Journal of Sensor Technology and Application 传感器技术与应用, 2024, 12(6), 757-763
Published Online November 2024 in Hans. https://www.hanspub.org/journal/jsta
https://doi.org/10.12677/jsta.2024.126083

Hans汉斯

基于40 nm数字芯片的复杂时钟结构分析及优化

李治民，王　爽

天津职业技术师范大学电子工程学院，天津

收稿日期：2024年8月30日；录用日期：2024年10月22日；发布日期：2024年10月30日

摘　要

在复杂时钟结构芯片设计的物理实现中，基于Innovus工具采用传统时钟树综合流程得到的时钟树，具有时序违例大、插入缓冲器单元多、功耗大等问题，会给整个芯片设计带来挑战和困难。本文在传统时钟树流程基础上，采取在时钟树综合之前编写一个时序约束文件来分段长时钟树的方法进行改进优化。与传统方法的结果相比，最终得到一个级数较低的高质量时钟树，该时钟树时序违例小，违例路径减少85条，插入的缓冲器数目减少了2676个。新方法能有效地降低了芯片的总功耗以及节省了大量的空间面积，解决了局部绕线阻塞问题，并提高了芯片的工作性能。

关键词

物理实现，Innovus，时钟树，缓冲器，功耗

Analysis and Optimization of Complex Clock Structure Based on 40 nm Digital Chip

Zhimin Li, Shuang Wang

School of Electronic Engineering, Tianjin University of Technology and Education, Tianjin

Received: Aug. 30th, 2024; accepted: Sep. 22nd, 2024; published: Oct. 30th, 2024

Abstract

In the physical implementation of complex clock structure chip design, the clock tree obtained based on the Innovus tool using the traditional clock tree synthesis process has problems such as large timing violations, many insertion buffer units, and large power consumption, which will bring challenges and difficulties to the entire chip design. Based on the traditional clock tree process, this paper adopts the method of writing a timing constraint file to segment the clock tree before the

文章引用: 李治民, 王爽. 基于 40 nm 数字芯片的复杂时钟结构分析及优化[J]. 传感器技术与应用, 2024, 12(6): 757-763. DOI: 10.12677/jsta.2024.126083

clock tree synthesis. Compared with the results of the traditional method, a high-quality clock tree with a low series is obtained, which has small timing violations, 85 fewer illegal paths, and 2676 buffers inserted. The new method can effectively reduce the total power consumption of the chip, save a lot of space area, solve the problem of local winding blockage, and improve the working performance of the chip.

Keywords

Physical Implementation, Innovus, Clock Tree, Buffer, Power

1. 引言

随着集成电路工艺节点不断突破，芯片设计规模越来越大，逐步接近摩尔定律的物理极限，IC 设计也面临着重重困难和挑战。数字集成电路设计中，数字电路门级网表在流片之前，必须要满足时序收敛以及将所有违反物理规则的错误都清空后，才能保证流片后的芯片可以正常工作。时序是贯穿整个数字芯片设计流程中，有着至关重要的地位[1]。因此，时钟树的设计是数字后端的关键，高质量的时钟树可以高效率地完成时序收敛[2]。高质量的时钟树通常追求时钟插入延时小、时钟偏移小、串扰小且公共路径长。在实际工程项目中，单一的时钟信号早已无法满足大规模的数字芯片设计，通常是由数十个或数百个时钟信号构成一个复杂时钟电路结构设计，用来满足整个芯片的设计功能[3]。同时，在时钟树设计阶段还需要考虑功能模式和测试模式，设计一个可以同时使多个模式下的时钟树平衡，并满足芯片的时序收敛要求已经成为行业的一大挑战。由于设计规模越来越大，时钟结构越来越复杂，以及不同模式下的时钟树平衡要求，传统时钟树综合方法已经不太适用，无法构建出一个高质量的时钟树，难以达到时序收敛的要求。

本文以一款具有时钟产生电路功能的数字芯片为例，采用中芯国际 40 nm 工艺，该设计共有 100 个时钟 clock，最高时钟频率为 1.25 GHz。通过对传统方法进行改进，设计出一个高质量的时钟树，且满足时序收敛要求的时钟网络，同时使芯片面积高效利用，在功耗方面也优于传统的时钟树综合方法。

2. 复杂时钟结构概述

2.1. 时钟树的实现

时序分析在 IC 设计中是必要的，若时序不满足设计要求，会使数据在传输过程中无法正确传达，使工作采集数据是错误的，会导致生产的芯片无法正常工作，从而导致设计失败。因此，需要通过做一个好的时序约束和高质量的时钟树来解决这个问题，最终目的是为了让芯片能够正常工作。

