

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G11C 7/22 (2006.01)
G11C 11/409 (2006.01)
G11C 8/16 (2006.01)



[12] 发明专利说明书

专利号 ZL 200610054989. X

[45] 授权公告日 2009 年 8 月 19 日

[11] 授权公告号 CN 100530417C

[22] 申请日 2001. 11. 26

[21] 申请号 200610054989. X

分案原申请号 01139358.0

[30] 优先权

[32] 2000. 12. 20 [33] JP [31] 387891/2000

[32] 2000. 12. 27 [33] JP [31] 398893/2000

[32] 2000. 12. 27 [33] JP [31] 399052/2000

[32] 2001. 2. 9 [33] JP [31] 034361/2001

[32] 2001. 2. 14 [33] JP [31] 037547/2001

[73] 专利权人 富士通微电子株式会社

地址 日本东京

[72] 发明人 松崎康郎 铃木孝章 山崎雅文

川崎健一 鎌田心之介

[56] 参考文献

US4796232 1989. 1. 3

US5659711A 1997. 8. 19

US5768211A 1998. 6. 16

US5319596A 1994. 6. 7

US5001671 1991. 3. 19

US5946262A 1999. 8. 31

US6078527A 2000. 6. 20

审查员 刘晓燕

[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 王永刚

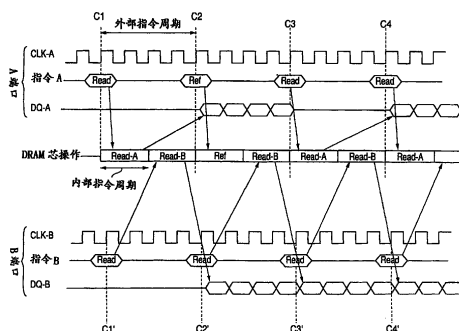
权利要求书 1 页 说明书 100 页 附图 139 页

[54] 发明名称

基于动态随机存取存储器核心的多端口存储器

[57] 摘要

半导体存储器件包括数量为 N 的多个外部端口，每个外部端口都接收指令，和一个内部电路，它在输入到一个外部端口的指令的最小时间间隔中至少实施 N 次存取操作。



1. 一种半导体存储器件，包括：

数量为 N 的多个外部端口，每个外部端口都接收指令，其中 N 等于或者大于 2；和

内部电路，它在输入到一个外部端口的各指令的最小时间间隔期间至少执行 N 次存取操作，

其中所述内部电路包括：

基于动态型存储单元实现的单元阵列；和

刷新电路，它确定刷新存储单元的定时，

其中在第一模式中，响应于从所述半导体存储器件外部输入到 N 个外部端口中的至少一个端口的刷新指令刷新存储单元，在第二模式中，按所述刷新电路指示的定时刷新存储单元。

2. 权利要求 1 所述的半导体存储器件，其中当 N 个外部端口中的至少一个端口处于不激活状态时采用第二模式。

3. 权利要求 1 所述的半导体存储器件，其中从器件外部在所述 N 个外部端口中选择一个用于接收刷新指令的外部端口。

4. 权利要求 3 所述的半导体存储器件，其中当在 N 个外部端口中所述用于接收刷新指令的外部端口处于不激活状态时采用第二模式。

5. 权利要求 3 所述的半导体存储器件，进一步包括一个模式寄存器，它指示 N 个外部端口中的用于接收刷新指令的外部端口。

基于动态随机存取存储器核心的多端口存储器

本申请是 2001 年 11 月 26 日提交的名称为“基于动态随机存取存储器核心的多端口存储器”中国专利申请 No.01139358.0 的分案申请。

技术领域

本发明一般涉及半导体存储器件，特别是涉及装备有多个端口的半导体存储器。

背景技术

多端口存储器，它们是装备有多个端口的半导体存储器，可以分成不同的类型。当下文中使用术语“多端口存储器”时，它指的是具有多个端口的存储器，该多端口存储器允许从任何一个端口独立地存取到一个公共存储器阵列。这样一个存储器可以有一个 A 端口和一个 B 端口，并允许对于公共存储器阵列从与 A 端口链接的 CPU 和从与 B 端口链接的 CPU 独立地进行读/写操作。

一个多端口存储器装备有一个称为判优器的判优电路。判优器确定从多个端口接收的各存取要求的优先权，存储器阵列的控制电路根据确定的优先权一个接一个的进行存取操作。例如，一个存取要求越早到达一个端口，就会给予该存取越高的优先权。

在这种情形中，因为随机地从多个接口存取存储器阵列，所以在执行了读或写的存取操作后需要立即使存储器阵列复位，从而保证存储器阵列已经为下一次存取作好了准备。即，如果响应一个来自一个给定端口的存取使一条字线保持在一个选出的状态，和如一般在 DRAM 中使用的列存取操作中那样连续地动移各列地址以便读出连续的数据，则在该操作期间来自另一个端口的存取将一直等待着。因此，在每次读或写操作后需要立即使存储器阵列复位。

常规地,已经典型地将一个 SRAM 用作一个多端口存储器的一个存储器阵列。这是因为一个 SRAM 允许高速随机存取,而且可以进行非破坏性读操作。

在一个具有两个端口的多端口存储器中,例如,一个 SRAM 存储单元具有两组字线和各位线对。一个端口用一组字线和一个位线对实施读/写操作,另一个端口用另一组字线和一个位线对实施读/写操作。在这种方式中,能够从两个不同的端口独立地实施读/写操作。然而,因为当两个端口企图在同一时间将数据写入同一存储单元时不可能同时进行两个写操作,所以给予一个端口进行写操作的优先权,而给予另一个端口一个 BUSY (忙碌) 信号。这称为一个 BUSY 状态。

当开发一个系统使它有改善的性能时,可以由该系统处理数据量也增加了。结果,一个多端口存储器需要很大的容量。然而,SRAM 型多端口存储器有一个缺点,即存储单元的尺寸大。

为了消除这个缺点,在一个多端口存储器中采用一个 DRAM 阵列是可以理解的。为了得到比多端口 SRAM 高得非常多的电路密度,需要用于多端口存储器的一个 DRAM 存储单元以与一个典型的 DRAM 单元相同的方式只与一条字线和一条位线连接。如果以这样一种方式用 DRAM 元件制成存储块,则如果一个端口正在对一个给定的存储块执行读或写操作,另一个端口就不能存取该存储块。这是因为在一个 DRAM 存储单元中只可以有一个非破坏性读操作。即,当读信息时,不能选择在同一条存储块中的另一条字线直到在存储单元中这个信息被放大和恢复与一条字线和一条位线被预先充电为止。

因为这个原因,如果一个给定的端口存取一个正在被另一个端口存取的存储块,则将检测出一个 BUSY 状态。只有当多个端口同时对同一个存储单元发出写要求时,在 SRAM 型多端口存储器中才会出现 BUSY 状态。另一方面,当多个端口同时对同一个存储单元发出任何类型的存取要求时,在 DRAM 型多端口存储器中会出现 BUSY 状态。所以在 DRAM 型存储器中出现 BUSY 状态的概率非常大于在 SRAM 型存储器中出现 BUSY 状态的概率。进一步,一旦处在 BUSY 状态中,

DRAM 型多端口存储器就会受到不能进行希望的操作或由于等待时间处理变得很慢这种问题的困扰。

然而，与 SRAM 型多端口存储器不同，DRAM 型多端口存储器需要一个周期地进行的刷新操作以便保持存储的信息，从而必须采取某种措施保证适当的刷新定时。

因此，本发明的目的是提供能够消除特别与 DRAM 有关的各问题的 DRAM 型多端口存储器。

发明内容

本发明的一个一般的目的是提供能够基本上消除由已有技术的限制和缺点引起的一个或多个问题的半导体存储器件（多端口存储器）。

我们将在下列描述中提出本发明的特点和优点，并且一部分特点和优点将从描述和所附各图变得很清楚，或者可以根据描述中提供的指导通过实践本发明来了解这些特点和优点。我们将通过用使普通的技术人员能够实施本发明的那种完整，清楚，扼要和精确的术语在说明书中具体指出由一个多端口存储器能够实现和得到的本发明的各目的以及各其它的特点和优点。

为了实现这些和其它的优点及根据本发明的目的，如在这里具体地和广泛地描述的那样，本发明提供这样一个半导体存储器件，该器件包括每个端口都接受指令的数量为 N 的多个外部端口和一个在输入到一个外部端口的各指令的最小时间间隔期间至少执行 N 次存取操作的内部电路。

进一步，提供一个判优电路，该判优电路确定一个内部电路执行输入到 N 个外部端口的多个指令时的指令执行次序。

在上面描述的本发明中，当指令进入 N 个端口时，所有的与 N 个端口对应的 N 个指令在任何给定的端口的最小指令周期内被一个接着一个地执行。因此，一个与任何给定端口有关的存取指令出现在器件外部中并在最小指令周期内被执行。在这种情形中，只有当从多个端口存取同一个地址时才会发生一个 BUSY 信号。这样就可以得到和

SRAM型多端口存储器的BUSY状态出现概率一样低的BUSY状态出现概率。

而且在本发明的半导体存储器件中，内部电路包括一个存储单元阵列，该阵列是由许多动态型存储单元和一个定义各存储单元被刷新的定时的刷新电路组成的。在第一模式中，响应一个输入到N个外部端口中的至少一个端口的刷新指令刷新各存储单元，在第二模式中，在刷新电路确定的定时刷新各存储单元。

如上面描述的本发明具有第一个操作模式，其中响应一个来自一个外部端口的指令执行刷新操作，和第二个操作模式，其中响应内部刷新电路执行刷新操作。因为这个配置，允许一个外部端口像一个用于刷新管理的端口那样地操作，以便在各恒定的时间间隔接收刷新指令，或者如果这个用于刷新管理的端口处在一个去激活状态，则内部刷新电路实施刷新操作。这使按照系统配置以一种灵活的方式管理刷新操作成为可能。

附图说明

图 1 是用于解释本发明原理（第 1 方面）的图；

图 2 是表示当只有一个端口正在被使用时实施的一个刷新操作的图。

图 3A 到 3C 是在 2 个端口，3 个端口和 N 个端口的情形中用于解释本发明原理的图；

图 4 是表示根据本发明（第 1 方面）的多端口存储器的第 1 实施例的方框图；

图 5 是与到一个判优器的指令输入有关的电路方框图；

图 6A 和 6B 是表示判优器的一个配置的电路图；

图 7 是表示判优器操作的定时图；

图 8 是与到一个 DRAM 芯的地址输入有关的电路方框图；

图 9 是与数据输出有关的电路方框图；

图 10 是表示一个传输信号发生电路的配置的电路图。

图 11 是与数据输入有关的电路方框图；

图 12 是表示当连续输入 Read (读) 指令时实施的操作的定时图；

图 13 是表示当连续输入 Write (写) 指令时实施的操作的定时图；

图 14 是表示当 A 端口和 B 端口两者都操作在最大时钟频率上时的定时图；

图 15 是表示当 A 端口和 B 端口两者都操作在最大时钟频率上时的定时图；

图 16 是表示当指令从 Read 指令改变到 Write 指令时的操作的定时图；

图 17 是表示当指令从“Read”改变到“Write”时输入一个刷新指令的定时的图；

图 18 是表示当使一个端口去激活时实施的操作的定时图；

图 19 是表示当使两个端口去激活时实施的操作的定时图；

图 20A 和 20B 是表示 DRAM 芯操作的定时图；

图 21 是表示当只使一个端口操作时实施的两倍速率操作的定时图；

图 22 是表示当通过使时钟频率两倍高使数据传输速率两倍时两倍速率操作的定时图；

图 23 是用于解释本发明 (第 1 方面) 的第 2 实施例的图；

图 24 是表示根据本发明 (第 1 方面) 的多端口存储器的第 2 实施例的方框图；

图 25A 和 25B 是用于解释连续模式的定时图；

图 26 是表示当对于 A 端口的 Read 指令和 B 端口的 Write 指令产生 BUSY 信号时实施的操作的定时图；

图 27 是表示当对于 A 端口的 Read 指令和 B 端口的 Write 指令产生 BUSY 信号时实施的操作的定时图；

图 28 是表示当对于 A 端口的 Write 指令和 B 端口的 Write 指令产生 BUSY 信号时实施的操作的定时图；

图 29 是表示当对于 A 端口的 Write 指令和 B 端口的 Write 指令

产生 BUSY 信号时实施的操作的定时图；

图 30 是表示在一个能够处理由控制器发出的中断指令的配置中操作的定时图；

图 31 是表示根据本发明（第 1 方面）的第 2 实施例的多端口存储器的一个地址比较器，一个 BUSY I/O 系统，和一个中断系统的配置的图；

图 32 是表示主器件操作的定时图；

图 33 是表示从器件操作的定时图；

图 34 是表示当两个端口的写地址相同时实施的主器件操作的定时图；

图 35 是表示当两个端口的写地址相同时实施的从器件操作的定时图；

图 36 是表示当两个端口的写地址相互匹配使控制器发出中断指令时主器件操作的定时图；

图 37 是表示当两个端口的写地址相互匹配使控制器发出中断指令时从器件操作的定时图；

图 38 是用于解释本发明（第 2 方面）的原理的图，表示当对于两个端口实施读操作时的情形；

图 39 是用于解释本发明（第 2 方面）的原理的图，表示脉冲串长度为 4 的例子；

图 40 是表示在 2 和 3 个端口的情景情形中在一个最小外部指令周期和各内部操作周期之间的关系图；

图 41 是表示在 n 个端口的情形中在一个最小外部指令周期和各内部操作周期之间的关系图；

图 42 是表示根据本发明（第 2 方面）的一个实施例的多端口存储器的配置的图；

图 43A 到 43C 是表示根据本发明（第 2 方面）的实施例的多端口存储器的配置的图；

图 44 是表示根据第 1 实施例与指令处理有关的各装置的配置的

图；

图 45 是表示根据第 1 实施例与指令处理有关的各装置的配置的图；

图 46 是判优器的实施例；

图 47 是表示根据第 1 实施例与地址处理有关的部分的配置的图；

图 48 是表示根据第 1 实施例与数据输出有关的部分的配置的图；

图 49 是表示图 48 的传输信号发生电路的图。

图 50 是表示根据第 1 实施例与数据输入有关的部分的配置的图；

图 51 是表示当将 Read 指令接连地输入两个端口时实施的操作的图；

图 52 是表示当将 Read 指令接连地输入两个端口时实施的操作的图；

图 53 表示当接连地输入 Write 指令时的例子；

图 54 表示当 A 端口和 B 端口两者都操作在最大时钟频率上进行 Read 操作时实施的操作；

图 55 表示当 A 端口和 B 端口两者都操作在最大时钟频率上进行 Read 操作时实施的操作；

图 56 是表示当 A 端口和 B 端口两者都操作在最大时钟频率上进行 Write 操作时实施的操作的图；

图 57 是表示当两个端口都操作在最高频率，并用内部产生的刷新指令使从 Write 指令改变到 Read 指令时实施的操作的时间图；

图 58 是表示当两个端口都操作在最高频率，并用内部产生的刷新指令使从写指令改变到读指令时实施的操作的时间图；

图 59A 和 59B 是 DRAM 芯操作的定时图；

图 60 是表示刷新电路的配置的电路图；

图 61 是表示第 2 判优器的电路配置的图；

图 62 表示两个端口经受 Write→Read 的指令改变，和在 REF 传输禁止期间发生一个刷新定时器事件的情形；

图 63 表示两个端口经受 Write→Read 的指令改变，和在 REF 传

输禁止期间发生一个刷新定时器事件的情形；

图 64 表示两个端口如上面情形一样经受 Write→Read 的指令改变，但是在 REF 传输禁止期间前发生一个刷新定时器的情形；

图 65 表示两个端口如上面情形一样经受 Write→Read 的指令改变，但是在 REF 传输禁止期间前发生一个刷新定时器的情形；

图 66 表示只有 A 端口经受 Write→Read 的指令变迁，和在 REF 传输禁止期间发生一个刷新定时器事件的情形；

图 67 表示只有 A 端口经受 Write→Read 的指令变迁，和在 REF 传输禁止期间发生一个刷新定时器事件的情形；

图 68 表示在两个端口连续写的情形的定时图；

图 69 表示在两个端口连续写的情形的定时图；

图 70 表示与图 57 和图 58 所示的第 1 实施例操作对应的第 2 实施例操作的定时图；

图 71 表示与图 57 和图 58 所示的第 1 实施例操作对应的第 2 实施例操作的定时图；

图 72 表示与图 56 所示的第 1 实施例操作对应的第 2 实施例操作的定时图；

图 73 是用于解释本发明（第 3 方面）的原理的图，表示对于两个端口实施读操作的情形；

图 74 是用于解释本发明（第 3 方面）的原理的图，表示脉冲串长度为 4 的例子；

图 75 是表示在 2 和 3 个端口情形中在一个最小外部指令周期和各内部操作周期之间关系的图；

图 76 是表示在 N 个端口情形中在一个最小外部指令周期和各内部操作周期之间关系的图；

图 77 是表示根据本发明（第 3 方面）的一个实施例的多端口存储器的配置的图；

图 78A 到 78C 是表示根据本发明（第 3 方面）的上述实施例的多端口存储器的配置的图；

图 79 是表示根据第 1 实施例与指令处理有关的各装置的配置的图；

图 80 是表示根据第 1 实施例与指令处理有关的各装置的配置的图；

图 81 是判优器的实施例；

图 82 是表示指令寄存器的配置的图；

图 83 是表示指令寄存器的配置的图；

图 84A 和图 84B 表示寄存器控制电路的操作；

图 85 是表示指令寄存器操作的图；

图 86 是表示指令寄存器操作的图；

图 87 是表示根据第 1 实施例与地址处理有关的部分的配置的图；

图 88 是表示根据第 1 实施例与数据输出有关的部分的配置的图；

图 89 是表示图 88 的传输信号发生电路的图；

图 90 是表示根据实施例与数据输入有关的部分的配置的图；

图 91 是表示根据实施例与数据输入有关的部分的配置的图；

图 92 表示当将 Read 指令接连地输入两个端口时实施的操作；

图 93 表示当将 Read 指令接连地输入两个端口时实施的操作；

图 94 表示当将接连地输入 Write 指令时的例子；

图 95 表示当 A 端口和 B 端口两者都操作在最大时钟频率上进行 Read 操作时实施的操作；

图 96 表示当 A 端口和 B 端口两者都操作在最大时钟频率上进行 Read 操作时实施的操作；

图 97 是表示当 A 端口和 B 端口两者操作在最大时钟频率上进行 Write 操作时实施的操作的图；

图 98 是表示当两个端口操作都在最高频率，并用内部产生的刷新指令使从 Write 指令改变到 Read 指令时实施的操作的时间图；

图 99 是表示当两个端口都操作在最高频率，并用内部产生的刷新指令使从 Write 指令改变到 Read 指令时实施的操作的时间图；

图 100A 和 100B 是 DRAM 芯操作的定时图；

图 101 是表示根据本发明（第 4 方面）的多端口存储器的实施例的方框图；

图 102 是表示根据本发明（第 4 方面）的多端口存储器操作的一个例子的定时图；

图 103 是表示根据本发明（第 4 方面）的多端口存储器操作的一个例子的定时图；

图 104 是表示根据本发明（第 4 方面）的多端口存储器操作的另一个例子的定时图；

图 105 是表示指令译码器寄存器的方框图；

图 106 是根据本发明（第 4 方面）的实施例的判优器的方框图；

图 107 是表示判优器操作的定时图；

图 108 是地址缓冲器/寄存器和地址改变电路的方框图；

图 109 是存储块的方框图；

图 110A 和 110B 是表示存储块操作的定时图；

图 111 表示根据本发明（第 5 方面）的多端口存储器的第 1 实施例；

图 112 表示多端口存储器的 I/O 电路 5010 和存储块 MB 的详细情况；

图 113 表示地址比较电路的详细情况；

图 114 表示比较器的详细情况；

图 115 表示当加到加到输入/输出端口 PORT-A 和 PORT-B 的行地址信号 RA 相互匹配时实施的比较器操作；

图 116 表示当行地址信号 RA 在输入/输出端口 PORT-A 和 PORT-B 之间不匹配时比较器的操作；

图 117 表示当加到输入/输出端口 PORT-A 和 PORT-B 的行地址信号 RA 在时钟信号 CLKA 与时钟信号 CLKB 具有不同周期的条件下匹配时比较器的操作；

图 118 表示提供给图 112 所示的判优电路的判优控制电路；

图 119 表示当加到输入/输出端口 PORT-A 和 PORT-B 的行地址

信号匹配时实施的判优控制电路的操作;

图 120 表示当加到输入/输出端口 PORT-A 和 PORT-B 的行地址信号 RA 相互匹配时实施的操作;

图 121 表示当时钟信号 CLKA 和 CLKB 的周期相同, 时钟信号 CLKA 的相位超前时钟信号 CLKB 的相位多于半个周期时实施的操作;

图 122 表示当几乎同时加到输入/输出端口 PORT-A 和 PORT-B 的行地址信号 RA 相互不同时的操作;

图 123 表示多端口存储器的第 2 实施例和根据本发明(第 5 方面)控制多端口存储器的方法;

图 124 表示多端口存储器的第 3 实施例和控制根据本发明(第 5 方面)控制多端口存储器的方法;

图 125 表示判优控制电路的详细情况;

图 126 表示当加到输入/输出端口 PORT-A 和 PORT-B 的行地址信号相互匹配时实施的判优控制电路的操作;

图 127 表示当输入/输出端口 PORT-A 和 PORT-B 接收有效指令 ACT 和相同的行地址信号 RA 时实施读操作的方法;

图 128 表示当将有效指令 ACT 和相互不同的行地址信号 RA 加到输入/输出端口 PORT-A 和 PORT-B 时实施读操作的方法;

图 129 表示当输入/输出端口 PORT-A 和 PORT-B 接收有效指令 ACT 和相同行地址信号 RA 时实施写操作的方法;

图 130 表示对于输入/输出端口 PORT-A 连续地实施写操作和读操作, 和对于输入/输出端口 PORT-B 相继地实施指向与输入/输出端口 PORT-A 的写操作的行地址信号 RA 相同的行地址信号 RA 的写操作和指向与输入/输出端口 PORT-A 的读操作的行地址信号 RA 相同的行地址信号 RA 的写操作的情形;

图 131 表示对于输入/输出端口 PORT-A 连续地实施写操作和读操作, 和对于输入/输出端口 PORT-B 相继地实施指向与输入/输出端口 PORT-A 的写操作的行地址信号 RA 相同的行地址信号 RA 的读操作

和指向与输入/输出端口 PORT-A 的读操作的行地址信号 RA 相同的行地址信号 RA 的写操作的情形；

图 132 表示在时钟信号 CLKA 和 CLKB 具有不同的时钟周期的情形中当加到输入/输出端口 PORT-A 和 PORT-B 的行地址信号相互匹配时实施的操作；

图 133 表示多端口存储器的第 4 实施例和根据本发明(第 5 方面)控制多端口存储器的方法；

图 134 表示当输入/输出端口 PORT-A 和 PORT-B 接收有效指令 ACT 和相同的行地址信号 RA 时实施读操作的方法；

图 135 表示当将有效指令 ACT 和不同的行地址信号 RA 加到输入/输出端口 PORT-A 和 PORT-B 时实施读操作的方法；

图 136 表示将有效指令 ACT 和相同的行地址信号 RA 加到输入/输出端口 PORT-A 和 PORT-B, 实施写操作, 接着加上有效指令 ACT 和不同的行地址信号 RA, 导致实施写操作的情形；

图 137 表示将有效指令 ACT 和相同的行地址信号 RA 加到输入/输出端口 PORT-A 和 PORT-B, 实施写操作, 接着加上有效指令 ACT 和相同的行地址信号 RA, 导致在输入/输出端口 PORT-A 实施读操作和在输入/输出端口 PORT-B 实施写操作的情形；

图 138 表示将有效指令 ACT 和相同的行地址信号 RA 加到输入/输出端口 PORT-A 和 PORT-B, 实施写操作和读操作, 接着加上有效指令 ACT 和不同的行地址信号 RA, 导致实施写操作和读操作的情形；

图 139 表示根据多端口存储器的第 5 实施例的多端口存储器的操作和控制本发明(第 5 方面)的多端口存储器的方法。

具体实施方式

下面, 我们参照所附各图描述本发明的各实施例。

[本发明第 1 方面]

我们首先描述本发明(第 1 方面)的原理。

图 1 是用于解释本发明（第 1 方面）的原理的图。虽然图 1 表示用于解释 2 个端口情形中的原理的图，但是即便提供了两个以上的端口（N 个端口）也可以得到相同的操作。

等效于内部电路（DRAM 芯）操作的两个周期的时间跨度定义为外部指令周期的一个周期。即，芯操作周期是外部指令周期的速率的两倍。由内部存储器在两倍速率以指令越早到达就越早处理指令的这种次序对进入 A 端口和 B 端口的指令进行处理。然后将输出数据传送到每个端口。即，在一个芯操作周期中实施包括选择一条字线，放大数据，选择一条列线，读和写操作，预冲电操作的一系列操作，从而对相关的存储块完成一个存取操作。

例如，在与图 1 的 A 端口有关的外部指令周期的定时 C1，在 A 端口进入一个 Read 指令。进一步，在一个与 B 端口有关的外部指令周期的定时 C1'，在 B 端口进入一个 Read 指令。因为 A 端口的 Read 指令的定时稍微早一些，所以这个 Read 指令在进入 B 端口的读指令之前被执行。这里，一个外部指令周期对应于 4 个时钟脉冲周期。如图 1 所示，在与 1 个芯操作周期对应的 2 个时钟脉冲周期中执行和完成每个 Read 指令。因此，响应在等效于一个外部指令周期的 4 个时钟脉冲周期的时间间隔中进入 A 端口和 B 端口的 Read 指令，能够实施读操作，而不会产生一个 BUSY 状态即便来自 A 端口的读存取和来自 B 端口的读存取都指向同一个存储块。这通过在 2 个时钟脉冲周期中执行和完成每个存取来实现。

在这种方式中，即便多个端口同时存取同一个存储块，因为内部存储器能够以两倍的速度进行相继的和连续的处理，所以也不会产生一个 BUSY 状态。

而且，当如图 1 所示从器件外部（例如在 A 端口）给出一个刷新指令时，在该器件内部能够实施刷新操作而不会影响来自另一个端口（即本例中的 B 端口）的存取。在这种情形中，可以选择多个端口中的一个（即图 1 例中的 A 端口）作为进行刷新管理的端口，总是从这个端口进入刷新指令。

而且，数据输出可以取来从多个列地址并行地读出数据和通过在输出时将并行数据变换成串行数据输出数据的脉冲串的形式。这增加了数据传输速率并使响应连续的 Read 指令连续地输出数据成为可能。

图 2 是表示当只有一个端口正在被使用时实施的刷新操作的图。

如图 2 所示，当提供 2 个端口，例如，A 端口和 B 端口时，不需要让 2 个端口都操作。在器件内提供一个刷新定时器使内部产生刷新指令成为可能。

如图 2 所示，例如，当一个端口（例如，B 端口）不在操作时能够内部产生刷新指令，从而执行刷新指令而不会影响在 A 端口的存取。

现在我们考虑一个例子，其中当进行刷新管理时，控制器 A 控制 A 端口，控制器 B 控制 B 端口。在这种情形中，如果存在一个如上述的内部刷新功能，则当只用 A 端口时 B 端口能够完全停止。这就能够减少由于下面的系统操作的改变引起的功率消耗。

图 3A 到 3C 是当 2 个端口，3 个端口和 N 个端口时用于解释本发明原理的图。

如上面所描述的，本发明也可应用于有 3 个或更多端口的多端口存储器。图 3A 表示在如图 1 和图 2 所示的提供 2 个端口的情形中 1 个端口的操作。图 3B 表示在 3 个端口的情形中 1 个端口的操作，图 3C 表示 N 个端口的情形。如图 3C 所示，可以适当地将内部操作周期的长度设定在 $1/N$ 与 N 个端口存储器的情形中的外部指令周期一样长。

下面，我们描述根据本发明的一个实施例的半导体存储器件。

图 4 是表示根据本发明的多端口存储器的第 1 实施例的方框图。在这个例子中，提供了具有 2 个端口，即 A 端口和 B 端口的配置。

图 4 的多端口存储器 10 包括 A 端口 11，B 端口 12，自刷新电路 13，DRAM 芯 14，判优器 15，刷新指令寄存器 16，指令寄存器 A 17，指令寄存器 B 18，刷新地址寄存器 19，地址寄存器 A 20，地址寄存器 B 21，写数据寄存器 A 22，写数据寄存器 B 23，传输门 A 24，和传输门 B 25。

A 端口 11 包括模寄存器 31, CLK 缓冲器 32, 数据 I/O 电路 33, 地址缓冲器 34, 和指令译码器 35。进一步, B 端口 12 包括模寄存器 41, CLK 缓冲器 42, 数据 I/O 电路 43, 地址缓冲器 44, 和指令译码器 45。在 A 端口 11 和 B 端口 12, 分别与时钟信号 CLKA 和 CLKB 同步地独立地建立到外部总线的存取和来自外部总线的存取。模寄存器 31 和 41 能够在其中存储对于各端口的模式设定如数据等待时间和脉冲串长度。在这个实施例中, A 端口 11 和 B 端口 12 两者都具有各自的模寄存器, 使每个端口都能进行模式设定。然而, 可以将模寄存器只安排在一个端口中, 例如, 使得对于 2 个端口的设置可以由对这一个端口的设置来实现。

自刷新电路 13 包括刷新定时器 46 和刷新指令发生器 47。自刷新电路 13 在器件中产生刷新指令, 分别从 A 端口 11 和 B 端口 12 接收信号 CKEA1 和 CKEB1。信号 CKEA1 和 CKEB1 是分别用 CLK 缓冲器 32 和 42 对外部信号 CKEA 和 CKEB 进行缓冲得到的。用外部信号 CKEA 和 CKEB 暂停各端口的时钟缓冲器并使各端口去基活。如果使 A 端口 11 和 B 端口 12 中的一个进入去激活状态, 则自刷新电路 13 开始它的操作。在模寄存器 31 和 41 中进行了设置, 使得一个端口负责刷新管理的情形中, 当负责刷新管理的的端口变得不操作时可以激活自刷新电路 13。

进一步, DRAM 芯包括存储器阵列 51, 译码器 52, 控制电路 53, WriteAmp (写放大器) 54 和读出缓冲器 55。存储器阵列 51 在其中存储被写和被读的数据, 并包括 DRAM 存储单元, 单元门晶体管, 字线, 位线, 读出放大器, 列线, 列门等。译码器 52 对被存取的地址进行译码。控制电路 53 控制 DRAM 芯 14 的操作。WriteAmp 54 放大写入存储器阵列 51 的数据。读出缓冲器 55 放大从存储器阵列 51 读出的数据。

将到 A 端口 11 的输入传输给地址寄存器 A 20, 刷新指令寄存器 16, 指令寄存器 A 17 和写数据寄存器 A 22。进一步, 将到 B 端口 12 的输入加到地址寄存器 B 21, 刷新指令寄存器 16, 指令寄存器 B 18

和写数据寄存器 B 23。

判优器（判优电路）15 确定指令进入的次序，以便确定将用于在 A 端口 11 和 B 端口 12 之间进行处理的优先权给予哪个指令。以确定的次序，判优器 15 将指令，地址和数据（在写操作的情形中）从各寄存器传输给 DRAM 芯 14。DRAM 芯 14 根据接收的数据进行操作。在 Read 指令的情形中，将从 DRAM 芯 14 读出的数据传输给输入相应指令的端口，然后将该数据从并行数据转换成串行数据，接着与这个端口的时钟同步地输出。

图 5 是与输入到判优器 15 的指令有关的电路的方框图；

指令译码器 35 包括输入缓冲器 61，指令译码器 62 和 $(n-1)$ 时钟延迟电路 63。而且，指令译码器 45 包括输入缓冲器 71，指令译码器 72 和 $(n-1)$ 时钟延迟电路 73。指令寄存器 A 17 包括读指令寄存器 17-1 和写指令寄存器 17-2。而且，指令寄存器 B 18 包括读指令寄存器 18-1 和写指令寄存器 18-2。

在 Read 指令的情形中，分别通过指令译码器 62 或 72 将输入到输入缓冲器 61 或 71 的指令传输给读指令寄存器 17-1 或 18-1，而没有任何定时操作。在 Write 指令的情形中，由 $(n-1)$ 时钟延迟电路 63 或 73 使进入的指令延迟 $(n-1)$ 时钟，然后在当输入一系列要被写的脉冲串数据的第 n 个数据（即最后一个数据）时的定时将它传输给写指令寄存器 17-2 或 18-2。

在刷新指令的情形中，将从 A 端口 11，B 端口 12 或刷新指令发生器 47 提供的刷新指令传输到刷新指令寄存器 16。因为并不如此经常地出现刷新指令，所以不需要提供多个刷新指令寄存器。进一步，从模寄存器 31 和 41 提供输入到刷新指令发生器 47 的自刷新设置信息，该信息指出各端口中的哪一个负责刷新管理。

判优器 15 检测将指令传输到各指令寄存器的次序，并以这个次序将指令一个接着一个地传输到 DRAM 控制电路 53。

当接收指令时（或当接近指令执行的结束时），DRAM 控制电路 53 产生 RESET1 信号，让判优器 15 为下一个指令作好准备。在这个

实施例的特定的配置中，当 RESET1 信号结束时 DRAM 控制电路 53 接收下一个指令。

接收 RESET1 信号时，判优器 15 将复位信号 ResetRA, ResetWA, ResetRB, ResetWB 和 ResetREF 中的一个加到指令寄存器 A 17, 指令寄存器 B 18 和刷新指令寄存器 16 中相应的一个。通过这个操作，使在其中存储了已经传输给 DRAM 芯 14 的指令的指令寄存器复位，在这个指令寄存器中准备接收下一个指令。

图 6A 和 6B 是表示判优器 15 的配置的电路图。

如图 6A 所示，判优器 15 包括比较器 80-1 到 80-10, AND (“与”) 电路 81-1 到 81-5, AND 电路 82-1 到 82-5, AND 电路 83-1 到 83-5, 延迟电路 84-1 到 84-5, 倒相器 85 到 87, NAND (“与非”) 电路 88 和倒相器 89 和 90。比较器 80-1 到 80-10 每个都具有相同的电路配置，如图 6B 所示，包括 NAND 电路 91 和 92 和倒相器 93 和 94。

将来自指令寄存器 A 17 的读指令信号 RA2 和写指令信号 WA2, 来自指令寄存器 B 18 的读指令信号 RB2 和写指令信号 WB2, 和来自刷新指令寄存器 16 的刷新指令 REF2 加到判优器 15。对于由选择 5 个指令信号中的 2 个得到的全部 10 个组合, 10 个比较器 80-1 到 80-10 根据指令到达的定时确定哪一个指令比另一个早。

每个比较器比较 2 个指令的定时，将各输出中的一个设置在 HIGH (高)，它与在其它输入前已经接收了 HIGH 的输入相应。例如，比较器 80-1 到 80-4 中的每一个确定来自 A 端口 11 的读指令信号 RA2 或 4 个其它的指令中对应的一个中哪一个是较早的。如果读指令信号 RA2 比 4 个其它的指令中的任何一个早，则将从 AND 电路 81-1 输出的读指令信号 RA31 设置在 HIGH。当 RESET1 信号是 LOW (低) 时，从判优器 15 将这个读指令信号 RA31 加到 DRAM 芯 14 作为读指令信号 RA3。

当 DRAM 芯 14 接收指令时，DRAM 芯 14 产生是 HIGH 的 RESET1 信号。由倒相器 85 到 87, NAND 电路 88 和倒相器 89 将这个 RESET1 信号变换成脉冲信号，并加到 AND 电路 83-1 到 83-5。当

Read 指令信号 RA31 是 HIGH 时，例如，通过延迟电路 84-1 产生使其中具有接收到的指令的指令寄存器复位的信号。

图 7 是表示判优器 15 的操作的定时图。

具有图 7 中列举的名字的信号表示在图 6A 的各位置中。图 7 是表示当将 Read 指令加到 A 端口 11 和 B 端口 12 上时判优器 15 的操作。如图 7 所示，选择与 A 端口 11 对应的 Read 指令 RA2 作为具有优先权的指令，从而产生 RA31，使得芯电路实施读操作 READ-A。响应由此产生的复位信号 RESET1，使读指令信号 RA2 复位。对应地，选择与 B 端口 12 对应的 Read 指令 RB2，从而产生 RB31。当复位信号 RESET1 变成 LOW 时，将读指令信号 RB3 加到芯电路，从而执行读操作 READ-B。

图 8 是与输入到一个 DRAM 芯 14 的地址有关的电路方框图。

A 端口 11 的地址缓冲器 34 包括输入缓冲器 34-1，传输门 34-2 和 OR (“或”) 电路 34-3。加入具有与从图 5 所示的指令译码器 62 输出的读指令信号 RA1 的前沿对应的脉冲的脉冲信号作为加到 OR 电路 34-3 的一个输入端的 RA1P。进一步，加入具有与从图 5 所示的指令译码器 62 输出的写指令信号 WA1 的前沿对应的脉冲的脉冲信号作为加到 OR 电路 34-3 的另一个输入端的 WA1P。下文中，在它的信号名称的末端具有字母“P”的信号代表具有从对应的信号名称的信号的前沿产生的脉冲的信号。

B 端口 12 的地址缓冲器 44 包括输入缓冲器 44-1，传输门 44-2 和 OR 电路 44-3。

地址寄存器 A 20 包括地址锁存器 101，传输门 102，地址锁存器 103，传输门 104，传输门 105，地址锁存器 106，和传输门 107。进一步，地址寄存器 B 21 包括地址锁存器 111，传输门 112，地址锁存器 113，传输门 114，传输门 115，地址锁存器 116，和传输门 117。

刷新地址寄存器 19 包括刷新地址计数器/寄存器 19-1，倒相器 19-2，和传输门 19-3。由刷新地址计数器/寄存器 19-1 产生和保持刷新地址。

通过上述电路配置的操作，当从器件外输入 Read 指令或 Write 指令时，将与指令一起进入的地址传输到地址锁存器 101 或 111。在 Read 指令的情形中将地址传输到地址锁存器 105 或 116 而不需任何时间操作。在 Write 指令的情形中在取得一系列写数据的最后一个数据的定时将地址传输到地址锁存器 103 或 113。

如图 8 的电路配置所示，响应与各从判优器 15 传输到 DRAM 芯 14 的指令信号 RA3, WA3, RB3, WB3 和 REF3 对应的脉冲信号 RA3P, WA3P, RB3P, WB3P 和 REF3P，将地址信号从一个地址锁存器传输到 DRAM 芯 14。

图 9 是与数据输出有关的电路方框图。

与数据 I/O 电路 33 的数据输出有关的部分包括数据锁存器 121，传输门 122，数据锁存器 123，并行串行变换器 124，输出缓冲器 125 和传输信号发生电路 126。而且，与数据 I/O 电路 43 的数据输出有关的部分包括数据锁存器 131，传输门 132，数据锁存器 133，并行串行变换器 134，输出缓冲器 135 和传输信号发生电路 136。

从存储器阵列 51 读出的数据被读出缓冲器 55 放大，分别通过传输门 A 24 或传输门 B 25 加到数据 I/O 电路 33 或数据 I/O 电路 43。如果执行的指令与从 A 端口 11 的数据读出有关，则传输门 A 24 打开，而如果执行的指令与从 B 端口 12 的数据读出有关，则传输门 B 25 打开。以这种方式提供的数据被数据锁存器 121 或 131 锁存和保持。

传输门 122 或 132 响应从传输信号发生电路 126 或 136 提供的传输信号在一个相应的端口接收读指令后打开预定等待时间。于是分别将数据锁存器 121 或 131 的数据传输到数据锁存器 123 或 133。此后用并行串行变换器 124 或 134 将数据从并行数据变换到串行数据。然后将该数据传输给输出缓冲器 125 或 135，并从那里输出。

图 10 是表示传输信号发生电路 126 或 136 的配置的电路图。

传输信号发生电路 126 或 136 包括触发器 141 到 144 和多路复用器 145。将读指令信号 RA1 或 RB1 加到触发器 141，并连续地与时钟信号 CLKA1 或 CLKB1 同步地从一个触发器传输到下一个。将等待

时间信息 A 和 B 加到多路复用器 145。这个等待时间信息例如用时钟周期的数目确定等待时间的长度。根据等待时间信息，多路复用器 145 选择一个相应的触发器的 Q 输出，并将它作为数据传输信号输出。

图 11 是与数据输入有关的电路方框图；

与数据 I/O 电路 33 的数据输入有关的部分包括数据输入缓冲器 151，串行并行变换器 152 和数据传输装置 153。与数据 I/O 电路 43 的数据输入有关的部分包括数据输入缓冲器 154，串行并行变换器 155 和数据传输装置 156。

分别用串行并行变换器 152 或 155 将串行地输入到数据输入缓冲器 151 或 154 的数据转换成并行数据。当输入最后一个数据时，将并行数据传输到写数据寄存器 A 22 或写数据寄存器 B 23。当将 Write 指令从判优器 15 传输到 DRAM 芯 14 时，响应表示与 Write 指令到 DRAM 芯 14 的传输相应的定时的信号 WA3P 或 WB3P，将写数据寄存器 A 22 或写数据寄存器 B 23 的数据传输到 DRAM 芯 14。

图 12 是表示当连续进入 Read 指令时实施的操作的定时图。

A 端口 11 和 B 端口 12 分别与具有不同频率的时钟信号 CLKA 和 CLKB 同步地操作。在这个例子中，A 端口 11 用最大时钟频率操作，而 B 端口 12 用较低的时钟频率操作。

A 端口 11 具有下列设置：读指令周期=4 (CLKA)，数据等待时间=4，和脉冲串长度=4。B 端口 12 具有下列设置：读指令周期=2 (CLKA)，数据等待时间=2，和脉冲串长度=2。在每个端口的模寄存器中设置数据等待时间和脉冲串长度。

将由端口接收的指令存储在各指令寄存器中。将刷新指令存储在刷新指令寄存器中。判优器监视这些指令寄存器，并以接收指令的次序将指令传输给 DRAM 芯。当完成上一个指令的处理时传输下一个指令。

将从 DRAM 芯读出的数据从读出缓冲器传输到各端口数据锁存器（请参见图 9）。以后将数据从并行变换成串行，作为脉冲串输出与外部时钟同步地输出。

虽然曾经从 A 端口输入刷新指令，但是不影响 B 端口的操作，如图 12 所示。

图 13 是表示当连续输入 Write 指令时实施的操作的定时图。

在写操作时从器件外部输入的数据取脉冲串输入的形式。Write 指令存储在写指令寄存器中的定时是输入脉冲串输入的最后一个数据的定时。

如图 13 所示，从 A 端口提供的刷新指令不影响 B 端口的操作。

图 14 是表示当 A 端口和 B 端口两者都操作在最大时钟频率上时的定时图。

如图 14 所示，在这些端口的时钟信号之间可能存在相位差。两个端口具有下列设置：读指令周期=4，数据等待时间=4，和脉冲串长度=4。如从图可见的那样，甚至当两个端口都操作在最大时钟频率和连续输入 Read 指令时关于操作也没有问题。

图 15 是表示当 A 端口和 B 端口两者都操作在最大时钟频率上时的定时图。在图 15 中，两个端口连续地接收 Write 指令。

如图 15 所示，在这些端口的时钟信号之间可能存在相位差。两个端口具有下列设置：写指令周期=4，数据等待时间=4，和脉冲串长度=4。如从图可见的那样，甚至当两个端口都操作在最大时钟频率和连续输入 Write 指令时也能够进行适当的操作。

图 16 是表示当指令从 Read 指令改变到 Write 指令时的各操作的定时图；

如图 16 所示，当与“Write→Read”或“Read→Write”的指令间隔比较时指令传输“Write→Read”需要一个额外的指令间隔。这是因为我们传输 Write 指令以便在当进入脉冲串输入的最后一个数据时的定时对它进行处理。相反地，为了对它进行处理传输一个 Read 指令的定时定义为进入 Read 指令的定时，使得当相继的指令是“Write→Read”时需要提供一个额外的指令间隔。这样一个需要可以认为是将取脉冲串输入形式的输入数据变换成并行数据这个事实引起的。如果只输入一块数据而不是如脉冲串输入那样输入 4 块数据，则甚至当两

个相继的指令都是“Write→ Read”时也不需要提供一个额外的指令间隔。

在这种如对于一个 Write 写指令只输入一块数据那样的配置中，即使用与“Write→ Write”或“Read → Read”的情形中相同的指令间隔，也能够对于“Write→ Read”指令连续性适当地进行操作。

图 17 是表示当指令从“Read”改变到“Write”时输入刷新指令的定时图。

在图的顶部，表示出应该进入刷新指令的定时。在如说明的期间中在任何定时都能适当地进入刷新指令。例如，即便在图 17 所示的定时输入刷新指令，刷新指令只有当完成前一个 Write 写指令时才开始刷新操作，直到将刷新指令保存在备用状态时为止。因此，只要刷新指令落在与这个备用状态对应的期间内在任何定时都能适当地进入刷新指令。

图 18 是表示当使一个端口去激活时实施的操作的定时图；

如图 18 所示，当使一个端口（即，图 18 中的 A 端口）去激活时，根据刷新定时器内部产生刷新指令，从而执行刷新操作。

图 19 是表示当使两个端口去激活时实施的操作的定时图。

如图 19 所示，当使两个端口去激活时，根据刷新定时器内部产生刷新指令，从而执行刷新操作。

图 20A 和 20B 是表示 DRAM 芯操作的定时图。

图 20A 表示读操作的情形，图 20B 表示写操作的情形。在如图 20A 和 20B 所示的操作定时，在完成整个操作前通过字线选择，数据放大，写回，和预充电的相继操作发出进入的指令。

图 21 是表示当只使一个端口操作时实施的两倍速率操作的定时图；

通过使两个端口中的一个停止操作，可以将到操作端口的指令输入间隔缩短一半。当这种情形发生时，外部指令的最快周期和内部动作的最快周期彼此相同。在图 21 的例子中，缩短了指令间隔而没有改变时钟频率。在这个情形中，因为脉冲串长度也变得较短，所以数据

传输速率与当用两个端口时的相同。

图 22 是表示当通过使时钟频率两倍高使数据传输速率两倍时两倍速率操作的定时图；

在图 22 中，当使两个端口中的一个停止操作时，将输入到操作端口时钟信号设置在高两倍的频率上。与此相关，指令输入的时间间隔缩短一半。在这个情形中，因为脉冲串长度与当用两个端口时的相同，所以数据传输速率为当用两个端口时的两倍那样快。

此外，因为只将外部时钟信号输入 I/O 电路装置，所以如果将该电路装置设计得能应付高速操作则实际上容易完成两倍速率操作。

图 23 是用于解释本发明的第 2 实施例的图；

一般，根据其用度扩大存储器。这同样应用于多端口存储器的情形，可能存在为了扩大存储空间提供多个多端口存储器的情形。

多端口存储器包括判优器，并检测哪一个指令较早进入各端口，接着以检测出的次序执行指令。甚至当在几乎相同的定时将指令输入各端口时，也为相继地执行指令确定一个次序。在图 23 所示的例子中，提供多个多端口存储器 200-1 到 200-n，从 A 端口控制器 201 和 B 端口控制器 202，将相同的指令加到多端口存储器 200-1 到 200-n。即便同时将指令加到 A 端口和 B 端口，由于信号线有不同的长度和/或电源噪声的影响，指令到达每个多端口存储器的相对定时也可能稍有不同。在这个情形中，每个多端口存储器的判优器能够以从存储器到存储器不同的次序执行指令。

如果到 A 端口的指令和到 B 端口的指令指向不同的地址，则存储器件之间执行指令的不同次序可能不会成为一个问题。然而，当各指令是对于同一个地址时，就会发生问题。

例如，在当写存取同一个存储单元后读出数据时与当写存取同一个存储单元前读出数据时之间检索的数据是不同的。而且，当在写入 A 端口的数据后写入 B 端口的数据时 B 端口的数据保留在存储器中，而如果以相反的次序进行操作则 A 端口的数据保留在存储器中。

如果以上面描述的方式从存储器到存储器执行指令的次序不同，

则关于数据的可靠性就存在严重的问题。

因此，当用多个多端口存储器时，需要使判优器作出的决定在存储器之间保持一致。为此，本发明的第2实施例指定多端口存储器中的一个为主器件200-1，并用其余的器件作为从器件200-2到200-n。从器件遵守由主器件的判优器作出的决定。

图24是表示根据本发明的多端口存储器的第2实施例的方框图。本例子的配置具有两个端口即A端口和B端口。

与图4所示的第1实施例的不同包括A端口11A和B端口12A分别具有BUSY信号I/O装置36和46这个事实和提供地址比较器26比较A端口的地址和B端口的地址这个事实。如果地址比较器26检测出地址匹配，因此产生匹配信号，则判优器15A将转变DRAM芯的操作模式以便开始连续模式。

图25A和25B是用于解释连续模式的定时图。

如图（图20）表示的第1实施例的操作所示，将DRAM芯的操作分成ROW（行）操作和COLUMM（列）操作。在本发明中，进行ROW操作，COLUMM操作和预充电操作作为一系列的连续执行操作，这定义了单个内部操作周期。

在第2实施例中的连续模式与通常的DRAM的列存取操作相同，对于同一个存储单元反复执行一个指令。即，这个模式在ROW操作后多次执行COLUMM操作后进行预充电。当连续地加上对同一个存储单元地址的Write指令时，执行后面的指令而不执行前面的指令。这是因为即便相继地执行了这些Write指令，由前面的指令写入的数据将被后面的指令的数据覆盖。

如图25A所示，连续模式允许使操作缩短到比通常内部操作的2个周期短，从而提供额外的时间。将由这个额外时间得到的边边缘分配给在ROW操作和COLUMM操作之间的一个点（下文中将这个边边缘称为Wait（等待）期间）。在这个Wait期间中，实施用于使主器件和从器件之间的指令执行次序一致的处理。

下面，我们说明用BUSY信号使主器件和从器件之间的操作一致

的过程。

为了保证在主器件和从器件之间有相同的指令执行次序用 **BUSY** 信号。**BUSY** 信号 I/O 装置 36 和 46 用作输出主器件 200-1 中的 **BUSY** 信号的 **BUSY** 输出电路,和用作在从器件 200-2 到 200-n 中接收 **BUSY** 信号的 **BUSY** 输入电路。将指示主器件标识或从器件标识的信息存储在模寄存器 31 或 41 中。

存储器件接收来自一个端口的指令,开始如图 20A 和 20B 所示的操作。

当从其它的端口输入指令,存取在 **ROW** 操作期间的同一个地址时,地址比较器 26 产生一个匹配信号。在重复这个匹配信号时,判优器 15A 向 **DRAM** 芯 14 的控制电路 53 提供连续模式信号。响应连续模式信号,**DRAM** 芯 14 转移到连续模式如图 25B 所示。

在 **Wait** 期间中,主器件 200-1 根据判优器 15A 作出的决定产生 **BUSY-A** 信号或 **BUSY-B** 信号。在这个例子中,对于一个被判优器 15A 识别的较早已经收到指令的端口产生 **BUSY** 信号。

类似地,在 **Wait** 期间中,从器件检测由主器件产生 **BUSY** 信号,改变由它自己的判优器 15A 作出的决定,以便如果它不同于 **BUSY** 信号的指示就遵从主器件。然后根据改变了的指令次序实施 **COLUMN** 操作。

图 26 是表示当对于 A 端口的 **Read** 指令和 B 端口的 **Write** 指令产生 **BUSY** 信号时实施的操作的定时图。

在这个实施例中,**BUSY** 信号具有一个指示选择的逻辑级“L”。而且,优先地传输和非同步地接收 **BUSY** 信号。这是因为需要在有限的 **Wait** 期间内迅速地交换 **BUSY** 信号。

在图 26 的例子中,因为 A 端口的 **ReadA2** 比 B 端口的 **WriteB2** 早,所以主器件在 **Wait** 期间中产生指示 A 端口的 **BUSY** 信号。从器件接收这个 **BUSY** 信号,并依靠 A 端口的 **ReadA2** 比 B 端口的 **WriteB2** 早。然后,主器件和从器件以首先 **ReadA2** 然后 **WriteB2** 的次序执行在连续模式中的列操作。

图 27 是表示当对于 A 端口的 Read 指令和 B 端口的 Write 指令产生 BUSY 信号时实施的操作的定时图。而图 26 说明 A 端口的 Read 指令较早的情形。图 27 表示 B 端口的 Write 指令较早的情形。

图 28 是表示当对于 A 端口的 Write 指令和 B 端口的 Write 指令产生 BUSY 信号时实施的操作的定时图

图 28 所示的操作例子是关于 A 端口的 Write 指令比 B 端口的 Write 指令早的情形。即, 因为 A 端口的 WriteA2 比 B 端口的 WriteB2 早, 所以产生指示 A 端口的 BUSY 信号, 并加到从器件上。在这种情形中, 因为通过执行 A 端口的 Write 指令被写入的数据将立即被覆盖, 所以只有 B 端口的写指令因为 WriteB2 因为它较后进入而被执行。

图 29 是表示当对于 A 端口的 Write 指令和 B 端口的 Write 指令产生 BUSY 信号时实施的操作的定时图。

图 29 所示的操作例子是关于 B 端口的 Write 指令比 A 端口的 Write 指令早的情形。在这种情形中, 因为通过执行 B 端口的 Write 指令被写入的数据将立即被替代, 所以只有 A 端口的写指令 WriteA2 被执行。在这个例子中, 将 A 端口的时钟频率设置得稍低于 B 端口的时钟频率。虽然当比较指令 WriteA2 和 WriteB2 时对于 A 端口指令输入稍早, 但是在接收最后一个数据输入时是 B 端口较早。因此, 确定 B 端口的 Write 指令比 A 端口的 Write 指令早。

上面提供的描述还没有参考关于 A 端口的 Read 指令和 B 端口的 Read 指令的组合起来的情形。因为不管相对的定时如何, 数据的可靠性不受影响, 所以在这个情形中不需要产生 BUSY 信号。

图 30 是表示在能够处理由控制器发出的中断指令的配置中的操作的定时图。

“中断指令”是当开始 BUSY 信号时指令改变由主器件的判优器作出的决定的指示。造成中断的方法包括:

- a) 作为指令输入;
- b) 提供专用的端子销;
- c) 用特定的地址组合; 和

d) 用 **BUSY** 信号。

方法 d) 用控制器对与其产生 **BUSY** 信号的端口不同的端口提供 **BUSY** 信号，并安排主存储器和从存储器对它进行检测。

在图 30 的例子中，当对于 A 端口的 Write 指令和 B 端口的 Write 指令发生 **BUSY** 信号时产生中断。如将图 28 和图 29 结合起来进行的描述那样，当 Write 和 Write 的组合产生 **BUSY** 信号时只有 A 端口的 Write 指令和 B 端口的 Write 指令中的一个被执行。结果，将失去较早进入的数据。

在图 30 中，A 端口的 WriteA2 比 B 端口的 WriteB2 早，使产生指向 A 端口的 **BUSY** 信号。接收到主器件产生的 **BUSY** 信号后，控制器产生中断指令以便防止删除 A 端口的 Write 数据。

主器件和从器件从控制器接收中断指令，改变由判优器作出的决定，接着根据中断指令在等待期间结束后执行 Write 操作。即，判优器改变它们的决定指出 A 端口的指令 WriteA2 比 B 端口的指令 WriteB2 晚，实施与 WriteA2 有关的写操作。这能够防止 A 端口的写数据被删除。在 Write→Write 组合的情形中，全部需要只是执行写操作一次，使得可以分配比 Read→Write 组合或 Write→Read 组合的连续模式较长的等待期间。于是可以利用这个等待期间响应 **BUSY** 信号执行中断指令。

下面，我们描述用于实现上述操作的地址比较器，**BUSY I/O** 系统和中断系统的配置。

图 31 是表示根据本发明的第 2 实施例的多端口存储器的地址比较器，**BUSY I/O** 系统，和中断系统的配置的图。

地址比较器 26 比较存储在地址寄存器中的地址，并当 A 端口 11 的地址和 B 端口 12 的地址之间存在匹配时输出匹配信号。而且，为了指出哪两个地址是匹配地址，产生信号 ARA, AWA, ARB 和 AWB。例如，当 A 端口的 Write 指令的地址和 B 端口的 Write 指令的地址显示匹配时，将 AWA 和 AWB 设置在“H”。NAND 电路 208 到 210 每个都得到这些信号的一个逻辑 NAND，使得 N1, N2 和 N3 中的一个变

成“L”。

在图 31 左边（在地址比较器 26 的下面）提供了 BUSY 信号 I/O 装置 36 和 46 与中断电路。根据模寄存器 31 或 41 的设置，BUSY 和 I/O 硬件控制装置 211 在主器件情形中响应匹配信号的检测产生激活信号（主），和在从器件情形中产生激活信号（从）。激活信号（主）激活 BUSY 输出电路 212 和 213，而激活信号（从）激活 BUSY 输入电路 214 和 215。

在判优器中，将选出的指令作为指令次序中的第一个输出到输出端 RA3, WA3, RB3 和 WB3 中的一个（即输出端中的一个为“H”）。在主器件情形中，RA3 到 WB3 被锁存器 216 和 217 响应信号 N4 锁存起来，信号 N4 是由与匹配信号的前沿对应的脉冲组成的。根据锁存的数据输出 BUSY-A 信号和 BUSY-B 信号。

在从器件情形中，如果接收到是“L”的 BUSY-A 信号，则将从中断电路 218 输出的信号 N10 设置在“L”。如果接收到是“L”的 BUSY-B 信号，则将从中断电路 219 输出的信号 N11 设置在“L”。当信号 N10 和 N11 处于去激活状态时，它们是“H”，当检测出 BUSY 信号或中断指令时它们变成“L”。

中断检测装置 220 检测从控制器提供的中断指令，并输出中断信号 A 或 B。给予中断信号对进入的 BUSY 信号的优先权，并将它们作为信号 N10 和 N11 传输出去。

图 31 底部所示的三个比较器 80-3, 80-5 和 80-6 是判优器 15A 的比较电路的一部分（请参见图 6A 和图 24）。这些比较器对于需要 BUST 确定的指令组合进行比较。

图 32 是表示一个主器件的操作的定时图。图 33 是表示一个从器件的操作的定时图。

这三个定时图说明 A 端口的 Read 指令的地址和 B 端口的 Write 指令的地址相互匹配的情形。图 32 的主器件决定 A 端口较早，图 33 的从器件决定 B 端口较早。在这种情形中，主器件的比较器 80-3 输出是“L”的 N21 和是“H”的 N22。进一步，从器件的比较器 80-3 输出

是“H”的 N21 和是“L”的 N22。主器件产生 BUSY-A 信号，从器件在接收 BUSY-A 信号时将 N10 改变成“L”。因为在这个时间点 N1 是“L”，所以通过 NOR (“或非”) 电路 221 和倒相器 222 将 N10 的 LOW 信号加到从器件的比较器 80-3 上。对应地，从器件的比较器 80-3 的输出改变到是“L”的 N21 和是“H”的 N22。在这种方式中，改变了由判优器作出的决定。

现在我考虑与上述情形相反的 A 端口的 Write 指令的地址和 B 端口的 Read 指令的地址相互匹配的情形。在这种情形中，从器件的比较器 80-5 输出被改变，从而改变在从器件中由判优器作出的决定。

比较 WA2 和 WB2 的比较器 80-6 具有一个不同于比较器 80-3 和 80-5 的外围电路配置。这是因为当响应 Write 和 Write 组合产生 BUSY 信号时，A 端口的指令和 B 端口的指令中只有一个将被保留。

图 34 是表示当两个端口的写地址相同时实施的主器件的操作的定时图。图 35 是表示当两个端口的写地址相同时实施的从器件的操作的定时图。

现在我们考虑如图 34 所示主器件决定 A 端口较早，和如图 35 所示从器件决定 B 端口较早的情形。在地址比较器 26 刚刚产生一个匹配信号的瞬间，主器件的比较器 80-6 输出是“L”的 N25 和是“H”的 N26，和从器件的比较器 80-6 输出是“H”的 N25 和是“L”的 N26。主器件将 RA3, WA3, RB3 和 WB3 锁存在这个状态，并输出一个 BUSY-A 信号。

当如在本情形中那样在 Write-Write 组合中产生 BUSY 信号时，需要删除一个已较早进入的 Write 指令。为了这个目的，提供倒相器 231，NOR 电路 232，NAND 电路 233 和 234，倒相器 235 和 236。响应匹配信号，HIGH 边沿脉冲电路 230 产生信号 N4 的“H”脉冲。通过某个逻辑操作将信号 N4 和信号 N3 组合起来，产生信号 N31 中的“H”脉冲。在这个例子中，N26 对于主器件是“H”，使 N33 产生“H”脉冲，导致 N25 改变成“H”和 N26 改变成“L”。这里，延迟电路 237 和 238 用于提供一个能够用来在改变发生前产生 BUSY 信号的额外时间，并

防止当将已经改变了的状态反馈回到 NAND 电路 233 和 234 时再次被改变。在从器件中，将 N25 改变成“L”，将 N26 改变成“H”。

如以前描述的那样，主器件产生 BUSY-A 信号，接收这个信号的从器件使它的 N10 改变成“L”。因为在这个特定瞬间 N3 是“L”，所以从器件的比较器 80-6 再次被反转，导致使 N25 改变成“H”，使 N26 改变成“L”。

延迟电路 250 接收信号 N4，并使这个信号延迟一个预定时间长度，从而产生一个 Wait 期间。这里，当选择 N1 或 N2 时选择 Delay（延迟）（ t_1 ），当选择 N3 时选择 Delay（ t_2 ）。

提供 NAND 电路 251 和 252 与倒相器 253 和 254 用于当 Wait 期间结束时从指令寄存器清除被跳过的 Write 指令。例如，如果在 Wait 期间结束时 N25 是“L”和 N26 是“H”，则将执行 A 端口的 Write 指令。因此，为了从寄存器删除 B 端口的 Write 指令产生 RESTWB2。因为在 Wait 期间中需要通过 BUSY 接收或中断改变决定，所以在这个期间指令寄存器中的指令被完整地保留下来。

图 36 是表示当两个端口的写地址相互匹配使控制器发出中断指令时主器件操作的定时图。图 37 是表示当两个端口的写地址相互匹配使控制器发出中断指令时从器件操作的定时图。

如图 36 所示，在主器件中的指令选择状态由于中断而反转，而且，如图 37 所示，在从器件中的指令选择状态由于 BUSY 信号而反转，然后进一步由于中断而反转。这里，由于中断而反转状态的操作与由于 BUSY 信号而反转状态的操作相同，我们将省略它的详细描述。

在上述第 2 实施例的操作中，设计从一个给定的指令到下一个接着的指令扩展的指令周期使它甚至在产生 BUSY 信号或中断指令后也不改变。

在图 26 中，例如，虽然响应 ReadA2 发生 BUSY 信号，ReadA2→ReadA3 的指令间隔与 ReadA1→ReadA2 的指令间隔相同。要求在 Wait 期间处理 BUSY 信号和中断信号。因为这个原因，当由于长的系统总线，大量的从器件，控制器的慢响应等 BUSY 信号或中断信号的

交换需要长时间时，就需要较长的 Wait 期间。

为了消除这个问题，当延迟跟随 BUSY 信号和中断信号的下一个指令输入时可以扩展 Wait 期间。即，当加长 Wait 期间时可以扩展 ReadA2→ReadA3 的指令间隔使它比在图 26 中的 ReadA1→ReadA2 的指令间隔长。

为了延迟一个指令输入，可以在设计书中说明指令输入的延迟，和可以设计控制器使它根据数据表进行操作。如图 31 所示通过加长延迟电路 250 的延迟时间达到 Wait 期间的扩展。如果根据用度需要调整 Wait 期间，则在延迟电路 250 中可以提供两条或多条延迟线，使通过一个模寄存器的设置改变延迟长度的设置成为可能。

当 Wait 期间以这种方式扩展时，除了响应 Write-Write 指令组合产生 BUSY 信号的情形外在其它情形中也能够提供长的 Wait 期间。考虑到这一点，甚至当响应 Read-Write 或 Write-Read 指令组合出现 BUSY 信号时控制器也可以发出一个中断指令。

在上面描述的本发明中，当指令进入 N 个端口时，在任何给定的端口的一个最小指令周期内一个接着一个地执行与 N 个端口对应的所有的 N 指令。因此，一个与任何给定端口有关的存取操作出现在器件外部中在最小指令周期内被实施。在这个情形中，只有当从多个端口存取同一个地址时才会发生 BUSY 信号。于是可以得到 BUSY 信号发生概率，该概率与 SRAM 型多端口存储器的 BUSY 信号发生概率一样低。

而且，在本发明的半导体存储器件中，内部电路包括一个存储单元阵列，该阵列由各动态型存储单元和一个定义存储单元被刷新的定时的刷新电路组成。在第 1 模式中，响应输入到在 N 个端口中的至少一个的刷新指令刷新存储单元，在第 2 模式中，在刷新电路确定的定时刷新存储单元。

即，上面描述的本发明具有第 1 操作模式，在该模式中响应来自一个外部端口的指令实施刷新操作，和第 2 操作模式，在该模式中响应内部刷新电路实施刷新操作。因为这个配置，允许一个外部端口作

为一个用于刷新管理的端口进行操作，以便在恒定的间隔接收刷新指令，或者如果这个用于刷新管理的端口处在去激活状态则内部刷新电路实施刷新操作。这使根据系统配置以一种灵活的方式管理刷新操作成为可能。

[本发明的第 2 方面]

下面我们描述本发明的第 2 方面。

存在若干种多端口存储器。下文中，涉及具有多个端口的存储器，并允许从各端口相互独立地存取一个公共存储器阵列。例如，两个端口型的多端口存储器装备有一个 A 端口和一个 B 端口，并允许从与 A 端口链接的 CPU-A 和从与 B 端口链接的 CPU-B 独立地进行到公共存储器的读/写存取。

作为这类多端口存储器，具有 SRAM 存储器阵列的存储器是已知的，其中在复制的设置中提供各字线和各位线对，每个存储单元都与 2 组字线和位线对连接。然而，这种多端口存储器具有电路密度低的问题，其中需要提供字线和位线对的复制组。

为了消除这个问题，可以用与具有多个处理器配置的计算机所用的共用存储器相同的机构。共用存储器具有提供给公共存储器的多个端口。典型地，将 SRAM 用作存储器，并用离散的 IC（集成电路）制成多个端口。当从多个端口同时进行存取时，因为存储器阵列是共用的，所以不能同时进行与多个端口相应的操作。防止这种问题发生的最简单的方法是对每个端口产生 BUSY 信号以便防止当从一个端口进行存取时到另一个端口的存取。然而，这引起限制存储器用度的问题。考虑到这一点，为公共存储器提供称为判优器的判优电路，判优电路确定多个端口接收的存取要求的优先权。构造存储器阵列的控制器，以优先权的次序执行与存取要求对应的操作。例如，以到达的次序即以将存取要求加到各端口的次序处理存取要求。

在这个情形中，存储器阵列最终随机地从多个端口被存取。因此，不能提供在同一个行地址连续地存取相继的列地址的列存取操作，而

这样的列存取操作典型地可以在 DRAM 中得到。即，选择，为读/写操作存取和复位存储单元，响应单个存取实施所有这些操作。

当制成一个共用存储器时，一般地，按常规将 SRAM 用作存储器阵列。这是因为 SRAM 能够进行高速随机存取操作，并且因为不需要刷新操作所以能够容易地使用 SRAM。然而，单块芯片的多端口存储器常规地具有以字线和位线对的复制组，在具有通常的 SRAM 配置的存储器阵列的基础上的单块芯片的多端口存储器还没有在实践中使用。

总结一下，多端口存储器和共用存储器是用 SRAM 制成的，而不用需要刷新操作的 DRAM。

当系统不断地提供高性能时要被处理的数据量增加，并且多端口存储器也需要具有大的容量。可以用动态型存储单元 (DRAM) 阵列制成多端口存储器，DRAM 阵列比 SRAM 具有较高的电路密度，从而以低的成本提供具有大存储容量的多端口存储器。然而，存储单元的刷新操作成为一个问题。

在常规的 DRAM 中，需要从器件外部以恒定间隔在读/写指令之间提供刷新指令。为此，在以 DRAM 为基础的系统中的控制器器件具有用于刷新管理的定时器和/或控制电路。然而，在用以 SRAM 为基础的多端口存储器的系统中不提供这样的电路。甚至在以 DRAM 为基础制成存储器的情形中，在这些系统中需要能以与常规的多端口存储器相同的方式使用这种存储器。即，具有由 DRAM 组成的存储器阵列的多端口存储器需要由它自己来进行刷新操作。

本发明的目的是提供具有由 DRAM 芯组成的存储器阵列，而且不需要考虑刷新操作就能使用的多端口存储器，从而以低的成本提供具有大存储容量并且容易使用的多端口存储器。

图 38 是用于解释本发明原理的图，表示当对于两个端口实施读操作时的情形。

在能够实施 3 个内部操作周期的最小间隔上提供加到两个外部端口，A 端口和 B 端口的指令。即，将一个外部指令周期设置在比 3 个

内部操作周期需要的持续时间长的长度上。分别将时钟信号 CLKA 和 CLKB 输入到 A 端口和 B 端口，与时钟信号同步地进行在器件的一个外部装置和各外部端口之间的地址和数据的交换。使地址（图中未画出）与指令同时进入。当在最小外部指令周期上使读指令进入 A 端口和 B 端口时，判优电路通过将优先权给予首先到达的输入信号，控制芯的操作。如上所述在一个外部指令周期中能够实施 3 个内部操作，在这个外部指令周期中在存储器阵列上执行 2 个读操作，接着将读数据输出到 A 端口和 B 端口。A 端口和 B 端口两者都保持检索的数据，并在下一个跟随的外部指令周期开始时，即与从输入读指令的第 4 个时钟信号同步地输出检索的数据。即，在这个情形中数据的等待时间是 4。

提供刷新定时器作为内部电路，刷新定时器在它自己身上产生刷新指令。因为如上所述在一个外部指令周期中能够实施 3 个内部操作，所以当产生刷新指令时在单个外部指令周期中能够执行指令 A，指令 B 和刷新指令。在下一个跟随的外部指令周期开始时输出读数据。在这个方式中，可以从器件外部存取多端口存储器而与刷新操作没有任何关系。

在图 38 的例子中，响应一个读指令输出一项读数据。即，脉冲串长度为 1。所以，在一个时钟脉冲周期中完成读数据的输出后，在外部指令周期的 3 个余下的时钟周期中外部端口不输出任何数据，这导致无效的数据传输。可以通过加长脉冲串长度来消除这个问题。

图 39 是用于解释本发明原理的图，表示脉冲串长度为 4 的例子。在这个例子中，与上述情形相同，将 2 个外部端口的外部指令周期设置在能够供应 3 个内部操作周期的长度上。进一步，一个外部指令周期对应 4 个时钟周期。与时钟信号同步地在单个外部指令周期中从一个外部端口 4 次输出数据。所以，如果根据一个外部指令周期的时钟周期的数目设置脉冲串长度，则在两个端口都能实现无间隙的读操作，从而非常大地增加了数据传输速率。在这个情形中，要求与脉冲串长度一样多的数据项响应单个存取内部地输入存储器阵列或内部地从存

存储器阵列输出。例如，如果外部端口的数据输入/输出线的数目为 4，和脉冲串长度为 4，则需要保证由单个存取操作从存储器阵列输出 16 位数据或将 16 位数据输入存储器阵列。

我们应该注意到 A 端口和 B 端口不一定同步操作，并且只要将最小周期设置得等于 3 个内部操作周期需要的持续时间，就可以将各外部指令周期相互独立地设置在任何定时。

而且，外部端口的数目也可以是任何数目。如果将外部端口的数目设置为 n ，则将每个端口的外部指令周期设置在能够进行 $n+1$ 个内部操作周期的最小周期上。如果满足这个要求，则甚至当刷新操作被执行时在一个外部指令周期中也可以实施各端口要求的所有操作，从而允许使用多端口存储器而与刷新操作没有任何关系。

图 40 和图 41 是表示当有 2 个，3 个和 n 个端口时在一个最小外部指令周期和各内部操作周期之间的关系的图。

如这些图所示，如果端口的数目为 2，则最小外部指令周期具有能够供应 3 个内部操作的长度，并且如果端口的数目为 3，则最小外部指令周期是可以进行 4 个内部操作的时间长度。进一步，如果端口的数目为 n ，则最小外部指令周期等于可以执行 $n+1$ 个内部操作的时间长度。

图 42 和图 43A 到 43C 是表示根据本发明的实施例的多端口存储器的配置的图。图 42 表示 DRAM 芯和它的相关电路，图 43A 表示 A 端口，图 43B 表示 B 端口。进一步，图 43C 表示刷新电路。图 43A 到 43C 所示的电路与图 42 的各部分连接。

如这些图所示，这个实施例的多端口存储器包括 DRAM 芯 2011，用于控制确定操作次序并保证以确定的次序实施操作的判优器 2026，多组暂时存储指令，地址和数据的寄存器，2 个由 A 端口 2030 和 B 端口 2040 组成的外部端口，和刷新电路 2050。

A 端口 2030 和 B 端口 2040 分别包括模寄存器 2031 和 2041，CLK 缓冲器 2032 和 2042，数据 I/O 电路 2033 和 2043，地址输入电路 2034 和 2044，与指令输入装置 2035 和 2045，它们根据从器件外部提供的

各分开的时钟频率进行操作。将数据等待时间和脉冲串长度存储在模寄存器 2031 和 2041 中，使它们能被分别地设置。数据 I/O 电路 2033 和 2043 装备有根据脉冲串长度实施输出/输出数据的并行到串行变换和串行到并行变换的机构。

刷新电路 2050 包括刷新定时器 2051 和刷新指令发生器 2052。刷新定时器 2051 在预定间隔上产生刷新开始信号，刷新指令发生器 2052 对应地产生刷新指令。

分别将加到 A 端口和 B 端口的指令，地址和写数据存储在寄存器中。也将刷新指令存储在刷新指令寄存器 2027 中，并将刷新地址存储在刷新地址计数器/寄存器 2018 中。

判优器 2026 根据指令到达的次序确定执行指令的次序，并以确定的次序将指令传输给 DRAM 芯 2011 的控制电路 2014。进一步，判优器 2026 将传输信号传输给对应的地址寄存器和对应的数据寄存器（在写操作情形）。在 DRAM 芯 2011 中，控制电路 2014 响应提供的指令，控制译码器 2013，写放大器（WriteAmp）2015，和读出缓冲器 2016，从而实施对于存储器阵列 2012 的存取操作。在写操作的情形中，译码器 2013 为了进行写操作对存取的地址进行译码，以便激活在存储器阵列 2012 中的字线和列信号线，导致将存储在写数据寄存器 A2022 和 B2023 中的写数据通过 WriteAmp 2015 写入存储器阵列 2012。在读操作的情形中，以类似的方式存取存储器阵列 2012，导致通过传输门 A 2024 和 B 2025 将读数据从读出缓冲器 2016 传输到各端口的数据输出电路。根据 DRAM 芯 2011 的操作周期控制传输门的传输定时，并由控制电路 2014 确定传输定时。与对应的外部时钟信号同步地从每个端口的数据输出电路输出输出数据。

下面，我们描述与指令处理，地址处理和数据处理中的每一个有关的详细情形。

图 44 和图 45 是表示根据第 1 实施例与指令处理有关的装置配置的图。在与图 42 和图 43A-43C 中相同的部件上加上相同的参照数字。这也同样应用于其它的图。

如图 44 所示, A 端口的指令输入装置 2035 包括输入缓冲器 2036, 指令译码器 2037 和 $(n-1)$ 时钟延迟电路 2038, B 端口的指令输入装置 2045 包括输入缓冲器 2046, 指令译码器 2047 和 $(m-1)$ 时钟延迟电路 2048。这里, n 和 m 是脉冲串长度。而且, 如图 45 所示, 指令寄存器 A 2028 包括 Read 指令寄存器 AR 和 Write 指令寄存器 AW, 指令寄存器 B 2029 包括 Read 指令寄存器 BR 和 Write 指令寄存器 BW。

输入缓冲器 2036 和 2046 要求以与各时钟信号 CLKA1 和 CLKB1 同步地加上 Read 指令, 指令译码器 2037 和 2047 进行译码处理。指令译码器 2037 和 2047 在读指令情形中分别产生 RA1 和 RB1, 而在写指令情形分别产生 WA1 和 WB1。分别将信号 RA1 和 RB1 传输给 Read 指令寄存器 AR 和 BR, 不需要任何定时操作, 而信号 WA1 和 WB1 分别被 $(n-1)$ 时钟延迟电路 2038 和 $(m-1)$ 时钟延迟电路 2048 延迟, 直到输入脉冲串数据的最后一个数据项为止, 接着被传输给 Write 指令寄存器 AW 和 BW。而且, 将由刷新电路 2050 产生的刷新指令 REF1 传输给刷新指令寄存器 2027。

判优器 2026 检测将指令传输给这 5 个指令寄存器 AR, AW, BR, BW 和 2027 的次序, 并以检测到的次序将这些指令一个接着一个地传输给 DRAM 控制电路 2014。DRAM 控制电路 2014 执行接收的指令, 产生信号 RESET1, 要求判优器 2026 当指令执行结束或接近结束时发送下一个指令。响应 RESET1 信号, 判优器使存储被执行指令的指令寄存器复位, 并将下一个指令传输给 DRAM 控制电路 2014。

图 46 是判优器 2026 的实施例。指令到达图 45 的 5 个指令寄存器的次序由比较器 2053 检测出来如该图 46 所示。每个比较器 2053 比较两个指令寄存器的定时, 将它的输出在首先输入“H”的情形中改变成“H”。AND 门 2054 通过检查相关的比较器 2053 的所有相关输出是否是“H”来确定给定的指令是否在所有 4 个其它指令前输入。如果一个响应的指令是最早的则响应各指令的信号 RA3, WA3, RB3, WB3 和 REF 变成“H”, 并且将响应的指令的地址等被传输给 DRAM 芯

2011, 。当 DRAM 芯 2011 执行该指令时, 从 DRAM 芯 2011 产生信号 RESET1, 并产生用于使被执行指令的指令寄存器复位的信号 (ReasetRA , ReasetWA 等)。当使被执行指令的指令寄存器复位时, 接收这个被执行指令的比较器 2053 的输出改变, 将在次序中的下一个指令传输给 DRAM 芯 2011。在这个方式中, 以指令输入的次序执行指令。

图 47 是表示根据第 1 实施例与地址处理有关的部分的配置的图。下文中, 在它的信号名称的末端具有字母“P”的信号代表具有从对应的信号名称的信号的前沿产生的脉冲的信号。如图所示, 地址输入电路 2034 和 2044 分别包括输入缓冲器 2057A 和 2057B 与传输门 2058A 和 2058B。进一步, 地址寄存器 A 2019 和地址寄存器 B 2020 分别包括地址锁存器 A1 和 B1, 传输门 2060A 和 2060B, 地址锁存器 A2 和 B2, 传输门 2062A 和 2062B, 与传输门 2063A 和 2063B。通过地址总线 2017 将从传输门 2062A, 2062B, 2063A 和 2063B 提供的地址传输给 DRAM 芯 2011。进一步, 通过传输门 2064 和地址总线 17 将从刷新地址计数器/寄存器 2018 提供的刷新地址传输给 DRAM 芯 2011。

当从器件外部输入 Read 指令或 Write 指令时, 分别通过传输门 2058A 或 2058B 将加到输入缓冲器 2057A 或 5027B 的地址和输入指令同时传输给地址锁存器 A1 或 B1。在 Read 指令的情形中, 通过传输门 2063A 或 2063B 与到 DRAM 芯的指令传输同步地将地址传送给 DRAM 芯 2011。在 Write 指令的情形中, 进一步在上次数据采集的定时将地址传输给地址锁存器 A2 或 B2, 然后, 与到 DRAM 芯的指令传输同步地通过传输门 2062A 或 2062B 传输给 DRAM 芯。进一步, 刷新地址计数器/寄存器 2018 产生并在其中保存刷新地址, 然后与到 DRAM 芯的刷新指令传输同步地通过传输门 2064 将该地址传输给 DRAM 芯 2011。

图 48 是表示根据第 1 实施例与数据输出有关的部分的配置的图。图 49 是表示图 48 的传输信号发生电路的图。A 端口 2030 和 B 端口 2040 的各数据 I/O 电路 2033 和 2043 分别包括为了输出数据的电路

2065A 和 2065B 与为了输入数据的电路 2074A 和 2074B，我们将在后面描述它们。如图所示，将通过读出缓冲器 2016 从存储器阵列 2012 读出的数据通过数据总线 2021 和传输门 2024 或 2025 分别传输给为了输出数据的电路 2065A 或 2065B。

为了输出数据的电路 2065A 和 2065B 分别包括数据锁存器 A1 或 B1，传输信号发生电路 2067A 和 2067B，传输门 2069A 或 2069B，数据锁存器 A2 或 B2，并行到串行变换器 2070A 和 2070B，以及输出缓冲器 2071A 和 2071B。

由 DRAM 芯 2011 的控制电路 2014 根据内部操作对传输门 2024 和 2025 进行控制。如果被执行的指令是 Read-A（即，对于 A 端口的读操作），则传输门 2024 将打开。如果被执行的指令是 Read-B，则传输门 2025 将打开。数据锁存器 A1 和 B1 在其中存储数据，然后在各端口中接收 Read 指令后的一个确定的等待时间，在这些端口通过传输门 2068A 或 2068B 引入这个等待时间，将数据传输给各数据锁存器 A2 或 B2。然后，由并行到串行变换器 2070A 和 2070B 对数据进行变换，接着分别传输给输出缓冲器 2071A 和 2071B，并从那里输出。

如图 49 所示，传输信号发生电路 2067A 和 2067B 采用一系列的触发器 2072 使各 Read 指令 RA1 和 RB1 延迟由等待时间设定确定的许多时间周期，从而产生数据传输信号 2002。因为来自传输门 2068A 或 2068B 的读数据传输响应数据传输信号 2002，所以读数据从读操作的定时开始被延迟与等待时间相当的许多周期后结束。

图 50 是表示根据第 1 实施例与数据输入有关的部分的配置的图。为了输入数据的电路 2074A 和 2074B 分别包括数据输入（Din）缓冲器 2075A 和 2075B，串行到并行变换器 2076A 和 2076B，以及数据传输装置 2077A 和 2077B。分别通过 Write 数据寄存器 2022 和 2023，数据传输装置 2078A 和 2078B，以及数据总线 21 将来自数据传输装置 2077A 和 2077B 的写数据传送给 WriteAmp 2015，并被写入存储器阵列 2012。

根据脉冲串长度将串行输入数据从串行变换到并行，然后在输入

最后一个数据项的定时传输给 Write 寄存器 2022 和 2023。当从判优器 2026 将 Write 指令传输给 DRAM 芯 2011 时,通过数据传输门 2078A 和 2078B 也将对应的数据传输给 DRAM 芯 2011。

图 51 到图 58 是表示第 1 实施例的多端口存储器操作的时间图。图 51 和图 52, 图 54 和图 55 与图 57 和图 58 是为了便于说明将单个时间图分成两部分的图, 一个表示时间图的第一个一半, 另一个表示时间图的第二个一半, 它们之间存在一些重叠。

图 51 和图 52 表示当将 Read 指令相继地输入两个端口时实施的操作。A 端口和 B 端口, 它们分别具有有相互不同频率的时钟信号 CLKA 和 CLKB, 与接收的时钟信号同步地取得指令, 地址和写数据, 并与时钟信号同步地输出检索的数据。在这个例子中, A 端口操作在最大时钟频率, 而 B 端口操作在稍低的时钟频率。对于 A 端口, Read 指令周期=4 (CLKA), 数据等待时间=4, 和脉冲串长度=4。对于 B 端口, Read 指令周期=2 (CLKB), 数据等待时间=2, 和脉冲串长度=2。在各端口的模寄存器 2031 和 2041 中分别设置数据等待时间和脉冲串长度。在这个例子中, 响应一个指令与时钟信号同步地实施数据的输入/输出 4 次, 在输入读指令后的 4 个时钟信号中输出检索的数据。

分别将加到 A 端口和 B 端口的指令存储在指令寄存器 2028 和 2029 中。当刷新定时器 2051 产生信号时, 刷新指令寄存器 2027 将刷新指令存储在它的里面。判优器 2026 监视这些指令寄存器, 并以发出指令的次序将这些指令传输给 DRAM 芯 2011。当完成上一个指令的处理后传输下一个指令。将从 DRAM 芯 2011 读出的数据从读出缓冲器 2016 传输到各端口的数据锁存器 2069A 和 2069B, 然后将数据从并行数据变换成串行数据, 接着作为脉冲串数据与各外部时钟信号同步地输出。

如图所示, 将指令 Read-A2 输入 Read 指令寄存器 AR 和将指令 Read-B2 输入 Read 指令寄存器 BR。在此之前, 发生一次刷新, 并将刷新指令输入刷新指令寄存器。根据发出指令的次序, 判优器 2026 以 Read-A2→Ref→Read-B2 的次序将这些指令传输给 DRAM 芯 2011,

然后由芯执行这些指令。甚至当在内部实施刷新操作时，从外部看来数据是在一个预定数据等待时间后输出的。这样，就不需要考虑任何刷新操作。

图 53 表示在与上述相同的条件下接连地输入 Write 指令的一个例子。也以脉冲串输入的形式给出在 Write 操作时间从器件外部输入的数据。在输入最后一块数据的定时将 Write 指令存储在 Write 指令寄存器 AW 中。在这个情形中，甚至当在内部产生和执行刷新指令时也不需要考虑任何刷新操作。

图 54 和图 55 表示当 A 端口和 B 端口两者都操作在最大时钟频率上进行 Read 操作时实施的操作。图 56 是表示当 A 端口和 B 端口两者操作在最大时钟频率上进行 Write 写操作时实施的操作的图。在这个情形中，在两个端口的时钟信号中可能存在相位差。对于两个端口，Read 指令周期=4，Write 指令周期=4，数据等待时间=4，和脉冲串长度=4。从图可见，在这个情形中也能够适当地实施操作。

图 57 和图 58 是表示当两个端口都操作在最高频率，并用内部产生的刷新指令从 Write 指令改变到 Read 指令时实施的操作的时间图。这是指令最拥挤的情形。

如图所示，DRAM 芯 2011 以 Ref→Write-A1→Write-B1→Read-A2→Read-B2 的次序进行操作，在它们之间没有任何间隙。在这个例子中，在输入 Write 指令后的 6 个时钟脉冲输入 Read-A2 和 Read-B2。即便这些定时超前 2 个时钟信号，也不可能超前 DRAM 芯的内部操作。由从输入 Read 指令开始的数据等待时间对读数据的输出定时进行控制。如果 Read-A2 和 Read-B2 的输入定时是超前的，则也需要将数据输出定时向前移动。在这个情形中响应 Read-B2 的数据输出定时太接近 DRAM 芯操作的开始时间，使得不能适当地执行 Read-B2。因为这个原因，需要将 Write →Read 变迁的公共间隔设置得相当长如本例中的 6 个时钟脉冲。

关于 Read → Write 的公共间隔，因为除非完成了 Read 数据的输出，不能将 Write 数据输入 DQ 端子中，所以公共间隔不可避免地变

长。

图 59A 和 59B 是表示 DRAM 芯 2011 操作的图。图 59A 表示 Read 操作，图 59B 表示 Write 操作。如这些图所示，响应单个指令以字线选择→数据放大→写回→预充电的次序实施一系列操作，从而完成整个操作。

如上所述，在第 1 实施例中从 Write 指令到 Read 指令的指令变迁的时间中指令间隔被加长。在第 2 实施例中对此进行了改进。当在第 1 实施例中一个相关的指令间隔是 6 个时钟脉冲周期时，第 2 实施例能将它缩短为 5 个时钟脉冲周期。

本发明的第 2 实施例的多端口存储器具有与第 1 实施例的多端口存储器类似的配置，不同之处只是刷新电路具有图 60 所示的配置。图 61 是表示第 2 刷新电路 2083 的配置的电路图。

如图 60 所示，第 2 实施例的刷新电路包括将图 43C 的刷新定时器 2051 和刷新指令发生器 2052 组合起来的定时器/刷新指令发生器 2081，第 2 刷新指令寄存器 2082 和第 2 判优器 2083，并将从第 2 判优器 2083 输出的刷新指令输入到刷新指令寄存器 2027。与第 1 实施例相同将刷新指令寄存器 2027 的刷新指令 REF2 输入到判优器 2026。在这个配置中，也将在完成刷新操作后从判优器 2026 输出到刷新指令寄存器 2027 的复位信号 ResetREF 加到第 2 刷新指令寄存器 2082。

在第 2 实施例的刷新电路中，沿着刷新指令的路径提供第 2 判优器 2083。如果我们预期如在 Write 指令→Read 指令的指令变迁的情形中一样指令很拥挤，则第 2 判优器 2083 延迟刷新指令到刷新指令寄存器 2027 的传输。第 2 判优器 2083 检查是否用图 61 所示的电路配置发生从 Write 指令到 Read 指令的改变，并且如果检查出这样一个改变就延迟刷新指令从第 2 刷新指令寄存器 2082 到刷新指令寄存器 2027 的传输。

