

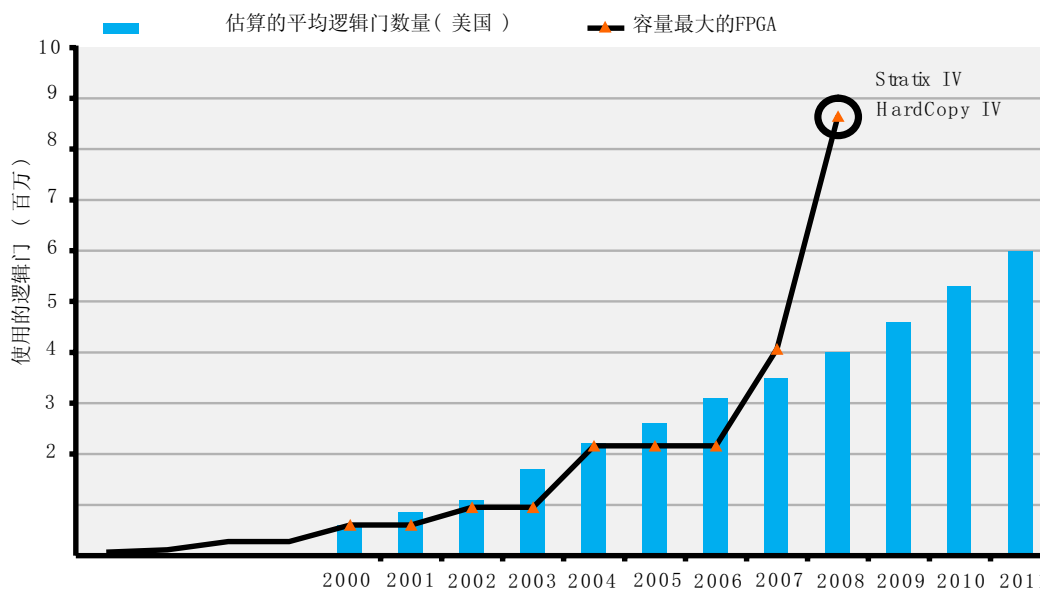
采用 Quartus II 渐进式编译技术提高效能

引言

设计人员开发的 FPGA 逻辑密度和性能在不断提高，面临的产品及时面市压力也越来越大。计算能力的提高速度不足以维持综合、布局布线编译时间不变。例如，在过去 10 年中，Altera® 器件逻辑单元数量增长了 35 倍，存储器位数增长了 100 倍以上，而计算能力仅增长了 10 倍。大型 FPGA 中容量更大、性能更好的设计导致编译时间越来越长。

当今的 FPGA 设计人员还需要应对编译时间问题，解决和以前 ASIC 设计类似的时序逼近问题。2008 年，Altera 密度最大的 Stratix® IV 和 HardCopy® IV 设计的逻辑门数量平均达到 ASIC 设计的两倍，如图 1 所示。如果没有 Altera 在缩短编译时间上的努力，这些大型设计在每一次设计改动时，可能需要一整天的时间进行编译，导致效率低下，浪费设计时间。

图 1. FPGA 超过了 ASIC 设计的平均规模 (1)



注释:

(1) 数据基于 artner Dataquest 报告，11/21/07。

获得成功的关键是在 FPGA 设计和调试阶段能够迅速进行迭代。设计人员希望缩短每一设计迭代的编译时间，能够有信心保持时序逼近结果以减小完成设计所需要的设计迭代次数。为解决这些问题，FPGA 采用了 ASIC 设计中成熟的方法。FPGA 和 EDA 供应商提供以前只有 ASIC 设计工具中才有的渐进式设计和编译功能。这些功能包括支持设计重用、设计变化以及工程顺序变化的自上而下方法，以及包括团队设计流程的自下而上设计方法。

Altera 的 Quartus® II 设计软件的渐进式编译功能满足了当今 FPGA 设计人员对提高设计效能的需求。本白皮书介绍渐进式编译流程是怎样提高高密度、高性能 FPGA 设计效能的。

