



1. 一种在数据处理系统中用于存取收回的方法，其中所述的数据处理系统具有总线主控和系统总线，所述总线主控耦合到所述系统总线，所述方法包括：

    储存存取收回控制信息，该信息指示出允许或不允许收回的至少一个条件；

    请求收回系统总线上当前待定的存取请求；以及

    响应于上述的请求步骤，使用存取收回控制信息，基于当前待定存取请求的至少一个特性而选择性地收回当前待定存取请求。

2. 权利要求 1 所述的方法，其中，所述的请求收回是基于当前解码的指令而进行的。

3. 权利要求 1 所述的方法，其中，所述至少一个特性包括当前待定存取请求的存取类型。

4. 权利要求 1 所述的方法，其中，所述数据处理系统包括耦合到系统总线的第二总线主控，并且其中，所述至少一个特性包括哪个总线主控请求了当前待定存取请求。

5. 权利要求 1 所述的方法，其中，所述存取请求具有相应的存取地址，并且其中，所述至少一个特性包括存取地址所落入的地址范围。

6. 权利要求 1 所述的方法，其中，如果存取收回控制信息指示出基于所述至少一个特性而允许收回当前待定存取请求的话，所述的选择性地收回进一步包括：

    通过响应于当前待定存取请求而提供虚拟值来收回当前待定存取请求。

7. 权利要求 1 所述的方法，其中，如果存取收回控制信息指示出基于所述至少一个特性而允许收回当前待定存取请求的话，所述的选择性地收回进一步包括：

通过用新的存取请求来替换在系统总线上当前待定存取请求，收回当前待定存取请求。

8. 一种在数据处理系统中用于存取收回的方法，其中所述的数据处理系统具有总线主控和系统总线，所述总线主控耦合到所述系统总线，所述方法包括：

储存对应于地址范围的存取收回控制信息，其中，所述的存取收回控制信息指示出对于相应的地址范围允许还是不允许收回；

请求收回系统总线上当前待定的存取请求，其中，所述的当前待定存取请求具有相应地址；

确定所述相应地址是否落入到所述地址范围内；以及

如果所述相应地址落入到所述地址范围内，基于存取收回控制信息，选择性地收回当前待定存取请求。

9. 一种数据处理系统，包括：

系统总线；和

总线主控，耦合到所述系统总线，其中，所述总线主控请求收回系统总线上当前待定的存取请求，所述总线主控包括收回控制逻辑和存取收回控制存贮电路，所述存取收回控制存贮电路储存存取收回控制信息，所述收回控制逻辑使用所述存取收回控制信息来基于当前待定存取请求的至少一个特性而选择性地收回当前待定存取请求。

10. 一种在数据处理系统中用于存取收回的方法，其中所述的数据处理系统具有总线主控和系统总线，所述总线主控耦合到所述系统总线，所述方法包括：

储存存取收回控制信息，该信息指示出允许或不允许收回的至少一个条件；

接收后续存取请求；

请求收回所述系统总线上当前待定的存取请求，其中，在从主控接收到当前待定存取请求之后，从主控接收所述后续存取请求；以及  
响应于上述的请求步骤，使用存取收回控制信息，基于后续存取请求的至少一个特性而选择性地收回当前待定存取请求。

## 具备总线存取收回的数据处理系统

### 技术领域

本发明涉及数据处理系统，更具体涉及具备存取收回的数据处理系统。

### 背景技术

不同数据处理系统可能在各种不同总线协议下操作，其中，这些总线协议可能定义，例如，怎样处理总线主控进行的存取请求。而且，不同的数据处理系统可能使用不同的总线仲裁机制来操作，其中，不同的总线仲裁机制也定义了怎样处理来自不同总线主控的请求。这些总线协议和总线仲裁可能在许多方面不够灵活，诸如关于存取收回（access retraction），其通常引入了不希望的延迟。这些延迟可能尤其会对数据处理系统内的高优先级的总线主控造成问题。而且，在试图将根据不同总线协议和总线仲裁机制（例如不同地处理存取收回）操作的系统集成时可能出现错误。

因此，需要一种数据处理系统，其具有改进的存取收回机制，从而可能允许例如具有降低了的延迟的更有效的总线协议或仲裁机制。

### 附图说明

通过举例说明本发明，而不是通过附图来限制本发明，在附图中，相似的附图标记指示类似的元素，其中：

图 1 以框图形式图示说明了根据本发明一个实施例的数据处理系统；

图 2 以框图形式图示说明了根据本发明一个实施例的图 1 的总线主控的一部分；

图 3-5 以框图形式图示说明了图 2 的存取收回控制寄存器的各种

不同实施例；

图 6 以流程图形式图示说明了根据本发明一个实施例的用于进行选择性存取收回的方法；

图 7 以框图形式图示说明了根据本发明一个实施例的系统互连；

图 8 以框图形式图示说明了图 7 的主控优先级寄存器的一个实施例；

图 9 以框图形式图示说明了图 7 的止动控制寄存器的一个实施例；

图 10-11 以框图形式图示说明了图 7 的存取收回控制寄存器的各种不同实施例；以及

图 12 以流程图形式图示说明了根据本发明可选替换实施例的进行选择性存取收回的例子。

技术人员将认识到，图中的元素是为了简便和清楚的目的而示出的，不必按比例绘制。例如，图中某些元素的尺寸可能相对于其他元素较为夸大，这是为了有助于改善对本发明的实施例的理解。

#### 具体实施方式

本发明的一个实施例提供基于例如当前待定的总线存取请求的特性而选择性超越（override）总线协议。一个实施例提供基于例如当前待定的存取请求的特性而选择性收回当前待定的存取请求。一种可选替换实施例提供基于例如当前待定存取请求之后提供的新请求的存取的特性而选择性收回当前待定的存取请求。（另外，选择性收回可以基于当前待定存取请求和后续的存取请求这二者。）在一个实施例中，用于选择性允许超越总线协议或收回当前待定存取请求的当前待定的存取请求可以包括对应于所请求的总线存取的地址范围、所请求的总线存取的存取类型（例如，读/写、指令/数据）、对应于所请求的总线存取的存取顺序、所请求的总线存取的总线主控、或上述的组合。

图 1-6 的图示和描述将提供数据处理系统的一个例子，其允许选择性收回由请求总线主控做出的当前待定的总线存取请求。存取可能

当前在系统总线 20 上待定，等待系统总线 20 上当前存取的完成。在某些实施例中，当前待定的存取请求实际上并不活动，直至系统总线 20 上进行中的当前存取完成。系统总线 20 可实现总线协议，其中，总线操作的地址与数据阶段交叠，使得地址总线驱动用于后续存取请求（“当前待定的”存取请求），同时等待进行中的当前存取的数据返回。一旦进行中的当前存取完成，这可能会占用多个时钟周期，则其数据阶段结束，且在该数据阶段的终止时驱动的地址（称为当前待定的存取请求）变为新的当前存取，并且地址总线然后可能通过新存取请求驱动。地址和数据阶段因此交叠，当前待定存取请求在数据阶段终止边界处变为新的“当前”存取请求。在某些情况下，如果总线主控知道当前待定的总线存取请求将不会被使用或不再需要的话，收回在系统总线 20 上的当前待定总线存取请求可能是总线主控所希望的。这样，就可能减少由于变得不必要或者应被丢弃的总线存取请求造成的延迟。因此，在一个实施例中，总线主控可以基于当前待定存取请求的特性而选择性收回当前待定存取请求，如参考图 1-6 所要描述的那样。

例如，图 1 图示说明了一种数据处理系统，其中，总线主控将指令连续预取到其流水线中。但是，这个连续存取模式可能由于流改变的指令，例如分支指令，而中断。当检测到流改变时，总线主控将指令预取流切换到流改变的目标（例如分支指令的目标）。因此，在此情况下，系统总线上总线主控的当前待定的存取请求可能对应于流指令改变之后的连续地址，其由于流到目标指令的改变，而不再被总线主控所需要。因此，总线主控可能需要收回这个现在不需要的当前待定存取请求。基于这个当前待定存取请求的特性（例如，可以与储存在存取收回控制寄存器中的控制信息进行比较），可以确定是否允许收回当前待定存取请求。如果允许，总线主控收回当前待定存取请求，由此减小延迟。

图 1 图示说明了数据处理系统 10 的一个实施例。数据处理系统

10 包括系统总线 20、总线主控 (master) 12、存储系统 14、总线受控 (slave) 16、总线受控 18、总线桥 22、总线主控 24、存储系统 26、其他受控 28、以及系统总线 30。总线主控 12、总线受控 16、总线受控 18、存储系统 14、以及总线桥 22 都双向耦合到系统总线 20。总线桥 22、总线主控 24、存储系统 26、其他受控 28 都双向耦合到系统总线 30。尽管在图 1 中只示出了两个受控 16 和 18，数据处理系统 10 可包括耦合到系统总线 20 的任何数量的受控。类似地，任何数量的总线主控和存储系统也可耦合到系统总线 20。而且，任何数量的总线主控、受控、存储系统可耦合到系统总线 30，并不限于图 1 中所显示的那样。

在一个实施例中，总线主控 12 可以是处理器，诸如微处理器、数字信号处理器等等，或者可以是任何其他类型的主控设备，诸如直接存储器存取 (DMA) 单元。受控 16 和 18 可以是任何类型的受控设备，诸如可由总线主控 12 存取的存储器，或者是外围设备，诸如通用异步接收器发射器 (UART)、实时时钟 (RTC)、键盘控制器等等。存储系统 14 可以是任何类型的存储系统，例如随机存取存储器 (RAM)，并且可以包括耦合到用于储存数据和指令的存贮电路的存储控制器。

操作中，总线主控 12 将存取请求置入系统总线 20，其请求对耦合到系统总线 20 上的其他设备的存取，例如对存储系统 14、受控 16 或 18 的存取，或者经由总线桥 22 对总线主控 24、存储系统 26、或其他受控 28 的存取。总线桥 22 允许系统总线 20 与系统总线 30 进行对接，从而使得总线主控 12 也可以存取驻留在系统总线 30 上的设备。在一个实施例中，系统总线 20 根据与系统总线 30 不同的总线协议进行操作。总线桥 22 如本领域所知进行操作，因此这里不更详细地描述。类似地，上面提供的总线主控 12、存储系统 14 和受控 16 和 18 的相同例子同样分别适用于总线主控 24、存储系统 26 和其他受控 28。

