



(12) 发明专利申请

(10) 申请公布号 CN 103119564 A

(43) 申请公布日 2013. 05. 22

(21) 申请号 201180043805. 9

(74) 专利代理机构 北京市柳沈律师事务所  
11105

(22) 申请日 2011. 07. 01

代理人 谢强

(30) 优先权数据

102010027287. 6 2010. 07. 16 DE

(51) Int. Cl.

G06F 11/267(2006. 01)

(85) PCT申请进入国家阶段日

2013. 03. 12

(86) PCT申请的申请数据

PCT/EP2011/061098 2011. 07. 01

(87) PCT申请的公布数据

W02012/007295 DE 2012. 01. 19

(71) 申请人 西门子公司

地址 德国慕尼黑

(72) 发明人 C. 希尔德纳

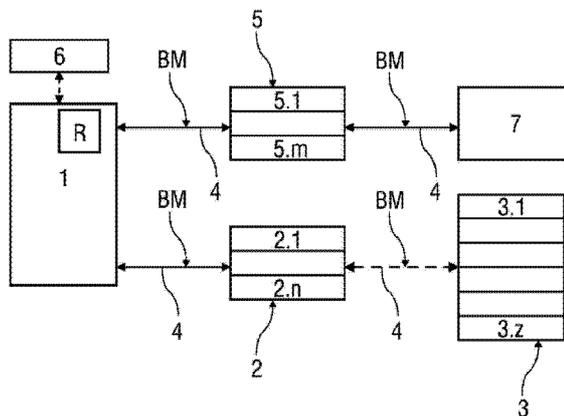
权利要求书1页 说明书5页 附图2页

(54) 发明名称

用于检查处理器的主存储器的方法和装置

(57) 摘要

本发明涉及用于检查处理器(1)的主存储器(3)的方法和装置,该处理器包括高速缓冲存储器(2)和多个寄存器(6)和/或寄存器存储器(R)。按照本发明,在运行存储器测试(T)之前,中断当时可能运行的启动序列,将临时对于存储器测试(T)所需的数据写入至少一个寄存器(6)或保持在那里,并且激活高速缓冲存储器(2)对主存储器(3)的存取。在此,通过高速缓冲存储器(2)这样进行对主存储器(3)的存取,使得比特模式(BM)被写入高速缓冲存储器(2)并通过该高速缓冲存储器写入主存储器(3)并且又从主存储器(3)中通过高速缓冲存储器(2)被读出并比较,其中,主存储器(3)的待测试的区域大于高速缓冲存储器(2)的大小,并且重新开始或继续可能在执行存储器测试(T)之前被中断的启动序列。



1. 一种用于检查处理器(1)的主存储器(3)的方法,该处理器包括高速缓冲存储器(2)和多个寄存器(6)和/或寄存器存储器(R),

其特征在于,在运行存储器测试(T)之前,中断当时可能运行的启动序列,将临时对于存储器测试(T)所需的数据写入至少一个寄存器(6)或保持在那里,并且激活高速缓冲存储器(2)对主存储器(3)的存取,其中,通过高速缓冲存储器(2)这样进行对主存储器(3)的存取,使得比特模式(BM)被写入高速缓冲存储器(2)并通过该高速缓冲存储器写入主存储器(3),并且又从主存储器(3)中通过高速缓冲存储器(2)被读出并比较,其中,所述主存储器(3)的待测试的区域大于所述高速缓冲存储器(2)的大小,并且在存储器测试(T)结束之后重新开始或继续可能在执行存储器测试(T)之前被中断的启动序列。

2. 根据权利要求1所述的方法,其特征在于,在存储器测试(T)结束之后将高速缓冲存储器(2)对主存储器(3)的存取分离。

3. 根据权利要求1所述的方法,其特征在于,在存储器测试(T)之前可选地运行的地址线和/或数据线的测试的情况下,只有在地址线和/或数据线的无错误的测试之后才开始所述存储器测试。

