

高密原型验证系统解决方案(下篇)

吴滔,林锐鹏

上海国微思尔芯技术股份有限公司

0 引言

我们在上篇中和大家探讨了用户在进行大规模复杂 SoC 设计原型验证时在全局时钟及复位同步,大规模设计分割以及高速接口与先进 Memory 控制器 IP 验证等方面遇到的关键困难,并提出了相应的解决方案帮助用户来克服这些困难。接下来我们会和用户探讨在大规模复杂 SoC 设计原型验证时用户常常会面临的大规模设计调试,系统部署与组网检测以及多用户多平台管理的挑战,并提出相应解决方案,来帮助用户应对这些挑战,缩短 SoC 的原型验证周期。

1 大规模复杂 SoC 设计原型验证中遇到的挑战

1.1 大规模设计调试挑战

大规模复杂 SoC 设计原型验证过程中,需要确保设计中的故障是可调试的,最大程度减少开发过程中的调试时间。当用户把设计下载到 FPGA 以后,第一次运行总会遇到各种失败情况。原因有可能是 FPGA 原型组网不正确,设计本身有问题或者是设计编译过程中引入的错误(如设计分割和管脚时复用导致的时序错误)。用户需要一个好的测试方法来确定硬件是否运行正确,所有设计中的功能模块

是否功能正常。那么,就需要外部逻辑分析仪或内在逻辑分析仪来侦测故障所在。目前,大多数用户都习惯于使用 FPGA 厂商提供的 FPGA 内嵌逻辑分析仪来进行 FPGA 原型验证系统上板测试阶段的调试。然而用 FPGA 厂商内嵌逻辑分析仪进行大规模 SoC 原型验证系统的调试也存在着不少限制。比如:

1) 内嵌逻辑分析仪的逻辑实现需要占用用户的 FPGA 逻辑资源,当用户的 FPGA 逻辑资源比较紧张时,内嵌逻辑分析仪的资源占用会影响用户设计的布线及时序,导致用户设计无法完成布线或者时序异常。

2) 内嵌逻辑分析仪的调试数据存储需要占用 FPGA 的内部存储器资源。然后 FPGA 的内部存储器资源十分有限,当用户需要进行深度调试数据存储时,FPGA 的内部存储器资源就会不够用。

3) 虽然某些 FPGA 厂商提供的内嵌逻辑分析仪支持多 FPGA 级联调试,但是当待测试设计的触发条件需要靠来自多颗 FPGA 的触发信号进行复杂逻辑运算来实现时,FPGA 厂商内嵌逻辑分析仪无法实现该功能。

4) 当用户想要调试的信号不在内嵌逻辑分析仪调试探针的信号列表的时候,用户往往需要重新运行一次设计编译流程来修改内嵌逻辑分析仪的调试探针信号。这类操作非常耗时,而且有可能因为设计重新编译而造成设计的时序发生变化,导致原来

出现的设计 bug 无法复现的情况。

5) 内嵌逻辑分析仪的调试探针信号数量有限, 当用户想要看更多的调试探针信号, 甚至是需要设计寄存器及内部存储器全可视调试功能时, FPGA 厂商提供的内嵌逻辑分析仪无法支持。

由于 FPGA 厂商内嵌逻辑分析仪的以上限制, 用户需要有更强大灵活的系统级调试方案来满足大规模复杂 SoC 设计在进行原型验证时的调试需求。

1.2 系统部署与组网检测挑战

由于大规模复杂 SoC 设计规模庞大, 即便使用业界最大规模的 FPGA, 搭建一个大规模 SoC 设计原型往往需要数十颗甚至是上百颗 FPGA, 为了满足如此复杂的原型验证系统搭建要求, 用户需要解决下面一些问题:

1) 首先大规模原型验证系统组网的部署方案就是一个问题。实验室的桌面一般只适合部署设计规模为几台 FPGA 原型验证平台组成的原型验证系统。当用户的设计规模上升到几十颗 FPGA 或更大规模时, 实验室桌面将无法满足不同用户的部署需求。需要有专门的空间及易于扩展的环境来进行大规模及大批量的原型验证系统的部署。此外, 部署环境的供电, 散热, 温度, 湿度, 防尘甚至安全性也需要有相应解决方案。

2) 用于大规模原型验证系统组网的互连线缆长度也是一个难题。普通的基于铜线传输的线缆长度有限, 最长只有 2 米左右, 仅能用于同一台或者物理距离靠近的两台原型验证平台组网, 无法适用于物理距离较远的原型验证平台组网的需求场景。而大规模 SoC 设计往往有多台甚至数十台原型验证平台组成, 如何实现距离较远的原型验证平台组网?

3) 在大规模 SoC 原型系统组网完成之后, 用户怎么样对复杂庞大的组网进行检测也是一个难题。有时候一套复杂的原型验证系统仅仅因为一根线缆没有插紧就可能导致整套原型验证系统工作不稳定, 而这种问题, 往往要耗用户大量的时间才能排查出来。

1.3 多用户多平台管理挑战

大规模复杂 SoC 设计原型验证项目团队往往包含有硬件, 逻辑, 软件, 测试等多组成员, 这些项目组的多组成员有时会分散在不同的地区甚至是国家, 各组成员需要能够方便的通过远程访问大量原型验证平台的方式来协同工作, 并合理分配原型验证平台访问时间窗口以充分利用原型验证平台资源。对多个用户多个平台进行原型验证平台资源管理和调度是一个复杂的工程。需要有一个完善的多用户多平台的远程管理机制把这些资源有效的管理并且将资源合理分配给团队的各个成员, 避免资源冲突, 提高整个团队的工作效率, 降低项目风险。

2 面向大规模复杂 SoC 设计的高密原型验证解决方案

为了应对大规模复杂 SoC 设计在原型验证系统设计调试, 系统部署与组网检测以及多用户多平台管理等方面的挑战, 用户需要有成熟通用的高密原型验证解决方案来帮助用户节省项目成本, 提高生产力, 并缩短整个 SoC 原型验证的周期。

2.1 大规模设计调试方案

大规模设计调试是原型验证领域的一个难题, 当高密原型验证用户的待验证 SoC 设计下载到 FPGA 原型验证系统无法正常运行的时候, 用户需要有效的调试方案来排查故障的原因。

2.1.1 使用 FPGA 厂商提供的 ILA 进行多 FPGA 级联调试

目前, 大多数用户都习惯于使用 FPGA 厂商提供的 FPGA 内嵌逻辑分析仪 (ILA) 来进行 FPGA 原型验证系统上的单颗 FPGA 调试。使用 FPGA 厂商内嵌逻辑分析仪的好处是成本低, 仅需配置一个 FPGA JTAG 线缆。调试环境的设置也相对传统逻辑分析仪相对简单很多。但对于高密原型验证用户来说, 常常会面临的问题是跨 FPGA 的调试需求。由于

高密原型验证系统的规模庞大,需要调试的多个模块往往会分散在几颗不同的 FPGA 中。需要有 cross-trigger 的功能,在 1 颗 FPGA 里的调试模块的 trigger 被触发之后,能同时抓取多颗 FPGA 里调试模块的 trace 数据。一些 FPGA 厂商提供的 ILA 可以实现同一个 FPGA 内不同时钟域,不同 FPGA 之间的 ILA 内核之间的交叉触发。

2.1.2 多 FPGA 并发深度调试方案

虽然高密原型验证用户可以用 FPGA 厂商提供的 ILA 进行多 FPGA 的级联调试,但是也存在着前述的用户逻辑资源占用,存储深度,Probe 数量及多 FPGA 的 trigger 信号触发的使用场景限制。为了满足用户以上的高密原型验证系统调试场景需求,用户需要有更强大的调试方案支持对大型 SoC 设计进行全系统的调试,利用最少的 FPGA 资源,完成多颗 FPGA 的并行深度调试。理想的多 FPGA 并发深度调试方案具有如下特点:

- 利用最少的用户 FPGA 资源,以避免对其设计的影响。
- 支持用户设置大量的静态探针信号,以便设计调试。
- 允许用户对于其整个 SoC 全系统的调试。支持同时对多颗 FPGA 进行并行调试。也可以进行级联,通过 cross-trigger 功能实现更多 FPGA 的级联调试。
- 支持高速的调试数据采样率以及高容量的波形存储空间。
- 支持以太网接口传输调试数据至 Host PC 进行分析。
- 支持复杂的内嵌逻辑分析仪触发条件设置。
- 支持动态探针及状态保存及恢复功能。

a) 为了减少用户 FPGA 资源利用,多 FPGA 并行调试方案把调试所需的逻辑资源及存储资源和用户的 FPGA 资源隔离开来,使用一个单独的多 FPGA 深度调试模块来实现调试方案所需的逻辑资源和存储资源。为了传输用户 FPGA 侧的大量静态探针调试数据,又不过多占用用户 FPGA 的宝贵普

通 I/O 资源,Debug FPGA 和用户 FPGA 之间可以用 Transceiver 来进行通信,以满足高带宽调试数据的传输需求。同时 Debug FPGA 的 Transceiver 数量需要尽可能的多,以支持多颗用户 FPGA 并发调试需求。更多 FPGA 的系统并发调试则可以通过 trigger in, trigger out 级联来支持。为了支持海量用户 Debug 数据的存储,需要有外部的大容量,高带宽 DRAM 存储器。

多 FPGA 并行调试模块支持对大型设计进行全系统的调试,利用少量的用户 FPGA 收发器 IO 资源,使用 Transceiver 接口与用户 FPGA 通信能完成多颗 FPGA 的并行深度调试。将来自多颗用户 FPGA 的调试数据存储在 Debug FPGA 外接的 DDR 存储器中,最高可实现 32GB 的数据缓存,当用户 trigger 条件触发之后,多 FPGA 调试模块再将存储在 DDR 存储器的用户调试数据通过千兆以太网上传到 Host PC 中进行分析。多 FPGA 调试模块支持数组比较,组合逻辑,事件计数,以及支持脚本语言编写的复杂有限状态机的触发条件设置。多个调试模块通过级联,可以支持 16 颗或更多 FPGA 的并发调试。

b) 除了基于静态探针的多颗 FPGA 的并行深度调试功能之外,多 FPGA 调试模块还提供了基于动态探针的调试功能,帮助用户实现寄存器及内部存储器全可视,在不重新编译 FPGA 的情况下,获取所有信号波形的功能。该功能使用 FPGA 厂商自带的 IP 实现调试信号的 Readback 和 Writeback 功能。FPGA 和 Host PC 之间通过千兆以太网通信,提高数据传输速度。

多 FPGA 调试模块的动态探针功能具有如下特点:

- 支持 DFF 及 BRAM 回读及回写
- Memory 后门不影响用户时序
- 支持基于数据流的调试数据回读模式,可以通过 Transceiver 接口将读回的数据送到专用的调试 FPGA 或外部 DDR 存储器中。

结合多 FPGA 调试模块的动态探针功能,用户可以实现类似于硬件仿真加速器的设计全可视功

能,并能进行 trigger 设置和断点调试,以满足高密原型验证用户系统级调试的动态探针调试需求。

2.1.3 后门调试通道方案

用户在原型验证系统调试阶段常常需要观测或改写设计中的一些关键寄存器及内部存储空间,这时候如果有一条方便用户使用的后面调试通道,可以很大的提升用户的调试效率。用户可以远程通过以太网接口使用后面调试通道对用户设计中需要调试的关键寄存器或片上存储空间进行读写访问。后门调试通道的一端通过以太网与用户的 Host PC 对接,另一端通过 AXI 总线和用户挂接在总线上的 AXI Slave 寄存器或内部存储器对接。用户可以方便的在 Host PC 运行脚本命令以实现对接在调试通道总线上的关键寄存器及内部存储空间的读写访问。图 1 为后面调试通道读写访问流程图。

