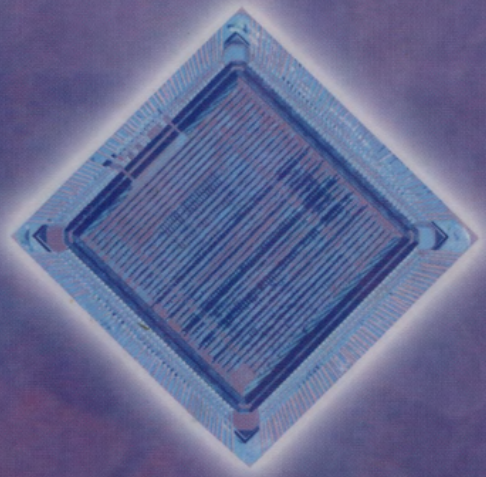
ATTACHMENT 6

Low Power Methodology Manual

For System-on-Chip Design



Michael Keating David Flynn
Robert Aitken Alan Gibbons Kaijian Shi

Springer

AMD EX1021
AMD v. Realtek - Page 268

U.C. BERKELEY LIBRARY

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

Low Power Methodology Manual

For System-on-Chip Design

Springer

AMD EX1021
AMD v. Realtek - Page 270

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

cm 11/27/07

TK7874
.66
L67
2007
ENG1

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.

Table of Contents

Preface ..... **xv**

1 Introduction **1**

1.1 Overview 1
1.2 Scope of the Problem 2
1.3 Power vs. Energy 3
1.4 Dynamic Power 4
1.5 The Conflict Between Dynamic and Static Power 7
1.6 Static Power 8
1.7 Purpose of This Book 10

2 Standard Low Power Methods **13**

2.1 Clock Gating 13
2.2 Gate Level Power Optimization 15
2.3 Multi VDD 16
2.4 Multi-Threshold Logic 17
2.5 Summary of the Impact of Standard Low Power Techniques 19

3 Multi-Voltage Design **21**

3.1 Challenges in Multi-Voltage Designs 22
3.2 Voltage Scaling Interfaces – Level Shifters 22
3.2.1 Unidirectional Level Shifters 23

AMD EX1021
AMD v. Realtek - Page 273

3.2.2 Level Shifters – High to Low Voltage Translation 23
3.2.3 Level Shifters – Low-to-High Voltage 24
3.2.4 Level Shifter Placement 25
3.2.5 Automation and Level Shifters 27
3.2.6 Level Shifter Recommendations and Pitfalls 28
3.3 Timing Issues in Multi-Voltage Designs 29
3.3.1 Clocks 29
3.3.2 Static Timing Analysis 30
3.4 Power Planning for Multi-Voltage Design 30
3.5 System Design Issues with Multi-Voltage Designs 31

4 Power Gating Overview 33

4.1 Dynamic and Leakage power profiles 33
4.2 Impact of Power Gating on Classes of Sub-systems 36
4.3 Principles of Power Gating Design 37
4.3.1 Power Switching – Fine Grain vs. Coarse Grain 38
4.3.2 The Challenges of Power Gating 39

5 Designing Power Gating 41

5.1 Switching Fabric Design 42
5.1.1 Controlling the Switching Fabric 44
5.1.2 Recommendations and Pitfalls for Power Gating Control 44
5.2 Signal Isolation 45
5.2.1 Signal Isolation techniques 45
5.2.2 Output or Input Isolation 47
5.2.3 Interface Protocols and Isolation 48
5.2.4 Recommendations and Pitfalls for Isolation 50
5.3 State Retention and Restoration Methods 50
5.3.1 State Retention Using Scan Chains 51
5.3.2 Retention Registers 54
5.3.3 Power Controller Design for Retention 56
5.3.4 Partial vs. Full State Retention 56
5.3.5 System Level Issues and Retention 58
5.3.6 Recommendations and Pitfalls for State Retention 58
5.4 Power Gating Control 59
5.4.1 Power Control Sequencing 60
5.4.2 Handshake Protocols 61
5.4.3 Recommendations and Pitfalls for Power Gating Controllers 63
5.5 Power Gating Design Verification – RTL Simulation 63

AMD EX1021
AMD v. Realtek - Page 274

5.5.1 Inferring Power Gating Behavior in RTL 64
5.5.2 Inferring Power Gating and Retention Behavior in RTL 68
5.6 Design For Test considerations 70
5.6.1 Power Gating Controls 70
5.6.2 Power Limitations during Scan Test 71
5.6.3 Testing the Switching Network 71
5.6.4 Testing Isolation and Retention 72
5.6.5 Testing the Power Gating Controller 73

6 Architectural Issues for Power Gating 75

6.1 Hierarchy and Power Gating 75
6.2 Power Networks and Their Control 78
6.2.1 External Power Rail Switching 79
6.2.2 On-chip Power Gating 81
6.3 Power State Tables and Always On Regions 82

7 A Power Gating Example 85

7.1 Leakage Modes Supported 85
7.2 Design partitioning 88
7.3 Isolation 92
7.4 Retention 94
7.5 Inferring Power Gating and Retention 95
7.6 Measurements and Analysis 96

8 IP Design for Low Power 101

8.1 Architecture and Partitioning for Power Gating 102
8.1.1 How and When to Shut Down 103
8.1.2 What to Shut Down and What to Keep Alive 103
8.2 Power Controller Design for the USB OTG 105
8.3 Issues in Designing Portable Power Controllers 108
8.4 Clocks and Resets 109
8.5 Verification 109
8.6 Packaging IP for Reuse with Power Intent 110
8.7 UPF for the USB OTG Core 111
8.8 USB OTG Power Gating Controller State Machine 114

9 Frequency and Voltage Scaling Design 121

9.1 Dynamic Power and Energy 122
9.2 Voltage Scaling Approaches 125
9.3 Dynamic Voltage and Frequency Scaling (DVFS) 125
9.4 CPU Subsystem Design Issues 129
9.5 Adaptive Voltage Scaling (AVS) 130
9.6 Level Shifters and Isolation 131
9.7 Voltage Scaling Interfaces – Effect on Synchronous Timing 132
9.8 Control of Voltage Scaling 136

10 Examples of Voltage and Frequency Scaling Design 139

10.1 Voltage Scaling - A Worked Example for UMC 130nm 139
10.1.1 ULTRA926 System Design Block Diagram 140
10.1.2 Voltage/Frequency Range Exploration 141
10.1.3 Synchronous Design Constraints 144
10.1.4 Simulated (predicted) Energy Savings Analysis 145
10.1.5 Silicon-Measured Power and Performance Analysis 145
10.1.6 Silicon-Measured ULTRA926 DVFS Energy Savings Analysis .. 147
10.2 Voltage Scaling – A worked Example for TSMC 65nm 150
10.2.1 ATLAS926 Case Study 150
10.2.2 Voltage/Frequency Range Exploration 151
10.2.3 Silicon-Measured Power and Performance Analysis 151

11 Implementing Multi-Voltage, Power Gated Designs.....155

11.1 Design Partitioning 158
11.1.1 Logical and Physical Hierarchy 158
11.1.2 Critical Path Timing 160
11.2 Design Flow Overview 160
11.3 Synthesis 162
11.3.1 Power Intent 162
11.3.2 Defining Power Domains and Power Connectivity 162
11.3.3 Isolation Cell Insertion 163
11.3.4 Retention Register Insertion 164
11.3.5 Level Shifter Insertion 166
11.3.6 Scan Synthesis 168
11.3.7 Always-On Network Synthesis 170
11.4 Multi Corner Multi Mode Optimization with Voltage Scaling Designs 171
11.5 Design Planning 173

AMD EX1021
AMD v. Realtek - Page 276

11.5.1 Creating Voltage Areas 173
11.5.2 Power Gating Topologies 175
11.5.3 In-rush Current Management 176
11.5.4 Recommendations: 176
11.6 Power Planning 177
11.6.1 Decoupling Capacitor Insertion 179
11.7 Clock Tree Synthesis 180
11.8 Power Analysis 183
11.9 Timing Analysis 184
11.10 Low Power Validation 185
11.11 Manufacturing Test 185

12 Physical Libraries 187

12.1 Standard Cell Libraries 187
12.1.1 Modeling of Standard Cell Libraries 188
12.1.2 Characterization of Standard Cell Libraries 189
12.2 Special Cells - Isolation Cells 190
12.2.1 Signal Isolation 191
12.2.2 Output Isolation vs. Input Isolation 193
12.2.3 Sneak DC Leakage Paths 193
12.2.4 Recommendations 194
12.3 Special Cells - Level Shifters 195
12.4 Memories 198
12.4.1 RAMs for Multi-Voltage Power Gated Designs 199
12.4.2 Memories and Retention 200
12.5 Power Gating Strategies and Structures 200
12.5.1 Power Gating Structures 201
12.5.2 Recommendations – Coarse Grain vs. Fine Grain 204
12.6 Power Gating Cells 204
12.7 Power Gated Standard Cell Libraries 206

13 Retention Register Design 209

13.1 Retention Registers 209
13.1.1 Single Pin "Live Slave" Retention Registers 209
13.1.2 Dual Control Signal "Balloon" Retention Register 212
13.1.3 Single Control Signal "Balloon" Retention Register 215
13.1.4 Retention Register: Relative layout 218
13.2 Memory Retention Methods 219
13.2.1 VDD Retention Method 219

AMD EX1021
AMD v. Realtek - Page 277

13.2.2 Source-diode Biasing Method 219
13.2.3 Source Biasing Method 221
13.2.4 Retention Latency Reduction Methods 222

14 Design of the Power Switching Network 225

14.1 Ring vs. Grid Style 225
14.1.1 Ring Style Implementation 226
14.1.2 Grid Style Implementation 227
14.1.3 Row and Column Grids 229
14.1.4 Hybrid Style Implementation 231
14.1.5 Recommendations - Ring vs. Grid Style 231
14.2 Header vs. Footer Switch 232
14.2.1 Switch Efficiency Considerations 232
14.2.2 Area Efficiency Consideration and L/W Choice 234
14.2.3 Body Bias Considerations 235
14.2.4 System Level Design Consideration 235
14.2.5 Recommendations – Header vs. Footer 235
14.3 Rail vs. Strap VDD Supply 236
14.3.1 Parallel Rail VDD Distribution 236
14.3.2 Power Strap VDD Distribution 238
14.3.3 Recommendations for Supply Distribution 239
14.4 A Sleep Transistor Example 239
14.5 Wakeup Current and Latency Control Methods 240
14.5.1 Single Daisy chain sleep transistor distribution 241
14.5.2 Dual Daisy chain sleep transistor distribution 241
14.5.3 Parallel Short Chain Distribution of the Main Sleep Transistor ... 243
14.5.4 Main Chain Turn-on Control 243
14.5.5 Buffer Delay Based Main Chain Turn-on Control 243
14.5.6 Programmable Main Chain Turn-on Control 244
14.5.7 Power-off Latency Reduction 244
14.5.8 Recommendations for Power Switching Control 245
14.6 An Example of a Dual Daisy Chain Sleep Transistor Implementation 246

APPENDIX A Sleep Transistor Design 249

A.1 Sleep Transistor Design Metrics 250
A.1.1 Switch Efficiency 250
A.1.2 Area Efficiency 253
A.1.3 IR Drop 253
A.1.4 Normal vs. Reverse Body Bias 254

AMD EX1021
AMD v. Realtek - Page 278

A.1.5 Recommendations 259
A.2 Layout Design for Area Efficiency 260
A.2.1 Recommendations 262
A.3 Single Row vs. Double Row 262
A.3.1 Recommendations 263
A.4 In-rush Current and Latency Analysis 263

APPENDIX B UPF Command Syntax 267

B.1 add_pst_state 268
B.2 connect_supply_net 269
B.3 create_power_domain 271
B.4 create_power_switch 273
B.5 create_pst 275
B.6 create_supply_net 276
B.7 create_supply_port 277
B.8 set_domain_supply_net 278
B.9 set_isolation 279
B.10 set_isolation_control 281
B.11 set_level_shifter 283
B.12 set_retention 285
B.13 set_retention_control 287
B.14 set_scope 288

Glossary 291

Bibliography 293

Index 297

AMD EX1021
AMD v. Realtek - Page 279

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

AMD EX1021
AMD v. Realtek - Page 280

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

AMD EX1021
AMD v. Realtek - Page 281

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.

AMD EX1021
AMD v. Realtek - Page 282

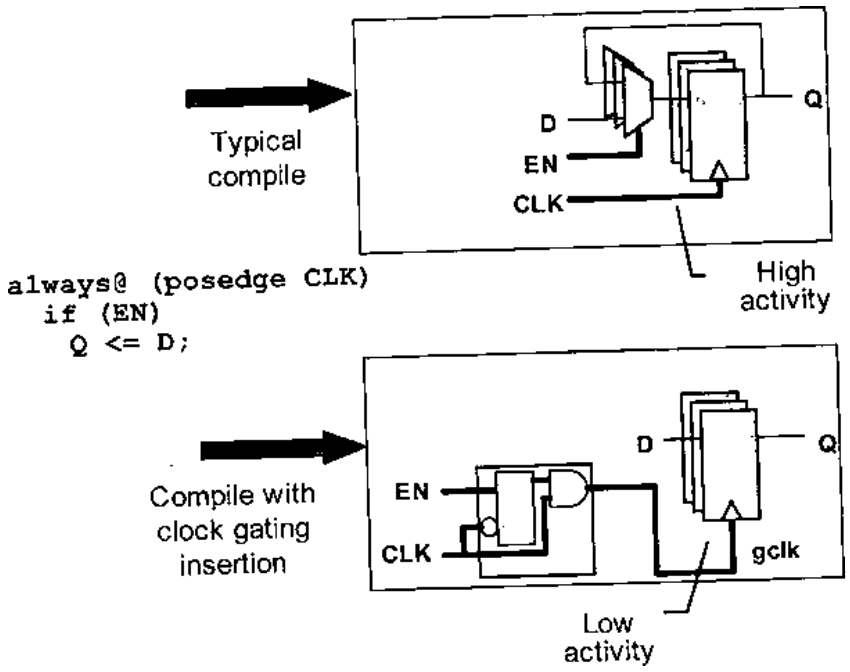


Figure 2-1 Clock Gating

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-

AMD EX1021
AMD v. Realtek - Page 283

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

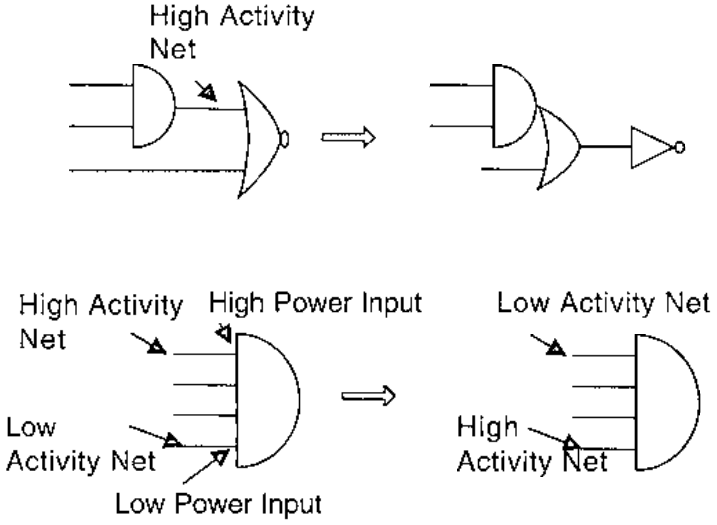


Figure 2-2 Examples of Gate Level Optimizations

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

AMD EX1021
AMD v. Realtek - Page 285

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

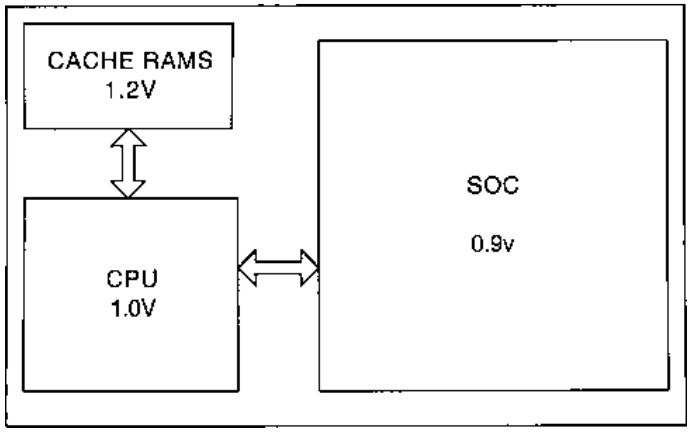


Figure 2-3 Multi-Voltage Architecture

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 voltage.

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.

AMD EX1021
AMD v. Realtek - Page 286

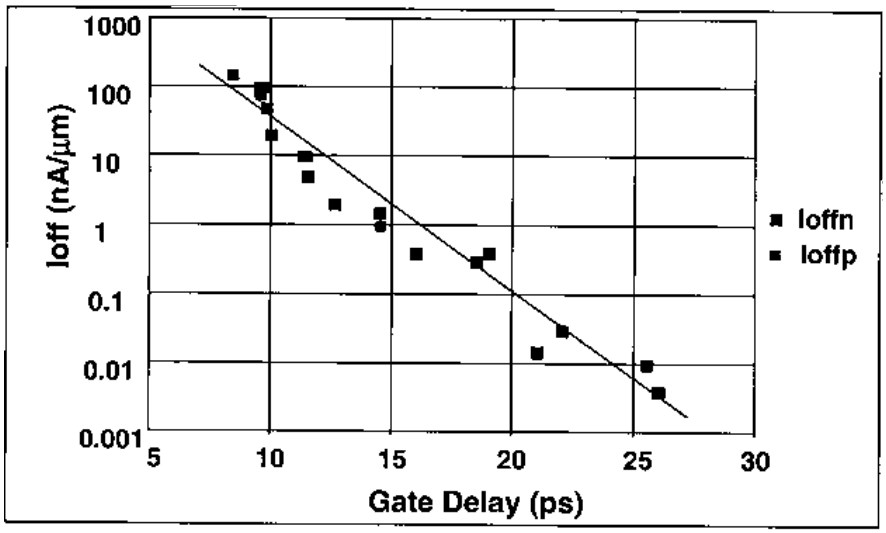


Figure 2-4 Delay vs. Leakage for 90nm

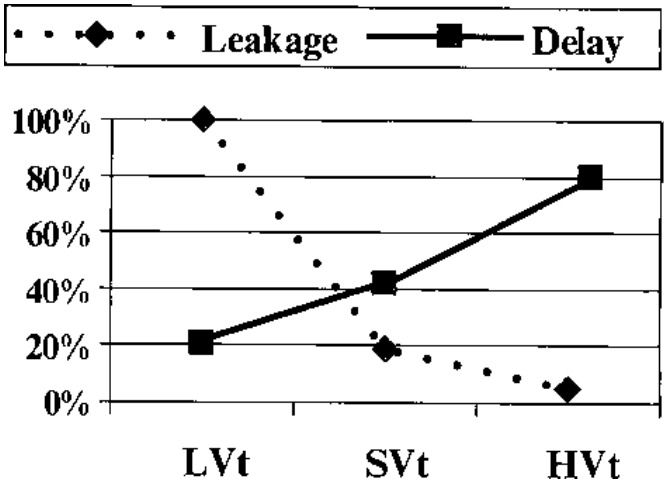


Figure 2-5 Leakage vs. Delay for a 90nm Library

It is now quite common to use a "Dual V_T" flow during synthesis. The goal of this approach is to minimize the total number of fast, leaky low V_T transistors by deploying them only when required to meet timing. This usually involves an initial synthesis targeting a primary library followed by an optimization step targeting one (or more) additional libraries with differing thresholds.

Usually there is a minimum performance which must be met before optimizing power. In practice this usually means synthesizing with the high performance, high

AMD EX1021
AMD v. Realtek - Page 287

leakage library first and then relaxing back any cells not on the critical path by swapping them for their lower performing, lower leakage equivalents.

If minimizing leakage is more important than achieving a minimum performance then this process can be done the other way around: we can target the low leakage library first and then swap in higher performing, high leakage equivalents in speed critical areas.

2.5 Summary of the Impact of Standard Low Power Techniques

Table 2-1 provides a brief summary of the cost/benefit of the techniques described in this chapter.

Table 2-1

Technique	Power Benefit	Timing Penalty	Area Penalty	Impact: Architecture	Impact: Design	Impact: Verification	Impact: Place & Route
Multi Vt	Medium	Little	Little	Low	Low	None	Low
Clock Gating	Medium	Little	Little	Low	Low	None	Low
Multi Voltage	Large	Some	Little	High	Medium	Low	Medium

References

1. Pokhrel, K. "Physical and Silicon Measures of Low Power Clock Gating Success: An Apple to Apple Case Study", SNUG, 2007 http://www.snug-universal.org/cgi-bin/search/search.cgi?San+Jose,+2007.

AMD EX1021
AMD v. Realtek - Page 288

CHAPTER 7

A Power Gating Example

The SALT technology demonstrator project provided a platform for testing the approaches to power gating and state retention described in this book. In this chapter we give some more details on the system design and RTL coding for this project.

The SALT chip is implemented in 90nm generic technology and contains an ARM processor, an AMBA bus and set of peripherals, and a Synopsys USB OTG digital core and PHY. The ARM core and the USB core are independently power gated. The ARM core uses full state retention; the USB uses partial state retention. Both the ARM core and the USB use switching fabrics of header switches; thus they switch VDD and use a common ground (VSS).

In this first section, we describe the power gating design for the processor. Figure 7-1 on page 86 shows a simplified block diagram of the SALT chip.

7.1 Leakage Modes Supported

Most battery-powered processor based designs have to deal carefully with the balance between performance (to support product features) and low power (to support long battery life). The performance requirement steered us to a higher performance, leakier process. To maintain long battery life, we needed to provide aggressive leakage management.

AMD EX1021
AMD v. Realtek - Page 289

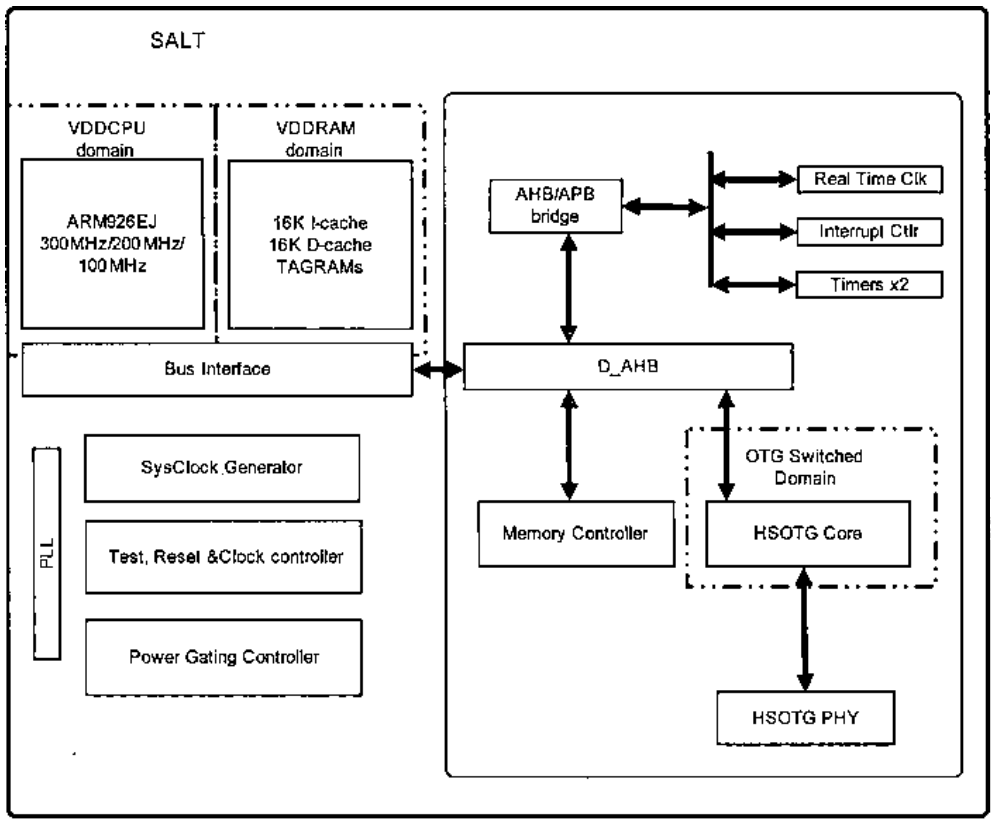


Figure 7-1 SALT Block Diagram

For the SALT project, the processor uses four low-power modes. In all modes, the power controller generates a **SLEEP** signal to enter the low power mode and the **WAKE** signal to exit. In order of increasing leakage savings – and increasing time to power up and power down – the modes are:

- **HALT**: SLEEP turns off the clocks to the processor, WAKE restarts the clocks. Power remains on.
- **SNOOZE**: SLEEP initiates power gating with state retention; power is switched off to the processor itself, but the cache memory remains powered up. WAKE initiates the power up and restore sequence; power is brought up as fast as safety allows.
- **HIBERNATE**: SLEEP initiates a sequence of scanning the processor's internal state to external memory; the VDDCPU power rail is then switched off. WAKE causes the power rail to be switched back on, and the scan chains are used to restore the internal state of the processor. A 32-bit AMBA-based bus is used to write to and read from an external memory. A 32-bit CRC provides integrity checking. The CRC is saved along with the scanned data and used to protect

AMD EX1021
AMD v. Realtek - Page 290

against restarting with corrupted state. Note that VDDCPU provides power to the processor logic, not cache. So in HIBERNATE mode, only the processor is powered down; the cache memory remains powered up.

- **SHUTDOWN**: SLEEP initiates the same power down sequence as in Hibernate, but now both the processor and the cache have their external power supplies turned off. Explicit code must be called to write back any dirty data in the cache memories before both the VDDCPU and VDDRAM supplies can be power-rail switched. This mode is the only mode not transparent to the operating system.

In addition to these modes, the SALT chip provides active leakage reduction through externally managed threshold scaling. This threshold scaling is done using back-bias control. Both P- and N-wells for the CPU standard cell area are brought out to the chip's pins. This enables experimental analysis of delay and leakage power characteristics as the bias is varied.

Using this well-biasing scheme, we defined three operating modes:

- **NORMAL**: with standard well-bias
- **TURBO-RUN** – with the wells forward biased for increased speed
- **POWERSAVER-RUN** – with the wells reversed biased for reduced leakage

The power controller is designed to ensure that well-bias voltages are only changed while the design is static and un-clocked. Any change between "normal" and back- or forward biased modes of active power management go through one of the HALT, SNOOZE, or HIBERNATE states. Figure 7-2 on page 88 shows the state diagram for the power modes.

AMD EX1021
AMD v. Realtek - Page 291

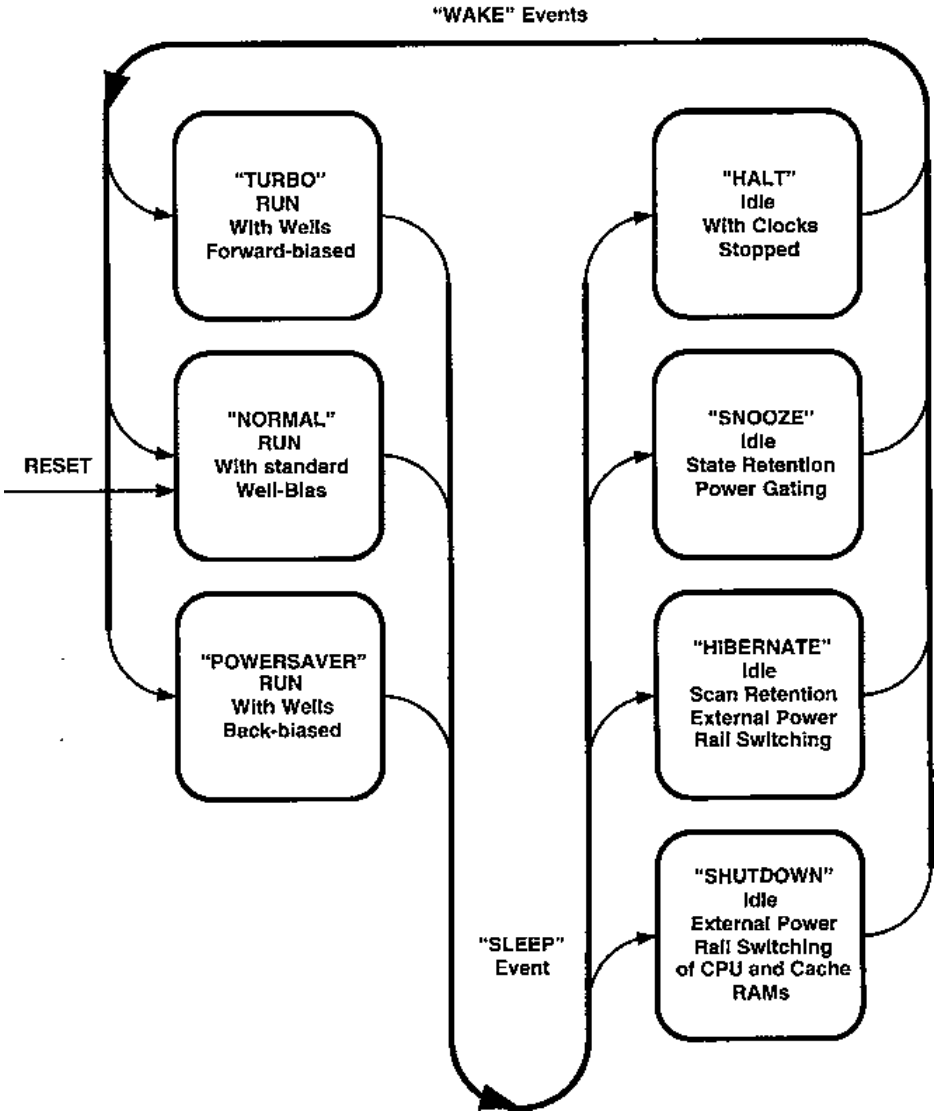


Figure 7-2 Power States for SALT

7.2 Design Partitioning

The RTL design is partitioned to allow the three primary power supplies to be mapped to the RTL design:

- **VDDSOC** is the "always-on" supply that powers the entire chip except the processor and its cache.

AMD EX1021
AMD v. Realtek - Page 292

- **VDDRAM** is an external switched power rail that supplies the Cache and MMU RAMs.
- **VDDCPU** is an external switched power rail that supplies the CPU standard cell area.

VDDSOC provides power to the digital side of the PLLs, the clock generators and the power management control blocks, plus all the real-time peripherals. These peripherals include a real-time clock and timers; these can generate wake-up events as part of their interrupt service requests.

VDDSOC also provides power to the USB OTG subsystem. The USB is power gated though its own power switching fabric using a separate power controller.

Using a separate power supply (VDDRAM) for the cache accomplishes several objectives: it allows the CPU to be powered down while keeping the cache powered, allowing faster restore. It allows the CPU and cache to be powered down while keeping the peripherals of the chip powered up. These peripherals control the wake up of the CPU – upon detecting an interrupt, the interrupt controller signals the power controller to power up the CPU. Finally, using a separate cache power supply allows development of detailed leakage and active power consumption profiles, so that we can determine the minimum operating voltage for the cache memory and the minimum voltage that assures that the cache retains is contents during CPU shut-down.

Similarly, using a separate supply (VDDCPU) for the CPU logic allows detailed leakage and active power consumption profiles to be measured as well as the time and energy cost to get in and out of each power saving state. Figure 7-3 shows the partitioning of the SALT chip.

AMD EX1021
AMD v. Realtek - Page 293

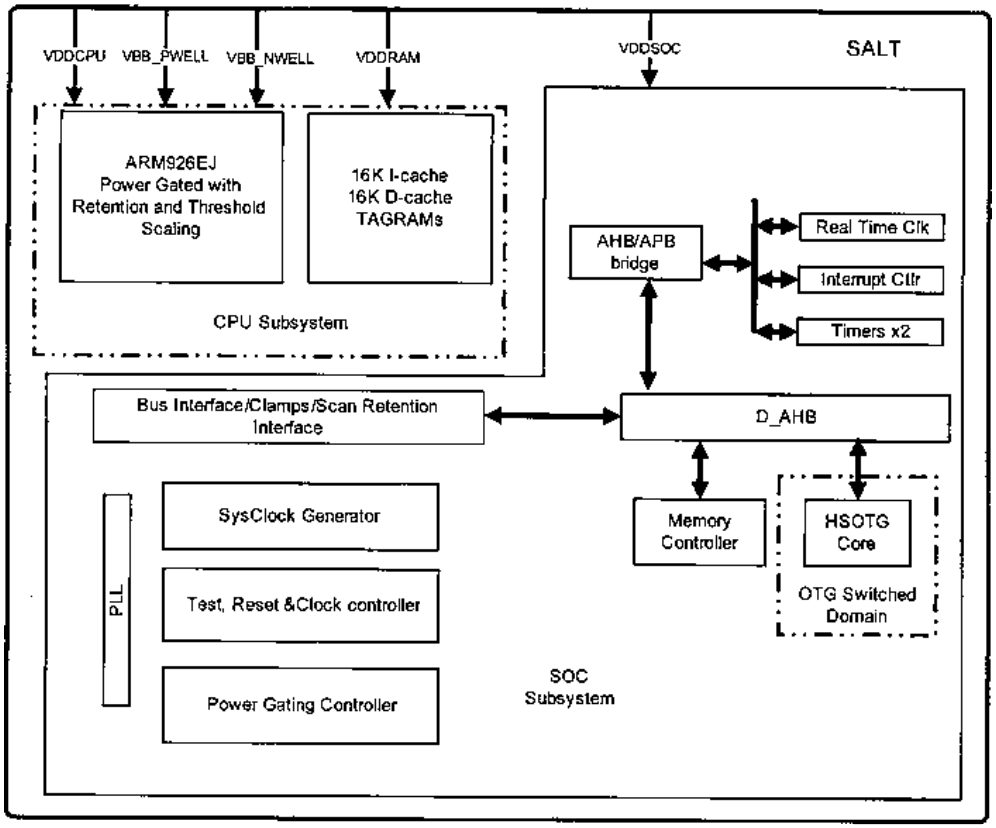


Figure 7-3 Partitioning of the SALT Chip

Since the original SALT design, the CPU sub-system has been re-implemented to improve design re-use. The Bus Interface Unit and the Power-Gating Controller are now integrated into a "VDDSOC" region within the CPU. This integration significantly reduces the number of signals – and blocks – that the SoC design team has to deal with when integrating the ARM core into the chip.

Although this makes the 3-supply-rail CPU subsystem slightly more complex to implement, the timing and internal power gating and isolation interfaces are now all abstracted away from the top-level SOC design. Any changes or enhancements to the low-power states supported by the IP block are independent of the top level system design.

Figure 7-4 shows the re-partitioned CPU subsystem.

AMD EX1021
AMD v. Realtek - Page 294

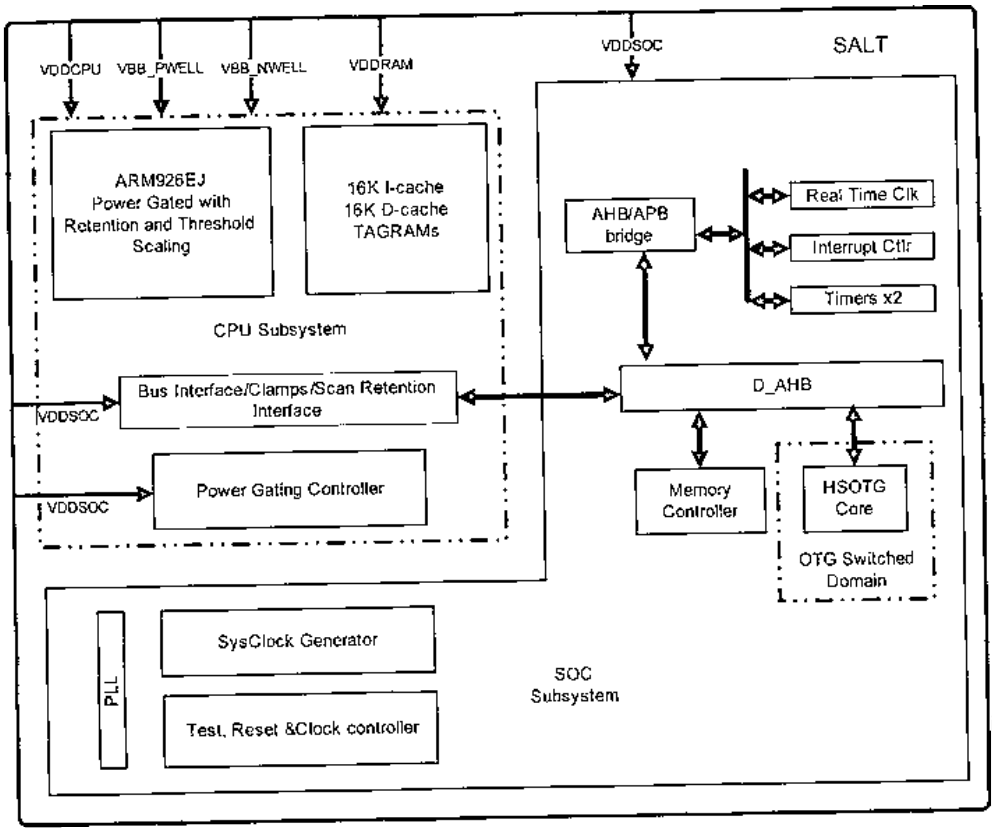


Figure 7-4 Re-Partitioned CPU Subsystem

Power Gating control and handshakes

The CPU power-gating control system manages:

- the interfaces to the external power supplies for hibernate and shutdown modes
- the bus-clock synchronous scan clock pulses for hibernation-mode save and restore.
- local header-switch power gating control for snooze mode
- isolation and state retention for snooze mode
- the handshake with the system clock generator to switch CPU clock frequencies for the different run modes

One of the goals of the SALT project is to develop a methodology that supports a wide range of library components and current in-rush management techniques. To do this, the CPU power controller uses a request-acknowledge handshake for every control signal: power gating, isolation, save, restore and reset.

AMD EX1021
AMD v. Realtek - Page 295

All the acknowledge signals are treated as asynchronous to the controller clock, and have local synchronizers to the state machine clock domain.

This approach ensures that the design is free of locally coded delays or counts. It also allows the acknowledge signals to be tied directly to the requests for some implementations or built as true handshakes. In the current SALT implementation, the acknowledges for isolation, save and restore, and reset, are all tied to their respective requests; as implemented, these functions are all very fast and require no delay in sequencing.

The power gating control of the header switches uses a true acknowledge. The power down request is daisy chained through the header switches, so that the acknowledge that arrives back at the controller has a delay equivalent to the switching time of the whole fabric.

A couple of additional design notes:

- After asserting reset, the initialization (power on reset) sequence pulses the save/restore signaling to flush out any X-s from the shadow retention flops. This may be useful when running functional test programs or vectors on the gate level netlist.
- All timing-dependent state machine transitions include a holding term that waits for the output asserted in that state to be acknowledged in order to maintain the timing-independent request/acknowledge sequencing
- The power-gating assertion and de-assertion use a request-acknowledge handshake.
- In the SALT project there are extra diagnostic control inputs to control the switch fabric which allow the power gating to be soft-sequenced or forced fully on and off, and only the power-gating acknowledge input to the state machine is used to determine when power is safely restored.

7.3 Isolation

The SALT project uses several different isolation techniques. The initial version of SALT was done before there was tool support for automatically inserting isolation cells. As a result, we manually inserted these cells in the RTL.

The interface between the CPU and cache is particularly timing critical, requiring careful design and timing analysis. We put isolation cells on all of the outputs from the CPU to the cache, but placed them in the cache's VDDRAM region. This was convenient in the SALT design because the cache is always powered up, making power routing to the isolation cells simpler.

AMD EX1021
AMD v. Realtek - Page 296

The VDDRAM region has input isolation cells instantiated as "Generic Library Cells." These are just RTL wrappers that can be mapped either to behavioral simulation models or to technology-specific clamp cells from the "Power Management Kit" for the standard cell library. This approach provides explicit instantiation of the cells on the many critical path signals, including clocks and resets, into the memories from the CPU core logic. Instantiating these cells in the RTL provides clean visibility of the clocks and critical signals that need to be managed carefully in the implementation flow. Clock balancing across isolation cells is not straightforward; isolation cells typically limit the flexibility the clock buffering tools in restructuring the buffer trees.

The outputs from the CPU to the Bus Interface Unit are isolated by cells placed in the CPU – that is, in the VDDCPU region. These output isolation cells pull-down all output signals at the AMBA bus interface in Snooze mode (local power gating), to guarantee clean SOC interface signals.

However when the CPU rail is switched off (Hibernate mode) these isolation cells lose their VDDCPU power and the outputs could again float. To address this problem, simple bus repeater or hold cells were added in the VDDSOC domain to clamp these outputs while avoiding any further gate delays. In addition, the isolate control signal from the power controller drives reset in the bus interface module to force logic-0 clamping of all bus interface protocol signals.

The power controller generates a single isolate control signal, but it is routed as two separate signals. On copy goes to the cache RAM without going through the CPU region, and one copy goes to the CPU without going through the cache. When VDDCPU has its power rail shut down during hibernate, the isolate signal for the cache is still powered up since all the buffers in its path are powered by VDDSOC or VDDRAM. Thus, the inputs of the cache are protected from floating signals (and data corruption) during hibernate.

The USB OTG block uses the alternative of instantiated AND gate cells in the RTL with suitable "don't-touch" attributes added to prevent logical optimization across these isolation boundaries. During synthesis we force these AND gates to be mapped to cells that do not crowbar with one floating input (as long as the other input is low).

All of the isolation cells in the SALT design were instantiated using a generic, technology-independent wrapper module. For hand instantiated cells, this provides a degree of design portability and enables simulation before a specific technology library has been selected.

The initial version of the SALT chip instantiates isolation cells in the RTL as described above. In a subsequent version, we use EDA tools to implement the isolation cells without modifying the RTL. The re-partitioning of the CPU subsystem to include VDDSOC also eliminates the need for bus repeater cells for the AMBA bus

AMD EX1021
AMD v. Realtek - Page 297

interface signals. Now the isolation cells get power from VDDSOC, which is always on. Thus, a single set of isolation cells is used to isolate the bus interface signals.

With the UPF-defined simulation semantics, it is possible to instantiate and simulate isolation cells by issuing a UPF tcl command (set_isolation); no hand instantiation or generic wrappers are required.

7.4 Retention

The SALT project incorporated several retention techniques to allow comparison and analysis of the area/time/energy cost functions for each technique on the same silicon.

The CPU uses full state retention. Given a fully validated CPU core, retaining every register bit state is the only safe approach to ensuring that the processor can be restarted with arbitrary control and data state. To retain only the architectural state, and use reset signal for non-architectural state, would require a serious verification project.

A key concern of any power gating design is to assure that retention registers do not get corrupted during the power down/power up sequence. On SALT we added a non-real-time diagnostic mechanism to the power control sequencing. It uses the "Hibernate" scan functionality to check-sum and save to external memory the contents of the all the shadow registers after the SAVE operation and then checksum and save the entire register contents after the RESTORE operation. These can then be compared to detect any corruption of the retention registers during power sequencing.

This approach allows error analysis for both random and location sensitive problems and the efficacy of the soft-start power gating sequencing in limiting in-rush current. This also turned out to be a valuable way of quantifying the safety margins of the retention flops and allowed them to be subjected to thermal and voltage shocks while in retention mode.

On the other hand the USB OTG core uses partial retention. The core is partitioned with persistent USB end-point data held in the Control and Status Register (CSR) block, while the Protocol Interface Engine (PIE) block contains only protocol state. Data for the current transaction is held in a RAM-based FIFO. During power down, the USB waits for the current transaction to finish, so the FIFO is empty. It then saves all the state of the CSR, using the standard retention register technique. On power up, the CSR has its state restored, and a reset is issued to the PIE. This resets the protocol engine, and it is ready for the next transaction.

The USB power gating design is discussed in more detail in Chapter 8.

AMD EX1021
AMD v. Realtek - Page 298

7.5 Inferring Power Gating and Retention

In the SALT project, we use a retention register cell with a single-pin control to control save and restore in an edge-triggered manner. Retention state is captured on the falling edge of an active-low **NRETAIN** signal and restored on rising-edge of the same **NRETAIN**:

Building on the same worked example described in Chapter 5 we can modify the UPF to show this retention behavior as follows:

```
set_retention_control my_retention_strategy
    -domain my_power_domain
    -save_signal {NRETAIN negedge}
    -restore_signal  {NRTAIN posedge}
```

Or, if we are using a simulator that does not support UPF, we can modify the RTL to show retention behavior as follows:

```
`ifdef  RTL_PG_EMULATE
    reg [3:0] state_SAVE = 4'bXXXX;
    wire PWR;
    assign PWR = pwr_req & pwr_ack;
`endif

always @ (posedge clk or negedge nrst
`ifdef  RTL_PG_EMULATE
    or negedge PWR or negedge NRETAIN
`endif

) begin
`ifdef  RTL_PG_EMULATE
    if (!PWR)
        current_state  <= 4'bXXXX;
    else if (!NRETAIN)
        current_state <= state_SAVE;
    else
`endif

   if (!nrst)
       current_state  <= 4'b0101;
   else
       current_state <= next_state;
end
```

```
`ifdef  RTL_PG_EMULATE
always @ (negedge NRETAIN)
    state_SAVE <= current_state;
`endif
```

7.6 Measurements and Analysis

We evaluated the SALT project silicon to understand the power gating and state retention improvements to leakage power, and the effect of the switching fabric on the functional performance.

Figure 7-5 shows the basic evaluation of the dynamic and leakage power as measured for both the cache memories and the standard cell logic for the ARM926EJ CPU subsystem stepping the power supply in 10% steps from 110% of nominal down to 70% of nominal. For this 90nm Generic process technology the nominal supply voltage is 1.0V.

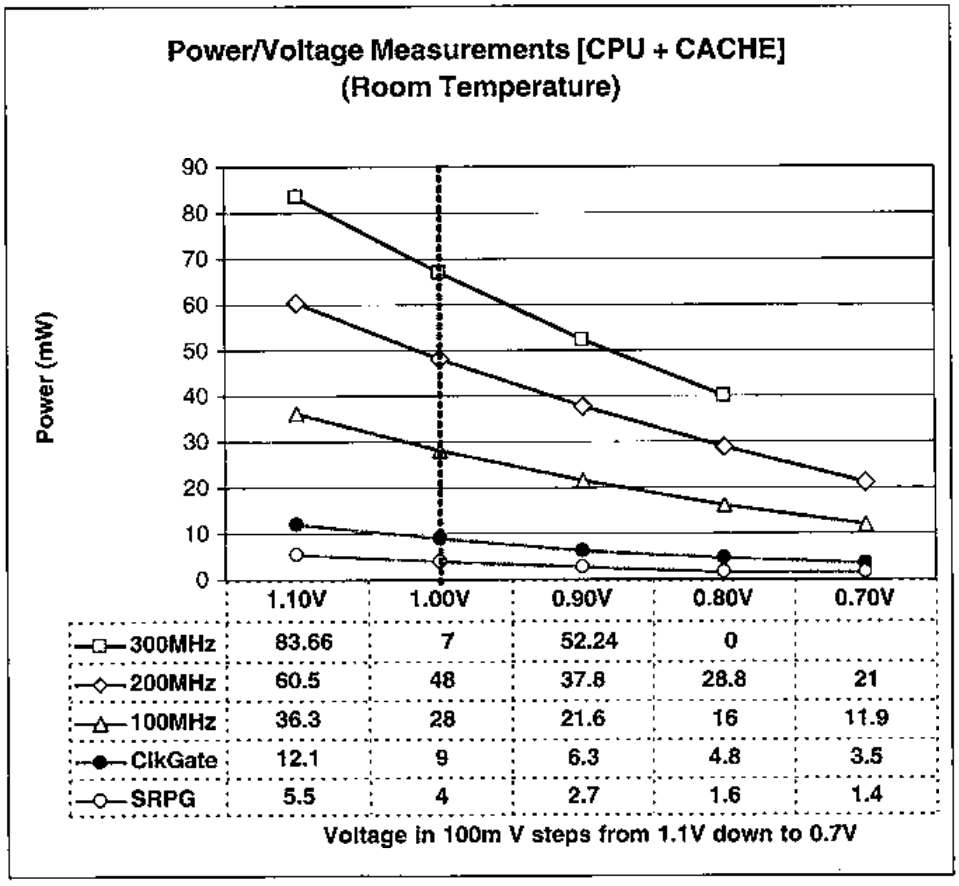


Figure 7-5 Power Measurements for SALT

The measurements are at the circuit board level – before any package/bond-wire/on-chip IR drop and the power gating fabric itself – so are very reasonable for "typical" silicon at room temperature.

