



[12] 发明专利说明书

[21] ZL 专利号 99106952.8

[45] 授权公告日 2003 年 10 月 15 日

[11] 授权公告号 CN 1124546C

[22] 申请日 1999.5.31 [21] 申请号 99106952.8

[30] 优先权

[32] 1998. 6. 1 [33] US [31] 09/087886

[71] 专利权人 国际商业机器公司

地址 美国纽约州

[72] 发明人 D·Q·恩古延

审查员 杨蕊

[74] 专利代理机构 中国专利代理(香港)有限公司

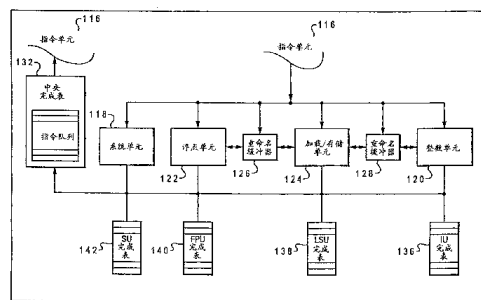
代理人 王勇 王岳

权利要求书 4 页 说明书 8 页 附图 5 页

[54] 发明名称 一种超标量处理器及在该超标量处理器中完成指令的方法

[57] 摘要

超标量处理器中的每个执行单元都有一个关联的完成表，该关联的完成表中含有一份所有已调度但未完成的指令的状态数据。用一个中央完成表来保存按调度单元和各执行单元所报告的每个已调度指令的状态。指令单元向完成表发送结束信号，完成表负责退役某特定类型的指令。中央完成表退役可能引起中断的指令和其结果要送到相同目标寄存器的指令。各执行单元关联的完成表退役余下的指令，并且执行单元将指令状态发送到中央完成表和每个执行单元。这就减少了要由中央完成表退役的指令的数量，增加了每个周期被退役的指令的数量。



1. 一种在超标量处理器中完成指令的方法，包含：
 - 向执行单元调度指令；
 - 将该指令的状态写到与该执行单元关联的完成表；
 - 5 将该指令的状态写到中央完成表；
 - 结束该指令并向该中央完成表发送结束信号；
 - 将该结束信号发送给与该执行单元关联的完成表；以及
 - 将指令完成信号发送到与该执行单元关联的完成表以及中央完成表。
- 10 2. 权利要求 1 的方法，其中，向执行单元调度指令的步骤进一步包含：
 - 探听该中央完成表内的指令队列，以检查是否有任何老指令有关联的可中断位被置位；
 - 置位该指令的等待位；
 - 15 将具有任何有可中断位的老指令的 ID 的指令的等待 ID 锁存。
3. 权利要求 2 的方法，其中，探听该中央完成表内的指令队列以检查是否有任何老指令有关联的可中断位被置位的步骤进一步包含：
 - 判定该老指令是否可能引起中断。
 - 20 4. 权利要求 2 的方法，进一步包含：
 - 响应比该指令更老的指令的完成；
 - 将该指令的等待 ID 与任何更老指令的 ID 比较；
 - 如果该指令的等待 ID 与任何更老指令的 ID 匹配，就清除该等待位；
 - 25 完成该指令；
 - 5. 权利要求 4 中的方法，其中，向执行单元调度指令的步骤进一步包含：
 - 判定该指令是否处于推测性路径。
 - 6. 权利要求 5 中的方法，进一步包含：
 - 30 如果判定该指令处于推测性路径，则等待要分解的推测性路径。
 - 7. 权利要求 5 中的方法，进一步包含：

- 判断是否有第二个老指令会与该指令针对相同的寄存器。
8. 权利要求 7 中的方法, 进一步包含步骤 a):
如果判定有第二个老指令与该指令针对相同的寄存器, 则将结束信号写到与该执行单元关联的该完成表。
- 5 9. 权利要求 8 中的方法, 进一步包含在步骤 a) 之后执行的步骤 b):
更新与该执行单元关联的完成表。
- 10 10. 权利要求 9 中的方法, 进一步包含在步骤 b) 之后执行的步骤 c):
完成该指令;
将完成数据发送给所有执行单元和该中央完成表。
- 15 11. 权利要求 7 中的方法, 进一步包含:
如果判定没有第二个老指令与该指令针对相同的寄存器, 则执行单元检查在其完成表中是否有尚未完成但可能引起中断的老指令。
- 15 12. 权利要求 11 中的方法, 进一步包含:
如果判定有任何老指令可能会引起中断, 则执行步骤 d): 将第二个、与该老指令关联的结束信号发送到该中央完成单元。
- 20 13. 权利要求 12 中的方法, 进一步包含在步骤 d) 之后执行的步骤:
e) 完成该指令或该老指令或该第二个老指令;
f) 更新与该执行单元关联的该完成表以及该中央完成表。
- 25 14. 一个处理器, 包含:
一个指令单元;
多个执行单元;
多个完成表;
一个中央完成表;
用于响应接收调度指令的命令而将指令从指令单元调度到多个执行单元之一去执行的装置;
用于将该指令状态写到与多个执行单元之一关联的多个完成表之一的装置;
- 30 用于将该指令状态写到中央完成表的装置;
用于结束该指令并向该中央完成表发送结束信号的装置;
用于将该结束信号发送给与多个执行单元之一关联的多个完成表

