[19] 中华人民共和国国家知识产权局

[51] Int. Cl[7]
G11C 29/00

[12] 发明专利说明书

[21] ZL 专利号 98119561.X

[45] 授权公告日 2004 年 3 月 31 日

[11] 授权公告号 CN 1144232C

[22] 申请日 1998.9.24 [21] 申请号 98119561.X

[30] 优先权

[32] 1997. 9.29 [33] US [31] 939546

[71] 专利权人 西门子公司

地址 联邦德国慕尼黑

[72] 发明人 冈纳·H·克劳斯 奥利弗·基尔

审查员 穆丽娟

[74] 专利代理机构 北京市柳沈律师事务所

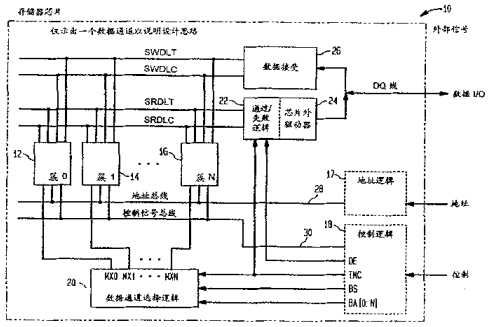
代理人 陶凤波

权利要求书 2 页 说明书 9 页 附图 2 页

[54] 发明名称 多簇存储器件的测试方法及其访问方法

[57] 摘要

一种测试多簇存储器件的方法包括：启动多个存储器簇中的两个或更多个，以参与测试；选择至少一个与每个被启动簇之中的存储单元相应的公用存储地址；同时向选定的每个被启动簇的存储单元中写入测试数据；同时读出以前向所选定的每个被启动簇的存储单元中写入的测试数据；和比较从每个被启动簇中读出的测试数据和从其它被启动簇读出的测试数据，如果确定为匹配，那么，表明通过状态，否则，表明失败状态。 本发明还涉及测试多簇存储器件的装置以及访问多簇存储器件的存储器簇的方法。



ISSN1008-4274

1. 一种测试带有多个存储器簇的多簇存储器件的方法，包括以下步骤：

启动所述多个存储器簇中的两个或更多个，以参与测试；

选择至少一个与每个被启动存储器簇之中的存储单元相应的公用存储地址；

同时向选定的每个被启动存储器簇的存储单元中写入测试数据；

同时读出以前向所选定的每个被启动存储器簇的存储单元中写入的测试数据；和

比较从每个被启动存储器簇中读出的测试数据和从其它被启动存储器簇读出的测试数据，如果确定为匹配，那么，表明通过状态，否则，表明失败状态。

2. 根据权利要求1所述的方法，其中，一测试图分别被写入每个被启动存储器簇中多个共同的选定存储单元中，每个存储器簇中的每个共同选定的存储单元被同时写入。

3. 根据权利要求2所述的方法，其中，以前写入多个共同选定的存储单元中的测试图的至少一部分被同时读取，并对其进行比较，以确定通过和失败状态。

4. 根据权利要求2所述的方法，其中，每个被启动存储器簇中的多个选定的存储单元构造于多个字线之中，其中，存储在非对应字线中的测试图的多个部分被同时从被启动存储器簇中读出，并对其进行比较，以确定一通过/失败状态。

