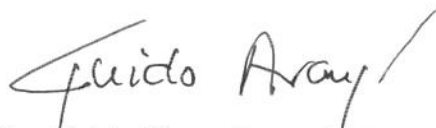


Impacto de Técnicas de Redução do Consumo de Energia no Projeto de SoCs Multimedia

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Yang Yun Ju e aprovada pela Banca Examinadora.

Campinas, Junho, 16, 2011.



Prof. Dr. Guido Costa Souza de Araújo
(Orientador)

Dissertação apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

FICHA CATALOGRÁFICA ELABORADA POR ANA REGINA MACHADO – CRB8/5467
BIBLIOTECA DO INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E
COMPUTAÇÃO CIENTÍFICA – UNICAMP

Y16i

Yang, Yun Ju, 1980-

Impacto de técnicas de redução do consumo de energia no projeto de SoCs Multimedia / Yang Yun Ju. – Campinas, SP : [s.n.], 2011.

Orientador: Guido Costa Souza de Araújo.

Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Computação.

1. Circuitos integrados de baixa tensão - Projeto auxiliado por computador. 2. Circuitos integrados - Integração em escala muito ampla - Projeto auxiliado por computador. 3. Circuitos integrados - Integração em escala muito ampla - Projetos e construção. I. Araújo, Guido Costa Souza de, 1962-. II. Universidade Estadual de Campinas. Instituto de Computação. II. Título.

Informações para Biblioteca Digital

Título em Inglês: The impact of design techniques in the reduction of power consumption of SoCs Multimedia

Palavras-chave em Inglês:

Low voltage integrated circuits - Computer-aided design

Integrated circuits - Very large scale integration - Computer-aided design

Integrated circuits - Very large scale integration - Design and construction

Área de concentração: Ciência da Computação

Titulação: Mestre em Ciência da Computação

Banca examinadora:

Guido Costa Souza de Araújo [Orientador]

Elmar Uwe Kurt Melcher

Paulo Cesar Centoducatte

Data da defesa: 16-06-2011

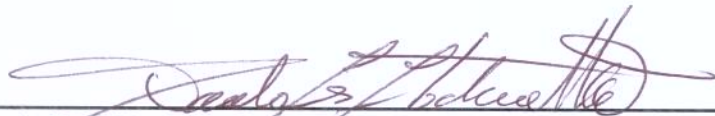
Programa de Pós Graduação: Ciência da Computação

TERMO DE APROVAÇÃO

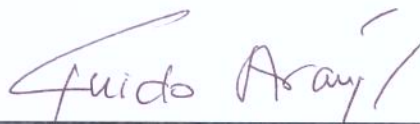
Dissertação Defendida e Aprovada em 16 de junho de 2011, pela Banca examinadora composta pelos Professores Doutores:



Prof. Dr. Elmar Uwe Kurt Melcher
LAD / UFCG



Prof. Dr. Paulo Cesar Centoducatte
IC / UNICAMP



Prof. Dr. Guido Costa Souza de Araujo
IC / UNICAMP

Impacto de Técnicas de Redução do Consumo de Energia no Projeto de SoCs Multimedia

Yang Yun Ju

Junho, 16, 2011

Banca Examinadora:

- Prof. Dr. Guido Costa Souza de Araújo (Orientador)
- Prof. Dr. Elmar Uwe Kurt Melcher
Departamento de Sistema de Computação, UFCG
- Prof. Dr. Paulo Cesar Centoducatte
Instituto de Computação, UNICAMP
- Prof. Dr. Mário Lúcio Côrtes(Suplente)
Instituto de Computação, UNICAMP
- Prof. Dr. Norian Marranghello(Suplente)
Departamento de Ciências de Computação e Estatística, UNESP

Resumo

A indústria de semicondutores sempre enfrentou fortes demandas em resolver problema de dissipação de calor e reduzir o consumo de energia em dispositivos. Esta tendência tem sido intensificada nos últimos anos com o movimento de sustentabilidade ambiental.

A concepção correta de um sistema eletrônico de baixo consumo de energia é um problema de vários níveis de complexidade e exige estratégias sistemáticas na sua construção. Fora disso, a adoção de qualquer técnica de redução de energia sempre está vinculada com objetivos especiais e provoca alguns impactos no projeto. Apesar dos projetistas conheçam bem os impactos de forma qualitativa, as detalhes quantitativas ainda são incognitas ou apenas mantidas dentro do 'know-how' das empresas.

Neste trabalho, de acordo com resultados experimentais baseado num plataforma de SoC¹ industrial, tentamos quantificar os impactos derivados do uso de técnicas de redução de consumo de energia. Nós concentramos em relacionar o fator de redução de energia de cada técnica aos impactos em termo de área, desempenho, esforço de implementação e verificação.

Na ausência desse tipo de dados, que relacionam o esforço de engenharia com as metas de consumo de energia, incertezas e atrasos serão frequentes no cronograma de projeto. Esperamos que este tipo de orientações possam ajudar/guiar os arquitetos de projeto em selecionar as técnicas adequadas para reduzir o consumo de energia dentro do alcance de orçamento e cronograma de projeto.

¹System on Chip

Abstract

The semiconductor industry has always faced strong demands to solve the problem of heat dissipation and reduce the power consumption in electronic devices. This trend has been increased in recent years with the action of environmental sustainability.

The correct conception of an electronic system for low power consumption is an issue with multiple levels of complexities and requires systematic approaches in its construction. However, the adoption of any technique for reducing the power consumption is always linked with some specific goals and causes some impacts on the project. Although the designers know well that these impacts can affect the design in a quality aspect, the quantitative details are still unknown or just be kept inside the company's know-how.

In this work, according to the experimental results based on an industrial SoC² platform, we try to quantify the impacts of the use of low power techniques. We will relate the power reduction factor of each technique to the impact in terms of area, performance, implementation and verification effort.

In the absence of such data, which relates the engineering effort to the goals of power consumption, uncertainties and delays are frequent. We hope that such guidelines can help/guide the project architects in selecting the appropriate techniques to reduce the power consumption within the limit of budget and project schedule.

²System On Chip

Agradecimentos

Agradeço

.... a Deus pelo sol e ar de todos os dias.

.... à minha família pela educação e apoio. Principalmente meus pais e irmãos.

.... ao meu orientador Prof. Guido Costa Souza de Araújo, idealizador de projeto universitário para treinamento de projetistas de Circuito Integrado, o Brazil-IP, do qual, adquiri grande parte dos conhecimentos da indústria semicondutora e o fluxo de projeto real.

.... ao prof. Elmar Melcher, diretor técnico de projeto Brazil-IP, por ter me instruído grande parte das competências técnicas de projeto de circuito integrado e ter me conduzido ao 'first silicon success' do primeiro chip da minha carreira profissional.

.... aos meus amigos e companheiros de guerra da primeira turma de Brazil-IP (André, Wilson, Isaac, Zurita, Osman, Felipe, Samuel) que passamos tantas noites trabalhando em 'debugar' os FPGAs e silícios, nos intervalos entre uma prova e outra.

.... aos colegas (alunos, professores e técnicos) da LSC/IC/UNICAMP que me ajudaram, mesmo que seja indiretamente, nesta longa jornada.

.... a todos aqueles que contribuíram, com boa fé, para a realização deste trabalho.

Sumário

Resumo	vii
Abstract	ix
Agradecimentos	xi
1 Introdução	1
1.1 Roadmap de Tecnologia de Projeto de Semicondutores	2
1.2 Motivação	3
1.3 Publicação	4
1.4 Organização de Dissertação	4
2 Low Power Fundamentals and Techniques	7
2.1 Power dissipation in CMOS technology	7
2.1.1 Dynamic power dissipation	7
2.1.2 Static power dissipation	9
2.1.3 Power-Related Effects	13
2.2 Power Analysis Model and Estimation Method	16
2.2.1 Internal Power Analysis	17
2.2.2 Leakage Power Analysis	19
2.2.3 Net Power Analysis	19
2.2.4 Power Estimation Methods	20
2.3 Low Power Techniques Fundamentals	22
2.3.1 Clock Gating	22
2.3.2 Operand Isolation	25
2.3.3 Multiple Threshold Voltage	28
2.3.4 Multiple Supply Voltage	29
3 Related works	35
3.1 NEC mobile phone system SoC	35

3.2	Fujitsu low-power test chip	37
3.2.1	NXP Low-Power SoC	37
4	Low Power Design Flow	41
4.1	Standard cell based design flow	41
4.2	Low Power Design Flow	44
5	Case study results	51
5.1	Leon3 Multiple Processor Platform	52
5.2	The MPEG-2 Video Decoder	54
5.3	Platform Configuration (Hardware/Software)	56
5.3.1	Sparc V8 Core	58
5.3.2	SRAM and SDRAM Memory Controller	61
5.3.3	Timers	62
5.3.4	Generic Input/Output Ports	63
5.3.5	UART Controller	63
5.3.6	IRQ Controller	63
5.3.7	External Memory Simulation Model	63
5.3.8	MPEG-2 Decoder Modification	64
5.3.9	Extract Switching Activity	67
5.4	Manufacturing Process	69
5.4.1	General Purpose Node	70
5.4.2	Low Power Node	72
5.5	Baseline Flow	73
5.5.1	System Requirement and Spec Definition	73
5.5.2	Library Qualification	73
5.5.3	Register Transfer Level Simulation	75
5.5.4	Logic Synthesis	75
5.5.5	Design For Test	75
5.5.6	Gate Level Co-Simulation	76
5.5.7	Placement and Routing	76
5.5.8	Static Timing Analysis	76
5.5.9	Parasitic Extraction	76
5.5.10	Post Layout Co-Simulation	76
5.5.11	Baseline Power Estimation	77
5.5.12	Experimental Results	77
5.6	Clock Gating Optimization	88
5.6.1	Logic Synthesis	88
5.6.2	Design for Test	88

5.6.3	Gate Level Co-Simulation	89
5.6.4	Placement and Routing	89
5.6.5	Static Timing Analysis	89
5.6.6	Parasitic Extraction	90
5.6.7	Post Layout Co-Simulation	90
5.6.8	Power Estimation	90
5.6.9	Experimental Results	90
5.6.10	Impacts of Clock Gating	99
5.7	Operand Isolation	100
5.7.1	Details of Implementation and Results	101
5.8	Multiple Threshold Voltage Optimization	101
5.8.1	System Specification and Requirement	102
5.8.2	Logic Synthesis (Mixed and Incremental Strategy)	102
5.8.3	Design For Test	104
5.8.4	Static Timing Analysis	105
5.8.5	Post-Layout Co-Simulation	105
5.8.6	Backend Tasks	105
5.8.7	Experimental Results and Impacts	105
5.9	Multiple Supply Voltage Optimization	108
5.9.1	System Specification and Requirement	108
5.9.2	Library Qualification of Level Shifters	109
5.9.3	Power Architecture and Power Domain Definition	110
5.9.4	Register Transfer Level Co-Simulation	112
5.9.5	Logic Synthesis	112
5.9.6	Power Rail Planning and Power Grids	113
5.9.7	Placement and Routing	114
5.9.8	Clock Tree Synthesis	117
5.9.9	Power Sign-Off	118
5.9.10	Post-layout Co-simulation	119
5.9.11	Power Estimation	119
5.9.12	Experimental Results and Impacts	119
5.10	Summary of all the experimental results	122
6	Conclusão e Trabalhos Futuros	123
6.1	Conclusão	123
6.2	Trabalhos Futuros	124
	Bibliografia	126

Lista de Tabelas

3.1	Fujitsu SoC Table	38
5.1	Library Corner Case Table	70
5.2	Low Power Library Corner Case Table	71
5.3	Low Power Design Library Qualification Table	74
5.4	Baseline Power Result at 300MHz	79
5.5	Baseline Power Result at 100MHz	80
5.6	Clock Gating Power Result at 300MHz	91
5.7	Impact Comparison between Baseline Flow and Clock Gating Flow	100
5.8	Result of the Mixed Synthesis	106
5.9	Power distribution of the Mixed Synthesis	107
5.10	Result of the Incremental Synthesis	108
5.11	Power distribution of the Incremental Synthesis	108
5.12	Multiple Power Supply result at 300MHz	120
5.13	Summary of All the Experimental Results	122

Lista de Figuras

2.1	Simplified Inverter Model	8
2.2	N-Type MOSFET in (a) Cut-Off Region (b) Linear Region	10
2.3	(a) High K Metal Gate (b) Oxide layer modeled as capacitor	12
2.4	Long term impact of electromigration effect	15
2.5	EDA tools power estimation model	16
2.6	Internal Power Model (2 Input AND)	18
2.7	Clock Gating concept	22
2.8	Clock Gating cell with DFT feature	25
2.9	Operand Isolation Concept	26
2.10	(a) Current leakage x Timing Delay (b) Library, Leakage and Delay Graph	29
2.11	Multiple V_{dd} Example	30
2.12	Voltage variation vs. Timing delay	33
3.1	Architecture of NEC 65nm Cell Phone SoC	36
3.2	Architecture of NXP 65nm SoC	38
4.1	Cell Based Design Flow	42
4.2	Low Power Design Flow	46
4.3	Power Intent Representation in the Low Power Design	47
5.1	Leon3 SoC example	53
5.2	Temporal and Spatial Redundancy of Frame Pictures	55
5.3	Basic Operations of MPEG-2 Encoder	56
5.4	MPEG-2 Video Encoder/Decoder Operations	57
5.5	Leon3 System for Power Estimation	58
5.6	Leon3 Combined Memory Controller Unit	61
5.7	MPEG-2 Decoder Data Flow	65
5.8	System Simulation and TCF File Construction	69
5.9	Inverter Circuit with Well-Bias	73
5.10	Baseline at 300MHz - Area by Module	81
5.11	Baseline at 300MHz - Total Power/Leakage Power by Module	82

5.12	Baseline at 300MHz - Power and Gate Numbers by Module	83
5.13	Baseline at 300MHz - Power/Gate Coefficient by Module	84
5.14	Baseline at 100MHz - Total Power/Leakage Power by Module	85
5.15	Baseline Power Comparison(100MHz X 300MHz)	86
5.16	Leakage Power Variation of Clock Gating Optimization	92
5.17	Area Variation of Clock Gating Optimization	93
5.18	Area and Leakage Power Variation of Clock Gating Optimization	94
5.19	Total Power Variation of Clock Gating Optimization	95
5.20	Total Power Comparison between Clock Gating and Baseline Flow	96
5.21	Power/Gate Coefficient Variations (Clock Gating and Baseline Flow)	97
5.22	Clock Gating Desclosing	98
5.23	(a) Incremental V_{th} Synthesis (b) Mixed V_{th} Synthesis	103
5.24	High-to-Low Level Shifter	110
5.25	Low-to-High Level Shifter	111
5.26	Power Domain of Leon3 System	112
5.27	High-to-Low Level Shifter in Destine Domain	114
5.28	Placement of Level Shifters during the Clock Tree Synthesis	115
5.29	Level Shifter Placement	116
5.30	Clock Tree of Multiple Power Domain	117
5.31	Clock Tree Synthesis at Multiple Power Domain	118

Capítulo 1

Introdução

A indústria de semicondutores sempre enfrentou fortes demandas em resolver problema de dissipação de calor e reduzir o consumo de energia em dispositivos. Esta tendência tem sido intensificada nos últimos anos com o movimento de sustentabilidade ambiental. Cada vez mais, a indústria semicondutora é obrigada a aplicar técnicas de redução de energia no nível do circuito e do sistema.

A demanda por baixo consumo de energia é um aspecto bem relevante no domínio de dispositivos móveis. Apesar de tempo de operação ativa dos equipamentos móveis ainda é limitado pela capacidade das baterias, o número de vendas deste segmento chega em centenas de milhões de unidades anuais. Mais do que nunca, junto com a preocupação crescente sobre o uso de recursos naturais, a redução de consumo de energia é um aspecto crítico em quase todos os imagináveis mercados de usuários finais.

Outro tipo de aplicação que também exige baixo consumo de energia é o centro de dados formado por sistema de fazenda de servidores ('Server Farm System'). Estes sistemas são compostos pelos numerosos poderosos servidores, isolados numa sala que deve ser mantida sob uma temperatura razoável. Considerando que cada servidor consome entre 100 ~ 200 W de potência, a maior parte deste consumo transforma-se em calor, que exige grandes sistemas de ar condicionado na refrigeração. Cerca de um terço a metade de energia consumida por uma fazenda de servidores é usada para resfriar os equipamentos[20]. Na maioria dos centros de dados (85%), o sistema de refrigeração é responsável por 65% do consumo de energia e os equipamentos de TI ¹ são responsáveis por apenas 33%. Os últimos 2% de consumo são perdidos nos sistemas de conversão de energia [20].

Os automóveis também incorporam grande quantidade de produtos eletrônicos, que são capazes de suportar até centenas de unidades de micro controladores. Eles controlam a força de tração dos motores, detectam as falhas, controlam a frenagem, o painel

¹Tecnologia da Informação

de instrumentos, o sistema de navegação, o sistema de entretenimento e garantem a alimentação do sistema em situações críticas. Quanto menor fica o consumo de energia que estes sistemas requerem, menor será o consumo de energia desses veículos.

No mercado de PC, os laptops e notebooks podem ser vistos como um computador que não deve ficar muito longe das tomadas elétricas. Com apenas poucas horas de duração, a bateria serve principalmente como uma ponte quando os laptops são movidos de um ponto a outro. Neste caso, a redução de consumo de energia é uma necessidade natural.

As eletrônicas de equipamentos médicos também exigem alta eficiência em gerenciar o consumo de energia para monitorar/manter saúde dos pacientes ao longo prazo (marca-passos artificiais, por exemplo).

Todos os exemplos citados acima mostram a importância de reduzir o consumo de energia. No entanto, a concepção correta de um sistema eletrônico de baixo consumo de energia é um problema de vários níveis de complexidade e exigem estratégias sistemáticas na sua construção. As perdas de energia podem acontecer no alto nível devido ao software de aplicação ou no baixo nível por causa de processos de manufatura de transistores que apresentam alta corrente de fuga. Assim como muitos aspectos de um projeto eletrônico, o problema de otimização de consumo de energia depende fortemente da capacidade dos projetistas em selecionar e utilizar as técnicas de redução de energia em todos os níveis de projeto.

1.1 Roadmap de Tecnologia de Projeto de Semicondutores

De acordo com o ITRS 2009², a atual tecnologia de projeto semicondutores enfrenta dois conjuntos de desafios oriundos de duas diferentes complexidades — *a complexidade de silício* e *a complexidade do sistema* [26]. A primeira complexidade se refere aos impactos causados pelo encolhimento de processo de manufatura dos transistores, junto com o uso de novos materiais na arquitetura de interconexão de sistema.

A segunda refere-se ao crescente número de transistores em uma única pastilha de silício e a demanda do consumidor por equipamentos de maiores funcionalidades, menores custos e menor tempo de lançamento ao mercado.

Várias outras complexidades acontecem em heterogeneidade dos componentes durante a integração do sistema de *SoC*³. Especialmente para este caso, a especificação e validação de conceito tornam-se desafiador. Para lidar com essas duas complexidades,

²International Technology Roadmap for Semicondutores, versão 2009

³System on Chip

a tecnologia de projeto deve fornecer novas metodologia para otimizar o custo de projeto e novas ferramentas para melhorar o reuso dos blocos de propriedade intelectual.

A outra tendência clara dada pelo ITRS–2009 é sobre os desafios no gerenciamento de consumo de energia. Como a redução de tamanho de transistores, a tecnologia de projeto é obrigada a enfrentar vários novos cenários:

- produtos baseados em UMP⁴ exigem cada vez mais capacidades de armazenamento e desempenho, com severas restrições em consumos de energia (tanto em estado de operação como em estado de 'stand-by').
- O aumento de densidade de potência agrava os efeitos térmicos e impacta a confiabilidade e o desempenho de projeto. A diminuição das tensões de alimentação agrava severamente a corrente de fuga e os efeitos de ruído. Estas tendências desafiam desde os recursos de interconexão nos chips, as máquinas ATE⁵ até os teste de burn-in. Especialmente para os sistemas que são alimentados pela alta tensão ou manufaturados em processos avançados (90nm, 65nm, 45nm), os projetos tendem a trabalhar numa temperatura elevada e rapidamente apresentarão vários mecanismos de falha em silício.
- O consumo de energia estática provocado pelas correntes de vazamento varia exponencialmente com os parâmetros de manufatura de transistores (largura de portas, espessura do óxido e a tensão limiar). Como consequência disso, análises de desempenho e a otimização de potência tornam-se desafiador.

1.2 Motivação

De acordo com as mensagens dadas pelo *ITRS*–2009, vimos que o custo de projeto é um fator fundamental que afeta, e afetarão a indústria de semicondutores nas próximas décadas. Este momento, a indústria de EDA⁶ ainda falta ferramentas ou soluções viáveis que sejam capazes de balacear as restrições orçamentárias (em termos de tempo ou de custo) com o fluxo de projeto.

No entanto, a partir de dados de vários trabalhos relevantes de projetos SoC⁷, vimos que a tecnologia de redução de consumo de energia foi melhorada significativamente devido aos esforços e colaborações entre a indústria EDA, as empresas semicondutoras e a comunidade acadêmica. No contexto do fluxo de projeto, as técnicas de redução

⁴Unidade de Micro Processamento

⁵Equipamentos de Testes Automáticos

⁶Electronic Design Automation

⁷System on Chip

de energia demonstram diferentes capacidades em reduzir o consumo em distintos níveis de abstração. Infelizmente, várias informações interessantes dos projetos são omitidas pelas empresas sob acordos de confidencialidade ou proteção industrial. Tais informações incluem: arquitetura de baixo consumo de energia, representação única de intenção de consumo de energia, esforço de implementação ('front-end' / 'back-end') e os impactos de desempenho de área.

A adoção de qualquer técnica de redução de energia sempre está vinculada com objetivos especiais e provoca alguns impactos no projeto. Normalmente, a equipe de projeto indiretamente admite todos as possíveis inconveniências derivadas das técnicas. Apesar dos projetistas conheçam bem os impactos de forma qualitativa, as detalhes quantitativas ainda são incognitas ou apenas mantidas dentro do 'know-how' das empresas.

Neste trabalho, de acordo com resultados experimentais baseado num verdadeiro plataforma SoC⁸, tentamos quantificar os impactos derivados do uso de técnicas de redução de consumo de energia. Nós concentramos em relacionar o fator de redução de energia de cada técnica aos impactos em termo de área, desempenho, esforço de implementação e verificação.

Na ausência desse tipo de dados, que relacionam o esforço de engenharia com as metas de consumo de energia, incertezas e atrasos serão frequentes no cronograma de projeto.

Esperamos que este tipo de orientações possam ajudar/guiar os arquitetos de projeto em selecionar as técnicas adequadas para reduzir o consumo de energia dentro do alcance de orçamento e cronograma de projeto.

1.3 Publicação

Uma versão reduzida do presente trabalho, incluindo desde o capítulo 2 até o capítulo 6, será submetida para o **First Workshop on Circuits and System Designs (WCAS 2011) em João Pessoa, Paraíba, Brasil.**

1.4 Organização de Dissertação

A dissertação será apresentada da seguinte forma:

- Capítulo 1: **Introdução**
- Capítulo 2: **Low Power Fundamentals and Techniques** descreve as fontes de dissipação de energia em transistores CMOS seguido de alguns conceitos fundamen-

⁸System on Chip

tais de análise de potência nas ferramentas de EDA ⁹. Depois disso, fundamentos de algumas técnicas de redução de consumo de energia são apresentados.

- Capítulo 3: **Related Works** descreve alguns projetos relevantes de redução de consumo de energia e seus resultados.
- Capítulo 4: **Low Power Design Flow** apresenta as diferenças entre o fluxo de projeto ASIC convencional e o fluxo de projeto de redução de consumo de energia.
- Capítulo 5: **Case Study Results** apresenta um estudo de caso baseado numa plataforma SoC (Leon3) quando as técnicas de redução de consumo de energia são aplicadas. Descrevemos também as vantagens e desvantagens das técnicas utilizadas.
- Capítulo 6: **Conclusão e Trabalhos Futuros**. Concluimos a dissertação e mostramos possíveis trabalhos futuros.

⁹Electronic Design Automation

Capítulo 2

Low Power Fundamentals and Techniques

Before we developing the fundamentals of low power techniques, we briefly discuss some mechanism of power consumption and several power related effects in CMOS circuits.

2.1 Power dissipation in CMOS technology

The traditional analysis approach considers a simple inverter as basic model to explain components of power consumptions. Just like in the Figure 2.1, the power dissipation happens when the inverter is in use and the total power can be decomposed into two classes : the static power and the dynamic power.

Roughly, the total power of an inverter gate can be described as the follower equation

$$P = P_{dynamic} + P_{static} \quad (2.1.1)$$

and each one of these two components are described in the next two subsections. We also denote the dynamic power as $P_{dynamic}$, and the static as P_{static} .

2.1.1 Dynamic power dissipation

The dynamic power ($P_{dynamic}$) is usually the dominant component¹ occurs only when the node voltage is switched. The model from Figure 2.1 is simplified as it assumes that all capacitance is connected at the output node. The output capacitance is divided into two components C_1 and C_2 . Assume first the input goes to **ground** (logic zero), then

¹From 90 nm process forward, although, this picture suffers deep changes. Please consult section 2.1.2 for more details.

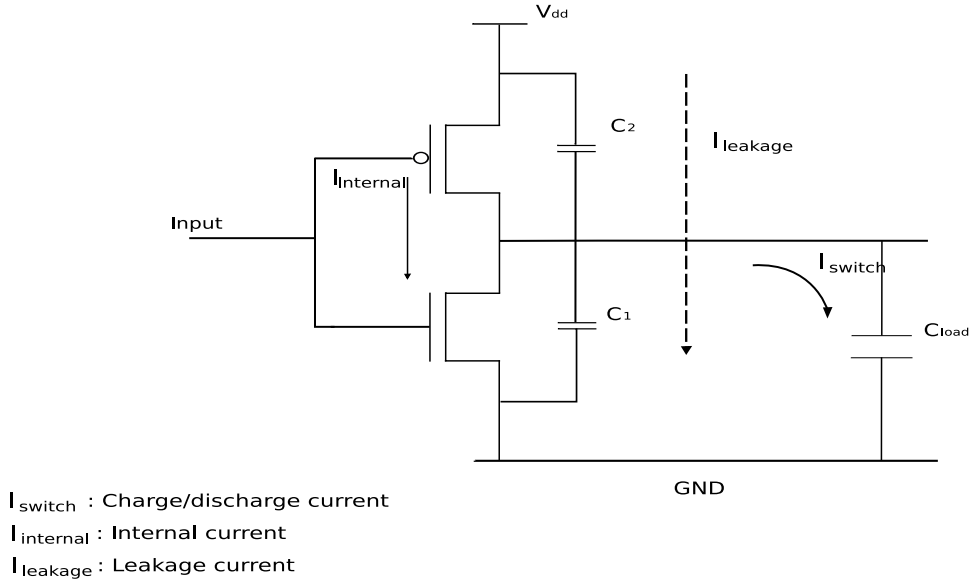


Figura 2.1: Simplified Inverter Model

the C_1 is connected to ground, and the C_2 is to V_{dd} (note²). The charge of C_1 changes from C_1V_{dd} to zero, as the NMOS transistor switches to GND. The charge of C_2 changes from zero to C_2V_{dd} through V_{dd} . During the next phase, when the input voltage returns to low again, C_2 is shorted through the PMOS transistor, and C_1 is charged back to C_2V_{dd} . As a result, we take a charge of $(C_1 + C_2)V_{dd}$ is taken from the power supply for a full switch cycle, or a total energy of $(C_1 + C_2)V_{dd}^2$. In a system whose frequency equals f Hz, we will get the power consumption value $P = f(C_1 + C_2)V_{dd}^2$. If we simplify the gate capacitance as $C_{load} = C_1 + C_2$ using the miller effect, and introduce the switching probability for the input node as α , we can express the dynamic power consumption of the CMOS inverter as:

$$P_{dynamic} = C_{load}V_{dd}^2\alpha f \quad (2.1.2)$$

Therefore, when a static CMOS gate is switched by an input signal with a nonzero rise or fall time, both NMOS and PMOS transistor will conduct simultaneously for a short period. During this interval, there will be a short-circuit current, flowing directly between V_{dd} and ground and not participating in charging or discharging of C_{load} . So, for the total component of the dynamic power, we should take into account the short-

² C_1 and C_2 are result from gate oxide capacitance(C_{ox}), gate junction capacitance(C_j) and the gate interconnect capacitance (C_{int})

circuit effect. Hence the equation 2.1.2 should be corrected to:

$$P_{dynamic} = C_{load}V_{dd}^2\alpha f + P_{short} \quad (2.1.3)$$

where

$$P_{short} \approx V_{dd}I_{short}\frac{\tau_{in}}{4}2f \approx V_{dd}^2f\frac{C}{10} \quad (2.1.4)$$

The second term in equation 2.1.3 represents the short-circuit power. When the inverter's input is around $\frac{V_{dd}}{2}$ during the turn-on and turn-off switching transients, both the PMOS and NMOS are on and a short circuit current I_{short} flows from V_{dd} to ground. The width of this short-circuit current pulse is about $\frac{1}{4}$ of the input rise and fall time [17][23].

If we combine the equation 2.1.3 with Equation 2.1.4, we have:

$$P_{dynamic} = C_{load}V_{dd}^2\alpha fk \quad (2.1.5)$$

Since the inverter is in the dynamic operation mode, the probability (α) to have a short circuit in each switching transition is 100%. As the result, the dimensionless factor k becomes $\alpha + 0.1 = 1.10$, which can be lower bounded to 1 [17][23]. The dynamic power equation, therefore, can be expressed in a simplified form:

$$P_{dynamic} = C_{load}V_{dd}^2\alpha f \quad (2.1.6)$$

2.1.2 Static power dissipation

The static dissipation is the power consumed even a gate is not switching. When the transistors are nominally OFF, they leak small amount of current. Leakage effect include **subthreshold conduction** between the source and the drain, **gate leakage** from the gate terminal to the body, and **junction leakage** from source to body and drain to body. The subthreshold conduction is caused by thermal emission of carriers over the potential barrier set by the threshold voltage. The gate leakage is a quantum-mechanical effect caused by the tunneling through the thin gate dielectric. Junction leakage is caused by the current through the PN junction between the source/drain diffusion and the body terminal.

According to this description, the static power can be understood as the sum of the three types of dissipation and can be represented in the next equation[23]:

$$P_{static} = (I_{sub} + I_{gate} + I_{jun})V_{dd} \quad (2.1.7)$$

where the I_{sub} denotes the subthreshold leakage current, the I_{gate} denotes the gate leakage current and the I_{jun} denotes the junction leakage current. For each one of components, we present a brief description in next :

Subthreshold Leakage

In an n-MOS transistor, the substrate is composed of p-type silicon, which has positively charged mobile holes as carriers, as show in Figure 2.2(a) and 2.2(b).

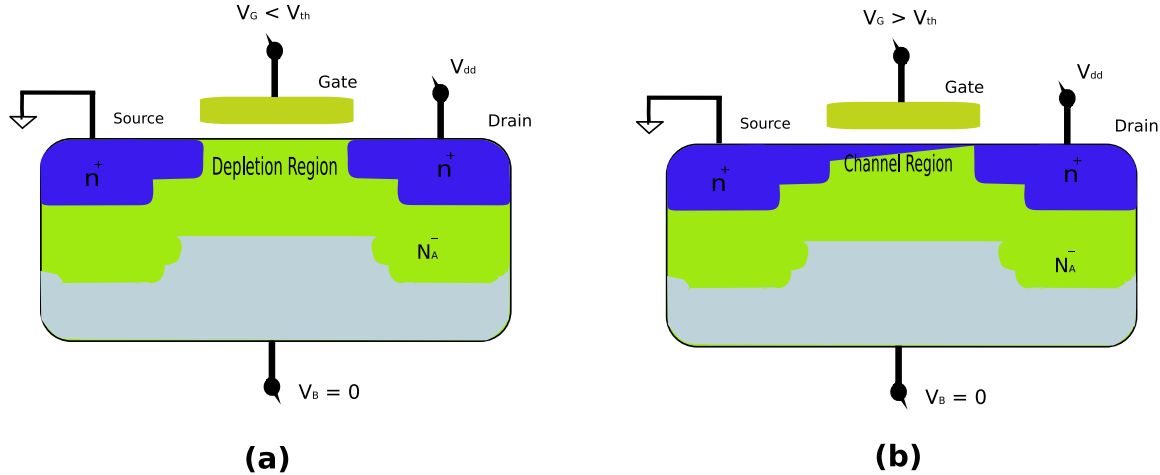


Figure 2.2: N-Type MOSFET in (a) Cut-Off Region (b) Linear Region

When a positive voltage is applied on the gate, an electrical field causes the holes to be repelled from the interface, creating a depletion region containing fixed negatively charged acceptor ions. A further increase in the gate voltage eventually causes electrons to appear at the interface, in what is called an inversion layer, or a channel (as show in Figure 2.2(b)).

The gate voltage at which the electron density at the interface is the same as the hole density in the neutral bulk material is called the threshold voltage. Practically speaking, the threshold voltage is the voltage at which there are sufficient electrons in the inversion layer to make a low resistance conducting path between the *MOSFET* source and drain.

If the gate voltage (V_{gs}) is below the threshold voltage, as show in Figure 2.2(a), the transistor is turned-off (we say that the transistor is in the subthreshold region or weak-inversion region). Ideally no current should flow from the drain to the source of the transistor. However, there is a small current (I_{ds}) that can be described by the next equation [23]:

$$I_{ds} = I_{off} \times 10^{\frac{V_{gs} + n(V_{ds} - V_{dd}) - k\gamma V_{sb}}{S}} \times \left(1 - e^{\frac{-V_{ds}}{v_T}}\right) \quad (2.1.8)$$

where the I_{off} is the subthreshold current at $V_{gs} = 0$ and $V_{ds} = V_{dd}$. V_{gs} is the voltage between the source and the gate terminal, while V_{ds} is the voltage between the drain and the source terminal. V_{sb} is the voltage between the source and the body terminal. The n is a process-dependent factor, affected by the depletion region characteristics and is typically ranging 1.3 - 1.7 for CMOS process. The v_T represents the thermal voltage and it equals to 26mV at the room temperature. The S represents the subthreshold slope, and indicates how much the gate voltage must drop to decrease the leakage current by an order of magnitude. A typical value for this is 100 mV/decade at the room temperature[23].

Gate Leakage

According to the quantum mechanics, the electron cloud surrounding an atom has a probabilistic spatial distribution. For gate oxide thinner than 15 - 20 Å, there is a nonzero probability that an electron in the gate will find itself on the wrong side of the oxide where it will get whisked away through the channel[23]. This effect of carriers crossing a thin barrier is called tunneling and results in leakage current through the gate.

Two physical mechanism for gate tunneling are called Fowler-Nordheim (FN) tunneling and direct tunneling. FN tunneling is most important at high voltage. Direct tunneling is most important at low voltage with thin oxide and is the dominant leakage component. The direct tunneling can be describes as [1]

$$I_{gate} = WA \times \left(\frac{V_{DD}}{t_{ox}}\right)^2 \times e^{-B \frac{t_{ox}}{V_{DD}}} \quad (2.1.9)$$

where the W represents the width of the gate, t_{ox} is the unit area of the gate oxide, while the A and B are experimental constant, depend on the manufacturing technology.

Junction Leakage

The PN junction between diffusion and the substrate or the well form diodes. The substrate and well are tied to GND or VDD to ensure these diodes do not become forward biased in normal operation. However, the reverse-biased diodes still conduct a small amount of current I_D , that can be described as :

$$I_D = I_S \times (e^{\frac{V_D}{v_T}} - 1) \quad (2.1.10)$$

where the I_D and I_S stand respectively for the diode current and diode reverse-biased saturation currents. The terms D and S are not related with the drain or source. The I_S depends on doping levels and on the area and perimeter of the diffusion region and V_D is the diode voltage. Normally this type of dissipation ranges from 0.1-0.01 fA/ μm^2 , almost negligible compared with the other two mechanism of leakage[23].

In process with feature size above the 180nm, leakage was typically insignificant except in very low power applications. In 90nm and 65nm process, threshold voltage has reduced to the point that the subthreshold leakage reaches level of 1 - 10s of nA per transistor, which is significant when multiplied by million or billion transistors on a chip. In 45nm, the oxide thickness reduces to the point that the gate leakage becomes comparable to subthreshold leakage unless the HKMG dielectric are employed[23]. Overall, leakage has become an important design considerations in nanometer process.