如图 61 所示，当各端口接到从器件外部转装置提供的 Write 指令时，使禁止 REF 传输的信号 A 和 B 去激活，在一个时钟脉冲周期后再被激活，接着在接到最后一个数据项后的若干个时钟脉冲周期（即，

在本例中 3 个时钟脉冲)再被去激活。图 61 的 3 个 CLK 延迟装置 2084A 和 2084B 包括触发器等,并分别被 WA1 和 WB1 复位,这导致当通过延迟装置时 WA1D 和 WB1D 被复位。得到禁止 REF 传输的信号 A 和 B 的逻辑 AND 以便产生一个禁止 REF 传输的信号。因为只有当两个端口都经受从 Write 指令到 Read 指令的改变时在这个例子中才会出现问题,而当只有一个端口经受这种改变时不存在问题,所以得到这个逻辑 AND。进一步,禁止 REF 传输的信号 A 和 B 在接到 Write 指令后只对一个时钟周期去激活的原因是这给出一个额外的时间以便完成在接收最后一个数据项前的刷新操作。进一步,提供延迟装置 2086 为了相对于时钟信号稍微延迟定时,以便增大在禁止 REF 指令传输的信号和从器件外部提供的指令之间的相关定时中的差别。

图 62 到图 69 是表示第 2 判优器的操作的定时图。图 70 到图 72 是表示第 2 实施例的多端口存储器操作的定时图。图 62 和图 63,图 64 和图 65,图 66 和图 67,图 68 和图 69,图 70 和图 71 是为了便于说明起见将单个时间图分成两半的图,一个表示时间图的第 1 个一半,另一个表示时间图的第 2 个一半,它们之间存在一些重叠。

图 62 和图 63 表示两个端口经受 Write→Read 指令改变,并当 REF 传输禁止期间发生刷新定时器事件的情形。在这个情形中,在完成 Read-A2 和 Read-B2 后实施刷新操作 Ref。

图 64 和图 65 表示与上述情形相同两个端口经受 Write→Read 指令改变,但是在 REF 传输禁止期间前发生刷新定时器的情形。在这个情形中,在实施刷新操作 Ref 后实施 Read 操作。

图 66 和图 67 说明只有 A 端口经受 Write→Read 指令变迁,并在 REF 传输禁止期间中发生刷新定时器事件的情形。在这个情形中,在完成 Write 指令后实施刷新操作 Ref,然后实施 Read 操作。

图 68 和图 69 显示在两个端口继续 Write 的情形。在这个情形中,一当在接着最后一个数据输入,输入 Write 指令后,就使 3 个 CLK 延迟装置 2084A 和 2084B 去激活。

图 70 和图 71 是表示与图 57 和图 58 所示的第 1 实施例的操作对

应的第 2 实施例的操作的时间图。与第 1 实施例比较使 Write→Read 指令变迁的指令间隔从 6 个时钟脉冲缩短到 5 个时钟脉冲。

图 72 是表示与图 56 所示的第 1 实施例的操作对应的第 2 实施例的操作的时间图。虽然与第 1 实施例比较改变了关于刷新操作的指令执行次序，但是保持有次序的操作。

如上所述，第 2 实施例可以在任何条件下适当地实施操作，并能够将 Write 指令→Read 指令变迁的指令间隔缩短到 5 个时钟脉冲周期。

如上所述，本发明允许当以 DRAM 芯为基础制成存储阵列时使用多端口存储器而不需要考虑任何刷新操作，从而以低成本提供具有大容量和容易使用的多端口存储器。

[本发明的第 3 方面]

下面我们描述本发明的第 3 方面。

存在若干种多端口存储器。下文中，涉及具有多个端口的存储器，并允许从各端口相互独立地存取一个公共存储器阵列。例如，两个端口型的多端口存储器装备有 A 端口和 B 端口，并允许从与 A 端口链接的 CPU-A 和从与 B 端口链接的 CPU-B 独立地进行到公共存储器的读/写存取。

作为这类多端口存储器，具有 SRAM 存储器阵列的存储器是已知的，其中在复制组中提供各字线和各位线对，每个存储单元都与 2 组字线和位线对连接。然而，这个多端口存储器具有电路密度低的问题，其中需要提供字线和位线对的复制组。

为了消除这个问题，可以用与具有多个处理器配置的计算机所用的共用存储器相同的机构。一个共用存储器具有提供给公共存储器的多个端口。典型地，将 SRAM 用作存储器，并用离散的 IC 制成多个端口。当从多个端口同时进行存取时，因为存储器阵列是共用的，所以不能同时进行与多个端口对应的操作。防止这种问题发生的最简单的方法是对每个端口产生一个 BUSY 信号以便防止当从一个端口作出

存取时到另一个端口的存取。然而，这引起限制存储器用度的问题。考虑到这一点，为一个公共存储器提供一个称为判优器的判优电路，判优电路确定多个端口接收的存取要求的优先权。构造存储器阵列的控制器，以优先权的次序执行与存取要求相应的操作。例如，以到达的次序即以将存取要求加到各端口的次序处理存取要求。然而，这并不改变当另一个端口的指令正在被处理时不能执行新指令的情况。在这种情形中需要传输 BUSY 信号，存取存储器的器件需要具有处理 BUSY 信号的机构。

随机地从多个接口存取存储器阵列。因此，不提供在同一个行地址上连续地存取相继的列地址的列存取操作，但是在 DRAM 中典型地可以利用这种列存取操作。即，选择一个存储单元，为了读/写操作存取该存储单元，使该存储单元复位，所有这些操作都是在响应单个存取时被执行的。

当完成一个共用存储器时，一般，常规地将 SRAM 用作存储器阵列。这是因为 SRAM 能够进行高速随机存取操作，又，因为不需要刷新操作，所以很容易使用 SRAM。而且，单块芯片的多端口存储器通常具有字线和位线对的复制组，但是在具有通常的 SRAM 配置的存储器阵列的基础上的单块芯片的多端口存储器还没有在实践中使用。

总结一下，多端口存储器和共用存储器是用 SRAM 制成的，而不用需要刷新操作的 DRAM。

当系统不断地提供高性能时要被处理的数据量增加，并且多端口存储器也需要具有大的容量。可以用动态型存储单元 (DRAM) 阵列制成多端口存储器，DRAM 比 SRAM 具有较高的电路密度，从而以低的成本提供具有大存储容量的多端口存储器。然而，存储单元的刷新操作成为一个问题。

在常规的 DRAM 中，需要从器件外部以恒定间隔在读/写指令之间提供刷新指令。为此，在以 DRAM 为基础的系统中的控制器器件具有用于刷新管理的定时器和/或控制电路。然而，在用以 SRAM 为基础的多端口存储器的系统中不提供这样一个电路。甚至在以 DRAM 为

基础制成存储器的情形中，在这些系统中需要能以与常规的多端口存储器相同的方式使用这种存储器。即，具有由 DRAM 组成的存储器阵列的多端口存储器需要由它自己来进行刷新操作。

当判优器输出忙碌信号时，存在着如上所述使用存储器相当麻烦的问题。

本发明的目的是提供具有由 DRAM 芯组成的存储器阵列，而且不需要考虑刷新任何刷新操作就能使用的多端口存储器，从而以低成本提供具有大存储容量并且容易使用的多端口存储器。

为了消除上述的问题，配置本发明的多端口半导体存储器件使它能够以在长度为每个外部端口的最小输入周期 m ($m \geq 2$) 倍的时间周期中执行 n 个内部操作，其中满足 $mN < n < m(N+1)$ 。

上述条件要求将 N 个端口的每一个的最小指令周期设置在允许 N 个内部操作周期的时间周期加比单个内部操作周期短的时间周期 α 上。例如当 $N=2$ 时，将每个端口的最小外部指令周期设置在允许 2 个内部操作周期的时间周期加时间周期 α 上。这里，时间周期 α 比一个内部操作周期短。

本发明利用允许 2 个内部操作周期的时间周期以便消除由于判优器输出忙碌信号引起的存储器使用麻烦的问题，并利用时间周期 α 解决刷新操作问题。

图 73 是用于解释本发明（第 3 方面）原理的图，表示对于两个端口实施读操作的情形。

使到两个外部端口，A 端口和 B 端口的指令在最小间隔进入，在该间隔中可以实施内部操作周期 2.2 次。即，内部操作周期的 2.2 倍等于最小外部指令周期，并将外部指令周期设置得比允许实施内部操作周期 2.2 次的时间周期长。分别将时钟脉冲 CLKA 和 CLKB 输入 A 端口和 B 端口，与对应的时钟脉冲同步地实施指令，地址和数据到外部端口的输入和从外部端口的输出。虽然未加说明，但是与指令同时输入地址。当在最小外部指令周期中将读指令加到 A 端口和 B 端口时，如图所示，判优电路对当实施芯操作时将优先权给予首先到达的指令

进行控制。

DRAM 芯实施两个读操作，在一个外部指令周期中从存储器阵列读出数据，并将数据输出到 A 端口和 B 端口。A 端口和 B 端口分别保存检索的数据，并与各时钟脉冲信号的特定的时钟定时同步地输出检索的数据，这些时钟脉冲信号是从输入读指令开始的第 6 个时钟脉冲。即，在这个情形中数据等待时间是 6。

提供刷新定时器作为内部电路，并在它自身产生刷新指令。当不发生刷新操作时，器件的内部电路以例行程序的方式操作以便在一个外部指令周期中实施与指令 A 和 B 对应的两个操作。因为在一个外部指令周期中能够执行内部操作 2.2 次，所以 DRAM 芯将具有在完成两个内部操作后留下的一个额外时间 $t\alpha$ 。

当内部产生刷新指令时，器件的内部电路快速操作。这里，快速意味着执行操作而不产生额外时间 $t\alpha$ 。当产生刷新指令时，器件实施刷新操作。因为同时将指令输入 A 端口和 B 端口，所以要被处理的指令将积累起来。器件快速地一个接一个地执行指令而不提供额外时间 $t\alpha$ 。虽然将指令一个接一个地输入 A 端口和 B 端口，但是只有在比外部指令周期长的间隔发生刷新指令，只有指令 A 和指令 B 必须被执行直到产生下一个刷新指令。因为内部指令的处理速度较快，所以在结束时将不会有积累的指令。换句话说，内部处理将赶上外部指令的输入。此后，器件回到它的例行程序操作。通过考虑到外部端口的数目，内部操作周期的数目，刷新间隔等确定额外时间 α 。

因为当内部刷新指令和输入到另一个端口的指令立即在 Read 指令前发生时定时变得最坏，所以需要将与 Read 指令 (RD) 对应的数据输出的延迟时间 (数据等待时间) 设置在内部操作的 3 个周期中 (在两个端口的情形中)。然而，因为稍长于两个内部操作周期的外部指令周期就是器件适当操作所需要的一切，所以数据传输速率是相当高的。

如上所述，本发明能够取消来自器件外部的刷新操作，并将外部指令周期设置得稍长于两个内部操作周期。不需要刷新来自外部装置

的控制，甚至当在内部执行刷新操作时，这对外部装置是完全看不见的，并且不影响从外部装置看的器件操作的方式。因此，可以从每个外部端口进行到存储器的存取而不用考虑其它端口。

在这个方式中，本发明能够提供用 DRAM 存储单元的多端口存储器，它有大容量和快的数据传输速率，同时允许使用存储器而不用考虑任何刷新操作就像它在 SRAM 的基础上完成一样。

在图 73 的例子中，响应一个读指令与外部时钟脉冲同步地输出一项读数据。即，脉冲串长度为 1。所以，在一个时钟脉冲周期中完成读数据的输出后，在外部指令周期的 3 个留下的时钟脉冲周期中外部端口不输出任何数据，这导致无效的数据传输。这个问题能够通过加长脉冲串长度来消除。

图 74 是用于解释本发明原理的图，表示脉冲串长度为 4 的例子。在这个例子中，如以前的情形那样，将两个外部端口的外部指令周期设置在能够提供 2.2 个内部操作周期的长度上。进一步，一个外部指令周期对应于 4 个时钟脉冲周期。与时钟脉冲同步地以提供数据等待时间为 6 的方式在单个外部指令周期中从一个外部端口输出数据 4 次。所以，如果根据一个外部指令周期的时钟脉冲周期的数目设置脉冲串长度，则在两个端口中都能够达到无间隙的读操作，从而非常大地促进了数据传输速率的提高。在这个情形中，需要响应单个存取将与脉冲串长度一样多的数据项内部地输入存储器阵列或从存储器阵列输出。例如，如果一个外部端口的数据输入/输出数数目为 4 和脉冲串长度为 4，则需要保证通过单个存取操作将 16 位数据从存储器阵列输出或输入存储器阵列。

我们应该注意到 A 端口和 B 端口不一定同步地操作，只要设置最小周期等于对于 N 个内部操作周期加上比单个内部操作周期短的持续时间 α 需要的持续时间，就能够相互独立地将各外部指令周期设置在任何定时。

图 75 和图 76 是表示在 2, 3 和 N 个端口的情形中在一个最小外部指令周期和各内部操作周期之间的关系的图。如图所示，如果端口

数为 2，则最小外部指令周期是允许 2 个内部操作的时间长度加上 α ，如果端口数为 3，则最小外部指令周期是允许 3 个内部操作的时间长度加上 α 。进一步，如果端口数为 N，则最小外部指令周期等于其间能执行 N+1 个内部操作的时间长度加上时间长度 α 。

图 77 和图 78A 到图 78C 是表示根据本发明的一个实施例的多端口存储器的配置的图。图 77 表示 DRAM 芯及其相关电路，图 78A 表示 A 端口，图 78B 表示 B 端口。进一步，图 78C 表示刷新电路。图 78A 到图 78C 所示的电路与图 77 的各部分连接。

如这些图所示，本实施例的多端口存储器包括 DRAM 芯 3011，用于控制确定操作次序并保证以确定的次序实施操作的判优器 3026，指令寄存器 3025，它暂时存储从判优器 3026 提供的指令，并将这些指令以接到它们的次序传输给 DRAM 芯 3011 的控制电路 3014，暂时存储各端口的指令，地址和数据的多组寄存器，2 个由 A 端口 3030 和 B 端口 3040 组成的外部端口，和刷新电路 3050。

A 端口 3030 和 B 端口 3040 分别包括模寄存器 3031 和 3041，CLK 缓冲器 3032 和 3042，数据 I/O 电路 3033 和 3043，地址输入电路 3034 和 3044，与指令输入装置 3035 和 3045，它们根据从器件外部提供的各分开的时钟频率进行操作。将数据等待时间和脉冲串长度存储在模寄存器 3031 和 3041 中，使它们能被分别地设置。数据 I/O 电路 3033 和 3043 装备有根据脉冲串长度实施输出/输出数据的并行到串行变换和串行到并行变换的机构。

刷新电路 3050 包括刷新定时器 3051 和刷新指令发生器 3052。刷新定时器 3051 在预定间隔上产生刷新开始信号，和刷新指令发生器 3052 对应地产生刷新指令。

分别将加到 A 端口和 B 端口的指令存储在指令寄存器 A 28A 和指令寄存器 B 28B 中。分别将地址存储在地址寄存器 A 19A 和地址寄存器 B 19B 中，分别将写入的数据存储在 Write 数据寄存器 A 22A 和 Write 数据寄存器 B 22B 中。进一步，将刷新指令存储在刷新指令寄存器 3027 中，并将刷新地址存储在刷新地址计数器/寄存器 3018 中。

判优器 3026 根据指令到达的次序确定执行指令的次序,并以确定的次序将指令传输给指令寄存器 3025。指令寄存器 3025 以从判优器 3026 接到指令的次序将这些指令传送给 DRAM 芯 3011 的控制电路 3014。当 DRAM 芯处理给定指令时,使控制电路 3014 处于能够接收下一个指令的状态。对应地,指令寄存器 3025 下一个指令传送给控制电路 3014。同时将从判优器 3026 提供的指令暂时存储在指令寄存器 3025 中。进一步,指令寄存器 3025 除了将指令传输给 DRAM 芯 3011 的控制电路 3014 外也将传输信号传输给对应的地址寄存器和对应的数据寄存器(在写操作情形)。在 DRAM 芯 3011 中,控制电路 3014 响应提供的指令,控制译码器 3013,写放大器(WriteAmp) 3015,和读出缓冲器 3016,从而对于存储器阵列 3012 实施存取操作。在写操作的情形中,译码器 3013 为了进行写操作对存取的地址进行译码,以便激活在存储器阵列 3012 中的字线和列信号线,导致将存储在 Write 数据寄存器 A 和 B 中的写数据通过 WriteAmp 3015 写入存储器阵列 3012。在读操作的情形中,以类似的方式存取存储器阵列 3012,导致分别通过标记为 3024A 和 3024B 的传输门 A 和 B,将读数据从读出缓冲器 3016 传输到各端口的数据输出电路。根据 DRAM 芯 3011 的操作周期,控制传输门的传输定时,并由控制电路 3014 确定传输定时。与对应的外部时钟同步地从每个端口的数据输出电路输出输出数据。

下面,我们描述与每个指令处理,地址处理和数据处理有关的详细情形。

图 79 和图 80 是表示根据第 1 实施例与指令处理有关的装置的配置的图。在与图 77 和图 78A-78C 相同的部件上加上相同的参照数字。这也同样应用于其它的图。

如图 79 所示,A 端口的指令输入装置 3035 包括输入缓冲器 3036,指令译码器 3037 和 $(n-1)$ 时钟延迟电路 3038, B 端口的指令输入装置 3045 包括输入缓冲器 3046,指令译码器 3047 和 $(m-1)$ 时钟延迟电路 3048。这里, n 和 m 是脉冲串长度。而且,如图 80 所示,指令

寄存器 A 包括 Read 指令寄存器 AR 和 Write 指令寄存器 AW, 指令寄存器 B 包括 Read 指令寄存器 BR 和 Write 指令寄存器 BW。

输入缓冲器 3036 和 3046 要求以与各时钟脉冲 CLKA1 和 CLKB1 同步地加上 Read 指令, 指令译码器 3037 和 3047 进行译码处理。指令译码器 3037 和 3047 在 Read 指令情形分别产生 RA1 和 RB1, 而在 Write 指令情形分别产生 WA1 和 WB1。分别将信号 RA1 和 RB1 传输给 Read 指令寄存器 AR 和 BR, 不需要任何定时操作, 而信号 WA1 和 WB1 被 (n-1) 时钟延迟电路 3038 和 (m-1) 时钟延迟电路 3048 延迟, 直到输入脉冲串数据的最后一个数据项为止, 接着分别被传输给 Write 指令寄存器 AW 和 BW。而且, 将由刷新电路 3050 产生的刷新指令 REF1 传输给刷新指令寄存器 3027。

判优器 3026 检测将指令传输给这 5 个指令寄存器 AR, AW, BR, BW 和 3027 的次序, 并以检测到的次序一个接着一个地将这些指令传输给指令寄存器 3025。在接到从判优器 2026 发出的指令时, 指令寄存器 3025 将指令接收确认传输给判优器 3026。响应指令接收确认, 判优器 3026 将下一个指令发送给指令寄存器。

指令寄存器 3025 以从判优器 3026 接到指令的次序将指令一个接着一个地传输给 DRAM 芯 3011 的控制电路 3014。DRAM 芯的控制电路 3014 执行接到的指令, 并当指令执行结束或接近结束时将准备好接收指令的信号传输给指令寄存器 3025。响应准备好接收指令的信号, 指令寄存器 3025 将下一个指令传输给控制电路 3014。同时, 将从判优器 3026 提供的指令暂时存储在指令寄存器 3025 中。

图 81 是判优器 3026 的实施例。指令到达图 80 的 5 个指令寄存器 (Read 指令寄存器 AR, Write 指令寄存器 AW, Read 指令寄存器 BR, Write 指令寄存器 BW 和刷新指令寄存器 3027) 的次序由比较器 3053 检测出来如该图所示。每个比较器 3053 比较两个指令寄存器的定时, 并首先输入“H”时将它的输出改变成“H”。AND 门 3054 通过检测相关比较器 3053 的所有相关输出是否是“H”来确定一个给定的指令是否在所有 4 个其它的指令前输入。如果一个对应的指令是最早的并

且被传输给指令寄存器 3025, 则响应各指令, 信号 RA31, WA31, RB31, WB31 和 REF31 变成“H”。如果 RA2 是 RA2 到 REF2 中最早的, 则与 RA2 连接的比较器具有在与 RA2 连接的一边上为“H”的输出。在这个特定的瞬间, 指令接收确认还没有产生(=“L”), 使得 N1=“H”, 导致 RA3 为“H”。于是将指令发送给指令寄存器 3025。

指令寄存器 3025 当接收指令时产生指令接收确认。当发生这种情况时, 在节点 N1 产生“L”脉冲, 导致 RA3 到 REF3 都为“L”。同时, 将产生 ResetRA 到 ResetREF 中的一个。如果 RA31 为“H”, 则产生 ResetRA, 从而使 Read 指令寄存器 AR 复位。对应地, RA2 变成“L”, 于是 RA31 到 REF31 中的一个变成“H”, 指示中下一个在线的指令。当在“L”脉冲末端 N1 变成“H”时, 将下一个在线的指令传输给指令寄存器 3025。此后重复上述的操作。

图 82 和图 83 是表示指令寄存器 3025 的配置的图。将指令寄存器 3025 分成两半并表示在两个图中。

指令寄存器 3025 主要包括移位寄存器 3092, 移位寄存器 3092 在其中存储指令, 将这些指令连续地输出到 DRAM 芯 3011, 并包括将从判优器 3026 接收的指令传输到移位寄存器 3092 的开关(SW1-SW3) 3082-3084。在这个例子中, 移位寄存器 3092 具有三级配置, 并包括用于存储指令的寄存器 3085-3087, 指示寄存器 3085-3087 的存储状态的标志 3088-3090, 和使寄存器 3085-3087 的状态复位的复位数据装置 3091。在没有指令存储在寄存器 3085-3087 的状态中, 标志 3088-3090 都处于低的状态 (FL1-FL3=“L”), 使得开关 3082 (SW1) 接上。通过 SW1 将第 1 指令存储在寄存器 3085 中, 使得 FL1 变成“H”。当 FL1 变成“H”时, “H”边沿脉冲电路 3093 产生脉冲, 将指令接收确认传输给判优器 3026。

如果在这个特定的瞬间 DRAM 芯 3011 认定准备好接收指令的信号, 则门 3097 打开将寄存器 3085 的指令传输给锁存器 3098, 然后将指令发送给 DRAM 芯 3011 的控制电路 3014。同时, 将对应于该指令的地址等传输给 DRAM 芯 3011。DRAM 芯 3011 当根据接到的指令开

始操作时取消准备好接收指令的信号。于是门 3097 关闭。寄存器控制电路 3096 产生促使寄存器 3086 的数据移到寄存器 3085 和寄存器 3087 的数据移到寄存器 3086 的移位信号。如果在移位信号产生前寄存器 3086 中没有存储指令，则移位操作导致使寄存器 3085 复位和使 FL1 变成“L”。寄存器控制电路 3096 在产生移位信号同时产生禁止传输信号以便断开 SW1-SW3，从而当移位操作时禁止将数据传送到移位寄存器 3092。当通过 SW1 将第 1 指令（指令 1）加到寄存器 3085 时，如果 DRAM 芯 3011 正在执行前一个指令则将指令存储在寄存器 3085 中。FL1 变成“H”，它断开 SW1，并进一步在一个预定延迟后断开 SW2。这里，预定延迟与从产生指令接收确认到使判优器输出复位的时间周期对应。如果在 DRAM 芯 3011 准备好接收指令前加上来自判优器 3026 的下一个指令（指令 2），则通过 SW2 将指令存储在寄存器 3086 中。FL2 变成“H”，它产生指令接收确认并断开 SW2，接着在一个预定延迟时间后进一步断开 SW3。当 DRAM 芯处在能接收指令的状态时，产生准备接收指令信号，打开门 3097，使得将寄存器 3085 的指令 1 传输给锁存器 3098，然后传输给 DRAM 芯 3011。DRAM 芯 3011 当根据指令 1 开始操作时取消准备接收指令信号。相应地，门 3097 关闭。寄存器控制电路 3096 产生移位信号，该移位信号将寄存器 3086 的指令 2 移到寄存器 3085，并将寄存器 3087 的内容（复位状态）移到寄存器 3086。寄存器 3085 结束指令 2 的存储，寄存器 3086 和 3087 终止在复位状态。因为 FL1 为“H”，FL2 和 FL3 为“L”，所以接上 SW2 而断开 SW1 和 SW3。

复位数据装置 3091 与在它左边的移位寄存器 3092 的寄存器连接。提供这个配置的目的是当在直到寄存器 3087 的整条路径上指令都被存储时，用接着的移位信号将寄存器 3087 的指令移到寄存器 3086。在这个方式中，指令寄存器 3025 暂时积累从判优器 3026 发送的指令，检测 DRAM 芯 3011 的状态，接着一个又一个地传输指令。

将指令产生检测信号输入寄存器控制电路 3096。当从判优器 3026 传输指令时产生指令产生检测信号。图 84A 和图 84B 表示寄存器控制

电路 3096 的操作。当到寄存器控制电路 3096 的准备接收指令信号去激活时产生移位信号和禁止传输信号。然而，当就在准备接收指令信号去激活前立即从判优器 3026 传输指令时，优先的是只在将较早接收的指令传输给移位寄存器 3092 后才实施移位操作。因此，进行比较以便确定准备接收指令信号的后沿和指令产生检测信号的前沿中哪一个较早。如果前者较早，则响应前者的后沿产生移位信号和禁止传输信号，如果后者较早，则响应后者的后沿产生移位信号和禁止传输信号。

图 85 和图 86 是表示指令寄存器 3025 操作的图。这里对关于在表示输入指令最拥挤的定时条件的 Write→Read 指令传输的时间产生刷新指令的情形进行说明。图中所示的 SW1 到 SW3 的数目指出连接的 SW，并说明了连接 SW 的持续时间。进一步，电阻 1 到 3 分别对应于寄存器 3085 到 3087。

图 87 是表示根据第 1 实施例与地址处理有关的部分的配置的图。下文中，在它的信号名称的末端具有字母“P”的信号代表具有从对应的信号名称的信号的前沿产生的脉冲的信号。如图所示，地址输入电路 3034 和 3044 分别包括输入缓冲器 3057A 和 3057B，传输门 3058A 和 3058B。进一步，地址寄存器 3019A 和地址寄存器 3019B 分别包括地址锁存器 A1 到 A4 和 B1 到 B4，传输门 3059A 到 3063A 和 3059B 到 3063B。通过地址总线 3017 将从传输门 3062A，3062B，3063A，3063B 提供的地址传输到 DRAM 芯 3011。进一步，通过传输门 3064 和地址总线 3017 将从刷新地址计数器/寄存器 3018

提供的刷新地址传输到 DRAM 芯 3011。

当从器件外部输入 Read 指令或 Write 指令时，分别通过传输门 3058A 或 3058B 将加到输入缓冲器 3057A 或 3057B 的地址和输入指令同时传输给地址锁存器 A1 或 B1。在 Read 指令的情形中，通过传输门 3061A 和 3063A 或 3061B 和 3063B 和地址锁存器 A4 或 B4 与到 DRAM 芯的指令传输同步地将地址传送给 DRAM 芯 3011。在 Write 指令的情形中，进一步在上次数据采集的定时将地址传输给地址锁存器 A2 或 B2，然后，与到 DRAM 芯的指令传输同步地通过传输门 3062A

或 3062B 传输给 DRAM 芯 3011。进一步,刷新地址计数器/寄存器 3018 产生并在其中保存刷新地址,然后与刷新指令到 DRAM 芯的传输同步地通过传输门 3064 将该地址传输给 DRAM 芯 3011。

图 88 是表示根据第 1 实施例与数据输出有关的部分的配置的图。图 89 是表示图 88 的传输信号发生电路的图。A 端口 3030 和 B 端口 3040 的各数据 I/O 电路 3033 和 3043 分别包括为了输出数据的电路 3065A 和 3065B 及为了输入数据的电路 3074A 和 3074B。如图所示,将通过读出缓冲器 3016 从存储器阵列 3012 读出的数据通过数据总线 3021 和传输门 3024A 或 3024B 分别传输给为了输出数据的电路 3065A 或 3065B。

为了输出数据的电路 3065A 和 3065B 分别包括数据锁存器 A1 或 B1,传输信号发生电路 3067A 和 3067B,传输门 3068A 或 3068B,数据锁存器 A2 或 B2,并行到串行变换器 3070A 和 3070B,以及输出缓冲器 3071A 和 3071B。

传输门 3024A 和 3024B 由 DRAM 芯 3011 的控制电路 3014 根据内部操作进行控制。如果执行的指令是 Read-A (即,对于 A 端口的读操作),则传输门 3024A 将打开。如果执行的指令是 Read-B,则传输门 3024B 将打开。数据锁存器 A1 或 B1 在其中存储数据,然后在通过传输门 3068A 和 3068B 引入等待时间的各端口中接收 Read 指令后的一个预定等待时间,将数据传输给各数据锁存器 A2 或 B2。然后,由并行到串行变换器 3070A 和 3070B 对数据进行变换,接着分别传输给输出缓冲器 3071A 和 3071B,并在那里输出。

如图 89 所示,传输信号发生电路 3067 (即 3067A 或 2067B) 采用一系列的触发器 3072 使各 Read 指令 RA1 或 RB1 延迟由等待时间设定确定的许多时钟脉冲周期,从而产生数据传输信号 3002。因为响应数据传输信号 3002,通过传输门 3068A 和 3068B 传输读数据,所以读数据从读操作的定时开始被延迟与等待时间设置相同的许多时钟脉冲周期后结束。

图 90 和图 91 是表示根据本实施例与数据输入有关的部分的配置

的图。为了输入数据的电路 3074A 和 3074B 分别包括数据输入 (Din) 缓冲器 3075A 和 3075B, 串行到并行变换器 3076A 和 3076B, 以及数据传输装置 3077A 和 3077B。分别通过第 1 个 Write 数据寄存器 3078A 和 3078B, 数据传输门 3079A 和 3079B, 第 2 个 Write 数据寄存器 3080A 和 3080B, 数据传输门 3081A 和 3081B, 以及数据总线 3021 将来自数据传输装置 3077A 和 3077B 的 Write 数据 WDA 和 WDB 传送给 WriteAmp 3015, 然后被写入存储器阵列 3012。

根据脉冲串长度将串行输入数据从串行变换到并行, 然后在输入最后一个数据项的定时传输给第 1 个 Write 寄存器 3078A 和 3078B。当从指令寄存器 3025 将 Write 指令传输给 DRAM 芯 3011 时, 也将对应数据传输给 DRAM 芯 3011。

图 92 到图 99 是表示第 1 实施例的多端口存储器的操作的定时图。图 92 和图 93, 图 95 和图 96, 图 98 和图 99 是为了便于说明起见将单个时间图分成两半的图, 一个表示时间图的第 1 个一半, 另一个表示时间图的第 2 个一半, 它们之间存在一些重叠。

图 92 和图 93 表示当将 Read 指令相继输入两个端口时实施的操作。A 端口和 B 端口, 它们分别具有有相互不同频率的时钟信号 CLKA 和 CLKB, 与接收的时钟信号同步地取得指令, 地址和写数据, 并与时钟信号同步地输出检索的数据。在这个例子中, A 端口操作在最大时钟频率, 而 B 端口操作在稍低的时钟频率。对于 A 端口, Read 指令周期=4 (CLKA), 数据等待时间=6 (CLKA), 和脉冲串长度=4。对于 B 端口, Read 指令周期=2 (CLKB), 数据等待时间=3 (CLKB), 和脉冲串长度=2。在各端口的模寄存器 3031 和 3041 中分别设置数据等待时间和脉冲串长度。对于 A 端口, 响应一个指令与时钟信号同步地实施数据的输入/输出 4 次, 在输入读指令后的 6 个时钟脉冲周期输出检索的数据。对于 B 端口, 响应一个指令与时钟信号同步地实施数据的输入/输出 2 次, 在输入读指令后的 3 个时钟脉冲周期输出检索的数据。

分别将加到 A 端口和 B 端口的指令存储在指令寄存器 3028A 和

3028B 中。当刷新定时器 3051 产生信号时，刷新指令寄存器 3027 在其中存储刷新指令。判优器 3026 监视这些指令寄存器，并以发出指令的次序将这些指令传输给指令寄存器 3025。指令寄存器 3025 暂时存储接收的指令，并以接到它们的次序将它们连续地传输给 DRAM 芯 3011。即，在完成上一个传输指令的处理后传输下一个指令。

如图所示，将指令 Read-A2 输入 Read 指令寄存器 AR 和将指令 Read-B2 输入 Read 指令寄存器 BR。在此之前，发生一次刷新，并将刷新指令输入刷新指令寄存器。根据发出指令的次序，判优器 3026 以 Read-A2→Ref→Read-B2 的次序将这些指令传输给 DRAM 芯 3011，然后由芯执行这些指令。

由于芯的操作在 Read-B1 和 Read-A2 之间存在额外时间，并直到这个点都实施正常的和例行程序操作。当发生刷新时，在 Read-A2 后立即实施刷新而在其间没有任何时间间隙。此后，相继实施 Read-B2，Read-A3 等而没有任何时间间隙直到执行 Read-A5。与正常的和例行程序操作相反，直到这个点都执行快速操作。

由于刷新指令的执行，内部操作相对于来自器件外部的指令输入显示出一些延迟。快速操作补偿该延迟，在执行指令 Read-A5 前赶上。在 Read-A5 和 Read-B5 之间再次存在额外时间，指出回到正常的和例行程序操作。将通过读出缓冲器 3016 从 DRAM 芯 3011 读出的数据通过传输门传输给接收对应的 Read 指令的端口的数据锁存器（数据锁存器 A1 或 B1）。数据锁存器 A1 或 B1 为数据提供时间调整，然后将数据传输给数据锁存器 A2 或 B2，并与对应端口的时钟信号同步地输出数据。

甚至当在内部实施刷新操作时，从外部看来数据是在一个预定数据等待时间后输出的。这样，就不需要考虑任何刷新操作。

图 94 表示当在与上述相同的条件下接连地输入 Write 指令时的例子。也以脉冲串输入的形式给出在 Write 操作时从器件外部输入的数据。在输入最后一个数据块的定时将 Write 指令存储在 Write 指令寄存器 AW 中。在这个情形中，甚至当在内部产生和执行刷新操作时也

不需要考虑任何刷新操作。

图 95 和图 96 表示当 A 端口和 B 端口两者都操作在最大时钟频率上进行 Read 操作时实施的操作。图 97 是表示当 A 端口和 B 端口两者都操作在最大时钟频率上进行 Write 操作时实施的操作的图。在这个情形中，在这两个端口的时钟脉冲信号中可能存在相位差。对于两个端口，Read 指令周期=4，Write 指令周期=4，数据等待时间=6，和脉冲串长度=4。如从图可见的那样，在这个情形中也可以适当地实施操作。

图 98 和图 99 是表示当两个端口都操作在最高频率，并用内部产生的刷新指令经受从 Write 写指令到 Read 读指令的改变时实施的操作的时间图。这是指令最拥挤的情形。

如所说明的那样，DRAM 芯 3011 以 Ref→Write-A1→Write-B1→Read-A2→Read-B2 的次序进行操作，在它们之间没有任何间隙。在这个例子中，在输入 Write 指令后 6 个时钟脉冲输入 Read-A2 和 Read-B2。即便这些定时超前 2 个时钟脉冲，也不可能超前 DRAM 芯的内部操作。由从输入 Read 指令的数据等待时间对读数据的输出定时进行控制。如果 Read-A2 和 Read-B2 的输入定时是超前的，则也需要使数据输出定时是超前的。例如，如果在 Write-B1 后 4 个时钟脉冲输入 Read-B2，则响应 Read-B2 的数据输出定时太接近 DRAM 芯操作的开始时间，使得不能适当地执行 Read-B2。因为这个原因，需要将 Write →Read 变迁的指令间隔设置得相当长如本例中的 6 个时钟脉冲。

关于 Read → Write 的指令间隔，因为除非完成了 Read 数据的输出，不能将 Write 数据输入 DQ 端子中，所以指令间隔不可避免地变得很长。

图 100A 和 100B 是表示 DRAM 芯 3011 操作的定时图。图 100A 表示 Read 操作，图 100B 表示 Write 操作。如这些图所示，响应单个指令以字线选择→数据放大→写回→预充电的次序实施一系列操作，从而完成整个操作。当接到指令时 DRAM 芯 3011 使准备接收指令信

号去激活,并当完成或接近结束指令的执行时产生准备接收指令信号。

如上所述,本发明允许当以 DRAM 芯为基础制成存储器阵列时使用多端口存储器而不需要考虑任何刷新操作,从而以低成本提供有大容量和容易使用的多端口存储器。

[本发明的第 4 方面]

下面我们描述本发明的第 4 方面。

多端口存储器,它们是装备有多个端口的半导体存储器,可以分成不同的类型。当下文中使用术语“多端口存储器”时,它指的是具有多端口的存储器,该存储器允许从任何一个端口独立地存取公共存储器阵列。这样一个存储器可以具有 A 端口和 B 端口,并允许对于公共存储器阵列从与 A 端口链接的 CPU 和从与 B 端口链接的 CPU 独立地进行读/写操作。

一个多端口存储器装备有称为判优器的判优电路。判优器确定从多个端口接收的存取要求的优先权,存储器阵列的控制电路根据确定的优先权一个接一个的进行存取操作。例如,存取要求越早到达端口,就会给予该存取要求越高的优先权。

在这种情形中,因为随机地从多个接口存取存储器阵列,所以在执行了读或写的存取操作后需要立即使存储器阵列复位,从而保证存储器阵列为下一次存取作好了准备。即,如果响应来自给定端口的存取要求使一条字线保持在选择状态,和如一般在 DRAM 中使用的列存取操作中那样连续地动移各列地址以便读出连续的数据,则当该操作期间来自另一个端口的存取要求将一直等待着。因此,在每次读或写操作后需要立即使存储器阵列复位。

常规地,已经典型地将 SRAM 用作多端口存储器的存储器阵列。这是因为 SRAM 允许高速随机存取,而且可以进行非破坏性读操作。

在具有两个端口的多端口存储器中,例如,一个 SRAM 存储单元具有两组字线和位线对。一个端口用一组字线和位线对实施读/写操作,另一个端口用另一组字线和位线对实施读/写操作。在这种方式中,

能够从两个不同的端口独立地实施读/写操作。然而，因为当两个端口企图在同一时间将数据写入同一存储单元时不可能同时进行两个写操作，所以给予一个端口进行写操作的优先权，而给予另一个端口 **BUSY**（忙碌）信号。这称为 **BUSY** 状态。

当开发系统使它有改善的性能时，由该系统处理的数据量也增加了。结果，多端口存储器需要很大的容量。然而，**SRAM** 型多端口存储器具有存储单元的尺寸大的缺点。

为了消除这个缺点，在多端口存储器中采用 **DRAM** 阵列是可以理解的。为了得到比多端口 **SRAM** 高得非常多的电路密度，需要用于多端口存储器中一个 **DRAM** 存储单元以与典型的 **DRAM** 单元相同的方式只与一条字线和一条位线连接。如果以这样一种方式用 **DRAM** 元件制成存储块，如果一个端口正在对给定存储块执行读或写操作，则另一个端口不能存取该存储块。这是因为在 **DRAM** 存储单元中只可以有破坏性读操作。即，当读取信息时，不能选择在同一个存储块中的另一条字线直到这个信息被放大和存储在存储单元中，字线和位线被预先充电为止。

只有当多个端口同时对同一个存储单元提出写要求时，在常规的 **SRAM** 型多端口存储器中才会出现 **BUSY** 状态。因此，**DRAM** 型多端口存储器需要具有一个独特的与常规的 **SRAM** 型多端口存储器不同的 **BUSY** 状态控制功能。

进一步，与 **SRAM** 型多端口存储器不同，**DRAM** 型多端口存储器需要周期地实施刷新操作以便保持存储的信息，从而必须采取某种措施保证适当的刷新定时。

因此，本发明的目的是提供能够消除特别与 **DRAM** 有关的问题的 **DRAM** 型多端口存储器。

根据本发明，半导体存储器件包括多个 N 个外部端口，它们中的每一个都接收指令，多个与各外部端口对应的 N 条总线，多个与 N 条总线连接的存储块，比较由输入到 N 个各外部端口的指令存取的地址的地址比较器，和判优电路，它确定当地址比较电路根据地址比较检

测出到同一个存储块的存取时，存取同一个存储块的各指令中的哪一个或哪几个要被执行，和存取同一个存储块各指令中的哪一个或哪几个不被执行。

在上面描述的本发明中，如果从器件外部输入端口的指令企图存取同一个存储块时，判优电路确定各指令中的哪一个要被执行，和各指令中的哪一个不被执行。例如，比较指令定时，执行较早的指令，而不执行其它的一个或多个指令。当存在不被执行的指令时，产生 **BUSY** 信号等并输出到器件外部。这使得甚至当在以 **DRAM** 芯为基础的多端口存储器中指令存取相互发生冲突时也可以实施适当的存取操作并实现适当的 **BUSY** 控制。

根据本发明的一个方面，存储块包括在动态型存储单元的基础上制成的存储单元阵列，而半导体存储器件包括定义刷新存储单元的定时的刷新电路。在第一模式中，响应输入到 N 个外部端口中的至少一个端口的刷新指令刷新存储单元，在第二模式中，在刷新电路指出的定时刷新存储单元。

上面描述的本发明具有一个操作模式，其中响应来自器件外部的指令执行刷新操作的操作模式，和响应来自内部刷新电路的指令执行刷新操作的操作模式。这使在这样一种方式，即将预定外部端口指定作为一个用于在恒定的间隔接收刷新指令进行刷新管理的端口中用多端口存储器成为可能，或者使在这样一种方式，即当所有的外部环部端口都处于去激活状态时内部刷新电路开始刷新操作中用多端口存储器成为可能。因此，本发明为遵从系统要求的灵活的刷新管理提供了基础。

下面我们将参照所附各图描述本发明（第 4 方面）的实施例。

图 101 是表示根据本发明的多端口存储器的实施例的方框图。在这个例子中，配置是这样的，即提供两个端口，A 端口和 B 端口。

图 101 的多端口存储器 4010 包括 A 端口 4011，B 端口 4012，自刷新电路 4013，存储块 4014-1 到 4014-n，判优器 4015，刷新地址计数器 4016，地址改变电路 4017，地址改变电路 4018，地址比较器 4019，

总线 A 4020-1 和总线 B 4020-2。

A 端口 4011 包括模寄存器 4031, CLK 缓冲器 4032, 数据 I/O 电路 4033, 指令译码寄存器 4034, 地址缓冲器/寄存器 4035 和 BUSY 信号 I/O 装置 4036。进一步, B 端口 4012 包括模寄存器 4041, CLK 缓冲器 4042, 数据 I/O 电路 4043, 指令译码寄存器 4044, 地址缓冲器/寄存器 4045 和 BUSY 信号 I/O 装置 4046。在 A 端口 4011 和 B 端口 4012, 与各时钟信号 CLKA 和 CLKB 同步地独立地建立到外部总线的存取和来自外部总线的存取。模寄存器 4031 和 4041 能够在其中存储对于各端口的模式设定如数据等待时间和脉冲串长度。在这个实施例中, A 端口 4011 和 B 端口 4012 两者都具有各自的模寄存器, 使每个端口都能进行模式设定。然而, 可以将模寄存器只安排在一个端口中, 例如, 使得对于 2 个端口的设置可以由对这一个端口的设置来实现。

自刷新电路 4013 包括刷新定时器 4046 和刷新指令发生器 4047。自刷新电路 4013 在器件中产生刷新指令, 分别从 A 端口 4011 和 B 端口 4012 接收信号 CKEA1 和 CKEB1。信号 CKEA1 和 CKEB1 是分别用 CLK 缓冲器 4032 和 4042 对外部信号 CKEA 和 CKEB 进行缓冲得到的。用外部信号 CKEA 和 CKEB 暂停各端口的时钟缓冲器并使各端口去激活。如果使 A 端口 4011 和 B 端口 4012 两者都进入去激活状态, 则自刷新电路 4013 开始它的操作。

存储块 4014-1 到 4014-n 每个都与内部总线 A 4020-1 和内部总线 B 4020-2 连接。存在多个外部端口 (即, A 端口和 B 端口), 其中 A 端口 4011 通过总线 A 4020-1 与存储块 4014-1 到 4014-n 中每一个接口, 和 B 端口 4012 通过总线 A 4020-2 与存储块 4014-1 到 4014-n 中每一个接口。

如果在同一个时间输入来自 A 端口 4011 的存取要求和来自 B 端口 4012 的存取要求, 则假定这些存取要求是指向不同的存储块的, 存取存储块就对应于这些存取要求独立地实施它们的操作。

如果来自 A 端口 4011 的存取要求和来自 B 端口 4012 的存取要求

是指向同一个存储块的，则判优器（判优电路）4015 确定指令到达的次序，并执行第 1 个到达的指令而删除第 2 个到达的指令。当删除指令时，判优器 4015 产生 BUSY 信号以便通知外部控制器已经被删除第 2 个到达的指令的存取要求。

地址比较器 4019 确定进入两个端口的存取要求中是否指向同一个存储块。详细的说，地址比较器 4019 比较包含在进入两个端口的地址中的块选择地址。如果它们是相同的，则将匹配信号加到判优器 4015。

当 A 端口 4011 或 B 端口 4012 处于激活状态时，从 A 端口 4011 和 B 端口 4012 输入刷新指令。

如果进入两个端口中的一个端口的刷新指令存取同一个存储块如输入到两个端口中的另一个端口的读指令或写指令所做的那样，则判优器 4015 确定指令到达的次序。如果刷新指令比其它指令晚，则取消刷新指令。在这个情形中，判优器 4015 产生 BUSY 信号，并将它加到器件外部。当检测出 BUSY 信号时，外部控制器在切断 BUSY 信号后再次向多端口存储器 4010 提供刷新指令。

如果刷新指令比其它指令早，或者从自刷新电路 4013 提供自刷新指令，则判优器 4015 产生计数信号，并将它加到刷新地址计数器 4016。

刷新地址计数器 4016 响应计数信号对地址进行计数，从而产生刷新地址。需要从判优器 4015 提供计数信号的理由是因为如上所述刷新指令能够被取消，所以计数操作应该只响应从判优器 4015 实际发出的刷新指令进行。这里，在实施刷新操作后实施计数操作。

如果输入到 A 端口 4011 的指令是 Read 指令(读出指令)或 Write 指令(写入指令)，则地址改变电路 4017 将从外部输入到 A 端口 4011 的地址传输给总线 A 4020-1。如果输入到 A 端口 4011 的指令是刷新指令，则将由刷新地址计数器 4016 产生的地址传输给总线 A 4020-1。

如果输入到 B 端口 4012 的指令是 Read 指令(读出指令)或 Write 指令(写入指令)，则地址改变电路 4018 将从外部输入到 B 端口 4012 的地址传输给总线 B 4020-2。另一方面，如果输入到 B 端口 4012 的

指令是刷新指令,则将由刷新地址计数器 4016 产生的地址传输给总线 B 4020-2。

如上所述,如果 A 端口 4011 和 B 端口 4012 两者都处于去基活状态,则自刷新电路 4013 根据作为内部电路提供的刷新定时器 4046 的定时信号产生刷新指令。在这个实施例中,通过总线 A 4020-1 将自刷新指令和刷新地址传输给存储块 4014-1 到 4014-n。因为自刷新不与 A 端口 4011 和 B 端口 4012 的指令冲突,所以不需要判优器 4015 确定优先权。然而,因为需要由判优器 4015 产生计数信号,所以也将自刷新指令提供给判优器 4015。

图 102 是表示根据本发明的多端口存储器 4010 操作的一个例子的定时图。

指令 Read-x 是指向存储块 4014-(x+1)的 Read 指令。首先将 Read-0 输入到 A 端口 4011,然后将 Read-3 输入到 B 端口 4012。在这个情形中,存取的存储块是不同的,使得存储块 4014-1 和存储块 4014-4 并行地操作。

此后,将 Read-1 输入到 A 端口 4011,接着将 Read-1 输入到 B 端口 4012。因为在这个情形中存取的存储块是相同的,所以产生匹配信号,取消输入到 B 端口 4012 的指令。而且,从 B 端口 4012 的 BUSY 信号 I/O 装置 4046 输出 BUSY-B (负逻辑值)。

B 端口 4012 的外部控制器检测 BUSY-B,在切断这个信号后再次向多端口存储器 4010 提供 Read-1。

图 103 是表示根据本发明的多端口存储器 4010 操作的另一个例子的定时图。

图 103 所示的操作直到将第 2 个指令 Read-1 输入 A 端口 4011 和 B 端口 4012,产生 BUSY-B 为止都与图 102 的相同。在这个例子中,在响应输入到 B 端口 4012 的 Read-1 发生 BUSY-B 后,为了在 BUSY-B 结束前存取另一个存储块进入读指令 Read-2。在这个方式中,只要下一个指令是指向另一个存储块的甚至在认定 BUSY 的周期中也能够输入下一个指令。

图 104 是表示根据本发明的多端口存储器 4010 操作的又一个例子的定时图。

图 104 的例子表示输入 Write 指令的情形。将 Read 指令输入 A 端口 4011，接着将 Write 指令输入 B 端口 4012。

在这个实施例中，输入/输出数据是脉冲串型的。即，通过从多个列地址读出并行数据，并当数据输出时在数据 I/O 电路 4033 和 4043 中将它变换成串行数据得到数据输出。串行地输入数据输入，然后在数据 I/O 电路 4033 和 4043 中将它变换成并行数据，接着将并行数据写入相关存储块的多个列地址中。使用这种脉冲串操作能够增加数据传输速率。在这个例子中，脉冲串长度为 4，使连续地输出/输入 4 个数据项。

在 Write 操作的情形，除非输入所有的 4 个数据项，否则不能开始 Write 操作。所以，判优器 4015 能够确定 Write 操作的优先权的定时是给出一系列串行数据输入的最后项的定时。

在图 104 中，A 端口 4011 的第 3 个指令输入 Read-3 和 B 端口 4012 的第 2 个指令输入 Write-3 企图存取同一个存储块。虽然 B 端口 4012 的 Write-3 依据指令到各端口的输入定时在其它的指令输入前面，但是在写数据最后一项进入前给出了 A 端口 4011 的 Read-3。因此，判优器 4015 确定 A 端口 4011 的指令在其它指令的前面，并取消 B 端口 4012 的指令。

如图 101 所示，A 端口 4011 和 B 端口 4012 分别具有 CLK 缓冲器 4032 和 4042，并从器件外部接收不同的时钟信号。各时钟信号可以具有相同或不同的相位和频率。

图 105 是表示指令译码器寄存器 4034 和 4044 的方框图。

指令译码器寄存器 4034 包括输入缓冲器 4061，指令译码器 4062 和 (n-1) 时钟延迟电路 4063。指令译码器寄存器 4044 包括输入缓冲器 4071，指令译码器 4072 和 (n-1) 时钟延迟电路 4073。

如果输入到输入缓冲器 4061 或 4071 的指令是 Read 指令 (RA1, RB1) 或刷新指令 (REFA, REFB)，通过指令译码器 4062 或 4072

将输入指令传输给判优器 4015 而不需要任何定时操作。在 Write 指令 (WA1, WB1) 的情形, 输入指令被(n-1)时钟延迟电路 4063 或 4073 延迟(n-1)个时钟周期, 在给出一系列脉冲串写输入的最后的第 n 个数据项的定时传输给判优器 4015。

图 106 是根据本发明的实施例的判优器 4015 的方框图。

判优器 4015 包括寄存器 4081, 延迟电路 4082, 传输门 4083, 寄存器 4084, 寄存器 4085, 延迟电路 4086, 传输门 4087, 寄存器 4088, NOR 电路 4091 和 4092, NAND 电路 4093 到 4096, 倒相器 4097 到 4101, 与 NOR 电路 4102 和 4103。

将从指令译码寄存器 4034 或 4044 传输过来的指令分别存储在寄存器 4081 或 4085 中。当将指令输入给予 A 端口 4011 时, 在是倒相器 4097 的输出端的节点 N1 产生 HIGH 信号。当将指令输入给予 B 端口 4012 时, 在是倒相器 4100 的输出端的节点 N2 产生 HIGH 信号。将 N1 的信号或 N2 的信号中较早的一个锁存在节点 N3 或 N4 中。

如果在 A 端口 4011 和 B 端口 4012 之间块选择地址不匹配, 则地址比较器 4019 产生为 LOW 的匹配信号。所以, 在这个情形中, 将 N5 和 N6 设置在 HIGH。响应这些 HIGH 信号, 传输门 A 4083 和传输门 B 4087 两者都打开, 无例外地将寄存器 4081 和 4085 的指令传输给寄存器 4084 和 4088。

如果在 A 端口 4011 和 B 端口 4012 之间块选择地址匹配, 则地址比较器 4019 产生为 HIGH 的匹配信号。所以, 在这个情形中, 在节点 N5 和 N6 的信号电平受到在节点 N3 和 N4 的信号电平的控制。如果 A 端口 4011 较早, 则将 N5 设置在 HIGH, 将 N6 设置在 LOW。响应 N5 的 HIGH 状态, 传输门 A 4083 打开, 将 A 端口 4011 的指令传输给寄存器 4084。进一步, N6 的 LOW 状态关闭传输门 B 4087, 不将 B 端口 4012 的指令传输给寄存器 4088。

而且, 根据 N5 和 N6 的信号电平, 产生复位信号 BUSY1-A 和 BUSY1-B, 使各寄存器 4081 和 4085 复位。例如, 如果选择 A 端口 4011 的指令, 则产生 BUSY1-B 和使寄存器 4085 复位。

不需要确定自刷新指令的优先权,在寄存器 4084 的输出级使自刷新指令与 A 端口 4011 的刷新指令 REFA 组合起来。在这个方式中对于 A 端口 4011 产生的刷新指令信号 REFA2 与 B 端口 4012 的刷新指令信号 REFB2 组合起来以便产生计数信号。响应刷新指令的发生,从判优器 4015 将计数信号提供给刷新地址计数器 4016。

图 107 是表示判优器 4015 操作的定时图。

图 107 表示在 A 端口 4011 和 B 端口 4012 之间块选择地址匹配,和 A 端口 4011 的 Read 指令 RA1 比 B 端口 4012 的 Read 指令 RB1 早的情形。在与上述相同的方式中,节点 N5 和 N6 的信号电平受到节点 N3 和 N4 的信号电平的控制,节点 N3 和 N4 的信号电平反应出节点 N1 和 N2 的信号电平,并从判优器 4015 传输出 Read 指令 RA2。取消 B 端口 4012 的 Read 指令而不输出,并产生 BUSY1-B 信号。

图 108 是地址缓冲器/寄存器和地址改变电路的方框图。

在图 108 中,具有在信号名称(例如 RA1)的末端加上字母“P”的信号名称(例如 RA1P)的信号是通过在具有后一个信号名称(例如 RA1)的信号的前沿定时产生脉冲而产生的。

A 端口 4011 的地址缓冲器/寄存器 4035 包括输入缓冲器 4035-1,传输门 4035-2 和 OR 电路 4035-3。对于从图 105 所示的指令译码器 4062 输出的读指令信号 RA1,将前沿转变为脉冲,产生脉冲信号 RA1P,然后将它加到 OR 电路 4035-3 的一个输入端。对于从图 105 所示的指令译码器 4062 输出的写指令信号 WA1,将前沿转变为脉冲,产生脉冲信号 WA1P,然后将它加到 OR 电路 4035-3 的另一个输入端。将 OR 电路 4035-3 的输出加到传输门 4035-2 作为发出进行数据传输的指令的传输定向信号。

B 端口 4012 的地址缓冲器/寄存器 4045 包括输入缓冲器 4045-1,传输门 4045-2 和 OR 电路 4045-3。对于 B 端口 4012 的地址缓冲器/寄存器 4045 的配置与对于 A 端口 4011 的地址缓冲器/寄存器 4035 的配置相同。

地址改变电路 4017 包括地址锁存器 4017-1,传输门 4017-2 和

4017-3, 地址锁存器 4017-4, 与 OR 电路 4017-5 和 4017-6。OR 电路 4017-5 接收信号 RA1P 和 WAD1P, 并将它的输出加到传输门 4017-2 作为传输指示信号。OR 电路 4017-6 接收信号 REFAP 和 SR-AP, 并将它的输出加到传输门 4017-3 作为传输指示信号。

地址改变电路 4018 包括地址锁存器 4018-1, 传输门 4018-2 和 4018-3, 地址锁存器 4018-4, 与 OR 电路 4018-5。OR 电路 4018-5 接收信号 RB1P 和 WBD1P, 并将它的输出加到传输门 4018-3 作为传输指示信号。又将信号 REFBP 加到传输门 4018-2 作为传输指示信号。

当从器件外部输入 Read 指令或 Write 指令时, 将与指令一起输入的地址传输给地址改变电路 4017 或 4018。在 Read 指令的情形中, 将指令传输给地址锁存器 4017-4 或 4018-4 而不需要任何定时操作。在 Write 指令的情形中, 在取得一系列写数据输入的最后一项的定时将指令传输给地址锁存器 4017-4 或 4018-4。

在刷新指令的情形中, 在信号 REFA, REFB 或 ER-A 的定时将由刷新地址计数器 4016 产生的刷新地址传输给地址锁存器 4017-4 或 4018-4。

图 109 是存储块的方框图。

图 109 表示存储块 4014-1 作为存储块 4014-1 到 4014-n 的一个例子。存储块 4014-1 到 4014-n 具有相同的配置。

存储块 4014-1 包括存储器阵列 4111, 控制电路 4112, 总线选择器 4113 和 4114, 读出放大器缓冲器 4115 和写放大器 4116。存储器阵列 4111 包括 DRAM 存储单元, 存储单元门晶体管, 字线, 位线, 读出放大器, 列线, 列门等, 并存储用于读操作和写操作的数据。控制电路 4112 控制存储块 4014-1 的操作。写放大器 4116 放大写入存储器阵列 4111 的数据。读出缓冲器 4115 放大从存储器阵列 4111 读出的数据。

控制电路 4112 与总线 A 4020-1 和总线 B 4020-2 连接, 并响应与它自己的存储块对应的相关的存储块选择地址被选出来。当选出时, 控制电路 4112 从已经发出相关的存储块选择地址的一条总线取得指

令。如果取得了总线 A 4020-1 指令，则控制总线选择器 4113 使它向存储器阵列 4111 发送总线 A 4020-1 的地址信号。进一步，控制总线选择器 4114 使它将读出缓冲器 4115 或写放大器 4116 与总线 A 4020-1 的数据线连接起来。如果取得了总线 B 4020-2 指令，则控制总线选择器 4113 使它向存储器阵列 4111 发送总线 B 4020-2 的地址信号。进一步，控制总线选择器 4114 使它将读出缓冲器 4115 或写放大器 4116 与总线 B 4020-2 的数据线连接起来。如果控制电路 4112 取得的指令是刷新指令，则总线选择器 4114 不需要操作。

如上所述选择一条总线，然后，作为一系列的连续操作，连续地实施字线选择，存储单元数据放大，或者 Read, Write 或者 Refresh (刷新)，和预充电操作。

图 110A 和 110B 是表示存储块操作的定时图。

图 110A 表示读操作的情形，图 110B 表示写操作的情形。在图 110A 和 110B 所示操作定时，响应单个指令实施字线选择，数据放大，或者读操作或者写操作，写回（数据恢复）操作和预充电操作，从而完成要求的操作。

在本发明（第 4 方面）中，如果从器件外部输入端口的指令企图存取同一个存储块，则判优电路确定各指令中的哪一个要被执行，和各指令中的哪一个不被执行。例如，比较指令定时，执行较早的指令，而不执行(各)其它的指令。当存在一个不被执行的指令时，产生 BUSY 信号等并输出到器件外部。这使得甚至当在以 DRAM 芯为基础的多端口存储器中指令存取相互发生冲突时也可以实施适当的存取操作并实现适当的 BUSY 控制。

进一步，本发明具有用于响应来自器件外部的指令实施刷新操作的操作模式，和用于响应来自内部刷新电路的指令实施刷新操作的操作模式。这使得在这样一种方式，即将预定外部端口指定作为用于在恒定的间隔接收刷新指令进行刷新管理的端口中用多端口存储器成为可能，或者使得在这样一种方式，即当所有的外部环部端口都处于去激活状态时内部刷新电路开始刷新操作中用多端口存储器成为可能。

因此，本发明为遵从系统要求的灵活的刷新管理提供了基础。

[本发明的第 5 方面]

下面我们描述本发明的第 5 方面。

多端口存储器具有两组或多组输入/输出端子（即，多个输入/输出端口），并实施与接到的信号相应的存储操作。与普通的存储器不同，可以同时执行读操作和写操作。例如，如果系统中存在多条总线，如果多个控制器（CPU 等）需要用各条总线，则通过将多端口存储器的输入/输出端口与各条总线连接起来能够制成该系统。这就消除了用特殊设计的控制逻辑电路（FIFO 逻辑电路等）的需要。

而且，我们也将多个端口存储器开发成图象存储器（一般为双端口报告存储器）。图象存储器具有随机存取端口，通过这些端口能够实现到任何存储单元的存取，和与显示装置交换数据的串行存取端口。

这类多个端口存储器在存储单元区域中采用 SRAM 存储芯或 DRAM 存储芯。

然而，我们还必须开发这样的多端口存储器，它们接收在各输入/输出端口的不同时钟信号并与时钟信号同步地随机地存取一个存储单元区域。即，我们还不知道如何制成电路的详细情况和如何控制这种时钟同步的多端口存储器。

而且，常规的多端口存储器（具体地双端口存储器）具有分别对于各组输入/输出端口的位线和读出放大器。因为这个原因，存在存储器芯的布局尺寸变大，从而不希望地增大多端口存储器的芯片尺寸的问题。

因此，本发明的目的是提供允许能够进行随机存取的时钟同步的多端口存储器。

本发明的目的是进一步提供在各组输入/输出端口接收相互不同的时钟信号，并以可靠的方式进行操作的多端口存储器。

而且本发明的目的是提供能够通过与其它输入/输出端口的状态无关地接收在任何时间的指令信号驱动存储芯的多端口存储器。

而且本发明的目的是提供具有减少的芯片尺寸的小的多端口存储器。

根据本发明(第5方面),多个存储芯中的一些在加到多个输入/输出端口的时钟信号和地址信号的基础上进行操作。每个输入/输出端口包括用于接收时钟信号的时钟端子,用于接收与时钟信号同步地提供的地址信号的地址端子,和用于输入/输出数据信号的数据输入/输出端口端子。为各存储芯提供控制电路。

如果地址信号指示将同一个存储芯加到两个或多个输入/输出端口,则控制电路使存储芯响应首先接到的地址信号进行操作。即,对于首先接收地址信号的输入/输出端口实施存储操作。可以如此定义存储芯,使它与各读出放大器区域相当,其中读出放大器区域是各读出放大器区域在其中一起操作的区域。由地址信号的上部分选择存储芯。由地址信号的下部分选择存储芯的存储单元。通过与首先接到的地址信号上部分对应的输入/输出端口输入或输出到与首先接到的地址信号上部分对应的输入/输出端口,将由地址信号的下部分选择的存储单元的数据信号输入到器件外部或从器件外部输出。

因为所要做的全部事情就是比较地址信号,所以能够将控制电路制成简单的电路。这使芯片尺寸减少。

因为每个输入/输出端口都具有时钟端子,所以对于每个输入/输出端口能够分别地控制时钟信号的频率。即,能够使具有不同操作频率的多个控制器与

多端口存储器连接起来。

在本发明的多端口存储器中,在用于取得地址信号的时钟信号的特定边沿前的预定设置时间安排地址信号。控制电路用在时钟信号的这个特定边沿前安排的地址信号确定地址信号到达的次序。因此,能够用首先接收的时钟信号的边沿确定地址信号到达的次序。这使在存储芯开始操作前可以识别具有优先权的输入/输出端口,从而实现高速存储操作。因为在预定定时(即时钟信号的边沿)比较地址信号,所以能够防止与存储操作无关的地址信号的错误比较。

根据本发明，多个存储芯中的一些在加到多个输入/输出端口的时钟信号和地址信号的基础上进行操作。每个输入/输出端口包括用于接收时钟信号的时钟端子，用于接收与时钟信号同步地提供的地址信号的地址端子，和用于输入/输出数据信号的数据输入/输出端口端子。为各存储芯提供控制电路。

如果将指示同一个存储芯的地址信号加到两个或多个输入/输出端口，则控制电路使存储芯响应首先接到的地址信号进行操作。此后，控制电路使存储芯响应地址信号以接收地址信号的次序进行操作。由地址信号的上部分选择存储芯。由地址信号的下部分选择存储芯的存储单元。通过与各地址信号对应的输入/输出端口，连续地从器件外部输入由地址信号下部分选择的存储单元的数据信号或将由地址信号下部分选择的存储单元的数据信号输出到器件外部。因此，对于所有的接收存储操作要求的输入/输出端口没有例外地实施存储操作。

即，在所有时间中多端口存储器都处于备用状态。与多端口存储器连接的控制器不一定要检测多端口存储器的忙碌状态。这通过硬件和软件简化了控制器的操作。因为所要做的全部事情就是比较地址信号，所以能够将控制电路制成简单的电路。这使芯片尺寸减少。

因为每个输入/输出端口都具有时钟端子，所以对于每个输入/输出端口能够分别地控制时钟信号的频率。即，能够使具有不同操作频率的多个控制器与

多端口存储器连接起来。

在本发明的多端口存储器中，每个输入/输出端口都具有指令端子，用于与控制存储芯操作的时钟信号同步地接收指令信号。在每个输入/输出端口中，在至少是读操作和写操作需要的存储芯的操作周期两倍长的间隔上，提供用于激活存储芯的指令信号。如果多端口存储器具有 2 个输入/输出端口或 4 个输入/输出端口，则可以分别将指令信号的间隔设置在操作周期的 2 倍或操作周期的 4 倍上。有了这些设置，多端口存储器处于响应外部控制器的备用状态。

如果在比预定间隔短的间隔上提供指令信号，则指令信号在防止

故障方面是无效的。如果将指令信号提供给不同的输入/输出端口，则即便间隔不比预定间隔短也接收这些指令信号。

根据本发明，进一步，从存储单元读出或写入存储单元的数据通过缓冲器在数据输入/输出端口和存储单元之间传输。缓冲器在其中存储具有预定在数量上等于两个或多个存储单元的位数的数据。

在开始读操作和写操作时，例如，将具有预先确定数目的数据从存储单元传输到缓冲器。在读操作中，从缓冲器读出与各地址信号对应的数据，并从数据输入/输出端口输出到外部装置。在写操作中，将与各地址信号对应的数据存储在缓冲器中，并在写操作结束时立即将缓冲器的数据写入存储单元。

在这种方式中，容易实施页面操作。一般，当页面操作时存储芯（读出放大器等）必须保持在激活状态。如果不提供本发明的缓冲器，则当对于输入/输出端口实施的页面操作时不可能对于另一个输入/输出端口实施存储操作。在本发明中，在开始操作时将存储单元的数据传输给缓冲器，使得在这以后能够立即使存储芯去激活。结果，甚至当页面操作时与多端口存储器连接的控制器也不一定要检测多端口存储器的忙碌状态。

下面我们参照所附各图描述本发明（第5方面）的实施例。

图 111 表示根据本发明（第5方面）的多端口存储器的第1实施例。用 CMOS 工艺在硅基片上形成多端口存储器 M。

多端口存储器 M 包括两个输入/输出端口 PPRT-A 和 PORT-B，I/O 电路 5010，它向端口 PPRT-A 和 PORT-B 输出信号和从端口 PPRT-A 和 PORT-B 输入信号，和多个存储块 MB。每个存储块 MB 都包括 DRAM 存储芯（包括存储单元，读出放大器线 SA 等），并进一步包括图中未画出的控制电路，译码器等。每个存储单元包括存储与数据信号值相应的电荷的电容器。根据通过端口 PPRT-A 和 PORT-B 提供的行地址信号选择一个存储芯。响应给定的存储芯的选择同时激活在给定的存储芯中的读出放大器线 SA 的所有读出放大器。即，响应激活指令 ACT 激活存储芯，我们将在后面对此进行描述，并

选择在这个存储芯中的所有的存储单元区域。根据读出放大器激活后提供的列地址信号在存储芯上读出读数据或写入写数据。

图 112 表示多端口存储器 M 的 I/O 电路 5010 和存储块 MB 的详细情况。在图中，每条由粗线表示的信号线都由多条线组成。

I/O 电路 5010 包括模寄存器 5012a 和 5012b，时钟缓冲器 5014a 和 5014b，

数据输入/输出缓冲器 5016a 和 5016b，地址缓冲器/寄存器 5018a 和 5018b，指令缓冲器 5020a 和 5020b，与忙碌缓冲器 5022a 和 5022b，分别与输入/输出端口 PPRT-A 和 PORT-B 对应。模寄存器 5012a 和 5012b 是用于从器件外部设置多端口存储器 M 的操作模式的寄存器。

时钟缓冲器 5014a，地址缓冲器/寄存器 5018a，和指令缓冲器 5020a 分别将时钟信号 CLKA，地址信号 ADDA 和指令信号 CMDA 加到存储块 MB 上，就如从器件外部加上一样。数据输入/输出缓冲器 5016a 用于从存储块 MB 输出数据信号 DQA 和将数据信号 DQA 输入存储块 MB。忙碌缓冲器 5022a 向器件外部输出忙碌信号/BSYA。时钟缓冲器 5014b，地址缓冲器/寄存器 5018b，和指令缓冲器 5020b 分别将时钟信号 CLKB，地址信号 ADDB 和指令信号 CMDB 加到存储块 MB 上，就如从器件外部加上一样。数据输入/输出缓冲器 5016b 用于从存储块 MB 输出数据信号 DQB 和将数据信号 DQB 输入存储块 MB。忙碌缓冲器 5022b 向器件外部输出忙碌信号/BSYB。时钟信号 CLKA 和 CLKB，地址信号 ADDA 和 ADDB 和指令信号 CMDA 和 CMDB，数据信号 DQA 和 DQB，与忙碌信号/BSYA 和/BSYB 分别通过时钟端子，地址端子，指令端子，数据输入/输出端子和忙碌端子进行传输。提供激活指令 ACT 和操作指令（例如读指令 RD，写指令 WR）等作为用于控制存储芯操作的指令信号 CMDA 和 CMDB。

提供每个地址信号 ADDA 和 ADDB 作为相互分开的行地址信号 RA 和列地址信号 CA。在输入/输出端口 PPRT-A 中，与时钟信号 CLKA 的前沿同步地提供行地址信号 RA，列地址信号 CA 和指令信号 CMDA。在输入/输出端口 PPRT-B 中，与时钟信号 CLKB 的前沿

同步地提供行地址信号 **RB**，列地址信号 **CB** 和指令信号 **CMDB**。在这个方式中，多端口存储器 **M** 分别接收专门在输入/输出端口 **PPRT-A** 和 **PORT-B** 使用的时钟信号 **CLKA** 和 **CLKB**，并与时钟信号 **CLKA** 和 **CLKB** 同步地进行操作。

存储块 **MB** 包括时钟缓冲器 **5024a** 和 **5024b**，指令锁存器 **5026a** 和 **5026b**，数据锁存器 **5028a** 和 **5028b**，行地址锁存器 **5030a** 和 **5030b**，列地址锁存器 **5031a** 和 **5031b**，与列地址锁存器 **5032a** 和 **5032b**，分别与输入/输出端口 **PPRT-A** 和 **PORT-B** 对应。存储块 **MB** 包括判优电路 **5034**，控制信号锁存器 **5036**，列地址计数器 **5038** 和存储芯 **5040**，它们对于输入/输出端口 **PPRT-A** 和 **PORT-B** 是公用的。存储芯 **5040** 具有与时钟信号同步地取得指令信号 **RAS**，**CAS** 和 **WE**，行地址信号 **RA** 和列地址信号 **CA** 的形式。

当激活从判优电路 **5034** 提供的启动信号/**ENA** 时，与输入/输出端口 **PPRT-A** 对应的模寄存器 **5012a**，时钟缓冲器 **5024a**，指令锁存器 **5026a**，数据锁存器 **5028a**，行地址锁存器 **5031a**，和列地址锁存器 **5032a** 进行操作。当激活从判优电路 **5034** 提供的启动信号/**ENB** 时，与输入/输出端口 **PPRT-B** 对应的模寄存器 **5012b**，时钟缓冲器 **5024b**，指令锁存器 **5026b**，数据锁存器 **5028b**，行地址锁存器 **5031b**，和列地址锁存器 **5032b** 进行操作。

即，在激活启动信号/**ENA** 时，时钟缓冲器 **5024a** 向存储芯 **5040** 的时钟端子 **CLK** 提供时钟信号 **CLKA**。进一步，指令锁存器 **5026a** 向控制信号锁存器 **5036** 提供锁存的指令信号 **CMDA**，和行地址锁存器 **5031a** 向存储芯 **5040** 的行地址端子 **RA** 提供锁存的行地址信号 **RA**（例如，与上地址位对应）。而且，列地址锁存器 **5032a** 向列地址计数器 **5038** 提供锁存的列地址信号 **CA**（例如，与下地址位对应），和数据锁存器 **5028a** 与存储芯 **5040** 的数据输入/输出端子 **DQ** 和输入/输出缓冲器 **5016a** 交换数据信号。

类似地，在激活启动信号/**ENB** 时，时钟缓冲器 **5024ba** 向存储芯 **5040** 的时钟端子 **CLK** 提供时钟信号 **CLKB**。进一步，指令锁存器 **5026b**

向控制信号锁存器 5036 提供锁存的指令信号 **CMDB**，和行地址锁存器 5031b 向存储芯 5040 的行地址端子 **RA** 提供锁存的列信号 **RA**。而且，列地址锁存器 5032b 向列地址计数器 5038 提供锁存的列地址信号 **CA**，和数据锁存器 5028b 与存储芯 5040 的数据输入/输出端子 **DQ** 和输入/输出缓冲器 5016b 交换数据信号。

控制信号锁存器 5036 根据接收的指令信号 **CMDA** 和 **CMDB** 产生用于使存储芯 5040 操作的行地址选通信号 **RAS**，列地址选通信号 **CAS**，和写启动信号 **WE**，并将产生的信号加到存储芯 5040 上。而且，控制信号锁存器 5036 向判优电路 5034 提供指示读操作和写操作中的一个的读/写指令信号 **RWCMD**。

列地址计数器 5038 根据关于从模寄存器 5012a 和 5012b 提供的脉冲串长度的信息以及地址信号 **ADDA** 和 **ADDB** 产生列地址信号 **CA**，并向存储芯 5040 输出列地址信号。

判优电路 5034 包括地址比较电路 5042 和判优控制电路 5044。地址比较电路 5042 比较在从输入/输出端口 **PPRT-A** 和 **PORT-B** 提供的地址信号 **ADDA** 和 **ADDB** 之间的行地址信号 **RA**，并决定它们中哪一个较早到达。判优控制电路 5044 产生忙碌信号 **/BSYA** 和 **/BSYB** 与启动信号 **/ENA** 和 **/ENB**，用于使内部电路根据地址比较电路 5042 的比较结果进行操作。

图 113 表示地址比较电路 5042 的详细情况。

地址比较电路 5042 包括两个地址匹配电路 5042a 和一个地址比较器 5042b，地址比较器 5042b 检测地址到达的次序。地址匹配电路 5042a 包括多个 **EOR** 电路 5042c，每个 **EOR** 电路 5042c 比较在地址信号 **ADDA** 和地址信号 **ADDB** 之间的行地址信号 **RA** 的对应位，并进一步包括多个 **nMOS** 晶体管 5042d，它们与各 **EOR** 电路 5042c 对应。每个 **nMOS** 晶体管 5042d 的栅极都与对应的 **EOR** 电路 5042c 的输出端连接，它们的源极接地和它们的漏极相互连接。每个 **EOR** 电路 5042c，当行地址信号 **RA** 的位值在输入/输出端口 **PPRT-A** 和 **PORT-B** 之间相互匹配时输出低电平信号，当行地址信号 **RA** 的位值不匹配时输出高

电平信号。响应来自 EOR 电路 5042c 的低电平信号切断 nMOS 晶体管 5042d，响应来自 EOR 电路 5042c 的高电平信号接通 nMOS 晶体管 5042d。即，从地址匹配电路 5042a 输出的匹配信号/COIN1 和/COIN2 当行地址信号 RA 的所有的位在对应的位之间匹配时变成浮动的，并当行地址信号的至少一个位在对应的位之间不同时变成低电平信号。将两个地址匹配电路 5042a 分别安排在存储块 MB 的上端和下端如图 111 所示（即，安排得接近输入/输出电路 5010）。地址匹配电路 5042a 接近输入/输出电路 5010 的安排可以缩短地址信号 ADDA 和 ADDB 到地址匹配电路 5042a 的整个路经上的传播延迟。因此，可以在较早的定时比较地址信号 ADDA 和 ADDB，从而得到高速操作。

比较器 5042b 接收匹配信号/COIN1 和/COIN2 和时钟信号 CLKA 和 CLKB，并输出首先到达信号/FSTA 和/FSTB。

图 114 表示比较器 5042b 的详细情况。

比较器 5042b 包括脉冲发生器 5042e，它们分别与时钟信号 CLKA 和 CLKB 的前沿同步地产生正脉冲 PLSA 和 PLSB，并进一步包括触发器 5042f，它在它的输入端子接收脉冲 PLSA 和 PLSB。比较器 5042b 接收匹配信号/COIN1 和/COIN2 并输入到分别输出脉冲 PLSA 和 PLSB 的各倒相器。将在比较器 5042b 中产生各脉冲信号的 NAND 门制成小尺寸的电路元件，使得当从 NAND 门输出的信号具有与匹配信号/COIN1 和/COIN2 冲突的信号电平时将优先权给予匹配信号/COIN1 和/COIN2。触发器 5042f 当接收脉冲 PLSA 时使首先到达信号/FSTA 下降到低电平，当接收脉冲 PLSB 时使首先到达信号/FSTB 下降到低电平。

图 115 表示当加到输入/输出端口 PORT-A 和 PORT-B 的行地址信号相互匹配时实施的比较器 5042b 的操作。在这个例子中，时钟信号 CLKA 和 CLKB 具有相同的周期。

图 113 所示的地址匹配电路 5042a 当行地址信号 RA 匹配时使匹配信号/COIN1 和/COIN2 处于浮动状态 (Hi-z)。对应地，分别与时钟信号 CLKA 和 CLKB 的前沿同步地产生脉冲 PLSA 和 PLSB (图

115- (a))。图 114 所示的触发器 5042f 响应在其它信号之前接收的脉冲 PLSA 使首先到达信号/FSTA 激活 (图 115- (b))。在使首先到达信号/FSTA 去激活后使与以后接收的脉冲 PLSB 对应的首先到达信号/FSTB 激活 (图 115- (c))。

图 116 表示当行地址信号 RA 在输入/输出端口 PORT-A 和 PORT-B 之间不匹配时比较器 5042b 的操作。在这个例子中, 时钟信号 CLKA 和 CLKB 具有相同的周期。

地址匹配电路 5042a 当行地址信号 RA 甚至一个位都不匹配时使每个匹配信号/COIN1 和/COIN2 降到低电平 (图 116- (a))。对应地, 图 114 所示的脉冲发生器 5042e 迫使脉冲信号 PLSA 和 PLSB 降到低电平而与时钟信号 CLKA 和 CLKB 无关 (图 116- (b))。因此, 首先到达信号/FSTA 和/FSTB 保持在高电平 (图 115- (c))。

图 117 表示当加到输入/输出端口 PORT-A 和 PORT-B 的行地址信号 RA 在时钟信号 CLKA 具有一个与时钟信号 CLKB 的周期不同的周期的条件下匹配时比较器 5042b 的操作。在这个例子中, 设置时钟信号 CLKB 的周期等于时钟信号 CLKA 的周期的两倍。分别与时钟信号 CLKA 和 CLKB 的前沿同步地取得行地址信号 RA。在图中, 实线表示的行地址信号 RA 说明加到输入/输出端口 PORT-A 和 PORT-B 的信号, 虚线表示的行地址信号 RA 说明由图 112 所示的各行地址锁存器 5030a 和 5030b 锁存的信号。

当行地址信号 RA 匹配时, 以与图 115 相同的方式使匹配信号/COIN1 和/COIN2 处于浮动状态 (Hi-z)。当匹配信号/COIN1 和/COIN2 处于浮动状态 (Hi-z) 时, 图 114 所示的脉冲发生器 5042e 发挥作用, 使得分别与时钟信号 CLKA 和 CLKB 的前沿同步地产生脉冲信号 PLSA 和 PLSB 和首先到达信号/FSTA 和/FSTB。

图 118 表示提供给图 112 所示的判优电路的判优控制电路 5044。

判优控制电路 5044 包括控制电路 5044a 和 5044b, 它们分别与输入/输出端口 PPRT-A 和 PORT-B 对应。控制电路 5044a 接收复位信号 RESETA, 延迟时钟信号 DCLKA, 有效指令信号 ACTA, 首先到

达信号/FSTA, 忙碌信号/BSYA, 并输出启动信号/ENA 和忙碌信号/BSYB。控制电路 5044b 接收复位信号 RESETB, 延迟时钟信号 DCLKB, 有效指令信号 ACTB, 首先到达信号/FSTB, 忙碌信号/BSYB, 并输出启动信号/ENB 和忙碌信号/BSYA。

当完成与输入/输出端口 PPRT-A 和 PORT-B 对应的读或写操作时, 在各预定周期中使复位信号 RESETA 和 RESETB 激活。延迟时钟信号 DCLKA 和 DCLKB 是分别通过使时钟信号 CLKA 和 CLKB 延迟得到的。当将有效指令 ACT 加到输入/输出端口 PPRT-A 和 PORT-B 时使有效指令信号 ACTA 和 ACTB 激活。

图 119 表示当加到输入/输出端口 PORT-A 和 PORT-B 的行地址信号匹配时实施的判优控制电路 5044 的操作。在这个例子中, 时钟信号 CLKA 和 CLKB 的周期是相同的。与时钟信号 CLKA 同步地提供有效指令 ACT, 立即接着与时钟信号 CLKB 同步地提供有效指令 ACT。

控制电路 5044a 与延迟时钟信号 DCLKA 的前沿同步地取得低电平的首先进达信号/FSTA, 并使忙碌信号/BSYB 激活(图 119-(a))。响应有效指令信号 ACTA 的激活和忙碌信号/BSYA 的去激活状态, 控制电路 5044a 激活启动信号 ENA(图 119-(b))。因为控制电路 5044b 与延迟时钟信号 DCLKB 的前沿同步地取得高电平的首先进达信号/FSTB, 所以不激活忙碌信号/BSYA(图 119-(c))。虽然控制电路 5044b 接受激活状态的有效指令信号 ACTB, 但是因为激活了忙碌信号/BSYB, 所以控制电路 5044b 不激活启动信号 ENB(图 119-(d))。

响应启动信号 ENA 的激活, 将加到输入/输出端口 PPRT-A 的信号传输给存储芯 5040。激活存储芯 5040, 根据加到输入/输出端口 PPRT-A 的读指令 RD 实施读操作。在完成读操作, 控制电路 5044a 响应复位信号 RESETA 的激活使启动信号 ENA 和忙碌信号/BSYB 去激活(图 119-(e))。

下面, 关于其操作进一步描述上述多端口存储器 M。

图 120 表示当加到输入/输出 PORT-A 和 PORT-B 的行地址信号

RA 相互匹配时实施的操作。在这个例子中, 时钟信号 CLKA 和 CLKB 具有相同的周期, 时钟信号 CLKA 的相位稍微超前时钟信号 CLKB 的相位。通过各模寄存器 5012a 和 5012b 将输入/输出端口 PORT-A 和 PORT-B 的脉冲串长度两者都设置得等于 4。这里, 脉冲串长度是当写或读操作时输出和输入的数据项的数目。

与时钟信号 CLKA 的前沿同步地, 输入/输出端口 PORT-A 接收有效指令 ACT(指令信号 CMDA)和行地址信号 RA(地址信号 ADDA) (图 120-(a))。在输入/输出端口 PORT-A 接收信号后立即与时钟信号 CLKB 的前沿同步地, 输入/输出端口 PORT-B 接收有效指令 ACT(指令信号 CDMB)和行地址信号 RA(地址信号 ADDB) (图 120-(b))。这里, 指令信号 CMDA 和 CDMB 与地址信号 ADDA 和 ADDB 在时钟信号 CLKA 和 CLKB 的各前沿前的一个预先设置时间 t_s (即按照定时说明) 设置它们的信号电平。

因为加到端口 PORT-B 的行地址信号 RA 与加到端口 PORT-A 的行地址信号 RA 相同, 所以一个接着一个地产生首先到达信号/FSTA 和/FSTB 如图 115 所示。判优控制电路 5044 如结合图 119 所描述的响应首先到达信号/FSTA 和/FSTB 激活启动信号 ENA 和忙碌信号/BSYB (图 120-(c) 和 (d))。在这个方式中, 通过用当设置时间 t_s 时提供的行地址信号 RA 和通过利用具有较早相位的时钟信号(在本例中为 CLKA)的前沿, 确定两个地址信号中首先到达的一个。此后, 与行地址信号 RA 对应的存储芯 5040 响应启动信号 ENA 的激活进行操作(图 120-(e))。

响应忙碌信号/BSYB, 控制器如与输入/输出端口 PORT-B 连接的 CPU 认定加到多端口存储器 M 的有效指令 ACT 是无效的。

输入/输出端口 PORT-A 与时钟信号 CLKA 的下一个前沿同步地, 接收读指令 RD(即指令信号 CMDA)和列地址信号 CA(地址信号 ADDA) (图 120-(f))。输入/输出端口 PORT-B 与时钟信号 CLKB 的前沿同步地接收读指令 RD(指令信号 CDMB)和列地址信号 CA(地址信号 ADDB) (图 120-(g))。与各时钟信号 CLKA 和 CLKB

(根据定时说明)的下一个前沿同步地,在有效指令 ACT 后提供读指令 RD (或写指令 WR)。与忙碌信号/BUSY 有关,输入/输出端口 PORT-B 连接的控制器可以不提供读指令 RD 和列地址信号 CA。

存储块 MB 连续地输出数据作为数据信号 DQA (Q0-Q3) (图 120- (h)) 如它们是从与加到输入/输出端口 PORT-A 的列地址信号 CA 对应的存储单元读出的那样。在接收读指令 RD 后的 2 个时钟脉冲,输出数据信号 DQA。在输出与脉冲串长度 (=4) 一样多的数据信号 DQA 后,存储芯 5040 实施预充电操作 (图 120- (i)), 从而完成一个存储周期。响应读操作的完成使启动信号 ENA 去激活 (图 120- (j))。这里,预充电操作使用于将数据传输到存储单元和从存储单元传输出来的位线充电到预定电位,使与行地址操作有关的电路去激活。在每个存储操作中自动地实施预充电操作。根据存储在对应的模寄存器中的输入/输出端口 PORT-A 的脉冲串长度或输入/输出端口 PORT-B 的脉冲串长度中较长的一个确定预充电操作的定时。在这个实施例中,如果脉冲串长度为 4,则存储周期 (即单个读或写操作需要的时间周期) 被固定在 4 个时钟周期。即,总是在接收到有效指令后的预定时间完成读操作和写操作。

与用于输出数据 Q1 的时钟信号 CLKA 同步地,将下一个有效指令 ACT 加到输入/输出端口 PORT-A (图 120- (k))。因为在这个特定的瞬间不将指令信号 CMDB 加到输入/输出端口 PORT-B,所以用图 113 所示的地址比较电路 5042 比较行地址信号 RA,产生指示不匹配的结果。因此,不激活忙碌信号/BSYA 和/BSYB,只激活启动信号 ENA (图 120- (l))。将首先到达信号/FSTA 和/FSTB 保存在高电平,如图 116 所示。

存储芯 5040 根据加到输入/输出端口 PORT-A 的行地址信号 RA 进行操作,如前面描述的那样 (图 120- (m))。存储块 MB 根据与下面的时钟信号 CLKA 同步地提供的读指令 RD 和列地址信号 CA 一个接着一个地输出数据信号 DQA (Q0-Q3) (图 120- (n))。

在与输入/输出端口 PORT-A 对应的存储芯 5040 的操作完成后,

将有效指令 ACT 和读指令 RD 连续地加到输入/输出端口 PORT-B(图 120- (o))。因为在这个特定的瞬间不将指令信号 CMDA 加到输入/输出端口 PORT-A, 所以存储芯 5040 对于输入/输出端口 PORT-B 进行操作, 从而输出数据信号 DQB (图 120- (p))。

虽然在图中未画出, 但是响应与时钟信号的前沿同步地提供的行地址信号 RA 和刷新指令实施恢复在存储单元的电容器中电荷的刷新操作, 其中行地址信号 RA 确定存储芯 5040 要被刷新。或者通过输入/输出端口 PORT-A 或者通过输入/输出端口 PORT-B 能够要求刷新操作。在这个方式中, 由一个存储芯 5040 的装置根据从器件外部提供的地址信号实施刷新操作。

