第36卷 第20期 计 算 机 工 程 2010年10月
Vol.36 No.20 Computer Engineering October 2010
·博士论文· 文章编号：1000—3428(2010)20—0019—03 文献标识码：A 中图分类号：TP368.1

基于VPM和随机激励驱动机制的处理器核仿真建模

许 彤[1,2]，张仕健[1,2]，吕 涛[2,3]
(1. 中国科学院计算技术研究所微处理器中心，北京 100190；2. 中国科学院研究生院，北京 100049；
3. 中国科学院计算技术研究所计算机系统结构重点实验室，北京 100190)

摘 要：为提高处理器核仿真模型的效率，提出处理器IP仿真模型由内核行为模型和总线功能模型(BFM)两部分构成，是SOC设计验证的关键部分。本文基于SimpleScalar架构对龙芯1号处理器进行虚拟处理器模型VPM行为建模，IPC平均误差为2.3%，%，速度达到每秒1M1 000 000条指令/秒；。提出基于可控随机事件机制实现的的总线功能模型，BFM可以，为片上系统(SoC)设计提供了激励主动生成方案和片上互连验证功能。实验结果证明，该方法对处理器IP仿真建模具有普适意义，能够被无缝融入SoC流程中。
关键词：IP仿真模型；SimpleScalar模拟器；可控随机事件；总线功能模型；龙芯1号处理器

Processor Core Simulation Modeling Based on VPM and Random- StimuliDrive n mechanismMechanism

XU Tong[1,2], ZHANG Shi-jian[1,2], LV Tao[2,3]
(1. Microprocessor Center, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China;
2. Graduate University of Chinese Academy of Sciences, Beijing 100049, China;
3. Key Laboratory of Computer System and Architecture, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China)

【Abstract】ToIn order to improve processor core simulation modeling efficiency, As key part of SOC design and verification, simulation model of processor IP cores consists of core behavior model and Bus Functional Model (BFM). A Aa virtual processor modeling methodmodel based on SimpleScalar architecture is proposed, and the model aiming at Godson-1 processor reached reaches 1 MIPS 000 000 per second with average IPC precision error of 2.3%. A controllable random- event Bus Function Model(BFM) is presented, providing active stimuli generation and on-chip bus verification function for SoC design. This Experimental result proves that the solution has broad applicability in processor core modeling and can

1 概述

根据摩尔定律，到2010年片上片内晶体管数可达20亿个，系统级芯片片上系统(SoC，System-On-Chip)设计成为芯片设计的发展方向。处理器IP核作为SOCSoC的核心控制部件或基本运算单元，其仿真模型被用于SOCSoC软硬件协同设计、硅前的仿真验证和SOCSoC快速构架评估、性能评估，是主流SOCSoC集成开发中必不可少的部分[1]。处理器核仿真模型的实现与处理器的指令集、体系结构、外部接口和运行环境等密切相关，涉及处理器体系结构、快速仿真、SOCSoC验证方法学等多个领域。如何为处理器提供快速准确的仿真模型，是对处理器IP核设计者的挑战。

龙芯1号是一款以单发射、动态流水线、AMBA接口为特征的32位RSIC处理器，支持乱序执行、LD/ST寻址模式和32位指令长度[2]，主要面向嵌入式SOCSoC应用。龙芯1号处理器IP有软核和硬核两种2种形式，仿真模型的实现是处理器IP化建模的一个重点。

1 2 常用的处理器IP仿真模型实现方法

处理器仿真模型模拟处理器的运算时序、内部功能行为以及其与外部设备通信的接口行为。衡量处理器IP仿真模型有三3个标准：精度、，速度和SOCSoC流程兼容性。高精度的仿真模型需要对处理器的细节进行模拟，必然带来速度的降低。处理器仿真模型可分为以计算和流水线控制调度(以下简称运算)为中心的处理器内核行为模型和以协议处理为中心的总线功能模型(BFM, Bus Functional Model, BFM)两部分。实现处理器仿真模型运算与片上通信的解耦，有利于降低SOCSoC集成难度。

