

嵌入式存储器的通用 SystemC 模型设计

妙 维, 李国宏, 章 琦, 汪 辉

(中国科学院 上海高等研究院, 上海 201210)

摘 要: 本文提出了一个基于 SystemC 的通用存储器模型, 可以作为构造 SoC 的通用 IP 模型. 该模型采用统一的事务级接口, 内部控制逻辑和存储器构成采用时钟精准模型, 并且提供了一系列配置参数用来反映影响性能的带宽、延时、以及仲裁机制. 该模型成功预测了一个视频解码器中共享 SRAM 所采用 bank 数量对性能的影响.

关键词: SystemC; 建模; 存储器; SoC

中图分类号: TN402

文献标识码: A

文章编号: 1000-7180(2013)07-0033-04

A Generic Embedded Memory Model Using SystemC

MIAO Wei, LI Guo-hong, ZHANG Qi, WANG Hui

(Shanghai Advanced Research Institute, Shanghai 201210, China)

Abstract: This paper proposes a novel generic memory model written in SystemC as a generic SystemC IP. The model has unified transaction-level interface, while the internal controlling logic and memory organization are cycle-precision. There are subtle parameters to configure the model so that the bandwidth, latency and arbitration rule can be controlled to explore design space. The model has successfully predicted the performance loss caused by a shared memory in a video decoding hardware design when the number of banks to build the memory changes.

Key words: SystemC; modeling; memory; SoC

1 引言

建立芯片模型是在早期进行芯片架构决策的有效方法, 通过建模不仅可以对芯片的性能做出分析, 还可以在硬件没有完成之前开发软件, 不仅提高了产品成功率, 而且缩短了研发周期. 设计人员早期采用 C/C++ 语言进行硬件建模. 但是随着软硬件复杂度的提高, C/C++ 语言难以再满足要求. OSCI 适时推出了 SystemC 语言来适应新的需求^[1]. 如今 SystemC 已经被广泛应用于 SoC 软硬件建模中.

目前大部分 SystemC 建模方面的文献是作者对自己所设计芯片整体模型描述, 这种针对特定芯片设计的文献虽然都有参考价值, 但是其模型本身借鉴性有限. 另一方面, 构成 SoC 硬件的基本组件因为可重用的缘故, 其模型设计具有更好的可借鉴性. 这方面的研究成果相对较少, 而且集中在总线

建模方面^[2-4], 也可以看到多核设计方面的建模方法, 比如文献^[5].

除了上述总线、CPU 架构外, 片内或片外存储器同样是系统关键模块, 对系统性能有着至关重要的影响. 文献^[6]对 SystemClick 框架做了扩展, 从而通过存储器模型能够分析共享存储器对性能的影响. 文献^[7]针对多通道 DRAM 做了建模, 并且分析了不同的配置和使用情况对性能的影响. 这两篇文献都对所设计的模型做了详尽的分析, 具有很好的参考价值, 但是两者的通用性欠佳. 本文提出了一个可配置的通用存储器 SystemC 模型.

2 SystemC

SystemC 可被视作 C++ 硬件建模和仿真的扩展库, 特别适合集成电路的建模, 但它事实上可以用来描述更多的系统^[8-9]. SystemC 通过确定仿真核

收稿日期: 2012-10-20; **修回日期:** 2012-11-16

基金项目: 国家重大科技专项 (2011ZX02505-002); 上海市科委项目 (10DZ1500600)

心(simulation kernel)算法定义了处理并行事件和仿真时间的机制. SystemC 还定义了模块、端口、接口、通道等概念来组成一个复杂 SoC 的连接关系和层次结构. 因此,它解决了标准 C++ 语言用于复杂硬件建模时的难题.

3 存储器的建模

3.1 存储器模型概述

为了能帮助设计人员分析系统性能并选择最合适的存储器构造,存储器模型需要反映出对性能造成影响的关键参数,即带宽、延时和仲裁机制. 而这些参数又是由存储器的结构以及其它参数决定的. 另一方面,本文希望提出一个通用的存储器模型,因此会有更多的参数用于配置存储器以模拟某种特定类型的存储器特征,比如多端口 SRAM.

图 1 给出了所设计的存储器模型的结构. 它包括三个部分:一是模型接口;二是端口及 bank 可配置的存储器;三是由读写通路及仲裁机制构成的存储器控制器. 接口部分采用了事务级建模,而内部模块采用了时钟精准的建模,这样在保证时间精度的同时,方便了接口连接以及提高了仿真速度.

