

文章编号: 1001-2486(2002)04-0053-04

嵌入式微处理器的多层次可配置仿真工具*

张鲁峰, 赵文辉, 李思昆

(国防科技大学计算机学院, 湖南长沙 410073)

摘要: 为了适应嵌入式定制微处理器的开发的需要, 克服现有微处理器仿真工具的某些不足, 文中介绍了微处理器的仿真验证工具——VCPU 的设计和实现方法。同其它微处理器仿真工具相比, 该工具具有多层次, 可配置, 功能全面的特点。利用该工具, 设计者可在设计早期进行系统集成验证, 缩短设计周期并减少设计错误。该工具已经在嵌入式微处理器开发过程中得到成功应用。

关键词: 软硬件协同设计; 嵌入式微处理器; 多层次仿真工具

中图分类号: TP391 **文献标识码:** A

A Multi-Level Configurable Embedded Microprocessor Simulation Tool

ZHANG Lu-feng, ZHAO Wen-hui, LI Si-kun

(College of Computer, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: In order to meet the need of the embedded application specific microprocessors development and overcome some disadvantages of current microprocessor simulation tools, a new embedded microprocessor simulation tool — VCPU is introduced in the paper. Being different from other microprocessor simulation tools, VCPU is a flexible full-featured multi-level virtual prototyping. The designer can use VCPU to verify the embedded system, both hardware and software in the early design stage. Errors can be found and corrected before fabrication, thus the design cycle is reduced. VCPU has been used in embedded microprocessor development successfully.

Key words: HW/SW co-design; embedded microprocessor; multi-level simulation tool

嵌入式系统的核心是微处理器。应用的发展对于嵌入式微处理器的功能和性能提出了更高的要求, 许多微处理器面向应用进行“定制设计”, SOC (System On a Chip) 技术的兴起使微处理器不再以芯片, 而是以芯核 (Core) 的方式存在于系统中, 使得这种定制设计成为可能。另一方面, 产品更新周期缩短, 对于目前的微处理器设计验证提出了新的要求。

仿真目前仍然是微处理器验证的主要手段。国内外都开展了微处理器模拟器方面的研究, 开发出多种工具, 比较典型的有: MIPS 结构的 SPIM, ARM 公司的 ARMulator, 以及在微处理器体系结构研究中得到广泛应用的 DLX 模拟器和 SimpleScalar 工具^[4]。国内的北京大学微处理器研发组为支持其 JB-Core 微处理器核, 设计了 JBCODES 软硬件协同设计平台^[1], 东北大学开发了面向微控制器的嵌入式软件虚拟开发环境^[2]。

目前的微处理器模拟工具在一些方面还不适应嵌入式微处理器核的开发和应用:

- 大部分微处理器模拟器面向单一处理器结构, 可配置性不强。SOC 中的嵌入式微处理器核往往会根据应用需求改变指令集和硬件配置。比如, ARM 公司的 ARM9 系列处理器, 就可以根据需要扩充 DSP 指令成为 ARM9E, 扩充 Java 解码器成为 ARM9J, 扩充指令压缩部件成为 ARM9T, 扩展 Cache 和存储器管理部件 (MMU) 内存保护部件成为 ARM920/922/940。以上这些扩充还可以进行组合。这就要求嵌入式微处理器的模拟器也应该能够进行配置, 以仿真各种体系结构的变化。

- 大多数微处理器模拟工具面向单一抽象层次。而在系统开发和微处理器设计过程中需要侧重不同、抽象层次不同的微处理器仿真模型。

- 嵌入式系统设计中功耗设计非常重要, 目前的模拟器对于低功耗设计缺乏支持。

* 收稿日期: 2002-01-30
基金项目: 863 SOC 重大专项预启动项目资助 (863-SOC-Y-2-1)
作者简介: 张鲁峰 (1974-), 男, 博士生。

· 微处理器模拟器对嵌入式软件的开发支持不足。

针对现有微处理器模拟工具的不足,面向嵌入式微处理器核 TS-1 的开发,作者设计实现了嵌入式微处理器仿真工具—VCPU (Virtual CPU)。VCPU 能够支持一系列微处理器的体系结构设计、硬件设计验证以及嵌入式软件开发。

