



(12) 发明专利

(10) 授权公告号 CN 101031887 B

(45) 授权公告日 2010.05.26

(21) 申请号 200580033073.X

(22) 申请日 2005.09.29

(30) 优先权数据

10/953,887 2004.09.29 US

(85) PCT申请进入国家阶段日

2007.03.29

(86) PCT申请的申请数据

PCT/US2005/035375 2005.09.29

(87) PCT申请的公布数据

W02006/039595 EN 2006.04.13

(73) 专利权人 英特尔公司

地址 美国加利福尼亚州

(72) 发明人 S·穆克赫吉 J·埃默

S·赖因哈德特 C·韦弗

(74) 专利代理机构 中国专利代理(香港)有限公司
72001

代理人 李静岚 梁永

(51) Int. Cl.

G06F 11/00 (2006.01)

G06F 11/16 (2006.01)

(56) 对比文件

US 20010034824 A1, 2001.10.25, 说明书
0037-0040 段, 权利要求书.

CN 1434941 A, 2003.08.06, 全文.

US 20010037445 A1, 2001.11.01, 全文.

Reinhardt S K et. Transient

fault detection via simultaneous
multithreading. 27th Annual International
Symposium on Computer Architecture 28
2. 2000, 28(2), 25-36.

Vijaykumar, T.N. Pomeranz, I.

Cheng, K. Transient-Fault Recovery Using
Simultaneous Multithreading. 29th Annual
International Symposium on Computer
Architecture. 2002, 2987-98.

审查员 刘清泉

权利要求书 2 页 说明书 4 页 附图 5 页

(54) 发明名称

在冗余多线程环境中执行检验指令的方法、
设备和系统

(57) 摘要

描述了一种用于在冗余多线程环境中的检验
指令的方法和设备。在一个实施例中,当 RMT 需要
时,处理器会在前导线程和结尾线程中发出检验
指令。该检验指令会对于每个线程单独地向下传
输到各个流水线中,直到它到达每个流水线末端
的缓冲器。然后,在提交该检验指令之前,该检验
指令查找其相应部分并对该指令进行比较。如果
该检验指令匹配,则提交该检验指令并退出,否则
描述错误。

1. 一种用于在冗余多线程环境中执行检验指令的方法,包括:
在前导线程和结尾线程中生成检验指令;
等待来自该前导线程和结尾线程的同等检验指令;和
比较来自该前导线程和结尾线程的同等检验指令,其中前导线程包括检验指令和选择的指令,结尾线程包括检验指令。
2. 如权利要求1所述的用于在冗余多线程环境中执行检验指令的方法,其中在前导线程中生成检验指令还包括在该选择的指令前插入检验指令。
3. 如权利要求1所述的用于在冗余多线程环境中执行检验指令的方法,其中该检验指令传输经过该前导线程和结尾线程的相应流水线。
4. 如权利要求1所述的用于在冗余多线程环境中执行检验指令的方法,还包括提交该检验指令。
5. 如权利要求4所述的用于在冗余多线程环境中执行检验指令的方法,还包括如果对于来自前导线程和结尾线程的相应检验指令的比较结果是匹配,则为该前导线程存储该选择的指令。
6. 如权利要求1所述的用于在冗余多线程环境中执行检验指令的方法,其中该前导线程和结尾线程由单个处理器执行。
7. 如权利要求1所述的用于在冗余多线程环境中执行检验指令的方法,其中该前导线程和结尾线程由多个处理器执行。
8. 一种用于在冗余多线程环境中执行检验指令的设备,包括:
用于执行前导线程指令的前导线程电路;
用于执行结尾线程指令的结尾线程电路;和
用于提交来自该前导线程和结尾线程的相应检验指令的提交单元,其中该前导线程指令包括检验指令和所选择的指令。
9. 如权利要求8所述的用于在冗余多线程环境中执行检验指令的设备,其中该结尾线程指令包括检验指令。
10. 如权利要求8所述的用于在冗余多线程环境中执行检验指令的设备,其中该前导线程电路和结尾线程电路包括流水线。
11. 如权利要求10所述的用于在冗余多线程环境中执行检验指令的设备,其中该前导线程和结尾线程由单个处理器执行。
12. 如权利要求10所述的用于在冗余多线程环境中执行检验指令的设备,其中该前导线程和结尾线程由多个处理器执行。
13. 如权利要求9所述的用于在冗余多线程环境中执行检验指令的设备,还包括耦合到该前导线程电路和结尾线程电路的缓冲器。
14. 如权利要求13所述的用于在冗余多线程环境中执行检验指令的设备,其中该前导线程的检验指令和结尾线程的检验指令等待缓冲器中的相应检验指令。
15. 如权利要求8所述的用于在冗余多线程环境中执行检验指令的设备,其中如果该相应的检验指令匹配,则存储该选择的指令。
16. 如权利要求15所述的用于在冗余多线程环境中执行检验指令的设备,其中如果该相应的检验指令不匹配,则提交单元产生错误。