图 121 表示当时钟信号 CLKA 和 CLKB 相同, 时钟信号 CLKA 的相位超前时钟信号 CLKB 的相位多于半个周期时实施的操作。加到多端口存储器 M 的指令信号 CMDA 和 CMDB 与地址信号 ADDA 和 ADDB 与图 120 情形中的相同。

在这个例子中, 当将有效指令 ACT 和行地址信号 RA 加到输入/输出端口 PORT-A 时 (图 121- (a)), 还没有将指令信号 CMDB 与地址信号 ADDB 加到输入/输出端口 PORT-B。因此, 激活启动信号 /ENA (图 121- (b)), 存储芯 5040 对于输入/输出端口 PORT-A 进行操作 (图 121- (c))。此后, 将有效指令 ACT 和与输入/输出端口 PORT-A 相同的行地址信号 RA 加到输入/输出端口 PORT-B (图 121- (d))。

图 118 所示的控制电路 5044b 根据首先到达信号 /FSTA 的激活和启动信号 /ENA 的激活, 激活忙碌信号 /BSYB (图 121- (e))。响应忙碌信号 /BSYB, 与输入/输出端口 PORT-B 连接的控制器如 CPU 认定加到多端口存储器 M 的有效指令 ACT 是无效的。以后的操作与上述图 120 的相同。

图 122 表示当几乎同时加到输入/输出端口 PORT-A 和 PORT-B 的行地址信号 RA 相互不同时的操作。时钟信号 CKLA 和 CLKB 具有相同的时钟周期, 时钟信号 CLKA 的相位稍微超前时钟信号 CLKB

的相位。通过模寄存器 5012 对于输入/输出端口 PORT-A 和端口 PORT-B 两者都将脉冲串长度设置得等于 4。

当行地址信号 RA 不同时，不同的存储芯 5040 进行操作。于是图 114 所示的比较器 5042b 使首先到达信号/FSTA 和/FSTB 两者去激活。即，不进行地址判优。判优控制电路 5044 对应首先到达信号/FSTA 和/FSTB 的去激活状态和有效指令信号 ACTA 和 ACTB 的激活，使启动信号/ENA 和/ENB 激活（图 121-（a）和（b））。结果，相关的存储芯 5040 响应加到输入/输出端口 PORT-A 的有效指令 ACT 和行地址信号 RA 进行操作（图 121-（c）），另一个存储芯 5040 响应加到输入/输出端口 PORT-B 的有效指令 ACT 和行地址信号 RA 进行操作（图 121-（d））。即，输入/输出端口 PORT-A 和 PORT-B 相互独立地进行操作。因为行地址信号 RA 相互不同，既不激活忙碌信号/BSYA 也不激活忙碌信号/BSYB。

在上述的实施例中，当输入/输出端口 PORT-A 和 PORT-B 接收两个指示同一个存储芯 5040 分别与时钟信号 CLKA 和 CLKB 同步的行地址信号 RA 时，存储芯 5040 对于在两个行地址信号 RA 中首先到达的一个进行操作。即，这样我们就能够制成时钟同步型的多端口存储器 M。

判优电路 5034 通过比较行地址信号 RA 满足对它预期的所有要求，于是能够通过简单的配置制成。因此，能够使多端口存储器 M 的芯片尺寸减小。

因为输入/输出端口 PORT-A 和 PORT-B 具有各自的时钟端子 CLKA 和 CLKB，所以能够分别地对输入/输出端口 PORT-A 和 PORT-B 中的每一个设置时钟信号 CLKA 和 CLKB 的频率。即，操作在不同操作频率上的多个控制器能够与多端口存储器 M 连接。

进一步，用设置在时钟信号 CLKA 和 CLKB 的相关前沿前的行地址信号 RA 决定两个地址中首先到达的一个。即，利用地址信号的设置时间识别首先到达的一个。因此，在存储芯 5040 开始它的操作前能够识别将给予优先权的输入/输出端口，从而实现高速存储操作。进

一步，因为根据具有较前相位的时钟信号 CLKA（或 CLKB）的前沿确定首先到达的一个，所以能够进一步提高存储操作速度。

在判优电路 5034 中，地址比较电路 5042 比较行地址信号 RA，判优控制电路 5044 与用于取得有效指令 ACT 的时钟信号 CLKA 和 CLKB 同步地检查地址匹配。因为总是在预定定时（即在定时信号的边沿）相互比较行地址信号 RA，所以可以防止由与存储操作无关的地址信号引起的存储芯 5040 的不正常操作。

图 123 表示多端口存储器的第 2 实施例和根据本发明（第 5 方面）控制多端口存储器的方法。用相同的数字标记与第 1 实施例相同的部件，并省略对它们的详细描述。

在这个实施例中，以第 1 实施例的四分之一大小形成存储块 MB（在图中用粗线框表示）。即，同时被激活的读出放大器的数目是第 1 实施例的四分之一。除了存储块 MB 的大小外，配置与第 1 实施例相同。因为图 123 的多端口存储器 M 具有较少的同时被驱动的读出放大器，所以在存储操作时的功率消耗减少了。

这个实施例能够产生与上述第 1 实施例相同的优点。此外，在本实施例中能够降低功率消耗。

图 124 表示多端口存储器的第 3 实施例和根据本发明（第 5 方面）控制多端口存储器的方法。用相同的数字标记与第 1 实施例相同的部件，并省略对它们的详细描述。

在这个实施例中，在每个存储块 MB 中提供数据寄存器（缓冲器）5046a 和 5046b，它们暂时存储在数据锁存器 5028 和存储芯 5040 之间的各数据信号 DQA 和 DQB。数据寄存器 5046a 和 5046b 与输入/输出端口 PORT-A 和 PORT-B 中的任何一个结合起来进行操作。而且，判优电路 5034 的判优控制电路 5048 不同于第 1 实施例的判优控制电路 5044。判优控制电路 5048 不输出忙碌信号 /BSYA 和 /BSYB，在 I/O 电路 5010 中不提供忙碌缓冲器。其它配置几乎与第 1 实施例相同。即，在输入/输出端口 PORT-A 和 PORT-B 中，分别通过时钟端子，地址端子，指令端子，和数据输入/输出端子，传输时钟信号 CLKA 和

CLKB, 地址信号 ADDA 和 ADDB, 指令信号 CMDA 和 CMDB, 与数据信号 DQA 和 DQB。存储块 MB 包括 DRAM 存储芯 5040, 并进一步包括图中未画出的控制电路, 译码器等。存储单元包括根据数据信号值存储电荷的电容器。

甚至当输入/输出端口 PORT-A 和 PORT-B 同时接收对于同一个地址信号 RA 进行存储操作的要求时, 这个多端口存储器 M 也能够对输入/输出端口 PORT-A 和 PORT-B 两者实施存储操作, 我们将在后面对此进行描述。因此, 如第 1 实施例那样不需要向器件外部输出忙碌信号/BSYA 和/BSYB。

在每个输入/输出端口 PORT-A 和 PORT-B 中, 设置加上有效指令 ACT 间隔等于存储芯 5040 的操作周期的 2 倍以上(根据定时说明)。如果在同一个输入/输出端口 PORT-A (或 PORT-B) 中有效指令 ACT 间隔小于上面确定的周期, 则取消所加的有效指令 ACT。加到不同的输入/输出端口的有效指令 ACT 的间隔不受限制。

如第 1 实施例那样与跟随用于接收有效指令 ACT 的定时的时钟信号的特定的定时同步地提供读指令 RD 和写指令 WR。存储芯 5040 随着它的操作被自动地充电。在这个实施例中, 例如, 将时钟信号 CLKA 和 CLKB 的周期 t_{CLK} 设置在 10 ns, 将脉冲串长度 BL 设置在 4, 将数据等待时间 DL 设置在 4。数据等待时间 DL 定义从输入读指令 RD 到输出数据的时钟周期的数目。在模寄存器 5012a 和 5012b 中设置脉冲串长度 BL 和数据等待时间 DL。

图 125 表示判优控制电路 5048 的详细情况。

通过将控制电路 5048a 和 5048b 分别加到第 1 实施例的控制电路 5044a 和 5044b 构造判优控制电路 5048。与输入/输出端口 PORT-A 对应的控制电路 5048a 从控制电路 5044a 接收复位信号 RESETA 和反向信号 RVS 以及启动信号/ENA0 和忙碌信号/BSYB, 并输出启动信号/ENA。与输入/输出端口 PORT-B 对应的控制电路 5048b 从控制电路 5044b 接收复位信号 RESETB 和反向信号 RVS 以及启动信号/ENB0 和忙碌信号/BSYA, 并输出启动信号/ENB。在与第 1 实施例的启动信

号/ENA 和/ENB 相同的定时产生启动信号/ENA0 和/ENB0

图 126 表示当加到输入/输出端口 PORT-A 和 PORT-B 的行地址信号相互匹配时实施的判优控制电路 5048 的操作。在这个例子中,时钟信号 CLKA 和 CLKB 的周期是相同的。与时钟信号 CLKA 同步地将有效指令 ACT 加到输入/输出端口 PORT-A。此后立即与时钟信号 CLKB 同步地将有效指令 ACT 加到输入/输出端口 PORT-B。与输入/输出端口 PORT-A 连接的控制器要求写操作,与输入/输出端口 PORT-B 连接的控制器要求读操作。

控制电路 5044a 和 5044b 的操作几乎与上述第 1 实施例的(图 119)相同。控制电路 5044a 与延迟时钟信号 DCLKA 前沿同步地取得低电平的首先到达信号/FSTA,并激活忙碌信号/BSYB(图 126-(a))。因为控制电路 5044b 与延迟时钟信号 DCLKB 前沿同步地取得高电平的首先到达信号/FSTB,不激活忙碌信号/BSYA(图 126-(b))。控制电路 5048a 响应忙碌信号/BSYB 的激活和反向信号 RVS 的低电平,激活启动信号/ENA(图 126-(c))。控制电路 5048b 响应忙碌信号/BSYA 的激活和反向信号 RVS 的低电平,使启动信号/ENB 去激活(图 126-(d))。

与时钟信号 CLKA 和 CLKB 的下一个定时同步地,分别提供写指令 WR 和读指令 RD(图 126-(e))。响应写指令 WR 和读指令 RD,产生反向信号 RVS 的控制电路(图中未画出)激活反向信号 RVS(图 126-(f))。

控制电路 5048a 和 5048b 分别响应反向信号 RVS 的激活,切换启动信号/ENA 和/ENB 的电平(图 126-(g))。然后,首先实施对于输入/输出端口 PORT-B 的读操作(图 126-(h))。在完成读操作后,激活复位信号 RESETB,并使反向信号 RVS 去激活(图 126-(i))。控制电路 5048a 和 5048b 响应反向信号 RVS 的去激活,使启动信号/ENA 和/ENB 的电平回复到它们各自原来的电平(图 126-(j))。然后,响应启动信号/ENA 的激活实施对于输入/输出端口 PORT-A 的读操作(图 126-(k))。

在完成读操作后，激活复位信号 RESETA (图 126- (l))，并使忙碌信号/BSYB 去激活 (图 126- (m))。控制电路 5048a 响应忙碌信号/BSYB 的去激活，使启动信号/ENA 去激活 (图 126- (n))。在本实施例的这个方式中，当行地址信号 RA 相同时和当第 1 个到达的指令要求写操作，接着第 2 个到达的指令要求读操作时，如此控制存储芯 5040 使它首先实施读操作。在存储器 LSI，如具有多端口存储器的 DRAM 中，通过在接收要写的数据后驱动存储芯执行写操作，和通过首先驱动存储芯然后输出数据实施读操作。因此，当在写操作后实施读操作时，全部操作周期通常变成等待时间。在这个实施例中，当写操作和读相互竞争时首先执行读操作，从而缩短全部操作周期和改善传输数据信号的数据总线的使用效率。

下面，我们描述根据第 3 实施例的多端口存储器 M 的操作。

图 127 表示当输入/输出端口 PORT-A 和 PORT-B 接收有效指令 ACT 和相同的行地址信号 RA 时实施读操作的方法。时钟信号 CLKA 的相位稍微超前时钟信号 CLKB 的相位。即，有效指令 ACT 到输入/输出端口 PORT-A 的输入稍微早于有效指令 ACT 进入输入/输出端口 PORT-B。

对于输入/输出端口 PORT-A，响应有效指令 ACT 实施读操作 READ (图 127- (a))。将从存储单元读出的数据存储在数据寄存器 5046a (或 5046b) 中。然后，对于输入/输出端口 PORT-B，响应有效指令 ACT 实施读操作 READ (图 127- (b))。在判优电路 5034 的控制下完成读操作 READA 后对于输入/输出端口 PORT-B 实施读操作 READB (图 127- (c))。将通过读操作 READB 从存储单元读出的数据存储在数据寄存器 5046b (或 5046a) 中 (图 127- (d))。在这个方式中，甚至当将有效指令 ACT 和相同的行地址信号 RA 基本上同时加到输入/输出端口 PORT-A 和 PORT-B 时，也能够对于输入/输出端口 PORT-A 和 PORT-B 中的每一个连续地实施读操作(或写操作)。存储芯 4050 在完成读操作 READA 和 READB 中的每一个后自动地实施预充电操作，从而完成存储周期。

与图所示的第 5 到第 8 个时钟信号 CLKA 同步地输入读指令 RD 后, 连续地输出存储在输入/输出端口 PORT-A 对应的寄存器 5046a 中的检索数据作为输出数据 Q1-Q3 (图 127- (e))。与图所示的第 5 到第 8 个时钟信号 CLKB 同步地输入读指令 RD 后, 连续地输出存储在输入/输出端口 PORT-B 对应的寄存器 5046b 中的检索数据作为输出数据 Q1-Q3 (图 127- (f))。

输入/输出端口 PORT-A 和 PORT-B 两者在第 1 有效指令 ACT 后的 4 时钟脉冲, 接收下一个有效指令 ACT, 进一步分别实施读操作 READA 和 READB (图 127- (g) 和 (h))。当在每 4 个时钟周期中加上有效指令 ACT 一次时, 能够连续地输出检索的数据而没有任何间隙(即, 无间隙读)。而且通过在每 4 个时钟周期中接收有效指令 ACT 一次得到随机存取操作。

图 128 表示当将有效指令 ACT 和相互不同的行地址信号 RA 加到输入/输出端口 PORT-A 和 PORT-B 时实施读操作的方法。

对于首先已经接收有效指令 ACT 和行地址信号 RA 的输入/输出端口 PORT-A, 响应有效指令 ACT 实施读操作 READA (图 128- (a))。将从存储单元读出的数据存储在数据寄存器 5046a 中 (图 128- (b))。然后, 输入/输出端口 PORT-B, 响应有效指令 ACT 实施指向另一个与用于读操作 READA 的存储芯不同的存储芯 5040 的读操作 READB (图 128- (c))。即, 相互独立地实施读操作 READA 和读操作 READB。将通过读操作 READB 从存储单元读出的数据存储在数据寄存器 5046b 中 (图 128- (d))。

与图所示的第 5 到第 8 个时钟信号 CLKA 同步地输入读指令 RD 后, 连续地输出存储在寄存器 5046a 中的检索数据作为输出数据 Q0-Q3 (图 128- (e))。与图所示的第 5 到第 8 个时钟信号 CLKB 同步地输入读指令 RD 后, 连续地输出存储在输入/输出端口 PORT-B 对应的寄存器 5046b 中的检索数据作为输出数据 Q0-Q3 (图 128- (f))。

输入/输出端口 PORT-A 和 PORT-B 两者在第 1 有效指令 ACT 后

的 4 个时钟脉冲，接收下一个有效指令 ACT，进一步分别实施读操作 READA 和 READB（图 128-（g）和（h））。

图 129 表示当输入/输出端口 PORT-A 和 PORT-B 接收有效指令 ACT 和相同的行地址信号 RA 时实施写操作的方法。

在输入/输出端口 PORT-A 和 PORT-B 中，与跟随用于接收有效指令 ACT 的前沿的下面的各时钟信号 CLKA 和 CLKB 的前沿同步地提供写指令 WR，列地址信号 CA 和第 1 写数据 Q0 和 Q0（图 129-（a）和（b））。此后，与各时钟信号 CLKA 和 CLKB 同步地提供写数据 Q1-Q3 和 Q0-Q3（图 129-（c）和（d））。将写数据 Q0-Q3 和 Q0-Q3 分别存储在各数据寄存器 5046a 和 5046b 中（图 129-（e）和（f））。对于首先接收有效指令 ACT 和行地址信号 RA 的输入/输出端口 PORT-A，与取得写数据 Q3 的时钟信号 CLKA 的特定的定时同步地实施写操作 WRITEA（图 129-（g））。在完成写操作 WRITEA 后实施与输入/输出端口 PORT-B 对应的写操作 WRITEB（图 129-（h））。通过写操作 WRITEA 和 WRITEB，将存储在各数据寄存器 5046a 和 5046b 中的写数据 Q0-Q3 和 Q0-Q3 写入与列地址信号 CA 对应的存储单元，从而完成写操作。

在写操作中，在每 4 个时钟周期中提供一组写数据一次，使得写数据能够连续地进入而没有任何间隙（即，无间隙写）。

图 130 表示对于输入/输出端口 PORT-A 连续地实施写操作和读操作，和对于输入/输出端口 PORT-B 相继地实施指向与输入/输出端口 PORT-A 的写操作的行地址信号 RA 相同的行地址信号 RA 的写操作，和指向与输入/输出端口 PORT-A 的读操作的行地址信号 RA 相同的行地址信号 RA 的写操作的情形。第 1 写操作的定时与图 127 相同，并省略对它的说明。

在输入/输出端口 PORT-B，在与图 127 相同的定时提供与第 2 写操作对应的有效指令 ACT（图 130-（a））。因为不将指令信号加到输入/输出端口 PORT-A，所以在取得写数据 Q0-Q3 后立即实施写操作 WRITEB（图 130-（b））。

在输入/输出端口 PORT-A, 与图所示的第 7 个时钟信号 CLKA 同步地提供下一个有效指令 ACT (图 130- (c))。虽然在图中未画出, 但是在这个特定的瞬间激活对于输入/输出端口 PORT-B 的启动信号/ENB。结果, 在完成写操作 WRITEB 后实施读操作 READA (图 130- (d))。因为多端口存储器 M 以接收各指令的次序执行写操作 WRITEB 和读操作 READA, 所以在完成写操作前被读的存储单元的数据不变。

此外, 因为输入/输出端口 PORT-A 能够输出存储在与输入/输出端口 PORT-B 对应的数据寄存器 5046b 中数据作为检索数据, 所以可以在输入/输出端口 PORT-B 的写操作 WRITEB 前实施输入/输出端口 PORT-A 的读操作 READA。

图 131 表示对于输入/输出端口 PORT-A 连续地实施写操作和读操作, 和对于输入/输出端口 PORT-B 相继地实施指向与输入/输出端口 PORT-A 的写操作的行地址信号 RA 相同的行地址信号 RA 的读操作和指向与输入/输出端口 PORT-A 的读操作的行地址信号 RA 相同的行地址信号 RA 的写操作的情形。对于输入/输出端口 PORT-A 的第 1 写操作的定时和对于输入/输出端口 PORT-B 的第 1 读操作的定时分别与图 129 的写操作和图 128 的读操作相同。

在输入/输出端口 PORT-A, 与图所示的第 7 和第 8 时钟信号 CLKA 同步地提供有效指令 ACT 和读指令 (图 131- (a))。因为在这个特定的瞬间不将有效指令 ACT 加到输入/输出端口 PORT-B, 所以实施对于输入/输出端口 PORT-A 的读操作 READA (图 131- (b))。

下面, 在输入/输出端口 PORT-B, 与图所示的第 8 和第 9 时钟信号 CLKB 同步地提供有效指令 ACT 和写指令 WR (图 131- (c))。在接收数据 Q0-Q3 后, 实施对于输入/输出端口 PORT-B 的写操作 (图中未画出)。

图 132 表示在时钟信号 CLKA 和 CLKB 具有不同的时钟周期的情形中当加到输入/输出端口 PORT-A 和 PORT-B 的行地址信号相互匹配时实施的操作。在这个例子中, 时钟信号 CLKB 的周期长度等于

时钟信号 CLKA 的周期长度的两倍。

在输入/输出端口 PORT-A, 当在每 4 个时钟周期中加上一组有效指令 ACT 和读指令 RD 一次, 以与图 127 相同的方式实施读操作。在输入/输出端口 PORT-B, 也当在每 4 个时钟周期中加上一组有效指令 ACT 和读指令 RD 一次。向输入/输出端口 PORT-B 输入第 1 有效指令 ACT 的时间比向输入/输出端口 PORT-A 输入第 1 有效指令 ACT 的时间晚一些 (图 132- (a))。因此, 与图 127 的情形相同在读操作 READA 后实施读操作 READB (图 132- (b))。在两个读操作 READA 之间执行与输入/输出端口 PORT-B 的跟随的有效指令 ACT 对应的下一个读操作 READB (图 132- (c))。

本实施例能够提供与上述的第 1 实施例相同的优点。此外, 在每个输入/输出端口 PORT-A 和 PORT-B 中, 本实施例用等于存储芯 5040 的操作周期的 2 倍以上的有效指令 ACT 的间隔 (按照定时说明)。因此, 甚至当加到输入/输出端口 PORT-A 和 PORT-B 的行地址信号 RA 相同, 也肯定能够对于每个端口实施读操作和写操作。因此, 控制多端口存储器 M 的控制器不需要检测多端口存储器 M 的忙碌状态。这样就简化了控制器的控制 (借助硬件和软件)。

图 133 表示多端口存储器的第 4 实施例和根据本发明 (第 5 方面) 控制多端口存储器的方法。用相同的数字标记与第 1 和第 3 实施例相同的部件, 并省略对它们的详细描述。

在这个实施例中, 提供页面缓冲器 5050a 和 5050b 代替上述第 3 实施例的数据寄存器 5046a 和 5046b。页面缓冲器 5050a 和 5050b 与输入/输出端口 PORT-A 和 PORT-B 中的至少一个结合起来进行操作。其它配置几乎与第 3 实施例完全相同。

每个页面缓冲器 5050a 和 5050b 都包括锁存器, 其中存储在存储芯 5040 中的所有存储单元的数据。在开始读操作和写操作时, 将存储在选出的存储芯 5040 的存储单元中的数据读出到页面缓冲器 5050a (或 5050b)。在读操作中, 响应列地址信号 CA 输出锁存在页面缓冲器 5050a 中的数据作为数据信号。在写操作中, 首先根据列地址信

号 CA 将数据信号写入页面缓冲器 5050a。此后，在完成写操作时将页面缓冲器 5050a 的数据写入存储单元。

下面，我们描述第 4 实施例多端口存储器 M 的操作。

图 134 表示当输入/输出端口 PORT-A 和 PORT-B 接收有效指令 ACT 和相同的行地址信号 RA 时实施读操作的方法。时钟信号 CLKA 的相位稍微超前时钟信号 CLKB 的相位。即，输入到输入/输出 PORT-A 的有效指令 ACT 比输入到输入/输出 PORT-B 的有效指令 ACT 稍微早一些。

在输入/输出端口 PORT-A，响应有效指令 ACT 实施读操作 READA（图 134-（a））。从由读操作 READA 选出的存储芯 5040 的所有存储单元读出数据，并将检索数据存储在页面缓冲器 5050a（或 5050b）中的一个（图 134-（b））。另一方面，在输入/输出端口 PORT-B，行地址信号 RA 与加到输入/输出端口 PORT-A 的相同，所以不实施与有效指令 ACT 相应的读操作。

在输入/输出端口 PORT-A，与如图所示的第 1 和第 5 时钟信号 CLKA 同步地加上读指令 RD（图 134-（c）和（d））。在接收各读指令 RD 后与第 5 到第 12 时钟信号 CLKA 同步地连续输出存储在页面缓冲器 5050a 中的数据作为输出数据 Q0-Q7（图 134-（e））。即，实施页面读操作。

由于同样的原因，在输入/输出端口 PORT-B，与如图所示的第 1 和第 5 时钟信号 CLKB 同步地加上读指令 RD（图 134-（f）和（g））。在接收各读指令 RD 后与第 5 到第 12 时钟信号 CLKB 同步地连续输出存储在页面缓冲器 5050a 中的数据作为输出数据 Q0-Q7（图 134-（h））。在这个方式中，如果行地址信号 RA 是相同的，则输入/输出端口 PORT-A 和 PORT-B 共用一个页面缓冲器 5050a（或 5050b）。

输入/输出端口 PORT-A 和 PORT-B 两者都在第 1 有效指令 ACT 后的 8 个时钟周期，接收下一个有效指令 ACT（图 134-（i）和（j））。因为行地址信号 RA 是相同的，所以只实施读操作 READA（图 134-（k））。不实施对于输入/输出端口 PORT-B 的读操作 READB。通

过在每 4 个时钟周期中加上读指令 RD 一次能够连续地输出读数据而没有任何间隙（即，无间隙读）。

图 135 表示当将有效指令 ACT 和不同的行地址信号 RA 加到输入/输出端口 PORT-A 和 PORT-B 时实施读操作的方法。与输入/输出端口 PORT-A 对应的读操作的定时和图 134 相同。

在首先接收有效指令 ACT 和行地址信号 RA 的输入/输出端口 PORT-A，响应有效指令 ACT 实施读操作 READA（图 135-（a））。将从存储芯 5040 的所有存储单元读出的数据存储在各页面缓冲器 5050a 中（图 135-（b））。在输入/输出端口 PORT-B，响应有效指令 ACT 对于与读操作 READA 的不同的存储芯 5040 实施读操作 READB（图 135-（c））。即，将由读操作 READB 从存储芯 5040 的所有存储单元读出的数据存储在各页面缓冲器 5050b 中（图 135-（d））。此后，以与结合图 134 描述的相同方式实施读操作。在这个方式中，当行地址信号 RA 相互不同时，独立的实施读操作 READA 和读操作 READB，分别将检索数据存储在各页面缓冲器 5050a 和 5050b 中。

图 136 表示将有效指令 ACT 和相同的行地址信号 RA 加到输入/输出端口 PORT-A 和 PORT-B，并实施写操作，接着加上有效指令 ACT 和不同的行地址信号 RA，导致实施写操作的情形。

在输入/输出端口 PORT-A 和 PORT-B，与时钟信号 CLKA 和 CLKB 的各前沿同步地加上有效指令 ACT 和相同的行地址信号 RA。图 133 所示的判优电路 5034 认定输入/输出端口 PORT-A 首先接收有效指令 ACT，并实施读操作 READA（图 136-（a）），以便将数据从存储单元传输到页面缓冲器 5050a（或 5050b）。

从由读操作 READA 选出的存储芯 5040 的所有存储单元读出数据，并存储在页面缓冲器 5050a（或 5050b）中（图 136-（b））。另一方面，在输入/输出端口 PORT-B，因为行地址信号 RA 与加到输入/输出端口 PORT-A 的相同，所以响应有效指令 ACT 不实施读操作。

此后，在输入/输出端口 PORT-A，与如图所示的第 1 和第 5 时钟信号 CLKA 同步地加上写指令 WD 和列地址信号 CA（图 136-（c）

和 (d))。将与钟信号 CLKA 同步地连续加上的写数据 Q0-Q7 写入页面缓冲器 5050a 中 (图 134- (e))。即, 实施页面写操作。

在输入/输出端口 PORT-B, 与如图所示的第 1 和第 5 时钟信号 CLKB 同步地加上写指令 WR 和列地址信号 CA (图 136- (f) 和 (g))。将与时钟信号 CLKB 同步地一个接一个地加上的写数据 Q0-Q7 写入共用的列页面缓冲器 5050a 中 (图 134- (h))。在这个方式中, 如果行地址信号 RA 相同, 则在写操作中输入/输出端口 PORT-A 和 PORT-B 共用同一个页面缓冲器 5050a (或 5050b)。

在首先接收有效指令 ACT 的输入/输出端口 PORT-A, 与取得写数据 Q7 的时钟信号 CLKA 的特定定时同步地实施写操作 WRITEA (图 136- (i))。在完成写操作 WRITEA 后实施与输入/输出端口 PORT-B 对应的写操作 WRITEB (图 136- (j))。

此后, 在输入/输出端口 PORT-A 和 PORT-B, 与时钟信号 CLKA 和 CLKB 的各前沿同步地加上有效指令 ACT 和相互不同的行地址信号 RA。图 133 所示的判优电路 5034 认定首先将有效指令 ACT 加到输入/输出端口 PORT-A, 并一个接着一个地实施读操作 READA 和 READB (图 136- (k) 和 (l))。

从由读操作 READA 选出的存储芯 5040 的所有存储单元读出数据, 并存储在页面缓冲器 5050a (或 5050b) 中 (图 136- (m))。进一步, 从由读操作 READB 选出的存储芯 5040 的所有存储单元读出数据, 并存储在另一个页面缓冲器 5050b (或 5050a) 中 (图 136- (n))。

在输入/输出端口 PORT-A, 与如图所示的第 13 和第 17 时钟信号 CLKA 同步地加上读指令 RD 和列地址信号 CA (图 136- (o) 和 (p))。将与钟信号 CLKA 同步地一个接着一个加上的写数据 Q0-Q7 存储在页面缓冲器 5050a 中 (图 136- (q))。

类似地, 在输入/输出端口 PORT-B, 与如图所示的第 13 和第 17 时钟信号 CLKB 同步地加上写指令 WR 和列地址信号 CA (图 136- (r) 和 (s))。将与钟信号 CLKB 同步地一个接着一个加上的写数据 Q0-Q7 写入页面缓冲器 5050b 中 (图 136- (t))。在这个方式中, 当行地址

信号 RA 不同时用页面缓冲器 5050a 和 5050b。

在首先接收有效指令 ACT 和行地址信号 RA 的输入/输出端口 PORT-A, 与取得写数据 Q7 的时钟信号 CLKA 的特定定时同步地实施写操作 WRITEA (图 136- (u))。在完成写操作 WRITEA 后实施与输入/输出端口 PORT-B 对应的写操作 WRITEB (图 136- (v))。通过写操作 WRITEA 和 WRITEB, 分别将存储器在页面缓冲器 5050a 和 5050b 中的写数据 Q0-Q7 写入与列地址信号 CA 对应的存储单元, 从而完成写操作。

图 137 表示将有效指令 ACT 和相同的行地址信号 RA 加到输入/输出端口 PORT-A 和 PORT-B, 实施写操作, 接着加上有效指令 ACT 和相同的行地址信号 RA, 导致在输入/输出端口 PORT-A 实施读操作和在输入/输出端口 PORT-B 实施写操作的情形。第 1 写操作的定时与图 137 相同, 我们将省略对它的描述。

在输入/输出端口 PORT-A 和 PORT-B, 与图所示的第 12 时钟信号 CLKA 和 CLKB 的各前沿同步地提供有效指令 ACT 和相同的行地址信号 RA (图 137- (a) 和 (b))。图 133 所示的判优电路 5034 认定首先将有效指令 ACT 加到输入/输出端口 PORT-A, 并实施读操作 READA (图 137- (c))。从由读操作 READA 选出的存储芯 5040 的所有存储单元读出数据, 并存储在页面缓冲器 5050a (或 5050b) 中 (图 137- (d))。在输入/输出端口 PORT-B, 因为行地址信号 RA 与加到输入/输出端口 PORT-A 的信号相同, 所以不实施与有效指令 ACT 对应的写操作。

此后, 在输入/输出端口 PORT-A, 与如图所示的第 13 和第 17 时钟信号 CLKA 同步地加上读指令 RD (图 137- (e) 和 (f))。在接收各读指令 RD 后与如图所示的第 17 到第 24 时钟信号 CLKA 同步地连续输出存储在页面缓冲器 5050a 中的数据 (图 137- (g))。

在输入/输出端口 PORT-B, 与如图所示的第 13 和第 17 时钟信号 CLKB 同步地加上写指令 WR (图 137- (h) 和 (i))。将与钟信号 CLKB 同步地连续加上的写数据 Q0-Q7 存储在共用的页面缓冲器

5050a 中 (图 137- (j))。

此后, 在输入/输出端口 PORT-B, 与取得写数据 Q7 的时钟信号 CLKB 的特定定时同步地实施写操作 WRITEB (图 137- (k))。

图 138 表示将有效指令 ACT 和相同的行地址信号 RA 加到输入/输出端口 PORT-A 和 PORT-B, 实施写操作和读操作, 接着加上有效指令 ACT 和不同的行地址信号 RA, 导致实施写操作和读操作的情形。

在输入/输出端口 PORT-A 和 PORT-B, 与时钟信号 CLKA 和 CLKB 的前沿同步地提供有效指令 ACT 和相同的行地址信号 RA (图 138- (a) 和 (b))。判优电路 5034 确定首先将有效指令 ACT 加到输入/输出端口 PORT-A, 并实施读操作 READA (图 138- (c))。从读操作 READA 选出的存储芯 5040 的所有存储单元读出数据, 并将读出数据存储于页面缓冲器 5050a (或 5050b) 中 (图 137- (d))。另一方面, 在输入/输出端口 PORT-B, 行地址信号 RA 与加到输入/输出端口 PORT-A 的那些相同, 所以不实施与有效指令 ACT 对应的读操作。

此后, 在输入/输出端口 PORT-A, 与第 1 和第 5 时钟信号 CLKA 同步地加上写指令 WR (图 138- (e) 和 (f))。将与时钟信号 CLKA 同步地连续加上的写数据 Q0-Q7 存储在页面缓冲器 5050a 中 (图 137- (g))。

在输入/输出端口 PORT-B, 与第 1 和第 5 时钟信号 CLKB 同步地加上读指令 RD (图 138- (h) 和 (i))。在接收各读指令 RD 后与第 5 到第 12 时钟信号 CLKB 的定时同步地一个接着一个地输出存储在页面缓冲器 5050a 中的数据作为输出数据 Q0-Q7 (图 138- (j))。在输入/输出端口 PORT-A, 与取得写数据 Q7 的时钟信号 CLKA 的特定定时同步地实施写操作 WRITEA (图 138- (k))。

然后, 在输入/输出端口 PORT-A 和 PORT-B, 与时钟信号 CLKA 和 CLKB 的前沿同步地提供有效指令 ACT 和相互不同的行地址信号 RA (图 138- (l) 和 (m))。判优电路 5034 认定首先将有效指令 ACT 加到输入/输出端口 PORT-A, 并连续实施读操作 READA 和 READB

(图 138-(n) 和 (o))。从由读操作 READA 选出的存储芯 5040 的所有存储单元读出数据, 并将读出的数据存储在页面缓冲器 5050a (或 5050b) 中的一个(图 138-(p))。进一步, 从由读操作 READB 选出的存储芯 5040 的所有存储单元读出数据, 并将读出的数据存储在页面缓冲器 5050b (或 5050a) 中的另一个(图 138-(q))。

在输入/输出端口 PORT-A, 与时钟信号 CLKA 的第 13 和第 17 定时同步地加上写指令 WR (图 138-(r) 和 (s))。将与时钟信号 CLKA 同步地一个接着一个加上的写数据 Q0-Q7 写入页面缓冲器 5050a 中(图 138-(t))。

类似地, 在输入/输出端口 PORT-B, 与时钟信号 CLKA 的第 13 和第 17 定时同步地加上写指令 WR (图 138-(u) 和 (v))。将与时钟信号 CLKB 同步地一个接着一个加上的写数据 Q0-Q7 写入页面缓冲器 5050b 中(图 138-(w))。

本实施例能够提供与上述的第 3 实施例相同的优点。进一步, 在本实施例中, 用作对于存储芯 5040 的所有存储单元的暂时数据存储的页面缓冲器 5050a 和 5050b 位于数据锁存器 5028 和存储芯 5040 之间。这使多端口存储器 M 能够实施页面读操作和页面写操作。

当将相同的行地址信号 RA 加到输入/输出端口 PORT-A 和 PORT-B 时, 共用同一个页面缓冲器 5050a。这防止写入存储单元的数据通过覆盖操作被破坏。

当将相同的行地址信号 RA 加到输入/输出端口 PORT-A 和 PORT-B 时, 响应一个端口只实施读操作。因此, 与对于两个端口实施的各读操作的情形比较能够减少操作时的功率消耗。使用页面缓冲器 5050a 和 5050b 甚至当实施页面操作时也消除了对于控制多端口存储器 M, 检测多端口存储器 M 的忙碌状态的控制器的需要。因此, 控制器等的控制(借助硬件和软件)变得较容易了。

图 139 表示根据多端口存储器的第 5 实施例的多端口存储器的操作和控制本发明的多端口存储器的方法。用相同的数字标记与第 4 实施例相同的部件, 并省略对它们的详细描述。

这个实施例具有用于通常的脉冲串操作的读指令 RD 和写指令 WR 两者与用于页面操作的读指令 PRD 和写指令 PWR 两者。多端口存储器 M 的电路配置基本上与第 4 实施例相同。

在图 139 中, 将有效指令 ACT 和相同的行地址信号 RA 加到输入/输出端口 PORT-A 和 PORT-B (图 139-(a) 和 (b))。与时钟信号 CLKA 和 CLKB 的下一个周期同步地, 加上读指令 PRD (图 139-(c) 和 (d)), 实施页面读操作 (图 139-(e))。页面读操作的定时与图 134 相同, 并省略对它的详细描述。

此后, 将有效指令 ACT 和相同的行地址信号 RA 加到输入/输出端口 PORT-A 和 PORT-B (图 139-(f) 和 (g))。与时钟信号 CLKA 和 CLKB 的下一个周期同步地, 加上读指令 RD (图 139-(h) 和 (i))。对于各输入/输出端口 PORT-A 和 PORT-B 连续实施读操作 READA 和 READB (图 139-(j) 和 (k))。即完成通常的读操作 (即脉冲串读操作)。

这个本实施例能够提供与上述的第 4 实施例相同的优点。因为这个实施例准备了用于页面操作的读指令 PRD 和 PWR 以及用于通常操作的读指令 RD 和 WR, 所以多端口存储器 M 响应所加的指令信号不仅能够实施页面操作而且能够实施通常操作。

上述实施例已经指向一个将本发明用于多路复用地址信号的多路复用型的多端口存储器的例子。但是本发明不限于这些特定的实施例。例如, 也可以将本发明用于同时接收地址信号的非多路复用型的多端口存储器。

上述实施例已经指向一个将本发明用于具有两个输入/输出端口 PORT-A 和 PORT-B 的多端口存储器 M 的例子。但是本发明不限于这些实施例。例如, 也可以将本发明用于具有 4 个输入/输出端口的多端口存储器。在这个情形中, 将所加的有效指令 ACT 的间隔 (按照定时说明) 设置得等于或大于存储芯的操作周期的 4 倍。

在上述实施例中, 对将本发明用于具有同步 DRAM 存储芯的多端口存储器的例子进行了描述。但是本发明不限于这种形式的实施例。

例如，也可以将本发明用于具有同步 SRAM 存储芯的多端口存储器。

进一步，在上述的多端口存储器中，可以将对于存储芯操作的要求作为指令信号输入。将这样的指令信号与时钟信号同步地加到一个输入/输出端口的指令端子。可以将该指令信号分成一个用于激活存储块的一个特定存储区域的有效指令和一个指示在这个存储区域中或者实施读操作或者实施写操作的动作指令，并且可以连续地加上这些指令。由于同样的原因，也可以在时分基础上一个接着一个地加上地址信号。通过在加上有效指令后的预定时钟周期上加上动作指令将读操作周期和写操作周期固定在恒定的周期上。

如果存储块的存储单元由 DRAM 单元构成则需要刷新操作。对于由加在任何一个输入/输出端口上的地址信号指示的刷新地址实施刷新操作。这个配置能够使在多端口存储器中的控制电路的尺寸减到最小，从而能够减小芯片尺寸。

在读操作和写操作后自动地实施将与存储单元连接的位线复位到预定电位的预充电操作。这使从开始各操作的预定时间周期内完成读操作和写操作成为可能。即，能够将读周期时间和写周期时间固定为恒定的。

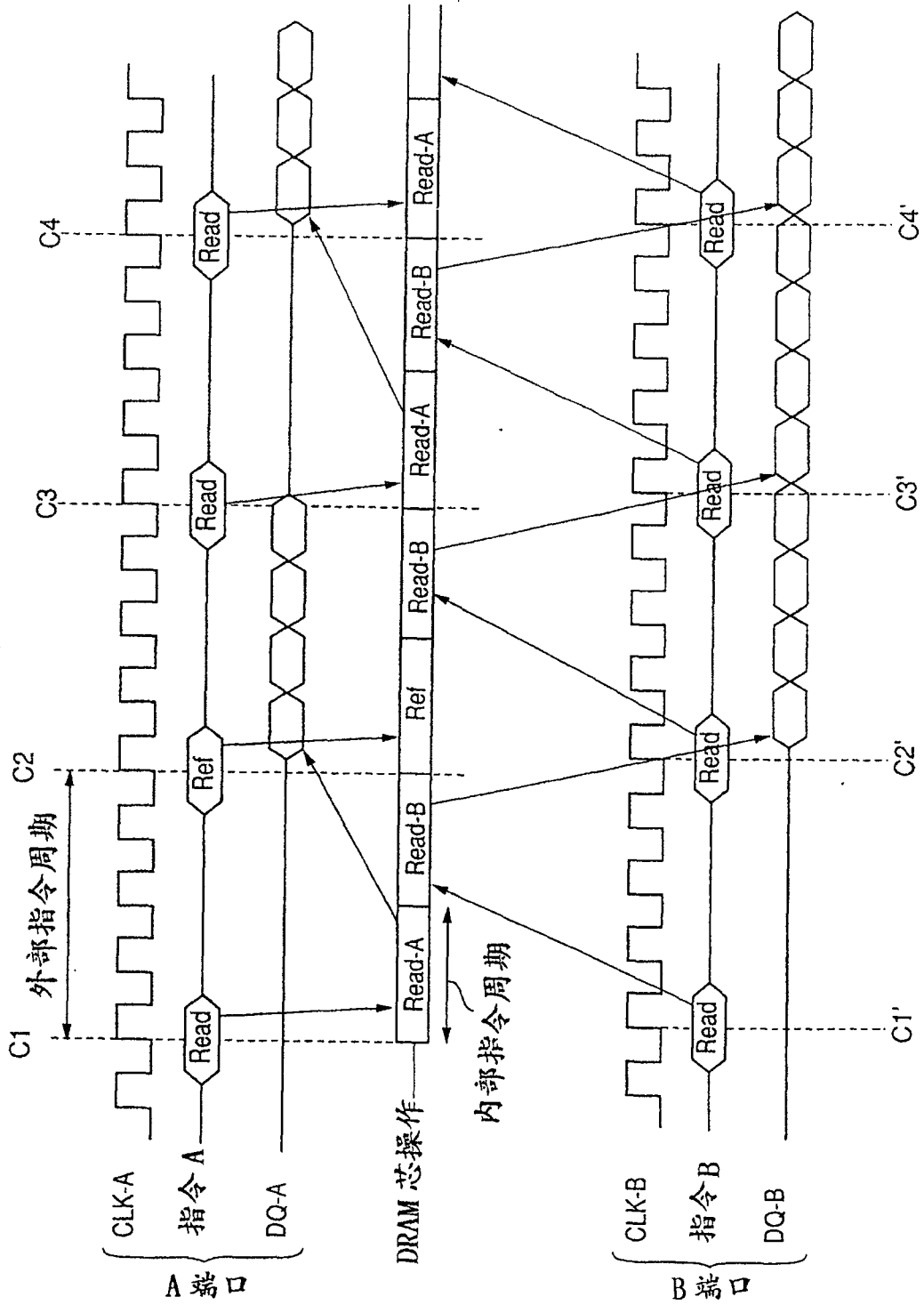
而且，可以为每个输入/输出端口提供忙碌端子以便输出忙碌信号。当加到一个输入/输出端口的地址信号与加到另一个输入/输出端口的地址信号相同时和当对于后一个输入/输出端口执行存储操作时，输出忙碌信号。用这种配置，与多端口存储器连接的控制器很容易知道还没有实施所要求的操作。

进一步，本发明不限于这些实施例，而且可以作出不同的变化和修改而没有偏离本发明的范围。

例如，已经参考为了同步只用一个前沿或一个后沿的配置描述了本发明的第 1 到第 5 方面。然而对于本领域技术人员来说显然能够容易地改变上述任何一个配置，使它与为了同步用前沿和后沿两者的 DDR（双数据速率）操作匹配。我们有意使这样一个明显的改变处在本发明的范围内。

本申请基于向日本专利局提出的日本优先权专利申请 2000 年 12 月 20 日递交的 No.2000-387891, 2001 年 2 月 9 日递交的 No.2001-034361, 2001 年 2 月 14 日递交的 No.2000-037547, 2000 年 12 月 27 日递交的 No.2000-398893 和 2000 年 12 月 27 日递交的 No.2000-399052, 这里我们将这些专利申请的全部内容作为参考。

图 1



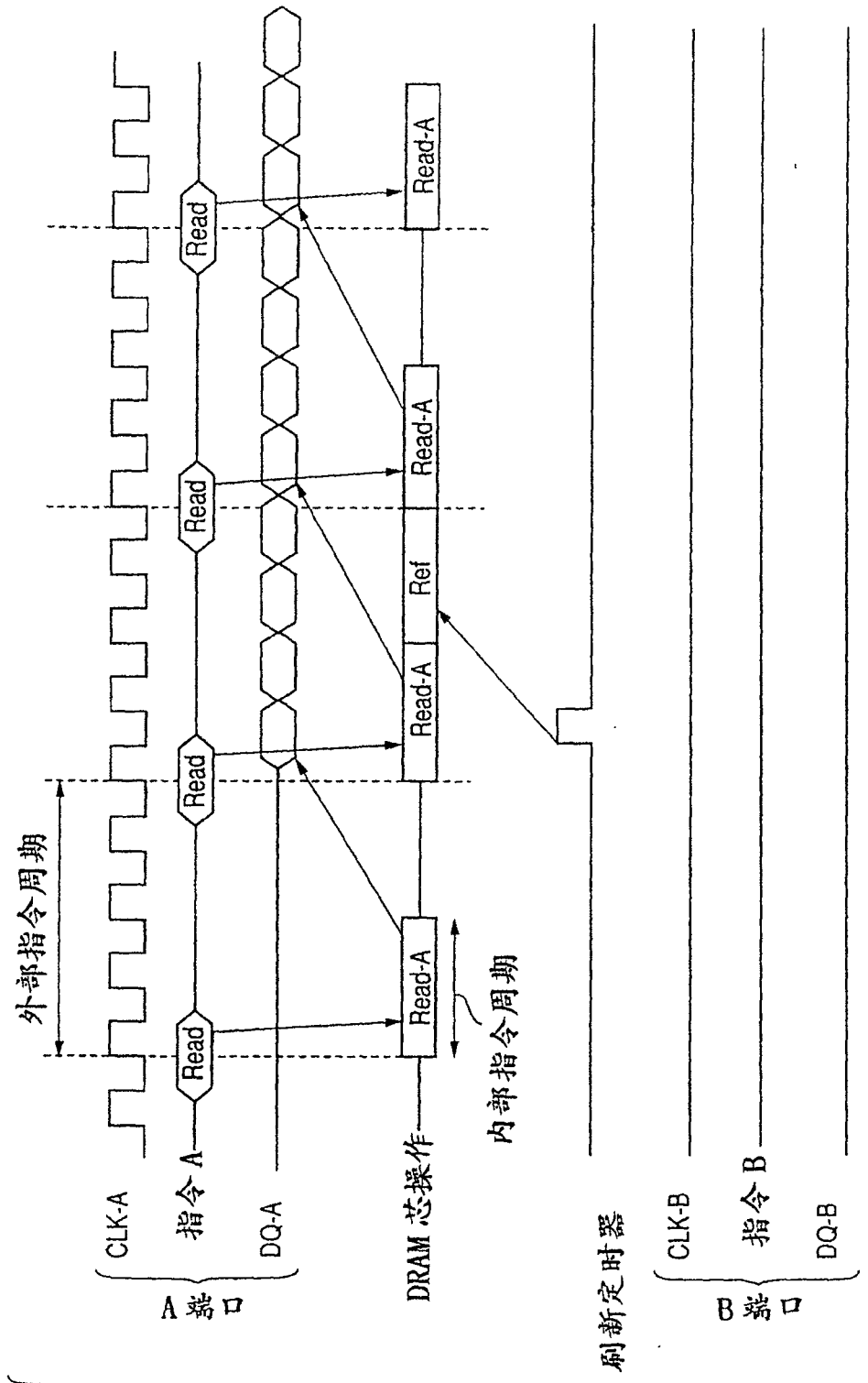
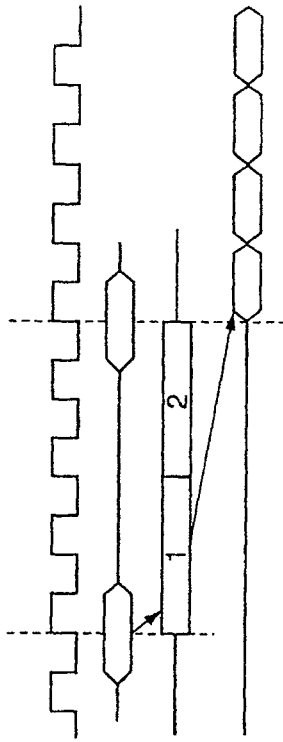
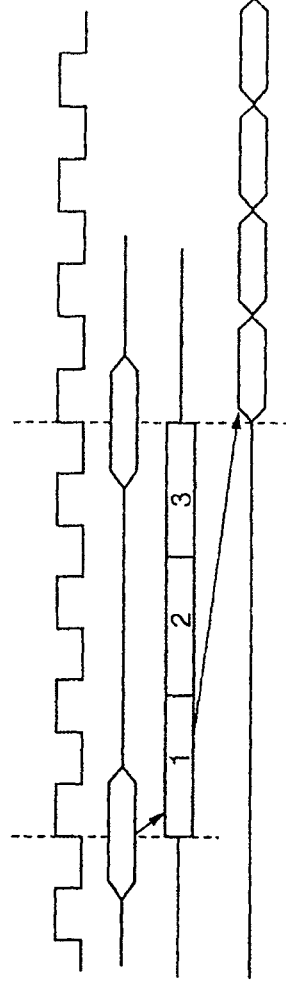


图 2



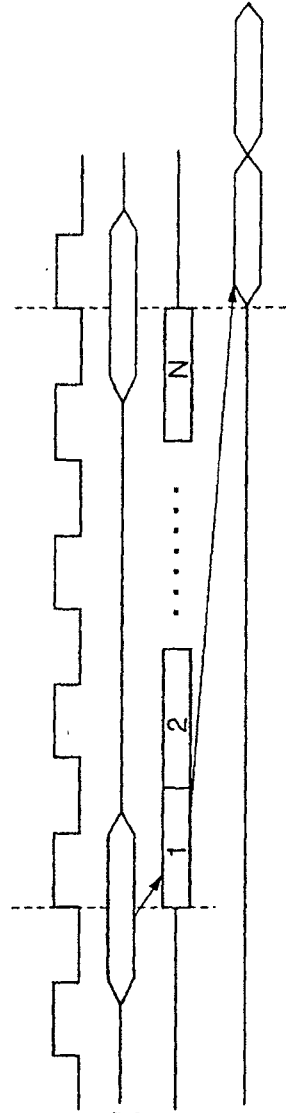
一个端口的时钟信号
 一个端口的最小指令周期
 DRAM芯操作周期
 一个端口的DQ

图 3A



一个端口的时钟信号
 一个端口的最小指令周期
 DRAM芯操作周期
 一个端口的DQ

图 3B



一个端口的时钟信号
 一个端口的最小指令周期
 DRAM芯操作周期
 一个端口的DQ

图 3C

图 4

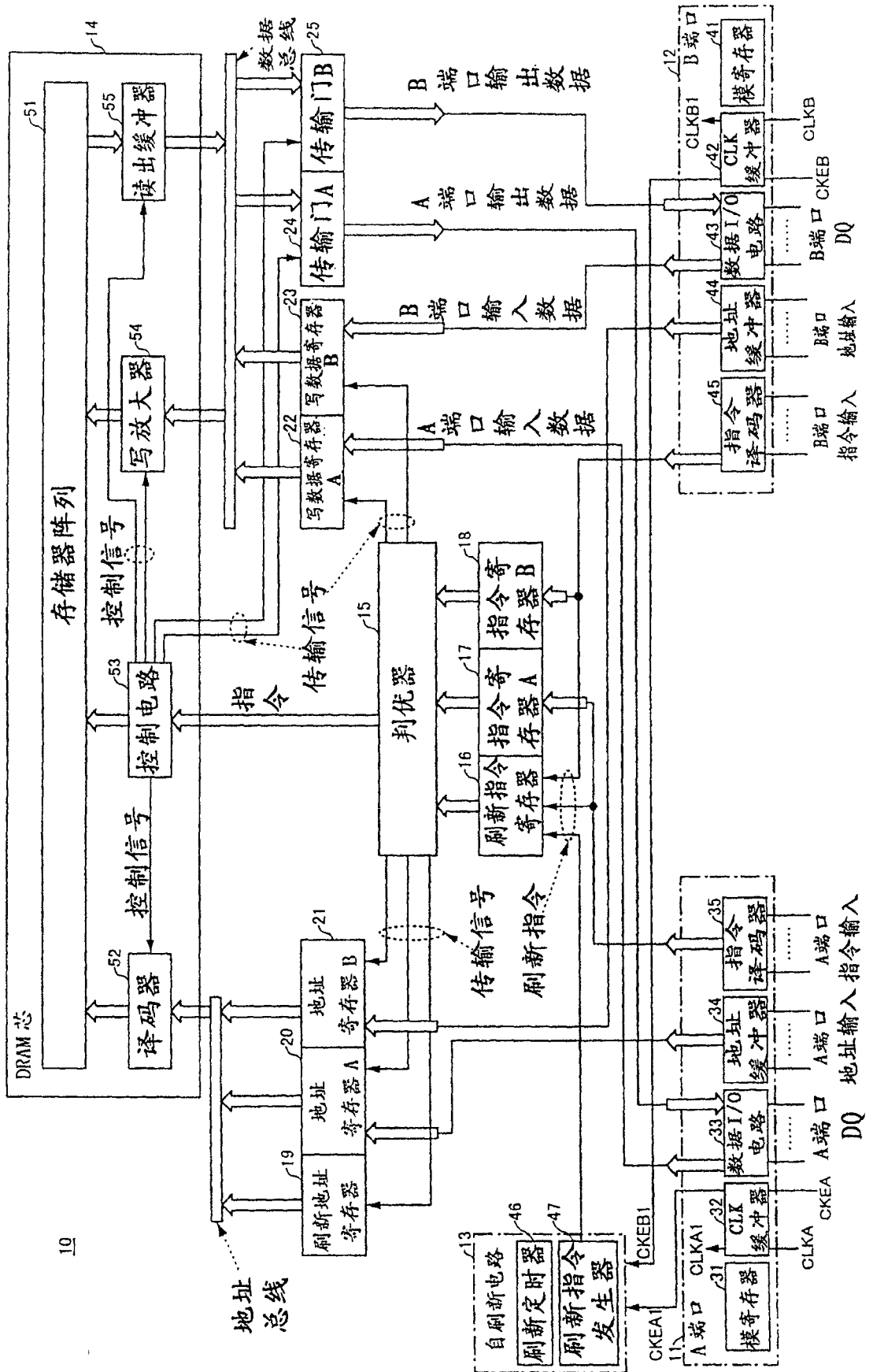


图 5

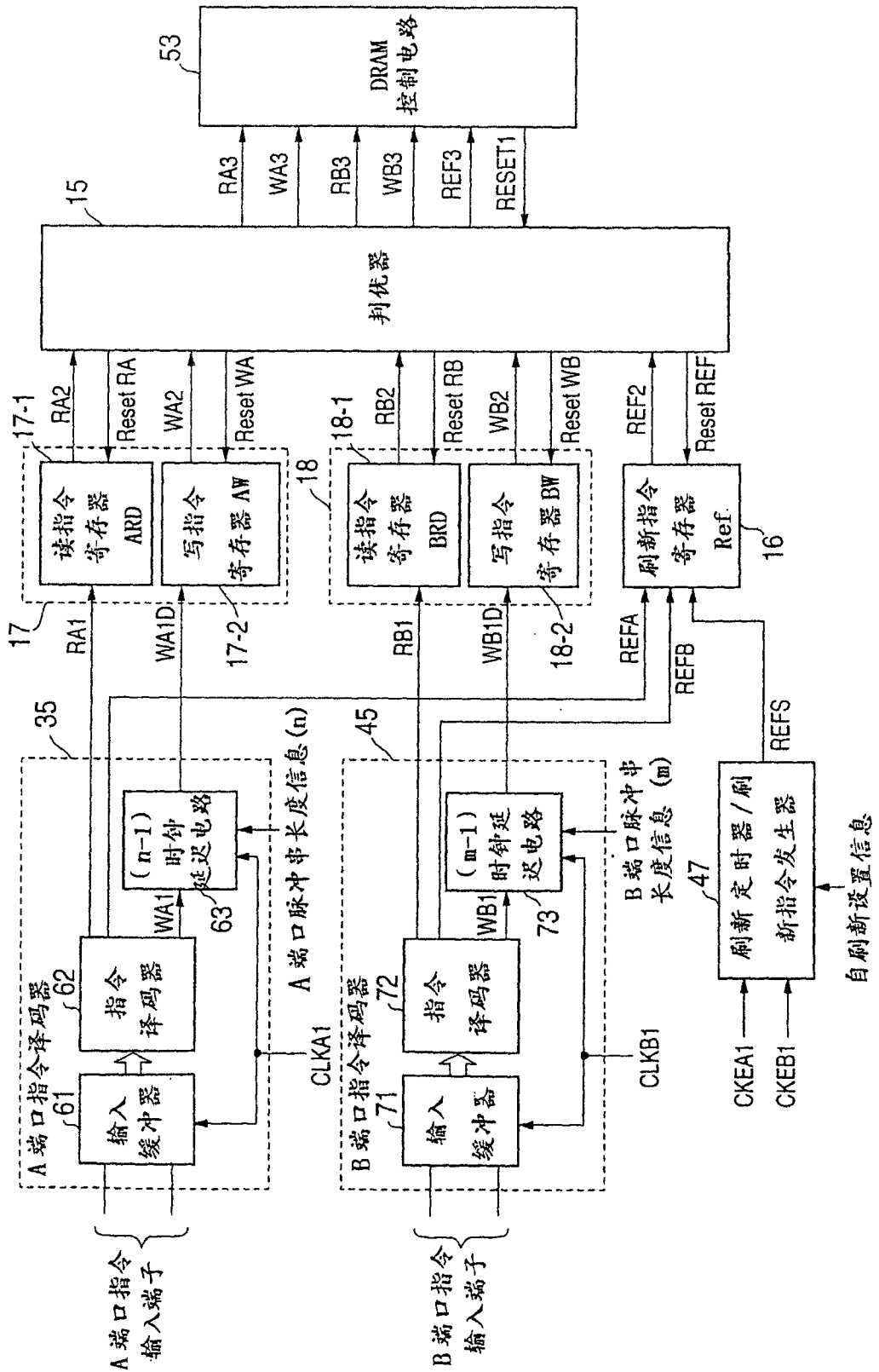


图 6A

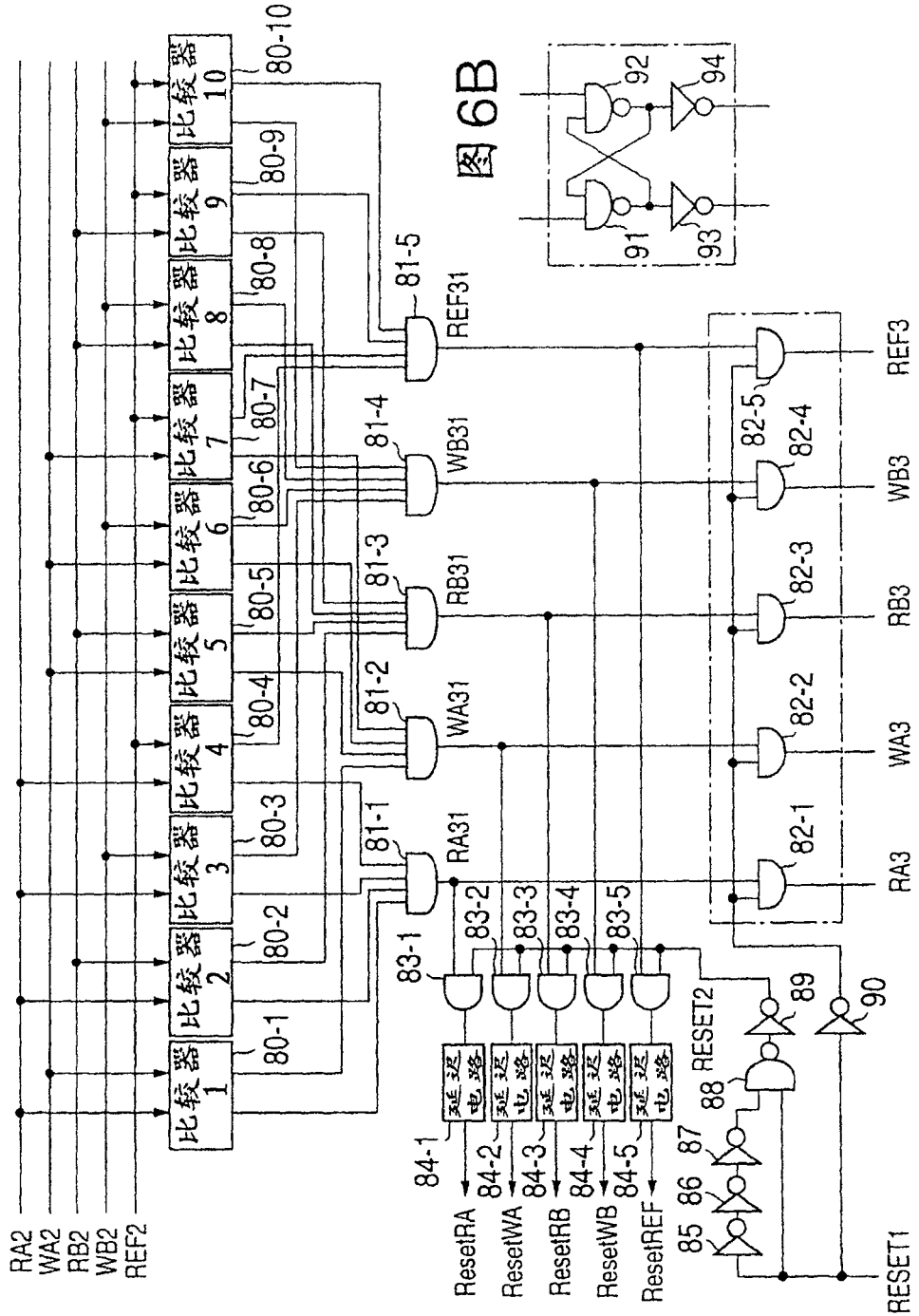
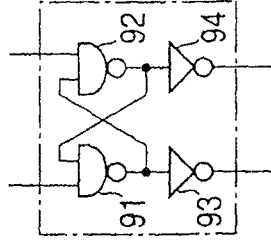


图 6B



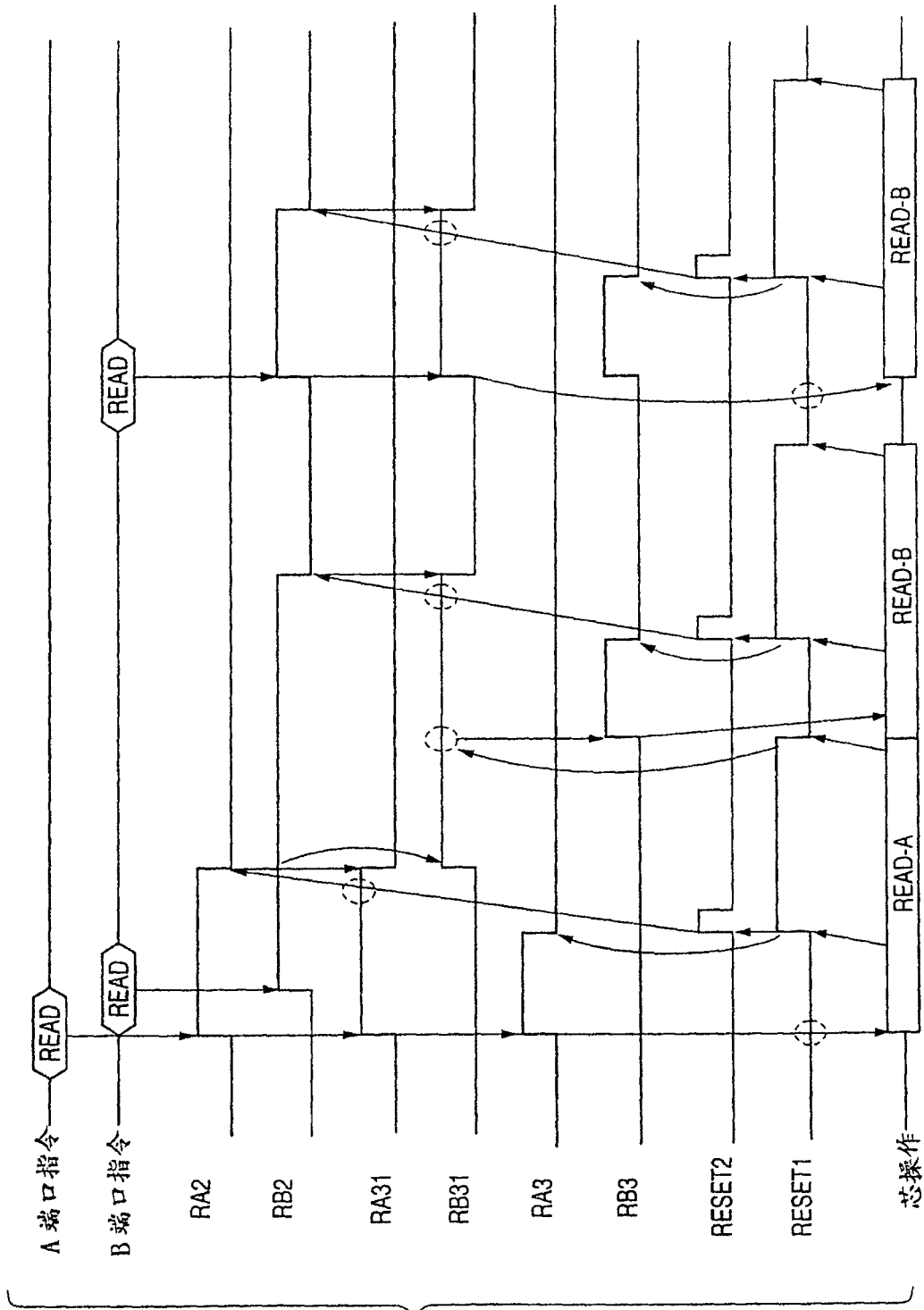


图 7

图 8

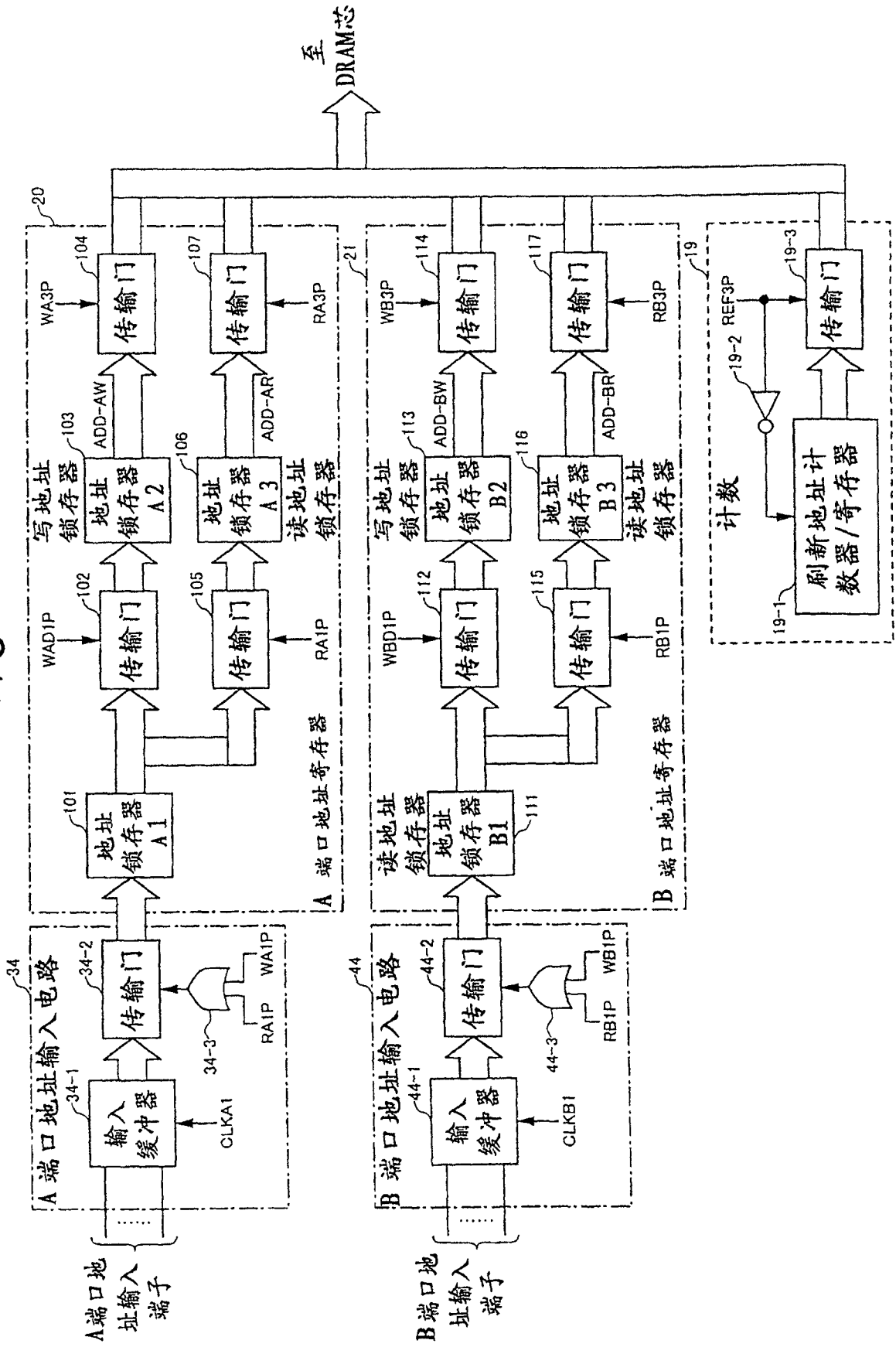


图 9

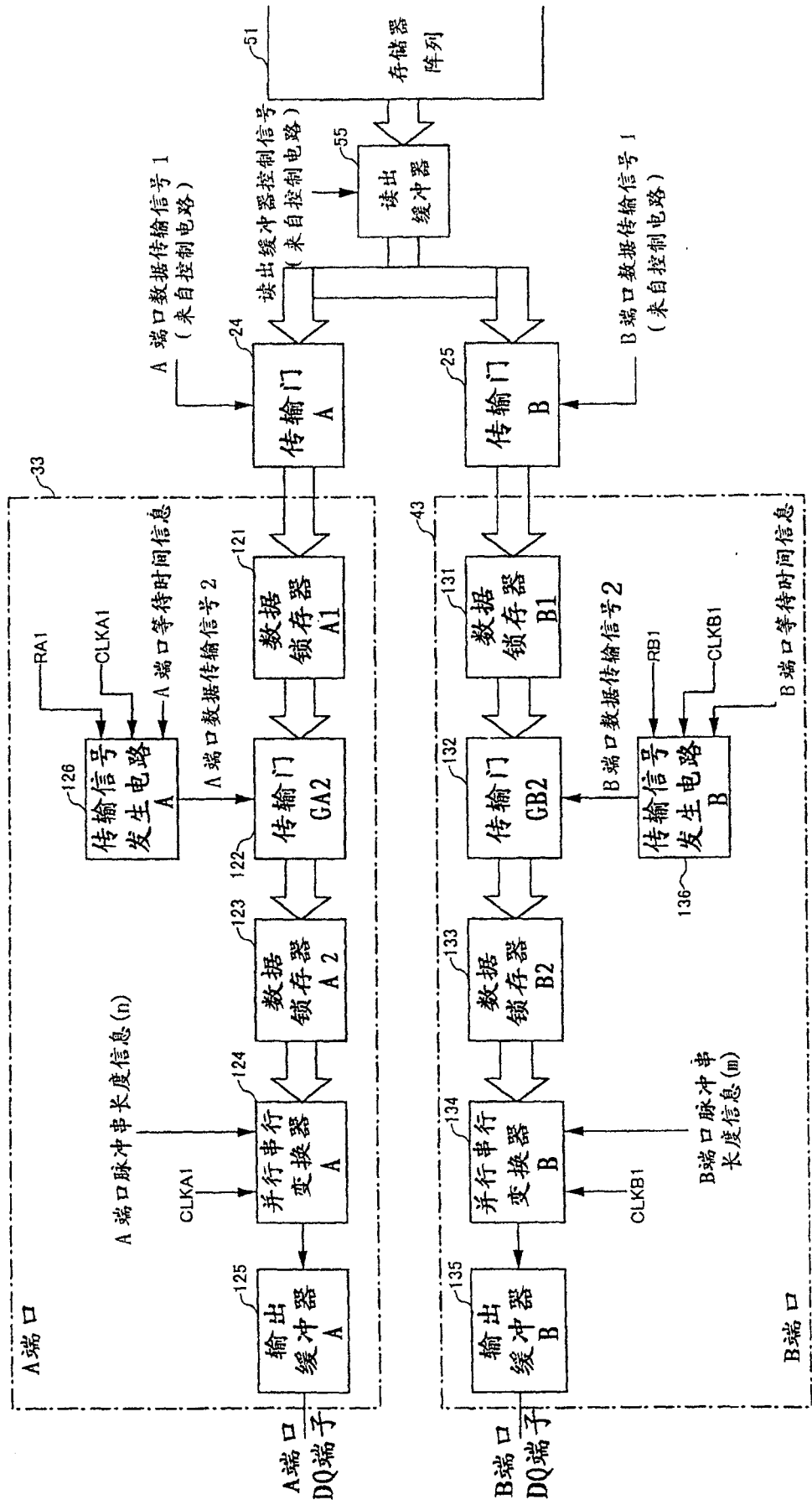


图10

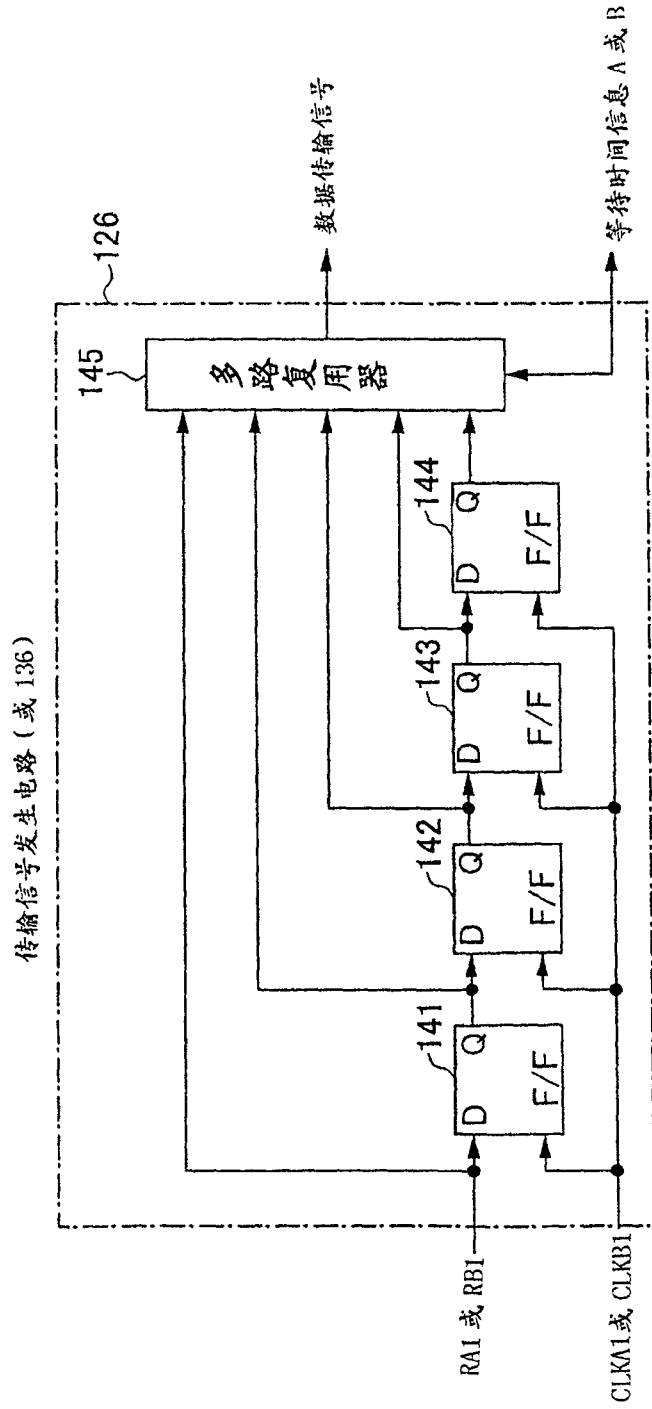
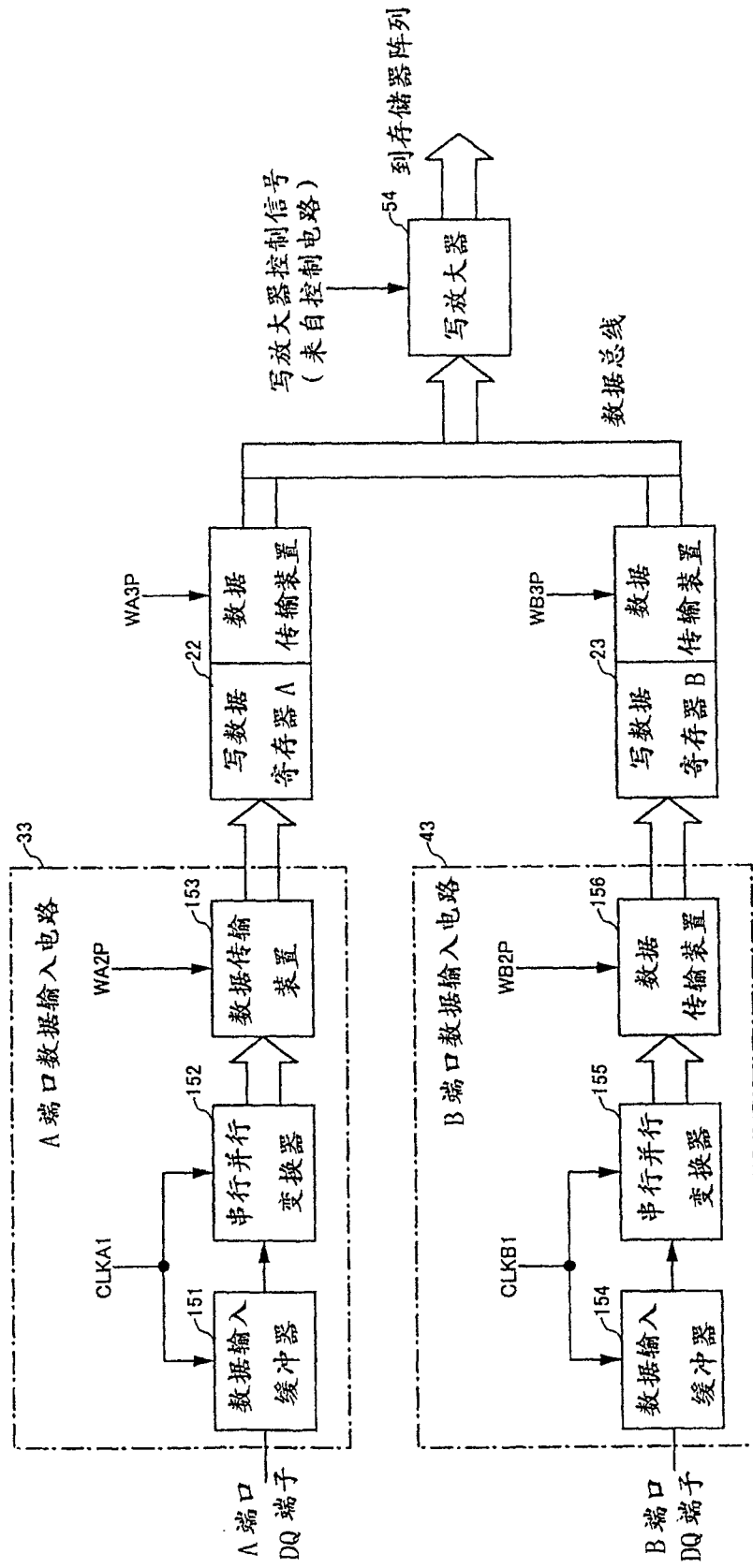


