



(21)申请号 201310416585.0

(22)申请日 2013.09.13

(65)同一申请的已公布的文献号

申请公布号 CN 103677756 A

(43)申请公布日 2014.03.26

(30)优先权数据

13/620047 2012.09.14 US

(73)专利权人 通用电气公司

地址 美国纽约州

(72)发明人 W.D.史密斯二世 S.N.U.艾哈迈德

J.M.迪克马

(74)专利代理机构 中国专利代理(香港)有限公

司 72001

代理人 易皎鹤 汤春龙

(51)Int.Cl.

G06F 9/38(2006.01)

(56)对比文件

US 2008/0196037 A1,2008.08.14,说明书
第15,33,44-45,52,61-69段和附图1,3.

CN 1441348 A,2003.09.10,全文.

US 6633969 B1,2003.10.14,全文.

审查员 陈敏

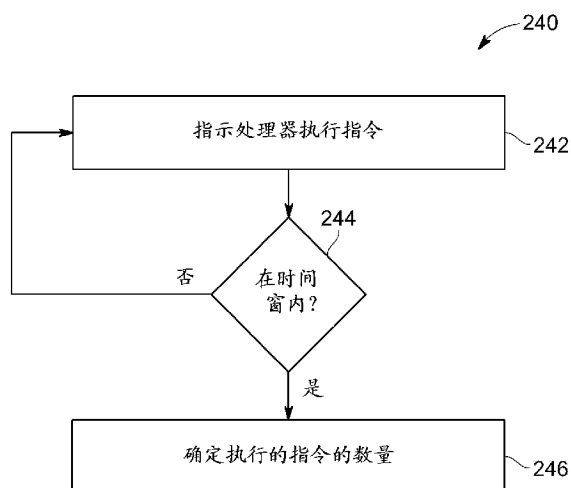
权利要求书3页 说明书13页 附图6页

(54)发明名称

用于使处理器指令执行同步的系统和方法

(57)摘要

用于控制处理器指令执行的系统和方法。在一个示例中,用于使由处理器执行的指令的数量同步的方法包括指示第一处理器经由第一组迭代来迭代地执行指令直到经过预定时段。在该第一组迭代的每个迭代中执行的指令的数量小于在该第一组迭代中的之前迭代中执行的指令的数量。方法还包括指示第二处理器经由第二组迭代来迭代地执行指令直到经过预定时段。在该第二组迭代的每个迭代中执行的指令的数量小于在该第二组迭代中的之前迭代中执行的指令的数量。该方法包括确定是否要执行额外的指令。



1. 一种用于使由多个处理器执行的指令的数量同步的方法,包括:

指示第一处理器经由第一多个迭代来迭代地执行指令直到经过预定时段,其中在所述第一多个迭代的每个迭代中执行的指令的数量小于在所述第一多个迭代中的之前迭代中执行的指令的数量;

指示第二处理器经由第二多个迭代来迭代地执行指令直到经过预定时段,其中在所述第二多个迭代的每个迭代中执行的指令的数量小于在所述第二多个迭代中的之前迭代中执行的指令的数量;

确定在所述预定时段期间由所述第一处理器执行的指令的第一总数量;

确定在所述预定时段期间由所述第二处理器执行的指令的第二总数量;并且

如果指令的所述第二总数量小于指令的所述第一总数量则指示所述第二处理器执行第一计算数量的指令。

2. 如权利要求1所述的方法,包括如果指令的所述第一总数量小于指令的所述第二总数量则指示所述第一处理器执行第二计算数量的指令。

3. 如权利要求1所述的方法,包括如果指令的所述第一总数量大于或等于指令的所述第二总数量则指示所述第一处理器延迟处理。

4. 如权利要求1所述的方法,包括如果指令的所述第二总数量大于或等于指令的所述第一总数量则指示所述第二处理器延迟处理。

5. 如权利要求1所述的方法,包括指示第三处理器经由第三多个迭代来迭代地执行指令直到经过所述预定时段,其中在所述第三多个迭代的每个迭代中执行的指令的数量小于在所述第三多个迭代中的之前迭代中执行的指令的数量。

6. 如权利要求5所述的方法,包括确定在所述预定时段期间由所述第三处理器执行的指令的第三总数量。

7. 如权利要求6所述的方法,包括如果指令的所述第三总数量小于指令的所述第一总数量或指令的所述第二总数量则指示所述第三处理器执行第三计算数量的指令。

8. 如权利要求6所述的方法,包括如果指令的所述第三总数量大于或等于指令的所述第一总数量并且如果指令的所述第三总数量大于或等于指令的所述第二总数量则指示所述第三处理器延迟处理。

9. 如权利要求6所述的方法,包括如果指令的所述第一总数量小于指令的所述第二总数量或指令的所述第三总数量则指示所述第一处理器执行第二计算数量的指令。

10. 如权利要求6所述的方法,包括如果指令的所述第一总数量大于或等于指令的所述第二总数量并且如果指令的所述第一总数量大于或等于指令的所述第三总数量则指示所述第一处理器延迟处理。

11. 如权利要求6所述的方法,包括如果指令的所述第二总数量小于指令的所述第一总数量或指令的所述第三总数量则指示所述第二处理器执行第一计算数量的指令。

12. 如权利要求6所述的方法,包括如果指令的所述第二总数量大于或等于指令的所述第一总数量并且如果指令的所述第二总数量大于或等于指令的所述第三总数量则指示所述第二处理器延迟处理。

13. 一种用于使由多个处理器执行的指令的数量同步的方法,其包括:

接收到系统内的输入数据;

将所述输入数据提供给多个处理器；

使用所述输入数据执行应用码以从所述多个处理器产生相应的多个输出；以及

对所述相应的多个输出应用分布式一致性算法来产生一致性输出；

其中所述多个处理器包括：

第一处理器，其配置成经由第一多个迭代来迭代地执行指令直到经过预定时段，其中在所述第一多个迭代的每个迭代中执行的指令的数量小于在所述第一多个迭代中的之前迭代中执行的指令的数量；以及

第二处理器，其配置成经由第二多个迭代来迭代地执行指令直到经过预定时段，其中在所述第二多个迭代的每个迭代中执行的指令的数量小于在所述第二多个迭代中的之前迭代中执行的指令的数量；

其中，确定在所述预定时段期间由所述第一处理器执行的指令的第一总数量，确定在所述预定时段期间由所述第二处理器执行的指令的第二总数量，如果指令的所述第二总数量小于指令的所述第一总数量则指示所述第二处理器执行计算数量的指令。

14. 如权利要求13所述的方法，其中，执行所述输入数据包括如果指令的所述第二总数量大于或等于指令的所述第一总数量则指示所述第二处理器延迟处理。

15. 一种用于使处理器同步的系统，其包括：

第一处理器，其配置成经由第一多个迭代来迭代地执行指令直到经过预定时段，其中在所述第一多个迭代的每个迭代中执行的指令的数量小于在所述第一多个迭代中的之前迭代中执行的指令的数量；

第二处理器，其配置成经由第二多个迭代来迭代地执行指令直到经过预定时段，其中在所述第二多个迭代的每个迭代中执行的指令的数量小于在所述第二多个迭代中的之前迭代中执行的指令的数量；以及

第三处理器，其配置成经由第三多个迭代来迭代地执行指令直到经过预定时段，其中在所述第三多个迭代的每个迭代中执行的指令的数量小于在所述第三多个迭代中的之前迭代中执行的指令的数量；

其中，确定在所述预定时段期间由所述第一处理器执行的指令的第一总数量，确定在所述预定时段期间由所述第二处理器执行的指令的第二总数量，并且确定在所述预定时段期间由所述第三处理器执行的指令的第三总数量；

其中，指示执行指令的总数量最小的处理器执行计算数量的指令。

16. 如权利要求15所述的系统，所述系统包括第四处理器，所述第四处理器配置成如果指令的所述第一总数量小于指令的所述第二或第三总数量则指示所述第一处理器执行计算数量的指令，并且如果指令的所述第一总数量大于或等于指令的所述第二和第三总数量则指示所述第一处理器延迟处理。

17. 如权利要求15所述的系统，所述系统包括第四处理器，所述第四处理器配置成如果指令的所述第二总数量小于指令的所述第一或第三总数量则指示所述第二处理器执行计算数量的指令，并且如果指令的所述第二总数量大于或等于指令的所述第一和第三总数量则指示所述第二处理器延迟处理。

