



(12) 发明专利

(10) 授权公告号 CN 102640119 B

(45) 授权公告日 2014. 09. 17

(21) 申请号 201080056820. 2

(22) 申请日 2010. 12. 02

(30) 优先权数据

102009054637. 5 2009. 12. 15 DE

(85) PCT国际申请进入国家阶段日

2012. 06. 14

(86) PCT国际申请的申请数据

PCT/EP2010/068720 2010. 12. 02

(87) PCT国际申请的公布数据

W02011/082904 DE 2011. 07. 14

(73) 专利权人 罗伯特·博世有限公司

地址 德国斯图加特

(72) 发明人 B. 米勒 M. 费尔希 Y. 科拉尼

H. 班斯基

(74) 专利代理机构 中国专利代理(香港)有限公

司 72001

代理人 李少丹 李家麟

(51) Int. Cl.

G06F 11/16(2006. 01)

(56) 对比文件

US 5784383 A, 1998. 07. 21, 全文.

EP 0969372 A2, 2000. 01. 05, 全文.

US 2009/0249034 A1, 2009. 10. 01, 全文.

DE 3009355 A1, 1981. 09. 17, 全文.

审查员 孙丹

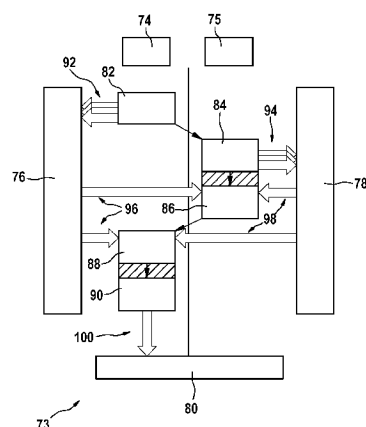
权利要求书2页 说明书6页 附图3页

(54) 发明名称

用于运行计算单元的方法

(57) 摘要

本发明涉及一种用于运行计算单元(2)的方法,所述计算单元具有至少两个计算内核(4,6),其中分别给至少两个计算内核(4,6)中的至少两个分配签名寄存器(18,20),所述签名寄存器具有多个输入,其中由至少两个计算内核(4,6)中的至少两个执行至少一个任务(12,14),其中利用每个任务(8,10)计算算法,其中将由每个计算内核(4,6)计算的结果(12,14)写入所分配的签名寄存器(18,20)中并且对写入签名寄存器(18,20)中的结果(12,14)进行比较。



1. 用于运行计算单元(2,50,73)的方法,所述计算单元具有至少两个计算内核(4,6,52,54,74,75),其中给至少两个计算内核(4,6,52,54,74,75)中的至少两个分别分配具有多个输入的签名寄存器(18,20,56,58,76,78),其中由计算内核(4,6,52,54,74,75)中的两个执行至少一个任务(8,10,64,66,82,84,86,88,90),其中利用每个任务(8,10,64,66,82,86,88,90)计算算法,其中将由每个计算内核(4,6,52,54,74,75)所计算的结果(12,14,70,71,92,94)写入所分配的签名寄存器(18,20,56,58,76,78)中并且对写入签名寄存器(18,20,56,58,76,78)中的结果(12,14,70,71,92,94)进行比较,

其特征在于,

以预先给定的顺序开始所述任务(8,10,64,66,82,84,86,88,90),其中由每个计算内核(4,6,52,54,74,75)执行被构造为应用作业的第一任务(8,10,64,66,82,84,86,88,90),由所述第一任务分别激活至少一个后续的被构造为比较作业的第二任务(8,10,64,66,82,84,86,88,90),其中利用被构造为比较作业的每个任务(8,10,64,66,82,84,86,88,90)对两个签名寄存器(18,20,56,58,76,78)的值进行比较。

2. 根据权利要求1所述的方法,其中由任务(8,10,64,66,82,84,86,88,90)对写入签名寄存器(18,20,56,58,76,78)中的结果(12,14,70,71,92,94)进行比较。

3. 根据权利要求1或2所述的方法,其中规定,所述算法应该在计算时提供相同的结果(12,14,70,71,92,94),其中由所述至少两个计算内核(4,6,52,54,74,75)中的至少两个计算相同的算法和/或其中由所述至少两个计算内核(4,6,52,54,74,75)中的至少两个计算不同的算法。

4. 根据权利要求1或2所述的方法,其中使至少两个任务(8,10,64,66,82,84,86,88,90)彼此同步。

5. 根据权利要求1或2所述的方法,其中由每个任务(8,10,64,66,82,84,86,88,90)将比较结果写入自己的存储单元中,和其中对比较结果相互进行比较。