5. 一种有选择地访问多簇存储器件的存储器簇的方法，其中，多簇存储器件包括一公用内部数据通道，它工作连接到多个存储器件之上，上述方法包括以下步骤：

提供一簇启动信号，簇启动信号显示多个存储器簇的哪些存储器簇被启动；

通过公用内部数据通道允许从被启动的存储器簇中顺序内部读取数据或向被启动的存储器簇中顺序内部写入数据；和

通过公用内部数据通道拒绝从未启动的存储器簇中顺序读取数据或向未启动的存储器簇中顺序内部写入数据。

6. 一种测试带有多个存储器簇的多簇存储器件的方法，该方法包括以下步骤：

启动多个存储器簇当中的两个或更多个，以参加测试；

选择被启动存储器簇中与存储单元相应的至少一个公用存储地址；

向被启动存储器簇的选定存储单元中并行写入测试数据；

读取以前并行写入被启动存储器簇的选定存储单元中的测试数据；和

比较从被启动存储器簇中读出的测试数据和从其它被启动存储器簇中读出的测试数据，如果确定为匹配，那么显示一通过状态，否则显示一失败状态。

7. 如权利要求6所述的方法，其中，一测试图分别被并行写入被启动存储器簇中多个共同选定的存储单元。

8. 根据权利要求7所述的方法，其中，以前写入多个共同选定的存储单元的测试图的至少一部分被并行读出，并对其进行比较，以确定通过和失败状态。

9. 根据权利要求7所述的方法，其中，被启动存储器簇中多个选定存储单元构造于多个字线内，其中，存储在非相应字线中的测试模式被从被启动存储器簇中并行读出，并对其进行比较，以确定一通过和失败状态。

多簇存储器件的测试方法及其访问方法

本发明涉及一种多簇存储器件，更具体地说，涉及一种改进的用于多簇存储器件的多簇联锁方案和相应的测试方式。

众所周知，多簇存储器件(multibank memory device)中的整个存储阵列被分成功能一致的存储器簇(memory banks)。每个存储器簇典型地有相同数目的字线和位线，这些簇可以彼此独立地启动。结果，用于启动一个簇的行地址不必和用于启动其它簇的其它行地址相同。

众所周知，通过行地址来启动字线会导致与启动字线相连的所有存储单元被闩锁在用于每个相应位线的读出放大器中。在传统的标准操作模式中，每个读和写命令提供簇选择信息和一列地址，以选择一簇的启动字线的一具体位线。也就是，首先，簇选择信息被用来选择一个簇，只有在这个簇中列地址用于选择从相应读出放大器中读出的数据(或写入数据)。然而，由于这里仅有一个公享数据通道，只有一个簇可以存储相关系统写入数据线上的数据或驱动系统读出数据线上的数据。

结果，传统的多簇存储器件的测试一次只能执行一个簇。虽然这种方法可以进行存储器簇的测试，但是逐个对簇顺序执行这种方法是过于费时的，因此这是不允许的。

根据本发明的一方面，一种测试带有多个存储器簇的多簇存储器件的方法，包括以下步骤：启动所述多个存储器簇中的两个或更多个，以参与测试；选择至少一个与每个被启动存储器簇之中的存储单元相应的公用存储地址；同时向选定的每个被启动存储器簇的存储单元中写入测试数据；同时读出以前向所选定的每个被启动存储器簇的存储单元中写入的测试数据；和比较从每个被启动存储器簇中读出的测试数据和从其它被启动存储器簇读出的测试数据，如果确定为匹配，那么，表明通过状态，否则，表明失败状态。

根据本发明的另一方面，一种有选择地访问多簇存储器件的存储器簇的方法，其中，多簇存储器件包括一公用内部数据通道，它工作连接到多个存储器件之上，上述方法包括以下步骤：提供一簇启动信号，簇启动信号显示

多个存储器簇的哪些簇被启动；通过公用内部数据通道允许从启动的簇中顺序内部读取数据或向启动的簇中顺序内部写入数据；和通过公用内部数据通道拒绝从未启动的簇中顺序读取数据或向未启动的簇中顺序内部写入数据。

最好是，本发明提供一多簇 DRAM 器件测试模式和一簇联锁方案，多簇DRAM 器件测试模式大大减少了测试所有相关的多簇DRAM 器件的簇总测试时间，簇联锁方案在测试过程和正常操作模式过程中根据簇提供一数据通道锁定特征。

参考附图，通过下面对本发明实施例的详细介绍，本发明的这些和其它目的、特征和优点就会更加明显。附图中：

图 1 是一方框图，示出了根据本发明实现簇联锁方案和测试模式的多簇 DRAM 器件；

图 2 是本发明数据通道簇选择逻辑布置用的实例性逻辑真值表；

图 3 是本发明通过/失败逻辑和芯片外驱动器块用实例性逻辑真值表。

首先，参考图 1，通过一多簇存储器件(multibank memory device)10 的相关部分示出了本发明的实施例。这种存储器件包括：随机存取存储器(RAM)；动态随机存取存储器(DRAM)；同步 DRAM(SDRAM)；只读存储器(ROM)或归并 DRAM －逻辑电路(嵌入或 DRAM)。在一个实施例中，存储器件包括一 DRAM 。应当理解，为了清楚起见，仅示出了一位数据通道；然而，很明显本发明的实施可延伸至包括具有较大数目平行数据位(如 8, 16, 32, 64 位数据总或更大)的更实用的数据总线结构。

