[19]中华人民共和国专利局

[11] 公开号 CN 1104785A

[12] 发明专利申请公开说明书

[21]申请号 94105517.5

[51]Int.Cl[6]

[43]公开日 1995 年 7 月 5 日

G06F 11/00

[22]申请日 94.5.16

[30]优先权

[32]93.5.17 [33]US[31]063,380

[71]申请人 美国电话电报公司

地址 美国纽约

[72]发明人 迈克埃·L·拉森

韦恩·R·威尔科克斯

[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

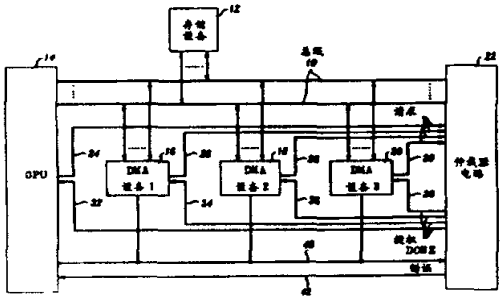
代理人 范本国

说明书页数: 附图页数:

[54]发明名称 考虑历史反馈及错误检测与修正的动态可编程总线仲裁器

[57]摘要

一种使用唯一的历史寄存器的仲裁电路，该历史寄存器和代表总线请求的一个值结合起来以索引一个表。历史寄存器请求的所有可能组合和相对应的授权一起存储在该表中。表的一个块被该历史寄存器选择，然后利用该请求对该块寻址以确定哪一个请求接收授权。该授权被移位进历史寄存器。更有益地，多于一个的表可被存储在存储器里，各表可由仲裁器控制器选择。



权 利 要 求 书

1. 一种为多个请求者的请求提供分配仲裁的系统，这种仲裁是以对这些请求者的请求所实施的先前授权的历史为基础的，所述系统包括：

从所述多个请求者中接收请求的输入装置；

所定义的多个授权的存储装置，所述的多个授权由对各种(当前)请求和历史请求可能组合所形成的(各个)授权所组成；

记录一个或多个先前授权的历史装置；

把所述的历史装置、所述的输入装置和所述的存储装置连接起来的控制装置，以把按所述输入装置接收的所述请求和按所述历史装置得到的一个或多个先前授权应用到所述存储装置中，从而在所述多个授权里选择一个应答的授权，并且对所述请求者中的一个提供反映所述选定授权的授权指示。

2．按照权利要求1的系统，其中所述历史装置由一个先进先出(*FIFO*)队列组成，而且其中的各个授权通过所述的*FIFO*队列传送。

3. 按照权利要求2的系统，其中所述存储装置由多个块组成，所述多个块的各个块包括一个所述多个授权的子集，其中所述控制装置通过将所述*FIFO*队列作为一个地址来选择所述块中

的一个块，还通过利用所述请求去选择所述选定块中的一个单元以从所述选定块中选择所述授权中的一个授权。

4. 按照权利要求 2 的系统，其还包括一个可写屏蔽寄存器，而且其中所述屏蔽寄存器选择性地改变所述历史方式的内容区段。

5. 一种为多个请求者的总线分配请求提供仲裁的系统，所述系统包括：

从所述多个请求者中接收请求的输入装置；

包含有多个查找表的存储装置，每个所述查找表具有多个授权，所述各个授权中的每一个和可能的请求输入中的一个相对应。

把所述的输出装置，所述的输入装置和所述的查找表连系起来的控制装置，以选择所述多个查找表中的一个表，并把所述请求应用到所述查找表以选择所述多个授权中的一个，而且对所述请求者中的一个提供反映所述选定授权的授权特许信号。

6．按照权利要求 5 的系统，还包括一个记录先前总线请求授权历史的历史寄存器，所述历史寄存器和所述控制装置相连接，从而所述控制装置也把从所述历史寄存器得到的所述先前总线请求授权应用到含有按所述输入装置得到的所述请求的所述查找表上以从所述多个授权中选择一个授权。

7. 按照权利要求 5 的系统，其所述多个查找表中的一个表或多个表可被动态地更新。

8．按照权利要求 6 的系统，还包括和所述历史寄存器连接的

屏蔽寄存器，所述屏蔽寄存器是动态可写的并且能够重写历史寄存器从而改变所述已规定的仲裁状态。

9. 按照权利要求 5 的系统，当由所述控制装置提供所述授权信号时，所述控制装置还进行差错检验。

10. 按照权利要求 9 的系统，其所述表的各条项目包括奇偶性指示，并且所述差错检验包括对所述授权的奇偶性检验。

11. 按照权利要求 9 的系统，其中所述表包括一个循环冗余检验值，并且所述差错检验包括进行循环冗余检验。

12. 按照权利要求 5 的系统，其中所述表包含在一个随机存取存储器(*RAM*)里。

13. 按照权利要求 5 的系统，其中所述表包含在一个只读存储器(*ROM*)里。

14. 按照权利要求 5 的系统，其中所述表中的一个放在 *ROM* 中，而其余的所述多个表放在 *RAM* 里。

15. 按照权利要求 14 的系统，其中所述控制装置进行差错检验，作为对所述控制装置测出出现一个错误的反应，所述控制装置把所述确定的仲裁状态单独地应用于所述 *ROM* 中的所述表。