17. 如权利要求 8 所述的用于在冗余多线程环境中执行检验指令的设备, 其中该检验指令被二进制译码器设置于该选择的指令之前。

18. 如权利要求 8 所述的用于在冗余多线程环境中执行检验指令的设备, 其中该选择的指令是存储指令。

19. 一种用于在冗余多线程环境中执行检验指令的系统, 包括:

第一处理器, 包括:

用于执行前导线程检验指令的前导线程电路;

用于执行结尾线程检验指令的结尾线程电路; 和

用于退出来自该前导线程和结尾线程的相应检验指令的退出单元, 其中该前导线程指令包括检验指令和所选择的指令;

到第二处理器的第一接口;

到输入 / 输出设备的第二接口; 和

耦合到第二接口的音频输入 / 输出设备。

20. 如权利要求 19 所述的用于在冗余多线程环境中执行检验指令的系统, 其中该结尾线程指令包括检验指令。

21. 如权利要求 19 所述的用于在冗余多线程环境中执行检验指令的系统, 其中该前导线程电路和结尾线程电路包括流水线。

22. 如权利要求 20 所述的用于在冗余多线程环境中执行检验指令的系统, 其中如果来自该前导线程和结尾线程的相应检验指令匹配, 则存储该选择的指令。

23. 如权利要求 22 所述的用于在冗余多线程环境中执行检验指令的系统, 还包括与该前导线程电路和结尾线程电路耦合的缓冲器。

24. 如权利要求 22 所述的用于在冗余多线程环境中执行检验指令的系统, 其中如果该相应的检验指令不匹配, 则该退出单元产生错误。

25. 如权利要求 19 所述的用于在冗余多线程环境中执行检验指令的系统, 其中该检验指令被二进制译码器设置于该选择的指令之前。

26. 如权利要求 19 所述的用于在冗余多线程环境中执行检验指令的系统, 其中该选择的指令是存储指令。

27. 如权利要求 19 所述的用于在冗余多线程环境中执行检验指令的系统, 其中该第一和第二接口是点对点接口。

在冗余多线程环境中执行检验指令的方法、设备和系统

技术领域

[0001] 本发明涉及在冗余多线程环境中执行检验指令的方法、设备和系统。

背景技术

[0002] 当前的冗余执行系统通常是应用自检验和硬件实现的检验电路。与检验电路相似的是将要比较来自两个线程的结果的比较指令（例如存储地址和数据）。有可能在两个线程中复制该比较指令以通过复制获得自检验的效果。

[0003] 遗憾的是，通过复制该比较指令，该体系会丧失冗余多线程（RMT）的性能优势。RMT 性能优势来自于具有充分隔开的前导（leading）和结尾（trailing）线程，以使得该前导线程能够预取高速缓冲损失和分流结尾线程的错误预测。如果该比较指令被复制，不仅需要额外的队列、导致更高的开销（overhead），而且该体系还将由于在两个方向需要同步而不能保持两个线程充分隔开。因而，需要一种能够获得更低的故障率而不会牺牲 RMT 的性能优势的指令。

发明内容

[0004] 按照本发明的第一方面，提供了一种用于在冗余多线程环境中执行检验指令的方法，包括：在前导线程和结尾线程中生成检验指令；等待来自该前导线程和结尾线程的同等检验指令；和比较来自该前导线程和结尾线程的同等检验指令，其中前导线程包括检验指令和选择的指令，结尾线程包括检验指令。

[0005] 按照本发明的第二方面，提供了一种用于在冗余多线程环境中执行检验指令的设备，包括：用于执行前导线程指令的前导线程电路；用于执行结尾线程指令的结尾线程电路；和用于提交来自该前导线程和结尾线程的相应检验指令的提交单元，其中该前导线程指令包括检验指令和所选择的指令。

[0006] 按照本发明的第三方面，提供了一种用于在冗余多线程环境中执行检验指令的系统，包括：第一处理器，包括：用于执行前导线程检验指令的前导线程电路；用于执行结尾线程检验指令的结尾线程电路；和用于退出来自该前导线程和结尾线程的相应检验指令的退出单元，其中该前导线程指令包括检验指令和所选择的指令；到第二处理器的第一接口；到输入 / 输出设备的第二接口；和耦合到第二接口的音频输入 / 输出设备。

