(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 102648453 B
(45)授权公告日 2015.11.25

(21)申请号 201080055347.6
(22)申请日 2010.11.22
(30)优先权数据
12/624,626 2009.11.24 US
(85)PCT国际申请进入国家阶段日
2012.06.06
(86)PCT国际申请的申请数据
PCT/US2010/057561 2010.11.22
(87)PCT国际申请的公布数据
WO2011/066202 EN 2011.06.03
(73)专利权人 超威半导体公司
地址 美国加利福尼亚州
(72)发明人 奥斯温·霍斯蒂
(74)专利代理机构 上海胜康律师事务所 31263
代理人 李献忠
(51)Int.Cl.
G06F 9/445(2006.01)
G06F 15/177(2006.01)

(56)对比文件
CN 101464813 A ,2009.06.24,
US 2006/0149959 A1 ,2006.07.06,
US 2008/0162878 A1 ,2008.07.03,
US 6158000 A ,2000.12.05,
WO 00/17750 A1 ,2000.03.30,

审查员 章媛

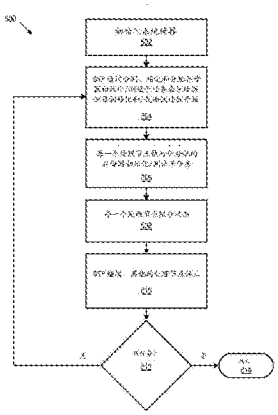
权利要求书2页 说明书8页 附图4页

(54)发明名称

存储器初始化的方法和装置

(57)摘要

在一个具有多个处理节点的系统中,控制节点将任务分成多个子任务并将子任务分配给一个或多个额外的处理节点,所述处理节点执行分配的子任务并将结果返回给控制节点,从而使多个处理节点有效地和快速地执行了存储器初始化及所有分配的子任务的测试。



CN 102648453 B

1. 一种导致多个处理节点执行包括存储器初始化任务的启动过程任务的方法，包括：

在控制处理节点将存储器初始化任务分成多个存储器初始化子任务，

通过分配给所述多个处理节点中的一个以 DIMM 存储器读取串行存在检测值的存储器初始化子任务，和分配给所述多个处理节点中的另一个以执行不依赖于串行存在检测值的复杂的初始化任务的存储器初始化子任务，将所述多个存储器初始化子任务分配在所述多个处理节点之间，使得每一个存储器初始化子任务都有相应的处理节点，

在所述相应的处理节点执行每一个存储器初始化子任务以生成子任务的结果，和

在所述控制处理节点结合来自所述多个处理节点的子任务的结果，其中所述多个存储器初始化子任务可以在所述多个处理节点并行执行或按顺序执行。

2. 如权利要求 1 所述的方法，还包括在分配所述多个存储器初始化子任务之前初始化所述多个处理节点之间的通信链接。

3. 如权利要求 1 所述的方法，在所述相应的处理节点执行每一个存储器初始化子任务包括规划调度每一个存储器初始化子任务在所述相应的处理节点执行。

4. 如权利要求 1 所述的方法，在所述相应的处理节点执行每一个存储器初始化子任务包括执行软件程序。

5. 如权利要求 1 所述的方法，还包括在所述控制处理节点接收关于所述存储器初始化子任务在所述相应的处理节点的执行的状态报告。

6. 如权利要求 1 所述的方法，其中，在所述多个处理节点中的一个与在所述多个处理节点中的另一个并行执行各自相应的存储器初始化子任务。

7. 一种计算机系统，包括：

系统存储器；和

多个处理节点，所述处理节点中的每一个都包括处理器核、与所述多个处理节点中的至少另外一个的通信接口和用于管理进出系统存储器的数据流的存储器控制器，所述多个处理节点包括主处理节点和一个或多个执行处理节点，其中所述主处理节点配置为通过将所述存储器初始化任务分成多个存储器初始化子任务和将所述多个存储器初始化子任务分配给所述一个或多个执行处理节点以使每一个被分配的存储器初始化子任务都有相应的执行处理节点来在系统存储器上执行存储器初始化任务，其中所述多个存储器初始化子任务可以在所述一个或多个执行处理节点并行或按顺序执行以便分配给所述多个执行处理节点中一个的存储器初始化子任务不按顺序不依赖于分配给所述多个执行处理节点中的另一个的其它的存储器初始化子任务而执行。

8. 如权利要求 7 所述的计算机系统，其中，每一个执行处理节点配置为执行由主处理节点分配给它的每一个存储器初始化子任务并由此生成子任务的结果。

9. 如权利要求 7 所述的计算机系统，其中，所述主处理节点进一步配置为结合从所述一个或多个执行处理节点来的子任务结果。

10. 如权利要求 7 所述的计算机系统，其中，所述系统存储器包括多个双列直插存储器模块。

11. 如权利要求 7 所述的计算机系统，其中，所述主处理节点进一步配置为接收从所述一个或多个执行处理节点来的所述存储器初始化子任务的状态。

12. 如权利要求 7 所述的计算机系统，其中，所述主处理节点是执行存储器初始化子任

务的所述一个或多个执行处理节点中的一个。

13. 如权利要求 7 所述的计算机系统，其中，所述一个或多个执行处理节点中的一个与所述一个或多个执行处理节点中的另一个并行执行各自相应的存储器初始化子任务。

14. 如权利要求 7 所述的计算机系统，其中，所述主处理节点通过基于在所述一个或多个执行处理节点执行所述多个初始化存储器子任务的结果程序化所述主处理节点的所述存储器控制器中的一个或多个寄存器来执行所述存储器初始化任务。

15. 一种计算机系统，包括：

存储器阵列；和

多个处理节点，所述处理节点中的每一个都包括处理器、用于与所述存储器阵列通信的存储器控制器和所述多个处理节点中的至少另外一个的通信接口；

其中，所述多个处理节点中的一个将存储器初始化任务分成多个子任务和将所述多个子任务分配给所述多个处理节点；

其中，所述多个处理节点中的每一个能够在执行所分配的子任务的同时从所述存储器阵列的任何部分获取信息和将子任务的执行结果返回给所述多个处理节点中的所述一个，从而使得所述多个处理节点有效地和快速地执行所述存储器阵列的存储器初始化；以及