传统的“固定式”编译和渐进式编译

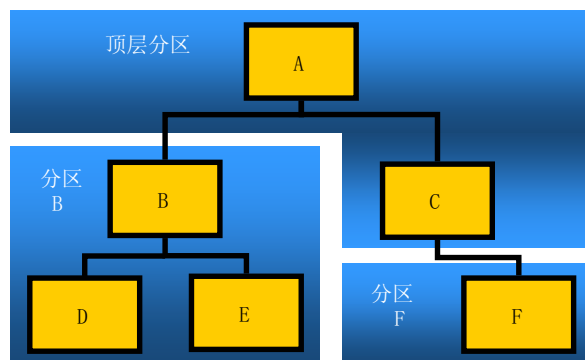
传统上，层次设计在逻辑综合和适配（或者布局布线）之前被展开为单个网表。每次修改设计后，都需要对设计重新编译。这样做的原因之一是通过处理整个设计，编译器可以进行全局优化以提高资源利用率和时

序性能。然而，对整个设计进行编译会导致在微小的设计改动时也需要很长的编译时间。而且，设计人员也不愿意看到一部分设计的改动会影响设计中其他部分的时序逼近。在小型设计中，例如 CPLD 或者低密度 FPGA 设计，当简单的按键式编译很容易达到时序要求时，设计人员能够成功地使用固定式编译流程。当编译时间和保留时序逼近结果并不重要时，固定式设计是令人满意的。

对设计中的其他部分进行修改时，使用渐进式流程可以保留未修改逻辑部分的结果和性能不变。这样能够大大缩短设计编译时间，每天可以进行更多的设计迭代，更有效的达到时序逼近。渐进式设计流程还支持简单的设计重用，让设计人员采用基于团队的设计方法在当今多地区设计环境中独立建立并优化设计模块。建议在大型设计和高密度器件设计中采用渐进式编译，相对于器件体系结构速率性能要求较高的设计也可以采用该方法。

当使用渐进式编译时，设计层次被映射到设计分区中，在编译过程中单独处理它，以实现渐进式编译功能。设计中的每个实体或者实例不会被自动考虑为设计分区；设计人员必须将顶层以下的一个或者多个设计层次指定为渐进式编译设计分区。当分区被声明后，该分区中的每个层次成为同一分区的组成部分。当为已有分区中的层次建立新分区时，新的低层分区中的逻辑不再是上层分区的组成部分。图 2 所示是一个设计层次实例，其中，实例 B 和 F 被指定为设计分区。分区 B 包括子实例 D 和 E。“顶层”默认分区含有顶层模块 A 以及实例 C，原因是它没有被分配到任何其他分区中。

图 2. 设计层次中的分区



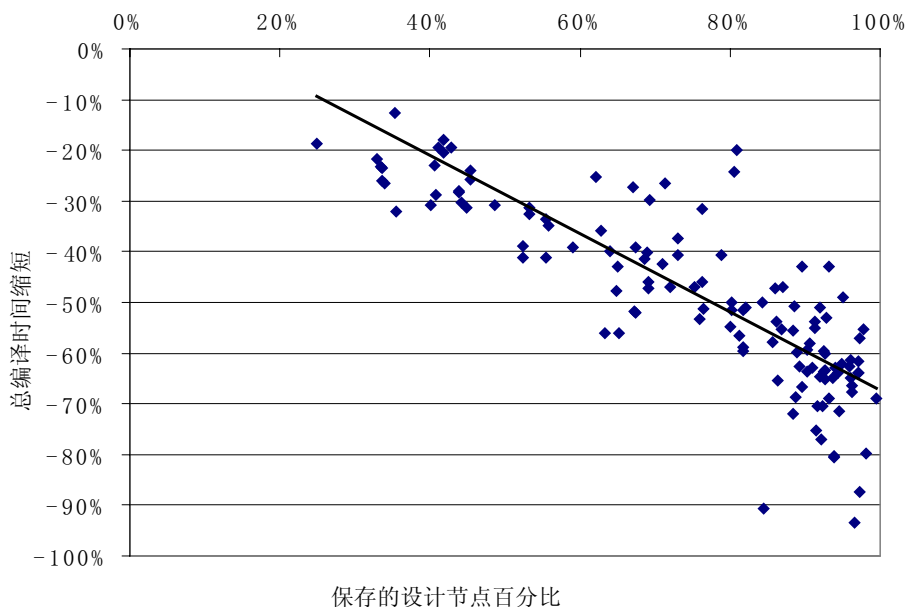
建立分区可以防止软件在综合或者布局期间跨过分区边界进行逻辑优化，还可以让每一分区每次单独进行综合或者布局布线，实现渐进式编译。

