

基于全定制 IP 设计的漏电流功耗仿真计算方法

张 锋 周玉梅 黄令仪

(中国科学院微电子研究所, 北京 100029)

摘要: 提出了一种自顶向下的基于晶体管级的全定制 IP 漏电流功耗计算方法, 该方法计算快速高效, 实用性强, 取代了以往完全依靠软件仿真进行功耗计算的技术。在设计龙芯 II 号 CPU 中的全定制 IP 时应用了此方法, 该芯片采用的是中芯国际 0.18 μm CMOS 工艺技术。为了验证该方法, 把计算结果与 Synopsys 公司的 Nanosim 仿真结果进行对比, 误差只有 10% 左右。由于软件仿真需要大量的测试激励与计算时间, 而该方法不需要外加测试激励便可以计算出全定制 IP 漏电流功耗, 并能快速找到其模块所在位置, 使设计周期大为缩短, 因此完全可以针对这种计算方法开发相应软件及进行应用。

关键词: 漏电流功耗; 堆积因子; 有效宽度; 高阻态

EEACC: 2570A; 1280; 1265D

中图分类号: TN 402

文献标识码: A

文章编号: 0253-4177(2004)09-1169-06

1 引言

随着集成电路制造技术的不断进步, 芯片特征尺寸不断减小, 功耗问题日益严重, 尤其是原来一直不被重视的亚阈值电流所带来的静态功耗, 即漏电流功耗已成为影响整个芯片功耗的重要因素^[1-3]。当尺寸达到 70nm 时, 漏电流功耗会占到总功耗的 20% 左右^[4], 所以漏电流功耗的估计对于大规模电路设计已经必不可少。文献[4]总结了基于系统结构级、门级、晶体管级的三种漏电流功耗计算方法。对于前两类的计算, EDA 软件公司已推出了比较成熟的软件产品, 例如 Synopsys 公司的 power compiler, Cadence 公司的 power analysis 等, 而对于晶体管级的仿真测试软件通常采用的是 Hspice^[5]。它可以精确地仿真出漏电流功耗, 但缺点是仿真速度慢, 只适合用于小型电路。目前有几家 EDA 公司相继推出了基于大规模电路晶体管级的 Snice 仿真软件, 其中有 Hsim(Nassda 公司), Ultrasim(Cadence 公司), Nanosim(Synopsys 公司)等, 这些软件比

Hspice 的仿真速度都要快上百倍, 也都可以比较准确地计算出电路的漏电流功耗, 但都有一定的缺陷。首先, 晶体管级的仿真不同于门级仿真。后者的测试激励容易描述, 但是要给出晶体管级的测试激励则是一件很困难的工作, 一个 n 输入的电路, 就需要有 2^n 个输入向量。Nanosim 有专门的命令把门级的测试向量转变为晶体管级的测试向量^[6], 但是这种方法需要找到每一个被测状态, 对于大规模的数字电路, 尤其是数模混合电路的功耗计算并不可行。其次, 晶体管级的仿真是不能准确测到顶层单元以下每一个模块的漏电流功耗, 要测到每一个功能模块的功耗, 必须再单独描述模块的测试向量, 这对于上万门的多端口电路来说非常困难。所以, 为便于设计, 需要一种简单的计算方法帮助设计者快速估算出漏电流功耗。

2 基本原理

目前广泛使用的漏电流功耗计算模型有两种, 一种是采用遗传算法, 另一种是根据漏电流计算的

张 锋 男, 1977 年出生, 博士研究生, 研究方向为 CMOS 电路高速低功耗设计。

周玉梅 女, 研究员, 博士生导师, 从事双极与 BiCMOS 模拟集成电路设计。

黄令仪 女, 研究员, 从事 VLSI 数字电路设计。

2003-09-18 收到, 2003-11-05 定稿

© 2004 中国电子学会

经验公式采用统计的方法。本文基于经验公式的计算模型，把它与全定制的设计方法相结合，提出了一种自顶向下的基于晶体管级大规模数模混合电路漏电流功耗的计算方法。

计算芯片的漏电流功耗，首先要计算它的漏电流，下面是基于 BSIM 3 MOS 晶体管漏电流计算模型^[7~9]：

$$I_{\text{subth}} = A e^{\frac{-q}{nkT}(V_G - V_S - V_{\text{TH0}} - \gamma V_S + \eta V_{\text{DS}})} \left(1 - e^{\frac{-qV_{\text{DS}}}{kT}}\right) \quad (1)$$

