软件学报 ISSN 1000-9825, CODEN RUXUEW
*Journal of Software*, 2022,33(8):2995−3014 [doi: 10.13328/j.cnki.jos.006610]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

# 软硬件综合 AADL 可靠性建模及分析方法*

陆 寅[1], 秦树东[1], 郭 鹏[2,3], 董云卫[1]

[1](西北工业大学 计算机学院, 陕西 西安 710072)

[2](南京航空航天大学 计算机科学与技术学院, 江苏 南京 211106)

[3](中国航空工业集团公司 西安航空计算技术研究所, 陕西 西安 710068)

通信作者: 董云卫, E-mail: yunweidong@nwpu.edu.cn

**摘 要**: 目前嵌入式系统广泛应用于航空电子、远程医疗、汽车电子等具有高可靠性要求的系统中. 随着嵌入式系统的复杂度越来越高, 为了保障系统的高可靠性需求, 需要在系统开发的早期设计阶段对系统的可靠性进行分析评估, 以提高系统的开发效率. 嵌入式系统中软硬件功能的失效都会对系统可靠性产生影响, 而 AADL 的可靠性模型缺乏对硬件构件错误的影响及传播机制进行刻画分析的能力. 综合考虑软硬件错误发生失效后对系统可靠性的影响, 提出了一种面向系统架构级别的软硬件综合可靠性分析方法. 该方法基于电子电路设计中事务级建模方法, 扩展了 AADL 事务级错误模型的语法和语义, 来支持 AADL 对硬件构件错误传播的硬件功能行为建模, 在此基础上, 利用 AADL 模型实例化机制实现对嵌入式系统可靠性建模, 刻画了错误行为在硬件构件之间、软硬件构件之间的传播与影响. 同时, 定义了 AADL 硬件构件事务级错误模型到广义随机 Petri 网模型的映射规则, 实现了系统软硬件综合的可靠性行为仿真计算模型组合, 支持嵌入式系统的软硬件综合可靠性分析. 开发了软硬件综合可靠性建模与分析工具原型, 并以某型飞机空气增压系统为例, 在航空电子系统架构设计中进行尝试, 验证了该方法在复杂嵌入式系统设计中进行软硬件综合可靠性分析的可行性与优越性.

**关键词**: 架构分析与设计语言; 复杂嵌入式系统; 事务级错误模型; 软硬件综合; 可靠性分析

**中图法分类号**: TP311

中文引用格式: 陆寅, 秦树东, 郭鹏, 董云卫. 软硬件综合 AADL 可靠性建模及分析方法. 软件学报, 2022, 33(8): 2995−3014. http://www.jos.org.cn/1000-9825/6610.htm

英文引用格式: Lu Y, Qin SD, Guo P, Dong YW. Hardware-software Integrated Reliability Modeling and Analysis Using AADL. Ruan Jian Xue Bao/Journal of Software, 2022, 33(8): 2995−3014 (in Chinese). http://www.jos.org.cn/1000-9825/6610.htm

## Hardware-software Integrated Reliability Modeling and Analysis Using AADL

LU Yin[1], QIN Shu-Dong[1], GUO Peng[2,3], DONGYun-Wei[1]

[1](School of Computer Science, Northwest Polytechnical University, Xi'an 710072, China)

[2](College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China)

[3](Xi'an Aeronautics Computing Technique Research Institute, AVIC, Xi'an 710068, China)

**Abstract**: The embedded system has been wildly applied in safety-critical system, such as aviation system, automobile systems, and telemedicine. However, reliability is not a property of these embedded systems that can be easily assured, for that the complexity of system architecture also increased rapidly. Thus, the reliability analysis and verification should be conducted in early design stages, so that to provide highly reliable and qualified systems while avoid economy and efficiency lose. In an embedded system, the system reliability is affected by both hardware errors, software defects, and hardware-software interactive failures. Although many achievements have been accomplished in the field of hardware-software integrated reliability analysis, they are not suitable to be applied in the early

* 基金项目: 国家自然科学基金(62192733)

本文由"形式化方法与应用"专题特约编辑陈立前副教授、孙猛教授推荐.

收稿时间: 2021-09-08; 修改时间: 2021-10-14; 采用时间: 2022-01-10; jos 在线出版时间: 2022-01-28

stages of system design and implementation. The SAE architecture analysis and design language (AADL) has provided an effective means of system architecture design and non-functional property verification, but it is not capable of hardware-software integrated reliability analysis for that its error model annex concentrates on software component error behavior modeling, and it cannot effectively describe the hardware error impact and propagation mechanism. An architecture level hardware-software integrated reliability modeling and analysis method, which considers the impact of both hardware, software and hardware-software interactive errors simultaneously, is proposed in this study. Combined with the transaction level modeling method in electronic circuit design, the proposed method extends the syntax and semantics of AADL in transaction level error behavior modeling to support the fine description of hardware component error and error propagation. Mapping rules from the enhanced AADL reliability model to generalized stochastic Petri net model are also proposed, so that the reliability model can be converted into calculation model to complete the hardware-software integrated reliability analysis and assessment of embedded system. A prototype IDE toolkit which implements the proposed method is developed to do testing and evaluation. It is used to do reliability modeling and analysis of avionic system, which is the control system of an air boost control system belongs to a certain type airplane. The result shows that, the proposed methods is capable of hardware-software integrated reliability modeling and analysis of complicated embedded system, and will provide refined analysis result compared with traditional AADL based methods.

**Key words**: AADL; complex embedded system; transaction level error model; hardware-software integrated; reliability analysis

近年来随着信息技术水平的不断提高, 安全攸关领域嵌入式系统的软件规模越来越大, 由此产生的可靠性隐患对系统安全性造成越来越严重的影响. 因此在这类系统的开发中, 可靠性指标已经成为系统评估的重要标准之一. 然而, 在嵌入式系统中软件子系统负责信息处理与决策判断, 硬件子系统负责信息采集与动作执行, 并为软件子系统提供运行平台, 两者之间耦合紧密的交联关系造成软硬件子系统之间故障影响的相互传播, 并演变出新的故障模式. 例如: 由硬件构件失效引起的软件运行错误. 在 Ravishankar 对 IBM 大型机操作系统失效行为的研究中, 发现有大约 35%的可观测故障来源于此类软硬件交联错误行为[1]. Roy[2]通过对电力系统中相位测量设备的可靠性进行分析与评估, 证明了软硬件交联故障会对系统可靠性分析与评价结果产生影响, 尤其是对安全关键系统而言这一影响不可忽视. 因此在进行嵌入式系统, 尤其是安全关键嵌入式系统的可靠性分析时, 需要综合考察软硬件子系统的故障模式, 以及软硬件子系统之间的故障传播与影响, 进行软硬件综合的可靠性分析与评价.

由于软硬件子系统遵循不同的故障机理, 如何进行软硬件综合的可靠性建模与分析一直是困扰研究者的一个难题. Immonen[3]和 Sinha[4]在其综述中分别总结和对比分析了这一领域自 20 世纪 90 年代以来的一些研究成果. 根据研究所遵循的技术路线, 这些方法可以分为结果综合与模型综合两大类. 前者将系统中的故障模式详细区分为软件失效、硬件故障和软硬件交联故障等几种模式, 通过可靠性测试分别建立各类故障模式的统计学模型, 然后使用状态机模型描述几类故障模式之间的转化迁移关系, 从而达到系统综合可靠性分析的目的, 如文献[2]. 这类方法对软硬件交联失效模式的描述与分析能力不足, 影响最终的评价结果. 后者则根据软硬件不同的失效机理分别为软硬件子系统建立可靠性评价模型, 然后应用系统可靠性框图等技术手段进行模型综合, 得到系统可靠性评价结果, 如文献[5−7]. 此类方法虽然能够兼顾软硬件交联失效对可靠性的影响, 但是两类不同机制可靠性模型的综合需要进行大量的条件设定, 仍然会影响最终的评价结果.

这两类技术路线都是在应用统计学方法建立不同故障模式可靠性模型的基础上进行综合分析, 因而存在如下的缺陷: 首先, 由于软硬件子系统故障机理的不同, 在对两者的可靠性模型或可靠性分析结果进行综合时, 不得不进行大量的条件假设以简化分析. 这些条件假设会为最终的模型引入缺陷和误差, 影响分析结果. 因此需要一种软硬件一体的建模方法, 应用一种相对统一的模式来描述软硬件故障模式及其在系统中的传播行为, 以降低分析误差; 其次, 从本质上而言, 这两条技术路线所应用的统计学建模方法, 在建模过程中需要通过对系统及其构件进行可靠性测试, 从而获得准确的模型参数. 因此更加适用于在系统投入运行后进行系统可靠性变化规律的分析、验证与评估.

文献[8,9]提出了两种适用于可编程逻辑控制电路(programmable logical controller, PLC)的软硬件一体可靠性建模与评估方法, 通过建立混成关联模型(hybrid relation model, HRM)[8]或运行时态可靠性模型(run-time reliability model, RRM)[9]将 PLC 程序转化为等效贝叶斯网络模型, 然后通过将软件程序所控制的逻辑单元及

其可靠性参数映射到 HRM/RRM 模型节点的方式完成软硬件一体的架构模型建模. 最终以贝叶斯网络分析算法为手段, 实现 PLC 整体可靠性的预计与评估. 该方法中 HRM 或 RRM 模型的构造是通过对 PLC 编程软件程序运行时逻辑的分析完成, 具有一定的应用局限性, 不适用于以通用处理器为核心的嵌入式计算系统; 而且文献[8,9]所述工作面向 PLC 软件程序施行, 不能在计算系统设计的早期阶段应用.

然而, 对于安全关键嵌入式系统的整个生命周期而言, 更加需要在系统设计与实现的早期阶段进行可靠性分析, 通过采取一种行之有效的技术手段建立系统中的故障与故障传播行为的准确模型, 从而能够找出影响系统可靠性的关键故障及其传播路径, 以指导系统设计方案的优化, 满足系统的可靠性技术指标要求.

Tumer 等人[10−12]将基于系统功能模型的失效状态识别与传播分析方法(functional-failure identification and propagation, FFIP)应用在嵌入式系统可靠性建模与分析中, 在系统设计的早期阶段进行故障传播行为分析以协助进行系统的可靠性设计. 因为在这一阶段还不能通过可靠性测试获取系统组成部分的可靠性参数, 所以 FFIP 方法的模型不宜应用于系统的可靠性评估. 同时, 尽管 FFIP 能够建立一种基于信息流与控制流的功能概念模型, 利用系统中的信息流与控制流刻画软硬件模块之间的交互关系, 从而较好地满足软硬件一体的可靠性建模需求, 但是由于系统的功能概念模型通常与系统实际架构差异较大, 所以应用 FFIP 进行的可靠性分析结果在用于指导系统设计优化时缺乏说服力.

