



[12] 发 明 专 利 说 明 书

[21] ZL 专利号 00818155.1

[45] 授权公告日 2005 年 7 月 6 日

[11] 授权公告号 CN 1209706C

[22] 申请日 2000.8.8 [21] 申请号 00818155.1

[30] 优先权

[32] 2000. 1. 3 [33] US [31] 09/476,937

[86] 国际申请 PCT/US2000/021752 2000.8.8

[87] 国际公布 WO2001/050252 英 2001.7.12

[85] 进入国家阶段日期 2002.7.3

[71] 专利权人 先进微装置公司

地址 美国加利福尼亚州

[72] 发明人 J·B·凯勒 T·S·格林

W-H·利恩 R·W·哈达德

审查员 刘静_2

[74] 专利代理机构 北京纪凯知识产权代理有限公司

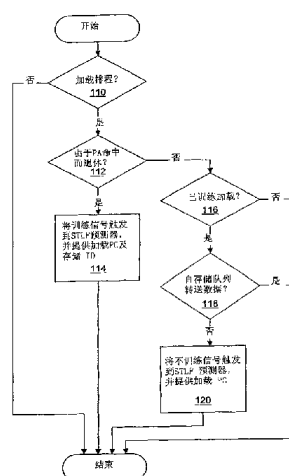
代理人 戈 泊 程 伟

权利要求书 3 页 说明书 36 页 附图 10 页

[54] 发明名称 具有不训练的存储至加载转送预测器

[57] 摘要

一处理器 10 采用一存储至加载转送(STLF)预测器 60, 该 STLF 预测器 60 可指示配发的加载作业对一存储作业之一相依性。针对在一先前执行期间干扰了加载作业的执行之一存储作业, 而指示该相依性。因为指示了与存储作业有关的相依性, 所以不会将加载作业的时程安排在该存储作业之前, 及(或)不会在该存储作业之前先执行加载作业。响应执行一特定的加载作业及存储作业并侦测到该干扰, 而以该特定加载作业及存储作业的信息来训练该 STLF 预测器 60。此外, 如果该 STLF 预测器 60 将一加载作业指示为相依于一特定的存储作业, 且并未实际发生该相依性, 则可以不训练该 STLF 预测器 60(例如, 可删除与一特定加载作业及存储作业有关的信息)。



1. 一种处理器 (10), 包含:

5 存储至加载转送“STLF”预测器 (60), 该 STLF 预测器 (60) 被耦合而接收派发了第一加载内存作业的指示, 其中该 STLF 预测器 (60) 的配置被设定成: 响应该 STLF 预测器 (60) 内存储的且指示在一先前执行期间的第一存储内存作业干扰了该第一加载内存作业的对应于该第一加载内存作业的信息, 而响应该第一加载内存作业, 以便指示该第一加载内存作业对该第一存储内存作业的相依性;

10 该 STLF 预测器包括至少一个表, 该表记录与加载内存作业有关和与干扰加载内存作业的存储内存作业有关的信息, 以及该 STLF 预测器进一步包括一个耦合到至少一个表的电路, 其中该 STLF 预测器的配置被设定成: 以一个相对第一加载内存作业的加载识别码存取至少一个表, 以及其中所述电路被设置成响应至少一个表的一个输出, 指示该
15 第一加载内存作业对该第一存储内存作业的相依性, 而所述一个表的一个输出是响应于以加载识别码进行的存取; 以及

耦合到该 STLF 预测器 (60) 的执行管线 (72), 其中该执行管线 (72) 的配置被设定成: 在该第一存储内存作业响应该相依性之前, 禁止该第一加载内存作业的执行; 且其中该执行管线 (72) 的配置被
20 设定成: 在该第一加载内存作业的执行期间, 侦测并无该相依性; 且其中该执行管线 (72) 的配置被设定成响应该并无该相依性而产生不训练信号;

其中该 STLF 预测器 (60) 被耦合而接收该不训练信号, 且该 STLF 预测器 (60) 的配置被设定成: 更新该 STLF 预测器中存储的信息, 以
25 便不指示在该先前执行期间该第一存储内存作业干扰了该第一加载内存作业。

2. 如权利要求 1 所述的处理器, 其中该信息包含对应于该第一加载内存作业的有效指示, 且其中该 STLF 预测器 (60) 的配置被设定成:
30 将该有效指示置于无效状态, 以便更新该信息。

3. 如权利要求 1 所述的处理器，其中该信息包含对应于该第一加载内存作业的有效指示，且其中该有效指示包含一位向量，且其中如果至少一个位处于有效状态，则该有效指示处于有效状态，且其中该 STLTF 预测器（60）的配置被设定成：将该位向量的第一位置于无效状态，以便更新该信息。

4. 如权利要求 1 所述的处理器，其中如果将该第一加载内存作业的时程安排在该第一存储内存作业的时程之前，且该第一加载内存作业相依赖于该第一存储内存作业，则该第一存储内存作业干扰了该第一加载内存作业。

5. 如权利要求 1 所述的处理器，其中该执行管线（72）包含存储队列（68），且其中该执行管线（72）的配置被设定成：如果在该第一加载内存作业中并未发生自该存储队列（68）转送数据，则侦测到该并无相依性。

6. 一种方法，包含下列步骤：

派发一第一加载内存作业，且响应一 STLTF 预测器中的指示在一先前执行期间的该第一存储内存作业干扰了该第一加载内存作业的对应于该第一加载内存作业的信息，而响应该第一加载内存作业的派发，并指示该第一加载内存作业对该第一存储内存作业的相依性；

禁止将该第一加载内存作业的时程安排在该第一存储内存作业的时程之前；

在该第一加载内存作业的执行期间，侦测到并无该相依性；以及响应该侦测，而更新指示在该先前执行期间该第一存储内存作业干扰了该第一加载内存作业的该信息，以便不指示在该先前执行期间该第一存储内存作业干扰了该第一加载内存作业。

7. 如权利要求 6 所述的方法，其中该信息包含对应于该第一加载内存作业的有效指示，且其中该更新步骤包含下列步骤：将该有效指示置于无效状态。

8. 如权利要求 6 所述的方法，其中该信息包含对应于该第一加载内存作业之一有效指示，且其中该有效指示包含一个位向量，且其中如果该位向量中的至少一个位处于有效状态，则执行该指示，且其中该更新步骤包含下列步骤：选择该位向量的第一位，并将该第一位置于无效状态。

9. 如权利要求 6 所述的方法，其中该侦测步骤包含下列步骤：在该第一加载内存作业中，侦测到并未从存储队列（68）转送数据。

10

10. 一种存储至加载转送（STLF）预测器，该 STLF 预测器的配置被设定成指示一第一加载内存作业对一第一存储内存作业的相依性，其中该 STLF 预测器响应该加载内存作业的指示，而存储对应于该第一加载内存作业的信息，并产生该相依性；该 STLF 预测器包括至少一个表，该表记录与加载内存作业有关和与干扰加载内存作业的存储内存作业有关的信息，以及该 STLF 预测器进一步包括一个耦合到至少一个表的电路，其中该 STLF 预测器的配置被设定成：以一个相对第一加载内存作业的加载识别码存取至少一个表，以及其中所述电路被设置成响应至少一个表的一个输出，指示该第一加载内存作业对该第一存储内存作业的相依性，而所述一个表的一个输出是响应于以加载识别码进行的存取。

具有不训练的存储至加载转送预测器

5 发明背景

技术领域

本发明涉及处理器，尤涉及处理器内的存储至加载转送机制。

背景技术

10 处理器通常包含存储队列，用以缓冲存储已被执行过但仍然处于预测状态的内存作业。可将该等存储内存作业存放在存储队列中，直到该等内存作业退休为止。在退休之后，可将该等存储内存作业提交给高速缓存及（或）内存。在本文的用法中，内存作业是一种指定一内存与一主存储器间之一数据转移的作业（但是也可在高速缓存中完成该数据转移）。加载内存作业指定将数据自内存转移至处理器，而存储内存作业指定将数据自处理器转移至内存。内存作业可以是包含一内存作业的一指令的蕴含部分，或者可以是明确的加载 / 存储指令。在本发明中，可将加载内存作业更简洁地称为“加载”。同样地，可将存储内存作业更简洁地称为“存储”。

20 虽然以预测方式执行存储作业并使该等存储作业在存储队列中等候可提高效能（其方式为自指令执行管线中取出该等存储作业，且可执行其它后续的指令），但是后续的加载作业可能存取该存储队列中的该等存储作业所更新的记忆位置。虽然使各存储作业在该存储队列中等候并不必然直接影响到处理器的效能，但是如果因存取该存储队列中的各存储作业所更新的记忆位置而延迟了后续的加载作业，则可能会影响到效能。存储队列通常被设计成：如果一加载作业命中该存储队列，则转送该存储队列中存储的数据。在本文的用法中，如果一加载内存作业存取一存储内存作业所更新的至少一个字节，则将存储该存储内存作业的一存储队列数据项称为被该加载内存作业“命中”

25

30 (“hit”)。

为了进一步增加效能，最好是以相对于较旧的存储作业为非循序的

方式执行较新的加载作业。较新的加载作业与较旧的存储作业之间通常并无相依性，因而不须等候较旧的存储作业的执行。因为加载作业提供用于执行相依指令的操作数，所以执行该等加载作业时仍可执行其它的指令。然而，如果容许相对于较旧的存储作业为非循序的方式

5 执行较新的加载作业，则在执行加载作业时，只侦测到命中该存储队列可能无法确保正确的程序执行，这是因为可能尚未执行完某些较旧的存储作业（因而在执行该等加载作业时，可能并不知道这些存储作业的存储地址，且可能无法侦测出该等加载作业与该等某些较旧的存储作业间的相依性）。因此，可能需要有用来侦测在一较新的加载作业所相依的一较旧的存储作业之前先执行该较新的加载作业的情形的

10 硬件，然后可响应该侦测，而执行矫正行动。例如，可清除指令并重新提取指令，或者以某一其它适用的方式重新执行指令。在本文的用法中，如果一存储作业更新一加载作业所存取的内存的至少一个字节，而该存储作业比该加载作业旧，且该存储作业比更新该字节的任何其它存储作业新，则该加载作业“相依于”（“dependent” on）该存储作业。

15 但是很不幸，若是不正确地以非循序方式执行加载作业，然后采取后续的矫正行动以获致正确的执行，则此种方式可能会降低效能。

请注意，可将加载、存储、及其它指令作业称为比其它的指令作业旧或新。如果一第一指令在程序顺序（亦即执行程序时指令的顺序）

20 中位于一第二指令之前，则该第一指令比该第二指令旧。如果一第一指令在程序顺序中位于一第二指令之后，则该第一指令比该第二指令新。

发明内容

25 前文所概述的该等问题将大部分为本发明所述的一处理器所解决。该处理器通常可将较新的加载作业之时程安排在较旧的存储作业之前，及(或)可在较旧的存储作业之前先执行较新的加载作业。此外，该处理器可侦测一较旧的存储作业干扰较新的加载作业执行的情形，并针对此种情形而采取矫正行动。该处理器采用一存储至加载转送

30 （StoreToLoadForward；简称 STLF）预测器，该 STLF 预测器可指示配发的加载作业对一存储作业之一相依性。针对在一先前执行期间

干扰了加载作业的执行之一存储作业，而指示该相依性。因为指示了
对该存储作业的相依性，所以不会将加载作业的时程安排在该存储作
业之前，及(或)不会在该存储作业之前先执行加载作业。由于加载与存
储之间有较低的干扰，所以可增加效能。该 STL F 预测器包括至少一个
5 表，该表记录与加载内存作业有关和与相接口于加载内存作业的存储内
存作业有关的信息，以及该 STL F 预测器进一步包括一个耦合到至少一个
表的电路，其中该 STL F 预测器的配置被设定成：以一个相对第一加
载内存作业的加载识别码存取至少一个表，以及其中所述电路被设置成
响应至少一个表的一个输出，指示该第一加载内存作业对该第一存储内
10 存作业的相依性，而所述一个表的一个输出是响应于以加载识别码进行
的存取；以及

耦合到该 STL F 预测器 (60) 的执行管线 (72)，其中该执行管线
(72) 的配置被设定成：在该第一存储内存作业响应该相依性之前，禁
止该第一加载内存作业的执行；且其中该执行管线 (72) 的配置被设定
15 成：在该第一加载内存作业的执行期间，侦测并无该相依性；且其中该
执行管线 (72) 的配置被设定成响应该并无该相依性而产生不训练信号：
其中该 STL F 预测器 (60) 被耦合而接收该不训练信号，且该 STL F 预
测器 (60) 的配置被设定成：更新该 STL F 预测器中存储的信息，以便
不指示在该先前执行期间该第一存储内存作业干扰了该第一加载内存
20 作业。响应执行一特定的加载作业及存储作业并侦测到该干扰，而
以该特定加载作业及存储作业的信息来训练该 STL F 预测器。此外，
如果该 STL F 预测器将一加载作业指示为相依于一特定的存储作业，
且并未实际发生该相依性，则可以不训练该 STL F 预测器(例如，可删
除与一特定加载作业及存储作业有关的信息)。例如，在一实施例中，
25 将加载作业指示为相依于特定的存储作业，但是当执行该加载作业
时，并未自该处理器内的一存储队列转送存储数据，则不训练该 STL F
预测器。

在一实施例中，该 STL F 预测器一加载作业的 PC 所指示的一
第一表中，记录干扰该加载作业的一存储作业的 PC 的至少一部分。
30 一第二表维护最近派发的存储作业的存储 PC 之一对应部分、以及用
来识别该等最近派发的存储作业的标记。一派发的加载作业的 PC 用

来自该第一表中选择一个存储 PC。将所选择的该存储 PC 与该第二表中存储的该等 PC 比较。如果侦测到一相符，则自该第二表读取对应的标记，并将该标记用来指示该加载作业的一相依性。

在另一实施例中，该 STLF 预测器在一加载作业的 PC 所指示的一第一表中，记录指定给该加载作业的标记与干扰该加载作业的一存储作业间的差异。一派发的加载作业的 PC 用来自该表中选择一个差异，并将该差异加入指定给该加载作业的该标记。因此，可产生该存储作业的一标记，且可指示该加载作业对该存储作业之一相依性。

广义而言，所考虑的一处理器包含一 STLF 预测器、及耦合到该 STLF 预测器之一执行管线。该 STLF 预测器被耦合接收派发了一第一加载内存作业之一指示，且该 STLF 预测器的配置被设定成：响应该 STLF 预测器内存储的且指示在一先前执行期间一第一存储内存作业干扰该第一加载内存作业的信息，而指示该第一加载内存作业对该第一存储内存作业之一相依性。该执行管线的配置被设定成：在该第一存储内存作业响应该相依性之前，禁止该第一加载内存作业的执行。该执行管线的配置被设定成：在该第一加载内存作业的执行期间，侦测并无该相依性。该执行管线的配置被设定成响应并无相依性而产生一不训练信号。该 STLF 预测器被耦合接收该不训练信号，且该 STLF 预测器的配置被设定成：更新该 STLF 预测器中存储的信息，以便不指示在该先前执行期间该第一存储内存作业干扰了该第一加载内存作业。此外，所考虑的一计算机系统包含该处理器及一输入 / 输出(I/O)装置，且该 I / O 装置的配置被设定成在该计算机系统与该 I/O 装置可耦合的另一计算机系统之间通信。

此外，也考虑了一种方法。响应指示在一先前执行期间一第一存储内存作业干扰了一第一加载内存作业的信息，而指示该第一加载内存作业对该第一存储内存作业之一相依性。禁止将该第一加载内存作业的时程安排在该第一存储内存作业的时程之前。在该第一加载内存作业的执行期间，侦测到并无该相依性。更新指示在该先前执行期间该第一存储内存作业干扰了该第一加载内存作业的信息，以便不指示在该先前执行期间该第一存储内存作业干扰了该第一加载内存作业。