其中

$$A = \mu_0 C_{\text{ox}} \frac{W}{L_{\text{eff}}} \times \left[\frac{kT}{q} \right]^2 \times e^{1.8} \quad (2)$$

式中 V_G 、 V_D 和 V_S 分别为晶体管的栅、漏与源电压； V_{TH0} 为零偏置时的阈值电压； γV_S 为线性体硅影响因子； η 为 DIBL（漏层阻挡减少）的系数； C_{ox} 为单位栅电容； L 与 W 分别为晶体管的沟道长度和宽度。

这种 BSIM 3 漏电流模型详细描述了影响漏电流大小的因素，由此可看出，公式(1)能够精确地计算出单管的漏电流，仿真软件 Hspice 也是采用的这种计算方法。虽然这个公式计算准确，但它计算繁琐，由于考虑了过多的参数，所以它不可避免地需要大量计算时间。另外还有一套计算漏电流功耗的经验公式^[9]：

$$I_{\text{leak}} = I_{\text{off}} \times \frac{W_{\text{tot}}}{X_s} \times X_t \quad (3)$$

式中 I_{off} 是在给定温度下一个单管的漏电流； W_{tot} 是所有晶体管（N 管与 P 管）的宽度； X_s 是基于观察与测算的堆积因子； X_t 是温度因子。其中， I_{off} 与 X_t 在一定温度下都为常量，也就是说只要计算出 W_{tot} ， X_s 就可以计算出总的漏电流值。 X_s 是计算漏电流的关键因子，它的含义是衡量每个单元导通漏电流大小的能力。要计算 W_{tot}/X_s 可以由下式求得

$$W_{\text{tot}}/X_s = \sum W_i f_i / X_{s_i} \quad (4)$$

式中 W_i 与 f_i 分别为每个单元的宽度以及每个单元出现的概率； X_{s_i} 为每个单元的堆积因子。

由公式(3)可以推出公式(5)：

$$X_s = (L_{\text{inv}}/L_{\text{design}}) N \quad (5)$$

式中 L 为漏电流模型的有效宽度； $N = W_{\text{design}}/W_{\text{inv}}$ 。由公式(5)算出堆积因子 X_s ，并由公式(3)，(6)得出漏电流功耗大小：

$$P_{\text{leakage}} = V_{\text{dd}} I_{\text{leak}} \quad (6)$$

3 基于全定制设计漏电流功耗计算

全定制设计与半定制设计的漏电流功耗计算方法完全不同，这主要是由设计流程不同造成的。图 1 是目前普遍采用的半定制设计流程与某些全定制 IP（如存储器）设计流程。可以看出，全定制与半定制设计的最大区别是，全定制设计在行为级设计之后直接过渡到以晶体管级为主的电路设计（也存在门级设计），这样是为了设计出理想的高速低功耗的电路形式。而半定制设计中有了标准单元门延迟的限制，常常无法达到设计的速度要求。在半定制设计中，通常是依靠计算门级电路的反转率来实现整个电路漏电流功耗的预估，但是漏电流功耗与输入激励之间的关系十分紧密，所以这种计算方法测得结果与实际情况往往会有很大差别。在全定制设计中，采用的主要还是晶体管级设计，电路仿真多采用 Spice 仿真。但是，对于一个由几万个晶体管组成的多端口 IP 来说，无论采用哪种 Spice 仿真软件，都需要大量的测试激励和仿真时间，这样会给设计者带来不便。根据以上算法模型，以及在龙芯 II 号全定制 IP 的设计经验，本文总结了一套适用于全定制 IP 的计算漏电流功耗的流程与方法，如图 2 所示。