·(1)处理器内核行为建模

处理器内核模型可以分为主机软件执行、指令集模拟(ISS，Instruction Set Simulation, ISS)、周期精确模型拟(CAM, Cycle Accurate Modeling, CAM)和虚拟处理器模型(VPM，Virtual Processor Model, VPM)四4类，其中，后三3种适用于SOCSoC方法学。ISS仅保证指令功能的正确性，速度可达每秒500K 000条－~ 7MIPS7 000 000条指令，但ISS精度不高，仅适用于SOCSoC早期的软件功能评估。CAM方法通常用使用HDL，精度高但速度慢，如ARM的Realview CAM模型速度为每秒100K 000条－~200K200 000条指令/秒。VPM方法将目标处理器的结构映射到一个非周期精确的可执行模型，模型与处理器微体系结构一致，但需要对细节进行必要的简化，以实现精度与速度的平衡。目前静态流水

基金项目：国家自然科学基金资助项目杰出青年基金项目(60325205)；国家“863”计划基金资助重点项目(2002AA110010)；中科院计算所知识创新课题基金资助项目(20056230)
作者简介：许 彤(1975－)，男，副研究员、博士研究生，副研究员，主研方向：处理器设计、，IC设计与，嵌入式应用技术；张仕健，博士；、吕 涛，博士
收稿日期：20082010-03-031 **E-mail**：xutong@ict.ac.cn

线的 ARM7 处理器中,，VPM 的方法速度可以达到每秒 1M1 000 000 条-~ 2M2 000 000 条指令/秒，但对于动态流水处理器而言，其复杂调度导致 VPM 速度变慢。VPM 通常精度误差通常为 5%-%~40%%，模拟速度为每秒 200K 000 条－~500K 000 条指令/秒。

·(2)总线功能建模

BFM 模拟处理器 IP 核的片上通信行为，其实现具体通常有无定时模拟、周期近似估计、事务级模型(Transactional Level Model, TLM)和引脚精确模型(Pin Accurate Model, PAM)四 4 种实现方式。其中，TLM 抽象级别适中，能够提供较好的精度和速度的较好平衡，通常用于复杂互连的 SOCSoC 验证中；引脚精确模型可以保证模型的 cycle 级一致性，可以满足 SOCSoC 集成时协议精确处理的要求。

2 3 基于 SimpleScalar 的处理器 IP 内核行为建模

处理器 IP 的内核行为建模和处理器设计中的分析建模有一定的相似性，二者两者都专注于处理器本身的行为。本文考虑将处理器分析建模的执行驱动方法应用于 IP 内核行为建模当中。在执行驱动方法中，指令解释独立于模型实现，采用快速的功能级指令模拟器生成程序踪迹，如实反映微体系结构并具有较高的精度和速度，适合于适用于处理器核的 VPM 建模。

SimpleScalar 模拟器[3]是执行驱动模拟的一个典型工具集，包括快速功能模拟器、Cache 层次评估模拟器和超标量多发射处理器微体系结构模拟器等，为处理器行为模拟提供了较完善的参考体系。SimpleScalar 的典型速度约为每秒 300K300 000 条指令/秒，其对结构描述的精确度不高够，如 sim-outorder 的 IPC (Instruction Per Cycle)与实际设计平均误差相差可达 36%。

VPM 建模优化是和与处理器的微体系结构紧密相关的。本文针对龙芯 1 号处理器核，基于 SimpleScalar 工具集的框架进行仿真建模，将龙芯 1 号微结构映射到 SimpleScalar 架构上。首先基于处理器微结构对 SimpleScalar 进行校准工作，随后采用一系列优化方法对模拟器进行性能优化以提高速度和精度，最后进行模型的验证工作。

23.1 结构校准

SimpleScalar 中 sim-outorder 是 SimplesScalar 中最复杂、最详细的模拟器，它将指令重命名到 RUU 队列上，模拟了由取指、分派、发射执行、写回和提交五 5 级动态流水线组成的结构。龙芯 1 号处理器的动态流水结构和 sim-outorder 有较大的相似性，IP 模型映射选择了 ss-mips 版本的 sim-outorder 作为构建基础。

