北京大学

硕士研究生学位论文

题目：低功耗 ASIC 芯片设计方案的实现与比较

姓　　名：吴　杰

学　　号：10617071

院　　系：软件与微电子学院

专　　业：电子与通信工程

研究方向：集成系统芯片（SOC）

导　　师：于敦山

二〇〇八 年 四 月

摘要

随着CMOS工艺的发展，器件特征尺寸不断缩小，芯片集成度和性能不断提高，使得功耗问题变得越发的突出，为了避免芯片过热导致电路性能和可靠性的下降，人们需要花费更多的成本去解决芯片封装与散热问题。另一方面，消费电子和移动便携产品需求的日益增长需要电池能够维持更长的使用寿命，所以如何从电路角度降低功耗成为今天芯片设计领域亟待解决的重要课题。

多年来CMOS电路的低功耗理论研究发展十分迅速，许多新理论、新方法的提出对降低电路功耗起到了非常好的效果。同时我们也注意到，虽然集成电路设计是一门十分强调知识密集型和工程实践型的学科，但是由于自身产业链分工的细化以及高风险、高投入的特点使得工程应用型文章相比于理论研究成果的文章明显匮乏。因此本文从实际工程应用出发利用Cadence Design Systems, Inc.提供的双音多频数字接收电路和Encounter数字集成电路设计平台，结合中芯国际90 nm低功耗单元库，实现了基于多种低功耗技术的逻辑综合与物理综合设计。文中采用的是一些业界普遍使用和逐渐推广的低功耗技术，并证明可以在工程设计中予以实现。

本研究工作首先从CMOS逻辑单元的功耗模型入手，回顾了动态功耗和静态功耗的来源以及EDA工具计算、建立功耗的模型，之后分别简要介绍门控时钟技术，操作数隔离技术，多供电多电压技术，电压关断技术以及多阈值单元组合技术等低功耗方法的设计思想，并对时序库中功耗以及相关特殊单元的定义也作了介绍。其次，基于RTL Compiler逻辑综合工具并结合多供电多电压技术，门控时钟技术，操作数隔离技术和多阈值单元组合技术对双音多频接收器进行逻辑综合，所得到的门级网表再利用Encounter实现电路的后端物理设计。文中还按照逻辑综合与物理实现的流程和主要步骤给出了相关工具的命令和必要的解释。此外还分别介绍了基于Encounter平台和ETS工具的签收验证方法，使电压降，地电压升高，信号噪声和电迁移等影响电路性能和可靠性的问题得以解决。

研究工作最后使用不同方式组合上述低功耗技术，对它们各自在降低电路功耗方面所起的作用进行了比较，进一步验证了其在实际工程应用中降低功耗的作用和效果。因此本工作的重点是结合低功耗理论、逻辑综合和物理设计的实施方法，应用当代先进的EDA工具平台来检验最终结果。

关键词：低功耗、多供电多电压、逻辑综合、物理设计

Abstract

With the development of CMOS process, unceasing miniaturization of device's characteristic size, continuous increase of chip's density from integration and its performance, power dissipation has become a leading issue. In order to avoid decreased performance and reliability of chip circuits due to overheating, more manufacturing cost is required to solve problems of chip's package and heat radiation. On the other hand, with continuous expansion of consumer electronics, mobiles and portable devices, battery's life expectancy has to be prolonged, so as to meet consumers' requirements. How to reduce power dissipation in light of circuit turns to be a prominent task demanding prompt solution in the field of chip design and its implementation.

In the last decade, theoretical research of CMOS circuit's low power dissipation has been growing significantly. New theories and methods work well in reducing power dissipation of circuit. Meanwhile, the author has also noticed theoretical theses prevail over engineering application-oriented ones far and away. All this is a result of integrated circuit's thinning division of industry chain and its high risk, high devotion characteristics albeit integrated circuit is a course with a great emphasis on knowledge intensity and engineering practicability. Thus, this thesis will employ DTMF digital receive circuit using Encounter digital integrated circuit design platform of Cadence Design Systems, Inc. and apply SMIC 90 nm standard cell library of low power dissipation, to logic synthesis and physical synthesis based on numerous low power dissipation techniques. Experimental labs of low power dissipation technologies are to be used further to demonstrate the application of modern low power theories.

The thesis starts firstly with power dissipation model of CMOS logical cells then looks back the origin of dynamic power dissipation and static power dissipation, calculation and power dissipation model with EDA tool, low power dissipation techniques, such as clock gating, operating isolation, MSMV, PSO and combination of MTV cells are reviewed, together with the necessary special cells for the low power design in timing library. Secondly, based on RTL Compiler, the new generation of logic synthesis tools, combined with technologies mentioned above to carry out a logic synthesis of DTMF digital receiver. Thirdly, importing the gate level netlist into SoC Encounter to implement the physical design. According to flow and main steps of logic synthesis and physical

implementation the thesis will bring forth demands of relative tools and explanations if necessary. In addition, it will introduce sign-off verification method based on Encounter ETS tools respectively for the purpose of resolving problems affecting circuit performance and reliability such as IR-drop and ground bounce, signal noise and electromigration.

Finally the author will combine low power dissipation technologies through different methods, compare the outcomes and discuss the practical values. Therefore, the key point of the work is to verify the final result using modern advanced EDA platform, which was combined with low power consumption theory, logic synthesis and physical implementation method.

Keyword: low power; MSMV; logic synthsis; physical design

图目录

表目录

版权声明

北京大学学位论文原创性声明和使用授权说明

原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品或成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本声明的法律结果由本人承担。

论文作者签名：吴杰 日期：2008年5月27日

学位论文使用授权说明

本人完全了解北京大学关于收集、保存、使用学位论文的规定，即：

按照学校要求提交学位论文的印刷本和电子版本；

学校有权保存学位论文的印刷本和电子版，并提供目录检索与阅览服务；

学校可以采用影印、缩印、数字化或其它复制手段保存论文；

在不以赢利为目的的前提下，学校可以公布论文的部分或全部内容。

（保密论文在解密后遵守此规定）

论文作者签名：吴杰，导师签名：

日期：2008年5月27日

第一章 功耗模型与功耗计算

为满足基于标准单元库的数字集成电路设计方法及流程，CMOS 逻辑电路的功耗理论模型与 EDA 工具构建的功耗模型虽在本质上是完全一致的，但在实际表述时却存在差异。本章将根据两种不同功耗模型介绍电路功耗产生的机理为低功耗技术的实现提供理论依据。

1.1 CMOS 逻辑电路功耗理论模型

以数字电路最基本的反相器为例，功耗包括动态功耗（dynamic power）和静态功耗功耗（static power）两部分（表 1- 1），动态功耗是电路状态发生变化时产生的功耗，它由跳变功耗(switching power) 和短路功耗（short power）两部分组成，跳变功耗是对电路本征电容和线负载电容充放电时产生的功耗（图 1- 1）。短路功耗是电路在翻转过程中上拉 P 管和下拉 N 管同时短暂导通，形成电源到地的通路电流而产生的功耗，短路功耗与输入信号的频率、翻转率，输出负载，器件阈值以及电压大小有关，（图 1- 2）。

总功耗 (P_{total})	动态功耗 (dynamic power)	跳变功耗 (switching power)	$P_{SW} = V_{DD}^2 C_L f_{clk} \alpha$
		短路功耗 (short power)	$P_{SC} = I_{SC} V_{DD}$
	静态功耗 (static power)	泄漏功耗 (leakage power)	$P_{ST} = V_{DD} I_{leakage}$
$P_{total} = C_L V_{DD}^2 f_{clk} \alpha + V_{DD} I_{short} + V_{DD} I_{leak}$			

表 1- 1 功耗组成的理论模型

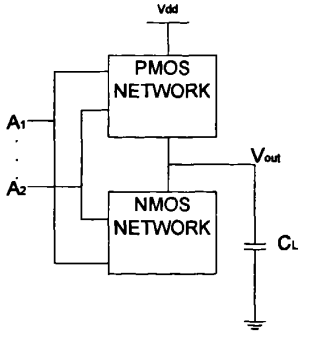


图 1- 1 跳变功耗

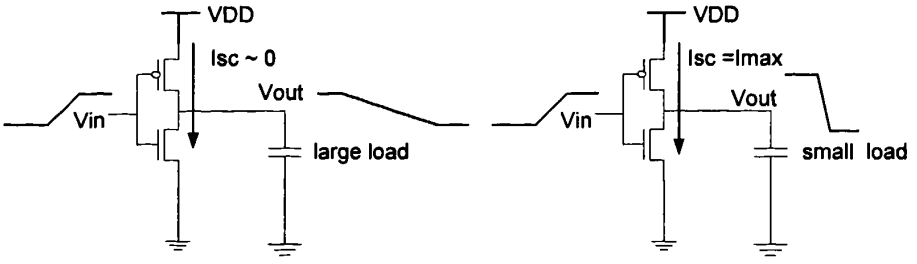


图 1- 2 短路电流引起的短路功耗

当逻辑门不执行操作或者不从一种状态转换到另一种状态时，由 PN 结反偏电流、亚阈值电流、栅泄漏电流等效应引起的功耗称为静态功耗，也称为泄漏功耗。泄漏电流主要有以下几个来源，（图 1- 3）分别是 I1：PN 结反偏电流，I2：亚阈值电流，I3：DIBL，I4：GIDL，I5：串通电流，I6：窄沟道效应电流，I7：栅泄漏电流，I8：热载流子效应。对于 130nm 以下工艺，亚阈值泄漏电流和栅泄漏电流起主要作用，从亚阈值电流（公式 1-1）可以看出泄漏电流 I_{leak} 与阈值电压 V_{th} 成反比，阈值越低，泄漏电流越大。虽然单个逻辑门的泄漏功耗非常地小，但对于上万门的 SoC 设计来说泄漏功耗所占总功耗的比例已变的越来越大。特别是到 90 nm 工艺后由于阈值电压的降低和栅氧化层厚度的减小使得亚阈值泄漏电流和栅泄漏电流呈指数级增加，泄漏功耗将占到总功耗的近 50%左右。

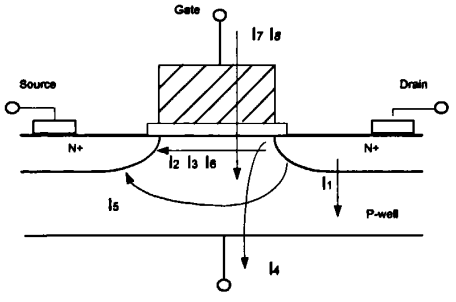


图 1- 3 CMOS 器件泄漏电流

$$I_{leak} = I_0 \exp(\frac{-qV_{th}}{mkT}) \qquad (1\text{-}1)$$

1.2 EDA 工具的功耗模型建立

第 1.1 节中将 CMOS 反相器的功耗分为动态功耗和静态功耗两部分，而基于标准单元库的 ASIC 设计由于设计方法学的需要，通常将总功耗分为单元功耗（instance power）和线网功耗（net power）两部分（图 1-4）。

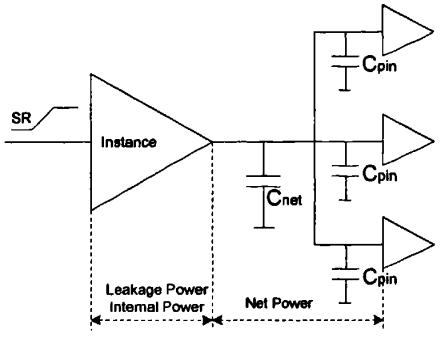


图 1-4 EDA 工具建立的功耗模型

单元功耗包括内部功耗（internal power）和泄漏功耗两部分，内部功耗指电源地对器件本征电容进行充放电时所消耗的功耗。它与跳变功耗功的区别在于：跳变功耗不仅指电源地对器件本征电容进行充放电时产生的功耗，还包括对互连线电容以及互连线上扇出输入电容充放电消耗的功耗（表 1-2）。

EDA 工具采用这种模型定义功耗是因为标准单元库设计完成后，每个单元在输入信号不同翻转时间和负载下泄漏功耗与内部功耗可通过工具仿真得到，因此单元功耗可采用查表（look-up table）的方式在时序库中得到。而后端电路设计方案千差万别，不同的模块摆放，不同的布线金属层，布线位置，布线长短以及扇出数都会影响到线网实际的寄生参数，所以在电路布线后考虑实际互连线 RC 得到的线网功耗才是准确的。

总功耗 (P_{total})	单元功耗 (instance power)	内部功耗 (internal power)	$P_{internal}$
		泄漏功耗 (leakage power)	$P_{leakage}$
	线网功耗 (net power)	线网功耗 (net power)	P_{net}

表 1-2 功耗组成的 EDA 模型

1.3 功耗的计算

单元的内部功耗在时序库中以查表的方式给出，它是输入信号翻转时间 SR 和输出负载电容 C_L 的函数，此外实际工作中单元的输出状态不可能在每个时钟周期都发生翻转，因此单元消耗的动态功耗还与信号的翻转率（toggle rate）有关（公式 1-2）。

$$P_{internal} = \sum_{PerArc}\left(TR_{arc_{ij}} \times \phi\left(SR_i, C_j\right)\right)$$

$$\phi = 0.5 \times rise\_power(SR,C) + 0.5 \times fall\_power(SR,C) \qquad (1\text{-}2)$$

式中 TR 为信号翻转率， SR 为输入信号翻转时间。

泄漏功耗可从时序库中得到，通常用唯一的常数表示单元泄漏功耗值，但在 90nm 工艺节点下，为了更精确的表示电路泄漏功耗，时序库的泄漏功耗依据输入信号的不同组合状态而不同。见公式 1-3，其中 k 是输入信号的状态组合，$P_{state\_leakage}$ 是某一输入状态下单元的泄漏功耗，$probability_{state}$ 是该组合发生的概率。

$$P_{cell\_leakage} = \sum_{state=1}^{k} P_{state\_leakage} \times probability_{state} \qquad (1\text{-}3)$$

第二章 低功耗技术概述

芯片的功耗来自于动态功耗和泄漏功耗。本章针对上述两方面将分别介绍工程应用中的几种低功耗设计技术，包括降低动态功耗所采用的门控时钟技术、操作数隔离技术、多供电多电压技术，动态频率降低技术、门级优化技术，以及降低静态泄露功耗使用的多阈值单元组合技术，电压关断技术和衬底反偏电压技术。

2.1 动态功耗优化技术

当电路状态发生跳变时，电源地对负载电容的充放电产生跳变功耗，输入信号翻转使电路 P 管和 N 管同时导通形成电源到地的直流通路产生短路功耗，跳变功耗与短路功耗共同构成了电路的动态功耗。

2.1.1 门控时钟技术

门控时钟技术对送到全芯片寄存器的时钟网络进行管理，其设计思想是当模块处于空闲状态时关断时钟在寄存器时钟端的翻转以消除不必要的功耗消耗。（图 2-1-a）中 HDL 代码是存储电路最常见的一种描述方式，经过逻辑综合可得到（图 2-1-b） 所示电路，从图可知当 En 使能信号无效时，存储电路输出值保持不变，但此时时钟信号始终在存储电路时钟输入端进行跳变，电路中存在冗余的开关动作，消耗额外的功耗。这类情况可以在存储电路时钟输入端加入控制电路，当输出状态不需要改变时关断电路的时钟输入，从而达到降低功耗的目的（图 2- 1-c）。

门控时钟技术不仅消除了存储电路内部和时钟网络上不必要的功耗消耗，而且由于将原电路中的多路选择器以控制逻辑代替，通常是一个控制逻辑控制一组存储器，所以门控时钟技术还可减小芯片面积[1]。例如 32 位的寄存器堆插入门控时钟可减少 32 个多路选择器占有的面积，取而代之的是几个门控电路，因此有效地节省了芯片面积。

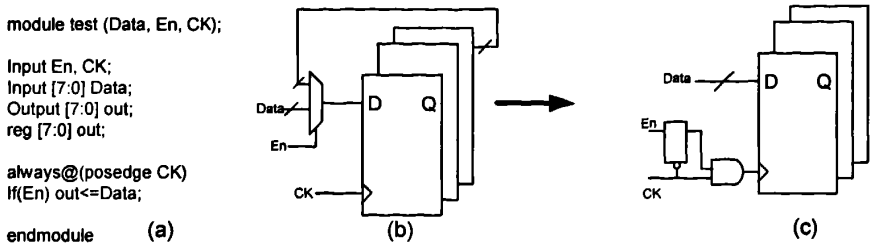


图 2- 1 插入门控时钟

另一方面，门控时钟技术也会带来一些潜在的问题：1）增加时钟偏差和时钟网络延时，2）使能控制信号毛刺影响电路功能的正确性，3）时序对门控单元的物理位置敏感，4）结构的复杂化增加了 DFT 的设计难度和时序收敛的难度，5）增大了门控时钟的逻辑综合结果与时钟树综合结果的误差。

2.1.2 操作数隔离技术

门控时钟技术有效地减小了时序电路时钟端冗余跳变产生的动态功耗，但当数据通路组合逻辑模块的输入信号发生不必要的翻转引起电路操作时同样会产生额外地功耗消耗，图 2- 2-a 的 HDL 代码经过逻辑综合后会得到图 2- 2-b 所示逻辑电路原理图，任一输入信号的翻转都会引起加法电路的操作，而当此时所得结果并不被下一逻辑使用时，就会造成加法电路不必要的功耗消耗。

操作数隔离技术（operand isolation）对于减小由于数据通路模块跳变产生的功耗非常有效，（图 2- 2-c）在电路中插入额外隔离单元（如与门），将使能信号 en 作为输入信号进入数据通路模块的控制信号，当 en 无效时，与门输出逻辑为“0”加法器操作数被隔离，因此不会产生由于加法电路操作而引起的冗余功耗。

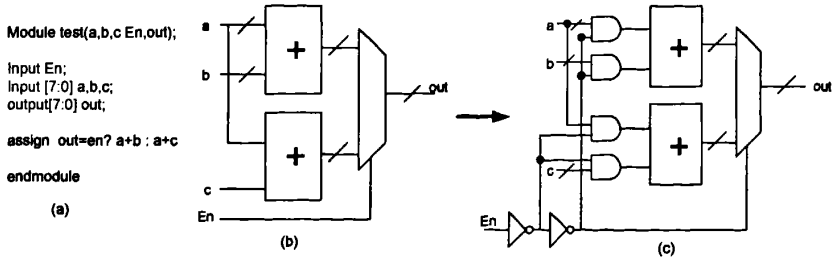


图 2- 2 插入操作数隔离单元

2.1.3 多供电多电压技术

动态功耗正比于供电电压的平方，因此减小电路的供电电压是降低功耗的最有效方法。芯片内含有许多不同功能的模块，它们通常在不同的时钟频率下工作，因此对性能要求也不尽相同，对于工作速度较低的模块可以适当的降低电压，这样在保证电路时序要求的前提下使功耗也相应降低。多电源多电压技术早期被用在高端的全定制设计中，而对于 ASIC 设计来说由于没有自动化设计方法和不同电压下特称单元库的支持，所以长期以来没有在 ASIC 领域得到广泛的推广。如今随着集成电路产业链的发展 MSMV 在 ASIC 领域的实现成为可能。采用这种方法的关键是要合理地分割电路模块，减少接口数量和提高信号电压的转换效率。图 2-3 是工作在不同电压阈（power domain）的模块的示意图。

