



(12) 发明专利申请

(10) 申请公布号 CN 102326155 A

(43) 申请公布日 2012. 01. 18

(21) 申请号 201080008296. 1

代理人 秦晨

(22) 申请日 2010. 01. 22

(51) Int. Cl.

(30) 优先权数据

12/389, 153 2009. 02. 19 US

G06F 12/06 (2006. 01)

G06F 12/08 (2006. 01)

G06F 9/06 (2006. 01)

(85) PCT申请进入国家阶段日

2011. 08. 19

(86) PCT申请的申请数据

PCT/US2010/021780 2010. 01. 22

(87) PCT申请的公布数据

W02010/096233 EN 2010. 08. 26

(71) 申请人 飞思卡尔半导体公司

地址 美国得克萨斯

(72) 发明人 W·C·莫耶 R·G·柯林斯

(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 11038

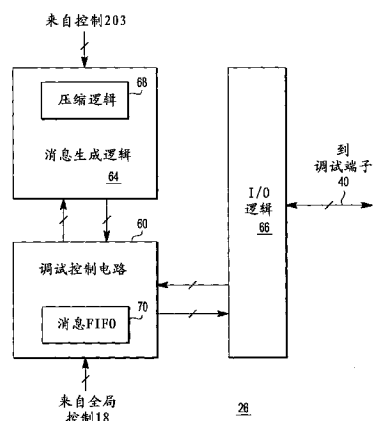
权利要求书 2 页 说明书 13 页 附图 8 页

(54) 发明名称

用于调试的地址转换跟踪消息的生成

(57) 摘要

一种通过允许外部调试工具具有实时跟踪功能来生成调试消息的数据处理系统 (10) 及方法。数据处理器 (20、22、24) 执行多个数据处理指令并且使用存储器 (30) 来存储信息。调试电路 (26) 生成包括地址转换跟踪消息的调试消息。存储器管理单元 (16) 具有用于实施地址转换的地址转换逻辑 (205), 以使地址在虚拟形式和物理形式之间转换。调试电路 (26) 包括与存储器管理单元 (16) 耦接, 用于在一个或多个地址转换映射被修改时接收通知的消息生成电路 (64)。消息生成电路 (64) 响应于检测到地址转换映射发生修改而生成地址转换跟踪消息并且将地址转换跟踪消息提供给调试电路 (26) 外部。



1. 一种生成调试消息的数据处理系统,包括:
数据处理器,用于执行多个数据处理指令;
存储器,与所述数据处理器耦接,用于存储和提供信息给所述数据处理器;
调试电路,与所述数据处理器耦接,用于生成包括地址转换跟踪消息的调试消息;
存储器管理单元,与所述调试电路和数据处理器耦接,所述存储器管理单元包括用于实施地址转换以使地址在虚拟形式和物理形式之间转换的地址转换逻辑,其中:
所述调试电路包括:
消息生成电路,与所述存储器管理单元耦接,用于当一个或多个地址转换映射被修改时接收通知,所述消息生成电路响应于检测到地址转换映射的修改而生成地址转换跟踪消息并且将所述地址转换跟踪消息提供给所述调试电路外部。
2. 根据权利要求1所述的数据处理系统,其中所述存储器管理单元每当任何地址转换映射被更新时发送指示符信号给所述消息生成电路。
3. 根据权利要求2所述的数据处理系统,其中所述消息生成电路还包括:
压缩逻辑,用于选择性地去除所述地址转换跟踪消息中的至少一个字段。
4. 根据权利要求3所述的数据处理系统,其中所述地址转换跟踪消息中由所述压缩逻辑去除的所述至少一个字段中的每个都包含用于该字段的预定的普遍值。
5. 根据权利要求4所述的数据处理系统,其中所述预定的普遍值由所述数据处理系统的用户控制,所述预定的普遍值被存储在可由所述数据处理系统的所述用户访问的存储位置中。
6. 根据权利要求3所述的数据处理系统,其中被去除的所述至少一个字段是地址转换大小字段或者用于识别进程标识符或地址空间之一的地址转换识别字段。
7. 根据权利要求1所述的数据处理系统,其中所述一个或多个地址转换映射被存储在转换旁视缓冲器(TLB)中。
8. 根据权利要求7所述的数据处理系统,其中所述存储器管理单元每当所述TLB写TLB条目或使条目无效时发送指示符信号给所述消息生成电路。
9. 根据权利要求1所述的数据处理系统,还包括:
调试端子,与所述调试电路耦接;以及
调试器,与所述调试端子耦接,用于从所述调试电路接收所述地址转换跟踪消息。
10. 一种用于生成调试消息的方法,包括:
以数据处理器执行多个数据处理指令;
将调试电路耦接至所述数据处理器用于生成包括地址转换跟踪消息的调试消息;
将存储器管理单元耦接至所述调试电路和数据处理器,所述存储器管理单元包括用于实施地址转换以使地址在虚拟形式和物理形式之间转换的转换旁视缓冲器(TLB);
检测TLB条目何时被修改;
响应于检测到TLB条目的修改而生成地址转换跟踪消息;以及
将所述地址转换跟踪消息提供给调试端子。
11. 根据权利要求10所述的方法,还包括:
每当所述TLB写TLB条目或者使条目无效时,将来自所述存储器管理单元的指示符信号发送给所述消息生成电路。

12. 根据权利要求 11 所述的方法,还包括:

通过去除所述地址转换跟踪消息中的至少一个字段来压缩所述地址转换跟踪消息的长度。

13. 根据权利要求 12 所述的方法,还包括:

使被去除的所述至少一个字段与地址转换大小字段或者用于识别地址空间或进程标识符的地址转换识别字段相关联。

14. 根据权利要求 12 所述的方法,还包括:

使预定的普遍值与所述地址转换跟踪消息的所述至少一个字段中的的每一个相关联;以及

仅当如果所述预定的普遍值针对所述至少一个字段存在时去除所述至少一个字段。

15. 根据权利要求 14 所述的方法,还包括:

由所述数据处理系统的用户控制所述预定的普遍值;以及

将所述预定的值存储在多个用户可访问的寄存器中的一个或多个内。

16. 一种数据处理系统,包括:

全局控制电路,用于控制所述数据处理系统;

数据处理器,与所述全局控制电路耦接,所述数据处理器包括与指令解码器耦接的取指单元,所述指令解码器与用于执行多个数据处理指令的一个或多个执行单元耦接;

存储器管理单元,与所述全局控制电路和所述数据处理器耦接,所述存储器管理单元包括用于实施地址转换以使地址在虚拟形式和物理形式之间转换的地址转换逻辑;

总线接口单元,用于使所述存储器管理单元与全局互连通过接口连接;

存储器,与所述全局互连耦接,用于存储来自所述数据处理器器的信息以及向所述数据处理器提供信息;以及

调试电路,与所述全局控制电路和所述存储器管理单元耦接,所述调试电路生成包括地址转换跟踪消息的调试消息,其中:

所述调试电路包括:

消息生成电路,与所述存储器管理单元耦接,用于当一个或多个地址转换映射被修改时接收通知,所述消息生成电路响应于检测到地址转换映射的修改而生成地址转换跟踪消息并且将所述地址转换跟踪消息提供给所述调试电路外部。

17. 根据权利要求 16 所述的数据处理系统,其中所述消息生成电路还包括:

压缩逻辑,用于选择性地去除所述地址转换跟踪消息中的至少一个字段。

18. 根据权利要求 17 所述的数据处理系统,其中所述地址转换跟踪消息中由所述压缩逻辑去除的所述至少一个字段中的每个都包含用于该字段的预定的普遍值。

19. 根据权利要求 18 所述的数据处理系统,其中所述预定的普遍值由所述数据处理系统的用户控制,所述预定的普遍值被存储在可由所述数据处理系统的所述用户访问的存储位置中。

20. 根据权利要求 16 所述的数据处理系统,其中所述一个或多个地址转换映射被存储在转换旁视缓冲器(TLB)中,并且所述存储器管理单元每当所述 TLB 写 TLB 条目或使条目无效时发送指示符信号给所述消息生成电路。

用于调试的地址转换跟踪消息的生成

技术领域

[0001] 本公开内容一般地涉及半导体,并且更具体地,涉及实施地址转换并生成地址转换跟踪消息(trace message)的数据处理系统。

背景技术

[0002] 被称为 IEEE IST05001 的 IEEE 标准,或 Nexus 调试标准,是已建立的支持生成实时调试消息的实时调试标准。Nexus 调试标准给外部跟踪重构工具规定了用于在系统之内识别预定的操作条件的机制。调试过程同样使用于数据处理系统的代码开发中。实时地提供调试信息,没有侵扰数据处理系统的正常操作,这是非常希望的,以保持对系统操作的透明性。

[0003] 调试消息由数据处理系统生成,数据处理系统含有或者用于每个程序事件(程序跟踪消息发送)或者用于数据事件(数据读消息发送、数据写消息发送)的地址和数据信息,以及其它调试信息。地址信息典型地为虚拟地址信息,该虚拟地址信息是必须要转换以识别被称为物理地址的物理存储位置的格式。地址与正在执行的程序的关联是调试过程的重要部分,从而能够监控实际程序流程和系统数据变量的动态值。虚拟到物理地址的映射或转换必须执行。但是,外部调试器通常不具有必要的虚拟到物理的映射信息以快速地转换调试消息的地址部分,特别是当这些映射正由于请求调页或其它重映射操作而动态地改变时。虚拟地址的使用为调试用户提供了一个经由使用虚拟地址用于软件应用的文本和数据部分的通过编译和连接功能来获得的程序列表来跟踪计算机程序的直通途经。用户通常不了解虚拟地址是如何由操作系统(OS)转换成物理地址。因此,需要进行跟踪以指示虚拟地址。不幸的是,某些程序列表无法获得,例如用于在运行时间执行的预编译的软件模块的程序列表。此类模块的实例是 OS 调用和库函数。在该实例中,没有程序列表是可获得的。因而,要正确地跟踪和解释所执行程序的部分是很困难的。虚拟地址正确转换成物理地址需要知道程序计数器在何处并且需要检查物理存储器以确定地址在何处。对于执行多个程序的系统,存储映射动态地变化并且随时在存储页面之间变换。在这种情况下,地址转换显著地妨碍了调试消息生成和解释的实时进行。

附图说明

[0004] 本发明以实例的方式示出并且不受附图所限制,在附图中相同的参考符号指示类似的元件。在附图中的元件出于简明和清晰起见来示出而不一定要按比例画出。

[0005] 图 1 以框图的形式示出了本发明的一种形式的具有调试消息生成的数据处理系统根据;

[0006] 图 2 以框图的形式示出了图 1 的地址转换逻辑的一种形式;

[0007] 图 3 以框图的形式示出了图 1 的调试电路的一种形式;

[0008] 图 4 以图表的形式示出了图 2 的示例性 MMU 更新寄存器;

