

[19] 中华人民共和国国家知识产权局

[51] Int. Cl⁷

G06F 1/32

G06F 12/00 G06F 13/00



[12] 发明专利说明书

[21] ZL 专利号 95196998.6

[45] 授权公告日 2004 年 4 月 14 日

[11] 授权公告号 CN 1145869C

[22] 申请日 1995.12.20 [21] 申请号 95196998.6

[30] 优先权

[32] 1994.12.23 [33] US [31] 08/363,735

[86] 国际申请 PCT/US1995/016601 1995.12.20

[87] 国际公布 WO96/32671 英 1996.10.17

[85] 进入国家阶段日期 1997.6.23

[71] 专利权人 英特尔公司

地址 美国加利福尼亚州

[72] 发明人 D·M·卡米恩 J·克劳福特

审查员 齐 霁

[74] 专利代理机构 中国专利代理(香港)有限公司

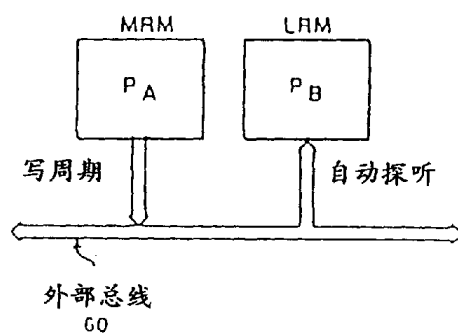
代理人 邹光新 陈景峻

权利要求书 1 页 说明书 9 页 附图 5 页

[54] 发明名称 具有减少功率操作特性的高速缓存一致性多道处理计算机系统

[57] 摘要

多处理器系统在操作的减少功率模式期间保持超高速缓存一致性。多处理器系统具有耦合到外部总线(60)上的第一处理器和第二处理器(20),用于利用主存储器执行数据事务处理。在操作的减少功率模式期间,第二处理器(20)的内部时钟(ICLK)与第二处理器(20)的部分内部逻辑去耦合,同时,内部时钟信号(CLK)保持耦合到第二处理器(20)的部分内部逻辑上,该部分内部逻辑被用于监视和响应外部总线(60)上的数据量以保持高速缓存的一致性。在操作的减少功率模式期间,第二处理器连续执行探听和写回处理,以维持高速缓存一致性多处理器系统。



ISSN 1008-4274

1. 一种多处理器计算机系统，包括：
总线；
耦合到所述总线上的主存储器；
- 5 耦合到所述总线上的第一处理器，用于利用所述主存储器执行数据事务处理；和
- 第二处理器，该第二处理器具有耦合到所述总线上的一个总线单元、具有相关标记阵列的局部高速缓存、用于产生时钟信号的锁相环 PLL 以及一个用于当总线单元、锁相环 PLL 和相关标记阵列保持与时
- 10 钟信号耦合时，通过使时钟信号和第二处理器的部分内部逻辑去耦合来减少功耗的装置；
- 所述第二处理器的总线单元还包括用于当处于操作的减少功率模式时探听所述总线的装置和用于向第一处理器传送指出所述第一处理器产生的在该总线上的写周期是针对在第二处理器的局部高速
- 15 缓存的高速缓存存储界中存储了一个修改后的数据复本的该主存储器的一个存储单元的装置。
2. 如权利要求 1 所述的多处理器计算机系统，其特征是所述第二处理器还包括一个用于在完成第一处理器的写周期之后将经过修改的高速缓存存储界写回到所述主存储器中的装置。
- 20 3. 如权利要求 2 所述的多处理器计算机系统，其特征是所述第一处理器还包括一个响应第一处理器的信号传送装置的装置，用于在第二处理器完成写回经过修改的高速缓存存储界之后，再次产生所述写周期。
4. 如权利要求 1 所述的多处理器计算机系统，其特征是所述去耦
- 25 装置包括一个微编码的暂停指令。
5. 如权利要求 1 所述的多处理器计算机系统，其特征是所述去耦装置包括一个外部引线，当被确认时，该外部引线使所述时钟信号和部分内部逻辑去耦。

具有减少功率操作特性的高速缓存一致性
多道处理计算机系统

5

发明领域

本发明通常涉及以多处理器为基础的计算机系统领域，特别是，涉及到具有调节功耗特性的多道处理器计算机系统。

10 半导体技术的进步使得能够压缩集成电路的个体尺寸以允许更多的晶体管被制造在一个单一的半导体基片上。例如，当前正在制造的大多数复杂微处理器通常都包括由数百万个晶体管构成的的一个单一的集成电路(IC)。虽然这些令人惊异的技术进步可以显著地增加当前现代化计算机系统的性能和数据处理能力，但是，这些进步增加了功耗。当然，功耗的增加意味着必须从所述 IC 中消散更多的热量。

15 由于过量的功耗和热量消除现在是计算机设计人员所面对的一个关键问题，所以，研制了各种功率节省技术以使计算机系统内的电源的电流电平最小化。很多这种技术都采用了在不处于使用状态时关闭多处理器电源以便节省能量的策略。但是，这种技术不是没有缺点的。

