



[12] 发明专利说明书

[21] ZL 专利号 94194017.9

[43] 授权公告日 2003 年 1 月 22 日

[11] 授权公告号 CN 1099633C

[22] 申请日 1994.8.16 [21] 申请号 94194017.9
 [30] 优先权
 [32] 1993.9.23 [33] GB [31] 9319662.4
 [86] 国际申请 PCT/GB94/01793 1994.8.16
 [87] 国际公布 WO95/08801 英 1995.3.30
 [85] 进入国家阶段日期 1996.5.3
 [71] 专利权人 ARM 有限公司
 地址 英国剑桥郡
 [72] 发明人 D·V·贾加
 [56] 参考文献
 COMPCON SPRING ARM6 A HIGH PERFORMANCE LOW
 POWER COMSUMPTION 1993-02-01 M MULLER
 SAN FRANCISCO CA US
 S. B. FURBER VLSI RISC ACHITECTURE AND ORGANI-
 ZATION

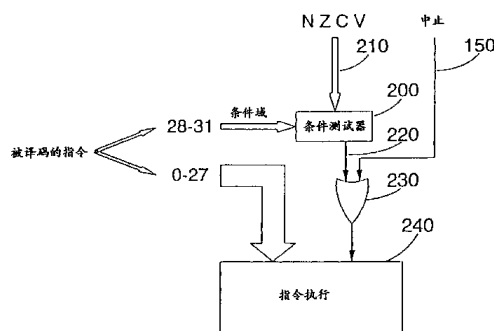
1989-01-01 MARCEL DEKKER INC NEW
 YORK US
 审查员 陈炜
 [74] 专利代理机构 中国专利代理(香港)有限公司
 代理人 王勇 王忠忠

权利要求书 2 页 说明书 10 页 附图 4 页

[54] 发明名称 数据处理指令的执行

[57] 摘要

执行连续的数据处理指令的数据处理装置包括：为响应一条或多条指令访问数据存储器的存储器访问设备，该存储器访问设备包括为检测每一存储器访问是否无效的设备；响应由以前执行的指令产生的该装置的处理状态和在每一指令执行时可操作的用于检测是否该指令应该被执行的条件测试设备；以及响应存储器访问设备和条件测试设备的条件控制设备，如果存储器访问设备检测出由先行指令起动的存储器访问无效或条件测试设备检测出当前指令不应该被执行，则防止此设备对当前指令完全执行。



1. 数据处理装置, 其中以流水线方式执行连续的数据处理指令, 所述装置包括:

5 响应一条或多条所述数据指令访问数据存储器的存储器访问设备(110), 所述存储器访问设备包括检测每一存储器访问是否无效的设备; 和

响应由以前执行的指令产生的所述装置的处理状态和在每一指令的执行时可操作的条件测试设备(200), 用于检测该指令是否应该被执行;

10 其特征在于:

响应所述存储器访问设备(110)和所述条件测试设备(200)的条件控制设备, 如果所述存储器访问设备(110)检测出由先行指令起动的存储器访问无效或所述条件测试设备(200)检测出所述当前指令不应该被执行, 用于防止当前指令的完全执行。

15 2. 按照权利要求1的装置, 其中:

所述装置包括一个或多个用来存储所述数据处理装置的当前处理状态的数据指示性的处理标志;

每一指令包括一个定义为要被执行的该指令所需要的所述处理标志状态的条件代码;

20 所述条件测试设备(200)是可操作的以将由在每一指令中的所述条件代码定义的处理标志的所需状态和所述处理标志的实际状态相比较。

3. 按照权利要求2的装置, 包括四个处理标志位, 分别表示:

25 (i) 是否所述装置的一个以前的数据处理操作产生一个负结果(N);

(ii) 是否所述装置的一个以前的数据处理操作产生一个零结果(Z);

