



(12)发明专利

(10)授权公告号 CN 104156263 B

(45)授权公告日 2017.06.16

(21)申请号 201410195247.3

(51)Int.Cl.

(22)申请日 2014.05.09

G06F 9/48(2006.01)

(65)同一申请的已公布的文献号

申请公布号 CN 104156263 A

(56)对比文件

US 2012/0226843 A1,2012.09.06,

CN 1952899 A,2007.04.25,

CN 102822802 A,2012.12.12,

US 2007/0143514 A1,2007.06.21,

(43)申请公布日 2014.11.19

(30)优先权数据

1308598.0 2013.05.14 GB

审查员 孙骏童

(73)专利权人 国际商业机器公司

地址 美国纽约

(72)发明人 A·柯尼格 C·里彻特纳

P·M·洛博 B·C·布洛克

T·戈洛克勒

(74)专利代理机构 北京市中咨律师事务所

11247

代理人 于静 张亚非

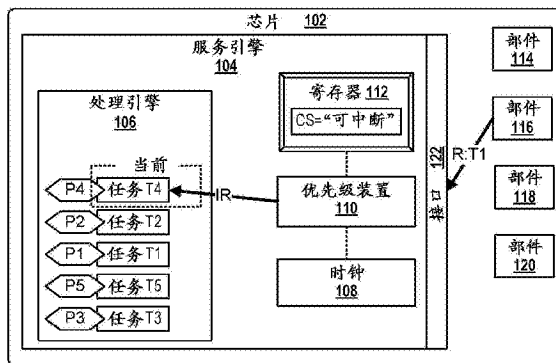
权利要求书3页 说明书8页 附图2页

(54)发明名称

芯片部件管理任务中断方法、芯片及包括所述芯片的组件

(57)摘要

本发明涉及一种芯片部件管理任务中断方法、芯片及包括所述芯片的组件。一种集成电路芯片,包括服务引擎,其适于管理芯片的一个或多个部件并包括:处理引擎;数据存储器,存储任务中的当前执行任务的当前状态;接口,用于接收执行任务中的另一任务的请求,当前执行任务具有第一优先级,另一任务具有第二优先级;时钟,适于测量接收所述请求与当前时间之间已经过的时间间隔;控制引擎,适于在以下情况中断当前执行任务并触发所请求任务的执行:a)第二优先级高于第一优先级,及b1)所存储的当前状态指示当前任务可正常中断;和/或b2)所测量的时间间隔超过阈值,由此在b2)中执行中断和触发而不考虑当前执行的任务是否可被或已被正常终止。



1. 一种包括服务引擎(104)的集成电路芯片(102),所述服务引擎(104)适于经由一个或多个任务(T1-T5)管理所述芯片的一个或多个部件(114、116、118、120),所述服务引擎包括:

-处理引擎(106),其适于执行所述一个或多个任务;

其特征在于,所述服务引擎进一步包括:

-数据存储器(112),其存储所述任务中的一个当前执行的任务(T4)的当前状态(CS);

-接口(122),其用于接收执行所述任务中的另一任务(T1)的请求(R),所述当前执行的任务(T4)具有第一优先级(P4),所述另一任务(T1)具有第二优先级(P1);

-时钟(108),其适于测量接收所述请求与当前时间之间已经过的时间间隔;

-控制引擎,其适于:

响应于所述第二优先级(P1)高于所述第一优先级(P4),并且所测量的时间间隔超过阈值,中断所述当前执行的任务并触发所请求的任务的执行,而不考虑所述当前执行的任务是否可被正常终止或者是否已被正常终止;

响应于所述第二优先级(P1)高于所述第一优先级(P4),并且所存储的当前状态指示当前任务可正常中断,中断所述当前执行的任务并触发所请求的任务的执行;以及

响应于所述第二优先级(P1)高于所述第一优先级(P4)并且所存储的当前状态指示当前任务不可正常中断以及所测量的时间间隔不超过阈值,触发当前执行的任务的正常终止或者等待当前执行的任务自己主动将当前状态值设为“可中断”后,终止或中断当前执行的任务并触发所请求的任务的执行。

2. 根据权利要求1的集成电路芯片,其中所述请求包括指示所请求的任务的优先级的属性值。

3. 根据权利要求1的集成电路芯片,其中根据每个任务的类型给出该任务的优先级。

4. 根据权利要求1-3之一所述的集成电路芯片,

-其中所请求的任务是多个第一任务中的一个,并且所述当前执行的任务是多个第二任务中的一个;

-其中所述第一任务负责引导、初始化和/或关闭所述芯片的处理器核,负责防止所述处理器核过热,和/或负责管理所述处理器核的供电;

-其中所述第二任务负责监视所述部件,具体地说,负责监视所述芯片的处理器核,负责对所述部件计时,负责创建使用统计信息,和/或负责识别所述部件的错误和无效状态;

-其中每个第一任务的优先级高于所述第二任务中的任一任务。

5. 根据权利要求1-3之一所述的集成电路芯片,

-其中所述处理引擎适于持续监视所述当前执行的任务的所述当前状态,并且适于相应地更新存储在所述数据存储器中的所述当前状态;和/或

-其中所述当前执行的任务包括可执行指令,当被执行时,所述可执行指令将存储在所述数据存储器中的所述当前状态设为在所述指令中指定的值。

6. 根据权利要求1-3之一所述的集成电路芯片,其中中断所述当前执行的任务包括:

-触发所述当前执行的任务的正常终止或者等待所述任务自己主动正常终止;以及

-在完成所述正常终止之后或在完成所述正常终止期间,将所述当前状态“可中断”存储到所述数据存储器(112),从而使得所述控制引擎能够根据情况b1)触发所请求的任务的

