基于VHDL的系统规范语言的研究*

董社勤 高国安 陈 爽

(哈尔滨工业大学计算中心 哈尔滨 150001)

摘要 将VHDL语言应用到嵌入式系统的设计提供支持是扩展VHDL对系统级设计支持的一种研究途径.本文考察了几个典型的对系统级设计提供支持的规范语言,它们都将VHDL集成到自己的设计方法之中.在此基础上,作者总结了支持系统级设计的规范语言的基本要素,指出将形式描述方法与VHDL相结合可提供一种更有效的设计支持工具.

关键词 嵌入式系统,系统规范,规范语言,系统级设计.

中图分类号 TP312

反应式系统(Reactive System)是与环境保持密切相互作用的一类系统,在任何时候,系统内都有几个任务并发地执行,系统随时准备接受来自于环境的输入,并保证在要求的时间内予以响应.反应式系统亦称为嵌入式系统,其应用极其广泛.如何用统一的表示形式对这种包含了软件和硬件及其复杂的环境成分的系统的行为进行描述正成为计算机科学的一个新的研究领域.对环境成分的建模要求有关学科如机械学、机器人学、航空电子学等的专业知识.另一方面,许多计算机科学中已有的工具和方法可以通过改进而应用在嵌入式系统的设计中.在系统设计初期即系统级设计阶段,根据设计要求为这种系统建立起一个概念模型是至关重要的,作为文档它不但要被具有不同知识背景的设计参与者所理解,而且应成为嵌入式软件和硬件设计的良好起点.

VHDL(very high speed integrated circuit hardware description language)作为工业标准语言已获得了绝大多数商用CAD工具的支持,并被广大硬件设计者所接受.[1~4]它提供的丰富手段可以方便地产生能够运行的硬件系统描述模型.VHDL有多个抽象级别(从部分开关级到部分算法级[5])的描述能力,对这些抽象级别之间的任何方式的混合描述,它都可以提供模拟支持.

将系统级建模方法与VHDL相结合,既可以利用VHDL丰富的支持工具实现系统硬件部分设计的自动化,又可以利用VHDL的模拟环境对系统的完整模型进行模拟分析,从而为嵌入式系统的设计提供快速原型设计方法的设计支持.

---

* 作者**董社勤**,1964年生,博士,助研,主要研究领域为专用实时系统的建模与分析技术,VHDL,实时计算机图象生成,嵌入式系统.**高国安**,1937年生,教授,博士导师,主要研究领域为CIMS,机电控制及自动化.**陈爽**,1949年生,研究员,主要研究领域为实时图象生成系统,高速计算机结构,VHDL.

本文通讯联系人:董社勤,哈尔滨150001,哈尔滨工业大学计算中心

本文1996-06-20收到修改稿

本文首先考察了几个典型的基于VHDL的系统规范语言.在此基础上,作者总结了支持系统级设计的规范语言的基本要素.本文探讨的是VHDL的一种全新的应用研究领域.目前在我国,VHDL仅限于初期的应用开发阶段,仅为少数人知悉.希望本文能引起相关研究者的兴趣.

1 基于VHDL的系统规范语言

1.1 SpecCharts

SpecCharts[6]将整个系统用StateCharts来描述[7~9],而将系统中要完成的具体操作直接用VHDL来描述.SpecCharts的设计者认为在主要以系统行为为描述对象的系统级,规范语言应提供对行为分解,即行为层次模型的支持.一个行为被定义为一组动作和一组描述每一动作什么时候发生的条件.动作则是可以改变系统值(如变量值或系统状态)的任何事件.一个行为可表示为3种形式之一,即①并发子行为;②顺序子行为;③叶行为(VHDL程序).行为层次模型是指①一个行为的动作可定义为一组并发或顺序子行为;②在任何时候一个行为能激活和/或制止一个子行为的发生;③一个行为能向其它行为表明它已完成.