High K Metal Gate (HKMG)

The HKMG is a new manufacturing approach³ emerges to overcome the subthreshold leakage problem[6]. A different material for oxide isolation layer is chosen to design the dielectric layer. Different from the conventional *CMOS* isolation material (silicon dioxide based), the HKMG technology replace the isolation layer by a hafnium-based one, which presents a high dielectric constant k [8].

Silicon dioxide has been used as a gate oxide material for decades. As transistors have decreased in size, the thickness of the silicon dioxide gate dielectric has steadily decreased to increase the gate capacitance and thereby drive current and device performance. As the thickness scales below 2 nm, leakage current due to tunneling effect was increased drastically, which lead to larger power consumption and reduced device reliability. Replacing the silicon dioxide gate dielectric with a high- k material allows increased gate capacitance without the correspondent leakage effects[8].

The principle of high-K technology is based on modification of gate oxide capacitance of a MOSFET gate. Typically the insulator layer can be modeled as a parallel plate capacitor, just like Figure 2.3.

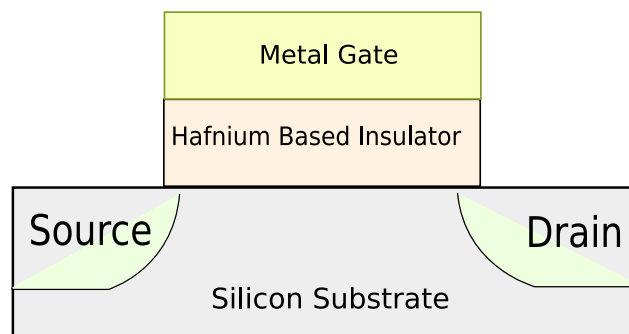


Figura 2.3: (a) High K Metal Gate (b) Oxide layer modeled as capacitor

If we ignore quantum mechanical and depletion effects from the silicon substrate and

³Specially, begins with 45nm process

gate, the capacitance C of this parallel plate capacitor can be described by

$$C = \frac{K \times \varepsilon_0 \times A}{t} \quad (2.1.11)$$

where :

A : capacitor area

K : relative dielectric constant of the material (3.9 for silicon dioxide)

ε_0 : permittivity of free space

t : thickness of the capacitor oxide insulator

Since leakage limitation constrains further reduction of t , an alternative method to increase the gate capacitance is to modify the k by replacing silicon dioxide with a high- k material. In such a scenario, a thicker gate layer might be used which can reduce the leakage current flowing through the structure as well as improving the gate dielectric reliability. Based on this principle, semiconductor foundries now shrink gate equivalent oxide thickness at 32 nm from 18Å to just 10Å (1 nm) and, at the same time, gate leakage is reduced by 10 times[8]. From several IDM's⁴ process specification manual, the usage of HKMG roughly offers a 40% reduction in power, and 20% increase in performance with V_{dd} at 1V[8].

The implementation of high- k gate dielectric is one of several strategies developed to allow further miniaturization of microelectronic components, colloquially referred to as extending Moore's Law.

2.1.3 Power-Related Effects

As the technology node reduced, some power-related effects impact deeply on the design reliability. Next, we list some of them:

IR-Drop

This is a voltage reduction occurred on the supply network (V_{dd}) or ground (GND) of integrated circuits. The IR term comes from the *Ohm's* law, which states that current I flows through an effective resistance R , therefore, introduces a voltage drop as given by the equation $V = IR$.

Usually, designers assume the availability of an ideal power supply that can instantly deliver any amount of current to maintain the specific voltage throughout the chip. However, the combination of increasing current per-unit of area on the die and narrower metal line widths⁵ causes localized voltage drops within the power grid, leading to decreases power supply voltage at transistors. These localized drops in the power

⁴IDM stand for Integrated Device Manufacturer

⁵which causes an increase in the power-grid resistance

supply voltage decrease the operating voltage of the chip, potentially causing timing and functional failures.

IR-Drop of Ground

This is a voltage increase that occurs on the ground network (GND). All the current that is sourced to the network combined with a finite resistance of the network leads to localized increase on the ground voltages around the chips. This increase in the ground voltage also decrease the operating voltage of the chip, resulting in the timing and functional failures.

The number of failures from poor power-grid network has become significant only recently, many designers still do not look at power distribution as a potential source of chip failures. Symptoms of IR -drop of VDD and GND could include the following:

1. Non-Functional chips: the total chip malfunction happens when the global IR drop of VDD or GND exceed the safe operation range. The failure resembles a logical functional failure or a manufacturing problem, although functional simulation indicates that the design is logically correct.
2. Intermittent or Data-dependent functional failures: this situation happens when we face with a great number of cells in close proximity switching simultaneously and causing local IR drop of VDD or GND . A higher power-grid resistance on that specific region of the chip can also provoke the same effect. In normal operation, this kind of "*sensitization*" might not occur, except when a specific data input activates the problem. As result, this effect appears on that part of chip.
3. Intermittent or Data-dependent timing failures : Just like the intermmitent functional failure, some specific data inputs can cause IR drop of VDD or GND and induce a timing failure problem by changing the interconnection resistance or capacitance characteristics. In most cases, the whole chip speed will be slowed down, appearing to be *setup* or *hold* timing violations.
4. Timing library problem : traditionally, the timing library of a process are modeled by assuming certain region of supply IR drop in the design. This approach adds significant timing margins and makes performance estimation highly unpredictable. Chips can fail because the IR drop was actually higher, or they can operate much faster than the simulation results.
5. Over-designed power grid : too prevent possible IR drop effects, supply structure and interconnection routing are heavily designed and consume a significant area of chip.

Electromigration

It is used as a general term to describe failure mechanisms in the metal wires of a chip caused by the move of metal atoms in a wire because of high current stress. In such case, electrons collide with diffusion metal atoms and impel the atoms in the direction of electron flow. This collision produces a mass flux opposite to that of the current, and the divergence in this mass flux can result in damage to the conductor in form of void or killocks. Over a long period of time, the metal atom may move in the direction of the electron flow, causing two principal failure mechanism :

1. If enough atoms are moved, the wire effectively breaks and becomes an open circuit (Figure 2.4). In the long term, the interconnecting wires will breakdown, and in some specific cases, the design shows failure symptoms at the post-manufacturing test stage.



Figure 2.4: Long term impact of electromigration effect

2. If enough atoms move to the same location, a short to an adjacent metal wire can be create. This phenomenon is commonly known as fusion.

Therefore, the electromigration is a long-term wearout mechanism of the chip interconnect wires. Its typical failure symptoms are primarily a change in either the timing or the functionality of a chip over time. In some case, the wires that provide unique connectivity in the circuit will cause a total function failure. Wires that are inherently redundant, such as a meshed power grid, exhibit symptoms of higher IR drop of V_{DD} or GND after the failures damage the power grid section. In the case of shorts connections between wires on the chip, total functional failure occurs.

Due to this, starting from the 130nm process, the power rail analysis methodologies have been wildly incorporated into the physical verification in order to validate the power network before the tape-out. Rails analysis should accurately highlight the IR-drop of *VDD* or *GND* and the electromigration problems, which could be caused by open circuits, missing vias/via arrays, high current densities, lack of power stripes and narrow widths of power routing. This technique is also suitable to identify long period reliability problems of chips under the influence of electromigration effects.

2.2 Power Analysis Model and Estimation Method

The preview sections (2.1.1 and 2.1.2) described the CMOS power dissipation sources of the semiconductor manufacturing process. In this section, we describe some fundamental concepts of power analysis modeling and estimation. The classical power estimation tools subdivided power consumption into three components: the internal power, the leakage power and the net power. Figure 2.5 shows a simplified power estimation model.

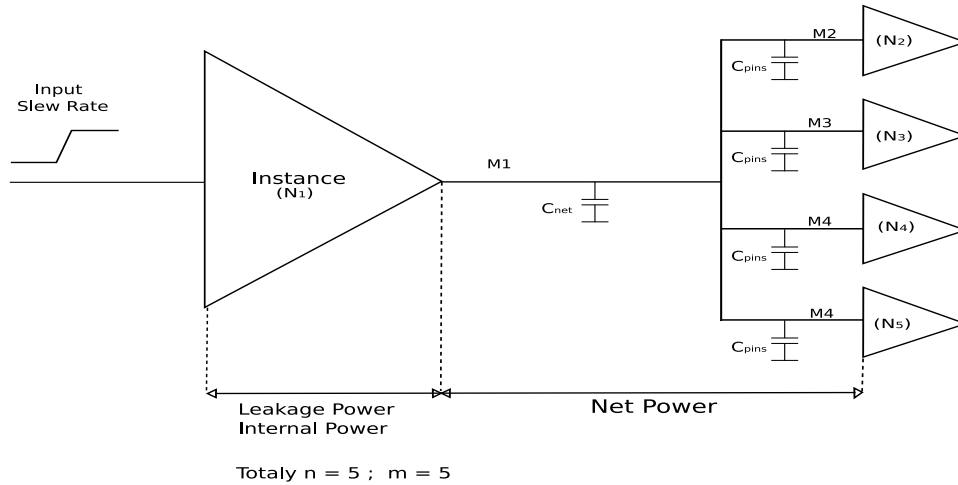


Figura 2.5: EDA tools power estimation model

Usually the power estimator returns an average power consumption as opposed to cycle-by-cycle power consumption. The classical estimators need switching information (input/output switching probability and pin/net toggle rates) to calculate the power. The follow equation shows how a generic power engine calculates the power dissipation:

$$P_{total} = \sum_{i=1}^n P_{instance} + \sum_{j=1}^m P_{net} \quad (2.2.1)$$

where,

$$\sum_1^n P_{instance} = \sum_1^n P_{leakage} + \sum_1^n P_{internal} \quad (2.2.2)$$

and

$$\sum_1^m P_{net} = \sum_1^m \frac{1}{2} \times C_{load} \times V_{dd}^2 \times ToggleRate \quad (2.2.3)$$

We note that,

- $P_{leakage}$ is the leakage power in a cell, which is obtained from the technology library. Recent power analysis tools usually support both constant and state-dependent leakage power models.
- $P_{internal}$ is the internal power of the cell, which is obtained and calculated from the technology library.
- n is the number of cells in the design.
- P_{net} is the net power consumption in the design (see section 2.2.3).
- m is the number of nets in the design.

In the next three sections some basic concepts related to each one of the above components will be explained.

2.2.1 Internal Power Analysis

The internal power dissipation of a cell includes short circuit power and the switching power when charging or discharging the internal net capacitance. The values of the internal power could be calculated from the power look-up tables in the library. These tables are a function of the input slew rate (SR) and the output load capacitance C_{load} . Some libraries also have three-dimensional power tables which are function of the input slew, output load capacitance and a second output load capacitance.

Instead of one table, a cell can contain two tables to model the internal power at more advanced process technology: one table to rise transitions and another to fall transitions. Taking into account the path dependence, the internal cell power is determined as follow way:

$$P_{internal} = \sum_{per\ arc} (TR_{arc_{ij}} \times \Phi(S_i, C_j)) + \sum_{per\ pin} (TR_i \times \Phi(S_i)) \quad (2.2.4)$$

Now, considering the Figure 2.6 as a simplified model for internal power estimation, we can analysis the equation 2.2.4:

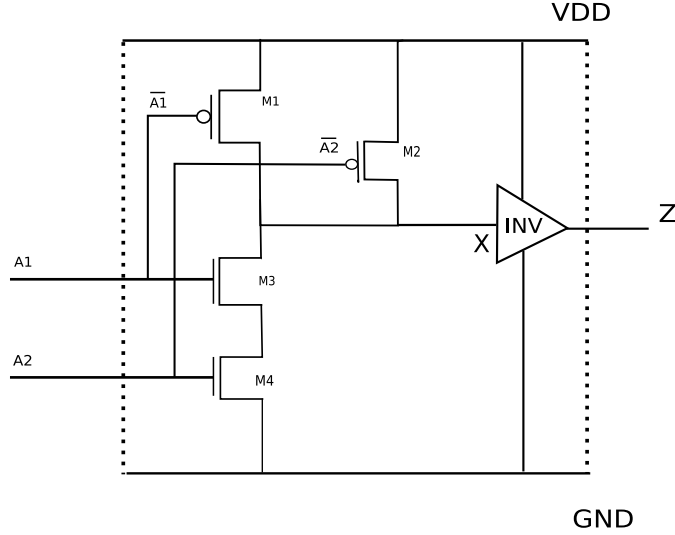


Figura 2.6: Internal Power Model (2 Input AND)

- TR is the effective toggle rate of an *arc* or a *pin*. The effective toggle rate of arc_{ij} depends on the probability that the *arc* gets activated and on the toggle rate on the input pin_i . The probability that the *arc* gets activated is determined by the function of output pin_j and the probabilities of the other inputs pins. In our case, the arc_{ij} could be the path from $A1$ to Z or $A2$ to Z . $A1$, in our case, represents the pin_1 , and the $A2$ represents the pin_2 .
- Φ is the power calculated from a look-up table in the process technology file.
- S_i is the slew rate of the $input_i$ causing a toggle rate on $output_j$. In our case, the slew rates of input pins can be represented by $S_1(A1)$ and $S_2(A2)$, causing toggle rate on Z ($output_1$).
- C_j is the load capacitance of $output_j$. In our case, it can be presented by C_1 , which is the load capacitance at Z ($output_1$).
- The total internal power of this gate, then, can be described as :

$$P_{internal} = Arc_{A1 \rightarrow Z} + Arc_{A2 \rightarrow Z} + Pin_{A1} + Pin_{A2} \quad (2.2.5)$$

where

$$Arc_{A1 \rightarrow Z} = TR_{A1 \rightarrow Z} \times \Phi(SlewRate_{A1}, C_{output_Z})$$

$$Arc_{A2 \rightarrow Z} = TR_{A2 \rightarrow Z} \times \Phi(SlewRate_{A2}, C_{output_Z})$$

$$Pin_{A1} = TR_{A1} \times \Phi(S_{A1})$$

$$Pin_{A2} = TR_{A2} \times \Phi(S_{A2})$$

2.2.2 Leakage Power Analysis

The leakage power dissipation component should be obtained from the technology library of the chosen manufacturing process (more specifically, from the logical synthesis libraries). In some technology, library vendors provide cells with different threshold voltages, enabling the possibility to perform some leakage power optimization techniques. For old process, it was usually modeled as a constant value, and in more advanced nodes, some cell leakage power is a function of the input state. making the leakage estimation more accurate. In this case, the leakage power for the specific cell is determined as follow equation :

$$P_{leakage} = \sum_{state=1}^k (P_{state_leakage} \times Probability_{state}) \quad (2.2.6)$$

where,

- K is the number of total possible states of the cell.
- $P_{state_leakage}$ is the leakage of the cell in a specific state(logic Zero or One).
- $Probability_{state}$ is the probability that the cell stays in the actual state.

2.2.3 Net Power Analysis

The net power includes dissipation during net/pin capacitance charging and discharging process. The total power consumed by a determined net can be expressed as :

$$P_{net} = \frac{1}{2} \times C_{load} \times V_{dd}^2 \times TR \quad (2.2.7)$$

where,

- C_{load} is the sum of parasitic capacitance from a net and it associated pins.

- V_{dd} is the supply voltage, expressed usually in volts
- TR is the toggle rate associated with the net switching activity.

2.2.4 Power Estimation Methods

After introduced power analysis models from the preview sections, we present some power estimation methods. The two most common approaches for the power estimation are the vector-based and the propagation-based techniques. We describe both of them in following sections:

Vector-based Estimation

The vector-driven method uses switching activity files (output of a discrete event simulator) to obtain the number of transitions associated to each net and I/O pin. Normally, the switching activity files are dumped in VCD⁶, SAIF⁷ or TCF⁸ formats. These files result from a good functional coverage simulation, and provide more accurate result when estimate power.

The power engine calculates the transition probability of each pin/net based on the activity information. After this step, the duty cycle of each net (which is responsible for state dependent internal/leakage power calculation) is also estimated. Finally, the power estimation is performed by the engine based on the previews sections (2.2.1, 2.2.2 and 2.2.3).

The most complicated step of estimation approach happens when one tries to produce a reliable input vector with good functional and power coverage. The vector-based approach is apply in the follow conditions:

- Gate-level simulation is possible at the full-chip level.
- Gate-level simulation provides sufficient functional coverage for the design.
- The switching activity vector include information that could cause highest power consumption.

⁶Value Change file format, was defined along with the Verilog HDL by the IEEE Standard 1364-1995 in 1995

⁷Switching Activity Interchange Format

⁸Toggle Count Format, from Cadence Design System™

Propagation-based Estimation

The propagation-based approach is vector-independent and provides coverage for all nets in a design. However, the accuracy depends on a good starting value, i.e, an initial information about the switching probabilities at the primary inputs of the design. Simple examples are clock, reset or enable inputs. To obtain an accurate prediction without information about the switching probabilities of these inputs is difficult and frequently results in inaccurate prediction and overestimated power consumption. We can divide the propagation method into several categories:

- Propagation through combinational cells: this is based on the library functions of each manufacturing process. The power engine get the previously defined combinational cell from the technology file and uses these functions to propagate the switching information through the synthesized netlist. The engine propagates duty cycle through combinational cell, at the same time, breaks combinational loops to estimate a precise result. The method is based upon internal heuristic of each power engine and activity information from other neighboring pins.
- Propagation through sequential cells: this is based on the activity of the pins like: inputs, set, reset or scan enable. However, most of these cells are in a sequential loop, like the state machine whose activity propagation mechanism is based on heuristic algorithms. A big drawback of the method is imprecise when the sequential cells from a loop structure, and the heuristic diminish the switching activity toward Zero.
- Propagation through macro cells⁹: the major component of power in macro cells is the internal power which is highly sensitive to the activity on reads and writes signals. A small change in activity on these two signals can cause a large change in the internal power numbers. Basically, the power estimation is based on process library as the case of combinational cells.

In general, this method is very useful in the early stages of the design, due to simplicity and speed. As long as the project approximates the sign-off step, the vector-based approach is highly recommended instead.

⁹these ones can be a memory cell or a third-party IP

2.3 Low Power Techniques Fundamentals

After a brief description about CMOS power dissipation and estimation methods, now we will present several power reduction techniques used in this work, including their advantages and drawbacks.

2.3.1 Clock Gating

The clock gating is a classical and powerful technique in optimizing dynamic dissipation. It is based on eliminating unnecessary clock toggle activities in the storage elements¹⁰.

Any storage element¹⁰ in the digital design is based on flip-flops. By analyzing the basic functionality of any flip-flop, we find that even though data is loaded into the design with very low frequency, the clock signal keeps the toggle behavior at every clock cycle.

Very often, the clock signal also drives a large capacitive load which makes clock signal a major source of dynamic dissipation. By inserting a special control signal (associated with a special gating logic) to control a group of flip-flops, one can eliminate unnecessary clock toggles and reduce the dynamic power effectively. We show the clock gating concept and implementation in Figure 2.7

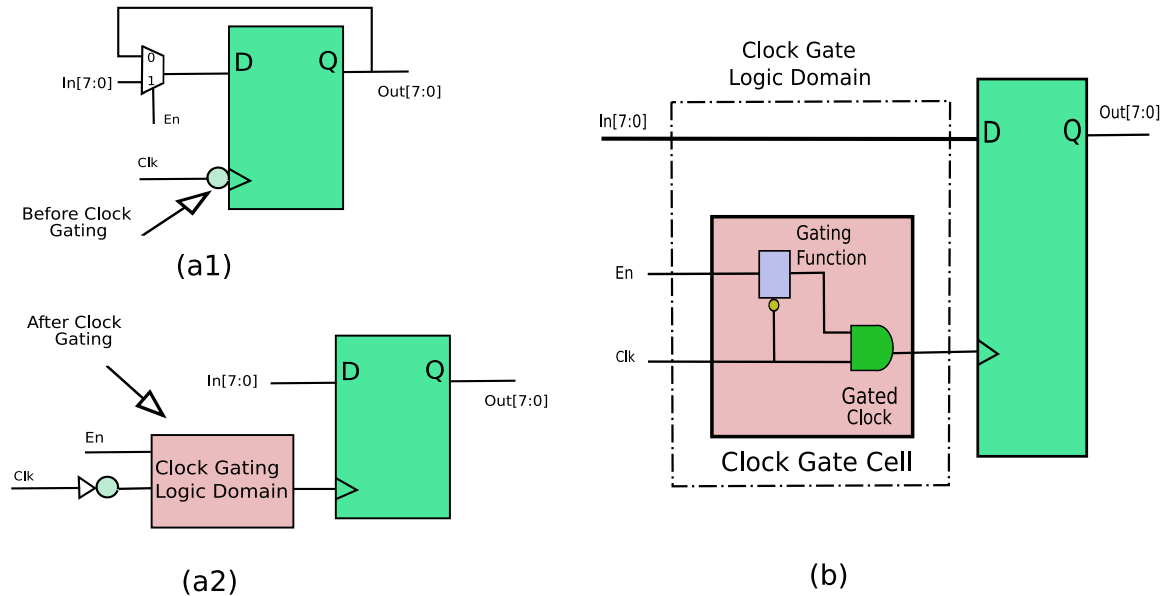


Figura 2.7: Clock Gating concept

¹⁰For example, registers, static rams, dynamic rams

Clock Gating can be implemented by inserting clock gating circuits into a sequential cell (Figure 2.7(a1)), altering the enable signal to gate the clock pin of the flip-flop just like the 2.7(a2). As a result, the dynamic power dissipation could be reduced:

- The dynamic power is not dissipated during the idle period when the sequential circuit is turned-off by the gating function (see in Figure 2.7(b)).
- Dynamic power is saved if the enable signal is down. From the Figure 2.7(b), with the enable signal set to be in down state, the switching activity in clock gating cell is reduced to zero, remaining the leakage dissipation.
- With replacement of an enable circuit in the original design (Figure 2.7(a1)), the power dissipation at the input pin is reduced to net power, which is much lower than when the circuit was enabled.

Possible Impacts, Requirements and Constraints

We understand how the clock gating could be used in reducing dynamic dissipation, it also brings some drawbacks and requirements to the implementation and DFT¹¹ structure. The impacts can happen in follower topics like: power, area, DFT, timing, PDK¹², verification effort and back-end implementation.

Power Reduction

The clock gating features moderate dynamic power reduction and is a well-known technique to optimize clock network dissipation. However, it shows almost zero effect on leakage power. Since it just controls the switching activity of sequential cells, and cannot reduce the current leakage phenomenon on gates which is basically process and temperature dependent. According to studies [10], the clock signal in digital design is responsible for 15%-45% of total consumptions. By applicating this technique, with the construction of activity pattern based clock tree, the total power can be reduced around 15%-20% [5].

Timing and Functional Verification

The timing is not critical in a clock gating design, since the additional logic does not introduce significant delay. No extra effort is demanded on functional verification, since the clock gating insertion algorithm does not modify the logical functionality and

¹¹Design For Test

¹²Process Design Kit

must be timing driven. Equivalence checking tools should be applied to warranty that the logical equivalence between the clock gating netlist and the RTL¹³ design.

Back-end Implementation

Although the power aware placement and routing algorithm are well explored and studied by the EDA industry, most of the back-end implementation tools present wirelength overhead problem and cell overlaps after clock gating insertion. The correct (not necessary optimal) placement and routing algorithm for this technique must feature clock gating aware clock tree construction, non-overlap insertion and zero skew clock routing to further reduce power and clock skew [20]. Since the sequential logic's activity pattern plays an important role in a gated clock tree construction, several EDA tools propose technique for register clustering based on activity and transition patterns to optimize the clock network before the final placement and routing stage.

Area

Because several enable signal (more precisely, muxes) can be replaced with one clock-gate cell, so, this technique can result in area reduction. Therefore, a bad strategy of clock gating insertion (sometimes, after the placement stage) could increase the area and provokes routing congestion and wirelength overhead problem. In general, this technique does not show an expressive impact in final silicon area.

Process Design Kit

The clock gating technique requires the integrated clock gating cells. Sometimes, the logical synthesizer must support the insertion algorithm, and in such case, standard cells are used to build customized clock gating logic. It is worth to remember that the clock gating aware DFT cells must be part of PDK to enable the circuit testability structure.

Applicability in Macrocells

It is not possible to apply clock gating technique in reducing macroblock's power consumption. If the dominant power consumption of the design derives from macrocells, this technique will not show any significant result.

Design For Test

The design team must pay a special attention on the DFT control logic while choosing the clock-gating style. The reliability of observability logic must be warranted after the insertion of clock gating. As we know, any kind of DFT techniques require

¹³Register Transfer Level

injection of a well known stimulus into DUT's¹⁴ inputs. These stimulus can be applied by a simulator, an ATE¹⁵ machine or another DFT logic.

We refer the ability of one circuit to apply stimulus to DUT as *controllability*, while, the ability to evaluate the output response is called *observability*. Depending on the chosen circuit, the gated clock net can no longer be controlled by the DFT circuit and consequently, reducing the fault coverage rate of circuit. The worst of all, the clock gating control logic (driven by the enable signal) could no longer be observable.

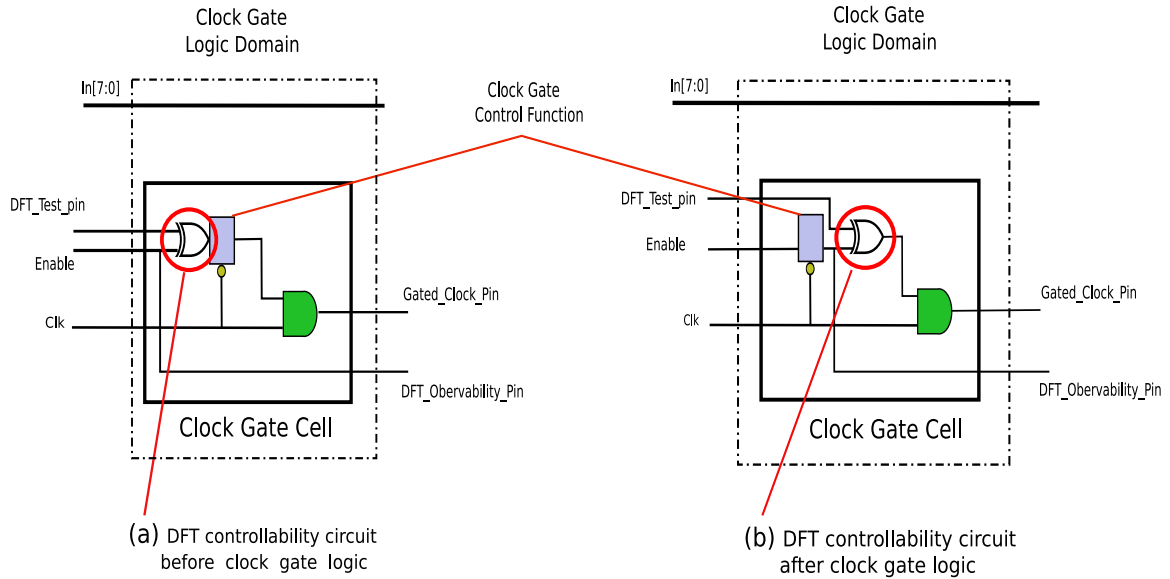


Figure 2.8: Clock Gating cell with DFT feature

A common approach to correct this potential problem is to increment the controllability and observability logic with clock gating cells, just like in Figure 2.8.

With this extra logic inside the clock gating cells (in style 2.8(a) or in style 2.8(b)), the DFT tools can generate coverage vector and perform the fault coverage simulation of the clock gating circuit. Once the design is sampled out from the foundry, the ATE machines are able to detect manufacturing faults according to the clock gating aware test vectors.

2.3.2 Operand Isolation

Although the clock gating is very effective in reducing the dynamic power dissipation of a digital circuit, it can only save power on sequential elements and clock circuit.

¹⁴Design Under Test

¹⁵Automated Test Equipment

Whenever a module performs an operation whose result is not used by the downstream circuit (i.e combinational circuit), power is being consumed for an otherwise redundant computation. **Operand Isolation** is a dynamic power optimization technique to reduce power dissipation of datapath blocks by selectively blocking the propagation of the switching activity through the circuit[19][9]. We can show this technique by the example of Figure 2.9

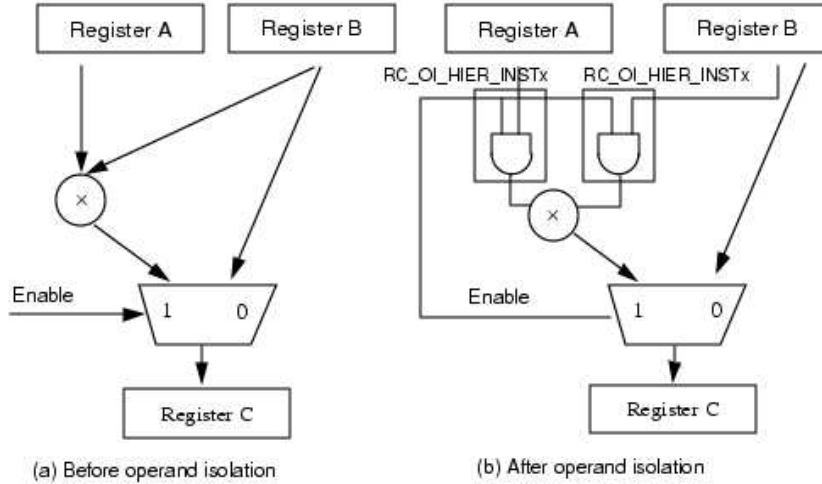


Figura 2.9: Operand Isolation Concept

The idea of operand isolation is to identify redundant operations and, using special isolation circuitry, prevent switching activity from propagating into a module whenever it is about to perform a redundant operation. Therefore, the transition activity of the internal nodes of the module and, to a certain extent, its transitive fanout is reduced significantly, resulting in lower power consumption[19].

From the design of Figure 2.9(a), register *C* uses the results of the multiplier when the *Enable* signal is activated. When *Enable* is turned down, register *C* uses the results from the register *B*, but the multiplier continues its computation. Because the multiplier dissipates most of the power, the total amount of power wasted during idle is quite significant. The **Clock Gating** cannot be used in this case, because the output of register *B* is always used by the multiplier.

Operand Isolation is a suitable solution in this case. It can shut-off (isolate) the functional unit (operand) when the results of the multiplier are not necessary. Figure 2.9(b), this technique inserts *AND* gate at the inputs of the multiplier and uses an extra enable logic to control the signal transitions. As a result, no dynamic power is dissipated when the result of the multiplier is not needed.

This is a pretty mature and automated technique in digital design. Normally,

after the lexical and syntactical analysis of input RTL design are performed, the logic synthesizer engine can identify some possible redundant computation blocks (such as adders and multipliers) and insert isolation logic.

Possible Impacts, Requirements and Constrains

Operand Isolation brings some overhead and requirements in the implementation flow and silicon area. We continue our discussion according in follower topics: power, area, DFT, timing, PDK¹⁶, logic synthesis, verification effort and back-end implementation.

Power reduction

According to the recent research results[19], the operand isolation can reduce up to 30% of power consumption in designs which presents intensive control signal and redundant computation logic. For those cases which present intensive operation of state machines or sequential logic, this technique cannot achieve expressive results.

Timing and Functional Verification Effort

There is major issues on timing and functional verification when using operand isolation circuits. The extra circuit does not add significant delays, after the insertion of isolation cells, the netlist can be verified by co-simulation and equivalence tools.

Logic Synthesis

The logic synthesizer must feature operand isolation algorithm to detect redundant logic and automatize the isolation circuit insertion. In the early day of this technique, the isolation cells only were manually inserted after a long analysis by RTL designers, this procedure is error prone and extremely time consuming.

Area

According to several theoretical and industrial results[9, 19], the operand isolation can slightly increase the final silicon area by adding several extra isolation cells. This penalty is worth if the power reduction is expressive. Normally this technique is suitable for the multimedia designs, which have massive redundant logic in the combinational logic and the memory cells are dominant factor in the final silicon area.

Process Design Kit

Operand Isolation algorithm modify just the RTL design and inserting the isolation circuits during logic synthesis. No special requirements are required for the

¹⁶Process Design Kit

PDK cells.

Applicability in Macrocells

As the case of Clock Gating, the Operand Isolation cannot show any advantaged in optimizing macrocells' power consumption, since they are pre-designed with fixed architecture and power consumption.

Design For Test

The post-manufacturing test structure can be easily implemented after the insertion of isolation logic, since the testability circuit just add observability and controllability circuits around the sequential cells whose logic functionality were not changed by the isolation algorithm.

2.3.3 Multiple Threshold Voltage

As described in the section 2.1.2, the leakage power is the power dissipated by the current leak in the transistor (subthreshold leakage, gate leakage, junction leakage). Leakage power is usually modeled in the library and characterized for several operation conditions (different temperature and supply voltages).

In the most advanced nodes ¹⁷ library specifies cell leakage power as a function of the input state which improve accuracy. In these cases, the leakage power is a function of pin switching activity.

EDA¹⁸ vendors have already automated the multiple V_{th} optimization. One of big concerns in earlier days was about the timing impact associated with this approach. From Figure 2.10(a), we see the relationship between delay and leakage for a generic 90nm process.

From Figure 2.10(b), we can see some representative curves for leakage *vs.* delay for a multiple V_{th} library.

As explained earlier, subthreshold leakage depends exponentially on threshold voltage (V_{th}). While, the timing delays has a much weaker dependence on threshold voltage[23]. Many libraries vendors or process manufacture offer two or three versions of cells (low V_{th} cells, standard V_{th} cells and high V_{th} cells) which provide opportunities to make trade-offs between the leakage power and the performance during the physical design.

Since the leakage power is specified by the library as the function of input states, the switching activities of design becomes important for an accurate leakage power

¹⁷ specially at 90nm and below

¹⁸ Electronic Design Automation

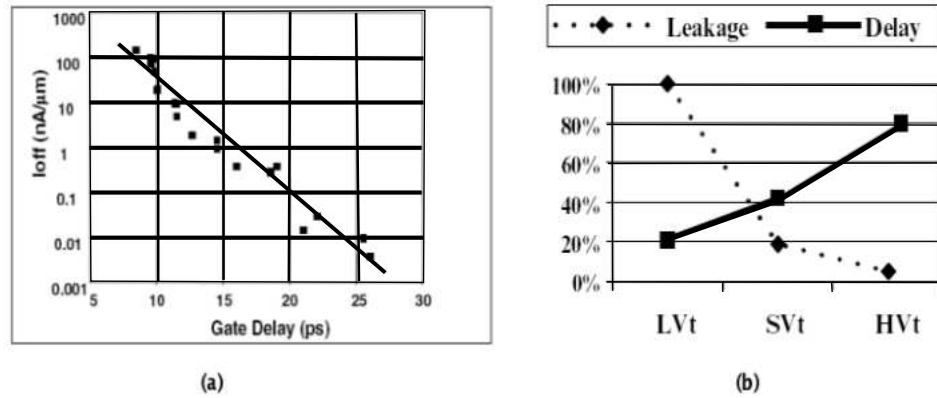


Figura 2.10: (a) Current leakage x Timing Delay (b) Library, Leakage and Delay Graph

optimization during the logical synthesis and the power optimization.

Possible Impacts, Requirements and Constraints

Multiple V_{th} technique cannot reduce dynamic power, since the modifications in threshold voltage just reduces the current leak in subthreshold regions. The total power consumption is substantially reduced just when the design remains in standby mode. When multiple threshold voltage libraries are used, a larger leakage power savings result, but the performance degradation and area impact must be very carefully analyzed. The key point of this technique is how the design team will identify the most leak regions and critical path to achieve a good trade-off between the power and timing constraints.

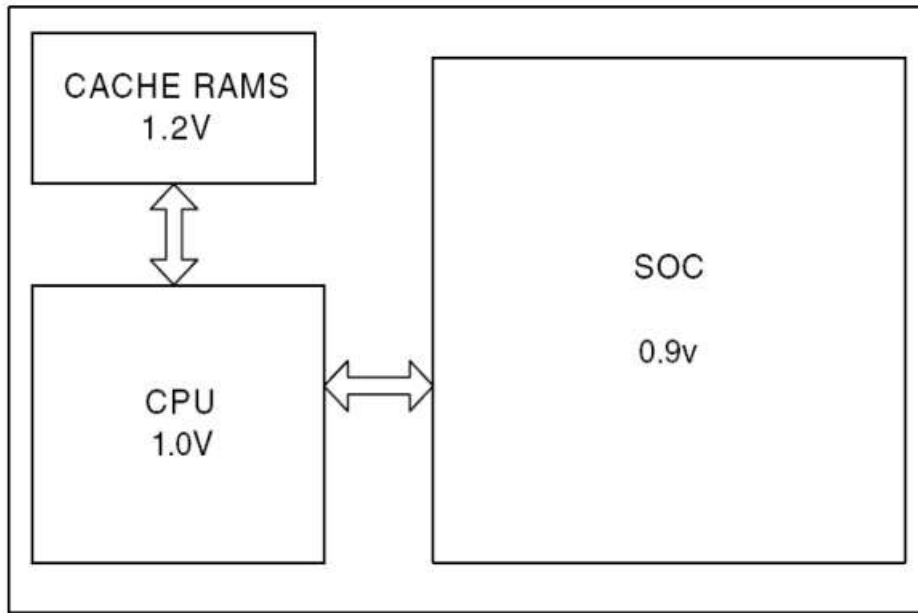
2.3.4 Multiple Supply Voltage

Multiple supply voltage optimization is one of most aggressive power saving techniques. In today's industrial environment, several supply voltages are present in the design. In spite of this approach have received great attention of design teams, it comes with several drawbacks and impacts in the physical designs and the verification process.