执行。

7. 根据权利要求1-3之一所述的集成电路芯片，

-其中在所测量的时间间隔超过所述阈值的情况下，所述服务引擎适于将存储在所述数据存储器中的所述当前状态设为“可中断”，而不考虑所述当前执行的任务是否已被正常终止或是否可被正常终止，从而使得所述控制引擎能够根据b2) 触发所请求的任务的执行。

8. 根据权利要求1-3之一所述的集成电路芯片，其中所述请求是所述服务引擎所支持的多个请求中的一个，每个所支持的请求对应于所述任务中的一个，每个所述请求被分配一个请求特定的阈值，其中相对于特定于所述请求的阈值而执行所测量的时间间隔是否超过所述阈值的判定。

9. 根据权利要求8的集成电路芯片，其中由所述部件中的一个或片上部件管理单元生成所述请求，所述请求包括其请求特定的阈值。

10. 根据权利要求1-3之一所述的集成电路芯片，其中所述服务引擎是片上微控制器，其中所述部件包括硬件和/或固件，并且其中所述任务是存储在所述芯片上的计算机可解释指令。

11. 根据权利要求1-3之一所述的集成电路芯片，其中所述处理引擎仅可操作以一次执行一个任务。

12. 根据权利要求1-3之一所述的集成电路芯片，其中包括所述当前状态的所述数据存储器是以单个位值的形式存储所述当前状态的寄存器，所述当前状态为“可中断”或“不可中断”。

13. 根据权利要求1-3之一所述的集成电路芯片，其中所存储的当前状态为“可中断”或“不可中断”或“可有条件地中断”，其中所存储的状态值为“可有条件地中断”的当前执行的任务的中断使得所述控制引擎能够在所请求的任务或所述芯片的部件满足额外条件的情况下中断该当前执行的任务。

14. 一种包括根据任一上述权利要求的多个集成电路芯片(102.1-102.4)的组件(202)，每个芯片包括一个服务引擎(104.1-104.4)，每个芯片的部件包括一个由其相应服务引擎管理的处理器核(116.1-116.4)。

15. 一种用于经由一个或多个任务管理集成电路芯片(102)的一个或多个部件(114、116、118、120)的计算机实现的方法，所述芯片包括服务引擎(104)，所述方法由所述服务引擎执行并且包括：

-执行一个或多个任务(T1-T5)中的一个当前任务；

-在所述芯片的数据存储器(112)中存储当前执行的任务(T4)的当前状态(CS)；

-从所述部件之一接收执行所述任务中的另一任务(T1)的请求(R)，所述当前执行的任务(T4)具有第一优先级(P4)，所述另一任务(T1)具有第二优先级(P1)；

-测量接收所述请求与当前时间之间已经过的时间间隔；

-响应于所述第二优先级(P1)高于所述第一优先级(P4)，并且所测量的时间间隔超过阈值，中断所述当前执行的任务并触发所请求的任务的执行，而不考虑所述当前执行的任务是否可被正常终止或者是否已被正常终止；

-响应于所述第二优先级(P1)高于所述第一优先级(P4)，并且所存储的当前状态指示当前任务可正常中断，中断所述当前执行的任务并触发所请求的任务的执行；以及

-响应于所述第二优先级(P1)高于所述第一优先级(P4)并且所存储的当前状态指示当前任务不可正常中断以及所测量的时间间隔不超过阈值,触发当前执行的任务的正常终止或者等待当前执行的任务自己主动将当前状态值设为“可中断”后,终止或中断当前执行的任务并触发所请求的任务的执行。

芯片部件管理任务中断方法、芯片及包括所述芯片的组件

技术领域

[0001] 本发明涉及集成电路芯片领域。更具体地说,本发明涉及芯片部件管理任务的中断控制。

背景技术

[0002] 现在的集成电路芯片包括大量需要管理的功能和部件。一般而言,使用充当服务模块并可与某些片上传感器和控制元件互操作的专用芯片硬件部件执行所述管理。

[0003] 有些管理任务必须比其它任务更迫切地被执行,例如,用于防止不一致或有害的系统状态的任务。存在两种方法和对应的芯片体系结构来确保更快地执行高度紧急的任务:A) 包括单个管理单元的芯片,这种芯片使用当前执行的任务的无条件中断,其适于紧急任务,以及B) 包括两个或更多管理单元的芯片,其中选择性地保留至少一个管理单元来执行低优先级任务,并且其中选择性地保留至少一个另一管理单元来执行被请求的高优先级任务。

[0004] 根据方法A), 芯片使用单个管理单元执行芯片部件管理任务,此时一次只能执行一个任务。在需要执行紧急任务的情况下,所述紧急任务立即由管理单元执行,并且可能突然终止当前正在所述时刻执行的其它所有任务。此类芯片制造成本低,但是突然中断当前执行的任务可能导致不一致的系统状态,甚至可能损坏芯片部件。这种芯片体系结构不支持正常终止,因为选择正常终止会承受当前执行的任务陷入无限循环的风险,当前任务可能再也不能正常终止,因此紧急任务再也无法执行。当紧急任务是为了防止损坏芯片部件时,无法执行紧急任务可能导致损坏芯片。

[0005] 根据方法B), 芯片包括多个管理单元。在这种体系结构中,低优先级任务被选择性地分配给所述管理单元中的一个,而紧急任务被选择性地分配给所述管理单元中的另一个。因为针对紧急任务独占地保留一个管理单元,所以可确保立即执行紧急任务。这种方法的缺点是为紧急任务保留的管理单元多数时间处于空闲状态。因此,可用硬件资源的利用率不高。而且,此类多管理单元芯片的制造成本更高。