The worst-case timing sign-off target frequency for the CPU design was 300MHz. The first three measurements show the dynamic power with the CPU operating at 300MHz, 200MHz, and 100MHz respectively.

The "ClkGate" measurements reflect the baseline leakage when the clock for the entire CPU is gated off.

The "SRPG" (Save Restore Power Gating) measurements are the leakage during power gating. Thus it reflects the leakage of the switches, the retention latches and the always-on control buffer trees, and in this case also includes the baseline leakage for the cache memories that are not power-gated.

AMD EX1021
AMD v. Realtek - Page 301

Figure 7-6 shows the measured leakage power for the power-gated "VDDCPU" power domain across the temperature range of key interest for battery-powered products. The vertical scale is plotted logarithmically:

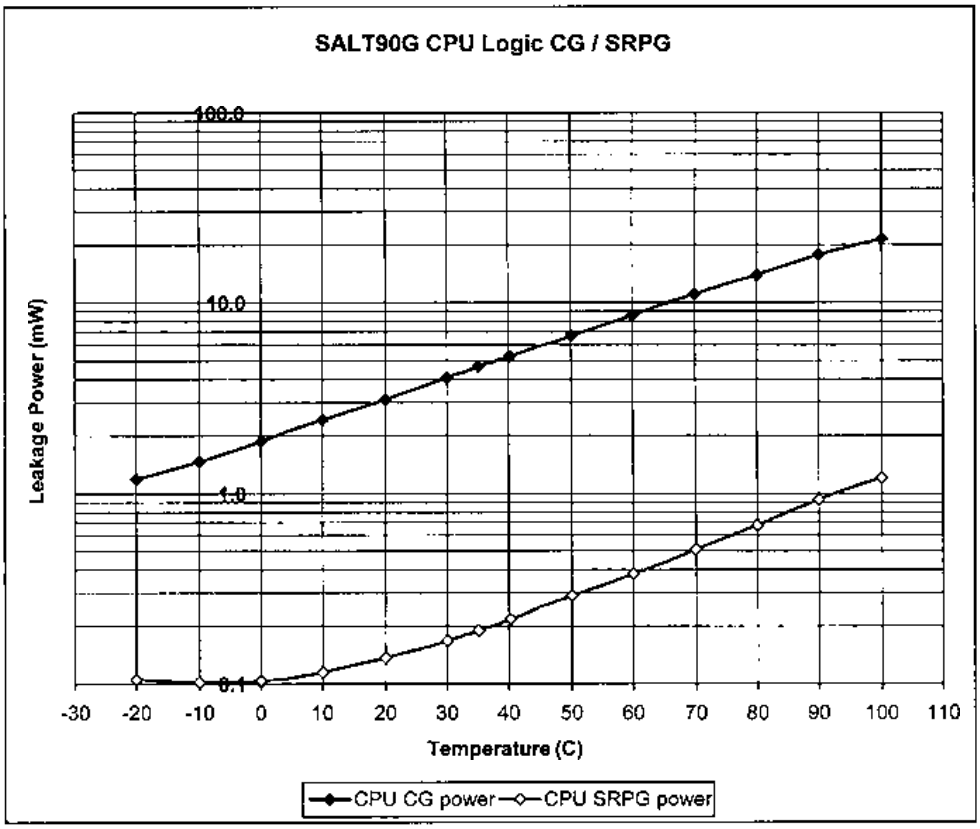


Figure 7-6 Leakage Power on SALT

The upper curve plotted is the baseline leakage when the clocks are stopped. The lower curve is the measured leakage power with State-Retention Power-Gating. Across the entire temperature range the leakage power savings are more than 10-fold and greater than 25-fold around room temperature.

Figure 7-7 shows the measured leakage power for not only the power gated standard cell logic but also the cache memories, which were not available with power-gating for this project:

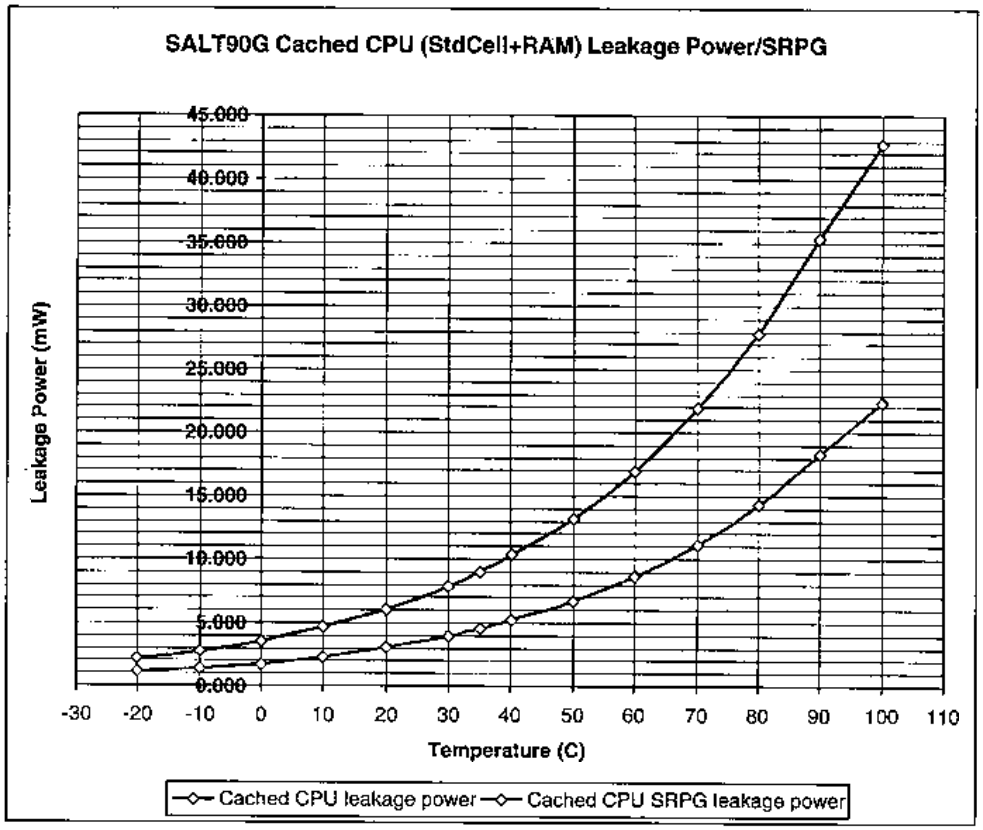


Figure 7-7 Leakage vs. Temperature

The upper curve is the baseline total leakage for the CPU subsystem (logic plus RAMs) and the lower curve shows the power with the logic portion power gated with state retention.

The leakage power of the RAM now dominates the leakage power but there is still a factor of two leakage power saving for the CPU subsystem. With integrated power gated RAMs the savings could be improved to closer to those achieved for the logic portion.

Figure 7-8 on page 100 shows the detailed comparison of the leakage power savings of state retention relative to the baseline leakage. The ratio of baseline leakage to SRPG leakage is plotted, on a linear scale, to show how this characteristic varies across the temperature range.

AMD EX1021
AMD v. Realtek - Page 303

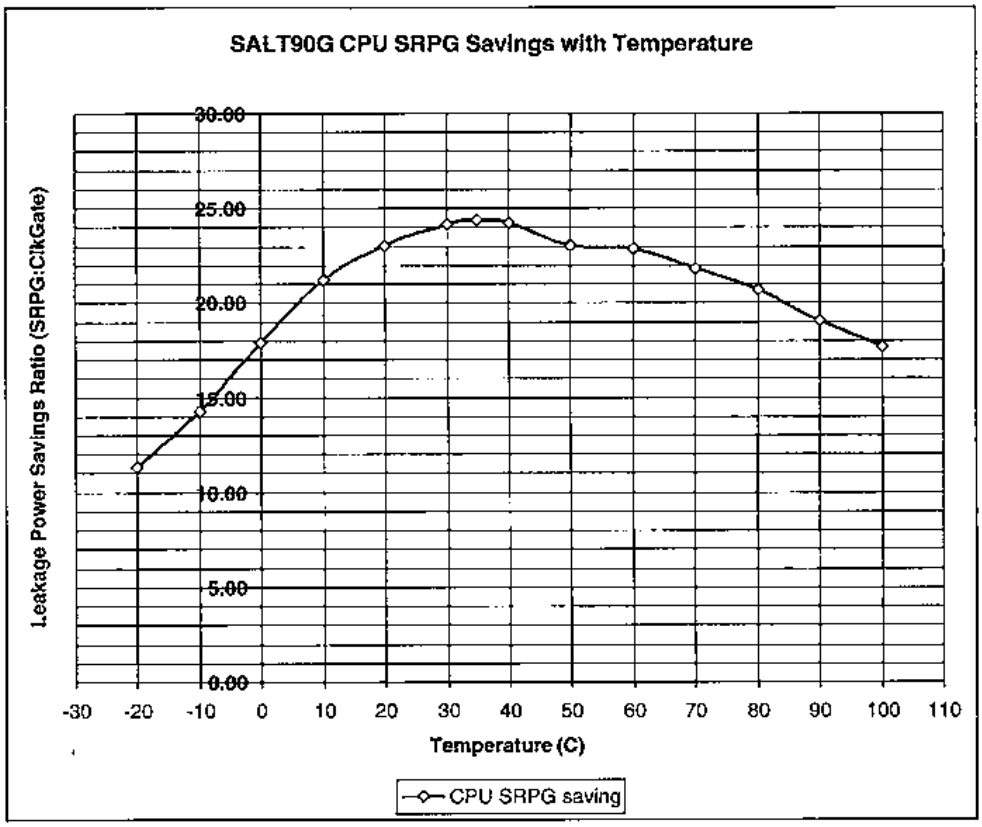


Figure 7-8 Leakage Savings vs. Temperature

The deterioration of the savings at the high end of the temperature range can be explained by the leakage behavior of the High-V_T switches. The very best power-gating leakage power saving for this particular design and technology was achieved at 35C.

In summary, power gating techniques when applied to standard-cell logic produced leakage power savings of between 10x and 25x over the baseline leakage. Because cache memories are typically tuned for performance and so exhibit a fairly high leakage power, these too benefit from leakage reduction techniques. This topic is addressed in Chapter 12.

The SALT technology demonstrator has been an effective vehicle for developing State-Retention Power-Gating techniques and methodologies and for analyzing the costs and benefits of power gating on a 90nm process. These techniques and methodologies will be highly appropriate at 65nm and below.

AMD EX1021
AMD v. Realtek - Page 304

CHAPTER 11

Implementing Multi-Voltage, Power Gated Designs

This chapter describes the implementation of designs that use power gating and multi-voltage techniques. It highlights the areas in the implementation process that are specific to multi-voltage and power gating designs.

To illustrate the implementation process, we use a multi-voltage power gated ARM1176JZF-S microprocessor as an example design.

The ARM1176JZF-S integrates a number of technologies including power gating and dynamic voltage scaling. It employs the IEM techniques described in Chapter 9; it dynamically monitors and predicts the performance requirements of multiple applications, and tunes the processor's operating voltage and frequency to match the requirements. These techniques reduce the processor's energy consumption by 25% - 50%. In order to exploit this IEM technology, the ARM1176JZF-S processor has been architected for a low power implementation.

The ARM1176JZF-S is illustrated in Figure 11-1 on page 156 and consists of a cache sub-system, central core CPU, memory management sub-system and AXI interface.

AMD EX1021
AMD v. Realtek - Page 305

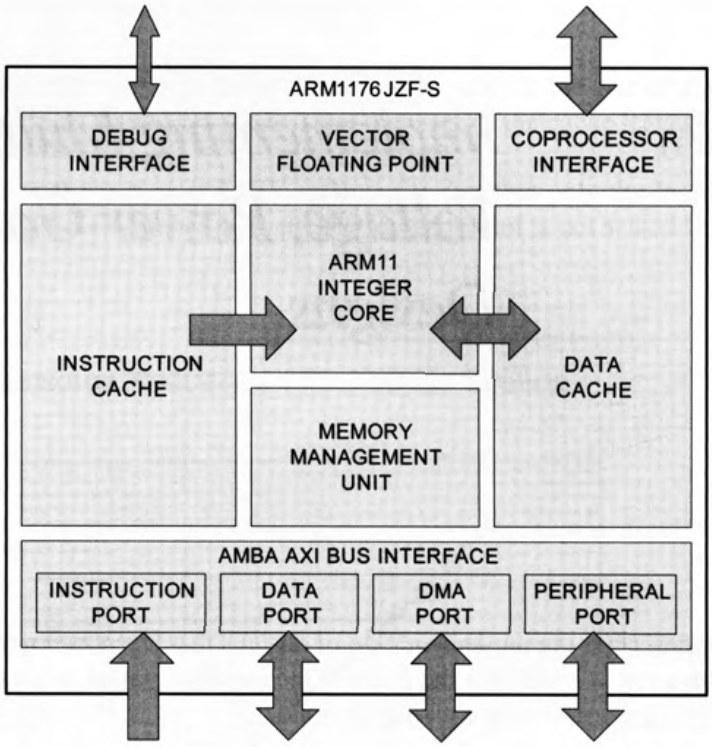


Figure 11-1 ARM1176JZF-S Synthesizable Applications Processor

There are three power domains within the ARM1176JZF-S

- An always-on power domain (VSOC) containing the logic that interfaces to the SoC through the AXI interface. The logic in this power domain also manages the asynchronous mode of the processor when performance scaling is employed.
- A shutdown power domain (VCPU) that contains all of the core CPU logic. This power domain can operate at multiple voltages and can also be powered down.
- An always-on power domain (VRAM) that contains the cache memory instances. Typically, memory cannot be scaled to the same degree as standard cells and during voltage scaling the VRAM and VCPU power domains need to be scaled independently. Also, when VCPU is shutdown, the VRAM power domain can be placed into retention. However, as discussed earlier in this book, if the cache is placed into retention state during sleep then the wake-up from sleep will take longer since the time for the power supply to the memories to stabilize will need to be factored into the wake-up time.

Figure 11-2 below shows the basic structure of the ARM1176JZF-S from a power domain perspective.

AMD EX1021
AMD v. Realtek - Page 306

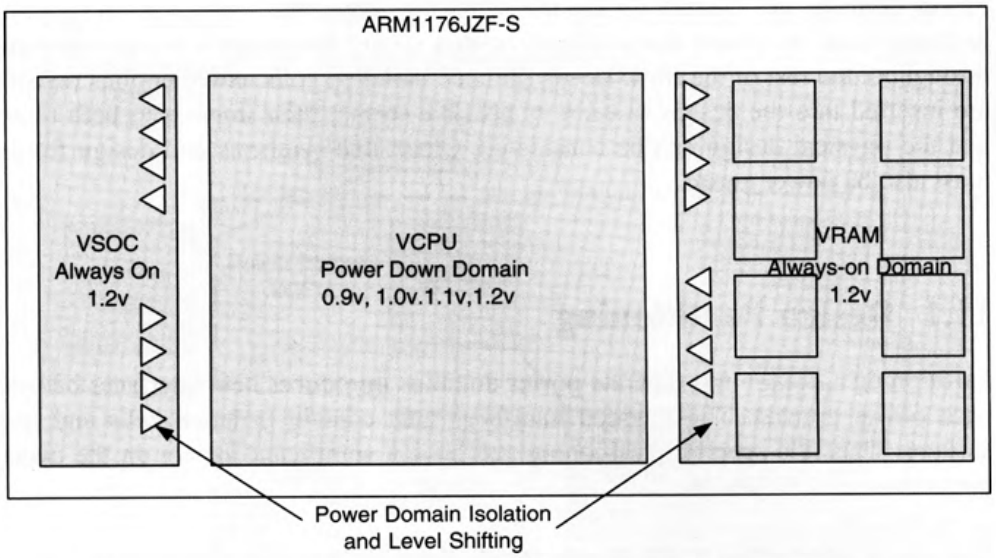


Figure 11-2 ARM1176JZF-S Multi-voltage, power gated design

VSOC always operates at full voltage – the same voltage as the logic sitting on the AXI bus externally to the processor. The VCPU power domain can operate at a number of voltages and can be shutdown. VRAM operates at the same voltage as VSOC or is placed into retention if VCPU is shutdown. Placing a memory into its retention state typically means stopping the clock(s), disabling the memory and reducing its supply voltage to the retention voltage.

To implement this design, we use an implementation process that allows for the logical and physical partitioning of multiple power domains with multiple voltage levels. The implementation process also supports the insertion of level shifters, isolation cells, and retention flops. We will also provide power switching networks for the VCPU power domain. As we proceed with the implementation, we pay particular attention to management of voltage drop, impacts on timing from our multi-voltage partitioning and the electrical integrity of our design, particularly at the power domain interfaces.

For the discussion in the rest of this chapter, we introduce two definitions:

A *Power Domain* is a logical entity and is a collection of design elements that share a primary power supply.

A *Voltage Area* is a physical entity that represents a geographic area of the chip that is used to place the logic within a given power domain. It is possible for a power domain to have a number of disjoint voltage areas depending on the nature of the chip. However, in the discussion that follows, it is assumed that there is a one to one relationship between a voltage area and a power domain.

AMD EX1021
AMD v. Realtek - Page 307

Power domains are created during the synthesis phase; the voltage areas (physical realizations of the power domains) are created during design planning and managed throughout the rest of the flow. Levels shifters, isolation cells and retentions registers are inserted into the design as early as possible so that their impacts to both timing and the physical design can be considered. Clock tree synthesis and design for test must also be power aware.

11.1 Design Partitioning

Partitioning a design into separate power domains introduces new interfaces between each power domain. These added interfaces may contain isolation cells and level shifters. Thus, the specific partitioning can have a significant impact on the overall performance of the design.

The job of partitioning a design into power domains is a joint responsibility of the system architect(s), RTL designer(s), and implementation engineer(s). Between them they understand the target application, the power and performance objectives, and the limitations of the target technology.

11.1.1 Logical and Physical Hierarchy

When mapping the logical hierarchy to power domains, we want to assign hierarchical functional units in their entirety wherever possible. Synthesis engines operate top down on a cell in the hierarchy and the best Quality of Results (QoR) is achieved if the entire hierarchy can be manipulated under the same goals and constraints.

Gathering up various hierarchical cells from different levels of the design and assigning them to power domains can make timing closure significantly more difficult.

Figure 11-3 shows the logical hierarchy of the ARM1176JZF-S after some logical grouping of top level modules in preparation for a low power implementation.

AMD EX1021
AMD v. Realtek - Page 308

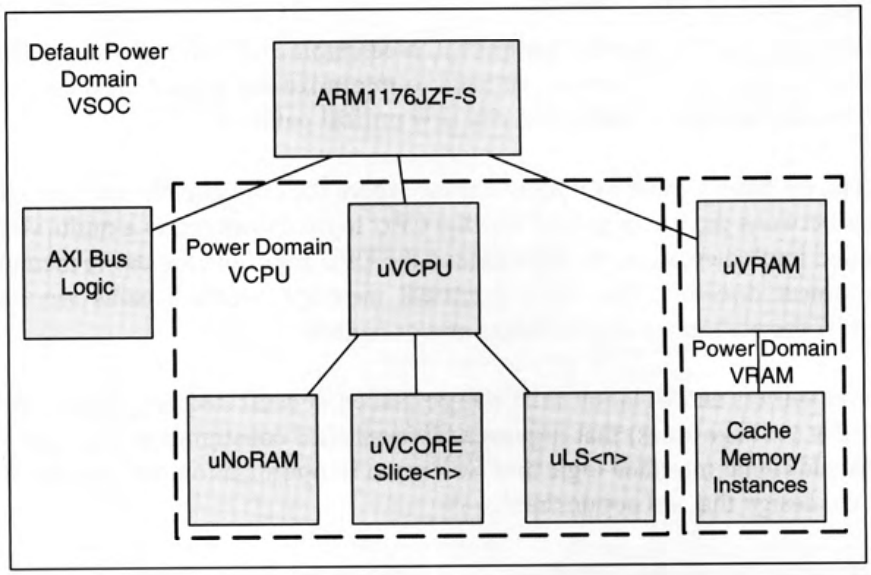


Figure 11-3 Alignment of power domain to logical hierarchy

In this example, we map the entire CPU to the VCPU domain, the whole cache – logic and RAMS – to the VRAM power domain, and the AXI bus to the VSOC power domain.

This partitioning meets several objectives:

- In some power modes, we power gate the CPU, but keep the cache powered up to enable a fast restart.
- For multi-voltage, we will operate the CPU at several different voltages, depending on work load. But in many technologies there is not enough headroom to run the cache RAMs at these lower voltages.

By maintaining a high correlation between the power domain structure and the logical hierarchy we can minimize the number of signals that cross the power domain interfaces. This minimizes the amount of logic we need to introduce to maintain electrical integrity. This in turn minimizes the QoR impact to our design and makes physical implementation easier.

For the same reasons, we have implemented the minimum number of power domains required to meet our low power objectives. Creation of unnecessary power domains makes the implementation process more complex and can negatively impact QoR.

AMD EX1021
AMD v. Realtek - Page 309

11.1.2 Critical Path Timing

In most designs, the critical paths are known, predictable and well understood. During design partitioning for low power, we need to minimize the impact on these critical paths and make sure we do not create any new critical paths.

In our case, we have a cached microprocessor, where the critical paths are through the interfaces between the cache system and the CPU logic. However, in a multi-voltage, power gated implementation, we have placed the CPU logic and the cache memory in separate power domains. Our already critical memory interface paths are further impacted by the addition of level shifters and/or isolation logic.

These level shifters and isolation cells will be placed in dedicated geographic areas in the floorplan (voltage areas) that impose additional hard constraints on any optimization. This additional interface logic and restriction in optimization may create critical paths in the design that are not desirable.

This is also true for off-chip interfaces. Any logic in the design that is operating at a voltage level different to that of the IO will need to be level shifted prior to the IO interface thereby adding additional delay to these off-chip paths.

When partitioning a design into multiple power domains we also need to avoid the situation where logic in always-on blocks is dependant on state set in powered down blocks.

Investing the time to architect power domains carefully can help minimize the impact on physical design and system timing.

Recommendations:

- Ensure that the power domain hierarchy aligns with logical hierarchy where possible
- Pay particular attention to the critical path timing in the design – especially around power domain interfaces

11.2 Design Flow Overview

As show in Figure 11-4, the design flow for a multi-voltage, power gating design follows that of a standard implementation with a few exceptions.

AMD EX1021
AMD v. Realtek - Page 310

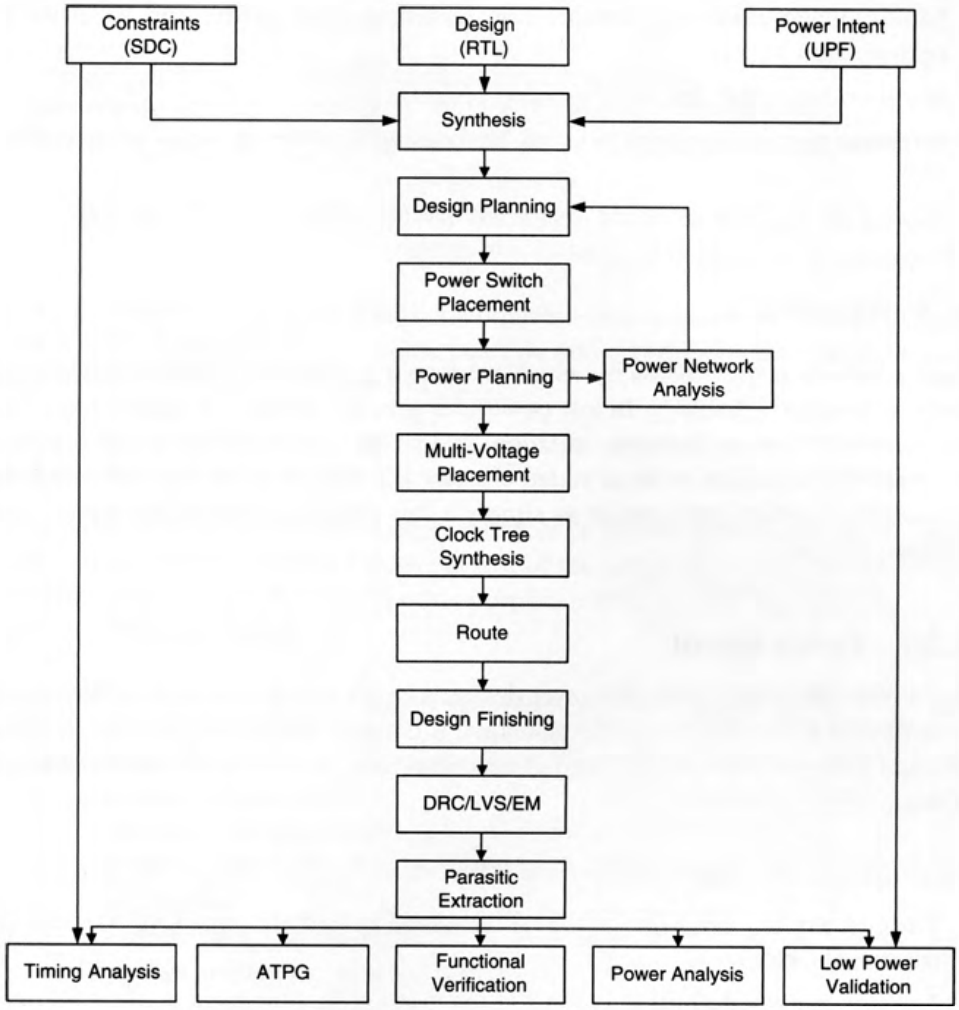


Figure 11-4 Design Flow Overview

Primarily, the differences between the standard design flow and that for a multi-voltage, power gated design are:

- Specification of the power intent
- Creation of power domains during synthesis
- State retention synthesis with always-on network management
- Multi-voltage physical design partitioning and the addition of MTCMOS switch cells
- Multi-voltage power network synthesis for MTCMOS power gating during power planning phase
- Early power network analysis to validate power gating switch topology

AMD EX1021
AMD v. Realtek - Page 311

- Multi-voltage placement optimization including level shifter and isolation cell optimization
- Multi-voltage clock tree synthesis and optimization
- Intensive post route power network analysis with power-up sequencing verification

11.3 Synthesis

Logic synthesis is the process by which we map a generic RTL representation of the design to a target technology. In low power designs, we include the power intent – the specification of power domains, isolation, and so on – at synthesis so that optimization includes the effects of these added circuits. We can view the logic synthesis of a multi-voltage power gated design as simply a low power overlay to the typical logic synthesis process.

11.3.1 Power Intent

This power intent describes the power domains in the design and defines how power is distributed to the various power domains. It defines where level shifters, isolation cells and retention flops are required. It describes how power is switched on and off to the various power domains.

Power intent can be captured in several ways

- Through explicit definition in the RTL by hand instantiating the level shifters, isolation cells, etc.
- Through explicit definition in the RTL by using HDL pragmas.
- Through the use of tool-specific commands.
- Through the use of a accompanying power file in a standard format such as the Unified Power Format (UPF).

For the purposes of the discussion here, we will use the UPF format.

11.3.2 Defining Power Domains and Power Connectivity

In UPF, power domains are specified with the following command.

create_power_domain *domain_name*
[**-elements** *list*]
[**-include_scope**]
[**-scope** *instance_name*]

AMD EX1021
AMD v. Realtek - Page 312

Where:

domain_name	Identifies the new power domain
-elements *list*	Indicates which elements are in the domain.
-include_scope	Indicates that the scope of the domain should be included in the domain.
-scope *inst_name*	specifies that the scope (instance) in which the power domain is created. Default is the current scope.

As discussed previously, it is strongly recommended that the power domain hierarchy align with the logical hierarchy where possible and that the elements list be a collection of hierarchical cells within the design.

11.3.3 Isolation Cell Insertion

Isolation logic must be inserted at the interfaces of power gated blocks such that the logic in these blocks is isolated from the rest of the design during shutdown. Specific isolation cells are typically used for this purpose and these cells are available in most physical IP libraries today.

In UPF, isolation cells are specified by the following two commands:

set_isolation *isolation_name*
-domain *domain_name*
<**-isolation_power_net** *net_name* | **-isolation_ground_net** *net_name* |
-isolation_power_net *net_name* **-isolation_ground_net** *net_name* |
-no_isolation>
[**-elements** *list*] [**-clamp_value** <**0** | **1** | **latch** | **Z**>]
[**-applies_to** <**inputs** | **outputs** | **both**>]

Where:

isolation_name	specifies the isolation strategy name.
domain_name	is the name of the domain to which the strategy is to be applied.
-elements	indicates which elements of the domain will have their interfaces isolated.
net_name	identifies the supply net(s) used to supply the isolation logic inferred by this strategy.
-no_isolation	does not isolate the port, pin, or design element specified in the elements list.
clamp_value	is the value to which the input or output shall be clamped. The default is **0**.
applies_to	indicates whether the domain's input ports, output ports, or both are isolated. The default is **outputs**.

We recommend isolating all outputs from a shutdown block. These recommendations can be met by using the default value for the **applies_to** and **-elements**.

AMD EX1021
AMD v. Realtek - Page 313

Having defined an isolation strategy for a shutdown block, we must now also define how the isolation is controlled. The **set_isolation_control** command does this.

set_isolation_control *isolation_name*
-domain *domain_name*
-isolation_signal *signal_name*
[**-isolation_sense** <**high** | **low**>]
[**-location** <**self** | **parent** | **sibling** | **fanout** | **automatic**>]

Where:

signal_name	identifies the signal that causes the specified element to drive its clamp value.
-location	specifies where the isolation cells are placed in the logic hierarchy. The default is **automatic**.

Location determines where the level shifters will be inserted:

- Self means in the module whose output is being shifted
- Parent means in the parent of the module whose output is being shifted
- Sibling means a new module is created at the same level as the module whose output is being shifted. The shifters are put in this new module.
- Fanout means that the shifters will be place in all the destination modules of the shifted signals.
- Automatic means that the tools are free to put the shifters in the appropriate location.

There can be multiple styles of isolation cell each with specific requirements on where they are placed and how they are power routed. Thus, we recommend that the location of these cells be determined automatically by the synthesis tool based on the nature of the isolation cell being targeted. The default value of **automatic** for the **–location** switch accomplishes this.

11.3.4 Retention Register Insertion

Retention registers are synthesized automatically during the logic synthesis phase once the retention style and associated attributes have been defined.

AMD EX1021
AMD v. Realtek - Page 314

In UPF, the state retention strategy and the specification of state retention control are separated and are specified by the following commands.

set_retention *retention_name*
-domain *domain_name*
<**-retention_power_net** *net_name* | **-retention_ground_net** *net_name* |
-retention_power_net *net_name* **-retention_ground_net** *net_name*>
[**-elements** *list*]

Where:

retention_name	specifies the retention strategy name
domain_name	specifies the domain for which this strategy is applied.
net_name	identifies the supply net(s) used to supply "always-on" power to the retention registers inferred by this strategy.
elements *list*	indicates which objects in the power domain.

Selecting a subset of elements in a design which shall have their state retained during shutdown is an architectural decision and should not be made during the implementation phase. Implementing partial state retention in a design must be a carefully managed process to ensure that the design can come out of a sleep mode correctly. The interaction and protocols between state retention functionality and the system reset controls must be well defined and honored during wakeup.

Therefore, the list of elements provided to the **set_retention** command is an integral part of the design architecture specification and should be identified prior to the implementation phase.

Having defined the retention strategy it is now necessary to define the retention control network.

set_retention_control *retention_name*
-domain *domain_name*
-save_signal {{*net_name* <**high** | **low** | **posedge** | **negedge**>}}
-restore_signal {{*net_name* <**high** | **low** | **posedge** | **negedge**>}}

Where:

save_net	identifies the signal that causes the register values to be saved into the shadow registers.
restore_net	identifies the signal that causes the register values to be restored from the shadow registers.

The specific values for the options in this command are again related to the architecture of the design and the specific retention registers available in the library.

AMD EX1021
AMD v. Realtek - Page 315

Note: The set_retention_control command also has optional arguments for defining assertions about the save and restore signals to assist in verification. See the full UPF specification of details on these arguments.

11.3.5 Level Shifter Insertion

Level shifters are inserted automatically during synthesis once the location and usage rules have been defined.

In UPF, level shifters are specified by the following command:

set_level_shifter *level_shifter_name*
-domain *domain_name*
[**-elements** *list*]
[**applies_to <inputs | outputs | both>**]
[**-threshold** *value*]
[**-rule <low_to_high | high_to_low | both>**]
[**-location <self | parent | sibling | fanout | automatic>**]
[**-no_shift**]

Where:

level_shifter_name	is the name of the level shifter strategy, which is used by the tools for reporting.
domain_name	is the name of the domain to which the strategy is to be applied.
elements	indicates which elements of the domain will have their interfaces shifted. Default is all the interfaces.
applies_to	indicates whether to shift the (elements of the) domain inputs, outputs, or both. The default is both.
threshold	defines the voltage, in Volts, for determining when to insert level shifters. If the difference between two domains is greater than the threshold, level shifters are inserted. The default is 0V.
rule	determines whether to insert level shifters for interfaces that go from a lower voltage to a higher one, a higher one to a lower one, or both. Default is both.

AMD EX1021
AMD v. Realtek - Page 316

We strongly recommended that level shifters be used on all power domain interfaces where an up shift is required, since this prevents crowbar currents and improves edge rates and therefore timing.

In general, we recommend that level shifters be used on all power domain interfaces where a down shift is required. It is safe to overdrive the input in the lower voltage domain. However, the timing characteristics of the destination cell will assume a correctly driven input signal operating at the same voltages the output driver of the cell. There will be an error in calculating the delay if a level shifter is not used.

Thus, we recommend **-rule both.**

The placement of the level shifters is important. The output driver for a level shifter requires more supply current than the input stage. For this reason, we recommend placing the level shifter in the destination domain – the domain that the level shifter output drives. This assures a high quality power connection to the output stage of the level shifter cell.

There are two ways to do this with the **set_level_shifter** command. We can say:

applies_to inputs
-location self

Or we can say:

applies_to outputs
-location fanout

There is no general, ironclad rule to prescribe the difference in voltage levels above which level shifters are required. This decision is technology and library dependant.

In certain cases, when the voltage difference between the two power domains is less than the threshold voltage then level shifters are not strictly required. However, when making this decision, the tolerance of the power supplies should be considered. It may be the case that when both power domains are being powered by ideal supplies that the voltage difference is tolerable, however when worst case variation between the power supplies is considered, the difference may be too great and level shifting required.

Using the default **-threshold** of 0V is a safe initial value. If timing across critical interfaces becomes a problem, we can revisit this issue and specify a different value.

In any case, we do need to specify unambiguously which power domains are at what voltage level, so the tools know where to insert level shifters. The UPF power state

AMD EX1021
AMD v. Realtek - Page 317

table constructs (**create_pst** and **add_pst_state**) provide an explicit mechanism to specify voltage levels.

In the diagram below, a net is entering the VSOC power domain which is at a different voltage from the cell driving this net. In this situation, the net must not be buffered (in the SOC domain) prior to the level shifter. Otherwise we defeat the whole purpose of using a level shifter. The requirement to avoid buffering of level shifter input effectively forces the level shifter cells to be placed close to a power domain boundary. Modern tools address these issues automatically.

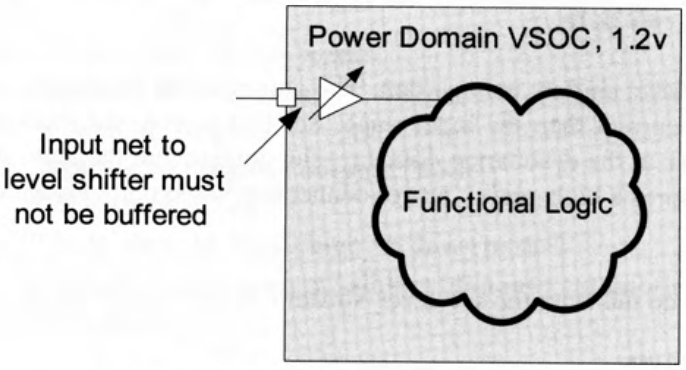


Figure 11-5 Input Nets on Level Shifters are Protected

Recommendations:

- Define a comprehensive power intent for the design that can be implemented and verified
- Define the power domains early and manage the power intent throughout the entire design process
- Define isolation logic and state retention as part of the design description (in the RTL or UPF file) to ensure full design verification
- Include isolation and state retention synthesis in the overall logic synthesis step – not as an afterthought. The area and delay cost of isolation and state retention should be an integral part of the overall cost function for the design.
- Ensure that all control networks for always-on logic (switch cells and retention registers) are buffered using always-on buffers to ensure that they remain alive during power down.

11.3.6 Scan Synthesis

As described in Chapter 5, low power designs pose special problems for scan testing.

The typical DFT challenges faced in multi-voltage design include:

AMD EX1021
AMD v. Realtek - Page 318

- Power aware architecture and reordering of scan chains
- Automatic insertion of level shifters and isolation logic on DFT signaling across power domains
- Timing issues in scan chains that cross power domains operating at multiple voltage levels
- Intelligent routing of the scan enable signal(s) to minimize power domain crossing

Figure 11-6 shows the situation we want to avoid.

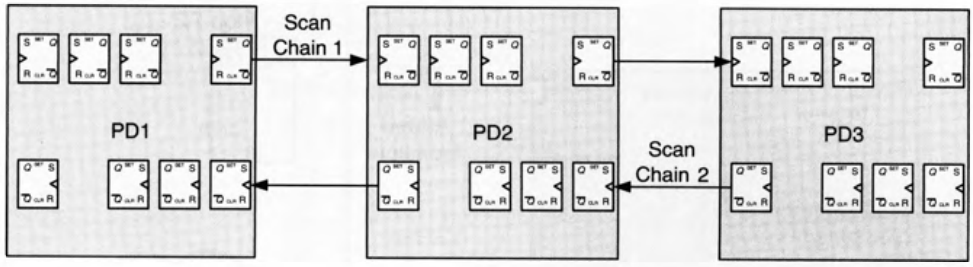


Figure 11-6 Power Domain Mixing of Scan Chains

This situation, where scan chains cross power domains, can cause real problems. At the least, level shifters and isolation cells may have to be inserted.

In the worst case, keeping all three power domains powered up during test is not possible. Since switching activity in scan mode can be much higher than in normal operation, scan testing with all domains powered up may exceed the max power for the chip.

For these reasons, we need to make every effort to implement the scan chain structure in a power aware style, where the scan paths are localized within each power domain. The scan elements can then be reordered within the power domain during placement optimization to minimize wire length. We also avoid requiring the clock skew on the scan paths to be balanced across domains, which would add unnecessary complexity in clock tree synthesis.

Figure 11-7 on page 170 shows another example of the difficulty in managing scan chains across domains. Here a scan chain comes from power domain VCPU, which is a power-down domain, and connects to Scan Flop 2. Since this connection did not exist in the original design description, the path is not isolated and therefore additional isolation logic needs to be added. This may mean that we have to add more UPF commands.

AMD EX1021
AMD v. Realtek - Page 319

Recommendations:

- Try to implement a scan strategy that is power domain aware – minimize the number of times scan chain(s) cross power domain interfaces to restrict the number of isolation cells and/or level shifters.
- Scan chain reordering should also be power domain aware and where possible scan chains should be reordered within the power domain only.

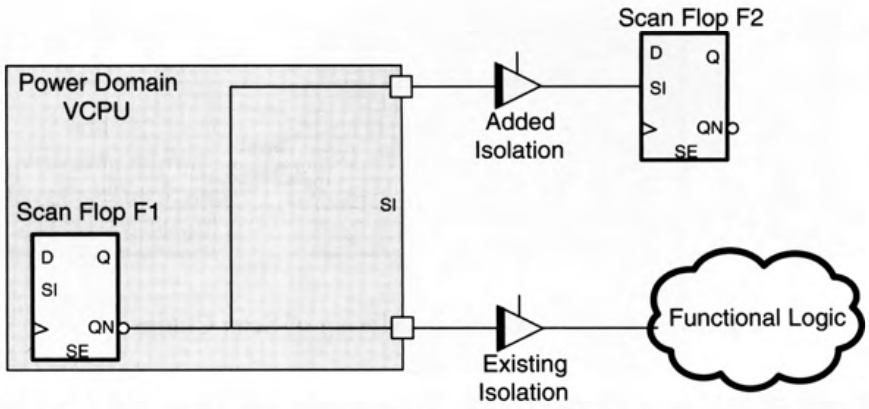


Figure 11-7 Additional Isolation Needed For Scan Chain Creation

11.3.7 Always-On Network Synthesis

When implementing any power gating design, we need to make sure that the control signals from the power gating controller remain alive during shutdown. These control signals include the controls for the power switches, isolation cells, and retention registers.

The networks of buffers that distribute these control networks are referred to as always-on networks and must use only always-on buffers and inverters. These always-on cells are very similar to their standard counterparts but are connected to the always-on supply instead of the local switched supply. This ensures that the signals on these control networks remain alive during shutdown.

In most EDA solutions available today, identification of these always-on networks is fully automated within the synthesis process. Retention registers, isolation cells and power supply switches have specific pins annotated with always-on attributes that enable the synthesis of always-on networks. Similar attributes exist on specific buffers and inverters in the library that the synthesis tool can then select to buffer these networks.

AMD EX1021
AMD v. Realtek - Page 320

A basic power gated supply with state retention register is shown in the diagram below. The control of the power gating switch (SLEEP) and the retention register control (RETN) are both buffered by always-on networks in the design. These pins are connected to the power gating controller through a network of always-on buffers and inverters that are connected to always-on power supply VDD. The local logic in the design is connected to the switched power supply VVDD delivered by the power gating switch.

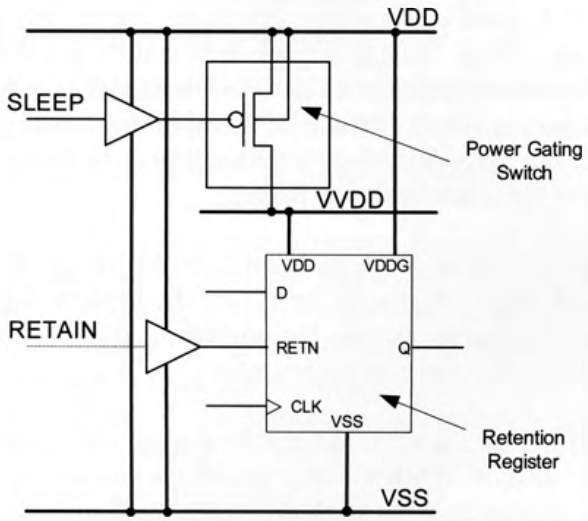


Figure 11-8 Always-on Networks Required in Power Gated Designs

Although the synthesis and physical design tools understand always-on power routing, we still need to verify the always-on nets as part of Low Power Validation at the end of the design process. The design and specification of these networks is complex; they are usually inferred from the commands that specify the isolation and retention strategies. It is easy to make a mistake in these commands but sophisticated rule-based checkers can help spot any mistakes in this area.

11.4 Multi Corner Multi Mode Optimization with Voltage Scaling Designs

It is commonplace in modern SoC designs to have a variety of functional modes. With multi-voltage design we introduce a number of additional modes to the design, including various performance levels and sleep modes, such as light sleep, hibernate and complete shutdown. These additional modes need to operate at a number of different corners (process, voltage and temperature points). The combination of these

AMD EX1021
AMD v. Realtek - Page 321

multiple modes each operating at different corners can be termed as scenarios. During the implementation process we need to both optimize and analyze the design for each of these scenarios.

For a dynamic voltage scaling design, we are concerned with ensuring full device operation at each of the various performance levels that are specified for the design. Specifically, we need to ensure that as we scale the voltage in conjunction with scaling the frequency that we have an optimal implementation at each supported combination of voltage and frequency. In a single voltage design, our worst case corner (for setup) is usually at the lowest voltage, highest temperature and worst case process. However, as we scale the performance levels in our design we may find that our worst case corner is some intermediate combination of voltage and frequency that does not reside at the edge of our operating window. Identification of this corner is important and an integral part of the implementation process.

This process is managed in the EDA tools through the use of multi-corner, multi-mode analysis. The designer can specify the modes of operation of the design and the corners at which the design is to operate. The implementation tools use these scenario specifications as constraint corners for optimization.

For the ARM1176JZF-S we intend to operate the design with varying voltage levels applied to the VCPU domain. When we vary the voltage at which VCPU operates, we also vary the clock frequency. Each of these modes, each of these combinations of voltage and frequency, has a number of associated of timing corners representing some variation in voltage, temperature and process. In addition, each of these modes will have a set of associated timing constraints, unique to each corner, which the implementation of the design must satisfy.

In using multi-corner, multi-mode techniques we need to get the best balance between quality of results and the turn time for each pass of implementation. Specifying dozens of operating modes and corners will likely result in enormous run times for the implementation tools and will not necessarily yield the best quality of results. Specifying a few well considered modes and corners for implementation followed by a more comprehensive set of modes and corners for signoff is typically the best approach.

Each defined mode needs to be accompanied with a comprehensive set of constraints (SDC) that are used during both the implementation of the design as well as the signoff.

We specify the constraints for the restricted set of scenarios at synthesis and carry them through the entire implementation flow. We then use the constraints for the full set of scenarios for signoff static timing analysis.

AMD EX1021
AMD v. Realtek - Page 322

11.5 Design Planning

The design planning process is critical to the implementation of a low power design. It is during design planning that we are presented with the majority of challenges and have the greatest opportunity to adversely affect the QoR of the design if we are not careful. Implementing a multi-voltage power gated design is going to impact the resulting QoR. Degradation of QoR can be caused by a number of factors including:

- Reduced operating voltage for certain parts of the logic
- Added delay introduced by level shifters and isolation cells
- Increased IR drop across switch networks in the power mesh
- Increased congestion caused by the use of switch cells and retention flops
- Placement restrictions due to physically bounded voltage areas

The physical design phase of the implementation starts with a determination of the overall topology of the design. Typically, the design topology will be governed by many factors, most of which are not necessarily low power related. The location of inputs and outputs for a design will be heavily influenced by the nature of the application into which the design is to be placed. Similarly, the location of the memory system and other blocks of hard IP within the design will dominate the topology, thereby reducing the flexibility and degrees of freedom available to the designer.

We need to overlay the power intent on the physical design process such that we do not limit the resulting QoR unnecessarily or place undue restrictions on the placement optimization process.

11.5.1 Creating Voltage Areas

During design planning, we will be physically bounding our power domain with voltage areas. This will ensure that all logic that constitutes a specific power domain is placed close together, allowing power to be delivered to that logic more efficiently.