例如在使用用于协同完成系统任务的两个或多个处理器的多处理器(MP)计算机系统中有这样的问题。如果一个微处理器的电源被关闭(例如，由于它的当前任务已经完成或处于待用状态)，该系统中的另一个微处理器可以继续的系统总线上执行数据事务处理。问题是某些总线事务处理可能试图在电源关闭状态下或处于待用状态的微处理器中读/写以修改状态存储的数据。除非具有某种机构来监视总线活动和修正共享存储单元，否则，将丧失数据一致性。因此，MP 计算机系统需要一个机构，该机构使待用处理器发觉并响应于试图访问变坏了的数据进行访问的总线活动。

25 如将要看到的，本发明提供一种多道处理器计算机系统，在该系统中，各个处理器监视总线的数据量，以保持高速缓存的一致性，同时以减少功率的模式执行操作。根据本发明，一个待用或电源关闭的处理器30 通过将修改后的数据以减小功率的操作模式写回到系统总线来响应某

些总线事务处理。另外，本发明的实现没有等待时间或来自操作系统的干预。因此，本发明提供了与外部总线相互作用的完全透明的方式，同时，使功耗最小化。

发明概要

- 5 多处理器计算机系统被描述成包括一个被耦合到主存储器以及第一和第二处理器上的总线。第一和第二处理器被耦合到总线，以便利用所述主存储器执行数据处理。第二处理器具有一个被耦合到所述总线上的总线单元、一个具有相关的标记阵列的局部高速缓存器、一个用于产生时钟信号的锁相环(PLL)、和一个在总线单元、PLL和相关的标记阵列耦合到时钟信号上的时候，使时钟信号和第二微处理器的某些内部逻辑去耦合以减少功耗的装置。第二微处理器的总线单元还包括一个用于当以减少功率模式进行操作时，探听所述总线的装置和一个用于产生信号以向第一微处理器指出由第一微处理器在所述总线上使用的写周期是针对在第二微处理器的局部高速缓存器中的修改的高速缓存存储界的装置。
- 10
- 15

通过遵照和根据附图进行的详细描述可以更加完整地理解本发明。但是，这些描述并不构成对本发明的限制，所示特定实施例仅用于对本发明的解释和理解。

图1示出了根据本发明运行的一个微处理器的一般方框图。

- 20 图2示出了在本发明一个实施例中使用的时钟控制结构的状态转换。

图3是示出了本发明一个方面的操作的定时图。

图4A-4D示出了在本发明一个实施例中执行的各种例子的操作。

- 25 图5示出了根据本发明的一个测试寄存器，它包括允许软件禁止某些特性的多个比特。

- 本发明是一个多处理器计算机系统，用于在使功耗最小化的同时保持高速缓存的一致性。在下面的描述中，出现了很多诸如特定信号、协议、设备类型等的特殊名称，用于提供对本发明的透彻理解。但是，应当理解，这些特殊名称并不需要被用于去实践本发明。在其它的例子中，公知的结构、电路块和构造并没有详细示出，以避免使本发明显得含糊不清。
- 30

图1示出了微处理器20的方框图，该微处理器20包括了本发明的

各种特性。在一个实施例中，微处理器 20 包括一个由加州 Santa Clara 的 Intel 公司制造并在市场上可以买到的 PentiumTM(奔腾)处理器。虽然本发明将结合图 1 所示的实施例进行描述，但是，应当理解，本发明的整个构思可以适用于很多不同类型的计算机系统，包括与奔腾处理器兼容的 5 微处理器，或使用不同数据处理器体系结构的微处理器。

微处理器 20 包括锁相环(PLL)电路 30，该电路提供沿着线 46 耦合到集成电路各功能单元块的一个内部时钟信号(ICLK)。例如，图 1 出示在线 46 上出现的 ICLK 信号被耦合到所述 IC 的大多数内部逻辑上，这些内部逻辑包括指令高速缓存 25、数据高速缓存 26、控制 ROM21 和 10 由处理器的整数单元 22 及浮点(FP)单元 23 组成的内部逻辑。所述 ICLK 信号还被耦合到中断逻辑单元 29 的一部分上。指令高速缓存 25、数据高速缓存 26 和总线单元 40 中的每一个都被耦合到一个 64 比特数据总线和一个 32 比特的地址总线上。

除了 ICLK 信号以外，PLL 电路 30 还提供一个被沿着线 45 耦合到指令高速缓存 25 和数据高速缓存 26 的标记比较阵列的第二时钟信号 15 (CLK)。线 45 上的 CLK 信号还被耦合到部分中断逻辑单元 29 的一部分和总线单元 40 上。(下面将讨论 ICLK 和 CLK 时钟信号之间的区别。)在正常操作期间，微处理器 20 利用约 100MHz 的时钟(即：ICLK 和 CLK)频率进行操作。处理器 20 的变化也可以支持各种内核/总线比值。20 例如，在另外的一个例子中，可以支持 50MHz 和 60MHz 的总线频率。

