

HUNAN UNIVERSITY



湖南大学

# 基于AADL的车用嵌入式实时系统 建模方法研究

---

答 辩 人: 杨阳

指导老师: 李仁发教授, 曾庆光教授

研究方向: 嵌入式软件

# 大纲

---

1. 研究背景及意义
2. 基础理论
3. 本文主要工作
4. 总结与展望

# 研究背景和意义

---

## □ 背景

- 汽车电子化程度不断提高，如何设计高质量的车用嵌入式实时系统，节约开发时间成为研究的重点。
- 本课题来源于国家核高基重大专项：实时嵌入式操作系统及开发环境。

## □ 意义

- 本文为车用嵌入式实时系统开发提供一种可靠的模型设计方法，有助于增强系统的灵活性，提高开发效率，促进车用嵌入式系统实时性能的改进。

# 基础理论—AADL

- 2004年，SAE提出体系结构设计语言AADL(Architecture Analysis and Design Language)。
- AADL是一种专用于复杂嵌入式实时系统的组件设计方案。
- 支持时间关键性系统的性能分析和可靠性的早期预测。

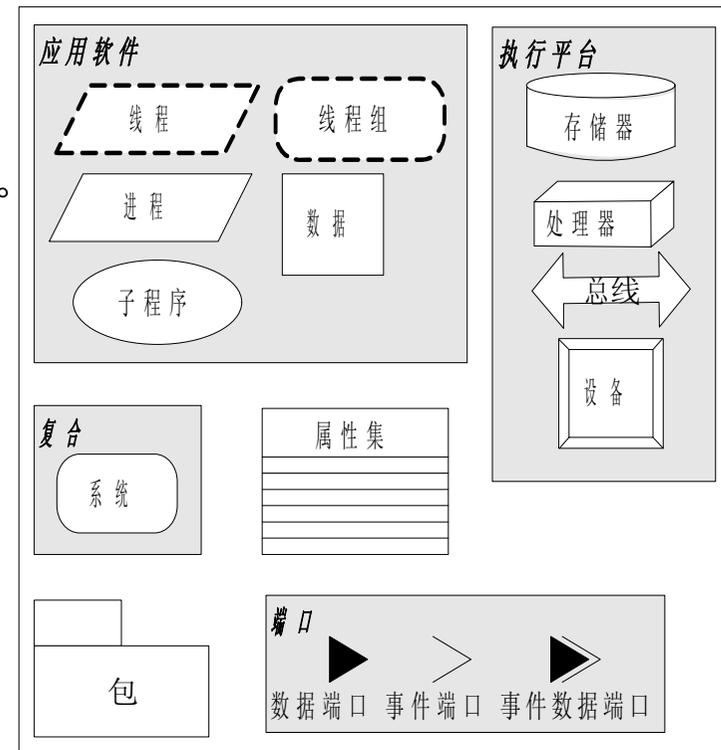


图1 AADL建模元素

# 基础理论—AADL

- AADL建模的组件声明构成
- 完整的AADL模型包括:
  - 静态体系结构
  - 执行模型
  - 行为扩展

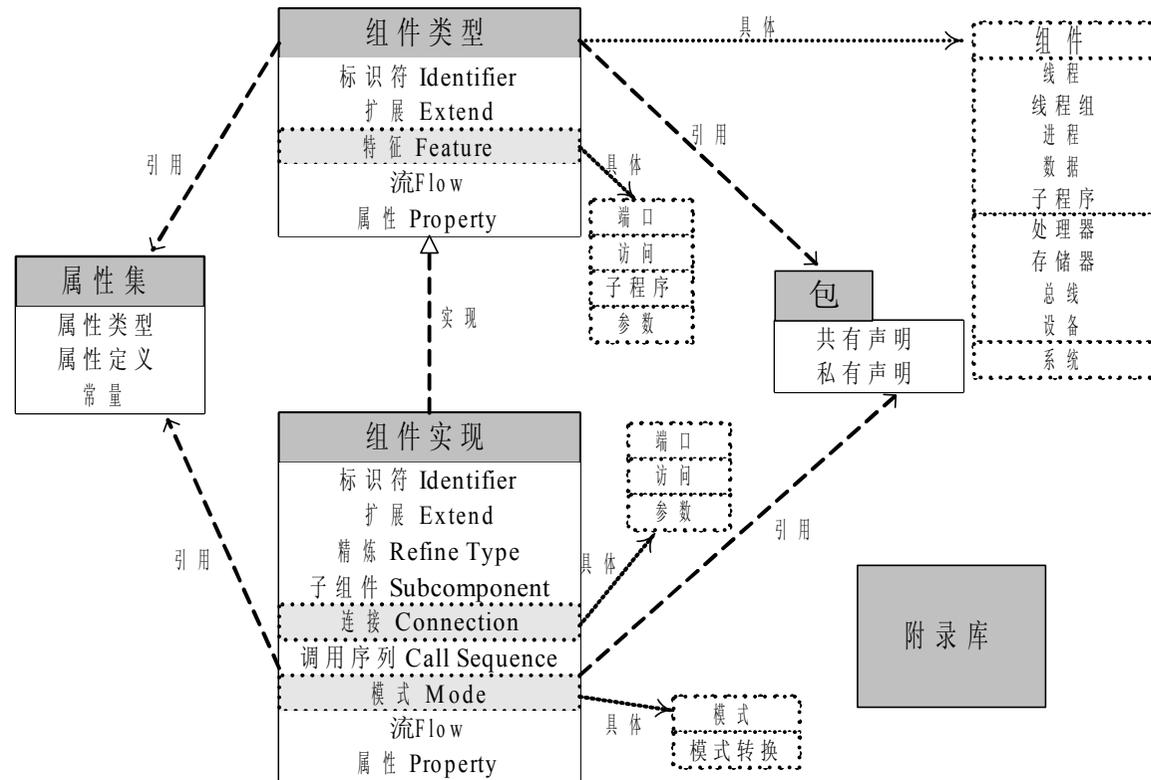


图2 AADL组件声明

# 主要工作

---

1. 模型需求分析
2. AADL模型设计
3. CCS建模实例
4. CCS模型验证

# 需求分析

需求分析 > 模型设计 > 建模实例 > 模型验证

- OSEK/VDX是汽车电子开发领域的主流标准
- 为汽车电子系统开发提供开放式架构和统一接口

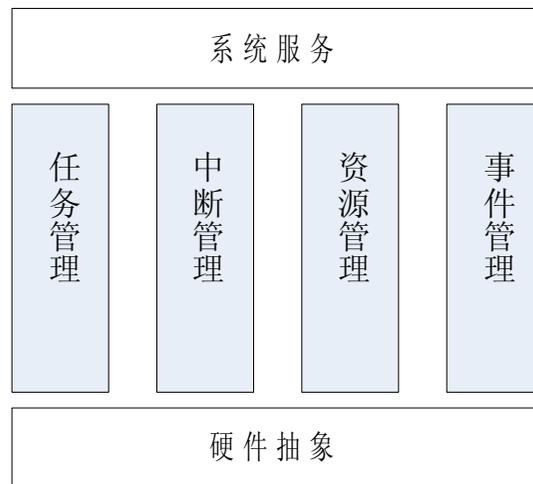


