



# [12] 发明专利申请公布说明书

[21] 申请号 200680021365.6

[43] 公开日 2008年7月2日

[11] 公开号 CN 101213522A

[22] 申请日 2006.6.29  
 [21] 申请号 200680021365.6  
 [30] 优先权  
     [32] 2005.6.30 [33] US [31] 11/173,835  
 [86] 国际申请 PCT/US2006/025959 2006.6.29  
 [87] 国际公布 WO2007/005818 英 2007.1.11  
 [85] 进入国家阶段日期 2007.12.14  
 [71] 申请人 英特尔公司  
     地址 美国加利福尼亚  
 [72] 发明人 P·B·拉库纳斯 J·S·埃默尔  
     A·比斯瓦斯 S·S·慕克吉  
     S·E·拉希

[74] 专利代理机构 永新专利商标代理有限公司  
 代理人 王 英

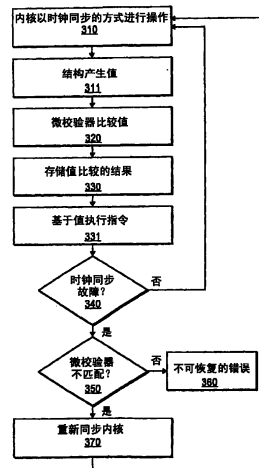
权利要求书4页 说明书11页 附图6页

## [54] 发明名称

降低时钟同步双模冗余系统中的不可纠正错误率

## [57] 摘要

公开了用于降低时钟同步双模冗余系统中的不可纠正错误率的装置和方法的实施例。在一个实施例中，该装置包括两个处理器内核、微校验器、全局校验器以及故障逻辑。微校验器用于检测来自一个内核中的结构的值是否与来自另一内核中的相应结构的值相匹配。全局校验器用于检测两个内核之间的时钟同步故障。故障逻辑用于如果存在时钟同步故障，而校验器检测到不匹配，则使这两个内核重新同步。



- 1、一种装置，包括：
  - 包括第一结构的第一内核；
  - 包括第二结构的第二内核；
  - 微校验器，用于检测来自所述第一结构的第一值是否与来自所述第二结构的第二值相匹配；
  - 全局校验器，用于检测所述第一内核与所述第二内核之间的时钟同步故障；以及
  - 故障逻辑，用于如果所述全局校验器检测到所述时钟同步故障且所述微校验器检测到所述第一值与所述第二值之间的不匹配，则使所述第一内核和所述第二内核重新同步。
  
- 2、根据权利要求 1 所述的装置，其中所述微校验器包括用于将所述第一值与所述第二值相比较的比较器。
  
- 3、根据权利要求 1 所述的装置，其中所述全局校验器包括用于将所述第一内核的第一输出与所述第二内核的第二输出相比较的比较器。
  
- 4、根据权利要求 1 所述的装置，其中所述故障逻辑还用于如果所述全局校验器检测到所述时钟同步故障且所述微校验器检测到所述第一值与所述第二值相匹配，则指示检测到不可纠正的错误。
  
- 5、根据权利要求 1 所述的装置，其中：
  - 所述第一内核还包括第三结构和第四结构；
  - 所述第二内核还包括第五结构和第六结构；
  - 所述第一结构包括第一指纹逻辑，以基于来自所述第三结构的第三值和来自所述第四结构的第四值产生所述第一值；并且
  - 所述第二结构包括第二指纹逻辑，以基于来自所述第五结构的第

五值和来自所述第六结构的第六值产生所述第二值。

6、根据权利要求1所述的装置，其中：

所述第一内核的架构状态与所述第一值无关；并且  
所述第二内核的架构状态与所述第二值无关。

7、根据权利要求6所述的装置，其中：

所述第一结构是第一预测结构；并且  
所述第二结构是第二预测结构。

8、根据权利要求1所述的装置，其中所述故障逻辑还用于如果所述全局校验器检测到所述时钟同步故障且所述微校验器检测到所述不匹配，则使所述第一值和所述第二值被重新产生。

9、根据权利要求8所述的装置，其中：

所述第一结构是第一高速缓存；  
所述第一结果是第一高速缓存项；  
所述第二结构是第二高速缓存；以及  
所述第二结果是第二高速缓存项。

10、根据权利要求9所述的装置，其中所述故障逻辑还用于如果所述全局校验器检测到所述时钟同步故障且所述微校验器检测到所述不匹配，则使所述第一高速缓存项和所述第二高速缓存项被重新加载。

11、一种方法，包括：

校验来自第一内核中的第一结构的第一值是否与来自第二内核中的第二结构的第二值相匹配；

检测所述第一内核与所述第二内核之间的时钟同步故障；以及  
如果检测到所述第一值与所述第二值之间的不匹配，则使所述第

一内核和所述第二内核重新同步。

12、根据权利要求 11 所述的方法，还包括如果所述第一值与所述第二值相匹配，则指示检测到不可纠正的错误。

13、根据权利要求 12 所述的方法，还包括：

基于来自所述第一内核中的第三结构的第三值和来自所述第一内核中的第四结构的第四值产生所述第一值；以及

基于来自所述第二内核中的第五结构的第五值和来自所述第二内核中的第六结构的第六值产生所述第二值。

14、根据权利要求 13 所述的方法，其中：

产生所述第一值包括基于所述第三值和所述第四值产生校验和；  
并且

产生所述第二值包括基于所述第五值和所述第六值产生校验和。

15、根据权利要求 11 所述的方法，还包括：

预测第一指令是否将由所述第一内核基于所述第一值来执行；以  
及

预测第二指令是否将由所述第二内核基于所述第二值来执行。

16、根据权利要求 11 所述的方法，还包括如果检测到所述不匹配，则重新产生所述第一值和所述第二值。

17、根据权利要求 16 所述的方法，还包括：

将所述第一值与所重新产生的第一值相比较；

将所述第二值与所重新产生的第二值相比较；

如果所述第二值与所重新产生的第二值相匹配，则使所述第一内核与所述第二内核同步；以及

如果所述第一值与所重新产生的第一值相匹配，则使所述第二内

核与所述第一内核同步。

18、根据权利要求 16 所述的方法，其中所述第一结构是第一高速缓存，所述第一值是第一高速缓存项，所述第二结构是第二高速缓存，并且所述第二值是第二高速缓存项，其中重新产生所述第一值和所述第二值包括重新加载所述第一高速缓存项和所述第二高速缓存项。