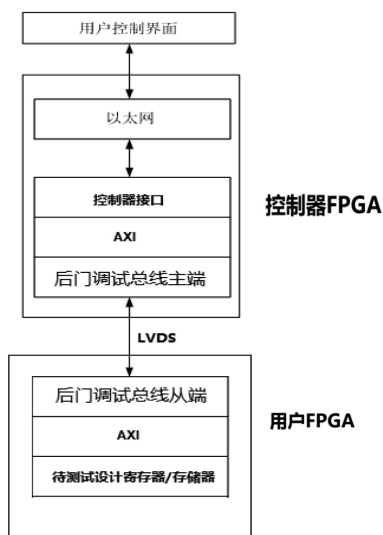


图 1 后门调试通道读写访问流程图

此外,用户也可以利用后门调试通道将设计中的程序镜像文件通过用户 FPGA 在线烧写到高密原型验证平台外接的存储器中,如:SPI Flash, NAND Flash 等。在这种使用模式下用户需要在用户 FPGA 中先运行一个接在后门调试总线从端的带有外部存储器访问控制器的设计,通过该设计完成程序镜像文件在外部存储器上的烧写,然后再将后门调试总线从端设计替换成用户自己的待测试设计,进行后续的原型系统功能调试。使用后门调试通道在线烧

写镜像文件到外部存储器的带宽能达到几十 Mbyte/秒,烧写效率比用传统的 ARM JTAG 方式进行在线烧写要高出数十倍,可以为用户节省大量的程序镜像文件烧写等待时间。

2.2 系统部署与组网检测方案

为了满足用户对大规模原型验证系统批量部署的需求,高密原型验证系统需要解决好部署方案,系统组网及组网测试这三方面的问题。

2.2.1 部署方案

a) 当用户的 SoC 设计能够用几台原型验证平台容纳时,对应的原型验证系统的部署方案相对简单,在实验室环境的桌面上即可完成。在用户搭建高密原型验证系统时,可以采用上下堆叠的方式来对少数几台原型验证平台部署,如图 2 所示。

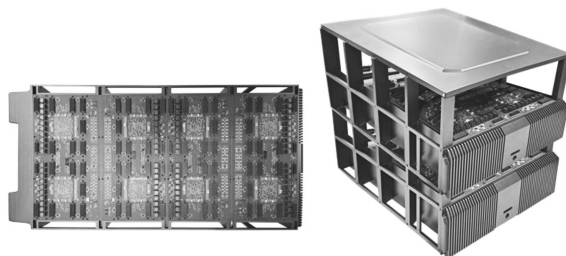


图 2 基于上下堆叠的方式部署原型验证系统

基于上下堆叠的部署方案的好处是:部署方案简单,成本低,并且节省了单套原型验证系统占用的实验室桌面空间,以方便用户在空间有限的实验室环境里部署更多的原型验证系统。

这种方式的局限是:首先用户的原型验证平台需要能支持上下堆叠。其次,受原型验证平台物理机械强度,线缆长度,原型验证系统供电及散热等因素的限制,采用上下堆叠方式进行原型验证系统部署时,一般不应超过 4 台。如果用户的设计规模大于 4 台原型验证平台的容量,这种基于上下堆叠的部署方案无法满足用户的需求。

b) 当用户的设计规模上升到几十颗 FPGA 或更大规模时,或是需要部署大批量的 FPGA 原型验证平台时,用户往往会选择机房环境,使用标准机柜来进行大规模,大批量原型验证系统的部署。

● 首先, 机房环境的容纳空间足够大, 一个 42U, 19 英寸的标准机柜最多可以部署 10 台 4U 高度的原型验证平台。通过增加机柜数目, 用户可以方便的在一个机房内部署几百台甚至更多的原型验证平台。