与功能概念设计阶段进行的可靠性分析相比, 在系统架构设计阶段进行可靠性分析和系统设计的优化具有更高的可行性. 在这一阶段中建立的系统架构模型准确地反映了系统的组织架构、构件类型, 以及构件之间的相互关联关系. 依托系统架构模型, 能够以构件参数的形式引入并记录系统部组件的可靠性参数, 以及系统可靠性分配方案参数, 从而为基于模型的可靠性分析与评估打下基础. 美国汽车工程师协会 SAE 联合其他部门于 2004 年提出的 AS5506 架构分析与设计语言(architecture analysis and design language, AADL)既是一种面向嵌入式系统架构模型建模的技术规范. AADL 采用自顶向下的系统工程思想以及层次化建模的方式, 清晰准确地刻画系统内的层次结构. 同时, AADL 还是一个开放性架构的技术标准, 能够通过扩展附录子语言的形式满足不同需求的建模需求. 卡内基梅隆大学的研究报告[13]表明, AADL 适用于在实时嵌入式系统研发的早期阶段架构建模, 进而基于模型进行分析验证, 发现并消除架构设计中的隐患. 2006 年发布的 AADL 错误模型附录(error model annex, EMA)是面向系统可靠性建模与分析的一个附录子语言, 它支持嵌入式系统中的错误源、错误行为和错误传播建模. 通过模型复合将 AADL 的架构模型与错误模型相结合构成系统的可靠性模型, 为运用形式化方法或仿真技术对嵌入式系统进行可靠性分析与验证提供了一种可能. 但由于 AADL 模型不支持对系统功能的建模描述, 应用 AADL 建立的可靠性模型不支持直接应用 FFIP 方法进行可靠性分析. 董云卫等人[14]提出使用广义随机 Petri 网(generalized stochastic Petri net)作为形式化的数学分析模型进行可靠性分析的方法, 实现了基于 AADL 建立的可靠性模型进行嵌入式系统的可靠性分析与评估. 魏晓敏等人[15]针对 AADL 进行了危害附录子语言(hazard model annex, HM)扩展, 应用 AADL 及 HM 附录对嵌入式系统建立安全性模型, 进行系统安全性分析与评估, 通过扩展并建立 HM 形式化描述模型, 结合确定随机 Petri 网进行安全性分析、PSSA 安全性指标分配以及基于模型的开发过程早期安全性分析与验证. 肖明睿等人[16]面向 ARP4761 中共因分析与特定风险分析的需求, 对 AADL 进行了特定风险模型附录子语言(particular risk model annex, PRM)的扩展, 支持外部环境物理因素建模和人因模型建模, 结合 AADL 架构模型与 EM 和 HM, 构成嵌入式系统及其部署应用环境的综合型安全性模型, 支持基于场景的系统安全性分析与验证.

但是由于 AADL 是一门面向嵌入式系统架构建模, 尤其是软件子系统架构建模的语言规范, 在其构件的模型语义设计上存在硬件构件模型语义高度抽象的固有缺陷, 表现在系统可靠性模型中缺乏对硬件构件故障机理、故障传播、以及软硬件交互错误传播与转化行为的有效描述. 因此上述基于 AADL 的研究成果中普遍缺乏对软硬件交联失效的建模与分析, 无法满足架构设计阶段系统软硬件综合可靠性分析的需求. 针对这一问题, 本文提出通过在 AADL 硬件构件内部进行事务级行为建模的方式, 对构件内部错误原因与错误传播行为规律进行建模, 从而增强 AADL 硬件构件错误模型的建模能力, 以支持软硬件综合的可靠性分析. 本文工作的创新点包括以下 3 个方面: 首先, 提出以事务级模型的方式进行硬件构件内部功能模型建模、错误行为

和错误传播与转化规律建模; 其次, 建立了 AADL 的事务级错误模型附录子语言, 用于在事务级行为的尺度上建立错误模型, 描述硬件故障发生机制及其内部错误传播与转化规律, 满足软硬件综合可靠性建模的需求; 最后提出一套基于 AADL 事务级错误模型的可靠性分析算法, 进行软硬件综合的可靠性分析与评估.

本文第 1 节针对 AADL 在嵌入式系统硬件子系统方面建模能力不足的问题, 进行构件事务级错误模型附录扩展, 给出硬件构件的事务级模型、事务级错误模型的形式化语义, 以及 AADL 事务级错误模型(transaction level error model, TLEM)附录扩展方案. 第 2 节首先结合外场可更换模块(LRM)这一种典型的嵌入式计算系统阐述如何应用 TLEM 建立软硬件综合的可靠性模型, 其次通过制定事务级错误模型向 GSPN 网络模型的转换规则, 提出基于 AADL 的软硬件综合可靠性分析的计算方法. 第 3 节设计一种 AADL 软硬件综合可靠性建模与分析工具原型架构, 以某型飞行器空气增压系统的 LRM 模块为演示验证用例, 验证上述软硬件综合的嵌入式系统可靠性建模与分析算法有效性. 最后对本文的工作进行总结和展望.

## 1 AADL 事务级错误模型扩展

本文工作通过建立硬件构件的事务级错误模型, 在构件内部事务处理的层面上精细化描述错误传入与传出硬件构件的行为, 以及错误对构件内部事务单元(transaction module, TM)的映像和单元间错误的传播与转化的规律, 达到软硬件综合可靠性建模的目的. TLEM 以硬件构件事务级模型(transaction level model, TLM)为基础, 通过模型嵌套为其内部事务单元复合硬件构件事务单元错误模型(transaction module error model, TEM), 最后补充定义硬件构件内部 TM 间错误传播行为模型, 以及硬件构件向系统中相邻构件进行错误传播的行为模型. TLEM 模型详述如下.

### 1.1 计算机硬件事务级模型

架构分析与设计语言 AADL 的建模能力偏重于嵌入式软件架构与非功能属性建模, 对硬件构件内在执行逻辑建模能力不足, 不利于进行可靠性分析所需的错误机制与错误传播分析. 事务级模型是一种面向事务处理过程的硬件电路功能模型抽象建模方法. 硬件组件的事务级模型着眼于组件内部模块的功能分配、模块之间的交互协作关系, 且硬件功能单元运行具有事务性特征, 通过将硬件内不同的功能模块划分为事务单元, 有效刻画硬件内部信息交互过程, 并且此过程并不涉及硬件的电路逻辑和实现工艺, 在硬件开发周期中广泛用来对的早期设计阶段对硬件的功能行为进行建模, 指导硬件开发. 因此, 在系统可靠性设计中, TLM 模型也可以用来分析软件和硬件的功能综合影响, 实现对系统可靠性的分析与评估.

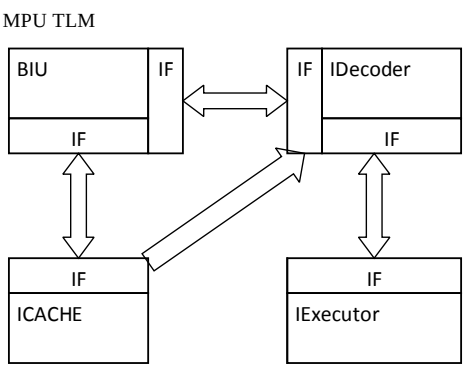



图 1 简化后的嵌入式处理器事务级模型

文献[17]中以 SystemC 为描述语言建立了嵌入式微处理器 MIPS R3000 事务级模型, 总共包含指令缓存(ICACHE)、指令取指(IFETCH)、指令解码(DECODE)、指令执行(EXEC)、数据缓存(DCACHE)以及浮点运算(FPU)这 6 个事务单元. 在本文中面向硬件构件事务级故障模式分析的需求对其进行简化, 建立通用嵌入式处理器事务级模型如图 1 所示.

处理器事务级模型由总线接口(BIU, 相当于 IFETCH 单元)、高速缓存(CatchRD, 相当于 ICACHE 与 DCACHE 单元)、指令解析(IDecoder)和指令执行(IExecutor)4 个单元构成. 单元间通过事务的请求与处理进行数据传递和同步触发, 通过相互协作完成程序的调入、执行、运行结果缓存等功能. 图中 IF(transaction interface)是 TM 的事务接口. TM 通过 IF 调用 TLM 仿真平台提供的通信服务, 进行相互间的事务请求与响应, 实现对信息的传递、变换和处理, 单元功能及其调用关系见表 1.

表 1 嵌入式处理器事务级简化模型组成

| 单元名称 | 编号 | 功能描述 | 单元间调用关系 |
|---|---|---|---|
| 总线接口 BIU | $tm_0$ | 产生寻址地址, 包括指令地址和数据地址 | CacheRD, IDecoder |
| 高速缓存 CacheRD | $tm_1$ | 缓存程序指令和数据; 完成指令地址寻址和指令读取 | CacheRD |
| 指令解析 IDecoder | $tm_2$ | 进行指令解析, 包括取操作数 | BIU, CacheRD, IExecutor |
| 指令执行 IExecutor | $tm_3$ | 执行程序指令 | IDecoder |

在 TLM 中, 硬件组件功能的一次执行可以视为一组相关的事务按照一定的次序依次执行的过程. 但在硬件功能运行的每一时刻, 只有一项事务被激活, 因此可以使用有限行为状态机来刻画 TLM, 并通过扩展硬件构件事务级模型附录的方式, 基于 AADL 对嵌入式系统硬件构件的事务级行为进行建模. 具体方法是, 将正在运行的事务视为硬件构件处于某行为状态; 将其前驱事务的提交、以及后继事务的发起视作硬件构件行为状态的迁移; 将事务提交的成功与否视作状态迁移的触发条件. 由此得到的刻画硬件功能运行方式的事务处理行为状态机模型, 见定义 1.

**定义 1**. 嵌入式系统硬件构件的事务级模型是一个四元组 $TLM=(TM,T,E,tm_0)$. 其中,

(1) $TM$ 是硬件构件中事务单元的集合, $TM=\{tm_0,tm_1,...,tm_0\}$, $tm_i$ 代表事务单元 $i$ 被激活, 硬件处在执行 $tm_i$ 事务功能的行为状态;

(2) $T: TM \to TM$, 表示硬件构件运行过程中内部事务单元激活状态在不同单元间的迁移关系;

(3) $E$ 是触发硬件构件事务单元间激活状态迁移发生的事件集合. 迁移的触发过程可以表示为事件的一个逻辑表达式 $t(e)$, 即 $\exists e \in E$, 是迁移 $t \in T$ 的使能条件. 当 $t(e)=\text{TRUE}$ 时, 使得 $tm_i \xrightarrow{t(e)} tm_j$ 发生, 其中, $tm_i$和$tm_j \in TM$, 并且$i \neq j$;

(4) $tm_0 \in TM$ 是硬件构件执行时首先激活的事务单元.

与电子电路设计与仿真中使用的事务级模型不同. 基于 AADL 及其扩展附录实现的硬件构件事务级模型要求事务单元所承担的事务具有唯一性. 因此在硬件电路设计中承担多个事务功能的事务单元. 在建模时需要拆分成多个单元进行建模. 基于定义 1 建立的处理器事务级模型实例见第 2.1 节中的图 6 所示.

### 1.2 嵌入式系统运行平台事务单元错误模型

硬件构件的事务级模型使用事务单元激活状态的迁移来刻画硬件功能的实现过程. 在这一过程中. 如果相关的事务单元的事务执行出现错误. 则会直接影响硬件构件整体的功能实现. 因此建立硬件构件精细化的错误模型首先需要建立硬件构件的事务级错误模型. 在构件内部事务级层面刻画错误出现的原因和错误的传播和转化行为与规律. 定义事务单元错误模型 TEM 如定义 2 所示.

**定义 2**. 硬件构件事务单元错误模型是一个四元组 $TEM=(ES,ET,EC,es_0)$. 其中,

(1) ES (transaction module error state)是事务单元执行时可能发生的错误状态集合. $ES=\{es_0,es_1,es_0,\ldots,es_m\}$. $es_i \in ES$ 表示事务单元在执行过程中可能处于的错误状态;

(2) ET (transaction module error state transition)是事务单元错误状态间的迁移关系 $ET: ES \to ES$;

(3) EC (transaction module error condition)是导致事务单元错误状态迁移发生的事件的集合. 迁移的触发过程可以表示为事件的一个逻辑表达式 $et(ec)$, 即 $\exists ec \in EC$. 是迁移 $et \in ET$ 发生的使能条件. 当 $et(ec)=$TURE 时. 使得 $es_i \xrightarrow{et(ec)} es_j$ 发生. 其中, $es_i$和$es_j \in ES$, 并且$i \neq j$;

(4) $es_0$ 是事务单元错误模型的初始状态, 代表着事务功能正常执行并提交.

事务单元的错误模型刻画了事务单元在错误事件 ec 的作用下在不同错误状态间迁移的行为规律, 按照错误状态的行为特征, 可以分为以下几种情况.