附图说明

[0007] 根据以下对于如附图所示优选实施例的说明，本发明的各种特征将变得清楚，其中在所有附图中，相似的参考数字表示相同的部件。该附图不需要按比例绘制，而是将重点放在说明本发明的原理上。

[0008] 图 1 是多线程体系的一个实施例的框图。

[0009] 图 2 是示出了一种生成检验指令的方法的流程图。

[0010] 图 3 是示出了在任一线程中的检验指令的一个实施例的流程图。

[0011] 图 4 是检验指令的一个实施例的框图。

[0012] 图 5 是可以提供多线程处理器环境的系统的框图。

[0013] 图 6 是可以提供多线程处理器环境的可选系统的框图。

具体实施方式

[0014] 在以下说明中,为了解释而不是限制的目的,给出了特定的细节例如结构、体系、接口、技术等,以便提供对于本发明各个方面的完整理解。然而,本领域技术人员将会清楚,有了本公开的教导,就可以在除了这些特定细节之外的其他例子中实现本发明的各个方面。在某些例子中,省略了对于公知设备、电路和方法的说明,以免由于不必要的细节而使得对于本发明的描述不清楚。

[0015] 描述了用于冗余多线程环境中的检验指令的方法和设备。在以下说明中,为了解释的目的,给出了许多特定细节以便提供对于本发明的完整认识。然而,本领域技术人员将会清楚,不需要这些特定细节也可以实现本发明。

[0016] 图 1 是冗余多线程体系的一个实施例的框图。在冗余多线程体系中,可以通过作为单独的线程执行一个程序的两个副本而检测故障。

[0017] 每个线程具有相同的输入,比较其输出以判断是否发生错误。可以相对于这里被称为“复制范围 (sphere of replication)”的概念来描述冗余多线程。复制范围是逻辑或物理冗余操作的边界。

[0018] 在复制范围 100 内的部件 (例如执行前导线程 105 的处理器和执行结尾线程 110 的处理器) 进行冗余操作。相反,在复制范围 100 之外的部件 (例如存储器 115) 不进行冗余操作。通过其他技术例如用于存储器 115 的错误校正码来提供故障保护。其他设备可以位于复制范围 100 外部,和 / 或可以使用其他技术来提供对于复制范围 100 之外的设备的故障保护。

[0019] 进入复制范围 100 的数据通过输入复制代理 (agent) 120 进入,该输入复制代理 120 复制该数据并且将该数据的副本发送到前导线程 105 和结尾线程 110。类似地,从复制范围 100 输出的数据通过输出比较代理 125 输出,该输出比较代理比较该数据并判断是否发生错误。改变复制范围 100 的边界导致性能与硬件数量的权衡。例如,复制存储器 115 将通过避免存储指令的输出比较而允许更快地访问存储器,但是将会由于使得系统的存储器数量加倍而增加系统成本。

[0020] 本发明的一个实施例提供了一种用于检验在 RMT 软件实施中的检验电路的机构。因为 RMT 比较所提交的指令的输出 (需要逐条指令地进行比较),所以它也可以以软件形式实现。如果该 RMT 的软件实现比较了每条指令,那么将导致相当数量的开销。然而,相反,RMT 允许仅仅比较存储指令和仅仅复制加载指令,这将显著减少 RMT 实施方案的软件开销。

[0021] 图 2 示出了一种生成检验指令的方法。起初,与多数计算机相同,编译器生成指令。根据该编译器,计算机现在具有一个二进制程序,其可以是但不限于存储指令序列 200。然后,二进制译码器可以在该二进制程序的每个存储指令前插入一个检验指令 205。该二进制译码器可以是本领域公知的任何二进制译码器。在译码该二进制程序的基础上,系统为前导线程和结尾线程创建二进制程序。用于前导线程的二进制程序向存储指令中添加检验指令 210。用于结尾线程的二进制程序用前导线程的同等 (peer) 检验指令替换该存储指令

215。

[0022] 图 3 示出了检验指令的一个实施方案。当 RMT 需要比较时,处理器会在前导线程和结尾线程中发出检验指令 300。每个检验指令可以承载来自每个线程的 64 位数量。该检验指令可以对于每个线程独立地向下传播到单独的流水线中 305,直到它到达每个流水线末端的缓冲器。该检验指令在该缓冲器中等待它的同等检验指令 310。然后,这两个检验指令会比较它们承载的 64 位数量 315。当不匹配时,二者都会报告错误。当匹配时,它们会使得处理器提交检验指令 320。该流水线可以来自 CMP 中的不同处理器或者来自与 SMT 处理器中相同的多线程处理器。

[0023] 在这个实施方案中,该检验指令不阻止前导指令处理来自该指令对列的指令。相反,它只是阻止后退指针直到来自结尾线程的相应检验指令显露出来。而且,如果该环境不是 RMT 环境,那么该检验指令可以被视为 NOP。

