



(12) 发明专利

(10) 授权公告号 CN 103176930 B

(45) 授权公告日 2015. 08. 12

(21) 申请号 201310113269. 6

(22) 申请日 2013. 04. 02

(73) 专利权人 无锡江南计算技术研究所
地址 214083 江苏省无锡市滨湖区军东新村
030 号

(72) 发明人 吴新军 丁琳 韩娇 罗茂盛
卢姝颖 吴志勇 欧阳伟

(74) 专利代理机构 北京众合诚成知识产权代理
有限公司 11246

代理人 龚燮英

(51) Int. Cl.

G06F 13/20(2006. 01)

(56) 对比文件

CN 101604301 A, 2009. 12. 16, 全文.

CN 2023835669 U, 2012. 08. 15, 全文.

CN 202421950 U, 2012. 09. 05, 全文.

CN 201673382 U, 2010. 12. 15, 全文.

陈虎成等. 一种总线扩展的实现方法. 《系统工程与电子技术》. 2000, 第 22 卷 (第 12 期),

第 33 - 35 页.

邱力为等. PCI 总线扩展技术分析与系统设计. 《总线扩展技术分析与系统设计》. 2002, 第 123 - 134 页.

宋克柱等. PCI-to-PCI 桥及其应用设计. 《电子技术》. 2002, 第 133 - 135 页.

赵锡凯等. PCI 总线扩展技术研究. 《计算机工程与科学》. 1997, 第 19 卷 (第 2 期), 第 12 - 15 页.

审查员 彭明明

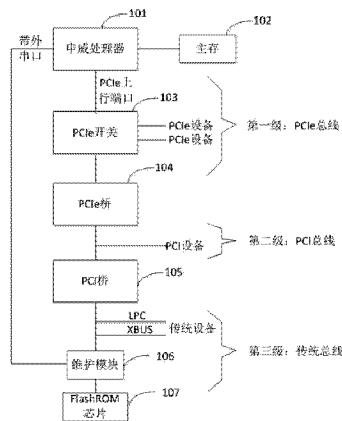
权利要求书1页 说明书6页 附图4页

(54) 发明名称

一种基于标准 PCIe 上行端口的 IO 扩展架构方法

(57) 摘要

本发明提供了一种基于标准 PCIe 上行端口的 IO 扩展架构方法。使用标准 PCIe 上行端口的分层多级总线扩展,第一级为 PCIe 总线,第二级为 PCI 总线,第三级为传统总线,其中 BIOS 挂在传统总线下;通过带外的同步串行通路注入申威处理器所需的初始配置和初始执行代码,初始执行代码只需一次深度优先的枚举,就完成了传统设备的访问通路构建,使得申威处理器可以快速获取 BIOS 内容;通过带外的同步串行通路,可向 CPU 发出不可屏蔽中断,用于睡眠状态下的唤醒等目的。



1. 一种用于申威处理器的基于标准 PCIe 上行端口的 I/O 扩展架构方法, 其特征在于包括:

进行分层多级总线扩展, 其中多级总线的根在于申威处理器的标准 PCIe 接口, 处理器提供带外的串口, 按照总线带宽的高低确定总线离处理器的远近, 将 BIOS 芯片挂在最远的地方;

执行申威处理器硬件的初始配置、并初始执行代码, 通过带外串口将申威处理器所需的初始配置和初始执行代码注入到申威处理器的内部寄存器和指令高速缓存中, 完成处理器的启动配置, 使处理器执行启动代码;

通过申威处理器的初始执行代码完成存控的训练和主存的初始化, 并通过一次 PCI 设备深度优先的枚举操作, 完成申威处理器到 BIOS 访问通路的构建。

2. 根据权利要求 1 所述的基于标准 PCIe 上行端口的 I/O 扩展架构方法, 其特征在于还包括: 将传统设备挂接在离处理器最远的总线上, 通过各级总线对优先级的配置进行仲裁。

3. 根据权利要求 1 或 2 所述的基于标准 PCIe 上行端口的 I/O 扩展架构方法, 其特征在于, 初始执行代码内含 PCI 设备枚举功能, 用于将传统 I/O 空间分配给首个下一级总线。

4. 根据权利要求 1 或 2 所述的基于标准 PCIe 上行端口的 I/O 扩展架构方法, 其特征在于, 带外串口采用同步的物理层, 且时钟和数据的相位关系在系统端可配, 链路层以包的形式传输, 请求和响应之间没有严格的时序要求。

