中国集成电路 CIC
China Integrated Circuit

高密原型验证系统解决方案(上篇)

吴滔,林铠鹏
上海国微思尔芯技术股份有限公司

0 引言

随着当今 SoC 设计规模的快速膨胀，仅仅靠几颗当代最先进的 FPGA 已经无法满足原型验证的需求。简单的增加系统的容量,会遇到系统时钟复位同步，设计分割以及高速接口和先进 Memory 控制器 IP 验证等多重困难。此时,一个商用成熟的能解决以上多重挑战的通用高密原型验证系统方案显得十分有必要。

本文主要分析了用户在进行大规模 SoC 设计原型验证过程中在全局时钟及复位同步，大规模设计分割以及高速接口和先进 Memory 控制器 IP 验证等方面遇到的困难，并提出了相应的解决方案来帮助用户来克服这些困难。

1 大规模 SoC 设计原型验证中遇到的挑战

1.1 全局时钟及复位方案挑战

全局时钟和复位的解决方案是构建一个大规模复杂 SoC 原型验证系统首先要考虑的问题。SoC 原型验证系统需要保证系统的全局时钟和复位能够同步的送到原型验证系统的每一个 FPGA 节点,这是大规模 SoC 原型验证系统能够正常工作的先决条件。随着用户的 SoC 设计规模的增大,对原型验证系统的全局时钟和复位的同步方案的要求也会变得复杂。

1. 如果用户的设计规模用单台 FPGA 原型验证平台就可以容纳，那么原型验证系统的全局时钟和复位同步的实现难度会比较简单，只需要实现单台平台上的多颗 FPGA 的时钟复位同步即可。

2. 如果用户的设计规模再大一些，需要用多台 FPGA 原型验证平台组成一套验证系统，那么原型验证系统的全局时钟和复位同步的实现难度就会高一些，需要实现多台验证平台上的多颗 FPGA 的时钟复位同步。

3. 如果用户的设计规模进一步扩大，需要用数十台甚至上百台 FPGA 原型验证平台才能组成一套验证系统，那么原型验证系统的全局时钟和复位的同步的实现会变得十分复杂，需要用时钟复位模块级联的方式实现大量平台上的多颗 FPGA 的时钟复位同步。

1.2 大规模设计分割挑战

由于大规模复杂 SoC 设计规模庞大，把设计映射到多个 FPGA 组成的网络(也称设计分割)是大规模原型系统实现过程中必不可少的重要环节。然而大规模设计分割也给用户的原型验证带来了不少的难题：

1. 首先要解决的是设计综合时间长的问题。一

个设计规模达到几亿门甚至数十亿门的大规模 SoC 设计，如果采用整个设计先综合，再分割的方式，设计综合的时间会非常的漫长，一次综合流程就要花费数天甚至数周的时间。遇见设计规模特别大的设计，甚至会出现编译软件或服务器崩溃的情况。显然，面对大规模 SoC 设计用户需要采用并行编译的方式。

2. 其次，当设计原始的 SoC 设计分割到多颗 FPGA 中，原型设计的全局时钟树处理也是一个难题。用户的 SoC 设计全局时钟树往往很复杂，需要怎样处理来保证复杂的全局时钟树在每颗 FPGA 上同步的实现？

3. 将一个大规模 SoC 设计分割到多颗 FPGA 时，FPGA 之间的互连信号往往有成千上万根，FPGA 的 IO 管脚数远远无法满足互连信号数的需求。人工写互连接口 Wrapper 设计，进行 FPGA 之间互连信号的管脚复用是吃力不讨好的工作。需要有自动化的管脚复用的方案帮用户减轻负担，提高效率。

1.3 高速接口和先进 Memory 控制器 IP 验证挑战

大规模复杂 SoC 设计原型验证时一些高速接口，如 PCIe Gen3/Gen4 接口，对应的 PCIe 控制器 IP 在 FPGA 里跑的速度都很低，无法对接外部实际 PCIe 主机或者设备。另外，一些先进的 Memory 控制器 IP，如 DDR5, LPDDR4/5, HBM2/3 缺少 FPGA 厂商提供的 PHY 解决方案，无法在 FPGA 原型验证系统中运行起来。遇到这些棘手的问题，有些用户甚至选择放弃这些高速接口及 Memory IP 在原型验证系统中的验证，但也因此会给用户的 SoC 设计原型验证带来风险。