18. 如权利要求15所述的系统，所述系统包括第四处理器，所述第四处理器配置成如果指令的所述第三总数量小于指令的所述第一或第二总数量则指示所述第三处理器执行计

算数量的指令,并且如果指令的所述第三总数量大于或等于指令的所述第一和第二总数量则指示所述第三处理器延迟处理。

用于使处理器指令执行同步的系统和方法

技术领域

[0001] 本文公开的主旨涉及处理器,并且更特别地,涉及用于使处理器指令执行同步的系统和方法。

背景技术

[0002] 处理器(例如,中央处理单元(CPU)、微处理器,等)用于在多种不同的应用中执行指令。例如,处理器可用于在具有高完整性、可用性和保证性要求的安全关键和/或任务关键的应用中执行指令。这样的应用可包括航空、火箭、飞船、军事用途、核电厂等等。在这样的应用中,两个或以上的处理器可配置成在规定的时间量内执行特定数量的指令。例如,两个或以上的处理器可配置成采用同步方式操作使得两个或以上的处理器大致上同时开始和/或结束操作。遗憾地,如果两个或以上的处理器未大致上同时开始和/或结束操作,来自这两个或以上的处理器的结果可能不可靠和/或不可用。

发明内容

[0003] 根据实施例,用于使由处理器执行的指令的数量同步的方法包括指示第一处理器经由第一组迭代来迭代地执行指令直到经过预定时段。在该第一组迭代的每个迭代中执行的指令的数量小于在该第一组迭代中的之前迭代中执行的指令的数量。方法还包括指示第二处理器经由第二组迭代来迭代地执行指令直到经过预定时段。在该第二组迭代的每个迭代中执行的指令的数量小于在该第二组迭代中的之前迭代中执行的指令的数量。该方法包括确定在预定时段期间由第一处理器执行的指令的第一总数量并且确定在预定时段期间由第二处理器执行的指令的第二总数量。该方法还包括如果指令的第二总数量小于指令的第一总数量则指示第二处理器执行第一计算数量的指令。

[0004] 根据另一个实施例,方法包括接收到系统内的输入数据、将该输入数据提供给多个处理器、使用该输入数据执行应用码以从处理器产生相应数量的输出以及对该相应数量的输出应用分布式一致性算法来产生一致性输出。处理器包括第一处理器,其配置成经由第一组迭代来迭代地执行指令直到经过预定时间段。在该第一组迭代的每个迭代中执行的指令的数量小于在该第一组迭代中的之前迭代中执行的指令的数量。处理器还包括第二处理器,其配置成经由第二组迭代来迭代地执行指令直到经过预定时段。在该第二组迭代的每个迭代中执行的指令的数量小于在该第二组迭代中的之前迭代中执行的指令的数量。

[0005] 根据另外的实施例,用于使处理器同步的系统包括第一处理器,其配置成经由第一组迭代来迭代地执行指令直到经过预定时间段。在该第一组迭代的每个迭代中执行的指令的数量小于在该第一组迭代中的之前迭代中执行的指令的数量。系统还包括第二处理器,其配置成经由第二组迭代来迭代地执行指令直到经过预定时段。在该第二组迭代的每个迭代中执行的指令的数量小于在该第二组迭代中的之前迭代中执行的指令的数量。系统包括第三处理器,其配置成经由第三组迭代来迭代地执行指令直到经过预定时段。在该第三组迭代的每个迭代中执行的指令的数量小于在该第三组迭代中的之前迭代中执行的指

令的数量。

[0006] 提供一种用于使由多个处理器执行的指令的数量同步的方法,其包括:

[0007] 指示第一处理器经由第一多个迭代来迭代地执行指令直到经过预定时段,其中在所述第一多个迭代的每个迭代中执行的指令的数量小于在所述第一多个迭代中的之前迭代中执行的指令的数量;

[0008] 指示第二处理器经由第二多个迭代来迭代地执行指令直到经过预定时段,其中在所述第二多个迭代的每个迭代中执行的指令的数量小于在所述第二多个迭代中的之前迭代中执行的指令的数量;

[0009] 确定在所述预定时段期间由所述第一处理器执行的指令的第一总数量;

[0010] 确定在所述预定时段期间由所述第二处理器执行的指令的第二总数量;并且

[0011] 如果指令的所述第二总数量小于指令的所述第一总数量则指示所述第二处理器执行第一计算数量的指令。

[0012] 优选的,所述方法包括如果指令的所述第一总数量小于指令的所述第二总数量则指示所述第一处理器执行第二计算数量的指令。

[0013] 优选的,所述方法包括如果指令的所述第一总数量大于或等于指令的所述第二总数量则指示所述第一处理器延迟处理。

[0014] 优选的,所述方法包括如果指令的所述第二总数量大于或等于指令的所述第一总数量则指示所述第二处理器延迟处理。

[0015] 优选的,所述方法包括指示第三处理器经由第三多个迭代来迭代地执行指令直到经过所述预定时段,其中在所述第三多个迭代的每个迭代中执行的指令的数量小于在所述第三多个迭代中的之前迭代中执行的指令的数量。

[0016] 优选的,所述方法包括确定在所述预定时段期间由所述第三处理器执行的指令的第三总数量。

[0017] 优选的,所述方法包括如果指令的所述第三总数量小于指令的所述第一总数量或指令的所述第二总数量则指示所述第三处理器执行第三计算数量的指令。

[0018] 优选的,所述方法包括如果指令的所述第三总数量大于或等于指令的所述第一总数量并且如果指令的所述第三总数量大于或等于指令的所述第二总数量则指示所述第三处理器延迟处理。

[0019] 优选的,所述方法包括如果指令的所述第一总数量小于指令的所述第二总数量或指令的所述第三总数量则指示所述第一处理器执行第二计算数量的指令。

[0020] 优选的,所述方法包括如果指令的所述第一总数量大于或等于指令的所述第二总数量并且如果指令的所述第一总数量大于或等于指令的所述第三总数量则指示所述第一处理器延迟处理。

[0021] 优选的,所述方法包括如果指令的所述第二总数量小于指令的所述第一总数量或指令的所述第三总数量则指示所述第二处理器执行第一计算数量的指令。

[0022] 优选的,所述方法包括如果指令的所述第二总数量大于或等于指令的所述第一总数量并且如果指令的所述第二总数量大于或等于指令的所述第三总数量则指示所述第二处理器延迟处理。

[0023] 提供一种方法,其包括:

- [0024] 接收到系统内的输入数据；
- [0025] 将所述输入数据提供给多个处理器；
- [0026] 使用所述输入数据执行应用码以从所述多个处理器产生相应的多个输出；以及
- [0027] 对所述相应的多个输出应用分布式一致性算法来产生一致性输出；
- [0028] 其中所述多个处理器包括：
- [0029] 第一处理器，其配置成经由第一多个迭代来迭代地执行指令直到经过预定时段，其中在所述第一多个迭代的每个迭代中执行的指令的数量小于在所述第一多个迭代中的之前迭代中执行的指令的数量；以及
- [0030] 第二处理器，其配置成经由第二多个迭代来迭代地执行指令直到经过预定时段，其中在所述第二多个迭代的每个迭代中执行的指令的数量小于在所述第二多个迭代中的之前迭代中执行的指令的数量。
- [0031] 优选的，执行所述输入数据包括确定在所述预定时段期间由所述第一处理器执行的指令的第一总数量并且确定在所述预定时段期间由所述第二处理器执行的指令的第二总数量。
- [0032] 优选的，执行所述输入数据包括如果指令的所述第二总数量小于指令的所述第一总数量则指示所述第二处理器执行计算数量的指令。
- [0033] 优选的，执行所述输入数据包括如果指令的所述第二总数量大于或等于指令的所述第一总数量则指示所述第二处理器延迟处理。
- [0034] 提供一种用于使处理器同步的系统，其包括：
- [0035] 第一处理器，其配置成经由第一多个迭代来迭代地执行指令直到经过预定时段，其中在所述第一多个迭代的每个迭代中执行的指令的数量小于在所述第一多个迭代中的之前迭代中执行的指令的数量；
- [0036] 第二处理器，其配置成经由第二多个迭代来迭代地执行指令直到经过预定时段，其中在所述第二多个迭代的每个迭代中执行的指令的数量小于在所述第二多个迭代中的之前迭代中执行的指令的数量；以及
- [0037] 第三处理器，其配置成经由第三多个迭代来迭代地执行指令直到经过预定时段，其中在所述第三多个迭代的每个迭代中执行的指令的数量小于在所述第三多个迭代中的之前迭代中执行的指令的数量。
- [0038] 优选的，所述系统包括第四处理器，所述第四处理器配置成如果指令的所述第一总数量小于指令的所述第二或第三总数量则指示所述第一处理器执行计算数量的指令，并且如果指令的所述第一总数量大于或等于指令的所述第二和第三总数量则指示所述第一处理器延迟处理。
- [0039] 优选的，所述系统包括第四处理器，所述第四处理器配置成如果指令的所述第二总数量小于指令的所述第一或第三总数量则指示所述第二处理器执行计算数量的指令，并且如果指令的所述第二总数量大于或等于指令的所述第一和第三总数量则指示所述第二处理器延迟处理。
- [0040] 优选的，所述系统包括第四处理器，所述第四处理器配置成如果指令的所述第三总数量小于指令的所述第一或第二总数量则指示所述第三处理器执行计算数量的指令，并且如果指令的所述第三总数量大于或等于指令的所述第一和第二总数量则指示所述第三