6. 根据权利要求1或2所述的方法,其中用信号通知签名寄存器(18,20,56,58,76,78):最后的结果何时被写,其中由每个参与的计算内核(4,6,52,54,74,75)上的软件给签名寄存器(18,20,56,58,76,78)发送信号,或者其中在写第一结果之前通过相应的软件用信号通知相应的签名寄存器(18,20,56,58,76,78):多少值应该被写。

7. 计算单元,其具有至少两个计算内核(4,6,52,54,74,75),其中给所述至少两个计算内核(4,6,52,54,74,75)中的至少两个分别分配具有多个输入的签名寄存器,其中计算内核(4,6,52,54,74,75)中的两个被构造用于执行至少一个任务(8,10,64,66,82,84,86,88,90),利用每个任务(8,10,64,66,82,84,86,88,90)计算算法,并且将所计算的结果(12,14,70,71,92,94)写入所分配的签名寄存器(18,20,56,58,76,78)中并且其中计算单元(2,50,73)的至少一个组件被构造用于对写入签名寄存器(18,20,56,58,76,78)中的结果(12,14,70,71,92,94)进行比较,

其特征在于,

以预先给定的顺序开始任务(8,10,64,66,82,84,86,88,90),其中每个计算内核(4,6,52,54,74,75)执行被构造为应用作业的第一任务(8,10,64,66,82,84,86,88,90),所述第一任务分别激活至少一个后续的被构造为比较作业的第二任务(8,10,64,66,82,84,86,88,90),其中被构造为比较作业的每个任务(8,10,64,66,82,84,86,88,90)对两个签名寄

存器(18,20,56,58,76,78)的值进行比较。

8. 根据权利要求7所述的计算单元,其中这些计算内核(4,6,52,54,74,75)具有比较运算符,其被构造用于比较所述结果(12,14,70,71,92,94)。

用于运行计算单元的方法

技术领域

[0001] 本发明涉及用于运行计算单元的方法、计算单元、计算机程序和计算机程序产品。

背景技术

[0002] 已知不同的方法用于提高电子控制系统或例如可以应用在车辆中的电子控制设备的安全性。除了协调或投票方法(其中计算的结果由多数决定构成)之外,还存在所谓的比较方法,其中将两个计算单元的结果相互比较并且在不同的情况下切断控制设备。在后一种情况下应认为的是,控制设备通过切断在每种情况下采取安全的状态。在迄今的比较方法中,附加地使用硬件。计算单元以所定义的恒定的时钟偏移或者时钟相同地工作,所述计算单元的结果应该被比较。附加的比较逻辑比较所有数据,所述数据经由作为总线构造的网络离开计算单元,在此考虑可能的时钟偏移。

发明内容

[0003] 以此为背景介绍一种具有独立权利要求的特征的方法和计算单元以及具有并列权利要求的特征的计算机程序和计算机程序产品。本发明的其他扩展方案从从属权利要求和说明书中得出。

[0004] 此外利用本发明提供与至少两个 MISR (多输入移位寄存器)或 MIS 寄存器或多输入签名寄存器和因此具有多个输入的至少两个签名寄存器的异步比较。

[0005] 在本发明的范围中规定,在计算单元的至少两个计算内核或处理器中的至少两个(通常两个计算内核)上分别开始作业或任务或者过程,其中两个任务在一种实施形式中计算相同的算法并且在使用所分配的 MISR 的情况下存储中间结果。接着,对分配给至少两个作业的至少两个 MISR 的内容进行比较。至少两个计算内核中的至少两个一般不时钟准确地被同步,其中所述至少两个计算内核也可以称为内核(Core)并且在所述至少两个计算内核上计算作业。

[0006] 计算单元包括多个组件,也即至少两个计算内核,其被构造用于典型地冗余地执行任务或作业。作业通常是软件作业。作为计算单元的其他组件,设置签名寄存器,其中至少两个计算内核中的至少两个被分配给这种签名寄存器。此外,计算单元可以具有硬件模块,例如目标硬件或存储单元。组件的功能此外通过执行任务或作业来实施。在本发明的范围中,任务例如包括应用作业、比较作业和写作业。

[0007] 在具有所定义的恒定的时钟偏移的专用比较逻辑情况下规定,至少两个计算内核中的至少两个一般执行相同的程序代码。典型地不进行不同程序的暂时并行处理。这样的系统的性能相应于单处理器系统的性能。如果存在以下可能性:例如在 DCSL 系统中,暂时切断比较器或比较器装置,所述 DCSL 系统具有在运行模式和比较模式之间的转换装置,则必须耗费地对计算内核或处理器进行同步,如果应该将所述计算内核或处理器切换到比较模式的话。

