

基于 System C 的数字基带处理单元设计

叶友鹏,卜刚,张龔

(南京航空航天大学,江苏南京 211106)

摘要:随着集成电路设计规模的不断扩大,系统变得更加庞大和复杂,在设计系统芯片的各个流程中,像系统定义、软硬件划分、设计实现等都变得越来越复杂。如何满足日益复杂的 SoC 设计要求成为了集成电路设计的重要因素,业界一直在寻找一种系统级语言能够在更高层次上对软件和硬件实现描述。System C 正是在这种情况下,由 Synopsys 公司和 CoWare 公司积极响应目前各方对系统级设计语言的需求而合作开发的。System C 相当于面向系统级设计的 C++ 扩展库,是一种设计人员可以通过 System C 准确有效设计出软件算法模型、硬件结构和系统架构设计的方法。设计人员可以通过常用的 C++ 开发工具中添加 System C 的类库来实现对系统的模型设计,快速地实现仿真和优化设计,还可以研究不同的算法模型,这样就可以为硬件和软件设计人员提供一个可执行规范。因为可执行规范本质上是一个 C++ 程序,在面向对象描述尤其是针对事务处理级模型上有着许多优势,成为软件和硬件设计人员的一个设计标准。文中主要通过 System C 来建立 RFID 通信算法模型,主要基于 ISO/IEC 18000-6C 通信协议。系统包括基带处理单元模型和总线模型等,通过模型来评估系统的性能和总线带宽的需求。为系统设计提供一套自顶向下的设计方法。

关键词:算法设计;System C;ISO/IEC 18000-6C;软硬件划分;RFID;仿真验证

中图分类号:TP31

文献标识码:A

文章编号:1673-629X(2019)08-0007-05

doi:10.3969/j.issn.1673-629X.2019.08.002

Design of Digital Baseband Processing Unit Based on System C

YE You-peng, BU Gang, ZHANG Yan

(Nanjing University of Aeronautics & Astronautics, Nanjing 211106, China)

Abstract: With the continuous expansion of the scale of integrated circuit design, the system becomes larger and more complex. In each process of designing system chips, system definition, software and hardware division, design and implementation, etc. become more and more complex. The IC design community has been considering how to meet SoC design requirements and has been looking for a system-level design language that can simultaneously implement higher-level software and hardware descriptions. It was in this context that System C was developed by Synopsys and CoWare in response to the current needs for system-level design languages. System C is equivalent to C++ extension library for system-level design, which is a method by which designers can accurately and effectively design software algorithm model, hardware structure and system architecture design. Designers can add the class library of System C to the common C++ development tools to realize the model design of the system, realize the simulation and optimization quickly, and study different algorithm models, so as to provide an executable specification for hardware and software designers. Because an executable specification is essentially a C++ program with many advantages in object-oriented description, especially for transaction-level models, it has become a design standard for software and hardware designers. We mainly establish the RFID communication algorithm model through System C, which is mainly based on ISO/IEC 18000-6C communication protocol. The system includes a baseband processing unit model and a bus model. Models are used to assess system performance and bus bandwidth requirements, which provides a top-down design approach to system design.

Key words: algorithm design; System C; ISO/IEC 18000-6C; hardware and software division; RFID; simulation verification

0 引言

现代片上系统的设计难度来源于其复杂性,而片上系统正在变得越来越复杂。设计复杂性的不断增加

促使电子系统设计(electronic system level, ESL)方法学 System C 和 Verilog/System Verilog 相结合,提供了当今芯片设计所需的一套从 ESL 到 RTL 的标准解决

收稿日期:2018-09-06

修回日期:2019-01-09

网络出版时间:2019-03-27

基金项目:江苏省自然科学基金(BK2012792)

作者简介:叶友鹏(1993-),男,硕士研究生,研究方向为片上系统设计;卜刚,教授,研究方向为集成电路设计。

网络出版地址:<http://kns.cnki.net/kcms/detail/61.1450.TP.20190327.1624.030.html>

方案^[1]。

对于未来的复杂系统,在进行 RTL 设计前需要考虑更多的因素,包括总线带宽能否满足吞吐量、存储资源是否足够、系统架构是否恰当等。所以系统级仿真就变得尤为重要。这些仿真还要考虑模型上需要运行大量的软件,覆盖系统所需要的所有功能^[2]。