图3 兼容OSEK OS的模型功能需求

- VSP模型结合架构和行为级设计

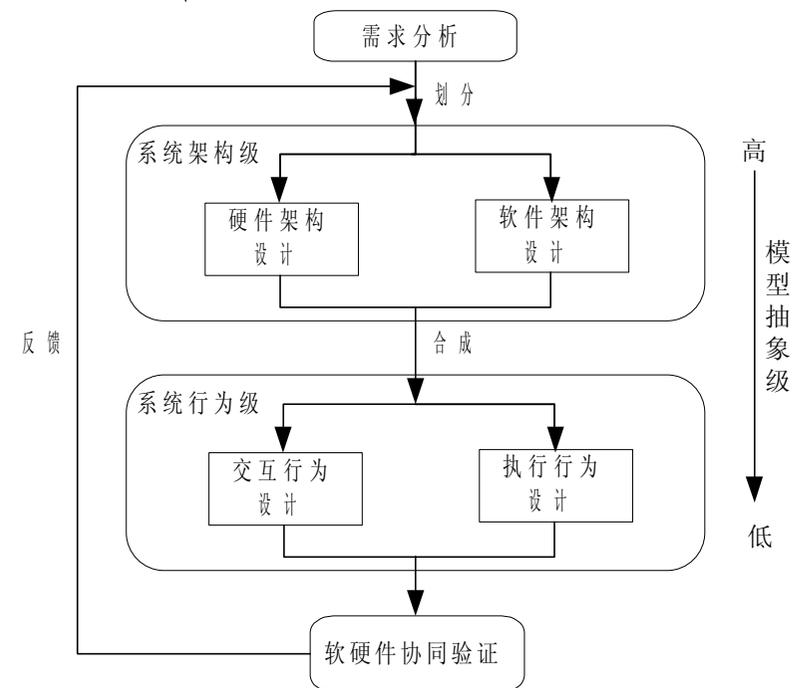


图4 车用嵌入式实时系统VSP模型设计

# AADL模型设计——系统级建模方法

需求分析 模型设计 建模实例 模型验证

## 兼容OSEK OS规范的ECU视图

- 应用层
- OSEK OS
- 硬件层

## 将ECU需求映射到AADL架构级和行为级模型中

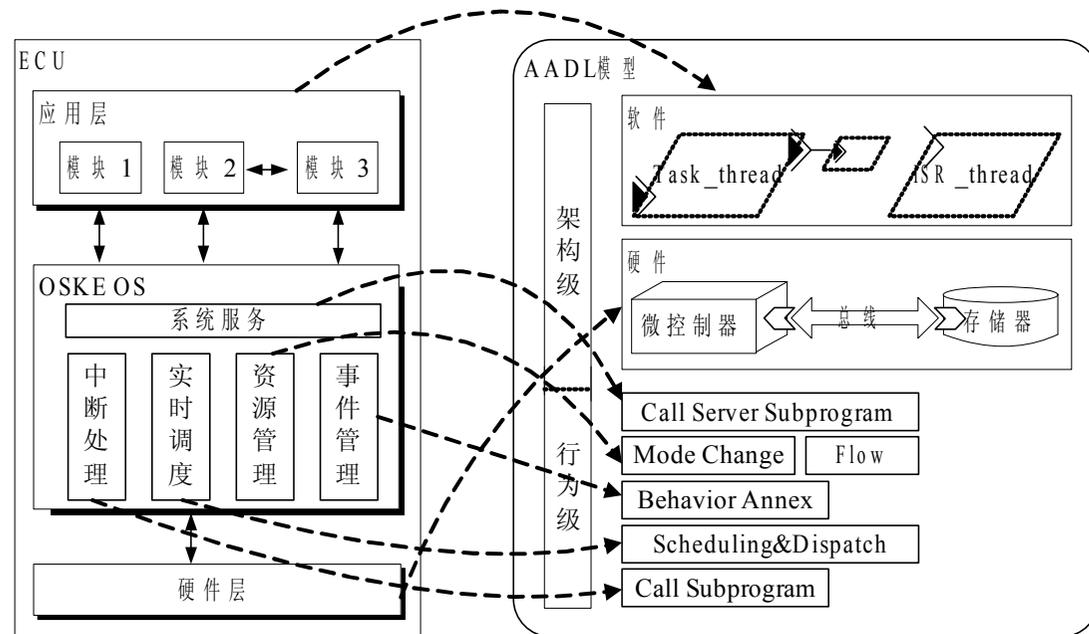


图5 兼容OSEK OS的AADL建模方案

# AADL模型设计——软件架构模型

需求分析 模型设计 建模实例 模型验证

```

----- ISR1模型设计 -----
thread implementation ISR1.impl
properties
  ---other properties
  Priority=>***; ---Higer than Task
  Dispatch_Protocol => Aperiodic;
annex behavior {**
  ---thread behavior
  **};
end ISR1.impl;

----- ISR2模型设计 -----
thread implementation ISR2.impl
properties
  ---other properties
  Priority=>***; ---Higer than Task
  Dispatch_Protocol => Aperiodic;
annex behavior {**
  ---thread behavior
  **};
Calls {
  identifier: subprogram server_subprogram.impl;
}; ---call a server subprogram
end ISR2.impl;
  