[0006] 例如,US 8347158 B2公开了一种跨功能总线传输请求的方法。此方法无法保证所请求的事务获取最大服务时间。

[0007] US 7984247 B2对进站请求设置优先级,但不提供能够确保正常地执行其它任务作业终止的系统。

发明内容

[0008] 本发明的各实施例的目标是提供在不对芯片或其部件造成损害的情况下,及时执行紧急任务的改进型芯片和对应的方法。

[0009] 所述目标通过独立权利要求的主题解决。从属权利要求中描述有利的实施例。如果未明确指出,则实施例的特征可自由地彼此组合。

[0010] 如此处使用的那样,“服务引擎”是用于管理(即,启动、监视和/或控制)芯片的一

个或多个部件的微控制器。优选地,所述服务引擎单纯地包括硬件、固件,或者包括硬件和固件的组合。根据各优选实施例,所述服务引擎被实现为片上部件。这可提供能够自引导和/或包括片上电源管理、维护、监视、计时、错误识别和错误恢复逻辑的“智能”芯片。在两个或更多请求实例同时请求特定部件的情况下,所述服务引擎的任务是判定首先服务于哪一个请求,以及为哪个对应的请求者分配所请求的部件作为资源。

[0011] 如在此使用的那样,“集成电路芯片”、“微芯片”或“芯片”是一个半导体材料(一般为硅)板(“芯片”)上的一组电子电路。该芯片在制作上可远小于使用单独部件制成的分离电路。这些芯片可用于计算机、移动电话、家用数字电器以及多种其它设备。此处使用的芯片的一个实例为IBM PowerProcessor芯片系列的处理器芯片。

[0012] 如此处使用的那样,芯片“部件”可以是基于任何种类的硬件和/或固件的芯片部件,例如存储器、数据存储器、总线接口、网络控制器、微处理器、GPS模块、密码模块、处理器等。部件可包括或可连接到用于读出部件状态信息(例如,温度、可用内存量等),以及用于将此信息报告给所述服务引擎的传感器元件。所述信息可由所述服务引擎评估并用作执行某些芯片部件管理任务时的输入。部件还可包括或连接到控制单元,这些控制单元可操作以从所述服务引擎接收控制命令,所述控制命令指示所述部件是否应该被启动、关闭或者应该以特定方式执行操作。

[0013] 如在此使用的那样,“任务”是在所述服务引擎的固件和/或硬件元件中实现的数据处理例程。根据所述实施例,可具有用于引导芯片的任务,用于关闭芯片的任务,用于将特定芯片部件分配给芯片内部或芯片外部请求实例的任务,用于切换芯片部件操作模式的任务,维护、错误恢复和部件监视任务等。

[0014] 如在此使用的那样,“寄存器”是作为芯片一部分的数据存储部件。所述寄存器可包括静态或动态RAM。所述寄存器可实现为位于处理器存储层次结构顶部的处理器寄存器,或者实现为主要功能是充当所述服务引擎或另一芯片部件的数据存储的单独寄存器。

[0015] 任务的“优先级”指示开始执行所述任务的迫切性。所述迫切性可完全或至少部分地反映技术要求,例如防止系统不一致的要求,防止损坏芯片硬件的要求和/或防止数据丢失的要求。损坏或数据丢失例如可由芯片过热导致。根据所述实施例,可在对特定任务的请求中或在所请求的任务中例如采取属性值的形式显式地指定所述优先级。备选地,任务的优先级可隐含地给出,例如,根据所请求的任务类型、已提交任务请求的部件类型,或者与所述服务引擎中实现的对应优先化逻辑结合的任务或任务请求的其它某些特征。

[0016] 如此处使用的那样,表达“正常终止任务”包含以可控的方式或突然终止所述任务,其中这种执行状态保证所述终止不会使芯片及其部件处于不一致和/或有害的状态。以可控的方式退出任务暗示该任务本身,或者服务引擎或芯片部件的其它任何例程所执行的某些处理例程保证所述服务引擎管理的芯片部件都不会被所述终止转换为不一致和/或有害的状态。

[0017] 在一方面,本发明涉及一种包括服务引擎的集成电路芯片。所述服务引擎适于经由一个或多个任务管理所述芯片的一个或多个部件。所述服务引擎包括适于执行所述一个或多个任务的执行引擎。所述服务引擎进一步包括存储所述任务中的一个当前执行的任务的当前状态的数据存储器。

[0018] 所述服务引擎进一步包括用于接收执行所述任务中的另一任务的请求的接口。所

述请求可以接收自所述部件中的一个,或者接收自位于所述芯片外部的请求者。所述当前执行的任务具有第一优先级,另一所请求的任务具有另一优先级,该优先级在此被称为“第二优先级”。

[0019] 所述服务引擎包括适于测量接收所述请求与当前之间经过的时间间隔的时钟。

[0020] 所述服务引擎进一步包括适于在以下情况下中断当前执行的任务并触发所请求的任务的执行的控制引擎:

[0021] a) 所述第二优先级高于所述第一优先级,以及

[0022] b1) 所存储的当前状态指示当前任务可正常中断;和/或b2) 所测量的时间间隔超过阈值,由此在b2) 的情况下执行所述中断和所述触发,而不考虑所述当前执行的任务是否可被正常终止或者是否已被正常终止。

[0023] 所述特征可以是有利的,因为它们可允许使用单线程服务引擎,这种引擎对于低优先级和高优先级任务而言制造成本均很低,由此可几乎始终正常终止低优先级任务,并确保最迟在经过时间阈值之后保证启动高优先级任务。