[0024] 图 4 是经过两个流水线的检验指令的一个示例的框图。假定一个存储指令为: $R1 \rightarrow [R2]$,将寄存器 R1 中的值存储到由寄存器 R2 中的地址指示的存储器位置中。该存储指令可以利用检验指令而被复制到前导线程和结尾线程中。前导线程中的存储指令可以包括该检验指令和存储指令,如下所示:

[0025] Checkerinst R1

[0026] Checkerinst R2

[0027] 存储: $R1 \rightarrow [R2]$ 。因而,当传播经过其流水线 400 时,该前导线程可以包含该检验指令以及存储指令。

[0028] 结尾线程中的存储指令可以仅包括该检验指令,如下所示:

[0029] Checkerinst R1

[0030] Checkerinst R2。因而,当传播经过其流水线 405 时,结尾线程不具有该存储指令。

[0031] 来自流水线 400 中的前导线程的 checkerinst R1 等待缓冲器 410 中的同等检验指令。来自流水线 405 中的结尾线程的 checkerinst R1 等待缓冲器 415 中的同等检验指令。该检验指令总是寻找或等待其相应部分或同等物。如果存在镜像线程 (mirror thread),那么该检验指令将寻找或等待缓冲器 410、405 中的线程,以确定存在该镜像线程并且比较该检验指令。

[0032] 来自前导线程的 checkerinst R1 和来自结尾线程的 checkerinst R1 可以由于提交顺序而配对,并且比较寄存器区分符和 R1 的值以确保寄存器中不发生错误。如果没有发现错误,那么提交该检验指令 420。一旦该检验指令提交,就存储 R1 的值。R1 的值被传输到提交点并存储。从而,该系统能够同步检验所有存储而不是如前所作的逐个检验。

[0033] 图 5 是一个能够提供多线程处理器环境的系统的框图。图 5 所示的系统用于表示系统范围。可选的系统可以包括更多、更少和 / 或不同的部件。

[0034] 系统 500 包括总线 510 或其他通信设备以传输信息,和耦合到总线 510 的处理器 520 以处理信息。系统 500 还包括随机存取存储器 (RAM) 或其他动态存储器和静态存储器,例如硬盘或其他存储设备 535 (称为存储器),通过存储器控制器 530 耦合到总线 510 以存储用于被处理器 520 执行的信息和指令。存储器 535 还可以用于存储处理器执行指令期间的临时变量或其他中间信息。存储器控制器 530 可以包括一个或多个部件以控制一种或多种存储器和 / 或相关的存储设备。系统 500 还包括耦合到总线 510 的只读存储器 (ROM) 和

/ 或其他静态存储设备 540 以存储用于处理器 520 的静态信息和指令。

[0035] 系统 500 还可以通过总线 510 耦合到输入 / 输出 (I/O) 接口 550。I/O 接口 550 提供到 I/O 设备 555 的接口,其可以包括例如用于向计算机用户显示信息的阴极射线管 (CRT) 或液晶显示器 (LCD),包括字母数字和其他键的字母数字输入设备,和 / 或光标控制设备例如鼠标、轨迹球或光标方向键。系统 500 还包括用于提供对网络的访问的网络接口 560,该网络例如是局域网,有线或无线的。

[0036] 指令被从存储设备通过有线或无线的远程连接等 (例如通过网络接口 860 而经网络) 提供到存储器 535,该存储设备例如是磁盘、只读存储器 (ROM) 集成电路、CD-ROM、DVD。

[0037] 现在参照图 6,系统 600 概括示出了一个系统,其中处理器、存储器和输入 / 输出设备由多个点对点接口互连。系统 600 还可以包括多个处理器,其中为了清楚起见,仅显示了两个处理器 605、610。处理器 605、610 每个可以包括本地存储器控制集线器 (MCH) 615、620 以与存储器 625、630 连接。处理器 605、610 可以使用点对点接口电路 640、645 通过点对点接口 635 交换数据。处理器 605、610 每个可以使用点对点接口电路 665、670、675、680 通过各个点对点接口 655、660 与芯片组 650 交换数据。芯片组 650 还可以通过高性能图形接口 690 与高性能图形电路 685 交换数据。

