

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 11/36 (2006.01)



[12] 发明专利申请公布说明书

[21] 申请号 200780039694.8

[43] 公开日 2009年9月9日

[11] 公开号 CN 101529391A

[22] 申请日 2007.8.24

[21] 申请号 200780039694.8

[30] 优先权

[32] 2006.10.24 [33] US [31] 60/853,756

[32] 2007.5.11 [33] GB [31] 0709182.0

[86] 国际申请 PCT/GB2007/003223 2007.8.24

[87] 国际公布 WO2008/050076 英 2008.5.2

[85] 进入国家阶段日期 2009.4.24

[71] 申请人 ARM 有限公司

地址 英国剑桥郡

[72] 发明人 S·A·福德 A·D·赖德

K·E·克尼博恩

E·格林利-埃文斯

[74] 专利代理机构 中国专利代理(香港)有限公司

代理人 张雪梅 徐予红

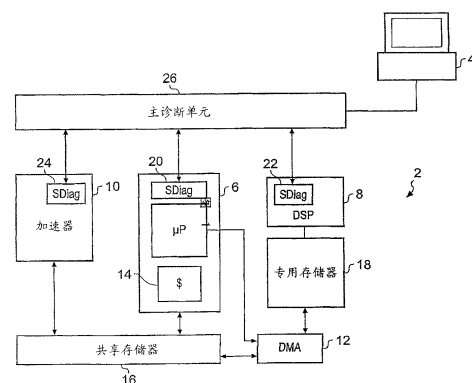
权利要求书4页 说明书7页 附图4页

[54] 发明名称

在非对称多处理器设备上执行诊断操作

[57] 摘要

提供了非对称多处理器设备(2)，其中相应的从诊断单元(20、22、24)与对应的执行机构(6、8、10)相关联。主诊断单元(26)跟踪线程执行在不同的执行机构(6、8、10)之间的迁移使得该诊断机构(20、22、24、26)能够跟随给定线程的执行并且将该信息提供给程序员。该执行机构(6、8、10)可以是多种多样的，例如通用处理器(6)、直接存储器存取单元(12)、协处理器、VLIW 处理器、数字信号处理器(8)和硬件加速器(10)。该非对称多处理器设备(2)还典型地包括非对称存储器分级体系，例如包括全局存储器、共享存储器(16)、专用存储器(18)和高速缓冲存储器(14)中的两个或更多个。



1. 一种非对称多处理器设备, 包括:

多个执行机构, 该多个执行机构响应于相应的程序指令来执行数据处理操作, 程序指令执行的线程在所述多个执行机构之间时分多路复用;

多个从诊断单元, 该多个从诊断单元中的每一个与相应的执行机构相关联以在其上执行诊断操作;

主诊断单元, 该主诊断单元耦合到所述多个从诊断单元并且对给定线程的执行在执行机构之间的迁移做出响应以对哪个所述从诊断单元当前是活动的以执行关于所述给定线程的诊断操作做出对应的变化, 使得所述主诊断跟踪哪个执行机构正在执行所述给定线程的变化。

2. 如权利要求 1 所述的非对称多处理器设备, 包括非对称存储器分级体系使得所述多个执行机构不共享公共的存储器映射。

3. 如权利要求 1 和 2 中的任一项所述的非对称多处理器设备, 其中所述多个执行机构中的至少一个执行机构具有与所述多个执行机构中的一个或多个其它的执行机构不同的处理体系架构。

4. 如权利要求 1、2 和 3 中的任一项所述的非对称多处理器设备, 其中所述执行的迁移是通过同步远程过程调用从一个执行机构到另一执行机构的。

5. 如前述权利要求中的任一项所述的非对称多处理器设备, 其中所述主诊断单元提供来自执行机构的诊断输出并且对用于所述给定线程的执行机构的变化做出响应以在所述诊断输出中进行对应的变化。

6. 如前述权利要求中的任一项所述的非对称多处理器设备, 其中所述主诊断单元将关于所述给定线程的诊断命令送至与当前正在执行所述给定线程的执行机构相关联的从诊断单元。

7. 如前述权利要求中的任一项所述的非对称多处理器设备, 其中所述主诊断单元在那些执行机构中的每一个上设置关于所述给定线程的一部分的断点, 在那些执行机构中的每一个上所述部分能够被执行。

8. 如权利要求 7 所述的非对称多处理器设备, 其中所述断点设置为具有与由相应的执行机构用于所述部分的存储器地址相匹配的存储器地址。

9. 如权利要求 7 所述的非对称多处理器设备, 其中当所述断点被触发时, 所述主诊断单元确定哪个线程已经触发了所述断点。

10. 如前述权利要求中的任一项所述的非对称多处理器设备, 包括多个存储器并且其中所述主诊断单元关于能够存储数据项的所述多个存储器中的每一个为所述数据项设置观察点。