其中，所述多个处理节点中的所述一个通过基于在所述多个处理节点执行所述多个子任务的结果程序化所述存储器控制器中的一个或多个寄存器来执行存储器初始化任务，从而与通过只在所述多个处理节点中的一个上执行所述存储器初始化任务来启动所需要的时间相比减少了启动时间要求。

16. 如权利要求 15 所述的计算机系统，其中，所述存储器阵列包括通过双数据速率 (DDR) 总线与所述多个处理节点相连的多个双列直插存储器模块。

存储器初始化的方法和装置

技术领域

[0001] 本发明一般性地涉及数据处理系统。在一方面，本发明涉及用于在系统启动初始化期间进行存储器初始化的方法和装置。

背景技术

[0002] 数据处理或计算系统被设计为给予一个或多个用户独立的计算能力，可以在包括例如大型机、小型机、工作站、服务器、个人计算机、互联网终端、笔记本计算机和嵌入式系统等许多形式中找到。一般地，计算机系统结构设计为包括提供一个或多个高速高带宽访问所选择的系统组件、相关的存储器和控制逻辑（典型地是在系统板上）的微处理器核和为系统提供输入和/或输出(I/O)的若干外围设备。例如，图1示出了一种用于传统的计算机系统100的举例式结构。计算机系统100包括一个或多个处理器102，其跨越“北”桥104与系统存储器108相连。通常，存储器阵列108包括一个或多个存储器模块，还可包括用于增加或替换存储器模块的存储器插槽。北桥电路104可以编程为与多种存储器模块接口，如图所示，北桥电路104为多个存储器模块108所共享。结果，如果板上组装了不同的存储器模块，北桥电路104就必须用使每一个存储器模块都正常运行的参数进行编程。所绘出的北桥电路104通过高速高带宽的总线（例如，存储器总线107）连接到存储器108，还通过高速高带宽的总线（例如，ALINK或PCI总线）连接到“南”桥112。“南”桥112连接到诸如外设组件互连(PCI)总线110(而它又连接到网络接口卡(NIC)120)、串行AT附件(SATA)接口114、通用串行总线(USB)接口116和低引脚数(LPC)总线118(而它又连接到超级输入/输出控制器芯片(SuperI/O)122和BIOS存储器124)之类一个或多个I/O设备。可以理解，如果需要，可在计算机系统100中包括其它的总线、设备和/或子系统，诸如缓冲存储器、调制解调器、并行或串行接口、SCSI接口等。还有，北桥104和南桥112可以用单个芯片或多个芯片实现，这导致了一个集合式的术语“芯片组”。可替换地，北桥芯片的功能的全部或部分可以存在于处理器102内。

[0003] 计算机系统典型地包括一组称作基本输入/输出系统(BIOS)的内置的软件例程，它提供了在系统硬件和操作系统软件之间的软件接口，使得程序员和用户可以与系统硬件进行交互。BIOS指令存储于非易失性存储器124(例如ROM(只读存储器)、PROM(可编程ROM)、EPROM(可擦除PROM)、EEPROM(电可擦除PROM)、闪存RAM(随机存取存储器)或类似物)内，并用来控制在上电(power up)时的重要的计算机系统功能，包括测试和初始化存储器、盘存和初始化系统、测试系统。在上电时的这些功能被称作“系统启动”或“启动系统”，在每次系统上电或复位时会发生。在图1所示的传统的计算机系统100中，BIOS在称为启动纽带处理器的处理器102中的一个上运行，以启动计算系统100。在操作中，启动纽带处理器102通过北桥104与存储器阵列108通信。北桥电路104包括存储器控制器106，存储器控制器106连接到与诸如双倍数据速率(DDR)总线之类存储器总线107接口的一个或多个通道控制器。北桥电路104还在例如是外设组件互连(PCI)总线的标准总线109上与一个或多个南桥112通信。南桥112与一个或多个输入/输出(I/O)设备120、122、124

通信，但能够将更多的或更少的设备（未图示）附在南桥 112 上。

[0004] 一旦进行系统初始化，启动纽带处理器 102 就在系统启动期间执行所有的存储器测试和清除，这会占用大量时间。大型服务器系统往往需要若干分钟来启动，同时其它的处理器则处于闲置状态。在所描述的例子中，在启动纽带处理器上执行的 BIOS 通过从启动纽带处理器 102 基于来自 DIMM 存储器 108 的、在 DDR 存储器总线 107 上的信息编程寄存器来初始化北桥电路 104 的存储器控制器 106。存储器初始化可包括核实存储器模块数、验证存储器的正确操作（无卡位）和将存储器初始化或清除为已知值。因为有大容量存储器 108（例如，8、16 或 32GB 的存储器），传统系统会需要若干分钟来初始化存储器，特别是在“DDR 训练”过程使用启动纽带处理器 102 通过连续执行按顺序排列的存储器初始化任务而不考虑任务间的依赖关系来初始化存储器的情况下。当有大数目的存储器 108 附在每个控制器 106 上时，这一延迟加剧。

[0005] 因此，存在着对改进型存储器初始化设备、方法和系统的需求，它们解决了上述命名的发明者发现的在本领域的各种问题，在本领域技术人员参考附图和随后的详细说明审阅本发明申请的其余部分后，传统的解决方案和技术的各种局限和不足对他们来说将变得明显，然而，应该理解，本相关技术部分的说明无意于作为对所说明的主题题材是现有技术的认可。

发明内容

[0006] 从广义上讲，本发明的实施例提供了用于通过在核间分配存储器初始化任务使得执行可以不按顺序且不依赖于前面的任务进行，从而在一个多核计算机系统内进行存储器初始化的系统、方法和装置。在所选择的实施例中，提供了一种包括多个处理节点或多个核和多个系统存储器部件的系统。在所选择的实施例中，选择处理节点或核中的一个来提供控制节点 / 核功能将存储器初始化任务分成各个由多个 CPU 节点 / 核处理的子任务。各个子任务被发送给或分配给其它的“从”核来执行。在每个从核处，子任务被规划为，在从核有资源处理子任务时就执行。结果，在给定从核处的给定的子任务的执行时间可能会或可能不会与在其它从核处的其它子任务的执行重叠。结果，有可能是从核在并行或串行地执行子任务。在子任务处理后，它们被控制核重组以完成原来的任务。这将允许任务执行同时进行，导致了减少的整体执行时间和最佳的性能，因为有可能是从内核并行执行子任务。