[0009] 图 5 以图表的形式示出了根据现有技术的示例性的转换旁视缓冲器(TLB)的写条

目指令；

[0010] 图 6 以图表的形式示出了根据现有技术的示例性 TLB 无效地址指令；

[0011] 图 7 以图表的形式示出了根据现有技术的传统模式的示例性程序跟踪消息；

[0012] 图 8 以图表的形式示出了根据现有技术的历史模式的示例性程序跟踪消息；

[0013] 图 9 以图表的形式示出了根据现有技术的示例性数据跟踪消息；

[0014] 图 10 以图表的形式示出了根据本发明的一种实施例的具有字段压缩的示例性地址转换跟踪消息；

[0015] 图 11 以图表的形式示出了根据本发明的一种实施例的示例性地址转换跟踪消息；

[0016] 图 12 示出了根据现有技术的可与信息生成一起使用的地址压缩的一个实例；

[0017] 图 13 以图表的形式示出了根据本发明的一种实施例的示例性地址转换跟踪消息；

[0018] 图 14 以图表的形式示出了根据本发明的一种实施例的具有历史划界 (history delimitation) 的示例性地址变换跟踪消息；以及

[0019] 图 15 以图表的形式示出了根据本发明的一种实施例的具有历史划界的示例性程序关联跟踪消息。

具体实施方式

[0020] 许多数据处理系统通常使用地址转换逻辑，例如转换旁视缓冲器 (TLB) 来将虚拟地址映射到物理地址。在地址转换逻辑之内的条目（例如，TLB 条目）可以通过使用处理器指令或者由数据处理系统内的硬件来修改或无效化，数据处理系统通过执行软件执行所需要的地址转换表搜索和 TLB 更新（例如，通过执行硬件“查表”或相似类型的硬件搜索）动态地维持地址转换。根据本发明的一个方面，调试消息（例如，地址转换跟踪消息）响应于地址转换逻辑内的条目的修改或无效化而生成。这些地址转换跟踪消息然后能够提供给外部开发系统，用于提高调试性能。而且，在虚拟地址映射被改变的情况下，程序关联消息同样可以以适当的时序生成，以便提供关于分支历史信息 and 序列计数信息的更准确的信息。同样，根据本发明的另一方面，效率可以通过压缩调试消息的一个或多个字段，通过对调试消息进行字段压缩，或者在可能时通过合并多个消息（例如程序关联消息）来进一步提高。

[0021] 在图 1 中示出的是生成调试消息的数据处理系统 10。全局互连 12 在数据处理系统 10 之内。全局互连 12 的一种形式是系统总线。可以使用其它形式的互连，包括，例如，交叉开关、点对点连接以及光学或无线传输技术。总线接口单元 (BIU) 14 经由双向耦接与全局互连 12 耦接。双向耦接的一种形式是双向多导体总线，其中多导体总线在此以穿过导体的斜线来表示。BIU 14 与存储器管理单元 (MMU) 16 双向耦接。MMU 16 经由双向多导体总线耦接至全局控制电路 18 的第一输入 / 输出端子。全局控制电路 18 的第二输入 / 输出端子经由双向多导体总线耦接至取指单元 20 的第一输入 / 输出端子。取指单元 20 具有经由多导体总线与指令解码器 22 的输入耦接的输出。指令解码器 22 的输出耦接至执行单元 24 的输入。执行单元 24 的一种形式包括至少一个算术逻辑单元、至少一个浮点单元和至少一个乘法器电路。寄存器文件 (register files) 25 在执行单元 24 之内。指令解码器 22

的输入 / 输出端子耦接至全局控制电路 18 的第三输入 / 输出端子。执行单元 24 的第一输入 / 输出端子耦接至全局控制电路 18 的第四输入 / 输出端子。执行单元 24 和取指单元 20 也与 MMU 16 双向耦接。调试电路 26 具有与全局控制电路 18 的第五输入 / 输出端子耦接的输入 / 输出端子。加载 / 存储单元 28 具有与全局控制电路 18 的第六输入 / 输出端子双向耦接的第一输入 / 输出端子。加载 / 存储单元 28 具有与 BIU 14 的第一输入 / 输出端子耦接的第二输入 / 输出端子。加载 / 存储单元 28 具有与执行单元 24 的第二输入 / 输出端子耦接的第三输入 / 输出端子。BIU 14 的第二输入 / 输出端子耦接至全局控制电路 18 的第七输入 / 输出端子。加载 / 存储单元 28 的输出提供数据虚拟地址并且与 MMU 16 的第一输入和调试电路 26 的第一输入耦接。取指单元 20 的输出提供指令虚拟地址并且与 MMU 16 的第二输入和调试电路 26 的第二输入耦接。MMU 16 的第一输出提供数据物理地址并且与 BIU 14 的第一输入和调试电路 26 的第三输入耦接。MMU 16 的第二输出提供指令物理地址并且与 BIU 14 的第二输入和调试电路 26 的第四输入耦接。

[0022] 存储器 30 经由双向耦接与全局互连 12 耦接。调试电路 26 具有经由双向多导体与多个调试端子 40 耦接的第二输入 / 输出端子。该多个调试端子 40 耦接至外部开发系统 36, 外部开发系统 36 通常称为调试器或外部调试器。在所示出的形式中, BIU 14、MMU 16、全局控制电路 18、取指单元 20、指令解码器 22、具有寄存器文件 25 的执行单元 24、调试电路 26 和加载 / 存储单元 28 共同形成了数据处理器 42, 如图 1 中的虚线框组所示。虽然全局控制电路 18 在图 1 中被示出于分离的位置内, 但是应当很好理解的是, 全局控制电路 18 的电路和功能控制还可以按照分布的方式来实现并且包含于数据处理系统 10 的其它各种系统块的任一种之内。此外, 在所示出的实施例中, 全局控制 18 包括进程标识符 (PID) 寄存器 19, 用于存储当前正执行的进程的进程标识符 (PID)。全局控制 18 还将该 PID 提供给 MMU 16。

[0023] 在操作中, 数据处理系统 10 经由全局互连 12 与设备 (没有示出) 通信。与数据处理器 42 通信的信息通过 BIU 14 来传递。取指单元 20 在全局控制电路 18 的控制之下从 BIU 14 中检索出数据处理指令 (即, 处理器指令)。所检出的指令在全局控制电路 18 的控制之下顺序地与用于解码的指令解码器 22 通信。执行单元 24 执行指令并且生成数据, 该数据经由通过全局控制电路 18、BIU 14 和全局互连 12 的耦接或者存储在高速缓存 (cache) (没有示出) 内或者放置于存储器 30 中。数据处理器 42 和数据处理系统 10 的操作的调试使用调试电路 26 来执行, 调试电路 26 生成由外部开发系统 36 进行分析的调试消息。响应于此类来自外部开发系统 36 的激活而进入操作的测试或调试模式。在所示出的形式中, 调试电路 26 被配置用于接收数据地址和指令地址, 其中这些地址可以是虚拟地址或物理地址。数据地址是数据驻留于其中的地址, 而指令地址是指令驻留于其中的地址。指令虚拟地址由取指单元 20 提供给调试电路 26 和 MMU 16。虚拟地址是未转换的地址, 它需要某种进一步的处理或转换以获得信息驻留于其中的物理存储位置的转换地址。该转换地址称为物理地址。MMU 16 将指令物理地址提供给 BIU 14 和调试电路 26。在一种形式中, 虚拟或未转换的地址可以是逻辑地址。在另一种形式中, 未转换地址可以是有效地址。有效地址在转换成物理地址之前必须首先转换成虚拟地址。加载 / 存储单元 28 将数据虚拟地址提供给调试电路 26 和 MMU 16。MMU 16 将数据物理地址提供给 BIU 14 和调试电路 26。

[0024] 调试电路 26 然后使用一个或多个所接收地址的至少一部分来形成用于外部开发

系统 36 的调试消息,这将在下面更详细地讨论。调试消息的格式可以改变并且其实例将在下面结合图 7、8、9、10、11、13、14 和 15 来讨论。

[0025] 在图 2 中示出的是图 1 的 MMU 16 的一部分的示例性实施例。MMU 16 包括控制电路 203、MMU 更新寄存器 204 和地址转换逻辑 205。在一种形式中,地址转换逻辑 205 被实现为具有编号为 0 到 N 的 N+1 个条目的转换旁视缓冲器 (TLB),其中每个条目都包括虚拟地址 209、对应的物理地址 211、对应的转换大小 (TSIZ) 210、对应的转换 ID (TID) 212、对应的属性 213 以及对应的有效性字段 215。在可选实施例中,地址转换逻辑 205 可以以不同方式实现。控制电路 203 与 MMU 更新寄存器 204 双向耦接并且将修改指示符和地址转换信息提供给调试电路 26 和地址转换逻辑 205。控制 203 还与全局控制 18 双向耦接并且从全局控制 18 接收 PID。MMU 更新寄存器 204 可以包括一个或多个寄存器并且与执行单元 24 双向耦接。

[0026] 在操作中,条目 0 到 N 存储用于将虚拟地址转换到物理地址的地址映射信息。例如,指令或数据虚拟地址被提供给 MMU 16 (例如,来自取指单元 20 或加载 / 存储单元 28),在此它与存储于地址转换逻辑 205 内的虚拟地址 209 进行比较。如果找到匹配条目 (其中匹配条目也是有效条目,如有效性字段 215 所示),则将对应的物理地址 211 作为该指令或数据的物理地址提供给调试电路 26 和 BIU 14。匹配条目的对应 TSIZ 210 提供与转换地址对应的页面大小并且匹配条目的对应 ITD 212 提供地址空间标识符。当 TID 字段为 0 时,匹配条目应用于所有进程,因为该条目是全局的。但是,当 TID 212 为非零时,匹配条目只有当所接收的虚拟地址和所接收的 PID 两者分别与地址转换逻辑 205 中的有效条目的虚拟地址 209 和 ITD 212 匹配时才确定。此外,通过使用 MMU 更新寄存器 204,控制 203 能够响应于由处理器 42 执行的处理器指令来更新或修改地址转换逻辑 205 内的条目,这将在下面参照图 4 更详细地讨论。当控制 203 修改或更新地址转换逻辑 205 之内的条目时,修改指示符被提供给地址转换逻辑 205,以及给调试电路 26。

[0027] 在图 3 中示出的是图 1 的调试电路 26 的一部分的示例性实施例。调试电路 26 包括调试控制电路 60、消息生成逻辑 64 和输入 / 输出 (I/O) 逻辑 66。消息生成逻辑 64 包括压缩逻辑 68 并且被耦接以从 MMU 16 内的控制 203 接收修改指示符和地址转换信息。调试控制电路 60 包括消息先进先出存储电路 (FIFO) 70 并且与消息生成逻辑 64 和 I/O 逻辑 66 双向耦接。调试控制电路 60 还从全局控制电路 18 接收信息。I/O 逻辑 66 与调试端子 40 双向耦接。