芯片设计最初的时钟结构较为简单，通常只有两个或三个时钟源就可以完成较为简单的功能设计，在做时钟树综合时，时钟信号从根节点出发，根据树状图形一级一级向下传递，到达至叶节点完成信号同步，得到一个叶节点较为平衡的高质量时钟树。该时钟树通过插入少量的缓冲单元，而且时钟偏差小，后续在进行时序收敛阶段的工作时，人工修改的部分较少，能够加快芯片设计的流程进展[4]。本文设计一个时钟产生电路来生成不同时钟频率信号，用来满足芯片对时钟信号的需求，使时钟信号的数增加至十几个，不同频率的时钟信号之间存在交互。在本次设计中还会设不同的工作模式，如实现逻辑功能的

工作模式和规模量产后的用于检查的测试模式。面对时钟信号多而复杂的情况，在进行时钟树综合阶段时，传统的综合方法并不能保证不同模式下的时钟树都做到平衡。在传统方法下，通过人工修改进行反复迭代工作，也很难满足最终的时序收敛的验收工作[5]。

2.2. 复杂时钟结构的特点

在数字芯片设计物理实现过程中，复杂时钟结构除了时钟信号的数量众多之外，还有以下几个特点：

1) 部分模块的寄存器受不同的时钟驱动，即实现部分功能需要多个时钟源共同驱动一个模块才能实现；

2) 芯片设计规模比较大，而芯片的面积是有限的，在做时钟树平衡时，由于会插入大量的缓冲器单元导致面积不够使用，对时钟路径长度要求比较高；

3) 芯片时钟结构逻辑复杂，在布局绕线时，用于时钟树的时序缓冲单元容易与组合逻辑单元造成区域阻塞，对时序检查造成不利的影响。

3. 时钟树综合流程

3.1. 传统时钟树综合方法

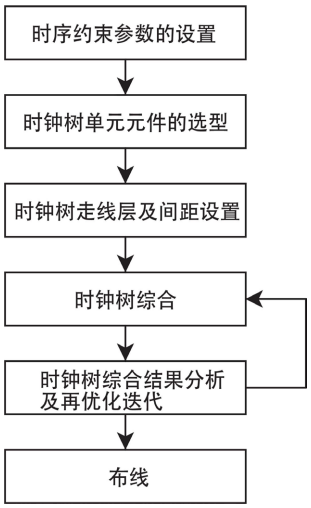

Figure 1. Clock tree comprehensive flow chart
图 1. 时钟树综合流程图

传统的时钟树综合方法如图 1 所示，根据 Cadence 公司的 Innovus 工具推荐的时钟树综合流程，进行一系列设置，生成一个可基于 ccopt 引擎进行时钟树综合的 spec 文件，从而完成时钟树的生成[6]。

第一步需要对时序约束参数的设置，如对 target skew、target latency 进行设置，以上约束设置是用来引导工具使生成合理的时钟偏移数值、时钟走线延时数值，设置值过小，导致设计过于悲观，设置值过大，会导致有限的资源浪费，对芯片的整体设计都会产生不利的影响，对 max fanout、max transition 进行设置，以上设置需要参照工艺规则的要求进行定义。

第二步需要对时钟树用到的 cell 进行指定，如缓冲单元 buffer cell、时钟单元 inverter cell 和时钟门控单元 clock gating cell 进行选型，通常会选择一个系列同一个型号的单元，同时去掉系列内驱动能力最弱的和最强的两个单元型号，最弱的两个 cell delay 比较大，而最强的两个由于驱动能力太强，电流比较大，可能会有电迁移效应。

第三步需要对时钟树走线的层数，以及 leaf、trunk、top 的走线进行 NDR 绕线规则设置。在 innovus

里面，clock path 的终点与最后一级缓冲单元相连的 net 被称为 leaf net，其他的 clock net 都被称成为 trunk net。一般为了减少 trunk net 上的 net delay，通常采用双倍线宽双倍间距的 NDR 绕线规则。通过走高层金属并增加线宽来减少 net delay；通过增加走线间距来避免出现串扰问题。而 leaf net 由于长度较短，数量较多，一般用常规的绕线规则。倘若也使用上述规则的话，会造成绕线资源紧张且 net delay 也没有任何的改变。

