



[12] 发明专利申请公开说明书

[21] 申请号 03805711.5

[43] 公开日 2005 年 7 月 13 日

[11] 公开号 CN 1639691A

[22] 申请日 2003.8.7 [21] 申请号 03805711.5

[30] 优先权

[32] 2002.9.12 [33] EP [31] 02020602.5

[32] 2002.12.12 [33] EP [31] 02027848.7

[86] 国际申请 PCT/EP2003/008794 2003.8.7

[87] 国际公布 WO2004/034261 德 2004.4.22

[85] 进入国家阶段日期 2004.9.10

[71] 申请人 西门子公司

地址 德国慕尼黑

[72] 发明人 A·韦伯 D·施纳贝尔

P·佩莱斯卡

[74] 专利代理机构 中国专利代理(香港)有限公司

代理人 吴立明 张志醒

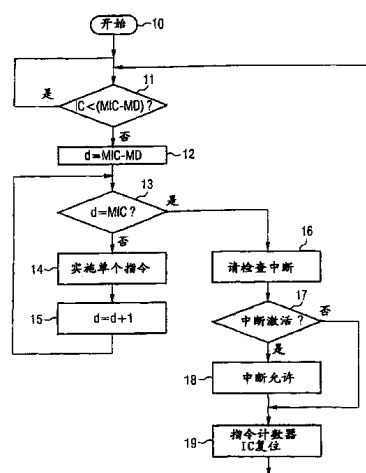
权利要求书 3 页 说明书 10 页 附图 2 页

[54] 发明名称 尤其是容许故障系统的处理器的用于事件同步的方法

[57] 摘要

对于冗余的系统一再规定了等同构造的处理器插件板, 这些处理器插件板以锁步运行来工作。实现锁步系统的基本前提是所有包含在插件板中的部件的, 即 CPU、芯片组、主存储器等等的确定性特性。确定性的特性在此意味着, 如果部件在等同的时刻获得等同的激励, 这些部件则在无故障的情况下在等同的时刻提供等同的结果。确定性的特性此外还以采用时钟同步的接口为前提。在系统中异步的接口在许多情况下促成某种时间的不清晰度, 由此不能维持系统的时钟同步的总特性。为了仍然能够执行锁步运行, 本发明规定了一种用于同步外部事件的方法, 这些外部事件输送给一个处理器 (CPU) 和影响该处理器 (CPU), 据此将这些外部事件暂存, 并在指令执行中的等同的位置上给该处理器演示, 其中, 避免了由现代处理器并行执行指令

的能力所产生的问题, 其方式是在达到指令执行中的所希望的位置之前禁止处理器的并行执行, 并随后以单步模式准确地达到该位置。



1. 用于同步外部事件的一种方法，这些外部事件被输送给一个模块（CPU）和影响该模块（CPU），其中，规定所述的模块（CPU）用于并行执行第一数量的指令，

5 - 据此将所述的外部事件暂存，其中，在所述模块的单独工作模式中为了处理通过所述模块的至少一个执行单元（EU）来调用存储的外部事件，以及

 - 其中，所述的模块在执行可预先给定的第二数量（MIC）指令之后进入该工作模式中，其方式是

10 - 计数器（IC）求出从最后离开所述的单独工作模式时开始的由所述执行单元所执行指令的数量，

 - 如果所述的计数器（IC）大于或等于来自所述第二数量指令和第三数量（MD）指令的差值，则将所述的模块移到单个指令执行模式中，该第三数量（MD）指令从所述第一数量指令中求出，

15 - 所述的模块保留在所述的单个指令执行模式中，直至所述的计数器（IC）达到了所述的第二数量（MIC）指令时为止，所述的模块随即转换到所述的单独工作模式中，并在离开所述的单独工作模式时重新初始化所述的计数器（IC）。

