

# 基于 FPGA 的 SoC/IP 验证平台的设计与应用

罗鹏, 许应, 封君

北京大学深圳集成微系统重点实验室, 广东 深圳 518055

**摘要:** SoC (系统芯片) 是大规模集成电路的发展趋势。SoC设计必须依靠完整的系统级验证来保证其正确性。基于FPGA的验证平台能够缩短SoC验证时间, 并提高验证工作的可靠性, 还具有可重用性。本文利用Altera公司的FPGA设计了一个基于片上总线的SoC原型验证平台, 并将VxWorks嵌入式操作系统应用于此平台, 通过软硬件协同验证的方法, 验证了平台的可靠性。该平台在CF卡及通用智能卡SoC芯片验证中得以应用。

**关键词:** SoC; 验证平台; VxWorks; FPGA

## Design and Application of FPGA Based SoC/IP Verification Platform

Luo Peng, Xu Ying, Feng Jun

Key Lab of Integrated Micro-system in Shenzhen, Guangdong China 518055

**Abstract:** SoC is the trend of VLSI. The validity of SoC depends on thorough verification. Efficient and reusable verification platform based on FPGA is useful. It can save the verification time and increase the reliability of verification. Using Altera Cyclone II FPGA, we design a configurable platform for SoC/IP verification. The real-time embedded operation system, VxWorks, is applied in this platform. The function of the whole SoC platform is validated by hardware/software co-verification. This platform has been applied in the verification of CF card and USIM card chip.

**Keywords:** SoC; Verification Platform; VxWorks; FPGA

### 1 引言

随着集成电路设计技术和半导体工艺技术的不断进步, IC设计的规模和设计复杂度飞速增长, SoC (System on Chip, 系统芯片) 成为大规模集成电路的发展趋势, 这使得传统的仿真工具已经无法完全解决验证的问题, 因而根据 SoC/IP 系统的特点提高验证的效率和验证的可重用性变得十分必要。由于 SoC 系统越来越多地涉及实时环境下的应用 (如移动通讯及视频处理等), 验证工作需要能够在接近实时频率的条件下进行才能保证其可靠性。而利用 FPGA 可以在接近实时的条件下进行验证, 并能进行软硬件协同调试[1]。

目前，FPGA（现场可编程门阵列）器件的规模迅速增长，包含大量的 IP 和存储单元，如预先被构建在 FPGA 结构中的 RISC 处理器和 DSP 模块等，以及多种 I/O 接口。这些特性使得 FPGA 成为实现 SoC 功能验证方案的理想选择。

利用 FPGA 做 SoC 系统验证和调试也面临很多挑战，比如 FPGA 的容量和大规模的 SoC 相比还是太小；高端 FPGA 的价格较高等等。针对 SoC 芯片的特点，本文采用 ARM 处理器 +FPGA 的方案，主板使用 ARM926EJS 平台，包含诸如 CPU、存储器等硬件资源，其他相对低速逻辑放在子板中，则子板使用一款中低端的 FPGA 就能实现。主板与子板间通过标准总线互连，例如 AMBA 总线。当所验证的 IP 频繁改变时，只需将包含此变化的 FPGA 重新配置。这种方法比传统的仿真系统高效可靠，同时又兼顾成本和可重用性，方便实用。为提高平台的验证效率与可重用性，平台采用 VxWorks 嵌入式操作系统，设计了便于调试的内存映射方式，工作站上运行的仿真测试向量可以直接移植到验证平台上做 FPGA 验证。此外，通过总线桥的使用，该平台可同时适用基于各种总线的 SoC 系统。

第二部分介绍了平台的硬件设计；第三部分为 VxWorks 软件平台的建立；第四部分介绍了平台的软硬件协同验证方法；第五部分为该平台在 CF 卡读卡器芯片验证中的应用；第六部分总结全文。

## 2 验证平台硬件的设计

不带微处理器的 ASIC 模块或自己设计微处理器都可以通过 FPGA 开发平台来验证，而设计中如果使用了 IP 化的微处理器，则通常有三种验证方法。

- 拿到 IP 化的微处理器的原代码和自身的设计绑定在一起进行验证，但是除了具有一定影响力的企业，一般是拿不到原代码的；
- 只对自身设计进行验证，这种方法显然不能验证自身设计和 IP 的连接以及整个系统；
- 直接使用 FPGA 厂商提供的微处理，如 Altera 的 NIOS2，这样就可以直接在 FPGA 中验证整个系统。

