-7 e-ISBN 978-0-387-71819-4

Printed on acid-free paper.

9 8 7 6 5 4 3 2
Corrected at second printing, 2008

springer.com

TRADEMARKS

DISCLAIMER

Table of Contents

Preface

The Low Power Methodology Manual is the outcome of a decade-long collaboration between ARM and Synopsys commercially and the two of us personally. In 1997 ARM and Synopsys worked together to develop a synthesizable ARM7 core. Dave was the ARM lead on the project; Mike's team executed the Synopsys side of the project. This led to a similar project on the ARM9.

Shortly after these projects, the two of us embarked on a series of technology demonstration projects. We both felt that we needed to use our products as our customers do in order to understand how to make these products better. So we developed a test chip that combined ARM and Synopsys IP and took it through to silicon. We did the RTL design and verification personally, and borrowed resources to do the implementation. The experience was incredibly illuminating, and we hope it contributed to improving the IP and tools from both companies.

We quickly realized that low power was one of the key concerns of our customers, and SoC designers in general. So we followed our initial project with several low power technology demonstration projects. The final project was the SALT (Synopsys ARM Low-power Technology demonstrator) project, for which we received working silicon late last year. These projects explored clock gating, multi-voltage, dynamic voltage scaling, and power gating. In all these projects we found that there is no substitute for direct first-hand experience doing low-power IP-based designs. We learned, in the most concrete way possible, exactly what our customers go through on an SoC design.

For years we have been talking about writing a book on low power design. With our experience on the SALT project, our work with customers on low power designs, and

our participation in developing the UPF low-power standard, we feel that we are finally in a position to publish our insights and perspectives.

In doing so, we have enlisted the aid of our co-authors. The two of us are primarily front-end engineers, with a background in system architecture and RTL design. Kaijian and Rob bring a great depth of technical expertise in the physical and circuit design aspects of low power. Alan has developed low power flows for the ARM processors and did the implementation of SALT. As a result, he brings a unique perspective on the implementation issues in low power design.

We cannot overstate the contribution of our co-authors. Without their insights and expertise - as well as the material they contributed directly - this book could not have been written.

Like all our joint projects, this book was partly a formal joint project of the two companies and partly (perhaps mostly) driven by the personal commitment of the authors, aided and abetted by many others. We got considerable help from many people for whom this was not part of their job description. These kind souls took time out of their busy schedules, including evenings and weekends, to help us at every step of our journey, from the first joint chip development to the completion of this book. They helped in the architecture, design and tape out of test chips, the building and debugging of boards, and the review and editing of the final manuscript.

It is impossible to list them all, but we list some of the many who contributed to this effort: Anwar Awad, John Biggs, Pin-Hung Chen, Sachin Rai, David Howard, and Sachin Idgunji.

We would also like to thank the staffs of TSMC and UMC for fabricating the technology demonstrators and enabling us to derive the results referenced in the worked examples.

Dave Flynn
Cambridge, UK

Mike Keating
Palo Alto, CA

CHAPTER 1

Introduction

1.1 Overview

The design of complex chips has undergone a series of revolutions during the last twenty years. In the 1980s there was the introduction of language-based design and synthesis. In the 1990s, there was the adoption of design reuse and IP as a mainstream design practice. In the last few years, design for low power has started to change again how designers approach complex SoC designs.

Each of these revolutions has been a response to the challenges posed by evolving semiconductor technology. The exponential increase in chip density drove the adoption of language-based design and synthesis, providing a dramatic increase in designer productivity. This approach held Moore's law at bay for a decade or so, but in the era of million gate designs, engineers discovered that there was a limit to how much new RTL could be written for a new chip project. The result was that IP and design reuse became accepted as the only practical way to design large chips with relatively small design teams. Today every SoC design employs substantial IP in order to take advantage of the ever increasing density offered by sub-micron technology.

Deep submicron technology, from 130nm on, poses a new set of design problems. We can now implement tens of millions of gates on a reasonably small die, leading to a power density and total power dissipation that is at the limits of what packaging, cooling, and other infrastructure can support. As technology has shrunk to 90nm and below, the leakage current is increasing dramatically, to the point where, in some 65nm designs, leakage current is nearly as large as dynamic current.