16. 一种以先前总线请求授权的历史为基地的对多个请求者的请求提供总线分配仲裁的方法，所述方法包括：

接收多个请求为所述总线使用；

为响应所述多个请求，把历史寄存器应用于所述多个请求以

确定一种仲裁状态；

把所述结果用作为索引进入仲裁状态表，所述仲裁状态表由多个仲裁状态及相对应的多个授权所组成；和

把按所述确定仲裁状态索引到的所述多个授权中的一个授权发送给所述请求者。

17. 按照权利要求 16 的方法，还包括，在发送所述授权之前，对所述授权进行差错检验。

18. 按照权利要求 17 的方法，其中所述差错检验包含奇偶性检验。

19. 按照权利要求 17 的方法，其中所述差错检验包含循环冗余检验。

20. 按照权利要求 16 的方法，其中所述仲裁状态表包含多个状态表，其中所述结果寻址到所述多个仲裁状态表中的一个仲裁状态表，并且其中为对确定已出现错误作出反应，使用一个和所述仲裁状态表不同的仲裁状态表。

21. 按照权利要求 16 的方法，其中所述仲裁状态表包括多个仲裁状态表，所述方法进而包括确定使用所述多个仲裁状态表中的那一个仲裁状态表。

22. 按照权利要求 16 的方法，其中所述历史寄存器是动态可变的，并且其中把历史寄存器应用于所述请求也包括改变所述历史寄存器。

23. 一种为多个请求者的请求提供分配仲裁的方法，包括：

接收多个请求；

在所述接收的请求以及一个或多个先前请求的基础上确定一个请求授权；而且

根据所述确定的授权允许所述多个请求者中的一个请求者。

说　　明　　书

考虑历史反馈及错误检测与修正的动态可编程总线仲裁器

本发明涉及计算机总线仲裁领域，更具体地，涉及考虑先前总线使用及提供可供选择总线仲裁算法的总线仲裁领域。

随着更多的设备和计算机系统的主存储器总线相连，带来关于总线的更多争用。为响应这种争用，已提出许多不同的仲裁电路以便对总线，进而对存储器资源，进行公平的或优先的访问。

例如，1993 年 3 月 16 日 *G. B. Marenin* 等提出的美国 5,195,185 号专利公开一种仲裁设备，它在优先级模式的基础上确定总线访问，并且还具有一种可动态地改变任何设备优先级的系统。但是，这个系统不提供基本仲裁模式上的灵活性；它只对设备自身优先级的提高进行响应。它也不考虑那个设备先前曾用过该总线以便给其它的用户以均等或优先。最后，该系统锁定在优先模式上，它不能够在适当时从优选模式转化为均等模式。

更早的总线优先级模式是在 1990 年 12 月 18 日由 *Makris* 等发表的美国 4,979,100 号专利中描述的。*Makris* 公开一种仲裁器电路，它对各个处理单元提供二个不同级别的访问总线的优先级并保

持对不同优先级别的相对访问时间比率的跟踪。这种系统通过较高优先级处理单元提供对较低优先级总线使用的中断。这种系统需要复杂的系统开销以跟踪这种总线的使用率，而且它不响应任一系统的先前的总线使用率。

这两种系统都不提供灵活的、动态可改变的仲裁系统，这种灵活、动态可改变的仲裁系统可从已知状态启动而且当条件改变时可从一种仲裁模式变化到另一种仲裁模式，这两种系统也都不对过去的分布起反应。因此，技术上的一个问题是，还没有一种能在授权选择中考虑历史反馈以提供动态优先级模式的总线使用仲裁器。

通过一种仲裁电路这个问题得到解决并且获得了技术上的进展，这种仲裁电路把先前的总线访问计及为判定的一部分以为请求者服务。具体地，它提供一个独特的历史寄存器，该寄存器产生一个体现过去总线请求的值，这个值和一个体现当前总线请求的值结合起来来访问存储器的一个单元。对历史请求和现在请求的各种可能组合定义了一个潜在授权并把潜在授权存储在存储器的各个单元上。潜在授权的选择是这样达到的，即通过把历史寄存器用作为存储器地址的高位位和把当前请求用作为存储器地址的低位位来选择存储器的一部分以确定被裁决的一种授权输出。同时该授权还移位到历史寄存器里。更好地，可把多于一个的表存储到存储器以可选择性地提供多种算法。