图 2 图示说明了根据本发明一个实施例的总线主控 12 的一部分。总线主控 12 包括指令定序器 40、指令缓冲存贮 66、总线接口单元 (BIU)

70 以及执行单元 80。BIU 70 提供送往和来自系统总线 20 的信息，并且双向耦合到指令缓冲存贮 66 和指令定序器 40。指令缓冲存贮 66 双向耦合到指令定序器 40。指令定序器 40 包括指令寄存器 (IR) 46、指令解码器 44、定序器控制逻辑 42、存取收回控制寄存器 48、收回控制逻辑 50 以及地址生成逻辑 52。IR 46 耦合到地址生成逻辑 52 以及指令解码器 44，指令解码器 44 耦合到定序器控制逻辑 42。定序器控制逻辑 42 耦合到收回控制逻辑 50 和地址生成逻辑 52，收回控制逻辑 50 耦合到地址生成逻辑 52 以及地址收回控制寄存器 48。执行单元 80 耦合到指令解码器 44 和 BIU 70，并且包括加载/存贮单元 82。

操作中，指令定序器 40 控制指令和数据从例如存储系统 14（或其他受控）经由 BIU 70 的预取和取操作。例如，指令可以预取到指令缓冲存贮 66，而数据可以经由 BIU 70，取或预取以及提供到执行单元 80（例如，加载/存贮单元 82）。IR 46 从指令缓冲存贮 66 接收要解码的下一指令，并将其提供给指令解码器 44，对指令进行解码。随后将指令提供给执行单元 80 进行执行，并且对指令作进一步处理。指令在解码后的处理是本领域公知的，因此这里不再更加详细地进行描述。也就是，执行单元 80，包括加载/存贮单元 82，如本领域公知进行操作。

定序器控制逻辑 42 控制指令到指令缓冲存贮 66 的预取和取操作。例如，定序器控制逻辑 42 指示地址生成逻辑 52，生成哪个地址（和对应属性）经由地址/属性 74 到 BIU 70。例如，在一个实施例中，定序器控制逻辑 42 指示地址生成逻辑 52 连续预取开始于特点位置的指令。因此，BIU 70 经由地址/属性 74 从地址生成逻辑 52 接收要被预取的地址。地址生成逻辑 52 声明 (assert) 请求 72，以向 BIU 70 指示，地址和对应的属性现在在地址/属性 74 上，而 BIU 70 可经由请求 72 向指令定序器 40 返回握手信号。在一个实施例中，BIU 70 包括地址队列（未示出），其中放置经由地址/属性 74 接收到的地址和属性，直至系统总线 20 可用于接收对应于地址队列中的下一地址的存取请求。BIU 70 提供待定地址 58 给指令定序器 40，以向指令定序器 40（以及收回控制

逻辑 50) 指出对应于在系统总线 20 上当前待定(等待在总线当前数据阶段完成时激活)的地址请求的地址。待定地址 58 因此可以是比较经由地址/属性 74 提供给 BIU 70 的最近地址更早的地址。注意,握手控制和其他控制或状态信息可以在 BIU 70 和指令定序器 40 之间经由 CTL/STATUS 76 和/或请求 72 进行传递。

注意,在一个实施例中,定序器控制逻辑还可用于控制由 BIU 70 提供给执行单元 80 的数据的预取和取操作。例如,如果指令解码器 44 解码加载或存贮指令,地址生成逻辑 52 可生成读/写数据的适当地址,或者可接收来自加载/存贮单元 82 的对应地址。这个地址还可经由地址/属性 74 提供给 BIU 70。而且,定序器控制逻辑 42 可指示地址生成逻辑 52 从一系列地址中预取数据。

一旦定序器控制逻辑 42 检测到流的改变,诸如检测到分支指令,定序器控制逻辑 42 将指令预取流切换到流改变的目标。也就是,定序器控制逻辑 42 指示地址生成逻辑 52 生成开始于目标地址的预取地址,其中,地址生成逻辑 52 可基于从 IR 46 所提供的信息来计算目标地址。生成的地址提供给 BIU 70, BIU 70 将来自新指令预取流的指令提供给指令缓冲存贮 66。在一个实施例中,来自新指令预取流的指令覆盖先前存在的指令,使得 IR 46 和指令解码器 44 操作在正确指令上。而且,注意,在一个实施例中,如果指令缓冲器 66 为空,数据(包括取或预取的指令)可直接提供给 IR 46,这样就跳过了指令缓冲器 66。而且,握手和其他状态和/或控制信息可以在指令定序器 40 和指令缓冲存贮 66 之间经由控制/状态 64 而传递。

定序器控制逻辑 42,基于指令解码器 44 所解码的当前指令,可确定是否应该收回在系统总线 20 上的当前待定存取请求(对应于待定地址 58)。例如,在流改变的情况中,已知不再需要当前待定预取存取请求,因为现在将取开始于新目标地址的指令。如果允许该当前待定存取请求首先完成,主控 12 将延迟,因为流指令改变的目标不能被

接受和激活，直至当前待定预取请求（要被丢弃的请求）完成。在另一例子中，定序器控制逻辑 42 可能希望允许数据存取比指令预取优先。在此例子中，如果当前待定存取请求是指令预取请求，定序器控制逻辑 42 一旦检测到指令请求指令解码器 44 解码数据存取，就可能想要收回当前待定存取请求（即，指令预取请求），由此数据存取可能更早执行。

如果定序器逻辑 42 希望收回系统总线 20 上当前待定存取请求（对应于待定地址 58），定序器控制逻辑 42 声明收回请求 54，向收回控制逻辑 50 指示：应该收回当前待定存取请求。在一个实施例中，收回控制逻辑 50 使用储存在总线主控 12 中（例如储存在存取收回控制寄存器 48 中）的存取收回控制信息，基于当前待定存取请求的特性来确定是否允许收回当前待定存取请求。这些特性可包括，例如，对应于当前待定存取请求的地址（例如待定地址 58）、存取类型（例如读/写、指令/数据、突发/非突发等等）、存取的顺序或次序、存取的地址（例如，该地址范围正在被存取或者该设备正在被存取）、请求收回的总线主控（例如在多主控系统中）、或上述的任意组合。例如，在一个实施例中，储存在存取收回控制寄存器 48 中的信息可对应于特定地址范围。也就是，提供给收回控制逻辑 50 的信息可取决于当前待定存取请求的地址。在该实施例中，地址比较逻辑 60 可用于比较待定地址 58（对应于当前待定存取请求）和各种地址范围。存取收回控制寄存器 48、地址比较逻辑 60 和地址范围、以及用于允许或不允许收回请求的当前待定地址请求的特性，将在下面结合图 3-5 进行更加详细地讨论。

在另一实施例中，收回控制逻辑 50 使用储存在总线主控 12 中（例如储存在存取收回控制寄存器 48 中）的存取收回控制信息，基于当前待定存取请求之后新请求的存取的特性而确定是否允许收回当前待定存取请求。这些特性可包括，例如，对应于新存取请求的地址、存取类型（例如读/写、指令/数据、突发/非突发等等）、存取的顺序或次序、存取的地址（例如，该地址范围正在被存取或者该设备正在被存取）、

请求收回的总线主控（例如在多主控系统中）、或上述的任意组合。例如，在一个实施例中，储存在存取收回控制寄存器 48 中的信息可对应于特定地址范围。也就是，提供给收回控制逻辑 50 的信息可取决于新存取请求的地址。在该实施例中，地址比较逻辑 60 可用于比较新请求的地址和各种地址范围。如上所提及的，存取收回控制寄存器 48、地址比较逻辑 60 和地址范围、以及用于允许或不允许收回请求的当前待定地址请求的特性，将在下面结合图 3-5 进行更加详细地讨论。但是，注意，为图 3-5 所提供的结合当前待定存取请求的描述还将类似地适用于后续的存取请求。

回过头来参看图 2，如果收回控制逻辑 50 确定允许收回，收回控制逻辑 50 声明收回指示 62，以向 BIU 70 通知：通过移置或替换当前待定存取请求为对系统总线 20 的新的存取请求，来收回系统总线 20 上的当前待定存取请求。该新的存取请求可以是，例如，在 BIU 70 的地址队列中等待的下一个存取请求，如上所述，或者可以是供应给 BIU 70 的新生成的请求。在可替换实施例中，BIU 70 可以通过立即将预定值返回到指令缓冲存贮 66 来收回当前待定的存取请求。该预定值可以是任何虚拟值，因为其将会被丢弃。这样，当前待定的存取请求快速完成，下一存取请求可以开始，而不会延迟主控 12。其他的替换实施例可以使用其他方法来收回当前待定的存取请求。在某些实施例中，不返回任何值到指令缓冲存贮 66。

但是，如果收回控制逻辑 50 不允许收回当前待定的存取请求，收回控制逻辑 50 不会声明（即，其否定，或继续否定）收回指示 62。这样，主控 12 等待，直至由于流的变化而在进行新指令预取流之前完成当前待定存取请求。收回控制逻辑 50 还可以向地址生成逻辑 52 指示收回不被允许。

因此，注意，在一个实施例中，存取收回控制寄存器 48 和收回控制逻辑 50 可用于选择性超越总线协议，以便，例如，减少或防止延迟。

在一个实施例中，系统总线 20 的总线协议可能要求已经开始并且在系统总线 20 上当前待定的现有预取存取请求必须被允许完成，即使其可能在完成时被丢弃。例如，在流变化的例子中，已知不再需要当前待定预取存取请求，因为现在将从新目标地址开始取指令。但是，系统总线 20 的总线协议可能仍旧要求完成在系统总线 20 上当前待定的预取，使得主控 12 延迟。如上所述，定序器控制逻辑 42 可请求收回当前待定存取请求，从而防止该延迟。在此情况下，收回控制逻辑 50 基于储存在存取收回控制寄存器中的信息以及当前待定存取请求的特性，来选择性地超越总线协议：收回当前待定存取请求并且不使其像通常一样完成。因此，在允许收回的情况下，总线协议被超越，以防止延迟。在其他的实施例中，可使用不同类型的超越控制寄存器来超越总线协议的其他方面。

