

基于有用时序偏差的时序优化方法

李雪艳¹, 廖一鹏²

(1.福州大学 阳光学院, 福建 福州 350015;

2.福州大学 物理与信息工程学院, 福建 福州 350108)

摘要: 提出了一种利用有用时序偏差来提高电路性能的方法, 利用时钟偏差规划算法在时钟树综合之前对时序偏差重新调整规划, 以提高电路的性能。使用 ISCAS89 作为实验对象来验证算法并进行了分析。

关键词: 时钟树综合; 时序偏差; ISCAS89

中图分类号: TN402

文献标识码: A

文章编号: 1674-7720(2013)04-0081-04

The timing optimization based on useful timing skew

Li Xueyan¹, Liao Yipeng²

(1.Sunshine College of Fuzhou University, Fuzhou 350015, China;

2.College of Physics and Information Engineering, Fuzhou University, Fuzhou 350108, China)

Abstract: The article introduces a method which makes use of the useful skew to enhance the performance of the circuit. Using the algorithm of local optimization of timing skew, we make plan for the clock skew in order to improve the performance. And we use ISCAS89 to verify our algorithm and analyze the result.

Key words: clock tree synthesis; clock skew; ISCAS89

在数字集成电路设计中, 基于标准单元的 ASIC 设计具有成本低、移植性好、上市时间短等优点, 在各种芯片设计领域占据了越来越重要的地位^[1]。ASIC 设计的广泛应用使得 EDA 的方法在 IC 设计中得到了充分的发展。在工艺进入深亚微米之后, 对芯片设计尤其是芯片的物理设计提出了更高的要求, 芯片的时序收敛周期也变得越来越大。本文所研究的时钟偏差规划基于标准单元 ASIC 设计流程, 能够提高芯片的性能。

在现在的超大规模电路设计中, 一般采用寄存器传输级设计。在这种设计方法中, 时钟信号是所有数据传送的基准, 数据传送从时钟发送沿开始, 到时钟接收沿结束。在同一时钟域中, 时钟源所扇出的单元, 包括寄存器、存储器在内, 数量非常巨大, 是一个庞大的时钟网络。如图 1 所示, 为了使同一时钟源能够驱动到它所扇出的所有节点, 并使其到这些节点的延时基本相同, 时钟网络通常采用缓冲器树的结构, 这一结构称为时钟树^[2]。

时钟树在时钟信号起点以及时钟信号所要驱动的

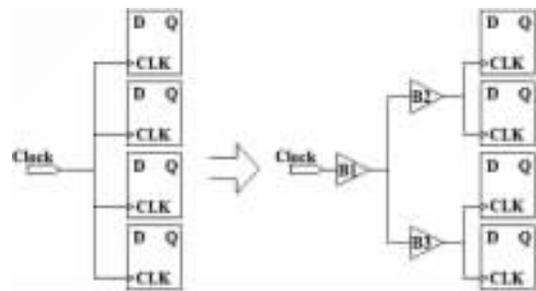


图 1 时钟缓冲树结构

所有寄存器之间, 加入若干缓冲器(buffer), 形成树状架构。在这种架构中, 时钟信号到达寄存器会产生偏差(skew), 所有寄存器时钟端的时钟信号并不是同时到达的。现在工具优化的目标是“零时钟偏差”, 零时钟偏差是一种让时钟信号同时到达所有寄存器的设计方法。实际上利用时钟偏差构建时钟树, 并不会比零时钟偏差的时钟树对工艺变化更敏感, 通过适当的时钟偏差优化, 还可以提高电路对工艺参数变化的容忍度^[3]。在关键路径上的一些时钟偏差会对时序有一定的优化作用, 从而提高电路的最大运行速度, 这种时钟偏差即为有用时钟