5. 根据权利要求 4 所述的基于标准 PCIe 上行端口的 I/O 扩展架构方法, 其特征在于, 链路层的包的传输是连续, 不可暂停, 以命令码和响应码标识包的开始和长度。

6. 根据权利要求 1 或 2 所述的基于标准 PCIe 上行端口的 I/O 扩展架构方法, 其特征在于还包括: 使中断使用一个命令码和包格式, 以包的方式传递给处理器。

7. 根据权利要求 1 或 2 所述的基于标准 PCIe 上行端口的 I/O 扩展架构方法, 其特征在于还包括: 增强或弱化其中的某级总线挂接设备的能力。

一种基于标准 PCIe 上行端口的 IO 扩展架构方法

技术领域

[0001] 本发明涉及涉及计算机和通信领域,具体地说,本发明涉及一种基于标准 PCIe 上行端口的 IO 扩展架构方法。

背景技术

[0002] 目前,基于 X86 处理器的系统,使用专有的系统总线,如 Intel 的 QPI 总线和 AMD 的 HT 总线,传统设备挂接在系统总线延伸出的 DMI 或者 A-Link 上,PCIe/PCI 桥接电路也挂在系统总线上,逻辑关系如图 1 所示。

[0003] X86 处理器 A1 一上电就可以访问传统设备(包括 BIOS) A3,在 BIOS 执行过程中通过 PCIe/PCI 桥 A2 配置 PCIe/PCI 通路和设备。

[0004] 但是,申威处理器没有使用国外公司专有的系统总线,但要利用市场上众多的外部设备,就集成了标准的 PCIe 根联合体,没有直接挂接传统设备的接口。申威处理器无法做到一上电就访问传统设备。

发明内容

[0005] 本发明解决的问题是基于申威处理器的系统的 I/O 扩展问题和处理器上电启动问题。

[0006] 为了解决上述问题,根据本发明,提供了一种用于申威处理器的基于标准 PCIe 上行端口的 IO 扩展架构方法,其包括:进行分层多级总线扩展,其中多级总线的根在于申威处理器的标准 PCIe 接口,处理器提供带外的串口,按照总线带宽的高低确定总线离处理器的远近,将 BIOS 芯片挂在最远的地方;执行申威处理器硬件的初始配置、并初始执行代码,通过带外串口注入到申威处理器的内部寄存器和指令高速缓存中,完成处理器的启动配置,使处理器执行启动代码;通过申威处理器的初始执行代码完成存控的训练和主存的初始化,并通过一次 PCI 设备深度优先的枚举操作,完成申威处理器到 BIOS 访问通路的构建,由此可以实现快速执行 BIOS 代码。

[0007] 带外串口基于共同时钟,数据收发速率高达 25Mbps,系统端的数据收发可以选择在时钟的上升沿或者下降沿,一对一的双向传输方式;在深度睡眠状态下,PCIe 上行端口停止工作,用于唤醒的中断通过带外串口传递到申威处理器的中断控制器,使处理器推出深度睡眠状态。

[0008] 优选地,将传统设备挂接在离处理器最远的总线上,通过各级总线仲裁优先级的配置。从而,保证各级设备在带宽、延迟等方面的服务质量要求。

[0009] 优选地,初始执行代码内含 PCI 设备枚举功能,用于将传统 I/O 空间分配给首个下一级总线。

[0010] 优选地,带外串口采用同步的物理层,且时钟和数据的相位关系在系统端可配,链路层以包的形式传输,请求和响应之间没有严格的时序要求。

[0011] 优选地,链路层的包的传输是连续,不可暂停,以特定的命令码和响应码标识包的

开始和长度。

[0012] 优选地,中断使用一个特定的命令码和包格式,以包的方式传递给处理器。

[0013] 优选地,根据实际系统的需要,增强或弱化其中的某级总线挂接设备的能力。

附图说明

[0014] 结合附图,并通过参考下面的详细描述,将会更容易地对本发明有更完整的理解并且更容易地理解其伴随的优点和特征,其中:

[0015] 图 1 示意性地示出了传统设备的逻辑关系。

[0016] 图 2 是本发明 I/O 扩展架构方法的一种实施方式示意图;

[0017] 图 3 是本发明 I/O 扩展架构方法的一种实施例示意图;