```

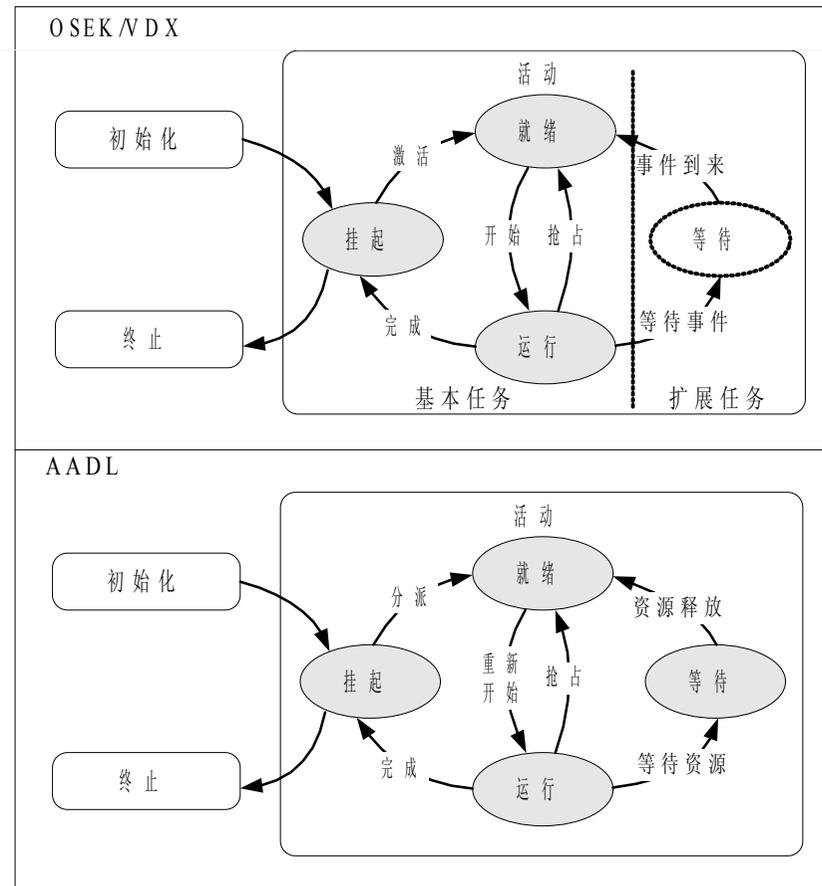


图6 任务与线程的状态转换对比

# AADL模型设计——硬件架构模型

需求分析 模型设计 建模实例 模型验证

- 处理器
  - Scheduling protocol: RMS
  - Cycle-time: 1000ps
- 存储器
  - Memory protocol: read\_write
  - Word-size: 8Bits