事务单元错误模型的初始态 $es_0$; 在此状态下事务单元功能正常执行并提交正确的执行结果, 不向后继单元输出错误.

第 1 类错误状态 $es_1$. 当事务单元激活时, 若输入的参数中包含有错误, 而单元又未能识别, 那么单元会按照正常情况启动事务处理, 从而导致事务单元从初始态 $es_0$ 变迁到错误状态 $es_1$. 此时, 若单元的事务处理能

够执行完毕, 则进行事务提交. 但提交的结果中包含错误, 即发生了错误传播, 导致错误向后继事务单元传递. 在图 2 所示事务单元错误模型中, 假设处理器的取指单元 $tm_0$ 输出的指令地址包含错误, 但并未超出合法指令边界地址. 所以指令缓存单元 $tm_1$ 无法识别该错误, 在初始态 $tm_1$ 的 $es_0$ 状态启动了事务处理. 从而进入第 1 类错误状态 $es_1$, 从错误的指令地址上获取了下一条指令, 然后通过事务提交传递给指令译码单元 $tm_2$, 造成来自 $tm_0$ 的错误通过 $tm_1$ 继续向 $tm_2$ 传播.

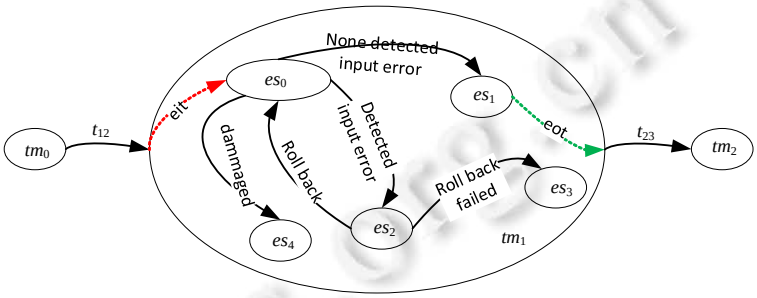



图 2 PowerPC 高速缓存事务单元错误模型示意图

第 2 类错误状态 $es_2$. 当事务单元激活时, 若事务单元能够检测到输入参数中包含错误, 则进入第 2 类错误状态 $es_2$, 并进行事务回滚, 回滚后若事务处理成功. 在初始态 $es_0$ 下进行事务提交, 不发生错误传播; 若回滚失败, 则进入第 3 类错误状态 $es_3$, 产生异常中断, 等待外界干预.

第 3 类错误状态 $es_3$. 若事务单元通过事务回滚也未能消除错误. 则进入第 3 类错误状态 $es_3$, 并产生异常中断事件, 等待高层干预解决单元出错的问题. 此时因为没有事务提交, 不会向后继事务单元输出错误. 仍以图 2 中 PowerPC 事务级模型中的指令缓存单元 $tm_1$ 为例. 若 $tm_1$ 发现指令地址越界, 则会要求事务回滚. 以重新获取指令地址; 若取得的地址仍然具有越界错误, 则进入 $es_3$ 并产生异常中断, 请求操作系统处理.

第 4 类错误状态 $es_4$. 事务单元事务执行过程中, 若因单元硬件损伤、输入参数异常导致单元停止运行, 不再接收事务请求, 也不进行事务提交, 则认为硬件构件已失去运行能力, 处于错误状态 $es_4$, 需通过系统冗余机制进行容错处理. 例如典型的三加一冗余机制下, 若一个 CPU 发生了硬件损伤停止运行, 系统会通过其他两个通道的对比识别这一错误, 隔离故障通道, 并启用备份通道.

### 1.3 嵌入式系统运行平台的硬件构件事务级错误模型

在硬件构件事务级模型 TLM 与事务单元错误模型 TEM 的基础上, 将事务单元 $tm$ 与其错误模型 TEM 进行嵌套复合, 并补充错误传播模型, 构成硬件构件的事务级错误模型.

**定义 3**. 基于事务的硬件构件错误模型是一个五元组 $TLEM=(HS,HT,F,in\_PROP,out\_PROP)$, 它由硬件构件事务级模型和事务单元错误模型嵌套组合而成的. 硬件构件事务级模型中的每一个状态, 都嵌套了一个事务单元的错误模型. 其中,

(1) $HS=TM\cup\left(\bigcup_{i=1}^{n}ES_i\right)$ 是基于事务的硬件构件错误模型的错误状态集合. 其中,

(a) $TM$ 是硬件构件中事务单元的集合. $TM=\{tm_0,tm_1,tm_2,...,tm_n\}$;

(b) $ES_i$ 是事务单元 $tm_i$ 执行时可能发生的错误状态集合;

(2) $HT$ 是硬件构件错误模型的错误状态之间的迁移关系集合. 它由 TLM 行为状态迁移关系、TEM 错误状态的迁移关系以及 TLM 行为状态与 TEM 错误状态间的迁移关系复合而成. 即 $HT=T\cup\left(\bigcup_{i=1}^{n}ET_i\right)\cup EIT\cup EOT$, 其中,

(a) $T$ 是硬件构件事务单元执行过程中不同事务行为状态之间的迁移关系. $T:TM\to TM$;

(b) $ET_i$ 是事务单元 $tm_i$ 的错误状态间的迁移关系. $ET_i:ES_i\to ES_i$, $ES_i$ 是 $tm_i$ 的错误状态集合;

(c) EIT (error input transition). TLM 行为状态向 TEM 行为状态的向下迁移, 每一个事务单元有且只有一个错误向内传播迁移 $eit_i$, 当事务单元 $tm_i$ 被激活时 $eit_i$ 立即发生. 由前导单元传播的错误随 $eit_i$ 进入 $tm_i$ 并导致

$tm_i$ 的错误状态发生变迁. 因此, $eit_i$ 必然指向 $tm_i$ 的初始错误状态;

(d) EOT (error output transition)表示 TEM 错误状态触发时. 错误向上传播到错误模型所在的事务单元的迁移的集合. $EOT=\left(\bigcup_{i=1}^{n}EOT_i\right)$, 其中, 子集 $EOT_i$ 的下标 $i$ 表示该子集中的迁移发生在 TLM 中的事务单元 $tm_i$ 内. $EOT_i$ 是事务单元 $tm_i$ 可能处于的所有错误状态下被触发时, 导致事务单元出错的迁移的集合, 即 $EOT_i$ 中的元素 $eot_{ij}$(假设事务单元 $tm_i$ 的错误模型有 $m$ 个错误状态, $0<j<m$). 表示事务单元 $tm_i$ 处在错误状态 $es_{ij}$ 时, 该错误状态能够通过迁移 $eot_{ij}$ 导致事务单元出错.

(3) $F$ 是 TLEM 的错误状态间迁移发生的事件集合. 变迁的触发过程可以表示成为事件的一个逻辑表达式 $ht(f)$, 即 $\exists f\in F$, 是迁移 $ht\in HT$ 的使能条件, 当 $ht(f)=\text{TRUE}$ 时, 使得 $hs_i \xrightarrow{ht(f)} hs_j$ 发生, 其中, $hs_i$ 和 $hs_j\in HS$, $i\neq j$, 且有: 当 $ht\in T$ 时. $ht(f)=ht(e), e\in E$; 当 $ht\in ET_i$ 时, $ht(f)=ht(ec), ec\in EC_i$; 当 $ht\in EIT$ 时, $ht(f)$恒为真. 即向下错误传输 $eit$ 发生时, 事务单元必从其内嵌的初始状态迁移到某一错误状态; 当 $ht\in EOT$ 时, $ht(f)=(hs_i\in ES_i$, 且$hs_i$不是事务单元内嵌错误模型的初始态);

(4) *in_PROP* 是硬件构件向内错误传播集合. 传播始于硬件构件具有向内传输性质的特征属性, 如向内传输端口、总线访问需求等; 传输到硬件构件事务级模型的初始事务单元 $tm_0$, $tm_0\in TM$;

(5) *out_PROP* 是硬件构件向外错误传播的集合. 传播始于硬件构件事务级模型的事务单元 $tm_i$, 传输到硬件构件具有向外传输性质的特征属性. 如向外传输端口、绑定关系等.

基于定义 3 建立的硬件构件事务级错误模型实例参见第 3.2 节中图 14 所示. 图中展示了 powerPC 处理器事务级错误模型, 包括内部 4 个事务单元及其内嵌的事务单元错误模型以及相关的错误传播模型.

### 1.4 AADL事务级错误模型附录扩展规则

本文工作采用采用附录扩展的方式为 AADL 语言扩展了硬件构件的事务级错误模型附录, 扩展内容包括硬件事务级错误模型附录库(TLEM Annex Library)和附录子句(TLEM Annex Subclause), 扩展附录子语言使用扩展巴克斯范式表述硬件构件事务级错误模型建模的文法规则, 并在 Eclipse 框架内实现了文法解析功能.

```
tlem_error_model_library::=annex
TLEM({**tem_library_constructs**})none);
tem_library_constructs..=[tlem_error_type_library]
tlem_error_type_library::=
  tlem error types
        [use types error_type_library_list];
        {tlem_error_type_definition}+
  end types;
tlem_error_type_definition::=defining_tlem_error_type_identifier: type;
```

图 3 事务级错误模型附录库的语法规则示例

硬件事务级错误模型附录库声明了事务级错误模型中特有的错误类型, 并且在定义时使用了 use types 关键字形成对 AADL EMA 中错误类型定义的引用, 因而与 EMA 定义的错误类型兼容, 保证了嵌入式系统整体错误模型组装过程中错误类型的适配性. 图 3 中展示了硬件事务级错误模型附录库的部分内容, 可见对 EMA 错误类型的引用声明, 以及使用扩展巴克斯范式定义了 TLEM 附录错误类型.

TLEM 通过其附录子句与 AADL 架构模型关联, 图 4 中展示了扩展定义的 TLEM 附录子句语义和语法. 在其中以扩展巴克斯范式给出了事务级错误模型、事务级模型状态机、硬件构件错误传入与错误传出行为的定义, 以及 TEM 与 TLM 的复合语法.

(1) 图 4 中区域Ⓐ中定义了硬件构件事务级错误模型的主体结构是 hcem error behavior, 这是对定义 3 TLEM 的具体实现. 其中包括了事务级模型状态机 tlm_state_machine 和硬件构件事务级错误模型向内、向外错误传播 in propagations, out propagations.

(2) 在区域Ⓑ中给出了 tlm_state_machine 的定义, 这是定义 1 TLM 的具体实现. 其中, TLM 的事务单元集合 *TM* 在实现中被定义为事务级状态机的状态集{tlm_state}, 而 tlm_state 使用了复合状态 **composite state** 的语义, 以便与 TEM 进行嵌套复合.