11. 如权利要求 10 所述的非对称多处理器设备, 其中所述观察点设置为具有与相应存储器内的用于所述数据项的存储器地址相匹配的存储器地址。

12. 如权利要求 10 所述的非对称多处理器设备, 其中当所述观察点被触发时, 所述主诊断单元确定哪个线程已经触发了所述观察点。

13. 如前述权利要求中的任一项所述的非对称多处理器设备, 其中所述从诊断单元是从跟踪单元, 所述主诊断单元是主跟踪单元, 并且当形成用于所述给定线程的跟踪输出时, 所述主跟踪单元跟踪所述给定线程的执行在不同执行机构之间的迁移。

14. 如前述权利要求中的任一项所述的非对称多处理器设备, 其中所述从诊断单元是从剖析单元, 所述主诊断单元是主剖析单元, 并且当形成用于所述给定线程的剖析输出时, 所述主剖析单元跟踪所述给定线程的执行在不同执行机构之间的迁移。

15. 如前述权利要求中的任一项所述的方法, 其中所述多个执行机构包括下列项中的一个或多个:

通用处理器;

直接存储器存取单元;

协处理器;

VLIW 处理器;

数字信号处理器; 和

硬件加速器单元。

16. 如权利要求 4 所述的方法, 其中所述非对称存储器分级体系包括下列项中的两个或更多个:

能够由所有所述执行机构访问的全局共享存储器;

能够由所述执行机构中的至少两个访问的共享存储器; 和

能够由所述执行机构中的一个访问的专用存储器。

17. 一种在非对称多处理器设备上执行诊断操作的方法, 所述方法包括以下步骤:

响应于相应的程序指令，用多个执行机构执行数据处理操作，程序指令执行的线程在所述多个执行机构之间时分多路复用；

用与其相关联的相应从诊断单元在所述多个执行机构上执行诊断操作；

使用主诊断单元，该主诊断单元耦合到所述多个从诊断单元并且对给定线程的执行在执行机构之间的迁移做出响应，以对哪个所述从诊断单元当前是活动的以执行关于所述给定线程的诊断操作做出对应的变化，使得所述主诊断跟踪哪个执行机构正在执行所述给定线程的变化。

18. 如权利要求 17 所述的方法，其中所述非对称多处理设备包括非对称存储器分级体系使得所述多个执行机构不共享公共的存储器映射。

19. 如权利要求 17 和 18 中的任一项所述的方法，其中所述多个执行机构中的至少一个执行机构具有与所述多个执行机构中的一个或多个其它的执行机构不同的处理体系架构。

20. 如权利要求 17、18 和 19 中的任一项所述的方法，其中所述执行的迁移是通过同步远程过程调用从一个执行机构到另一执行机构的。

21. 如权利要求 17 到 20 中的任一项所述的方法，其中所述主诊断单元提供来自执行机构的诊断输出并且对用于所述给定线程的执行机构的变化做出响应以在所述诊断输出中做出对应的变化。

22. 如权利要求 17 到 21 中的任一项所述的方法，其中所述主诊断单元将关于所述给定线程的诊断命令送至与当前正在执行所述给定线程的执行机构相关联的从诊断单元。

23. 如权利要求 17 到 22 中的任一项所述的方法，其中所述主诊断单元在那些执行机构中的每一个上设置关于所述给定线程的一部分的断点，在那些执行机构中的每一个上所述部分能够被执行。

24. 如权利要求 23 所述的方法，其中所述断点设置为具有与由相应的执行机构用于所述部分的存储器地址相匹配的存储器地址。

25. 如权利要求 23 所述的方法，其中当所述断点被触发时，所述主诊断单元确定哪个线程已经触发了所述断点。

26. 如权利要求 17 到 25 中的任一项所述的方法，其中所述非对称多处理设备包括多个存储器并且所述主诊断单元关于能够存储数据项的所述多个存储器中的每一个为所述数据项设置观察点。

27. 如权利要求 26 所述的方法，其中所述观察点设置为具有与相应的存储器内的用于所述数据项的存储器地址相匹配的存储器地址。

28. 如权利要求 26 所述的方法，其中当所述观察点被触发时，所述主诊断单元确定哪个线程已经触发了所述观察点。

29. 如权利要求 17 到 28 中的任一项所述的方法，其中所述从诊断单元是从跟踪单元，所述主诊断单元是主跟踪单元，并且当形成用于所述给定线程的跟踪输出时，所述主跟踪单元跟踪所述给定线程的执行在不同执行机构之间的迁移。

30. 如权利要求 17 到 29 中的任一项所述的方法，其中所述从诊断单元是从剖析单元，所述主诊断单元是主剖析单元，并且当形成用于所述给定线程的剖析输出时，所述主剖析单元跟踪所述给定线程的执行在不同执行机构之间的迁移。