19、一种系统，包括：

动态随机存取存储器；

包括第一结构的第一内核；

包括第二结构的第二内核；

微校验器，用于检测来自所述第一结构的第一值是否与来自所述第二结构的第二值相匹配；

全局校验器，用于检测所述第一内核与所述第二内核之间的时钟同步故障；以及

故障逻辑，用于如果所述全局校验器检测到所述时钟同步故障且所述微校验器检测到所述第一值和所述第二值之间的不匹配，则使所述第一内核与所述第二内核重新同步。

## 降低时钟同步双模冗余系统中的不可纠正错误率

### 技术领域

本发明涉及数据处理领域，尤其涉及在数据处理装置中抑制错误的领域。

### 背景技术

随着集成电路制造技术的改进不断使微处理器和其他数据处理装置具有更小的尺寸和更低的工作电压，这些设备的制造商和用户变得更加关心软错误现象。当 $\alpha$ 粒子和高能中子碰撞集成电路并改变存储在电路节点上的电荷时，产生软错误。如果电荷改变足够大，则节点上的电压可能从表示一种逻辑状态的电平改变为表示一种不同的逻辑状态的电平，在这种情况下，存储在该节点上的信息被破坏。通常，软错误率随着电路尺寸的减小而增加，这是因为当电路密度增加时碰撞粒子击中电压节点的可能性增加。同样，随着工作电压减小，表示不同逻辑状态的电压电平之间的差值减小，因此改变电路节点上的逻辑状态需要更少的能量，并且产生更多的软错误。

阻挡导致软错误的粒子是极其困难的，因此数据处理装置常常包括用于检测并且有时用于纠正软错误的技术。这些错误抑制技术包括双模冗余（“DMR”）和三模冗余（“TMR”）。利用 DMR，两个相同的处理器或处理器内核以时钟同步（lockstep）的方式执行相同的程序，并且对它们的结果进行比较。利用 TMR，以时钟同步的方式运行三个相同的处理器。

使用 DMR 或 TMR 可以检测任何一个处理器中的错误，这是因为错误将导致结果出现差异。TMR 提供了如下优点，即可以通过假定三个处理器当中的两个处理器的匹配结果是正确的结果来实现从错误中的恢复。

通过在将所有结果提交给寄存器或者所有结果被允许影响系统的架构状态（architectural state）之前校验所有的结果，DMR 系统中的恢复也是可行的。然后，如果检测到错误，则通过重新执行自最后一个检查点以来的所有指令，可以实现恢复。然而，由于延迟时间或其他设计限制，这种方法并不实用。另一种方法是加入回退机制，如果检测到错误，其允许恢复原来的架构状态。由于设计的复杂性，这种方法也不实用，并且可能存在以下问题：，由于发生非确定性事件，例如异步中断或不等幂的输出操作的重新执行，而使从先前状态重新执行的结果可能与原始结果不同。

另外，因为 DMR 和 TMR 的实现需要会受软错误影响的附加电路，并且因为它们可以检测到不然会变为未被检测到的但又不会引起系统故障的错误，所以 DMR 和 TMR 实际上可能增加错误率。例如，用于进行预测的结构中的错误可能导致不正确的预测，其中所述结构用于预测应该推测性地执行程序的那个分支，但是当最终评定分支条件时，处理器将自动恢复。

## 附图说明

借助实例说明本发明，但不限于附图。

图 1 示出本发明在多核处理器中的实施例；

图 2 示出使用微校验指纹逻辑（microcheck fingerprint logic）来降低内核间带宽（cross-core bandwidth）的本发明的实施例；

图 3 示出本发明在用于降低时钟同步双模冗余系统中的不可纠正错误率的方法中的实施例；

图 4 示出本发明在用于降低时钟同步双模冗余系统中的不可纠正错误率的方法中的另一实施例；

图 5 示出本发明在用于降低时钟同步双模冗余系统中的不可纠正错误率的方法中的另一实施例；

图 6 示出本发明在时钟同步双模冗余系统中的实施例。

## 具体实施方式

以下说明用于降低时钟同步双模冗余系统中的不可纠正错误率的装置和方法的实施例。在以下说明中，为了对本发明有更加全面的理解，阐明了许多具体的细节，例如部件和系统结构。然而，本领域技术人员应该理解的是，可以在没有这些具体细节的情况下实施本发明。另外，没有对一些公知的结构、电路、技术等进行详细的说明，以避免不必要地使本发明含混不清。

DMR 可以用于提供错误检测和纠正。然而，通过检测到不会引起系统故障的错误，DMR 也可能增加错误率。本发明的实施例可以通过使用微校验器检测这种“假”错误以便将它们忽略，来降低 DMR 系统中的错误率。其他实施例可以通过如下方式来降低 DMR 系统中的错误率：将微校验器用于特定结构，例如高速缓存，为其重新产生值并将该值与原始值进行比较以确定两个处理器中的哪一个应该与另一个处理器的状态同步，由此避免完全回退机制的成本。本发明的这些实施例是可取的，其提供了 DMR 的一些益处（例如，错误检测和纠正能力），同时减少了一些缺点（例如，假错误、完全恢复能力的成本）。

而且，本发明的实施例是可取的，其避免了用奇偶校验或纠错码机制保护特定的结构，所述机制可能很昂贵，并且对于不能破坏架构状态的结构也可能是不必要的。根据本发明的实施例，将这些结构连接到微校验器可以在不需要通过奇偶校验或其他方式确定两个 DMR 内核中的哪一个出现了错误的情况下，提供从错误中恢复的能力。

