

考虑物理布局布线约束的 快速时序收敛

Timing Explorer

面临挑战

随着集成电路工艺的快速发展，工艺结点已经从45nm下降至28nm乃至20nm。更小的物理尺寸给传统的IC设计带来了巨大的困难和挑战，在规定的窗口内完成时序收敛也越来越困难。工程师在ECO阶段修复时序违反往往要付出很大的时间和精力。新出现的问题和约束给这一任务变得尤为困难。

复杂的收敛性变差

时序收敛的一个很大障碍就是用作sign-off的时序分析引擎和ECO工具的时序优化引擎并不相同。这样的不一致性会带来收敛性变差，造成时序优化方案最终实现上不能达到预期的结果。加上现有的ECO工具无法考量物理因素的影响，当决定缓冲器单元插入位置时会造成布线的变化，带来更多的不确定性。

迭代次数变多

收敛性变差迫使工程师会反复执行ECO操作来试图达到时序收敛。这些迭代不仅仅会耗费大量的时间和资源，而且不能保证最终能得到合理的方案。

物理因素制约

在复杂的芯片设计中，单元物理布局和布线通常都非常拥挤。还有很多系统中包含着多电压域设计，ECO操作需要使用特殊的缓冲器单元，这也使得物理布局布线问题更加复杂。

过多的缓冲器插入

低功耗是目前芯片设计一个重要的目标。过多的缓冲器单元插入不但会增加芯片面积，还会增加芯片的功耗，使其竞争力大大下降。

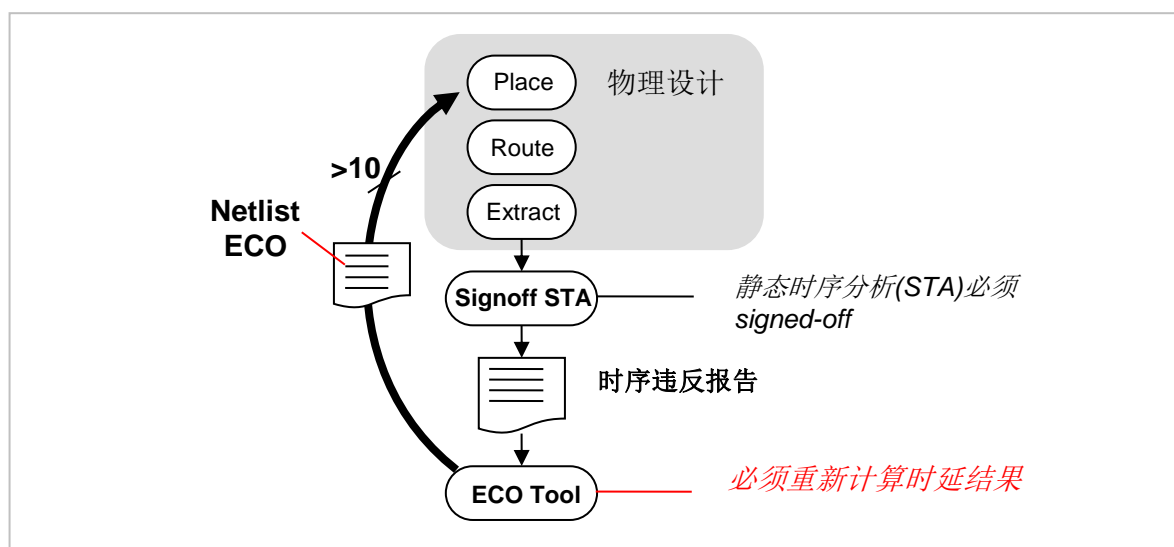


Figure 1. 传统的时序优化ECO流程

因此，面对先进工艺带来的问题和挑战，要找到时序收敛的解决方案，必须同时综合考虑这些问题。

考虑物理布局布线约束的快速时序收敛

解决问题

TimingExplorer工具可以提供最准确高效的MCMC时序ECO解决方案。和目前市场上的ECO工具相比，它不但考虑了物理布局因素，还综合考虑了物理布线带来的影响。其先进的优化算法，灵活的流程控制，能快速实现时序收敛，并且已经通过了20nm先进工艺验证。

