

面向 SoC 的系统级设计语言

王兆菊, 龚龙庆

(西安微电子技术研究所, 陕西 西安 710071)

摘 要: 目前集成电路的设计已经进入 SoC 时代, 介绍了 SoC 设计所用的系统级设计语言的发展情况, 以及典型的系统级设计语言在不同设计层次上的描述能力; 并介绍了几种典型的系统级设计语言, 给出了这些语言之间的特点对比, 最后阐述了对系统级设计语言的看法, 展望了其发展前景。

关键词: 系统级设计语言; SystemVerilog; SystemC; SpecC

中图分类号: TP312

文献标识码: A

文章编号: 1673-629X(2007)11-0084-04

System - Level - Design Language for System - on - Chip

WANG Zhao-ju, GONG Long-qing

(Xi'an Microelectronic Technique Institute, Xi'an 710071, China)

Abstract: Currently the design of IC is in SoC era. Firstly presents the development of the system level design language for system-on-chip, and discusses its description capability at different abstract level. Also introduces some typical system level design languages and the comparison of those languages. Finally the points of view and the prospects for future of those languages are given.

Key words: system-level-design language; SystemVerilog; SystemC; SpecC

0 引言

随着集成电路的特征尺寸进入超深亚微米, 单块集成电路上的晶体管数目越来越多, 设计的复杂性日益提高, 以硬/软件协同设计 (Hardware/Software Co-design)、IP 核复用和超深亚微米技术为支撑的 System-on-chip (SoC) 已成为国际超大规模集成电路的发展趋势和现时集成电路的主流。与此同时, 集成电路设计也从原来的寄存器传输级 (RTL) 转向系统级 (ESL)。

在系统级设计中, 根据抽象级别的不同, 电路设计可分为六个不同的抽象级别, 如图 1 所示。

(系统规范) Spec.
(系统架构) Arch.
(行为级) Beha.
(寄存器传输级) RTL
(门级) Struct
(晶体管级) Phy.

图 1 SoC 系统设计中的抽象层次

同时, 嵌入式电子系统是与应用密切相关的, 不同类型的系统往往对应于不同的计算模型, 需要使用不同的设计语言进行描述。也就是说, 按照传统方法设计当今的 SoC 需要靠多种语言来完成: 系统描述语言, (如 Matlab); 硬件描述语言 (如 Verilog 或 VHDL), 还有硬件验证语言 (如 Vera-e 语言) 以及软件编程语言 C/C++、Java 等。与此同时业界关于各种语言的融合和派生问题由来已久, 但一直没有一个统一语言被业界普遍认可。

1 系统级设计语言的发展

21 世纪初期是系统级设计语言发展变化最迅速的时期。各种系统设计语言如雨后春笋, 已经出现的系统级设计语言可以分成两大类:

第一类是通过经典语言的扩展得到的语言, 如 SystemVerilog。SystemVerilog 在向高层次发展方面, 对原来的 Verilog 进行了根本性的修改。它混合了 Verilog、C/C++ 和 SuperLog, 给设计者提供了最强的能力。SystemVerilog 是对于 IEEE 1364 2001 Verilog 的扩展, 主要扩展的目的是使得 Verilog 语言能够支持大规模的设计并达到更高级的抽象。这类对传统语言扩展的方法的优点是有利于设计者的平稳过渡。

第二类是利用软件领域的语言和方法, 如 C/C++

收稿日期: 2007-01-30

作者简介: 王兆菊 (1983-), 女, 河南封丘人, 硕士研究生, 研究方向为嵌入式计算机系统软硬件协同设计技术; 龚龙庆, 研究员, 研究方向为嵌入式计算机应用。

+, Java, UML 等等。这类语言有 SystemC, SpecC, HardwareC, HandelC 等。可以把这些语言分成两类:一类是在标准 C 语言上进行扩充,以 SpecC 为代表;另一类是利用 C++ 的可扩充性,以 SystemC 为代表,它提供一组硬件的基本元件,这些元件都可以扩充,以便在更高的层次上支持硬件。

2 系统级设计语言的设计能力

在进行具体的系统设计过程中,选择所用的语言时,不但要考虑设计所处的抽象层次,同时还要考虑所选择的设计语言是否能够捕获系统的需求、是否可以同时能够描述硬件和软件,以及能否进行功能验证和描述测试激励等等。

VHDL 语言的抽象建模范围可以覆盖从行为级一直到精确的门级;Verilog HDL 语言的描述范围则一直到晶体管级;SystemVerilog 在扩充 Verilog 的基础上,可以进行软硬件描述、进行功能验证等;Vera, Sugar, e 等语言作为系统级验证语言,主要用于功能验证和编写测试激励;SystemC 在扩充 C++ 的基础上,可以同时描述系统的软件和硬件,直至系统的 RTL 级描述,同时 SystemC 的验证库 (Verification Library) 可以进行相应的验证和激励的编写;Matlab 一般则是在系统开发的初期捕获系统的需求,开发相应的信号处理算法^[1]。图 2 是几种系统设计语言在设计能力方面的比较。