响应侦测到并无该相依性，而执行该更新。

附图说明

若配合各附图而参照下文的详细说明，将可易于了解本发明的其它日的及优点，这些附图有：

图 1 是一处理器的一实施例的方块图。

图 2 是可用于图 1 所示处理器的一例示管线的管线图。

图 3 是一对映单元、一排程器、一 AGU / TLB、及一加载 / 存储单元的一实施例的详细方块图。

10 图 4 是图 3 所示的一存储至加载转送(STLF)预测器的一实施例的方块图。

图 5 是图 3 所示 STLF 预测器的一第二实施例的方块图。

图 6 是图 4 或 5 所示 STLF 预测器的一实施例中加载的训练及不训练的流程图。

15 图 7 是可用于图 4 或 5 所示 STLF 预测器的一控制电路的一实施例的方块图。

5

图 8 是图 3 所示的一相依性单元的一实施例的方块图。

图 9 是包含图 1 所示处理器的一计算机系统的一实施例的方块图。

10 图 10 是包含图 1 所示处理器的一计算机系统的一第二实施例的方块图。

具体实施方式

处理器概要

15 现在请参阅图 1，图中示出一处理器 10 的一实施例的方块图。其它的实施例也是可行的，也是可考虑的。在图 1 所示的实施例中，处理器 10 包含一指令线预测器 12、一指令高速缓存（I 高速缓存）14、一对准单元 16、一分支预测 / 提取 PC 产生单元 18、多个译码单元 24A - 24D、一预测器未命中译码单元 26、一微码单元 28、一对映单元 20、一退休队列 32、一架构更名文件 34、一未来文件 20、一排程器 36、一整数缓存器文件 38A、一浮点缓存器文件 38B、一整数执行核心 40A、一浮点执行核心 40B、一加载 / 存储单元 42、一数据高速缓存（D 高速缓存）44、一外部接口单元 46、及一 PC 存储区 48。指令线预测器 12 耦合到预测器未命中译码单元 26、分支预测 / 提取 PC 产生单元 18、PC 存储区 48、及对准单元 16。亦可将指令线预测器 12 耦合到 I 高速缓存 14。I 高速缓存 14 耦合到对准单元 16 及分支预测 / 提取 PC 产生单元 18，分支预测 / 提取 PC 产生单元 18 又耦合到 PC 存储区 48。对准单元 16 又耦合到预测器未命中译码单元 26 及译码单元 24A - 24D。译码单元 24A - 24D 又耦合到对映单元 30，且译码单元 24D 耦合到微码单元 28。对映单元 30 耦合到退休队列 32（退休队列 32 耦合到架构更名文件 34、未来文件 20、排程器 36、及 PC 存

20

25

30

器文件 38A - 38B, 缓存器文件 38A - 38B 又相互耦合, 且耦合到各别的执行核心 40A - 40B。执行核心 40A - 40B 又耦合到加载 / 存储单元 42 及排程器 36。执行核心 40A 又耦合到 D 高速缓存 44。加载 / 存储单元 42 耦合到排程器 36、D 高速缓存 44、及外部接口单元 46。D 高速缓存 44 耦合到各缓存器文件 38。外部接口单元 46 耦合到一外部接口 52 及 I 高速缓存 14。将把本文中以一代号加上一字母来表示的各组件整体地只以该代号来表示。例如, 将把译码单元 24A - 24D 整体地表示为译码单元 24。

在图 1 所示的实施例, 处理器 10 采用一种可变位组长度的复杂指令集运算 (Complex Instruction Set Computing; 简称 CISC) 指令集架构。例如, 处理器 10 可采用 x86 指令集架构 (亦称为 IA-32。其它的实施例可采用其中包括固定长度指令集架构及精简指令集运算 (Reduced Instruction Set Computing; 简称 RISC) 指令集架构的其它指令集架构。在此类架构中可省略图 1 所示的某些特征。

分支预测 / 提取 PC 产生单元 18 的配置被设定成可将一提取地址 (提取 PC) 提供给 I 高速缓存 14、指令线预测器 12、及 PC 存储区 48。分支预测 / 提取 PC 产生单元 18 可包含一用来协助产生提取地址的适用的分支预测机制。指令线预测器 12 响应该提取地址, 而将对应于多个指令的对准信息提供给对准单元 16, 并可提供一次一提取地址, 以便提取所提供的指令信息识别的该等指令之后的各指令。视需要的状况, 可将次一提取地址提供给分支预测 / 提取 PC 产生单元 18, 或可将次一提取地址直接提供给 I 高速缓存 14。分支预测 / 提取 PC 产生单元 18 可自 PC 存储区 48 接收一陷阱地址 (在侦测到一陷阱时), 且该陷阱地址可包含分支预测 / 提取 PC 产生单元 18 所产生的提取 PC。否则, 可利用分支预测信息及来自指令线预测器 12 的信息来产生该提取 PC。一般而言, 指令线预测器 12 存储对应于处理器 10 先前以预测方式提取的各指令的信息。在一实施例中, 指令线预测器 12 包含 2000 个数据项, 而每一数据项定位在本文中称为一“闪存线”(“line”)的指令之一组的一个或多个指令。可将该闪存线的指令放入排程器 36, 而由处理器 10 的指令处理管线同时处理该闪存线的指令。

I 高速缓存 14 是一用来存储指令字节的高速缓存。根据一实

施例，I 高速缓存 14 可包含诸如一容量为 128K 字节且采用 64 字节闪存线的四路组关系型组织。然而，任何 I 高速缓存结构也是适用的（其中包括直接对映式结构）。

对准单元 16 自指令线预测器 12 接收指令对准信息，并自 I 高速缓存 14 接收对应于该提取地址的各指令字节。对准单元 16 根据所提供的该指令对准信息，而将各指令字节选入至每一译码单元 24A - 24D。更具体而言，指令线预测器 12 提供一对应于每一译码单元 24A - 24D 的指令指针。该指令指针定位所提取的该等指令字节内之一指令，以便传送到对应的译码单元 24A - 24D。在一实施例中，可将某些指令传送到一个以上的译码单元 24A - 24D。因此，在所示的该实施例中，来自指令线预测器 12 的一指令线最多可包含 4 个指令，但是其它的实施例可包含较多或较少的译码单元 24，以便在一指令线内提供较多或较少的指令。

各译码单元 24A - 24D 将提供给该译码单元的指令译码，且每一译码单元 24A - 24D 产生用来识别对应于该等指令的一个或多个指令作业（或 ROP）的信息。在一实施例中，译码单元 24A - 24D 最多可为每一指令产生两个指令作业。在本文的用法中，指令作业（或 ROP）是一种执行核心 40A - 40B 内的一执行单元的配置被设定成以一单一实体的方式执行的作业。简单的指令可对应于一单一的指令作业，而较复杂的指令可对应于多个指令作业。可将某些较复杂的指令实施为微码单元 28 内的微码例程（在本实施例中，经由译码单元 24D 而自处理器内的一只读存储器提取微码例程）。此外，在其它的实施例中，每一指令可采用一单一的指令作业（亦即，在这些实施例中，指令及指令作业是同义的）。

PC 存储区 48 存储每一指令提取的提取地址及指令信息，并负责在发生异常状况（例如处理器 10 采用的指令集架构所界定的指令陷阱、分支错误预测、或其它微架构界定的陷阱）时使指令提取改向。PC 存储区 48 可包含一循环式缓冲器，用以存储对应于处理器 10 内可能尚未处理的多个指令线的指令的提取地址及指令信息。PC 存储区 48 可响应一指令线的指令的退休，而舍弃对应的数据项。PC 存储区 48 可响应一异常状况，而将一陷阱地址提供给分支预测 / 提取 PC 产生单

元 18。排程器 36 可提供指令退休及异常状况信息。在一实施例中，对映单元 30 将一顺序号码 (R#) 提供给每一指令，以便识别处理器 10 内尚未处理的各指令的顺序。排程器 36 可将 R# 送回到 PC 存储区 48，以便识别碰到异常状况的指令作业、或退休的指令作业。

- 5 对准单元 16 在侦测到指令线预测器 12 中之一未命中时，将来自 I 高速缓存 14 的对应的指令字节传送到预测器未命中译码单元 26。预测器未命中译码单元 26 将该指令译码，而对一指令线的指令强制执行处理器 10 在设计上所加诸的任何限制（例如，最大指令作业数目、最大指令数目、及在分支指令上的终止）。在终止一指令线时，预测器未命中译码单元 26 将该信息提供给指令线预测器 12，以供存储。请注意，
10 可将预测器未命中译码单元 26 的配置设定成在将各指令译码时即派发该等指令。在替代实施例中，预测器未命中译码单元 26 可将该指令线的指令信息译码，并将译码后的指令信息提供给指令线预测器 12 以供存储。然后可在指令线预测器 12 中重新尝试未命中的提取地址，并可
15 侦测到一命中。

- 除了指令线预测器 12 中有一未命中时将指令译码之外，可将预测器未命中译码单元 26 的配置设定成：如果指令线预测器 12 所提供的指令信息是无效的，则将各指令译码。在一实施例中，处理器 10 并未尝试保有指令线预测器 12 中与 I 高速缓存 14 内的指令一致的信息
20 （例如，当各指令在 I 高速缓存 14 中被取代或变成无效时，可以不主动地使对应的指令信息无效）。译码单元 24A - 24D 可验证所提供的该指令信息，且可在侦测到无效的指令信息时，通知预测器未命中译码单元 26。根据一特定实施例，处理器 10 支持下列的指令作业：整数（包括算术、逻辑、移位 / 旋转、及分支运算）、浮点（包括多媒体运算）、
25 以及加载 / 存储。

- 将译码后的指令作业以及来源及目的地缓存器编号提供给对映单元 30。对映单元 30 的配置被设定成：将实体缓存器编号 (PR#) 指定给每一指令作业的每一目的地缓存器操作数及来源缓存器操作数，而执行缓存器更名。该实体缓存器编号识别缓存器文件 38A - 38B 内的
30 各缓存器。对映单元 30 又提供用来更新指定给每一指令作业的一来源操作数的每一实体缓存器编号的各指令作业，而提供该等指令作业的

相依性之一指示。对映单元 30 根据对应的逻辑缓存器编号，而以指定给每一目的地缓存器的实体缓存器编号（及对应的指令作业的 R#）更新未来文件 20。此外，对映单元 30 将目的地缓存器的逻辑缓存器编号、所指定的实体缓存器编号、以及先前指定的实体缓存器编号存储在退休队列 32。当（排程器 36 向对映单元 30 指示）有指令退休时，退休队列 32 更新架构更名文件 34，并空出不再使用的任何缓存器。因此，架构更名文件 34 中的该等实体缓存器编号识别用来存储处理器 10 的被提交的架构状态的该等实体缓存器，而未来文件 20 则呈现处理器 10 的推测状态。换言之，架构更名文件 34 存储对应于每一逻辑缓存器之一实体缓存器编号，而呈现了每一逻辑缓存器的被提交的缓存器状态。未来文件 20 存储对应于每一逻辑缓存器之一实体缓存器编号，而呈现了每一逻辑缓存器的推测式缓存器状态。

根据对映单元 30 所指定的该等 R#，而将该指令线的指令作业、来源实体缓存器编号、及目的地实体缓存器编号存储在排程器 36。此外，可将对一特定指令作业的相依性标注为对该排程器中存储的其它指令作业的相依性。在一实施例中，各指令作业在退休之前都是保留在排程器 36 中。

在满足了针对每一指令作业而标注的相依性之前，排程器 36 存储该指令作业。排程器 36 可响应安排好一特定指令作业的执行时程，而决定该特定指令作业将在哪一时脉周期更新缓存器文件 38A - 38B。执行核心 40A - 40B 内不同的执行单元可采用不同数目的管线阶段（以及因而产生的不同的执行时间）。此外，在一管线内，某些指令可能会有比其它指令长的执行时间。因此，产生一用来量测该特定指令作业的执行时间的计时（以时脉周期数为单位）。排程器 36 等候指定数目的时脉周期（直到在读取缓存器文件的该等相依指令作业之前或同一时间发生更新为止），然后指示：可安排与该特定指令作业相依的各指令作业的时程。请注意，一旦已满足了一指令的相依性，则排程器 36 可安排该指令的时程（亦即，相对于该指令在排程器队列中的顺序为非循序）。

整数及加载 / 存储指令作业根据来源实体缓存器编号，而自缓存器文件 38A 读取来源操作数，并将该等来源操作数传送到执行核心 40A

以供执行。执行核心 40A 执行该指令作业，并更新指定给缓存器文件 38A 内的目的地的实体缓存器。此外，执行核心 40A 将干指令作业读 R# 以及与该指令作业有关的异常状况信息（如果有任何异常状况）回饱给排程器 36。缓存器文件 38B 及执行核心 40B 可以一类似的方式对
5 浮点指令作业进行作业（并可将于浮点存储的存储数据提供给加载 / 存储单元 42）。

在一实施例中，执行核心 40A 可包含诸如两个整数单元、一支单元、及两个地址产生单元（具有对应的转换后援缓冲区（Translation Look-aside Buffer; 简称 TLB））。执行核心 40B 可包含一浮点 / 多
10 媒体乘法器、一浮点 / 多媒体加法器、及一用来将存储数据传送到加载 / 存储单元 42 的存储数据单元。执行单元的其它配置也是可行的。

加载 / 存储单元 42 提供了 D 高速缓存 44 的一接口，用以执行内存作业，并安排未命中 D 高速缓存 44 的各内存作业的填满作业。执行核心 40A 执行一地址产生，并将数据（自 D 高速缓存 44 或加载 /
15 存储单元 42 内的一存储队列）转送到缓存器文件 38A - 38B，而完成加载内存作业。在执行核心 40A 产生存储地址时，可将该等存储地址（直接经由执行核心 40A 与 D 高速缓存 44 间的联机）提供给 D 高速缓存 44。将一存储队列数据项分配给该等存储地址。根据设计上的选择，可同时提供存储数据，或者可随后提供存储数据。在存储指令
20 退休时，将数据存储在 D 高速缓存 44（但是指令退休与更新 D 高速缓存 44 之间可能有某一延迟。此外，加载 / 存储单元 42 可包含一加载 / 存储缓冲器，用以存储未命中 D 高速缓存 44 的加载 / 存储地址，以供后续的高速缓存填满（经由外部接口单元 46），并重新尝试该未命中加载 / 存储作业。加载 / 存储单元 42 的配置进一步被设定成处理
25 加载 / 存储内存的相依性。

D 高速缓存 44 是一用来存储处理器 10 所存取的数据的高速缓存。虽然 D 高速缓存 44 可包含任何适用的结构（其中包括直接对映式结构及组关系型结构），但是 D 高速缓存 44 之一实施例可包含一容量为 128 K 字节且具有 64 字节闪存线的 2 路组关系型高速缓存。

30 外部接口单元 46 的配置被设定成经由外部接口 52 而与其它装置通信。可使用任何适当的外部接口 52，其中包括 L2 高速缓存的接口、

以及用来将处理器 10 连接到其它装置的一个或多个外部总线。外部接口单元 46 提取用于 I 高速缓存 14 及 D 高速缓存 44 的填满数据, 并将舍弃的更新后闪存线自 D 高速缓存 44 写到该外部接口。此外, 外部接口单元 46 亦可执行处理器 10 所产的不可存储在高速缓存的读取及写入。

然后请参阅第 2 图, 图中示出处理器 10 的一实施例可采用的一组例示管线阶段之一例示管线图。其它的实施例可采用不同的管线, 这些管线包含与图 2 所示管线不同的较多或较少的管线阶段。以垂直线划定图 2 所示的该等阶段。每一阶段是用来同步处理器 10 内的各存储组件 (例如缓存器、锁存器、及正反器等) 的一时脉信号之一时脉周期。