图 1 是一个计算机系统的方块图，其中包括一个根据本发明示

范性实施例的仲裁器电路；

图 2 是图 1 中所示仲裁器电路的方块图；

图 3 为图 2 仲裁器电路的状态图；

图 4 为图 2 仲裁器电路的时序图；

图 5 是历史寄存器的一个说明；

图 6 是对本发明示范性实施例中历史寄存器初始化时的状态的一个说明；

图 7 是一个存储器块例子说明基于请求的授权分配；以及

图 8—12 说明按照本发明的示范性实施例历史寄存器里的数据移位。

图 1 是说明本发明典型实施例的方块图，在这里是一个包括总线 10 和存储设备 12 的计算机系统 5。出于描述本发明最佳实施例的目的，有 4 个设备共享总线 10 和存储器 12。中央处理机（*CPU*）是一个通用 *CPU*，正如技术上周知地那样它对存储在存储器 12 的数据进行操纵并执行其 它 功能。在本示范性实施例中，有 3 个直接存储器存取设备（*DMA*）16，18 和 20，它们也和总线 10 相连。通过总线 10*DMA* 设备 16，18 和 20 直接读和写存储器 12 里的单元。为了防止多于一个的设备同时访问总线 10，按照本发明的典型实施例，仲裁器电路 22 控制对总线 10 以及从 而对存储器 12 的访问。仲裁器电路不限止于对存储器的访问，通常地还可控制总线访问，或者控制对任何设备的访问。

当设备 14，16,18 和 20 中的一个想要访问总线 10(和/或存储器 12)时，它分别在请求引线 24,26,28 和 30 上提出一个请求。如后面将要说明的那样，仲裁器电路 22 确定哪一个设备的请求将被授权，并相应地在授权引线 32,34,36 或 38 中的一个上产生授权信号。在本最佳实施例中，在某一时刻仅发出一个授权。但是，从技术熟练的人的角度这种系 统可以想象能发出多种授权。当该设备结束它对总线访问时，在引线 40 上发出一个"*done*"信号。如同在下面将进一步描述的那样，如果在总线仲裁期间检测出错误，仲裁器电路通过错误引线 42 通知 *CPU*14。

现转向图 2，图中展示了根据本发明典型实施例的一个仲裁器电路 22。本典型实施例的仲裁器 22 包括一个仲裁器控制器 102，一个仲裁器存储器 104，其主要是 *RAM*(随机存取存储器)但还包括一个 *ROM*(只读存储器) 块 105，一个授权锁存器 106，历史寄存器 108，请求锁存器 110，初始化设备(本发明最佳实施例中包括 *ROM*) 112，算法选择寄存器 114，屏蔽寄存器 116 和表选择寄存器 118。源于设备 14,16,18 和 20(图 1) 在请求引线 24,26,28 和 30 上传输的用来控制总线 10 的请求到达请求锁存器 110。本典型实施例中，这些请求的到达把请求锁存器触发到把所有的请求引线相" 或"并且在 *ARBREQ* 引线 120 上发出一个指示控制器 102 有一总线请求待决的信号。作为响应，控制器 102 通过负载线 122 发出一个信号把请求锁存器 110 里的请求锁存起来。此时，该请求通过总线 124 送

到存储器104。接着来自历史寄存器的数据可在历史总线126上得到，历史寄存器中数据的生成将在后面说明。

总线124上请求位所形成的值和由历史总线126上历史寄存器的位所形成 的值被结合起来并被用来作为输出下一个授权的存储器104中的一个地址。历史寄存器值和请求锁存器值的各种可能的组合和按照选 定 的 仲裁算法所决定的关联授权一起被存储在存储器104里。这样，通过确定以前授权和当前请求的那种组合要求下一个授权可以很简单地实行一种算法。以这种方式，任何算法都可被用来选择授权。在下文中参照图5—12叙述了一个例子。更好地，来自历史寄存器108的值被用来选择存储器104的一个节或一个块，该值组成该存储地址的高位位；而由锁存在请求锁存器110中的请求所形成的值组成该存储地址的低位位，从概念上通过对它的变址该值被用来访问这个块。存储器104在授权总线128上返回按照这个地址所选择的授权。

存储器104的一部分组成*ROM*105。*ROM*105至少含有一种构成历史寄存器108和请求锁存器110所有可能状态的算法。另外，*ROM*可以是小的并构成一个由请求锁存器110的4个位寻址的授权算法。无论何时控制器102检测出错误，*ROM*105都能被使用，以便进入一个已知的状态，或者，作为可选择的，进行初始化。

存储器104把选定的授权输出放到授权总线128上并输送给控制器102，历史寄存器108和授权锁存器106。输出由所断言的N个

位(在本典型实施例中，$N=4$）中的 1 组成，其定义选定的授权。当控制器 102 在 *ARBREQ* 线路 120 上接收该仲裁请求时，它在线路 130 上提供一个授权锁存器允许信号以让该授权可被传送给授权线 32,34,36 和 38 上。如后面进一步说明的那样，控制器 102 还断言移位线路 132 以使历史寄存器并行地把授权总线 128 的内容移位进历史寄存器 108，控制器还把最旧的先前授权移位出去。当接收该授权的设备完成它的存储访问时，它断言"*done*"线路 40 并由控制器 102 接收，接着开始新的周期。