[0038] 芯片组 650 可以通过总线接口 695 与总线 616 交换数据。在任一系统中,在总线 616 上可以有各种输入 / 输出 I/O 设备 614,在一些实施例中包括低性能图形控制器、视频控制器和网络控制器。在一些实施例中,另一个总线桥 618 可以用于允许总线 616 和总线 620 之间的数据交换。在一些实施例中,总线 620 可以是小型计算机系统接口 (SCSI) 总线、集成驱动电子设备 (IDE) 总线、或通用串行总线 (USB)。其他的 I/O 设备可以与总线 620 连接。这些可以包括键盘和光标控制设备 622,包括鼠标、音频 I/O 624、通信设备 626,包括调制解调器和网络接口、和数据存储设备 628。软件代码 630 可以存储在数据存储设备 628 上。在一些实施例中,数据存储设备 628 可以是固定磁盘、软盘驱动器、光盘驱动器、磁光盘驱动器、磁带、或非易失性存储器包括闪存。

[0039] 在整个说明书中,术语“指令”一般用于表示指令、宏命令、命令包或用于编码处理器操作的多种其他机制的任意一种。

[0040] 在以下描述中,为了解释和非限制性的目的,给出了具体的细节,例如特定结构、架构、接口、技术等,以便提供对于本发明各方面的完整理解。然而,获得本公开的教导的本领域技术人员将会清楚,可以在与这些具体细节不同的其他示例中实施本发明的各个方面。在某些例子中,省略了对于公知设备、电路和方法的说明,以免让不必要的细节模糊了对于本发明的说明。