在传统的设计方法中,系统级或者高层次算法往往使用 C、C++ 等进行描述和设计,而在寄存器传输级使用硬件描述语言进行描述。现在使用最广泛的两种硬件描述语言分别是 VHDL 和 Verilog HDL,从系统级设计到具体的电路级时 C 和 C++ 的描述必须手工转化为使用 VHDL 和 Verilog HDL。这种设计方法学存在一个非常大的缺点,即使用不同的语言进行系统描述的不一致性。系统的复杂度在提高,人们非常迫切需要一种能够实现全部设计的语言。这种语言能够描述不同的抽象级别和实现不同层次的设计,能够实现软硬件的协同设计和验证,而且仿真速度要足够快。这就需要一种全新的能够描述所谓系统级的语言^[3]。文中基于 ISO/IEC 18000-6C 协议,采用系统级设计语言 System C 建立基带数字信号处理算法的模型,划分进程,搭建数字基带处理单元的 SoC 系统,仿真验证得到正确的系统模型,最后通过综合产生算法模型对应的具体电路和设计报告。

1 System C 简介

System C 是由 Synopsys 提出的,目的是以一种系统设计的思想进行系统设计。它将软件算法与硬件实现很好地结合在一起,提高了整个系统设计的效率和正确性。System C 是一个 C++ 类库,可以让设计人员在设计硬件的同时开始着手于软件设计,既可以描述软件算法,还可以实现对硬件结构的描述。在 C++ 的开发工具中加入 System C 类库就可以实现对系统级模型的开发,而且能够实现快速的仿真和验证。作为一种新的设计方法学,提供了系统级设计的方法,同时作为一种可执行规范,为设计人员提供了一个设计标准^[4]。

2 System C 设计流程

System C 具有所有硬件描述语言所共有的基本特征,包括模块、进程、端口和信号等。在 System C 中,时钟信号由特殊的库函数进行描述,可以直接配置,简化了对时钟信号的定义和描述。为了描述电路中的实际行为,System C 采用与传统硬件描述语言基本的调度模型—基于(delay)延迟。为了实现多个进程同步、通信和其他特性,System C 支持自定义不同的接口(Interface)、端口(Port)和通道(Channel)。接

口作为方法(Method)的集合,没有具体实现这些方法,在 C++ 中是纯虚函数。在 System C 中,进程只调用通道提供的接口方法,接口方法调用和支持不同抽象级别的混合建模是通信细化的基础^[5]。System C 创建的进程可以是静态也可以是动态的。进程描述了事物的行为,可以是一个复杂的状态机,也可以是一个特定应用的软件模型。多个动态创建的软件进程可以进行同步。为了支持验证,System C 在核心语法的基础上扩展了一个专门的验证库。为了支持 ESL 建模,System C 在核心语法的基础上扩展了 TLM2.0 事物处理级建模库,支持非定时、近似定时和松散定时事物处理级模型^[6]。总之,System C 提供了一整套完整的建模方案,主要包括的内容如图 1 所示。

验证库、事物处理级建模库	
基本通道 信号、互斥、信号量、FIFO	
模块 (Module) 端口 (Port, Export) 进程 (Process) 接口 (Interface) 通道 (Channel) 事件 (Event)	数据类型 4 值数据类型 4 值向量数据类型 比特和比特向量 任意精度整形 定点数据类型 C++ 用户自定义数据类型
C++ 语言标准	

图 1 System C 架构

3 System C 开发环境搭建

System C 可以在 Windows 和 Linux 系统中搭建开发环境, System C 的开发工具有很多,但是实质上 System C 只是 C++ 增加的一个类库,因此任何一个符合 ANSI 标准的 C++ 编译工具都可以对 System C 程序进行编译链接,从而生成可执行文件。需要做的则是如何把 System C 的头文件和库文件整合到 C++ 开发环境中去。文中采用的 System C 版本是 SystemC-2.0.1,通过在 Visual C++ 6.0 中添加 System C 库来搭建开发环境。先建立一个空白的 C++ 工程,然后在设置选项中选中 C++ 语言并勾选支持时间类型信息(RTL),软件上设置支持 RTL 级描述,如图 2 所示。