2 面向大规模复杂 SoC 设计的高密原型验证解决方案

为了满足大规模 SoC 设计在原型验证系统领域的复杂需求，用户迫切需要有一系列面向大规模 SoC 设计的高密原型验证系统的软硬件通用解决方案来帮助用户在系统时钟复位同步，设计分割及高速接口及先进 Memory 控制器 IP 验证等方面取得重要突破，使得用户能够解决大规模 SoC 原型验证面临的关键困难，降低项目风险。

2.1 大规模设计全局时钟及复位解决方案

一个通用的高密原型验证系统需要能做到根据用户的 SoC 设计规模灵活的提供适合的全局时钟及复位方案。

1. 如果用户的设计规模用单台 FPGA 原型验证平台就可以容纳，用户的全局时钟复位方案则只需要保证单个 FPGA 原型验证平台的全局时钟和复位可以同步的送到单台平台的每颗 FPGA 上即可。在这种需求背景下我们可以考虑用下面的方案来设计全局时钟及复位的功能：

- 单台 FPGA 原型验证平台的全局时钟和复位的设定和操作需要能够通过软件远程实现，以方便用户使用。
- 在单台 FPGA 原型验证平台上可实现多个可编程全局时钟，使得用户可以根据设计需求灵活的设置所需的全局时钟频率。
- 可编程的全局时钟和全局复位通过 1 推多的驱动芯片以及等长的板级走线来保证全局时钟和全局复位同步的传送到每一颗用户 FPGA。

除了以上满足单个原型验证平台的全局时钟和复位基本需求之外，我们还需要考虑到将来当用户设计规模进一步扩大时，现有的全局时钟和复位方案的可扩展性：

- 单台 FPGA 原型验证平台除了使用本平台的本地时钟复位源之外，也可以接受来自外部的时钟复位源，以支持全局时钟复位的输入级联扩展。本地的时钟复位源和外部时钟复位源可以在软件的控制下通过选择器来切换。
- 单台 FPGA 原型验证平台的每一个用户 FPGA 输出的时钟复位源以及外部输入的时钟复位

源可以在软件的控制下通过多路选择器来输出到该FPGA原型验证平台的外部时钟复位输出口，以支持全局时钟复位的输出级联扩展。

图1就是一个具有可扩展性的基于单台FPGA原型验证平台的全局时钟同步系统架构图（全局复位系统也可采用类似的架构来实现）。

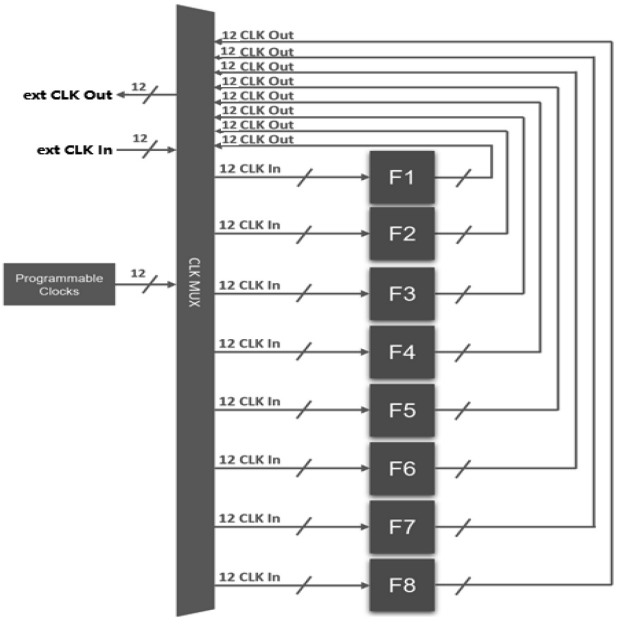

图1 单个原型验证平台的全局时钟同步系统

2. 如果用户的设计规模需要用多台FPGA原型验证平台来组成一套验证系统，就需要一种能实现多台FPGA原型验证平台上的FPGA时钟复位同步的解决方案。

比较常用的一种支持多台FPGA原型验证平台的FPGA时钟复位同步的方案是采用一个外置的全局时钟复位模块(CLKM)将全局时钟和复位通过等长的时钟线缆同步的送到多个FPGA原型验证平台(FPP)。FPP通过外部时钟输入接口接收外部送来的全局时钟复位源再同步的传送到FPP上的每一颗FPGA。其全局时钟复位同步系统组网如图2所示。