在一个实施例中，总线单元 40 和中断逻辑单元 29 包括使微处理器 20 能够被用于一个 MP 系统中的逻辑电路。例如，由总线单元 40 发出和接收的总线周期以及由所述中断逻辑单元 29 接收和提供的中断信号与公知的 PentiumTM 处理器相兼容。微处理器 20 还包括一个用于保持 25 高速缓存一致性和对外部总线进行仲裁的逻辑。

在一个实施例中，微处理器 20 还包括功率管理特性，该特性允许所述处理器进入减少功耗状态(即：操作的 HALT 或 STANDBY 模式)。当在正常操作模式中运行于峰值功率时，微处理器 20 可能损耗的功率可达 20 瓦。而在操作的 HALL 或 STANDBY 模式时，功耗可以只有 700 30 毫瓦。微处理器 20 执行功率管理功能和操作系统独立功能的一种方式 是借助于操作的系统管理模式。系统管理模式(SMM)由一个中断(SMI)、交替地址空间和一个指令(SRET)组成。系统管理中断使一个系

统管理中断请求被锁存在逻辑单元 29 之内。当在指令边界上识别所述锁存 SM#时，微处理器 20 进入 SMM。

继续参考图 1，在一个实施例中，微处理器 20 包括一个外部引线(标记为 STPCLK#)和一个可以被用于调节所述微处理器功率的外部电路。所述 STPCLK#引线在申请日为 11/3/92 的共同未决专利申请 No 07/970,576、发明名称为“用于在微处理器中异步停止时钟的方法和装置”中有更加详细的描述，上述专利文献在这里一并作为参考，并且，该申请被指定给本发明的受让人。所述 STPCLK#引线提供一个敏感电平 SM1，该敏感电平 SM1 将所述处理器置于低功率 STANDBY 状态。根据本发明，在处于操作的 STANDBY 模式的同时，微处理器 20 响应专用(即：ADS#)和外部探听(即：EADS#)请求。

微处理器 20 还支持被称之为 AUTO-HALT 的一个机构 (mechanism)，每当 HALT 指令被执行时，上述机构将所述处理器置于功率减少的操作模式。微处理器 20 响应包括专用探听在内的所有停止中断事件和在处理器处于减少功率状态(没有相关的 STPCLK#的引线被确认)时产生的处理器之间的中断事件。

当微处理器 20 被用于正常工作模式时，时钟线 46(ICLK)和 45(CLK)被使能，从而使内核(internal core)时钟信号被提供给集成电路的所有单元。当处理器输入作为确认 STPCLK#引线或执行 HALT 指令结果的低功率或 STADBY 时，线 46 上的内核时钟信号(ICLK)被禁止。但是，应当理解，禁止线 46 不能够改变 PLL 电路 30 的连续操作。换言之，PLL30 连续在被耦合到微处理器某些部分的线 45 上产生内核 CLK 频率。保持运行(即：被 CLK 驱动)的微处理器 20 的这部分包括 PLL30、高速缓存单元 25 和 26 的标记比较阵列、中断逻辑单元 29 的一部分和总线单元 40。根据本发明，提供给微处理器 20 被选择这部分的功率允许处理器监视和响应为在 MP 系统中保持超高速缓存一致性所需的外部总线数据量。即：在操作的减少功率模式下，总线仲裁和高速缓存被有意地保持为活动状态。

现在参看图 2，该图示出了根据本发明一个实施例的某个关键部分 5 的特性状态图。在一般工作条件下，即在状态 51，微处理器 20 被激活以执行指令。这表示处理器的满功率状态，在这种状态下，ICLK 和 CLK 信号被耦合到 IC 内部逻辑的所有部分上。

由方框 53 表示的 STOP-GRANT(停止 - 允许)状态可以通过确认微处理器 20 的 STPCLK#引线进行输入。在 STOP-GRANT 状态中, 集成电路工作于减少功耗模式, 在这种模式下, 处理器的大多数内部功能单元都没有被激活(即: ICLK 断)。另一方面, 总线单元 40、超高速缓存器 25 和 26 的标记阵列、PLL30 和中断逻辑单元 29 的一部分保持被激活。一旦 STOP-GRANT 总线周期被放置在所述总线上且 BRDY#信号被返回, 处理器就处于所述 STOP-GRANT 状态。本技术领域内的专业技术人员将 BRDY#理解为它表示外部系统在数据引线上具有用于响应读出的预先传送的有效数据, 或它表示外部系统(例如, 主存储器、其它处理器等)已经接收了用于响应写请求的处理器数据。在一个实施例中, 在 STPCLK#解确认之后的约十个时钟周期内, 处理器返回到正常执行状态。RESET 也使所述处理器脱离 STOP-GRANT 状态, 返回到正常状态。