渐进式编译是怎样提高效能的

如前所述，渐进式编译使设计层次能够分到逻辑分区中进行综合和适配。重新编译设计时，软件可以使用更新后的 HDL 代码和设置，或者重新使用并保留每一分区的编译结果。重点对某一分区进行编译，同时没有变化的分区不进行重新编译，从而缩短了设计迭代时间。也可以针对设计分区采用物理综合等优化方法，同时保持其他逻辑模块不动。

图 3 中的基准测试表明，设计节点的 80% 以上保持不动，设计改动后进行重新编译，平均编译时间缩短了大概 50% 到 70%。这意味着，在一次固定式编译相同的时间内，可以进行两到三次设计迭代。

图 3. 保持设计中未改动部分不变时，缩短了适配时间。



渐进式编译技术的另一主要优点是保持性能不变。通过只对设计中的特定分区进行编译，其他分区的时序性能保持不变。保持性能不变减少了设计迭代时间，使设计人员能够更高效的达到时序逼近。

渐进式编译同时支持自上而下和自下而上设计方法。在自上而下的流程中，不同的设计人员或者知识产权 (IP) 提供者可以分开建立并验证 HDL 代码，然后，由一个设计人员管理整个设计最终的 Quartus II 工程。采用自下而上设计流程后，每个设计人员或者 IP 提供者可以在分开的工程中完成设计布局布线优化，然后将每个低层工程导入到一个顶层工程中。这些方法支持基于团队的设计流程，由其他地方的团队成员或者第三方建立设计分区。不同的分区被集成到一起，而不会互相影响。当其他设计分区丢失或者不完整时，渐进式流程还可以用于编译和优化别的设计分区。

渐进式流程适用的设计环境

在设计流程的不同阶段都可以采用渐进式编译。在下面的设计环境中，解释了渐进式编译是怎样以各种方式提高设计效能的。

当改变源文件时，缩短编译时间

渐进式编译不需要对设计中的其他部分进行重新编译便能够对分区中的源代码进行更新。例如，这一功能可以用于编译后发现 HDL 源文件中有错误的情况。当找到问题所在的设计部分，而且不希望影响其他设计模块的时序性能时，只编译受影响的代码，而保持设计的其他部分不动。

如果设计被分成区以便进行渐进式编译，设计人员可以在 Quartus II 软件中更新单个源文件，如下所述。首先，修改并保存 HDL 源文件。设置软件在没有变化的分区中重新使用后适配网表，让软件自动重新编译修改过源文件的分区。这样，和重新处理整个设计相比，大大缩短了编译时间，同时保持未改动模块的性能不变，不需要再进行额外的时序逼近处理。

在加入其他逻辑前，优化设计部分结果

渐进式编译还可以用于优化一组隔开的分区，然后锁定布局保持结果不变，直到完成设计的其他部分。例如，可以为某些 IP 建立分区，在与其他定制逻辑合并前，通过指令完成优化。如果每个分区中含有一个关键通路，分区之间不相关，那么最适合采用这一流程。

对于这种环境，打开需要的其他优化选项，只对含有关键时序 IP 模块的分区进行编译。把其他的设计模块分配为“空”分区，这样，不会编译这些逻辑。关键分区达到时序逼近后，采用后适配网表，保持其组成和布局不变。现在，不用再重新访问这一逻辑模块的时序逼近，对其他分区进行逻辑编译。

采用在系统逻辑分析器进行渐进式调试

渐进式调试是渐进式设计流程中非常强大的功能。当加入内部逻辑分析器来调试设计，或者修改逻辑分析器设置而不修改逻辑设计或者其布局时，这一流程能够缩短编译时间。Quartus II 渐进式编译保留最初设计的结果，加入 Altera SignalTap® II 逻辑分析器，不需要重新编译最初的源代码。

采用 SignalTap II 逻辑分析器不用建立任何设计分区来进行渐进式调试；分析器自动划出自己单独的设计分区来完成渐进式编译。设置软件来保留所有设计分区后适配结果，包括默认的顶层分区。设置分析器来探测设计网表中的后适配节点名称，Quartus II 适配器会在下一次编译期间，在后适配网表中加入 SignalTap II 逻辑，并不需要修改已有的设计结果。

实现基于团队的设计环境和自下而上的设计流程

一个包括几个低层设计的工程可以被分成多个工程，由不同的设计人员分开实现。在这种环境下，每个顶层工程例化每个低层设计，每个模块的设计人员可以单独优化他们的设计，确保将编译结果传给工程组长之前达到时序逼近。