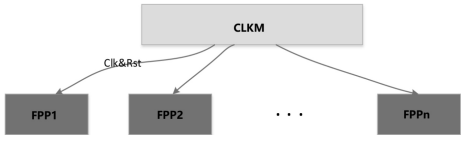

图2 基于全局时钟模块的多个FPP全局时钟同步系统

这个方案的优点是单套FPP的全局时钟复位方案的实现简单，直接采用图一中的全局时钟复位方案接入CLKM送来的外部时钟复位源即可完成多台FPP的全局时钟复位同步。

但是由于CLKM上的时钟驱动芯片的驱动能力有限，配套的高性能时钟线缆长度最长也只有1米左右，这样的全局时钟复位同步系统只适合支持设计规模小于十台FPP的系统。对于更大规模的原型验证系统,则需要更加复杂的解决方案。

3. 当用户的设计规模进一步增大，需要用数十台甚至更多的FPP平台来组成一套验证系统,此时就需要用级联的方式来实现大量平台上的多颗FPGA的时钟复位同步。这种情况下,用户可以使用多个CLKM采用Master/Slave的方式级联以支持更大规模系统的时钟复位同步。设为Master的CLKM产生全局时钟和复位，通过等长的时钟线缆下发给多个下级Slave CLKM。再由下级Slave CLKM下发全局时钟和复位给多个下游FPP。其中Master CLKM的时钟复位源既可以来自于本身的可编程时钟芯片也可以来自于任意一个FPP输出的反馈时钟。图3是用CLKM进行全局时钟复位级联的拓扑结构图。

这个方案通过分层级联的方式，可以实现几十

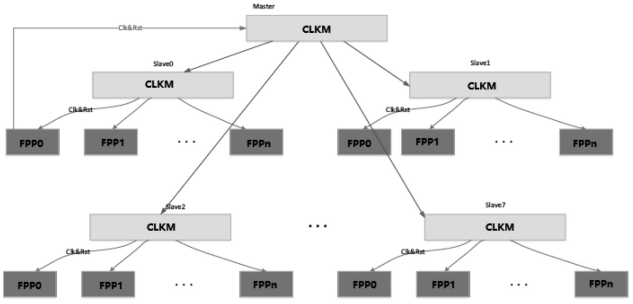

图3 基于全局时钟模块级联的全局时钟同步系统

个甚至上百台FPP的全局时钟组网,为几十亿门级的超大规模原型验证系统的全局时钟复位架构提供了解决方案。这个方案的技术难点是，由于每个CLKM上用到的1推多时钟驱动芯片的延迟会有微小的偏差,通过CLKM的分层级联会带来时钟延迟的偏差积累，会导致最后送到每个FPGA的全局时钟同步产生不可忽视的偏差进而造成用户的超大规

模原型系统无法完成全局时钟同步。为了解决这个问题,我们可以在 Master/Slave CLKM 模块之间增加反馈时钟线及带时钟相位反馈校准的 PLL 芯片以实现多级 CLK 模块之间的时钟相位校准。

当用户有了支持全局时钟模块级联及时钟相位校准的通用解决方案，即使用户的设计规模不断的扩大,我们也可以通过分层的方式,对用户的全局时钟进行分级校准，以保证超大规模原型系统能够实现全局时钟及复位的同步。

2.2 大规模设计分割方案

设计分割往往是大规模复杂 SoC 设计用户在进行原型验证系统验证时遇到的最大的挑战。要做好设计分割需要处理好大规模设计并行编译，全局时钟树复制,自动化管脚复用,系统时钟约束以及系统时序分析等多方技术难点。

2.2.1 大规模设计综合

一个设计规模达到几亿门甚至数十亿门的 SoC 设计,如果将整个设计映射在 FPGA 里做设计综合,综合的时间往往要长达数天甚至数周，这对于在原型验证阶段经常需要修改代码，重新综合设计的用户是无法接受的。显然,面对大规模 SoC 设计用户需要采用将整个设计分块并行综合的方式来进行。

a）一种方式是在大规模 SoC 设计的顶层 RTL 下例化子模块黑盒。将设计规模较大,综合时间较长的一些子模块黑盒化，这样可以大大缩短黑盒化之后的顶层设计的综合时间。同时,被设为黑盒的多个设计子模块也可以被分发到多个服务器上进行并行的综合。

● 这种方式的优点是使用效果明显，门槛低。用户无需使用特别的软件和技术，只需要工程师花一些时间将选定的子模块设置为黑盒进行并行编译,就可以大大缩短整个大规模设计的综合时间。

● 缺点是用户需要对大规模 SoC 设计的子模块资源占用情况比较了解，清楚哪些模块适合设为黑盒进行并行综合,哪些不需要。另外,用户手动修改代码设置子模块黑盒是一个费时费力的工作,一不小心就会出错。