SpecCharts用方框代表行为,用有向弧表示顺序行为之间的顺序关系,虚线则将方框内并发的行为分隔开来.在任何时候,一个行为或是活动的,或是不活动的,一个活动行为或是正在执行的,或是已经完成的.对于执行,如是叶行为,便执行其代码;如不是叶行为,它便激活和/或撤销某些子行为的执行.行为的完成是指一个行为已完全执行完毕,但尚未被其父行为撤销:如是叶行为,则其代码执行已结束;如不是叶行为,则其所有的子行为都已变为不活动,控制到达一个特殊的"停止点",从该点不可能有子行为再被激活.SpecCharts用有向弧去撤销一个行为的执行.转换仅在同一父状态的同一级子状态之间发生.

为适应系统级通信和同步描述的要求,2个并发的行为可以通过一个具有相关协议的通道实现通信.协议可以用一个VHDL程序来定义.

SpecCharts除了用图形方式描述系统外,还可以用文本方式.在设计阶段的任何时候都可以将SpecCharts的描述翻译成VHDL,从而利用VHDL的支持环境进行模拟验证.

1.2 VHDL/S

形式化方法有适于表达系统概念模型的抽象形式,并有相应的抽象设计方法以及建立在形式基础上的证明法则.如果通过一个友好的用户界面将形式化方法与系统后续的设计过程联系起来,则有可能部分地解决系统级设计的设计支持问题.

VHDL/S[10]以VHDL的体系结构描述为结合点将VHDL,基于StateCharts[7~9]的规范语言,时序图和时态逻辑(Temporal Logic)集成到一个语言框架之中.这些语言范例具有共同的关于时间流的语义基础,因而设计者可以采用最恰当的描述风格描述系统的一个部分.设计者对随时间响应的系统行为的描述不会因为在几个语言范例之间的前后切换而受到影响.在VHDL/S中,一个部件的描述可由VHDL实体声明给出接口描述,其体系结构的描述可选择下面4种方式之一:①一个VHDL的结构或行为体描述;②一个时序波形图方式的描述;③一个基于StateCharts的描述;④一组时态逻辑公式.[11]

最顶层总是VHDL实体声明,这样可以使现有的VHDL元件库无需修改便可以集成

到 VHDL/S 中.

VHDL/S 选择了一个一阶时态逻辑语言来表达初期的系统设计要求,它可以将关于一个部件的设计表述与其环境的表述区别开来,因而可以支持模块化的证明策略,从而利用现有的仅能处理有限规模问题的定理证明系统.用时态逻辑直接描述实际系统是非常困难的,因而 VHDL/S 用传统的时序波形图来作为时态逻辑的图形表示.通过图形符号波形图可以表示事件间不同的时态关系,也可以表示事件间更抽象的半序和因果关系,还可以用与 VHDL 语法一致的注释来表明不同波形区域的数据依赖关系.一个时序图可集中描述一个部件的一个方面,这样一个部件的完整描述就是几个时序图的合取.时序图转换为时态逻辑描述后可用于部件设计正确性的证明(相对于设计要求的正确性)以及系统有关性质的推导.[12]时序图向 VHDL 的转换是通过首先将时序图转换为 LOTOS 的一个扩展表示 T-LOTOS[13],然后从 T-LOTOS生成 VHDL 程序.[14]

VHDL/S 的基于状态的规范语言是建立在 D. Harel 的 StateCharts 之上.一个状态图由一组状态及状态之间的转换组成,状态可以有子状态,以 AND 和 OR 两种方式组成,分别表示并行组织和顺序组织.一个 AND 状态是活动的,则其所有的子状态也是活动的;一个 OR 状态是活动的,则其仅有一个子状态是活动的.一个状态图的最终结构是一个 AND/OR树,其叶子被称为基本状态.转换弧用形如"事件[条件]/动作"的标志来标记,其意义是当事件发生时且在当前数据空间的解释下条件成立,则从源状态进入目标状态,同时完成动作.一个转换可有多个源和目标且可以连接层次结构中处于不同级别的状态(不同于 SpecCharts).动作则由顺序 VHDL 语句组成.