当处于操作的减少功率模式时, 通过监视总体通信量处理器 20 识别到总线 40 的输入信号以保持超高速缓存连续性(例如, 无效和外部探听)的输入。例如, 当处于 STOP-GRANT 状态时, 处理器锁存在外部中断信号(例如, SM1#、NM1、INTR、FLUSH#、R/S#和 INIT)上的事务。所有这些中断都是在 STPCLK#解除确认之后, 如当再进入正常状态时采用的。

只要当执行 HALT 指令时, 便进入由图 2 的方框 50 表示的 AUTO-HALT 状态。在 AUTO-HALT 状态下, 内部时钟(ICLK)与大部分内部逻辑去耦, 同时, 连续时钟信号 CLK 保持所述芯片中被选择功能单元的运行。根据 INTR、NM1、SM1#、RESET 或 INIT 的发生, 处理器事务回到正常工作状态, 即; 状态 51。如前面解释的, AUTO-HALT 状态通过停止将时钟信号加到处理器的大多数内部逻辑电路上而导致功率的有效减少。总线单元 40、PLL30、标记阵列和中断逻辑电路保持被激活以支持探听和允许快速重新开始。任何外部中断都可以使处理器脱离 AUTO-HALT 状态 50 并返回到正常工作状态 51。当工作于 AUTO-HALT 状态 50 时, FLUSH#信号(低电平有效)使齐平事件被锁存并如所示被方框 55 激活。对于所描述的实施例来讲, FLUSH#信号迫使处理器将所有的经过修改的存储界写回到数据高速缓存 26 中, 并使它的内部高速缓存失效。然后, 通过处理器指出写回操

作和失效操作的完成产生 FLUSH 确认特定信号。

R/S#输入(低电平有效)的确认使得处理器停止正常的执行,并将其置于 IDLE 状态。图 2 示出了一个使来自 AUTO-HALT 状态 50 的一个事务输入 PROBE-MODE 状态 56 的 R/S#事件。提供 R/S#引线以便和特殊调试端口一起用于调试所述处理器。R/S#引线从高到低的事务中断所述处理器,并使其在下一个指令边界处停止执行。R/S#引线的解确认使得来自 PROBE-MODE 状态 56 的事务返回到 AUTO-HALT 状态 50。

图 2 还包括状态 52, 该状态表示在确认 STPCLK#引线或执行 HALT 指令之后的情况下,当处于减少功率模式时,所述处理器的探听能力。即使在这两种情况之一中功耗被减少,处理器仍能够经过单元 40 连续驱动总线信号,处理器的内部机构状态被维持。专用和处理器之间的探听被支持以用于高速缓存失效和写回周期。如果所述探听需要运行写回周期,那么,功耗将在一个很短的周期内增加。根据本发明,仅利用硬件而不需要微码介入就能够完全对所述探听进行处理。

在 STOP-GRANT 和 AUTO-HALT 状态中,微处理器 20 通过保持由 CLK 使能的高速缓存单元标记比较逻辑支持所述探听-包括专用和处理器之间的探听。这示于图 1,其中,线 45 上的 CLK 信号被耦合到高速缓存 25 和 26 的标记比较逻辑上。另一方面, ICLK 信号被禁止到高速缓存以使功耗最小化。当探听发生时, MESI(即:经过修改的、排它的、共享的、有效的)高速缓存协议比特被修正成需要上升时的值。只有在需要写回周期时,除了标记阵列以外的处理器部分(例如,至少是数据高速缓存、相关的高速缓存控制逻辑和在数据高速缓存及总线接口之间的总线)被激活。在另外一个实施例中,在检测到一个可探听事项并随后提供稍微增加功率的状态以允许执行所述探听操作之前,通过例如使所有高速缓存的入口失效的禁止某个状态下处的标记比较逻辑电路或关掉标记阵列电源可以实现更加积极的功率节约。

注意,图 1-4 仅仅输出了本发明的一个解释性实施例,但是,比较复杂的实现手段可以提供包括不同时钟方案变化在内的功率节省技术。就所使用的特殊协议而言,一旦执行了探听,其实质内容是相同的。例如,第一处理器不是在系统总线上写回数据,而是第一处理器可以撤回写入并保持污线(dirty line)。由此,本技术领域内的专业技术人员可

以懂得执行本发明的很多种不同方式。

图3是一个时序图，它示出了 STPCLK#请求和 STOP-GRANT 总线周期之间的等待时间。注意，对于所示出的这个实施例来讲，在 STPCLK#请求和 STOP-GRANT 总线周期之间大约有 10 个时钟的延迟。这个等待时间取决于当前指令、CPU 写缓冲器内数据的量以及系统存储器的性能。