[0007] 根据本发明的各个实施例，提供了一种方法来造成多个处理节点执行诸如存储器初始化任务之类启动过程的任务。在该方法的一个典范实施例中，控制处理节点将启动过程的任务（例如，存储器初始化任务）分成多个启动过程的子任务（例如，存储器初始化子任务）然后在可包括控制处理节点的多个处理节点之间分配子任务，使得每个子任务有相应的处理节点。在分配子任务之前，控制处理节点初始化多个处理节点之间的通讯连接。为了提供子任务分配的一个例子，可将处理节点中的一个分配为从 DIMM 存储器读取串行存在检测 (SPD) 值的存储器初始化子任务，而将另一个处理节点分配为执行复杂的初始化任务中的不依赖于 SPD 值的存储器初始化子任务。每个存储器初始化子任务被规划执行，然后在相应的处理节点执行以生成子任务的结果。在所选择的实施例中，使用软件程序来在相应的处理节点执行存储器初始化子任务。控制处理节点接收有关存储器初始化子任务在相应的处理节点处的执行的状态报告，然后在控制处理节点将来自多个处理节点的子任务

的结果结合，其中多个存储器初始化子任务可在多个处理节点并行执行或按顺序执行。例如，处理节点中的一个可以与另一个执行它的相应的存储器初始化子任务的处理节点并行执行分配给它的存储器初始化子任务。

[0008] 在其它的实施例中，提供了一种计算机系统，其中包括系统存储器（例如，基于DIMM的阵列）和多个处理节点，其中每个处理节点包括处理器核、至其它的处理节点的通讯接口和用于管理来往系统存储器的数据流的存储器控制器。处理节点中的一个是主处理节点，它被配置为通过将存储器初始化任务分成存储器初始化子任务和将存储器初始化子任务分配给一个或多个执行处理节点（其中可包括主处理节点）使得每一个分配的存储器初始化子任务有一个相应的执行处理节点的方式来在系统存储器上执行存储器初始化任务。每个执行处理节点被配置为执行由主处理节点分配给它的每一个存储器初始化子任务并由此产生子任务的结果。主处理节点被配置为从执行处理节点接收存储器初始化子任务的状态并合并来自执行处理节点的子任务，通过基于来自在执行处理节点处执行的多个存储器初始化子任务的结果编程主处理节点的存储器控制器中的一个或多个寄存器以执行存储器初始化任务。在所选择的实施例中，主处理节点划分和分配存储器初始化子任务，以使分配给执行处理节点中的一个的存储器初始化子任务能够不按顺序不依赖于分配给另一个执行处理节点的其它存储器初始化子任务执行。这样，第一执行处理节点可以在另一个执行处理节点执行其相应的存储器初始化子任务时并行执行它的存储器初始化子任务，使得多个存储器初始化子任务可以在执行处理节点并行或按顺序执行。

[0009] 在另外的实施例中，提供了一种计算机系统，其包括存储器阵列和多个处理节点。在多个双列直插存储器模块通过双数据速率（DDR）总线连接到多个处理节点时，存储器阵列可以实现。每一个处理节点可以作为处理器、用于与存储器阵列通信的存储器控制器和至所述多个处理节点中的至少一个另外的处理节点的通信接口实现。在操作中，第一个处理节点将存储器初始化任务分成多个子任务和将多个子任务分配给处理节点。每个处理节点可以在执行分配的子任务的同时从存储器阵列的任何部分获得信息。另外，每个处理节点将子任务的执行结果返回给第一个处理节点，从而使多个处理节点有效地和快速地执行存储器阵列的存储器初始化。这样，第一个处理节点被配置为通过基于在处理节点处执行的多个子任务的结果编程存储器控制器中的一个或多个寄存器执行存储器初始化任务，从而与通过只在处理节点中的一个上执行存储器初始化任务来启动将需要多少时间相比减少了启动时间要求。

附图说明

[0010] 通过参考附图，本领域的技术人员可以更好地理解本发明，本发明的众多的目的、特征和优势变得明显。在所有的几个附图中，使用相同的参考编号来指定相似或类似的部件。

[0011] 图 1 示出了多核计算机系统的简化的体系结构的框图。

[0012] 图 2 示出了根据本发明的选择实施例的、具有分布式多核存储器初始化的计算系统体系结构。

[0013] 图 3 示出了根据本发明的选择实施例的典范的处理节点。

[0014] 图 4 示出了典范的 DIMM（双列直插式存储器模块）。

[0015] 图 5 示出了根据本发明的所选择实施例的优化存储器初始化和测试的流程图。

具体实施方式

[0016] 提供了用于在主核的控制下将存储器初始化任务分成子任务的分布式多核存储器测试和初始化的方法和装置。主核然后发送或分配子任务由多个核单独执行，一旦完成后，就将结果送回主核重组以完成原来的任务。通过将存储器初始化任务分配给各个核，子任务的执行可以不按顺序和不依赖于先前的任务实现。这种分布式的执行允许某些任务不按顺序不等待先前的子任务来完成，从而通过减少存储器初始化时间所需的时间长度提高了设备的性能。

[0017] 现在将参考附图详细说明本发明的各种说明实施例。虽然在下面的说明中提出了各个细节，但可以理解，本发明可以没有这些具体细节而执行，并且可以对本文说明的发明做出大量与具体实现相关的决定来实现设备设计者的特定目标，如符合在各个实现中会有所不同的工艺技术或与设计相关的限制。尽管这样的开发努力可能是复杂和费时的，但对于得益于本公开的本领域的普通技术人员而言，这仍是常规工作。例如，为了避免使本发明受限或费解，以框图的形式而不是详尽地示出了所选择的方面。以对存储在计算机存储器中的数据进行操作的算法和指令的形式说明了此处提供的详细说明的某些部分。本领域的技术人员使用这样的说明和表述来对本领域的其他技术人员说明和传达他们的工作的实质。一般地，算法是指导致期望的结果的步骤的自洽的 (self-consistent) 的序列，其中“步骤”是指操纵物理量，但可以，但非必然地需要，采用能存储、传输、结合、比较和进行其它操作的电信号或磁信号的形式。通常的使用是这些信号指位、值、要素 (element)、符号、字符、术语、数字或类似信号。这些术语以及类似的术语可与适当的物理量关联，只是应用到这些数量上的方便的标志。可以理解，在下面的讨论中，除非特别明确声明，在整个说明中，使用诸如“处理”或“计算”或“运算”或“确定”或“显示”或类似的术语的讨论是指将在计算机系统的寄存器和存储器中以物理（电子）量表示的数据操作和转换为在计算机系统的存储器或寄存器或其它的这样的信息存储、传输或显示设备中以物理（电子）量类似地表示的其它数据的计算机系统或类似的电子计算设备。