而在 SoC 设计中，芯片规模和复杂度进一步提高，直接使用一块高端 FPGA 成本较高，同时 SOPC 只能在一定程度上仿真 SoC，在速率等指标上受限制而不能做到硬件级别的仿真。因此我们选择利用微处理厂商提供的开发平台搭建自己的验证平台。整个验证平台由两个部分构成，一个是 ARM 公司的 Versatile 平台[2]，一个是我们要开发的 FPGA 扩展平台。Versatile 板上有一个 ARM926EJS 的内核，AHB 总线和一些外设，其最大的特点就是把片上总线引出到芯片的外部引脚，利用这条总线我们就可以扩展系统。扩展平台包括 FPGA 芯片和一些可灵活增减的模块，比如 memory，音视频接口等。FPGA 扩展平台就是为了验证用户自定义的芯片和系统，通过更新系统上的 FPGA 的代码，可以适应不同的系统的测试需要，为待测试的系统或 IP 提供验证与测试的环境。

### 2.1 验证平台的整体概述

验证平台硬件设计包括对主板 Versatile 平台的分析，以及设计基于主板 Logic Tile 板间总线的 FPGA 扩展平台。验证平台的整体框图如下：

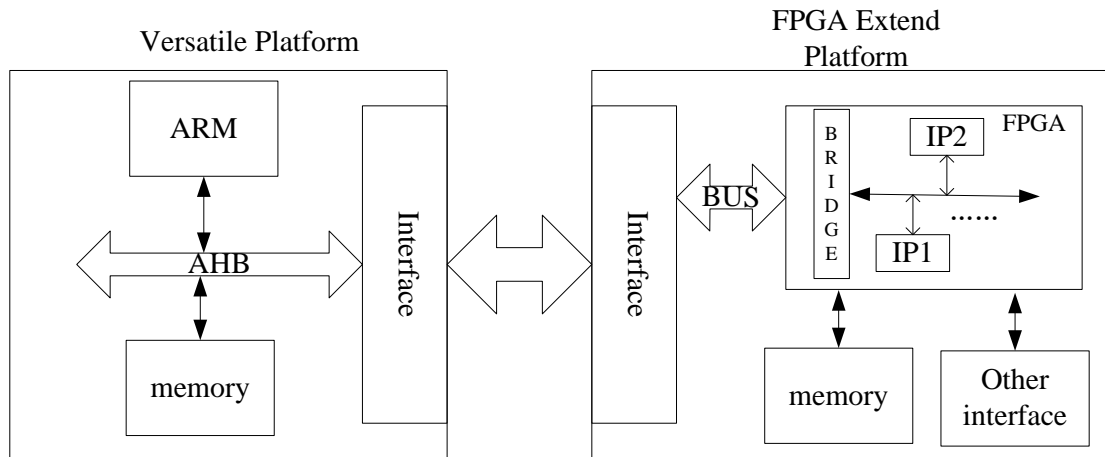


图1 验证平台的整体框图

图1中左边是ARM公司的Versatile平台，右边是我们要开发的FPGA扩展平台。此外，我们需要完成总线及总线桥软核的设计和FPGA实现。通过在FPGA上下载总线软核及待验证IP模块，就能利用扩展平台上的资源实现芯片和系统的验证。

## 2.2 Versatile平台分析

Versatile平台包括一个ARM926的硬核，AHB总线和诸多外设。我们使用Logic Title这个高速接口进行扩展，它具有丰富的资源，不仅包括AHB总线，还有DMA、UART等接口，通过这个Title扩展平台可以利用Versatile平台上的大部分资源。下图是Versatile开发板的功能模块示意图，AMBA总线为多层AHB总线加APB总线的结构。一共有6个总线控制器（master），ARM9内核指令和数据master，两个DMA master，CLCDC master及扩展到板级总线Logic Tile上的扩展master。总线从模块（slave）包括存储设备，串口，定时器等。定时器、串口等低速外设挂载在APB总线上，通过APB桥和AHB总线继而各个master相连。AHB M1，AHB M2与AHB S就是扩展子板的总线接口。