4. 根据上述权利要求中任一项所述的方法,其特征在于,作为比特模式(BM)使用具有零和/或一的模式。

5. 根据上述权利要求中任一项所述的方法,其特征在于,借助存储器测试(T)在操作系统开始之前测试所述主存储器(3)的一个或多个存储器单元(3.1至3.z)。

6. 根据上述权利要求中任一项所述的方法,其特征在于,将临时的、对于启动序列所需的数据在存储器测试(T)之前在寄存器(6)中进行中间存储,并且在存储器测试(T)期间或之后又使用/读出。

7. 根据上述权利要求中任一项所述的方法,其特征在于,将从所述比特模式(BM)的比较所产生的结果写入所述处理器(1)的寄存器(6)中。

8. 根据上述权利要求中任一项所述的方法,其特征在于,使用多个高速缓冲存储器(2,5),其中,一个高速缓冲存储器(2)用于存储临时数据,例如程序变量,而另一个高速缓冲存储器(2)用于存储程序代码。

9. 根据上述权利要求中任一项所述的方法,其特征在于,所述程序代码被存储在读取存储器(7)中。

10. 根据上述权利要求中任一项所述的方法,其特征在于,将运行存储器测试(T)的程序代码在主程序中或者在子程序中无跳回选项地实现。

11. 一种用于检查处理器(1)的主存储器(3)的装置,该处理器包括高速缓冲存储器(2)和多个寄存器(6),

其特征在于,在主存储器(3)和处理器(1)之间布置至少一个高速缓冲存储器(2,5),使得在存储器测试(T)的运行期间通过高速缓冲存储器(2,5)能够这样进行对主存储器(3)的存取,使得预定的比特模式(BM)能够被写入高速缓冲存储器(2,5)中并通过该高速缓冲存储器写入主存储器(3)中,并且又能够被从主存储器(3)中读出,其中,处理器(1)将从主存储器(3)中又读出的比特模式(BM)与额定比特模式进行比较,其他情况下高速缓冲存储器(2)与主存储器(3)分离,其中,高速缓冲存储器(2)的大小小于主存储器(3)的待测试区域。

## 用于检查处理器的主存储器的方法和装置

### 技术领域

[0001] 本发明涉及一种用于检查处理器的主存储器的方法。此外,本发明还涉及一种用于检查处理器的主存储器的装置。

### 背景技术

[0002] 在现代的计算机系统中通常的是,处理器具有尽可能大并且低成本的主存储器。对于通过处理器处理程序来说,在此需要不同的存储器存取,诸如加载/读取、存储和/或写数据、运算和/或命令。

[0003] 由于安全性和/或运行要求的原因,通常要不时地检查主存储器的部件。由主存储器的大小和处理器对主存储器的相对长的存取时间决定了,存储器测试需要许多时间并且例如违背了处理器的短的启动时间的要求。

### 发明内容

[0004] 因此,本发明要解决的技术问题是,加速用于检查处理器的主存储器的方法。此外,本发明要解决的技术问题是,简化用于检查主存储器的装置。

[0005] 就方法来说的技术问题按照本发明通过在权利要求 1 中给出的特征解决。关于装置,上述技术问题通过在权利要求 11 中给出的特征解决。

[0006] 本发明的有利构造是从属权利要求的内容。

[0007] 在用于检查处理器的主存储器(该主存储器具有多个存储器单元)的方法中,在进行存储器测试之前中断当时可能运行的启动序列并且将临时对于存储器测试所需的数据,诸如程序变量,写入至少一个寄存器或保持在那里。此外,激活高速缓冲存储器对主存储器的存取,其中对主存储器的存储器单元的存取通过高速缓冲存储器在存储器测试期间这样按照本发明进行,使得比特模式被写入高速缓冲存储器并通过该高速缓冲存储器写入主存储器并且从主存储器中通过高速缓冲存储器又被读出并比较,其中主存储器的待测试的区域大于高速缓冲存储器的大小。然后,重新开始或继续可能在执行存储器测试之前被中断的启动序列(Hochlaufsequenz)。

