

嵌入式系统仿真开发环境的体系结构*

雷 剑** 罗克露

(电子科技大学计算机科学与工程学院 成都 610054)

【摘要】对软件体系结构进行了研究,设计了iBus软总线,由总线核和总线适配器组成一个基础结构,作为通信通道连接系统功能部件。总线核嵌入系统主程序、动态创建总线适配器和挂接各个系统功能部件,实现了软件系统集成和即插即用。基于iBus体系结构,提出了一个可扩展和稳定的集成框架,可用于构造嵌入式系统仿真开发环境。

关键词 嵌入式系统; 仿真开发环境; 软总线; 软插件; 软件适配器; 软件体系结构

中图分类号 TP311.5 文献标识码 A

Software Bus Based Architecture of Embedded System Simulation Development Environment

Lei Jian Luo Kelu

(School of Computer Science and Engineering, UEST of China Chengdu 610054)

Abstract The development of embedded application is becoming more and more popular, and simulation development tool is a rapid, effective and low-cost tool for it. A software bus architecture named iBus has been researched and designed based on in-depth investigation of the use of Software Architecture. As a communication channel for the interconnection of system components, iBus is expected to be a firm infrastructure which is composed of bus core and bus adapter. The bus core embeds itself into main program, and then create bus adapters dynamically to connect system components. By this means, iBus can achieve system integration and the function of plug and play in software system. Based on iBus architecture, an integrated framework which is extensible and stable has been presented for efficient construction of the embedded system simulation development environment.

Key words embedded system; simulation development environment; software bus; software component; software adapter; software architecture

目前,嵌入式系统开发已成为计算机工业的热点之一,嵌入式系统应用渗透到信息家电、工业控制、通信与电子设备、人工智能设备等领域,然而嵌入式系统的软件与目标硬件紧密相关,软件的开发与硬件环境的选择、设计和配置相互影响,硬件平台和嵌入式应用的复杂性不断提高,导致软件开发周期长、开发成本昂贵、软件功能调试和性能测试不能及时完成,软件质量无法保障^[1,2]。仿真开发平台是嵌入式系统开发重要的支持工具,如何支持各种硬件厂商提供的种类繁多的硬件平台,满足各类用户不同的软件需求,与现有成熟的辅助开发工具和软件集成以充分利用其提供的功能,一直是仿真开发的难题和迫切需要解决的问题。本文从软件体系结构的角度,研究和设计了一种称为iBus的软总线结构,并提出嵌入式软件仿真开发环境的集成框架。

2003年6月9日收稿

* 广东省科技攻关项目,编号:20021591

** 男 28岁 硕士生 主要从事嵌入式实时系统应用与开发方面的研究

1 iBus体系结构

1.1 软总线加软插件

对于大规模复杂软件系统，其总体结构设计远比算法和数据结构的选择更重要^[3-5]，本文则在对系统体系结构深入研究的基础上，设计了iBus软总线体系结构，软件的结构建立在软总线加软插件思路。软总线是一种相对于硬件总线的概念，将所有的系统功能部件以相同的方式连结在一个用来相互通信的结构性部件上。系统功能部件作为软插件，挂接到总线上，部件之间的交互通过总线的控制和管理来完成^[6]。系统部件如硬件插件，可以随意添加和删减，极大地增强了系统的灵活性、可靠性和扩展性。

1.2 iBus的结构模型

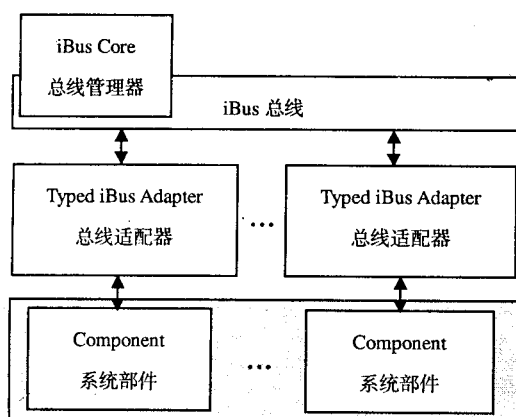


图1 iBus 结构模型

iBus的结构模型如图1所示。

1) 形式化描述如下：

$System = (iBus, Components, Connections)$

$iBus = \{iBus\ Core, iBus\ Adapter_1, \dots, iBus\ Adapter_n\}$

$Components = \{Syscomponent_1, \dots, Syscomponent_n\}$

$Connections = \{ConnectionCategory1, ConnectionCategory2\}$

$ConnectionCategory1 = \{ \langle Core, Adapter_i \rangle | 1 \leq i \leq n \}$

$ConnectionCategory2 = \{ \langle Adapter_i, Syscomponent_i \rangle | 1 \leq i \leq n \}$