(iii) 是否由所述装置的一个以前的数据处理操作置一进位位(C); 和

30 (iv) 是否在所述装置的一个以前的数据处理操作时, 产生一个算术溢出(V)。

4. 按照权利要求 1 到 3 中任何一个的装置，其中所述存储器访问设备（110）包括：

给所述数据存储器（120）传送一存储器地址的设备；和

5 给所述数据存储器连续传送数据或从所述数据存储器接收数据的设备。

5. 按照权利要求 1 到 3 中任何一个的装置，其中：

所述存储器访问设备（110）是可操作的以产生一中止控制信号（150）指明一存储器访问无效；

10 所述条件测试设备（200）是可操作的以产生一条件失效控制信号（220）指明所述当前指令不应该被执行；和

所述装置包括为结合所述中止控制信号和所述条件失效控制信号以产生一组合的控制信号用来供应所述条件控制设备（240）的设备（230）。

15 6. 按照权利要求 5 的装置，其中所述结合设备（230）包括一逻辑或门。

7. 按照权利要求 1 到 3 中任何一个的装置，其中该装置的数据处理操作由一时钟信号控制。

8. 包括按照前面的任何一个权利要求的装置的集成电路。

20 9. 数据处理方法，其中以流水线方式执行连续的数据处理指令，所述方法包括的步骤有：

响应一条或多条所述指令访问数据存储器；

检测每一存储器访问是否无效；

在每一指令执行时检测该指令是否应该被执行，这依赖于所述装置由以前执行的指令产生的处理状态；

25 其特征在于步骤：

如果检测出由先行指令起动的存储器访问无效或者所述当前指令不应该被执行，则防止完全执行当前指令。

数据处理指令的执行

发明背景

发明领域

本发明涉及数据处理指令的执行。

现有技术的叙述

一些数据处理器包括中央处理器（CPU），它在当前执行的数据处理指令的控制下通过一中间存储器管理单元能够访问存储在随机存取存储器（RAM）中的数据。以前提出的这种数据处理器例子是ARM6处理器，它已由高级里斯克（Risc）机器股份有限公司在1993年发表的“ARM6数据表”中叙述。

在存储器访问时，存储器管理单元可以产生一个中止信号，指明当前存储器访问不可能完成。中止信号产生的原因可能很多。在一个例子中，一个被中止的存储器访问能在使用虚拟存储器的数据处理系统中出现，其中数据在RAM和较慢的磁盘存储器之间交换而形成一种假像，认为可编址的存储器空间大于所提供的RAM的数量。在这种系统中，如果相应于所需虚拟地址的数据当前保存在磁盘存储器中而不是在RAM中，则在这些数据在访问前有一延迟，在该延迟中数据必须从磁盘存储器传输到RAM中。在这种情况下，当前的存储器访问被中止，稍后重新尝试访问这些数据。

从存储器管理单元提供的中止信号产生得太晚，以至不能停止启动失效的存储器访问的指令的执行，但是可以代而用于取消后继的数

据处理指令的执行，亦即在起动失效的存储器访问的指令的后面的数据处理指令。这是有用的，因为以后的指令可能依赖存储器访问成功。

使用中止信号以取消紧接着起动失效的存储器访问指令的指令执行需要对中止信号严格定时。或者，必须提供一复杂的机构来“擦除”紧接的指令在其执行完成后产生的效果。

附图中的图 1 是一原理定时图，表明在上面提到的以前提出的数据处理器中中止信号在数据写操作时（其中数据写到 R A M 中）的定时需求。

参考图 1，时钟信号 1 0 控制数据处理器执行数据处理指令。在起动一数据写操作时，数据处理器向存储器管理单元提供一存储器地址 2 0，在时钟信号半个周期后写到该地址的数据 3 0 由数据处理器输出。

如果存储器管理单元检测出存储器地址 2 0 无效（例如，因为相应于该地址的数据当前保存在虚拟存储器系统中的磁盘存储器中），则由存储器管理单元产生一中止信号 4 0 并提交给数据处理器。

在数据写指令后面的下一指令马上执行，因为在要写的数据被放在数据总线上之后无须等待（在正常环境下）从存储器管理单元来的响应。相应地，为使中止信号及时到达以便取消直接跟随指令的执行，上面提到的以前提出的数据处理器要求中止信号在数据处理器输出要写的数据之前半个时钟信号周期有效。