屏蔽寄存器 116 是可从总线 10 的数据部分写入的。屏蔽寄存器 116 的值用作把通过历史寄存器 108 的某特定位的移位屏蔽掉。这具有把先前被断言过的授权的历史擦除掉的作用。为了改变已安装的算法，屏蔽寄存器 116 可被任何设备写。更有利地，存储器 104 包含多于一种的仲裁算法。屏蔽寄存器 116 还和表选择寄存器 118 相连。为了选择存储器里含有所需算法的那一部分，表选择寄存器 118 可通过屏蔽寄存器 116 由总线 10 数据部分上的数据来更新。一个表可被布置为优先级模式而另一个表可被布置为均等模式，这样提供了在任何给定时间某种类型的模式可被使用的灵活性。

本发明的最佳实施例还包括一个独特的错误检测机制。可供选择地，存储器 104 里的各个授权可具有和它关联的，如图 7 所示的，循环冗余校验（*CRC*）值、奇偶性值或其它数学错误检查值。这个值通过授权总线 128 和授权一起发送给控制器 102。接着控制器

102 可执行必要的检验并把它和在存储器 104 里得到的值进行比较。如果错误被发现，可在错误线路 42 上发出一个错误信号，并把它发送给 *CPU*14 以作进一步的反应(图 1)。

当初始化请求从复位线路 134 上进入并在仲裁器初始化(设备) 112 上接收时仲裁器电路 22 被初始化。经过复位信号线路(图中为清楚起见没有表示) 仲裁器初始化(设备)112 把复位信号发送给控制器 102，存储器 104，历史寄存器 108，授权锁存器 106，屏蔽寄存器 116 和请求锁存器 110。作为初始化的一部分，控制器 102 使存储在仲裁器初始化 *ROM*112 中的指定分配算法经过线负载控制总线 136 装入到存储器 104 里。被装入的数据内容是由算法选择寄存器 114 控制的。选择寄存器 114 是由 *CPU*14(图 1) 通过总线 10 的地址线路和数据线路加载的。总线 138 把一个索引提供给仲裁器初始化(*ROM*)112，以控制哪一个仲裁算法要被卸载到操作存储器 104 中。

现转到图 3，图 2 中控制器 102 中的一个状态机被显示出来，它描述不同的状态以及描述通向和离开控制器 102 的各条引线的结果电平。当初始化时进入空闲状态 310， 只要请求锁存器 140(图 2)上不出现请求，控制器 102 总保持在空闲状态 310。当接到一个或更多的请求时，状态机 300 在下一个时钟周期内从空闲状态 310 转换到状态 *S*1 320。在 *S*1 状态，如在下面图 4 中所说明的那样，控制器使得负载信号进入低电平，控制器还引起授权允许线路 130 被断

言。

然后状态机 300 在下一个时钟周期内从 *S*1 状态 320 转换到 *S*2 状态 330。在"*done*"信号被断言以前状态机 300 一直保持在 *S*2 状态 330 上。当"*done*"信号被断言时，状态机 300 从状态 *S*2 330 转换到状态 *S*3 340，在那时负载信号被断言并且"*done*"信号被释放后授权允许信号也被释放。只要"*done*"信号被断言状态机 300 总保持在状态 *S*3 340 上。如果"*done*"信号被释放而且 *ARBREQ* 引线被断言，状态机转换回到 *S*1 状态 320，这是因为接收到另一个仲裁请求，或者相反地这是因为当请求引线第一次断言时存在着多个仲裁请求。当"*done*" 和请求引线都不再被断言时状态机也从 *S*3 状态 340 转换回到空闲状态 310。

现转到图 4，其为图 3 所示状态机 300 的时序图，在图中表示状态机完全围绕状态机 300 的外部转换。仲裁器电路 22 通过时钟总线 140(图 2)接收时钟信号。在空闲状态的一些时点，接收到一个或更多的请求，状态机从空闲状态转换到 *S*1 状态。控制器断言移位线路 122 并把负载线路置为低电平。在这时，被锁存的请求和历史寄存器的内容被发送给存储器 104，而且授权被生成并放到授权线 128 上，造成线路变为高电平。同时，它断言授权锁存线路 130 以把授权输出对线路 32，34，36 和 38(图 1 和图 2)闭锁。在 *S*1 状态的一个时钟周期之后，状态转换到 *S*2 状态，在这个状态，为响应授权，*ARBREQ* 引线 120 变为低电平并且控制器断言移位引线 132，该断言

时间为一个时钟周期。然后控制器在状态 *S*2　330 下等待直到“*done*”信号被断言为止，以指明选定的设备已完成操作，在这个时刻，转换到状态 *S*3　340，在 *S*3 状态下负载信号被断言在 *S*3 状态中的某一瞬间“*done*”信号变为低电位。对此响应，控制器把授权锁存信号置为低电平并且转换回到空闲状态。