快速收敛

如何能快速到达时序收敛是ECO工具需要面对的一个问题。当Sign-off STA工具报告出时序违反，ECO工具需要做出相应的调整，并把修复方案返回物理布局布线工具去实现。但是这中间会产生各种一致性问题，需要进行许多次迭代来得到最终结果。因此在芯片投片生产流程中，ECO时序修复成了关键性的一环。

时序和版图相结合

有两个因素影响结果的一致性：时序计算和物理版图。ECO工具通常都是基于时序违反报告和部分时序图来进行时序计算。当修复了一个时序违反问题，也许会带来另外一个时序违反问题。计算精度也无法保证与Sign-off的STA工具相一致。

物理版图考虑则是值得注意的另一方面。仅仅基于网表连接关系的ECO改变会在物理布局和布线上带来意想不到的影响。这些影响也会影响到时序计算，产生一致性问题。

ECO优化工具

目前有三种ECO优化方案来帮助实现时序收敛。第一种情况，用户使用的是script脚本，由于基于少数部分时序违反报告，它仅可以对时延值进行粗略的估算，采用的也是简单算法来修复时序违反，而且无法预估对物理版图的影响。因此这种方法在数值精度和物理方面的一致性都存在问题。

第二种情况是基于STA工具来进行ECO优化。虽然时延值精度得到了保证，但是运算速度随着工作场景增多会显著变慢，而且STA引擎缺乏对物理信息的了解，最终的优化方案可能无法在版图上实施。

第三种方案是基于物理布局布线工具进行ECO优化。它有着自己的时序计算方法，与Sign-off的STA工具并不相同，因此数值精度上大打折扣。使用起来需要设置很大的余量来确保不破坏时序约束，因此会加入比较多的缓冲器单元，导致功耗和拥挤度问题恶化。

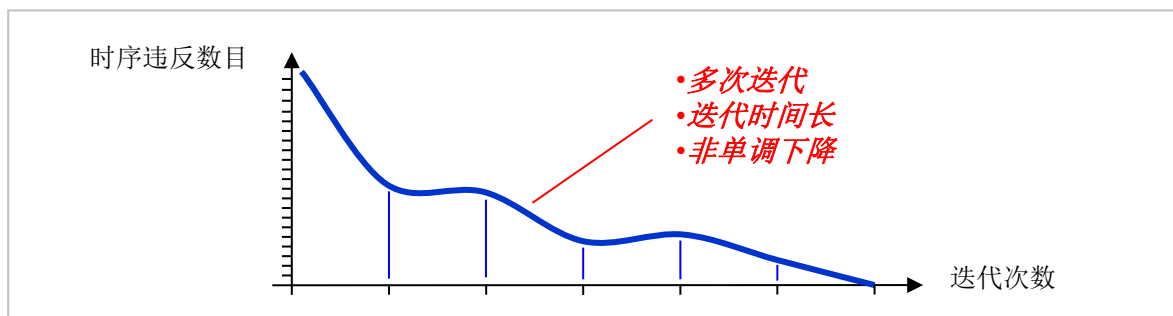


Figure 2. 传统ECO流程的时序违反数目变化

上图显示的是传统ECO流程中修复时序违反的迭代情况。不同工作场景的时序约束，不同的优化目标在不同的阶段加以考量。因此迭代不但次数非常多，而且时间非常长，时序违反还会出现非单调下降，甚至反复的现象。它占用了芯片投片生产之前宝贵的时间安排。