[0008] 于是,在存储器测试结束之后又将高速缓冲存储器与主存储器分离并且重新开始或继续在执行存储器测试之前被中断的启动序列。此外,将从主存储器又读出的比特模式与所产生的额定比特模式比较。特别地,在操作系统开始之前运行存储器测试,其中当时运行的启动序列是处理器的计算机程序的初始化程序。

[0009] 这样的方法在使用至少一个具有明显更短的存取时间的快速的高速缓冲存储器的条件下,可以实现更安全的和相对于现有技术来说明显更快的对于主存储器的测试过程。在此,逐级和/或逐块地进行主存储器的测试。

[0010] 处理器优选构造为微处理器。微处理器是其中所有组件布置在一个微芯片上的处理器。

[0011] 合适地,为了特别是逐字地、逐单元地和/或逐块地检查主存储器,作为比特模式

(Bitmuster)使用具有零和 / 或一的模式。在此,例如将主存储器划分为相同大的互相独立的区域,例如字、块,并且由此是逐字的或逐块的,这些区域可以在时间上不同地被读取或写入。例如将先后跟随的存储器字周期性地写入先后跟随的存储器组或块中或从中读取。通过测试先后跟随的存储器组或块可以缩短存取时间,因为至主存储器的数据总线的宽度大于处理器的字宽。

[0012] 为了主存储器的持续安全的运行,在逐单元地检查的情况下多次测试主存储器的一个或多个存储器单元。与之类似地,在逐块的测试中多次测试一个或多个存储器块。利用已知的值和该值的补数(逆)来描述每个存储器单元(Speicherzelle),从而每个比特必须一次保持为值“1”并且一次保持为值“0”。

[0013] 在一种简单的实施方式中,周期性地测试主存储器的一个或多个存储器单元和 / 或块。替换地或附加地,测试可以事件控制地进行。例如存储器测试可以由引导加载器(Bootlader)在操作系统开始之前运行。此外在错误的程序运行之后,由处理器自动激活主存储器测试并且运行至少一次或多次。

[0014] 合适地,在运行主存储器的存储器测试之前运行地址和 / 或数据线的测试。在此,地址和 / 或数据线的测试通过处理器对主存储器的直接存取而不是通过高速缓冲存储器来进行。由此,地址和 / 或数据线的测试以通常方式通过直接存取在存储器测试之前被运行。借助地址和 / 或数据线的测试在存储器测试之前的运行特别地识别制造错误,例如断线、短路。用于识别存储器芯片错误的存储器测试,特别地在两个前面的测试,即地址线和数据线的测试正常地、也就是无错误地进行的条件下才运行。

[0015] 为了避免数据损失,将临时的程序数据或变量在存储器测试期间、特别是在地址和 / 或数据线的测试之后并且在存储器测试之前,在寄存器中进行中间存储(zwischengespeichert)。在存储器测试运行之后这些程序数据或变量又可以被读出并且写入到高速缓冲存储器和 / 或主存储器中。特别地,如果用于中间存储临时数据的寄存器数量不够,则重新开始并且由此重复启动序列,以便重新建立临时数据。对于在存储器测试期间丢失的临时数据不再被需要的情况,继续启动序列。

[0016] 为了分析所运行的存储器测试,将由比特模式的比较得出的结果写入处理器的寄存器中。

[0017] 在本发明的扩展中,使用多个高速缓冲存储器,其中一个高速缓冲存储器用于存储程序代码(也称为程序或指令高速缓冲存储器)而另一个高速缓冲存储器用于存储当前的,即,当时所使用的数据和 / 或变量,诸如程序变量和地址数据,(也称为数据高速缓冲存储器)。在此,存储程序代码的高速缓冲存储器被用于加速对程序代码的存取。数据高速缓冲存储器(即,存储临时数据的高速缓冲存储器)被用于特别是存储和加速对主存储器的存取。