These changes are having a significant effect on how chips are designed. The power density of the highest performance chips has grown to the point where it is no longer

possible to increase clock speed as technology shrinks. As a result, designers are designing multi-processor chips instead of chips with a single, ultra-high speed processor.

For battery-powered devices, which comprise one of the fastest growing segments of the electronics market, the leakage of deep submicron processes is a major problem. To combat this problem, designers are using aggressive approaches at every step of the design process, from software to architecture to implementation. These approaches include power gating, where blocks are powered down when not in use, and multi-threshold libraries that can trade-off leakage current for speed.

For all applications, the total power consumption of complex SoCs presents a challenge. To address this challenge, designers are moving from a monolithic approach for power the chip—where a single supply voltage is used for all the non-IO gates of the design—to a multiple supply architecture, where different blocks are run at different voltages, depending on their individual requirements. And in some cases, designers are using voltage scaling techniques to change the supply voltage (and clock frequency) to a critical block depending on its workload and hence required performance.

This book describes a number of the techniques designers can use to reduce the power consumption of complex SoC designs. Our approach is practical, rather than theoretical. We draw heavily upon the experience we have gained in doing a series of technology demonstrator chips over the last several years. We believe the techniques we describe can be used today by chip designers to improve significantly the chips they design.

1.2 Scope of the Problem

Today some of the most powerful microprocessor chips can dissipate 100-150 Watts, for an average power density of 50-75 Watts per square centimeter. Local hot spots on the die can be several times higher than this number.

This power density not only presents packaging and cooling challenges; it also can pose problems for reliability, since the mean time to failure decreases exponentially with temperature. In addition, timing degrades with temperature and leakage increases with temperature.

Historically, the power in the highest performance chips has increased with each new technology node. But because of the issues posed by the power density, the International Technology Roadmap for Semiconductors (ITRS) predicts that the power for these chips will reach a maximum of 198 Watts in 2008; after that, power will remain constant.

Already, the total power consumption of microprocessor chips presents a significant problem for server farms. For these server farms, infrastructure costs (power, cooling) can equal the cost of the computers themselves.

For battery-powered, hand-held devices, the numbers are smaller but the problem just as serious. According to ITRS, battery life for these devices peaked in 2004. Since then, battery life has declined as features have been added faster than power (per feature) has been reduced.

For virtually all applications, reducing the power consumed by SoCs is essential in order to continue to add performance and features and grow these businesses.

Until recently, power has been a second order concern in chip design, following first order issues such as cost, area, and timing. Today, for most SoC designs, the power budget is one of the most important design goals of the project. Exceeding the power budget can be fatal to a project, whether it means moving from a cheap plastic package to an expensive ceramic one, or causing an unacceptably poor reliability due to excessive power density, or failing to meeting the required battery life.

These problems are all expected to get worse as we move to the next technology nodes. The ITRS makes the following predictions:

Table 1-1

Node	90nm	65nm	45nm
Dynamic Power per cm2	1X	1.4X	2X
Static Power per cm2	1X	2.5X	6.5X
Total Power per cm2	1X	2X	4X

Needless to say, many design teams are working very hard to reduce the growth in power below these forecast numbers, since even at 90nm many designs are at the limit of what their customers will accept.

1.3 Power vs. Energy

For battery operated devices, the distinction between power and energy is critical. Figure 1-1 on page 4 illustrates the difference. Power is the instantaneous power in the device. Energy is the area under the curve—the integral of power over time. The power used by a cell phone, for example, varies depending on the what it is doing—

whether it is in standby with the cover closed, or open and the display is powered up, or downloading from the web. The height of the graph in Figure 1-1 shows the power, but it is energy—the area under the curve—that determines battery life.

