

基于AADL的中断控制设计方法*

李振松, 顾斌

(北京控制工程研究所, 北京 100190)

摘要: 介绍了结构分析与设计语言(AADL)的应用研究和优点,以某航天器控制分系统部分中断类型为例,针对AADL语言自身直接对中断问题描述能力上的不足,提出了两种基于AADL的中断控制设计方法,并进行了详细阐述和比较,为系统设计阶段对中断控制的抽象描述提供了一种思路和方法,从而有利于实现早期阶段对模型可调度性等方面验证的途径。

关键词: AADL; 中断; 模型; 可调度性

中图分类号: TP272

文献标识码: A

文章编号: 1674-7720(2011)10-0083-04

Designing method of interrupt controlling based on AADL

Li Zhensong, Gu Bin

(Beijing Institute of Control Engineering, Beijing 100190, China)

Abstract: The research of application and advantage of AADL(Architecture Analysis and Design Language) is introduced. Some types of interrupt of one spacecraft control system as an example, two designing methods for interrupt controlling based on AADL is present and depicted in detail to compare, in order to make up the disadvantage of the language in describing the interrupt, thereby providing one way to abstractly describe interrupts in the designing phase of system and to service the early validate of the model, such as schedulability.

Key words: AADL; interrupt; model; schedulability

结构分析与设计语言AADL(Architecture Analysis and Design Language)是2004年由美国汽车工程师协会(SAE)在MetaH UML等基础上提出的一种模型设计语言,最初为主要面向航空电子的设计分析语言,后经过不断完善和修正,发展成为一种面向嵌入式实时系统体系结构的分析设计语言,并发布为SAE AS5506标准,以期能够提供一种标准而又足够精确的方式,设计与分析嵌入式实时系统的软、硬件体系结构及功能与非功能属性,采用单一模型支持多种分析的方式,将系统的设计、分析、验证、自动代码生成等关键环节融合于统一框架之下^[1]。AADL的语法简单、功能强大、可扩展的优点使之具有非常广阔的应用前景,特别适合于工业控制、汽车、航空航天等任务关键和实时领域。

中断是嵌入式程序设计中的难点。中断是指CPU暂停执行当前程序,转而执行更紧急的程序,并能在执行结束后自动恢复执行原程序的过程。中断的执行涉及硬

件、软件及软硬件交互的多方面因素。

考虑到中断在嵌入式系统设计中的重要性及AADL语言本身在对中断过程描述上的不足,本文以某航天器控制系统部分中断类型为例,在设计层面对中断过程采取不同的抽象描述,并进行比较,从而为中断过程的设计提供参考。

1 AADL与中断

1.1 AADL概述

AADL标准由文本和图形语言、XML/XMI交换格式、错误模型附件等一批相应附件组成,其中AADL语言是AADL标准的核心^[2,3]。AADL语言和UML一样基于面向对象的思想,组件类型和组件实现分别与面向对象程序设计中的类和对象的概念相似,组件实现是对客观世界中嵌入式系统的组件进行抽象,组件类型则是对组件实现的抽象和归纳^[4]。类型声明定义了组件的接口元素和外部观察属性(接口特征、流说明及内部属

* 基金项目: 国家自然科学基金重大研究计划项目(90818024)

技术与方法 Technique and Method

性值等);实现声明定义了组件内部结构:子组件及其连接,子程序调用顺序、模式,流实现和属性;组件实现可以生成实例。

AADL 中主要组件分为软件组件、执行平台和系统组件类。其中软件组件主要包括数据、子程序、线程、线程组和进程;执行平台组件包括处理器、存储器、总线和设备;系统组件类做为一个软件组件和执行平台的组合体的抽象,由系统组件来表示。AADL 的组件之间的交互通过定义的接口与连接进行数据交换与传输。同时 AADL 标准还为系统的控制流和数据流的相关机制提供了运行时语义,如消息传递、事件传递、线程调度协议、时间要求等,从而实现对部分非功能属性的分析。通过对模型模式及模式转移条件的设计,AADL 还可以描述实时系统架构的动态行为。另外,AADL 可以通过新属性集定义及附件对 AADL 描述能力进行扩展。

