

专用实时系统的系统级设计方法^{*}

System Level Design Method of Application Specific Real-time System

董社勤 石教英

陈 爽

(浙江大学 CAD&CG 国家重点实验室 杭州 310027)(哈尔滨工业大学计算中心)

Abstract System level design method of application specific real-time system was discussed systematically in this paper. Key points of a specification language which can support system level design was also summarized.

Keywords Real-time system, Formal method, System level design, VHDL

1 系统级设计

复杂的计算机应用,如嵌入控制、信号处理、自动化、计算机实时图象生成等,所要求的系统往往是专用实时系统,且具有分布、复杂、并发等特点。一个专用实时系统通常包含了通用硬件、软件,专用硬件、软件,并通过一系列传感器,执行机构与环境发生交互作用。

在系统设计之初根据设计需求为这种系统建立一个概念模型是至关重要的。作为文档它不但要被来自不同的工程领域的设计参与者所理解,而且应成为专用实时系统硬件和软件设计的良好起点。系统的概念模型最终将转换成系统设计方案并付诸实施。为了获得对系统行为的深入了解并确认系统的设计和实现满足了系统的设计需求,应能对系统的概念模型及由概念模型得到的系统实现进行分析。

文[3]评述了多种系统建模方法,尽管这些方法多属数字电子系统设计领域,但随着计算机应用及自动化技术的日益发展,复杂的机电一体化系统已大量涌现。这类复杂系统的传统设计方法通常是将设计问题人工划分到不同的工程领域,系统的各个部分采用各领域独有的方法独立地加以设计,然而,随着产品生命周期的缩短,系统复杂性的提高以及市场竞争的加剧,一个更系统化的系统级设计过程是非常必要的,而且这样一个设计过程必须获得基于计算机的方法的支持,从而最终得到一个集成的并行工程(CE)环境。

欧洲设计自动化国际会议已开始将系统规范(或规格说明)作为一个独立的专题加以研讨,从发表的文章看,这些方法均是对已有的软硬件设计方法的继承和发展。由于这类机电一体化的系统往往应用在极高可靠性的场合,且在系统建模之初关注的是系统主要行为的抽象,因而形式化方法受到研究者的普遍重视。考虑到系统的设计要求来自不同工程领域的设计参与者的密切合作,模型描述的简洁、直观、易懂、精确等要求不容忽视,因而图示式规范语言的研究也受到人们的重视。将形式化方法与图示式规范语言相结合则产生了兼具二者之长的规范方法。

2 系统模型及系统级设计方法

形式化方法在软件工程领域的应用与研究已取得了丰富的成果,目前已成长为一个备受瞩目的研究领域^[2]。形式化方法在硬件系统设计与验证方面的应用亦受到研究者的重视,这二者之间的关系可解释如下:

程序不再被看作是输入到计算机中去的客体,也不再被看作是只有通过对各种输入数据集进行逐步的解释才能理解的“处方”,而被看作是适于数学分析与逻辑推理的静态正文。与正文类似的是电路图,而对程序的逐步解释则类似于从晶体管到晶体管的信号追踪,考虑到电路中的并发成分比比皆是,进行追踪是绝对徒劳无益的,可是模拟器仍按这种极不现实的范例在建造。计算可以表征为随时间变

^{*} 本课题得到航天预研基金资助。董社勤,博士后,副研究员。

化的状态,状态用相关变量的集合来表示。一个程序规定状态变换序列,状态可以用逻辑命题与谓词(亦叫做断言)来刻画,其变元就是程序变量,因而电路应该可用正文形式定义,而实际电子电路可以据此按照一组固定的规则得到。对物理现象的模拟仍然需要,但可以把它简化为单个时钟周期期间的信号传播,可以把它简化为有限的几种情况,正文形式必须用硬件描述语言给出。最重要的是,这种语言必须经过严格定义,可以使用数学变元。其用途不是实现模拟,而是进行证明。

可见,在系统级将一个系统的软件和硬件抽象到一个统一的形式框架之下是充分而必要的。但这仍然是不够的,只有将专用实时系统的工作环境、软件和硬件用一个统一的观点组织起来,才能实现真正意义上的系统设计支持。

试考虑一个典型的机器人控制系统的例子。它是一个多传感器的机器人控制器,也是一个实时系统,通过位置、力和视觉等传感器控制一个六自由度的机器人臂。该系统的视觉传感、运动控制以及执行机构接口等建立在一组专门定制的电路板系统之上。这些相互连接的电路板又连向一个单片计算机和一个工作站。该机器人系统既包括了通用和专用的硬件和软件,数字硬件和模拟硬件,又包含了机电部件,并与环境发生交互作用。机器人臂及其模拟子系统通常用一组微分方程来描述,软件子系统通常用具有通信通道和共享数据结构的一组过程来描述,硬件子系统也可以用一组具有通信机制的过程来描述。我们可以将这类系统进行化简,使其具有更清晰的抽象结构,如图1所示。