From the CMOS power dissipation of section 2.1.1, the dynamic power consumption is proportional to factor the V_{dd}^2 . Lowering V_{dd} value on some blocks help in reducing power significantly. Unfortunately, increases the propagation delay of the gates in the design.

In order to detail the multiple V_{dd} approach in SoC platform, we consider a simple example in Figure 2.11.

This simple example contains 1 CPU unit (1.0V domain), 1 cache RAM memory (1.2V) and the entire peripheral hardware (BUS, I/O ports, external memory controller,

Figura 2.11: Multiple V_{dd} Example

timers) was supplied by 0.9V voltage domain.

The cache $RAMs$ ¹⁹ are supplied with highest voltage because they are on the critical timing path. The CPU ²⁰ is run at a second voltage level as it determines system performance, but can be run at slightly lower voltage than the cache and still have the overall CPU subsystem performance determined by the cache speed. The rest of the SoC can run at a lower voltage without impacting the overall system performance. Mixing blocks at different VDD supplies adds some complexity to the design, and not only need to add I/O pins to supply the different power rails, but also need a more complex power grid and level shifters on signals running between blocks.

By extending the idea of several supply levels, a more complex power strategies can be implemented. We can provide different voltages to our processor, for example, depending on its workload, or we can provide different voltages to a RAM - a low voltage to maintain memory contents when the memory is not being accessed, and a higher voltage to supports reads and writes. Several multiple voltage strategies can organized according to the following list :

- Static Voltage Scaling (SVS): different blocks or subsystems are given different, fixed supply voltages.

¹⁹Random access memory

²⁰Central processing unit

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

Possible Impacts, Requirements and Constraints

This technique does not just bring advantages, usually it comes with following impacts in the final design. In the past, ad-hoc manual approaches to multiple V_{dd} design lacked a holistic view and increased design and implementation time. Several industrial results showed 2x productivity drop in the back-end phase with using the manual approach.

This productivity penalty was due to lack of tool functionality to support the scalability of implementation of the voltage islands, when several supply voltage exist in design. Multiple V_{dd} can impact the final design in the following metrics: scalability, area, PDK, timing, power grid, back-end implementation, DFT, power sign-off.

Scalability

The multiple V_{dd} design contains a number of problem with respect to designing and verifying the Voltage Islands. They are:

- Isolation gates for power switching or level shifters for voltage scaling, introduces verification challenges. Checks must run to verify proper isolation, proper connectivity to the right power domains, proper partitioning of the netlist, and correct behavior of the interface.
- Level shifters, which are standard cells operating with two voltage supplies, create a constraint for the layout implementation. They are buffers that translate the signal from one voltage swing to another, specially designed to offer a voltage scaling interfaces between different voltage domains.
- Always-on logic, resulting from the buffering of the control logic for retention or global nets in power down blocks, require a proper connection to their voltage supplies.

- Voltage islands and on-chip switches create a challenge for power distribution and limit alternatives of floorplanning and flexibility. More effort is required to connecting the power sources to the voltage domains.
- Communication between voltage islands may create logical paths spanning power domain, increasing the number of corners and modes, and the number of static timing analysis (STA) runs.

PDK (Cell library and Level Shifters)

Since the signals in the design will go between blocks that use different power rails, level shifters become mandatory. One special situation of usage of level shifter happens when two power domains present very close voltage levels. For example, in a 0.9V and 1.2V domains. The 0.9V signal driving a 1.2V gate will turn on both the NMOS and PMOS networks, causing crowbar currents. In addition, standard cell libraries are characterized to operate best with a clean, fast input that goes rail to rail. Failure to meet this requirement may result in signals exhibiting significant rise or fall-time degradation between the driver cell in one domain and the receiver cell in another voltage domain. This in turn can lead to timing closure problems and even excessive crowbar switching currents.

Timing Library and STA²¹

With a single supply for the entire chip, timing analysis can be done at a single performance point, since the libraries are characterized in a conformal way and the tools perform the analysis very straightforward.

With multiple blocks running at different voltages and with libraries that may not be characterized at the exact voltage in use, the timing analysis becomes much more complex.

From the simulation results between voltage variation and timing delays (Figure 2.12), we can see a performance changes

At the 90nm node and below, designers face much more challenges when using multiple power domains in projects. As the supply voltage changes from one domain to another, changed aggravately the signal integrity and the noise margins issues. Normally, we the power domain definitions must be include by the STA process. This implies more engineering effort and more timing cost.

Power Planning and Power Grids

²¹Static Timig Analysis

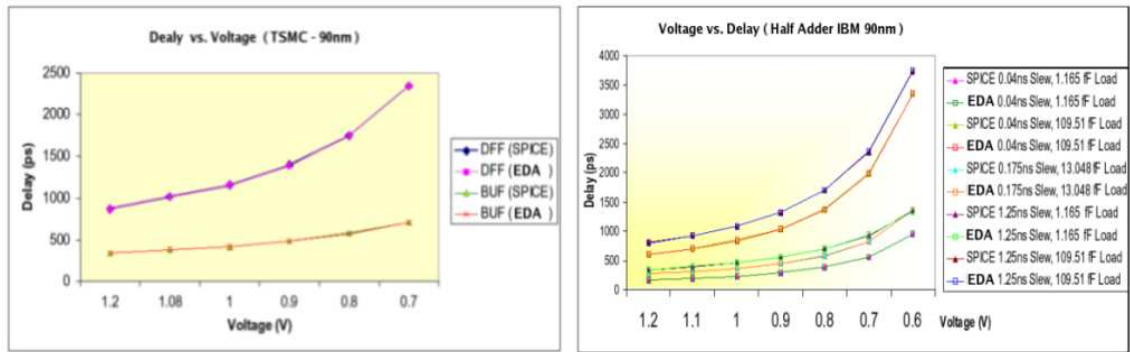


Figura 2.12: Voltage variation vs. Timing delay

Due to the existence of multiple power domains, the floorplanning is required to be more careful and detailed. The power grids become more complex, the power rails must be prepared to supply various voltage areas in the design. As each power domain employs a different power strategy, it is natural that different power rails routing must be performed inside each voltage area.

Minimizing the voltage drop across each of these power rails is key to meet the performance goal. Unfortunately, many of the techniques that we employ in a low power design provoke voltage drop and make noise problem worse. For example, when the design is operating at reduced voltage levels, the available margin for voltage drop is considerably reduced.

Placement and Routing

Multiple voltage designs present significant challenges for placement and routing. We separate in two topics to show their possible impacts.

1. Level shifter placement: In fact, level shifters do not affect the functionality of the design; from a logical perspective they are just buffers. For this reason, modern implementation tools can automatically insert level shifters where they are needed. No change to the RTL is required. Many tools now allow the designer to specify a level shifter placement strategy to place the low-to-high level shifters in the lower domain, the higher domain, or between them.
2. Level shifter routing: Leading commercial routers today are all low-power aware and will honor the power intent of 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.

Clock Tree Synthesis

The impact on the overall power consumption of the clock tree is significant. In the case of dealing with a multiple-voltage optimization, we have additional restrictions on manipulating the clock tree to meet both the power and the performance requirements.

In a single clock multiple-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 was modified constantly when changing the voltage domain, and this makes the skew management very difficult on the entire design.

Design for 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, yielding an overall lower power design that meets the needs of the system application. However, in most of post-manufacturing test, the design must be tested at nominal supply voltage, which makes the design to consume significantly more power during test than it would during normal operation. If the design has only been packaged, considering just the functional power dissipation, then the failures in the design and the package could happned during the test.

Power Sign-Off

Having completed the implementation of the multiple-voltage, it is necessary to verify the integrity of the power network. Two aspects of power network are critical: the voltage drop and the long term electromigraton effects. It is common, several type of EDA tools were used inside this task. Several reworks and interactions were required to fix all the violations and warranty the final design.

Capítulo 3

Related works

The reason on reducing power consumption differs from application to application. Many times, it depends the system requirements and the decision of the project manager to achieve the power targets (sacrificing area or performance).

In this chapter, we list several examples of low power SoC designs, showing their features in terms of power, area, speed and implementation cost.

3.1 NEC mobile phone system SoC

The NEC Corporation¹ (**Nippon Denki Kabushiki Gaisha**) is a Japanese multinational *IT* company, provides information technology and network solutions to business enterprises, communications services providers and governmental organizations.

The mobile phone **SoC** from the NEC is a cell phone chip, designed at 65nm process, and currently is used by the NEC cell phone family and others. His fundamental architecture was originated as early as 2003. Since then had undergone evolutions, enhancements, standard changings and process migration[16].

Figure 3.1 shows the architecture diagram of the entire SoC, including all the different modes of operation, peripheral hardwares and the baseband functionality.

This chip is an example of complexity and high level integration with a constrained power goal (in the wireless market, the battery life is a paramount).

There are two implementations of this SoC and their respective features are listed below (the *M1* and the *M2*):

- M1 presents 7 million CMOS gates, and was designed in a 90nm process. The CPU shows the maximum performance when it was running at a 250 MHz clock. The

¹<http://www.nec.co.jp>

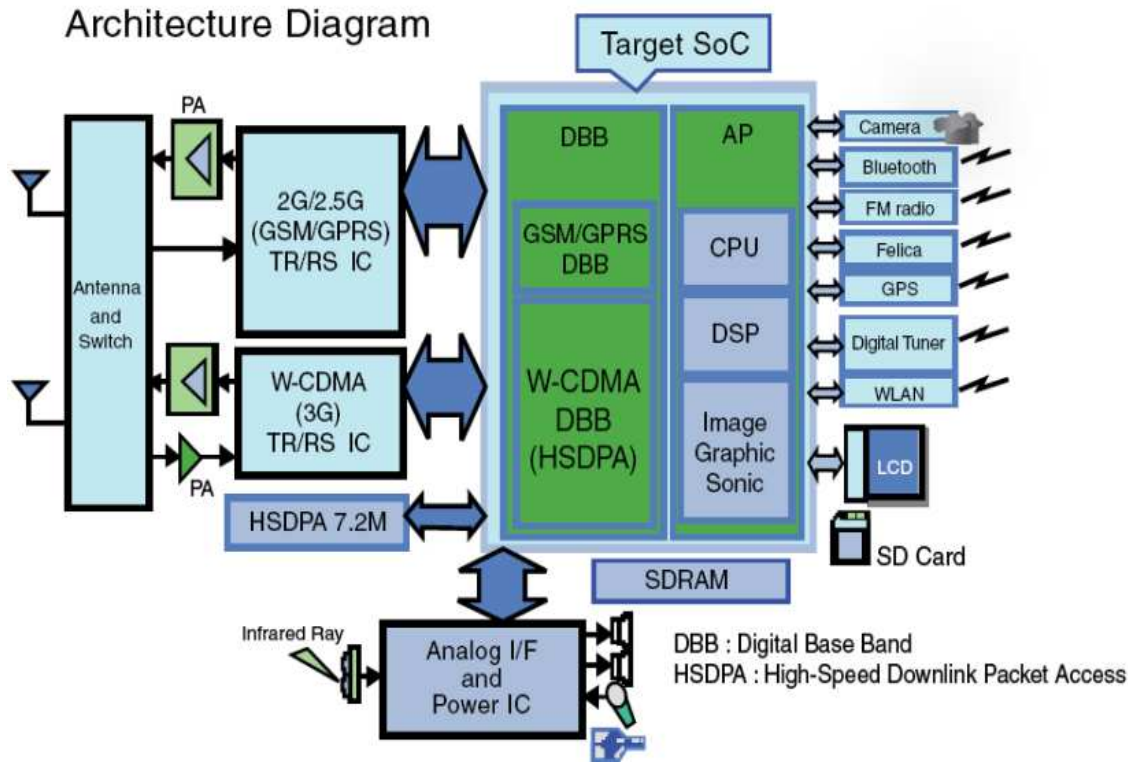


Figura 3.1: Architecture of NEC 65nm Cell Phone SoC

M1 SoC interfaces with a 8M bits external memory (SRAM²).

- M2 had twice the silicon area (15 million of CMOS gates) and was designed in a 65nm process. The CPU was running at a 500MHz clock (maximum performance) and interfaced a 12M bits external memory (SRAM).

Power results from [16] indicated that if the M1 had been implemented without any advanced power management, its power would have been completely unacceptable for both active and leakage power. However, by deploying advanced power reduction techniques including dynamic clock controls, multiple power domains with power shut-off, back bias and multiple V_{th} , the M2 delivered twice the performance with the same power specification as the M1 chip. These techniques reduced active power by more than 50% and leakage by more than 60% compared with the classical design methodology.

The NEC team did not reveal any details about the implementation and verification challenges. The total project time and the engineering effort were also unknown.

²Static Random Access Memory

3.2 Fujitsu low-power test chip

Fujitsu, Inc³ is a Japanese multinational computer hardware and IT⁴ services company. He focuses on providing IT-driven business solutions, products and services of personal computing, telecommunications and advanced microelectronics.

Fujitsu designed a complex chip, which has 940K instances, 11 power domains and 19 different power modes[16]. Taped-out successfully in June of 2007, validating both the power goals and the design flow, satisfying the objective of the project. The power savings achieved by this SoC include: 35% dynamic power reduction with standby power reduced by a factor of 100 times. Low-power techniques adopted in this design were:

- Multiple Threshold Voltage optimization (3 different V_{th} cell libraries).
- Clock Gating.
- Multiple Supply Voltages (MSV).
- Dynamic Voltage/Frequency Scaling (DVFS).
- Power Shut-Off (PSO).
- Adaptive Voltage Scaling (AVS).

Some statistics for the advanced low-power design showed that power reduction techniques with power intent file (details in section 4.2) produced excellent results in area which was translated into cost savings, performance preservation and superior engineering productivity. Some interesting result from this project can be summarized in Table 3.1:

This design emphasizes the importance of power intent during the total design flow. However, information associated with each techniques, as well as the purpose of the chip has been omitted by company.

3.2.1 NXP Low-Power SoC

The NXP⁵ is the name for the semiconductor company founded by the Philips⁶. This multinational corporation manufactures and markets chip set and contactless card for MIFARE, used by many major transaction systems all over the world.

³<http://www.fujitsu.com/global>

⁴Information Technology

⁵It stands for **N**ext **eX**Perience, <http://www.nxp.com>

⁶<http://www.philips.com/global>

Design Parameter	With Power Intent File	Without Power Intent File
Area Penalty	Varies widely depending on engineering expertise.	The area penalty, including all the low-power techniques, was less than 2%
Timing/Performance	Risk of performance impact	No significant impact on timing design or performance
Productivity	Months of additional engineering effort for manual implementation of low-power techniques, verification; high risk still	Design cycle was extended by only 2 to 4 weeks (mainly logic design and verification) to incorporate all the power management techniques.

Tabela 3.1: Fujitsu SoC Table

The NXP developed a complex SoC[16] that challenged the current architecture and implementation flow. This ARM⁷ based SoC was successfully taped-out in 2007 and his architecture diagram can be found in Figure 3.2. Some relevant characteristics and remarkable results are listed below:

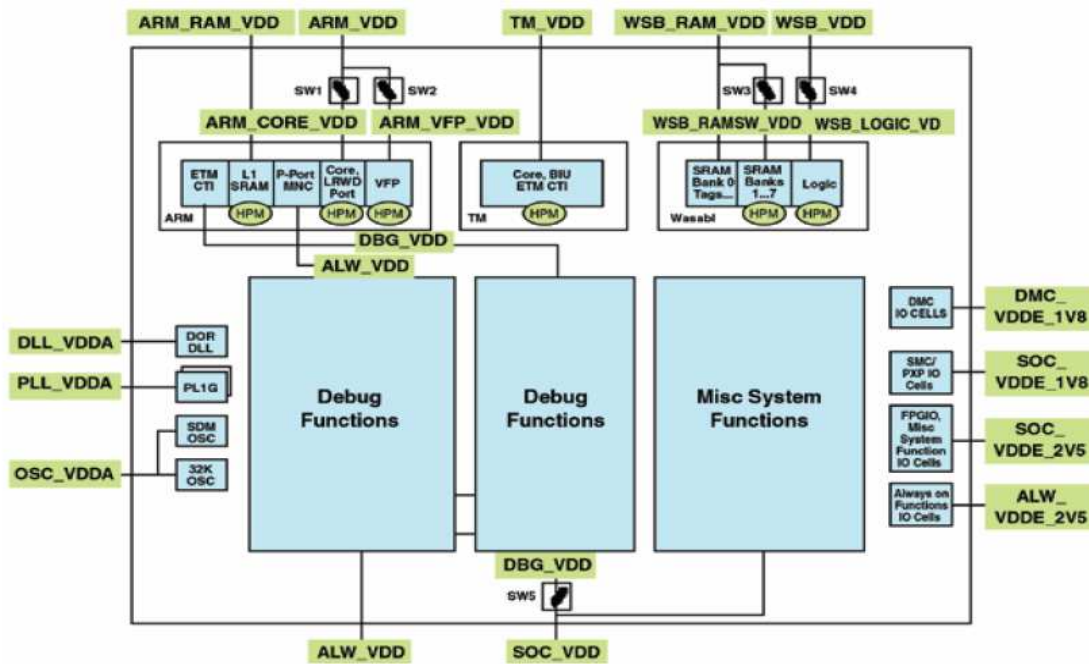


Figura 3.2: Architecture of NXP 65nm SoC

- There are 3 voltage-scalable logic sections, 3 on-chip switchable domains and 5

⁷Advanced RISC Machine

off-chip switchable domains.

- The 3 major power consumers (RISC⁸ CPU⁹, VLIW DSP¹⁰ and L2 Cache) are controlled by using DVFS¹¹.
- High-bandwidth expansion ports, enabling the platform to be extended with graphics or cellular modem subsystems.
- The die size of the chip is 42 mm^2 and was fabricated in a 65nm CMOS process.
- A 50% engineering effort was saved by implementing advanced power reduction techniques. Before the adoption of the power intent format (Figure 4.3), the NXP experienced, 2x productivity drop in the implementation phase.
- The power intent based simulation helped in detecting a critical bug: a time-out mechanism was being powered down in one particular mode, which could have caused deadlock conditions on the communication bus.

This SoC design demonstrates a scalable implementation of multiple V_{dd} technique. It has several voltage islands inside the same chip. All the tools of the flow understood the same power intent representation, with the highest possible level of abstraction, compensated by the throughput overhead, introduced by multiple power supplies. The level shifters, retention logic and on-chip switches¹² were logically inserted, verified and analyzed. The power modes and the operating conditions were managed during the synthesis.

Some detailed information like area impact, performance degradation and verification efforts were briefly described or omitted by the company.

⁸Reduced Instruction Set Computer

⁹Central Processing Unit

¹⁰Very Long Instruction Word Digital Signal Processor

¹¹Dynamic Voltage and Frequency Scaling

¹²Retention logics and switches are used in the power gating optimization. They are not part of the work scope.

Capítulo 4

Low Power Design Flow

In this chapter, we describe the classical digital design flow and highlighting some additional steps that are fundamental in a low power design. Our goal is to clarify the differences between the two flows, showing possible challenges when adopting a low-power design flow.

4.1 Standard cell based design flow

The classical digital design uses two different methodologies: the full-custom and the semi-custom. The semi-custom allowed the expansion of CMOS technology in the modern electronic industry. This methodology is a sequence of well defined steps based on the principle of design abstraction. The 5 abstraction levels are: system, module, gate, circuit and device.

This design approach was enabled by the emergence of EDA¹ tools that support tasks like: simulation at various complexity levels, design verification, automatic layout generation and design synthesis. To avoid redesign and rechecking, the most used cells (such as basic gates, arithmetic modules and memory cells) have been assembled in cell libraries.

The recent IP² reuse approach also contributes significantly to reinforce this technique, reducing the silicon re-spin risk, the development cycle/cost and the time-to-market pressure.

Inside the Cell/IP libraries, vendors usually provide not only layout but also a complete documentation and characterization of the Cells/IPs, which include integration guides, simulation models (for timing and power estimations/verification) and a set of manufacturing rules.

¹Electronic Design Automation

²Intellectual property

The standard cell design flow can be summarized in Figure 4.1. For each one of these steps, we present a brief description.

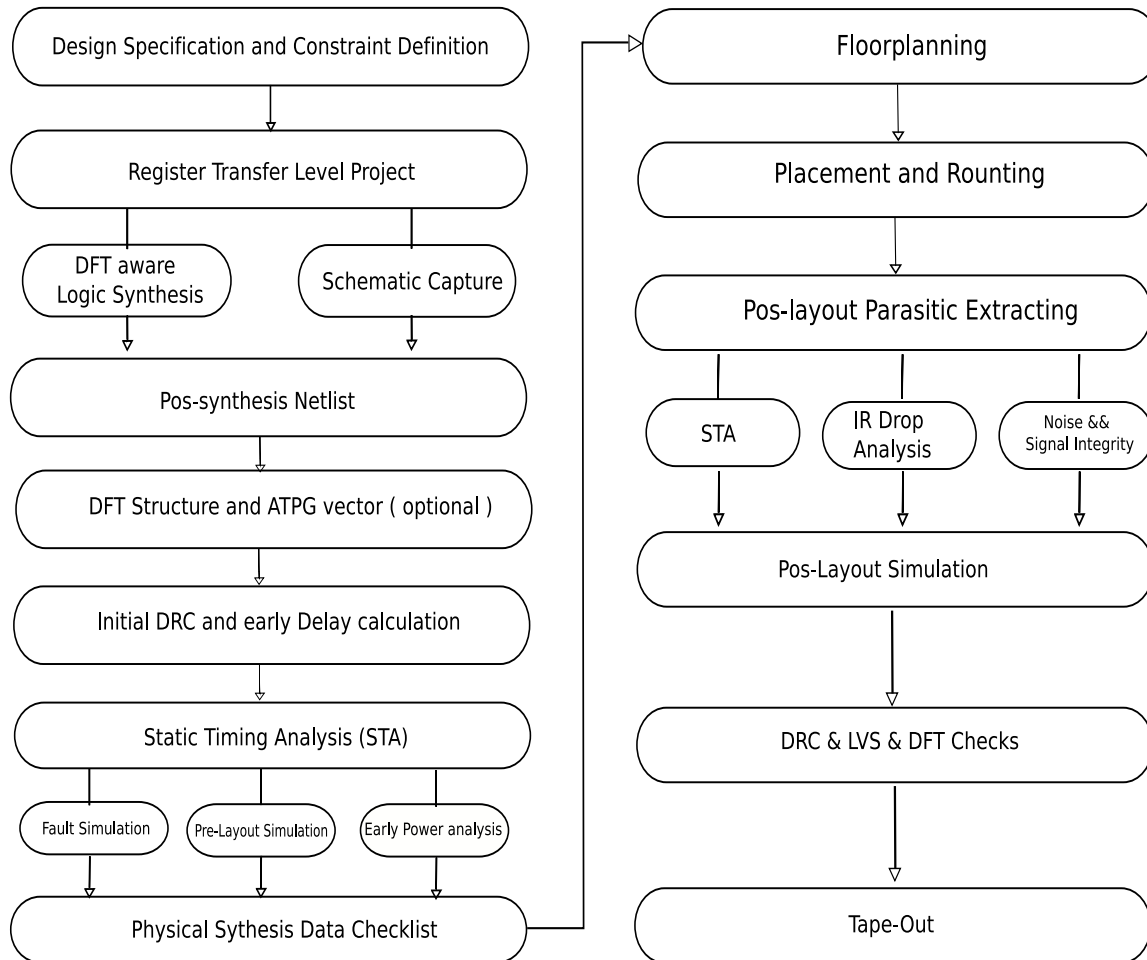


Figura 4.1: Cell Based Design Flow

Design specification and constraint definition

This is the entry point of any ASIC design. A variety of methods is used to extract design specification and constraint. These methods include modeling languages, schematic development and techniques of block diagrams. Recent ESL³ design (followed by the high-level language synthesis) emerged as the next entry point of design standard.

Register Transfer Level design

The RTL design is used to capture the structure description of the designs.

³Electronic System Level

The main design languages adopted in this step are the so-called hardware description languages. Several important HDLs are Verilog, System Verilog, VHDL and SystemC. This step models a Finite State Machine which is logically equivalent to the design spec and then performs a rigorous functional verification process.

Design-for-Test aware Logic Synthesis

In this step, the logic synthesizer translates modules described in HDL into a functionally equivalent netlist, which is a representation in form of gates and wires. The netlist of reusable cells or macrocells can then be inserted to complete the full design functionality. During this step, the structure that enables the design testability is produced, inserted and ready for layout generation.

Post-Synthesis simulation with DFT⁴ vector

It this step, the design is checked for functional equivalence. Performance analysis is done, based on estimated parasitic and manufacturing parameters. If some functional bugs are found, extra iteration over former steps are necessary. Post-manufacturing test vector and testability coverage rate are also defined in this step.

Floorplanning

Based on the estimated size from the netlist description, the floorplanning maps the logical description of the netlist into a physical description. Several tasks are done here: chip size estimation, macrocells pre-placement, pin assignment, I/O⁵ definition, power rail planning.

Placement

Location of the standard cells are defined to a particular position. Space is set aside for interconnections between standard cells.

Routing

At this step, interconnections of cells and macroblocks are wired and the clock tree is built respecting the timing and manufacturing constraints.

Post-Layout Parasitic Extraction

After the model of the chip is generated based on the correct device sizes and wire lengths, the extraction tools estimate the layout parasitic capacitance and wire resistance with enough accuracy, which is important to estimate the path delay of the entire design.

⁴Design For Test

⁵input and output pins location are defined

IR-drop, Noise and Signal Integrity Analysis

Several power-related effects (section 2.1.3) can seriously damage the chip performance and functionality after manufacturing. The IR-drop effect represents voltage loss in some spot of die and might result in timing problem. Signal integrity and noise problems in ICs may have many drastic consequences, involving die failure in field, performance penalty, loss of silicon yields.

Special EDA tools must be incorporated to the design flow to analyze these effects and results should be carefully considered in order to decide whether any re-design is required. Modern signal integrity tools perform all these steps automatically, producing reports that give the designers a clean list of problems that need to be fixed[16].

Post-Layout Simulation and Sign-Off Verification

The functionality and performance of the chip is verified in the presence of the layout parasitic. The final sign-off task should be performed using the manufacturing rules. Very often, at this step, designers detect some errors and return to the initial steps of the project, interacting over the design until it meets the goals and constraints.

Tape-Out

Once the design is found to meet all design goals and functions, a binary file is generated, containing all the essential information for mask generation. This file is sent out to an ASIC⁶ vendor or a foundry. This is an important moment in the life of a chip.

4.2 Low Power Design Flow

Digital designs become severally complex while process geometries, operation voltages and schedules are contracted in advanced nodes. As result, the predictability across the design flow becomes paramount and each process node emerges with a new set of challenges. At 130nm process, issues like timing closure, area and signal integrity are considered as primary concerns. In the 90nm process, power budget management becomes very important on top of previous factors. The 65nm process node presents additional challenges in manufacturing variations such as, CMP⁷, lithography distortion and thermal profiling. At 45nm and below, process nodes not only present all the above mentioned challenges but also have need to consider dopant fluctuations, oxide thickness variations and other issues. Consequently, the engineering effort to meet the overall design closure is becoming more and more costly[16].

⁶Application Specific Integrated Circuit

⁷Chemical Mechanical Polishing

We showed in section 4.1, the standard cell design flow (with cell reuse methodology) demonstrates a high scalability and a great ability in reducing the design cycle and the silicon risk. With the increased complexity of the process nodes, the power-related effects (section 2.1.3) arise as the principal evidences that the conventional approach is not enough to meet today's project constraint.

From another angle, the semi-custom flow was developed from a pre-designed cell library, which contains a fixed timing and power information (PDK⁸). This fact does reduce the capability of designer in managing performance and changing the power characteristic of any semi-custom design.

To achieve the power goal and still keeping the advantages of cell based approach, successive power optimization techniques must be applied in each design stage and abstraction hierarchy.

From the description of section 2.3, we know that power reduction techniques always induce a new set of impacts on the design. It is fundamental to have a new methodology that evidences potential impacts and reduces silicon risks. A typical low power design flow can be summarized in the Figure 4.2. In which, we highlight in colors the main difference between the low power flow when compared with the classical flow.

The colored part are additional steps required to implement a low power design. For each one of them, we present a brief description to clarify its importance:

System Requirement and Project Specification

Just like in the conventional flow, the specifications like speed, area, DFT⁹, mass production volume and the power constraints need to be defined.

Power Architecture, Power Intent Representation and Power Technique selection

Adoption of power intent representation and its validation in the system level context must be done after the spec definition. The low power design flow needs to specify the desired power architecture and the power intent representation for each major step and task.

The conventional design flow has failed to address the additional considerations for incorporating the advanced low-power techniques. Consequently, design teams often resorted to methodologies that were ad-hoc or highly inflexible.

According to the old methodologies, it was required that designers to manually model the impact of power techniques during the simulation and provided the same power

⁸Process design kit

⁹Design For Test

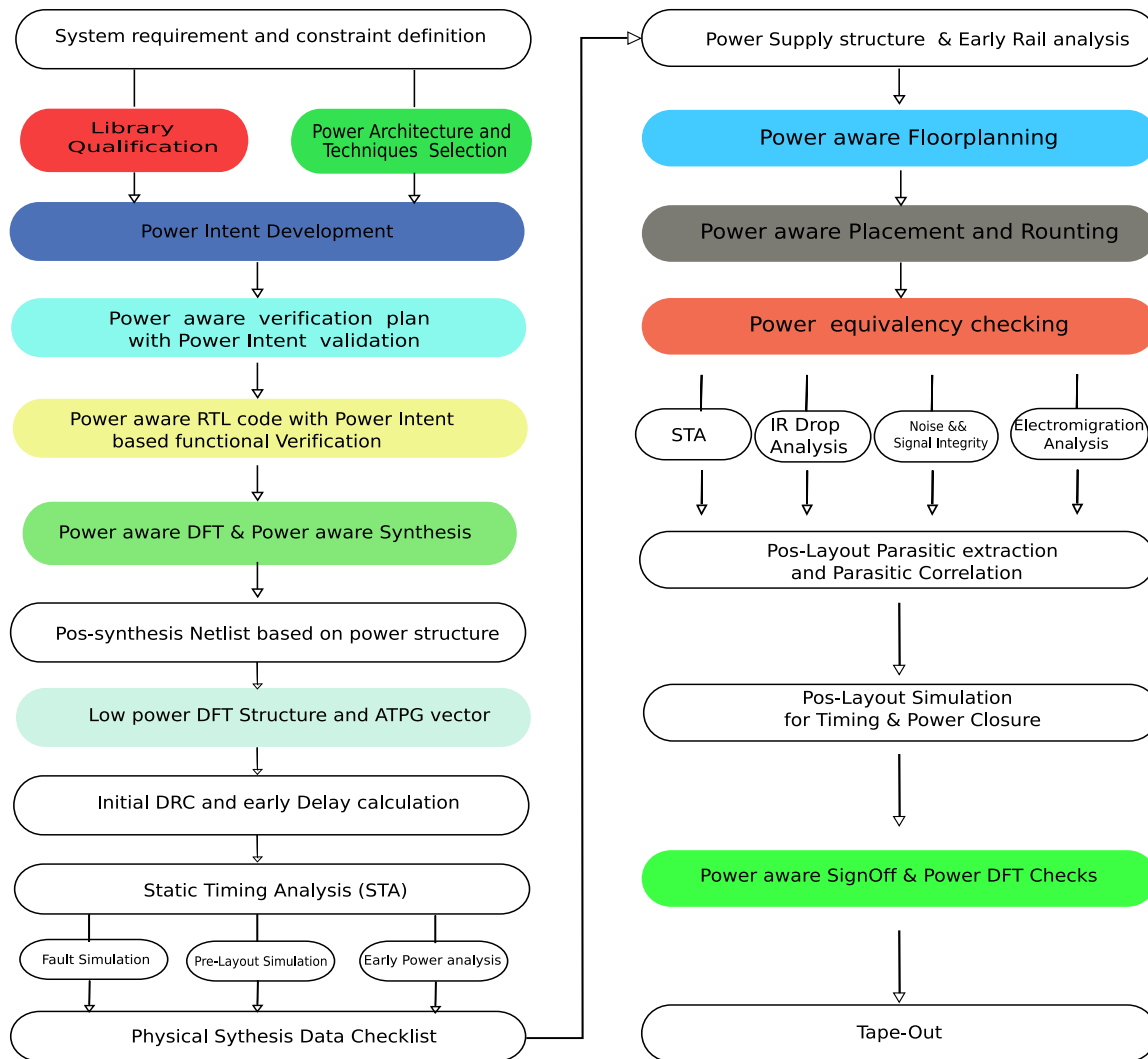


Figura 4.2: Low Power Design Flow

information at each step: one for synthesis, one for placement, one for verification and yet another for equivalence checking. Even after all that manual work, the old flow had no warranty of consistency.

This means that there was no way to assure that what was verified matched with what was implemented. The results were lower productivity, delay in time-to-market, increased silicon failure risks and inferior trade-offs among performance, timing and power.

Now, with the concept of power intent, as shown in Figure 4.3, all the design stages are integrated and truly reflect the correct architecture and power techniques.

This means, several steps must be connected by the power intent: functional verification, logical synthesis, DFT, Sign-Off, power estimation and equivalence check.

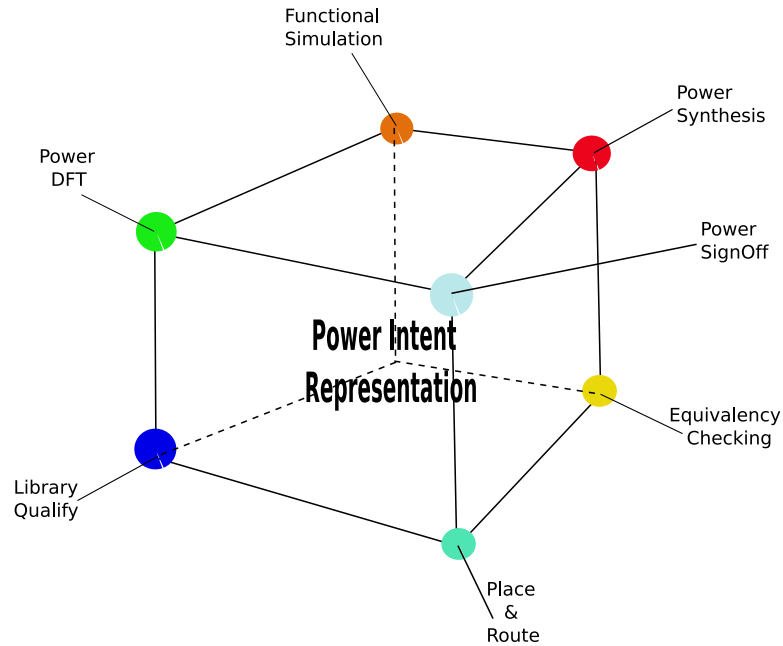


Figura 4.3: Power Intent Representation in the Low Power Design

The major goal is to reduce the effort of the design team in implementing the advanced power techniques. To formalize this objective, recently, some industry coalitions have developed two official standards¹⁰.

Library Qualification

This step is responsible to examine the PDK¹¹ library capability to implement the chosen low power techniques. This task is fundamental to ensure that the data is complete and valid before the project actually begins. Normally, this qualification process is a recommended step for all kinds of designs, and is not limited just in low power project. Some times, to achieve more consistent and reliable to library quality, some trial run of RTL to GDSII flow is recommended and can reveal some unexpected issues.