根据本发明一个实施例，在图 6 的流程 140 中图示说明了当前待定存取请求的收回。流程 140 开始于模块 144，这里首先确定存取请求在系统总线 20 上当前待定(否则，没有要收回的当前待定的存取请求)。流程随后进行到判定菱形框 146，确定是否应该收回当前待定的存取请求。因此，如上所述，定序器控制逻辑 42 可确定应该收回当前待定存取请求。如果否，流程前进到模块 148，这里系统(例如，数据处理系统 10)等待当前待定存取请求完成。但是，如果当前待定存取请求应该被收回，流程从判定菱形框 146 前进到模块 150，这里生成对应于当前待定存取请求的收回请求(例如收回请求 54)。在模块 150 之后，流程前进到判定菱形框 152，这里基于当前待定存取请求的特性来确定是否允许收回当前待定的存取请求(例如，由收回控制逻辑 50 确定)。如果在判定菱形框 152 允许收回的话，流程前进到模块 156，这里允许收回(例如收回指示 62)并且收回当前待定的存取请求。但是，如果不允许收回的话，流程前进到模块 154，这里拒绝收回请求并且系统等待当前待定存取请求完成。因此，当在系统总线 20 上有当前待定存取请求时使用流程 140，其中，该当前待定存取请求可以选择性被收回。

回过头来参看图 2，在一个实施例中，收回控制逻辑 50 使用储存在存取收回控制寄存器 48 中的存取收回控制信息，基于当前待定存取请求的特性而确定是否允许收回当前待定存取请求。图 3-5 图示说明了存取收回控制寄存器 48 的各种实施例，指示出允许收回的条件（或者，可替换地，指示出不允许收回的条件）。参看图 3 的实施例，存取收回控制寄存器 48 提供对应于待定存取类型（例如指令或数据、读或写、等等）的控制信息。例如，图 3 的存取收回控制寄存器 48 包括三个字段：指令读收回控制字段 100、数据读收回控制字段 102、以及数据写收回控制字段 104。在一个实施例中，每一字段都是一比特，当被声明时，表示允许该类型存取的收回。例如，如果声明了指令读收回控制字段 100，并且如果当前待定存取请求是指令读类型的存取（例如，由对应于当前待定存取请求的属性来表示，这里，在一个实施例中，待定地址 58 还扩充有属性），则允许收回当前待定存取请求。但是，如果指令读收回控制字段 100 被否定，则将拒绝收回。因此，根据当前待定存取请求的特性（例如存取请求类型是读还是写，是数据还是指令），收回控制逻辑 50 可使用储存在存取收回控制寄存器 48 中的存取收回控制信息来允许或拒绝来自定序器控制逻辑 42 的收回请求。

在图 3 的实施例中，注意，存取类型用于指示是否允许收回，而不管正在被存取的设备是什么。例如，如果指令读收回控制字段被声明，对于指令读类型的存取，允许收回，而不管是对存储系统 14、受控 16 还是受控 18 进行存取。但是，在下面将描述的替换实施例中，除了存取类型之外，还可以基于正在被存取的设备来允许或拒绝收回。

在另一替换实施例中，图 3 的字段可对应于当前待定请求之后的新请求的特性，这里，收回控制逻辑 50 可使用对应于新的后续请求的存取收回控制信息来允许或拒绝来自定序器控制逻辑 42 的收回请求（收回当前待定存取请求）。例如，在这个实施例中，如果指令读收回控制字段 100 被声明，并且如果当前待定存取请求之后的新存取请求是指令读类型的存取，则允许当前待定存取的收回。

图 4 图示说明了存取收回控制寄存器 48 的另一实施例，其包括多个（例如 4 个）地址范围收回控制字段：地址范围 1 收回控制字段 106、地址范围 2 收回控制字段 108、地址范围 3 收回控制字段 110、地址范围 4 收回控制字段 112。图 4 的每一收回控制字段因此可对应于特定地址范围（其中，在替换实施例中，存取收回控制寄存器 48 可包括任意数量的地址范围收回控制字段）。例如，地址范围可标识存储器中（例如图 1 的存储系统 14 中）所有或特定范围的位置，或者可标识耦合到系统总线 20 的任意设备（诸如受控 16 或受控 18）内的所有或特定范围的位置。例如，在一个实施例中，地址范围之一可对应于耦合到系统总线 20 的总线桥 22 的地址范围，其中，该地址范围可指示对于耦合到系统总线 30 的任何设备来说，允许还是拒绝收回。可替换地，地址范围之一可具体对应于耦合到系统总线 30 的任何设备（其经由总线桥 22 耦合到系统总线 20）内的位置范围。

在一个实施例中，每一地址范围收回控制字段的地址范围都在诸如图 2 的地址比较逻辑 60 的地址比较逻辑中定义。例如，可以通过使用上下边界或使用开始位置和掩码来定义地址范围。参见图 2 的实施例，地址比较逻辑 60 因此可比较待定地址 58（对应于定序器控制逻辑 42 希望收回的当前待定地址）与每一定义的地址范围，以确定待定地址 58 落入哪个范围（如果有的话）。例如，如果待定地址 58 落入地址范围 2，则（图 4 的）地址收回控制寄存器 48 的地址范围 2 收回控制字段 108 用于指示是否允许收回待定地址 58。例如，如果其被声明，则允许收回了；但是，如果其被否定，则拒绝收回。但是，如果待定地址 58 没有落入对应于图 4 实施例的范围 1-4 中任何一个，则不允许收回待定地址 58。在可替换实施例中，地址范围收回控制字段所定义的范围可定义不允许收回的范围，则其他所有地址位置都可被收回。

注意，图 3 和 4 的实施例可合并，使得对于允许还是拒绝收回的判定基于存取类型和地址范围这两者。例如，在一个实施例中，对于

每一地址范围，对应指令读、数据读、和数据写控制字段（或其子集）可用于进一步定义何时允许收回。例如，参看图 1，地址范围收回控制字段可对应于受控 16，使得如果对受控 16 进行存取请求，则对应于该存取请求的特定地址将落入这个范围中。而且，对于对应于受控 16 的特定地址范围，用户可能希望允许仅对数据读和写允许收回，而不对指令读允许收回。在此情况下，具体到特定地址范围的数据读、数据写和指令读收回控制字段用于确定是否允许存取。因此，可使用当前特定存取的特性的任何组合来确定是否允许收回。

图 5 图示说明存取控制寄存器 48 的又一个实施例，其中，基于前一个存取类型或基于存取类型的次序或顺序来允许或拒绝收回。例如，在图 5 的实施例中，存取收回控制寄存器 48 包括指令后指令（IFI）收回控制字段 114、指令后数据读（DRFI）收回控制字段 116、指令后数据写（DWFI）收回控制字段 118、数据读后指令（IFDR）收回控制字段 120、数据读后数据读（DRFDR）收回控制字段 122、数据读后数据写（DWFDR）收回控制字段 124、数据写后指令（IFDW）收回控制字段 126、数据写后数据读（DRFDW）收回控制字段 128、以及数据写后数据写（DWFDR）收回控制字段 129。因此，在图 5 的实施例中，基于当前待定存取和前一个存取类型来允许或拒绝收回。例如，定序器控制逻辑 42 可追踪前一个存取类型，使得当需要收回当前待定存取请求（对应于特定地址 58）时，可基于前一个存取来允许或拒绝。例如，如果前一个存取是数据写，且要收回的当前待定存取是指令存取（即指令读），则收回控制逻辑 50 使用 IFDW 收回控制字段 126 来确定是否允许收回。因此，如果字段 126 被声明，则允许当前待定存取的收回；但是，如果被否定，就拒绝收回。在另一实施例中，当涉及当前待定存取的特性时，图 5 中所述的字段可基于新请求的存取的特性而应用。例如，如果当前待定存取是指令读，而新请求的存取是数据写，则检查 DWFI 收回控制字段 118 来确定是否允许收回当前待定的指令存取。

如同图 3 和 4 的实施例一样，图 5 中字段 114-128 定义的特性可组合其他特性（诸如图 3 的字段 100-104 和/或图 4 的字段 106-112 所定义的特性）来使用。而且，存取收回控制寄存器 48 可按需要包括具有任意数量字段的任意数量的寄存器（一个或多个）。此外，可按需要定义任意数量的地址范围，其边界储存在收回控制逻辑 50 中或主控 12 中。而且，注意，图 3-5 的收回控制字段被描述为单一比特字段，其可以被声明或否定来指示允许还是拒绝收回。但是，在替换实施例中，每一字段可以按需要包括任意比特数，以指示收回的允许与否或收回的等级。此外，可使用任意类型的存贮电路来储存在存取收回控制寄存器 48 中储存的存取收回控制信息。

还要注意，在替换实施例中，不是使用图 3-5 中所述的当前待定存取的特性，而是可以类似方式替代地使用当前待定存取请求之后的新存取的特性，来确定是否允许当前待定存取请求的存取收回。因此，在这些替换实施例中，图 3-5 的字段中的信息将对应于新存取请求，而不是当前待定的存取请求，以便确定是否允许当前待定存取请求的存取收回。在又一替换实施例中，可使用任意数量的存取收回控制寄存器或存取收回控制寄存器中的字段，其中，当前待定存取请求的选择性收回可基于当前待定存取请求和新存取请求这两者。例如，存取收回控制寄存器 48 可包括一个寄存器或一组寄存器，对应于当前待定存取地址的特性，并且包括另一个寄存器或另一组寄存器，对应于新（即后续的）存取请求的特性。