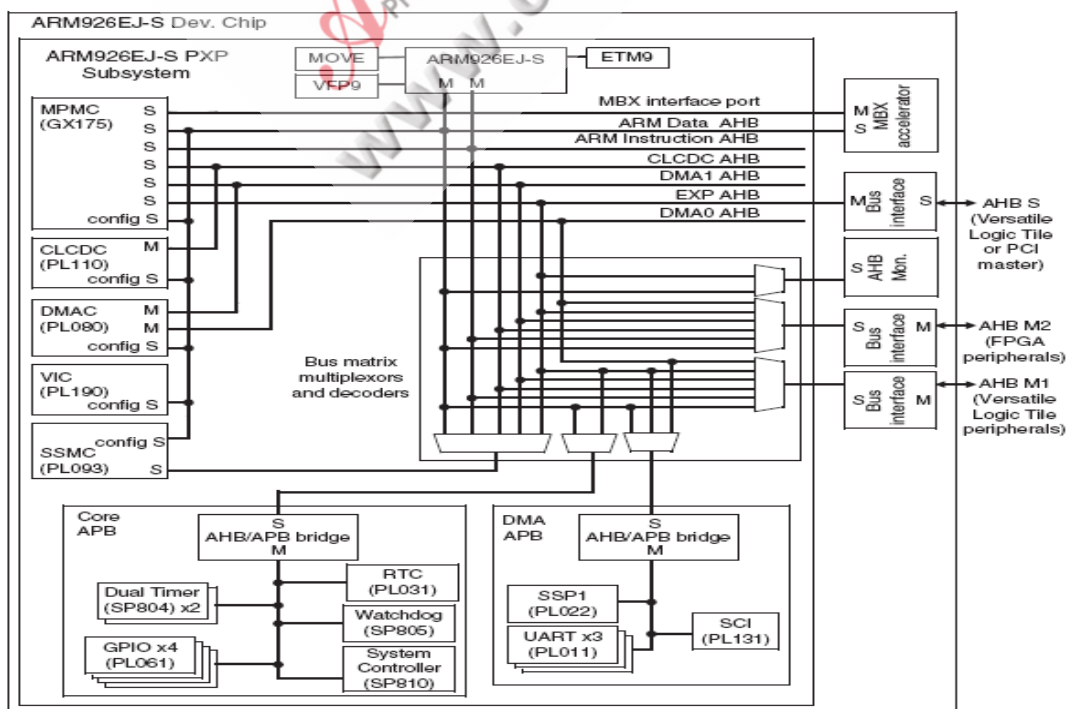


图2 Versatile开发板的功能模块示意图

## 2.3 FPGA扩展平台设计

如图 3，FPGA 扩展平台通过 Logic Tile 总线与主板连接，该总线包括 AHB 接口以及其他通用 IO 口、中断、DMA 通道等丰富的资源。该扩展平台具有灵活的时钟系统与电源系统，总线可以和主板时钟同步也可异步，电源可由主板提供也可由扩展板自己提供。外部存储器有 SRAM 与 EEPROM，在调试初期代码下载在 SRAM 中，方便反复擦写调试，最后调试成功的代码则存放在 EEPROM 中。并可以提供串口与 PC 机直接相连。

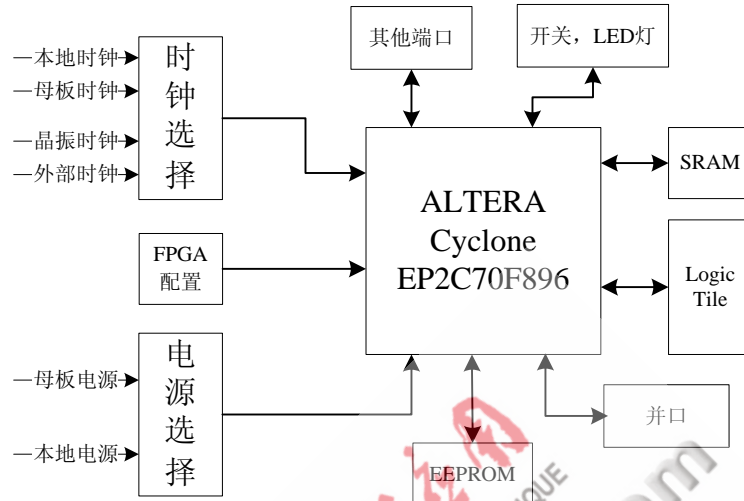


图 3 扩展平台单板结构图[3]

