AADL在中断控制系统建模中的应用

任 飞[1] 谯婷婷[2] 刘俊波[3] 邵杨锋[3]
(电子科技大学计算机科学与工程学院 成都 610054)[1]
(中航工业西安飞行自动控制研究所 西安 710065)[2] (西安电子科技大学计算机学院 西安 710071)[3]

摘 要 随着中断控制在嵌入式实时系统中的不断广泛应用,中断控制的可靠性是系统设计中面临的重要问题。虽然基于体系结构分析与设计语言(AADL)的形式化方法为这一问题的解决提供了思路,但AADL自身缺少有效的元素和方法来描述和建模中断。为此,提出一种结合AADL与中断控制器的中断控制系统设计方法,并运用GSPN可靠性计算模型对可靠性进行分析,从而为AADL在航电系统中的应用提供了思路。

关键词 AADL,中断控制系统,中断控制器,GSPN
中图法分类号 TP311 **文献标识码** A

Application of AADL in Modeling Interrupt Control System

REN Fei[1] QIAO Ting-ting[2] LIU Jun-bo[3] SHAO Yang-feng[3]
(School of Computer Science & Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China)[1]
(AVIC, Xi'an Flight Automation Control Research Institution, Xi'an 710065, China)[2]
(School of Computer Science and Technology, Xidian University, Xi'an 710071, China)[3]

Abstract With the wide applications of interrupt control in embedded real-time systems, the reliability of interrupt control system is a key problem during the procedure of system design. Although the architecture analysis and design language (AADL)-based formal methods can handle this problem effectively, it fails to provide effective elements and methods for describing and modeling interrupt. For this purpose, this paper proposed a design method of interrupt control system by combining the merits of AADL and interrupt controller, and utilized reliability computation model based on GSPN to analyze the reliability of the designed system, leading to a way for the application of AADL in avionics system.

Keywords AADL, Interrupt control system, Interrupt controller, GSPN

1 引言

如今,嵌入式实时系统已经广泛应用于航空航天、医疗设备、汽车控制等领域[1],这些实时系统对性能以及可靠性等方面都有较高要求,特别是航天航空系统更为如此。这就要求处理器能在规定的时间内完成数据的处理,从而直接考验了处理器的工作效率。中断作为提高计算机工作效率、增强计算机功能的一项重要技术,在当前的嵌入式系统中应用十分广泛,无论是在单片机还是各种测控系统都能发现中断的应用。然而,中断控制系统的设计是系统设计所面临的难点问题之一,如何实现中断处理程序的有效正确验证直接影响了处理器性能的好坏。特别是随着系统规模日益扩大、功能日益复杂,如何保证中断控制系统的可靠性已成为系统设计中的一个重要课题。在以航天航空系统为代表的安全关键领域,在系统设计中对可靠性的考虑稍有不甚,就会导致系统不能正确运行,从而浪费大量的人力物力去测试与修复问题系统。虽然现有的测试软件与方法能在一定程度上查找出系统执行上的问题,但对系统整体可靠性的保证与逻辑验证却力不从心。因此,在系统设计与开发早期,迫切需要一种严谨的方法来描述与验证系统。

针对这一问题,形式化方法应运而生,它以数学为基础,能够完整地、无二义地描述系统性质。近年来,AADL作为一种体系结构设计与分析语言(Architecture Analysis and Design Language, AADL),它的形式化描述特性能够发现系统在设计阶段因为不一致性、二义性和不完整性所导致的系统错误和缺陷,从而及时更正这些错误和缺陷,避免在系统实现后增加不必要的检测与维护费用,甚至避免运行中出现不可挽回的损失。AADL适用于高安全实时嵌入式系统,它用组件描述系统模型,并且能够实现对系统可靠性及任务可调度性等性能的分析[2,3]。刘倩等人[4]提出了利用模型检测工具UPPAAL对其线程组件在非抢占型调度策略下的可调度性进行形式化分析和验证,并实现了从AADL模型到UPPAAL中模型的模型转换工具。汤小明等人[5]运用AADL实现了飞行管理系统的可调度性、可靠性以及通信延迟等分