● 一些自动化的分割软件,能够帮助用户将选定的子模块自动设成黑盒，从而达到事半功倍的效果，省去了工程师手动修改代码进行子模块黑盒化的繁琐工作,避免了人工修改代码出错的情况发生。自动化的分割软件支持 LSF/SLURM 功能,能帮助用户高效的在多个 HPC 上进行多个子模块的并行综合。

b）另一种方式是使用基于 RTL 分割工具对大规模 SoC 设计进行 RTL 级分割。RTL 分割工具对读入的大规模 SoC 设计进行设计分析和资源估算,再基于设计分析和资源估算的结果对用户的大规模 SoC 设计进行分层分割，将基于 RTL 的大规模 SoC 分割为以目标 FPGA 为顶层的多个 RTL 模块。RTL 分割完成之后，用户可以调用综合工具对分割后的 FPGA 顶层 RTL 模块进行并行综合，因而大大缩短整个设计的综合时间。

● 这种方式和第一种方式相比的优点是采用成熟的 RTL 分割工具,帮助用户先完成设计的 RTL 级切割。减少了用户手动进行设计分割和设置黑盒的时间,工作量与风险。

● 不足之处在于基于 RTL 级分割的资源估算的精准度以及 RTL 模块的切割边界的优化效果不如网表级分割。

c）一些自动化的分割软件,能够将 RTL 级分割和网表级分割的优点结合起来以帮助用户在节省综合时间的同时取得更好的设计分割结果：

● 在第一阶段使用 RTL 级分割技术将用户的设计初步分割到多个 FPGA，再将分割后的多个 FPGA 顶层 RTL 设计并行综合，以帮助用户节省大量的综合时间。

● 在第二阶段使用网表级分割技术将初步分割的模块进行基于时序驱动的二次优化，使得分割模块间的资源分配更加合理,切割边界更加精准,以帮助用户取得更合理高效的设计分割结果。

2.2.2 全局时钟树处理

大规模复杂 SoC 设计的全局时钟树设计往往很

复杂，在把设计分割到多颗 FPGA 的时候将会面临怎样进行全局时钟树处理，才能使复杂的全局时钟树在每颗 FPGA 上同步的问题。

a）一种方法是将用户的全局时钟树进行简化，再将简化的全局时钟树复制到每一颗用户 FPGA。比如把设计中的多路全局时钟合并成一路全局时钟。或是把一个时钟管理单元生成出来的多路时钟移除，换成直接从 IO 端口接入进来的多路全局时钟。这种处理方法的优点是无需额外的软件或硬件的支持。工程师对设计的时钟树模块源代码进行直接改动以实现 SoC 设计的全局时钟树简化。缺点是简化后的设计逻辑功能和原始的设计有时会有区别，进而降低用户的原型验证的覆盖率和准确性。另外这种方法的使用场景也受到限制，如果用户的多路全局时钟无法合并，或者用户在移除替换时钟管理模块时，发现用户 FPGA 从 IO 端口接入进来的多路全局时钟数目不够用的时候，则无法完成全局时钟树的简化。

b）另一种方法是在 SoC 设计的某一颗 FPGA 中实现整个 SoC 设计全局时钟树，再将全局时钟树产生的时钟通过专用的反馈时钟线或者专用的快速时钟线同步的分发的每一颗用户 FPGA 中。这种做法采用预留硬件资源的方式来解决全局时钟树的同步问题。优点是对于用户的时钟树网络改动小，仅仅需要将全局时钟树的时钟输出接到时钟反馈线或专用的快速时钟线上即可。缺点是占用的硬件资源较多，硬件需要预留多根专用的时钟反馈线或快速时钟线。由于 FPGA 上的全局时钟资源有限，这样的时钟反馈线及快速时钟线注定无法预留太多。当一个用户的全局树的反馈时钟输出数量太多的时候，时钟反馈线及快速时钟互连线就会不够用。另外在一个大规模的 SoC 原型系统里实现时钟反馈线及快速时钟线的物理路径往往比较长，因而会带来较大的时钟延迟，这在一定程度上会给设计的全局时钟性能和全局时钟同步精准度带来负面影响。

c）还有一种方法是将 SoC 设计的复杂全局时钟树通过设计分割软件完整的复制到每一颗用户 FPGA 中，以保证每一颗 FPGA 的全局时钟树同步。这种方式对分割软件的要求比较高，需要软件能够准确分析出用户 SoC 设计的完整全局时钟树，再进行全局时钟树的复制，将复制出来的全局时钟树以网表或者 RTL 代码的形式自动的插入到分割后的用户设计中。同时为了保证每一颗 FPGA 中的全局时钟树的同步，软件需要锁定每颗 FPGA 在全局时钟树上对应的时钟输入管脚，时钟管理单元的物理位置和参数设置，并通过软件技术解决时钟管理单元的时钟输出初始相位对齐的问题。这些需求，采用用户自己研发的设计分割软件往往很难做到，需要使用专门的通用设计分割软件来帮助用户实现复杂全局时钟树复制的需求。