1.2 中断

中断是现代计算机普遍采用的一项技术,嵌入式系统也不例外。一般来说中断的采用主要是用于故障处理、进行实时处理或者实现主机与 I/O 设备的并行工作等。引起中断的原因或者能够发生中断请求的设备成为中断源。中断源可以是软件方面造成的,也可以是硬件方面的原因。中断的复杂性体现在中断源的多样性上。实现中断的硬件和软件所组成的系统称为中断系统^[5]。当多个中断源同时向 CPU 发出中断请求时,CPU 根据各中断源的不同中断优先级别将其排队,优先处理中断优先权高的。中断发生后,一般的处理流程^[5]如图 1 所示。

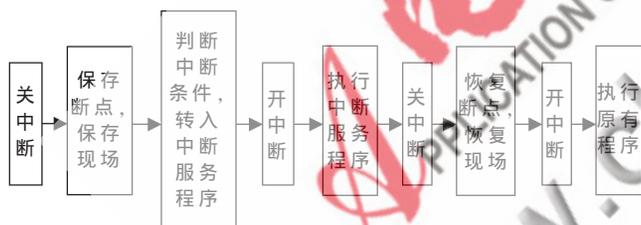


图 1 中断的一般处理流程

在处理某一个中断的过程中又发生了新的中断,并且新的中断类型的优先权更高,这时 CPU 中断正在执行的中断服务程序,转而执行新的中断处理,即发生了中断嵌套。中断嵌套的发生一方面给系统的运行带来了灵活性,同时也增加了系统设计的难度和运行的不确定性。

2 基于 AADL 的中断设计

AADL 语言本身并没有直接来描述中断的元素和方法,为了能够描述中断过程以方便进行早期可调度性等方面的验证,以简化的航天器控制系统的中断设计为例,采用 AADL 语言的不同方式进行中断过程的描述,增强 AADL 的设计范式。

2.1 模式转换设计

中断的发生可以看做是从中断源来的事件,在

AADL 中,通过事件端口的触发导致组件内部模式的转换来描述中断发生的动态过程,即在不同的模式下动态地配置执行组件。将控制系统正常运行时设计为 Normal 模式,在 Normal 模式下发生星载数据处理(OBDH)数据接收中断和发动机中断情况分别设计为 IntOBDHdeal 模式和 IntShutJetdeal 模式,在这三种模式下组件内分别执行相应的线程表征的任务,即周期任务线程(CycTask,在线程组 AppTask_group 内)、OBDH 数据接收处理线程(IntOBDH)和发动机计数器中断处理线程(IntShutJet)。由于可能存在中断嵌套,故增加了模式 OBDHdeal_Int-ShutJet、IntShutJet_OBDHdeal 来分别表征在执行 OBDH 数据接收中断的情况下发生发动机计数器中断事件时的运行模式及在执行发动机计数器中断程序的情况下发生 OBDH 的数据接收中断事件时的运行模式。

不同的工作模式表征了组件内部不同的子组件配置,在 Normal 模式下仅有线程组 AppTask_group 处于活跃态,IntShutJetdeal 模式下线程组 AppTask_group 和相应的中断处理线程 IntShutJet 处于活跃态,但是线程组 AppTask_group 由运行状态转移到了暂停状态,IntShutJet 处于运行状态。当 IntShutJetdeal 模式下发生 OBDH 数据接收处理中断事件时,即发生了中断嵌套,此时组件进入到 IntShutJet_OBDHdeal 工作模式,线程组 AppTask_group、暂停状态的中断处理线程 IntShutJet 及运行状态的中断处理线程 IntOBDH 均处于活跃态。与 IntOBDHdeal 模式不同的是,在 IntOBDHdeal 模式下中断处理线程 IntShutJet 是处于非活跃态,即未配置该线程组件。模式转换设计下的中断过程如图 2 所示。