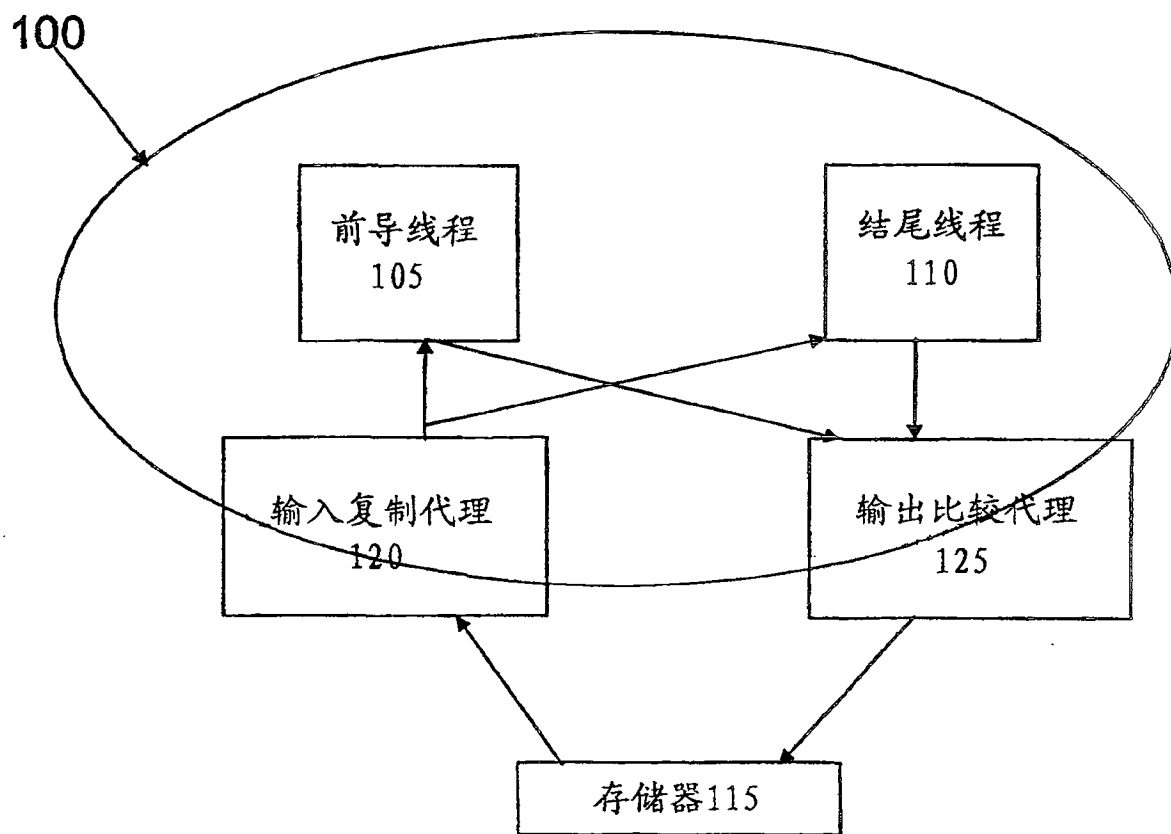


图 1

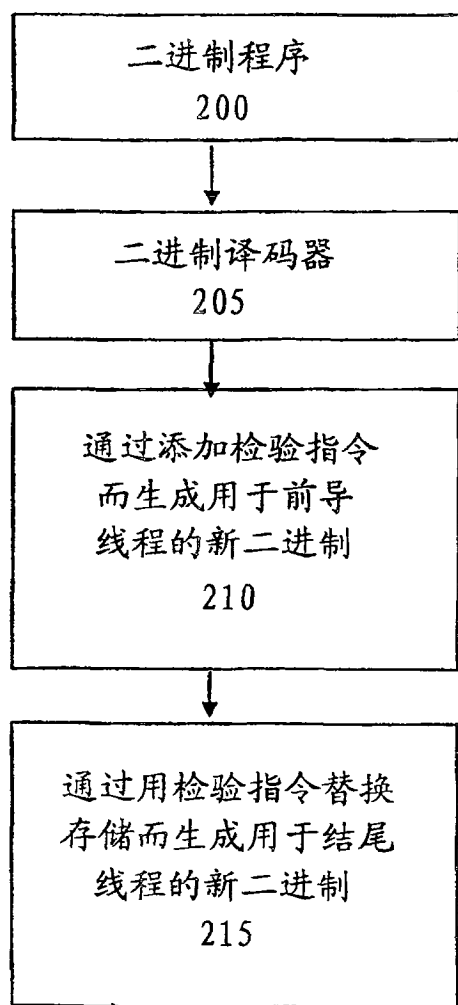


图 2