图11



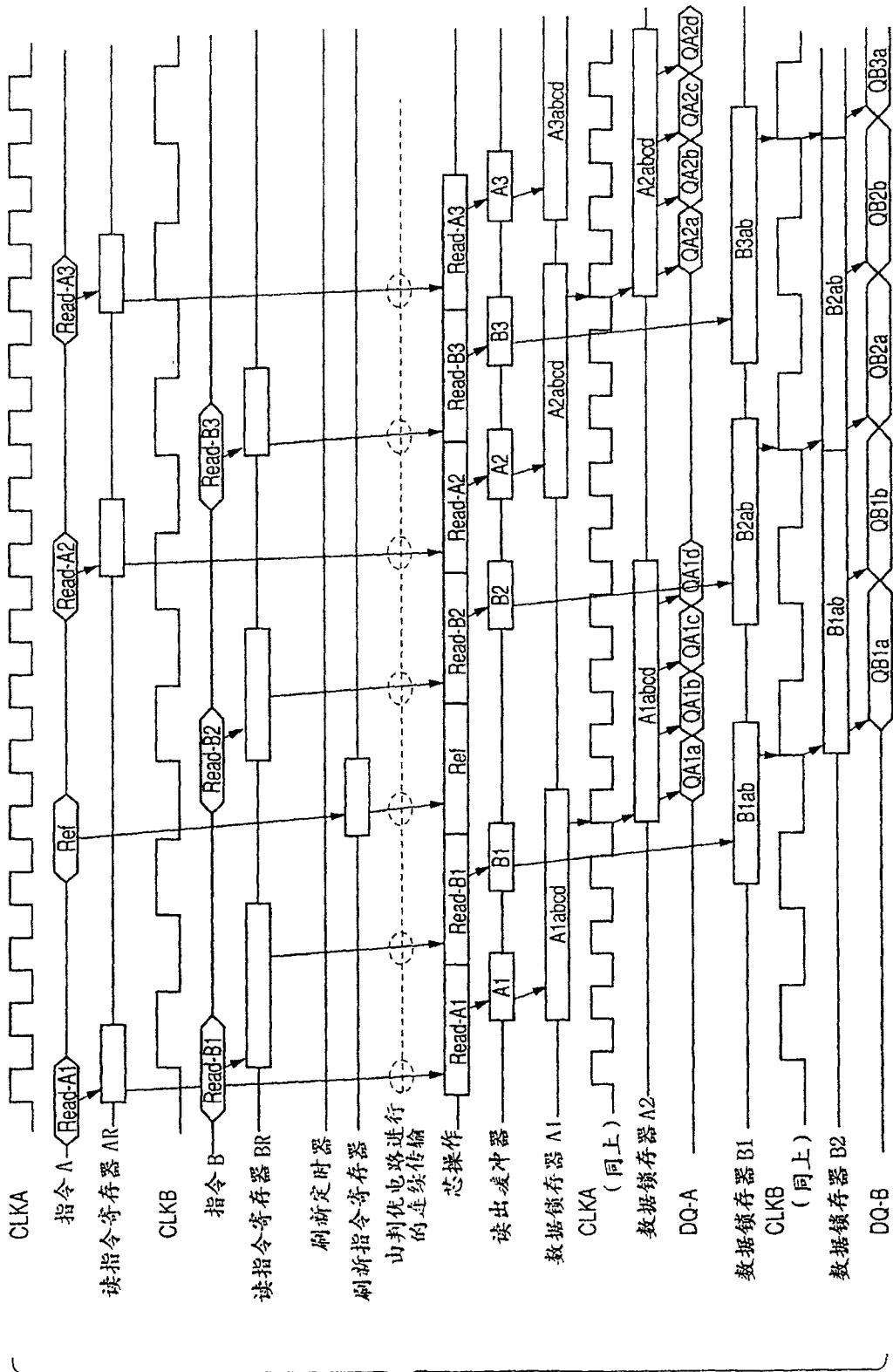


图12

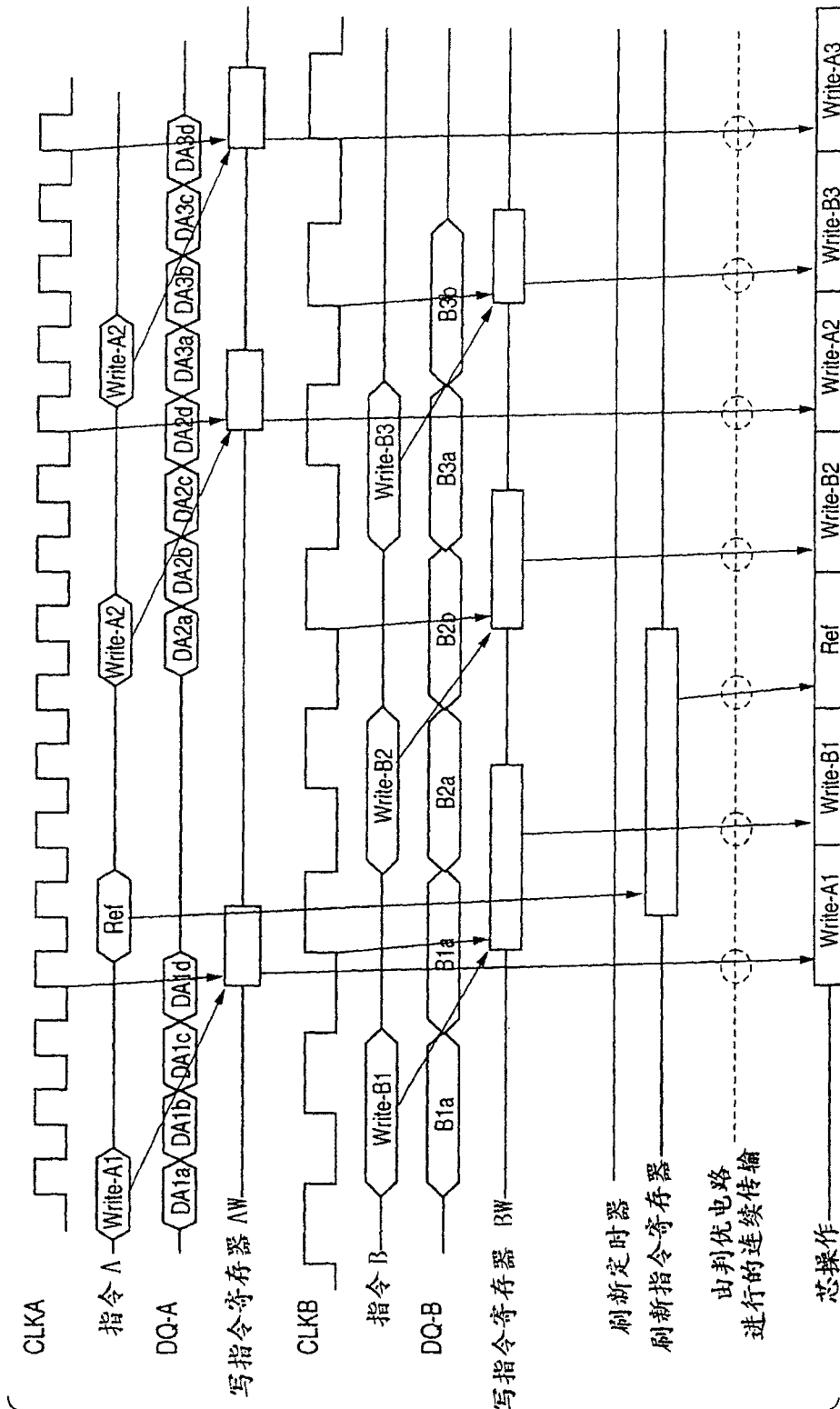


图 13

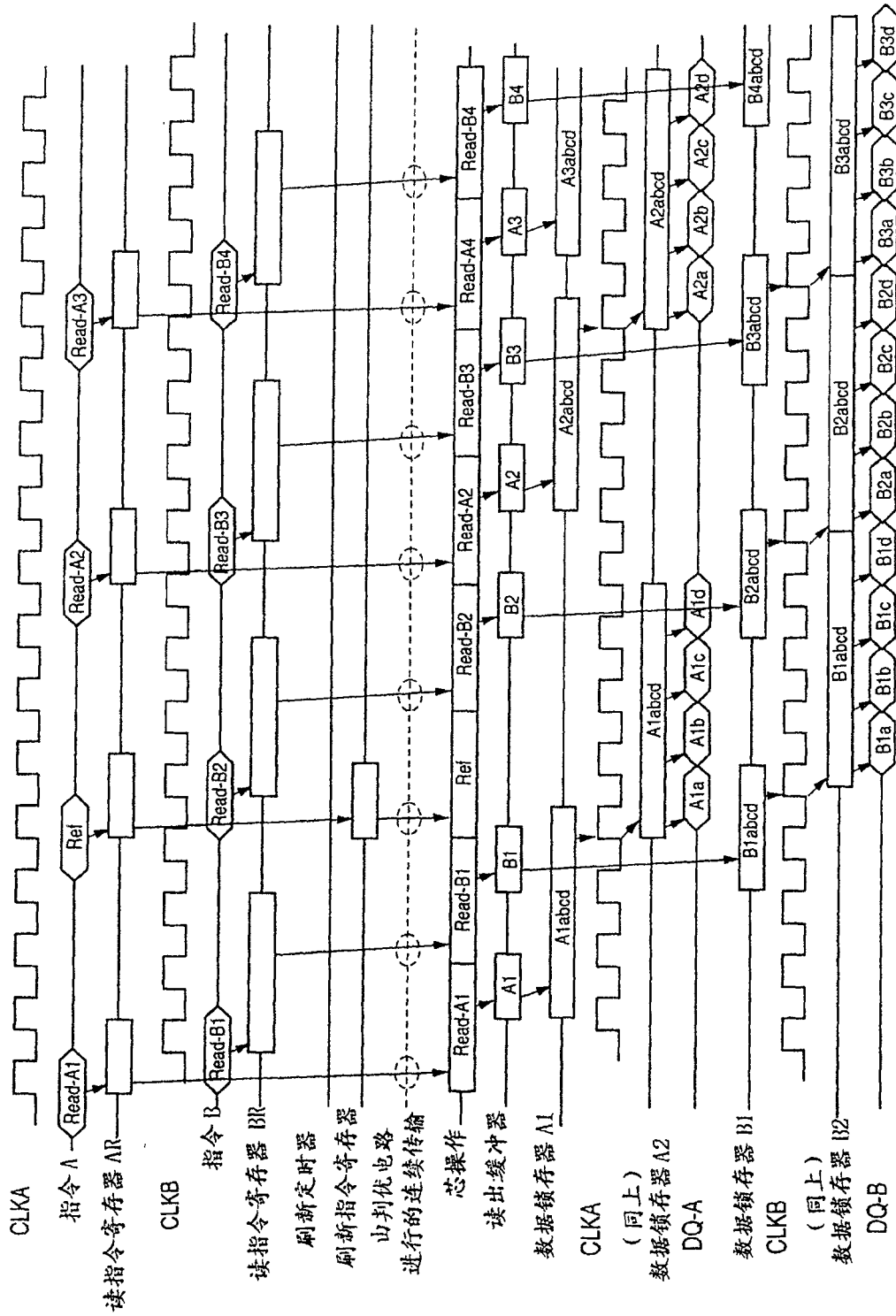


图 14

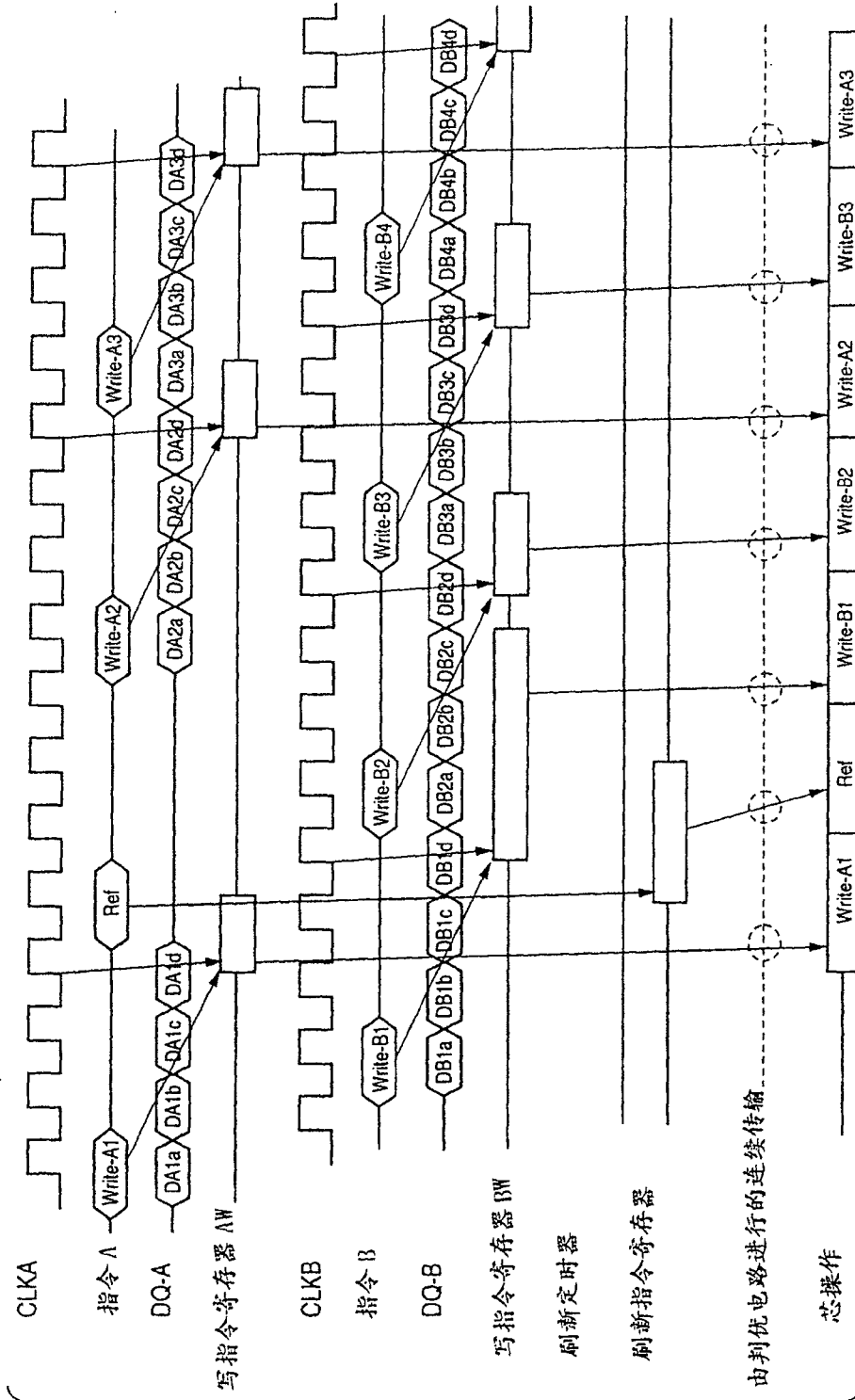


图 15

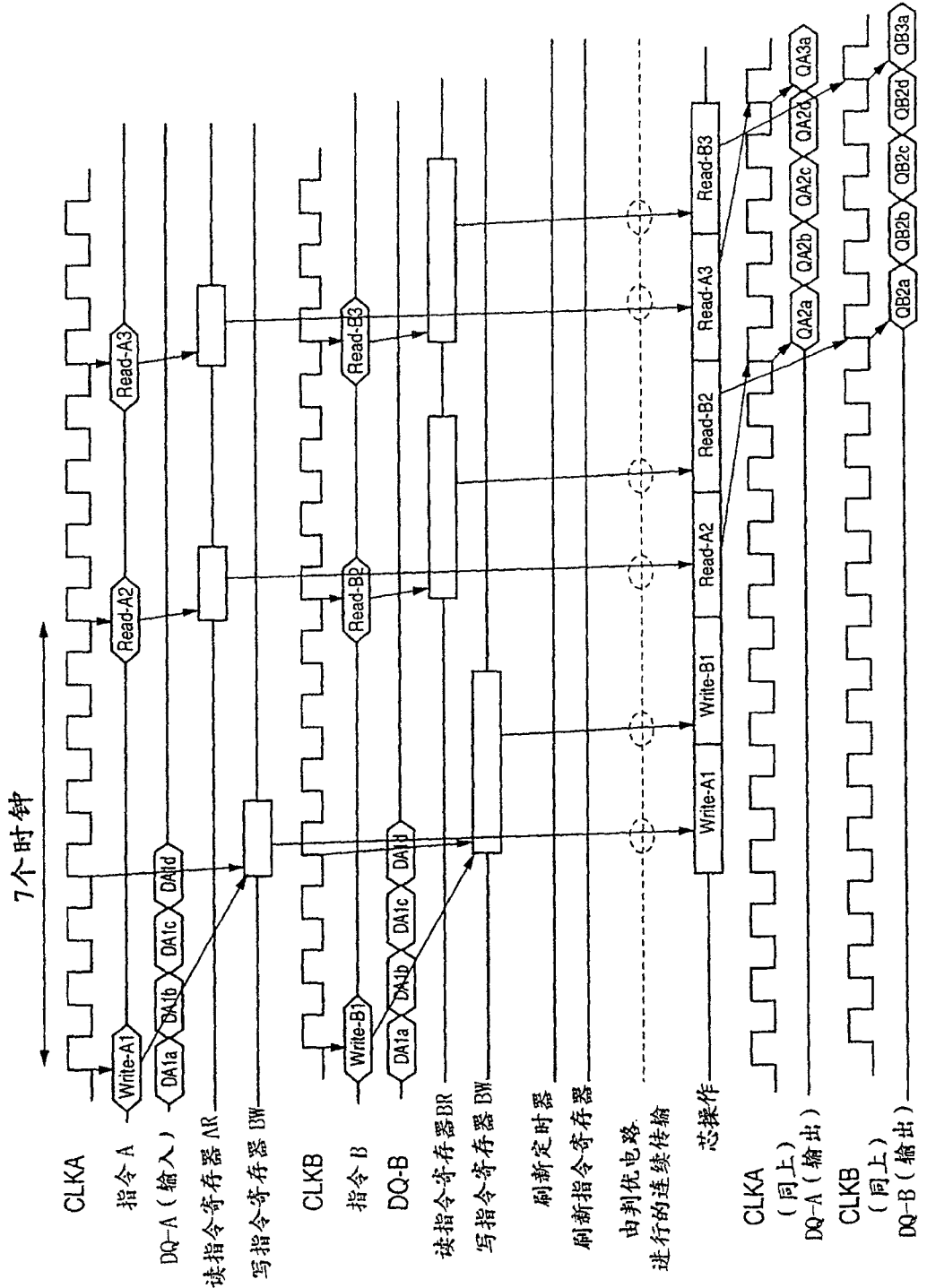


图16

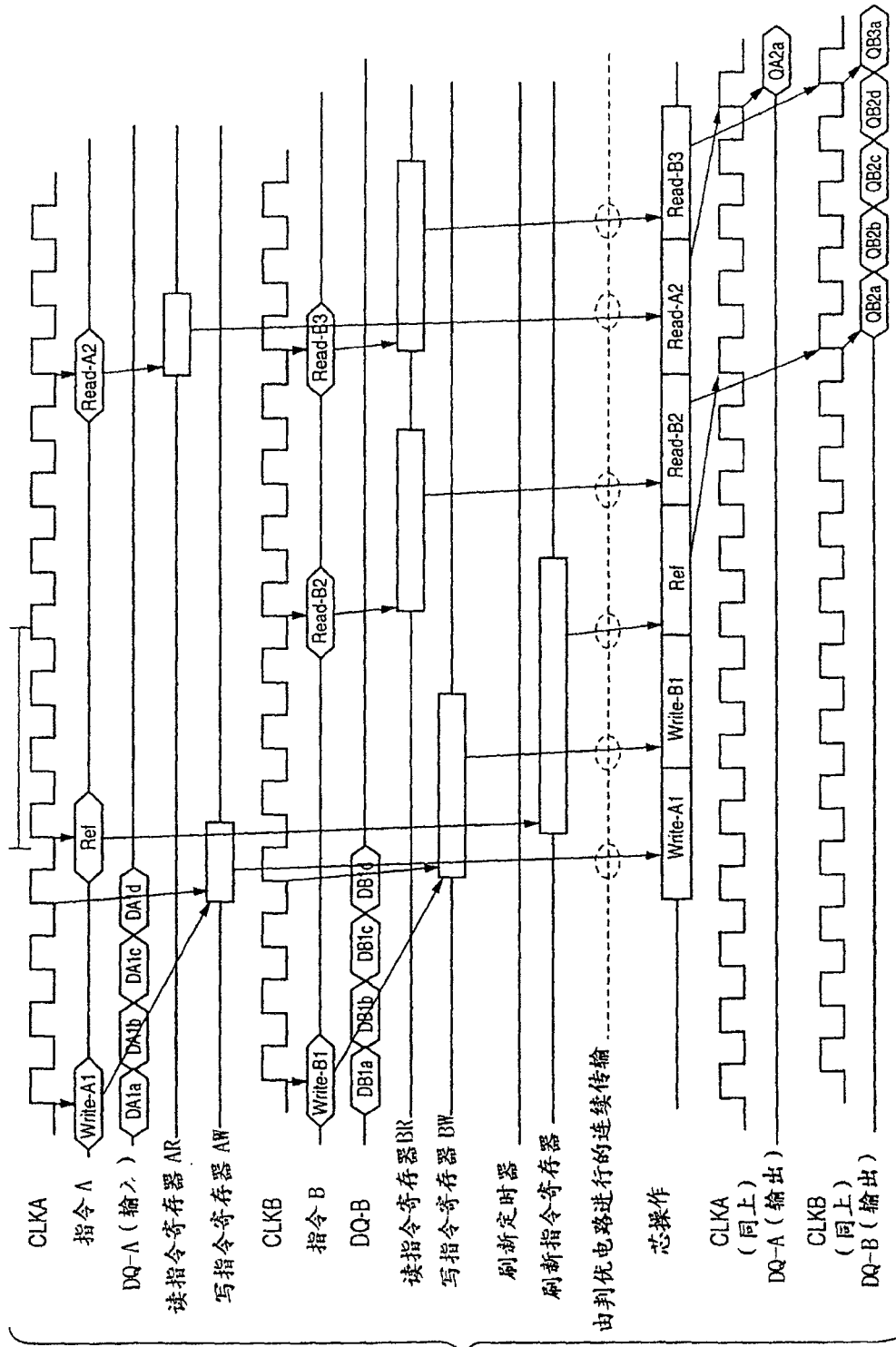


图 17

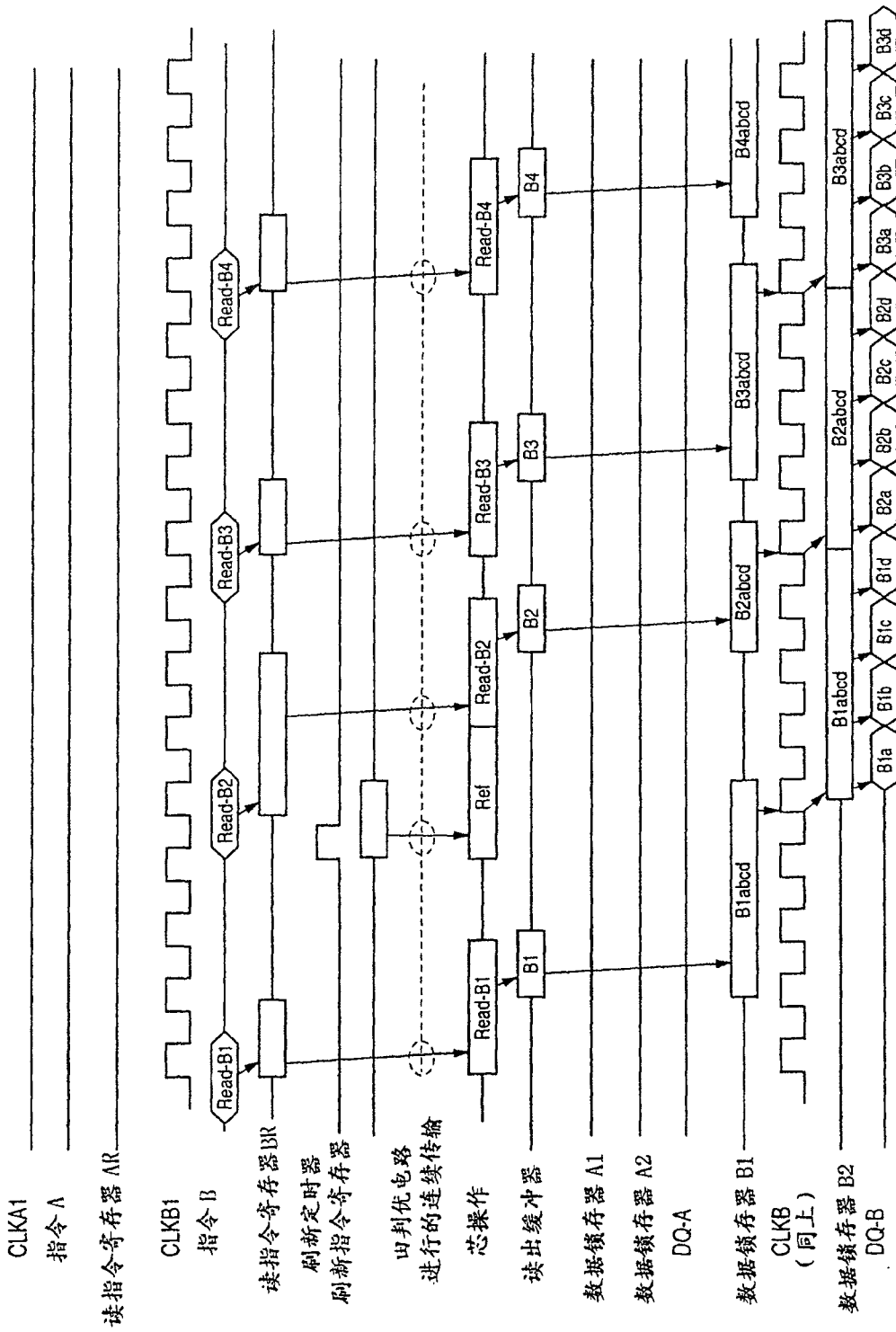


图18

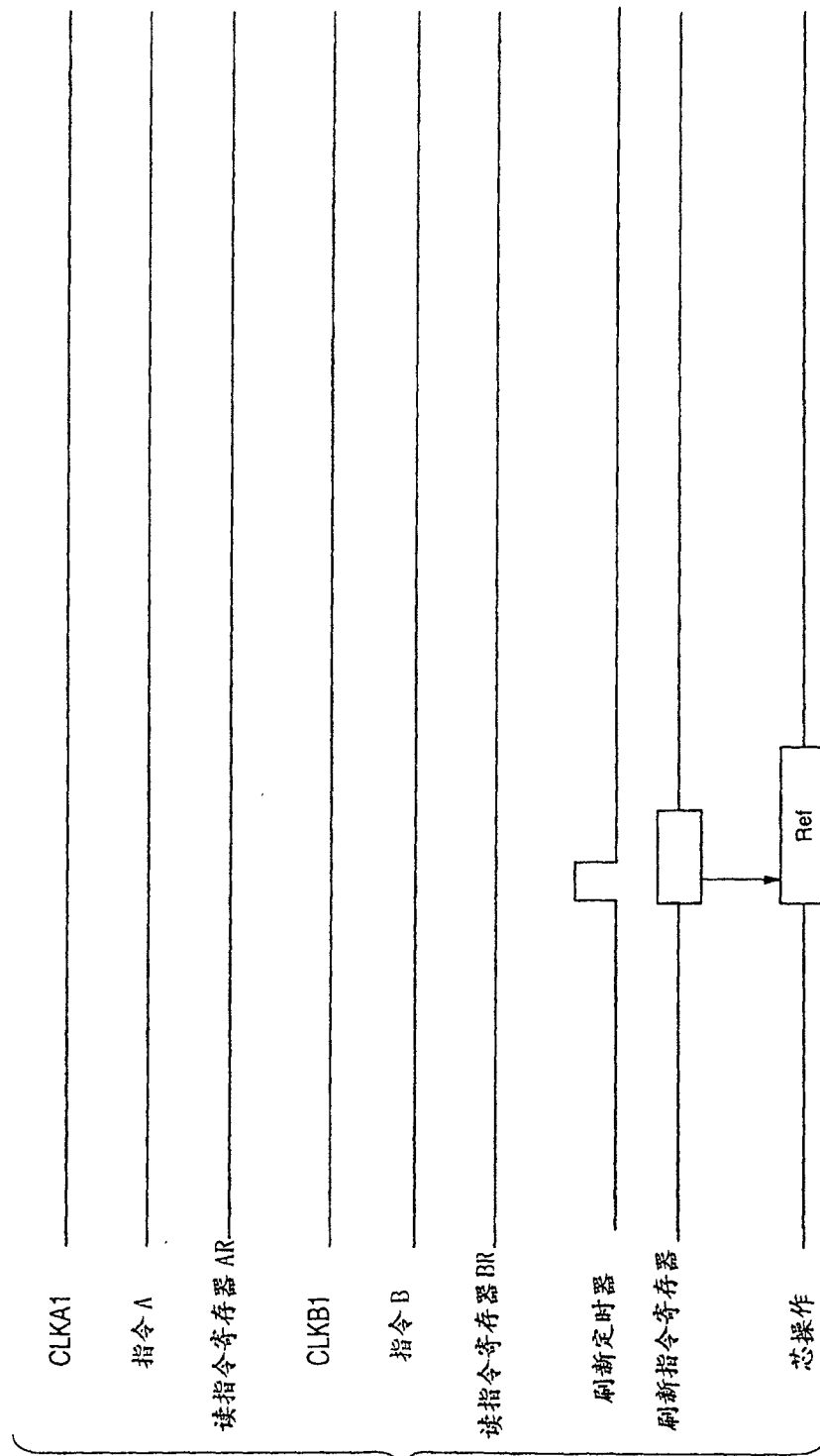


图19

图 20B

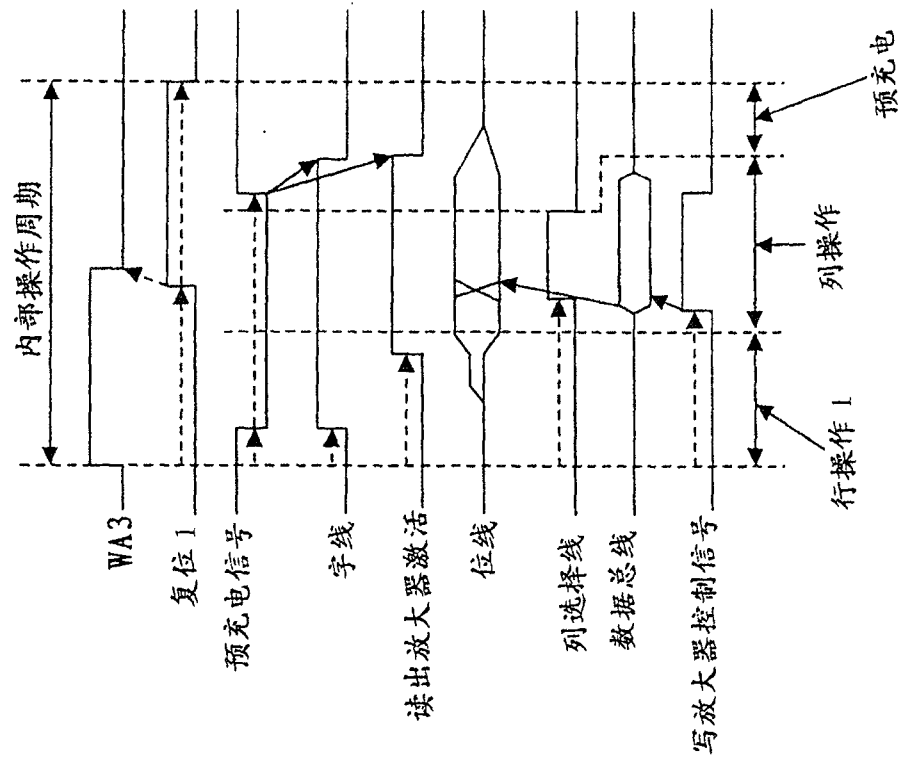
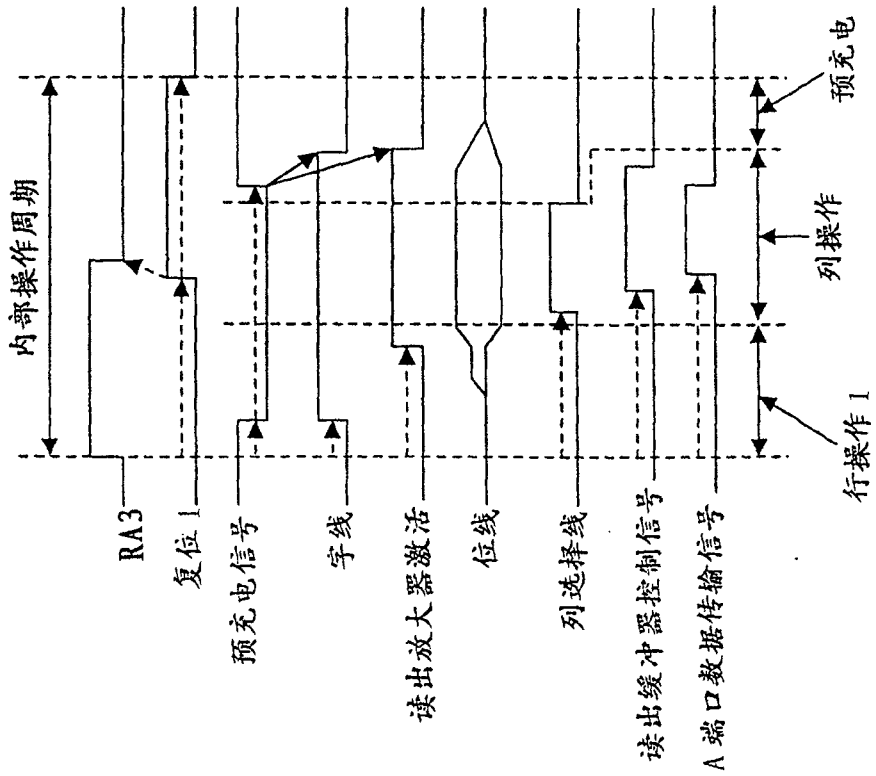


图 20A



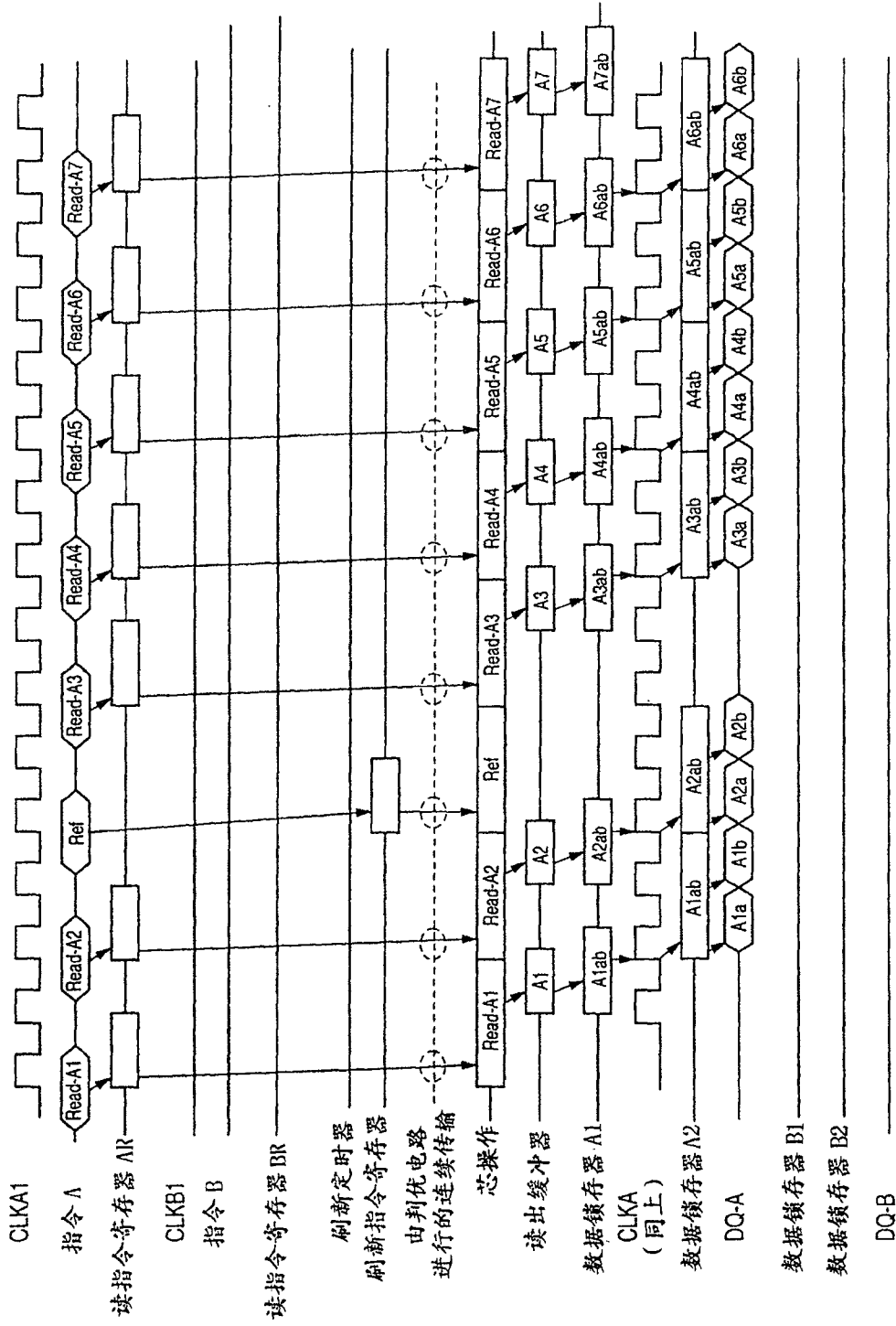


图 21

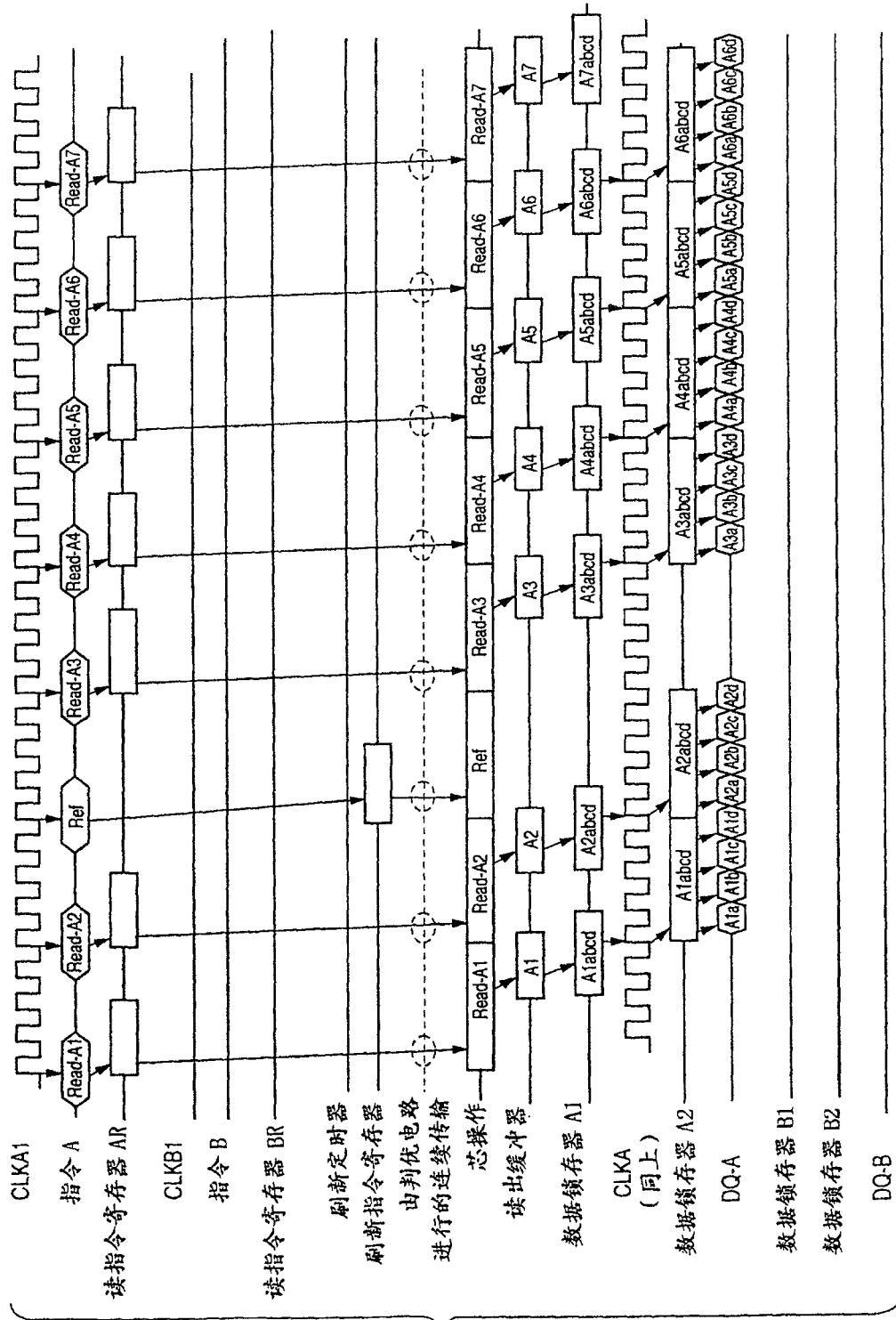


图22

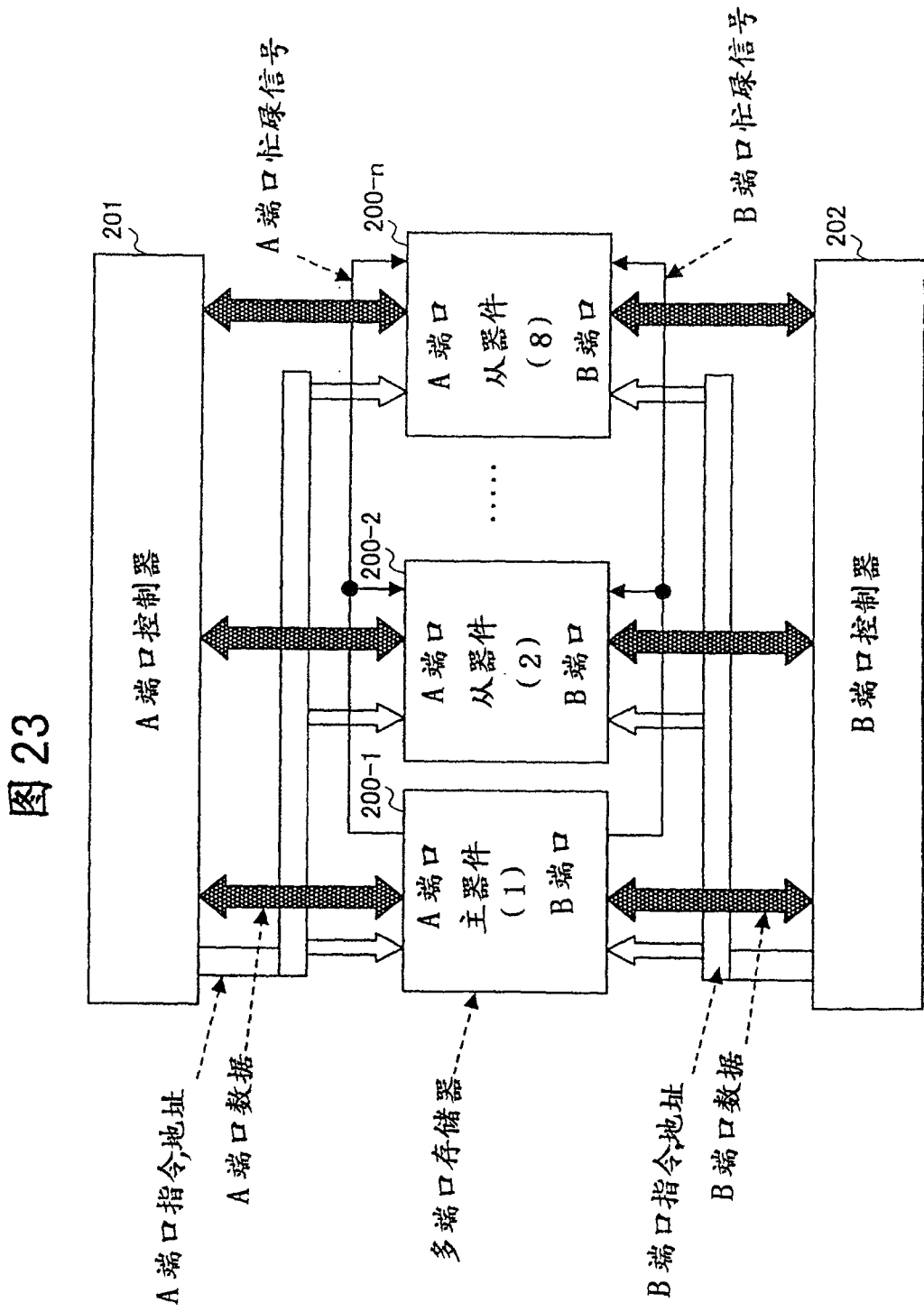


图 24

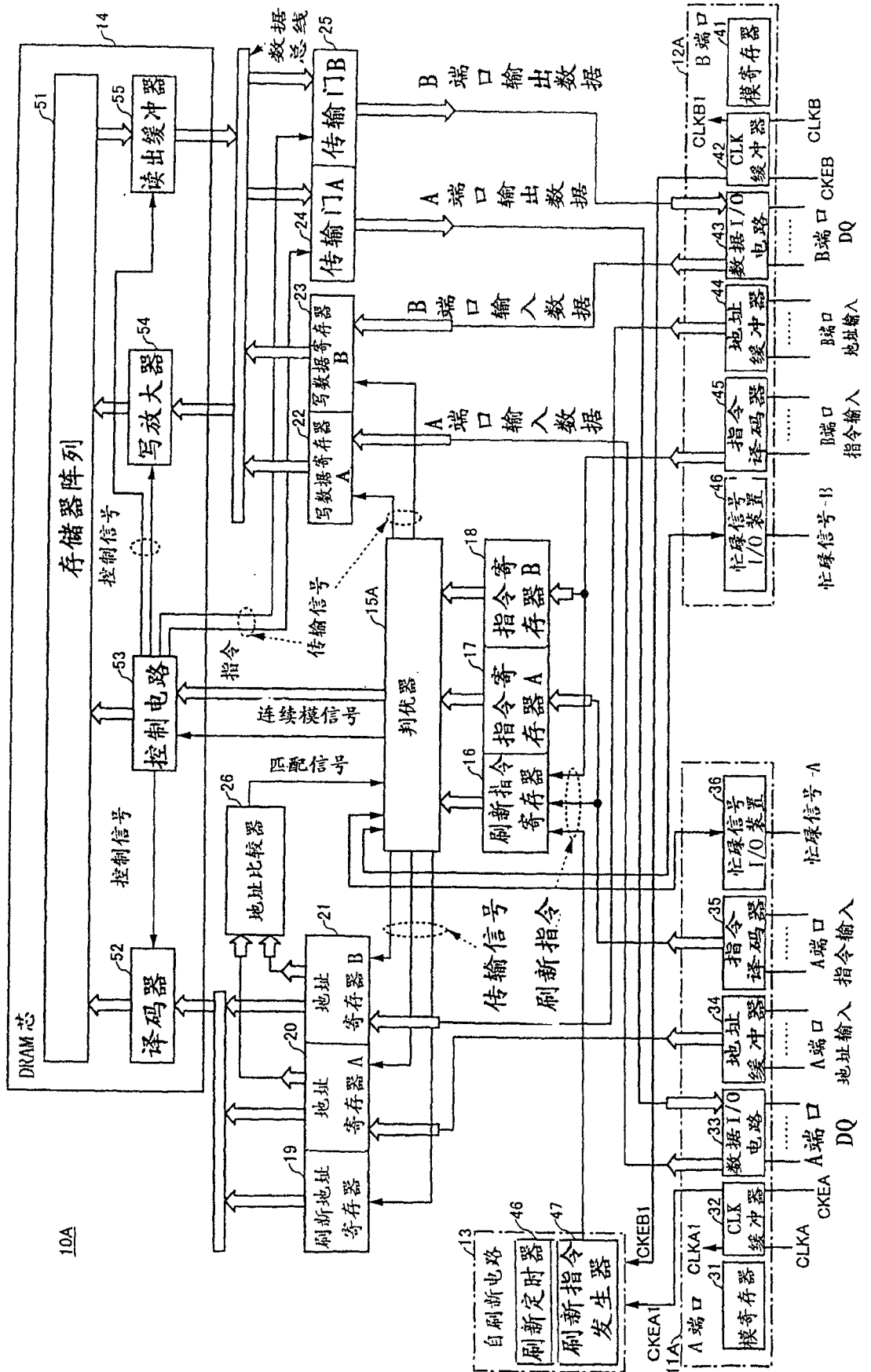


图25A

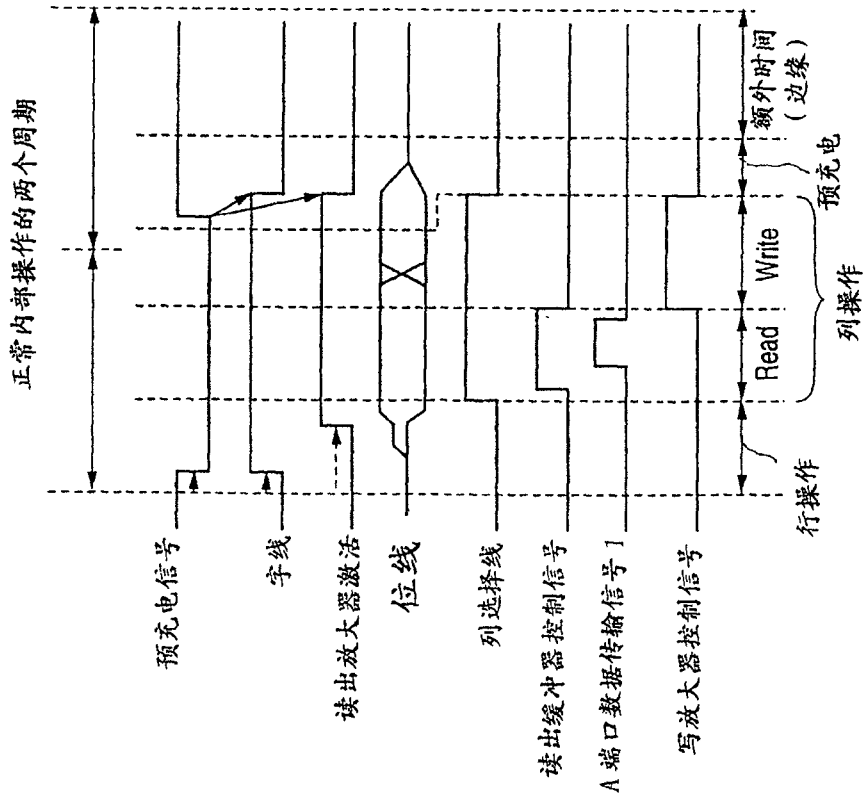
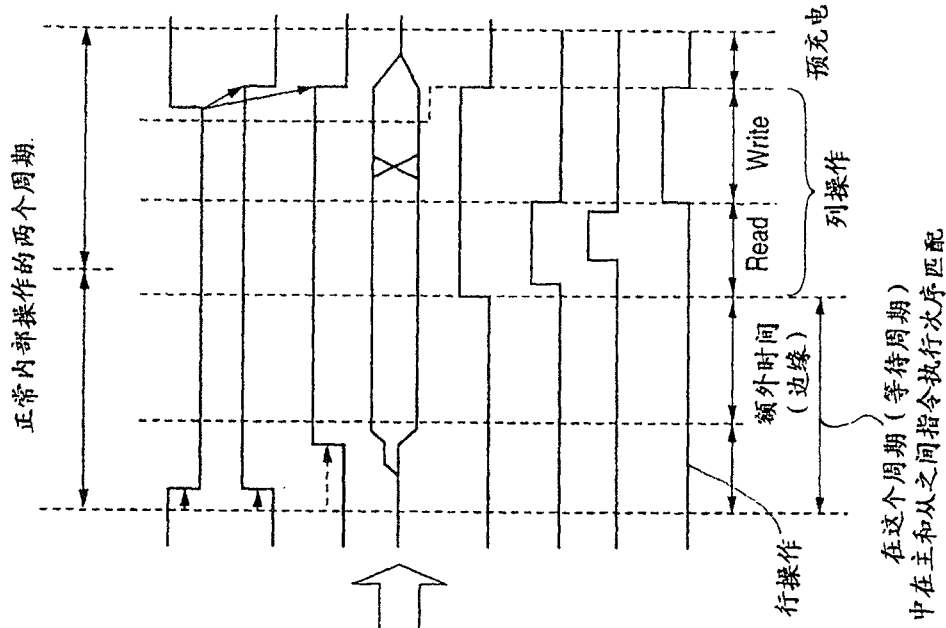


图25B



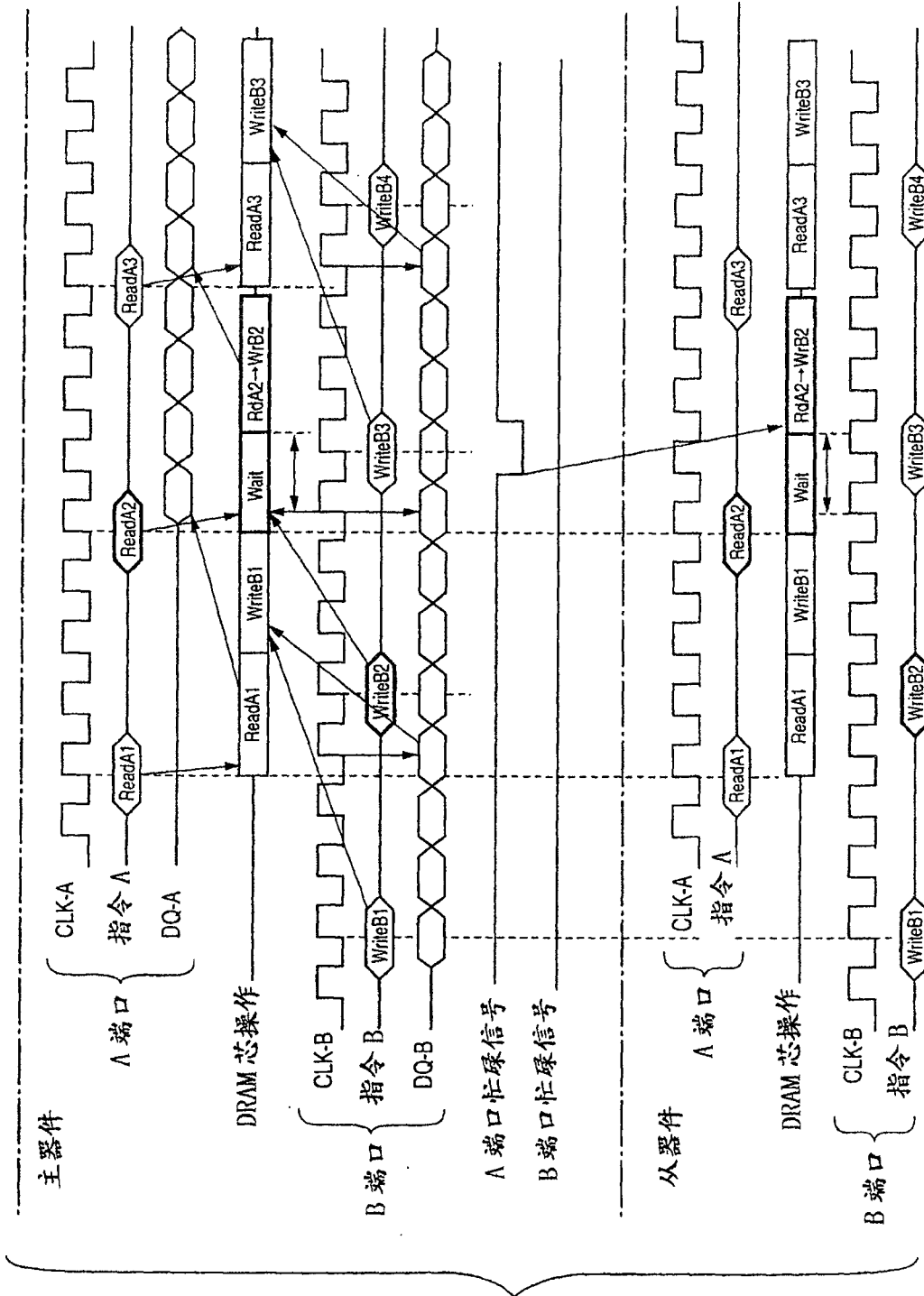


图 26

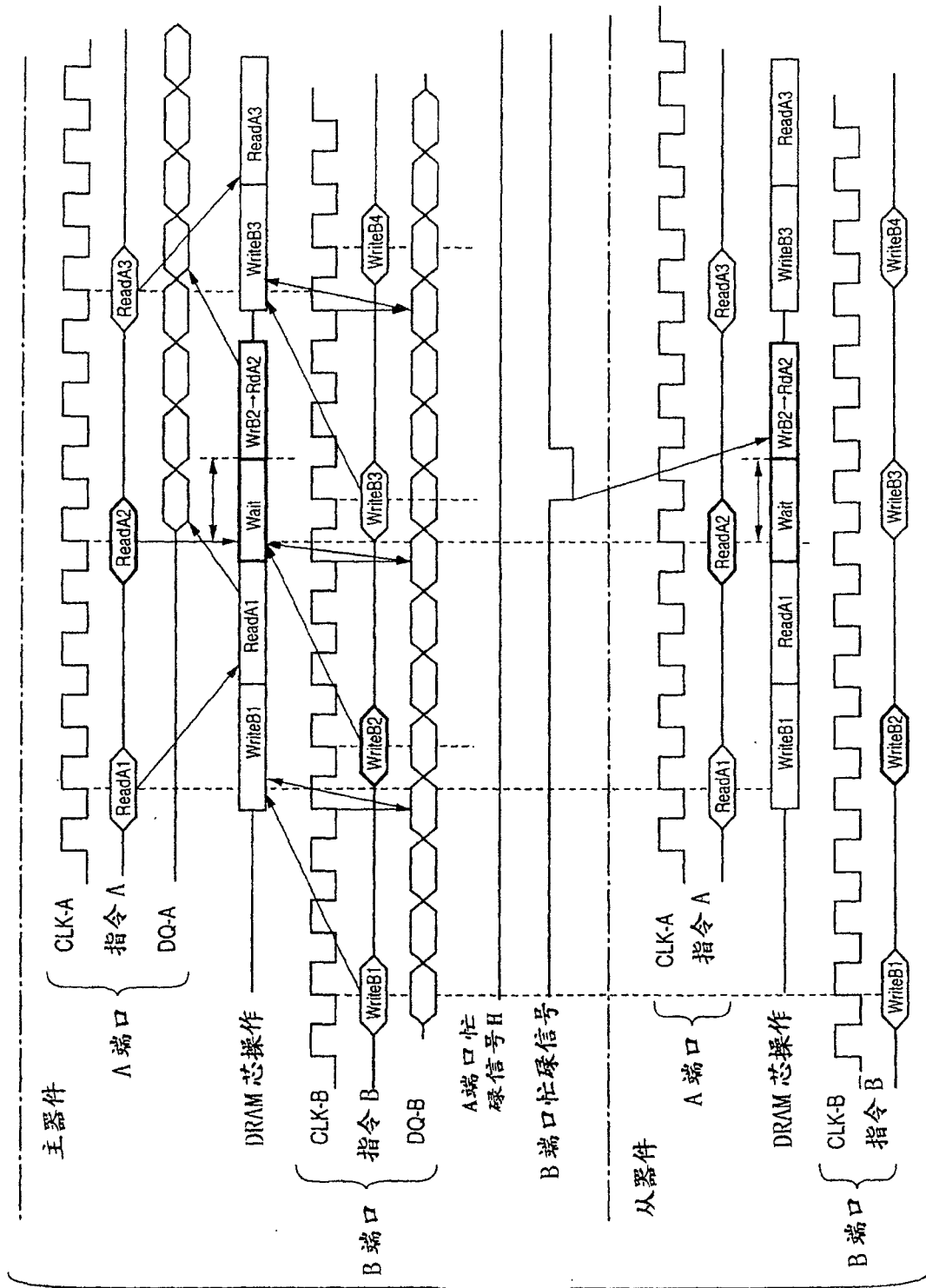


图 27

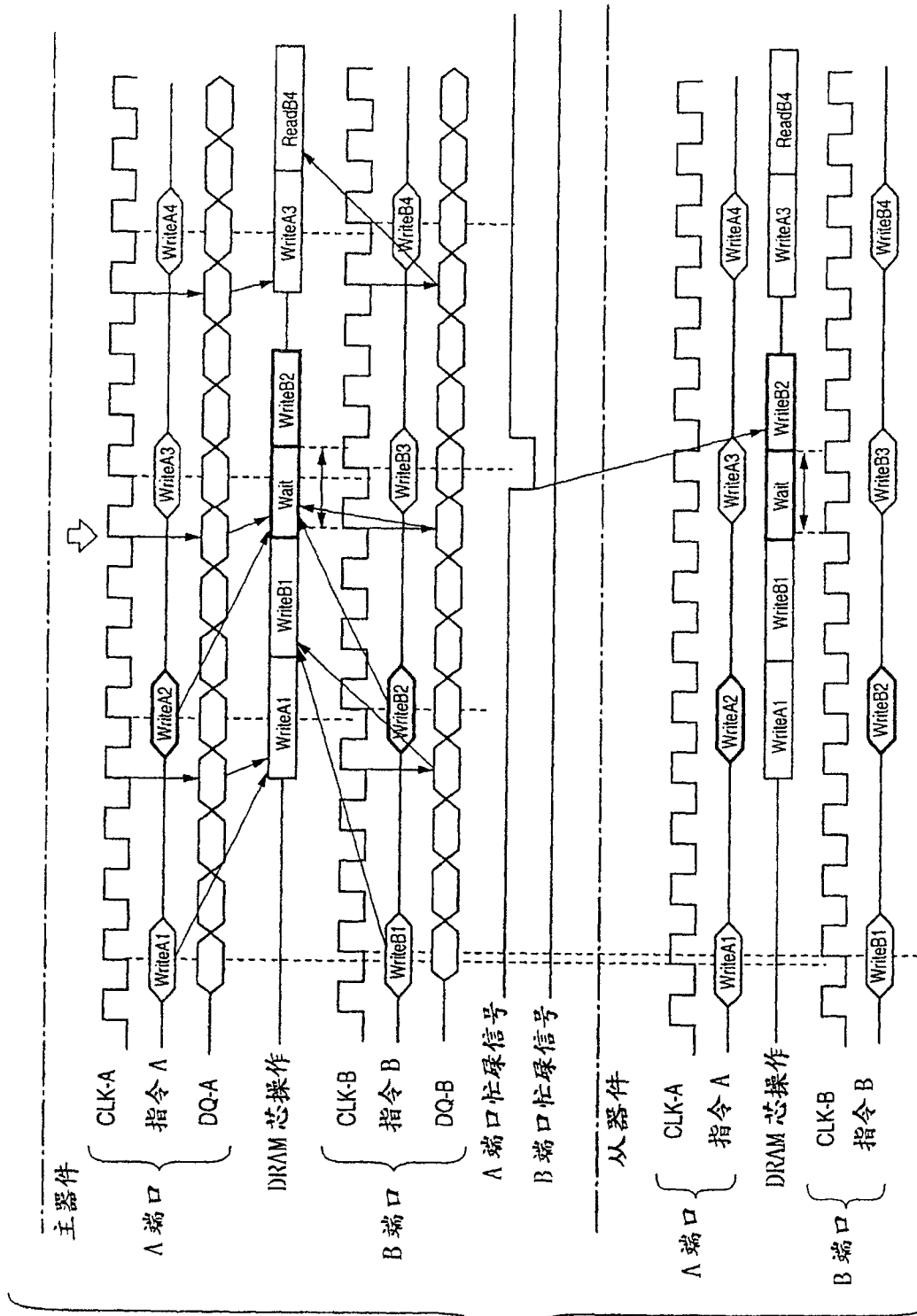


图 28

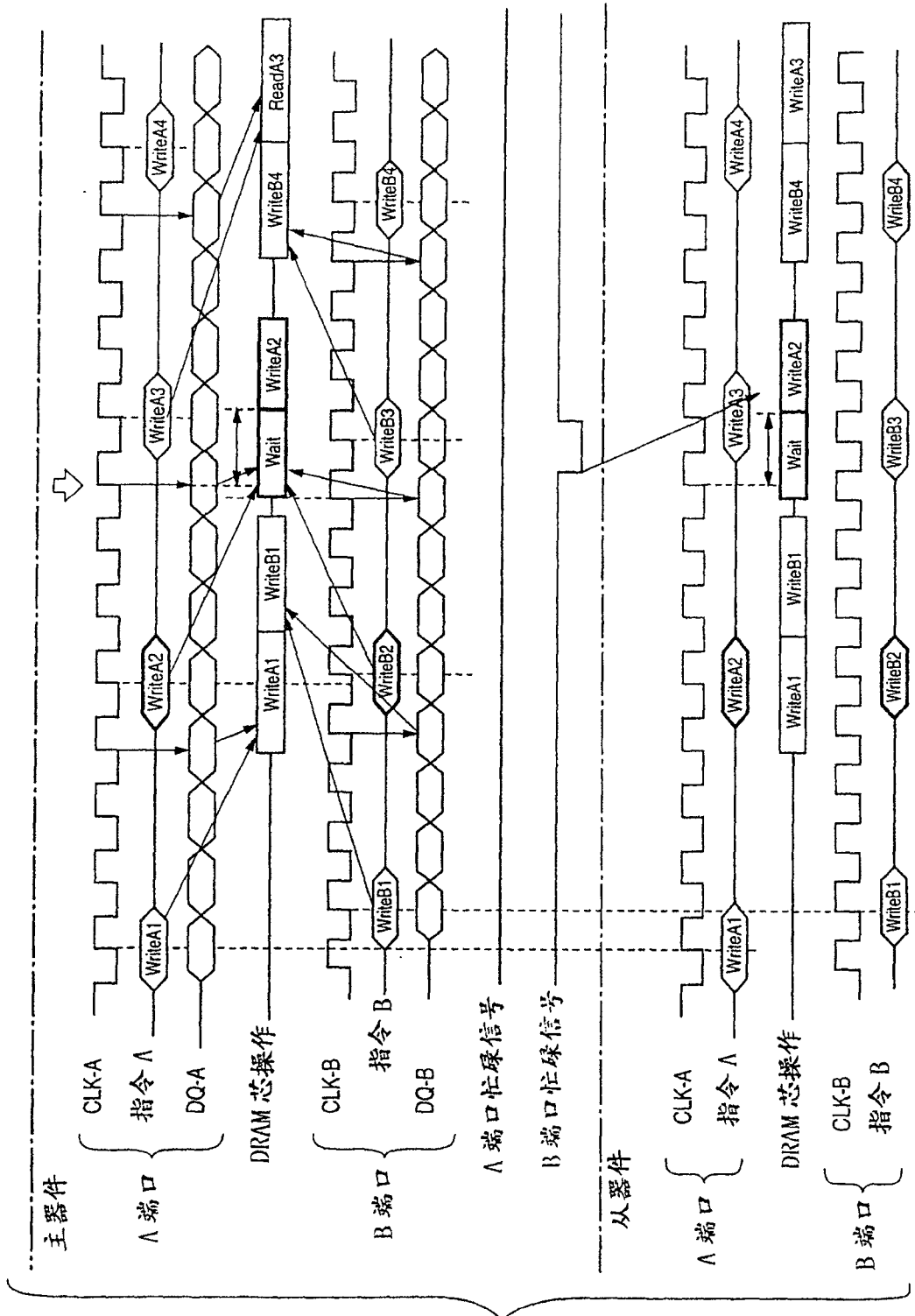


图 29

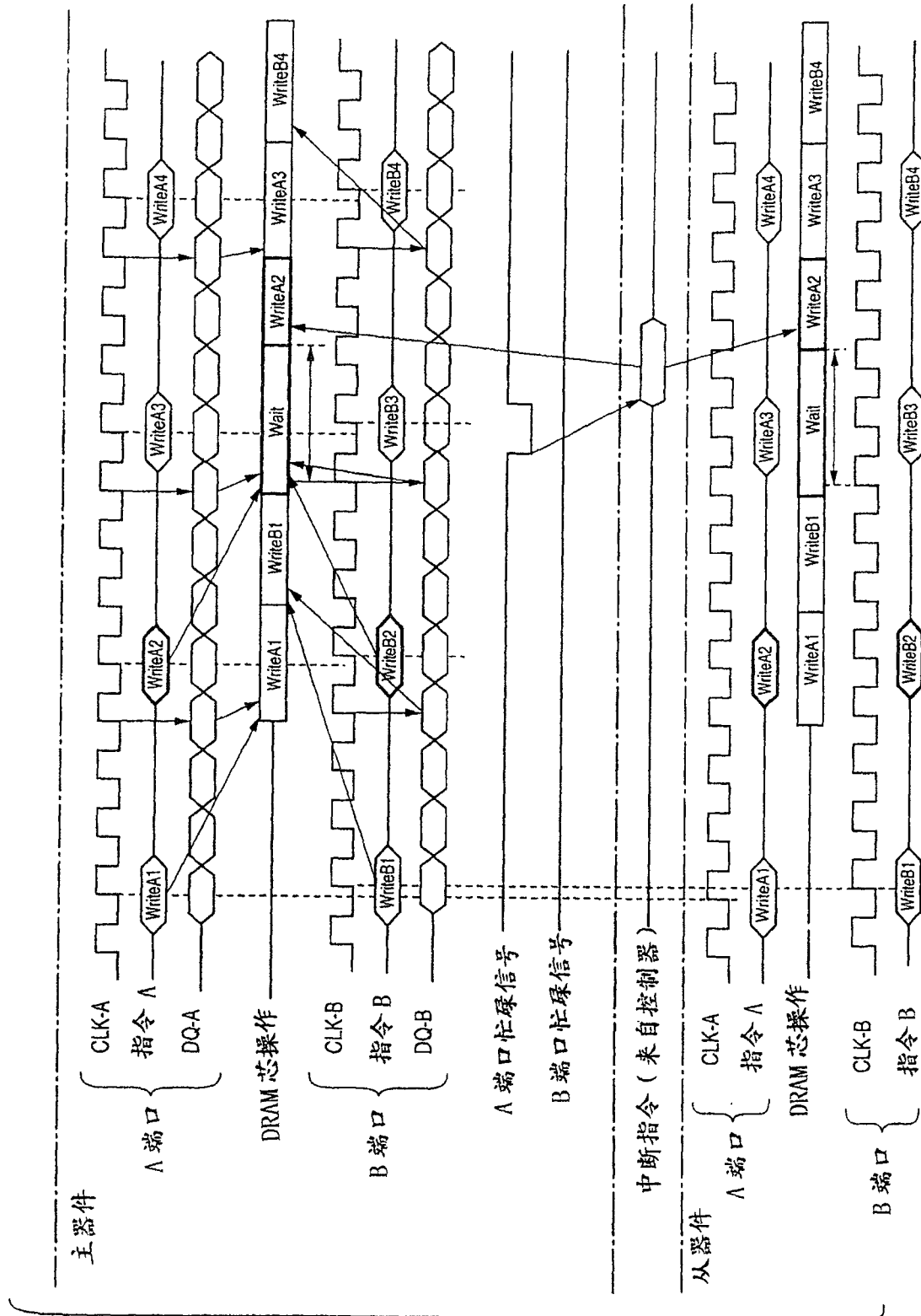
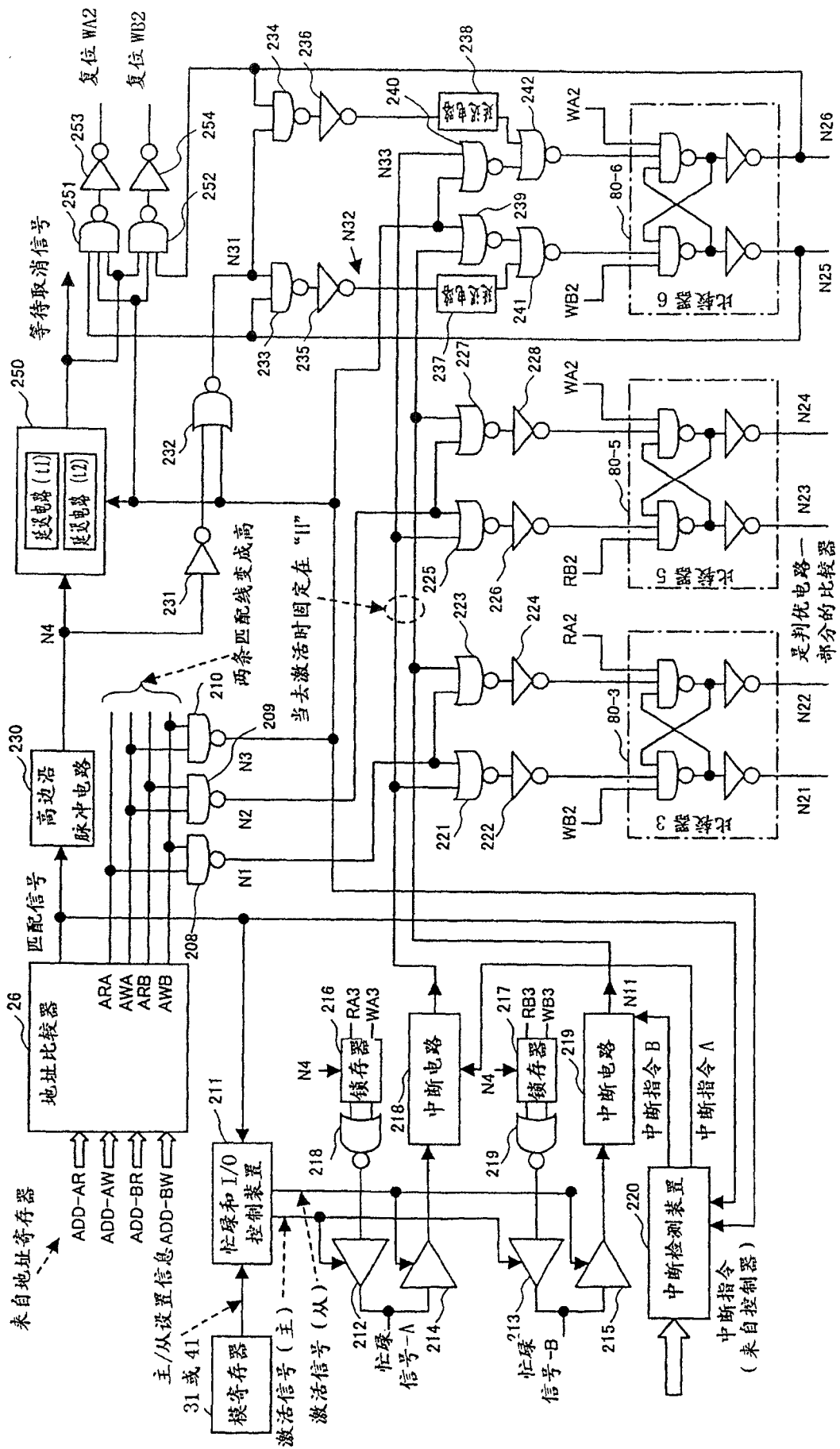


图 30

图31



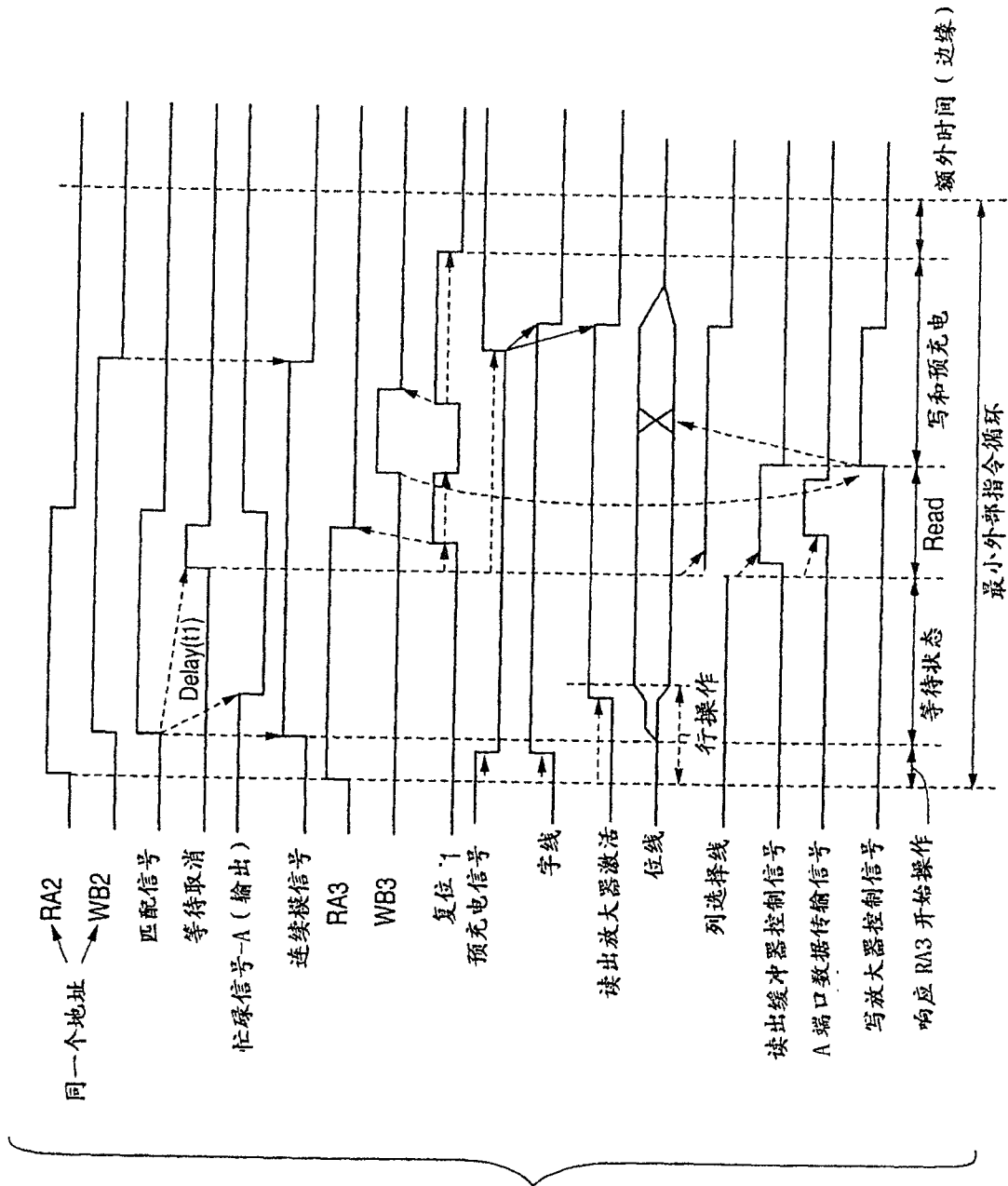


图32

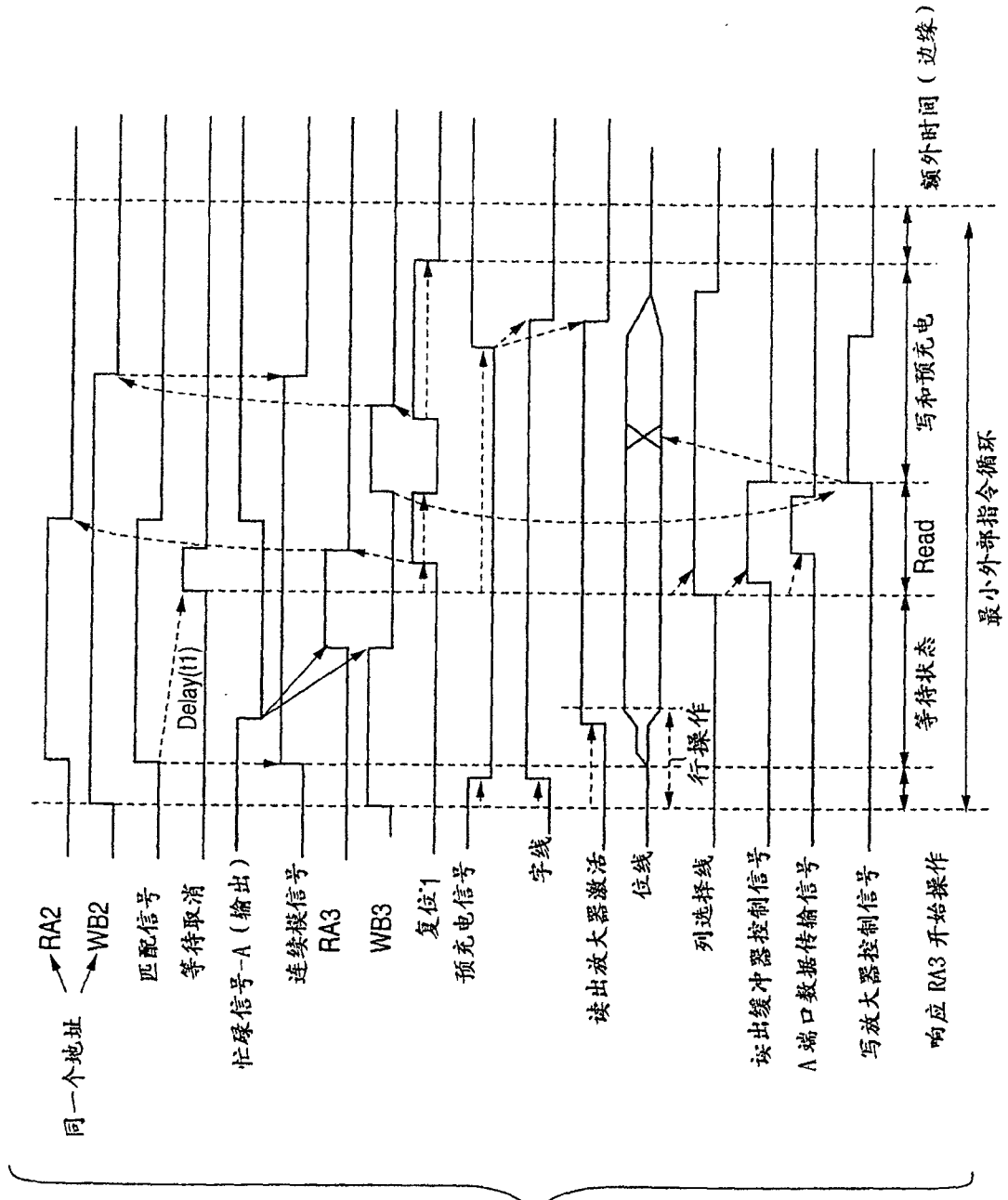


图 33

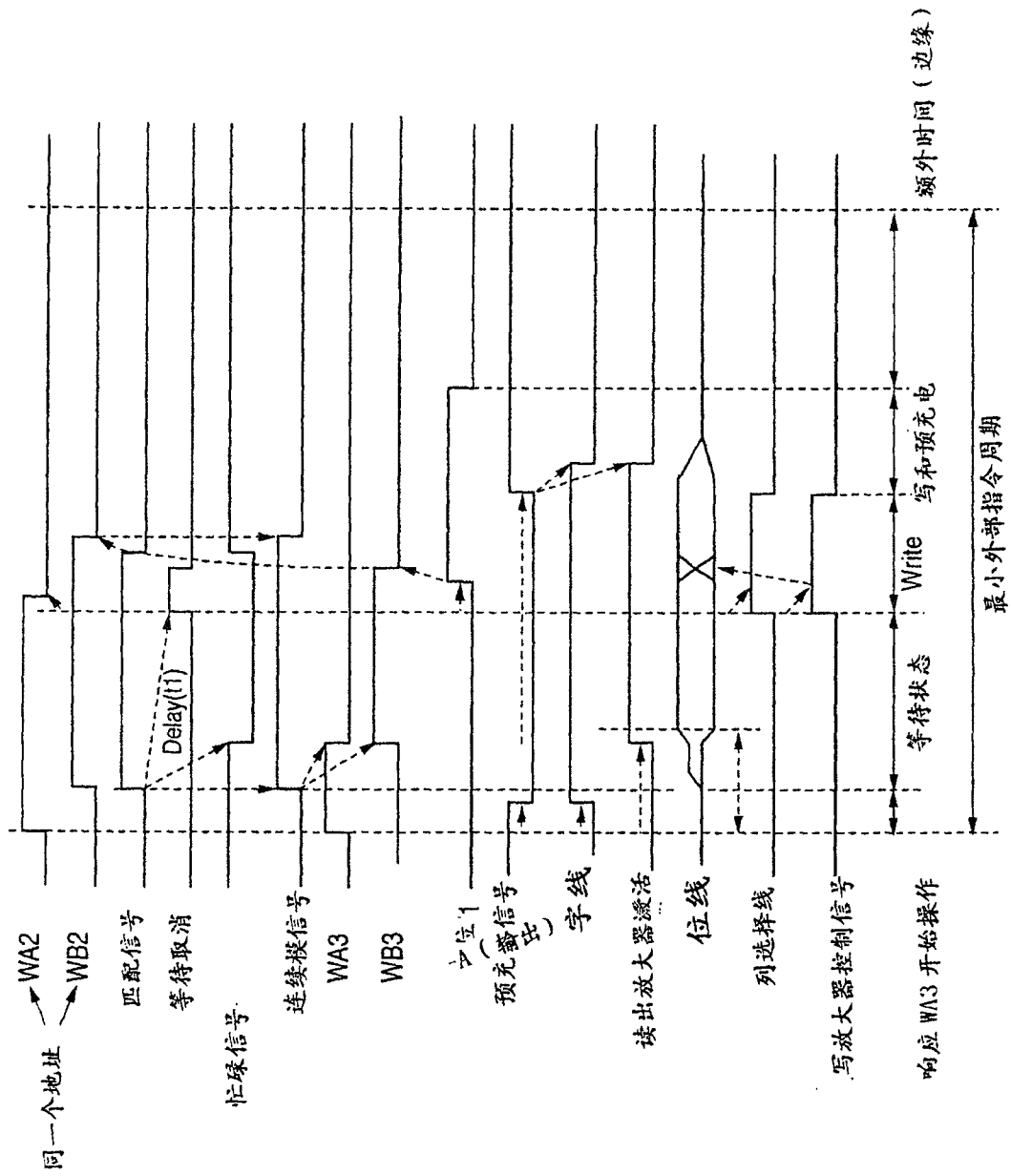


图 34

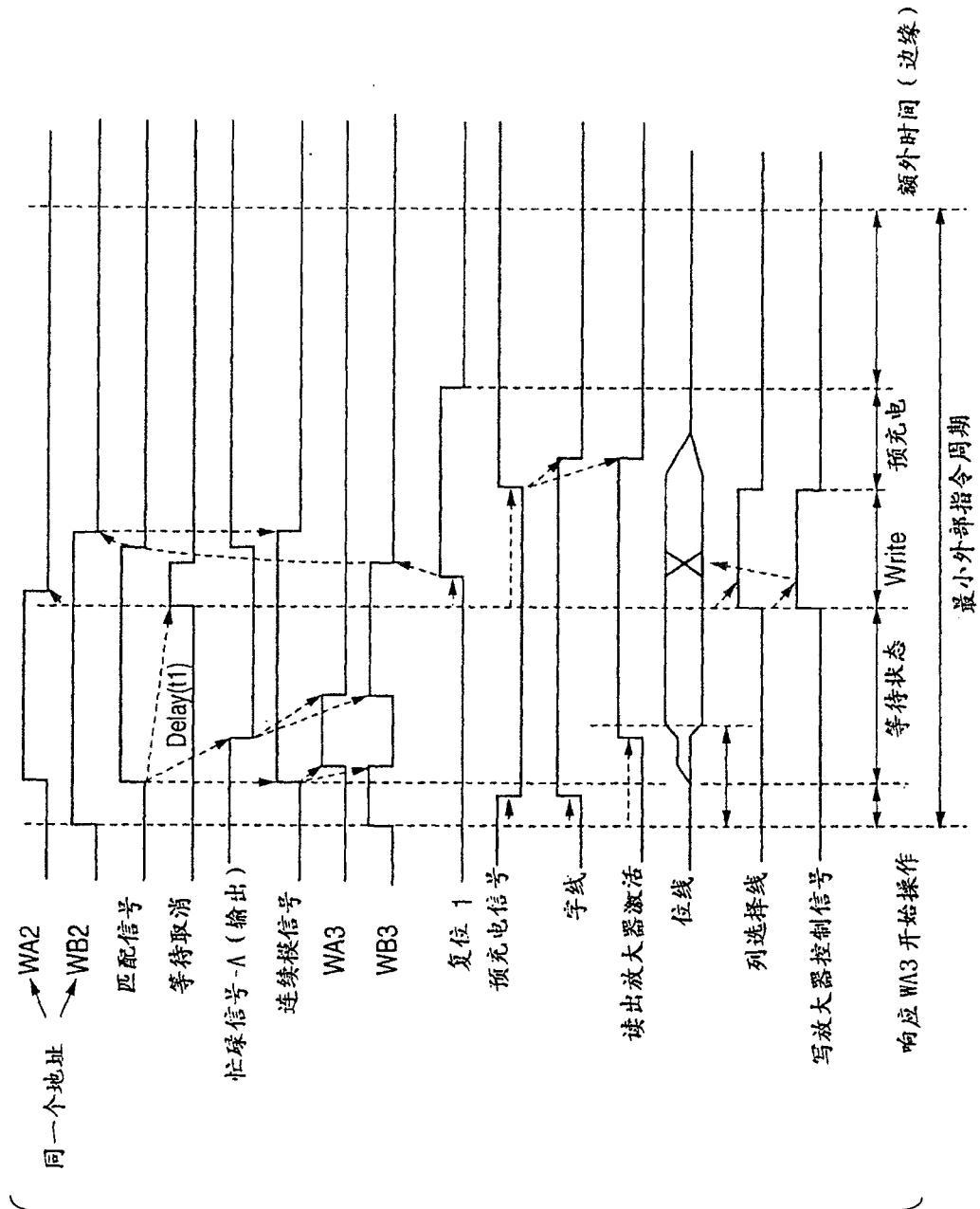


图 35

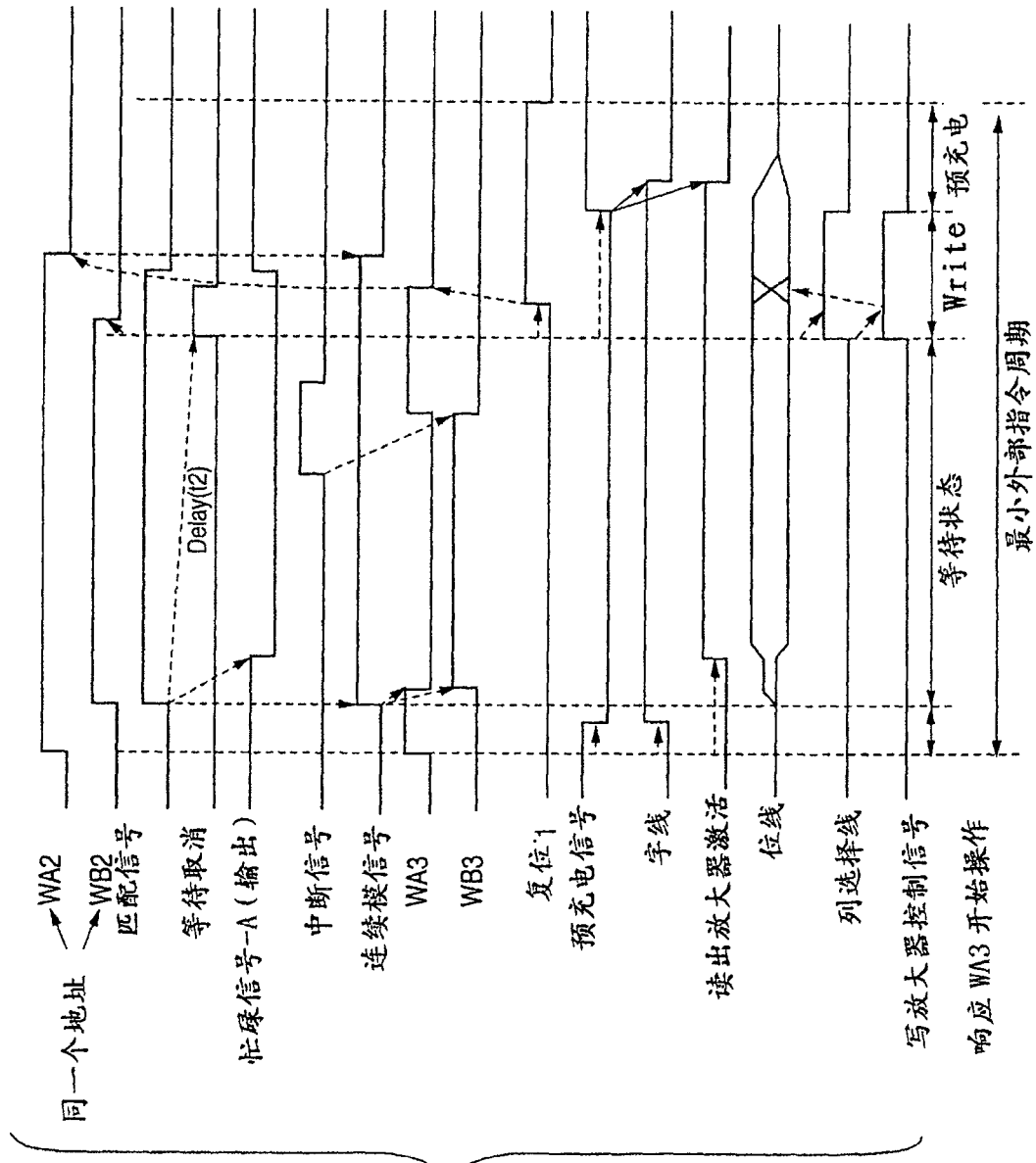


图 36

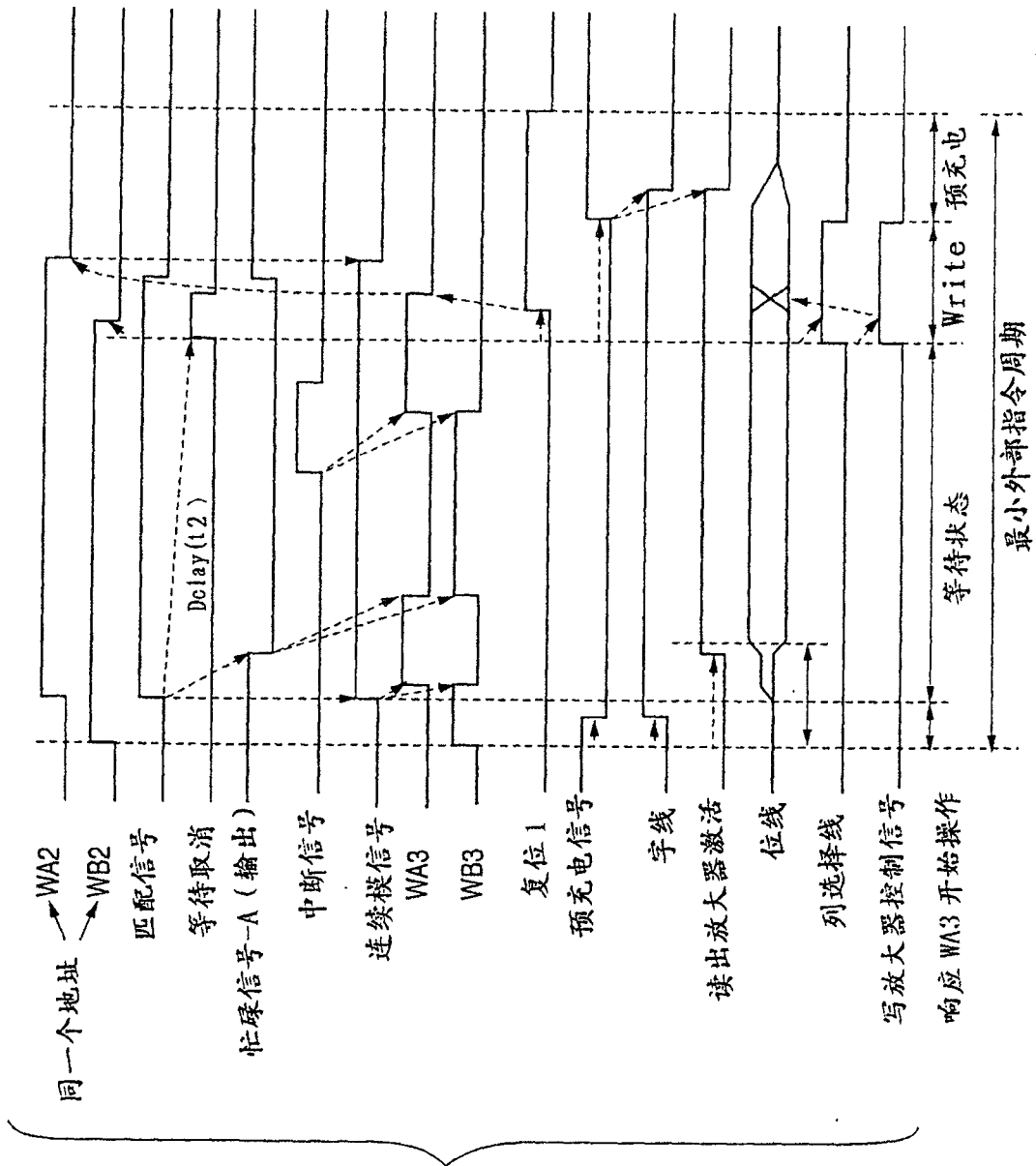


图 37

图38

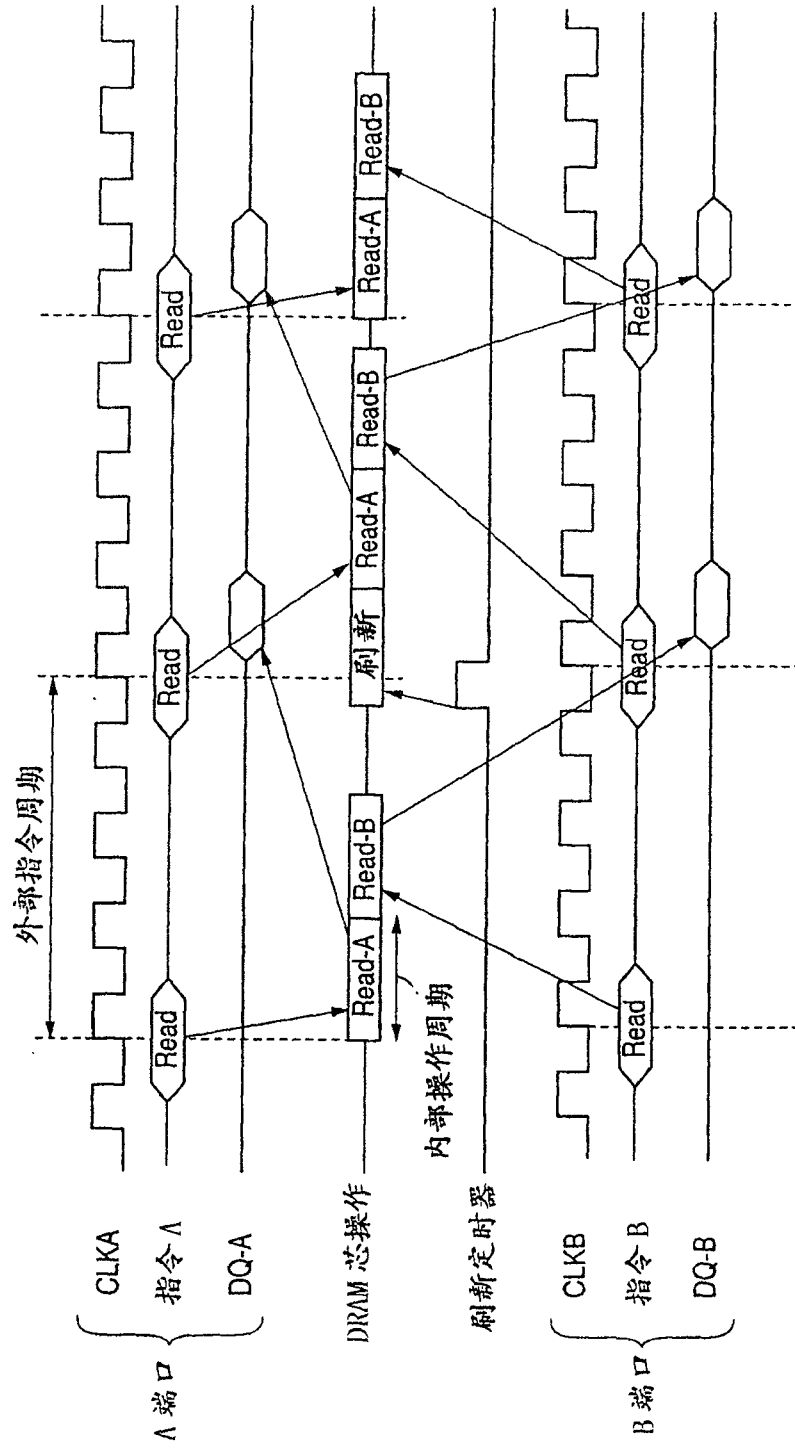
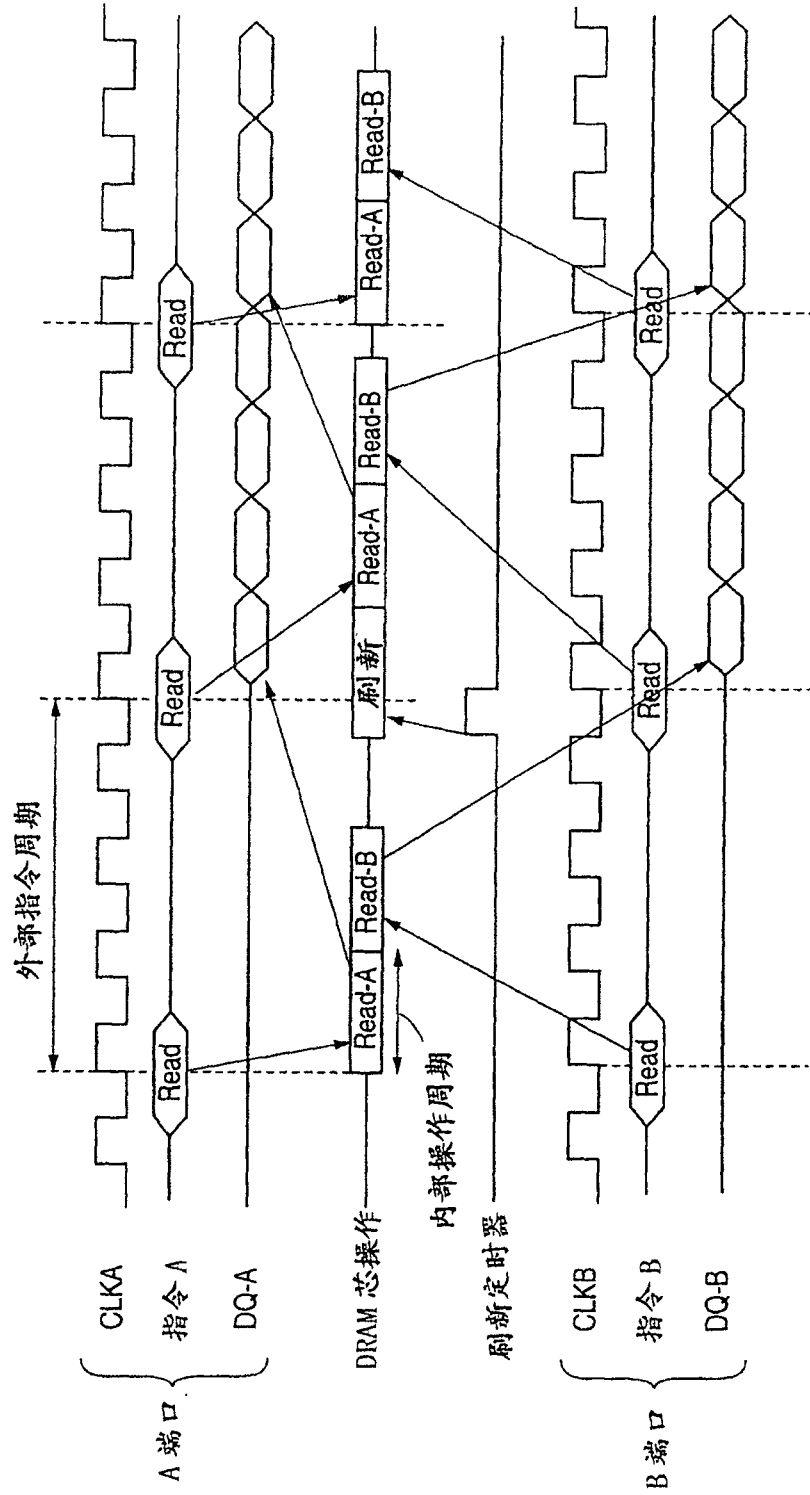