在实际中，这个定时限制很难达到，需要存储器管理单元特别快的操作（伴随相应的该管理单元高的功率消耗）。

上面提到的以前提出的数据处理器也提供其整个指令集的条件执行。这是靠直至把四个处理标志的当前状态与由在每一指令中包含的条件码定义的各自的状态比较而实现的。这种比较与该指令的执行同时进行。如果处理标志的状态与由条件代码指定的状态不匹配时，禁止指令的完整执行。

另一描述这类数据处理器文献是M. 米勒的文章“ARM6：高性能低功耗宏单元”，（M. Muller, COMPCONSpring93, pages80-87）。它描述了上面提到的ARM6处理器的存储器中止和条件指令执行特征。

发明总结

本发明提供一种数据处理装置，其中连续的数据指令以流水线的方式执行，该装置包括：响应一条或多条指令而访问数据存储器的存储器访问设备，该存储器访问设备包括检测每一存储器访问是否无效的设备；以及响应该装置由以前执行的指令产生的处理状态并在每一指令执行时可操作用于检测该指令是否应该被执行的条件测试设备；其特征在于响应存储器访问设备和条件测试设备的条件控制设备，如果该存储器访问设备测试出由先行指令起动的存储器访问无效或者条件测试设备测试出当前指令不应该执行，则防止一当前指令的完全执行。

在按照本发明的数据处理装置中，使用一个完整的条件指令集，还使用有条件地阻止每一指令完整执行的机构来处理存储器的中止。从而与先行指令起动的存储器访问有关的存储器中止信号可以在每一指令执行时与条件测试设备测试该指令是否应该执行同时接收。这可

以允许每一指令的存储器中止的处理比在上面描述的以前提出的数据处理装置所允许的要晚一些。

在一个优选的实施例中，该装置包括一个或多个用来存储该数据处理装置的一个当前处理状态的数据指示性的处理标志；每一指令包括一个定义对被执行的指令所需要的处理标志的状态的条件代码；和条件测试设备可以操作用于比较由在每一指令中的条件代码所定义的处理标志所需要的状态和该处理标志的实际状态。条件代码可以规定，一个特定的处理标志应该置位成一个特定的逻辑状态，或该处理标志的状态对当前指令是否应该执行没有影响。在极端的情况下，一个可能的条件代码可以规定一个特定的指令应该执行而无视任何处理标志的状态。

处理标志可以规定该装置的处理状态的各种特征。在优选的实施例中该装置包括四个处理标志，分别表示：

- (i) 该装置的一个以前的数据处理操作是否产生一个负结果；
- (ii) 该装置的一个以前的数据处理操作是否产生一个零结果；
- (iii) 是否由该装置的的一个以前的数据处理操作置一进位位；和
- (iv) 是否在该装置的一个以前的数据处理操作时期产生一算术溢出。

优选存储器访问设备包括：把存储器地址传送给数据存储器的设备；相继把数据传送给数据存储器或从数据存储器接受数据的设备。

为了使条件控制设备可以方便地响应存储器访问设备以及条件测试设备两者，优选：存储器访问设备是可操作的以产生一中止控制信号来指明一存储器访问是无效的；条件测试设备是可操作的以产生一条件失效控制信号来指明当前指令不应该执行；以及该装置包括把中

止控制信号和条件失效控制信号结合的设备以产生一组合控制信号用来提供给条件控制设备。

在一个有利的简单的实施例中，结合装置包括一逻辑或门。

在一方便的实施例中，该装置的数据处理操作由一时钟信号控制。

从第二方面来看，本发明提供一个包括上面定义的装置的集成电路。

从第三方面来看，本发明提供一种数据处理方法，其中连续的数据处理指令以流水线方式执行，该方法包括的步骤有：响应一条或多条指令访问数据存储单元；检测每一存储器访问是否无效；以及在每一指令执行时检测该指令是否应该被执行，这依赖于由以前执行的指令产生的该装置的处理状态；其特征在于步骤：如果检测出由先行指令起动的存储器访问无效或当前指令不应该被执行时，则阻止当前指令的完整执行。