互不相关的并发部件中的转换可以同时发生,组成系统的一个步骤.步骤可以认为是"瞬时的",时间以有限长度的片段的方式仅在相继的步骤之间流逝.这与 VHDL 的时间模型是一致的.[15]

为了使用模型检验法证明一个状态图描述满足一个用时态逻辑公式表述的设计要求,VHDL/S 首先将 StateCharts 翻译成 Petri 网,然后再将网表示翻译成 BDD(binary decision diagram)表示进行模型检验.[16]状态图向 VHDL 的转换是比较直接的.

1.3　TL

TL (task level specification system)[17]将一个嵌入式系统看成一组任务,用 TL 语言描述的系统模型被翻译成有色 Petri 网进行分析,从而产生任务并发、任务激活、通信频率、消息长度、任务同步等信息.[18]然后设计者可以交互地将系统模型划分成硬件部分和软件部分.硬件部分可以翻译成 VHDL,从而利用 VHDL 的支持环境.由于整个系统都使用 TL 语言描述,因而可以全部翻译成 VHDL 进行全系统的模拟.

一个系统便是一组任务.一个任务或是一组新任务的一种安排的描述或是一组顺序语句.当用顺序语句描述一个任务时,第一个顺序语句通常是一个同步、通信或互斥语句,每个任务只能包含这 3 个语句中的一个.这些语句的执行由其它任务产生的触发来启动,其它顺序语句则对变量、私有数据和消息数据实施操作.

TL 语言的通信和同步用通信和互斥语句来描述,它可以描述通信和同步的超时情况,可以定义通道,通道可以共享,可以为每一个通道描述一个协议.

为隐含信息或便于构造模型可将一组任务合并成一个任务.TL 有 4 种任务组织方式,

即并发、并行、相继和中断. 一个不包含其它任务的任务为叶任务,它只描述顺序型的数据处理. 一个并发任务可由一组并发任务组成,每个并发任务由相继任务组成,一个相继任务通常有一个前驱和一个后继任务,每个相继任务是一个并行结构任务或一个叶任务. 并行结构任务是一组当其执行条件满足时可并行执行的一组描述(任务),它只执行一次. 一个并行结构任务中的一组并行任务后接一个特殊任务,当所有并行任务结束之后它便被启动.

TL 可以表示任务的正常中断和非正常中止. TL 的任务被翻译成 VHDL 的块,每个 VHDL 的叶子块对应 TL 的一个叶任务,它包含一个以 wait 语句起始的过程,该 wait 语句相应于叶任务的启动条件.

1.4 SIERA

SIERA(system integration of embedded real-time application)[19]是为象机器人控制器这一类复杂的实时嵌入式系统提供的快速原型支持工具. SIERA 将这类专用实时系统看成一个静态的(数量有限且不变的)顺序过程的层次网络,这些过程并发执行并通过严格定义的通信机制交互作用. 每个过程有自己的输入输出端口,一个输出端口通过通信通道连向一个输入端口. 每个通道有其缓冲深度和类据类型,缓冲深度为零表明同步通信,缓冲深度大于零表明异步通信. 每个端口有自己的数据类型和通信协议,端口通信协议描述了当一个通道满或空时过程的行为. 过程网络模型将传感器、执行机构等使得系统可以与外部物理世界交互作用的子系统看作为预先存在的过程,这些过程与系统的其它部分并行地运行在它们自己的电气或机械部件上.

为实现快速原型设计,SIERA 提供了两组适于系统抽象模型模拟的 VHDLpackage,一个用于支持过程网络中的通信机制,另一个用于对传感器和执行机构等连续时间子系统进行建模. SIERA 将 VHDL 的 Package 扩展成可以带参数的 Package 模板,对于一个特定的通道或连续时间子系统,可以通过一个预处理程序将特定的类型参数或系数矩阵代入相应的 package 模板,从而产生特定通道或连续时间子系统的 VHDL 描述.