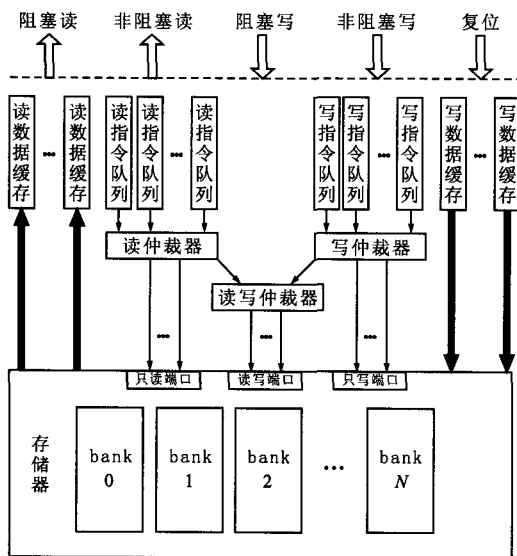


图 1 可配置存储器模型结构

3.2 存储器接口

基于 SystemC 的存储器模型接口,由 SystemC 定义的接口基类派生而来. 整个存储器模型是一个 SystemC 模块,由于实现了所定义的存储器模型接口,它成为 SystemC 概念中的 hierarchical channel. 因此,该存储器模型将如同其它 SystemC 定义的通道,比如 FIFO,一样作为基本组件来连接其它的模块从而构造出 SoC 硬件系统. 所设计存储器具有如

下所述事务级 SystemC 接口.

阻塞读接口:该接口用来读取给定地址的给定大小的数据,当数据没有读出时,调用这个接口的模块线程将被堵塞,直到数据读出为止. 除了地址和数据,该接口还包含调用者的标志信息.

非阻塞读接口:与阻塞读接口不同的是,如果在调用时读指令不能被接受,则返回失败. 否则,不等待数据返回便成功退出. 这样调用接口的模块线程可以在等待时间做其它事情. 调用者通过监听存储器模型广播的当前读出数据多对应调用者标记来在之后读出数据.

阻塞写接口:该接口用来向存储器给定地址写入给定大小的数据. 当指令或写数据不能被存储器模型接收时,调用的线程被堵塞直到接受为止. 除了地址和数据,该接口还包含调用者的标志信息.

非阻塞写接口:与阻塞写接口不同的是,当指令或数据不能被接受时,返回失败而不等待. 否则返回成功.

复位接口:用于复位存储器模型.

为了不强制要求外部提供存储器模型需要的时钟,存储器模型接口中不包含时钟输入. 模型内部有一个专门的时钟产生线程.

3.3 存储器实体

存储器模型中的存储器实体可以由多个 bank 组成,并构成一个子模块. 根据 bank 的数量和种类,该存储器子模块具有不同数量的只读端口、只写端口和读写端口. 为了能尽可能模拟不同的存储器类型,存储器子模块可配置参数如表 1 所示.

表 1 存储器配置参数

参数	说明
bank 的数量	说明由几个 bank 构成
bank 的大小	所有 bank 采用同样的大小
bank 的端口数和类型	虽然通常情况下每个 bank 的端口数量和类型是一样的,这里仍然对每个 bank 进行配置
读端口字节宽度	通常情况下和其它端口字节宽度相同
写端口字节宽度	通常情况下和其它端口字节宽度相同
读写端口字节宽度	通常情况下和其它端口字节宽度相同
允许的 burst 长度	存储器允许的 burst 长度
时钟周期	设置存储器时钟周期
时钟相位	设置存储器时钟相位
地址配置	不同的地址配置,即行、列、bank 地址在整个地址中的字段分配对系统性能有重要影响
不能响应时间	在一段时间内存储器不能响应任何读写操作
不能响应周期	发生不能响应时间的周期
bank 切换时间	切换 bank 造成的延时
行切换时间	切换行造成的延时

3.4 存储器控制器

存储器控制器由指令队列、数据缓存、以及仲裁器构成.这一部分对系统的性能有重要影响.读指令队列和写指令队列用来分别缓存读写指令.当它们写满时,新的读写操作将失败或堵塞.指令队列的数量取决于访问存储器模型的模块有几个优先级.相同优先级的指令会被放到同一个队列.读仲裁器和写仲裁器分别根据队列中的读地址和写地址,读写数据缓存器的状态,以及指令优先级来决定发送哪个指令到哪个端口.而如果读指令和写指令指向了相同的读写端口,读写仲裁器将再一次做出仲裁.这里的每个仲裁器除了上述功能外,具体的仲裁机制以虚函数的形式由具体的实现来决定.在该存储器模型中实现了默认的基于优先级的轮询算法.存储器控制器的配置参数如表 2 所示.

表 2 存储器控制器的配置参数

参数	说明
读指令队列深度	所有队列采用相同的设置
写指令队列深度	所有队列采用相同的设置
优先级数量	允许的优先级总数
每个优先级的饥饿值	当饥饿值满时,该优先级所对应的队列会请求优先权
读数据缓存深度	存放读出数据的缓存深度
读数据缓存字节宽度	存放读出数据的缓存字节宽度
写数据缓存深度	存放写入数据的缓存深度
写数据缓存字节宽度	存放写入数据的缓存字节宽度
时钟周期	不同于上一节存储器的时钟周期
时钟相位	不同于上一节存储器的时钟相位

3.5 参数配置

存储器模型的众多参数需要在构造函数中配置完成.在运行阶段,不能再更改配置.运行阶段的配置更改将是我们下一步的研究内容,可以用于类似于文献[8]的自演化系统.

