

IPMI 在 VPX 系统中的应用与设计

朱红育 李 郁 付学斌

(西安电子工程研究所 西安 710100)

【摘要】本文介绍了智能平台管理接口 IPMI 的基本功能及工作原理,研究了 IPMI 在 VPX 系统管理中的实际应用,并给出了基于低功耗 ARM 处理器的 IPMI 模块实现方法。

关键词:智能平台管理接口;I²C;VPX

中图分类号:TN911.7;TP336 **文献标志码:**A **文章编号:**1008-8652(2013)04-065-05

Application and Design of IPMI in VPX System

Zhu Hongyu, Li Yu, Fu Xuebin

(Xi'an Electronic Engineering Research Institute, Xi'an 710100)

Abstract: The basic function and operation principle of Intelligent Platform Management Interface (IPMI) is introduced. Practical application of IPMI in VPX system management is studied; and the method of implementing IPMI module based on low power consumption ARM processor is presented.

Keywords: Intelligent Platform Management Interface (IPMI); I²C; VPX

0 引言

随着工业控制、信号处理和国防等领域对总线带宽、传输实时性要求的不断提高,传统的 VME、CPCI 等并行总线标准逐渐力不从心,VPX 作为一种基于高速串行总线技术的标准已经悄然兴起。除了引入了目前最新串行总线技术以支持更高的背板带宽外,VPX 还引入了智能平台管理接口(IPMI),用以监控系统的运行状态,提供系统的可靠性。本文对智能平台管理接口 IPMI 的功能做了详细的介绍,给出了智能平台管理接口在 VPX 系统中的实现方法。

1 智能平台管理接口介绍

IPMI(智能平台管理接口)是一种开放标准的硬件管理接口规格,定义了嵌入式管理子系统进行通信的特定方法。用户可以利用 IPMI 监视服务器的物理健康特征,如温度、电压、风扇工作状态、电源

状态等。

自诞生以来智能平台管理接口 IPMI 已经得到了多家供应商的支持,这使得其逐渐成为了一个完整地包括服务器和其他系统(如存储设备、网络和通信设备)的硬件管理规范,在服务器以及基于 ATCA(高级通信计算机架构)的电信设备中的到了广泛的应用^[1]。

基于 IPMI 的智能平台管理系统的主要组件如图 1 所示^[2]。

a. 分布式管理控制器——管理和监控系统中每个 FRU(现场可更换单元)的工作状态和故障状态,包括 ChMC(机框管理器)和 IPMC(板载智能平台管理控制器)。

b. IPMI 其他辅助功能部件——提供在分布式控制单元间以及与系统的总控制单元间的通信、管理和控制功能,如 IPMB(智能平台管理总线)接口和 IPMI 以太网接口。

c. 单板的基于 TCP/IP 的高层管理服务,如:远程启动、SNMP(简单网络管理协议)管理、远程磁盘

服务和远程管理控制协议(RMCP)^[3]。

ChMC 负责系统机框内所有单板的上下电控制、温度和电压监控信息的记录和告警、单板运行状态上报、风冷风扇的转速控制等功能。

IPMC 是 FRU 上的管理控制器,主要职责是管理 FRU(包括上电、复位、电压温度监控等)、收集 FRU 上的关键事件。

IPMC 和 ChMC 之间的数据链路是两条 IPMB (I²C)总线,采用冗余的双总线乒乓传输架构大大提高了该数据链路的可靠性。

FRU 是现场可更换单元的总称,现场可更换的板卡、冷却风机盘等均可称之为 FRU。

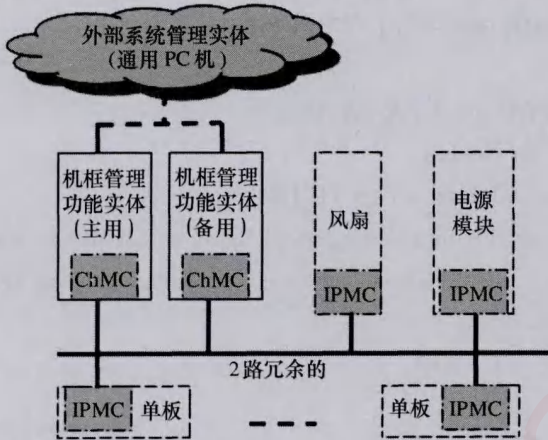


图1 IPMI 智能平台管理系统组成框图

IPMI 基本工作原理如下:

IPMI 一般使用独立的电源供电,且先于系统功能电路上电,板载 IPMC 上电后,先获取单板位置信息(槽位号),然后通过两条 IPMB 总线向 ChMC 发送单板信息及上电请求。

ChMC 在对单板信息进行确认后,向各单板依次发送上电命令。

各单板 IPMC 接收到上电命令后,控制本板负载进行上电,同时开始监控单板电压、温度并通过专用串行接口从 CPU 获取单板工作状态。

当单板电压、温度或工作状态不正常时 IPMC 发送告警到 ChMC。同时响应 ChMC 的查询命令,将本板电压、温度及工作状态等信息上报给 ChMC。

ChMC 可以通过网络接收来自上位机监控系统的查询请求,并向各单板 IPMC 下发温度、电压、工作状态等查询请求,自动向上位机监控系统上报系统告警信息同时记录系统工作日志。

2 IPMI 在 VPX 系统中的应用

VPX 系统在设计时也引入了 IPMI 智能平台管理接口,在 P0 端子上预留了 SM1—SM4 四个单端信号给两路 IPMB 总线使用。

图2 给出了 IPMI 在 VPX 系统中的应用框图。

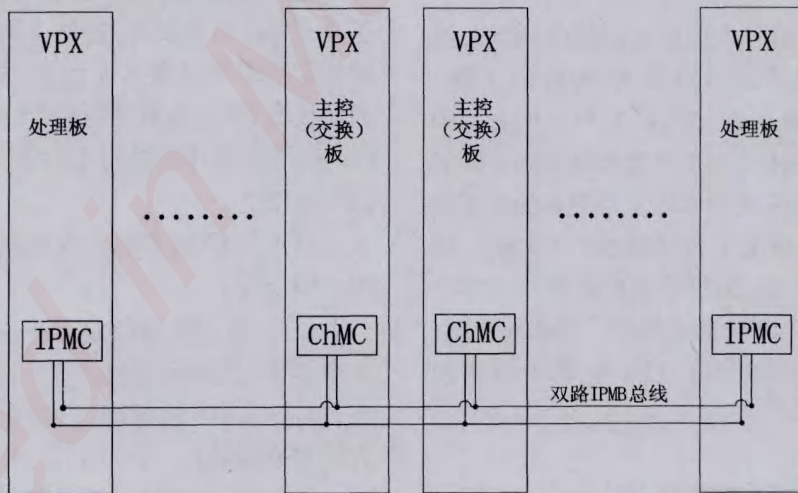


图2 IPMI 在 VPX 系统中的应用框图

考虑到 ChMC 在 IPMI 中的重要性,ChMC 设计多采用 1+1 冗余备份机制以提高 IPMI 系统的可靠性。两个 ChMC 单元上电后通过主备竞争机制确认各自工作状态,同时通过心跳信息相互检测,当发现对端工作异常时向其发送硬复位信号,并自动进行

主备切换,确保 IPMI 系统能够正常工作。

ChMC 在物理上可以是两块独立的板卡,也可以是系统主控交换板卡上的独立电路(主控交换板卡大都数情况下也采取 1+1 冗余备份的方法提高系统可靠性)。

板载 IPMC 是各 VPX 单板上一个独立的电路单元,采用背板 3.3V_{AUX} 进行供电。

2.1 IPMB 总线设计实现

IPMB 总线采用了 I²C 通信技术,定义在 P0 端子的四个单端信号端子 SM1 - SM4 上,鉴于 I²C 总线的固有特点,虽然采用了冗余的乒乓通信方式,总线锁死的现象仍无法避免,因此设计中需要注意如下方面:

a. 总线隔离及热插拔功能实现

依据 IPMI 规范,IPMB 总线具有热插拔功能,因此设计选用 Linear 公司的 LTC4307,该芯片是一颗专用的 I²C 双向隔离缓冲器,提供了上升时间加速器从而降低负载过重的总线上升时间,1V 预充电压以降低总线干扰,还具备总线锁死自动检测和恢复功能^[4]。

b. 总线驱动能力计算

I²C 信号最大的容性负载能力为 400pF,因此必须对 IPMB 的负载能力进行合理的设计。根据容性负载选择合适的上拉电阻,以保证总线的驱动能力^[5]。

对于一个单板来说,容性负载包括以下几个部分:

- 到背板的连接器容性负载:2pF
- 连接器 PCB 通孔容性负载:1pF
- 引线 and 过孔容性负载:9pF
- I²C 驱动器:10pF

此外背板走线的容性负载也需要考虑在内。

在背板上通过计算好的上拉电阻,将 IPMB 总线上拉到 3.3V_{AUX} 电源,单板的插拔不会影响到总线的驱动能力。

c. 总线锁死预判和处理

I²C 总线的锁死根据原因可分为两种情况,即:从设备未响应导致总线锁死或主设备故障导致总线锁死。

在硬件上每个接入 IPMB 总线的节点板卡都设计了 LTC4307 隔离缓冲器,当该器件检测 SCL 或 SDA 信号线保持常低 30ms 以上时,会自动将节点板卡从总线上断开,同时向板卡侧发送 16 个 SCL 时钟以及一个 Stop 信号,从而使总线恢复正常状态。

在软件上,也应当设计相应的监测和预判机制,以保证 IPMB 总线的稳定工作。

IPMC 单元一旦检测到无法从 IPMB 总线上获取到心跳信息,立即控制 LTC4307 将本板卡从

IPMB 总线上断开,记录故障告警信息,再复位 I²C 控制器,重新打开 LTC4307 尝试与 ChMC 进行通信,如果依然失败,将板卡从总线上断开,并点亮告警指示灯提示人工干预。

ChMC 单元监测所有板卡的心跳信息(包括冗余 ChMC),当检测到某一板卡的心跳信息丢失,而其他板卡的心跳信息正常时,确认该板卡拔出或故障,记录并上传告警信息。当检测到所有心跳信息丢失时,确认本板故障,立即控制 LTC4307 将本板从 IPMB 总线上断开,记录故障告警信息,再复位 I²C 控制器,重新打开 LTC4307 尝试与其他板卡进行通信,如果依然失败,将板卡从总线上断开,并点亮告警指示灯提示人工干预。

2.2 机框管理单元 ChMC 设计

IPMI 设计的应依据如下原则:

a. 低功耗——VPX 系统提供给每槽位 IPMI 模块的 3.3V_{AUX} 电源功耗小于 1A;

b. 硬件一体化设计——节省为系统功能部分电路节省 PCB 面积,同时低复杂度的硬件设计降低硬件故障风险;

c. 软件设计简单化以降低软件故障风险,保证 IPMI 部分软件具有极高的可靠性;

依据上述原则,ChMC 硬件电路应满足如下需求:

a. 具备 2 路 I²C 接口连接到两路背板 IPMB 总线。

b. 具备 2 路 UART(通用异步串口)接口,1 路用作和功能单元 CPU 进行通信,以获取板卡运行状态信息,1 路用作调试接口。

c. 具备多通道 ADC 模块,用以对功能电路各供电电源进行监测。

d. 具备 LAN(以太网)通信接口,用以连接远端 PC 接受上位系统管理系统控制。

e. 具备丰富的 GPIO 资源,用以控制功能电路上下电、复位、槽位信息获取、单板温度监控等功能。

f. 具备看门狗电路,以增加系统软件的可靠性。

TI 公司的 ARM Cortex™ - M3 处理器 LM3S6918,单芯片即可满足上述功能,该处理器运行速率最高可达 50M,集成了 RTC、看门狗功能,对外接口十分丰富,具备 2 路 I²C 接口,2 路 UART 接口,集成 10/100 以太网 MAC 和 PHY,最多可达 38 个 GPIO,集成 8 通道 12 位 ADC,典型功耗仅为 100 多毫安。