When architecting voltage areas, we need to try to optimize the relative location of functional blocks and their communication with other blocks in the design. Where possible, we want to route power consuming busses efficiently and avoid unnecessary scenic routes around power domains. As with most design planning tasks, the creation of a voltage area from a power domain will be an iterative process and will involve identifying the best size and shape of the voltage area and how best to fit that voltage area in the chip.

We determine the size and shape of the voltage area based on the performance requirements of the logic and how densely it can be packed. Also, we consider the type of logic that is to be placed - does the voltage area contain a number of hard IP blocks that would present a significant challenge to the router if placed in the middle

AMD EX1021
AMD v. Realtek - Page 323

of the design? Would using rectilinear voltage areas with these types of power domain help in the floorplanning process? We also need to make sure that voltage areas do not overlap.

Since voltage areas by definition contain logic that employ differing power strategies, care must be taken at the interface to these voltage areas to ensure that the voltage domains are truly isolated. We recommend that each voltage area be surrounded by a guard band providing some degree of physical isolation between the power domains. Guard bands present a physical placement constraint for the voltage area and they should be large enough to provide the necessary isolation but not so large that they use up valuable silicon real estate and impact the QoR. Figure 11-9 on page 174 shows how the voltage areas may be created.

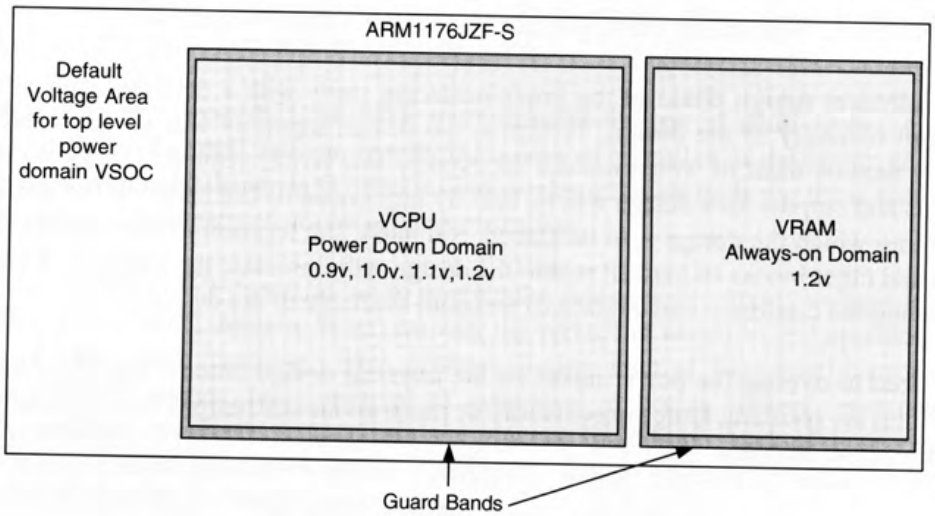


Figure 11-9 Mapping of Power Domain to Voltage Area

As mentioned in Chapter 7, nesting of power domains is a logical concept and should be avoided in the physical design. If they are absolutely required, care must be taken to ensure that the physical bounds of the nested voltage areas are separate from the voltage area of the parent power domain.

We also do not recommend splitting a power domain into multiple separate physical regions on the chip. Although this is supported in most EDA tools today, having to do this should prompt the designer to reconsider the design partitioning.

Level shifters place specific constraints on design planning. For electrical integrity, the level shifters need to sit as close to the boundary of a voltage area as is possible, as they have input signal nets that are protected from buffering. Long un-buffered nets can create timing problems in a design.

AMD EX1021
AMD v. Realtek - Page 324

When there are many signals between two power domains, the large number of level shifters required can present a challenge. For example, power domains that provide data on a number of wide data busses will naturally require large numbers of levels shifters, all of which will be pulled as close to the physical boundary of the voltage area as is possible. This can result in multiple site rows being consumed with level shifters around the periphery of a voltage area, leading to degradation in QoR for those nets that have their level shifter placed furthest from the boundary.

In these situations, it is recommended that power domains be floorplanned in such a way as to align data busses across power domain boundaries and give as large a dimension as possible to common interfaces. This will allow for spreading of the level shifters, resulting in a smaller physical separation between the logic in each power domain.

11.5.2 Power Gating Topologies

Where local power gating is employed, there are a number of alternative topologies for the placement of the power gating switches. These are discussed in detail in Chapter 14.

For most designs, including the ARM1176JZF-S, we use a grid design for the power network. With modern tools, we have two options for specifying the placement of the switches. We can specify an x and y increment, and the tool will place the switches evenly throughout the network. Or we can give the tool a design goal (such as current capability or IR drop) and the tool can do switch placement exploration and return with a list of options for the designer.

The size and placement of the switches is critical in determining the IR drop across the switch network. This in turn is critical in determining timing, since delay is very power-supply dependent.

In UPF, power gating switch cells are specified by the following command

create_power_switch *switch_name*
-domain *domain_name*
-output_supply_port {*port_name supply_net_name*}
{**-input_supply_port** {*port_name supply_net_name*}}*
{**-control_port** {*port_name net_name*}}*
{**-on_state** {*state_name input_supply_port* {*boolean_function*}}}*
[**-on_partial_state** {*state_name input_supply_port* {*boolean_function*}}]*
[**-ack_port** {*port_name net_name* [{*boolean_function*}]}]*
[**-ack_delay** {*port_name delay*}]*
[**-off_state** {*state_name* {*boolean_function*}}]*
[**-error_state** {*state_name* {*boolean_function*}}]*

AMD EX1021
AMD v. Realtek - Page 325

Where:

switch_name	specifies the name of the switch instance to create;
domain_name	specifies domain containing the switch.
-output_supply_port	identifies the output supply port of the switch and the net where this port connects.
-input_supply_port	identifies the input supply port of the switch and the net where this port connects.
-control_port	identifies a control port on the switch and the net where this control port connects.
-ack_port	identifies the acknowledge port on the switch and the signal net where this port connects.
-ack_delay	identifies the acknowledge port and delay on the switch where this port connects.
-on_state	defines what value of the control port turns the switch on.
-on_partial_state	defines what value of the control port turns puts the switch in a current-limited state.
-off_state	defines what value of the control port turns the switch off..
-error_state	identifies any error states, which if defined on the switch can be flagged during simulation or analysis.

11.5.3 In-rush Current Management

The power-up sequence for a power gated design is critical for limiting the in-rush current, so that we do not cause voltage spikes that could corrupt retention registers or data in adjacent domains.

One effective technique, employed on ARM1176JZF-S, uses a buffered switch which provides a buffered (SLEEPOUT) version of the switch control input pin (SLEEP). This allows daisy chaining of the sleep control network throughout the design by connecting the SLEEP pin of one switch to the SLEEPOUT pin of another switch.

The exact topology of this daisy chain can be user defined, but the end goal is the same – a buffered sleep network that can be turned on quickly but at the same time limiting the in-rush current. This usually entails grouping sets of switches together, and driving the whole group with the same SLEEP pin, so they all turn on at the same time. One of the SLEEPOUT signals from the group is connected to the SLEEP pin of the next group. Thus, groups of pins are daisy chained. The speed vs. noise trade-off now involves determining the size of the group. This can be done through power analysis.

11.5.4 Recommendations:

- Voltage areas should be guard banded where necessary.
- Placement optimization must be constrained to ensure that cells in a power domain are placed inside the associated voltage area.

AMD EX1021
AMD v. Realtek - Page 326

- New cells created during the physical design optimization process must be placed in the appropriate voltage area and powered by the correct power supply.
- Short feed-through paths in a voltage area should be protected from repeater insertion during design optimization and design-rule fixing to maintain wire connection.
- The isolation cells and level shifters should be placed at the voltage area boundary to ensure minimal impact to timing.
- It may be necessary to experiment with various power switch topologies and compare the results to determine which is most suitable for the design in question.
- Consider the power routing impacts for the switch topology chosen.

11.6 Power Planning

With a physical topology in place we can now begin to consider how to provide power to the various voltage areas in the design. Since by definition, each power domain employs a different power strategy, it is likely that we will be routing different power rails to each voltage area.

Minimizing the voltage drop across each of these power rails is a key part of meeting the performance goal. Unfortunately, many of the techniques that we employ in a low power design can make the voltage drop and noise problem worse. For example, when the design is operating at reduced voltage levels, our available margin for voltage drop is considerably reduced. When we change performance levels or power-up a block after shutdown, the on-chip current gradients become worse which can lead to noise injection. And in power gating blocks, there is an IR drop across the power switches.

Most EDA tools provide some level of automated power network synthesis (PNS) capability for the distribution of power across a design. We recommend highly that the implementation of a multi-voltage design utilize this type of PNS capability wherever possible. The number of concurrently changeable variables in a multi-voltage system is such that attempting to construct a power plan manually may result in a suboptimal result both in terms of routing resources consumed and the voltage drop on the rails.

Power network synthesis allows the designer to specify the absolute constraints on the power plan (such as maximum voltage drop, routing layers and via requirements). It ensures that the power budget for the design is satisfied and provides integrated voltage drop and electro-migration analysis. Most level shifters available today that are shifting from a low voltage to a higher voltage require two power supplies. Cells that combine level shifting and isolation require a secondary supply that provides always-on power to the cell. During the power planning process it is necessary to hookup the

AMD EX1021
AMD v. Realtek - Page 327

primary power pins to the appropriate voltage rails. There are various ways that this can be achieved depending on which EDA tools are being used. For example, for isolating level shifters, the primary power supply pins can be hooked up to the appropriate voltage rails through PNS and the secondary power pin (for always-on power) can be connected by the power router later in the design flow when the final placement of these cells is frozen.

The sleep transistor power network is composed of three components: a permanent power network, a virtual power network and an array of the sleep transistors. All of these components contribute to the quality of the sleep transistor power network in terms of IR drop, routing resources and silicon area. Consequently, the synthesis of the sleep transistor power network becomes a challenge. Historically, most power-gating designs used scripts to insert and place the switch cells, and create power straps and rails based on heuristic rules. Today, the sleep transistor power network can be synthesized using a power planning tool PNS [PNS] that considers all the three components.

Recommendations:

- Use a power planning tool which supports switch cells to create the sleep transistor power network including switch cell insertion and placement.
- Run both static and dynamic IR drop analysis on a design with switch cells to verify the power integrity. Significant static IR drop violations usually require increasing the number of switch cells and adjusting their positions.
- Dynamic IR drop violations are commonly fixed by decoupling capacitor insertion. Accurate power-on sequence I-V responses have to be obtained by transient analysis to assure that all IR drop violations are fixed.
- In the case of any IR drop violations, try to fix the violations in the permanent power network, because any changes in the switch cells distribution and virtual power network usually require re-analyzing wakeup latency and peak current.
- Run wakeup in-rush current analysis to check that the max in-rush current and wakeup latency meet the defined constraints. For small designs, it is possible to run this analysis in SPICE. At chip level, the wakeup analysis is done using rail analysis tools.
- Run dynamic IR drop analysis covering the wakeup period to detect any violations caused by current spikes, through common power rails, on the parts of the design that are alive while the power-gated block is waking up.

AMD EX1021
AMD v. Realtek - Page 328

11.6.1 Decoupling Capacitor Insertion

An integral part of the power planning process involves ensuring that the noise coupling between different power supply rails and between cells in the same power domain is minimized. This power and ground noise can degrade timing and possibly cause functional failures. Inserting decoupling capacitors (decap cells) in the design is the method commonly used to address these issues.

Decoupling capacitor insertion becomes more challenging in power-gating designs than in normal designs. Implementing in-rush current control techniques, such as daisy chaining the power-gating structure to sequentially charge the power-gated block, reduces the in-rush current. However, the turn-on current of each switch, particularly of those switches which are close to the head of the daisy chain and turned on first, can still generate significant power and ground noise pulses in the rest of the design, through the permanent power supply and common ground networks. Adding decoupling capacitors to the permanent power supply nets is an effective way to reduce this noise. It is most effective to insert the decoupling capacitors next to the switch cells which are the sources of the noise during wakeup.

Apart from the power and ground noise generated by the in-rush current, there is also the noise generated by cell switching currents in the power-gated block during normal operation. Decap insertion is a common method to address this noise. However, decap insertion becomes more challenging in the power-gated design. In sub-90nm technology, decap cells exhibit high leakage power that needs to be controlled. Also, the decap cells added to the virtual power network have significant impact on the wakeup latency and in-rush current. The larger the decoupling capacitance, the larger the in-rush charge current and the longer the charge time (latency) at wakeup. Therefore, the optimization of de-cap insertion in the power-gating design becomes very important to achieve maximum noise reduction with minimum added capacitance at the virtual power network. This can be done by identifying noise hot spots using dynamic IR drop analysis tools and then inserting just enough capacitance at the hot spots to reduce the noises meeting defined noise target. It is worth noting that advanced EDA tools have implemented such optimal decap insertion methods.

Recommendations:

- Add as much decoupling capacitance as permitted in the permanent power network at the positions close to the switch cells. This achieves the maximum effectiveness and minimum impact on the wakeup latency and in-rush current. It is convenient to integrate the decap into the switch cell to simplify decap insertion. The maximum capacitance of the decap on the permanent power network is constrained by the leakage and area penalties.

AMD EX1021
AMD v. Realtek - Page 329

- To fix dynamic IR drop violations in the post-layout stage, it is preferable to add decoupling capacitance to the permanent power network close to the violation spots, if the violations are related to the permanent power network. The rest of the violations have to be fixed by adding decap to the virtual power network at the violation sports.

11.7 Clock Tree Synthesis

The impact on the overall power consumption of the clock tree in a design is significant. In many cases, more than half of the overall power consumption of a design may be due to the clock network. There are a number of ways to mitigate the effects of the clock tree power, including clock gating and minimizing clock tree insertion delays.

However, when we are dealing with a multi-voltage design, we have additional restrictions on how we can manipulate the clock tree to meet both our low power requirements and our performance requirements at each performance level. In a single clock multi-voltage design, the clock is used in multiple power domains and hence crosses a number of voltage area boundaries. As the clock passes through each voltage area its latency is modified based on the voltage at which those voltage areas are operating. If the clock path buffering and data path buffering are not well balanced across voltage areas then skew management becomes very difficult.

The diagram in Figure 11-10 illustrates the issue. In this particular situation, both the data and the clock are being sent by the CPU to the memory. The data path buffering is split between the VCPU and VRAM power domains with the clock path buffering residing primarily in the VRAM power domain. Assume that we have very good skew minimization when both power domain VCPU and VRAM are operating at the same supply voltage. When the voltage on power domain VCPU is reduced as part of performance scaling then we run the risk of slowing the data path relative to the clock and introducing setup violations in power domain VRAM.

AMD EX1021
AMD v. Realtek - Page 330

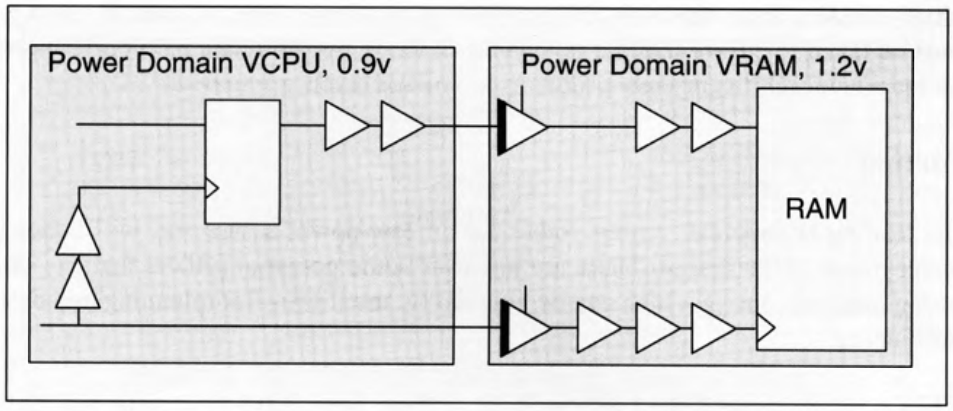


Figure 11-10 Clock Tree Synthesis Challenge in a Multi-Voltage Design

In order to help alleviate this problem, clock tree synthesis algorithms in current EDA tools are multi-voltage aware and use a bottom-up approach to constructing the clock tree. Each voltage area is processed in turn and the clock networks for each voltage area are constructed to minimize the skew. These low level clock tree structures are then merged by the construction of higher level clock trees that join together the sub-tress to form an overall clock network for the design.

Adopting this approach helps locate the buffering for the clock tree in the same voltage area as the data path buffering. This clock buffer clustering prevents the clock network from winding its way in and out of a number of voltage areas, requiring level shifters to be inserted every time the clock network crosses a voltage area boundary. This level shifter insertion process would add significant delay to the clock tree and impact the ability to minimize the clock skew and power. Figure 11-11 illustrates this clustering approach. Note that the VRAM and VPCU domains have level shifters ("LS" in the figure) in their clock trees.

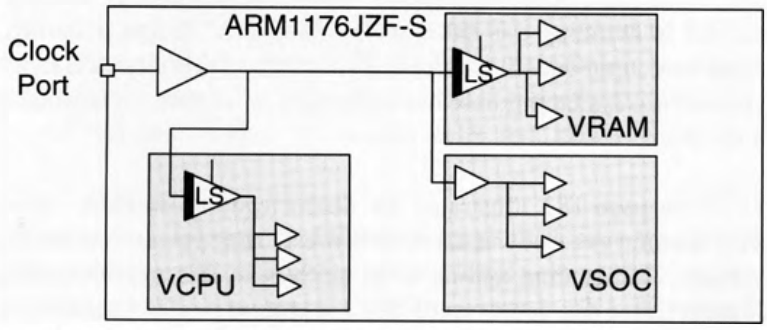


Figure 11-11 Bottom Up Clock Tree Clustering

This approach lends itself well to voltage areas that have a good mix of clock and data buffering such as blocks of standard logic. However, when we have a voltage area

AMD EX1021
AMD v. Realtek - Page 331

which contains little standard cell logic but rather many blocks of hard IP, extra care must be taken to ensure that the imbalance of clock tree and data path buffering will not introduce significant skew problems as voltage levels are scaled.

Routing

The leading commercial routers today are all low-power aware and will honor the power intent in the design. There are however some potential pitfalls that can occur during detailed routing which can be avoided by intelligent floorplanning and design partitioning.

When a design is partitioned into multiple voltage areas, hard placement and routing restrictions exist that can impact the routing of the design. This can lead to degradation of QoR. It is therefore important to consider routing of the design, especially on-chip bus structures and global control networks, when floorplanning the design.

If the best route for a net from one part of the design to another is across one or many voltage areas then a decision has to be made on how to deal with the net.

- Route the net through the voltage area(s) and then add the appropriate level shifters to each voltage area crossing
- Detour the route around the voltage area(s) to the destination

With the first option, adding level shifters to every point at which the net crosses a voltage area will be expensive in terms of area, power and delay. The second option will introduce an incremental delay compared to a direct connection that would be available in the single voltage design; however the impact of this approach is considerably less than with the first option.

Both of these options will impact the QoR and so clearly we want to avoid this situation. The cost (in time) of rectifying this situation at the detailed routing stage will be expensive and so avoiding this situation by considered design planning earlier in the design cycle is recommended. Complete avoidance of this situation in a multi-voltage design may not always be possible but making every attempt to minimize the problem is recommended.

Figure 11-12 on page 183 illustrates the detour routing problem. Here a net going from power domain A to power domain C must cross power domain B which is at a higher voltage. Rather than adding level shifters to this net and routing it through power domain C, we can detour route this net around power domain C.

AMD EX1021
AMD v. Realtek - Page 332

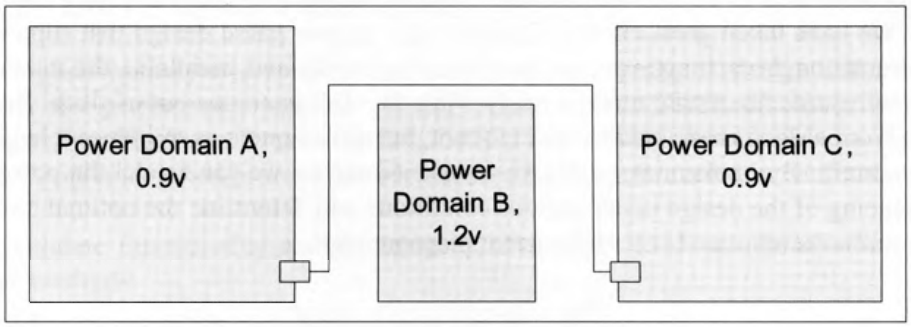


Figure 11-12 Detour Routing Around a Power Domain

11.8 Power Analysis

Having completed the implementation of our multi-voltage, power gated design, it is now necessary to verify the integrity of the power network. Two aspects of our power network are critical: the voltage drop seen by the standard cells in the power gated blocks and the profile of the in-rush current during power-up sequencing.

Performing power rail analysis of the design is critical. There will be many power rails in the design and the integrity of each needs verifying and the interaction between rails (coupling) also needs to be verified. Using a comprehensive rail analysis tool, we can perform extensive analysis of the power networks in the design. The coarse level power network analysis was completed during the power planning phase. What we are looking for here are areas in the design where we have excessive voltage drop due to local level clustering which results in power "hot" spots.

Implementation tools today are very good at performing local clustering of registers, clock gates and clock tree buffers to ensure that aggressive timing goals are met. This however can make the voltage drop problem worse in those regions of the chip, especially in regions with switched power rails. Since logic spreading in these areas is not viable (this will impact the QoR) we must deal with the voltage drop problem in a different manner and typically sizing the switch cells and power mesh is the best approach.

The rail analysis gives us valuable data on the voltage drop seen by every standard cell in the design. Where we have an unacceptably large voltage drop we can try sizing or duplicating power switch cells and sizing the power mesh to alleviate the problem. Creation of instance specific voltage drop data for a design can then be used by a signoff static timing to measure the impact of voltage drop on the timing of the design.

AMD EX1021
AMD v. Realtek - Page 333

The second area of power analysis that needs to be addressed is that of in-rush current. We have taken great effort in architecting a power gated design that supports a fast transition from functional mode to standby mode, and maintains the necessary state to resume functional mode quickly when the design comes out of sleep. Recovering from a sleep mode quickly is important, but not so quick as to induce a large in-rush current. By performing extensive transient analysis we can model the power-up sequencing of the design under various conditions and determine the optimal balance between wake-up speed and rush current management.

11.9 Timing Analysis

A multi-voltage power gated design will need to function in a number of different modes each operating at different corners. During implementation, we selected a reduced set of scenarios - operating modes and corners – and used these as constraints. Hopefully we chose the best and worst cases and produced a design that will work under all scenarios.

The purpose of signoff static timing analysis is to prove that we have, in fact, implemented a design that works under all scenarios. As such it is necessary to accurately specify all of these scenarios and provide all the necessary technology libraries required to analyze the design at the supported performance levels.

In particular, we need to make sure that the libraries – especially level shifters – are characterized for the voltages as which we run signoff static timing analysis.

Run times will now be significantly longer than for a design that is not implemented with aggressive low power techniques.

During the fabrication process for a CMOS based design, it is expected that there will be minor variations in resistance and capacitance of both the transistors and the metallization. These variations will impact the timing paths in different regions of the chip. We recommend that a small window of uncertainty be placed around each of these operating corners to provide a greater level of confidence in the final signoff results. This window of uncertainty can help counter any on-chip variation (OCV) that may exist across the die. On-chip variation will derate the path delays in unfavorable directions, such as speeding up clock and slowing down data for setup checks and visa-versa for hold checks.

AMD EX1021
AMD v. Realtek - Page 334

11.10 Low Power Validation

Having successfully implemented our multi-voltage power gated design that meets all power and performance targets, it is now necessary to validate the integrity of the design. Specifically we want to ensure that the low power intent provided at the start of the implementation process has been successfully implemented in the final design.

We validate functionality and the integrity of our low power implementation using three methods:

- Gate level logic simulation
- Equivalence checking
- Rule-based methods

Clearly gate level simulation of our final design can tell us if the design still remains functional with the low power structures. Specifically, we can validate that the design:

- resets cleanly at startup
- can be placed into various sleep modes
- behaves appropriately during shutdown
- powers-up successfully after shutdown

Formal equivalence checking tools can prove that the gate level netlist is equivalent to the original RTL plus UPF code.

A rule-base tool can tell us that the power structure of the final gate level netlist makes sense. It can validate that the isolation cells and level shifters are placed in the correct domains, and that all nets requiring isolation or level shifting have the appropriate cells in place. It can check that cells that required always on power – such as retention registers, isolation cells, and buffers of power control signals – do, in fact, have the appropriate supplies. Finally, we can use these tools to find situations where there are redundant isolation cells or level shifters.

11.11 Manufacturing Test

In a typical voltage scaled system, various parts of the design will be running at reduced power supply levels. During the implementation process, the design can be optimized at these voltage levels thereby yielding an overall lower power design that meets the needs of the system application. However, in most manufacturing test situations, the design will be tested at nominal supply rail voltage levels which will cause the design to consume significantly more power during test than it would during nor-

AMD EX1021
AMD v. Realtek - Page 335

mal operation. If the design has only been packaged assuming functional power dissipation numbers then thermal failure of both the design and package at test can result. In addition, this increased power consumption in the core can lead to significant voltage drop in the power distribution network which in turn leads to functional failure. It is therefore necessary to consider the requirements of both the functional and test modes of the design when implementing voltage scaling.

When performing manufacturing test on designs implemented in very small geometries (90/65/45 nm) it is necessary to use a variety of test models and test strategies. Specifically, testing the design with stuck-at faults models now needs to be complimented with transition fault delay testing. Delay fault testing is particularly susceptible to changes in the voltage and frequency at which a design operates. Similarly, path based testing is also dependant upon the voltage at which the design is being operated. As will be discussed later in this chapter, the critical path of a voltage scaled design is not the same path at each voltage level at which the design operates. Each performance level supported by the design may well have a unique critical path. Each of these paths needs to be tested during at-speed test in a manufacturing test environment.

However, as we reduce the supply voltages at which we test our design, delay behavior changes dramatically. In particular, weak circuit elements and defective circuit elements do not behave in the same way. In addition, the tester environment differs from the system environment in noise content, heat dissipation and pin load. As a result, care needs to be taken to distinguish between correct behavior and defective behavior. In many cases, final decisions on correct timing and voltage levels will need to wait until silicon is available.

Automating the manufacturing test process will require performing stuck-at fault testing and delay fault testing at several operating voltages, and also may involve both nominal and high temperature test. Developing the appropriate mix of tests in order to guarantee sufficient quality at reasonable cost is a complex problem. Different tradeoffs between test time and coverage will apply to different products.

AMD EX1021
AMD v. Realtek - Page 336

APPENDIX B *UPF Command Syntax*

This chapter describes the syntax for selected UPF commands referenced in the text.

Excerpts from the "Unified Power Format(UPF) Standard, Version 1.0" reproduced by permission of Accellera. Copyright(c) 2006-2007 by Accellera. Accellera does not warrant or represent the accuracy or content of the excerpted material, and expressly disclaims any express or implied warranty. Accellera Standards excerpts are supplied "AS IS."

The full standards document in its entirety can be found under www.accellera.org.

AMD EX1021
AMD v. Realtek - Page 337

B.1 add_pst_state

Purpose	Define the states of each of the supply nets for one possible state of the design	
Syntax	**add_pst_state** *state_name* **-pst** *table_name* **-state** *supply_states*	
Arguments	*state_name*	The power state.
	-pst *table_name*	The power state table (PST) to which this state applies.
	-state *supply_states*	The list of supply net state names , listed in the corresponding order of the **-supplies** listing in the **create_pst** command.
Return value	Return a `1` if successful or a `0` if not.	

The **add_pst_state** command defines the states of each of the supply nets for one possible state of the design.

It is an error if the number of *supply_state_names* is different than the number of supply nets within the PST.

Syntax example:

```
create_pst           pt -supplies { PN1   PN2   SOC/OTC/PN3 }
add_pst_state s1 -pst pt -state    { s08   s08   s08         }
add_pst_state s2 -pst pt -state    { s08   s08   off         }
add_pst_state s3 -pst pt -state    { s08   s09   off         }
```

AMD EX1021
AMD v. Realtek - Page 338

B.2 connect_supply_net

Purpose	Connect a supply net to supply ports and/or pins	
Syntax	**connect_supply_net** *net_name* [**-ports** *list*] [**-pins** *list*] [<**-cells** *list* |**-domain** *domain_name*>] [<**-rail_connection** *rail_type* | **-pg_type** *pg_type*>]* [**-vct** *vct_name*]	
Arguments	*net_name*	The name of the supply net.
	-ports *list*	A list of ports to connect.
	-pins *list*	A list of pins on cells to connect.
	-cells *list*	A list of cells to use for **-rail_connection** or **-pg_type**.
	-domain *domain_name*	The domain to use for **-rail_connection** or **-pg_type**.
	-rail_connection *rail_type*	The rail type (for older libraries).
	-pg_type *pg_type*	The power/ground pin type.
	-vct *vct_name*	A VCT defining how values are mapped from UPF to an HDL model or from the HDL model to UPF.
Return value	Return the fully qualified name of the supply net if successful or a 0 if not.	

Any ports, pins, cells, supply nets, or domains are referenced relative to the current UPF scope .

The **connect_supply_net** command provides an explicit connection of a supply net to any port and overrides (has higher precedence than) the auto-connection semantics that might otherwise apply. –**domain** or **-cells** is required when the **-rail_connection** or **-pg_type** options are specified.

Use:

-ports to connect to supply ports;

-pins to connect to pins on library cells;

-cells to connect to all pins of the appropriate type (power or ground) on the specified cells;

AMD EX1021
AMD v. Realtek - Page 339

–rail_connection to connect to pins having this (rail) type; only use this if the **-cells** or **-domain** options are specified;

–pg_type to connect only to pins having this power/ground type (the pin type defined in the library model is used for determining the connection); only use this if the **-cells** or **-domain** options are specified;

-vct to indicate for every HDL port to which the port is connected, the supply net state shall be converted if it is being propagated into the HDL port, or the HDL port value shall be converted if it is being propagated onto the supply net. **-vct** is ignored for any connections of the supply net to pins or supply ports defined in UPF.

The following also apply.

- The **–ports** and **–pins** options are mutually exclusive with the **–cells**, **–domain**, **-rail_connection**, and **-pg_type** options.
- The **-rail_connection** and **-pg_type** options are mutually exclusive with each other.
- A supply net can only be connected to a port, pin, or cell in the same *extent* as the domain of that supply net.
- It is an error if *net_name* has not been previously created; in this case, a 0 shall be returned.
- It is an error if any design element specified in this command does not exist.
- It is an error if the value conversions specified in the VCT do not match the type of the HDL port.

Syntax examples:

```
connect_supply_net v09 -ports {VDD U18/v9 U21/v9}

connect_supply_net pd1_vdd
  -ports pll_inst/vdd
  -vct upf2vlog_vdd
```

AMD EX1021
AMD v. Realtek - Page 340

B.3 create_power_domain

Purpose	Define a power supply distribution network for a set of design elements	
Syntax	**create_power_domain** *domain_name* [**-elements** *list*] [**-include_scope**] [**-scope** *instance_name*]	
Arguments	*domain_name*	The new power domain; this shall be a simple (non-hierarchical) name.
	-elements *list*	Use this set of design elements in the power domain.
	-include_scope	Include the scope of the domain in the extent of the power domain.
	-scope *instance_name*	Create the power domain within this logic hierarchy scope.
Return value	Return the fully qualified name of the domain (from the current scope) that is created or a null string if the power domain is not created.	

The **create_power_domain** command defines a power supply distribution network, usually for a list of design elements in the power domain. A *power domain* is a logical grouping of one or more design elements. A power domain shall have one primary power net and one primary ground net. A power domain may have additional supply nets, supply ports, and/or switches.

A power domain is functional only if the primary power and ground supply nets are specified.

–elements (or the **add_domain_elements** command) specifies the complete set of design elements contained within the power domain, i.e., those instances and all their children (unless specified otherwise).

- *list* is specified respective to the current scope; it is not influenced by the **–scope** argument.
- If **–include_scope** is also specified, the scope of the domain is included in the extent of the domain.
- When neither **–elements** nor **–include_scope** is specified, the power domain consists of the current scope and any of its children not specified (as elements) in another **create_power_domain** command.

AMD EX1021
AMD v. Realtek - Page 341

–scope specifies the scope, i.e., the instance, where the domain shall be created. This scope is the current scope for this command; it defines the domain boundary within the logic design. If **–scope** is not specified, the power domain shall be created within the current scope.

Syntax examples:

```
create_power_domain PD1 -elements {top/U1}

set_scope /top/U1
create_power_domain PD2
```

AMD EX1021
AMD v. Realtek - Page 342

B.4 create_power_switch

Purpose	Define a switch in the power domain	
Syntax	**create_power_switch** *switch_name* **-domain** *domain_name* **-output_supply_port** {*port_name supply_net_name*} {**-input_supply_port** {*port_name supply_net_name*}}* {**-control_port** {*port_name net_name*}}* {**-on_state** {*state_name input_supply_port* {*boolean_function*}}}* [**-on_partial_state** {*state_name input_supply_port* {*boolean_function*}}]* [**-ack_port** {*port_name net_name* [{*boolean_function*}]}]* [**-ack_delay** {*port_name delay*}]* [**-off_state** {*state_name* {*boolean_function*}}]* [**-error_state** {*state_name* {*boolean_function*}}]*	
Arguments	*switch_name*	The name of the switch instance to create; this shall be a simple name.
	-domain *domain_name*	The domain containing the switch.
	-output_supply_port *port_name supply_net_name*	The output supply port of the switch and the net where this port connects.
	-input_supply_port *port_name supply_net_name*	The input supply port of the switch and the net where this port is connected.
	-control_port {*port_name net_name*}	A control port on the switch and the net where this control port connects.
	-on_state {*state_name input_supply_port* {*boolean_function*}}	A named state, the *input_supply_port* for which this is defined, and its corresponding Boolean function.
	-on_partial_state {*state_name input_supply_port* {*boolean_function*}}	A named state, the *input_supply_port* for which this is defined, and its corresponding Boolean function.
	ack_port {*port_name net_name* [{*boolean_function*}]}	A named state, the *input_supply_port* for which this is defined, and its corresponding Boolean function where the switch is in a current-limited state.
	-ack_delay {*port_name delay*}	The acknowledge port on the switch and the signal net where this port connects. Optionally, a boolean function can also be specified.
	-off_state {*state_name* {*boolean_function*}}	The acknowledge port and delay on the switch where this port connects.

AMD EX1021
AMD v. Realtek - Page 343

	-error_state {*state_name* {*boolean_function*}}	Any error states, which if defined on the switch can be flagged during simulation or analysis.
Return value	Return the full path name of the created switch if successful or a null string if not.	

The **create_power_switch** command defines an instance of a power switch in the power domain. The switch is created within the scope of the power domain. The switch is either on or off.

The switch is on if the value on the control port(s) equals an "on" state Boolean expression; this drives the output port to the "on" state. If a value is specified on the input supply port, then that value is driven on the output supply port.

If the switch is not on, it is off, and an "off" state is driven onto the output port. Some "off" states can be identified as error states. The simulation semantics for these error states is tool dependent.

If a *boolean_function* is specified for **–ack_port**, the result of *boolean_function* is driven on **–ack_port**'s *port_name delay* time units after a control port transition. Otherwise, a logic 1 shall be driven on the *port_name delay* time units after the switch is closed and a logic 0 shall be driven *delay* time units after the switch is opened. *delay* (default is 0) may be specified as a unit-less natural integer or as a Verilog time unit. If specified as a natural integer, the time unit shall be the same as the simulation precision.

Any **–ack_port**, **-on_state**, or **-error_state** *boolean_function*s shall be SystemVerilog Boolean expressions.

The following also apply.

- It is an error if a specified control port does not exist. The existing connectivity of the control ports is unmodified.
- All states not covered by three states (on, off, and error) are anonymous error states.
- Each state name shall be unique.
- Any conflicting state definition is an error.
- No synthesis semantics are associated with any Boolean function, except the mapped switch has to provide that functionality.

Syntax example:

AMD EX1021
AMD v. Realtek - Page 344

```
create_power_switch sw1
-domain PD_SODIUM
-output_supply_port {vout VN3}
-input_supply_port {vin1 VN1}
-input_supply_port {vin2 VN2}
-control_port {ctrl_small ON1}
-control_port {ctrl_large ON2}
-control_port {ss SUPPLY_SELECT}
-on_state {partial_s1 vin1 {ctrl_small & !ctrl_large & ss}}
-on_state {full_s1 vin1 {ctrl_small & ctrl_large & ss}}
-on_state {partial_s2 vin2 {ctrl_small & !ctrl_large & !ss}}
-on_state {full_s2 vin2 {ctrl_small & ctrl_large & !ss}}
-error_state {no_small {!ctrl_small & ctrl_large}}
```

B.5 create_pst

Purpose	Create a power state table with a specific ordering of supply nets	
Syntax	**create_pst** *table_name* **-supplies** *list*	
Arguments	*table_name*	The power state table name (PST).
	-supplies *list*	The list of supply nets or ports to include in each power state of the design.
Return value	Return the name of the power state table if it is created or the null string if not.	

The **create_pst** command creates a PST, using a specific order of supply nets.

A *power state table* is used for implementation — specifically for synthesis, analysis, and optimization. It defines the legal combinations of states, i.e., those combinations of states that can exist at the same time during operation of the design.

The power state table has no simulation semantics. It is tool-dependent whether simulation tools report an error if an illegal (unspecified) combination of states occurs.

It is an error if a specified supply net has not already been created.

Syntax example:

```
create_pst MyPowerStateTable -supplies {PN1 PN2 SOC/OTC/PN3}
```

AMD EX1021
AMD v. Realtek - Page 345

B.6 create_supply_net

Purpose	Create a power or ground supply net	
Syntax	**create_supply_net** *net_name* **-domain** *domain_name* [**-reuse**] [**-resolve <unresolved | one_hot | parallel>**]	
Arguments	*net_name*	The name of the supply net; this shall be a simple (non-hierarchical) name.
	-domain *domain_name*	The domain in whose scope the supply net is to be created.
	-reuse	Extend *net_name* as a supply net within *domain_name*. No new nets are created.
	-resolve <unresolved | one_hot | parallel>	A resolution mechanism which determines the state and voltage of the supply net from the state and voltage values supplied by each of the individual switches. The default is **unresolved**.
Return value	Return the fully qualified name (from the current scope) of the created net or a null string if the net is not created.	

The **create_supply_net** command creates a supply net. The net is defined for the power domain, created in the logic hierarchy at the same scope as *domain_name*, and propagated through implicitly created ports and nets through the logic hierarchy as required .

The following also apply.

- It is an error if *domain_name* does not indicate a previously created power domain.
- When **-reuse** is specified, it is an error if *net_name* does not already exist.
- When **-resolve unresolved** is specified, this supply net only allows a single driver.

Syntax example:

```
create_supply_net v09
   -domain PD1
```

AMD EX1021
AMD v. Realtek - Page 346

B.7 create_supply_port

Purpose	Create a port on a power domain	
Syntax	**create_supply_port** *port_name* [**-domain** *domain_name*] [**-direction** <**in** | **out**>]	
Arguments	*port_name*	The name of the supply port. Hierarchical names are allowed, unless **-domain** is also specified.
	-domain *domain_name*	The domain where this port defines a supply net connection point.
	-direction <**in** | **out**>	The direction of the port. The default is **in**.
Return value	Return the fully qualified name (from the current scope) of the created port or a null string if the port is not created.	

The **create_supply_port** command defines a supply port at the scope of the power domain when **-domain** is specified, or at the current scope if **-domain** is not specified.

-direction defines how state information is propagated through the supply network as it is connected to the port. If the port is an input port, the state information of the external supply net connected to the port shall be propagated into the domain. Likewise, for an output port, the state information of the internal supply net connected to the port shall be propagated outside of the domain.

It is an error if *port_name* creates a name conflict in the logic hierarchy or specifies a previously created port.

It is an error if *domain_name* does not indicate a previously created power domain.

Syntax example:

```
create_supply_port VN1
   -domain PD1
  reg/wire, Bit, Logic
```

AMD EX1021
AMD v. Realtek - Page 347

B.8 set_domain_supply_net

Purpose	Set the default power and ground supply nets for a power domain	
Syntax	**set_domain_supply_net** *domain_name* **-primary_power_net** *supply_net_name* **-primary_ground_net** *supply_net_name*	
Arguments	*domain_name*	The domain where the default supply nets are to applied.
	-primary_power_net *supply_net_name*	The primary power supply net.
	-primary_ground_net *supply_net_name*	The primary ground net.
Return value	Return 1 if it succeeds and 0 if it fails.	

The **set_domain_supply_net** command associates the default power and ground supply nets to the logic elements of the power domain.

The *primary power and ground nets* are the default nets connected to the logic elements (or inferred cells) of the power domain. These power and ground nets are used for all the elements in the power domain. At the gate level, this means all power/ground pins of all the inferred gates are connected to the primary power/ground nets, unless specified otherwise in a **connect_supply_net**, **set_retention**, or **set_isolation** command.

It is an error if *domain_name* does not indicate a previously created power domain.

It is an error if *domain_name* already contains a primary power and ground net supply.

Syntax example:

```
set_domain_supply_net PD1
-primary_power_net PG1
-primary_ground_net PG0
```

AMD EX1021
AMD v. Realtek - Page 348

B.9 set_isolation

Purpose	Specify the elements in the domain to isolate using the specified strategy	
Syntax	**set_isolation** *isolation_name* **-domain** *domain_name* <**-isolation_power_net** *net_name* | **-isolation_ground_net** *net_name* | **-isolation_power_net** *net_name* **-isolation_ground_net** *net_name* | **-no_isolation**> [**-elements** *list*] [**-clamp_value** <**0** | **1** | *latch* | **Z**>] [**-applies_to** <**inputs** | **outputs** | **both**>]	
Arguments	*isolation_name*	Isolation strategy name.
	-domain *domain_name*	The domain for which this strategy is applied.
	<**-isolation_power_net** *net_name* | **-isolation_ground_net** *net_name* | **-isolation_power_net** *net_name* **-isolation_ground_net** *net_name* | **-no_isolation**>	The supply net(s) used to supply the isolation logic inferred by this strategy. Does not isolate the port, pin, or design element specified in the elements list.
	-elements *list*	A list of design elements, input ports/pins, output ports/pins, and nets to which this strategy is applied.
	-clamp_value <**0** | **1** | *latch* | **Z**>	The value to which the input or output shall be clamped. The default is **0**.
	-applies_to <**inputs** | **outputs** | **both**>	Whether the domain's input ports, output ports, or both are isolated. The default is **outputs**.
Return value	Return `1` if it succeeds and `0` if it fails.	

The **set_isolation** command specifies the elements in the domain to isolate using the specified strategy — the isolation enable signal common to all, clamp value, and location.

–isolation_supply_nets can specify a single power net, a single ground net, or both. If only an isolation power net is specified, then the primary ground serves as the

AMD EX1021
AMD v. Realtek - Page 349

isolation ground. If only an isolation ground net is specified, then the primary power net serves as the isolation power.

At least one of **-isolation_power_net** or **-isolation_ground_net** shall be specified, unless **-no_isolation** is specified. If only **-isolation_power_net** is specified, the primary ground net shall be used as the isolation ground supply. If only **-isolation_ground_net** is specified, the primary power net shall be used as the isolation power supply. If both are specified, then these options specify the supply nets to use as the isolation power and ground nets.

The isolation power and ground nets are automatically connected to the implicit isolation processes and, when **-clamp_value** is *latch*, to the isolation value register described in section 5 of the full specification or to the power and ground pins of an isolation cell if an isolation cell is explicitly mapped with the **map_isolation_cell** command. The switched off and on semantics for the implicit isolation process and isolation register, if present, are as described in section 5 of the full specification.

If there are multiple isolation strategies for one domain then, per strategy, **–elements** can be used to specify the elements to isolate. If **–elements** is specified, the elements shall be in the *domain_name*. If **–elements** directly specifies a port by name (not implicitly, by specifying the port's instance or an ancestor of that instance), then the isolation strategy shall apply to that port regardless of whether that port's mode matches the one specified by the **–applies_to** option. When **–elements** is not specified, this is equivalent to using the elements list that defines the power domain.

-clamp_value can be:

- logic 0
- logic 1
- *latch* (the value of the non-isolated port when the isolation signal becomes active)
- logic Z

The following also apply.

- This command never applies to *inout* ports.
- If **-no_isolation** is specified, then all the ports specified in **-elements**, explicitly or implicitly, and matching the **-applies_to** (mode) shall not be isolated.
- It is an error if **-no_isolation** is specified with any other arguments other than **-domain**, **-elements**, or **-applies_to**.
- It is an error if more than one power or ground net is specified.
- It is an error if multiple isolation strategies specify the same design element, pins, ports, or nets.

AMD EX1021
AMD v. Realtek - Page 350

Syntax example:

```
set_isolation outputs_only
  -domain PD1
  -isolation_power_net VDDbackup
  -clamp_value 1
  -applies_to outputs
```

B.10 set_isolation_control

Purpose	Specify the control signals for a previously defined isolation strategy	
Syntax	**set_isolation_control** *isolation_name* **-domain** *domain_name* **-isolation_signal** *signal_name* [**-isolation_sense** <**high** | **low**>] [**-location** <**self** | **parent** | **sibling** | **fanout** | **automatic**>]	
Arguments	*isolation_name*	Isolation strategy name.
	-domain *domain_name*	The domain where the strategy applies.
	-isolation_signal *signal_name*	The signal that causes the specified element to drive its clamp value.
	-location <**self** | **parent** | **sibling** | **fanout** | **automatic**>	Where the isolation cells are placed in the logic hierarchy. The default is **automatic**.
Return value	Return 1 if it succeeds and 0 if it fails.	

The **set_isolation_control** command allows the specification of the isolation control signal and sense separate from the **set_isolation** command for those situations where the isolation strategy is known, but the control signals are not known until later.

Excepting the **set_isolation_control** command is executed within the current scope, and the addition of the **–location** option, the semantics here are equivalent to having specified the isolation control signal and sense with the **set_isolation** command.

–location defines where the isolation cells are placed in the logic hierarchy.

a) **self** – the isolation cell is placed inside the model/cell being isolated.

b) **parent** – the isolation cell is placed in the parent of the cell /model being isolated.

AMD EX1021
AMD v. Realtek - Page 351

c) **sibling** – a new sibling is created into which the isolation cells are placed.

d) **fanout** – isolation occurs at all fanout locations (sinks) of the port being isolated.

e) **automatic** – the implementation tool is free to choose the appropriate locations (the default).

Syntax example:

```
set_isolation outputs_only
-domain PD1
-isolation_power_net VDDbackup
-clamp_value 1
-applies_to outputs

set_isolation_control outputs_only
-domain PD1
-isolation_signal cpu_iso
-isolation_sense low
-location parent
```

AMD EX1021
AMD v. Realtek - Page 352

B.11 set_level_shifter

Purpose	Specify a level shifter strategy	
Syntax	**set_level_shifter** *level_shifter_name* **-domain** *domain_name* [**-elements** *list*] [**-applies_to** <**inputs** | **outputs** | **both**>] [**-threshold** *value*] [**-rule** <**low_to_high** | **high_to_low** | **both**>] [**-location** <**self** | **parent** | **sibling** | **fanout** | **automatic**>] [**-no_shift**]	
Arguments	*level_shifter_name*	Level shifter strategy name (used only for reporting).
	-domain *domain_name*	The domain for which this strategy is applied.
	-elements *list*	A list of design elements, pins, ports, or nets to which this strategy is applied.
	-applies_to <**inputs** | **outputs** | **both**>	Whether the domain's input ports, output ports, or both are level shifted. The default is **both**.
	-threshold *value*	The voltage threshold (in volts) for determining when level shifters are required. The default is 0.
	-rule <**low_to_high** | **high_to_low** | **both**>	Which type of level shifters are required. The default is **both**.
	-location <**self** | **parent** | **sibling** | **fanout** | **automatic**>	Where the level shifter is placed in the logic hierarchy. The default is **automatic**.
	-no_shift	Can be specified with the **–elements** option to prevent the insertion of level shifters on the specified ports/pins and nets.
Return value	Return 1 if it succeeds and 0 if it fails.	

