(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 102077271 A
(43)申请公布日 2011.05.25

(21)申请号 200980125004.X

(22)申请日 2009.04.22

(30)优先权数据
61/053,519 2008.05.15 US
12/212,805 2008.09.18 US

(85)PCT申请进入国家阶段日
2010.12.30

(86)PCT申请的申请数据
PCT/US2009/041446 2009.04.22

(87)PCT申请的公布数据
WO2009/140037 EN 2009.11.19

(71)申请人 苹果公司
地址 美国加利福尼亚

(72)发明人 I·亨德利 A·G·萨姆皮特

(74)专利代理机构 中国国际贸易促进委员会专利商标事务所 11038
代理人 马浩

(51)Int.Cl.
G09G 5/18(2006.01)
G09G 5/36(2006.01)
G06F 1/08(2006.01)
G06F 1/32(2006.01)

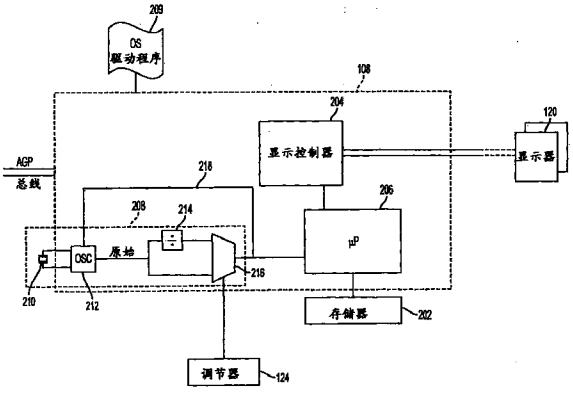
权利要求书 3 页 说明书 7 页 附图 3 页

(54)发明名称

图形处理单元的热管理

(57)摘要

某些实施例包括具有热管理能力的图形处理。该图形处理单元包括显示控制器，耦连到显示控制器的微处理引擎；和耦连到显示控制器和微处理引擎的时钟电路。该时钟电路可以进一步包括耦连到显示控制器的原始时钟信号，耦连到原始时钟信号的分频器，和耦连到分频器的多路复用器。分频器可以生成原始时钟信号的分频版本，该分频版本可被与原始时钟信号一起耦连到多路复用器。多路复用器可以向微处理引擎选择性地提供原始时钟信号和/或该时钟信号的分频版本，从而微处理引擎可以接收独立于图形处理单元的工作的定时信号，并且导致更少的假信号。



CN 102077271 A

1. 一种图形处理单元 (GPU)，包括 ：

显示控制器 ；

耦连到显示控制器的微处理引擎 ；

耦连到显示控制器和微处理引擎的时钟电路，该时钟电路进一步包括 ：

耦连到显示控制器的原始时钟信号 ；

耦连到原始时钟信号的分频器，其中分频器生成原始时钟信号的分频版本 ；

耦连到分频器和原始时钟信号的多路复用器，其中多路复用器向微处理引擎选择性地提供原始时钟信号或原始时钟信号的分频版本作为定时信号。

2. 如权利要求 1 所述的 GPU，其中，基于脉冲宽度调制 (PWM) 信号的脉冲宽度，多路复用器向微处理引擎选择性地提供原始时钟信号或原始时钟信号的分频版本。

3. 如权利要求 1 所述的 GPU，其中分频器被调整，以使得原始时钟信号具有基本上为零的频率。

4. 如权利要求 1 所述的 GPU，其中微处理引擎以减小的执行速率执行操作，而显示控制器以基本上相同的执行速率操作。

5. 如权利要求 1 所述的 GPU，其中提供给微处理引擎的定时信号表现为原始时钟信号和原始时钟信号的分频版本的平均。

6. 如权利要求 4 所述的 GPU，还包括提供给多路复用器的至少一个附加时钟信号，其中提供给微处理引擎的定时信号表现为原始时钟信号、原始时钟信号的分频版本和该至少一个附加时钟信号的平均。