ChMC 的具体实现框图如图 3 所示。

由于 ChMC 负责整个机框的管理,所以 ChMC 应是整个系统中可靠性最高的组件之一。除了在软硬件设计上注重其可靠性设计外,大多数设计中均

采用主备冗余设计方案来进一步提高该单元的可靠性。而竞争电路是主备冗余设计中的关键,下面给出一种 ChMC 主备竞争电路的实现方法(见图 4)。

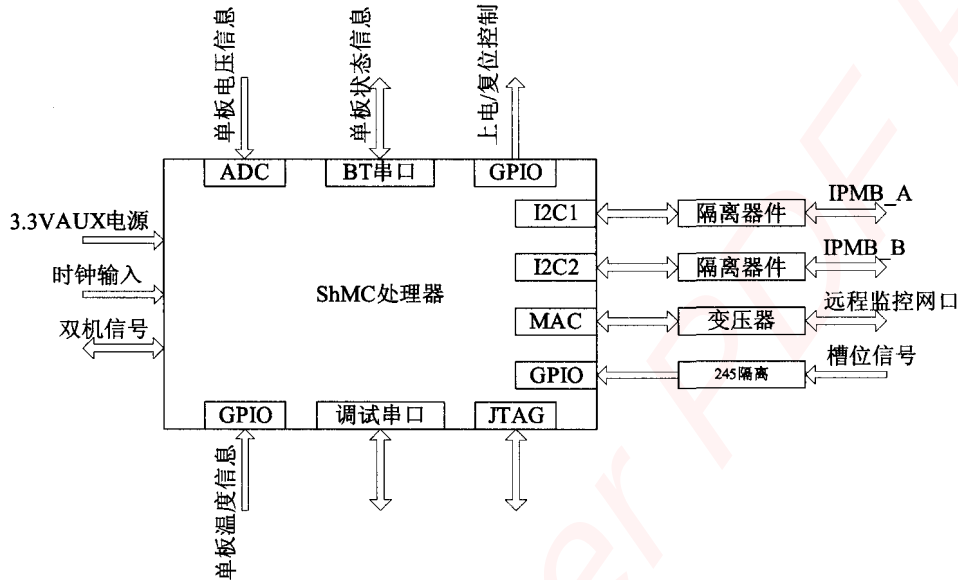


图 3 机框管理单元 ChMC 实现框图

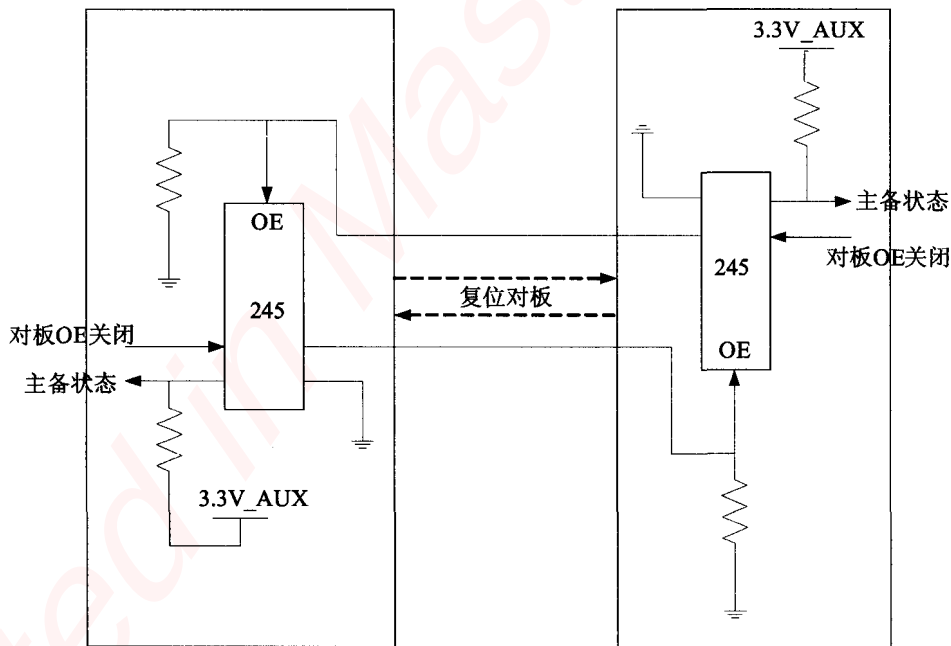


图 4 ChMC 主备竞争电路

具体工作原理如下:

ChMC 上电工作后,先获取槽位号,接着发出对板 OE 关闭信号,然后获取本板主备状态信息。

当两个单板先后(插入机框)上电时,后上电单板上的 245 使能信号 OE 会被先上电的单板关闭,

同时它发出的对板 OE 关闭信号被本板 245 隔离掉了,因此后上电的单板读取的主备状态为“1”,即备用状态。先上电的单板 245 使能端默认有效,读取的主备状态为“0”,即主用状态。