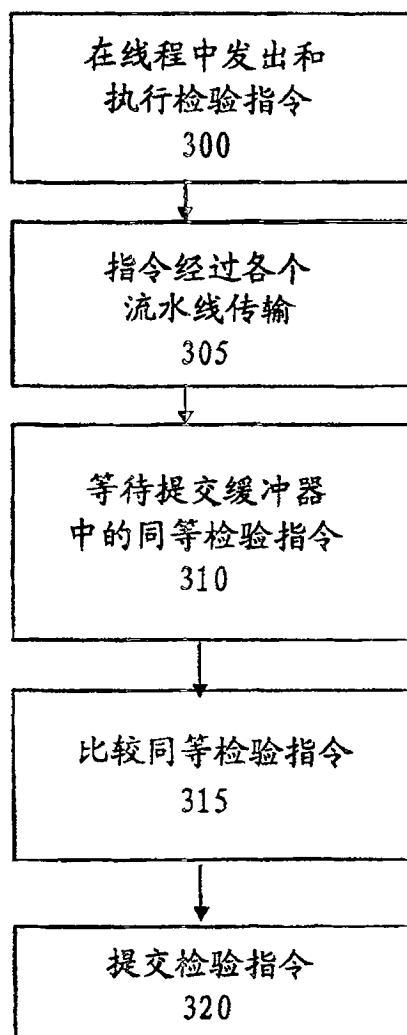


图 3

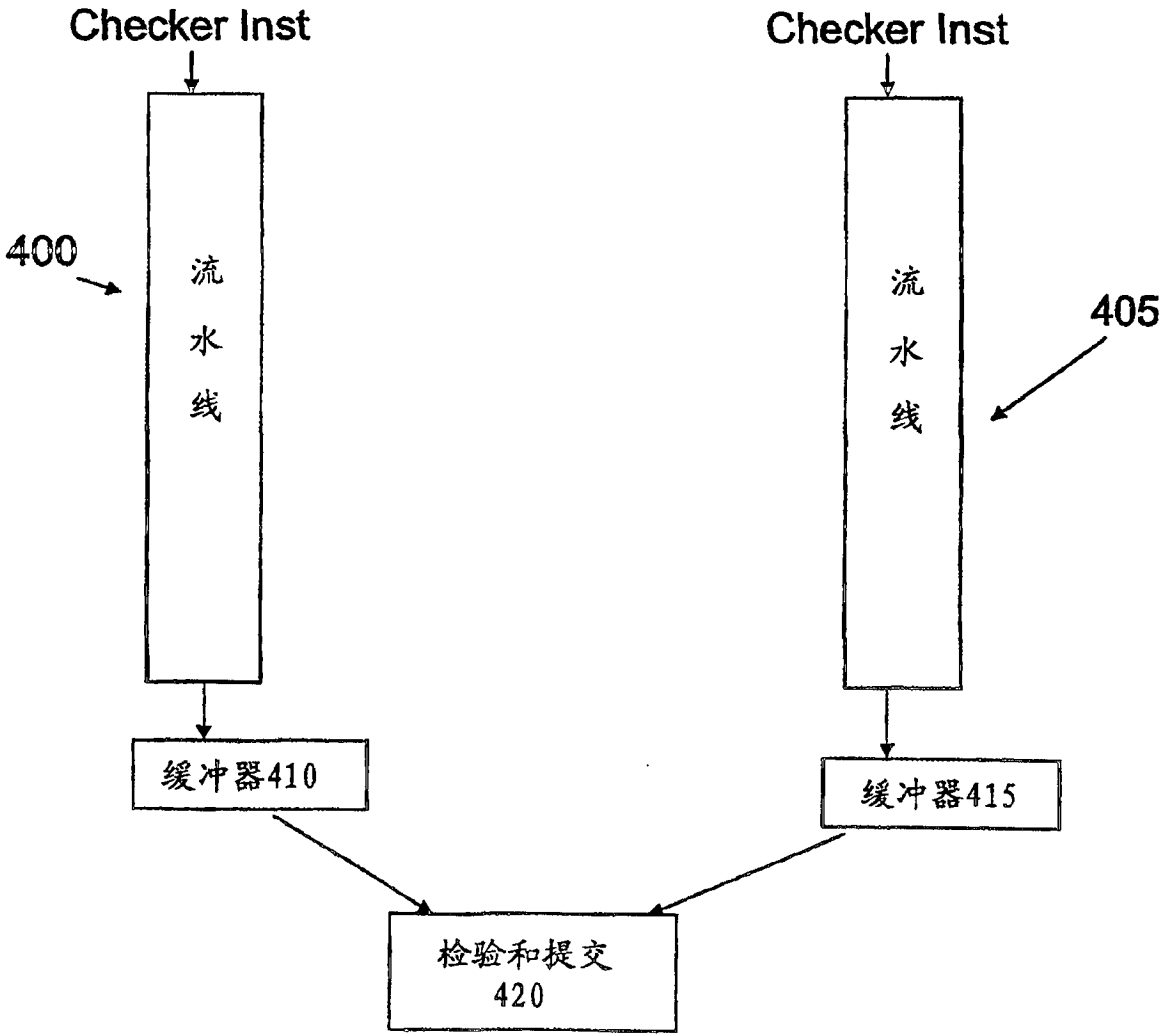


图 4