[0018] 现在参考图 2，图 2 示出了根据本发明的选择实施例的具有分布式多核存储器初始化的计算系统体系结构 200。在描绘的计算系统 200 中有多个通过链接 203 相互通信的处理节点 202[0:3]。例如，每一个处理节点 202 包括处理器核、存储器控制器和链接接口电路。链接 203 可以是根据诸如超传输 (HyperTransport(HT)) 协议之类数据分割传输 (split transaction) 总线协议的双向点至点链接。链接 203 可以包括下行数据流和上行数据流。链接信号通常包括诸如时钟、控制、命令、地址和数据信息和对在设备间流动的流量进行资格化和同步化的链接单边带信号。处理节点 202 的每一个存储器控制器与相应的存储器阵列 206[0:3] 进行通信。处理节点 202 和存储器阵列 206 在系统的“相干”部分内，其中所有的存储器事务操作是一致的。北桥设备可以包含在一个或多个处理节点 202 内，或者可以作为通过另一个 HT 链接耦合到处理节点 202 中的单独的北桥设备 208 提供。在任一情况下，北桥设备可以通过另一个 HT 链接耦合到南桥 210。此外，一个或多个 I/O 设备 212 可以耦合到南桥 210。BIOS ROM214 可以耦合到南桥 210。北桥 208、南桥 210 和 I/O 设备 212 是在系统的“非相干”部分内。

[0019] 每一个存储器阵列 206 可以包括几个板上组装或未组装的存储器插槽来增加或替换存储器模块。例如，在服务器系统 200 中，每一个存储器插槽可提供 512 兆字节 (MB) 的存储能力，使得整个服务器系统 200 有一个例如 32 千兆字节 (GB) 的存储能力的大存储器。每一个处理节点 202 的存储器控制器可以不同地编程，但必须编程以与耦合到相关处理节点 202 的存储器模块 206 的本地化品种接口。可以理解，系统 200 可以比显示的更复杂。例如，在系统的相干部分 202 可以有额外的处理节点。虽然处理节点 202 是以阶梯式体系结构说明的，但处理节点 202 能够以各种不同的方式互联，能够有通过额外的 HT 链接的更复杂的相互耦合。

[0020] 图 3 示出了根据本发明的选择实施例的典范的处理节点 202。如所示的那样，处理节点 202 包括处理器核 302、多 HT 链接接口 304 和存储器控制器 306。处理器核 302 为处理器节点执行的存储器初始化和测试任务执行代码指令。例如，如果处理器节点作为控制节点或主节点，那么处理器核 302 就执行用于将存储器初始化和测试任务分成指定和分配给其它的处理器节点的多个子任务的代码指令。可替换地，如果处理器节点作为执行节点或从节点之一，那么处理器核 302 就执行用于获取、调度和执行指定的存储器初始化和测试子任务并将结果返回到控制节点 / 主节点的代码指令。交叉横梁 (crossbar)308 将请求、响应和广播消息传输给处理器 302 和 / 或适当的 HT 链接接口 304。请求、响应和广播消息的传输由位于每一个处理节点的多个配置路由表引导，配置路由表必须通过 BIOS 配置。此外，存储器控制器 306 包含必须由 BIOS 编程的多个用于运行参数的配置寄存器。

[0021] 图 4 示出了典范的 DIMM(双列直插式存储器模块)400。可以理解，几个 (通常是八个)DIMM 可以做成存储器阵列 206，其中每一个 DIMM 400 含有多个诸如双数据速率 (DDR) 存储器芯片之类随机存取存储器 (RAM) 集成电路或芯片 402[1:N]。此外，DIMM400 能够含有 ECC(纠错码) 电路 404。ECC 电路 404 存储有允许发现和纠正存储器错误的纠错代码。此外，DIMM 400 能够含有 SPD(串行存在检测) 电路 406。SPD 电路 406 包含指定 DIMM 400 的操作范围的只读信息和与一个人会在数据表中发现的信息类似于的其它信息。例如，SPD 电路 406 标识了 DIMM 400 的存储器的存储容量和诸如最小周期时间、CAS 延迟等之类的操作参数。

[0022] 一旦系统初始化后，每一个存储器模块 206 必须进行初始化和测试。这可以包括核实存储器模块的板上组装、验证存储器的正确操作 (无卡位) 和将存储器初始化或清除为已知值。每一个存储器模块可以利用 ECC 定期擦除以纠正任何存储器错误。

[0023] 图 5 示出了根据本发明的选择实施例的、优化的存储器初始化和测试序列 500 的流程图。序列始于步骤 502，通过初始化系统链接，使得每一个处理节点能够与其它的处理节点和存储器通信。这通常是由在启动纽带处理器上运行的 BIOS 执行的。可替换地，在硬连线系统中，处理节点能够一旦系统上电就自动配置为相互通信和与存储器通信。

[0024] 在步骤 504，被指定为启动纽带处理器 (BSP) 的处理节点通过将存储器初始化 / 测试任务分成多个子任务然后将每一个子任务指定给和分配到执行处理节点来导致存储器初始化和 / 或测试过程开始。例如，BSP 能够将开始的消息随指定子任务发送给每一个处理节点。另外或可替换地，能够将一个位写入每一个处理节点来指明存储器初始化和测试过程即将开始，使得所分配的子任务能够由 BSP 提供或由执行处理节点检索。可替换地，诸如在任意的处理器上执行的代码写入到一个或多个寄存器时，引导的中断能够导致过程开