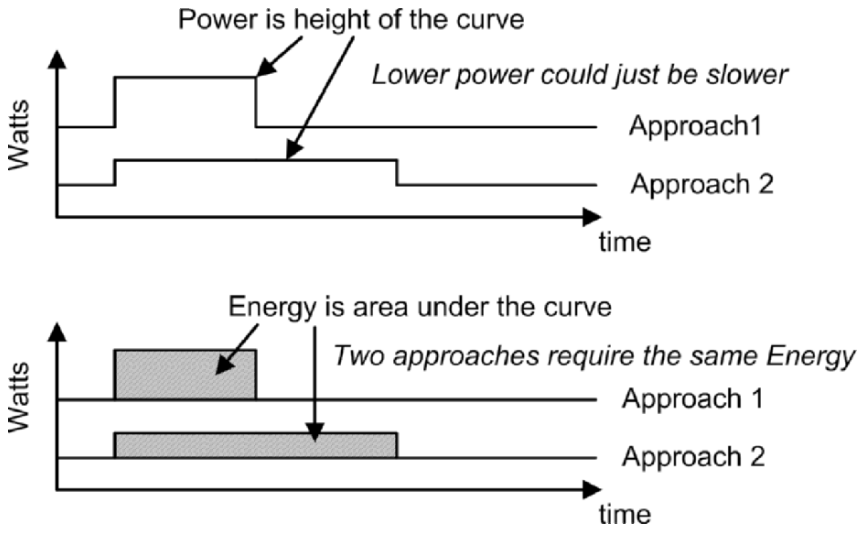

Figure 1-1 Power vs. Energy

1.4 Dynamic Power

The total power for an SoC design consists of dynamic power and static power. Dynamic power is the power consumed when the device is active—that is, when signals are changing values. Static power is the power consumed when the device is powered up but no signals are changing value. In CMOS devices, static power consumption is due to leakage.

The first and primary source of dynamic power consumption is switching power—the power required to charge and discharge the output capacitance on a gate. Figure 1-2 on page 5 illustrates switching power.

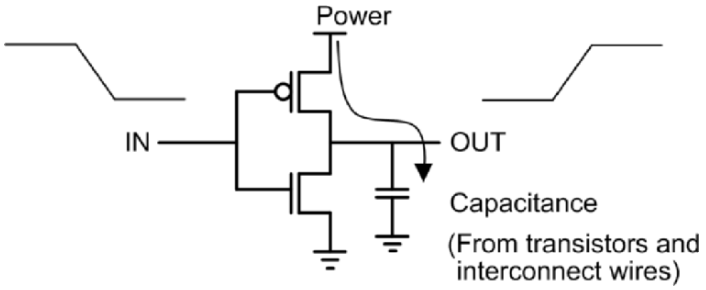

Figure 1-2 Dynamic Power

The energy per transition is given by:

$$Energy/transition = C_L \bullet V_{dd}^2$$

Where C_L is the load capacitance and V_{dd} is the supply voltage. We can then describe the dynamic power as:

$$P_{dyn} = Energy/transition \bullet f = C_L \bullet V_{dd}^2 \bullet P_{trans} \bullet f_{clock}$$

Where f is the frequency of transitions, P_{trans} is the probability of an output transition, and f_{clock} is the frequency of the system clock. If we define

$$C_{eff} = P_{trans} \bullet C_L$$

We can also describe the dynamic power with the more familiar expression:

$$P_{dyn} = C_{eff} \bullet V_{dd}^2 \bullet f_{clock}$$

Note that switching power is not a function of transistor size, but rather a function of switching activity and load capacitance. Thus, it is data dependent.

In addition to switching power, internal power also contributes to dynamic power. Figure 1-3 on page 6 shows internal switching currents. Internal power consists of the short circuit currents that occur when both the NMOS and PMOS transistors are on, as well as the current required to charge the internal capacitance of the cell.

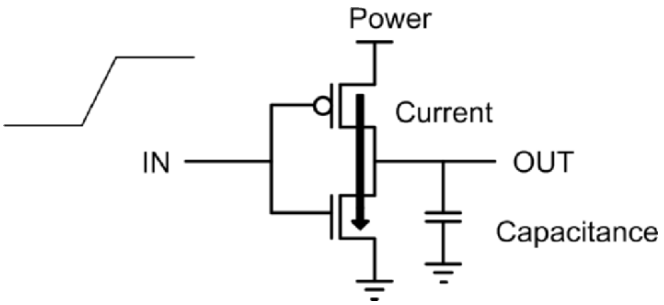

Figure 1-3 Crowbar Current

If we add the expression for internal power to our equation, we can describe the dynamic power as:

$$P_{dyn} = \left(C_{eff} \bullet V_{dd}^2 \bullet f_{clock}\right) + \left(t_{sc} \bullet V_{dd} \bullet I_{peak} \bullet f_{clock}\right)$$

Where t_{sc} is the time duration of the short circuit current, and I_{peak} is the total internal switching current (short circuit current plus the current required to charge the internal capacitance).

