



[12] 发明专利申请公布说明书

[21] 申请号 200580009824.4

[43] 公开日 2007 年 3 月 28 日

[11] 公开号 CN 1938671A

[22] 申请日 2005.3.25

[21] 申请号 200580009824.4

[30] 优先权

[32] 2004.3.30 [33] US [31] 10/814,425

[86] 国际申请 PCT/US2005/010163 2005.3.25

[87] 国际公布 WO2005/098579 英 2005.10.20

[85] 进入国家阶段日期 2006.9.26

[71] 申请人 英特尔公司

地址 美国加利福尼亚州

[72] 发明人 E·C·萨姆森

[74] 专利代理机构 中国专利代理(香港)有限公司

代理人 杨凯王勇

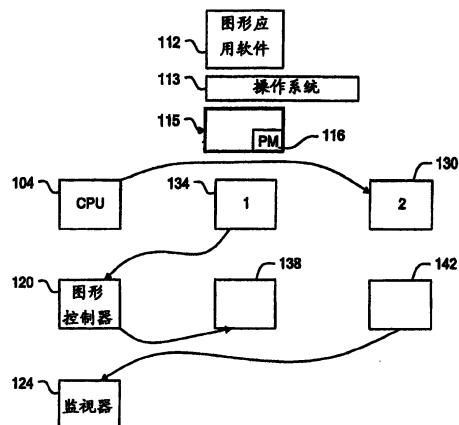
权利要求书 3 页 说明书 10 页 附图 3 页

[54] 发明名称

通过请求调整处理器的工作点来管理功耗

[57] 摘要

为处理器提供了具有实时需求的工作量。根据该实时需求的期限裕度，为该处理器设置了处理器时钟频率的要求。也描述了其他实施例，并要求保护这些实施例。



1.一种管理电子系统中的功耗的方法，包括：

为所述系统的第一处理器提供待执行的第一任务；

为所述系统的第二处理器提供待执行的第二任务，其中，所述第二任务的执行将使用所述第一任务的执行结果；

根据所述第二任务的完成时刻与其期限的时间间隔，请求调整所述第一和第二处理器之一的工作点，以更好地管理所述电子系统中的功耗。

2.如权利要求1所述的方法，其中，如果所述时间表明满足期限要求，则所请求的调整是降低处理器时钟频率。

3.如权利要求1所述的方法，其中，如果所述时间表明不满足期限要求，则所请求的调整是增加处理器时钟频率。

4.如权利要求1所述的方法，其中，所述第一和第二任务与通过所述系统描述和渲染图像有关，该方法还包括根据用于显示图像的目标帧速率来计算所述第一和第二任务的期限。

5.如权利要求1所述的方法，还包括测量所述第一处理器完成所述第一任务所需时间与所述第二处理器完成所述第二任务所需的时间，其中，所请求的调整建立在所述测量的基础上。

6.如权利要求4所述的方法，还包括：所述第二任务一旦完成，所述第二处理器就向所述第一处理器发出中断信号。

7.如权利要求6所述的方法，其中，所述第一和第二处理器根据三缓冲图形处理执行它们各自的任务。

8.一种方法包括：

为处理器提供具有实时需求的工作负荷；并根据所述实时需求的期限裕度，为所述处理器设置处理器时钟频率要求。

9.如权利要求8所述的方法，其中，所述实时需求是用于显示图像帧的目标帧速率。

10.如权利要求 9 所述的方法，其中，所述裕度是(i)所述处理器识别图像中的一个或多个图形表面的完成与(ii)渲染所述一个或多个图形表面的开始之间的时间测量值。

11.如权利要求 9 所述的方法，其中，根据(i)所述处理器识别图像的一个或多个图形表面所需的时间估计值和(ii)渲染所述一个或多个图形表面所需的时间估计值以及(iii)所述目标帧速率来计算所述裕度。

12.如权利要求 9 所述的方法，其中，所述裕度是(i)所述处理器渲染图像的完成与(ii)显示图像的开始之间的时间测量值。

13.如权利要求 9 所述的方法，其中，根据(i)所述处理器渲染图像所需的时间估计值与(ii)所述目标帧速率来计算所述裕度。

14.一种系统包括：

中央处理器(CPU);

与所述 CPU 相连的图形控制器；

与所述图形控制器相连的监视器；

包含指令的存储器，当所述指令由所述 CPU 执行时，(i)识别图像中各个待渲染表面的模型，其中，所述图像将由所述图形控制器进行渲染，然后按照目标帧速率显示在所述监视器上，和(ii)根据所述目标帧速率的期限裕度来规定所述 CPU 和图形控制器之一的工作点