如图所示，一多簇 DRAM 器件 10 包括 N 个簇。多簇存储器件优选为多簇 SDRAM 器件，尽管其它类型的多簇 DRAM 器件也可以实施本发明的技术。各个簇，簇 0(块 12)、簇 1(块 14)、至簇 N(块 16)，通过分别选择信号簇线 MX0, MX1 至 MXN，与数据通道簇选择逻辑 20 工作连接。另外，每个存储器簇工作连接到两条系统读出数据线，即 SRDLT(system read data line true，系统读出数据线真)和 SRDLC(system read data line complement，系统读出数据线补码)，它们也工作连接到测试模式通过/失败逻辑 22 上，这一点将在下面解释，并且测试模式通过/失败逻辑 22 本身工作连接到片外驱动器 24(OCD, off chip driver)上。同样，每个存储器簇工作连接到两条系统写数据线，即 SWDLT(system write data line true，系统写数据线真)和 SWDLC(system

write data line complement，系统写数据线补码)上，它们工作连接到数据接受器 26 上。外部数据输入/输出线(DQ 线)工作连接到 OCD 24 和数据接受器 26 上。另外，如图所示，每个存储器簇工作连接到一地址总线 28 和一控制信号总线 30。总线 28 的地址信号可以通过地址逻辑 17 用传统的方法产生，而控制信号(如读/写命令)可以通过控制逻辑 18 用传统的方法产生。地址信号和控制信号分别从外面由片外 DRAM 控制器电路提供到地址逻辑 17 和控制逻辑 18 中。

响应于外部控制信号(如 DRAM 控制器，未显示)，控制逻辑 18 产生簇启动(BA[0:N])信号、一簇选择(BS)信号、和一压缩测试模式(TMC, compression test mode)信号。这三个信号提供到数据通道簇选择逻辑 20。另外，通过/失败逻辑 22 接受 TMC 信号和一输出允许(OE, output enable)信号(也是由控制逻辑 18 产生的)。通过这里描述的本发明的教导，本领域的技术人员就能够实施这种逻辑控制电路(控制逻辑 18)，所以，在此不再赘述。

尽管 SRDLT 和 SWDLT 以分开的数据线进行了说明，但是它们可以是一条双向数据线。同样， SRDLC 和 SWDLC 可以是另一条双向数据线。另外，采用真/补码数据线的设计是可以用其它方式替代的。另外，也可以采用带有数据有效信号的一单条读数据线和一单条写数据线。

总的说来，本发明提供了一种独特的测试模式，它明显地减少了与多簇 DRAM 器件的簇测试有关的时间。根据本发明，如下面的详细说明所述，测试模式同时测试存储单元的所有被启动存储器簇，其通过对数据位的并行写/读来测试。结果，一条写命令等效于分别对应 N 个不同簇的 N 条顺序写命令；而一条读命令提供通过/失败信号，作为读所有 N 个簇的压缩的结果。例如，如果对每个存储器簇测试模式相同，那么，在每个存储器簇的启动过程中，行地址是相同的。此时，如果所有簇的读出数据和写入数据相同，则测试的结果就是“通过”。如果数据不一致，于是测试到“失败”，器件可被取下，或者，如果是在预熔测试中(下面将介绍)，存储器的失败部分可以被修复或用备用部分来替代。

虽然，本发明的测试模式在每个存储器簇中可以用相同的局部测试模式，结果，在启动过程中，每个存储器簇的行地址相同，测试模式还可以压缩非对应字线。很明显，当被压缩的数据来自芯片的远离区域时，可以使多位错误导致错误测试结果的可能性减为最小。这一点将在下面解释。