图 1 示出本发明在多核处理器 100 中的实施例。通常，多核处理器是包括一个以上执行内核的单个集成电路。执行内核包括用于执行指令的逻辑。除执行内核之外，多核处理器可以包括在本发明范围内的专用或共享资源的任何组合。专用资源可以是专用于单内核的资源，例如专用的一级高速缓存，或者可以是专用于内核的任何子集的资源。共享资源可以是由所有内核共享的资源，例如支持多核处理器和另一部件之间的接口的共享外部总线单元或共享的二级高速缓存，或者可以是由内核的任何子集共享的资源。本发明还在多核处理器之外的装置中实现，例如在具有至少两个处理器的多处理器系统中实



现，其中每个所述处理器具有至少一个内核。

处理器 100 包括内核 110 和内核 120。内核 110 和 120 可以基于各种不同类型的处理器中的任何一种的设计，例如 Pentium®处理器系列、Itanium®处理器系列、或来自 Intel 公司的其他处理器系列中的处理器，或者来自另一公司的另一处理器。处理器 100 还包括全局校验器 130 和微校验器 140。

根据任何公知的用于检测 DMR 系统中的时钟同步故障的技术，例如利用比较器电路，全局校验器 130 将来自内核 110 的输出与来自内核 120 的输出进行比较。例如，当内核 110 和 120 同步运行具有相同输入的不同程序副本时，可以对内核 110 和 120 的输出进行比较。

内核 110 包括结构 111，其可以是任何电路、逻辑、功能块、模块、单元或其他结构，所述其他结构当内核 110 和 120 以时钟同步的方式操作时产生或保持与来自包括在内核 120 中的相应结构 121 的相应值相匹配的值。

在一个实施例中，结构 111 和 121 可以是不能改变处理器 100 或包括处理器 100 的系统的架构状态的结构。例如，结构 111 和 121 可以是预测结构，例如条件分支预测器、跳转预测器、返回地址预测器、或存储器相关性（memory dependence）预测器。

在另一实施例中，结构 111 和 121 可以是其内容在包括处理器 100 的系统中的其他处被复制、或被重新产生的结构。例如，结构 111 和 121 可以是高速缓存结构，其中每个未修改的高速缓存线或项是可以通过从系统中的更高级高速缓存或其他存储器中重新加载高速缓存线或项来重新产生的值。

微校验器 140 将来自结构 111 的值与来自结构 121 的相应值相比较。在不同的实施例中，所比较的值可以根据结构 111 和 112 的性质而不同，并且例如可以是指示是应该进行条件分支还是应该发生跳转的单个比特、多比特预测返回地址、或多比特高速缓存线或项。因此，微校验器 140 的性质在不同的实施例中可以不同，并且可以根据任何公知的技术，例如利用专用电路或门电路或比较器电路，来进行比较。

在一个实施例中，可以将微校验器 140 配置为保留其比较结果至

少直到时钟同步的程序执行已经到达这样一个点为止,即在该点全局校验器 130 所检测到的时钟同步故障不能被归因于在微校验器 140 所比较的值之间的不匹配。例如,如果微校验器是组合逻辑,并且所比较的值保持静态至少直到到达每个时钟同步故障检测点为止,则可以在没有任何特殊存储元件的情况下实现微校验器 140 的这种配置,或者可以利用寄存器或其他存储微校验器 140 的结果的存储元件来实现微校验器 140 的这种配置。在其他实施例中,不需要将微校验器配置为保留其比较结果。

处理器 100 还包括故障逻辑 150。故障逻辑 150 可以是任何硬件、微代码、可编程逻辑、处理器抽象层、固件、软件、或其他命令处理器 100 对全局校验器 130 检测到时钟同步故障进行响应的逻辑。在全局校验器 130 检测到时钟同步故障时,如果微校验器 140 已经检测到来自结构 111 的值和来自结构 121 的相应值之间的不匹配,则故障逻辑 150 使内核 110 和内核 120 如下所述那样被重新同步。然而,如果微校验器 140 没有检测到来自结构 111 的值和来自结构 121 的相应值之间的不匹配,则根据任何公知的指示系统故障的方法,例如报告故障码和中断操作,故障逻辑 150 指示检测到不可纠正的错误。

尽管图 1 仅仅示出内核 110 中的结构 111 和内核 120 中的结构 121 向微校验器 140 提供输入,但是在本发明的范围内可以使用任何数量的结构和微校验器。例如,图 2 示出使用每个内核多个结构、单个微校验器和用于降低内核间带宽的指纹逻辑的本发明的实施例。

在图 2 中,处理器 200 包括内核 210 和 220、全局校验器 230、微校验器 240 以及故障逻辑 250。内核 210 包括结构 211、213 和 215,并且处理器内核 220 包括结构 221、223 和 225。

结构 211 包括根据来自结构 213 和 215 的值产生指纹的指纹逻辑 212,其中结构 213 和 215 可以是如以上相对于图 1 中的结构 111 所述的任何结构。同样,结构 221 包括根据与指纹逻辑 212 所用的方法相同的方法、基于来自结构 223 和 225 的值产生指纹的指纹逻辑 222。

指纹逻辑 212 和指纹逻辑 222 可以利用任何公知的将两个或多个值组合成单个值的方法(例如,使用循环冗余校验器产生校验和)来

实现。可以使用指纹逻辑 212 和指纹逻辑 222，以便微校验器 240 可以检测结构 213 与 223 以及结构 215 与 225 之间的不匹配，而不是对于结构 213 和 223 使用一个微校验器，对于结构 215 和 225 使用另一个微校验器。

指纹逻辑 212 和指纹逻辑 222 还可以用于降低内核间带宽。例如，指纹逻辑 212 可以用于组合来自结构 213 和 215 的值，从而使指纹逻辑 212 的输出中的比特的数量小于这两个值中的比特的总数量。尽管在一些实施例中，希望指纹逻辑 212 对于每一输入组合输出唯一的值，但是在其他实施例中，希望在与减少连接到微校验器 240 的每一输入的比特数量的交换中接受来自微校验器 240 的小于 100% 的准确度。微校验器 240 的小于 100% 的准确度是可以接受的，这是因为微校验器 240 未能检测到可纠正的时钟同步故障将被解释为不可纠正的时钟同步故障，而不是被解释为可以导致系统破坏的正确时钟同步操作。