处理器延迟处理。

附图说明

[0041] 当下列详细描述参照附图(其中所有图中类似的符号代表类似的部件)阅读时,本发明的这些和其他特征、方面和优势将变得更好理解,其中:

[0042] 图1是可采用两个或以上的同步处理器的高完整性系统的实施例的框图;

[0043] 图2是可采用两个或以上的同步处理器的高完整性系统的另一个实施例的框图;

[0044] 图3是具有多核处理器(其具有管理程序来控制多个处理器的操作)的高完整性系统的实施例的框图;

[0045] 图4是图3的多核处理器的图;

[0046] 图5是用于控制由处理器执行的指令的方法的实施例的定时图;

[0047] 图6是用于使多个处理器同步的方法的实施例的定时图;

[0048] 图7是用于控制由处理器执行的指令的方法的实施例的流程图。

具体实施方式

[0049] 将在下文描述本公开的一个或多个特定实施例。为了提供这些实施例的简洁说明,实际实现的所有特征可不在说明书中描述。应该意识到在任何这样的实际实现的开发中,如在任何工程或设计项目中,必须做出许多实现特定的决定来达到开发者的特定目标,例如遵守系统相关和业务相关的约束,其可在实现之间变化。此外,应该意识到这样的开发努力可能是复杂并且耗时的,但对于具有该公开的利益的那些普通技术人员仍将是设计、制作和制造的例行任务。

[0050] 当介绍本发明的各种实施例的要素时,冠词“一(a/an)”、“该”和“所述”意在表示存在要素中的一个或多个。术语“包括”、“包含”和“具有”意在为包括性的并且表示可存在除列出的要素外的附加要素。

[0051] 图1是可采用两个或以上的同步处理器的高完整性系统10的实施例的框图。如可意识到的,该高完整性系统10可以是高完整性(例如,达到正确答案的高度可能性)、高可用性(例如,能够传递答案的运作系统)和/或高保证性(例如,系统正运行正确软件的置信度)系统。例如,高完整性系统10可设计具有小于近似 1.0×10^{-9} 每小时(例如,每飞行小时)的失效时间(FIT)率。某些实施例可包括三个或以上的同步处理器用于提供高完整性和高可用性系统。因此,高完整性系统10可设计成在实时安全关键航空电子设备应用(例如,飞行器)中使用。

[0052] 高完整性系统10包括存储器12,用于保持应用程序14。此外,该应用程序14配置成向一个或多个处理器16、18和20提供可执行指令。如可意识到的,处理器16、18和20中的每个的应用程序14是相同功能的应用程序并且向一个或多个处理器16、18和20提供可执行指令。因此,设计高完整性系统10使得平台(例如,处理器16、18和20)提供完整性和可用性。在这样的配置中,平台可包括复杂设计,而应用程序14包括简化设计。在某些实施例中,应用程序14可设计成提供完整性。在这样的配置中,平台可包括简单的设计,而应用程序14包括复杂设计来提供高完整性、高可用性和高保证性的逻辑。如在本实施例中图示的,处理器16、18和20向系统总线22提供结果。例如,可从处理器16、18和20向系统总线22提供来自执

行指令的结果。

[0053] 尽管仅图示三个处理器16、18和20,高完整性系统10可包括更少或更多的处理器。例如,在某些实施例中,可在具有小于近似 1.0×10^{-11} 每小时的FIT率的设计中使用五个或六个处理器,其中每个处理器具有近似 1.0×10^{-3} 的FIT率。应该注意许多假设和变量用于计算FIT率。作为另一个示例,可在具有小于近似 1.0×10^{-15} 每小时的FIT率的设计中使用六个、七个或八个处理器。处理器16、18和20可每个接收相同的输入并且可设计成提供相同的输出。因此,来自处理器16、18和20的输出可实时同步来证实来自处理器16、18和20的输出是相同的。利用同步的处理器16、18和20,高完整性系统10可产生可靠的结果。

[0054] 由同步处理器16、18和20产生的可靠结果可满足高完整性、可用性和保证性要求。如此,高完整性系统10可用于安全关键和/或任务关键的应用。因此,高完整性系统10可在不使用专门硬件的情况下以高效且成本有效的方式操作。

[0055] 图2是可采用两个或以上的同步处理器的高完整性系统26的实施例的框图。该高完整性系统26包括第一复制状态机28和第二复制状态机30。该第一复制状态机28具有输入32(例如,输入数据),其等同地提供给每个处理器34、36和38。在执行指令后,处理器34、36和38提供同步输出。之后,第一复制状态机28执行分布式一致性40(例如,算法),其导致来自第一复制状态机28的单个输出。此外,第二复制状态机30具有输入42(例如,输入数据),其等同地提供给每个处理器44、46和48。在某些实施例中,输入32和42是相同的。在执行指令后,处理器44、46和48每个提供同步输出。之后,第二复制状态机30执行分布式一致性50(例如,算法),其导致来自第二复制状态机30的单个输出。

[0056] 来自第一和第二复制状态机28和30的输出提供冗余,其便于使用分布式一致性40或分布式一致性50。例如,如果第一复制状态机28的分布式一致性40确定答案,则使用该答案。然而,如果第一复制状态机28的分布式一致性40未确定答案,使用来自第二复制状态机30的分布式一致性50的答案。从而,第一和第二复制状态机28和30的冗余便于提高可用性。应该注意使用本实施例,可获得小于近似 1.0×10^{-15} 每小时的FIT率。

[0057] 如可意识到的,高完整性系统26可对分布式一致性使用任何的算法。例如,除其他外,高完整性系统26可使用基础paxos、多-paxos、便宜paxos、快速paxos、通用paxos、拜占庭多-paxo、快速拜占庭多-paxo、较少存储paxo、应对paxos、一个正确的paxos、快速paxos B、信息理论拜占庭paxos、Chandra-Toueg (CT)、Mostefaoui-Raynal (MR)、查询更新(Q/U)、混合法定人数集(HQ)、象鼻虫(zyzzyva)、土豚(aardvark)、直立、Brooks-Iyengar、实际拜占庭容差(PBFT)和/或aliph。

[0058] 如上文描述的,处理器34、36和38可每个提供同步输出。此外,处理器44、46和48可每个提供同步输出。如可意识到的,处理器可采用提供实时同步的任何适合的方式来同步。因此,高完整性系统26的分布式一致性40和50可提供可靠和/或可用结果。

[0059] 图3是具有第一多核处理器62(其具有管理程序来控制多个处理器的操作)的高完整性系统60的实施例的框图。高完整性系统60包括第一多核处理器62、第二多核处理器64和第三多核处理器66。尽管本文描述多核处理器,高完整性系统60可包括具有任何适合数量的核(例如,1、2、4、8、16个,等)的处理器。此外,多核处理器62、64和66中的每个包括用于执行指令的应用程序处理器68、70和72,和配置成控制应用程序处理器68、70和72的指令执行并且执行分布式一致性算法的I/O处理器74。多核处理器62、64和66使用系统总线76和78

彼此通信并且与外部装置通信。尽管图示两个系统总线76和78,高完整性系统60可包括任何适合数量的总线(例如,1、2、4个,等)。如图示的,来自多核处理器62、64和66中的每个的应用程序处理器68可形成复制状态机80。此外,来自多核处理器62、64和66中的每个的应用程序处理器70可形成复制状态机82。此外,来自多核处理器62、64和66中的每个的应用程序处理器72可形成复制状态机84。在某些实施例中,从多核处理器62、64和66分配的处理器核的数量可改变,使得可分配小于总可用数量的处理器核、可从多核处理器62、64和66中的每个分配不同数量的处理器核,等等。在某些实施例中,高完整性应用可使用完整性检查而不是高可用性,并且可对高完整性应用仅分配两个处理器核。在其他实施例中,高完整性应用可通过从每个多核处理器62、64和66对高完整性应用分配超过三个的处理器核而实现高可用性。