第四步，完成上述一系列设置后，将这些内容生成一个 spec 类型的约束文件，使 ccopt 引擎根据这个约束文件进行时钟树综合[7]，命令如下：

```
create_ccopt_clock_tree_spec -file xxx.spec
source xxx.spec
ccopt_design -cts
```

第五步，完成时钟树综合后，通过查看报告文件进行分析，对 spec 约束文件的内容进行更新修改[8]，删除原有的时钟树结构，重复第四步阶段进行时钟树综合，反复进行迭代优化工作，最终得到一个好的时钟树结构可以进行后续的设计工作。

对于时钟结构较为简单的设计，使用传统时钟树综合方法，通常会得一个功耗性能相当的高质量时钟树结构。然而面对复杂时钟结构，采用传统的流程会导致综合时间过长，甚至会出现综合失败的情况，这对设计时间紧张是非常不利的。同时，传统流程生成的 spec 文件，并不会对复杂时钟结构进行分析，不能对一些关键路径和特殊路径进行设置处理，导致时钟树综合过程变长，使整体时钟路径长度变得很长。

3.2. 时钟电路图分析法

在上一节了解传统时钟树综合方法后，该方法对复杂时钟进行时钟树综合具有一定的局限性。面对复杂时钟结构需要对传统时钟树综合方法流程进行一个优化改进，即在进行时钟树综合之前，通过 Synopsys 公司的 Verdi 工具，对该复杂时钟结构进行可视图分析。根据该电路图，对一些特殊的时钟电路图重新定义新的起点和终点，并且对时钟汇点进行约束设置，在时钟树综合之前添加一个新的时序约束文件[9]，从而实现时钟树分段长 tree 的效果。以下对本次时钟产生电路设计中的几个较为常见的时钟电路图进行分析说明。

如图 2 所示，该电路为不同模式的工作进行选择，MUX 器件只可以通过 I0 或 I1 一个端口。由同源时钟引出两条分支后，再经 MUX 电路的时钟结构。在传统时钟树综合方法中，因为是同源时钟电路，该电路上下两条分支是需要做平衡的，即两条支路到达 MUX 的时钟延时尽可能接近。上路分支穿过的是普通标准单元，到达 MUX 延时是 2 ns，而下路分支穿过的是高频单元，到达 MUX 的延时高达 15 ns，工具会在上路分支插入 clock inverter 单元来做平衡。clock inverter 单元平均延时大约为 20 ps 左右，因此会在上路分支插入大量的 clock inverter 单元。事实上，由于上下两条支路没有交互，这两条支路虽然是同源时钟，但并不需要做同步，可以在下路分支设一个为 ignore pin 的终点，在做时钟树综合时让工具对该点独自长 tree，不必与另一支路做时序平衡。相关脚本命令为：

```
set_case_analysis 0 [get_pins xx/xx/U_DONT_THUCH_CLKMUX2/S]
set_ccopt_property sink_type -pin xx/xx/U_DONT_THUCH_DLL/CP ignore
```

如图 3 所示，该电路是时钟切换电路，实现不同功能时，切换对应的时钟信号。由不同源的两个时钟通过 MUX 选择器进行时钟切换传递给 MUX 后的逻辑电路。在经 MUX 之前，不同的时钟信号也会与其他支路进行交互。对于不同源的异步电路一般不需要做时序分析，但是有 MUX 器件进行时钟切换的电路，这里的正确时序约束就很重要[10]。在传统的时钟树综合方法中，会对来自不同源的时钟，分别

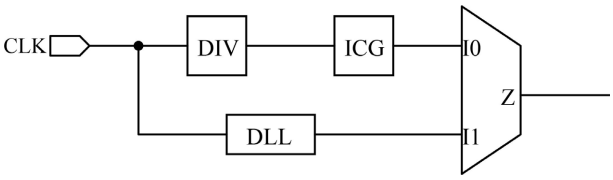

Figure 2. The clock passes through a large delay cell circuit
图 2. 时钟通过大延时单元电路

进行时钟树综合分析。在复杂时钟结构的设计中，MUX 器件是常用的电路结构，同时每一个时钟都会有许多的支路，不合理的时序约束会使工具对每条支路进行综合，导致分析时间过长，且无法顺利完成时钟树的综合。对于该时钟切换电路，可以在 MUX 的 Z 端创建一个新的起点，同时对 I0 端和 I1 端进行 ignore 处理，表示这两个不同的时钟信号不进行 balance 处理，同时 Z 端重新长一条时钟树。相关脚本命令为：

```
create_clock -name CLKMUX -period $clock_period xx/U_DONT_THUCH_CLKMUX2/Z
set_ccopt_property sink_type -pin xx/U_DONT_THUCH_CLKMUX2/I0 ignore
set_ccopt_property sink_type -pin xx/U_DONT_THUCH_CLKMUX2/I1 ignore
```