[0018] 图 4 是本发明 I/O 扩展架构方法的另一种实施例示意图;

[0019] 图 5 是本发明系统启动方法一种实施方式示意图;

[0020] 图 6 是本发明系统唤醒方法的一种实施方式示意图。

[0021] 需要说明的是,附图用于说明本发明,而非限制本发明。注意,表示结构的附图可能并非按比例绘制。并且,附图中,相同或者类似的元件标有相同或者类似的标号。

具体实施方式

[0022] 为了使本发明的内容更加清楚和易懂,下面结合具体实施例和附图对本发明的内容进行详细描述。

[0023] 为解决现有技术的上述问题,在本发明提供的用于申威系统的 I/O 扩展架构方法中,使用标准 PCIe 上行端口的分层多级总线扩展,第一级为 PCIe 总线,第二级为 PCI 总线,第三级为传统总线,其中 BIOS 挂在传统总线下;通过带外的同步串行通路注入申威处理器所需的初始配置和初始执行代码,初始执行代码只需一次深度优先的枚举,就完成了传统设备的访问通路构建,使得申威处理器可以快速获取 BIOS 内容;通过带外的同步串行通路,可向 CPU 发出不可屏蔽中断,用于睡眠状态下的唤醒等目的。

[0024] 其中,在本发明中,术语“传统设备”指的是计算机中除处理器之外的其它设备。

[0025] 更进一步地说,为解决现有技术的上述问题,本发明提供一种基于标准 PCIe 上行端口的 I/O 扩展架构方法,包括:

[0026] 进行分层多级总线扩展,其中多级总线的根在于申威处理器的标准 PCIe 接口,处理器还提供带外串口,按照总线带宽的高低确定总线离处理器的远近(即,总线带宽越高,离处理器越近),传统设备(包括 BIOS 芯片)挂接在离处理器最远的总线上,。通过各级总线仲裁优先级的配置,保证各级设备在带宽、延迟等方面的服务质量要求;例如,挂在 PCI 总线上的音频播放器对带宽和延迟的服务质量要求。可以根据实际系统的需要,增强或弱化其中的某级总线挂接设备的能力。

[0027] 执行申威处理器硬件的初始配置、并且初始启动代码,通过带外串口注入到申威处理器的内部寄存器和指令高速缓存中,完成处理器的初始配置,使处理器执行初始启动代码。

[0028] 申威处理器通过执行初始启动代码完成存控的训练和主存的初始化,代码中还内含 PCI 设备枚举功能,通过一次 PCI 设备深度优先的枚举操作,将传统 I/O 空间分配给首个

下一级总线,完成申威处理器到 BIOS 访问通路的构建。申威处理器将 BIOS 代码取到主存的合适位置,然后执行 BIOS 代码。

[0029] 在深度睡眠状态下,PCIe 上行端口停止工作,用于唤醒的中断通过带外串口传递到申威处理器的中断控制器,使处理器推出深度睡眠状态。

[0030] 带外串口信息传输,物理层基于共同时钟,一对一的双工传输方式,数据收发速率高达 25Mbps,系统端的数据收发可以选择在时钟的上升沿或者下降沿,从而调整时钟和数据的相位关系,降低板级的工程要求;链路层以包的形式传输,请求和响应之间没有严格的时序要求。包的传输是连续,不可暂停,以特殊的命令码和响应码标识包的开始和长度。唤醒中断使用一个特殊的命令码和包格式,以包的方式传递给处理器。

[0031] 通过采用上述配置,本发明解决了基于申威处理器的系统的 I/O 扩展问题和处理器上电启动问题。

[0032] 下面将描述本发明的具体实施例,以使得本领域技术人员可以清楚本发明的具体实现方式。

[0033] 图 2 是本发明 I/O 扩展架构方法的一种实施方式示意图。

[0034] 参照图 2 所示,根据本发明一种实施方式的基于标准 PCIe 上行端口的 IO 扩展架构包括:申威处理器 101、主存 102、PCIe 开关 103、PCIe 桥 104、PCI 桥 105、维护模块 106 和 BIOS 芯片 107。

[0035] 其中,申威处理器 101 直接挂接主存 102 以集成存控,并且与 PCIe 开关 103 相连以集成标准 PCIe 接口;此外,申威处理器 101 还可具备低速串口(带外),以便与维护模块 106 相连;