2. 按权利要求 1 的方法，

20 其特征在于，

 将指令的所述第三数量选择为等于或大于最大并行执行的指令的所述第一数量。

3. 按权利要求 1 或 2 之一的方法，

 其特征在于，

25 在具有至少两个模块的冗余系统中，为所述的模块（CPU）规定了等同序列的指令，并通过所述的模块在所述的单独工作模式中调用等同的外部事件。

4. 按权利要求 3 的方法，

 其特征在于，

30 通过控制装置让较快速的模块（CPU）保留在所述的单独工作模式中，直至较缓慢的模块达到所述单独工作模式的终点时为止。

5. 按权利要求 1 至 4 之一的方法，其特征在于，在由监控软件模

块监控所述的计数器（IC）时，由所述监控软件模块促使执行的指令的数量单独来采集，并从所述的计数器（IC）中扣除。

6. 处理器模块（CPU），该处理器模块（CPU）至少具有以下的内容：

- 5 - 至少一个执行单元（EU），
- 至少一个计数器元件（IC），用于从最后转换到单独工作模式中时开始对由所述执行单元所执行的指令进行计数，
- 至少一个寄存器元件（MIR），该寄存器元件（MIR）的内容（MIC）通过指令可预先给定或固定地预先给定，
- 10 - 至少一个比较器元件（K）和至少一个控制元件（S），用于通过所述的计数器元件（IC）对达到了小于所述寄存器元件（MIR）值的可预先给定的值作出响应地，将所述的执行单元（EU）转接到单个指令执行模式中，以及用于对所述的计数器元件（IC）与所述寄存器元件（MIR）的一致性作出响应地，将所述的执行单元转接到所述的单独
- 15 工作模式中，其中，通过所述的处理器模块（CPU）调用在所述单独工作模式中暂存的、输送给该处理器模块（CPU）的外部事件，这些外部事件影响该处理器模块（CPU）。

7. 由至少两个处理器模块（CPU0，CPU1）组成的系统，这些处理器模块分别至少具有以下的内容：

- 20 - 至少一个执行单元（EU），
- 至少一个计数器元件（IC），用于从最后转换到所述单独工作模式中时开始对由所述执行单元所执行的指令进行计数，
- 至少一个寄存器元件（MIR），该寄存器元件（MIR）的内容通过指令可预先给定或固定地预先给定，
- 25 - 至少一个比较器元件（K）和至少一个控制元件（S），用于通过所述的计数器元件（IC）对达到了小于所述寄存器元件（MIR）值的可预先给定的值作出响应地，将所述的执行单元（EU）转接到单个指令执行模式中，以及用于对所述的计数器元件（IC）与所述寄存器元件（MIR）的一致性作出响应地，将所述的执行单元（EU）转接到所述的
- 30 单独工作模式中，其中，通过这些处理器模块调用在所述单独工作模式中暂存的、输送给这些处理器模块的外部事件，这些外部事件影响这些处理器模块。

8. 按权利要求 7 的系统, 该系统附加地具有在所述处理器模块 (CPU) 中的至少两个之间的通信连接(L0, L1), 这些处理器模块(CPU) 执行等价的指令序列, 其中, 规定了所述的通信连接用于传输所述单独工作模式的同步信息。

5

尤其是容许故障系统的处理器的用于事件同步的方法

在电信系统中，在数据中心和另外的高度可用的系统中，在许多
5 情况下采用直至数百个所谓的处理器插件板，以便规定必要的计算功能。这种处理器插件板典型地由处理器或 CPU（中央处理器）、芯片组、主存储器和外围模块组成。

典型处理器插件板每年出现硬件故障的概率位于个位数的百分率
范围内。由于大量汇总成系统的处理器插件板而产生了涉及年度的、
10 任意硬件部件的很高的失效概率，其中，如果未采取适当的预防措施，这种个别失效则可能引起整个系统的失效。

尤其在电信系统上，以增长的程度也在数据中心的提出了对高的
系统可用性的要求。例如以百分数来表达该系统可用性，或给定每年
最高允许的失效时间。典型的要求例如是 >99.999%的可用性，或一年
15 之中最高数分钟的不可用性。由于在硬件故障的情况下，处理器插件板的更换和业务的重建通常要求位于数十分钟至数小时范围内的时间，对于硬件故障的情况必须在系统层面上采取相应的预防措施，以便可以满足对系统可用性的要求。

用于遵守对系统可用性的这种高要求的已知的解决方案规定了冗
20 余的系统部件。可以将已知的方法划分成两个主要的组：基于软件的方法和基于硬件的方法。

在基于软件的方法中典型地采用中间件。但是基于软件的解决方案证明是灵活性小的，因为在这种系统中仅仅可以采用为该特别的冗余格式开发的那些（应用）软件。这显著地限制了可采用（应用）软件的范围。除此之外，软件冗余原理的应用软件的开发在实践中是极其费事的，其中，该开发附加地产生了复杂的测试方法的后果。
25

基于硬件方法的基本原理基于，将冗余性封装在硬件层面上，使得这对于软件是透明的。由硬件本身所管理冗余的主要优点在于，不由冗余原理来影响应用软件，并因而在大多的情况下每个任意的软件
30 可以得到采用。

容许硬件故障系统的在实践中可以经常遇到的原理是所谓的锁步

(Lockstep) 原理, 该系统的冗余对于软件是透明的。锁步意味着, 同样类型时钟同步地运行等同构造的硬件, 例如两个插件板。通过硬件机构来确保, 冗余的硬件在给定的时刻获知等同的输入激励, 并因此必须导致等同的结果。将冗余部件的结果进行比较, 在有偏差的情况下查明故障, 并引入合适的措施 (向操作人员报警, 部分的或全部的安全关断, 系统重新启动)。