考虑物理布局布线约束的快速时序收敛

考虑物理布线的ECO操作

在复杂的SoC设计当中，并不是所有物理布局拥挤区域同时也是物理布线拥挤区域。因此，仅仅基于物理布局的ECO优化是不够的，不能保证和版图的一致性。TimingExplorer内置了物理引擎，它可以有效的考虑物理布线约束：

- 自动分析并发现布线拥挤区域，避免在这些区域上进行ECO操作
- 修复Max-transition时序违反时候，可以遵循线网走线拓扑，减少对物理布线的影响

改善流程

下图给出了TimingExplorer工具提供的改善流程。它一方面直接使用Sign-off STA的时序数据输入，保证了数据精度的一致性。另一方面从物理设计中读取物理布局布线约束条件，在时序优化过程中考虑了对版图的影响，保证了优化方案能够顺利实施，也保证了和物理布局布线工具良好的一致性。最终结果会同时输出网表ECO变化和物理位置ECO变化。这样经过两到三次少数轮迭代即可完成时序收敛。

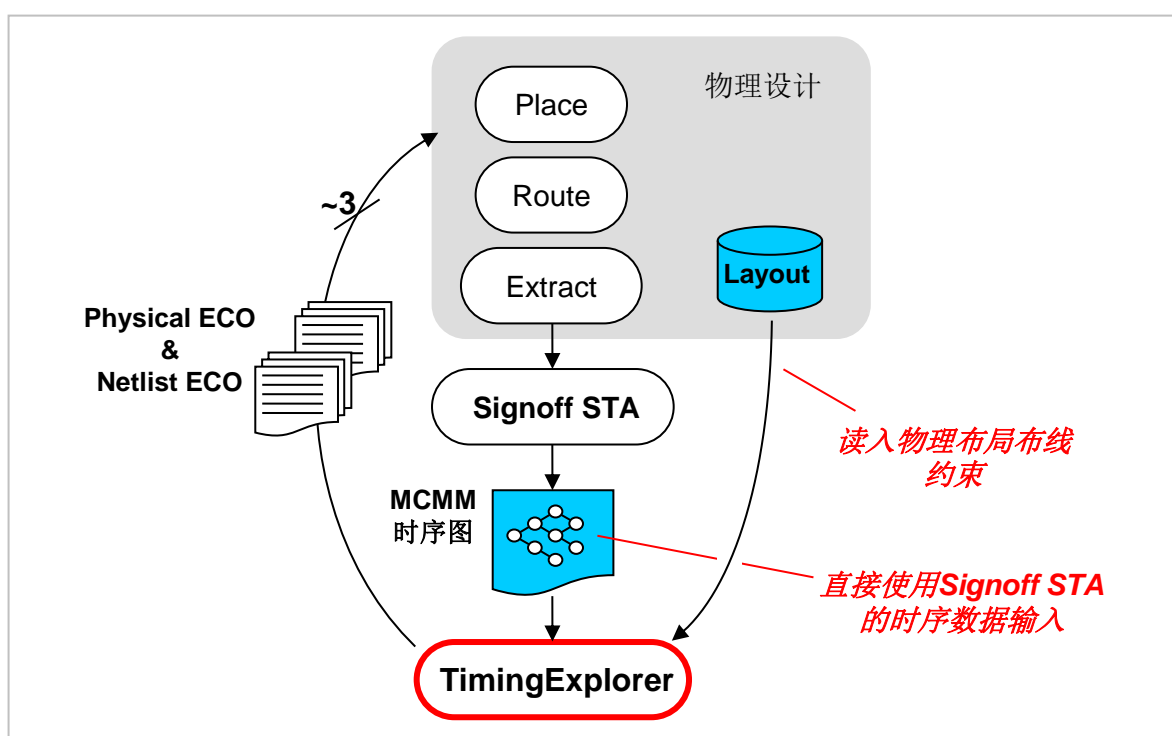


Figure 3. 考虑物理布局布线约束的时序优化ECO流程

时序收敛是一个非常复杂的过程，需要同时考虑多个时序目标，例如Setup, Hold, 以及Max-transition等。同时必须考虑芯片设计在不同工作场景和不同工作模式下的时序条件约束，还有芯片面积和功耗等优化目标。