As long as the ramp time of the input signal is kept short, the short circuit current occurs for only a short time during each transition, and the overall dynamic power is dominated by the switching power. For this reason, we often simplify the use the switching power formula

$$P_{dyn} = C_{eff} \bullet V_{dd}^2 \bullet f_{clock}$$

But there are occasions when the short circuit current (often called crowbar current) is of interest. In particular, we will discuss ways of preventing excess crowbar current when we talk about how to deal with the floating outputs of a power gated block.

There are a number of techniques at the architectural, logic design, and circuit design that can reduce the power for a particular function implemented in a given technology. These techniques focus on the voltage and frequency components of the equation, as well as reducing the data-dependent switching activity.

There are a variety of architectural and logic design techniques for minimizing switching activity, which effectively lowers switching activity for the gates involved. An interesting example is [1], which describes how engineers have used micro-architecture modifications to reduce power significantly in Intel processors.

Because of the quadratic dependence of power on voltage, decreasing the supply voltage is a highly leveraged way to reduce dynamic power. But because the speed of a gate decreases with decreases in supply voltage, this approach needs to be done carefully. SoC designers can take advantage of this approach in several ways:

- For blocks that do not need to run particularly fast, such as peripherals, we can use a lower voltage supply than other, more speed-critical blocks. This approach is knows as multi-voltage.
- For processors, we can provide a variable supply voltage; during tasks that require peak performance, we can provide a high supply voltage and correspondingly high clock frequency. For tasks that require lower performance, we can provide a lower voltage and slower clock. This approach is known as voltage scaling.

Another approach for lowering dynamic power is clock gating. Driving the frequency to zero drives the power to zero. Some form of clock gating is used on many SoC designs.

1.5 The Conflict Between Dynamic and Static Power

The most effective way to reduce dynamic power is to reduce the supply voltage. Over the last fifteen years, as semiconductor technology has scaled, V_{DD} has been lowered from 5V to 3.3V to 2.5V to 1.2V. The ITRS road map predicts that for 2008 and 2009 high performance devices will use 1.0V and low power devices will use 0.8V.

The trouble with lowering V_{DD} is that it tends to lower I_{DS}, the *on* or drive current of the transistor, resulting in slower speeds. If we ignore velocity saturation and some of the other subtle effects that occur below 90nm, the I_{DS} for a MOSFET can be approximated by:

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

Where μ is the carrier mobility, C_{ox} is the gate capacitance, V_T is the threshold voltage and V_{GS} is the gate-source voltage. From this it is clear that, to maintain good performance, we need to lower V_T as we lower V_{DD} (and hence V_{GS}). However, lowering the threshold voltage (V_T) results in an exponential increase in the sub-threshold leakage current (I_{SUB}), as we show in the following sections.

Thus there is a conflict. To lower dynamic power we lower V_{DD}; to maintain performance we lower V_T; but the result is that we raise leakage current. Until now, this was a reasonable process, since static power from leakage current was so much lower than dynamic power. But with 90nm technology, we are getting to the point where static

power can be as big a problem as dynamic power, and we need to examine this conflict more carefully.

1.6 Static Power

There are four main sources of leakage currents in a CMOS gate (Figure 1-4)

- Sub-threshold Leakage (I_{SUB}): the current which flows from the drain to the source current of a transistor operating in the weak inversion region.
- Gate Leakage (I_{GATE}): the current which flows directly from the gate through the oxide to the substrate due to gate oxide tunneling and hot carrier injection.
- Gate Induced Drain Leakage (I_{GIDL}): the current which flows from the drain to the substrate induced by a high field effect in the MOSFET drain caused by a high V_{DG}.
- Reverse Bias Junction Leakage (I_{REV}): caused by minority carrier drift and generation of electron/hole pairs in the depletion regions.