Power Aware Verification plan and Power Intent Validation

This step integrates the power intent models to the functional verification plan. It is the first and the biggest challenge in the front-end flow of a low power design. This integration must cover the functionality of the entire design and meets the power spec without affecting the original functions. The most common bug happens in some advanced techniques like PSO and DVFS, which challenge the skill of designers in developing the

¹⁰The CPF, Common Power Format. The UPF, Unified Power Format

¹¹Process Design Kit

test vector to capture the worst power case that reveals the potential failures.

Power Aware RTL¹² Design

The RTL code should be developed and verified with the power intent file. The worst power test vector should be applied in the entire design to estimate the total power budget at a early stage. It is very common for interactions to happen between the verification team and the architecture team during this stage.

Several verification technique like hardware software co-simulation, PSL¹³, assertion based verification are intensively used to help the development.

Power Aware Synthesis with DFT¹⁴ Structure and ATPG¹⁵ Vector

Power techniques complicate the testability of chips and the DFT logic insertion. To test a low power design, there are two key issues. First, the design must be testable even in a high power condition.

In the ATE¹⁶ tester, the power consumption can be very larger then the operational power consumption (efficient test patterns cause switchings of very high percentage of the logic at a given time), and some chips can melt on the tester unless different blocks are shut down at different times[16].

Hence, for some advanced techniques like PSO¹⁷, the DFT structure must limiting the switch activity to test the advanced power logics (Level Shifters, Switch On/Off Cells and State Retention Cells).

Current EDA¹⁸ vendors combine the design of the DFT structures with the advanced test pattern generation. To reduce power consumption during manufacturing test, the power aware DFT structure can be controlled by some special logics in order to select the test path. Usually these control circuits are combined with power domain aware ATPG¹⁹ to test advanced power techniques and reduce the power consumption during the ATE test.

Power Aware Floorplanning

The floorplaning of the classical flow was an important task to prepare the design for timing closure and full chip routing. In the case of the low power design, beside

¹²**Register Transfer Level**

¹³Property Specification Language

¹⁴**Design For Test**

¹⁵**Automatic Test Pattern Generation**

¹⁶Automated Test Equipment

¹⁷Power Shut Off

¹⁸Electronic Design Automation

¹⁹Automatic Test Pattern Generation

these two factors, the floorplanner should also understand the power domains and create physical boundary for each domain.

Power Aware Placement and Routing

The placement tool should recognize the low power cells and place them into an appropriate power domain. The router should consider the placed cells and power domains boundaries to perform the interconnections, without introducing violations in timing or power.

Power Equivalence Check

Formal verification or equivalence check tools are heavily used throughout during the low power flow. Formal verification of low power designs encompasses two elements: the low power verification and the logical equivalence.

For the low power verification, it must ensure that the design is electrically correct from a low-power perspective and verifies if the physical implementation are robust enough. The additional low power structure increases the complexity of the equivalence checking, because several low power cells have been added to the netlist after the logic synthesis. The tools must prove that the synthesis engine has inserted these cells correctly and the netlist is logically equivalent to the RTL and meets the power intent.

Capítulo 5

Case study results

Modern SoC designs not only present a complex software-hardware interaction, but also group numerous IP¹ cores (million of gates) into a single die. This makes the power consumption a big concern to design teams, as well as one of the *hottest* challenges. In next, we list some of these challenges during the power optimization of SoC design:

- Aggressive power requirement and long term reliability.
- Balance power, timing and area goal, which are overlapped or potentially conflicting.
- Lack in consistent description of low power requirement, that could be used to guide the implementation (connecting the RTL description with the physical design).
- Complexity in integrating existent back-end tools (RTL-to-GDS), which are disjointed and lack of power intent based verification methodology.
- Meet low power library qualification criteria. (Several essential IPs required by power techniques were not present in the PDK²)

From section 2.3, we studied several low power techniques that can be applied at different phase of the project. Some techniques insert extra circuit into the original design, others combine process libraries with variate supply/threshold voltages.

In this chapter, we apply such techniques in a industrial SoC design to understand how effective they can be in reducing power consumption. There were two possible candidates which were made by industrial team and were available in the public domain. The first one was the *OpenSparc-T1* project, a multiple-core, 64-bit multiprocessor based

¹Intellectual Property

²Process Design Kits

SoC, copyrighted from the *Open Sparc*³ project (Sun Microsystems). The second option was the Leon3 multiple-processor platform from the *Gaisler Research*⁴.

The Sparc T1 did not provide enough functional verification capability for system level simulation when this project got started. Moreover, this platform is target for data center servers and cryptographic application, which are out of domain of the project: multimedia application for embedded system. Our second option was the Leon3 platform.

Along this chapter, several characteristics of the Leon3 platform are presented, and then we estimate the power consumption of the Leon3, showing how the power reduction techniques can be effective.

5.1 Leon3 Multiple Processor Platform

The Leon3 is a synthesizable SoC platform from the Gaisler Research⁵. The central processing unit is a 32-bit processor, compliance with the Sparc V8 architecture. The *IP Cores* were built in VHDL language, and they are highly configurable. The project's full source code is available under the GNU GPL license, allowing free and unlimited use for research and education. The CPU core is also available under a low-cost commercial license, allowing it to be used in any commercial application. The processor shows the following features:

- Sparc V8 instruction set with V8e extensions.
- Advanced 7-stage pipeline.
- Hardware multiply, divide and MAC⁶ units.
- High-performance, fully pipelined IEEE-754 FPU⁷.
- Separate instruction and data cache (Harvard Architecture) with snooping.
- Configurable caches: 1-4 ways, 1-256 kbytes/way. Random or LRU⁸ replacement.
- Local instruction and data scratch pad RAM, 1 - 512 Kbytes.
- Sparc Reference MMU (SRMMU) with configurable TLB⁹.

³<http://www.opensparc.net>

⁴<http://www.gaisler.com>

⁵<http://www.gaisler.com>

⁶Multiplier-Accumulator

⁷Floating Point Unit

⁸Least Recently Used

⁹Translation Lookaside Buffer

- AMBA-2.0 AHB bus interface.
- Advanced on-chip debug support with instruction and data trace buffer.
- Symmetric Multi-processor support (SMP).
- Robust and fully synchronous single-edge clock design.
- Extensively configurable.
- Large range of software tools: compilers, kernels, simulators and debug monitors.
- High Performance: 1.4 DMIPS/MHz, 1.8 CoreMark/MHz (with gcc-4.1.2).

The entire SoC platform is distributed as part of the GRLIB IP Library¹⁰, thus allowing integration into a complex SoC design. The GRLIB also includes a configurable LEON3 multiple-processor design, with up to 4 CPU's and a large range of peripheral blocks. One of possible configurations of the Leon3 SoC is shown in Figure 5.1:

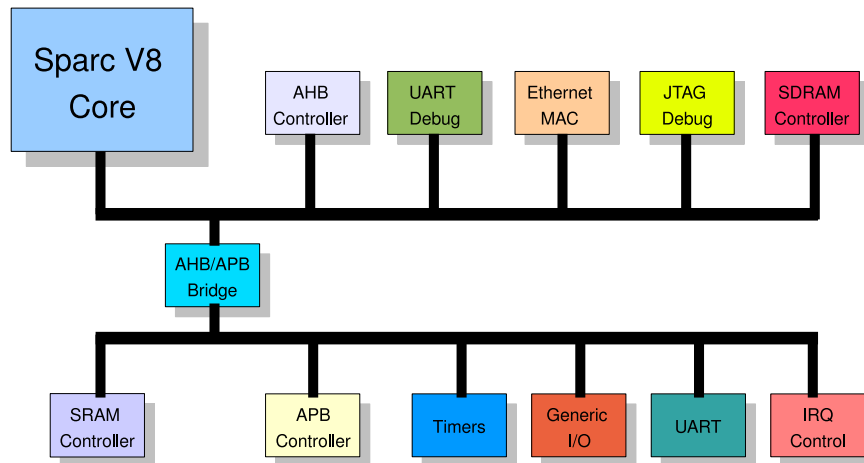


Figura 5.1: Leon3 SoC example

As shown, besides the Sparc V8 Core, it is possible to integrate several components/peripherals through AMBA¹¹ interface. In this example, the AMBA bus interconnects IPs through the AHB¹² and APB¹³ interfaces. The AHB is a single clock-edge protocol interface, presenting several bus masters, with features like burst transfers, pipelined operations, large bus-widths (32/64/128 bit).

¹⁰Gaisler Research Intellectual Property Library

¹¹Advanced Microcontroller Bus Architecture

¹²Advanced High-Performance Bus

¹³Advanced Peripheral Bus

A simple transaction on the AHB consists of an address phase and a subsequent data phase (without wait states, only two bus-cycles) and is used mainly to connect high-performance components. In our example, it connects the SDRAM¹⁴ Controller, the UART¹⁵ Debug Interface, the JTAG¹⁶ Debug Interface and the Ethernet MAC¹⁷.

The APB is designed for low bandwidth control accesses, like register interfaces on system peripherals. This bus has an address and data phase similar to the AHB, but in a much smaller complexity (for example no bursts). In our example, it is used to connect SRAM¹⁸ controller, the system timers, the generic input/out interface, the UART communication interface and the IRQ¹⁹ controller.

Given the Leon3 processor is compliance with the Sparc V8 instruction set, compilers and kernels for Sparc V8 can be used with Leon3. To simplify software development, the Gaisler Research provides BCC, a free C/C++ cross-compiler system based on gcc and sparc V8 C-library. The BCC includes a small run-time, with interrupt support and Pthreads library.

Linux support for Leon3 is provided through a special version of the SnapGear Embedded Linux distribution. The SnapGear Linux is a full source package, containing kernel, libraries and application code for rapid development of embedded Linux systems. The LEON3 port of SnapGear supports both MMU and non-MMU configurations, as well as the V8 Mul/Div instructions and floating-point unit (FPU). A single cross-compilation tool-chain is provided and is capable of compiling the kernel and applications.

The full fledged software tool-chains of Leon3 was fundamental in helping the development of the simulation environment for the power estimation.

5.2 The MPEG-2 Video Decoder

Any video information is a sequence of pictures and each picture is represented by an array of pixels. An uncompressed video presents a huge data rate for user-level applications and are becoming a large workload to a CPU²⁰ and the associated communication system. To reduce that, several video compression methods have been researched.

The MPEG standard defines an encoding and compression system for digital multimedia content. The MPEG-2 standard, known as ISO/IEC 13818, extends the basic MPEG system and provides compression support for TV quality transmission of digital

¹⁴Synchronous Dynamic Random Access Memory

¹⁵Universal asynchronous Receiver/Transmitter

¹⁶Joint Test Action Group

¹⁷Medium Access Control

¹⁸Static Random Access Memory

¹⁹Interrupt Request

²⁰Central of Processing Unit

video[14].

The video section of MPEG-2 defines a series of standards for video compression, whose algorithm achieves a very high compression rate by exploiting redundancy in the video information. It removes both the temporal and spatial redundancy present in a motion video. In Figure 5.2, we show this two types of redundancy.

The temporal redundancy arises when successive frames of the video display images of the same scene. In this case, very frequently, the content of the scenes could remains fixed or changed slightly between two successive frames. The spatial redundancy occurs because parts of the picture are often replicated (with minor changes) within a single frame of video.

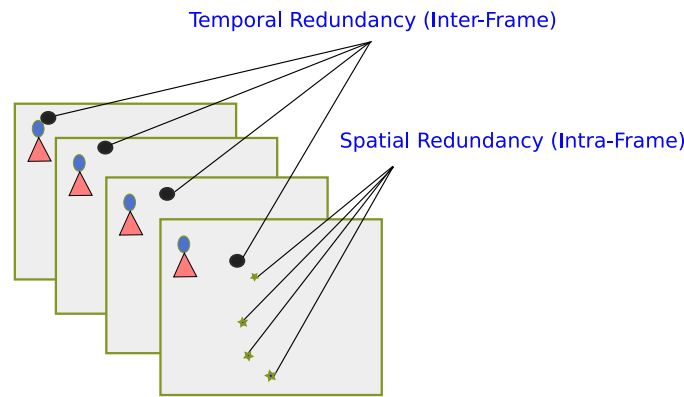


Figura 5.2: Temporal and Spatial Redundancy of Frame Pictures

According to the MPEG standard[14][13], there are three types of pictures in a video stream:

- Intra Pictures (I-Pictures): pictures are encoded using only information present in same picture and provides potential random access points into the compressed video data. It uses only transform coding and provides moderate compression.
- Predicted Pictures (P-Pictures): pictures are encoded based on the nearest previous *Intra* or *Predicted* pictures. This technique is called forward prediction and is just like *I*-pictures, *P*-pictures serve as a prediction reference for *Bidirectional* pictures and future *Predicted* pictures. Moreover, *Predicted* pictures use motion compensation to provide more compression than *Intra* pictures.
- Bidirectional Pictures (B-Pictures): in this case, the past and future pictures are used as references. This technique also be called bidirectional prediction, which provides the most compression quality and requires the largest computation time.

In Figure 5.3, we show the fundamental steps of the MPEG-2 compression algorithm: Discrete Cosine Transform (DCT), Signal Quantization and Run-Length Encoding.

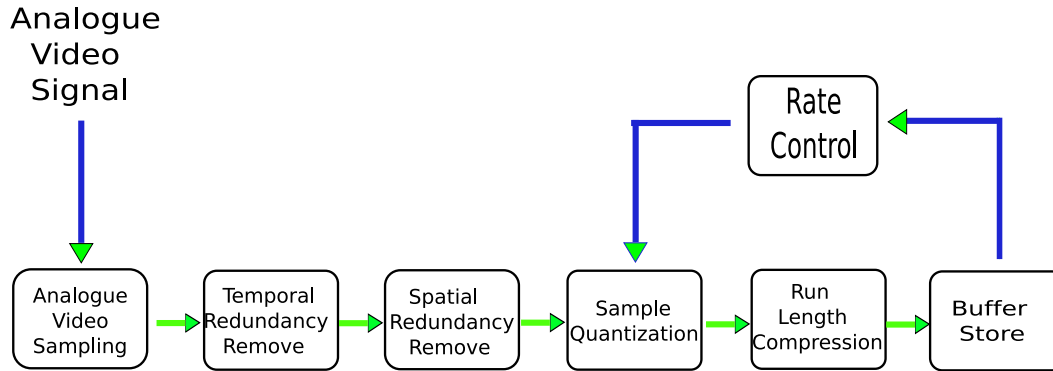


Figura 5.3: Basic Operations of MPEG-2 Encoder

As shown in Figure 5.3, the analogue video sampling is responsible for the raw data acquisition, followed by the discrete cosine transform and the redundancy elimination algorithm. The reduced redundancy signals are then quantized and encoded into a new compressed video format according to the Run-length/Huffman coding algorithm[14]. In Figure 5.4, we conceptually show the decoding and encoding process.

The decoding process can be thought as a reverse codification procedure. The first stage of the decoder is responsible for reconstructing the encoded data resulted from Huffman and Run-Length algorithm. Next, the motion vectors are parsed from the data stream (uncompressed from Run-length/Huffman coding) and be fed into the motion compensator. Also, from the data stream, the quantized DCT coefficients are extracted and are fed into the inverse quantizer to accomplish the data dequantization. After this, the IDCT²¹ transforms the requantized data back into the spatial domain, reconstructing the video data.

5.3 Platform Configuration (Hardware/Software)

In the section 2.2, we showed the CMOS power consumption is strongly dependent on the gate switching activity, the interconnection and the supply voltage. As described, the switching activity is the main component of power consumption.

Based on the understanding of the MPEG-2 decoding (high performance signal processing with dynamic and scene dependent behavior), an ISO/IEC DIS 13818-2 compliant implementation from **MediaBench** were selected to be our reference application[7].

²¹Inverse Discrete Cosine Transform

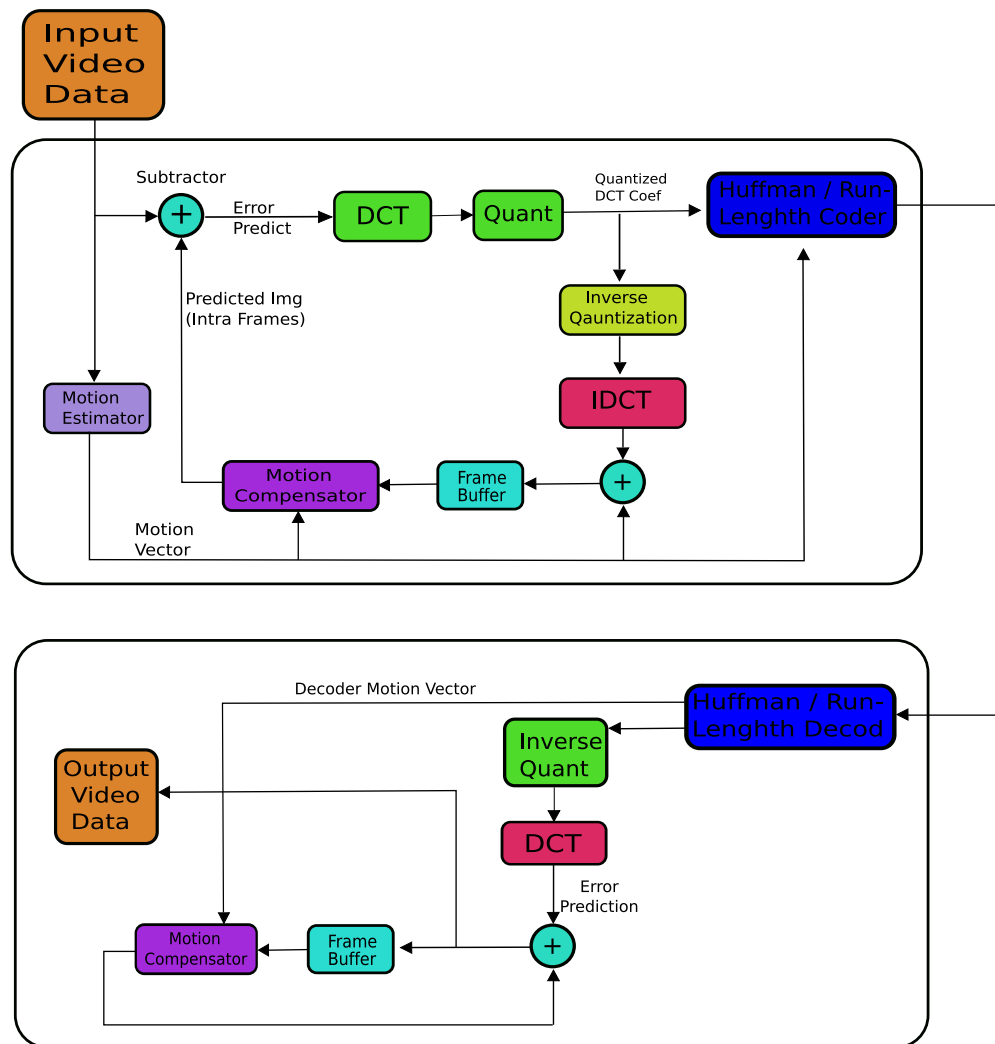


Figura 5.4: MPEG-2 Video Encoder/Decoder Operations

When the video decoding is performed on the Leon3 platform, we can capture the hardware switching activity and estimate the average power consumption (baseline), which is an important criteria to show the efficiency of each optimization technique.

In order to run the MPEG-2 decoder on the Leon3 platform, we did several software and hardware modifications. Some relevant changes are described in the following sequence: Leon3 Configuration - Figure 5.5 (**Sparc V8 Core, External SRAM and SDRAM controller, Timers, Generic I/O ports, UART, IRQ** and Memory Simulation Model), the modifications of MPEG-2 decoder and finally the extraction of switching activity.

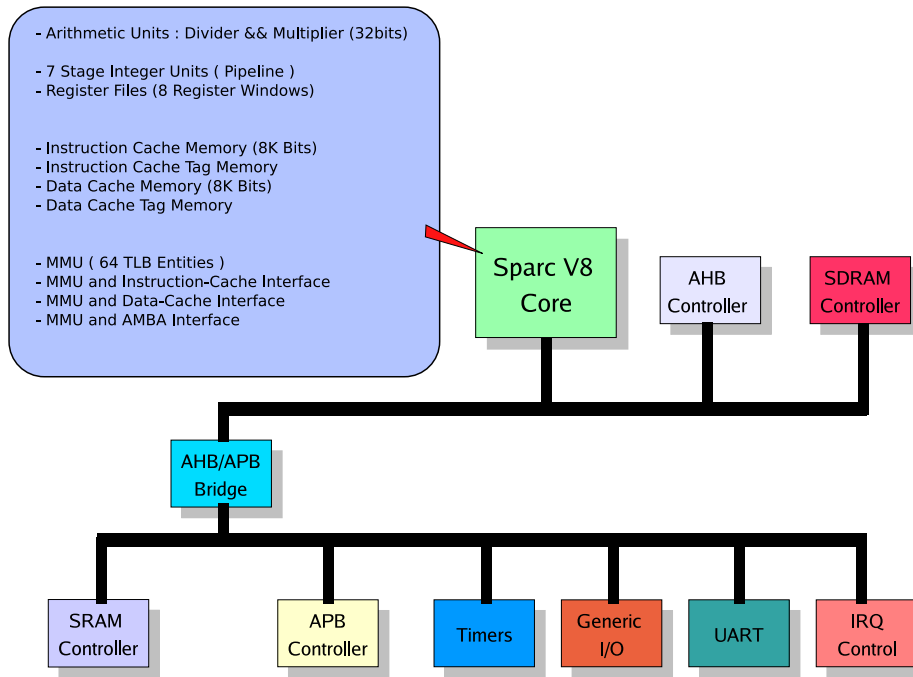


Figura 5.5: Leon3 System for Power Estimation

5.3.1 Sparc V8 Core

As mentioned in section 5.1, the core is a 32 bits processing unit, compliance with the Sparc V8 architecture. In our configuration, several characteristics of the Core were included:

- 32 bits integer divider and 32 bits multiplier unit (with MAC²² feature). They are fundamental hardwares since the application software requires intensive division and multiplication. With this support, the instructions of multiplication and division are available for software development (UMUL, UMULCC, SMUL, SMULCC, UDIV, UDIVCC, SDIV, SDIVCC). They make the CPU more suitable for this project and can be used to emulate the floating-point operations. We see a significant portion of the power were consumed by these two units during the software/hardware interaction. (section 5.5.12)
- We selected 8 register windows to build the register file unit. The standard Sparc V8 architecture allows 2 - 32 register windows, each register window containing 32 general-purpose registers (32bits in each one). They are used by the hardware, the application software and the operation system. We could making the logic

²²Multipier Accumulator

synthesizer to map all the 256 registers into flip-flop cells, however, demanding huge area and power. To optimize the area and power consumption, a memory compiler from TSMC was used to generate a two port synchronous memory model for the register file. It shows follower characteristics:

- 256 words, 32 bits in each word.
 - Two muxes for row and column decoder.
 - Four output drivers.
 - Maximum operation frequency is 350MHz, characterized in 0.9V, 1.0V and 1.1V processes.
 - The two-port register file is a fully static memory and is self-timed with a minimum read and write power consumption (45.81mA peak read current and 47.91mA peak write current).
 - Its physical dimension are 225um (height) and 175.6um (width), resulting a 39510um² layout area.
- The Sparc V8 **UMAC/SMAC** (Multiply-Accumulate) instructions were included into the system. These instructions implement a single-cycle multiplication (32-bits of each operand) with a 40-bits accumulator. The details of these instructions can be found in the Leon3 instruction (Sparc V8) set manual[15].
 - **Instruction Cache** was added to allow maximum performance. Systems which disable the cache to save area have the performance impacts in the order of 2–3 times. The instruction cache can be implemented as a multiple-set cache with 1–4 sets. Higher associativity usually increases the cache hit rate and hence the performance. The downside is a higher power consumption and an increased gate-count in the tag comparators.

In our implementation, we choose 1-set direct-mapped type, which frequently results in a good cache with performance [PattersonHannence]. The basic size of each set in the instruction cache is 1 Kbytes. Large set size result in a higher performance, but might affect the maximum frequency (on ASIC targets). In our configuration, we chose a 8 Kbytes cache.

Instruction caches typically benefit from larger line sizes, but on small caches it might be better with 16 bytes/line so as to limit eviction miss rate. In our configuration, we choose 32 bytes per line. There is no cache replacement algorithm for directed mapped cache. In the configuration of 2-4 cache sets, the replacement algorithm can be random, LRR (least-recently-replaced) or LRU type (least-recently-

used). The LRU scheme has typically the best performance but shows the highest area overhead [Patterson].

The total instruction cache size is the number of sets multiplied by the set size. In our case, the instruction cache was built by 2 parts: the data memory of the instruction cache and the tag memory. Both memories were generated by the TSMC 90nm memory compiler (90nm compliant).

- The size of the data memory was equal to $1 * 8$ kbytes and is configured to use a single-port, synchronous, fully static memory which has 2048 words with 32 bits length per position. Totalizing in a 8 Kbytes of storage capacity. It has 8 muxes for row and column decoders, using 6 output drivers. The maximum operation frequency is about 350Mhz, under 1.1V, 1V and 0.9V. The physical dimension of memory was $256.3\mu m$ (height) and $457.1\mu m$ (width).
- The tag memory was configured to use a single-port, synchronous, fully static memory. It presents 256 words of 35 bits each, totalizing in a 8 Kbits of storage capacity. It has 8 muxes for row and column decoders and 6 output drivers. The maximum frequency also under 1.1V, 1V and 0.9V. The physical dimension was $112.1\mu m$ (height) and $485.1\mu m$ (width).
- **Data Cache** was also added to allow a maximum performance. In the original configuration of the Leon3, the data cache can be implemented as a multiple-set cache with 1–4 sets.

In our configuration, we used a direct-mapped data cache, and was chosen to be 1 kbytes by set, with 32 bytes on cache lines to reduce the tag memory.

Similar to the case of the instruction cache, there is no cache replacement algorithm for directed mapped data cache. In this case, the data cache was also built by 2 parts: a data memory and a tag memory. To make the design be symmetrical in placement and routing, the memories of data cache system were chosen to have the same footprint as the instruction cache. (8 kbytes for the data memory, with the footprints equal $256.3\mu m \times 457.1\mu m$ and 8 kbits for the tag memory, with the footprints equal $112.1\mu m \times 485.1\mu m$).

- **Memory Management Unit** was enabled to add hardware support for virtual memory mapping capability. The Table Lookaside Buffers (TLBs) of instructions and data caches were splitted into two parts. Each one of TLBs presents 8 entries, separately for icahce and dcache. The replacement algorithm can be LRU (Least-Recently-Used) or Increment (simple incremental replacement scheme). The LRU was chosen to enhance the access performance [11]. Both the entries of the icahce

TLB and the dcache TLB were mapped directly into flip-flop cells, without using a dedicated memory (the number of entries were small - sixteen 64 bits entries in total).

5.3.2 SRAM and SDRAM Memory Controller

The access to the external SDRAM and SRAM memory was done by the memory controller unit. This module is part of Leon3 IP library, and is designed to act as a slave on the AHB bus. The memory controller is programmed through the memory configuration registers MCFG1, MCFG2 and MCFG3 which are connected to the APB bus. The memory bus supports four types of devices: prom, SRAM, SDRAM and local Input/Output registers. The memory bus can also be configured in 8 or 16-bit mode for applications with low memory and performance. In Figure 5.6, we show the usage of this unit in our SoC and the respective connection for each kind of storage devices.

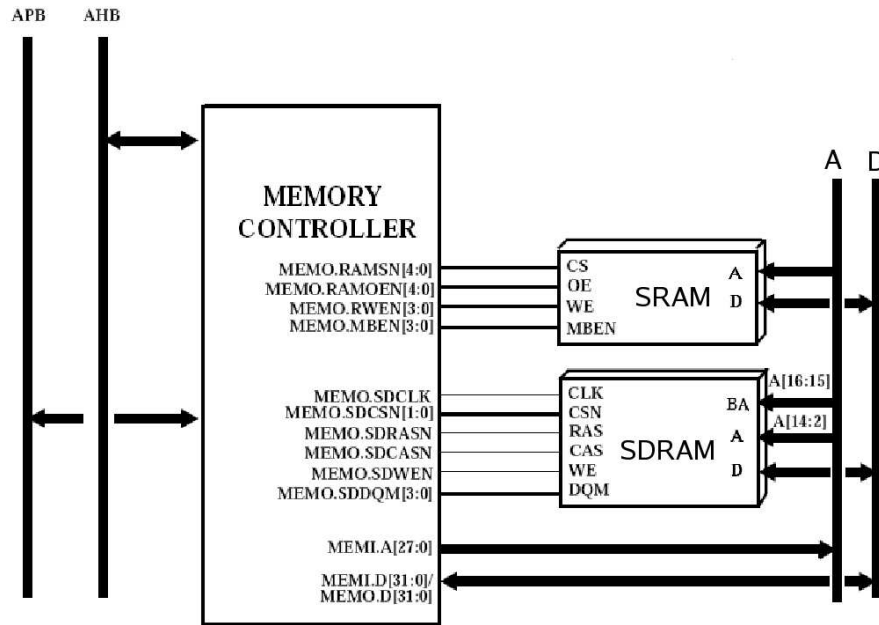


Figura 5.6: Leon3 Combined Memory Controller Unit

The PROM and local I/O registers are disabled to simplify the design complexity, alleviating the controller performance overhead. For SRAM memory access, this area can be up to 1 Gbytes, divided on up to five RAM banks. A read access to SRAM consists of two data cycles and three waitstates. On non-consecutive accesses mode, a lead-out cycle is added after a read cycle to prevent bus contention due to slow turn-off time of memories.

For Synchronous dynamic RAM (SDRAM) access is made through two banks of PC100/PC133 compatible devices. The SDRAM controller supports 64M, 256M and 512M devices with 8-12 column-address bits, and up to 13 row-address bits. The size of the two banks can be programmed in binary steps between 4 Mbytes and 512 Mbytes. Both 32 and 64-bit data bus width is supported, thus allowing the interface to 64-bit DIMM modules.

The memory controller can be configured to use either a shared or separate bus. In our SoC design, the bus is configured to have separate address and data parts with 32 bit width, connecting the controller and SDRAM devices.

A read cycle is started by performing an ACTIVATE command to the desired bank and row, followed by a READ command after the programmed CAS latency. The CAS Latency is the delay, in clock cycles, between the registration of a READ command and the availability of the first piece of output data. The latency can be set to one, two or three clocks. In our case, the CAS latency is configured to be 3 clock cycle, which support operation speed up to 166 MHz. The read cycle is terminated with a PRE-CHARGE command, no banks are left open between two accesses.

The write cycles are performed similarly to the read cycles, with the difference that WRITE commands are issued after activation. We also unsynchronized the SoC system internal clock with the external SDRAM clock phase. Here, the SDRAM controller output signals can be delayed with 1/2 clock with respect to the external SDRAM speed. This allow us to use an external SDRAM clock which in not strictly in phase with the internal clock speed. This propose makes the entire SoC system to be simulated in a different speed with respect to the SDRAM memory model (the PC100 SDRAM model used in this design presents maximum speed of 166 MHz, while the SoC system is simulated in 300 MHz at the baseline flow).

5.3.3 Timers

The General Purpose Timer Unit provides a typical prescaler and decrementing timers. We configured 4 timers for the SoC, and the Prescaler width is configured as 9 bits through the VHDL generic. The prescaler is used to divide the system clock down to 1 MHz; 9 bits allow the SoC to clock up to 512 MHz. The Timer width is configured as 32 bits. This width is recommended for the BCC compiler run-time and compatible with the Linux Kernel. The timer unit acts as slave on the AMBA APB bus, and is capable of asserting interrupt on timer underflow. The interruption signal can be configurable to be common to the whole unit or separated for each timer.

5.3.4 Generic Input/Output Ports

The general purpose input/output port unit is reserved to generic data communications. It also provides interrupt support and the port width was set to be 32 bits. Each one of bits can be individually set to be input or output port, and can optionally generate an interrupt signal. For interrupt generation, the input can be filtered for polarity and level/edge detection. The I/O ports are implemented as bi-directional buffers with programmable output enable. The input from each buffer is synchronized by two flip-flops in series to remove potential meta-stability.

5.3.5 UART Controller

The **U**niversal **A**synchronous **R**eceiver **T**ransmitter interface is provided for serial communication. The UART supports data frames with 8 data bits, one optional parity bit and one stop bit. Each UART has two FIFO²³ data buffers reserved for transmitter and receiver operation. The FIFO depth is set to 8 bytes and each UART contains a 12-bit down-counting scaler to generate the desired baud-rate. The scaler is clocked by the system clock and generates a UART tick each time it underflows. It is reloaded with the value of the reload register after each underflow. The resulting UART tick frequency should be 8 times the desired baud-rate. We set the UART default baud-rate equal to 115200 bits/s, and reconfigurable by the SoC operating system setups.

5.3.6 IRQ Controller

The interrupt controller unit were built for both AHB/APB. Interrupts from AHB and APB units are routed through the bus, combined together and propagated back to all units. The multiprocessor interrupt controller unit is attached to the AMBA bus as an APB slave, and monitors the combined interrupt signals. All the interrupt signals generated on the interrupt bus are forwarded to the IRQ unit and the controller prioritizes, masks and propagates the interrupt signals with the highest priority to the processor.

5.3.7 External Memory Simulation Model

The SDRAM external memory model, provided by the Micron Technology Inc.²⁴, was used to store the embedded software of the Leon3 system. It was responsible to provide input stimulus (video decoding information) and collect the switching activities of the entire system.

²³First In, First Out

²⁴www.micron.com/sdram

This 128 Mbits SDRAM memory is a high-speed CMOS, dynamic random-access memory and contains 134,217,728 bits. It is internally configured as a quad-bank DRAM with a synchronous interface (all the signals are registered on the positive edge of the clock signal).

Each one of the 33,554,432 bit banks is organized as 4,096 rows by 256 columns by 32 bits. Both the Read and Write accesses to the SDRAM are burst oriented and initialized with the registration of ACTIVE command, which is then followed by the READ or WRITE commands.

This memory was designed to operate in 3.3V at 166 MHz clock.

5.3.8 MPEG-2 Decoder Modification

After we finished the setup of the Leon3 platform, we modified the MPEG-2 video decoder as an embedded software.

The selected implementation of MPEG-2 decoder is an open source program developed by the MediaBench Consortium²⁵. The Mediabench group selected several multimedia and communication applications, which were designed to evaluate performance on several microprocessor platform.

This decoder was not optimized for speed or power and just emphasizes a correct implementation of the algorithm. In next, we list some features of this MPEG-2 decoder:

- Supports Simple, Main, SNR scalable and spatially scalable streams at all defined levels.
- Decodes MPEG-1 (ISO/IEC IS 11172-2) video bitstream.
- Support several output formats: Separate and interleaved Y,U,V component, Truevision TGA, PPM, X11²⁶ display.
- Support 8 bit ordered dither and interlaced to progressive scan conversion for X11.
- No support for temporal scalability and error concealment feature.

After analyzed the MPEG-2 source code, in order to explain modifications, we summarize the video decoding flow in Figure 5.7.