31. 如权利要求 17 到 30 中的任一项所述的方法，其中所述多个执行机构包括下列项中的一个或多个：

通用处理器；

直接存储器存取单元；

协处理器；

VLIW 处理器；

数字信号处理器；和

硬件加速器单元。

32. 如权利要求 20 所述的方法，其中所述非对称存储器分级体系包括下列项中的两个或更多个：

能够由所有所述执行机构访问的全局共享存储器；

能够由所述执行机构中的至少两个访问的共享存储器；和

能够由所述执行机构中的一个访问的专用存储器。

33. 一种计算机程序产品，其载有用于根据如权利要求 17 到 32 中的任一项所述的方法控制计算机以控制主诊断单元和多个从诊断单元的计算机程序。

在非对称多处理器设备上执行诊断操作

技术领域

本发明涉及数据处理系统。更具体地，本发明涉及提供非对称多处理器设备使用的诊断机构。

背景技术

例如调试机构、跟踪机构和剖析 (profiling) 机构的诊断机构在数据处理系统发展领域变得日益重要。随着数据处理系统变得更加复杂而用于它们的研制和测试的时间变得更短，就日益需要能够被用于识别与数据处理系统相关联的问题并且通过调整那些数据处理系统的设计和配置来提高它们的性能的强大且易于使用的诊断机构。

数据处理系统的另一趋势是日益增加的多处理器系统的使用。这些被用来通过允许 (典型地由程序或者任务的不同线程) 并行执行处理来提供较高的性能。这样的多处理器系统的一种形式称为对称多处理器系统。这样的对称系统典型地包括多个相同的处理器核，每个处理器核均具有相同的程序和数据存储器的相干视图 (coherent view)，并且操作系统软件负责将待执行的任务/线程分配给各个处理器并且负责在处理器之间迁移任务/线程。也就是说，程序指令执行的单一线程在操作系统控制下在多个处理器之间是时分多路复用的。在这样的对称系统内操作系统能够被用于确定各个任务/线程正在哪里执行并且将在单一处理器上执行的单一线程的外观提供给程序员，即使当操作系统调度将执行从一个处理器迁移到另一处理器时也是如此。这种方法限于相同的处理器并且迁移是通过该操作系统执行的，即不被程序本身触发的情况。

虽然对称多处理器系统能够提高处理性能，但它们在某种程度上仍是低效的。举例来说，如果待执行的处理需要一些最好由 DSP 型核执行的操作 (例如高度数字密集和重复的) 和最好由通用处理器执行的其它任务 (例如流控制、用户输入等)，那么对称多处理器是实施这样的处理的相对低效的方式。这已经是公认的，并且提供非对称多处理 (AMP) 系统是已知的。这样的系统的例子是由德克萨斯仪器公司设计的 OMAP 平台。在这样的平台内多个不同的处理器配有的每一个具有使它比其它处理器更适合某些任务的特性。举例来说，非对称多处理器可以包括 DSP

核以及通用微处理器核。虽然这样的非对称多处理器就它们能够以相对较低成本和较低功耗提供的处理性能而言具有强劲的优势，但是由于它们不同类的性质，它们更加难以编程和研制。为了充分利用这样的非对称多处理器系统，对于线程之间的任务迁移正常的是由程序本身而不是在操作系统控制下执行。此外，不同处理器的不同处理体系架构意味着显著不同的诊断机构可以适用于这些处理器中的每一个。AMP 系统中同步的远程过程调用能够被视为与 SMP 系统中操作系统控制的线程迁移类似的从一个处理器到另一个处理器的程序控制的线程迁移，但这种简单的系统视图不被现有诊断机构支持，现有诊断机构是根据哪个处理器执行这些操作来划分系统的。这具有这样的结果：与对称多处理系统相关联的相对简单的诊断技术不能容易地用于非对称多处理器系统中，即使对这样的非对称处理系统进行编程的复杂度和难度意味着诊断机构更加重要。在传统的 AMP 系统中，在每个处理器上运行的代码被分开处理。用于每个处理器的代码被调试，就好像它是与其它处理器上的单独的程序通信的单独程序。

发明内容

从本发明的一个方面来看，提供一种非对称多处理器设备，包括：

多个执行机构，该多个执行机构响应于相应的程序指令来执行数据处理操作，程序指令执行的线程（thread of program instruction execution）在所述多个执行机构之间是时分多路复用的；

多个从诊断单元，该多个从诊断单元中的每一个与相应的执行机构相关联以在其上执行诊断操作；

主诊断单元，该主诊断单元耦合到所述多个从诊断单元并且对给定线程的执行在执行机构之间的迁移做出响应以对哪个所述从诊断单元当前是活动的以执行关于所述给定线程的诊断操作做出对应的变化，使得所述主诊断跟踪哪个执行机构正在执行所述给定线程的变化。

本技术提供了这样的结构，其中非对称多处理器内的相应执行机构配备有它们自己的从诊断单元。这些从诊断单元耦合到负责跟踪线程在执行单元之间的迁移的主诊断单元并且从而使得能够提供给定线程的执行的单一视图，即使当这在非对称多处理器系统内的不同执行机构之间划分开的时候也是如此。这显著地减轻了识别和理解在进行编程配置时出现的问题的任务，这是因为执行的各个线程可以在这样的不同类环境之间被跟

