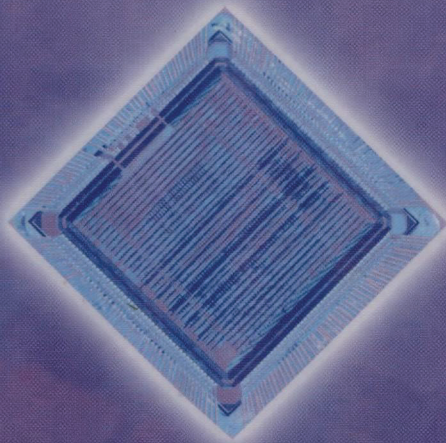


Low Power Methodology Manual For System-on-Chip Design



Michael Keating David Flynn
Robert Aitken Alan Gibbons Kaijian Shi

 Springer

AMD-ITC-1350-0190509

REALTEK EXHIBIT 2013

AMD v. Realtek

IPR2023-00920

U.C. BERKELEY LIBRARY

AMD-ITC-1350-0190510

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

Low Power Methodology Manual

For System-on-Chip Design

 Springer

AMD-ITC-1350-0190511

engr
015783169

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.

Copyright © 2007 by Synopsys, Inc. & ARM Limited. All rights reserved.

All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, LLC, 233 Spring Street, New York, NY 10013, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden. The use in this publication of trade names, trademarks, service marks and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

9 8 7 6 5 4 3 2 1

springer.com

cmr 11/27/07

AMD-ITC-1350-0190512

TK7874
.66
L67
2007
ENGI

TRADEMARKS

Synopsys and NanoSim are registered trademarks of Synopsys, Inc.

ARM and AMBA are registered trademarks of ARM Limited. ARM926EJ-S, ARM1176JZF-S, AHB and APB are trademarks of ARM Limited. Artisan and Artisan Components are registered trademarks of ARM Physical IP, Inc.

“ARM” is used to represent ARM Holdings plc; its operating company ARM Limited; and the regional subsidiaries ARM INC.; ARM KK; ARM Korea Ltd.; ARM Taiwan; ARM France SAS; ARM Consulting (Shanghai) Co. Ltd.; ARM Belgium N.V.; AXYS Design Automation Inc.; AXYS GmbH; ARM Embedded Technologies Pvt. Ltd.; and ARM, Inc. and ARM Norway, AS.

All other brands or product names are the property of their respective holders.

DISCLAIMER

All content included in this Low Power Methodology Manual is the result of the combined efforts of ARM Limited and Synopsys, Inc. Because of the possibility of human or mechanical error, neither the authors, ARM Limited, Synopsys, Inc., nor any of their affiliates, including but not limited to Springer Science+Business Media, LLC, guarantees the accuracy, adequacy or completeness of any information contained herein and are not responsible for any errors or omissions, or for the results obtained from the use of such information. THERE ARE NO EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE relating to the Low Power Methodology Manual. In no event shall the authors, ARM Limited, Synopsys, Inc., or their affiliates be liable for any indirect, special or consequential damages in connection with the information provided herein.

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.

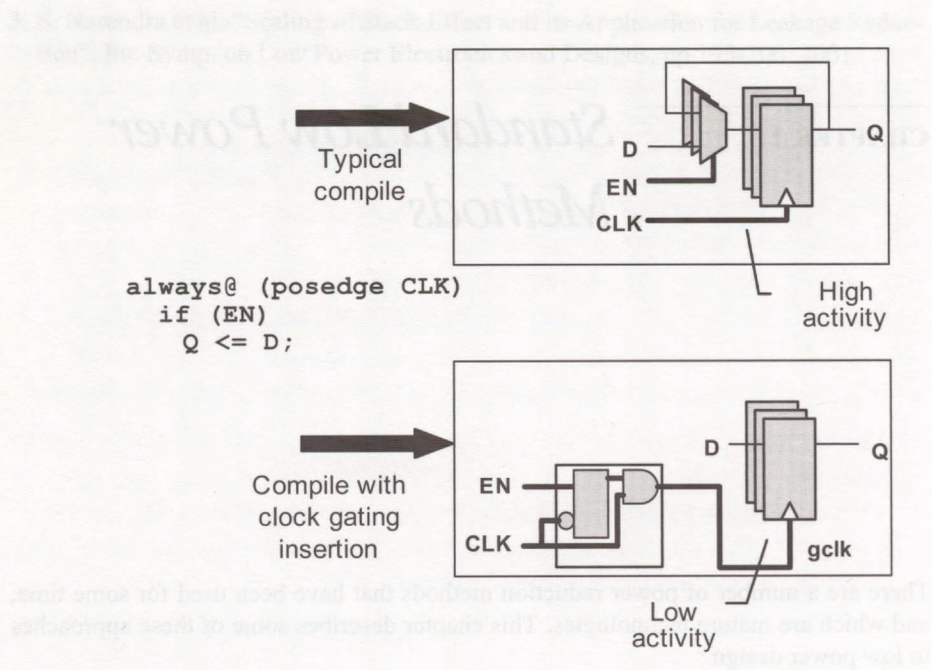


Figure 2-1 Clock Gating

In the original RTL, the register is updated or not depending on a variable (EN). The same result can be achieved 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

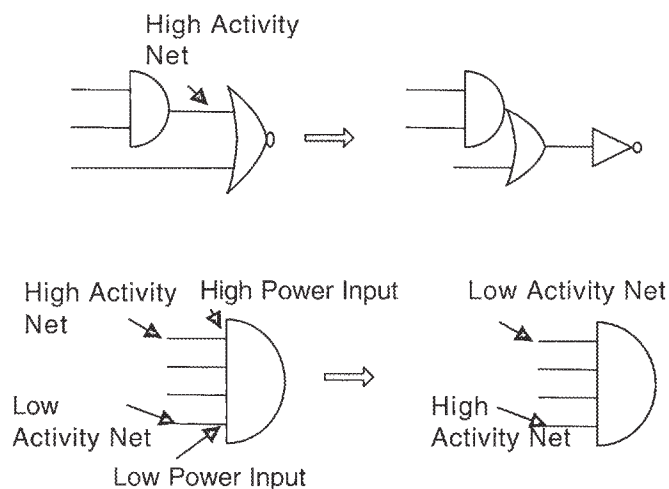


Figure 2-2 Examples of Gate Level Optimizations

The techniques discussed in the previous chapter are mature; engineers have been using them for some time, and design tools have supported them for years. With this chapter, we begin discussing more recent and aggressive approaches to reducing power: power gating and adaptive voltage scaling.