As shown, the decoding sequence contains three steps: video data input, decode video bitstream and store decoded frames. The first part consists in detecting the video file and constructing a bitstream buffer, filled out with compressed video data. The final

²⁵2009, <http://euler.slu.edu/~fritts/mediabench>

²⁶Display library of Unix based operation system

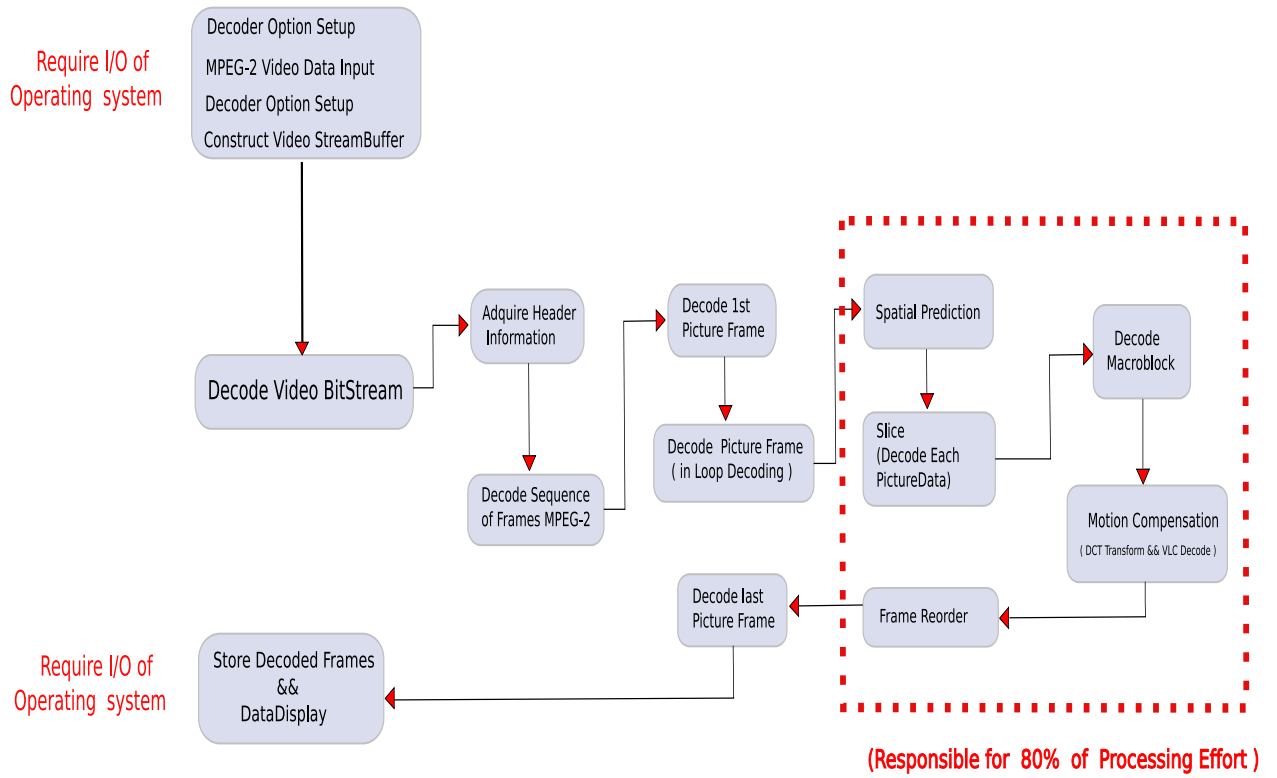


Figura 5.7: MPEG-2 Decoder Data Flow

step consists in storing the uncompressed video bitstream into a hard disk. Both the first (**MPEG2 Video Data Input**, from the Figure 5.7) and the third step (**Store Decoded frames**, from the Figure 5.7) require I/O features of the host hardware and operating system to work properly.

We note that the existence of the operating system interfered the switching activity of entire SoC system and made the power estimation be imprecise and error prone. In order to workaround this issue, several adjustments in the MPEG-2 software and in the SoC system were done.

In next, we describe how the MPEG-2 decoder were adapted, modified and cross-compiled to be run on our SoC system without the operating system and I/O dependence.

1. First, all the modifications and tests were done using an X86 machine (Intel Pentium 4, 2.6 GHz), then the decoder was cross-compiled (using the BCC) to generate the binary code for our target platform. Because our Leon3 processor was configured to implement the Sparc V8 multiply and division instructions, the *-mv8* switch was required during both the compiling and linking (the BCC compiler does not issue these instructions by default). The *-mv8* switch improved the performance

on compute-intensive applications and floating-point emulation, just like our case. Purposely we choose a implementation of MPEG-2 from MediaBench because it the float-point emulation of MPEG-2 from MediaBench introduces significant performance impact on the decoding speed of video streams. However, these kinds of stimulus vectors can worsen significantly the total power dissipation (almostly the worst case of power dissipation), and make the power estimation of baseline flow and other optimization cases were built based on the worst case of switching activity.

To input the video stream into the decoder without using any storage device or operating system, we decided to extract a small amount of frames from a video stream, and stored the compressed frames in form of static vector, which in turn, would be loaded into the SDRAM before the video decoding (**Decode Video Bitstream**, from the Figure 5.7)

In a better explanation, we stored the frames of compressed video bitstream in its corresponding decimal representation, building a static vector as bellow :

```

1      extern unsigned char static_buffer [2048*20] = {
2
3      0,  0,  1, 186,  68,
4
5      0,  4, 25, 156, 121,
6
7      0, 70, 83, 248,  0,
8
9      0,  1,  2, 123, 178,
10
11     2,  5,  1, 24,  118,
12
13     7,  0,  11, 19,  201,
14
15     .....
16     .....
17     .....
18     .....
19     } ;

```

Each one of these values represents 1 unit of the compressed data, which is 8 bits in width. To a better understanding, we called this vector as input-vector.

We also built another static vector, called as output-vector to store 10 decoded frames, resulting from the input vector. This output-vector was used to check if the decoding was done correctly during the simulation.

In our case, the input-vector contained 10 compressed video frames and the output-vector contained 10 uncompressed pictures.

2. Before the simulation engine initialized the decoding procedure, both the input-vector and the output-vector were loaded into the SDRAM model. During the simulation, the input-vector would continuously feeding the decoder with compressed video to generate switching activities until the end of the simulation.

The decoded frames were compared with the output-vector to ensure the results. This task could not be considered as part of decoding, hence their switching activity would not be used to estimate the total power.

By using this strategy, we eliminated the I/O dependence in receiving and storing the video data and the entire decoding was free of any interference of operating system. This approach made the video decoding as the unique source of switching activity and the power estimation could be clean and straightforward.

3. Another relevant aspect in the Figure 5.7 is about the effort of frame decoding. As we can see, the each frame was decoded by the spatial prediction, followed by the macroblock decoding, the motion compensation and finally the reordering of frames. This steps were responsible for about 80% of the entire flow.

In the baseline RTL simulation, these steps usually took 12-15 hours in a Intel X86 machine with Quad Core CPU, working at 2.4 GHz and running the Centos 4.7 operating system. The simulators were **Modelsim v6.2b**, from the **Mentor Graphics Inc.** and **NCSIM v8.1** from the **Cadence Design System**. The simulation time became worse when we performed the post-layout simulation with path delays. The results can be found in the section 5.5 and the section 5.5.12.

5.3.9 Extract Switching Activity

In section 2.2.4 we showed that the switching activity is fundamental to estimate estimation and can be annotated in several formats: VCD, SAIF or TCF. We adopted the Toggle Count Format (TCF) to represent and store switching information of each pin/nets of the entire system.

TCF was generated from the simulation process and stores (1) toggle counts, which indicate how often a pin or net switched between logic 1 and logic 0 during the

simulation interval and (2) Probability of a pin or net to be still in logic 1. Next, the TCF syntax is shown and briefly described:

```

1      tcf file ( ) {
2          tcfversion:<TCF file version> ;
3          generator:<Tools to output TCF file> ;
4          date:<TCF creation date> ;
5          duration:<Simulation duration to build TCF> ;
6          unit:<Timing Unit> ;
7          instance ( <name> ) {
8              pin ( ) {
9                  <pin1-name>: "logic(1)-prob" "toggle-num" ;
10                 <pin2-name>: "logic(1)-prob" "toggle-num" ;
11                 .....
12             } ;
13             net ( ) {
14                 <net1-name>: "logic(1)-prob" "toggle-count" ;
15                 <net2-name>: "logic(1)-prob" "toggle-count" ;
16                 .....
17             } ;
18         } ;
19     } ;
20 
```

According to the TCF syntax, each instance contains several nets and pins. Each net and pin is described by their name, their probability to keep in logic one and the total toggle number during the simulation.

The simulator managed all the timing issues and data flow of the video decoding, at the same time, computed/saved the corresponding switching activity along the simulation.

The Figure 5.8 shows how the simulator built a virtual hardware/software environment after the design description was loaded. In our case, all the Leon3 SoC system was loaded and the SDRAM simulation model was filled with the MPEG-2 decoder and video frames.

After the simulation was initialized according to the design parameters (100

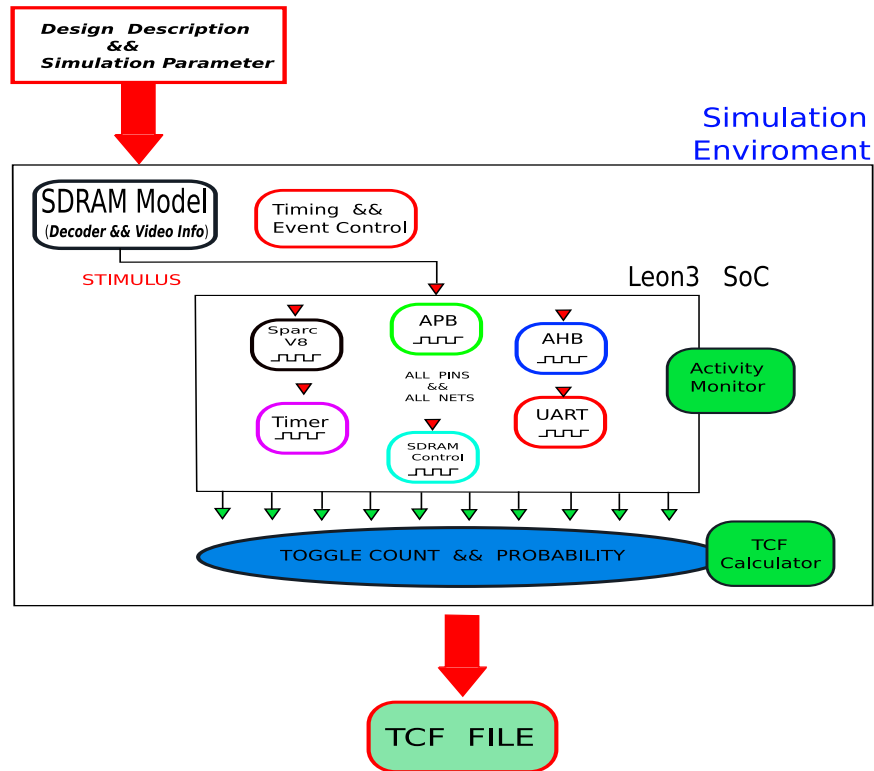


Figura 5.8: System Simulation and TCF File Construction

and 300 MHz clock, in our case), the simulator would store the switching information and monitor all the pins and nets to calculate their switching number and probability. By the end of the simulation, the TCF file would be filled out with precise switching data.

5.4 Manufacturing Process

Several manufacturing processes were analyzed to develop this project. There were three possible candidates from TSMC ²⁷: the 45nm, 65nm and 90nm process. The 90nm node was chosen because it was the unique one which memory compiler is available for the academic users. As part of 90nm process, several characterizations were available for different requirements. They can be listed below:

- General Purpose node (G): proper for generic ASIC application in which neither power nor performance are critical. In this node, the standard cells are designed to support techniques like multiple threshold voltages and clock gating.

²⁷Taiwan Semiconductor Manufacturing Company, the world leader of pure-play foundry

- High Performance node (GT): proper for applications that the performance is critical without power concern. The standard cells are designed to support multiple threshold voltages.
- Low Power node (LP): proper for applications which show critical power constraints. It is specially designed to support advanced techniques to meet power and performance goals. Several special cells are available to support the multiple threshold voltages, multiple supply voltages, clock gating, power gating and back-bias.

In this project, the general purpose node (G) was chosen to build the baseline flow and the low power node (LP) for power optimization. It is convenient to explain some manufacturing characteristics of the two processes before we continue.

5.4.1 General Purpose Node

The general purpose node is identified as the TCBN90G TSMC 90nm standard-cell library. It was designed to support 1.0V and 1.2V nominal power (V_{dd}) and provides low leakage option for a large-gate-count design. It does not provide the de-rating factors, instead, the library was characterized at the MAXIMUM, TYPICAL and MINIMUM corners. The library also includes characterization for several corner cases, shown in table 5.1.

Corner Characterizations of TCBN90G				
Corner Case	Designator	Speed	Voltage	Temperature
Worst	WCCOM	SLOW	$V_{dd} * 0.9$	125°C
Typical	NCCOM	TYPICAL	V_{dd}	25°C
Best	BCCOM	FAST	$V_{dd} * 1.1$	0°C
Low Temp	LTCOM	FAST	$V_{dd} * 1.1$	-40°C

Tabela 5.1: Library Corner Case Table

Leakage power for 90nm technology is an important factor in total chip power management. Related with the sub-threshold current and the gate leakage voltage, the leakage power is also strongly dependent on the logic state of input pins. In the synthesis libraries, all the possible states of the input pins were characterized and associated with a specific leakage consumption.

For instance, in the case of an ND3D1 cell (three-input NAND gate), the worst-case leakage power is 5.742 nW (when all three inputs are 1) and the best-case leakage power is 0.040 nW (when all three inputs are 0). For the power management, the best-case state can be used as an idle state for power saving purpose. For the leakage power estimation, we can use the average leakage power (between the best and the worst case).

The library provides 4 categories of cells and each one presents different functions. They are described briefly in next:

1. Combinational Logic Cells: the combinational cells include following types: Inverters, Buffers, AND, OR, NAND, NOR, XOR, NXOR, Full-Adder, Half-Adder. Each type of these combinational cells present 5 different driving strength.
2. Sequential Logic Cells: sequential cells include latches, flip-flops and scanable flip-flop cells. Each flip-flop has a scanable version, optimized to the DFT²⁸ structure.
3. Datapath Cells: 34 datapath cells are provided for performance sensitive blocks (ALU, for example). They were optimized to be used in the high performance design.
4. Special Purpose Cells: this category contains clock-gate cells, antenna fix cells, filler cells, delay cells and decoupling cells. The clock-gate cells are reserved to implement automatic clock-gating insertion during logic synthesis. Antenna fix cells are designed to prevent antenna effects during the manufacturing, resulted from CMP²⁹ effect.

Filler cells provide continuity for the power rails, as well as for the n-wells. They are used to fill the gap between adjacent cells after the placement and routing. Delay buffers are reserved for clock tree synthesis and timing closure during physical design. Decoupling cells provide capacitors between power and ground to reduce switching noise, very important in high frequency design. They also can be used to replace the filler cells after the placement and routing.

Corner Characterizations of TCBN90LP					
Corner Case	Designator	Speed	Voltage	Temperature	Leakage
Worst	WCCOM	SLOW	$V_{dd}=1.08$	125°C	Low/High/Nom V_{th}
Typical	NCCOM	TYPICAL	$V_{dd}=1.2$	25°C	Low/High/Nom V_{th}
Best	BCCOM	FAST	$V_{dd}=1.32$	0°C	Low/High/Nom V_{th}
Low Temp	LTCOM	FAST	$V_{dd}=1.32$	-40°C	Low/High/Nom V_{th}
0.7 Worst	WC0D7COM	SLOW	$V_{dd}=0.84$	125°C	Low/High/Nom V_{th}
0.7 Typical	NC0D7COM	TYPICAL	$V_{dd}=0.84$	25°C	Low/High/Nom V_{th}
0.77 Best	BC0D77COM	FAST	$V_{dd}=0.924$	0°C	Low/High/Nom V_{th}

Tabela 5.2: Low Power Library Corner Case Table

²⁸Design For Test

²⁹Chemical and Mechanical Polishing

5.4.2 Low Power Node

The low power node (LP) is identified as the TCBN90LP 90nm low power library. It is designed to have same physical characteristic as TCBN90G process (the cells presents same footprints and corner characterizations) with additional power requirements.

It also presents special purpose cells, appropriate to implement advanced power management techniques. Corner cases are provided for 3 different threshold voltages: Low V_{th} , High V_{th} and Nominal V_{th} (see section 2.1.2). The table 5.2 summarizes the corner cases as in TCBN90LP node. The nominal V_{dd} is 1.2V.

The TCBN90LP process also provides special features for those designs which demand multiple power supply. In this case, their timing/power model operates under 0.7/0.77V condition (the last 3 rows of table 5.2).

The library contains 4 categories of cells, similar to the TCNB90G process and provides several additional cells listed below:

- Level Shifter Cells are provided to help in the implementation of multiple voltage designs. When a signal propagates from low-voltage to high-voltage block, it is necessary a VDD_{Low} to VDD_{high} level shifter to ensure there is no abnormal leakage (low-voltage V_g cannot turn-off the PMOS device completely). The High-to-Low level shifter is required when the signal propagates from the high to the low voltage domains. In this case, it is not necessary to consider the abnormal PMOS behavior, because a normal buffer can be used as level shifters. Please refer to the section 5.9 for more details.
- Isolation Cells are provided to implement the technique of power shut-off. In this approach, several power domains are presented and designers can cut down the power supply for certain block in order to reduce the power consumption.
- Well-Bias Cells: traditionally, inverters have PMOS/NMOS bulk voltages tied to the V_{dd} and Gnd respectively.

Figure 5.9 shows how the Well-Bias technique is implemented to dynamically modify the CMOS threshold voltage, controlling the bulk terminal. The bulk terminal are connected to an external power source (VPP and VBB) and with this modification, the leakage current can be dynamically controlled according to the system requirements and so, saving the unnecessary dissipation due to leakage effect.

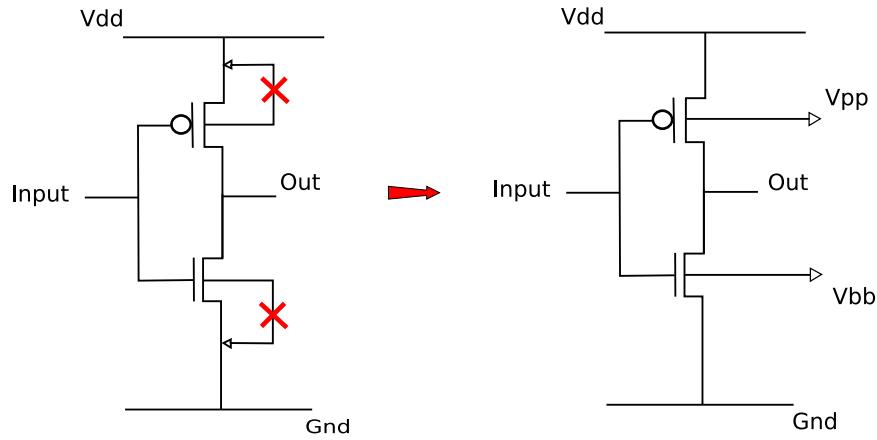


Figura 5.9: Inverter Circuit with Well-Bias

5.5 Baseline Flow

As mentioned in section 5.3, the average power consumption of the hardware without any power optimization is an important metric to evaluate the efficiency and impact of each low power technique. In this section, we describe in detail how the average power were estimated based on the extracted switching activity, moreover, we show the experimental results and explain how they were used to evaluate other power optimization techniques.

Based on the classical flow presented in section 4.1, we describe the baseline flow in details.

5.5.1 System Requirement and Spec Definition

Beside the modification of the Leon3 system and the MPEG-2 decoder (section 5.3), we also define the clock speed in 100MHz and 300MHz to evaluate the relationship between the speed and power consumption. We also prepared several testbench to simulate the RTL design, the gate level netlist and the ATPG³⁰ vector.

5.5.2 Library Qualification

After the manufacturing process was analyzed and selected, the library qualification was performed. It checked if the standard cell data was complete and satisfied the requirements of each power technique. This task was done in three steps:

³⁰Automatic Test Pattern Generation

- Basic library checks for the design feasibility and implementation. It checks if the timing/power/physical information is present and if they are compatible with the EDA tools. We performed a test run using a small design to validate this step.
- Interaction with Foundry or PDK³¹ vendors for problems and solutions. As academic users just access the front-view of the process kit, the TSMC informs that several confidential information would not be provided in the library (like GDSII view, DRC/LVS rules, Spice files). This made several tasks (e.g manufacturing sign-off and power sign-off) be partially completed. In a commercial version of the PDK, these data was provided, so as, we could finish up such tasks.
- Cell requirement for power techniques: in this step, the design team should plan the power intent representation³² and checking the corresponding cells required by each technique. Usually, a correlation table is built to associate each technique with the library cell, as shown in the Table 5.3, where

Library Qualification Table	
Basic Techniques	Required cells
Clock Gate	Integrated clock gating cell with built-In half latch for glitching removal
Multi Vth	High Vth, Low Vth, Nominal Vth
Operand Isolation	Standard Cells from <i>PDK</i> ¹
Advanced Techniques	
Multiple Supply Voltage	Level shifter cell and their fillers
Power Gate (<i>PSO</i> ²)	Isolation cell, Combo <i>cell</i> ³
Multi Supply with <i>PSO</i> ²	<i>SRPG</i> ⁴ cell, Isolation cell, Always-on cell, Level Shifter cells, Header/Foot coarse grain cell, special filler for ring style power gate.
<i>DVFS</i> ⁵	Isolation cell, Level shifter cell, Header/Foot coarse grain cell

Tabela 5.3: Low Power Design Library Qualification Table

1 stands for **P**rocess **D**esign **K**it ;

2 stands for **P**ower **S**hut **O**ff ;

3 means this cell integrates both level-shift and isolation features ;

4 stands for **S**tate **R**etention **P**ower **G**ate ;

5 stands for **D**ynamic **V**oltage and **F**requency **S**caling.

³¹Process Design Kit

³²can be CPF, UPF or any other industrial standard, conform the 4.2

By doing so, we confirmed that the clock gating cells, the level shifter cells, the isolation cells, the antenna fix cells are present in PDK, accompanied by their respective documentation. In the case of a design team, if any required cell is not present in the library, the team must develop an in-house version and validate it to prevent silicon risk or license it from some third party IP vendor. If the chosen technique cannot be implemented due to the lack of cells, a new strategy must be used.

5.5.3 Register Transfer Level Simulation

The RTL Soft/Hardware Co-simulation was used to ensure that the hardware configurations were compatible with the embedded software. We validate the entire design using an Altera StratixII FPGA development kit and, then logic simulators from the **Mentor Graphics** and the **Cadence Design System**.

When the entire simulation was running on the FPGA, 10 frame was decoded in 3 seconds. Apparently, when the platform was prototyped on FPGA, the performance was limited to a 50MHz clock speed which is not enough to produce a real time decoding.

The logic simulator completes the entire flow in about 16 hours, with the simulation clock set to be 300MHz. The simulation system indicated that the 16 hours was equivalent to 550ms in real time. This means we could decode about 18 frames per second if the decoding could be done by our target hardware. The machine used for simulation was an Intel Quad Core processor, with 2.4 GHz clock and 4 GB DDR2 RAM memory.

5.5.4 Logic Synthesis

The entire system was analyzed and synthesized by the **Cadence RTL Compiler**. Timing and power libraries were required to achieve the design specs. The physical information of each standard cell was used to estimate the total area. There were no issues when performing this task, the entire process took about 1 hour to map all the system into the netlist.

5.5.5 Design For Test

The post-manufacturing test structures were inserted automatically using the **RTL Compiler**. Then, the **Encounter Test** of **Cadence Design System** was used to generate the corresponding ATPG vector, so as to achieve a 95% coverage. Both the DFT insertion and synthesis took about 1 hour to be finished.

5.5.6 Gate Level Co-Simulation

Gate level co-simulation was used to ensure that the output netlist is logically equivalent to the original RTL design. This step could not be used to extract the switching activity because the this netlist was not correct in term of timing and power (e.g no clock tree, no delay buffers, no power rails and no antenna fix cells). This step took about 240 hours to decode 10 videos frames.

5.5.7 Placement and Routing

We imported all the physical, timing, power information of each standard cell and macro cells. The back-end information of the SRAM memories was also imported with the synthesized netlist (e.g tag ram of the instruction cache, data ram of the instruction cache, tag ram of the data cache, data ram of the data cache, the register file) to complete the physical synthesis.

The entire process was done in about 2.5 - 3 hours, without timing violations (setup and hold time were met). It was necessary several interactions to eliminate the DRC³³ and fixing the antenna effects. With these additional steps, the entire run was extended to 4.5 hours.

5.5.8 Static Timing Analysis

The STA was used to ensure the timing closure in all the paths. There were no critical path both in the setup and hold time. The system was timing closure at 300MHz clock speed.

5.5.9 Parasitic Extraction

We successfully extracted parasitic elements from the nets and gates in the layout. These data were fundamental to calculate the path delay and then estimate the power. The output data was dumped in SPEF³⁴(10 minutes).

5.5.10 Post Layout Co-Simulation

With all the physical information and path delay in form of SDF³⁵, the final netlist was imported into the adapted testbench and the co-simulation procedure was repeated to extract the precise switching activity. This process took about 240 hours to decode 10

³³Design Rule Check

³⁴Standard Parasitic Extraction File

³⁵Standard Delay File

MPEG-2 frames, and the TCF file (section 5.3.9) contained all the switching information of the gates, nets, buffers, appropriate to estimate the total power.

5.5.11 Baseline Power Estimation

Voltage Storm was the power estimation engine from the **Cadence Design System**. It required some data to estimate the power dissipation: the final netlist, the power views of each macrocells and standard cells, the parasitic data, the clock speed and the TCF file. We estimated the power of the entire system (300MHz and 100MHz clock) and the results are shown in section 5.5.12.

5.5.12 Experimental Results

In this section, we show the baseline results the design was simulated at 300MHz and 100MHz. These results were estimated using the switching activities from the post-layout simulation. We also show some analysis results in terms of area, leakage power, total power and power-gate coefficient of each module.

Notice that the follower symbols were used to denote the design parameters and the module names. In all the forwarding sections, these notations were adopted in tables and figures.

- **Design Parameters Symbols**

- **Area** denotes area of gates in each module. Unit is mm^2 .
- **Switching** denotes power dissipation by switching activity in nets and gates for each module. Unit in μW .
- **Internal** denotes gate power dissipation in a short circuit situation. Unit is μW .
- **Leak** denotes power dissipation of the modules due to leakage current. Unit is μW .
- **Clock Power** denotes power dissipation of the clock tree for each module. Unit is μW .
- **Total Gates** denotes the total number of gates presented in each module.
- **Total Power** denotes the total power dissipation in each module. Unit is μW .
- **Power/Gate Coef** denotes Power-Gate coefficient, which is basically the ratio of the power dissipation of module and its number of gates. Unit is $\mu W/Gate$.

- **Module Symbols**

- **ahbctrl** stands for AHB bus controller.
 - **apbctrl** stands for APB bus controller.
 - **apbuart** stands for UART connected in APB bus.
 - **div32** stands for 32 bits division unit.
 - **mul32** stands for 32 bits multiplier unit.
 - **Dtag** stands for tag memory of data cache unit.
 - **Ddata** stands for memory of data cache unit.
 - **Itag** stands for tag memory of instruction cache unit.
 - **Idata** stands for data memory of instruction cache unit.
 - **gptimer** stands for 4 system timers.
 - **grgpio** stands for generic Input/Output unit.
 - **irqmp** stands for multiple-processor interruption request unit.
 - **mmu_icache** stands for interface of MMU and instruction cache unit.
 - **mmu_dcache** stands for interface of MMU and data cache unit.
 - **mmu_acache** stands for interface of MMU and AMBA bus.
 - **pipeline** stands for 7 stage integer pipeline.
 - **regfile** stands for memory unit of register file.
 - **sdmctrl** stands for SDRAM/SRAM memory controller unit.
 - **mmu** stands for Memory Management Unit.
- Table 5.4 shows several design result parameters of the baseline flow simulated at 300MHz.
 - Figure 5.10 shows the area of each module at the baseline flow. It is interesting to note that the cache is responsible for about 40% of the entire layout area, but representing only 26% of the total power.

Compared with the MMU unit, which was built from flip-flop cells, although it had 6% of the total area, but was responsible for 11% of the total power. This fact shows that the memory model from the foundry presents a better power efficiency in term of dynamic and static consumption.

Module Name	Area	Internal	Switching	Leak	Clock Pwer(%)	Total Pwer	Total Gates	Pwer/Gate Coef
ahbctrl	20465.6	1217	356.7	239	412.59	1813	8459	0.21
apbctrl	9481.7	1072	146.1	111	387.4	1329	3919	0.34
apbuart	3680.4	1009.1	302.4	91.5	238.51	1403	1521	0.92
mul32	13495.9	3963	684.5	210.9	730.16	4858	5578	0.87
div32	5585.9	1216	240.1	628	378.6	1519	2309	0.66
Dtag	54374.68	9551.61		170	x	9721.65	22480	0.43
Ddata	121261.77	9915.89		387	x	10302.94	50133	0.21
Itag	54374.68	11465.27		170	x	11635.3	22480	0.52
Idata	121261.77	10810.71		387	x	11197.75	50133	0.22
gptimer	14685.4	3816	468.4	194	658.1	4478	6070	0.74
grgpio	6032.7	2500	294.8	736.4	517.5	2869	2493	1.15
irqmp	2461.1	926.6	95.33	282.5	351.3	1050	1017	1.03
mmu_icache	4540.8	1450	227.1	559.3	404.4	1733	1877	0.92
mmu_dcache	13946.7	4124	694.3	167	629.1	4989	5765	0.87
mmu_acache	10381.6	657.2	464.2	165	351.3	1287	4291	0.3
pipeline	36360.6	18150	3664	406	172.4	20220	21358	0.95
regfile	79642.4	3638		380	x	4018	37343	0.11
sdmctrl	3283.7	1239	182.4	38.66	388.7	1460	1357	1.08
mmu	60457.5	17251		992	6567.8	18244	25726	0.71
Total without Layout	641310	x		5788	5788	x	274309	x
Total with Layout	916567.22	121500	28650	6315.26	9848.1	155969	302158	0.52

Tabela 5.4: Baseline Power Result at 300MHz

Module Name	Area	Internal	Switching	Leak	Clock Pwer(%)	Total Pwer	Total Gates	Pwer/Gate Coef
ahbctrl	20465.6	1161	225.3	237.5	412.9	1624	8459	0.19
apbctrl	9481.7	855	144.2	111.1	328.34	1110	3919	0.28
apbuart	3680.4	902.12	278.1	90.5	201.514	1270.72	1521	0.84
mul32	13495.9	1013.6	154.3	181.1	172.25	1359	5578	0.24
div32	5585.9	258.5	55.43	628	80.91	1093.3	2309	0.47
Dtag	54374.68	1992.6		170	x	2163	22480	0.1
Ddata	121261.77	2055.54		387	x	2422	50133	0.05
Itag	54374.68	2202.73		170	x	2373	22480	0.11
Idata	121261.77	2032.41		387	x	2419	50133	0.05
gptimer	14685.4	795.1	97.58	193.6	139	1086	6070	0.18
grgpio	6032.7	520.9	61.42	736.4	109.7	655.9	2493	0.26
irqmp	2461.1	926.6	95.33	282.5	351.3	1050	1017	1.03
mmu_icache	4540.8	301.8	47.26	559.3	86.12	405	1877	0.22
mmu_dcache	13946.7	859.9	144.6	167.1	132.9	1172	5765	0.2
mmu_acache	10381.6	136.9	96.71	165.1	68.5	398.7	4291	0.09
pipeline	36360.6	18150	3664	406	172.4	9220	21358	0.43
regfile	79642.4	2891		380	x	3271.66	37343	0.09
sdmctrl	3283.7	258.2	38.01	380.66	82.85	334.9	1357	0.25
mmu	60457.5	1631.94		992	3261.14	10631.94	25726	0.41
Total without Layout	641310	x		5788	x	x	274309	x
Total with Layout	916567.22	37980	8952	6309.5	2797	55750.00	302158	0.18