技术与方法 Technique and Method

偏差^[4]。有时钟偏差现在已经成为集成电路设计中一个备受关注的研究方向。

1 静态时序分析

静态时序分析是通过在最悲观的情况下检查所有时序路径可能存在的时序违规的方法来保证一个设计在时序上的性能。因此它只考虑每一个逻辑门的最差延时,并不理会电路的逻辑功能^[5]。

图2中虚线所示为一条常见的时序路径,它的起点是寄存器FF1的CLK端,终点为FF2的D端。在时序分析中建立时间检查时,要求这条路径满足的约束条件如式(1)~式(3)所示。

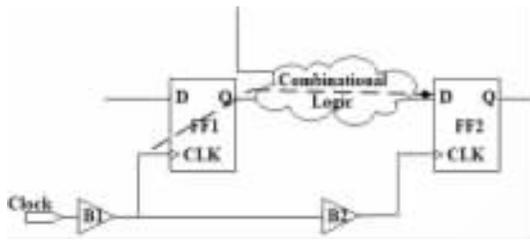


图2 建立时间检查示意图

$$T_{required}(FF2) = T_p + L(FF2) - T_{setup}(FF2) \quad (1)$$

$$T_{arrival}(FF2) = L(FF1) + T_{CLK \rightarrow Q}(FF1) + T_{max}(Comb) \quad (2)$$

$$Slack(FF2) = T_{required} - T_{arrival} \geq 0 \quad (3)$$

其中, T_p 为电路运行的时钟周期; L 表示时钟源到寄存器所经历的延时; T_{setup} 表示寄存器的建立时间, 这是标准单元库中寄存器的一个重要参数, 表示时钟有效接收沿到达之前数据必须保持不动的时间。Slack 表示这条路径的时序裕量, 当裕量大于等于零时, 说明电路不存在时序违规, 在周期为 T_p 时运行不会有时序问题。

从式(1)、式(2)可以看出, 电路的性能不仅决定于电路组合路径的延时, 时钟源到寄存器的延时 L 也是一个很重要的影响因素^[6]。如果时钟源到寄存器的时钟端的延时不同, 就会出现时钟偏差。

静态时序分析理论表明, 设计存在时钟偏差并不意味着时序恶化, 正的时钟偏差(信号与时钟同向)可以修正建立时间违规, 而负的时钟偏差(信号与时钟反向)可以修正保持时间违规^[7]。时钟偏差规划正是通过协调时钟延时来优化时序、提高性能的。

2 算法描述

如图3所示, 以 Launch_reg 与 Capture_reg 两个寄存器组成的一条路径为当前路径, 定义以 Launch_reg 为终

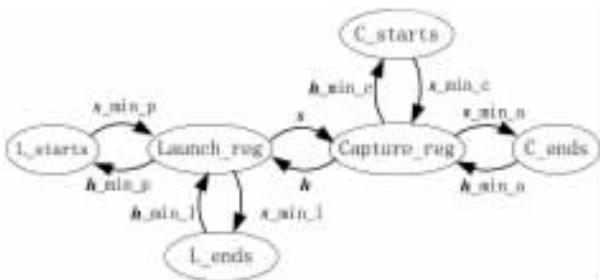


图3 路径的定义约束图

点的路径是当前路径的前级, 以 Capture_reg 为起点的路径为当前路径的后级。算法的实质是先进行建立(setup)时间裕量的平分, 再根据不同的情况来进行调整, 以达到所有路径都无时序违规的目的。该方法在提高性能的同时, 兼顾前后级时序电路的裕量, 不至于使它们全成为临界边(裕值为0的边)。

通过时序分析工具获得当前路径, 前级以及后级路径的时序信息作为时序均分的初始值, 然后将各级时序裕量进行求和均分。在约束图中将某节点的时钟延时增加 Δ , 则以它为终点的路径的裕量增加 Δ , 而以它为起点的路径的裕量减少 Δ ; 如果将某节点的时钟延时减少 Δ , 情况则相反。在均分以后必须使得保持时间不能违规, 寄存器的保持时间必须大于0, 否则数据将无法被寄存器准确地捕获。