对于用常微分方程描述的连续时间子系统,SIERA 用一个 VHDL 过程为每一个子系统实现了一个专用模拟器,并用一种特殊的通信机制保证这类 VHDL 过程网络能够有效地以分布的方式求解一组常微分方程. 一个通用的采用龙格-库塔法求解常微分方程的 VHDL 过程作为产生这些专用模拟器的 package 模板.

1.5 SADT

在制造工业中,只要有了一个部件的蓝图或设计图纸,任何一个熟练的技师都可以按照这一图纸准确无误地加工出这一部件. SADT(structured analysis and design technique)[20]是 70 年代末由 SofTech 公司的 Douglas Ross 受设计蓝图这一思想的启发提出的一个软件系统的描述方法. SADT 可以用功能方法或数据对象方法对系统建模. 功能方法也称之为活动(activity)建模,它以系统分解过程中的功能和活动作为建模的着眼点.

$IDEF_0$ 则是 SofTech 公司 1981 年在集成计算机辅助制造 ICAM(integrated computer aided manufacturing)项目中为美国空军物资实验室研制的需求分析和建模技术,它是建立在 SADT 子集之上的一个图形化方法. $IDEF_0$(iCAM definition zero)用一系列的层次地关连着的方框图来描述系统的功能分解. 在建模之初首先为系统产生一个代表着系统最高抽象级别的方框图,将整个系统表示为一个功能. 这个框图称之为环境模型,它表示了系统如

何与环境交互作用.然后对这一环境模型进行逐级分解,直至得到整个系统的描述.一个完整的 $IDEF_0$ 模型是由功能分解产生的一个层次型的以活动为结点的树状结构.

一个 $IDEF_0$ 图形描述由活动及其接口组成,活动用一个方框表示,称之为活动框,它代表系统的一个功能.接口由带箭头的连线表示,代表功能之间的相互作用.输入箭头从活动框的左边进入,代表该活动的输入数据.控制箭头从活动框的上方进入,代表控制信息.输出箭头从活动框的右边引出,代表该活动产生的数据.机制箭头从活动框的下方进入,代表负责完成该功能的人或设备.

一个活动如何和何时将一组输入转换为输出,在 $IDEF_0$ 中并没有提供描述方法.D. L. Eickmeier 等人为 $IDEF_0$ 扩展了决策表技术,该表将一个叶结点的活动的输入和控制作为前提,将输出作为在前提下产生的结果,可以在表中列举所有前提下产生的结果.这样就可以使一个用 SADT 描述的系统模型翻译成能够执行的 VHDL,从而实现软件规范的模拟验证,为软件系统设计提供快速原型设计支持.

2　系统规范语言的基本要素

综上所述,我们可以归纳出以下系统规范语言的基本要素:

(1) 系统模型的建立过程是通过对系统设计要求的分析和综合而提出系统总的行为方式的过程.规范语言作为模型描述手段应对这一过程提供支持.因而规范语言应包括:①关于系统的定义.如一个系统是一个设计实体(VHDL/S),一组任务(TL),一组过程(SIERA)等;②系统的基本对象及其性质.如 TL 的基本对象是任务,任务可以中断;③系统的组织原则,也就是系统基本对象间的关系,如父子关系,并发关系等.

(2) 基本对象是规范语言描述和分析系统的基本手段,是规范语言关于分析和解决问题的方法的具体体现,因而必须具有很强的概括能力以保证其适用范围.特别是在系统设计的早期阶段,基本对象应尽量与人们思考分析问题的概念一致.例如将一个复杂的嵌入式系统看作一组通过通信通道相互作用的并发过程,运用"并发过程"这一基本对象就可以很容易地描述系统与环境之间的交互作用.