15.如权利要求 14 所述的系统，其中，所述工作点是一个值，该值表示(i)时钟频率、(ii)相对于所述时钟频率的偏移量和(iii)增大或减小时钟频率的方向这些值之一。

16.如权利要求 14 所述的系统，其中，当由所述 CPU 执行时，所述指令根据所述图形控制器完成渲染任务的时刻与该任务的期限之间的这段时间来规定所述图形控制器的工作点

17.如权利要求 16 所述的系统，其中，当由所述 CPU 执行时，所述指令根据所述 CPU 完成规定图像中的各图形对象的数学模型所

需时间和所述图形控制器渲染所述图像所需时间来规定所述 CPU 的工作点。

18.如权利要求 14 所述的系统，其中，当由所述 CPU 执行时，所述指令根据所述系统中记录的值来规定所述 CPU 和图形控制器的工作点，且所述值表示所述图形控制器完全渲染图像实际经历的时间和所述处理器完成识别所述图像中的各图形对象的模型实际经历的时间。

19.如权利要求 14 所述的系统，其中，所述规定的工作点是所述 CPU 的高性能模式和低性能模式这两者之一。

20.如权利要求 19 所述的系统，其中，与所述低性能模式相比，在所述高性能模式下，所述 CPU 在较高的时钟频率和较高的电源电压下工作。

21.一种产品包括：

具有数据的机器可读媒体，当处理器访问所述数据时，所述媒体请求用于目标处理器的工作点要求，以(i)在所述目标处理器处理工作负荷时，减少所述目标处理器的功耗和(ii)满足所述工作负荷中的多个任务的多个完成期限，其中，所述请求的要求建立在任务完成时刻与其期限之间所经历时间的基础上。

22.如权利要求 21 所述的产品，其中，所述数据用来将所请求的工作点要求定为所述处理器的工作时钟频率的降低。

23.如权利要求 21 所述的产品，其中，所述数据是计算机操作系统程序的一部分。

24.如权利要求 21 所述的产品，其中，所述数据是计算机系统驱动器程序的一部分。

通过请求调整处理器的工作点来管理功耗

技术领域

本发明的一个实施例涉及电子系统的功耗管理，而这是根据实时需求的期限裕度(deadline margin)、通过调整系统内的处理器的工作点来实现的。也描述了其他实施例，并要求保护这些实施例。

背景技术

电子计算系统中的功耗管理、尤其是诸如由电池供电(因此能量供应有限)的笔记本电脑等便携式系统的功耗管理已成为热点问题。随着便携式系统继续扩充其功能和消耗更多功率，人们开发了各种不同技术，以更好地管理它们的功耗，例如，尽可能减少它们的功耗。

计算机系统的由时钟控制的处理器元件(如中央处理器(CPU)和图形处理器(或控制器))的制造商已经在这些元件内建立某种机构，以允许根据命令调整它们的性能和工作点。例如，在系统电源从线路功率变为电池时，加州圣克拉拉的英特尔公司开发的 SpeedStep 技术允许系统中运行的软件根据 CPU 处理器的利用率等因素动态地请求改变 CPU 的时钟频率和工作电压。在这种算法中，通过向现有或实际频率应用比例系数确定了新的处理器时钟频率。比例系数是根据处理器最近的“使用”计算出来的。例如，若处理器仅在 25% 的时间内处于忙状态(处理给定的工作负荷)，则上述频率将逐渐下降到其当前值的约 25%。这种频率降低会减少功耗，而不会影响工作负荷的按时完成。当然，这是以工作负荷未大幅增加为前提的。

附图说明

在附图的各图中，以举例而非限制的方式示出了本发明的实施例，其中，各图中相同的附图标记表示相同的元件。应当指出，在本公开中，所提到的本发明的“某一实施例”不一定指同一个实施例，实际上，它们意味着至少一个。