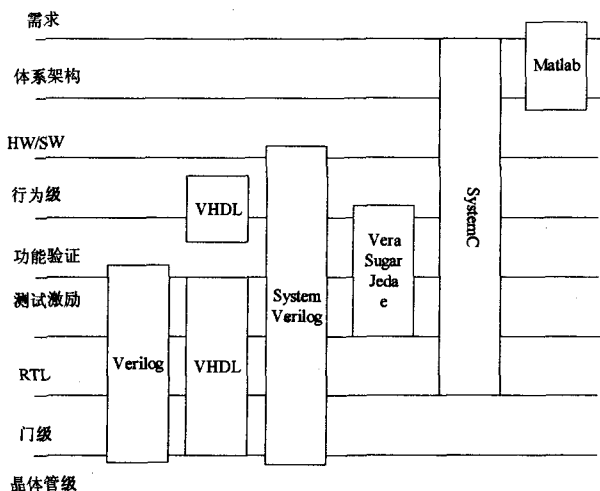


图 2 几种设计语言在设计能力方面的比较

但是,这并不意味着一种语言不可以用在不同的应用中。例如, SystemC 可以用于验证平台和描述 RTL 结构^[2],而 SystemVerilog 也可以用于编写高层次的事务处理级建模。但是,每一种语言只有在用于自己的重点应用时,才能达到最佳的效率,这点对于复杂的项目特别适用。在这些项目中,不同的任务分属于

不同的设计团队,对设计团队的技能要求也不同。注重实效的解决方案以及符合设计团队的多项技术要求的方法是同时使用 SystemC 和 SystemVerilog 来开发和验证当今设计流程需要的虚拟原型的事务处理级模型。所以,在选择具体的语言时,要根据实际需要选择最合适的。

3 具有代表性的几种系统级设计语言

3.1 SystemVerilog

Accellera 标准组织提议的 SystemVerilog^[3]是一个新的统一设计语言,很适合 SoC 的设计,它是对 Verilog 语言的高层次扩展。这些扩展为 Verilog 提供了强有力的增强,比如: C 语言的数据类型、结构体、合并和未合并数组、接口、声明等。SystemVerilog 是 C/C++、Superlog 和 Verilog 的混合体,极大地扩展了抽象结构层次的设计建模能力,它还把设计和验证融为一体, Accellera 称它为 HDVL (Hardware Design and Verification Language),也就是说,它将硬件描述语言 (HDL) 与现代的高层次验证语言 (HVL) 结合了起来。由于拥有这样的概念以及它与 Verilog 的向上兼容性,使其对于进行当今高度复杂的设计验证的验证师具有相当大的吸引力。上述这些特点,以及 SystemVerilog 是一项得到了所有主要 EDA 供应商支持的 IEEE 标准的事实,使得 SystemVerilog 实质上成为了硬件设计和验证的首选语言。

Cadence 日前宣布其基于 SystemVerilog 的验证解决方案得到了迅猛发展,用该语言进行试验的客户增加到了 150 家,他们或者将该语言应用于创建功能原型项目,或者应用到主流产品开发。该公司将 SystemVerilog 的加速发展归结为多种原因,包括 EDA 供应商对 SystemVerilog 语言支持的大幅提升、拓展的多语言功能、Cadence Incisive Plan-to-Closure Methodology 以及更新的高级 SystemVerilog 验证技术。

Cadence 的一份用户调查报告显示,目前客户正在使用 SystemVerilog 的不同功能。在这些客户中,大约 57% 正在使用设计构建, 60% 正在使用断言, 还有 57% 正在使用测试平台构建。150 家正在使用 SystemVerilog 搭配 Cadence Incisive 验证平台和解决方案的客户中,有超过 50% 用于主流产品开发。

3.2 SystemC

SystemC^[4]就是在 C++ 中加上硬件类库和仿真核。一个熟悉 C++ 语言的用户只要了解类库中各种类引入的语义,就可以用 SystemC 编程。用 SystemC 可以在不同抽象层次描述系统,在系统的最高层可以用 C/C++ 描述系统的功能和算法;在系统的硬件实

现部分可以在行为级到 RTL 级用 SystemC 的类来描述;系统的软件部分可以用 C/C++ 语言描述。而且系统的不同部分可以在不同的抽象层次上描述,这些描述在系统仿真时可以协同工作。用 SystemC 不但可以描述要开发的系统本身,还可以描述系统的测试平台(Testbench)以提供测试信号用于系统的仿真。