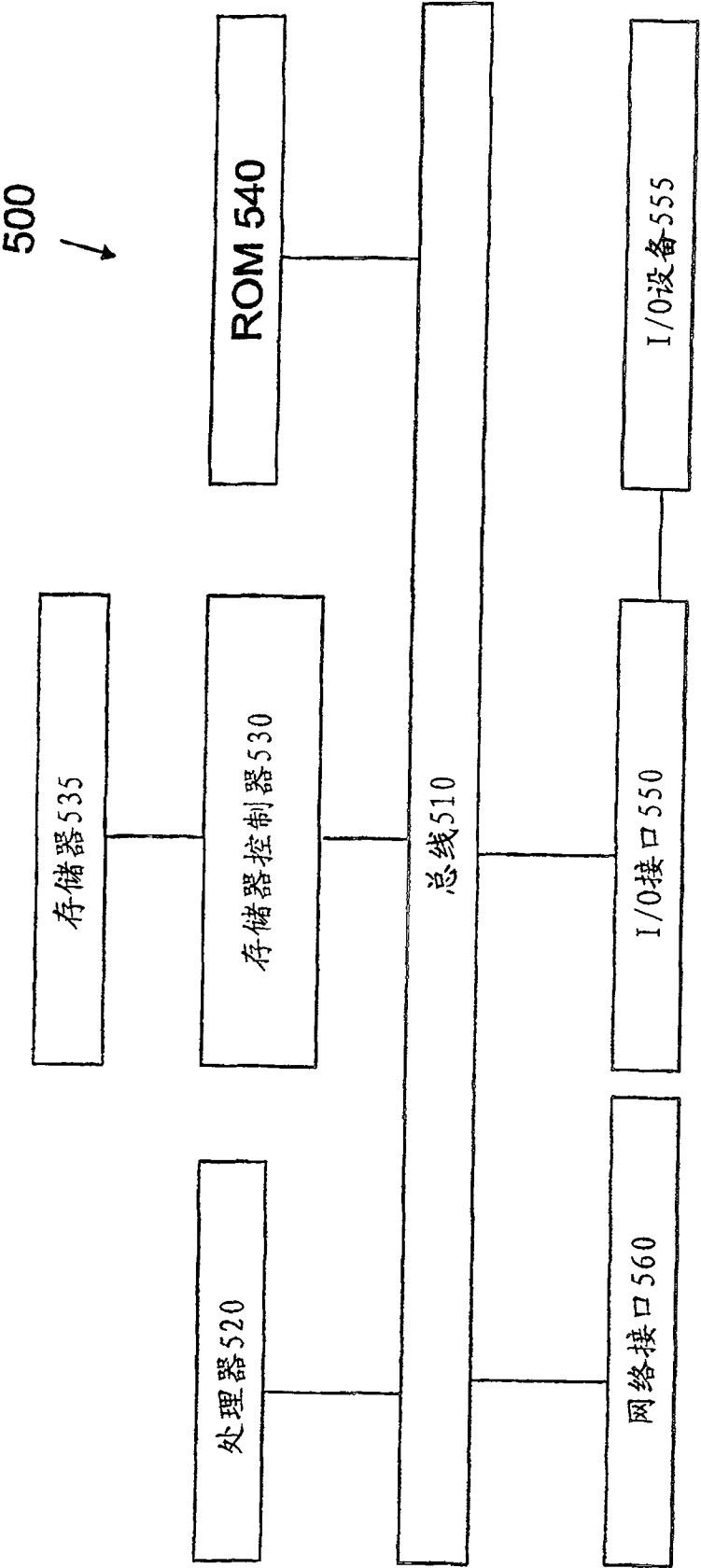


图 5

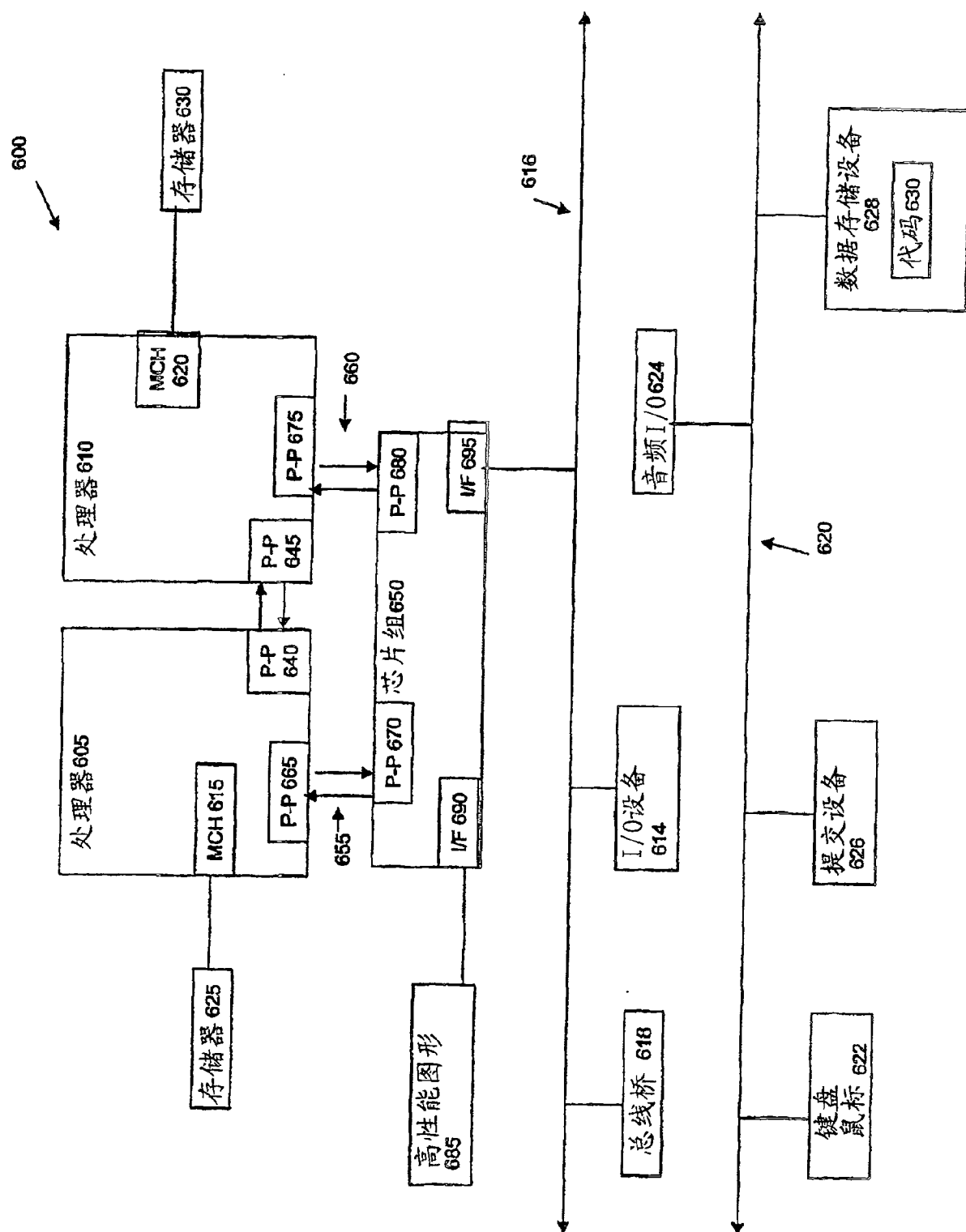


图 6