图 1 示出了具有双缓冲显示功能的计算机系统的概念图。

图 2 示出了可用于功耗管理的双缓冲渲染的时序图。

图 3 示出了具有三缓冲显示功能的计算机系统的概念图。

图 4 示出了可用于功耗管理的三缓冲渲染的时序图。

图 5 示出了用于在电子系统中管理功耗的方法的流程图。

具体实施方式

在许多情况下，提供给计算机系统的处理器的工作负荷都具有实时需求。例如，操作系统程序可能规定了在系统的监视器上显示图像(有时称为图像帧)的视频帧速度的最小值。在这种情况下，这便决定了：在作出每个工作点转移决定时，如果没有同时考虑到 CPU 和图形控制器，便有可能导致帧丢失(因为处理器时钟频率不够高)。此外，在应用上述基于使用的方法(为 CPU 和/或图形控制器设置处理器时钟频率要求)时，对处理器时钟频率的不正确调整同样可导致帧丢失。

根据本发明的一个实施例，处理器时钟频率要求是根据实时需求的期限裕度来设定的。可以将期限裕度宽泛地定义为从任务完成到任务期限的这段时间。换言之，本发明的这个实施例不确定处理器到现在为止有多忙(何时更新工作点要求)，而是考虑处理器的给定任务的完成时刻到任务期限有多近。该处理器可以是计算机系统的 CPU 或图形控制器，而该实时需求可以是系统中用于显示视频的目标帧速度。这种方法可用于更好的管理系统功耗，同时提供足够的计算能力来防止因图像处理方案造成的帧丢失。其他的计算应用(例如涉及两个或更多处理器(如 CPU 和图形控制器部分)的顺序或级联

处理方案)也可以从这种方法中获益。

现在转到图 1, 其中 示出了计算机系统中的双缓冲图形处理方案的概念框图。该实例系统具有中央处理器(CPU)104, 如一个或多个加州圣克拉拉的 Intel 公司出品的 PENTIUM 处理器。CPU104 所用的数据可以存放在存储器 108 中, 该存储器可以由不同类型的机器可读媒体如固态存储器(例如: 动态随机存取存储器、静态随机存取存储器、或其他类型的固态易失性或非易失性媒体)组成, 在一些实施例中, 它还可包括磁转盘驱动器、光盘驱动器或其他大容量存储设备。存储器 108 具有几个部分, 包括一个存储图形应用软件 112 的部分, 和另一个具有功率管理(PM)软件 116 的部分。PM 软件 116 可以是驱动器程序 115 或操作系统 113 的一部分。这些程序由 CPU104 和/或图形控制器 120 访问并执行。可以将图形控制器封装为系统接口集成电路(也被称为系统芯片)的一部分。或者, 图形控制器 120 可以独立封装, 例如, 可以将该控制器封装在印刷线路板形式的图形适配卡上, 且该图形适配卡通过系统芯片组(未示出)与 CPU104 进行通讯。

在图 1 的实例中, 由图形应用软件 112 引入工作负荷, 且 在图形控制器 120 的辅助下由 CPU 来执行上述图形应用软件。该工作负荷可包括描述图像中的表面(如识别一个或多个图形对象或图元), 在图形控制器 120 的帮助下渲染该表面, 然后提供用于在监视器 124 上显示的所得表面。图 1 描述了双缓冲图像处理方案的过程的实例。在这样的方案中, 分配了两个存储部分 130, 134(或缓冲器), 每个存储部分存储由图形应用软件 112 描述的各自的图像。相比之下, 图 3 和图 4 表示了具有三个这样的存储部分的三缓冲方案。

回到图 1, 在 CPU104 完成定义第一个表面和在第一部分 134 中存储数据的任务后, 可转向下一个任务。该任务是描述第二个表面, 并将其数据存放在第二部分 130 中。同时, 图形控制器 120 可以访问第一部分 134, 以渲染上述图像的表面。然后, 将经过渲染的图像

写入第三部分 138。同时，第四部分 142 中存在前一幅图像的经过渲染的表面，因此可以将其传送到监视器 124 中，同时进行上述的其他操作。注意，一旦将表面送至监视器 124，则需要将该图像以某一速度(如大于 60Hz 的速率)在监视器 124 中刷新。

虽然上述用于图形处理的顺序方法可以允许实现相对较高的帧速度(即每单位时间在监视器 124 上显示数目相对较多的帧)，但是它也消耗了系统总功率的很大一部分。为显示场景中的运动，对于 dx7 画质的计算机动画图像，预计最小帧速度为 10 到 25 帧/秒，而对于自然场景，则在 30 和 60 帧/秒之间。然而，也可采用其他帧速度。本文提供的功率管理(PM)软件 116 用于在 CPU104 和/或图形控制器 120 执行任务时，通过基于实时需求的期限裕度请求调整它们的工作点来更好地管理系统中的功耗。可以将该实时需求规定为操作系统 113 中的目标帧速度，这样，在操作系统上运行的所有应用程序均需一直满足目标帧速度的要求(例如，不允许帧丢失)。或者，根据应用软件 112 的决定或命令，可以对目标帧速度进行调整。