现在参考图 2，示出了本发明独特测试模式和锁定功能的实施例，其对应于输入到器件 10 的数据通道簇选择逻辑 20 的信号(BS, BA, TMC)和从此逻辑 20 输出的信号(MX). 请注意，为了方便在表和该说明书之间参考对照，在表中插入了列号(C)和行号(R). 列 1 代表本发明允许和不允许(禁止)测试模式的测试模式信号(TMC). 所以，通过赋予逻辑 0 为禁止条件和逻辑 1 为允许条件，可以看出行 1 - 6(TMC = 0)表示器件 10 的正常操作(非测试模式或测试模式不允许)，而行 7(TMC = 1)表示器件处于测试模式中。当然，必须理解的是所采用的逻辑标记也可以互换过来，即，逻辑 0 表示允许功能，而逻辑 1 表示禁止功能。这种标记对本发明并不十分重要。

另外，列 2 - 5 代表簇选择(BS)信号的各个位。虽然，在最高有效位(MSB)(列 2)和最低有效位(LSB)(列 5)之间示出的是两位，根据所选取的簇数， BS 信号的位数可多可少。另外，列 6 - 9 表示各个簇启动(BA)信号，其中，列 6 表示簇 0 的启动(BA0 信号)；列 7 表示簇 1 的启动(BA1 信号)；列 9 表示簇 N 的启动(BAN 信号)。列 8 仅表示在簇 1 和簇 N 之间所有可能被启动的簇。同样，列 10 - 13 表示各个选择簇(MX)信号，其方式与 BA 信号所用方式相似。

下面将介绍正常操作(非测试模式)的例子。行 1 示出了簇 0 的选择和启动。 BS 信号的各位均是逻辑 0，这表示通过外部控制电路(DRAM 控制器)而选择了簇 0 。接着， BA0 信号(列 6)被设定为逻辑 1，它表示簇 0 被启动。BA 信号的剩余部分是“无关紧要”(“ don't cares ”)(×)，因为，在此处无论其它簇是否被启动都没有关系，因为，根据 BS 信号的条件，只有簇 0 被选取。因此， MX0 信号(列 10)也被设定为 1 ，所以，从内部选择簇 0 ，如图 1 所示。所有其它 MX 信号都是逻辑 0，从而不会从内部选取这种未选择(也就是，没有通过 BS 信号选址)的簇。这样行 1 说明了簇 0 的读或写操作。是读操作还是写操作是通过外部 DRAM 控制器用传统的方法来确定的。

接着，行 2 显示通过 BS 信号试图选择簇 0 。然而，因为 BA0 信号没有被启动，结果 BA0 信号是逻辑 0， MX0 信号没有被设定，所以，簇 0 没有被从内部选择。所以，如果一簇没有被由适当设定的 BA 信号启动，那么不管 BS 信号的条件如何，簇永远不会被选取。

行 3 和行 4 分别示出了与行 1 和行 2 所示相同条件，但是它们是关于簇 1 的。也就是，行 3 是簇 1 读或写数据操作，行 4 示出了簇 1 不能执行读/写

操作。同样，行5和6显示的是簇N的相同内容。

所以，可以理解，通过控制BA信号，从而提供本发明独特的联锁特性。正如参考行2,4和6显示和所介绍的，即使为了选择各个簇的簇选择信号(BS)被设定为逻辑 1 ，实际上簇也不会通过适当选择簇(MX)信号而从内部被选取，除非设置了适当的簇启动(BA)信号。所以，本发明能锁定和释放多簇DRAM 器件中的一个簇和/或多个簇。也就是，如果某些簇没有被启动，那么，它们就不能访问器件的内部数据线，所以它们不执行读/写操作。这种独特的联锁方案可以用于正常操作(这一点上面已进行了介绍)，也可以用于测试模式(下面将解释)。

参考图2中表的行7，它示出了测试模式(TMC)信号被设定为逻辑1，所以，能启动本发明的测试模式。结果，提供到器件中的簇选择(BS)信号的条件没有关系(“don't care”)。然而，簇启动(BA)信号确定哪个簇处于测试之中。列6－9表示簇0～N被启动；然而，测试模式中被启动的簇较少(可少至一个)。MX信号根据BA信号来设定。如表中所示，MX0～MXN被设定，从而从内部选择簇0～N。列6－12中所用的名称(BA0,BA1，…，BAN)用于突出BA信号和MX信号之间的对应关系；然而，根据这里所用的逻辑标记，每个信号以逻辑1的形式出现，表示信号允许(也称使能)。当然，如上所述，逻辑0可以用来表示允许状态，而逻辑1表示禁止状态。