[0028] 在操作中,消息生成逻辑 64 是实现按照多种预定格式中所选择的一种形成调试消息的功能的逻辑电路。消息生成逻辑 64 形成调试消息,其中压缩逻辑 68 可以对部分或全部消息执行消息压缩,并且这些调试消息然后被提供给调试控制电路 60,在此这些调试消息存储于消息 FIFO 70 内。调试消息经由 I/O 逻辑 66 从消息 FIFO 70 路由至调试端子 40。在一种实施例中,每当地址转换逻辑 205 中的条目被修改或被更新时,消息生成逻辑 64 生成地址转换跟踪消息。例如,每当控制 203 断言 (assert) 了向地址转换逻辑 205 和消息生成逻辑 64 两者表明在地址转换逻辑 205 中的条目正在被修改的修改指示符时,消息生成逻辑 64 就基于从控制 203 接收到的地址转换信息来生成地址转换跟踪消息。注意,在此所使用的条目修改也包括条目的无效。这些地址转换跟踪消息的格式将在下面参照图 10、11、13 和 14 更详细地描述。

[0029] 在图 4 中示出的是 MMU 更新寄存器 204 的一个实例。在所示出的实施例中, MMU 更新寄存器 204 包括 4 个寄存器: MA0、MA1、MA2 和 MA3。这些寄存器被用来更新地址转换逻辑 205 内的条目。在所示出的实施例中, 这些寄存器每个都是包括多个字段的 32 位寄存器。MA1 包括 2 位 TLBSEL 字段和 6 位 ESEL 字段。TLBSEL 字段存储指示哪个 TLB 将要被更新的值。在所示出的实施例中, 仅示出了一个 TLB (地址转换逻辑 205), 在该 TLB 中 TLBSEL 字段可以不存在; 但是, 在可选实施例中, 可以实现任何数量的 TLB。ESEL 字段存储指示 TLB 的条目号的值 (例如, 在地址转换逻辑 205 中的条目 0 到 N 之一)。MA1 包括 8 位 TID 字段和 5 位 TSIZE 字段。TID 字段存储指示转换标识符字段的值并且 TSIZE 字段存储指示页面大小的值。MA2 包括 22 位字段 VPN 字段, 其存储指示虚拟页面号的值。MA2 还包括存储属性的各种字段, 属性例如 VLE (页面正使用可变长度编码用于指令)、W (页面是直写的 (Writethrough))、I (页面是高速缓存禁用的)、M (页面“要求存储器相关性 (memory coherence required)”)、G (页面是受保护的 (Guarded)) 及 E (页面字节序 (page Endianness))。MA3 包括 22 位字段 PPN 字段, 其存储指示物理页面号的值并且存储多个属性 (例如, U0-U3 (用户定义属性)、UX、SX、UW、SW、UR 和 SR (用户 (U) 和超级用户 (S) 的读取 (R)、写入 (W) 和执行 (X) 许可))。注意, 在可选实施例中, MMU 更新寄存器 204 可以包括比所示出的那些寄存器更多的寄存器, 并且存储于其中的信息可以使用任何数量的寄存器按多种不同的格式来组织。此外, 每个字段可以具有存储信息的指拨 (the appropriate of information) 所需的更多或更少的位。注意, 可以将 TID 看作是对虚拟地址的扩展, 从而虚拟地址能够作为“与 TID 连串的 VPN”来计算。因此, 这些寄存器可以由全局控制 18 响应于处理器 42 的一个或多个处理器指令 (例如, 移动至专用寄存器指令) 来更新。然后, 存储于这些寄存器内的信息被用来响应于由处理器 42 所执行的处理器指令来更新地址转换逻辑 205 中的条目。在本发明的可选实施例中, 对地址转换逻辑 205 (例如, TLB) 的更新可以由控制逻辑响应于 TLB 未命中来执行。这种控制逻辑可以实现为控制逻辑 203 的一部分或者可以实现于 MMU 16 或处理器 42 内的其它地方。在一种实施例中, 这种对 TLB 未命中响应的控制逻辑通过搜索存储于存储器 30 内的或者在数据处理系统 10 之内的其它地方的一个或多个转换表自动获得地址转换条目信息, 并且随后基于从地址转换表中检索出的信息执行地址转换逻辑 205 内的地址转换修改, 不使用由处理器 42 显式执行的处理器指令。

[0030] 例如, 图 5 示出了当前存在的处理器指令的一个实例, TLB 写条目 (TLBWE) 指令, 该 TLB 写条目指令可以用来更新 TLB 条目 (即, 在地址转换逻辑 205 内的条目)。TLBWE 指令促使在 MMU 更新寄存器 204 内的某些字段的内容被写入地址转换逻辑 205 中的单个条目之内。被写的条目 (tlb_entry_id) 例如由 MA0 的 TLBSEL 和 ESEL 字段指定。一旦执行 TLBWE 指令, 该所识别条目 MA1、MA2 和 MA3 中的适当信息来更新。也即, 在 MA1 的 TID 字段和 TSIZE 字段中的值被存储于地址转换逻辑 205 的所识别条目的 TSIZ 字段 210 和 TID 字段 212 之内。类似地, 在 MA2 中的 VPN 的值, 以及 PPN 的值和 MA3 的属性被存储到所识别条目的对应的字段虚拟地址 209、物理地址 211 和属性 213 之内。此外, 所识别条目的有效位 215 被置位以指示有效条目。

[0031] 图 6 示出了当前存在的处理器指令的另一个实例, TLB 无效化条目 (TLB_INV) 指令, 该 TLB 条目无效化指令可以被用来无效地址转换逻辑 205 的条目。TLB_INV 指令格式归类 (species) 两个源寄存器, RA 和 RB, 它们在 TLB_INV 指令执行时被用来计算 RA+RB 的有

效地址 (EA)。也就是, EA 等于 RA 的内容加上 RB 的内容。该 EA 地址然后被用来在地址转换逻辑 205 内找出匹配条目, 并且一旦找到匹配条目, 该匹配条目的有效位 215 被置位以指示无效条目。源寄存器 RA 和 RB 可以是位于例如处理器 42 的寄存器文件 25 之内的任何两个通用寄存器。

[0032] 在本发明的一种实施例中, 当地址转换逻辑 205 的条目被更新时, 例如每当或者 TLBWE 或者 TLB_INV 指令被执行时, 控制 203 经由修改指示符通知调试电路 26 并且, 响应于此, 在调试电路 26 中的消息生成逻辑 64 生成可以 (经由 I/O 逻辑 66, 由 FIFO 70) 提供给调试端子 40 的地址转换跟踪消息。以这种方式, 调试器 26 不需要显式地 (explicitly) 从地址转换逻辑 205 请求地址转换信息, 因为该信息在修改发生时被自动发送。基于 TLBWE 的地址转换跟踪消息和基于 TLB_INV 的地址转换跟踪消息的实例将参照图 10、11 和 13 来描述。此外, 当 TLBWE 或 TLB_INV 指令被执行时, 控制 203, 使用存储于以上所描述的 MMU 更新寄存器 204 中的信息, 将适当的控制和信息提供给地址转换逻辑 205 以适当地更新所识别的或匹配的条目。因此, 注意, 除了修改指示符之外, 在需要时, 控制 203 可以将其它信息连同修改指示符一起提供给地址转换逻辑 205 和调试电路 26。在可选实施例中, 其它更新机制可以用来促使地址转换逻辑 205 之内的条目被修改, 并且响应于此, 控制 203 能够提供适当的信令给调试电路 26 以指示修改以及与所需修改相关的信息, 从而允许调试电路 26 正确地生成地址转换跟踪消息。

[0033] 在图 7、8 和 9 中示出的是由当前可获得的调试逻辑生成的跟踪消息。在图 7 中示出的是传统模式中的程序跟踪间接分支消息 80。在图 8 中示出的是历史模式中的程序跟踪间接分支消息 81。当在处理器 42 上执行的指令执行间接分支时, 程序跟踪间接分支消息被生成。间接分支是其目标地址不直接提供于分支指令内的, 而是存储于另一个位置 (例如, 寄存器) 中, 或者需要另外动态计算或确定 (例如, 用于子例程调用) 的分支。直接分支是其目标与指令一起直接提供的分支, 例如具有与特定的地址位置对应的或者由指令中的偏移值提供的标签, 并且因而典型地是能够由调试器基于对程序指令值的了解来确定的静态值。在图 9 中示出的是数据跟踪消息 82。

[0034] 参照图 7, 跟踪消息 80 具有四个示出字段。虚拟地址字段含有跟踪消息 80 的虚拟地址。序列计数字段含有表示自上一程序跟踪消息以来已经执行的指令数的值。源处理器字段含有识别跟踪消息 80 所关联的是哪个处理器值。该字段信息在多处理器系统中是有价值的。转移代码字段含有将跟踪消息 80 识别为具有特定的预定字段格式的程序跟踪间接分支消息的值。在图 7 的实例中, 表示具有所示形式的程序跟踪消息的值是“000100”, 指示程序跟踪间接分支消息使用传统的分支跟踪模式操作来生成。

[0035] 参照图 8, 跟踪消息 81 具有五个示出字段。虚拟地址字段含有跟踪消息 81 的虚拟地址。分支历史字段含有多个 1 位值, 其中, 对于自上一程序跟踪消息以来所采用的每个直接分支, 对应的 1 位值被置位或被清除以指示究竟是采用直接分支还是不采用。序列计数字段含有表示自上一程序跟踪消息以来已经执行的指令数的值。源处理器字段含有识别跟踪消息 80 所关联的是哪个处理器的值。该字段信息在多处理器系统中是有价值的。转移代码字段含有将跟踪消息 80 识别为具有特定的预定字段格式的程序跟踪间接分支消息的值。在图 8 的实例中, 表示具有所示形式的程序跟踪消息的值是“011100”, 指示程序跟踪间接分支消息是使用历史模式分支跟踪操作来生成的。

[0036] 参照图 9,跟踪消息 82 具有五个示出字段。(一个或多个)数据值字段含有与跟踪消息 82 关联的一个或多个数据值。虚拟地址字段含有跟踪消息 82 的虚拟地址。数据大小字段含有表示数据的长度的或者数据字所含有的位数的值。源处理器字段含有识别跟踪消息 82 所关联的是哪个处理器的值。该字段信息在多处理器系统中是有价值的。转移代码字段含有将跟踪消息 80 识别为具有特定的预定字段格式的数据跟踪消息的值。在图 9 的实例中,表示具有所示形式的数据写入跟踪消息的值为“000101”。表示具有所示形式的数据读跟踪消息的值是“000110”。

[0037] 图 10 和 11 分别示出了根据本发明的实施例的地址转换跟踪消息 84 和 86。响应于 TLBWE 指令的执行,如上所述,调试消息(例如,地址转换跟踪消息 84 或者地址转换跟踪消息 86)可以由调试器 26 生成。因此,注意,这些地址转换跟踪消息可以称为基于 TLBWE 的地址转换跟踪消息。注意,地址转换跟踪消息 84 是基于 TLBWE 的地址转换跟踪消息的字段压缩版本,而地址转换跟踪消息 86 是非字段压缩版本。