实现锁步系统的基本前提是, 所有包含在插件板中的部件的, 也就是 CPU、芯片组、主存储器等等的节拍方式确定性的特性。节拍方式确定性的特性在此意味着, 在无故障的情况下, 如果部件在等同的节拍时刻获得等同的激励, 这些部件则在等同的节拍时刻提供等同的结果。节拍方式确定性的特性此外还以采用时钟同步的接口为前提。在系统中, 异步的接口在许多情况下促成某种时间的不清晰度, 由此不能够维持系统的时钟同步的总特性。

但是异步的接口恰恰对于芯片组和 CPU 提供了提高效能方面的技术优点, 由此使得按照锁步法的时钟同步的工作方式成为不可能。现代的 CPU 此外还越来越多地采用使得时钟同步的工作方式不可能的机构。例如这是向外部不可见的内部的修正措施, 例如在访问超高速缓存存储器时的内部可修正故障的修正, 这些修正措施可能导致指令执行的稍微滞后, 或推测性地 (spekulative) 执行指令。一个其它的实例是未来越来越多地实现 CPU 内部的无节拍的执行单元, 这些执行单元实现了速度和功率损失方面的显著优点, 但是阻碍了 CPU 的时钟同步或确定性的工作。

欧洲专利申请 02020602 公布了一种用于同步外部事件的方法, 这些事件被输送给 CPU 和影响这些 CPU, 据此将外部的事件暂存, 其中, 通过执行单元在用于处理的 CPU 的单独工作模式中调用所存储的外部事件, 以及其中, CPU 对满足由指令可预先给定或固定地预先给定的条件作出响应进入该工作模式。该方法也称为“仿真锁步运行”。

EP 02020602 有利地规定了, 如果由 CPU 的比较器元件来求出计数器与最大指令寄存器 (MIR) 的一致性, 则实施转换到单独的工作模式中, 其中, 由指令可预先给定 MIR 的内容, 而计数器含有从最后转换到单独的工作模式时开始的由执行单元所执行指令的数量。

但是现代的 CPU 不能如此来停止, 以至于它们在准确数量的指令

之后停止。其原因在于，可以并行执行多个在一个共同的时刻结束的指令。因此例如在一个节拍中可以在所有冗余的 CPU 上执行 99 个指令，在下一个节拍中例如由于不同的实施方案于是在一个 CPU 上是 100 个指令，但是在另一个 CPU 上是 101 个指令。因此不能在指令执行中在等同位置上演示外部的事件，例如中断。

本发明的任务是提出一种方法，用该方法即使当在执行一个和同一个指令之后不能确保冗余 CPU 的停止时，则也在冗余 CPU 的指令执行中在相同位置上演示外部的事件。

通过一种按权利要求 1 特征的用于同步外部事件的方法、通过一种按权利要求 6 特征的处理器、和通过一种按权利要求 7 特征的系统来解决该任务。在从属权利要求中说明了有利的改进方案。

按本发明规定了用于同步外部事件的一种方法，这些外部事件被输送给一个模块 CPU 和影响该模块 CPU，其中，规定模块 CPU 用于并行执行第一数量的指令，

— 据此将外部事件暂存，其中，在模块的单独工作中为了处理模式通过模块的至少一个执行单元 BU 来调用存储的外部事件，以及

— 其中，模块在执行可预先给定的第二数量 MIC 指令之后进入该工作模式中，其方式是

— 计数器 IC 求出从最后离开单独工作模式时开始的由执行单元所执行指令的数量，

— 如果计数器 IC 大于或等于来自第二数量指令和第三数量 MD 指令的差值，则将模块移到单个指令执行模式中，该第三数量 MD 指令从第一数量指令中求出，

— 模块保留在单个指令执行模式中，直至计数器 IC 达到了第二数量 MIC 指令时为止，模块随即转换到单独工作模式中，并在离开单独工作模式时重新初始化计数器 IC。

所述的第三数量指令在此以最大数量的并行执行的指令为依据，并用来补偿在具有并行执行指令能力的 CPU 的停止时的不清晰度。优选将第三数量选择为等于或大于最大并行执行指令的第一数量。

在具有至少两个模块 CPU 的冗余系统中，为模块 CPU 规定了等同序列的指令，并通过模块在单独的工作模式中调用等同的外部事件。

通过控制装置让较迅速的模块 CPU 保留在单独的工作模式中，直至较缓慢的模块达到单独工作模式的终点时为止。

- 通过软件、通过微码、或通过专用的硬件可以实现本发明的方法。在由监控软件模块监控计数器 IC 时，由监控软件模块促使执行指令的数量单独来采集，并从计数器 IC 中扣除。