图39



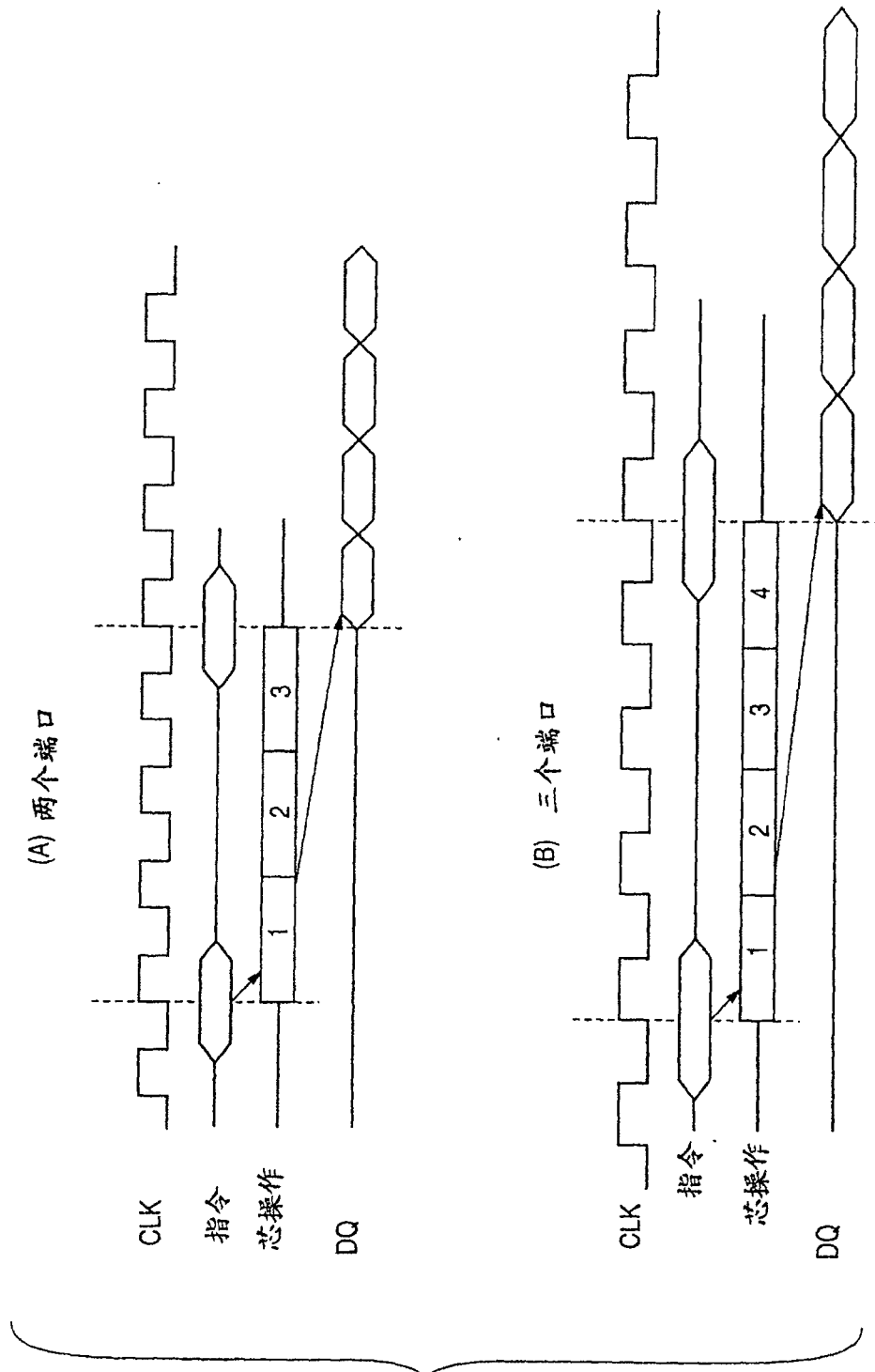


图40

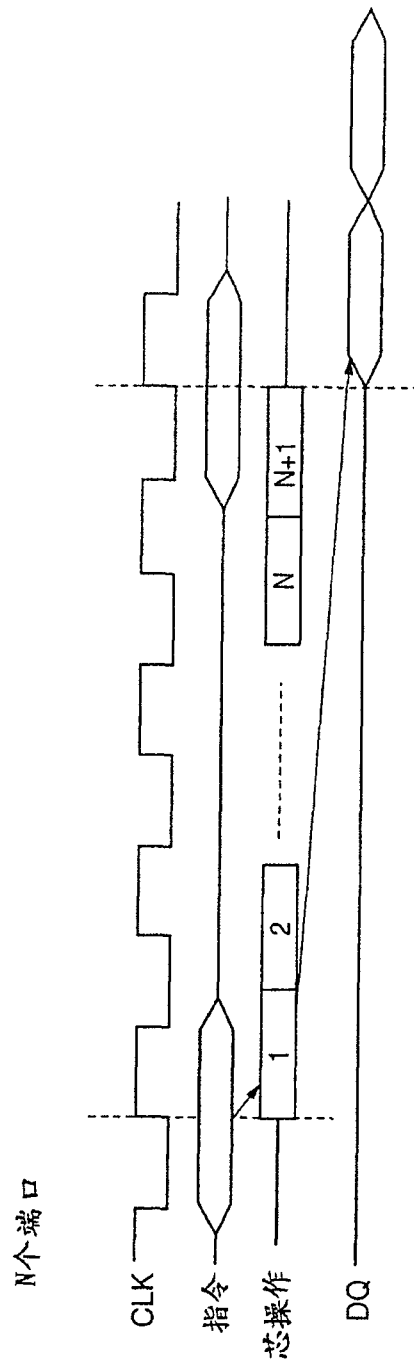


图 41

图42

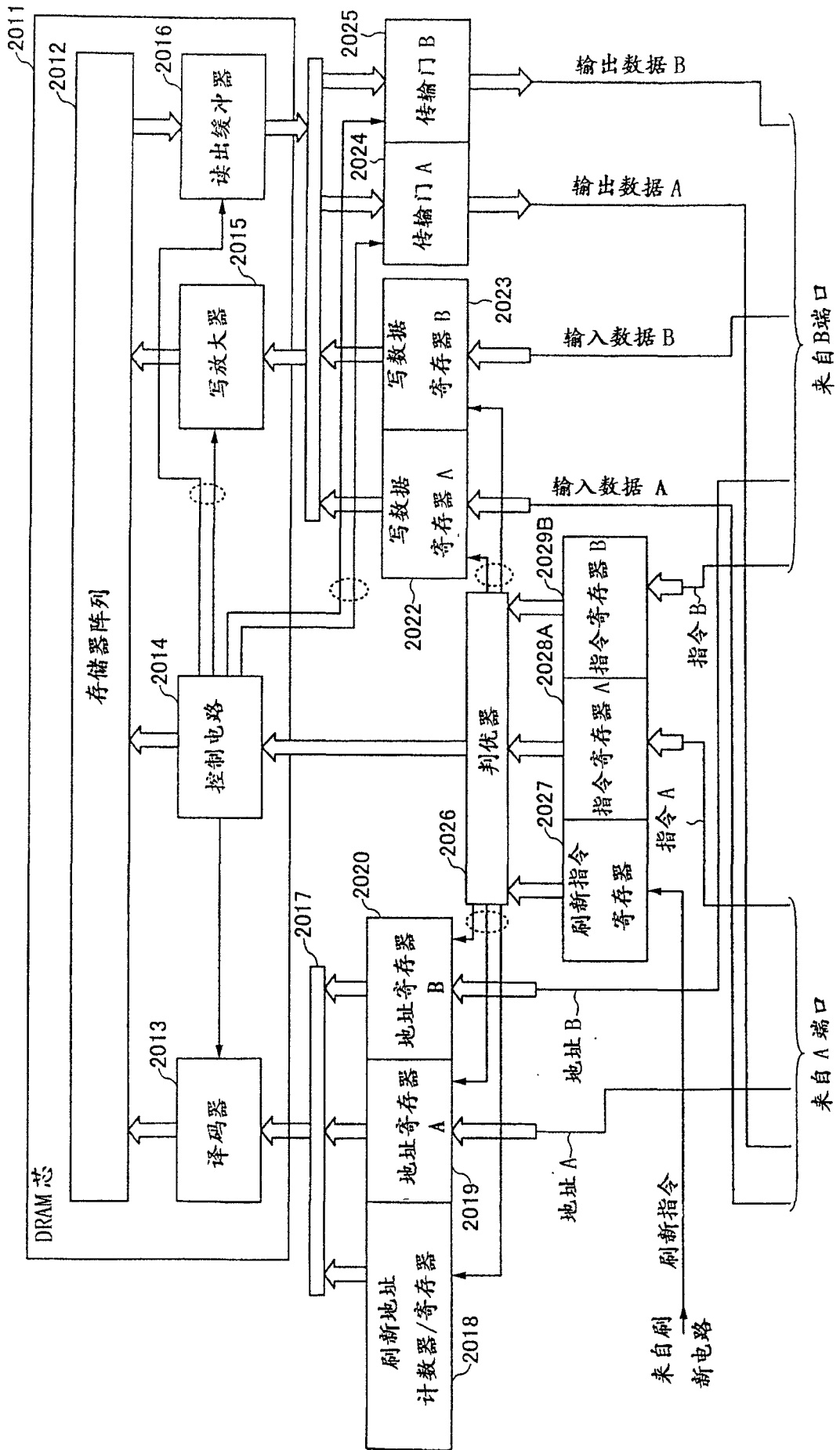


图 43A

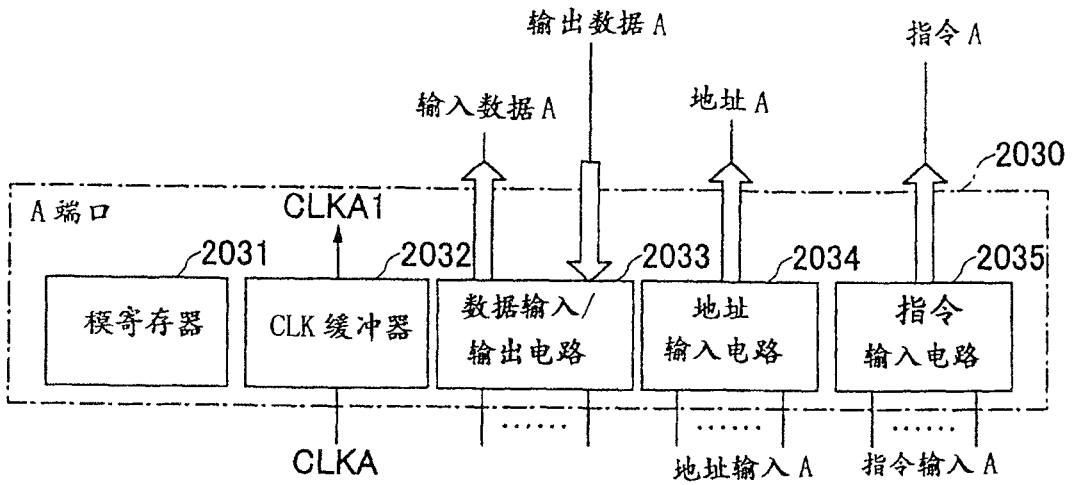


图 43B

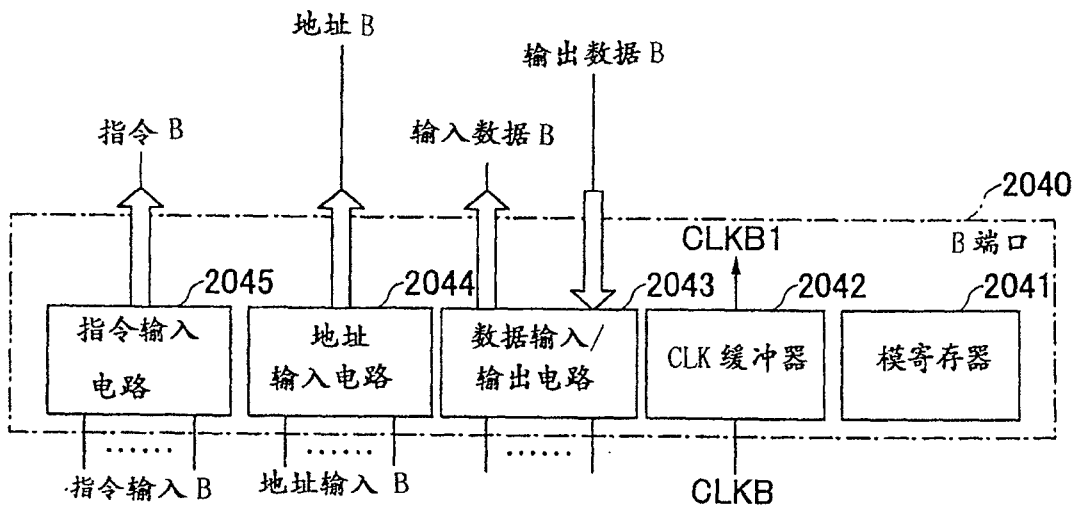


图 43C

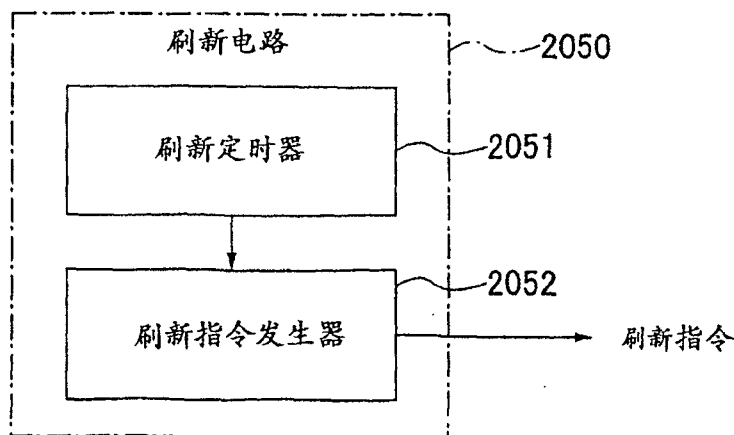


图 44

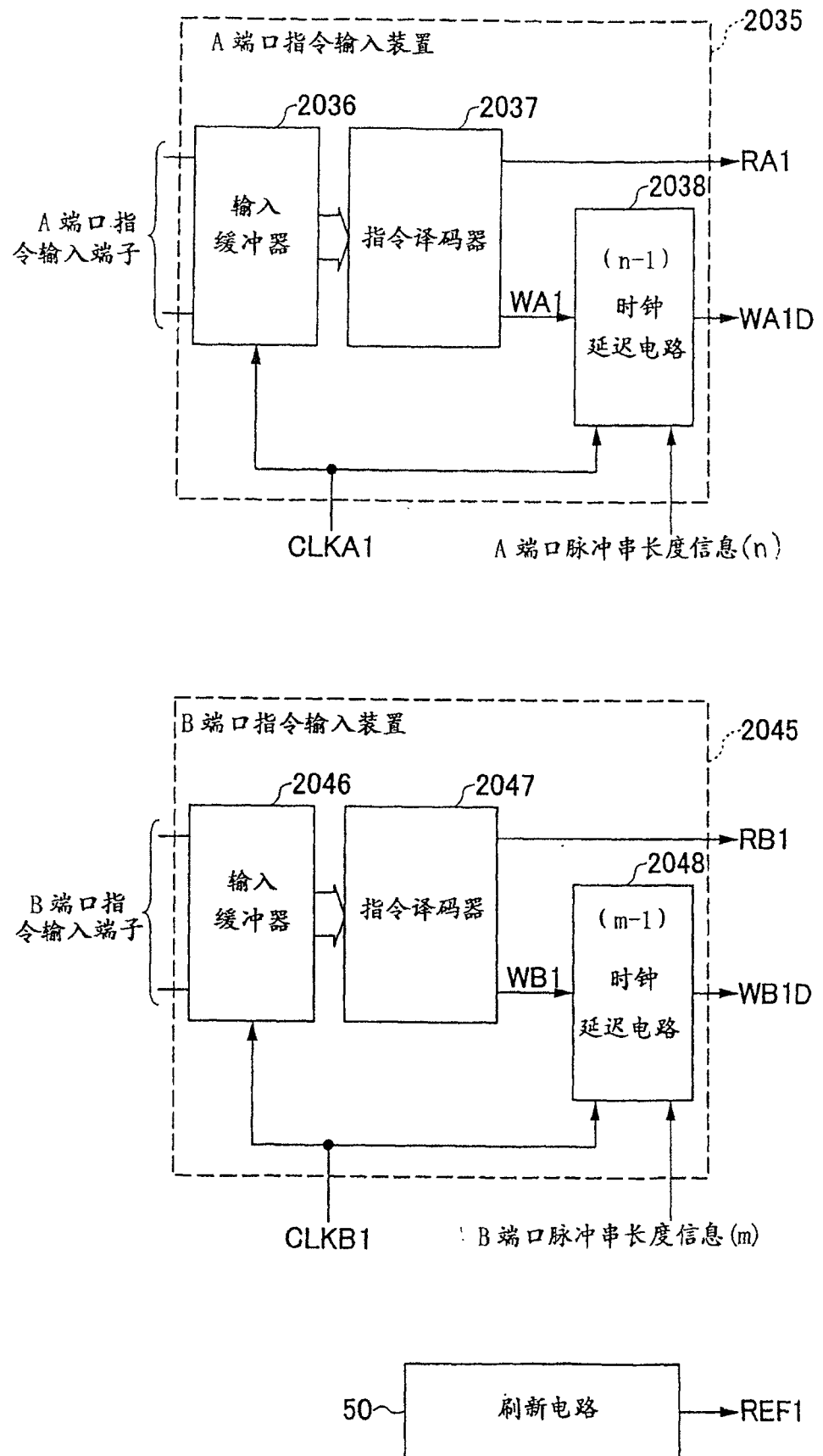


图 45

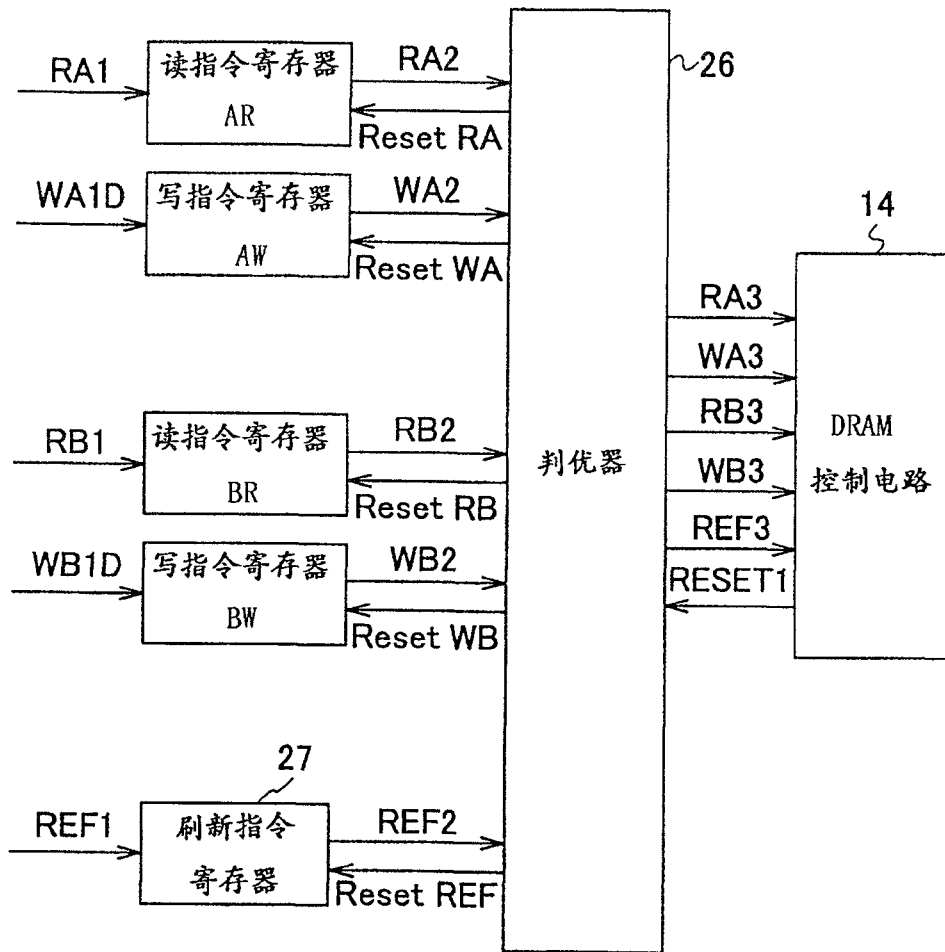


图 46

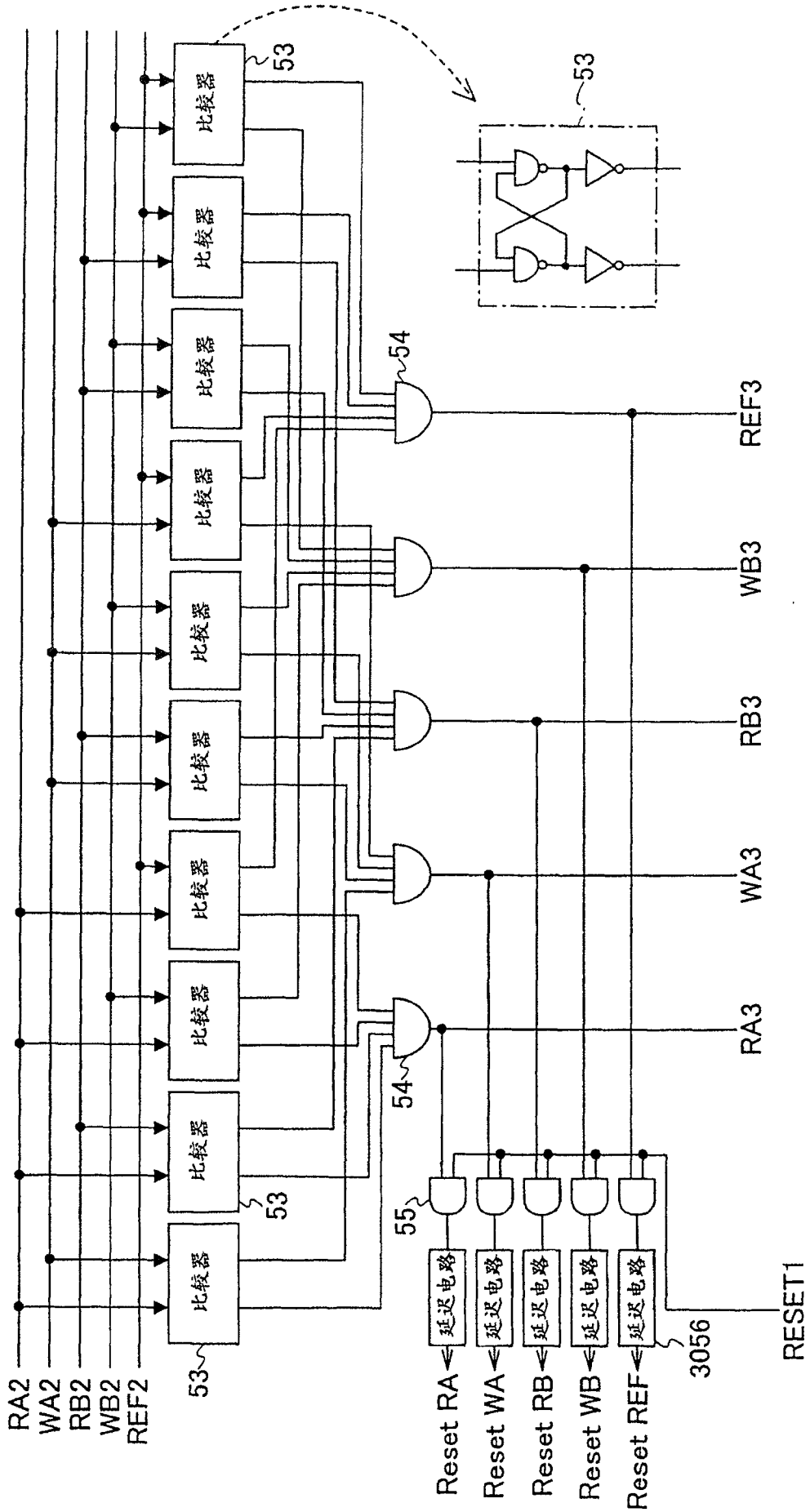


图 47

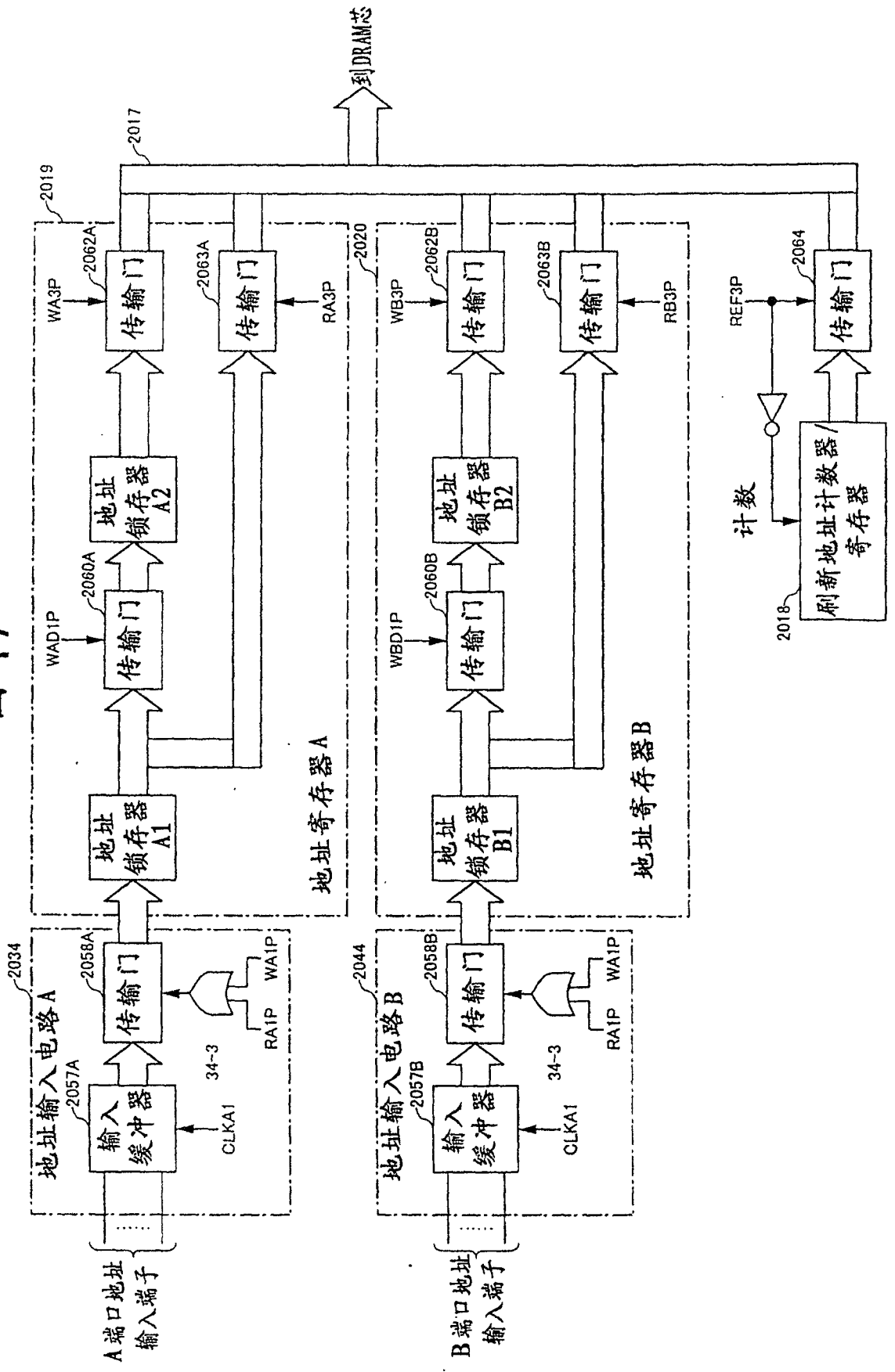


图 48

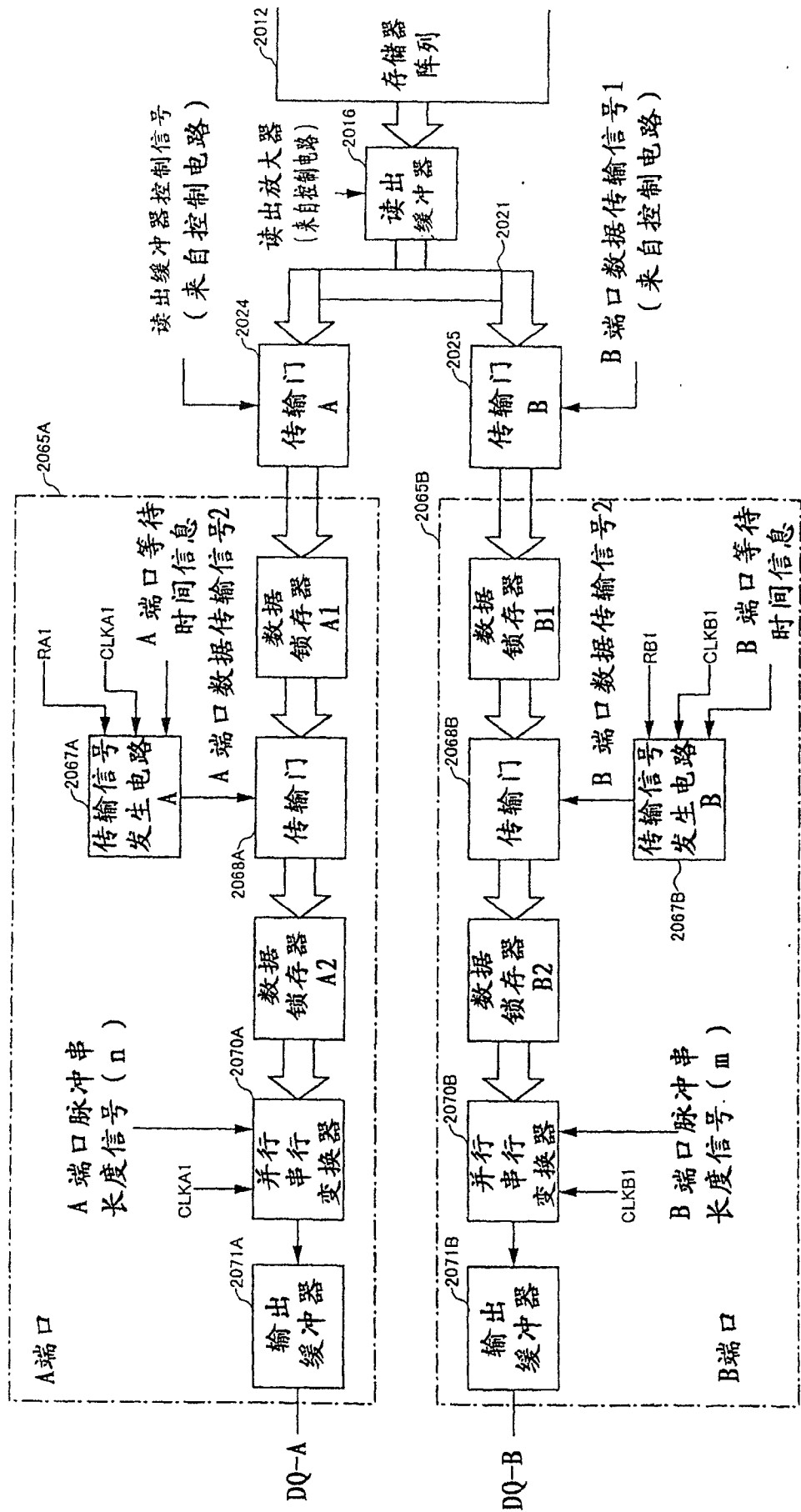


图49

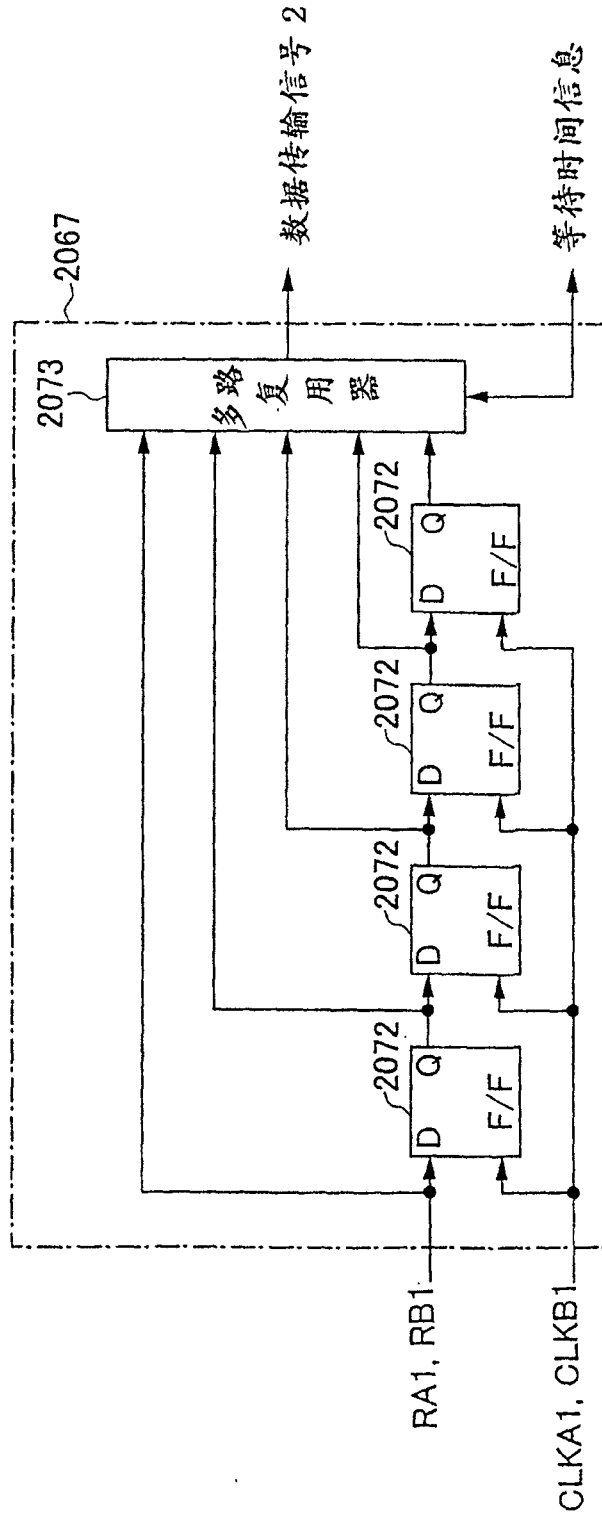
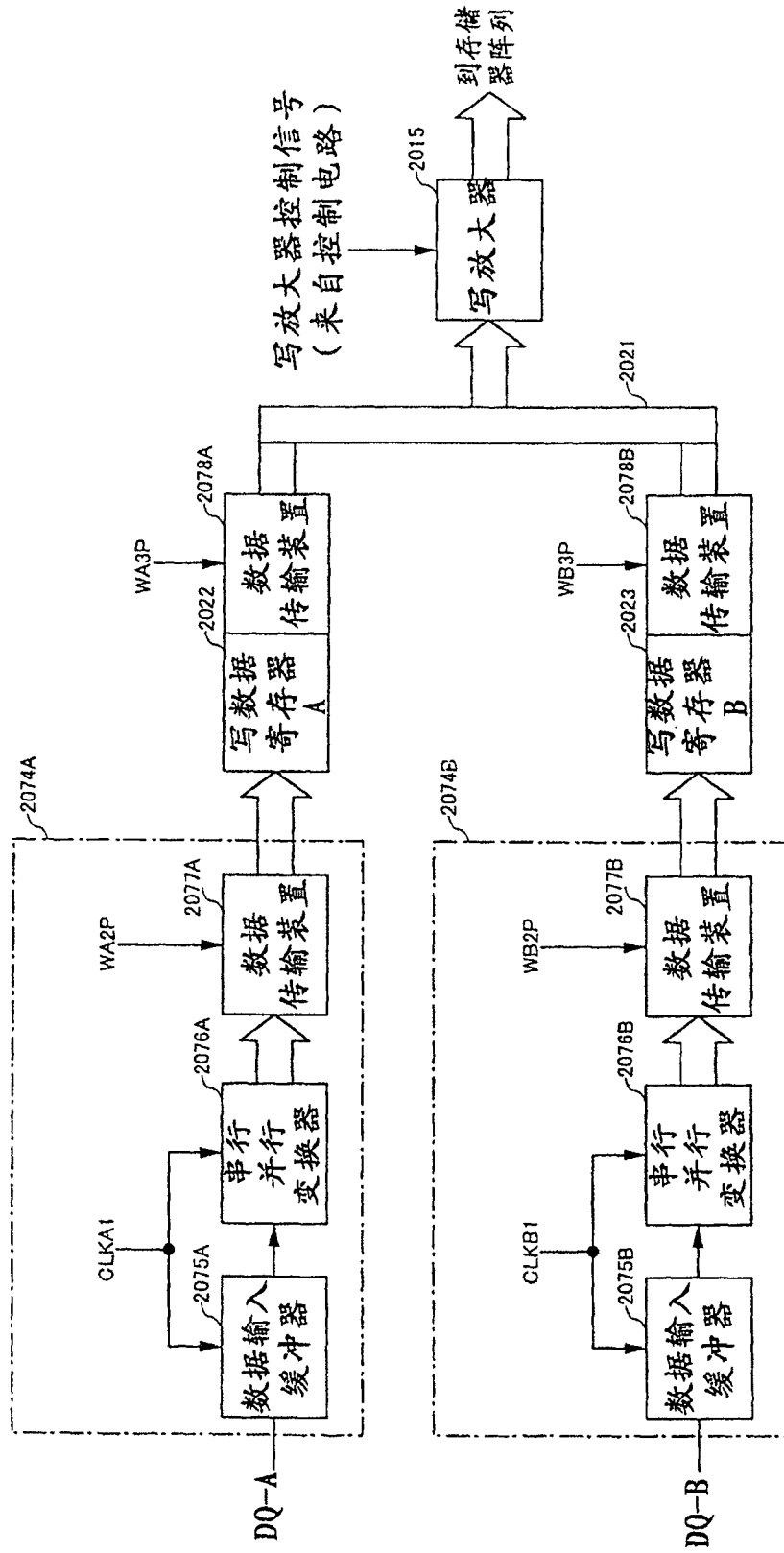


图 50



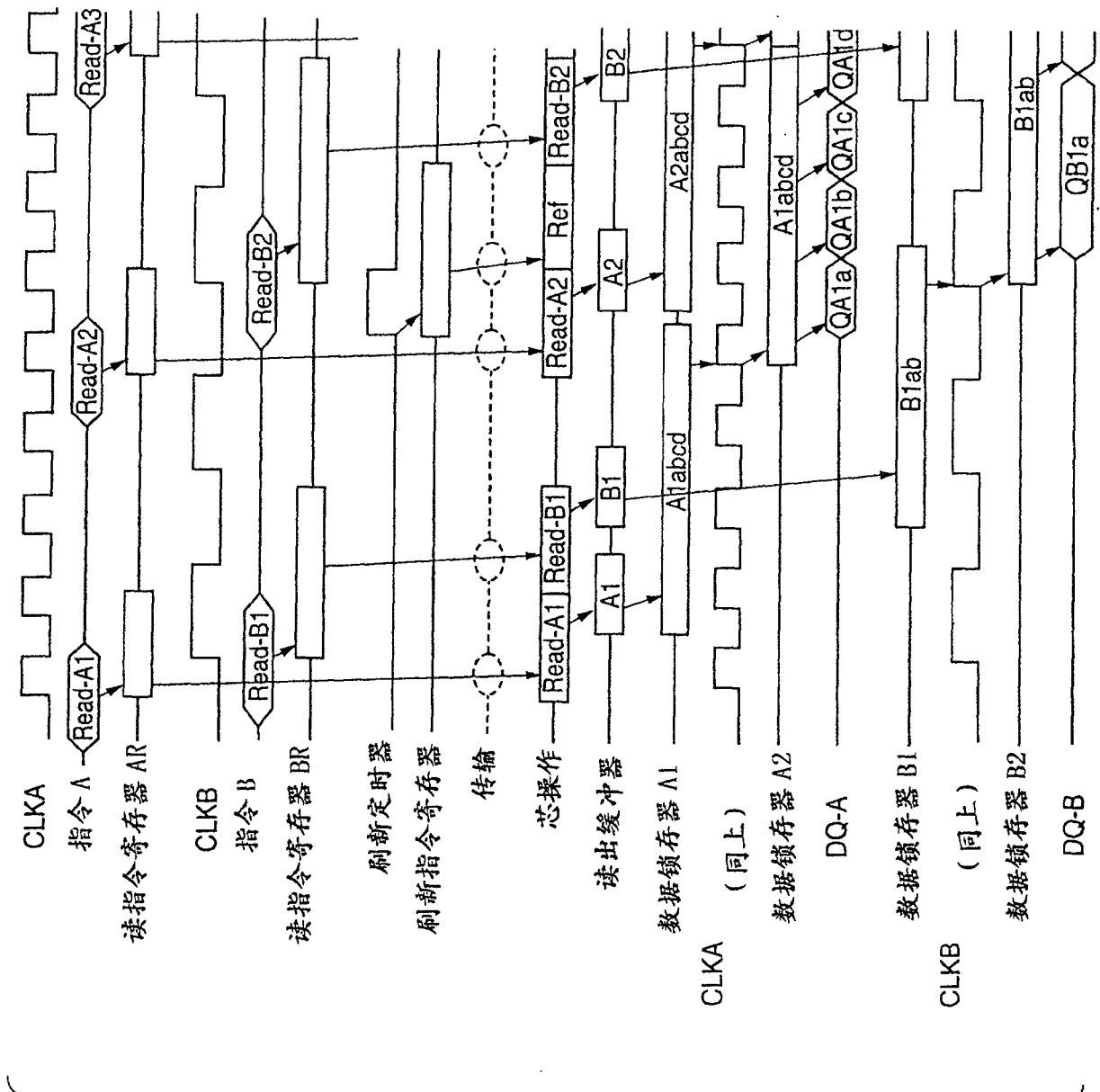
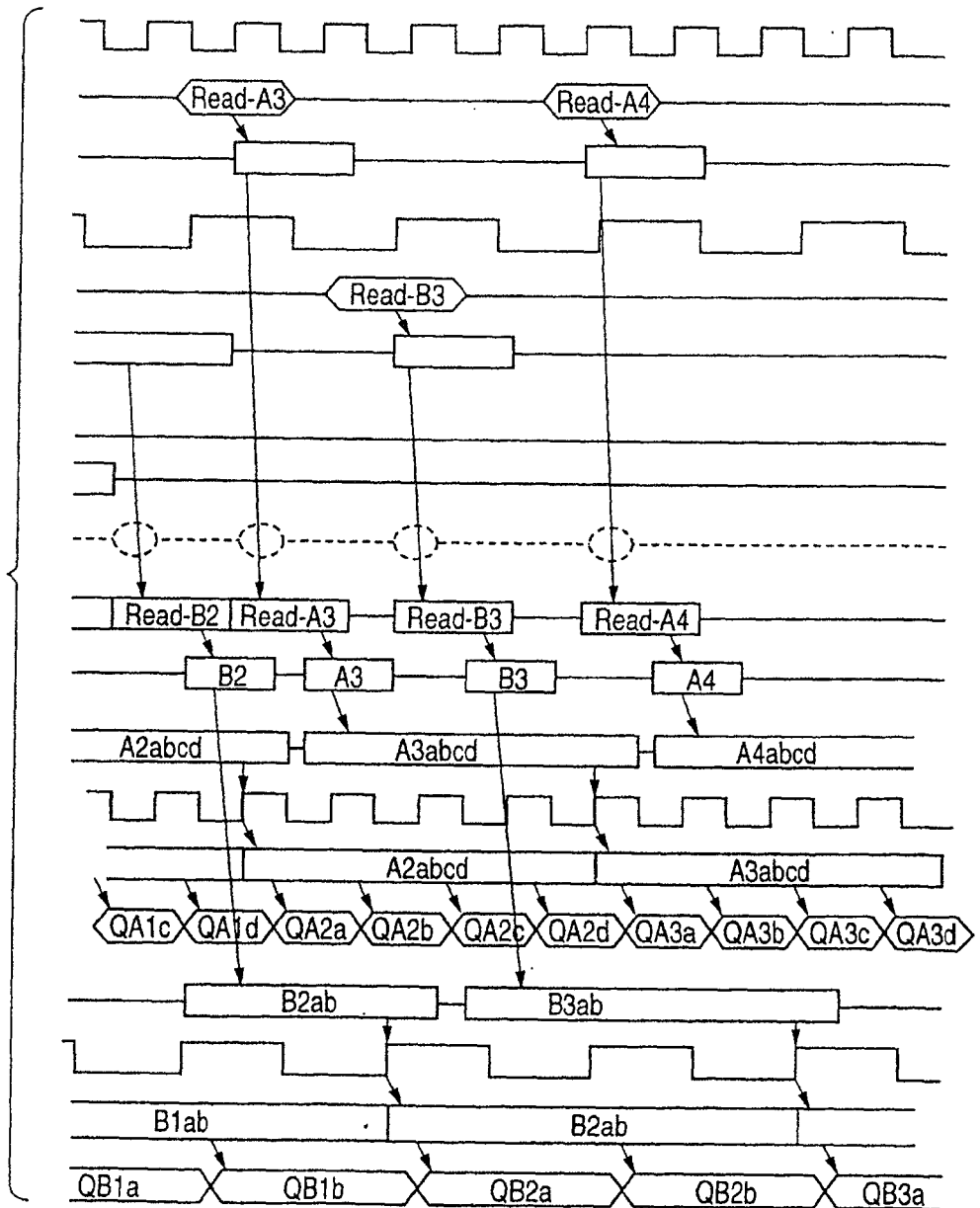


图 51

图 52



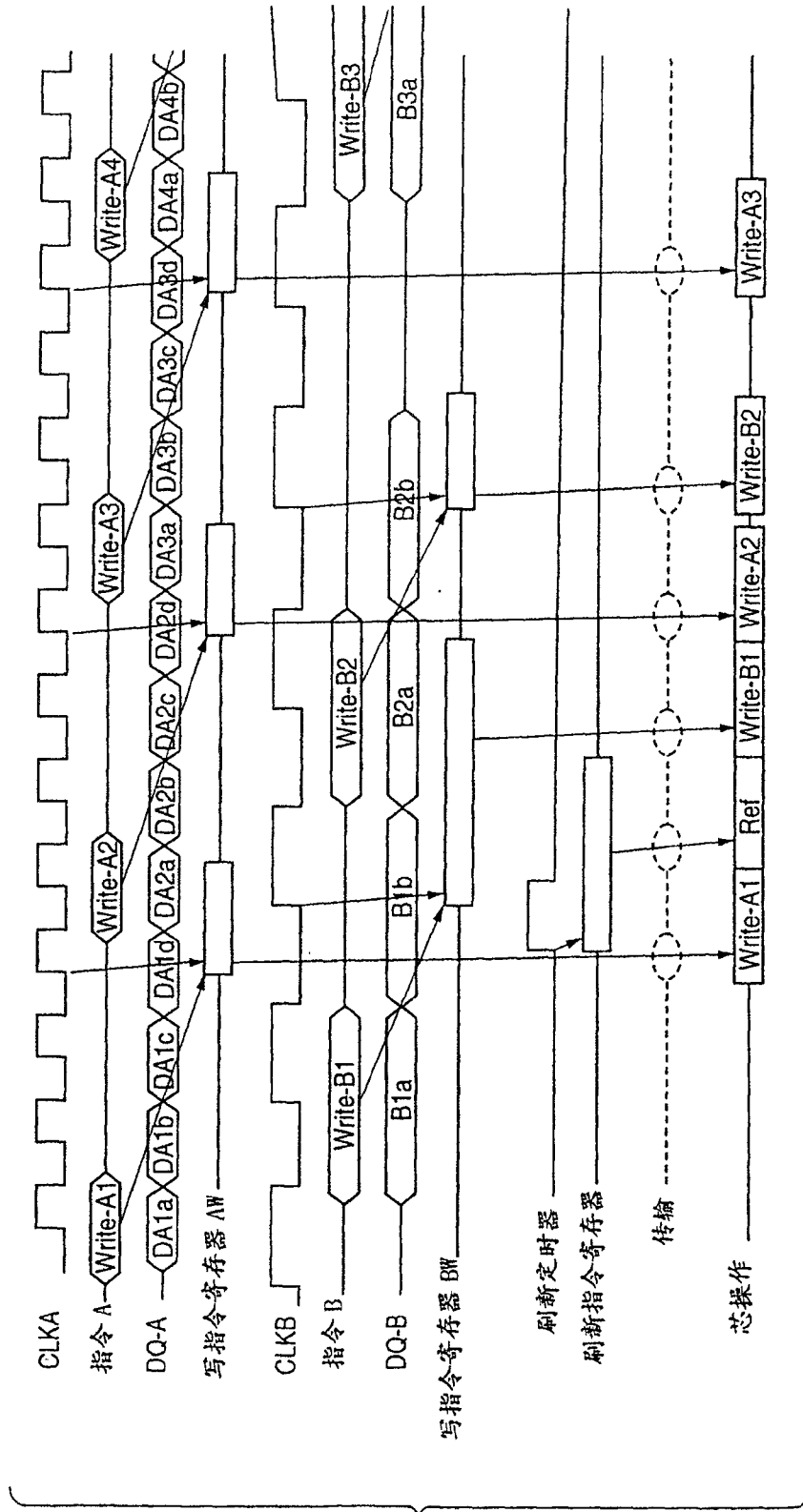


图 53

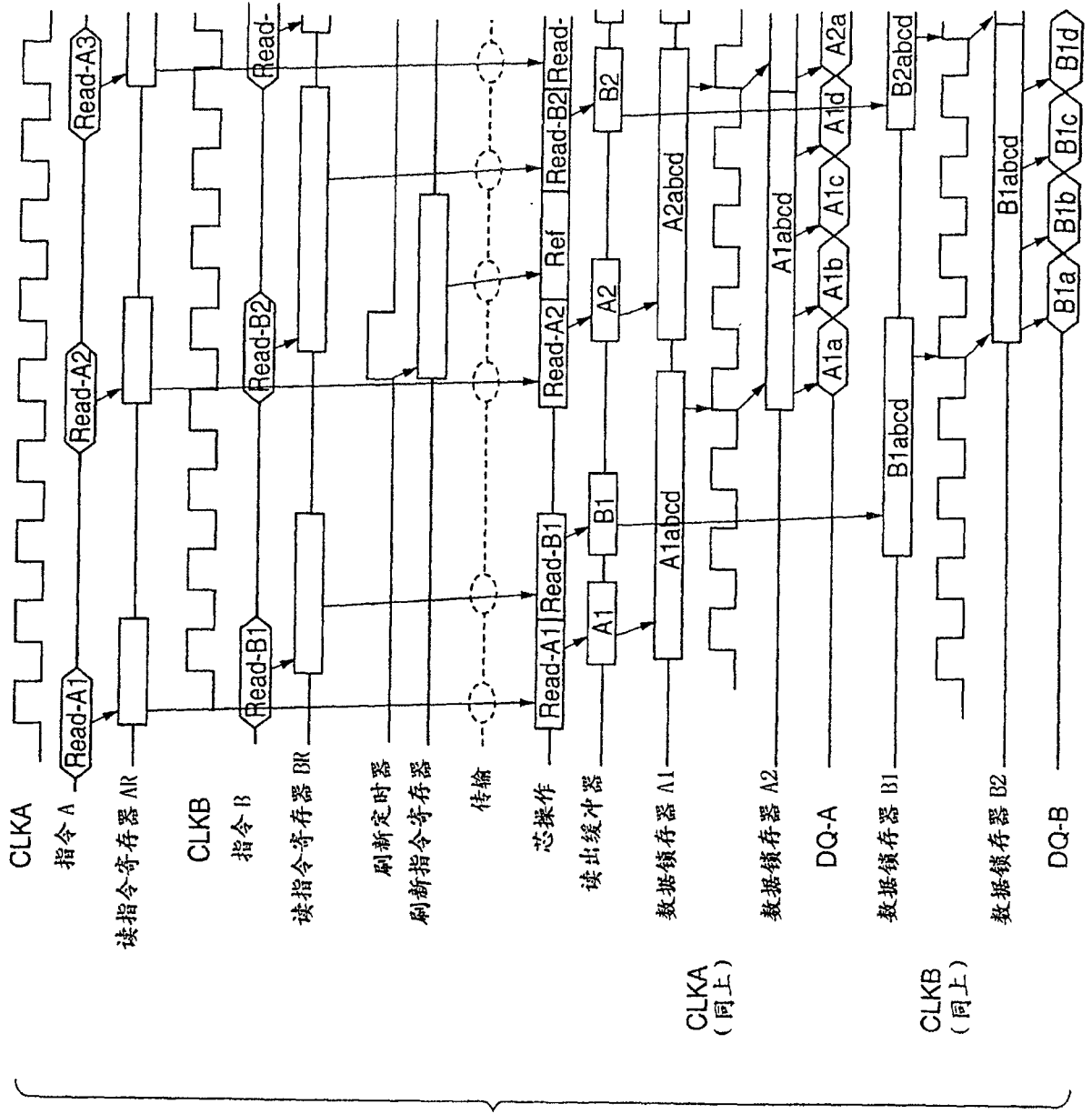
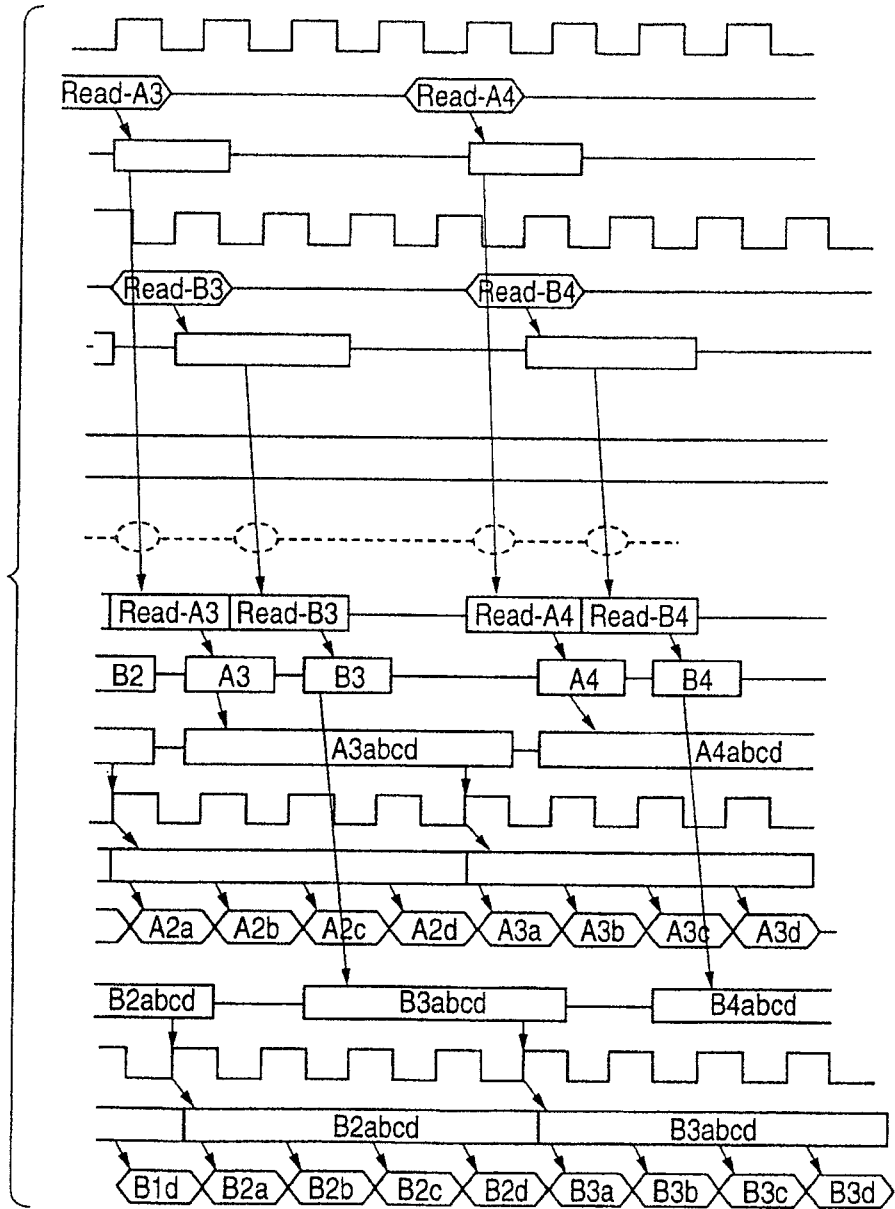


图 54

图 55



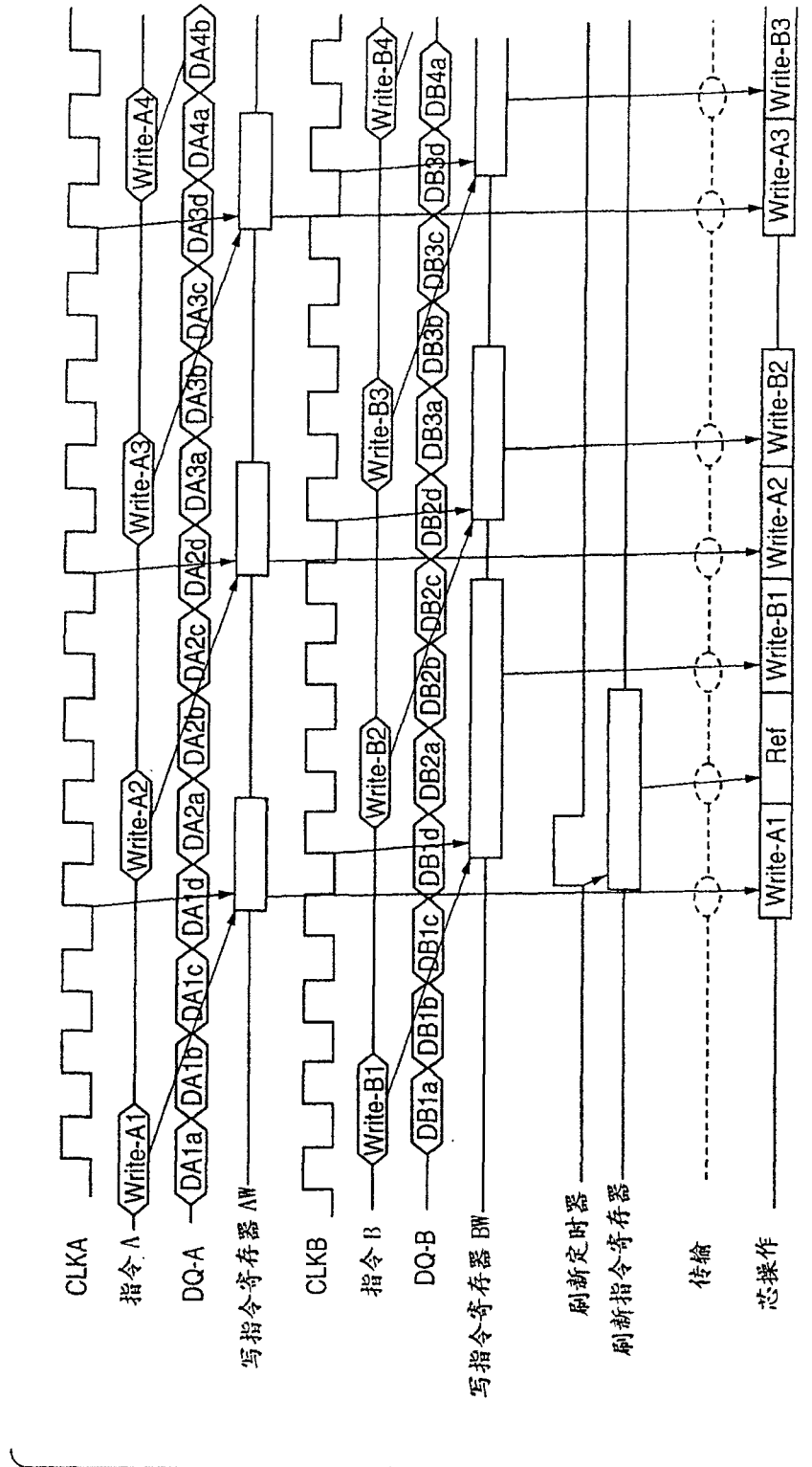


图 56

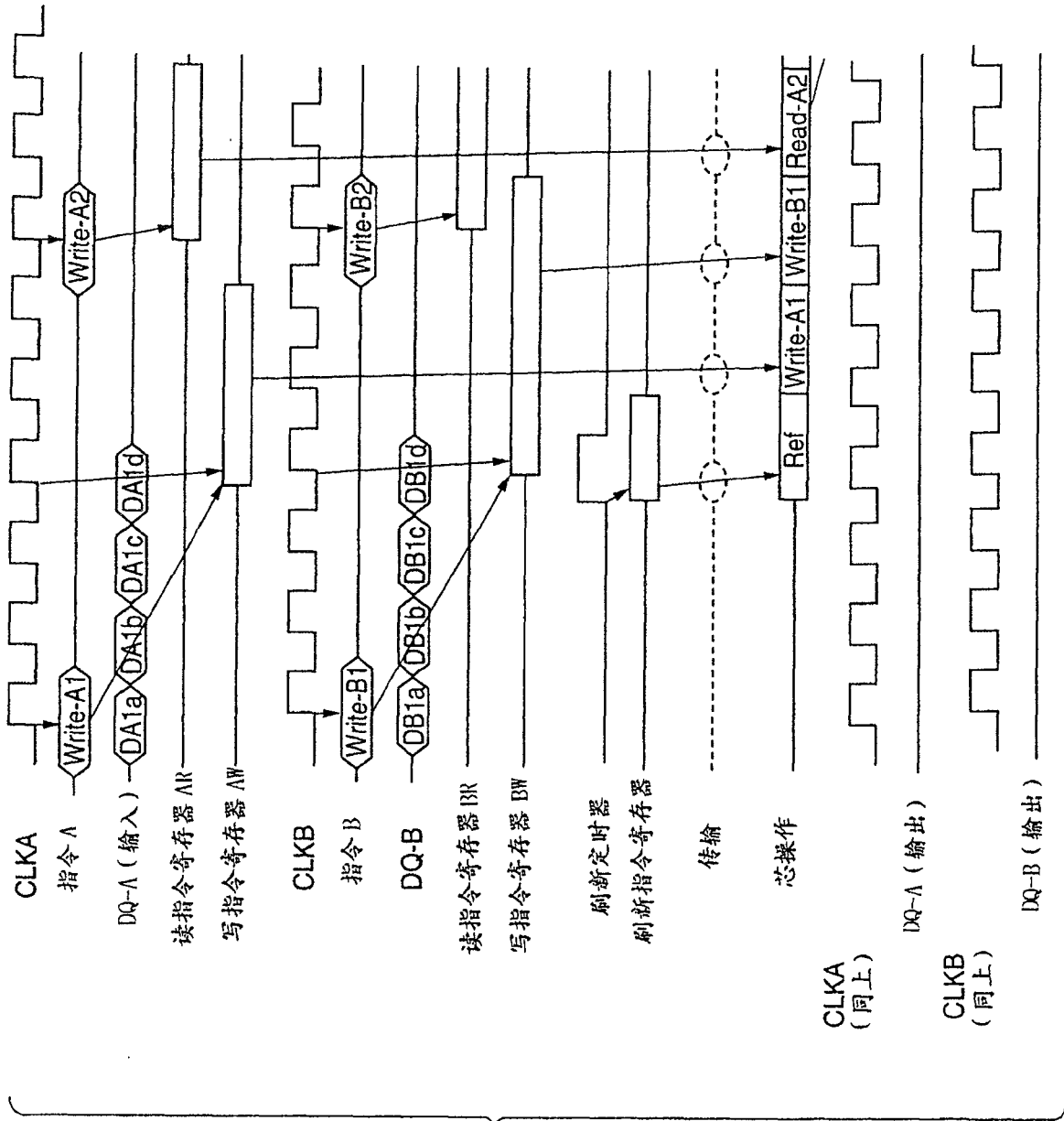


图 57

图 58

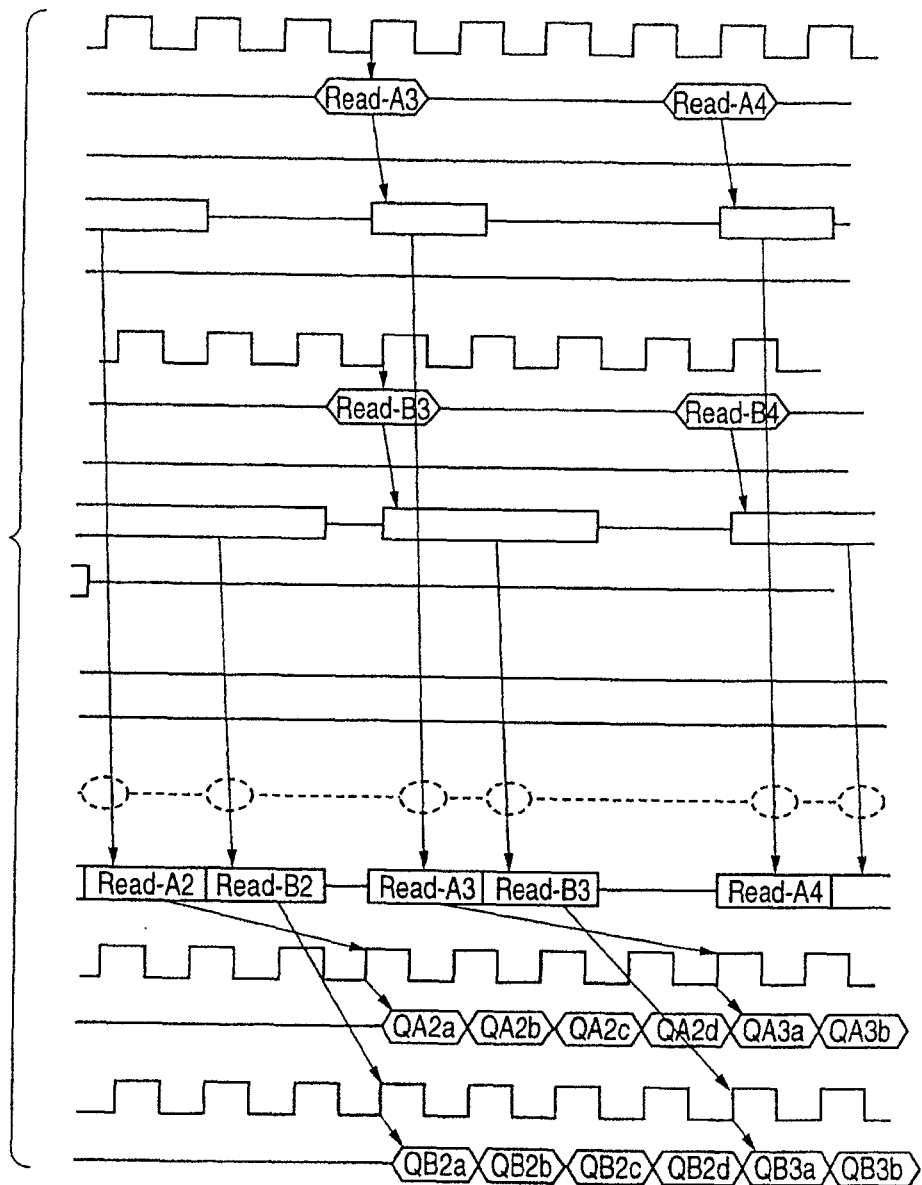


图 59A

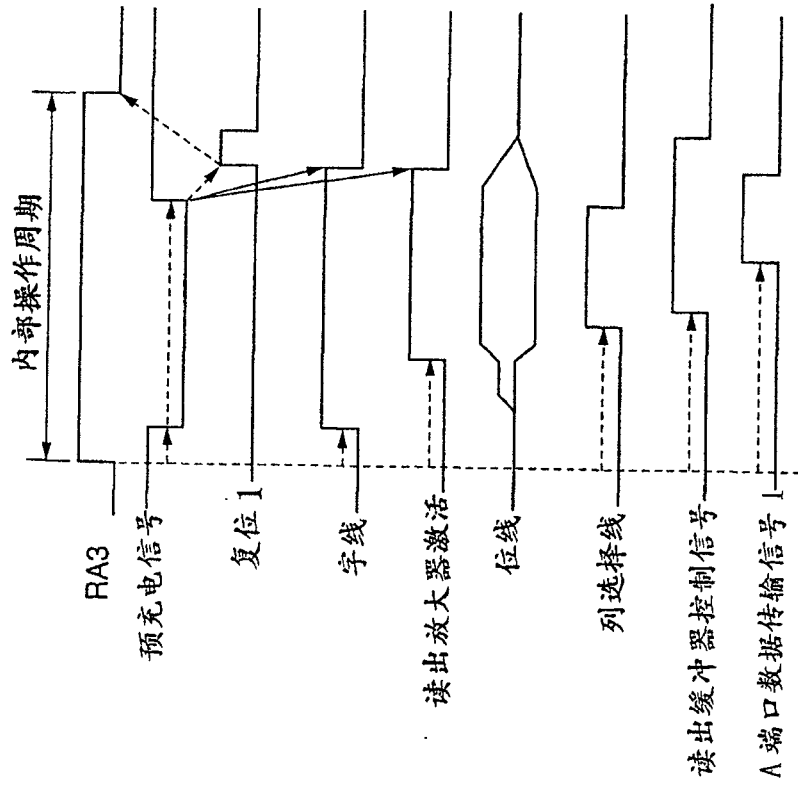


图 59B

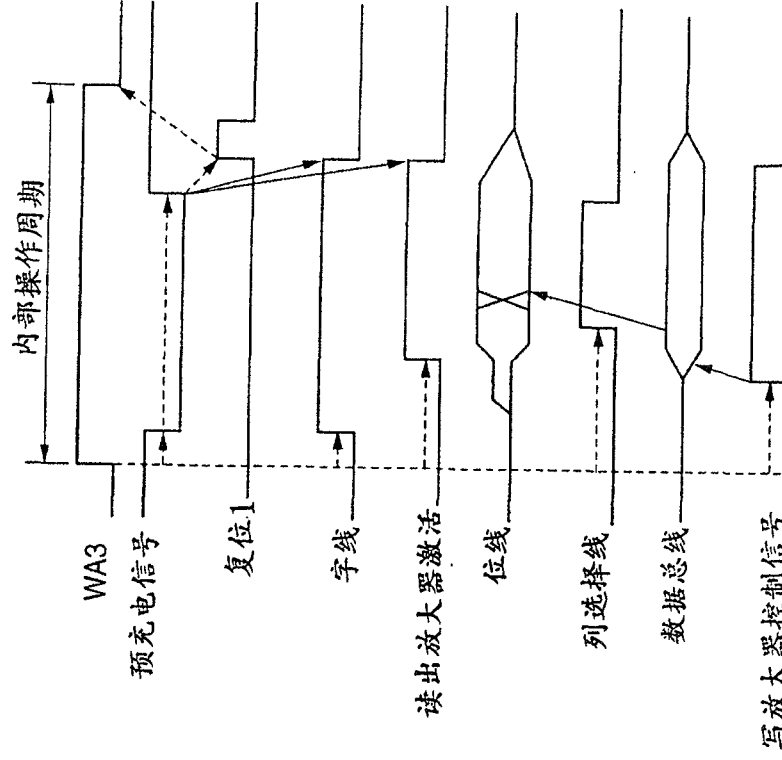


图60

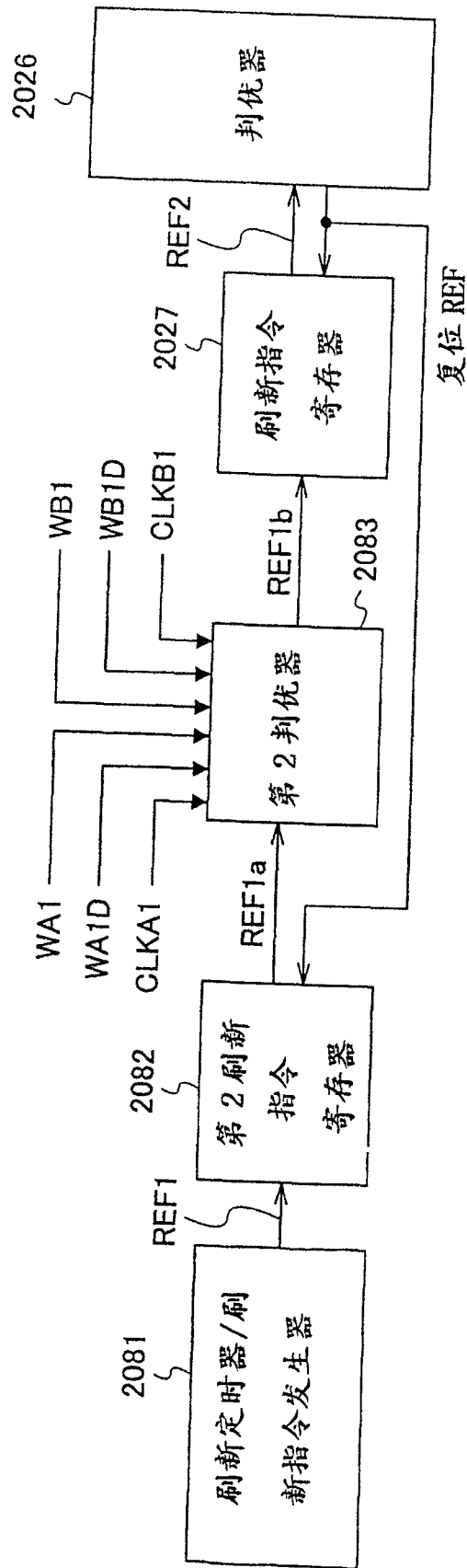
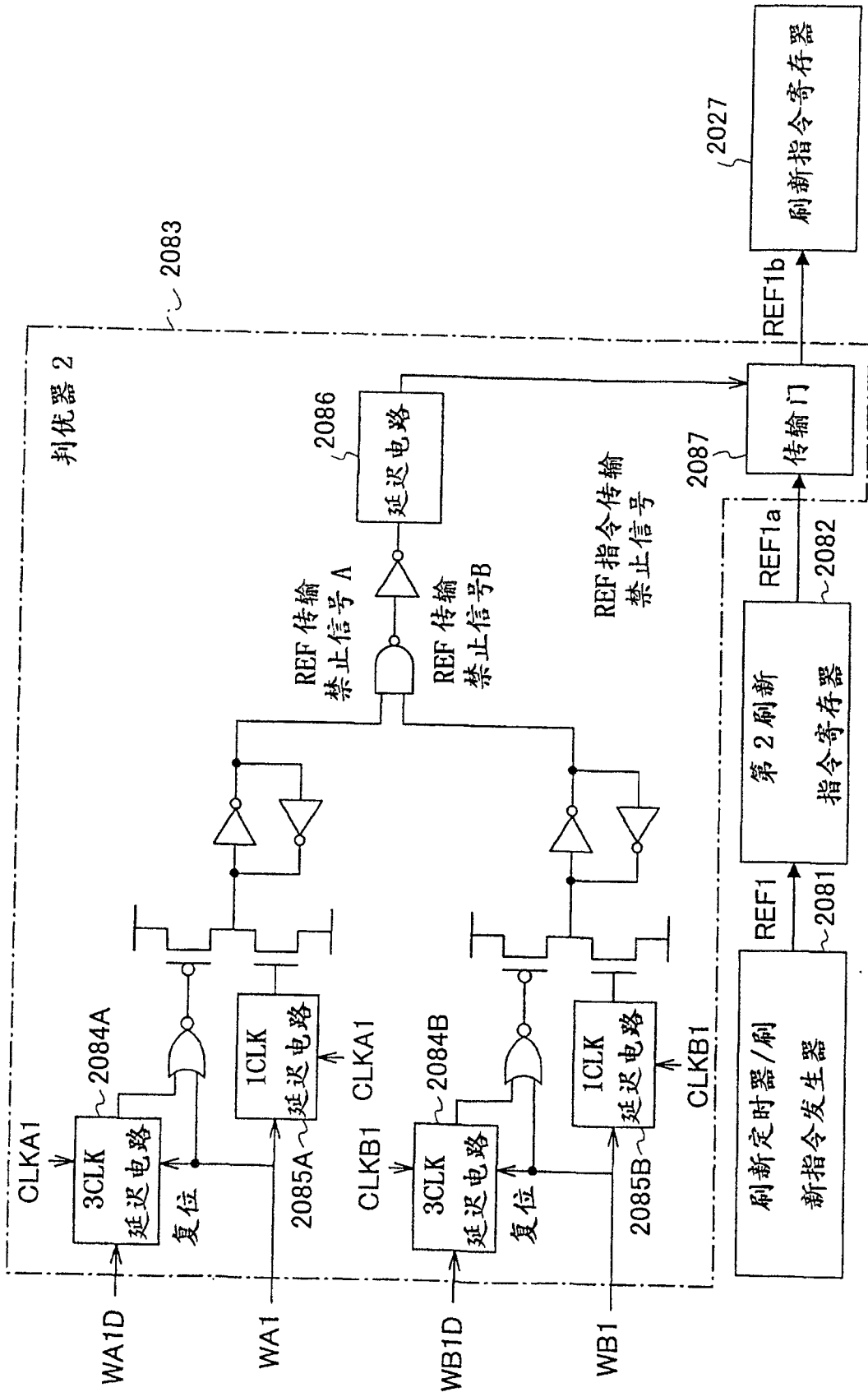