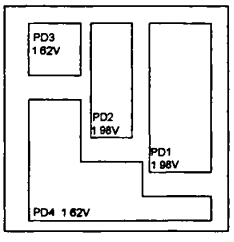


图 2-3 电压域模块示意图

2.1.4 动态电压频率调节技术

动态电压频率降低技术是近年来微处理器低功耗设计的一个突破，并逐渐在实时处理系统中得到广泛的关注，这种技术在保持系统正常工作的前提下允许动态地调节电路工作电压和频率，不仅能够减小电路的功耗而且延长了电路的使用寿命。但 DVFS 作为一种先进的设计方法实现起来比较复杂，方案设置不好不仅不能达到设计者良好的初衷，反而可能导致系统失效。

下面结合图 2-4 来介绍一下 DVFS 的技术思想[2]。在一个实时系统里假设任务 A 需在 t_2 时刻前完成，系统的工作电压为 V_1 的情况下，任务 A 将在 t_1 时刻完成，那么在执行下一个任务前系统将处于空闲状态。假设 $\triangle t=t_2-t_1=t_1-t_0$，系统此时的工作频率为 f，如果系统处于空闲状态所消耗的能量为 0，那么系统处理 A 任务所消耗的能量为 $E=P_1\triangle t=V^2f\triangle t$。如果工作电压降为 $V_2=0.5V_1$ 时，任务可以在时间 t_2 处理完毕，而此时工作频率由于工作电压的降低线性降为最大频率的一半，即 0.5f，那么

由以上假设可以得到工作电压 V_2 时完成任务 A 所消耗的能量为 $E=P_2\triangle t=V^2f\triangle t/4$。由此可见，任务 A 在规定的时限内处理完毕，满足系统的实时性，而工作电压为 V_2 的情况节省了 75%的能耗。

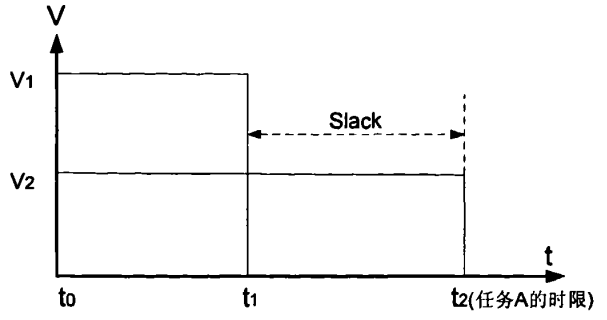


图 2-4 DVFS 技术示例

DVFS 技术能否在低功耗设计中发挥作用，仅有芯片的支持是不够的还需要软件与硬件的综合设计。一个典型的 DVFS 系统的工作流程如下[3]：

1）采集与系统负载有关的信号，计算当前的系统负载，这个过程可以由软件实现也可以用硬件实现，软件实现一般是在操作系统的核心调用中安放钩子，特别是调度器，根据其调用的频率来判断系统的负载。硬件实现通过采集一些核心信号中断线、Cache、内存总线的使用情况等，计算当前的系统负载。

2）根据系统的当前负载，预测系统在下一时间段需要的性能。这一方法有多种预测算法，即可由软件实现，也可以由硬件实现。

3）将预测的性能转化成需要的频率，从而调整芯片的时钟设计。

4）根据新的频率计算相应的电压，通知电源管理模块调整电路供电电压。

另外，在调整频率和电压时，要特别注意调整的顺序。当频率由高到低调整时，应先降频率，再降电压；相反，当频率升高时，应该先升电压，再升频率。

基于软件的 DVFS 实现，一般通过在操作系统的核心调用中安放钩子来收集系统调用的信息，判断当前的系统负载。在预测下一时间段的系统负载时，需要利用前面几个时间段的实际负载值，然后利用不同的预测算法进行预测。预测算法有：先前值法（previous value PV）、移动平均负载算法（moving average workload

MAW）、最小均方根法（least mean square, LMS）等。以上这些算法各有其优缺点，例如 LMS 算法类似于自适应滤波器，能够自动调整参数，但面临着收敛速度的问题。

基于硬件的 DVFS 系统要求每个任务必须在规定的时限内完成，否则视为无效操作，硬件实现一方面增强了负载计算的准确性，另一方面减轻了 CPU 用于负载跟踪与性能预测的负担，但硬件的弊端是缺乏对预测算法选择的灵活性，电路主要的活动信号与空闲信号被采集后，送到硬件模块进行性能预测，得到的结果与预先设置的门限进行比较，如果预测的性能需求高于上限，则请求调高频率，反之，如果预测性能低于下限，则请求降低频率。这种请求作为一种中断信号发给控制模块设置相应的频率和电压。

2.1.5 门级功耗优化技术

门级功耗优化技术一般采用以下几种方法：

1）在不违反时序的前提下，选用较小的逻辑门从而减小动态功耗（图 2-5）。

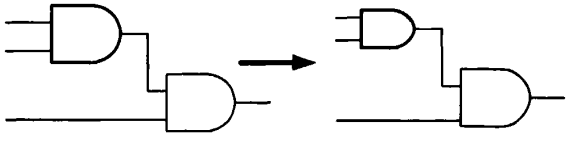


图 2-5 调整单元尺寸

2）在满足时序的前提下，删除多余的反相器和缓冲电路，（图 2-6）。

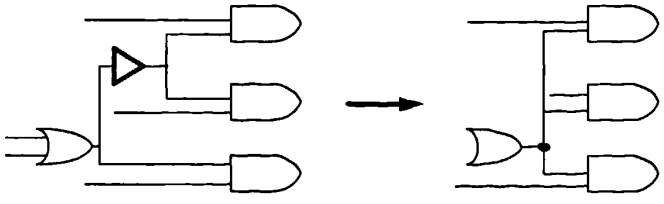


图 2-6 删除冗余逻辑单元

3）逻辑重建

如图 2-7 所示，(A)为逻辑重建前的电路结构，(B)为逻辑重建后的电路结构。A，B，C，D 是四个输入信号，其中 A 的节点翻转率最低，而 C 的节点翻转率最高，在(A)图中，C 信号的传播造成两个内部节点的翻转率都很高。在保证逻辑功能一致的情况下，重建的电路如图(B)所示，其内部节点的翻转率明显减小，从而降低了动态功耗。

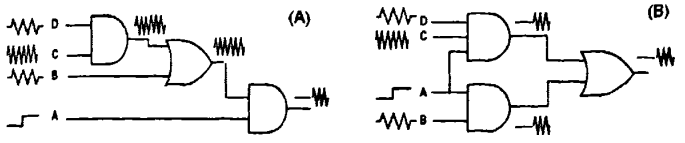


图 2-7 逻辑重建

2.2 泄漏功耗优化技术

CMOS 工艺技术的进步使得器件尺寸和供电电压不断减小，为保证器件性能，阈值电压也相应随之降低，使得亚阈值泄漏电流呈指数级增加。此外 COMS 器件栅氧厚度的缩小，栅泄漏电流增大，也增加了电路的泄漏功耗。有数据表明 90nm 工艺节点下泄漏功耗已逐渐取代动态功耗成为功耗的主要组成部分，因此对低功耗设计而言，有效的泄漏功耗优化技术变得非常地重要。

2.2.1 多阈值单元组合技术

阈值电压高的单元漏电流较小，但工作频率较低；反之，阈值电压较低的单元漏电流较大，但工作频率较高。因此为减小漏电流，可以对电路中的关键路径采用低阈值单元，非关键路径则选用高阈值单元。通过逻辑综合工具获得最优化的高阈值和低阈值单元的组合，能够在满足设计性能要求的同时尽可能地降低漏电流，减小泄漏功耗（图 2-8）。

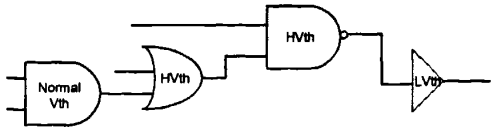


图 2-8 多阈值电压单元组合

2.2.2 电压关断技术

电压关断（power shutdown）技术是降低电路功耗，特别是泄漏功耗的有效方法。这一方法的实现依赖于电压门技术（power gating）的使用，它通常在逻辑单元与电源/地的连接处插入额外的 N 型或 P 型 MOS 管，构成电路与电源/地的虚连以减小泄漏电流。目前主要采用两种电压门结构，Fine-Grain 结构和 Coarse-Grain 结构。

Fine-Grain 结构如图 2- 9，在单元内部加入额外的 MOS 管（也称睡眠管），当两管导通时，电路正常工作，两管关闭时则切断电路与实际电源/地的连接。睡眠管的阈值电压较高。出于减小电路导通电阻的考虑，更多情况下每个单元只由一个 N 管或 P 管构成电压门，此外，为了使之能够承受很大的电流冲击需要调整门控管的体积，如果门控管设计得太大，不但会增大面积，还加大了延迟，影响电路性能；如果太小，影响系统的抗噪声性能，降低系统可靠性，甚至会导致电路根本无法工作。因此，在电路设计阶段对电路最大电流进行一个准确的估计是非常必要的[6]。

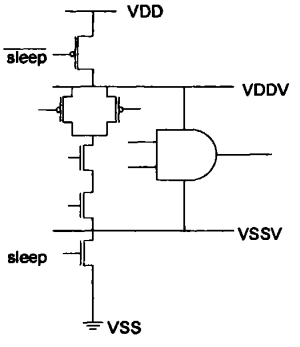


图 2- 9 Fine-Grain 结构

为了解决 Fine-Grain 所引入的以上几点问题，Coarse-Grain 结构应蕴而生，它利用开关电路控制一组单元电路与电源/地线轨道的连接，从而减小了每个单元的面积和额外的单元端口。图 2- 10 中 Coarse-Grain 结构在 stripe 和 Followpins 之间插入一个可控晶体管，代替传统供电网络直接相连的方式实现了对单元供电电压的控制。由于开关电路控制着较多单元的电源/地线的连接所以其单元数量和管子尺寸的选择都比较关键，一般可以选择使用动态 IR-drop 分析工具，来确保电路供电电压满足芯片性能要求。关于动态 IR-drop 分析将在 5.3.8.3 节予以介绍。

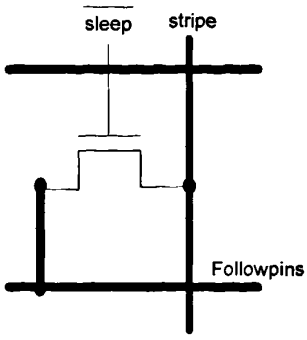


图 2- 10 Coarse-Grain 结构

Coarse-Grain 结构的电源控制单元有两种插入方式：ring switch 和 column switch。ring switch 是在电压域模块与芯片供电网络之间插入开关单元的一种方法（图 2- 11）。column switch 与图 2- 10 所反映的思想一致用开关单元控制 power rail 电源网络（图 2- 12）。此外为了避免电压重新上电产生过大的瞬态过冲电流烧毁芯片，开启时要采用缓冲链依次开启。

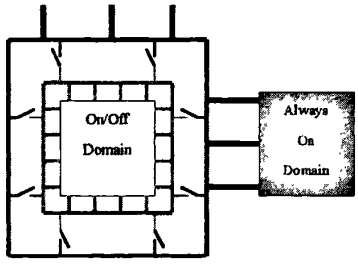


图 2- 11 Ring Switch

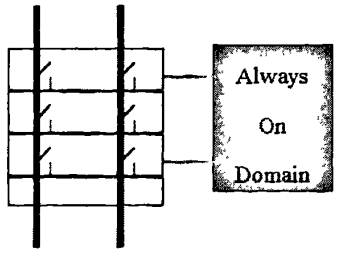


图 2- 12 Column Switch

切断时序电路的供电电压会丢失数据，目前解决这一问题较为先进的方法是采用状态保留功率阀（SRPG state retention power gating）存储单元，SRPG 存储单元属于 Fine-Grain 的一种衍生结构，它在传统的存储单元基础上做了一些改进，电路内部加入一个由高阈值器件组成的锁存电路，该锁存器引入一常通供电源来保证存储单元断电后保存其数据，待重新上电后再恢复保存值。通常主触发器（Master FF）的供电电压由 Fine-Grain 结构的电压门控制，连接正常的电压值。从触发器（Slave FF）连接一常通的低供电电压。当电压关断后数据被从触发器保存起来，如图 2- 13 所示是 SRPG 单元的原理示意图，图 2- 14 是 SRPG 单元的工作原理波形图。

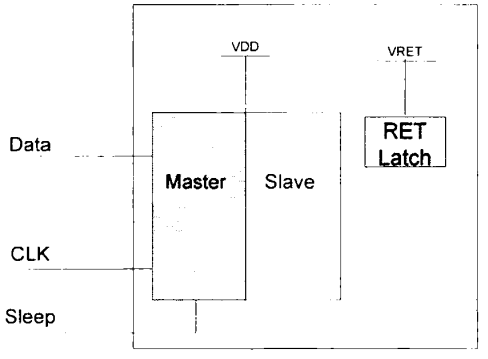


图 2- 13 SRPG 单元示意图

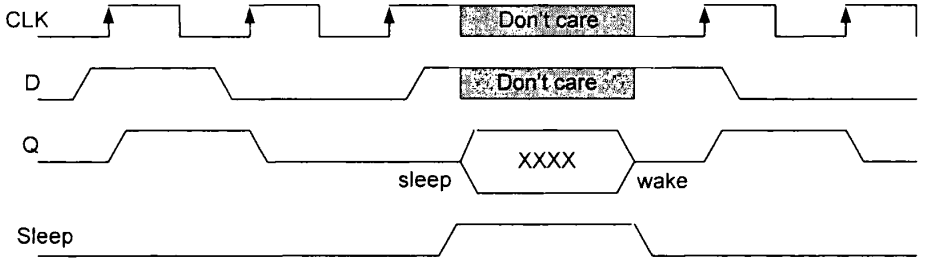


图 2- 14 SRPG 工作原理波形

电压关断技术引入的另一个问题是电压域的关断使得电路输出值处于浮空的未知状态 X，当其作为另一个处于供电状态的电压域的输入信号时可能使电路在临近信号线的干扰下状态发生跳变产生不必要的功耗。因此，电压域关断后，应在电压域之间插入隔离单元（isolation cell）以避免类似情况的发生。隔离单元可采用与门

实现（图 2- 15），将与门的一个输入端连接电源关闭控制信号，当信号有效时隔离单元的输出均为低电平，这样就避免了输出端未知状态 X 的出现。一般除复位信号外将隔离单元的输出都拉为低电平。

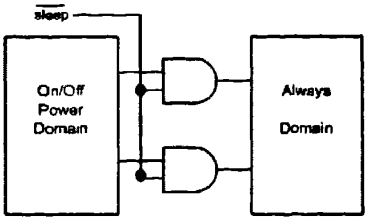


图 2- 15 插入隔离单元

2.2.3 衬底反偏电压技术

根据 pn 结理论，当 MOS 管衬底与源端处于反偏时耗尽区变厚，使得耗尽层中的固定电荷数增加。由于栅电容两边电荷守衡，因此在栅上电荷没有改变的情况下，耗尽层电荷的增加，必然导致沟道中可动电荷的减少，从而导致导电水平下降。若要维持原有的导电水平，必须增加栅压，即增加栅上的电荷数。对器件而言，衬底偏置电压的存在，将使 MOS 晶体管的阈值电压升高，NMOS 的 Vtn 更正，PMOS 的 Vtp 更负，即阈值电压的绝对值提高了。因此当电路处于闲置状态时提高衬底的反偏电压值可提高晶体管阈值减小泄漏电流。虽然反偏电压提高阈值的理论已经非常成熟，但这一技术需引入额外的电压源，而且需在三阱工艺中实现因此大大限制了它的应用，特别是当特征尺寸减小到 65 nm 及其以下工艺时，栅泄漏电流成为泄漏电流的最主要组成部分，衬底反偏电压技术在降低泄漏功耗方面的作用将不再明显。

第三章 时序库的功耗定义

时序库是定义单元时序和功耗的重要库文件，本章主要介绍库文件对功耗的定义及工具使用规则，此外还将专门针对低功耗技术使用的一些单元，如门控时钟单元、操作数隔离单元，状态保留功率单元等的时序库定义做一相关介绍。

3.1 内部功耗说明

1）内部功耗采用时序库（liberty）定义的电容单位（capacitive_load_unit），电压单位（voltage_unit）和时间单位（time_unit）计算功耗单位。假设时序库中的单位如下，

```
library (lib_example) {
...
capacitive_load_unit  (0.001pf);
current_unit: "1 uA";
time_unit: "1 ns";
voltage_unit: "1V";
leakage_power_unit: "1 nw";
... }
```

RTL Compiler 低功耗引擎使用上述定义计算内部功耗单位[7]，(公式 3-1)。

$$power\_unit = cap\_unit \times \frac{Volt\_unit^2}{time\_unit} = (0.001pF) \times \frac{1V^2}{1ns} = 1\mu W \qquad (3\text{-}1)$$

2）时序库使用关键字 internal power 定义标准单元库中各单元的内部功耗，如果时序库中的单元没有 internal power 定义则代表该单元不消耗任何内部功耗。internal power 可以基于单元级（cell level）定义，也可在单元的端口描述内（within pin group）定义。多数情况使用第二种方法。此外 internal power 内可以采用关键字 power 描述单元端口的平均内部功耗（图 3-1），也可以使用关键字 fall_power 和 rise_power 分别描述端口上升和下降时消耗的功耗（图 3-2）。计算公式见 1.3 节公式 1-3。

```
cell {name} {
    pin (name {
        internal_power () {
            power (template name)  {
                [ index_1 ( "float,  ...,  float") ; ]
                [ index_2 ( "float,  ...,  float") ; ]
                [ index_2 ( "float,  ...,  float") ; ]
                values ("float, ..., float") ;
            }  }  }  }
```

图 3- 1 端口内定义平均内部功耗

```
pin {Z} {
 direction : output ;
 function : "(A B)" ;
 internal_power () {
      rise_power (POWER_DEFAULT_C_INT) {
          index_1  (" 0.2180,  0.8120,  1.2080,  1.6040,  2.0000 ") ;
          index_2  (" 0.0059,  0.0234,  0.0468,  0.0934,  0.1401 ") ;
          values ( " 0.04778,  0.05298, 0.05595,  0.05999,  0.06251 " ,\
                  ...
                  "0.09101, 0.09485,  0.09742 ,  0.10114,  0.10362 ") ; }
       fall_power (POWER_DEFAULT_C_INT)  {...
           ...}
       related_input :  "A" ;
}
```

