PLUNIFY

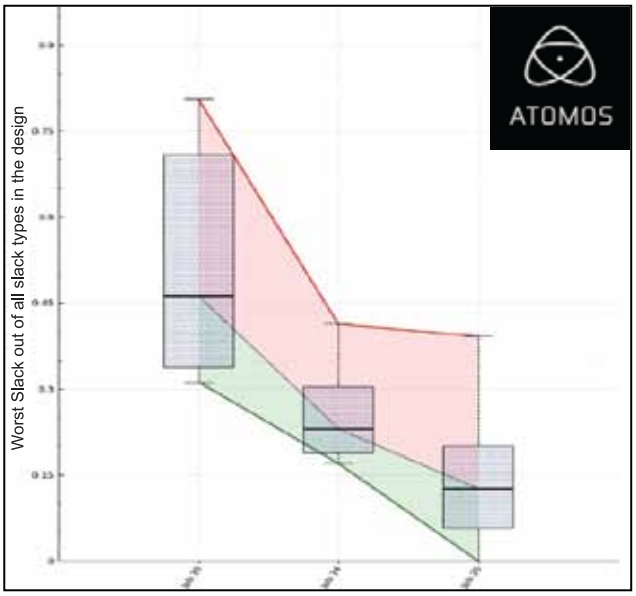
在亚马逊 EC2 云端使用 Xilinx工具和 InTime 优化设计

摘要

这篇白皮书阐明了InTime和Xilinx软件是如何通过调整编译参数以及运行并行编译来优化FPGA时序性能的。InTime通过机器学习来决定一个FPGA设计的综合和布局布线的最佳配置组合。通过和计算服务器的连接，InTime迅速地优化时序，同时也解决了用户流程自动化所面临的限制。

介绍

传统时序优化的方法和训练方向主要集中在检查和改善RTL代码或是时序约束。尽管这种方法行之有效，但在实战中因为技术和商业方面的限制，很多更改都不可能真正的执行下去。比如某些对设计比较大的修改，可能会让产品的发布日期承担滞后的风险。当下盛行可重复使用的设计模块，设计中经常会出现不能轻易更改的第三方IP核。最糟糕情况的解决方案（ “worst-case scenario” solution）也不过就是把目标器件升级到一个更大的或是有更快速度等级的版本，尽管它们都将带来一笔不小的花费。



万幸的是，当今FPGA工具（比如Xilinx的Vivado）都有很多开关和设置选项来帮助时序收敛。InTime的方法，就是通过调整FPGA工具的编译过程来解决用户的时序问题和其他性能问题。Xilinx 的ISE和Vivado软件包含了很多综合和布局布线的参数，每一项都至少有两个值可以直接影响最终结果。InTime帮助设计师深入挖掘这些工具的性能，以达到要求的结果。

上图是一个客户案例，X轴代表采用不同综合和布局布线的编译批次，Y轴显示的是失败最差余量（Failing Worst Slack, 0代表时序通过）的绝对值，单位为纳秒。在这里您可以清楚地看到InTime把失败最差余量从-0.45ns (-450ps) 减少到了0ns，仅通过调整编译参数就达到了时序目标，而且对设计没有任何变动。

很多情况下，用户经常使用默认的综合和布局布线的参数。因为不确定会有什么后果，很少有人会尝试改变这些参数。再加上很多参数是相互联动的，多个参数如果被同时设置错误会让时序变得更加糟糕；所以调试参数这项任务，就变得更加举步维艰。

了解InTime的流程

InTime 使用机器学习来探索FPGA编译过程中不同的参数设置。下面要说明的技术重点关注在性能达到峰值之前，生成足够的数据点。

这里有一个非常关键的概念，叫做“配方”。InTime的优化技术被归结成不同的配方，配方又被分类成“Learning”配方和“Last Mile”配方。

Learning 配方
拥有的数据越多，产生的结果越好。

Last Mile 配方
“父版本”的结果越好，它的结果就越好。更多数据没有直接帮助。

分类的原因是因为编译是一种高强度的计算过程。获得新数据的运行时间成本很高（一般人耐心也很有限），所以每个配方不能无限地运行，必须要根据结果改善的情况来限制Learning的运行数。一旦结果达到稳定（由花费时间和结果改善所决定的投资回报率逐渐减少），用户就会切换到Last Mile配方。Last Mile 配方采用一种高度随机的技巧，设计离目标性能越近，这个配方就工作地越好。例如，如果把目前取得的最好结果当作一个参考，Last Mile配方会随机对不同的逻辑单元进行布局。