7. 如权利要求 4 所述的 GPU，其中使用提供给多路复用器的 PWM 信号来修改以定时信号表示的平均值。

8. 如权利要求 6 所述的 GPU，其中所述 PWM 信号基于对 GPU 温度变化的测量结果。

9. 如权利要求 7 所述的 GPU，其中所述 PWM 信号基于二极管结的温度变化。

10. 如权利要求 1 所述的 GPU，其中所述定时信号包括的源于原始时钟信号的转变的数目多于源于原始时钟信号的下分频版本的转变的数目。

11. 如权利要求 9 所述的 GPU，其中在 GPU 正在执行指令时，修改源于原始时钟信号的转变与源于原始时钟信号的下分频版本的转变之间的比例。

12. 如权利要求 10 所述的 GPU，其中定时信号内的转变总数基本上恒定，而不论它们之间的比例被如何修改。

13. 如权利要求 10 所述的 GPU，其中所述定时信号在连续的时间段内的转变总数根据从由 PWM 信号、变化的电平或寄存器设置组成的组中选择的信号而改变。

14. 如权利要求 1 所述的 GPU，其中所述定时信号被反馈给时钟电路，以对定时信号进行偏斜消除。

15. 如权利要求 1 所述的 GPU，其中独立于微处理引擎正在执行的代码内的中断，修改所述定时信号。

16. 如权利要求 1 所述的 GPU，其中在保持微处理引擎执行的操作的总数的同时，修改定时信号的转变分布。

17. 一种控制 GPU 的方法，包括以下动作 ：

向显示控制器提供原始时钟信号 ；

生成原始时钟信号的分频版本；和

向微处理引擎提供定时信号，所述定时信号选择性地包括原始时钟信号或原始时钟信号的分频版本；

其中微处理引擎以减小的速率执行操作，而显示控制器以基本上相同的执行速率操作。

18. 如权利要求 14 所述的方法，其中定时信号表现为原始时钟信号和原始时钟信号的分频版本的平均。

19. 如权利要求 14 所述的方法，其中原始时钟信号的分频版本被调整，以使得原始时钟信号的频率基本上为零。

20. 如权利要求 14 所述的方法，还包括以下动作：基于从由 PWM 信号、变化的电平、或寄存器设置组成的组中选择的信号，改变定时信号的组成。

21. 如权利要求 14 所述的方法，还包括以下动作：在定时信号内包括数目比源于原始时钟信号的下分频版本的转变数目多的源于原始时钟信号的转变。

22. 如权利要求 19 所述的方法，还包括以下动作：在 GPU 正在执行指令时，修改源于原始时钟信号的转变与源于原始时钟信号的下分频版本的转变之间的比例。

23. 如权利要求 20 所述的方法，其中定时信号内的转变总数基本上恒定，而不论它们之间的比例被如何修改。

24. 如权利要求 20 所述的方法，其中独立于微处理引擎正在执行的代码内的中断，发生修改动作。

25. 如权利要求 22 所述的方法，还包括以下动作：停止热病毒的操作。

26. 如权利要求 16 所述的方法，还包括以下动作：在保持微处理引擎执行的操作的总数的同时，修改定时信号的转变分布。

27. 一种计算机系统，包括：

中央处理单元 (CPU)；

耦连到 CPU 的 GPU；

耦连到 GPU 的一个或多个显示器；和

耦连到 GPU 的调节器；

其中 GPU 还包括：

显示控制器；

耦连到显示控制器的微处理引擎；

耦连到显示控制器和微处理引擎的时钟电路，该时钟电路进一步包括：

耦连到显示控制器的原始时钟信号；

耦连到原始时钟信号的分频器，其中分频器生成原始时钟信号的分频版本；

耦连到分频器和原始时钟信号的多路复用器，其中多路复用器向微处理引擎选择性地提供原始时钟信号或原始时钟信号的分频版本作为定时信号。

28. 如权利要求 25 所述的计算机系统，其中，基于 PWM 信号的脉冲宽度，多路复用器向微处理引擎选择性地提供原始时钟信号或原始时钟信号的分频版本。

29. 如权利要求 26 所述的计算机系统，其中所述分频器被调整，以使得原始时钟信号具有基本上为零的频率。

30. 如权利要求 26 所述的计算机系统，还包括操作系统 (OS) 驱动器，其中 OS 驱动器控制 PWM 信号的脉冲宽度。

31. 如权利要求 26 所述的计算机系统，其中所述调节器内的预定算法控制 PWM 信号的脉冲宽度。

32. 如权利要求 25 所述的计算机系统，其中微处理引擎以减小的执行速率执行操作，而 GPU 的一部分以基本上相同的执行速率操作。

33. 如权利要求 29 所述的计算机系统，其中微处理引擎以减小的执行速率执行操作，而显示控制器以基本上相同的执行速率操作。

34. 如权利要求 25 所述的计算机系统，其中使用提供给多路复用器的 PWM 信号来修改以定时信号表示的平均值。

35. 如权利要求 30 所述的计算机系统，其中所述 PWM 信号基于对计算机系统中的温度变化的测量结果。

36. 如权利要求 31 所述的计算机系统，其中所述 PWM 信号基于二极管结的温度变化。

37. 如权利要求 25 所述的计算机系统，其中所述定时信号包括的源于原始时钟信号的转变的数目多于源于原始时钟信号的下分频版本的转变的数目。

38. 如权利要求 33 所述的计算机系统，其中在 GPU 正在执行指令时，修改源于原始时钟信号的转变与源于原始时钟信号的下分频版本的转变之间的比例。

39. 如权利要求 34 所述的计算机系统，其中定时信号内的转变总数基本上恒定，而不论它们之间的比例被如何修改。

40. 如权利要求 25 所述的计算机系统，其中独立于微处理引擎正在执行的代码内的中断，修改所述定时信号。

41. 如权利要求 36 所述的计算机系统，其中所述计算机系统是便携式的。

42. 如权利要求 37 所述的计算机系统，其中所述计算机系统包括至少两个显示器。

43. 如权利要求 25 所述的计算机系统，其中在保持微处理引擎执行的操作的总数的同时，修改定时信号的转变分布。

图形处理单元的热管理

[0001] 相关申请的交叉引用

[0002] 本申请根据 35U.S.C. §119(e) 要求提交于 2008 年 5 月 15 日的题目为“Thermal Management of Graphics Processing Units”的美国临时专利申请No.61/053,519,以及提交于 2008 年 9 月 18 日的题目为“Thermal Management of Graphics Processing Units”的美国非临时专利申请 No.12/212,805 的优先权,通过引用将其完整结合在此。

技术领域

[0003] 本发明一般地涉及电子设备的热管理,并且更具体地涉及提供图形处理单元的热管理。

背景技术

[0004] 电子设备普遍存在于社会中,并且可见于从腕表到计算机的任意物品。这些电子设备的复杂度和精巧性通常随着每代增加,结果,较新的电子设备通常比其先辈消耗更多的功率量。随着功耗的增加,电子设备内的电路可能生成更大程度的热量,这对于电路的工作可能是有害的。

[0005] 现代电子设备的趋势是使得每一代更小巧,这加剧了这个问题。结果,来自相继各代电子设备的每单位体积的温度可能上升到对用户或设备本身有潜在危险的水平。出于这个原因,可以给微处理器和其它电路配备散热器和 / 或风扇,以便将热量传出芯片并将微处理器保持在安全操作范围内。还可以实施附加的热管理技术,诸如选择性地关闭电子设备的特别耗电的元件。