图 3- 2 端口内分别定义上升和下降内部功耗

3.2 泄漏功耗说明

单元泄漏功耗同样定义在时序库中，工具利用时序库的如下信息计算单元的泄漏功耗。

1）泄漏功耗单位采用关键字 leakage_power_unit 定义。

leakage_power_unit : "1 nW"

2）当单元的泄漏功耗由唯一的常数表示时，时序库利用关键字 cell_leakage_power 定义。如果 cell_leakage_power 定义不存在，工具则使用 defalut_cell_leakage_power 定义的功耗值确定单元的泄漏功耗。如果上述两种定义都不存在，工具将库中的默认泄漏功耗密度值（default_leakage_power_density）与单元面积（area）相乘计算泄漏功耗（图 3- 3）。当以上三种功耗定义都不存在时表示单元不消耗任何泄漏功耗，

```
    default_leakage_power_density : 0.0 ;
    default_cell_leakage_power     :0.0 ;
cell  (ADDFX1) {
    area : 185.6 ;
    pin {A} {
        direction : input ;
        capacitance : 0.01131 ;
    }
    pin {B} {
        ...
    }
    cell_leakage_power : 0.1458 ;
```

图 3- 3 cell_leakage_power 定义泄漏功耗

3）先进工艺的泄漏功耗值根据输入信号的不同组合状态有所不同，时序库利用关键字 leakage_power 分别定义不同输入组合的泄漏功耗。对于没有定义的输入组合状态，工具使用上述的 cell_leakage_power 值代替（图 3- 4）。

```
cell_leakage_power : 53057.365200 ;
leakage_power () {
        when : " !A & !B & ! CI " ;
        value : ... ; }
leakage_power () {
        when : " !A & !B & CI" ;
        value : ...; }
leakage_power () {
        when : " !A & B & !CI" ;
        value : ...; }
leakage_power () {
        when : " !A & B & CI" ;
        value : ...; }
...
```

图 3- 4 leakage_power 定义基于状态的泄漏功耗

3.3 时序库对特殊单元的定义

3.3.1 门控时钟单元的时序库说明

如图 3- 5 所示 clock_gating_integrated_cell 定义了集成门控时钟单元的属性，属性可由四个字符串构成，第一个字符串表示门控时钟采用哪种时序单元，有三个选项：latch, flip 和 none。第二个字符串定义时序单元是上升沿透明还是下降沿透明，用 posedge 或 negedge 表示。第三个字符串是可选项，指明测试控制逻辑（测试模式或扫描使能信号）处于时序单元之前还是时序单元之后或是不存在测试控制逻辑，可能的值有：precontrol, postcontrol 和 no entry。如果第三个字符串的定义存在那么

第四个字符串用来表示是否希望有可观测逻辑（可观测输出信号），用 obs 或 no entry 表示。另外，门控时钟的时钟输入端，使能端和时钟输出端分别用 clock_gate_clock_pin，clock_gate_enable_pin 及 clock_gate_out_pin 定义。当使用逻辑综合工具插入门控时钟时，要确保单元的 dont_use 和 dont_touch 属性为 false。此外，有些单元库采用一些简单的组合门电路，诸如与门作为门控时钟，这类单元使用 is_clock_gating_cell 定义，但如果希望 RC 能够调用组合逻辑作为门控时钟单元除具有 is_clock_gating_cell 属性外，还应在库中加入 clock_gating_integrated_cell : none_posedge 定义。

默认情况下，RC 选择具有 latch_posedge_precontrol 或 latch_negedge_precontrol 属性的单元作为门控时钟单元。

```
cell (cgprecontrol) {
    area : 1;
    clock_gating_integrated_cell : "latch_posedge_precontrol " ;
    don't_use : false ;
    pin {ENL} {
        direction : internal ;
        internal_node : "ENL" ;
    }
    pin {EN} {
        direction : input ;
        capacitance : 0.021007 ;
        clock_gate_enable_pin : true ;
```

图 3- 5 门控时钟单元的时序库定义

3.3.2 电平转换单元的时序库说明

电平转换单元（level shifter）用于 MSMV 设计，图 3- 6 中时序库中利用 is_level_shifter: true 定义其为电平转换单元。电平转换单元最多有两个输入端口，一

个信号端，一个使能端。多数情况只有一个信号输入端。库分别使用 input_signal_level 和 output_signal_level 后跟字符串的方式定义输入输出端口电压。level_shifter_enable_pin: ture 则指定端口为使能端口。当使用逻辑综合工具插入电平转换单元时要确保单元的 dont_use 和 dont_touch 属性为 false。

```
cell {LVLHLEHX2} {
    cell_footprint : lvlhleh ;
    is_level_shifter: true :
    area : 5.883500 ;
    pin {A} {
        direction : input ;
        input_signal_level : VDDH;
        capacitance : ... ;
        internal_power () {
            power_level: VDDH;
            ...
        }
    }
    pin (EN) {
        direction : input ;
        input_signal_level : VDDH ;
        level_shifter_enable_pin : true ;
        capacitance ; ... ;
        internal_power () {
            power_level: VDDH;
            ...
        }
```

```
pin {Y} {
    direction : output ;
    output_signal_level : VDDL;
    capacitance : 0.0 ;
    function : "(A EN)" ;
    internal_power () {
        power_level : VDDL ;
        related_pin : " A" ;
```

图 3-6 电平转换单元的时序库定义

3.3.3 隔离单元的时序库说明

隔离单元是多供电电压技术中需要的一种特殊单元。时序库使用 is_isolation_cell: true 指定该单元为隔离单元，此外隔离单元的一个输入端口必须使用 isolation_cell_enable_pin: true 定义其为使能端。

3.3.4 状态保留功率阀的时序库说明

如图 3-7 所示 power_gating_cell : "string" 定义单元为状态保留功率阀（SRPG），具有相同字符串（string）的单元在时序优化时可进行交换。power_gating_pin (string, "phase")定义的是 SRPG 的状态保留端口属性，string 表示状态保留模式下供电网络名称。phase 的值为 "0" 或 "1"，分别表示当该端口电压为高/低电平时，电路进入睡眠（sleep）模式。

```
cell ( SRPG_HD_H_SDFPRQXS ) {
    area : 10.800 ;
    Power_gating_cell : "srpg_dff1" ;
    pin ( pg ) {
        direction : input ;
        power_gating_pin (power_pin_1, "0");
```

图 3-7 状态保留功率阀时序库定义

第四章 DTMF 接收器设计与实现环境

双音多频（Dual Tone Multiple Frequency）信号是业界标准通信信号，普遍用于程控电话拨号信息的传输。DTMF 信号由一组低音频信号和一组高音频信号以一定方式的组合构成。它将拨号盘上 0~9，A~D，星号（*）和井字号（#）共 16 个字符用 4 个高音频信号和 4 个低音频共 8 个音频的组合进行表示，高音频的标准值分别为：1209Hz，1336Hz，1477Hz，1633Hz。低音频的标准值分别是：697Hz，770Hz，852Hz，941Hz。每个信号由一个来自高音频和一个低音频的正弦信号叠加而成[4]。按键“4”的 DTMF 信号产生示意图如图 4- 1 所示。

DTMF 信号拨号方式速度快，误码率低，可靠性高等优点有着广泛的应用。本文涉及的是 DTMF 信号的接收电路。

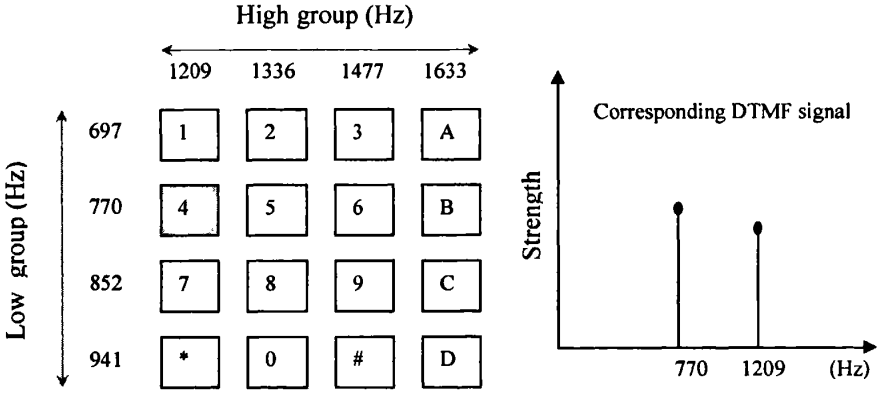


图 4- 1 DTMF 信号产生示意图

4.1 DTMF 信号接收器

DTMF 接收器的原理图如下所示（图 4- 2），串行接口接收经过“U-律”压缩处理的 PCM 串行编码，每收到 8 个 bit，即一个字节交由 DMA 模块，DMA 根据内存总线访问裁决器的信号决定是否可以将数据送往内存。8 比特 U-律 PCM 数据在送往内存前需将其转换成 16 位线性 PCM 编码，再由数据选择模块将其写入内存。DSP 模块读取内存数据，通过 Goertzel 算法进行数字信号处理，处理后的 16 位数据

由结果特征转换单元进行处理从而得到最终由 8 位二进制数表示的 ASCII 格式数据。DTMF 信号接收器的相关参数见（表 4- 1）。

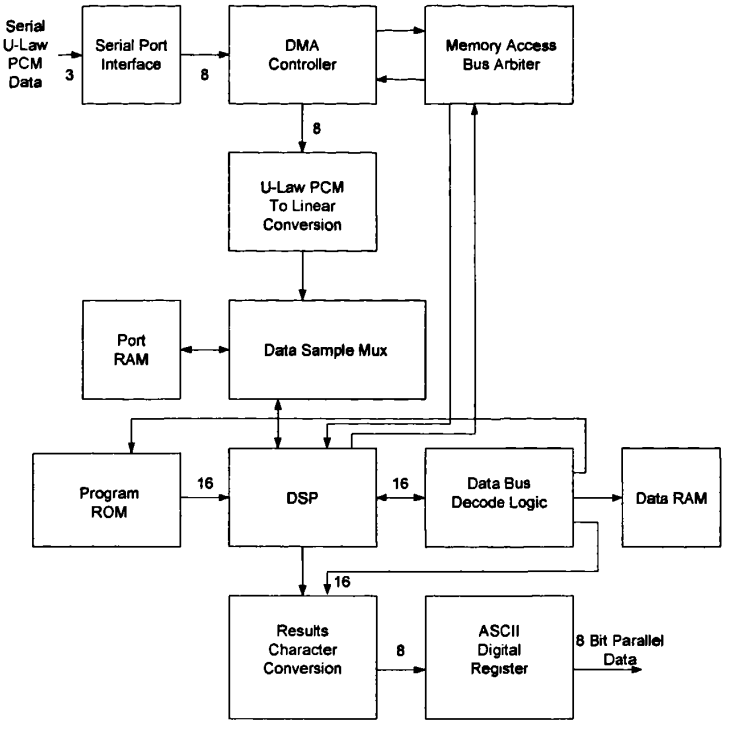


图 4- 2 DTMF 接收器原理框图

标准单元个数	7000+门
I/O 个数	61 个
工作频率	166 MHz

表 4- 1 DTMF 接收器单元个数

4.2 设计准备

DTMF 接收器的 RTL 级 Verilog 硬件描述语言已经实现并通过功能验证。本文将基于 RTL 描述，结合几种低功耗方法实现从逻辑综合到布局布线再到最终 sign-off 验证的完整后端设计，总结完善基于 Cadence Soc Encounter 数字集成电路设计平台的低功耗设计流程以供工程参考。

4.2.1 硬件环境

- Linux Server：IBM e326m

- 内存：15GB (free)
- 处理器个数：4 个 Dual Core （AMD Opteron™ Processor 270）
- CPU 速度：2.0 GHz

4.2.2 软件环境

- 操作系统：Red Hat Enterprise Linux WS release 3 <Taroon Update 5> Kernel 2.4.21-37. Elsmp on an x86_64
- EDA 配置（表 4- 2）：

EDA 功能	EDA 工具	版本
逻辑仿真工具	Cadence® NC-Verilog® Simulator	IUS56QSR1
逻辑综合工具	Encounter™ RTL Compiler	RC62USR1
物理实现工具	SoC Encounter	SOC52
寄生参数提取工具	Fire & Ice QX	EXT511
电压降分析工具	VoltageStorm PE Cell-Level Rail Analysis	ANLS611
噪声延时分析工具	CeltIC NDC	SOC52

表 4- 2 工具配置

4.2.3 物理单元库

本设计基于 SMIC 90 nm 低泄漏逻辑标准单元库，采用 9 层铜互连工艺，两层顶层金属。

物理单元库（LEF）包括三类：标准单元物理库，I/O 物理库以及模块（SRAM）物理库。其中，标准单元物理库又包括正常和高阈值标准单元库，电压转换单元物理库，见表 4- 3。

LEF (P&R Abstract Views)	Cells
smic90tech_9m.lef	TECH LEF (9M)
smic_009_std.lef	STD Cells
hs_ll_hvt_v12_v06p.lef	STD Cells With High Threshold
ISLN_LVLSHFT.lef	Low Power Multi-VDD Kits
SPSRAM90n256x16.lef	SRAM - SPSRAM90n256x16
SPSRAM90n512x16.lef	SRAM - SPSRAM90n512x16
SP90NLLD2_V0p1_9MT.lef	I/O (9M)

表 4- 3 物理单元库

4.2.4 时序单元库

时序单元库（.lib）物理单元库（LEF）相互配片使用。它包括最快，典型，最慢三种仿真环境下的时序模型：1.2V 正常/高阈值标准单元逻辑、静态随机存储器 、IO；1.0V 正常阈值标准单元逻辑、1.2V~1.0V 电压转换单元，见表 4- 4[8]。

Timing Libraries (SS)	Cells
scc090nll_v10_12_ss125_lpmk.lib	Low Power Multi-VDD Kits (1.0V - 1.2V)
scc090nll_v10_ss125.lib	Std Cells (1.0V)
scc090nll_v12_10_ss125_lpmk.lib	Low Power Multi-VDD Kits (1.2V – 1.0V)
scc090nll_hvt_v12_ss125.lib	High Threshold Std Cells (1.2V)
scc090nll_v12_ss125.lib	Std Cells (1.2V)
SPSRAM90n256x16_SS.lib	SRAM (1.2V)
SRSRAM90n512x16_SS.lib	SRAM (1.2V)
SP90NLLD2_V0p1_max.lib	I/O (1.2V/2.5V)
Timing Libraries (TT)	Cells
scc090nll_v10_12_tt25_lpmk.lib	Low Power Multi-VDD Kits (1.0V - 1.2V)
scc090nll_v10_tt25.lib	Std Cells (1.0V)
scc090nll_v12_10_tt25_lpmk.lib	Low Power Multi-VDD Kits (1.2V – 1.0V)
scc090nll_hvt_v12_tt25.lib	High Threshold Std Cells (1.2V)
scc090nll_v12_tt25.lib	Std Cells (1.2V)
SPSRAM90n256x16_TT.lib	SRAM (1.2V)
SPSRAM90n512x16_TT.lib	SRAM (1.2V)
SP90NLLD2_V0p1_typ.lib	I/O (1.2V/2.5V)
Timing Libraries (FF)	Cells
scc090nll_v10_12_ff-40_lpmk.lib	Low Power Multi-VDD Kits (1.0V - 1.2V)
scc090nll_v10_ff-40.lib	Std Cells (1.0V)
scc090nll_v12_10_ff-40_lpmk.lib	Low Power Multi-VDD Kits (1.2V – 1.0V)
scc090nll_hvt_v12_ff-40.lib	High Threshold Std Cells (1.2V)
scc090nll_v12_ff-40.lib	Std Cells (1.2V)
SPSRAM90n256x16_FF.lib	SRAM (1.2V)
SPSRAM90n512x16_FF.lib	SRAM (1.2V)
SP90NLLD2_V0p1_min.lib	I/O (1.2V/2.5V)

表 4-4 时序单元库

4.2.5 DTMF 实现结构

为提高设计效率 DTMF 均采用脚本实现，结构如图 4-3 所示，前端设计按照四种设计方案分为四个目录，每个方案均依据单元库、设计数据、RTL 代码、逻辑仿真和逻辑综合功能分成六个工作目录。对于后端设计而言尽量采用相同的布局方案以最大化的降低由于布局差异的不同产生对功耗的影响，设计同样按照四种方案分为四个工作目录，而对于单元库，设计数据，脚本等将作为公共数据供工具调用。关于方案目录的名词解释可以参考第六章表 6-1。此外由于设计流程复杂，涉及的工具和数据较多并且目录结构与具体设计关系密切，因此这里将只列出三级目录结构仅供设计参考。

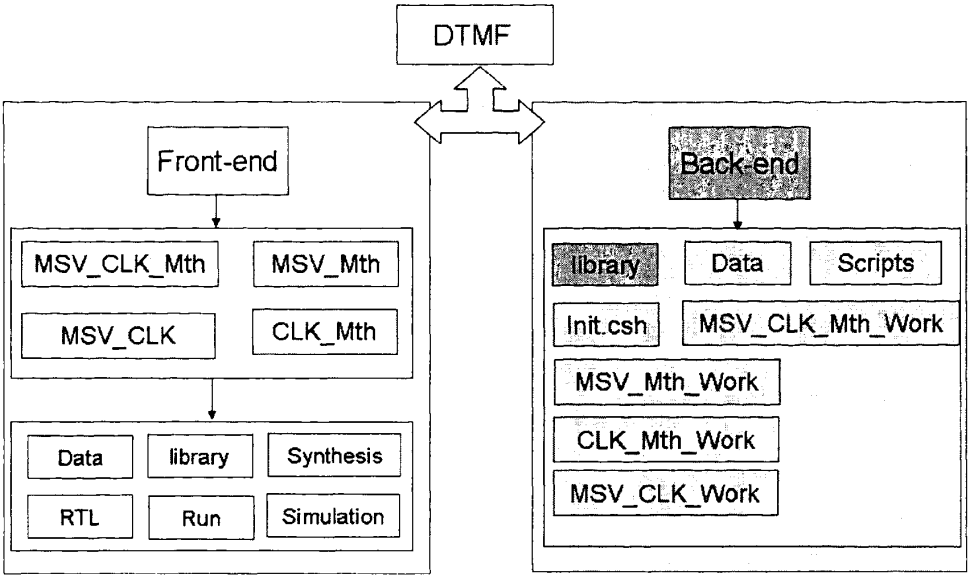


图 4- 3 DTMF 实现结构

第五章 DTMF 低功耗设计技术实现

本章介绍使用 RTL Compiler 逻辑综合工具，结合多供电多电压技术，门控时钟技术，操作数隔离技术和多阈值单元组合技术对双音多频接收器进行逻辑综合，并用 Encounter 实现电路的后端物理设计，对工具的主要命令给予了必要的解释。此外介绍了基于 Encounter 平台和 ETS 工具的 sign-off 验证流程和方法，使电压降，地电压升高，信号噪声和电迁移等影响电路性能和可靠性的问题得以解决。