本发明还规定了处理器模块 CPU，该处理器模块 CPU 至少具有以下的内容：

- 至少一个执行单元 EU，
- 至少一个计数器元件 IC，用于从最后转换到单独工作模式中时开始对由执行单元所执行的指令进行计数，
- 至少一个寄存器元件 MIR，该寄存器元件 MIR 的内容 MIC 通过指令可预先给定或固定地预先给定，
- 至少一个比较器元件 K 和至少一个控制元件 S，用于通过计数器元件 IC 对达到了小于寄存器元件 MIR 值的可预先给定的值作出响应地，将执行单元 EU 转接到单个指令执行模式中，以及用于对计数器元件 IC 与寄存器元件 MIR 的一致性作出响应地，将执行单元转接到单独工作模式中，其中，通过处理器模块 CPU 调用在单独工作模式中暂存的、输送给该处理器模块 CPU 的外部事件，这些外部事件影响该处理器模块 CPU。

- 可以将这些处理器中的多个有利地汇总成一个系统，其中，该系统附加地具有在处理器模块 CPU 中的至少两个之间的通信连接 L0，L1，这些处理器模块 CPU 执行等同的指令序列，其中，规定了通信连接用于传输单独工作模式的同步信息。

- 本发明的主要优点在于，在容许硬件故障的平台上实现了采用任意新的或已有的软件，其中，在该平台上可以采用支持本发明的 CPU，而不存在对 CPU 的时钟同步的确定性工作方式的要求，以及其中，可以采用异步的高速接口或链路。此时本发明考虑了以下的情况，具有并行执行指令能力的现代的 CPU 不是在任何情况下能够在准确数量的指令之后停止 (anhaltend)。

- 其它的优点是：

- 不必锁相耦合地运行彼此冗余的插件板和 CPU。
- CPU 不必是完全相同的，它们仅须在相同数量的已执行机器指令

之后停止，并转换工作模式。

- 可以用不同的时钟频率来运行 CPU。

- 在涉及指令的推测性执行方面，CPU 可以不同地表现，因为仅评价成套 (komplettierten) 的指令。

- 5 - 例如基于在使数据失真的 α 粒子出现之后修正的、等同 CPU 的 CPU 内部不同的执行时间仅仅导致，在稍微不同的时刻达到同步模式。以下联系三个附图来详述本发明的实施例。

图 1 展示了本发明方法的流程图。

图 2 示意地展示了本发明的处理器模块。

- 10 图 3 示意地展示了本发明的由根据图 2 的两个处理器模块组成的系统。

图 1 中以图形示出了作为流程图的本发明方法。在流程开始之前必须确定或初始化以下的值：

- 计数器 IC (指令计数器)，该计数器 IC 含有由 CPU 执行的指令的、或机器指令的数量。

- 指令的数量 MIC (最大指令计数器)，CPU 应在该指令的数量 MIC 之后转换到特别的工作模式中，用于处理外部的事件。

- 指令的数量 MD (最大偏差)，该指令的数量 MD 考虑了 CPU 停止的、由于指令执行的并行性而出现的最大不清晰度。

- 20 流程随着指令计数器 IC 的当前值与来自 MIC 和 MD 的差值的比较 (方框 11) 而开始。如果指令计数器的值小于该差值，则以正常的工作模式继续指令执行；可以并行执行指令。

如果指令计数器的值达到或超过来自 MIC 和 MD 的差值，则用来自 MIC 和 MD 的差值装载寄存器 d (方框 12)，并且流程进入环路，在该环路的开始存在着寄存器 d 是否已达到值 MIC 的查询 (方框 13)。在该环路中以单步模式来进行指令执行。

只要值 d 未达到值 MIC，则在每个环路执行中执行单个的指令 (方框 14)，并在重新检查环路条件之前 (方框 13)，使值 d 增量 (方框 15)。通过该行动确保了，尽管在并行指令执行的正常工作中，仍然准确地在 MIC 指令之后实现进入单独的工作状态。

30 如果值 d 达到了值 MIC (方框 13)，流程则进入单独的工作模式。单独的工作模式首先检验，在执行 MIC 指令期间中断请求是否到达，

以及为了由所有冗余的 CPU 同时来执行,是否已暂存了该中断请求(方框 16/17)。如果中断请求已到达,则处理这些中断请求(方框 18),其中,在程序执行中的等同位置上由所有冗余的 CPU 进行该处理,并且所有的寄存器、存储器内容、...是等同的。如果未出现中断请求,则跳越该步骤。

终止单独的工作模式,并在指令计数器 IC 复位之后(方框 19)重新接受(aufnehmen)具有并行处理指令的正常工作模式。于是可以处理中断请求。不以单独的工作模式,而是以正常的模式实现中断程序的执行。以特别的工作模式仅仅进行读入中断矢量,在此之后又离开该特别的模式。是否处理中断于是例如取决于,在该时刻是否恰好允许中断。如果恰好在处理中断,和/或清除“中断标志”,则不允许中断。

