

FPGA 电源设计适合并行工程吗？

如果设计师可以在开发过程早期就满足基于 FPGA 的设计，提出的功耗要求和约束条件，那么在系统的最终实现阶段就能形成极具竞争力的优势。然而，根据整个技术文献中这种自我暗示式的反复祷告，今天基于 FPGA 的系统中还有什么会使得完全遵循这个建议变得不切实际或过于困难呢？尽管能够使用各种[开发工具](#)，如专门针对 FPGA 项目开发的早期功耗预估器和功耗分析器，但对电源设计师来说，在设计过程早期就考虑最坏情况而不是最佳情况的电源系统是有好处的，因为在许多方面仍有太多的不确定性，比如在硬件设计完成和功耗可以测量之前，静态小电流状态与全速工作状态之间的动态负载要求将如何波动。

采用并行工程(CE)技术，可以为在项目中使用的 FPGA 器件的开发团队，提供一种快速方便地在当前设计的处理性能、材料清单(BOM)成本和效率之间寻找和实现最有效平衡的方法吗？理解并行工程如何影响一个团队的设计工作，以及它如何影响开发团队从项目一开始就解决 FPGA 及系统其余部分的电源要求的能力，都有助于回答这个问题(参考副标题“并行工程”)。

并行工程，是一种有助于设计团队更加快速地发现 and 解决一起协作产生最终设计的各门科目之间假设脱节问题的机制。任何开发团队在设计开始就完全正确地获得一个复杂系统的全部要求基本上是不可能的——因此尽可能早地发现、判断和放弃假设与设计决策的脱节、并用能够以可能最低的成本指导项目更接近理想结果的条件与决策来代替是更加高效的一种方法。

后期设计阶段和最坏情况下，FPGA 电源系统设计的复杂性和潜在后果足以证明采用并行工程方法的合理性吗？为了回答这个问题，我们需要理解：FPGA 电源系统设计师面临的设计复杂性和不确定性根源是什么，这些原因如何影响他们在设计电源时必须做出的权衡决策？

复杂性和不确定性

设计团队中的每位成员都在经历复杂性和不确定性的增加——幸运的是，提高集成度和抽象水平多少可以降低一些复杂性和不确定性，并有助于将整体复杂性保持在人类设计师能够理解和应付的范围内。正如任何会在设计后期增加其影响力的科目一样，上游设计假设与决策可能会形成额外的复杂性与不确定性来源，如果能够较早地协调与交流，可以最大程度地减小这些复杂性与不确定性。

电源设计是在复杂性日益增加的系统中这些潜在的下游科目之一。在本例中，让我们从电源设计师的角度看一下复杂性和不确定性的来源。影响电源设计的两个关键 FPGA 因素是电压和电流要求。

FPGA 电压要求趋势正在推升复杂性，因为它们要求日益增多的电源轨。今天的高端 FPGA 不再只是需要两个电源轨用于内核和 I/O 单元以及可能第三个电源轨用于辅助功能，而是要求十个以上的外部驱动电源轨。

为什么需要的电源轨数量增加得如此显著呢？SRAM 单元可能要求比内部逻辑门稍微高一点的电压，以确保可靠的全速工作，同时还需较低的电压用于待机模式。工业标准会防止不同的 I/O 单元共享相同的电源轨，从而增加所需的电源轨数量，因为它们可能将不同的 I/O 单元和物理收发接口锁定到具有不同电源噪声极限和电压值的不同电源。举例来说，以太网工作时的 I/O 电压可能不同于 I2C 总线。一种是板上总线，另一种是外部总线，但两者都可以用 FPGA 实现。减少抖动或提高敏感电路(如低噪声放大器、锁相环、收发器和精密模拟电路)的噪声余量，也可能增加对更多电源轨的需求，因为它们无法与较高噪声元件共享相同的电源轨，即使它们工作在相同的电压。

除了要求日益增多的电源轨外，当前 FPGA 的工作电压也要比以前的 FPGA 低，因为这有助于降低功耗，提高集成度，但也增加了复杂性，因为电源必须能够保持越来越严格的电压容差要求(见图 1)。举个例子，自从用 130nm 工艺生产 FPGA 以来，基于 28nm 技术节点的 FPGA 的内核电压纹波容差的公开幅度已经降低了一半还多。误差预算百分比已经从 5% 下降到 3%，并正在向 2% 迈进。保持电压容差要求与理解并满足 FPGA 电流要求有关。