5.1 MSMV 设计方法学概述

在第二章简要介绍了几种低功耗实现技术，在设计中可以根据实际需要选择不同方案。MSMV 是低功耗设计中较为先进的一种技术，该技术还可同时支持门控时钟，操作数隔离，多阈值单元组合等技术，因此下面将以 MSMV 技术为主线并结合上述低功耗方法介绍它们在 Cadence 工具环境下的实施过程。

前端设计：逻辑综合是将电路的 RTL 描述利用综合工具和标准单元库转化成门级电路的过程，这一步骤是后端物理设计的基础，也是低功耗技术实现的重要环节。MSMV 最主要的实现手段是电压域的创建和划分，电压域就是把整个电路按不同供电电压化分成的多个子模块，每一个子模块可看作一个独立的电压域，拥有独立的供电网络。在逻辑综合阶段创建电压域不仅在功能上满足了设计要求，而且为后端电源网络设计提供了便利。电压域创建后最重要的一点是要把时序库模型对应到相应电压值的电压域下（属于同一电压域的所有时序库也被称作时序域），以使逻辑综合工具能够正确的计算电路时序，选取正确的逻辑单元。

门控时钟和操作数隔离单元在逻辑综合阶段插入，插入前向工具明确单元库中所选门控时钟和操作数隔离单元的名称。level shifter 是不同电压域间信号电平的转换电路， RC 将其插入在两个电压域之间。为了得到较精确的功耗数据，RC 可使用逻辑仿真工具产生的 TCF（toggle count format），SAIF(switching active interchange format)或 VCD（value change dump）三种包含信号跳变信息的数据格式文件分析优化电路功耗。综合后工具得到两个最基本的数据文件：门级网表和设计约束文件（synopsys design constrains）。为保证网表与 RTL 逻辑功能一致可以采用形式验证

的方法对网表予以检查，由于我们对网表进行了门级仿真，可以保证其逻辑功能，因此跳过等效性检查而直接交由后端。

后端设计：布图阶段在核心区创建相应的电压域区域，电压域只包含属于该域的模块和单元。与逻辑综合类似不同的电压域要与相应电压的时序库对应，这一点很重要。多电压设计除了通常电源网络所必需的 Core Ring/Stripe 和 Block Ring/Stripe，一般还要给电压域设计自己的 Ring 和 Stripe 使每个电压域有良好的供电网络。布局阶段需强调的一点是，布局前要向工具指明设计中用到的电压转换单元以及其所连接的两个不同电压值的电压域和单元所处的电压域区域，这样布局工具就会自动地把 level shifter 放置在指定电压域的边界处提高布线质量。关于物理设计阶段电压域的创建和 level shifter 的摆放都可由逻辑综合工具输出的相应文件产生，物理设计工具根据这些文件可自动完成相关操作。

为了在时钟树综合（clock tree synthesis）阶段进一步达到降低功耗的目的，可以采用低功耗的时钟树综合技术（LP-CTS），这种技术利用最优化的时钟门布局，克隆与反克隆等方法，在确保时钟功耗优化的同时满足时序和物理目标的双重需求，达到更好的负载分配将时钟偏差降低到最小。

MSMV 布线与单一电压布线不同是，同一电压域内单元之间的互连要尽量在同一域内实现，而电压域间单元的互连则应避免跨过多的电压域。如图 5- 1 所示，电压域 PD1 中单元 A 与 B 的互连只在 PD1 中实现，电压域 PD3 中单元 C 与 D 的互连也只在 PD3 中实现。属于不同电压域的 B，C 单元的互连通过默认的电压域而没有经过电压域 PD2。当然这种布线方法的实现依赖于 EDA 工具。

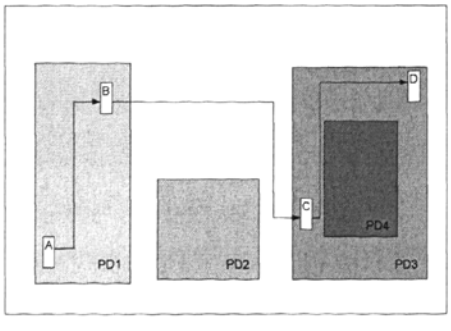


图 5- 1 考虑电压域的走线原则

多电压技术使得电路中的某些模块使用了较低的供电电压，因此更容易产生由IR-drop效应所引起的时序违反，对于静态IR-drop效应可以采用改进优化电源网络的ECO方法解决，动态IR-drop效应可以使用去耦电容（decouple capacitance）。去耦电容处于电源和地线之间，当某一时刻电路中大量单元同时发生信号状态的翻转引起充放电流增大导致电源轨道电压下降，这种情况下去耦电容像一个备用电池依靠平时存储的电荷向电源线提供额外的电流减小动态IR-drop效应对时序的影响（图5-2）。从另一方面来讲，插入去耦电容会增加电路的泄漏功耗，所以在什么地方插入，插入多少去耦电容都要仔细的规划，在电路性能与功耗间进行折衷。

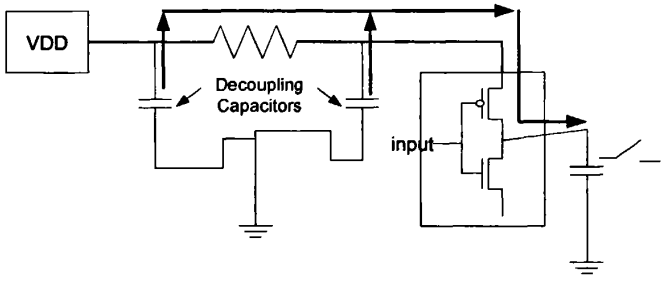


图5-2 去耦电容工作原理

5.2 MSMV逻辑综合

DTMF接收器的逻辑综合采用Encounter™ RTL Compiler工具实现，除采用MSMV技术外还将结合门控时钟，操作数隔离和多阈值单元技术[9~11]。

创建时序域：设计使用1.2伏和1.0伏两个不同时序域。

使用变量名替代时序库名（temperature 125℃）

```
set  v12_list  { \
                scc090nll_hvt_v12_ss125.lib \
                scc090nll_v12_10_ss125_lpmk.lib \
                scc090nll_v10_12_ss125_lpmk.lib \
                scc090nll_v12_ss125.lib \
                SPSRAM90n256x16_SS.lib \
```

```
            SPSRAM90n512x16_SS.lib \
            SP90NLLD2_V0p1_max.lib \
           }
set v10_list  { \
            scc090nll_v10_ss125.lib \
         }
```

创建 1.2V 和 1.0V 时序域 pd_vdd12 和 pd_vdd10，将单元的时序库与相应时序域对应。

```
create_library_domain {pd_vdd12  pd_vdd10}
set_attribute library  $ v12_list  pd_vdd12
set_attribute library  $ v10_list  pd_vdd10
```

注意使用 set_attribute library *library_list library_domain* 命令指定的第一个时序域 *library_domain* 将成为默认的时序域。当某一模块的时序域被删除后该模块将对应到默认的时序域下。另外当时序域所包含的时序库具有不同的操作环境时，第一个时序库的操作环境为时序域的默认操作环境，使用下面的命令可改变默认的时序域。

```
set_attribute  default true desired_library_domain
```

PLE：为更准确地在逻辑综合阶段考虑互连线寄生参数对时序的影响，RC 的“物理版图预估”（physical layout estimator）功能通过读入 LEF 和电容表文件代替 RC 技术库自带的线负载模型（wire-load model），这种方法使生成的网表在物理设计阶段时序更易收敛。

```
set_attribute  lef_library  $LEF_LIBRARY
set_attribute  cap_table_file  $CAP_TABLE
set_attribute  interconnect_mode  ple
```

RTL 级逻辑功耗分析：在网表映射前可使用 report power –rtl –detail 命令对 RTL 级代码做功耗预估，该功能的实现需要在 RTL 代码编译（elabration）前指定下列三条命令。

set_attribute lp_power_analysis_effort {high|medium|low} /

set_attribute lp_power_unit nW /

set_attr hdl_track_filename_row_col *true* /

插入门控时钟：启用门控时钟插入技术必须在 RTL 代码编译操作之前设置以下的操作命令开启该项功能。RC 可以根据 RTL 的代码风格自动插入门控时钟单元。

set_attribute lp_insert_clock_gating true /

set_attribute lp_clock_gating_prefix clock_gating_ /

设置一个门控时钟至少控制 4 个最多控制 8 个存储单元。

set_attr lp_clock_gating_min_flops 4 /des*/*

set_attr lp_clock_gating_max_flops 8 /des*/*

指定门控单元：如果时序库中含有集成门控时钟单元或用户定制的门控时钟模块，使用下面的命令指定单元名称和要插入的门控时钟电路模块。

set_attribute lp_clock_gating_cell *path_name_for_cell* {*design*|*subdesign_list*}

set_attribute lp_clock_gating_module *module_name* {*design*|*subdesign_list*}

插入操作数隔离：启用操作数隔离技术同样必须在 RTL 代码编译操作前设置下列命令开启该项功能。RTL 编译时 RC 低功耗引擎会鉴别出数据通路模块并自动加入操作数隔离单元，值得注意的是 RC 只会在输入位数大于等于 8 的数据通路加入操作数隔离单元。

set_attribute lp_insert_operand_isolation true /

set_attribute lp_operand_isolation_prefix operand_isolation_ /

应用外部设置的建立时间和保持时间检查门控时钟单元的时序。

set_clock_gating_check -setup 1.0 ns -rise *$object*

set_clock_gating_check -hold 0 ns -rise *$object*

使用多阈值单元组合技术

set_attribute lp_multi_vt_optimization_effort medium /

根据泄漏功耗的要求，可以选择三种不同的优化级别 low，medium 和 high。RC 默认为 low，此时 RC 使用所有的单元库进行映射，重映射和递增映射。medium：工具只有在重映射，递增映射和时序不满足时才会选择低阈值单元。high：工具只有进行递增映射和在时序不满足时才会选择低阈值单元（图 5- 3）。值得注意的是 medium 和 high 会引起电路面积的增大，所以选择时要在功耗和面积间进行权衡。

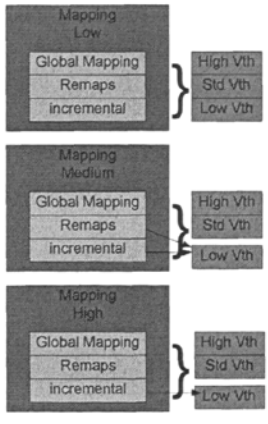


图 5- 3 泄漏功耗优化单元映射原则

动态功耗优化：比较电路动态功耗与规定的最大动态功耗对电路进行优化，以满足功耗要求。

set_attribute max_dynamic_power *power_constraint*

泄漏功耗优化：与动态功耗优化类似，指定最大泄漏功耗值。为使工具最大化地降低泄漏功耗，可将值设为 0。

set_attribute max_leakage_power *power_constraint*

如果泄漏功耗与动态功耗的约束同时施加，默认情况下 RC 把对泄漏功耗优化的优先级排在动态功耗优化之前。如果希望提高动态功耗优化的优先级，可使用命令 set_attribute lp_optimize_dynamic_power_first *true*。此外使用权重因子 w (0~1 之间) 也可以调节工具优化两类功耗的优先级。权重因子 w 根据公式 5-1 得出功耗优化的权重。当使用权重因子时，动态功耗优化优先级设定命令将被忽略。此外如果时序库使用基于输入信号状态的泄漏功耗模型（见 3.2），为增加泄漏功耗预估的准确性可以读入跳变活动信息。

set_attribute lp_power_optimization_weight *weight*

$$wighted\_power = w \times leakage\_power + (1-w) * dynamic\_power \qquad (5\text{-}1)$$

时序域与模块映射：MSMV 中不同供电电压的模块要与先前定义的时序域对应，这一点非常重要。对应的顺序也有一定要求，即从顶层模块开始从顶层到底层依次建立对应关系。这里要与前面介绍创建时序域时提到时序库与时序域的对应区别开，这里是将模块化划分到不同的时序域中。

set_attribute library_domain *library_domain design (top_design)*

set_attribute library_domain *library_domain design (sub_design_A)*

set_attribute library_domain *library_domain design (sub_design_B)*

插入 level shifter：电压转换单元插入前要指明插入单元的名称和所连接的两个时序域，如下命令所示。

define_level_shifter_group -from_library_domain *library_domain*

-to_library_domain *library_domain* [-name *string*]

-libcells *cell_list*

电压转换单元只能插入在两个不同值电压域之间，凡不满足该条件 RC 均不予以操作。如果 dedicate_level_shifter 选项不指定具体单元名称，工具从上述定义电压转换单元组命令的-libcells 选项中选取单元。

level_shifter insert [-from_library_domain *library_domain*]

[-to_library_domain *library_domain*] [-location {*from* | *to*}]

[-dedicate_level_shifter *libcell*]

检查插入的电压转换单元并输出报告。

level_shifter check

report level_shifter -detail

读入信号跳变数据文件：向 RC 提供信号跳变数据能够得到更精确的电路功耗。RC 支持三种数据格式 TCF（toggle counter format），SAIF (Synopsys switching activity interchange format) 和 VCD (value change dump)。

read_tcf *path/file_name*

read_saif *path/file_name*

read_vcd *path/file_name*

设计者可以覆盖信号跳变信息文件中某些信号线的跳变数据重新进行设定，RC 提供了下列两条命令 1）设定信号线逻辑为“1”的概率，2）设定信号翻转率。

set_attribute lp_asserted_probability *float* /designs/*design*/*/nets/*net*

set_attribute lp_asserted_toggle_rate *float* /designs/*design*/*/nets/*net*

如果某些信号线的跳变信息没有被记录到跳变文件中，命令 set_attribute lp_power_analysis_effort 设置的努力程度（low/medium/high）将决定 RC 采用哪种方法计算功耗，一种情况，如果努力程度为 medium/high，工具把已有信号线

（fanin cone）的跳变活动值传递到未知值信号线上作为未知信号线的值。另一种情况，如果命令设置的努力程度为 low 或者 fanin cone 内的信号线同样没有指定跳变活动值，RC 将使用默认概率 0.5 和默认每纳秒 0.02 的翻转率分析功耗，此外还可以用下列命令设置默认跳变信息。上述三种努力程度在功耗分析精确度和工具运算的速度上有所不同，low 时工具内存消耗小，计算速度快但精确度不高，相反 high 时功耗计算准确但要占用较大的内存，花费较长时间。

set_attribute lp_default_probability *float* /designs/*design*

set_attribute lp_default_toggle_rate *float* /designs/*design*

set_attribute lp_toggle_rate_unit *toggle_unit*

综合：施加电路设计约束，使用 synthesize 命令对 RTL 代码进行逻辑综合。综合过程分两步完成，第一步，参数 to_generic 利用内部通用逻辑对 RTL 代码进行优化处理，如逻辑结构优化和去处冗余逻辑等操作。第二步参数 to_mapped，利用标准单元库将 RTL 代码映射成门级网表。映射结果按“努力”（effort）程度有三种选择，分别是 low, medium 和 high。RC 默认的努力程度为 medium。

设计报告：逻辑综合结果的质量用下列命令输出面积，时序，功耗等报告进行分析。

1）面积报告：report area

2）时序报告：report timing

3）功耗报告：report power

4）电压转换单元报告：report level_shifter -hier –detail

P&R 接口文件：逻辑综合的最后一步是导出后端布局布线工具所需的数据文件，包括门级网表和设计约束。

1）输出门级网表：write_hdl

2）输出设计约束：write_sdc

此外，RC 还专门提供了与 Cadence SoC Encounter 平台的接口命令 write_encounter，除输出门级网表和设计约束外还包括 Encounter 的输入配置（.conf）模板文件，实现 MSMV 技术相关命令文件（*msv.setup），电压转换单元模板文件（.vsf）[5]。

从上述命令中我们可大致较为清楚地了解 MSMV, 多阈值单元组合，门控时钟和操作数隔离技术在 RC 中的实现方法。针对其它低功耗组合技术只需把相关命令做一些简单调整即可实现，这里就不再赘述。

5.3 MSMV 物理设计

5.3.1 物理设计输入数据

物理实现[12~15]离不开单元库及各类数据文件的支持，基于 Encounter 平台的后端物理设计需要如下一些库文件及数据文件：

1）Verilog 门级网表文件

2）LEF（Library Exchange Format）物理库文件

3）Liberty 时序库文件

4）SDC（Synopsys Design Constraints）设计约束文件

设计输入还可包括下列可选数据：

1）输入输出单元（I/O PAD）的物理位置文件（I/O assignment file ）

2）Sign-off 寄生参数抽取技术文件（technology file）

3）Encounter RC-extraction 工具寄生参数抽取电容表文件（capTable）

4）寄生参数提取和 IR-drop 分析所需的单元库数据（cell library）

5）用于信号完整性分析的噪声库文件 (.cdB)

6）其它内容：a) 用于时钟树综合（CTS）和原地优化（IPO）的单元 footprint 名称，b) 电源，地线网络名称，c) 寄生参数提取的比例因子（可选）。

需要特别指出的是对于 MSMV 的实现还可以准备以下两个文件：1）创建和划分电压域的初始文件。2）指定电压转换单元位置属性文件。这两个文件实际是 TCL 命令的批处理文件，关于其语法及使用将在 5.3.2 节予以介绍。为了提高数据的输入效率，Encounter 平台提供了以 conf 为扩展名的输入配置文件，设计者只要将数据存放路径和文件名称写入文件对应项，工具就可以自动导入指定的数据。

encounter> loadConfig *$path/config.conf* 0

encounter> commitConfig

5.3.2 创建电压域

利用 createPowerDomain 命令创建电压域，每个电压域有独立的名称并与相应的时序库对应。

createPowerDomain *Name* –maxTimingLibs *timing_lib_lists* -minTimingLibs *timing_lib_lists*

loadShifter 命令装载电压转换单元。 loadShifter -infile *file.vsf*

检查电压域创建和电压转换单元装载情况，输出报告。

verifyPowerDomain –xNetPD *file.rpt* -isoNetPD *fileName.rpt*

向各个电压域分配模块，并指定模块的电源地端口与电源网络的连接关系。

modifyPowerDomainMember *pdName* –instance *instancelist* –power{VDD: *VDD1 VDD2...*} –ground {GND: *GND1 GND2...*}

5.3.3 时序分析环境设置

时序收敛是后端设计人员最关注的一个问题，Encounter 提供的硅虚拟原型（SVP silicon virtual prototyping）为设计人员在早期发现问题提供了可能，SVP 是一