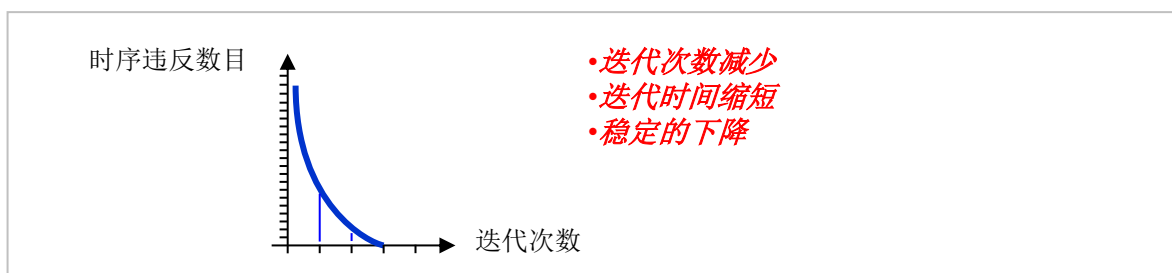


Figure 4. 改进的流程可以实现快速时序收敛

考虑物理布局布线约束的快速时序收敛

上图给出了应用TimingExplorer来进行时序优化的情况。可以看到时序收敛的迭代次数得到了有效的减少，每次迭代花费的时间也得到了降低。它采用先进的算法和流程控制，同时考虑多个工作场景下的时序约束，经过两至三轮迭代，所有的时序违反都已经被基本修复干净。由于内部引擎充分考虑到物理布局与布线影响，最终的优化方案与STA和PR工具也保持了高度的一致性。这大大缩短了芯片设计在时序收敛环节花费的时间，保证了产品能够按时上市。

处理物理问题

支持多电压域

下图显示的是在包含两个不同电压域的设计中修复Max-transition时序违反。在这个例子中，线网的驱动端和接收端单元位于Always-on电压域，而线网走线通过了On/Off电压域。为了修复Max-transition时序违反，ECO工具必须选用特殊的缓冲器单元，位置选择沿着布线方向，并且保存在正确的层次结构中。