到稿日期:2012-07-20 返修日期:2012-11-05 本文受航空科学基金资助项目(20100718004)资助。
任 飞(1968—),男,高级工程师,主要研究方向为安全、智能多媒体终端可信软件平台、云计算安全、分布式内容网等,E-mail:renfei777@gmail.com;谯婷婷(1980—),女,硕士,高级工程师,主要研究方向为嵌入式操作系统。

析。针对武器控制系统的建模和性能验证问题，文献[6]建立了系统控制导弹发射流程的 AADL 模型。文献[7]则利用马尔可夫链提出了一种基于模型测试的 AADL 架构验证方法，将 AADL 错误模型和 AADL 结构模型转换为扩展的马尔可夫链模型，实现了对 AADL 模型的测试。贾璐等人[8]提出了一种航空电子系统安全性分析的设计和实现方案，为航空电子系统安全性的设计提供了验证方法。鉴于 AADL 在嵌入式软件建模中的良好应用前景，针对软件安全面临的安全等级和流延迟两类问题，文献[9]提出了一种基于 AADL 的软件可靠性验证方法。虽然 AADL 在航电系统等安全关键领域已经发挥出了重要作用，但 AADL 语言本身并没有直接用来描述中断的元素和方法。文献[10]中提出了一种利用中断控制线程处理中断的方法，该方法利用事件来模拟中断，将所有的中断提交给中断控制线程，由中断线程完成断点信息的保存和优先级的判定等。然而，该方法通过 CPU 执行中断控制线程来达到中断控制的效果，难免会影响 CPU 的利用率。鉴于上述不足，本文将 AADL 与中断控制器相结合，提出一种中断控制系统建模方法，通过硬件实现对中断的处理，将 CPU 从中断控制中解放出来，从而使其能够在中断控制器处理中断请求的同时并行地执行正常任务，提高 CPU 的利用率。

本文第 2 节简要介绍本文涉及的基本概念，包括 AADL、中断及中断控制器等；第 3 节从中断控制系统、中断处理过程与中断处理流程的测试 3 方面详细讨论了中断控制系统的设计；第 4 节基于广义随机 Petri 网，对中断控制系统的可靠性进行建模；最后总结全文。

2 基本概念

2.1 AADL

2004 年，美国自动化工程师协会(Society of Automotive Engineers，SAE)发布了 AS5506 标准并定义了 AADL 核心，目的是提供一种标准且有效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性质，通过采用单一模型支持多种分析的方式，包括可用性和可靠性分析、数据质量分析、实时性能分析、资源消费分析、安全性分析，将系统设计、分析、测试、自动代码生成等关键环节融合于统一框架之下。AADL 语法简单、功能强大并且可扩展，拥有广阔的应用前景，特别适用于航天航空、汽车控制等实时性要求较高的领域。AADL 的元素及关系如图 1 所示。

AADL 通过构件、连接等概念描述系统的软、硬件体系结构，通过特征、属性描述系统功能与非功能性质，通过模式变换描述运行时体系结构演化[11]。AADL 构件被定义为两部分：类型与实现。构件类型描述对外的功能接口(输入输出端口等)，构件实现则描述构件的内部结构(子构件、连接等)。AADL 定义了 3 类构件：软件构件、执行平台构件以及系统构件。软件构件用于软件体系结构建模，包括数据(data)、线程(thread)、线程组(thread group)、进程(process)、子程序(subprogram)构件；执行平台构件用于硬件体系结构建模，包括处理器(processor)、存储器(memory)、总线(bus)、外设(device)构件；系统构件组合所有的构件，层次化地建立系统的体系结构。组件交互使用端口(数据和事件输入输出端口)、端口组、子组件(subcomponent)访问、子程序调用、数据交换和共享，定义了功能接口和组件之间的通信。