种设计方法学，不能简单地将它理解为一种快速设计和验证布图的技巧，在后面的介绍中我们将看到 SVP 的概念贯穿于后端的整个流程。

逻辑综合得到的门级网表能否通过后端第一次时序检验是设计者非常关心的问题，因此布图前可以先对没有进行任何“加工”的原始设计进行一次初步的时序检查，检查前要设定好时序分析所需的工作环境，寄生参数比例因子等影响静态时序分析 (STA，Static Timing Analysis) 结果的相关参数。

MSMV 设计要针对电压域分别设置工作环境。

setOpCond -powerDomain *domainName* –min *minOpCond* –max *maxOpCond*

指定分析建立时间和保持时间所使用的时序库, -min 指定保持时间分析，-max 指定建立时间分析。Min 和 Max 分别指最好情况和最坏情况。

setTimingLibrary –min {Min | Max} –max {Min | Max}

Eg: setTimingLibrary –min Min -max Max （保持时间分析用最好情况, 建立时间分析用最坏情况)

考虑片上误差（on-chip variation）和 IR-drp 效应，建立时间分析 early path 设置负 10% 误差，保持时间分析 late path 设置正 10%误差以增加时序约束，一般误差范围设置在 5%-10%之间，（图 5- 4）。

setTimingDerate -max -early *value(0.9)* -late *value(1.0)*

setTimingDerate –min -early *value(1.0)* –late *value(1.1)*

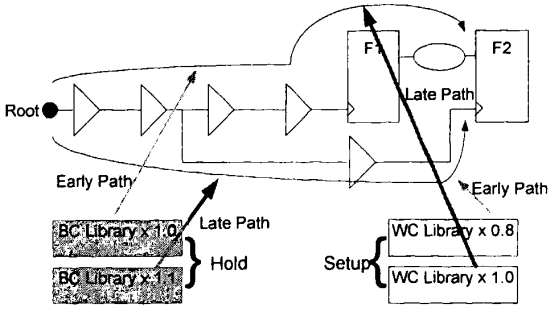


图 5- 4 建立保持时间分析的 Early Path 与 Late Path

为了使 Encounet 寄生参数提取工具与 sign-off 寄生参数提取工具的抽取结果更好地拟合（correlation），分别设置默认模式(default 速度快，不包含耦合电容信息），详细模式（detail 包含耦合电容信息，影响 SI 效应）的电容比例因子，电阻（resistor），交叉耦合电容（cross-coupling capacitance）比例因子。同时为保证 Encounter 寄生参数提取工具的精度，提取前需提供电容表文件，该文件可从 foundry 获得，如果没有电容表也可借助 foundry 的互连线技术格式文件 (ICT interconnect technology format) 和 generateCapTbl 工具产生得到。

generateCapTbl -ict *fileName.ict* -output *fileName.CapTbl*

setRCFactor –defcap *valule* –detcap *value* –xcap *value* –res *value*

由于以上四项比例因子的数值与工艺参数和设计有关，设计之初通常无法准确知道设置多少为宜，因此对一个新的设计来说可以采用 Encounter 下的 Ostrich 工具计算比例因子。先对设计做一个快速的布局布线，分别用 Encounter 寄生参数提取工具和 Sign-off 寄生参数提取工具对设计进行 RC 抽取，得到两个 SPEF 文件，再将得到的这两个 SPEF 文件交由 Ostrich 工具进行对比，从而获得针对该工艺和设计的比例因子。其中 –all 选项表示获得 default RC 和 detail RC 因子，useOstrich 选项表示使用 Ostrich 工具，默认情况下 generateRCFactor 命令调用 spefCapCmp Perl 脚本进行比较。

setPlaceMode –fp

placeDesign

trialRoute

runQX

rcOut *fileName1.spef*

generateRCFactor -all -spefIn *fileName1.spef* - useOstrich

考虑金属填充物对互连线寄生电容的影响，具体值的设置参考晶园厂对金属密度（metal density）的要求，取稍大于平均值即可。

setExtractRCMode –assumeMetFill *valule(0-1)*

设置时序分析模式，通常采用最坏情况分析建立时间，最好情况分析保持时间。此外为了避免 OCV 带来的“悲观时钟收敛”问题，设置分析模式时采用“悲观时钟收敛消除”方法（Clock Reconvergence Pessimism Removal，CRPR）予以解决。CRPR 也称作 CPPR（Common Path Pessimism Removal)。

setAnalysisMode –bcWc -crpr

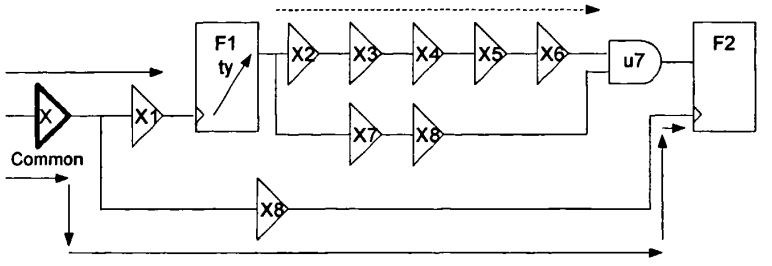


图 5- 5 时钟收敛悲观移除（CRPR）原理

时序分析条件设置完成后，进行布图布局前在不考虑互连线负载的情况下分析电路是否满足时序要求。如果时序报告中存在建立时间违例，那么在后期的设计中实现时序收敛会变得比较困难。这种情况要及时查找问题的原因，如果是由于逻辑综合约束过紧造成的则需要返回上一步重新综合或是修改 RTL 代码。若时序没有问题就可以进行下一步布图与布局设计。timeDesign 命令将实验布线（trial Route），寄生参数提取，静态时序分析和设计规则违例分析功能组合在一起，使时序分析变得非常容易。

timeDesign –preplace

timeDesign –preplace -hold

5.3.4 布图规划

布图规划(floorplan)质量的好坏直接影响到芯片面积、时序收敛和后期设计的难易程度，布图规划阶段一般包括 3 个主要工作：1）die size 设计；2）IO 和宏模块的

摆放；3）供电网络设计。决定布图规划的三个重要参考因素是IO位置，模块的位置和信号流。IO的摆放顺序和设计对ASIC来说一般已经在方案之初由系统工程师制订完成，因此设计人员只要根据SPEC编写正确的IO物理位置文件，导入Encounter，各个IO就可以按要求自动进行摆放。虽然这种方法减少了设计人员布图阶段的工作量，但在某种整程度上说也增加了设计潜在的难度，使设计自由度受到限制。本设计中共有61个IO单元，其中芯片左边一对是供给芯片内部1.2V正常供电电压IO，芯片右边一对是供给低电压1.0V IO。关于供电I/O数量的选择，将在本节供电网络设计部分介绍。需要特别强调的是，在MSMV设计中由于多个供电电压的存在，不同供电I/O之间要用特殊的I/O单元隔离，这与数字I/O和模拟I/O相互隔离类似。

宏单元的面积一般较大，它是影响芯片面积和走线难易的主要因素。Encounter工具提供了自动和手工布置宏单元的方法。自动布置的实现依赖于种子（seed）文件，只要把希望工具自动布置的所有宏单元名称逐行写入一文本文件（该文件习惯以seed作为扩展名），导入并执行planDesign命令，工具就可以实现宏单元的自动摆放。自动摆放主要以数据流向为依据，设计之初大量模块的摆放不仅繁琐而且难以找出适合的位置，因此选择这种方法不仅能提高工作效率而且为设计者提供了设计思路。工具自动摆放完成后再根据实际情况对结果进行手工优化。

自动布置宏模块命令：

planDesign –seed *filename.seed*

手工布置可精确地控制宏模块的位置，也是布图普遍采用的方法，飞线是布图非常有用的工具，它提供给设计者有关模块间的布线连接信息，利用鼠标移动模块时飞线的走向会随着模块的位置而变化，通过飞线判断模块摆放位置的原则是尽量避免飞线过多的交叉，应使相关模块之间的飞线看上去干净整齐，同时避免相关模块摆放距离较远而引起潜在互连线过长导致的时序收敛问题。

在本设计中共有3个宏模块，分别是两个256Kx16位的嵌入式RAM和一个512Kx16的嵌入式ROM.。根据IO位置和飞线的判断比较（图5-6），选择将三个存储器并列放置在芯片的左上角区域。relativeFPlan命令提供了精确的物理定位功

能，利用参考物的相对位置控制宏单元的摆放和朝向。此外，modifyPowerDomainAttr 命令可调整电压域的形状大小和位置。

relativeFPlan --relativePlace

relativeFPlan --relativePlace *domain_name*

modifyPowerDomainAttr *domain_name* -box llx lly urx ury

modifyPowerDomainAttr *domain_name* -minGaps T B L R

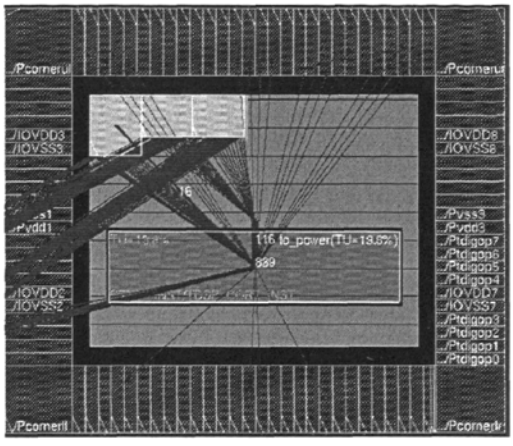


图 5- 6 飞线与物理单元位置

IP 宏模块的特点是端口通常处在模块的外围边界利于互连，而模拟模块除端口外其它区域会做成不可布线区域，避免因信号线从模块顶部穿过带来干扰，影响正常功能。数字模块的抗干扰性较强，有时会允许高层互连线从顶部通过。因此，宏模块边界处互连线密度较大，非常容易造成布线资源紧张。解决方法是在宏模块周围设置不可布局区域，工具把这一区域称作 halo。halo 的大小根据情况自由设置，区域内工具不会放置标准单元。

addHaloToBlock 20 20 20 20 -allBlock

供电网络设计可以采用以下的经验方法进行预估。以 SMIC 90nm 单元库为例，见表 5- 1 是与电源网络设计有关的物理数据。

逻辑综合统计功耗 (mW)	5.8
低电压域单元数目/总单元数目	4880/6746=0.72
I/O 平均供电电流 (mA)	137
互连金属最大电流密度(mA/um)	1.38 (layer 1), 1.63(layer2~7) , 4.57(layer8~9)

互连金属最小宽度(um)	0.12 (layer 1)，0.42（layer 8~9）
允许最大互连金属宽度 (um)	12 (layer1~8), 10 (layer 9)

表 5- 1 电源网络设计相关数据

逻辑综合统计功耗 5.8 mW, 考虑 50%功耗裕量，预估总功耗 8.7 mW。其中低电压域单元所占总单元比例为 72%，近似得出低电压域（1.0V）消耗功耗 6.26 mW, 所需电流 6.26mA，正常电压域（1.2V）消耗功耗 2.44 mW，所需电流 2.0 mA，因此两电压域各只需一对逻辑供电 I/O。考虑 Core Ring 的线宽，以金属 8，9 为 Core Ring 布线层，所需金属宽度分别为（6.26/4.57）=1.37 um 和（2.0/4.57）=0.44 um，所以四周每边金属宽度分别只需（1.37/4=0.34 um）和（0.44/4=0.11 um）,均小于最小宽度，所以这里 core ring 每边采用金属最小宽度 0.42 um，且每边只需一条。 strip 设计的参考数据以高电压域为例，每个标准单元平均电流为（2.44 mA/1866=1.3 uA），金属 1 作为 follow-pin 的最大供给电流为（0.12 x 1.38=0.17 mA），由此推出每条 follow-pin 能够供给 130 个标准单元（170 uA /1.3 uA =130），也就是说大约每行 130 个标准单元需要一对电源/地 strip，同时因为上下两层单元共用一条 follow-pin，所以一条 Strip 每行（row）可供给单元数量要减半为 75 个。

可以看到由于单元数量较少，加之采用低功耗技术使电源 Core Ring 金属使用最小宽度即可，这一情况在今天十万甚至百万门级设计中已不可能出现，但估算的方法是普遍适用的。对于 strip 设计的数量和金属宽度与具体设计息息相关，因此很难给出具体估算方法，然而上述提到的 strip 供给单元数量具有一定参考价值。设计 Core Ring，电压域 Ring，Stripe 和 Follow-Pin 命令如下：

addRing -type core rings –nets {*name_list* } –layer_top *layer* –width_top *width* –spacing_top *spacing*

addRing -type block rings -around power domain –nets {*name_list*} –layer_top *layer* –width_top *width* –spacing_top *spacing*

addStripe -nets {*name_list*} –over_power_domain 1 –layer *layer* –width *width* –spacing *spacing* -set_to_set _distance *real* -start_from {left | right | bottom | top}

sroute –powerDomains {*domain_name*} –nets {name_list} –noBlockPins –noPadPins –noPadRings –noStripes

根据经验设计的供电网络能否保证芯片性能要求，为尽可能在早期发现供电网络存在的电压降（IR-drop），电迁移（EM electromigration）等问题，Encounter 提供了的电源轨道（power rail）分析功能，与 sign-off 电源轨道分析相比，Encounter 电源轨道分析可以在布局后进行，其速度较快，可在早期阶段发现问题，减少和降低了后期修正的难度和可能，但准确度不高，这也是 SVP 方法学的体现。sign-off 电源轨道分析所得数据最为准确，但须在实际布线后进行，并且花费时间较长。电源轨道分析采用的方法与上述经验法相类似，都是从功耗数据入手，因此工具要先进行芯片的功耗计算，得到基于每个实例单元（cell-instance）的功耗值再进行电源网络分析。功耗分析有三种模式：静态模式，统计模式，动态模式，设计者可依据情况选择。静态模式需提供信号翻转技术（TCF Toggle Count Format）文件，统计模式要提供时钟频率和信号活动因子，动态模式分析基于 VCD 文件。功耗数据得到后在进行电源轨道分析前还要向工具提供电源/地的 I/O PAD 位置（Pad Location）数据，不同的电压域要有各自的 I/O 位置数据，电源轨道分析时向工具指定所有电源/地线名称，并输入相应的 I/O 位置数据，所有电压域的电源/地网络就可同时进行分析。分析结果在图形界面反映出的色彩变化可以作为设计者判断、修改电源网络的依据，（图 5- 7）。

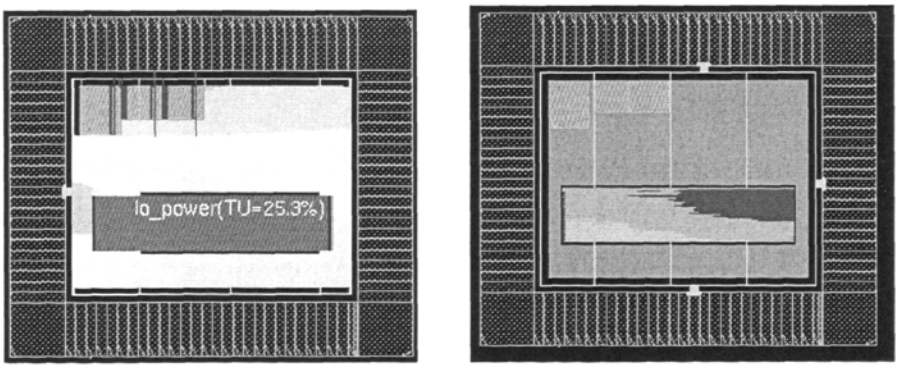


图 5- 7 电源网络分析结果

5.3.5 布局(placement)

布局要做的工作是确定除 floorplan 阶段已摆放模块外其它所有单元的物理位置，主要是标准单元的位置，这一步将基本确定最终版图的结果。工具区分单元是否已经摆放是通过单元在工具中具有的状态决定的，Encounter 工具提供了四种状

态，分别是：UNPLACED, PLACED, FIXED, COVER。标准单元在布局前的状态都为 UNPLACED，布局后自动变为 PLACED，当时钟树综合完成后所有时序单元状态将变为 FIXED。布图阶段手动布置的 I/O，宏模块等状态会自动变为 FIXED，因此工具在自动布局时只会对状态为 UNPLACE 的单元进行操作，而不会影响到布图阶段已摆放的单元。

扫描链（scan chain）是今天大多数可测试性设计使用的方法，它将电路中原本的时序单元转换为可扫描型的时序单元，将它们一一串接起来，形成近似移位寄存器的电路结构，利用这种移位寄存器特性用特定的测试信号，以先后顺序送入电路，再利用移位寄存器将测试结果依次输出。扫描链由前端设计人员插入，后端设计出于对布线资源的考虑，要把逻辑综合阶段插入的扫描链重新连接，这一步称为扫描链重组（scan chain reorder）。扫描链重组在布局后操作，但在布局前，为避免由于扫描链连接关系使工具布局结果损害时序收敛，因此要向工具指明扫描链成员，这样工具在布局时就不会考虑扫描链连接关系。

扫描链信息的指定有两种方式，大多数是利用扫描链的 DEF 文件，如果扫描链个数少、结构比较简单可以使用命令指定扫描链的起始端口和终点端口，之后使用 scanTrace 命令从起始端根据路径连接关系依次找到该扫描链上的单元。

defIn *scanFile*

specifyScanChain *scanChainName* –start *pinName* –stop *pinName*

scanTrace -verbose

自动布局功能只需一条命令即可实现，但命令执行之前还有一些与门控时钟和 MSMV 技术相关的布局参数需要设定。

setPlaceMode –clkGateAware –fixedShifter -timingDriven -ignoreScan

-clkGateAware 使工具在布局时考虑门控时钟的作用，这一功能的实现需时钟树说明文件（clock tree specification）的支持。

-fixedShifter 使布局后的电压转换单元状态变为 FIXED，不允许工具再对其位置进行改变。

-timingDriven 时序驱动的布局，工具布局时将时序收敛作为主要布局原则，可以实现更好的时序收敛。

-ignoreScan 使工具摆放扫描链时忽略扫描链的连接关系。

planDesign 命令实现自动布局功能，布局后利用 checkPlace 命令检查布局结果。

planDesign

checkPlace

单元物理位置的确定将影响电路时序，因此为确保布局后时序收敛，要对时序结果进行检查。-preCTS 参数表明设计处于时钟树综合之前。

timeDesign –preCTS

如果时序检查结果有建立时间违反则要在本阶段予以优化，不要搁置到下一阶段再修正那样会增加时序收敛的难度。优化前设置合适的优化模式参数以达到期望的效果，优化努力有三种情况分别是 lowEffort，mediumEffort 和 highEffort，工具默认为 mediumEffort，当设计的时序收敛难度较大时可以使用 highEffort。需要注意的是，在时钟树综合前使用 optDesign 命令不能对保持时间进行优化，也就是说 –hold 参数此时无效，optDesign 默认对建立时间和设计规则违例（DRV Design Rule Volaiton）进行优化，设计规则违例包括输出端最大电容和输出端信号最大转换时间两项内容，而输出端最大扇出的优化不包含在内，需要通过优化模式设置命令设置参数-fixFanoutLoad 指定。leakagePowerEffort 参数提供了后端进一步进行泄漏功耗优化的功能，low 选项使工具在布线后进行优化，high 选项可在布线前进行优化。optimaizeNetsAcrossDiffVoltPDs 参数使工具对布线后一些穿越额外电压域的互连线进行优化。

setOptMode –optimizeNetsAcrossDiffVoltPDs –lowEffort | -mediumEffort | -highEffort -fixFanoutLoad -leakagePowerEffort {none | low | high}

optDesign -preCTS

布局后各个单元的物理位置已大体确定，接下来重组扫描链，keepHierPorts 仅对 MSMV 设计有效。

scanReorder –keepHierPorts

5.3.1 节介绍物理设计数据输入时提到电压转换单元位置属性文件，（图 5-8），利用该文件工具将所有电压转换单元摆放到电压域的边界处有利于信号互连。

Cell	From PD	To PD	Placed PD	Enable pin (optional)
LEVS_HL	Hi_power	Lo_power	Lo_power	En
LEVS_LH	Lo_power	Hi_power	Hi_power	En