2.2.3 基于高速收发器的管脚复用

从设计分割模式的角度来分，基于高速收发器（Transceiver）IO 的通用的设计分割方案有以下两种分割模式：

● 使用专门的设计分割软件进行大规模 SoC 设计的自动化分割，并利用 FPGA 的 Transceiver IO 实现基于单拍传输的管脚时分复用（TDM）IP。这种方式的好处是对于用户比较省力，省时间。局限是管脚时分复用自动化分割方式会带来额外的性能开销，分割之后的大规模 SoC 设计系统最高只能工作在 20MHz 左右，无法满足某些大规模芯片设计的原型验证速度需求。

● 使用基于 FPGA 之间的 Transceiver IO 进行总线协议的分割。这种方式的好处是，分割之后的大规模 SoC 设计系统时钟能工作在几十 MHz 到 100MHz 以上。局限是基于 Transceiver 的总线协议分割要求 FPGA 之间的分割边界只能是总线接口，并且用户需要手动例化 Transceiver 接口 IP 设计，以实现 FPGA 之间的总线接口切割，比较费时费力。

总体来说，这两种分割模式都有各自的优势，也有一定的局限性。

a）基于 Transceiver 的单拍传输分割方案

基于 Transceiver 的单拍传输分割方案的核心技术是基于 Transceiver 的单拍传输 TDM IP。先进的基

于 Transceiver 的单拍传输 TDM IP 具有如下特点：

● 实现 FPGA 之间周期精确的信号单拍传输。

● 可选择的管脚时分复用比范围广，为每通道 32 倍至 8192 倍。

● 可选择的 Transceiver 线速率范围宽，为 10gbps 至 26gbps。

● 原始模式 TX 端到 RX 端 GT 的延迟低，约为 25ns。

● 实现基于 CRC 的帧错误检测，检测畸形 / 损坏的数据包。

● 带心跳监控的自动链接管理功能，重新连接时会重新协商。

使用基于 Transceiver 的单拍传输 TDM IP 结合自动化分割工具，可以让用户的大规模设计快速的分片，实现跨机架的单拍传输。基于 Transceiver 的单拍传输 TDM IP 的管脚复用比和用户系统速率曲线如图 4 所示。

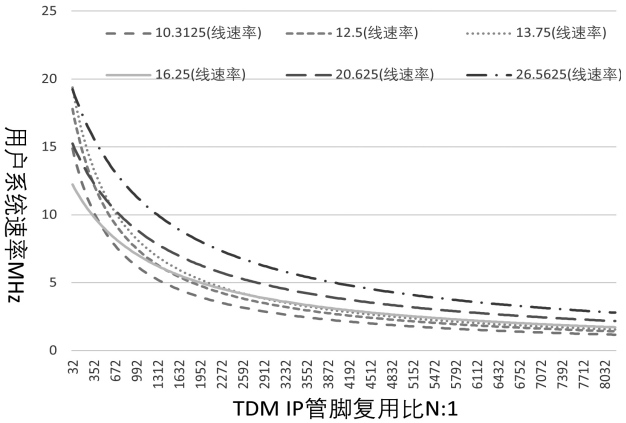

图 4 基于高速 Transceiver 的单拍传输 TDM IP 管脚复用比及系统速率对应表

从图 4 可以看到，当基于 Transceiver 的单拍传输 TDM IP 工作在 26Gbps 的线速率时，可以达到系统性能最优。此时：

● 当时分复用比设为较低的 512:1 时，此时 1 根 4-lane 的 Transceiver 线缆可以传 2,048 根信号，系统速率最高可以工作在 14MHz 左右。

● 当时分复用比设为最高的 8192:1 时，此时 1 根 4-lane 的线缆可以传 32,768 根信号，系统速率最高也可以工作在 2MHz 以上。

从用户的大规模设计性能需求来看，基于 Transceiver 的单拍传输 TDM IP 的管脚复用方案结合自动化分割工具已经可以满足大部分的用户性能需求。

b）基于 Transceiver 的总线协议分割方案

如果用户的设计在分割之后需要运行在几十 MHz 到 100MHz 以上，则使用基于 Transceiver 的单拍传输 TDM 方案无法满足用户的性能需求。这时候用户可以使用基于 Transceiver 的总线协议分割方案来达到更高的系统性能目标。先进的基于 Transceiver 的总线协议分割 IP 能够以最小的开销和延迟将 AXI 流量映射到高速收发器，它具有如下特点：