因此，随着测试模式被允许，启动(用BA信号)和选择(用MX信号)所需簇，一单条写(或写信号组)命令可以被发出(还是通过外部DRAM控制器)，然后，一测试模式被(用地址总线28提供列地址)并行写入被启动的每个存储器簇的存储单元(如，每个存储器簇一个存储单元，除非一写信号组命令被发出，此时，每个存储器簇多于一个存储单元被写入)。如上所述，单条写命令相当于N个不同簇的N条顺序写命令，N是被启动存储器簇的数目。另外，BA信号可以起着为此时没有选来进行测试的簇锁定访问数据线。当测试模式数据被写入启动存储块以后，提供一读命令，获取通过或失败信息，作为并行读取的压缩结果，并比较所有N个簇的数据。

应当理解，测试模式中的确切的测试图(test pattern)对本发明并不重要，熟悉该技术的人都可以根据所用测试图来实施确定通过/失败状态的可接受的方法。然而，现在参考图3，示出了一真值表，它显示了连同芯片外驱动器24(图1)一起实施通过/失败逻辑22(图1)有关的可能结果的实例。因为通

过/失败逻辑的确切实施取决于所采用的数据线的类型，其实施对本发明并不重要，结果，表 3 仅表示由本发明簇联锁方案和测试模式提供的独特并行写/读操作中的处理结果的实例。

列和行数也是提供来为了方便表格和说明书之间的对照。列 1 - 4 表示在测试模式为禁止时(TMC = 0)，系统读数据线真/补码(SRDLT 和 SRDLC)、输出允许(OE)信号和器件外部输入/输出数据(DQ)线的逻辑条件的实例。列 5 - 8 表示当测试模式启动时(TMC = 1)，上述各项逻辑条件的实例。

所以，首先参考列 1 - 4，从中可以看出，在正常操作(非测试模式)时，SRDLT 和 SRDLC 将数据送至芯片外驱动器 24。所以，如列 1 所示，当 SRDLT 是逻辑 0 时，互补线 SRDLC 是逻辑 1。然后，正如所知道的那样，如果 OE 信号被设定，那么，在 DQ 线上的输出数据位为逻辑 0。另外，当 SRDLT 为逻辑 1 时，SRDLC 为逻辑 0，假设 OE 信号被设定，那么，DQ 线是逻辑 1。

在列 3 中，SRDLT 和 SRDLC 都被预先设定为逻辑 1，结果，DQ 线位于高阻(HiZ)状态，没有数据从芯片外驱动器向外输出。列 4 示出了一无效状态，更确切地说，如果器件将正常模式正确操作时，不可能出现的状态。也就是，SRDLT 和 SRDLC 都为逻辑 0。这种状态要求 SRDLT 和 SRDLC 同时放电，这是不可能发生的，因为在正常操作模式中一次仅选择一个簇而其它簇没有启动。所以，列 4 中的 DQ 表示为无效/不可能。当然，如果采用相反的逻辑极性，那么，SRDLT 和 SRDLC 的状态为逻辑 1 时为无效/不可能，而 SRDLT 和 SRDLC 的状态为逻辑 0 时表示高阻态(HiZ)输出。

如上所述，为了显示本发明，采用真/补码数据线，这是随意性的选择。真/补码数据线的优点在于当两个线都被预先设定为逻辑 1 时，一个或两个预先设定的线被一存储单元置为逻辑 0，那么，你就知道 DQ 线上输出的是无效数据。用单个数据信号时，必须提供数据有效信号以表示从器件中正式输出了有效数据。然而，根据本发明，两种方法都可以采用。