图 5- 8 具有 level shifter 和 isolation cell 功能单元

5.3.6 时钟树综合（CTS）

数字集成电路都在时钟脉冲信号的控制下工作，时钟信号的布线网络称为时钟网络，通常时钟网络连接有大量的时序单元，而时钟驱动单元的驱动能力是有限的，不可能设计一个具有足够驱动能力的单元来驱动所有的时序电路，因此时钟树综合即是将时钟网络分割成较小的部分而建立以多级缓冲电路驱动的时钟网络过程。对同步电路来说时钟树综合的目标是使同一时钟域的时钟信号到达所有时序电路时钟端的时间完全相同或使时钟偏差（skew）保持在允许的范围内。Encounter 提供了根据时序约束文件自动插入缓冲电路生成时钟树的时钟树综合技术，这一技术的实现要基于时钟树说明文件。时钟树说明文件定义了所有时钟域的时钟属性，包括：时钟源端口名，时钟周期，时钟最大/最小延时，最大时钟偏差，时钟端口最大转换时间，缓冲单元名称以及缓冲单元输出最大转换时间等。工具将根据时钟树说明文件规定的指标自动生成时钟树网络。-forceReconvergent 参数避免由于多工作模式而引起时钟树不收敛。

cleanupSpecifyClockTree

specifyClockTree -clkfile *fileName*

ckSynthesis -check -forceReconvergent -report *fileName*

ckSynthesis –forceReconvergent

除基于时钟树说明文件实现时钟树综合外，少数情况下也可以借助 clockDesign 命令实现，clockDesign 命令基于设计约束文件和 SetCTSMode 命令设置的参数综合时钟网络。时钟树综合后利用 reportClockTree 命令对结果进行检查，saveClockNets 将时钟插入的缓冲单元保存为以 ctsntf 为后缀的文件，在布线阶段可作为布线指导文件供布线器使用。

setCTSMode

clockDesign

reportClockTree

report_clock_timing

report_clocks -type

saveClockNets -output *fileName.ctsntf*

时钟树综合完成后时钟延时（clock latency）和时钟偏差（clock skew）被确定下来，而时钟抖动（clock jitter）由于与许多不可预知因素有关，因此抖动是时钟树综合后仍需人为设定的参数，在做进一步的时序分析前需要重新设定约束文件的 clock uncertainty，去除约束文件中设置的时钟偏差。如果时钟树综合结果引起时序违例，则可利用时序优化命令进行优化。此外如果有保持时间的违反，采用的策略可以是在布线后修正或者先修正绝对值大于 0.1 ns 的路径，待布线后再修正所有保持时间违例。

set_clock_uncertainty *value* [*clock_name*]

set_propagated_clock [all_clocks]

timeDesign -postCTS

timeDesign -postCTS -hold

optDesign -postCTS

setOptMode -holdTargetSlack -0. 1

optDesign -postCTS -hold

降低时钟功耗除采用门控时钟技术外，Encounter 还提供了在物理设计阶段进一步降低时钟功耗的克隆与反克隆技术（图 5-9）。克隆技术的目的是增加具有相同控制信号的门控时钟单元个数以减小门控单元的总负载。当门控单元摆放位置距离所控制寄存器距离较远，但却与其它门控单元（具有相同的控制信号）控制的寄存器距离较近时，采用反克隆技术可减少多余的门控单元，重新分配所控制的寄存器资源。克隆与反克隆技术不仅能降低时钟网络的功耗，而且有利于时序收敛和布线资源的优化。

克隆与反克隆时钟门命令如下

ckCloneGate -forceReconvergent

ckDecloneGate -forceReconvergent

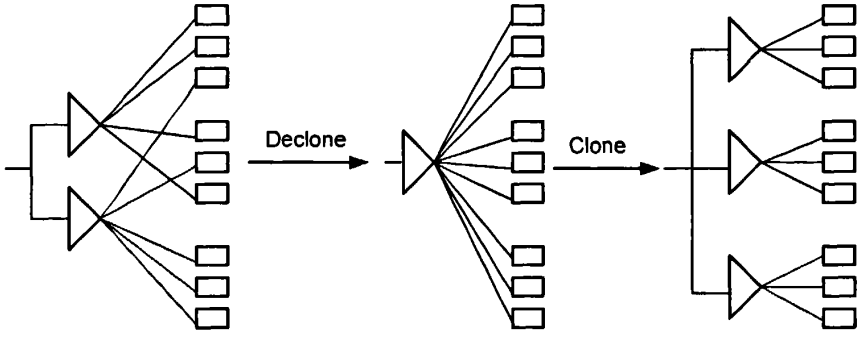


图 5-9 克隆与反克隆技术

5.3.7 试验布线（Trial Route）

布局结果的好坏直接影响到布线质量，加之实际布线过程需要花费大量的硬件资源和时间，因此对较大规模的设计来说设计人员能否在实际布线前预估布线结果就显得非常地必要。试验布线器是一种快速的布线工具，借助它设计人员可以看到当前布局的哪些地方过于拥挤，造成需求的布线轨道数量大大超过了实际所能提供的布线资源。根据这些信息对布局进行调整，减少过于拥挤的区域，不仅提高了实际布线的成功率也节省了大量的时间。

试验布线命令

trialRoute

5.3.8 详细布线（Detail Route）

Encounter NanoRoute 布线器分两步完成互连，全局布线（global route）和详细布线（detail route）。全局布线利用全局布线单元（gcell, global routing cells）将芯片分割成大量正方形区域，每一个区域被称为 gcell。全局布线的目标是将布线资源合理进行分配，减小布线阻塞区域，因此它更像是实际布线前的方案制定评估。详细布线按照全局布线方案将实际金属线铺设在轨道（track）上，在尽量不引起短路，断路和设计规则违反的前提下实现芯片的互连。NanoRoute 布线器具有 SMART（Signal Integrity Manufacturing Awareness Routability and Timing）功能，即在布线过程同时考虑信号完整性，可制造性，可布线性和时序收敛问题。参数 routeWithHonorPD 使布线过程考虑多电压域情况。布线前相关参数设置如下：

```
setNanoRouteMode -routeWithTimingDriven true
setNanoRouteMode -routeWithSiDriven true
setNanoRouteMode -routeSiEffort normal
setNanoRouteMode -drouteFixAntenna true
setNanoRouteMode -routeInsertAntennaDiode true
setNanoRouteMode -routeAntennaCellName ${CELL(ANTENNA)}
setNanoRouteMode -drouteMinimizeViaCount ture
setNanoRouteMode -drouteUseMultiCutVia ture
setNanoRouteMode  -routeWithHonorPD
globalDetailRoute
```

布线后互连线的寄生电阻，寄生电容和耦合电容值被确定下来，此时计算电路的延时会更加准确，Encounter 平台提供了两种延时计算工具，1）FE-DC

2）SignalStorm，FE-DC 是 Encounter 默认延时计算器，常被用在布线前的时序计算，它的特点是速度快但准确度不高。布线后通常采用 SignalStorm 延时计算工具，它将实际寄生参数考虑进来因此准确度高但需花费时间较长。setDelayCalMode 命令设置延时计算工具，ecsmOnDemand 表明 Encounter 仅转换设计中所用到的单元。

setDelayCalMode -signalStorm -ecsmOnDemand

timeDesign -postRoute

timeDesign -postRoute -hold

前面已经提到门控时钟是在逻辑综合阶段由 RC-Complier 工具自动插入的，门控单元可以由设计者选择指定，如不指定工具则自动从单元库中选择具有 clock_gating_integrated_cell 定义又符合属性设置（3.3.1 节）的单元作为门控时钟门。采用具有锁存器结构的门控时钟单元有利于电路时序，如果单个与门作为时钟门电路将会使时序收敛变得复杂，见图 5- 10 为保证时钟信号正确的施加到存储单元 FF，控制信号 En 要很好的与时钟 Clk1 配合满足 Clk1 对建立时间和保持时间的要求。我们都知道对于单周期路径而言，保持时间检查的发起时间（launch clock）和捕获时间 (capture)是在同一个时钟沿，（图 5- 11）。但对于上述电路的情况为了保证 FF 时钟信号的正确性，对控制信号 En 保持时间的检查捕获时间比发起时间推后了半个时钟周期，这就使得满足保持时间变得较为困难，容易产生保持时间违例。

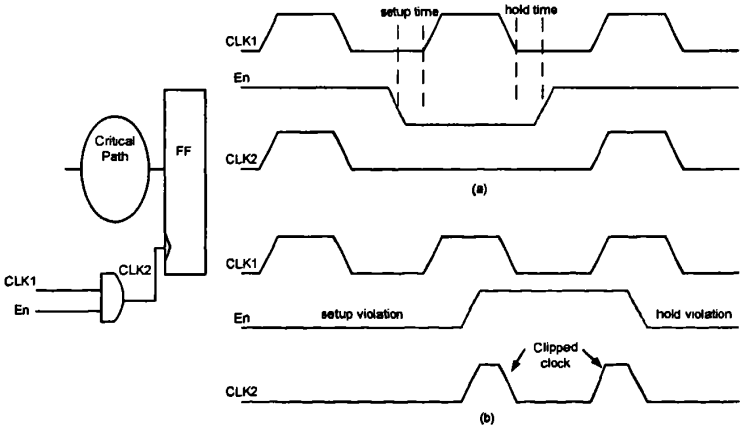


图 5- 10 与门逻辑的门控时钟电路

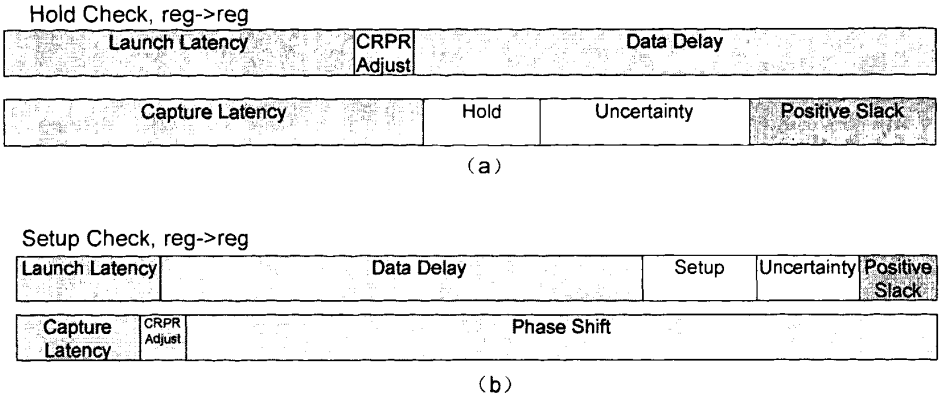


图 5- 11 保持时间与建立时间

布线后很容易造成电路产生许多新的时序违例路径，再次使用 optDesign 命令进行优化。此时参数为 postRoute。正如 CTS 部分所介绍的保持时间违例的修正可以在此阶段进行。

optDesign -postRoute

optDesign -postRoute -hold

optDesign -postRoute -hold -incr

对优化的结果重新进行分析。

timeDesign –postRoute

timeDesign –postRoute -hold

互连线间的信号噪声不仅影响电路功能而且会降低电路性能，特别是互连线间距的逐渐缩小使信号完整性成为设计者越发关注的问题。Encounter 平台集成的 CeltIC 信号噪声分析工具以参数 si 的形式与时序分析命令 timeDesign 结合检查噪声对建立时间和保持时间的影响。如果信号噪声引起时序违例可继续使用优化命令。

timeDesign -postRoute -si

timeDesign -postRoute -si -hold

optDesign -postRoute -si

optDesign -postRoute -si -hold -drv

5.3.9 可制造性设计与签收检查

可制造性设计（DFM Design For Manufacturing）是要缩小芯片设计与制造之间的鸿沟，提升产品的生产良率。掩模阶段的 DFM 技术主要包括离轴照明（OAI，Off-Axis Illumination），光学邻近校正（OPC，Optical Proximity Correction），移相掩模（PSM，Phase Shifted Mask），次分辨率辅助图形（sub-resolution assistance feature）等方法。设计阶段采用的 DFM 技术主要包括放置填充单元（filler cell），添加金属层（metal fill），修正天线效应（antenna effect），填充凹槽（fill notch），物理检查等。

verifyConnectivity -type regular –reportfile *fileName.rpt*

verifyConnectivity -type special –reportfile *fileName.rpt*

verifyGeometry -allowSameCellViols -allowRoutingBlkgPinOverlap

-allowRoutingCellBlkgOverlap -noSameNet -reportfile *fileName.rpt*

verifyProcessAntenna -reportfile *fileName.rpt*

verifyMetalDensity -reportfile *fileName.rpt*

fillNotch –reportfile *fileName.rpt*

foreach layerNum {1 2 3 4 5 6 7 8 9} {

setMetalFill -layer $layerNum -activeSpacing 0.6 -windowSize 100 100 \

-windowStep 50 50 -minDensity 20 -maxDensity 70}

addMetalFill

setFillerMode -noDRC -core {*fillerCellList*}

addFiller -powerDomain *powerDomainName*

Sign-off 检查是芯片物理设计完成后对电路所做的最完整，最精确的检查，这一阶段寄生参数提取工具采用三维模型提取并计算互连线电阻电容值。检查内容包

括全芯片功耗分析、供电网络(power rail)分析、考虑信号噪声和电压降（IR-dorp）效应在内的静态时序分析，因此 sign-off 检查是对芯片功耗，供电网络和时序所做的流片前最精确的分析。SoC Encounter 平台集成了 sign-off 验证所需的全部工具，因此在 Encounter 下就可实现所有验证。

选用 signalStorm 延时计算器。

setDelayCalMode -signalStorm -ecsmOnDemand

timeDesign 命令结合 signoff 参数将使 Encounter 采用三维寄生参数提取工具计算电容电阻值进行时序分析。

timeDesign -signoff

timeDesign -signoff -hold

如果存在时序违例，一般可通过 ECO 手工修正。

refinePlace -preserverRouting

ecoRoute

考虑信号噪声对电路时序的影响。

timeDesign -signoff -si

timeDesign -signoff -si -hold

计算芯片功耗，输出的功耗文件作为 VoltageStorm 的输入文件进行 IR-drop 分析。

updatePower –noRailAnalysis -postCTS -toggleFile *toggleFile* \

-reportInstancePower *powerFile netName*

eval runVStorm -net *netName* -libs *libCell.cl* -powerFile *powerFile* \

-ppFile *padPinFile* -analyzeTC 1 -analyzeRC 1 -analyzeER 1 \

-analyzeVC 1 -analyzeIV 1

将得到的 IR-drop 文件载入工具，考虑 IR-drop 效应对时序的影响。

setIrDropInstVoltage –max -min -early –late -list -infile {*inFiles*}

timeDesign -signoff -si

timeDesign -signoff -si -hold

本节关于 Encounter 平台的 sign-off 检查隐含了很多细节，诸如使用的工具、分析方法、分析流程以及相互关系等，在下一节（5.3.10）将基于 Cadence 的 ETS sign-off 检查工具详细介绍这方面内容。

5.3.10 签收检查（Sign-off）

Encounter Timing System 简称 ETS，是可脱离 Encounter 平台而独立（standalone）运行的签收验证工具，它将时序引擎 CTE，噪声分析引擎 CeltIC 以及延时计算引擎 SignalStorm_NDC 集成在一起，可兼容第三方工具的数据格式。ETS Sign-off 验证流程如图 5- 12 所示。布线后设计以 DEF 或 GDSII 作为信息交互格式交由寄生参数提取工具（ Fire & Ice QX，Cadence QRC）得到 SPEF 或 DSPF 格式表示的基于单元（cell-instance）的互连线寄生电阻电容供功耗分析工具、延迟计算工具和噪声分析工具使用。另一方面，PowerMeter 功耗计算工具得到基于单元的功耗消耗文件和电流二进制数据中间文件供 IR-drop 分析工具 VoltageStorm 使用获得每个单元实际的供电电压值。最后将以上产生的寄生参数文件，电压降文件，Verilog 网表文件以及其它相关文件一并交由 CeltIC_NDC 延时计算器，得到标准延时文件（SDF）和时序违例报告。

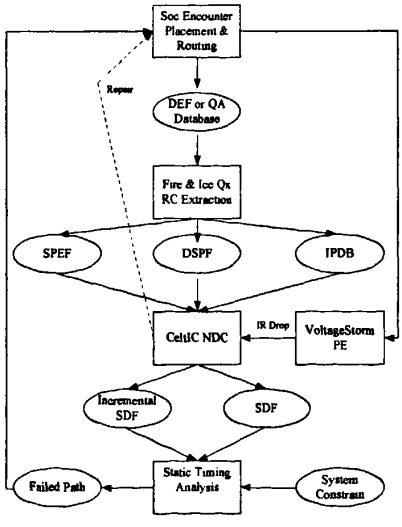


图 5- 12 ETS sign-off 检查流程

5.3.10.1 Fire & Ice QX 寄生参数提取

不论是使用 Cadence Fire & Ice QX 工具进行 sign-off 寄生参数提取，还是 VoltageStorm 工具进行供电网络分析都需要提供以 cl 为后缀名的“电源网格单元库”，该单元库可由 Foundry 提供也可由 Cadence LibGen 工具产生，（图 5- 13）。LEF 和 GDS 库是产生单元库所需的源文件，寄生参数提取工具的技术文件也是 LibGen 的输入文件之一，它提供了各层工艺信息。LibGen 把源文件中的每个单元转化成由电阻，电容和电流源模型组成的拓扑网络。拓扑网络有 Port, Floorplan, Detail, Abstract 和 Reduced 五种情况，其中 LEF 库可产生 Port 和 Floorplan 两种拓扑网络，而另外三种必需借助 GDSII 库产生。当寄生参数提取工具的技术文件中各工艺层名称与 LEF 库或 GDSII 库名称不一致时可使用层映射（layer mapping）文件建立对应关系。LibGen 产生的单元库以文件夹形式存在，包含的文件除技术文件外均以二进制形式出现，设计者使用时不能对文件夹内容做任何改动[16~17]。