[0036] PCIe 开关 103 的上行端口与申威处理器的 PCIe 接口相连,PCIe 开关 103 可扩展出多个 PCIe 端口,用于连接 PCIe 扩展槽或者 PCIe 设备,每个设备需占用一个 PCIe 端口;

[0037] PCIe 桥 104 的上行端口与 PCIe 开关 103 相连,可扩展出标准 PCI 总线,一条 PCI 总线上挂接 PCI 扩展槽、PCI 设备等多个负载;

[0038] PCI 桥 105 的上行端口与 PCIe 桥相连,可扩展出多种传统总线(例如 LPC(Low Pin Count)总线,XBus (eXtended Bus)总线),用于挂接传统设备,传统设备具有相对小的地址空间,且这些空间的基地址选择范围很小。

[0039] 维护模块 106 的一端与申威处理器的串口(带外)相连,另一端作为传统设备,挂在传统总线上,可以与申威处理器进行带内通信。维护模块外挂 FlashROM 芯片。

[0040] FlashROM 芯片 107 用于存放申威处理器的初始配置和初始启动代码,还可以存放 UEFI BIOS 等信息,实现这些信息的非易失存储。

[0041] 图 3 是本发明 I/O 扩展架构方法的一种实施例的示意图。

[0042] 参照图 3 所示,在本发明 I/O 扩展架构方法的一个实施例中,I/O 扩展架构包括:第二代申威处理器(申威-2 处理器) 201、DDR3 主存 202、8PortPCIe 开关 203、PCIe-to-PCI 桥 204、PCI-to-LPC 桥 205、维护模块 206、FlashROM 芯片 207。

[0043] 其中,申威-2 处理器 201 集成 DDR3 存控,直接挂接 DDR3 主存 202;集成标准 PCIe Gen2X8 接口,与 8-Port PCIe Gen2 开关 203 相连;而且申威-2 处理器 201 具备低速串口(带外),与维护模块相连;

[0044] 8-Port PCIe Gen2 开关 203 的上行端口与申威处理器的 PCIe Gen2X8 接口相连,

PCIe 开关可扩展出 7 个 PCIe 端口,用于连接 PCIe 扩展槽或者 PCIe 设备,每个设备需占用一个 PCIe 端口(包括 PCIe-to-PCI 桥 204);

[0045] PCIe-to-PCI 桥 204 的上行端口与 8-Port PCIe Gen2 开关 203 相连,扩展出标准 32-bit PCI 总线,一条 PCI 总线上最多可挂载 4 个以上负载,连接 PCI 扩展槽或者 PCI 设备(包括 PCI-to-LPC 桥 205);

[0046] PCI-to-LPC 桥 205 的上行端口与 PCI 总线相连,扩展出 LPC 总线,用于挂载 SIO (Super I/O)、EC (Embedded Controller) 等传统设备。

[0047] Mini-BMC+ 维护模块 206 将轻量级的 BMC 和维护模块两个传统设备集成在一起,一方面与申威 -2 处理器的串口(带外)相连;另一方面作为传统设备,挂在 LPC 总线上,可以与申威处理器进行带内通信。维护模块外挂 FlashROM 芯片 207。

[0048] FlashROM 芯片 207,通过 SPI 接口与 Mini-BMC+ 维护模块 206 相连,用于存放 mini-BMC 的固件,存放申威处理器的初始配置和初始启动代码、BIOS 等信息,实现这些信息的非易失存储。

[0049] 图 4 是本发明 I/O 扩展架构方法的另一种实施例示意图。

[0050] 参照图 4 所示,本发明 I/O 扩展架构方法的另一个实施例中,I/O 扩展架构包括第二代申威处理器(申威 -2 处理器)301,DDR3 主存 302,6PortPCIe 开关 303,PCIe-to-PCI 桥 304,PCIe 复合设备 305,PCI-to-LPC 桥 306,mini-BMC+ 维护模块 307,FlashROM 芯片 308,其中,

[0051] 申威 -2 处理器 301,集成 DDR3 存控,直接挂载 DDR3 主存 302;集成标准 PCIe Gen2X8 接口,与 6-Port PCIe Gen2 开关 303 相连;具备低速串口(带外),与 mini-BMC+ 维护模块相连;