可供选择地，在存储器 120 里对每个授权伴随着一个附加字段，如图 7 所示，其表示接收授权的设备在断言 *done* 线路 40 之前所要使用的周期数。控制器 102 利用这个信息对时钟周期计数（从时钟线路 140 上）并且在 *done* 信号被断言前稍许一点开始下一个状态。以这种方式，按照本发明典型实施例的仲裁电路可被做成流水线布局。

历史寄存器和存储器的操作现在和图 5—12 联系起来给予说明。在本示范性实施例中，历史寄存器 500（图 5）由 20 位组成，这 20 位每按 4 位又分成 5 组。四位的四种置值中的每一种代表一个先前授权。在图 5 中，*H*0 是最后的授权。*H*1 是 *H*0 前面的授权，*H*2 是 *H*1 前面的授权，而 *H*3 是 *H*2 之前的授权。当作出一个授权时，它被移位进 *H*0。而 *H*0 被移位进 *H*1，依次类推，并且 *H*3 移位出历史寄存器。最后的 4 位是来自请求锁存器的“请求位”。它们是索引位。

图 5 中的历史寄存器用来对含有授权位的存储器块定址。“索引”位直接地映射在请求锁存器 10 里找到的设备请求位。在本讨论中，假定每次在历史寄存器的各个先前授权里只有一个位被置位，而

且该模式中所有的不可寻址的地址都产生一个出错信号。在 *H0*—*H3* 里，每一组中只有一个位可被设置，因为在本典型实施例中每次只给出一个授权。可供选择地，这 4 个寄存器(*H0*—*H3*)还可被编码以节省地址空间并使 *RAM* 变小。

在操作中，请求进入系统(通过设备 14，16，18 和 20)并产生对存储器 104 的一个地址，其在图 6 中表示为"*xxxx*"。这个地址包含 16 个可能的项目。该地址识别 *RAM* 中含有 4 位授权的一个单元。这些授权是根据例如图 7 中表示的特定的算法确定的。图 7 表示一个含有简单优选权授权算法的存储器块；但是，仲裁模式不必一定为所示的"优先"模式。还请注意所有的请求引线的变化都被覆盖。各种变化都可使 *G3* 到 *G0* 的一个授权位被设置(选择的)而其它授权位被置为 0(未选择的)。

存储器节的输出被仲裁器用来授权一个请求(放在图 1 的授权线路 32，34，36 和 38 上) 并且还被装入 *H0* 的地址部分。假设请求位为〔1001〕，按照图 7 里的表，它产生设置为〔0001〕的授权 0。在下个周期 *RAM* 地址按图 8 所示出现。

历史寄存器地址还编址为 *RAM* 中的一个 16 个单元块(但不是如图 7 中所示的相同块)。通过地址的请求部分选择 16 个单元中的一个。16 个单元中 的数据是具有把请求 0 约定为在最后周期内授权的环形签名式算法的一个例子。它意味着如果有任何其它的请求是待决的，它们将在请求 0 之前被对待。在这种情况下下一个授权

将在优先级的基础上从其它的请求中选择，尽管情况不必总是这样。

假设下一个周期的请求位模式仍旧为〔1001〕。它的结果是在这个周期内给出请求3。在下一个周期上*RAM*地址如图9所示。这是通过*H0—H3*地址部分的地址所选择的一个新的16单元部分。该部分包含有关请求引线的任何可能变化的授权。在本示范性实施例中采用环形签名式算法并约定请求0和3已被授权的情况下，选择新的授权位。如果下个周期的请求引线为〔1010〕，*RAM*将响应并授权1，因为授权3已被授权而授权1还没有被授权。在下一个周期上*RAM*地址将如图10所示。

因为请求0，1和3已被先期授权，这些16个单元被选择。注意请求3被断言但未使，在下一个周期里，请求地址为〔1100〕。因为请求2还未被对待，*RAM*以授权2〔0100〕作为响应。再次注意由于环形签名式算法请求3未被使用。

在下一周期*RAM*的新地址如图11所示。所有的请求被使用，如果下个周期的请求地址是〔1011〕，结果将是授权0，因为它是最早被使用的。新的*RAM*地址则如图12所示。

*RAM*的各部分构成一集单元，该集按照环形签名式算法定义授权位的各种可能状态。所使用的算法由系统设计员决定。此外，有可能创立反映运用仲裁器的计算机的特殊性的特定算法。简言之，在使用同一硬件下，根据需要仲裁器可以是通用的或者是专用的。