处理器 20 的数据高速缓存 26 利用 MESI 协议实现高速缓存的一致性。数据高速缓存中的存储界可以处于 Modified(经过修改的)、Exclusive(排它的)、Shared(共享的)或 Invalid(无效的)状态，同时，在指令高速缓存 25 中的存储界可以处于有效或无效状态。本发明直接用于可能出现的下述情况，即两个或多个处理器共享一个计算机系统内的公用数据。只要可能，所述处理器的局部高速缓存器都可以试图高速缓存数据。在处理器 20 的一个实施例中，包括一个专用高速缓存器一致机构，用于保证处理器之间数据的一致性。假如在两个处理器中的一个处理器内高速缓存了任一数据而另一个处理器试图对这个数据进行访问，那么，包含有所述数据的处理器将通知请求处理器它已经高速缓存了所述数据。超高速缓存存储界和包含有数据的处理器的状态将依据当前状态和另一个处理器已经形成的请求类型而变化。

根据本发明，基本的一致性机理要求所述处理器不运行周期、不占用总线(这里，称做最低最近总线主控器或 LRM)并不探听所有 MRM 总线活动(MRM 意为占用总线的最高最近总线主控器)。然后，运行总线周期的 MRM 处理器等待来自 LRM 处理器的一个表示数据被包含在 LRM 高速缓存中的指示。

为了更好地理解本发明，下面来考虑在图 4A-4D 中表示的任意接口交换的例子。首先，假设处理器工作于减少功率模式或未占用外部总线 60。这将使处理器 P_B 处于 AUTO-HALT 或 STOP-GRANT 状态。另外，假设处理器 P_A 只在外部总线 60 上运行一个写周期。再有，假设在该总线上的写周期是针对在处理器 P_B 中的被修改后(M)状态内的一个存储界的。这是在图 4A 中所示的情况。由于总线单元和处理器 P_B 中的内部中断及探听逻辑被保持激活，所以，不必考虑大多数其它内部逻辑电路被关断电源，处理器 P_B 就可以自动地探听外部总线 60，以便监视由处理器 P_A 启动的写周期。

图 4B 示出了处理器 P_B 向处理器 P_A 指出所述写事务已经命中一个修改后状态的超高速缓存存储界。这是通过确认 PHITM# 信号被耦合到处理器上进行的。处理器 P_B 也确认一个专用仲裁引线 BPBREQ#, 用于指出处理器 P_B 已经产生了一个总线请求(在这个例子中, 假设处理器 P_A 当前占用外部总线 60)。注意, HITM# 信号也被耦合到总线 60, 以禁止另一个总线主控器在所述存储界被完全写回之后对所述数据进行访问。然后, 处理器 P_A 就如同不存在处理器 P_B 一样完成在外部总线 60 上的写周期。

但是, 在处理器 P_B 有机会将修改后的数据写回到系统存储器之前, 只有在外部总线 60 上完成所述写周期时才发生所述外部探听。图 4B 示出了处理器 P_B 确认 HITM# 信号, 以便通知系统探听地址已经被高速缓存在双处理器对中, 且该地址处于修改后的状态。在这个例子中的外部探听命中了与使 PHITM# 信号被确认的同一个存储界。

在图 4C 中, 在外部总线 60 上发生一个仲裁交换, 并且现在处理器 P_B 占用该总线。此时, 处理器 P_B 写回所述 M 状态存储界。从计算机系统的观点来看, 似乎是使用一个单一的处理器来完成探听事务。注意, 在图 4C 中, 在两个处理器之间耦合的两个专用仲裁引线被用于指出总线所有权被授予处理器 P_B (或者是在完成写回操作之后处理器 P_A 请求得回所有权)。

最后, 在图 4D 中, 在处理器 P_B 授权总线返回到处理器 P_A 之后, 处理器 P_A 再次运行原始写周期。认识下述一点是非常重要的, 即; 在由图 4A-4D 表示的全部探听和写回操作中, 处理器 P_B 一直保持的操作的减少功耗模式下。除了对高速缓存一致性有要求以外, 由于在由功耗要求限定的计算机系统中, 上述特性能够提供极大的优点, 所以, 它是本发明的一个关键特性。

现在参看图 5, 这里示出了一个特殊的测试寄存器 12, 该寄存器 12 包括多个比特, 以允许软件禁止微处理器 20 的某些特性。例如, 可以禁止 AUTO-HALT 特性以将寄存器 12 中的比特 6 设置成“1”。在这个设置中, 执行 HALT 指令不禁止所述内部时钟(ICLK)被耦合到处理器任何一个功能单元上。在一个实施例中, 利用缺省使能 AUTO-HALT 特性, 即: 在 RESET 之后, 寄存器 12 中的比特 6 被置成“0”。

可由软件通过测试寄存器 12 控制的其它特性包括用于多处理器系

统的可予程编中断控制(APIC)。在本发明的一个实施例中,所述处理器插入了一个先行处理器 SMI 控制器,该控制器支持在一个复杂的多处理器环境中的中断以及在一个简单的单处理器环境中的中断。局部中断控制器模块在三线串联总线上与 I/O APIC 模块(诸如由 Intel 公司制造的部件 no.8259A)通信。当寄存器 12 中的比特 4 被设置成“1”时, APIC 特性被完全禁止。这意味着所述 APIC 电路不能够传送或接收任一处理器的中断。从 APIC 寄存器空间的读出或向该寄存器空间的写入都要经过外部总线。当寄存器 12 的比特 5 被设定时,双处理器仲裁硬件也可以被禁止。当这个比特被设定为“1”时,处理器刚一变成最高最近总线主控器(MRM),专用双处理器特性(例如, PHIT#、PHITM#、PBREQ#和 PBGRNT#引线)就被禁止。如果这个比特别连续设置成“0”,那么, DP 特性被重新使能。可以由寄存器 12 使能/禁止的其它特性包括 SMM 输入和输出消息(比特 7),和快速执行跟踪消息(比特 8)。