图 3 示出本发明在用于降低包括图 1 的处理器 100 的时钟同步双模冗余系统中的不可纠正错误率的方法 300 中的实施例，其中结构 111 和 121 是不能改变架构状态的结构，例如预测结构。

在方框 310 中，内核 110 和 120 以时钟同步的方式进行操作。在方框 311 中，结构 111 产生第一值，而结构 121 产生第二值。第一值可以与第二值匹配，也可以与第二值不匹配。在方框 320 中，微校验器 140 将来自结构 111 和 121 的值相比较。在方框 330 中，存储在方框 320 中的比较结果。

在方框 331 中，内核 110 基于结构 111 所产生的值执行第一指令，而内核 120 基于结构 121 所产生的值执行第二指令。第一和第二指令可以是相同的指令，也可以是不同的指令。通过指示条件分支预测、跳转预测、返回地址预测、存储器相关性预测、或者任何其他不能改变架构状态的预测或结果，第一和第二值可以用作用于确定执行什么指令的基础。

方法 300 从方框 331 直接前进到方框 340，或在内核 110 和 120 执行了任何数量的附加指令之后前进到方框 340。

在方框 340，全局校验器 130 将来自内核 110 和 120 的输出相比较。如果输出相匹配，则在方框 310 中内核 110 和 120 的时钟同步操作继续进行，不管存储在方框 330 中的结果如何，都不受任何错误纠正、恢复或通知技术（notification technique）的影响。然而，如果在方框 340 中全局校验器 140 检测到时钟同步故障，则方法 300 前进到方框 350。

如果存储在方框 330 中的结果指示来自结构 111 的值与来自结构 121 的值相匹配，则方法 300 从方框 350 前进到方框 360。在方框 360 中，例如通过报告故障码并中断系统，故障逻辑 150 指示检测到不可纠正的错误。

如果存储在方框 330 中的结果指示在来自结构 111 和 121 的值之间不匹配，则方法 300 从方框 350 前进到方框 370。在方框 370 中，故障逻辑 150 导致内核 110 和 120 的重新同步。可以通过改变内核 110 的架构状态以与内核 120 的架构状态相匹配，或反之亦然，来实现该重新同步。然后，方法 300 返回到方框 310。

图 4 示出本发明在用于降低包括图 1 的处理器 100 的时钟同步双模冗余系统中的不可纠正错误率的方法 400 中的实施例，其中结构 111 和 121 是其内容在系统中的其他处被复制或可以被重新产生的结构，例如高速缓存。

在方框 410 中，内核 110 和 120 以时钟同步的方式进行操作。在方框 411 中，导致对结构 111 中的未修改的高速缓存线进行加载的指令由内核 110 执行，并且导致对结构 121 中的未修改的高速缓存线进行加载的指令产生第二值。方法 400 从方框 411 直接前进到方框 420，或在内核 110 和 120 执行了任何数量的附加指令之后前进到方框 420。

在方框 420 中，微校验器 140 将来自结构 111 的值（例如，在方框 411 中被加载的高速缓存线）与来自结构 121 的值（例如，在方框 411 中被加载的高速缓存线）相比较。在方框 430 中，存储方框 420 中的比较结果。

方法 400 从方框 430 直接前进到方框 440，或在内核 110 和 120 执行了任何数量的附加指令之后前进到方框 440。

在方框 440 中，全局校验器 130 将来自内核 110 和 120 的输出相比较。如果输出相匹配，则在方框 410 中内核 110 和 120 的时钟同步操作继续进行，不管存储在方框 430 中的结果如何，都不受任何错误纠正、恢复或通知技术的影响。然而，如果在方框 440 中全局校验器 140 检测到时钟同步故障，则方法 400 前进到方框 450。

如果存储在方框 430 中的结果指示来自结构 111 的值与来自结构 121 的值相匹配，则方法 400 从方框 450 前进到框 460。在方框 460 中，例如通过报告故障码且中断系统，故障逻辑 150 指示检测到不可纠正的错误。

如果存储在方框 430 中的结果指示在来自结果 111 和 121 的值之间不匹配，则方法 400 从方框 450 前进到方框 470。在方框 470 到 473 中，故障逻辑 150 导致内核 110 和 120 的重新同步。

在方框 470 中，例如通过重新加载在方框 411 中加载的高速缓存线，在系统中的其他处发现，或者不然重新产生来自结构 111 和 121 的值。所重新产生的一个值（例如，如果从该值在系统中被复制的地方获得其单个副本）或多个值（例如，如果从该值在系统中被复制的地方获得每一结构该值的副本）可以被加载到一个寄存器或多个寄存器中，或者其它的一个或多个位置，其提供用来与来自结构 111 和 121 的值进行比较。或者，可以将来自结构 111 和 121 的值移至寄存器、或其它提供用来与所重新产生的一个或多个值进行比较的位置，例如通过重新执行在方框 411 中执行的指令可以获得所述重新产生的一个或多个值。

在方框 471 中，将所重新产生的一个或多个值与来自结构 111 和 121 的值相比较。如果所重新产生的值与来自结构 111 的值相匹配，则在方框 472 中，例如通过改变内核 120 的架构状态以与内核 110 的架构状态相匹配，使内核 120 与内核 110 同步。如果所重新产生的值与来自结构 121 的值相匹配，则在方框 473 中，例如通过改变内核 110 的架构状态以与内核 120 的架构状态相匹配，使内核 110 与内核 120 相匹配。方法 400 从方框 472 和 473 返回到方框 410。