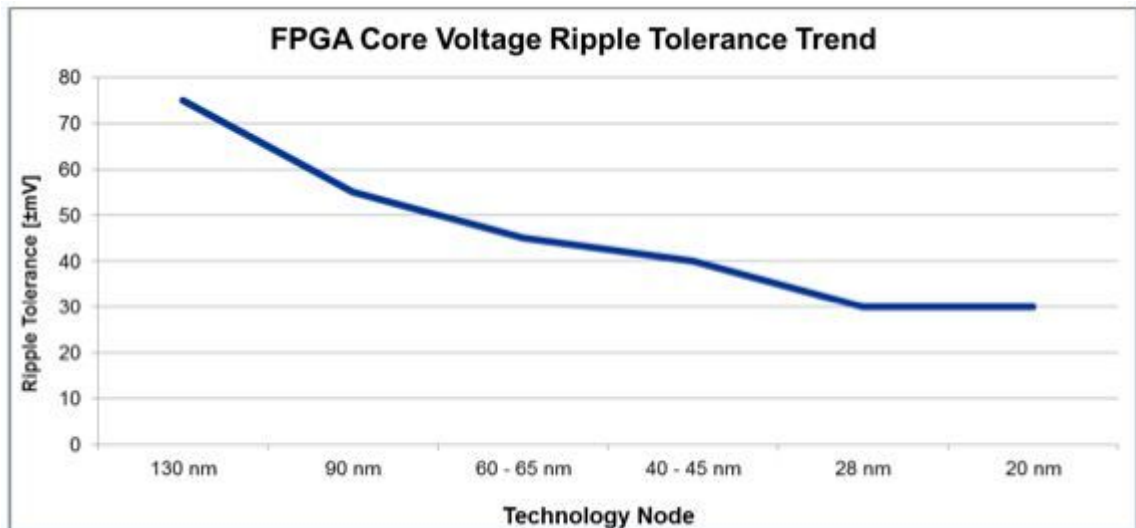


图 1：经过 4 代工艺技术节点的发展，平均电压纹波容差下降了一半还多，对电源设计师来说这就是增加复杂性的原因。

FPGA 电流特征趋势正在推动复杂性的提高，因为 FPGA 中更高的密度和包含的外设/功能/IP 模块的数量正在呈摩尔定律增长——每两代工艺节点相比，相同面积的硅片所容纳的模块数量基本要翻倍。虽然提供给 FPGA 的电压是固定的，但每个电压的工作电流不是固定的，会根据 FPGA 逻辑的实现方法变化而发生波动。

当内部逻辑门块或 I/O 单元在高利用率和低利用率之间转换时，电流波动异常剧烈。随着 FPGA 切换到更高的处理速率，消耗电流将增加，电压将趋于下降。一个好的电源设计要防止压降超过电压瞬时门限。同样，当 FPGA 切换到较低处理速率时，电流消耗将下降，电压将趋于提高，电源设计应防止其超过相应的门限。总之，可能会实质影响电源设计的大量不确定性源自 FPGA 设计师如何在 FPGA 上实现系统。

这类不确定性特别影响 **FPGA** 系统，部分原因是因为使用 **FPGA** 的关键特性之一是，设计师可以创建任何大小的处理资源和任意数量的冗余处理资源，以便与软件可编程处理器相比能用较短的时间和/或较低的功耗解决他们的问题。因此，虽然软件可编程处理器拥有可以同时操作的有限处理资源，但 **FPGA** 提供了创建专门的、最优的和定制的处理资源的机会，不过要求定制电源设计。

供电

理解和管理 **FPGA** 设计师如何在设计周期早期在 **FPGA** 上实现高处理状态和低处理状态之间的转换，将显著影响电源设计师优化电源设计和满足系统功耗要求的可选方法。

FPGA 中的每个电源轨没有要求也没有必要采用独立的电源，因为这样会增加成本，占用太多宝贵的 电路板 空间。相反，电源设计师可以使用分布式电源网络，由降压稳压器将系统电源降下来，然后分配给各个负载点稳压器再提供每个电压轨。每个稳压器设计提供恒定的输出电压，只要确保输入电压和输出负载电流在设计范围内。

有两种基本类型的稳压器：线性型和开关型。与开关稳压器相比，线性稳压器更容易实现，可以提供更小噪声或更小电压纹波的更稳定输出，并且使用成本更低，占用的电路板面积更小。然而，它们的电源转换效率要比开关稳压器低很多，特别是当输入输出压差较大时。例如使用线性稳压器从 5V 产生 1V，其转换效率只有 20%，比开关稳压器的近 85% 转换效率低得多。

转换效率是指输出功率与输入功率之比，较低的效率意味着稳压器在消耗功率，而不是 **FPGA** 在消耗功率。因此对于具有大工作电流的 **FPGA** 应用来说，开关稳压器比线性稳压器更适合。一些高端 **FPGA** 系统的快速 I/O 节点电流可达 80A。另外，由于浪费的功耗引起的温升将影响用于维持系统元件性能的散热器或空气对流所需的空气空间。一般来说，如果没有空气对流，每平方英寸铜耗散 1W 的功率将导致温度升高 10℃。