1 VCPU 的设计实现

1.1 多层次的处理器建模

不同设计人员对于微处理器仿真原型的细化层次和仿真的侧重点有不同的需求。在进行功能验证时,仅需要微处理器原型能够正确运行程序,在性能分析时还要得到程序运行速度,而微处理器体系结构设计者关心分支预测成功率、Cache 命中率等参数。为了适应这些要求,VCPU 被设计为一个多层次的仿真工具集,各个层次的工具是渐进细化的关系,具有一致性。目前 VCPU 包含 4 个层次的模拟器:

(1) 功能模拟器。功能模拟的执行速度最快,实际上是将被仿真程序转换为仿真宿主机的程序运行,以检验被模拟的测试程序是否符合行为描述。功能模拟器在性能模拟器上简化得到,可以得到程序的踪迹、指令种类和条数以及被仿真程序运行结果。

(2) 性能模拟器。比功能模拟器增加了对于嵌入式微处理器指令类型、数目和节拍以及程序访存的统计。可以得到程序的性能、运行指令踪迹、访存次数。并对于微处理器内部的部件进行建模和仿真分析,能准确记录模拟时钟,流水线的占用情况等。

(3) 硬件描述语言模型仿真器。基于硬件描述语言的仿真,由可综合的硬件描述语言 Verilog HDL 描述构成,需 Verilog HDL 模拟器进行仿真。除了运行程序外,同时对于硬件设计进行了验证。

(4) FPGA 硬件原型。提供了最快的仿真速度和对目标系统尽可能真实的仿真。

这 4 个层次的模拟工具,可以充分满足嵌入式软件开发过程的需要。

1.2 性能模拟器的结构和仿真过程

在 VCPU 的结构的设计原则是同真实处理器结构相对应。整个模拟工具包含了对于存储器、指令/数据 Cache、译码控制部件、流水线和浮点部件以及特殊功能部件的仿真。模拟器结构最上层是被模拟的测试程序代码。在模拟器的核心,是由指令执行模块、分支预测器、处理器执行资源管理、Cache 模块、寄存器管理模块、测试程序的执行文件加载模块、存储器管理、调试器及执行结果统计模块等构成的运行系统,执行模块实施指令语义的正确模拟。

被仿真的目标程序首先调入 VCPU 的虚拟存储器中,仿真器就利用指令地址(即 PC)从虚拟存储器中得到指令,从程序入口点开始逐条指令地模拟系统输入的执行过程,VCPU 以程序驱动方式运行,直接解释运行程序。这种方法比踪迹驱动的模式更忠实于程序的动态行为。仿真的同时,统计包括指令执行条数、目标程序运行时间、寄存器、内存、Cache、指令流水部件等在内的系统资源使用情况。根据这些性能评价数据,可了解被仿真程序在当前微处理器体系结构上运行的性能。VCPU 还提供了方便的统计接口,添加统计量非常容易。为了帮助进行低功耗设计,性能模拟器包含了对功耗的统计。

一组真实程序仿真实验表明,性能模拟器同国外商用仿真工具的仿真结果(程序性能)相差在 5% 以内。

1.3 VCPU 的配置方法

进行体系结构模拟过程中,用户可以根据嵌入式微处理器的体系结构设计需求,对 VCPU 的某些参数进行配置。VCPU 采用模块化的构造方法,提供了基于配置文件的灵活配置方式,通过配置可以改变功能部件的数量和种类(浮点部件、乘法器、分支预测部件的策略),Cache 的层次、组织方式(关联),增加特别的指令等。在配置文件中还可以指定模拟结果的统计方法,如起始指令,执行的指令条数等。关于各种不同部件的配置放置在多个配置文件中,程序模拟器、性能模拟器和体系结构模

型模拟器共享同样的配置文件。

1.4 多层次存储器的仿真和系统调用仿真

Cache 在高端的嵌入式微处理器中广泛使用，对于嵌入式系统多层次的存储器系统的仿真是 VCPU 的重要组成部分。VCPU 在仿真宿主机上建立虚拟存储器仿真真实处理器的主存，其组织也同真实存储器一样，采用段页式的组织方法，划分为指令段、数据段、堆栈空间，存储器的最小单位是可定义大小的存储页。VCPU 工具提供了在虚拟存储器中动态分配存储空间的系统调用，