[0052] 6-Port PCIe Gen2 开关 303,上行端口与申威处理器的 PCIe Gen2X8 接口相连,PCIe 开关可扩展出 5 个 PCIe 端口,用于连接 PCIe 扩展槽或者 PCIe 设备,每个设备需占用一个 PCIe 端口;

[0053] PCIe-to-PCI 桥 304,上行端口与 6-Port PCIe Gen2 开关 303 相连,扩展出标准 32-bit PCI 总线,一条 PCI 总线上最多可挂载 4 个以上负载,连接 PCI 扩展槽或者 PCI 设备;

[0054] PCIe 复合设备 305,内部集成多个 PCIe 设备,上行端口与 6-Port PCIeGen2 开关 303 相连,扩展出 32-bit 局部总线,挂载 mini-BMC+ 维护模块 307;

[0055] PCI-to-LPC 桥 306,上行端口与 PCI 总线相连,扩展出 LPC 总线,用于挂载 SIO (Super I/O)、EC (Embedded Controller)、TPM 等传统设备。

[0056] Mini-BMC+ 维护模块 307 将 mini-BMC 和维护模块两个传统设备集成在一起,一方面与申威 -2 处理器的串口(带外)相连;另一方面挂在局部总线上,可以与申威处理器进行带内通信。Mini-BMC+ 维护模块外挂 FlashROM 芯片 308,两个传统设备共享 FlashROM 芯片 308。

[0057] FlashROM 芯片 308 通过 SPI 接口与 Mini-BMC+ 维护模块 307 相连,用于存放 mini-BMC 的固件,存放申威处理器的初始配置和初始启动代码、BIOS 等信息,实现这些信息的非易失存储。

[0058] 图 5 是本发明系统启动方法一种实施方式示意图。

[0059] 参照图 5 所示,本发明系统启动方法一种实施方式包括:mini-BMC 自检 401,系统配置保存 402,初始配置申威处理器 403,申威处理器初始化 404, BIOS 通路构建 405,取 BIOS 并执行 406。

[0060] 下面将具体描述各个步骤。

[0061] mini-BMC 自检 401:上电后(包括待机电源)最先执行,包括 mini-BMC 自身运行环境的自检,以及系统配置的检测,在系统具备开工条件的情况下,执行下一步骤,系统配置保存 402;

[0062] 系统配置保存 402:mini-BMC 将系统开工需要的一些信息,保存在 Flash ROM 中,之后让维护模块根据这些信息执行下一不步,初始配置申威处理器 403;

[0063] 初始配置申威处理器 403:在申威处理器处于复位状态下,配置好各类时钟发生器(如 PLL)的参数,处理器的工作模式,甚至包括处理器内部的指令 Cache,一些配置、控制寄存器,之后让申威处理器执行下一步骤,申威处理器初始化 404;

[0064] 申威处理器初始化 404:在申威处理器退出复位状态后,开始执行指令 Cache 中的代码;初始化内部寄存器和数据 Cache;对存控进行训练,以确定存控最终的配置参数;初始化主存,初始化 PCIe 接口;之后执行 BIOS 通路构建 405;

[0065] BIOS 通路构建 405:按照深度优先算法,进行 PCIe/PCI 设备的枚举,将传统 I/O 空间分配给首个下一级总线,传统设备是该算法找到的 PCI 总线树状组织结构中的首个末端设备;之后执行取 BIOS 并执行 406;

[0066] 取 BIOS 并执行 406:申威处理器通过操作两个传统 I/O 空间的寄存器来获取 FlashROM 中的 BIOS 和一些配置信息,将 BIOS 放入特定主存区域,并用配置信息构建 BIOS 运行所需要的一些表格;之后执行 BIOS,完成系统的启动。

[0067] 参照图 6 所示,本发明系统唤醒方法的一种实施方式包括:深度睡眠状态 501,唤醒事件 502,维护中断 503,推出睡眠 504,恢复运行现场 505,正常运行状态 506。

[0068] 下面将具体描述各个步骤。

[0069] 深度睡眠状态 501:在此状态下,申威处理器核心(Core)处于复位状态,其 PCIe 接口也停止工作,链路断开,只有低速的带外接口仍在工作;

[0070] 唤醒事件 502:在深度睡眠状态下,发生的 PME(平台管理事件)等中断,首先传递给维护模块,维护模块根据配置(唤醒中断屏蔽码),确定哪些中断可以唤醒处理器;