● 其次, 机房环境的供电能力有保障, 每个 19 英寸标准机柜可以提供数千瓦甚至更高的供电能力。当用户有几百台原型验证平台在机房同时运行时, 不用担心因为供电不够导致原型验证系统工作不正常。

● 散热也是一个重要因素。当用户有几百台原型验证平台在同时运行时, 发热量会很大。机房环境提供了专门的散热冷风道, 配合用户的原型验证平台基于机房环境的冷风道设计, 可以实现大量原型验证平台的快速散热。

● 另外, 机房环境的温度, 湿度及防尘有专门的控制, 有专门的门禁系统和机房运维人员, 这对于高密原型验证系统的长期稳定的运行增加了多重保障。

理想的通用高密原型验证系统需要既能支持中等规模设计的基于堆叠方式的实验室部署方案, 也能支持大规模设计基于机房的部署方案。从而能够满足不同设计规模和系统部署数量的不同用户需求。

2.2.2 系统组网

大规模高密原型验证系统的 FPGA 之间需要大量的通用 IO 和 Transceiver IO 进行互连。尽可能的提供足够高的互连带宽和灵活性以满足不同用户的组网需求。通用 IO 可以采用桥接板或者基于通用 IO 线缆来实现互连。高速 Transceiver IO 需要有基于铜线和光纤的互连方式以满足不同距离的互连需求。

a) 当用户的设计规模不太大, 采用几台原型验证平台通过堆叠的方式进行组网时, 使用桥接板和基于铜线的线缆即可完成系统组网。单台原型验证平台内的 FPGA 之间可以用互连桥接板或较短的普通线缆实现 GPIO 及高速收发器 IO 的互连, 多台原

型验证平台之间用长一些的普通线缆也可实现 GPIO 及高速收发器 IO 的互连。

b) 当用户的设计规模更大, 需要有大量原型验证平台部署在多个机柜上才能完成系统组网时, 用户需要使用基于光纤的线缆完成物理距离较远的原型验证平台及跨机柜的原型验证平台之间的基于高速收发器的系统组网。物理距离较近的原型验证平台内的 FPGA 之间依旧可以用基于铜线的普通线缆实现 GPIO 及高速收发器 IO 的互连。图 3 是一个在机房环境里进行大规模 FPGA 原型验证平台 (FPP) 跨机柜组网的示意图。

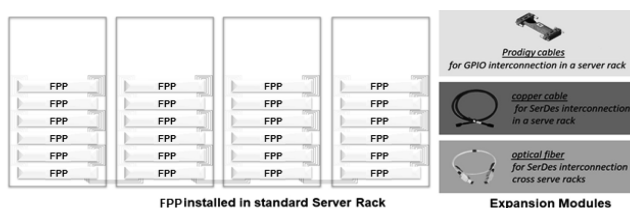


图 3 大规模原型验证系统跨机柜组网

2.2.3 组网检测

在大规模 SoC 原型系统组网完成之后, 用户还需要有高效的组网检测方案, 帮助用户检查组网过程中是否有硬件连通性及稳定性问题。原型验证系统组网根据使用的 FPGA IO 类型分为 GPIO 组网和高速收发器组网, 与之对应的组网自动测试方案也分为两种:

a) GPIO 组网检测方案

常用的 GPIO 组网检测方案是根据用户原型验证系统的拓扑结构图的互连关系, 编写专用的 FPGA 端 GPIO 组网检测设计, 再下载到用户 FPGA 中运行检测程序, 得到组网检测结果。基于这种组网检测方案的优点是检测方案简单, 只需要按照原型验证系统的拓扑结构依葫芦画瓢来进行每组互连关系的检测即可。这种方案的缺点也很明显: 这种组网检测方案不具有通用性, 每次系统组网的拓扑结构发生变化, 组网检测设计就需要做相应的调整, 导致每次组网拓扑结构发生变化, 用户都要花费较长的时间更新 FPGA 端组网检测设计。