[0038] 首先参照图 11,跟踪消息 86 具有八个示出字段。字段的定位是任意的。压缩的物理地址字段含有跟踪消息 86 的压缩形式的物理地址。也就是,该字段表示(地址转换逻辑 205 的)TLB 的触发了跟踪消息的生成的所修改条目的物理页面号地址的压缩版本。压缩的虚拟地址字段含有跟踪消息 86 的压缩形式的虚拟地址。也就是,该字段表示所修改条目的虚拟页面号地址的压缩版本。TID 字段含有表示所修改条目的 TID 值的值。TSIZ 字段含有表示所修改条目的 TSIZ 值的值。源处理器字段含有识别跟踪消息 86 所关联的是哪个处理器的值。该字段信息在多处理器系统中是有价值的。转移代码(TCODE)字段含有将跟踪消息 84 识别为具有特定的预定字段格式的地址转换跟踪消息的值。在图 10 的实例中,表示地址转换跟踪消息的值是“100001”。事件代码(ECODE)字段含有进一步识别预定的字段格式的值。也就是,它可以用来在不同类型的地址转换跟踪消息之间进一步区分。在图 11 的实例中,表示具有所示形式的非字段压缩的基于 TLBWE 的地址转换跟踪消息的值是“1101”。注意,在图 11 的实例中,物理地址和虚拟地址字段每个都含有压缩值。一种压缩方法将在下面参照图 12 来描述。但是,注意,在可选实施例中,这些字段中只有一个可以压缩,或者这些字段都不可以压缩。此外,在需要时,对每个字段都可以使用任何的位数,并且消息可以包括比所示出的更多或更少的信息。

[0039] 回过来参照图 10,图 10 是图 11 的地址转换跟踪消息 86 的字段压缩版本(具有五个示出字段)。也就是,注意,对于地址转换跟踪消息 84,TID 和 TSIZ 字段没有作为调试消息的一部分包含于其中,并且因而没有经由调试端子 40 来传输。而且,在图 10 的实例中,ECODE 字段的值不同于图 11 的值,因为在图 10 中值“1100”表示具有所示形式的字段压缩的基于 TLBWE 的地址转换跟踪消息。注意,针对地址转换跟踪消息 84 中的剩余字段的描述与以上针对地址转换跟踪消息 86 所提供的那些描述是相同的。因此,注意,在字段压缩模式中,地址转换消息通过不包含所选择字段的方式来压缩。(在一种实施例中,该字段压缩可以由压缩逻辑 68 执行)。

[0040] 在一种实施例中,对于字段压缩的地址转换跟踪消息,这些含有预定的普遍值(prevalent value)的字段是不包含于消息中的字段。例如,在一种实施例中,TID 和 TSIZ 字段两者都倾向于具有普遍值,并且因而不需要在每个地址转换跟踪消息中被传输。例如,在一种实施例中,TID 值 00000000 指示地址转换条目对所有进程 ID 值都将是可获得的,而

不是仅限于匹配单个进程 ID(PID) 值。在本实施例中,可以将全零的 TID 值看作是普遍的 TID 值。在可选实施例中,表示具有预定的非零 TID 值的主导进程 (predominant process) 的不同 TID 值可以定义为是普遍值。在一种实施例中,对虚拟和物理页面的特定的页面大小可以是普遍的。例如,在许多系统中,4 千字节 (4KB) 的页面大小是主导的页面大小值,并且从而可以定义为预定的普遍值,因为大部分地址转换条目都将使这样的值在所存储的 TLB 条目的 TSIZ 字段中编码。对于这些常见的情形,字段压缩允许减小必须传输到外部开发系统 36 的最大消息长度。该减少可以提高调试消息的带宽,并且还可以有利于优化调试控制电路 60 的消息 FIFO 70 的大小。在一种实施例中,将要从消息中去除的或者不传输给外部开发系统 36 的字段的预定的普遍值由系统 10 的用户所控制。也就是,该值能够存储于用户可编程的存储位置内,例如,在 MMU 更新寄存器 204 之内。

[0041] 图 12 示出了一种用于压缩调试消息的特定字段,例如消息的物理和虚拟地址字段,以获得例如图 10 和 11 的压缩的物理地址和压缩的虚拟地址的方法。(在一种实施例中,这种压缩可以由压缩逻辑 68 来执行)。图中提供了标记为 A1 和 A2 的两个地址。地址 A2 是用于生成调试消息的当前地址,例如将要包含于地址转换跟踪消息中的物理或虚拟地址。地址 A1 是用于生成前一调试消息的前一地址,例如分别为包括于前一地址转换跟踪消息内的前一物理或虚拟地址。地址 A1 和 A2 的实际值只是示例性的,并且应当理解,任何地址值都可以使用。所修改地址通过在地址 A1 和 A2 上执行逻辑操作来产生。在一种实施例中,逻辑操作是异或操作,在该异或操作中地址 A1 和地址 A2 的各自对应的位位置进行了异或操作。所产生的修改地址被示出于图 12 中,在该修改地址中生成了 20 个前导零。地址中从最低有效位到最高有效二进制 1 位的部分被分组,如图 12 所示,并且这种分组形成了用作调试消息中的地址(例如,压缩的物理地址或压缩的虚拟地址)的修改地址 M1,导致了许多消息的更小的平均消息大小。能够由之前发送的消息地址重新生成的冗余信息被去除。相反地,地址 A2 可以由调试器根据调试消息地址按下面的方式重新生成。前一地址 A1 与地址消息 M1 进行异或操作。通过在地址消息 M1 前面添加所需的前沿零使地址消息 M1 扩展回到 32 位的格式。异或操作产生了地址 A2。因而,图 12 示出了如何实现从地址转换成调试消息地址以及从调试消息地址转换成地址。注意,当调试第一次启用时在第一跟踪消息中发送的修改地址 M1 含有去除了前导零的完全地址。调试器能够假定全零的前一地址值作为初始地址,作为将修改地址 M1 扩展到用于消息的完全地址的基础。

[0042] 图 13 示出了根据本发明的一种实施例的地址转换跟踪消息 88。响应于 TLB_INV 指令的执行,如上所述,诸如地址转换跟踪消息 88 的调试消息可以由调试器 26 生成。因此,注意,地址转换跟踪消息 88 可以称为基于 TLB_INV 的地址转换跟踪消息。跟踪消息 88 具有四个示出字段。字段的定位是任意的。TLB_INV 虚拟地址字段含有跟踪消息 88 的未压缩的虚拟地址。也就是,该字段含有(如以上所描述的)被用来在地址转换逻辑 205 中找出匹配条目以便使它无效化的有效地址(使用在 RA 和 RB 中的值算出的)的计算值。(作为选择,注意,TLB_INV 虚拟地址也可以被压缩)。源处理器字段含有识别跟踪消息 88 所关联的是哪个处理器的值。该字段信息在多处处理器系统中是有价值的。转移代码(TCODE)字段含有将跟踪消息 88 识别为具有特定的预定字段格式的地址转换跟踪消息的值。在图 13 的实例中,表示地址跟踪消息的值是“100001”。事件代码(ECODE)字段含有进一步识别预定的字段格式的值。也就是,它可以用来在不同类型的地址转换跟踪消息之间进一步区分。

在图 13 的实例中,表示具有所示形式的基于 TLB_INV 的地址转换跟踪消息的值是“1110”。注意,在需要时,可以将任何位数用于每个字段并且消息可以包括比所示出的更多或更少的信息。

[0043] 如同以上参照图 8 所讨论的,分支历史字段对于用于不生成程序跟踪消息的具体采用分支(例如,用于直接分支)的地址转换提供更好的认识是有用的。但是,当虚拟存储映射的改变在生成程序跟踪消息的那些执行分支(例如,间接分支)之间发生时,分支历史字段在超出映射的改变发生的时刻之外不再提供准确的信息。因此,在一种实施例中,可以生成包括分支历史信息直到虚拟地址映射改变时的程序关联消息。也就是,程序关联消息能够在地址转换跟踪消息响应于地址转换逻辑 205 中的条目的修改而生成时生成。例如,程序关联消息能够在基于 TLBWE 的地址转换消息生成时生成。此外,该程序关联消息还能够关于该基于 TLBWE 的地址转换消息正确地排序,以允许调试器(例如,外部开发系统 36)于分支历史被累积时在映射就绪的情况下执行程序跟踪重构。以这种方式,程序关联消息能够给自上一程序跟踪消息生成以来直到地址转换逻辑 205 中的条目由 TLBWE 指令修改时所执行的指令正确地提供分支历史信息 and 指令计数信息(即,序列计数信息)。在一种实施例中,与地址转换跟踪消息对应的程序关联消息在地址转换消息之前提供。

[0044] 图 15 示出了具有可以被生成并且针对响应于地址转换逻辑 205 中的条目修改而生成的地址转移跟踪消息正确地排序的分支历史划界的程序关联消息 92。程序关联消息 92 具有五个示出字段。字段的定位是任意的。分支历史字段含有多个 1 位值,其中,对于自上一程序跟踪消息以来所采用的每个直接分支,对应的 1 位值被置位或被清除以指示究竟是要采用直接分支还是不采用。该分支历史字段含有这种用于所采用的每个直接分支的信息,直到地址转换逻辑 205 中的条目被修改并且地址转换跟踪消息(例如,基于 TLBWE 的地址转换跟踪消息 84 或 86)被生成时。序列计数字段(也称为指令计数字段)含有表示自上一程序跟踪消息以来直到地址转换逻辑 205 中的条目被修改时已经执行的指令数的值。源处理器字段含有识别程序关联消息 92 所关联的是哪个处理器的值。该字段信息在多处理器系统中是有价值的。转移代码字段含有将程序关联消息 92 识别为具有预定字段格式的历史划界的程序关联消息的值。在图 15 的实例中,表示具有所示形式的程序跟踪消息的值是“100001”。事件代码(ECODE)字段含有进一步识别程序关联消息 92 的预定字段格式的值。也就是,它可以被用来在不同类型的程序关联跟踪消息之间进一步区分。例如,事件代码可以用来表示程序关联消息是否是合并的消息。(合并的消息将在下面更详细地描述)。在图 15 的实例中,ECODE 字段的值表示程序关联消息不是合并的,意味着它仅由一个触发源(例如,在地址转换逻辑 205 中的条目的修改)引起。注意,在需要时,可以将任何位数用于每个字段并且消息可以包括比所示出的更多或更少的信息。(注意,在另一可选实施例中,分支历史字段或指令计数字段中只有一个可以包含于程序关联消息中)。

[0045] 图 14 示出了具有历史划界的地址转换跟踪消息 90。也就是,在图 14 的实例中,响应于地址转换逻辑 205 的条目被修改,不是生成地址转换跟踪消息(例如,基于 TLBWE 的地址转换跟踪消息)和程序关联消息两者,可以生成进一步包括分支历史消息和指令计数的单个地址转换跟踪消息。也就是,基于 TLBWE 的地址转换跟踪消息(例如,跟踪消息 84 或 86)的信息还可以包括分支历史信息或指令计数信息或两者,直到地址转换逻辑 205 的条目被修改时。在一种实施例中,地址转换跟踪消息 90 包括字段压缩的基于 TLBWE 的地址