2.2 中断与中断控制器

中断是 CPU 在执行一个任务时，对系统发生的某个紧急事件所做出的响应。应用到航天测控系统中时，这个紧急事件可以是外部测量设备发的，也可以是测控系统自身因为紧急状况而发出的。此时，系统就不得不中断当前任务去处理紧急件。当测量的数据正在被 CPU 处理时，应急设备由于紧急状况停止数据处理而占用 CPU，这种情况称为中断嵌套。中断嵌套的发生与中断的优先级密切相关，高优先级中断会打断低优先级中断。本文以航天测控系统为例，提出一种结合 AADL 与硬件的中断控制系统设计方法。中断控制系统中的关键部分是中断控制器，中断控制器可以实现对外部中断优先级的判断，为 CPU 提供中断向量码。外部中断所对应的中断程序的入口地址存储在中断向量表中，CPU 根据中断向量码查询中断向量表即可得到中断处理程序的入口地址。中断控制器的设计以可编程中断控制器 8259A 为原型，有 8 个中断输入端用来接收外部中断的请求信号，外部中断源的请求锁存在中断请求寄存器 IRR(8 位)，中断屏蔽寄存器 IMR(8 位)用于屏蔽特定中断请求信号，内部服务寄存器 ISR(8 位)记录 CPU 正在服务的中断。这些寄存器通过总线与 CPU 进行通信。

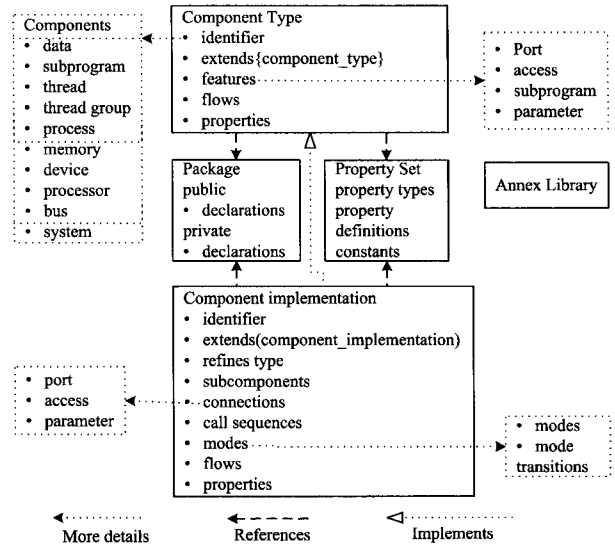

图 1 AADL 元素及关系

3 中断控制系统的设计

3.1 中断控制系统的实现

本文在 AADL 中引入中断控制器 8259A，实现了中断控制系统的建模与测试。中断控制器 8259A 接收设备传来的中断信号进行中断控制，向 CPU 传送中断请求和中断向量码，CPU 根据中断向量码调度相应的中断处理程序进行处理。系统的 AADL 图形表示如图 2 所示，其中 Data_Acquisition 和 Alertor 模拟航天测控系统中的数据采集设备和预警设备。数据采集设备在搜集外部环境变量值后会通过总线(Bus)将其交由 CPU 处理，此时需要一个中断信号(连接 IR7 端口)告知 CPU 来接收和处理这些数据；预警设备是在外部环境变量超出飞行器承受范围时产生的中断信号(连接 IR0 端口)，该信号告知系统需要做出紧急处理来改变飞行姿态。