Tabela 5.5: Baseline Power Result at 100MHz

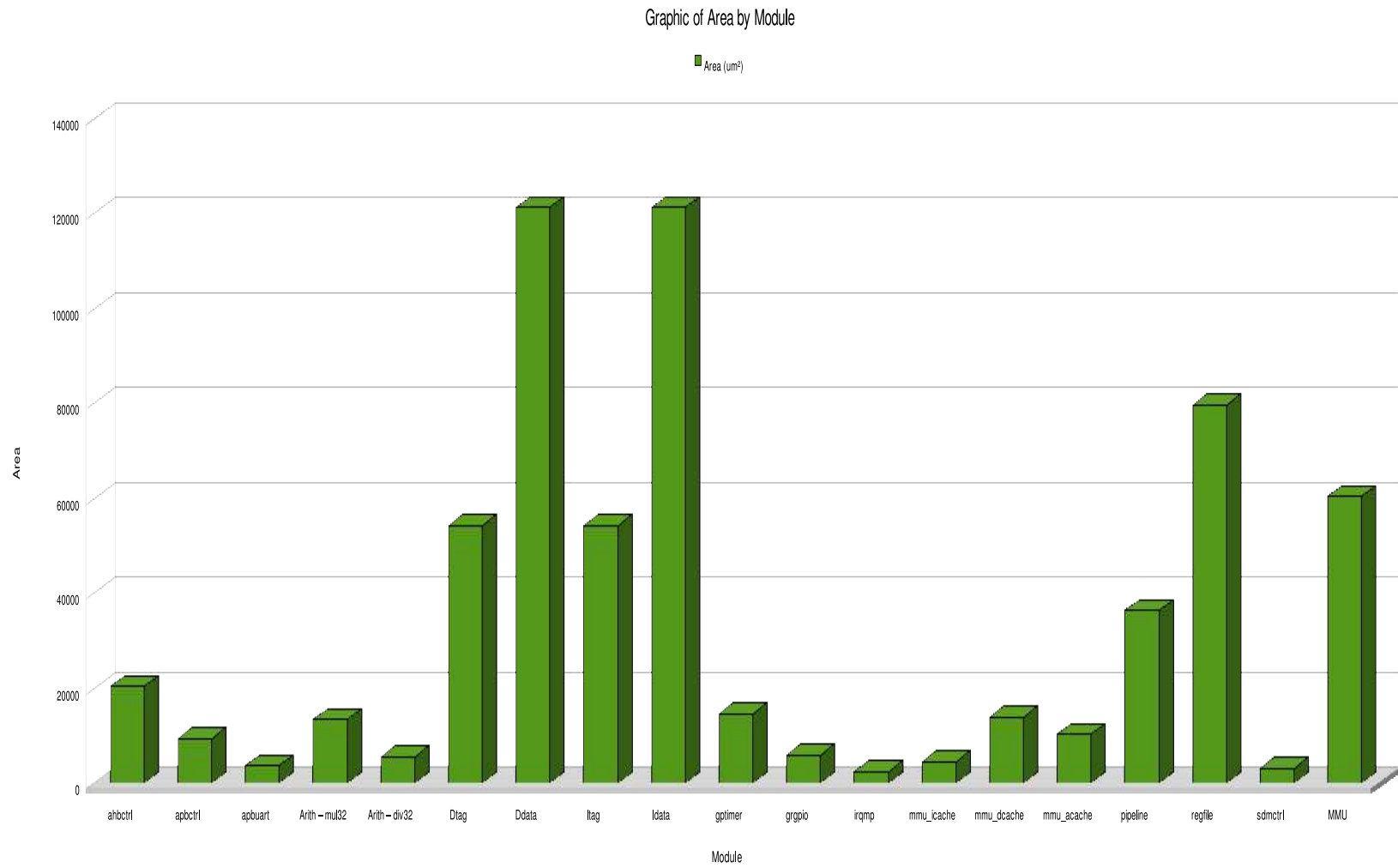


Figura 5.10: Baseline at 300MHz - Area by Module

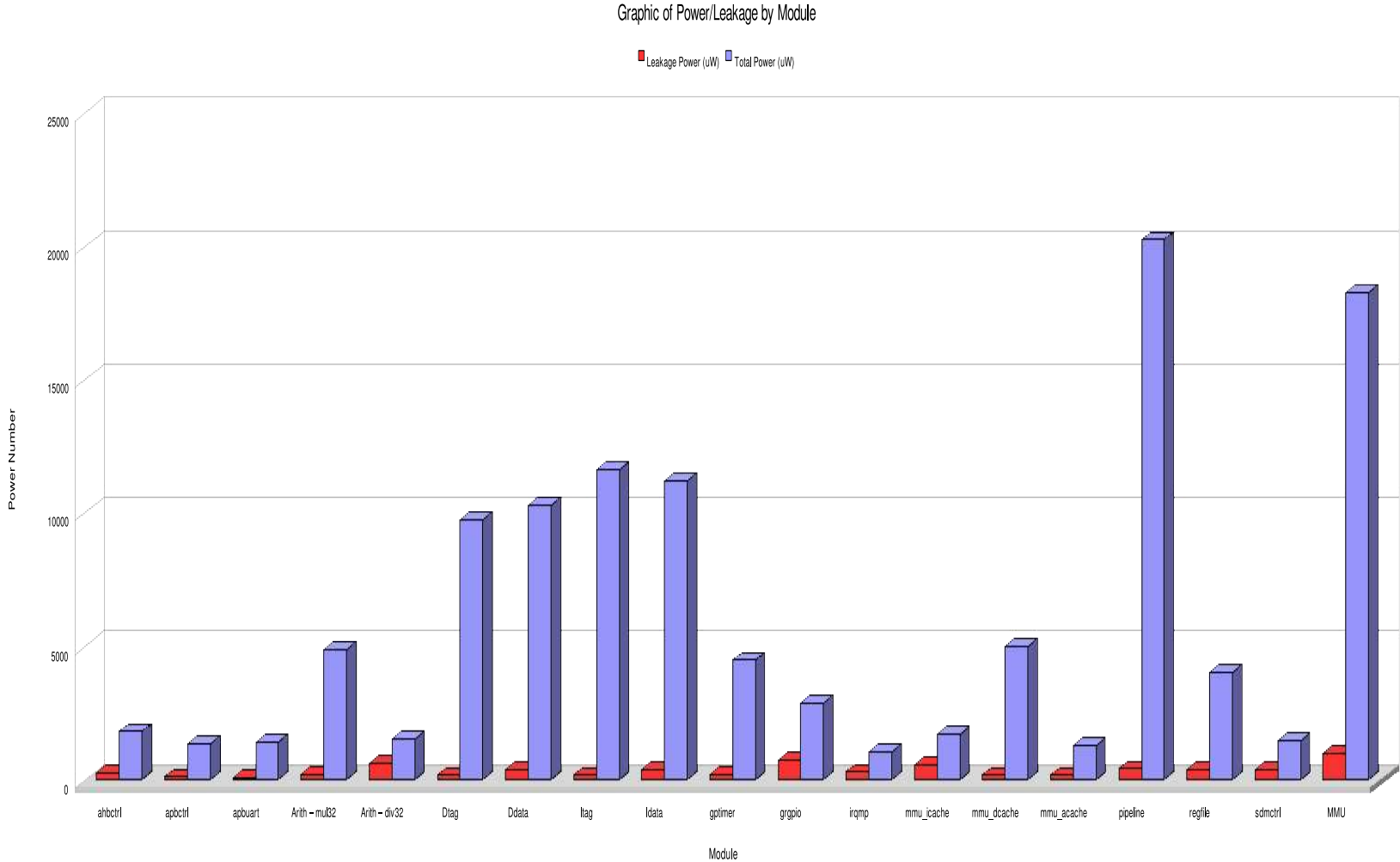


Figura 5.11: Baseline at 300MHz - Total Power/Leakage Power by Module

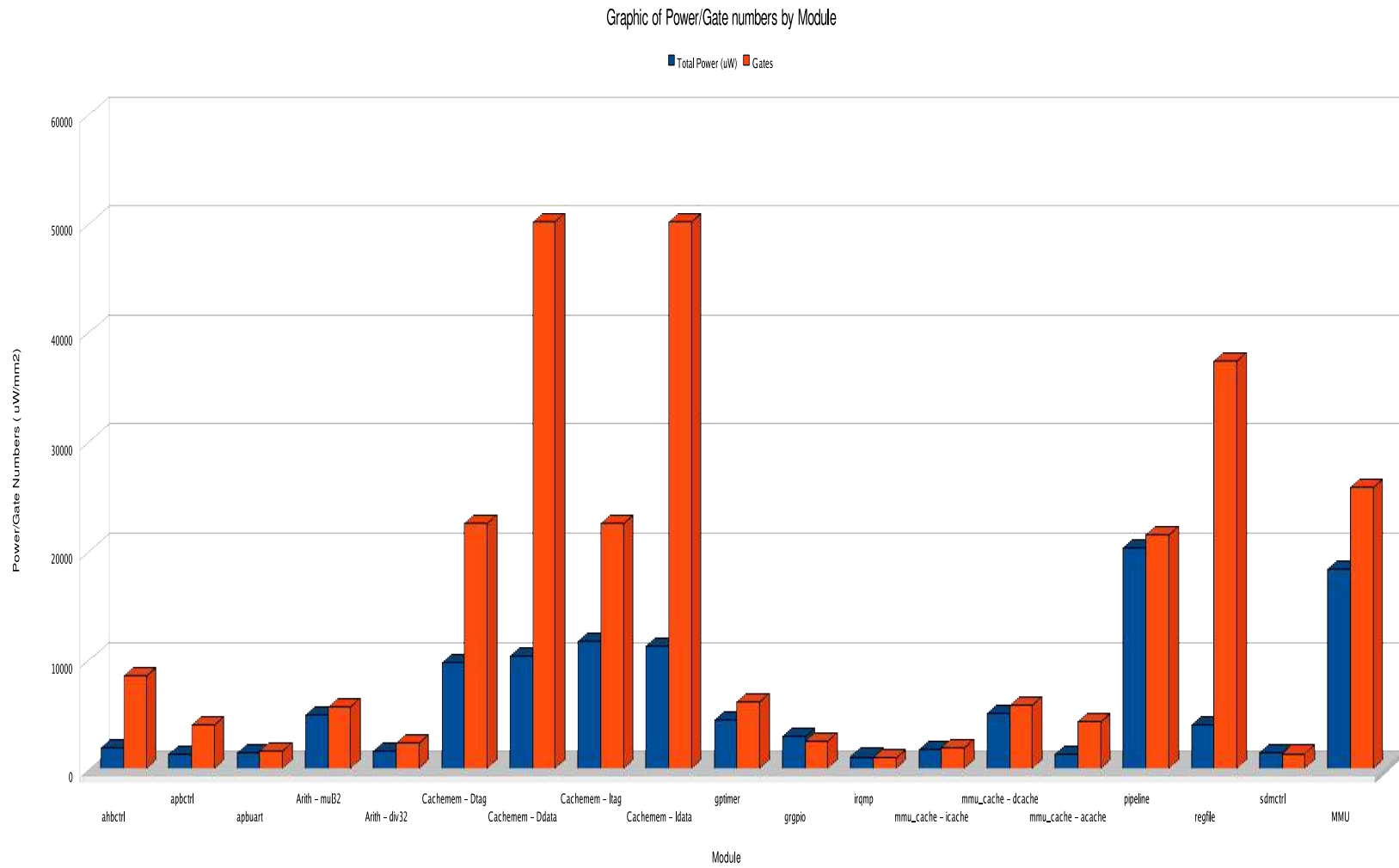


Figura 5.12: Baseline at 300MHz - Power and Gate Numbers by Module

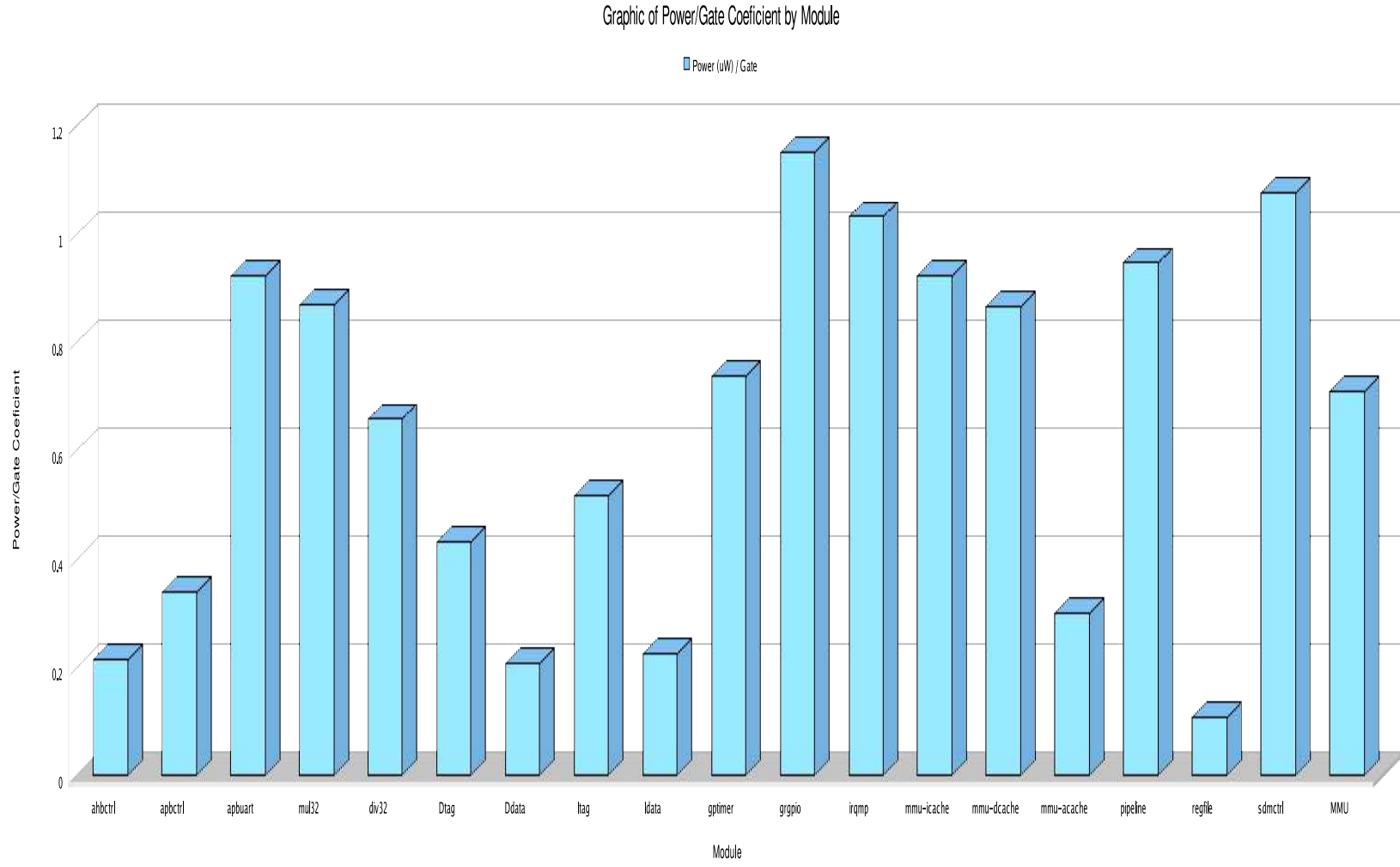


Figura 5.13: Baseline at 300MHz - Power/Gate Coefficient by Module

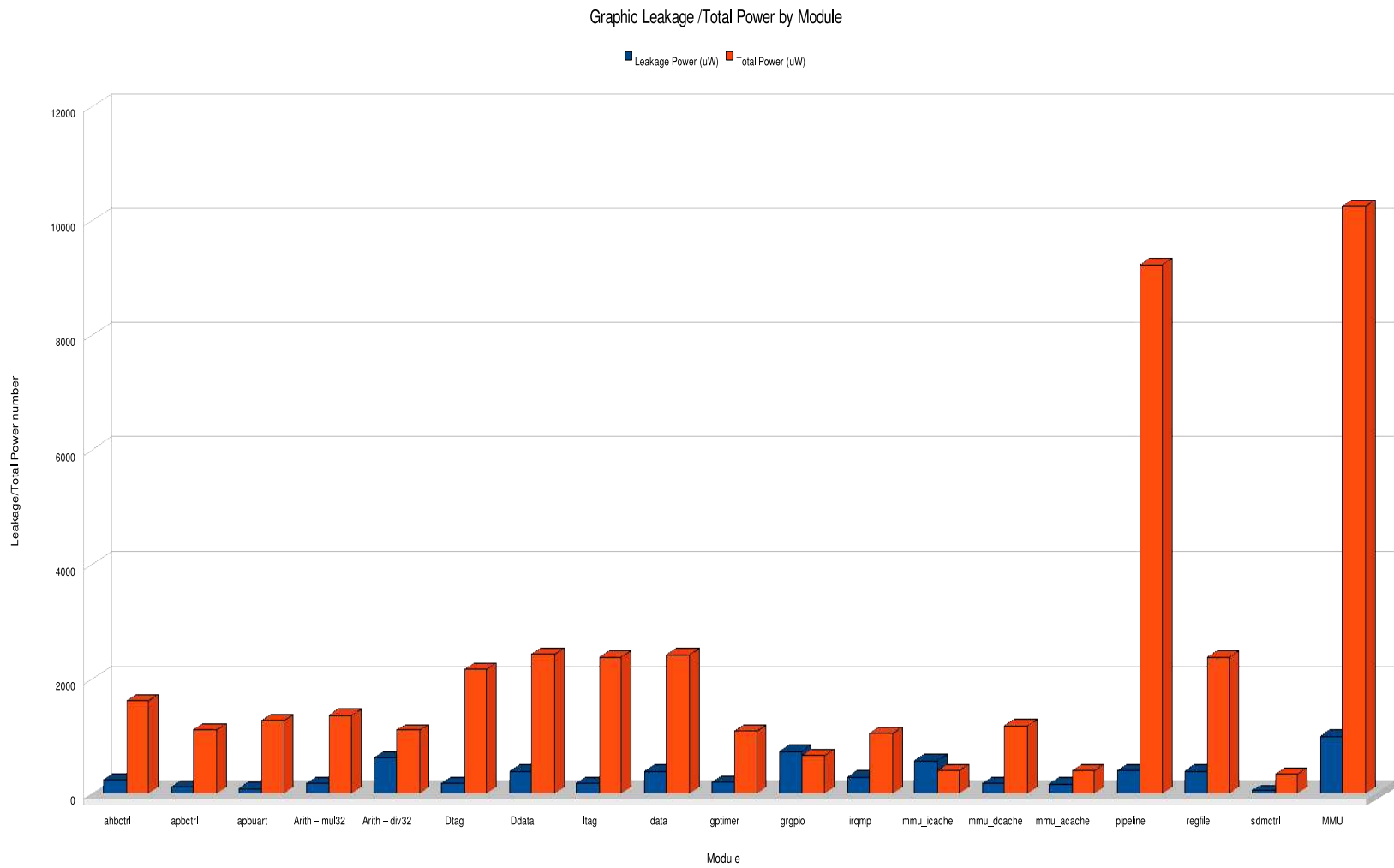


Figura 5.14: Baseline at 100MHz - Total Power/Leakage Power by Module

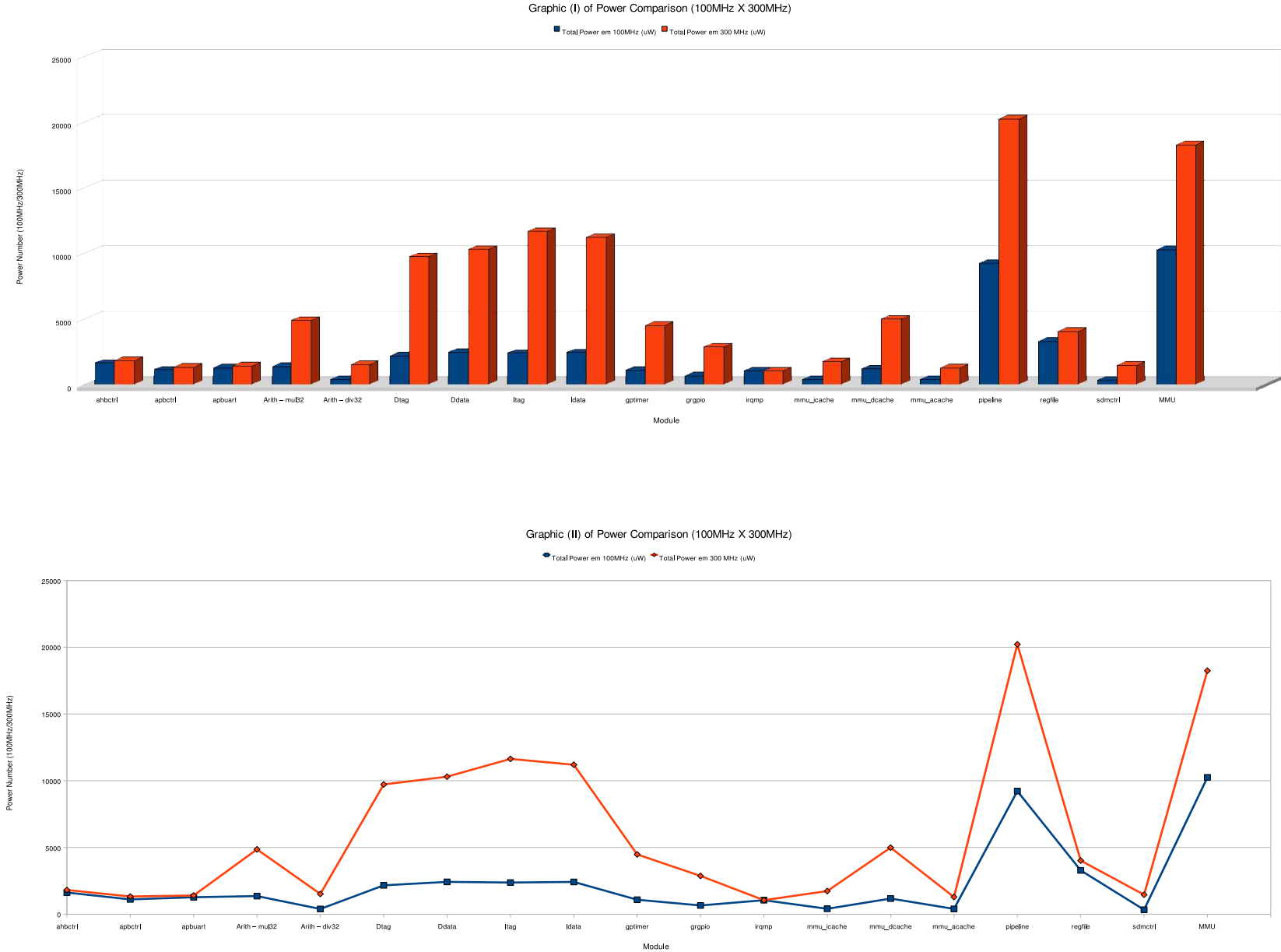


Figura 5.15: Baseline Power Comparison(100MHz X 300MHz)

- Figure 5.11 shows the leakage power of each module at baseline flow. In the same graphic, we also show the leakage power consumption with respect to the total power consumption. The experiment revealed that the leakage power was not relevant, and just represented 4% of the total power dissipation. It was important to remember that, in the worst temperature simulation model (-40°C at $V_{dd} = 0.77\text{V}$), the leakage power could be about 5 times worse.
- Figure 5.12 shows the absolute value of the total power consumption and gate number in each module. The most power intensive modules are respectively: the **Pipeline**, the **MMU** and the **Cache System**, which is very different from the sequence of gate numbers: **Cache System**, **Register File** and **MMU** unit. This demonstrates that the module which presents a larger area not necessary demands a larger power. It depends strongly on the switching activity.

To understand more precisely the relationship between the gate number and the power consumption at each module, we computed the power-gate ratio (dividing the power consumption by the gate number) and put their results in the Figure 5.13.

It is very clear that the "hottest" modules are respectively: **Generic I/O ports**($1.15\mu\text{W/gate}$), **SDRAM Control Unit**($1.08\mu\text{W/gate}$), **Interrupt Request Unit**($1.03\mu\text{W/gate}$), **Pipeline**($0.95\mu\text{W/gate}$), **Apbuart**($0.92\mu\text{W/gate}$), **32 bits Divider**($0.87\mu\text{W/gate}$). This analysis is an important guide to help our optimization strategy and deciding what are the appropriate techniques to reducing the power consumption.

- Table 5.5 shows results when the baseline design was simulated at 100MHz. Without changing in any physical feature of the design, by just cutting the clock speed down to 100MHz, we see a significant reduction in the total power. The power was reduced from 1.56 mW down to 0.558 mW, a 35% reduction factor. As the power dissipation of each gate is reduced, the power rails, the interconnection and the parasitic element also demand less power compared with the design at 300MHz.
- Figure 5.15 shows the power consumption of the baseline flow at 100MHz and 300MHz. We see several modules are benefited by the change of clock speed. They are respectively **Pipeline**, **Cache System** and **MMU unit**. Although the dynamic frequency scaling was not part of this project, these number show that if it was implemented, the dynamic power of those modules would be considerably reduced.
- Figure 5.14 shows the leakage power consumption at 100MHz. There was no significant changes at all the modules and their percentage was raised to 11.31% of

the total power dissipation. This shows that leakage power cannot benefit from the frequency changes, therefore, other power reduction strategy must be taken.

5.6 Clock Gating Optimization

After we shown the baseline results in the section 5.5.12, we saw that the clock speed was decisive in dynamic power reduction, specially for those modules with huge amount of flip-flops (e.g MMU, Pipeline). The dynamic power reduction by frequency scaling is based on the sacrifice of performance, which is not always allowed by the specification.

Based on the theory of clock gating (section 2.3.1), we implemented this technique and analyzed his efficiency by brig up several parameters and comparing them with the results of the baseline flow.

Next, we describe the implementation of clock gating according to the sequence: logic synthesis, DFT structure, gate level co-simulation, placement and routing, STA, parasitic extraction, post-layout co-simulation and finally the power estimation.

5.6.1 Logic Synthesis

The logic synthesis was performed directly without the RTL co-simulation, as there was no modification in the logical functionality. The physical, the timing and the power data from the TCBN90LP were loaded into the synthesizer and the automatic clock gating insertion was performed during the logic synthesis.

One of important tasks to implement this technique was choosing the correct clock gating cells (the type of integrated clock-gating cell determines how a glitch on the enable signal is handled).

There were no special issues during the flow, and the synthesis reported that the inserted clock gating circuit demanded extra area on the final layout, when compared with the baseline flow. This step demanded about 1 heure, close to the duration of baseline flow.

5.6.2 Design for Test

The DFT structure was responsible for adding extra circuit to "control" and "observe" the clock-gated cells (see reference in 2.3.1 for controllability and observability). Depending on the type of clock-gating logic, the gated clock net could no longer be controlled during the post-manufacturing test, resulting in a reduced fault coverage rate of the design. Also, the enable signal driving the control logic could no longer be observed.

In our case, the DFT structure was a big issue during the synthesis flow. The test coverage was seriously impacted (56% in the first run), and several runs accompanied by the ATPG simulations were required to detect the critical failings. The synthesis flow, then, was broken into 4 phases:

1. Clock Gating cells are selected from library, with control signal specification.
2. Construction of the **Observability Logic** for the clock gating cells.
3. Insertion of the **Observability Logic** for the clock gating cells.
4. Incremental mapping into the final netlist.

We successfully rose the coverage rate up to 93.78%, and this process demanded almost 5 work days to be completed since the first run. We detected all the failing points and finally fixed the coverage rate. The ATPG simulation vectors were also generated to be used in post-manufacturing test.

5.6.3 Gate Level Co-Simulation

As in the case of baseline simulation, the post-synthesis co-simulation was fundamental to ensure the logical functionality was not changed by the synthesizer. The clock path and the timing slack violations were potential factor to fail the simulation. In our flow, this steps has took about 250 hours, slightly longer then the baseline flow.

5.6.4 Placement and Routing

As mentioned in the section 2.3.1, most of the back-end tools present a wirelength overhead and cell overlappings after the clock gating insertion. The appropriate placer and router should be clock gating aware during the construction of clock tree. It should perform a non-overlap insertion and a zero-skew routing.

The **SoC Encounter** platform of the **Cadence Design System** was adopted to complete this task. The entire process took about 6.5-8 hours, and we eliminated the DRC³⁶ violation and fixed the antenna effects.

5.6.5 Static Timing Analysis

Several runs of STA were required to eliminate numerous negative skew caused by the setup timing violations. After we detected that the skew problem could not be

³⁶Design Rule Check

solved by the timing optimization engine, we repeated the placement and route phases and pushed the integrated clock gating cell (ICG) as close as possible to the flip-flops to reduce the skew. The timing debug process took about 5 days to fix all the timing violations.

5.6.6 Parasitic Extraction

The parasitic element was extracted without issues after the STA. We successfully outputted the parasitic elements into the SPEF³⁷ format (20 minutes). As in the baseline flow, the SPEF was used by the power engine to estimate the final power consumption.

5.6.7 Post Layout Co-Simulation

The adapted testbench loaded the netlist of the layout and the Standard Delay File (SDF) to perform the post-layout co-simulation. The precise TCF file was extracted by the end of the simulation. This process took about 250 hours to decode 10 frames, similarly as the baseline flow.

5.6.8 Power Estimation

The **Voltage Storm of Cadence Design System** was used to estimate the power. As in the case of the baseline flow, we loaded the SPEF, the power views of memories, the clock data, the post-layout netlist with Toggle Count File (TCF) to estimate the power of the flow.

5.6.9 Experimental Results

- The simulation results can be found in Table 5.6. It contains several design parameters for each module and accompanied by the power values. We can see, the total power was reduced in about 20% when compared with the baseline flow.

The layout area was almost the same as in the baseline flow and the leakage power was reduced in about 6.75%. The reduction in the leakage power (4.71% of total power) didn't impact the total amount of dissipation (the dynamic power still dominant).

- In the first run of clock gating, all the modules reveled an area increase except the macrocells. Trying to investigate these variations, we detected that the design was about 30% larger when the clock gating circuits were inserted. To optimized the

³⁷Standard Parasitic Extraction File

Module Name	Area	Internal	Switching	Leak	Clock Pwer(%)	Total Pwer	Total Gates	Pwer/Gate Coef
ahbctrl	18806.9	1170	323.1	218.3	494.2	1711	7774	0.22
apbctrl	10759	601.5	101.2	120.7	272.2	823.4	4447	0.19
apbuart	4124.7	709.1	201.4	92.5	210.63	1003	1705	0.59
mul32	11198.5	2869	678.7	178.9	812.9	3726	4629	0.8
div32	5791.6	1088	59.29	759.5	473.3	1223	2394	0.51
Dtag	54374.68	9551.61		170.04	x	9721.65	22480	0.43
Ddata	121261.77	9915.89		387.04	x	10302.94	50133	0.21
Itag	54374.68	11465.27		170.04	x	11635.3	22480	0.52
Idata	121261.77	10810.71		387.04	x	11197.75	50133	0.22
gptimer	12425	795.1	97.58	166.3	605.7	1285	6136	0.21
grgpio	5952.8	520.9	61.42	781.8	4634	1456	2460	0.59
irqmp	2687.7	926.6	95.33	326.2	414.7	869.7	1111	0.78
mmu_icache	4790.8	301.8	47.26	628.6	445.1	1150	1980	0.58
mmu_dcache	14144	859.9	144.6	164.6	763.3	2883	5846	0.49
mmu_acache	9280.1	136.9	96.71	100.4	464.1	1049	3836	0.27
pipeline	34242.2	18150	3664	407.2	1759.2	11728	14154	0.83
regfile	79119.8	2891		380.1	x	4018	38704	0.1
sdmctrl	3769.1	258.2	38.01	49.16	510.6	1336	1558	0.86
mmu	61455.5	1631.94		991.52	6567.8	15220	26726	0.57
Total with Layout	925732.89	99630	20276.98	5922.38	7648.1	125726.61	311686	0.4