踪，例如在 AMP 系统中的处理器之间划分的代码能够被作为单一实体进行调试。

理解这样的非对称多处理器系统的操作的复杂度和困难是复杂的，并且相应地，在包括非对称存储器分级体系使得该多个执行机构不共享公共的存储器映射的系统中本技术的优势被增强。在具有这样的非对称存储器分级体系的系统内，由不同执行机构使用的不同存储器映射的复杂度使得程序员更加难以理解使用与单独的执行机构相关联的单独诊断单元，因为相同的数据项可能被不同的执行机构以不同的方式引用。本技术的能够跟踪线程变化的主诊断单元能够为程序员考虑这些存储器映射的差异从而显著减轻程序员的任务。

虽然非对称多处理器可以由于具有非对称的存储器分级体系而是非对称的，它还可以由于至少一些在它们的处理体系结构方面有差异的执行机构而是非对称的。通常是这种情况，使得不同的处理器能够针对于不同类型的任务上并且本技术在帮助程序员处理这样的复杂度方面是有用的。

如上所述，本技术能够处理这样的系统，其中程序本身控制任务在执行机构之间的迁移。在没有本技术的情况下从诊断意义上来说处理任务在执行机构之间的迁移将是困难的，达到这个目的的特别有效的方法是从一个执行机构到另一执行机构的同步远程过程调用启动线程执行的迁移。在这种情况下该迁移正在明确地由程序执行，并且该远程过程调用可以由例程库中的一个例程执行，所述例程库能够在被调用者执行机构上执行并且针对 (target) 不同的目的地执行机构。

当主诊断单元提供来自自由执行机构执行的给定线程的诊断输出并且对于该线程的执行机构的变化做出响应以在该诊断输出中进行对应的变化使得程序员能够认识到执行机构的迁移已经发生的时候，本技术的可用性得到了增强。常常存在这样的情况，程序的问题是迁移相关联的并且相应地该信息对程序员是有用的。

与该不同执行机构相关联的从诊断单元可能需要编程以在执行的特定线程上执行诊断操作。主诊断单元已经在由不同执行机构执行线程时跟踪了该线程，故该主诊断单元能够在相关的时间将关于给定线程的诊断命令送至与该给定线程相关联的从诊断单元。

该诊断单元能够采取多种不同的形式。例如，它们可以是断点单元、观察点单元、跟踪单元和/或剖析单元。这些诊断单元能够由纯硬件、硬

件和软件的混合或者纯软件提供。一些诊断单元更适合以硬件实施，例如需要特别高的带宽以便跟踪处理器的实时操作的跟踪单元。

将理解该执行机构可以采取如以上讨论的多种不同的形式。例如，该执行机构可以包括通用处理器、直接存储器存取单元、协处理器、VLIW处理器、数字信号处理器和/或硬件加速器单元中的一个或多个。这些除通用处理器以外的各种形式的执行机构能够被认为是各种不同形式的专用处理器。

上述讨论的非对称存储器分级体系类似地能够以各种不同的方式形成。提供的存储器可以是高速缓冲存储器、专用存储器、共享存储器或者全局存储器。这些存储器将以非对称的形式混合在一起，这对于最终的实施和非对称多处理器设备的使用来说可能是高效的，但是它增加了在系统上执行诊断性操作的难度。将理解，并行系统典型地包含多个线程，每个线程可以在该多个执行机构之间时分多路复用。

从本发明的另一方面来看，提供了一种在非对称多处理器设备上执行诊断操作的方法，所述方法包括步骤：

响应于相应的程序指令，用多个执行机构执行数据处理操作，程序指令执行的线程在所述多个执行机构之间是时分多路复用的；

用与其相关联的相应从诊断单元在所述多个执行机构上执行诊断操作；

使用主诊断单元，该主诊断单元耦合到所述多个从诊断单元并且对给定线程的执行在执行机构之间的迁移做出响应，以对哪个所述从诊断单元当前是活动的以执行关于所述给定线程的诊断操作做出对应的变化，使得所述主诊断跟踪哪个执行机构正在执行所述给定线程的变化。

附图说明

现在将仅以示例的形式参考附图描述本发明的实施例，其中：

图 1 示意性地示出根据本技术的一个例子的非对称多处理器设备；

图 2 示意性地示出程序内的给定线程的执行在不同执行机构之间的迁移并且由主诊断单元对这种迁移的跟踪；

图 3 是示意性地示出启动诊断操作和跟踪线程迁移的流程图；和

图 4 是示意性地示出根据一个示例性实施例的对本技术的诊断机构进行初始编程的流程图。

具体实施方式

图 1 示意性地示出连接到通用计算机 4 的非对称多处理器设备 2。该通用计算机用来编程和控制诊断操作。该非对称多处理器设备 2 包括形式为通用处理器 6、数字信号处理器 8、加速器 10 和直接存储器存取 (DMA) 单元 12 的多个执行机构。将理解可以在非对称多处理器设备内提供执行机构的不同组合, 例如 VLIW 处理器和/或协处理器。此外, 这样的非对称多处理器设备可以包括多个给定类型 (它们可以是相同的) 的执行机构以及一个或多个不同类型的执行机构, 例如具有多个 DSP 单元的单一通用处理器。