[0024] 所述特征可允许通知所述服务引擎在执行当前的低优先级任务期间,是否以及何时可以安全地中断所述任务以开始执行高优先级任务。

[0025] 根据各实施例,通过接收请求而触发所述时间间隔的测量。

[0026] 如果所述第一优先级高于所述第二优先级,并且在所存储的状态指示所述当前任务在所经过的时长尚未超过所述阈值的同时不可中断的情况下,所述服务引擎可通过所述处理引擎触发所述当前执行的任务的正常终止。成功的正常终止之后,所存储的当前状态值被更新为指示所述当前任务“可中断”的值。只要尚未经过所测量的时间间隔,所述服务引擎便可持续并重复地评估当前状态值,并且在判定所更新的当前状态值等于“可中断”时开始执行所请求的任务。

[0027] 根据各实施例,所述处理引擎仅可操作以一次执行一个任务。换言之,所述处理引擎和对应的服务引擎可以是单线程引擎。这样,具有单线程服务引擎的低制造成本芯片可在保证高优先级任务服务时间方面提供与多线程服务引擎芯片相同的功能,并且不会不正常地终止低优先级任务。

[0028] 根据各实施例,所述芯片仅包括用于管理芯片部件的单服务引擎。这进一步降低了制造成本。

[0029] 根据各实施例,所述请求包括指示所请求的任务优先级的属性值。这可以是有利的,因为请求实例具有影响任务优先级别的装置,这样可影响有关当前执行的任务是否可中断的决策。

[0030] 备选地,每个任务的优先级作为属性值连同其相应任务一起存储在存储部件中。任务的优先级还可根据任务类型给出。在这种情况下,所述服务引擎的决策逻辑使用所请求任务的类型以及当前执行的任务类型来判定是否应该中断当前执行的任务。该实施例可能对于仅需维护少量任务的简单芯片体系结构尤其有利。例如,可能仅支持用于持续监视芯片部件状态参数的第一任务(正常情况下应该一天运行24小时)和用于在所监视的芯片温度超过阈值时关闭部件的第二任务。该第二任务可能仅偶尔被请求以防止有害的系统状态。每当可操作以识别当前执行的任务类型和所请求的任务类型的服务引擎接收到针对第二任务的请求时,将正常终止持续执行的第一任务(如果可能)。

[0031] 根据各实施例,所请求的任务是多个第一任务中的一个,当前执行的任务是多个第二任务中的一个。所述第一任务负责引导、初始化和/或关闭所述芯片的处理器核,负责防止所述处理器核过热,和/或负责管理所述处理器核的供电。所述第二任务负责监视所述部件,具体地说,负责监视所述芯片的处理器核,负责对所述部件计时,负责创建使用统计信息,和/或负责识别所述部件的错误和无效状态。每个第一任务的优先级高于所述第二任务中的任一个。所述特征可以是有利的,因为所述服务引擎可提供一般维护任务,但是可以正常中断它们以开始执行所述第二任务中的任何一个更重要的任务,因为可能需要这些任务以防止损坏所述芯片和/或其部件。

[0032] 根据各实施例,所述处理引擎适于持续监视所述当前执行的任务的当前状态,并且适于相应地更新存储在所述数据存储器中的所述当前状态。例如,所述服务引擎可以在有限时长内执行低优先级监视任务。该任务可以完成其执行并在不被其它任务进行任何中断的情况下正常终止。所述服务引擎将识别该任务已终止并存储当前状态值“可中断”(尽管因为当前任务已经终止,当前根本没有任何任务执行)。根据另一实例,可以请求中等优先级任务。在开始所述中等优先级任务之时,所述服务引擎将当前状态值“不可中断”存储到存储介质,从而阻止所述任务被所请求的具有较高优先级的任务立刻中断(低优先级任务在任何情况下均不会中断中等优先级任务)。所述中等优先级任务的当前状态值“不可中断”只能在以下情况下被高优先级任务中断:其被所述服务引擎正常中断,或者从接收到对应于高优先级任务的请求以来所经过的时长超过时间阈值。

[0033] 作为补充或备选,所述当前执行的任务包括可执行指令,当被执行时,所述可执行指令将所述数据存储器中存储的所述当前状态设为在所述指令中指定的值。

[0034] 例如,特定任务可在不损害任一部件或者不导致任一部件出现不一致的情况下随时中断。在开始执行所述任务时,所述服务引擎将值“可中断”存储到所述数据存储器。

[0035] 或者,例如,任务可以包括在突然中断时可导致芯片损坏的第一子任务,以及包括在突然中断时不导致任何损坏的第二子任务。在开始所述第一子任务时,所述服务引擎存储值“不可中断”。在已经终止所述第一子任务并开始执行所述第二子任务时,所述服务引擎将值“可中断”存储到数据存储器。这样,有关何时将当前状态值更改为“可中断”或“不可中断”的决策应该由某些作为任务一部分的指令做出,和/或由将所述当前执行的任务的优先级与所请求的任务的优先级以及其它因素(例如,已存储的当前状态值、已经过的时间等)相比较的所述服务引擎做出。所述特征可提供高度灵活的细粒度规范,指定是否以及何时可中断特定任务。

[0036] 根据各实施例,中断所述当前执行的任务包括触发所述当前执行的任务的正常终止或者等待所述当前执行的任务自己主动正常终止。在完成所述正常终止之后或期间,将所述当前状态“可中断”存储到所述数据存储器。所述状态值“可中断”指示所述当前执行的任务已经终止或者可以在不导致损坏的情况下立刻终止(即,甚至可由中间中断正常终止)。这使得所述控制引擎能够根据情况b1)触发所请求的任务的执行。例如,在判定所请求的任务的优先级高于当前执行的任务的优先级时,所述处理引擎可跳到当前任务指令内指定所述任务的某些正常退出命令的部分。通过执行所述指令,正常终止当前任务。