如图 2 所示,通过事件端口的触发实现模式之间的转换,以虚线表示触发过程,在 Normal 模式下的执行组件为一线程组 AppTask_group,其运行中的状态转移过程如其内部的转移图所示。当运行中发生高级中断时,由运行状态 runing 转移到暂停状态 Pause_run,当中断返回后再由暂停状态 Pause_run 转移至运行状态 runing 继续运行。OBDH 数据接收处理线程(IntOBDH)和发动机计数器中断处理线程(IntShutJet)的内部状态迁移过程基本相同,在此不再赘述。

该设计方法抽象过程简单明了,但是由于在中断发生多级嵌套时所设计的模式状态就会成几何倍数增长,极易造成状态空间的爆炸,不利于系统的设计描述,更不利于后续的验证工作。

2.2 中断控制层设计

为了更加灵活地描述中断设计过程,可以增加一个中断控制监控层的方法来处理中断的发生。中断控制线程设计模式如图 3 所示。

通过设计中断控制线程来实现对内部中断和外部中断的统一管理,包括中断的优先级及对应的类型编码、中断的发生和处理过程。中断控制线程通过中断事件的触发来动态执行,有中断响应(Int_respond)、中断处

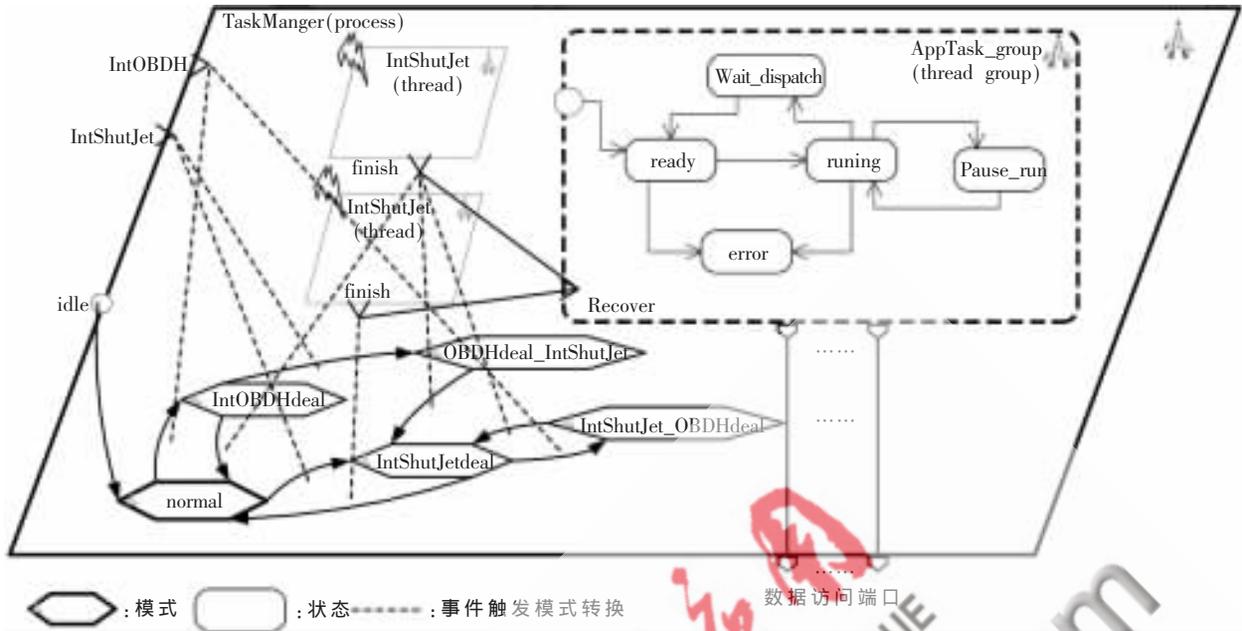


图2 模式转换设计的中断描述



图3 中断控制线程设计模式

理(Int_deal)及中断返回(Int_return)三个并行状态模式,其中中断响应状态包含了关中断、保存断点、保存现场、判断中断条件(包括有效性、类型及优先级等)、向正在执行线程发送暂停事件 Pause 及向中断处理线程发送触发事件 Run、开中断等这一系列过程;中断处理状态则是等待中断运行程序的返回事件 finish;中断返回状态下则是通过判断中断向量表数据(Int_Vector)来决定是否存在中断嵌套的发生,从而发送触发事件到相应的线程。如没有发生中断嵌套,即中断向量表队列除周期任务外为空,则发送事件 Recover 重新执行原周期性任务线程