应该理解上述的实施例仅用于说明本发明的原则，在不违背本

发明范围的情况下，技术熟练的人可设计多种变型。因而，这意味着这些变型被包括在权利要求书的范围之内。

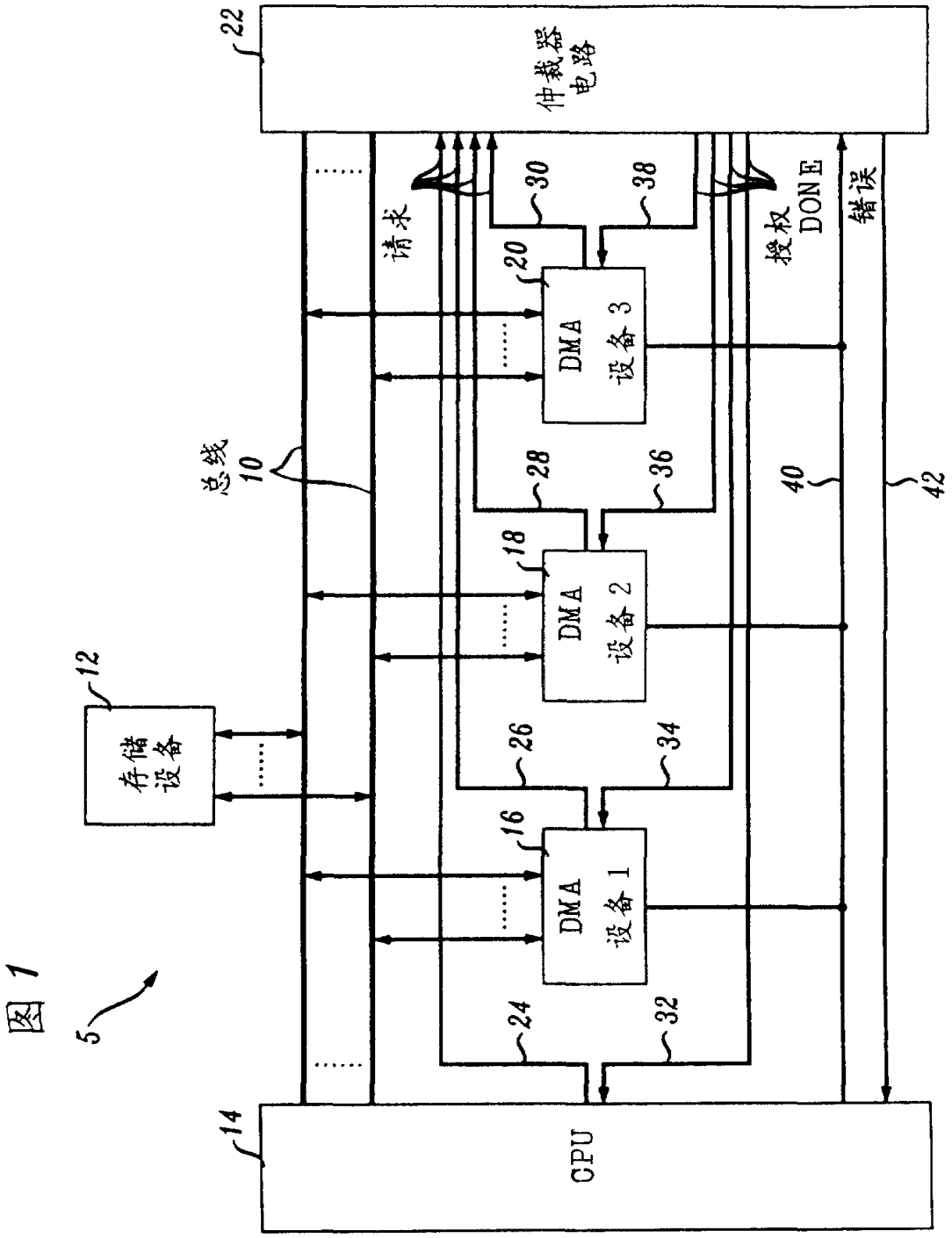

22
仲裁器电路
请求
30
38
授权
DONE
错误
20
DMA 设备 3
总线
10
28
36
40
42
18
DMA 设备 2
12
存储设备
26
34
16
DMA 设备 1
24
32
14
CPU