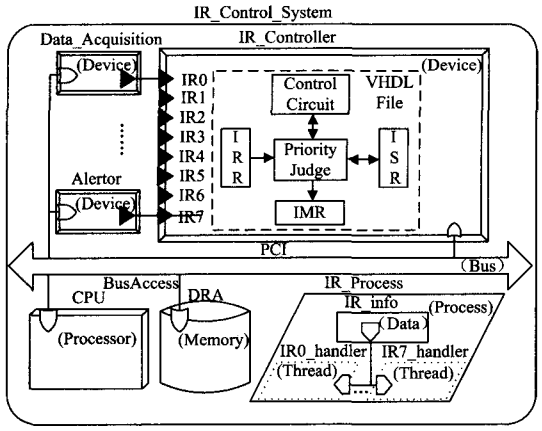

图 2　中断控制系统的 AADL 图形表示

基于上述对设备的描述，本文利用 AADL 来建模中断控制系统。AADL 提供了一系列的组件对系统建模，其中中断控制器采用 AADL 中的 Device 组件建模，将 Device 组件的 Source_Text 属性和超高速集成电路硬件描述语言（Very-High-Speed Integrated Circuit Hardware Description Language，VHDL）文件进行映射，VHDL 文件用来描述中断控制器的内部逻辑。利用 Processor 和 Memory 组件分别对处理器和内存建模，将中断处理程序 IR_Handler 用线程建模，中断请求信号 IR0 对应 IR_Handler0。在 AADL 中，线程不能独立存在，如图 1 所示，中断处理程序都位于 Process 组件中。Process 组件提供受保护的地址空间，将 Process 绑定到 Memory 上，Processor、Memory 和 Device 通过请求总线接入通信。AADL 使用具有正确语法和语义声明的规范进行建模，这些规范可以表示 AADL 文本、可扩展标记语言（Extensible Markup Language）或者 AADL 图形，这 3 种表示法是等效的。图 3 即为中断控制系统的部分 AADL 文本表示。

```
processor CPU
  features
    BusAccess: requires bus access
PCI;
end CPU
memory DRAM
  features
    BusAccess: requires bus access
PCI;
end DRAM;
device IR_Controller
  features
    BusAccess: requires bus access
PCI;
    IR0: in data port;
    IR1: in data port;
    IR2: in data port;
    IR3: in data port;
    IR4: in data port;
    IR5: in data port;
    IR6: in data port;
    IR7: in data port;
end IR_Controller;
process implementation IR_Process
  subcomponents
    IR_Handler0: thread IR_Thread0.Impl;
IR_Handler7: thread IR_Thread7.Impl;
  IR_Info: data DataType1.DataImpl1;
  connections
DataAccessConnection1: data access IR_Info.i -> IR_Handler0.i;
DataAccessConnection2: data access IR_Info.i -> IR_Handler7.i;
end IR_Process;
system implementation IR_System
  subcomponents
    PCI: bus BusType1.imp;
    CPU: processor IR_Processor.ProcessorImpl1;
    DRAM: memory IR_Memory.MemoryImpl1;
    IR_Process: process bIR_Process.ProcessImpl1;
    IR_Controller: device DeviceType1.IR_controller;
  connections
    BusAccessConnection1: bus access PCI -> RAM;
  BusAccessConnection2: bus access PCI -> IR_Controller;
  BusAccessConnection2: bus access PCI -> CPU;
end IR_System;
```

图 3　中断控制系统的 AADL 文字表示

3.2　中断处理过程

如图 3 所示，在飞行器正常飞行过程中，IR_Controller 接收来自数据采集设备（Data_Acquisition）和预警设备（Alertor）产生的中断请求，进行判优后传递给 CPU，CPU 暂停当前任务去执行中断处理程序。中断判优分为两种情况：(1) 当多个中断同时到时，中断控制器会将 IRR 相应位置 1，通过取 IRR 的最低位实现多中断同时到达的判优；(2) 如果 CPU 正在处理数据采集设备所采集的数据，外界环境变量超出飞行器承受阈值，此时，中断控制器会接受预警中断信号并将 IRR 第 0 位置 1，之后再与 ISR 内容比较（ISR 当前第 7 位为 1）。若预警中断优先级高于数据采集中断，中断控制器会向 CPU 传递中断请求和预警中断处理程序的向量码，CPU 查询中断向量表，转至预警中断处理程序入口地址开始执行，这样就可以实现中断的嵌套。

3.3　中断处理流程的测试

基于上述中断控制系统的设计，本节提供一种中断处理流程测试方法。首先，将每个中断处理程序的初始调用时间点和结束时间点信息存储到一个 Data 组件中，如图 4 所示。在中断都得到处理后，通过分析 Data 组件中存储的信息，可以还原出中断处理程序的执行流程，再与预期的执行流程做比较，便可判断中断控制系统的处理逻辑是否正确。