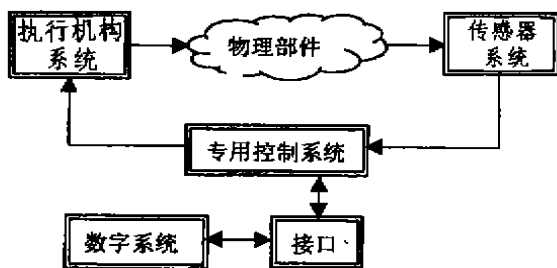


图1 专用实时系统的抽象模式

如果我们将过程建模的概念加以推广,在这类系统的概念建模阶段将执行机构系统和传感器系统也分别看作是一组过程,这些过程之间以及它们与专用控制系统之间通过一定的方式相互通信,则可以用一种统一的观点为这种系统建立概念模型。

通常对系统模型的分析有两种方法,一种是动态分析即模拟,另一种是建立在模型的数学基础之上的分析和证明,称之为静态分析。模拟分析有许多缺点,但它是分析验证模型的最通用方法,如软件工程中的快速原型方法等。静态分析可以克服模拟分析的不足,但它要求使用者有良好的数学基础,目前大多数的形式分析技术尚难以应用到实际系统的设计中,如果能用模拟方法实现形式模型的部分或全部语义,那么便可以为专用实时系统的分析提供更为有力的支持。作者认为,一个既支持模拟分析又支持静态分析的概念模型可为用户提供更多的可供选择的分析手段。图2概括了我们所提出的方法。

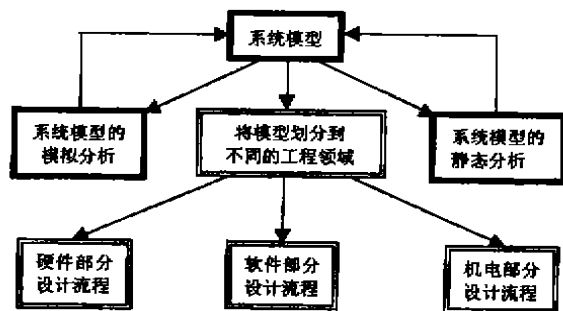


图2 专用实时系统的设计流程

首先我们希望为这一类系统建立起一个图示式的形式化模型描述手段,并初步建立起基于其形式基础的静态分析技术,最后用一种模拟语言实现对系统模型的模拟,从而为设计者提供快速原型设计支持。这些内容包括了如图2中黑线框中的几个部分。以下讨论以硬件系统为主,但它不难推广到软件系统和环境的建模。

通常对系统模型的模拟使用高级模拟语言,或使用编程语言编制专用模拟器。目前已成为IEEE标准的硬件描述语言VHDL(Very high speed integrated circuit Hardware Description Language)可以对数字硬件系统进行多个抽象级别的描述,并支持这些级别之间任意混合方式描述的模拟。VHDL的面向过程的观点使得它可以非常方便地表示并发表行为。同时围绕VHDL的各种自动化设计工具的研究和开发使得VHDL在新一代设计自动化技术中发挥着至关重要的作用,因而选用VHDL作为模型的模拟语言可以实现系统硬件部分设计的自然过渡。通过这一方式可以将VHDL的设计支持能力扩展到系统级。这从某种意义上说是VHDL的一种非传统应用,迄今为止,大多数的VHDL应用仍局限在

较低级的硬件建模方面。总之选用 VHDL 硬件描述语言是基于如下几点:①专用实时系统中存在大量的专用集成电路及专用数字电路系统。②VHDL 是 IEEE 的工业标准,获得了广泛的支持,在新一代设计自动化技术中占据着极其重要的位置。③VHDL 的应用已扩展到硬件设计以外的领域,包括软件系统快速原型设计、机电系统快速原系统设计等。

3 基于 VHDL 的专用实时系统规范语言的研究

在数字硬件设计领域有一种被广泛接受的六级抽象设计模式^[4],为了引出关于系统级的概念,我们从行为的角度自下而上来介绍这六级模式。在该模式的最低一级是电子电路级,电路的行为使用微分方程来刻画。开关级则主要用于对 MOS 电路设计的抽象,它将一个晶体管抽象成一个理想的通断开关,开关之间的连接则通过离散电容。电路的行为表现为连续时间轴上的离散值。门级是数字电路设计的传统一级,它以布尔代数为基础描述与时间无关的电路行为。采用寄存器、ALU(算术逻辑部件)等作为基本设计单元,将这些部件之间的操作看作是寄存器之间的数据传输,在传输过程中数据可以被改变,数据传输以时钟沿计数为标志,这就是所谓的寄存器传输级的抽象。在算法级则从系统控制器的角度考虑系统的组织,以并发算法描述系统,可不必强调操作的精确描述而只须在必要时说明有关操作的逻辑时间,当然也可以以时钟沿计数来给出精确时间,大量的操作作为内部功能而可以被隐含。最高的系统一级则是对指令系统算法实现的一种抽象,系统常被看作是一组能提供特定服务功能的相互协作的处理器。这些处理器通常有自己的指令系统和与其它子系统交互作用的协议。系统行为的描述可采用集中式和分布式。集中式描述用一个高度并发的算法描述整个系统的行为。在分布式描述中,不同部件根据特定的状态和事件决定它对其它部件的服务请求。集中式可以看作是算法级的抽象,分布式则看作是对寄存器传输级的抽象,在系统级,时间的前进是某种原因作用的结果。