更先进的 GPIO 组网检测方案采用连接器自动扫网

的方式进行 GPIO 组网检测,相比常用的 GPIO 组网检测方案具有更好的通用性。采用基于自动扫网的 GPIO 组网检测方案,即使用户的 GPIO 组网拓扑结构发生变化,通用的 FPGA 端 GPIO 组网检测设计也不需要做任何更改,用户可以快速知道 GPIO 组网拓扑结构变化后的原型验证系统中所有 GPIO 连接器之间的互联情况,对于出现故障的互连通道可以精确定位。并可以自动生成业界标准的设计约束文件(Design constraint SDC/XDC)和 HDL 顶层系统板级描述文件,方便用户快速地进行后续的设计和仿真。

b) 高速收发器组网检测方案

同样,先进的自适应高速收发器组网检测方案也具有更好的通用性。自适应高速收发器组网检测方案利用交叉线缆的特性,采用自适应的方式来进行高速收发器组网检测。采用基于自适应测试的高速收发器组网检测方案,即使用户的高速收发器组网拓扑结构发生变化,通用的 FPGA 端高速收发器组网检测设计也不需要做任何更改,用户可以快速知道拓扑结构变化后的原型验证系统中所有高速收发器连接器之间的互联情况,对于出现故障的互连通道可以精确定位。

2.3 多用户多平台管理方案

当用户的高密原型验证系统的规模不大,部署数量不多的情况,对原型验证系统的管理相对比较简单。通过公司的实验室或者 IT 管理人员可以人工设定若干个原型验证系统的使用和访问规则和权限,分配给一个设计团队的多个成员进行共享。

但是当用户的高密原型验证系统的规模进一步扩大,部署数量变得很多的情况,高密原型验证系统的管理会变得繁琐。在实际使用时,一套大规模原型验证系统往往会分时段分配给多名用户甚至多个设计团队来使用,这样更加增大了原型验证系统的管理复杂度。一些先进的原型验证解决方案供应商,开发了基于云的原型验证平台管理系统来满足大规模设计原型验证项目团队多组成员远程分配,使用和

管理庞大复杂的原型验证资源的需求。基于云的原型验证平台管理系统提供了一个基于浏览器的接口帮助用户进行多个维度的远程管理—FPGA 云资源,用户项目及多用户。可以实现大规模原型验证系统中每一个 FPGA 节点的时钟电压配置,FPGA 版本的加载,FPGA 状态的监控以及对组网的操作日志。配合管理功能,管理员还可实现原型验证系统在项目组人员之间的调度,统计高密原型验证系统的使用率及项目组成员的活跃度,测试主机的构建及平台使用状态的查询,为高密原型验证系统的高效使用提供便利。

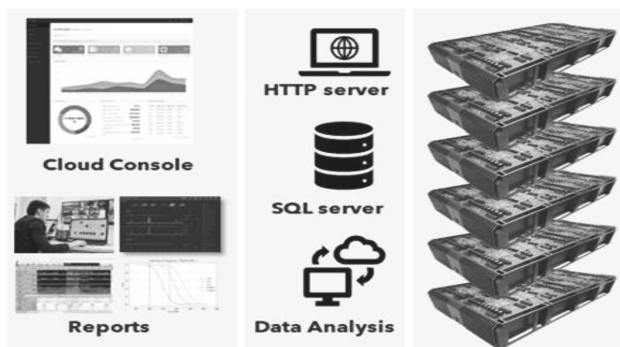


图 4 基于云的原型验证平台管理系统

基于云的原型验证平台管理系统能够帮助用户及项目经理：

- 集中资源管理：使用远程 Check-in/out 的机制将多套原型验证系统平台分配给项目组成员,避免出现多个用户争抢同一套原型验证平台资源的情况,预约和排队机制可以实现资源的充分利用。