15

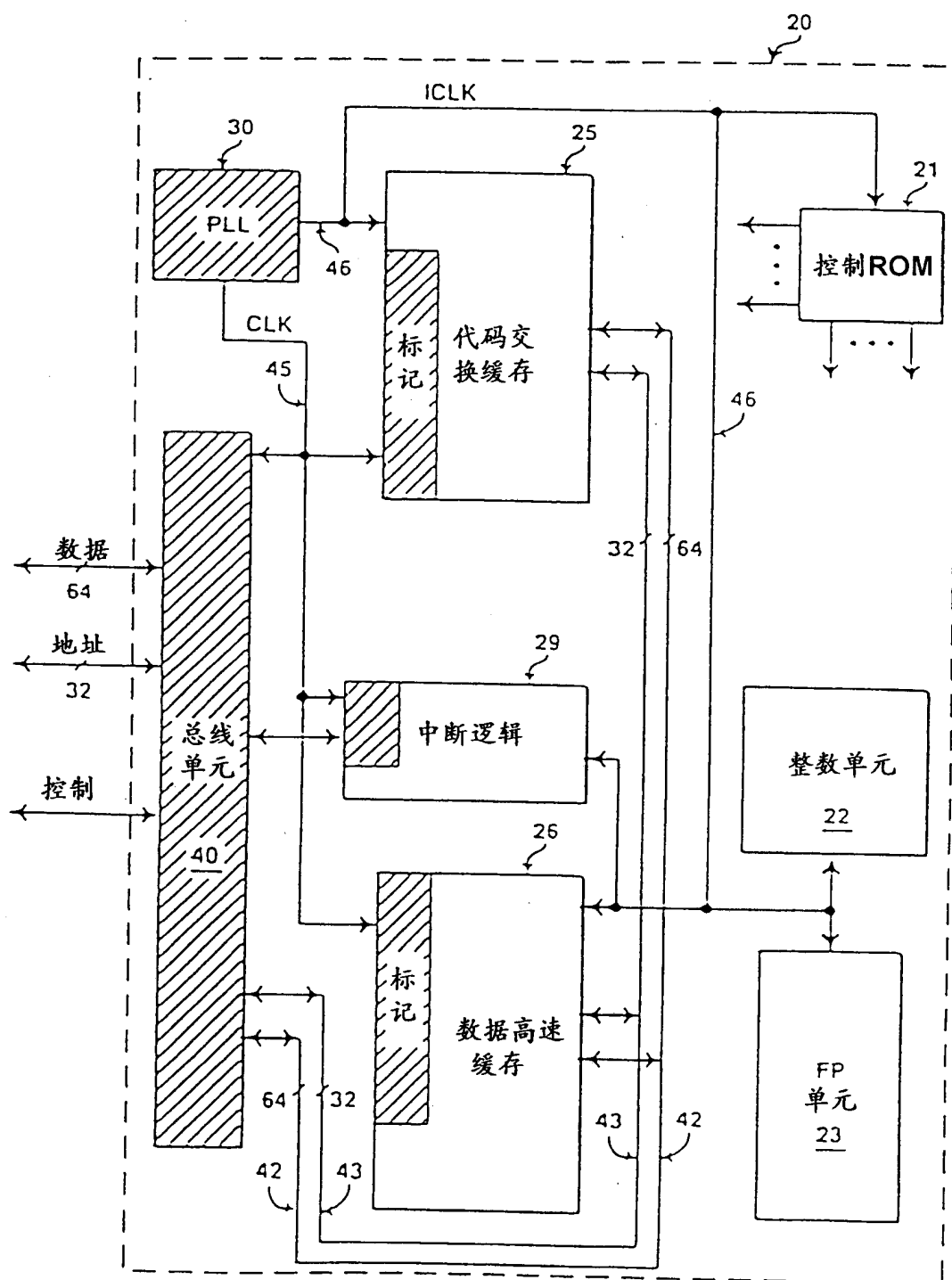


图 1

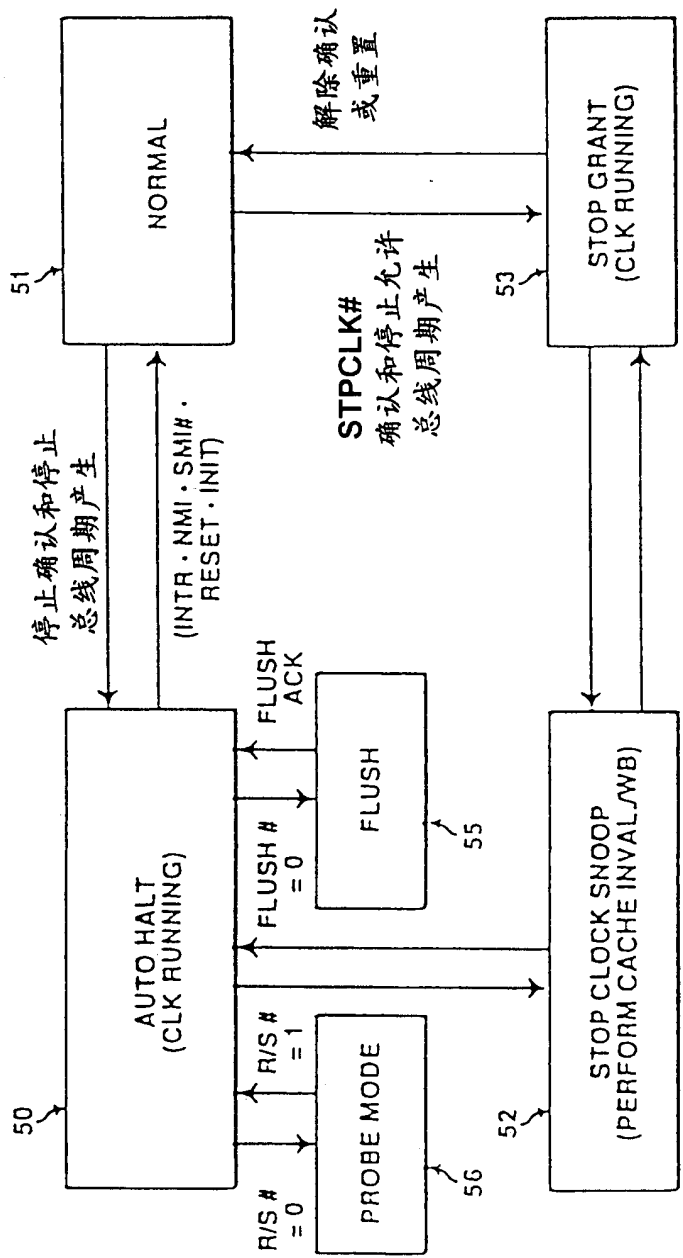


图 2

