

SystemC 在 SoC 总线交易级建模的研究与应用

陈琳¹, 刘瑰^{2,3}, 徐晏琦¹, 何红旗²

(1. 解放军信息工程大学电子技术学院, 郑州 450004; 2. 解放军信息工程大学信息工程学院, 郑州 450002; 3. 江南计算技术研究所, 无锡 214083)

摘要: 采用了 SystemC, 结合 SoC 片上总线, 探讨了在交易级的建模方法, 并结合数字视频后处理芯片给出了建模实例。基于 SystemC 的 SoC 总线模型有效地克服了 SoC 软硬件协同设计的时间瓶颈问题, 提高了开发效率, 缩短了产品的开发周期。

关键词: 系统 C 建模语言; 片上系统; 交易级平台; 视频后处理

Study and Application to Transaction Level Modeling of SoC Bus Using SystemC

CHEN Lin¹, LIU Gui^{2,3}, XU Yanqi¹, HE Hongqi²

(1. Institute of Electronic Technology, PLA Information Engineering University, Zhengzhou 450004; 2. Institute of Information Engineering, PLA Information Engineering University, Zhengzhou 450002; 3. Jiangnan Institute of Computing Technology, Wuxi 214083)

【Abstract】 Using SystemC, this paper discusses the methodology of transaction level modeling in connection with on-chip bus models and with the chip of video post-processing, gives an example of transaction level modeling. The time bottleneck of current hardware/software co-design method can be eased effectively. This model can improve efficiency and reduce the cost of products.

【Key words】 SystemC; SoC; Transaction level bench; Video post-processing

片上系统是一种由硬件和软件构成的混合系统, 常用的开发方法是硬件工程师使用硬件描述语言(如 VHDL、Verilog)建立硬件模型, 软件工程师使用高级语言(如 C、C++)来建立软件模型, 这种语言描述的不一致性会带来大量时间开销, 降低协同验证的仿真性能, 不能满足开发工作对不同层次模型快速仿真的需求, 从而导致系统设计过程的反复。为此, 一种新的系统级的设计方法 SystemC 被提出。SystemC 是一个 C++类库, 用 SystemC 可以很方便地实现一个软件算法在硬件上的应用, 完成片上系统的设计。本文采用了系统级建模语言 SystemC 对 SoC 总线在更高的抽象级(相对于 RTL 级)的应用, 即交易级进行了软硬件统一建模, 本模型比传统的软硬件协同设计方法更灵活, 可以提高开发效率, 降低开发成本, 并有效地保证系统开发的质量。

1 SystemC 及交易级建模

SystemC 是 OSCI(Open SystemC Initiative)组织制定和维护的一种基于 C++的建模平台, 它完全是用 C++语言编写的, 由 C++类库和仿真内核构成, 支持门级、RTL 级、系统级等各个抽象层次上硬件的建模和仿真, 而且是开放源代码的。SystemC 支持硬件/软件协同设计, 能够描述由硬件和软件组成的复杂系统结构, 支持在 C++环境下对硬件、软件和接口的描述。

SystemC 语言的一个重要特征是支持系统的交易级建模(TLM)。交易(transaction)成为越来越重要的一个概念。在 SoC 领域, “交易”是指在系统模型中两个组件间的一次数据交换。这个交换与采用的协议无关, 通常不涉及具体的总线时序等细节。读交易和写交易保证模块之间同步操作是事件交易。另外, 模块之间的中断也是一种交易。TLM 模型较 RTL(register transfer level)级容易编写, 而且模拟速度比 RTL

级提高了 100~1 000 倍, 每秒能仿真至少 100 000 个交易。在 SoC 设计中, SystemC 可以快速开发交易级模型, 根据系统的初始功能规范快速建立硬件的可执行规范。TLM 高的模拟速度允许提前开发和验证硬件相关的应用软件。通过在 TLM 中加入时序细节, 可以评估系统的性能, 探索系统的结构。SystemC 的模块本质也促进了所开发构件在系统间的复用。

2 基于 SoC 总线的交易级建模

2.1 SoC 总线模型

图 1 给出了一个集成到一个芯片内部的 SoC 总线模型, 包括片上总线、总线仲裁器、数字信号处理器、微处理器、存储器和其他专用集成电路。这样一个复杂系统同时包括软件和硬件, 在采用 SystemC 作为建模语言的情况下, 整个系统可以方便地实现软硬件协同设计。