[0008] 这里所述的发明能够在设计上实现冗余计算的参量的比较,而计算内核不必时钟

相同地或以恒定的时钟偏移地工作。不使用专用比较逻辑,其连接到计算内核上。代替地,为此目的使用在至少两个计算内核中的至少两个中所包含的比较运算符。该方法不检验所有结果(通常中间结果)的正确性,而是仅检验特别有关的结果。为了最小化待比较参量的数量,使用两个 MISR (多输入移位寄存器)。

[0009] 本发明计算单元被构造用于执行所介绍的方法的所有步骤。在此,也可以由计算单元的各个组件执行该方法的各个步骤。此外,计算单元的功能或计算单元的各个组件的功能可以被实施为该方法的步骤。此外,可能的是,该方法的步骤被实现为计算单元的至少一个组件或整个计算单元的功能。

[0010] 本发明此外涉及具有程序代码装置的计算机程序,以便当在计算机上、尤其是在本发明计算单元中执行计算机程序时执行所述方法的所有步骤。

[0011] 具有存储在计算机可读的数据载体上的程序代码装置的本发明计算机程序产品被构造用于当在计算机上、尤其是在本发明计算单元中执行计算机程序时执行所述方法的所有步骤。

[0012] 本发明的其他优点和扩展方案由说明书和附图得出。

[0013] 不言而喻,前述的和下面仍要阐述的特征可以不仅以分别所说明的组合的方式、而且以其他组合的方式或者单独地被使用,而不偏离本发明的范围。

附图说明

[0014] 图 1 以示意图示出用于执行本发明方法的第一实施形式的本发明计算单元的第一实施形式。

[0015] 图 2 以示意图示出用于执行本发明方法的第二实施形式的本发明计算单元的第二实施形式。

[0016] 图 3 以示意图示出用于执行本发明方法的第三实施形式的本发明计算单元的第三实施形式。

具体实施方式

[0017] 本发明根据实施例在附图中示意性地示出并且下面参照附图详尽地予以描述。

[0018] 所述图相关联地和全面地被描述,相同的附图标记表示相同的组件。

[0019] 在执行本发明方法的第一实施形式时本发明计算单元 73 的第一实施形式在图 1 中示意性地示出。该计算单元 73 包括分配有具有多个输入(MISR₀)的第一签名寄存器 76 的第一计算内核 74 (Core₀)、分配有具有多个输入(MISR₁)的第二签名寄存器 78 的第二计算内核 75 和目标硬件 80。

[0020] 该第一实施形式涉及一种方法,其中不使用同步点、例如屏障。这里通过以预先给定的顺序开始多个任务(作业)82、84、86、88、90 来保证时间依赖性。在此,每个作业激活其自己的后续作业。在该实施形式情况下以及在随后的实施形式情况下规定,签名寄存器(多输入移位寄存器,MISR)的值属于任务上下文并且在切换任务 82、84、86、88、90 时可以被确保或再次建立。所使用的任务 82、84、86、88、90 与相应的计算内核 74、75 绑定并且因此仅在这些计算内核上被实施。

[0021] 首先,在第一步骤中在第一计算内核 74 (Core₀) 上激活第一任务 82 或应用作业

A_0 并且由操作系统被带入实施。在实施期间,在应用作业 A_0 中执行算法并且将由应用开发者定义的结果 92、也即中间和最终结果写入第一签名寄存器 76 MISR₀ 中。在计算了所有结果 92 并且将所有中间和最终结果写入 MISR 中之后,在第二计算内核 75 (Core₁) 上激活第二任务 84,这里是应用作业 A_1 。

[0022] 在第二步骤时,在应用作业 A_1 中现在同样进行对结果 94 的计算。在此可以使用与在第一任务 82 (应用作业 A_0) 中相同的以及不同的算法。在此情况下决定的是,相同的结果 94 被计算并且相同的中间和最终结果被写入第二签名寄存器 78 中。以何种方式和方法计算结果 94 因此是无关紧要的。如果所有结果 94 被计算并且所有有关的中间和最终结果被写入第二签名寄存器 78 (MISR₁) 中,则作为第三任务 86,比较作业 B_1 在第二计算内核 75 (Core₁) 上被激活。