使用 FPGA 扩展平台的目的是为了验证用户自定义的芯片和系统，硬件设计上它应该包括：

- 电源系统，3.3V 和 5V 的电源可以通过 Title 来提供，也可以通过扩展板上自带的稳压电路提供；
- 时钟系统，时钟系统和母板使用的是同一个全局时钟，通过控制位的选择可以使用 Versatile 平台上的时钟源或是扩展板上的晶振；
- FPGA 芯片的配置，如果扩展平台比较简单可以采用配置芯片直接配置，如果系统复杂可以外加 PLD 来控制整个扩展平台和对 FPGA 配置；
- FPGA 的调试，可以将 FPGA 通过 JTAG 扫描链中进行调试，也通过电缆和 PC 主机连接进行调试；
- 总线，连接 FPGA 和 Title 上的相应信号；
- DMA、中断等信号等；
- 外设接口等。

## 3 验证平台软件的设计

前一部分我们讨论了验证平台的硬件部分，在这个硬件平台上运行的程序就是验证平台的软件部分。SoC 功能仿真与设计验证平台除了硬件设计之外，相应的软件设计也是需要考虑的重点。作为一个 SoC 芯片，在微处理器和 DSP 上运行的软件也占相当大的比例。提供良好的调试手段有利提高整体的开发效率。

软件部分的实现有两种方式，一种是编写汇编和 C 代码直接在平台上运行，此种方法适用于验证规格简单，没有并行处理的应用；另一种是先移植嵌入式操作系统到平台上，然后在操作系统的基础上用 C 编写验证代码，此种方法适用于复杂的验证项目。移植操作系统还有一个优点就是隔离上层软件和底层硬件，包括底层驱动的操作系统负责上层软件和底层硬件的通信。这样程序员就不需要关心硬件，类似于一般的 C 语言编程，编程难度大大降低。所以我们采用后一种方法，把底层硬件完全封装起来，软件设计的进度和硬件同步，而不需等硬件设计完了之后才开始，从而减少了项目验证时间。

### 3.1 Vxworks 嵌入式系统简介

VxWorks 操作系统是美国 WindRiver 公司于 1983 年设计开发的一种嵌入式实时操作系统。多年来，它以其良好的可靠性和卓越的实时性被广泛地应用在通信、军事、航空、航天等高精尖技术及对实时性要求极高的领域中。VxWorks 操作系统具有如下的特点[4]：

- 它支持高度的系统可裁剪性，可根据具体应用定制系统，使系统对资源的需求最小，利用率最高。它具有多达 1800 个功能强大的应用程序接口（API），适用方面广，可以适用于从最简单到最复杂的产品设计。
- 其微内核 wind 性能优越，具有快速多任务切换、抢先式任务调度、中断延迟小等特点，满足关键任务实时性的需求。它可靠性高，可以广泛应用于从 ABS 刹车系统到星际探索系统的关键任务。
- VxWorks 操作系统运行环境支持多种硬件平台，具有较好的兼容性，可以用于所有的流行的 CPU 平台，如 Motorola 68K、Intel/Cyrix/AMDx86、MIPS、Intel i960、ARM 等。
- 它支持应用程序的动态链接和动态下载，应用程序各模块可分别编译、下载、动态链接，方便易用。

### 3.2 Vxworks BSP 的移植

VxWorks 分离了底层硬件和上层应用程序，利用 BSP（板级支持包）对不同的硬件加以封装以提供给应用程序一致的环境，所以 VxWorks 的开发主要是 BSP 的移植。为了有较高的运行速度，我们对于 VxWorks 内核采用 kernel 加载方式运行，系统启动时读取压缩文件在存储器 SRAM 里解压，然后开始执行。因此我们需要根据硬件平台上的存储空间映射来移植一个引导程序 Bootloader。

BSP 的移植包括内存映射，中断驱动，定时器驱动，串口驱动。通常的内存映射是把编译后的镜像烧录在非易失存储器，比如 flash 中，启动后程序由 flash 复制到 RAM 中，内存重映射后程序在 RAM 中运行。由于调试阶段烧录 flash 耗费时间且具有一定风险，因此本文将镜像直接下载到 RAM 中运行，修改了 VxWorks 常用的内存映射方式，将 flash 的部分地址空间与 RAM 叠在一起，这样物理上的一段 RAM 对应了 VxWorks 中的 RAM 与 flash，程序可以在 RAM 中直接运行。中断驱动的设计关键在中断产生后如何判别正确的中断源，做出正确的中断响应。定时器驱动用于产生 VxWorks 多任务调度所需要的系统时

钟。串口驱动在系统调试初期非常有用，其关键在于理解上层文件操作如何与下层硬件操作相结合。

完成以上代码修改后，我们对源代码进行编译生成了压缩格式的内核镜像文件，其占的存储空间仅为 304KB。母板加载 VxWorks BSP 后，使用超级终端或其他串口调试软件可看到如下图 logo，则证明 VxWorks BSP 移植成功。