[0006] 除了具有增加的功耗之外,许多现代设备还具有比其先辈更强大的图形能力。个人计算机尤其如此,用户可能为每个计算机使用多个监视器,每个监视器可能能够呈现复杂的计算机图形图像。然而,许多现代计算机的热管理技术可能阻碍计算机系统提供复杂图形能力的能力。例如,当微处理器进入低功率模式时,由于处理器未在执行指令,可能呈现一个或多个屏幕假信号 (glitch)。在具有多个显示器的计算机系统和 / 或正在播放电影的计算机系统中尤其如此。

[0007] 因此,需要向计算机系统提供防止屏幕假信号的热管理。

发明内容

[0008] 某些实施例包括具有热管理能力的图形处理单元 (GPU)。该 GPU 可以包括显示控制器,耦连到显示控制器的微处理引擎,和耦连到显示控制器和微处理引擎的时钟电路。该时钟电路还可以包括 :耦连到显示控制器的原始时钟信号 ;耦连到原始时钟信号的分频器 ;和耦连到分频器的多路复用器。该分频器可以生成原始时钟信号的分频版本,该分频版本可与原始时钟信号一起耦连到多路复用器。多路复用器可以选择性地向微处理引擎提供原始时钟信号或该时钟信号的分频版本,从而微处理引擎可以接收独立于 GPU 的操作的定时信号,并且导致更少的假信号。

[0009] 其它实施例可以包括控制 GPU 的方法，该方法包括如下动作：向显示控制器提供原始时钟信号；生成原始时钟信号的分频版本；和向微处理引擎提供选择性地包括原始时钟信号或原始时钟信号的分频版本的定时信号。以这种方式，微处理引擎可以以减小的速率执行操作，而显示控制器以基本上相同的执行速率操作。

[0010] 其它实施例可以包括具有热管理能力的计算机系统。该计算机系统可以包括中央处理单元 (CPU)；耦连到 CPU 的 GPU；耦连到 GPU 的一个或多个显示器；和耦连到 GPU 的调节器 (regulator)。该 GPU 可以包括：显示控制器；耦连到显示控制器的微处理引擎；和耦连到显示控制器和微处理引擎的时钟电路。该时钟电路还可以包括：耦连到显示控制器的原始时钟信号；耦连到原始时钟信号的分频器；和耦连到分频器的多路复用器。分频器可以生成原始时钟信号的分频版本，该分频版本与原始时钟信号一起耦连到多路复用器。多路复用器可以向微处理引擎选择性地提供原始时钟信号或该时钟信号的分频版本，从而微处理引擎可以接收独立于 GPU 的操作的定时信号，并且导致在一个或多个显示器上更少的假信号。

附图说明

[0011] 图 1 示出了示例的计算机系统；

[0012] 图 2 示出了实施热管理的示例 GPU；

[0013] 图 3A 示出了示例的脉冲宽度调制信号；

[0014] 图 3B 示出了可以由图 3A 的脉冲宽度调制信号引起的示例时钟；

[0015] 图 3C 示出了可以由图 3A 的脉冲宽度调制信号引起的另一个示例时钟。

[0016] 在不同附图中使用的相同参考标号表示类似或相同的项。

具体实施方式

[0017] 下面的讨论描述了在防止屏幕假信号的同时，向图形处理单元提供热管理的各种实施例。虽然可能详细描述了这些实施例中的一个或多个，但是公开的实施例不应当被解释为或用于限制本公开的范围，包括权利要求的范围。另外，本领域的技术人员将会理解，下面的描述具有广泛的应用。因此，对任意实施例的讨论仅仅意图示例，而不旨在表示包括权利要求在内的本公开的范围被局限于这些实施例。

[0018] 某些实施例包括具有热管理能力的图形处理单元 (GPU)。该 GPU 可以包括显示控制器，耦连到显示控制器的微处理引擎，和耦连到显示控制器和微处理引擎的时钟电路。该时钟电路还可以包括：耦连到显示控制器的原始时钟信号；耦连到原始时钟信号的分频器；和耦连到分频器的多路复用器。分频器生成原始时钟信号的分频版本，该分频版本可与原始时钟信号一起耦连到多路复用器。多路复用器可以向微处理引擎选择性地提供原始时钟信号或该时钟信号的分频版本，从而微处理引擎可以接收独立于 GPU 的操作的定时信号，并且导致更少的假信号。

[0019] 图 1 示出了可以在一个实施例中实施的示例计算机系统 100。在探究图 1 的细节之前，应当注意在图 1 中列出并且在下面提及的组件仅是一种可能的实现的例子。可以在其它实现中使用其它组件、总线和 / 或协议，而不脱离该详细描述的精神和范围。

[0020] 现在参考图 1，计算机系统 100 包括中央处理单元 (CPU)102，CPU 102 可通过 CPU

总线电耦连到桥逻辑器件106。桥逻辑器件106有时根据其相对于其它系统组件（诸如南桥119）的位置而被称为“北桥”。北桥106可以通过存储器总线电耦连到主存储器阵列104，并且还可以通过高级图形端口（AGP）总线电耦连到GPU 108。一般地，AGP总线是给计算机系统100的主板附加图形功能的工业标准方法。北桥106还可以通过例如主扩展总线（BUS A），诸如PCI总线或EISA总线，将CPU 102、存储器104和GPU 108耦连到该系统中的其它外设。

[0021] 使用BUS A的总线协议操作的各种组件可以存在于该总线上，诸如音频设备110、IEEE1394接口设备112和网络接口卡（NIC）114。这些组件可被集成到PCB上，或它们可被插入连接到BUS A的扩展槽118中。如果计算机系统100中提供了别的次扩展总线，可以使用另一个桥逻辑器件119将主扩展总线BUS A电耦连到次扩展总线（未示出）。如上所述，由于其相对于其它系统组件的位置，桥逻辑器件119有时被称为“南桥”。