The **set_level_shifter** command can be used to set a strategy for level shifting during implementation. *Level shifters* are the placed on signals that have sources and sinks operating at different voltages, as their associated design elements are connected to different supply nets. If a level shifter strategy is not specified on a particular power domain, the default level shifter strategy consists of all elements in the power domain and uses the default strategy settings.

If **–elements** is specified, the elements shall be in the *domain_name*. If **–elements** is used to specify a port or pin, a level shifter is inserted on that port regardless of any

AMD EX1021
AMD v. Realtek - Page 353

-threshold or **-rule** specifications. The **-threshold** specification defines how large the voltage difference between the driver and sink needs to be before level shifters are inserted. Normally, this threshold is determined from the cell libraries; use this option to override the library values.

–rule can be **low_to_high**, **high_to_low**, or **both**. If **low_to_high** is specified, signals going from a lower voltage to a higher voltage get a level shifter when the voltage difference exceeds that specified by **–threshold**. If **high_to_low** is specified, signals going from a higher voltage to a lower voltage get a level shifter when the voltage difference exceeds that specified by **–threshold**. If **both** is specified, it is equivalent to having specified both rules in the strategy.

–location defines where the level shifter cells are placed in the logic hierarchy. All necessary supplies need to be available in the specified location.

a) Self – the level shifter cell is placed inside the model/cell being shifted.
b) Parent – the level shifter cell is placed in the parent of the cell /model being shifted.
c) Sibling – a new sibling is created into which the level shifter cells are placed.
d) Fanout – level shifter occur at all fanout locations (sinks) of the port being shifted.
e) Automatic – the implementation tool is free to choose the appropriate locations.

The following also apply.

— This command never applies to *inout* ports.
— It is an error if the specified location is not within the logic design starting at the design root.

Syntax example:

```
set_level_shifter shift_up
  -domain PowerDomainZ
  -applies_to outputs
  -threshold 0.02
  -rule both
```

AMD EX1021
AMD v. Realtek - Page 354

B.12 set_retention

Purpose	Specify which registers in the domain need to be retention registers and set the save and restore signals for the retention functionality	
Syntax	**set_retention** *retention_name* **-domain** *domain_name* <**-retention_power_net** *net_name* | **-retention_ground_net** *net_name* | **-retention_power_net** *net_name* **-retention_ground_net** *net_name*> [**-elements** *list*]	
Arguments	*retention_name*	Retention strategy name (used only for reporting).
	-domain *domain_name*	The domain for which this strategy is applied.
	<- **retention_power_net** *net_name* | **-retention_ground_net** *net_name* | **-retention_power_net** *net_name* **-retention_ground_net** *net_name*>	The supply net(s) used to supply the retention registers inferred by this strategy.
	-elements *list*	A list of objects in the power domain: design elements, named processes, or sequential `reg` or signal names to which this strategy is applied.
Return value	Return 1 if it succeeds and 0 if it fails.	

The **set_retention** command specifies which registers in the domain need to be retention registers and identifies the save and restore signals for the retention functionality. Only the registers implied in the elements list shall be provided retention capabilities. If a design element is specified, then all registers within the design element acquire the specified retention strategy. If a process is specified, then all registers inferred by the process acquire the specified retention strategy. If a `reg`, signal, or variable is specified and that object is a sequential element, then the implied register acquires the specified retention strategy. Any specified `reg`, signal, or variable that does not infer a sequential element shall not be changed by this command.

At least one of **-retention_power_net** or **-retention_ground_net** shall be specified. If only **-retention_power_net** is specified, the primary ground net shall be used as the retention ground supply. If only **-retention_ground_net** is specified, the primary power

AMD EX1021
AMD v. Realtek - Page 355

net shall be used as the retention power supply. If both are specified, then these options specify the supply nets to use as the retention power and ground nets.

The retention power and ground nets are automatically connected to the implicit save and restore processes and shadow register or to the power and ground pins of a retention cell when that retention cell is explicitly mapped with the **map_retention_cell** command. The switched off and on semantics for the implicit save and restore processes and shadow register are as described in section 5 of the full specification.

If **-save_signal** is specified, then **-restore_signal** shall be specified. If **-save_signal** is not specified, then **-restore_signal** shall not be specified. If the save and restore signals are not specified, then they shall be specified in a **set_retention_control** command.

If **–elements** is specified, the elements shall be in the *domain_name*. When **–elements** is not specified, this is equivalent to using the elements list that defines the power domain.

The following also apply.

— It is an error if *domain_name* does not indicate a previously created power domain.

— It is an error if more than one power or ground net is specified.

Syntax example:

```
set_retention my_retention
 -domain PDA
 -retention_power_net volt_high
```

AMD EX1021
AMD v. Realtek - Page 356

B.13 set_retention_control

Purpose	Specify the control signals and assertions for a previously defined retention strategy	
Syntax	**set_retention_control** *retention_name* **-domain** *domain_name* **-save_signal** {{*net_name* <**high** | **low** | **posedge** | **negedge**>}} **-restore_signal** {{*net_name* <**high** | **low** | **posedge** | **negedge**>}} [**-assert_r_mutex** {{*net_name* <**high** | **low** | **posedge** | **negedge**>}}]* [**-assert_s_mutex** {{*net_name* <**high** | **low** | **posedge** | **negedge**>}}]* [**-assert_rs_mutex** {{*net_name* <**high** | **low** | **posedge** | **negedge**>}}]*	
Arguments	*retention_name*	Retention strategy name (used only for reporting).
	-domain *domain_name*	The domain for which this strategy is applied.
	-save_signal *save_net*	The signal that causes the register values to be saved into the shadow registers.
	-restore_signal *restore_net*	The signal that causes the register values to be restored from the shadow registers.
	-assert_r_mutex {{*net_name* <**high** | **low** | **posedge** | **negedge**>}}	The restore signal for assertion.
	-assert_s_mutex {{*net_name* <**high** | **low** | **posedge** | **negedge**>}}	The save signal for assertion.
	-assert_rs_mutex {{*net_name* <**high** | **low** | **posedge** | **negedge**>}}	Both signals (save and restore) for assertion.
Return value	Return 1 if it succeeds and 0 if it fails.	

The **set_retention_control** command allows the specification of the retention control signal and sense separate from the **set_retention** command for those situations where the retention strategy is known, but the control signals are not known until later. As the assertions are related to the save and restore signals, they can also be specified with this command.

Excepting the **set_retention_control** command is executed within the current scope, the semantics here are equivalent to having specified the retention control signals, senses, and assertions with the **set_retention** command.

AMD EX1021
AMD v. Realtek - Page 357

The **set_retention** command can also be used to specify any assertion options. Each option creates one or more assertions, which verification tools can trigger when the indicated RTL signals are active simultaneously. If **-assert_rs_mutex** does not specify a list of signals, this indicates the save and restore signals themselves are mutually exclusive.

The following also apply.

— The save signal shall be an existing net, port, or pin in the design.

— The restore signal shall be an existing net, port, or pin in the design.

Syntax example:

```
set_retention my_retention_strategy
-domain PDA

set_retention_control my_retention_strategy
-domain PDA
-save_signal {power_controller_inst/save_1 high}
-restore_signal {power_controller_inst/restore_1 low}
-assert_rs_mutex {clock_a posedge}
```

B.14 set_scope

Purpose	Specify the current UPF scope	
Syntax	**set_scope** *instance*	
Arguments	*instance*	The instance that becomes the current scope upon completion of the command.
Return value	Returns the current scope prior to execution of the command as a full path string relative to the current design top if successful and the null string if it fails (e.g., if the instance does not exist).	

If the **set_scope** command is called with no arguments or the UPF scope is not set, the scope is set to the top-level design.

If *instance* is `.`, the scope remains at the current instance. If *instance* is `..`, the context is moved up one level in the instance hierarchy. If *instance* begins with `/`, the scope returns to the instance whose name follows the `/`, relative to the design top.

AMD EX1021
AMD v. Realtek - Page 358

Syntax examples:

```
set_scope foo/bar

set_scope ..
```

AMD EX1021
AMD v. Realtek - Page 359

Glossary

Power Domain: A collection of design elements that share a single primary supply connection and, at least conceptually, share a common power strategy.

Isolation: Isolation is a technique for controlling the behavior of a signal that is driven by a powered down power domain. Isolation consists of driving the signal to a know state - 1, 0, or latching it to a previous value when the power domain is powered down.

Retention: Retention is a technique for retaining the state value of registers in a powered down power domain.

Isolation Cells: Cells (gates) that perform the isolation function in a design. Also knows as clamp cells or clamps.

Level Shifters: Cells (typically buffers) that translate inputs with one voltage swing to an output with a different voltage swing.

Retention Register: A register than extends the functionality of a normal register (flip-flop) with the ability to retain its memory during power down, assuming an appropriate second (always on) supply as well as save and restore signaling.

Shadow Register: The section of a retention register retains the register state during power down. Also known as a balloon registers (due to the topology of some implementations).

Power Switch: At the RTL and architectural level, a power switch allows the power to a power domain to be switched on or off (also known as power gated).

AMD EX1021
AMD v. Realtek - Page 360

Switching Network: Physically, the power switch is implemented as a switching network of transistors. Also known as a switching fabric.

Switching Transistor: The individual switching transistor that makes up the switching network. Also know as a sleep transistor.

AMD EX1021
AMD v. Realtek - Page 361

Low Power Methodology Manual

For System-on-Chip Design

"Tools alone aren't enough to reduce dynamic and leakage power in complex chip designs - a well-planned methodology is needed. Following in the footsteps of the successful Reuse Methodology Manual (RMM), authors from ARM and Synopsys have written this Low Power Methodology Manual (LPMM) to describe [such] [a] low-power methodology with a practical, step-by-step approach."

Richard Goering, Software Editor, EE Times

"Excellent compendium of low-power techniques and guidelines with balanced content spanning theory and practical implementation. The LPMM is a very welcome addition to the field of low power SoC implementation that has for many years operated in a largely ad-hoc fashion."

Sujeeth Joseph, Chief Architect – Semiconductor & Systems Solutions Unit, Wipro Technologies

"The LPMM enables broader adoption of aggressive power management techniques based on extensive experience and silicon example with real data that every SOC designer can use to meet the difficulties faced in managing the power issues in deep submicron designs."

Anil Mankar, Sr. VP Worldwide Core Engineering and Chief Development Officer, Conexant Systems Inc.

"Managing power, at 90nm and below, introduces significant challenges to design flow. The LPMM is a timely and immediately useful book that shows how combination of tools, IP and methodology can be used together to address power management."

Nick Salter, Head of Chip Integration, CSR plc.

ABOUT THE AUTHORS:

Michael Keating is a Synopsys Fellow in the company's Advanced Technology Group, focusing on IP development methodology, hardware and software design quality and low power design.

David Flynn is an ARM R&D Fellow and has been with the company since 1991, specializing in low power System-on-Chip IP deployment and methodology.

Robert Aitken is an ARM R&D Fellow. His areas of responsibility include memory architecture, design for testability and design for manufacturability.

Alan Gibbons is a Principal Engineer at Synopsys, with a focus on development of advanced methodology and technology for ARM processor-based system design.

Kaijian Shi is a Principal Consultant in the Professional Services Group of Synopsys, specializing in low power design methodology and implementation.

ARM SYNOPSYS

Advancing Design Solutions



Springer

springer.com

ISBN 978-0-387-71818-7



AMD EX1021
AMD v. Realtek - Page 362

ATTACHMENT 6A

```
001    9913953754406531
005    20071127  TAPE OCLC1127: 0054
008    071011s2007    nyua     b    001 0 eng d
020    ##$a9780387718187
020    ##$a0387718184
035    ##$a(CUY)UCB-b126938039-01ucs_ber
035    ##$a(OCoLC)156812961 $9ExL
040    ##$aBTCTA $cBTCTA $dBAKER $dYDXCP $dIWA $dCUY
090    ##$aTK7874.66 $b.L67 2007 $9LOCAL
245    00$aLow power methodology manual : $bfor system-on-chip
design / $cMichael Keating ... [et al.].
260    ##$aNew York, NY : $bSpringer, $cc2007.
300    ##$axvi, 300 p. : $bill. ; $c25 cm.
504    ##$aIncludes bibliographical references (p. [293]-296) and
index.
650    #0$aLow voltage integrated circuits.
650    #0$aSystems on a chip.
700    1#$aKeating, Michael, $d1950 May 2-
968    ##$aReclvl: f Addate: 071011 Addid: INN Moddate: 090126
Modid: JFK $9LOCAL
996    ##$eGLADN17627999 $9LOCAL
996    ##$ab126938039 $bUpdated: 01-09-12 $cCreated: 03-23-09
$dCatDate: 10-11-07 $9LOCAL
```

ATTACHMENT 6B

OCLC 156812961 **No holdings in XXX - 67 other holdings**

Books							**Rec Stat**		**Entered**	20070507	**Replaced**	20220428213332.9
Type	a	**ELvl**	7	**Srce**	d	**Audn**		**Ctrl**		**Lang**	eng
BLvl	m	**Form**		**Conf**	0	**Biog**		**MRec**		**Ctry**	nyu
		Cont	b	**GPub**		**LitF**	0	**Indx**	1		
Desc	a	**Ills**	a	**Fest**	0	**DtSt**	s	**Dates**	2007 ,		

010 2007928355
040 BTCTA $b eng $c DLC $d BTCTA $d BAKER $d YDXCP $d IWA $d IXA $d OHX $d AU@ $d OCLCF $d SOI $d OCLCQ $d OCLCO
020 9780387718187 $q (alk. paper)
020 0387718184 $q (alk. paper)
020 9780387718194 $q (e-ISBN)
020 0387718192 $q (e-ISBN)
042 lccopycat
050 00 TK7874.66 $b .L67 2007
082 04 621.381 $2 22
090 $b
245 00 Low power methodology manual : $b for system-on-chip design / $c Michael Keating [and others].
260 New York, NY : $b Springer, $c ©2007.
300 xvi, 300 pages : $b illustrations ; $c 25 cm
336 text $b txt $2 rdacontent
337 unmediated $b n $2 rdamedia
338 volume $b nc $2 rdacarrier
504 Includes bibliographical references (pages 293-296) and index.
505 0 Standard low power methods -- Multi-voltage design -- Power gating overview -- Designing power gating -- Architectural issues for power gating -- A power gating example -- IP design for low power -- Frequency and voltage dsaling design -- Examples of voltage and frequency scaling design -- Implementing multi-voltage, power gated designs -- Physical libraries -- Retention register design -- Design of the power switching network.
650 0 Low voltage integrated circuits.
650 0 Systems on a chip.
650 6 Circuits intégrés à faible consommation. $0 (CaQQLa)201-0281964
650 6 Systèmes sur une puce. $0 (CaQQLa)201-0368201
650 7 Low voltage integrated circuits. $2 fast $0 (OCoLC)fst01003178
650 7 Systems on a chip. $2 fast $0 (OCoLC)fst01141473
700 1 Keating, Michael, $d 1950 May 2-
856 41 $3 Table of contents $u http://catdir.loc.gov/catdir/enhancements/fy0825/2007928355-t.html
856 42 $3 Contributor biographical information $u http://catdir.loc.gov/catdir/enhancements/fy0825/2007928355-b.html
856 42 $3 Publisher description $u http://catdir.loc.gov/catdir/enhancements/fy0825/2007928355-d.html

ATTACHMENT 6C

LC control no.: sh 95004622
LCCN Permalink: https://lccn.loc.gov/sh95004622
HEADING: Low voltage integrated circuits

000 00953cz a2200265n 450
001 4868654
005 20120330082726.0
008 950608i| anannbabn |a ana
010 __ |**a** sh 95004622
035 __ |**a** (DLC)sh 95004622
035 __ |**a** (DLC)215339
040 __ |**a** DLC |**c** DLC |**d** DLC
053 _0 |**a** TK7874.66
150 __ |**a** Low voltage integrated circuits
450 __ |**a** Low power consumption chips
450 __ |**a** Low power integrated circuits
450 __ |**a** Reduced voltage integrated circuits
550 __ |**w** g |**a** Integrated circuits
550 __ |**w** g |**a** Low voltage systems
670 __ |**a** Work cat.: 95-21368: Low-voltage low-power analog integrated circuits, 1995.
670 __ |**a** ASCI on Firstsearch, 06/08/95 |**b** (subject term: Low voltage integrated circuits; abstract: reduced voltage integrated circuits)
670 __ |**a** Wilson bus. abstr. CD-ROM, 82-94 |**b** (in abstract: low-power-consumption chips)
675 __ |**a** EA, 1990.
906 __ |**t** 9526 |**u** te04 |**v** 1
953 __ |**a** jg12 |**b** jf08

◀ Previous Next ▶

Save, Print or Email Records (View Help)

Select Download Format: Text (Labelled Display) | Press to SAVE or PRINT

Email Text (Labelled Display) to:
Press to SEND EMAIL



Library of Congress
URL: https://www.loc.gov/

Mailing Address:
101 Independence Ave, S.E.
Washington, DC 20540

Library of Congress Authorities
URL: http://authorities.loc.gov/
Library of Congress Online Catalog
URL: https://catalog.loc.gov/

Questions, comments, error reports: Contact Us

ATTACHMENT 6D

LC control no.: sh2002000568

LCCN Permalink: https://lccn.loc.gov/sh2002000568

HEADING: Systems on a chip

000 00973cz a2200265n 450

001 5636724

005 20190825061713.0

008 020425|| anannbabn |a ana

010 __ |**a** sh2002000568

035 __ |**a** (DLC)5636724

035 __ |**a** (DLC)sh2002000568

035 __ |**a** (DLC)284544

040 __ |**a** DLC |**b** eng |**c** DLC |**d** DLC

053 _0 |**a** TK7874.888

150 __ |**a** Systems on a chip

450 __ |**a** SOC design

450 __ |**a** Systems on chip

550 __ |**w** g |**a** Embedded computer systems

670 __ |**a** Work cat.: 2002069402: Test resource partitioning for system-on-a-chip, 2002: |**b** CIP contents (design philosophy; SOC)

670 __ |**a** ASTI on FirstSearch, Apr. 25, 2002 |**b** (descriptor: system-on-a-chip)

670 __ |**a** WWW.us.design-reuse.com |**b** (article: Software key to SoC design; text: Systems on chip; presentation at Embedded Systems Conference)

906 __ |**t** 0221 |**u** te04 |**v** 1

910 __ |**a** Record saved by td04 on 01/29/2019 at 10:56:39

952 __ |**a** 10 bib. record(s) to be changed

953 __ |**a** jg12

◀ Previous Next ▶

Save, Print or Email Records (View Help)

Select Download Format: Text (Labelled Display) Press to SAVE or PRINT

Email Text (Labelled Display) to:

Press to SEND EMAIL



Library of Congress
URL: https://www.loc.gov/

Mailing Address:
101 Independence Ave, S.E.
Washington, DC 20540

Library of Congress Authorities
URL: http://authorities.loc.gov/
Library of Congress Online Catalog
URL: https://catalog.loc.gov/

Questions, comments, error reports: Contact Us

ATTACHMENT 6E

Copyright United States Copyright Office

Try the Copyright Public Records System (CPRS) pilot with enhanced search features and filters.

Help | Search | History | Titles | Start Over

Public Catalog

Copyright Catalog (1978 to present)
Search Request: Left Anchored Title = low power methodology manual
Search Results: Displaying 1 of 1 entries

previous next

Labeled View

Low Power Methodology Manual.

Type of Work: Text
Registration Number / Date: TX0007433405 / 2011-10-13
Application Title: Low Power Methodology Manual.
Title: Low Power Methodology Manual.
Description: Book, 300 p.
Copyright Claimant: Synopsys, Inc., Transfer: By written agreement. Address: 700 East Middlefield Road, Mountain View, CA, 94043, United States.
Date of Creation: 2007
Date of Publication: 2007-07-31
Nation of First Publication: United States
Authorship on Application: Michael Keating. Authorship: text.
David Flynn. Authorship: text.
Robert Aitken. Authorship: text.
Alan Gibbons. Authorship: text.
Kaijian Shi. Authorship: text.
Copyright Note: C.O. correspondence.
ISBN: 978-0-387-71818-7
Names: Keating, Michael
Flynn, David
Aitken, Robert
Gibbons, Alan
Shi, Kaijian
Synopsys, Inc.

previous next

Save, Print and Email (Help Page)
Select Download Format Full Record Format for Print/Save
Enter your email address: Email

Help Search History Titles Start Over

Contact Us | Request Copies | Get a Search Estimate | Frequently Asked Questions (FAQs) about Copyright | Copyright Office Home Page | Library of Congress Home Page

ATTACHMENT 7

ISPD-97

1997 International Symposium on Physical Design

April 14-16, 1997
Embassy Suites at Napa Valley, Napa, CA, USA

Sponsored by
ACM Special Interest Group on Design Automation (SIGDA)
In cooperation with
IEEE Circuits and Systems Society

Additional support from

Avant! Corporation
Cadence Design Systems, Inc.
Intel Corporation
Synopsys, Inc.
U.S. National Science Foundation

AMD EX1021
AMD v. Realtek - Page 377

QA76.75
.I58
1997

The Association for Computing Machinery
1515 Broadway
New York New York 10036

Copyright 1997 by the Association for Computing Machinery, Inc.(ACM). Permission to make digital or hard work for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists requires prior specific permission and/or a fee. Request permission to republish from: Publications Dept. ACM, Inc. Fax +1 (212) 869-0481 or <permissions@acm.org> For other copying of articles that carry a code at the bottom of the first or last page, copying is permitted provided that the per-copy fee indicated in the code is paid through the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923.

ACM ISBN: 0-89791-927-0

Additional copies may be ordered prepaid from:

ACM Order Department
PO Box 12114
Church Street Station
New York, NY 10257

Phone: 1-800-342-6626
(US and Canada)
+1-212-626-0500
(all other countries)
Fax: +1-212-944-1318
E-mail: acmpubs@acm.org

ACM European Service Center
108 Cowley Road
Oxford OX 4 1 JF
UK

Phone: +44-1-865-382338
Fax: +44-1-865-381338
E-Mail: acm_europe@acm.org
URL:http://www.acm.org

ACM Order Number: 477975

Printed in the USA

AMD EX1021
AMD v. Realtek - Page 378

Table of Contents

Foreword iv

Symposium Organization v

Table of Contents vi

Keynote Address

- Physical Design: Past and Future
  T. C. Hu and E. S. Kuh

Session 1: Placement and Partitioning

Moderators: D. Hill (Synopsys)
J. Frankle (Aristo Technology)

- Faster Minimization of Linear Wirelength for Global Placement **4**
  C. J. Alpert, T. Chan, D. J. Huang, A. B. Kahng, I. Markov, P. Mulet and K. Yan
- Network Flow Based Multi-Way Partitioning with Area and Pin Constraints **12**
  H. Liu and D. F. Wong
- Partitioning-Based Standard-Cell Global Placement with An Exact Objective **18**
  D. J. Huang and A. B. Kahng
- VLSI/PCB Placement with Obstacles Based on Sequence Pair **26**
  H. Murata, K. Fujiyoshi and M. Kaneko

Invited Address

- The Quarter Micron Challenge: Integrating Physical and Logic Design
  R. Camposano

Session 2: Synthesis and Layout

Moderators: R. Camposano (Synopsys)
C. Sechen (Washington)

- Timing Driven Placement in Interaction with Netlist Transformations **36**
  G. Stenz, B. R. Riess, B. Rohfleisch and F. M. Johannes
- Regular Layout Generation of Logically Optimized Datapaths **42**
  R. X. T. Nijssen and C. A. J. van Eijk
- Minimizing Interconnect Energy Through Integrated Low-Power Placement and Combinational Logic Synthesis **48**
  G. Holt and A. Tyagi

AMD EX1021
AMD v. Realtek - Page 379

Session 3: Contexts (Invited)

- Design Technology Trends Based on NTRS Evolution
  P. Verhofstadt and C. D'Angelo
- Microprocessor Architecture, Circuit, and Physical Design Trends
  R. Panwar

Invited Address

- Lithography and Dimensional Trends for Future Processes – Implications for Physical Design
  P. K. Vasudev

Session 4: Routing

Moderators: C. L. Liu (Illinois)
D. F. Wong (UT-Austin)

- On Two-Step Routing for FPGAs **60**
  G. G. Lemieux, S. D. Brown and D. Vranesic
- A Simple and Effective Greedy Multilayer Router for MCMs **67**
  Y.-J. Cha, C. S. Rim and K. Nakajima
- Performance Driven Global Routing for Standard Cells **73**
  J. Cong and P. Madden

Session 5: Steiner Tree Constructions

Moderators: M. Marek-Sadowska (UCSB)
N. Sherwani (Intel)

- Min-Cost Flow based Min-Cost Rectilinear Steiner
  Distance-Preserving Tree Construction **82**
  J. D. Cho
- Efficient Heuristics for the Minimum Shortest Path Steiner Arborescence Problem
  with Applications to VLSI Physical Design **88**
  J. Cong, A. B. Kahng and K.-S. Leung
- Provably Good Routing Tree Construction with Multi-Port Terminals **96**
  C. Bateman, C. S. Helvig, G. Robins and A. Zelikovsky

AMD EX1021
AMD v. Realtek - Page 380

Session 6: Back-End Design Methodology

Moderators: E. Yoffa (IBM)
M. Weisel (Intel)

- A Roadmap of CAD Tool Changes for Sub-Micron Interconnect Problems **104**
  L. Scheffer
- C5M - A Control Logic Layout Synthesis System for
  High-performance Microprocessors **110**
  J. Burns and J. Feldman
- A VLSI Artwork Legalization Technique Based on a New Criterion
  of Minimum Layout Perturbation **116**
  F.-L. Heng, Z. Chen and G. E. Tellez

Session 7: Poster Presentations

Moderators: G. Robins (Virginia)
J. D. Cho (SungKyunKwan)

- A Pseudo-Hierarchical Methodology for High Performance Microprocessor Design **124**
  A. Bertolet, K. Carpenter, K. Carrig, A. Chu, A. Dean, F. Ferraiolo, S. Kenyon, D. Phan, J. Petrovick, G. Rodgers, D. Willmott, T. Bairley, T. Decker, V. Girardi, Y. Lapid, M. Murphy, P. A. Scott and R. Weiss
- Concurrent Transistor Sizing and Buffer Insertion
  by Considering Cost-Delay Tradeoffs **130**
  J. Kim, C. Bamji, Y. Jiang and S. Sapatnekar
- Towards a New Benchmarking Paradigm in EDA **136**
  N. Kapur, D. Ghosh and F. Brglez
- How Good are Slicing Floorplans? **144**
  F. Y. Young and D. F. Wong
- Slicibility of Rectangular Graphs and Floorplan Optimization **150**
  P. DasGupta and S. Sur-kolay
- Power Optimization for FPGA Look-Up Tables **156**
  M. J. Alexander
- A Matrix Synthesis Approach to Thermal Placement **163**
  C. C.-N. Chu and D. F. Wong
- Preserving HDL Synthesis Hierarchy for Cell Placement **169**
  Y.-W. Tsay, W.-J. Fang, A. C.-H. Wu and Y.-L. Lin

Session 8: Poster Session Discussion

AMD EX1021
AMD v. Realtek - Page 381

Session 9: Performance Optimization

Moderators: W. W.-M. Dai (UCSC)
L. Jones (Motorola)

- EWA: Exact Wire Sizing Algorithm **178**
  R. Kay, G. Bucheuv and L. Pileggi
- Minimization of Chip Size and Power Consumption of High Speed VLSI Buffers **186**
  D. Zhou, X. Y. Liu and X. L. Wang
- Closed Form Solution to Simultaneous Buffer Insertion/Sizing and Wire Sizing **192**
  C. C.-N. Chu and D. F. Wong

Session 10: Design Methodology Futures (Invited)

- Chip Hierarchical Design System (CHDS): A Foundation for Timing-Driven Physical Design into the 21st Century
  R. G. Bushroe, S. DasGupta and R. Steele
- Physical Design 2010: Back to the Future?
  A. R. Newton

Invited Address

- Physical Design Realities for Digital's StrongARM and Alpha Microprocessors
  W. J. Grundmann

Session 11: Core Directions (or, Do The Right Thing) (Invited)

- Physical Design Challenges of Performance
  D. P. LaPotin
- **Panel:** Physical Design R&D: What's Missing?
  G. Smith, Moderator; W. W.-M. Dai, E. Hsieh, M. Hunt, K. Keutzer, D. P. LaPotin and N. Sherwani

Invited Papers

- Physical Design: Reminiscing and Looking Ahead **206**
  E. S. Kuh
- Physical Design: Mathematical Models and Methods **207**
  T. C. Hu
- The Quarter Micron Challenge: Integrating Physical and Logic Design **211**
  R. Camposano
- Chip Hierarchical Design System (CHDS): A Foundation for Timing-Driven Physical Design into the 21st Century **212**
  R. G. Bushroe, S. DasGupta, A. Dengi, P. Fisher, S. Grout, G. Ledenbach, N. S. Nagaraj and R. Steele
- The Future of Logic Synthesis and Physical Design in Deep Submicron **218**
  K. Keutzer, A. R. Newton and N. Shenoy
- Physical Design Challenges of Performance **225**
  D. P. LaPotin, U. Ghoshal, E. Chiprout and S. R. Nassif

Author Index **227**

AMD EX1021
AMD v. Realtek - Page 382

A Pseudo-Hierarchical Methodology for High Performance Microprocessor Design

A. Bertolet, K. Carpenter, K.Carrig, A. Chu, A. Dean, F. Ferraiolo, S.Kenyon, D. Phan, J. Petrovick, G. Rodgers, D. Willmott
T.Bairley†, T. Decker†,V.Girardi†, Y. Lapid†, M. Murphy†, P. A. Scott†, R.Weiss†
IBM Corporation, Essex Junction, Vermont
†Cadence Design Systems, San Jose, California

***Abstract* - This paper reports on a highly effective methodology to construct complex high performance microprocessors. Critical aspects of the methodology include an integrated database for design control, algorithmic power grid generation, fully customized clock network insertion, timing driven placement and routing, an integrated timing closure strategy, and incremental checking that includes formal netlist verification, DRC and LVS. The methodology places particular emphasis on continuously improving the integration process and incrementally improving both the design and the interoperability of the tools. The final chip tape-out was 17 calendar days from the final netlist.**

I. INTRODUCTION

The current generation of high performance superscaler microprocessors pushes chip transistor counts of 3-5 million. The architectural complexity and cycle time requirements create critical challenges in clock distribution, robust power distribution, timing driven design, engineering change management, and full-chip verification. To deal with these challenges, a design methodology was crafted to rapidly and repeatedly integrate the chip while providing fully customizable clock and power distribution and an automated engineering change (EC) facility to deal with last minute timing and logic corrections. A "pseudo-hierarchical" design approach was applied to manage the floorplan, placement, and routing optimization of the standard cell control logic. The methodology is an evolution from a traditional hierarchical approach and results in a die size reduction of 96mm^2 on the same netlist.

The methodology was applied to a highly superscaler PowerPC™ microprocessor. The chip was fabricated in an 0.35µm (L_{eff} = 0.25µm), 2.5V CMOS n-well technology with six levels of metal: five for global routing and one for local interconnect. The die, as depicted in Figure 1, is 10.4mm X 14.4mm and contains 6.5 million transistors. The chip includes 48KByte of cache and 121 custom macrocells. Random control logic is implemented with approximately 67,000 standard cells instantiated at the top level of the hierarchy. In addition, there are approximately 32,000 master-slave latches (25,300 of which are in custom macrocells) that require a highly tuned clock distribution network.

Permission to make digital/hard copies of all or part of this material for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication and its date appear, and notice is given that copyright is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires specific permission and/or fee
ISPD'97 Napa Valley, California USA
Copyright 1997 ACM 0-89791-927-0/97/04 ..$3.50

The design environment consists of third-party and proprietary IBM CAD tools interfaced to Cadence Design Framework II™ (DFII). DFII is used for data and design process management. Using the DFII extension language, SKILL™, enables the team to customize and extend the environment as needed.

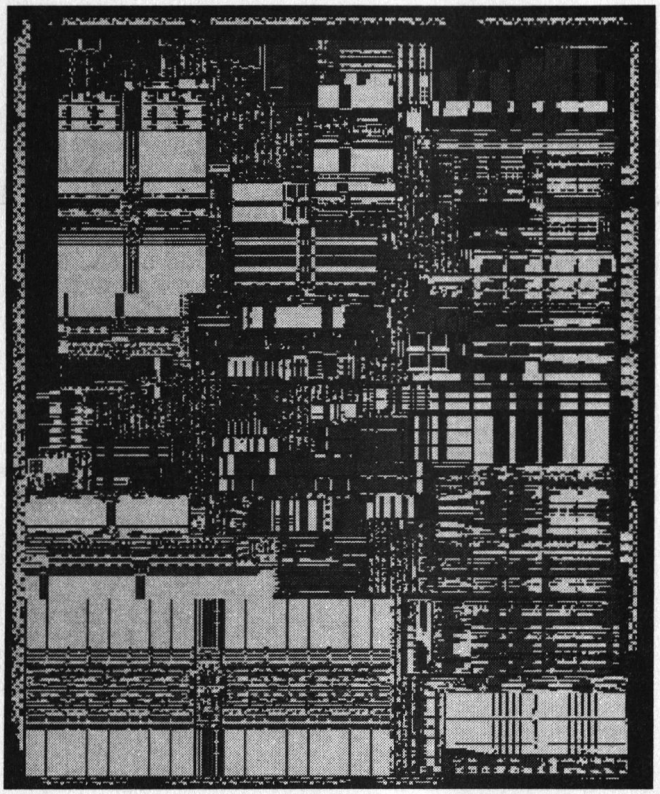


Figure 1. Chip layout (metal layers not shown)

II. METHODOLOGY OVERVIEW

The cornerstone of the design methodology is the ability to rapidly and repeatedly integrate and verify the chip. The notion of "construct by correction" was utilized to repeatedly integrate the chip over a three month period. Each iteration incrementally improved the overall quality of the design, and enabled real-time feedback of parasitic RC values to the timing tool. In order to iterate on the design, a consistent design view of VHDL, synthesized netlist, and physical design data is essential. As such, facilities were developed to enforce consistency by creating incomplete or missing physical design data such that it matched exactly the specifications of the incoming netlist (Figure 2). This enabled floorplanning, placement, and routing to continue, despite errors, throughout the earlier phases of the integration

This material may be protected by Copyright law (Title 17 U.S. Code)
AMD EX1021
AMD v. Realtek - Page 383

process to provide valuable timing feedback with routing parasitics. Any problems encountered are corrected concurrently and captured on the next pass through the process. In addition, techniques were developed to make early full-chip DRC and LVS assessments without final routing data. The net result of the incremental process was a final netlist to tapeout cycle of 17 days, which included two EC passes, a crosstalk assessment, and a fast-path assessment.

Despite the fact that the focus of the methodology is on iterative improvement and rapid integration, technical capabilities are not sacrificed. The final integration cycle includes a fully custom, algorithmically generated power distribution on five layers of metal that includes legal standard cell core site definition based on power analysis. In addition, a fully customized clock distribution network is generated with a resultant typical skew of 180pS. Furthermore, multiple routing passes are performed to handle special classes of nets. The methodology focuses on tight integration of these sophisticated capabilities to maintain quality and productivity.

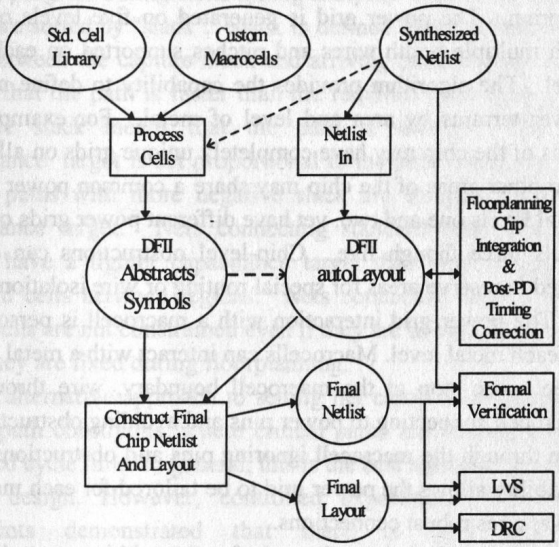


Figure 2 Methodology Overview

III. DATABASE AND METHODOLOGY CHECKS

The chip integration methodology discussed above uses the DFII database to represent all chip design views from the original synthesized netlist to final chip. The DFII database models both the logical and physical structure of the chip so all changes could be represented.

There are two key components to the database:

1. The DFII autoLayout view contains full-chip netlist, placement and global routing information.
2. The DFII abstract views for macrocell and standard cell components (essential placement and routing data derived from the layout views).

Critical to the "construct by correction" method is the ability to rapidly integrate incrementally improved versions of the design. A facility called "Process Cells" manages and enforces design consistency. The full-chip synthesized netlist is the "golden" chip representation. Process Cells analyzes the netlist and checks for a unique symbol in the standard cell or macrocell library referenced by each instance in the netlist. If a symbol is missing, or has incorrect terminals, a symbol is created which matches the terminals required in the netlist. The symbols are an important link to the original netlist, since they will be referenced to create a full-chip schematic netlist for final verification. Next, an inventory of layouts that corresponds to the symbols is built. If a layout is missing, a "black box" consisting of a place and route boundary can be generated automatically or by the macrocell designer. Finally, an abstract is created from each layout. During abstract generation, the symbol and layout are examined for consistency. Terminals in the layout but not in the symbol are deleted, while terminals in the symbol but missing from the layout are created. At this point, Process Cells has ensured that a consistent set of symbols and abstracts are available to construct a full-chip autoLayout view of the synthesized netlist, and to proceed with floorplanning, placement, and routing. This innovative approach allows floorplanning, placement, and routing to occur with *no* layout, symbol, or schematic cellviews, and to incrementally add them to the chip integration process as they are completed.

In order to avoid problems in the abstract generation process, macrocell designers are provided with a methodology checking program. This program ensures that layouts meet guidelines established for placement and routing. The key checks are:

- Pin names match names found in the netlist.
- Pins are on-grid.
- Pins are on correct layer and accessible to router.
- Manufacturable shapes are enclosed by a place and route boundary by half the minimum design rule space.
- Clock pin(s) meets the clock router methodology.

Process Cells provides an important link to the macrocell design team. The system is highly automated and is designed to tolerate problems in the macrocells, including violations of the methodology checks. When a problem is identified, it is communicated to the macrocell design team for resolution while chip integration proceeds. When a new type of error is found, enhancements are made to Process Cells so that future chip integration runs can tolerate such an error, and the methodology checks are enhanced to prevent reoccurrence of the error. Hence, each iteration of the methodology results in faster processing of the chip.

Once abstracts are generated and the netlist imported into DFII, chip integration can take place. The remainder of the paper details the various chip integration operations summarized in Figure 3

IV. FLOORPLANNING

During floorplanning, the macrocell blocks (i.e. cache and dataflow macrocells) are pre-placed at fixed locations while the standard cells are optimized with a "pseudo-hierarchical" floorplanning approach.

AMD EX1021
AMD v. Realtek - Page 384

Pseudo-hierarchical floorplanning is a technique whereby the design can be hierarchically planned and timed without incurring any of the physical design complexities caused by doing true hierarchical floorplanning, placement and routing. It also minimizes potential global suboptimizations due to local optimizations inherent in a hierarchical design approach.

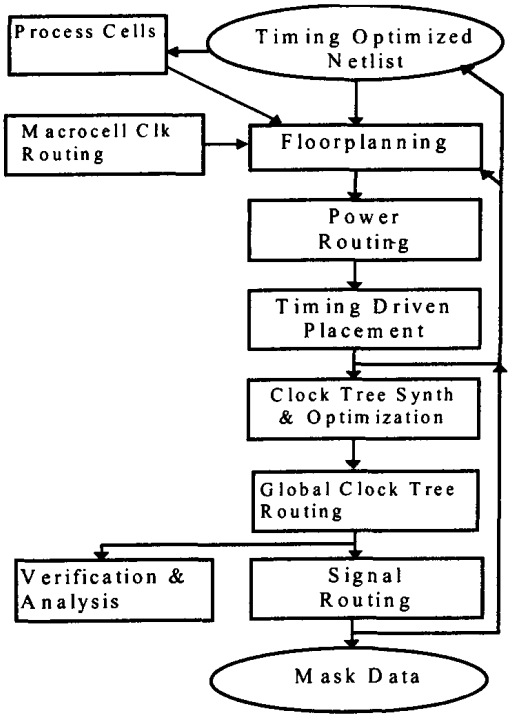


Figure 3 Chip Integration Process

Hierarchical designs have certain advantages over flat designs: early timing prediction, design complexity management, design size management, and concurrent design. However, difficulties arise from concurrent design of the global and lower levels of the hierarchy. Routing resource and area allocations between levels of hierarchy must be carefully and constantly managed to prevent local routing congestion and area growth. Additional problems exist in optimal soft/hard-block pin placement, global area optimization with rectangular soft/hard-blocks, and the timing optimization of critical paths which span multiple levels of the hierarchy.

This pseudo-hierarchical floorplanning approach assigns hierarchical entities of random logic to overlapping regions in the floorplan. The region controls provide a good estimate of the global route lengths for early timing prediction and analysis. Region size, location, and composition are optimized to meet the cycle-time objective (timing closure). Region composition is iteratively determined by analyzing timing constraints and connectivity constraints. Due to tool integration issues, true timing driven floorplanning is not used. Instead, the floorplan is incrementally updated with placement results. The members of each region are color coded so that the floorplanner can graphically analyze the interaction of different regions and interactively modify region size, location, and composition. In this design, 45 logic entities specified in the VHDL are dissolved and reconstituted as 23 tightly integrated regions. Overlapping regions allow cell intermixing between adjacent VHDL logic entities to minimize wire length and congestion. Logic function that is not critical to the timing of the design, such as the self-test control logic, is placed in a region located in the least congested area of the die. Once all logic macrocells have been assigned to regions (more than one macrocell can be assigned to the same region), a flat placement and routing of the entire design can be completed in a matter of a few hours and fed back to the floorplan.

This pseudo-hierarchical floorplanning approach couples the timing control of a hierarchical design with the advantage of a flat design in placing the standard cells in non-rectangular areas. This "pouring" of the standard cells into the natural outline of large pre-designed macrocells yields higher density and performance.

V. POWER ROUTING

After custom macrocell placement, a power distribution network is algorithmically generated using a custom SKILL program. The power grid is generated on five levels of metal with multiple width wires and pitches supported on each metal level. The algorithm provides the capability to define multiple power terrains by area and level of metal. For example, two areas of the chip may have completely unique grids on all layers. Two other areas of the chip may share a common power grid on metal levels one and two, yet have different power grids on metal levels three though five. Chip-level obstructions can also be added to reserve areas for special routing or wire isolation.

The power grid interaction with a macrocell is personalized for each metal level. Macrocells can interact with a metal level in three ways: stop at the macrocell boundary, wire through the macrocell connecting to power pins and avoiding obstructions, or wire through the macrocell ignoring pins and obstructions. This flexibility allows the power grid to be tailored for each macrocell and assures robust connections.

Power analysis is performed after the grid is personalized. If large IR drops are discovered because of truncation at macrocell boundaries, the underpowered rails are removed from the network. This guarantees that all remaining areas of the power network are adequately connected.

Standard cell core placement sites are now added to the floorplan. By design convention, all standard cells make power connections on the metal-one layer. The algorithm searches the list of all metal one rails and adds core sites in open areas that have the required first metal power rails. The IR drop analysis guarantees that the generated core sites have sufficient power connections and no further analysis is required. The chip is ready for standard cell placement.

VI. TIMING DRIVEN PLACEMENT

The objective of standard cell placement during floorplanning is to provide accurate feedback to timing analysis for iterative improvement of region composition and macrocell placement. Only region constraints are employed during these

AMD EX1021
AMD v. Realtek - Page 385

iterations to ensure fast turn-around-time. Once the floorplan is optimized, the placement is further refined by asserting timing constraints to improve critical path delay. This design flow consists of multiple iterations of placement, global routing, parasitic extraction, and timing analysis (Figure 4). During each iteration, the cycle time is measured using the extracted net capacitance and RC. Drive strength of standard cells is readjusted to match the net loading, and the design is re-timed with the newly selected cells. A set of critical paths is identified from this timing run and a new set of tighter capacitance targets is generated to steer placement to a more optimal solution. These steps are repeated until the target cycle time is met or no further improvement is possible. The placement engine, Cadence QPlace™, supports various forms of timing constraints with minimal impact on run time. Net capacitance constraints are applied to nets in the critical paths. This directs QPlace to work harder to minimize these nets, thus improving the chip operating frequency.

The net capacitance targets are generated by invoking a custom program during static timing analysis. Nets in the critical paths are sorted by "slack". Slack is defined as the difference in time between the capture clock and arriving data; positive slack means that the path is faster than the required cycle time while negative slack means that the path is slower. The net capacitance target is set proportional to the path slack, i.e. nets in the paths with more negative slack are assigned a tighter capacitance target. Nets connecting standard cells within a region have a tighter capacitance target than nets connecting standard cells between regions. Nets connecting large custom macrocells are not constrained even if they are in the critical path since they are fixed during floorplanning.

An alternative approach to setting net capacitance targets is to use path constraints, where critical paths are defined with a specified cycle time. In general, this is the best method for timing driven design. However, controlled experiments with path constraints demonstrated that there is only marginal improvement over the net capacitance constraint method for this design. This is due to the iterative refinement of the net capacitance targets over many chip integration passes. Additionally, path constraints are more disruptive to the methodology since the timing analysis tool is not tightly integrated with the placement tool.

The final stage of placement exercises an algorithm which inspects the entire chip for n-well contacts to the power supply. The algorithm optimally inserts n-well contact cells in the standard cell rows to meet the design rule specification.

VII. CLOCK METHODOLOGY

Clock design is a major part of the overall chip integration methodology and often a critical path to a fast design turn-around time. The hierarchical clock design methodology used here streamlines this process while minimizing clock skew and power.

The system clock distribution network contains three stages, a large global clock buffer (GCB), approximately 20 regional clock buffers (RCBs) and several hundred local clock buffers (LCBs). Control signals on the LCB inputs are used for clock gating and test clock generation. A single phase of the clock is distributed to the inputs of all LCBs. The LCBs generate two pairs of complementary clocks, one pair to the master latches and the other to the slave latches. The portion of clock trees within the custom macrocells are designed and routed as part of the macrocell layouts. The clock interface to a macrocell layout consists of only one input pin. This simplifies global (top-level) clock routing and helps identify inaccessible clock pins. Variable width clock routing of macrocells occurs concurrently with floorplanning so that chip-level information is used to guide the optimization of the macrocell clock tree.