[0023] 在作为第三任务 86 的比较作业 B_1 中,在第三步骤中读取并且比较两个 MISR 的值。只有当在检验时确定出内容一致性时,作为第三任务 88 在第一计算核心 74 (Core₀) 上激活比较作业 B_0 。在所有其他情况下,执行相应的误差处理并且尤其是作为比较作业或后续作业 B_0 不激活或开始第四任务 88。

[0024] 在第四步骤中,利用比较任务 B_0 同样读取和比较两个 MISR 的值 96、98。只有当在检验时确定出内容一致性时,作为第五任务 90 在第一计算内核 (Core₀) 上激活写作业 C_0 。在所有其他情况下,执行相应的误差处理并且作为后续作业通常不激活或开始写作业 C_0 。

[0025] 所计算的结果 100 在第五步骤中由第五任务 90 (写作业 C_0) 写到目标硬件 80 上。

[0026] 因此冗余地通过第三任务 86 (比较作业 B_1) 和第四任务 88 (比较作业 B_0) 比较两个 MISR,以便在计算内核 74、75 中排除或发现误差。如果计算内核 74、75 之一应该错误地解释两个签名寄存器 76、78 的比较,假如所述两个签名寄存器例如内容不同的话,并且但是该区别由于误差是未知的,则另一计算内核 74、75 到目标硬件 80 上的写操作通过第五任务 90 (写作业 C_0) 阻止,因为通过其的处理链事先被中断。

[0027] 在第一实施形式的变型方案情况下,如果要执行的代码的顺序被保持,则可以将作业 A_1 和 B_1 以及作业 B_0 和 C_0 分别组合成一个任务或一个作业。单个作业或组合的作业是否更有意义,取决于应用、时间规划(调度)和作业的运行时间。对此可以逐情况地判决,由此可以优化调度。

[0028] 在用于运行计算单元 2 的方法的根据图 2 所示的第二实施形式中,在第一步骤中规定,出于安全原因冗余地在两个计算内核 4、6 或者内核上实施计算。在此,及时地激活两个任务 8、10 或作业 T_w 和 T_r ,其分别绑定在两个计算内核 4、6 (C_w 和 C_r) 或处理器之一上。可能地在通过操作系统 16 相应地支持情况下可以给作业分配相同的优先级。如果应该出于优先级原因而仅开始作业之一并且第二作业必须等待,则这可以被接受。不需要精确的同时性。

[0029] 在第二步骤中,首先利用每个任务 8、10 计算算法。将结果 12、14、也即有关的中间结果和所有有关的最终结果写到包括多个输入(MISR)的签名寄存器 18、20 上。在一种实施形式中,应用开发者确定将哪些结果 12、14 写入到 MISR 中。在此定义,将哪些结果 12、14 和 / 或哪些类型的结果 12、14 写入到相应的签名寄存器 18、20 中。因此,给每个计算内核 4、6 分配自己的签名寄存器 18、20,所述签名寄存器在开始任务 8、10 时被初始化。如果作业或任务 8、10 应该由同样冗余地工作和同样使用 MISR 的另一任务中断,则 MISR 连同任

务 8、10 和从而作业的上下文一起可能借助于操作系统 16 存储和稍后修复。

[0030] 在第三步骤中,在结束所有计算之后实现任务 8、20 的同步 36,例如通过所谓的屏障。由此确保,两个作业已经结束了其计算。

[0031] 在第四步骤中进行两个 MISR 的内容的比较。所述内容在正常情况下必须是相同的。每个作业将比较 22、24 的结果写入自己的存储单元 26、28 中,所述存储单元根据架构可以被构造为本地 RAM。

[0032] 在下一第五步骤中,作业在分别检验 30、32 的范围中比较其比较结果。为此,为作业进行同步 38。利用该措施可以确定:在计算内核 4、6 之一中的比较操作是否有错误的。如果情况如此,则可以忽略签名寄存器 18、20 中的差别。

[0033] 在最后的步骤中,执行写过程 40,其中由第一任务 8 T_w 将有关的数据从第一计算内核 4 C_w 写到目标硬件 42 上。

[0034] 这里所需要的同步 36、38 可以通过屏障或相互的事件机制例如按照开放式系统和其用于机动车中的电子装置的接口的 OSEK 操作系统标准来实现。

[0035] 为了同步 36、38 任务 8、10 或应用 T_w 和 T_R ,在扩展方案中也可以使用扩展的 MISR。为此对 MISR 扩展附加的寄存器,其中写到 MISR 中的值的数目被计数。该扩展的功能性可以完全地转化成硬件。与 MISR 的内容同样地,也对该计数寄存器初始化并且用值来修复。为了同步,作业 T_w 和 T_R 可以在一个过程之后用相应其他 MISR 的计数寄存器的值检验自己的计数寄存器的值。如果其他计数寄存器的值相应于自己的计数寄存器的值,则可以继续处理。可替代地,这里也可以预先给定预期的值并且根据该预期的值检验当前值。