附图的简要描述

本发明以参照附图的示例的方式描述，描述中相同的部分以相同的参考数字表示，其中：

图 1 是原理定时图，表明由以前提出的数据处理器在写操作时中止信号的定时需要；

图 2 是按照本发明的实施例的一个数据处理装置的原理框图；

图 3 是一中央处理器的一部分的原理框图；

图 4 是原理定时图，表明由图 2 中的数据处理装置在一数据写操作时中止信号的定时需要。

优选实施例的说明

现在参考图 2，图中示出按照本发明的一个实施例的数据处理装置的原理框图。该装置包括一中央处理器（CPU）100，一存储器管理单元（MMU）110和一随机存取存储器（RAM）120。CPU 100和MMU 110由用于交换存储器地址的地址总线130和用于交换数据的数据总线140连接。从MMU 110到CPU 100还提供中止控制线150用于传输失效的或无效的存储器访问的中止指示性信号。

对CPU 100和MMU 110提供一时钟信号以控制这两个单元的操作。每一数据处理指令由CPU 100以特定数目的时钟信号周期执行（取决于指令的性质）并且存储器访问由MMU 110与时钟信号同步执行。

MMU 110在CPU 100和时钟信号的控制下访问存储在RAM 120中的数据。相应地MMU通过多条地址和数据线160连接到RAM上。

图 3 是中央处理器 110 的部分原理框图。CPU 100 使用指令流水线技术，以使处理和存储操作基本上连续地进行。典型地，当一个数据处理指令被执行时，其后继指令被译码而第三条指令从存储器中取出。这种安排被称为三级执行流水线。

在图 3 中示出的 CPU 100 的部分中，从存储器中取出而后被译码的数据处理指令被送去执行。数据处理指令是 32 位的数据字，其中第 28 到 31 位形成一个四位的条件域。其余的位（位 0 到 27）定义响应该指令而要被执行的操作和在一些场合下定义要被执行的操作的操作符。

条件域（位 28 到 31）传输到条件检测器 200，其对条件域的位与 16 个预先定义的条件代码进行比较。这 16 个条件代码定义分别称为 N、Z、C 和 V 标志的四个处理器标志 210 中的一个或多个状态。这些标志表示一个由以前执行的指令产生的 CPU 100 的处理状态。

N 标志表示从 CPU 100 的以前的运算操作得到一个负结果；Z 标志表示从以前的 CPU 100 操作得到的一个零（等于）结果；C 标志表示在以前的 CPU 操作时是否置一进位位；而 V 标志表示在以前的 CPU 操作时产生一算术溢出。

在条件域的四位和应用于N、Z、C和V标志的条件之间的关系

示于下面的表中:

0000=EQ-Z置位(等于)

0001=NE-Z清除(不等)

0010=CS-C置位(无符号大于或相等)

0011=CC-C清除(无符号小于)

0100=MI-N置位(负)

0101=PL-N清除(正或零)

0110=VS-V置位(溢出)

0111=VC-V清除(无溢出)

1000=HI-C置位和Z清除(无符号大于)

1001=LS-C清除和Z置位(无符号小于或相等)

1010=GE-N置位和V置位, 或N清除和V清除(大于等于)

1011=LT-N置位和V清除, 或N清除和V置位(小于)

1100=GT-Z清除, 和要么N置位和V置位或者N清除和V清除(大于)

1101=LE-Z置位, 或者N置位和V清除, 或者N清除和V置位(小于或等于)

1110=AL-总是

1111=NV-总不

条件测试器 2 0 0 测试上面所列标志的状态, 这依赖于当前指令的条件域的哪一位被置位。这种比较发生在当前指令的执行中。仅当适当的标志被置于由条件域指定的状态, 则当前指令才被允许完成其执行。

如果总是 (A L) 条件被指定, 则该指令被执行而不管标志如何。总不 (N V) 条件码阻止指令的执行而不管标志 2 1 0 的状态。

条件测试器 2 0 0 产生一输出信号 2 2 0，指明是否当前指令应该完全执行。输出信号 2 2 0 使用一或门 2 3 0 与从 MMU 110 到 CPU 1 1 0 的中止控制线 1 5 0 上提供的中止信号相结合。于是如果条件测试器 2 0 0 指明当前指令不应该被完全执行或者该中止信号被 MMU 110 所确立，则或门 2 3 0 的输出被置位。