在自顶向下的设计方法中这六级抽象模式的各级之间可以认为是一种从上到下的求精关系。在现实应用中,纯粹的数字系统非常少见。通常一个完整的系统由来自各个不同领域的部件组成。例如一个数字控制的汽车发动机便是由机械部件、模拟电路、数字硬件以及运行在数字硬件之上的高级软件程序

组成,它们协作完成一个从热力学和机械方面来看是高度复杂的操作。因而一般意义上的系统级应考虑到系统的所有这些方面。在传统的系统设计中,最初的系统设计方案往往集中了多个领域的专家知识,系统的规格描述即规范以及系统模型往往由易于理解和交流的形式如状态图、框图、文字、公式等表示。系统的规格描述是对系统设计要求的表示,而系统模型则是相对于规格描述的最初的系统设计方案,在这个方案之上对系统进行划分,明确各子系统的设计要求以及各子系统的接口定义从而使系统设计推进到各子系统的设计。这一过程可以反复进行直到系统设计完成。

为了对以上系统设计过程提供有效的支持,人们开发了多种系统描述模型和支持该描述模型的分析验证工具^[4,5]。例如对于数字硬件系统来说 VHDL 硬件描述语言就可以支持从部分开关级描述到部分算法级描述的设计抽象。随着数字控制技术,集成电路技术以及计算机系统的广泛应用,以数字系统为核心的复杂系统大量涌现,这就是所谓的嵌入式系统。这类系统往往有很高的可靠性和实时响应要求。为了对这类系统提供设计支持并继承数字系统设计领域的丰富成果,人们开展了基于 VHDL 的技术研究。例如将 VHDL 发展为系统性能的分析工具^[6]。而以 VHDL 为基础的系统模型描述语言即系统规范语言的研究更受到人们的广泛重视,它可以使 VHDL 的描述能力真正扩展到对系统级设计提供支持。

SIERA(System Integration of Embedded Real-time Application)^[7]是针对嵌入式控制等复杂的一类专用系统提出的一个快速原型 CAD 支持系统。快速原型设计的一个重要步骤是对系统高级规范描述的模拟。在这一级别进行模拟通常采用专用高级模拟语言或使用编程语言编制专用的模拟器,SIERA 选用 VHDL 来模拟系统规范及与系统交互的环境。SIERA 将这类专用实时系统中的通用软件、硬件,专用软件、硬件看成一个静态的由一组顺序过程组成的层次网络,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道连向一个输入端口。每个通道有其缓冲深度和数据类型,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。每个端口有其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。过程网络模型也将传感器、执行机构、机械子系统等看作为

预先存在的过程,使得系统可以与外部物理世界交互作用,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之上。为实现快速原型设计,SIERA 提供了两组适于高抽象级别系统模拟的 VHDL Package,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、执行机构等连续时间子系统的建模,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。一般采用龙格-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。

4 支持系统级设计的系统规范语言

综上所述,我们可以归纳出以下支持系统级设计的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,规范语言作为模型表达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)、一组任务、一组过程(SIERA)等;②系统的基本对象及其性质,如基本对象是任务,任务可以中断等;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体现,因而必须具有很强的概括能力以保证其表达能力和适用范围。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实时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及其与环境之间的交互作用。

3) 系统的行为表现在两个方面,一是对象之间的相互作用,二是对象的内部行为。在系统设计的早期阶段,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。

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。由于系统设计的早期阶段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。通常有两种方法用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,从而检验对某些输入条件的响应情形,如 SIERA,另一个则是使用基于某种数学理论的规范语言如时态逻辑等的静态分析技术。例如用一组时态逻辑公式描述系统需求(模型),然后利用时态逻辑证明规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。在系统设计的初期,应建立与系统的具体实现细节无关的抽象系统模型,因而采取的任何模块化模式应保持在高级别的抽象程度上。模块化模式指的是系统基本对象及基本对象之间的通信机制。

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,因而任何为系统级设计提供支持的规范语言应允许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可以作全系统的分析和结构考察。如 SIERA 通过 VHDL 的过程及其定义在过程之间的特殊通信机制将软件、硬件过程和常微分方程描述的连续时间子系统的过程集成在一个系统模型之下,从而实现了快速原型设计的支持。

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。

结论 形式化方法与 VHDL 环境结合得比较成功的范例是正在研究的 VHDL/S 规范语言^[9],由于形式化方法的应用要求有较强的数学背景,因而如何利用其优点并使其具有简便易用的界面是一个挑战。作者对文[9]所提出的模型建立方法进行了

(下转第 89 页)