

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 9/46 (2006.01)



[12] 发明专利申请公开说明书

[21] 申请号 200510135238.6

[43] 公开日 2006年7月5日

[11] 公开号 CN 1797348A

[22] 申请日 2005.12.29
 [21] 申请号 200510135238.6
 [30] 优先权
 [32] 2004.12.29 [33] US [31] 11/027,392
 [71] 申请人 英特尔公司
 地址 美国加利福尼亚州
 [72] 发明人 布拉廷·萨哈 马修·默藤
 塞巴斯蒂恩·希利 戴维·库法迪
 珀尔·哈马伦德

[74] 专利代理机构 北京英特普罗知识产权代理有限公司
 代理人 齐永红

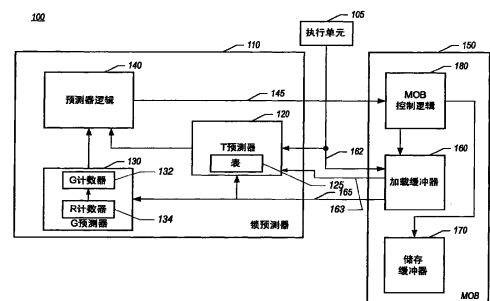
权利要求书 3 页 说明书 9 页 附图 4 页

[54] 发明名称

预测处理器中的竞争

[57] 摘要

在一个实施方案中，本发明包括预测器，用于预测要在程序中执行的操作的竞争。所述操作可以基于所述预测的结果来处理，所述预测可以基于多个独立的预测。在一个实施方案中，如果没有竞争被预测到，则可以优化所述操作。其他的实施方案被描述并被要求保护。



1. 一种装置，包括：
用于预测要被执行的操作的竞争的预测器。
5
2. 如权利要求 1 的装置，其中，所述预测器包括用于储存对应于锁变量的地址信息及所述锁变量的竞争发生的表。
3. 如权利要求 2 的装置，其中，所述预测器还包括用于对线程中的竞争的发生进行计
10 数的计数器。
4. 如权利要求 3 的装置，还包括用于基于来自所述表和所述计数器的信息预测所述锁
操作的竞争的预测器逻辑。
- 15 5. 如权利要求 4 的装置，还包括用于基于所述预测器逻辑的输出优化所述操作的控制器。
6. 如权利要求 1 的装置，其中，所述预测器基于锁操作的锁变量的地址来预测所述竞
20 争。
7. 如权利要求 1 的装置，其中，所述预测器基于第一和第二独立预测来预测所述竞争。
8. 如权利要求 1 的装置，还包括耦合到所述预测器的缓冲器，所述缓冲器将所述操作
25 储存在条目中，所述条目具有与其相关联的竞争指示符。
9. 如权利要求 8 的装置，其中，如果所述条目被探查，则所述竞争指示符指示竞争。
10. 一种系统，包括：
第一预测器，用于预测锁操作是否有竞争；
30 第二预测器，用于指示程序中竞争的水平；以及
耦合到所述第一预测器并耦合到所述第二预测器的动态随机访问存储器（DRAM），
以储存所述程序。
11. 如权利要求 10 的系统，还包括第一控制器，所述第一控制器用于基于所述第一预
35 测器和所述第二预测器的结果产生组合预测。
12. 如权利要求 11 的系统，还包括用于接收所述组合预测的第二控制器。
13. 如权利要求 12 的系统，其中，如果所述组合预测指示没有竞争，则所述第二控制

器将把至少一个优化施加于所述锁操作。

14. 如权利要求 10 的系统, 其中, 所述第一预测器包括储存器, 所述储存器用于至少储存对应于所述程序的锁变量的地址的一部分。

5

15. 如权利要求 10 的系统, 其中, 所述第二预测器包括计数器, 所述计数器用于对所述程序中的竞争发生进行计数。

10 16. 如权利要求 15 的系统, 还包括复位机制, 所述复位机制用于在预定数量的处理器周期之后复位所述计数器。

17. 如权利要求 15 的系统, 其中, 所述计数器包括增/减计数器。

18. 一种方法, 包括:

15 预测锁操作在执行期间是否有竞争; 以及
如果预测所述锁操作没有竞争, 则优化所述锁操作。

19. 如权利要求 18 的方法, 还包括使用第一预测器和第二预测器来独立地预测。

20 20. 如权利要求 19 的方法, 还包括组合所述第一预测器和所述第二预测器的结果。

21. 如权利要求 19 的方法, 还包括将第一有竞争锁变量的地址插入所述第一预测器的第一条目。

25 22. 如权利要求 21 的方法, 还包括: 如果所述第一有竞争锁变量再次遭遇竞争, 则递增所述第一预测器的所述第一条目的计数。

23. 如权利要求 22 的方法, 还包括基于所述计数来在所述第一预测器中预测所述第一有竞争锁变量的竞争。

30

24. 如权利要求 19 的方法, 还包括: 如果程序的任何锁操作有竞争, 则递增所述第二预测器中的计数器。

35 25. 如权利要求 19 的方法, 还包括: 如果所述条目被探查, 则置位对应于所述锁操作的缓冲器条目中的竞争指示符。

26. 如权利要求 25 的方法, 还包括基于所述竞争指示符来更新所述第一预测器和所述第二预测器其中的至少一个。

27. 一种制品，包括包含指令的机器可读储存介质，所述指令如果被机器执行，则使得所述机器能够执行一种方法，所述方法包括：

预测要被执行的操作的竞争；以及
基于所述预测来处理所述操作。

5

28. 如权利要求 27 的制品，其中，所述方法还包括：如果所述条目被探查，则置位对应于所述锁操作的缓冲器条目中的竞争指示符。

29. 如权利要求 28 的制品，其中，所述方法还包括基于所述竞争指示符来更新所述第一预测器和所述第二预测器其中的至少一个。

10

30. 如权利要求 27 的制品，其中，所述方法还包括以两种方式独立地预测所述竞争。

31. 一种装置，包括：

15

用于预测操作是否有竞争的第一预测器，所述第一预测器具有带有多个条目的表；
指示全局竞争水平的第二预测器，所述第二预测器包括全局计数器；以及
耦合到所述第一预测器和所述第二预测器的预测器控制器，所述预测器控制器用于基于来自所述第一预测器和所述第二预测器的输入产生最终预测。