注意，上面提供的描述是结合单独一个主控系统（例如图 1 的数据处理系统 10）来进行描述的。但是在替换实施例中，数据处理系统 10 可包括多个总线主控。在此实施例中，用于确定允许还是拒绝收回的当前待定存取的特性还可包括哪一总线主控进行该存取请求。例如，可使用分立的地址收回控制寄存器（或者一组地址收回控制寄存器），诸如结合图 3-4 所描述的那样，其中每一寄存器（或每一组寄存器）对应于不同的总线主控。在这个实施例中，BIU 70 还可指示收回控制逻辑

辑 50: 哪一总线主控进行了当前待定的存取 (对应于待定地址 58) 。例如, 每一总线主控可具有对应的唯一的总线主控标识符, 其经由系统总线 20 与每一存取请求一起提供。而且, 注意, 每一总线主控可按需要具有不同的对应存取收回控制寄存器和/或字段。

因此, 可以认识到, 图 1-6 所述的实施例允许基于当前待定存取的一个或多个特性、或者当前待定存取请求之后的新存取请求的一个或多个特性、或者当前待定存取请求和新存取请求中每一存取请求的一个或多个特性, 来选择性收回当前待定存取。这样, 总线主控 12 可更好地控制其存取请求, 这样减少了延迟并且改善了性能。这一个或多个特性可包括例如, 存取类型 (例如, 读/写、指令/数据、突发/非突发等等)、存取的顺序或次序、存取的地址 (例如, 正在存取哪个地址范围、或正在存取哪一设备)、请求收回的总线主控 (例如在多主控系统中)、或者上述的任意组合。而且, 注意, 可以动态地、一个存取接一个存取地执行上面结合图 1-6 所述的选择性收回。而且, 可使用存取的特性来选择性超越总线协议的其他方面, 类似于上述的选择性收回。例如, 是否一次请求完全完成突发存取, 或者是否可以在完成之前中断该突发存取。

上面结合图 1-6 所述的存取收回的概念还可应用于总线仲裁, 其也可以基于当前待定存取请求 (和/或当前待定存取请求之后的新存取) 的一个或多个特性来收回当前待定存取。例如, 图 7 图示说明根据本发明一个实施例的能够将 M 个主控耦合到 N 个受控的系统互连 200。在图示说明的实施例中, 每一主控端口 0-2 可与每一受控端口 0 和 1 进行通信。每一主控端口可具有耦合到其的一个或多个主控, 每一受控端口可具有耦合到其的一个或多个受控。注意, 如果超过一个主控或受控耦合到特定端口, 则可能需要附加的仲裁逻辑 (未示出) 来在分别耦合到相同主控或受控端口的多个主控或受控之间进行仲裁。而且, 注意, 耦合到主控端口的主控的类型和耦合到受控端口的受控的类型可能是与上面结合总线主控 12 和受控 16、18 所述相同的类型。

在这里讨论的实施例中，将假定每一主控端口具有耦合到其的对应的主控。例如，尽管没有显示出来，主控端口 0 具有耦合到其的主控 0，主控端口 1 具有耦合到其的主控 1，主控端口 2 具有耦合到其的主控 2。类似地，将假定每一受控端口具有耦合到其的对应的受控，其中，例如，受控端口 0 耦合到受控 0，而受控端口 1 耦合到受控 1。

而且，注意，任何数据处理系统，诸如数据处理系统 10，可利用系统互连，诸如系统互连 200，来替换系统总线。例如，回过头来看图 1，可替换系统总线 20 而使用系统互连 200，其中，总线主控 12 可耦合到主控端口 0-2 之一，受控 16 可耦合到受控端口 0 或 1 之一，受控 18 可耦合到受控端口 0 或 1 中另一个。而且，附加的主控，诸如总线主控 12，可耦合到主控端口 0-2 中的剩余两个主控端口。还要注意，图 7 图示说明了三个主控端口和两个受控端口，但应该理解，系统互连 200 可按需要包括任意数量的主控端口和任意数量的受控端口。

在图 7 的系统互连 200 中，可能希望收回从主控到受控端口的存取请求，以便于后续请求。例如，主控的存取请求可能在进行中（例如，当前存取请求的地址可驱动到受控端口上，由受控取出或接受，因此等待相应的数据从受控返回）。但是，数据的返回可能被延迟，由此增加完成进行中的存取请求所需的等待状态的数量。当该存取请求进行中并且等待返回数据时，相同的主控或不同的主控可能提供另一存取请求给相同的受控端口。该存取请求可能由系统互连内的仲裁逻辑授权，因此变为驱动到受控端口上的当前待定的存取请求。但是，尽管其授权并驱动到受控端口上，该当前待定存取请求不能被受控执行，直到当前进行中的存取请求完成（即数据返回）。因此，可能希望基于当前待定存取请求的特性（和/或当前待定存取请求之后的新存取请求的特性），来收回当前待定存取请求，以便于后面的到相同端口的存取请求，例如，如果后续的存取请求具有更高优先级的话。收回的当前待定的存取请求随后可储存在对应于受控端口的受控端口逻辑内，直到将后续存取请求驱动到受控端口上之后。可替换地，可以

储存收回的当前待定存取请求，直到随后再次将其驱动到受控端口上的稍后时间点。这样，可提高效率。关于总线仲裁的存取收回的进一步的细节将结合图 7-12 进行描述。

图 7 的系统互连 200 包括主控端口 0-2、受控端口 0 和 1、受控端口 0 逻辑、受控端口 1 逻辑，其中每一主控端口 0-2 经由受控端口 0 逻辑耦合到受控端口 0，每一主控端口 0-2 经由受控端口 1 逻辑耦合到受控端口 1。受控端口 0 逻辑包括仲裁逻辑 216、收回控制逻辑 218、主控优先级寄存器 210、止动（parking）控制寄存器 212、存取收回控制寄存器 214、以及存取选择器 208。每一主控端口 0-2 分别经由导线 202-204 耦合到受控端口 0 逻辑的存取选择器 208、仲裁逻辑 216、和收回控制逻辑 218 以及耦合到受控端口 1 逻辑。存取选择器 208 在受控端口 0 提供来自导线 202-204（分别对应于主控端口 0-2）之一的信息给导线 205。也就是，导线 205 还可称为受控总线，其耦合到对应的受控，即受控 0。仲裁逻辑 216 将当前授权 220 提供给存取选择器 208 和收回控制逻辑 218。仲裁逻辑 216 还将待定授权 222 和收回请求 224 提供给收回控制逻辑 218，并且从收回控制逻辑 218 接收收回指示 226。仲裁逻辑 216 耦合到主控优先级寄存器 210 和止动控制寄存器 212。收回控制逻辑 218 耦合到主控优先级寄存器 210、止动控制寄存器 212 和存取收回控制寄存器 214。

注意，没有显示受控端口 1 逻辑内的电路；但是，其与受控端口 0 逻辑内的电路类似，将在下面进行更详细地描述。也就是，受控端口 1 逻辑的操作类似于受控端口 0 逻辑的操作。

在操作中，受控端口 0 逻辑控制从耦合到主控端口 0-2 的主控到受控端口 0 的存取。仲裁逻辑 216 经由导线 202-204 接收对受控端口 0 的存取请求，并且确定对哪个授权以及何时授权。仲裁逻辑 216 随后将当前授权 220 提供给存取选择器 208，当前授权 220 是指出已经授权哪个存取请求的控制信号。存取选择器 208 基于当前授权 220，提供来

自从主控端口 0-2 之一到受控端口 0 的授权存取请求的信息，由此将地址、数据和控制信号从当前授权存取请求的主控驱动到耦合于受控 0 的导线 205。在确定对哪个存取请求授权时，仲裁逻辑 216 可使用主控优先级寄存器 210 和止动控制寄存器 212。在将信息从当前授权存取请求的主控驱动到导线 205 之后，受控 0 取出或接受存取请求。受控 0 一旦取出存取请求，就接收并储存对应于当前授权存取请求的地址，以便提供或接收对应于存取请求的请求的数据。一旦取出存取请求，其变为当前进行中的存取请求。

随后，仲裁逻辑 216 可通过提供当前授权 220 给对应于新存取请求的存取选择器 208 而授权新存取请求。对应于该新存取请求（来自主控 0-2 之一）的信息随后被存取选择器 208 驱动到耦合于受控 0 的导线 205。但是，该新存取请求保留当前待定存取请求，同时，前面授权的存取请求仍旧在进行中。也就是，在所图示说明的实施例中，直到仍旧进行中的前面授权的存取请求完成之后，才能由受控 0 取出或执行新存取请求。在一个实施例中，在受控 0 接收到对应于存取请求的地址或将地址提供给主控 0-2（其中，主控 0-2 中的请求主控接受数据）之后，完成数据前面授权的存取请求。

由于前面授权的存取请求可能是在未知时期内进行的，仲裁逻辑 216 可决定当前待定存取请求（等待前一授权存取请求完成的那个存取请求）应该收回，以便于后续的存取请求。在此情况下，仲裁逻辑 216 通过声明收回请求 224 来请求收回当前待定的存取请求。收回控制逻辑 218 使用来自存取收回控制寄存器 214 的信息，基于当前待定存取请求和/或后续存取请求的至少一个或多个特性（这里，这一个或多个特性可以根据经由导线 202-204 从请求主控提供的信息、经由待定授权 222 提供的当前待定存取请求的地址、和/或根据主控优先级寄存器 210 和止动控制寄存器 212 中的信息来确定）来确定是否允许收回当前待定的存取请求。如果收回控制逻辑 218 确定允许收回当前待定存取，其声明收回指示 226，而如果不允许收回，则收回指示 226 被否定（或

者保持被否定)。如果允许收回,则仲裁逻辑,经由当前授权 220 和存取选择器 208,可将受控端口 0 上导线 205 上的当前待定存取请求替换为新的后续存取请求。仲裁逻辑 216 随后可以储存被收回的当前待定存取请求,由此在将新的后续存取请求驱动到导线 205 上之后,可以再次驱动收回的请求。可替换地,其可以在稍后的时间点被再次驱动。