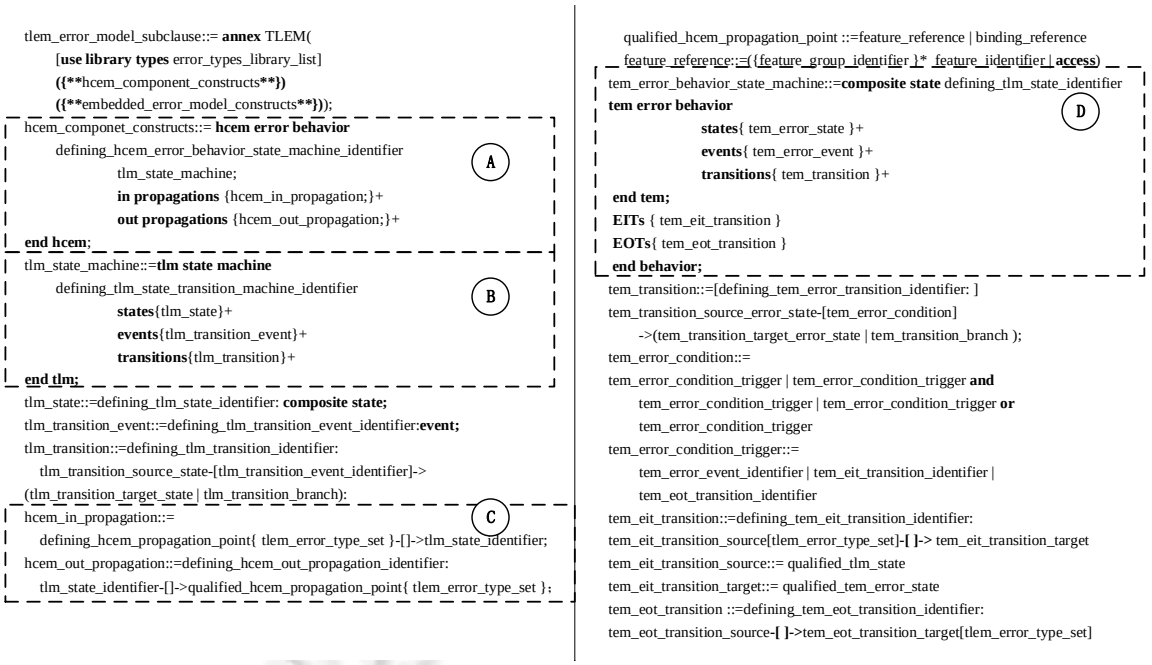



图 4 事务级错误模型子句的语法规则示例

(3) 在区域Ⓒ中包含了 in propagations, out propagations 的具体定义. 可以看到其中定义了事务级错误模型错误传播点 hcem_propagation_point 作为错误传入和传出 TLM 的起止点, 而 hcem_propagation_point 的定义则引用了 AADL EMA 中错误传播点的语义, 形成了语法和语义上的兼容. 因而 TLEM 能够通过 EMA 中定义的错误传播与相邻构件的 EMA 模型进行模型组装.

(4) 区域Ⓓ中的内容说明了 tlm_state 复合状态所包含的内容是事务单元错误行为状态机 tem_error_behavior_state_machine, 这是定义 2 TEM 的具体实现. 由于 TLM 和 TEM 在实现中都被定义为状态机模型, 因此事务单元间的错误传播能够以简单的状态迁移形式加以实现, 如区域Ⓓ中 EOT 与 EIT 的定义.

## 2 软硬件综合的 AADL 模型可靠性分析

AADL 硬件构件的事务级错误模型为建立准确详实的软硬件交互错误传播行为模型提供了技术支持, 结合 AADL 架构模型和 AADL 的错误模型共同构成目标系统软硬件综合的可靠性模型. 由于这一 AADL 语言描述的模型属于半形式化模型, 不支持进一步的行为分析和量化计算, 因此还需要通过模型转换将其转换到有限状态自动机模型, 再借助模型演算工具软件完成计算分析[12]. 本文提出面向 AADL 硬件构件事务级错误模型的可靠性分析算法, 实现了 TLEM 模型向 GSPN 计算模型的转换, 支持软硬件综合的可靠性分析.

### 2.1 目标系统软硬件综合可靠性建模

目标系统软硬件综合的可靠性建模包括架构模型建模、软硬件子系统的错误模型建模、安全关键硬件构件的事务级错误模型建模, 以及模型组装 4 个步骤, 最终得到如图 5 所示层次结构的可靠性模型. 可以看到, AADL 的 EMV2 附录支持软件子系统各个层次的错误模型建模, 但却缺乏硬件底层错误行为的精细化描述能力, 需要使用本文提出的 TLEM 附录进行建模补充.

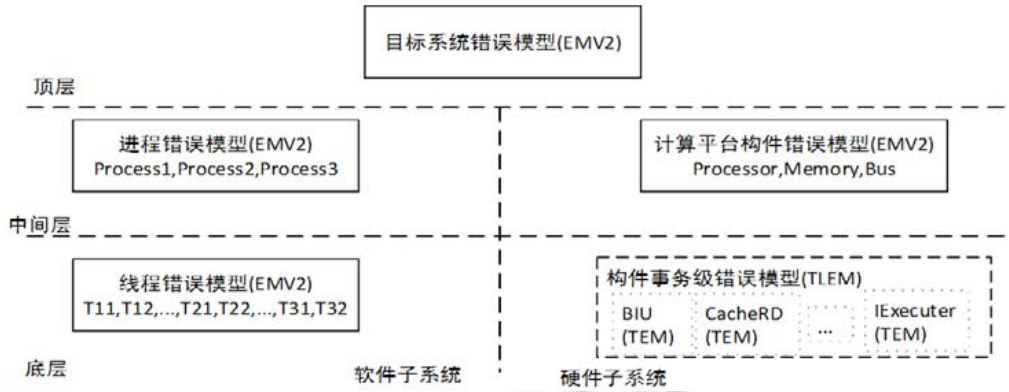



图 5 目标系统软硬件综合可靠性模型的层次结构

目标系统整体的可靠性模型建模步骤说明如下.

(1) 架构模型建模

使用 AADL 提供的预制构件遵循 AS5506 规范建立目标系统的架构模型, 通过 AADL 定义的各类连接关系描述构件之间、构件与子系统之间, 以及子系统与顶层系统之间的包含关系与交互行为.

(2) 软硬件子系统错误模型建模

使用 AADL 的 EMV2 附录子语言为架构模型中的构件建立错误行为自动机模型, 然后根据构件的接口定义及连接关系, 使用 EMV2 的错误传播语义描述构件之间的故障传播与影响行为. 在错误模型建模过程中, 需要注意根据错误传播关系找到系统中安全关键的硬件构件, 即位于软硬件子系统之间错误传播路径中、且完成了错误类型转换的构件. 此类构件需要使用 TLEM 附录进行精细化的事务级错误模型建模.

(3) 安全关键硬件构件的事务级错误模型建模

使用本文扩展的 AADL TLEM 附录子语言为安全关键的硬件构件进行精细化错误模型建模. 首先建立硬件构件的事务级模型 TLM, 其核心是构件内部的事务单元, 以及事务级行为状态机. 其中, 事务单元需要声明为复合状态类型, 以便通过内嵌复合事务单元错误模型 TEM; 其次, 为事务单元建立 TEM 模型, 并与 TM 状态进行嵌套复合; 最后根据事务模型中的事务迁移, 建立事务单元间错误迁移的 EOT、EIT 关系, 完成安全关键硬件构件的事务级错误模型 TLEM 建模. 以第 1.1 节中讨论的嵌入式处理器简化事务模型为例, AADL 事务级模型 TLM 及其文本描述如图 6 所示. 可以看到, BIU、CacheRD、IDecoder、IExecutor 事务单元都被声明为 AADL composite state 类型. 图 7 展示了其中 IExecutor 单元的事务单元错误模型 TEM.

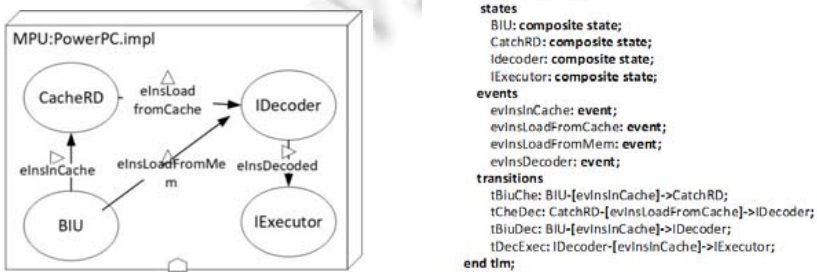



图 6 PowerPC 处理器简化事务级模型及其 AADL 文本描述

```
composite stator IExecutor
tem error behavior
  states
    esExecutor : initial state;   -- initial, error free state
    esExProp : final state;       -- error propagation state
  events
    evExFaultInstruction : event; --未知的错误
  transitions
    etExProp: esExecutor -[evExFaultinstruction]-> esExProp;
end tem;
  EIT:
    eitExFaultInst: Executor {errFaultInstruction} -[]-> Executor.esExecutor;
  EOT:
    eotExFaultInst: Executor.esExProp -[]-> Executor {errFaultData};
  end composite;
  **);
end GPM.impl;
```

图 7　IExecutor 事务单元错误模型 AADL 文本描述

(4) 模型组装和软硬件综合可靠性模型

最后, 为硬件构件的事务级错误模型 TLEM 补全 in_PROP 和 out_PROP 错误状态迁移, 从而与构件的错误状态建立联系, 描述硬件构件内部的事务单元错误构件整体错误状态产生的影响. 而硬件构件的错误状态又会继续通过目标系统 EM 模型中的错误传播路径向相邻构件(或子系统)进行传播, 至此即可完成硬件构件的事务级错误模型 TLEM 与目标系统 EM 错误模型的组装. 系统 EM 模型以附录模型的形式与系统的 AADL 架构模型进行组装, 三者共同共构成目标系统的软硬件综合可靠性模型.

在 AADL 的可靠性模型中, 错误分为瞬时错误和永久性错误. 瞬时错误会被模型中的错误接收器吸收, 不会在系统中传播, 永久性错误一经产生不会无缘无故的消失, 除非被模型中容错机制吸收, 或者被自动修复. 否则, 必然通过构件之间的联系扩散传播, 并对构件/系统的错误状态产生影响. 因此在建立可靠性模型的过程中必须对模型内各层级的错误进行传播路径的完备性检查, 这一检查过程的本质是对嵌入式系统内故障模式及其传播和影响分析过程, 而硬件构件的事务级错误模型刻画了软硬件错误经过硬件构件向后传播/发生转化的行为, 保障了分析过程错误传播路径的完整性和正确性.

### 2.2 软硬件综合可靠性模型向GSPN模型转换

由第 2.1 节建模方法的阐述可知, 嵌入式系统软硬件综合可靠性模型由目标系统的错误模型和安全关键硬件构件的事务级错误模型通过模型组装构成. 文献[12]讨论了系统错误模型向 GSPN 计算模型转换方法, 文献[13]中的定义 1–定义 4 讨论了这种模型转换方法的正确性. 在本文中提出硬件构件事务级错误模型向 GSPN 模型的转换方法, 以及 TLEM 模型与 EMA 模型错误传播连接向 GSPN 模型的转换方法. 从而实现了 AADL 软硬件综合可靠性模型向 GSPN 计算模型的完全转换. 转换方法详述如下.

已知广义随机 Petri 网模型可以表述为一个六元组, 见定义 4.

**定义 4**. 广义随机 Petri 网模型可以定义为一个六元组 $(P,T,A,W,M_0,\lambda)$, 其中,

(1) $P$ 是模型中所有位置的集合, 表示为 $P=\{p_1,p_2,p_3,...,p_n\}$;

(2) $T$ 是变迁集合, 表示为 $T=T_t\cup T_i$, 且 $T_t\cap T_i=\varnothing$. 其中, $T_t$ 为延时变迁集合 $T_t=\{t_1,t_2,...,t_k\}$, $T_i$ 为瞬时变迁集合 $T_i=\{t_{k+1},t_{k+2},...,t_n\}$;

(3) $A$ 是弧的集合, $A=A_{pt}\cup A_{tp}\cup A_i$, 其中, $A_{pt}\subseteq P\times T$ 是位置到变迁的弧的集合, $A_{tp}\subseteq T\times P$ 是变迁到位置的弧的集合, $A_i\subseteq P\times T$ 代表从位置到变迁的禁止弧的集合;

(4) $W$ 刻画了瞬时变迁中权重参数, $W:T_i\to N^+$, $N^+$限定为非零的自然数;

(5) $M_0:P\to N_0$, $N_0$是自然数, 为初始标识, 可以使用一维向量进行表示;

(6) $\lambda=\{\lambda_1,\lambda_2,\lambda_3,...,\lambda_k\}$ 是与延时变迁集相关联的变迁实施平均速率的集合.