借助示出的流程可以将本发明方法直接实施为指令序列,即实施为软件。此时软件负责,在多个处理器的指令执行的等同位置上演示中断,其方式是在 CPU 中如此来编程指令计数器,使得在扣除“停止不清晰度”MD 的所希望数量 MIC 的要执行指令之后,该指令计数器引起异常(Exception)、譬如调试异常、或高度优先的不可闭锁的中断、譬如不可掩蔽的中断 NMI。例如在 MD=3 个指令的不清晰度和所希望数目 MIC=1000 个指令时,用 $1000-3+1=998$ 来编程计数器 IC。取决于指令的内部会聚,即在 IC=998、或 IC=999、或 IC=1000 个指令之后使 CPU 停止。随即实施的软件读出指令计数器,以便查明,处理器在哪个位置上实际已停止。此时该软件具有以下的特性,相应地修正自己指令的执行。如果软件查明,CPU 例如在 999 个指令之后已停止,则由异常软件控制地,通过单步工作来补充执行所希望的第 1000 个指令。这在所有冗余的 CPU 上实现,使得在码中的等同位置上已使所有的 CPU 停止。

在此位置上必须给这个/这些 CPU 演示可能存在的中断请求。对此例如存在着以下的可能性:

- CPU 可以读出中断控制器的寄存器,该寄存器随即释放掩蔽的中断信号。CPU 借助该中断信号识别中断请求,并向中断控制器发送中断确认循环。中断控制器随即提供中断矢量,并重新掩蔽中断信号。

- 替代地软件可以读出寄存器,该寄存器的值已经提供关于中断

类型的信息，即中断矢量。如果在指令执行中的该时刻允许中断，软件随后则自己触发相应的中断（通过软件）。

此外也可以以微码指令的形式来实现流程。现代的 CPU 在许多情况下具有借助微码来控制指令执行的广泛的可能性。例如常常将这些可能性用来排除或回避设计错误。

为了本发明方法的目的如此来改变微码，使得在扣除“停止不清晰度”MD 的所希望数目的要执行指令 MIC 之后，CPU 中断正常的指令执行，并转移到微码中。微码读出已执行指令的数量 IC，并如此来控制通过单步的执行，使得在所希望的位置 MIC 上停止指令执行。

在此位置上又必须给这个/这些 CPU 演示可能存在的中断请求。对于此同样存在着多种可能性：

- 通过微码来释放用微码掩蔽的中断信号，并且如果存在着中断，则将 CPU 转移到相应的中断程序中。随后又通过微码来掩蔽中断。

- 替代地可以促使 CPU，生成中断确认循环和读取中断矢量。于是通过微码如此来给 CPU 演示该中断矢量，使得该 CPU 在离开了单独的模式之后转移到相应的中断程序中。

此外可以在译码软件中进行实施。数个 CPU 拥有简单的、但是很迅速的、大多超标量（superskalar）的 RISC 或 VLIW 处理器核心。通过译码软件将本身的指令组，例如 IA-32 转换成简单的码，并由 RISC/VLIW 处理器来实行。在此情况下译码软件实施本方法的任务，类似于以微码的实施。像在微码实施中那样进行中断请求的演示。

本发明方法的最有效的实施是硬件实施，示出在图 2 中。在此由处理器内部的硬件单元 S 在扣除不清晰度的所希望的位置上来停止并行的指令实行、求出指令计数器读数 IC、以及由处理器内部的硬件单元 S 借助单步 ES 将执行单元 EU 引导到码中的所希望的位置上。该方法的主要优点是显著降低了的负面的性能影响。

图 2 以示意图展示了本发明的处理器模块 CPU。在此，仅示出了对于本发明是重要的组成部分。CPU 包括一个或多个执行单元 EU、至少一个比较器 K、至少一个用于对由执行单元 EU 所执行的指令进行计数的计数器 IC、一个控制装置 S、和至少一个寄存元件 MIR，其内容由指令可预先给定或固定地预先给定。此外还示意地示出了往/返中断寄存器的通信连接（图 3）。

不将影响程序流程的外部事件直接输送给 CPU, 而是首先通过合适构成的硬件来暂存。在图 2 中示出的 CPU 中可以进行本方法的转换, 其方式是用来自值 MIC 和值 MD 的差值来装载寄存器 MIR。比较器 K 将所执行运算的数目与该寄存器值进行比较, 并将该比较的结果给控制单元 S 发送信号。替代地, 比较器也可以仅传送一个事件到控制装置上, 如果 IC 的值已达到了 MIR 的值, 则产生该事件。如果出现了该事件, 或已对两个寄存器的相等性发送信号, 控制装置 S 则再次查询指令计数器, 以便读取实际执行指令的数目。由于在 MIR 中通过用值 MIC-MD 的装载已经考虑了不清晰度, 因此可以由控制装置将单步模式通过线路 ES 给执行单元发送信号, 该执行单元单个地促使执行指令, 直至指令计数器的值达到了预先给定值 MIC 时为止。控制装置 S 为此拥有以下的可能性, 只要指令计数器非自动地对以单步执行的指令进行计数, 就使指令计数器 IC 增量。