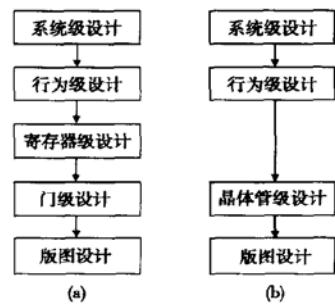


图 1 半定制(a)与全定制(b)设计流程
Fig. 1 Half custom(a) and full custom(b) design flow

在计算开始时把数字电路和模拟电路分开计算，是因为两种电路有着不同的计算方法，用前面所述的计算堆积因子的方法已不能简单地计算出模拟电路的漏电流功耗，要算出模拟电路漏电流功耗，必须根据所设计的模拟电路的特点，找到相应的方法。

对于大规模的数字电路，通常先要把复杂电路简单化，其中最为直接的办法是把要计算的模块展开到最基本的门单元，统计基本单元出现的概率以

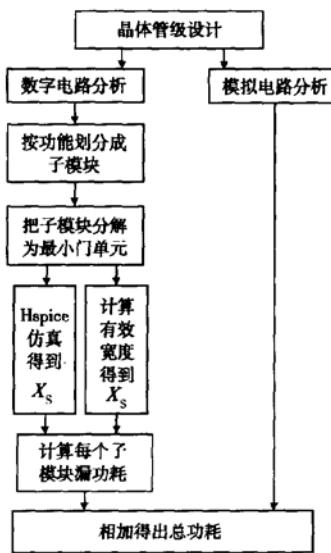


图 2 全定制漏电流功耗分析方法

Fig. 2 Analysis method of full custom leakage power

及相应的 X_s 来计算漏电流功耗. 较为麻烦的办法是, 计算初期先按照晶体管的数目以及子模块的大小平均划分, 以后就可以快速地计算出漏电流功耗. 但经过多次验证发现, 把大规模的数字电路按照每个子模块的功能划分, 是一种实用的计算方法. 按照功能划分的电路计算方法一般会比前两种的计算速度慢, 原因是前两种分解电路方法应用起来快速直接, 设计者可以根据整个 IP 的晶体管级网表, 编写程序直接进行处理计算, 设计者甚至不需要了解电路的形式与实现的功能. 按照功能划分则需要清楚地了解每一个子模块的功能, 工作量大. 但基于功能划分的计算方法有突出的优点:(1)最初的设计中, 电路形式的不合理在所难免, 按照功能划分, 往往可以在设计的同时很快找到漏电流功耗大的模块, 并做出相应的改进, 所以说它是可以完全与设计同步的;(2)本文所提出的数字电路漏电流功耗的计算方法, 需要统计电路中所有工作的晶体管的尺寸总和, 即 W_{tot} , 采用功能划分可以使计算更加清晰从而快速得出 W_{tot} . 下面分别就数字电路与模拟电路提出各自的计算方法.

3.1 数字电路漏电流功耗计算

计算 X_s 的一般方法是:

方法(1), 是由公式(5)通过计算得出每个单元的漏功耗, 公式(5)中只要计算出一个反相器的漏电流功耗, 然后由基本单元与反相器的有效宽度比例,

得出相应的 X_s . 根据漏电流的产生模型, 可以得出漏电流的有效宽度公式. 对反相器, 漏电流模型为:

设输入端为 A

$$A = 1, \text{ 有效宽度 } L = \frac{1}{2} \times W_p(L_p/u) \quad (7)$$

$$A = 0, \text{ 有效宽度 } L = \frac{1}{2} \times W_n(L_n/u) \quad (8)$$

可得公式

$$L_{inv} = \frac{1}{2} \times W_p(L_p/u) + \frac{1}{2} \times W_n(L_n/u) \quad (9)$$

对于二输入与非门, 漏电流模型为:

设两个输入端分别为 A 与 B

$$A = 1, B = 1, \text{ 有效宽度 } L = \frac{1}{4} \times W_p(L_p/u) \quad (10)$$

$$A = 1, B = 0, \text{ 有效宽度 } L = \frac{1}{4} \times W_n(L_n/u) \quad (11)$$

$$A = 0, B = 1, \text{ 有效宽度 } L = \frac{1}{4} \times W_n(L_n/u) \quad (12)$$

$$A = 0, B = 0, \text{ 有效宽度 } L = \frac{1}{4} \times W_n S_n(L_n/u) \quad (13)$$

其中 S_n 为 N 管串连的影响因子, 原因是输入都为 0 时, 两个串连的 N 管同时关断, 此时漏电流最小, 即有效宽度最小, 可以忽略. 可得二输入与非门的有效宽度公式:

$$L_{2nand} \approx \frac{1}{2} \times W_p(L_p/u) + \frac{1}{2} \times W_n(L_n/u) \quad (14)$$

同理可得三输入与非门的有效宽度公式:

$$L_{3nand} \approx \frac{3}{8} \times W_p(L_p/u) + \frac{1}{2} \times W_n(L_n/u) \quad (15)$$

由公式(5)可以看出, 假如给定 L_{inv} 为一个基本单位, 就可以得出其他单元的 X_s . 假定一个标准反相器的 $W_p = 2\mu m$, $W_n = 1\mu m$, 并假定 $L_p = L_n = 0.18\mu m$, 则带入公式(5), (9), (14), 可以得到 $X_s = 2(L_{inv}/L_{2nand}) \approx 2$, 同理可得三输入与非门 $X_s \approx 3.6$.

但是方法(1)存在一定的局限性, 尤其是在晶体管级设计中, 通常会出现很多的电路形式, 很难找到它的有效宽度, 这就需要另外一种计算方法.

方法(2), 采用 Hspice 对每个门级单元仿真, 然后根据每一个测试激励计算出相应的漏电流并代入公式(3), 可算出它的 X_s , 然后取平均值. 如表 1 所示, 这是 $0.18\mu m$ 工艺下的一个二输入与非门的漏

电流功耗测算. 可以计算出 $X_s \approx 1.95$, 通过这种方法可以得出每个单元的 X_s .

表 1 二输入与非门的漏电流功耗与堆积因子

Table 1 2 inputs NAND gate leakage power and stack factors

输入激励	0 0	0 1	1 0	1 1
漏电流/pA	2.3	12.7	11.6	16.1
堆积因子 X_s	4.1	1.3	1.4	1.0

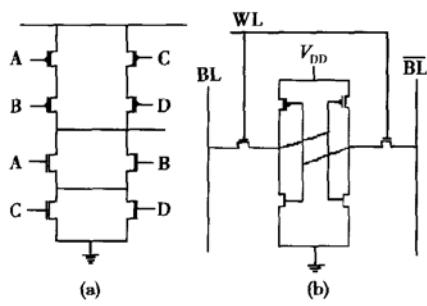


图 3 或非门(a)与 SRAM 六管单元(b)电路

Fig. 3 Schematic of NOR gate (a) and basic SRAM (b) cells

方法(1),(2)对不同的电路有着不同的实用性,可以根据实际需要选定计算方法. 如图 3 中对于四输入或非门,采用方法(1)计算 X_s 相对简单,而对于 CMOS SRAM 六管单元,如果用方法(1)则不容易计算出它的有效宽度,如果采用方法(2)会更实际,可以用 Hspice 仿真分别测出这四种状态的漏电功耗,然后计算出 X_s .

3.2 模拟电路漏电功耗计算

模拟电路的漏功耗计算不同于数字电路,数字电路的电路形式规则,重复单元多,而模拟电路的形式多种多样,即便是同一种电路结构,不同的晶体管尺寸都会引起功耗的很大不同,所以不容易找到它的功耗计算规律. 同时模拟电路的漏电功耗也有自己的特点,很多模拟电路的静态功耗通常会很大,占总功耗的比例很大,而且很难确定这部分功耗是“有用功”,还是“无用功”,这就为功耗计算带来了困难. 本文给出了一种基于大规模的数模混合电路的漏电功耗计算方案,如图 4 所示.