由定义 3 和定义 4 可知, 为实现从 TLEM 模型到 GSPN 模型的转换, 可以按照从内到外的次序依次完成 3 组转换. 事务单元错误行为模型的转换、事务单元间错误传播行为的转换、事务单元到硬件构件错误迁移行为的转换. 然后通过添加禁止弧和吸收变迁, 来防止不正确的迁移行为以及吸收多余的令牌. 转换规则陈述如下.

令函数 $TLEM\_GSPN(TLEM)=\{P,T,A,W,M_0,\lambda\}$ 将硬件构件的错误模型转换为一个 GSPN 模型, 则有硬件构件错误模型 TEM 到 GSPN 计算模型的转换.

**定义 5**. 事务单元错误模型 $TEM=(ES,ET,EC,es_0)$ 到 GSPN 模型元素转换.

(1) $TLEM\_GSPN(es)=p_{es}$, 将 TEM 中事务单元错误状态转换到 GSPN 模型中的位置, 其中, 初始状态 $es_0$ 转换得到的包含一个令牌的初始位置 $p_{es_0}$;

(2) $TLEM\_GSPN(ec)=t_{ec}$, 将 TEM 中错误事件映射到 GSPN 模型中的变迁, 其中, 服从泊松分布的错误事件转换为 GSPN 中的延时变迁, 服从概率分布的错误事件转换为瞬时变迁;

(3) $TLEM\_GSPN(et)=A_{pt}\cup A_{tp}$, 将 TEM 中错误状态之间的变迁转换为 GSPN 模型中位置到变迁的弧, 以及变迁到位置的弧.

根据定义 3, TEM 间通过 *eot*、*t* 和 *eit* 完成事务单元间的错误传播. 因此在进行硬件构件事务级错误模型中 TLM 事务行为状态机向 GSPN 模型的转换时, 需要逐次完成 *eot*、*eit* 和 *t* 的转换.

首先进行事务单元向上错误传播依赖关系 *eot* 的转换. 由定义 3 可知, 事务单元向上错误传播关系不仅完成了某 *error_type* 类型的错误向事务单元外部传播, 同时还造成事务单元错误状态的变迁 $et_{eot}$. 因此在将 *eot* 转换到 GSPN 模型时, 转换规则如定义 6 所示, 转换结果构成如图 8(a)所示的子网结构.

**定义 6**. 事务单元向上错误传播行为 *eot* 的 GSPN 转换规则如下:

$$TLEM\_GSPN(eot)=\left(p_{tm\_type},t_{eot},a_{pt\_eot},a_{pto\_eot}\right),$$

$$TLEM\_GSPN\left(et_{eot}\right)=\left(p_{eot\_state},p_{eot\_state'},a_{pt\_eot},a_{tp\_eot}\right),$$

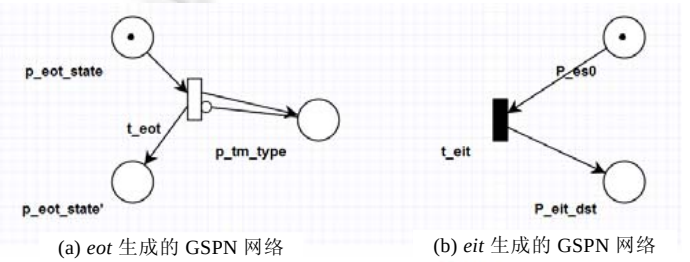
其中,(1) $p_{tm\_type}$ 是事务单元 *tm* 在 eot 作用下映射到 GSPN 网络中得到的位置, 简称 *tm* 的传播位置. 在图 8(a)中记作 *p_tm_type*, 表示在此状态下事务单元 *tm* 传出 *error_type* 类型错误, 事务单元的每一个 eot 转换产生一个传播位置 $p_{tr}$;

(2) $t_{eot}$ 表示 *eot* 行为的延时变迁. 在图 8(a)中记作 *t_eot*. *tm* 的每一个 *eot* 转换生成一个 $t_{eot}$;

(3) $a_{pt\_eot}$ 是从 $t_{eot}$ 指向 $p_{tm\_type}$ 的弧;

(4) $p_{eot\_state}$, $p_{eot\_state'}$ 分别表示变迁 $t_{eot}$ 的源位置和目的位置. 在图 8(a)中记作 *p_eot_state* 和 *p_eot_state'*, $p_{eot\_state}$, $p_{eot\_state'}\in P_{es}$ 是事务单元 *tm* 的错误状态转换得到的位置. 在一些特殊情况下, 事务单元 *tm* 在 *eot* 发生前后错误状态没有变化. 那么定义 6 中的 $p_{eot\_state}$ 与 $p_{eot\_state'}$ 为同一个位置. 反映在图 8(a)中表现为 *p_eot_state* 与 *p_eot_state'* 重合. 即 *p_eot_state* 在 *t_eot* 发生后重新获得了令牌. 因此图 8(a)所示的子网结构中还需要加上从 *t_eot* 到 *p_tm_type* 的禁止弧 $a_{pt\_eot}$;

(5) $a_{pto\_eot}$ 是从 $p_{tm\_type}$ 指向 $t_{eot}$ 的禁止弧. 图 8(a)中为 *t_eot* 到 *p_tm_type* 的禁止弧. 表示在事务单元在一个 *eot* 执行期间($p_{tm\_type}$ 包含一个令牌), 不再允许 *eot* 迁移.




(a) *eot* 生成的 GSPN 网络 (b) *eit* 生成的 GSPN 网络

图 8 事务单元间 *eot* 与 *eit* 错误传播关系的模型转换结果

在图 8(a)中, *eot* 关系及其对 *tm* 状态影响 $et_{eot}$ 经过变换生成了从 *p_eot_state* 到 *p_*tm*_type* 的迁移, 和从 *p_eot_state* 到 *p_eot_state'*的迁移, 其中, *t_eot* 是一个延时变迁, 表示 *eot* 是一个概率事件. *t_eot* 到 *p_tm_type* 的禁止弧防止在 *eot* 已经发生、*p_tm_type* 已获得令牌的情况下 *t_eot* 重复执行.

与之类似, 事务单元向下错误传播行为 *eit* 到 GSPN 的转换包括 *eit* 迁移的转换, 以及 *eit* 造成的事务单元错误状态迁移的转换, 如定义 7. 转换结果构成如图 8(b)所示子网结构.

**定义 7**. 事务单元向下错误传播行为 *eit* 向 GSPN 模型元素转换规则如下:

$$TLEM\_GSPN(eit)=\{t_{eit}\},$$

$$TLEM\_GSPN(et_{eit})=(p_{es0},p_{eit\_dst},a_{pt\_eit},a_{tp\_eit}),$$

其中,

(1) $p_{es_0}$ 是事务单元初始错误状态 $es_0$ 映射得到的位置, 在图 8(b)中记作 *p_es*0;

(2) $t_{eit}$ 标识 *eit* 行为的立即变迁, 在图 8(b)中记作 *t_eit*;

(3) $p_{eit\_dst}$ 是变迁 $t_{eit}$ 的目的位置, 在图 8(b)中记作 *p_eit_dst*, *peit_dst* 属于集合 $P_{es}$;

(4) $a_{pt\_eit}$ 是从源位置 $p_{es_0}$ 指向变迁 $t_{eit}$ 的弧;

(5) $a_{tp\_eit}$ 是从变迁 $t_{eit}$ 指向目的位置 $p_{eit\_dst}$ 的弧.

在图 8(b)中, *eit* 关系经过变换生成了 *p_es*0 到 *p_eit_dst* 的状态迁移. 包括源位置 *p_es*0, 迁移 *t_eit* 和目的位置 *p_eit_dst*, 其中, *t_eit* 是一个立即变迁. 表示传入事务单元的错误必然对当前单元的行为产生影响.

TLEM 中事务单元间的错误传播由 *eot*, *eit* 和事务单元间的事务迁移 *t* 共同完成. 因此在进行 GSPN 模型转换时通常将三者一起转换, 构成图 9 所示的子网结构. 其中, 事务迁移 *t* 转换为源和目的事务单元间的弧, 以及用于防止令牌错误传播的吸收变迁. 具体的转换规则见定义 8.

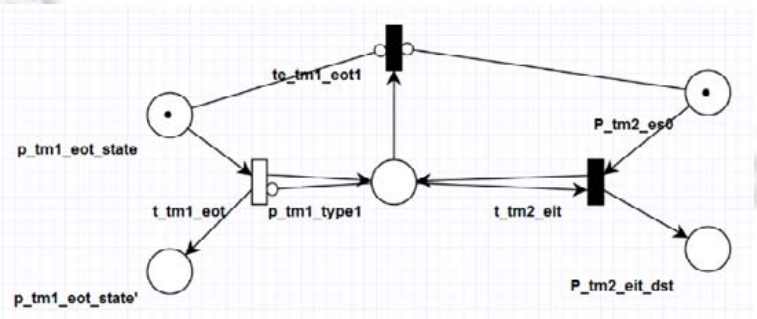



图 9 两事务单元间一组 *eit* 与 *eot* 错误传播关系联结的转换规则

**定义 8**. *eot* 与 *eit* 的匹配向 GSPN 模型元素转换规则.

$$TLEM\_GSPN(eot-eit)=(TE,A_{pt},A_{tp},AE_{pt},AE_{pto\_eot},AE_{pto\_eit}),$$

其中,

(1) $A_{pt}$ 是弧的集合, 包括从源事务单元传播位置 $p_{tm\_type}$ 指向目的事务单元的 *eit* 变迁 $t_{eit}$ 的弧. 在图 9 中表现为从 *p_tm*1*_type*1 到 *t_tm*2*_eit* 的弧. 表示事务单元 *tm*1 的 *eot* 触发了 *tm*2 的 *eit* 行为;

(2) $A_{tp}$ 是弧的集合, 包括从目的事务单元的 *eit* 变迁 $t_{eit}$ 指向源事务单元的传播位置 $p_{tm\_type}$ 的弧. 在图 9 中表现为从 *t_tm*2*_eit* 到 *p_tm*1*_type*1 的反向弧, 表示尽管事务单元 $tm_2$ 在 *eot* 的影响下发生错误状态变迁, 但这个变迁并不会使改变 *tm*1 的错误状态. 在转换得到的 GSPN 模型中, 这个弧使得 *p_tm*1*_type*1 在 *t_tm*2*_eit* 发生后并不会丢失令牌. 同时定义 6 中的禁止弧 $a_{pto\_eot}$ 也能够发生作用, 防止定义 6 所述特殊情况下 *eot* 的重复触发;

(3) $TE=\{te_{tms\_eot}\}$ 为吸收变迁的集合. 在图 9 中记作 *te_tm*1*_eot*1, 由于 $A_{tp}$ 的存在, *p_tm*1*_type*1 在 $tm_2$ 的 *eit* 发生后并不会失去令牌. 虽然在定义 6 所述特殊情况下反向迁移 $A_{tp}$ 能够避免 $tm_1$ 的 *eot* 的重复执行. 但是

在一般情况下($p\_tm1\_eot\_state$ 迁移到 $p_tm1_eot_state'$的情况), $p\_tm1\_type1$ 重新持有的令牌成为一个额外生成的多余令牌, 需要通过吸收变迁 $te\_tm1\_eot1$ 消除;

(4) $AE_{pt}$ 是弧的集合, 包括从源事务单元的传播位置 $p_{tm\_type}$ 指向吸收变迁 $te_{tms\_eot}$ 的弧, 用于传播 $p_{tm\_type}$ 中多余的令牌. 在图 9 中 $AE_{pt}$ 中的元素表现为从 $p\_tm1\_type1$ 到吸收变迁 $te\_tm1\_eot1$ 的弧;

(5) $AE_{pto\_eot}$ 是弧的集合, 包括连接源事务单元 $eot$ 发生时的源错误状态位置 $p_{eot\_state}$ 与 $eot$ 吸收变迁 $te_{tms\_eot}$ 的禁止弧. 在图 9 中表现为从 $p\_tm1\_eot\_state$ 到吸收变迁 $te\_tm1\_eot1$ 的弧. $a_{epto\_eot}$ 阻止了定义 6 所述特殊情况下吸收变迁的执行, 避免令牌被吸收后重新使能 $t_{eot}$, 形成 $t_{eot}$ 与 $te_{tms\_eot}$ 的环路;

(6) $AE_{pto\_eit}$ 是禁止弧的集合, 包括连接目的事务单元初始状态位置 $p_{es_0}$ 与吸收变迁 $te_{tms\_eot}$ 的禁止弧. 在图 9 中表现为从 $p\_tm2\_es0$ 到吸收变迁 $te\_tm1\_eot1$ 的弧, 用于在目的事务单元 $tm_2$ 能够接收错误并进行向下错误传播时, 禁止吸收变迁消除 $p_{tm\_type}$ 中的令牌. 防止源事务单元 eit 在具备执行条件时, 令牌被意外吸收.

