

基于 Innovus 的复杂时钟结构分析及实现

曾晋伟

(深圳市中兴微电子技术有限公司, 四川 成都 610041)

摘要: 在先进工艺节点下, 随着设计规模越来越大, 时钟频率越来越高以及时钟结构越来越复杂, 最终整个设计收敛对于时钟质量的依赖越来越明显。针对类似多输入动态 mux 复杂时钟、IP 模块多内部输出时钟等复杂的时钟结构, 采用分析时钟框图及基于 Innovus 工具从网表中提取时钟结构的分析方式进行时钟结构上的详细梳理, 提出针对时钟结构分析及 clock spec 的优化方法。同时在一个超大规模的 16 nm top design 上基于优化后的 clock spec 进行 CTS, 并结合 multi-tap 的 clock tree 做法, 从得到的结果可以发现在 run time、clock latency 等方面都有较大的提升, 能够满足项目要求的时钟长度等要求, 有效避免 block 接口的时序冲突。

关键词: Innovus; 物理实现; 时钟树; multi-tap CTS

中图分类号: TN402

文献标识码: A

DOI: 10.16157/j.issn.0258-7998.209803

中文引用格式: 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.

英文引用格式: Zeng Jinwei. Complicated clock structure analysis and implementation with Innovus implementation system[J]. Application of Electronic Technique, 2020, 46(8): 64–67.

Complicated clock structure analysis and implementation with Innovus implementation system

Zeng Jinwei

(Sanechips Technology Co., Ltd., Chengdu 610041, China)

Abstract: In advanced process node, as the design scale becomes larger and larger, the clock frequency becomes higher and the clock structure becomes more and more complicated, it is increasingly found that the closure of the design depends more and more on the clock quality. For complicated clock structures such as multi-input dynamic mux, IP modules with multiple internal output clocks, etc., the clock structure is analyzed, and the clock structure is extracted from the netlist based on the Innovus tool, clock spec will be updated based on these analysis. At the same time, CTS is performed on an ultra-large 16 nm top design based on the optimized clock spec, combined with the multi-tap clock tree methodology. From the results obtained, it can be found that the run time, clock latency and other aspects have been greatly improved. It can meet the requirements such as the clock length required by the project, and effectively avoid the timing conflict of the block interface.

Key words: Innovus; physical design; clock tree; multi-tap CTS

0 引言

随着集成电路工艺进入先进节点(Advanced Node), 以及应用场景的不断增加, 带来芯片设计规模越来越大以及时钟结构更加复杂, 针对时钟结构的分析与时钟的实现也更加困难。就时钟树综合(Clock Tree Synthesis, CTS)而言, 时钟结构复杂程度的增加, 可能会带来公共路径(Common Path)的长度减少, 片上误差(On Chip Variation, OCV)的影响增加, CTS 迭代时间(Turn-Around Time)增加, 以及时钟上功耗增加等问题。因此, 在物理实现中, CTS 变得越来越重要。

在本文中, 借助于 Cadence 公司的自动化布局布线工具 Innovus, 首先探讨了针对复杂时钟结构的时钟如何进行分析, 其次基于分析结果提出时钟实现上可能出现的

问题以及解决方案, 再次, 基于调整进行 CTS 实现, 并与传统 CTS 方案的结果进行对比, 最后对本文进行总结并对结论进行进一步分析。

1 复杂时钟结构概述

1.1 定义

本文中, 复杂时钟结构主要指符合如下条件的时钟:

- (1)同一个 Sink 点存在很多个时钟源头;
- (2)芯片规模很大, 时钟实现可用区域有限, 时钟长度(Clock Latency)要求很高;
- (3)时钟上逻辑关系复杂, 比如时钟树上的逻辑单元和锁相环(PLL)之间对拥塞(Congestion)及时序(Timing)非常敏感, 分频结构较为复杂, 同一时钟下的不同 Sink 点可能存在不同的 Skew 中, 且不一定有时序交互。

(Timing Talk)。

1.2 复杂时钟结构 CTS 实现的挑战

一般情况下,针对复杂时钟结构时钟树,CTS 实现上主要存在如下挑战:

(1) 目标时钟延迟(Clock Latency)要求较短,Common Path 由于时钟结构本身的关系可能天然导致长度较短,OCV 影响较大,各模块接口时序收敛难度较大;