- 状态报告：提供原型验证系统云资源使用率的分析报告及用户活跃度分析报告给项目经理。项目经理可以根据报告了解具体每天每一套原型验证平台的使用率以及每个项目组成员登陆云管理系统及使用原型验证系平台的活跃度情况,以帮助项目经理合理的分配和管理原型验证系统云资源。图 5 是一个原型验证系统云资源使用率的分析报告示例。项目经理可以一目了然的了解到项目组的 50 多台原型验证平台(图五中最左边一栏)当天在每个时间段的使用率是多少,统计精确的每 2 个小时。根据当天的原型验证平台的使用率分析,项目经理可以

Module	Day	00:00	02:00	04:00	06:00	08:00	10:00	12:00	14:00	16:00	18:00	20:00	22:00	24:00
1	13.51%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
2	21.11%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
3	13.09%	2.50%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
4	30.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
5	22.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
6	22.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
7	13.51%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
8	27.18%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
9	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
10	37.00%	17.50%	20.00%	17.50%	20.00%	21.67%	17.50%	17.50%	17.50%	17.50%	17.50%	17.50%	17.50%	17.50%
11	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
12	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
13	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
14	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
15	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
16	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
17	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
18	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
19	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
20	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
21	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
22	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
23	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
24	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
25	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
26	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
27	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
28	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
29	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
30	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
31	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
32	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
33	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
34	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
35	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
36	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
37	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
38	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
39	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
40	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
41	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
42	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
43	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
44	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
45	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
46	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
47	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
48	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
49	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
50	22.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

图5 原型验证系统资源使用率分析及报告

更合理的分配第2天的原型验证系统云资源给项目组成员。

● 多设计实例监控:用户可以在云管理系统上设定设计条件监控功能。等云管理系统监控到设定的设计条件满足之后,会自动发电子邮件给监控人员,监控人员则可以及时的远程对设计进行调整。

3 高密原型验证解决方案实例

用户A在SoC设计板级调试阶段对待测试设计的可调试性要求特别高,在进行板级调试时,需要能在一个调试窗口同时观测到多个FPGA的调试数据,并且希望能抓取尽可能深的调试数据。由于FPGA厂商提供的内嵌逻辑分析仪无法满足用户A的多FPGA并行调试及调试数据深度抓取的需求,用户A在SoC设计验证阶段花费了大量的调试时间和人力,但收效甚微。在采用了国微思尔芯提供的MDM Pro调试工具之后,用户能够在一个调试窗口同时观测到8个FPGA的调试数据,并实现了高达32GB的调试数据深度抓取,大大减轻了用户A在设计验证阶段的人力成本,并将SoC设计调试时间由一个月左右缩短到了一周。

用户B在SoC设计软件开发测试阶段需要同时用到50多套用8颗FPGA组成的原型验证系统,用来给上百位软件工程师进行软件功能并行开发及

测试用。这50多套原型验证系统的搭建,测试和使用管理对用户来说是一个费力不讨好的工作。在部署了国微思尔芯提供的Neuro系统之后,用户B可以高效快速的对多套原型验证系统进行系统搭建,组网测试和管理,项目经理可以从Neuro系统方便了解到每套原型验证系统的使用率,每个软件工程师的测试时间及测试记录,使得用户B在软件开发测试阶段的原型验证系统利用率大大提高,为公司节省了近40%原型验证系统资源,提高了生产力。

4 总结

针对用户在大规模复杂SoC验证过程中遇到的大规模设计系统调试,系统部署与组网检测,以及多用户多平台管理等诸多挑战,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应对这些挑战以帮助用户节省项目成本,提高生产力,并缩短整个SoC原型验证的周期。CC

参考文献

- [1] <https://www.s2ceda.com/en/product-prototyping>
- [2] <https://www.s2ceda.com/en/product-prototyping-lx>
- [3] <https://www.s2ceda.com/en/product-veloud>

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。