每个冗余 CPU 的控制装置 S 生成输送给中断模块的中断释放信号 IF。随即通过中断线路 INT 给所有冗余的 CPU 报告必要时暂存的中断请求。

替代与此地, 控制装置 S 生成自己 CPU 的中断, 如果在指令执行中的该时刻允许中断, 执行单元随即向中断模块发送中断确认循环。

在一个其它的替代方案中, 由控制装置 S 生成一个中断释放信号 IF, 将该中断释放信号 IF 按意义与中断信号 INT 进行 UND 逻辑连接, 即如果存在着逆转的信号, 或如果在多个线路上演示着 (präsentieren) 中断信号, 则应该相应地选择电路逻辑。同样可以将中断释放信号向 CPU 外部地, 例如传送到中断寄存器上。有可能因此释放了施加在中断线路 INT 上的中断, 并可以进行通常的中断处理, 例如读取中断矢量, 执行中断程序等等。

在中断处理之前对于排除单步模式和单独工作模式, 以及对于继续以正常模式执行指令, 向执行单元发送信号, 并通过信号 CL 使指令计数器复位。可以将控制装置直接作为硬件或以微码的形式来实现。

最后在图 3 中, 根据联系图 2 的以上说明来展示两个 CPU 的联接。在此没有来自图 2 中的细节地, 示出了第一处理器 CPU0 和第二处理器 CPU1。处理器各自通过总线 A/D 与所分配的中断模块交换地址和数据, 这些中断模块还包括中断寄存器 IR0, IR1。中断模块例如从输入

/输出模块 I/O 接收中断 INT1..INTn, 存储相应的特征数据, 和转送中断 INT 到处理器上。

根据本发明通过处理器仅在指令执行的某些位置上认可中断。这在与图 2 的关联中已经详述。

5 此外可以采用在此关联中阐述的中断释放信号, 以便给分配给每个处理器的中断模块发送信号, 可以开始中断处理。通过通信连接 L0, L1 相连接的中断模块可以交换这些信息, 并在它这方面当所有的处理器生成中断释放信号时, 才可以例如通过将中断矢量传送到处理器上来释放中断处理。

10 在一个替代方案中可以证明有利的是, CPU 不在指令执行的预先给定的位置 MIC 上, 而是在具有可并行执行指令的不清晰度的位置上停止, 并且于是通过单步将位于后面的处理器引导到指令执行的以下的位置上, 在该位置上已经停止了在指令执行方面进展得最远的那个处理器。为此需要处理器之间的通信。例如这可以如下来实现, 使得每个处理器将它本身已停止的位置写入硬件寄存器中, 并随后将其读回。寄存器等候到所有的处理器已写入了它们的值时为止, 并返回提供最高的值作为读取数据 (Lesedatum)。如果有必要, 所有的处理器则通过单步使它们的指令执行水平相等。于是如上所述给处理器演示中断请求。

20 具有 SMT (同时乘法处理) 能力的 CPU 必须拥有每个虚拟 CPU 或每种处理用的单独的控制装置。

CPU 还具有比较器 K, 该比较器 K 将已执行指令的数量, 即计数器 IC, 与寄存器 MIR 进行比较, 并在相等时生成例如中断请求, 该比较器 K 在由寄存器 MIR 预先给定指令的数目之后中断指令执行, 并将 CPU 25 连接到另外的工作模式中。在该工作模式中实行例如合适的微码, 或转移到中断业务程序中, 或通过硬件信号指明达到了该同步点。于是在该工作模式中如下来给冗余的 CPU 演示外部事件, 使得在离开该工作模式之后所有的 CPU 可以相同地评价这些事件, 并因此可以顺序地执行相同的指令。

30 例如在达到了由寄存器 MIR 预先给定的机器指令数量之后, CPU 转移到中断业务程序中, 在该中断业务程序中如此来查询中断信号的状态, 由所述硬件使这些中断信号远离 CPU, 使得一个必要时在稍后时

刻提出该查询的冗余 CPU 获得等同的消息 (Auskunft)。

在离开单独的工作模式时使计数器 IC 复位。随后向以下的程序位置跳回，在该程序位置上通过达到了由寄存器 MIR 预先给定的计数器值 IC 已实现了中断。在此之后，CPU 又执行由寄存器 MIR 预先给定的
5 机器指令数量，并在达到了寄存器值 MIR 时由计数器 IC 转换模式，并因此实现了外部事件的接受 (Annahme)。