● 支持 AXI-stream、AXI-MM 及 Side-Band 信号。

● 允许基于每条传输通道的 AMBA 配置。

● 能够在同一条传输通道同时支持 AXIS、AXI-MM 和 Side-Band 信号。

● AXI 主 / 从频率可达 250Mhz。

● 支持的高速 Transceiver 的线速率可达 26Gbps。

● Transceiver IP 的延迟低于 400ns。

在基于 Transceiver 的总线协议分割 IP 中，内部用户接口支持 AXIS，AXI4 和 AXIMM 接口，外部的收发器接口支持主流 FPGA 厂商的各类收发器接口。基于 Transceiver 的总线协议分割 IP 核中的每个通道（lane）支持 AXIS，AXI4 和 AXIMM 接口所允许的任意位宽。对于收发器的数据吞吐率（Data Throughput）由多个参数所决定：当吞吐率低于线速率（Line Rate）时，吞吐率是由时钟频率与输入位宽（Input Width）所决定的。当吞吐率接近线速率时，吞吐率则是由线速率以及 FIFO width 所决定的。

使用基于 Transceiver 的总线协议分割 IP 结合基于 AXI 协议分割的半自动化工具，可以使得用户的基于 AXIS/AXI4 和 AXIMM 总线协议分割的设计的运行速率提高到几十 MHz 到 100MHz 以上。

c）基于 Transceiver 的单拍传输 TDM 分割及总

线协议分割相结合的软件自动化分割方案。

从大部分的大规模 SoC 设计分割需求来看，设计分割节点往往一部分是切割在总线上，一部分是切割在普通互连线上。

● 切割在总线上的节点的数据传速率及带宽要求较高，对数据传输的延时相对低，适用于基于 Transceiver 的总线协议分割方案。

● 切割在普通互连线上的节点的数据传输带宽要求相对低，对数据传输的延时要求高，适用于基于 Transceiver 的单拍传输 TDM 分割方案。

为了结合基于 Transceiver 的总线协议分割方案及单拍传输 TDM 分割方案各自的优势，满足大规模 SoC 设计不同设计分割节点的数据传输带宽和数据传输延时的需求，一些自动化设计分割软件可以自动适配用户设计的分割节点适用类型，在用户的设计中自动插入基于 Transceiver 的总线协议分割及单拍传输 TDM 分割 IP 以帮助用户获得最优的设计分割结果和性能。

2.3 高速接口及先进 Memory 控制器 IP 验证方案

大规模复杂 SoC 设计用到的高速接口 IP 在 FPGA 原型验证阶段，受到 FPGA 工艺和结构的现限制，往往只能运行在一个很低的速度，无法直接对接外部的实际接口，典型的如 PCIe Gen3/Gen4 接口。这时，用户在进行 SoC 原型验证时就需要有相应的高速接口降速桥方案才能和外部的实际接口对接。

2.3.1 高速接口降速桥

高速接口降速桥 IP 的目标是在速度有限的 SoC 设计 FPGA 原型和硬件仿真系统中进行基于实际接口的测试。降速桥具有如下特点：

● 降速桥只会降低接口的性能，不会影响到高速接口的协议行为和功能。

● 降速桥在软件驱动及操作系统层面对于用户是透明的。

● 降速桥具有良好的验证场景适用性，用户可以在仿真环境，FPGA 原型及硬件仿真加速器下方便的使用降速桥和用户的设计对接和测试。用户可以方便从一种验证场景切换到另一种验证场景进行交叉验证比对，譬如从 FPGA 原型验证场景切换到仿真验证场景进行接口信号的调试。

降速桥 IP 主要有两种形式：

● 一种是基于硬件的降速桥，这种降速桥对降速桥 IP 对接的原型验证环境的要求比较严格，通常一家公司开发的硬件降速桥 IP 只能对接该公司自己的原型验证或硬件仿真系统环境。另外，基于硬件的降速桥 IP 的成本往往比较高。

● 一种是基于软核 IP 的降速桥，这种降速桥对对接的原型验证环境的要求比较宽松，可以灵活的映射到用户自建或者其他家通用的原型验证环境中。此外，基于软核 IP 的降速桥的成本相比硬件降速桥会比较低廉一些。