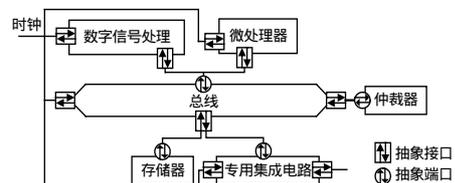


图 1 SoC 电子系统框架

一个 SoC 系统模型从逻辑上看包括总线主设备(master)、从设备(slave)、仲裁器(arbiter)和总线本身。通过 SystemC 中相应的端口、接口和通道连接相应的设备, 从而构成完整的 SoC 总线模型。主设备在总线系统中是主动部件, 它请求来

作者简介: 陈琳(1975—), 女, 硕士、讲师, 主研方向: SoC 软硬件协同设计, 系统级芯片设计; 刘瑰, 博士生、工程师; 徐晏琦, 博士生、讲师; 何红旗, 硕士、讲师

收稿日期: 2005-09-19 E-mail: chenlin916@163.com

自从设备的服务。主设备可以请示总线访问并发起交易，如存储器读/写。仲裁器对主设备的请示进行仲裁，确保同一时刻只有一个主设备使用总线；总线上的从设备可以地址范围相区分。在模拟初始化时从设备把它的地址范围通知给总线，也可以通过译码器对每次传输的地址进行译码，给相应的从设备发出选中信号。其模块间的抽象关系如图 2 所示。

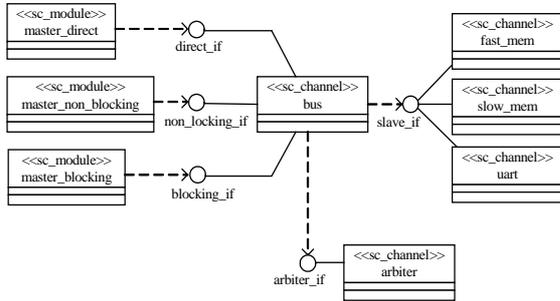


图 2 片上总线系统的逻辑

2.2 总线

总线是实现直接设备接口、阻塞型设备、非阻塞型设备的分层通道。在某些时钟的上升沿，总线会收集到来自各个主设备的从设备读写请求，并将这些请求加入到请求队列。在下降沿，总线将这些请求发给仲裁器进行仲裁，选择应本服务的那个主设备的请求进行服务。对于直接接口，请求不进入队列，直接进行总线读写操作并返回结果；非阻塞型接口，只将请求插入队列，然后读写操作返回。对于阻塞型接口，要等到所请示的读或写操作完成后的下一个时钟的上升沿操作才会返回。

2.3 仲裁器

总线仲裁器被用来从多个总线操作请求中选择哪个总线请求应在当前被服务。仲裁器接口 arbiter_if 定义为：

```
class arbiter_if
{
public virtual sc_interface{
public:
virtual request *
arbitrate(const request_vec &requests)=0;
};
```

它只有一个方法 arbitrate(const request_vec &requests)，输入参数为当前的请示队列，返回值为选定的被服务(或处理)的请求。

2.4 主设备

主设备模块被定义为：sc_port< bus_if>。sc_port 在 SystemC 2.0 中是所有端口的基类，在其基础上用户可以自定义端口，bus_if 是指定的总线接口类型，这样实现了本接口中方法的总线通道模块可以被连接到端口上，向主设备模块提供服务。

2.5 从设备

从设备接口(slave interface)定义了地址传输函数、读写函数，以响应主设备的地址传输和数据传输请求。读函数被定义为

```
virtual void
read(address_type address, //读操作地址
int number_of_bytes)=0; //字节数
```

写操作的定义与读操作类似，从设备通道模块实现了本接口定义的方法供总线模块调用。

3 SoC 总线模型的应用

在数字电视接收机中，主要包括信道解调、信源解码、

视频后处理 3 个核心芯片，其中数字视频后处理芯片是当前最有市场开发价值的数字电视核心芯片之一。本芯片的系统结构如图 3 所示，构成了一个 SoC 系统。本系统的功能是通过数字信号处理方法把众多格式的视频信号转换成统一格式的输出信号，从而在数字域实现多种信号到电视显示器件之间的无缝连接，大大降低对显示器件及其扫描等周边电路的性能要求，降低数字电视机的设计成本。同时，后处理芯片的数字处理功能可方便实现精确的伽玛校正、抗锯齿、运动补偿、轮廓增强等图像处理算法，实现消除闪烁，提升图像观赏质量和视觉效果。