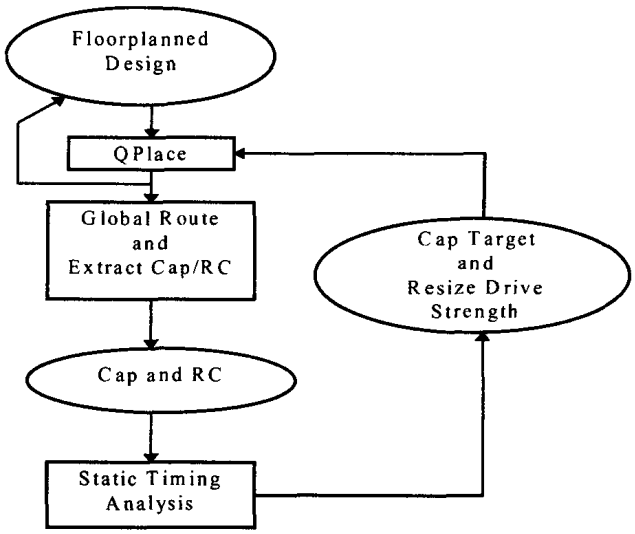


Figure 4 Timing Driven Design Iteration

After floorplanning, placement and power routing are complete, clock tree synthesis and optimization are performed at the global chip level to generate and place clock buffers. This is accomplished using IBM CO2, a tool that traverses the clock trees, identifying and reconfiguring equivalent nets as well as adding parallel copies of buffers to minimize clock skew [1]. The LCBs at the leaf nodes of the clock tree are placed at the RC centroid of the latch cluster to minimize RC induced skews. All the buffers are then snapped to legal placement locations and any cell overlaps resolved. Using IBM CLOCKTREE, a two-layer balanced router that can vary the width of each wire segment, clock routing is then performed on all but the lowest (local clock buffer to latch) levels in the clock tree; the local clock buffer to latch levels are routed with minimum width wires using Cadence Cell3™.

Once the clock design is complete, signal routing takes place. At the same time, the clock nets are extracted, verified and analyzed to ensure the skew objective is met. To ensure a functional design, early mode (fast path) analysis using the extracted SPICE netlist is performed and any problems are fixed either by replacing the LCB with one that has a late launch clock or by adding delays to the fast paths. The capacitance for the entire network including macrocells is 558 pF and the typical clock skew is 180 pS [2]. This compares favorably with the mesh

AMD EX1021
AMD v. Realtek - Page 386

scheme, where sizable networks having capacitance values of 1400 pF and 2000 pF (for 2 different chips) were required to achieve similar skews [3].

VIII. ROUTING

Once power routing, placement and clock-tree generation are complete, signal routing is carried out. Because five levels of metal were available for routing, congestion was not a major concern for this design. In addition, it was found through iterative analysis that wiring parasitics extracted after global routing were consistently within 5% of the final routed parasitics, insuring that the cycle time of the routed design could meet the prediction. Therefore, the primary issue during routing was on-chip signal coupling (crosstalk).

Factors contributing to crosstalk include the interaction of five levels of metal for routing, circuits on a net with small output drive devices or pass-gate input structures, fast signal slew rates, and the large die size which leads to long parallel nets with only an 0.9μm separation. Crosstalk avoidance therefore must be considered as part of the overall routing methodology.

One ideal attribute of a router is the ability to route nets such that the simultaneous switching of adjacent signals will not cause signals to fail noise margin guidelines. However, due to existing tool limitations, a modified wide-wire routing technique is adapted to isolate noise-sensitive signals from other nets. This approach is selected because of the availability of routing resources. With simulations and previous hardware indicating that major crosstalk problems occur between signals that are one track away from each other on the same level of metal (for this fabrication technology), noise-sensitive nets are routed first as triple-wide wires. All timing-critical nets are then routed, followed by the remaining nets in the design. The triple-wide wires are then resized back to their default (single) widths, creating an effective isolation region to other nets.

IX. EC METHODOLOGY

An essential component of the methodology is an EC process for making complex changes to the design without restarting chip integration. An ideal EC process handles an arbitrarily complex change, locates the new logic at the point of failure, impacts a minimum number of manufacturing levels, and does not effect chip cycle-time or area. This methodology addresses all aspects of the ideal EC process, while focusing on two areas in particular:

a) The ability to make complex logic changes using metalization layers only.
b) A technique to add gates as close as possible to connected logic in order to reduce any impact on cycle time.

To facilitate the process, a small EC library consisting of the following gates is created: inverter, NAND, NOR, AND-OR, delay line, terminator (capacitive load for late balancing of clocks), tie-up/down (to tie off gates disabled by an EC), and a latch. These gates are designed with a two-channel gate array primitive (unpersonalized) cell which contains two NMOS and two PMOS transistors, and is personalized only with metal. The EC gates are the same height as, and abut to, the standard cells and can be located in any legal placement site.

At the conclusion of the integration process, at the time the full-chip autoLayout view is converted to a layout view, gate array primitive cells are placed in all unoccupied standard cell placement locations on the chip. EC logic can subsequently be placed at any of these locations.

During final chip integration, one late EC required the transformation of a single cycle path to a half cycle path, the removal of approximately six gates and the addition of approximately twenty gates. The chip cycle time was maintained due to the richness of the EC library and the ability to place the EC elements in close proximity to the change.

X. DESIGN VERIFICATION

Key to the success of delivering working silicon on an abbreviated schedule is a method to quickly verify the final mask design against the high-level language description of the chip, which is verified by exercising billions of logic simulation cycles. This process is accomplished using formal verification techniques developed at IBM.

Once the VHDL description is verified, a reference model is generated using IBM's logic synthesis system [4]. After synthesis and timing optimization, formal verification is performed against the reference model to verify that logic and timing correction transforms maintained Boolean equivalence with the reference model [5]. This process also verifies that the many logic changes applied during the synthesis and timing correction process are properly incorporated into the design (Figure 5).

After physical design, the resulting schematic netlist is formally verified against the timing-optimized model using the same Boolean equivalency checking software. This process is critical in verifying:

1. Correct operation of data translation from the IBM synthesis system to the Cadence design framework.
2. Logic changes made during the physical design process, such as fanout repowering and clock distribution, are logically equivalent to the original design.
3. Last minute engineering changes (to fix functional bugs or critical path timing problems) are applied correctly.

To complete the formal verification cycle, custom macrocells instantiated in the schematic netlist are formally verified against their VHDL descriptions [6,7].

As a result of these processes and software described above, formal verification insures that the final schematic netlist used for full-chip logic-versus-schematic (LVS) verification is logically equivalent to the VHDL description of the chip.

Due to chip data volume and turnaround time considerations, Avant!™ Vericheck™, a hierarchical physical verification tool for full chip design rule checking (DRC) and LVS checking, is used. Coupled with a macrocell auditing scheme that ensures

AMD EX1021
AMD v. Realtek - Page 387

DRC and LVS correct standard cells and custom macrocells, hierarchical checking proves extremely advantageous in diagnosing full-chip design errors very quickly.

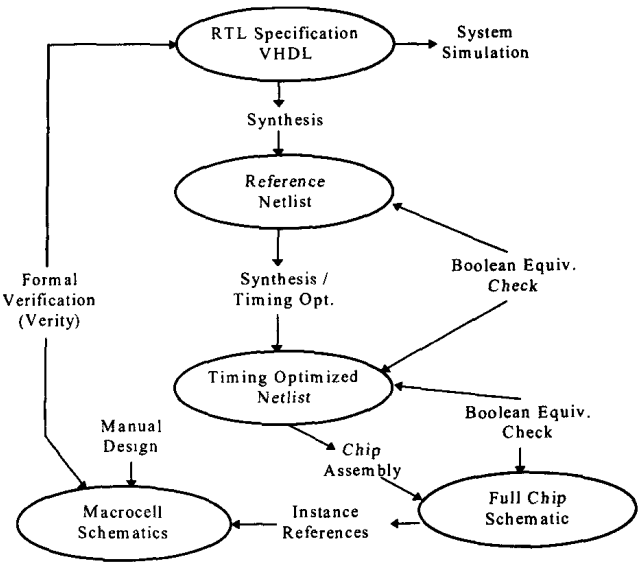


Figure 5 Verification Flow

Critical to the requirement for rapid turnaround time in the "construct by correction" method are two techniques developed to reduce the number of errors caught late in the design cycle.

First, DRC checking is executed on chips of increasing completeness. For example, an initial run is made on a chip that includes only the I/O and the power distribution network. Later, another run is made on a complete floorplan with all custom macrocells placed but with no global routing. With each iteration, more and more of the methodology and floorplan related errors are eliminated.

A second, more important technique is used to provide LVS results on a fully placed, but unrouted chip. Virtual nets are created by attaching text to the terminals of each instance in the design. The text identifies the net that would be attached to each terminal later by the global router. This approach provides critical full-chip LVS results many weeks earlier than the traditional approach.

XI. SUMMARY

This paper presented a robust design methodology for high performance microprocessors. The methodology focused on "construct by correction" to iteratively improve the design. The pseudo-hierarchical approach provides the benefits of hierarchical design in terms of timing prediction yet suffers none of the shortcomings of hierarchical design management and area suboptimizations. Indeed, the methodology succeeds by postponing critical design decisions to as late in the design cycle as possible. Regions evolve during the design cycle, rather than being predefined at the start, and placement and routing are globally optimized. The methodology was successfully implemented on a PowerPC microprocessor and is easily extensible to other high performance, highly integrated products, such as ASICs.

XII. ACKNOWLEDGMENTS

The authors would like to acknowledge the contributions of numerous colleagues whose efforts in many diverse areas made this work a success. In particular the authors would like to recognize the efforts of Dave Hathaway, Pete Osler, Betty Bouldin, Tad Wilder, John Doyle, Bruce Winter, Andrew Tickle, Gilles Lamant, Marc Bevis, Andrew Davis, Damian Artt, Don Perley, and Bob Malkemes.

Cell3, Design Framework II, QPlace and SKILL are trademarks of Cadence Design Systems, Inc. Vericheck is a trademark of Avant!, Inc. All other trademarks are owned by their respective holders.

REFERENCES

[1] D. J. Hathaway, R. R. Habra, E. C. Schanzenbach, and S. J. Rothman, "Circuit Placement, Chip Optimization, and Wire Routing for IBM IC Technology," IBM Journal of Research and Development, vol. 40, no. 4, 1996, pp. 453-460.

[2] K. M. Carrig, A. M. Chu, F. D. Ferraiolo, J. G. Petrovick, P. A. Scott, and R. J. Weiss, "A Clock Methodology for High Performance Microprocessors," to be published in Proceedings of the IEEE Custom Integrated Circuits Conference, May 1997.

[3] M.P. Desai, R. Cvijetic and J. Jensen, "Sizing of Clock Distribution Networks for High Performance CPU Chips," in Proc. Design Automation Conf., June 1996, pp. 389-394.

[4] D. Brand, R. Damiano, L.van Ginniken, and A. Drumm, "In the Driver's Seat of BooleDozer," Proceedings of the IEEE International Conference on Computer-Aided Design, October 1994, pp. 518-521.

[5] D. Kung, R. Damiano, T. Nix, and D. Geiger, "BDDMAP: A Technology Mapper Based on a New Covering Algorithm," Proceedings of the 29th ACM/IEEE Design Automation Conference, June 1992, pp. 484-487.

[6] D. Appenzeller, A Kuehlmann, "Formal Verification of a PowerPC Microprocessor," Proceedings of the IEEE International Conference on Computer Design, October 1995, pp. 79-84.

[7] A. Kuehlmann, A. Srinivasan, and D. LaPotin, "Verity - A Formal Verification Program for Custom CMOS Circuits," IBM Journal of Research and Development, vol. 39, pp. 149-165, January/March 1995.

AMD EX1021
AMD v. Realtek - Page 388

EXHIBIT 7A

```
001    994874143402947
005    20230311151806.0
008    990526t19971997nyua     b    101 0 eng d
020    ##$a0897919270
035    ##$a(OCoLC)ocm41436664
040    ##$aGAT $beng $cGAT
049    ##$aGATT
090    ##$aTK7874.75 $b.I58 1997
111    2#$aInternational Symposium on Physical Design $n(1st :
$d1997 : $cNapa Valley, Calif.)
245    10$aProceedings, 1997 International Symposium on Physical
Design : $bISPD-97 : Napa Valley, CA, April 14-16, 1997 :
proceedings / $csponsored by ACM Special Interest Group on Design
Automation (SIGDA) ; in cooperation with IEEE Circuits and Systems
Society.
246    30$a1997 International Symposium on Physical Design
246    30$aISPD-97
264    #1$aNew York, N.Y. : $bAssociation for Computing Machinery,
$c[1997]
264    #4$c©1997
300    ##$aix, 228 pages : $billustrations ; $c28 cm
336    ##$atext $btxt $2rdacontent
337    ##$aunmediated $bn $2rdamedia
338    ##$avolume $bnc $2rdacarrier
500    ##$a"ACM order number: 477975"--T.p. verso.
504    ##$aIncludes bibliographical references and index.
650    #0$aIntegrated circuits $xVery large scale integration.
$0http://id.loc.gov/authorities/subjects/sh85067125
650    #0$aIntegrated circuits $xDesign and construction.
$0http://id.loc.gov/authorities/subjects/sh85067118
655    #7$aConference papers and proceedings. $2lcgft
$0http://id.loc.gov/authorities/genreForms/gf2014026068
710    2#$aAssociation for Computing Machinery.
$0http://id.loc.gov/authorities/names/n78096869
710    2#$aACM Special Interest Group on Design Automation.
$0http://id.loc.gov/authorities/names/n85271173
710    2#$aIEEE Circuits and Systems Society.
$0http://id.loc.gov/authorities/names/n79143117
935    ##$a(GAT)487414-gtechdb-Voyager $9local
945    ##$aMARCIVE20210914
950    ##$agatech
```

ATTACHMENT 7B

OCLC 41436664 **No holdings in XXX - 6 other holdings**

Books

						Rec Stat		**Entered**	19990526	**Replaced**	20221025213749.4
Type	a	**ELvl**		**Srce**	d	**Audn**		**Ctrl**		**Lang**	eng
BLvl	m	**Form**		**Conf**	1	**Biog**		**MRec**		**Ctry**	nyu
		Cont	b	**GPub**		**LitF**	0	**Indx**	1		
Desc	a	**Ills**	a	**Fest**	0	**DtSt**	s	**Dates**	1997,		

040 GAT $b eng $c GAT $d OCL $d OCLCF $d OCLCQ $d OCLCO $d OCL $d OCLCO $d OCLCA $d OCLCQ $d OCLCO $d OCLCQ
020 0897919270
020 9780897919272
050 4 TK7874.75 $b .I58 1997
090 $b
111 2 International Symposium on Physical Design $n (1st : $d 1997 : $c Napa Valley, Calif.)
245 10 Proceedings, 1997 International Symposium on Physical Design : $b ISPD-97 : Napa Valley, CA, April 14-16, 1997 : proceedings / $c sponsored by ACM Special Interest Group on Design Automation (SIGDA) ; in cooperation with IEEE Circuits and Systems Society.
246 30 1997 International Symposium on Physical Design
246 30 ISPD '97
260 New York, N.Y. : $b Association for Computing Machinery, $c ©1997.
300 ix, 228 pages : $b illustrations ; $c 28 cm
336 text $b txt $2 rdacontent
337 unmediated $b n $2 rdamedia
338 volume $b nc $2 rdacarrier
500 "ACM order number: 477975"--Title page verso
504 Includes bibliographical references and index.
530 Also available on the World Wide Web via ACM Digital Library with title: Proceedings of the 1997 International Symposium on Physical Design.
650 0 Integrated circuits $x Very large scale integration $v Congresses.
650 0 Integrated circuits $x Design and construction $v Congresses.
650 6 Circuits intégrés à très grande échelle $0 (CaQQLa)201-0117255 $v Congrès. $0 (CaQQLa)201-0378219
650 6 Circuits intégrés $0 (CaQQLa)201-0363604 $x Conception et construction $0 (CaQQLa)201-0363604 $v Congrès. $0 (CaQQLa)201-0378219
650 7 Integrated circuits $x Design and construction. $2 fast $0 (OCoLC)fst00975545
650 7 Integrated circuits $x Very large scale integration. $2 fast $0 (OCoLC)fst00975602
655 7 Conference papers and proceedings. $2 fast $0 (OCoLC)fst01423772
710 2 Association for Computing Machinery.
710 2 ACM Special Interest Group on Design Automation.
710 2 IEEE Circuits and Systems Society.
740 0 Proceedings of the 1997 International Symposium on Physical Design.
856 41 $u http://www.acm.org/pubs/contents/proceedings/dac/267665/ $z Full text of the Conference proceedings (1997)

ATTACHMENT 7C

LIBRARY OF CONGRESS AUTHORITIES

LC control no.: sh2008104738

LCCN Permalink: https://lccn.loc.gov/sh2008104738

HEADING: Integrated circuits Design and construction Congresses

000 00566cz a2200169n 450

001 7450858

005 20110729160326.0

008 080211|| anannbabn |n ana

010 __ |**a** sh2008104738

035 __ |**a** (DLC)381541

035 __ |**a** (DLC)sh2008104738

040 __ |**a** DLC |**b** eng |**c** DLC

150 __ |**a** Integrated circuits |**x** Design and construction |**v** Congresses

667 __ |**a** Record generated for validation purposes.

670 __ |**a** Work cat.: Proceedings of the ... International Workshop on Design of Mixed-Mode Integrated Circuits and Applications,

906 __ |**t** 8888 |**u** te00 |**v** 0

953 __ |**a** te00

◀ Previous Next ▶

Save, Print or Email Records (View Help)

Select Download Format: Text (Labelled Display) Press to SAVE or PRINT

Email Text (Labelled Display) to: Press to SEND EMAIL

Library of Congress
URL: https://www.loc.gov/

Mailing Address:
101 Independence Ave, S.E.
Washington, DC 20540

Library of Congress Authorities
URL: http://authorities.loc.gov/
Library of Congress Online Catalog
URL: https://catalog.loc.gov/

Questions, comments, error reports: Contact Us

AMD EX1021
AMD v. Realtek - Page 394

ATTACHMENT 7D

Copyright United States Copyright Office

Try the Copyright Public Records System (CPRS) pilot with enhanced search features and filters.

Help | Search | History | Titles | Start Over

Public Catalog

Copyright Catalog (1978 to present)
Search Request: Left Anchored Title = ISPD '97
Search Results: Displaying 1 of 1 entries

previous next

Labeled View

1997 international symposium on physical design : ISPD-97 : Napa Valley,...

Type of Work: Text
Registration Number / Date: TX0005077849 / 1999-10-25
Title: 1997 international symposium on physical design : ISPD-97 : Napa Valley, CA, April 14-16, 1997 : proceedings.
Imprint: New York : Association for Computing Machinery, c1997.
Description: 228 p.
Copyright Claimant: Association for Computing Machinery, Inc.
Date of Creation: 1997
Date of Publication: 1997-03-31
Other Title: ISPD-97
International symposium on physical design
Names: Association for Computing Machinery, Inc.

previous next

Save, Print and Email (Help Page)	
Select Download Format	Full Record ▾ Format for Print/Save
Enter your email address:	Email

Help Search History Titles Start Over

Contact Us | Request Copies | Get a Search Estimate | Frequently Asked Questions (FAQs) about Copyright | Copyright Office Home Page | Library of Congress Home Page

ATTACHMENT 8

Proceedings of the

ASP-DAC 2003

Asia and South Pacific Design Automation Conference
2003

January 21-January 24, 2003
Kitakyushu International Conference Center
Kitakyushu, Japan

Sponsored by:

IEEE Circuits and Systems Society
ACM SIGDA
IEICE (Institute of Electronics, Information and Communication Engineers)
IPSJ (Information Processing Society of Japan)

Supported by:

JEITA (Japan Electronics and Information Technologies Industry Association)
STARC (Semiconductor Technology Academic Research Center)
Kitakyushu City
Kitakyushu Convention Bureau
ASIAS KYUSHU (Promotion Council of Northern Kyushu Science Cities' Development Plan)
Promotion Committee for Fukuoka System LSI R&D Hub

In Corporation with:

JIEP (Japan Institute of Electronics Packaging)

AMD EX1021
AMD v. Realtek - Page 398

Additional Copies may be ordered from:

IEEE Order Dept.
Hoes Lane
P.O. Box 1331
Piscataway, NJ 08854, U.S.A.

Copyright and Reprint Permission: Abstracting is permitted with credit to the source. Libraries are permitted to photocopy beyond the limit of U.S. copyright law for private use of patrons those articles in this volume that carry a code at the bottom of the first page, provided the per-copy fee indicated in the code is paid through Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923. For other copying, reprint or republication permission, write to IEEE Copyrights Manager, IEEE Operations Center, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331. All rights reserved. Copyright ©2003 by the Institute of Electrical and Electronics Engineers, Inc.

IEEE Catalog Number: 03EX627 (CD-ROM Version: 03EX627C)
ISBN: 0-7803-7659-5 (CD-ROM Version: 0-7803-7660-9)
Library of Congress: 2002112993

ASP-DAC 2003 Best Paper Award Candidates

1D-1 **SAT-based Sequential Depth Computation**
M. Mneimneh, K. Sakallah – *University of Michigan, USA*

2B-1 **Approximate Formulae Approach for Efficient Inductance Extraction**
A. Kurokawa – *Semiconductor Technology Academic Research Center, Japan,* ***T. Sato*** – *Hitachi Ltd., Japan,* ***H. Masuda*** – *Semiconductor Technology Academic Research Center, Japan*

3A-2 **Towards On-Chip Fault-Tolerant Communication**
T. Dumitras, S. Kerner, R. Marculescu – *Carnegie Mellon University, USA*

3C-1 **Statistical Delay Computation Considering Spatial Correlations**
A. B. Agarwal, D. Blaauw – *University of Michigan, USA,* ***S. Sundareswaran, V. Zolotov, M. Zhao, K. Gala, R. Panda*** – *Motorola,USA*

4C-1 **Efficient LUT-Based FPGA Technology Mapping for Power Minimization**
H. Li, W.-K. Mak, S. Katkoori – *University of South Florida, USA*

5A-1 **Fast Buffer Planning and Congestion Optimization in Interconnect-driven Floorplanning**
K. W. C. Wong, E. F.Y. Young – *The Chinese University of Hong Kong, Hong Kong*

5D-2 **Design of a Scalable RSA and ECC Crypto-Processor**
M.-C. Sun, C.-P. Su, C.-T. Huang, C.-W. Wu – *National Tsing Hua University,Taiwan*

7A-1 **Run-Time Energy Estimation in System-On-a-Chip Designs**
J. Haid, G. Kaefer, C. Steger, R. Weiss – *Graz University of Technology, Austria*

7D-2 **A comparison of the RTU hardware RTOS with a Hardware/Software RTOS**
J. Lee, V. J. Mooney III – *Georgia Institute of Technology, USA,* ***A. Daleby, Karl Ingstrom*** – *Malardalens University, Sweden,* ***T. Klevin, Lennart Lindh*** – *Malardalens University / RealFast, Sweden*

8B-1 **Congestion Driven Incremental Placement Algorithm for Standard Cell Layout**
Z. Li, W. Wu, X. Hong – *Tsinghua University, China*

8C-1 **Experience in Critical Path Selection For Deep Sub-Micron Delay Test and Timing Validation**
J.-J. Liu, L.-C. Wang, A. Krstic, K.-T. Cheng – *UC-Santa Barbara, USA*

8D-1 **Using Red-Black Interval Trees in Device-Level Analog Placement with Symmetry Constraints**
F. Balasa, S. C. Maruvada, K. Krishnamoorthy – *University of Illinois at Chicago, USA*

AMD EX1021
AMD v. Realtek - Page 400

ASP-DAC 2003
Table of Contents

ASP-DAC 2003 General Chair's Message iii
Techical Program Co-Chairs' Message iv
University LSI Design Contest v
Keynote Addresses vi
ASP-DAC 2002/VLSI 2002 Best Papers x
ASP-DAC 2003 Best Paper Award Candidates xi
ASP-DAC 2003 Organizing Committee xii
ASP-DAC 2003 Steering Committee xvi
ASP-DAC 2003 Technical Program Committee xx
University LSI Design Contest Committee xxiii
List of Reviewers xxiv

Session 1A
Bus Encoding and Memory Optimization

Co-Chairs: Luca Benini, Hiroshi Nakamura

1A-1 **BEAM: Bus Encoding Based on Instruction-Set-Aware Memories**
Yazdan Aghaghiri, Farzan Fallah, Massoud Pedram 3

1A-2 **Irredundant Address Bus Encoding Techniques Based on Adaptive Codebooks for Low Power**
Satoshi Komatsu, Masahiro Fujita 9

1A-3 **Multi-Parametric Improvements for Embedded Systems Using Code-Placement and Address Bus Coding**
Sri Parameswaran, Joerg Henkel 15

1A-4 **Memory Access Pattern Analysis and Stream Cache Design for Multimedia Applications**
Junghee Lee, Chanik Park, Soonhoi Ha 22

AMD EX1021
AMD v. Realtek - Page 401

Session 1B
DSM Interconnect and Gate Issues

Co-Chairs: Masanori Hashimoto, Kaushik Roy

1B-1 **A Statistical Gate Delay Model for Intra-chip and Inter-chip Variabilities**
Kenichi Okada, Kento Yamaoka, Hidetoshi Onodera 31

1B-2 **A Fast and Accurate Method for Interconnect Current Calculation**
Muzhou Shao, D. F. Wong, Youxin Gao, Huijing Cao, Li-Pen Yuan 37

1B-3 **Calculating the Effective Capacitance for the RC Interconnect in VDSM Technologies**
Soroush Abbaspour, Massoud Pedram 43

1B-4s **Reduction of Crosstalk Noise by Optimizing 3-D Configuration of the Routing Grid**
Atsushi Sakai, Takashi Yamada, Yoshifumi Matsushita, Hiroto Yasuura 49

1B-5s **Design Tools for 3-D Integrated Circuits**
Shamik Das, Anantha Chandrakasan, Rafael Reif 53

Session 1C
Embedded Software: Task Scheduling and Compilation

Co-Chairs: Nikil Dutt, Akira Fukuda

1C-1 **An On-line Approach for Power Minimization in QoS Sensitive Systems**
Jennifer L. Wong, Miodrag Potkonjak, Gang Qu 59

1C-2 **Energy Minimization of Real-time Tasks on Variable Voltage Processors with Transition Energy Overhead**
Yumin Zhang, Xiaobo Sharon Hu, Danny Z. Chen 65

1C-3 **Register Aware Scheduling for Distributed Cache Clustered Architecture**
Zhong Wang, Xiaobo Sharon Hu, Edwin H. -M. Sha 71

1C-4 **Data Partitioning for Maximal Scratchpad Usage**
Manish Verma, Stefan Steinke, Peter Marwedel 77

Session 1D
Combinational and Sequential Verification

Co-Chairs: Kiyoharu Hamaguchi, Yuji Kukimoto

1D-1 **SAT-based Sequential Depth Computation**
Maher Mneimneh, Karem Sakallah 87

1D-2 **Logic Verification Based on Diagnosis Techniques**
Veneris Andreas, Smith Alexander, Abadir Magdy 93

1D-3 **Algorithms for Compacting Error Traces**
Yirng-An Chen, Fang-Sung Chen 99

AMD EX1021
AMD v. Realtek - Page 402

1D-4s **Transaction-based Waveform Analysis for IP Selection**
Jian Liu, Eugene Shragowitz 104

1D-5s **An Automatic Interconnection Rectification Technique for SoC Design Integration**
Chun-Yao Wang, Shing-Wu Tung, Jing-Yang Jou 108

Session 2A
C-Based Specification and ASIP Design

Co-Chairs: Nagisa Ishiura, Wolfgang Rosenstiel

2A-1 **Typing Abstractions and Management in a Component Framework**
Frederic Doucet, Sandeep Shukla, Rajesh Gupta 115

2A-2 **Event-Driven Observability Enhanced Coverage Analysis of C Programs for Functional Validation**
Farzan Fallah, Indradeep Ghosh, Masahiro Fujita 123

2A-3 **Trace-driven Rapid Pipeline Architecture Evaluation Scheme for ASIP Design**
Jun Kyoung Kim, Tag Gon Kim 129

2A-4 **A Hardware/Software Partitioning Algorithm for SIMD Processor Cores**
Koichi Tachikake, Nozomu Togawa, Yuichiro Miyaoka, Jinku Choi, Masao Yanagisawa, Tatsuo Ohtsuki 135

Session 2B
On-Chip Inductance

Co-Chairs: Hidetoshi Onodera, Martin D. F. Wong

2B-1 **Approximate Formulae Approach for Efficient Inductance Extraction**
Atsushi Kurokawa, Takashi Sato, Hiroo Masuda 143

2B-2 **Accurate Prediction of the Impact of On-chip Inductance on Interconnect Delay Using Electrical and Physical Parameters**
Takashi Sato, Toshiki Kanamoto, Atsushi Kurokawa, Yoshiyuki Kawakami, Hiroki Oka, Tomoyashi Kitaura, Hiroyuki Kobayashi, Masanori Hashimoto 149

2B-3 **A Metric for Analyzing Effective On-Chip Inductive Coupling**
Guoan Zhong, Cheng-Kok Koh, Kaushik Roy 156

2B-4 **Determination of Worst-Case Crosstalk Noise for Non-Switching Victims in GHz+ Interconnects**
Jun Chen, Lei He 162

Session 2C
Circuit and Modeling

Co-Chairs: Hiroaki Ueno, Zhiping Yu

2C-1 **Recent Developments in ESD Protection for RF ICs**
Albert Z. H. Wang 171

AMD EX1021
AMD v. Realtek - Page 403

2C-2 **Temperature-Independence-Point Properties for 0.1μm-Scale Pocket-Implant Technologies and the Impact on Circuit Design**
Kazuya Hisamitsu, Hiroaki Ueno, Masayasu Tanaka, Daisuke Kitamaru, Mitiko Miura-Mattausch, Hans Juergen Mattausch, Shigetaka Kumashiro, Tetsuya Yamaguchi, Kyoji Yamashita, Noriaki Nakayama 179

2C-3 **Behavioral Modeling of EM Devices by Selective Orthogonal Matrix Least-Squares Method**
Yuichi Tanji, Masaya Suzuki, Takayuki Watanabe, Hideki Asai 184

Session 2D
Logic Optimization and Technology Mapping

Co-Chairs: Giovanni De Micheli, Shin-Ichi Minato

2D-1 **A BDD-based Fast Heuristic Algorithm for Disjoint Decomposition**
Tomas Bengtsson, Andrés Martinelli, Elena Dubrova 191

2D-2 **Logic Optimization for Asynchronous Speed Independent Controllers Using Transduction Method**
Hiroshi Saito, Hiroshi Nakamura, Masahiro Fujita, Takashi Nanya 197

2D-3 **Technology Mapping for Low Leakage Power and High Speed with Hot-Carrier Effect Consideration**
Chang Woo Kang, Massoud Pedram 203

2D-4s **Synthesis of High Performance Low Power PTL Circuits**
Debasis Samanta, M. C. Dharmadeep, Ajit Pal 209

2D-5s **A Technology Mapping Algorithm for Heterogeneous FPGAs**
Chi-Chou Kao, Yen-Tai Lai 213

Session 3A
SoC and NoC

Co-Chairs: Masahiro Fujita, Rajesh Gupta

3A-1 **Combining Architecture Exploration and a Path to Implementation to Build a Complete SoC Design Flow from System Specification to RTL**
Mohamed-Anouar Dziri, Firaz Samet, Flavio Wagner, Wander Cesario, Ahmed A. Jerraya 219

3A-2 **Towards On-Chip Fault-Tolerant Communication**
Tudor Dumitras, Sam Kerner, Radu Marculescu 225

3A-3 **Energy-Aware Mapping for Tile-based NoC Architectures under Performance Constraints**
Jingcao Hu, Radu Marculescu 233

Session 3B
Clock Synthesis and Capacitance Extraction

Co-Chairs: Masato Edahiro, Massoud Pedram

AMD EX1021
AMD v. Realtek - Page 404

3B-1 **Adaptive Wire Adjustment for Bounded Skew Clock Distribution Network**
Haydar Saaied, Dhamin Al-Khalili, Asim Al-Khalili, Mohamed Nekili 243

3B-2 **Power Minimization by Clock Root Gating**
Qi Wang, Sumit Roy 249

3B-3 **BBE: Hierarchical Computation of 3-D Interconnect Capacitance with BEM Block Extraction**
Taotao Lu, Zeyi Wang, Xianlong Hong 255

3B-4 **Improving Boundary Element Methods for Parasitic Extraction**
Shu Yan, Jianguo Liu, Weiping Shi 261

Session 3C
Analysis Methodologies for Circuits

Co-Chairs: Naoyuki Shigyo, Albert Z. H. Wang

3C-1 **Statistical Delay Computation Considering Spatial Correlations**
Aseem Bijay Agarwal, David Blaauw, Savithri Sundareswaran, Vladimir Zolotov, Min Zhao, Kaushik Gala, Rajendran Panda 271

3C-2 **Predicting Short Circuit Power from Timing Models**
Emrah Acar, Ravishankar Arunachalam, Sani R. Nassif 277

3C-3 **RCLK-VJ Network Reduction with Hurwitz Polynomial Approximation**
Zhanhai Qin, Chung-Kuan Cheng 283

Session 3D
Symbolic Simulation and Verification

Co-Chairs: Yirng-An Chen, Shinji Kimura

3D-1 **Gate-Level Simulation of Quantum Circuits**
George F. Viamontes, Manoj Rajagopalan, Igor L. Markov, John P. Hayes 295

3D-2 **Enhanced Symbolic Simulation for Efficient Verification of Embedded Array Systems**
Tao Feng, Li-C. Wang, Kwang-Ting (tim) Cheng, Manish Pandey, Magdy S. Abadir 302

3D-3s **Hardware Verification Using ANSI-C Programs as a Reference**
Edmund Clarke, Daniel Kroening 308

3D-4s **Evaluation of Multiple-Output Logic Functions Using Decision Diagrams**
Yukihiro Iguchi, Tsutomu Sasao, Munehiro Matsuura 312

Session 4A
Modeling for Floorplan

Co-Chairs: Dinesh P. Mehta, Yasuhiro Takashima

4A-1 **A Simulated Annealing Approach with Sequence-Pair Encoding Using a Penalty Function for Placement Problem with Boundary Constraints**
Satoshi Tayu 319

AMD EX1021
AMD v. Realtek - Page 405

4A-2 **Multi-level Placement for Large-Scale Mixed-Size IC Designs**
Chin-Chih Chang, Jason Cong, Xin Yuan 325

4A-3 **Selected Sequence-Pair: An Efficient Decodable Packing Representation in Linear Time Using Sequence-Pair**
Chikaaki Kodama, Kunihiro Fujiyoshi 331

4A-4s **An Extended Representation of Q-sequence for Optimizing Channel-Adjacency and Routing-Cost**
Changwen Zhuang, Keishi Sakanushi, Liyan Jin, Yoji Kajitani 338

4A-5s **Non-slicing Floorplans with Boundary Constraints Using Generalized Polish Expression**
De-Sheng Chen, Chang-Tzu Lin, Yi-Wen Wang 342

Session 4B
(Special Session) Panel Discussion: Anatomy of Platform-Based Design: Is It the Savior of UDSM SoC Design Crisis?

Organizers: *Tadahiko Nakamura, Takahide Inoue*
Moderator: *Takahide Inoue*
Panelists: *Bob Altizer, Ken Chen, Jun Iwamura, Masasuke Kishi, Grant Martin, Augusto De Oliveira*

Abstract 349

Session 4C
Reconfigurable Systems

Co-Chairs: Reiner Hartenstein, Kiyoshi Oguri

4C-1 **Efficient LUT-Based FPGA Technology Mapping for Power Minimization**
Hao Li, Wai-Kei Mak, Srinivas Katkoori 353

4C-2 **Optimal Reconfiguration Sequence Management**
Soheil Ghiasi, Majid Sarrafzadeh 359

4C-3s **On Improving FPGA Routability Applying Multi-level Switch Boxes**
Jiping Liu, Hongbing Fan, Yu-Liang Wu 366

4C-4s **An Image Retrieval System Using FPGAs**
Koji Nakano, Etsuko Takamichi 370

4C-5 **Logic Foundry: Rapid Prototyping of FPGA-based DSP Systems**
Gary Spivey, Shuvra Bhattacharyya, Kazuo Nakajima 374

Session 4D
Design Methodologies for Leading Edge Low-Power Design

Co-Chairs: Tohru Ishihara, Radu Marculescu

AMD EX1021
AMD v. Realtek - Page 406

4D-1 **Advanced Power Management Techniques: Going Beyond Intelligent Shutdown**
Luca Benini 385

4D-2 **Design Technologies for Low Power Microprocessors**
Toshihiro Hattori 390

4D-3 **Design Methodology of Low-Power CMOS RF-ICs**
Tsuneo Tsukahara 394

4D-4s **Minimizing Total Power by Simultaneous Vdd/Vth Assignment**
Ashish Srivastava, Dennis Sylvester 400

4D-5s **A Low Power CMOS Circuit with Variable Source Scheme (VSCMOS)**
Takeo Yasuda, Kohji Hosokawa 404

Session 5A
Performance Driven Floorplan

Co-Chairs: Yao-Weng Chang, Shin'ichi Wakabayashi

5A-1 **Fast Buffer Planning and Congestion Optimization in Interconnect-driven Floorplanning**
Keith W. C. Wong, Evangeline F. Y. Young 411

5A-2 **Interconnect-Driven Floorplanning by Searching Alternative Packings**
Chiu-Wing Sham, Evangeline F. Y. Young, Hai Zhou 417

5A-3s **Noise-Aware Buffer Planning for Interconnect-Driven Floorplanning**
Shu-Min Li, Yih-Huai Cherng, Yao-Wen Chang 423

5A-4s **Floorplanning with Power Supply Noise Avoidance**
Hung-Ming Chen, Li-Da Huang, I-Min Liu, Minghorng Lai, D. F. Wong 427

5A-5s **Simultaneous Floorplanning and Buffer Block Planning**
Iris Hui-Ru Jiang, Yao-Wen Chang, Jing-Yang Jou, Kai-Yuan Chao 431

5A-6s **A Buffer Planning Algorithm Based on Dead Space Redistribution**
Song Chen, Xianlong Hong, Sheqin Dong, Yuchun Ma, Yici Cai, Chungkuan Cheng, Jun Gu 435

Session 5B
(Special Session) Invited Talks: Virtual Core Based Reuse Methodology for SoC Design

Co-Chairs: Masahiro Fujita, Sandeep K. Shukla

5B-1 **VCore- Based Design Methodology**
Michiaki Muraoka, Hideyuki Hamada, Hiroaki Nishi, Toshihiko Tada, Yoichi Onishi, Toshinori Hosokawa 441

5B-2 **Synthesis for SoC Architecture Using VCores**
Hiroaki Nishi, Michiaki Muraoka, Rafael K. Morizawa, Hideaki Yokota, Hideyuki Hamada 446

5B-3 **VCore-based Platform for SoC Design**
Yoichi Onishi, Michiaki Muraoka, Makoto Utsuki, Naoyuki Tsubaki 453

AMD EX1021
AMD v. Realtek - Page 407

5B-4 **VCDS Tool Demonstration**
Rafael Kazumiti Morizawa 459

Session 5C
(Special Session) Invited Talks + Panel Discussion: Adaptive Computing: What Can It Do, Where Can It Go?

Organizer: *Robert Reuss*
Moderators: *Jose Munoz, Toshiaki Miyazaki*
Panelists: *Nader Bagherzadeh, Prith Banerjee, Brad Hutchings, Jose Munoz, Brian Schott*

Abstract 463

5C-1 **DARPA'S Adaptive Computing Systems Program**
Jose Munoz 464

5C-2 **Applications of Adaptive Computing Systems for Signal Processing Challenges**
Brian Schott, Robert Parker 465

5C-3 **Interactive Ray Tracing on Reconfigurable SIMD MorphoSys**
H. Du, M. Sanchez-Elez, N. Tabrizi, N. Bagherzadeh, M. L. Anido, M. Fernandez 471

5C-4 **An Overview of a Compiler for Mapping MATLAB Programs onto FPGAs**
Prith Banerjee 477

5C-5 **Issues in Debugging Highly Parallel FPGA-based Applications Derived from Source Code**
Karl Scott Hemmert, Brad L. Hutchings 483

Session 5D
Leading Edge Design Examples

Co-Chairs: Seongsoo Lee, Takashi Miyamori

5D-1s **Implementation of the Super-Systolic Array for Convolution**
Gi-Yong Song, Jae-Jin Lee 491

5D-2s **Design of a Scalable RSA and ECC Crypto-Processor**
Ming-Cheng Sun, Chih-Pin Su, Chih-Tsun Huang, Cheng-Wen Wu 495

5D-3s **A Reconfigurable, Power-Scalable Rake Receiver IP for W-CDMA**
Alessandro Bianco, Alberto Dassatti, Maurizio Martina, Andrea Molino, Fabrizio Vacca 499

5D-4s **Robust High-Performance Low-Power Carry Select Adder**
Woopyo Jeong, Kaushik Roy 503

5D-5s **Full-Custom vs. Standard-Cell Design Flow - An Adder Case Study**
Henrik Eriksson, Tomas Henriksson, Per Larsson-Edefors, Christer Svensson 507

5D-6s **A 500-MHz Low-Power Five-Port CMOS Register File**
Jiajing Wang, Qianling Zhang 511

AMD EX1021
AMD v. Realtek - Page 408

5D-7s **An Effective SDRAM Power Mode Management Scheme for Performance and Energy Sensitive Embedded Systems**
Ning-Yaun Ker, Chung-Ho Chen 515

5D-8s **Branch Predictor Design and Performance Estimation for a High Performance Embedded Microprocessor**
Sang-Hyuk Lee, Il-Kwan Kim, Lynn Choi 519

Session 6A
Design Space Exploration

Co-Chairs: Tohru Ishihara, Sri Parameswaran

6A-1 **Accelerating Design Space Exploration Using Pareto-Front Arithmetics**
Christian Haubelt, Jürgen Teich 525

6A-2 **Quality-Driven Design by Bitwidth Optimization for Video Applications**
Yun Cao, Hiroto Yasuura 532

6A-3 **Arbitrary Long Digit Integer Sorter HW/SW Co-Design**
Shun-Wen Cheng 538

Session 6B
(Special Session) Panel Discussion: Roles of Funding Agencies in Technology-driven Economic Development

Organizer: *Kazuo Nakajima*
Moderators: *Kazuo Nakajima, Brian Schott*
Panelists: *Tokinori Kozawa, Jose Munoz, Wolfgang Rosenstiel, Sakae Takahashi, Chen-Wen Wu*

Abstract 547

Session 6C
(Special Session) Invited Talk: Legal Protection for Semiconductor Intellectual Property

Chair: Masaharu Imai

6C-1 **Legal Protection for Semiconductor Intellectual Property (IP)**
Yoichi Oshima 551

Session 6D
(Special Session) Presentation and Poster Session: University LSI Design Contest

Co-Chairs: Tomohisa Wada, Shoji Kawahito

6D-1 **Design and Implementation of a Video-Oriented Network-Interface-Card System**
Ming-Chih Chen, Shen-Fu Hsiao, Cheng-Hsien Yang 559

AMD EX1021
AMD v. Realtek - Page 409

6D-2 **A Highly Efficient AES Cipher Chip**
Chih-Pin Su, Tsung-Fu Lin, Chih-Tsun Huang, Cheng-Wen Wu 561

6D-3 **Implementation of Fast CRC Calculation**
Tomas Henriksson, Dake Liu 563

6D-4 **Design of a CMOS Test Chip for Package Models and I/O Characteristics Verification**
Chetan Deshpande, Tom Chen 565

6D-5 **A Still Image Encoder Based on Adaptive Resolution Vector Quantization Employing Needless Calculation Elimination Architecture**
Masanori Fujibayashi, Toshiyuki Nozawa, Takahiro Nakayama, Kenji Mochizuki, Koji Kotani, Shigetoshi Sugawa, Tadahiro Ohmi 567

6D-6 **Speech Encoding and Encryption in VLSI**
KalyanChakravarti Kongara, Srinivas B. Mandalika 569

6D-7 **The Design of a i8080A Instruction Compatible Processor with Extended Memory Address**
Chiaki Kon, Naohiko Shimizu 571

6D-8 **The Design of a USB Device Controller IYOYOYO**
Tomoaki Kouyama, Hibiki Nano, Chiaki Kon, Naohiko Shimizu 573

6D-9 **MAPLE Chip: a Processing Element for a Static Scheduling Centric Multiprocessor**
Kenta Yasufuku, Riku Ogawa, Keisuke Iwai, Hideharu Amano 575

6D-10 **Finding the Best System Design Flow for a High-Speed JPEG Encoder**
Kazuo Sakiyama, Patrick Schaumont, Ingrid Verbauwhede 577

6D-11 **The Design of PCI Bus Interface**
Haruyasu Hayasaka, Hiroaki Haramiishi, Naohiko Shimizu 579

6D-12 **Low-Power Digital CDMA Receiver**
I Hsin Chen, Ja Sheng Liu, Yi Chen Tsai 581

6D-13 **Hardware Implementation of an EAN-13 Bar Code Decoder**
Jeroen De Maeyer, Harald Devos, Wim Meeus, Peter Verplaetse, Dirk Stroobandt 583

6D-14 **Error Correction Receiver Using Difference-set Cyclic Code**
Yukihiro Kato, Tomokazu Morita 585

6D-15 **Design of Digital CDMA Receiver**
Vijay Kumar Immadi, Srinivas B. Mandalika 587

6D-16 **Standard Cell Libraries with Various Driving Strength Cells for 0.13, 0.18 and 0.35μm Technologies**
Masanori Hashimoto, Kazunori Fujimori, Hidetoshi Onodera 589

6D-17 **A Nearest-Hamming-Distance Search Memory with Fully Parallel Mixed Digital-Analog Match Circuitry**
Tetsushi Koide, Hans Juergen Mattausch, Yuji Yano, Takayuki Gyohten, Yoshihiro Soda . 591

AMD EX1021
AMD v. Realtek - Page 410

Session 7A
System-Level Power Issues

Co-Chairs: Kazutoshi Kobayashi, Massoud Pedram

7A-1 **Run-Time Energy Estimation in System-On-a-Chip Designs**
Josef Haid, Gerald Kaefer, Christian Steger, Reinhold Weiss 595

7A-2 **SEA: Fast Power Estimation for Micro-Architectures**
Praveen G. N. Kalla, Joerg Henkel, Xiaobo (sharon) Hu 600

7A-3s **HyPE: Hybrid Power Estimation for IP-Based Programmable Systems**
Xun Liu, Marios C. Papaefthymiou 606

7A-4s **An Efficient IP-Level Power Model for Complex Digital Circuits**
Chih-Yang Hsu, Chien-Nan Jimmy Liu, Jing-Yang Jou 610

7A-5 **A Hierarchical Analysis Methodology for Chip-Level Power Delivery with Realizable Model Reduction**
Yu-Min Lee, Charlie Chung-Ping Chen 614

Session 7B
(Special Session) Invited Talks: Design Methodologies for 50M Gate ASICs

Co-Chairs: Jason Cong, Satoshi Matsushita

7B-1 **Optimality and Scalability Study of Existing Placement Algorithms**
Chin-Chih Chang, Jason Cong, Min Xie 621

7B-2 **IBM's 50 Million Gate ASICs**
Juergen Koehl, David E. Lackey 628

7B-3 **Silicon Virtual Prototyping: The New Cockpit for Nanometer Chip Design**
Wei-Jin Dai, Dennis Huang, Chin-Chih Chang, Michel Courtoy 635