SystemC 已于 2005 年 12 月作为 IEEE1666 实现标准化,而且已经宣称支持 OSCI TLM 标准的 EDA 供应商包括 Atrenta、Cadence、Calypto、Celoxica、ChipVision、CoWare、Forte 设计系统公司、明导资讯公司、Summit、Synfora 和新思公司,以及开放内核协议国际合作伙伴(OCP-IP)总线标准组织。SystemC 的用户范围也相当广泛,得到了包括意法半导体(ST Microelectronics)、荷兰皇家飞利浦电子(Royal Philips Electronics)、德国英飞凌科技(Infineon Technologies AG),以及 IBM 等大型公司开发人员的支持。关于 SystemC 的典型使用情况,根据最近有关 SystemC 的出版物以及各项调查中得知,SystemC 用户中的绝大多数正在使用这种语言来执行建模(68%)、体系架构开发(68%)、事务级建模(56%)和软硬件协调仿真(56%)。

3.3 SpecC

SpecC 是由加利福尼亚大学欧文分校提出的另外一种基于 C/C++ 的硬软件描述语言体系结构^[5],得到了日本东芝、日立等多家公司的支持。SpecC 是为系统级设计量身定做的语言,在标准 ANSI C 的基础上,引入了一些系统级设计的概念,加上为了建模硬件行为所必须的一些扩充。SpecC 力求成为一个可综合的语言(这个与 VHDL、SystemC 有很大的不同,后两个是面向模拟的,所以,在综合的时候,必须用可综合子集),未来发展的目标是开发出一套综合工具,可以直接综合出用 SpecC 写的模块。由于其完整而先进的设计方法学以及配套工具,在 2000 年 SpecC 已经被日本 IC 界作为业界标准。

4 设计语言特点比较

4.1 Verilog HDL 和 VHDL

Verilog HDL 与 VHDL 是目前两种最常用的硬件描述语言,同时也是 IEEE 标准化的 HDL 语言。归纳起来,它们主要有以下几点不同:

1)从推广过程来看,VHDL 偏重于标准化的考虑,而 Verilog HDL 与 EDA 工具的结合更为紧密;

2)与 VHDL 相比,Verilog HDL 的编程风格更加简洁明了、高效便捷;

3)目前市场上基本所有的主流 EDA/ESDA 工具都同时支持这两种语言,而在 ASIC 设计领域,Verilog

HDL 则稍占优势。

4.2 SystemC 和 SystemVerilog

目前 SystemVerilog 和 SystemC 是适合高级设计与验证使用的最重要语言,业界对 SystemC 或 SystemVerilog 哪个更适用于系统设计和验证有巨大的争论,通过支持面向对象的编程方法,这两种语言都能对更高层的抽象进行建模,而且都包含像随机化这样的验证组件。但是两者之间也有些区别,例如 SystemC 完全支持 TLM,包括在高层抽象进行描述、记录和事务浏览;而 SystemVerilog 不完全支持 TLM,但它支持声明、功能覆盖和改进的受限随机测试生成功能,因此也具有显著的验证优势。这两种语言在功能上多少有些重叠,因此用这两种语言建立系统时可以做到平滑转换。

SystemVerilog 是许多现成的专有和标准验证与设计语言积累的结果,因此它能与现有的 HDL 环境无缝集成,加快高级功能验证的构建,实现缺陷检测的自动化和设计的完全覆盖;它能改善建模质量,使之具有更好的可视性和更少的缺陷,并且能够通过支持受限随机测试、自动化、声明、覆盖和测试平台复用改善测试平台架构的质量。

SystemC 能让工程师在更高抽象层捕获设计需求,并使用包括软件在内的高层测试策略执行验证。系统级测试平台也可以在 RTL 级复用,实现验证产能的提高。SystemC 包括一个功能强大的并发建模层,非常方便创建无时序、并发的系统级架构模型。因为 SystemC 是 C++ 的扩展,因此它有许多固有属性,如类、模板和多种继承性,这些属性使之本身就非常适合建立功能验证的可复用事务级组件。用于开发软件的同一可执行平台常被用于整个系统的验证。这种验证在项目初期就会发生,而事务级模型可以成为整个系统非常好的参考。

就 SystemC 和 SystemVerilog 这两种语言而言,SystemC 扩展了 C++ 在硬件方面的适用范围,SystemVerilog 扩展了 Verilog 在基于对象和验证平台方面的使用范围,而这两种语言均支持诸如信号、事件、接口和面向对象的概念,但每一种语言又有自己明确的应用重点:

1)SystemC 对于体系架构开发编写抽象事务级(TL)模型,或者对执行建模来说最为有效。特别是对于具有很强 C++ 实力的团队和基于 C/C++ IP 集成要求(如处理器仿真器),以及为早期软件开发设计的虚拟原型来说,更是如此。