转换跟踪消息（例如，转换消息 84）的字段，以便减小消息大小。该大小缩减可能是有必要的，例如，使得生成消息能够恰当地适合于 FIFO 70 之内。也就是，TSIZ 和 TID 字段可以不包括在这种类型的具有历史划界的地址转换跟踪消息中。

[0046] 在图 14 所示的实施例中，跟踪消息 90 具有七个示出字段。字段的定位是任意的。压缩的物理地址字段含有跟踪消息 90 的压缩形式的物理地址。也就是，该字段表示（地址转换逻辑 205 的）触发了跟踪消息的生成的 TLB 的所修改条目的物理地址的压缩版本。所压缩的虚拟地址字段含有跟踪消息 09 的压缩形式的虚拟地址。也就是，该字段表示所修改条目的虚拟地址的压缩版本。分支历史字段含有多个 1 位值，其中，对于自上一程序跟踪消息以来所采用的每个直接分支，对应的 1 位值被置位或被清除以指示究竟是要采用直接分支还是不采用。该分支历史字段含有这种用于所采用的每个字节分支的消息，直到在地址转换逻辑 205 中的条目被修改时，从而触发跟踪消息 90 的生成时。序列计数字段含有表示自上一程序跟踪消息以来直到地址转换逻辑 205 中的条目被修改时已经执行的指令数的值。源处理器字段含有识别跟踪消息 90 所关联的是哪个处理器的值。该字段消息在多处处理器系统中是有价值的。转移代码 (TCODE) 字段含有将跟踪消息 90 识别为具有特定的预定字段格式的地址转换跟踪消息的值。在图 14 的实例中，表示地址转换跟踪消息的值是“100001”。事件代码 (ECODE) 字段含有进一步识别预定的字段格式的值。也就是，它可以用来在不同类型的地址转换跟踪消息之间进一步区分。在图 14 的实例中，ECODE 字段的值表示具有所示格式的历史划界的地址转换跟踪消息。注意，在需要时，可以将任何位数用于每个字段并且消息可以包括比所示出的更多或更少的信息。因此，注意，在生成了具有历史划界的跟踪消息（例如，跟踪消息 90）的情况下，分开的程序关联消息（例如，程序关联消息 92）不需要被生成。

[0047] 在一种实施例中，是否在地址转换逻辑 205 中映射改变发生时提供具有历史划界 92 的程序关联消息和地址转换跟踪消息或者是否提供具有历史划界 90 的单个地址转换跟踪消息的确定可以基于是否能够发生字段压缩以限制消息的最大尺寸。如果普遍值存在于地址转换修改中，则在一种实施例中，生成具有历史划界 90 的单个地址转换跟踪消息。如果普遍值不存在，这需要将 TSIZ 和 / 或 TID 信息提供给外部开发系统 36，则可以作出确定以便发送具有历史划界的程序关联消息（例如，消息 92）和地址转换跟踪消息（例如，消息 86）两者。在一种实施例中，不是字段压缩的具有历史划界的单个地址转换跟踪消息将超过在消息 FIFO 70 中的条目的大小，并且从而要求比最佳 FIFO 更大。通过基于消息压缩是否不足以允许具有历史划界的单个地址转换跟踪消息适应于预定的消息位长之内来选择性地生成具有历史划界消息 92 的程序关联消息，可以执行在消息 FIFO 70 中的条目的宽度优化，因为在某些实施例中，大部分的消息需要比地址转换跟踪消息 86 和 90 更少的位，并且在给出的条目中 FIFO 70 的许多存储容量是未用的。更好的优化可以是增加在 FIFO 70 中的条目数，同时使条目的宽度变窄。注意，在某些实施例中，关于是否由于含有预定的普遍值而可以从地址转换跟踪消息中去除一个或多个字段的确定可以被用来在具有历史划界的单个地址转换跟踪消息，或具有历史划界的程序关联消息和不具有历史划界的地址转换跟踪消息两者之间选择。在其它实施例中，附加的因素，例如压缩的虚拟地址和压缩的物理地址字段之一或两者的压缩程度可以用于确定最佳消息发送判定策略。

[0048] 在一种实施例中，（除了由地址转换逻辑 205 的条目的修改所引起的虚拟存储映

射的改变之外)附加的触发事件可以导致直到触发事件时的对提供分支历史和序列计数(即指令计数)的程序关联消息(PCM)的需求。在一种实施例中,处理器 42 可以能够透明地执行具有独特的二进制编码的多个指令集,例如,标准定长指令集和交替可变长度编码(VLE)指令集。在一种实施例中,正执行的指令集在指令页面之内是恒定的,但是在不同的存储器页面中可以不同,并且从而由外部调试器对在物理存储器中所存储的二进制值的解释取决于准确地获知哪个指令集存在于给出的页面中。例如,当处理器 42 跨过页面边界,导致执行模式切换进 VLE 指令序列或者从 VLE 指令序列中切换出来时,PCM 被生成,这在两种操作模式之间有效地中断了任何正在运行的指令计数和历史信息,从而需要程序关联消息以提供分支历史和序列计数,直到执行模式切换发生时。此外,在另一个实例中,当使用历史模式中的程序跟踪时(例如,在生成像跟踪消息 81 的跟踪消息时),当直接分支导致执行模式切换进 VLE 指令或者从 VLE 指令中切换出来时,PCM 也被生成。除了这些 PCM 类型的触发器之外,当程序跟踪消息发送由于达到预定的指令屏蔽(instruction mask)而变成被屏蔽时,或者当某些预定的处理器 42 异常或在数据处理系统 10 之内的其它系统事件发生时会发生其它例子。在这些情况下,不是由于这些可能发生的事件中的每个事件而生成程序关联消息(例如,当在跨过页面边界导致执行模式切换的同时虚拟映射的改变发生时),而是能够生成表示两个触发事件的单个程序关联消息。在该实例中,能够生成诸如图 15 的程序关联消息 92 的格式那样的格式,其中 ECODE 字段的值可以被用来指示它是合并的消息(即,由于多重触发事件而生成的消息)。在这种情况下,不同的 ECODE 值能够简单地指示程序关联消息是否是合并的程序关联消息,或者可以进一步指示,如果是合并的,是哪种触发器促使生成该程序关联消息。

[0049] 至此应当理解,本文已经提供了具有有效的实时调试寻址的数据处理系统。通过响应于地址转换逻辑 205 中的条目修改或无效化而生成地址转换跟踪消息,调试电路 26 不需要显式请求这种类型的地址转换信息。在一种实施例中,条目的一个或多个字段中的任一字段的任何修改是足够导致生成地址转换跟踪消息的。以这种方式,可以在使用地址转换的系统中,例如在使用一个或多个 TLB 来转换地址的系统中执行改进的调试。而且,在虚拟地址映射被改变的情况下,程序关联消息可以按适当的时序来生成,使得针对分支历史消息和序列计数信息提供更准确的信息。此外,效率可以通过压缩调试消息的一个或多个字段,通过对调试消息进行字段压缩,或者在可能时通过合并多个消息(例如程序关联消息)来进一步提高。地址转换映射的改变可以由用于修改 TLB 内容的控制指令的处理器指令执行引起,或者在可选实施例中可以由存储器管理单元硬件引起,存储器管理单元硬件自发地通过转换表搜索,例如通过执行查表以在发生 TLB 未命中时获得新的转换来维持 TLB。

[0050] 因为实现本发明的各种装置大部分包括本领域技术人员所知道的电子零件和电路,因而除了以上所描述的被认为是必要的电路细节外不再作更多的解释,以便于本发明的基础概念的理解和掌握,并且为了不混淆本发明的教导或分散对其的注意力。

[0051] 在适用时,以上实施例的某一些可以使用多种不同的信息处理系统来实现。例如,尽管图 1 及其讨论描述了示例性的存储系统的体系结构,但是该示例性的体系结构仅为了在讨论本发明的各个方面时提供有用的参考而给出。当然,关于该体系结构的描述出于讨论起见已经被简化,并且它只是可以按照本发明来使用的许多不同类型的适合的体系结构

中的一个。本领域技术人员应当意识到,在逻辑块之间的界限只是例示性的,并且可选实施例可以合并逻辑块或电路元件或者将可选的功能分解强加于各种逻辑块或电路元件之上。

[0052] 因而,应当理解,在此所示出的体系结构只是示例性的,并且实际上能够实现许多实现了相同功能的其它体系结构。在抽象的但仍然明确的意义,实现相同功能的任何零件布局都是有效地“关联的”,从而实现了所期望的功能。因此,在此结合以实现特定功能的任何两个零件能够被看作是彼此“关联的”,从而实现所期望的功能,不考虑体系结构或中间零件。类似地,这样关联的任何两个零件也能够被看作是彼此“可操作地连接的”,或者“可操作地耦接的”,以实现所期望的功能。

[0053] 而且,本领域技术人员应当意识到,在以上所描述的操作的功能之间的界限只是例示性的。多个操作的功能可以结合成单个操作,和 / 或单个操作的功能可以分布在附加的操作中。而且,可选实施例可以包括特定操作的多个实例,并且操作的顺序在各种其它实施例中可以改变。

[0054] 虽然本发明在此参照具体的实施例来描述,但是在不脱离下面的权利要求书所阐述的本发明的范围的情况下能够进行各种修改和改变。例如,可以使用任何数量的集成电路芯片。因此,说明书和附图将被看作是说明性的而不是限制性的,并且所有此类修改旨在包含于本发明的范围之内。在此针对具体实施例描述的任何优点、优势或问题的解决方案并不应当看作是任何或全部权利要求的关键的、必需的或基本的特征或元素。

[0055] 在此所使用的词语“耦接”并不必限制于直接耦接或机械耦接。

[0056] 而且,在此所使用的词语“一 (a)”或“一个 (an)”被定义为一个或多个。此外,诸如“至少一个”和“一个或多个”的引入性短语在权利要求中的使用不应当看作暗示着:另一权利要求的元件由不定冠词“一 (a)”或“一个 (an)”引入将含有该引入的权利要求元件的任何特定的权利要求限定于只含有这一个元件的发明,即使在相同的权利要求包括引入性短语“一个或多个”或“至少一个”以及诸如“一 (a)”或“一个 (an)”的不定冠词。对于不定冠词的使用同样如此。

[0057] 除非另有说明,否则诸如“第一”和“第二”的词语被用来任意区分此类词语所描述的元件。因而,这些词语并不一定要指示此类元件的时间或其它顺序。

[0058] 以下是本发明的各种实施例。

[0059] 项目 1 包括生成调试消息的数据处理系统,包括用于执行多个数据处理指令的数据处理器;与数据处理器耦接,用于存储和提供信息给数据处理器存储器的存储器;与数据处理器耦接,用于生成包括地址转换跟踪消息的调试消息的调试电路;与调试电路和数据处理器耦接的存储器管理单元,其中存储器管理单元包括用于实现地址转换的转换逻辑,以使地址在虚拟形式和物理形式之间转换。调试电路包括与存储器管理单元耦接的消息生成电路,用于在一个或多个地址转换映射被修改时接收通知。消息生成电路响应于检测到地址转换映射的修改而生成地址转换跟踪消息,并且将该地址转换跟踪消息提供给调试电路外部。项目 2 包括项目 1 的数据处理系统,其中存储器管理单元每当任何地址转换映射被更新时就发送指示符信号给消息生成电路。项目 3 包括项目 2 的数据处理系统,其中消息生成电路还包括用于选择性地去除地址转换跟踪消息的至少一个字段的压缩逻辑。项目 4 包括项目 3 的数据处理系统,其中地址转换跟踪消息中由压缩逻辑去除的至少一个字段每个都含有用于该字段的预定的普遍值。项目 5 包括项目 4 的数据处理系统,其中预定的普遍