如图 2 所示, 该例示的管线包含一 CAM0 阶段、一 CAM1 阶段、一指令线预测器 (Line Predictor; 简称 LP) 阶段、一指令高速缓存 (Instruction Cache; 简称 IC) 阶段、一对准 (ALignment; 简称 AL) 阶段、一译码 (DECode; 简称 DEC) 阶段、一对映 1 (M1 阶段、一对映 2 (M2 阶段、一写入排程器 (WR SC) 阶段、一读取排程器 (RD SC) 阶段、一缓存器文件读取 (RF RD) 阶段、一执行 (EX) 阶段、一缓存器文件读取 (RF WR) 阶段、及一退休 (RET) 阶段。某些指令在执行状态中采用多个时脉周期。例如, 图 2 中以分解图的形式示出内存作业、浮点运算、及整数乘法运算。内存作业包含一地址产生 (AGU) 阶段、一转换 (TLB) 阶段、一数据高速缓存 1 (DC1 阶段、及一数据高速缓存 2 (DC2 阶段。同样地, 浮点运算最多可包含四个浮点执行 (FEX1 - FEX4 阶段, 且整数乘法运算最多可包含四个 (IM1 - IM4 阶段。

在 CAM0 及 CAM1 阶段中, 指令线预测器 12 将分支预测 / 提取 PC 产生单元 18 所提供的提取地址与指令线预测器 12 中存储的指令线的地址比较。此外, 在 CAM0 及 CAM1 阶段中, 将该提取地址自一虚拟地址 (例如 x86 架构中的一线性地址转换为一物理地址。响应在 CAM0 及 CAM1 阶段中侦测到一命中, 在该指令线预测器阶段中自该指令线预测器读取对应的指令线信息。此外, I 高速缓存 14 在该指令线预测器阶段中开始一读取 (利用该物理地址)。在该指令高速缓

存阶段中完成了该读取。

请注意,虽然图2所示的管线采用两个时脉周期而使指令线预测器12命中一提取地址,但是其它的实施例采用一单一时脉周期(及阶段)来执行该作业。此外,在一实施例中,指令线预测器12将
5 I 高速缓存14的一次一提取地址及指令线预测器12中的次一数据项提供给一命中,因而肇因于一先前命中指令线预测器12的各提取可略过CAM0及CAM1阶段。

对准单元16在对准阶段中响应来自指令线预测器12的对应的指令线信息,而将I 高速缓存14所提供的指令字节对准译码单元24A -
10 24D。在该译码阶段中,译码单元24A - 24D将所提供的该等指令译码,而识别出对应于该等指令的ROP、以及操作数信息。对映单元30利用M1阶段中所提供的信息产生各ROP,并执行缓存器更名(更新未来文件20。在M2阶段中,将该等ROP及指定的更名记录在退休队列32。此外,决定每一ROP所相依的各ROP。每一ROP可以是该
15 未来文件中记录的相依于先前各ROP的缓存器作业,且亦可表现出其它类型的相依性(例如,依存于一先前串行化指令的相依性等)。

在写入排程器阶段中,将所产生的该等ROP写入排程器36。直到该阶段之前,一特定指令线的信息所定位的该等ROP都是以整体的方式流经该管线。然而,在将该等ROP写入排程器36之后,该等
20 ROP可在不同的时间中独立地流经其余的阶段。一般而言,一特定的ROP停留在该阶段,直到排程器36选择执行该ROP为止(例如,如前文所述,在已选择执行该特定ROP所相依的各ROP之后)。因此,一特定的ROP可能会在该写入排程器阶段与该读取排程器阶段之间有一个或多个时脉周期的延迟。在该读取排程器阶段中,该特
25 定的ROP参与排程器36内的选择逻辑,选择执行该特定的ROP,并自排程器36读取该特定的ROP。在缓存器文件读取阶段中,该特定的ROP然后继续进行自缓存器文件38A - 38B中之一个缓存器文件(取决于ROP的类型)的读取缓存器文件作业。

将该特定的ROP及操作数提供给对应的执行核心40A或40B,
30 且在执行阶段中对该等操作数执行该等指令作业。如前文所述,某些ROP有数个执行管线阶段。例如,经由一地址产生阶段(在该阶段中,

产生内存指令作业所存取的记忆位置的数据地址)、一转换阶段(在该阶段中,转换该地址产生阶段所提供的虚拟数据地址)、及用来存取 D 高速缓存 44 的一对数据高速缓存阶段,而执行内存指令作业(例如加载及存储)。浮点运算最多可采用 4 个执行时脉周期,且整数乘法运算可类似第最多可采用 4 个执行时脉周期。

在完成该等一个或多个执行阶段时,该特定的 ROP 在缓存器文件写入阶段中更新指定给该特定的 ROP 的实体缓存器。最后,在每一先前的 ROP 退休之后,该特定的 ROP 退休(在退休阶段)。一特定的 ROP 在该缓存器文件写入阶段与该退休阶段之间仍然可能要经过一个或多个时脉周期。此外,如此项技艺中习知的,一特定的 ROP 可能因管线停顿状况而停顿在任何阶段。

存储至加载转送

现在请参阅图 3,图中示出对映单元 30、排程器 36、一地址产生单元 / 转换后援缓冲区(Address Generation Unit/Translation Lookaside Buffer; 简称 AGU/TLB) 40AA 的一实施例的详细方块图。在图 3 所示的实施例中,排程器 36、AGU/TLB 40AA、及加载 / 存储单元 42 合而称为执行管线 72。对映单元 30 包含一存储至加载转送(STLF)预测器 60、一相依性单元 62、及一 R# 指定单元 64(用以将 R# 指定给指令作业)。排程器 36 包含一排程缓冲器 66 及一物理地址(Physical Address; 简称 PA)缓冲器 70。加载 / 存储单元 42 包含一存储队列 68。对映单元 30 被耦合,而自译码单元 24 接收指令作业及对应的程序计数器地址(Program Counter Address; 简称 PC),自排程器 36 接收一退休信号,且自执行管线 72 接收一训练 / 不训练接口(其中包括训练 / 不训练(T/UT)信号、一加载 PC(LPC)、及一存储识别码(SID))。对映单元 30 被耦合而将该等指令作业、PC、及相依性信息提供给排程器 36。更具体而言,STLF 预测器 60 被耦合而接收该等指令作业、PC、来自 R# 指定单元 64 的 R#、及训练 / 不训练接口,并将与对存储作业有关的加载相依性有关的信息相依性单元 62,而相依性单元 62 亦被耦合而接收该等指令作业及来自 R# 指定单元 64 的 R#。相依性单元 62 被耦合而将该相依性信息提供给排程器

36. 排程器 36 被耦合而将加载及存储作业以及对应的控制信息提供给 AGU/TLB 40AA。AGU/TLB 40AA 被耦合，而自缓存器文件 38A 接收对应的操作数，并将一物理地址及其它的控制信息连同（在一存储作业的情形中的）存储数据提供给存储队列 68。此外，AGU/TLB 40AA 5 被耦合而将该物理地址及控制信息提供给 PA 缓冲器 70，而 PA 缓冲器 70 耦合到排程缓冲器 66。存储队列 68 被耦合，而自排程器 36 接收一退休信号，并将一存储至提交作业及存储转送数据提供给 D 高速缓存 44。在一实施例中，AGU/TLB 40AA 是整数执行核心 40A 的一部分。

一般而言，对映单元 30 自译码单元 24 接收指令作业。对于每一加载内存作业而言，STLF 预测器 60 决定该 STLF 预测器是否有任何信息指示：在一先前执行期间，一存储内存作业干扰了该加载内存作业。如果一存储内存作业的确干扰了，则 STLF 预测器 60 将该存储内存作业的一指示提供给相依性单元 62。相依性单元 62 指示加载内存作业对该存储内存作业之一相依性（以及对地址操作数等的任何相依性），15 且因而该加载内存作业的时程不得安排在该存储内存作业的时程之前。因此，在该加载内存作业的现行执行期间，该存储内存作业不会干扰到。另一方面，如果 STLF 预测器 60 并未记录与一存储内存作业对一特定加载内存作业干扰有关的任何信息，则 STLF 预测器 60 并不向相依性单元 62 指示一相依性。该特定的加载内存作业可能对来源缓存器操作数有相依性，但不会对任何存储内存作业有相依性。20

在本文的用法中，如果一存储内存作业使额外的时脉周期加入一加载内存作业的执行，则该存储内存作业“干扰了”该加载内存作业。可能以管线停顿的方式加入该等额外的时脉周期，或者可能经由重新执行该加载内存作业而加入该等额外的时脉周期。本发明揭示事项的其余部分的焦点将集中在一实施例，而在该实施例中，如果一存储内存作业比一加载内存作业旧，则该存储内存作业干扰了该加载内存作业，该加载内存作业对该存储内存作业具有一相依性，且将该加载内存作业的时程安排在该存储内存作业的时程之前，及（或）执行该加载内存作业系在执行该存储内存作业之前。也可考虑其它的实施例。25 例如，可考虑一实施例，在该实施例中，并不将加载内存作业的时程安排在一存储内存作业的地址产生的时程之前，但是可将该加载内存

作业的时程安排在提供存储数据的时程之前。在该实施例中，如果有一相依性，且在执行该加载内存作业时，无法取得存储数据，则该存储作业将干扰到该加载作业。

对映单元 30 将该等指令作业、PC、及相依性传送到排程器 36，排程器 36 将该等指令作业写入排程缓冲器 66。排程缓冲器 66 包含多个数据项，每一数据项可存储与一指令作业有关的信息。排程缓冲器 66 中所示之一例示内存作业数据项包含一有效位及一类型字段（用以将该数据项识别为存储一内存作业，并识别存储了哪一类的内存作业（加载或存储））。对于加载内存作业而言，存储该加载内存作业的 PC（LPC）。也可存储额外的信息，以便协助以与干扰该加载内存作业的一存储内存作业有关的信息来训练 STLF 预测器 60。例如，可包含一存储识别码（SID）字段，用以存储在线现行执行期间已干扰了该加载内存作业的该存储内存作业之一指示、以及用来指示已重新尝试了该加载内存作业（由于被该存储内存作业干扰）且因而重新安排了重新执行该加载内存作业的时程之一重新尝试指示（R）。亦存储一训练指示（T），以便指示：STLF 预测器 60 在将该加载作业派发到排程器 36 时，侦测到该加载作业相依于一旧的存储作业。在一实施例中，该重新尝试指示可以是一位，而当该位被设定时，即指示重新尝试。同样地，该训练指示可以是一位，而当该位被设定时，即指示侦测到相依性。在其它的实施例中，可使用相反的方式，且可使用其它的编码。此外，可视需要而存储额外的信息（例如，容量信息及操作数 PR# 等），且其它类型的数据项（例如整数及浮点等）可以有不同的格式。在满足了所记录的每一相依性（其中包括 STLF 预测器 60 所识别的任何相依性）之后，排程器 36 即安排该内存作业的时程以供执行，并将该作业的加载 / 存储本质及其它的控制信息传送到 AGU/TLB 40AA。

更具体而言，因为一旦满足了排程器 36 所记录的每一相依性之后，排程器 36 即可安排一内存作业的时程以供执行，所以如果 STLF 预测器 60 并未指示较新的加载作业对较旧的存储作业有相依性，则可将各较新的加载作业的时程安排在较旧的存储作业的时程之前。对映单元 30 可侦测每一来源缓存器操作数的相依性，但是可能无法侦测所有的加载作业对较早的存储作业的相依性。一加载作业对一存储作业的相

依性系基于分别受到该加载作业及该存储作业影响的内存地址，且在该加载作业及存储作业的执行期间自该加载作业及存储作业的来源操作数而产生该相依性。STLF 预测器 60 侦测到加载作业对存储作业的某些相依性（如本文的说明），但是可能无法侦测到其它的相依性。

- 5 因此，处理器 10 采用将于下文中说明的 PA 缓冲器 70，以便侦测时程被安排在一较旧存储作业的时程之前的一较新的存储作业相依于该较旧的存储作业的情形。

AGU/TLB 40AA 接收该等内存作业及（响应来自排程器 36 的 PR# 而自缓存器文件 38A 读取的）操作数。AGU/TLB 40AA 加入该等操作数，以便产生一虚拟地址，并利用该 TLB 中缓冲存储的转换信息，而将该虚拟地址转换为一物理地址。AGU/TLB 40AA 将该物理地址及其它的控制信息提供给存储队列 68。如果该作业是一存储作业，则亦提供存储数据。AGU/TLB 40AA 所提供的该控制信息可以是该作业的加载或存储本质。AGU/TLB 40AA 亦将该物理地址及其它的控制信息
15 提供给 D 高速缓存 44 及 PA 缓冲器 70。

在本发明中，利用 PA 缓冲器 70 来侦测干扰了加载作业的存储作业。PA 缓冲器 70 包含多个数据项，而排程缓冲器 66 中的每一数据项有一个数据项。每一数据项可存储物理地址信息。当执行一加载作业时，即将提供给 PA 缓冲器 70 的该物理地址存储到对应于用来存储该加载作业的排程缓冲器数据项的一数据项。另一方面，当执行一存储
20 作业时，即将该物理地址与 PA 缓冲器 70 中存储的物理地址比较。如果发现有一相符，且对应的指令作业是比该存储作业新的一加载作业，则重新尝试该加载作业。一般而言，如果排程器 36 内的一内存作业的状态被复位为一未执行的状态，则在本发明中将该内存作业称为“被重新尝试”。重新尝试该内存作业时，随即将造成该内存作业被重新安
25 排时程且被重新执行。

如果在 PA 缓冲器 70 中侦测到一重新尝试的状况，则设定对应的排程缓冲器数据项中的重新尝试指示。此外，将 STLF 预测器 60 用来识别该存储作业的存储识别码存储在该排程缓冲器数据项的 SID 字段。在一实施例中，该存储 ID 可以是存储 PC。在另一实施例中，该
30 存储 ID 可以是该存储作业的 R#、或该存储作业的 R# 与该加载作

业的 R# 间的差异（差异 R#）。下文中将详述使用每一存储 ID 的实施例。然后当重新安排该加载作业的时程并重新执行该加载作业时，被设定的该重新尝试指示将使执行管线 72 利用该训练 / 不训练接口而在 STL F 预测器 60 中训练该加载作业及对应的存储作业（因而，后续

5 的执行可使该加载作业相依赖于该存储作业，而避免重新尝试该加载作业）。更具体而言，可触发该接口内的一训练信号，并将来自对应的排程器数据项的该加载 PC 及该存储 ID 提供给 STL F 预测器 60。请注意，可根据设计上的选择，而使该训练发生在执行管线 72 的任何阶段。

10 藉由指示该加载作业对在该加载作业的一先前执行期间干扰了该加载作业的存储作业之一相依性，可在安排该存储作业的时程之前，不安排该加载作业的时程。在此种方式下，可在第一次执行该加载内存作业期间侦测到该加载作业对该存储作业的相依性，并可响应该加载内存作业而转送存储数据。因此，可避免该加载作业的重新安排时程及重新执行。由于加载作业不需要重新安排时程及重新执行，所以

15 可提高效能。

另一方面，如果将该加载作业训练成相依赖于存储作业，且在一执行期间并无实际的相依性，则可能因延迟安排该加载作业的时程，而失掉一些效能。因此，执行管线 72 亦可侦测这些状况，并利用该训练

20 / 不训练接口来停止训练来自 STL F 预测器 60 的该加载作业及对应的存储作业。更具体而言，如果安排了一加载作业的时程，且在排程缓冲器 66 中设定了该加载作业的训练指示，则执行管线 72 决定该加载作业是否自存储队列 68 接收了所转送的数据。如果并未发生任何转送，则可能尚未证明该加载作业对一存储作业之一相依性。因此，执行

25 管线 72 可触发一不训练信号，并将该加载 PC 提供给 STL F 预测器 60。STL F 预测器 60 可不训练对应的该加载作业的信息。

在本文的用法中，术语“训练”（“train”）意指存储用来识别发生干扰了一加载作业的一存储作业的信息，且可包括更新用来指示该干扰再度发生的可能性（例如，如果过去重复地发生该状况，则很有可能