在图 7 的图示说明的实施例中,受控端口 0 逻辑包括主控优先级寄存器 210 和止动控制寄存器 212。这些寄存器的例子分别图示在图 8 和 9 中。例如,图 8 图示说明了主控优先级寄存器 210 的一个实施例,其包括对应于每一主控端口的主控优先级字段:主控 0 优先级字段 230、主控 1 优先级字段 232、主控 2 优先级字段 234。因此每一字段可包括对应主控端口的优先级。例如,在包括 3 个主控端口的图示的实施例中,可使用 2 比特值(例如 11、10、01、00)来指示优先级,其中,最高优先级主控端口可以被赋予 11 的优先级,第二高的被赋予 10 的优先级,第三高的(即 3 个中最低的)被赋予 01 的优先级。在此情况下,可以不使用 00 值。可替换地,优先级可以编号为从 10(最高优先级主控端口)到 00(最低优先级主控端口),其中可不使用值 11。在又一实施例中,优先级可编号为 10(最低优先级主控端口)到 00(最高优先级主控端口)。可替换地,可以使用其他方案来为每一端口分配优先级。而且,注意,在一个实施例中,如果主控端口不具有耦合到其的主控的话,相应的主控优先级字段可以用预定值编程以指出这一点,或者可以简单地给出最低优先级。

图 9 图示说明了止动控制寄存器 212 的一个实施例,其包括止动使能字段 236 和止动主控字段 238。在一个实施例中,止动使能字段 236 是一比特字段,当其被声明时,指示对于对应的受控端口(在本实施例中是受控端口 0)能够止动,而当其被否定时,指示对于对应的受控端口不能够止动。如果能够止动,由止动主控字段 238 中的值提供止动主控。例如,止动主控字段 238 可以是 2 比特值,其标识主控端

口 0-2 之一。如果能够止动，并且对应受控端口空闲（没有与任何主控端口收发信息），受控端口自动耦合到止动主控，从而使得止动主控可以同受控端口通信，而不需要仲裁间隔。例如，如果受控端口 0 空闲，仲裁逻辑 216 使用止动控制寄存器 212 中的信息，提供当前授权 220（指示止动主控）给存取选择器 208，存取选择器 208 将止动主控端口耦合到受控端口 0 的导线 205。在此情况下，注意，当受控端口空闲时，止动主控不需要明确请求对受控端口的存取（假定能够止动）。这样，如果被止动的主控是受控端口 0 的下一请求者，不需要任何仲裁间隔，而且对于被止动的主控来说，减少了存取受控端口 0 的延迟。如果不能够止动，则仲裁逻辑 216 和存取选择器 208 如上结合图 7 所述进行操作。

结合图 12 的流程 260 中提供的例子可进一步理解受控端口 0 逻辑的操作。流程开始于模块 262，这里，从主控 0 到受控端口 0 的存取请求被授权（具有对应的存取地址  $addr_y$ ），同时，从主控 1 到受控端口 0 的当前存取请求（具有对应的存取地址  $addr_x$ ）在进行中。因此，如上面结合图 7 所述的那样，注意，来自主控 0 的存取请求变为当前待定存取请求，其不能被受控 0 接受或取出，直至来自主控 1 的当前进行中的存取完成。流程随后前进到模块 264，这里，从主控 1（或者，在可替换实施例中，从任何其他主控）接收到对受控端口 0 的后续的或新的存取请求（具有对应的存取地址  $addr_z$ ）。如上所述，由于可能不知道当前进行中的存取请求（ $addr_x$ ）有多长，仲裁逻辑 216 可能希望收回当前待定的存取请求（ $addr_y$ ），以便于新接收的存取请求（ $addr_z$ ），因为，例如，新接收的存取请求可能是更加重要的或是在时间上的要求更苛刻的。也就是，仲裁逻辑 216 可能希望对待定存取请求重新排序，以便于后续接收的存取请求。因此，回过头来看图 12，流程前进到模块 266，这里，生成对应于对受控端口 0 的当前待定主控 0 存取请求（ $addr_y$ ）的收回请求。例如，其可能对应于仲裁逻辑 216 声明收回请求 224，如结合图 7 所述的那样。

回过来看图 12，在生成了收回请求之后，流程从模块 266 前进到判定菱形框 268，这里，基于当前待定存取请求的一个或多个特性（和/或后续存取请求（ $addr_z$ ）的一个或多个特性）来确定是否允许收回当前待定主控 0 存取请求（ $addr_y$ ）。例如，这个判定可以由收回控制逻辑 218 使用来自主控优先级寄存器 210、止动控制寄存器 212、待定授权 222 的信息、来自导线 202-204 或上述任意组合以及存取收回控制寄存器 214 的信息来做出。如果不允许收回，流程从判定菱形框 268 前进到模块 270，这里，收回请求被拒绝（通过否定例如收回指示 226），受控端口 0 逻辑等待进行中的存取请求（ $addr_x$ ）完成，以及等待当前待定存取请求（ $addr_y$ ）被取出（由此变为进行中的新存取请求），之后再将随后接收到的存取请求（ $addr_z$ ）驱动到受控端口 0 的导线 205 上。如果允许收回的话，流程从判定菱形框 268 前进到模块 272，这里，当前待定的存取请求（ $addr_y$ ）被收回（通过声明例如收回指示 226）。在此情况下，当前待定的存取请求（ $addr_y$ ）被替换为新的存取请求（ $addr_z$ ），从而使得新的存取请求（ $addr_z$ ）变为当前待定的存取请求。这样，授权存取请求的排序就可能根据仲裁逻辑的观点进行重新排序，将先前待定的存取请求（ $addr_y$ ）替换为新存取请求（ $addr_z$ ），这里，替换的存取请求（ $addr_y$ ）可储存在受控端口 0 逻辑中，使得其可以在  $addr_z$  之后或在某个稍后的时间点再次驱动到受控端口 0 上。根据耦合到受控端口 0 的受控 0 的观点，当前待定存取请求（ $addr_y$ ）被收回，新的当前待定的存取请求（ $addr_z$ ）被代替地驱动，在完成当前进行中的存取请求（ $addr_x$ ）之后由受控取出。

如上所述，存取收回控制寄存器 214 内的信息指示允许收回的条件。（可替换地，存取收回控制寄存器 214 指示不允许收回的条件。）因此，存取收回控制寄存器 214 中的信息用于基于当前待定存取请求（或后续存取请求）的一个或多个特性来选择性地收回当前待定的存取请求。这些特性可包括上面结合图 3-5 所述的那些特性的任意一个，或者下面将结合图 10 和 11 所述的那些特性中的任意一个，及其任意组合。例如，如上所述，用于做出判定的当前待定存取请求的特性可

包括存取类型、地址范围、数据类型排序、及其组合。例如，对于地址范围，一个或多个地址范围可用来标识特定受控端口（诸如受控端口 0），或者可用来标识经由特定受控端口存取的位置范围。在此情况下，也如上面结合图 4 所述的那样，可以简单地基于当前待定或后续接收的存取请求所存取的特定地址范围（例如受控端口）来允许或拒绝收回，或者基于正在存取的特定地址范围（例如受控端口）以及当前待定或后续接收的存取请求的附加特性来允许或拒绝收回。在一个实施例中，注意，地址范围可以替代地直接被标识为特定受控端口。

图 10 图示说明了图 7 的存取收回控制寄存器 214 的一个实施例，其中，关于允许还是拒绝收回的判定基于请求主控的优先级（例如优先级  $\delta$ ）。例如，如果进行后续存取请求（ $\text{addr}_z$ ）的主控（例如图 12 的例子中的主控 0）具有比进行当前待定存取请求（ $\text{addr}_y$ ）的主控（例如图 12 的例子中的主控 1）高的优先级的话，则收回控制逻辑 218 可允许存取收回（使用图 10 的存取收回控制寄存器 214 所提供的信息）。例如，图 10 的存取收回控制寄存器 214 包括每主控端口一个优先级  $\delta$  收回控制字段：主控优先级  $\delta$  收回控制字段 240、主控 1 优先级  $\delta$  收回控制字段 242、主控 2 优先级  $\delta$  收回控制字段 244。在这个例子中，每一字段指示出后续存取请求的请求主控与当前待定存取请求的请求主控之间的优先级  $\delta$ （即优先级差）应该是多少，才能允许收回当前待定存取请求。

在一个实施例中，每一字段可以是 2 比特的值，这里，“00”值表示优先级  $\delta$  需要大于或等于 0，“01”值表示优先级  $\delta$  需要大于或等于 1，“10”值表示优先级  $\delta$  需要大于或等于 2，“11”值表示无论优先级  $\delta$  为多少，都不允许收回。注意，可以使用其他的  $n$  比特值（这里  $n$  可以是大于 0 的任意整数）来表示允许收回所需的优先级  $\delta$ 。例如，参看图 12 的例子，主控 0 进行当前待定存取请求（ $\text{addr}_y$ ），主控 1 进行后续存取请求（ $\text{addr}_z$ ）；因此，如果请求收回，使用在主控 0 优先级  $\delta$  收回控制字段 240 中提供的信息，以及主控

0 优先级字段 230 和主控 1 优先级字段 232（结合图 8 描述），来确定是否允许收回。如果主控 0 优先级 delta 收回控制字段 240 指示出，主控 1 和主控 0 之间的优先级 delta（这里，优先级 delta = 主控 1 优先级 - 主控 0 优先级）足够，则允许收回。如果优先级 delta 是 1，而主控 0 优先级 delta 收回控制字段 240 指示出 delta 需要大于或等于 2，则不允许收回。但是，如果优先级 delta 是 2，则将允许收回。如果主控 0 优先级 delta 收回控制字段 240 指示出 delta 大于或等于 0，则只要主控 1 具有至少与主控 0 相同的优先级，就允许收回。