当两个单板已经在机框内并同时上电时,系统

设计默认某一槽位为主用 ChMC, 软件设计时将根据槽位号在输出对板 OE 关闭信号时做不同的延迟, 默认备用的 ChMC 在发出对板 OE 关闭信号时会延迟若干微秒, 以保证主用 ChMC 先工作。

IPMI 正常工作时, 主备 ChMC 定期互发心跳消息, 同时监控各单板 IPMC 发出的心跳消息, 当备用 ChMC 发现无法接收到主用 ChMC 心跳消息, 而又能监控到各 IPMC 心跳消息时, 即可确认主用 ChMC 故障, 发出主用 ChMC 复位信号, 将其复位, 主用 ChMC 复位后会释放对板 OE 关闭信号, 同时会接收到备用 ChMC 发出的 OE 关闭信号, 备用 ChMC 轮训到主备状态变为“0”时, 切换至主用状态, 原主用 ChMC 则会检测到主备状态指示变为“1”, 并据此切换本板至备用状态。

ChMC 复位过程中会自动通过 I²C 隔离器件将自己从 IPMB 总线上隔离, 复位完成后再接入 IPMB 总线。

主备竞争的信号在 VPX 连接器中用户自定义的单端信号上实现。

2.3 板载 IPMC 设计

板载 IPMC 和上述 ChMC 相比较, 最大的区别在于 IPMI 管理层软件的设计, 二者的硬件设计基本相似, 具体可参见 ChMC 单元进行实现。

板载 IPMC 除了监控本板电压、温度信息外, 还需要获取本板功能电路的运行状态, 将功能电路的故障机告警信息上报到 ChMC。功能电路状态的监控一般由本板 CPU 进行, 通过专用串行串口将监控信息上报给 IPMC^[6]。当 ChMC 作为一个硬件单元集成在某一功能板卡上时, 也实现上述功能。从可靠性和成本的角度出发, 大多数设计中均依据 IPMI

规范简化设计, 因此 IPMC 在设计上不考虑以太网通信的实现, 也没有主备冗余设计。

3 结束语

本文针对 VPX 系统管理的具体应用, 介绍了 IPMI 的一些基本功能, 着重描述了 IPMI 在 VPX 系统中的实现示例, 该方案以低功耗 ARM 处理器为核心, 即可以用作 ChMC 电路的设计也用作板载 IPMC 的设计, 仅需要在软件上做相应的改动即可, 此外还引入了 ChMC 1+1 冗余设计的概念, 具有良好的灵活性和很高的可靠性。

参考文献:

- [1] 谢勇祥, 曲道奎. ATCA 之机箱管理及高可靠性设计与实现[J]. 微计算机信息, 2007, 23(2-2): 42-43.
- [2] Intelligent Platform Management Interface Specification, V2. 0[S]. Intel, HP, NEC, Dell, 2005.
- [3] 李威, 尚学群, 陈建全. ATCA 管理模块模拟器的设计与实现[J]. 科学技术与工程, 2010, 10(5): 1271-1274.
- [4] 李彦英, 邹少云. PCA951X 在 ATCA 架构上的应用[J]. 集成电路应用, 2005: 70-71.
- [5] 赵辉, 董德存. I2C 总线技术及其应用实例[J]. 微型电脑应用, 2005, 21(4): 61-63.
- [6] 李彦, 陈德人, 梁达明. 基于 IPMI 的虚拟远程串口终端[J]. 现代机械, 2006, (3): 35-37.