30 再度发生此种状况）的信息。因此，训练可包括在该加载作业与该存储作业之间产生一较强的关联性。术语“不训练”（“untrain”）意指删

除用来识别发生干扰了一加载作业的一存储作业的信息，且可包括在删除该信息之前在该加载作业与该存储作业之间产生一较弱的关联性。

5 请注意，STLF 预测器 60 的该训练及不训练可能发生在任何的管线阶段，且可在与不训练不同的一阶段中执行训练。例如，在本实施例中，在重新安排加载作业的时程时，响应重新尝试指示而执行训练，因而可在图 2 所示的排程器读取阶段之后的任何阶段中执行训练。响应训练指示及加载作业并无存储转送，而执行不训练，因而不训练可在管线的稍后阶段中发生（例如，图 2 所示的 DC2 阶段）。

10 现在回到内存作业的执行，如果该内存作业是一存储作业，则存储队列 68 存储 AGU/TLB 40AA 所提供的信息。另一方面，如果该内存作业是一加载作业，则存储队列 68 将加载信息与存储队列数据项中的信息比较。如果侦测到命中一存储队列数据项，则读取对应的存储队列数据，并将该数据提供给 D 高速缓存 44，以供转送（图 3 所示的
15 存储转送数据）。

存储队列 68 至少将存储作业保存到排程器 36 使该等存储作业退休为止。排程器 36 经由退休信号通知存储队列 68，以便指示一个或多个存储作业的退休。存储队列 68 利用存储作业提交路径将该等退休的存储作业循序传送到 D 高速缓存 44。因此，各存储作业苦在存储队列
20 68 中保留到将该等存储作业提交给 D 高速缓存 44 为止，而可能因该等存储作业的退休而延迟该提交。

请注意，AGU 及 TLB 的各种组合都是可行的。例如，在一实施例中，可考虑一加载 AGU 及一独立的存储 AGU。可将该存储 AGU 耦合到存储队列 68 上的一写入埠，且可将该 加载 AGU 耦合到存储
25 队列 68 上的一比较埠。其它的实施例可视需要而包含用于加载、存储、或加载及存储的任何数目的 AGU。

请注意，虽然解说了图 3 所示各单元的某些细节，但是为了简化说明，而省略了与侦测命中存储队列的加载作业无关的其它细节及特征。例如，如前文中参照图 1 所述的，对映单元 30 可执行缓存器更名。

30 在一实施例中，在加载作业对较早的存储作业有指示的相依性时，STLF 预测器 60 在图 2 所示管线的 map2 阶段中作业。然而，根据各

种实施例，STLF 预测器 60 可在选择加载作业以供执行之前的任何管线阶段中作业。

前文的说明描述在重新执行加载作业期间的训练。然而，替代实施例可在不同的时间执行该训练。例如，一替代实施例可响应侦测到新尝试的状况（例如，在执行该加载作业所相依的存储作业时）而执行训练。

已在 STLF 预测器 60 中训练及不训练加载作业及存储作业的环境中说明了各加载作业（以及在一实施例中的各存储作业）的 PC。然而，请注意，在某些实施例中，只可将 PC 的一部分用来训练及不训练 STLF 预测器 60 中的加载作业及存储作业。例如，如将于下文中说明的，在一实施例中，可将加载 PC 及存储 PC 的 10 个最低有效位用于 STLF 预测器 60 中。

在本文的用法中，首字母缩略字“PC”用来表示一指令的程序计数器地址。PC 是指令在内存中的地址。换言之，PC 是用来自内存提取指令的地址。在多个指令作业可对应于一指令的实施例中，指令的 PC 也是每一指令作业（例如加载及存储内存作业）的 PC。

请注意，在前文所述的或将于后文中述及的某些实施例中，R# 用来识别指令作业。一般而言，可使用任何适当的标记。R# 识别指令作业的相对程序顺序，且可识别排程缓冲器 66 中指定给指令作业的数据项。其它的实施例可采用重新排序缓冲区（reorder buffer）标记或任何其它的标记来识别指令作业。此外，可在 STLF 预测器 60 的作业的前或同时的处理器 10 管线中的任何时点指定各 R# 或标记。

现在请参阅图 4，图中示出 STLF 预测器 60 的一第一实施例（STLF 预测器 60a）的方块图。STLF 预测器 60a 包含一加载 / 存储相依性表 80、一存储 PC/R# 表 82、一加载 / 存储相依性表（Load/Store Dependency Table；简称 LSDT）控制电路 84、一 ST/LD 相依性电路 86、一存储表控制电路 88、指令线内相依性检查电路 90、及一多任务器（multiplexor；简称 mux）92。加载 / 存储相依性表 80 被耦合而自译码单元 24 接收派发指令作业的 PC，且加载 / 存储相依性表 80 耦合到 LSDT 控制电路 84。此外，加载 / 存储相依性表 80 被耦合，而自执行管线 72 接收一加载 PC 及存储 PC，以便训练。加载 / 存储相依

性表 80 被耦合，而将存储 PC 提供给指令线内相依性检查电路 90 及存储 PC/R# 表 82，并将有效指示提供给 ST/LD 相依性电路 86。指令线内相依性检查电路 90 被耦合，而自译码单元 24 接收派发的指令作业的 PC、及每一指令作业的加载或存储本质之一指示，且被耦合而将命中信号提供给 ST/LD 相依性电路 86，并将一选择控制信号提供给 mux92。存储 PC/R# 表 82 被耦合，而接收派发的指令作业之 PC、及指定给该等指令作业的对应的 R#。存储 PC/R# 表 82 被耦合，而将存储 R# 提供给 mux92，mux92 又被耦合，而接收指定给派发的指令作业的 R#，并将存储 R# 提供给相依性单元 62。存储 PC/R# 表 82 被耦合而将命中信号提供给存储表控制电路 88，且被耦合而自存储表控制电路 88 接收控制信息。ST/LD 相依性电路 86 耦合到存储表控制电路 88，且被耦合而将相依性有效及相依所有信号提供给相依性单元 62。LSDT 控制电路 84 被耦合，而自执行管线 72 接收训练 / 不训练信号。

一般而言，一加载 PC 作为加载 / 存储相依性表 80 的索引，以便从多个数据项中选出一个数据项。该数据项存储一有效指示、以及在一先前执行期间可能干扰到该加载作业的一存储作业之一存储 PC (图 4 所示的 SPC)。该存储 PC/R# 表包含多个数据项，用以存储最近派发的存储作业的存储 PC、以及该存储作业的对应的 R#。如果加载 / 存储相依性表 80 中所选择数据项的存储 PC 命中了存储 PC/R# 表 82，则位该加载作业标注该加载作业对该存储作业的一相依性。在此种方式下，不会将该加载作业的时程 (及执行) 安排在该存储作业之前。因此，可避免在现行执行期间中的干扰。

更具体而言，当派发各指令作业时，利用该等指令作业的 PC 作为加载 / 存储相依性表 80 的索引。除非另有说明，否则本说明的其余部分的焦点将集中在 STLTF 预测器 60a 对与一派发的指令作业对应的一输入 PC 的响应。然而，请注意，STLTF 预测器 60a 可平行于每一派发的指令作业的每一 PC，而作出响应。加载 / 存储相依性表 80 响应该输入 PC，而输出一有效指示、及来自该被指向的数据项之一存储 PC。该存储 PC 输入到存储 PC/R# 表 82，并将该存储 PC 与存储 PC/R# 表 82 中存储的存储 PC 比较。例如，存储 PC/R# 表 82 可包

含一内容可寻址内存 (Content Addressable Memory; 简称 CAM)。存储 PC/R# 表 82 输出每一数据项的命中信号, 用以指示该存储 PC 是否命中该数据项。存储表控制电路 88 接收该等命中信号, 并选择该存储 PC 所命中的存储 PC/R# 表 82 中表示的最新的存储作业。所选择的该数据项将一存储 R# 输出到 mux92, 而 mux92 通常选择该存储 R#, 以便输出到相依性单元 62。

此外, 存储表控制电路 88 以是否在存储 PC/R# 表 82 中侦测到命中该指令作业之一指示通知 ST/LD 相依性电路 86。ST/LD 相依性电路 86 将一相依性有效信号提供给相依性单元 62。该相依性有效信号如果被触发, 则指示相依性单元 62 将要记录该指令作业对 mux92 所提供的存储 R# 识别的存储作业的一相依性。如果停止触发该相依性有效信号, 则该信号指示相依性单元 62 将不记录任何相依性。更具体而言, 在一实施例中, 如果符合下列的条件, 则 ST/LD 相依性电路 86 可触发该相依性有效信号, 这些条件为: (i) 该指令作业是一加载作业 (由来自译码单元 24 的加载 / 存储指示所决定); (ii) 来自加载 / 存储相依性表 80 被指向的数据项的有效指示指出是有效的; 以及 (iii) 来自被指向的数据项的存储 PC 命中存储 PC/R# 表 82。

图 1 所示的处理器 10 尝试在每一时脉周期中派发多个指令作业。因此, 在派发与加载 / 存储相依性表 80 所提供的存储 PC 符合的最新的存储作业时, 可能同时派发对应的加载作业。因此, 对于一特定的指令作业而言, 指令线内相依性检查电路 90 将加载 / 存储相依性表 80 所输出的存储 PC 与程序顺序在该特定的指令作业之前的每一同时派发的指令作业的 PC 比较。如果该先前指令作业的 PC 符合来自加载 / 存储相依性表 80 的存储 PC, 且该先前指令作业是一存储作业 (由译码单元 24 提供的加载 / 存储指示所指示), 则指令线内相依性检查电路 90 可执行下列步骤: (i) 向 ST/LD 相依性电路 86 指示命中了该对应的加载作业; 以及 (ii) 控制 mux92, 以使用侦测到被命中的该指令作业的 R# 取代存储 PC/R# 表 82 所提供的存储 R#。在此种方式下, 输出到相依性单元 62 的存储 R# 即是与该加载作业同时被派发的存储作业的 R#。此外, 如果符合下列的条件, 则 ST/LD 相依性电路 86 可触发该加载作业的该相依性有效信号, 这些条件为: (i) 该指

令作业是一加载作业（由来自译码单元 24 的加载 / 存储指示所决定）；

（ii）来自加载 / 存储相依性表 80 中被指向的数据项的有效指示指出是有效的；以及（iii）针对该加载作业而触发了来自指令线内相依性检查电路 90 的命中信号。

- 5 在一实施例中，ST/LD 相依性电路 86 可进一步触发该指令作业的相依所有信号。该相依所有信号如果被触发，则向相依性单元 62 指示要记录该指令作业对每一待处理的（已配发且未退休的）存储作业的相依性。该相依所有信号用来处理以干扰到加载作业的各存储作业的存储 PC 重复地训练一特定数据项的情形。因为加载 / 存储相依性表
- 10 80 响应一指令作业的 PC 而选择一数据项，且该数据项存储一个存储 PC，所以纵使 STLF 预测器 60a 指示对一存储作业的一相依性，仍然可能会干扰到不同的存储作业在不同的执行中会干扰到的各加载作业。为了以更佳的方式处理这些情形，加载 / 存储相依性表 80 中的该有效指示可以是一位向量。每当由执行管线 72 训练一数据项时，可将
- 15 该位向量中的一位置于有效状态（例如，根据设计上的选择而设定或清除该位）。如果每一该等位都处于有效状态，则可重复地训练该数据项，这是因为不同的存储作业在各次执行中干扰了该加载作业。因此，如果（i）该位向量中的每一位都处于有效状态；以及（ii）该指令作业是一加载作业，则可触发该相依所有信号。下文中将详细说明该
- 20 位向量的一实施例、以及将各位置于有效状态或无效状态的情形。

- 前文已说明了利用 STLF 预测器 60a 来指示加载作业对在先前执行期间可能干扰到加载作业的存储作业的相依性。此外，可以与加载作业及存储作业有关的信息训练 STLF 预测器 60a。存储 PC/R# 表
- 25 82 存储最近被派发的存储作业的存储 PC 及 R#。因此，存储表控制电路 88 可将存储 PC/R# 表 82 中的各数据项分配给被派发的各存储作业。存储表控制电路 88 自译码单元 24 接收每一指令作业的加载 / 存储指示，并将各数据项分配给每一派发的存储作业。以（自译码单元 24 接收的）存储 PC 及（自 R# 指定单元 64 接收的）对应的 R# 来更新所分配的该等数据项。在一实施例中，存储表控制电路 88 可将存
- 30 储 PC/R# 表 82 操作为最近被派发的存储作业的一先进先出（First-In, First-Out；简称 FIFO）缓冲器。因此，一旦以存储作业填满了存储

PC/R# 表之后, 后续派发的存储作业即取代了存储 PC/R# 表 82 内最旧的存储作业。此外, 一存储作业有可能在存储 PC/R# 表 82 中被后续派发的存储作业删除之前即已退休。因此, 存储表控制电路 88 可接收退休的存储作业的 R#, 并可删除具有对应的 R# 的数据项。

5 响应来自执行管线 72 的训练 / 不训练接口, 而训练加载 / 存储相依性表 80。更具体而言, 如果执行管线 72 触发训练信号, 则 LSDT 控制电路 84 更新加载 / 存储相依性表 80。执行管线 72 将作为输入的将要被训练的加载作业的 PC (图 4 所示的 LPC) 及干扰该加载作业的对应的存储 PC 提供给加载 / 存储相依性表 80。加载 / 存储相依性表
10 80 以该存储 PC 更新该加载 PC 所指向的数据项, 且 LSDT 控制电路 84 将该有效指示置于一有效状态。在一实施例中, 该有效指示可以是一位, 且该有效状态可以是设定 (或清除), 而无效状态可以是清除 (或设定)。在前文所述的另一实施例中, 该有效指示可以是一位向量。在该实施例中, LSDT 控制电路 84 可选择该位向量内的一位,
15 并在训练期间将该位置于有效状态。

此外, LSDT 控制电路 84 可响应执行管线 72 对不训练信号的触发, 而不训练一数据项。执行管线 72 仍然可提供将要不训练的加载作业的加载 PC, 但是在该不训练的情形中, 存储 PC 可以是一不必理会的讯息。加载 / 存储相依性表 80 指向该加载 PC 所指示的数据项, 且
20 LSDT 控制电路 84 将被指向的数据项的有效指示置于无效状态。在采用一有效位作为有效指示的一实施例中, 可清除 (或设定) 该位, 以便指示无效状态。在采用前文所述的位向量之一实施例中, 可将所选择的一位置于无效状态。在位向量的例子中, 如果其它的位保持在有效状态, 则该数据项仍然是处于有效状态。然而, 多个不训练事件最后也可能使每一其它的位变成无效状态。
25

如前文中参照图 3 所述的, 虽然将各 PC 说明为作为加载 / 存储相依性表 80 的索引, 且各 PC 存储在加载 / 存储相依性表 80 及存储 PC/R# 表 82, 但是各实施例可以该等 PC 的一部分的作为索引, 且只存储该等 PC 的一部分。加载 / 存储相依性表 80 内采用的数据项的数目可决定用来作为该表的索引的该部分。例如, 在一特定实施例中,
30 加载 / 存储相依性表 80 可以有 1000 个数据项, 因而可将该 PC 的

10 个位用来作为一索引（例如 10 个最低有效位）。通常可部分地根据该表所占用的区域与加载目标软件时该表一般所呈现的精确度间的取舍，而选择数据项的数目作为设计上的选择。用于加载 / 存储相依性表 80 及存储 PC/R# 表 82 中存储的存储 PC 的位数可以不同于用于索引的位数，且仍然可部分地根据该表所占用的区域与加载目标软件时该表一般所呈现的精确度间的取舍，而选择该位数作为设计上的选择。在一特定实施例