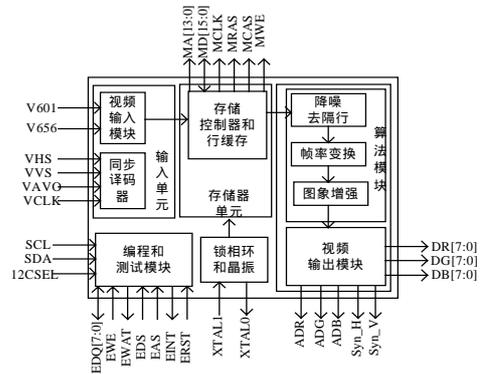


图 3 视频后处理芯片系统结构

3.1 存储器

本文在建立总线模型时，采用存储器 SDRAM 作为从设备，存储控制器和行缓存作为主设备接入总线。从设备存储器接入总线时，以 slave_if 为基类实现。其中 start_address() 和 end_address() 这两个函数为设定地址空间的首地址和尾地址，上电时对 SDRAM 进行初始化操作。

3.2 存储控制器和行缓存

存储控制器和行缓存模块由 SystemC 编写，本模块 MCTRL 提供对片外 SDRAM 的地址产生和读写交易。片外 SDRAM 用来存储有效的视频图像和运动矢量。模块 MCTRL 是采用带时序逻辑的隐式有限状态机设计而成的，并使用钟控线程 SC_THREAD 描述，实现其初始化、命令仲裁、猝发读/写、自动刷新、自刷新等操作。其内部控制状态如图 4。

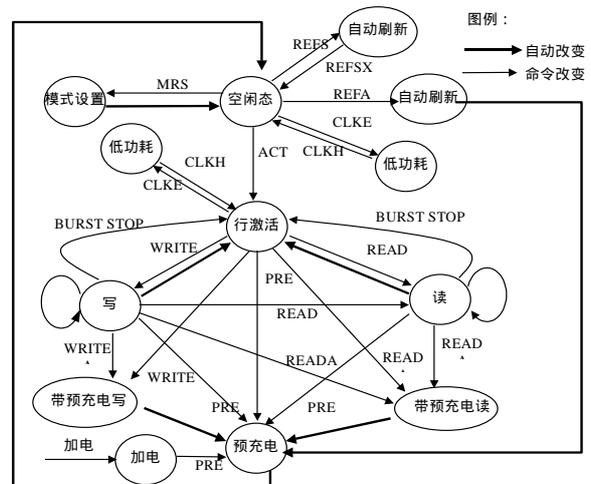


图 4 SDRAM 内部控制器状态转换

在 MCTRL 模块中定义了总线端口 sc_port<blocking_if> bus_port;这样它就可以读取地址为 m_address 的 SDRAM 上的 length 个双字的数据到数组 mdata 中。本模块对 SDRAM

的读写采用阻塞方式的方法(包括阻塞读操作和阻塞写操作),读操作定义为:

```
virtual status burst_read(
    unsigned int unique_priority,
    int *data,
    unsigned int start_address,
    unsigned int length=1,
    bool lock=false)=0;
```

写操作的定义与读操作类似,然后建立总线读写请求信号,完成读写的有限状态机设定。

3.3 结果分析

利用 SystemC 语言建立的总线模型在 VC++6.0 环境下编译成功,生成与项目名称相同的可执行文件 project.exe。为了查看波形,在源代码中指定生成一个 VCD 或 WLF 或 ISDB 波形文件。在 ModelSim 环境下只能打开 WLF 文件,因此利用命令 vcd2wlf 将 VCD 文件转化为 WLF 文件,命令格式为: vcd2wlf <source.vcd> <target.wlf>。这样在 ModelSim 的 wave 窗口中就可以看到波形。另外也可以使用免费的工具如 SystemC_win 编译和看波形。用 Vcdviewer 看 VCD 文件的波形,如图 5 所示。

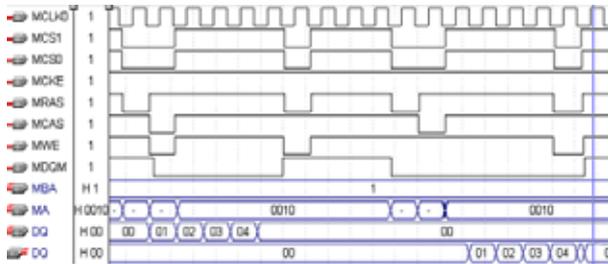


图 5 存储控制器的总线模型的仿真波形

(上接第 214 页)