在进行 PCIe Gen3/Gen4 接口验证时，不少客户会选择使用 PCIe 软核 IP 降速桥在自研或商用的原型验证平台上将用户低速 PCIe 控制器与真实的本地 Host PCIe 接口实现对接与映射。PCIe 软核 IP 降速桥的基本架构如下图所示。PCIe 上行端口（上行端口）和下游端口（DN-Port）是独立耦合的，适配器通过它初始化物理层并处理数据链接层。事务层数据包（TLP）在这两个端口之间转发。降速桥采用两个时钟域：本机域和被测设计（DUT）域。本机域以全速 PIPE 速率运行，与真实系统通信。DUT 域以降速的 PIPE 速率运行，降速桥通过该速率与 DUT 连接。DUT 域的运行速率最低可以为本机域的 1/64。除了基本功能外，适配器还可以记录调试状态，包括 LTSSM、数据包令牌等。

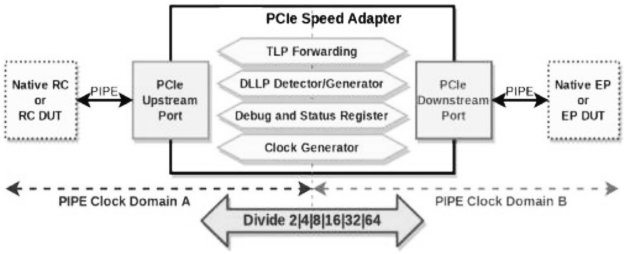

图 5 PCIe 降速桥 IP

使用基于软核 IP 的高速接口降速桥方案，可以更广泛的适配用户自有或商用的原型验证平台，帮助高密原型验证系统用户以低成本，高效的方式在 FPGA 原型系统里实现高速接口 IP 与真实接口的对接与测试。

2.3.2 Memory 控制器 IP 适配器

另外一个让 SoC 设计用户比较头疼的问题就是先进的 Memory 控制器 IP 在 FPGA 原型系统的验证，典型的如 LPDDR4/5，HBM2E/3 控制器 IP。这些控制器 IP 的标准比较新，主流 FPGA 供应商无法提供这些控制器 IP 的 PHY 解决方案，使得这些控制器 IP 无法在 FPGA 原型验证系统中运行起来。面对这个难题：

● 有些客户选择不验证 SoC 设计中用到的先进Memory 控制器 IP，将先进的 Memory 控制器 IP 替换成主流 FPGA 供应商能提供 PHY 解决方案的 DDR3 或 DDR4 控制器。这样做，FPGA 原型系统验证的难度是降低了，但是也因此改变了 SoC 设计的架构和行为，使得原型验证的准确性降低，为用户的 SoC 设计带来了额外的风险。

● 还有些客户选择自行开发先进 Memory 控制器 IP 在 FPGA 里的 PHY 接口，这样做可以保证原型验证的准确性，但是开发难度较高，开发时间较长，为用户的项目进度带来了不确定性。并且用户为了验证先进 Memory 控制器 IP 开发的 FPGA PHY 接口设计本身也有一个如何验证正确性的问题。

为了帮助高密原型验证用户解决先进的 Memory 控制器 IP 在 FPGA 原型系统的验证问题，一些商用原型验证解决方案供应商开发了基于 DFI 接口的 Memory 控制器 IP 适配器，可以将用户的 LPDDR4/HBM 等先进 Memory 控制器 IP 通过 DFI 接口转接到 FPGA 厂商的 DDR 控制器上，将用户在高密原型验证系统上通过 LPDDR4/HBM 控制器进行的 LPDDR4/HBM 存储器读写操作转变为了对高密原型验证系统上外接的 DDR 存储器的读写。图 6 是一个 DFI 接口的 LPDDR4 Memory 控制器 IP 适配器架构图。

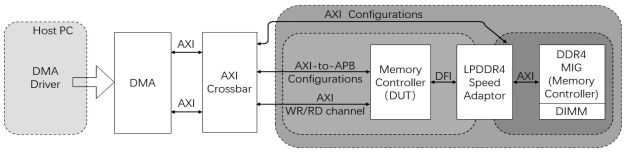

图 6 基于 DFI 接口的 LPDDR4 Memory 控制器 IP 适配器架构图

图 6 中 LPDDR4 Memory 写入和读取 pattern 是在 Host PC 中生成的，并由 DMA 驱动程序通过 PCIe 通道发送到 FPGA 上的 DMA。DMA 数据交互被转换为 AXI 流量用以配置 LPDDR4 DFI 适配器并配置和读写 LPDDR4 Memory 控制器（DUT）。LPDDR4 Memory 控制器再发送相应的 DFI 命令和数据传输到 LPDDR4 DFI 适配器。当 LPDDR4 DFI 适配器接收到写或读命令时，它将这些命令转换为 AXI 数据交互并传递到 DDR4 控制器，最终将数据写入 DDR4 存储器进行保存，或将数据从 DDR4 存储器读取出来，传送回 Host PC。