指令执行单元 2 4 0 接收每一指令的 0 到 2 7 位，其定义要执行的操作和在一些场合下，定义要执行的操作的操作数。在当前指令执行时（亦即在条件测试器 2 0 0 和或门 2 3 0 的传播和处理延迟后），指令执行单元 2 4 0 接收或门 2 3 0 的输出，指明当前指令是否应当完全执行。如果或门 2 3 0 的输出指明当前指令不应当完全执行，则当前指令被取消而不改变任何寄存器或与该装置有关联的存储器单元的状态。

条件测试器 2 0 0 和指令执行单元 2 4 0 可能如同上面提到的以前提出的 ARM 6 处理器的相应部件同样的形式。

使用图 3 装置的结果大大地缓解了中止信号的严格定时，所以由先行指令起动的有关存储器访问的中止信号可以在每一指令执行的后阶段置位。

如果所收到的中止信号作为一失效的取指令操作的结果，则该指令在上面提到的三级指令流水线中的后阶段上简单地被丢弃。

数据读操作后随一非存储器访问（内部）处理器周期以便留出从存储器读出的数据装入适当的处理器寄存器中的时间。相应地，如果接收到的中止信号是作为失效的数据读操作的结果，则随着读操作的内部周期给出由 MMU 1 1 0 从存储器供应的任何数据（这些数据很

可能是错误的)被忽略且不存储在打算寄存的位置(例如一处理器寄存器)上的时间。

图4中示出指令执行和接收数据写操作的中止信号的定时关系,其中由CPU100控制指令执行的时钟信号300与地址总线130和数据总线140的状态一起示出。

为了起动数据写操作,由CPU100把地址310放在地址总线上。在时钟信号的半个周期后由CPU100把数据320放在数据总线140上。这些操作一旦完成,CPU100就能在时钟信号300的下一相继周期时期执行下一指令。

在数据320被放在数据总线上的同时,如果收到一中止信号(在图4中用330表示)或者如果条件测试器200决定下一指令不应该被执行(其由条件测试器200在数据被放在数据总线后的半个周期之内决定并用340表示),则下一指令的执行被放弃。

虽然本发明的例证性的实施例在这里是参照附图详细叙述的,应该理解本发明并不限于这些精确的实施例,对熟悉本技术领域的人来说对其可以实行各种改变或修改而不离开由后面的权利要求所限定的本发明的范围。