现在参考列 5 - 8，示出了本发明的测试模式。必须理解的是预先确定的测试图被首先并行写入 SWDLT 和 SWDLC 上的启动数据簇。例如，簇启动信号(BA[0:N])表示哪个簇将被测试。TMC 变成逻辑 1，从而进入测试模式，此时，BS 信号被忽略(don't care，如图 2 所示)。OE 信号将为逻辑 0，以表示一写操作。接着，一列地址被地址逻辑 17 解码，提供到地址总线 28

上(行地址也被预先从此处提供)，并且，一写命令被控制逻辑 18 解码，提供到控制信号总线 30 上。如果一逻辑 1 被首先写入到每个存储器簇相同的列地址中，那么，SWDLT 将为逻辑 1(SWDLC 将为逻辑 0)，所以，对应于上述列地址，逻辑 1 被并行写入每个存储器簇的每个存储单元中。然后，列地址被改变或被加 1，相应于下一个列地址，SWDLT 上的下一个数据值(逻辑 0 和另一逻辑 1)被并行写入每个存储器簇中的每个存储单元中。所以，整个测试图用这种方法写入各个存储器簇中。

现在，当从启动的簇中读取数据时，一读命令被控制逻辑 18 解码，提供到控制信号总线 30 上，OE 信号变为逻辑 1。接着，对应于码的列地址的每个存储器簇的每个存储单元中的数据位被并行读出，提供到通过/失败逻辑 22(图 1)中。如果，所有的存储单元具有相同的数据(逻辑 1 或逻辑 0)，也就是，匹配，那么，将获得一测试通过状态；然而，若至少一个簇输出的值不同，那么，将会显示出测试失败的状态。

图 3 中的表示出了这个概念。列 5 表示 SRDLT 为逻辑 0，而 SRDLC 仍保持为逻辑 1。这表明与各列地址相应的所有存储单元正在存储一逻辑 0。假设这就是在写操作过程中写入这种单元中的值，那么，测试的那部分就存在通过状态。这是用 DQ 线上输出的逻辑 0 来表示的。另一方面，列 6 表示 SRDLC 为逻辑 0，而 SRDLT 仍保持为逻辑 1。这就表明与下一个各列地址相应的所有存储单元正在存储逻辑 1。还假设这就是在写操作过程中写入这种单元的值，那么，测试的这部分也为通过状态。与列 5 中的状态相同，结果以一逻辑 0 示于 DQ 线上。应当理解，DQ 线上的逻辑电平输出(如表 1 所示)是任意的，可以用多种方法来实施。也就是，可以这样形成通过/失败逻辑 22，使逻辑 1 在 DQ 线上输出来表示通过状态，而不是用表中所示的逻辑 0。另外，读数据线上的实际数值可以在 DQ 线上输出，作为通过或失败状态的标记，例如，列 5 中的 DQ 为逻辑 0，而列 6 中的 DQ 为逻辑 1。

列 7 显示 SRDLT 和 SRDLC 都不在驱动数据的状态，也就是，既得不到数据，又没有簇被启动，根据本发明的锁定特性，没有簇可以进入数据通道。所以，SRDLT 和 SRDLC 保持为逻辑 1，此时，DQ 线处于高阻抗(HiZ)状态。现在，参考列 8，SRDLT 和 SRDLC 都为逻辑 0。如果例如在列地址的帮助下写入簇单元的数值为逻辑 0，那么将会存在这种状态。通常，如果出现这种情况，SRDLT 将仍然为逻辑 0，然而，SRDLC 将为逻辑 1。使 SRDLC

也为逻辑0的唯一方式是其中具有写入的逻辑0的主存储单元的至少一个表示一逻辑1，使SRDLC为逻辑0。这表明这种存储单元出现了问题，因为开始写入的是逻辑0，而从中读出的为逻辑1。所以，根据表中列5所示通过/失败状态的逻辑电平输出，DQ线上输出逻辑1，来表示一失败状态。另外，也可以用其它逻辑或数据线状态来表示测试失败状态，如迫使DQ线为HiZ状态。

最后，表中列9显示出了发出写命令且测试模式没有启动时所发生的情况。OE信号变为逻辑0，结果，不再读取测试结果，所以，SRDLT和SRDLC无关紧要(don't care)(×)。