20

32. 如权利要求 31 的装置，还包括耦合来接收所述最终预测的排序控制器。

33. 如权利要求 31 的装置，其中，如果所述最终预测指示没有竞争，则所述排序控制器将把至少一个优化施加于所述操作。

25

34. 如权利要求 31 的装置，其中，所述表至少要储存对应于锁变量的地址的一部分。

35. 如权利要求 31 的装置，还包括用于复位所述全局计数器的复位机制。

预测处理器中的竞争

5 技术领域

本发明涉及计算机系统，并特别地，涉及如执行多个线程的系统。

背景技术

10

包括多处理器（MP）系统和单处理器系统的计算机系统可以包括多个“线程”，每一个线程均独立于其他的线程执行程序指令。使用多个处理器或者线程允许更有效率地并且以更高的速度来处理各种任务或函数，甚至多个应用程序。应用多线程或处理器意味着两个或更多个处理器或者线程可以共享储存在系统内的同一数据并且同时访问它们。但是，

15 在共享数据时必须注意保持存储器排序（memory ordering）。

20

出于数据一致性的目的，如果多个线程或处理器期望读、修改和写单个存储器位置，则不应该允许多个主体（agent）同时对数据进行操作。使多个处理器的使用进一步复杂的是数据经常储存在与处理器相关联、用于加速该处理器对数据的访问的缓存（cache）中。

20 因为这种缓存通常被局限于特定处理器，所以大多数近来的对数据的更新可以被置于系统中缓存的任何一个中。任何访问该数据的主体应该从最近更新过的缓存接收到有效或更新过的数据值，并且，从该缓存写回存储器的数据或转移到其他缓存的数据必须是当前的数据，以便保持缓存一致性（cache coherency）。

25

多线程（MT）软件使用不同的机制来在不同的线程之间交互作用和协调。一个常见的MT同步形式是信号量（semaphore）旋转锁（spin-lock）。信号量旋转锁机制是在访问共享的存储器变量或结构（即共享元素）时用来保证多个线程间的相互排斥（即防止同时访问）的锁操作。为了提供共享元素的唯一且一致的视图，它由锁变量保护。每一个需要访问共享元素的线程必须通过原子化信号量操作获得保护锁。为了获得该锁，线程要读锁

30 变量的值，将该值与预先确定的“自由”值比较，然后写“锁”值。这种读—修改—写操作必须要像是发生在一个步骤中，以使多个线程不同时地读“自由”值且写所述“锁”值，从而允许两个线程都相信它们已经获得了该锁。

35

在给定线程获得了关于锁变量的锁以后，其他期望访问该锁变量的线程通常必须一直等到最初的线程完成其锁操作为止。通常，寻求访问的其他线程将开始对锁变量地址的探查（snoop），以便对状态（即“自由”还是“锁定”）进行检查。发现了“锁定”值的线程将经常等一小段时间并再次探查，从而在小的探查一等待循环中旋转。当一个或更多个线程期望访问已经由另一个线程拥有的锁变量或者期望访问正被另一个线程访问的锁变量时，竞

争出现了。在运行时期间，当一次只有唯一一个主体寻求执行关于该锁变量的锁操作时，锁操作是无竞争的。当线程完成修改共享变量或结构时，它将把“自由”值写回锁变量。

5 因为同步在 MT 应用程序中经常发生，所以处理器应该高效率地实现锁操作，以使 MT 应用程序可以如期望的那样执行。如果锁操作在运行时无竞争，则它能够被大大地优化。这种优化可以包括相关联缓存线（cache line）的推测性预取、锁省略（lock elision）等。但是，如果锁操作有竞争，则这些优化技术导致高昂的代价。

10 因此，存在预测有竞争的锁操作的需求，从而使得处理器能够高效率地实现锁操作。

附图说明

- 图 1 是根据本发明的一个实施方案的处理器的一部分的框图。
图 2 是根据本发明的一个实施方案的方法的流程图。
15 图 3 是根据本发明的一个实施方案的系统的框图。
图 4 是根据本发明另一个实施方案的多处理器系统的框图。

发明内容

20 在一个实施方案中，本发明包括预测器，用于预测要在程序中执行的操作的竞争。所述操作可以基于所述预测的结果来处理，所述预测可以基于多个独立的预测。在一个实施方案中，如果没有竞争被预测到，则可以优化所述操作。其他的实施方案被描述并被要求保护。

25 根据本发明的一个方面，公开了一种装置，包括：用于预测要被执行的操作的竞争的预测器。

30 根据本发明的另一方面，公开了一种系统，包括：第一预测器，用于预测锁操作是否有竞争；第二预测器，用于指示程序中竞争的水平；以及耦合到所述第一预测器并耦合到所述第二预测器的动态随机访问存储器（DRAM），以储存所述程序。

根据本发明的又一方面，公开了一种方法，包括：预测锁操作在执行期间是否有竞争；以及如果预测所述锁操作没有竞争，则优化所述锁操作。

35 根据本发明的再一方面，公开了一种制品，包括包含指令的机器可读储存介质，所述指令如果被机器执行，则使得所述机器能够执行一种方法，所述方法包括：预测要被执行的操作的竞争；以及基于所述预测来处理所述操作。

根据本发明的又再一方面，公开了一种装置，包括：用于预测操作是否有竞争的第一

预测器，所述第一预测器具有带有多个条目的表；指示全局竞争水平的第二预测器，所述第二预测器包括全局计数器；以及耦合到所述第一预测器和所述第二预测器的预测器控制器，所述预测器控制器用于基于来自所述第一预测器和所述第二预测器的输入产生最终预测。