在取指阶段，龙芯 1 号处理器取指部件从 I-cacheCache 中取回一条指令，送入 IR 寄存器中，并根据预测机制猜测下一条指令的 PC。在译码及重命名阶段，译码部件将 IR 中的指令译码成内部统一格式，将目标寄存器重命名到操作队列上，进入操作队列。在发射阶段，从操作队列中取出一个操作，读取该操作寄存器结果，进入相应的保留站。在执行写回阶段，从保留站中取出源操作数都准备好的指令，进入相应的功能部件中执行，并将执行结果通过结果总线写入操作队列和保留站。在提交阶段，将操作队列中已写回的指令结果提交给结果构寄存器，如果指令发生了例外，则进行例外处理。

龙芯 1 号 VPM 模型根据上述结构对 sim-outorder 架构做了调整，并基于 sim-outorder 中的 RUU 队列、重命名机制、事件队列及 cacheCache 结构等并进行了针对性的修改:

(1)根据龙芯 1 号的功能部件对功能校准 VPM 进行校准，确定每个部件的发射延时、操作延时。

(2)在每个功能部件中增加保留站结构，指令从 RUU 队列发射到对应的保留站中等待，并取消了 sim-outorder 中的 ready queue 队列。

(3)在提交阶段检查分支指令是否猜测错误，如果发生错误，则清空流水线中的后续指令，再从正确的路径上重新取指。

(4)实现对访存系统的阻塞式关键字优先访问。访存指令和其他指令统一放在 RUU 队列中，取消 sim-outorder 中的 load/store 队列。只有当 RUU 队列中 store 指令是第一第 1 条指令时，store 指令才发射执行，并将结果写入内存。

(5)实现对龙芯 1 号 branch 指令延迟槽的支持。

VPM 实现中把相应部件参数化并生成配置文件，控制处理器的功能部件的数量、功能单元及保留站大小等属性。这种实现保证了 VPM 和与龙芯 1 号 IP 核可配置的一致性，也一定程度地实现了 VPM 的一定程度的灵活性。

23.2 性能优化

由于 sim-outorder 的速度不能满足要求，因此龙芯 1 号 IP 模型在实现中针对 sim-outorder 的特点，在程序设计上进行了改进，以提高了 VPM 的运行速度。主要采用以下方法:

（1）(1)采用时间复杂度低的新算法减少程序开销。在 TLB 和 Cache 的设计中，使用软件缓存和 hash 查找，降低查找的时间复杂度。

（2）(2)利用反向调用流水线减少数据复制。在每个时钟的上升沿时，本级流水从上级流水的输出锁存器中得到数据进行本级流水的运算，时钟结束前将结果输出到本级的输出锁存器中，数据在流水线中前进一个流水级，本级流水的数据在被写入之前即可被下级流水读取，这样大大降低了数据复制带来的开销。

（3）(3)减少内存的动态分配，缩小由此带来的时间延迟。系统初始化时分配足够的内存，保存在一个链表表示的资源池中，运行时在资源池中进行资源的分配和回收，。在 VPM 的保留站设计中大量应用了这一方案。

（4）(4)SMARTS 加速:。本文参考了 Wunderlich 的严格统计采样方法(SMARTS)[4]，利用统计原理和实际程序运行表现出来的统计特性对误差进行估计和控制。对于给定的误差范围和置信度，如果采样结果不能满足要求，SMARTS 自动调整为可能满足要求的采样频率。根据采样频率的多少不同，模拟器的执行速度会有所不同。

23.3 模型验证

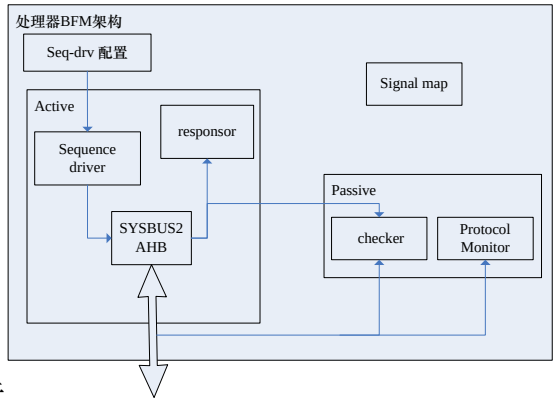
为保证运行速度，龙芯 1 号 VPM 没有完全再现处理器的所有细节。为确保模型误差可控，我们本文采取了一个系统化的验证流程来保证模型的准确性。VPM 以龙芯 1 号 RTL 模型为参考基准，在 RTL 和 VPM 中运行相同的典型测试程序，比较程序的运行结果以及流水线的执行时间。为了比较流水线，验证中在原始 RTL 和 VPM 中增加了大量检查点(checkpoint)，运行中对 checkpoint 也进行检查。