[0036] 在误差情况下,可以在两个位置处识别比较误差,也即当在第四步骤中比较 MISR 时和 / 或当在第五步骤中比较比较结果时。

[0037] 在两种情况下,存在以下可能性:通过适当的机制中断进一步的处理。这如何实现一般取决于应用的安全要求。

[0038] 在所述的第二实施形式的第一变型方案中,由两个任务 8、10 或作业 T_w 和 T_R 计算不同的算法,但是所述算法应该提供相同的结果。因此,可以仅仍将在两个算法中相同地出现的值写入 MISR 中。这使得能够使用各种各样的软件并且发现软件误差。此外,可以写入 MISR 中的值的数目被减少,但是在此可以观察到结果处的舍入误差。

[0039] 在第二实施形式的第二变型方案中,可以取消将结果写入存储单元 26、28 (RAM) 中。在比较误差情况下,如果任务 8、10 提供不同的值,在两个计算内核 4、6 上利用相应的误差处理中断比较。为了实现这样的行为,在第二内核上的比较操作与其比较测试无关地同样被结束,这例如可以通过同步点的时间保护来实现。由此得出,在目标硬件 42 上不进行写操作。

[0040] 在用于运行在图 3 中示意性示出的计算单元 50 的本发明方法的第三实施形式情况下规定,在系统层面上接受对硬件的短时有错误的影响。由此相对于第二实施形式得出:取消太多的同步屏障并且因此实现较高的平均性能或平均功率和两个内核的较弱耦合。

[0041] 本发明计算单元 50 的第三实施形式的基本构造在图 3 中示意性示出。该计算单元 50 包括被构造为 CPU1 的第一计算内核 52 和被构造为 CPU2 的第二计算内核 54,其中给每个计算内核 52、54 分配具有多个输入的签名寄存器 56、58。此外,图 2 示出计算单元 50 的目标硬件 60 和操作系统 (OS) 62。这里也由操作系统 62 同时地或在几乎相同的时刻开

始两个任务 64、66 或应用或作业。

[0042] 如也在第二实施形式中那样,这里操作系统结构不起作用,也就是说,该方法可以与是否设置一个操作系统 62 或多个操作系统 62 无关地被执行。但是在该情况下,两个计算内核 52 或内核中的第一个在写过程 68 中在无先前比较的情况下将其结果 70 和中间结果直接写到目标硬件 60 上,这在误差情况时也适用于有错误的结果 70。第二任务 66 的写过程 72 的结果 71 仅仅被写入第二签名寄存器 58 中。此外确保,至少将有关的中间结果不仅写入目标硬件 60 上而且写入第一签名寄存器 56 (MISR) 中。

[0043] 对此存在多种可能性。写过程 72 例如可以在软件中作为单独的写指令来实现。但是也可能的是,如此设计硬件,使得将分支构建入写路径中并且硬件因此在两个位置处同时地或至少对于软件透明地执行写指令。该分支可以通过控制命令接通和关断。这意味着,分支在每个中间结果的情况下被接通,如果中间结果应该被写入到 MISR 中的话。因此软件仅仍需要用于写的指令。

[0044] 第二计算内核 54 (Core) 获得相同的任务 66,但是在其处将所有相应的写指令仅引导到所分配的签名寄存器 58 (MISR) 中。

[0045] 因此,两个计算内核 52、54 不准确地计算相同者,因为所述两个计算内核具有不同的写特性。但是差别可以自动化地和典型地通过离线工具 (OLT) 或相应的工具产生。

[0046] 执行比较操作包括根据图 3 所述的实施形式的子变型方案或选项,其中子变型方案可以相互任意地组合。

[0047] 利用第一选项可能的是,在任务 64、66 中或在作业本身中使用比较操作。在该情况下,两个任务 64、66 中的每一个在最后的写之后等待 MISR,直至由硬件必要时通过软件给任务 64、66 提供信号为止,所述信号通知:也存在相应其他的 MISR 结果。为了 MISR 知道最后的结果是何时被写的,要么由在每个参与内核上的软件向 MISR 发送信号,要么在写第一值之前通过相应的软件用信号通知相应的 MISR:应该写多少值。该作业根据该信号再次变为激活的,执行比较,必要时输出相应的误差反应并且然后被结束。