图 1 的非对称多处理器设备 2 具有非对称存储器分级体系, 该存储器分级体系包括高速缓冲存储器 14、共享存储器 16 和专用存储器 18, 高速缓冲存储器 14 对于通用处理器 6 来说是本地的, 共享存储器 16 可由加速器 10、通用处理器 6 和 DMA 单元 12 访问。专用存储器 18 仅可由 DSP 单元 8 和 DMA 单元 12 访问。DMA 单元 12 由通用处理器 6 控制, 通用处理器 6 建立由 DMA 单元 12 执行的 DMA 操作。这些将典型地包括在专用存储器 18 和共享存储器 16 之间传输数据。将理解, 使用这样的布置, 输入数据块可以在通用处理器 6 的控制下从共享存储器 16 传输到专用存储器 18 内, 在专用存储器 18 中, 输入数据块能够经受 DSP 单元 8 的密集数值处理, 之后该处理的结果由通用处理器 6 从专用存储器 18 恢复到共享存储器 16 中。由 DSP 单元 8 执行的处理是同一程序线程的一部分, 该程序线程也至少部分地运行在通用处理器 6 上, 并且当到达需要由 DSP 单元 8 进行处理的该线程内的点时, 则从通用处理器 6 对 DSP 单元 8 进行同步远程过程调用以启动 DSP 单元 8 上的处理。在 DSP 单元 8 处理结束时, 对其它地方进行另外的同步远程过程调用或者可以简单地从该远程过程调用返回。

如图 1 所示, 执行机构 6、8 和 10 中的每一个包括相应的从诊断单元 20、22 和 24。DMA 单元 12 不具有自己的从诊断单元, 因为其仅仅在通用处理器 6 的控制下操作。将理解, 由不同的执行机构 6、8、10 提供的不同的处理器体系架构将意味着从诊断单元 20、22 和 24 具有不同的形式和能力以及需要不同类型的编程, 并且产生不同形式的输出。非对称多处理器设备 2 上的这种多样性通常给试图理解在不同执行机构 6、8、10 之间迁移的给定线程的执行的程序员提出了难题。

除了该多个从诊断单元 20、22、24 之外, 还提供了耦合到从诊断单元 20、22、24 中的每一个的主诊断单元 26。该主诊断单元经由从诊断单

元 20、22、24 来监控程序线程迁移，例如通过识别同步远程过程调用和返回来监控。主诊断单元 26 连接到通用计算机 4，该通用计算机 4 用来对待执行的诊断操作编程以及监控诊断输出并且将该诊断输出显示给程序员。主诊断单元 26 能够作为硬件部件、作为混合的硬件和软件部件或者作为在通用计算机 4 上执行的纯软件部件来提供。

从诊断单元 20、22 和 24 被示出为提供在它们各自的执行机构 6、8、10 中的每一个内，但是将理解从诊断单元 20、22、24 能够分别提供在相关的执行机构 6、8、10 的外部。从诊断单元能够在多个执行机构之间共享，例如监控写入共享存储器 16 或者从共享存储器 16 读取的数据的跟踪单元将执行对多于一个的执行机构 6、8、10 的跟踪。然而，这样的跟踪单元将与至少一个执行机构相关联，因为它执行对至少一个执行机构的跟踪。

图 2 示意性地示出给定的程序线程的执行在不同的执行机构之间的迁移。在该图示中，存在三种不同的执行机构。这些可以被认为是对应于图 1 的通用处理器 6、DSP 单元 8 和加速器 10，不过到不同执行机构的其它映射也将是同样可能的。如图所示的，程序线程在执行机构 1 上开始执行，并且然后经由同步远程过程调用被迁移到执行机构 2。在执行机构 2 上执行了一段时间以后返回到执行机构 1。随后迁移到如图所示的执行机构 2 和 3。在此期间，主诊断单元 26 监控在不同执行机构之间切换程序线程的执行的同步远程过程调用的发生，并且相应地跟踪在处理该程序线程中当前活动的是哪个执行机构。这就是在图 2 左手侧示出的跟踪信息。该跟踪信息被主诊断单元 26 使用，以当其在通用计算机 4 上被显示并且与程序员交互时，引导 (channel) 从不同的从诊断单元 20、22、24 取回的诊断信息来结合适当的程序线程显示。诊断信息显示包括识别对关心的程序中的点来说当前是活动的执行机构。举例来说，如果当线程在执行机构 2 上执行时，程序员希望停止 (halt) 该线程，则当主诊断单元 26 确定在处理该线程中执行机构 2 当前是活动的时，该停止命令将被传递到执行机构 2。