[0018] 优选地,程序代码和特别是包含了存储器测试的程序代码,被存储在读取存储器(ROM = 只读存储器)中。

[0019] 在本发明的扩展中,在主程序中实现包含了存储器测试的程序代码。由此,避免了子程序调用,后者要求高性能的堆栈存储器。替换地,存储器测试可以作为子程序实现。在这种情况下,在子程序调用中实现程序的继续;由于从高速缓冲存储器中可能丢失的程序变量,而防止了跳回到主程序。

[0020] 关于用于检查处理器的主存储器的装置,按照本发明在主存储器和处理器之间这样布置高速缓冲存储器,使得在存储器测试期间可以通过高速缓冲存储器这样进行对主存储器的存储器单元的存取,使得可以将可预定的比特模式写入高速缓冲存储器中,特别是其存储器单元中并且经过它们写入主存储器中,特别是其存储器单元中,并且从它们中通过高速缓冲存储器又读出,其中处理器将从主存储器中又被读出的比特模式与额定比特模式比较,在其他情况下高速缓冲存储器与主存储器分离并且提供用于接受临时数据、特别是程序数据,其中高速缓冲存储器的大小小于主存储器的待测试区域。

[0021] 高速缓冲存储器作为具有快速存取时间的中间存储器的使用,实现了主存储器的存储器测试的加速。优选地,在此高速缓冲存储器可以在存储器芯片本身上集成。

#### 附图说明

[0022] 以下根据实施例参考附图详细描述本发明的其他优点、特征和细节。其中,

[0023] 图 1 示意性示出了用于检查处理器的主存储器的装置的实施方式的框图,和

[0024] 图 2 示意性示出了对于主存储器的存储器测试的流程图。

[0025] 互相相应的部分在所有附图中具有相同的附图标记。

#### 具体实施方式

[0026] 图 1 示意性示出了用于检查处理器 1 的主存储器 3 的装置的实施方式的框图。处理器 1 可以是微处理器,其组件被布置在一个微芯片上(未详细示出)。

[0027] 在处理器 1 和主存储器 3 之间布置了高速缓冲存储器 2,作为中间存储器或缓冲存储器。处理器 1 以常规方式通过数据、地址、错误和控制线 4 与高速缓冲存储器 2 相连,并且该高速缓冲存储器与主存储器 3 相连。

[0028] 在此,处理器 1 为了检查主存储器 3 在中间连接高速缓冲存储器 2 的条件下存取主存储器 3。仅在存储器测试期间进行通过高速缓冲存储器 2 对主存储器 3 的存取,其他情况下高速缓冲存储器 2 与主存储器 3 分离(通过虚线 4 表示)。

[0029] 主存储器 3 是处理器 1 的通常的大工作存储器。例如通过 8 比特和 / 或 16 比特地址信道或线进行对主存储器 3 的存取。主存储器 3 在此包括存储器元件的组,它们被综合为存储器单元 3.1 至 3.z (= 最小可寻址的单位)。每个存储器单元 3.1 至 3.z 包括 8 比特 (=1 字节)。多个存储器单元 3.1 至 3.z,例如 4 或 8 个存储器单元 3.1 至 3.z,可以被综合为一个 32 比特或 64 比特存储器字、存储器块、存储器页和 / 或存储器组。由此可以对主存储器 3 的特别是先后跟随的、相同大的并且互相独立的区域进行寻址。

[0030] 高速缓冲存储器 2 是快速的中间或缓冲存储器,其被布置在处理器 1 的寄存器存储器 R 和主存储器 3 之间。在示出的实施例中,将高速缓冲存储器 2 布置在处理器 1 外部并且由此没有布置在处理器芯片上。替换地,高速缓冲存储器 2 也可以布置在处理器芯片上(未示出)。