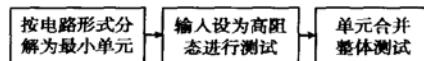


图 4 模拟电路漏电功耗分析流程

Fig. 4 Leakage power analysis flow of analog circuits

一般来说, 大规模数模混合电路中, 模拟电路部分面积通常较小, 所以它的静态功耗也很小, 但有时为了实现某种设计的要求, 在电路中会设计某些电源与地之间的直流通路, 这样就会带来较大的静态功耗. 由这部分通路所引起的静态功耗虽然不属于漏电功耗, 但它与漏电功耗性质相同, 其中很大一部分都属于无用的功耗. 之所以把输入设为高阻态进行仿真计算, 目的是找到电路在不加任何外部激励条件下所产生的直流功耗, 即“无用功耗”部分. 要得到电路的漏电功耗, 最终还要对顶层模拟电路用 Hspice 仿真, 这不会浪费很多的仿真时间, 因为模拟电路的晶体管数目通常不多, 而且输入端口较少, 对顶层模块来讲, 很容易找到测试激励, 所以可以快速计算出整个电路的静态功耗, 并由下式计算出漏电功耗.

$$P_{\text{leakage}} = P_{\text{static}} - P_{\text{direct}} \quad (16)$$

4 算法验证的结果与分析

在龙芯 II 号 CPU 设计中很多模块采用全定制设计, 如 3-8 译码器等. 其中最大的全定制模块是一个六读三写 64 位的寄存器堆^[10], 它大约由十万个晶体管组成, 而且是一个数模混合电路. 设计过程中为了验证这种计算漏电功耗方法的准确性, 分别用 Hspice, Nanosim 的仿真结果与算法的计算结果进行了对比, 如表 2 和表 3 所示.

这种计算方法有三个突出的优点:(1)对功耗估算较为准确. 实验证明, 这种估算方法与仿真结果差异不大. 由该计算方法可看出, 漏电功耗大小与晶体管的尺寸成线性关系, 算法的计算值是通过每一个最小基本单元的平均漏电功耗值累加得来, 而和最外围激励无关, 所以能够比较真实地反映出电路的正常工作状态.(2)计算快速. 由于计算方法简单, 可编程性强, 可以直接针对晶体管级的网表进行编程统计, 同时并不需要准备测试激励文件, 这样大大缩减了仿真时间. 本文的方法是采用 Pearl 语言编程实现的, 如果采用其他的编程语言也许还会提高效率.(3)非常适用于基于晶体管级的全定制 IP 设计, 原因是它即可以计算常规电路的漏电功耗, 又可以计算某些特殊电路的漏电功耗. 对某些特殊的电路形式, 在采用常规方法计算有效宽度有难度时, 完全可以用 Hspice 仿真测得它的 X_s .

表 2 Hspice 仿真结果和 Nanosim 仿真结果

Table 2 Hspice simulation results and Nanosim simulation results

	晶体管 个数	测试 激励数	Hspice 仿真结果			Nanosim 仿真结果		
			一次仿真 时间/min	总的仿真 时间/min	漏电流功耗 / μ W	一次仿真 时间/min	总的仿真 时间/min	漏电流功耗 / μ W
3-8 译码	60	2^3	0.1	0.1×2^3	4.2×10^{-3}	0.1	0.1×2^3	4.4×10^{-3}
8 位加法器	278	2^8	1	1×2^8	1.3×10^{-2}	0.5	0.5×2^8	1.4×10^{-2}
64 位加法器	$\sim 10^3$	2^{64}	10	10×2^{64}	3.1×10^{-1}	1.5	1.5×2^{64}	3.4×10^{-1}
64 位三写六读寄存器堆	$\sim 10^5$	2^{264}	—	—	—	30	30×2^{264}	2.52×10^2

注:(1)由于仿真时间不可能精确测量,所以只是一个时间估计值;(2)短线部分表示时间太长无法仿真.(3)漏功耗抽样选取平均值

表 3 通过算法计算的漏电流功耗

Table 3 Test the leakage power by algorithmic method

	算法计算结果				
	激励	测算时间(程序运行时间)/min	漏电流功耗/ μ W	与 Hspice 结果对比	与 Nanosim 结果对比
3-8 译码	无	0.1	4.7×10^{-3}	相差 11%	相差 6%
8 位加法器	无	0.5	1.5×10^{-2}	相差 14%	相差 7%
64 位加法器	无	1	3.6×10^{-1}	相差 15%	相差 6%
64 位三写六读寄存器堆	无	2	2.78×10^2	—	相差 10%