图 3 是示意性地示出由主诊断单元 26 执行的线程跟踪类型的流程图。在步骤 28，启动在执行机构 6、8、10 上处理不同的线程并且相关的从诊断单元 20、22、24 使用它们的诊断参数 (例如，断点、观察点、剖析参数、跟踪参数等) 来编程。在步骤 30，主诊断单元 26 识别哪些线程在哪些执行

机构上执行，并且该数据由主诊断单元 26 收集并保持在例如线程数据表内。将理解，哪个执行机构执行哪个特定线程可能不是由程序本身预先确定的，因为它可以是在运行时间根据负荷及其它因素而试探确定的。

在步骤 32，主诊断单元 26 监控以例如借助于同步远程过程调用的发布检查是否已经发生任何线程迁移。当检测到线程迁移时，处理然后进行到步骤 34，在步骤 34，指示哪个线程正在哪个执行机构上执行的数据被更新以考虑到不再执行线程的调用者和正在执行该线程的被调用者。

在步骤 36，(如果需要的话)改变对从诊断单元 20、22 和 24 的编程以考虑已经发生的迁移。例如，如果调用者执行单元正在监控关于线程的特定断点或者观察点，那么当该执行机构不再执行该线程时，继续监控那些断点和观察点则是不合适的，原因是由于那些断点或者观察点与响应于不同的程序线程而发生的执行相互混淆而可能给出错误的结果。在步骤 36 对从诊断单元进行任何必需的重新编程之后，处理返回到步骤 32，在步骤 32，可以执行对另外的线程迁移的监控。

图 4 示意性地示出在图 1 的系统内诊断操作的建立。在步骤 38，程序员使用通用计算机 4 指示要执行给定的诊断命令，并且该命令被发送到主诊断单元 26，在步骤 38，在主诊断单元 26 处接收该命令。步骤 40 对应于主诊断单元 26 确定哪个执行机构(或者在观察点的情况下，哪个存储器)与关于其将执行该诊断命令的线程相关联。步骤 42 然后为涉及已经接收的命令的线程编程/初始化相关的从诊断单元。