(3) 系统的行为表现在 2 个方面,一个是对象之间的相互作用,另一个则是对象的内部行为.在系统设计的初期阶段,对象的内部行为可以隐含,外部行为则应有所说明.外部行为体现在对象之间的通信和同步上,因而规范语言应有表示与其基本对象相适应的通信和同步机制.例如 VHDL 的信号就不适合系统级对象之间的通信关系的描述,因而多数规范语言都采用了指定通道和通道协议的方式实现基本对象间通信的描述.

(4) 一个系统与环境的交互作用以及它自己内部基本对象之间的相互作用都有某种时间限制或要求,有某种时间参照,因而规范语言应有时间表示机制.由于系统设计初期的时间是一种逻辑时间,时间的前进以系统内的因果关系为参照,许多规范语言并没有很好地解决这一问题.

(5) 一个好的规范语言应提供对模型的确认手段,从而表明得到的系统的模型满足了设计要求.通常有 2 种方法来确认规范语言所描述的系统模型:一是动态模型执行即模拟,即将该规范语言描述的系统模型编译成一个可以执行的模拟器,从而检验系统对某些输入条件的响应情形;另一种方法则是基于某种数学理论的规范语言,如时态逻辑等所支持的静

态分析技术. 例如 VHDL/S 中一个部件的波形图描述可以自动转换为相应的一组时态逻辑公式(模型),然后利用时态逻辑证明法则证明该模型满足某一设计要求(表示为一个时态逻辑公式).

(6) 规范语言管理大型系统模型的能力取决于该语言所采用的模块化模式的抽象程度. 所谓模块化模式指的是系统基本对象及基本对象之间的关系和相互作用机制. 在系统设计初期应建立与系统的具体实现细节无关的抽象系统模型,因而采取的模块化模式应保持在高级别的抽象程度上.

(7) 一个典型的系统往往涉及范围很广的许多不同的方面,如一个机器人控制系统. 根据特定的需要,一个系统模型有可能由一组不同抽象级别的描述组成. 因而为系统级设计提供支持的规范语言应允许将多种语言范例集于一个共同的环境之中,它们在语义上的连接应可以作全系统的分析和结构考察. 例如 VHDL/S 的一个体系结构体可以采用 4 种语言范例之一来描述,它们之间有共同的关于系统时间流的语义连接.

(8) 规范语言的风格是影响系统设计效率的重要因素. 应提倡图形界面、可视化方法在规范语言设计中的应用.

3 结 论

嵌入式系统的描述及设计支持环境正成为一个日益活跃的研究领域. 在嵌入式系统的设计中,通用或专用集成电路及其外围支持电路发挥着至关重要的作用,因而将现有的硬件和软件的描述及设计支持方法统一起来,为嵌入式系统设计提供支持是一种必然的趋势. 由于 VHDL 的工业标准的地位和它在现代集成电路设计中的大量应用,以及众多商用工具开发者和研究者的大力支持,VHDL 必将在新一代设计自动化技术中发挥重要的作用.[1]

由于嵌入式系统往往是可靠性要求极高的一类实时系统,为了保证其设计的正确性,人们发展了众多的形式化描述方法及相关的静态分析技术,这些方法的应用往往要求有良好的数学基础,因而受到较大的限制. 如果将模拟分析技术作为形式化方法的一个补充,这不失为一个明智的选择. 事实上,模拟分析技术仍是当前分析验证模型的主要技术,且为广大设计者所熟悉. 为形式化方法配备专用的模拟器显然是一种既费力又没有太多好处的工作. 用 VHDL 去实现形式化方法的语义(或部分语义),从而利用 VHDL 的支持环境进行模拟分析,其益处是显而易见的. 一个既支持严格的形式证明和抽象分析技术又支持模拟分析技术的模型描述方法显然更受欢迎. 一个明显的例子就是本文所提到的 VHDL/S 所涉及的 LOTOS 语言,它是一种形式规范方法,在对其作了定量时间表示的扩展后得到 T-LOTOS,T-LOTOS 就可以翻译成 VHDL(当然 LOTOS 也可以转换成 VHDL).[13,14] 值得注意的是,系统模型向 VHDL 的转换并不是一一对应的,应根据应用的需要选取一个 VHDL 的子集.