5

具体实施方案

参考图 1，示出了根据本发明的一个实施方案的处理器 100 的一部分的框图。如图 1 中所示，处理器 100 包括锁预测器 110 和存储器排序缓冲器 (MOB) 150。虽然在图 1 中只示出了这些元件，但是要理解，处理器 100 可以包括额外的部件，例如数据缓存、分支预测单元、流水线级、各种寄存器，等等。

锁预测器 110 可以用来预测程序内的锁操作在执行期间是否可能是有竞争的。在图 1 的实施方案中，锁预测器 110 包括第一预测器 (即表或 T 预测器) 120 和第二预测器 (即全局或 G 预测器) 130。T 预测器 120 和 G 预测器 130 可以独立地预测锁操作是否可能是有竞争的。作为结果的来自两个预测器的预测都被提供给预测器逻辑 140，预测器逻辑 140 可以是控制器等，用于以期望的方式对结果进行组合。虽然示出了两个单独的预测器，但是本发明的范围不限于此，并且，在一些实施方案中可以存在额外的预测器 (或者甚至是单独一个)。

20

由预测器逻辑 140 产生的最终预测通过线 145 提供给 MOB 150。更具体地说，最终预测被提供给 MOB 控制逻辑 180，MOB 控制逻辑 180 可以是控制器等，它用来控制 MOB 150 内的缓冲器，所述缓冲器包括加载缓冲器 160 和储存缓冲器 170。加载缓冲器 160 可以用来在加载微操作 (mop) 执行之前储存它们，并跟踪以前的加载或者其他的存储器操作，所述其他的存储器操作必须在给定的存储器操作自身可以被完成之前完成。类似地，储存缓冲器 170 可以用来储存例如储存操作的存储器操作，并用来跟踪以前的存储器操作 (一般是加载)，在可以提交 (commit) 给定的存储器操作自身之前，必须完成所述的以前的存储器操作。

当计算加载锁操作的地址时，对应于锁变量的地址可以通过线 162 从执行单元 105 提供给 T 预测器 120，以便开始预测。此外，操作可以被输入加载缓冲器 160 的条目。提供给 T 预测器 120 的这个地址可以用来访问 T 预测器 120 内的查找表 125，以便输出与锁变量的地址相关联的计数。该计数可以指示程序执行期间在那个锁变量上已经发生过的竞争次数。当然，有关加载锁操作的其他信息 (例如其程序计数器) 也可以被提供给 T 预测器以帮助表查找。

35

如图 1 中所示，在一些实施方案中，额外的线 163 可以耦合在加载缓冲器 160 和 T 预测器 120 之间，以便将对应于表变量的地址提供给 T 预测器 120。该线可以用于从加载缓冲器 160 获得关于重分派的预测。

如图 1 中进一步所示,来自 T 预测器 120 的计数可以从 T 预测器 120 提供给预测器逻辑 140。同时,当前存在的对线程内全局竞争的预测从 G 预测器输出,并提供给预测器逻辑 140。预测器逻辑 140 可以组合这些输入,以获得最终预测,然后,所述最终预测被提供
5 供给 MOB 控制逻辑 180。

依次,MOB 控制逻辑 180 可以基于预测来处理锁操作。例如,MOB 控制逻辑 180 可以基于最终预测来确定是否优化锁操作。虽然 MOB 控制逻辑 180 可以以各种方式来处理锁操作的执行,但是,如果预测指示不可能有竞争,则 MOB 控制逻辑 180 可以实现对
10 锁操作的优化。相反,如果预测指示有可能有竞争,则 MOB 控制逻辑 180 可以使得锁操作被保守地执行。

在一个实施方案中,T 预测器 120 可以包括在程序执行期间曾遭遇竞争的锁变量的地址表 125。虽然在图 1 的实施方案中示出了单个表,但是可以存在多个表,每一个均对应
15 于不同的线程。或者,单个表可以存在,并且在每一个条目中可以包括线程标识符,以便将条目与对应的线程相关联。

表 125 的每一个条目均可以具有饱和计数器(saturating counter),例如两位的饱和计数器。当表变量的地址在程序中第一次有竞争时,它被添加到表 125,并且该条目的计数器被初始化为零。如果对应于表变量的地址再次有竞争,则该条目的计数可以被递增。如果在程序执行期间计数饱和,则 T 预测器 120 可以预测:每一个后续
20 的访问该表变量的锁操作将是有竞争的。相反,如果地址未储存在该表中,或者与该表中的条目相关联的计数还没有饱和,则 T 预测器 120 可以预测:锁操作将是无竞争的。

在一些实施方案中,由于表条目竞争所致,表 125 内的条目可能被覆写。此外,在一些实施方案中,整个表可以被周期性地清除。在一个实施方案中,可以将 128 个条目的表用于该表。在这样的实施方案中,表 125 可以直接被映射,具有被用于每一个条目的 4 位的
25 标签。在一个实现中,表变量地址的低六位可以被屏蔽掉,接下来较高的七位地址可以用作进入表的索引,并且接下来的 4 个最高位可以被储存为标签。地址的剩余位可以被丢
30 弃。

在程序的执行期间,一些锁变量很少有竞争,而其他的则经常有竞争。在各种实施方案中,T 预测器 120 中的计数器的饱和本质可以捕获这种行为。即,经常竞争的锁变量可以很快地使 T 预测器 120 内的计数器饱和,而很少竞争的锁变量将持续地被 T 预测器 120
35 预测为无竞争。在一个实施方案中,可以使用饱和上/下计数器。利用这种计数器,当锁变量有竞争时,可以递增计数器条目,而当其不竞争时,计数器条目可以被递减。然后,T 预测器 120 内的逻辑可以将计数值与阈值进行比较,以确定给定的锁操作是否有可能有竞争。

在一个实施方案中，G 预测器 130 可以包括指示程序内的竞争量的全局（G）计数器 132。虽然在图 1 中被示出为单个计数器，但是可以存在多个计数器，每一个均与不同的线程相关联。