根据时序参数初值以及各级均分后的值可以算出本级、前级和后级路径新的保持时间和建立时间。判断前后级路径是否有保持时间违规, 若前后级有违规则将时序初始值设置为自身保持时间裕量, 调整新的时序初始值后继续进行时序裕量均分, 直到无法再调整为止。判断本级路径是否有保持时间违规, 若自身有违规则增量值 Δ 调整为原来的 k 倍 (k 为偏差在当前情况下实际调整值与最大调整值的比值) 作为最终调整的结果; 若无违规则以当前调整过的值作为结果。算法流程如图4

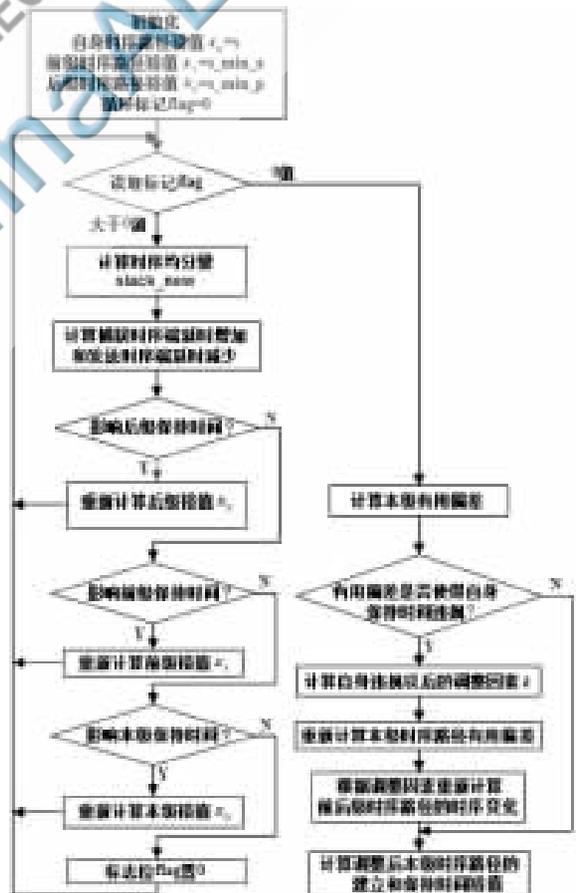


图4 局部时序偏差优化流程图

技术与方法 Technique and Method

所示,在具体实现算法中,使用了标记 flag 控制循环。算法使用 TCL (Tool Command Language) 语言实现,很方便地嵌入现有流程与现有工具集合起来使用。

通过静态时序分析工具得到关键路径建立时间,如图 5 所示。以此为例分析算法作用的过程:当前情况下时钟约束的周期为 10 ns,那么在 -2 ns 违规的情况下,时钟必须提供 12 ns 的周期,若通过局部时序偏差优化的方法将 -2 ns 的违规修正,则提高了电路的性能。

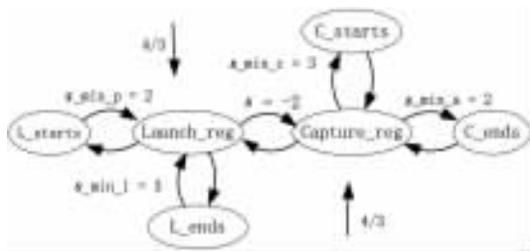


图 5 关键路径 (单位: ns)

Launch_reg 到 Capture_reg 的路径建立(setup)时间为 -2 ns,其前后级的建立时间均为 2 ns,均分以后,保持时间 s 、 $s_{min,p}$ 、 $s_{min,a}$ 都为 $2/3$ ns,如图 6 所示,此时以 Launch_reg、Capture_reg 为终点的路径建立时间减少了 $4/3$ ns, $s_{min,l} = 3 - 4/3 = 5/3$ ns, $s_{min,c} = 3 - 4/3 = 5/3$ ns,而以 Launch_reg、Capture_reg 为起点的路径建立时间则增加了 $4/3$ ns。