值由数据处理系统的用户控制,并且预定的普遍值存储于可由数据处理系统的用户访问的存储位置。项目 6 包括项目 3 的数据处理系统,其中被去除的至少一个字段是地址转换大小字段或者用于识别进程标识符或地址空间之一的地址转换识别字段。项目 7 包括项目 1 的数据处理系统,其中该一个或多个地址转换映射存储于转换旁视缓冲器 (TLB) 中。项目 8 包括项目 7 的数据处理系统,其中存储器管理单元每当 TLB 写 TLB 条目或者使条目无效化时就发送指示符信号给消息生成电路。项目 9 包括项目 1 的数据处理系统,并且还包括与调试电路耦接的调试端子;以及与调试端子耦接,用于从调试电路接收地址转换跟踪消息的调试器。

[0060] 项目 10 包括一种用于生成调试消息的方法,包括以数据处理器执行多个数据处理指令;将调试电路耦接至数据处理器,用于生成包括地址转换跟踪消息的调试消息;将存储器管理单元耦接至调试电路和数据处理器,其中存储器管理单元包括用于实现地址转换的转换旁视缓冲器 (TLB),以使地址在虚拟形式和物理形式之间转换;检测 TLB 条目何时被修改;响应于检测到 TLB 条目的修改而生成地址转换跟踪消息;以及提供地址转换消息给调试端子。项目 11 包括项目 10 的方法,并且还包括每当 TLB 写 TLB 条目或者使条目无效化时将指示符信号从存储器管理单元发送到消息生成电路。项目 12 包括项目 11 的方法,并且还包括通过去除地址转换跟踪消息的至少一个字段来压缩地址转换跟踪消息的长度。项目 13 包括项目 12 的方法,并且还包括使被去除的至少一个字段与地址转换大小字段或者用于识别地址空间或进程标识符的地址转换识别字段相关联。项目 14 包括项目 12 的方法,并且还包括使预定的普遍值与地址转换跟踪消息的至少一个字段中的每一个相关联;以及仅在如果预定的普遍值对该至少一个字段存在时才去除该至少一个字段。项目 15 包括项目 14 的方法,并且还包括由数据处理系统的用户控制预定的普遍值;以及将预定值存储于用户可访问的多个寄存器中的一个或多个寄存器之内。

[0061] 项目 16 包括数据处理系统,该数据处理系统具有用于它的全局控制电路;与全局控制电路耦接的数据处理器,其中该数据处理器包括与指令解码器耦接的取指单元,该指令解码器与一个或多个执行单元耦接,用于执行多个数据处理指令;与全局控制电路和数据处理器耦接的存储器管理单元,其中该处理器管理单元包括用于实现地址转换的地址转换逻辑,以使地址在虚拟形式和物理形式之间转换;用于使存储器管理单元和全局互连连接的总线接口单元;存储器与全局互连耦接,用于存储来自数据处理器器的信息以及将信息提供给数据处理器;以及与全局控制电路和存储器管理单元耦接的调试电路,其中调试电路生成包括地址转换跟踪消息的调试消息。调试电路包括与存储器管理单元耦接的消息生成电路,用于在一个或多个地址转换映射被修改时接收通知。消息生成电路响应于检测到地址转换映射的修改而生成地址转换跟踪消息,并且将该地址转换跟踪消息提供给调试电路外部。项目 17 包括项目 16 的数据处理系统,其中消息生成电路还包括用于选择性地去除地址转换跟踪消息的至少一个字段的压缩逻辑。项目 18 包括项目 17 的数据处理系统,其中地址转换跟踪消息中由压缩逻辑去除的至少一个字段每个都含有用于该字段的预定的普遍值。项目 19 包括项目 18 的数据处理系统,其中预定的普遍值由数据处理系统的用户控制,并且预定的普遍值存储于可由数据处理系统的用户访问的存储位置中。项目 20 包括项目 16 的数据处理系统,其中一个或多个地址转换映射存储于转换旁视缓冲器 (TLB) 中,并且存储器管理单元每当 TLB 写 TLB 条目或者使条目无效化时就发送指示符信号给信号生成电路。