VCPU 的 Cache 仿真部分可以通过配置，改变仿真 Cache 的容量、相关、访问策略等。同时可提供关于 Cache 的详细仿真数据。

真实程序往往会有一些在微处理器模拟器内部无法仿真的部分。比如向屏幕输出字符或者从串口读取数据等系统调用。被仿真的目标程序需通过宿主平台的程序完成系统调用等代码。当执行到这些调用时，被仿真的程序暂时挂起，模拟器调用宿主主机上的相应回调函数，而参数传递通过预定的寄存器和内存区域进行。执行完成后将结果写回到虚拟存储器中，然后恢复现场，继续执行挂起的被仿真程序。

2 基于硬件描述语言和 FPGA 的微处理器仿真原型

硬件描述语言原型是在嵌入式系统硬件的 Verilog HDL 描述开发完成后，利用性能模拟器的虚拟内存管理、系统调用仿真和软件调试功能配合 Verilog HDL 语言模拟器实现的模拟验证环境。目前使用 Sun 工作站下的 NC - Verilog 工具完成 Verilog 语言仿真。仿真过程启动后，软件调试工具加载目标程序至虚拟存储器，Verilog 模拟器加载微处理器的 Verilog 描述。Verilog 模拟器仿真微处理器核利用地址和读写控制信号向虚存发送读取指令和数据的命令。指令和数据从虚拟内存读出以后，经过模拟向量生成工具，转换为可以被 Verilog 仿真器接受的激励向量，发送到 Verilog 仿真器。而 Verilog 核执行仿真激励产生的数据，则写回虚存。如图 1 所示。

通过这种协同仿真的方式，可以自动的产生大量近似实际应用的模拟向量，在比较逼真的环境下验证微处理器核的 Verilog 描述。并且 Verilog 模拟器的仿真的结果可以通过软件调试工具来检查，可同微处理器的软模型模拟的结果相对照。而单纯的 Verilog 模拟器只能仿真部分系统，需手工产生模拟向量，通过查看波形来检查模拟结果。显然，该协同仿真工具大大提高了验证效率。在 TS - 1 微处理器核的开发过程中，设计者利用该仿真器高效地寻找了许多微处理器核的设计错误，避免了设计反复。

硬件描述语言仿真，无法解决的一个问题就是仿真精度和仿真速度的矛盾。在 Verilog 原型中，运行规模大一些的软件在速度上难以容忍。而且软模型也无法仿真某些真实系统的特性。FPGA 有接近硬件原型的执行速度，且可多次编程。随着 FPGA 规模的不断提高，FPGA 已成为验证小规模微处理器的有效手段。

常规的 FPGA 仿真系统同监控计算机的启动和通信问题较难解决，输入输出能力较弱。为了解决这个问题，在 TS - 1 处理器 FPGA 原型板的设计中，采用了双处理器交替运作的方式，在原型板上除

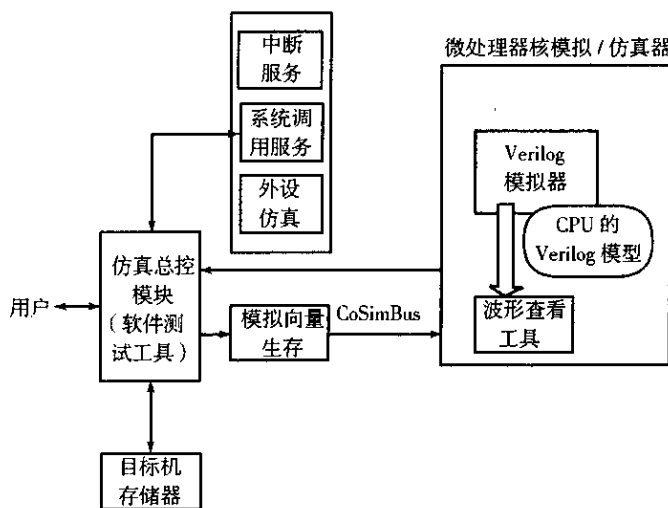


图 1 硬件描述语言模型仿真器

Fig.1 HDL Model Simulator

了FPGA实现的待验证微处理器核外，增设了一个微控制器，与FPGA交替运行，用来从监控计算机导入程序和返回仿真结果。这种基于FPGA的微处理器原型有结构简单，控制可靠，同主机通信能力强的优点。