- 总线

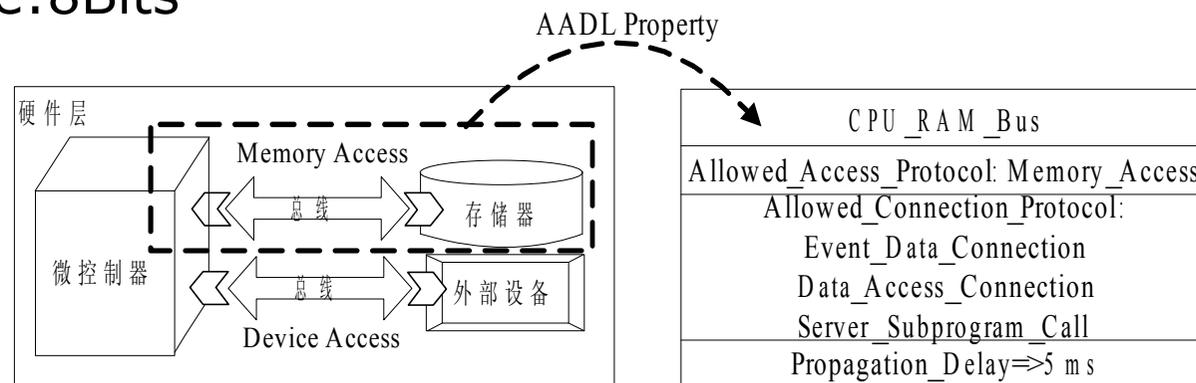


图7 AADL硬件层设计

# AADL模型设计——交互行为模型

需求分析 模型设计 建模实例 模型验证

- 基于端口的通信
- 资源同步

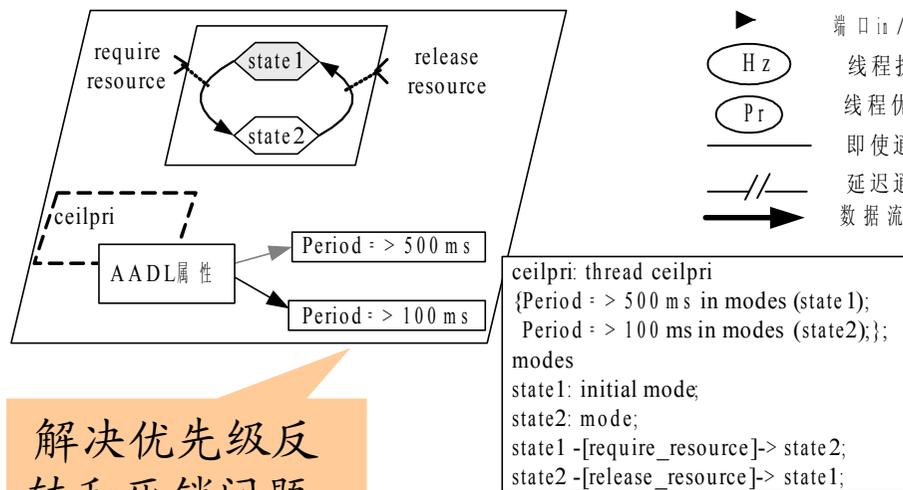
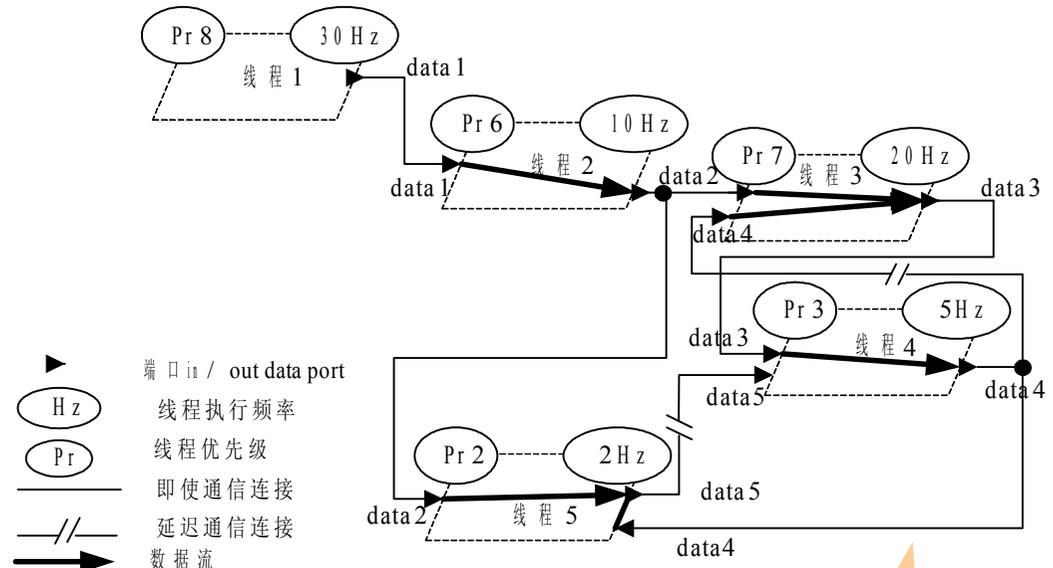