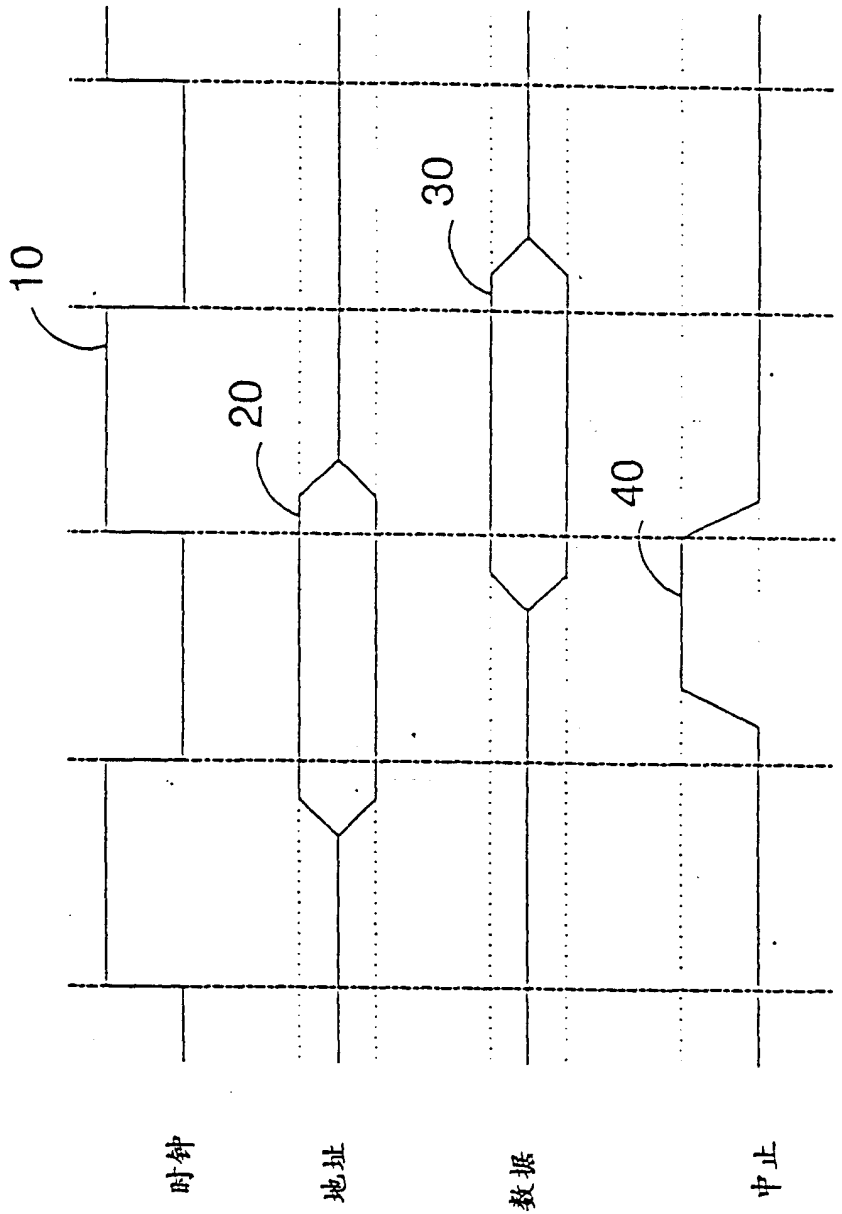


图1