设置完工程支持的选项后还需要添加 System C 的库函数使编译器识别 System C。通过添加包含路径来添加 System C 的库函数,添加路径选择到 SystemC-2.0.1 的 src 目录,设置后再添加仿真库 systemc.lib 文件到工程目录下,开发环境就算搭建完成了。

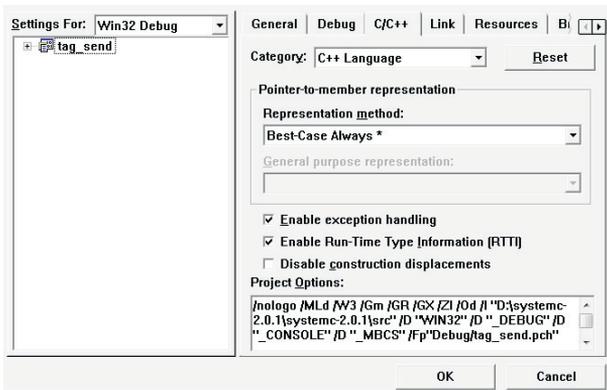


图 2 System C 环境配置

4 基带算法模型设计

文中设计的 RFID 的 SoC 系统,主要包括了基带

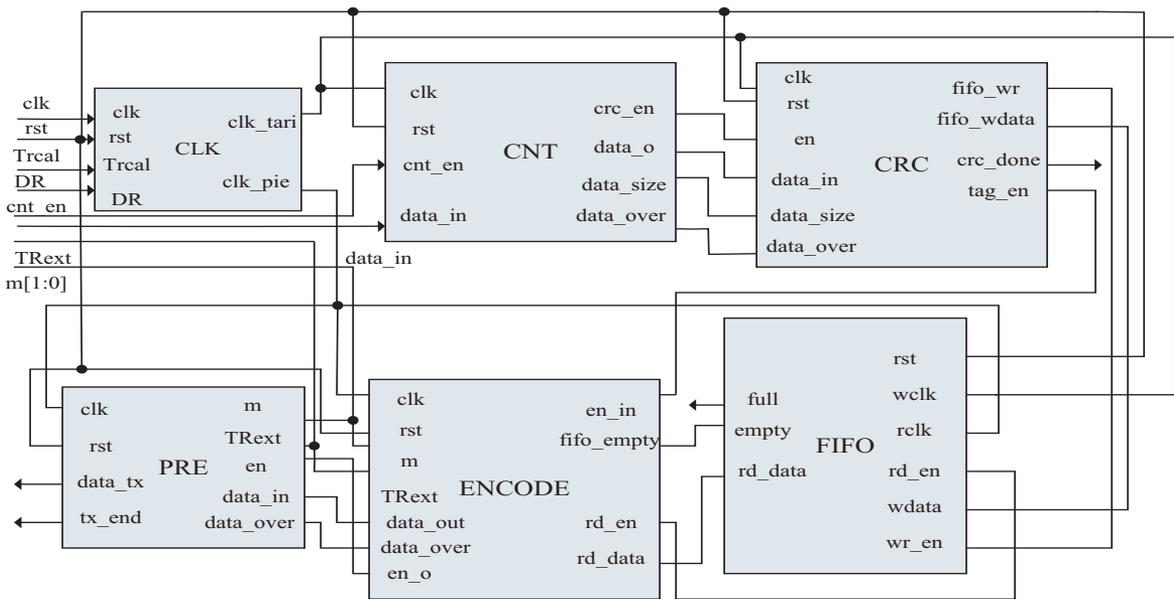


图 3 数字基带发送链路

5 仿真验证

系统建模采用 System C 设计,而 System C 本质是 C++库。所以模型设计的功能是否满足要求可以通过支持 C++语法的编译器通过添加 System C 的可执行类库来实现功能验证。不仅可以通过 C++编译环境对模型进行功能验证,还可以借助于 EDA 工具 Modelsim 对模型进行时序的仿真来观察模型的时序设计结果。

文中采用的是 Visual C++ 6.0 编译环境,在 Visual C++ 6.0 验证功能的正确性。通过 Modelsim 对模型的时序进行仿真,如图 4 所示。