作者采用 Henzinger 等人提出的描述模型[21]描述了一个实时图象生成系统(包含大量的硬件实现的并发或并行算法),并用 VHDL 实现了该模型的部分语义(最小延迟情况或最大延迟情况). 该模型支持时态逻辑分析方法.

参考文献

1 Deway A *et al*. VHDL special. IEEE D&T of Comp., June 1992.

2 Markowits M C. EDN hands-on VHDL design project(parts 1~6). EDN, Jan.~Mar. 1993.

3 Deway A *et al*. VHDL special. IEEE D&T of Comp., Apr. 1986.

4 IEEE Standard VHDL Language Reference Manual—std1076—1987, New York, IEEE 1988.
5 Rammig Franz J. Approaching system level design. In: Mermet J ed. VHDL for Simulation Synthesis and Formal Proofs of Hardware, Netherland, 1992.
6 Narayan S *et al*. System specification with the SpecCharts language. IEEE D&T of Comp., 1992, **DEC**.: 7～13.
7 Harel D. StateCharts: a visual formalism for complex system. Science of Computer Programming, 1987, **8(3)**: 231～274.
8 Drusinsky D *et al*. Using StateCharts for hardware description and synthesis. IEEE Trans. CAD-IC & System, 1989, **8**: 798～807.
9 Harel D. On visual formalisms. Comm. ACM, 1988, **31**: 514～530.
10 Helbig J *et al*. VHDL/S—integrating StateCharts, timing diagrams and VHDL. Microprocessing and Microprogramming, 1993, **38**: 571～580.
11 Manna Z *et al*. The temporal logic of reactive and concurrent systems: specification. Springer-Verlag, New York, 1992.
12 Fujita M *et al*. Logic design assistance with temporal logic. IFIP 7th Computer Hardware Description Languages and Their Applications, Tokyo, Aug. 1985.
13 Tiedemann W D *et al*. Introducing structure into behavioral description obtained from timing diagram specification. Microprocessing and Microprogramming, 1993, **38**: 581～588.
14 Kloos C Delgado *et al*. VHDL generation from a timed extension of the formal description technique LOTOS within the FORMAT project. Microprocessing and Microprogramming, 1993, **38**: 589～596.
15 Lipsett Roger *et al*. VHDL: hardware description and design. Intermetrics Inc. Boston, 1989.
16 Burch Terry R *et al*. Symbolic model checking for sequential circuit verification. IEEE Trans. CAD-IC & System, 1994, **13(4)**: 401～423.
17 Benders L *et al*. Task level behavioral description translation to IEEE VHDL. VHDL Forum, Marseille, 1991.
18 Benders L *et al*. Petri net modeling in embedded system design. CompEuro'92, 1992.
19 Srivastava M B. Rapid prototyping of hardware and software in a unified framework[Ph. D. thesis]. Univerisity of California, Berkeley, 1992.
20 Eickmeier D L. Formalization and validation of A SADT specification through executable simulation in VHDL. AFIT/GCS/ENG/91 D—6, 1991.
21 Henzinger T A, Manna Z, AND Pueli A. Timed transition systems, real time: theory in practice. Lecture Notes in Computer Science, Springer-Verlag, Berlin/New York, 1992, **600**: 226～251.

THE STUDY OF VHDL BASED SYSTEM SPECIFICATION LANGUAGES

DONG Sheqin　GAO Guoan　CHEN Shuang

(*Computer Center　Harbin Institute of Technology　Harbin*　150001)

Abstract　Applying VHDL to support the design of embedded system is a way to extend VHDL to support system level design. In this paper, the authors observe several typical specification languages which supporting system level design, all the languages integrated VHDL into their methodologies. Based on the observation, the authors summarize the basic key points of a specification language which supporting system level design, and point out that combining formal methods with VHDL can produce an effective design assistant tool.

Key words　Embedded system, system specification, specification language, system level design.

Class number　TP312