图9 AADL优先级天花板协议设计



# AADL模型设计——执行行为模型

需求分析 模型设计 建模实例 模型验证

- 实时调度
- 中断处理
- 系统服务

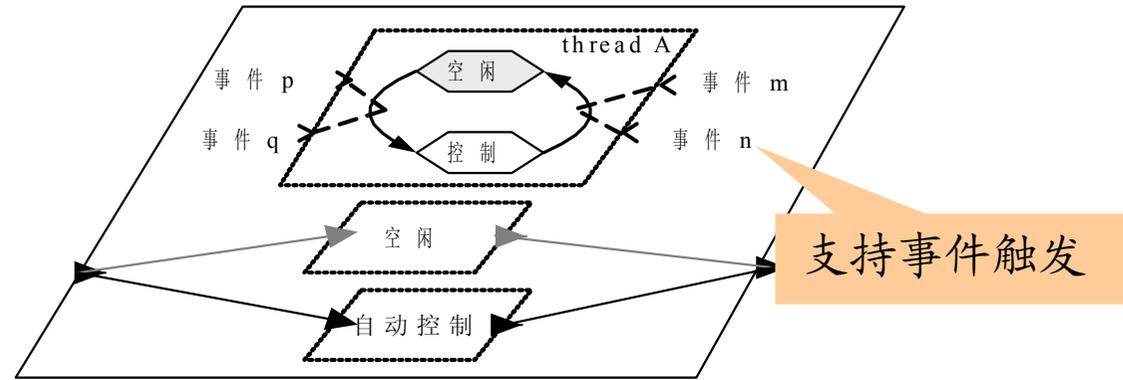


图10 模式切换

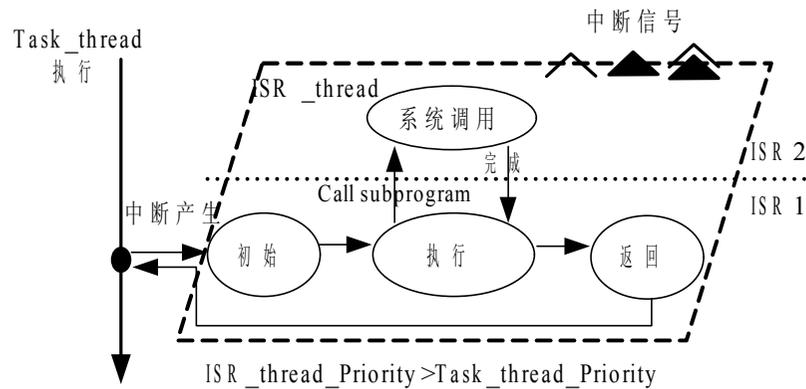


图11 ISR\_thread的调用执行

表1 OSEK与AADL中断设计相关元素对应关系

OSEK中断相关元素	AADL中断相关元素
Task	Task_thread
ISR1	ISR1_thread
ISR2	ISR2_thread with subprogram
System call	Call subprogram

# 建模实例——CCS模型设计

需求分析 > 模型设计 > **建模实例** > 模型验证

## □ CCS(Cruise Control System)操作步骤:

- 驾驶员发出巡航开启指令，系统从空闲转为控制；
- 根据设定巡航速度调节节气门开闭来控制车速；
- 刹车踏板或关闭指令使系统由控制模式转为空闲模式。

表2 CCS建模需求

建模层次	功能描述	建模需求
硬件系统层	数据存取，应用程序运行	Microcontroller, SDRAM, Bus, Device_input, Device_output
应用软件层	车速及节气门开合度计算，数据有效判断，中断处理程序	Compute_velocity & Compute_throttle_setting, Judge_Valid, Device_ISR
系统行为层	数据流传递，端口管理，事件管理，中断管理	Information flow, Port in system, Event in system, Interrupt in system