[0022] 在某些实施例中，图1所示的两个或更多个组件可被实现为单个组件。例如，在某些实施例中，GPU可与北桥106或计算机系统100的任何其它组件集成在一起。

[0023] 计算机系统100可以通过GPU 108耦连到一个或多个显示单元120。以这种方式，计算机系统100可以支持向一个或多个显示单元120呈现计算机生成的图形图像。在某些实施例中，诸如在膝上型计算机系统的情况下，所述一个或多个显示单元120中的至少一个显示单元可以集成到计算机系统100中。

[0024] 如虚线指示的，计算机系统100可包含在机壳122内。另外，机壳122可能具有有限的热容量或预算。例如，在某些实施例中，机壳122的热预算可以是32瓦。如前面所述，许多电子设备，诸如计算机系统100，被制造为位于越来越小的机壳122内，从而设备的热预算可能随着相继的产品代而减小。

[0025] 通过实施一个或多个功率调节电路124或方案，计算机系统100可以确保其不会超出其热预算。一个或多个功率调节电路124可以采取温度监视设备的形式。在某些实施例中，功率调节电路124的温度监视设备可以是一个或多个基于硅的二极管（未示出），其可以具有近似-2mV/℃的温度系数。随着温度上升，这些二极管上的电压可能下降。类似地，随着温度下降，这些二极管上的电压可能上升。功率调节电路124可以监视这种变化的电压，以便确定功率调节电路124和/或计算机系统100的工作温度。

[0026] 特别地，这些系统中的GPU可以具有很宽的工作功率变化，并且可能是计算机系统100内功耗最大的组件之一。例如，CPU 102可能消耗30瓦的最大功率量，而GPU 108可能消耗范围从5到18瓦的第二大功率量。在这个相同例子中，存储器104可能消耗近似3到4瓦的功率，而北桥106可能消耗2到4瓦的功率。

[0027] 由于GPU 108可能是计算机系统100中功耗最大的组件之一，常规的计算机系统通常试图对GPU 108执行热管理功能。不幸的是，在常规计算机系统上实施的热管理功能通常导致显示在一个或多个显示器120中至少一个显示器上的图像中的假信号。这些假信号可能是由于常规热管理电路通常仅具有用于控制计算机系统100内任意特定组件生成的热量的少数选项。例如，一个这种热管理选项是降低CPU 102的速度以使它仅消耗最少的功率量。这可以通过降低CPU的工作速度来实现，然而，由于可能不具有可用的足够处理能力来以及时的方式传递图像以便显示运动图形，这种行为通常在正由CPU呈现的图像中引入假信号。这些假信号可能影响基于运动的图形项的运行，诸如在计算机系统100上播

放影片。

[0028] 根据至少某些实施例，功率调节电路 124 可以在计算机系统 100 上实施热管理功能，而不引起在一个或多个显示器 120 上显示的图像中的假信号。图 2 示出了具有这种热管理方案的 GPU 108。参考图 2，GPU108 可以通过 AGP 总线接收数据，以及处理并且在一个或多个显示器 120 上显示该数据。另外，如图所示，GPU 108 可以从功率调节电路 124 接收 GPU ENABLE 信号（下面参考图 3A 更详细描述）。

[0029] 存储器 202 可耦连到 GPU 108。在某些实施例中，存储器 202 可以与计算机系统 100 中的存储器 104 相同。在其它实施例中，存储器 202 可以是专用视频存储器，诸如与存储器 104 相分离的视频随机访问存储器 (VRAM)。在工作期间，存储器 202 可以存储由 GPU 108 操作的数据。

[0030] 如图 2 所示，GPU 108 可以包括显示控制器 204、微处理引擎 206 和时钟电路 208。通过将图片格式数据转传递到一个或多个显示器 120，显示控制器 204 可以在一个或多个显示器 120 上呈现图像。在某些实施例中，用于在显示控制器 204 和一个或多个显示器 120 之间传递视频数据的格式是数字视频接口 (DVI) 标准。在其它实施例中，该格式是视频图像阵列 (VGA) 标准。包括 DVI 和 / 或 VGA 的实施例仅是例子，实际上，在替换实施例中可以使用其它标准和 / 或视频标准。微处理引擎 206 可以耦连到显示控制器 204，并且可以提供原始图像数据，显示控制器 204 然后将原始图像数据格式化为可被一个或多个显示器 120 显示的数据。

[0031] 操作系统 (OS) 驱动程序 209 可以耦连到 GPU 108，并且指挥应用在 GPU 108 上的执行。在计算机系统 100 上实现的实际 OS 驱动程序 209 可以是各种各样的。在某些实施例中，OS 驱动程序 209 可以是来自苹果公司的 Mac OS 驱动程序。在其它实施例中，该 OS 驱动程序可以是来自微软公司的基于 Windows 的驱动程序。另外，应当理解，OS 驱动程序 209 可以是来自任意适合的 OS 的任意适合的 OS 驱动程序。

[0032] 关于 GPU 108 的功耗，显示控制器 204 可以消耗相对恒定的功率量，而微处理引擎 206 可以具有随着正被执行的特定应用而变化的功耗。以这种方式，当 OS 驱动程序 209 指示微处理引擎 206 执行图形密集应用时，微处理引擎 206 可能消耗 GPU 108 的大部分功率。例如，显示控制器 204 可以占用 4 瓦的相对恒定的功耗，而微处理引擎 206 可能占用 1 到 18 瓦的可变功耗。因此如果机壳 122 的热预算是 18 瓦，并且显示控制器 204 和微处理引擎 206 正在消耗最大的功率量，则机壳的热预算可能超出近似 22％。这仅是为什么可能希望实施 GPU 108 的热管理的一个例子。另外，实施 GPU 108 的热管理可以使得整个计算机系统 100 更有能量效率。