之一的装置;

用于检查与多个执行单元之一关联的多个完成表之一的装置;以及

5 用于将指令完成信号发送给与多个执行单元之一关联的多个完成表之一的装置。

15. 权利要求 14 中的处理器, 进一步包含:

用于探听该中央完成表内的指令队列以检查是否有任何老指令有关联的可中断位被置位的装置;

用于置位该指令的等待位的装置;

10 用于将具有任何有可中断位的老指令的等待 ID 的指令 ID 锁存起来的装置。

16. 权利要求 15 的处理器, 进一步包含:

响应比该指令更老的任一指令的完成,

用于将该指令的 ID 与任何更老指令的等待 ID 比较的装置;

15 用于如果该指令的 ID 与任何更老指令的等待 ID 匹配, 就清除该等待位的装置;

用于完成该指令的装置;

17. 权利要求 16 中的处理器, 进一步包含:

用于判定该指令是否处于推测性路径的装置。

20 18. 权利要求 17 中的处理器, 进一步包含:

用于判定是否有任何老指令可能引起中断的装置。

19. 权利要求 18 中的处理器, 进一步包含:

用于判断是否有第二个老指令会与该指令针对相同的寄存器的装置。

25 20. 权利要求 19 中的处理器, 进一步包含:

用于将结束信号写到与该多个执行单元之一关联的多个完成表之一的装置。

21. 权利要求 20 中的处理器, 进一步包含:

用于更新与该多个执行单元之一关联的多个完成表之一的装置。

30 22. 权利要求 21 中的处理器, 进一步包含:

用于完成该指令的装置;

用于将完成数据发送给所有执行单元和该中央完成表的装置。

23. 权利要求 22 中的处理器，其中，用于判断该指令是否是推测性路径的装置进一步包含：

用于将该结束信号发送到该中央完成单元的装置。

24. 权利要求 23 中的处理器，其中，用于判定该老指令可能引起
5 中断的装置进一步包含：

用于将第二个、与该老指令关联的结束信号发送到该中央完成单元的装置。

25. 权利要求 24 中的处理器，其中，用于判定一个不同的老指令可能与该指令指向相同的寄存器的装置进一步包含：

10 用于发送一个与该第二个老指令关联的结束信号到该中央完成单元的装置。

26. 权利要求 25 中的处理器，进一步包含：

用于完成该指令或该任一老指令或该第二个老指令的装置；

15 用于更新与多个执行单元之一关联的多个完成表之一以及该中央完成表的装置。

一种超标量处理器及在该超标量处理器中完成指令的方法

5 本发明一般涉及超标量处理器，具体来说涉及每个周期完成更多指令，更具体来说，本发明涉及对超标量处理器的指令完成方案和完成单元的改进。

10 微处理器被分类为超标量的，如果该微处理器（“处理器”）每个时钟周期能够完成多条指令。超标量处理器的体系结构采用处理器内部的多个并行处理单元，以便能在每个时钟周期完成更多的指令。这些处理单元一般包括多个并行操作的执行单元、一个向执行单元发送指令和数据的调度单元、一个含有一个用于跟踪及退役（retire）指令的（“完成表”）的完成单元和用于为执行单元预加载指令的重命名缓冲器（重命名寄存器）。完成表的跟踪和退役特性提供了不按顺序完成指令的能力。

15 为了效率和速度而采用多个并行的处理单元，要求指令被“流水线化”。流水线是这样一种提取和解码指令的方法，它使得一个执行单元不必等待指令；执行单元在第一个指令已经完成之前就处理第二个指令。此外，当前的体系结构采用推测性执行（执行来自不同分支路径的指令）和分支预测（branch prediction）来提高处理器的性能。分支预测用于预测某指令下一次被执行时要转移的途径，这种预测 20 90%的时候都是正确的。

25 图4是表示超标量处理器中指令流的简化框图。多个指令被提取器402从指令高速缓存中提取出来后置于分支处理单元404或指令队列406。进入指令队列406之后，指令被调度单元408发送到各种执行单元。调度速率的相关因素中包括执行单元的忙状态、重命名（未予显示）的可用性以及完成表缓冲器的可用性。在当前处理器中，指令完成逻辑是在处理器内的一个单一单元—完成单元中执行的。完成单元418对指令在从调度到执行直至完成的过程中进行跟踪，允许不按顺序地执行指令。当执行单元结束指令时，指令的状态被执行单元 30 410、412、414或416传送到完成表418。完成单元然后就完成或退役该指令并向其余执行单元发送一个完成信号，以便将结束的数据写回到结构化寄存器中。