下面以中断请求信号 IR6 为例讨论具体的实现过程。

a）执行中断处理程序 IR6_Handler 时，先执行 CLI 指令关中断，向 Data 控件输出一条信息：IR6_Handler_Begin:12:33:04（中断处理程序名＋调用时间），再执行 STI 指令打开中断；

b）继续执行中断处理程序；

c）在中断处理程序 IR6_Handler 执行中断结束指令 EOI 之前，关中断，向 Data 控件输出一条信息 IR6_Handler_End:12:34:27（中断处理程序名＋结束时间），然后中断返回；

d）最后，根据 Data 中的信息可以还原出中断处理流程，如图 5 所示，将其与预期的中断处理流程进行比较，从而实现中断测试。

```
IR6_Handler-begin:12:33:04
IR3_Handler-begin:12:33:12
IR0_Handler-begin:12:33:17
IR0_Handler-begin:12:33:21
IR3_Handler-begin:12:34:23
IR6_Handler-begin:12:34:27
```

图 4　Data 中的信息

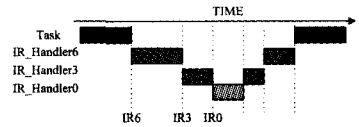

图 5　中断执行流程

4　中断控制系统的可靠性建模

对于中断控制系统而言，可靠性分析是十分重要的一个环节。目前，利用 AADL 错误模型虽然可以对 AADL 系统架构可靠性建模，但它只能表示为一个静态的系统模型。为了对本文设计的中断控制系统进行可靠性建模，利用文献[12]提出的基于 GSPN 的可靠性计算模型来实现。基于 AADL 的系统可靠性分析的流程具体为：首先根据系统需求建立 AADL 可靠性模型（AADL 架构模型＋错误模型）；然后根据转换规则将 AADL 可靠性模型转换到广义随机 Petri 网模型；随后利用文献[12]中的 GSPN 模型分析工具（PIPE2）对 GSPN 模型进行可靠性分析；最后根据可靠性分析结果，判断是否需要进行模型改造。图 6 及图 7 给出了图 2 中 CPU 的

（下转第 66 页）

[7] 贾小东,孙向辉,彭四伟. DHCP 协议缺点及其解决方案[J]. 计算机工程,2007,23:138-139

[8] Kerravala Z. As the value of enterprise networks escalates, so does the need for configuration management[R]. Boston: The Yankee Group,2004

[9] Chen Kai, Guo Chuan-xiong, Wu Hai-tao, et al. DAC: Generic and Automatic Address Configuration for Data Center Networks [C]//Proc of SIGCOMM 2010. NJ: ACM,2010:84-99

[10] Rodeheffer T, Thekkath C, Anderson D. Smart Bridge: A scalable bridge architecture[C]//Proc of SIGCOMM 2000. NJ: ACM 2000:201-211

[11] Myers A, Ng E, Zhang H. Rethinking the service model: scaling Ethernet to a million nodes[C]//Proc of Hot Nets 2004. NJ: ACM,2004:87-100

[12] Perlman R. Rbridges: Transparent routing[C]//Proc of Infocom 2004. NJ: IEEE,2004:105-118

[13] Kim C, Caesar M, Rexford J. Floodless in SEATTLE: a scalable Ethernet architecture for large enterprises[C]//Proc of SIGCOMM 2008. Vol 29, NJ: ACM, February 2011

[14] 解春欣,汪卫. 子图同构验证算法 OES [J]. 计算机工程,2011,3:74-78

[15] Darga P T, Sakallah K A, Markov I L. Faster Symmetry Discovery using Sparsity of Symmetries[C]//45st Design Automation Conference. 2008

(上接第 53 页)