PM 软件 116 可以由 CPU104、图形控制器 120 或系统的其他处理器(未示出)来执行，以基于实时需求的期限裕度来改变处理器时钟频率要求。另一方法是采用硬接线的逻辑电路(可或不可编程)，该电路能求出工作负荷的实时需求和确定期限裕度(相应的，由此设置处理器时钟频率要求)。可以在系统芯片组、封装的 CPU 或系统的其他 IC 器件中实施这种硬接线的逻辑电路。

现在转到图 2，该图示出了图 1 的系统中的双缓冲渲染的示范的时序图。此处该图也用来说明如何根据本发明的一个实施例和使用期限裕度来管理功耗。该时序图也说明了由错误确定的工作点转移造成的错过期限的问题。为了更好的理解时序图，假设功率管理技术的高级目标是在处理器处理给定的工作负荷的同时，重复进行工作点转移(如设置处理器时钟频率要求)，使得实际的期限裕度(当处理器处理工作负荷时)接近于零。应当争取在不错过任何期限的同时

实现这一点。

现在参阅图 2 所示波形，第一个波形表示由系统的 CPU 部分完成的工作，所以无论何时断言(assert)该图形，CPU 均会执行与给定的工作负荷相关的某个任务。在该实例中，数字 1、2、3...是指准备或描述待渲染的各个表面的任务。第二个波形表示由图形控制器部分执行的任务。在顺序处理中，图形控制器部分的任务 1 使用 CPU 部分完成的任务 1 的执行结果，图形控制器部分的任务 2 使用 CPU 部分完成的任务 2 的执行结果，等等。例如，图形控制器部分的每个任务可包括渲染已由 CPU 部分进行描述的图像的单个表面的一部分或其整体。另外，可以将每个任务定义为渲染单幅图像的所有表面(在此之前，作为另一任务，这些表面由不同的处理器来进行计算或定义)。

图 2 的第三行表示时间上的一些点，在这些点上断言了垂直消隐(Vsync)信号，这表明，应当为把新的图像实际送至监视器而做好准备。因此，对第一幅图像而言，时段 212 表示图形控制部分完成图像的时刻和该任务的由 Vsync1 表示的期限之间的时间间隔，而所述期限是应当将第一幅图像的表面送至监视器来进行显示的那个时点。类似的，对第二幅图像而言，时段 216 表示图形控制部分完成任务的时刻距离该任务的期限(由 Vsync2 表示)的接近程度。

如上所述，图 2 的时序图也说明了错过期限的现象。注意，大约在图形控制器部分完成它的任务 2 时，是如何发生工作点转移的。而上述工作点转移是由功率管理算法造成的，该算法确定，需要进行工作点转移，即减小图形控制器部分或 CPU 部分的处理器时钟频率。上述工作点转移导致 CPU 部分执行任务 3 的时间远多于执行任务 1 和 2 的时间。此外，图形控制部分执行任务 3 的时间也比其之前执行该任务的时间长。遗憾的是，该工作点的改变过于剧烈，以至错过了由 Vsync3 表示的期限，因为，从时序图可以看出，图形控制部分完成任务 3 的时刻滞后所述期限一个时段 218。

为避免图 2 所示的错过期限的情况，本发明的一个实施例基于任务的完成时刻与期限的时间间隔来请求调整 CPU 部分和图形控制器部分之一或这两者的工作点。在这种情况下，对于实时需求为目标帧速度(由 Vsync_{1,2,...}之间的时间间隔所定义)的情形，时段 212 和 216 是任务 1 和 2 满足期限裕度要求的实例，而时隙 218 表示任务 3 的实质上为“负”的期限裕度。注意，所请求的调整或工作点转移可建立在对 CPU 部分和/或图形控制器部分完成它们的各自任务所需时间的实际测量的基础上。或者，所请求的调整可建立在对时段 212、216 和 218 的一个或多个测量的基础上。

对处理器工作点的调整请求可能存在不同类型。例如，所规定的工作点可以是值或处理器的工作模式，该值或模式具有预定的时钟频率，且处理器核在该频率上工作。或者，所规定的工作点可以是相对于时钟频率的偏移量，该偏移量的正负取决于希望执行时间增加还是减少。而另一种方案是将工作点规定为用预定的频率增量来增加或减少时钟频率的方向。