5 G 预测器 130 可以用来指示程序内的竞争的全局视图。虽然在这里针对程序描述，但是在一些实施方案中，G 预测器 130 可以以线程为基础对竞争进行分析。无论何时锁操作有竞争，G 计数器 132 就可以被递增。G 预测器 130 还可以包括指示程序内的竞争水平的控制逻辑。在一个实施方案中，可以将计数器值与第一阈值和第二阈值进行比较。如果当前计数低于第一阈值，则 G 预测器 130 预测程序内的竞争为低。相反，如果计数器值超过了第一阈值但是低于第二阈值，则 G 预测器 130 可以预测程序内的竞争处于中等水平。如果当前计数在第二阈值之上，则 G 预测器 130 可以预测程序内的竞争为高。在一个实施方案中，第一阈值可以被设置为 5 并且第二阈值可以被设置为 25，尽管本发明的范围不限于此。而且，不是向预测器逻辑 140 发送预测，G 预测器 130 可以输出 G 计数器 132 的当前计数值，并且预测器逻辑 140 可以基于该值来预测竞争。

15 在一些实施方案中，计数器 132 可以被间隔性地复位。此外，计数器 132 可以是饱和计数器（即，被防止溢出）。以这种方式，G 预测器 130 可以捕获程序内的竞争突发。即，在程序的某个部分期间（例如给定数量的指令），竞争可能为高，而在程序的不同部分，竞争可以为低（或中等）。以这种方式，G 预测器 130 可以捕获竞争的突发。此外，G 预测器 130 可以起到对抗缘于不良书写的代码等的病态竞争情形的看门狗（watchdog）机制的作用。例如，当 T 预测器 120 中的表 125 被太多的锁和跟踪湮没时，G 预测器 130 可以基于其全局视图来预测竞争。

25 如上所述，计数器 132 可以以各种间隔复位。例如，在一个实施方案中，它可以大约每隔 30,000 个处理器周期被复位。在这样的实施方案中，可以每个处理器周期递增第二个计数器（即复位（R）计数器）134，并且，当第二计数器 134 溢出时，计数器 132 可以被复位。在一个实施方案中，R 计数器 134 可以是 15 位的计数器，且也包括在 G 预测器 130 内。在其他的实施方案中，R 计数器 134 可以被置于 G 预测器 130 之外。此外，在其他实施方案中，G 计数器 132 可以是饱和增/减计数器，所述计数器在发生竞争时被递增，并且在 30 在不发生竞争时被递减。

因此，独立的预测可以被从 T 预测器 120 和 G 预测器 130 提供给预测器逻辑 140。现在参考表 1，所示的是可以在预测器逻辑 140 中实施的状态图。如表 1 中所示，预测器逻辑 140 可以基于独立预测发展出最终预测。

35

表 1

G 元件	T 元件	预测
高	有竞争	有竞争
中等	有竞争	有竞争

低	有竞争	无竞争
高	无竞争	有竞争
中等	无竞争	无竞争
低	无竞争	无竞争

如表 1 中所示，G 预测器 130 可以起到看门狗的作用。当竞争的全局预测是高时，无论单个锁变量的竞争历史如何，总可以预测竞争。类似地，如果 G 预测器 130 指示全局竞争为低，则有可能单个锁变量将继续是无竞争的，而不管其历史。

5

相反，如果 G 预测器 130 预测竞争是中等的，则最终预测可以基于特定锁变量的竞争历史（即 T 预测器 120 的输出）。虽然在表 1 中以特定实现示出，但是要理解，在其他的实施方案中，来自两个独立的预测器的预测可以被以其他方式组合。

10 现在参考图 2，所示是根据本发明的一个实施方案的方法的流程图。如图 2 中所示，方法 200 是竞争预测器的一个实现，并且可以用来确定对于给定的锁操作，竞争是否是可能的。如果预测没有竞争，则锁操作可以被优化。方法 200 可以通过接收从保留站
15 （reservation station）分派的加载操作开始（框 210）。这个操作可以被写入 MOB 150（框 215）。同时，可以确定该加载是否是加载锁操作（菱形 220）。如果它不是，则可以正常执行该操作（框 230）。

相反，如果确定该加载是加载锁操作，则控制从菱形 220 直接传递到菱形 250，在菱形 250 中，地址可以被分派到基于地址来预测竞争的预测器（菱形 250）。例如，回头参考图 1 的预测器，锁变量的地址可以从线 162 提供给 T 预测器 120。依次，T 预测器 120
20 可以使用该地址（如上所述）来访问表 125，以便确定那里是否包含锁变量。如果是，则与锁变量相关联的计数被施加到 T 预测器 120 中的逻辑，T 预测器 120 中的逻辑将预测输出到预测器逻辑 140，预测器逻辑 140 也从 G 预测器 130 接收当前的预测。依次，预测器逻辑 140 可以基于这两个输入来产生最终的竞争预测。该预测被提供给 MOB 控制逻辑 180。

25

基于该预测，可以开始如下面进一步描述的协议。基于菱形 250 处的竞争预测，如果不满足某些条件，则协议可以阻塞（block）实际操作，而同时，如果预测是没有竞争，则仍然允许被优化的部分（例如预取或省略）继续下去。

30 如果在菱形 250 处预测了竞争，则可以执行保守的锁定协议（框 270）。例如，MOB 150 可以等待加载缓冲器 160 中是最老的微操作（uop）的加载锁操作，或者可以等待储存缓冲器 170 排出（drain），或者是类似的过程。

反之，如果预测没有竞争，则控制从菱形 250 传递到框 260。在那里，锁操作可以被
35 优化（框 260）。例如，加载锁操作可以被推测性地分派，以便预取对应于加载变量的缓

存线。也可以实现其他类似的这种优化。例如，在加载锁微操作和依赖性（dependent）微操作变成机器中最老的之前，或者储存缓冲器被排出之前，加载锁微操作和依赖性微操作可以被推测性地执行。