[0071] 维护中断 503:当维护模块确定要唤醒处理器后,通过串口向处理器发特殊的请求——维护中断请求;

[0072] 推出睡眠 504:当申威处理器收到维护中断请求后,PLL 开始提升核心工作频率,之后核心推出复位状态,开始执行指令;

[0073] 恢复现场 505:申威处理器首先判断维护中断的目的是否为了唤醒,然后执行一系列恢复现场的操作,恢复睡眠前进程的执行环境,处理引发唤醒的中断事件,之后处理器进入正常运行状态 506。

[0074] 此外,需要说明的是,除非特别指出,否则说明书中的术语“第一”、“第二”、“第三”等描述仅仅用于区分说明书中的各个组件、元素、步骤等,而不是用于表示各个组件、元素、步骤之间的逻辑关系或者顺序关系等。

[0075] 可以理解的是,虽然本发明已以较佳实施例披露如上,然而上述实施例并非用以

限定本发明。对于任何熟悉本领域的技术人员而言,在不脱离本发明技术方案范围情况下,都可利用上述揭示的技术内容对本发明技术方案作出许多可能的变动和修饰,或修改为等同变化的等效实施例。因此,凡是未脱离本发明技术方案的内容,依据本发明的技术实质对以上实施例所做的任何简单修改、等同变化及修饰,均仍属于本发明技术方案保护的范围内。