图61



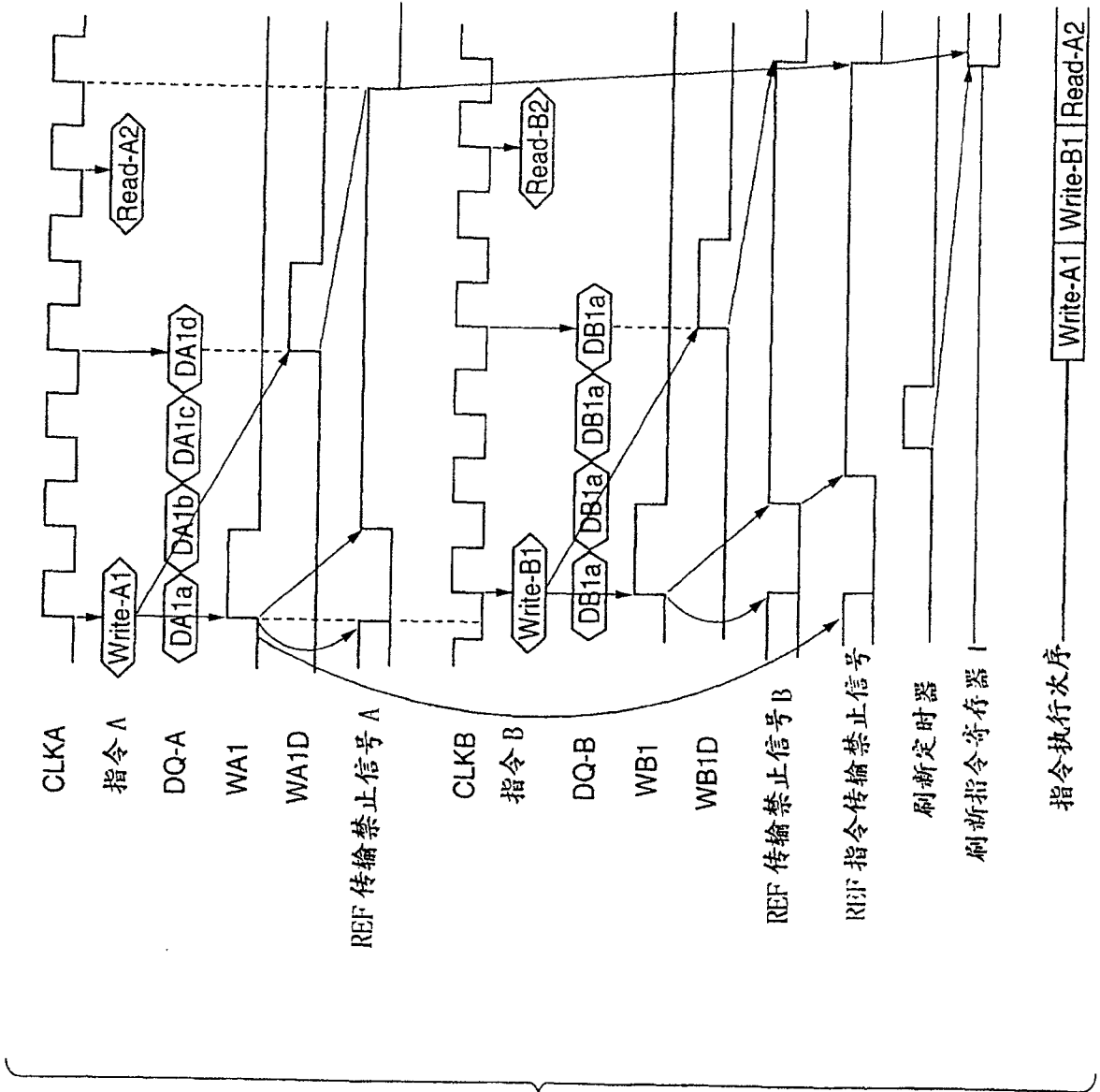
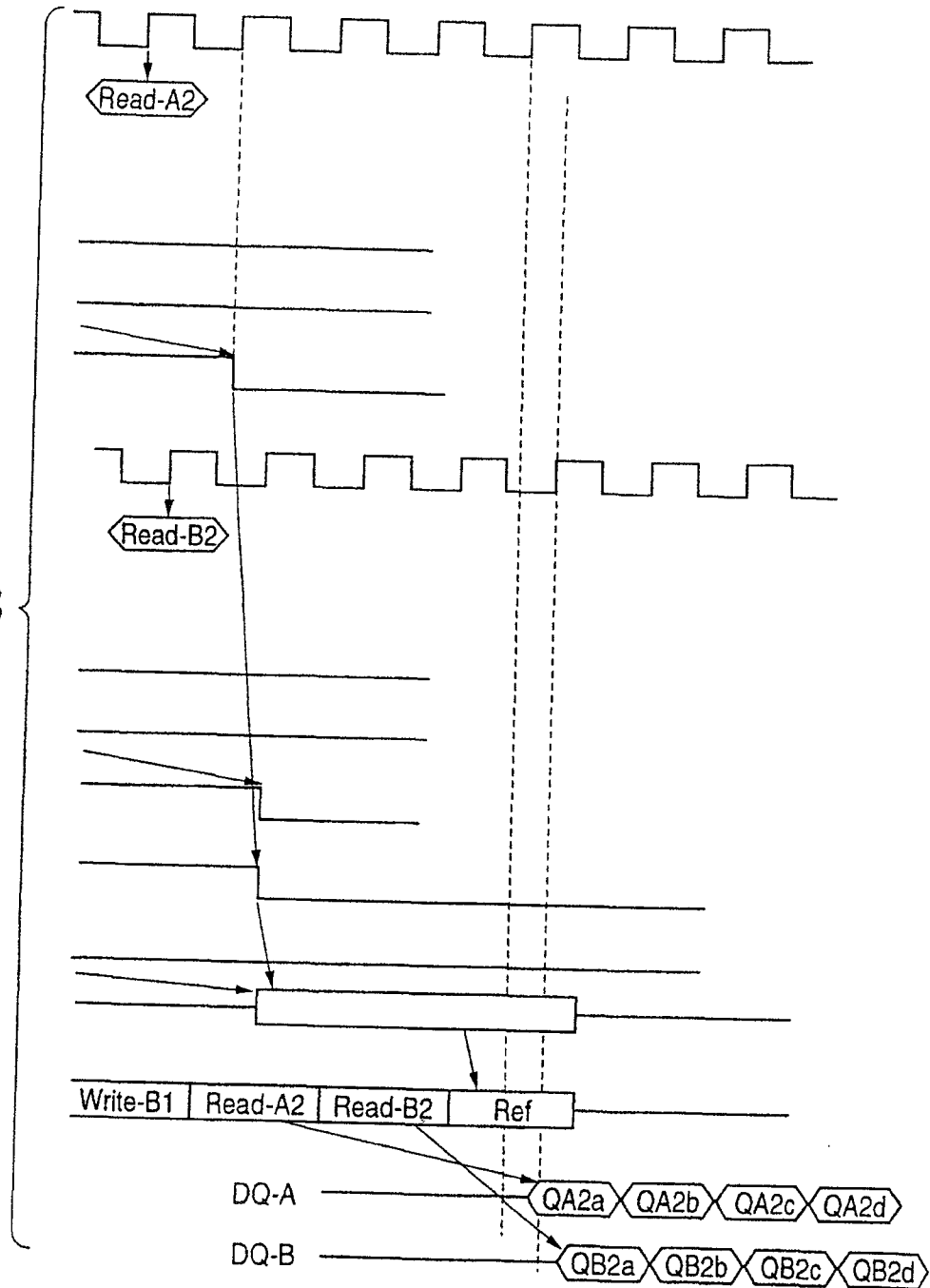


图 62

图 63



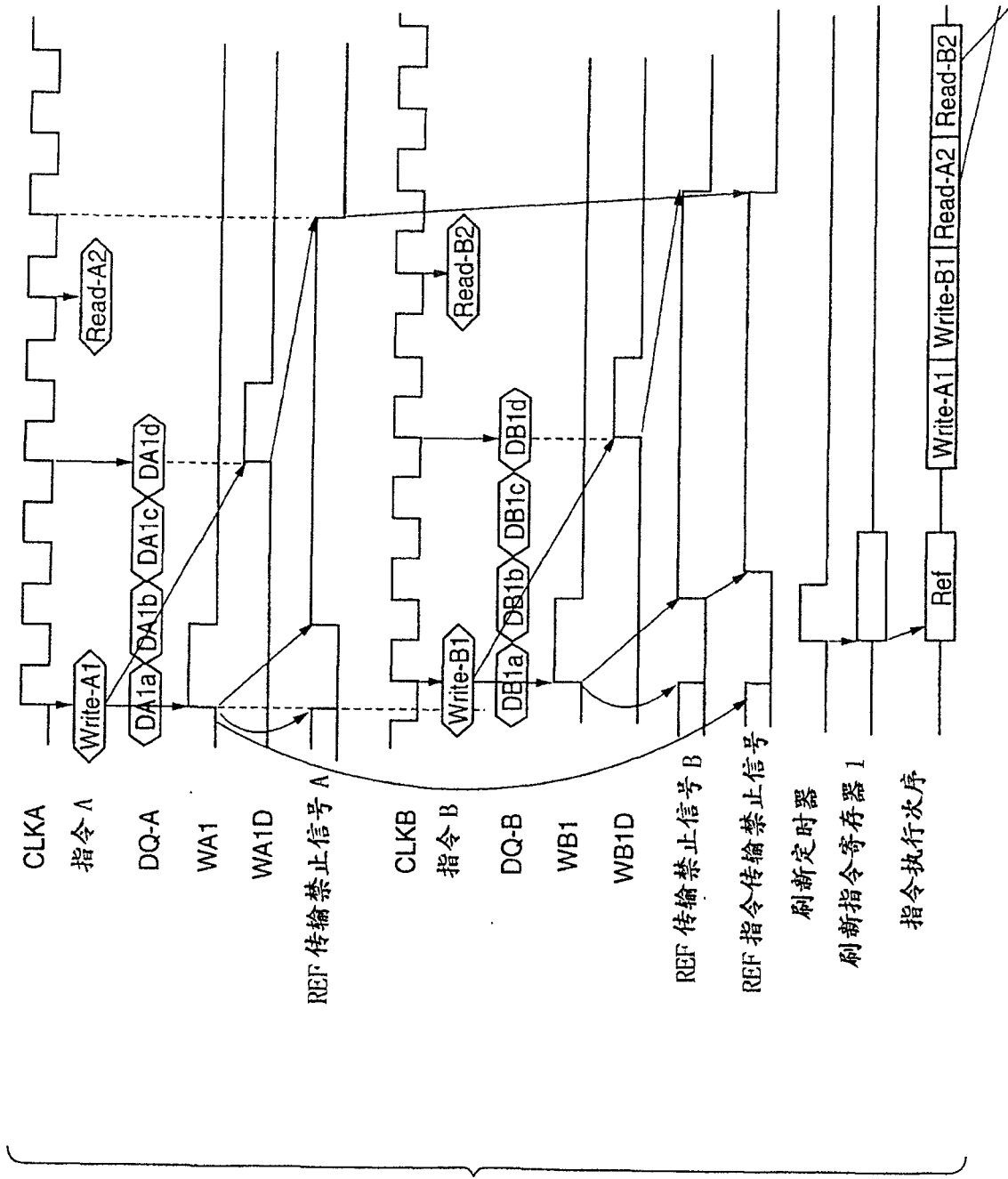
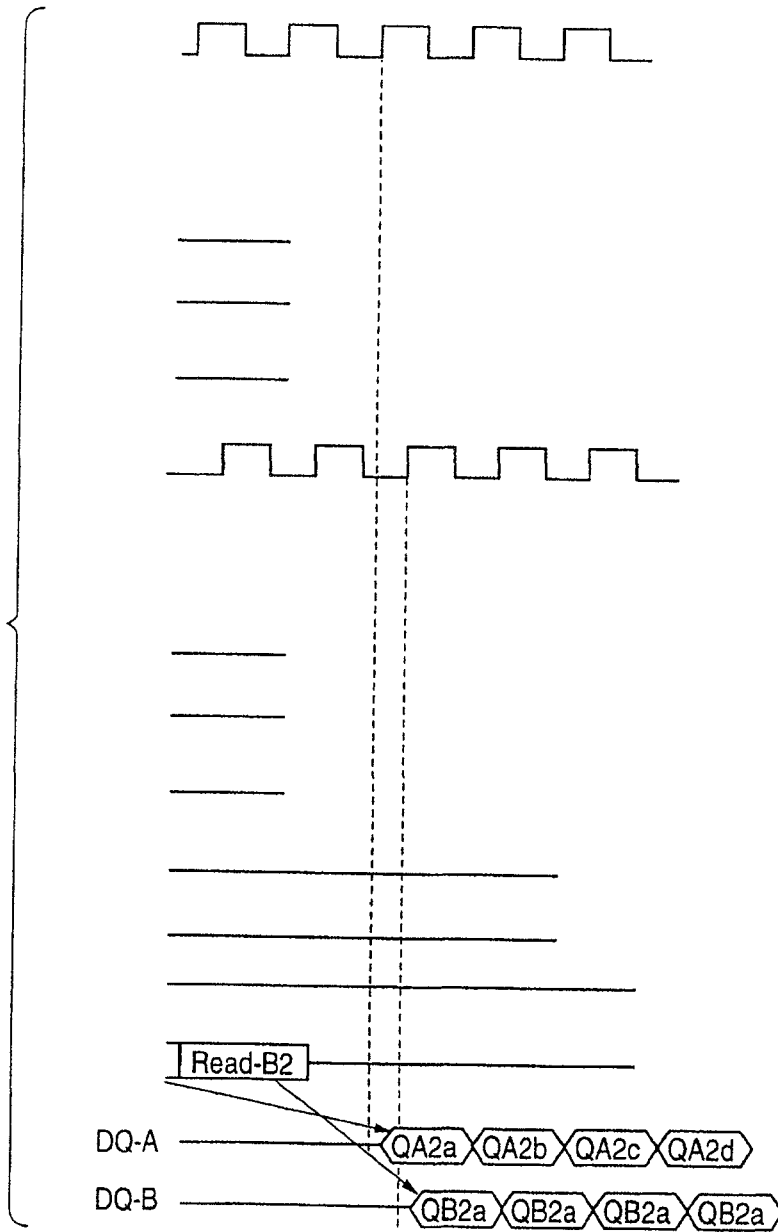


图 64

图 65



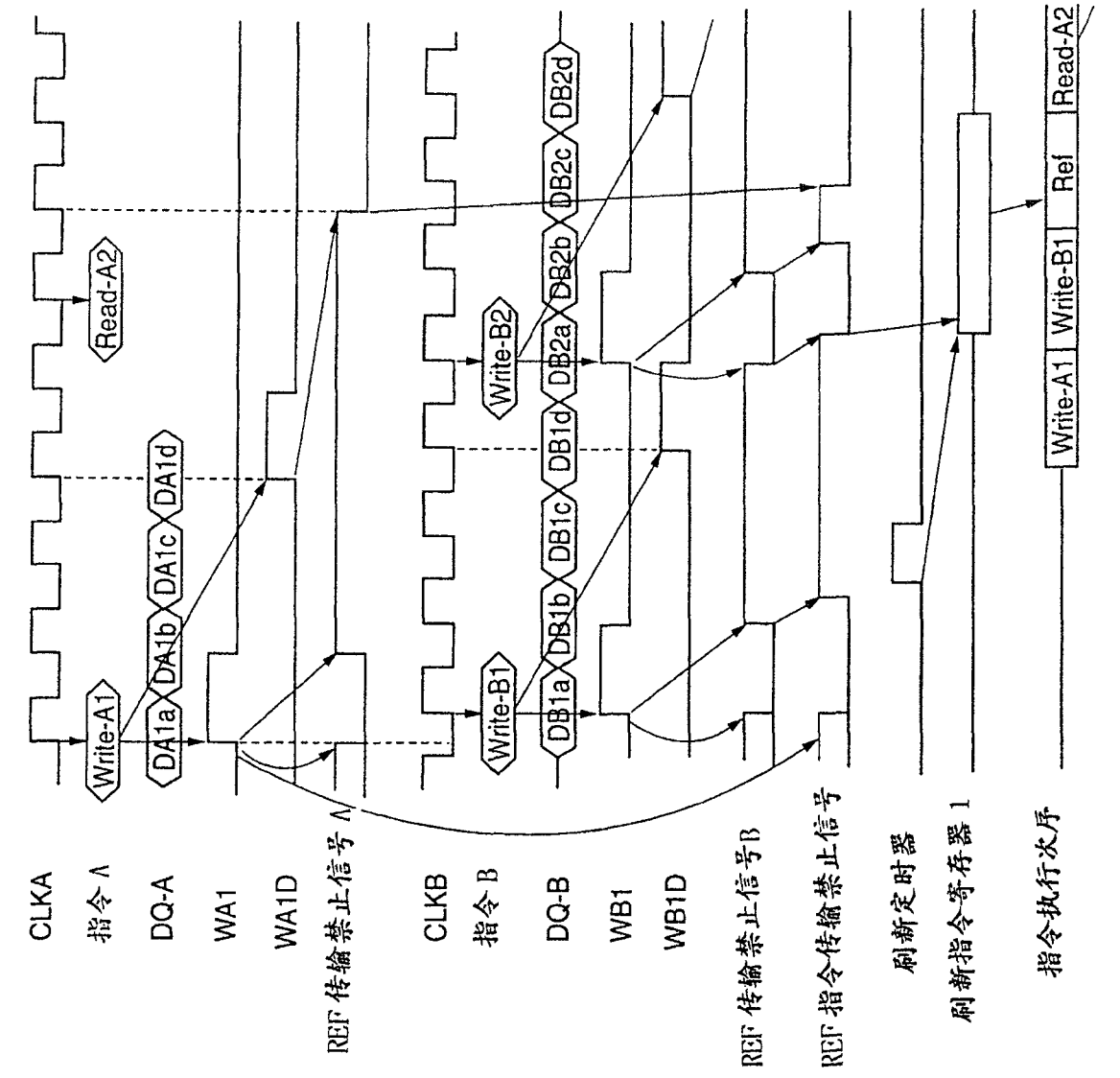
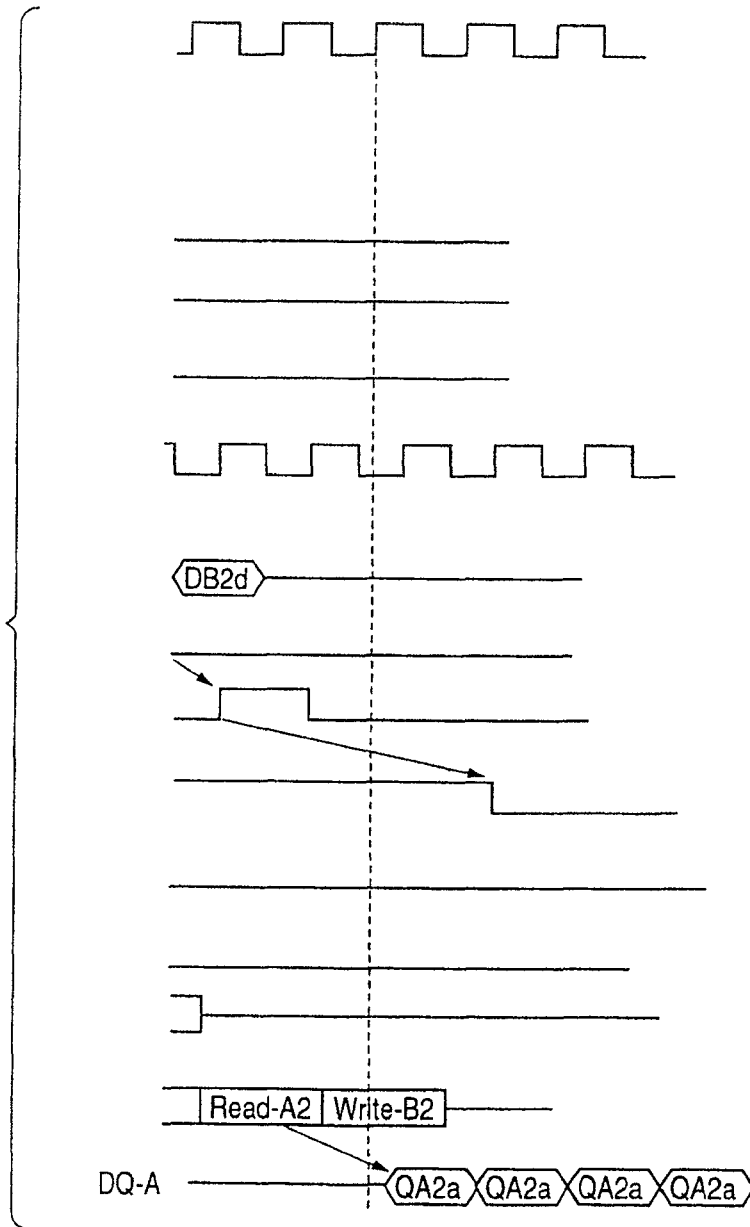


图 66

图 67



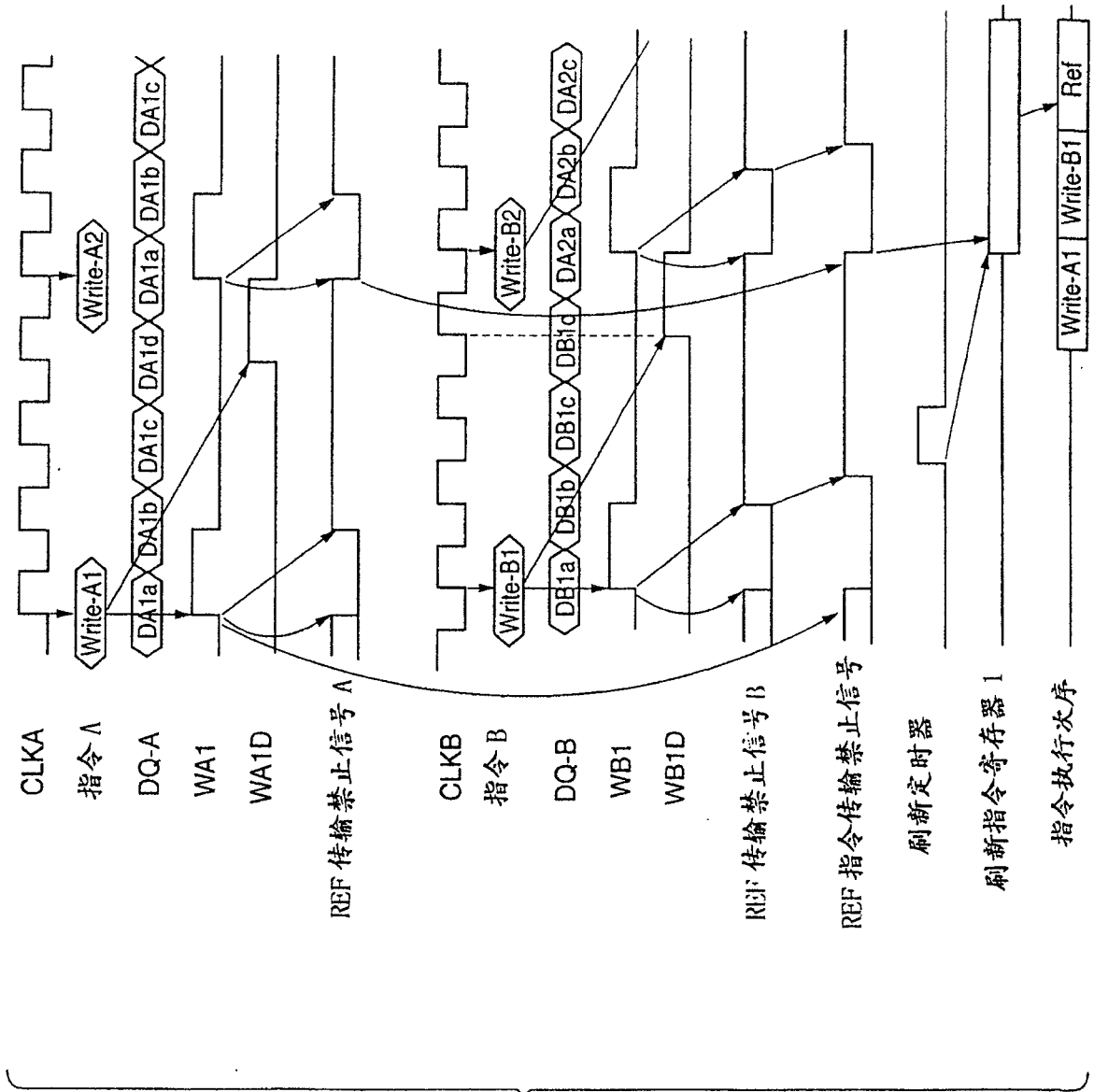
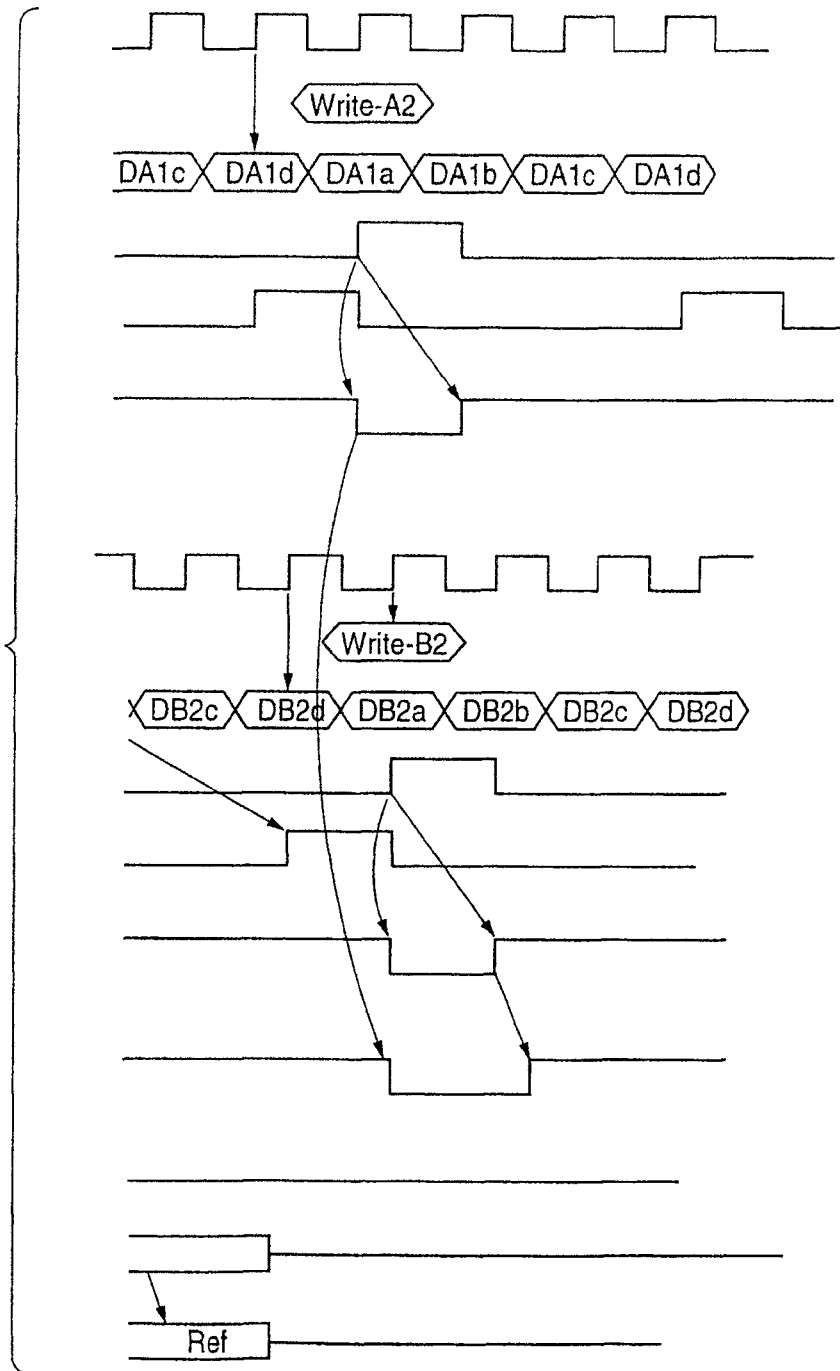


图 68

图 69



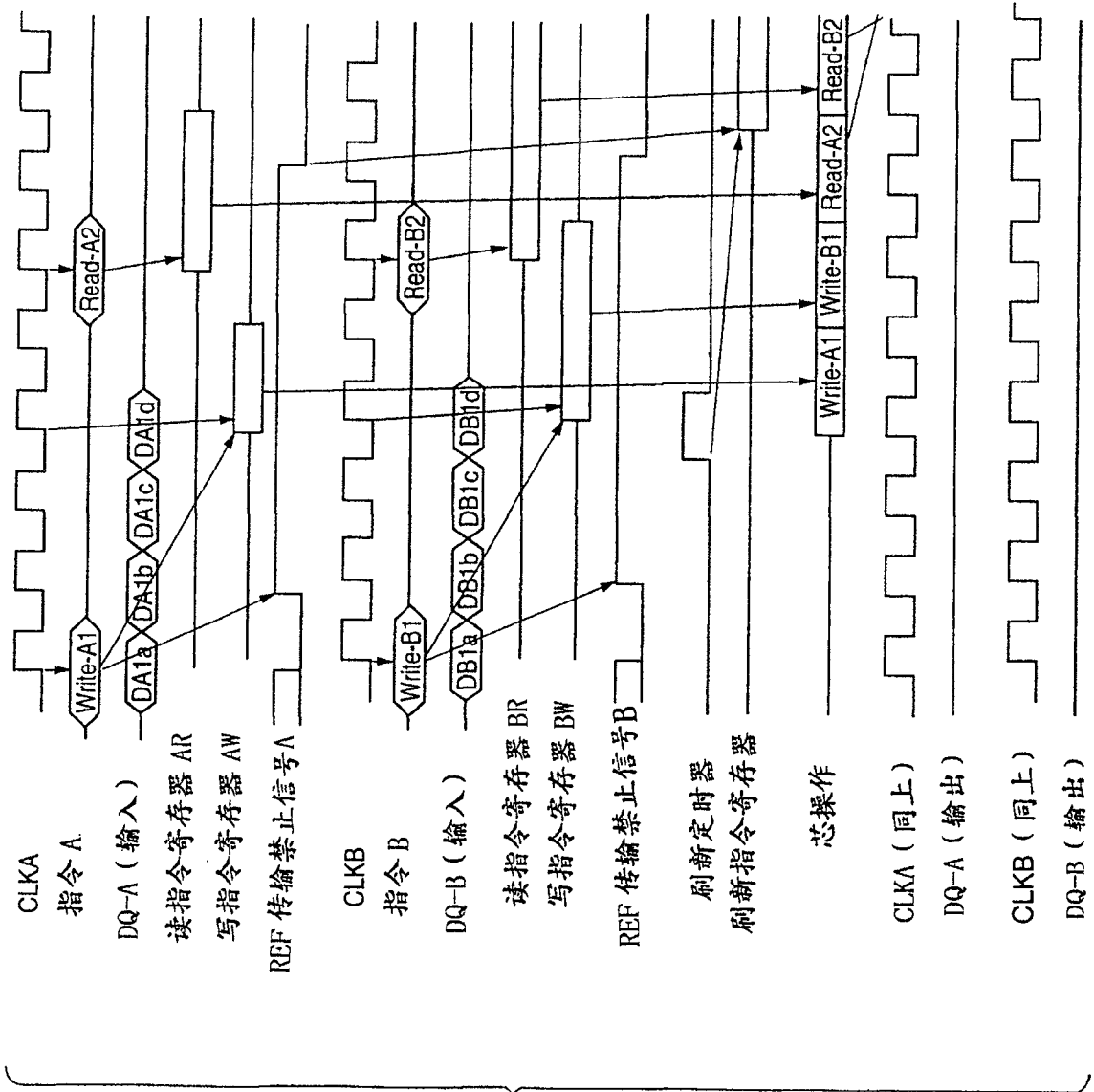
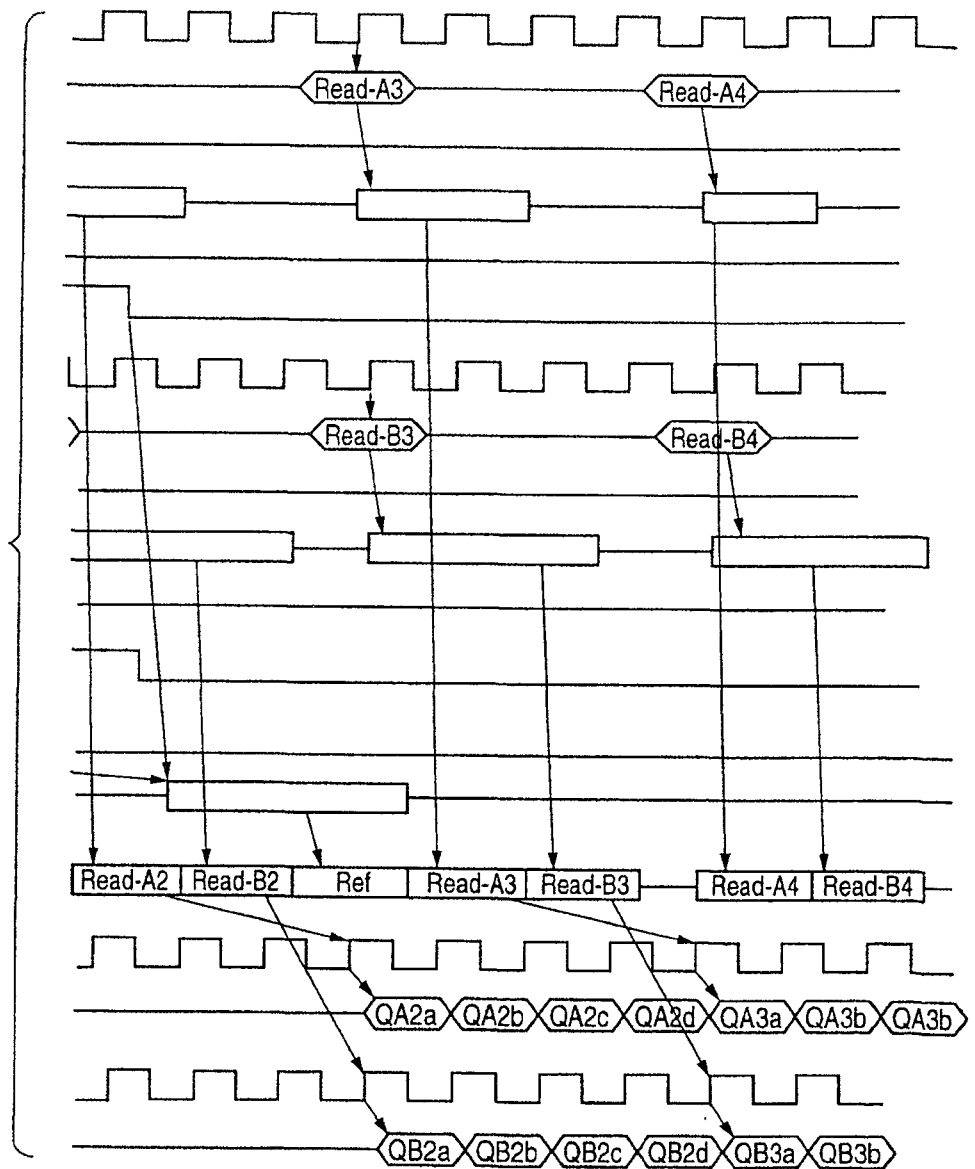


图 70

图 71



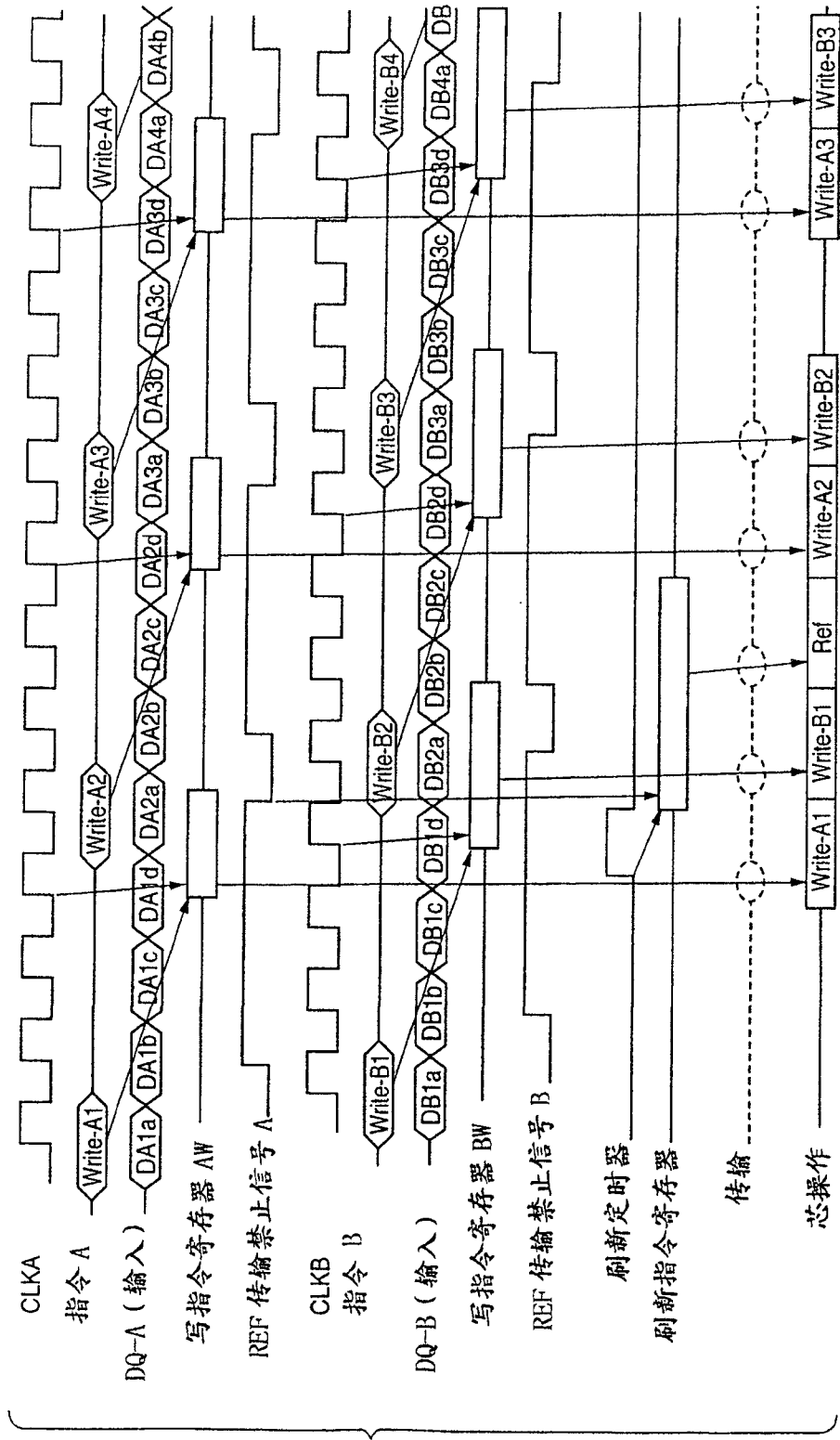


图 72

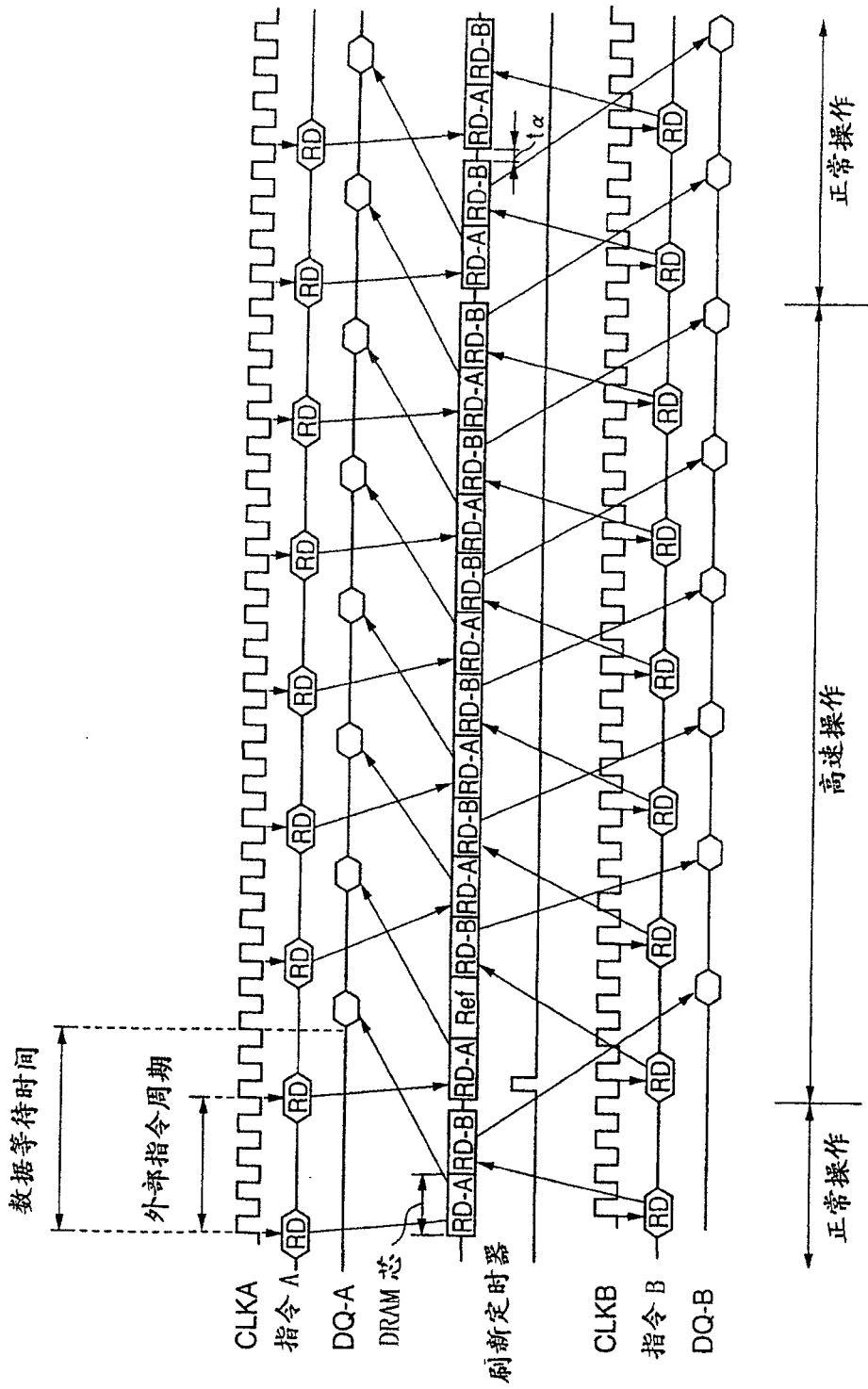


图73

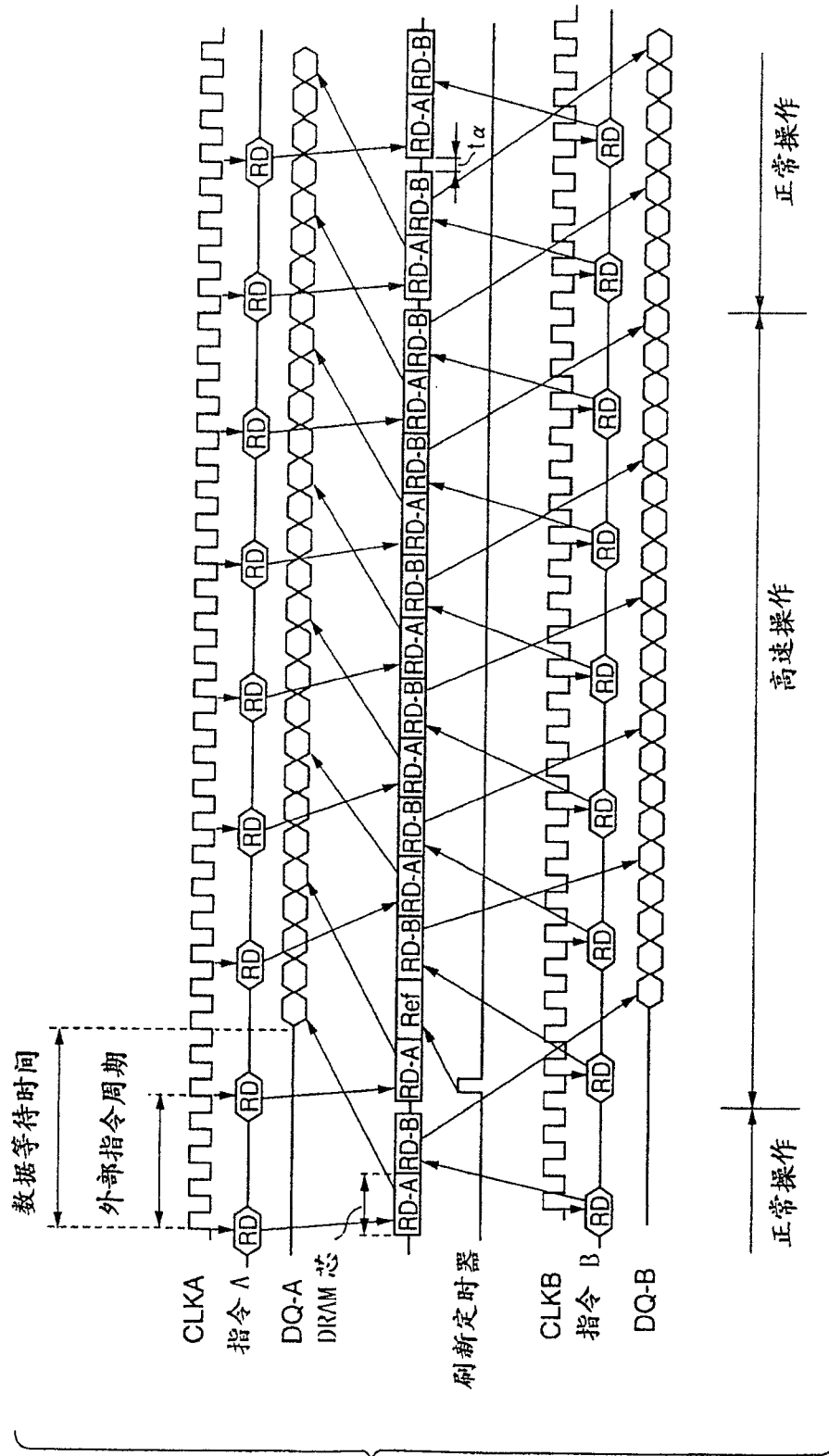


图 74

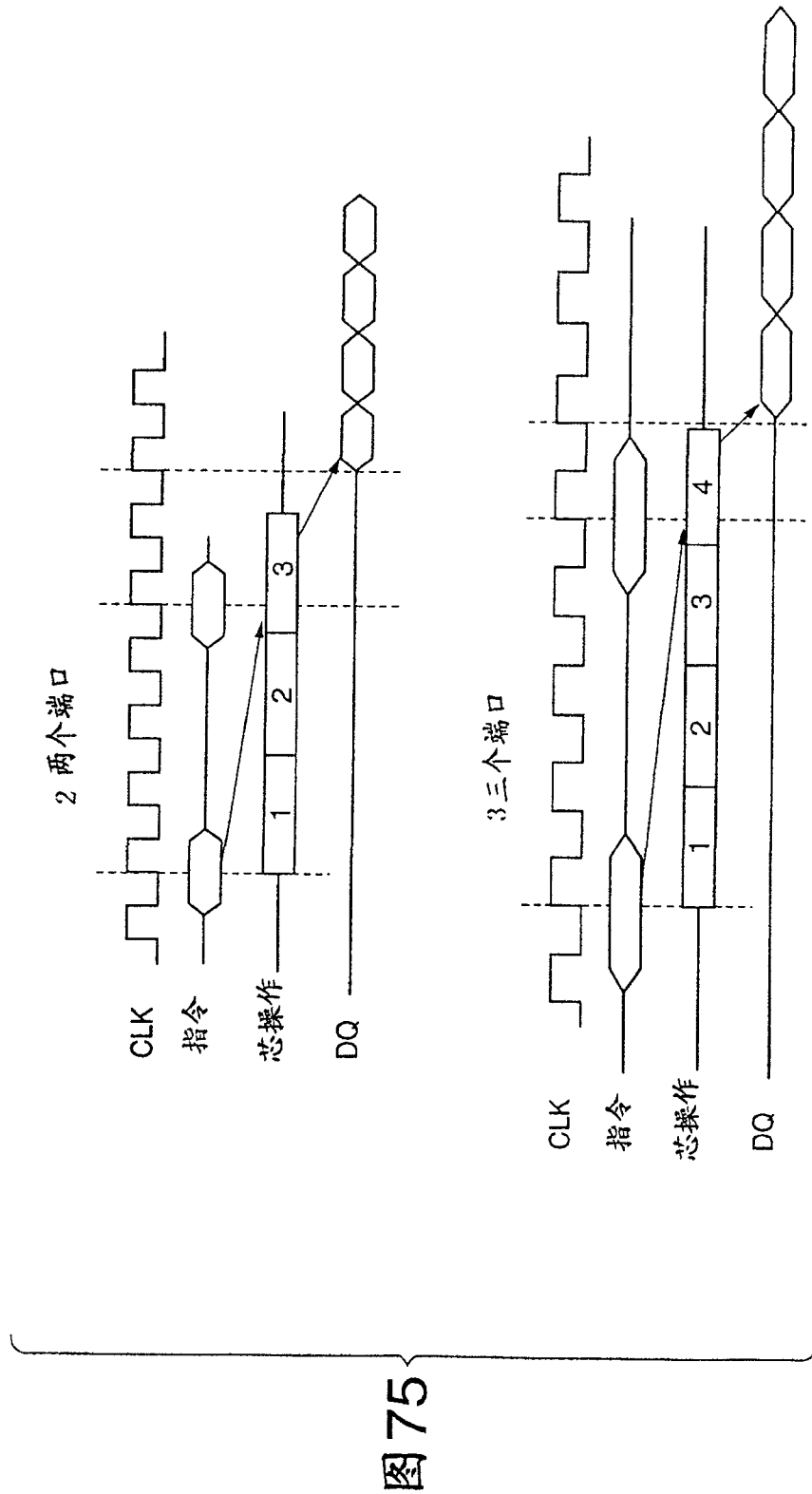


图75

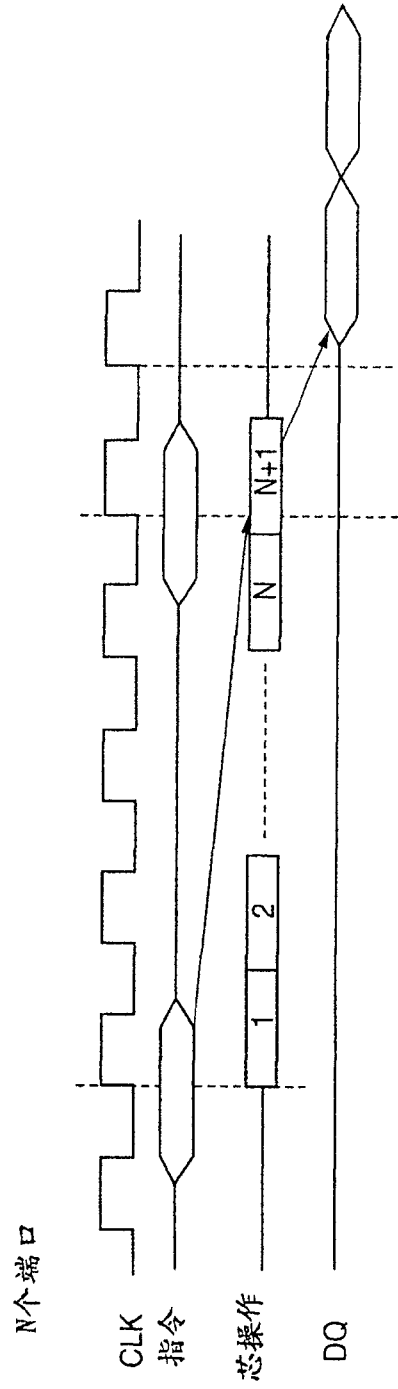


图 76

图 77

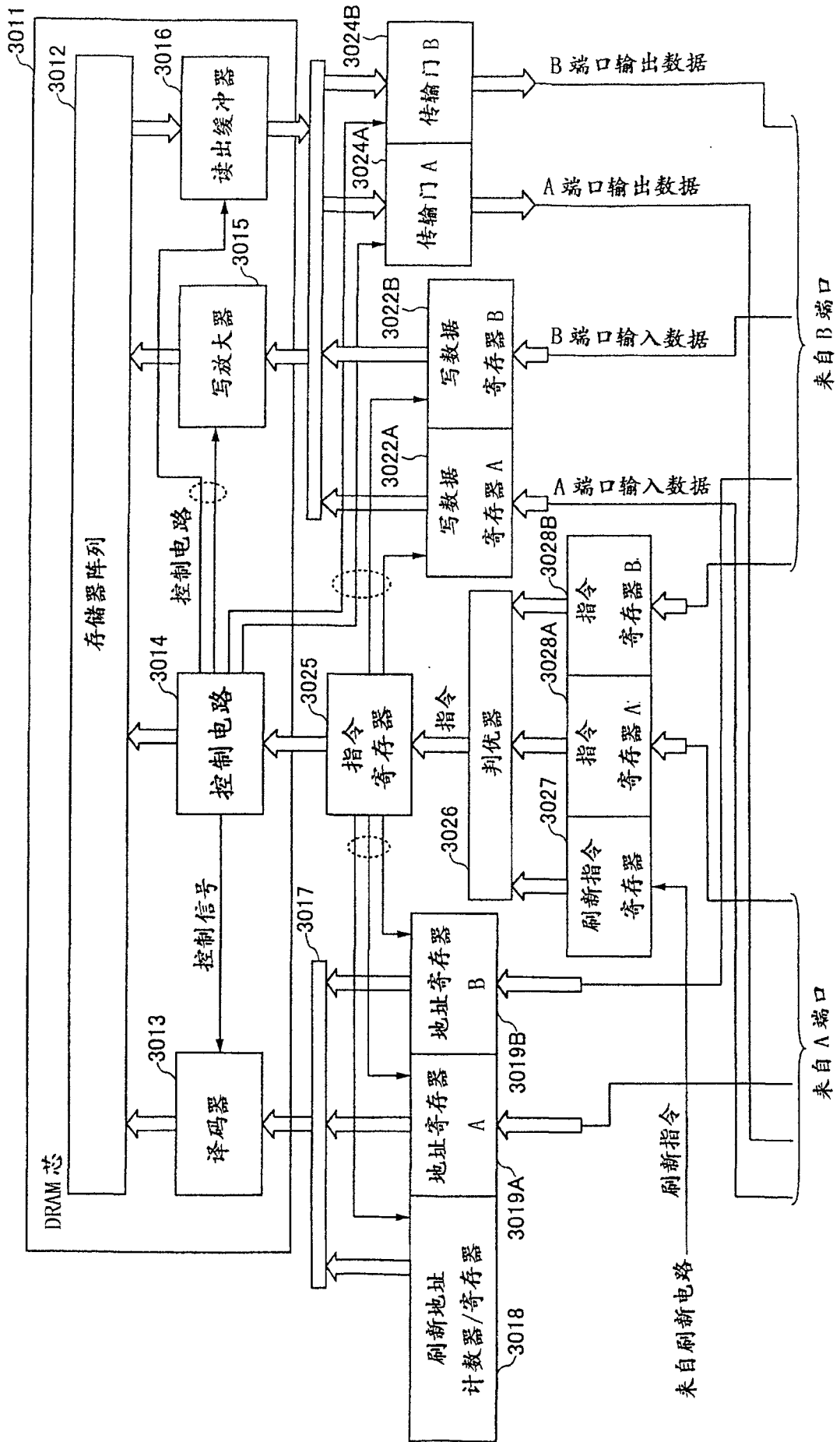


图 78A

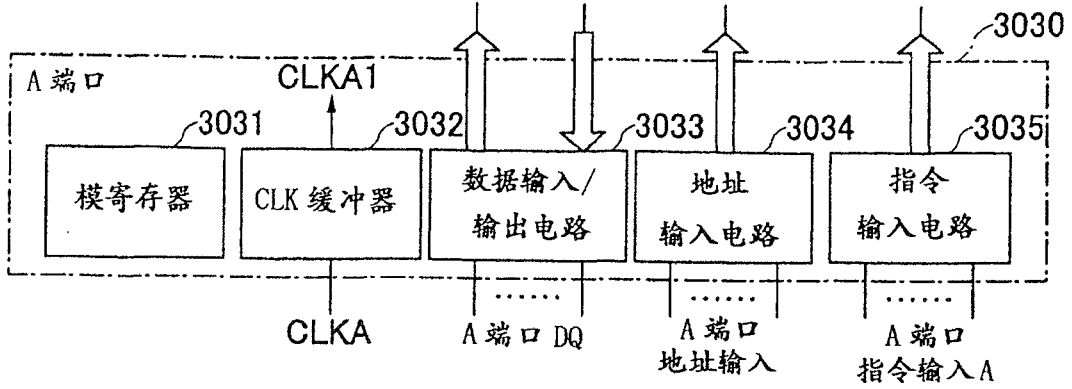


图 78B

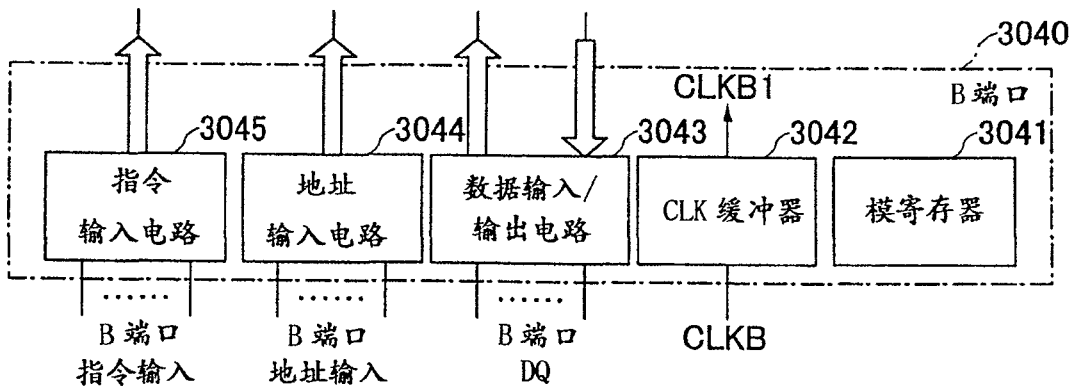


图 78C

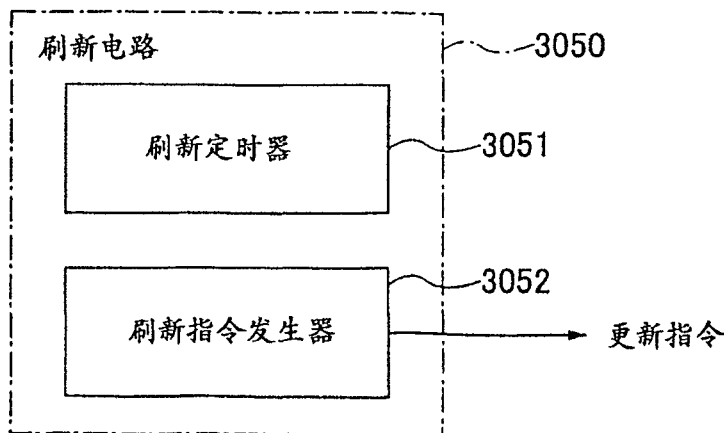


图 79

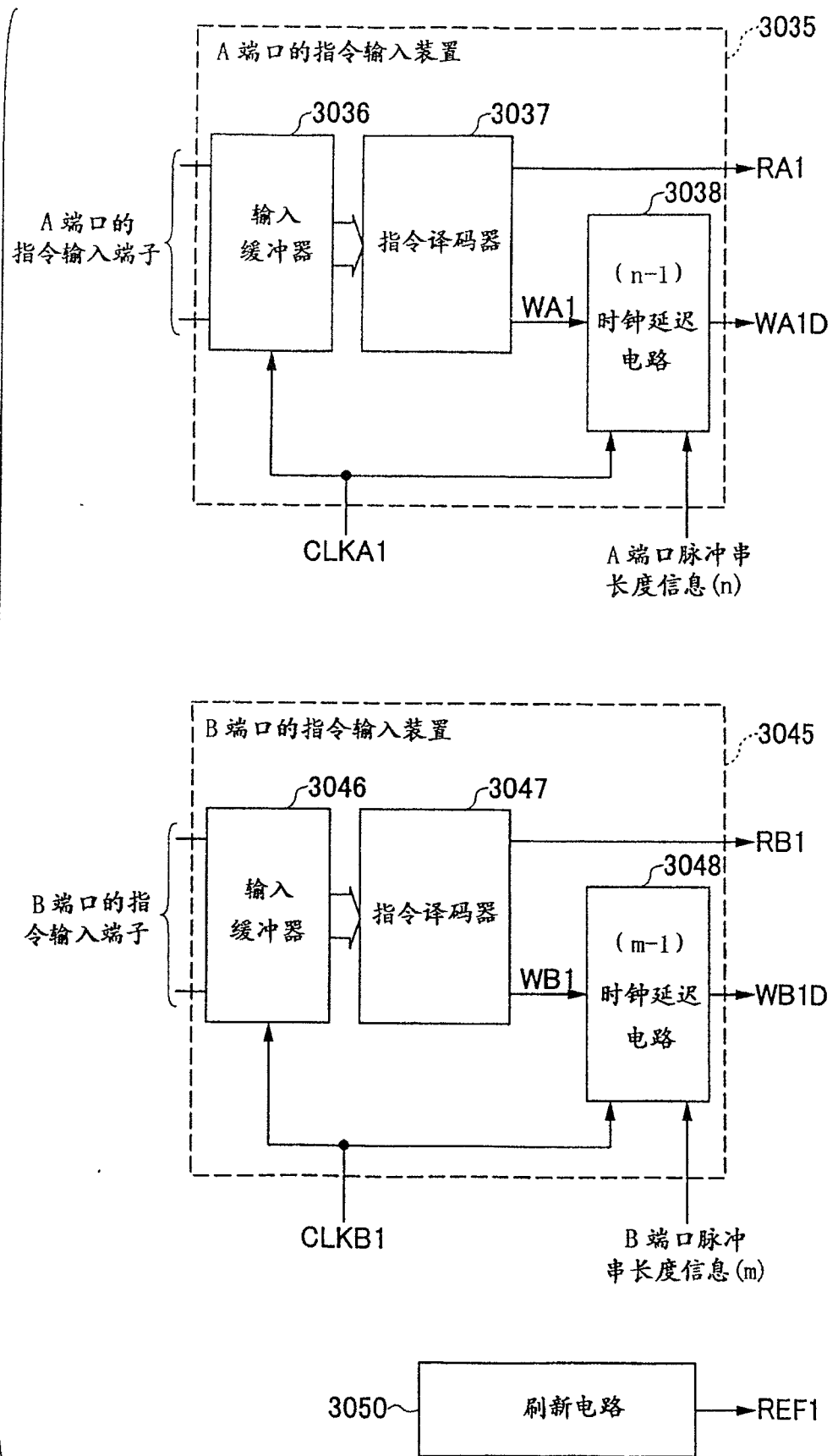


图 80

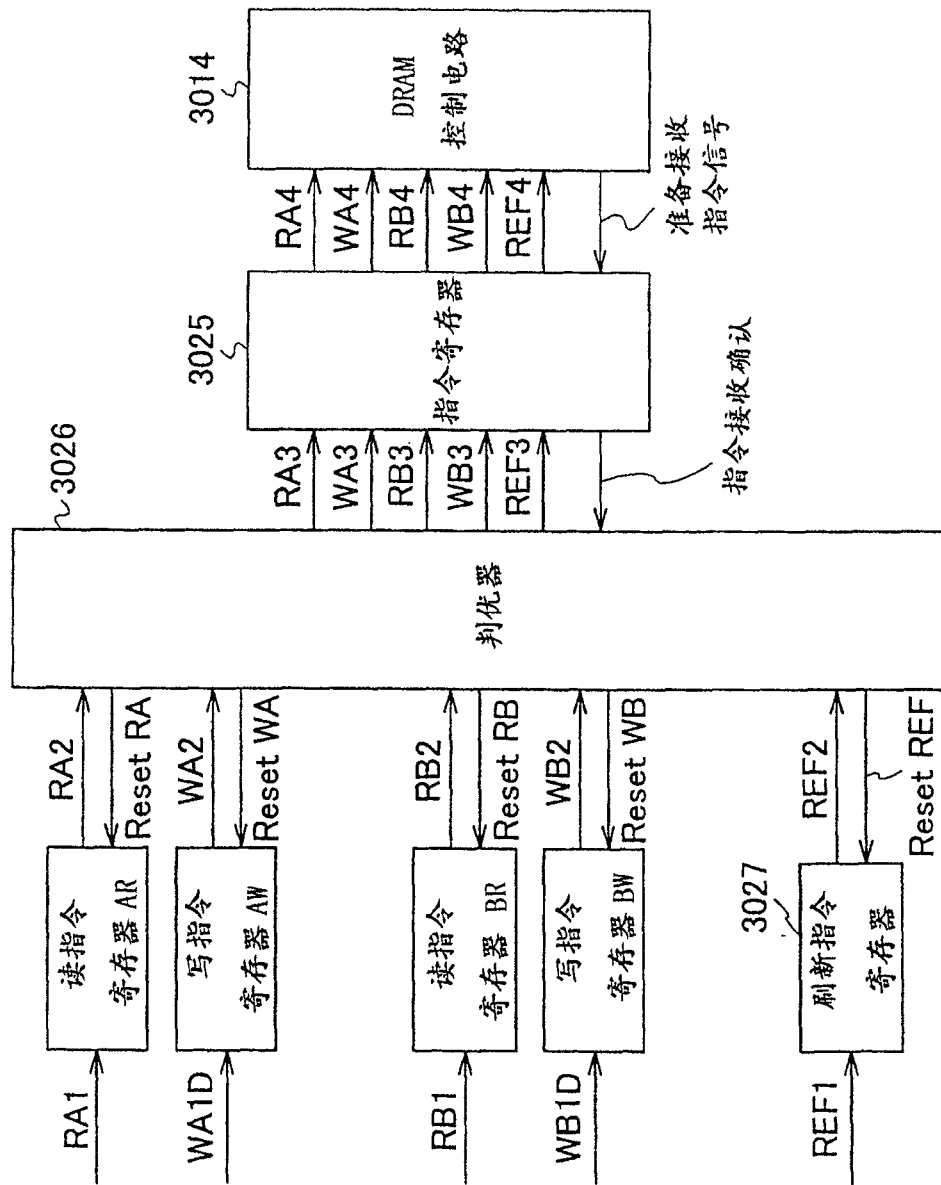


图 81

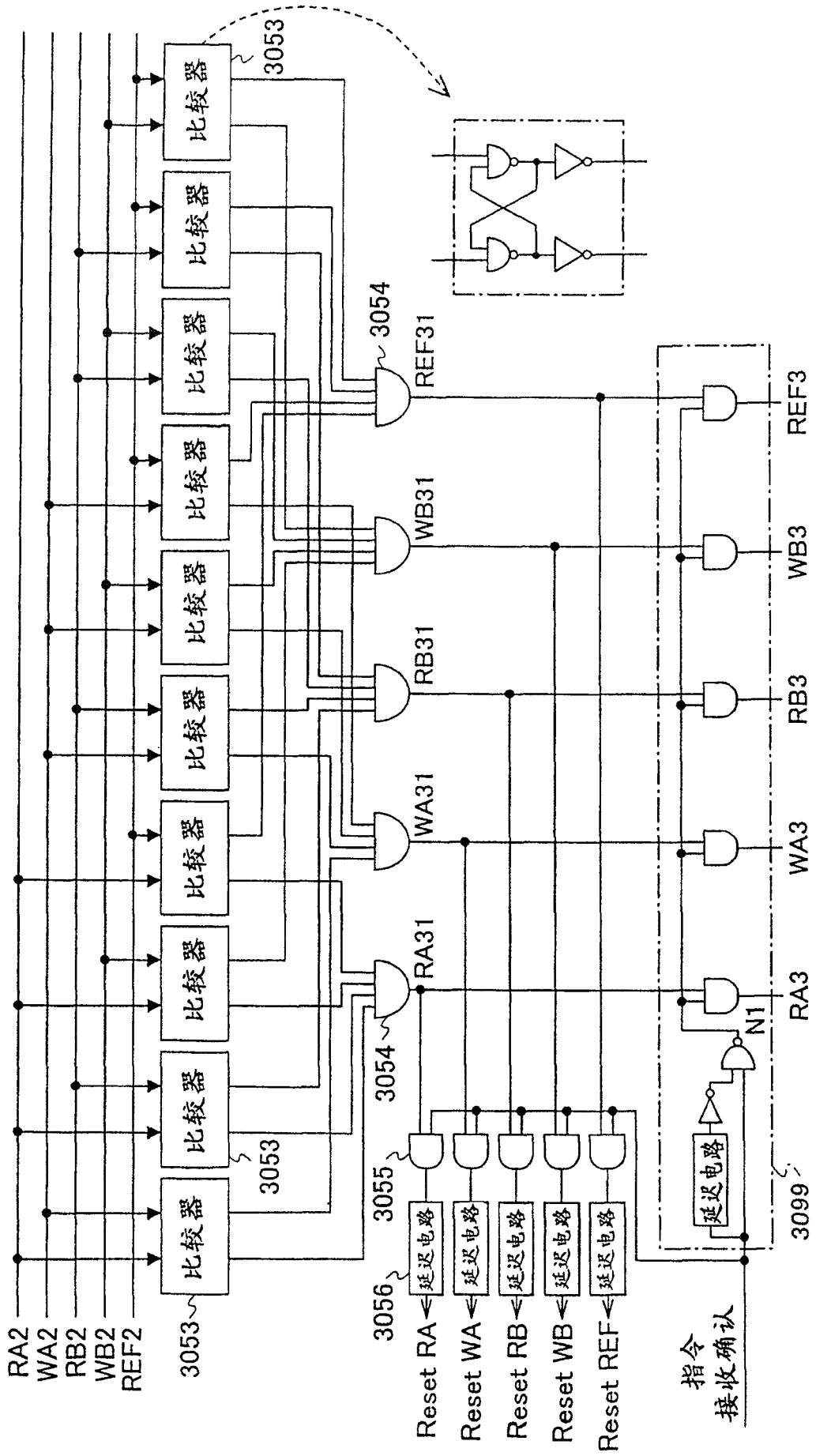


图 82

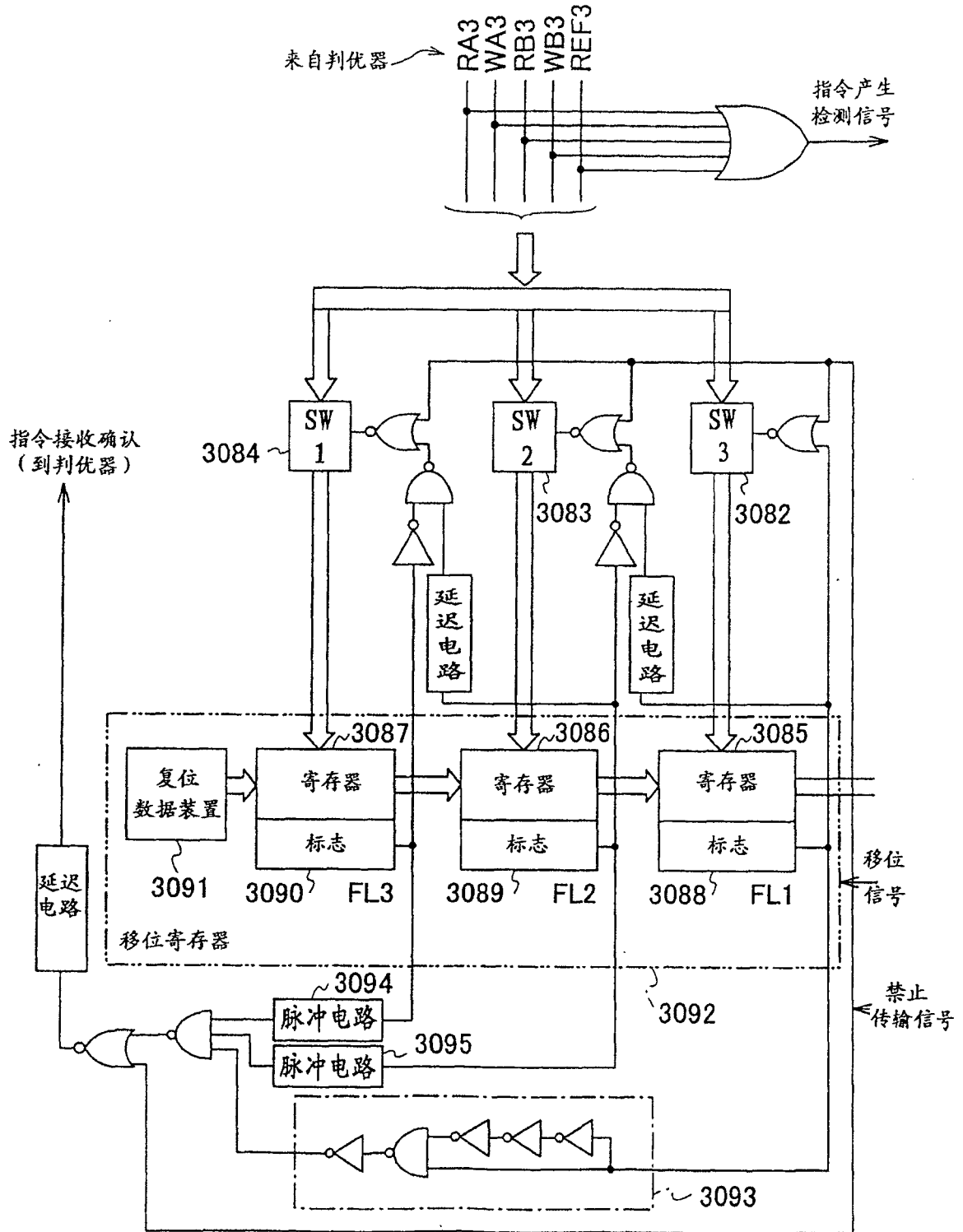


图 83

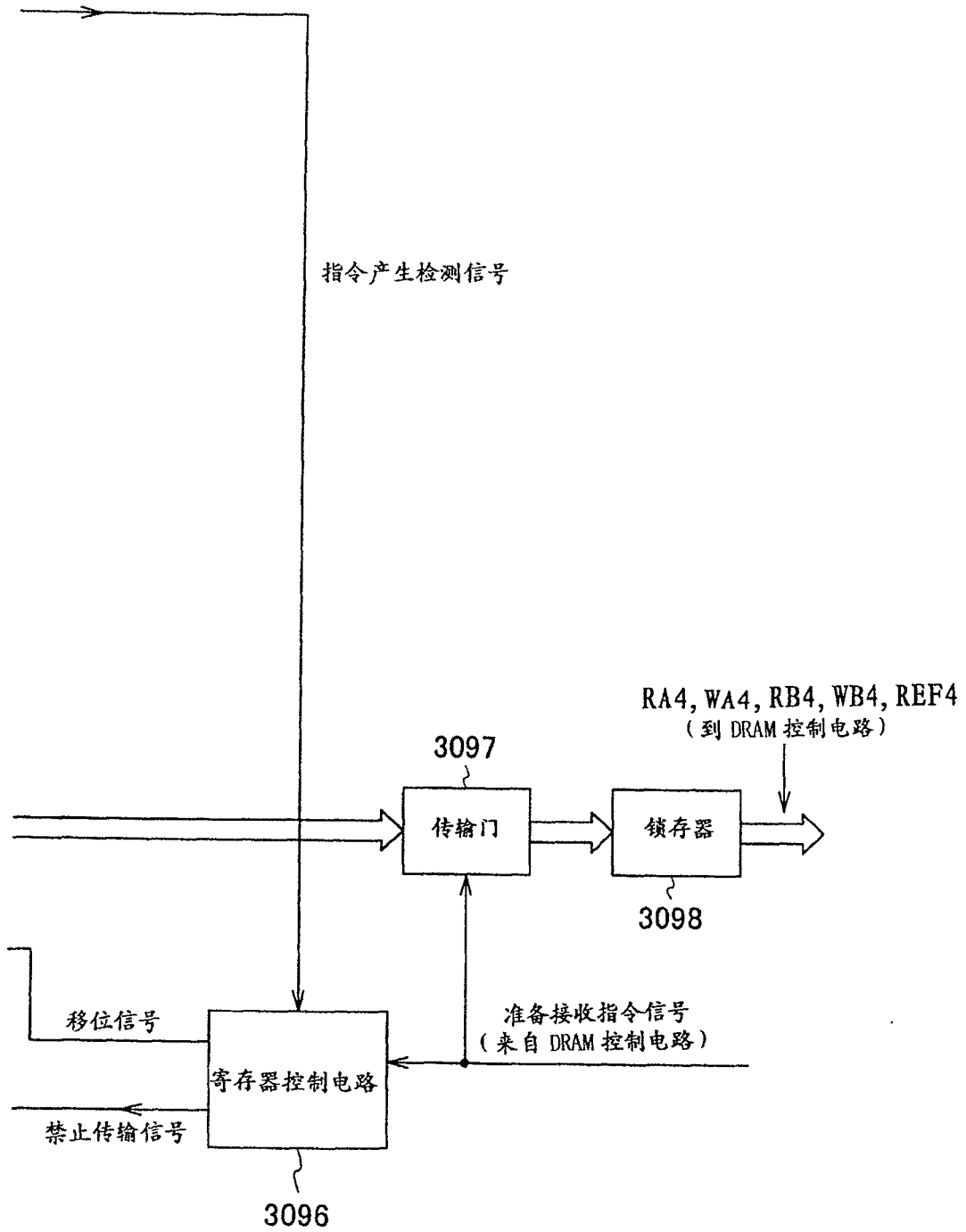


图 84A

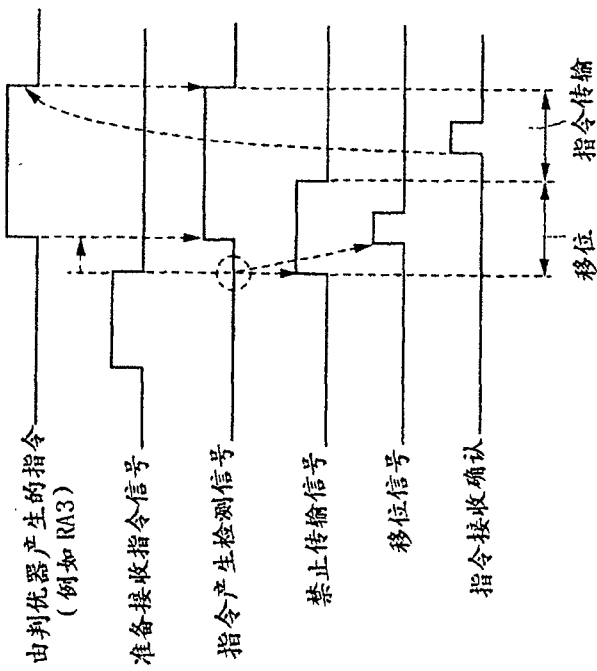
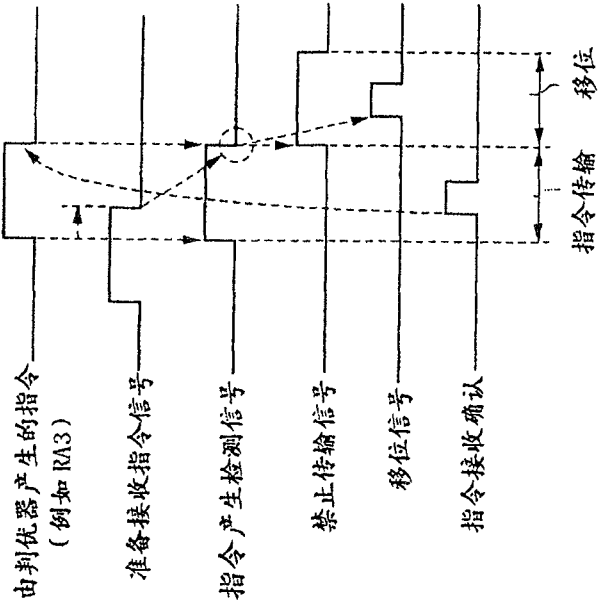


图 84B



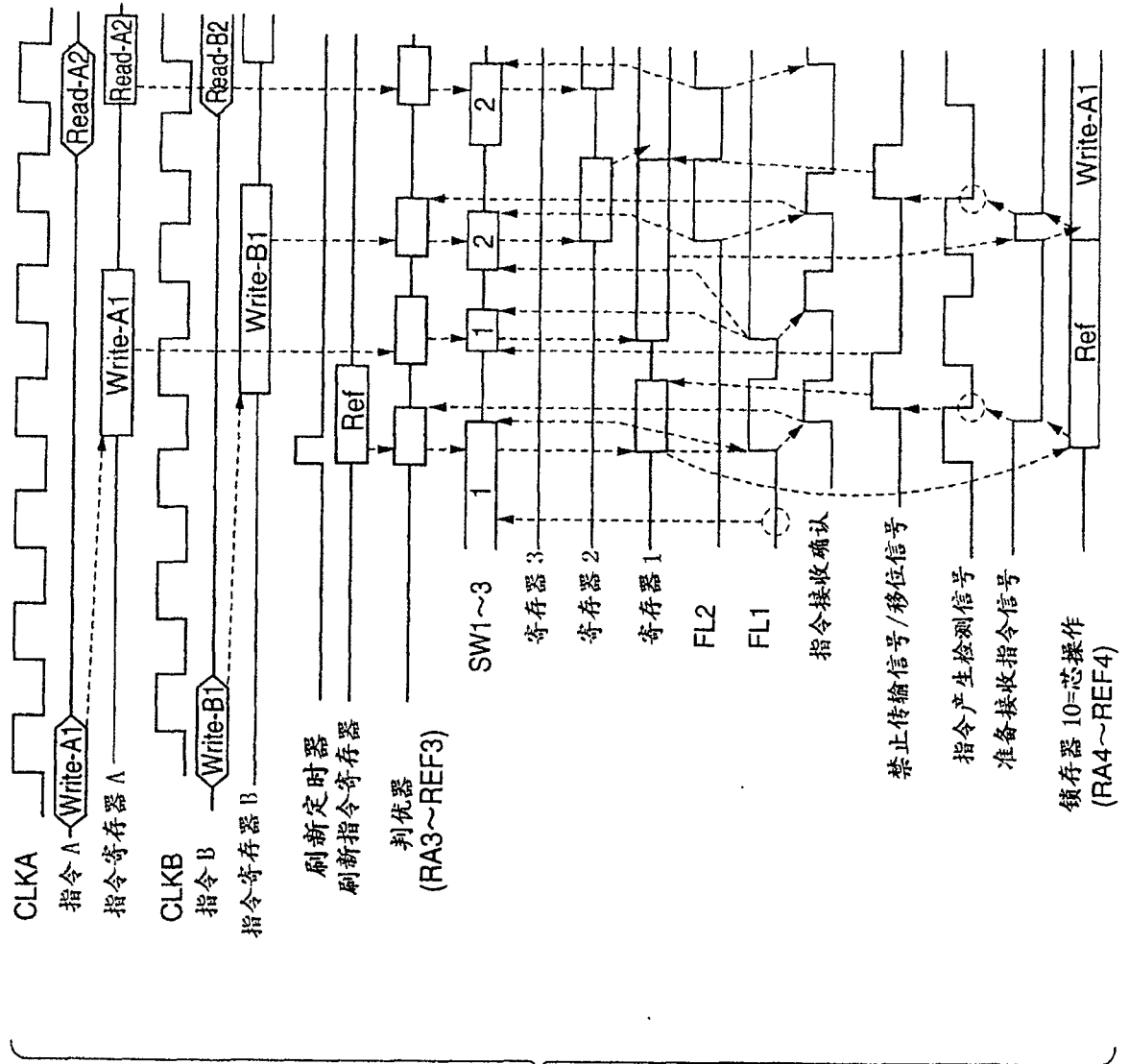


图 85

图 86

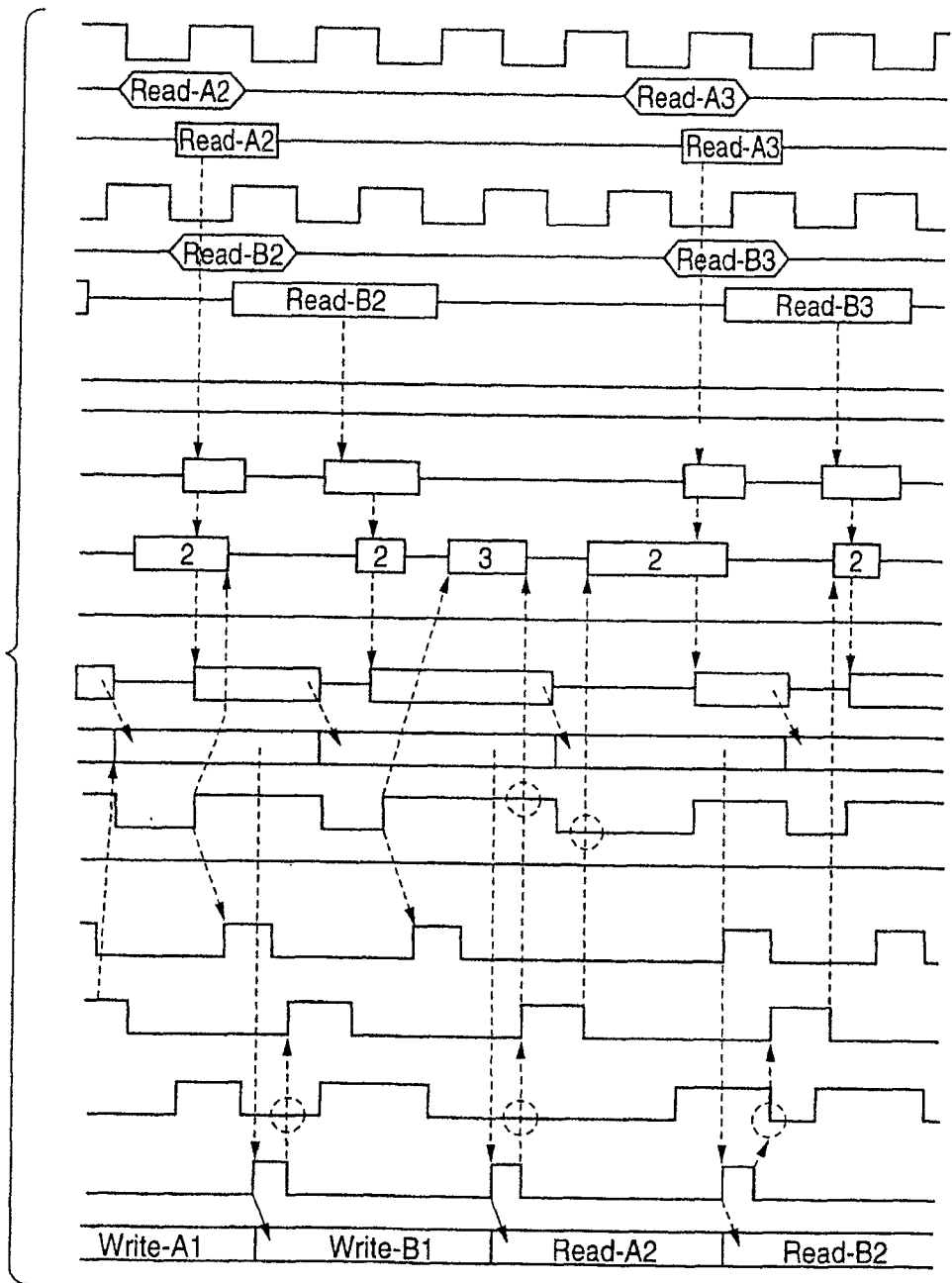


图 88

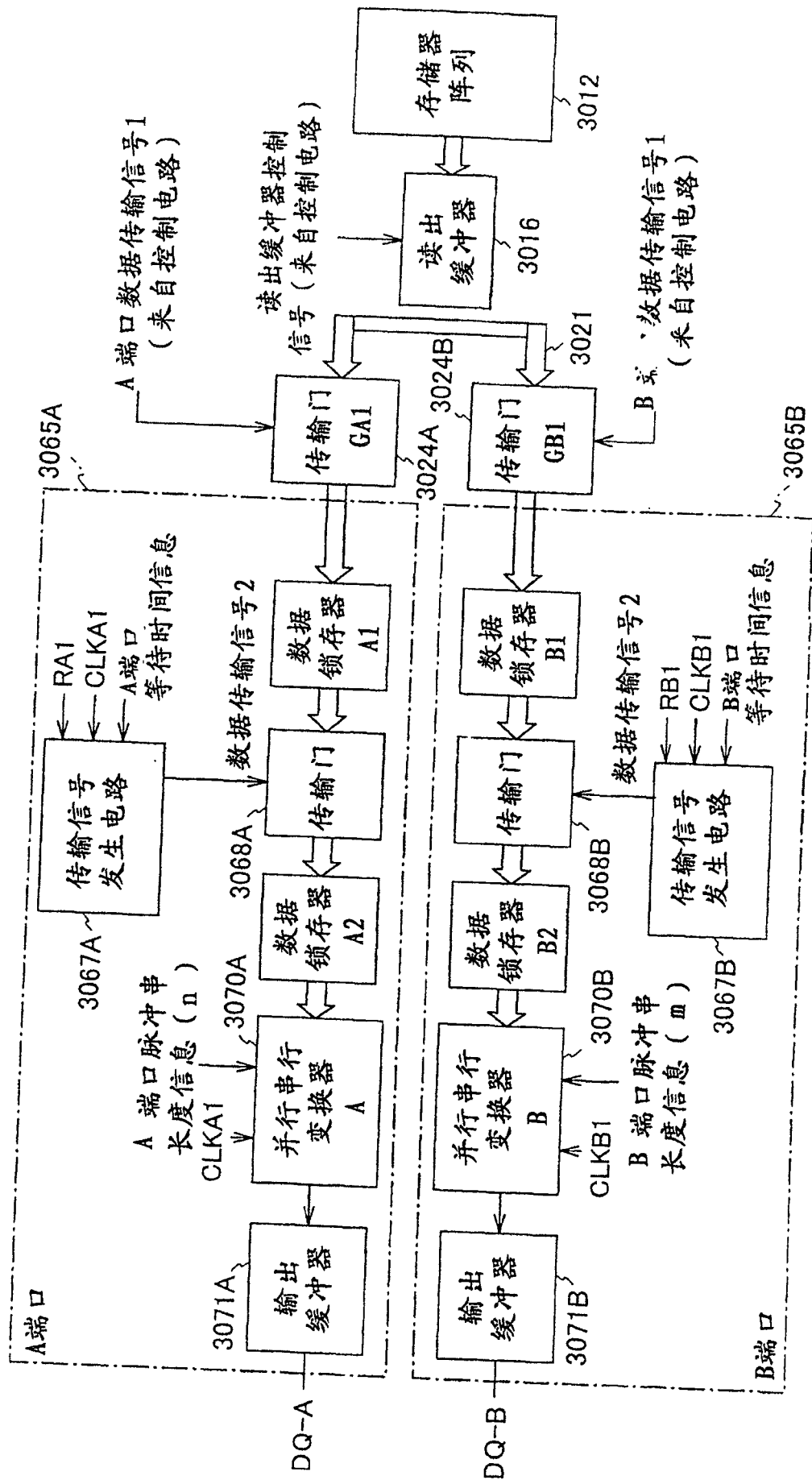
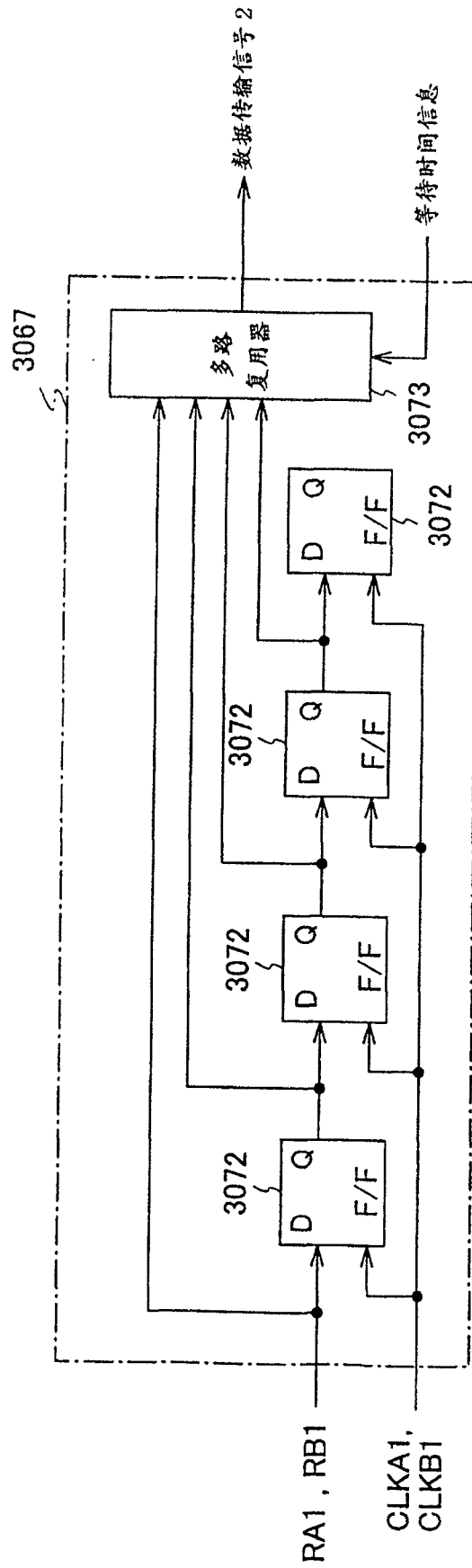