2) 基本特征如下：

(1) 软件总线iBus上挂接软件适配器iBus Adapter，系统功能部件Component并不直接连接到总线上，而是连接适配器。

iBus Core和iBus Adapter都运行在本地环境，保证了高效的本地

连接和通信，通过Adapter透明地支持本地或远程的系统功能部件；

(2) iBus Core作为总线管理器，控制和管理总线；

(3) 适配器仅与总线通信，相互之间不通信^[7]；

(4) 总线与所有的适配器建立通信；

(5) 适配器具备类型属性Typed，标识所能提供的服务；

(6) 使用消息机制实现异步通信，使用通知方式实现同步通信；

(7) 消息传递和通知的发布采用相同的机制实现，即发布/订阅^[8]。系统部件通过适配器挂接总线，向总线登记其感兴趣的消息和通知，部件可以根据需要向总线发出消息和通知，总线根据登记表进行分派，接收部件根据自身状态作出响应，如果有处理结果，同步通信立即返回，异步方式下仍然以消息方式返回，根据实际应用需求选择进行异步或同步通信；

(8) 拥有适配器动态创建和删除两级动态能力，以支持软件插拔和系统扩展，连接到适配器上的部件动态创建和替换。

1.3 iBus Core模型

iBus Core是iBus的核心结构部件，作为iBus的总线管理器，其结构模型如图2所示。iBus Core的主要功能如下：1) 动态建立和删除适配器，根据请求的参数，创建相应类型的适配器，由适配器连接和启动具体的功能部件，若创建成功，iBus Core向系统宣告有新的适配器加入，且可暂停一个适配器的使用，从而禁用该功能部件，或者从总线上直接删除此适配器；2) 接受来自适配器的消息/通知订阅，总线管理器提供登记服务，使用订阅表来记录各适配器所关心的消息/通知，以此作为消息和通知分派的依据；3) 存储-转发来自适配器的消息，总线以消息循环的方式，通过检查消息登记表，将接收到的消息分派给相应的适配器；4) 提供通知响应服务，通过检索通知登记表，向相应适配器发布通知。

1.4 iBus Adapter模型

iBus Adapter适配器连接系统功能部件与总线，其结构模型如图3所示。iBus Adapter功能如下：1) 订阅消息和通知，根据挂接的功能部件的要求，向总线订阅消息和通知；2) 接收消息并处理消息，以消息循

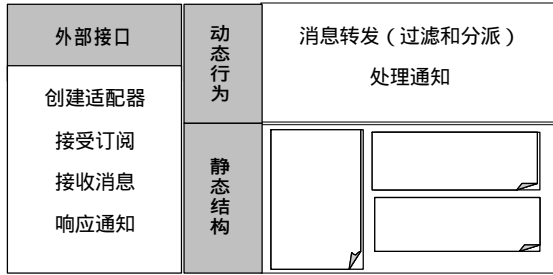


图 2 iBus Core 结构模型

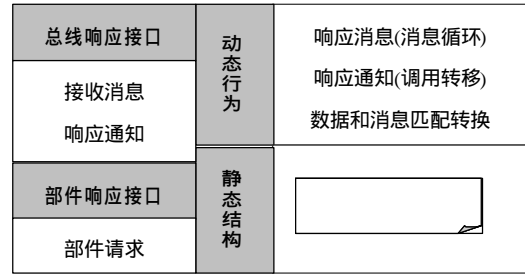


图 3 iBus Adapter 结构模型

环方式处理消息,在消息处理中使用部件提供的服务功能响应消息,提供服务或获取其他部件的服务;3) 响应通知,使用部件的功能提供相应的服务;4) 数据表示的格式转换,保证数据以统一的格式在部件与总线之间交换,保证了部件与部件之间数据交换,对于选择第三方的软件组件作为系统功能部件,只需关注其外部行为;5) 完成系统部件接口与总线接口的匹配,连接方式遵从总线->适配器->系统部件的模式,适配器充当数据和协议转换器;6) 完成消息识别,解释部件请求,转换为对应的系统标准消息并发送;7) 充当远程部件的本地代理^[8],透明支持异构环境。

1.5 基本工作原理

iBus的基本工作原理如图4所示,iBus工作过程如下:1) 适配器A发送消息*m*,送往总线;2) 消息*m*进入总线管理器的全局消息队列;3) 总线管理器检测到新消息*m*,调度程序根据消息登记表,检查到适配器B对消息*m*的订阅,将消息*m*送往B;4) 消息*m*进入适配器B的局部消息队列;5) 适配器B接收消息*m*,进行响应处理。