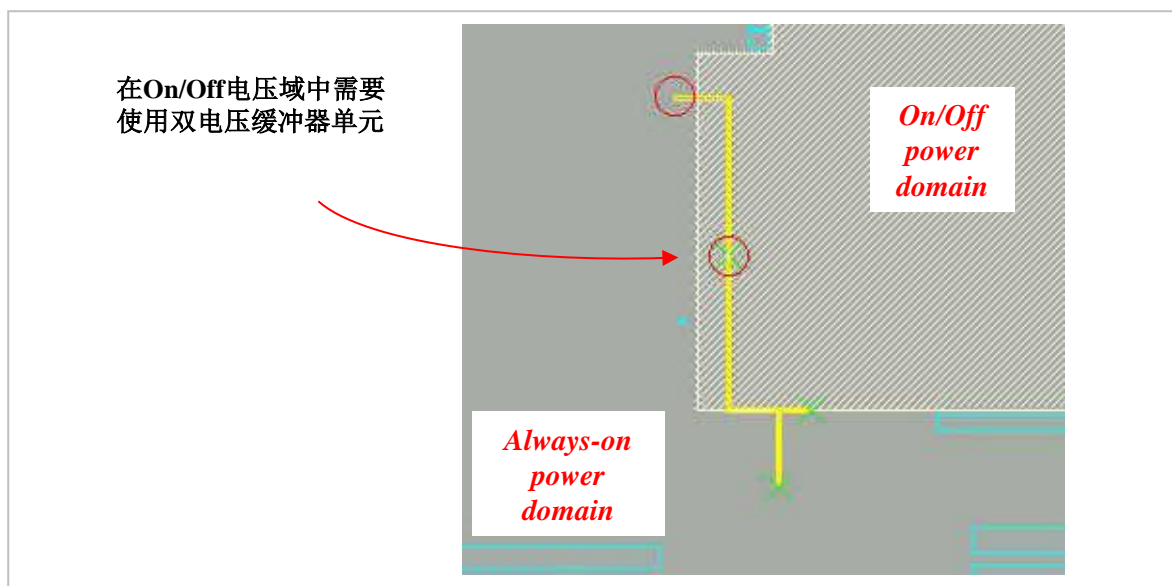


Figure 5. 多电压域下单Max-Transition时序修复

最小数目的缓冲器单元插入

下图给出了一个修复Hold时序违反的例子。在这个电路结构中，分别存在着Hold时序关键路径和Setup时序关键路径。在修复Hold时序的同时，必须遵守Setup时序的约束。如果在接收的寄存器端插入缓冲器单元来修复，就会破坏Setup时序，而且插入的缓冲器单元数目多，会增加系统的功耗。TimingExplorer工具会根据电路结构特点，自动分析并寻找合适的高效率修复点，既保证了不破坏Setup时序约束，又减少了缓冲器单元的数目，降低了系统功耗。

考虑物理布局布线约束的快速时序收敛

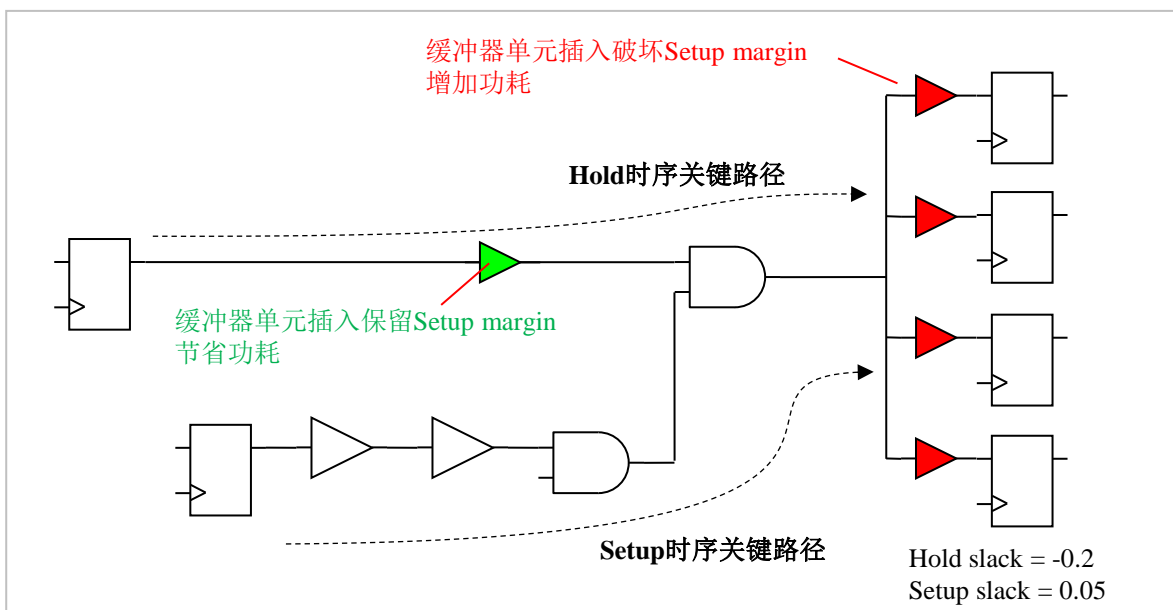


Figure 6. 有效的Hold时序修复：遵循Setup约束并且节省功耗

调整时钟网络

当设计流程到达ECO环节，所有时钟树都已经被CTS工具综合过了。但是如果时钟树综合的质量不高，会带来Setup及Hold时序收敛的麻烦。有些设计使用第三方的IP模块，可能会引起比较严重的时钟偏差问题，但是却不能修改IP模块内部的时钟设计。