[0060] 图4进一步图示第一多核处理器62的图。应该注意第二多核处理器64和第三多核处理器66可包括与第一多核处理器62相同的事物。如图示的,第一多核处理器62包括管理程序86(例如,虚拟机管理器(VMM))。在本实施例中,管理程序86是类型一管理程序;然而,在其他实施例中,可使用任何适合的管理程序(例如,类型二)。如可意识到的,类型一管理程序可以是直接在主机硬件上运行来控制硬件和管理客操作系统(例如,VMware vSphere、SPARC的Oracle VM服务器、Citrix XenServer、KVM、Microsoft Hyper-V管理程序,等等)的管理程序。此外,类型二管理程序可以是在常规操作系统环境(例如,BHyVe、VMware工作站、VirtualBox)内运行的管理程序。对于嵌入式实时系统,管理程序技术可基于在MILS RTOS或ARINC-653 RTOS(例如,Green Hills INTEGRITY和INTEGRITY-178、Wind River VxWorks MILS和653、LynuxWorks LynxSecure和LynxOS-178等)中使用的分离内核。第一多核处理器62还可包括芯片上资源88,其可提供给处理器68、70、72和74。例如,芯片上资源88可包括网络资源、分区资源、安全资源等等。

[0061] 应用程序处理器68(VM分区1)、应用程序处理器70(VM分区2)和应用程序处理器(VM分区3)可每个采用相似的样式配置。因此,应用程序处理器68、70和72每个使用与管理程序86交互来访问第一多核处理器62的硬件的客操作系统(客OS)90(例如,Linux、Windows、嵌入式RTOS,等)来操作。在某些实施例中,应用程序处理器68、70和72可每个使用“裸金属”环境(例如,Ada运行时期)而不是客OS 90来操作。此外,一个或多个应用程序92和94可由客OS 90执行来执行提供给客OS 90的指令。应用程序处理器68、70和72可包括“客户”,例如虚拟盘客户96和虚拟网络接口卡(NIC)客户98,其配置成访问对应的“服务器”(例如,驱动器)。此外,应用程序处理器68、70和72每个包括核结构100(例如,退休指令计数器(RIC)和时钟),其是在处理器68、70和72中实现的实函数。另外,应用程序处理器68、70和72还可包括虚拟时钟客户102,其用于访问虚拟时钟服务器。

[0062] 如图示的,I/O处理器74还包括客OS 90、虚拟盘96和虚拟NIC 98。此外,I/O处理器74配置成执行分布式一致性算法104。I/O处理器74还包括:盘驱动器110,其起到用于提供对盘资源(例如,经由虚拟盘客户96)的访问的服务器的作用;和网络接口卡(NIC)驱动器112,其起到用于提供对网络(例如,经由NIC客户98)的访问的服务器的作用。I/O处理器74配置成执行跨其他处理器68、70和72的分布式一致性逻辑。此外,I/O处理器74包括核结构14(例如,退休指令计数器(RIC)、时钟、NIC和盘)。此外,处理器74包括虚拟时钟服务器116,用于向虚拟时钟客户102提供虚拟时钟以便于处理器68、70和72之间的同步(例如,处理

巷)。因此, I/O处理器74配置成在应用程序处理器68、70和72与其他硬件之间对接, 由此使应用程序处理器68、70和72能够接收并且执行指令。此外, 如图示的, 处理器68、70、72和74中的每个包括客OS 90, 因此处理器可以独立地执行操作。应该注意应用程序处理器68、70和72可配置成作为没有隐变量(例如, 添加到被处理数据的信息)的复制状态机而操作。例如, 应用程序处理器68、70和72可配置成操作使得可接收信息(例如, 输入数据)、可执行指令并且以相同的方式(例如, 结果相同)从应用程序处理器68、70和72中的任一个输出结果。因此, 利用提供给应用程序处理器68、70和72的相同输入, 可获得相同的输出。此外, 输出可同步使得应用程序处理器68、70和72以同步的时间间隔开始和/或结束处理, 从而导致等同(例如, 一致)的结果而没有可观察的差异。

[0063] 图5是用于控制由处理器执行的指令的方法的实施例的定时图120。如图示的, 沿时间线122执行指令。第一时间同步信号124(例如, 时钟信号)在时间线122的左边图示。此外, 第二时间同步信号126朝时间线122的中心部分图示。持续时间128是第一时间同步信号124与第二时间同步信号126之间的时间。如可意识到的, 持续时间128可以是任何适合的持续时间。例如, 在某些实施例中, 持续时间128可以是近似 $250\mu\text{s} \pm 10\mu\text{s}$ 。此外, 第三时间同步信号130在时间线122的右边图示。持续时间132是第二时间同步信号126与第三时间同步信号130之间的时间差, 并且大体上与持续时间128相同。如可意识到的, 在定时图120中图示的相同的模式可在处理器操作期间重复。因此, 任何两个同步信号之间的持续时间可以是基于处理器、应用、时钟、定时等等的任何适合的时间(例如, 近似 $250\mu\text{s} \pm 10\mu\text{s}$)。

[0064] 在某些实施例中, 使在时间同步信号之间由处理器执行的指令的数量最大化而不在出现时间同步信号时执行指令, 这可是可期望的。因此, 在第二时间同步信号126之前停止在时间窗134内执行指令, 这可是可期望的。此外, 持续时间136(例如, 预定时段)图示第一时间同步信号124与时间窗134的起始时间之间的总时间。应该注意, 持续时间136小于持续时间128。在某些实施例中, 持续时间136可以是近似 $230\mu\text{s}$, 并且时间窗134的持续时间可以是近似 $20\mu\text{s}$ 。

[0065] 在一些实施例中, 由处理器执行的指令的总数量可受到控制。在一个示例中, 由处理器执行的指令的数量可遵循几何级数, 其中在每个迭代中执行的指令的数量粗略地是在之前迭代中执行的指令数量的一半。例如, 可计算处理器在持续时间136期间执行的指令的估计总数量(例如, 至少部分基于持续时间136)。在用于控制由处理器(例如, 处理器68、70、72)执行的指令的方法的第一迭代期间, 可执行估计总数量的指令的近似一半。例如, 在第一时间同步信号124和时间138之间, 可执行估计总数量的指令的近似一半。之后, 可(例如, 由处理器74)指示处理器(例如, 处理器68、70和72)在当前迭代期间迭代地执行多个指令(其小于在之前迭代中执行的较早指令数量)直到经过持续时间136。例如, 在时间138与时间140之间, 可指示处理器执行在第一时间同步信号124与时间138之间执行的指令的近似一半。此外, 在时间140与时间142之间, 可指示处理器执行在时间138与时间140之间执行的指令的近似一半。如图示的, 时间142不在时间窗134内。因此, 在时间142与时间144之间, 可指示处理器执行在时间140与时间142之间执行的指令的近似一半。时间144在时间窗134内, 因此, 经过持续时间136。如可意识到的, 处理器可在完成每个迭代后发送指示(例如, 指纹、到其他处理器的消息、要执行的额外指令的确定、生成定时器中断, 等)。

[0066] 在持续时间136期间执行的指令的总数量可采用任何适合的方式确定。例如, 在持

续时间136期间执行的指令的总数量可通过对在第一时间同步信号124与时间144之间执行的指令的数量求和(即,相加)而确定。作为另一个示例,在持续时间136期间执行的指令的总数量可通过访问(例如,读取)来自退休指令计数器(RIC)的计数而确定。

[0067] 在某些实施例中,在持续时间136期间执行的指令的总数量可小于期望的。因此,可指示处理器在经过持续时间136后执行计算数量的指令。相比之下,如果在持续时间136期间执行的指令的总数量大于或等于指令的期望总数量,处理器可在持续时间136后的时段期间延迟处理。

[0068] 再次,在第二时间同步信号126后的持续时间132期间,时间窗146可设置在第三时间同步信号130之前。在时间窗146内停止执行指令以控制在第三时间同步信号130之前执行的指令的数量,这可是可期望的。此外,持续时间148(例如,预定时段)图示第二时间同步信号126与时间窗146的起始时间之间的总时间。应该注意持续时间148小于持续时间132。

[0069] 同样,由处理器执行的指令的总数量可再次受到控制。例如,可计算(例如,至少部分基于持续时间148)预期处理器在持续时间148期间执行的指令的估计总数量。在用于第二时间同步信号126后控制由处理器执行的指令的方法的第一迭代期间,可执行估计总数量的指令的近似一半。例如,在第二时间同步信号126与时间150之间,可执行估计总数量的指令的近似一半。可指示处理器在当前迭代期间迭代地执行多个指令(其小于在之前迭代中执行的较早指令数量)直到经过持续时间148。例如,在时间150与时间152之间,可指示处理器执行在第二时间同步信号126与时间150之间执行的指令的近似一半。此外,在时间152与时间154之间,可指示处理器执行在时间150与时间152之间执行的指令的近似一半。如图示的,时间154不在时间窗146内。因此,在时间154与时间156之间,可指示处理器执行在时间152与时间154之间执行的指令的近似一半。时间156在时间窗146内,因此,经过持续时间148。如可意识到的,处理器可在完成每个迭代后发送指示并且执行额外的逻辑来使复制处理器之间的退休指令计数(RIC)同步。