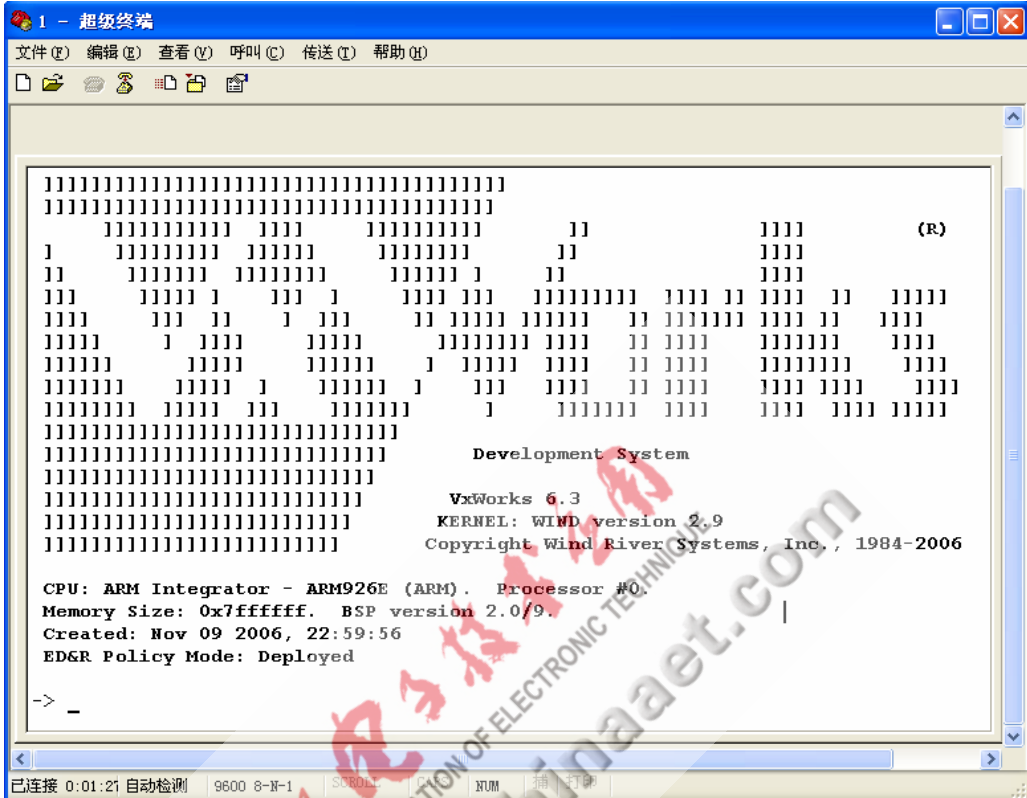


图 4 VxWorks 软件平台

### 3.3 软件平台的验证

为了对所开发的软件平台进行 debug，我们使用 Multi-ICE 仿真器，在 RealView Debug 下进行调试。软件仿真的界面如图 5 所示，处理器选择 ARM926ejs，编译内核文件后，处理器就可以通过读写存储器运行操作系统。RealView debug 还提供单步跟踪，设置断点，查看寄存器及存储器，搜索函数定位等功能帮助调试。

