

用 TestStand 构建通讯产品并行测试平台

An Instrument Parallel Testing Platform for Telecom Products Based on TestStand

作者：于建林 刘行

职务：软件工程师

公司：深圳市中兴通讯股份有限公司康讯研究所

应用领域：产品测试

挑战：使用成熟的测试策略实现通讯产品单板功能的并行测试方案

应用方案：利用NI TestStand 模块化、易用性的特点，再借助PXI总线的的系统结构来构筑通讯产品单板并行测试平台

使用的产品：Measurement Studio、Teststand2.01

介绍

提高测试效率进而提高测试设备的使用性是当前测试仪器发展和设计的研究前沿。本文分析了利用测试管理环境TestStand 2.0功能强大的内嵌式同步工具用来规划和管理并行测试系统，从而达到测试量和设备利用率之间的最佳平衡的实现方法，同时利用TestStand API定制并行测试用户界面，建立了一个并行测试平台，将其运用于通讯产品测试领域。

TestStand

National Instruments公司的TestStand是一种可立即执行的测试管理环境，用于组织、控制和执行自动化模型建立、认证和工厂测试。TestStand的功能完全由客户定义。为满足特定需求，用户可自行对其进行功能修改和改进，例如定义操作界面，报告生成格式，或根据需要定义执行顺序等。它建立在高速、多线程执行引擎基础上，TestStand的性能可满足最严格的测试吞吐量要求。TestStand的并行测试能力可让用户运行批测试(即测量一种PCB)或异步测试(即同时对一种被测器件测量不同的功能)。线程优先功能使处理能力得到优化，

并且同时执行多种过程。其内建项目管理工具可使编程人员增加生产率。

通讯产品具有单板数量多，功能测试点也相对较多的特点。利用TestStand多线程的特性构筑并行测试平台，将明显提高测试效率。

TestStand 的同步机制

TestStand的本身的同步机制以及TestStand所有变量和属性是线程安全的这一特性使得利用它来实现多线程并行测试提供了可能性。

TestStand的同步对象有：

Lock：保证资源的独占；

Semaphore：保证资源被有限制(有限个线程)享用；

Rendezvous：用来保证多个线程互相等待以到达同一个地点；

Queue：用来将数据从一个线程传递到下一个要处理该数据的线程；

Notification：用来通知一个或多个线程某个特定的事件发生或条件满足；

Wait：等待线程或执行完成，等待时间消逝；

Batch：批次；

Thread Priority：利用它，开发人员能将正在执行的线程的优先级设置到共7个不同的级别。

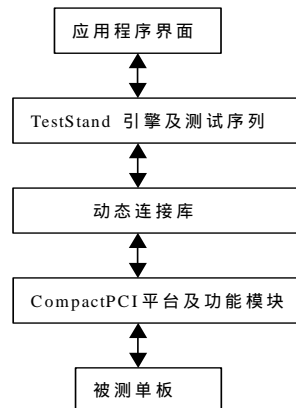
CompactPCI

CompactPCI是一种基于PCI总线的小巧而坚固的高性能总线技术。1994年PICMG(PCI工业计算机制造商

组织)提出了CompactPCI技术，该技术是嵌入式应用场合的理想选择。CompactPCI的三大核心技术：PCI局部总线，欧式插卡机械结构和气密式针孔接头。因为CompactPCI具有极高的带宽，其总线特别适用于高速数据通讯的应用。相对于其它通用测试平台而言，CompactPCI平台可靠性更高、而且具有热插拔、热切换、多处理器能力。因此，为了满足并行测试资源的需求，我们在系统构建中采用了CompactPCI平台。CompactPCI平台及其扩展功能卡给被测系统提供测试环境。目前，我们的扩展功能模块包括：数字IO模块、时钟模块、通讯模块、音频模块、总线接口模块、GPIB接口模块、开关模块等。这些功能模块卡提供了被测系统在电平，时钟，通讯，接口等各方面需要的资源，为TestStand测试并行管理环境的应用奠定了基础。

系统总体结构

充分利用TestStand测试管理环境的灵活性，我们并行测试平台的总体结构如下：



图一 系统总体结构框架

测试界面

TestStand 自带了四个 Operator Interface。考虑到通讯产品并行测试的特殊性以及TestStand 的开放性，我们自己在VC++6.0环境下定制了一个支持多块单板并行测试的测试界面。