[0070] 再次,在持续时间148期间执行的指令的总数量可采用任何适合的方式确定。例如,在持续时间148期间执行的指令的总数量可通过对在第二时间同步信号126与时间156之间执行的指令的数量求和(即,相加)而确定。作为另一个示例,在持续时间148期间执行的指令的总数量可通过访问来自RIC的计数而确定。

[0071] 在某些实施例中,在持续时间148期间执行的指令的总数量可小于期望的。因此,可指示处理器在经过持续时间148后执行计算数量的指令。例如,可指示处理器在时间156与时间158之间执行计算数量的指令。相比之下,如果在持续时间148期间执行的指令的总数量大于或等于指令的期望总数量,处理器可在持续时间148后的时段期间延迟处理。

[0072] 定时图120还可用于图示另一个实施例。例如,由处理器执行的指令的数量可基于剩余的持续时间而计算。例如,处理器在持续时间136期间执行的指令的估计总数量可基于持续时间136而计算。在用于控制由处理器(例如,处理器68、70、72)执行的指令的方法的第一迭代期间,可执行估计总数量的指令的近似一半。例如,在第一时间同步信号124处,可执行估计总数量的指令的近似一半。之后,可(例如,由处理器74)指示处理器(例如,处理器68、70、72)基于持续时间136的剩余时间的一半而执行估计数量的指令。此外,在时间140处,可指示处理器基于持续时间136的剩余时间的一半而执行估计数量的指令。如图示的,时间142不在时间窗134内。因此,在时间142处,可指示处理器基于持续时间136的剩余时间

的一半而执行估计数量的指令。时间144在时间窗134内,因此,经过持续时间136。如可意识到的,处理器可在它进入时间窗134后与其他处理器同步。为了与其他处理器同步,处理器的指纹可与另一个处理器的指纹比较。可如之前论述的那样确定在其期间执行所述多个指令的总时段。在某些实施例中,如果总时段小于预定时段,可指示处理器在经过持续时间136后执行第二数量的指令。此外,在一些实施例中,可指示处理器在持续时间136后延迟处理。使用上文描述的方法,可控制处理器以在某一时段内执行某一数量的指令。因此,因此,在时段内执行的指令的数量可被优化(例如,最大化)。此外,多个处理器可在一起控制来使处理器同步。

[0073] 图6是用于使多个处理器同步的方法的实施例的定时图180。因此,第一处理器的操作沿时间线182图示,第二处理器的操作沿时间线184图示,并且第三处理器的操作沿时间线186图示。此外,第一时间同步信号188(例如,时钟信号)在时间线182、184和186的左边图示。此外,第二时间同步信号190在时间线182、184和186的右边图示。另外,朝时间线182、184和186的右边的是时间段192,其中第一、第二和第三处理器的同步被证实和/或校正使得处理器在接收第二时间同步信号190时同步。

[0074] 持续时间194是第一时间同步信号188与第二时间同步信号190之间的时间。如可意识到的,持续时间194可以是任何适合的持续时间。例如,在某些实施例中,持续时间194可以是近似 $250\mu\text{s} \pm 10\mu\text{s}$ 。如可意识到的,在定时图180中图示的相同模式可在处理器操作期间如必要的那样(例如,在每个时间同步信号之间)经常重复。因此,两个连续同步信号之间的持续时间可以是任何适合的时间,例如基于处理器、应用、时钟、定时等等的时间(例如,近似 $250\mu\text{s} \pm 10\mu\text{s}$)。

[0075] 在某些实施例中,使在时间同步信号之间由处理器执行的指令的数量最大化并且使处理器彼此同步,这可是可期望的。因此,为了使处理器同步,在第二时间同步信号190之前的时间窗196、198和200内停止在每个处理器上执行指令,这可是可期望的。此外,持续时间202(例如,预定时段)图示第一时间同步信号188与时间窗196、198和200的起始时间之间的总时间。应该注意,持续时间202小于持续时间194。在某些实施例中,持续时间202可以是近似 $230\mu\text{s}$,并且时间窗196、198和200中的每个的持续时间可以是近似 $20\mu\text{s}$ 。

[0076] 在一些实施例中,由处理器执行的指令的总数量可受到控制。例如,可计算处理器中的每个在持续时间202期间执行的指令的估计总数量(例如,至少部分基于持续时间202)。如上文论述的,可使用任何适合的方法来确定在每个迭代期间执行的指令的数量。例如,指令的数量可基于几何级数(例如,之前的指令数量的近似一半)、剩余时间(例如,剩余时间窗的一半)等等。在用于控制由第一处理器执行的指令的方法的第一迭代期间,可执行估计总数量的指令的近似一半。例如,在第一时间同步信号188与时间204之间,可执行估计总数量的指令的近似一半。之后,可指示第一处理器在当前迭代期间迭代地执行多个指令(其小于在之前迭代中执行的较早指令数量)直到经过持续时间202。例如,在时间204与时间206之间(例如,在第二迭代期间),可指示第一处理器执行在第一时间同步信号188与时间204之间执行的指令的近似一半。此外,在时间206与时间208之间(例如,在第三迭代期间),可指示第一处理器执行在时间204与时间206之间执行的指令的近似一半。如图示的,时间208不在时间窗196内。因此,在时间208与时间210之间,可指示第一处理器通过执行在时间206与时间208之间执行的指令的近似一半而执行第四迭代。如图示的,时间210在时间

窗196内;因此,在执行四个迭代后经过持续时间202。如可意识到的,第一处理器可在完成每个迭代后发送指示(例如,指纹、到其他处理器的消息、要执行的额外指令的确定、生成定时器中断,等)。此外,第一处理器可在完成每个迭代后发送指纹(例如,RIC、存储器负载计数、存储器存储计数、支路指令计数、数据负载和存储、取支路,等等)。

[0077] 还可控制第二处理器以在经过持续时间202后停止执行指令。例如,在用于控制由第二处理器执行的指令的方法的第一迭代期间,可执行估计总数量的指令的近似一半。具体地,在第一时间同步信号188与时间212之间,可执行估计总数量的指令的近似一半。之后,可指示第二处理器在当前迭代期间迭代地执行多个指令(其小于在之前迭代中执行的较早指令数量)直到经过持续时间202。例如,在时间212与时间214之间(例如,在第二迭代期间),可指示第二处理器执行在第一时间同步信号188与时间212之间执行的指令的近似一半。此外,在时间214与时间216之间(例如,在第三迭代期间),可指示第二处理器执行在时间212与时间214之间执行的指令的近似一半。此外,在时间216与时间218之间(例如,在第四迭代期间),可指示第二处理器执行在时间214与时间216之间执行的指令的近似一半。如图示的,时间218不在时间窗198内。因此,在时间218与时间220之间,可指示第二处理器通过执行在时间216与时间218之间执行的指令的近似一半而执行第五迭代。如图示的,时间220在时间窗198内;因此,在执行五个迭代后经过持续时间202。与第一处理器相似,第二处理器可在完成每个迭代后发送指示(例如,指纹、到其他处理器的消息、要执行的额外指令的确定、生成定时器中断,等)。

[0078] 还可控制第三处理器以在经过持续时间202后停止执行指令。例如,在用于控制由第三处理器执行的指令的方法的第一迭代期间,可执行估计总数量的指令的近似一半。具体地,在第一时间同步信号188与时间222之间,可执行估计总数量的指令的近似一半。之后,可指示第三处理器在当前迭代期间迭代地执行多个指令(其小于在之前迭代中执行的较早指令数量)直到经过持续时间202。例如,在时间222与时间224之间(例如,在第二迭代期间),可指示第三处理器执行在第一时间同步信号188与时间222之间执行的指令的近似一半。此外,在时间224与时间226之间(例如,在第三迭代期间),可指示第三处理器执行在时间222与时间224之间执行的指令的近似一半。此外,在时间226与时间228之间(例如,在第四迭代期间),可指示第三处理器执行在时间224与时间226之间执行的指令的近似一半。另外,在时间228与时间230之间(例如,在第五迭代期间),可指示第三处理器执行在时间226与时间228之间执行的指令的近似一半。如图示的,时间230不在时间窗200内。因此,在时间230与时间232之间,可指示第三处理器通过执行在时间228与时间230之间执行的指令的近似一半而执行第六迭代。如图示的,时间232在时间窗200内;因此,在执行六个迭代后经过持续时间202。与第一和第二处理器相似,第三处理器可在完成每个迭代后发送指示和/或指纹。