定义 5–定义 8 给出了 TLEM 中硬件构件内部错误传播行为向 GSPN 网络模型转换的规则, 而 TLEM 又通过向内错误传播 *in_PROP* 与向外错误传播 *out_PROP* 与 EM 错误模型进行组装. 在将 *in_PROP* 和 *out_PROP* 关联关系向 GSPN 模型转换时, *in_PROP* 不产生实质上的 GSPN 网络元素, 只用于确定接收传入错误的事务单元. 即, 外界的错误直接作用于 *in_PROP* 所指向的事务单元的向下错误传输 *eit* 上, 而向外错误传播 *out_PROP* 在进行模型转换时, 被用于确定传出错误的事务单元 *tm_src*. 以后文第 3.2 节中图 14 处理器构件为例, 指令执行 EU 的 $es_1$ 是传出错误的事务单元. 因此, *out_PROP* 将该事务单元的 EOT 与系统错误模型中线程的 EM 连接. 由于 TLEM 在设计时已考虑到与 EMA 的兼容, *out_PROP* 的语义与 EMA 中向外错误传播保持一致. 所以硬件构件 TLEM 模型中的 *out_PROP* 与相邻 AADL 构件 EMA 模型向内错误传播匹配关系的传播转换规则可以参考文献[12], 两者一起进行转换后构成图 10 所示的连接关系子图. 细节这里不再赘述.

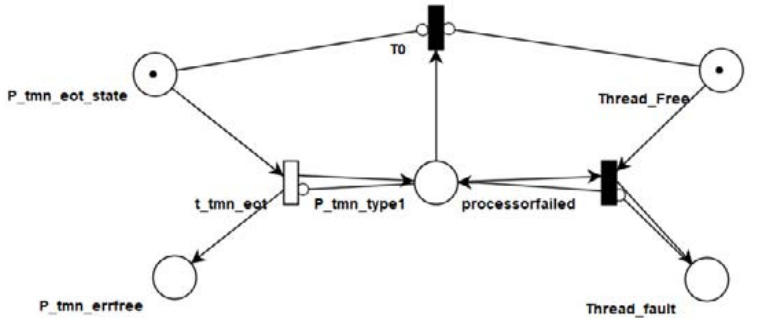



图 10 硬件构件事务级错误模型与 AADL 构件错误模型间的向外错误传播关系转换

## 3 工具实现与案例分析

基于前面建模与分析理论, 采用 Eclipse 插件开发技术, 开发了 AADL 模型的软硬件综合可靠性分析工具原型系统, 系统架构如图 11 所示. 该工具支持嵌入式系统模型、系统软件错误模型、系统硬件事务级模型的建模, 同时支持软硬件综合 AADL 模型向 GSPN 模型的自动转换以及软硬件综合的可靠性分析功能.

### 3.1 空气增压系统空压机控制单元架构建模

空气增压系统是利用飞行器发动机前级风扇进行空气加压和座舱供氧的航空机电系统. 空气增压系统由空压机控制器、外设传感器及机电执行机构共同构成, 其中, 空压机控制器用于根据调控指令控制机电执行机构提供稳定可靠的增压空气供应, 是一个基于 PowerPC 处理器的嵌入式应用系统. 本文以某型飞行器的空气增压系统的空压机控制器作为演示与验证案例的目标系统, 说明应用 TLEM 进行软硬件综合可靠性建模与分析的方法. 首先, 空压机控制器作为一个嵌入式系统由如下部分构成.

(1) 软件子系统包括输入进程、输出进程、控制进程、BIT 周期自检进程、故障处理进程、存储进程这 6

个系统进程, 内部各自包含多个子任务线程. 控制进程是系统功能主体进程, 输入进程与输出进程与控制进程功能密切相关. 其余 3 个进程故障不影响系统主体功能, 所以本文建模过程中暂不考虑这 3 个进程.

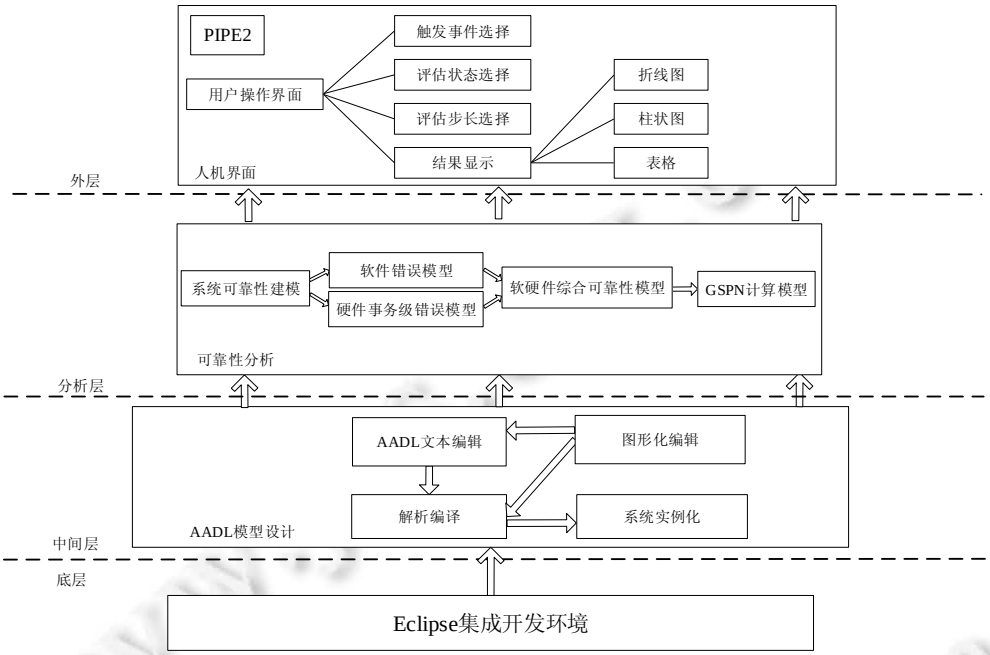



图 11 软硬件综合可靠性分析工具软件架构

(2) 硬件子系统包括嵌入式计算平台模型和外部设备(即传感器和执行机构的接口硬件). 进行系统架构建模时, 外设硬件均被映射为 AADL Device 构件, 通过数据端口与空压机控制系统建立连接. 由于本次演示验证中重点验证硬件构件事务级错误模型对系统可靠性分析的影响, 不考虑外部环境引入的错误, 因此这里忽略了外设的建模, 硬件子系统架构模型中仅包含计算平台模型, 内含处理器构件、总线构件及存储器构件.

(3) 空压机控制器系统软件部署在符合 ARINC653 规范的分区系统上. 系统中总共包含 3 个分区, 每个分区包含 1–3 个进程不等. 在演示用例中展示的 3 个进程正好分布在 3 个分区中. 采用 AADL 及其 Annex F ARINC653 附录建立系统架构模型, 如图 12 所示. 系统架构设计了 3 个虚拟处理器构件, 全部映射到 PowrePC 处理器, 并分别与用例中的输入进程、输出进程、控制进程构件进行绑定, 构成了 3 个系统分区.

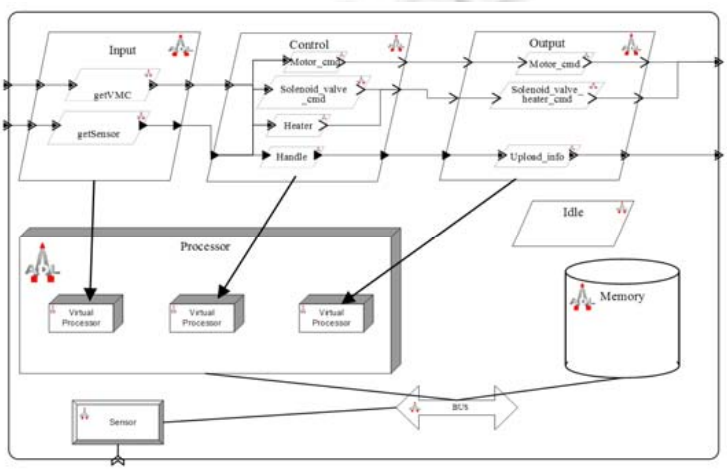



图 12 空气增压系统空压机控制器 AADL 架构模型

### 3.2 空压机控制器可靠性模型建模

(1) 空压机控制器的 EMV2 附录可靠性建模

根据第 2.2 节所述过程建立空压机控制器的 EMV2 错误模型. 其中, 部署在嵌入式计算平台上的线程是控制器软件子系统的基础构件. 当线程出现运行错误时, 会随数据流横向传播, 并向上层构件产生影响, 经过层层传递最终影响空压机控制器错误状态. 在本例中线程采用统一的错误模型, 描述线程在本身缺陷、环境因素和输入错误作用下的状态迁移. 以控制进程中的电机控制线程为例, 线程的 EMV2 错误模型如图 13 所示, 其中错误状态与状态迁移说明见表 2. 线程错误事件发生概率参数来源于空气增压系统空压机控制器需求说明文件, evReset 事件发生概率参数参考了文献[15]中相似线程模型.

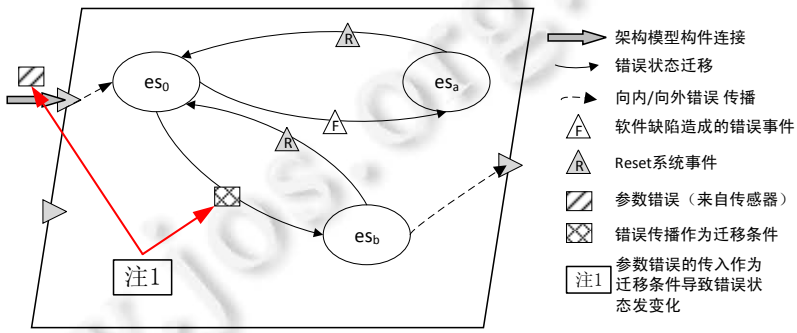



图 13 空压机控制系统中电机控制线程错误模型示意图

表 2 电机控制线程 EMV2 错误模型说明

| 名称 | 类型 | 说明 |
|---|---|---|
| $es_0$ | 错误状态 | 线程错误行为状态机初始态, 表示无故障运行状态 |
| $es_a$ | 错误状态 | 软件内部缺陷造成的错误状态 |
| $es_b$ | 错误状态 | 外部传入错误造成的错误状态, 会造成错误的向后传播 |
| epFail | 错误迁移 | $es_0$-[evDefect]->$es_a$, 线程在程序缺陷引发的错误事件作用下崩溃, 导致控制进程故障, 其中, evDefect 发生概率服从泊松分布, 默认取值 0.01 |
| epParam | 错误迁移 | $es_0$-[evParam]->$es_b$, 线程在收到输入参数错误的影响, 得到错误输出, 继续向后继线程传播, 其中, evParam 事件发生概率服从泊松分布, 默认取值 0.05 |
| epR | 错误迁移 | $es_a$, $es_b$-[evReset]->$es_0$, 线程在故障处理线程作用下, 按照周期调度时间重启, 恢复无故障状态, 其中, evReset 事件发生概率服从泊松分布, 默认取值 0.7 |

(2) 空压机控制器的软硬件综合可靠性建模

分析图 12 所示空压机控制器单元架构, 可知系统中处理器构件是软硬件子系统交互通道一个汇聚点. 软件子系统错误通过进程与处理器的绑定关系向硬件子系统扩散传播, 硬件故障引起的错误同样会通过处理器的转化转换为软件错误. 因此处理器是该控制单元中的安全关键硬件构件, 需要使用 TLEM 附录子语言建立事务级错误模型, 并与系统 EMV2 错误模型进行组装.