5 此外，在一些实施方案中，在从 MOB 150 重分派之后，预测器可以被访问，控制从菱形 220 传递倒菱形 250，以便为了预测重新检查。或者，可以储存和重新使用先前的预测。

10 在操作期间，MOB 150 监视进入的探查，以便确定任何其他的线程或处理器是否想要访问同一锁变量。MOB 150 继续这种监视，直到加载锁微操作已经从机器中退出为止。在 MOB 150 正监视锁变量时发生的探查将指示不当推测，并且将要求重新执行加载锁及其依赖性（dependent）操作。

15 在一个实施方案中，通过使额外的指示符与加载缓冲器 160 内的条目相关联来指示竞争，可以检测到加载操作的实际竞争。但是要理解，在其他实施例中可以用其他的方式检测竞争。

20 当为了加载操作分配了加载缓冲器 160 中的条目时，竞争指示符（它可以是单个的竞争位）可以被复位。如果探查操作在条目已经被分配之后并且在指令已经退出之前进入对应于加载操作的加载缓冲器，则竞争位可被置位。当对应于加载操作的加载缓冲器 160 的条目以后被解除分配时（在锁指令已退出之后），可以查看竞争位以检测竞争。更具体地说，条目及其相关联的竞争位可以被传递倒 T 预测器 120 和 G 预测器 130，以便对它们进行更新。即，如果竞争位被置位，则 T 预测器 120 和 G 预测器 130 内的计数器可以被递增。在图 1 的实施方案中，对于这种更新，可以存在单独的更新通道 165。

25 现在参考图 3，所示是根据本发明的一个实施方案的代表性计算机系统 300 的框图。如图 3 中所示，计算机系统 300 包括第一处理器 310a 和第二处理器 310b。在一个实施方案中，处理器 310a 可以经第二层（L2）缓存 305，通过存储器系统互连 320 耦合到缓存一致（coherent）共享存储器子系统（“一致性存储器（coherent memory）”）330。在一个
30 实施方案中，一致性存储器 330 可以包括动态随机访问存储器（DRAM），并且还可以包括用于在处理器 310a 和处理器 310b 之间共享一致性存储器 330 的一致性存储器控制器逻辑。

35 要理解，在其他的实施方案中，额外的处理器可以耦合到一致性存储器 330。此外，在某些实施方案中，一致性存储器 330 可以分部分实现，并分散开，以使系统 300 内处理器的子集与一致性存储器 330 的某些部分通信，而其他的处理器与一致性存储器 330 的其他部分通信。

如图 3 中所示，处理器 310a 可以包括锁预测器 312a、存储器排序缓冲器（MOB）314a，

以及执行单元 316a。如上所述，锁预测器 312a 可以耦合到 MOB 314a，以提供对锁竞争的预测。基于这些预测，MOB 314a 可以优化锁操作。然后，被优化的操作可以被传递到执行单元 316a，执行单元 316a 可以具有执行所述操作的流水线级。然后，有关操作的信息可以被传递回 MOB 314a，以指示操作完成且准备好了解除分配。此时，MOB 314a 可以用操作的竞争状态（即有竞争或无竞争）来对锁预测器 312a 进行更新。当然，在处理器 310a 中可以存在额外的元件，例如第一层（L1）缓存。

如图 3 中进一步所示，处理器 310b 中可以存在类似的处理器元件，处理器 310b 可以是多处理器系统的第二个核心处理器。虽然在图 3 的实施方案中示出为每一个均包括预测器，但是在一些实施方案中，可能只有单个核心包括这样的预测器，所述预测器可以由多个核心共享。更进一步，在一些实施方案中，多线程应用程序可针对每一个线程使用单独的预测器。在其他的实施方案中，锁预测器可以存在于单处理器系统中。

一致性存储器 330 也可以（经过中心链路（hub link））耦合到输入/输出（I/O）中心（hub）335，输入/输出中心 335 耦合到 I/O 扩展总线 355 和外部总线 350。在不同的实施方案中，I/O 扩展总线 355 可以耦合到各种 I/O 设备，例如键盘或鼠标，以及其他设备。外部总线 350 可以耦合到各种部件，例如外部设备 370，外部设备 370 可以是存储器设备，例如快闪存储器、插卡（add-in card），等等。尽管描述参考了系统 300 的具体部件，但是所示实施方案的很多修改是可能的。

在一些实施方案中，多处理器系统可以是点到点总线系统，例如在公共系统接口（CSI）系统中。现在参考图 4，所示是根据本发明的另一个实施方案的多处理器系统的框图。如图 4 中所示，多处理器系统是点到点总线系统，并且包括经过点到点互连 450 耦合的第一处理器 470 和第二处理器 480。第一处理器 470 包括处理器核心 474、存储器控制器中心（MCH）472，以及点到点（P-P）接口 476 和 478。类似地，第二处理器 480 包括相同的部件，即处理器核心 484、MCH 483，以及 P-P 接口 486 和 488。根据本发明的实施方案，每一个处理器 470 和 480 均可以包括预测电路。

如图 4 中所示，MCH 472 和 482 将处理器耦合到相应的存储器，即存储器 432 和存储器 434，所述存储器可以是在局部上连接到相应处理器的主存储器的一部分。

第一处理器 470 和第二处理器 480 可以通过 P-P 接口（interface）452 和 454 分别耦合到芯片组 490。如图 4 中所示，芯片组 900 包括 P-P 接口 494 和 498。此外，芯片组 490 包括将芯片组 490 耦合到高性能图形引擎 438 的接口 492。在一个实施方案中，先进图形端口（AGP）总线 439 可以用来将图形引擎 438 耦合到芯片组 490。AGP 总线 439 可以符合加州圣克拉拉的英特尔公司 1998 年 4 月公布的图形加速端口接口规范 2.0 修订本。或者，点到点互连 439 可以耦合这些部件。