[0031] 高速缓冲存储器 2 具有比主存储器 3 更小的例如以千或兆字节范围,例如 1MB 的、具有在纳秒范围内的最短的存取时间的存储器容量,而具有在毫秒范围内的低的存取时间的主存储器 3 的存储器容量处于兆、千兆或兆兆字节范围,例如 512MB。

[0032] 高速缓冲存储器 2 与主存储器 3 类似具有多个存储器单元 2.1 至 2.z,它们按照规

定相应地综合或分割为字、组和 / 或块,它们表示互相独立的地址范围。高速缓冲存储器 2 用于存储当前使用的数据和 / 或变量,特别是动态程序变量,如地址数据。

[0033] 此外,该装置还具有带有多个存储器单元 5.1 至 5.z 的至少另一个高速缓冲存储器 5。该另一个高速缓冲存储器 5 用于存储程序代码。在此,在高速缓冲存储器 5 之后连接读取存储器 7,例如 ROM 存储器(ROM= 只读存储器),在所述读取存储器中存储了程序代码。

[0034] 图 2 示意性示出了对于主存储器 3 的存储器测试 T 的流程图。

[0035] 一般地,定期检查主存储器 3。优选地,在操作系统开始之前进行存储器测试 T。

[0036] 相对于常规的测试方法,在按照本发明的方法中存储器 1 不是直接存取主存储器 3,而是通过高速缓冲存储器 2。在此,这样进行对处理器 1 的存取,使得至少一个比特模式 BM 被写入高速缓冲存储器 2 中并通过该高速缓冲存储器写入主存储器 3 中并且从该主存储器中又被读出。然后,将从主存储器 3 中又读出的比特模式 BM 与所产生的额定比特模式进行比较。在这两个比特模式不相同的情况下,推导出主存储器 3 的错误或传输错误。

[0037] 合适地,作为比特模式 BM,产生按照预定的长度,例如 8 比特、16 比特、32 比特长具有零和 / 或一的模式。例如作为比特模式 BM 使用地址数据。

[0038] 在该检查方法中多次地、特别是周期性地测试主存储器 3 的一个或多个存储器单元 3.1 至 3.z。

[0039] 具体地,在激活存储器测试 T 的情况下并且在运行存储器测试 T 之前,在第一步骤 S1 中可选地以常规方式通过直接存取和由此无需中间存储地进行地址线和 / 或数据线的测试(虚线示出)。在地址和 / 或数据线的可选的测试中,根据所产生的比特模式来测试确定的地址,以便识别制造错误,特别是断线和 / 或短路。只有在事先进行的地址和 / 或数据线的测试是无错误地进行的,才开始并运行实际的存储器测试 T。

[0040] 在步骤 S2 中对于存储器测试 T 的、例如程序测试的程序代码利用对写入和 / 或读取操作的检查来进行。程序代码的调用优选在主程序中进行。

[0041] 在为了测试主存储器 3,处理器 1 对高速缓冲存储器 2 进行存取之前中断当时运行的启动序列。

[0042] 可选地或附加地,在步骤 S2.1 中将临时的和非待测试的但是对于存储器测试 T 来说是所需的数据,诸如临时的程序变量、数据,在寄存器 6 和 / 或寄存器存储器 R 的一个中进行中间存储。寄存器存储器 R 例如是处理器 1 的当时不用的寄存器。替换地,寄存器存储器 R 和 / 或另外的寄存器 6 也可以布置在处理器 1 的外部。

[0043] 然后,在第三步骤 S3 中激活对高速缓冲存储器 2 的存取以测试主存储器 3。在此会产生无效的数据,特别是临时的程序数据。

[0044] 具体地,激活在处理器 1 中实现的测试例程,借助该测量例程,不是直接在主存储器 3 上,而是在高速缓冲存储器 2 上执行待测试的写入和 / 或读取操作,如命令、请求。在此,测试例程是例如按照机器语言(汇编语言)或更高高级的编程语言在使用优化的编译器的条件下写入的,以便在寄存器中而不是在高速缓冲存储器 2 的堆栈中或在主存储器 3 中保持并存储临时的数据。