[0033] 组件（例如 GPU 108）的可能变化的功耗可能给具有较小机壳的消费电子设备尤其带来挑战。由于许多计算机系统的微型化，机壳 122（图 1 所示）可能具有比较大机壳更小的热预算。结果，较小的电子设备通常在所消耗功率和 / 或所生成热量的可变量方面具有更小的超额裕度。例如，如果计算机系统 100 是台式计算机，其机壳 122 可能具有比类似配置（例如，类似处理器速度、存储器容量等）的膝上计算机更大的热预算，并且膝上计算机可能不能承受由于变化的功耗引起的功率超额。由于这些电子设备可能具有较小的热预算以及功耗方面较小的超额裕度，可能希望控制可能引起这种超额的可变功耗。

[0034] 可以部分地通过时钟电路 208 提供对可变功耗的控制。时钟电路 208 可以包括耦

连到振荡电路 212 的晶体 210。虽然晶体 210 被示出为耦连在 GPU 108 的两个端子之间，其它实施例可以采用单端子布置中的晶体 210，其中晶体 210 耦连在 GPU 108 的单个端子和地之间。振荡电路 212 可以是任意的类型，并且还可以包括时钟树和 / 或频率调制电路，诸如锁相环 (PLL)。

[0035] 从振荡电路 212 得到的信号可以是可耦连到显示控制器 204 的原始 (RAW) 时钟信号。原始时钟信号还可耦连到分频器 214，在分频器 214 处，该原始时钟信号被一个除数修改，然后被提供给多路复用器 216。原始时钟信号可以是来自晶体振荡器的频率合成信号。例如，在某些实施例中，原始时钟信号可以来自 PLL，PLL 合成来自晶体振荡器的相对频率稳定的时钟信号。其它实施例可以采用延迟锁定环 (DLL) 以实现相同功能。原始时钟频率范围的例子包括从大约 100MHz 到大约 1GHz。

[0036] 分频器 214 可以提供具有比原始时钟信号更低频率的原始时钟信号的下分频版本。在某些实施例中，分频器 214 的分频值可以包括 2 到 32。在其它实施例中，可以设置分频值以使得分频器的定时信号可以具有非常低的频率，并且在某些情况下，可以接近零。因此如果分频器 214 是能够将值设置在从 2 到 256 的范围的 3 位分频器，则分频器 214 可被配置为具有 256 的分频值，从而生成非常低的频率（在下面的图 3E 中以 308 示出）。

[0037] 在某些实施例中，微处理引擎 206 消耗的功率与来自分频器 214 的频率近似成比例，并且因此，可以通过控制分频器 214 的分频值来控制微处理引擎 206 消耗的功率。因此，在来自分频器 214 的定时信号的频率基本上为零的实施例中，微处理引擎 206 消耗的功率可以比定时信号不是基本上为零时要低。

[0038] 在工作过程中，多路复用器 216 可以在来自振荡电路 212 的原始时钟和来自分频器 214 的该原始时钟的下分频版本之间进行选择。多路复用器 216 可以基于来自功率调节电路 124 的 GPU_ENABLE 信号进行这种选择。GPU_ENABLE 信号可用于控制多路复用器在来自振荡电路 212 的原始时钟和来自分频器 214 的该原始时钟的下分频版本之间的选择，其中可以基于 GPU_ENABLE 信号的脉冲宽度来改变其中任一信号可被选择的时间段（如下面在图 3A 的情形中描述的）。作为在原始时钟或该原始时钟的下分频版本之间选择性地挑选的结果，提供给微处理引擎 206 的时钟信号在时间上可以是两种时钟速率的工作周期加权平均 (duty-cycle weighted-average)。在某些实施例中，多路复用器 216 对多于两种信号进行平均。

[0039] 通过选择性地应用原始时钟和其下分频版本，从多路复用器 216 提供给微处理引擎 206 的总时钟信号可以被配置为使得可以积极主动地 (proactively) 控制微处理引擎 206 的执行速度。即，可以触发微处理引擎 206 内的逻辑块（未示出）以根据来自多路复用器 216 的信号中的转变 (transition) 而操作（术语“转变”可用于指信号从高到低的运动和 / 或信号从低到高的运动）。这些逻辑块消耗一定的功率量，并且随着每次转变生成一定的热量。由于原始时钟或该原始时钟的下分频版本的平均可能包含较少的转变，因此 GPU 108 生成的热量可以减少。

[0040] 在某些实施例中，GPU_ENABLE 信号可以是图 3A 所示的脉冲宽度调制 (PWM) 信号的形式。如图所示，PWM 信号可以是具有如图 3A 的双端箭头所示的变化的脉冲宽度的信号。这些变化的脉冲宽度可以导致 GPU_ENABLE 的一个或多个不同周期，诸如图 3A 所示的周期 A 和 / 或周期 B。在某些实施例中，PWM 信号的宽度可以基于功率调节电路 124 内的预定算

法而变化。在其它实施例中，PWM 信号的宽度可以基于来自 OS 驱动程序 209 的输入而变化。其它实施例可以采用具有模拟电平或寄存器设置形式的 GPU_ENABLE 信号。

[0041] 当 GPU_ENABLE 信号为低时，多路复用器 216 可以选择性地将来自分频器 214 的原始时钟的下分频版本耦连到微处理引擎 206。类似地，当 GPU_ENABLE 信号为高时，多路复用器 216 可以选择性地将来自振荡电路 212 的原始时钟耦连到微处理引擎 206。图 3B 示出了示例性的所得到的提供给微处理引擎 206 的时钟信号 302。由于 GPU_ENABLE 的脉冲宽度在周期 B 中比在周期 A 中宽，所以时钟信号 302 中可能发生更多数目的转变。在这个例子中，周期 A 被示出为包括总共 12 个转变，其中 8 个来自原始时钟，4 个来自分频器 214。另一方面，周期 B 被示出为包含总共 18 个转变，其中 12 个来自原始时钟，6 个来自分频器 214。结果，时钟信号 302 在周期 B 期间比在周期 A 期间具有更高的平均频率，并且 GPU 108 可能在周期 B 期间比在周期 A 期间工作于更高的温度。