3 基于VCPU的软件调试环境

在VCPU工具和GCC编译器的基础上，我们构造了嵌入式软件开发环境^[4]，生产出实际的硬件之前，就可对系统中的软件进行充分地模拟调试及性能分析。如图2所示。

软件编码完成后，利用宿主平台上的交叉编译器将源程序编译为面向特定体系结构的目标程序。目前在gcc编译器上进行修改得到交叉编译器。

加载编译后的目标程序进VCPU的虚存并进入目标程序调试状态，试用VCPU的调试模块，用户可以像在真实微处理器上一样，进行程序的运行调试。调试功能包括：断点、单步执行、查看/修改变量和寄存器、反汇编指令等。

为了方便使用，VCPU工具以及软件调试环境包含了一个友好的图形用户界面，该界面基于Linux窗口环境KDE实现。为了使设计者能直观地了解微处理器流水线的动态运行过程，VCPU还包含一个在DLXView工具^[6]的基础上开发的流水线运行状态查看工具。

4 结束语

介绍了嵌入式微处理器仿真工具VCPU的设计及实现，该原型工具的主要优点是：支持对微处理器的多层次建模仿真，可配置性强，对于软件开发过程提供了良好支持。

VCPU工具要配合嵌入式微处理器软核的开发，增加对于超标量技术等高性能微处理器技术的仿真。功耗成为嵌入式系统设计中重要的设计因素，要在VCPU工具中完善对于功耗估计的支持。

参考文献：

[1] 崔光佐，程旭等. 面向处理器的系统级模拟、仿真及调试技术——基于软硬件协同设计的新方法[J]. 计算机研究与发展, 2001(3).

[2] 郭晓东，刘积仁，余克清等. 嵌入式系统虚拟开发环境的设计与实现[J]. 计算机研究与发展, 2000, 37(4): 413.

[3] 张鲁峰，李思昆，赵文辉等. 基于虚拟原型的软硬件联合设计方法研究[J]. 计算机工程与科学, 2001(A).

[4] 熊志辉，张鲁峰，赵文辉等. 嵌入式微处理器体系结构模拟与软件调试工具的设计与实现[C]. 中国计算机学会第七届计算机工程与工艺学术年会, 2001.

[5] Doug Burger, Todd M Austin. The SimpleScalar Toolset Version 2. 0[R]. University of Wisconsin - Madison, 1997.

[6] Renato J Figueiredo. A Network - Computing Infrastructure for Tool Experimentation Applied to Computer Architecture Education[C]. Workshop on Computer Architecture Education at the 27th International Symposium on Computer Architecture, 2000.

[7] Chang T C, Rudnick E M. A design verification environment for pipelined microprocessors[C]. Proc. Int. Workshop on Microprocessor Test and Verification, 1998.

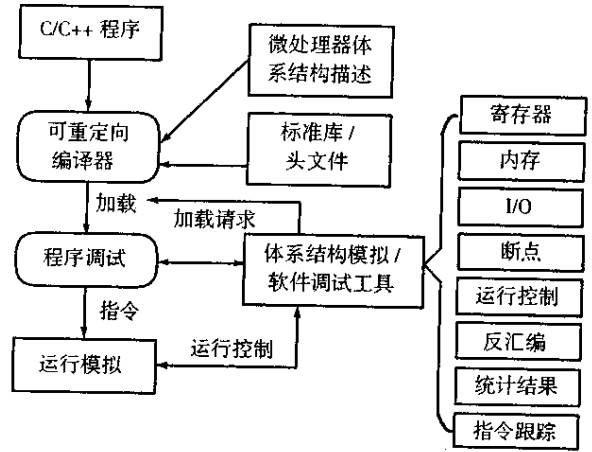


图2 嵌入式软件开发过程

Fig.2 Process of embedded software development