[0079] 在持续时间202期间由第一、第二和/或第三处理器执行的指令的总数量可采用任何适合的方式确定。例如,在持续时间202期间执行的指令的总数量可通过对在第一时间同步信号188与最后的迭代结束时的时间(例如,时间210、220、232)之间执行的指令的数量求和(即,相加)而确定。作为另一个示例,在持续时间202期间执行的指令的总数量可通过访问(例如,读取)来自RIC的计数而确定。

[0080] 在某些实施例中,在持续时间202期间由第一、第二和第三处理器执行的指令的总

数量可小于期望的。因此,可指示处理器在经过持续时间202后执行计算数量的指令。相比之下,如果在持续时间202期间执行的指令的总数量大于或等于指令的期望总数量,处理器可在持续时间202后的时段期间延迟处理。

[0081] 例如,如上文描述的,在持续时间202期间,第一处理器开始四个迭代,第二处理器开始五个迭代,并且第三处理器开始六个迭代。为了该说明目的,第一、第二和第三处理器可以是能够每秒执行二十亿个指令的1.0GHz处理器。此外,为了该说明,处理器配置成使用如上文描述的几何方法来操作。因此,要在持续时间202(例如,230 μ s)期间执行的估计总数量的指令可以是近似460,000个指令(例如,2,000,000,000 $\times$ 230 μ s)。因此,在第一迭代期间,可执行指令的总数量的近似一半;因此,在第一迭代期间,可执行近似230,000个指令。此外,在每个后续迭代期间,可执行在之前迭代中执行的指令的近似一半。如此,在第二迭代期间,可执行近似115,000个指令。此外,在第三迭代期间,可执行近似57,500个指令。此外,在第四迭代期间,可执行近似28,750个指令。在第五迭代期间,可执行近似14,375个指令,并且在第六迭代期间,可执行近似7,188个指令。如可意识到的,每个处理器的指纹应该在完成每个迭代后匹配。

[0082] 因此,到在第一处理器在窗196内停止迭代的时候(例如,在第四迭代后),第一处理器可执行近似431,250个指令(例如,230,000+115,000+57,500+28,750)。此外,到第二处理器在窗198内停止迭代的时候(例如,在第五迭代后),第二处理器可执行近似445,625个指令(例如,230,000+115,000+57,500+28,750+14,375)。此外,到第三处理器在窗200内停止迭代的时候(例如,在第六迭代后),第三处理器可执行近似452,813个指令(例如,230,000+115,000+57,500+28,750+14,375+7,188)。应该注意,第一、第二和第三处理器可基于任何数量的变量(例如,处理器性能中的变化)在给定的时段内执行不同数量的指令。

[0083] 如此,第一、第二和第三处理器中的每个可不同步(例如,可不执行相同数量的指令)。因此,第一处理器可执行近似21,563个指令(例如,14,375+7,188)来赶上第三处理器。此外,第二处理器可执行近似7,188个指令来赶上第三处理器。因此,在时间窗196和198期间,可指示第一和第二处理器执行额外的指令使得到时间窗196和198结束为止第一和第二处理器已经完成与第三处理器相同数量的指令。此外,在时间窗200的剩余部分期间可使第三处理器延迟以允许第一和第二处理器有时间执行指令。应该注意,第一和/或第二处理器可在第三处理器完成它的第六迭代之前开始执行额外的指令使得有足够的时间来执行指令。

[0084] 作为另一个示例,再次在持续时间202期间,第一处理器开始四个迭代,第二处理器开始五个迭代,并且第三处理器开始六个迭代。同样,为了该说明目的,第一、第二和第三处理器可以是能够每秒执行二十亿个指令的1.0GHz处理器。此外,为了该说明,处理器配置成使用基于如上文描述的几何方法来操作。因此,要在持续时间202(例如,230 μ s)期间执行的估计总数量的指令可以是近似460,000个指令(例如,2,000,000,000 $\times$ 230 μ s)。因此,在第一迭代指令期间,可对近似一半的总持续时间执行;因此,在第一迭代期间,可由每个处理器执行近似230,000个指令(例如,2,000,000,000 $\times$ 115 μ s)。此外,在每个后续迭代期间,处理器中的每个可基于它们相应的剩余时间执行指令。

[0085] 如此,在第二迭代期间,第一处理器可确定它具有近似108 μ s剩余;因此,第一处理器可执行近似108,000个指令(例如,2,000,000,000 $\times$ 54 μ s)以对近似一半的剩余时间执

行指令。此外,在第三迭代期间,第一处理器可确定它具有近似 $52\mu\text{s}$ 剩余;因此,第一处理器可执行近似52,000个指令(例如, $2,000,000,000 \times 26\mu\text{s}$)以对近似一半的剩余时间执行指令。另外,在第四迭代期间,第一处理器可确定它具有近似 $24\mu\text{s}$ 剩余;因此,第一处理器可执行近似24,000个指令(例如, $2,000,000,000 \times 12\mu\text{s}$)以对近似一半的剩余时间执行指令。因此,到第一处理器在窗196内停止迭代的时候,第一处理器可执行近似414,000个指令(例如, $230,000+108,000+52,000+24,000$)。

[0086] 转向第二处理器,在第二迭代期间,第二处理器可确定它具有近似 $110\mu\text{s}$ 剩余;因此,第二处理器可执行近似110,000个指令(例如, $2,000,000,000 \times 55\mu\text{s}$)以对近似一半的剩余时间执行指令。此外,在第三迭代期间,第二处理器可确定它具有近似 $54\mu\text{s}$ 剩余;因此,第二处理器可执行近似54,000个指令(例如, $2,000,000,000 \times 27\mu\text{s}$)以对近似一半的剩余时间执行指令。另外,在第四迭代期间,第二处理器可确定它具有近似 $24\mu\text{s}$ 剩余;因此,第二处理器可执行近似24,000个指令(例如, $2,000,000,000 \times 12\mu\text{s}$)以对近似一半的剩余时间执行指令。此外,在第五迭代期间,第二处理器可确定它具有近似 $10\mu\text{s}$ 剩余;因此,第二处理器可执行近似10,000个指令(例如, $2,000,000,000 \times 5\mu\text{s}$)以对近似一半的剩余时间执行指令。因此,到第二处理器在窗198内停止迭代的时候,第二处理器可执行近似428,000个指令(例如, $230,000+110,000+54,000+24,000+10,000$)。

[0087] 转向第三处理器,在第二迭代期间,第三处理器可确定它具有近似 $114\mu\text{s}$ 剩余;因此,第三处理器可执行近似114,000个指令(例如, $2,000,000,000 \times 57\mu\text{s}$)以对近似一半的剩余时间执行指令。此外,在第三迭代期间,第三处理器可确定它具有近似 $58\mu\text{s}$ 剩余;因此,第三处理器可执行近似58,000个指令(例如, $2,000,000,000 \times 29\mu\text{s}$)以对近似一半的剩余时间执行指令。另外,在第四迭代期间,第三处理器可确定它具有近似 $28\mu\text{s}$ 剩余;因此,第三处理器可执行近似28,000个指令(例如, $2,000,000,000 \times 14\mu\text{s}$)以对近似一半的剩余时间执行指令。此外,在第五迭代期间,第三处理器可确定它具有近似 $14\mu\text{s}$ 剩余;因此,第三处理器可执行近似14,000个指令(例如, $2,000,000,000 \times 7\mu\text{s}$)以对近似一半的剩余时间执行指令。此外,在第六迭代期间,第三处理器可确定它具有近似 $6\mu\text{s}$ 剩余;因此,第三处理器可执行近似6,000个指令(例如, $2,000,000,000 \times 3\mu\text{s}$)以对近似一半的剩余时间执行指令。因此,到第三处理器在窗200内停止迭代的时候,第三处理器可执行近似450,000个指令(例如, $230,000+114,000+58,000+28,000+14,000+6,000$)。如可意识到的,每个处理器的指纹可在完成每个迭代后不匹配。

[0088] 如此,第一、第二和第三处理器中的每个可不同步(例如,可不执行相同数量的指令)。因此,第一处理器可执行近似36,000个指令(例如, $450,000-414,000$)来赶上第三处理器。此外,第二处理器可执行近似22,000(例如, $450,000-428,000$)个指令来赶上第三处理器。因此,在时间窗196和198期间,可指示第一和第二处理器执行额外的指令使得时间窗196和198结束为止第一和第二处理器已经完成与第三处理器相同数量的指令。此外,在时间窗200的剩余部分期间可使第三处理器延迟以允许第一和第二处理器有时间执行指令。应该注意,第一和/或第二处理器可在第三处理器完成它的第六迭代之前开始执行额外的指令使得有足够的时间来执行指令。在处理器同步后,每个处理器的指纹应该匹配。例如,退休指令计数(RIC)应该匹配。