(2) 时钟偏斜(Skew)要求较高,特别是针对例如时钟频率超过 1 GHz 的高频时钟,Skew 很可能超过时钟周期的 10%;

(3) 静态功耗以及动态功耗要求较高,信号电迁移(Signal EM)不能出现有违例(Violation)的点。

1.3 传统 CTS 流程

传统的 CTS 流程,一般来说都是首先读入 CTS 前的数据(DataBase,DB),然后设置 NDR(Non-Default Rule)、CTS 用到的 Cell、CTS target 等信息,创建时钟定义文件(CTS spec);最后利用工具进行时钟树综合。具体的流程图如图 1 所示。

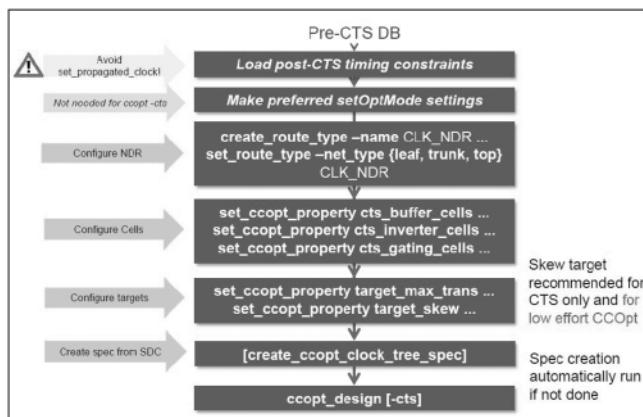


图 1 传统 CTS 流程图

1.3.1 采用传统 CTS 流程存在的问题

传统 CTS 流程自动化程度很高,但是同样也存在一些问题,具体主要有以下几点:

(1) 在进行时钟树平衡(Balance)的时候,通常只会考虑一个单独的慢工艺角(Slow Corner),并且会在整个时钟树的所有位置利用单元插入(Cell Insertion)、单元大小变化(Cell Sizing)以及走线长度的调整(Wire Length Adjustment)等方式来达到预设的时钟树综合目标;

(2) 针对快工艺角(Fast Corner),电阻电容寄生参数(RC Parasitic Parameters)的变化会导致在慢工艺角(Slow Corner)下平衡的时钟树在快工艺角下变得不平衡,时钟偏斜可能会增加,如图 2 所示^[2];

(3) 高频路径可能会有建立时间(Setup Time)和保持时间(Hold Time)之间的冲突,对时序收敛造成风险。

1.3.2 采用传统 CTS 流程实现复杂时钟结构存在的问题

复杂时钟结构如果采用传统的 CTS 流程,主要会面

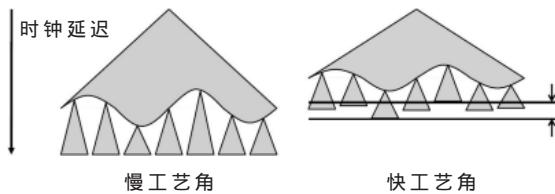


图 2 同一时刻不同工艺角下的时钟偏斜差异

临如下问题:

(1) 由于时钟结构复杂,自动生成 CTS spec 文件的时间可能会非常长,通常可能超过 10 小时,对于迭代来说这个时间是非常不利的;

(2) 自动生成的 CTS spec 文件中,针对不同 Skew Group 下的冲突 Sink,一般不会针对性做特殊处理,最终可能导致时钟长度很长;

(3) 如果设计中存在使时钟需要走在窄沟道(Narrow Channel)或者不友好的路径(Un-friendly Path)的情况,可能导致某些 CTS Cell 放置位置不合理导致 CTS 结果不理想的情况。

2 复杂时钟结构实现方法学

本部分基于一个实际的例子,来描述如何利用 Innovus 工具来实现复杂时钟结构的 CTS。

2.1 设计介绍

本部分采用了一个基于 TSMC 16 nm 工艺的一个芯片顶层来进行说明。整个芯片的大小大约为 22 mm×16 mm,共有超过 60 个模块,最高金属层为 M11,其中 M9 及以下层次可以用于信号走线,最高频率超过 1 GHz,各模块接口时序为同步,仅顶层有超过 26 万个寄存器,其中打拍寄存器超过一半,Signoff 方式为 AOCV(Advanced-OCV)。