5 结论

本文扩充了原来的理论基础,而且更加注重了理论的实用性,从实际情况考虑问题,弥补了很多仿真软件的不足之处,大大提高了晶体管级设计漏电功耗估算的效率,缩短了设计周期.另一方面,本文在原方法的基础上提出了模拟电路的功耗计算方案,用这种方法可以快速确定模拟电路的漏电功耗.为了验证该方法,把它应用于龙芯 II 号全定制 IP 的设计中,目前针对以上某些全定制模块的流片设计已成功,在测试了功耗后也验证了设计的合理性.所以这种漏电功耗计算方法对于 $0.18\mu\text{m}$ 以及更小线宽工艺的芯片设计均有普遍的指导意义,完全可以开发相应软件并投入使用.

参考文献

- [1] Afshin A, Farzan F, Massoud P. Runtime mechanisms for leakage current reduction in CMOS VLSI circuits. The International Symposium on Low Power Electronics and Design, 2002: 213
- [2] Bobba S, Hajj I N. Maxim leakage power estimation for CMOS circuits. Proceedings of IEEE Alessandro Volta Memorial Workshop on Low-Power Design, 1999: 116
- [3] Nachiket D S. Leakage power estimation model. Proceedings of Computer Science and Engineering, 1998: 539
- [4] Vivek T. Power challenges in the internet world. The Cool Chips Tutorial and an Industrial Perspective on Low Power Processor, 1999: 354
- [5] Synopsys corporation. Hspice user's manual
- [6] Synopsys corporation. Nanosim userguide
- [7] Sheu B J, Scharfetter D L, Ko P K, et al. BSIM: Berkeley short-channel IGFET model for MOS transistors. IEEE J Solid-State Circuits, 1987, 22(4): 558
- [8] Chen Zhanping, Wei Liqiong, Johnson M, et al. Estimation of standby leakage power CMOS circuits considering accurate modeling of transistor stacks. The International Symposium on Low Power Electronics and Design, 1998: 239
- [9] Jiang Wenjie, Vivek T, Erik de la, et al. Topological analysis for leakage prediction of digital circuits. Proceedings of the 15th International Conference on VLSI Design, 2002: 39
- [10] Wang Jiajing, Hua Lin, Shen Bo, et al. 500MHz, 32×32 bit, high-speed, five-port CMOS register file. Chinese Journal of Semiconductors, 2002, 23(12): 1320(in Chinese) [王佳静, 华林, 沈泊. 等. 一种 500MHz 32×32 bit 高速五端口 CMOS 寄存器堆. 半导体学报, 2002, 23(12): 1320]

Method of Leakage Power Simulation and Calculation Based on Full Custom IP Design

Zhang Feng, Zhou Yumei and Huang Lingyi

(Institute of Microelectronics, The Chinese Academy of Sciences, Beijing 100029, China)

Abstract: A top-down method of leakage power test based on transistor-level full custom IP design is presented. This method runs fast, which replaces the technology absolutely dependent on the software simulation. It is utilized in full custom IP design of Godson II CPU. The SMIC 0.18 μ m technology is used in this chip. To validate the method, we contrast the results to our method and the Synopsys' Nanosim simulation. The error is only about 10%. Because software simulation needs a lot of vectors and test time, the method can calculate its leakage power without any outside input vectors. It can quickly locate the place which has much power, thus shortens the test period. So the invention of the software by the arithmetic method can be widely used.

Key words: leakage power; stack effect; efficient width; high resistor state

EEACC: 2570A; 1280; 1265D

Article ID: 0253-4177(2004)09-1169-06

Zhang Feng male, was born in 1977, PhD candidate. His research interests are high-speed and low power CMOS circuit design.

Zhou Yumei female, professor. Her research interests include bipolar and BiCMOS analog circuits design.

Huang Lingyi female, professor. Her research interests include VLSI CMOS digital circuits design.