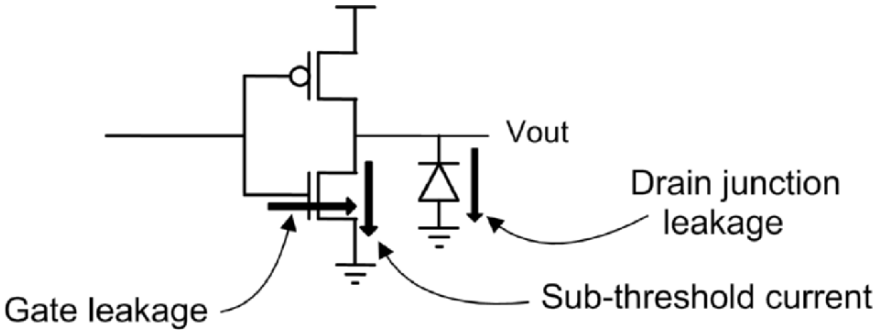

Figure 1-4 Leakage Currents

Sub-threshold leakage occurs when a CMOS gate is not turned completely off. To a good approximation, its value is given by

$$I_{SUB} = \mu C_{ox} V_{th}^2 \frac{W}{L} \cdot e^{\frac{V_{GS}-V_T}{nV_{th}}} .$$

Where W and L are the dimensions of the transistor, and V_{th} is the thermal voltage kT/q (25.9mV at room temperature). The parameter n is a function of the device fabrication process and ranges from 1.0 to 2.5.

This equation tells us that sub-threshold leakage depends exponentially on the difference between V_{GS} and V_T. So as we scale V_{DD} and V_T down (to limit dynamic power) we make leakage power exponentially worse.

Gate leakage occurs as a result of tunneling current through the gate oxide. The gate oxide thickness (T_{OX}) is only a few atoms thick in 90nm gates—this is so thin that tunneling current can become substantial. In previous technology nodes, leakage current has been dominated by sub-threshold leakage. But starting with 90nm, gate leakage can be nearly 1/3 as much as sub-threshold leakage. In 65nm it can equal sub-threshold leakage in some cases. At future nodes, high-k dielectric materials will be required to keep gate leakage in check. This appears to be the only effective way of reducing gate leakage.

Sub-threshold leakage current increases exponentially with temperature. This greatly complicates the problem of designing low power systems. Even if the leakage at room temperature is acceptable, at worst case temperature it can exceed the design goals of the chip.

There are several approaches to minimizing leakage current.

One technique is known as Multi-V_T: using high V_T cells wherever performance goals allow and low V_T cells where necessary to meet timing.

A second technique is to shut down the power supply to a block of logic when it is not active. This approach is known as power gating.

These two approaches are discussed in more detail in later chapters. For now, though, we mention three other techniques:

VTCMOS

Variable Threshold CMOS (VTCMOS) is another very effective way of mitigating standby leakage power. By applying a reverse bias voltage to the substrate, it is possible to reduce the value of the term (V_{GS}-V_T), effectively increasing V_T. This approach can reduce the standby leakage by up to three orders of magnitude. However, VTCMOS adds complexity to the library and requires two additional power networks to separately control the voltage applied to the wells. Unfortunately, the effectiveness of reverse body bias has been shown to be decreasing with scaling technology [2].

Stack Effect

The Stack Effect, or self reverse bias, can help to reduce sub-threshold leakage when more than one transistor in the stack is turned off. This is primarily because the small amount of sub-threshold leakage causes the intermediate nodes between the stacked transistors to float away from the power/ground rail. The reduced body-source potential results in a slightly negative gate-source drain voltage. Thus, it reduces the value of the term (V_{GS}-V_T), effec-

tively increasing V_T and reducing the sub-threshold leakage. The leakage of a two transistor stack has been shown to be an order of magnitude less than that of a single transistor [3]. This stacking effect makes the leakage of a logic gate highly dependent on its inputs. There is a minimum leakage state for any multi-input circuit; in theory this state applied just prior to halting the clocks to minimize leakage. In practice, applying this state is not feasible in most designs.