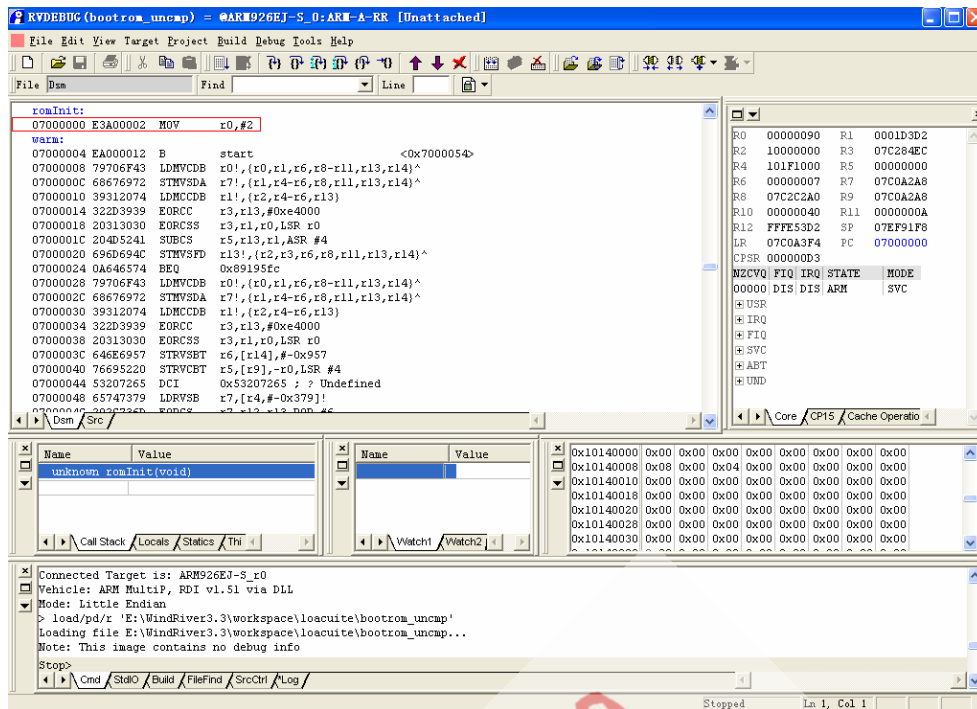


图 5 软件仿真界面

#### 4 软硬件协同验证方法

软件验证平台通过仿真器（ICE）和硬件处理器进行通讯并调试，处理器通过系统接口与下载到 FPGA 中的硬件设计通讯，并构成一个完整的芯片硬件模型[5]，如图 6 所示。

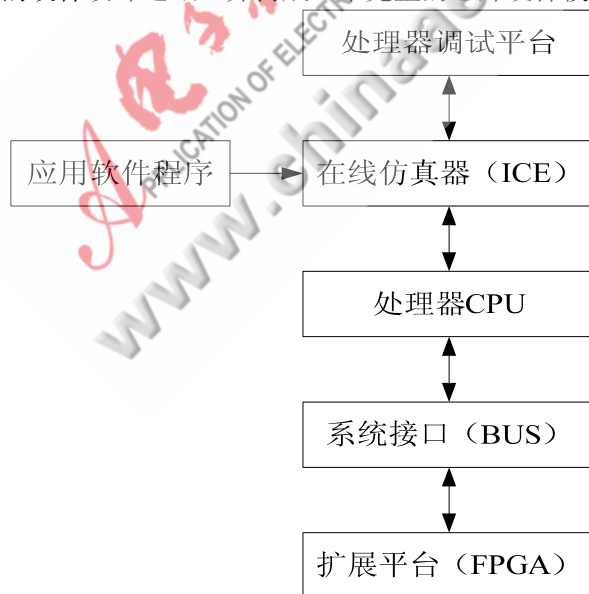


图 6 原型验证系统

基于以上的原型系统架构，通过 Multi-ice 仿真器，Realview Debug 软件开发平台可以直接控制验证平台上的 ARM9 内核，为嵌入式系统的开发与验证提供了便利，整个软硬件协同验证平台架构如图 7 所示。这样，我们就可以在该平台上完成 SoC 软硬件协同验证。