图 11 图示说明了图 7 的存取收回控制寄存器 214 的另一个实施例，其具有各种不同字段，包括最高优先级主控收回控制字段 246、延迟收回控制字段 248、止动主控收回控制字段 250、以及非止动主控收回控制字段 252。在一个实施例中，每一字段 246、250 和 252 是一比特字段，其要么被声明要么被否定。例如，如果最高优先级主控收回控制字段 246 被声明，则当后续请求来自具有最高优先级的主控时，允许收回当前待定存取请求。如果止动主控收回控制字段 250 被声明，则当后续请求来自止动主控（例如，由止动控制寄存器 212 中的止动主控字段 238 指示出）时，允许收回当前待定存取请求。如果非止动主控收回控制字段 252 被声明，则当后续请求来自任何非止动主控（止动主控字段 238 所指出的主控之外的任意主控）时，允许收回当前待定存取请求。延迟收回控制字段 248 可用于提供延迟（d），使得如果允许收回的话，就在 d 个等待状态时钟之后进行收回，这里 d 可以是大于或等于 0 的任意整数。例如，在一个实施例中，如果 d 是 0，则延迟收回没有被使能；但是，如果 d 是 1 或更大，则将当前待定存取请求的收回（如果被允许）延迟 d 个时钟。在替换实施例中，延迟值可以提供延迟（d），使得如果允许收回的话，实际上只在完成当前进行中的存取（例如，对应于  $addr_x$ ）所消耗的时间大于或等于延迟时，执行收回。也就是，在这个替换实施例中，如果当前进行中的存取请求将在小于延迟（d）的时间内完成的话，则不会发生收回，即使允许收回。在替换实施例中，延迟值可以以一种可替换的方式与完成当前进

行中的存取所需的时间长度进行比较，以便确定是否会发生收回。即，在一个实施例中，只有延迟值与完成当前进行中的存取请求所占用的时间相比较结果为预定结果，才会发生当前待定存取的收回（如果允许）。

注意，图 10 和 11 中字段 240-252 定义的每一特性可单独或任意组合使用，还可以与其他特性相组合来使用，诸如与图 3 的字段 100-104 和/或图 4 的 106-112 和/或图 5 的字段 114-128 所定义的特性相结合。而且，存取收回控制寄存器 214 可按需要包括具有任意数量字段的任意数量的寄存器（一个或多个）。还有，注意，图 10 和 11 的收回控制字段可按需要包括任意数量的比特，用来指示是否允许收回或者收回的等级。而且，可使用任意类型的存贮电路来储存在存取收回控制寄存器 214 内储存的存取收回控制信息。

因此，可以认识到仲裁怎样使用当前待定存取请求或后续存取请求或二者的特性来选择性地收回当前待定存取请求。这样，通过减小延迟和允许高优先级或高重要性存取较早进行，可提高效率。

注意，尽管图 1、2、7 图示说明的是双向和单向导线的使用，应该明白，可以替代地使用单向和双向导线的不同组合。例如，可使用多个单向导线来代替双向导线。或者，可以使用双向导线来代替单向导线。也可以经由单一导线串行传送信号或者经由多个导线并行传送信号。可替换地，信号可被时间复用于一个或多个导线上。

在前面的说明中，结合具体实施例描述了本发明。但是，本领域普通技术人员应该认识到，可以进行各种修改和变化，而不背离本发明如权利要求所述的范围。例如，应该理解，数据处理系统 10 可以是任意类型的数据处理系统，其可以位于单芯片或者集成电路（IC）或者二者组合上。而且，系统互连 200 可以位于与主控和受控相同的芯片或 IC 上，或者可以位于与主控和受控分立的 IC 上。因此，说明书

和附图只是说明性的，而非限制含义，所有这样的修改都希望被包括在本发明的范围之内。

以上已经关于具体实施例描述了利益、其他优点和问题的解决方案。但是，利益、优点、问题的解决方案以及可招致任何利益、优点或解决方案发生或食指更加显著的任何元素，都不应被解释为任意或所有权利要求的关键的、需要的或必要的特征或元素。这里所使用的术语“包括”、“包含”及其任何其他变化形式，意图涵盖非排他性的包涵，因此，包括一系列元素的处理、方法、物品或装置不仅仅包括列出的元素，还可包括未明确列出或这样的处理、方法、物品或装置所固有的其他元素。