应当注意，不仅可以根据实时需求的期限裕度来改变工作点，也可以根据工作负荷的预期变化(如果存在)来改变工作点。例如，在上述图形处理方案中，在每个帧的基础上，工作负荷可能发生意想不到的改变，使得难以提前预测需要多少时间来定义图像的表面和对其进行渲染。然而，可以用之前的帧来进行良好的预测(进行大规模的主题切换(如突然改变场景)时除外)。因此，对图形处理实施例中的处理器而言，可以用下面的公式基于当前的工作点来规定“下一个”工作点要求。

$$f(n)=f(n-1)*(1-x(n)/T)*S$$

如果 $f(n)>f_{max}$ ，则 $f(n)=f_{max}$

如果 $f(n)<f_{min}$ ，则 $f(n)=f_{min}$

其中， $f(n)$ 是工作点转移后的工作时钟频率， $f(n-1)$ 是当前或之前的工作频率， $x(n)$ 是以秒为单位的期限裕度(用于实时需求)， T 是完成任

务的总的时间， S 是安全因子(使工作点转移不那么激烈的系数)。因此，上述公式假设可以用之前帧的工作频率来较好地预测下一帧。

再次回忆起总体目标是在处理器处理工作负荷时，重复地设置处理器时钟频率要求，使得实际的期限裕度(当处理器处理工作负荷时)接近于零。在理想状况下，应该在不错过任何期限的情况下完成这个目标，尽管在实际中不能总是做到这一点。因此，该目标可以是为处理器有效地选择可能的最低工作点，但依然满足所述要求。可以预期的是，该目标更容易实现，尤其对存在多于一个处理器的顺序处理实施例而言，通过基于实时需求的期限裕度而非仅基于处理器的繁忙程度来确定处理器的工作点转移更容易实现这一目标。

如上所述，对图形处理实施例中的 CPU 部分而言，期限裕度可以是 CPU 识别图像中的一个或多个图形表面的完成时刻和渲染这些表面的起始时刻之间的时间间隔的实际测量值。或者，可以根据 CPU 识别图形表面所需的时间估计值得到该裕度，此处，可以通过仿真而非对工作中的产品进行实际测量得到该估计值。类似的，对图形控制器部分而言，期限裕度可以是图形控制器部分渲染图像的完成时刻和监视器显示图像的开始时刻之间的时间间隔测量值。可通过设计渲染引擎来将时间戳值(例如，从自由运行的硬件计数器中获得该值)写入主内存或系统芯片组中的某一位置来进行这些测量，且上述时间戳值表示完成帧渲染的时点。操作系统软件或其他软件(如图形驱动器)可以计算 CPU 或图形控制器部分执行它们各自任务所需(或实际所花费)的时间。

现在参阅图 3，其中 示出了具有三缓冲显示功能的计算机系统的概念图。在该实例中，当 CPU 写入其任务 1 的结果时，图形控制器可以从不同的缓冲部分读取之前完成的任务的结果。分配了第三个缓冲部分，以便让图形控制器和 CPU 更灵活的执行它们各自的任务。这可以进一步减少每个处理器因等待可用的缓冲部分所花的空闲时间。如图 4 的时序图所示，在三缓冲方案中，尽管不正确的执

行工作点转移仍将导致错过期限，但是三缓冲在开始时允许有更大的期限裕度，因而，它在每个转移点处降低处理器时钟频率时所面临的要求便更为松散。

在上述实施例中，需要提供一些机制，以便使第二处理器(如图形控制器)在完成读取缓冲器及相关任务后，第一处理器(如 CPU)便得知可重新使用的缓冲器已可用。一项这样的技术是提供一种软件，该软件询问 图形控制器，以得到特定任务的完成时刻。然而，与低功率模式(如睡眠 模式)相比，询问时 CPU 通常处于正常或主动的工作模式。因此，为进一步减少功耗，本发明的一个实施例为第二处理器配备了这样的功能：一旦第一处理器完成了给定的任务(这样主处理器使得有缓存可用)，第二处理器(如图形控制器)便中断第一处理器(从而因第一处理器的缘故，便使得缓冲器可用)。这将有助于在 CPU 执行其任务的各时段之间使 CPU 保持在较低的功耗模式。因此，参阅图 2，中断发信(interrupt signaling)功能可以让 CPU 部分在 220、224、226、228 等时段内处于低功耗模式(如睡眠 模式)。另外，或者除该改变以外，平均而言，CPU 在执行一系列图形任务时将工作于较低的时钟频率，因为已大大减轻了其查询职责。