图 1

5

图 2

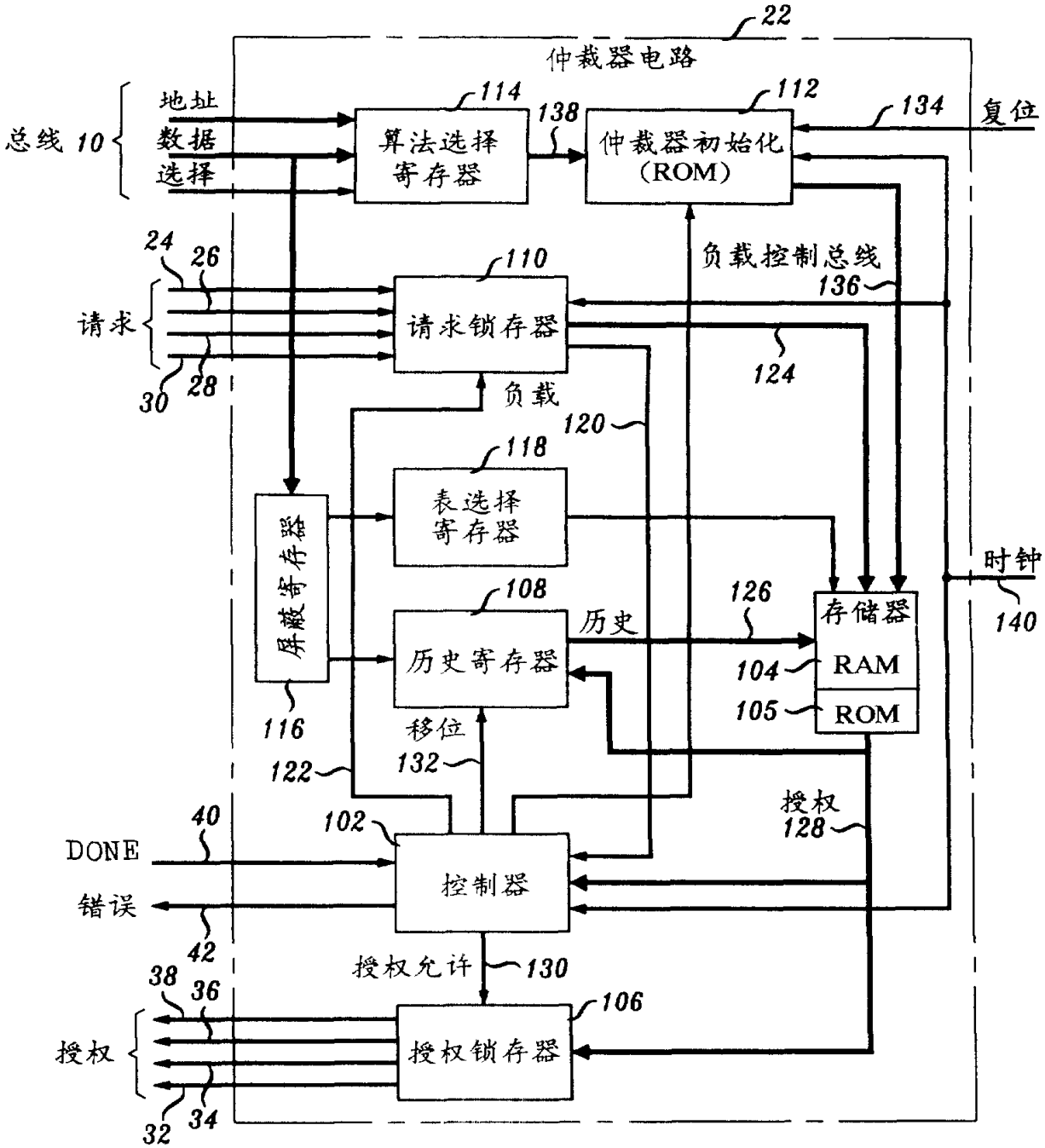

22
仲裁器电路
总线 10
地址
数据
选择
114
算法选择寄存器
138
112
仲裁器初始化 (ROM)
134
复位
24
26
请求
30
28
110
请求锁存器
负载控制总线
136
124
负载
120
118
表选择寄存器
屏蔽寄存器
116
108
历史寄存器
历史
126
存储器
104
RAM
105
ROM
时钟
140
移位
132
122
授权
128
40
DONE
102
控制器
错误
42
授权允许
130
38
36
授权
106
授权锁存器
32
34