有利地如下来构成 CPU 寄存器 MIR，使得通过软件或微码可以描述它们，以便确保，对于不同的使用领域在相适应的区间中实现中断处理，其方式是在中断处理的时窗之间相应地确定要执行指令的数量。

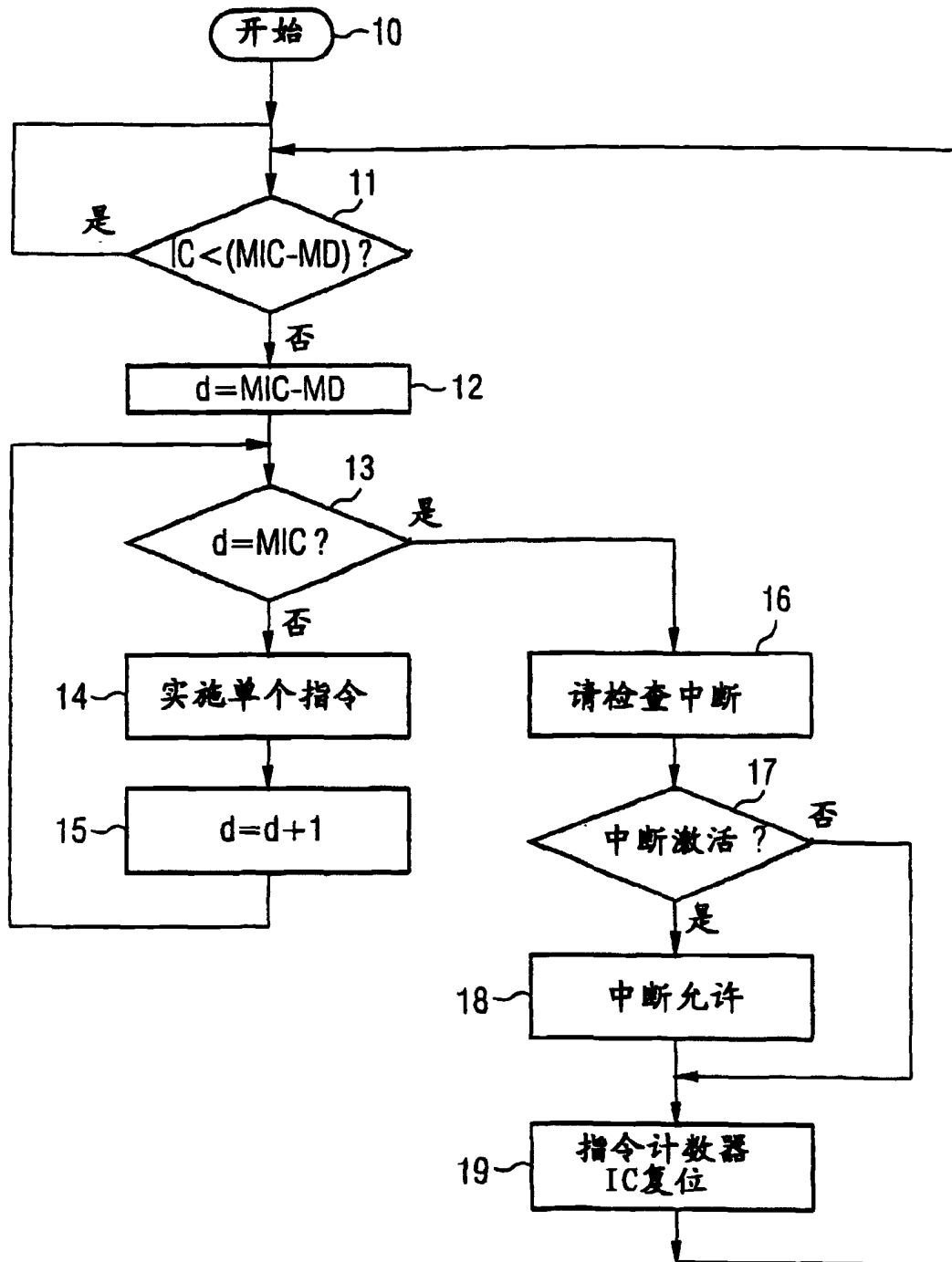