[0045] 然后,在步骤 S4 中相应于所实现的测试例程运行存储器测试 T。借助存储器测试 T,关于种类、频率和 / 或范围来定义和控制对所涉及的高速缓冲存储器 2 的存储器单元 2.1 至 2.z 的存取。作为存取的种类,例如定义和控制读取或写入操作。作为范围,例如规定主

存储器 3 的待测试的存储器单元 3.1 至 3.z 和 / 或存储器块的数量。

[0046] 具体地,在第一循环中,将与主存储器 3 的待测试的区域的大小相应数量的比特模式 BM 写入高速缓冲存储器 2 中。在此,高速缓冲存储器 2 小于主存储器 3 的待测试区域,从而在运行存储器测试 T 的情况下利用比特模式 BM 覆盖高速缓冲存储器 2,并且比特模式 BM 被写入主存储器 3 的待测试区域。然后在第二循环中,将主存储器 3 的待测试区域的比特模式 BM 通过高速缓冲存储器 2 又读出并且与额定比特模式比较。

[0047] 例如,逐块、逐字和 / 或逐单元的比特模式 BM 被写入高速缓冲存储器 2 的预定的存储器单元 2.1 至 2.z 中并从这些存储器单元中被写入主存储器 3 的对应的存储器单元 3.1 至 3.z 中,并且从这些存储器单元中通过高速缓冲存储器 2 又读出并与额定比特模式比较。将比较的结果例如中间存储在处理器 1 的预定的其他存储器 6 和 / 或寄存器存储器 R 中。

[0048] 然后,在步骤 S5 中,在结束对于存储器测试 T 的测试例程之后可选地将通过高速缓冲存储器 2 对主存储器 3 的存取解除激活。

[0049] 在步骤 S6 中,检查从主存储器 3 中读出的比特模式 BM 与额定比特模式的比较的存储的结果。为此,比较的结果被存储在寄存器 6 中和 / 或处理器 1 的寄存器存储器 R 中的一个中,从而该结果可以由在处理器 1 中实现的分析例程分析。

[0050] 然后在步骤 S7 中,又读出在步骤 S3 中和可选地在步骤 S2.1 中中间存储的临时数据,并且开始或启动处理器 1 的初始的启动序列,即,在存储器测试 T 运行之前被中断的启动序列被重新开始或可选地继续。

[0051] 存储器测试 T 可以多次地,例如周期性地重复或者事件控制地开始。特别地,存储器测试 T 在操作系统开始之前被运行。也可以执行其他步骤。

[0052] 存储器测试 T 特别地作为程序代码或测试例程在处理器 1 的主程序中实现。替换地,存储器测试 T 的程序代码可以作为子程序实现。在这种情况下将随后程序仅作为子程序调用,因为由于借助高速缓冲存储器 2 的存储器测试 T,临时数据,特别是地址数据在高速缓冲存储器 2 中可能丢失。为了实现安全的程序运行,由此将所有其他程序作为子程序调用。

[0053] 借助按照本发明的用于通过对高速缓冲存储器 2 的存取来检查主存储器 3 的方法,可以个别地测试主存储器 3 的所有存储器单元 3.1 至 3.z。通过使用高速缓冲存储器 2,明显地加速了存储器测试 T。