依次，芯片组 490 可以通过接口 496 耦合到第一总线 416。在一个实施方案中，第一

总线 416 可以是如 1995 年 6 月的 PCI 局部总线规范，生产版，修订本 2.1 定义的外设部件互连 (PCI) 总线，或者是例如 PCI Express 总线或其他第三代 I/O 互连总线的总线，尽管本发明的范围不限于此。

5 如图 4 中所示，各种输入/输出 (I/O) 设备 414 以及总线桥 418 可以耦合到第一总线 416，总线桥 418 将第一总线 416 耦合到第二总线 420。在一个实施方案中，第二总线 420 可以是低引脚数 (low pin count, LPC) 总线。各种设备可以耦合到第二总线 420，例如包括键盘/鼠标 422、通讯设备 426 和数据储存单元 428，在一个实施方案中，数据储存单元 428 可以包括代码 430。此外，音频 I/O 424 可以耦合到第二总线 420。

10

可以在计算机程序中实现实施方案，所述计算机程序储存在储存介质上，具有用于对计算机系统编程来执行所述实施方案的指令。储存介质可以包括但不限于：任何类型的盘片、半导体器件，或者任何类型的适于储存电子指令的介质，所述盘片包括软盘、光盘、致密盘只读存储器 (CD-ROM)、致密盘可读写存储器 (CD-RW)，以及磁光盘；所述
15 半导体器件例如只读存储器 (ROM)、诸如静态和动态 RAM 的随机访问存储器 (RAM)，可擦除可编程只读存储器 (EPROM)、电可擦除可编程只读存储器 (EEPROM)、快闪存储器、磁卡或光卡。其他的实施方案可以实现为被可编程控制设备执行的软件模块。

20

虽然这里描述为基于锁变量的地址来预测竞争，但是在其他实施方案中，预测也可以
20 基于锁定指令的线性指令指针 (LIP) (有时候也称为程序计数器)。此外，在其他实施方案中，预测可以基于 LIP 和锁变量地址的某种组合。在另一种实施方案中，预测也可以包含来自过去的分支历史/目标信息的信息。而且，其他操作的竞争预测可能受影响。

25

虽然已经参考有限数量的实施方案描述了本发明，但是熟练技术人员将理解由此产生的各种修改和变化。所附权利要求书旨在覆盖所有这种落入本发明的真正精神和范围的修改和变化。