调用TestStand 引擎，打开TestStand 生成的测试序列文件。序列文件执行时，通过调用相应的TestStand API实现对测试序列的控制如开始，中断，停止等操作。用户可以通过用户界面，实现对测

测试序列文件

测试序列文件是整个系统结构中处于核心地位。它于 TestStand 的 SEQ Editor 环境下开发。是实现并行测试策略的关键所在，也是联系测试界面和测试动态连接库的桥梁。相对于同一种单板，对应于一个测试序列。我们在单板的测试序列中，用 Thread Priority 对象来保证单板测试序列的运行优先级别。在多块单板共享一种硬件资源的时候，我们采用了 LOCK 对象来保证同一时刻对资

函数来实现的。事实上，TestStand 与所有主流测试编程环境兼容，如 NI LabVIEW, LabWindows/CVI, Measurement Studio 组件，微软 Visual Basic 和 Visual C++等，TestStand 能调用任何编译过的动态链接库 (DLLs)，ActiveX 自动化服务器，.exe 运行程序。这样，可以将测试项目模块化，多次调用。在我们的通讯产品单板并行测试平台中，我们采用的动态链接库有用 CVI 下开发出来的，也有在 Visual C++平台下开发的。这样，各模块的开发人员可以自由选择最熟悉的开发平台而不增加最后集成的难度。



图二 通讯产品单板并行测试界面

试序列中某测试项目是否测试进行定制。用户还可以设置并行测试的单板数量。测试过程中，测试界面将单板测试的各测试项目的测试情况实时报告。测试完成以后，软件将各单板的测试报告以网页的形式显示给用户。当被测单板测试失败时，将进行准确的故障定位，以方便对被测单板的维修和处理。

考虑到操作方便等原因，我们通过调用TestStand API，将TestStand 的用户管理模块以及报告文件格式、路径设置等功能也集成到了测试界面中。

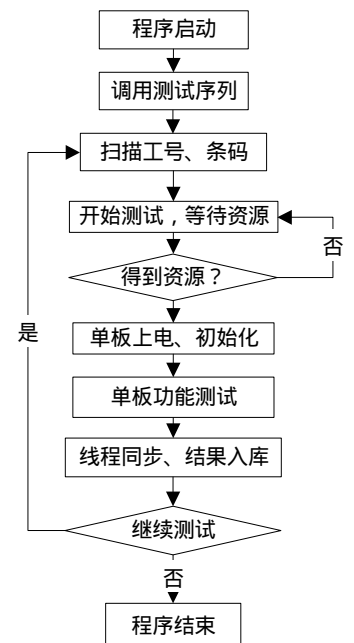
源的独占性。这样，当有访问资源冲突发生时，测试策略保证并行运行的测试序列分时占用硬件资源。我们用 Semaphore 对象来同步并行运行的多个测试线程。在各个线程之间，我们还利用了换各执行线程之间的信息。这样，即使有多个线程实例在运行，仍然可以彼此交换测试信息。

测试动态链接库

所有单板测试项目最终测试的实现是由测试序列调用的动态连接库的导出

测试流程

整个并行测试系统的测试流程如图三所示。测试界面实时检测共有n块单板在位，并显示出来。开始测试时，n个执行线程同时启动，获得相关资源后开始功能测试。当n个线程都结束后，系统将测试结果送入数据库。



图三 测试执行流程

结论

随着计算机技术、大规模集成电路技术和通讯技术的飞速发展，测试仪器技术领域发生了巨大的变化，虚拟测试仪器平台也进入了新的发展阶段。美国国家仪器公司(National Instruments)推出了测试管理环境 TestStand。为了提高测试效率，我们基于 TestStand 构建了一个并行测试平台，它具有以下优点：

1) 开放性：利用通用硬件平台构建虚拟仪器系统具有开放性，便于系统的升级和更新。

2) 易用性：丰富的软件资源和良好的人机交互图文界面使得测试系统非常易于使用。

3) 性能价格比高：相同的性能条件下开发费用和维护升级价格相对比较便宜。

4) 测试效率高：基于 TestStand 2.0 功能强大的内嵌式同步工具，我们可以在资源有限的情况下，能运行一个测试序列的多个运行线程实例，成倍地提高了测试效率。