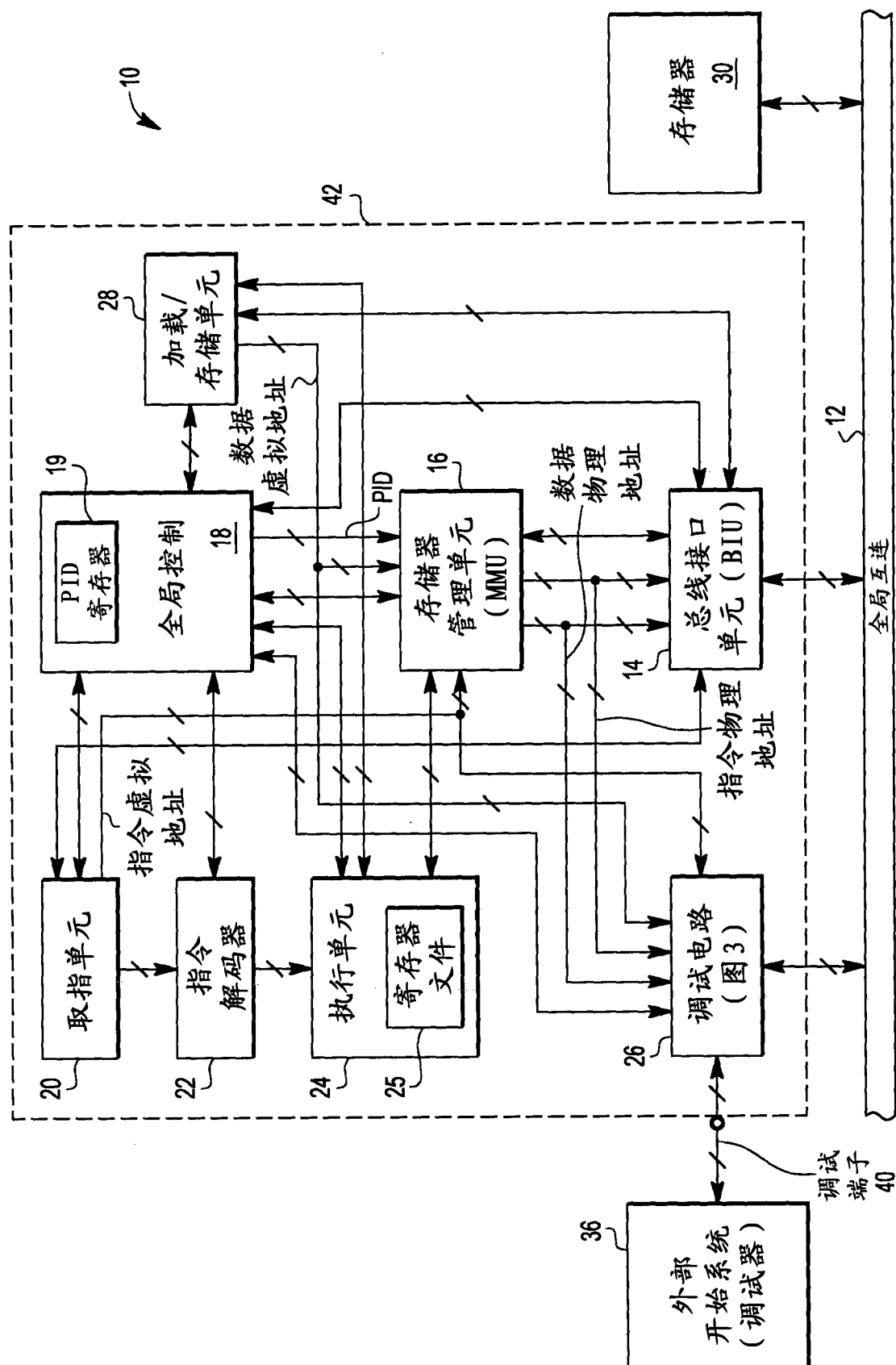


图 1

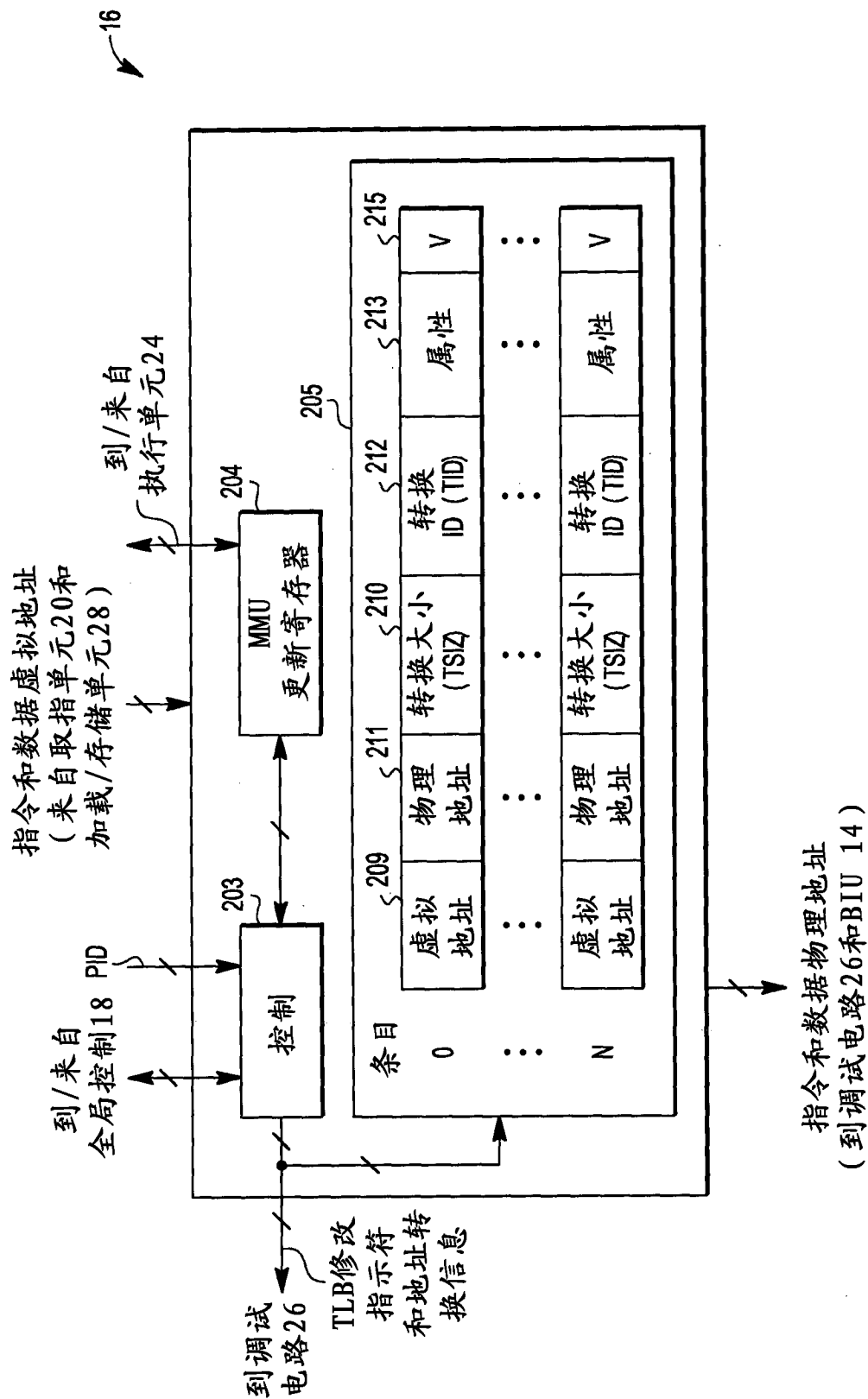


图 2

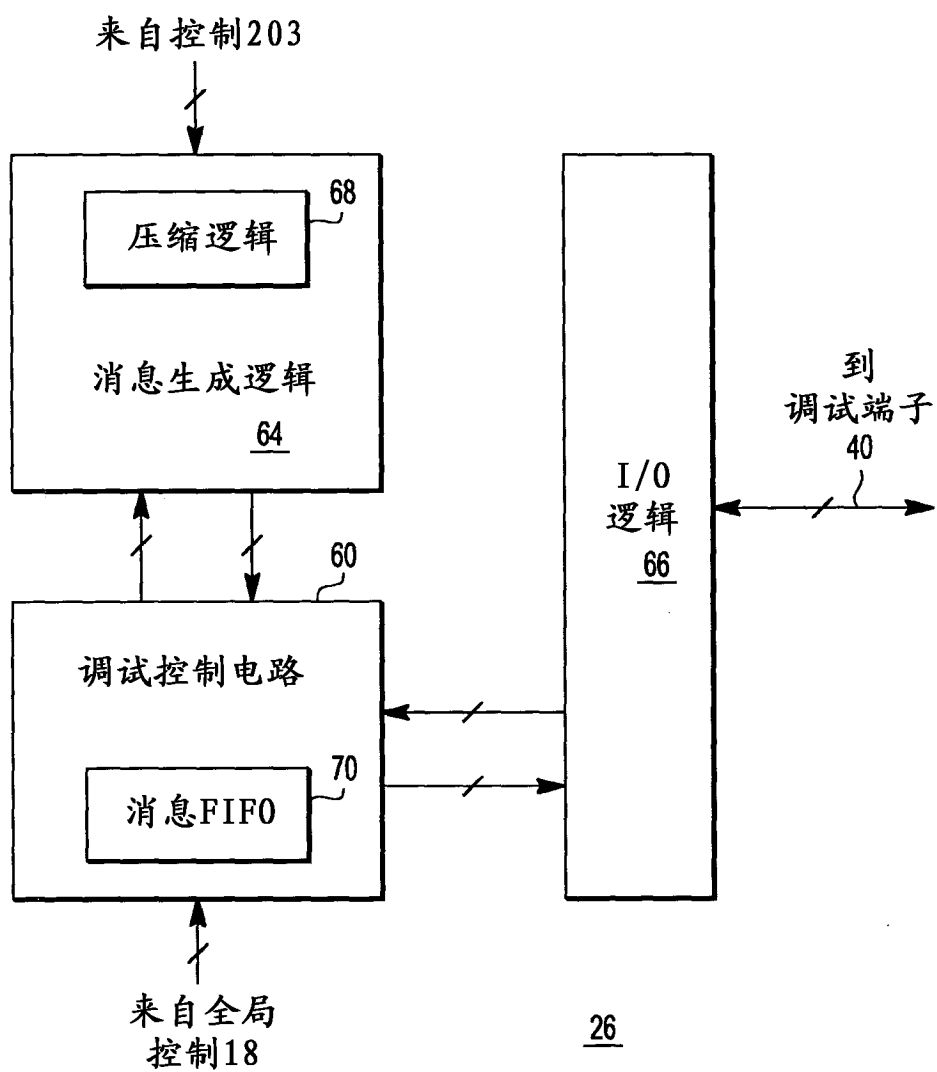


图 3

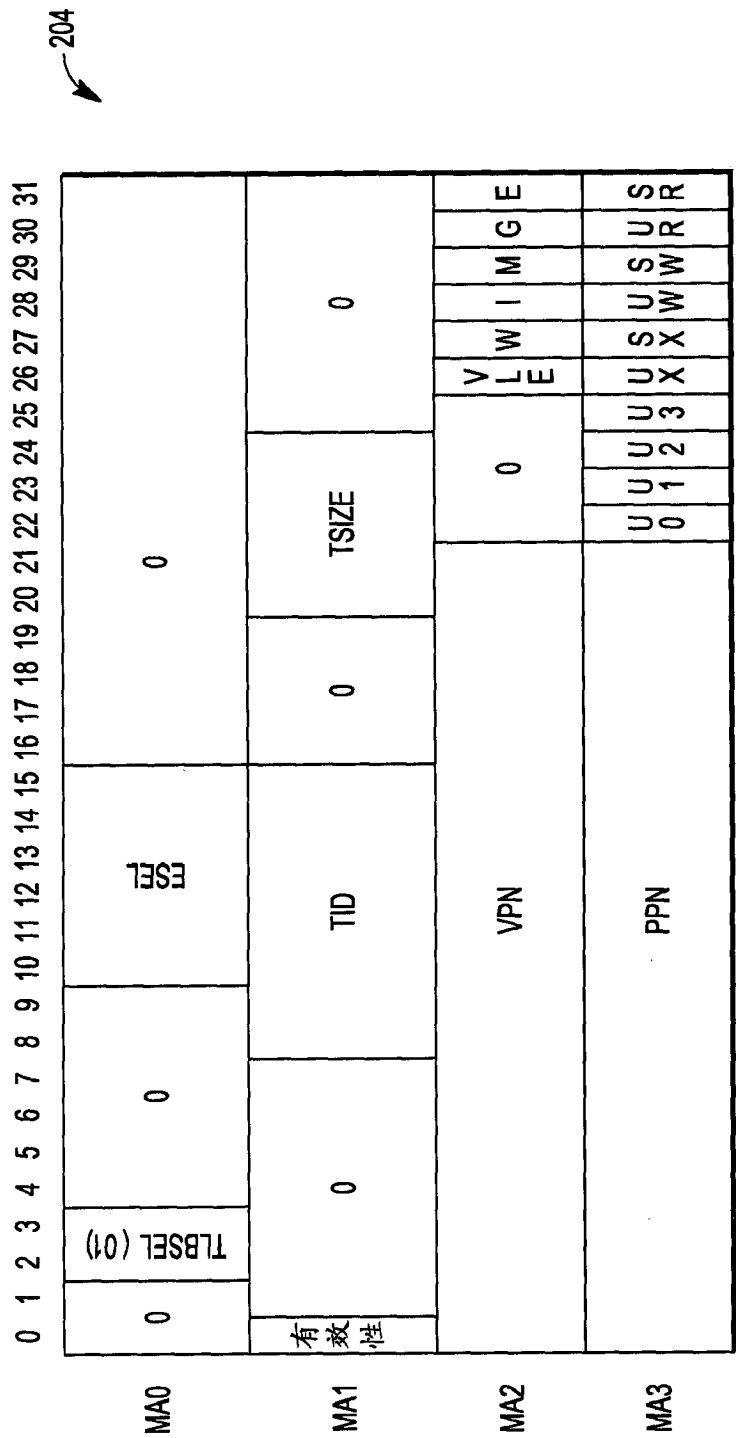


图 4

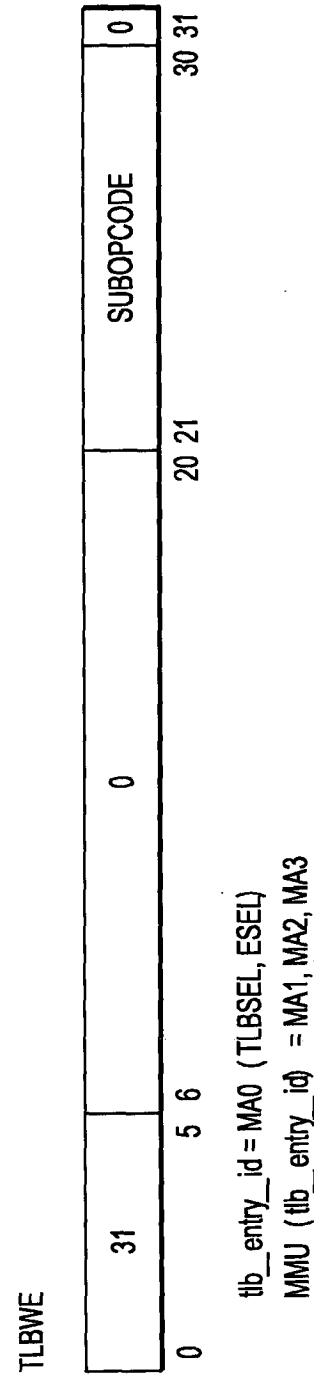


图 5 现有技术

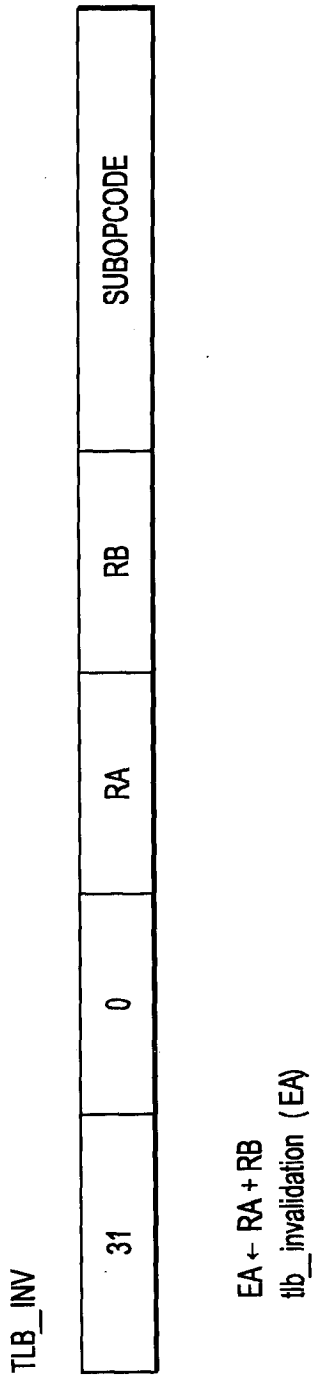


图 6 现有技术

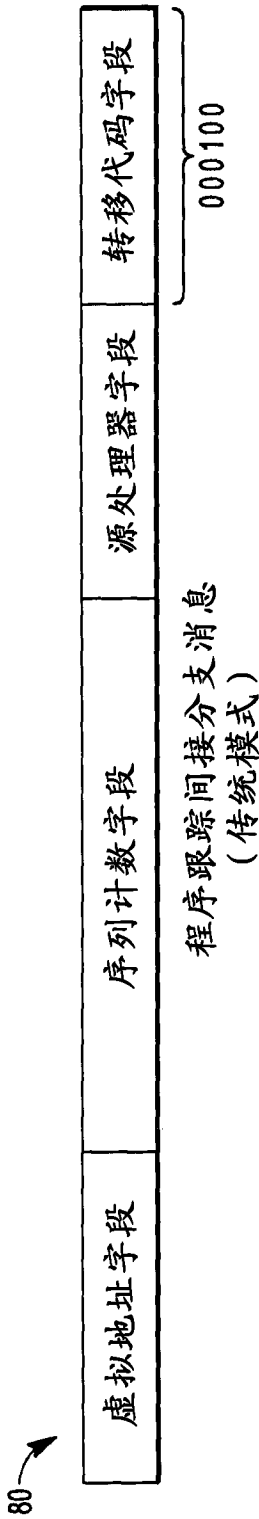


图 7 现有技术

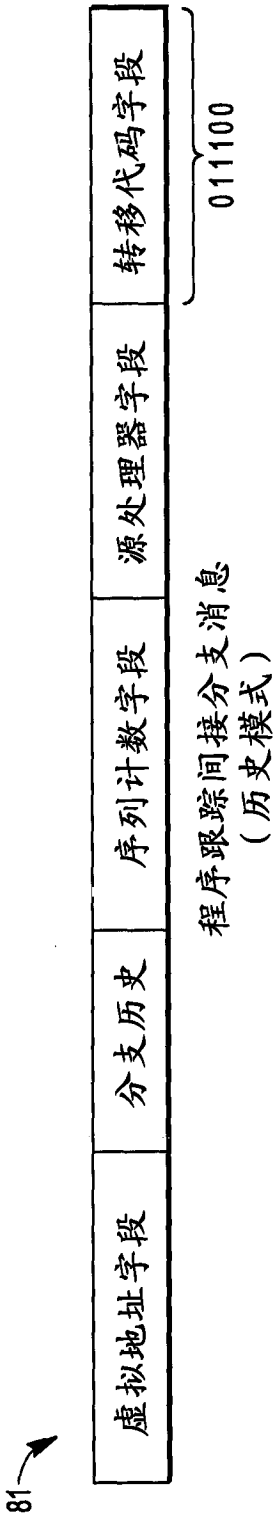


图 8 现有技术

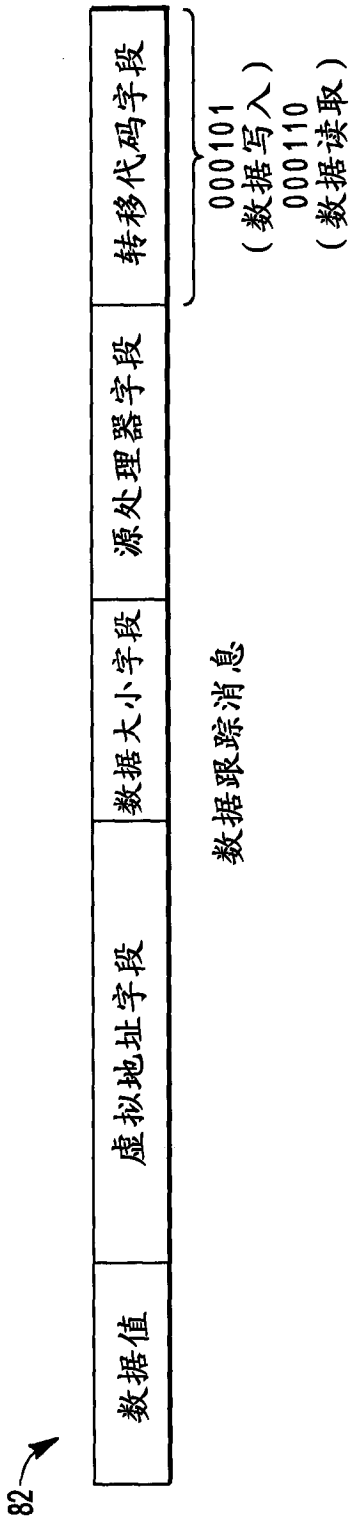


图 9 现有技术

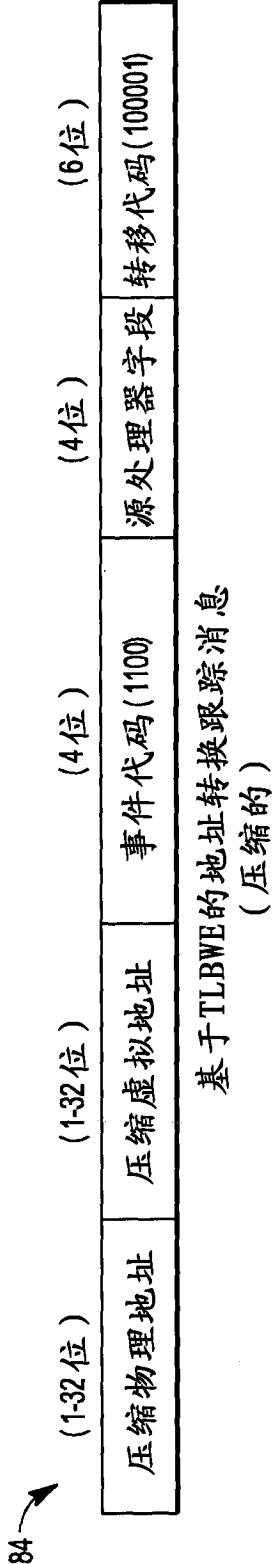


图 10

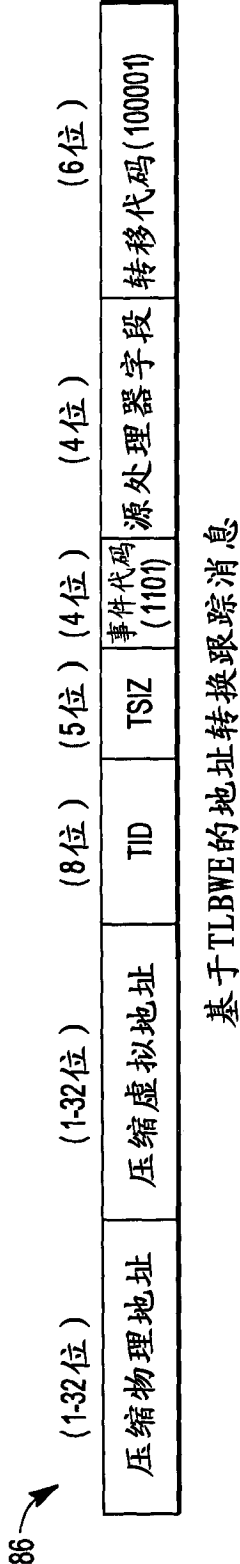