7B-4 **Design Flow and Methodology for 50M Gate ASIC**
Yatin Trivedi 640

Session 7C
(Special Session) Invited Talks: Mixed Signal Test

Co-Chairs: Kwang-Ting Cheng, Yasuo Sato

7C-1 **Efficient Loop-back Testing of On-chip ADCs and DACs**
Hak-Soo Yu, Jacob A. Abraham, Sungbae Hwang, Jeongjin Roh 651

7C-2 **A Novel LCD Driver Testing Technique Using Logic Test Channel**
Chauchin Su, Wei-Juo Wang, Chih-Hu Wang 657

7C-3 **An Implementation of Memory-based On-chip Analogue Test Signal Generation**
Salvador Mir, L. Rolíndez, C. Domingues, L. Rufer 663

AMD EX1021
AMD v. Realtek - Page 411

7C-4 **Delta-sigma Modulator Based Mixed-signal BIST Architecture for SoC**
Chee-Kian Ong, Kwang-Ting Cheng, Li Wang 669

Session 7D
Embedded Systems: Hardware/ Software Design Methodology and Optimization

Co-Chairs: Naehyuck Chang, Hiroaki Takada

7D-1 **Capturing and Analyzing Requirement- In Case of Software and Applying to Hardware -**
Akira Kawaguchi 677

7D-2 **A Comparison of the RTU Hardware RTOS with a Hardware/Software RTOS**
Jaehwan Lee, Vincent John Mooney III, Anders Daleby, Karl Ingstrom, Tommy Klevin, Lennart Lindh 683

7D-3s **Linux Kernel Customization for Embedded Systems by Using Call Graph Approach**
Che-Tai Lee, Zeng-Wei Hong, Jim-Min Lin 689

7D-4s **Topology Selection for Energy Minimization in Embedded Networks**
Dexin Li, Pai H. Chou, Nader Bagherzadeh 693

Session 8A
Design Validation Techniques

Co-Chairs: Vasily Moshnyaga, Hiroyuki Ochi

8A-1 **Semi-Formal Test Generation and Resolving a Temporal Abstraction Problem in Practice: Industrial Application**
Julia Dushina, Mike Benjamin, Daniel Geist 699

8A-2 **Scan-chain Based Watch-points for Efficient Run-Time Debugging and Verification of FPGA Designs**
Anurag Tiwari, Karen A. Tomko 705

8A-3s **A Novel Approach for Digital Waveform Compression**
Edwin Naroska, Shanq-Jang Ruan, Chia-Lin Ho, Said Mchaalia, Feipei Lai, Uwe Schwiegelshohn 712

8A-4s **A Deep Submicron Power Estimation Methodology Adaptable to Variations between Power Characterization and Estimation**
Daniel Eckerbert, Per Larsson-Edefors 716

Session 8B
Placement

Co-Chairs: Masaaki Yamada, Xin Yuan

8B-1 **Congestion Driven Incremental Placement Algorithm for Standard Cell Layout**
Zhuoyuan Li, Weimin Wu, Xianlong Hong 723

AMD EX1021
AMD v. Realtek - Page 412

8B-2 **Performance-Driven Multi-Level Clustering for Combinational Circuits**
C. N. Sze, Ting-Chi Wang 729

8B-3 **Cross Talk Driven Placement**
Jinan Lou, Wei Chen 735

8B-4s **VLSI Module Placement with Pre-placed Modules and Considering Congestion Using Solution Space Smoothing**
Sheqin Dong, Xianlong Hong, Xin Qi, Ruijie Wang, Song Chen, Jun Gu 741

8B-5s **A Path-based Timing-driven Quadratic Placement Algorithm**
Wenting Hou, Xianlong Hong, Weimin Wu, Yici Cai 745

Session 8C
Test Issues for Deep Sub-Micron Design

Co-Chairs: Tomoo Inoue, Rochit Rajsuman

8C-1 **Experience in Critical Path Selection for Deep Sub-Micron Delay Test and Timing Validation**
Jing-Jia Liu, Li-C. Wang, Angela Krstic, Kwang-Ting (tim) Cheng 751

8C-2 **On Effective Criterion of Path Selection for Delay Testing**
Masayasu Fukunaga, Seiji Kajihara, Sadami Takeoka, Shinichi Yoshimura 757

8C-3 **DFT Timing Design Methodology for At-Speed BIST**
Yasuo Sato, Motoyuki Sato, Koki Tsutsumida, Masatoshi Kawashima, Kazumi Hatayama, Kazuyuki Nomoto 763

8C-4 **An Automated Method for Test Model Generation from Switch Level Circuits**
Tim McDougall, Atanas N Parashkevov, Simon Jolly, Juhong Zhu, Jing Zeng, Magdy Abadir, Carol Pyron 769

Session 8D
Analog Circuits Design and Methodology

Co-Chairs: Yu-Chung Hung, Makoto Nagata

8D-1 **Using Red-Black Interval Trees in Device-Level Analog Placement with Symmetry Constraints**
Florin Balasa, Sarat C. Maruvada, Karthik Krishnamoorthy 777

8D-2 **Current-Driven Wire Planning for Electromigration Avoidance in Analog Circuits**
Jens Lienig, Goeran Jerke 783

8D-3 **Efficient DDD-based Term Generation Algorithm for Analog Circuit Behavioral Modeling**
Sheldon X. -D. Tan, C. -J. Richard Shi 789

8D-4 **5Gbps Serial Link Transmitter with Pre-emphasis**
Chih-Hsien Lin, Chun-Hong Wang, Shyh-Jye Jou 795

AMD EX1021
AMD v. Realtek - Page 413

Session 9A
Synthesis for Power Performance Optimization

Co-Chairs: Yutaka Tamiya, Li-C. Wang

9A-1 **Low Power Synthesis of Finite State Machines with Mixed D and T Flip-Flops**
Ali Iranli, Peyman Rezvani, Massoud Pedram 803

9A-2 **Don't Cares in Logic Minimization of Extended Finite State Machines**
Yunjian Jiang, Robert K. Brayton 809

9A-3s **Performance Optimization of Synchronous Control Units for Datapaths with Variable Delay Arithmetic Units**
Euiseok Kim, Hiroshi Saito, Jeong-Gun Lee, Dong-Ik Lee, Hiroshi Nakamura, Takashi Nanya 816

9A-4s **Integer Linear Programming-Based Synthesis of Skewed Logic Circuits**
Aiqun Cao, Naran Sirisantana, Cheng-Kok Koh, Kaushik Roy 820

Session 9B
Routing

Co-Chairs: Xinlong Hong, Yoichi Shiraiashi

9B-1 **Highly Scalable Algorithms for Rectilinear and Octilinear Steiner Trees**
Andrew B. Kahng, Ion Mandoiu, Alexander Zelikovsky 827

9B-2 **UTACO: A Unified Timing and Congestion Optimizing Algorithm for Standard Cell Global Routing**
Tong Jing, Xianlong Hong, Haiyun Bao, Yici Cai, Jingyu Xu, Chungkuan Cheng, Jun Gu 834

9B-3 **The Y-Architecture: Yet Another On-Chip Interconnect Solution**
Hongyu Chen, Feng Zhou, Chung-Kuan Cheng 840

9B-4s **A Novel Timing-Driven Global Routing Algorithm Considering Coupling Effects for High Performance Circuit Design**
Jingyu Xu, Xianlong Hong, Tong Jing, Yici Cai, Jun Gu 847

9B-5s **Graph Matching-Based Algorithms for Array-Based FPGA Segmentation Design and Routing**
Jai-Ming Lin, Song-Ra Pan, Yao-Wen Chang 851

Session 9C
DFT Optimizations

Co-Chairs: Alfred Crouch, Hiroshi Date

9C-1 **Routing-Aware Scan Chain Ordering**
Puneet Gupta, Stefanus Mantik, Andrew B. Kahng 857

9C-2 **Multiple Test Set Generation Method for LFSR-Based BIST**
YouHua Shi, Zhe Zhang 863

AMD EX1021
AMD v. Realtek - Page 414

9C-3 **A Seed Selection Procedure for LFSR-Based Random Pattern Generators**
Kenichi Ichino, Kohichi Watanabe, Masayuki Arai, Satoshi Fukumoto, Kazuhiko Iwasaki 869

9C-4s **Efficient BIST Design for Sequential Machines Using FiF-FoF Values in Machines States**
Roy S, Maulik U, Bandyopadhyay S, Basu S, Sikdar K Biplab 875

9C-5s **A New Design-for-Test Technique for Reducing SOC Test Time**
C V Guru Rao, D Roy Chowdhury 879

Session 9D
RF Circuits Design and Methodology

Co-Chairs: Wing-Hung Ki, Tsuneo Tsukahara

9D-1 **Periodic Steady-State Analysis of Coupled ODE-AE-CGE Systems for MOS RF Autonomous Circuit Simulation**
Xinyu Wu, Zaiman Chen, Jinmei Lai, Qianling Zhang, Omar Wing, Junyan Ren 885

9D-2 **A Frequency Separation Macromodel for System-Level Simulation of RF Circuits**
Xin Li, Peng Li, Yang Xu, Robert Dimaggio, Lawrence Pileggi 891

9D-3 **Nonlinear Distortion Analysis via Linear-Centric Models**
Peng Li, Lawrence T. Pileggi 897

9D-4 **Parasitic-Aware Design and Optimization of a Fully Integrated CMOS Wideband Amplifier**
Jinho Park, David J. Allstot 904

Author Index 909

AMD EX1021
AMD v. Realtek - Page 415

This material may be protected by Copyright law (Title 17 U.S. Code)

Leading Edge Low Power Design

Luca Benini
DEIS Universitá di Bologna
Bologna, ITALY

Abstract— Well into the System-on-Chip era, power consumption has emerged as one of the most critical challenges to design complexity scaling. Moving from a critical assessment of current technologies and architectures, we survey the distinguishing features of a design methodology that aims at energy consumption reduction, under guaranteed *quality of service* (QoS), as a main objective in system design.

I. Introduction

Systems on chips (SoCs) containing tens of millions of transistors are currently designed and produced. Billion-transistor chips are not a distant dream in the future. Several challenges arise from the complexity of designing multimillion- and billion-transistor chips. Design methods (and tools) must follow the rapid pace of evolution dictated by technology and application pull. As of today, a few fundamental trends of evolution have emerged.

- Technology is increasingly hard to master, both from a cost and a reliabili perspective. Hence, application-specific hardware platforms are becoming not viable from an economic viewpoint. Large-scale SoCs must be flexible and programmable after fabrication (to sell in large volumes), and they must contain a large amount of functional redundancy (to tolerate technology-induced faults and bring yield to acceptable levels).
- Systems on chips are designed using pre-existing components. Reuse becomes key for designing chips fast enough, because no design team can successfully complete a multi-million transistor design "from scratch" in the six to nine month window dictated by market pressure. Design methodologies must support component re-use in a plug-and-play fashion.
- While the traditional computing equipment market has reached a saturation region where margins are extremely low, great promise is shown by the emergence of "ambient intelligence" devices. These devices will materialize the vision of ubiquitous interactive access to world-wide information and communication resources from tiny wireless terminals, coupled extensive data collection and integration from a myriad of geographically dispersed sensors.

These trends have jointly contributed to the creation of a unprecedented *power crisis*: on one side technology and complexity control push for highly-reusable, standardized and programmable architectures, which are notoriously much less power efficient than dedicated hardware [11]. On the other hand, the largest market potential is for chips that can be easily embedded in tiny, battery powered (even battery-less) devices, while at the same time performing computationally intensive tasks.

Low power design has thus become a critical need for current and future SoCs. In this paper we will first review the trends of evolution of current technologies and architectures, with the purpose of identifying the most critical bottlenecks in system power consumption. Then, we will focus on a few leading-edge low power design techniques that can provide immediate relief to the power crisis.

II. Technology trends

It is a well-known fact that rapidly evolving silicon technology creates smaller and faster CMOS transistors, with decreasing switching power. At a fist sight, then, technology works for reducing power consumption. However, die area is not shrinking with technology (in other words, chips pack a quadratically increasing number of transistors). This would not be a problem if the power dissipation of basic CMOS gates decreased fast enough to compensate the increase in integration density. Unfortunately, this is not the case, as detailed next.

First, because of bounded propagation speed of electromagnetic waves, as opposed to the ever faster switching speed of transistors, silicon technology is becoming increasingly interconnect-dominated. Delays on wires are already dominating: global wires spanning a significant fraction of the chip have propagation delays exceeding the clock period. The same holds for switching power: the gate capacitance of a minimum-size transistor is decreasing with technology, but interconnect capacitance per unit length is not decreasing at the same speed (because of sidewall contributions) and interconnect length is not decreasing for global wires. Hence, the C factor in the well known switching power equation $P_{sw} = kCV_{DD}^2 f$ (where k represents switching activity) does not scale down as fast as minimum feature size [4, 13].

Second, in order do reap performance benefits (in other words, to satisfy quality of service requirements), chip clock frequency is scaled faster than technology [5]. This result is achieved by clever architectural optimizations that reduce the number of logic stages to be traversed within the clock cycle time. From a power viewpoint this is clearly a problem, be-

0-7803-7659-5/03/$17.00 ©2003 IEEE.

AMD EX1021
AMD v. Realtek - Page 416

cause power is directly proportional to switching frequency f. Performance constraints (or objectives) are also the main reason why supply voltage scaling is not drastic as one would desire for power minimization purposes. In fact, transistor switching speed decreases as $(V_{DD} - V_T)^{\alpha}$, with $1 < \alpha < 2$. This cannot be tolerated if performance is tightly constrained.

Third, deep submicron transistors are increasingly leaky in the OFF state. Source-to-drain current due to subthreshold conduction is the dominant cause of leakage, but drain-to-gate currents due to electron tunnelling across the gate oxide is also becoming significant. Furthermore, random variations of the number of dopant atoms in the channel region cause poor threshold control, and. many transistors have a threshold significantly lower than nominal. Unfortunately, subthreshold conduction is exponentially dependent on threshold voltage and transistors with lower threshold leak exponentially more than nominal transistors. As a result, chip standby power is becoming a significant concern. Clearly, subthreshold leakage also a heavy limiter to threshold voltage V_T reduction, with obvious negative impact on supply voltage scaling [5].

From this brief overview, it is apparent that power is critical in many ways. Power density grows significantly with technology, thereby creating huge problems in power supply distribution. Total power consumption in active state increases as well, with obvious negative consequences in battery lifetime as well as thermal design. Finally, even stand-by power is growing fast with technology, impacting battery lifetime in portable applications. For these reasons, technologists are pointing at power as the one of the most likely show-stoppers to technology scaling if adequate countermeasures are not taken.

III. Architectural Trends

As seen in the previous section, technology evolution is not going to solve the power consumption problem. On the contrary, many technologists refer to design innovation at the circuit, logic, architectural level as a way out from the crisis. Unfortunately, trends in this area are not favorable to energy efficiency at all. Even though a significant research effort is being devoted to power minimization, mainstream architectural design is moving toward energy-hungry architectures.

Most systems-on-chip are nowadays designed with a high degree of programmability. The majority contains one or more core processors, and many instantiate several programmable coprocessors (e.g., VLIW units for numerical computations, programmable IO processors, etc.); a few recent designs even embed a significant amount of bit-programmable logic (FPGA fabrics) [15]. Programmability is a common requirement when designing large-scale SoCs for two main reasons. First, it ensures functional flexibility, which widens the spectrum of applicability and the potential production volume. Second, it leaves margins for post-fabrication bug fixing and tuning, thereby enhancing yield. High yield and volume of sales are required to amortize ballooning mask development and fabrication costs. Second, processor-based architectures emphasize reuse of hardware modules (e.g., the cores themselves, the memories, etc.) as well as software components (e.g. libraries, operating systems, compilers). Reuse-centric methodologies are instrumental to reduce development risk and cost, and minimize time-to-market.

Yet, flexibility and reuse come at a price. The power-performance ratio (i.e., the energy) required by a processor to carry out a given task (e.g., MPEG decoding) is several orders of magnitude (3 or more) higher than what could be achieved with an application-specific architecture [14, 11]. The causes of the energy inefficiency of processor-based architectures are well understood. Processors have to fetch and decode instructions to control execution. This is a huge overhead with respect to an application-specific architecture with a hardwired controller.

Instruction execution relies on heavily shared memory structures (register files), while dedicated architectures can exploit distributed registers with faster read/write time and enormously lower power consumption per access. The sequential instruction execution flow inevitably limits the parallelism that can be extracted from an application, and therefore emphasizes storage requirements. In general processor-based architectures are memory-hungry and require complex memory hierarchies to achieve satisfactory performance thanks to the exploitation of the spatial and temporal locality principles. As a result, a dominant fraction of the power consumed by a processor is spent in accessing the memory hierarchy (i.e., in communicating with memory structures and in extracting/storing information in them). Storage-related costs are minimized in application-specific architectures thanks to the exploitation of distributed and custom-sized memories and the minimization of spurious memory accesses created as an artefact of sequential instruction execution.

Advanced processors, often required to attain performance goals, are even more power-hungry than simple processors, because they rely on various forms of speculative execution to increase the average number of instructions executed in a clock cycle. Well-known performance-enhancement techniques, such as speculating past branches, value prediction, prefetching, imply the execution of redundant instructions which can be committed only after they completed. In case of a wrong prediction, a double overhead is often incurred. Not only computation has been executed needlessly, but also its effects must be rolled back.

Fine-grained programmable fabrics can be one or more orders of magnitude more energy-efficient than processors [14] for some classes of computations, but they still incur a very significant overhead with respect to dedicated logic. Recent data shows that computation performed by an embedded FPGA fabric in a hybrid FPGA-ASIC chip is more than two orders of magnitude more power consuming and more than ten times slower than the same computation in dedicated logic [15]. The overhead in this case is mainly due to communication between fine-grain programmable logic elements, which is performed on massively redundant programmable wiring resembling a multi-stage network, as opposed to dedicated, instance specific wires. Additionally, programmable logic blocks are of-

AMD EX1021
AMD v. Realtek - Page 417

ten much more complex than the basic gates they mimic when programmed.

In summary, it quite clear that the power crisis is caused by the conjunction of two factors: technology does not help us as much as we need, and architectural evolution trends are working against energy efficiency. To escape this losing scenario, a multi-pronged approach is needed, which improves energy efficiency without compromising flexibility, while minimizing performance overhead. This hard to do, but it can be done, as we shall see in the following sections.

IV. Designing low-power hardware

In a nutshell, we believe that the key for addressing the power consumption challenge lies in two simple ideas: (i) exploit flexibility not only at the functional level, but also in the power dimension (i.e., design *power-manageable architectures*), (ii) enable application-dependent specialization without compromising reuse and design flow streamlining. The first idea implies adding hardware resources that do not have a computational task, but they dedicated to controlling the power level of functional units, detecting and exploding idleness, and locally trading off performance with power. The second idea requires tuning a hardware platform to a specific computational task, without significantly increasing design time and effort (i.e., with a push-button process that does not require re-design of hardware blocks).

The practical embodiments of the two above principles often require both hardware and software support. In this section we analyze hardware implications. In particular, we focus on power manageable hardware, because design practices in this area are well established and widely spread. On the contrary, application-dependent specializations for power efficiency are performed much more in a ad-hoc fashion and they have more limited applicability, therefore they are not covered here. The interested reader can refer to a few recently published books and survey papers for additional information [2, 10, 11].

A. Power-manageable architectures

Power manageable architectures ultimately aim at eliminating idle power consumption (i.e., the power consumed by a hardware component when it is not in use) and run-time slack in active state, through the run-time control of clock activity and frequency, supply voltage and device threshold. The benefit of idle power elimination is obvious, while run-time slack elimination translates to cubic power savings (and quadratic energy savings) if the circuit is slowed down while at the same time reducing the power supply. Even though slow-down coupled with dynamic voltage scaling is more effective than idle power reduction [3], the two techniques are not mutually exclusive. Clearly, slowing down a circuit reduces its idleness, but in many cases it cannot eliminate it: consider for example a MP3 player. When the device is active it operates under tight performance constraints (namely, real-time playback), and obviously it is not possible to stretch execution time beyond the duration of a music track, even if after it has been played out, the player remains idle for hours.

Leakage is a major concern in idle-power reduction, because it impacts battery lifetime even if the circuit is completely idle. Quiescent power specifications tend to be very tight. In fact, CMOS technology has traditionally been extremely power-efficient when transistors are not switching, and system designers expect low leakage from CMOS chips. To meet leakage power constraints, *multiple-threshold* and *variable threshold* circuits have been proposed [9]. In multiple-threshold CMOS, the process provides two different thresholds. Low-threshold transistors are fast and leaky, and they are employed on speed-critical sub-circuits. High-threshold transistors are slower but exhibit low sub-threshold leakage, and they are employed in non-critical units/paths of the chip.

Unfortunately, multiple-threshold techniques tend to loose effectiveness as more transistors become timing-critical. Variable-threshold circuits overcome this shortcoming by dynamically controlling the threshold voltage of transistors through substrate biasing. When a variable-threshold circuit becomes quiescent, the substrate of NMOS transistors is negatively biased, and their threshold increases because of the well known *body-bias effect*. A similar approach can be taken for PMOS transistors (which require positive body bias). Variable-threshold circuits can in principle solve the quiescent leakage problem, but they require standby control circuits that modulate substrate voltage. Needless to say, accurate and fast body-bias control is quite challenging, and requires carefully designed closed-loop control [9].

Idle power is not only caused by leakage, but it is also due to unneeded switching activity. Clock switching within idle functional units is the best known example of this problem. Clock gating is used to eliminate unneeded clock activity. Most low-power processors implement both hardware and software controlled clock gating through dedicated power-down instructions. A radical way to eliminate idle power (both leakage and dynamic) is to disconnect a unit from its power supply. Unfortunately, in this case state information is lost.

In active state, supply voltage can be controlled to reduce power, albeit in the limited ranges allowed in aggressively scaled technologies. *Multiple-voltage* and *variable voltage* techniques have been developed to this purpose [11]. In multiple-voltage circuits two or more power supply voltages are distributed on chip. Similarly to the multiple-threshold scheme, timing-critical transistors can be powered at a high voltage, while most transistors are connected to the low voltage supply. Multiple voltages are also frequently used to provide standard voltage levels (e.g., 3.3 V) to input-output circuits, while powering on-chip internal logic at a much lower voltage to save power. The main challenges in the multiple-voltage approach are in the design of multiple power distribution grids and of power-efficient level-shifters to interface low-voltages with high-voltage sections.

Variable-voltage techniques offer the possibility of modulating the power supply dynamically during system operation. In principle, this is a very powerful technique, because it gives the

AMD EX1021
AMD v. Realtek - Page 418

possibility to trade off power for speed at run time, and to finely tune performance and power to non-stationary workloads. In practice, the implementation of this technique requires considerable design ingenuity. First, voltage changes require non-negligible time, because of the large time constants of power supply circuits. Second, the clock speed must be varied consistently with the varying speed of the core logic, when supply voltage is changed.

Power manageable hardware can be designed at different levels of granularity. While early embodiments allowed only coarse-grained control (e.g., peripheral vs. core logic), the trend is toward fine-grained support to power management. An example of a state-of-the-art power management support is the "Voltage islands" technology introduced by IBM [6], where it is possible to instantiate multiple regions on a chip each powered by its own variable voltage supply, clocked by a dedicated variable frequency clock generation unit, and equipped with independent threshold voltage control circuitry.

It is important to remember that power manageable hardware is only one side of the equation. Transitions between power states (e.g. from idle to active, from fast to slow) have significant cost in performance and energy. Hence, virtually all power management schemes require significant software support both at design (and compilation) time and at execution time. The main challenges are to avoid power state transitions that cannot be amortized and to select the most suitable power state for a given operating condition.

V. Power management software

We generalize the notion of operating system (OS) to the middleware that provides support for the operation of SoCs. Note that system support middleware in current SoCs is usually quite simple, designed for a specific integrated core processor, under the assumption that a processor provides global, centralized control for the system. In the context of future SoCs, the prevailing paradigm will be peer to peer interaction among several, possibly heterogeneous processing elements. Thus, we think that system software will be designed as a modular distributed system. Each programmable component will be provided with system software to support its own operation and to manage its communication to other modules.

The system software creates an abstraction of the underlying hardware platform. In a nutshell, we can view the system as a network of components. Each component models a computational or storage unit. We observe that:

- Each component can operate at various service levels, providing corresponding performance and energy consumption levels. This abstracts the physical implementation of components with adjustable voltage and/or frequency levels, as well as with the ability to disable their functions in full or in part.
- Even if units can be power managed independently, power minimization for a complex system requires information exchange between its components. Locally optimal choices do not generally lead to a globally optimal solution.

Dynamic power management entails selecting the appropriate component state to service a workload with the minimum energy consumption. DPM *policies* are the control algorithms for state transitions [1]. Transitions among states have a finite delay and energy penalty. Thus, the definition of policies that maximize performance under energy constraints, is a non-trivial problem [1]. Whereas policies can be implemented in hardware (as a part of the control-unit of a component), much more flexibility and ease of integrations is achieved by software implementations. Thus a policy can be seen as a program that is executed at run-time by the system software. The fundamental premise for the applicability of DPM is that systems (and their components) experience non-uniform workloads during operation time. Such an assumption is valid for most systems, both when considered in isolation and when inter-networked. A second assumption of DPM is that it is possible to predict, with a certain degree of confidence, the fluctuations of workload. Workload observation and prediction should consume little energy.

Power management policies can be implemented in system kernel and be tightly coupled to process management. Indeed, process management has knowledge of currently-executing tasks and tasks coming up for execution. Process managers know also which components (devices) are needed by each task. Thus, policy implementation at this level of system software enjoys both a global view and an outlook of the system operation in the near future. Predictive wake-up of components is possible with the knowledge of upcoming tasks and required components. The system software can be designed to improve effectiveness of power management. Indeed, power management exploits idle times of components. The scheduler can sequence tasks for execution with the additional goal of clustering component operation, thus achieving fewer but longer idle periods. The inter-relation between power manager and scheduler is even stronger for variable-voltage units. In this area numerous variable-voltage scheduling algorithms have been proposed [11], which aim at running processing elements as slow as possible (to allow supply voltage minimization), without violating performance constraints. Experiments with implementing DPM policies at different levels of system software [8] have shown increasing energy saving as the policies have deeper interaction with the system software functions.

A. Application-level power management

A reasonable questions is why not letting the application programs control the service levels and energy cost of the underlying hardware components. There are typically two objections to such an approach. First, application software should be independent of the hardware platform for portability reasons. Second, system software supports generally multiple tasks. When a task controls the hardware, unfair resource utilization and deadlocks may become serious problems.

AMD EX1021
AMD v. Realtek - Page 419

For these reasons, it has been suggested [7] that application programs contain system calls that request the system software to control a hardware component, e.g., by turning it on or shutting it down, or by requesting a specific frequency and/or voltage setting. The request can be accepted or denied by the operating system, that has access to the task schedule information and to the operating levels of the components. The advantage of this approach is that OS-based power management is enhanced by receiving detailed service request information from applications, and thus is in a position to take better decisions.

Another approach is to let the compiler extract directly the power management requests from the application programs at compile time. This is performed by an analysis of the code, followed by a code modification phase that inserts power management requests in critical points of the computation. For instance, in a variable-voltage unit, voltage and speed change requests can be issued immediately after data-dependent control-flow branching, when the behavior of the program is such that additional slack is created with respect to the worst case execution time [12].

VI. Conclusion

Moving from the analysis of technology and architectural trends in gigascale silicon integration, we observed that power consumption has become one of the most menacing showstoppers. Thus, leading edge low power design techniques are more critical than ever. Power management has emerged as the most promising approach to tackle the power crisis, because it makes it possible to finely control the power-performance tradeoff at run time. The paper surveyed hardware and software technologies for power management support, focusing more on fundamental ideas than on specific embodiments. Even though many solutions exist, much work still needs to be done especially in light of the evolution of SoC architectures toward communication-dominated networks-on-chip.

References

[1] L. Benini, A. Bogliolo, G. De Micheli, "A Survey of Design Techniques for System-Level Dynamic Power Management," *IEEE Transactions on Very Large-Scale Integration Systems*, vol. 8, no. 3, pp. 299-316, June 2000.

[2] L. Benini, G. De Micheli, "System-Level Power Optimization: Techniques and Tools," *ACM Transactions on Design Automation of Electronic Systems*, vol. 5, no. 2, pp. 115-192, April 2000.

[3] T. Burd, et al., "A dynamic voltage scaled microprocessor system," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 11, pp. 1571-1580, Nov. 2000.

[4] R. Ho, K. Mai, M. Horowitz, "The Future of wires," *Proceedings of the IEEE*, January 2001.

[5] T. Karnik, S. Borkar, V. De, "Sub-90nm technologies – Challenges and opportunities for CAD," *IEEE/ACM International Conference on Computer-Aided Design* pp. 203-206, Nov. 2002.

[6] D. Lackey, et al., "Managing power and performance for Systems-on-chip designs using voltage islands," *IEEE/ACM International Conference on Computer-Aided Design* pp. 195-202, Nov. 2002.

[7] Y.-H. Lu, T. Simunic, G. De Micheli, "Software controlled power management," *IEEE/ACM International Workshop on Hardware-Software Codesign*, pp. 157-161, Sept. 1999.

[8] Y.-H. Lu, G. De Micheli, "Comparing system level power management policies," *IEEE Design & Test of Computers*, vol. 18, no. 2, pp. 10-19, March-April 2001.

[9] S. Martin, K. Flautner, T. Mudge, D. Blaauw, "Combined dynamic Voltage scaling and Adaptive body Biasing for lower power microprocessors under dynamic workloads," *IEEE/ACM International Conference on Computer-Aided Design* pp. 721-727, Nov. 2002.

[10] P. R. Panda, F. Catthor, N. D. Dutt, K. Danckaert, E. Brockmeyer, C. Kulkarni,A. Vandercappele, P. G. Kjeldsberg, "Data and Memory Optimization Techniques for Embedded Systems", *ACM Transactions on Design Automation of Electronic Systems*, vol. 6, no. 2, pp. 149-206, April 2001.

[11] M. Pedram, J. Rabaey, *Power-aware design methodologies* Kluwer, 2002.

[12] D. Shin, J. Kim, S. Lee, "Intra-task voltage scheduling for low-energy hard real-time applications," *IEEE Design & Test of Computers*, vol. 18, no. 2 pp. 20-30, March-April 2001.

[13] T. Theis, "The future of Interconnection Technology," *IBM Journal of Research and Development*, Vol. 44, No. 3, May 2000, pp. 379-390.

[14] H. Zhang, V. Prabhu, V. George, M. Wan, M. Benes, A. Abnous, J. Rabaey, "A 1-V Heterogeneous Reconfigurable DSP IC for Wireless Baseband Digital Signal Processing," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 11, pp. 1697–1704, Nov. 2000.

[15] P. Zuchowski et al., "A hybrid ASIC and FPGA architecture," *IEEE/ACM International Conference on Computer-Aided Design* pp. 187-194, Nov. 2002.

AMD EX1021
AMD v. Realtek - Page 420

ATTACHMENT 8A

MARC Bibliographic Record

```
LEADER    -----cam-a22-----z--4500
001       991021970823002122
008       220628u2003uuuuuuuuu-|-o----u|----|eng-d
035         $a(CKB)1000000000022044
035         $a(EXLCZ)991000000000022044
210         $aDESIGN AUTOMATION CONFERENCE, 2003. PROCEEDINGS OF THE ASP-
            DAC 2003. ASIA AND SOUTH PACIFIC
245   00    $aProceedings of the ASP-DAC Asia and South Pacific Design
            Automation Conference, 2003
260         $bIEEE
776         $z0-7803-7659-5
906         $aPROCEEDING
```

MMS IDs

Document ID:

9911034116802121

Network Electronic IDs:

9911034116802121

Network Physical IDs:

mms_gb_ids:

991007108475002123

mms_mad_ids:

991021970823002122

mms_plt_ids:

991013474681102128

mms_rf_ids:

991013797575202129

ATTACHMENT 8B

OCLC 51906952 **No holdings in XXX - 14 other holdings**

Books

Rec Stat (**Ent** 2003 **Rep** 20230216
ered 0324 **laced** 152818.8

Type	a	**ELvl**		**Srce**	d	**Audn**		**Ctrl**		**Lang**	eng
BLvl	m	**Form**		**Conf**	1	**Biog**		**MRec**		**Ctry**	nju
		Cont	b	**GPub**		**LitF**	0	**Indx**	1		
Desc	a	**Ills**	a	**Fest**	0	**DtSt**	s	**Dates**	2003 ,		

010 2002112993
040 ZCU $b eng $c ZCU $d E9X $d DLC $d YDXCP $d OCLCF $d OCLCQ $d OCLCO $d OCL $d OCLCO $d OCLCA $d OCLCQ $d VYM $d OCLCO $d OCLCQ
019 52684633
020 0780376595 $q (pbk.)
020 9780780376595 $q (pbk.)
020 0780376609 $q (CD-ROM version)
020 9780780376601 $q (CD-ROM version)
050 00 TK7874 $b .A759 2003
082 00 621.3815 $2 22
090 $b
111 2 Asia and South Pacific Design Automation Conference $d (2003 : $c Kitakyūshū-shi, Japan)
245 10 Proceedings of the ASP-DAC 2003 : $b Asia and South Pacific Design Automation Conference 2003 : January 21-January 24, 2003, Kitakyushu International Conference Center, Kitakyushu, Japan / $c sponsored by IEEE Circuits and Systems Society [and others] ; supported by JEITA (Japan Electronics and Information Technologies Industry Association) [and others] ; in corporation with JIEP (Japan Institute of Electronics Packaging).
246 30 ASP-DAC 2003
260 Piscataway, N.J. : $b Institute of Electrical and Electronics Engineers, $c ©2003.
300 xl, 914 pages : $b illustrations ; $c 30 cm
336 text $b txt $2 rdacontent
337 unmediated $b n $2 rdamedia
338 volume $b nc $2 rdacarrier
500 "IEEE Catalog Number: 03EX627 (CD-ROM Version: 03EX627C)"--Title page verso
530 Also available on CD-ROM.
504 Includes bibliographical references and index.
530 Also available via the World Wide Web with additional title: Design Automation Conference, 2003, Proceedings of the ASP-DAC 2003, Asia and South Pacific.
650 0 Integrated circuits $x Computer-aided design $v Congresses.
650 0 Electronic circuit design $x Data processing $v Congresses.
650 6 Circuits intégrés $0 (CaQQLa)201-0032322 $x Conception assistée par ordinateur $0 (CaQQLa)201-0380628 $v Congrès. $0 (CaQQLa)201-0378219
650 7 Electronic circuit design $x Data processing. $2 fast $0 (OCoLC)fst00906866
650 7 Integrated circuits $x Computer-aided design. $2 fast $0 (OCoLC)fst00975538

655 2 Congress $0 (DNLM)D016423
655 7 Conference papers and proceedings. $2 fast $0 (OCoLC)fst01423772
655 7 Conference papers and proceedings. $2 lcgft
655 7 Actes de congrès. $2 rvmgf $0 (CaQQLa)RVMGF-000001049
710 2 IEEE Circuits and Systems Society.
710 2 Denshi Jōhō Gijutsu Sangyō Kyōkai.
710 2 Erekutoronikusu Jissō Gakkai.
740 0 Design Automation Conference, 2003, Proceedings of the ASP-DAC 2003, Asia and South Pacific.
856 41 $3 IEEE Xplore $u http://ieeexplore.ieee.org/servlet/opac?punumber=8504

ATTACHMENT 8C

LIBRARY OF CONGRESS AUTHORITIES

LC control no.: sh2009127317

LCCN Permalink: https://lccn.loc.gov/sh2009127317

HEADING: Integrated circuits Computer-aided design Congresses

000 00492cz a2200169n 450
001 8083879
005 20110729162807.0
008 091110|| anannbabn |n ana
010 __ |**a** sh2009127317
035 __ |**a** (DLC)444277
035 __ |**a** (DLC)sh2009127317
040 __ |**a** DLC |**b** eng |**c** DLC
150 __ |**a** Integrated circuits |**x** Computer-aided design |**v** Congresses
667 __ |**a** Record generated for validation purposes.
670 __ |**a** Work cat.: Proceedings of the ASP-DAC'99, 1999
906 __ |**t** 8888 |**u** tc00 |**v** 0
953 __ |**a** tc00

◀ Previous Next ▶

Save, Print or Email Records (View Help)

Select Download Format: Text (Labelled Display) Press to SAVE or PRINT

Email Text (Labelled Display) to: Press to SEND EMAIL

Library of Congress
URL: https://www.loc.gov/

Mailing Address:
101 Independence Ave, S.E.
Washington, DC 20540

Library of Congress Authorities
URL: http://authorities.loc.gov/
Library of Congress Online Catalog
URL: https://catalog.loc.gov/

Questions, comments, error reports: Contact Us

AMD EX1021
AMD v. Realtek - Page 428

ATTACHMENT 8D

LIBRARY OF CONGRESS AUTHORITIES

LC control no.: sh2008102924

LCCN Permalink: https://lccn.loc.gov/sh2008102924

HEADING: Electronic circuit design Data processing Congresses

000 00519cz a2200169n 450
001 7440870
005 20110729160214.0
008 080208|| anannbabn |n ana
010 __ |**a** sh2008102924
035 __ |**a** (DLC)378490
035 __ |**a** (DLC)sh2008102924
040 __ |**a** DLC |**b** eng |**c** DLC
150 __ |**a** Electronic circuit design |**x** Data processing |**v** Congresses
667 __ |**a** Record generated for validation purposes.
670 __ |**a** Work cat.: 7th Asia and South Pacific Design Automation Conference, c2002
906 __ |**t** 8888 |**u** te00 |**v** 0
953 __ |**a** te00

◀ Previous Next ▶

Save, Print or Email Records (View Help)

Select Download Format: Text (Labelled Display) Press to SAVE or PRINT

Email Text (Labelled Display) to: Press to SEND EMAIL

Library of Congress
URL: https://www.loc.gov/

Mailing Address:
101 Independence Ave, S.E.
Washington, DC 20540

Library of Congress Authorities
URL: http://authorities.loc.gov/
Library of Congress Online Catalog
URL: https://catalog.loc.gov/

Questions, comments, error reports: Contact Us

AMD EX1021
AMD v. Realtek - Page 430

ATTACHMENT 8E

Copyright United States Copyright Office

Try the Copyright Public Records System (CPRS) pilot with enhanced search features and filters.

Help | Search | History | Titles | Start Over

Public Catalog

Copyright Catalog (1978 to present)
Search Request: Left Anchored Title = Proceedings of the ASP-DAC 2003: Asia and South Pacific Design Automation Conference 2003
Search Results: Displaying 1 of 1 entries

previous next

Labeled View

Proceedings of the ASP-DAC 2003 : Asia and South Pacific design automation...

Type of Work: Text
Registration Number / Date: TX0005705561 / 2003-03-20
Title: Proceedings of the ASP-DAC 2003 : Asia and South Pacific design automation conference, 2003.
Description: 913 p.
Copyright Claimant: IEEE
Date of Creation: 2003
Date of Publication: 2003-01-21
Other Title: Asia and South Pacific design automation conference, 2003
Names: IEEE

previous next

Save, Print and Email (Help Page)		
Select Download Format	Full Record	Format for Print/Save
Enter your email address:		Email

Help Search History Titles Start Over

Contact Us | Request Copies | Get a Search Estimate | Frequently Asked Questions (FAQs) about Copyright | Copyright Office Home Page | Library of Congress Home Page

AMD EX1021
AMD v. Realtek - Page 432

ATTACHMENT 9

WTS Number: 320391

WTS
ARTICLE DELIVERY

Order #188610

Rush

Request Date: 04/18/2023 08:17

Requester: Sylvia Hall-Ellis - San Jose State University

Delivery email: **sylvia.hall-ellis@sjsu.edu**

Reference: AWPatrick

Instructions:

would like cover, table of contents, and article for journal, please

would like title page, verso, table of contents, and conference paper for proceedings, please

would like title page, verso, table of contents, and elected pages for monograph, please

WTS staff contact: **Karen**

Citation:

Ran, Yajun, and Malgorzata Marek-Sadowska. "Via-Configurable Routing Architectures and Fast Design Mappability Estimation for Regular Fabrics." In Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, 2005 (pp. 25-32). ICCAD '05. Piscataway, NJ: IEEE, 2005.

Library/Supplier: IEEE

ISSN/ISBN/OCLC: 078039254, 9780780392540

An outreach service of the General Library System, University of Wisconsin - Madison
http://wts.wisc.edu | wts@library.wisc.edu | 608.262.5917

ICCAD-2005

International Conference on Computer Aided Design

November 6 - 10, 2005
DoubleTree Hotel
San Jose, CA

Sponsored by:

IEEE IEEE CASS/CANDE | IEEE CIRCUITS & SYSTEMS SOCIETY® | CANDE | IEEE COMPUTER SOCIETY | acm | sigda

In cooperation with:

IEEE | IEEE ELECTRON DEVICES SOCIETY® | SSCS

AMD EX1021
AMD v. Realtek - Page 435

Proceedings of the 2005 International Conference on Computer-Aided Design

Copyright Information

Copyright and Reprint Permission: Abstracting is permitted with credit to the source. Libraries are permitted to photocopy beyond the limit of U.S. copyright law for private use of patrons those articles in this volume that carry a code a the bottom of the first page, provided the per-copy fee indicated in the code is paid through Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923. For other copying, reprint or republication permission, write to IEEE Copyrights Manager, IEEE Operations Center, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331. All rights reserved. Copyright © 2005 by the Institute of Electrical and Electronic Engineers, Inc.

The papers in this book comprise the proceedings of the meeting mentioned on the cover and title page. They reflect the authors' opinions and, in the interests of timely dissemination, are published as presented and without change. Their inclusion in this publication does not necessarily constitute endorsement by the editors, the IEEE Computer Society, or the Institute of Electrical and Electronics Engineers, Inc.

Catalog Numbers

IEEE Catalog Number: 05CH37700
ISBN Number: 0-7803-9254-X Softbound Edition
ISSN Number: 1092-3152

Additional Copies

Additional copies can be ordered from:

ACM Order Department
P.O. Box 11414
New York, New York, 10286-1414

E-mail: orders@acm.org

(For U.S.A. and Canada) (For all other countries)
Phone: 1-800-342-6626 Phone: +1-212-626-0500
Fax: 1-212-944-1318 Fax: +1-212-944-1318

ACM Order Number 477053

Printed in U.S.A.