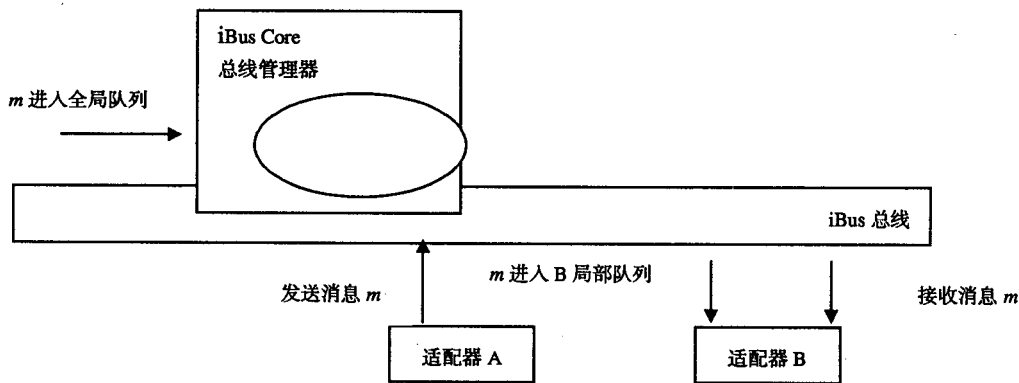


图4 iBus 基本工作原理

iBus具有如下特征:1) 总线和适配器运行于本地,具备较高的性能和通信保证;2) 通过适配器支持系统功能部件的分布和并行运行;3) 系统部件只通过适配器与总线通信,部件之间没有直接通信,保证了系统部件物理位置的透明;4) 系统部件可以动态迁移,可以灵活采用多种方式处理对系统产生的影响;5) 必须预定义系统标准消息,对于重用第三方组件必须进行消息转换,并需要组件提供者的支持。

1.6 结构验证

一个系统是由不同的部件连接构成,实现这种连接需要机制与协议两方面的支持^[4]。通过考察iBus结构对两方面提供的支持,可以证明其完整性和正确性:1) iBus的 Core和Adapter构成连接实现的结构基础,两者的通信连接保证了系统连接的实现;2) iBus定义的Adapter和Core本身是一致的,通过定义标准的消息通知和服务接口,Core与各个Adapter通信,而Adapter之间无直接通信,连接得到了良好的控制和协调。Core即是连接的实现实体,又充当了连接实现的保证机构。经过Adapter的匹配,系统部件与Core完成实际上一致,从而构建的整个系统一致。

2 嵌入式系统仿真开发环境集成框架

2.1 仿真开发环境集成模型

嵌入式仿真开发环境是一个复杂的大型系统，体现在以下方面：1) 需要实现许多功能，如硬件平台配置、代码编辑、编译、程序运行与调试，性能分析与优化，项目管理等，故应尽可能重用现存的软件或软件组件以降低成本，缩短周期；2) 使用cots产品可以保证系统性能，但这些软件因使用不同的编程语言，且运行在不同的平台上，从而导致系统的异构性；3) 系统部件在网络环境中实现和协同执行时导致系统的分布性。因此，整个环境的设计核心是开发或重用软件组件，实现软件部件互操作，集成为一个定义良好、相互协作的系统，可以充分利用iBus体系结构加以解决，仿真开发环境集成模型如图5所示。仿真环境模块如下：1) IDE和项目管理器是开发环境主程序，运行时首先加载和实例化iBus Core，形成通信基础，通过丰富的菜单、快捷按钮等元素形成以项目管理器为核心的集成GUI，IDE提供用户操作接口，访问和使用系统各部件，项目管理器完成项目信息的创建、参数配置和编辑保存，管理项目活动；2) 编辑器提供源代码开发和编辑功能，项目管理器内置或使用外部编辑器；3) 编译器提供对项目源代码的编译生成功能，生成目标硬件平台的可执行代码；4) 调试器提供源码级和汇编级代码调试功能，通过调试器可以检查程序的功能和逻辑正确性，提供多窗口调试信息显示，支持发送调试命令，控制程序的运行方式；5) 软件逻辑分析仪模拟硬件逻辑分析仪的工作原理，采集、显示仿真运行环境中的仿真构件的引脚信号，实现对程序运行状况的分析；6) 仿真环境管理器由仿真构件库、仿真配置器和仿真运行器三部分组成，其内部系统结构可以采用相同的总线结构或其他的体系结构实现，仿真构件库管理系统提供各种仿真硬件，包括仿真MCU、仿真IC和仿真外设，可以增加和删除这些仿真的硬件部件，遵循开发规范，仿真构件可以由第三方开发和提供，仿真配置器提供用户选择硬件和装配目标硬件板的功能，可以构建一个“目标软板”，供调试和测试使用，仿真运行器根据配置器提供的目标板，生成一个运行环境，模拟硬件的执行，提供程序运行的仿真环境。