2)SystemVerilog 对于 RTL 抽象模型和先进的验证平台的开发来说最有效率,因为它具备了执行这方

面任务所需的基础架构,例如受限随机激励生成,功能覆盖,以及断言等等。

3)SystemVerilog 显然是描述最终的 RTL 设计本身的首选语言,不仅在于其描述真实硬件和断言的能力,还在于其对工具支持方面的考虑。

4.3 SystemC 和 SpecC

SystemC 基于 C++ ,由 OSCI(Open SystemC Initiative)发布,主要由欧美的大量嵌入式系统设计相关企业支持;SpecC 基于 C,由 STOC(Spec Technology Open Consortium)发布,主要由日本企业支持。两种语言的设计初衷类似,都是为了能够解决嵌入式系统,尤其是 SoC 设计中传统规范语言的种种弱点。

虽然 C/C++ 在描述软件和系统级(算法级)规范时被广泛使用,但是由于 ANSIC C++ 缺乏对硬件电路的描述能力,因此在以它为基础设计嵌入式系统规范语言时,其语法和语义必需针对硬件模块进行扩充。对于建模而言,扩充的结构应具有抽象硬件模块的能力;对于系统的模拟而言,扩充的结构执行时的行为应等同于对应硬件。为了实现这些要求, SystemC 和 SpecC 采用了完全不同的策略: SystemC 的设计基于库,而 SpecC 基于语言。

两种语言各有优缺点,下面从几个方面对它们进行比较:

- 1)从易用性的角度来看, SystemC 优于 SpecC;
- 2)从扩充性的角度来看, SystemC 优于 SpecC;
- 3)从可互操作的第三方工具的支持情况来看, SystemC 优于 SpecC;
- 4)从综合的角度来看, SpecC 优于 SystemC;
- 5)从模拟的角度来看, SpecC 优于 SystemC^[6]。

4.4 小 结

研究资料表明,具有较高的抽象能力,同时能体现出硬件设计中的信号同步、时间延迟、状态转换等物理信息的语言,才能给工程师提供一个系统级设计的公共基础平台。在我们常用的设计语言中, C/C++ 和 Java 等高级编程语言有较高的抽象能力,但由于不能体现硬件设计的物理特性,硬件模块部分需重新用硬件描述语言设计,使得后续设计缺乏连贯性;而 VHDL/Verilog 最初目的并不是进行电路设计,前者是用来描述电路的,而后者起源于板级系统仿真,因此它们并不适合进行系统级的软件和算法设计,特别是现在系统中的功能越来越多地由软件来完成时。

表 1 是几种系统级设计语言和 C/C++ 在行为和结构描述、并发性、中断执行和同步设计等方面的对比。

表 1 具有代表性的几种设计特点对比

	C	C++	Java	VHDL	Verilog	HardwareC	SystemC
行为	○	○	○	○	○	○	●
结构	○	○	○	●	●	●	●
并发	○	○	◎	●	●	●	●
同步	○	○	◎	●	●	●	●
中断	◎	●	●	○	●	○	●
同步	○	○	◎	●	●	●	●
时序	○	○	○	●	●	◎	●
状态转换	○	○	○	○	○	○	●
复合数据类型	●	●	●	●	●	◎	○

注:○为不支持;◎为部分支持;●为支持

5 总 结

目前,嵌入式系统正处于一个新的发展阶段,系统复杂度的增加提高了设计的难度,而更大的市场压力则要求更短的产品上市时间,开发一个复杂系统的难度越来越大。同时系统设计语言的发展也是日新月异,出现了众多的设计语言。在选择具体的设计语言时,有时候很难做出选择。文中就这些语言的特点、设计能力等方面给出了简单的对比,希望能给大家一些有益的参考。

我们相信,未来的系统级设计语言必然是结合传统硬件描述语言和当代软件工程技术,具有完备的类型系统,支持继承、多态等而面向对象技术和高抽象级 IP 交换,并方便高层综合的设计语言,而基于 C/C++ 的系统设计语言(如 SystemC)和过经典语言的扩展得到的语言(如 SystemVerilog)则是目前较好的两种选择。

参考文献:

[1] Black D C, Donovan J. SystemC: From the ground up[M]. Boston: Kluwer Academic Publishers, 2004.

[2] Bhasker J. A SystemC Primer[M]. Allentown: Star Galaxy Publishing, 2002.

[3] Accellera. SystemVerilog 3.0: Accellera's Extensions to Verilog[M]. Napa, California: [s. n.], 2001.

[4] SystemC 用户手册[EB/OL]. 2002-07. www.systemc.org.

[5] Gajski D D, Zhu Jianwen, Dömer R, et al. SpecC: Specification Language and Methodology[M]. Boston: Kluwer Academic Publishers, 2000.

[6] 熊 悦, 龚育昌, 赵振西. 嵌入式系统规范语言[J]. 计算机研究与发展, 2003(3): 493-499.