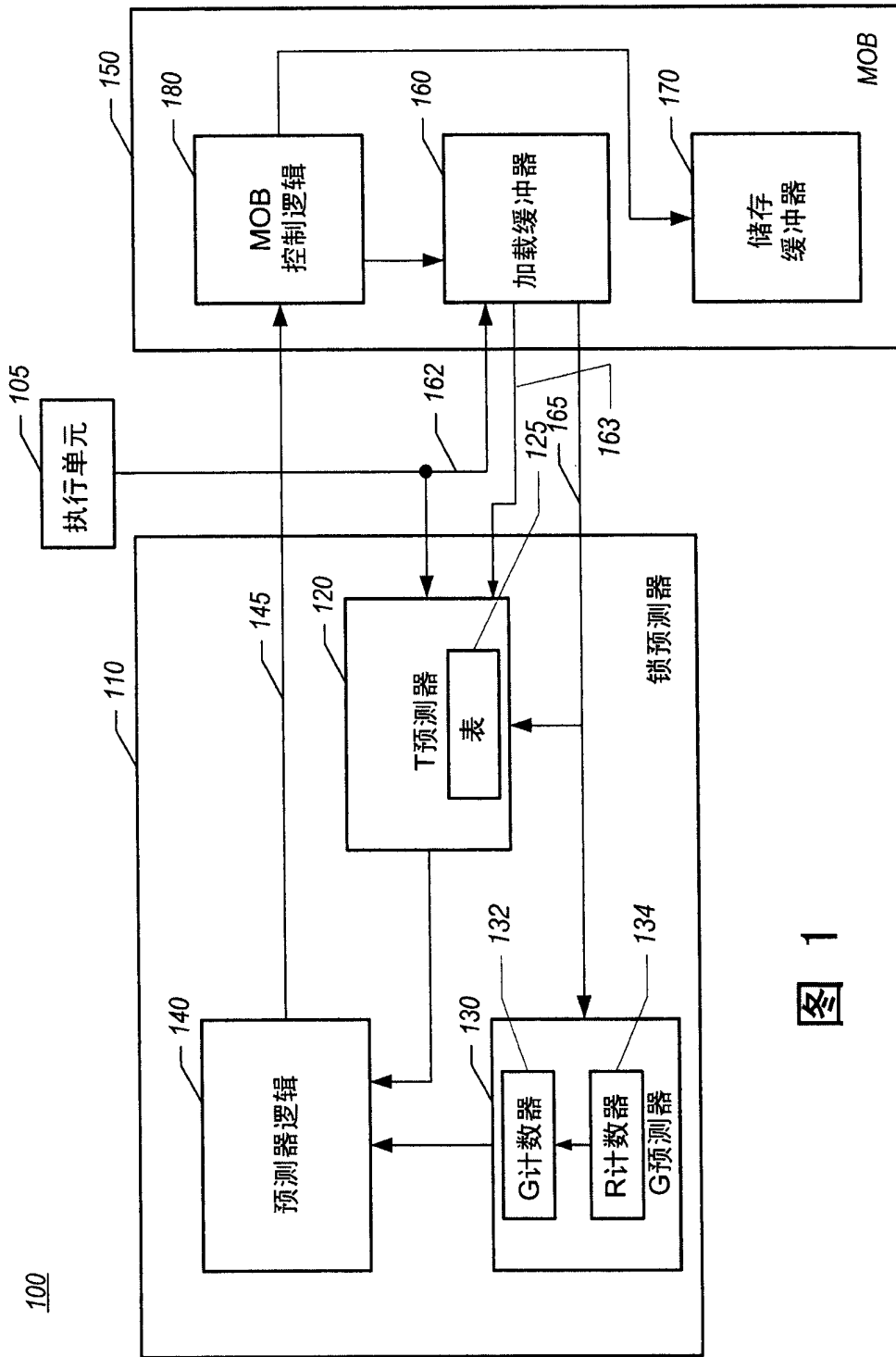


图 1

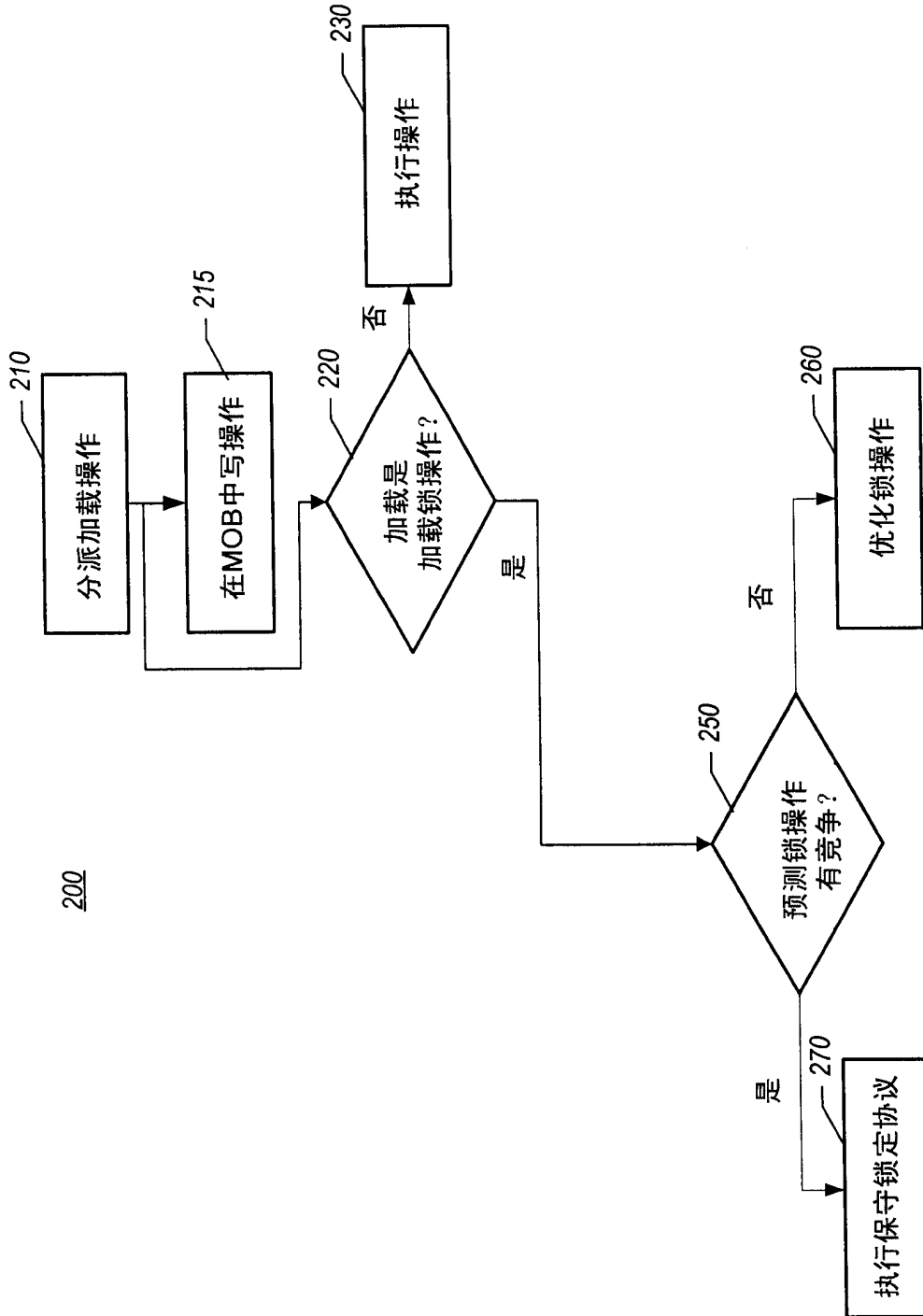


图 2