按照第 2.1 节所示步骤建立空压机控制器 PowerPC 处理器的事务级错误模型, 得到如图 6 所示处理器事务级架构模型, 以及图 14 所示处理器事务级错误模型. 表 3 中以处理器的指令译码(IDecoder)事务单元为例, 对模型内容进行了说明.

处理器的事务级错误模型以事务单元的错误状态迁移行为刻画了两类软硬件子系统之间的错误传播与转化.

(a) 软件子系统计算得到的错误参数传入硬件子系统造成硬件故障. 例如: 计算得到了错误的跳转指令目的地址, 在传入处理器硬件后, 经过取指单元处理取到了错误的指令内容, 送入指令译码单元引发错误迁移 etIDFaultIns.

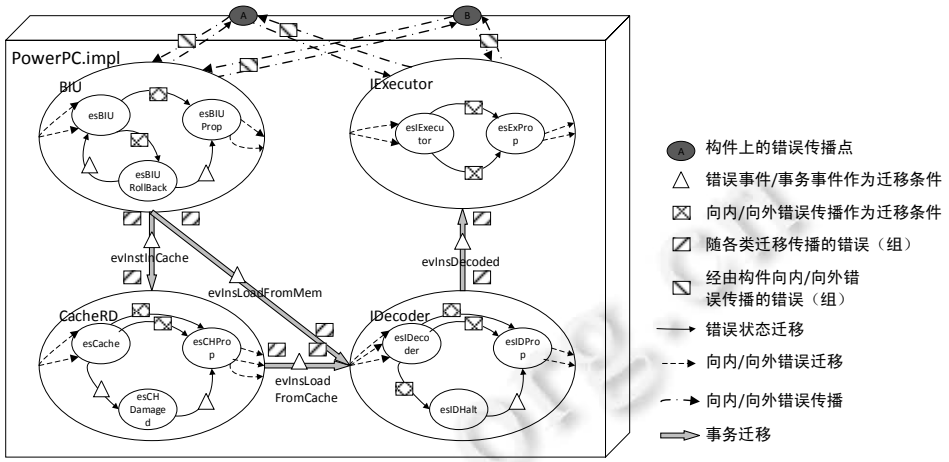



图 14　空压机控制器中 PowerPC 处理器事务级错误模型示意图

表 3　以指令译码单元为例的 PowerPC 处理器事务级错误模型说明

| 名称 | 类型 | 说明 |
|---|---|---|
| esIDecoder | 事务级错误状态 | 状态机初始态 |
| esIDProp | 事务级错误状态 | 指令译码单元未能识别出输入错误，执行事务并导致错误的继续传播 |
| esIDHalt | 事务级错误状态 | 执行译码单元识别出输入错误，停止执行，从而将软件故障引发的错误转化为处理器异常硬件错误 |
| evIDErrorIns | 错误事件 | 传入了错误的指令，会产生错误执行结果 |
| evIDErrorData | 错误事件 | 传入的指令参数错误，同样会产生错误的执行结果<br>错误事件发生概率服从泊松分布，默认取值 0.81 |
| evIDFaultIns | 错误事件 | 输入的数据不属于指令集，不可解析 |
| etIDErrorIns | 错误迁移 | esIDecoder-[evIDErrorIns]->esIDProp 因为执行了错误指令发生错误，会向后传播 |
| etIDErrorData | 错误迁移 | esIDecoder-[etIDErrorData]->esIDProp 因为指令使用了错误操作数造成错误，<br>同样会向后传播. EIT 事件发生概率服从泊松分布，默认取值 0.9，即大概率向后传播 |
| etIDFaultIns | 错误迁移 | esIDecoder-[evIDFaultIns]->esIDHalt 因为输入非机器指令，不可解析，导致硬件故障 |
| etIDTimeOut | 错误迁移 | esIDHalt-[evTimeOut]->esIDProp 硬件故障向上层传播，处理器失效 |

(b) 硬件子系统失效造成的错误会传入软件子系统并进一步造成故障. 例如: 因为环境干扰高速缓存发生了位翻转, 造成取出的指令操作数错误, 这一错误无法在处理器内部被检测, 因此会持续向后传播, 在译码单元造成了错误迁移 etIDErrorData. 译码单元错误状态 esIDProp 继续向后传播这一错误到执行单元, 最终通过处理器构件的 out-Prop 传播行为通过线程与处理器的绑定关系对线程造成影响. 按照第 2.1 节中所述步骤, 通过处理器构件的向外错误传播事件建立处理器到 Motor_cmd 线程构件之间的错误传播, 从而实现 TLEM 模型与 EM 模型的模型组装, 如图 15 中所示, 来自处理器的向外错误传播导致线程从 $es_0$ 状态迁移到 $es_c$ 状态, 如图 15 中注 2 所示.

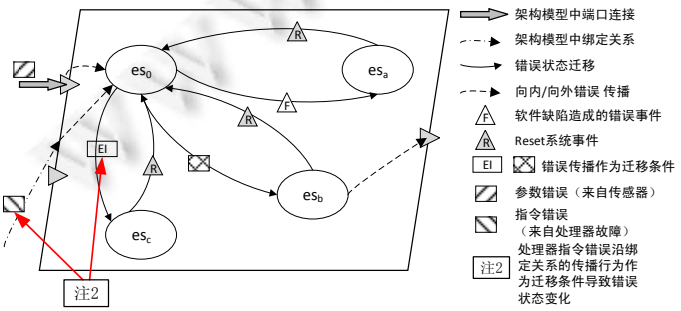



图 15　引入处理器指令错误后的空压机控制器电机控制线程错误模型

### 3.3 空压机控制器硬件综合可靠性分析

(1) 故障源识别

使用AADL及其附录子语言建立的目标系统可靠性模型中使用错误行为状态机刻画了系统中各级构件与子系统在内因、外因作用下进入错误状态的行为. 这些错误状态的变迁行为都可被视作潜在的系统故障源. 在可靠性模型中, 还通过错误传播刻画了构件错误对相邻构件及上层系统的影响. 当以顶层系统的错误状态作为顶层错误事件进行考察时, 可以沿模型中记载的错误传播路径进行反向溯源. 如果某个构件的错误行为最终影响到系统的错误状态, 即可将其标识为影响系统可靠性的故障源.

(2) 故障传播与影响分析

由于故障源在系统中所处的层次不同, 传播路径也有差异, 能够对系统可靠性产生影响的程度也不同. 论文实现的软硬件综合可靠性分析工具支持故障传播与影响程度的分析. 分析过程首先将目标系统软硬件综合的可靠性模型转换为 GSPN 模型, 然后导入到 PIPE2 工具软件进行定量计算. 模型转换依据本文第 2.2 节中定义的转换规则进行, 转换结果以 xml 格式文件缓存, 并导入到 PIPE2 工具. 其次, 分析过程选择需要进行故障传播影响强度分析的故障源与目标系统错误事件(通常是系统错误状态), 通过设置故障源的故障概率从 0 到 1 逐步增长, 对目标系统出错概率在其影响下的变化过程进行仿真, 计算故障源具有不同发生概率情况下系统进入出错状态的概率.

例如: 在空压机控制器演示验证模型中, 顶层系统具有 4 个错误状态.

Errorfree: 无错误正常运行态, 是错误行为状态机的初始态. 当 Input、Output、Control 这 3 个进程都处于 errorfree 状态时, 系统才会处于 errorfree 状态;

Error1: 显示错误状态 display_fault. 当 input 进程和 output 进程同时处于 nondatacmd(无数据显示)状态时, 系统将处于 display_fault 状态;

Error2: 数据处理错误态 data_fault. 当 Input、Output、Control 进程都处于 datahandlefault(数据处理错误时)状态时, 系统将处于 data_fault 状态;

Error3: 未知错误状态 error_fault. 当 Input 进程和 Control 进程同时进入 errorcmd(错误状态)状态时, 系统将处于 error_fault 状态.

为了验证硬件构件错误与系统软件之间的错误传播, 实验选择了处理器构件内部指令执行器(ACCSystem_imp_instance.CPU.IExecutor)的输入错误地址错误事件(errFaultAddress)作为故障源, 以系统整体错误状态作为考察对象(ACCSystem_imp_Instance_errorfree,error1,error2,error3)进行分析验证, 得到分析结果如图 16 所示. 图中横坐标是处理器内部指令执行器单元进入发生 errFaultAddress 错误的概率, 数值上从 0.00 到 1.00 变化, 每次变化步长是 0.1. 纵坐标显示系统进入某错误状态的发生概率. 图中曲线点划线标识 errorfree 态, 短划线标识 error1 态, 点线标识 error2 态, 连贯线标识 error3 态.

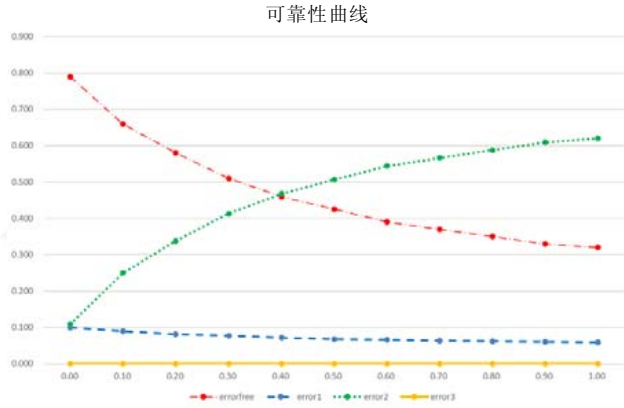



图 16 处理器故障对系统错误状态影响的可靠性分析结果

从仿真分析结果可看到，随着 EIT 错误事件的发生概率从 0 到 1 不断增加，系统处于正常状态 errorfree 的概率从 0.79 下降到 0.32，处于 error1 状态的概率变成了从 0.1 下降到 0.06，处于 error2 状态的概率变成了从 0.11 上升到 0.62，处于 error3 状态的概率为 0 保持不变. 由此可知，处理器硬件内部的 errFaultAddress 错误会对系统发生数据错误故障产生显著影响.

根据图 14 所述硬件构件事务级错误模型，errFaultAddress 错误是一个硬件无法识别的错误类型，表示输入 IExecutor 单元的指令方位地址并非程序正确执行应得的地址，但是因为并没有超出程序运行所分配的地址边界，未能触发处理器报警，因而得以另 IExecutor 单元迁移进入 esExProp 状态，将此错误继续向上传播，经由处理器与线程之间的绑定关系，最终令线程访问了错误的存储空间，发生了数据处理错误(datahandleError). 因为 Input、Output、Control 进程中的线程均与处理器构件 CPU 进行了绑定，所以在基于架构模型的可靠性分析中，认为 Input、Output、Control 这 3 个进程将同时受到 CPU.IExecutor 事务单元 errFaultAddress 错误的影响，并进入 datahandlefault 状态，从而令系统整体 ACCSystem_imp_Instance 也进入到 Error2 所指 data_fault 错误状态. 在图 16 所示实验结果中，随着 errFaultAddress 错误发生概率的增长，系统进入 Error2 状态的概率显著增长.