图 5 示出本发明在用于降低包括图 2 的处理器 200 的时钟同步双

模冗余系统中的不可纠正错误率的方法 500 中的实施例。

在方框 510 中，内核 210 和 220 以时钟同步的方式进行操作。在方框 511 中，结构 213 产生值，并且结构 223 产生值。来自结构 213 的值可以与来自结构 223 的值相匹配，也可以与来自结构 223 的值不匹配。在方框 512 中，结构 215 产生值，并且结构 225 产生值。来自结构 215 的值可以与来自结构 225 的值相匹配，也可以与来自结构 225 的值不匹配。

在方框 513 中，结构 211 基于来自结构 213 和 215 的值产生指纹值，并且结构 221 基于来自结构 223 和 225 的值产生指纹值。可以根据任何公知的用于组合值的技术（例如，使用循环冗余校验器产生校验和）来产生指纹值。

在方框 520 中，微校验器 240 将来自结构 211 和 221 的指纹值相比较。在方框 530 中，存储方框 520 中的比较结果。

在方框 540 中，全局校验器 230 将来自内核 210 和 220 的输出相比较。如果输出相匹配，在方框 510 中内核 210 和 220 的时钟同步操作继续进行，不管存储在方框 530 中的结果如何，都不受任何错误纠正、恢复、或通知技术的影响。然而，如果在方框 540 中全局校验器 240 检测到时钟同步故障，则方法 500 前进到方框 550。

如果存储在方框 530 中的结果指示来自结构 211 的指纹值与来自结构 221 的指纹值相匹配，则方法 500 从方框 550 前进到方框 560。在方框 560 中，例如通过报告故障码且中断系统，故障逻辑 250 指示检测到不可纠正的错误。

如果存储在方框 530 中的结果指示在来自结构 211 和 221 的值之间不匹配，则方法 500 从方框 550 前进到方框 570。在方框 570 中，故障逻辑 250 导致内核 210 和 220 的重新同步。可以通过改变内核 210 的架构状态以与内核 220 的架构状态相匹配，或反之亦然，来实现该重新同步。然后，方法 500 返回到方框 510。

在本发明的范围内，可以以不同的顺序执行图 3、4 和 5 所示的方法，可以省略所示的步骤，加入额外的步骤，或者将记录、组合、省略或额外的步骤组合。例如，如果微校验器的输出保持为静态直到

分别执行方框 350、450 或 550（检查微校验器的比较结果）为止，则可以省略方框 330、430 或 530（存储微校验器的比较结果）。

其中可以省略方框 330（存储微校验器的比较结果）的方法的其他实例是其中不需要保留微校验器输出的本发明的实施例。在一个这样的实施例中，方法可以从方框 320 的微校验器比较前进到基于微校验器比较的方框 350 的判定（或者，可以将方框 320 和 350 合并）。在该实施例中，如果微校验器检测到不匹配（在 320 或 350 中），则处理器现有的分支误预测恢复机制可以用于除掉推测状态，由此在方框 370 中将内核同步到非推测状态。如果微校验器没有检测到不匹配，则该实施例的方法可以前进到方框 331，以基于预测执行指令，然后前进到方框 340，以便全局校验器为时钟同步故障进行检测，然后，如果检测到时钟同步故障，则前进到方框 360，以指示不可恢复的错误。

图 6 示出本发明在时钟同步双模冗余系统 600 中的实施例。系统 600 包括多内核处理器 610 和系统存储器 620。处理器 610 可以是如以上对于图 1 和 2 所述的任何处理器。系统存储器 620 可以是任何类型的存储器，例如基于半导体的静态或动态随机存取存储器、基于半导体的闪存或只读存储器、或者磁盘或光盘存储器。可以以任何布置，利用任何组合总线或者直接或点对点连接，并且通过任何其他部件使处理器 610 和系统存储器 620 彼此耦合。系统 600 也可以包括任何总线（例如外围总线）或部件（例如输入/输出设备），在图 6 中未示出。

在系统 600 中，系统存储器 620 可以用于存储可以加载到诸如上述结构 111、121、213、215、223 和 225 的结构的价值。因此，系统存储器 620 可以是根据本发明的方法实施例的复制或重新产生的值的来源，例如，如图 4 的方框 470 所示。

可以在从生成到模拟到制造的各个阶段设计处理器 100、处理器 200 或者根据本发明的实施例设计的任何其他部件或部件的一部分。表示设计的数据可以以多种方式表示设计。首先，如在模拟中所使用的那样，可以使用硬件描述语言或另一功能描述语言表示硬件。另外或可选择地，在设计过程的某些阶段可以产生具有逻辑和/或晶体管

门电路的电路级模型。此外，在某些阶段，大部分设计达到可以利用表示各个器件的物理布局的数据对它们进行建模的级别。在使用常规的半导体制造技术的情况下，表示器件布局模型的数据可以是指定在用于掩模的不同掩模层上各种特征是存在或还是不存在的的数据，所述掩模用于制造集成电路。

在设计的任何表示中，可以将数据存储在任何形式的机器可读介质中。调制的或者以其他方式产生的用以传送这种信息的光波或电波、存储器、或者磁或光存储介质，例如盘片，可以是机器可读介质。这些介质中的任何一种可以“承载”或“指示”设计、或在本发明的实施例中使用的其他信息，例如在错误恢复例程中的指令。当传送指示或承载信息的电载波时，到执行电信号的复制、缓冲、或重发的程度，进行新的复制。因此，通信供应商或网络供应商的行为可以是对体现本发明的技术的对象（例如，载波）进行复制的行为。

因此，公开了用于降低时钟同步双模冗余系统中的不可纠正错误率的装置和方法。尽管对特定实施例进行了说明并且在附图中示出了所述特定实施例，但是应该理解的是，这些实施例仅仅是说明性的而不是限制本发明的宽广范围，并且本发明不限于所示和所述的具体结构和布置，因为对于本领域技术人员而言，在研究本公开文本时可以想到多种其他修改。在诸如发展迅速且不容易预见更大的进步的技术领域中，由于实现技术进步而得到帮助，使得可以在不脱离本公开的原则或所附权利要求的范围的情况下，轻易地修改所公开的实施例的布置和细节。