虽然开关稳压器的功效比线性稳压器大得多，但它有噪声问题，即具有更大的电压纹波，因而给电源设计师增加了缩小容差门限的挑战。在电路板上正确放置开关稳压器件是尽量减小电气噪声的关键，而其体积稍大的元件又增加了这种挑战性。

因此，在设计过程中足够早地了解有关功耗预算方面的正确知识，有助于电源设计师合理安排正确的电路板位置和电路板空间，以便使用更高效的开关稳压器，或更好地使用较低效率的线性稳压器。

早期规划

大部分 **FPGA** 功耗取决于 **FPGA** 设计师在系统开关频率、输出负载、供电电压、互连数量、互连开关百分比以及逻辑与互连模块结构方面的实现选择。这些选择反过来又影响电源设计师的判断和在系统设计方面的权衡，进而可能影响最终系统性能。

幸运的是，**FPGA** 电源设计师有多种工具和技术可以用来在设计过程早期分析电源问题。举例来说，大多数 **FPGA** 供应商提供的早期功耗预估器和功耗分析器可帮助设计师建立功耗预算。设计师可以使用基于软件的早期功耗预估器——基本上是美化过或文字形式的电子表格——在设计过程早期收集逻辑规模和工作速率的值与假设条件，然后估计系统各个部分将使用多少功耗(见图 2)。

Group	Regulator Input Voltage	Regulator Current Draw (A)	Voltage (V)	Load Current (A)	Load Current Margin	Parent Group	Regulator Type	POK	Suggested Enpirion Part	Pin Compatible Parts
1	5.00	3.347	0.85	16.735	30.00%	0	Switcher	Yes	2 X EN63A0QI	
2	5.00	1.160	2.50	1.972	30.00%	0	Switcher	Yes	EN5329QI	EN5319QI EN5339QI
3	5.00	0.814	1.80	1.923	30.00%	0	Switcher	Yes	EN5329QI	EN5319QI EN5339QI
4	2.50	0.914	1.50	1.295	30.00%	2	Switcher	No	EN6337QI	EN6347QI
5	5.00	0.159	3.00	0.225	30.00%	0	Switcher	Yes	EN6310	
6	5.00	0.355	1.00	1.508	30.00%	0	Switcher	No	EN6337QI	EN6347QI

图 2: 在这个基于软件的早期功耗预估器中显示的工作表可以根据规划的 FPGA 用途为每个电源轨提供建议的元件(Altera 提供)。

早期功耗预估器可以帮助电源设计师输入 FPGA 上不同子系统的使用率估计值。在工具实例中, 快照底部栏提供的工作表用于捕获每类资源的功耗值, 包括逻辑、内存、各种 I/O 和硬件信号处理资源。这些值还可以供工具中的其它工作表使用。

然而, 只是估计功耗是不够的——电源设计师需要设计电源树以支持 FPGA 设计师意图如何使用 FPGA。通过估计 FPGA 设计总的静态和动态功耗, 这些工具可以帮助电源设计师将功耗要求转换为合适的电源树, 同时确保设计符合设计团队/system 权衡决策, 并满足电流与电压要求。包括图 2 所示例子的一些工具可能会建议能够满足设计需求的电源管理器件。预估器反映了 FPGA 供应商和设计师的多年经验, 在基于实际设计实现的数字出来之前非常管用。

随着 FPGA 设计的进展, 电源设计师开始使用电源分析工具并根据详细信息更加精确地获知功耗和电源要求, 比如来自 FPGA 设计工具的网表输出, 设计是如何在 FPGA 中实现的等。

功耗预估和分析工具的目的是帮助设计师建立早期的功耗预算指南。有了早期的功耗预估, 设计团队就能同时开展 电路板 的设计搭建和 FPGA 设计, 不仅节省了时间, 而且一旦 FPGA 设计完成就会形成更具决定性的电路板设计, 从而帮助团队完成更高效的测试和优化。

电源设计师需要在系统硬件集成过程中测量和验证实际的 FPGA 性能, 因为 FPGA 对实际设计和环境工作条件非常敏感, 可能与估计和分析工具的结果有所出入。

关于协商

有关估计和系统集成功耗数值可能不同的最终免责声明并不会给我们带来很多信心。即使使用这些数值, 电源设计师仍然面临极大的不确定性。我们绕了一大圈又回到了 FPGA 电源设计是否为并行工程技术做好准备的原地。

与航空和微处理器团队如何能够理解上游设计决策如何影响下游设计要求的例子相似，理解电源设计要求如何受电路板与 FPGA 设计师早期选择的影响为每个团队成员提供了一种机制和机会，即他们可以在成本固定进设计之前就开始交流和协商如何优化整个系统，而不只是系统的一个部分。