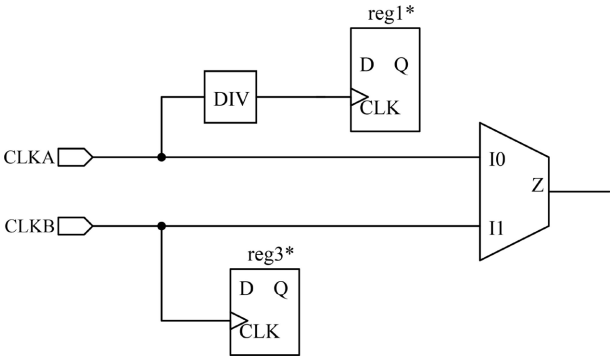

Figure 3. Non-homologous clock circuits
图 3. 非同源时钟电路

如图 4 所示，该电路为经过时钟切换电路后需进行 DIV 生成分频电路。由图 3 MUX 的 Z 端时钟引出几条不同分支的时钟结构。传统时钟树综合方法是工具以最长的一条分支作为参考，然后将所有分支的寄存器都做等长。由于不同分支的寄存器之间没有交互，不会有任何的时序检查，如果所有分支做等长，会导致最短的分支的寄存器延时增大，导致其出现不必要的时序违例。对于该 DIV 分频电路，可以在图 3 MUX 的 Z 端新建时钟树起点后，并且对或门电路的 A1 端和 A2 端进行 ignore 处理，使该电路的分支单独做时钟树综合，不用与其他电路进行综合。相关脚本命令为：

```
create_clock -name CLKMUX -period $clock_period xx/U_DONT_THUCH_CLKMUX2/Z
set_ccopt_property sink_type -pin xx/xx/U_DONT_THUCH_CLKOR/I0 ignore
set_ccopt_property sink_type -pin xx/xx/U_DONT_THUCH_CLKOR/I1 ignore
```

当完成对整个时钟产生电路的分析后，将所有的脚本命令进行编写一个完整的 sdc.tcl 文件，运用于时钟树综合的过程中。相关的命令脚本为：

```
create_constraint_mode -sdc_files ./../sdc.tcl -name cts
create_analysis_view -name view_cts -constraint_mode {cts} -delay_corner {name}
```

```
set_analysis_view -setup {view_cts} -hold {view_cts}
create_ccopt_clock_tree_spec -file xxx.spec
source xxx.spec
ccopt_design -cts
```

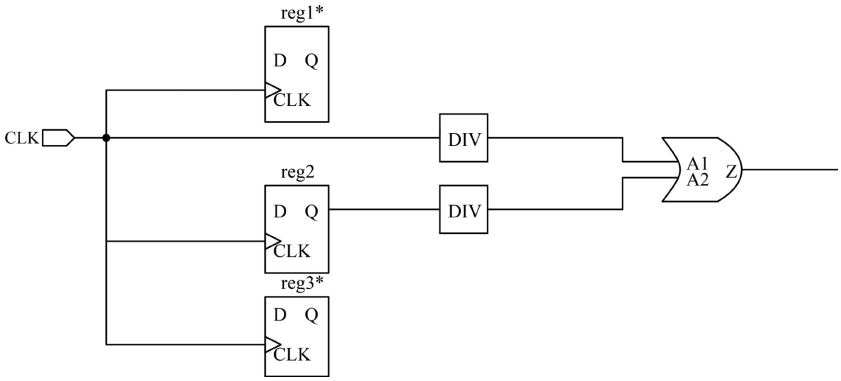

Figure 4. Different branch circuits of the same source clock
图 4. 同源时钟不同分支电路

4. 时钟树结果比较

分别使用传统时钟树综合流程和重新编写 SDC 时序约束文件进行分段长时钟树的方法进行比较。表 1 和表 2 分别是两种方法的时序报告对比，从表中可以看出，分段长时钟数的方法明显适用于复杂时钟结构的设计流程。与传统时钟树综合相比较，setup time 的最大时序违例(WNS)和总的时间裕度(TNS)分别为−0.197 ns 和−16.776 ns。同时，违例路径的数目减少了 85 条，分段长时钟树对关键路径的时序违例有明显的改善。都在可控的一个处理范围内，后续可以通过其他方法将违例修复[11]，相比传统时钟树流程的结果，新方法明显可以加快完成芯片的时序收敛要求。