中断发信实施例可用于上述的双缓冲和三缓冲两种情形。然而，在一些情况下，CPU 可能会引入相对较长的中断处理等待时间，该时间是从收到中断信号到实际处理中断(通过执行中断处理例程来改变任务或内容)的一段时间。因此，应当提前为 CPU 安排任务，以便为图形控制器提供足够的任务，使得它在等待 CPU 处理其中断请求时不发生闲置。例如，如果图形控制器完成某项任务而使得某 特定的缓冲器可以为 CPU 所用，则在 CPU 使得另一缓冲器可用之前，图形控制器不需等待过长时间便可进行其下一项任务。在该实例中，三缓冲方案可能比双缓冲方案更为有效。

现在转到图 5，在其中，用流程图示出了一个用于管理电子系统的功耗的更为通用的方法。这是一个顺序处理方案的实例，它包括

至少第一和第二处理器。操作以向第一处理器提供用于执行的第一任务(框 504)开始。也向该系统的第二处理器提供了另一用于执行的任务，其中，在执行任务时，第二处理器将使用第一任务的执行结果(框 508)。例如，这些任务可能与描述和渲染视频序列中的图像有关。第一处理器的任务可以是识别或描述待渲染的一个或多个表面的模型，而第二处理器的任务将包括实际渲染所定义的表面和创建显示图像。

注意，不必基于整个或完整的表面来定义每个图形任务的边界。例如，与完全定义整个表面然后渲染整个表面不同，可以部分地定义表面然后部分地渲染表面，在其中，定义某一部分，然后在第一和第二处理器定义和渲染随后的部分之前，渲染上述已经过定义的部分。也可以在不同于图形处理任务的任务中应用这种方法。

回到图 5，该流程图包括请求调整第一和/或第二处理器的工作点(框 512)的进一步操作。而这是为了更好的管理电子系统的功耗，即更有效地使用第一和第二处理器来执行它们的任务。与仅基于第一和/或第二处理器的利用率来进行上述调整不同，本发明的一个实施例用于至少部分地基于第二项任务的完成时刻和其期限的时间间隔来计算或确定这些调整。以上给出的实例是视频/图形处理方案，该方案具有目标帧速度形式的实时需求。CPU 和/或图形控制器的处理器时钟频率降低是建立在上述实时需求的期限裕度的基础上的，在这种情况下，期限裕度是指从特定表面或图像已被渲染成图像形式并已为显示准备就绪到预定将图像施加和显示在监视器上的这一时间点所余的这段时间。也可以用其他方法来定义图形处理方案的期限裕度。如上所述，该方法的总目标是：随着视频/图形序列的进行，将期限裕度逐步减少为零，同时尽量避免因处理器时钟频率的降低而造成过多的(或任何)帧丢失。

虽然上述功率管理技术可广泛应用于各种不同类型的电子系统(例如，包括台式计算机系统和服务器)，但这些技术在功耗上带来的

好处对便携式系统尤为理想，其中，便携式系统由电池或燃料电池供电，而这两种电池的能量供应均比较有限。

可以给出计算机程序产品或软件形式的本发明的一些实施例，其中，上述产品或软件可包括存储了指令的可机读媒体或计算机可读媒体，且这些指令可用于对计算机(或其他电子设备)进行编程，以执行根据本发明的实施例的处理。在其他实施例中，可以通过特定的含有微代码、硬接线逻辑电路的特定硬件器件或经过编程的计算机器件与定制的硬件器件的任何组合来执行这些操作。

可机读媒体可以是任意以通过机器(如一个或多个处理器组成的组、台式计算机、便携式计算机、生产工具或任何其他具有处理器的设备)访问的形式提供(即存储或发送)信息的机构，如可记录/非可记录媒体(如只读存储器(ROM)，随机存取存储器(RAM)，磁转盘存储媒体、光盘存储媒体)，和电、光、声或其他形式的传播信号(如载波、红外信号等)。

总之，本文描述了一种功耗管理技术的各个实施例。在前述的说明中，结合本发明的特定的示范性实施例对其进行了说明。然而，明显可以对本发明进行各种修改和变更，而不至于不背离如附加的权利要求所述的本发明的更广泛的主题和范围。例如，处理器工作点调整的给定实例是降低或增加处理器的时钟频率；该频率改变可伴随有处理器的电源电压改变，以进一步减小功耗。因此，应当将本说明书和附图视为说明而非限制性的。