[0089] 图7是用于控制由处理器(例如,处理器68、70、72)执行的指令的方法240的实施例

的流程图。指示处理器来执行指令(框242)。例如,可由另一个处理器指示处理器基于用于执行的时段或基于预定数量的指令来执行多个指令。在一个实施例中,可指示处理器执行以下指令:其对应于预期要在当前时间与未来的时间窗(例如,时间窗134、136)(其中期望停止执行指令)开始之间执行的指令的近似一半。在处理器已经执行指令后,做出关于当前时间是否在时间窗内的确定(框244)。如果当前时间不在时间窗内,方法回到框242,其中指示处理器执行额外的指令。如可意识到的,在每个迭代期间执行的指令的数量可大于每个后续迭代中的指令的数量。回到框244,如果当前时间在时间窗内,确定执行的指令的总数量(框246)。

[0090] 本发明的技术效果包括能够控制在单个处理器上的时间同步信号之间执行的指令的数量。此外,在具有多个处理器的系统中,可控制处理器使得每个处理器在连续同步信号之间执行相同数量的指令。因此,处理器可同步并且可适合于在高完整性和/或高可用性系统中使用。

[0091] 该书面描述使用示例来公开本发明,其包括最佳模式,并且还使本领域内任何技术人员能够实践本发明,包括制作和使用任何装置或系统和执行任何包含的方法。本发明的专利范围由权利要求限定,并且可包括本领域内技术人员想到的其他示例。这样的其他示例如果其具有不与权利要求的文字语言不同的结构元件,或者如果其包括与权利要求的文字语言无实质区别的等同结构元件则意在权利要求的范围内。

