



为 FPGA 设计的时序收敛 未雨绸缪

使用 FPGA 来实现设计的朋友们一定知道时序收敛有多重要，当然也一定清楚时序收敛有时会有多难达到。无法完成时序收敛会影响到方方面面，比如：

- 您的产品上市时间会延期，因为您的团队还在纠结于无法完成设计的时序目标。从研发和市场的角度来说，这大大增加了时间成本和机会成本。
- 有的时候为了达到收敛，必须使用更大的器件或者更快的速度等级。但是这些额外的花费将影响您产品的利润，特别是大规模量产的时候。
- 通常在项目较晚阶段会有新增加额外逻辑功能的需求。这会提高逻辑的密度，对时序产生不好的影响。即便是一小部分的逻辑发生改变，曾经达到时序目标的设计在这个时候也可能会失败。



网站: www.plunify.com/cn

Email: tellus@plunify.com

一般情况下，厂商工具在设计编译时表现优秀，但有时也力不从心。


• 编译选项

编译工具有着大量的选项，因为编译算法有时需要人工帮助来决定某个设计编译的最佳方案。即便是用厂商提供的模板来设定编译的这些选项，也很有可能无法达到您需要的结果。大量的编译选项让设计师在决定使用哪一个的时候十分为难。更有甚者，有时候一套编译选项在开发周期的某一个阶段对设计有效，但在以后的设计迭代中就不好用了，这无疑是雪上加霜。

• 摩尔定律

摩尔定律在一代又一代逻辑密度的增加中仍然有效，但是对过去几十年中一直照例增加的最大频率 (fmax) 已经不再有效了。这意味着编译的时间会增加，因为处理器的速度没有增加，FPGA 的逻辑容量反而持续在增加。厂商编译工具也在不断变得更加复杂来应对这个问题，但是这将是一场艰难的战役。当然，多线程和并行处理是一种手段，但是并不是所有的编译算法都会轻易地支持并行。





为什么对于厂商 FPGA 编译工具来说，时序收敛如此困难？

“在实际应用中这是十分不切实际的方法。因为即便逻辑布线复杂度 N 仅有 20，这也会产生 $20!$ 种，或大概 10^{18} 种可能的布线需要分析。”

- 从理论计算机科学的角度来说，综合，布局布线的优化属于一类叫做 **NP-完全** (NP-complete) 的算法。这是最难解决的问题类别之一，因为您必须筛选海量的备选方案来找到最佳的解决方法。
 - 算法的成本或是计算的复杂度，都取决于完成一个含有 N 个元素集合的解决方案所需要的计算量。这个数据可以帮助我们粗略的估算算法解决问题所需要的实际时长。多项式完全 (P-完全) 的问题的复杂度大概为 N^X - 一个简单的多项式。在这里， N 是可变化的元素数量， X 是一个常数。对于一个足够大的 N ，一个 NP-完全算法的复杂度比 P-完全算法的复杂度大得多。它们有非确定性的多项式函数，比如 $N!$ (阶乘) 或者 X^N (指数) 可以在 N 增大的情况下快速变大。对于 FPGA 编译，找到最佳解决方案的复杂度是 $N!$ 。
 - 另外，如果谁可以找到一个把 NP-完全转化成 P-完全的方法，那他就成了亿万富翁了，因为这是一个 **价值连城的难题**。
 - NP-完全问题的一个经典案例就是销售员出差问题。一个销售员必须搭乘航班去 N 个不同的城市，如果想要花费最少的时间，应该用什么顺序把这些城市都去一遍？这将会有 $N!$ 个不同的备选方案，因为有 $N!$ 种不同的路线把这些城市都去一遍。为了找到最佳方案，您需要计算每一条线路的时间花费然后选择最短的那条。在实际生活中，这是完全不可能的，因为即使 N 等于 20，那也就意味着有 $20!$ 或者大约 10^{18} 种线路要分析。那么更实际一点的做法是使用启发式的方法来更有效率地找到较好的线路，而不是用蛮力来寻找最佳线路。启发式方法的弊端是使用者或许并不知道自己离最佳方案有多近。
- 想象一个由 100 个 FPGA 查询表组成的 10 乘以 10 的数组。想找到基于时序考虑的最佳布局，您必须检查数组内每一个查询表的所有可能布局，因为并没有可以直接计算出最佳方案的办法。由此产生的搜索空间将达到 100 的阶乘($100! = 100 \times 99 \times 98 \times 97 \times \dots \times 5 \times 4 \times 3 \times 2 \times 1$ 或约等于 10^{157})。通过筛选每一种布局排列来寻找最佳布局将遥遥无期，即便使用世界上每一台计算机，所需的时间也都会超过宇宙现有的年龄！当今 FPGA 有着上万的查询表，所以利用这种搜索空间的方法来搜索根本是天方夜谭。除此之外，再考虑到综合和布线的搜索空间，你就会明白这种蛮力的解决方案绝对行不通。