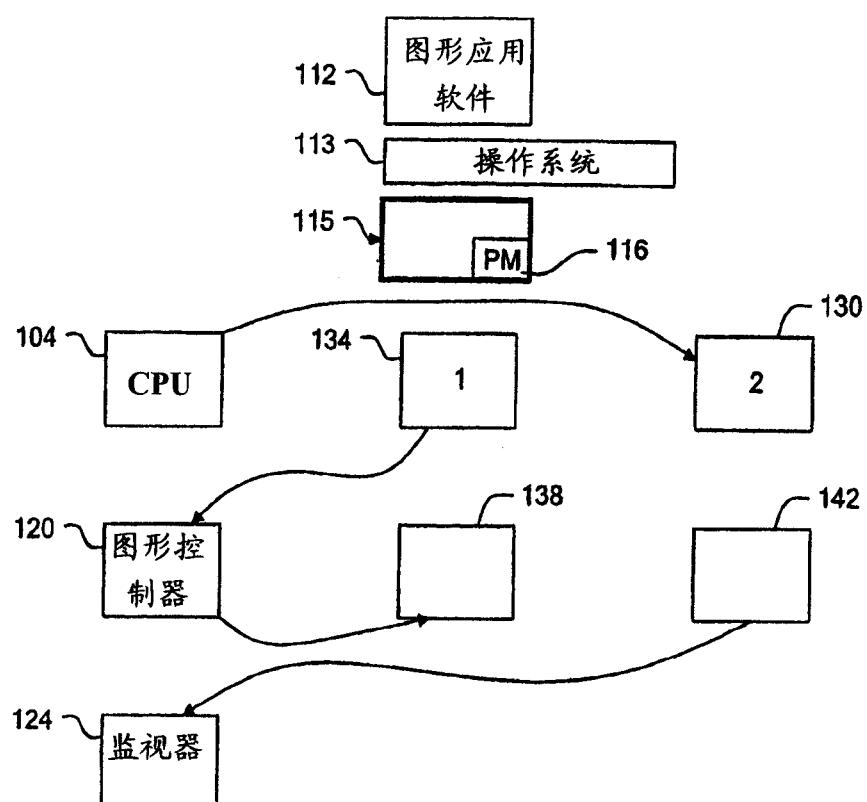


图 1

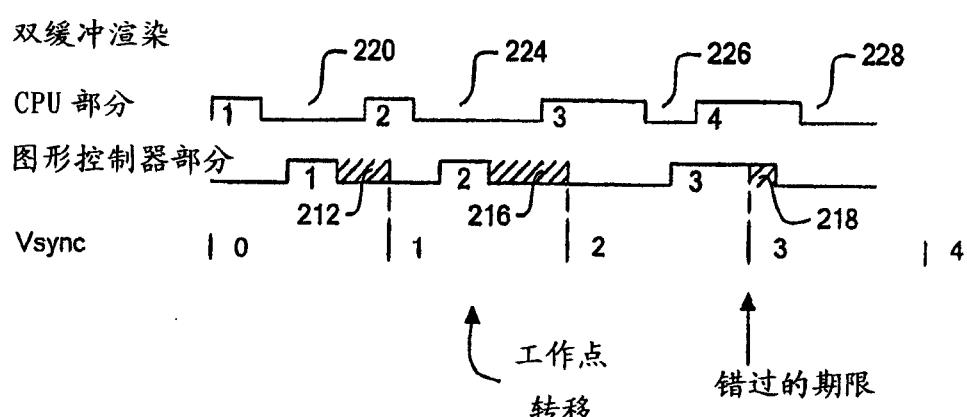


图 2

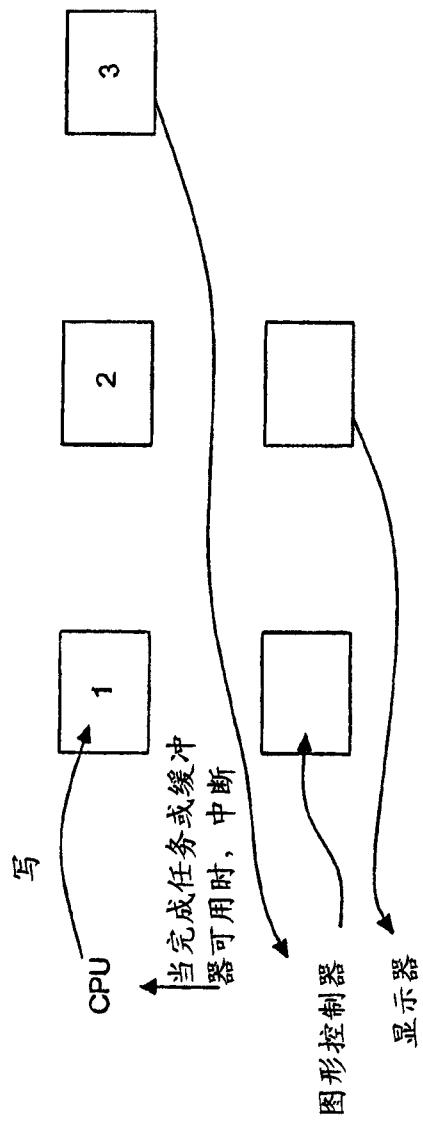