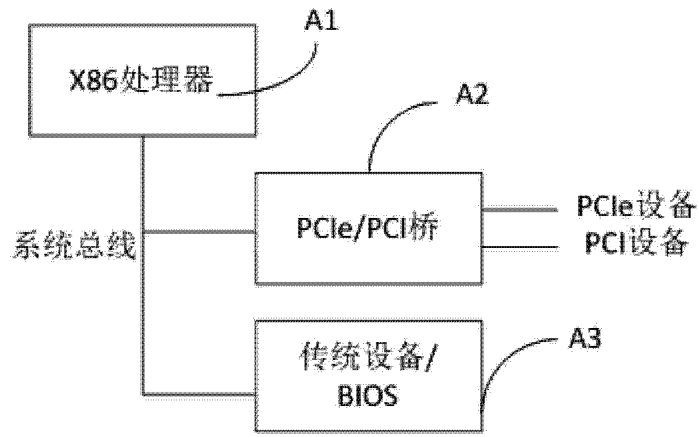


图 1

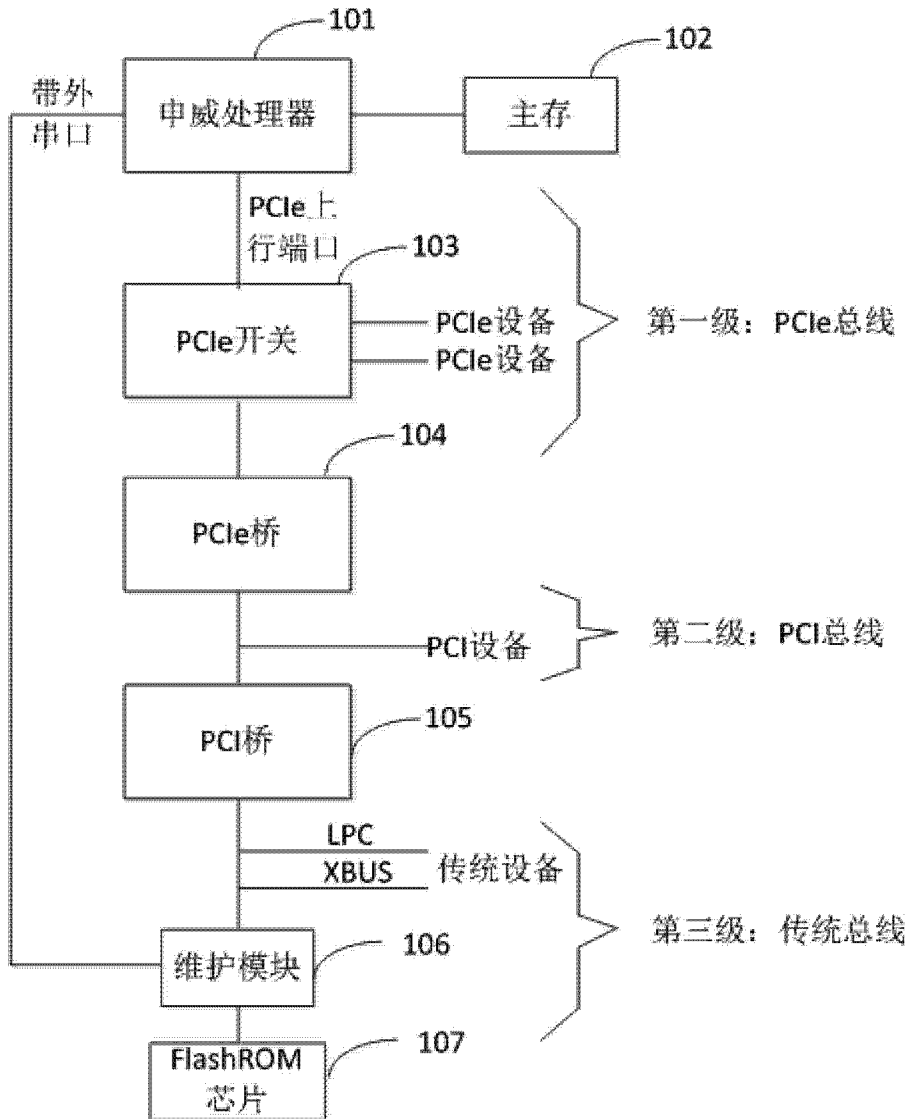


图 2

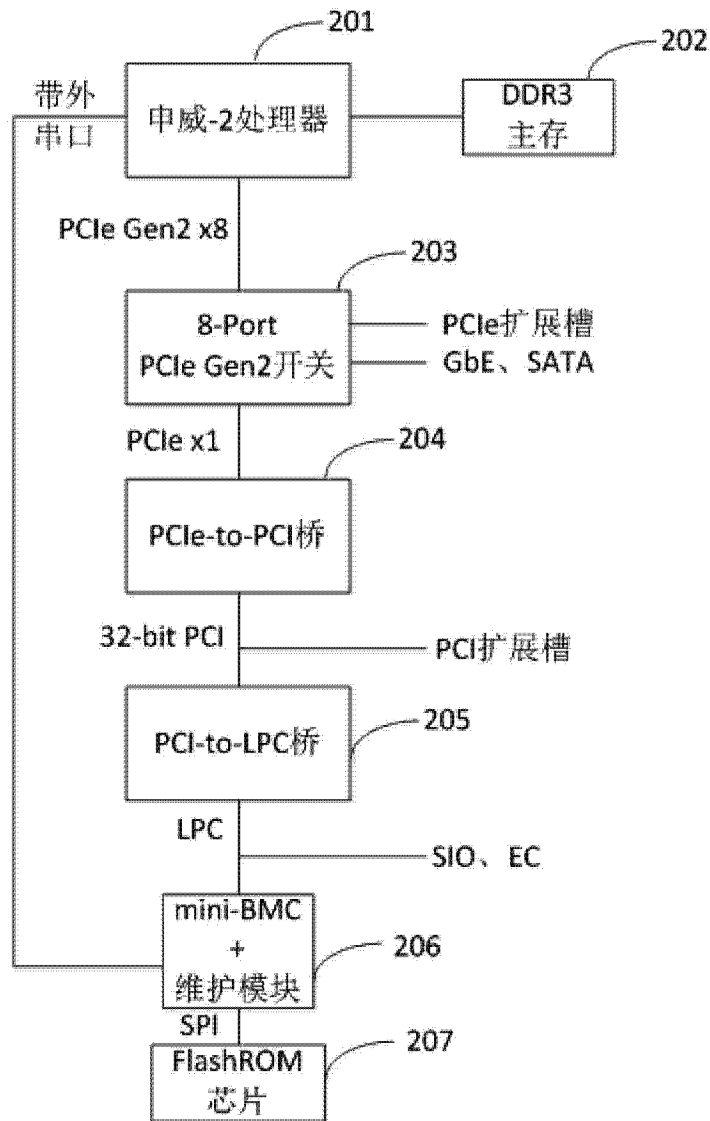


图 3