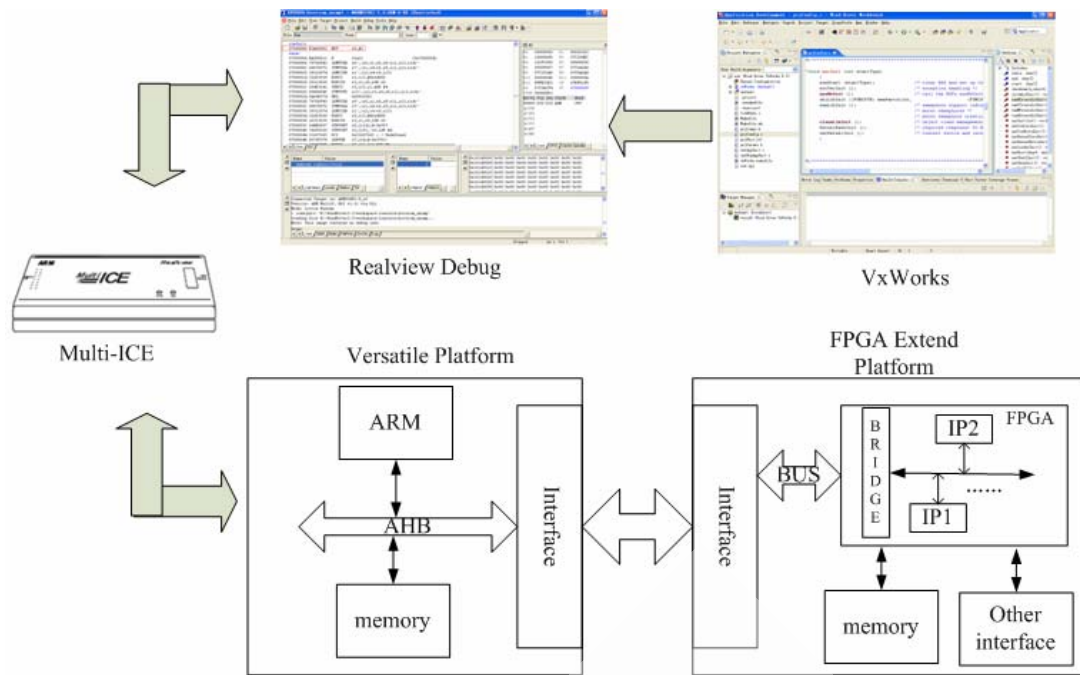


图7 软硬件协同验证平台

## 5 验证平台在 CF IP 验证中的应用

该平台已经应用于实验室多个 SOC 设计项目的验证，包括 CF 读卡器芯片、智能卡、WLAN 芯片等。以 CF (Compact Flash) 读卡器商用芯片为例[6]，根据规格书定义，它是一个带有 AHB 总线接口的 CF 卡与 Host 的接口适配器，它支持 PC card memory、PC card I/O 和 True IDE 等多种工作模式，有 169 个通过工作站上仿真的 Testcase 需要 FPGA 验证。如图 8 为验证平台结构图。在整个系统中单板部分即 CF Card Board，主要实现市场上通用的 CF 存储卡到总线的接口。单板一边通过标准的 CF 卡座和 CF 卡连接，另一边通过板上总线与 Versatile 平台连接。

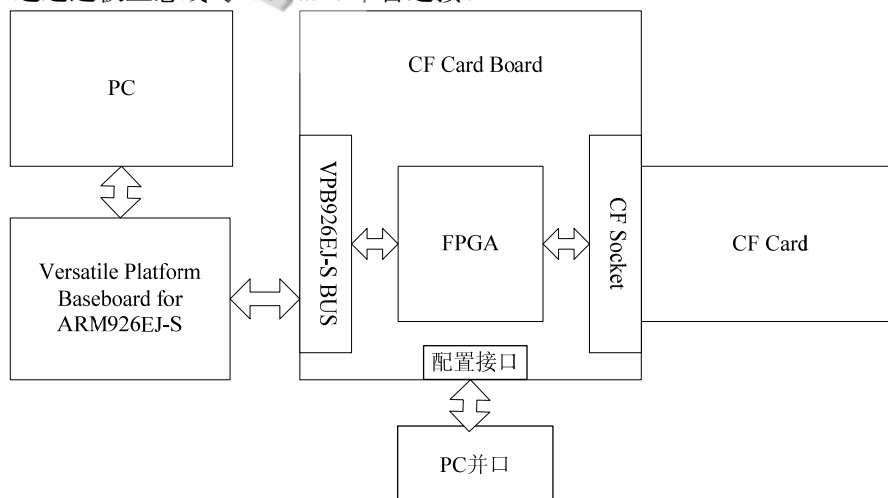


图8 CF 卡芯片验证



下图是 CF 卡芯片验证平台的详细情况，在 AHB slave 验证平台的基础上加入对 DMA 传输和中断的支持。由于控制信号经过不同的板上传输路径，有不同的延迟，所以除时钟和异步复位外信号需要经过 hclk 同步再进入 IP 模块。