在模型设计过程中，首先比较采用手工编写的反映结构特征的测试程序，然后比较这些程序在模拟器和 RTL 模型中执行的流水情况。如果差异比较大，分析并找出差异的原因，修改模拟器后再进行比较，直到它们之间的差异减少减小到可以接受的地步程度。经过不断的验证迭代，模型的准确度

得到保证。最终模型的正确性采用商用测试集验证，以保证验证结果的可信性可信度。

3　4　基于可控随机事件驱动机制的 BFM 架构

在现有的解决方案中，处理器核 BFM 往往仅对总线协议行为进行模拟，而忽略了处理器本身又是总线主 master(master)的特点，因此，不能主动对 SOCSoC 其他设备进行灵活的访问，这对 SOCSoC 验证造成了不便。针对这一问题，我们本文在实现总线协议的基础上，采用随机事件驱动的方法构建 BFM，为 BFM 自动生成可控的片上访问驱动。BFM 中只需要定义互连访问的事件，并不涉及具体处理器的行为。实现中采用验证语言和 RTL 相结合的方法：协议处理采用 HDL 实现周期精确的结构，其余部分采用 verisity 公司的 e 验证语言提供带可控约束的随机测试向量。e 语言是面向对象的语言，支持带约束的随机向量生成，在此基础之上可形成面向处理器的可控随机事件驱动。



基于

图 1　可控随机事件驱动的处理器 BFM 架构

可控随机事件驱动的处理器 BFM 架构的整体架构如图 1 所示。在龙芯 1 号处理器 IP 中，接口为 AMBA AHB 主设备[5]。龙芯 1 号 BFM 主要分为 Active 模块和 Passive 模块两部分。Active 部分通过序列驱动器和协议响应器模拟处理器 cacheCache 对外的行为(称为 SYSBUS 接口)，协议处理器(SYSBUS2AHB)进行 SYSBUS 和 AHB 接口的协议转换。Seq-drv 配置和 Signal map 用于对随机驱动的控制和 e 与 RTL 接口之间的定义。Passive 部分是可选的，负责对 AHB 总线的协议完整性检测以及数据传输正确性检验。

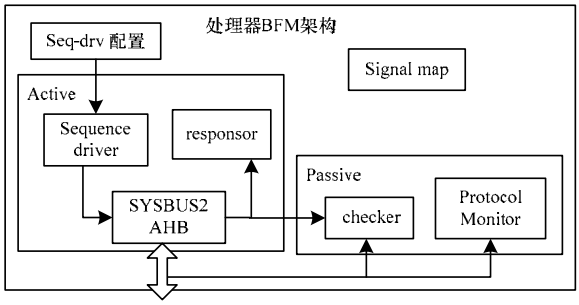


图 1　基于可控随机事件驱动的处理器 BFM 架构

verisity 工具中集成了丰富的可控制随机方式的类(class)，在序列发生器中如何采用 e 语言实现处理器的可控随机事件生成是 BFM 问题的关键。实现主要考虑两 2 点：一是事件驱动实现，二和是事件可控接口。

3(1).1　事件驱动实现

模型中事件驱动采用了面向对象的方法，根据 Cache 行为来定义事件来源。BFM 通过协议处理器模块将 Cache 行为转换为 AHB 总线行为。由于 AHB 总线事件需要考虑各种复杂情况以及 split/retry 等 AHB 协议响应的具体配置，而 Cache 接口所发出的事件只有读(read)、写(write)、重填(refill)三 3 种，因此，在 Cache 接口级定义事件能够大大简化实现方案。通过接收够虚拟 cacheCache 接口的事件序列，RTL 实现的协议处理器自动对 AHB 总线进行响应。实现中，事件序列项(item)定义为 e 语言的结构(struct)，如(表 1 图 2)。在 struct 实现中，还根据每项的特点定义了规则。