图 89



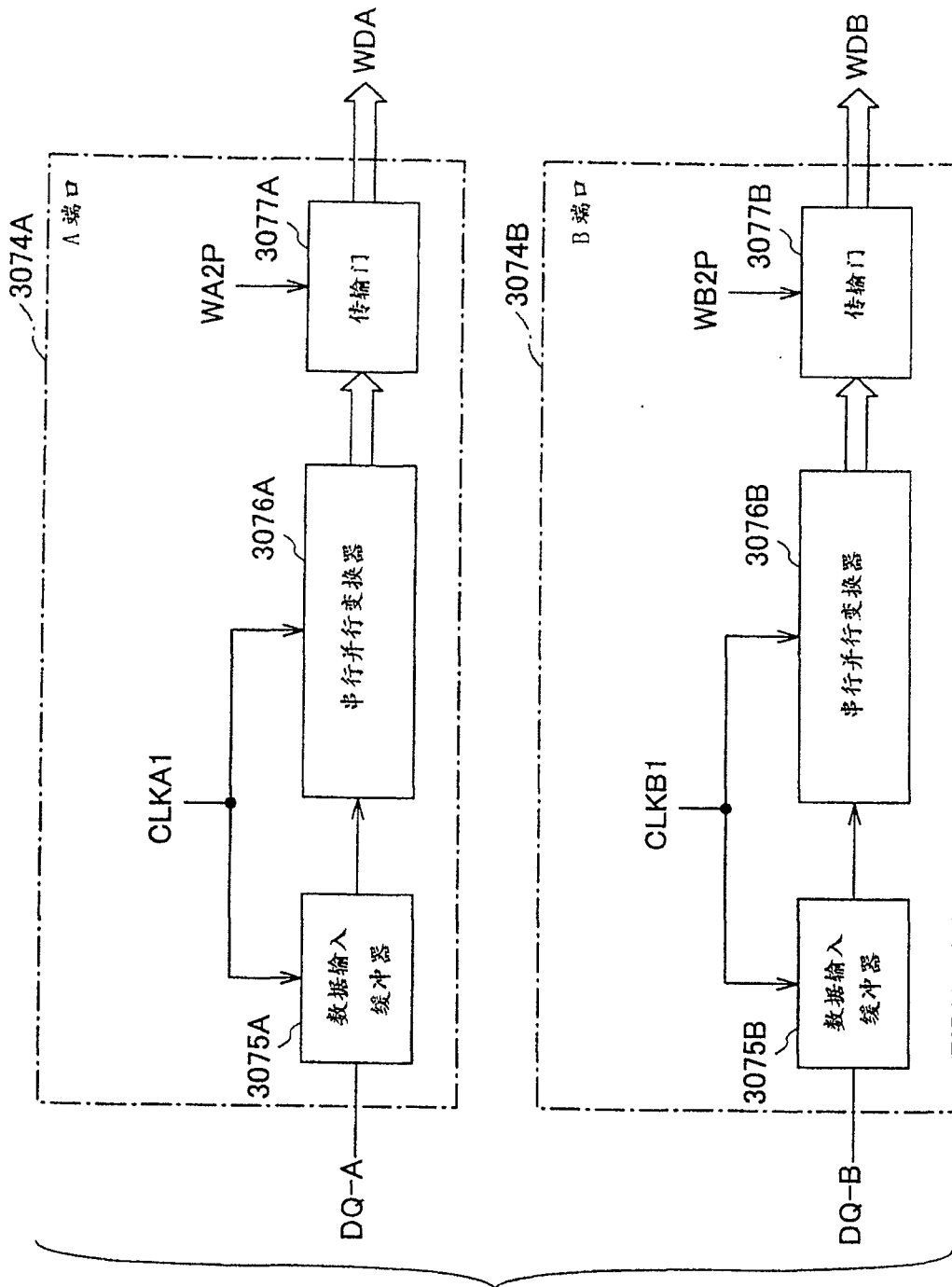


图 90

图 91

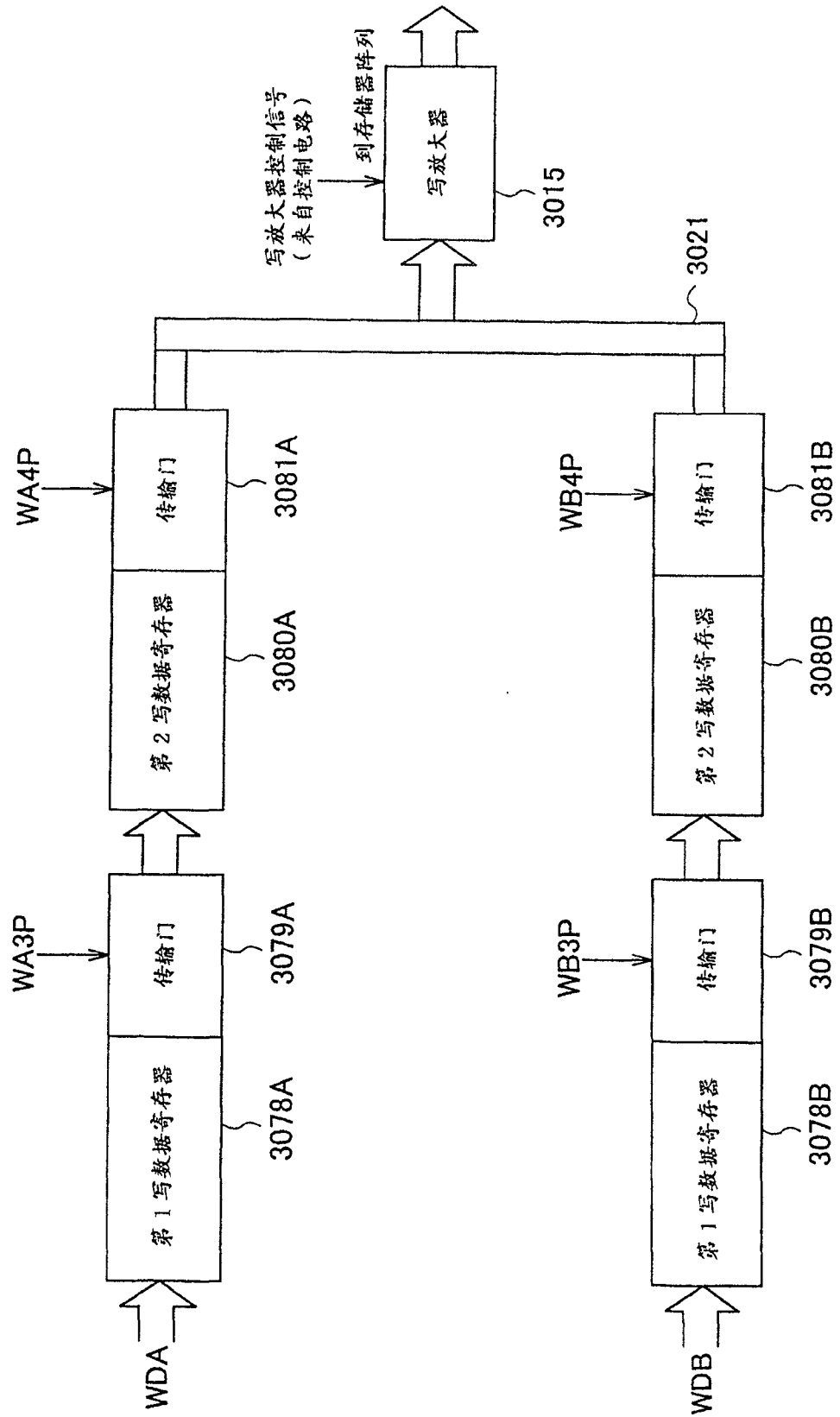
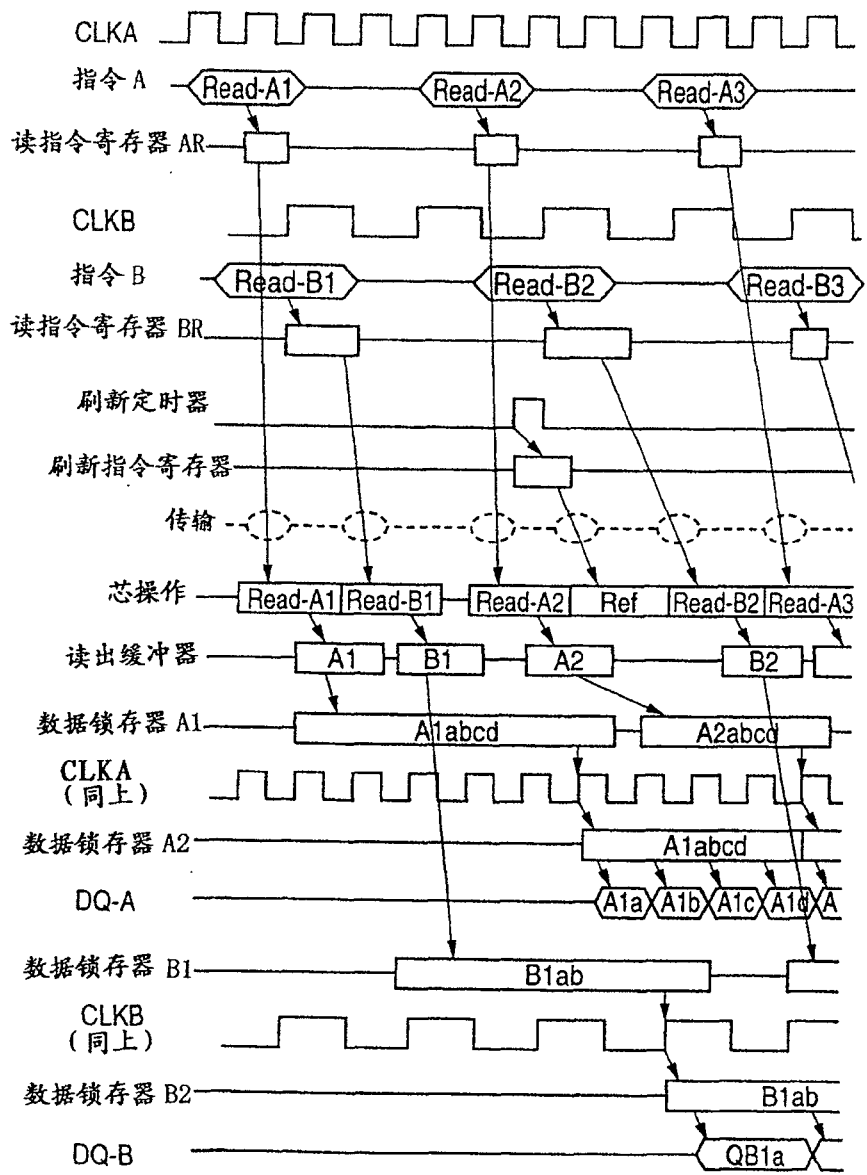


图 92



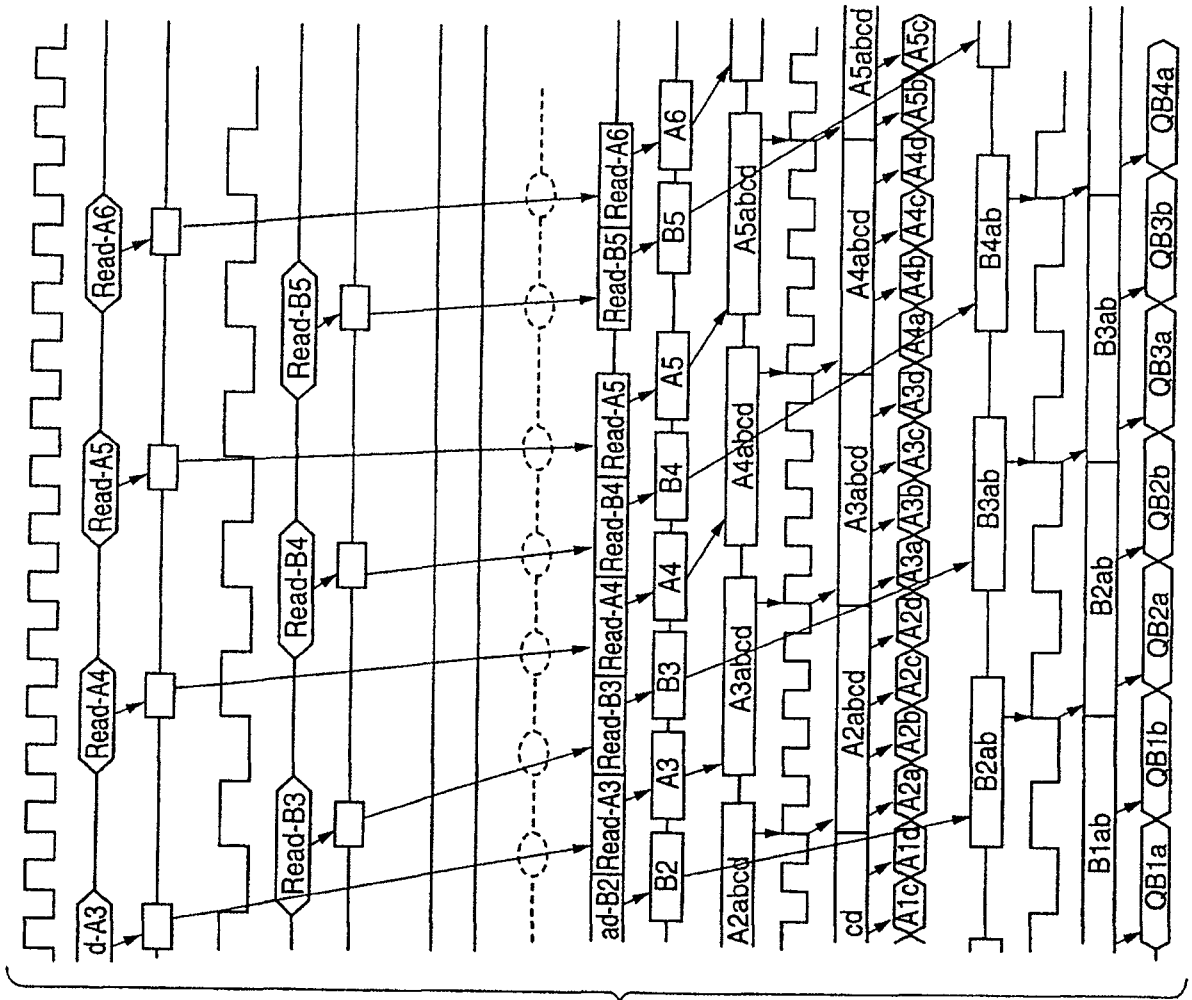


图 93

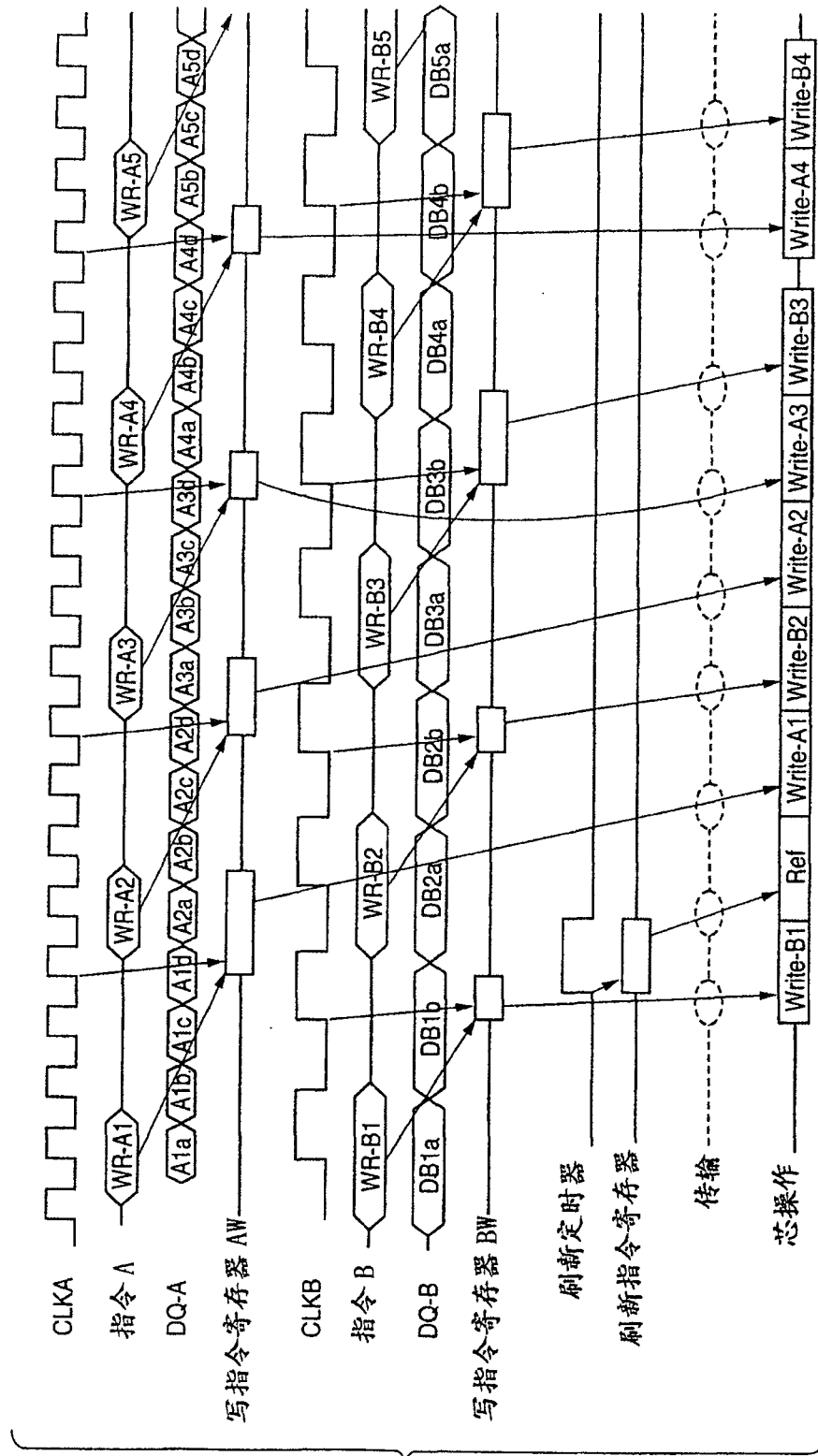


图 94

图 95

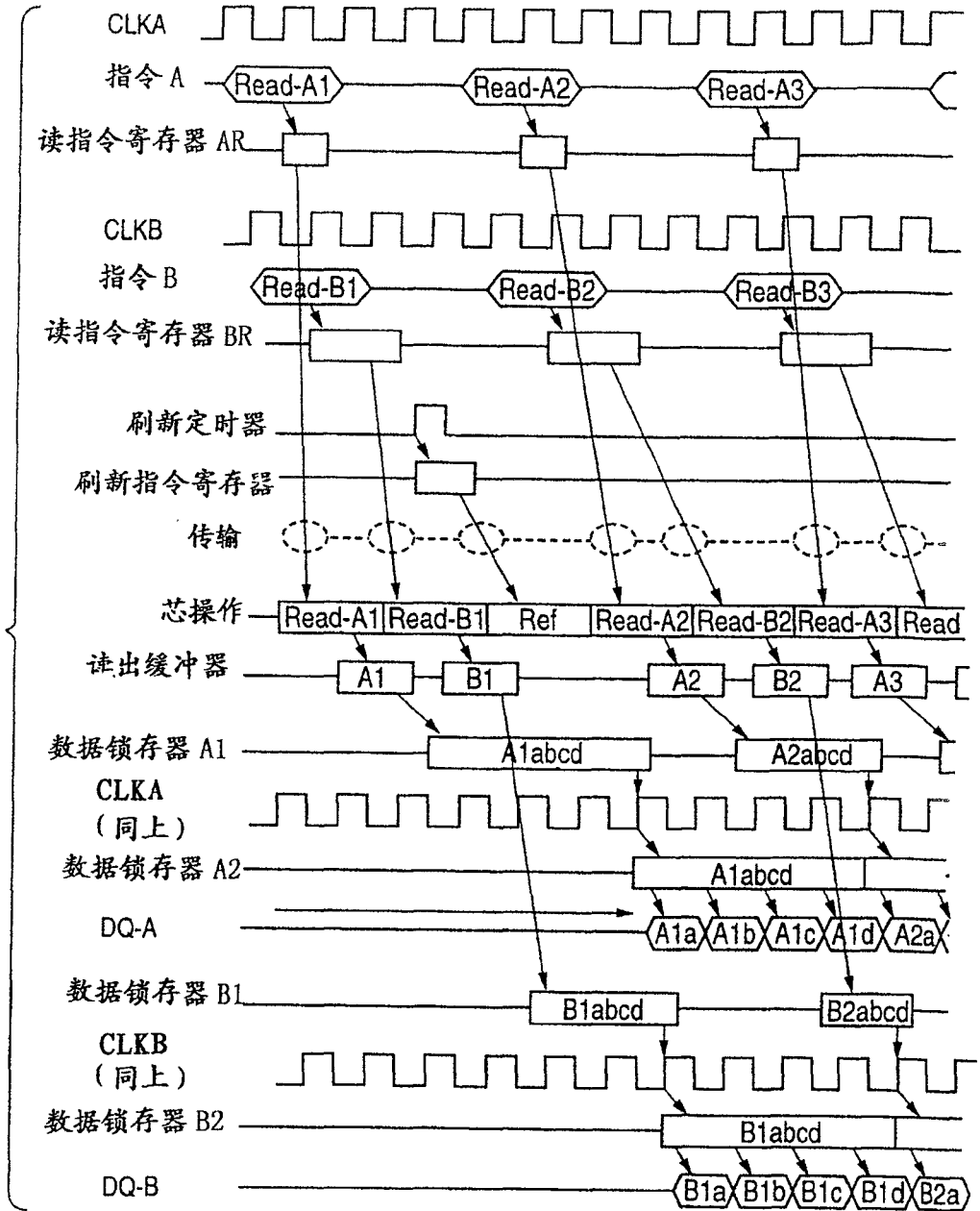
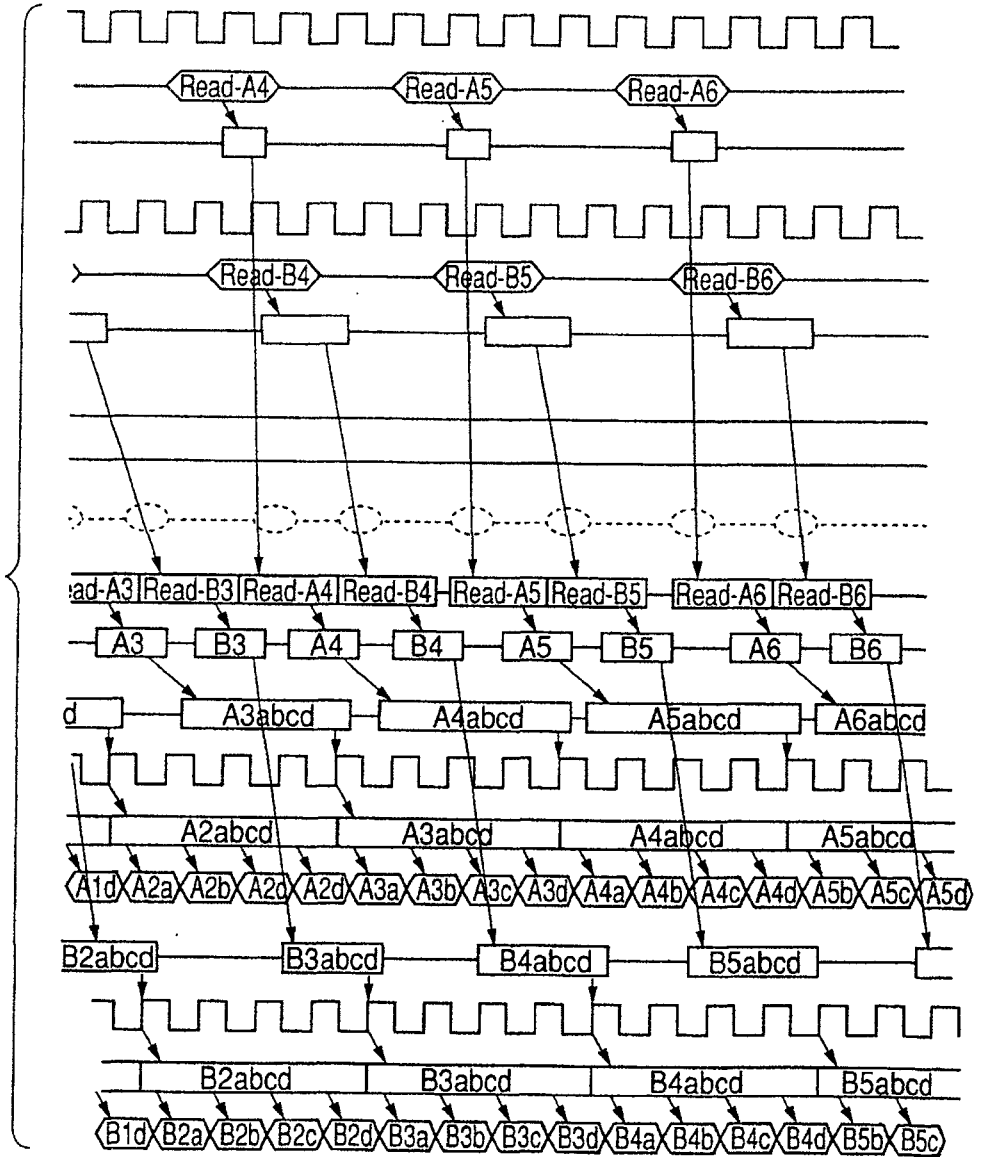


图 96



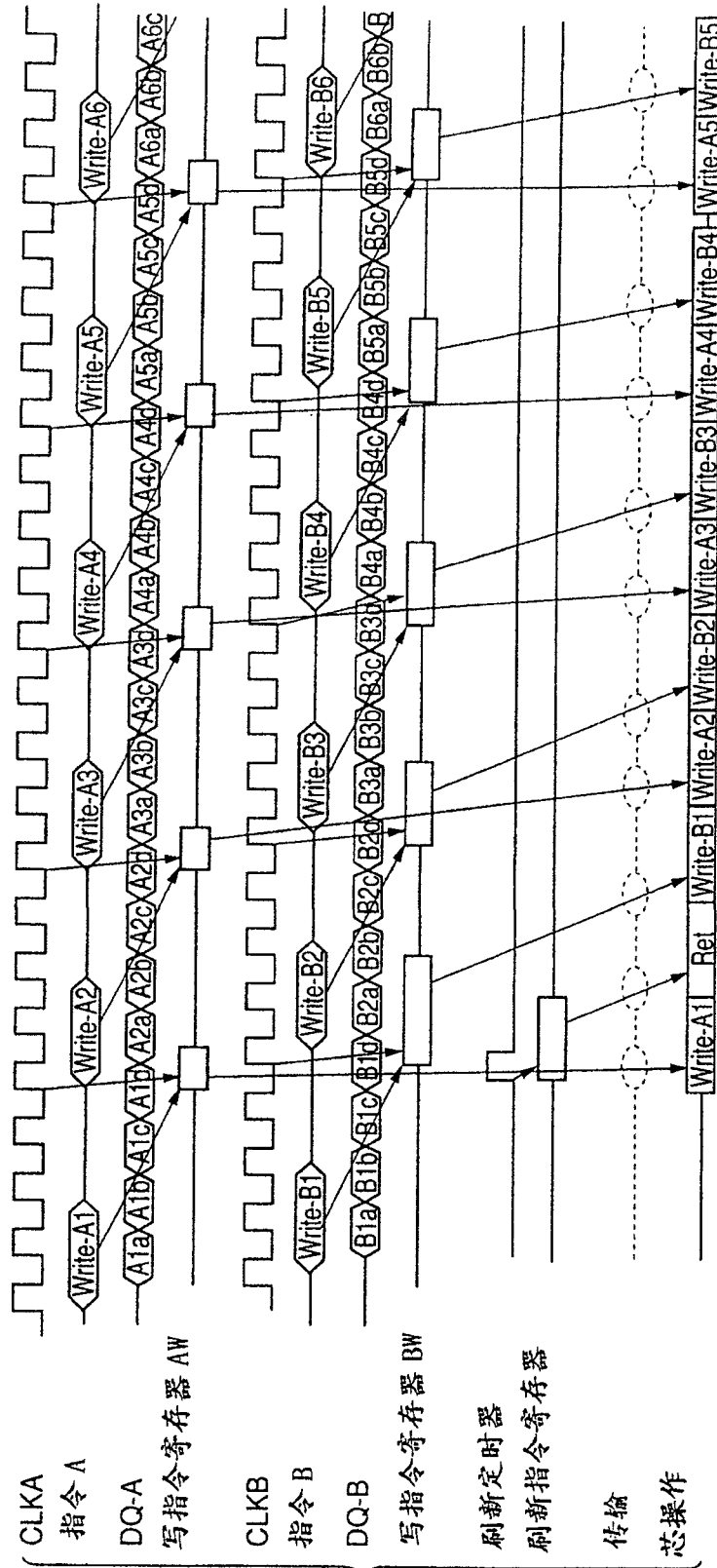


图 97

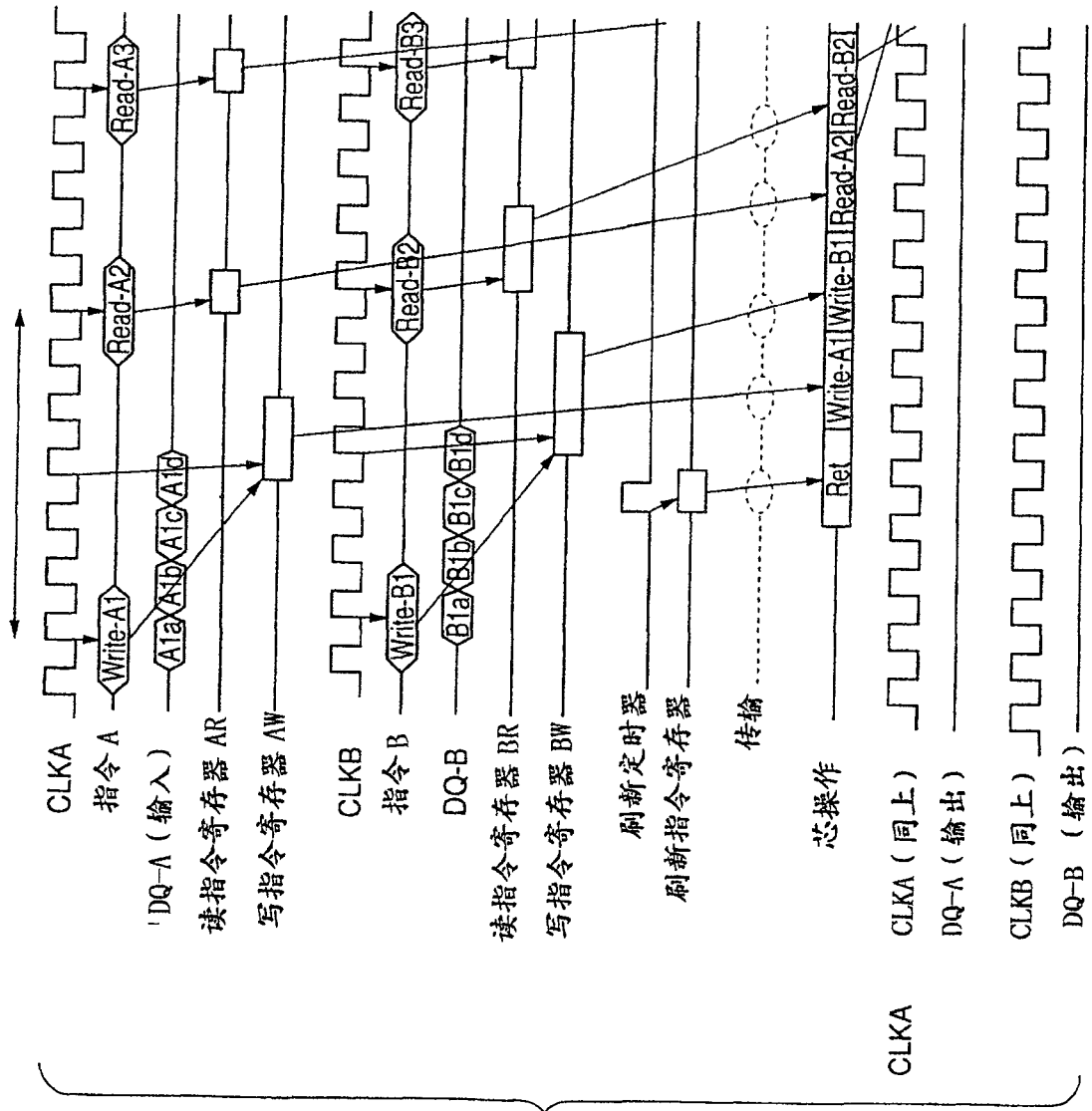


图 98

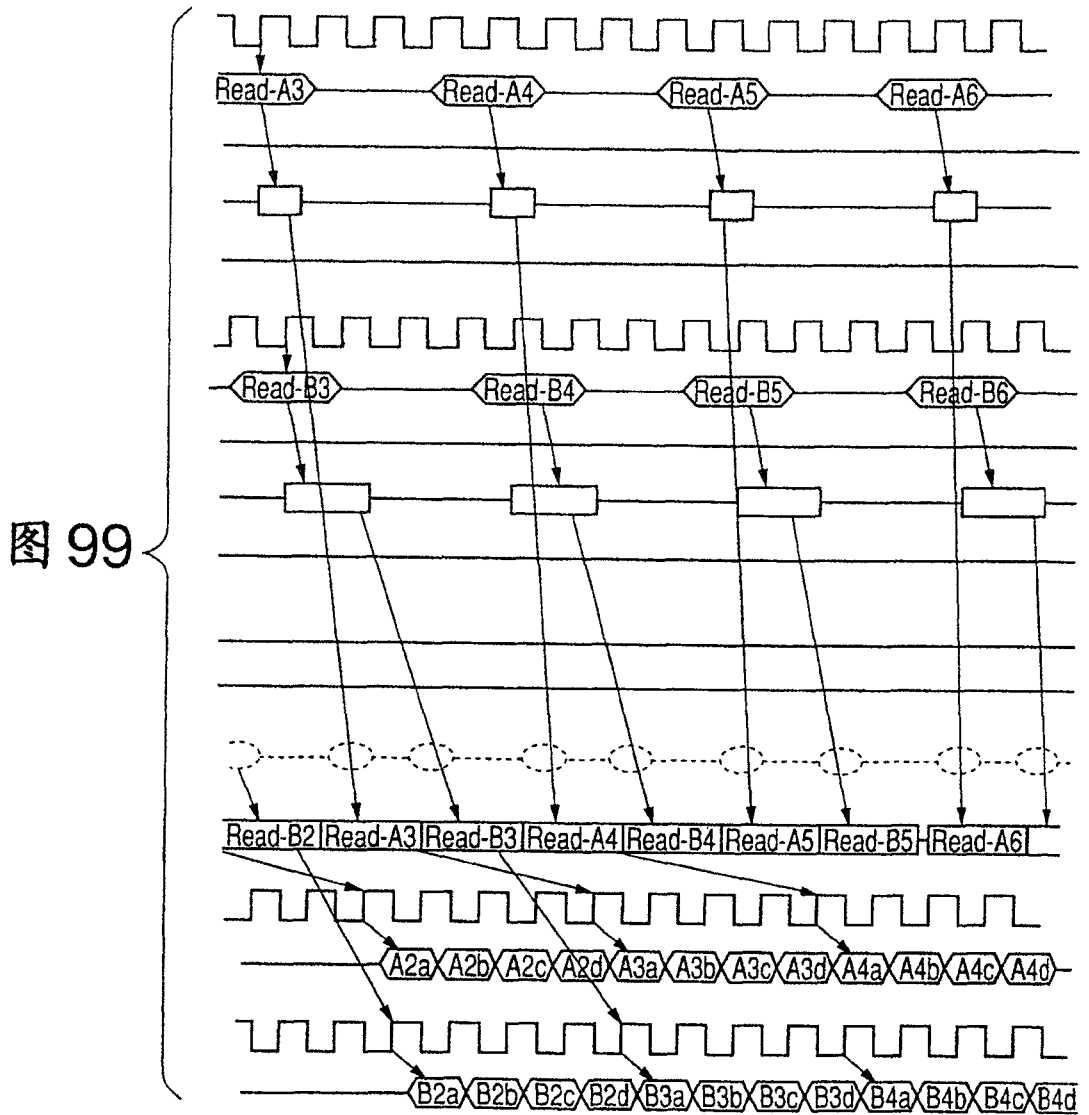


图100A

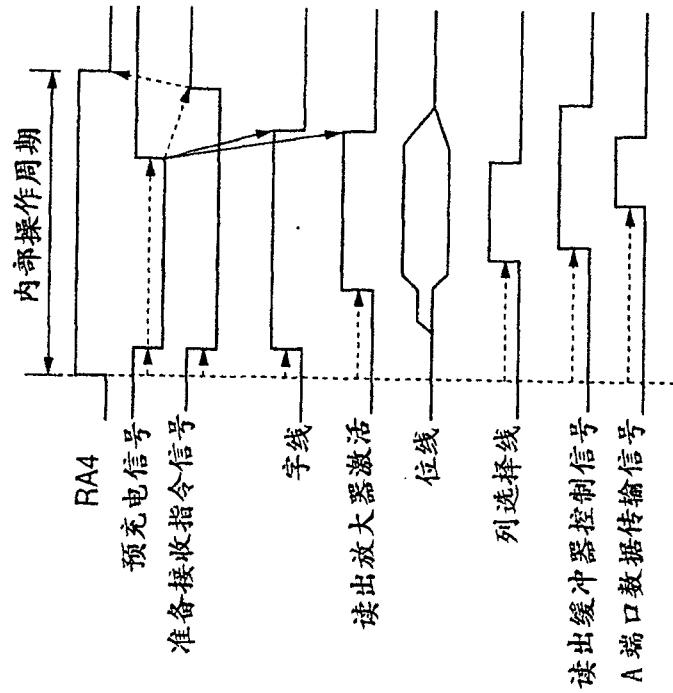


图100B

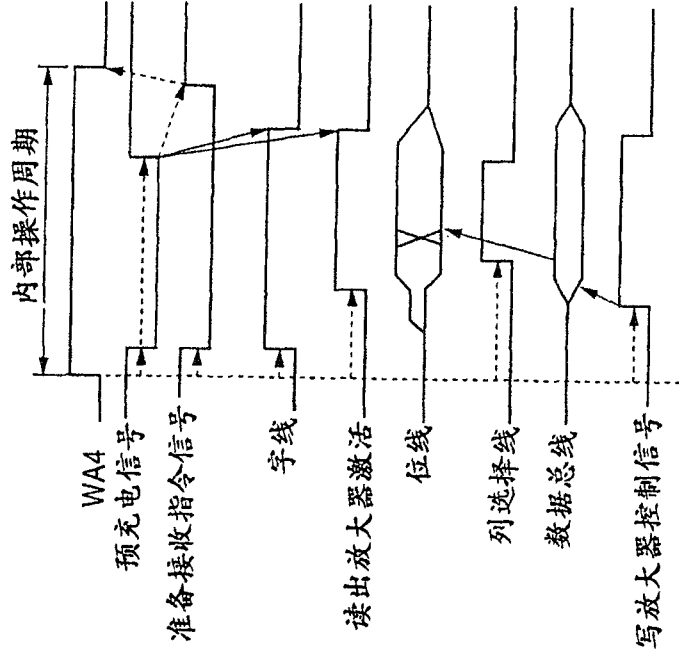


图 101

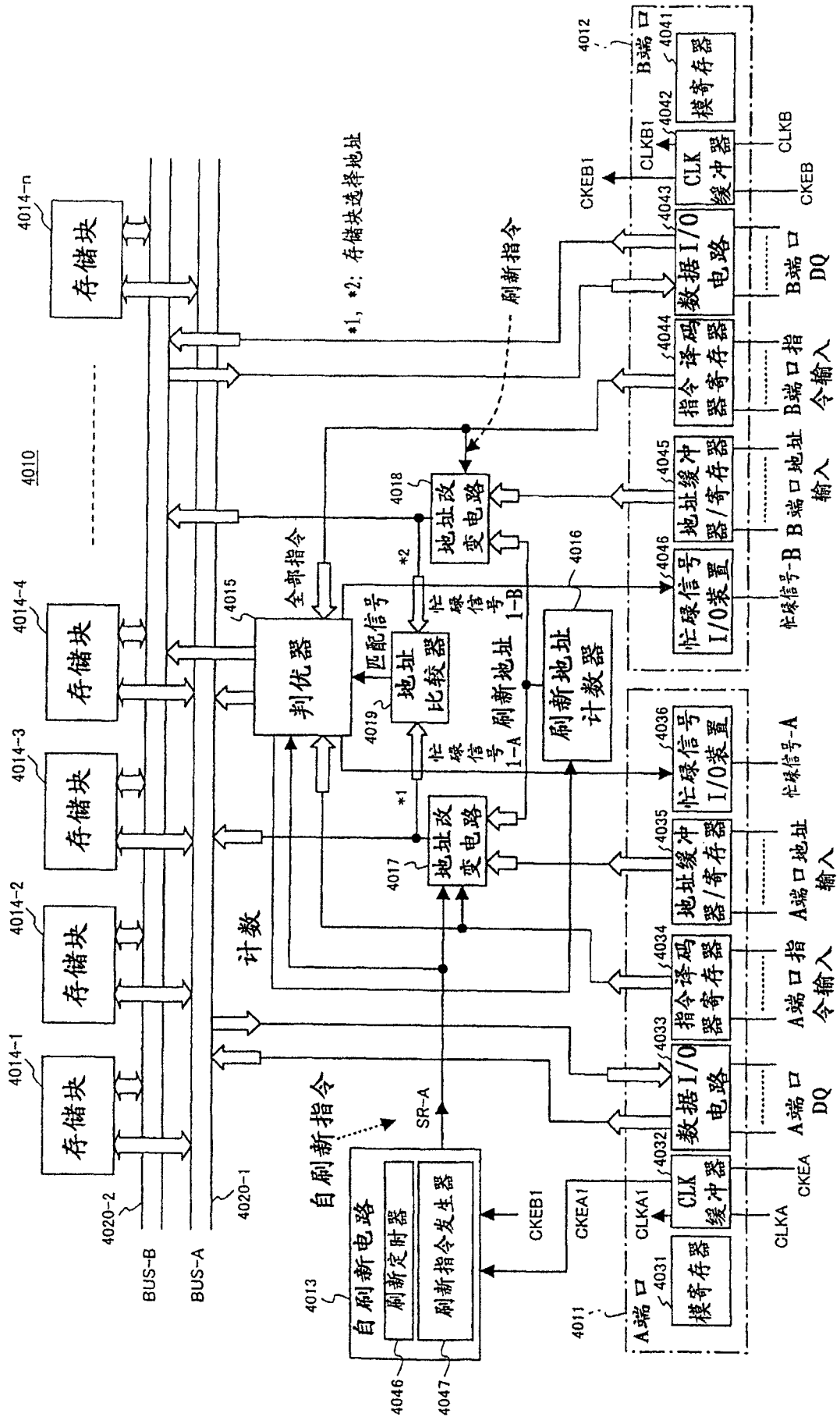


图102

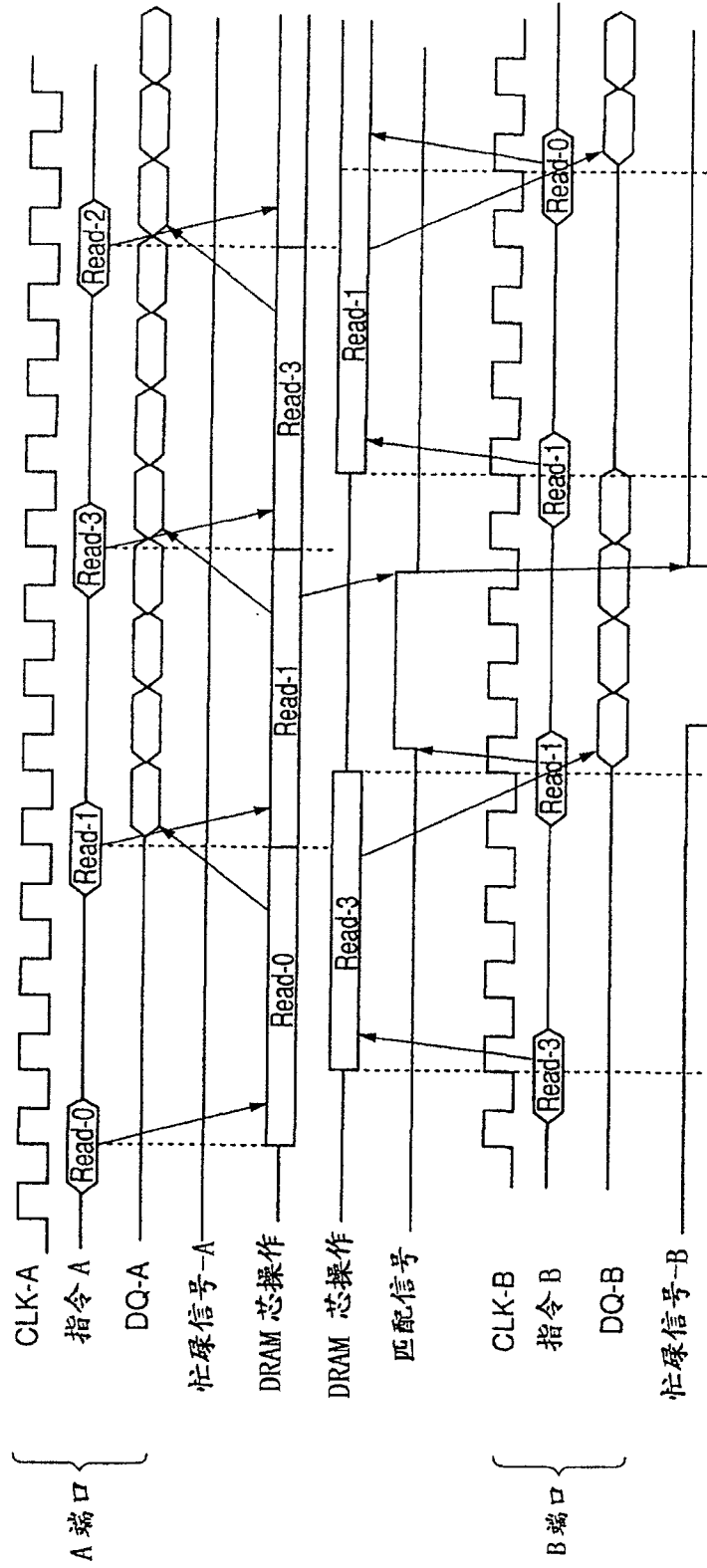


图 103

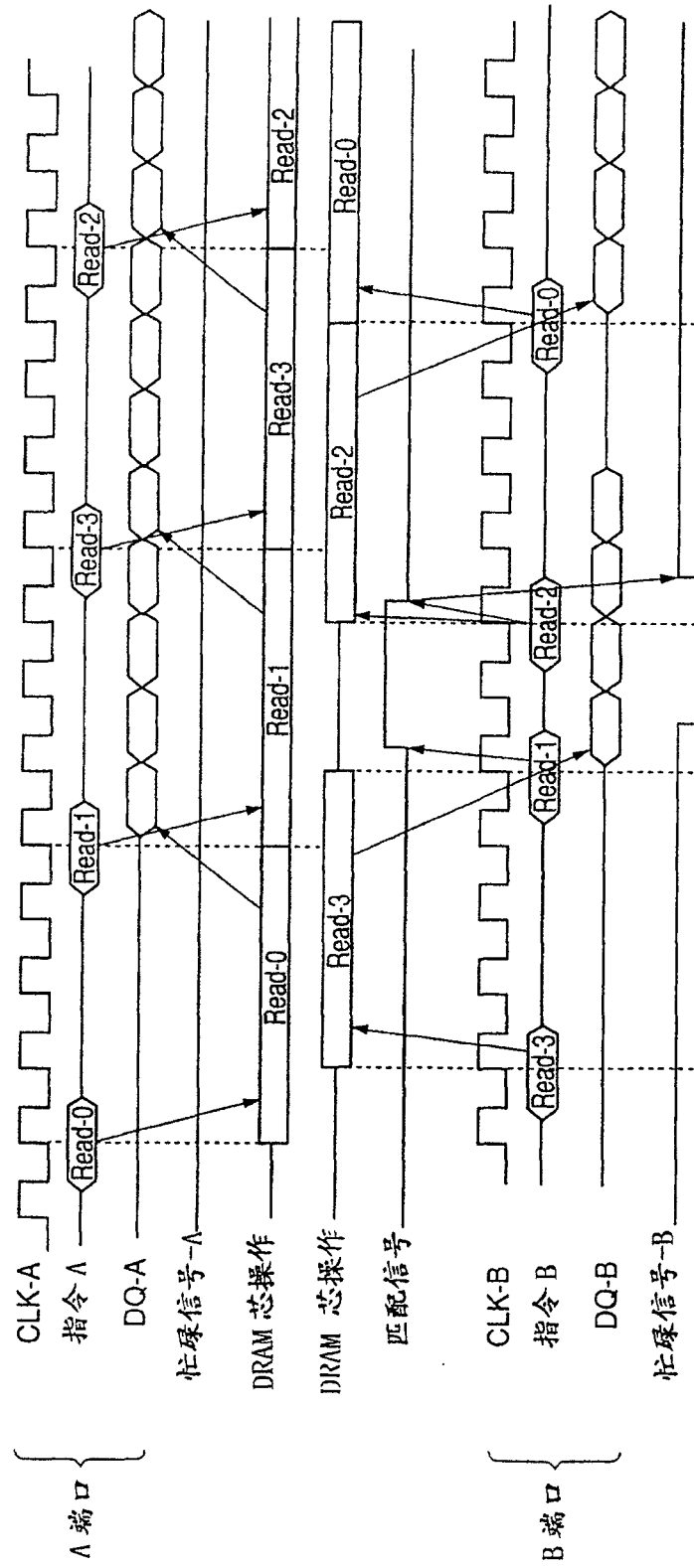
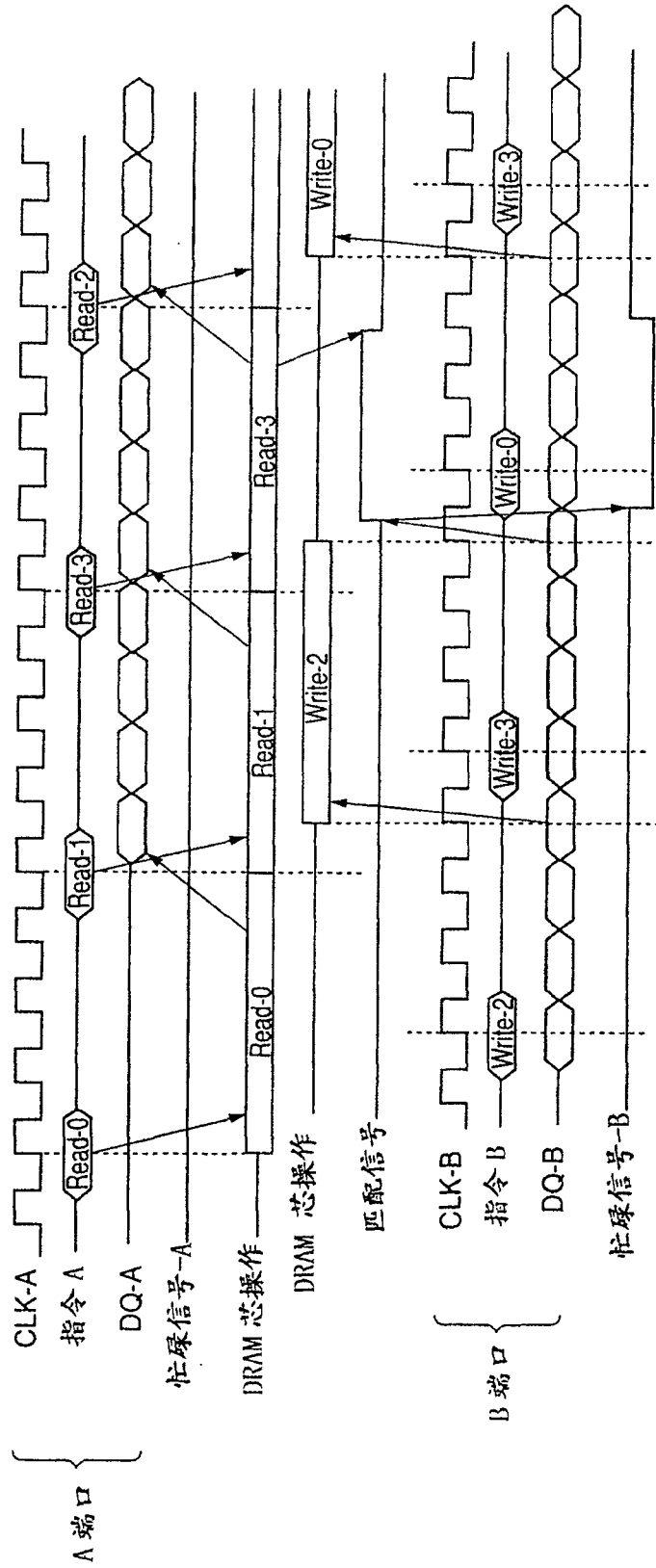


图 104



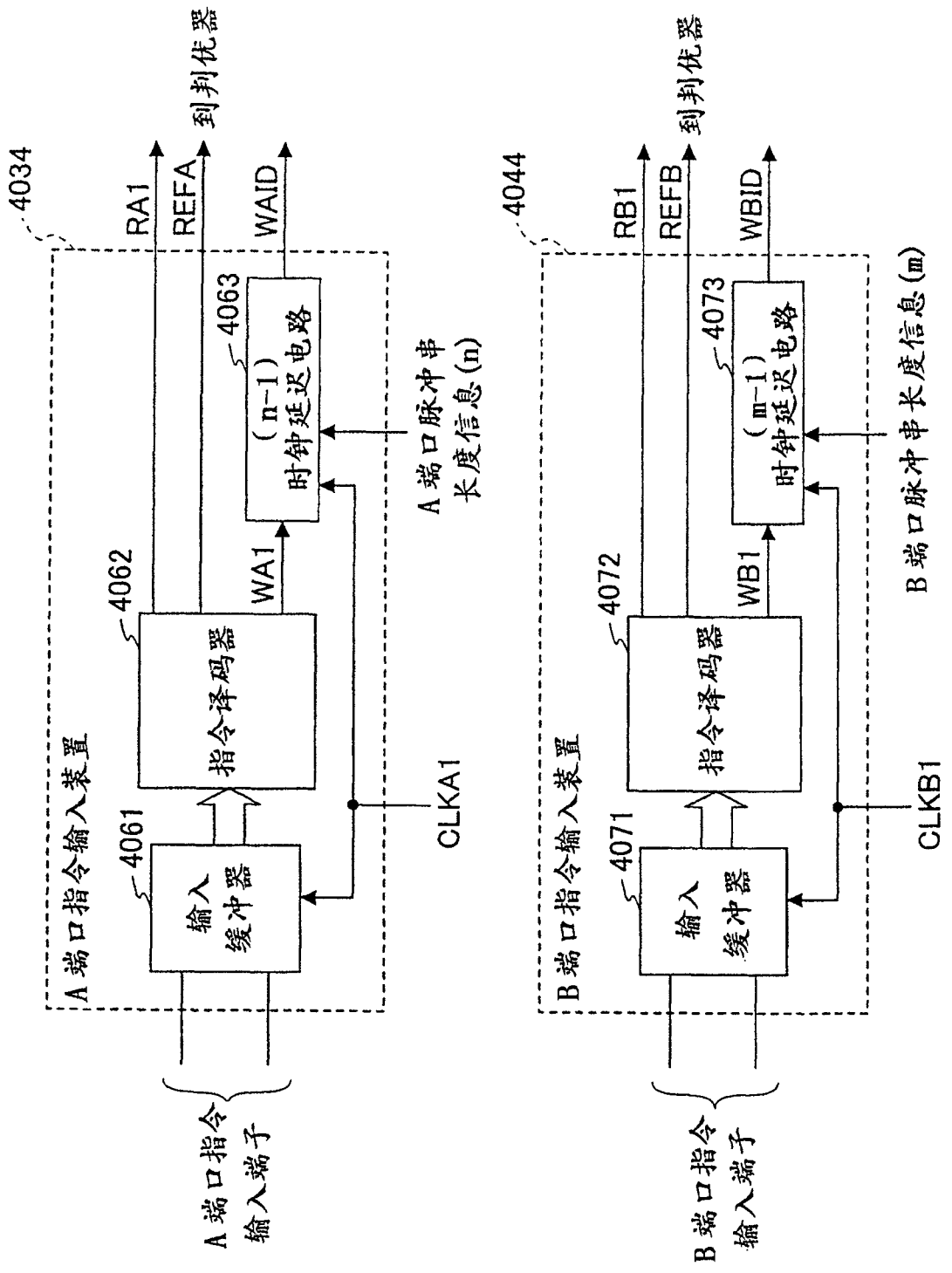
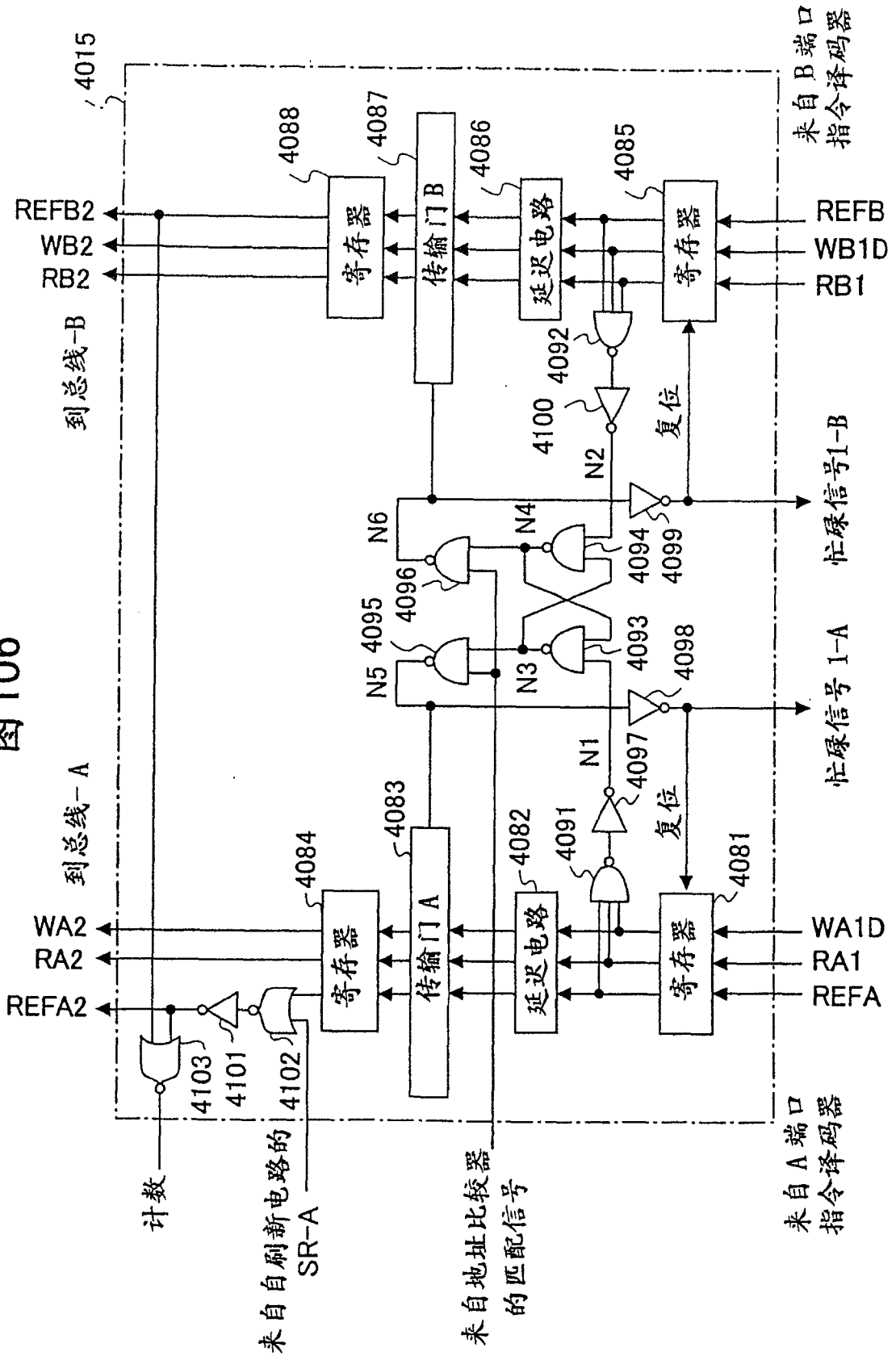


图 105

图 106



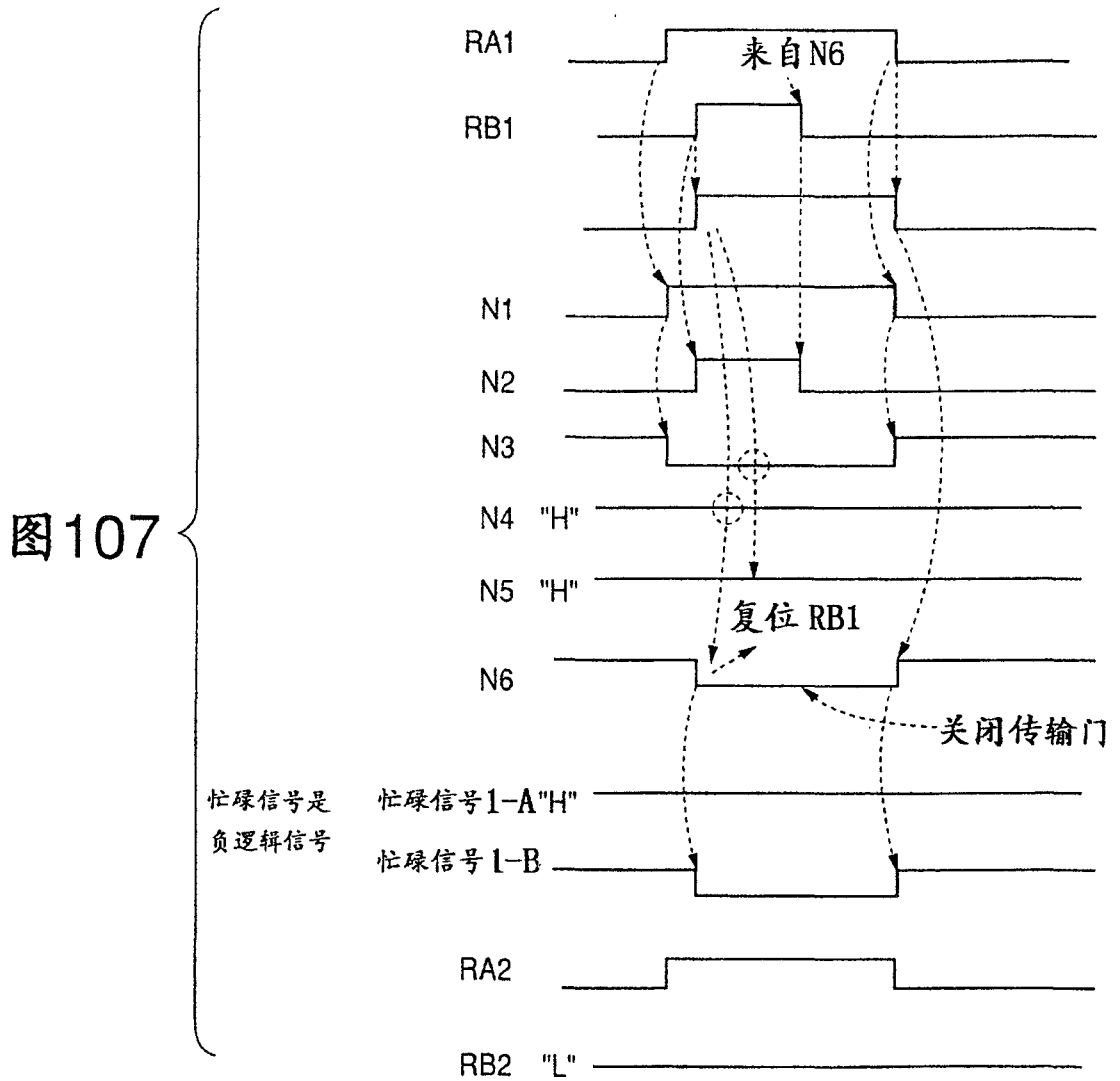


图 108

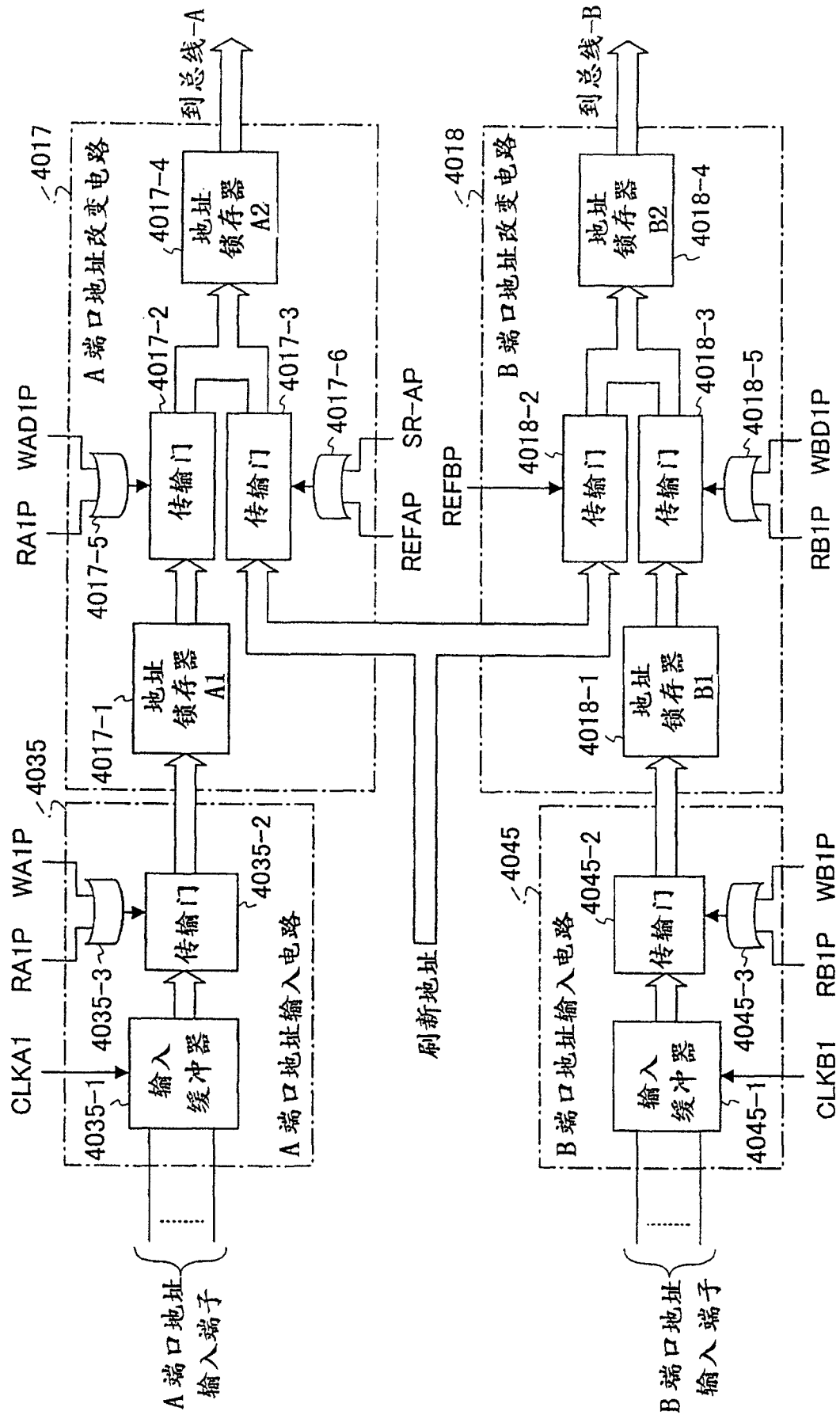


图 109

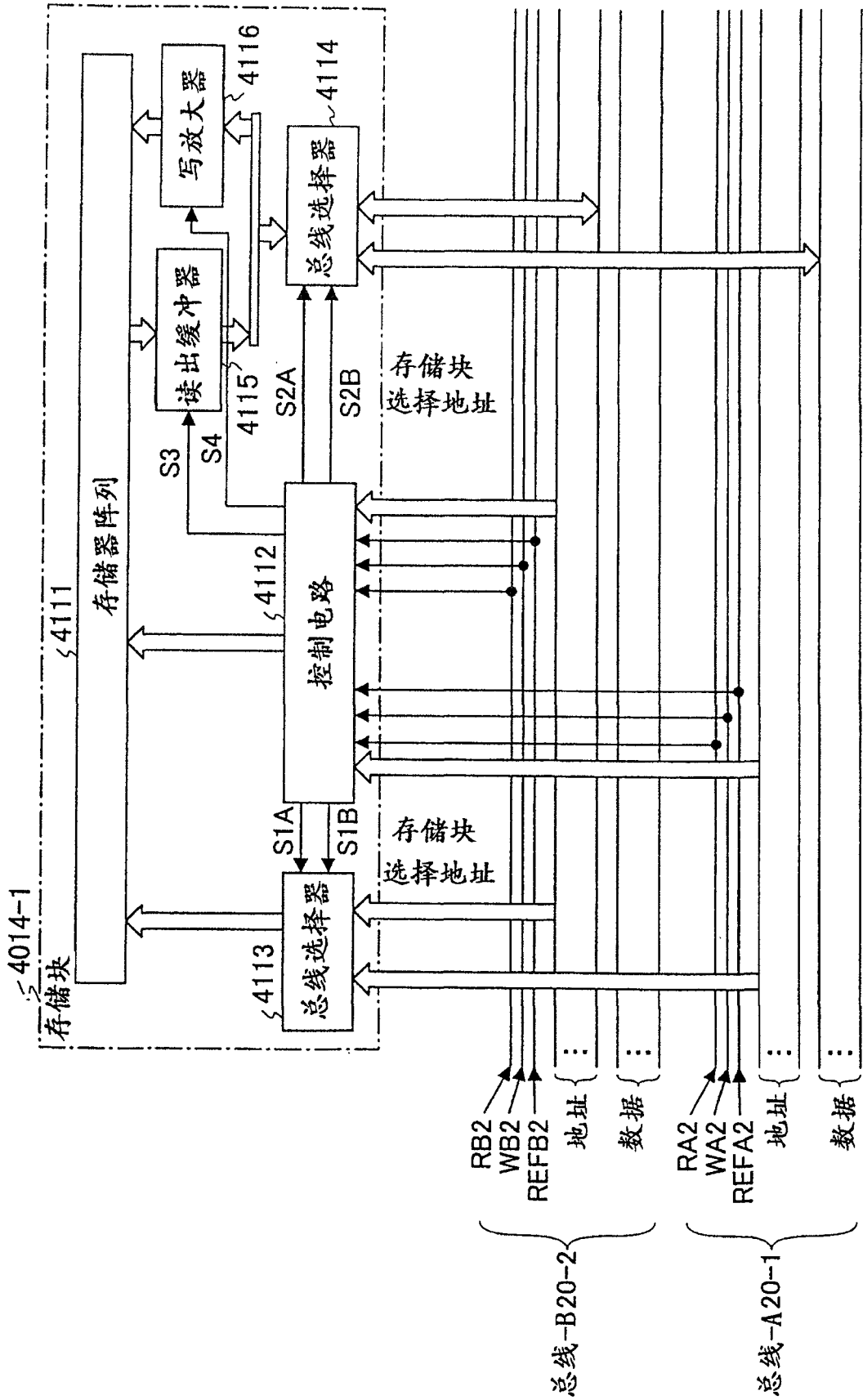


图110A

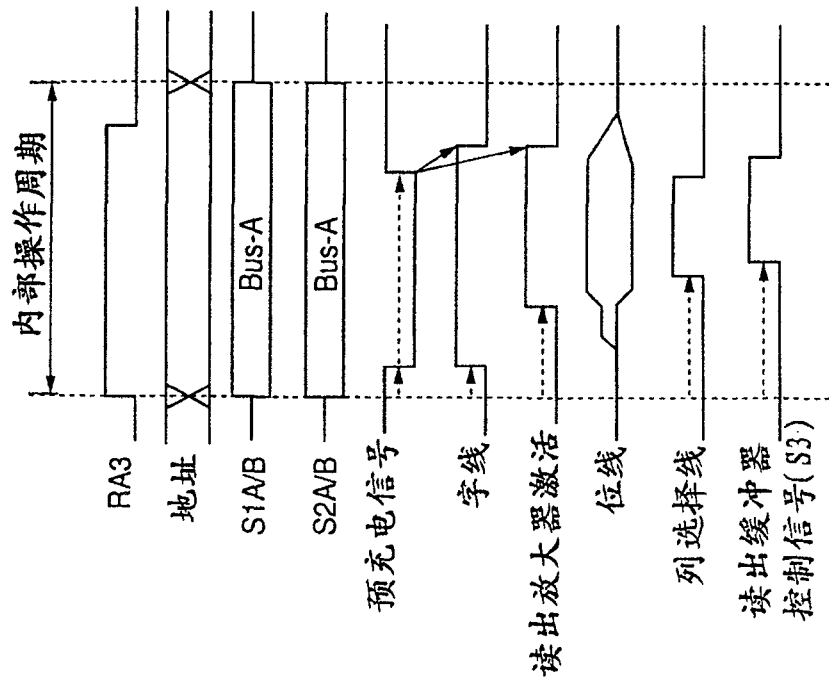


图110B

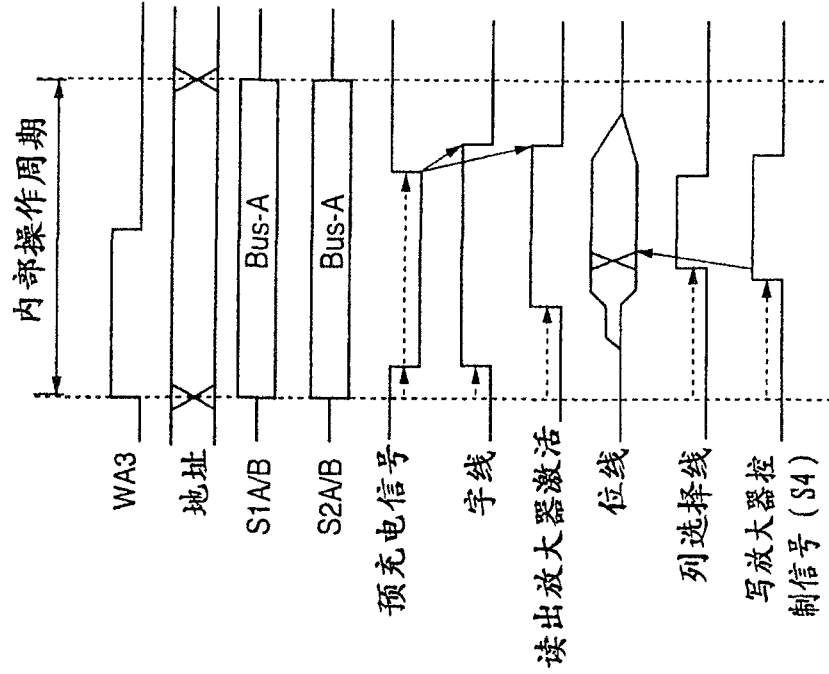


图111

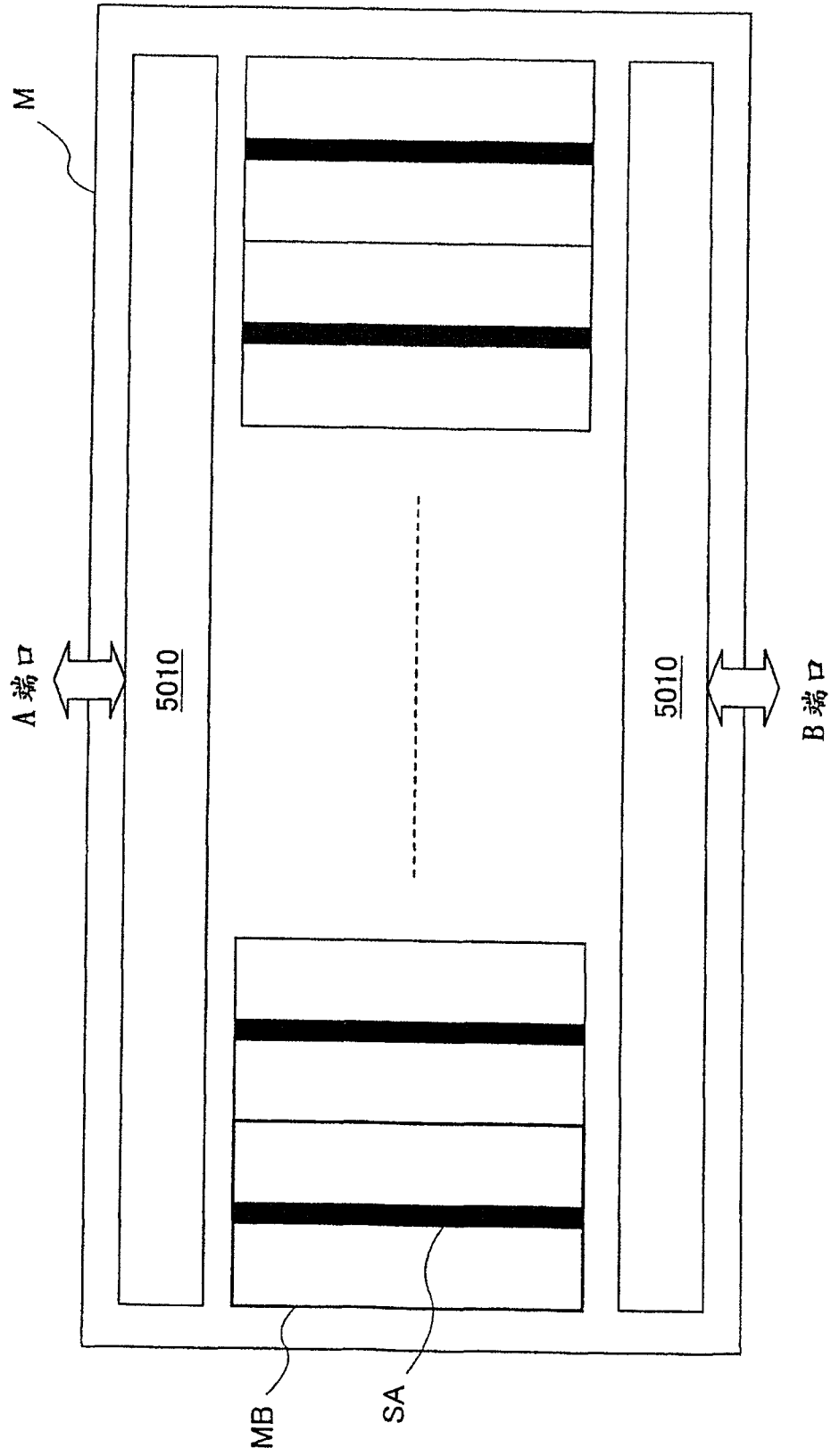


图 112

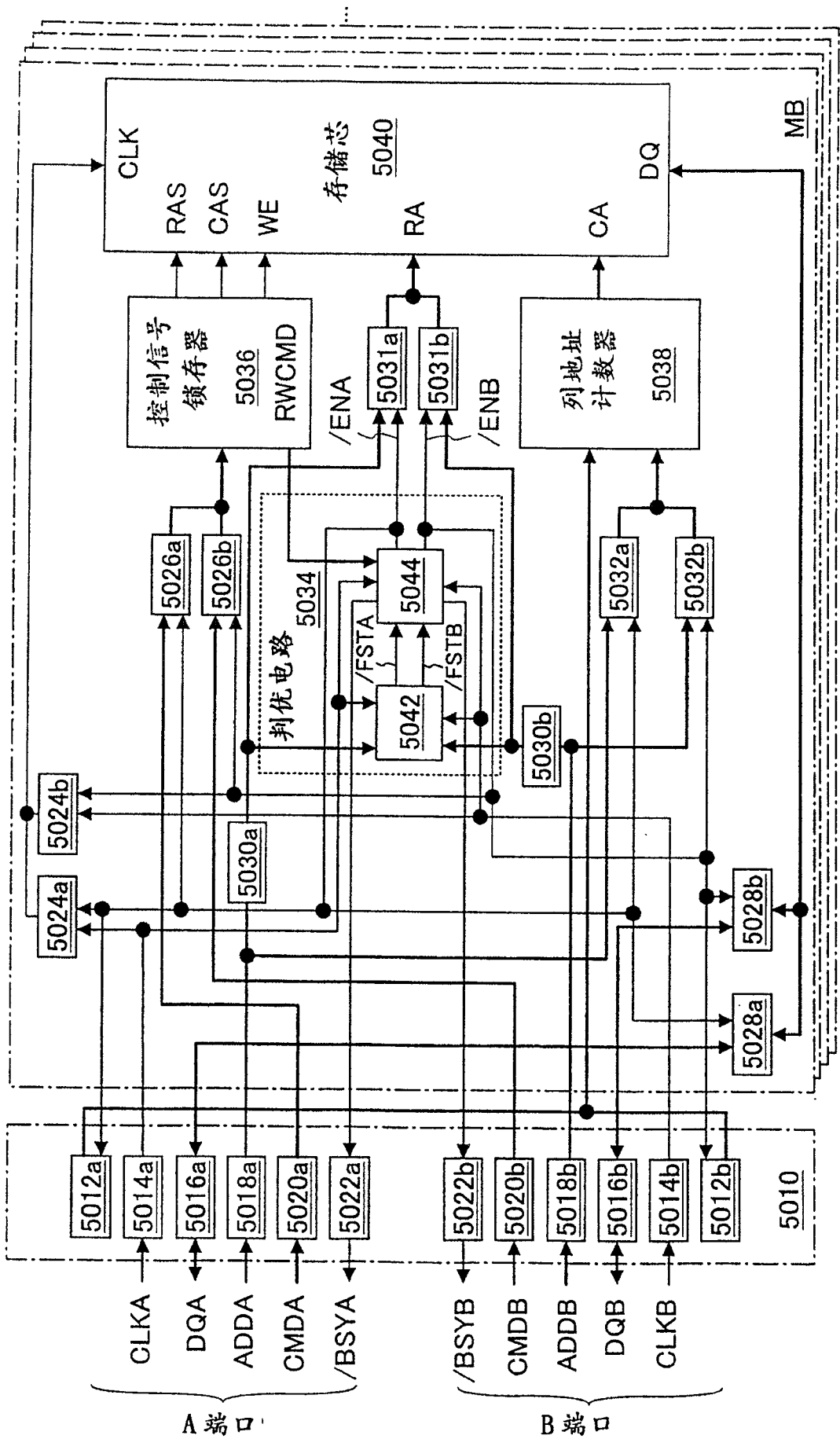


图113

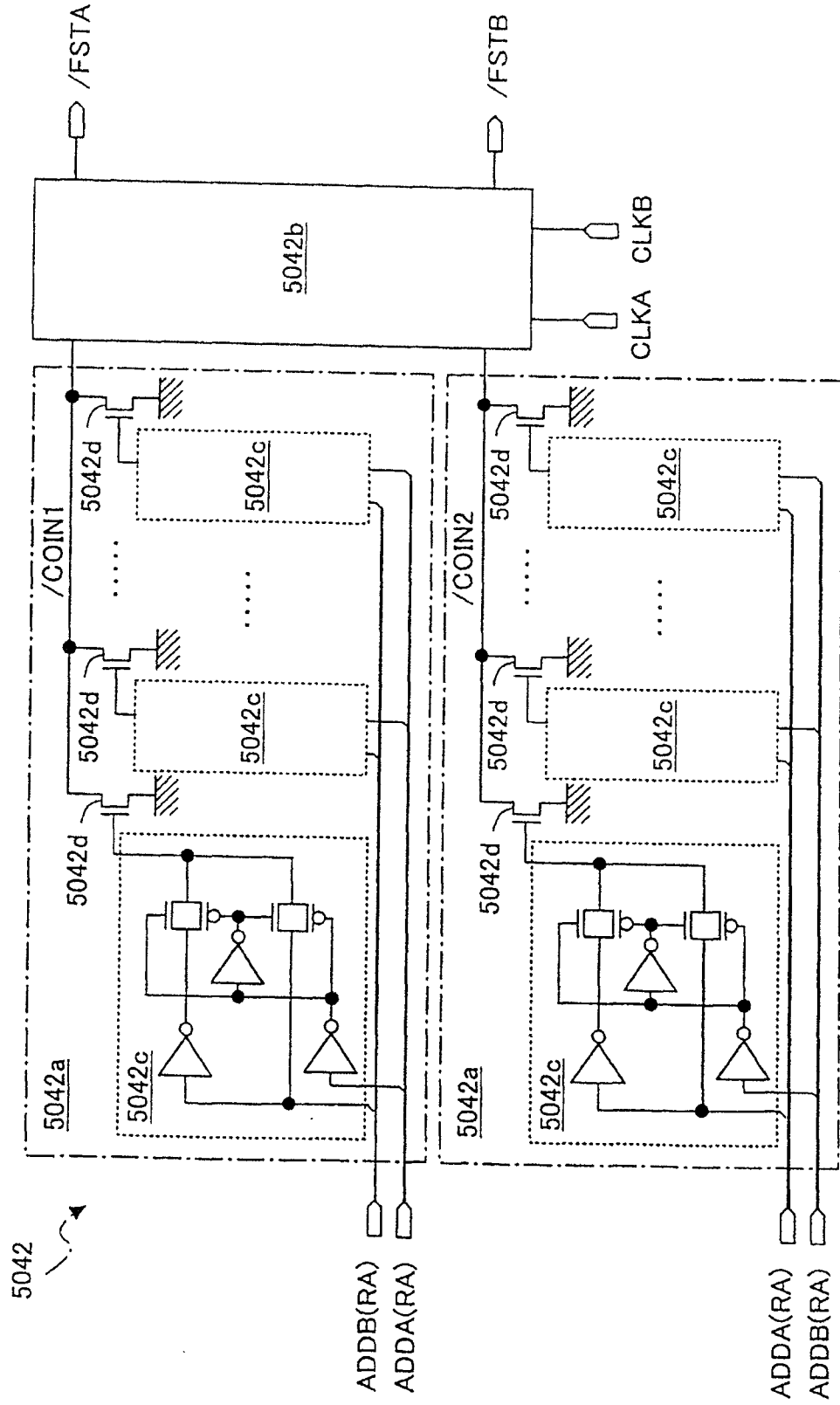
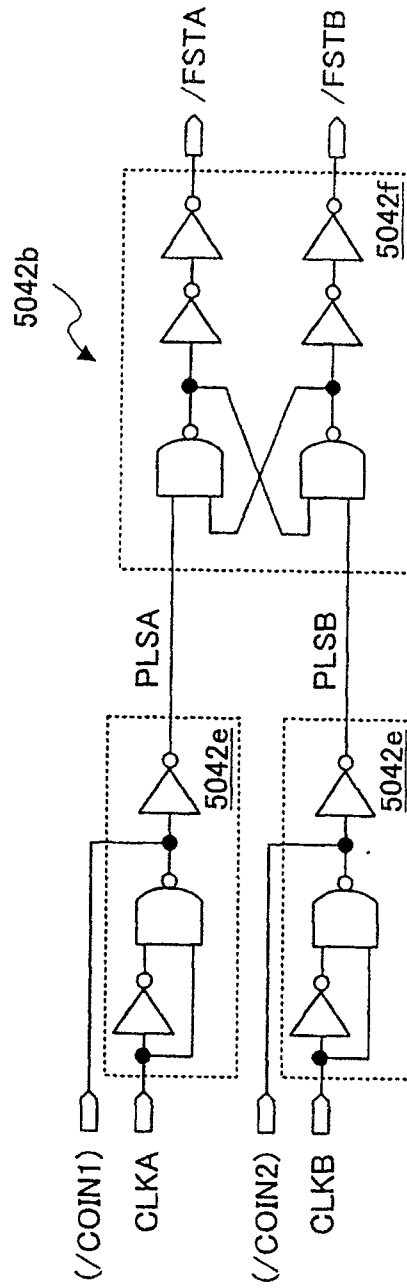