采用基于 DFI 的 Memory 控制器 IP 适配器方案，高密原型验证系统用户得以解决在 FPGA 原型系统里实现先进的 Memory 控制器 IP 在 FPGA 原型系统的验证难题。

3 高密原型验证解决方案实例

用户 A 的设计规模非常庞大，需要用 144 颗市面上最大颗逻辑容量的 FPGA 才能搭建出一套原型验证系统，急需解决的是超大规模 SoC 原型系统的设计编译耗时太长及全局时钟和复位同步的问题。在采用了国微思尔芯提供的设计分割工具 PPRO–CT 之后，用户在 1 个月时间内完成了整个超大规模原型验证系统的 RTL 级分割，设计并行综合，设计分割及增量编译的设计编译流程。在上板验证阶段，用户采用国微思尔芯提供的基于多个全局时钟模块级联的全局时钟同步系统之后，用户方便的实现了整个原型验证系统的多路全局时钟及复位的同步及远程控制。该解决方案帮助用户缩短了近 3 个月的 SoC 原型系统的验证时间。

（下转第 69 页）

外市场应用的需求。

6 结束语

通过以上技术方案，能够对国产化硬币识别产品的识别准确性和兼容性进行有效提升，满足金融设备关键部件自主可控的需求，整体系统方案成本适中，性能稳定，利于在国内外进行产品推广。

参考文献

[1] 冯军营. AVR 单片机在硬币识别系统中的应用研究[J]. 科技传播. 2013(2):194–195
[2] 濮阳槟. 电涡流传感器在硬币识别系统中的应用[J]. 电子世界. 2012(8):65
[3] 杨景昱. LDC1000 传感器在硬币识别中的应用研究[J]. 学术探讨. 2014(7):62–63
[4] 胡朝晖,张秀彬,秦证. 基于二级互补算法的硬币鉴别系统[J]. 微型机与应用. 2012(11):38–40
[5] 刘艺柱，杨瑞兰. 采用滑动平均滤波法提高硬币识别准确率的研究[J]. 制造业自动化. 2010，32(1):42–44
[6] 刘艺柱，周小川. 硬币识别器传感线圈参数设计及改进[J]. 磁性材料及器件. 2010,41(3):57–60

作者简介

胡鹏路,硕士、高级工程师。主要研究方向:智能设备。

上接第 55 页

用户 B 在设计中有用到先进的 LPDDR4 Memory 控制器，但是由于用户在原型验证阶段没有基于 FPGA 的 LPDDR4 物理层 IP 与 LPDDR4 Memory 控制器对接，用户 B 无法在 FPGA 原型系统上验证设计中的 LPDDR4 Memory 控制器。在采用了国微思尔芯提供的基于 DFI 接口的 Memory 控制器 IP 适配器方案之后，用户很快将对 LPDDR4 Memory 控制器的读写操作通过 DFI 接口转接到了 FPGA 厂商提供的 DDR4 Memory 控制器上，顺利的完成了设计中 LPDDR4 Memory 控制器的读写验证。

4 总结

随着 SoC 设计规模的快速膨胀，越来越多的用户在做大规模 SoC 原型验证时会遇到全局时钟复位同步，大规模设计分割，高速接口及先进 Memory 控制器 IP 验证等关键困难。针对这些困难，一些领先的原型验证系统方案提供商，如国微思尔芯，提供了一系列成熟可重用的通用高密原型验证系统产品和解决方案来解决这些困难，以帮助客户完成大规模复杂 SoC 的原型验证，降低项目风险。

接下来我们会在下篇中和大家探讨在做大规模 SoC 原型验证时遇到的大规模设计调试，系统部署与组网检测以及多用户多平台管理等方面的调整，并提出相应的解决方案来帮助用户应对这些挑战，缩短 SoC 的原型验证周期。

参考文献

[1] https://www.s2ceda.com/products/prodigy–logic–matrix
[2] http://www.avery–design.com/products/simaccel–fpga–accelerated–verification/
[3] High–Speed Transceiver Pin Multiplexing IP v2.0 LogiCORE IP Product Guide from Xilinx Inc

作者简介

吴滔,原型验证领域资深技术专家，现任国微思尔芯副总裁，主管国微思尔芯 IP 开发部和应用工程部。吴滔先生自 04 年加入思尔芯，拥有近 20 年的原型验证系统及 EDA 领域从业经验，一直专注于使用 FPGA 原型系统及辅助 EDA 工具来加速 ASIC/SoC 设计与验证的领域。