Floorplan 大致如图 3 所示,其中灰色部分为 top only 逻辑。

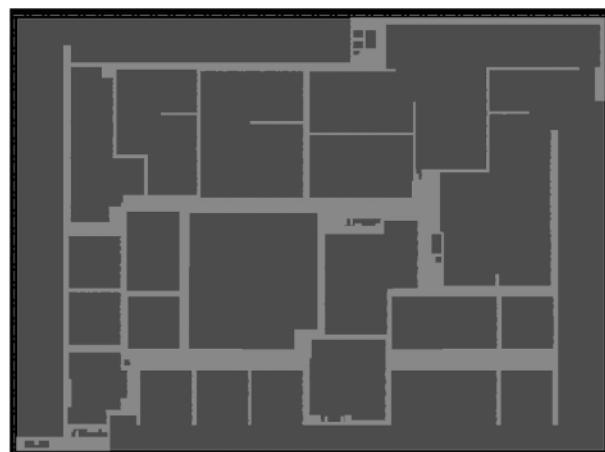


图 3 Floorplan 图

2.2 基于 Innovus 的时钟结构分析

本部分中,基于 Cadence 公司的 Innovus 工具,对时钟结构进行分析,主要的分析方式主要有 Schematic Viewer

和 Ccopt clock tree debugger 两种。

2.2.1 时钟结构分析

时钟结构框图如图 4 所示,以图中 Clock A 为例,可以看到,分频器后定义了 Generated Clock,且 top 和 block 之间接口时序检查的时钟即为 Clock A,在 ICG1 后的寄存器和分频器(DIV2)后的寄存器之间无时序检查。

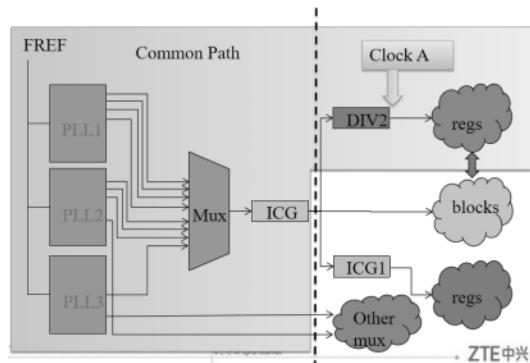


图 4 时钟结构框图

具体的时钟结构中,mux 为 8 输入动态 mux,总共输入 8 个频点,所有到模块的时钟为原始频率,且在 block 内部进行分频,然后用于接口,分频时钟在 top only 也用于部分异步桥逻辑。

物理位置上,PLL1 和 PLL2 距离很近,但和 PLL3 距离较远,且 PLL2 和 PLL3 还会输出其余频点给别的逻辑进行使用。

利用 Innovus 工具,也可以得到 8 输入动态 mux 的具体结构由 100 多个与非、或非以及时钟门控等单元组成,且根据要求,所有的这些时钟逻辑单元需要在实现的时候设置成 dont_touch。

2.2.2 复杂时钟结构实现

针对复杂时钟结构,通常会采用分析时钟结构后,提取出关键信息,利用增加 Region,预摆放时钟 Cell,以及调整 CTS spec,采用 multi-source 方式进行时钟实现等方式来实现。

2.2.2.1 Region 调整

如图 5 所示,上图为增加 region 前,可以看到动态 mux 的时钟逻辑 Cell 被工具放置在距离 PLL 较远的位置,通过设置 region,可以得到下图的效果,将时钟逻辑 Cell 放置到 PLL 边上,保证时钟在绕线的时候不“detour”。

2.2.2.2 spec 调整

基于时钟结构的具体分析,对 Clock A 后的寄存器分为两类,一类为没有二级 ICG 单元的控制,一类为有二级 ICG 单元控制的,且这类还可以再根据 ICG 的不同细化为两类(ICG1,ICG2),如图 6 所示。

以上三类寄存器之间没有相互的 timing talk。

在 spec 中,将整条时钟利用 ICG 作为分隔点的不同拆分成三个:

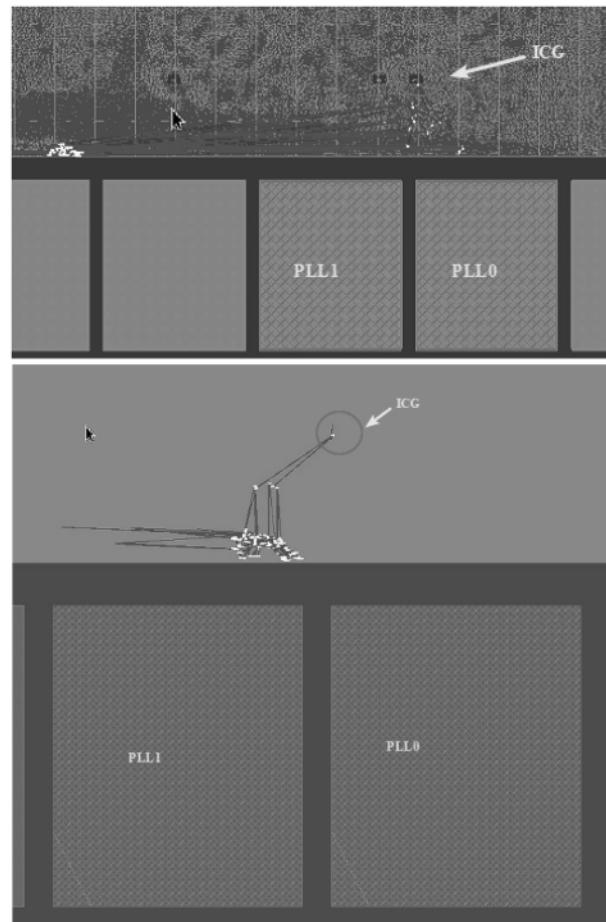


图 5 增加 region 前后时钟 Cell 位置

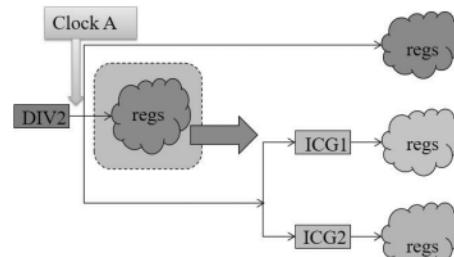


图 6 时钟结构细化

```
create_ccopt_clock_tree -name icg_**** -source ****
ICG/Q -no_skew_group
create_ccopt_clock_tree -name icg1_**** -source ****
ICG/Q -no_skew_group
create_ccopt_clock_tree -name icg2_**** -source ****
ICG/Q -no_skew_group
```