[0048] 在第二选项情况下规定,为了比较在每个计算内核 52、54 上放置自己的比较任务或自己的比较作业。这些作业由分别要监控的作业开始并且一旦其他内核的 MISR 值也可供使用,可以执行比较。

[0049] 第三选项的主题是在计算单元 50 中设置硬件比较器。该比较器例如可以独立地读取 MIS 寄存器。但是,典型地在结束作业之前,通过作业本身给比较器提供 MIS 寄存器。这也可以利用 OLT 实现。

[0050] 一般确保,与子变型方案无关地甚至在两个计算内核 52、54 上执行比较操作。通常,为此使用例如通过计时器的时间监控。在向 MISR 的最后写指令和结束比较操作之间的时间在此被限制。为此,在最终对 MISR 进行写之后,开始计时器。因此,通过时间监控定义在其内要结束比较操作的时间间隔的长度的值。如果在执行比较之前时间间隔到期,则探测到误差。

[0051] 在本发明方法的这里所述的实施形式情况下可以得出,计算内核 4、6、52、54、74、75 可能地处于无限循环或者出于其他原因提前结束或终止,例如由于不允许地访问受保护的存储单元。在任务 8、10、64、66、88、90 同步的所有位置处,可以通过时间条件识别这样的误差。首先达到同步点的计算内核 4、6、52、54、74、75 在那里仅等待有限的时间。如果超过

该时间,而无相应的其他计算内核 4、6、52、54、74、75 同样达到同步点,则引起误差。在这种运行时间监控情况下注意到,被等待的任务 8、10、64、66、82、84、86、88、90 可能已被中断,由此其总运行时间被影响。或者确定期限(最后期限),其足够长,以便一起考虑所有可能的中断,或者如果在第二计算内核 4、6、52、54、74、75 上计算不同于对应的作业的事物,则中断在第一计算内核 4、6、52、54、74、75 上的时间测量。