[0042] 可以用其它方式修改提供给 GPU 108 的时钟信号。在某些实施例中，发生在由多路复用器 216 提供的信号中的转变的平均数目可保持相对恒定，并且 GPU_ENABLE 信号的脉冲宽度可保持相对恒定，而这些转变的整体分布可以改变。图 3C 表示具有这些特性的示例信号 304。

[0043] 参考图 3C，可以理解，通过改变原始时钟的频率（例如，通过调整 PLL 输出），以及改变来自分频器 214 的信号的频率（例如，通过调整分频值），可以获得不同的转变分布。比较信号 302 和 304，它们在周期 A 和周期 B 期间分别具有相同数目的转变，并且因此 GPU 108 可以在周期 A 和周期 B 期间分别执行近似相同数目的操作。虽然信号 302 和 304 的任意给定周期期间的转变总数可能是相同的，但是转变的分布可以改变。即，在周期 A 期间，信号 304 可以包含较多的来自原始时钟的转变以及较少的来自分频器 214 的转变。结果，与如果将信号 302 提供给 GPU 108 相比，当将信号 304 提供给 GPU 108 时，GPU 108 可以在周期 A 的原始时钟部分中执行更多指令。类似地，在周期 B 期间，虽然在两个信号 302 和 304 中发生了相同数目的转变，但是在信号 304 的原始时钟部分期间发生了比信号 302 更多的转变。因此，即使 GPU 108 可能执行相同数目的操作，对于信号 302 和 304，同样时间内 GPU 108 生成的热量也可能不同。如果 GPU 108 的封装改变（例如，由于在制造过程中的某个后来时间的成本决策），并且导致 GPU 108 散热的能力改变，则可能希望具有上述特征。

[0044] 在某些实施例中，还可以通过延长提供给微处理引擎 206 的原始时钟的一些部分，实现可替换的转变分布。图 3D 示出了具有这种转变分布的示例定时信号 306。

[0045] 在其它实施例中，还可以通过将分频器 214 编程为使得原始时钟具有基本上为零的频率的分频值，来实现可替换的分布。例如，图 3E 示出了示例定时信号 308，其中原始时钟在信号周期的至少一部分上具有基本上为零的频率。

[0046] 通过将信号 302-308 提供给微处理引擎 206，可以通过独立于显示控制器 204 的操作来修改在微处理引擎 206 上执行的应用的执行速率，从而更精细地控制微处理引擎 206 的可变功耗要求。如果信号 302-308 被施加到显示控制器 204，这可能导致一个或多个显示器 120 上的假信号。

[0047] 另外，由于可以在微处理引擎 206 可能正在执行来自 OS 驱动程序 209 的命令的同时将信号 302-308 施加到微处理引擎 206，这可以使得在一个或多个显示器 120 上显示的图像中有更少的假信号。如果不向微处理引擎 206 提供信号 302-308，在微处理引擎 206 可

以实施热管理机制之前，OS 驱动程序 209 可能需要等待微处理引擎 206 完成任何特定的指令集。换言之，如果不提供信号 302-308，OS 驱动程序 209 可能必须将时钟修改放置在微处理引擎 206 的处理中断内。即使功率调节电路 124 可能指示需要实施热管理，在实施热管理技术之前等待处理中断发生可能使得微处理引擎 206 的温度继续升高。当功率调节电路 124 能够实施某种形式的热管理时（即，在处理中的下一个中断处），GPU 108 可能已经消耗了如此多的功率以至于可能需要采取激烈的措施，诸如完全关闭 GPU 108。例如，如果 GPU 108 消耗太多功率，并且 OS 驱动程序 209 不能实施热管理，计算机系统 100 可能简单地关闭 GPU 108 以防止灾难性的损坏。

[0048] 以这种方式关闭 GPU 108 可能导致呈现在一个或多个显示器 120 上的图像中的假信号。通过向微处理引擎 206 提供信号 302，由于微处理引擎 206 的功率可被主动地（与被动地相反）控制，因此可以防止这些假信号出现，从而最小化了 GPU 108 被灾难性关闭的次数。在便携系统中（其中热预算相对较小）可能特别希望实施这种热管理方案，这些便携系统支持多个显示器，并且可能需要由微处理引擎 206 进行附加处理（并且因此生成附加的热量）。

[0049] 在某些计算机系统中，被称为“热病毒”的应用可被恶意执行。这些热病毒的代码中蓄意不包含处理中断，从而计算机系统将由于热超载而关闭。通过实施信号 302，由于功率调节电路 124 可以控制生成的热量而不用考虑 OS 驱动程序 209 必须等待处理中断，这些热病毒的影响可被克服。

[0050] 在某些实施例中，振荡电路 212 可以沿着定时路径在各个时间点处对一个或多个定时信号进行偏斜消除 (de-skew)。例如，从分频器 214 到多路复用器 216 的信号可能被路由到 GPU 108 各处，从而引入时钟偏斜。在这些情况下，通过例如通过连接 218 将被怀疑的信号与由振荡电路 212 生成的信号进行比较，振荡电路 212 可以利用 PLL 消除这种偏斜。应当注意，连接 218 仅是能够向微处理引擎 206 提供定时信号的电路的一个代表，还可以使用更复杂的电路。