图 3

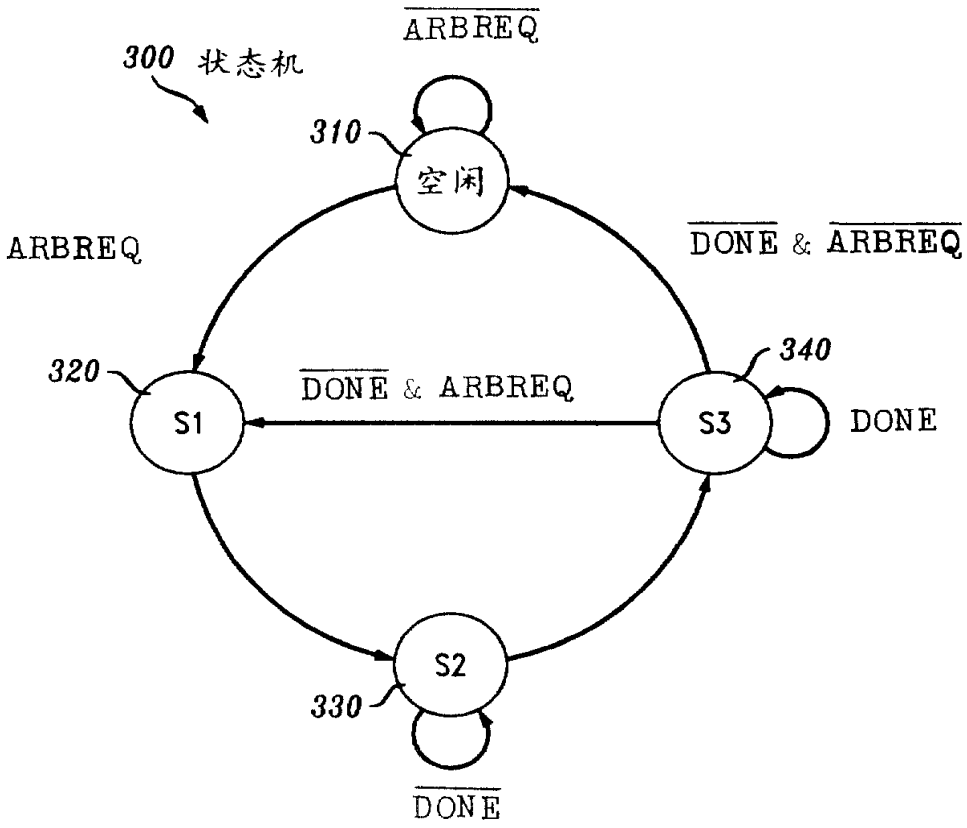

ARBREQ
300 状态机
310
空闲
ARBREQ
DONE & ARBREQ
320
340
DONE & ARBREQ
S1
S3
DONE
S2
330
DONE

图 4

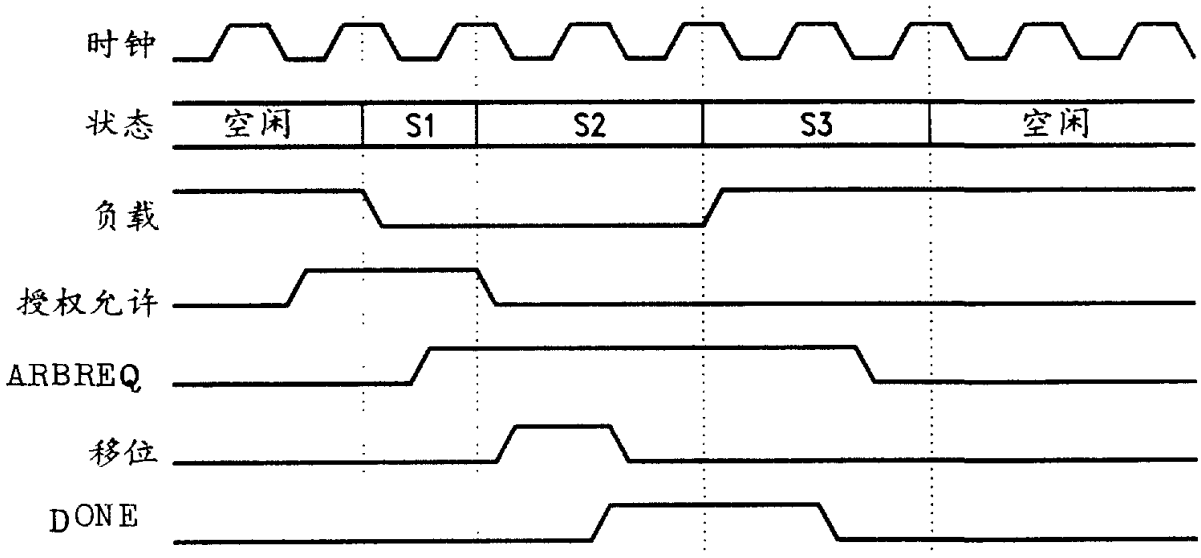

时钟
状态
空闲
S1
S2
S3
空闲
负载
授权允许
ARBREQ
移位
DONE

图 5

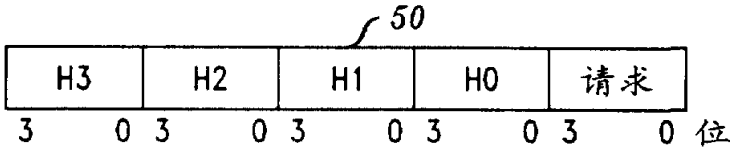


图 6

H3	H2	H1	H0	请求
0000	0000	0000	0000	XXXX

图 7

地址	G3	G2	G1	G0	周期数	错误代码
0000	0	0	0	0	0	0
0001	0	0	0	1	5	1
0010	0	0	1	0	4	0
0011	0	0	0	1	5	1
0100	0	1	0	0	2	0
0101	0	0	0	1	5	1
0110	0	0	1	0	4	0
0111	0	0	0	1	5	1
1000	1	0	0	0	1	0
1001	0	0	0	1	5	1
1010	0	0	1	0	4	0
1011	0	0	0	1	5	1
1100	0	1	0	0	2	0
1101	0	0	0	1	5	1
1110	0	0	1	0	4	0
1111	0	0	0	1	5	1

图 8

H3	H2	H1	H0	请求
0000	0000	0000	0001	XXXX

图 9

H3	H2	H1	H0	请求
0000	0000	0001	1000	XXXX

图 10

H3	H2	H1	H0	请求
0000	0001	1000	0010	XXXX

图 11

H3	H2	H1	H0	请求
0001	1000	0010	0100	XXXX

图 12

H3	H2	H1	H0	请求
1000	0010	0100	0001	XXXX