Table 1. Traditional clock tree comprehensive setup timing reports
表 1. 传统时钟树综合 setup 时序报告

方法	时序统计	Reg2reg
传统时钟树综合	WNS/ns	−2.903
	TNS/ns	−305.460
	Violating Paths	210

Table 2. Segmented clock tree setup timing report table
表 2. 分段长时钟树 setup 时序报告表

方法	时序统计	Reg2reg
分段长时钟树	WNS/ns	−0.197
	TNS/ns	−16.776
	Violating Paths	125

表 3 分别是两种方法在插入缓冲器数目、时钟数的级数、总功耗和总面积的比较。这些参数也可以明显反映出一个时钟树设计的好坏[12]。传统时钟树综合流程通过插入大量的缓冲器单元，使许多路径的

支路与最长路径等长，大量的缓冲器插入不可避免导致时钟树的级数、功耗和面积会增大。从表 2 可以看出，分段长时钟树使插入的缓冲器数目减少了 2676 个，使时钟树的最大级数仅为 8 级。由于缓冲器数目的大量减少，使总功耗降低了 6.2 mW，同时节省了 2729 um^2 的面积，能够有效的解决局部阻塞问题，提高了数字芯片的性能。

Table 3. Comparison of other important parameters under the two methods
表 3. 两种方法下的其他重要参数比较

方法	缓冲器数目	时钟树级数	总功耗(mW)	总面积(um^2)
传统时钟树综合	3257	67	11.3	34,447
分段长时钟树	581	8	5.1	31718

5. 总结

与传统时钟树综合流程相比，本文通过对时钟电路进行分析，编写一个适用于复杂时钟结构的时序约束文件(SDC)，然后再进行分段长时钟树，最终得到一个 setup time 的时序违例小，违例路径减少 85 条，通过插入少量的缓冲器数目，生成一个级数较低的高质量时钟树。高质量的时钟树降低了芯片的总功耗和节省的大量的面积，能够有效解决局部绕线阻塞问题，同时加快了整个芯片的时序收敛进程。面对复杂的时钟结构设计，本文所介绍的分段长时钟树方法，可以减少迭代时间周期，有效解决设计时序无法收敛的难点，能够提高芯片的工作性能，满足设计的需求，该方法是可以适用于芯片设计实践中的。

参考文献

[1] 孙佳佳, 赵庆哲. 在 IC 设计中应用 STA 处理时序问题的方法[J]. 微处理机, 2014, 35(4): 15-18.
[2] Wong, B.P., Mittal, Cao, Y. and Starr, G.W. (2005) Nano-CMOS Circuit and Physical Design. Wiley-IEEE Press.
[3] 刘毅. 复杂时钟分析优化平台[J]. 中国集成电路, 2015, 24(12): 52-55.
[4] 李旭, 周海斌, 王兴家, 等. 嵌入式高性能处理核的时序收敛设计方法[J]. 集成电路应用, 2021, 38(6): 6-7.
[5] 曾晋伟. 基于 Innovus 的复杂时钟结构分析及实现[J]. 电子技术应用, 2020, 46(8): 64-67.
[6] 翟金标, 李建成. 基于 28nm 数字芯片的分步式时钟树综合设计[J]. 中国集成电路, 2022, 31(8): 40-44.
[7] 陈力颖, 翦彦龙, 吕英杰. 基于 28nm 工艺的 CCOpt 技术高效时钟树设计[J]. 天津工业大学学报, 2019, 38(2): 62-67.
[8] 高明. 28 nm 工艺下双核 Cortex-A9 处理器芯片的物理设计[D]: [硕士学位论文]. 南京: 东南大学, 2016.
[9] Gangadharan, S. and Churiwala, S. (2015) Constraining Designs for Synthesis and Timing Analysis: A Practical Guide to Synopsys Design Constraints (SDC). Springer.
[10] 许立明, 李沛杰. ASIC 中时钟 MUX 电路结构时序约束的方法分析[J]. 集成电路应用, 2019, 36(11): 12-15.
[11] 沈潜锋, 龚敏. 基于 MS-ECO 的多模式多端角签收时序修复方法的研究[J]. 电子器件, 2018, 41(2): 479-483.
[12] 史冬霞, 王淑芬. 基于 40 nm 工艺的高效时钟树优化设计[J]. 电子元器件与信息技术, 2023, 7(1): 38-41.