图 1

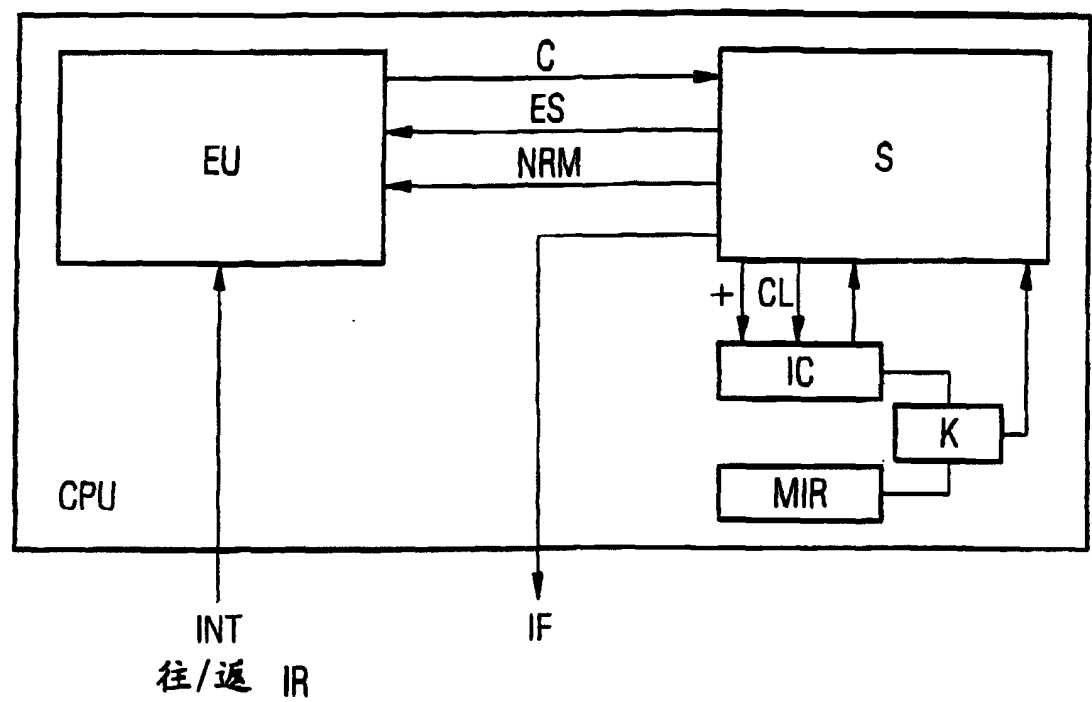


图 2

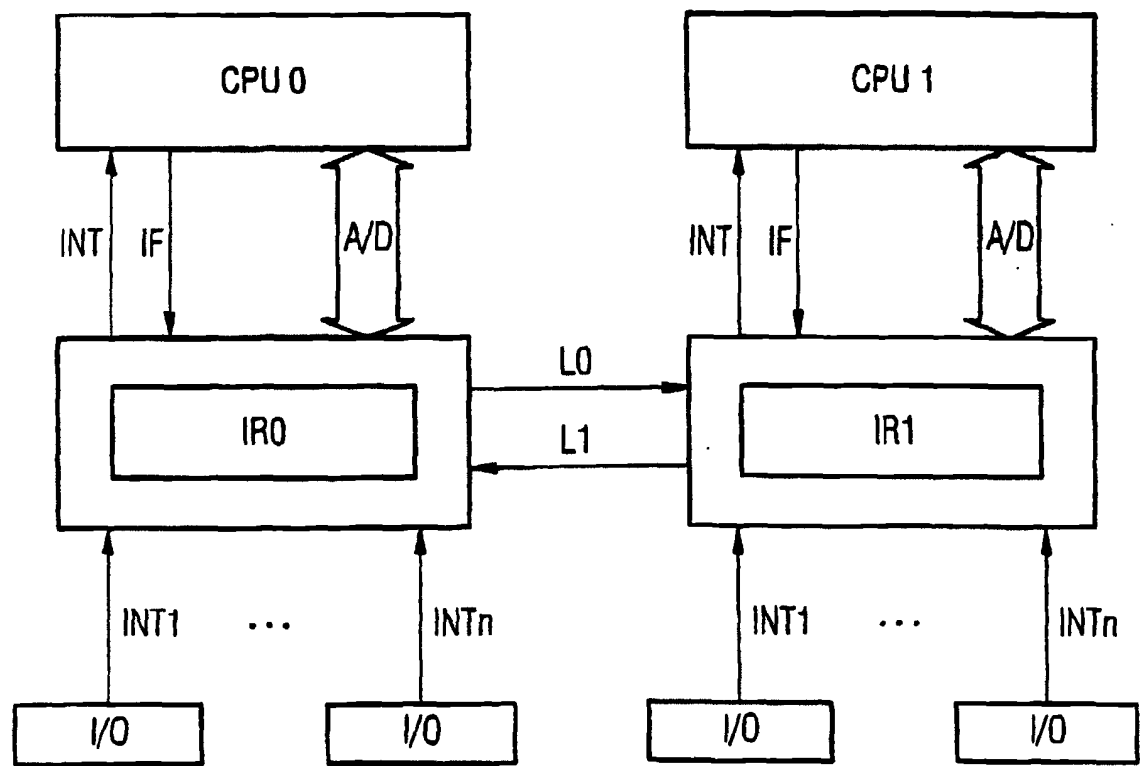


图 3