Long Channel Devices

From the equation for sub-threshold current, it is clear that using non-minimum length channels will reduce leakage. Unfortunately, long channel devices have lower dynamic current, degrading performance. They are also larger and therefore have greater gate capacitance, which has an adverse effect on dynamic power consumption and further degrades performance. There may not be a reduction in total power dissipation unless the switching activity of the long channel devices is low. Therefore, switching activity and performance goals must be taken in to account when using long channel devices.

1.7 Purpose of This Book

The purpose of the *Low Power Methodology Manual* is to describe the most effective new techniques for managing dynamic and static power in SoC designs. We describe the decisions that engineers need to make in designing low power chips, and provide the information they need to make good decisions. Based on our experience with real chip designs and a set of silicon technology demonstrators, we provide a set of recommendations and describe common pitfalls in doing low power design.

The process of designing a complex chip is itself very complex, involving many stakeholders and participants: systems engineers, RTL designers, IP designers, physical implementation engineers, verification engineers, and library developers. Communication between these disparate players is always a challenge. Each group has its own area of focus, its own priorities, and often its own language. One goal of this book is to give these groups a common language for discussing low power design and a common understanding of the issues involved in implementing a low power strategy.

The first low power decision an SoC design team must make, of course, is what power strategy to pursue—what techniques to use, when and where and on what section of the chip. This fundamental issue drives the structure of the book.

- Chapter 1 (this chapter) gives and over view of the challenges and basic approach to low power design.
- Chapter 2 discusses clock gating methods, Multi-V_T designs, logic-level power reduction techniques, and multi-voltage design.
- Chapter 3 gives a more detailed description of multi-voltage design, focusing on architecture and design issues.
- Chapter 4 gives an overview of power gating
- Chapter 5 addresses design aspects of power gating at the RTL level
- Chapter 6 provides an example of a power gated chip design at the RTL level
- Chapter 7 discusses architectural issues in power gating.
- Chapter 8 discusses issues in IP design for power gating, including an example.
- Chapter 9 discusses architectural and RTL level design issues in dynamic voltage and frequency scaling.
- Chapter 10 discusses some examples of voltage and frequency scaling
- Chapter 11 discusses implementation issues in low power design: synthesis, place and route, timing analysis and power analysis
- Chapter 12 discusses standard cell library and memory requirements for power gating.
- Chapter 13 discusses retention register design and data retention in memories
- Chapter 14 discusses the design of the power switching network
- Appendix A provides some additional information on the circuit design of sleep transistors and power switch networks.
- Appendix B provides detailed descriptions of the UPF commands used in the text.

Throughout the book, we will make reference to several low power technology demonstration projects that the authors have used to explore low power techniques. These projects include:

The SALT project (Synopsys ARM Low power Technology demonstrator) is a 90nm design consisting of an ARM processor and numerous Synopsys peripheral and IO IP. This project focused primarily on power gating techniques. Both the processor and the USB OTG core are power gated.

References

1. Baron, M., “Energy-Efficient Performance at Intel”, Microprocessor Report, December 11, 2006.

2. Neau, C. and Roy, K. “Optimal Body Bias Selection for Leakage Improvement and Process Compensation over Different Technology Generations,” *Proceedings of the ISLPED, 2003*

3. S. Narendra et al. “Scaling of Stack Effect and its Application for Leakage Reduction”, Int. Symp. on Low Power Electronics and Designs, pp.195-200, 2001

CHAPTER 2

Standard Low Power Methods

There are a number of power reduction methods that have been used for some time, and which are mature technologies. This chapter describes some of these approaches to low power design.:

- Clock Gating
- Gate Level Power Optimization
- Multi-V_{DD}
- Multi-V_T

2.1 Clock Gating

A significant fraction of the dynamic power in a chip is in the distribution network of the clock. Up to 50% or even more of the dynamic power can be spent in the clock buffers. This result makes intuitive sense since these buffers have the highest toggle rate in the system, there are lots of them, and they often have a high drive strength to minimize clock delay. In addition, the flops receiving the clock dissipate some dynamic power even if the input and output remain the same.

The most common way to reduce this power is to turn clocks off when they are not required. This approach is known as clock gating.

Modern design tools support automatic clock gating: they can identify circuits where clock gating can be inserted without changing the function of the logic. Figure 2-1 shows how this works.