[0037] 被设为“可中断”的任务状态值指示当前执行的任务已经被正常终止或者指示其立即终止不会导致芯片或其部件发生错误或不一致。

[0038] 根据各实施例,在所测量的时间间隔超过所述阈值的情况下,所述服务引擎适于将所述数据存储器中存储的所述当前状态设为“可中断”,而不考虑所述当前执行的任务是否已正常终止或是否可正常终止。这使得所述控制引擎能够根据b2) 触发所请求的任务的执行。

[0039] 根据某些实施例,所述请求是所述服务引擎支持的多个请求中的一个,每个所述支持的请求对应于所述任务中的一个。每个所述请求可被分配一个请求特定的阈值。备选地,所述任务特定的阈值可与相应任务的标识符一起存储在所述芯片的数据存储器中。相对于特定于所述请求的阈值而执行所测量的时间间隔是否超过所述阈值的判定。这可允许所述服务引擎执行灵活可变的细粒度任务管理和任务中断逻辑。

[0040] 备选地,可针对所支持的任务中的每一个使用单个阈值。

[0041] 根据各实施例,所述服务引擎支持的任何种类任务都存储在所述集成电路芯片的存储介质上,即,采取汇编指令的形式。

[0042] 根据各实施例,所述执行特定任务的请求由所述部件中的一个或片上部件管理单元产生。所述请求可以包括其请求特定的阈值。所述请求还可以接收自位于所述芯片外部的请求者,例如接收自外部硬盘控制器等。

[0043] 所述服务引擎可被实现为片上微控制器。所述部件可包括硬件和/或固件或它们的组合。所述任务是所述芯片上存储的计算机可解释指令,例如,以汇编程序语言或其它可轻松转换为操作码的语言编写的程序指令。所述服务部件的指令集可通过极为高效地执行所有支持的任务(具体地说,高优先级任务)的方式改编。所述特征可以是有利的,因为硬件实现的方法的执行速度远快于类似基于软件的方法。此外,基于软件的程序逻辑的执行可能因为其它基于软件的程序逻辑片段失败、执行操作系统失败等原因而停止。在根据上述任一实施例的服务引擎的基于硬件和/或固件实现中不可能发生此情况。

[0044] 根据各实施例,包括所述当前状态的所述数据存储是以单个位值的形式存储所述当前状态的寄存器。所存储的当前状态为“可中断”或“不可中断”。所述特征可以是有利的,因为寄存器和状态的单个位表示减小所用的存储空间,同时缩短读取和评估所存储的当前状态值的时间。

[0045] 根据各实施例,所存储的当前状态为“可中断”或“不可中断”或“可有条件地中断”。所存储的状态值为“可有条件地中断”的当前执行的任务的中断使得所述控制引擎能够在所请求的任务或所述芯片的部件满足额外条件的情况下中断所述当前执行的任务。此类额外条件可包括所述服务引擎的操作模式、随所述请求接收的附件参数、当前时间和日期、用户ID以及根据用户与云服务提供商之间合约的服务水平信息等。这样,例如当在云基础设施内使用所述芯片及其处理能力时,与特定用户的服务水平协议和/或特定时间和日期的预期处理负荷可能影响状态为“可有条件地中断”的特定任务是否为了满足所请求的任务而被中断。

[0046] 例如,第一用户与云服务提供商之间的处理能力服务水平协议所涵盖的服务质量和/或服务数量可能小于第二用户与所述提供商之间的服务水平协议,“可有条件地中断”状态可指示可以为了满足第二用户所请求的任务而中断所述第一用户的任务。在某些用例情景中,所述芯片的特定部件或资源至少可被临时分配给不同的用户(例如,取决于服务水平协议),并且有关用户ID的信息可被包括在所述部件或资源提交给所述服务引擎的任何

任务请求内。这样,云服务提供商可利用能够以多种不同的值表示和存储的状态所提供的细粒度中断控制机制。

[0047] 根据各实施例,所述服务引擎包括用户接口,例如图形界面或命令行接口,从而允许用户配置时间阈值和/或任务的优先级。

[0048] 在另一方面,本发明涉及一种包括根据上述实施例之一的多个集成电路芯片的组件。每个芯片包括一个服务引擎以及一个或多个要由所述服务引擎管理的部件。每个芯片的部件包括一个处理器核。所述核由其相应服务引擎管理。例如,可提供包括多个芯片的大型机,其中所述芯片具有各自的处理器核。每个所述芯片的处理器核可作为所述芯片的部件由相应的片上服务引擎管理。

[0049] 在另一方面,本发明涉及一种用于经由一个或多个任务管理集成电路芯片的一个或多个部件的计算机实现的方法。所述芯片包括服务引擎。所述方法由所述服务引擎执行并包括:

[0050] -执行一个或多个任务中的一个当前任务;

[0051] -在所述芯片的数据存储器中存储当前执行的任务的当前状态;

[0052] -从所述部件之一接收执行所述任务中的另一任务的请求,所述当前执行的任务具有第一优先级,所述另一任务具有第二优先级;

[0053] -测量接收所述请求与当前时间之间已经过的时间间隔;

[0054] -在以下情况下中断所述当前执行的任务并触发所请求的任务的执行:

[0055] a) 所述第一优先级高于所述第二优先级,以及

[0056] b1) 所存储的当前状态指示当前任务可正常中断;和/或b2) 所测量的时间间隔超过阈值,由此在b2) 的情况下执行所述中断和所述触发,而不考虑所述当前执行的任务是否被正常终止。

附图说明

[0057] 通过结合附图阅读下面对本发明实施例的更多具体描述,可以更好地理解本发明的上述及其它内容、特征和优点,这些附图是:

[0058] 图1是包括服务引擎和多个部件的芯片的框图,

[0059] 图2是包括多个芯片的组件,

[0060] 图3是由服务引擎或其部件执行的方法的流程图。

具体实施方式

[0061] 图1公开包括多个部件114、116、118和120的芯片102。所述部件例如可以是存储器或存储部件、总线接口、中央处理单元、网络控制器等。该芯片进一步包括充当管理芯片部件114-120的服务引擎104的微控制器。该服务引擎包括单线程处理引擎106,该处理引擎可操作以一次执行一个任务。

[0062] 根据图1,处理引擎当前执行任务T4。服务引擎支持的每个任务T1-T5可实现为在服务引擎的固件和/或硬件中体现的计算机可执行指令,例如,采取汇编指令的形式。每个任务具有优先级P1-P5。可根据任务类型,或者例如根据结合相应任务存储的属性值或根据对特定任务的请求中包含的属性值而给出任务的优先级。服务引擎包括存储当前执行的任

务T4的当前状态的寄存器112。该状态例如可以是“可中断”、“不可中断”或者“可有条件地中断”。

[0063] 服务引擎104包括接口122,其用于从芯片部件114-120中的一个或从芯片外部的部件接收请求R。根据图1,部件116经由接口122将执行任务T1的请求发送到服务引擎。根据所述请求的结果,服务引擎的时钟108开始监视自接收以来已经过的时长。

[0064] 服务引擎包括优先级装置10,该装置可操作以将当前执行的任务T4的优先级与所请求的任务T1的优先级P1相比较。服务引擎包括寄存器112或在操作上与其相连,寄存器112存储当前执行的任务的当前状态值CS。当从部件116接收到请求R时,优先级装置110将所请求的任务T1的优先级P1与当前执行的任务T4的优先级P4相比较。此外,服务引擎评估寄存器中存储的当前状态值CS。

[0065] 在优先级P1超过优先级P4的情况下以及在CS值为“可中断”的情况下,服务引擎触发处理引擎以立即开始执行所请求的任务T1。在当前状态值CS应该指示当前任务T4“不可中断”的情况下,服务引擎将命令发送到处理引擎以触发任务T4的正常终止。备选地,不是发送所述触发命令,而是服务引擎只需等待当前执行的任务自己主动终止,从而将当前状态值CS设为“可中断”。只要当前执行的任务将CS值设为“可中断”,便立即停止任务T4并开始所请求的任务T1的执行。此类中断在此也称为“正常中断”,很明显,任务T4下方的程序指令仅将当前状态值设为“可中断”,前提是根据已执行以及即将执行的指令的特性,确保在该特定执行点上突然终止所述任务不会对系统造成任何损害。一旦任务T4正常终止,CS值便被设为“可中断”并执行所请求的任务T1。仅在任务T4的正常终止未在预定时长(从接收到请求R开始测量)内完成的情况下,服务引擎才会强制终止任务T4以确保执行高优先级任务T1。

[0066] 例如,当前执行的任务T4可以是24/后台任务,即,应该一周运行7天,一天持续运行24小时的任务。后台任务具有低优先级并且用于持续监视芯片部件的状态,例如,用于计算使用统计信息。如果可能,任务T4应被持续执行并且仅应针对紧急错误纠正或T1之类的维护任务而被中断。所请求的任务T1可以是需要在短时间间隔内执行以防损坏芯片的电源管理任务。

[0067] 为确保高优先级任务T1可尽快中断低优先级任务T4并且为确保T4在T1终止之后立即继续执行,部件120(可以是监视微控制器)可持续将执行任务T4的请求发送到服务引擎。此类请求不会中断当前执行的任务T1,因为对低优先级任务T4的请求的优先级不高于当前执行的任务T1的优先级。但是,只要T1完成执行并且服务引擎接收到对任务T4的请求,服务引擎便重新开始执行T4。

[0068] 备选地,为确保高优先级任务T1可尽快中断低优先级任务T4,当前执行的任务T4可包括在所述任务终止时立即触发执行另一任务(例如,在请求R中指定的当前请求的任务T1或在任务T4的已执行指令中指定的另一任务)的程序指令。所述方法可具有以下优点:即,不需要重复地执行开始执行T4的请求,从而减小服务引擎的计算负荷。

[0069] 优选地,任务T1包括某些用于在开始T1时将CS值设为“不可中断”,并且在终止T1时将CS值设为“可中断”的指令。这样,只要高优先级任务T1完成执行,从部件120接收对低优先级任务T4的请求的服务引擎便开始监视任务T4。根据某些实施例,服务引擎可包括进一步的程序逻辑和用于临时存储已中断的低优先级任务的中间结果的存储器,并且在中断

的高优先级任务完成执行之后使用所述中间结果恢复低优先级任务。

[0070] 任务的优先级可根据属性显式指定或者可根据任务类型隐含地提供。任务类型可由发送对应任务请求的部件类型指示。例如,部件116可以是电源管理控制器并且所述部件提交的任何请求(包括请求R)可被服务引擎识别为具有高优先级的电源管理请求。

[0071] 图2示出图1中分别更详细描述包括多个芯片102.1-102.4的组件202。每个芯片包括一个或多个部件以及单线程服务引擎104.1-104.4。此类组件可构成多核处理器大型机,其中每个“核”对应于芯片之一。

[0072] 图3示出要由根据本发明的一个实施例的服务引擎或其某些部件执行的方法。在步骤302,服务引擎104执行一个或多个任务T1-T5中的一个当前任务T4。在步骤304,服务引擎将当前执行的任务的当前状态存储到芯片102的数据存储器112。在步骤306,服务引擎从多个部件116中的一个接收执行特定任务T1的请求R。当前执行的当前任务T4具有第一优先级P4,所请求的任务T1具有第二优先级P1。在接收到请求R时,时钟108开始在步骤308持续测量自接收到请求R以来已经过的时间。借助优先级装置110,服务引擎在步骤310判定第一优先级P4是否高于第二优先级P1。如果结果为是,则继续像步骤318所述的那样执行当前任务T4并且服务引擎等待进一步的请求。