# 建模实例——CCS模型设计

需求分析 模型设计 建模实例 模型验证

□ CCS模型设计采用架构和行为相结合的方式

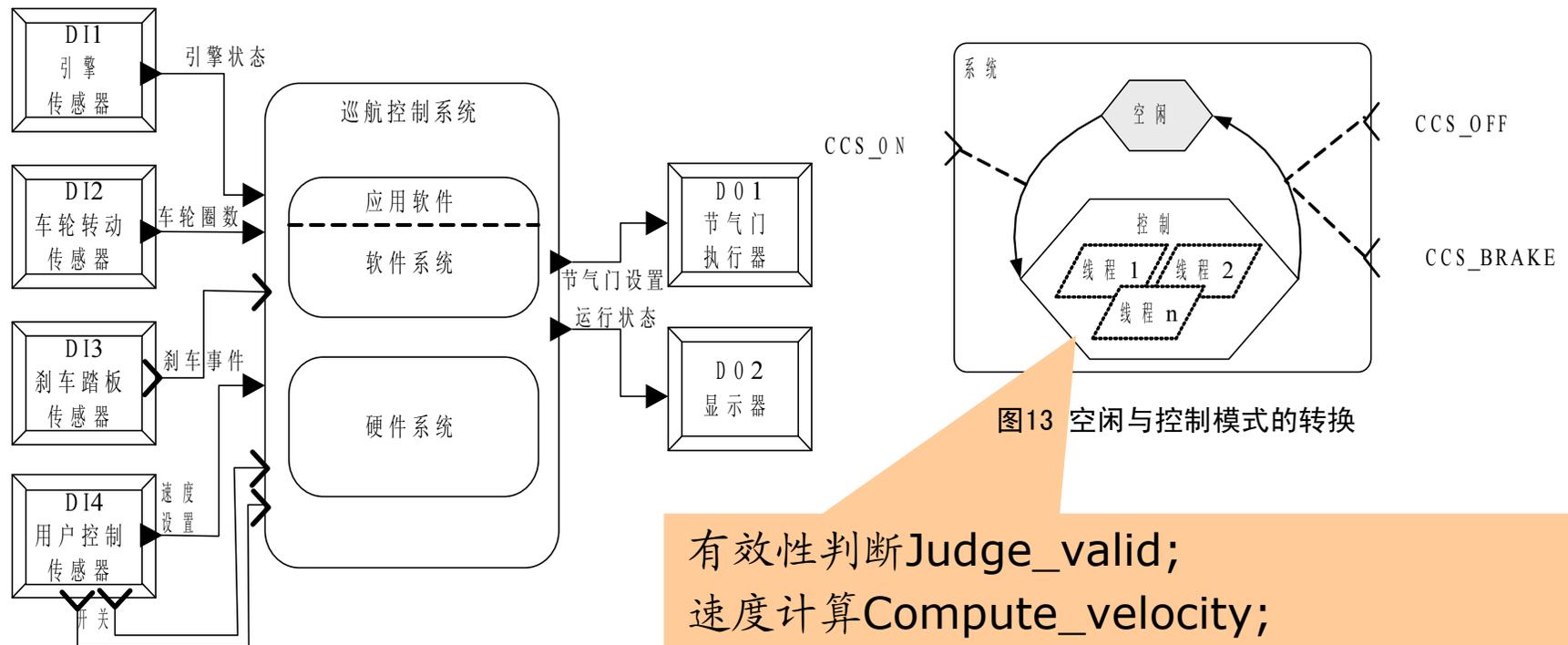


图12 外部设备与系统的信息交换

图13 空闲与控制模式的转换

有效性判断Judge\_valid;  
速度计算Compute\_velocity;  
节气门开合度计算Control\_throttle\_setting

# 模型验证——CCS系统架构

需求分析 > 模型设计 > 建模实例 > 模型验证

## □ AADL的层次结构

- Container
- System
- Process
- Thread

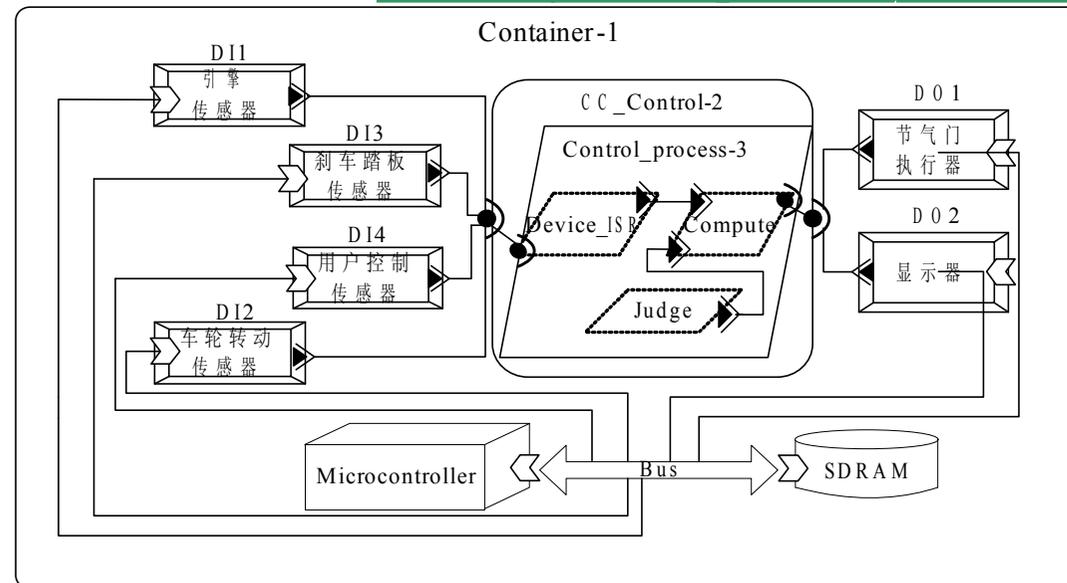


图14 CCS系统层次结构

表3 CCS端口及连接设计

端口名称	端口类型	端口连接描述
Device_ISR_thread.ok_to_run	out event data port	Device_ISR_thread.ok_to_run->Compute_thread.ok_to_run
Compute_thread.ok_to_run	in event data port	Device_ISR_thread.ok_to_run->Compute_thread.ok_to_run