应当理解，可以运行本发明的测试模式来提供一简单通过/失败结果，以显示整个阵列中的一个簇处于具有缺陷状态。这对测试图(test pattern)被写入整个存储阵列(或其中一部分)的总体测试存储器件来说很有用，存储器件在一定时间内处于加电状态，然后存储的测试图被读出，并与以前所写入的数据图(data pattern)进行比较。另外，代替使存储器件处于空载的状态，器件可以受到芯片中引入的传导性和/或辐射噪音的干扰，然后读出数据并进行比较，看一看是否由噪音对数据产生负影响。在预熔测试(prefuse testing)时，一产生简单通过/失败指示的单个读循环不足以查出失败源，此时，需要有多个N次读循环(N + 1)(直至固定的失败循环数)来辨别哪个或哪些簇失败。所以，一旦失败的簇被查出，受影响的字线或位线可以分别被一备用字线或位线取代。

另外，如上所述，本发明的测试模式允许压缩非对应字线。也就是，每个存储器簇对应的字线不必被读取(或写入)。所以，例如第一字线可以从第一被启动存储器簇中读出，第十五字线可以从第二被启动存储器簇中读出，第三十字线可以从第三被启动存储器簇中读出，等等。具有了这种能力，如果被压缩的数据出自器件的非响应字线(它位于芯片的远端区域)，那么，多位错误造成错误测试的可能性会减为最小。换句话说，根据测试图，多位错误被错构成通过状态的机会可以被大大地减少。

本发明的压缩测试模式可以和已有的压缩测试模式进行比较，在已有的模式中，一个簇的一个字线中的数个数据位在测试模式中可以被压缩，以帮助减少一个字线上所需顺序列访问命令的数目。在这种状态下，所用测试图受到限制，因为逻辑1和逻辑0的位置和距离必须适合压缩方案(compression

scheme)。

另外，在这里所介绍的测试过程中，任何写或读命令的簇选择(BS)信号被掩蔽，且每个启动的簇和其数据通道可以被内部选择。所以，处于测试模式的簇数目在用户使用簇启动和预加电命令的任何时候都能够变化。如上所述，非被启动存储器簇既不能存储数据，也不能提供数据，所以，它们的数据通道不被选取。

可以理解，传统的存储器件电路(如地址解码器等)可以是地址逻辑 17 和控制逻辑 18 或为简化起见未显示出部分的一部分。另外，尽管没有明确示出，所有未示出的传统控制信号(如数据通道时钟)都可以通过控制信号总线来提供。本领域的技术人员将会知道这种传统的机构和信号可以通过已知的方式来提供。