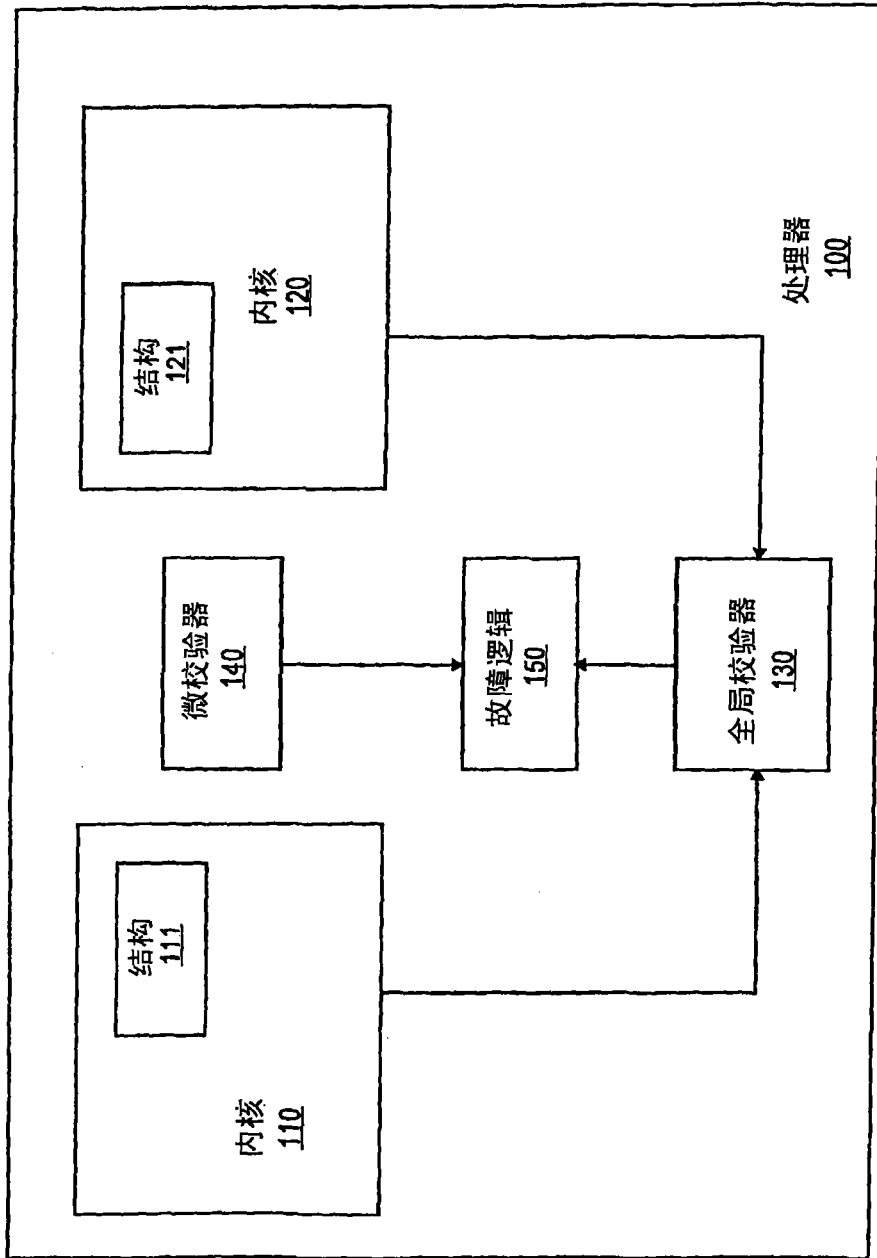


图1

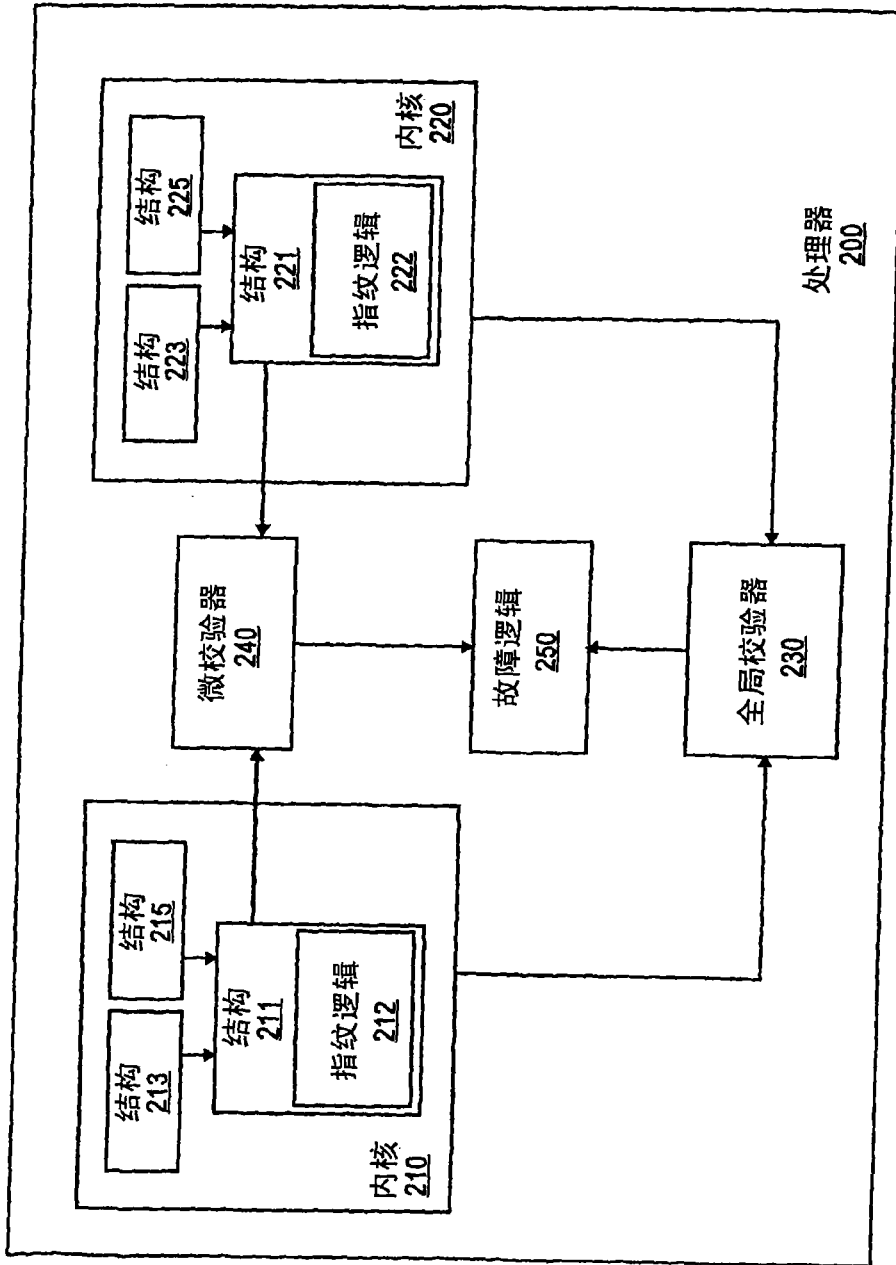


图2

方法 300