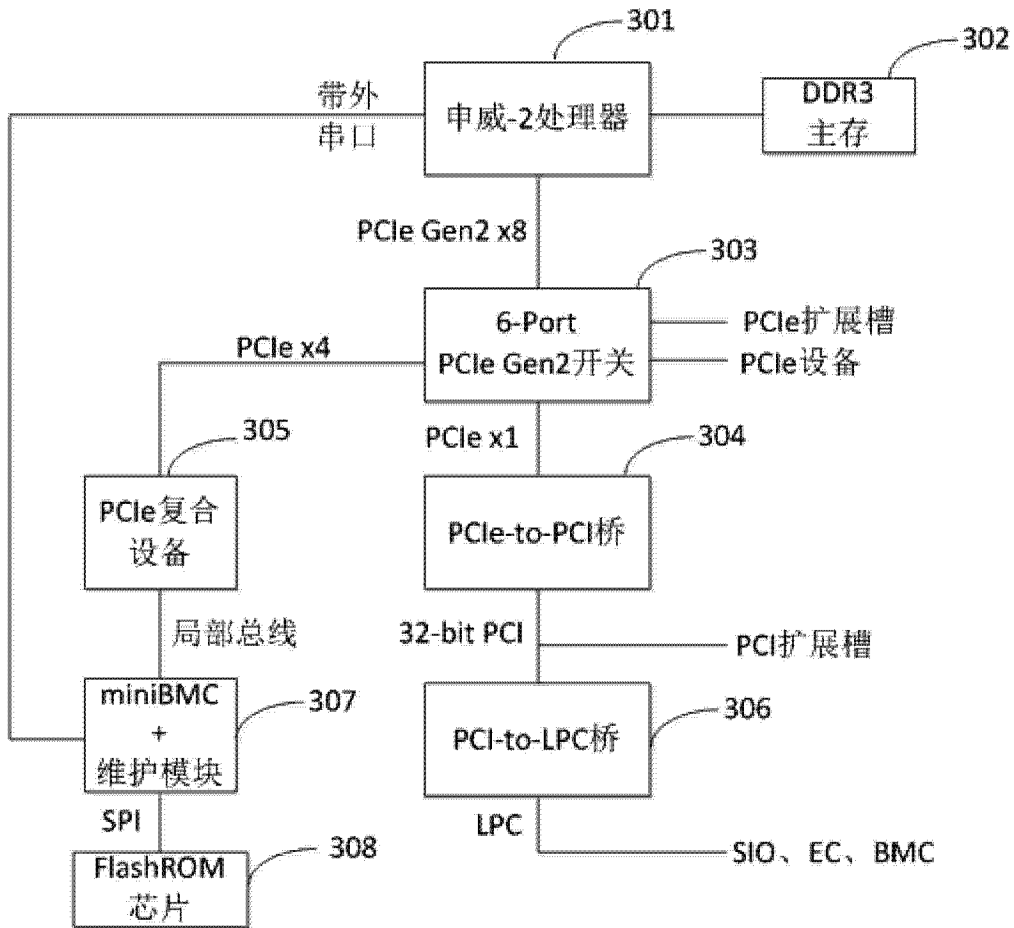


图 4

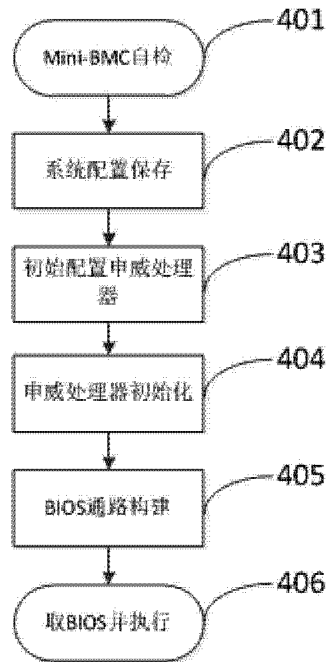


图 5

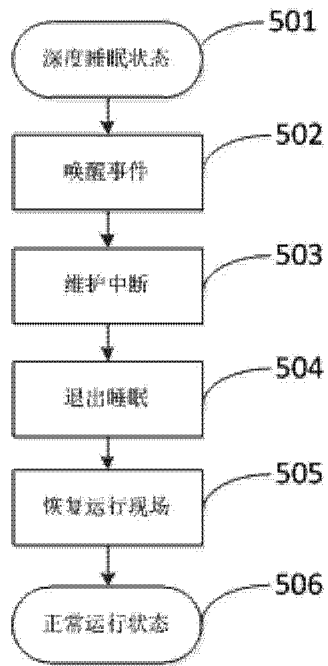


图 6