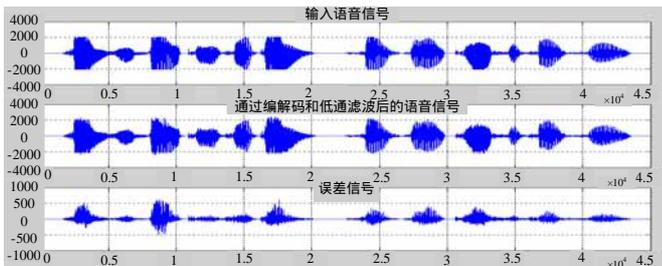


图 4 语音编解码效果示意图

3.3 嵌入式 CPU 的使用率

通过加载 QMC 驱动后,编写语音测试代码观测 MPC860 的占用率如表 2 所示。

表 2 CPU 占用率

	1 路全双工	2 路全双工	3 路全双工	4 路全双工
CPU 使用率	5%	9.6%	14.1%	18.7%

从表 2 可知,CPU 的处理能力足够胜任基本的话音服务。从而可以完成更多的语音传输和存储工作。使得整个系统的扩展性增强。

经过验证,程序完全实现了设定的设计功能,本模型能在 133MHz 下正确完成 SDRAM 的读写交易,提高了仿真速度,缩短了视频后处理芯片的开发周期。

4 结论

在交易级上对 SoC 总线采用 SystemC 进行建模的方法使软硬件开发基于同一语言环境,这一特点能够快速、有效地创建 SoC 模型,支持了软硬件的协同设计与验证,并有效克服了 SoC 软硬件协同设计的时间瓶颈问题。本模型充分体现了 SystemC 语言的优势,对片上系统 SoC 的发展有着重要意义。

参考文献

- 1 San Y. Towards a New Standard for System-level Design[C]. Proceedings of the Eighth International Workshop on Hardware/Software Codesign. San Diego: ACM Press, 2000-02.
- 2 Pasricha S. Transaction Level Modeling of SoC Using SystemC (2.0)[C]. Proc. of Synopsis User Group Conference, Bangalore, 2002-05.
- 3 Open SystemC Initiative, SystemC Homepage[EB/OL]. <http://www.systemc.org>, 2003.
- 4 Synopsys Inc. SystemC User's Guide (Version 2.0)[EB/OL]. <http://www.systemc-org>, 2002-04.
- 5 Rashinkar P, Paterson P. System on a Chip Verification Methodology and Techniques[M]. Lincoln: Kluwer Academic Publishers Group, 2002.
- 6 Keith J. Video Demystified-a Handbook for the Digital Engineer[M]. Eagle Rock, Virginia: LLH Technology Publishing, 1996.
- 7 陈 曦. SystemC 片上系统设计[M]. 北京: 科学出版社, 2004.
- 8 张俊亲. 基于 SystemC 的事务级 AMBA 总线模型的设计与实现[J]. 武汉大学学报, 2004, 50(1): 87-90.

4 结论

16kbps CVSD 编码算法简单,占用资源少,易于实现且有很强的抗误码能力,语音质量可以满足一般的通话要求。将 CVSD 的编解码器用 DSP 实现并与嵌入式系统相结合,更增加了应用的灵活性,扩展了其功能,可以广泛应用于数码录音,IP 电话和多媒体终端等含有语音业务的产品中。

参考文献

- 1 姚天任. 数字语音处理[M]. 武汉: 华中理工大学出版社, 1992.
- 2 TMS320VC5509 Fixed-point Digital Signal Processor Data Manual[R]. Texas Instrument Company, 2003-07.
- 3 MPC860T PowerQUICC™ Technical Summary[R]. Motorola Inc., 2001-12.
- 4 QMC Supplement to MC68360 and MPC860 User's Manuals[R]. Motorola Inc., 1997.
- 5 杨 俊, 唐 昆, 冯重熙等. 16kbps CVSD 与 64kbps PCM 编码数字转换算法[J]. 电子学报, 1994, 32(4).
- 6 国家技术监督局. 增量调制终端设备技术要求及测量方法[M]. 北京: 中国标准出版社, 1992: 25-35.