下图显示了设计中同步单元和内存IP之间存在着较大的时钟偏差，从而导致Setup时序很差，出现很多时序违反。而这种时序违反可以通过在内存IP模块的时钟端插入缓冲器单元来修复。如果在普通的数据路径上进行时序修复，将会引入大量的缓冲器单元，占用系统过多的资源。

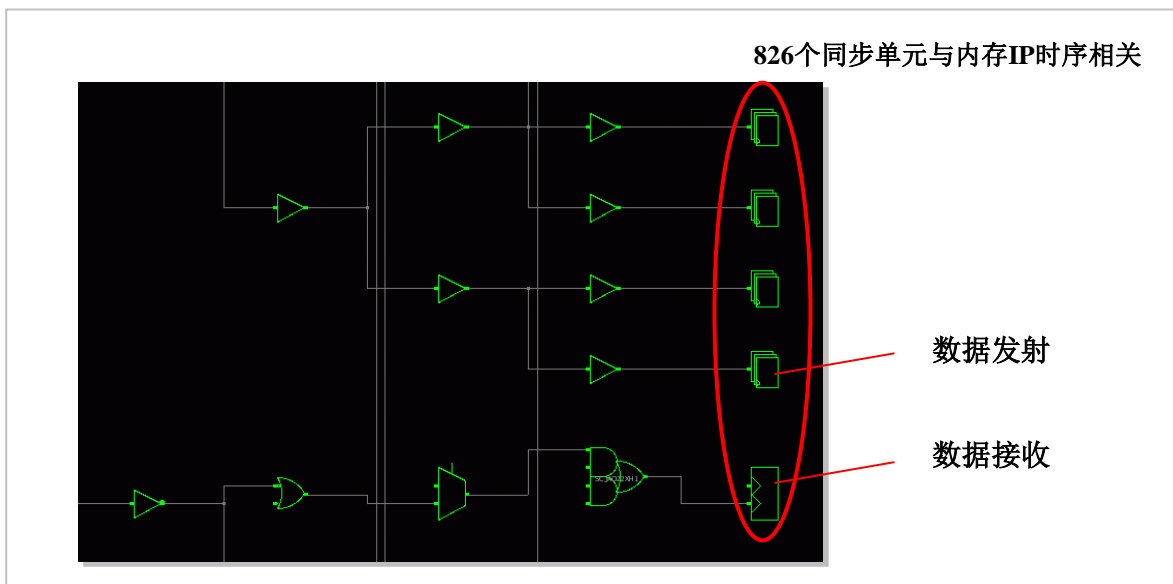


Figure 7.时钟偏差影响着时序结果

考虑物理布局布线约束的快速时序收敛

案例分析

高效的时序修复

下图显示了一个三百万示例的设计，12个工作场景，运用TimingExplorer工具进行时序优化，经过三次ECO迭代就成功的修复了几乎所有Hold时序违反，而传统的流程需要经过9次迭代，每次迭代都需要三天时间。