表 1　事件序列项定义

Item	内容	
操作	READ, WRITE, REFILL	
类型	UncacheCached	BYTE, HALF-WORD,TRI-BYTE, WORD,DWORD
	Cached	EIGHT-WORD
地址	Wraddr, Raddr	
数据	Dataout，, Datain	

图 2　事件序列项定义

随后在此基础上，本文定义了事件驱动器 5 个重要的的基本类(unit)和扩展类(extend)及其内部函数。，如表 2 图 3 所示。

表 2　事件驱动类定义及说明

基本类(unit)	扩展类(extend)及方向	说明
ict_godson_ sysbus_env_u	无	定义基本系统变量
ict_godson_ sysbus_seq	BASIC_OP，, DATABUS, ADDRBUS,	序列的三 3 种基本方向
	MAIN	调用其他方向
ict_godson_ sysbus_bfm_u	Ict_godson_sysbus_bfm_u (包括 read_op，, write_op，, refill_op，, fetch_item 等主要操作函数)	和与协议处理器的接口，定义了事件的信号级驱动过程
ict_godson_ sysbus_seq_drv_u	Ict_godson_sysbus_ seq_drv_u	
ict_godson_ sysbus_agent_u	Ict_godson_ sysbus_agent_u	主类，调用 driver, , bfm 和可选的 monitor

图 3　事件驱动类定义及说明

Cache 的事件操作通过高层类函数封装的调用实现，类的内部函数将 BFM 信号级的通信协议封装为事件级的通信接口。如表 2 图 3 中的 fetch_item 函数即就是一个高层调用函数，系统通过该高层函数调用具体的底层事件 read_op，, write_op, refill_op 等。

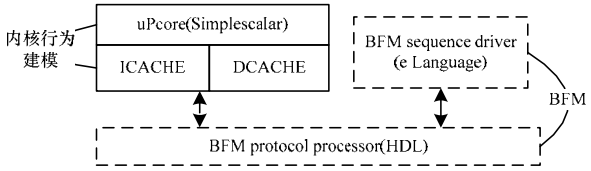
3.2　(2)事件可控接口

在按照上述方法定义了基本类和扩展类后，cacheCache 事件被逐层映射为随机激励源的控制。我们本文提供配置文件的模板，形成了最终 SOCSoC 用户看到的 BFM 可控事件接口。模板中可以通过简单的 e 语句对类扩展的方式加以约束，实现接口的扩展。通过对 BFM 的行为采用基于面向对象的方法包装，原本信号级驱动的接口成为可控随机的 Cache 事件接口。SOCSoC 开发者通过限定 Cache 操作的模式、组合、地址或数据，对 SOCSoC 地址空间自动产生大量带约束的随机访问，用于 SOCSoC 互连部分的性能验证以及外围 IP 核的总线协议验证。

4　5　实现效果

基于上述方法的龙芯 1 号处理器仿真模型的结构如图 24 所示。模型分别对处理器内核和接口进行建模，实现了内核

设计与接口的解耦。



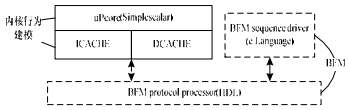


图 2　4　IP 仿真模型架构

最终对内核行为模拟验证的基准是来自 alpha21264 模拟器使用的 17 个商用典型微测试程序[6]。验证中所有程序的功能运行结果都与 RTL 一致，而 IPC 差异如图 35 所示。可以看出，其中 16 个程序的 IPC 差异在±5%%以内，只有一个程序 tst-loop-m13 的时间误差为 13.5%%，平均 IPC 差异为 2.3%%，表现出良好的一致性。tst-loop-m1 程序 3 的误差在于该程序访存相关非常复杂，引起当前 VPM 的 sim-outorder 简化架构与龙芯 1 号访存流水微结构的表现差异所致。这为模型的进一步改进指明了方向。