6 Catapult C 综合

设计模型经过仿真和验证满足要求后,可采用 Mentor 的高层次综合工具 Catapult C 对模型进行综

合。实现直接由定点的 C/C++/System C 代码得到可综合的 RTL 代码,可以在较短的时间得到大概的设计系统时序和面积等信息,从而为系统的可行性分析以及实现具体时间表上提供客观依据^[10]。通过对比在硬件模型和软件模型中的算法和功能来实现软硬件的划分,根据系统要求适当地分配硬件和软件的实现部分。可以在保持代码不变的前提下,通过 Catapult C 上施加不同的约束,即可得到不同效率的 RTL 代码^[11]。

数字处理单元和运行在指定处理器上的软件。数字基带链路主要把处理器要发送的数据根据协议按照特定的格式发送到射频端,软件主要实现对接受数据的识别和处理以及特定数据的发送^[7]。以阅读器的基带处理链路设计为例,链路对数据进行并转串,CRC 校验,PIE 编码,添加同步码和数据发送,根据系统的输入和输出建立基带处理模型^[8]。根据模型的输入输出以及硬件时序的要求,对输入的时钟进行分频,采用异步复位的方式,总线输出数据位宽为 32 位。因为编码和检验时钟和发送时钟的不一致,采用一个 FIFO 模块来对数据进行异步处理和数据缓存^[9]。最后采用模块化的建模思路针对具体的应用平台采用 APB 总线对模块进行读写,构造的电路大致如图 3 所示。

采用 Catapult C 实现 C/C++/System C 算法模型到具体 RTL 以及门级电路的流程如图 5 所示。

通过对数字基带处理单元进行建模,分析模型的具体参数。以设计好的 PIE 编解码模型为例,通过 Catapult 综合如何得到模型的参数,将设计好的 System C 模型放入到 Catapult C 建立的工程中,选择工艺参数、全局约束、逻辑综合工具、时钟频率、复位信

号和兼容库等,编译生成甘特图、RTL 电路以及资源 使用报告等输出文件^[12]。

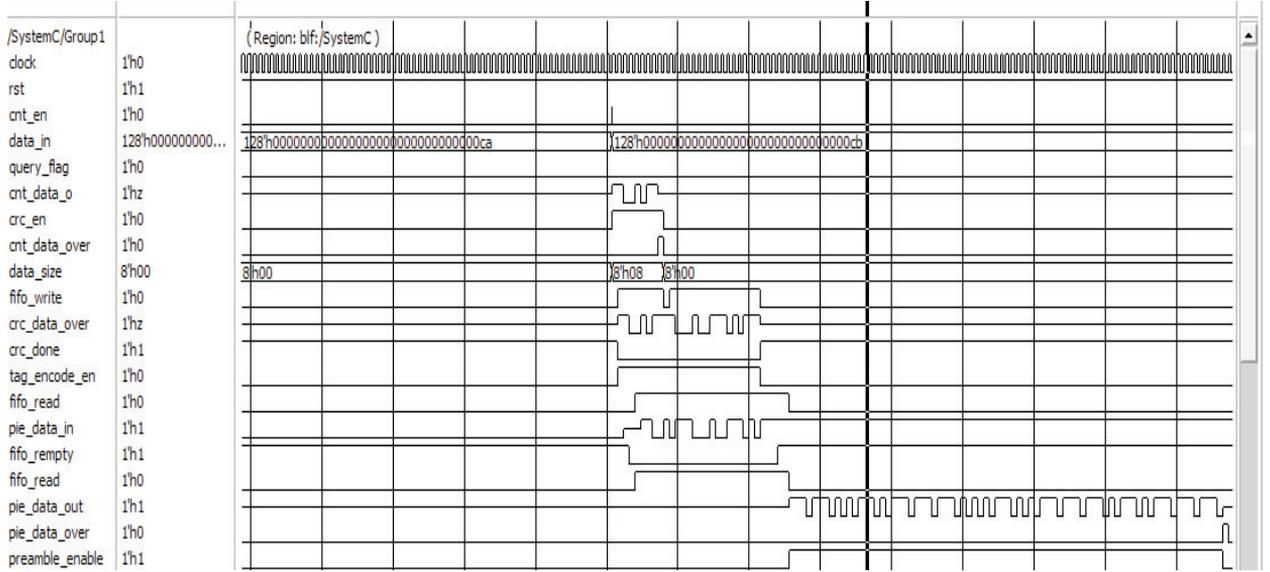


图 4 Modelsim 仿真基带模型

由生成的甘特图可知,程序的运行时钟周期有实际上进行运算的周期和冗余周期,但是可以通过设置流水线来消除冗余周期。然后根据程序可以生成 RTL 网表文件和其他输出文件,包括资源利用率、关键路径报告和 VHDL 代码等等^[13]。其中 PIE 编码模块 System C 算法程序生成的 RTL 文件如图 6 所示,图 6 是模块内部的门级电路图。