Port View：不包含单元内部的电源网络及互连线寄生信息，仅由单个电流源负载挂接在供电网络端口上。Port View 适用于结构简单和 IR-drop 效应不明显的模块，如标准单元电路。

Detail View::：Detail View 包含从 GDSII 数据中抽取的单元内部电源网络寄生电阻电容信息，分立电流源负载以及 5.3.10.2 节将要介绍的运用 PM 静态概率与跳变密度传输法分析静态功耗所需的门级网表。Detail View 是 LibGen 产生的包含最详细信息的单元库数据，适用于 IP 和定制模块电路，但产生过程会花费较大的硬件资源和时间。

Port View 数据量小、产生速度快，但准确度不高，Detail View 恰恰相反，数据量大、产生时间较长，但准确度高，因此为了提高效率，LibGen 还提供了另外几种单元库模型，它们数据模型的优缺点介于 Port View 和 Detail View 之间。

Abstract View：Abstract View 是 Detail View 的简化模型，它将电流源负载用电导模型代替，减少了电源网络寄生数据量，该模型可有效用于顶层分析，但无法分析模块内部的 IR-drop，因此可以采用 Detail View 分析低层模块，待顶层分析时改用 Abstract View 的方法。

Reduced View：Reduced View 包含模块内电流源负载的详细模型，并含有缩减的等效电阻网络寄生信息，因此该模型可用于有限的顶层 IR-drop 分析。

Floorplan View：根据 LEF 库文件指定电阻和电流源负载信息。

Encounter 平台下调用 LibGen 工具命令：

runLibGen –cmd *filename* -inputType {LEF | GDS | LEF&GDS} -techfile *filename.tch* -libraryName *lib_name* -layermapping *filename.map* -lefFileList *filename.lef* -gdsFileList *filename.gds*

独立调用 LibGen

libgen -cmd *command_file*

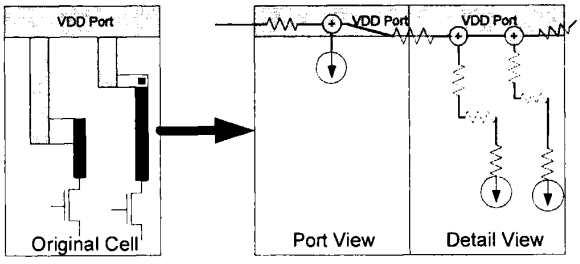


图 5- 13 Power-grid cell library

在 Encounter 平台下可直接调用 Fire & Ice QX 寄生参数提取工具，也可独立调用。表 5- 2 列出的是 Fire & Ice QX 的输入输出文件。输出包括互连线的标准寄生交换格式（SPEF Standard Parasitic Exchange Format）文件和详细标准寄生参数格式（DSPF Detail Standard Parasitic Format）文件。在后期的功耗分析和延时计算都会用它们。

输入文件	设计文件 (DEF)	电源网格单元库（.cl）	技术文件（techfile）	映射文件（layer mapping）
输出文件	SPEF		DSPF	

表 5- 2 Fire & Ice Qx 输入输出文件

Encounter 平台下调用 Fire & Ice QX 工具命令：

runQX -cmd *fileName* -techFile *fileName.tch* -libraryName *lib_name.cl* / -outputFileName *fileName.spef*

独立调用 Fire & Ice QX

qx -cmd *commandFile DEFfile*

5.3.10.2 功耗计算

我们用功耗计算工具 PowerMeter 计算得到每个实例单元的内部功耗，泄漏功耗，跳变功耗和总功耗，以 ASCII 格式记录上述信息，默认文件名 powermeter.pwr，该文件作为 VoltageStorm 的重要输入文件用于 IR-drop 的分析。PM 有静态模式分析和动态模式分析两种方式，需要注意的是这里的静态模式分析并不是指对泄漏功耗的分析，同样动态模式分析也不是专门针对跳变功耗和短路功耗的分析。静态 PM 分析可理解为电路的平均功耗计算，动态 PM 分析为实时分析，它利用 VCD 或 TWF 文件得到电路功耗和的二进制电流数据文件供动态 IR-drop 分析使用。静态 PM 分析流程见（图 5- 14），表 5- 3 是 PowerMeter 所需输入输出文件[18~19]。

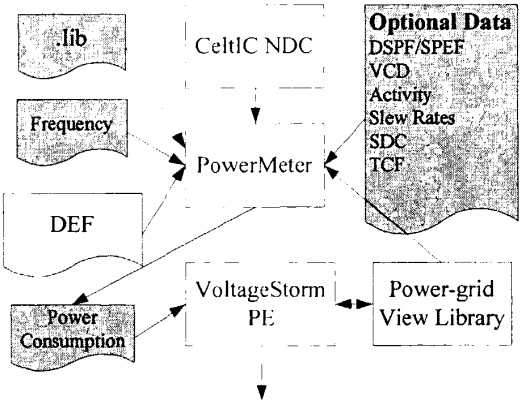


图 5- 14 PowerMeter 流程

输入文件	Liberty 时序库 (内部功耗 泄漏功耗)	SPEF 或 DSPF （互连线 R,C）	门级 VCD (static & dynamic)	RTL 级 VCD (static)
	libGen 单元库 （替代 lib 中不含有的单元）	SDC (时钟信息)	TWF (dynamic)	TCF (static)
输出文件	单元功耗文件 powermeter.pwr （static IR-drop analysis）		电源网络二进制电流数据 (dynamic IR-drop analysis) 平均/最大/均方根，ptiavg/max/irm	

表 5- 3 PowerMeter 输入输出文件

静态模式 PM 功耗计算：

1）向量驱动法：向量驱动法采用门级网表仿真生成的 VCD 文件作为信号跳变依据计算功耗，虽然 VCD 文件包含所有线网的完整跳变信息但静态模式的功耗计算并不会使用所有内容，而只考虑线网的翻转率（toggle rate），另一方面，找到能够覆盖门级电路主要功能和消耗最大功耗的测试向量非常困难且文件较大，因此对静态模式的功耗计算来说可以使用翻转计数格式 TCF（Toggle Count Format）文件[20]。

2）静态概率与跳变密度传输法：静态概率（static probability ）,也称 duty cycle，是指信号处于“1”的概率，它是信号处于“1”的时间与总时间的比值，(图 5- 15) 用 P(X)表示。跳变密度（transition density）指 1 秒钟内信号上下翻转的次数，用 D(X)表示。因此当知道单元输入信号的跳变密度和静态概率通过公式 5-1 就可得到输出端的跳变密度。确保该方法所得功耗准确度的最基本条件是能提供电路原始输入端，诸如使能信号，复位信号以及时钟信号的静态概率与跳变密度。

图 5- 15 Duty Cycle

$$D(y)=\sum_{i=1}^{n}P\left(\frac{dy}{dx_i}\right)D(x_i) \qquad (5\text{-}1)$$

3）混合法：混合法是普遍使用的一种方法，它是以上两种方法的组合。RTL 级仿真得到的 VCD 文件作为电路原始输入端跳变信息，这样一方面保证了输入端信号跳变的准确性，另一方面 VCD 文件也不会过大。而剩余线网的活动性则通过静态概率与跳变密度传输法计算得到。

动态模式 PM 电流数据计算：

1）向量驱动法：该方法与静态模式功耗计算使用的是同一个 VCD 文件，但工具关注的内容不同，动态模式分析更关注单元信号发生跳变的时刻和跳变值。PM 动态仿真时间一般为几个时钟周期，每个周期被分割为更小的时间间隔，130nm 工艺约为 50ps, 90nm 工艺约为 20~30ps，PM 工具计算每个时间间隔点电源网络的电流数据为动态 IR-drop 分析提供数据支持。

2）非向量法：非向量法也称时序窗法，即功耗与电源网络电流数据的计算基于时序窗文件 TWF（timing window format），与向量驱动法相比，使用 TWF 得到的电源网络电流准确度要低于 VCD 文件的使用，但运行速度要更快。

相比以上方法，基于 VCD 文件得到的功耗和电流数据信息是最为准确的。

5.3.10.3 电压降分析

供电网络完整性分析工具 VoltageStorm PE 可用于包括静态分析和动态分析，静态分析用于检查由于电源网络设计缺陷，如线宽不足，通孔缺失，供电网络开路，短路等造成的电压降（IR-drop），电迁移，地电压升高等问题，对 130_nm 以下工艺来说，除利用静态供电网络分析保证电源网络鲁邦性（Robust）外，动态分析也十分必要，动态分析能够使设计者知道电路中哪些地方需要更多的去耦电容来弥补瞬态电流过大造成的 IR-drop；哪些地方因存在多余去耦电容而占用了额外布局资源，消耗了不必要的泄漏功耗。VoltageStorm PE 输入输出文件如表 5- 4 所示。静态供电网络分析需要提供物理设计文件，LibGen 产生的单元库电源网络模型，PM 工具得到的基于单元的功耗消耗文件以及 I/O Pad 物理位置文件，输出是基于每个单元的实际供电电压文件。动态供电网络分析除用供电网络电流数据代替基于单元功耗文件外，其它输入文件完全相同，但输出文件包括去耦电容数量评估报告和展示动态 IR-drop 效应的电影文件。

输入文件	DEF/GDS/OA 设计文件	电源网格单元库	功耗文件（static）	供电网络电流数据文件 (dynamic)	供电 I/O 物理位置信息
输出文件	基于单元的 IR-drop 报告	去耦电容缺失报告（dynamic）	去耦电容密度报告（dynamic）	动态电影文件	

表 5- 4 VoltageStorm 输入输出文件

VoltageStorm PE 按上述表格所列内容导入静态分析所需输入文件后就可以进行 IR-drop 分析。命令如下，注意：打印（print）单元供电电压文件前必须先执行 filter iv 或 analyze iv 命令

instance_power_file [avg | peak] *supply_rang filename*

analyze *net_name nominal_voltage* avg *ir_limit* <analysis_type+>

filter iv auto

print analysis_type *fileName*

其中 analyze 命令指定的分析类型（analysis type）参数包括 ir, rj, er, rc, tc, iv, vc, vv, pi, 等内容，它们的具体含义分别是：

ir：电压降分析

rj：电流密度分析

er：电迁移分析

rc：电阻电流分析

tc：电流源电流值分析

iv：基于单元的供电电压分析

vc：电压源电流分析

vv：通孔电压分析

与静态分析相比，动态分析使用电流数据文件代替功耗文件。5.3.10.2 节中讲到，PM 产生的电源网络的动态电流文件是按照 50 ps（130 nm 工艺）或 20~30 ps（90 nm 工艺）时间间隔仿真得到的，每个单元瞬态下从电源网络获取的电流值可通过以下方法进行计算，假设某单元从 libGen 单元库的 Detail View 中计算得到 两电流源负载电流分别是 4 mA 和 6 mA,，因此总的静态电流为 10 mA，见图 5- 16-b。PM 计算得到的瞬态电源网络电流波形见图 5- 16-a，实线是峰值电流值（.ptimax），虚线是平均电流值（.ptiavg），以峰值电流为例，在第一个时间间隔

采样点，供电网络电流为 0.35 mA, 该值除以单元静态总电流 10 mA 得到比例因子 0.035，从而计算出单元每个电流源负载在第一个时间间隔点的电流分别为 4 mA x 0.035=0.14 mA 和 6 mA x 0.035=0.21 mA。依次类推，从而得到仿真时间内每个时间间隔点各单元的瞬态电流值。

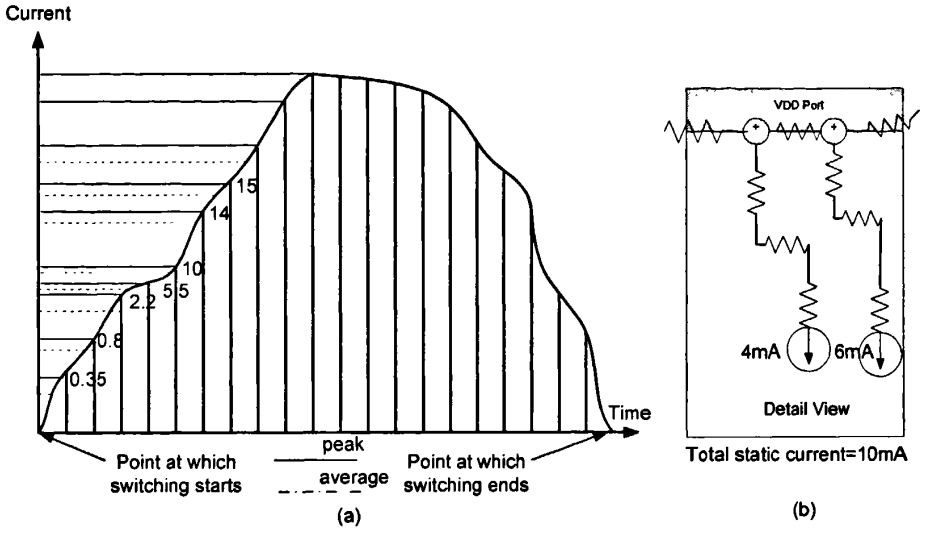


图 5- 16 PM 瞬态供电网络电流波形

动态 IR-drop 分析命令如下所示：

current_data_file *filename*

analyze *net_name nominal_voltage* dynamic *ir_limit* <worst_case_limit=value> <analysis_type+>

记录瞬态节点电压值，输出分析结果的电影文件

setvar save_transient_states true

5.3.10.4 有效电流源模型（ECSM）

供电电压的波动会影响电路性能，这种情况下如何准确地特征化单元和IP的时序信息对确保电路质量有着重要的影响。有效电流源模型与非线性延时模型（NLMD）相比将电压波动因素考虑在内能够更加精确地特征化单元的时序信息。liberty库定义的输入电容是一个固定值，信号的翻转时间和单元延迟时间是输入信号翻转时间和输出负载的函数。与NLMD库相比ECSM具有如下特征：1）输入电容值随信号频率变化而变化，2）根据输入信号翻转时间和输出负载电容在输出信号翻转时间（fall/rise_transition）组（group）内构建输出信号翻转的电压-时间（V-T）曲线，而不是.lib中某一固定数值，延时计算器调用V-T曲线，应用公式 $C_{eff} = \frac{Q}{V} = \frac{\int_0^t Idt}{V}$ 将V-T曲线转化为电流-电压曲线（I-V），利用电流电压曲线计算输出信号的翻转时间和单元延迟时间，(图5-17)。ECSM在库文件中选择特征化输出电压波形而不是电流波形是因为对ECSM产生工具来说电压波形的特征化速度更快，电压值更容易测量。图5-18是ECSM的.lib扩展定义描述[21]。

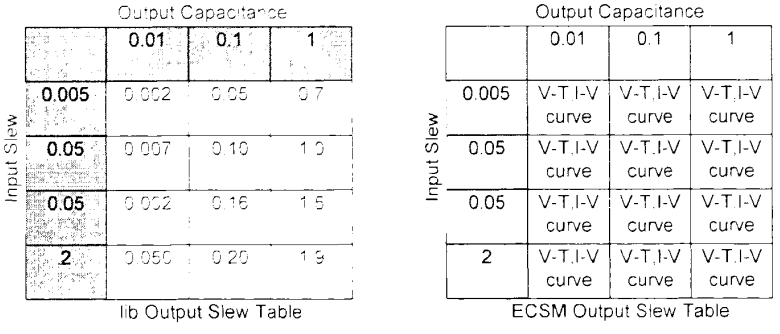
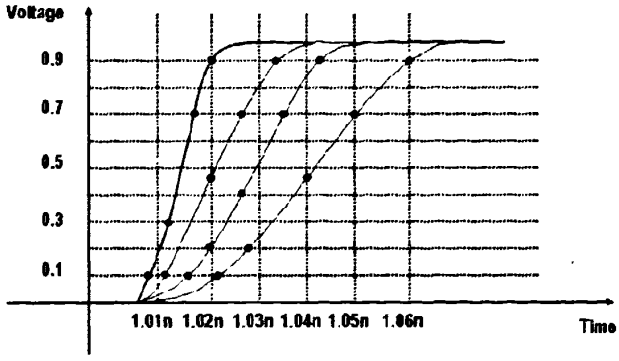


图5-17 ECSM与.lib时序特征化比较




```
rise_transition (temp_1x4) {
   index_1 : "0.1n" ;
   index_2 : "0.1p  0.2p  0.3p  0.4p" ;
   values   ("0.01n,  0.02n,  0.026n.  0.45n" ;
   ecsm_waveform ( "0" ) {
      index_1 : "0.1,  .3  .7  .9" ;
      values : "1.005n,  1.012n,  1.018n,  1.02n " ; }
   ecsm_waveform ( "1" ) {
      index_1 : "0.1,  .48  .7  .9" ;
      values : "1.011n,  1.02n,  1.027n,  1.032n " ; }
   ecsm_waveform ( "2" ) {
      index_1 : "0.1,  .2  .4  .7  .9" ;
      values : "1.015n,  1.02n,  1.028n,  1.035n,  1.07n " ; }
   ecsm_waveform ( "3" ) {
      index_1 : "0.1,  .2  .45  .7  .9" ;
      values : "1.021n,  1.029n,  1.04n,  1.06n,  1.07n " ; }
```

图 5- 18 电压时间曲线与 liberty 的 ECSM 扩展

5.3.10.5 信号完整性分析与延时计算

特征尺寸的不断缩小使得互连线寄生参数模型更加地复杂，物理设计工具在布线阶段为满足时序收敛的要求，会使得部分区域的连线密度过高，线间距过于紧密，因此互连线间距的缩小与互连线密度过大都使得噪声对电路的正常功能和性能产生着潜在的影响。同样，电压降导致的芯片性能下降也不能忽视，特别是 90 nm 后，电路工作电压已降到 1 V 左右，因此即使是微小的电压起伏也会给芯片性能带来深刻地影响。基于以上两方面因素，CeletIC_NDC 工具整合延时计算工具 SignalStorm 和信号完整性分析工具 CeltIC，实现了噪声分析以及噪声和电压降共同作用对芯片时序收敛的分析[24~25]。CeltIC_NDC 的输入输出文件见表 5- 5。

输入文件	设计文件（Verilog）	噪声库（.cdB）	时序库（.lib）	设计约束文件（.sdc）
	寄生参数文件（SPEF,DSPF）	电压降文件（.txt）	时序窗文件（.twf）	电压域创建文件（.msmv）
输出文件	噪声报告	延时报告	时序报告	ECO 工程更改计划文件
	标准延时格式文件（.sdf）	噪声抽象文件 ECHO,XILM	翻转率变化文件	