优化设计的步骤

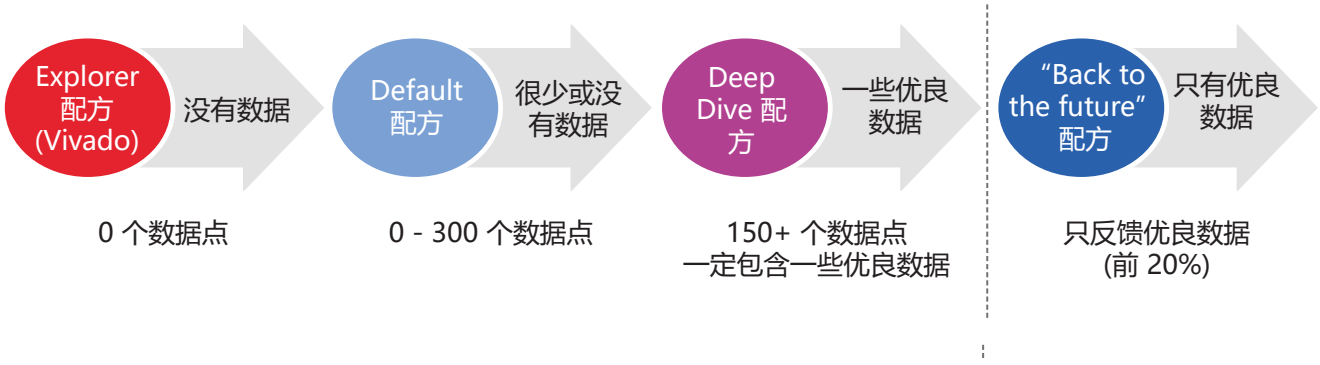
- **步骤0：设计建模**

为了减少集中在一个结果所需要的时间，InTime自带一个包含元数据的数据库。我们在长时间内在不同设计上测试，逐步归纳出在各种设计上用什么参数合适，然后把这些经验总结成了这个数据库。这样做的目标就是缩小需要调节参数的范围，只选取重要的，对某个设计最有效的参数进行调节。

- **步骤一：生成数据**

在这一步，InTime在每一轮的执行中，生成编译参数（也被称作“策略”）。设计师应该对每一轮进行配置，运行10到30个编译。有些配方会比其他的配方更合适，这取决于所得数据点（编译结果）的数量。

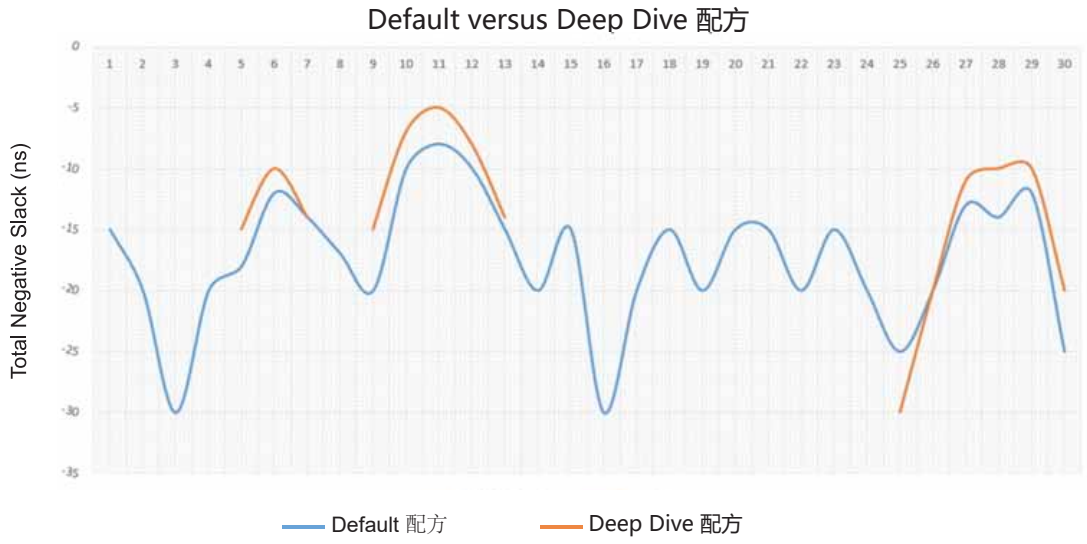
学习和分析只出现在每一轮的结尾或是下一轮的开始。作为一种指导，InTime需要在3到5轮内分析100个数据点来达到一个本地最优数据。



如果结果表示没有明显的改善，我们就需要运行更多的编译，因为这个配方还没有得到本地最优数据。然而，如果结果显著改善（和原始结果比较）并且改善已经不再继续，接下来就要更换配方了（参见Deep Dive配方）。

- **步骤二：使用“Deep Dive”配方**

一旦我们获得了几个优良的结果，或者结果的改善已经开始减缓，就要准备使用“Deep Dive”配方了。这个配方检查当前的结果，然后对本地最优数据与其周围的数据点做一个深入的分析；与之前的配方相比，在更短的时间内将结果改善了10%。当然，没有之前配方的结果，Deep Dive也不会这么有效。



- **步骤三：Auto Placement配方或Extra Optimization 配方**

最终，Last Mile配方会使用Vivado的设置，这些设置伪随机，并且对代码变化高度敏感。根据设计的具体情况，Last Mile配方可以生成仅仅9个编译或者多达100个编译。比如，在Vivado，随机的Placement Exploration配方可以轻易地生成100个编译，然而Extra Optimization则被限制在9个编译。

云端运行InTime和Vivado

您也可以在亚马逊网络服务（AWS）上面运行InTime来减少总共的运行时间，达到时序目标。通过把您的并行运行数加倍，您可以把您优化所需的时间减半。

InTime和Xilinx结为合作伙伴，为亚马逊机器映像（AMI）提供所有的预装软件许可。这可以让您更快的开始一个实例，不用任何安装就可以在云端运行您的FPGA项目。

结论

选择正确的综合和布局布线的参数一种十分强大的技能，它可以帮您达到设计性能目标，从FPGA工具中（如Vivado）获得的最大的受益。然而，把每一组参数都尝试一遍是完全不可能的。如下图所示，快速地聚焦到正确的参数组合，可以产生极好的结果（总负余量从-3000ns到-3ns）。使用云端服务也可以减少达到理想结果的总耗时。