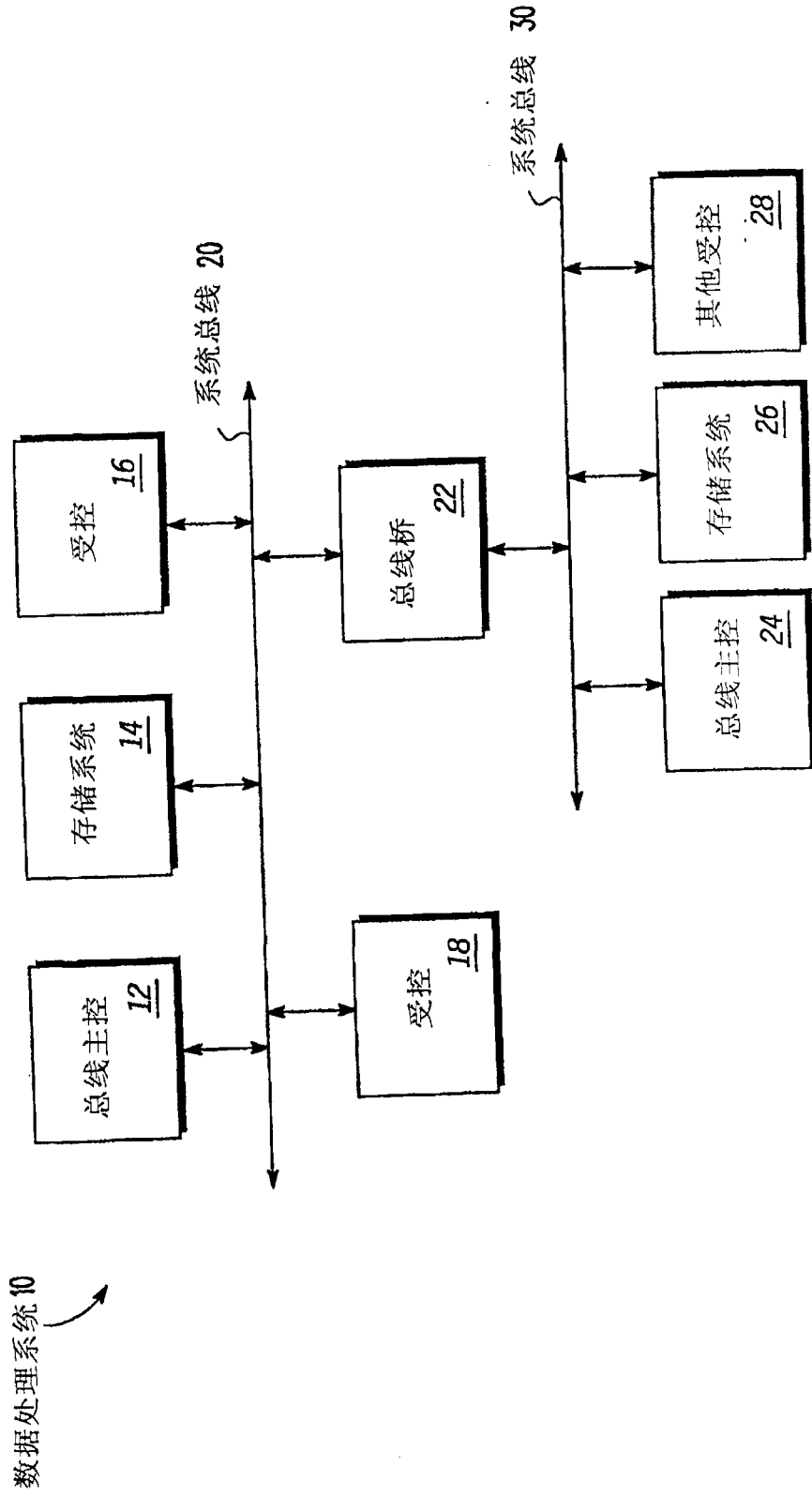


图1

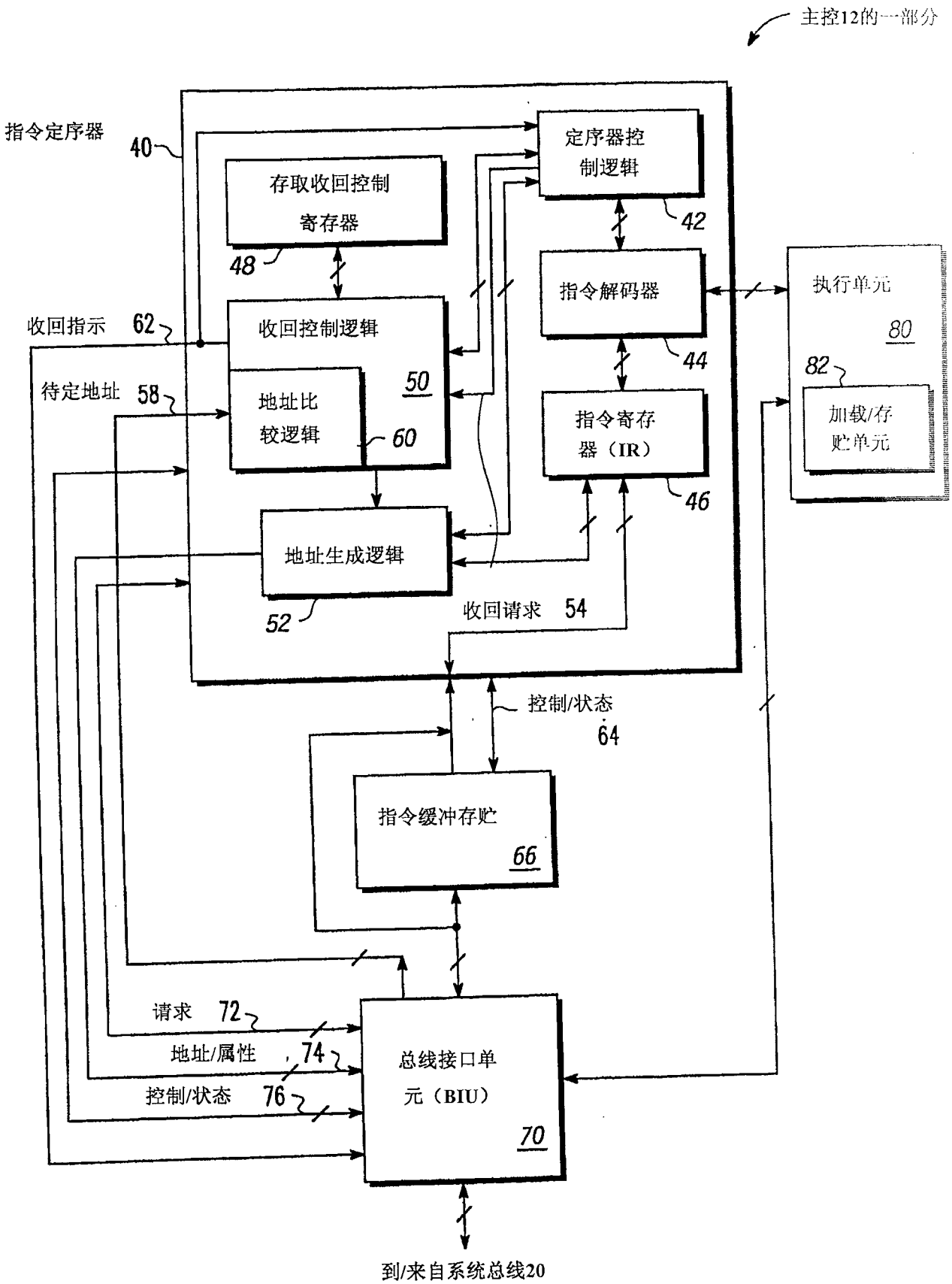


图2

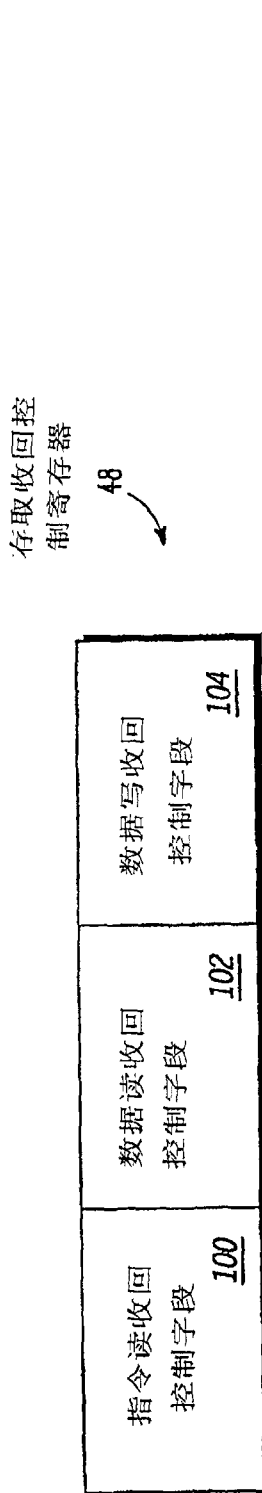


图3

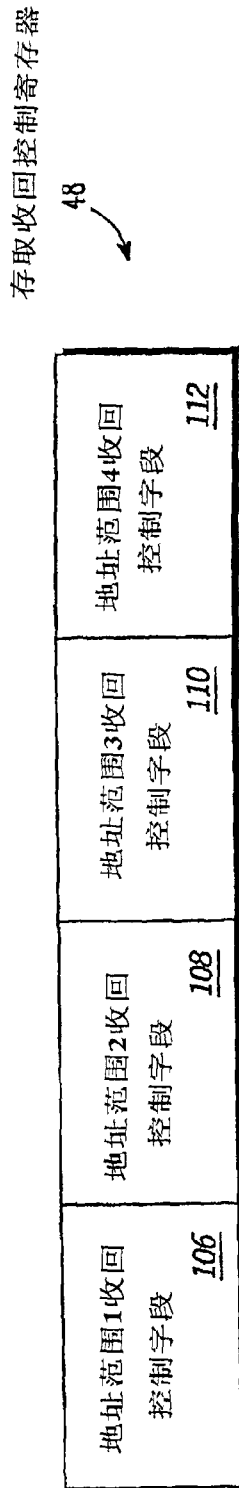


图4

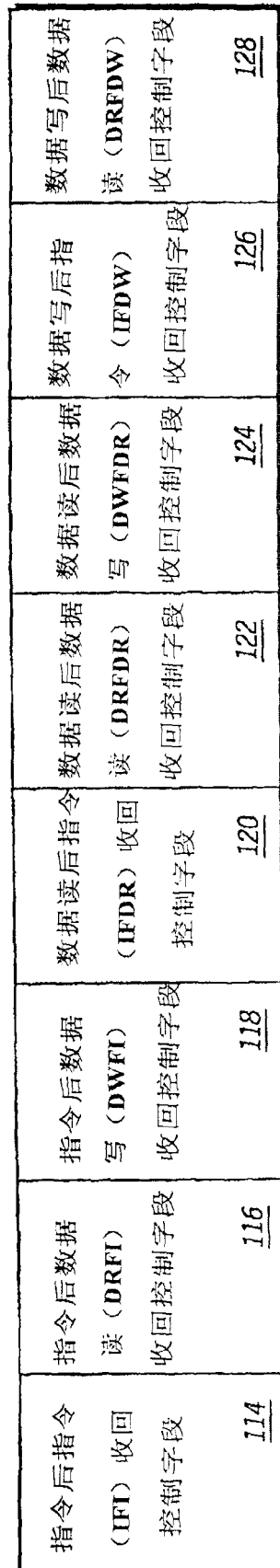


图5

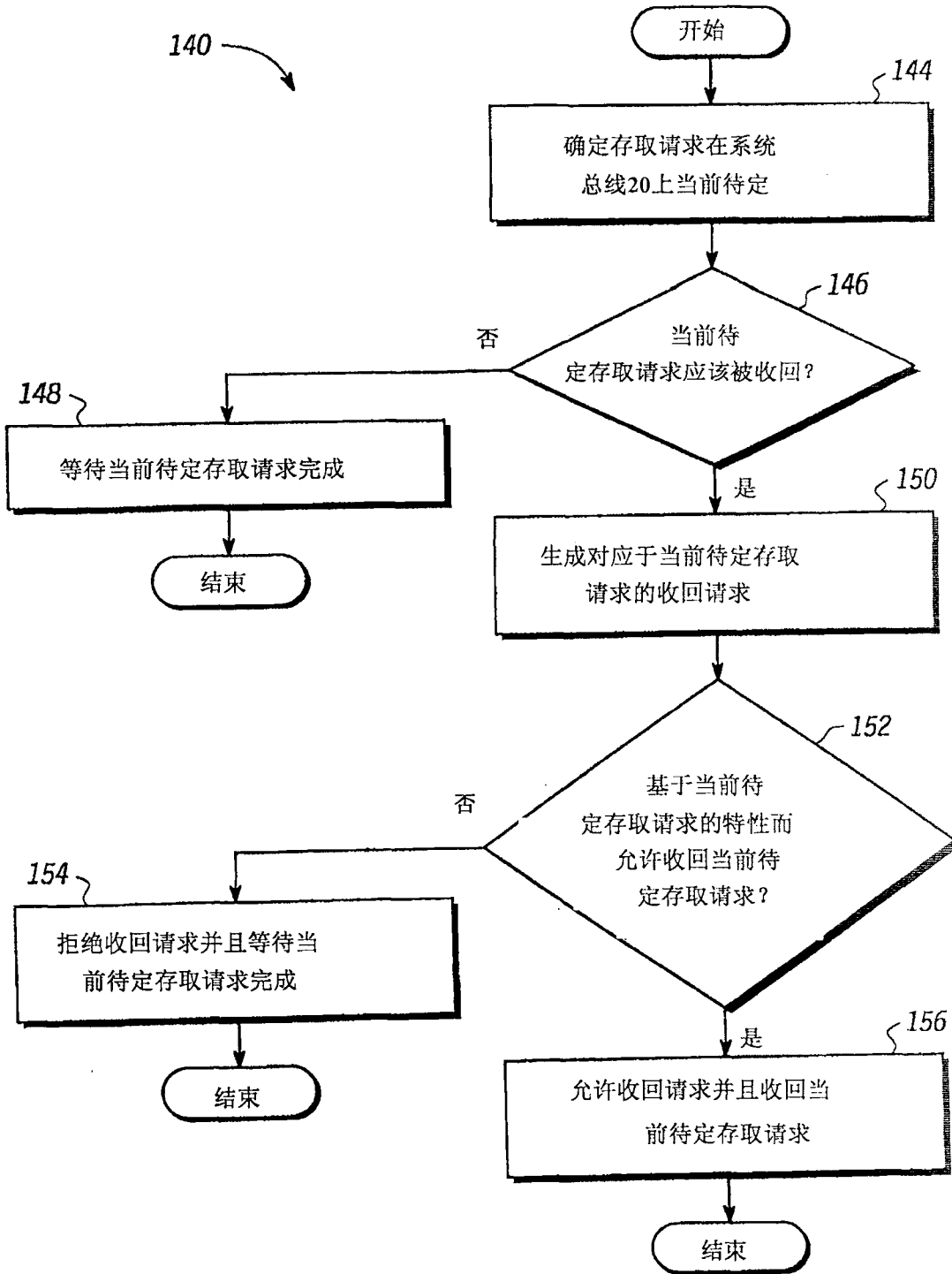


图6