表 5- 5 CeltIC_NDC 输入输出文件

CeltIC_NDC 具有三种分析模式：噪声分析（noise analysis），噪声延时分析（noise-on-delay analysis），电压降延时计算（IR-drop aware delay calculation）。这三种模式具有公共的输入数据，也有各自所需数据的数据读入命令和分析命令。

输入数据命令如下：其中-ir_drop 选项为电压降分析所需数据，-msmv 选项为多供电电压技术所需数据，即创建和划分电压域的初始文件。噪声分析需提供时序窗文件。

set_run_mode -process {180 nm |130 nm | 90 nm}

set_parm file_name_prefix *name*

read_dotlib liberty_file_list

load_netlist -verilog top.v -cdb cdb_file -spef spef_file -top top_name -vdd {vdd_net_name} -gnd {gnd_net_name} -ir_drop {ir_drop_file gnd_bounce_file} -msmv msmv_file

load_timing *twf_file* （noise analysis only）

process_netlist

除噪声分析外需读入设计约束文件

read_dc_script *sdc_file*

set_tw_convergence -iteration *number*

噪声分析命令：analysis_nosie

噪声延时分析命令：analysis_noise -delay

电压降延时计算：calculate_delay

analysis_noise -delay

分析结果若存在信号完整性问题或时序违例可以利用 CeltIC_NDC 产生 ECO 文件，再反标回布局布线工具进行修正。根据不同的物理设计工具，可产生不同格式的 ECO 文件，例如 se=Silicon Ensemble fe=First Encounter generic=Apollo & Magma 。ECO 解决信号完整性问题通常采用的方法有：1）插入缓冲器，2）增大或减小驱动单元，3）拉宽互连线间距，4）插入屏蔽互连线。产生 ECO 文件的命令如下：

report_eco -noise {buffins | resize | spacing | shieldnet | nofix}

-delay {resize | spacing | nofix }

-clock_nets {fix | nofix | only}

-format {se | fe | pks | nanoroute | generic}

-peak_thresh *value*

-fix_all_prop_noise

CeltIC_NCD 产生输出报告和标准延时文件命令如下：

generate_report -sort_by noise

generate_report -delay max

generate_report -delay min

generate_sdf *sdf_incfile.sdf*

report_timing -late

report_timing - early

write_sdf *sdf_filename.sdf*

第六章 DTMF 低功耗实现结果与比较

针对 SMIC 90 nm 单元库提供的低功耗技术文档，本研究采用了四种实现方案，它们分别是：1）多供电电压多电压技术与门控时钟，结合多阈值单元技术以期最大程度地降低总功耗，2）多供电电压多电压技术与多阈值单元技术结合研究门控时钟技术对动态功耗的影响，3）多供电电压多电压技术与门控时钟技术结合研究多阈值单元对泄漏功耗的影响，4）门控时钟技术与多阈值单元技术结合研究多供电电压多电压技术对动态功耗的影响。同时，在以上四种实现技术中，还都包含了操作数隔离技术，见表 6- 1。

四种低功耗技术实现方案解释		
MSV_CLK_Mth	1.2 V + 1.0 V	多供电电压+门控时钟+多阈值单元组合+操作数隔离
MSV_Mth	1.2 V + 1.0 V	多供电电压+多阈值单元组合+操作数隔离
MSV_CLK	1.2 V + 1.0 V	多供电电压+门控时钟+操作数隔离
CLK_Mth	1.2 V	门控时钟+多阈值单元组合+操作数隔离

表 6- 1 四种低功耗技术实现方案解释

6.1 逻辑综合结果的功耗数据分析

逻辑综合阶段使用的互连线寄生参数是从时序库的线负载模型或根据物理版图预估法计算得到，因此电路功耗是估算值。在逻辑综合阶段实现上述四种方案需要对脚本做一些小的改动，如打开（true）或关闭（false）门控时钟选项，多阈值单元优化选项，创建或删除电压域等。表 6- 2 是从 RTL Complier 工具得到的四种实现方案的功耗统计值。

Power Statistic With Different Logic Synthesis Techniques

Technique	Total Cells/Low	Leakage Power(w)	Internal Power(w)	Net Power(w)	Switching Power(w)
MSV_CLK_Mth	6746/4880	8.22E-05	4.42E-03	1.37E-03	5.80E-03
MSV_Mth	6821	8.37E-05	5.07E-03	1.78E-03	6.85E-03
MSV_CLK	6577	9.24E-05	4.43E-03	1.37E-03	5.80E-03
CLK_Mth	6695	6.38E-05	4.48E-03	1.41E-03	5.90E-03

表 6- 2 逻辑综合结果的功耗数据统计

对表 6-2 数据采用柱状图方式予以表示（见图 6-1，图 6-2），可以得出：

1）采用了 MSMV 和门控时钟技术的电路所消耗的动态功耗值几乎完全相同，多阈值单元对动态功耗的影响较小，见图 6-1 中柱 1(MSV_CLK_Mth)和柱 3(MSV_CLK)。

2）从（图 6-1 柱 1、2、3）中可以看出门控时钟对减小电路动态功耗起到比较明显的效果。尽管柱 4 使用的是单一供电电压 1.2 V，但由于采用了门控时钟使其与 MSMV 技术相比所消耗的动态功耗差距并不明显。

3）泄漏功耗方面，从图 6-2 柱 3(MSV_CLK) 中可以很清楚地看到多阈值单元的组合对减小泄漏功耗有一定的帮助。

从上述数据的分析结果可以看出门控时钟技术对减小动态功耗起到了比较明显的作用。在其它技术条件相同的前提下采用单一供电电压的电路会比 MSMV 技术消耗更多地动态功耗，虽然从数据反映出来的仅是 0.1 mW 的差别，但同时应该注意到本设计的规模非常小，总单元数不超过 7000 门，对于今天的设计而言电路的规模通常都在十万、百万门级，因此可以看出不同的低功耗技术会显著地影响电路总的功耗。

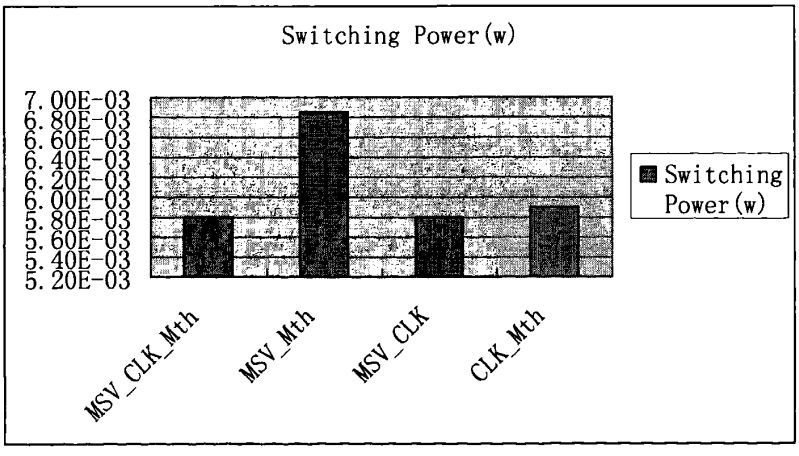


图 6-1 逻辑综合结果的动态功耗柱状图

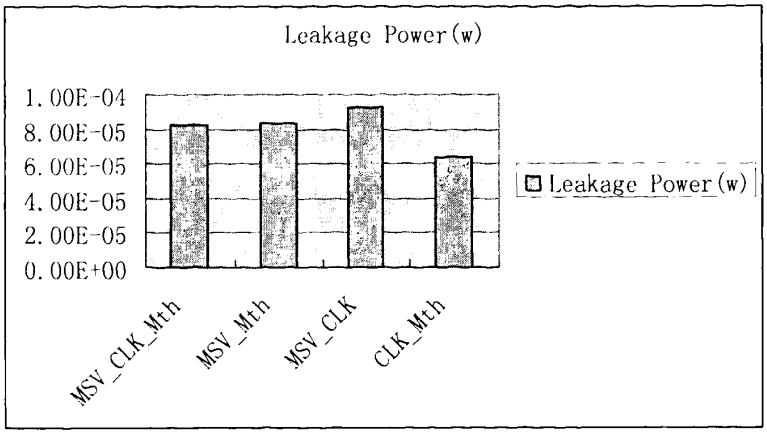


图 6-2 逻辑综合结果的泄漏功耗柱状图

6.2 物理综合结果的功耗数据分析

布局布线后互连线的实际走线和长度被确定下来，互连线的寄生参数通过 RC 提取工具计算得到，因此此时所到的功耗数据是较为真实的。为了避免因布图设计的不同而影响电路功耗，上述四种情况采用了基本相同的设计方案。从数据统计结果来看，时钟树综合和时序路径的优化使单元数量比逻辑综合的结果有所增加。与 6.1 节中所得结果类似（见表 6-3），

Power Statistic With Different Physical Synthesis Implementation

Technique	Cells	Leakage Power(w)	Internal Power(w)	Switching Power(w)
MSV_CLK_Mth	7700/5363	1.03E-04	4.03E-03	5.90E-03
MSV_Mth	7415/5105	9.60E-05	3.74E-03	5.60E-03
MSV_CLK	7386/5386	1.10E-04	4.14E-03	5.96E-03
CLK_Mth	7120	7.96E-05	5.84E-03	7.25E-03

表 6-3 物理综合结果的功耗数据统计

同样应用柱状图表示表 6-3 数据可以得出：结合 MSMV 和门控时钟技术（见图 6-3 柱 1，3）的电路消耗的动态功耗近似相等，而多阈值单元的组合（图 6-4 柱 1）能够近一步减小电路的泄漏功耗。此外，由于互连线寄生参数的影响，MSMV 技术在后端设计中较为明显地表现出了其在减小动态功耗方面的优势，仅对 7000 门的电路来说功耗就相差 1 mW 之多。同时，值得注意的是：从采用 MSMV 技术的三个柱状图（图 6-3 柱 1，2，3）来看，没有使用门控时钟的电路反而比其它两个使

用了该技术的电路消耗了更少的动态功耗，这与在逻辑综合阶段门控时钟表现出的结果恰恰相反。就这一问题回顾 2.1.1 节可知门控时钟会增大逻辑综合结果与时钟树综合的误差，增加时钟树的偏差和时钟网络延时使时序收敛的难度增大，因此这就潜在的需要加大电路尺寸，增加驱动单元的数量来满足时序要求，另一方面，由于电路规模小，寄存器数量少，门控时钟在抑制寄存器动态功耗方面起到的作用有限，基于这两个方面可以较为合理地解释上述问题。另外，从物理设计得到的泄漏功耗数据可以看出它与逻辑综合结果基本一致。

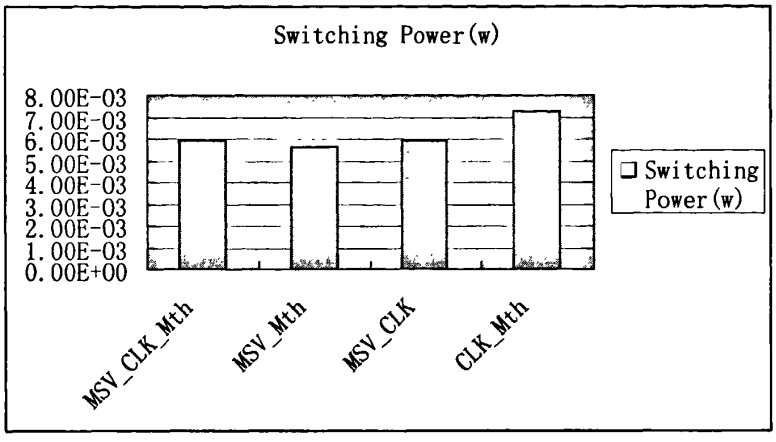


图 6-3 物理综合结果的动态功耗柱状图

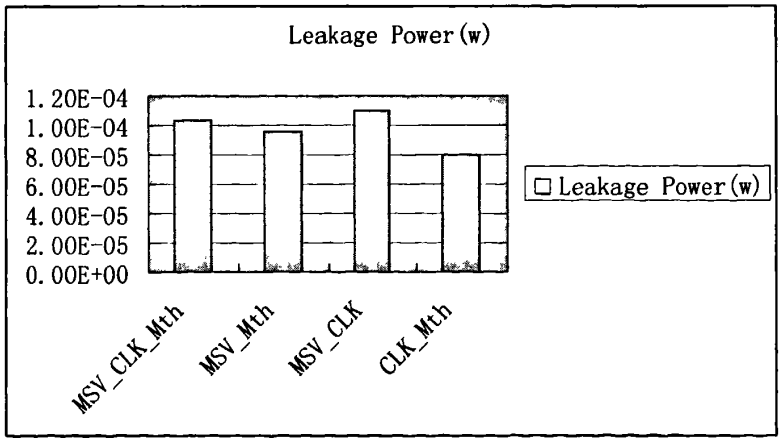


图 6-4 物理综合结果的泄漏功耗柱状图

综上所述，MSMV 技术能够很好地减小电路动态功耗，不会增加验证的难度，但对于逻辑综合和后端物理设计来说会使实施的难度增大。门控时钟和操作数隔离技术对于 RTL-Complier 逻辑综合工具来说实现起来较为容易，而且它们已经成为今天设计中所必然采用的一种技术手段，但门控时钟也会造成时序上的诸多问题，特别是在后端设计中由时钟门引起的时序违例问题变得更加地突出，问题也更加的复杂。操作数隔离技术与 RTL 代码的编写风格有关，因此降低数据通路模块产生的动态功耗要善于利用使能信号控制无谓的操作数跳变。多阈值单元优化是降低电路泄漏功耗较为容易实现的技术手段，所以只要单元库支持都应当作为基本的工程技术予以实现。

6.3 讨论与展望

集成电路工艺水平的不断进步使得芯片的性能越来越高，规模不断增大，在深亚微米阶段设计者通常都是在解决了芯片性能和面积后才会去考虑功耗问题。进入 90 nm 工艺节点后泄漏功耗进一步增加，由于功耗增大带来一系列诸如芯片成本增加，性能下降，可靠性降低等问题使得设计人员需要重新权衡功耗、性能和面积的优化关系，这一变化甚至已经影响到传统的芯片设计方法。功耗问题的解决是一个复杂的过程，从芯片系统级规划到最终版图实现，从标准单元库到电子设计自动化工具，这期中的每一环节都需要 IC 产业链的共同合作。特别是近几年来电子设计自动化工具的进步和低功耗标准单元库的支持使许多低功耗技术得以实现，大大提高了低功耗设计的质量和水平。

本文基于 SMIC 90 nm 低泄漏逻辑标准单元库，采用 Encounter 平台的低功耗设计方法并利用 DTMF 接收器实现了目前普遍使用的和较为先进的几种低功耗技术的逻辑综合和物理实施过程，从实际工程角度出发比较了它们在降低电路功耗方面所起到的实际效果，各自潜在的问题和具体实现的难易等，为今后设计提供工程参考。

当然本文还有很多尚可进一步完善的地方可在今后继续深入研究讨论，1）采用先进的低功耗约束文件通用功率格式（CPF Common Power Format）实现上述研究，缩小 RTL-GDSII 低功耗设计流程的鸿沟，使不同 EDA 工具采用相同的功耗约束，实现对验证、综合、测试、物理综合、布线到签收的几乎所有数字设计工具的

支持。2）SMIC 90 nm 低功耗泄漏单元库与 Encounter 平台可支持电压关断技术，利用这一先进的技术实现 DTMF 接收器设计，提供设计参考并与其它技术做更深入的结合与比较。

参考文献

[1] 张永新、芦生礼、茆邦琴，门控时钟的低功耗设计技术。微电子学与计算机，2004，21（1）：23~26

[2] 钟伟军、刘明业，软实时系统下动态电压/频率调节算法设计.北京理工大学学报，2005 年 10 月，第 25 卷.第 10 期 868.

[3] 卢春鹏，动态电压与频率调节在降低功耗中的作用 Microcontrollers & Embedded Systems.2007 年第 5 期：12

[4] 王明甲、傅兴华、张万里、陈茜，DTMF 编解码芯片的研究。微电子学与计算机，2006，23（3）：189~197

[5] Low Power in Encounter™ RTL Compiler, Product Version 6.2

[6] 陈静华，SOC 芯片低功耗设计。湖南大学硕士学位论文，2005 年 3 月

[7] Library Guide for Encounter™ RTL Compiler, Product Version 6.2

[8] SMIC-Cadence Reference Flow 3.0 User Guide Version0.1

[9] Encounter™ RTL Compiler Synthesis Flows, Product Version 6.2

[10] Command Reference for Encounter™ RTL Compiler, Product Version 6.2

[11] Setting Constraints and Performing Timing Analysis Using Encounter™ RTL Compiler, Product Version 6.2

[12] Encounter™ User Guide, Product Version 5.2.4

[13] Encounter™ Text Command Reference, Product Version 5.2.4

[14] Encounter™ Menu Reference, Product Version 5.2.4

[15] Encounter™ Timing Closure Guide, Product Version 5.2.1

[16] Fire & Ice QXC User Guide, Product Version 3.4 .1

[17] Fire & Ice QXC Command Reference Manual

[18] VoltageStorm Cell-Level Rail Analysis User Guide, July 2006

[19] VoltageStorm PowerMeter Manual, July 2006

[20] Toggle Count Format Reference, Product Version 1.0

[21] Si2 Effective Current Source Model (ECSM) Specification™ Version 2.0 14th Dec 2005

[22] Encounter Timing System User Guide, Product Version 6.1

[23] Encounter Timing System Text Command Reference, Product Version 6.1

[24] CeltIC NDC User Guide, Product Version 6.1.3

[25] CeltIC NDC Command Reference, Product Version 6.1.3

致谢

本论文能够顺利完成首先要衷心感谢我的指导教授于敦山博士和企业导师陈春章博士，两位老师不仅知识渊博，在学业上对我悉心指导、严格要求，亦在为人处事，治学作风上给予我深深的教诲让我受益良多，受用终身。

我还要再次感谢陈春章老师和 Cadence 公司在过去一年多来为我提供的宝贵的实习机会和良好，宽松的学习、工作环境。正是在他们的大力帮助下才使得论文项目能够如期顺利地完成。同时，衷心感谢给予我无私帮助和指导的工程师们（陈家福、蒋宁昱、冯江、李涛、刘康、杨晓敏、王镇山、于洪杰、马泉、李茉等），在与他们的项目交流中不仅学到了许多实际工程技术，更重要的是从他们身上看到了一位优秀集成电路工程师所应具有的素质。借此还要一并感谢我的同学们 (杨旭光、董章钊)，在上海的工作、学习期间我们一同协作得以共同进步。

最后我要特别感谢我的父母，感谢他们多年来对我的关怀和哺育，是他们的全力支持使我能心无顾虑地专心于学业研究。谨此将这份“不合格的答卷”献给他们。