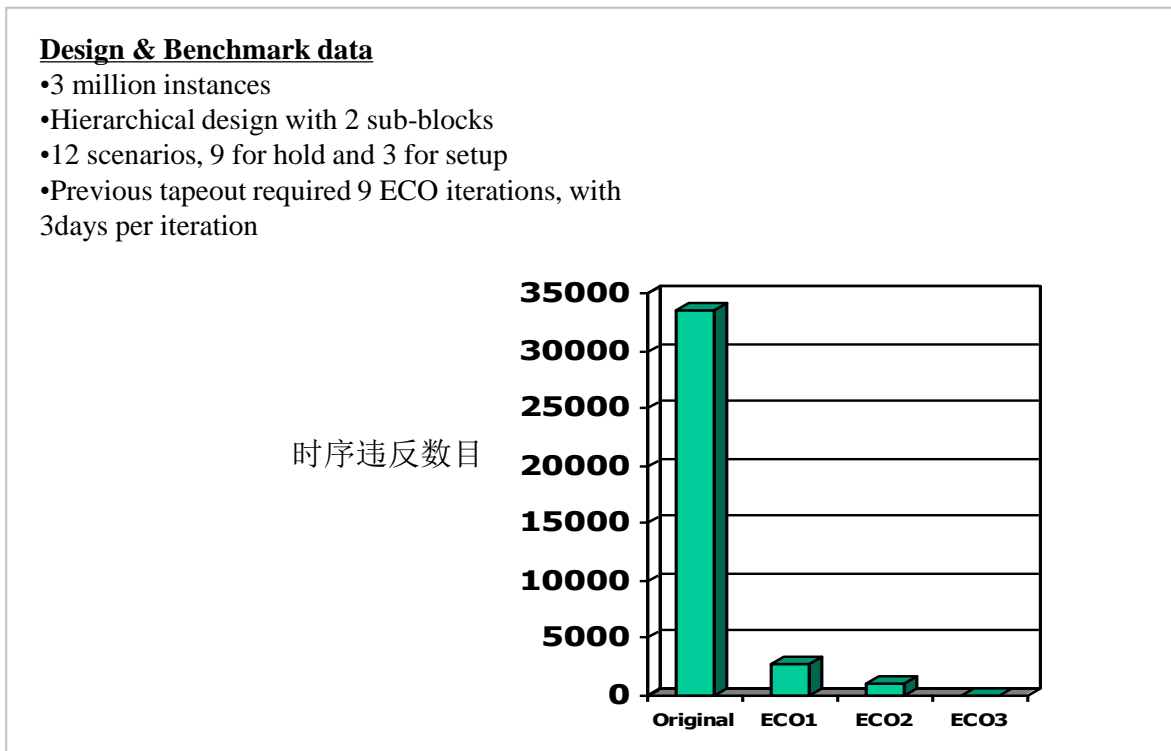


Figure 8. TimingExplorer的Hold时序修复结果

快速的运行时间

对于一个五百万示例的设计，下面表格显示了TimingExplorer和传统基于STA的ECO工具的时序优化时间。可以看出运行时间得到了近乎十倍的加速，而且得到的结果更好。

Design & Benchmark data

- 5 million instances total
- 5 scenarios; 12 for hold and 3 for setup

	Runtime (HRS)	Buffer count	Setup						Hold					
			Vio# Pre	Vio# Post	WNS Pre	WNS Post	TNS Pre	TNS Post	Vio# Pre	Vio# Post	WNS Pre	WNS Post	TNS Pre	TNS Post
Signoff STA-based ECO	36	26663	51	51	-0.12	-0.12	-2.3	-2.3	25259	5121	-0.82	-0.28	-13476	-245
Timing Explorer	3.6	25779	51	51	-0.12	-0.12	-2.3	-2.3	25259	212	-0.82	-0.12	-13476	-21.3

Figure 9. TimingExplorer和基于Signoff STA的ECO工具的Hold时序修复对比

Vio# = Number of violations
 Pre = Pre-Optimization
 Post = Post-Optimization
 WNS = Worst Negative Slack
 TNS = Total Negative Slack

考虑物理布局布线约束的快速时序收敛

多电压域处理

下面的表格显示了针对多电压域设计，TimingExplorer能够修复max-transition时序违反。传统的布局布线工具无法处理多电压域的情况。

Design & Benchmark data

Process	28 nm
Voltage area count	5
Instance count	4.5M
Gen signoff STA data time	23 min
Design setup time	18 min
Timing fix time	75 min
Total runtime	116 min