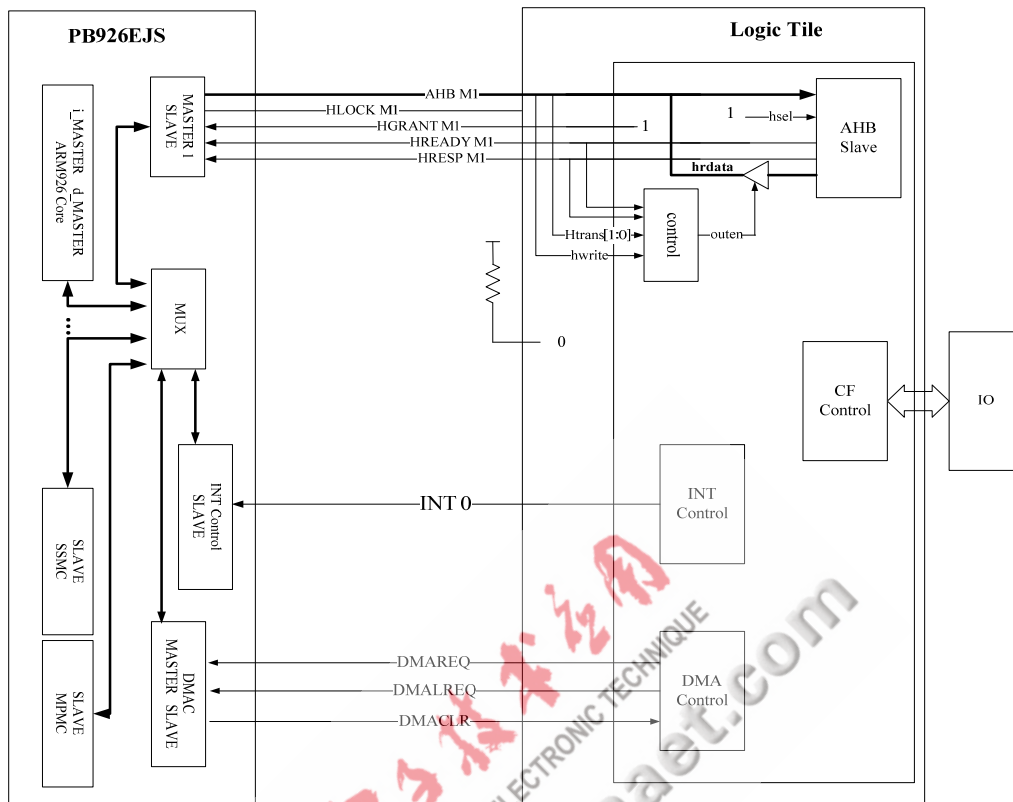


图 9 CF 卡读写器验证平台

最终的 FPGA 综合版本，最坏路径输入寄存器建立时间  $T_{su} < 5.756ns$ ；输入寄存器保持时间  $T_h > 0.297ns$ ；时钟至输出延迟时间  $T_{co} < 8.833ns$ ；时钟 cclk 最高时钟频率可达 68.98MHz，时钟 hclk 最高时钟频率可达 71.32MHz，满足规格书要求。

在以上验证平台中，所有 169 个 Testcase 均能验证，并可通过串口终端及 Quartus 的 SignalTap II 逻辑分析仪观察分析结果。为了验证对 CF 扇区读写访问功能，我们将内容确定的数据写入目标扇区中，再将写入的扇区数据读出，进行比较，验证设计的正确性。



图 10 SignalTap 上读数据的一段波形



图 11 使用读卡器读出的数据

## 6 总结

随着芯片向 SoC 的方向发展，流片费用、项目进度的要求，使得对一次流片成功率的要求不断提高。但是 SoC 需要软硬件协同仿真的特点，使得原有的仿真平台很难完成 100% 的覆盖。本文设计的验证平台可以通过量身定做扩展 FPGA 验证板，完成对不同 IP, SOC 芯片的上板仿真，对于中小型 IC 企业而言，可以节约开发成本，缩短开发周期，提高流片成功率。

## 参考文献

- [1] Prakash Rashinkar, System-on-a-chip Verification Methodology and Techniques, Kluwer Academic Publishers, 2002.
- [2] ARM Inc., Versatile Platform Baseboard for ARM926EJ-S User Guide, 2004.
- [3] ALTERA Inc., Cyclone Device Handbook, 2005.
- [4] 李忠民, ARM 嵌入式 Vxworks 实践教程, 北京航空航天大学出版社, 2006.
- [5] 林文敏, 基于 AMBA 总线的 SoC 平台的设计和验证, 计算机工程与应用, 2005, 28 期.
- [6] CompactFlash Association, CF+ and CompactFlash Specification (Rev 3.0), 2004.

原创性声明:

本人郑重声明, 此处提交的论文《基于 FPGA 的 SoC/IP 验证平台的设计与应用》, 是本人在导师指导下, 在北京大学期间进行研究工作所取得的成果。据本人所知, 论文中除已注明部分外不包含他人已发表或撰写过的研究成果。本声明的法律效果将完全由本人承担。

作者签字: 罗鹏

日期: 2008 年 8 月 26 日