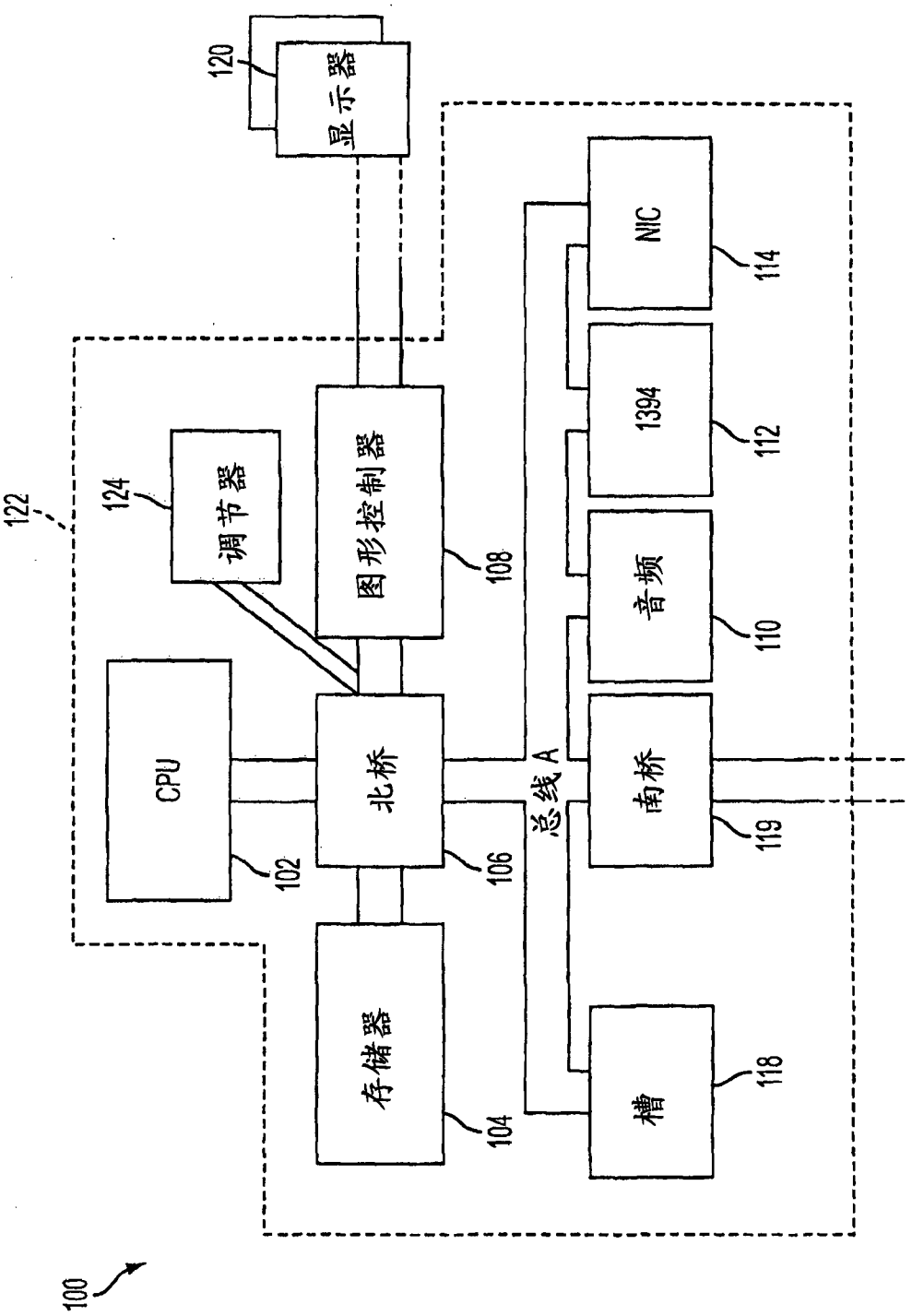

100
122
CPU
102
北桥
106
图形控制器
108
显示器
120
调节器
124
存储器
104
总线 A
南桥
119
NIC
114
1394
112
音频
110
槽
118