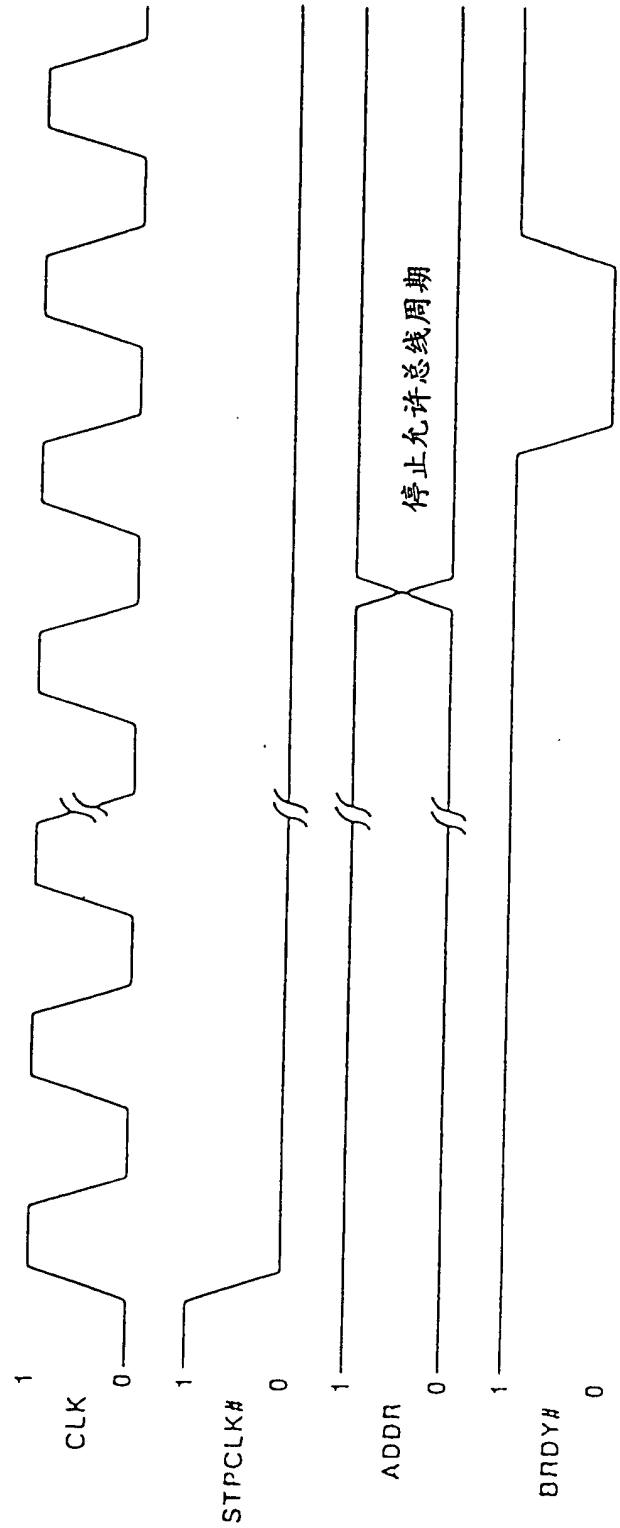


图 3

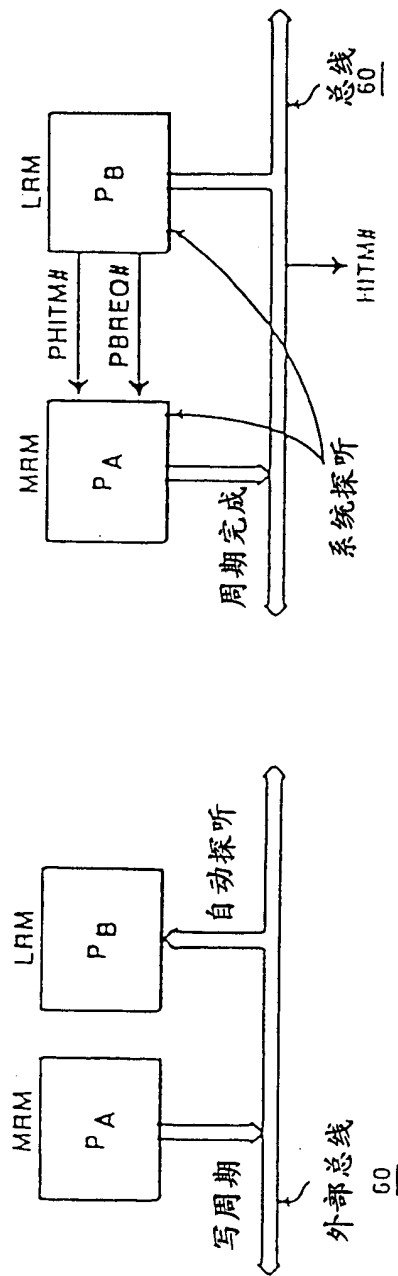


图 4 a

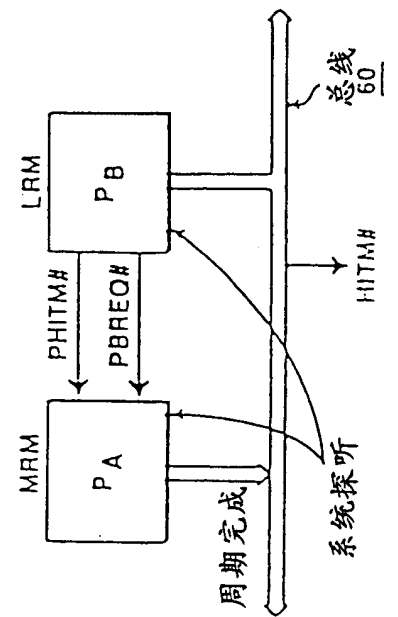