从而,整条 clock tree 最终会被拆分成 4 部分,如图 7 所示。

2.2.2.3 Multi-source clock tree 实现

在本部分,介绍如何利用 Innovus 工具进行 Multi-source clock tree 实现。

Multi-source clock tree 是在传统 CTS 方法的基础上提出的更加高级的 clock 实现方法,可以有效降低时钟

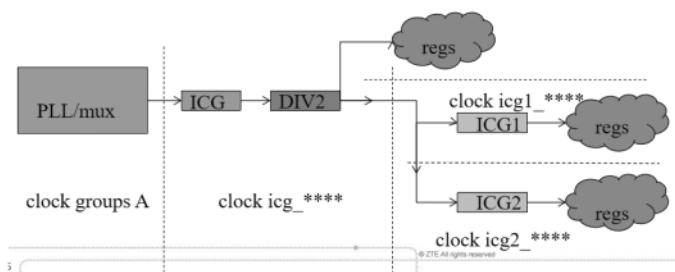


图 7 时钟拆分

长度,从而达到 OCV 影响的减少,具体的示意图及流程图如图 8 所示。

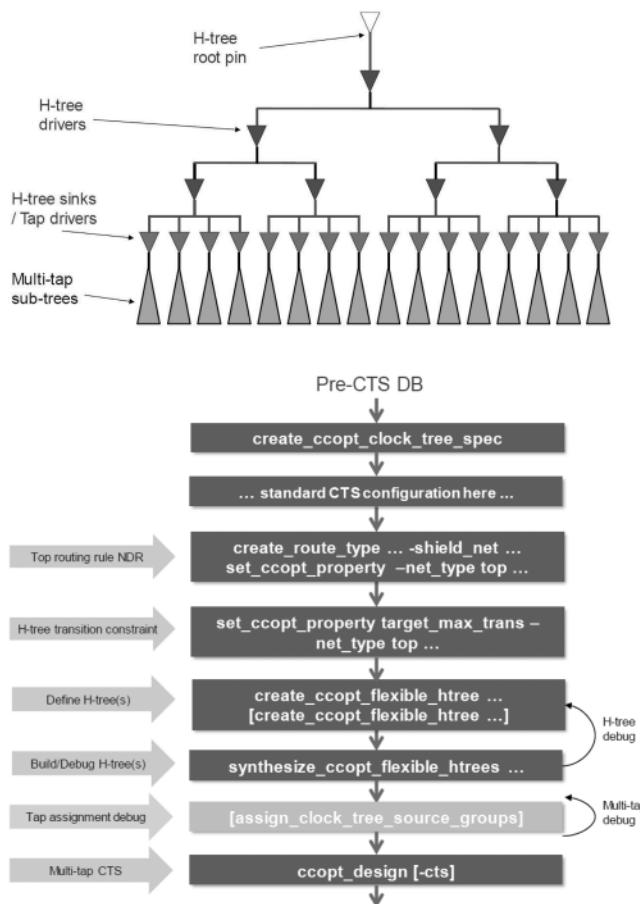


图 8 Multi-Source clock tree 实现

针对图 8 中的 H-tree 部分,为了更好地降低 clock latency,预先用 24 倍的反相器(Inverter)以及高层金属(M10,M11)提前进行预先绕线。但是为了降低天线规则、压降(IR-drop)、EM 等方面的影响,采用大驱动 Inverter 需要做如下处理:

(1)所有的 24 倍 Inverter 周围预留 15 μm 左右的位置用于填 DCAP Cell,降低因为驱动较大可能造成的 IR drop;

(2)所有的 24 倍 Inverter 针对性的加密电源网络,避免出现电源上的 EM 问题;

(3)针对可能出现的天线效应,提前放置二极管 Cell 等进行规避;

(4)Signal EM 方面,采用 NDR 规则走线,增加打孔,以达到规避。

在用 Multi-source CTS 流程中,在执行完 assign_clock_tree_source_groups 后,可以从 log 中看到每一个 tap 点对应的 Cell 分布,通常情况下都是很均匀的分布,且 sink 点距离 tap 点的位置符合预期,如果出现异常点,则需要具体分析该 tap 点后的 sink 异常的原因。

2.3 结果对比

结果对比如表 1 所示。

表 1 普通 CTS 与 Multi-source CTS 对比

	普通 CTS	Multi-source CTS	变化/%
延迟/ns	2.946	1.839	37.58
运行时间/h	22	20	9.09
Clock Inverter Count	42 064	39 311	6.54
Clock Tree Level	208	68	67.31

3 结论

针对复杂结构的时钟树,利用工具来进行时钟结构的识别及分析,可以在 CTS 甚至 pre-CTS 等阶段有效提高效率。

针对时钟结构中的同一时钟下的没有时序检查的寄存器,设置 Skew Group 来调整平衡策略,可以有利于时钟树做短。

采用大驱动的时钟反相器并通过 Multi-source 的方式长 Clock Tree,同样也是一种有效减少时钟长度的方式,并且针对受 OCV 影响较大的设计,这种方式也可以有效减少迭代时间,加速时序收敛。

参考文献

- [1] Cadence Innovus user guide[EB/OL].[2020-05-21].<https://www.cadence.com>.
- [2] Flexible H-tree and multi-tap clock flow in Innovus[EB/OL].[2020-05-21].<https://www.cadence.com>.

(收稿日期:2020-06-22)

作者简介:

曾晋伟(1991-),男,工程师,主要研究方向:数字集成电路物理实现。

版权声明

经作者授权，本论文版权和信息网络传播权归属于《电子技术应用》杂志，凡未经本刊书面同意任何机构、组织和个人不得擅自复印、汇编、翻译和进行信息网络传播。未经本刊书面同意，禁止一切互联网论文资源平台非法上传、收录本论文。

截至目前，本论文已经授权被中国期刊全文数据库（CNKI）、万方数据知识服务平台、中文科技期刊数据库（维普网）、DOAJ、美国《乌利希期刊指南》、JST 日本科技技术振兴机构数据库等数据库全文收录。

对于违反上述禁止行为并违法使用本论文的机构、组织和个人，本刊将采取一切必要法律行动来维护正当权益。

特此声明！

《电子技术应用》编辑部

中国电子信息产业集团有限公司第六研究所