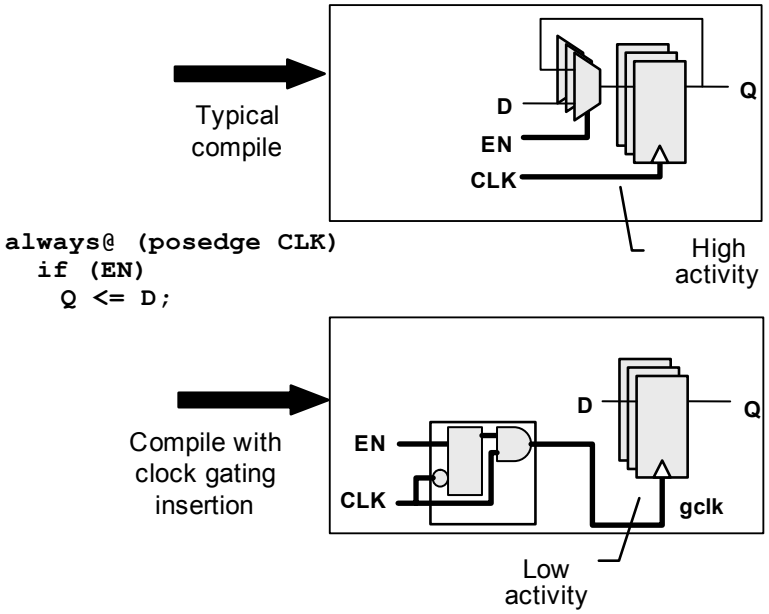

Figure 2-1 Clock Gating

In the original RTL, the register is updated or not depending on a variable (EN). The same result can be achieve by gating the clock based on the same variable.

If the registers involved are single bits, then a small savings occurs. If they are, say, 32 bit registers, then one clock gating cell can gate the clock to all 32 registers (and any buffers in their clock trees). This can result in considerable power savings.

In the early days of RTL design, engineers would code clock gating circuits explicitly in the RTL. This approach is error prone – it is very easy to create a clock gating circuit that glitches during gating, producing functional errors. Today, most libraries include specific clock gating cells that are recognized by the synthesis tool. The combination of explicit clock gating cells and automatic insertion makes clock gating a simple and reliable way of reducing power. No change to the RTL is required to implement this style of clock gating.

Results

In a recent paper [1], Pokhrel reports on a unique opportunity his team recently had to compare a (nearly) identical chip implemented both with and without clock gating. As a power reduction project, an existing 180nm chip without clock gating was re-

implemented in the same technology with clock gating. Only minor changes in the logic were implemented (some small blocks were removed and replaced by other blocks, for a small net increase in functionality).

Pokhrel reports an area reduction of 20% and a power savings of 34% to 43%, depending on the operating mode. (This savings was realized on the clock gated part of the chip; the processor was a hard macro and not clock gated. Power measurements were made on the whole chip when the processor was in IDLE mode; that is, the processor was turned off.) The power measurements are from actual silicon.

The area savings is due to the fact that a single clock gating cell takes the place of multiple muxes.

Pokhrel makes a couple of interesting observations:

- After some analysis and experiments, the team decided to use clock gating only on registers with a bit-width of at least three. They found that clock gating on one-bit registers was not power or area efficient.
- Much of the power savings was due to the fact that the clock gating cells were placed early in the clock path. Approximately 60% of the clock buffers came after the clock gating cell, and so had their activity reduce to zero during gating.

2.2 Gate Level Power Optimization

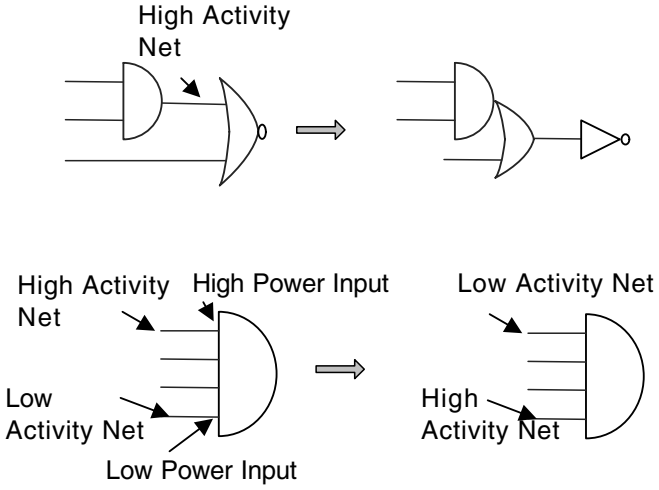

Figure 2-2 Examples of Gate Level Optimizations

In addition to clock gating, there are a number of logic optimizations that the tools can perform to minimize dynamic power. Figure 2-2 shows two of these optimizations.