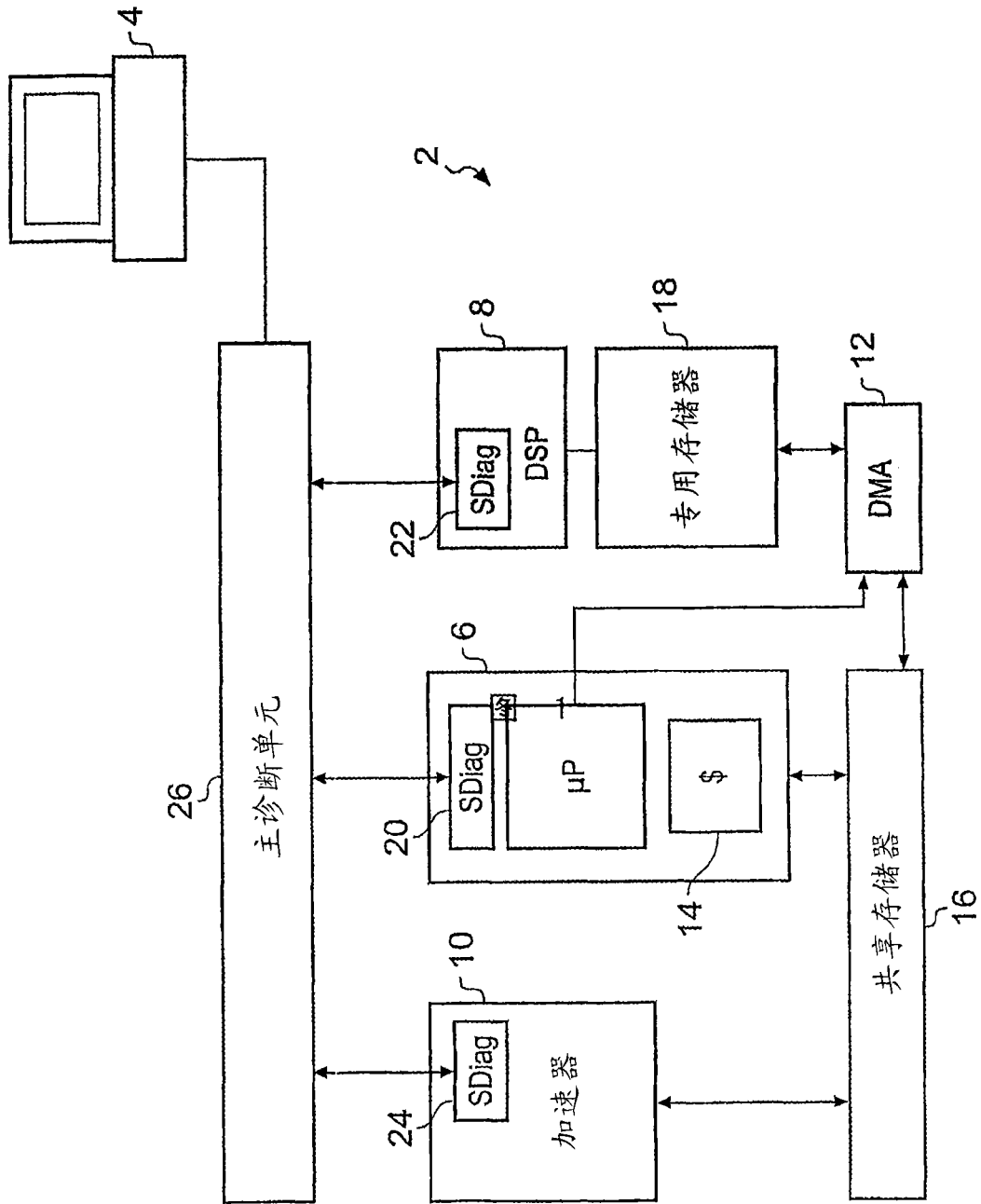
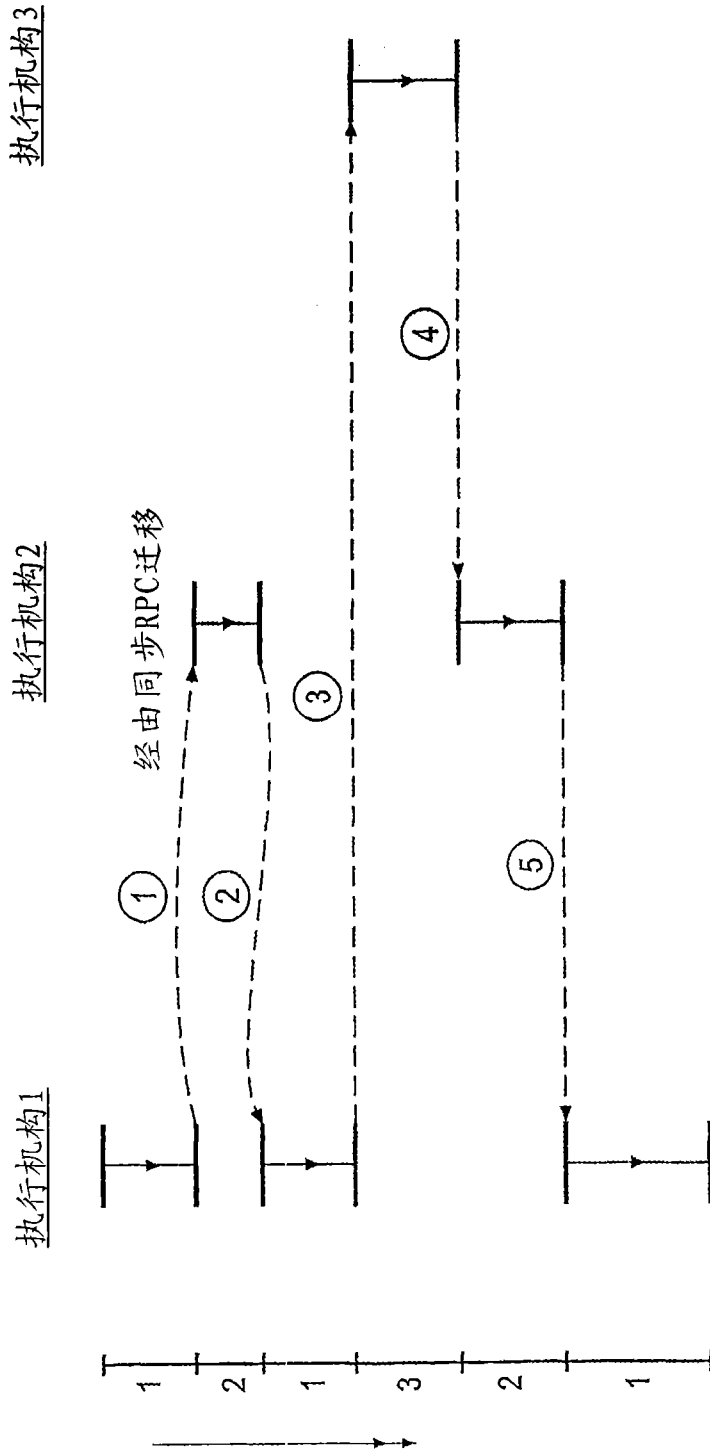