[0073] 在优先级P1高于P4的情况下,服务引擎在步骤312(路径b1)计算所存储的当前状态CS是否指示当前执行的当前任务T4可正常中断。如果可正常中断,则服务引擎突然中断T4(在T4仍运行的情况下)并在步骤316.1开始执行T1。

[0074] 如果所存储的状态指示当前任务不可中断,则服务引擎在步骤317触发当前执行的当前任务T4的正常终止或者等待当前执行的当前任务自己主动将当前状态值CS设为“可中断”,然后触发立即(正常)终止T4并触发执行任务T1(由此,服务引擎不会不确定地“等待”,如果自接收到针对更高优先级任务的请求R以来经过的时间超过阈值,则不正常地终止当前执行的当前任务)。

[0075] 在任何情况下,当正常终止任务T4时,所存储的T4的当前状态被更新为“不可中断”,并且针对更新后的值重复步骤312。此时,服务引擎将判定所存储的当前状态指示当前任务可正常中断并且将执行步骤316.1,该步骤包括中断当前执行的当前任务(如果有)并开始执行所请求的任务。

[0076] 在判定优先级P1超过P4的情况下,服务引擎将(独立于路径b1)重复地执行决策步骤314(路径b2)以判定所测量的自接收到请求R以来的时间间隔是否超过阈值。如果结果为是,服务引擎将中断当前执行的当前任务T4并触发执行所请求的任务T1,而不考虑当前执行的当前任务是否已被正常终止。只需忽略所存储的当前状态值CS或强制将所述值设为“可中断”并执行步骤312便可实现上述目的。如果结果为否,则服务引擎的时钟将继续执行步骤308。