Tabela 5.6: Clock Gating Power Result at 300MHz

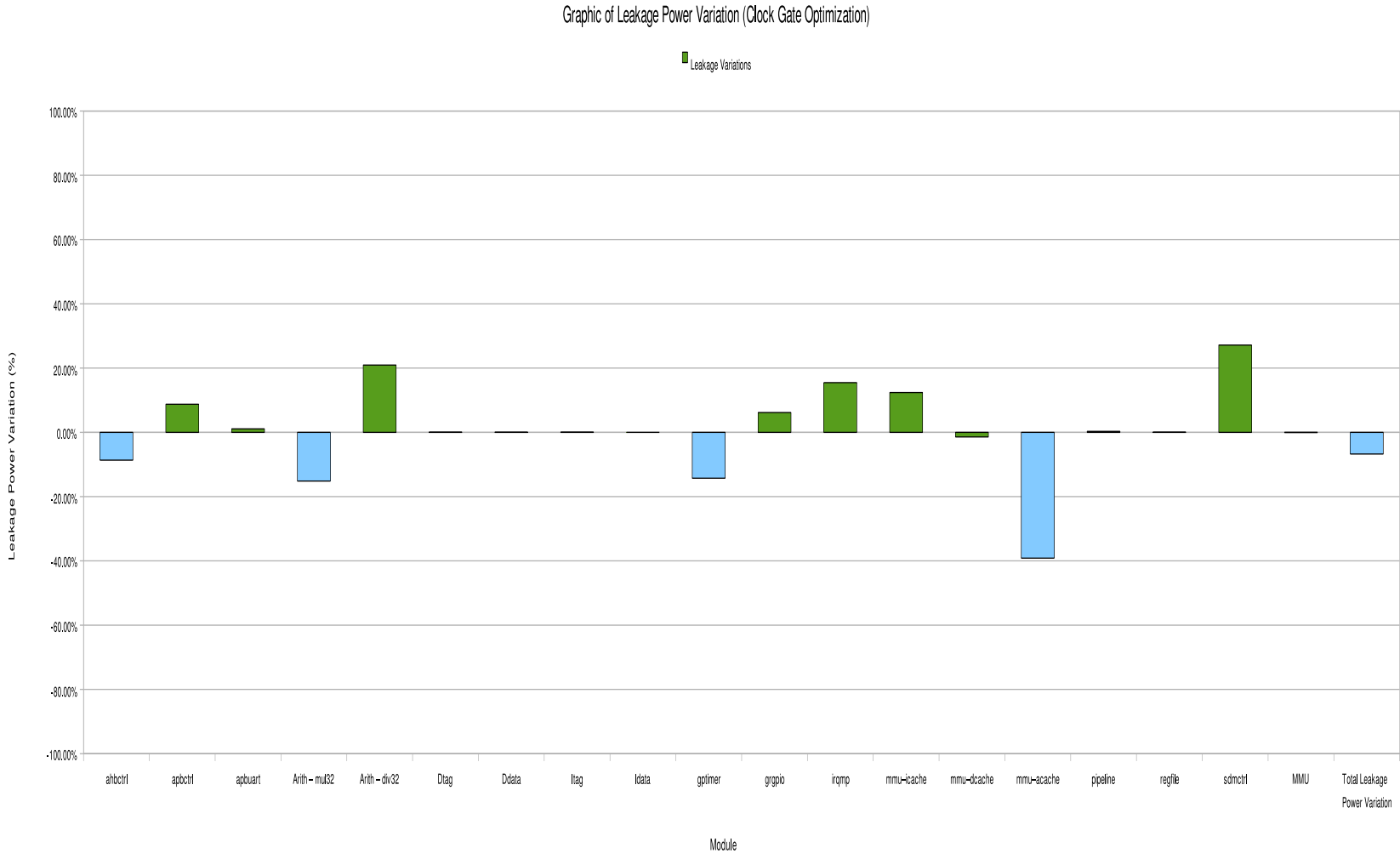


Figura 5.16: Leakage Power Variation of Clock Gating Optimization

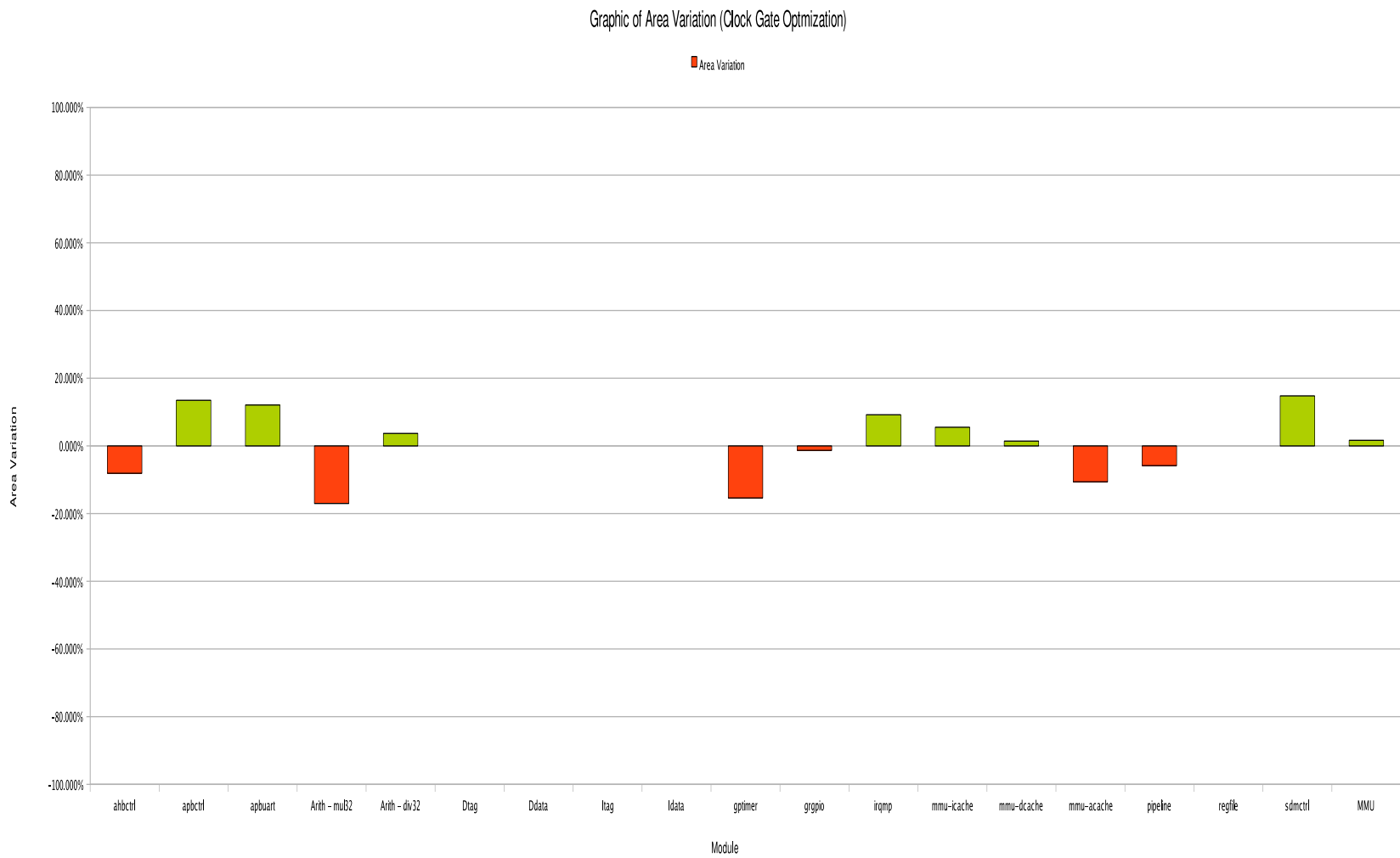


Figure 5.17: Area Variation of Clock Gating Optimization

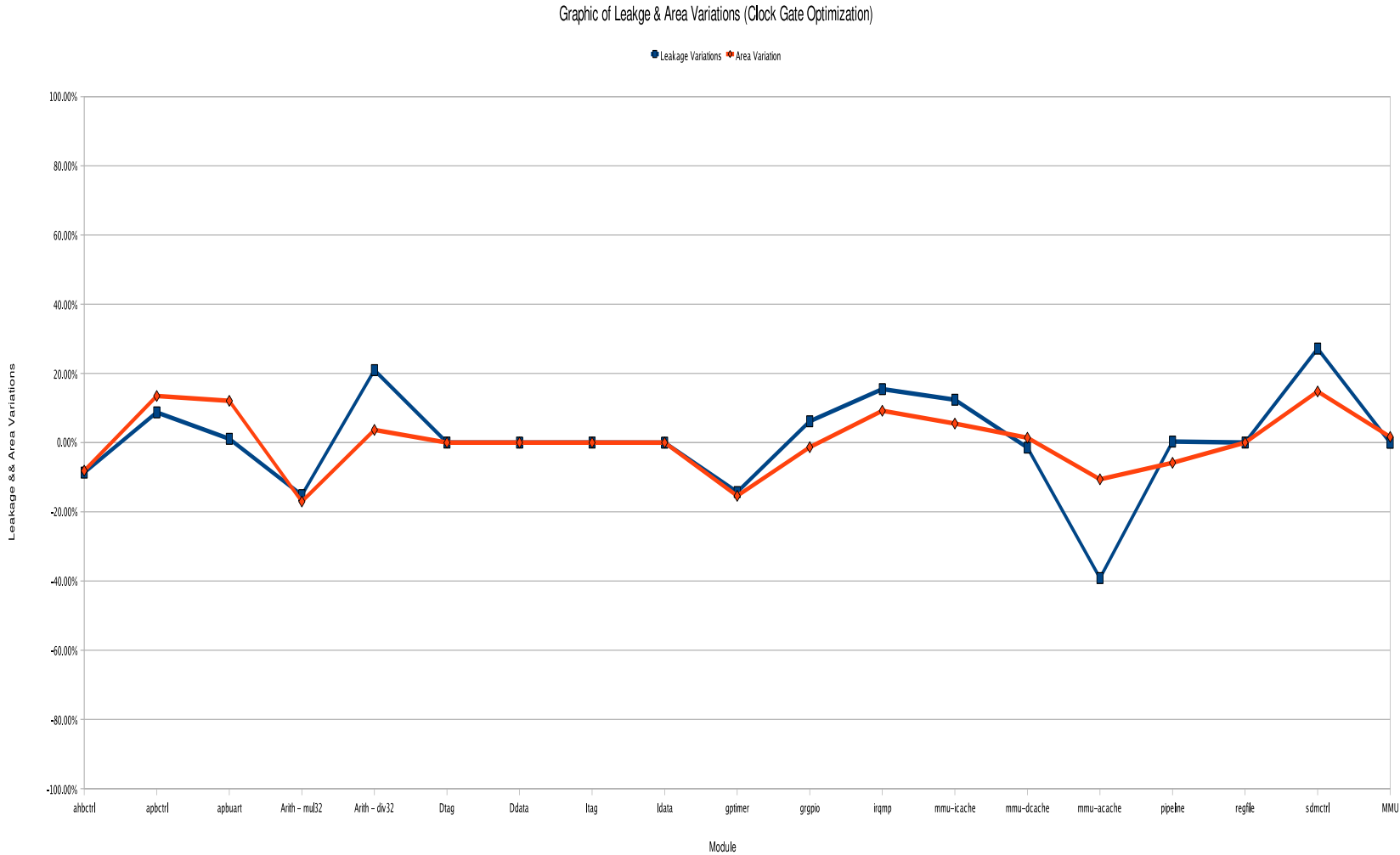


Figura 5.18: Area and Leakage Power Variation of Clock Gating Optimization

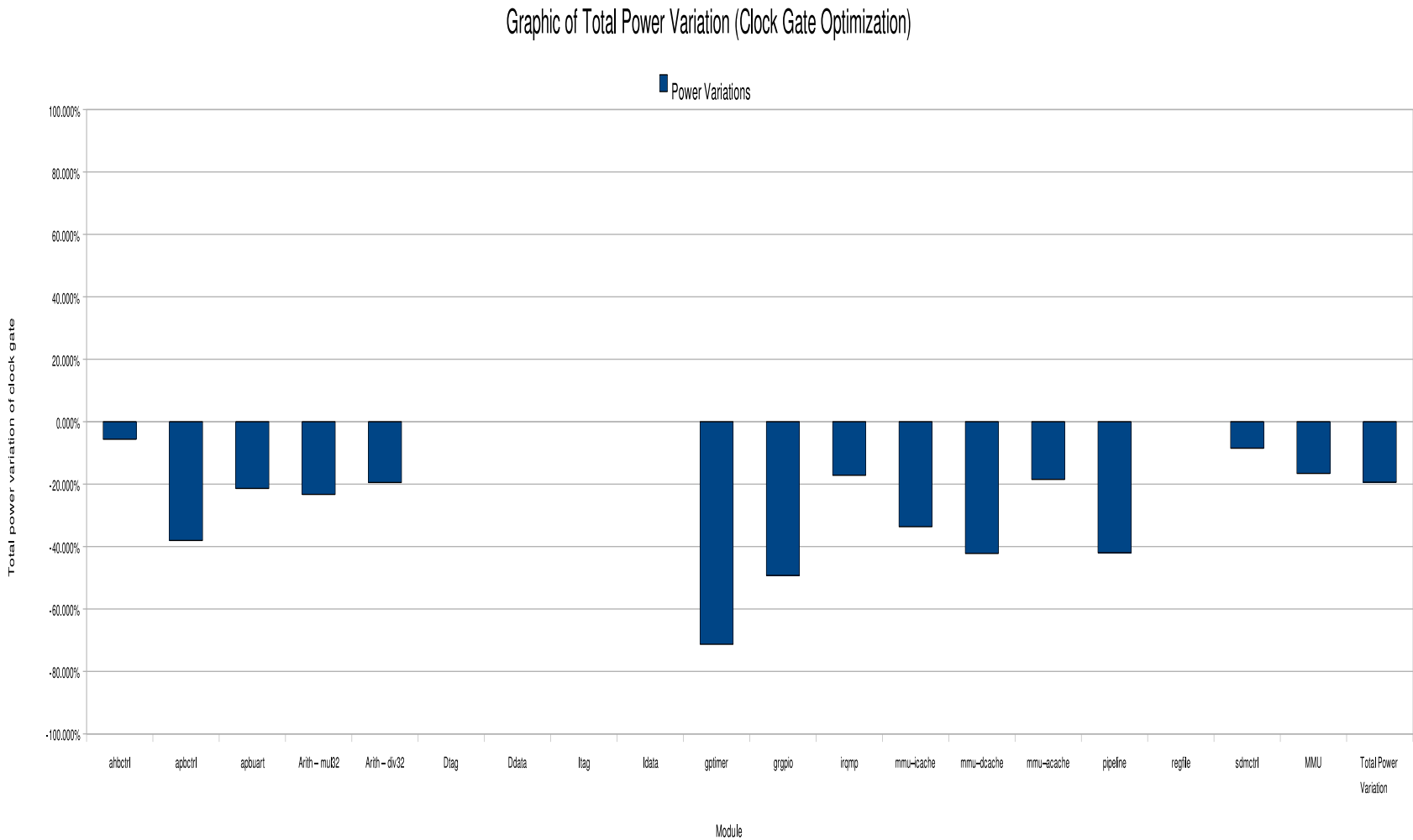


Figura 5.19: Total Power Variation of Clock Gating Optimization

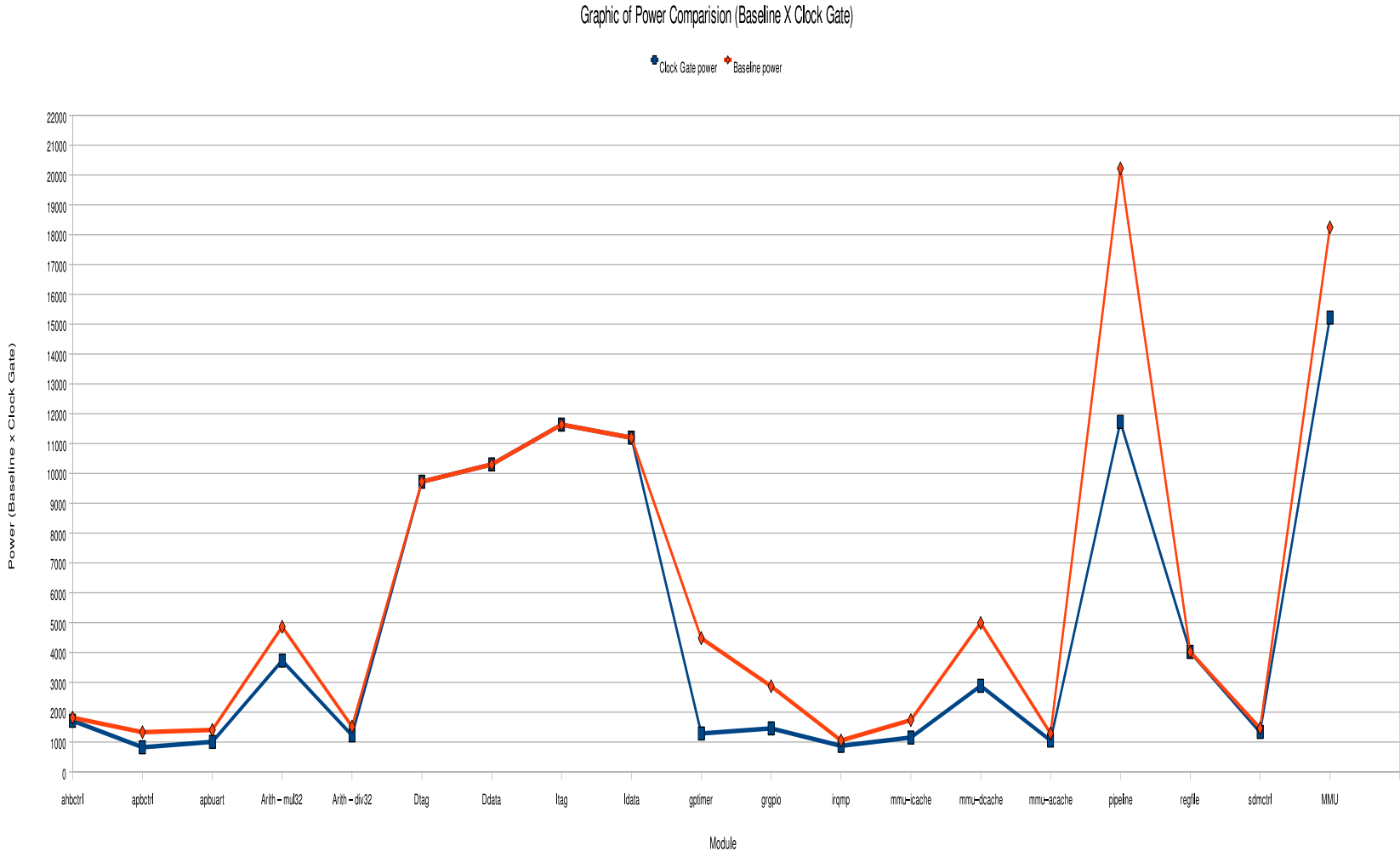


Figura 5.20: Total Power Comparision between Clock Gating and Baseline Flow

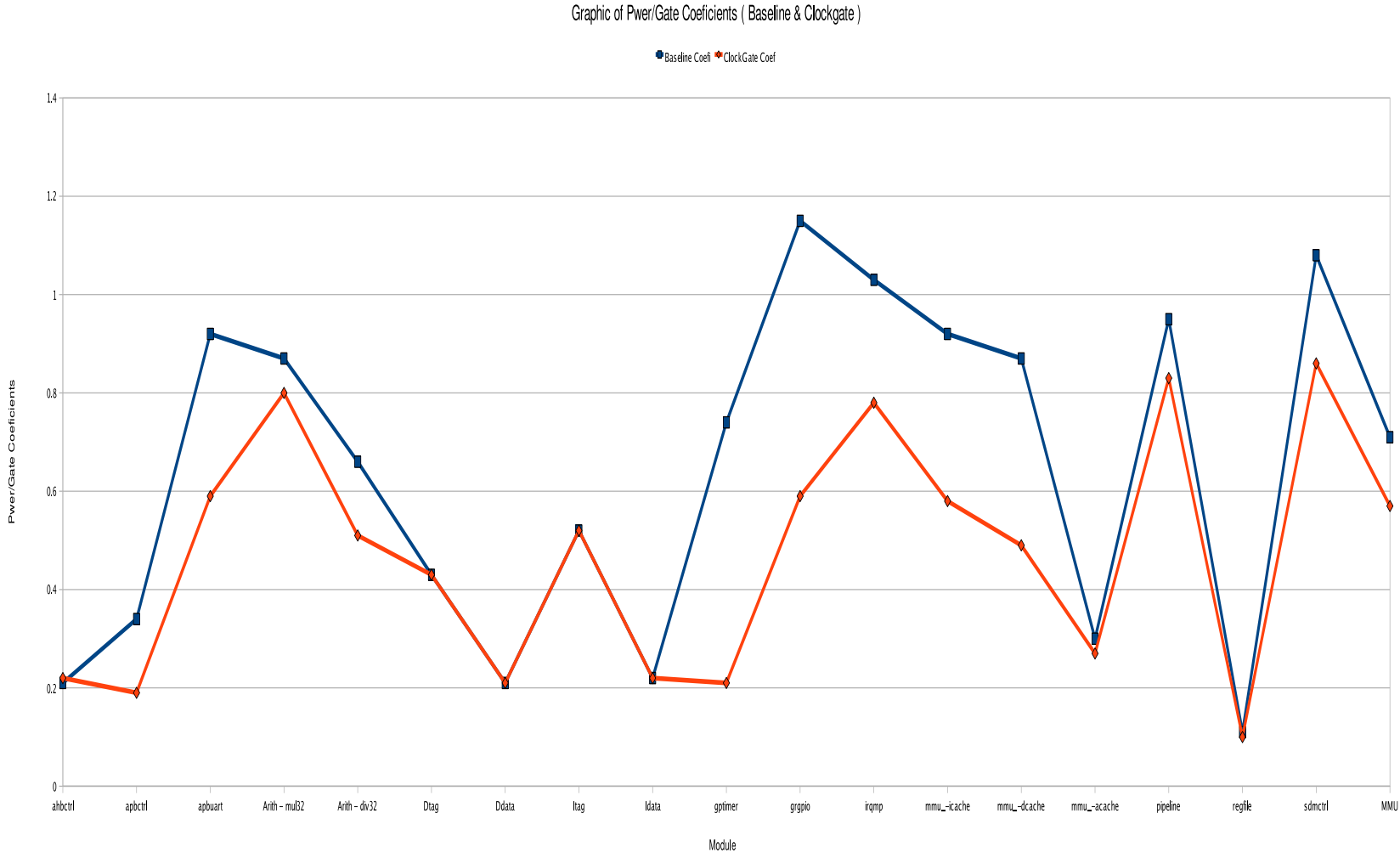


Figure 5.21: Power/Gate Coefficient Variations (Clock Gating and Baseline Flow)

area, we guided the synthesizer to merge all the repeated clock gating cells with the flip-flops (inside the same clock domain), and be controlled by a same enabling signal. This technique is known as **Clock Gating Descloning** and the Figure 5.22 shows this strategy.

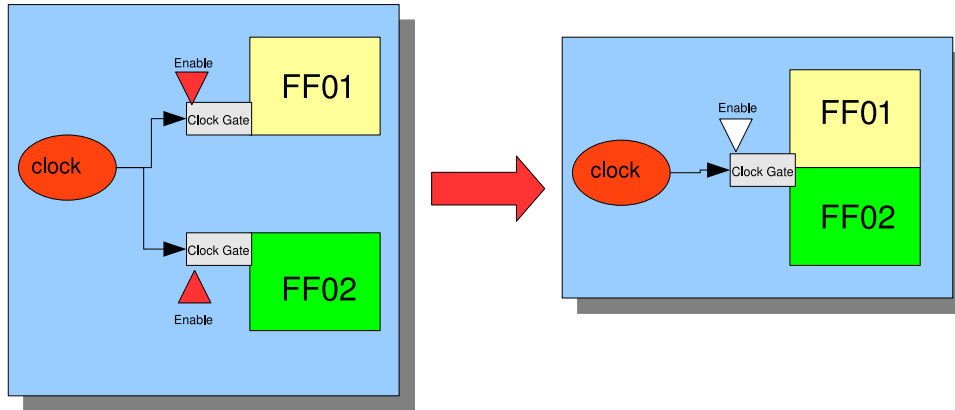


Figura 5.22: Clock Gating Descloning

This approach slightly reduced the silicon area of clock gating circuit, created a balanced clock tree, improved routing congestion and minimized the clock skew problem. Figure 5.17 shows the area variation of each module after the clock gating flow compared with the baseline flow. Notice that the cache system and register file (macro models from foundry) were insensitive to the technique, and area remains constant. The total layout area was 1% larger when compared to the baseline flow.

- Figure 5.16 shows the leakage variation of each module at the clock gating flow when compared with the baseline flow. From the graphic, we see that the cache system and the register files (macrocells from the foundry) were insensitive to the technique and the leakage dissipation remained constant as in the baseline flow. Some modules like the **mmu_acache**, the **gptimer**, the **mul32** showed a significant reduction in the leakage power. Modules like the **sdctrl**, the **div32** and the **irq** presented slight increase in the leakage power.

After we got this result, it seems not easy to interpret why modules showed different variations in the leakage power. In the Figure 5.18, we merged the curve of area variation with the curve of the leakage variation. We achieved a better understanding. Clearly, we see the leakage curve followed the curve of area variation. We concluded that the clock gating contributes in reducing the dynamic power, brings impact on the silicon area and indirectly influences the leakage power in several cases.

- Figure 5.19 shows the percentage of variations of the total power when compared with the baseline flow. We see that several modules were greatly benefited by this technique: **gptimer** (-71.304%), **pipeline** (-41.98%), **grgpio** (-49.25%), **mmu_dcach** (-42.21%).
- Figure 5.20 is a comparison between the total clock gating power with the baseline flow in absolute number. Although the macroblock cells (cache memory and register file) were not benefited neither in dynamic nor static dissipation, it was clear that the total power was reduced in about 19.39% when compared with the baseline flow (155.97 mW down to 125.73 mW, at a 300 MHz clock speed).
- The power variations in absolute number was not enough to show the power efficiency of each module, because it just prove the total difference, but did not consider the number of gates and area of each module.

Figure 5.21 shows the power-gate coefficient variation when compared with the baseline flow. The power that was dissipated by each gate at several modules was reduced after the clock gating. Although the macrocells remained intact as before, according to the curve, we see those modules which had a higher power-gate coefficient were the most benefited by this technique. This showed that the original RTL design wasn't efficient in power and the technique certainly prevented "superfluous" switching activities at clock pins.

5.6.10 Impacts of Clock Gating

Table 5.7 summarizes the main impacts of clock gating on the design. Next, we analyze and describe these impacts in details:

- There was no architecture impact after the application of the technique. The datapath, the transaction protocol, the timing constraint, the logical functionality remained the same.
- The DFT structure and the ATPG vector demanded more engineering effort to be implemented (Demanded about 5x time when compared with the baseline flow).
- The back-end tools were aware of the clock gating cells both in the placement and routing. It was required more effort in Routing, DRC and Antenna fixing when compared with baseline flow (2x times effort when compared with the baseline flow).
- The area impact caused by clock gating was small (insignificant) specially if we consider the improved power reduction (-19%).

Module Name	Area Impact	Total Pwer Impact	Leak Pwer Impact	Speed Impact
ahbctrl	-8.1%	-5.6%	-8.7%	0%
apbctrl	+13.4%	-38.0%	+8.7%	0%
apbuart	+12.0%	-21.4%	+1.1%	0%
div32	-17.0%	-23.3%	-15.2%	0%
mul32	+3.6%	-19.5%	+21.0%	0%
Dtag	0.0%	0.0%	0.03%	0%
Ddata	0.0%	0.0%	0.01%	0%
Itag	0.0%	0.0%	0.02%	0%
Idata	0.0%	0.0%	0.01%	0%
gptimer	-15.4%	-71.3%	-14.3%	0%
grgpio	-1.3%	-49.3%	+6.2%	0%
irqmp	+9.2%	-17.2%	+15.5%	0%
mmu_icache	+5.5%	-33.6%	+12.4%	0%
mmu_dcache	+1.4%	-42.2%	-1.4%	0%
mmu_acache	-10.6%	-18.5%	-39.2%	0%
pipeline	-5.8%	-42.0%	+0.3%	0%
regfile	0.0%	0.0%	+0.03%	0%
sdmctrl	+14.7%	-8.5%	+27.2%	0%
mmu	+1.6%	-16.6%	-0.05%	0%
Total with Layout	+1.0%	-19.4%	-6.8%	0%

Tabela 5.7: Impact Comparison between Baseline Flow and Clock Gating Flow

- The performance remained at 300MHz clock and the STA³⁸ demanded more engineering/timing effort to be done, when compared with the baseline flow.

5.7 Operand Isolation

As explained in section 2.3.2, the Operand Isolation can identify redundant operations and then uses special isolation circuit to prevent switching activity to propagate to downstream modules.

Different from the clock gating which reduces the dynamic power on clock path (sequential logics), this technique eliminates redundant computations in the combinational blocks. In this section, we present analysis implementation of Operand Isolation and explain possible reason for the results.

³⁸Static Timing Analysis

5.7.1 Details of Implementation and Results

In the Operand Isolation, we do not change the system architecture and logical functions, hence, the RTL simulation can be skipped and we go directly to the logic synthesis. The insertion of operand isolation circuit was done in three stages during the logic synthesis:

- Datapath block candidate detection (such as adders and multipliers).
- Isolation circuit insertion.
- Commitment Decision. Deciding if the inserted circuit could introduce any unwelcome consequence (for example, timing violation or routing congestion).

We first performed a code analysis of entire Leon3 system using the **RTL Compiler** and then mapped it into netlist. After finished the logic synthesis task (2.5 hours of duration), we identified that the technique was not proper to our design.

Outraged by this question, we repeatedly perform the synthesis process from module to module, analyzing the corresponding report to investigate the answer. In summary, there were two reasons that prevent the technique to be applied:

1. The enable circuit of operand isolation was inserted onto the timing-critical path, which in turn, cannot satisfy STA.
2. The final power was increased by adding an operand isolation instance to control the datapath blocks.

Most of the modules fit the first case. Just **pipeline** and **mul32** presented power issues after inserting the isolation logic. Since the technique cannot detect any proper datapath candidate to implement the control logic, we just stoped the design flow after analyzing the entire design. From the theoretical explanation of section 2.3.2, this technique should present about 15% of reduction in dynamic power with slight increase in layout area. Unfortunately, it could not be shown in our experiment.

5.8 Multiple Threshold Voltage Optimization

In general, a Low V_{th} library is used to optimize the speed of the critical path to ensure the timing requirements, but demands more leakage power. On the other hand, a high V_{th} library is used for the majority of the non-critical path to reduce the leakage dissipation.

In this flow, we changed the process library into the TCBN90LP (three threshold voltages) and added several optimization steps to perform the multiple-threshold optimization.

Implementation of multiple V_{th} optimization was described according to several topics listed next. Specially, we recall the task of logic synthesis. After analyzed each type of V_{th} library, two strategies were used to perform this task: the mixed V_{th} synthesis and the incremental V_{th} synthesis.

5.8.1 System Specification and Requirement

Our goal here is to reduce the threshold voltage without impacting the performance and to showing how effective is this technique in reducing the leakage power. Therefore, the clock speed was defined at 300MHz as the baseline flow and the area and total power were unconstrained.

5.8.2 Logic Synthesis (Mixed and Incremental Strategy)

The RTL³⁹ co-simulation does not require any physical information to generate switching activity. Hence the TCF file cannot store the switching data covering all the pins and nets. Therefore, the logic synthesis must be performed before any other steps.

After analyzed three types of V_{th} library of the TCBN90LP, we defined two strategies to optimize the logic synthesis, which can be summarized in Figures 5.23(a) and 5.23(b) and their descriptions are next:

Mixed Synthesis

The mixed V_{th} synthesis (Figure 5.23) imported all the libraries with different V_{th} , leaving the synthesizer to find an optimal leakage consumption. This approach demanded less engineering effort but strongly depended on the capability of the logic synthesizer to achieve a better result.

1° Incremental Synthesis

Different from the mixed V_{th} strategy, the incremental synthesis (Figure 5.23(a)) loaded the three V_{th} libraries at different steps with appropriate timing and power constraints.

First, the low V_{th} library was used to map the design into gates and generates a first version of netlist. This netlist was used by the backend tasks (placement, routing

³⁹Register Transfer Level

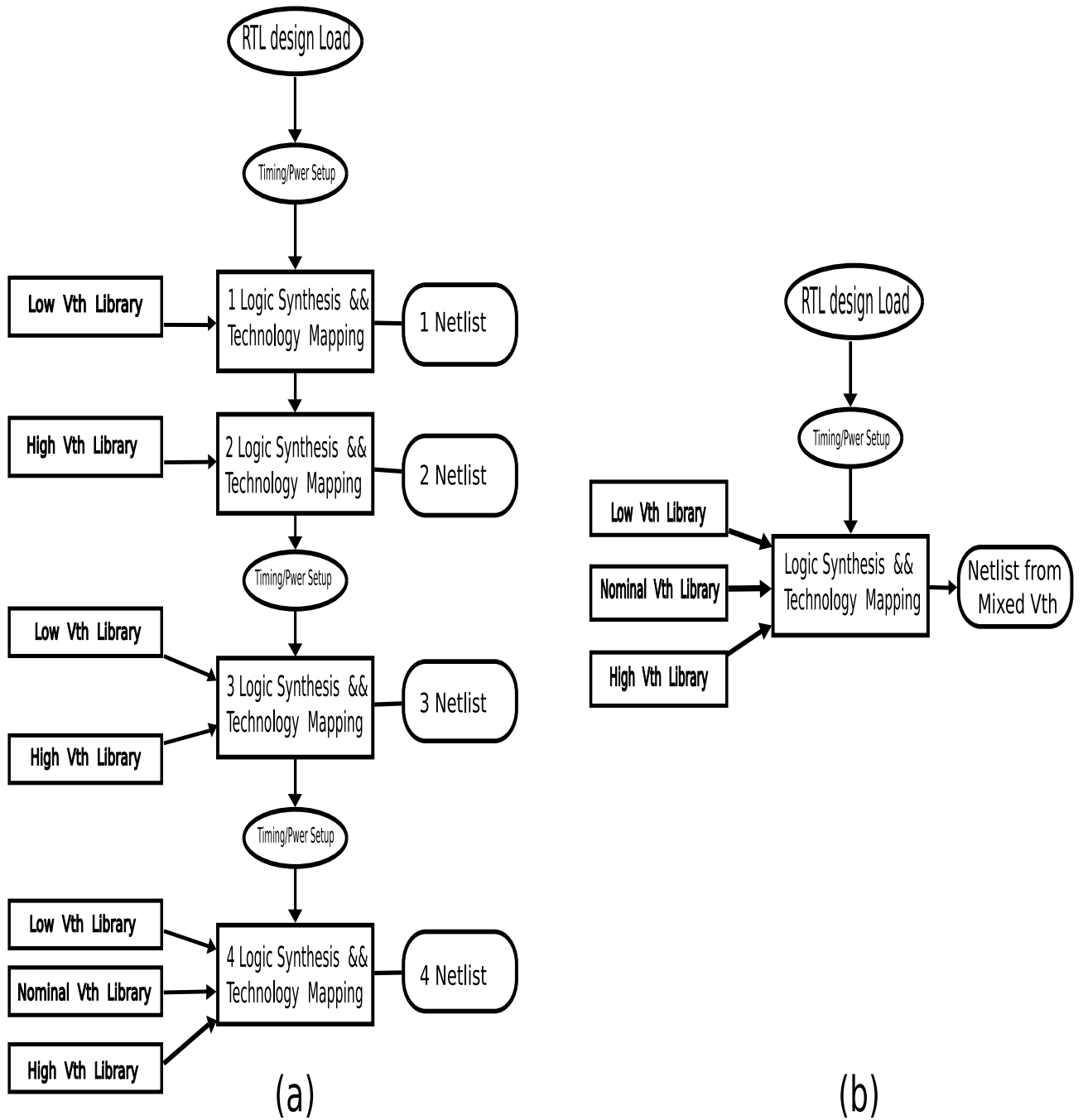


Figura 5.23: (a) Incremental V_{th} Synthesis (b) Mixed V_{th} Synthesis

and STA) and the post-layout simulation. The Dumped TCF file stored the switching activities of all the pins and nets, and fed the power engine in order to estimate the power consumption as in the baseline flow. The result showed that the performance was met but the leakage power was increased in almost 61% when compared with the baseline flow.

2° Incremental Synthesis

Without turning-off the tool, we changed the library to the high V_{th} and updated the timing/power constraints to the appropriate values. The new netlist replaced the low V_{th} cells (resulting from the first incremental synthesis) by the corresponding high V_{th} version. Taking this netlist into the back-end flow and generating the TCF file to estimate power, we noted that the timing could not be met and the leakage power was reduced by almost 68.2%.

3° Incremental Synthesis

We reloaded the low V_{th} library into the synthesis environment and updated the timing and power constraints. The resulting netlist contained cells with two different threshold voltages. The post-layout timing analysis and power estimation showed that the performance constraints was met with a significant reduction in the leakage power, almost -36.67% when compared with baseline flow. The dynamic voltage and layout area were slightly increased.

4° Incremental Synthesis

The netlist from the third incremental synthesis contained the low and the high V_{th} cells. After we updated constraints for both the timing and the power, the nominal V_{th} library was loaded to the synthesis environment to generate the fourth version of netlist.

The synthesizer was oriented to performs incremental mapping, which only improved the gate structure and optimized the leakage power without inducing any performance penalty. After analyzed the resulting netlist, we understood that the design was better because the synthesizer selected low V_{th} cells to fix any possible timing violations and improving leakage power and area with high V_{th} and nominal V_{th} cells (details in 5.8.7).

5.8.3 Design For Test

Since the TCBN90LP library presents scanable flip-flop cells with 3 threshold voltages, the DFT structure was built only after the fourth incremental synthesis. The

ATPG⁴⁰ vector was also generated for the final layout. The simulation of ATPG vector demanded almost the same effort as in the baseline flow.

5.8.4 Static Timing Analysis

A couple of timing issues appeared when only the high V_{th} library was used (the second incremental synthesis). In that case, it took about 5 hours to build the clock tree and several STA runs were required to meet a 250 MHz clock speed (still with timing violations). For the case of 1^o, 2^o and 3^o incremental synthesis, a 300 MHz of clock was met.

5.8.5 Post-Layout Co-Simulation

The post-layout simulation took a similar duration as in the baseline flow. It took about 240 hours to validate 1 netlist of each incremental synthesis, using the same hardware configuration of the baseline flow.

5.8.6 Backend Tasks

The placement, the routing, the parasitic extraction, the equivalence check were done as in the baseline flow. No extra design efforts were required.

5.8.7 Experimental Results and Impacts

The incremental synthesis loaded different V_{th} libraries at different steps of the flow. After analyzed and compared all the results, the netlist from the fourth incremental synthesis presented a better power efficiency with less area impacts. In this section, we compare the results of the mixed/incremental synthesis with the baseline flow, followed by the detailed analysis.

In the tables 5.8, 5.9, 5.10, 5.11 several design parameters were used to show the results and their meanings were listed in next:

- Macrocell Area: denotes the area of macrocells (**Register File** and **Cache Memories**).
- Comb Area: denotes the area of the combinational logic in the layout.
- Sequential Area: denote area of sequential logic in the layout.

⁴⁰Automated Test Pattern Generation

- Total Area: denote the total layout area after each flow.
- Performance: denote the clock speed achieved by the design after the layout.
- WNS: denote the worst negative slack, indicating the path with timing violations.
- Gate Number: denote the number of gates after the DRC and STA violation were fixed.
- N_{vt} , H_{vt} , L_{vt} percent: denotes the percentage of cells with Nominal/High/Low leakage after the layout was done.
- Dynamic: denotes the percentage of the dynamic power with respect to the total power.
- Leakage: denote the percentage of the static power with respect to the total power.
- Total power: denote the total power consumed by the final layout.

Results of Mixed Synthesis

Table 5.8 summarize the result of the mixed synthesis strategy. The corresponding power distribution is listed in table 5.9. All the power parameters are in μW .

Design Parameter	Leon3 System
Macrocell Area	37.9%
Comb Area	26.1%
Sequential Area	35.4%
Total Area	925731
Performance	300MHz
WNS	Meet
Gate Number	299158
N_{vt} percent	9.9%
H_{vt} percent	68.0%
L_{vt} percent	21.6%
Dynamic	149490
Leakage	3860
Total Power	153350

Tabela 5.8: Result of the Mixed Synthesis

By comparing the results of table 5.8 with the baseline flow, we see that the total power was reduced by 1.6% (155969 μW down to 153350 μW), the dynamic power

Group	Internal	Switch	Leakage	Total Pwer	Percent
Sequential	64450	3971	1175	69596	45.38%
Macrocells	42390	766.3	1494	44650	29.12%
Combinational	14560	23354	1191	39105	25.50%
Total Power	121400	28090	3860	153350	100%

Tabela 5.9: Power distribution of the Mixed Synthesis

was reduced by about 0.44% (150150 μW down to 149490 μW), while the leakage power was reduced around 38.87% (6315.26 μW down to 3860 μW). The final layout presents 67.91% of high leakage cells, 9.81% of nominal cells and 21.56% of low leakage cells.

The total area was increased in about 0.98% (916567.22 mm^2 to 925730.91 mm^2) when compared with the baseline flow at same speed (300MHz, without worst negative slack). The macrocell area did not not change, since they are foundry models with a fixed footprints (37.94% in the layout area). The total gate number was reduced by 0.992% (302158 down to 299158 gates) compared with the baseline flow.

The largest change happened in the leakage power, a reduction of 38.87% compared with the baseline flow. All the other parameters remained almost intact. These results revel a quick and effective way to reduce the leakage dissipation without many extra effort or performance/area impact.

We also see, as this technique depends strongly on the process library and the EDA tools, the RTL implementation cannot reduce any leakage power.

Results of Incremental Synthesis

Table 5.10 summarizes the results of all the incremental synthesis, whose netlist contains the nominal, the high and the low V_{th} cells. The power distribution is shown in the Table 5.11, separated by the combinational cells, the sequential cells and the macrocells cells.

Comparing the results of Table 5.10 with the baseline flow, we see the total power was reduced in about 3.7% (155969 μW down to 150231.3 μW) and the dynamic power was reduced in about 2.4% (150150 μW down to 146564.3 μW). The leakage power shows a significant reduction by about 46.7% (6315.26 μW to 3367 μW).

The total area was reduced in about 1.2% (916567.22 mm^2 down to 905730.91 mm^2) when compared with baseline flow at same speed (300MHz, without worst negative slack). The macrocell area remained intact (37.94% as the baseline). The total gate number was reduced in about 5% (302158 down to 297197 gates) compared with the baseline flow.

Without sacrificing the performance (clock at 300MHz), the result of incremental

Design Parameter	
Macrocell Area	37.94%
Comb Area	25.29%
Sequential Area	36.17%
Total Area	905730.91
Performance	300 MHz
WNS	Meet
Gate Number	287158
N_{vt} Lib	10.1%
H_{vt} Lib	26.78%
L_{vt} Lib	63.12%
Dynamic	146564.3
Leakage	3667
Total Power	150231.3

Tabela 5.10: Result of the Incremental Synthesis

Group	Internal	Switch	Leakage	Total Pwer	Percent
Sequential	64202	3771	1005	68978	46.51%
Macrocells	41219	761.3	1481	43461.3	29.14%
Combinational	14797	21814	1181	37792	24.11%
Total Power	120218	26346.3	3667	150231.3	100%

Tabela 5.11: Power distribution of the Incremental Synthesis

synthesis shows a better results in terms of the area and the power consumption. This method strongly depends on the process library and requires rigorous timing analysis to warranty that the design was free of timing violations. In this case, the DRC, the LVS, the DFT and the power analysis were not issues, and could be done as in the baseline flow.

5.9 Multiple Supply Voltage Optimization

This section explains the implementation and the experimental results of multiple V_{dd} optimization. We list all the relevant design parameters and analyze the impacts and improvements when this approach was applied.

5.9.1 System Specification and Requirement

As showed in section 2.3.4, the multiple V_{dd} optimization requires level shifters, moreover the PDK must be characterized for multiple supply voltages. The basic li-

brary analysis (Table 5.2) showed that the PDK TCBN90LP was characterized for 0.84V, 0.924V, 0.77V and 1.32V beside the nominal V_{dd} (1.2V).

Hence the power structure can be divided in several power domains, and the clock speed was defined in 300 MHz as in the baseline flow. We left the area and the power unconstrained to compare the results with other techniques.

5.9.2 Library Qualification of Level Shifters

From the Table 5.3, we see that several level shifters were present in the TCBN90LP. They basically enable signals to pass between blocks with different power rails (different V_{dd}). Such level shifter are analog blocks, built as **voltage buffer** that translates signals from one voltage swing to another.

The need of level shifter is obvious when we try to drive signals between domains which are supplied by radically distinct voltages. However, such shifters are also indispensable even when two voltage domains showing slight voltage variation. (case of TCBN90LP, 0.924V and 0.77V domains)

One fundamental reason is because both the NMOS and PMOS network are turned on when a 0.77V signal drives a 0.924V gate. When this happens, excessive crowbar currents show up and certainly lead to timing closure problem.

The best solution to this problem is providing level shifters between any domains that use different voltages. This approach limits any voltage swing and timing issues to the boundary of the voltage domains, leaving the timing unaffected. With this kind of interface, we have an easier approach to meet timing and enable the reuse when the voltage domains are different.

Next, we cite two different level shifters of the TCBN90LP library and explain their functionality.

High-to-Low Voltage Level Shifter

As shown in Figure 5.24, the High-to-Low level shifters are quite simple: essentially two inverters in series. The circuit of 5.24 only requires a single power rail from the lower power domain. Since there were two inverters in series, the High-to-Low level shifter only introduce a buffer delay, little impact on timing.