图 1



执行机构
生成线程序
的診斷数据

图 2

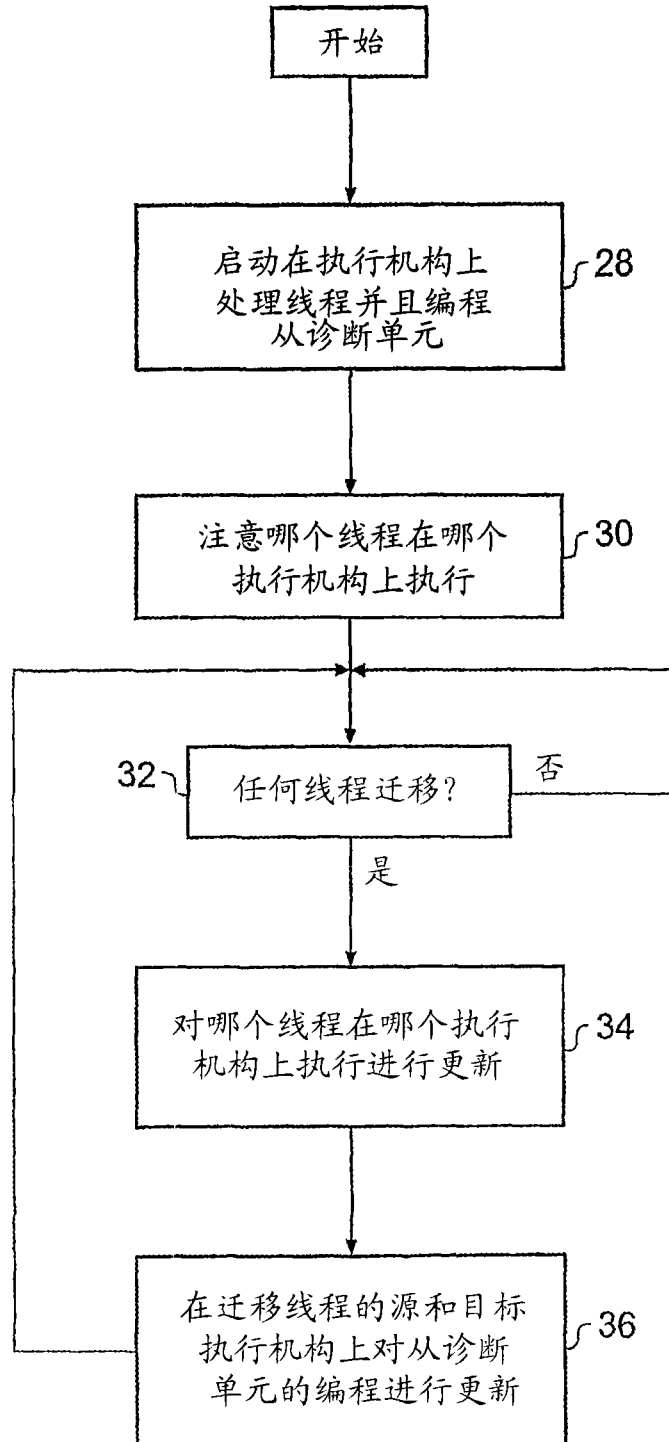


图 3

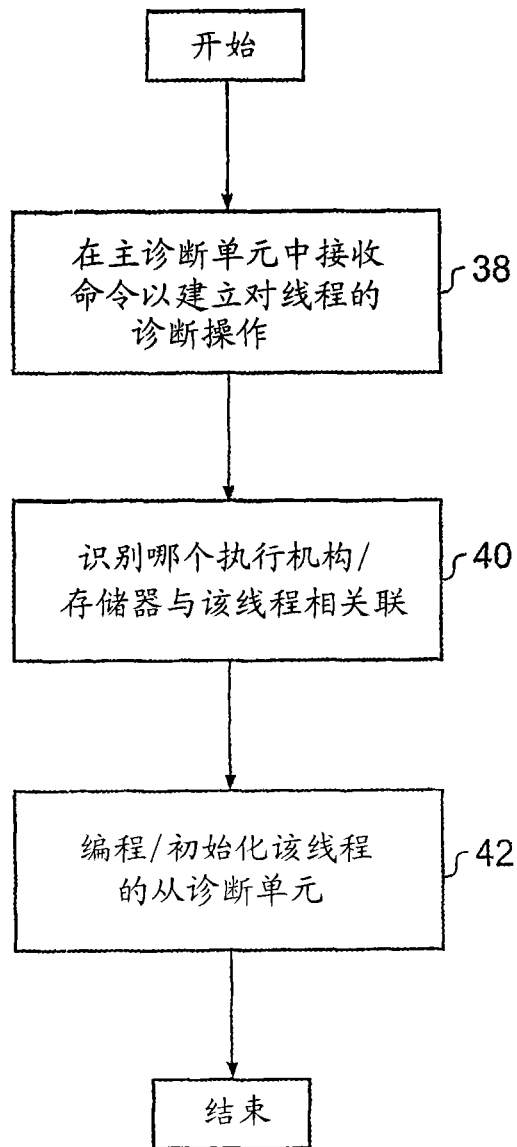


图 4