从项目开始就与开发团队的每位成员进行早期协调和交流的主要价值是，尽可能在设计过程早期发现不同领域专家——本例中是电源设计师、FPGA 设计师和电路板设计师以及位于或影响电路板的其它系统元件、FPGA 和电源系统的设计师——之间的假设脱节问题。在这点上，受影响的组员相互间可以讨论、争论甚至争吵，最终以较低成本开始和完成寻找及协商解决方案的过程，因为他们需要避免所有昂贵的重复工作以及后期设计阶段的要求更改。

能够有选择地使用更高效的开关型稳压器要求掌握有关系统功耗要求的相当准确的预知能力，以便在正确位置有合适大小的电路板空间能被分配给电源稳压器和元件，从而满足包括电压纹波容差要求在内的电源要求。一个不良或不准确的功耗预测结果可能意味着使用较低效率的稳压器来满足电压纹波要求，但这个“纹波”给设计的其余部分提出了额外的要求，比如需要更大的电源，需要散发更多的热量，甚至使 FPGA 工作在较慢的速率。

功耗估计和分析工具的一个重要价值是让每个人都尽早地关注功耗。在当代设计中，FPGA 可能是影响系统性能和功耗的主要因素——因此也可能是电源设计的主要考虑部分。从设计过程一开始就与电源设计师协同工作有助于讨论系统权衡决策以及如何使用 FPGA、然后使用工具获得更早更精确的功耗估计。

成功并不是在你开始之前就使所有系统要求变得正确，而是尽可能早的发现和放弃不好的决策，并以可能最低的成本用能够指导项目更接近理想输出的决策代替之。结果是在后续项目中你能收获到更早、更方便和更精确的功耗预测的好处。

在我开展一些航空集成产品团队项目(我们的并行工程版本)时，我们使用了 80-20 规则的一个变化版本：80%的项目成本是由前面 20%的设计工作确定的。在那以后，你可望做到的最好一点是应付剩余的 20%。当我们知道最少部分对项目最终成本有最大影响时，我们就能高效地做出决策。

这个表达可能并不完全准确，但似乎里面又暗含了某些智慧。更重要的是，它提供了在你纠结于 80%之前提出问题并尽可能合理地与拥有不同领域专家的团队成员合作的警告和提醒，以免在项目到达最终开发阶段时后悔莫及。

并行工程

在美国国防部防御分析研究所于 1988 年发表的一份报告中是这样描述并行工程的：它是针对产品及其相关工艺的一体化并行设计的一种系统性方法，包括制造和支持。这种方法试图引导开发人员从一开始就考虑从概念直到废弃的产品生命周期中的所有元素，包括质量、成本、计划和用户要求(参考文献 1)。并行工程与许多其它术语类似和重叠，比如协作工程、同步工程以及综合性产品开发。

早期，对于需要两个或多个人一起工作的每个项目来说，总是存在某种形式的并行工程，但现代并行工程依赖于信息和通信技术将更大的多科目开发团队组织在一起工作，并且用比以往任何时候都更快的速度分享新项目信息。虽然许多并行工程说明强调的是并行设计可以

缩短产品开发时间，但需要提醒的是，并行工程的最重要机制是它能用来更加快速地发现协同工作以生成最终设计的各种科目之间的假设脱节问题。随着生产系统中出现的复杂性和这些脱节问题结果的增加，较早发现和解决脱节假设的好处会超过因协调与通信而增加的成本。

航空项目是最早采用现代并行工程方法的，目的是发现负责原型创建、制造和维修产品的团队之间的脱节。许多最早发现的假设脱节发生在设计过程中上游和下游任务之间。例如，在原型创建期间作出的决策可能使得制造或维修更加困难和昂贵——与此同时，如果他们采用稍有不同的方法，可能就可以高效地满足他们的要求，对下游团队的任务要求几乎没有或非常少的影响。发现和解决载人安全系统中的脱节假设的好处要远远超过整个设计过程中因不同团队之间的协调与通信而增加的成本。

多年以后，新型微处理器的开发开始受益于处理器架构师、提供编译器等软件[开发工具](#)的团队和其它软件开发人员之间的早期并行或协作开发工作。处理器架构开始采纳可以简化编译器开发人员必须作出的假设的决定和功能——从而导致更快、更小和更高效的生成代码。选择不同但等效的架构方法的好处是，即使软件系统的复杂性和成本还在快速增加，但可以实现更好的编译后代码性能。

在上述每个案例中，尽早并频繁地与系统开发团队的其他成员开展协作和交流所需的额外付出完全可以被发现假设中的错误和脱节带来的更大好处所抵消，因为这时这些问题的协商和解决代价较低。