此外，也可部分地根据存储 PC/R# 表 82 所占用的区域与加载目标软件时该表一般所呈现的精确度间的取舍，而以该表中的数据项数目作为设计上的选择。在一特定实施例中，可使用 8-12 个数据项。

请注意，虽然上述的实施例可响应每一派发的指令作业的每一 PC，但是其它的实施例可限制 STLF 预测器 60a 所响应的同时执行的指令作业的数目。在这些实施例中，可响应来自译码单元 24 的该等加载 / 存储指示，而将 STLF 预测器 60a 的该等 PC 及 R# 多任务化，因而只将各加载作业的 PC 输入到加载 / 存储相依性表 80，且只将各存储作业的 PC 输入到存储 PC/R# 表 82，以供存储。在该实施例中，一旦到达了加载作业及（或）存储作业的限制，预测器未命中译码单元 26 可终止一指令线的指令作业。

请注意，作为用于有效指示的位向量、及用来处理被不同的执行中的不同存储作业干扰的各加载作业的相依所有信号之一替代方式，加载 / 存储相依性表 80 中的每一数据项可提供用于多个存储 PC 的存储作业、及对应的有效位。可将来自所选择的一数据项的每一存储 PC 与存储 PC/R# 表 82 比较，并可记录该加载作业对命中存储 PC/R# 表 82 的每一存储作业之一相依性。

现在请参阅图 5，图中示出 STLF 预测器 60 的一第二实施例（STLF 预测器 60b）。其它的实施例也是可行的，且也被考虑到。在图 5 所示的实施例中，STLF 预测器 60b 包含一加载 / 存储相依性表 100、一加法器电路 102、一加载 / 存储相依性表（LSDT）控制电路 104、一 ST/LD 相依性电路 106、及一或有的存储作业确认电路 108。加载 / 存储相依性表 100 被耦合而自译码单元 24 接收派发的指令作业的 PC，且又被耦合而自执行管线 72 接收一加载 PC 及差异 #R。此外，加载 / 存储

相依性表 100 耦合到 LSDT 控制电路 104，且被耦合而将有效指示提供给 ST/LD 相依性电路 106，并将差异 R# 提供给加法器电路 102。加法器电路 102 又被耦合，而自 R# 指定单元 64 接收派发的指令作业的 R#。加法器电路 102 被耦合，而将存储 R# 提供给相依性单元 62 及存储作业确认电路 108，而存储作业确认电路 108 被耦合而自相依性单元 62 接收一有效存储 R# 指示。存储作业确认电路 108 被耦合而将存储有效信号提供给 ST/LD 相依性电路 106，而 ST/LD 相依性电路 106 又被耦合，而自译码单元 24 接收对应于派发的指令作业的加载 / 存储指示。ST/LD 相依性电路 106 被耦合，而将相依性有效信号及相依所有信号提供给相依性单元 62。LSDT 控制电路 104 被耦合，而自执行管线 72 接收训练 / 不训练信号。

一般而言，STLF 预测器 60b 可以下文所述的方式响应一派发的加载作业。加载 PC 用来指向加载 / 存储相依性表 100，因而在多个数据项中选出一个数据项。所选择的该数据项存储一有效指示及一差异 R#。该有效指示指出是否已用与一具有该指向的 PC 的加载作业有关的信息训练了 STLF 预测器 60b，并因而指出该差异 R# 是否为有效的。该差异 R# 是该加载作业的 R# 与在一先前执行期间干扰了该加载作业的一存储作业的 R# 间的差异。因为指令顺序在执行期间通常不会改变，所以在现行执行期间，该加载作业的 R# 与该存储作业的 R# 间的差异通常与在先前执行期间的差异相同。加法器电路 102 将该差异 R# 加上指定给该派发的加载作业的 R#，以便产生一存储 R#，并将该存储 R# 提供给相依性单元 62。相依性单元 62 然后可记录该加载作业对该存储作业的一相依性。在此种方式下，不会将该加载作业的时程（及执行）安排在该存储作业之前。因此，可避免在现行执行期间的干扰。

更具体而言，当派发各指令作业时，利用该等指令作业的 PC 作为加载 / 存储相依性表 100 的索引。除非另有说明，否则本说明的其余部分的焦点将集中在 STLF 预测器 60b 对与一派发的指令作业对应的一输入 PC 的响应。然而，请注意，STLF 预测器 60b 可平行于每一派发的指令作业的每一 PC，而作出响应。加载 / 存储相依性表 100 输出一差异 R#、及来自所选择的数据项之一有效指示。加法器 102 将

该差异 R# 加上对应于该派发的指令作业的 R#，因而产生一存储 R#，并将该存储 R# 传送到相依性单元 62。请注意，加法器电路 102 可包含用于每一派发的指令作业之一加法器，且该加法器接收自加载 / 存储相依性表 100 输出的对应的差异 R#、及 R# 指定单元 64 指定
5 给该派发的指令作业的 R#。

ST/LD 相依性电路 106 接收该有效指示、以及来自译码单元 24 的该指令作业是一加载作业或一存储作业之一指示。ST/LD 相依性电路 106 与前文所述的 ST/LD 相依性电路 86 类似，将一相依性有效信号提供给相依性单元 62。如果 (i) 该指令作业是一加载作业（由来译码
10 单元 24 提供的加载 / 存储指示所决定）；以及 (ii) 来自被指向的数据项的有效指示指出是有效的，则 ST/LD 相依性电路 106 可触发该相依性有效信号。

此外，在一实施例中，STLF 预测器 60b 可采用存储作业确认电路 108。存储作业确认电路 108 自相依性单元 62 接收哪些 R# 对应于
15 待处理的存储作业之一指示。在一实施例中，该指示可以是一位向量，其中每一 R# 具有一个位，用以指示该 R# 是否对应于一存储作业。存储作业确认电路 108 决定加法器电路 102 所产生的该 R# 是否对应于一存储作业，并以该存储有效信号通知 ST/LD 相依性电路 106。如果该存储有效信号被触发，则所产生的该 R# 对应于一存储作业。另
20 一方面，如果该存储有效信号被停止触发，则所产生的该 R# 并不对应于一存储作业。对于采用存储作业确认电路 108 的实施例而言，ST/LD 相依性电路 106 要触发该相依性有效信号的一额外条件是来自存储作业确认电路 108 的该存储有效信号被触发。

仍然与 ST/LD 相依性电路 86 相似，ST/LD 相依性电路 106 的配置可被设定成：在该有效指示是一位向量的各实施例中，提供相依所有信号。就这一点而言，ST/LD 相依性电路 106 的作业可类似于 ST/LD
25 相依性电路 86。

响应来自执行管线 72 的该训练 / 不训练接口，而训练加载 / 存储相依性表 100。更具体而言，如果执行管线 72 触发该训练信号，则
30 LSDT 控制电路 104 更新加载 / 存储相依性表 100。执行管线 72 将作为输入的将要被训练的加载作业的 PC（图 5 所示的 LPC）及对应的

差异 R# 提供给加载 / 存储相依性表 100。加载 / 存储相依性表 100 以该差异 R# 更新该加载 PC 所指向的数据项，且 LSDT 控制电路 104 将该有效指示置于一有效状态。在一实施例中，该有效指示可以是一位，且该有效状态可以是设定（或清除），而无效状态可以是清除（或设定）。在前文所述的另一实施例中，该有效指示可以是一位向量。在该实施例中，LSDT 控制电路 104 可选择该位向量内的一位，并在训练期间将该位置于有效状态。

此外，LSDT 控制电路 104 可响应执行管线 72 对不训练信号的触发，而不训练一数据项。执行管线 72 仍然可提供将要不训练的加载作业的加载 PC，但是在该不训练的情形中，该差异 R# 可以是一不必理会的讯息。加载 / 存储相依性表 100 指向该加载 PC 所指示的数据项，且 LSDT 控制电路 104 将被指向的数据项的有效指示置于无效状态。在采用一有效位作为有效指示的一实施例中，可清除（或设定）该位，以便指示无效状态。在采用前文所述的位向量之一实施例中，可将所选择的一位置于无效状态。在位向量的例子中，如果其它的位保持在有效状态，则该数据项仍然是处于有效状态。然而，多个不训练事件最后也可能使每一其它的位变成无效状态。

如前文中参照图 4 所述的，虽然将各 PC 说明为作为加载 / 存储相依性表 100 的索引，但是各实施例可以只用该等 PC 的一部分的作为索引。加载 / 存储相依性表 100 内采用的数据项的数目可决定用来作为该表的索引的该部分。例如，在一特定实施例中，加载 / 存储相依性表 100 可以有 1000 个数据项，因而可将该 PC 的 10 个位用来作为一索引（例如 10 个最低有效位）。通常可部分地根据该表所占用的区域与加载目标软件时该表一般所呈现的精确度间的取舍，而选择数据项的数目作为设计上的选择。

请注意，虽然在本实施例中于训练期间将该差异 R# 提供给 STLF 预测器 60b，但是其它的实施例可提供在入籍存储 R#，且可在 STLF 预测器 60b 中计算该差异 R#。此外，各实施例可将该加载作业的差异 R# 及 R# 相加或相减，以便产生该存储作业的 R#。此外，存储作业确认电路 108 的一替代配置可查询排程器 36 中的加法器电路 102 所产生的该存储 R#，以便决定该指令作业是否为一存储作业。

现在请参阅第 6 图，图中示出执行管线 72 的一实施例对加载内存作业的作业的流程图。其它的实施例也是可行的，也是可考虑的。虽然为了易于了解而以一特定的顺序示出图 6 所示的该等步骤但是也可使用任何适当的顺序。更具体而言，可由执行管线 72 内的组合逻辑电路以平行的方式执行各步骤。此外，可在执行管线 72 内的不同状态下执行各步骤。可将与其它步骤有关的信息管线化到用来执行这些步骤的该等阶段。

在决定步骤 110 中，执行管线 72 决定是否已安排一加载作业的时程以供执行。如果尚未安排一加载作业的时程，则在该实施例中无法执行任何训练作业。如果已安排了一加载作业的时程，则执行管线 72 在决定步骤 112 中决定该加载作业是否因命中了物理地址缓冲器 70 而已退休。更具体而言，执行管线 72 可检查来自分配给该加载作业的排程缓冲器数据项的重新尝试指示。如果该加载作业因一物理地址缓冲器命中而退休，则执行管线 72 在步骤 114 中将训练信号触发到 STLTF 预测器 60，并将要被训练的加载作业及存储作业的加载 PC 及存储 ID 提供给 STLTF 预测器 60。

另一方面，如果该加载作业并未因一物理地址缓冲器命中而退休，则执行管线 72 在决定步骤 116 中决定该加载作业是否因 STLTF 预测器 60 的作业而有对一存储作业的相依性。换言之，执行管线 72 决定分配给该加载作业的排程缓冲器数据项中的该训练指示是否指示该加载作业已被训练过。如果该加载作业已被训练过，则执行管线 72 在决定步骤 118 中决定是否要为该加载作业而自存储队列转送数据。如果不转送数据，则该加载作业可能尚未被一存储作业干扰。因此，在此种情形中，执行管线 72 在步骤 120 中将不训练信号触发到 STLTF 预测器 60，并提供不训练的该加载作业的加载 PC。

请注意，亦可在执行一干扰了一加载作业的存储作业期间执行训练，而不是在因重新尝试而重新执行该加载作业期间执行训练。

现在请参阅图 7，图中示出一 LSDT 控制电路 130 实施例的一部分的方块图。在各实施例中，LSDT 控制电路 130 可用来作为 LSDT 控制电路 84 及（或）LSDT 控制电路 104。其它的实施例也是可行的，且是可考虑的。在图 7 所示的实施例中，LSDT 控制电路 130 包含一

控制电路 132、及耦合到该控制电路之一计数器电路 134。控制电路 132 被耦合而自执行管线 72 接收训练 / 不训练信号，且被耦合而将 Set_V[3:0] 信号及 Clear_V[3:0] 信号提供给加载 / 存储相依性表 80 或 100（根据实施例而定）。

- 5 LSDT 控制电路 130 的配置被设定成：在有效指示是位向量的实施例中，于训练及不训练期间，管理加载 / 存储相依性表中的有效指示。在本实施例中，该位向量中的每一位在被设定的情形下是处于有效状态，且在被清除的情形下是处于无效状态，但是在替代实施例中，该位向量中的每一位在被清除的情形下可以是处于有效状态，且在被设定的情形下是处于无效状态。还有其它的实施例可在该等位中将有效状态编码。
- 10

如果正在训练一数据项，则控制电路 132 响应计数器电路 134 所维护的该值，而选择该位向量中之一位为被设定。同样地，如果正在不训练一数据项，则控制电路 132 响应计数器电路 134 所维护的该值，而选择该位向量中之一位为被清除。计数器电路 134 中的每一值选择该位向量中的一个位。计数器电路 134 包含一计数缓存器、及用来递增该计数缓存器中的值之一递增器。因此，计数器电路 134 递增每一时脉周期。因此，在本实施例中，在一特定训练或不训练中所选择的该位可以是虚拟随机的。

15

- 20 在本实施例中，有效指示是 4 位的向量。因此，Set_V[3:0] 及 Clear_V[3:0] 内的一信号对应于该向量中的每一位。如果正在训练一数据项，则控制电路 132 触发对应于根据计数器电路 134 而选择的位的 Set_V[3:0] 信号。加载 / 存储相依性表 80 或 100 响应时，设定被指向的数据项的位向量中对应的位。另一方面，如果正在不训练一数据项，则控制电路 132 触发对应于根据计数器电路 134 而选择的位的 Clear_V[3:0] 信号。加载 / 存储相依性表 80 或 100 响应时，清除被指向的数据项的位向量中对应的位。如有需要，控制电路 132 亦可提供一写入起动信号，以便起动对被指向的数据项的更新。
- 25

各替代配置并不递增每一时脉周期的计数，而是可在需要时在每一训练或不训练事件之后递增该计数。此外，替代配置可选择一处于无效状态的位，而在训练期间将该位改变为有效状态；且可选择一处于

30

有效状态之位，而在不训练期间将该位改变为无效状态。

现在请参阅图 8，图中示出一相依性单元 62 实施例的一部分的方块图。其它的实施例也是可行的，且是可考虑的。图 8 所示的该部分可与维护一用来指示待处理的存储作业之一存储位向量有关。其它部分（图中未示出）的配置可被设定成记录被派发到排程器 36 的各指令作业的相依性。在图 8 所示的实施例，相依性单元 62 包含一控制电路 140、及转移至控制电路 140 的一位向量存储区 142。控制电路 140 又被耦合，而自译码单元 24 接收派发的指令作业的加载 / 存储本质之一指示，并自 R# 指定单元 64 接收所指定的 R#。此外，控制电路 140 被耦合而自排程器 36 接收退休的各 R# 及一异常结束指示。将该存储位向量自位向量存储区 142 传送到存储作业确认电路 108。

一般而言，当派发指令时，控制电路 140 自译码单元 24 接收存储内存作业的指示。自 R# 指定单元 64 提供各对应的 R#。位向量存储区 142 中的存储位向量包含每一 R# 之一位。控制电路 140 设定该存储位向量中对应于派发的存储作业的该等位。同样地，当排程器 36 使存储作业退休，且经由该等退休 R# 而指示该等存储作业时，控制电路 140 复位该存储位向量中的各对应的位。最后，如果通知了一异常结束，则控制电路 140 复位异常结束的该等存储作业的位。在一实施例中，当造成该异常结束的指令作业退休时，可通知异常结束。因此，该异常结束指示可以是用来清除该存储位向量之一信号。在其它的实施例中，该异常结束指示可识别异常结束的指令的 R#，且可只异常结束较新的存储作业。

在本文的用法中，术语“控制电路”意指操作各输入以便产生前文所述的输出的电路。一般而言，一控制电路可包含组合逻辑电路（静态的或动态的）、状态机、客户订制电路、及时脉同步存储装置（例如正反器、缓存器等）的任何组合。