4 应用实例

图 2 所示为一个 H. 264 解码器模块框图.该解码器共包括四个模块以流水线方式工作在宏块级别.其中熵解码产生运动矢量,变换系数等信息.残差恢复模块通过反量化、反变换操作得到残差.宏块预测模块进行帧内或帧间预测并和残差一起重建出宏块数据.去块效应模块实现 H. 264 去块效应滤波.这四个模块通过一个共享的 SRAM 来交换数据. RTL 设计人员根据设计指标率先完成了各模块微架构的设计,平均解码每个模块所需时间在指标

范围内.

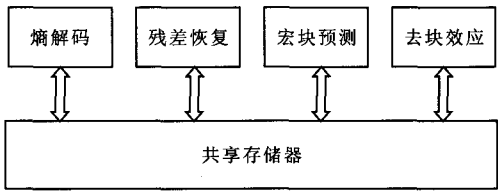


图 2 H. 264 解码器使用共享存储器在解码模块间交换数据

为了确定共享 SRAM 引入的性能损失和确定 SRAM 的设计,采用本文存储器模型进行分析.存储器的大部分参数已经被确定,需要决定采用几个 bank(每个 bank 是一个单端口 SRAM). 另一方面,各模块读写存储器的时间也基本确定从而有了可靠地负载模型.通过改变存储器模型的 bank 数,得到结果如图 3 所示.

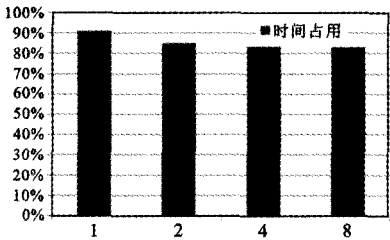


图 3 不同 bank 数所对应的单个宏块平均解码时间占允许时间的比例

由于 bank 的增加可以减少访问存储器的冲突,模型仿真得到的处理时间随 bank 数量的增加而减少.但是可以看到 bank 数量对所消耗的时间影响不大,这说明各解码模块访问存储器的时间分布已经比较均匀.由于一个 bank 的情况已经可以满足要求,所以最后确定 bank 数为 1. RTL 全部完成后的仿真结果所消耗的时间比模型所给出的略少一些,原因是 RTL 设计对访问共享 SRAM 的时间安排做了进一步细化的工作.

5 结束语

本文提出了一个通用的存储器 SystemC 模型,该模型采用统一的事务级接口,和可配置的时钟精准的内部控制逻辑以及存储器构成.通过一系列的参数配置,该模型能够帮助设计人员快速实现系统建模并在存储器设计和选择上做出合理判断.借助其通用性,它还可以帮助保持一致性和减少重复劳动.该模型被应用于视频解码案例中,成功预测了共享 SRAM 的 bank 数对性能的影响,证明了其有效性.

参考文献:

- [1] Ghosh A. System modeling with SystemC[C]// ASIC, 2001. Proceedings. 4th International Conference on. China: Shanghai, 2001: 18—20.
- [2] Mahfuzul S M, Tran J M D. Modeling for performance: SystemC model of a communication bus in a distributed network[C]// International Conference on Information and Communication Technology. Dhaka, 2007: 231—234.
- [3] Srinivasan S, Li L, Ruggiero M, . Dnaka, et al. Exploring architectural solutions for energy optimizations in bus-based system-on-chip[J]. Computers & Digital Techniques, IET, 2008, 2(5): 347—354.
- [4] Chen Lin, Sun Wanzhong, Wang Zhixin, et al. A SystemC-based transaction level modeling of on-chip-bus[C]// International Conference on Computer Science and Software Engineering. China: Wuhan, 2008: 146—149.
- [5] 奚杰, 陈杰, 朱玥. 利用 SystemC 实现多核系统的快速建模[J]. 微电子学与计算机, 2010, 27(7): 220—223.
- [6] Loeb H P, Sauer C. Exploration of embedded memories in SoCs using SystemC-based functional performance models[C]// Forum on Specification & Design Languages. Sophia Antipolis, 2009: 1—6.
- [7] Nikara J, Aho E, Tuominen P A, et al. Performance analysis of multi-channel memories in mobile devices. System-on-Chip. Tampere, 2009: 128—131.
- [8] 王晓艳, 姚睿. 使用 SystemC 设计片上自演化系统[J]. 微电子学与计算机, 2011, 28(6): 28—31.
- [9] 钟辉捷, 雷航. 基于 Petri 网和 systemC 的 SoC 系统描述[J]. 计算机应用, 2007, 27(2): 147—149.

作者简介:

妙 维 男, (1980—), 博士, 副研究员. 研究方向为数字电路设计和 SOC 设计方法学等.

李国宏 男, (1977—), 硕士, 高级工程师. 研究方向为数字电路设计和通信信号处理等.

章 琦 男, (1984—), 博士, 助理研究员. 研究方向为超低功耗数模混合电路设计等.

汪 辉 男, (1975—), 博士, 研究员. 研究方向为数模混合电路设计等.