始。写入的值能够引导一个或多个处理器对分配给它的子任务采取具体行动。例如，写入的值能够包括标识引导中断的目标（target）的节点标识器和通过直接或间接指定开始执行代码的地址表明一系列指令和子任务的执行的矢量。如果需要，硬件机制能够将启动消息发送给目标处理器来提示目标处理器开始执行由向量指定的分配的子任务。

[0025] 在步骤 506，每一个处理节点 202 执行分配给它的可能响应或可能不响应相关联的存储器 206 的存储器初始化和测试子任务。在图 2 的一个示例实施中，计算系统 200 包括四个核、一个控制核 202[0] 和三个从核 202[1:3]。存储器初始化 / 测试任务由控制核 202[1] 分成多个子任务（例如，子任务 1、子任务 2、子任务 3 等）。控制核 202[0] 将子任务 1 发送给第一从核 202[1]，将子任务 2 发送给第二从核 202[2]，和将子任务 3 发送给第三从核 202[3]。在步骤 504，每一个从核 202[1:3] 接收其子任务和规划执行。此外，控制核 202[0] 可继续执行不同的子任务或等待执行在从核上完成，或者可自己执行单独的子任务。每一个从核 202[1:3] 在它有资源处理任务时及时在一些点执行子任务，作为结果，执行时间可能会或可能不会与其它的从核重叠。作为结果，取决于在从核上的可用的资源，有可能是分配的存储器初始化 / 测试子任务由从核 202 以串行方式或并行来执行。子任务的执行能够由处理节点实施，或由在每一个处理节点的北桥内的电路实施。

[0026] 可以理解，控制核和从核可位于在独立的平台系统上。在这些实现中，控制核将子任务发送到每一个参与的平台系统，使那个系统上的一个或多个核能够每一个处理分配给它的子任务。在这种情况下，执行可串行或并行进行。作为存储器初始化任务执行的分布性质的结果，是并行执行还是串行执行的事实与任务的完成是不相干的。如果是并行执行任务，任务可能完成得更快，但传输数据的延迟可能会导致在控制核的串行化。

[0027] 在步骤 508，每一个处理节点将状态报告给 BSP。状态报告能够例如在存储器初始化和测试期间是连续的，或在存储器初始化和测试完成时是定期式的。此外，状态报告能够以多种方式报告。例如，启动纽带处理器能够定期发送一个查询到每一个处理节点，以询问所分配的子任务是否已经执行。可替换地，每一个处理节点能够发送一条向启动纽带处理器指示它的状态的消息。作为另一替换，每一个处理节点可以向本地寄存器或甚至是启动纽带处理器的本地存储器写入来报告状态。

[0028] 在所有的处理节点完成所分配的存储器初始化 / 测试子任务的执行后，启动纽带处理器继续系统启动，其它的处理节点停止，步骤 510。如果是存储器初始化 / 测试任务要执行（决定的肯定结果 512），过程就在步骤 504 重新开始。否则，过程就在步骤 514 结束。

[0029] 通过将存储器初始化 / 测试子任务分配到不同的处理节点，系统启动能够显著加快地完成。即使一些子任务依赖于先前的子任务的完成，其它的任务也能够不按顺序不等待先前的子任务完成来完成，通过不考虑依赖的任务和那些没有依赖的任务而进行分割、指定和分配子任务，能够改善整体性能和启动时间。这源于总的“系统启动”或“启动系统”的时间减少，因为一些子任务由第一节点处理而不必等待出现在它们前面的任务按顺序执行和它们不赖于在它们之前执行。当然，如果启动纽带处理器考虑到对每一个子任务的各自的依赖将存储器初始化 / 测试任务分成子任务，则存储器初始化可进一步改善。例如，可以向一个或多个指定的处理节点分配依赖于其它的子任务的结果的子任务，而可以向一个或多个指定的处理节点分配可以不按顺序完成不等待前面的子任务完成的子任务。

[0030] 如本文所述，分布式存储器初始化例程可以由 BIOS 代码实现，它能够通过将存储

器任务分割以由单独的节点分配和执行来配置、初始化和测试在一个多处理器系统的多个节点之间分布的存储器，造成改进的执行时间。例如，能够使一个核做从 SMBus（慢总线）读取 SPD 的任务，而能够使另一个核执行不依赖于 SPD 的复杂的初始化任务。所公开的分布式处理方法还能够改善某些 BIOS 算法的准确性。例如，开发了训练算法借以使 BIOS 优化以更快执行，借以生成能够用于确定延迟设置的最佳位置的数据眼。然而，所生成的数据眼受能够由执行训练算法的核处理的数据量的限制。这里公开的分布式处理方式能够用来通过使用主核收集数据和将数据分割以分布给处理信息的系统中的其它核并将结果返到主核来提高精度。在分配的子任务不按顺序执行时，所公开的分割和分配计划也导致了更快的启动时间。另一种可能的好处是，通过将子任务分配给每个核而不是让启动纽带处理器执行所有的 BIOS 初始化任务，在 BIOS 初始化期间处理核可更有效地使用。

[0031] 为了提供了进一步的例子和解释如何将存储器初始化任务分割和配置给总的子任务 st 数小于或等于系统中的总的核数的多个处理节点，现在参考以下的存储器初始化程序或算法：

[0032] T = {st[1]+st[2]+st[3]+…st[X]}，其中 st 是存储器初始化子任务；

[0033] C0 = 负责结合“st”创建“T”的主初始化核；

[0034] c[n] = 子任务核，其中 n = {1…N}；

[0035] N = 核的最大数，使得 N ≥ X；

[0036] X = 对任何 T 的子任务的最大数；

[0037] n - 当前的核数；

[0038] x- 当前的子任务数；

[0039] 步骤 1- 将子任务分配给核；

[0040] 步骤 2- 为了 c[n]；

[0041] 步骤 3- 为了 st[x]；

[0042] 步骤 4- 将 st[x] 初始化信息和代码从 c0 发送到 c[n]；

[0043] 步骤 5-c[n] 开始 s[x] 的执行；

[0044] 步骤 6-c0 从 c[n] 接收 st[x] 的结果并开始执行；

[0045] 步骤 7-c0 增加 n 和 x；

[0046] 步骤 8- 对所有的 X，c0 重复步骤 2 至 7；

[0047] 步骤 9- 在 n = N 时，移动到步骤 10；

[0048] 步骤 10-c0 收集分布式子任务的数据；

[0049] 步骤 11- 为了 c[n]；

[0050] 步骤 12-c0 从 c[n] 请求 st[x] 的数据；

[0051] 步骤 13-c[n] 完成执行并将结果发送给 c0；

[0052] 步骤 14-c0 增加 n 和 x；

[0053] 步骤 15- 对所有的 X，c0 重复步骤 11 至 14，直到所有的 n = N 为止；

[0054] 步骤 16-c0 通过再结合来自所有的 st[1-X] 的数据生成 T；

[0055] 步骤 17- 对下一个任务重复所有步骤。

[0056] 可以理解，上述程序能够以伪代码的形式和 / 或翻译成用于它们正在使用的特定指令集的体系结构的相应的汇编语言或高层次的语言码表示。此外，本文所述的操作能够