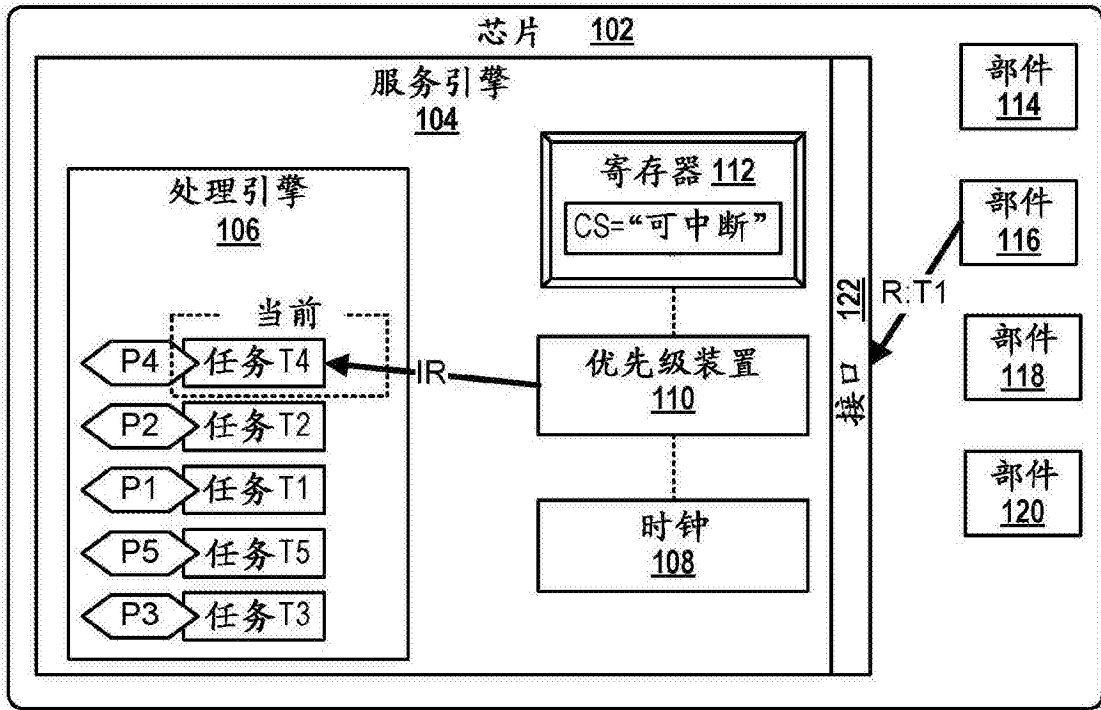


图1

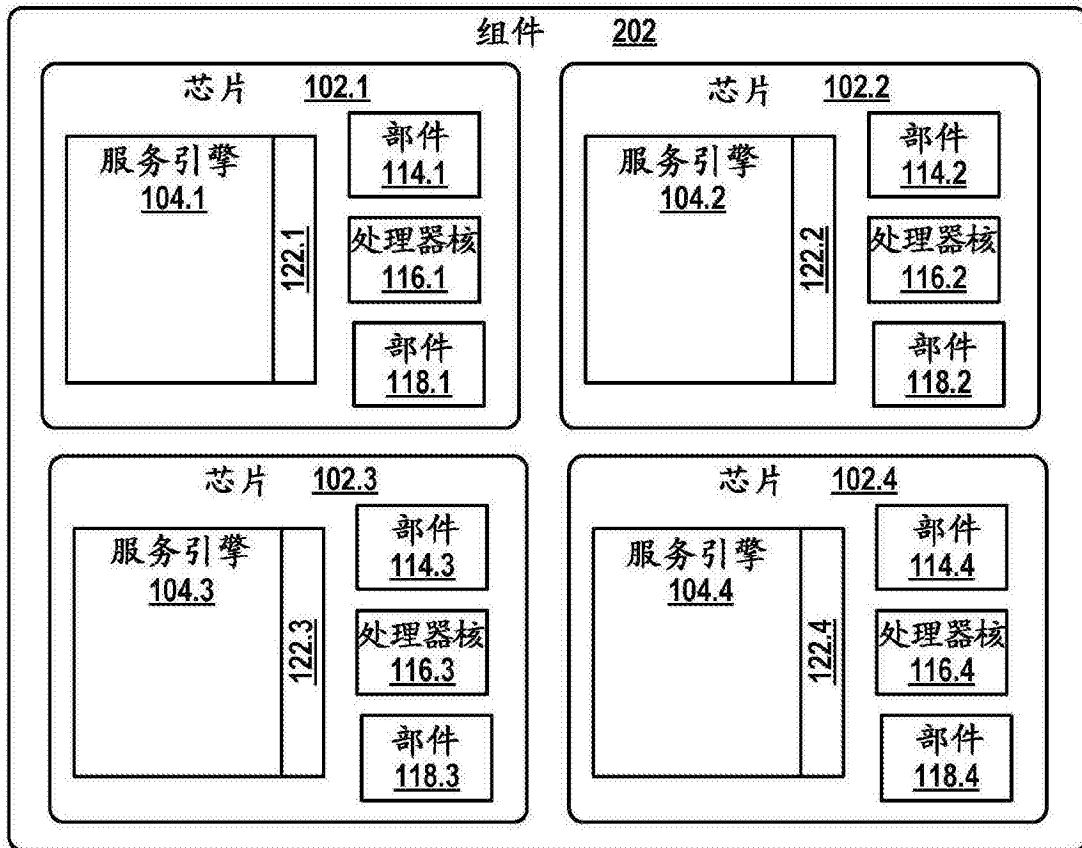


图2

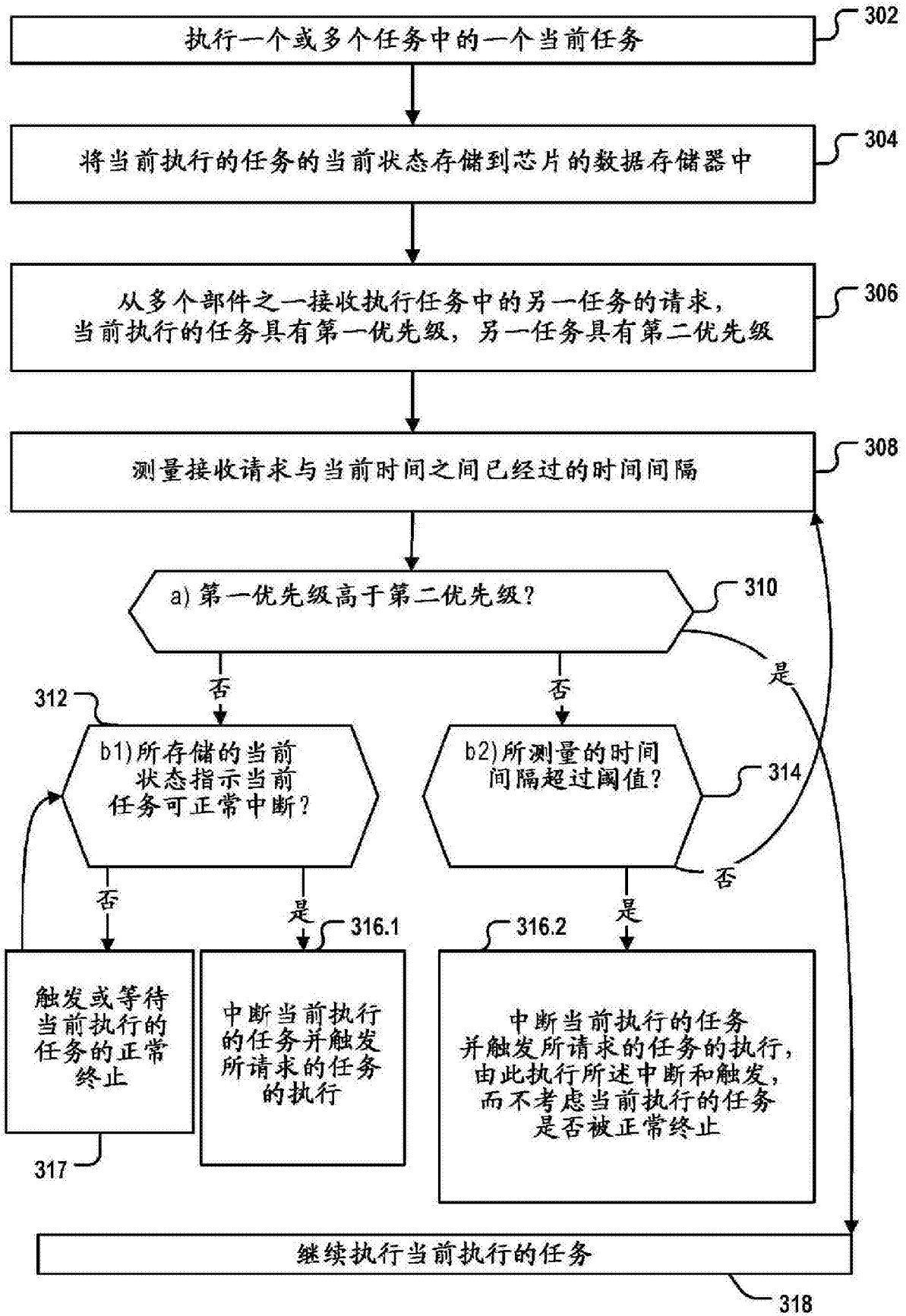


图3