Low-to-High Voltage Level Shifter

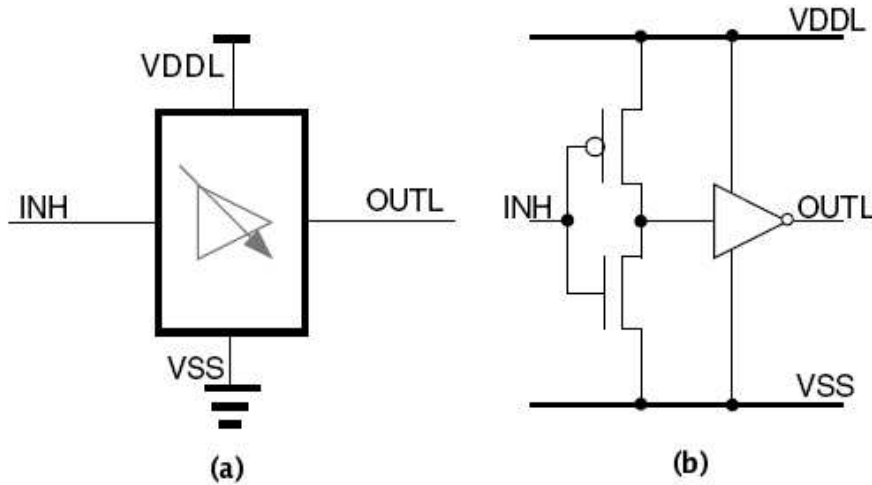


Figura 5.24: High-to-Low Level Shifter

Signals which go from a low ($VDDL$) to a higher voltage rail ($VDDH$) show more critical problem. An under-driven signal degrades the rise and fall times at the receiving inputs, which can lead to higher switching currents and reduced noise margins.

A Low-to-High level shifter is shown in Figure 5.25. It can buffer and invert a lower voltage signal (INL input) and uses the same to drive a cross-coupled transistor structure at the higher voltage. In this case, the input signal (INL) can be translated into a higher voltage ($OUTH$) and connected to the high voltage domain.

These cells were characterized over an extended voltage range to match the operating points of both high and low voltage domains, which enables an accurate static timing analysis between different voltages and operating conditions.

The Low-to-High level shifters introduce a significant delay when compared to the High-to-Low level shifters. This happened with the case of wide interfaces between the timing critical blocks. For example, between the SparcV8 core and the cache memory.

We successfully identified all the necessary corner characterizations (timing/power) of the level shifters, after a test run was done using the TBCN90LP library. Another test runs was done to validate the integration of EDA tools with the cells.

5.9.3 Power Architecture and Power Domain Definition

After analyzed the result of baseline flow (Table 5.4), we see that the total power consumption can be divided into 3 parts:

- Memory cells dissipation: the cache system was made based on the foundry memory

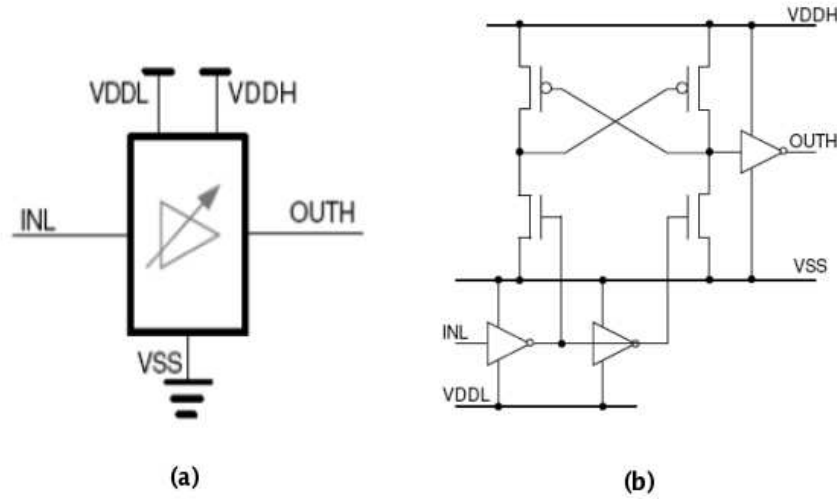


Figura 5.25: Low-to-High Level Shifter

models, which present a fixed power supply and leakage dissipation. They were responsible for about 30.05% of power consumption at 300MHz clock speed.

- Sparc V8 Core dissipation: without considering the register file, the Sparc V8 core contains div32, mul32, mmu_icahe, mmu_dcahe, mmu_acache, pipeline and the mmu. It was responsible for about 46.87% of total power consumption after the layout is done.
- Bus and peripheral components' dissipation: the ahb controller, the apb controller, the sdram memory controller, the interrupt controller, the apbuart unit, the gptimer, the grgpio are responsible for about 23.08% of total power after the layout was finished.

From the documentation of memory models, we know that the voltage supply of memories must be fixed in the nominal V_{dd} , which is 1.1V for the TCBN90LP node. Hence means all the cache system and the register file must be in the same power domain to keep the timing performance without causing impacts.

To keep the Sparc V8 core to follow the register file response, we scaled the voltage supply down to 0.924V and the remaining part of the design was 0.77V. Figure 5.26 summarizes this approach.

After the definition of the power domains, we built the power intent format, defining all the necessary level shifters to allow the signal passes through each regions without severe impacts.

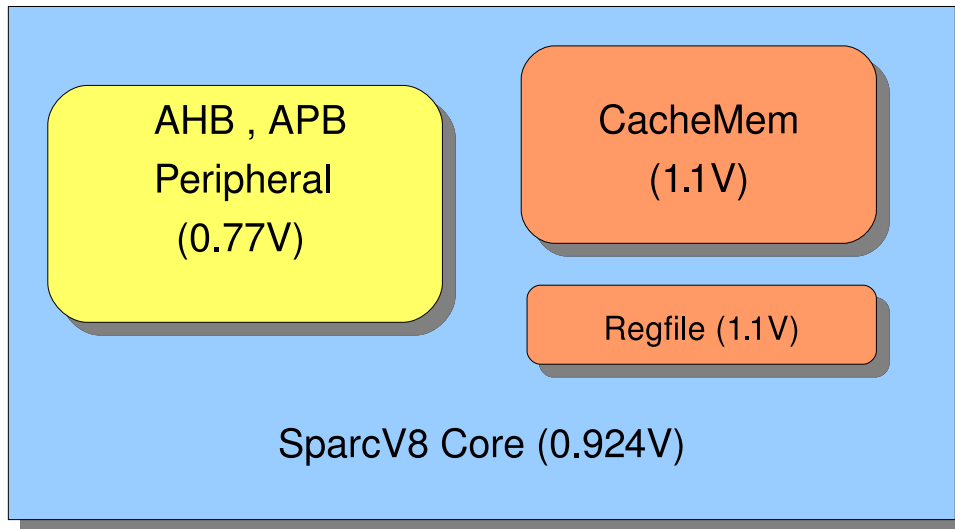


Figura 5.26: Power Domain of Leon3 System

5.9.4 Register Transfer Level Co-Simulation

The RTL simulation cannot be used to validate the timing impact derived from the the power supply changes. We just input the power intent format and simulate the logic functionality of all the modules when they were grouped in each voltage region. This step was used to ensure that the EDA tool can capture the power intent and correctly divided the design according to each voltage domain.

5.9.5 Logic Synthesis

In logic synthesis, we specified the level shifters to generate a mapped netlist. The synthesizer used follower steps to finish the technology mapping:

- Library domains creation in the design hierarchy.
- The level shifter were grouped according to the power intent specifications.
- Specify the input and the output voltage ranges, the direction, the valid locations for each type of level shifter.

The logic synthesis has took about 2 hours to be concluded and the back-end data was generated according to the power intent information.

5.9.6 Power Rail Planning and Power Grids

According to the power architecture and voltage domains (Figure 5.26), we know the physical design demands an extra-structure to support the multiple supply and requires more effort in the floorplanning. In such case, the structure of power grids become more complex.

After all the standard cells and macrocells were loaded into the design framework (netlist, LEF view, lib view, power intent files), we must consider how to provide power to the various voltage areas. By definition, each power domain employs a different power strategy and this results applying different power rails for each voltage region.

After many tentative to build the power network manually, we saw the a high voltage drop across each power domain, and impacted deeply the timing performance between the CPU core and the cache. After investigation, we decided to use the **Power Network Synthesizer** from the **Cadence SoC Encounter** framework to assist this step.

Power network synthesis allowed us to specify the absolute constraints on the power plan (such as maximum voltage drop, routing layers and via requirements) and ensures that the power budget for the design is satisfied. In the interaction with the tools, we successfully reduced the IR drop down to a tolerable range without timing impact.

Next, we summarize several key point of the power network implementation:

- It is fundamental to have a power synthesis tool which supports power budget calculations before placement and routing.
- It was required to run both the static and the dynamic IR drop analysis on the design to verify the power rail integrity. Significant static IR drops usually require increasing the number of level shifter cells and we must adjust their position at each power domain.
- Dynamic IR drop violations were fixed by inserting the decoupling capacitors after the placement and routing. Transient analysis was used to assure that all IR drop violations were fixed.
- In case of excessive IR drop violations, the power intent format must be reviewed and another logic synthesis must be done to generate a new division for the power domains.
- Larger width of power rings was inserted around the memories to minimize the IR drop and we made the core domain closer to the GND and the VDD pins of each memory block.

5.9.7 Placement and Routing

By observing severe IR drop issue during the power network synthesis, we pay a special attention in analysing the interface delays and the physical routing constraints when signals across the voltage boundary (specially between the Sparc V8 core and the memory blocks).

In next, we show three typical placement case of the level shifters between several power domains.

Placement of High-to-Low Level Shifter in the Destine Domain

The first placement issue happened when we must place a High-to-Low shifter crossing a third power domain. Figure 5.27 shows one typical situation of our design, in which two voltage domains (0.77V and 1.1V) were embedded into a third one (0.924V), and a High-to-Low shifter must be used to connecting signals which were located at different voltage domains.

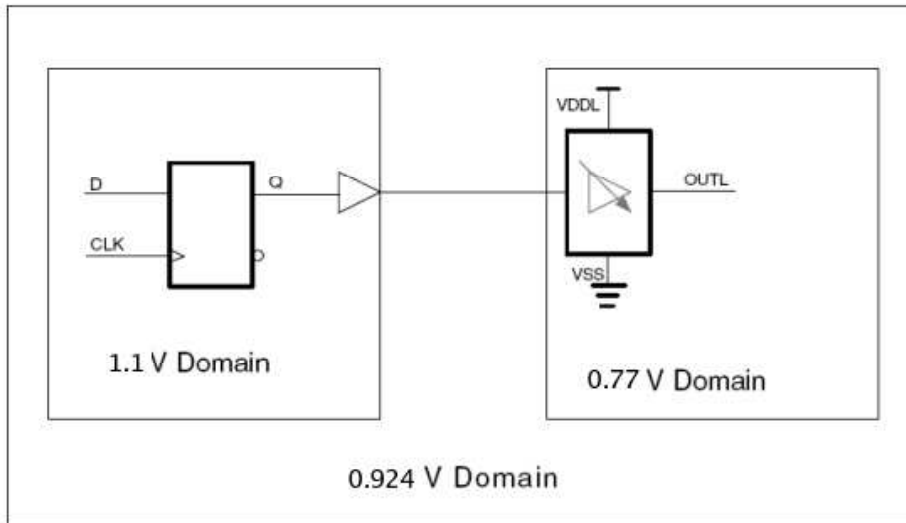


Figura 5.27: High-to-Low Level Shifter in Destine Domain

Because the High-to-Low shifter uses the voltage rail from the lower voltage domain, so it was placed in the lower voltage domain, even the distance between the 1.1V domain and the 0.77V domain were small enough.

Level Shifter Bufferings during the Clock Tree Synthesis

Another placement issues happened when buffering the level shifters during the clock tree synthesis. In our case, the buffers of the clock tree must be placed on the

0.924V domain and be supplied with 1.1V. After this, they could be connected to the level shifters of low voltage domain, without timing impacts. Figure 5.28 shows clearly this case.

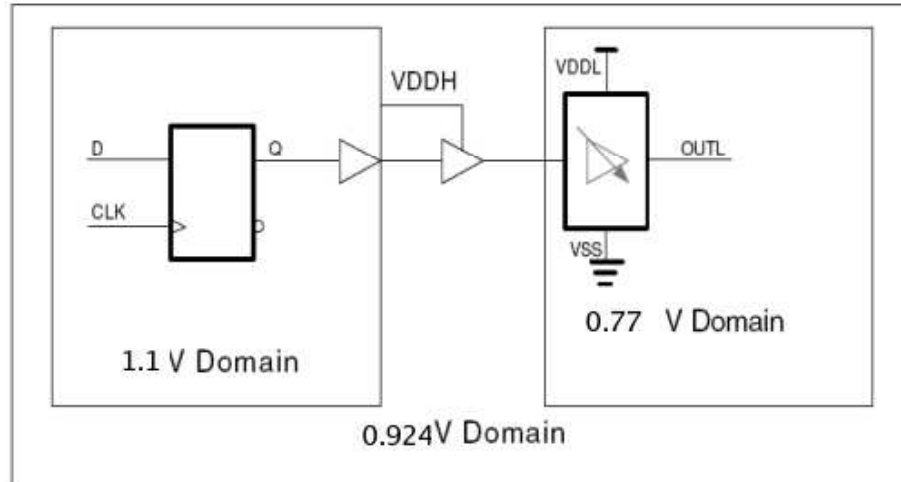


Figure 5.28: Placement of Level Shifters during the Clock Tree Synthesis

These clock buffers cannot be supplied by 0.924V. Since the signal from the high voltage domain swings between 1.1 and 0V and if the buffer were supplied with a smaller voltage, possible damages could happen.

In the case of Figure 5.28, the buffers of 0.924V domain are supplied by 1.1V (high voltage domain) then goes into the smaller voltage swing (0.77V to 0V).

Low-to-High Level Shifter

Figure 5.29 shows a signal passing from the 0.77V domain and goes into the 1.1V domain. In this case, power routing is a challenge no matter if the level shifter is placed or not. Because it requires both supply rails and at least one of the rails has to be routed from another domain.

Since the output driver (the OUTH output) requires more current than the input stage (the 0.77V domain), we placed the level shifter in the 1.1V domain and also put additional buffers in the 0.924V domain.

From the logical perspective, level shifters are just buffers and don't affect the functionality of the design. For this reason, the shifters can be inserted after the logic synthesis phase. No changes to the RTL were required.

After we analyzed those three cases of level shifters, with the automatic placer of the **SoC Encounter** framework, we finished the placement of standard cells and level shifters in about 20 minutes without many issues.

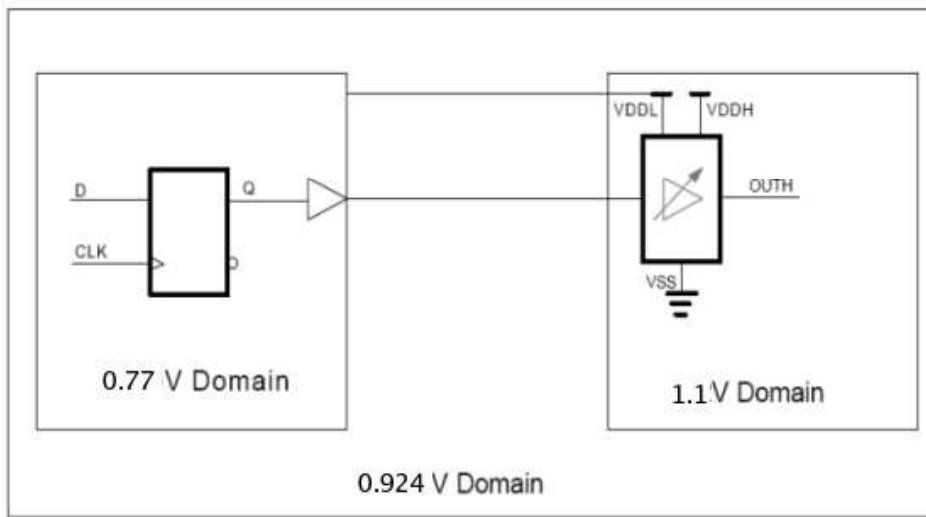


Figura 5.29: Level Shifter Placement

Next, we details the signal routing phase.

The router from the **Cadence SoC Encounter** is low-power aware and support the power intent file. There are, however, some potential pitfalls occurred during the detailed routing.

The memories were placed into a particular power domain using the hard placement (fixing their positions without help of automatic placer), and this approach introduced several routing restrictions. We repeatedly changed the position of the macrocells and several runs was required to complete the global route without timing violations.

Specifically recall for the clock tree routings. Signal was passed through many voltage areas and several runs were required to ensure the detailed routing of clock nets meet the timing constraints. Even after the phase of placement, additional level shifters were required to enable the nets crossing different voltage areas.

With helps of the automatic placer and router of the **SoC Encounter**, we still faced serious rework in fixing all the violations of the timing and DRC.

After analyzed the layout, we saw that the level shifter was an expensive solution when considering the area and the delay impacts. The area were greatly increased because the insertion and routing of shifters, and more efforts were required in fixing the timing violations during the STA task.

5.9.8 Clock Tree Synthesis

Because the voltage domains change the supply of each power domain and the insertion of level shifters also introduces significant timing delays, the construction of clock trees becomes particularly important.

In the CTS⁴¹, any degradation in rise and fall times across power domain boundaries can increase clock skew.

In our case, the CTS must be done at a multiple-voltage design (Figure 5.30), and we must deal with severe restrictions on designing the clock tree to meet the power and performance. These restrictions can be summarized as:

* Single clock source was used in multiple power domains and must cross different voltage areas.

* The latency was modified constantly and the design could present severe clock skew when the clock signal buffering cannot be balanced with the data path.

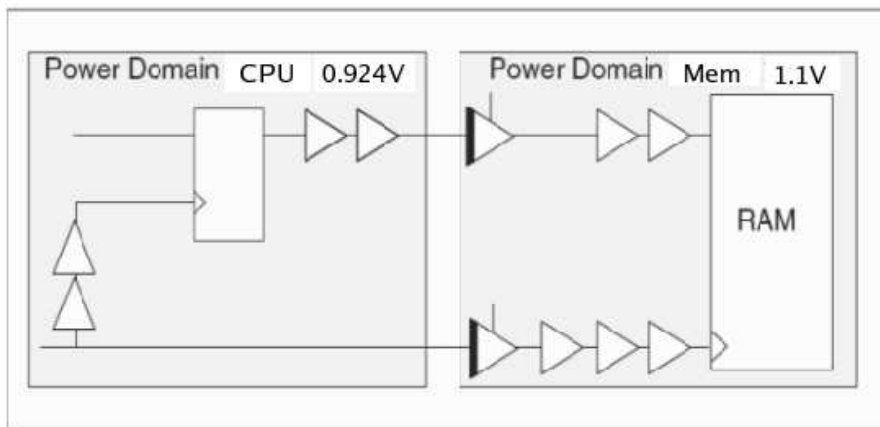


Figura 5.30: Clock Tree of Multiple Power Domain

In order to alleviate these restrictions, the clock tree synthesis must be multiple-voltage aware and use a bottom-up approach to construct the clock tree. For such purpose, each voltage domain must be processed in turn and the the clock network of each voltage area has to be constructed to minimize the skew.

After the low level clock tree were built (local clock tree clustering), then they must be merged to build an global clock network for the entire design. Figure 5.31

⁴¹clock tree synthesis

illustrates this clustering approach. Note that after the memory system and the core had respective level shifters inside their own clock trees, a global clock tree network must be built by connecting all the local clock trees into unique source.

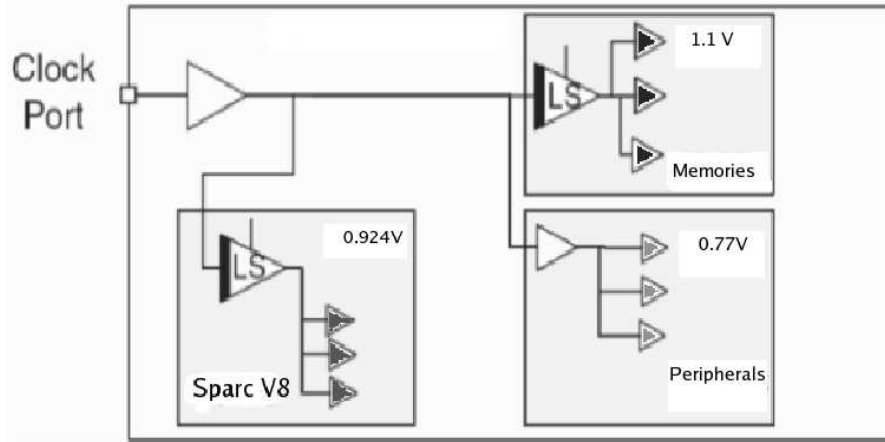


Figura 5.31: Clock Tree Synthesis at Multiple Power Domain

We spent a significant effort in fixing the timing issues after the clock tree was done. Several reruns were done and this process took about 2 weeks to be completed. The level shifter, in turn, also introduce significant impact in the final layout area.

5.9.9 Power Sign-Off

Having completed the implementation of our multiple-voltage, it was necessary to verify the integrity of the power network. Although the TSMC did not provide the full PDK for academic users (which prevented us to perform a complete manufacturing verification), we could complete the power network analysis using the results of layout.

By using the embedded power rail analysis engine from the **SoC Encounter Framework**, we performed an extensive analysis on the power network. Although the first power network analysis was completed during the power planning phase, we also detected several excessive voltage drop caused by local level clustering during the CTS.

Two aspects of our power network were critical: the voltage drop seen by the standard cells in the different voltage blocks and crowbar current in the regions of level shifters.

To solve this, we tried to resize the clock buffers, duplicating the level shifters and sizing the power mesh to alleviate the unacceptably voltage drop. After several days of interaction, we successfully fixed all the IR drop issues with significant area and timing impacts on the final layout.

With the commercial version of PDK, we should redo the analysis to have a better understanding about the IR drop issue, since the GDSII view and the complete power view of the macrocells and the standard cells could provide more precise estimations of the voltage drops of the entire design.

5.9.10 Post-layout Co-simulation

After the power sign-off analysis and the parasitic extraction, we successfully dumped the timing delay and outputted the SDF file to perform the post-layout simulation. Switching activity were extracted along the simulation, which took about 15 days, close to the baseline flow.

5.9.11 Power Estimation

The power estimation of multiple V_{dd} optimization was done as in the baseline flow. Additionally, we provided the **Voltage Storm** with the multiple voltage library and power data of each voltage domain. After that, we also specified the voltage of each domain with the precise switching activity (TCF from post-layout simulation).

5.9.12 Experimental Results and Impacts

In this section, we summarize the results of multiple V_{dd} optimization (Table 5.12) and highlight the impact and improvement.

Power Improvement

Table 5.12 shows a clear power improvement after the optimization was done. Compared with the baseline, the multiple V_{dd} reduced the power in about 44.68% (155969 down to 86282.05 μW). Beside this point, we also see the macrocells (cache memory and register file) almost remained in constant in the dynamic and the leakage dissipation. In total, they were responsible for 30.05% of the total power.

If we just considered the power reduction for other components of entire system, we saw the reduction factor is about 63.67% (109102.36 down to 39415.41 μW). The total dynamic was reduced in about 46.48% (150150 down to 80359.67 μW), while the leakage was reduced in about 6.634% (6315.26 down to 5922.38 μW). We can say, this technique present great result in reducing dynamic dissipation, however, cannot benefit the dissipation of macrocells and leakage effect.

Area and Gate Impacts

Module Name	Area	Internal	Switching	Leak	Power Domain (V)	Total Pwer	Total Gates	Pwer/Gate Coef
ahbctrl	214615.6	308.6	256.7	216	0.77	781.3	8559	0.09
apbctrl	9681.2	370.6	146.1	101.2	0.77	617.9	4119	0.15
apbuart	3187.3	281.4	202.4	90.2	0.77	574.0	1621	0.35
div32	14465.9	2995.3	684.5	211.2	0.924	3891	5871	0.66
mul32	5189.2	460.6	240.1	618.3	0.924	1319	2408	0.55
Dtag	54374.7	9551.6		170	1.1	9719.1	22480	0.43
Ddata	121261.8	9915.9		387	1.1	10282.8	50133	0.21
Itag	54374.7	11465.3		170	1.1	11734.1	22480	0.52
Idata	121261.8	10810.7		387	1.1	11091.7	50133	0.22
gptimer	15695.5	2425.4	468.4	184.2	0.77	3078	6171	0.5
grgpio	6134.6	227.3	104.8	736.1	0.77	1068.2	2583	0.41
irqmp	2769.3	201.5	95.3	291.2	0.77	588	1027	0.57
mmu_icache	4661.2	247.8	207.1	558.1	0.924	1013	1971	0.51
mmu_dcache	14716.2	1038.7	694.3	166.2	0.924	1899.2	5855	0.32
mmu_acache	11384.7	291.9	204.2	175.2	0.924	671.3	4398	0.15
pipeline	38354.6	7255	3664	404.2	0.924	11323.2	22348	0.51
regfile	79119.8	3638		380	1.1	4004	37343	0.11
sdmctrl	3481.7	597.65	182.01	37.16	0.77	817.2	1459	0.56
mmu	61256.5	1631.94		982.18	0.924	5721.3	27713	0.21
Layout	1050202.7	60082.7	20277	5922.4	(0.77, 0.924, 1.1)	86282.1	322197	0.518

Tabela 5.12: Multiple Power Supply result at 300MHz

Table 5.12 also shows an area and gate impact. The layout area was increased by about 14.58% (916567.22 to 1050202.72 μm^2) compared with the baseline. While the gate number was increased just in 6.63% (302158 to 322197 gates). We try to investigate the relationship between the two type of variations (area and gates), apparently, they were not well-correlated.

After observing better the layout, we see the interface between the SparV8 Core and the memory system (cache and register file) demanded much more area in the power structure and the routing of level shifters. The same happened with the interface between all peripherals and core. These larger area was required to minimize the timing violations when signals went between several different power domains. In essence, they did not add extra functionality to the design.

Performance Impacts

After the optimization, the design has the same clock speed as the baseline flow. However, we spent much effort in debugging the timing violations and solving the setup and hold time issues even after the clock tree and level shifter were built in the design.

The most evident advantage of multiple V_{dd} was the significant reduction of dynamic power, at the cost of a medium impact in the silicon area. Extra effort in fixing timing issues could be considered one of important factor when evaluate the adopting of this technique.

For those design which the silicon area was not considered critical, this technique can be considered an ideal choice to reducing the power dissipation.

5.10 Summary of all the experimental results

We summarize the experimental results in Table 5.13, showing their advantage and impact in terms of power (dynamic and leakage), area, speed and implementation impacts :

Parameter	Clock Gating	Operand Isolation	Multi V_{th}	Multi V_{dd}
Total Power	-19.00%	-10 ~ -15%	-3.70%	-44.68%
Dynamic Power	-20.14%	-10 ~ -15%	-2.40%	-46.48%
Leakage Power	-6.75%	0%	-46.70%	-6.63%
Area	+1.00%	~1%	-1.20%	+14.58%
Speed	0%	0%	0%	0%
Implem Impact	Synthesis, DFT	Logic Synthesis	Logic Synthesis, STA	Architecture, Synthesis, STA, Placement, Routing, Sign-Off

Tabela 5.13: Summary of All the Experimental Results

The line **Implem Effort** explains the phase that demanded most implementation effort during the project. We can see, the multiple V_{dd} approach shows implementation impact in almost all stages of the back-end design.

The Clock Gating can be considered as a less-effort approach in reducing the total power. As shown in the table 5.13, it reduced the total power in almost 20%, with very low effort during the implementation when compared with the multiple V_{dd} .

The results of Operand Isolation are theoretical. It showed no effect to our experimental platform.

This table provides a quick comparison of several aspect of the selected techniques, and also helps the project manager in deciding the optimization strategy when the project demands a trade-off between power, performance and area.

Capítulo 6

Conclusão e Trabalhos Futuros

Nesta dissertação, descrevemos e aplicamos várias técnicas de otimização de consumo de energia num projeto de SoC industrial. Para cada técnica adotada, alcançamos diferentes resultados na redução de consumo de energia. Também mostramos as detalhes de implementação e otimização de cada técnica, as quais normalmente não podem ser divulgadas pelas empresas devido aos acordos confidenciais. Em seguida, resumimos alguns pontos para concluir o nosso trabalho e apontar alguns possíveis trabalhos futuros.

6.1 Conclusão

- A extração de atividades de chaveamento é a etapa mais demorada dentro do nosso projeto. Foi feita ao longo da simulação pós-layout e levando cerca de 240 horas para decodificar apenas 10 quadros de vídeo. Mesmo para um pequeno número de vetores de teste e uma quantidade média de transistores (em nosso caso, foram cerca de 1.2 milhões), a simulação pós-layout exigia um longo intervalo de tempo. Como foi mostrado na seção 2.2 e nos resultados experimentais, entendemos que os vetores de estímulo influenciam diretamente a estimativa de consumo de energia. Com base nestes dados, percebemos que para um sistema de grande porte (com mais de 10 milhões de transistores, suportando diversos tipo de software de aplicação), a geração dos vetores de estímulo, que são capazes de provocar chaveamentos adequados num intervalo de tempo aceitável, torna-se uma tarefa desafiadora.
- Ao longo do nosso projeto, exceto para a técnica de **operand isolation**, demonstramos que todas as técnicas são capazes de abordar um determinado aspecto do projeto e reduzir a dissipação de potência.

Na otimização do **clock gating**, células especiais foram inseridas no código RTL e mapeadas para a tecnologia alvo e assim reduzindo moderadamente as atividades

de chaveamento nas células sequenciais. A otimização do **multiple threshold voltage** reduziu a dissipação de potência estática por corrente de vazamento através do uso de várias tensões limiares, sem alterar funcionalidades lógicas ou atividades de chaveamento. A otimização do **multiple supply voltage** focou arquitetura física do projeto, dividiu o projeto em ilha de tensões e inseriu vários elementos para aliviar os possíveis impactos. As atividades de chaveamento e as funcionalidades lógicas também não foram modificados.

- Este trabalho mostrou que a seleção e aplicação das técnicas de redução de consumo de energia são questões de nível de sistema. É fundamental que a equipe de projeto deve ter compreensão clara sobre a arquitetura do sistema e a tecnologia de manufatura. Não existe uma metodologia padrão para atacar o problema e não se pode alcançar as metas de consumo de energia aplicando apenas uma única técnica. Combinação de várias abordagens e estabelecer 'trade-offs' entre a área, a potência, a velocidade e o esforço de implementação/verificação sempre serão passos imprescindíveis na construção de um projeto desse gênero.
- O uso das macrocélulas (os SRAMs) facilitou a implementação do sistema de cache e o arquivo de registradores. Entretanto, nenhuma das técnicas selecionadas foram capazes de reduzir a dissipação de potência destes elementos. Para este caso, duas possíveis abordagens para reduzir a dissipação são: escolher uma versão de baixo consumo de energia dos modelos das macrocélulas ou alterar o comportamento de acesso aos caches para reduzir as atividades de chaveamento.
- Usando a técnica de **clock gating**, junto com a **multiple threshold voltage** com aplicação de múltiplas tensões de alimentação em diversas regiões do projeto, conseguimos mais redução no consumo de energia. No entanto, o tempo de simulação pós-layout continua sendo uma grande limitação para otimizar o tempo de fluxo.
- É importante e interessante notar que os resultados/impactos das técnicas eram muito genéricas e estes dados podem ser aplicados em qualquer plataforma que tenha características semelhantes (tanto no nível de arquitetura como no software de aplicação). Podemos usar estes resultados (melhorias e impactos) para avaliar o fator de redução de consumo de potência e prever possíveis problemas antes de realmente iniciar um novo projeto.

6.2 Trabalhos Futuros

- Um dos possíveis trabalhos futuros é investigar melhor as abordagens de geração de vetores de estímulo. Estes vetores devem ser menores em tamanho e mais precisos

em detectar picos altos ou baixos de consumo de energia.

- Importante lembrar que essas técnicas focam a fase 'back-end' do projeto ASIC. Eles não são adequados quando o projeto exige estimativa/otimização de consumo de energia em estágios iniciais (por exemplo, projeto no nível de sistema eletrônico ou projeto de modelagem comportamental). Outro possível trabalho é tentar comparar a eficiência destas técnicas com aquelas técnicas que são adotadas pelos projetos de alto nível.
- Também foi identificado que a metodologia de verificação funcional durante a fase de RTL ainda possui pouca preocupação com relação ao consumo de energia. Neste projeto, dependíamos intensamente do "arquivo de intenção de potência" e um conjunto específico de ferramentas de EDA para concluir a simulação de projeto RTL levando aspecto de potência em consideração. Um outro possível trabalho seria adicionar as informações de consumo de energia tanto em desenvolvimento de código RTL como em metodologia de verificação funcional, fazendo com que a verificação possa ser integrada à representação única de intenção de potência, aliviando as dependências de ferramentas EDA.

Referências Bibliográficas

- [1] W. Bowhill A. Chandrakasan and F. Fox. *Design of High Performance Microprocessor Circuits*. Piscataway, NJ:IEEE Press, second edition, 2001.
- [2] R. W. Broderson A. P. Chandrakasan. *Low Power Digital CMOS Design*. Kluwer Academic Publishers, first edition, 1995.
- [3] Bashir M. Al-Hashimi. *System on a Chip: Next Generation Electronics (Circuits, Devices and Systems)*. Institution of Engineering and Technology, first edition, January 2006.
- [4] ARM. *AMBA-2.0 specification, Rev2.0*, March 1999.
- [5] De Micheli G. Benini, Luca. Automatic synthesis of low-power gated-clock finite-state machines. volume 15 Issue:6, pages 630–643. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, IEEE Council on Electronic Design Automation, 1996.
- [6] Robert Chau. *Advanced Metal Gate/High-K Dielectric Stacks for High-Performance CMOS Transistors*. American Vacuum Society 5th International Conference on Microelectronics and Interfaces,, march 2004.
- [7] MediaBench Consortium. *MPEG-2 Video Decoder - ISO/IEC 13818-2*, Dez 1996. <http://euler.slu.edu/~fritts/mediabench>.
- [8] Intel Corporation. *High K and Metal Gate Research*. Intel Corporation, Dez 2009. <http://www.intel.com/technology/silicon/high-k.htm>.
- [9] A. Correale. Overview of the power minimization techniques employed in the ibm powerpc 4xx embedded controllers. pages 75–80. ACM/IEEE International Symposium on Low Power Designs, ACM/IEEE, Apr 1995.
- [10] Dake Liu , Christer Svensson. Power Consumption Estimation in CMOS Chips. volume 29. IEEE Journal of Solid-State Circuits, IEEE, June 1994.

- [11] John L. Hennessy David A. Patterson. *Computer Organization and Design: The Hardware/Software Interface*. Morgan Kaufmann, fourth edition, october 2008.
- [12] International Organization for Standardization. *IEEE Standard Specifications for the Implementation of 8 by 8 Inverse Discrete Cosine Transform, IEEE Std 1180-1990*, 1990.
- [13] International Organization for Standardization. *International Standard ISO/IEC IS 11172: Coding of Moving Pictures and Associated Audio for Digital Storage media up to about 1.5Mbits/s, Part2 : Video*, 1993.
- [14] International Organization for Standardization. *International Standard ISO/IEC DIS 13818: Generic Coding of Moving Pictures and Associated Audio, Part2 : Video*, 2007.
- [15] SPARC International Inc. *The SPARC Architecture Manual, version 8*. SPARC International Inc., Dez 1991.
- [16] Power Forward Initiative. *A Practical Guide to Low-Power Design*, first edition, 2008.
- [17] Borivoje Nikoli Jean M. Rabaey, A. P. Chandrakasan. *Digital Integrated Circuits, a Design Perspective*. Printice Hall Electronics and VLSI serie, Pearson Education Inc., second edition, 2003.
- [18] Kiat Seng Yeo Kaushik Roy. *Low Voltage, Low Power VLSI Subsystems*. McGraw-Hill Professional, May 2004.
- [19] R. Mehra J. Sproch M. Munich, B. Wurth and N. Wehn. Automating rt-level operand isolation to minimize power consumption in datapath. Paris, France, 2000. Design Automation and Test in Europe, DATE.
- [20] Christopher Malone and Christian Belady. EAC & PUE: metrics to characterize IT equipment & data center energy use. Digital Power Forum, 2006.
- [21] Pierre Bricaud Michael Keating. *Reuse Methodolgy Manual - For System on A chip Designs*. Kluwer Academic Publishers, second edition, 2001.
- [22] Siva G. Narendra and Anantha Chandrakasan. *Leakage in Nanometer CMOS Technology*. Springer Publications, first edition, 2006.
- [23] David Money Harris Neil. H. E. Weste. *CMOS VLSI Design, a Circuit and System Perspective*. Addison-Wesley, Pearson Education Inc., fourth edition, Jun 2009.

- [24] Gaisler Research. *GRLIB User's Manual*, Dez 2008.
- [25] Gaisler Research. *Snapgear for LEON manual*, Dez 2008.
- [26] International Technology Roadmap For Semiconductors. *International Technology Roadmap For Semiconductors - Design 2009 Edition*. ITRS, Dez 2009.