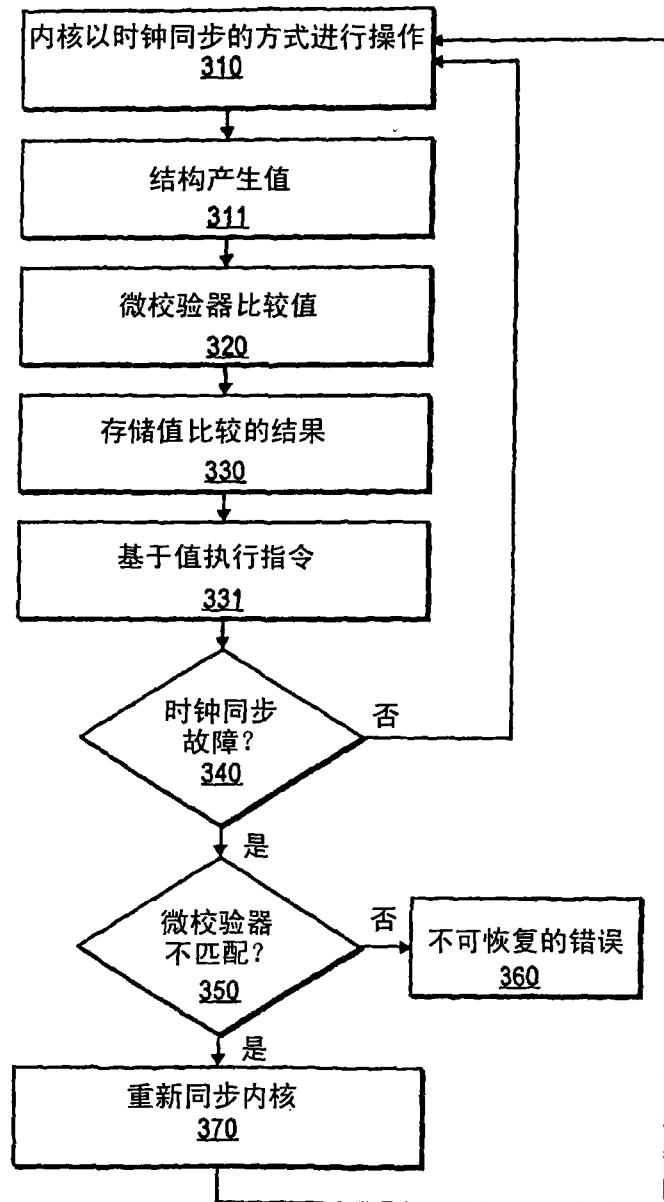


图3

方法 400

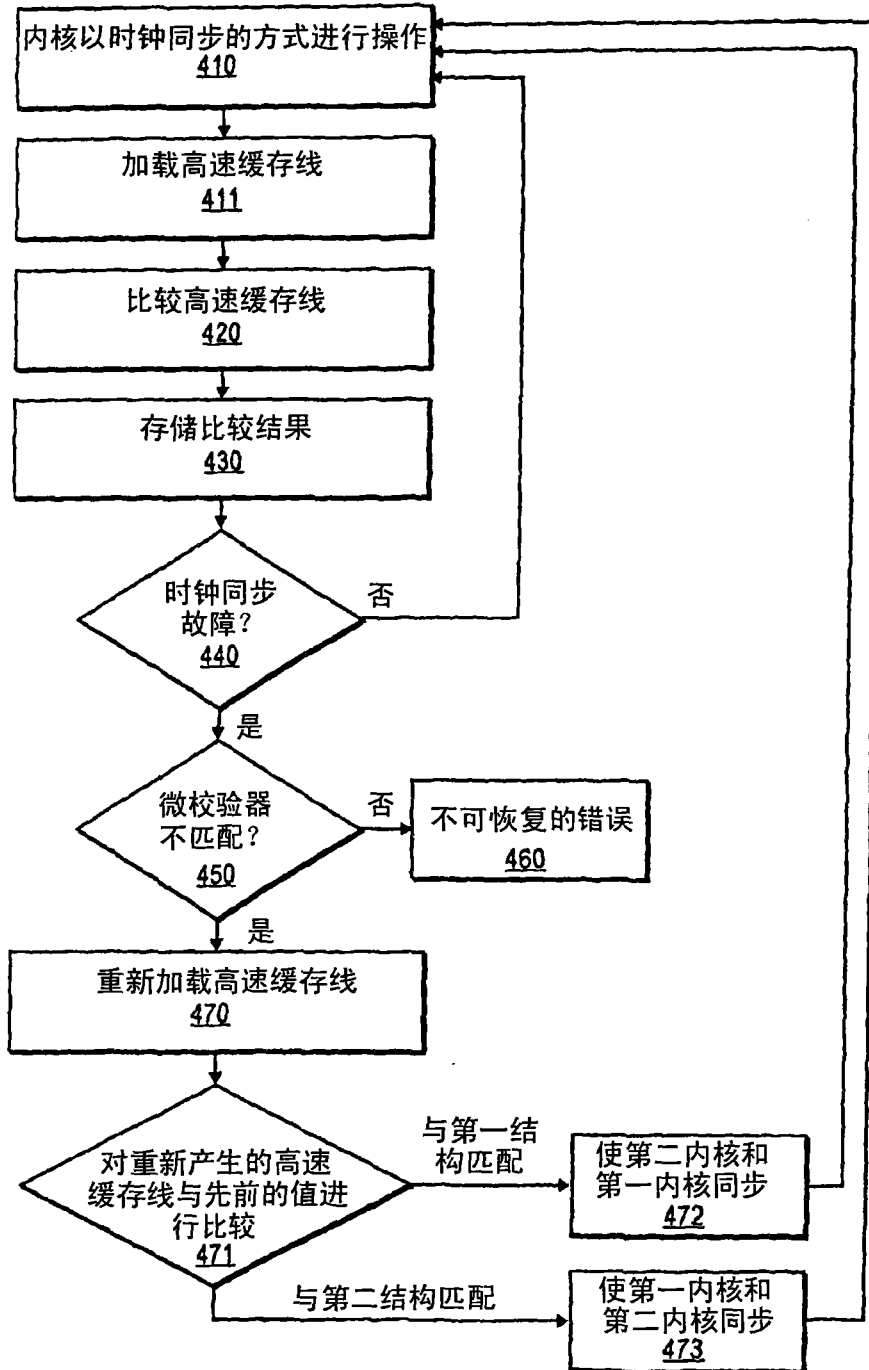