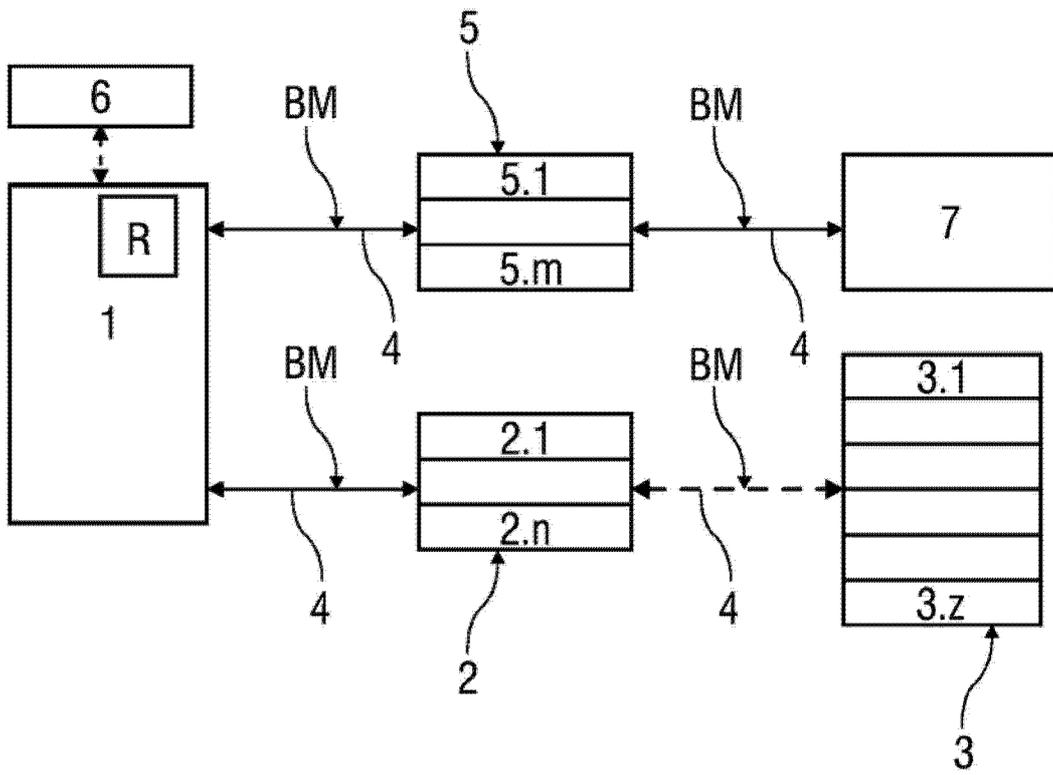


图 1

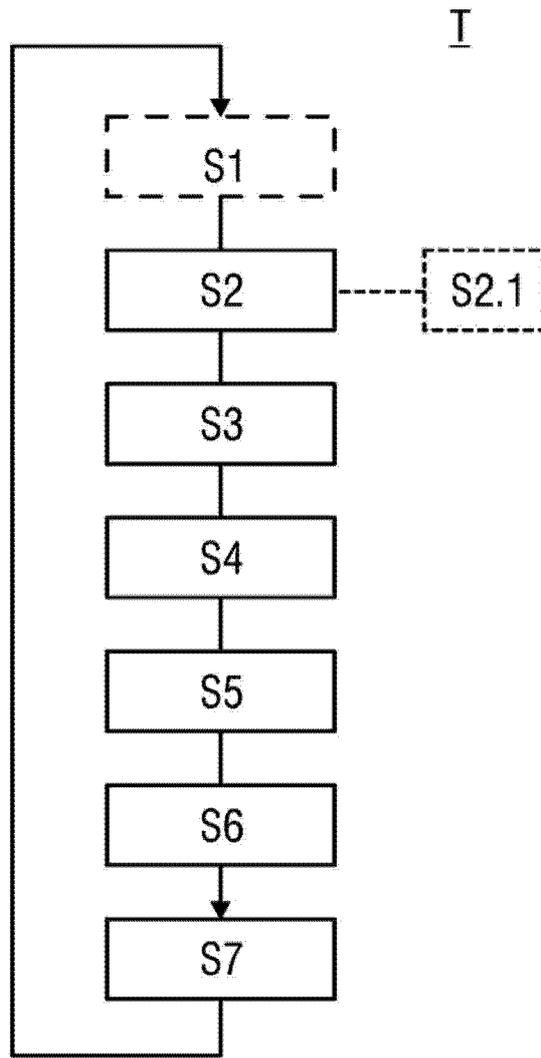


图 2