计算机系统

现在请参阅图 9，图中示出一计算机系统 200 的一实施例的方块图，该计算机系统 200 包含处理器 10，处理器 10 经由一总线桥接器 202 而耦合到各种系统组件。其它的实施例也是可行的，且是可考虑的。

在所示的该系统中，主存储器 204 经由一内存总线 206 而耦合到总线桥接器 202，且一图形控制器 208 经由一 AGP 总线 210 而耦合到总线桥接器 202。最后，该等多个 PCI 装置 212A - 212B 经由一 PCI 总线 214 而耦合到总线桥接器 202。可进一步设有一辅助总线桥接器 216，
5 以便经由一 EISA/ISA 总线 220 而将一电气接口提供给一个或多个 EISA 或 ISA 装置 218。处理器 10 经由一 CPU 总线 224 而耦合到总线桥接器 202，且处理器 10 耦合到一或有的 L2 高速缓存 228。CPU 总线 224 及 L2 高速缓存 228 的接口合而可包含外部接口 52。

总线桥接器 202 在处理器 10、主存储器 204、图形控制器 208、与
10 连接到 PCI 总线 214 的各装置之间提供了一接口。当自连接到总线桥接器 202 的其中一个装置接收到一作业时，总线桥接器 202 识别该作业的目标（例如一特定的装置，或者在 PCI 总线 214 的情形中为 PCI 总线 214 上的目标装置。总线桥接器 202 将该作业绕送到该目标装置。总线桥接器 202 通常将一作业自来源装置或总线所用的协议转换为目
15 标装置或总线所用的协议。

辅助总线桥接器 216 除了将一 ISA/EISA 总线接口提供给 PCI 总线 214 之外，亦可视需要而进一步设有额外的功能。计算机系统 200 内亦可包含在辅助总线桥接器 216 之外或与辅助总线桥接器 216 整合的一输入 / 输出控制器（图中未示出），用以视需要而提供对键盘及
20 鼠标 222 之作业支持，并提供各种序列及并行端口。在其它的实施例，亦可进一步在处理器 10 与总线桥接器 202 之间将一外部高速缓存单元（图中未示出）耦合到 CPU 总线 224。在替代实施例中，可将该外部高速缓存耦合到总线桥接器 202，并可将该外部高速缓存的高速缓存控制逻辑电路整合到总线桥接器 202。进一步在处理器 10 的背部配置中
25 示出 L2 高速缓存 228。请注意，L2 高速缓存 228 可与处理器 10 分离，可连同处理器 10 而整合到一卡匣（例如 slot 1 或 slot A）中，甚至可将 L2 高速缓存 228 连同处理器 10 而整合到一半导体基材中。

主存储器 204 是一种用来存储应用程序且处理器 10 主要用来执行的内存。一适当的主存储器 204 包含动态随机存取内存（Dynamic
30 Random Access Memory；简称 DRAM）。例如，多个记忆组的同步 DRAM（Synchronous DRAM；简称 SDRAM）或 Rambus DRAM

(RDRAM) 是适用的。

PCI 装置 212A - 212B 用来例示各种周边装置，例如网络适配卡、视讯加速器、声卡、硬盘机或软盘机或磁盘驱动器控制器、小型计算机系统接口 (Small Computer System Interface; 简称 SCSI) 适配卡、
5 及电话卡。同样地，EISA 或 ISA 装置 218 用来例示各种周边装置，例如一调制解调器、一声卡、以及诸如 GPIB 或专业总线适配卡等的各种数据撷取卡。

设有图形控制器 208，用以控制一显示器 226 上的文字及影像的表现处理。图形控制器 208 可采用此项技艺中习知的一典型图形加速器，
10 以便呈现可被有效地移进及移出主存储器 204 的三维数据结构。图形控制器 208 可因而是 AGP 总线 210 的一主控装置，其方式为图形控制器 208 可要求及接收对总线桥接器 202 内的一目标接口的存取，因而可存取主存储器 204。一专用的图形总线可迅速地自主存储器 204 提取数据。对于某些作业而言，图形控制器 208 的配置可进一步被被
15 设定成在 AGP 总线 210 上产生 PCI 协议交易。总线桥接器 202 的 AGP 接口因而可包含用来支持 AGP 协议交易以及 PCI 协议目标及发出装置协议的功能。显示器 226 是可呈现影像及文字的任何电子显示器。一适用的显示器 226 包含一阴极射线管 (Cathode Ray Tube; 简称 CRT) 及一液晶显示器 (Liquid Crystal Display; 简称 LCD) 等。

20 请注意，虽然前文中将 AGP、PCI、以及 ISA 或 EISA 总线用来作为例子，但是可视需要而以任何总线架构替代。又请注意，计算机系统 200 可以是包含额外的处理器 (例如图中示为计算机系统 200 的一或有组件的处理器 10a 之一多重处理计算机系统。处理器 10a 可类似于处理器 10。更具体而言，处理器 10a 可以与处理器 10 相同。可经
25 由一独立的总线(如图 9 所示)而将处理器 10a 连接到总线桥接器 202，或者处理器 10a 可与处理器 10 共享 CPU 总线 224。此外，可将处理器 10a 耦合到与 L2 高速缓存 228 类似的一或有的 L2 高速缓存 228a。

现在请参阅图 10，图中示出计算机系统 300 的另一实施例。其它的实施例也是可行的，且是可考虑的。在图 10 所示的实施例中，计
30 算机系统 300 包含数个处理节点 312A、312B、312C、及 312D。每一处理节点经由每一各别处理节点 312A - 312D 内包含的一内存控制器

316A - 316D 而耦合到一各别的内存 314A - 314D。此外，各处理节点 312A - 312D 包含用来在处理节点 312A - 312D 之间通信的接口逻辑电路。例如，处理节点 312A 包含：用来与处理节点 312B 通信的接口逻辑电路 318A、用来与处理节点 312C 通信的接口逻辑电路 318B、及用来与另一处理节点（图中未示出）通信的接口逻辑电路 318C。同样地，处理节点 312B 包含接口逻辑电路 318D、318E、及 318F；处理节点 312C 包含接口逻辑电路 318G、318H、及 318I；以及处理节点 312D 包含接口逻辑电路 318J、318K、及 318L。处理节点 312D 被耦合，而经由接口逻辑电路 318L 与多个输入 / 输出装置（例如在链环配置中的装置 320A - 320B 通信。其它的处理节点可以一种类似的方式与其它的 I/O 装置通信。

处理节点 312A - 312D 实施一用于处理节点间通信的封包式链路。在本实施例中，将该链路实施为若干组单向线路（例如，线路 324A 用来将封包自处理节点 312A 传送到处理节点 312B，且线路 324B 用来将封包自处理节点 312B 传送到处理节点 312A）。如图 10 所示，其它组的线路 324C - 324H 用来在其它的处理节点之间传送封包。一般而言，每一组线路 324 可包含一条或多条数据线、对应于该等数据线之一条或多条时脉线、以及用来指示所传送的封包类型之一条或多条控制线。可在一高速缓存一致的方式下操作该链路，以便在各处理节点之间通信；或者可在一无一致性的方式下操作该链路，以便在一处理节点与一 I/O 装置之间通信（或一总线桥接器至诸如 PCI 总线或 ISA 总线等的传统结构的一 I/O 总线的通信。此外，可利用如图所示的各 I/O 装置间之一链环结构而以一非一致性操作该链路。请注意，要自一处理节点传送到另一处理节点的一封包可通过一个或多个中间节点。例如，如图 10 所示，由处理节点 312A 传送到处理节点 312D 的一封包可通过处理节点 312B 或处理节点 312C。可使用任何适当的路由算法。计算机系统 300 的其它实施例可包含比图 10 所示的较多或较少的处理节点。

一般而言，可在各节点之间的线路 324 上以一个或多个位时间的方式传送该等封包。一位时间可以是对应的的时脉线上的时脉信号的上升缘或下降缘。该等封包可包含：用来激活交易的命令封包、用来保

持高速缓存一致性的探查封包、及响应探查及命令的响应封包。

处理节点 312A - 312D 除了包含内存控制器及接口逻辑电路之外，还可包含一个或多个处理器。广义而言，一处理节点包含至少一个处理器，且可视需要而选择包含一内存控制器，用以与一内存及其它的逻辑电路通信。更具体而言，一处理节点 312A - 312D 可包含处理器 10。外部接口单元 46 可包含该节点内的接口逻辑电路 318、及内存控制器 316。

内存 314A - 314D 可包含任何适用的内存装置。例如，一内存 314A - 314D 可包含一个或多个 RAMBUS DRAM (RDRAM)、同步 DRAM (SDRAM)、静态 RAM 等。将计算机系统 300 的地址空间分配到各内存 314A - 314D。每一处理节点 312A - 312D 可包含一用来决定哪些地址被对映到哪些内存 314A - 314D 以及应对一特定地址的一内存要求绕送到哪一处理节点 312A - 312D 之内存对映表。在一实施例中，计算机系统 300 内的一地址之一致性点是被耦合到用来存储对应于该地址的字节之内存之内存控制器 316A - 316D。换言之，内存控制器 316A - 316D 负责确保以一种高速缓存一致性的方式进行对应的内存 314A - 314D 的每一内存存取。内存控制器 316A - 316D 可包含用来连接到内存 314A - 314D 的控制电路。此外，内存控制器 316A - 316D 可包含要求队列，用以让各内存要求排队等候。

一般而言，接口逻辑电路 318A - 318L 可包含各种缓冲器，用以自该链路接收封包，并缓冲存储将要经由该链路传送的封包。计算机系统 300 可采用让和适用的流程控制机制，用以传送封包。例如，在一实施例中，每一接口逻辑电路 318 将每一类型的缓冲器的数目存储在用来连接该接口逻辑电路的链路的另一端上的接收器内。除非接收的接口逻辑电路具有一可存储一封包的未被占用的缓冲器，否则前述的接口逻辑电路将不会传送该封包。当向前传送一封包而空出一接收的缓冲器时，该接收的接口逻辑电路将一讯息传输到该传送的接口逻辑电路，告知已空出该缓冲器。可将此种机制称为一“基于配给许可的系统”。