尽管这里已参考附图介绍了本发明的实施例，可以理解，本发明不局限于这些实施例，本领域的技术人员在不背离本发明的发明范畴的前提下可以对本发明进行各种变化和修改。

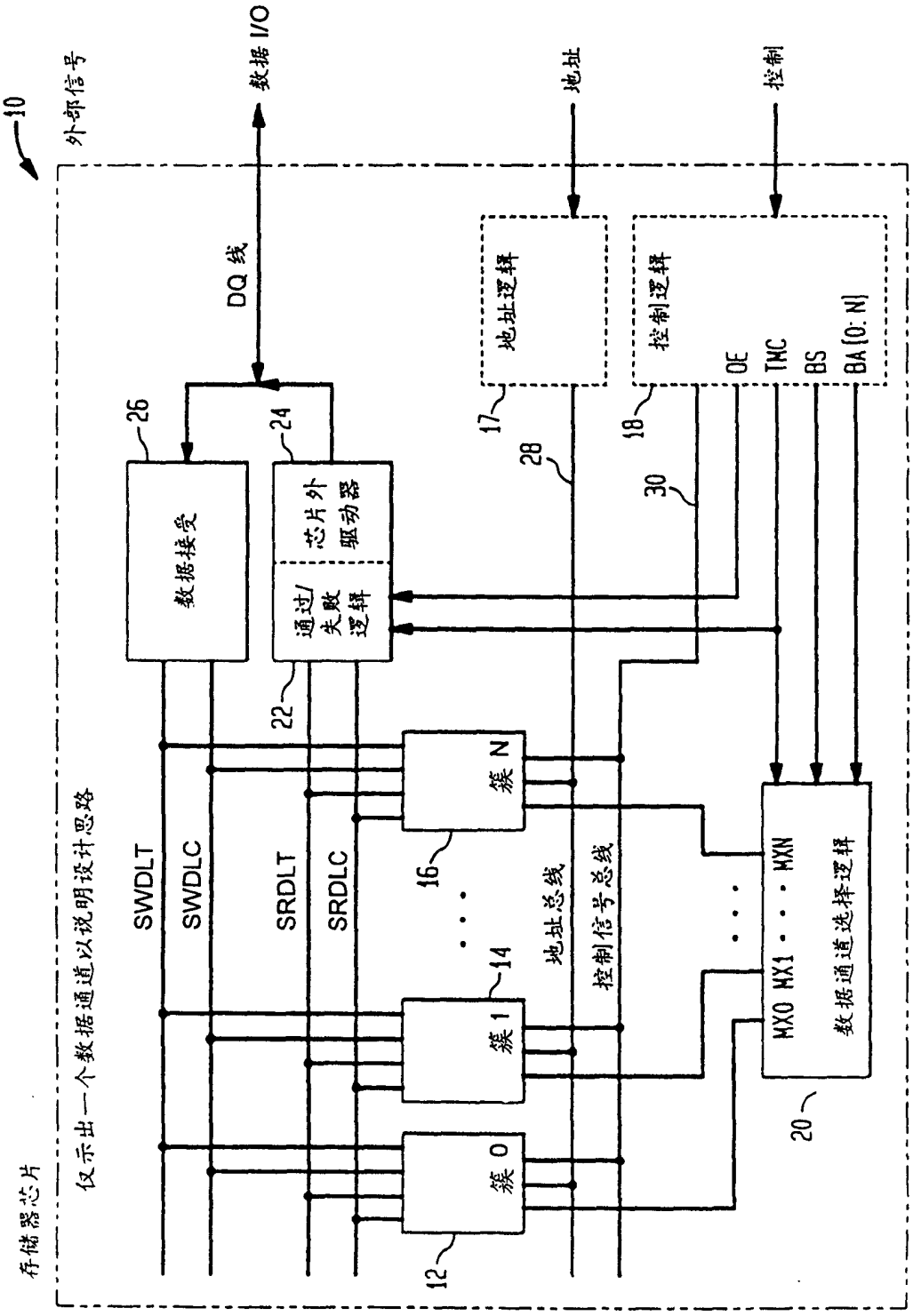

10
外部信号
数据 I/O
地址
控制
DQ 线
地址逻辑
控制逻辑
OE
TWC
BS
BA[0:N]
17
18
26
24
28
30
数据接受
芯片外驱动器
通过/失败逻辑
22
SWDLT
SWDLC
SRDLT
SRDLC
簇 N
簇 1
簇 0
16
14
12
地址总线
控制信号总线
MX0 MX1 · · · MXN
数据通道选择逻辑
20
仅示出一个数据通道以说明设计思路
存储器芯片

图 1

R \ C	1	2	3	4	5	6	7	8	9	10	11	12	13
	TMC	BS				BA				MX			
		MSB	···		LSB	0	1	···	N	0	1	···	N
1	0	0	0	0	0	1	X	X	X	1	0	0	0
2	0	0	0	0	0	0	X	X	X	0	0	0	0
3	0	0	0	0	1	X	1	X	X	0	1	0	0
4	0	0	0	0	1	X	0	X	X	0	0	0	0
5	0	1	1	1	1	X	X	X	1	0	0	0	1
6	0	1	1	1	1	X	X	X	0	0	0	0	0
7	1	X	X	X	X	BA0	BA1	···	BAN	BA0	BA1	···	BAN

图 2

R \ C	1	2	3	4	5	6	7	8	9
1	SRDLT	0	1	1	0	0	1	1	0	X
2	SRDLC	1	0	1	0	1	0	1	0	X
3	TMC	0	0	0	0	1	1	1	1	X
4	OE	1	1	1	1	1	1	1	1	0
5	DQ	0	1	高阻	无效或不可能	0 通过	0 通过	高阻	1 失败	高阻

图 3