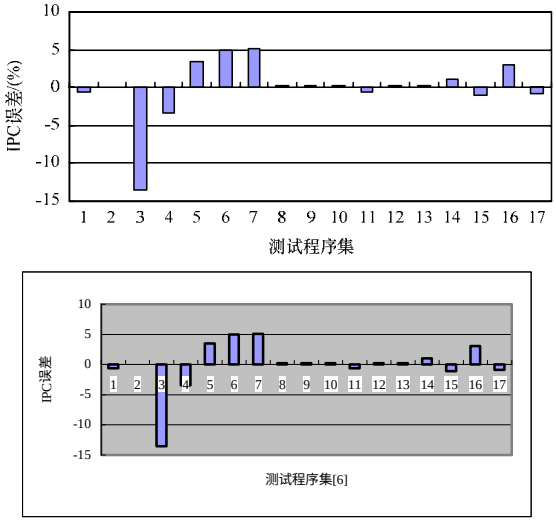


图 3　5　龙芯 1 号 IP 内核行为模型 IPC 比较结果

基于 SimpleScalar 的龙芯 1 号内核行为模型在 Intel Pentium 4 CPU 2.8 GHz 机器上的平均运行速度在约为每秒 500k500 000 条指令/秒左右，采用 SMART 加速技术后平均运行速度在　每秒 1M1 000 000 条指令/秒以上。这个速度和与静态调度处理器 ARM7 模拟器的速度相同，达到了是常见动态流水处理器模拟速度(每秒 2 000 条－~500K500 000 条指令/秒)的数倍。龙芯 1 号 VPM 模型和一般 ISS 的速度在同一量级，但在精度上却远高于 ISS。

基于 e 语言开发的可控事件驱动机制的龙芯 1 号 BFM，在以周期精确的方式模拟处理器总线 master 的通信行为的同时，能根据需要灵活地对 SOCSoC 中不同地址空间进行不同特征的访

(下转第 24 页)问，拓展了 BFM 的功能。该 BFM 被用于一款 SOCSoC 的设计当中，通过对 SOCSoC 总线地址空间的扫描访问，很快发现了某个外围 IP 的协议不兼容问题，而且随机产生的访问流量能够作为 SOCSoC 互连架构探索的激励，为该 SOCSoC 互连部分的架构验证提供了方法。这种本文方法将对处理器 IP 核的运算和通信进行了解耦，大大简化了 SOCSoC 验证难度，可以无缝地集成到现有的 SOCSoC 流程中。

56　结论及未来工作结束语

本文针对处理器 IP 核仿真模型，，本文提出了一种基于 SimpleScalar 的处理器 IP 核行为模型实现方法，和一种基于可控随机事件驱动的 BFM 实现架构，为 SOCSoC 开发提供了灵活的主动事件驱动接口，能够加快 SOCSoC 互连的验证工作。未来将一方面对内核行为模型进行访存优化，进一步提高内核模型的准确性，；一方面将并研究将两 2 个模型融合起来形成一个整体 IP 模型的方法。

参考文献

[1] Biggs J Biggs, Gibbons A Gibbons. Reference Methodology for Enabling Core Based Design[RZ],. European Synopsys User Group, March 2002.

[2] 胡伟武, 唐志敏. 龙芯 1 号处理器结构设计[J],]. 计算机学报, 2003, 26(4): 385-394.

[3] Austin T Austin, Larson E Larson, Ernst D Ernst. SimpleScalar: An Infrastructure System Modeling[J],. IEEE Computer, 2002, 35(2), Feb.: 2002. pp. 59-67.

[4] Wunderlich R E Wunderlich, Wenisch T F Wenisch, Falsafi B Falsafi, and J C Hoeet al. SMARTS: Accelerating Micro-architecture Simulation via Rigorous Statistical Sampling[JC]//, Proceeding. of the 30th Annual International Symposium on Computer Architecture. [S. l.]: ACM Press, 2003SIGARCH Comput. Archit. News 31, 2,　JuneMay 2003. pp.84-957.

[5] ARM Co., Ltd.. AMBA™ Specification (Rev 2.0)[RZ], ARM ltd.,. 1999.

[6] Desikan R Desikan, Burger D Burger, Keckler S W Keckler. Measuring Experimental Error in Microprocessor Simulation[C],]//Proc of the 28th Annual International Symposium on Computer Architecture., Toronto, Canada: [s. n.], 2001. pp.266-277.

编辑　张　帆