组 AppTask_group。

中断向量表中的元素含有暂停执行线程的断点信息及所有待执行的中断处理程序的入口地址。通过对中断向量表的操作,可以将中断触发事件记录并动态处理,如果当前正在执行任务时发生一个或者多个高优先级中断时,从中断向量表中取出队首(非屏蔽中断),向执行任务发送暂停事件,该任务由执行态 running 转移到暂停态 Pause_run,同时将断点位置等必要信息通过事件数据端口传送给中断控制线程记录在中断向量表中。进而中断控制线程发送相应的触发事件 Run 到中

技术与方法 Technique and Method

断处理线程,从而执行中断处理过程。当发生中断嵌套时处理过程类似。待中断处理线程执行完毕后,发送 finish 事件给中断控制线程,将该中断从中断向量表中删除。再处理中断向量表中的队首中断类型事件,如果队首为一个被打断的处理线程,则通过向量表中的断点信息及发送 Recover 事件重新执行暂停状态下的处理线程。

中断控制线程在中断响应和中断返回的过程中都有一个关中断、开中断的过程,即在此过程中不响应除了电源掉电等不可屏蔽的中断外的任何其他类型的中断。为实现此过程,可以将中断向量表定义为一个互斥类型数据,来处理此过程中到来的中断事件。

通过将中断控制线程及其内部的子程序采用行为附件的状态机来描述,可以方便地将状态迁移过程以及线程的执行时间、截止时间等信息转换为 UPPAAL 下的时间自动机模板^[6],线程间的交互通过通道关联起来,用形式化的方法来实现在设计早期阶段验证某些性质,比如线程的可调度性验证。

通过对中断过程进行基于 AADL 的设计,可以看出 AADL 语言本身虽然没有直接用来描述中断过程的元素,但是通过对设计方法的改进,AADL 还是可以对其进行抽象设计描述的。通过模式转换设计与中断控制层设计方法的描述,可以看出在设计过程中可以很方便地将组件的动态行为转化为模式状态间的迁移过程,从而为后续采用有限状态机等形式化验证方法进行可调度性方面的验证工作提供基础。通过比较,不难看出模式转换设计方法虽然简单易理解,但是随着中断类型及中断嵌套层数的增加,模式状态的数量呈

几何倍数增加,描述能力很有限。而通过增加中断控制层的方法,设计中断向量表来记录中断事件的发生,则可以规避状态空间的爆炸问题,更加适用于描述实际的系统。

参考文献

- [1] 杨志斌,皮磊,胡凯,等.复杂嵌入式实时系统体系结构设计与分析语言:AADL[J].软件学报,2010,21(5):899-915.
- [2] SAE Aerospace.SAE AS5506A Standard[S/OL].[2004-11].(2009-01).http://www.sae.org/technical/standards/AS5506A.
- [3] PETER H F,DAVID P G,JOHN J H.The Architecture Analysis&Design Language(AADL):An Introduction.Technical Note[EB/OL].[2006-01-01].http://www.Sei.cmu.edu/pub/documents/06.reports/pdf/06tn011.pdf.
- [4] 王翰博,周兴社,董云卫,等.结构分析和设计语言 AADL 研究[J].计算机工程与应用,2009,45(16):1-4.
- [5] 赵晓玲,朱丽敏.计算机组成原理与汇编语言程序设计[M].北京:机械工业出版社,2003.
- [6] 刘倩,桂盛霖,李允,等.基于 UPPAAL 的 AADL 模型可调度性验证[J].计算机应用,2009,29(7):1820-1824.

(收稿日期:2010-12-21)

作者简介:

李振松,男,1983年生,硕士研究生,主要研究方向:嵌入式软件体系结构设计。

顾斌,男,1968年生,研究员,主要研究方向:软件工程、嵌入式软件设计。