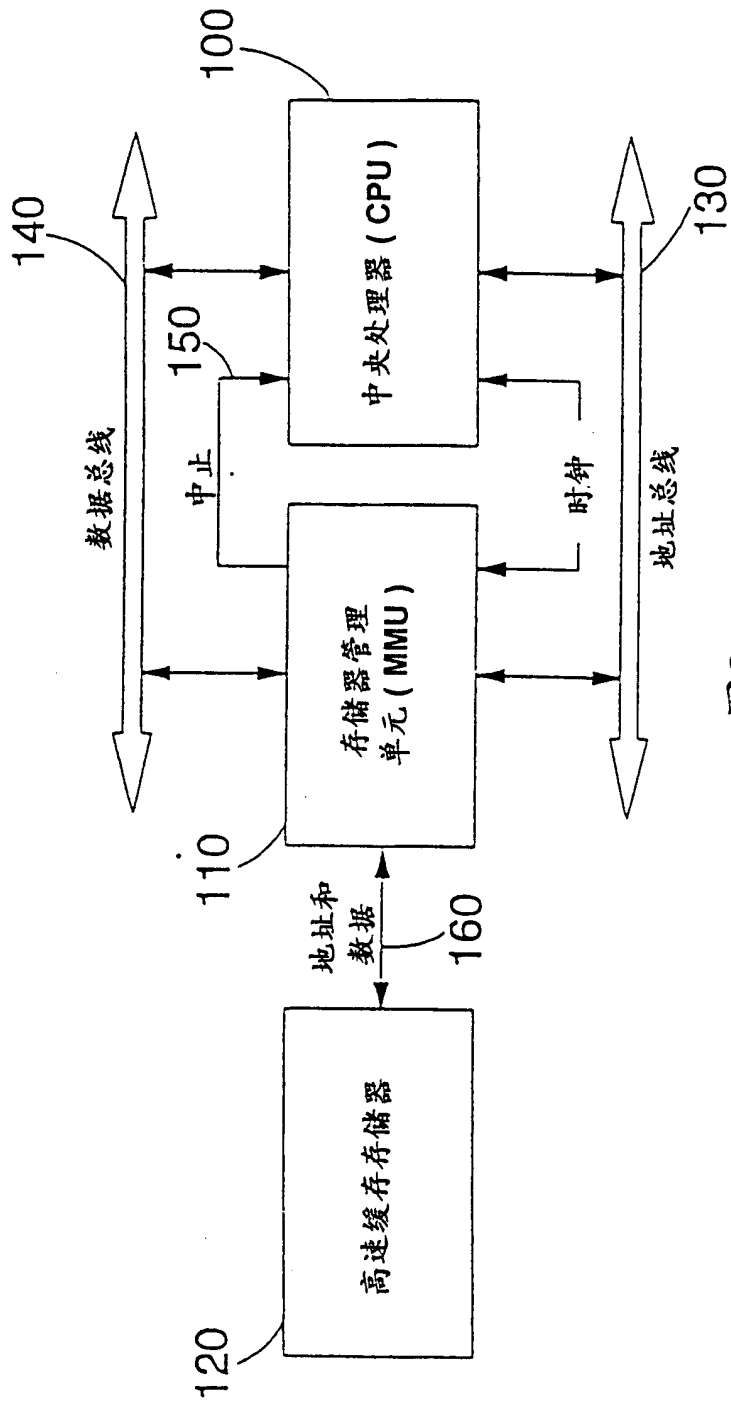


图2

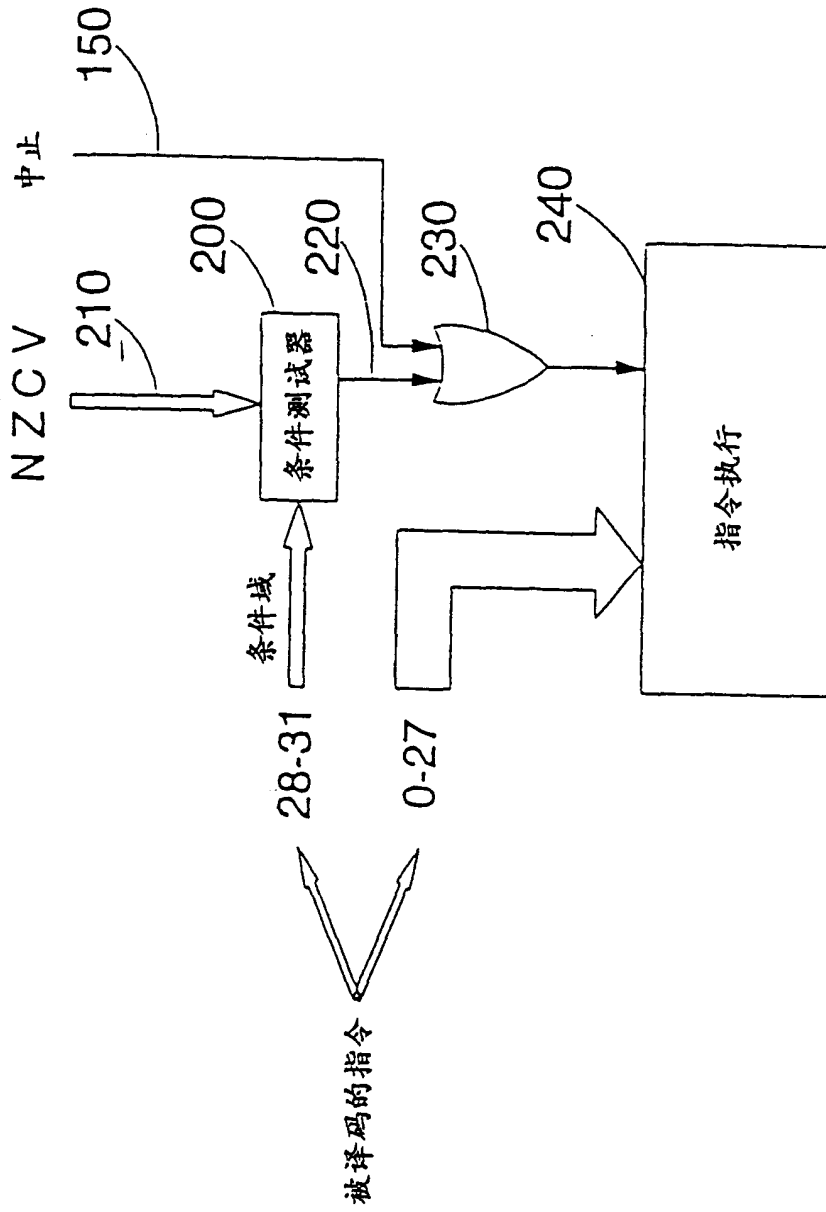