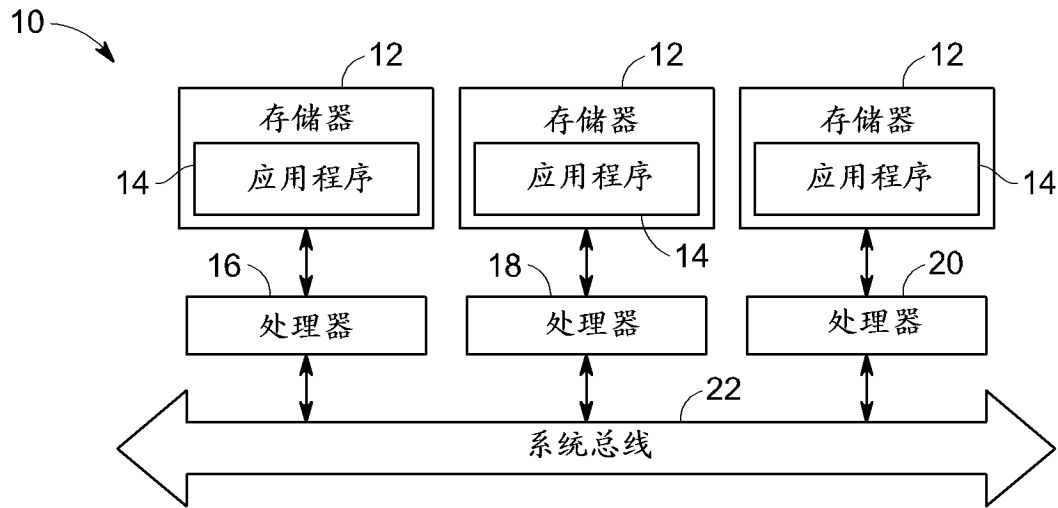


图 1

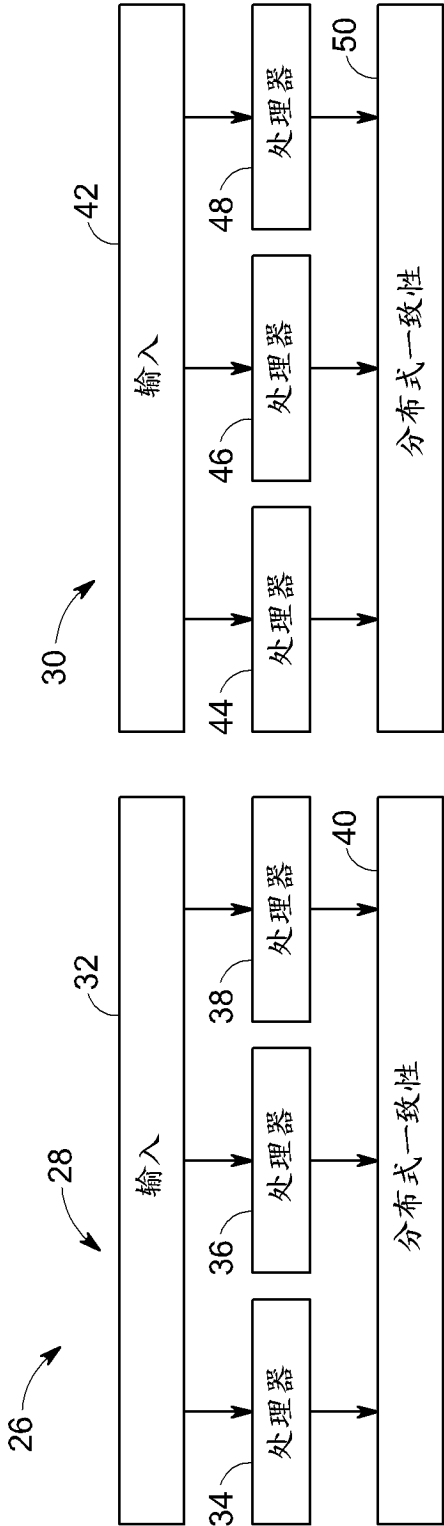


图 2

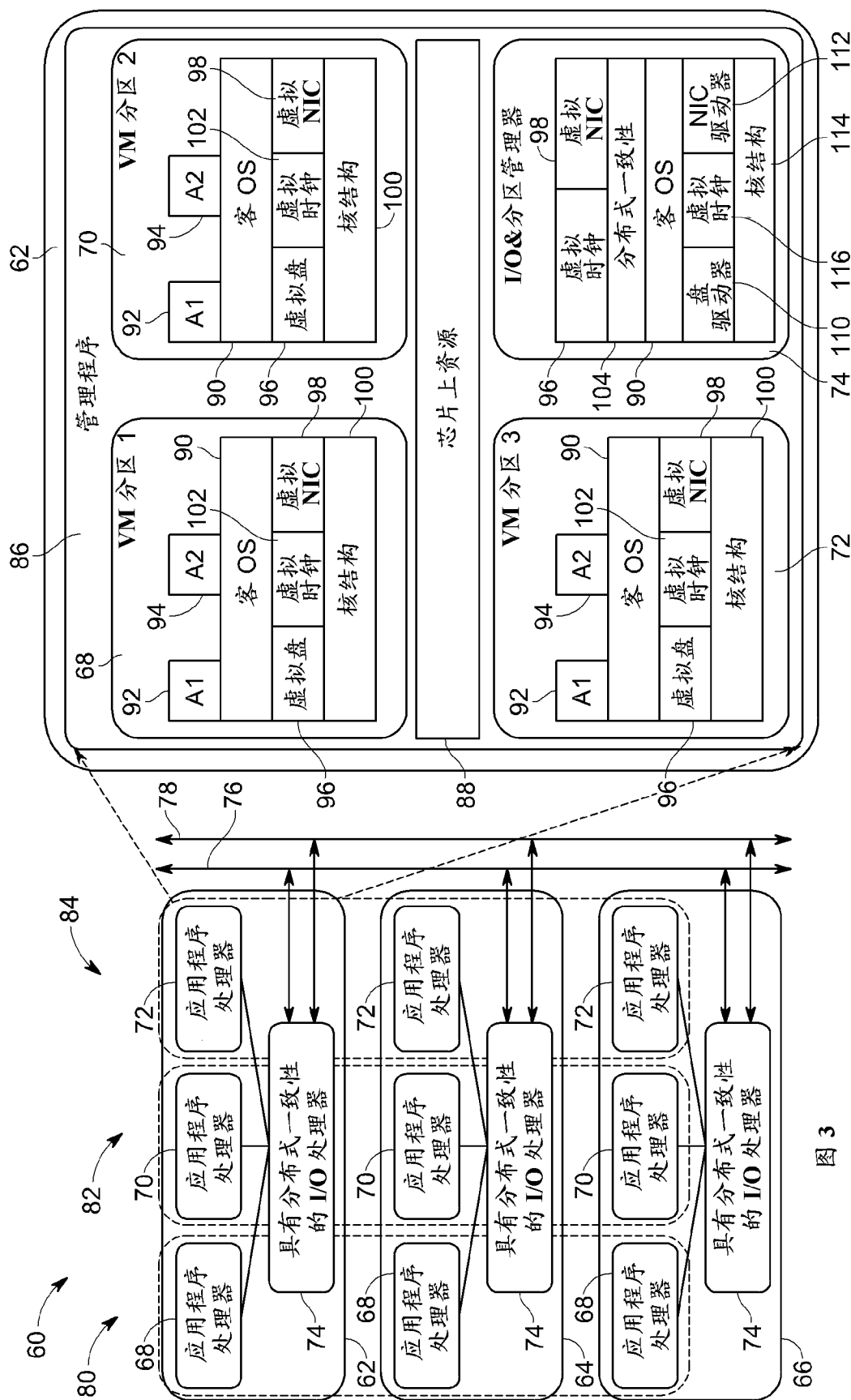


图 4

图 3

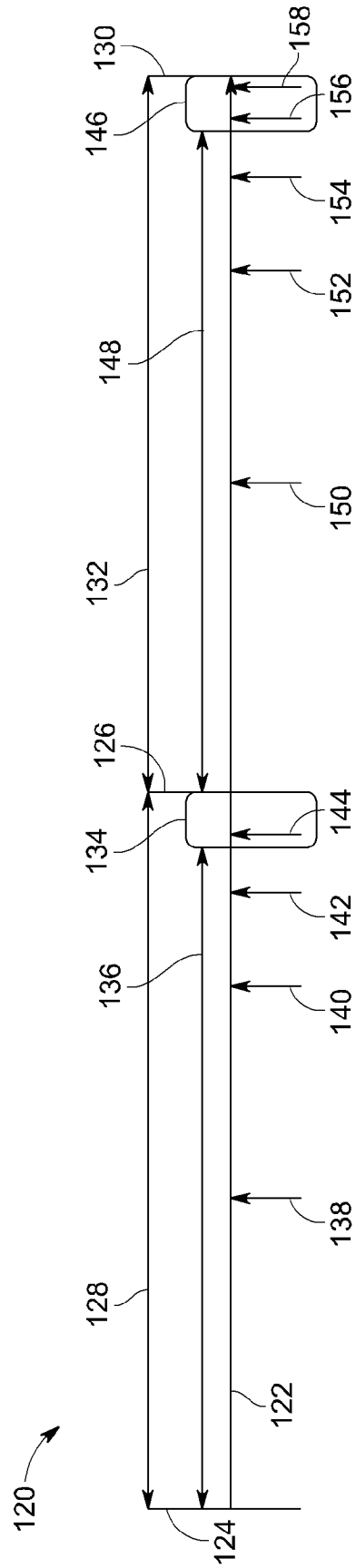


图 5

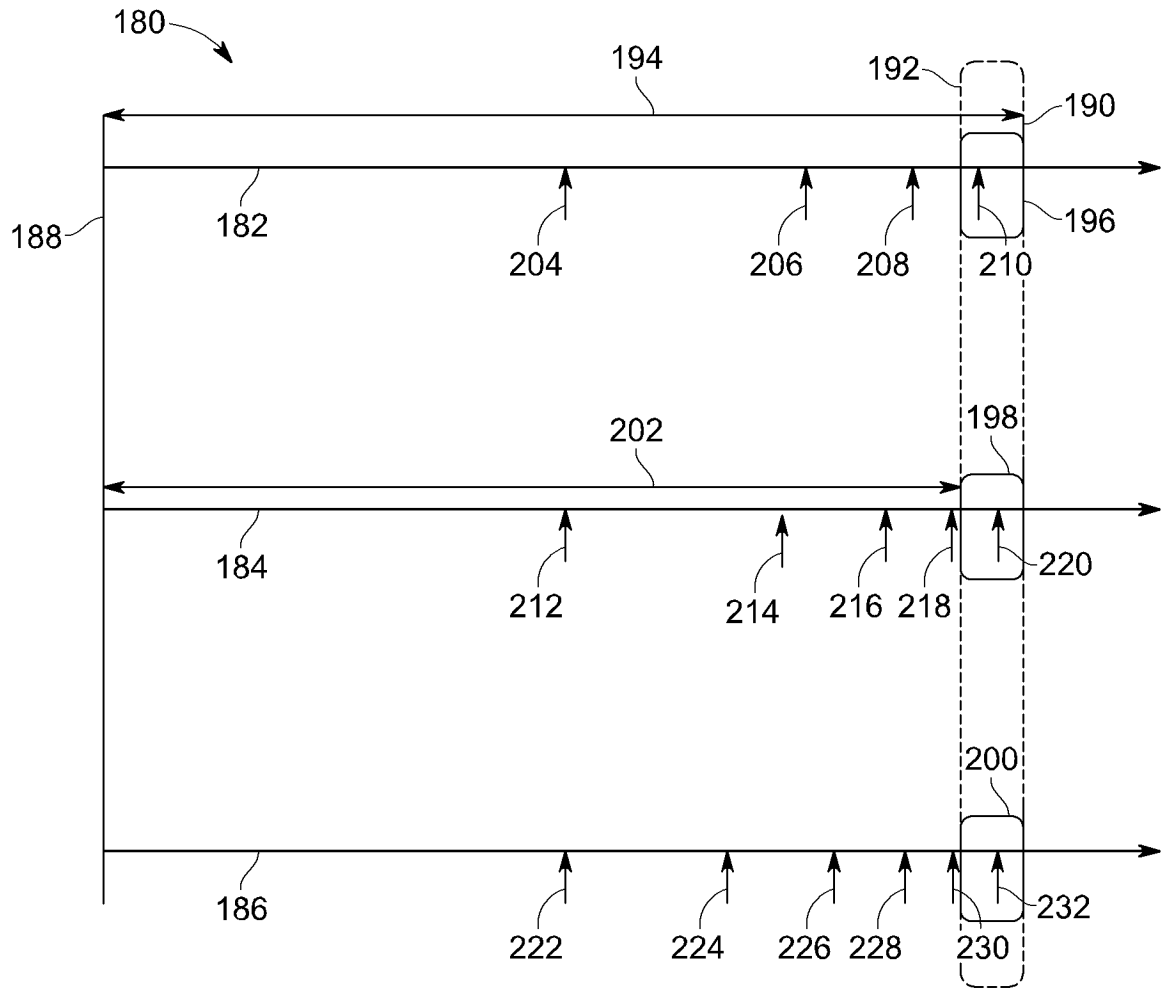


图 6

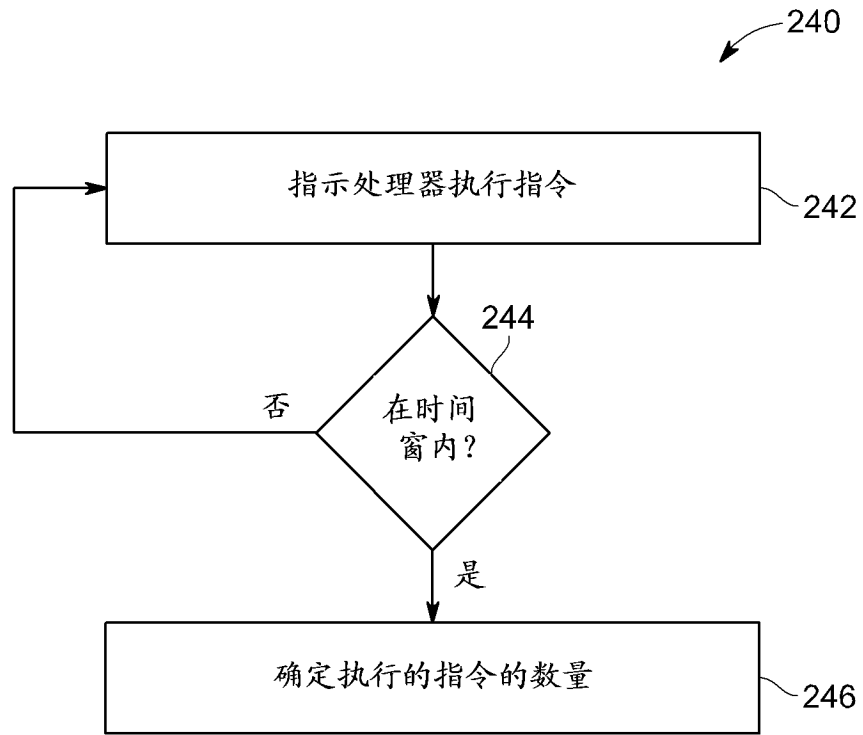


图 7