[0052] 本发明可以应用于所有控制设备,在所述控制设备中必须满足一定的安全要求,例如多核系统或已经具有多个同类的但是无关的计算单元的设备。

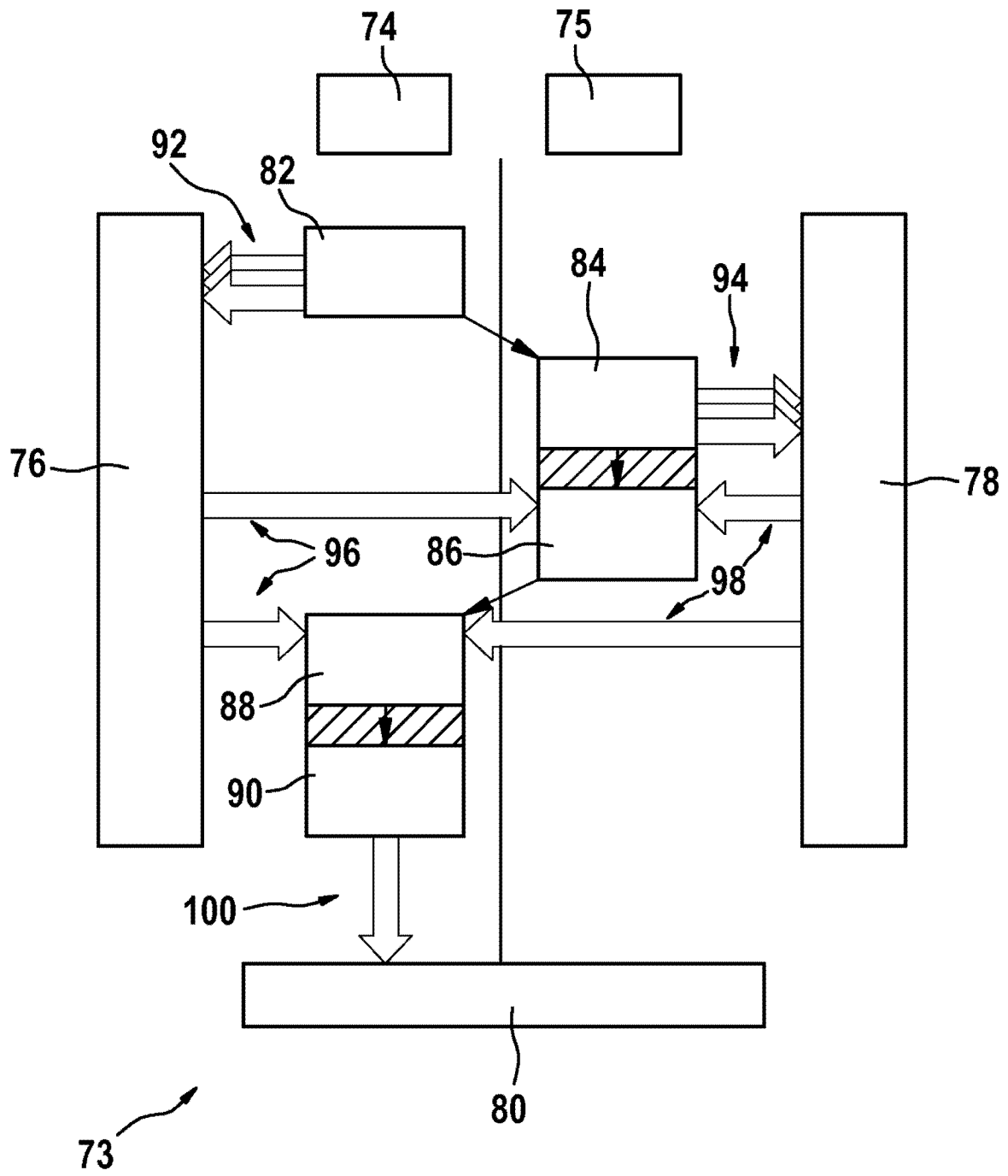