装置 320A - 320B 可以是任何适用的 I/O 装置。例如，装置 320A - 320B 可包含网络适配卡、视讯加速器、声卡、硬盘机或软盘机或磁盘

驱动器控制器、小型计算机系统接口（SCSI）适配卡、电话卡、调制解调器、以及诸如 GPIB 或专业总线适配卡等的各种数据摄取卡。

熟习此项技艺者在完全了解前文所述的揭示事项之后，将可易于作出多种变化及修改。将把下文的权利要求诠释为包含所有此类的变化
5 及修改。

工业上的应用

本发明可适用于处理器及计算机系统。

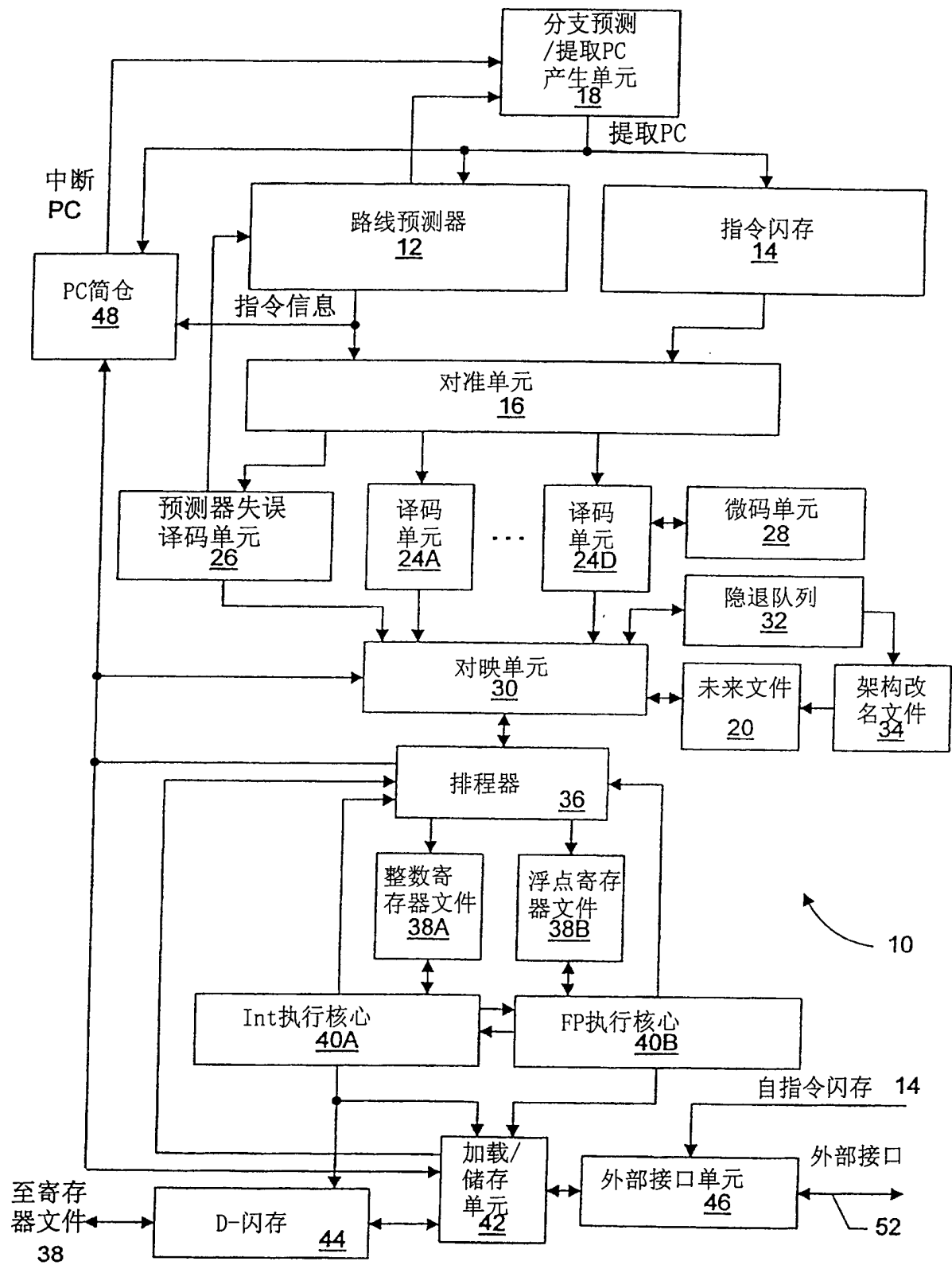


图1

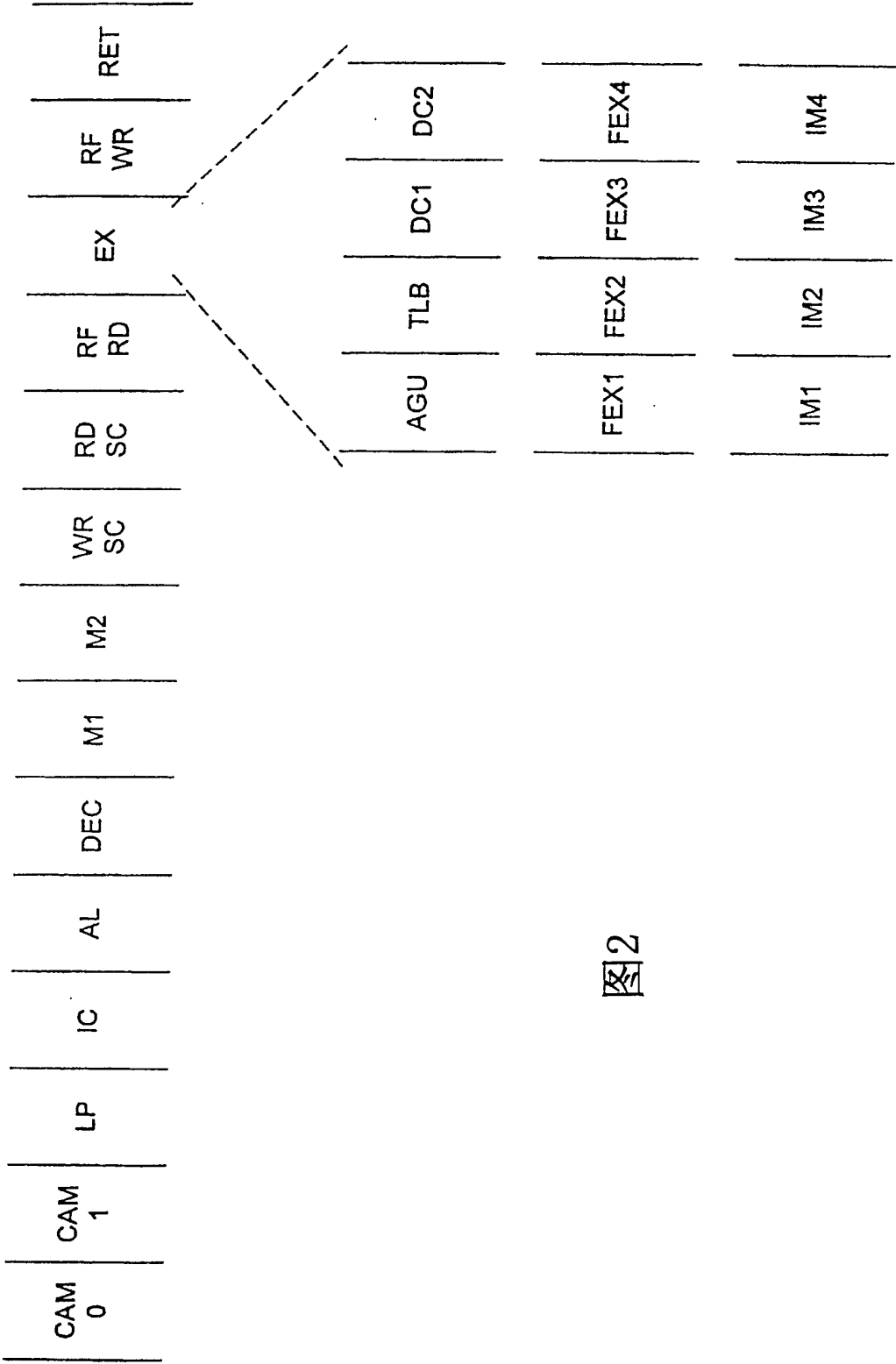


图2