- 既然找到设计的最佳方案已经行不通，FPGA 厂商必须为设计编译来部署一套启发式的算法，提供优秀结果的同时不能占用太长时间运行。对于最大的 FPGA 或是特别困难的设计，厂商的软件团队会尽力提供一个合理的解决方案，一般也要求小于 12 小时的计算时间。在争取可能的最佳时序得分的同时，把计算限制在一个合理的时间范围内，这是一种现实的折衷。然而，这种方法有时候会危害到时序收敛。每一年，厂商软件开发团队的首要目标之一，就是提升他们编译工具的时序收敛能力。考虑到这个难题的性质，在最近二十多年，这一直都是他们的首要目标，并且在未来相当长的一段时间里也将如此。

无法完成时序收敛应该怎么办？

- 改变设计，减少逻辑数量，重构时钟树 (clocking trees) 或者采用流水线 (pipelining)
- 减少设计的功能
- 采用更大的器件或者更快的速度等级
- 做一些布局规划或者增加/改变约束来影响编译
- 直接发布性能较低的产品
- 调整编译选项，希望可以找到正确的设置来收敛时序
- 从厂商的 FAE 那里寻求帮助
- 使用 Plunify 的 InTime



Plunify 的 InTime 时序收敛工具如何工作？

- InTime 是 Xilinx 和 Intel(Altera) FPGA 开发环境下的一个辅助软件。
- InTime 通过使用厂商编译工具运行一系列的并行编译，来把机器学习应用到过程中。这种智能手段，将专注于一套优化的编译工具选项，并且这些选项只针对您的设计。
- InTime 所找到的改善后的编译选项，将像一个模板一样，应用到后续厂商工具的使用中。所以综合，布局布线将更有可能达到您的时序要求。
- InTime 不需要对 RTL 进行任何人工的更改和干预。
- InTime 在您系统防火墙的内部也可以运行，并且可以无障碍地使用您已经购买的厂商工具。它支持本地安装，使用情况完全在您的掌控之中。
- 一旦初始 RTL 创建之后，您可以在开发周期内的任何一个时间点使用 InTime。
- 使用 InTime，您不必精通所有的编译工具的选项。您可以更加专注于设计本身。
- 您使用 InTime 的次数越多，它就与拥有越完备的数据库。这个数据库永远安全地在您系统的防火墙内部，它会在您以后的设计中更加高效地发挥作用。
- InTime 尽管不能完全保证为所有设计达到时序收敛，但是它几乎总能提升一个设计的时序得分。
- InTime 不仅可以帮您达到时序目标，也可以帮您节省时间和花费。

InTime 已经成功地为上世界上很多 FPGA 设计师服务过!

- 无论公司大小, 也无论何种 FPGA 设计, InTime 都成功地解决了时序问题。 [点击这里](#) 看看客户们是如何评价的。
- Plunify 是 [Intel](#) 和 [Xilinx](#) 的官方合作伙伴。

免费试用 InTime!

- 申请 InTime 的免费试用: <http://www.plunify.com/cn/free-evaluation/>
- 其他相关信息: <http://www.plunify.com/cn/product>
- 如果您有更多疑问或者有意向购买, 请发送邮件至 tellus@plunify.com 联系我们