图 1

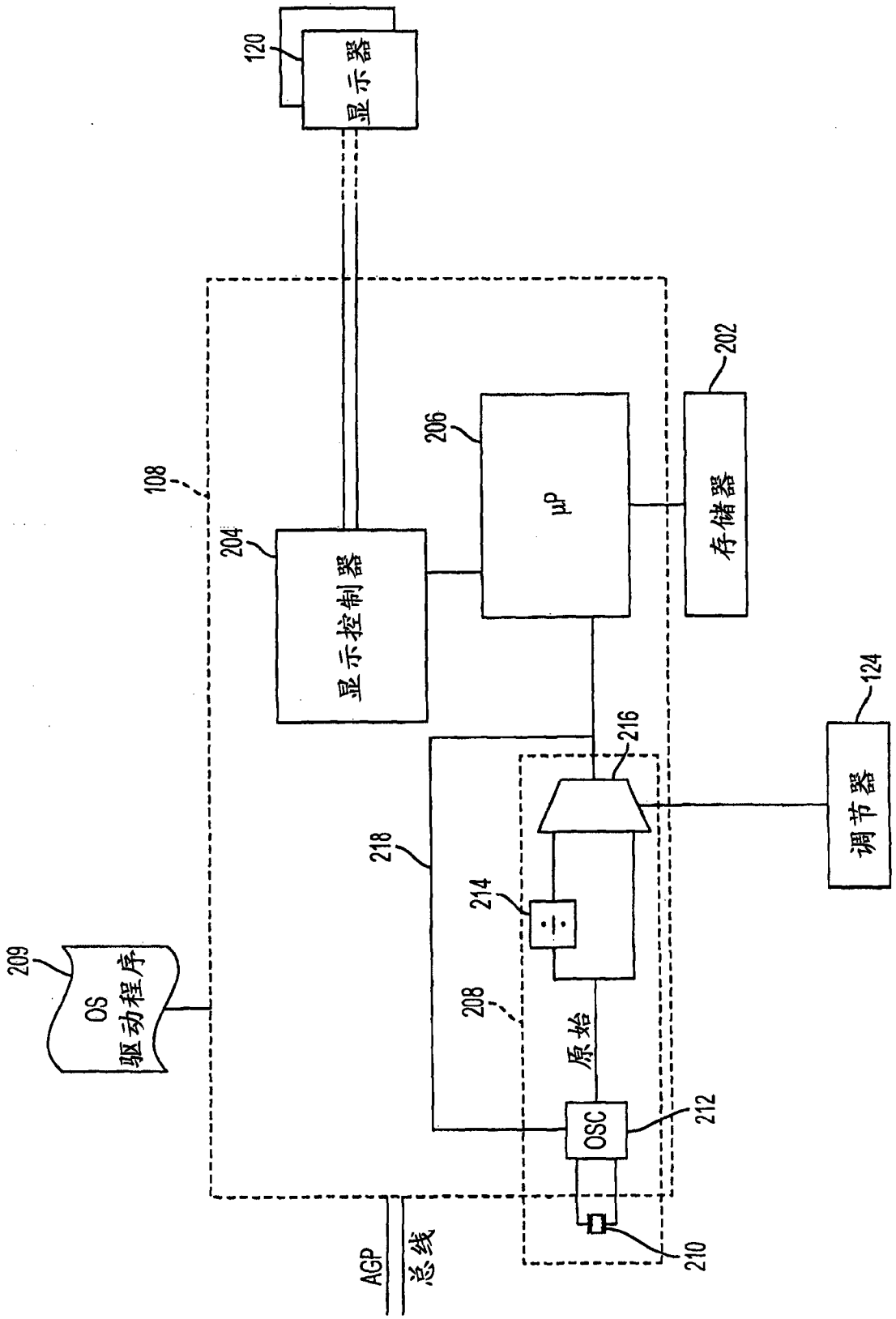


图 2

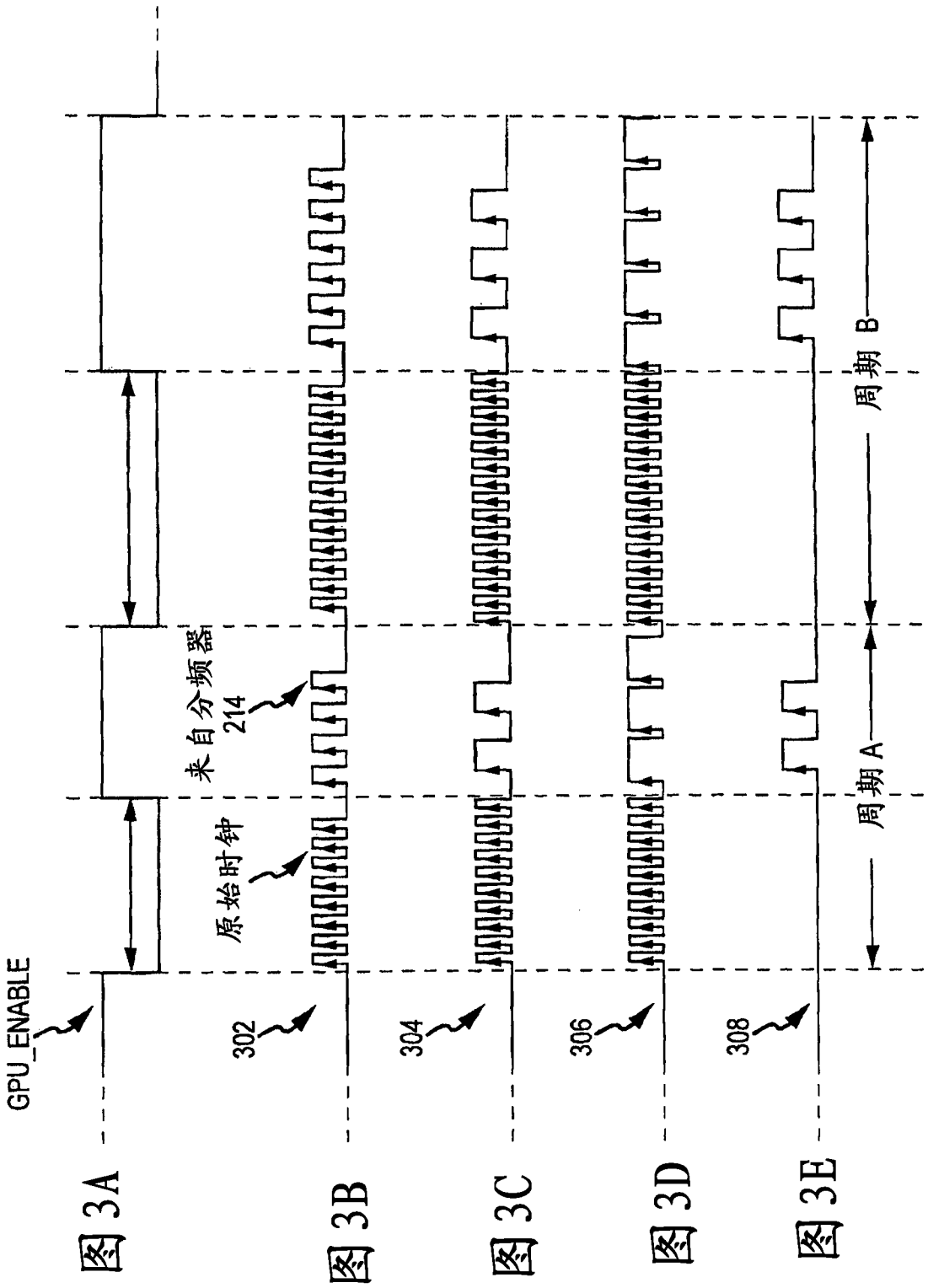


图 3A 图 3B 图 3C 图 3D 图 3E