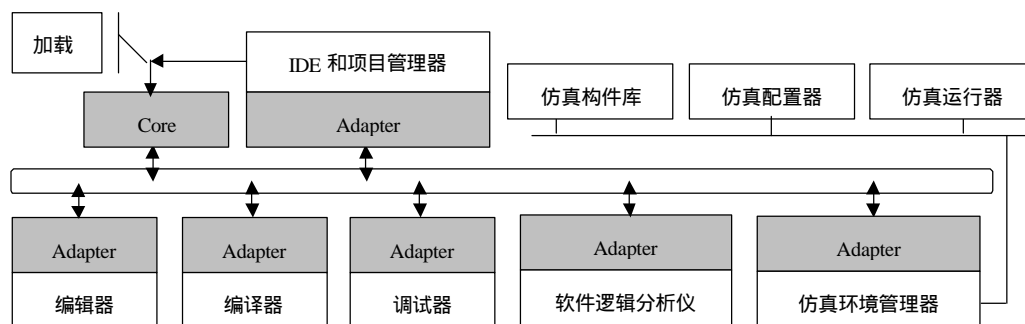


图5 仿真开发环境集成模型

2.2 集成验证

通常认为，一个开发环境考虑三个层次的集成^[9]，通过iBus结构对此三个集成层次的支持，可以验证本体系的正确性：1) 数据集成，iBus结构支持使用与语言无关的数据格式，通过使用适配器，可以转换不同编程语言的数据结构和不同组件识别的数据格式；2) 控制集成，iBus使用消息驱动实现部件异步通信，这是松耦合的互操作，使用通知机制实现部件的同步通信，这是一种紧耦合；3) 界面集成，iBus没有自己的GUI，但通过嵌入到主程序中，并首先加载执行，构造基础运行结构，使用主程序提供的GUI作为界面集成的实体。

(下转第678页)

参 考 文 献

- [1] Wang H, Kuo C C J. Image protection via watermarking on perceptually significant wavelet coefficients[C]. Proceedings of the IEEE Multimedia Signal Processing Workshop, Redondo Beach, California, 1998. 278-284
- [2] Hundur D, Hatzinakos D. A robust digital image watermarking method using wavelet-based fusion[C]. Proceedings of the International Conference on Image Processing, Santa Barbara, California, 1997. 544-547
- [3] Wilson T A, Rogers S K, Myers L R. Perceptual-based hyperspectral image fusion using multiresolution analysis[J]. Optical Engineering, 1995, 34(15): 3 154-3 164
- [4] Xia X, Boncelet C G, Arce G R. Wavelet transform based watermark for digital images[J]. Optics Express, 1998, 3(12): 497-511
- [5] Nill N B. A visual model weighted cosine transform for image compression and quality assessment[J]. IEEE Tanscation on Communication, 1985, 33(6):551-557
- [6] Tan S H, Ngan K N. Classified perceptual coding with adaptive puantization[J]. IEEE Transaction on Circuits and systems for Video Technology, 1996, 37(6): 375-383
- [7] Villasenor J D, Belzer B, Liao J. Wavelet filter evaluation for image compression[J]. IEEE Transactions on Image Processing, 1995, 4(8): 1 053-1 060

编辑 漆 蓉

 (上接第672页)

3 结 束 语

本文对嵌入式系统仿真开发集成环境的构造进行了深入的研究,从软件体系结构的高度设计了iBus软总线结构,提出了一个可扩展和稳定的集成框架,在嵌入式系统开发领域具有广阔的前景。

本文研究工作得到校青年基金(YF020806)的资助,在此表示感谢。

参 考 文 献

- [1] 魏 忠, 蔡 勇, 雷红卫, 等. 嵌入式开发详解[M]. 北京: 电子工业出版社, 2003
- [2] 陈定君, 郭晓东, 张应辉, 等. 嵌入式软件仿真开发系统的研究[J]. 电子学报, 2000, 28(3): 137-139
- [3] Mary S, David G. Software architecture: perspectives on an emerging disciplines[M]. NewJersey: Prentice Hall, 1996
- [4] 万建成, 卢 雷. 软件体系结构的原理、组成与应用[M]. 北京: 科学出版社, 2002
- [5] Purtilo J M. The polyolith software bus[J]. ACM Transaction on Programming Languages and Systems, 1994, 16(1): 151-174
- [6] Dale R. Inside Com[M]. Seattle: Microsoft Press, 1997
- [7] Frank B, Ragine M, Hans R, *et al.* Pattern-oriented software architecture. Volumn1: A System of Patterns[M]. New York: John Wiley&Sons, 1996
- [8] Eric G, Richard H, Ralph J, *et al.* Design patters: elements of reusable object-oriented software[M]. Boston: Addison Wesley Longman, 1995
- [9] Bergstra J A, Klint P. The toolBus-a component interconnection architecture[R]. Programming Research Group, University Of Amesterdam, 1994

编辑 徐培红