(1) 软硬件综合的可靠性分析

嵌入式系统的可靠性不仅受软硬件子系统自身失效概率的影响，还受软硬件交联失效行为的影响. 如果在系统可靠性分析与评估过程中忽视软硬件子系统交互造成的故障传播与转化，会对分析结果带来误差.

在空压机控制器演示验证模型，output 进程 failed 状态是一个直接影响系统可靠性的软件失效状态. 如果基于 AADL 及其 EMV2 附录子语言建立空压机控制器的可靠性模型，并使用本文实现的软硬件综合可靠性分析工具进行分析时，得到的结果如图 17 所示. 使用软硬件综合可靠性分析工具对第 3.2 节建立的软硬件综合可靠性模型进行分析时，得到分析结果如图 18 所示. 将两次分析结果进行对比，见表 4.

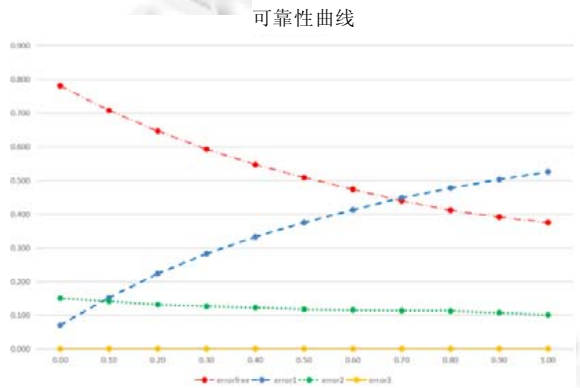



图 17 无处理器事务级模型的可靠性分析结果图

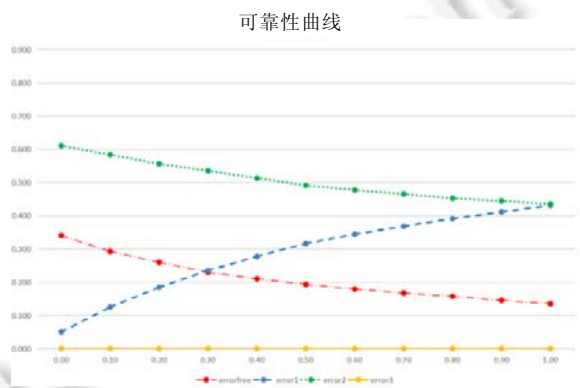



图 18 包含处理器事务级模型的可靠性分析结果

表 4 包含与不包含处理器事务级错误模型的系统可靠性分析结果对比

| Output Failed 发生概率 | 系统 errorfree 态 | | 系统 error1 态 | | 系统 error2 态 | | 系统 error3 态 | |
|---|---|---|---|---|---|---|---|---|
| | **A** 注 1 | **B** 注 2 | **A** | **B** | **A** | **B** | **A** | **B** |
| 0 | 0.780 | 0.340 | 0.07 | 0.05 | 0.15 | 0.610 | 0 | 0 |
| 0.5 | 0.517 | 0.197 | 0.363 | 0.316 | 0.12 | 0.487 | 0 | 0 |
| 1 | 0.375 | 0.135 | 0.525 | 0.430 | 0.1 | 0.435 | 0 | 0 |

注 1: A——使用未包含处理器事务级模型的可靠性模型进行分析

注 2: B——使用包含处理器事务级模型的可靠性模型进行分析

对比两张图可以看出，随着 Failed 错误事件的发生概率从 0 到 1 不断增加时，相较于无硬件事务级错误模型的空压机系统来说，添加了处理器事务级错误模型后的空压机控制器处于 errorfree 正常状态的概率明显降低，处于 error2:data_fault 的概率明显升高. 系统在受到执行平台构件处理器自身的可靠性影响后，即使在 Failed 错误事件不发生的情况下，依然会有 61%的概率产生 error2 的错误，产生此结果的原因是因为处理器在

处理线程任务的过程中, 由于受处理器内部各事务单元间事务传递过程中可能产生的错误影响, 从而导致系统进入 error2 状态, 最终影响系统的可靠性概率. 又因为 Failed 事件会导致系统进入 error1 错误状态, 随着 Failed 错误事件的发生概率不断升高, 系统处于 error1 状态的概率不断增加, 因此系统处于 errorfree 和 error2 状态的概率也随之不断降低.

## 4 总结与展望

本文主要研究了嵌入式系统中综合考虑软硬件情况下 AADL 模型的可靠性, 针对 EMV2 对安全关键硬件中错误的产生与传播描述不足的问题, 进行硬件事务级错误模型扩展, 完成针对硬件平台的错误模型建模, 并与软件错误模型、系统架构模型综合, 组成软硬件综合的可靠性模型, 在此基础上, 提出事务级错误模型向 GSPN 模型的转换规则, 基于此进行软硬件综合的可靠性分析. 该方法可以在设计初期从系统架构层面完成复杂嵌入式系统的软硬件综合一体化建模, 综合有效考虑软硬件对系统的影响, 通过分析错误在嵌入式系统内部全部子系统中的扩散过程及影响范围, 进而考察其对系统的可靠性影响变化趋势, 并通过这种变化趋势指导系统架构模型的设计优化. 本文设计并实现了软硬件综合的 AADL 可靠性建模与分析工具原型, 结合飞机空气增压机系统开展实验分析, 通过实验可知: 在 AADL 中, 使用论文提出的方法建立可靠性模型, 能够考察硬件子系统内部错误对软件子系统、以及对嵌入式系统整体产生的影响趋势; 且相较于单一利用 EMV2 建立软件错误模型而言, 结合硬件事务级错误模型可以更加准确的反应系统实际运行情况下的可靠性概率.

此外, 本文基于 AADL 扩展的硬件事务级错误模型与 EMV2 之间的耦合需要继续完善, 目前本文工作只针对关键硬件构件处理器进行了事务级模型建模, 事务级模型与 AADL 架构模型之间的耦合方式只涵盖了进程绑定处理器的连接关系, 尚未对工作的可扩展性进行测试验证. 例如演示验证系统中的传感器设备可视作一个安全关键的外部数据源硬件构件, 传感器故障造成的错误输出会形成怎样的错误传播, 如何影响系统中部署应用程序的运行, 还需要对嵌入式系统中软硬件之间的故障传播机理进行进一步的研究才能准确进行可靠性模型建模. 这些研究内容将在后继工作中展开.

**References**:


[1] Iyer RK, Velardi P. Hardware-related software errors: Measurement and analysis. IEEE Trans. on Software Engineering, 1985, SE-11(2): 223−231. [doi: 10.1109/TSE.1985.232198]

[2] Roy DS, Murthy Ch. Reliability analysis of phasor measurement unit incorporating hardware and software interaction failures. IET Generation, Transmission & Distribution, 2015, 9(2): 164−171. [doi: 10.1049/iet-gtd.2014.0115]

[3] Immonen A, Niemelä E. Survey of reliability and availability prediction methods from the viewpoint of software architecture. Software & System Modeling, 2008, 7(49): 49−65. [doi: 10.1007/s10270-006-0040-x]

[4] Sinha S, Goyal NK, Mall R. Survey of combined hardware-software reliability prediction approaches from architectural and system failure viewpoint. Int'l Journal of System Assurance Engineering and Management, 2019, 10(4): 453−474. [doi: 10.1007/s13198-019-00811-y]

[5] Purwantoro Y, Bennett S. Decomposition technique for integrated dependability evaluation of hardware-software systems using stochastic activity networks. In: Proc. of the 25th EUROMICRO Conf., Informatics: Theory and Practice for the New Millenium. Milan: IEEE, 1999, 2: 142−145. [doi: 10.1109/EURMIC.1999.794773]

[6] Yu M, He ZhY, Qian QQ. Reliability analysis of combined hardware/software system based on Markov process. Acta Electronica Sinica, 2010, 38(2): 473−479 (in Chinese with English abstract).

[7] Compare M, Baraldi P, Bani I, *et al*. Industrial equipment reliability estimation: A Bayesian Weibull regression model with covariate selection. Reliability Engineering & System Safety, 2020, 200: 106891. [doi: 10.1016/j.ress.2020.106891]

[8] Jiang Y, Zhang HH, Liu H, *et al*. System reliability calculation based on the run-time analysis of ladder program. In: Bertrand M, Luciano B, eds. Proc. of the 9th Joint Meeting on Foundations of Software Engineering. Association for Computing Machinery, 2013. 695−698. [doi: 10.1145/2491411.2494570]

[9] Jiang Y, Zhang HH, Song X Y, *et al*. Bayesian-network-based reliability analysis of PLC systems. IEEE Trans. on Industrial Electronics, 2013, 60(11): 5325−5336. [doi: 10.1109/TIE.2012.2225393]

[10] Kurtoglu T, Tumer IY. A graph-based fault identification and propagation framework for functional design of complex systems. Journal of Mechanical Design, 2008, 130(5): 051401−051409. [doi: 10.1115/1.2885181]

[11] Jensen DC, Tumer IY, Kurtoglu T. Modeling the propagation of failures in software driven hardware systems to enable risk-informed design. In: Proc. of the ASME 2008 Int'l Mechanical Engineering Congress and Exposition. Boston: ASME, 2008, 16: 283−293. [doi: 10.1115/IMECE2008-68861]

[12] Tumer IY, Smidts CS. Integrated design-stage failure analysis of software-driven hardware systems. IEEE Trans. on Computers, 2011, 60(8): 1072−1084. [doi: 10.1109/TC.2010.245]

[13] Feiler PH, Goodenough JB, Gurfinkel A, *et al*. Reliability improvement and validation framework. Technical Report, CMU/SEI-2012-SR-013, Pittsburgh: Software Engineering Institute, Carnegie Mellon University, 2012. [doi: 10.1184/R1/6583043. v1]

[14] Dong YW, Wang GR, Zhang F, *et al*. Reliability analysis and assessment tool for AADL model. Ruan Jian Xue Bao/Journal of Software, 2011, 22(6): 1252−1266 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4014.htm [doi: 10.3724/SP.J.1001.2011.04014]

[15] Wei XM, Dong YW, Li XL, *et al*. Architecture-level hazard analysis using AADL. Journal of Systems and Software, 2018, 137: 580−604. [doi: 10.1016/j.jss.2017.06.018]

[16] Xiao MR, Dong YW, *et al*. Architecture-level particular risk modeling and analysis for a cyber-physical system with AADL. Frontiers of Information Technology & Electronic Engineering, 2020, 21: 1607−1625. [doi: 10.1631/FITEE.2000428]

[17] Shin YJ, Tahar S, Habibi A. A systemc transaction level model for the MIPS R3000 processor. In: Proc. of 4th Int'l Conf. on Sciences of Electronic, Technologies of Information and Telecommunications. Tunisia: EEE, 2007. 1−8.

**附中文参考文献:**

[6] 于敏, 何正友, 钱清泉. 基于 Markov 过程的硬/软件综合系统可靠性分析. 电子学报, 2010, 38(2): 473−479.

[14] 董云卫, 王广仁, 张凡, 高磊. AADL 模型可靠性分析评估工具. 软件学报, 2011, 22(6): 1252−1266. http://www.jos.org.cn/1000-9825/4014.htm [doi: 10.3724/SP.J.1001.2011.04014]

**陆寅**(1975−), 男, 博士, 讲师, CCF 专业会员, 主要研究领域为嵌入式系统设计, 嵌入式系统可靠性工程.

**郭鹏**(1987−), 男, 高级工程师, CCF 专业会员, 主要研究领域为嵌入式系统建模和仿真, 调度规划.

**秦树东**(1995−), 男, 硕士, 主要研究领域为嵌入式系统分功能属性分析, 嵌入式系统可靠性工程.

**董云卫**(1968−), 男, 博士, 教授, 博士生导师, CCF 杰出会员, 主要研究领域为嵌入式软件设计与验证, 信息物理融合系统, 嵌入式软件智能合成方法.