图 6 偏差优化以后的关键路径 (单位: ns)

偏差优化后时序最差的路径由原来的 -2 ns 变为 +2/3 ns,周期由原来的 12 ns 提高到了 10 ns 以内。

提高电路性能:正的时钟偏差,其效果相当于减小逻辑路径的时间延迟。因此有效地利用时钟偏差,对时钟信号到达各个寄存器的时间进行适当的安排,时钟周期的最小值将可能小于电路中所有相邻两个寄存器间的信号延迟时间的最大值,而达到提高电路性能的目的^[8]。

增加电路稳定性:令 T_{spec} 为电路实际运行的时钟周期,但由于工艺的变化,使得电路分析结果时钟周期的最小值为 T_{min} ,很明显, T_{min} 必须小于 T_{spec} 才能合乎设计要求。如前文所述,利用时钟偏差,可缩小时钟周期的最小值。令 T_{spec} 和 T_{min} 的差值为 ΔT ,由于工艺参数变化等因素, ΔT 的值变得越大,越有助于确保电路运行正确。同理,通过时钟偏差优化,也可以提高保持时间裕量。

3 实验过程

利用时钟偏差的区域性特点,可以改善电路的性能及稳定性,但目前业界广泛使用的集成电路时钟树综合

工具(如 IC Compiler),其目的都是缩小时钟偏差,即使工具附带有用偏差约束的时钟树综合选项,但因为所消耗的资源太高,而且最终的结果也不尽如人意。本文提出的偏差优化算法及其流程,可以很好地与现有的集成电路设计流程进行整合。时钟偏差规划所针对的是门级网表,必须要在逻辑综合以后。虽然越往后的流程,设计所拥有的信息越全面准确,但是时钟树综合以后时钟结构已经固定,再进行优化的余地较小,所以时钟偏差规划要位于时钟树综合以前。时钟偏差规划嵌入传统流程的最佳位置在布图规划、布局之后,时钟树综合之前。嵌入偏差优化后的新的物理设计流程如图 7 所示。

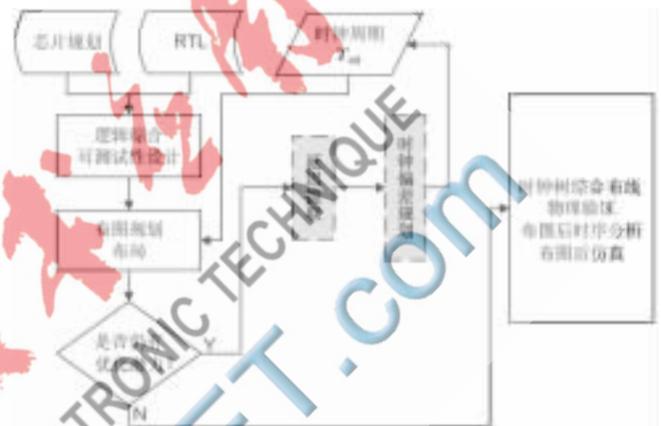


图 7 偏差优化嵌入后新流程

本文以 ISCAS89 中部分电路为实验用例,对它们进行时钟偏差的优化来改善芯片的性能和稳定性。实验中首先选取“SMIC 130nm Logic013G Process 1.2-Volt SAGE-XTM v2.0 Standard Cell Library”标准单元库作为逻辑综合的目标库(Target Library),采用 Synopsys Design Compiler 将实验用例的 RTL 代码综合成门级网表,采用 Synopsys IC Compiler 进行布图规划。

在完成布图规划后,嵌入的流程有两步:

(1)延迟提取。延迟提取是实施时钟偏差规划的数据基础。此步骤在静态时序分析工具中进行。

(2)时钟偏差规划。利用规划来获得性能最优的时钟树方案,并生成适合时钟树综合使用的命令。

时钟偏差规划优化以后,设计中原来的关键路径使用时钟偏差的方法得以修正,减少了工具的负荷,更大地发挥了工具的优化能力,得到更好的优化效果。优化完成以后再进行一次时钟偏差规划调整,直到继续优化的程度非常小,不值得继续循环优化为止。

4 实验结果

时钟偏差优化的结果使得性能得到提升。性能的提升可以用最小周期的减小值或百分比表示。初始时钟最小周期记为 T ,偏差优化后的最小时钟周期记为 T_{new} ,而性能的提升(performance enhancement)为: $\Delta T = T - T_{new}$,为更加形象地显示性能提高的幅度,亦可采用性能提升的百分比 $\Delta T/T$ 来表示。每个测试用例时钟偏差优化前后

技术与方法 Technique and Method

的性能提升比例如表 1 所示。

表 1 时钟偏差优化性能提升比例

实例名称	初始最小周期 T/ps	优化后 T_{new}/ps	性能提升值 $\Delta T/\text{ps}$	提升比例/%
s1423	2 051	2 006	45	2.2
s5378	1 975	1 904	71	3.6
s15850	3 878	3 623	255	6.6
s13207	2 050	1 877	173	8.3
s38584	4 038	3 671	367	9.1
s35932	1 980	1 942	38	2.0

局部偏差优化算法对实验用例的性能有不同程度的提升, s38584 的提升幅度最大为 9.1%。

高性能的系统芯片中, 时钟信号对于系统功能和性能都具有至关重要的作用。随着工艺尺寸的缩小, 时钟偏差并不等比例缩小, 因此以零偏差为目标的时钟树综合已经难以适应工艺进步的需要。时钟偏差优化作为提升电路性能及稳定性的研究越来越深入, 但如何有效地利用时钟偏差, 设计方法及工具尚不完善。对学术界经典的 ISCAS89 中不同规模的电路进行实例测试, 结果表明算法具有很强的实用性, 能够很好地提升电路的性能。

参考文献

- [1] RABAHEY J M, CHANDRAKASAN A. 数字集成电路——电路、系统与amp;设计(第 2 版)[M]. 周润得, 译. 北京: 电子工业出版社, 2004: 237-238.
- [2] Synopsys, Inc. Astro user guide[DB/OL]. (2007-09-01) [2012-03-29]. [http://wenku.baidu.com/view/584b8b20a-](http://wenku.baidu.com/view/584b8b20a-f45b307e8719720.html)

f45b307e8719720.html.

- [3] HUANG S H, NIEH Y T. Synthesis of non zero clock skew circuits[J]. IEEE Transactions on ComputerAided Design of Integrated Circuits and Systems, 2006, 25(6): 961-976.
- [4] MADER R, FRIEDMAN E G, LITMAN A. Large scale clock skew scheduling techniques for improved reliability of digital synchronous VLSI circuits[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2002(1): I-357-I-360.
- [5] BHASKER J, CHADHA R. Static timing analysis for nanometer designs[M]. Berlin: Springer Verlag, 2009: 179-360.
- [6] 陈春章, 艾霞, 王国维. 数字集成电路物理设计[M]. 北京: 科学出版社, 2008: 93-127.
- [7] Synopsys, Inc. Prime time user guide[DB/OL]. (2006-06-01) [2012-03-29]. <http://wenku.baidu.com/view/530afd0e795-63c1ec5da71b0.html>.
- [8] WANG K, DUAN L, CHENG X. Extensive slack balance: an approach to make front-end tools aware of clock skew scheduling[C]. In the Proceedings of the 43rd DAC, 2006: 376-385.

(收稿日期: 2012-10-19)

作者简介:

李雪艳, 女, 1983 年生, 在读研究生, 主要研究方向: 电路与系统。

廖一鹏, 男, 1982 年生, 硕士, 讲师, 主要研究方向: 物联网技术, 电路与系统。