Table of Contents

Conference Committee ..... **iii**
Foreword ..... **iv**
Awards ..... **v**
Technical Program Committee ..... **vi**
Reviewers ..... **viii**
Keynotes ..... **x**
Tutorial 1: **Characterization and Gate Modeling for 90NM and Below** ..... **xi**
Tutorial 2: **Package-Chip Co-Design** ..... **xii**
Tutorial 3: **ESL Design: Why the Time is Right and what are the Key Enabling Technologies** ..... **xiii**
Tutorial 4: **Simulation Technologies and Tools for the Nano Era** ..... **xiv**
Tutorial 5: **Yield Maximization Algorithms** ..... **xv**
Sunday Panel: **Exploiting Nanoscale Devices for Future Computing: Where Do We Go From Here?** ..... **xvi**
Monday Panel: **Chip Design in the Nano Era: is the Whole Bigger or Smaller than the Sum of the (Broken) Parts?** ..... **xvii**

SESSION 1A **Memory and Arithmetic Optimizations**
Moderators: ***Seda Ogrenci Memik - Northwestern Univ., Evanston, IL***
Soheil Ghiasi - Univ. of California, Davis, CA

1A.1 **Storage Assignment During High-Level Synthesis for Configurable Achitectures** **3**
Wenrui Gong, Gang Wang, Ryan Kastner

1A.2 **Performance-Driven Read-After-Write Dependencies Softening in High-Level Synthesis** **7**
Rafael Ruiz Sautua, Maria C. Molina, J.M. Mendias, R. Hermida

1A.3 **An Exact Algorithm for the Maximal Sharing of Partial Terms in Multiple Constant Multiplications** **13**
Paulo F. Flores, Jose C. Monteiro, Eduardo C. Costa

SESSION 1B **Design Manufacturing Interaction**
Moderators: ***Sani Nassif - IBM Corp., Austin, TX***
David Blaauw - Univ. of Michigan, Ann Arbor, MI

1B.1 **FPGA Device and Architecture Evaluation Considering Process Variations** **19**
Ho Yan P. Wong, Lerong Cheng, Yan Lin, Lei He

1B.2 **Via-Configurable Routing Architectures and Fast Design Mappability Estimation for Regular Fabrics** **25**
Yajun Ran, Malgorzata Marek Sadowska

1B.3 **SPIDER: Simultaneous Post-Layout IR-Drop and Metal Density Enhancement with Redundant Fill** **33**
Kwok Shing Leung

SESSION 1C **Detailed Placement**
Moderators: ***Yao-Wen Chang - National Taiwan Univ., Taipei, Taiwan***
Yegna Parasuram - Sierra Design Automation, Inc., Santa Clara, CA

1C.1 **Computational Geometry Based Placement Migration** **41**
Tao Luo, Haoxing Ren, Charles J. Alpert, David Z. Pan

1C.2 **An Efficient and Effective Detailed Placement Algorithm** **48**
Min Pan, Natarajan Viswanathan, Chris Chu

1C.3 **Post-Placement Rewiring and Rebuffering by Exhaustive Search for Functional Symmetries 56**
Kaihui Chang, Igor L. Markov, Valeria Bertacco

1C.4 **Wirelength Optimization by Optimal Block Orientation 64**
Xin Hao, Forrest Brewer

SESSION 1D **Digital, Analog and RF Test**
Moderators: ***Kartik Mohanram - Rice Univ., Houston, TX***

1D.1 **Parametric Test Development for RF Circuits Targeting Physical Fault Locations and Using Specification-Based Fault Definitions 73**
Erkan Acar, Sule Ozev

1D.2 **Response Shaper: A Novel Technique To Enhance Unknown Tolerance for Output Response Compaction 80**
Mango Chao, Seongmoon Wang, Srimat T. Chakradhar, Kwang-Ting Cheng

1D.3 **Test Planning for the Effective Utilization of Port-Scalable Testers for Heterogeneous Core-Based SOCs 88**
Anuja Sehgal, Krishnendu Chakrabarty

1D.4S **Cocktail Scan Based on Random Access Scan Toward Low Power and High Efficiency Test 94**
Shih Ping Lin, Chung Len Lee, Jwu E. Chen

1D.5S **A Statistical Study of the Effectiveness of BIST Jitter Measurement Techniques 100**
David A. Bordoley, Hieu Nguyen, Mani Soma

SESSION 2A **Embedded Tutorial: Design Trends**
Moderators: ***Borivoje Nikolic - Univ. of California, Berkeley, CA***
Thomas Burd - Consultant, Berkeley, CA

2A.1 **The Circuit Design of the Synergistic Processor Element of a Cell Processor 111**
Osamu Takahashi, R. Cook, S. Cottier, S.H. Dhong, B. Flachs, K. Hirairi, A. Kawasumi, H. Murakami, N. Noro, H. Oh, S. Onishi, J. Pille, J. Silberman

2A.2 **Adaptive Designs for Power and Thermal Optimization 118**
Richard McGowen

2A.3 **Digital RF Processor (DRP) for Cellular Phones 122**
Robert Bogdan Staszewski, Khurram Muhammed, Dirk Leipold

SESSION 2B **Physical Design for Manufacturing**
Moderators: ***Duane S. Boning - Massachusetts Institute of Tech., Cambridge, MA***
Andrew B. Kahng - Univ. of California at San Diego, La Jolla, CA

2B.1 **A Layout Dependent Full-Chip Copper Electroplating (ECP) Topography Model 133**
Jianfeng Luo, Qing Su, Charles Chiang, Jamil Kawa

2B.2 **Interval-Valued Statistical Modeling of Oxide Chemical-Mechanical Polishing 141**
James Ma, Claire Fang, Rob A. Rutenbar, Xiaolin Xie, Duane S Boning

2B.3 **Fast and Efficient Phase Conflict Detection and Correction in Standard-Cell Layouts 149**
Charles Chiang, Andrew B. Kahng, Subarna Sinha, Xu Xu

SESSION 2C **Large-Scale Layout Techniques**
Moderators: ***Igor Markov - Univ. of Michigan, Ann Arbor, MI***
David Z. Pan - Univ. of Texas, Austin, TX

2C.1 **IMF: Interconnect-Driven Multilevel Floorplanning for Large-Scale Building-Module Designs 159**
Tungchieh Chen, Yao-Wen Chang, Shyh-Chang Lin

2C.2 **Robust Mixed-Size Placement Under Tight White-Space Constraints 165**
Jason Cong, Michail Romesis, Joseph Shinnerl

2C.3 **Intrinsic Shortest Path Length: A New, Accurate A Priori Wirelength Estimator 173**
Sherief Reda, Andrew B. Kahng

SESSION 2D **Novel Ideas and Logic Synthesis**
Moderators: ***Michel Berkelaar - Magma Design Automation, Inc., Eindhoven, Netherlands***
Andreas Kuehlmann - Cadence Design Systems, Inc., Berkeley, CA

2D.1 **Synthesis Methodology for Built-In At-Speed Testing 183**
Yinghua Li, Alex Kondratyev, Robert K. Brayton

2D.2 **Clustering for Processing Rate Optimization 189**
Chuan Lin, Hai Zhou, Jia Wang

2D.3 **ConvexFit: An Optimal Minimum-Error Convex Fitting and Smoothing Algorithm with Application to Gate-Sizing 196**
Sanghamitra C. Roy, Weijen Chen, Charlie Chen

SESSION 3A **Embedded Tutorial: Opportunities and Challenges with Double-Gated Devices**
Moderators: ***Tanay Karnik - Intel Corp., Hillsboro, OR***
Kevin Nowka - IBM Corp., Austin, TX

3A.1 **FinFETs for Nanoscale CMOS Digital Integrated Circuits 207**
Tsu-Jae King

3A.2 **Physics-Based Compact Modeling for Nonclassical CMOS 211**
Vishal P. Trivedi, J.G. Fossum, L. Mathew, M.M. Chowdhury, W. Zhang, G.O. Workman, B.Y. Nguyen

3A.3 **Double-Gate SOI Devices for Low-Power and High-Performance Applications 217**
Kaushik Roy, Hamid Mahmoodi, Saibal Mukhopadhyay, Hari Ananthan, Aditya Bansal, Tamer Cakici

3A.4 **Thermal Simulation Techniques for Nano Transistors 225**
Jeremy Rowlette, Eric Pop, Sanjiv Sinha, Mathew Panzer, Kenneth E. Goodson

SESSION 3B **Routing and Application Specific NoC Architectures**
Moderators: ***Marcello Lajolo - NEC-Labs America, Inc., Princeton, NJ***
Tony Givargis - Univ. of California, Irvine, CA

3B.1 **An Automated Technique for Topology and Route Generation of Application Specific On-Chip Interconnection Networks 231**
Krishnan Srinivasan, Karam S. Chatha, Goran Konjevod

3B.2 **Deadlock-Free Routing and Component Placement for Irregular Mesh-based Networks-on-Chip 238**
Martin Schäfer, Thomas Hollstein, Heiko Zimmer, Manfred Glesner

3B.3 **Application-Specific Network-on-Chip Architecture Customization via Long-Range Link Insertion 246**
Umit Y. Ogras, Radum Marculescu

3B.4 **NoCEE : Energy Macro-Model Extraction Methodology for Network on Chip Routers 254**
Jeremy Chan, Sri Parameswaran

SESSION 3C **Memory Driven Code and Architecture Optimizations**
Moderator: ***Margarida Jacome - Univ. of Texas, Austin, TX***

3C.1 **Architecture and Compilation for Data Bandwidth Improvement in Configurable Embedded Processors 263**
Jason Cong, Guoling Han, Zhiru Zhang

3C.2 **Code Restructuring for Improving Cache Performance of MPSoCs 271**
Guilin Chen, Mahmut Kandemir

3C.3 **Two-Dimensional Data Locality: Definition, Abstraction, and Application 275**
Mahmut Kandemir

3C.4 **Integrating Loop and Data Optimizations for Locality within a Constraint Network Based Framework 279**
Guilin Chen, Ozcan Ozturk, Mahmut Kandemir, Ibrahim Kolcu

SESSION 3D **Exploiting Arithmetic Constructs in Verification**
Moderators: ***Geert Janssen - IBM Corp., Yorktown Heights, NY***
Yirng-An Chen - Synopsys, Inc., Mountain View, CA

3D.1 **System Level Verification of Digital Signal Processing Applications Based on the Polynomial Abstraction Technique 285**
Tarvo Raudvere, Ashish Kumar Singh, Ingo Sander, Axel Jantsch

3D.2 **Equivalence Verification of Polynomial Datapaths with Fixed-Size Bit-Vectors using Finite Ring Algebra 291**
Namrata Shekhar, Priyank Kalla, Florian Enescu, Sivaram Gopalakrishnan

3D.3 **RTL SAT Simplification by Interval Arithmetic Reasoning 297**
Ganapathy Parthasarathy, Madhu K. Iyer, Kwang T. Cheng, Forrest Brewer

3D.4 **Runtime Integrity Checking for Inter-Object Connections 303**
Guilin Chen, Mahmut Kandemir

SESSION 4A **Buffers and Voltage Islands**
Moderators: ***Jiang Hu - Texas A&M Univ., College Station, TX***
Charles Chiang - Synopsys, Inc., Mountain View, CA

4A.1 **Optimal Post-Placement Voltage Island Generation Under Performance Requirement 309**
Huaizhi Wu, IMin Liu, Yusu Wang, Martin Wong

4A.2 **Buffer Insertion Under Process Variations for Delay Minimization 317**
Liang Deng, Martin Wong

4A.3 **Efficient Algorithms for Buffer Insertion in General Circuits Based on Network Flow 322**
Ruiming Chen, Hai Zhou

SESSION 4B **Sequential Circuit Optimization**
Moderators: ***Victor Kravets - IBM Corp., Yorktown, NY***
Narendra Shenoy - Synopsys, Inc., Mountain View, CA

4B.1 **Trade-off between Latch and Flop for Min-Period Sequential Circuit Designs with Crosstalk 329**
Chuan Lin, Hai Zhou

4B.2 **Flip-Flop Insertion With Shifted-Phase Clocks For FPGA Power Reduction 335**
Hyeonmin Lim, Kyungsoo Lee, Youngjin Cho, Naehyuck Chang

4B.3 **Acyclic Modeling of Combinational Loops 343**
Amit Gupta, Charley Selvidge

SESSION 4C **Power Grid Verification**
Moderators: ***Rajendran V. Panda - Freescale Semiconductor, Inc., Austin, TX***
Eli Chiprout - Intel Corp., Chandler, AZ

4C.1 **Fast Algorithms for IR Drop Analysis in Large Power Grid 351**
Yu Zhong, Martin D.F. Wong

4C.2 **Incremental Partitioning-Based Vectorless Power Grid Verification 358**
Dionysios Kouroussis, Imad A. Ferzli, Farid N. Najm

4C.3 **Static Timing Analysis Considering Power Supply Variations 365**
Sanjay Pant, David Blaauw

SESSION 4D **Nanoelectronics**
Moderators: ***Sandeep Shukla - Virgina Tech., Blacksburg, VA***
Krishnendu Chakrabarty - Duke Univ., Durham, NC

4D.1 **Hybrid Semiconductor/Nanoelectronic Circuits: Devices, Architectures, and Design Automation 375**
Andre' DeHon, Konstantin K. Likharev

4D.2 **Performance Analysis of Carbon Nanotube Interconnects for VLSI Applications 383**
Navin Srivastava, Kaustav Banerjee

SESSION 5A **Variability in Design**
Moderators: ***Kevin Nowka - IBM Corp., Austin, TX***
Thomas Burd - Consultant, Berkeley, CA

5A.1 **DiCER: Distributed and Cost-Effective Redundancy for Variation Tolerance 393**
Di Wu, Ganesh Venkataraman, Qiuyang Li, Jiang Hu, Rabi N Mahapatra

5A.2 **Worst-Case Analysis to Obtain Stable Read/Write DC Margin of High Density 6T-SRAM-Array with Local Vth Variability 398**
Yasumasa Tsukamoto, Nii Koji, Susumu Imaoka, Yuji Oda, Shigeki Ohbayashi, Tomoaki Yoshizawa, Hiroshi Makino, Koichiro Ishibashi, Hirofumi Shinohara

5A.3 **Noise Margin Analysis for Dynamic Logic Circuits 406**
Suwen Yang, Mark R. Greenstreet

SESSION 5B **Efficient Analog Design Space Exploration Techniques**
Moderators: ***Mar Hershenson - Sabio Labs Inc., Palo Alto, CA***
L. Miguel Silveira - Univ. of Lisbon, Lisbon, Portugal

5B.1 **Efficient Analog Platform Characterization Through Analog Constraint Graphs 415**
Fernando De Bernardinis, Alberto Sangiovanni Vincentelli

5B.2 **Performance-Centering Optimization for System-Level Analog Design Exploration 422**
Xin Li, Jian Wang, Lawrence T. Pileggi, Wanju Chiang, Tun-Shin Chen

5B.3 **Hierarchical Performance Macromodels of Feasible Regions for Synthesis of Analog and RF Circuits 430**
Anuradha Agarwal, Ranga Vemuri

SESSION 5C **Dynamic Voltage Scaling**
Moderators: ***Sandeep Shukla - Virginia Tech., Blacksburg, VA***
Tajana Simunic Rosing - Univ. of California, Palo Alto, CA

5C.1 **Battery Optimization vs. Energy Optimization: Which to Choose and When? 439**
Ravishankar Rao, Sarma Vrudhula, Naehyuck Chang

5C.2 **Dynamic Voltage Scaling for the Schedulability of Jitter-Constrained Real-Time Embedded Systems** **446**
Bren C. Mochocki, Xiaobo S. Hu, Razvan Racu, Rolf Ernst

5C.3 **Optimal Integration of Inter-Task and Intra-Task Dynamic Voltage Scaling Techniques for Hard Real-Time Applications** **450**
Jaewon Seo, Nikil D. Dutt, Taewhan Kim

5C.4 **Compiler-Directed Voltage Scaling on Communication Links for Reducing Power Consumption** **456**
Feihui Li, Guangyu Chen, Mahmut Kandemir

SESSION 5D **Biochips and DNA-Based Nanofabrication**
Moderators: ***Aridam Mukherjee - Univ. of North Carolina, Charlotte, NC***
André DeHon - California Institute of Tech., Pasadena, CA

5D.1 **Design Automation Issues for Biofluidic Microchips** **463**
Tamal Mukherjee

5D.2 **Design of DNA Origami** **471**
Paul W.K. Rothemund

5D.3 **Kauffman Networks: Analysis and Applications** **479**
Elena V. Dubrova, Maxim Teslenko, Andres Martinelli

SESSION 6A **Efficient Simulation and Synthesis Methodologies for Analog Circuits**
Moderators: ***Gu-Yeon Wei - Harvard Univ., Cambridge, MA***
Jaijeet Roychowdhury - Univ. of Minnesota, Minneapolis, MN

6A.1 **Parameterized Model Order Reduction For Nonlinear Dynamical Systems** **487**
Brad Bond, Luca Daniel

6A.2 **Fast-Yet-Accurate PVT Simulation by Combined Direct and Iterative Methods** **495**
C.J. Richard Shi, Bo Hu

6A.3 **Robust Automated Synthesis Methodology for Integrated Spiral Inductors with Variability** **502**
Arthur B. Nieuwoudt, Yehia Massoud

SESSION 6B **Technology Mapping and Timing Analysis**
Moderators: ***Ankur Srivastava - Univ. of Maryland, College Park, MD***
Mukul Prasad - Fujitsu, Sunnyvale, CA

6B.1 **Statistical Technology Mapping for Parametric Yield** **511**
Ashish K. Singh, Murari Mani, Michael Orshansky

6B.2 **Reducing Structural Bias in Technology Mapping** **519**
Satrajit Chatterjee, Alan Mishchenko, Robert Brayton, Xinning Wang, Timothy Kam

6B.3 **Improving the Efficiency of Static Timing Analysis with False Paths** **527**
Shuo Zhou, Bo Yao, Hongyu Chen, Chungkuan Cheng, Mike Hutton, Truman Collins, Sridhar Srinivasan, Nanchi Chou, Peter Suaris

SESSION 6C **Power Aware System Architecture and Software Optimizations**
Moderators: ***Elaheh Bozorgzadeh - Univ. of California, Irvine, CA***
Youngsoo Shin - KAIST, Daejeon, Korea

6C.1 **Total Power-Optimal Pipelining and Parallel Processing Process Variations in Nanometer Technology** **535**
Nam Sung Kim, Taeho Kgil, Keith Bowman, Vivek De, Trevor Mudge

6C.2 **Serial-Link Bus: A Low-Power On-Chip Bus Architecture 541**
Maged M. Ghoneima, Muhammad Khellah, James Tschanz, Yehea Ismail, Vivek De

6C.3 **New Decompilation Techniques for Binary-level Co-processor Generation 547**
Greg Stitt, Frank Vahid

SESSION 6D **Cellular Array Architectures**
Moderators: ***Konstantin Likharev - SUNY, Stony Brook, NY***
Rick Kiehl - Univ. of Minnesota, Minneapolis, MN

6D.1 **Cellular Wave Computers and CNN Technology - An SoC architecture with xK Processors and Sensors 557**
Tomas Roska

6D.2 **Eliminating Wire Crossings for Molecular Quantum-dot Cellular Automata Implementation 565**
Amitabh Chaudhary, Danny Chen, X Sharon Hu, Michael T Niemier, Ramprasa Ravichandran, Kevin Whitton

SESSION 7A **Variability Aware Clocking**
Moderators: ***Martin D.F. Wong - Univ. of Illinois at Urbana-Champaign, Urbana, IL***
Ali Dasdan - Synopsys, Inc., Mountain View, CA

7A.1 **Statistical Timing Analysis Driven Post-Silicon-Tunable Clock-Tree Synthesis 575**
Jengliang Tsai, Lizheng Zhang, Charlie Chen

7A.2 **TACO: Temperature Aware Clock Tree Optimization 582**
Minsik Cho, Suhail Ahmed, David Z. Pan

7A.3 **Statistical Based Link Insertion for Robust Clock Network Design 588**
Douglas Lam, Cheng Kok Koh, Yiran Chen, Jitesh Jain, V. Balakrishnan

7A.4 **Practical Techniques for Minimizing Skew and Its Variation in Buffered Clock Networks 592**
Ganesh Venkataraman, Nikhil Jayakumar, Jiang Hu, Peng Li, Sunil khatri, Anand Rajaram, Charles Alpert, Patrick McGuinness

SESSION 7B **Oscillator Analysis**
Moderators: ***Jacob White - Massachucetts Institute of Tech., Cambridge, MA***
Alper Demir - KOC Univ., Sariyer-Istanbul, Turkey

7B.1 **An Efficient and Robust Technique for Tracking Amplitude and Frequency Envelopes in Oscillators 599**
Ting Mei, Jaijeet Roychowdhury

7B.2 **Oscillator-AC: Restoring Rigour to Linearized Small-Signal Analysis of Oscillators 604**
Ting Mei, Jaijeet Roychowdhury

7B.3 **A Multi-Harmonic Probe Technique for Computing Oscillator Steady States 610**
Kapil D. Boianapally, Jaijeet Roychowdhury, Ting Mei

7B.4S **An Efficient and Accurate Algorithm for Autonomous Envelope Following with Applications 614**
Tianrong Zhang, Dan Feng

7B.5S **Steady-State Analysis of Voltage and Current Controlled Oscillators 618**
Amit Mehrotra, Suihua Lu, David C. Lee, Amit Narayan

SESSION 7C **Power Noise and Thermal Issues**
Moderators: ***Syed Alam - Freescale Semiconductor, Inc., Austin, TX***
Emrah Acar - IBM Corp., Yorktown Heights, NY

7C.1 **Timing-Aware Power Noise Reduction in Layout 627**
Chao Yang Yeh, Malgorzata Marek Sadowska

7C.2 **High Efficiency Full-Chip Thermal Simulation Algorithm 635**
Yong Zhan, Sachin S. Sapatnekar

7C.3 **Fast Thermal Simulation for Architecture Level Dynamic Thermal Management 639**
Pu Liu, Hang Li, Lingling Jin, Wei Wu, Sheldon Tan, Jun Yang, Zhenyu Qi

7C.4 **Variational Analysis of Large Power Grids by Exploring Statistical Sampling Sharing and Spatial Locality 645**
Peng Li

SESSION 7D **Nanocomputing**
Moderators: ***Andras Moritz - Univ. of Massachusetts, Amherst, MA***
Iris Bahar - Brown University, Providence, RI

7D.1 **The Impact of the Nanoscale on Computing Systems 655**
Seth Goldstein

7D.2 **Computer-Aided Design for DNA Self-Assembly: Process and Applications 662**
Chris Dwyer

7D.3 **A Mapping Algorithm for Defect-Tolerance of Reconfigurable Nano-Architectures 668**
Mehdi Baradaran Tahoori

SESSION 8A **Extraction and Modeling for Interconnect Structures**
Moderators: ***Vikram Jandhyala - Univ. of Washington, Seattle, WA***
Dipanjan Gope - Intel Corp., Santa Clara, CA

8A.1 **FastSies: A Fast Stochastic Integral Equation Solver for Modeling the Rough Surface Effect 675**
Zhenhai Zhu, Jacob K. White

8A.2 **Efficient Statistical Capacitance Variability Modeling with Orthogonal Principle Factor Analysis 683**
Rong Jiang, Wenyin Fu, Charlie Chung Ping Chen, Vince Lin, Janet Wang

8A.3 **Reducing Pessimism in RLC Delay Estimation Using an Accurate Analytical Frequency Dependent Model for Inductance 691**
Mosin Mondal, Yehia Massoud

SESSION 8B **Timing and Power Optimization**
Moderators: ***Louis Scheffer - Cadence Design Systems, Inc., San Jose, CA***
Yehea Ismail - Northwestern Univ., Evanston, IL

8B.1 **Statistical Critical Path Analysis Considering Correlations 699**
Yaping Zhan, Andrzej J. Strojwas, Mahesh Sharma, David Newmark

8B.2 **Discrete Vt Assignment and Gate Sizing Using a Self-Snapping Continuous Formulation 705**
Saumil Shah, Ashish Srivastava, Dushyant Sharma, Vladimir Zolotov, Dennis Sylvester, David Blaauw

8B.3 **Formalizing Designer's Preferences for Multiattribute Optimization with Application to Leakage-Delay Tradeoffs 713**
Sarvesh Bhardwaj, Sarma Vrudhula

SESSION 8C **System-Level Variability Modeling**
Moderators: ***Yu Kevin Cao - Arizona State Univ., Tempe, AZ***
Emrah Acar - IBM Corp., Yorktown Heights, NY

8C.1 **Projection-Based Performance Modeling for Inter/Intra-Die Variations 721**
Xin Li, Jiayong Le, Lawrence T. Pileggi, Andrzej Strojwas

8C.2 **System-Level Power and Thermal Modeling by Orthogonal Polynomial based Response Surface Approach (OPRS) 728**
Janet M. Wang, Bharat Srinivas, Dongsheng Ma, Charlie Chung-Ping Chen, Jun Li

8C.3 **Accurate Estimation and Modeling of Total Chip Leakage Considering Inter- and Intra-Die Process Variations 736**
Amit Agarwal, Kunhyuk Kang, Kaushik Roy

SESSION 8D **Routing**
Moderators: ***Hardy K.S. Leung - Magma Design Automation, Inc., Santa Clara, CA***
Kevin Chao - Intel Corp., Hillsboro, OR

8D.1 **Thermal Via Planning for 3-D ICs 745**
Jason Cong, Yan Zhang

8D.2 **A Routing Algorithm for Flip-Chip Design 753**
Jia-Wei Fang, I-Jye Lin, Ping-Hung Yuh, Yao-Wen Chang, Jyh-Herng Wang

8D.3S **An Escape Routing Framework for Dense Boards with High-Speed Design Constraints 759**
Muhammet M. Ozdal, Martin D.F. Wong, Philip Honsinger

8D.4S **Optimal Routing Algorithms for Pin Clusters in High-Density Multichip Modules 767**
Muhammet M. Ozdal, Martin D.F. Wong, Philip Honsinger

SESSION 9A **New Frontiers in High-Level Synthesis**
Moderators: ***Anmol Mathur - Calypto Design Systems, Inc., Santa Clara, CA***
James Hoe - Carnegie Mellon Univ., Pittsburgh, PA

9A.1 **Weighted Control Scheduling 777**
Aravind Vijayakumar, Forrest D. Brewer

9A.2 **Hardware Synthesis from Guarded Atomic Actions with Performance Specifications 784**
Daniel L. Rosenband, Prof. Arvind

9A.3S **Fast Timing Closure by Interconnect Criticality Driven Delay Relaxation 792**
Love Singhal, Elaheh Bozorgzadeh

SESSION 9B **Advances in Model Order Reduction Approaches**
Moderators: ***Ram Achar - Carleton Univ., Ottawa, Canada***
Joel Phillips - Cadence Berkeley Labs, Berkeley, CA

9B.1 **Fast Balanced Stochastic Truncation Via A Quadratic Extension of the Alternating Direction Implicit Iteration 801**
Ngai Wong, Venkataramanan Balakrishnan

9B.2 **Parameterized Interconnect Order Reduction with Explicit-and-Implicit Multi-Parameter Moment Matching for Inter/Intra-Die Variations 806**
Xin Li, Peng Li, Lawrence T Pileggi

9B.3 **A More Reliable Reduction Algorithm for Behavioral Model Extraction 813**
Dmitry G. Vasilyev, Jacob K. White

9B.4 **An Efficient Method for Terminal Reduction of Interconnect Circuits Considering Delay Variations** **821**
Pu Liu, Tan Sheldon, Hang Li, Zhenyu Qi, Jun Kong, Bruce McGaughy, Lei He

SESSION 9C **Statistical Timing Analysis**
Moderators: ***Florentin Dartu - Intel Corp., Hillsboro, OR***
Tom Spyrou - Precision Software Development, Sunnyvale, CA

9C.1 **Statistical Timing Analysis With Two-sided Constraints** **829**
Khaled R. Heloue, Farid N. Najm

9C.2 **An Unified Framework for Statistical Timing Analysis With Coupling and Multiple Input Switching** **837**
Debjit Sinha, Hai Zhou

9C.3 **Defining Statistical Sensitivity for Timing Optimization of Logic Circuits With Large-Scale Process and Environmental Variations** **844**
Xin Li, Jiayong Le, Mustafa Celik, Lawrence T. Pileggi

SESSION 9D **Problem Structure in Formal Verification**
Moderators: ***James H. Kukula - Synopsys, Inc., Hillsboro, OR***
David Cyrluk - Kestrel Institute, Palo Alto, CA

9D.1 **Verification of Executable Pipelined Machines with Bit-Level Interfaces** **855**
Pete Manolios, Sudarshan K Srinivasan

9D.2 **A Complete Compositional Reasoning Framework for the Efficient Verification of Pipelined Machines** **863**
Pete Manolios, Sudarshan K. Srinivasan

9D.3 **Post-Verification Debugging of Hierarchical Designs** **871**
Moayad Fahim Ali, Sean Safarpour, Andreas Veneris, Magdy S. Abadir, Rolf Drechsler

9D.4 **Efficient LTL Compilation for SAT-based Model Checking** **877**
Roy Armoni, Sergey V Egorov, Ranan Fraer, Dmitry Korchemny, Moshe Y Vardi

9D.5 **SAT based solutions for Consistency Problems in Formal Property Specifications for Open Systems** **885**
Suchismita Roy, Sayantan Das, Prasenjit Basu, Pallab Dasgupta, Partha P. Chakrabarti

SESSION 10A **Analytical Placement**
Moderator: ***Bill Halpin - Synplicity, Inc., Sunnyvale, CA***

10A.1 **Architecture and Details of a High Quality, Large-Scale Analytical Placer** **891**
Andrew B. Kahng, Sherief Reda, Qinke Wang

10A.2 **Mixed-Size Placement via Line Search** **899**
Kristofer Vorwerk, Andrew Kennings

10A.3 **A Hybrid Linear Equation Solver and its Application in Quadratic Placement** **905**
Haifeng Qian, Sachin S. Sapatnekar

SESSION 10B **Embedded Tutorial: Hardware and Software Design of Energy Efficient Sensor PlatForms**
Moderator: ***Anand Raghunathan - NEC Labs America, Inc., Princeton, NJ***

10B.1 **Energy-Efficient Platform Designs for Real-World Wireless Sensing Applications** **913**
Pai H. Chou, Chulsung Park

10B.2 **Power-Aware Microsensor Design** **921**
Brian Schott, Michael Bajura

10B.3 **System Software Techniques for Low-Power Operation in Wireless Sensor Networks 925**
David E. Culler, Prabal K. Dutta

SESSION 10C **Improving the Accuracy of Static Timing Analysis**
Moderators: ***Tom Spyrou - Precision Software Development, Inc., Sunnyvale, CA***
Frank Liu - IBM Corp., Austin, TX

10C.1S **Expanding the Frequency Range of AWE via Time Shifting 935**
Ahmed M. Shebaita, Chirayu S. Amin, Florentin Dartu, Yehea Ismail

10C.2S **A Sliding Window Scheme for Accurate Clock Mesh Analysis 939**
Hongyu Chen, Chao Yang Yeh, Subodh M. Reddy, Gustavo Wilke, Hoa Van Nguyen, William W. Walker, Rajeev Murgai

10C.3 **Accurate Delay Computation for Noisy Waveform Shapes 947**
Amit Jain, Vladimir Zolotov, David Blaauw

10C.4 **Pessimism Reduction in Crosstalk Noise Aware STA 954**
Murat R. Becer, Vladimir Zolotov, Rajendran Panda, Amir Grinshpon, Ilan Algor, Rafi Levy, Chanhee Oh

SESSION 10D **Embedded Tutorial : Formal Equivalence Checking Between System-Level Models and RTL**
Moderator: ***Anmol Mathur - Calypto Design Systems, Inc., Santa Clara, CA***

10D.1 **Formal equivalence checking between system-level models and RTL 965**
Alfred Koebel, Yuan Lu, Anmol Mathur

SESSION 11A **Embedded Tutorial: Emergent Communication**
Moderators: ***Vikram Jandhyala – Univ of Washington, Seattle, WA***
L. Miguel Silveira – INESC ID/IST, Technical Univ. of Lisbon, Portugal

11A.1 **CDMA/FDMA-Interconnects for Future ULSI Communications 975**
Frank Chang

11A.2 **The Feasibility of On-Chip Interconnection Using Antennas 979**
Ken O, K. Kim, B. Floyd, J. Mehta, H. Yoon, C.M. Hung, D. Bravo, T. Dickson, X. Guo, R. Li, N. Trichy, J. Caserta, W. Bomstad, J. Branch, D.J. Yang, J. Bohorquez, J. Chen, E.Y. Seok, L. Gao, A. Sugavanam, J.J. Lin, S. Yu, C. Cao, M.H. Hwang, Y.P. Ding, S.H. Hwang, H. Wu, N. Zhang, J.E. Brewer

11A.3 **Global Signaling Over Lossy Transmission Lines 985**
Michael P. Flynn, Joshua J. Kang

SESSION 11B **Addressing Emerging Challenges for SoCs**
Moderators: ***Wolfgang Nebel - Oldenburg Univ. and OFFIS, Oldenburg, Germany***
Margarida Jacome - Univ. of Texas, Austin, TX

11B.1 **A Cache-Defect-Aware Code Placement Algorithm for Improving the Performance of Processors 995**
Tohru Ishihara, Farzan Fallah

11B.2 **Improving Scratch-Pad Memory Reliability Through Compiler-Guided Data Block Duplication 1002**
Feihui Li, Guangyu Chen, Mahmut Kandemir, I Kolcu

11B.3 **Multi-Clock Latency-Insensitive Architecture and Wrapper Synthesis 1006**
Ankur Agiwal, Montek Singh

11B.4 **Memory Access Optimization of Dynamic Binary Translation for Reconfigurable Architectures 1014**
Sejong Oh, Tag Gon Kim

SESSION 11C **Statistical Optimization**
Moderators: ***Nagib Z. Hakim - Intel Corp., Santa Clara, CA***
Sachin Sapatnekar - Univ. of Minnesota, Minneapolis, MN

11C.1 **Parametric Yield Maximization using Gate Sizing based on Efficient Statistical Power and Delay Gradient Computation 1023**
Kaviraj Chopra, Saumil Shah, Ashish Srivastav, Dennis Sylvester, David Blaauw

11C.2 **Gate Sizing using Incremental Parameterized Statistical Timing Analysis 1029**
Matthew R. Guthaus, Natesan Venkateswaran, Chandu Visweswariah, Vladimir Zolotov

11C.3 **Timing Yield Driven Statistical Gate Sizing Optimization 1037**
Debjit Sinha, Narendra V. Shenoy, Hai Zhou

SESSION 11D **Making Model Checking Practical**
Moderators: ***Valeria Bertacco - Univ. of Michigan, Ann Arbor, MI***
Vigyan Singhal - Jasper Design Automation, Inc., Mountain View, CA

11D.1 **Simulation-based Bug Trace Minimization with BMC-based Refinement 1045**
Kaihui Chang, Valeria Bertacco, Igor L Markov

11D.2 **Complementary Use of Runtime Validation and Model Checking 1052**
Ali A. Bayazit, Sharad Malik

11D.3 **Scalable Compositional Minimization via Static Analysis 1060**
Fadi A. Zaraket, Jason Baumgartner, Adnan Aziz

11D.4 **Transition-by-Transition FSM Traversal for Reachability Analysis in Bounded Model Checking 1068**
Minh D. Nguyen, Dominik A Stoffel, Wolfgang Kunz, Markus Wedler

11D.5 **Automatic Phase Abstraction for Formal Verification 1076**
Per Bjesse, James Kukula

This material may be protected by Copyright law (Title 17 U.S. Code)

Via-Configurable Routing Architectures and Fast Design Mappability Estimation for Regular Fabrics

Yajun Ran
Department of Electrical and Computer Engineering
University of California, Santa Barbara, CA 93106
Email: ranyj@ece.ucsb.edu

Malgorzata Marek-Sadowska
Department of Electrical and Computer Engineering
University of California, Santa Barbara, CA 93106
Email: mms@ece.ucsb.edu

Abstract— **In this paper, we describe a new via-configurable routing architecture which shows much better throughput and performance than the previous structures. We demonstrate how to construct a single-via-mask fabric to reduce further the mask cost, and we analyze the penalties which it incurs. To solve the routability problem commonly existing in fabric-based designs, an efficient white-space allocation scheme is suggested, which provides a fast design convergence and early prediction of the circuit mappability to a given fabric.**

I. Introduction

In recent years, rapid technological progress has made it possible to integrate more than one-billion transistors on a single chip. However, the manufacturability issues have presented a serious challenge for nanometer technologies. Pattern distortion in sub-wavelength lithography requires tremendous effort for post-layout optical proximity correction (OPC). Erosion and dishing in chemical-mechanical polishing (CMP) processes can cause significant interconnect/dielectric thickness variations. Tackling these problems in the design stage is not easy since they are related to neighborhood layout patterns. An accurate, efficient analysis of layout patterns is difficult to do during design optimization steps. Moreover, the extremely high cost of design and manufacturing has effectively dissuaded designers from using the most advanced technologies.

An alternative solution to these problems is to design circuits based on pre-fabricated fabrics, i.e. *structured-ASICs*. A fabric usually consists of repeated logic blocks. It has pre-fabricated transistors and partially fixed metal layers. Due to the high layout-pattern regularity, a fabric is much easier to manufacture. A base block in a fabric can be well characterized, and OPC effort for a block is small. Also, the fixed metal segments are uniformly distributed and thus help to reduce variations caused by the CMP. Moreover, a designer needs only to customize a few metal and/or via masks to complete a design, which can greatly reduce the mask cost as well as the design and manufacturing time. With the use of hard-wired switches, the performance can be significantly improved over that of FPGA-based designs.

There are many types of structured-ASICs. Several base logic blocks have been proposed in the literature, such as a via-configurable lookup table (LUT) [1], LUT/NAND3-mixed block [2], gain-based semi-universal-logic block [3]. There are also works on fixed routing structures such as direct-connected crossbar [1] and via-configurable routing [4]. However, the literature offers little discussion about the throughput of existing routing structures. For a fabric with fixed wire segments, routability is a more serious problem than in customized routing. The inflexibility of pre-designed wiring may significantly degrade circuit performance and require more chip area.

Given fixed logic and routing structures in a fabric, a densely-packed design might not be routable. To solve this problem, some blocks might be left only partially used to relieve congestion. This is called white-space allocation in cell-based designs. White-space in a fabric-based design does not affect the regularity of the layout patterns since all the patterns are still there, though unused. Early white-space allocation can help to achieve faster design convergence by predicting and relieving congestion before routing. For a fabric with fixed capacity, both the logic and the available routing resources affect whether a design can be mapped to the fabric. In this case, a fast and tight routing-resource-aware white-space allocation scheme is essential for quick estimation of design mappability and design convergence. Moreover, the routing-resource-aware design methodology can help a fabric vendor to explore quickly the trade-off between the number of logic blocks and the number of routing tracks. When trading off routing resources, a vendor can balance the die cost by deciding whether to put more tracks on one layer or to use more metal layers. Consequently, a series of well-tailored fabric chips with different routing capabilities, not only logic capacities, can be provided such that designers can choose the best match between a design and a fabric. Given a good estimation on routing efficiency, this area estimator can be very useful for all the fabrics with fixed routing structures, such as FPGAs and via-programmable fabrics, as well as fabrics with metal-programmable masks.

Most traditional white-space problems in literature come from fixed-die cell placement. In a fixed-die cell placement the amount of white-space is given, and the problem is how to properly distribute it. In [5], white-space is evenly distributed. In [6]–[8], white-space is distributed to congested regions based on routing estimation. In [9], a congestion-minimizing framework allocates white-space based on Rent's rule with no explicit congestion prediction. With the guide of Rent's rule, efficient clustering and placement techniques were developed to reduce congestion in FPGA-based designs [10], [11]. All of these approaches try to reduce congestion using the given white-space but they cannot completely eliminate the routing congestion. Our white-space allocation problem is different because we want to quickly determine how much white-space is needed and where it should be put to obtain a routable design.

In this paper, we first study the via-configurable routing architectures. We propose a new basic routing structure which provides larger throughput than do the routing architectures in [1] and [4]. We show also how to construct a single-via-mask fabric and discuss the penalty it might impose on the fabric's throughput. We then propose a fast white-space allocation scheme to determine how much white-space we need to add, and where to add it, to completely eliminate, not just reduce, the routing congestion. In our design flow, a fast global routing is performed to obtain an accurate congestion estimation. We allocate white-space in-place by calculating routing demand-supply relations. We use our efficient incremental placement and global routing algorithms to move cells away from congested regions. Finally, we perform detail routing to find out whether a

0-7803-9254-X/05/$20.00 ©2005 IEEE.

AMD EX1021
AMD v. Realtek - Page 449

design is indeed routable. These steps complete the flow and validate our approach.

The paper is organized as follows. In Section II we introduce our new base routing structure. In Section III we show the construction of a single-via-mask fabric. We introduce the test fabric architecture in Section IV and describe our routability-driven design flow in Section V. We report experimental results in Section VI. Section VII summarizes the contributions made by our research.

II. Base Routing Structures

In this paper, we assume that metal-1 (M1) and metal-2 (M2) layers in a fabric are used for implementing cell functions. An input or an output of a gate is related to an M2 segment which can be accessed through vias to a metal-3 (M3) segment. We focus on the routing structures above the M2 layer.

In a via-configurable routing architecture, there is a potential via connection whenever segments from two adjacent metal layers intersect. As a result, switching a signal from vertical to horizontal direction, or vice versa, can be easily realized through a via connection between the two intersecting segments. To continue a signal in one direction in a fixed wiring scheme, some dedicated short segments in the adjacent layer, called *jumpers* (shown in Fig. 1), are commonly used [1].

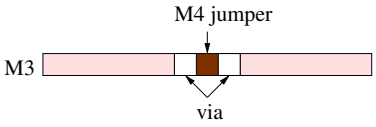
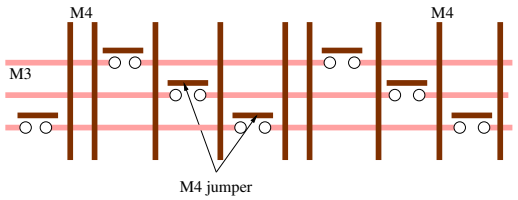


Fig. 1. A jumper connects two segments in one direction.

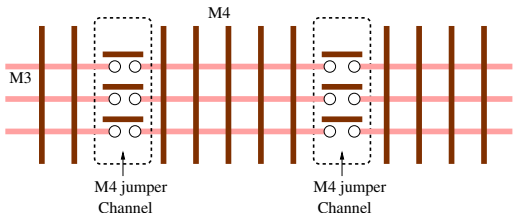
When planning wire segments, we have the choice to stagger or align the segments. Due to the use of short jumpers in a via-configurable routing structure, closely staggered wires are not favored. Figure 2 shows an example. The jumpers in a staggered wire segment scheme (Figure 2(a)) are distributed. As a result, the jumpers become the obstacles for vertical wires since they are in the same layer. Therefore, we adopt an aligned wire segment scheme as shown in Figure 2(b) where jumpers are aligned and form a dedicated jumper channel.

Aligned wire segments naturally form crossbars (as shown in Figure 3(a)) which provide the highest switching flexibility [1], [4]. In a crossbar structure, any horizontal segment can connect to any vertical segment, and vice versa. In a fabric, the width and height of a crossbar are usually determined by the size of a base block. A crossbar on top of a block provides pin access to the cells in the block, relays a signal in the same direction, or switches a signal to the other direction. In fixed wiring structures, there might be some dangling wires. Figure 3(b) shows an example. The signal only uses partial segments a and c. Partial segments b and d are redundant, called dangling wires. Dangling wires incur extra capacitance, degrade performance, and increase power consumption of a circuit. Using finer granularity of wire segments could reduce the impact of dangling wires.

Once the size of a base crossbar has been decided, the design of a routing structure in a fabric is reduced to determining the interconnections between two crossbars. In [1], two crossbars connect to each other using parallel jumpers. This is the *direct-jumper crossbar* structure which we depict in Fig. 4(a). Two types of wire segments are available in the crossbar [4], as shown in Fig. 4(b). The shorter segment is used only to switch a signal in the local crossbar or to access a local pin. The longer segment connects to a segment in

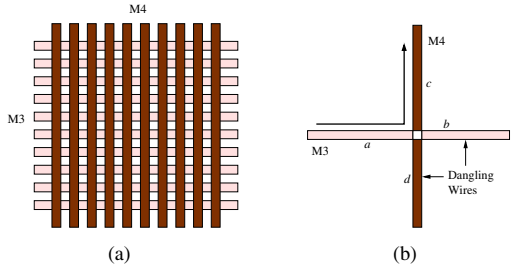


(a) Staggered wire segments



(b) Aligned wire segments

Fig. 2. Staggered versus aligned wire segments with jumpers.



(a) (b)

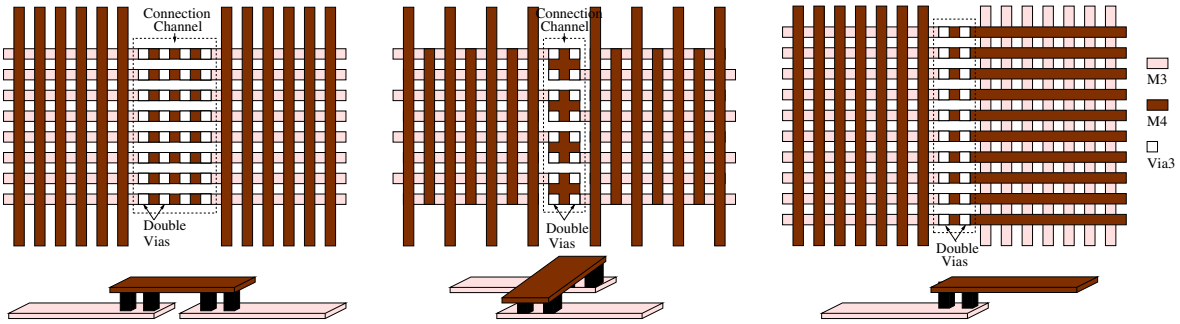
Fig. 3. A crossbar (a) and dangling wires (b).

an adjacent crossbar by a perpendicular jumper. We call this scheme a *jog-jumper crossbar* structure. Using a perpendicular jumper helps to pack two crossbars one wire-pitch closer, which increases the capacity of a crossbar by 2. From the figures, one can see that the jumpers occupy several tracks and reduce the number of available routing segments. Suppose that the width of a crossbar is n (in terms of wire pitches) and double vias are used for a connection. The number of available routing segments is $n-4$ for the direct-jumper, and $n-2$ for the jog-jumper crossbar. However, the number of segments which can connect to adjacent crossbars is only $(n-2)/2$ for the jog-jumper crossbar because short segments can only be used locally.

In both direct-jumper and jog-jumper structures, all the wire segments in one metal layer are oriented in the same direction. As a result, a jumper from another layer is necessary to connect two wires in the same layer. In this paper, we propose a new crossbar structure, a *crossover crossbar*, shown in Fig. 4(c). In the crossover crossbar structure, a base crossbar is rotated by 90 degrees with respect to its neighbor. In this way, two crossbars can be stacked at the ends. The crossover portion provides potential via connections since two wire segments leading in the same direction are now residing in two different metal layers. A route will alternate between the two layers when it passes through the continuous crossbars. Compared to the jumper crossbars, the crossover crossbar provides $n-2$ segments, and all of the segments can connect to adjacent crossbars.

The advantages of the crossover crossbar structure are summarized as follows.

AMD EX1021
AMD v. Realtek - Page 450



(a) Direct-jumper (b) Jog-jumper (c) Crossover

Fig. 4. Different crossbar structures

- *Higher capacity*. The crossover crossbar structure has $n-2$ connecting segments compared to $n-4$ in the direct-jumper crossbar and $(n-2)/2$ in the jog-jumper crossbar.
- *Fewer vias and smaller via resistance*. The crossover connection needs only two vias, whereas both jumper connections need four vias (assuming that double vias are used for each connecting point to improve yield). As a result, the via resistance in the crossover connection is only $R/2$, but it is R for both jumper connections where R is the resistance of a single via.
- *Flexible connection to the upper layers*. In both jumper crossbar structures, a net route using an M3 segment needs a specific M4 segment to reach an M5 segment if we want to route it in the same direction using M5. However, since in the crossover crossbar structure, half of the net's route uses M4, a direct connection to an M5 segment is possible without wasting an M4 segment.

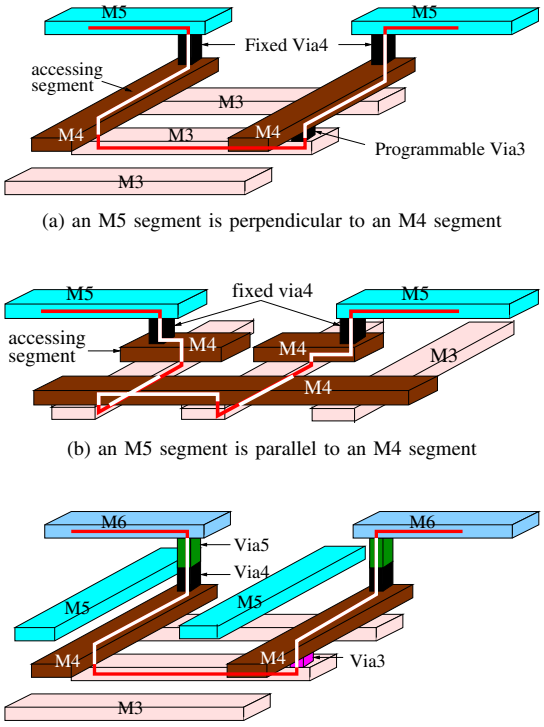
The upper layer M5 and M6 can use the same crossbar schemes. In general, the segments in M5 and M6 layer have larger granularity and are used to route long nets.

III. Single-Routing-Via-Mask Fabrics

In the routing structures discussed above, all the metal masks are fixed, and all the via masks are programmable. One way to reduce cost further is to use fewer via masks. In this section we shall describe how to construct a single-routing-via-mask fabric.

Suppose we use six metal layers for a fabric, let via3 (between M3 and M4) be programmable and other via masks be fixed. To make all the wire segments usable for routing, all the wires below M3 must be brought to M3, and the wires above M4 must be brought to M4 by fixed connections (fixed via4, via5 and wires) to access the potential vias between M3 and M4 segments.

We attach two dedicated M4 segments, called *accessing segments*, to both ends of each M5/M6 segment. Fig. 5(a) shows an example. The M5 segment is connected to the accessing M4 segment by fixed vias (via4). Because every accessing M4 segment has several potential connections to the M3 segments by programmable vias (via3), the M5 segment can connect to the M3 segments. Going up through one more via (from M3 to above M4), the M5 segment can also connect to other M4 segments. Moreover, through the accessing M4 segment of another M5/M6 segment, a signal can go back to the upper layer again. The red line in Fig. 5(a) shows a path of a signal, from M5 down to M4, M3, and then up to M4 and M5 layer. When an accessing M4 segment is parallel to an M5 segment (see Fig. 5(b)), a route may take a longer detour. Similarly, an M6 segment is connected to the



(a) an M5 segment is perpendicular to an M4 segment

(b) an M5 segment is parallel to an M4 segment

(c) an M6 segment is perpendicular to an M4 segment

Fig. 5. M5/M6 segment connection in a single-via-mask fabric. Bold lines represent the potential signal path.

accessing M4 segment by stacked vias (via4 and via5) to access the potential via sites between the M3 and M4 (shown in Fig. 5(c)). In this way, a route can make use of fixed M5/M6 wire segments but will need to come down to M3/M4 each time it changes to another wire segment.

The penalty of a single-routing-via-mask fabric can be summarized below.

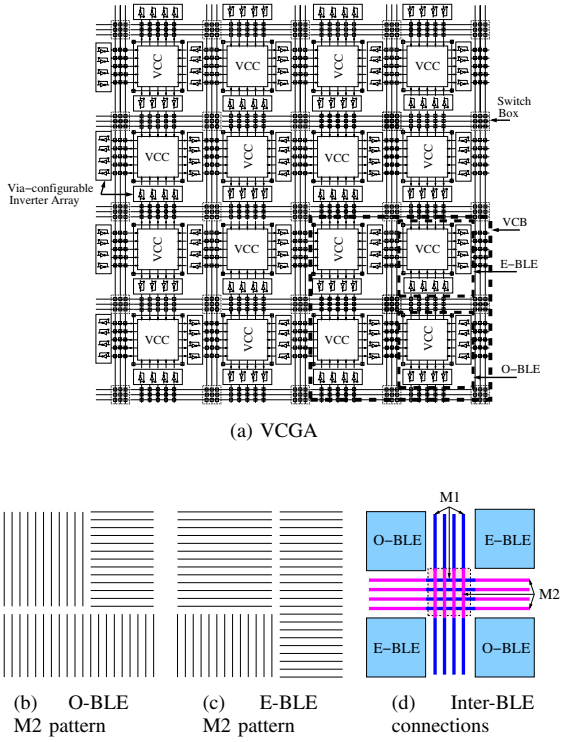
- Two dedicated accessing M4 segments have to be attached to each M5/M6 segment at two ends. As a result, the M4 segments available for local routing are reduced, which could worsen routability.
- An M3/M4 segment can only access an M5/M6 segment at

AMD EX1021
AMD v. Realtek - Page 451

its terminal (through the accessing segment). However, it can connect to a M5/M6 segment in the middle by programmable vias. This causes some performance penalty but significantly improves routability.