嵌入式资源免费下载

总线协议:

1. [基于 PCIe 驱动程序的数据传输卡 DMA 传输](#)
2. [基于 PCIe 总线协议的设备驱动开发](#)
3. [CANopen 协议介绍](#)
4. [基于 PXI 总线 RS422 数据通信卡 WDM 驱动程序设计](#)
5. [FPGA 实现 PCIe 总线 DMA 设计](#)
6. [PCI Express 协议实现与验证](#)
7. [VPX 总线技术及其实现](#)
8. [基于 Xilinx FPGA 的 PCIE 接口实现](#)
9. [基于 PCI 总线的 GPS 授时卡设计](#)
10. [基于 CPCI 标准的 6U 信号处理平台的设计](#)
11. [USB30 电路保护](#)
12. [USB30 协议分析与框架设计](#)
13. [USB 30 中的 CRC 校验原理及实现](#)
14. [基于 CPLD 的 UART 设计](#)

VxWorks:

1. [基于 VxWorks 的多任务程序设计](#)
2. [基于 VxWorks 的数据采集存储装置设计](#)
3. [Flash 文件系统分析及其在 VxWorks 中的实现](#)
4. [VxWorks 多任务编程中的异常研究](#)
5. [VxWorks 应用技巧两例](#)
6. [一种基于 VxWorks 的飞行仿真实时管理系统](#)
7. [在 VxWorks 系统中使用 TrueType 字库](#)
8. [基于 FreeType 的 VxWorks 中文显示方案](#)
9. [基于 Tilcon 的 VxWorks 简单动画开发](#)
10. [基于 Tilcon 的某武器显控系统界面设计](#)
11. [基于 Tilcon 的综合导航信息处理装置界面设计](#)
12. [VxWorks 的内存配置和管理](#)
13. [基于 VxWorks 系统的 PCI 配置与应用](#)
14. [基于 MPC8270 的 VxWorks BSP 的移植](#)

Linux:

1. [Linux 程序设计第三版及源代码](#)
2. [NAND FLASH 文件系统的设计与实现](#)
3. [多通道串行通信设备的 Linux 驱动程序实现](#)
4. [Zsh 开发指南-数组](#)
5. [常用 GDB 命令中文速览](#)
6. [嵌入式 C 进阶之道](#)
7. [Linux 串口编程实例](#)

Windows CE:

1. [Windows CE.NET 下 YAFFS 文件系统 NAND Flash 驱动程序设计](#)
2. [Windows CE 的 CAN 总线驱动程序设计](#)
3. [基于 Windows CE.NET 的 ADC 驱动程序实现与应用的研究](#)
4. [基于 Windows CE.NET 平台的串行通信实现](#)
5. [基于 Windows CE.NET 下的 GPRS 模块的研究与开发](#)
6. [win2k 下 NTFS 分区用 ntldr 加载进 dos 源代码](#)
7. [Windows 下的 USB 设备驱动程序开发](#)
8. [WinCE 的大容量程控数据传输解决方案设计](#)
9. [WinCE6.0 安装开发详解](#)
10. [DOS 下仿 Windows 的自带计算器程序 C 源码](#)
11. [G726 局域网语音通话程序和源代码](#)
12. [WinCE 主板加载第三方驱动程序的方法](#)
13. [WinCE 下的注册表编辑程序和源代码](#)
14. [WinCE 串口通信源代码](#)
15. [WINCE 的 SD 卡程序\[可实现读写的源码\]](#)

PowerPC:

1. [Freescale MPC8536 开发板原理图](#)
2. [基于 MPC8548E 的固件设计](#)
3. [基于 MPC8548E 的嵌入式数据处理系统设计](#)

RT Embedded <http://www.kontronn.com>

4. [基于 PowerPC 嵌入式网络通信平台的实现](#)
5. [PowerPC 在车辆显控系统中的应用](#)
6. [基于 PowerPC 的单板计算机的设计](#)

ARM:

1. [基于 DiskOnChip 2000 的驱动程序设计及应用](#)
2. [基于 ARM 体系的 PC-104 总线设计](#)
3. [基于 ARM 的嵌入式系统中断处理机制研究](#)
4. [设计 ARM 的中断处理](#)
5. [基于 ARM 的数据采集系统并行总线的驱动设计](#)
6. [S3C2410 下的 TFT LCD 驱动源码](#)
7. [STM32 SD 卡移植 FATFS 文件系统源码](#)
8. [STM32 ADC 多通道源码](#)
9. [ARM Linux 在 EP7312 上的移植](#)

Hardware:

1. [DSP 电源的典型设计](#)
2. [高频脉冲电源设计](#)
3. [电源的综合保护设计](#)
4. [任意波形电源的设计](#)
5. [高速 PCB 信号完整性分析及应用](#)
6. [DM642 高速图像采集系统的电磁干扰设计](#)