在这种环境下的工程组长首先建立一个最终含有整个设计的工程，为每个子设计建立空设计，将设计约束传给低层设计的设计人员（通常包括平面布局约束，以避免资源位置冲突）。每个团队成员单独优化他们的设计，验证时序要求，然后导出分区及其布局信息。工程组长将每个设计模块的信息导入到顶层工程中，编译整个设计。

如果一个分区中所有的关键时序通路都是分开的，那么不需要进一步的工作，顶层设计就能够满足其时序要求。如果交叉分区通路不能满足时序要求，工程组长向低层设计人员提供其他信息，帮助他们优化每一设计。Quartus II 软件的一个特性能够自动产生脚本，将整个设计的信息传给低层模块的每个设计人员，例如链接分区的端口位置等。

按照指南来建立渐进式设计

渐进式编译流程要比固定式编译需要进行更多的前端规划。例如，必须构造源代码或者设计层次以保证逻辑能够正确分组，进行优化。比起在设计后期重新构造代码，更容易在设计早期实现正确的逻辑分组。设计分区后，设计人员需要将每个分区分配到器件中的某个物理位置，以建立设计平面规划。不好的分区或者平面规划分配会劣化设计面积利用率和性能，很难达到时序逼近。和固定式编译相比，渐进式编译通常需要设计人员严格按照良好的设计习惯进行设计。

规划设计时，设计人员应记住每个分区的大小和范围，随着设计的发展，知道设计的不同部分会怎样变化。经常变化的逻辑应和设计中的固定部分分开。设计层次应该和一个分区中的关键时序逻辑分开，最好和寄存端口边界分开，这样，软件能够高效的优化每个独立分区。

当规划设计时，使用 Altera 文档中提供的指南可以确保得到良好的结果。随着 FPGA 容量的增大，并且越来越复杂，按照良好的设计习惯进行设计对所有设计流程都非常重要。坚持按照所推荐的层次同步设计进行设计，能够使设计更加可靠，并且容易调试。使用渐进式编译流程会在工程中增加额外的步骤和需求，但是保持了关键模块的性能不变，缩短了编译时间，大大提高了设计效能。

结论

随着 FPGA 设计容量和性能的提高，渐进式编译方法在 FPGA 设计领域的应用越来越广泛。Altera 的 Quartus II 渐进式编译技术的编译时间平均缩短了 60%，保持设计性能不变，支持基于团队的设计流程，提高了设计人员的效能。通过一定的设计规划和分区，该技术缩短了设计迭代时间，大大减少了开发时间。

通过渐进式编译来支持性能不变和基于团队的流程使设计人员以更少的设计迭代，更高效的达到时序逼近，更迅速的将产品推向市场。

详细信息

- *Quartus II 手册第一卷，分层和团队设计的 Quartus II 渐进式编译：*
www.altera.com/literature/hb/qts/qts_qii51015.pdf
- *Quartus II 手册第一卷，渐进式编译分区和平面规划分配的最佳实践：*
www.altera.com/literature/hb/qts/qts_qii51017.pdf
- *在 40-nm 工艺节点实现世界上最先进的定制逻辑器件：*
www.altera.com/literature/wp/wp-01058-stratix-iv-40nm-process-node-custom-logic-devices.pdf

致谢

- Jennifer Stephenson，Altera 公司软件应用工程 MTS 应用工程师。
- Scott Brissenden，Altera 公司软件和 IP 工程性能分析组高级软件工程师。
- Albert Chang，Altera 公司软件营销资深产品营销工程师。



101 Innovation Drive
San Jose, CA 95134
www.altera.com

版权 © 2008 Altera 公司。保留所有版权。Altera、可编程解决方案公司、程式化 Altera 标识、专用器件名称和所有其他专有商标或者服务标记，除非特别声明，均为 Altera 公司在美国和其他国家的商标和服务标记。所有其他产品或者服务名称的所有权属于其各自持有人。Altera 产品受美国和其他国家多种专利、未决应用、掩模著作权和版权的保护。Altera 保证当前规范下的半导体产品性能与 Altera 标准质保一致，但是保留对产品和服务在没有事先通知时的变更权利。除非与 Altera 公司的书面条款完全一致，否则 Altera 不承担由使用或者应用此处所述信息、产品或者服务导致的责任。Altera 建议客户在决定购买产品或者服务，以及确信任何公开信息之前，阅读 Altera 最新版的器件规范说明。