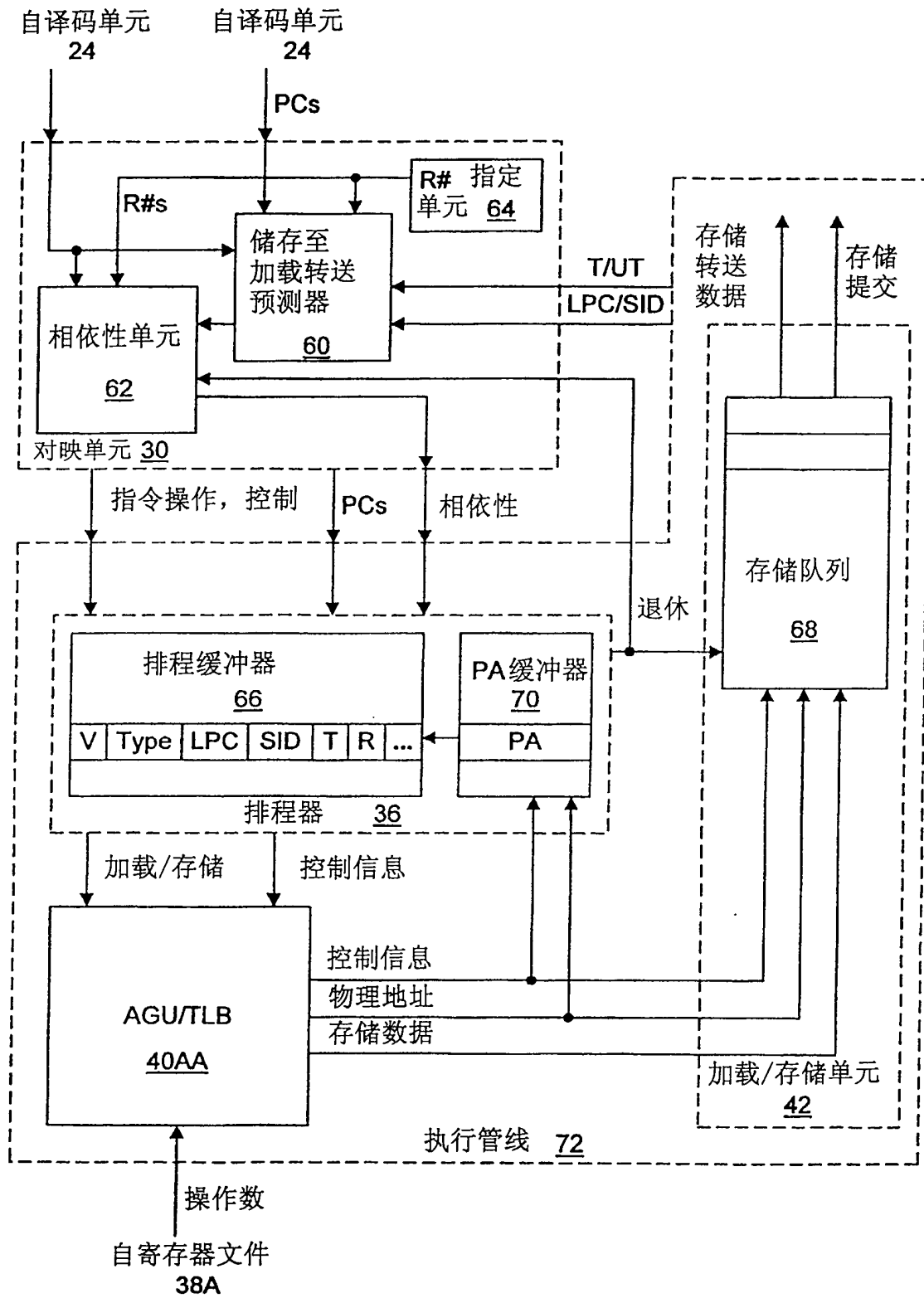


图3

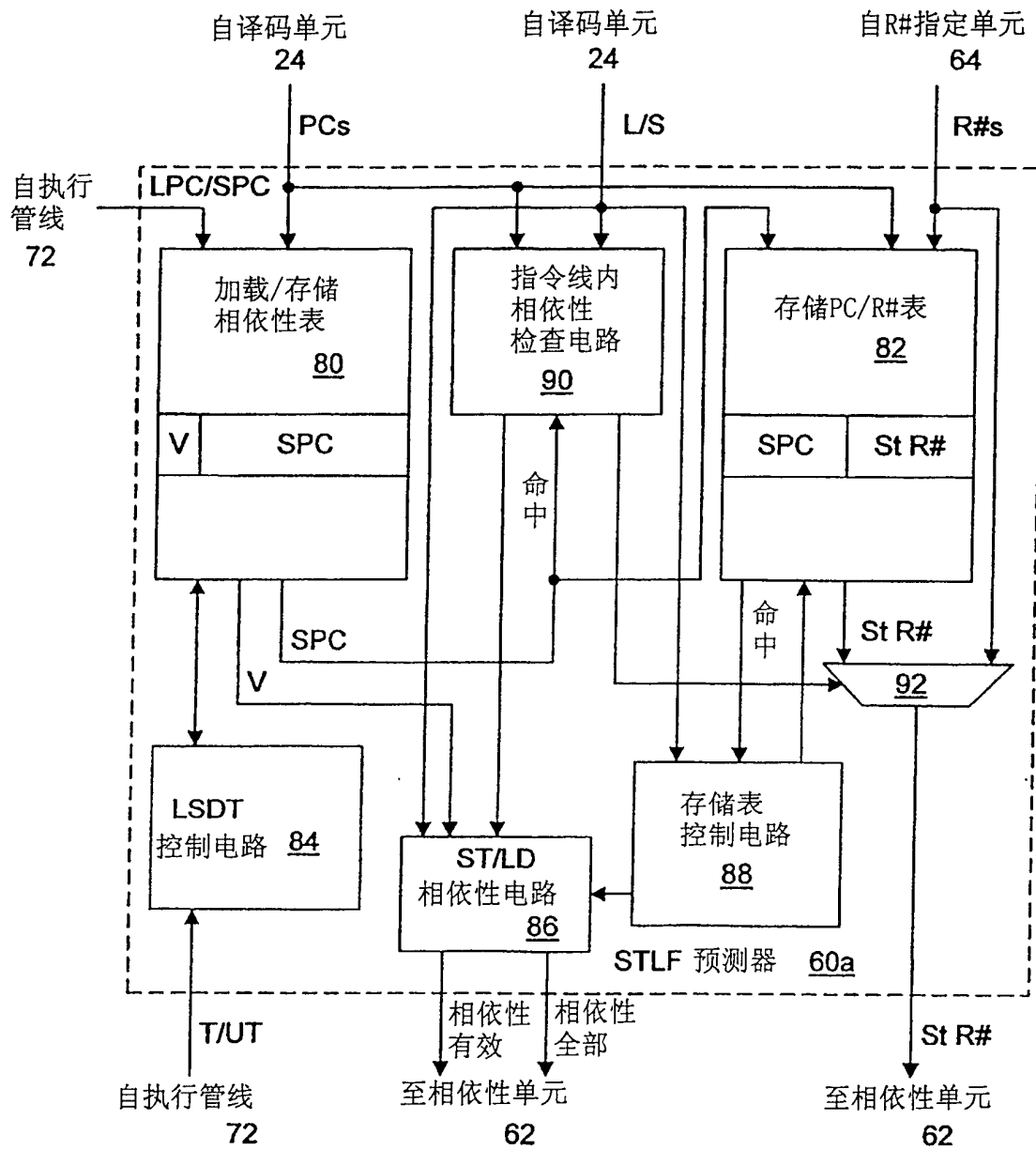


图4

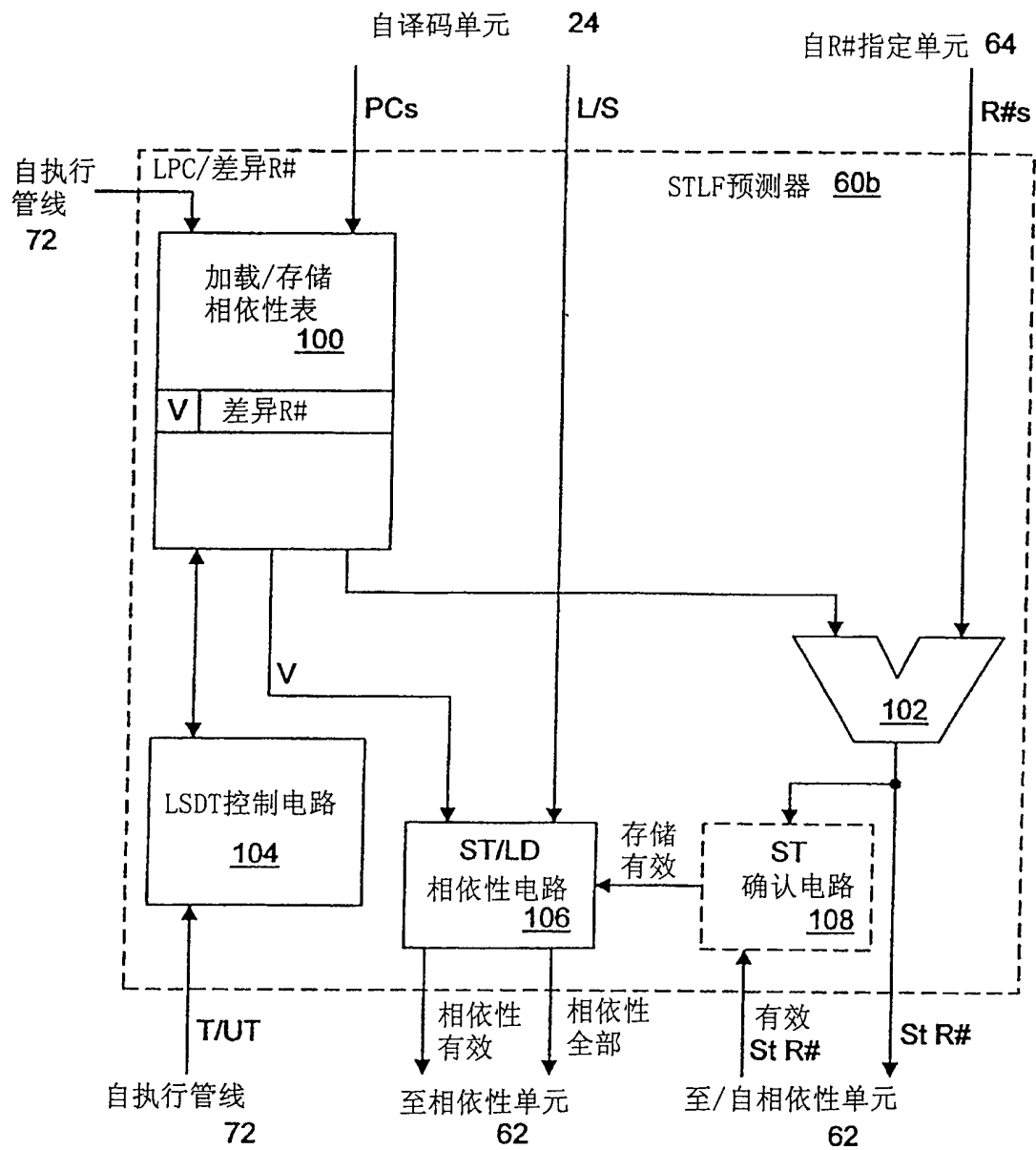


图5

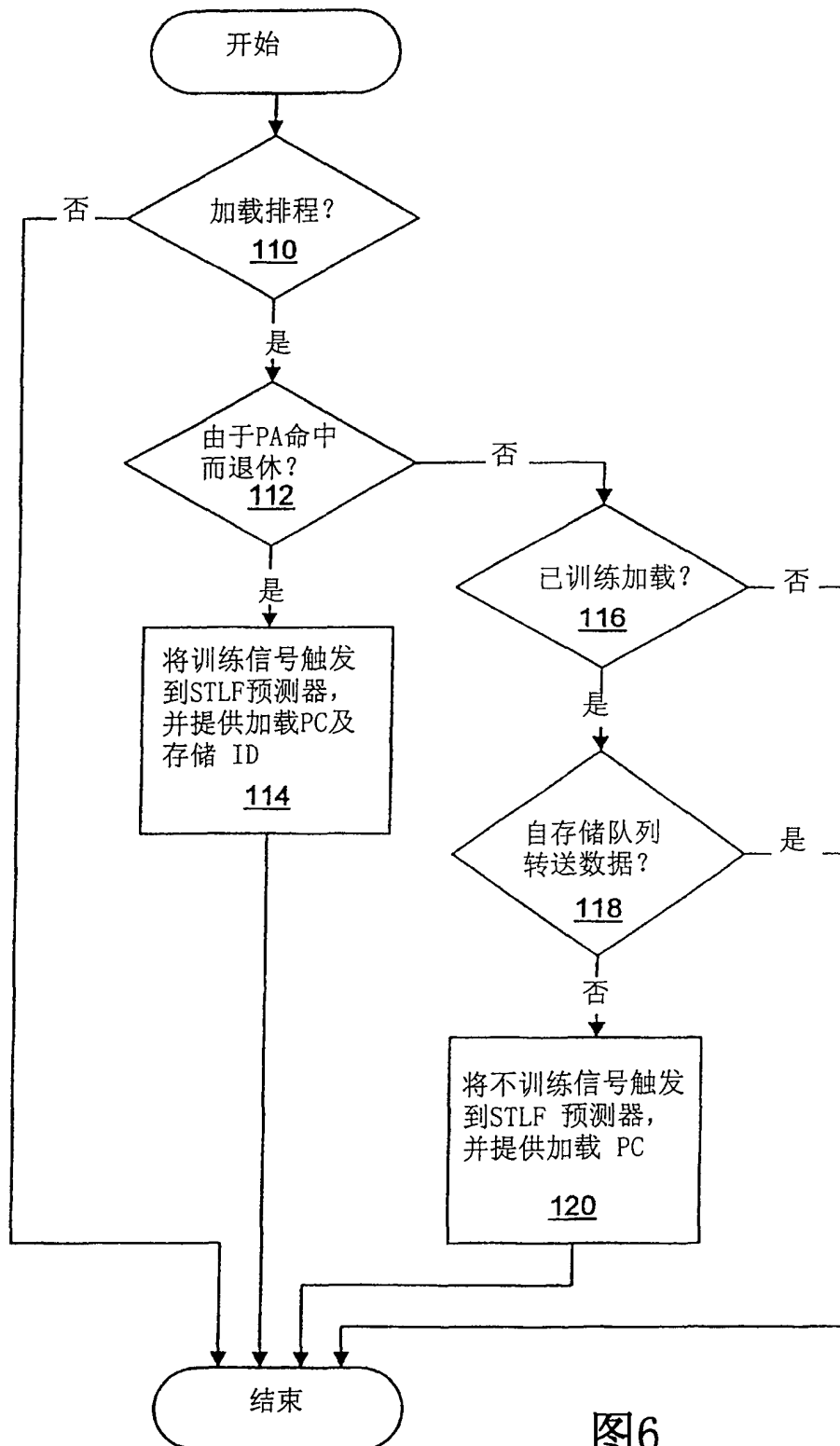


图6

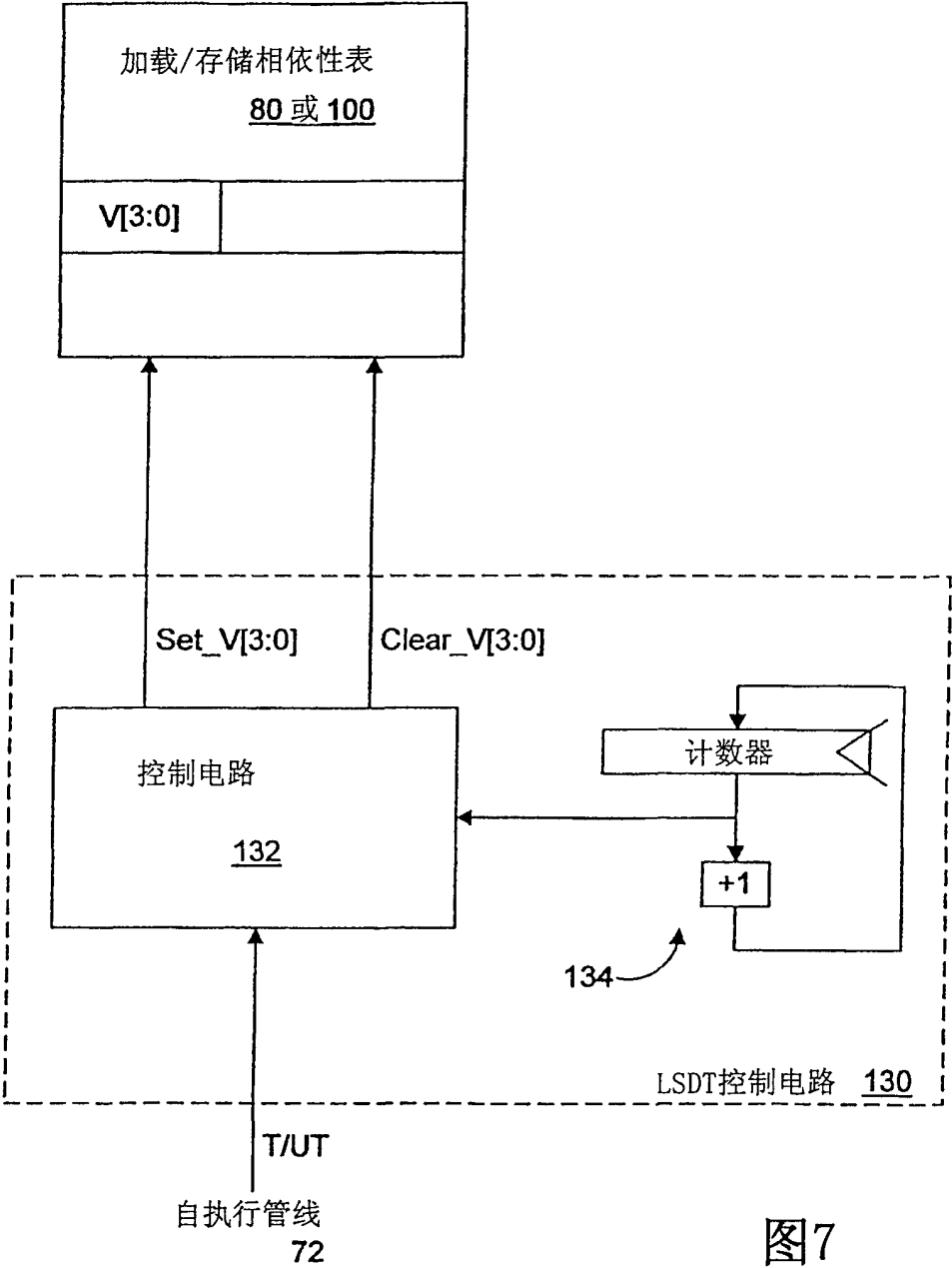


图7

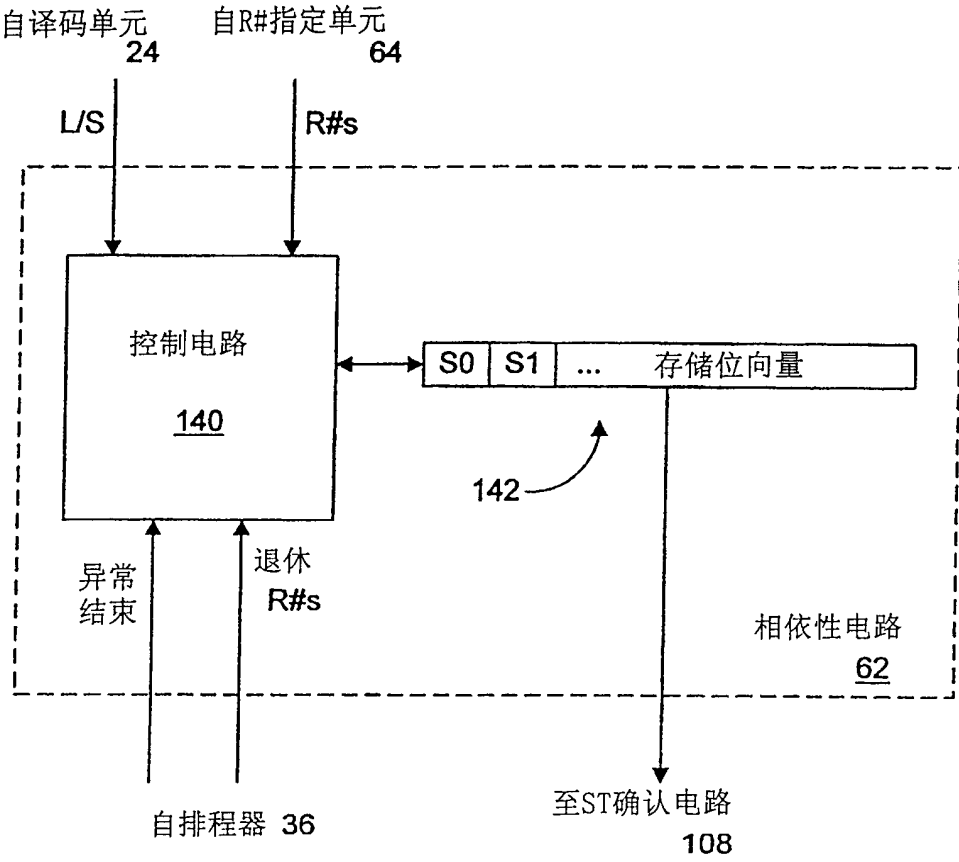


图8

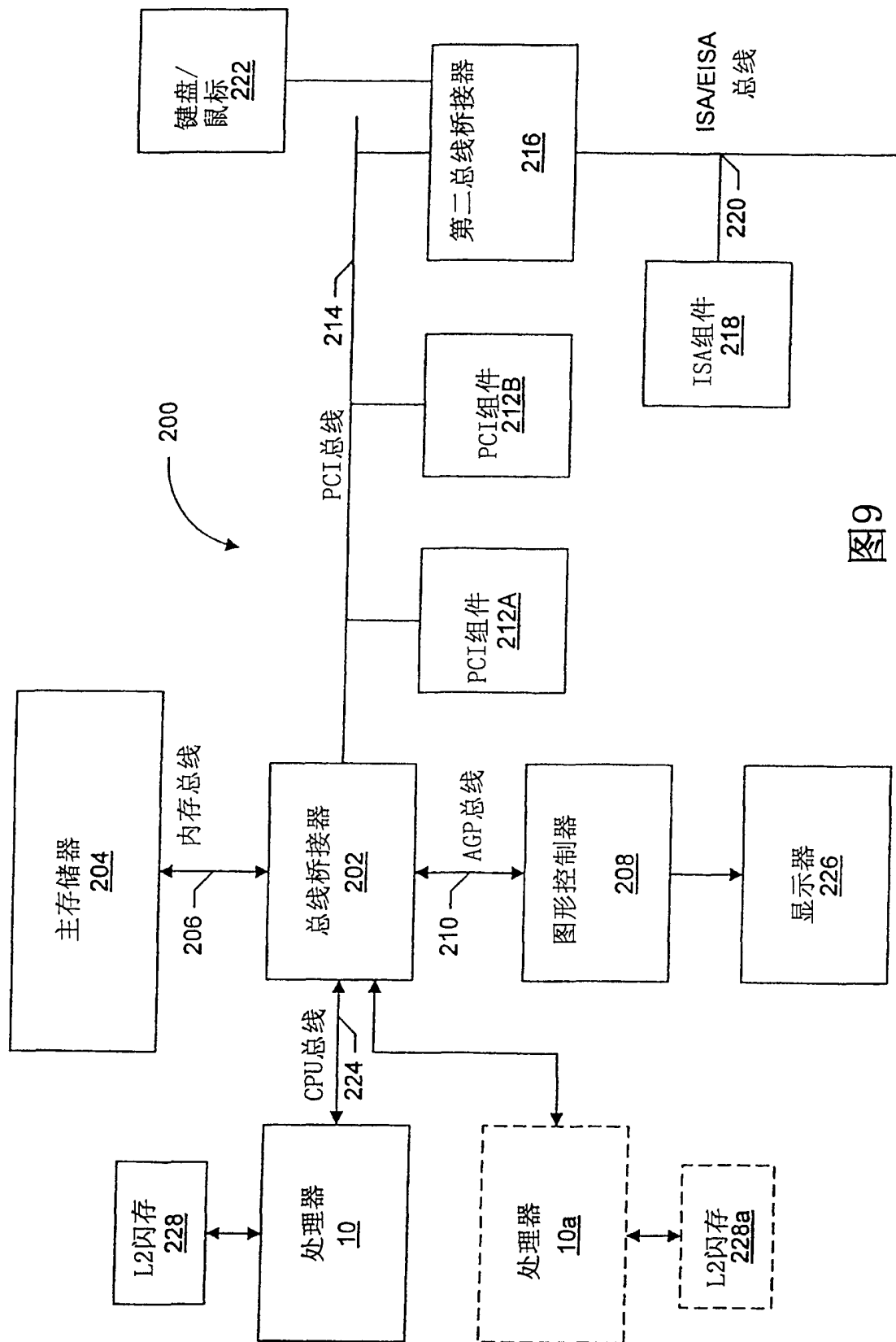


图9

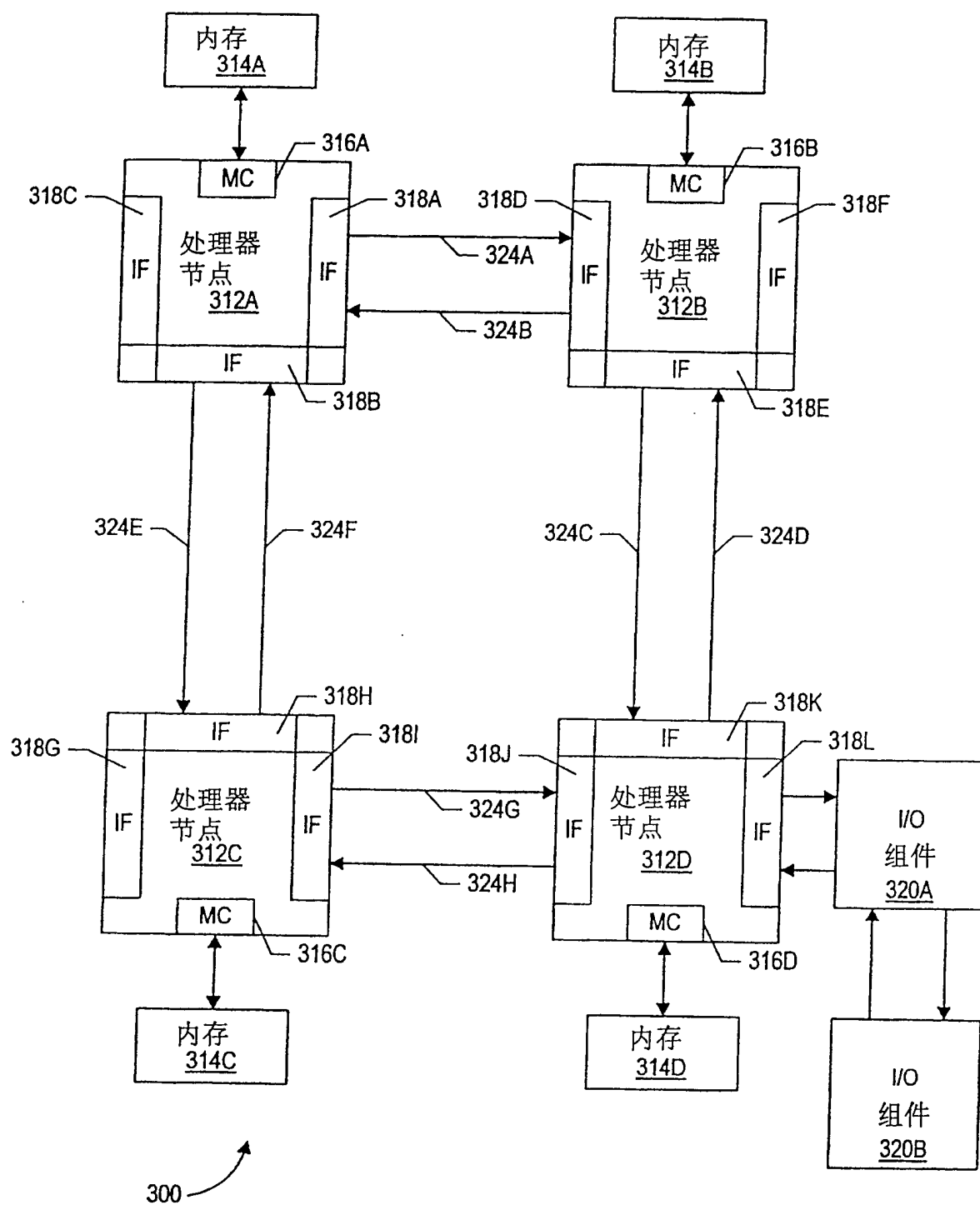


图10