图 11

消息生成:
A1= 0000 0000 0000 0011 1111 1100 0000 0000
A2= 0000 0000 0000 0011 1111 0011 0110 0000
A1 ⊕ A2= 0000 0000 0000 0000 0000 1111 0110 0000
地址消息(M1) = 1111 0110 0000

地址重建
A1 ⊕ M1= A2
A1= 0000 0000 0000 0011 1111 1100 0000 0000
M1= 0000 0000 0000 0000 0000 1111 0110 0000
A2= 0000 0000 0000 0011 1111 0011 0110 0000

图 12 现有技术

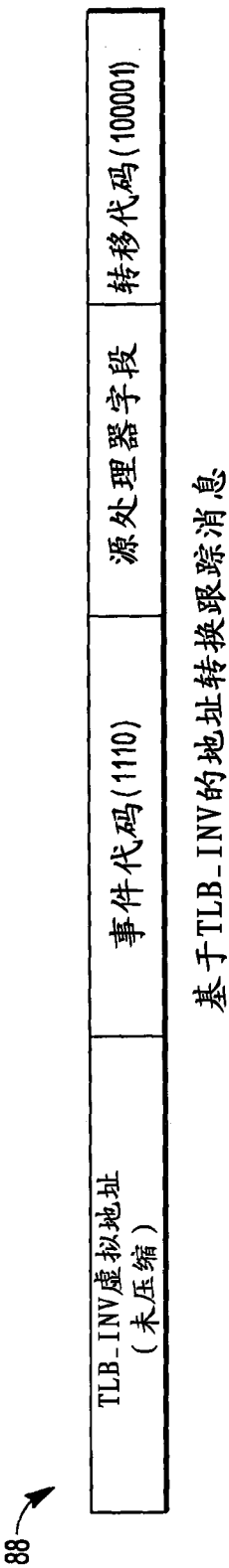


图 13

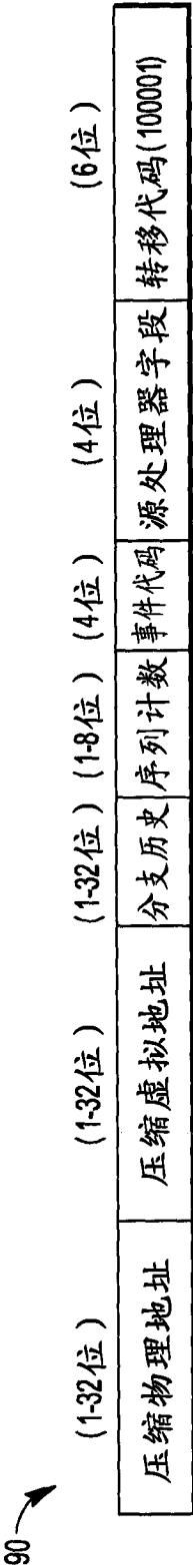


图 14

具有历史划界的地址转换跟踪消息

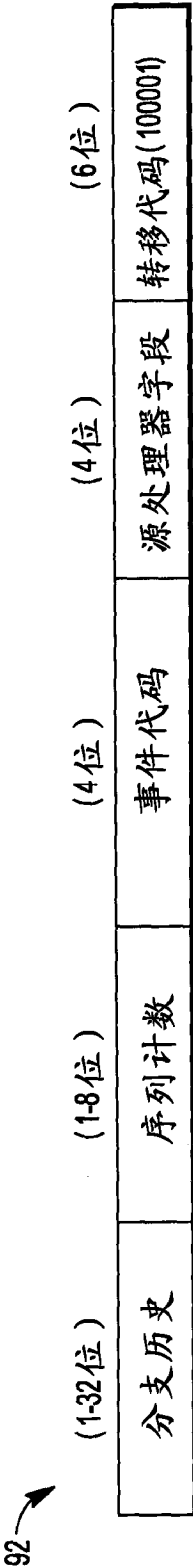


图 15

具有历史划界的程序关联消息