图 1

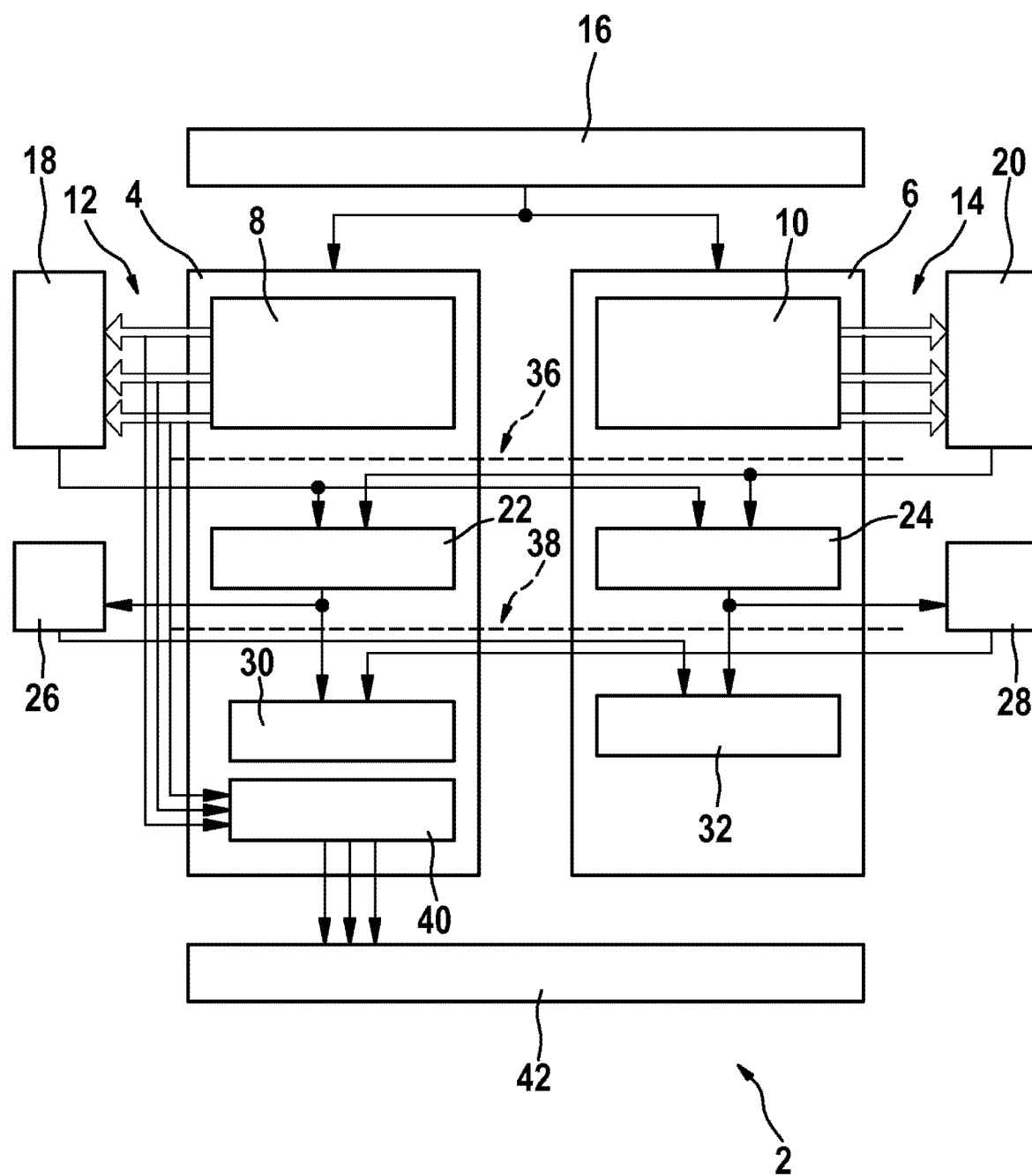


图 2

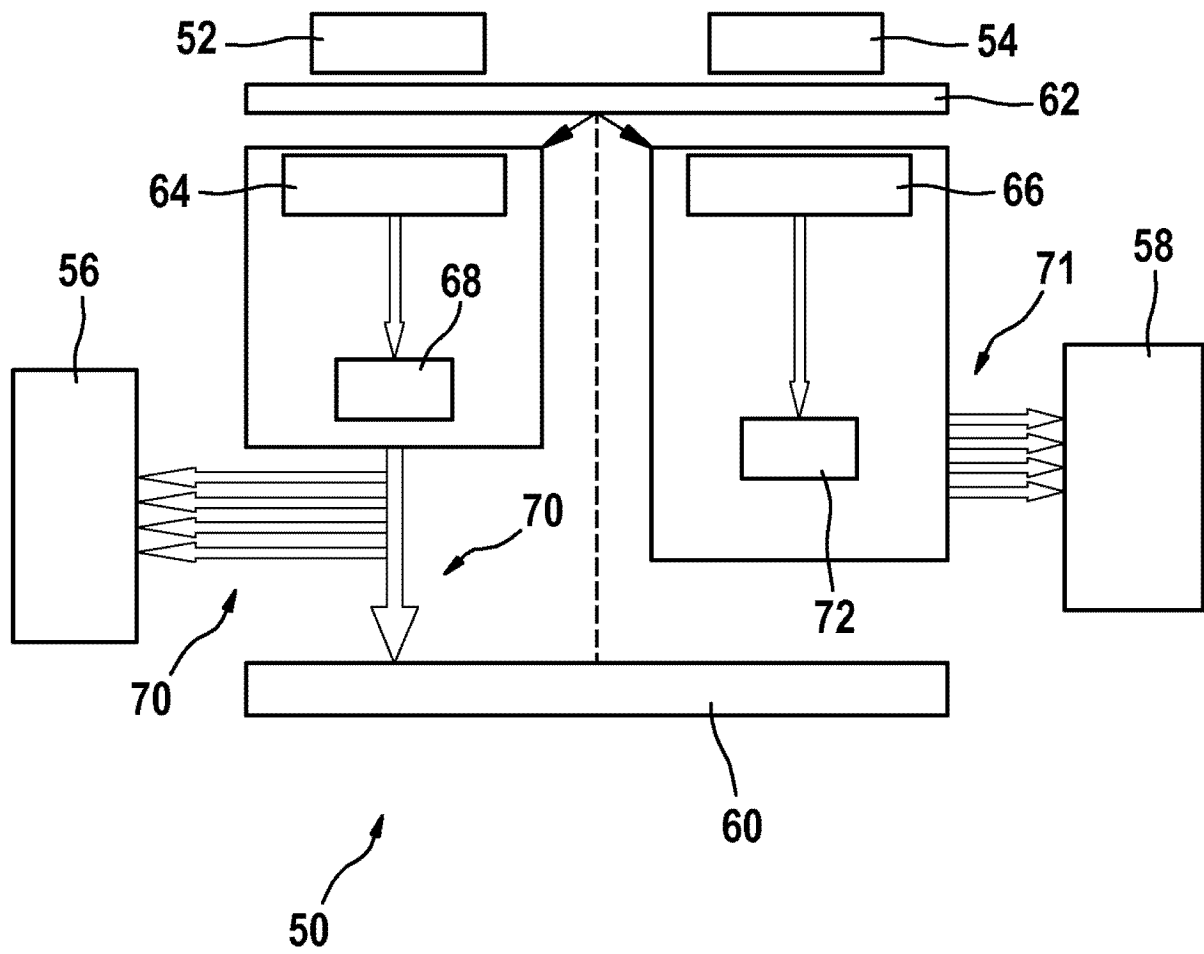


图 3