根据程序对算法的编译综合后生成的报告,可以看到设计模型使用的资源包括了乘法器、加法器等资源的报告,根据甘特图还可以看到运行占用的时钟数从而改善设计,通过 RTL 网表文件观察生成的具体电路结构,其中包括了电路的关键路径报告等。最后生成的 VHDL 代码还可以移植到其他平台上,实现从 C/C++/System C 算法直接到 RTL 级的设计流程,快速可行而且简便^[14]。

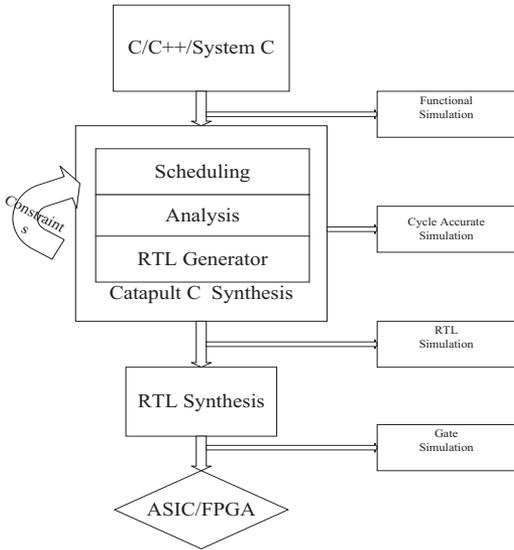


图 5 Catapult 设计流程

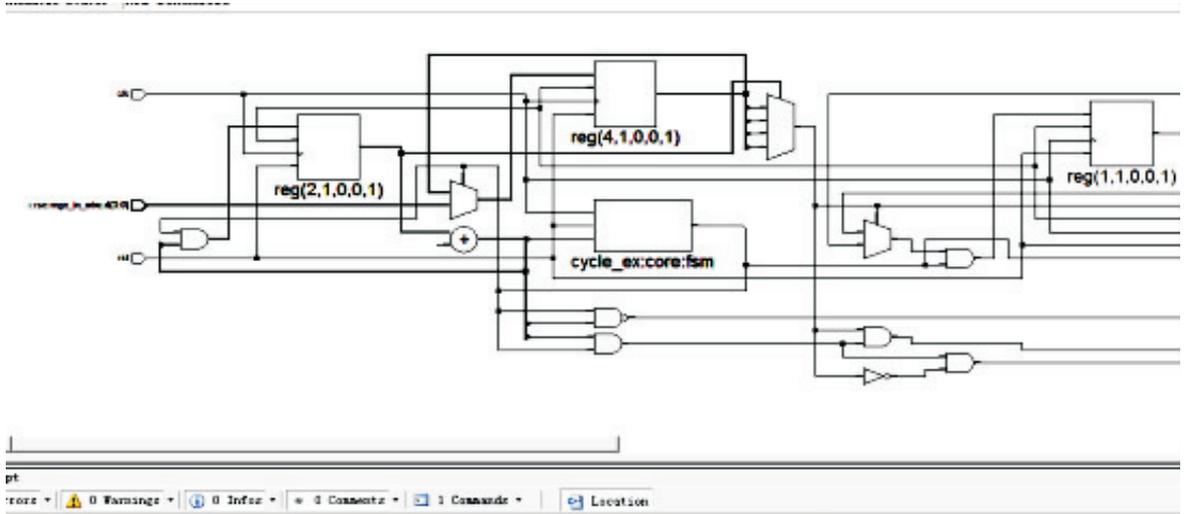


图 6 PIE 编码程序模块具体门级电路

从算法的 C/C++/System C 代码开发,到生成 RTL 级代码都可以在 Catapult 中完成。算法的优化实现通过 Catapult 综合之后将直接反映到硬件实现上,无需耗费大量时间在编写 RTL 级代码上,使设计者能更多地专注于算法本身。模型设计完成后,可以通过修改约束条件或者对系统的架构,就可以生成对应条件的 RTL 级代码,生成不同的设计方案^[15]。通过对比不同设计方案的报告,包括面积、功耗、关键路径等信息,得出最好的实现方案。