	Violations	WNS	TNS	Buffer Count	Dual-rail Buff Count	Gate Sized
Pre-ECO	213386	-2.14	-14356.8	--	--	--
ECO pass1	148	-1.43	-132.6	48217	3342	6216

↑ 剩余的时序违反是由于物理布局空间不足所导致

WNS = Worst Negative Slack
TNS = Total Negative Slack

Figure 10. 多电压域设计下单Max Transition时序修复

显著缩短迭代时间

正是因为TimingExplorer具有上述优势和特点，它可以将时序收敛环节压缩在一天之内完成，这对于芯片投片前，紧张的时间安排来说意义重大，可以保证芯片按时投片和产品上市时间。

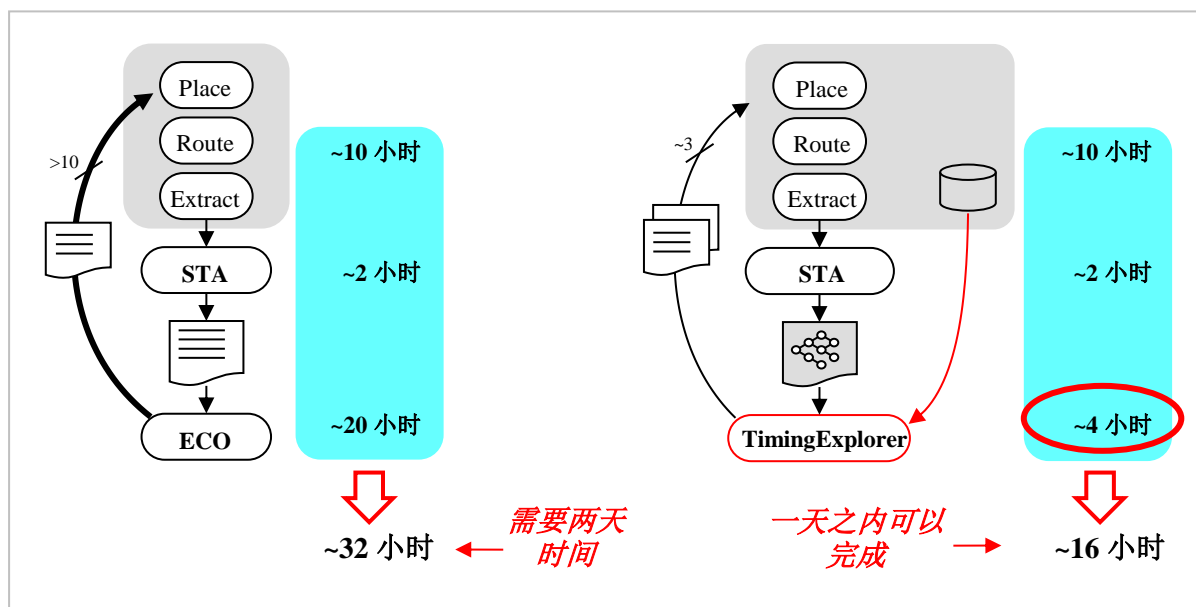


Figure 11. 改善的流程大大缩短了迭代时间

结论

随着集成电路设计进入到20nm先进工艺条件，时序ECO工具必须和sign-off STA引擎保持良好的一致性，并且能够充分考虑物理布局和物理布线因素的约束。当复杂芯片设计包含多个工艺角和多个工作模式的时候，工具应该综合考虑MCOMM约束条件，完成Setup/Hold/Max-transition等多目标的时序优化，进一步提高系统性能。

TimingExplorer工具出色地提供了上述功能，已经成功应用在多颗28nm和20nm等先进工艺条件下的芯片设计中。多线程的架构设计，考虑物理布局布线约束的先进算法，可以快速处理包含多电压域的层次化的复杂设计，优化方案和sign-off STA工具保持了非常好的一致性，可以显著的缩短ECO迭代周期，加速芯片上市时间。