包括通过由计算机系统的用户直接输入的指令和/或由软件模块执行的步骤。本文提及的任何步骤的功能可对应于模块或模块的部分的功能。除了软件模块外，上述流或流的部分可以作为应用指令实现。本文所指的操作可以是模块或模块的部分（例如，软件、固件或硬件模块）。例如，本文讨论的软件模块可包括脚本、批处理或其它可执行文件或这样的文件的组合和/或部分。软件模块可包括编码在计算机可读介质上的计算机程序或其子程序。

[0057] 此外，本领域的技术人员将认识到，模块间的界限只是说明性的，替代的实施例可合并模块或施加可替代的功能模块分解。例如，本文讨论的模块可以分解成作为多个计算机过程执行的子模块。此外，替代的实施例可以结合特定模块或子模块的多个实例。此外，本领域的技术人员将认识到，在典范的实施例中所述的操作仅是为了说明。操作可以合并或操作的功能可以根据本发明分布在额外的操作中。因此，本文所述的流及其操作和因而模块可以在配置为执行流的操作的计算机系统上执行，和/或可从计算机可读介质执行。所述的流可实施在用于配置计算机系统执行所述流的、机器可读和/或计算机可读的介质中。因此，软件模块可以存储在计算机系统的存储器内和/或传送给计算机系统的存储器来配置计算机系统执行模块的功能。

[0058] 上面公开的具体实施例仅仅是说明性的，不应作为对本发明的限制，这是因为，本发明可以被修改和以不同但对受益于本文的教义的本领域的技术人员而言明显是等同的方式实施。因此，上述说明无意于将本发明限制为规定的特殊形式，与此相反，是为了包括可包括在由附加的权利要求定义的本发明的精神和范围内的这些可替代物、修改和等同物，是以本领域的技术人员应该明白，他们可以在最广泛的形式上不脱离本发明的精神和范围而做出各种改变、变换和替代。

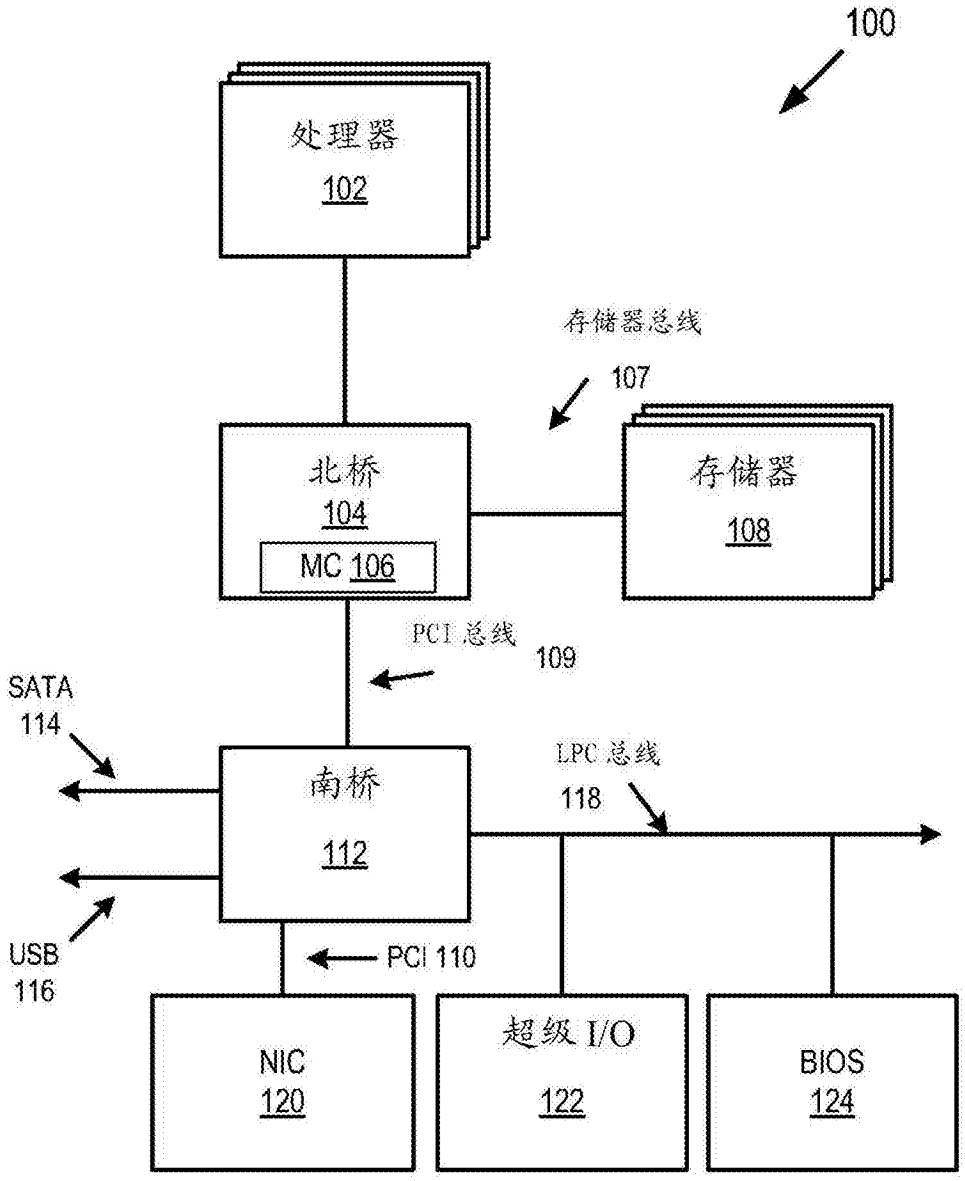

100
处理器
102
存储器总线
107
北桥
104
MC 106
存储器
108
PCI 总线
109
SATA
114
南桥
112
LPC 总线
118
USB
116
PCI 110
NIC
120
超级 I/O
122
BIOS
124