图3

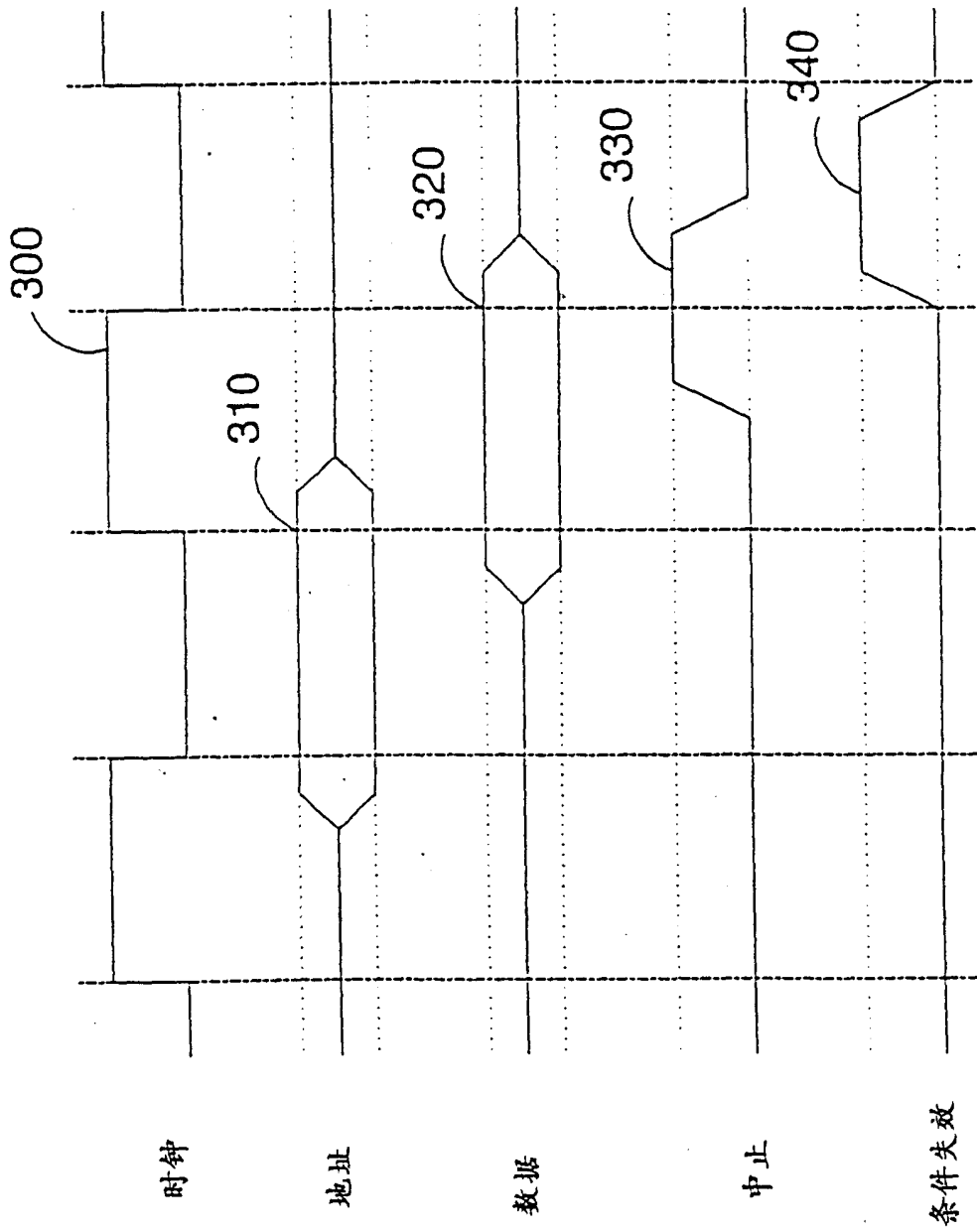


图4