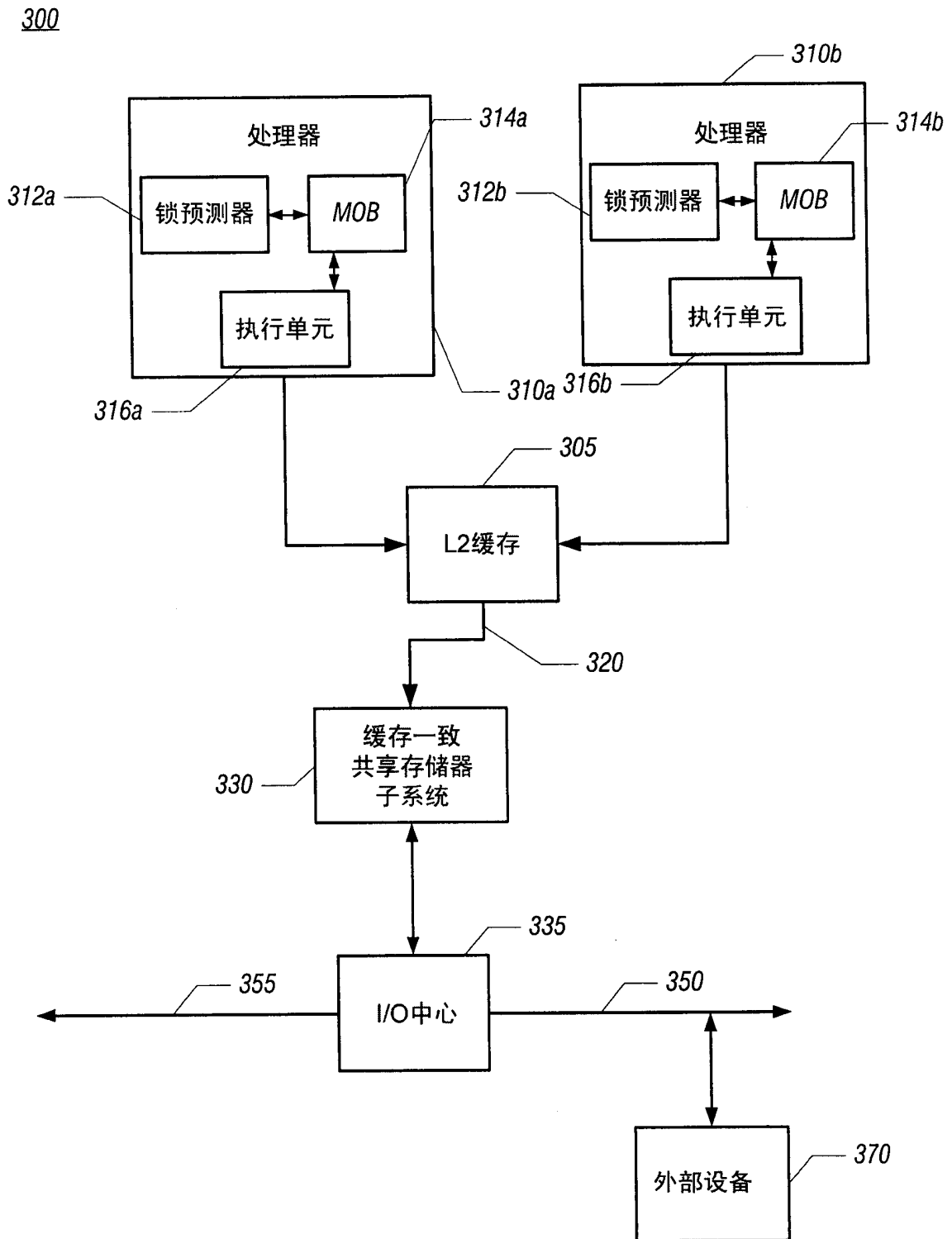


图 3

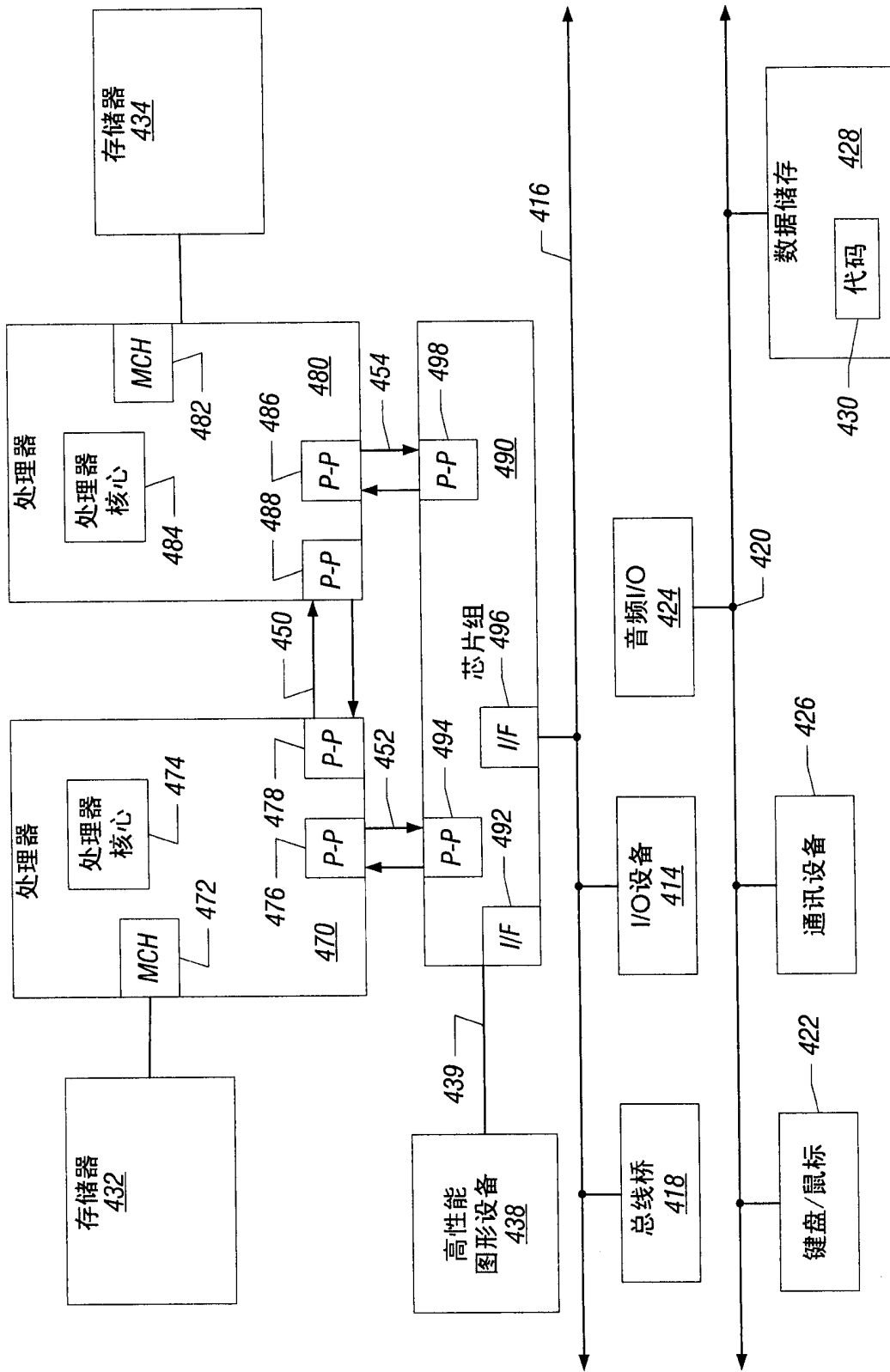


图 4