图 1

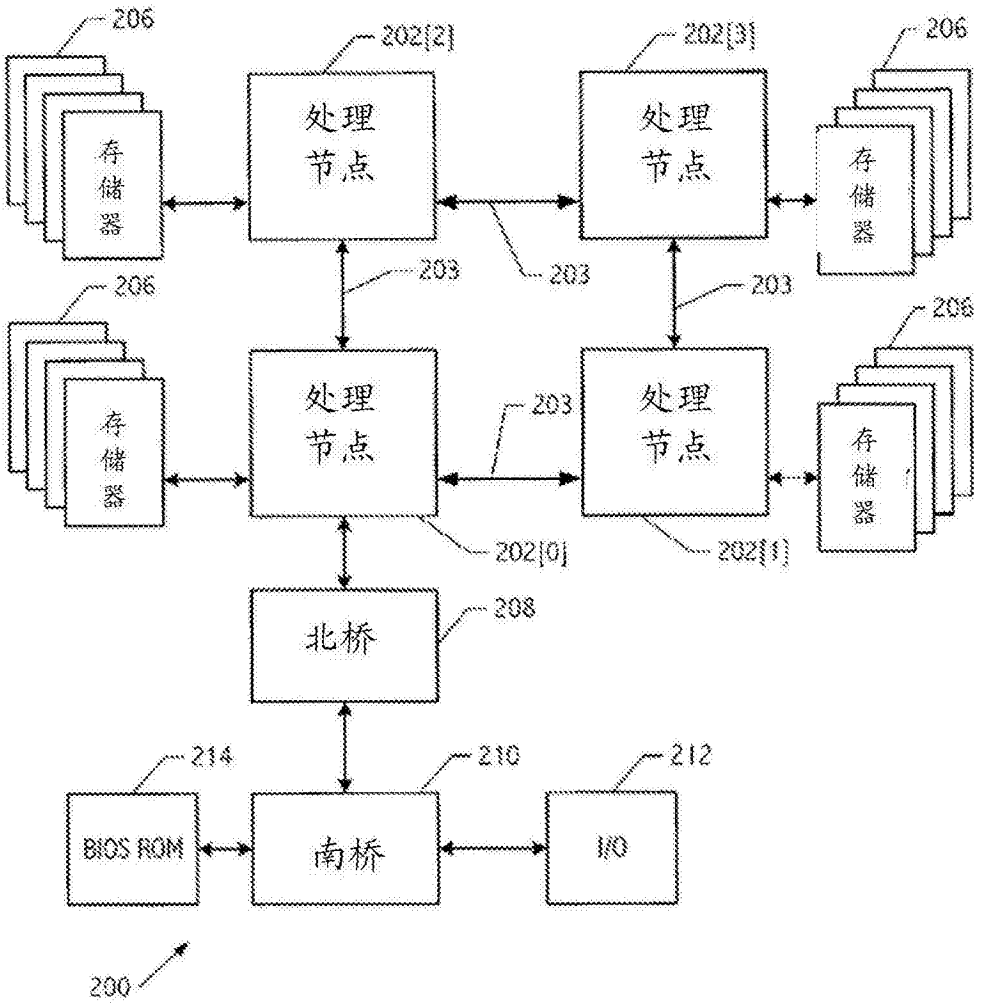

206
202[2]
202[3]
206
处理节点
处理节点
存储器
存储器
203
203
203
206
206
处理节点
处理节点
存储器
存储器
203
202[0]
202[1]
北桥
208
214
210
212
BIOS ROM
南桥
I/O
200