Judge_thread.wheel_pulse	out event data port	Judge_thread.wheel_pulse->Compute_thread.wheel_pulse
Compute_thread.wheel_pulse	in event data port	Judge_thread.wheel_pulse->Compute_thread.wheel_pulse

# 模型验证——CCS端到端的流延迟

需求分析 > 模型设计 > 建模实例 > 模型验证

- 传输延迟
- 期望延迟

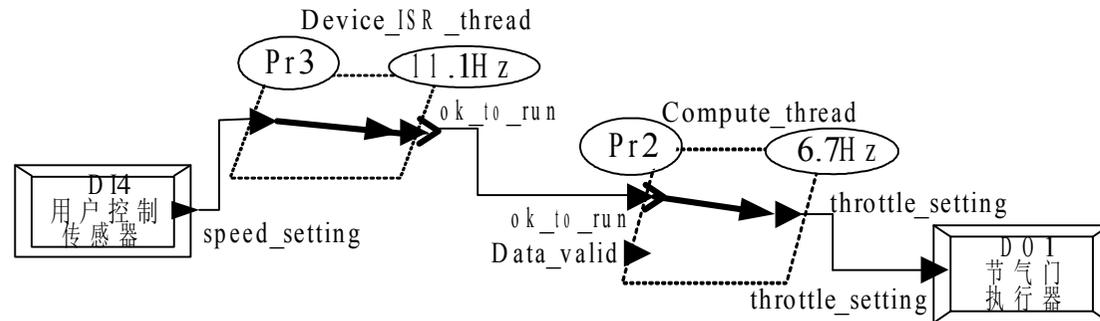


图15 传感器到执行器的端到端延迟验证

表4 端到端的流延迟分析

模型元素	截止期	采样延迟	流延迟描述	总延迟	期望延迟
源端DI4.speed_setting	25 ms	15 ms	15 ms	25 ms	300 ms
speed_setting端口连接	0 ms	0 ms	0 ms	25 ms	300 ms
Device_ISR:FS1	0 ms	0 ms	50 ms	75 ms	300 ms
ok_to_run端口连接	0 ms	0 ms	0 ms	75 ms	300 ms
C_D_S:FS1	0 ms	0 ms	40 ms	115 ms	300 ms
throttle_setting端口连接	0 ms	0 ms	0 ms	115 ms	300 ms
终端DO1:throttle_setting	40 ms	25 ms	20 ms	180 ms	300 ms