IV. Fabric architecture

The various routing structures are tested on a via-configurable gate array (VCGA) [12] shown in Fig. 6(a). A VCGA consists of an array of via-configurable blocks (VCBs). Each VCB contains four basic logic elements (BLEs). A BLE contains one via-configurable cell (VCC) and two via-configurable inverter arrays, which can implement both combinational and sequential functions customized by a via1 mask. There are two types of BLEs, odd-BLE (O-BLE) and even-BLE (E-BLE), which have different orientations. Fig. 6(b) and (c) show the M2 patterns for an E-BLE and O-BLE, respectively. All the inputs and outputs of a BLE are provided through the M2 segments. To reduce the routing stress on the top metal layers, the VCGA provides also a few inter-BLE M1/M2 wire segments for short connections between neighboring BLEs as shown in Fig. 6(d). A via-configurable switch box is used to switch a signal between horizontal M2 segments and vertical M1 segments.



(a) VCGA

(b) O-BLE M2 pattern (c) E-BLE M2 pattern (d) Inter-BLE connections

Fig. 6. Via-configurable gate array

V. Fast design mappability estimation and validation

Although the fixed routing structures can lead to much better manufacturability and significantly reduce mask cost, they might cause serious routability problems due to their inflexibility. When we map a design to a fabric with fixed routing structures, we want to know quickly whether the mapping is feasible. This not only depends on whether the fabric contains enough logic for the design, but also depends on whether the fabric can provide enough routing resources. The mappability estimation should consider the required white-space to accommodate the routing needs.

In this section we describe our routability-driven design flow. Fig. 7 outlines our flow. Given a circuit netlist, we first pack it into circuit blocks such that each of them can be mapped to a master block. The main objective of the packing step is to achieve high design density and low inter-block connections. We adopt the algorithm in [13]. We then place circuit blocks. After we have an initial cell placement, we do a fast global routing by constructing Steiner trees for each net. We calculate the routing usage over each block and obtain an accurate estimation of congestion over each region. By combining the routing demands over each row and column with the routing supply provided by the given routing architecture, we determine whether and where extra master blocks must be inserted to provide enough white-space to eliminate congestion. After allocating white-space, we can immediately know whether the design is mappable at this stage by comparing the size of the fabric and the size of the design with required white-space. To validate our white-space allocation algorithm and complete our design flow, we perform an incremental cell movement to reduce congestion. With a routing tree already constructed, we can have an accurate evaluation on congestion change due to the cell movement. The evaluation is also very fast since it involves only a few partial routing tree segments which connect to the moved cell. The routing trees are incrementally updated when we move a cell. At the end, we rip-up nets passing through congested regions and re-route them using a more effective, yet more expensive method, e.g., maze search. The global routing results are then fed into a detailed router to complete the design. In the following section, we shall describe in detail the congestion-aware white-space allocation and incremental cell movement.

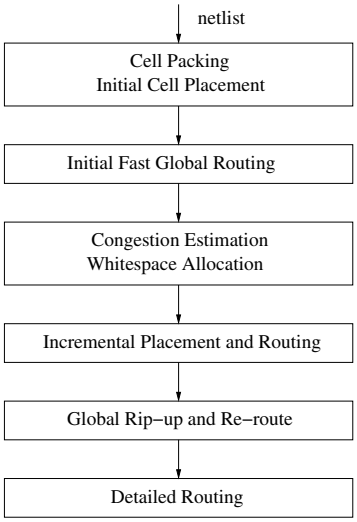


Fig. 7. Routability-driven design flow.

A. Congestion-Aware White-space Allocation

After initial global routing, we have an accurate estimation of routing usage over each region. When the routing demands exceed the routing supply, congestion occurs. A few congested spots could be relieved by re-routing some nets if there are extra routing resources nearby. When there are excessive overflows, e.g., very limited routing resources in some routing architectures, or highly complicated interconnects in some designs, we have to allocate some white-space, i.e., add more master blocks, to accommodate the routing needs.

AMD EX1021
AMD v. Realtek - Page 452

Conceptually, we need to answer two questions for white-space allocation: how much white-space should be allocated, and where they should be put. The first question is hard to answer without considering the capability of white-space reducing congestion. For the second question, ideally, white-space should go to congested regions. However, the manner of implementation is not trivial. If we force the white-space at congested regions and move away cells which were originally there, it's hard to guarantee that no new congestion will occur, particularly when the stresses of routing in regions around congested spots are also heavy. This method is equivalent to adding extra master blocks at the boundary of the original placement and then moving them to the congested regions as shown in Fig. 8(a).

To overcome the above difficulties, we propose a fast row/column based in-place white-space allocation scheme. We directly insert a row (column) of master blocks into the congested region and shift the other rows (columns) (Fig. 8(b)). As a result, the original cell placement is shifted. The existing routing is shifted or expanded across the allocated row/columns. Therefore, the original congestion estimation for each row/column does not change. The inserted empty blocks provide a space near the cells to be moved from congested regions. In this way, the white-space is quickly allotted to congested regions without generating new congestion. Moreover, the insertion of new rows (columns) is based on in-place routing demand-supply estimation. Hence the amount of white-space needed can be accurately determined.

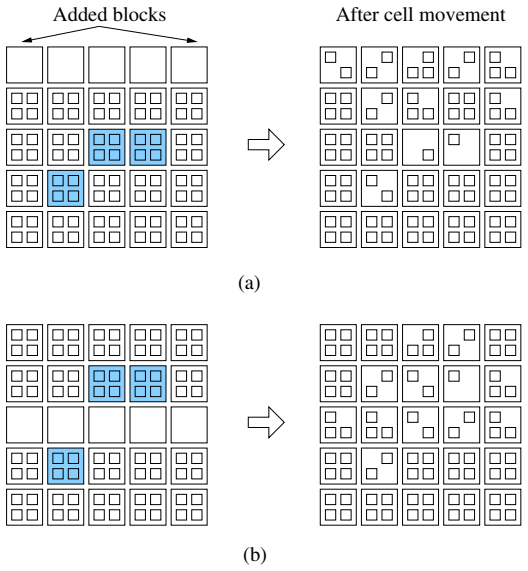


Fig. 8. White-space allocation and cell movement to eliminate congestion. Filled blocks represent congested blocks.

To determine where to insert a row, we scan rows from bottom to top, accumulating the routing usage over rows. An empty row is inserted when the following demand-supply condition is satisfied.

$$\sum_{r=i}^{j} U(r) > \alpha((j-i+1)+1)C_h \quad (1)$$

$$U(r) = \sum_{i=1}^{W} u_h(i,r)$$

$$C_h = Wc_h \quad (2)$$

where $u_h(r,i)$ is the number of used horizontal track segments at block $B(i,r)$, c_h is the number of total horizontal track segments at a block, and W is the number of blocks in a row. In other words, $U(r)$ is the total routing usage of r-th row, and C_h is the total routing supply of a row. The scan goes from row i to row j. α is the routing efficiency factor. Because not every net can be routed completely straight, and extra jogs consume routing resources, α is set around 0.7 for our experiments. The left-hand side of the inequality represents the total routing demands over the scanned rows, and the right-hand side represents the total routing supply of the scanned rows plus the empty row to be inserted. When the above condition is satisfied, we insert an empty row after the most congested row among the recently scanned rows. We then move the scanning point to the next unscanned row and continue the above procedure.

In practice, we observe that a large span of scanned rows may not be effective in reducing congestion. The reason is that cell movement is difficult due to other constraints if available empty sites are far distant from congested spots. Therefore, we choose a local scan window of width Δ. Only when the condition (1) is satisfied over a local scan window, we insert an empty row. If not, we shift the scan window one row forward. The width of a scan window is a function of circuit width W, i.e., $\Delta = \beta W$, where β is a constant around 0.1 in our experiments. Algorithm 1 shows the pseudo-code of our white-space allocation scheme.

Algorithm 1 Empty Row Allocation

```
1: calculate routing usage U(r) for each row r;
2: Δ = βW;
3: sum_usage := 0;
4: nScannedRow := 0;
5: {H is the circuit height}
6: for r := 1 to H do
7:    sum_usage := sum_usage + U(r);
8:    nScannedRow++;
9:    if sum_usage > α×(nScannedRow+1)×C_r then
10:      H++;
11:      idx := the most congested row between row (r-
         nScannedRow) and row r;
12:      insert a row of blocks at row (idx+1);
13:      shift cells in row (idx+1) to row (H) up by one row;
14:      sum_usage := 0;
15:      nScannedRow := 0;
16:   else if nScannedRow > Δ then
17:      sum_usage := 0;
18:      nScannedRow := 0;
19:   end if
20: end for
```

Empty columns are inserted in a similar way. By inserting rows/columns of blocks based on the initial congestion estimation, we hope to provide enough tight routing resources for a circuit so that no design iterations are required to achieve routable designs.

The only problem of this white-space allocation scheme is that it might allocate more than enough white-space. But from experiments we see it provides a tight estimation. The simplicity and great efficiency make it very useful for fast design mappability estimation.

B. Increment Cell Placement and Global Routing

After congestion estimation, and possibly white-space allocation, we have some empty sites around the congested regions. If no extra blocks are allocated, it usually means the routing congestion can be eliminated by local cell swapping. In either case, we incrementally

AMD EX1021
AMD v. Realtek - Page 453

move away a cell from a block which has hard constraint violations or congestion.

In our flow, the incremental cell movement algorithm starts with evaluating the costs for each block and inserting them in a priority queue where the block with the largest cost is at the top. Each time we choose a cell from the block with the highest cost it implies that moving away the cell can result in the largest cost reduction for that block. The movement stops when no gain can be achieved. We adopt the move approach from [14], where incremental placement is used for layout-driven optimization. We try the following cell movements:

- Move to a neighboring block.
- Move to a fanin block of the cell.
- Move to a fanout block of the cell.
- Move to a block where its topological sibling cell lies in.

We evaluate all the possible moves and each time choose the best one.

Unlike [14] which uses a probabilistic wire-length model, we utilize the existing global routes and hence have a deterministic congestion measure. When a cell is moved to another block, the pins belonging to the cell move accordingly. This invalidates the original routing trees. However, when a cell is moved only one block, e.g. moving to one of its four nearest neighbors, the routing trees are only locally changed and can be updated quickly. Fig. 9 shows an example where pin A is moved to a right block, and edge a moves to a' and b is discarded. In this example, we need only to examine the routing grids through which the edges a, b and a' pass.

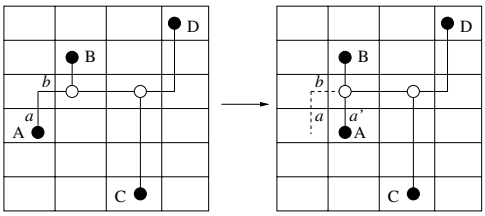


Fig. 9. Incremental Steiner-tree modification due to cell movement where solid dots represent pins and hollow dots represent Steiner points.

When a cell is moved to another block at a distance larger than one, we simply re-construct the Steiner trees for nets whose pins are moved. In this way, we always maintain a valid global routing and accurate congestion estimations.

Due to the nature of Steiner-tree construction used in initial global routing, routing congestion might not be completely eliminated with only the above incremental cell movement. Rip-up and re-route are usually required as a post-processing step. In our flow, we rip-up nets passing through the congested regions and re-route them using the *PathFinder* algorithm proposed in [15]. Because we have already allocated enough routing space, it is generally easy to fully eliminate all the congested areas. Finally, the global routing results are fed into a detailed router to complete the design.

VI. Experiments and Discussions

A. Via-Configurable Routing Structures

We performed a set of experiments to examine particular routing structures and trade-offs. The M3/M4 crossbar on top of a BLE in our base architecture can accommodate 20×20 wire tracks, which along with the four M1/M2 inter-BLE wires results in a base unit of 24×24 wire pitches. To evaluate particular routing architectures without depending on white-space allocation algorithm, for each circuit we use a binary search to determine the minimum number of extra tracks between blocks which need to be added to achieve routability. It is the same approach as used in most of the FPGA research based on VPR [16]. All the experimental results are based on $0.18mum$ technology parameters. The resistance of a single via is 3 Ω, and the wire capacitance is $180fF/mm$.

We first investigated the capabilities of the three-base crossbar structures discussed in Section II. The experiments were run on a set of large MCNC benchmarks. We used only four metal layers. Table I lists the results. The column *#BLEs* gives the circuit sizes in terms of the number of packed BLEs. The columns under *Extra routing tracks* show the number of added tracks per row (column) required to successfully route a design using each crossbar structure. From the table, we can see that the crossover crossbar structure requires fewer routing resources than the jog-jumper (*J-jumper* in the table) and the direct-jumper (*D-jumper* in the table) crossbar structures. Consequently, due to the smaller total circuit area, the crossover crossbar structure also yields the best results in terms of the total wire-length and critical path delay. On average, a circuit implemented using the crossover crossbar structure has 29% and 10% less total wire-length compared to jog-jumper and direct-jumper crossbar structures, respectively. It is 21% faster than the jog-jumper implementation and 8% faster than the direct-jumper crossbar. The experiment demonstrates the larger throughput of the crossover crossbar structure.

We then examined the multi-layer routing structures with differing M5/M6 segment lengths. Both M3/M4 and M5/M6 use crossover crossbar structures. Fig. 10 shows the results for the circuit `alu4`. From the figure, we observe that the total wire length increases when the M5/M6 segment lengths increase. This is because, given the same routing area, there are fewer routing resources for longer M5/M6 segments. As a result some routes will use only parts of the M5/M6 segments, which increases the route wire-length. We observe that a circuit is not routable when the M5/M6 segment length is larger than 6, unless we add extra routing area. When the M5/M6 segment length increases, the number of used vias first decreases rapidly and then slowly increases. The decrease is predictable since routing a long net using shorter segments will take more segments than using the longer segments and thus will lead to more vias. But when the M5/M6 segments are too long, the scarcity of available segments causes more detours for a route and hence increases the number of vias. The critical timing increases with a longer M5/M6 segment length. We observed that the delay penalty caused by extra vias with length-1 segment is less than 3% of the critical path delay, which is much smaller than the delay increase (around 10$\sim$15%) due to larger total wire length when having longer M5/M6 segment lengths. This example shows that when routing resources are insufficient and M5/M6 segments are required to provide local routing, longer M5/M6 segments would cause a performance problem. For some cases (for example, `bigkey`) where M3/M4 segments are sufficient for local routing, longer M5/M6 segments can result in better performance due to reduced vias.

Next we examine the single-via-mask routing structure described in Section III (There are actually two via masks in a VCGA since the via1 mask is required for cell function implementation). Due to the fixed M5/M6 segment connections, a single-via-mask fabric is less flexible for routing. Table II shows the results. In this experiment, we chose length-4 M5/M6 segments and length-12 M4 accessing segments for a single-via-mask fabric. Column *area incr* gives the area increase (extra area required for a routable design) and *delay incr* gives the delay increase compared to the multi-via-mask implementation with length-1 M5/M6 segments. We observed that

AMD EX1021
AMD v. Realtek - Page 454

TABLE I

TESTS ON MCNC BENCHMARKS FOR THREE CROSSBAR STRUCTURES WITH FOUR METAL LAYERS.

Circuit	#BLEs	Extra routing tracks			Total wire-length			Critical path delay (ns)		
		J-jumper	D-jumper	Crossover	J-jumper	D-jumper	Crossover	J-jumper	D-jumper	Crossover
alu4	36 × 35	13	8	5	2263485	1925622	1691041	6.5	5.9	5.4
apex2	40 × 39	13	8	4	2749598	2327341	1969921	7.4	6.5	5.8
bigkey	32 × 31	9	4	2	1419207	1155193	1065322	5.0	4.4	4.0
des	33 × 32	8	4	2	1225206	11058739	953461	7.1	6.6	6.3
elliptic	48 × 47	10	4	2	3352078	2727386	2474522	21.8	17.7	17.7
ex1010	45 × 42	18	10	7	4620563	3645988	3239009	10.8	9.1	8.3
frisc	46 × 44	8	3	1	2505372	1988937	1829493	19.3	17.3	16.3
misex3	35 × 34	13	8	5	2117885	1797122	1576889	6.7	5.9	5.5
pdc	37 × 36	15	9	6	2641750	2181373	1933786	12.2	10.7	9.9
seq	39 × 38	14	6	5	2701991	2149068	1959774	7.1	6.1	5.6

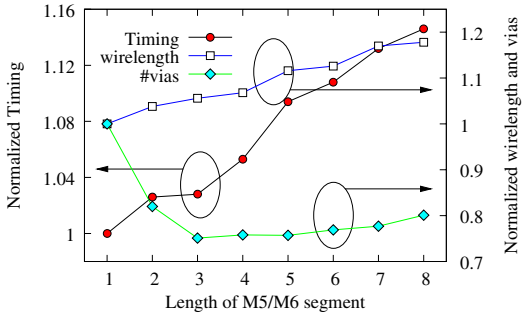


Fig. 10. The impact of different M5/M6 segment lengths for `alu4`.

TABLE II

SINGLE-VIA-MASK VS. MULTI-VIA-MASK FABRICS.

Circuit	area incr (%)	delay incr (%)	Used M5/M6 WL(%)	
			length-1	length-4
alu4	17.4	14.3	39.2	46.0
apex2	17.4	16.0	38.0	43.9
bigkey	0	2.5	44.2	49.0
des	0	5.3	47.2	51.1
elliptic	0	3.9	46.6	50.8
ex1010	46.0	25.4	42.9	51.4
frisc	0	2.0	39.7	45.0
misex3	17.4	15.2	39.5	46.6
pdc	26.6	16.7	40.2	46.1
seq	17.4	15.2	41.4	47.4

some designs with larger interconnect complexity are not routable without adding extra routing area, although all of them are routable using multi-via masks. Columns *Used M5/M6 WL* indicate the reason. Compared to a multi-via-mask with length-1 M5/M6 segments, a circuit implemented by a single-via-mask fabric uses on average only 42% of the M5/M6 wires that it would use in a multi-via-mask implementation (Column *length-1*). Even compared to a multi-via-mask fabric with length-4 M5/M6 segments (the same as in the single-via-mask fabric), the M5/M6 segments still account for on average 48% of those used in the multi-via-mask implementations (Column *length-4*). This is because in the single-via mask, a segment can connect only to two endpoints of an M5/M6 segment, but it can connect to some middle points of an M5/M6 segment in the multi-via-mask fabric. If extra routing area is needed, the delay will increase significantly.

Fixed wiring in a fabric can cause numerous dangling wires compared to the customized metal masks, particularly when a signal switches from one direction to another in a crossbar. By cutting dangling wires off to mimic the structured routing in [4], we have observed that about 15∼25% of delay improvement can be achieved compared to the implementation with fixed metal masks in the multi-via-mask fabric.

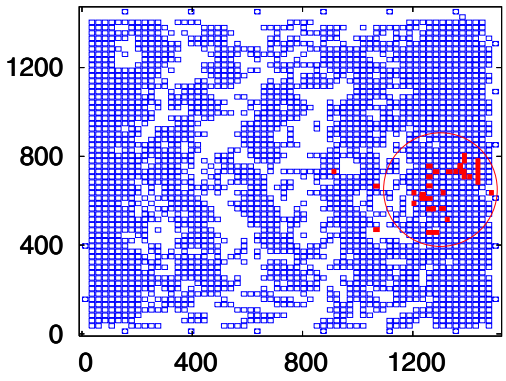
B. Routability-Driven Design Flow

We also experimented with our routability-driven design flow. In these experiments we assume that only four metal layers are used for the via-configurable fabric. With six metal layers there would be no routability issue for the available benchmark circuits which, compared to the modern designs, are relatively small.

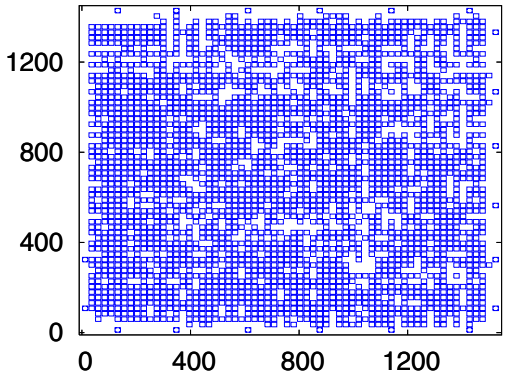
We performed the experiments using two flows. The first flow was the same as the one used in the previous experiments, i.e., adding more inter-block tracks instead of allocating white-space to solve the routability problem. For fixed routing tracks, the second flow described in Section V added white-space to solve the routability problem. To compare our post-placement white space allocation scheme with the in-placement white-space allocation scheme, we also performed the experiments using Dragon from UCLA [7] which is the only publicly available placer capable of allocating white-space. Because Dragon cannot decide how much white-space is required, we determine, via a binary search method, the minimum white-space required to generate a congestion-free design.

We did experiments on the via-configurable gate array with crossover crossbar routing structure. Table III shows the results. The columns under *Area increase* give the percentage of area increase over the original densely packed design for various methods. Column *Adding tracks* lists the results of the first flow (varying routing tracks). Columns *WSA by Dragon* and *WSA by ours* give the results from the second flow (allocating white-space with fixed tracks) by Dragon and our algorithm. The columns labeled *diff* show the area differences between our method, the varying-track method, and Dragon. From the table, we observe that on average, our white-space allocation scheme requires 17% more area than the varying-track method, but 3% less than Dragon. Unlike in Dragon, we need several iterations to determine the amount of white-space. However, experimentally we observed that all the circuits were successfully routed in one shot using our white-space allocation and incremental cell movement algorithm. In some cases Dragon's results are worse than ours because of inaccurate congestion estimation. Fig. 11 shows as an example the circuit `ex1010`. From the figure, we can see that Dragon allocated too much white-space in the center of the placement area, but the congestion actually occurred close to the boundary. In contrast, our white-space allocation algorithm allocated white-space properly due to the accurate congestion estimation based on the actual global routing. Considering that the first flow is unrealistic for a fabric-based design, our white-space allocation provides a very tight design

AMD EX1021
AMD v. Realtek - Page 455



(a)



(b)

Fig. 11. Final placement after white-space allocation and incremental cell movement for `ex1010` using different white-space allocation schemes: Dragon (a) and ours (b). There is no congestion using our white-space allocation algorithm, but some remaining congestion (shown as the bold lines in the circle (a)) occurs if using Dragon.

implementation in terms of final circuit area.

TABLE III

COMPARISON OF DIFFERENT CONGESTION ELIMINATION SCHEMES

Circuit	Area increase (%)			diff (%)	
	adding tracks	WSA by Dragon	WSA by ours	with tracks	with Dragon
alu4	46	64	61	+15	-3
apex2	36	77	60	+24	-17
bigkey	0	0	10	+10	+10
des	17	5	12	-5	+7
elliptic	17	32	47	+30	+15
ex1010	67	135	103	+36	-32
frisc	9	26	39	+30	+13
misex3	46	71	55	+9	-16
pdc	56	74	62	+6	-12
seq	46	64	59	+13	-5
AVG				+17	-3

VII. Conclusions

In this paper we have examined various aspects of via configurable routing structures. We have proposed a new crossover crossbar structure which achieves much higher throughput than the jog-jumper and the direct-jumper crossbar structures. As a result, the area and timing of a circuit implemented using the crossover crossbar structures can be greatly improved. We have shown how to construct a single-via-mask fabric. Compared to the multi-via-mask fabric, due to routing inflexibility, the single-via-mask fabric can incur up to 25% delay penalty and 46% area penalty. We further proposed a routability-driven design flow for fabric with fixed routing resources. The flow features a fast yet efficient white space allocation scheme and a congestion-aware incremental cell movement method for reducing congestion. Experimental results show that our flow can achieve results comparable to those of the variable-tracks-flow or in-placement white-space allocation, but it is much more practical and efficient in the context of fabric-based designs. Our flow could be used by a fabric vendor to quickly evaluate different architectures.

VIII. Acknowledgments

This work was supported in part by NSF grant #CCR 0098069, and in part by the California MICRO program through IBM. The authors gratefully acknowledge equipment grant from Intel.

References

[1] C. Patel, A. Cozzie, H. Schmit, and L. Pileggi, "An architectural exploration of via patterned gate arrays," in *Proc. of ISPD*, 2003, pp. 184–189.

[2] L. Pileggi, H. Schmit, A. J. Strojwas, P. Gopalakrishnan, V. Kheterpal, A. Koorapaty, C. Patel, V. Rovner, and K. Y. Tong, "Exploring regular fabrics to optimize the performance-cost trade-off," in *Proc. of DAC*, 2003, pp. 782–787.

[3] B. Hu, H. Jiang, Q. Liu, and M. Marek-Sadowska, "Synthesis and placement flow for gain-based programmable regular fabrics," in *Proc. of ISPD*, 2003, pp. 197–203.

[4] V. Kheterpal, A. J. Strojwas, and L. Pileggi, "Routing architecture exploration for regular fabrics," in *Proc. of DAC*, 2004, pp. 204–207.

[5] A. E. Caldwell, A. B. Kahng, and I. L. Markov, "Can recursive bisection produce routable placements?" in *Proc. of DAC*, 2000, pp. 260–263.

[6] U. Brenner and A. Rohe, "An effective congestion driven placement framework," *IEEE Trans. on CAD*, vol. 22, no. 4, pp. 387–394, April 2003.

[7] X. Yang, B.-K. Choi, and M. Sarrafzadeh, "Routability-driven white space allocation for fixed-die standard-cell placement," *IEEE Trans. on CAD*, vol. 22, no. 4, pp. 410–419, April 2003.

[8] C. Li, M. Xie, C.-K. Koh, J. Cong, and P. H. Madden, "Routability-driven placement and white space allocation," in *Proc. of ICCAD*, 2004, pp. 394–401.

[9] B. Hu and M. Marek-Sadowska, "Congestion minimization during placement without estimation," in *Proc. of ICCAD*, 2002, pp. 739–745.

[10] A. Singh, G. Parthasarathy, and M. Marek-Sadowska, "Interconnect resource-aware placement for hierarchical FPGAs," in *Proc. of ICCAD*, 2001, pp. 132–136.

[11] A. Singh and M. Marek-Sadowska, "Efficient circuit clustering for area and power reduction in FPGAs," in *Proc. of FPGAs*, 2002, pp. 59–66.

[12] Y. Ran and M. Marek-Sadowska, "The magic of a via-configurable regular fabric," in *Proc. of ICCD*, 2004, pp. 338–343.

[13] B. Hu and M. Marek-Sadowska, "Wire length prediction based clustering and its application in placement," in *Proc. of DAC*, 2003, pp. 800–805.

[14] D. P. Singh and S. D. Brown, "Incremental placement for layout-driven optimizations on FPGAs," in *Proc. of ICCAD*, 2002, pp. 752–759.

[15] C. Ebeling, L. McMurchie, S. A. Hauck, and S. Burns, "Placement and routing tools for the Triptych FPGA," *IEEE Trans. on VLSI Systems*, vol. 3, no. 4, pp. 473–482, December 1995.

[16] V. Betz and J. Rose, "VPR: A new packing, placement and routing tool for FPGA research," in *Proc. of FPGAs*, 1997, pp. 213–222.

AMD EX1021
AMD v. Realtek - Page 456

EXHIBIT 9A

MARC Bibliographic Record

```
LEADER    01575cam-a2200421zu-4500
001       991021971127102122
005       20210806235836.0
008       160829s2005 xx o eng d
020           $a1-5386-0197-4
035           $a(CKB)1000000000022706
035           $a(SSID)ssj0000393852
035           $a(PQKBManifestationID)12146529
035           $a(PQKBTitleCode)TC0000393852
035           $a(PQKBWorkID)10378562
035           $a(PQKB)10679731
035           $a(EXLCZ)991000000000022706
040           $aPQKB
041           $aeng
050   _0      $aTK7867.I3713 2005
082           $a621.3815
110           $aIEEE/ACM International Conference on Computer-Aided
              Design$eCorporate Author
111   2_      $aIEEE/ACM International Conference on Computer-Aided Design
```

```
245    10    $aICCAD-2005 : International Conference on Computer Aided
             Design, November 6-10, 2005, DoubleTree Hotel, San Jose, CA

264    31    $a[Place of publication not identified]$bIEEE$c2005

336          $atext$btxt

337          $acomputer$bc

338          $aonline resource$bcr

500          $aBibliographic Level Mode of Issuance: Monograph

546          $aEnglish

650    _0    $aElectronic circuit design$xData processing$vCongresses

650    _0    $aComputer-aided design$xTesting$vCongresses

650    _0    $aElectronic circuits$vCongresses

650    _7    $aElectrical & Computer Engineering$2HILCC

650    _7    $aElectrical Engineering$2HILCC

650    _7    $aEngineering & Applied Sciences$2HILCC

711          $aIEEE/ACM International Conference on Computer-Aided Design

710    2_    $aIEEE Circuits and Systems Society$eContent Provider

776          $z0-7803-9254-X

740    0_    $aComputer-Aided Design, 2005, ICCAD-2005, IEEE/ACM
             International Conference on.

906          $aPROCEEDING
```

MMS IDs

Document ID:

9911034467102121

Network Electronic IDs:

9911034467102121

Network Physical IDs:

mms_gb_ids:

991007108530202123

mms_lc_ids:

991016913436502125

mms_mad_ids:

991021971127102122

mms_ml_ids:

991012780349802124

mms_pl_ids:

99901546219002127

mms_plt_ids:

991013477658802128

mms_rf_ids:

991013794492702129

mms_st_ids:

991013936966302131

mms_ww_ids:

991015924504702133

AMD EX1021
AMD v. Realtek - Page 460

ATTACHMENT 9B

OCLC 62520971 **No holdings in XXX - 46 other holdings**

Books

						Rec Stat		**Entered**	20051209	**Replaced**	20220623141741.1
Type	a	**ELvl**	l	**Srce**	d	**Audn**		**Ctrl**		**Lang**	eng
BLvl	m	**Form**		**Conf**	1	**Biog**		**MRec**		**Ctry**	nyu
		Cont	b	**GPub**		**LitF**	0	**Indx**	1		
Desc	a	**Ills**	a	**Fest**	0	**DtSt**	s	**Dates**	2005,		

010 2006296183
040 NRC $b eng $c NRC $d BAKER $d IXA $d E9X $d DLC $d BTCTA $d C$$A $d OCLCF $d OCLCA $d OCLCQ $d OCL $d OCLCO $d OCLCA $d AAA $d OCLCO
019 1090663497
020 078039254X $q (Softbound ed.)
020 9780780392540 $q (Softbound ed.)
037 $b IEEE Xplore POP
050 00 TK7867 $b .I3713 2005
082 00 621.3815 $2 22
090 $b
111 2 IEEE/ACM International Conference on Computer-Aided Design $d (2005 : $c San Jose, Calif.)
245 10 ICCAD-2005 : $b International Conference on Computer Aided Design, November 6-10, 2005, DoubleTree Hotel, San Jose, CA / $c sponsored by IEEE Circuits & Systems Society [and others].
246 1 $i Title on page ii: $a Proceedings of the 2005 International Conference on Computer-Aided Design
246 30 International Conference on Computer Aided Design
246 14 IEEE/ACM digest of technical papers
246 18 IEEE/ACM International Conference on CAD-05
246 1 $i Also known as: $a ICCAD '05
260 Piscataway, N.J. : $b IEEE ; $a New York, N.Y. : $b ACM, $c ©2005.
300 xxx, 1082, A5 pages : $b illustrations ; $c 28 cm
336 text $b txt $2 rdacontent
337 unmediated $b n $2 rdamedia
338 volume $b nc $2 rdacarrier
500 "A conference for the CAD and design professional"--Cover.
500 "ACM SIGDA"--Cover.
500 "IEEE Catalog Number: 05CH37700, ACM Order Number 477053"--Page ii.
504 Includes bibliographical references and author index.
530 Also issued online with additional titles: Computer-Aided Design, 2005, ICCAD-2005, IEEE/ACM International Conference on; Proceedings of the 2005 IEEE/ACM International Conference on Computer-Aided Design.
505 0 Storage assignment during high-level synthesis for configurable architectures / Wenrui Gong / Gang Wang / Ryan Kastner -- Performance-driven read-after-write dependencies softening in high-level synthesis / Rafael Ruiz Sautua / Maria C. Molina / J. M. Mendias / R. Hermida -- An exact algorithm for the maximal sharing of partial terms in multiple constant multiplications / Paulo F. Flores / Jose C. Monterio / Eduardo C. Costa -- FPGA device and architecture evaluation considering process variations / Ho Yan P. Wong / Lerong Cheng / Yan Lin / Lei He -- Via-configurable routing architectures and fast design

mappability estimation for regular fabrics / Yajun Ran / Malgorzata Marek Sadowska -- SPIDER : simultaneous post-layout IR-drop and metal density enhancement with redundant fill / Kwok Shing Leung -- Computational geometry based placement migration / Tao Luo / Haoxing Ren / Charles J. Alpert / David Z. Pan -- An efficient and effective detailed placement algorithm / Min Pan / Natarajan Viswanathan / Chris Chu -- Post-placement rewiring and rebuffering by exhaustive search for functional symmetries / Kaihui Chang / Igor L. Markov / Valeria Bertacco -- Wirelength optimization by optimal block orientation / Xin Hao / Forrest Brewer -- Parametric test development for RF circuits targeting physical fault locations and using specification-based fault definitions / Erkan Acar / Sule Ozev -- Response shaper : a novel technique to enhance unknown tolerance for output response compaction / Mango Chao / Seongmoon Wang / Srimat T. Chakradhar / Kwang-Ting Cheng -- Test planning for the effective utilization of port-scalable testers for heterogeneous core-based SOCs / Anuja Sehgal / Krishnendu Chakrabarty -- Cocktail scan based on random access scan toward low power and high efficiency test / Shih Ping Lin / Chung Len Lee / Jwu E. Chen -- A statistical study of the effectiveness of BIST jitter measurement techniques / David A. Bordoley / Hieu Nguyen / Mani Soma -- The circuit design of the synergistic processor element of a cell processor / Osamu Takahashi / R. Cook / S. Cottier / S. H. Dhong / B. Flachs / K. Hirairi / A. Kawasumi / H. Murakami / N. Noro / H. Oh / S. Onishi / J. Pille / J. Silberman -- Adaptive designs for power and thermal optimization / Richard McGowen -- Digital RF processor (DRP) for cellular phones / Robert Bogdan Staszewski / Khurram Muhammed / Dirk Leipold -- A layout dependent full-chip copper electroplating (ECP) topography model / Jianjeng Luo / Qing Su / Charles Chiang / Jamil Kawa -- Interval-valued statistical modeling of oxide chemical-mechanical polishing / James Ma / Claire Fang / Rob A. Rutenbor / Xiaolin Xie / Duane S. Boning -- Fast and efficient phase conflict detection and correction in standard-cell layouts / Charles Chiang / Andrew A. Kahng / Subarna Sinha / Xi Xu -- IMF : interconnect-driven multilevel floorplanning for large-scale building-module designs / Tungchieh Chen / Yao-Wen Chang / Shyh-Chang Lin -- Robout mixed-size placement under tight white-space constraints / Jason Cong / Michail Romesis / Joseph Shinnerl -- Intrinsic shortest path length : a new, accurate a priori wirelenght estimator / Sherief Reda / Andrew B. Kahng -- Synthesis methodology for built-in at-speed testing / Yinghua Li / Alex Kondratyev / Robert K. Brayton -- Clustering for processing rate optimization / Chuan Lin / Hai Zhou / Jia Wang -- ConvexFit : an optimal minimum-error convex fitting and smoothing algorithm with application to gate-sizing / Sanghamitra C. Roy / Weijen Chen / Charlie Chen -- FinFETs for nanoscle CMOS digital integrated circuits / Tsu-Jae King -- Physic-based compact modeling for nonclassical CMOS / Vishal P. Trivedi / J. G. Fossum / L. Mathew / M. M. Chowdhury / W. Zhang / G. O. Workman / B. Y. Nguyen -- Double-gate SOI devices for low-power and high-performance applications / Kaushik Roy / Hamid Mahmoodi / Saibal Mukhopadhyay / Hari Ananthan / Aditya Bansal / Tamer Cakici -- Thermal simulation techniques for nano transistors / Jeremy Rowlette / Eric Pop / Sanjiv Sinha / Mathew Panzer / Kenneth E. Goodson -- An automated techniques for topology and route generation of application specific on-chip interconnection networks / Krishnan Srinivasan / Karam S. Chatha / Goran Konjevod -- Deadlock-free routing and component placement for irregular mesh-based networks-on-chip / Martin Schafer / Thomas Hollstein / Heiko Zimmer / Manfred Glesner -- Application-specific network-on-chip architecture customization via long-range link insertion / Umit Y. Ogras / Radum Marculescu -- NoCEE : energy marco-model extraction methodology for networks on chip routers / Jeremy Chan / Parameswaran -- Architecture and compilation for data bandwidth improvement in configurable embedded processor / Jason Cong / Guoling Han / Zhiru Zhang -- Code restructuring for improving cache performance of MPSoCs / Guilin Chen / Mahmut Kandemir -- Two-dimensional data locality : definition, abstraction, and application / Mahmut Kandemir -- Integrating loop and data optimizations for locality within a constraint network based framework / Guilin Chen / Ozcan Ozturk / Mahmut Kandemir / Ibrahim Kolcu -- System level verification of digital signal processing applications based on the polynomial abstraction technique / Tarvo Raudvere / Ashish Kumar Singh / Ingo Sander / Axel Jantsch -- Equivalence verification of polynomial datapaths with fixed-size bit-vectors

using finite ring algebra / Namrata Shekhar / Priyank Kalla / Florian Enescu / Sivaram Gopalakrishnan -- RTL SAT simplification by interval arithmetic reasoning / Ganapathy Parthasarathy / Madhu K. Iyer / Kwang T. Cheng / Forrest Brewer -- Runtime integrity checking for inter-object connections / Guilin Chen / Mahmut Kandemir -- Optimal post-placement voltage island generation under performance requirement / Huaizhi Wu / IMin Liu / Yusu Wang / Martin Wong -- Buffer insertion under process variations for delay minimization / Liang Deng / Martin Wong -- Efficient algorithms for buffer insertion in general circuits based on network flow / Ruiming Chen / Hai Zhou -- Trade-off between latch and flop for min-period sequential circuit designs with crosstalk / Chuan Lin / Hai Zhou -- Flip-flop insertion with shifted-phase clocks for FPGA power reduction / Hyeonmin Lim / Kyungsoo Lee / Youngjin Cho / Naehyuck Chang -- Acyclic modeling of combinational loops / Amit Gupta / Charley Selvidge -- Fast algorithms for IR drop analysis in large power grid / Yu Zhong / Martin D. F. Wang -- Incremental partitioning-based vectorless power grid verification / Dionysios Kouroussis / Imad A. Ferzli / Farid N. Najm -- Static timing analysis considering power supply variations / David Blaauw / Sanjay Pant -- Hybrid semiconductor/nanoelectronic circuits : devices, architectures, and design automation / Konstantin K. Likharev / Andre DeHon -- Performance analysis of carbon nanotube interconnects for VLSI applications / Kaustav Banerjee / Navin Srivastava -- DiCER : distributed and cost-effective redundancy for variation tolerance / Di Wu / Ganesh Venkataraman / Rabi N. Mahapatra / Jiang Hu / Qiuyang Li -- Worst-case analysis to obtain stable read/write DC margin of high density 6T-SRAM-array with local Vth variability / Yasumasa Tsukamoto / Nii Koji / Susumu Imaoka / Yuji Oda / Hirofumi Shinohara / Tomoaki Yoshizawa / Hiroshi Makino / Koichiro Ishibashi / Shigeki Ohbayashi -- Noise margin analysis for dynamic logic circuits / Mark R. Greenstreet / Suwen Yang -- Efficient analog platform characterization through analog constraint graphs / Alberto Sangiovanni Vincentelli / Fernando De Bernardinis -- Performance-centering optimization for system-level analog design exploration / Xin Li / Jian Wang / Tun-Shin Chen / Wanju Chiang / Lawrence T. Pileggi -- Hierarchical performance macromodels of feasible regions for synthesis of analog and RF circuits / Ranga Vemuri / Anuradha Agarwal -- Battery optimization vs. energy optimization : which to choose and when? / Ravishankar Rao / Naehyuck Chang / Sarma Vrudhula -- Dynamic voltage scaling for the schedulability of jitter-constrained real-time embedded systems / Bren C. Mochocki / Rolf Ernst / Razvan Racu / Xiaobo S. Hu -- Optimal integration of inter-task and intra-task dynamic voltage scaling techniques for hard real-time applications / Jaewon Seo / Taewhan Kim / Nikil D. Dutt -- Compiler-directed voltage scaling on communication links for reducing power consumption / Feihui Li / Mahmut Kandemir / Guangyu Chen -- Design automation issues for biofluidic microchips / Tamal Mukherjee -- Design of DNA origami / Paul W. K. Rothemund -- Kauffman networks : analysis and applications / Elena V. Dubrova / Andres Martinelli / Maxim Teslenko -- Parametrized model order reduction for nonlinear dynamical systems / Luca Daniel / Brad Bond -- Fast-yet-accurate PVT simulation by combined direct and iterative methods / Bo Hu / C. J. Richard Shi -- Robust automated synthesis methodology for integrated spiral inductors with variability / Yehia Massoud / Arthur B. Nieuwoudt -- Statistical technology mapping for parametric yield / Ashish K. Singh / Michael Orshansky / Murari Mani -- Reducing structural bias in technology mapping / Satrajit Chatterjee / Alan Mischenko / Timothy Kam / Xinning Wang / Robert Brayton -- Improving the efficiency of static timing analysis with false paths / Bo Yao / Hongyu Chen / Chungkuan Cheng / Mike Hutton / Shuo Zhou / Sridhar Srinivasan / Nanchi Chou / Peter Suaris / Truman Collins -- Total power-optimal pipelining and parallel processing process variations in nanometer technology / Nam Sung Kim / Taeho Kgil / Keith Bowman / Vivek De / Trevor Mudge -- Serial-link bus : a low-power on-chip bus architecture / Maged M. Ghoneima / Muhammad Khellah / Vivek De / Yehea Ismail / James Tschanz -- New decompilation techniques for binary-level co-processor generation / Frank Vahid / Greg Stitt
650 0 Electronic circuit design $x Data processing $v Congresses.
650 0 Computer-aided design $v Congresses.
650 0 Electronic circuits $x Testing $v Congresses.

650 6 Conception assistée par ordinateur $0 (CaQQLa)201-0124858 $v Congrès. $0 (CaQQLa)201-0378219
650 7 Computer-aided design. $2 fast $0 (OCoLC)fst00872701
650 7 Electronic circuit design $x Data processing. $2 fast $0 (OCoLC)fst00906866
650 7 Electronic circuits $x Testing. $2 fast $0 (OCoLC)fst00906898
655 7 Conference papers and proceedings. $2 fast $0 (OCoLC)fst01423772
710 2 IEEE Circuits and Systems Society.
710 2 ACM Special Interest Group on Design Automation.
740 0 Computer-Aided Design, 2005, ICCAD-2005, IEEE/ACM International Conference on.
740 0 Proceedings of the 2005 IEEE/ACM International Conference on Computer-Aided Design.
856 41 $3 IEEE Xplore $u http://ieeexplore.ieee.org/servlet/opac?punumber=10431
856 41 $u http://portal.acm.org/toc.cfm?id=1129601&coll=portal&dl=ACM&type=proceeding&idx=SERIES388&part=Proceedings&WantType=Proceedings&title=International%20Conference%20on%20Computer%20Aided%20Design $z Restricted to ACM Digital Library subscribers

ATTACHMENT 9C

LIBRARY OF CONGRESS AUTHORITIES

LC control no.: sh2008101464
LCCN Permalink: https://lccn.loc.gov/sh2008101464
HEADING: Computer-aided design Congresses

000 00498cz a2200169n 450
001 7437993
005 20110729160115.0
008 080206|| anannbabn |n ana
010 __ |**a** sh2008101464
035 __ |**a** (DLC)377000
035 __ |**a** (DLC)sh2008101464
040 __ |**a** DLC |**b** eng |**c** DLC
150 __ |**a** Computer-aided design |**v** Congresses
667 __ |**a** Record generated for validation purposes.
670 __ |**a** Work cat.: Computer aided control systems design 2000 (CACSD 2000), c2001
906 __ |**t** 8888 |**u** te00 |**v** 0
953 __ |**a** te00

◀ Previous Next ▶

Save, Print or Email Records (View Help)

Select Download Format: Text (Labelled Display) Press to SAVE or PRINT

Email Text (Labelled Display) to: Press to SEND EMAIL

Library of Congress
URL: https://www.loc.gov/

Mailing Address:
101 Independence Ave, S.E.
Washington, DC 20540

Library of Congress Authorities
URL: http://authorities.loc.gov/
Library of Congress Online Catalog
URL: https://catalog.loc.gov/

Questions, comments, error reports: Contact Us

AMD EX1021
AMD v. Realtek - Page 467

ATTACHMENT 9D

LIBRARY OF CONGRESS AUTHORITIES

LC control no.: sh 85042282
LCCN Permalink: https://lccn.loc.gov/sh85042282
HEADING: Electronic circuits Testing

000 00338cz a2200145n 450
001 4694182
005 20110729104924.0
008 860211|| anannbabn |n ana
010 __ |**a** sh 85042282
035 __ |**a** (DLC)sh 85042282
035 __ |**a** (DLC)40867
040 __ |**a** DLC |**c** DLC
150 __ |**a** Electronic circuits |**x** Testing
906 __ |**t** 8340 |**u** fk05 |**v** 0
953 __ |**a** xx00

Previous Next

Save, Print or Email Records (View Help)

Select Download Format: Text (Labelled Display) Press to SAVE or PRINT

Email Text (Labelled Display) to: Press to SEND EMAIL



Library of Congress
URL: https://www.loc.gov/

Mailing Address:
101 Independence Ave, S.E.
Washington, DC 20540

Library of Congress Authorities
URL: http://authorities.loc.gov/
Library of Congress Online Catalog
URL: https://catalog.loc.gov/

Questions, comments, error reports: Contact Us

ATTACHMENT 9E

Copyright United States Copyright Office

Try the Copyright Public Records System (CPRS) pilot with enhanced search features and filters.

Help | Search | History | Titles | Start Over

Public Catalog

Copyright Catalog (1978 to present)
Search Request: Left Anchored Title = International Conference on Computer Aided Design
Search Results: Displaying 2 of 6 entries

previous next

Labeled View

2005 IEEE/ACM international conference on computer-aided design.

Type of Work: Text
Registration Number / Date: TX0006287270 / 2005-11-22
Title: 2005 IEEE/ACM international conference on computer-aided design.
Copyright Claimant: IEEE
Date of Creation: 2005
Date of Publication: 2005-11-11
Copyright Note: Cataloged from appl. only.
Other Title: International conference on computer-aided design
Names: IEEE

previous next

Save, Print and Email (Help Page)

Select Download Format Full Record Format for Print/Save

Enter your email address: Email

Help Search History Titles Start Over

Contact Us | Request Copies | Get a Search Estimate | Frequently Asked Questions (FAQs) about Copyright | Copyright Office Home Page | Library of Congress Home Page

AMD EX1021
AMD v. Realtek - Page 471