图 2

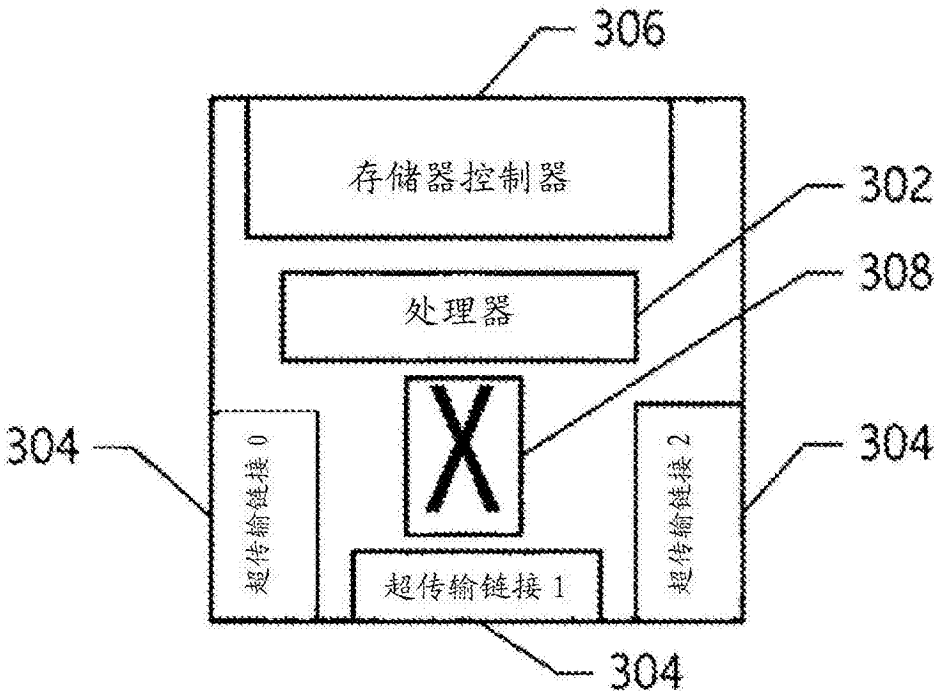

306
存储器控制器
302
处理器
308
304
超传输链接 0
304
超传输链接 1
超传输链接 2
304

图 3

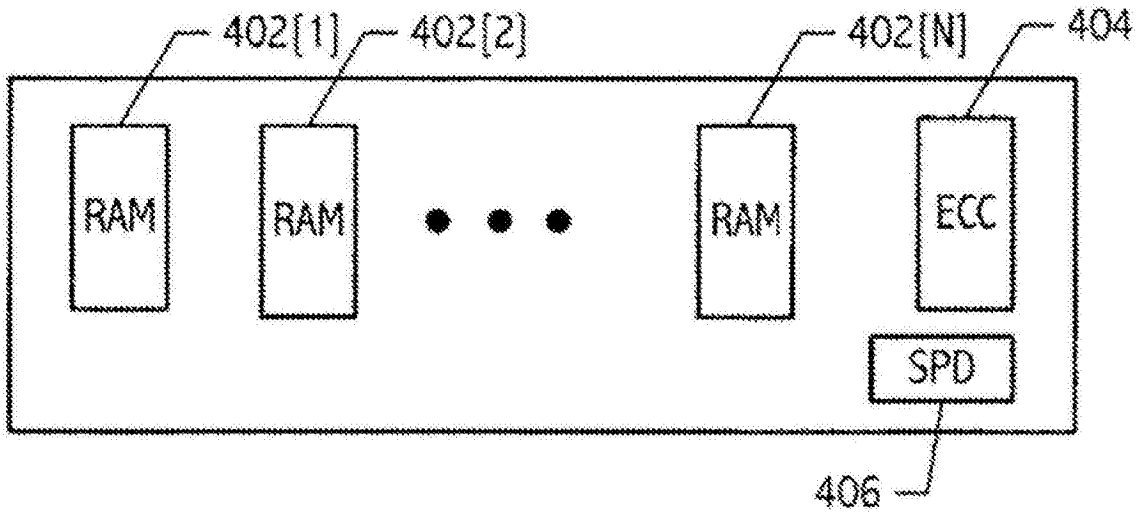

402[1]
402[2]
402[N]
404
RAM
RAM
RAM
ECC
SPD
406

图 4

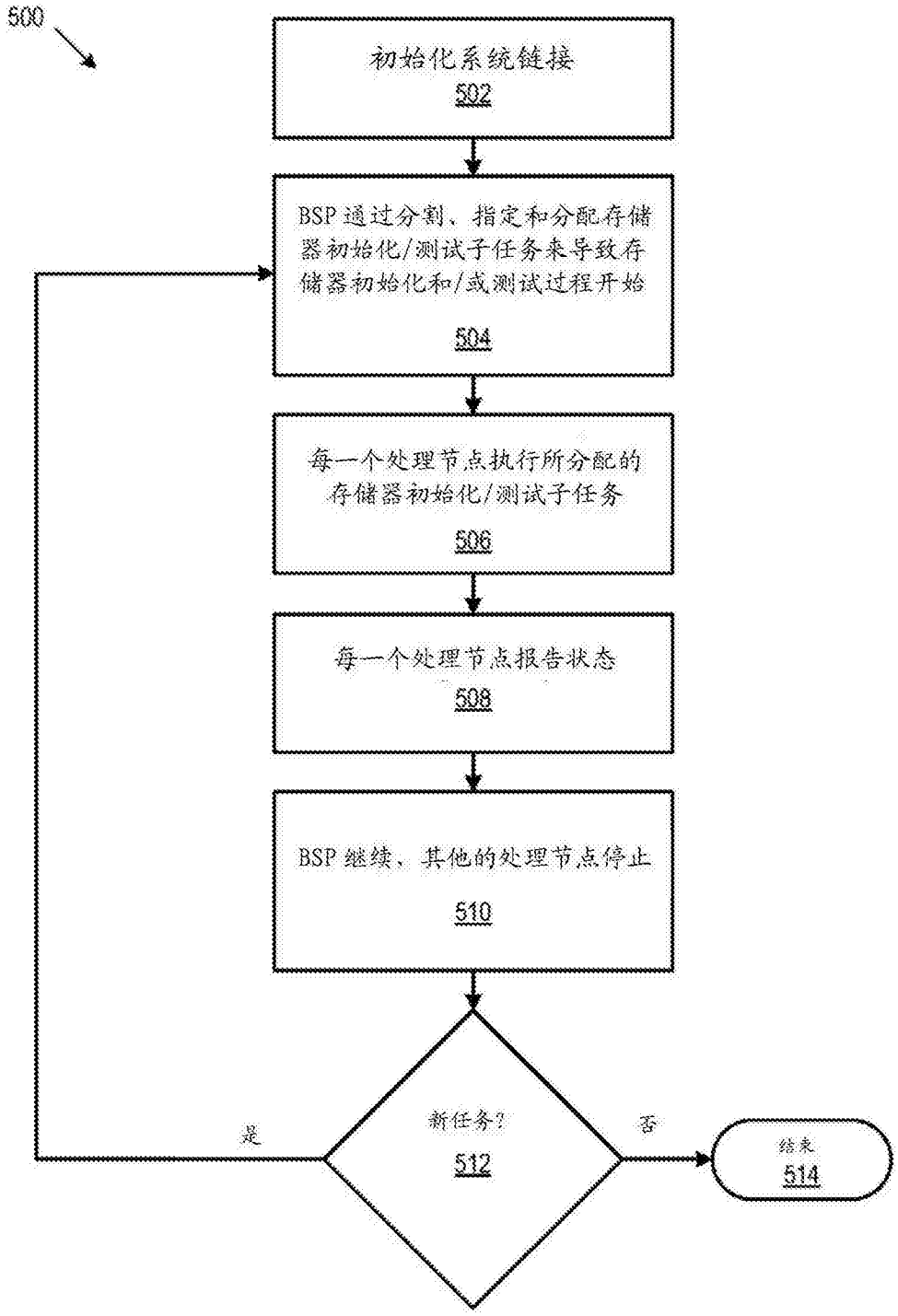

500
初始化系统链接
502
BSP 通过分割、指定和分配存储器初始化/测试子任务来导致存储器初始化和/或测试过程开始
504
每一个处理节点执行所分配的存储器初始化/测试子任务
506
每一个处理节点报告状态
508
BSP 继续、其他的处理节点停止
510
新任务?
512
是
否
结束
514

图 5