图 3

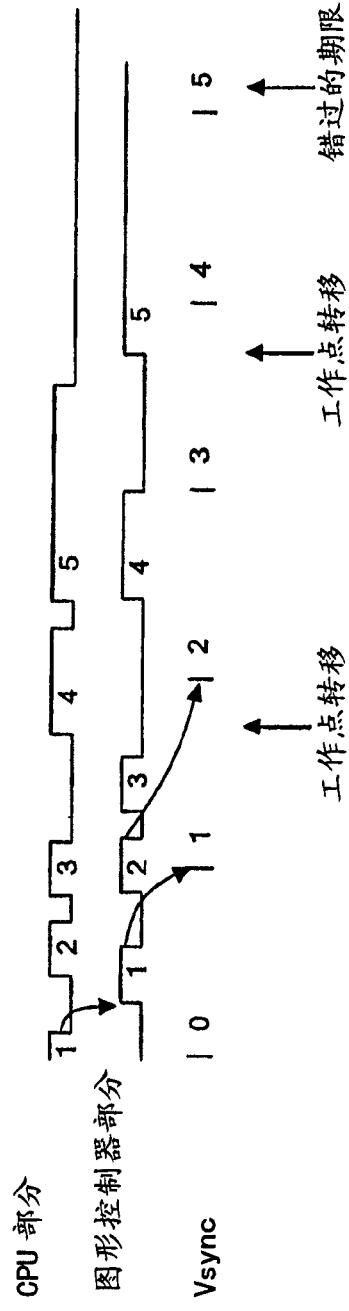
三缓冲渲染

图 4

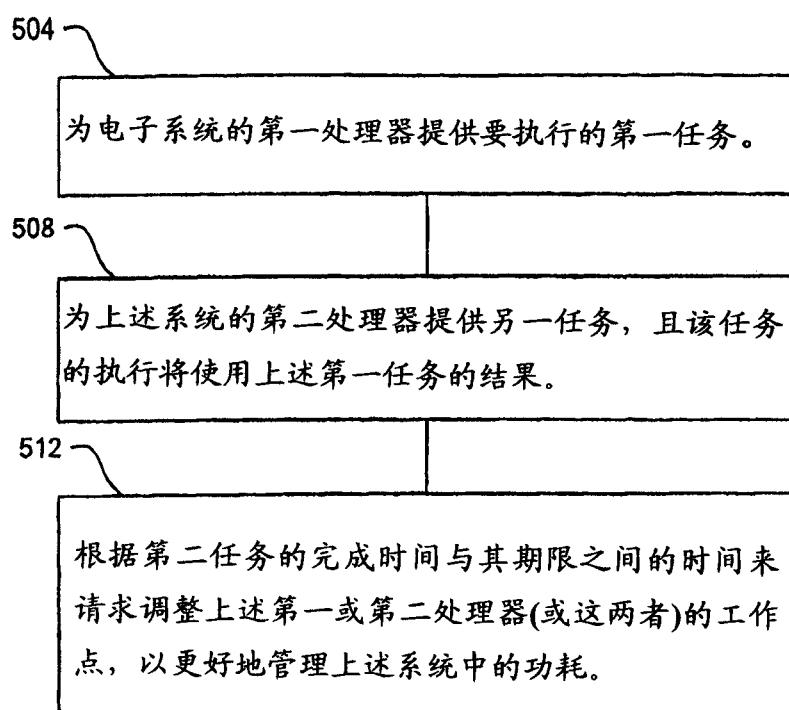


图 5