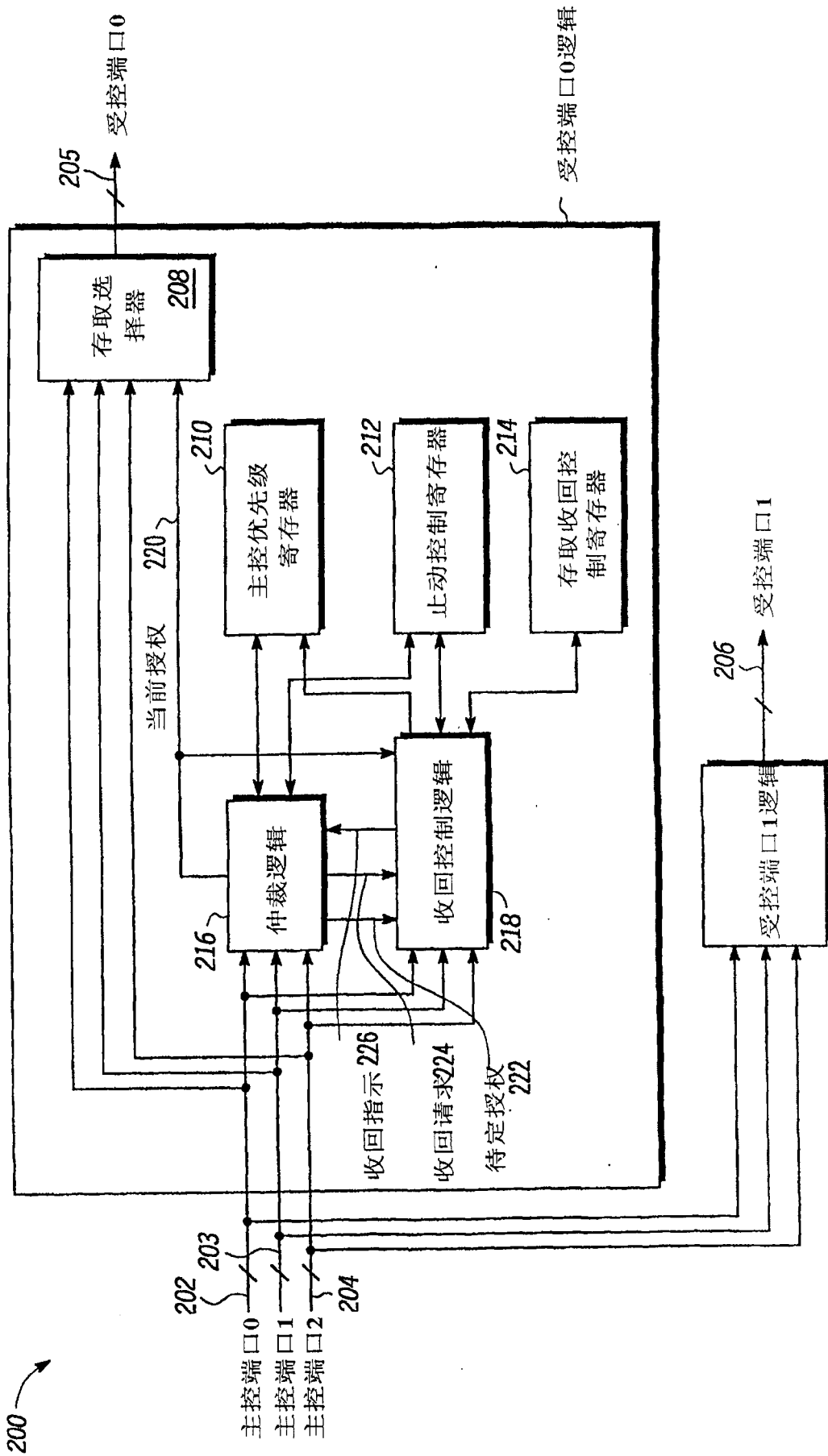


图7

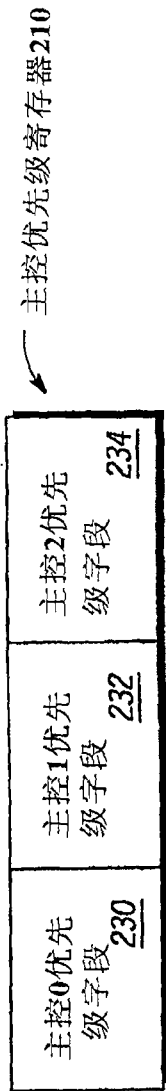


图8

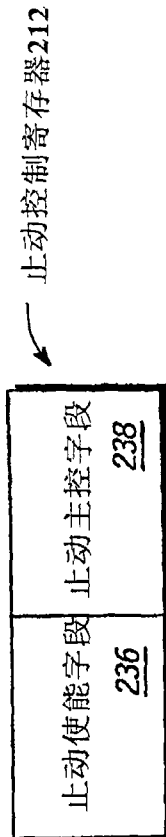


图9

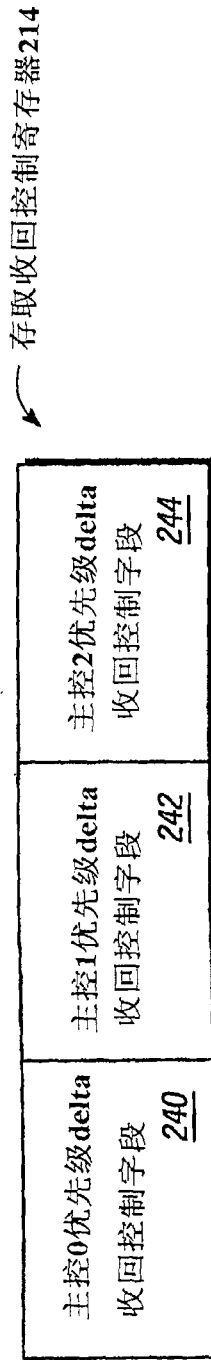


图10

存取收回控制寄存器214

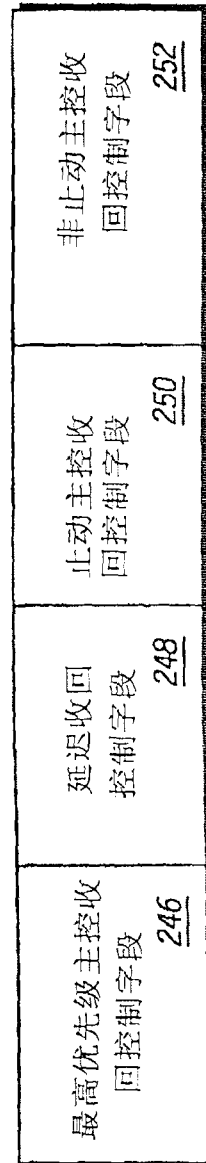


图11

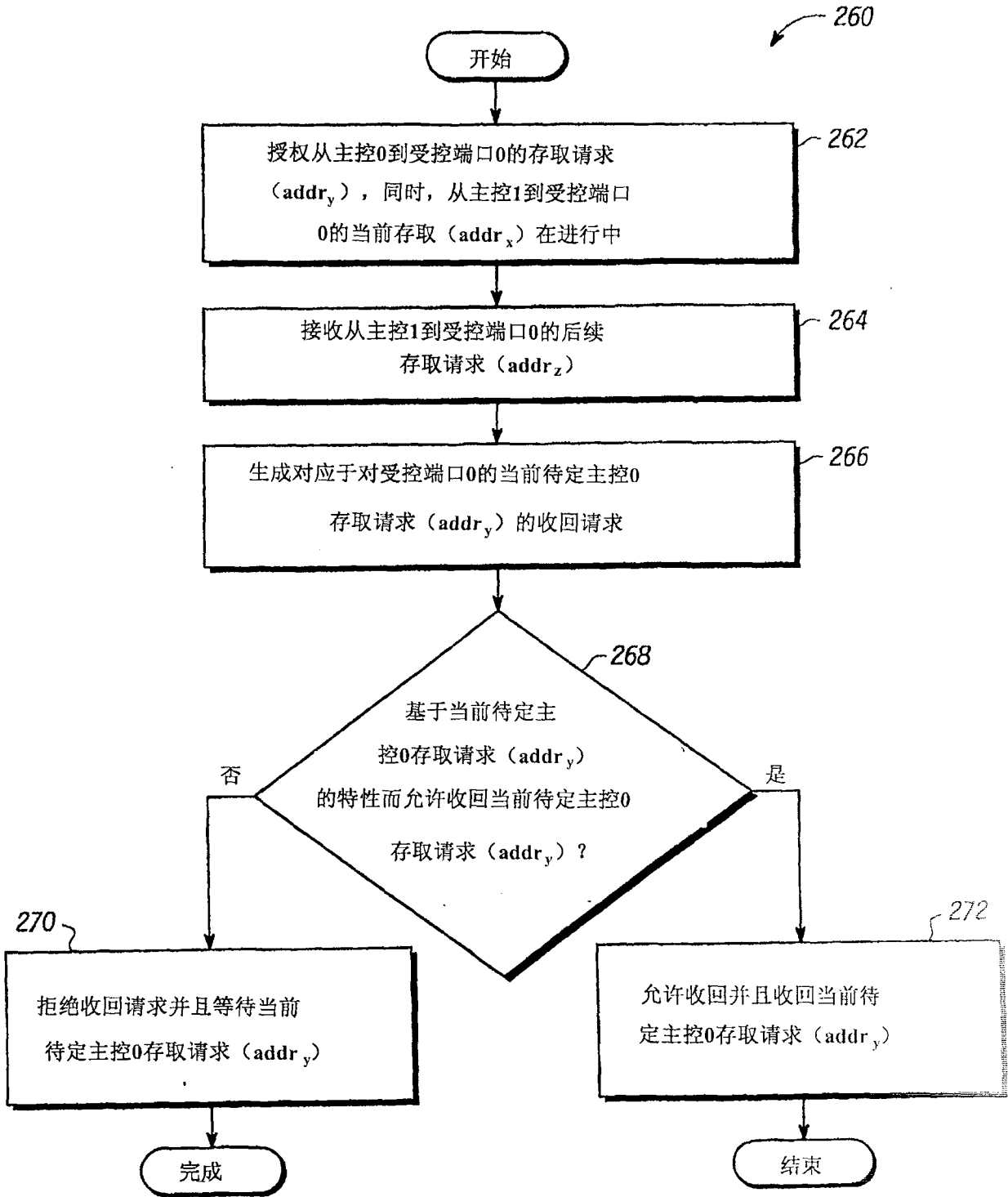


图12