Both of these techniques rely on moving away from the traditional approach of using a single, fixed supply rail for all of the (internal) gates in a design. (IO cells have had a separate power supply in most chips for many years).

The most basic form of this new approach is to partition the internal logic of the chip into multiple voltage regions or power domains, each with its own supply. This approach is called Multi-Voltage design. It is based on the realization that in a modern SoC design, different blocks have different performance objectives and constraints. A processor, for instance, may need to run as fast as the semiconductor technology will allow. In this case, a relatively high supply voltage is required. A USB block, on the other hand, may run at a fixed, relatively low frequency dictated more by the protocol than the underlying technology. In this case, a lower supply rail may be sufficient for the block to meet its timing constraints – and a lower supply rail means that its dynamic and static power will be lower.

Once we have crossed the conceptual barrier of having separate supplies, there are more complex power strategies we can contemplate: we can provide different voltages to our processor, for example, depending on its workload. Or we can provide different voltages to a RAM – a low voltage to maintain memory contents when the memory is not being accessed, and a higher voltage that supports reads and writes. We can even consider dropping the supply voltage to zero – that is, power gating.

For the sake of discussion we provide the following categorization of multi-voltage strategies:

- Static Voltage Scaling (SVS): different blocks or subsystems are given different, fixed supply voltages.
- Multi-level Voltage Scaling (MVS): an extension of the static voltage scaling case where a block or subsystem is switched between two or more voltage levels. Only a few, fixed, discrete levels are supported for different operating modes.
- Dynamic Voltage and Frequency Scaling (DVFS): an extension of MVS where a larger number of voltage levels are dynamically switched to follow changing workloads.
- Adaptive Voltage Scaling (AVS): an extension of DVFS where a control loop is used to adjust the voltage.

3.1 Challenges in Multi-Voltage Designs

Even the simplest multi-voltage design presents the designer with some basic challenges:

- Level shifters. Signals that go between blocks that use different power rails often require level shifters – buffers that translate the signal from one voltage swing to another.
- Characterization and STA. With a single supply for the entire chip, timing analysis can be done at a single performance point. The libraries are characterized for this point, and the tools perform the analysis in a straight-forward manner. With multiple blocks running at different voltages, and with libraries that may not be characterized at the exact voltage we are using, timing analysis becomes much more complex.
- Floor planning, power planning, grids. Multiple power domains require more careful and detailed floorplanning. The power grids become more complex.
- Board level issues. Multi-voltage designs require additional resources on the board – additional regulators to provide the additional supplies.
- Power up and power down sequencing. There may be a required sequence for powering up the design in order to avoid deadlock.

3.2 Voltage Scaling Interfaces – Level Shifters

When driving signals between power domains with radically different power rails, the need for level shifters is clear. Driving a signal from a 1V domain to a 5V domain is a problem – the 1V swing may not even reach threshold in the 5V domain. But the internal voltages in today's chips are tightly clustered around 1V. Why would we need level shifters on signals going from a 0.9V domain to a 1.2V domain?

One fundamental reason is that a 0.9V signal driving a 1.2V gate will turn on both the NMOS and PMOS networks, causing crowbar currents. This issue is discussed later in this chapter.

In addition, standard cell libraries are characterized for – and operate best with – a clean, fast input that goes rail to rail. Failure to meet this requirement may result in signals exhibiting significant rise- or fall-time degradation between the driver cell in one domain and the receiver in another voltage domain. This in turn can lead to timing closure problems and even excessive crowbar switching currents.

The best solution is to make sure each domain gets the voltage swings (and rise- and fall-times) that it expects. We do this by providing level shifters between any domains that use different voltages. This approach limits any voltage swing and timing characterization issues to the boundary of voltage domains, and leaves the internal timing of the domain unaffected. This kind of clean interfacing makes timing closure – and reuse – much easier.

3.2.1 Unidirectional Level Shifters

The design of a level shifter to provide an effective voltage swing between one different voltage rails is an analog design problem. And for analog design reasons, these cells are typically only designed to shift one direction - either from a higher voltage to a lower one, or from a lower voltage to a higher one. Later in this chapter we provide some example designs that show the difference between the two types of cells.

For static voltage scaling, this limitation on level shifters is not a problem. But for the other forms of multi-voltage, where supply voltages can change during operation, it does pose a challenge. The designer must architect and partition the design such that voltage domains have a defined relation to neighboring domains – such as “always higher”, “always lower”, or “always the same.” With this restriction, it then becomes straightforward to implement the interface with the appropriate level shifting components.

Designing interfaces that can operate in both directions may appear attractive from a system perspective but requires non-standard implementation components and tooling.

3.2.2 Level Shifters – High to Low Voltage Translation

On the face of it, simply overdriving a CMOS input from an output buffer on a higher voltage rail does not appear to be a problem – there are no latch-up or breakdown issues, simply a “better”, faster edge compared to normal CMOS logic high or low level switching levels.