图114



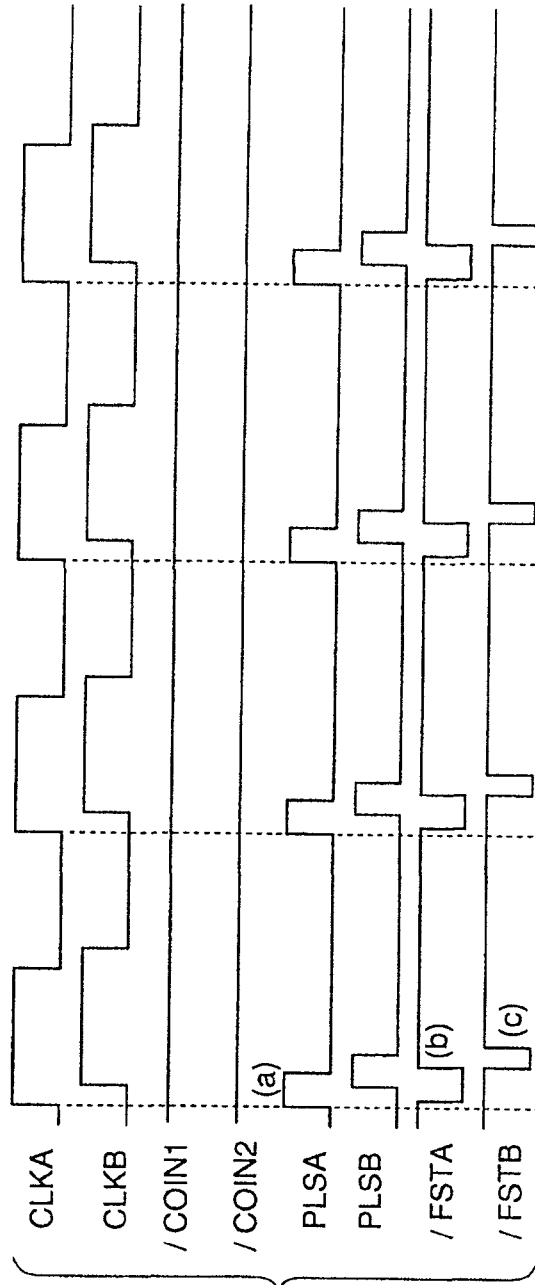


图 115

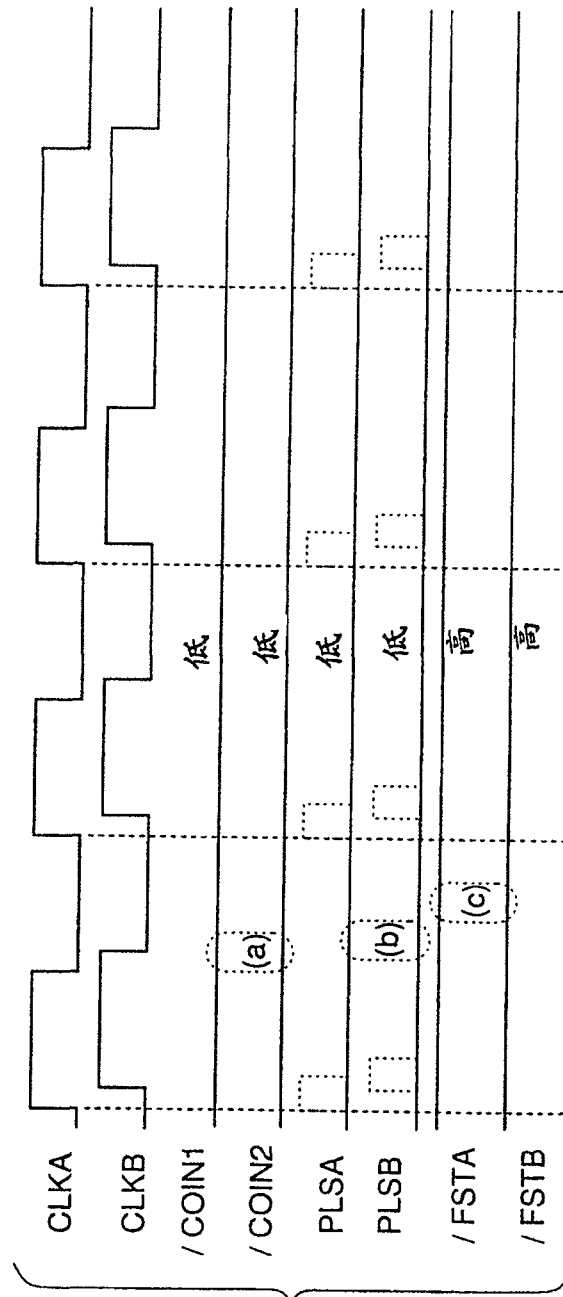


图 116

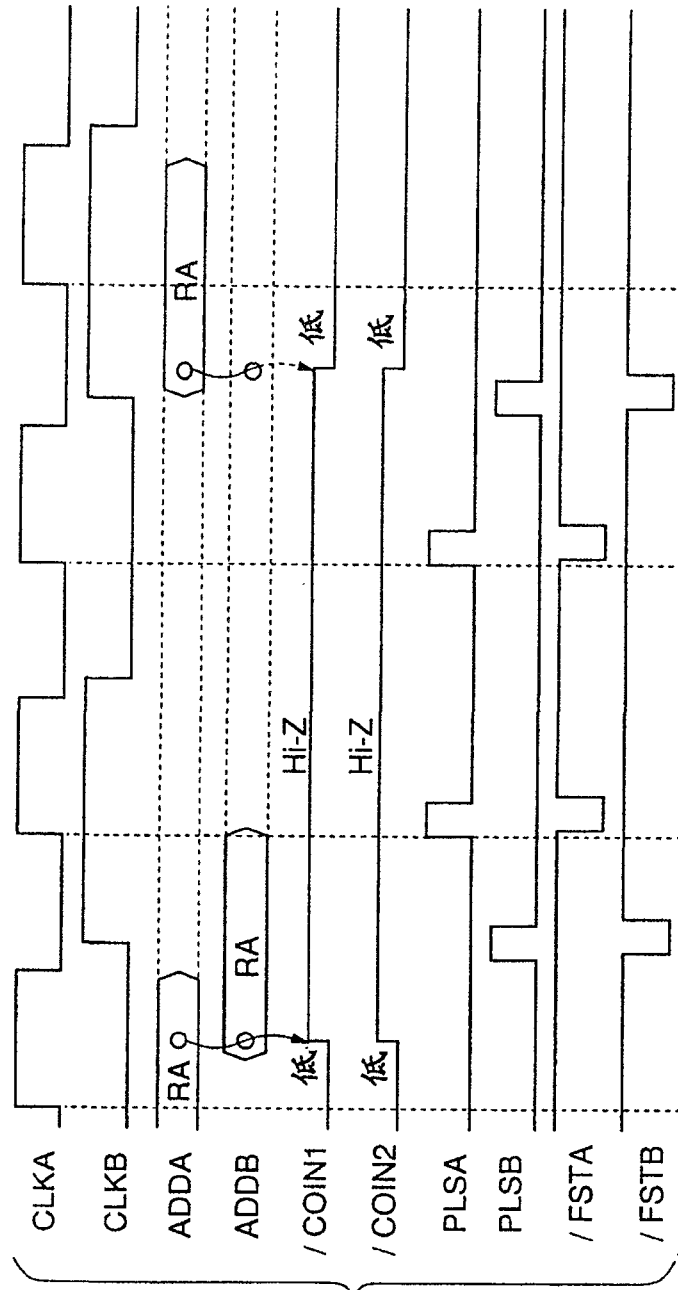
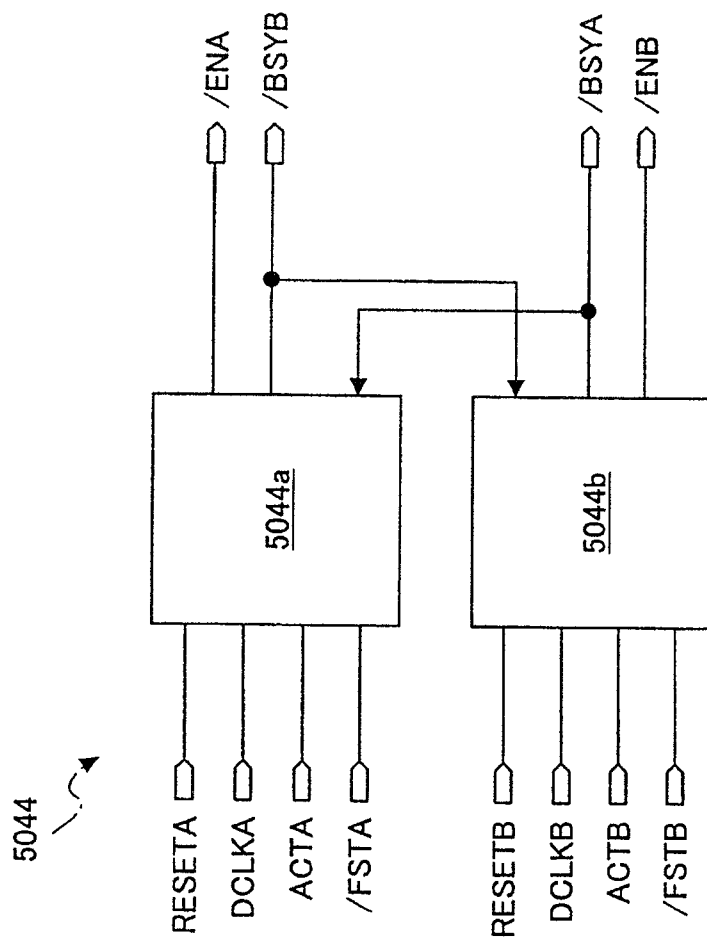


图117

图 118



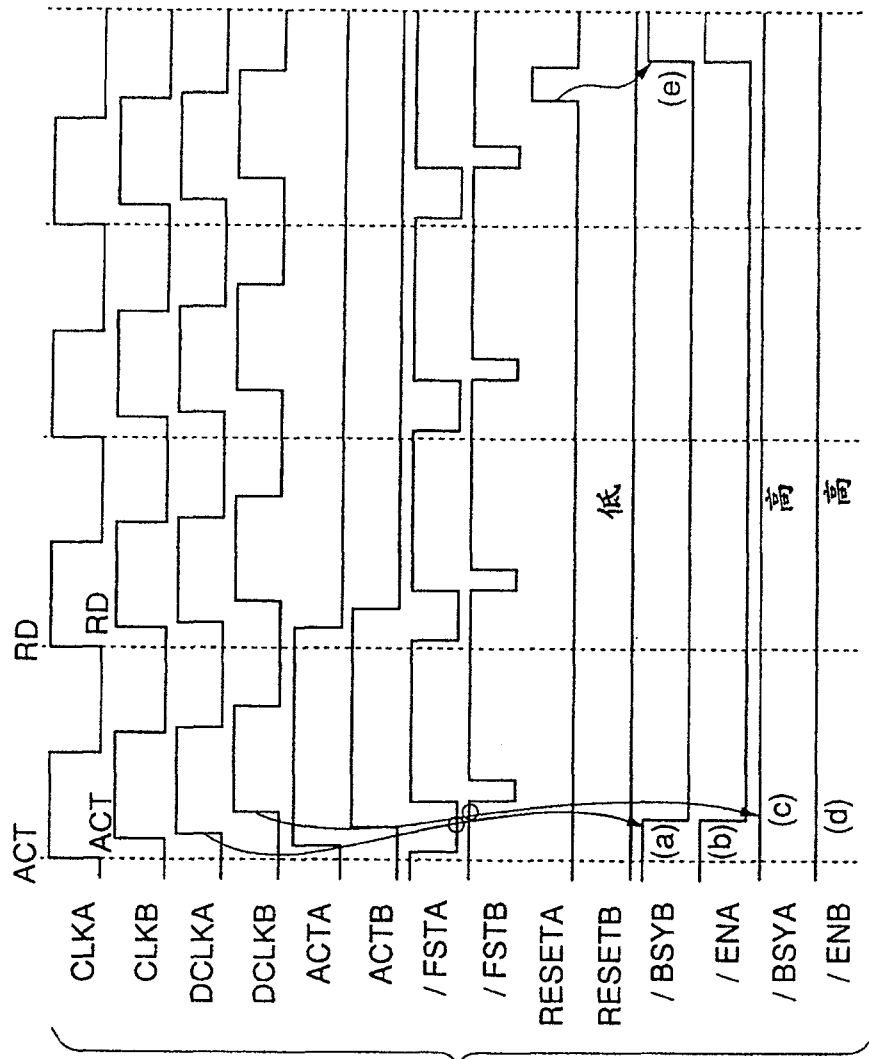


图 119

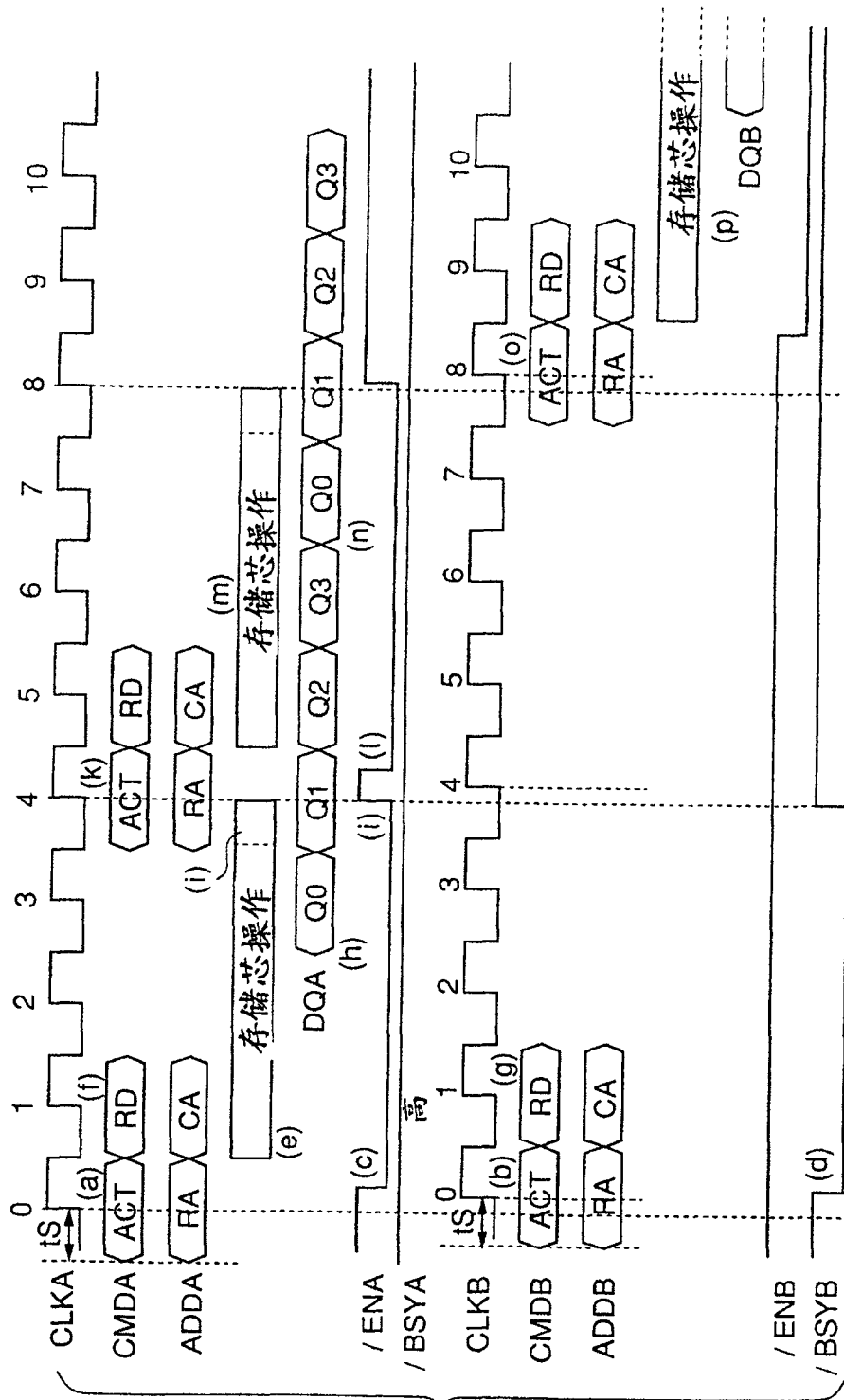


图120

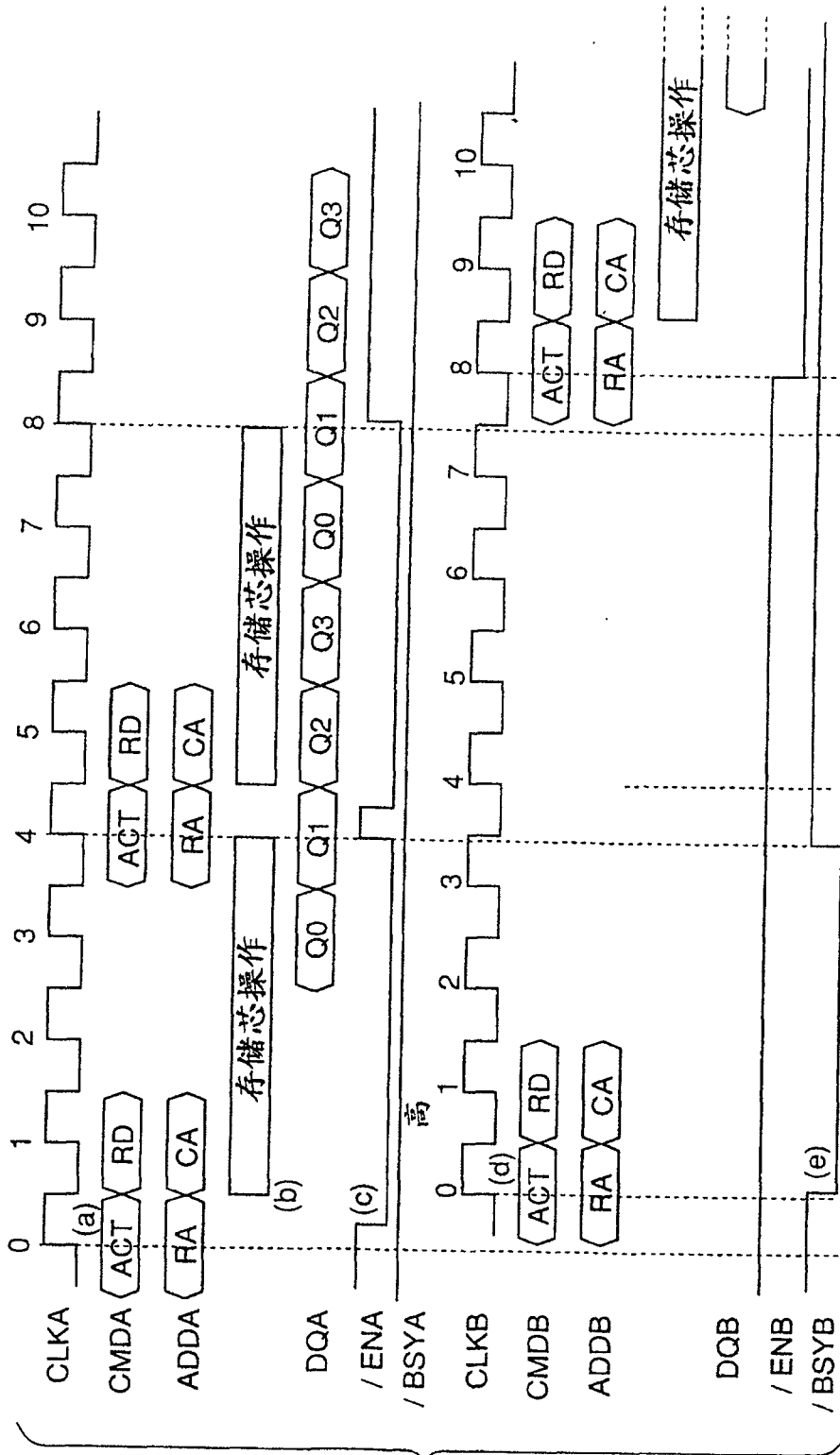


图 121

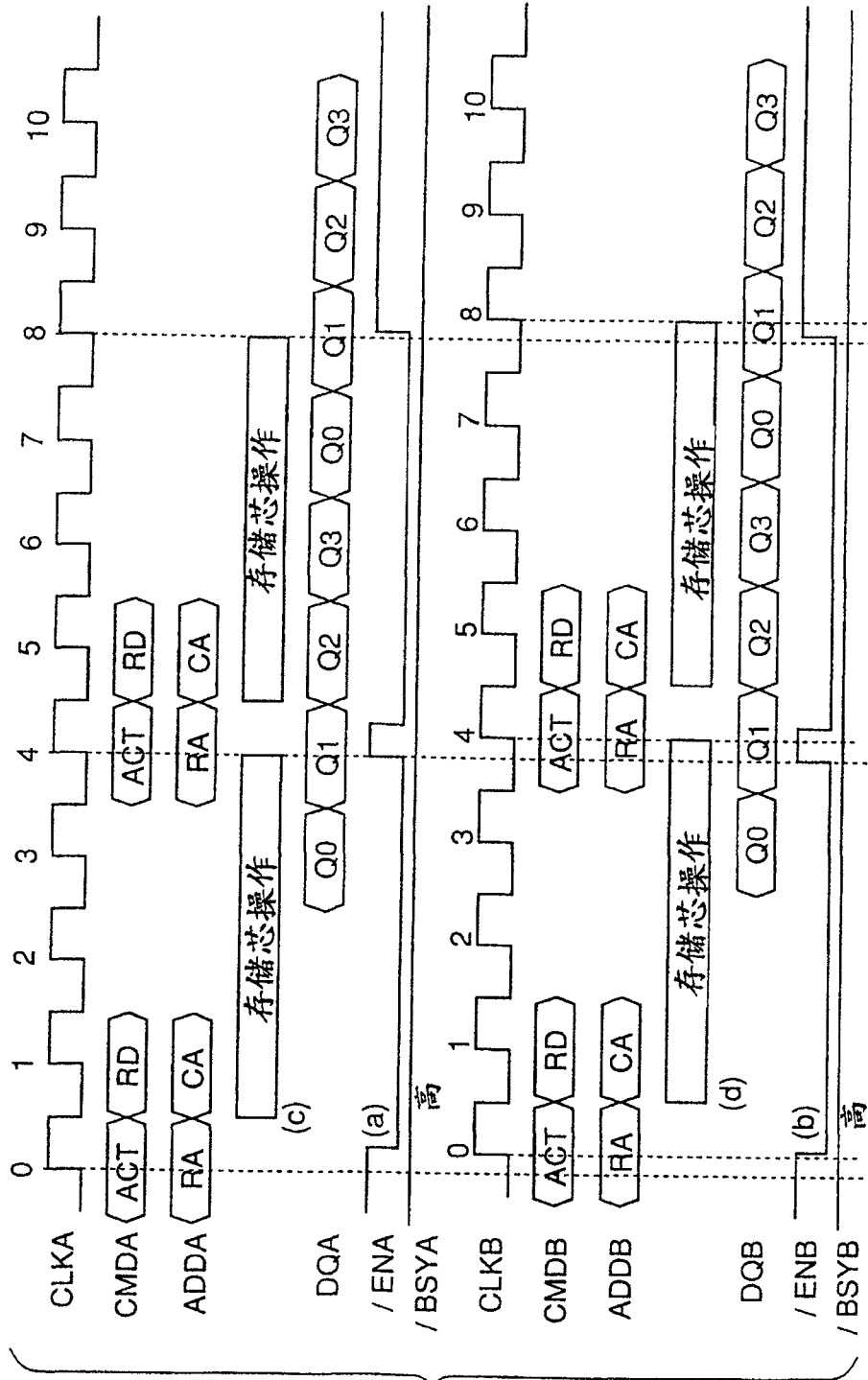


图122

图123

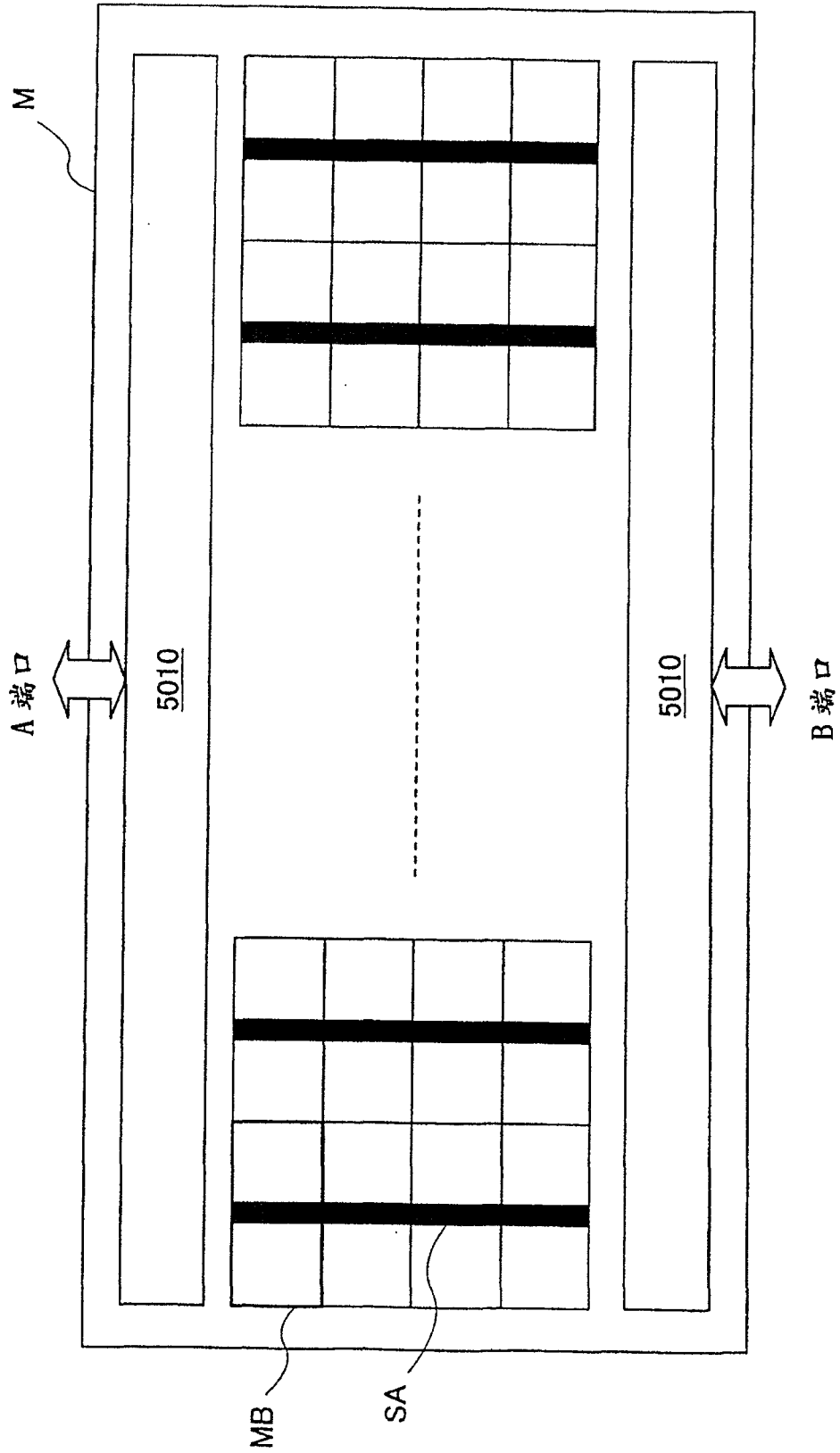


图 124

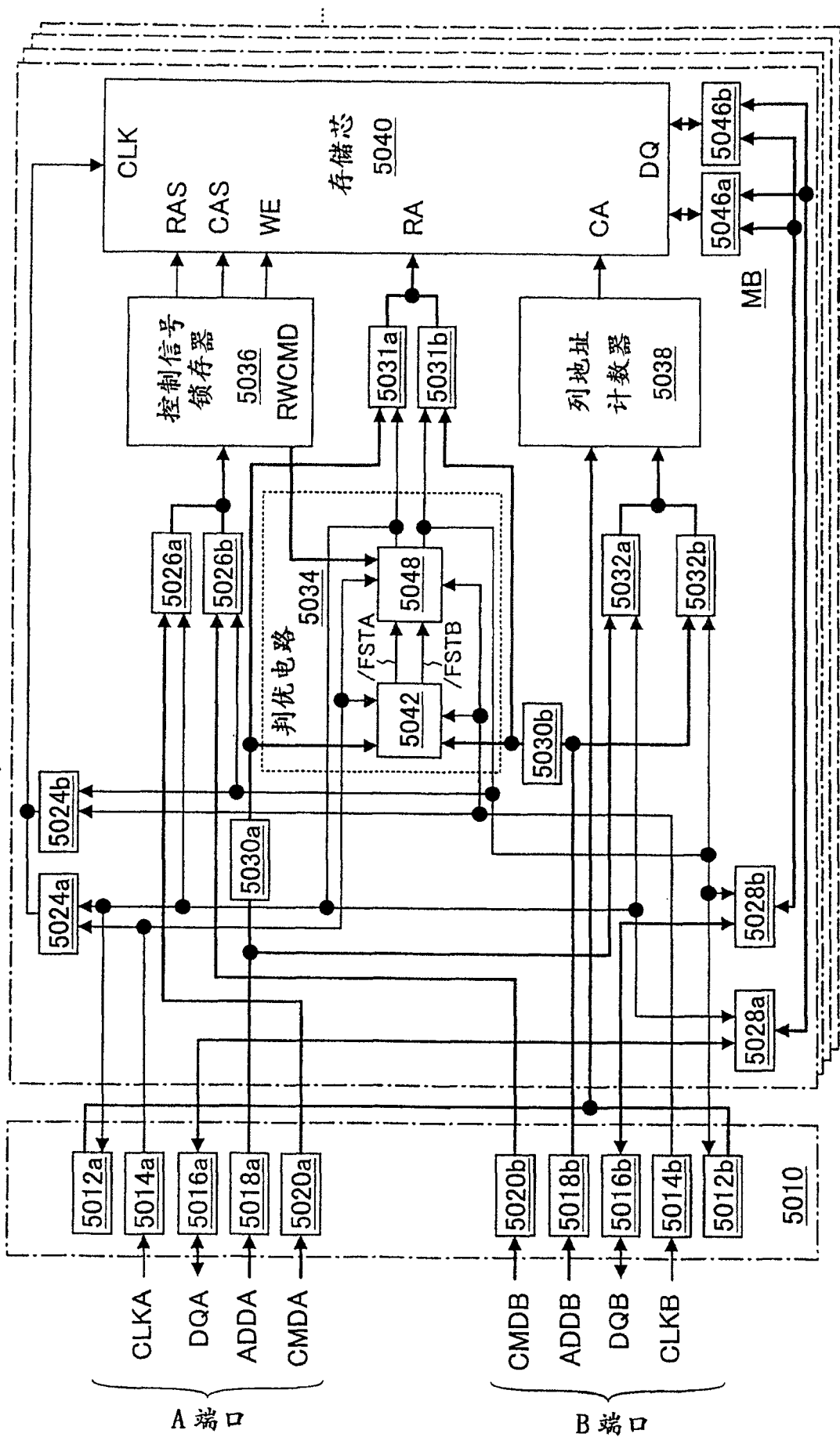
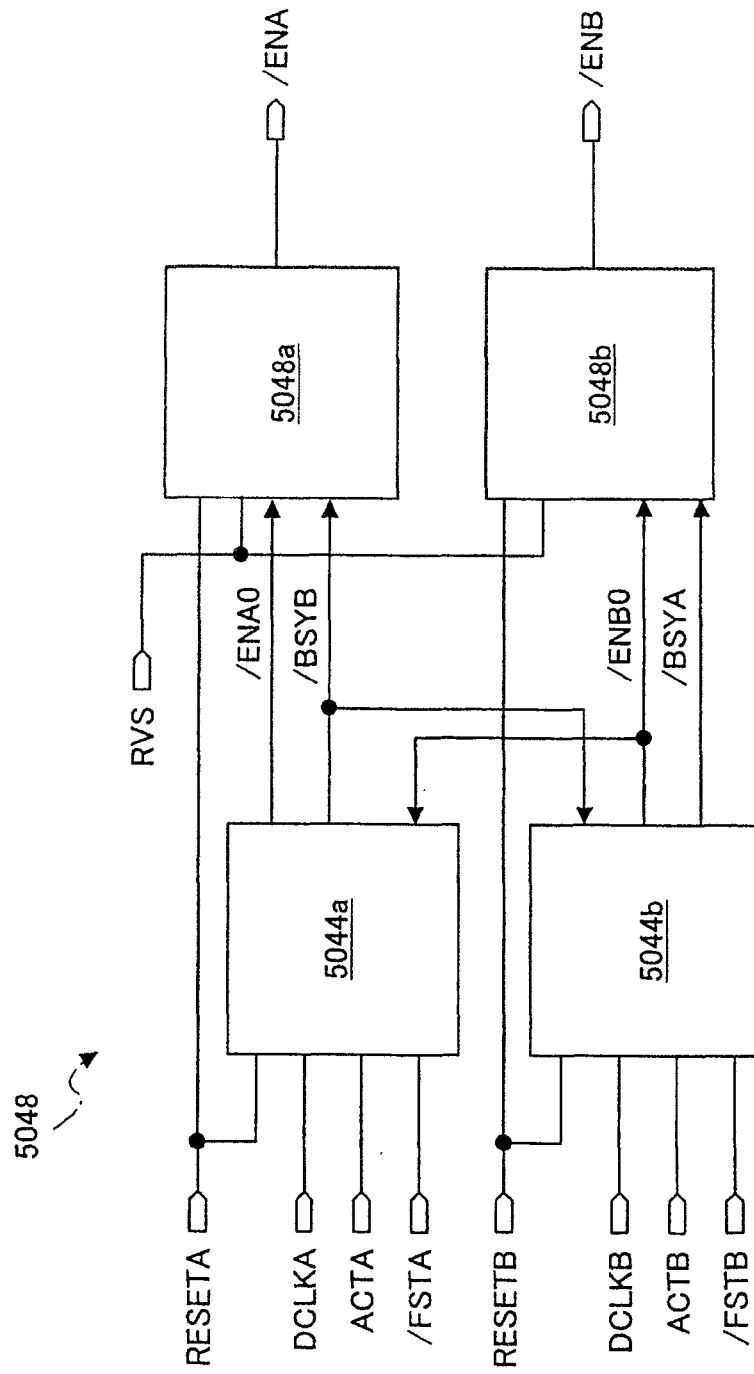


图125



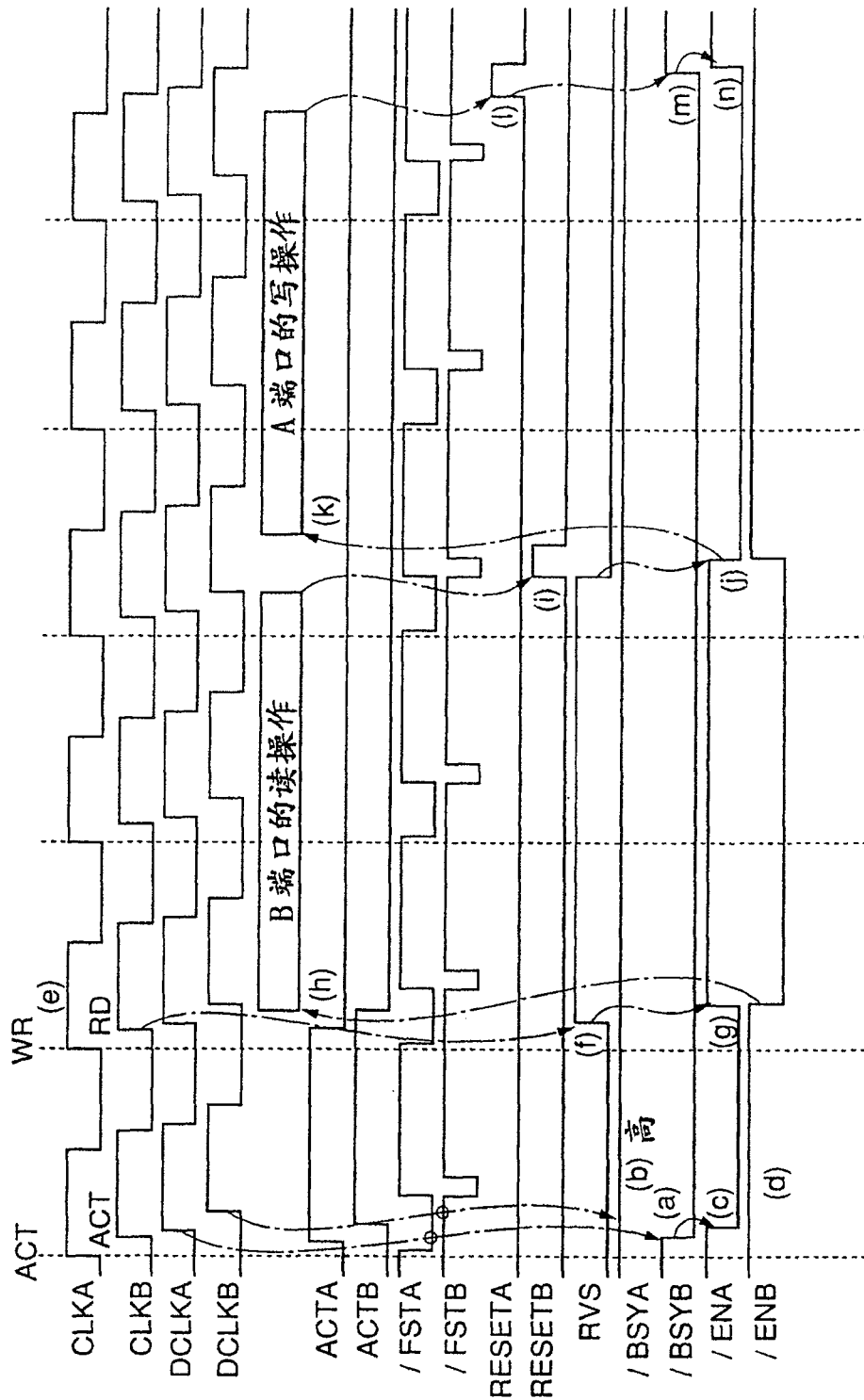


图 126

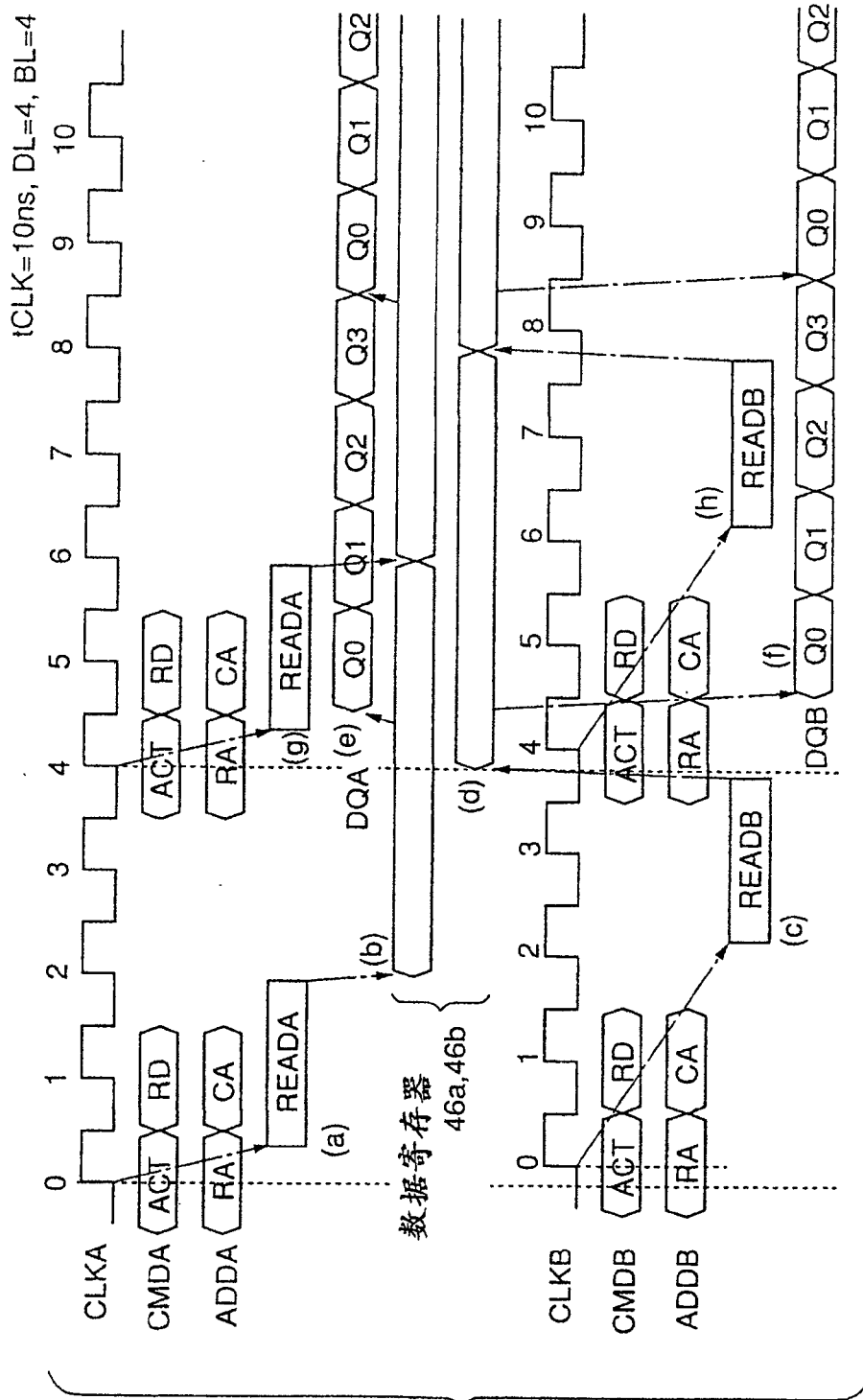


图127

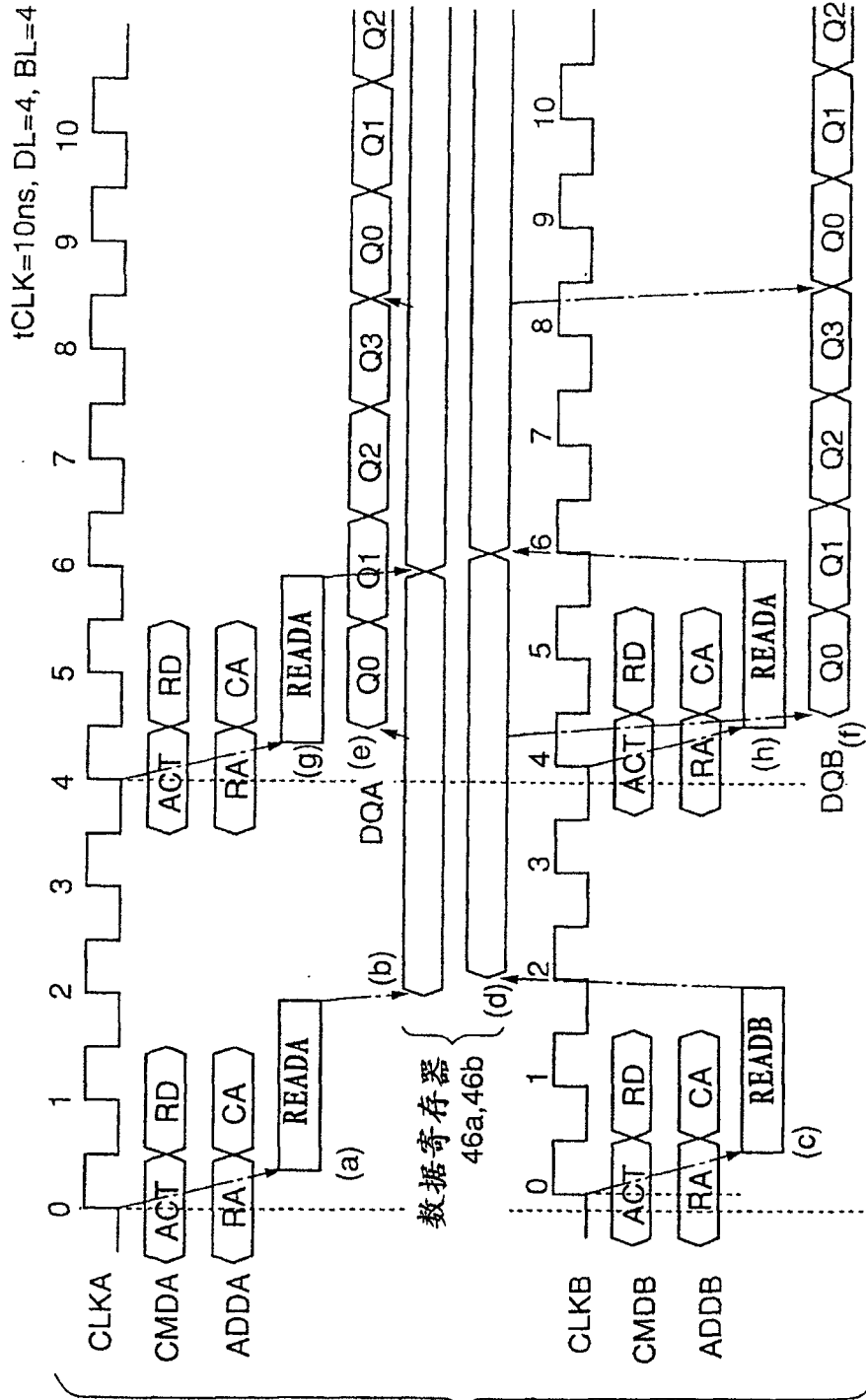


图 128

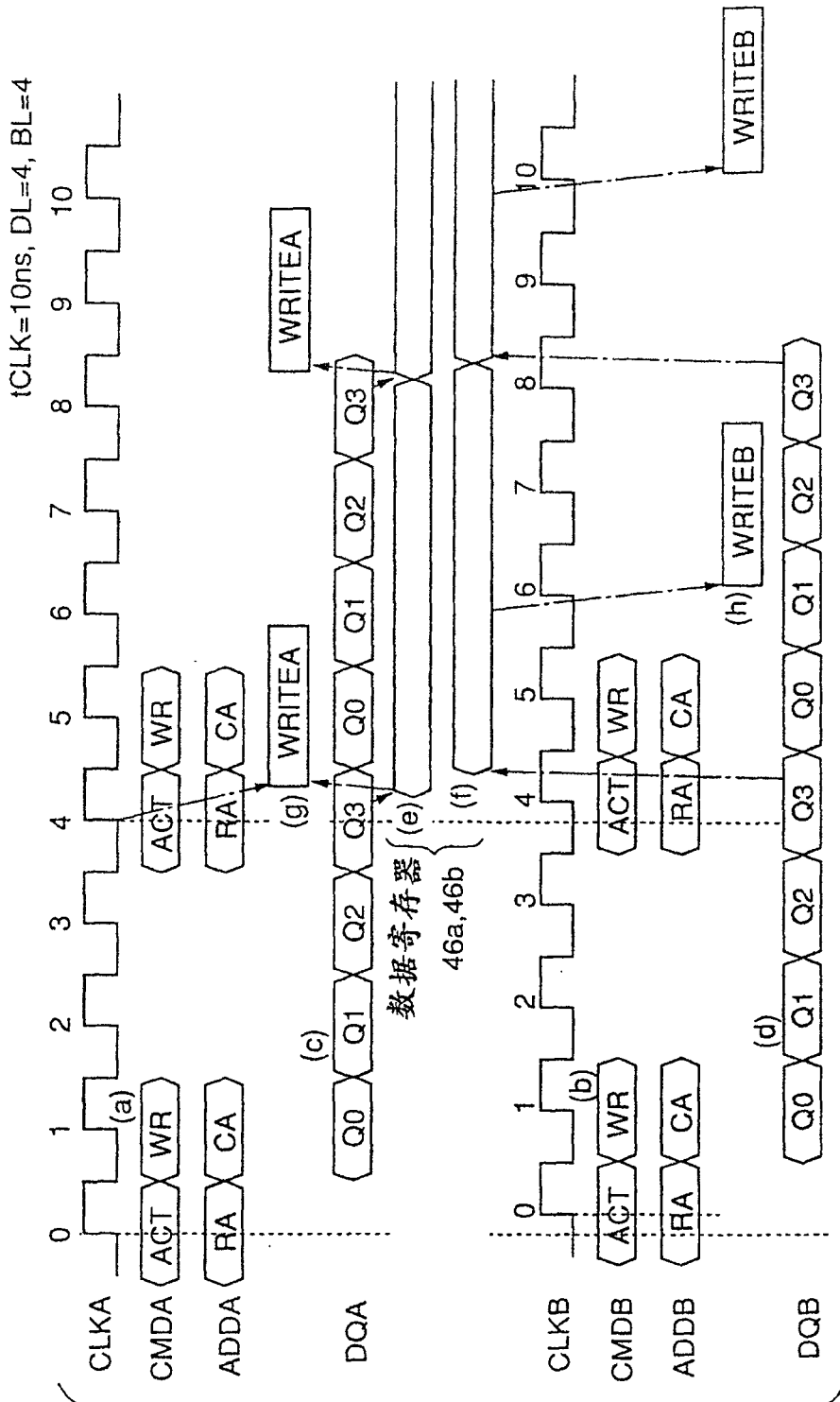


图129

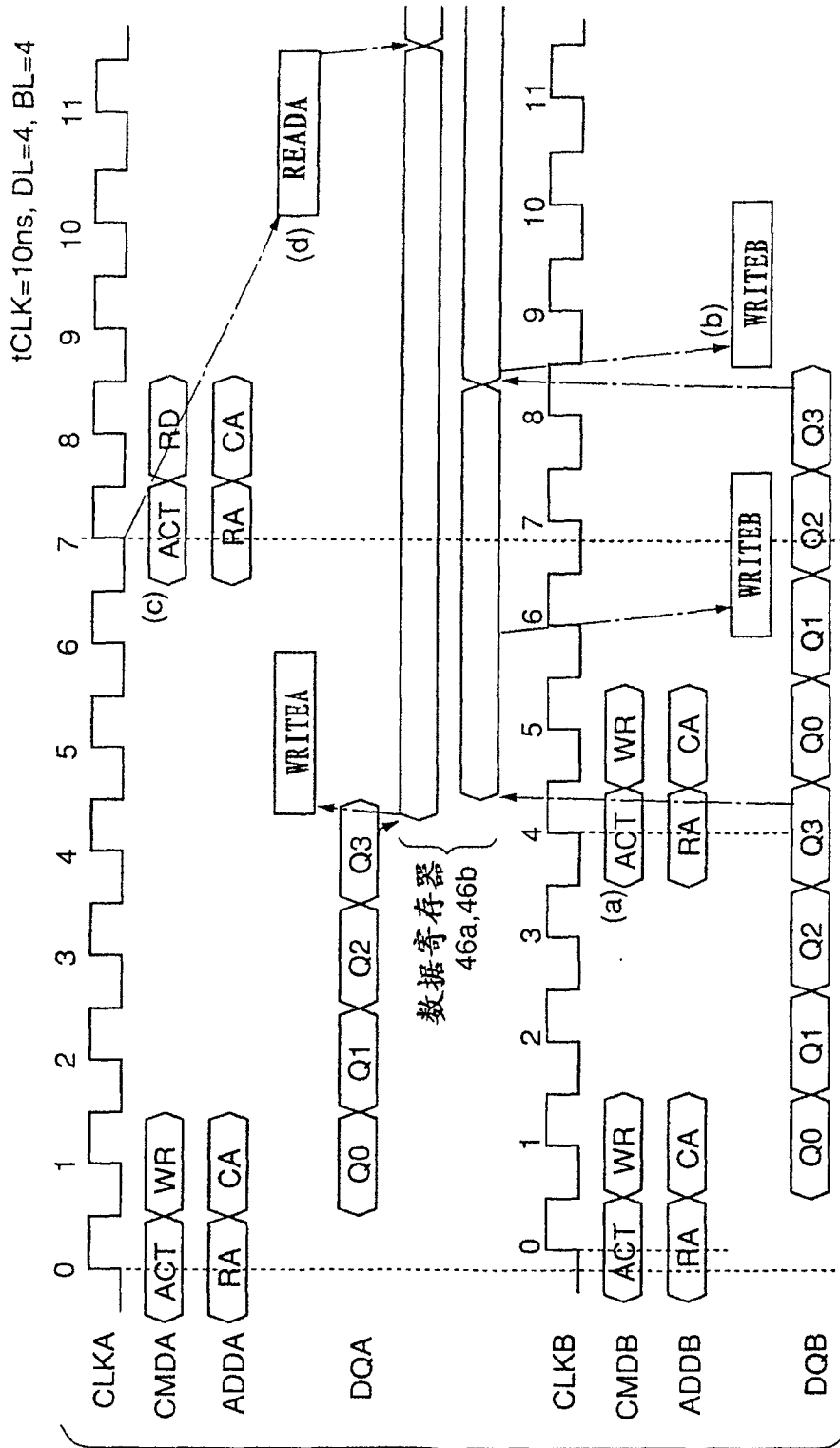


图 130

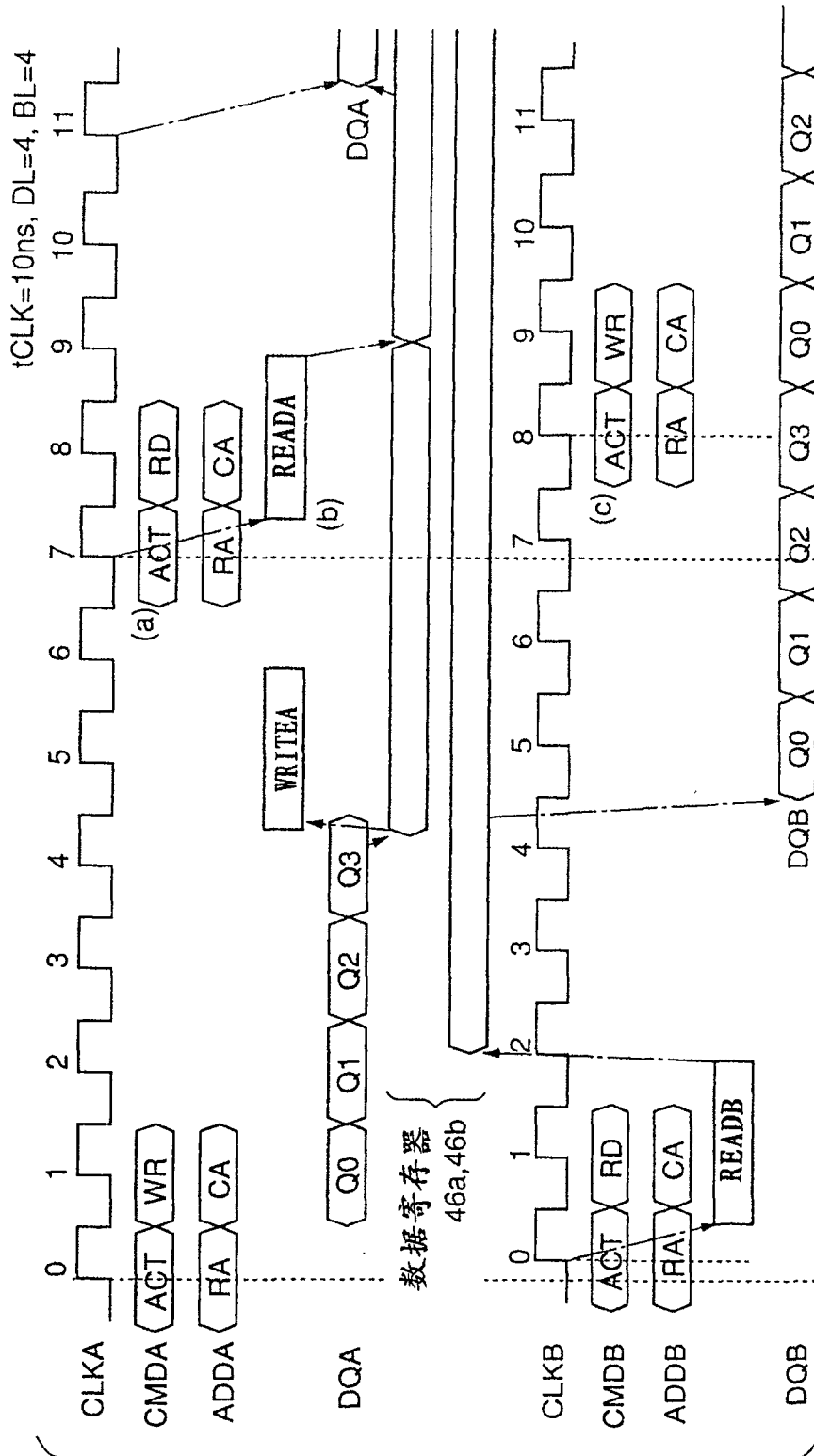


图131

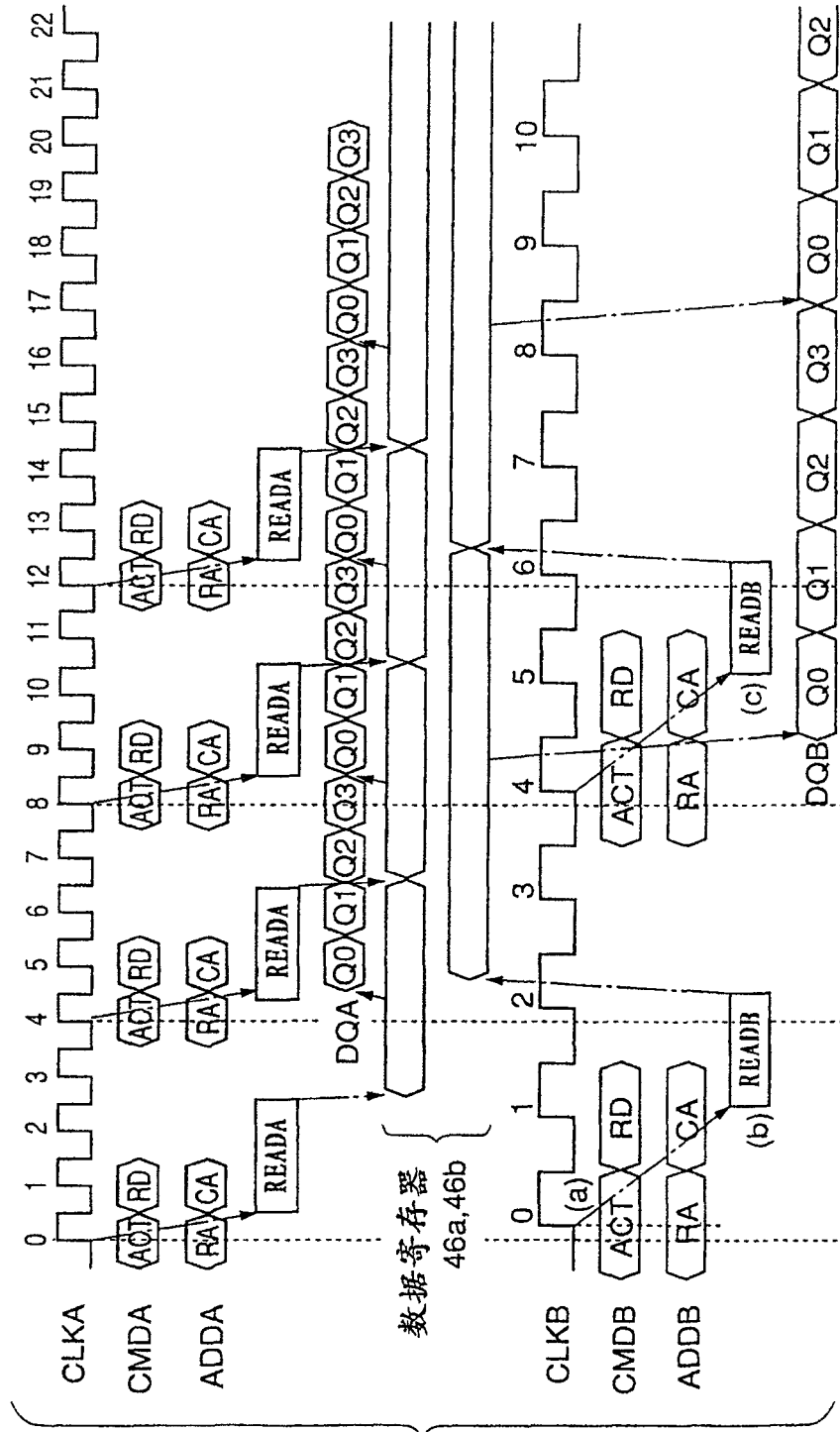
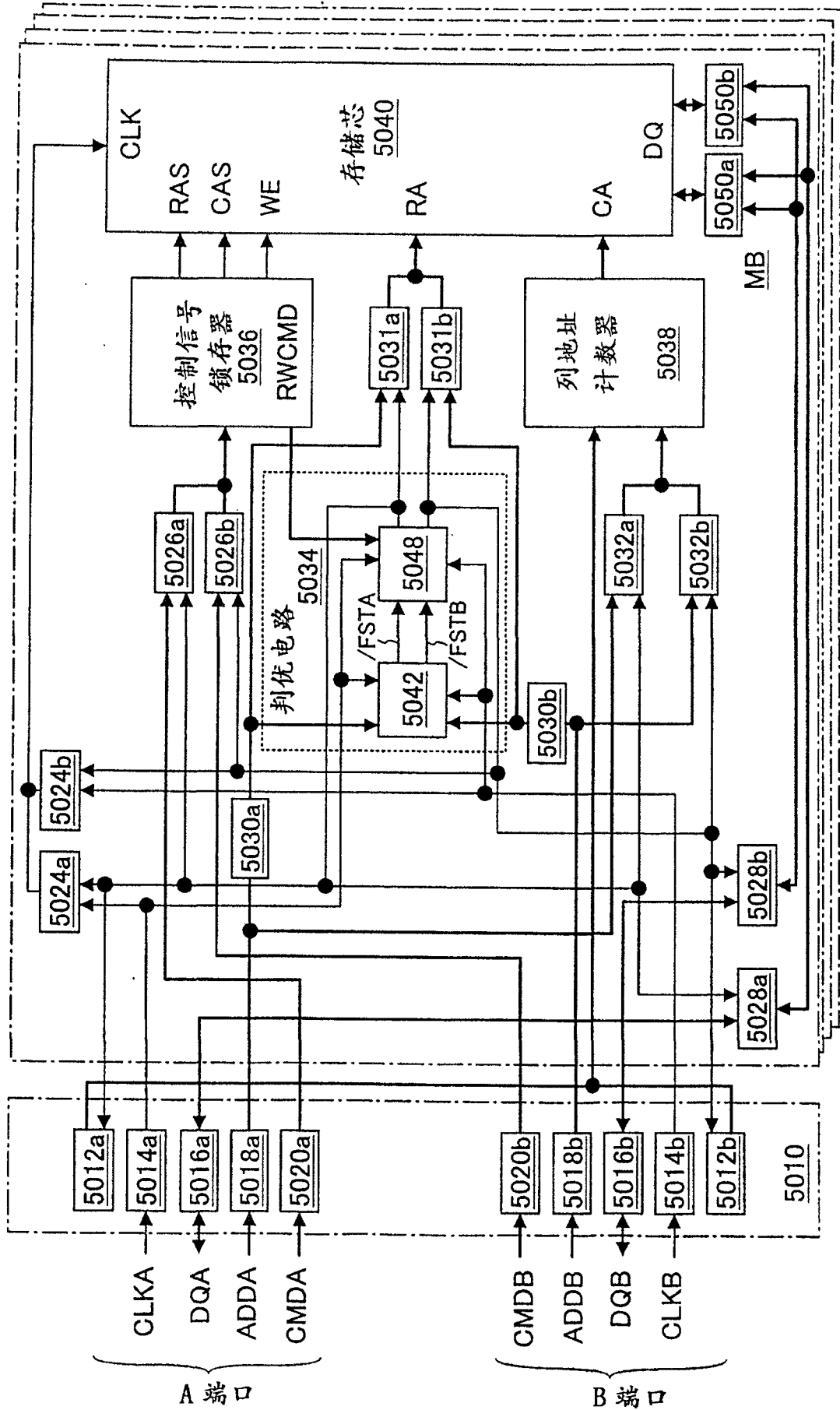


图132

图 133



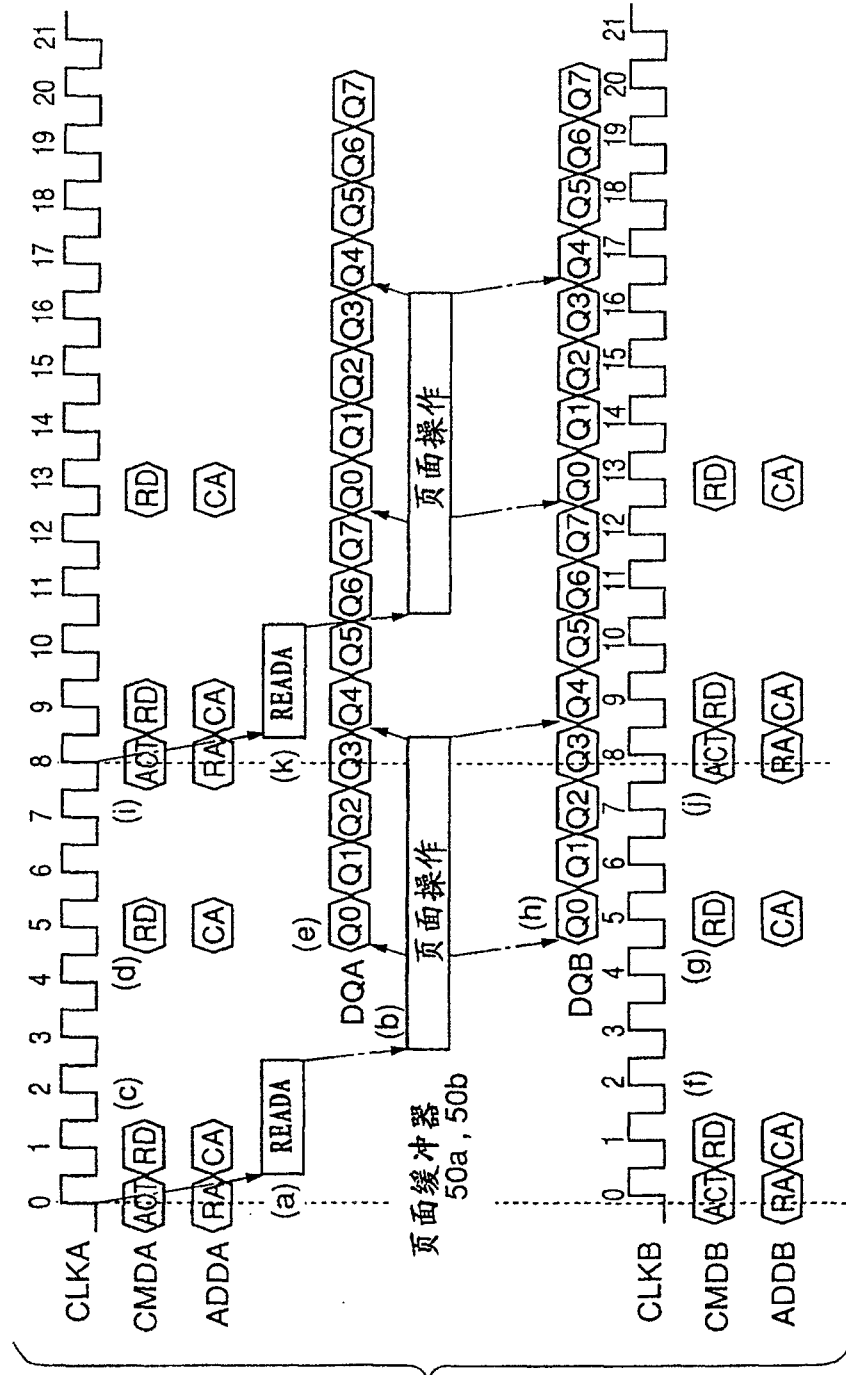


图134

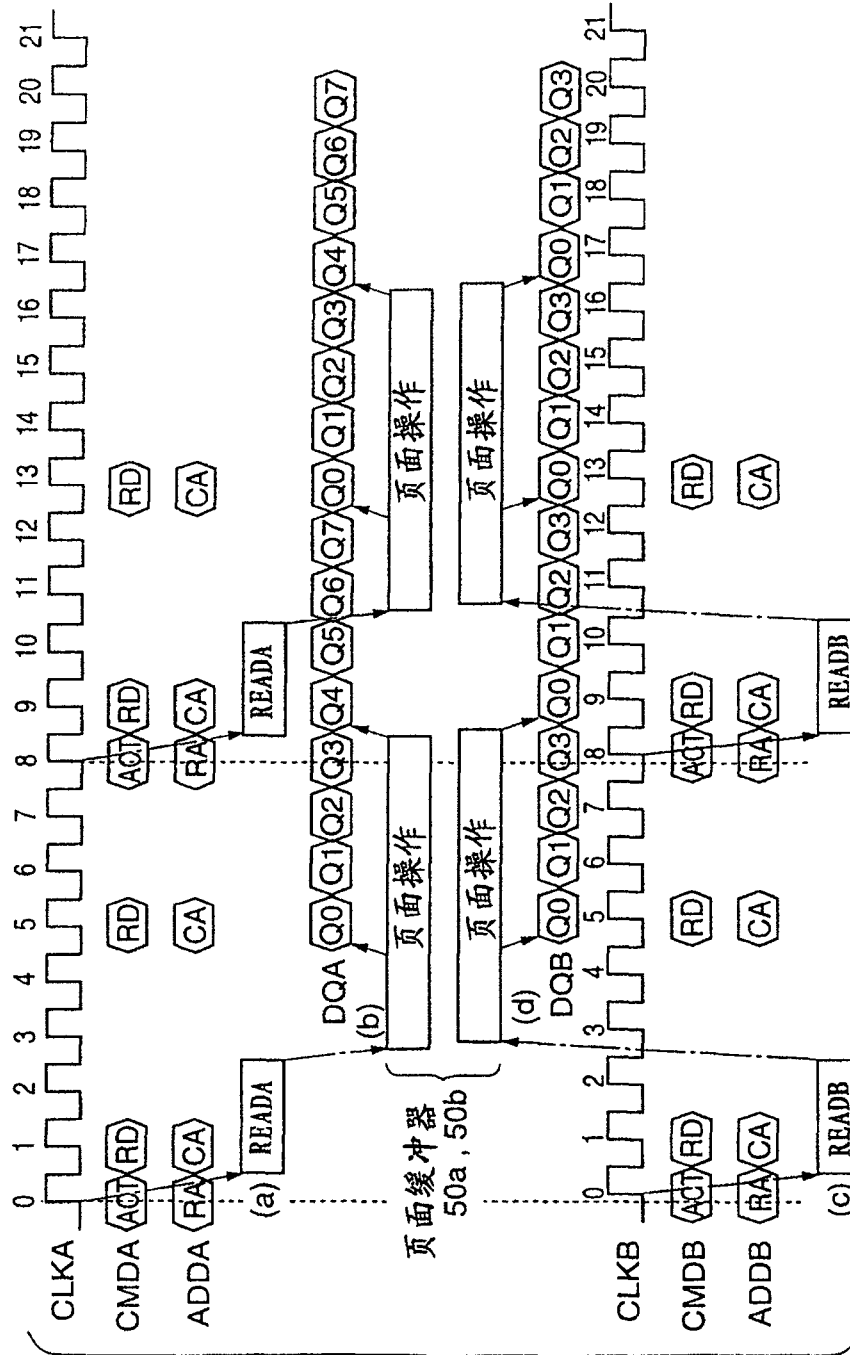


图135

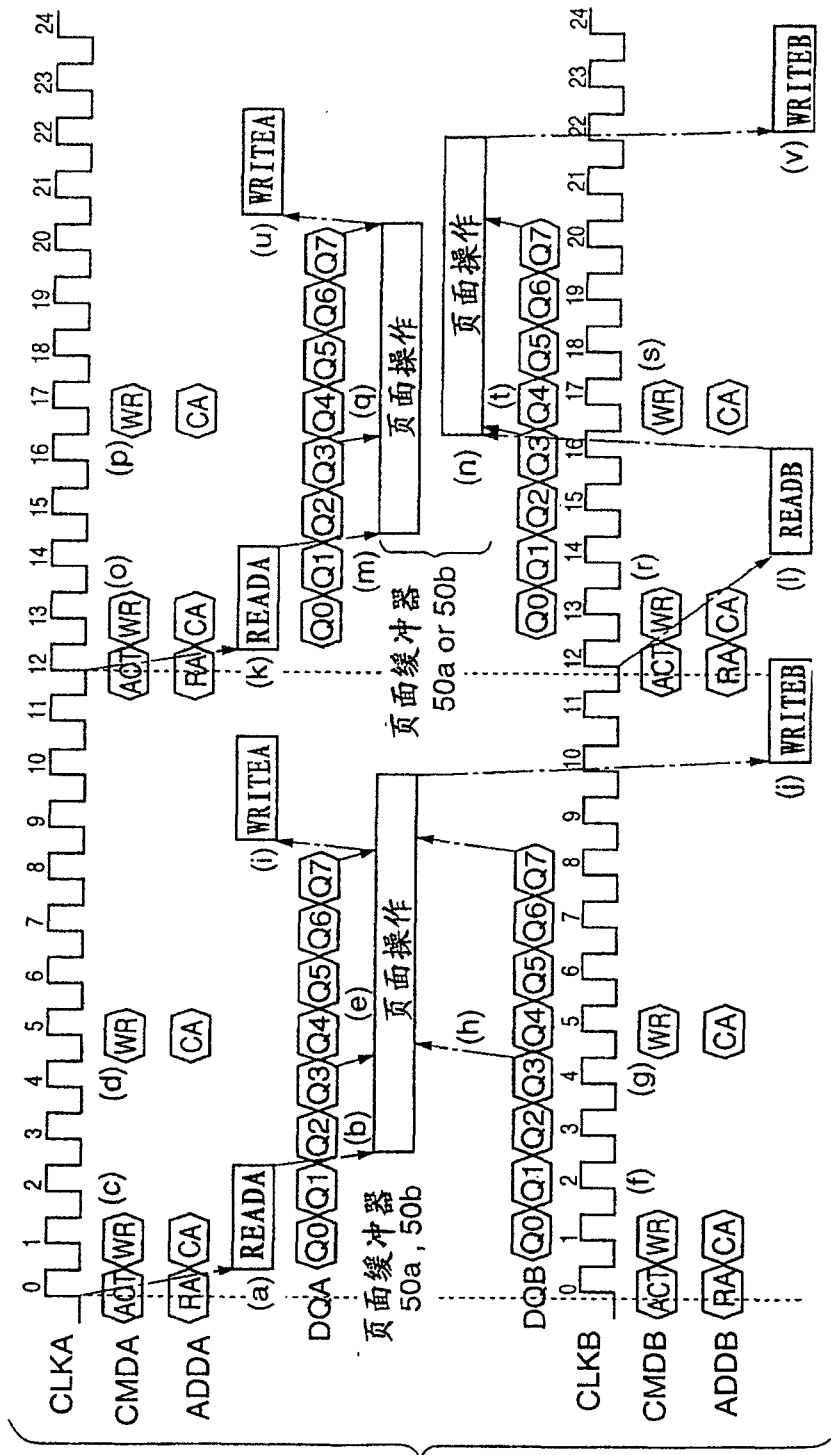


图 136

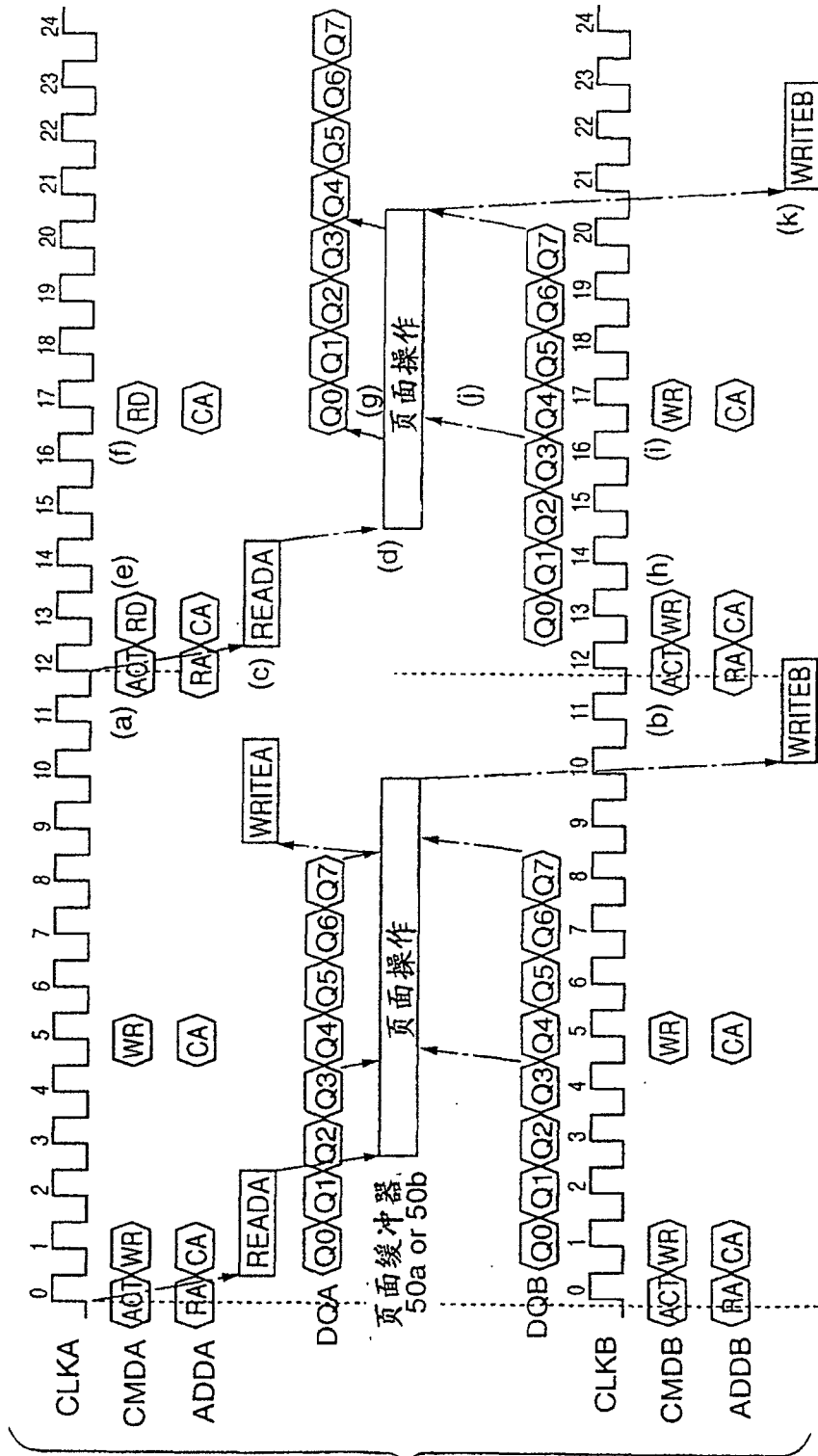


图 137

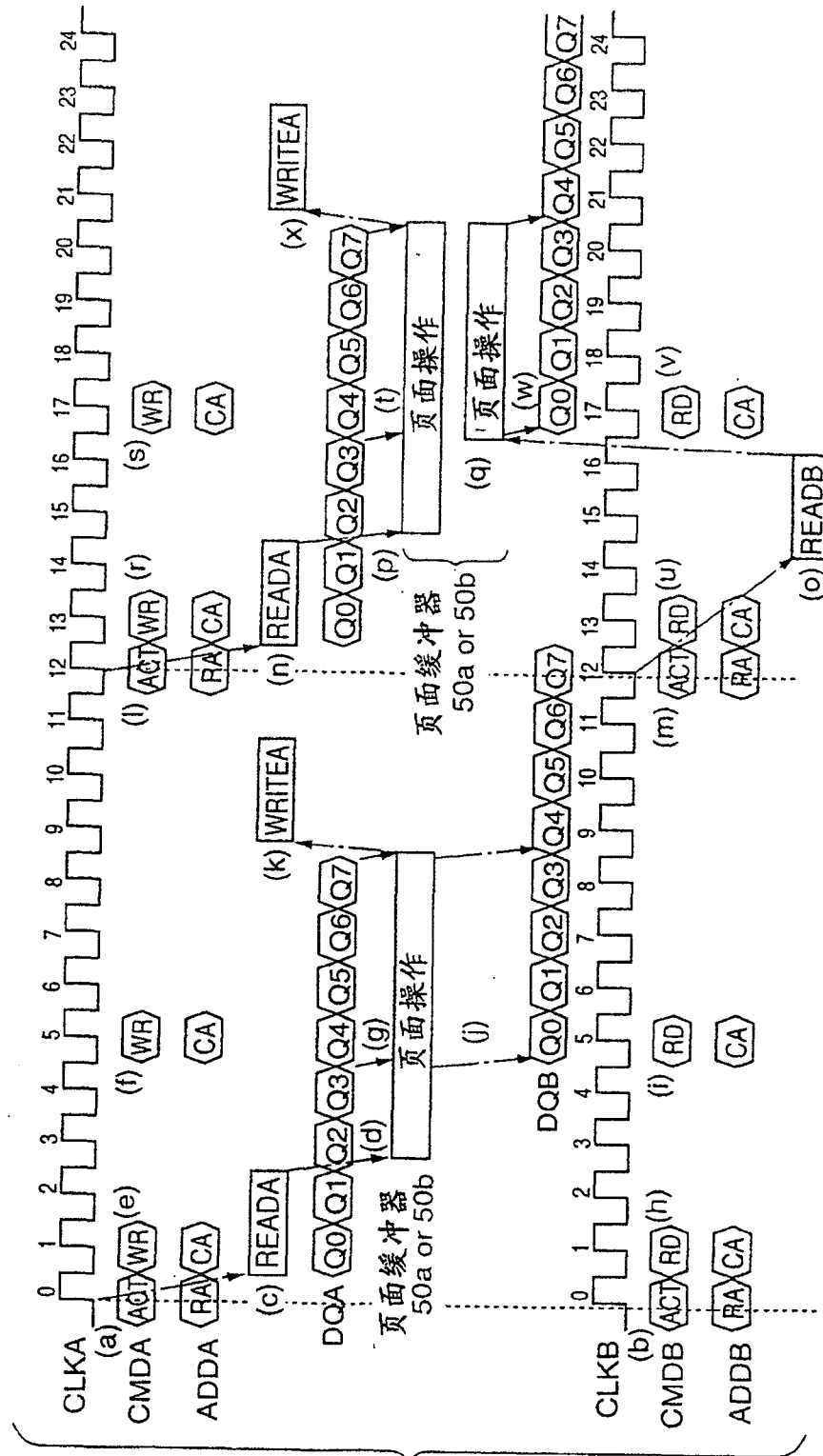


图138

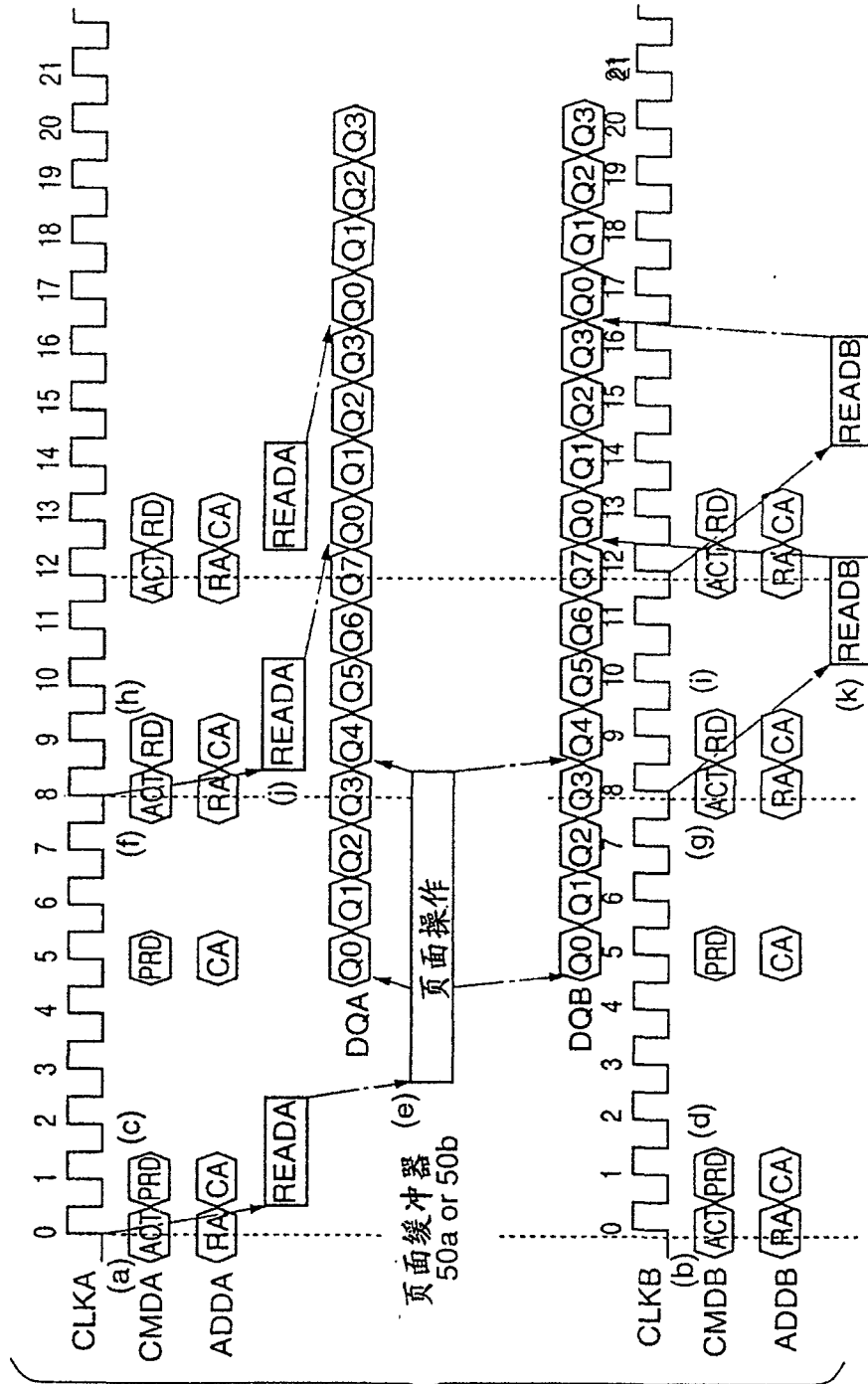


图 139