图 4 b

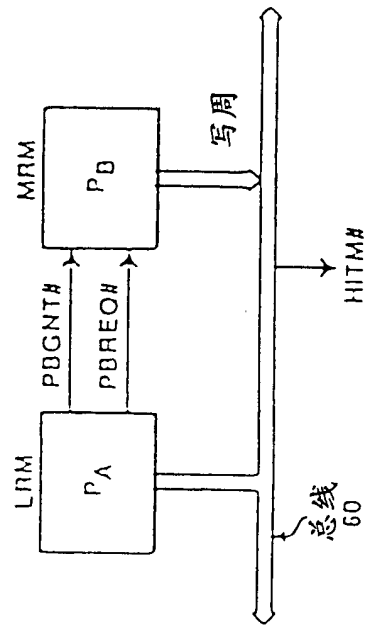


图 4 c

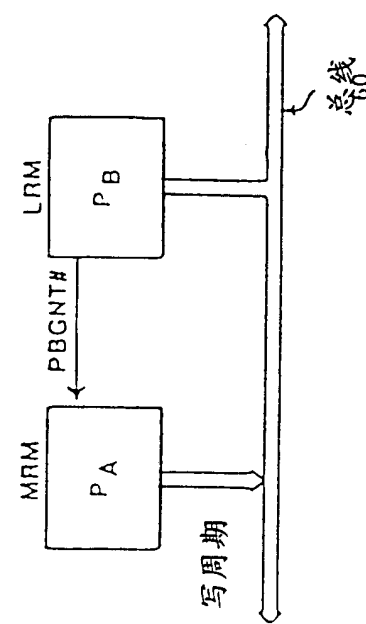


图 4 d