At the top of the figure, an AND gate output has a particularly high activity. Because it is followed by a NOR gate, it is possible to re-map the two gates to an AND-OR gate plus an inverter, so the high activity net becomes internal to the cell. Now the high activity node (the output of the AND gate) is driving a much smaller capacitance, reducing dynamic power.

At the bottom of the figure, an AND gate has been initially mapped so that a high activity net is connected to a high power input pin, and a low activity net has been mapped to a low power pin. For multiple input gates there can be a significant difference in the input capacitance - and hence the power - for different pins. By remapping the inputs so the high activity net is connected to the low power input, the optimization tool can reduce dynamic power.

Other examples of gate level power optimization include cell sizing and buffer insertion. In cell sizing, the tool can selectively increase and decrease cell drive strength throughout the critical path to achieve timing and then reduce dynamic power to a minimum.

In buffer insertion, the tool can insert buffers rather than increasing the drive strength of the gate itself. If done in the right situations, this can result in lower power.

Like clock gating, gate level power optimization is performed by the implementation tools, and is transparent to the RTL designer.

2.3 Multi V_{DD}

Since dynamic power is proportional to V_{DD}^2, lowering V_{DD} on selected blocks helps reduce power significantly. Unfortunately, lowering the voltage also increases the delay of the gates in the design.

Consider the example in Figure 2-3. Here the cache RAMS are run at the highest voltage because they are on the critical timing path. The CPU is run at a high voltage because its performance determines system performance. But it can be run at a slightly lower voltage than the cache and still have the overall CPU subsystem performance determined by the cache speed. The rest of the chip can run at a lower voltage

still without impacting overall system performance. Often the rest of the chip is running at a much lower frequency than the CPU as well.

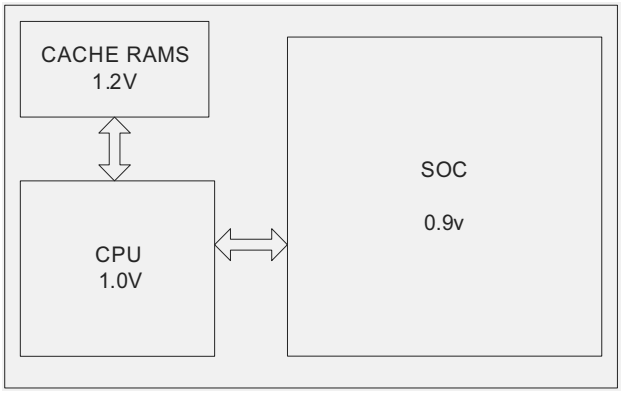

Figure 2-3 Multi-Voltage Architecture

Thus, each major component of the system is running at the lowest voltage consistent with meeting system timing. This approach can provide significant savings in power.

Mixing blocks at different V_{DD} supplies adds some complexity to the design – not only do we need to add IO pins to supply the different power rails, but we also need a more complex power grid and level shifters on signals running between blocks. These issues are described in more detail later in the book.

2.4 Multi-Threshold Logic

As geometries have shrunk to 130nm, 90nm, and below, using libraries with multiple V_T has become a common way of reducing leakage current.

Figure 2-4 shows the relationship between delay and leakage for a 90nm process. Figure 2-5 shows some representative curves for leakage vs. delay for a multi-V_T library. As explained earlier, sub-threshold leakage depends exponentially on V_T. Delay has a much weaker dependence on V_T.

Many libraries today offer two or three versions of their cells: Low V_T, Standard V_T, and High V_T. The implementation tools can take advantage of these libraries to optimize timing and power simultaneously.