7 结束语

现代片上系统的设计难度来源于其复杂性,设计复杂性促使电子设计系统设计方法学 System C 和 Verilog/System Verilog 相结合,提供了当今芯片设计所需的一套从 ESL 到 RTL 的标准解决方案^[16]。ESL 设计方法学的目标是在一种更高的层次上对系统级芯片进行描述和设计的方法,在这种更高层次上,对事物的描述非常抽象,可以对芯片的特性进行功能性描述就可以,而没有必要考虑具体的 RTL 级细节是如何实现的^[17]。事物处理级建模是 ESL 设计的关键,对事务处理级的建模能够在芯片设计的早期实现对抽象层次原型的描述,描述了系统各种抽象行为和每个功能单元所要实现的具体操作(事物处理)。这种事物处理级模型因为抽象层次高,所以仿真速度要比具体的 RTL 级电路的仿真速度快出若干个数量级^[4]。文中采用 ESL 设计和验证方法学,根据 ISO/IEC 18000-6C 协议标准,实现了从 System C 算法到 RTL 级设计和验证。针对具体的算法设计特定的硬件架构,不仅速度快而且资源使用率低,比传统的设计快好几个周期。利用高层次综合工具 Catapult 还可以实现算法到具体工艺或者器件的实现。总之,作为一种先进的 SoC 设计方法,面对未来更加复杂的片上系统,完全能够胜任设计带来的挑战。

参考文献:

[1] HENNESSY J L, PATTERSON D A. 计算机体系结构量化研究方法[M]. 贾洪峰,译. 北京:人民邮电出版社,2013.

- [2] FLYNN M J, LUK W. Computer system design: system-on-chip[M]. [s. l.]: Wiley, 2011: 15-20.
- [3] 李挥, 陈鑫. System C 电子系统级设计[M]. 北京: 科学出版社, 2010.
- [4] 夏宇闻, 甘伟. System C 入门[M]. 北京: 北京航空航天大学出版社, 2008.
- [5] 饶永, 徐成. 基于 System C 的片上网络系统级建模研究[J]. 计算机仿真, 2007, 24(10): 307-310.
- [6] 陈鑫, 黄毅. 片上系统设计思想与源代码分析[M]. 北京: 电子工业出版社, 2008: 352-374.
- [7] WANT R. An introduction to RFID technology[J]. IEEE Pervasive Computing, 2006, 5(1): 25-33.
- [8] 王玲, 丛静. RFID 中间件读写器管理与协调研究[J]. 计算机应用与软件, 2011, 28(12): 198-201.
- [9] 夏宏, 吴济文. 超高频 RFID 读写器系统的设计与实现[J]. 计算机应用, 2012, 32(8): 2369-2373.
- [10] 谢正, 张开锋. 基于 Catapult C 的 DCT 算法设计[J]. 信息化研究, 2011, 37(4): 42-45.
- [11] 秦贵军. 基于 Catapult C 的汽车轮胎力估计的 FPGA 实现[D]. 长春: 吉林大学, 2013.
- [12] NANE R, SIMA V M, PILATO C, et al. A survey and evaluation of FPGA high-level synthesis tools[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuit and Systems, 2016, 35(10): 1591-1604.
- [13] GAL B L, CASSEAU E. Latency-sensitive high-level synthesis for multiple word-length DSP design[J]. EURASIP Journal on Advances in Signal Processing, 2011, 2011: 927670.
- [14] GEORGE N, NOVO D, ROMPF T, et al. Making domain-specific hardware synthesis tools cost efficient[C]//International conference on field-programmable technology. Kyoto, Japan: IEEE, 2013: 120-127.
- [15] COUSSY P, HELLER D, CHAVET C. High-level synthesis: on the path to ESL design[C]//IEEE international conference on ASIC. Xiamen, China: IEEE, 2011: 1098-1101.
- [16] 陈琳, 徐金甫, 章轶, 等. 基于 System C 的事物级建模研究[J]. 微电子学与计算机, 2006, 23(11): 213-216.
- [17] 张玉峰, 杨志家. 基于 System C 事物级的建模仿真研究[J]. 微计算机信息, 2005, 21(26): 71-73.