错误模型及 GSPN 模型。

```
error model CPUEM
features
  Error_Free:initial error state;
  TempErr:error state;
  PermErr:error state;
  ErrND:error state;
  Failed:error state;
  Temp_Fault:error event {Occurrence⇒Poisson λ1};
  Perm_Fault:error event {Occurrence⇒Poisson λ2};
  Recover:error event {Occurrence⇒Poisson λ3};
  Repair:error event {Occurrence⇒Poisson λ4};
  Detect:error event {Occurrence⇒Poisson λ5};
  NonDetect:error event {Occurrence⇒Poisson λ6};
  PerceiveFail:error event {Occurrence⇒Poisson λ7};
  H_Err:out error propagation {Occurrence⇒fixed λ8};
  H_FailedVisible:out error propagation {Occurrence⇒fixed λ9};
  H_OK:out error propagation {Occurrence⇒fixed λ10};
end CPUEM;
error model implementation CPUEM.impl
transitions
  Error_Free-[Perm_Fault]→ PermErr;
  Error_Free-[Temp_Fault]→ TempErr;
  TempErr-[Recover]→ Error_Free;
  PermErr-[Detect]→ Failed;
  PermErr-[NonDetect]→ ErrND;
  ErrND-[PerceiveFail]→ Failed;
  Failed-[Repair]→ Error_Free;
  TempErr-[out H_Err]→ TempErr;
  Failed-[out H_FailedVisible]→ Failed;
  Error_Free-[out H_OK]→ Error_Free;
end CPUEM.impl;
```

图 6 CPU 的错误模型

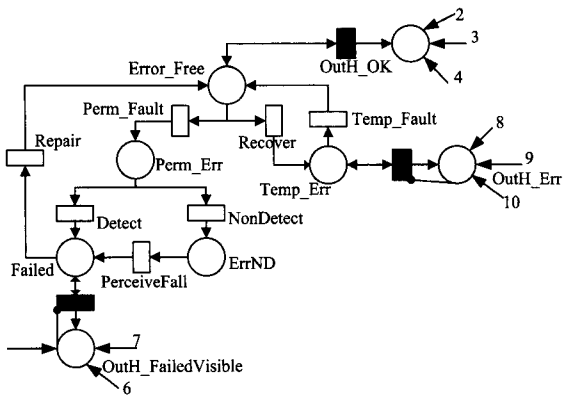

图 7 CPU 的 GSPN 模型

结束语 本文提出了一种基于 AADL 进行中断控制系统建模的方法,并利用基于 GSPN 的可靠性计算模型对可靠性进行分析。通过将中断控制器与 AADL 相结合,所提方法不仅弥补了 AADL 在中断控制建模中的不足,而且中断控制器的使用减少了 CPU 的负担,CPU 按照中断控制器提供的中断信号与中断向量码便可跳转至中断服务程序入口地址开始执行,从而避免了 CUP 参与中断优先级判决。

参 考 文 献

[1] Feiler P H, Gluch D P, Hudak J J. The Architecture Analysis & Design Language: An Introduction [R]. Carnegie Mellon University,2006

[2] SEI AADL Team. An extensible Open Source AADL Tool Environment(OSATE)[R]. SEI Carnegie Mellon University,2004

[3] Sokolsky O, Lee I, Clarke D. Schedulability analysis of AADL models[C]//Proc. 20th Int. Parallel and Distributed Proceedings Symposium,2006. USA: IEEE,2006:164-172

[4] 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

[5] 汤小明,苏罗辉,宋科璞. 飞行管理系统 AADL 建模与分析[J]. 计算机技术与发展,2010,20(3):191-194

[6] 许凌权,冯金富,左伟,等. 基于 AADL 的武器控制系统性能验证方法[J]. 电光与控制,2010,17(6):77-96

[7] 王庚,周兴社,张凡,等. AADL 模型的测试方法研究[J]. 计算机科学,2009,36(11):127-130

[8] 贾璐,胡林平,田丹. 基于 AADL 的航空电子系统安全性分析[J]. 航空计算技术,2009,39(5):58-61

[9] 谯婷婷,王乐,耶国栋. 基于 AADL 的软件可靠性验证[J]. 计算机应用,2012(s2)

[10] 李振松,顾斌. 基于 AADL 的中断控制设计方法[J]. 微型机与应用,2011,30(10):83-86

[11] 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

[12] 董云卫,王广仁,张凡,等. AADL 模型可靠性分析评估工具[J]. 软件学报,2011,22(6):1252-1266