# 模型验证——CCS实时调度

需求分析 > 模型设计 > 建模实例 > 模型验证

- 任务调度
- 实时抢占
- 中断处理

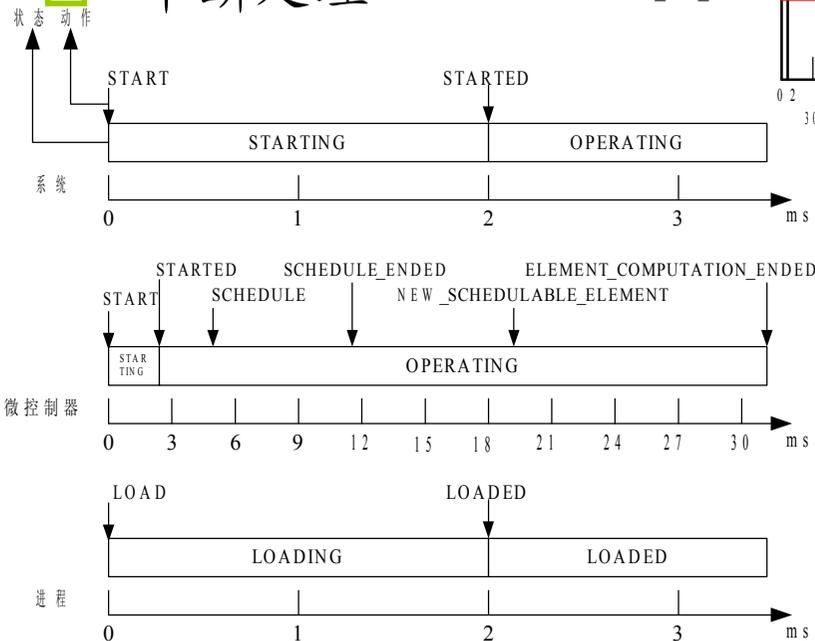


图17 CCS模型组件执行情况

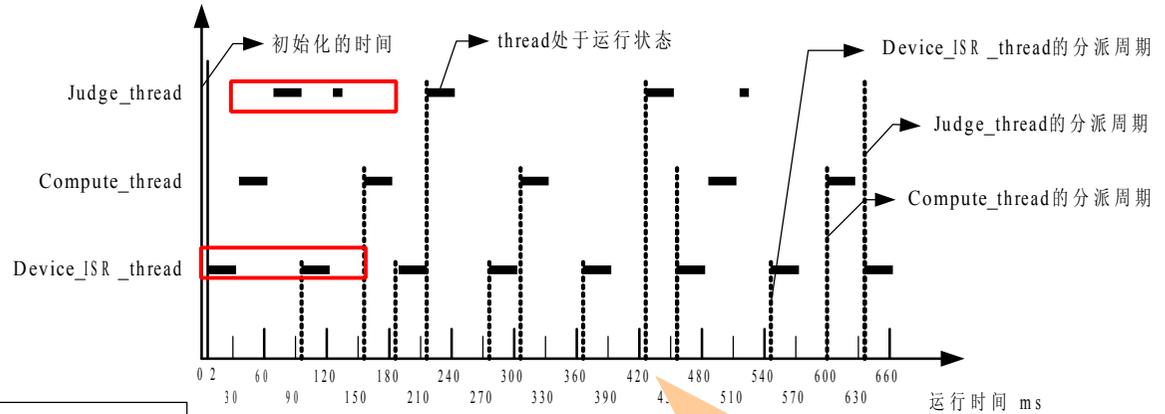
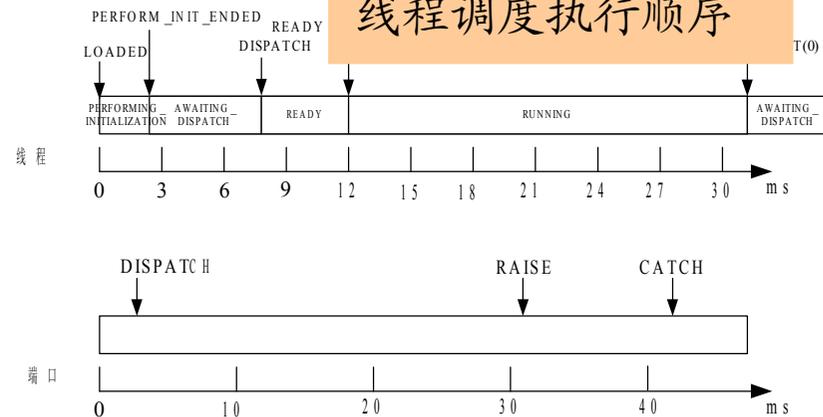


图16 CCS模型调度情况



线程调度执行顺序

系统中各个组件的状态

# 总结与展望

---

## □ 总结

- 将VSP模型设计与车用嵌入式实时系统的特点综合考虑，提出了系统架构和行为抽象级别的建模策略。
- 为基于AADL的车用嵌入式实时系统建模提供了较为全面的设计方法和验证方法。
- 针对汽车巡航控制系统，在OSATE实验平台上分别对系统的架构模型和行为模型进行设计，并进行了架构、端到端流延迟和实时调度验证。

## □ 展望

- 建立符合OSEK COM和OSEK NM的改进模型。
- 利用Error Model Annex建立故障树模型。
- 研究模型转换与自动代码生成过程。

---

# 谢谢!