图 4

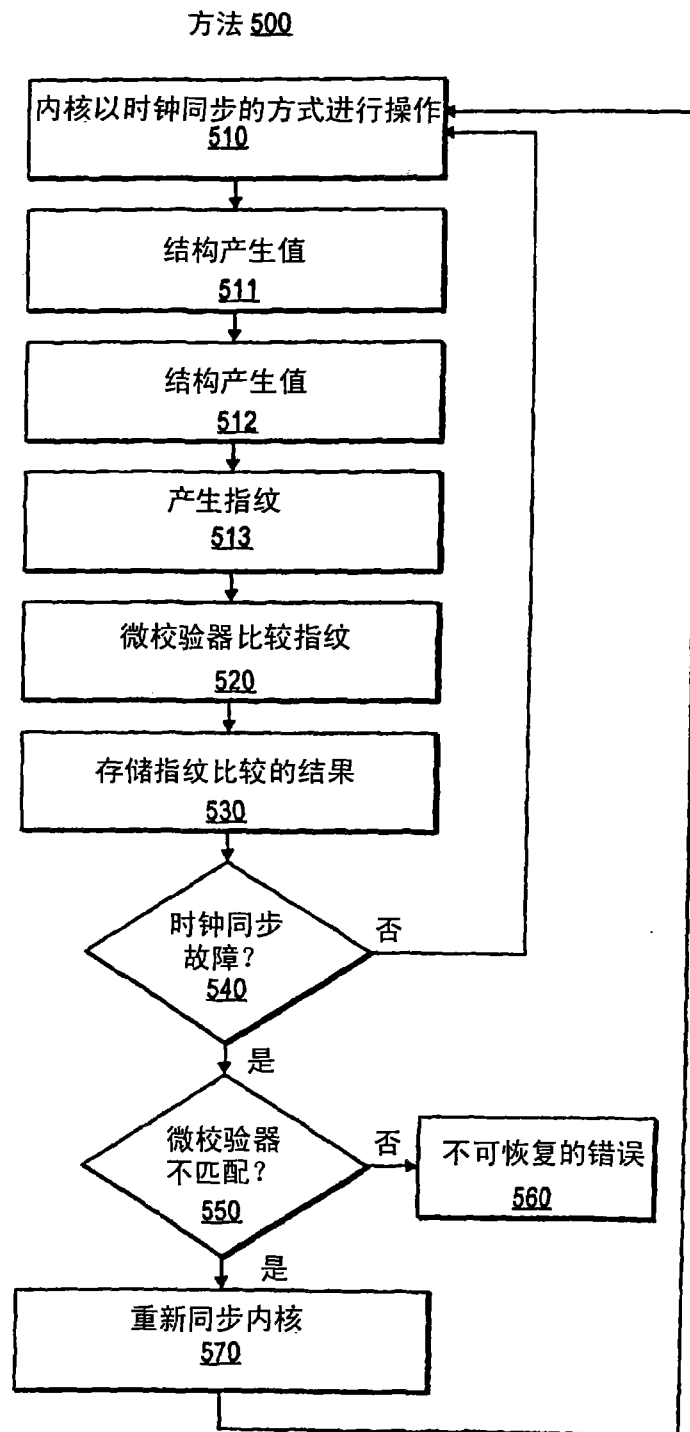


图5

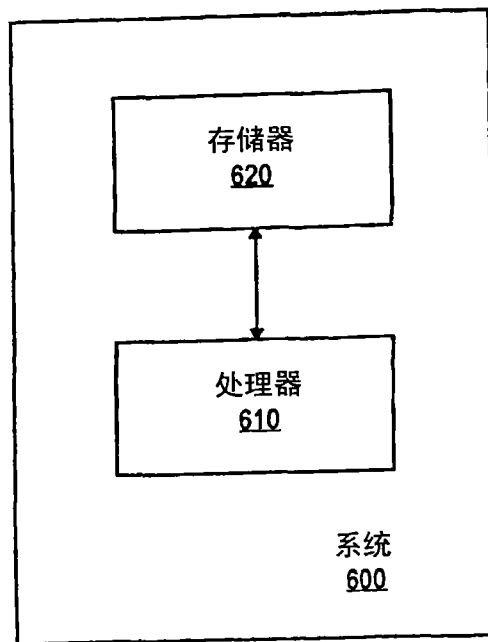


图6