然而，完成单元每个周期所能完成或退役的指令的数目是固定的；这是一种可能导致在指令调度逻辑中产生瓶颈的限制因素。所有指令都由完成单元 418 跟踪、完成并提交给特定的结构化寄存器。如果完成单元 418 停顿，处理器 400 的效率和速度就会降低，其一个原因
5 因是执行单元 410、412、414 和 416 不能发出结束信号来更新完成单元 418 中的对应指令状态。执行单元 410、412、414 和 416 由于结束信号不能被传送到一个完整的完成表 418 队列而停顿。由于重命名缓冲器（未予示出）现在不能向执行单元 410、412、414 和 416 传送指令，而继续阻塞（backup）。调度单元 408 作出判定，即重命名缓冲器
10 器已满，没有位置存放其它指令。所以，除非重命名缓冲器中有空间可用，否则调度单元 408 将不调度指令，导致处理器停顿。

完成单元 418 瓶颈的第二个原因是，尽管完成单元 418 能同时退役多条指令，但完成单元必须按程序顺序来退役指令，如果完成表 418 条目 0（第一个指令退役位置）的指令因为某指令仍然在执行而不能
15 被退役，这指令完成就可能会停顿。

处理器每时钟周期能执行多条指令的速度，受到处理器的完成表完成或退役指令的能力的限制。

因此，人们希望提供这样一种完成指令的方法与装置，它要排除常规完成单元产生的瓶颈。人们也希望每个周期能完成更多指令。人们
20 们进一步希望提高完成表在退役指令时的效率。

因此，本发明的一个目的是通过每周期完成更多指令来提高处理器性能。

本发明的另一个目的是消除由常规完成单元引起的指令完成瓶颈。

25 本发明的另一个目的是提供一种提高完成表在退役指令时的效率的方法和装置。

现在来说明上述目的是如何实现的。超标量处理器中的每个执行单元都有一个关联完成表，关联完成表中含有一份所有已调度但未完成的指令的状态数据。用一个中央完成表来保存按调度单元和各执行
30 单元所报告的每个已调度指令的状态。指令单元向完成表发送结束信号，完成表负责退役某特定类型的指令。中央完成表退役可能引起中断的指令和其结果要针对相同目标寄存器的指令。各执行单元关联的

完成表退役余下的指令，并且执行单元将指令状态发送到中央完成表和每个执行单元。这就减少了要由中央完成表退役的指令的数量，增加了每个周期被退役的指令的数量。

5 以下的详细说明，透彻地阐述了本发明的以上以及其它目的、特点和优点。

后附的权利要求书中陈述了被视为本发明特有的新颖特征。然而，欲完全理解本发明本身和优选应用方式及其进一步的目的和优点，就要参阅以下结合各附图对示意性实施例的详细说明。附图简介：

10 图 1A 是表示能在其中实现本发明的优选实施例的处理器和数据处理系统有关部分的高级框图。

图 1B 是按照本发明优选实施例的具有多个执行单元和多个完成表的数据处理器的高级框图。

图 2 表示按照本发明优选实施例的中央完成表指令队列。

15 图 3 是表示按照本发明优选实施例的分布式完成逻辑的方法的高级流图。

图 4 表示的高级流图表示经过超标量处理器内完成单元的指令流。

20 现在参考各图，特别是参考图 1A，该图是表示能在其中实现本发明的优选实施例的处理器和数据处理系统有关部分的框图。处理器 100 是一个单一的集成电路超标量处理器，诸如（位于美国纽约州 Armonk 的）IBM 公司提供的 PowerPC™ 处理器。因此，处理器 100 包括的各种单元、寄存器、缓冲器、存储器和其它部分，所有都是由集成电路形成的。处理器 100 也按照精简指令集计算（RISC）技术操作。

25 处理器 100 包括一级（L1）指令高速缓存和数据高速缓存（I 高速缓存和 D 高速缓存）102 和 104，各自分别都有关联的存储管理单元（“I MMU”和“D MMU”）106 和 108。如图 1 中所示，处理器 100 通过总线接口单元 114 与系统地址总线 110 和系统数据总线 112 相连。指令通过总线接口单元 114 从系统存储器（未于示出）被检索到
30 处理器 100 并被存储在指令高速缓存 102，而通过总线接口单元 114 检索的数据则被存储在数据高速缓存 104。指令是由指令单元 116 从指令高速缓存 102 中按需提取的，指令单元包括指令提取逻辑、指令

转移预测逻辑、指令队列和调度单元。

指令单元 116 内的调度单元将适当的指令调度到各执行单元，诸如系统单元 118、整数单元 120、浮点单元 122 或加载/存储单元 124。系统单元 118 执行条件寄存器逻辑、专用寄存器转移以及其它系统指令。整数或“定点”单元 120 对整数进行加法、减法、乘法、除法、移位或循环操作，从整数或通用寄存器（“GPR 文件”）126 检索操作数并在其中存储结果。浮点单元 122 进行单精度和/或双精度的乘法/加法操作，从浮点寄存器（“FPR 文件”）128 检索操作数并在其中存储结果。

10 加载/存储单元 124 按需要将指令操作数从数据高速缓存 104 加载到整数或浮点寄存器 126 或 128，并将能从整数或浮点寄存器 126 或 128 得到的指令结果存储到数据高速缓存 104。加载与存储队列 130 被用于在数据高速缓存 104 与整数或浮点寄存器 126 或 128 之间的这些转移。完成单元 132 包括重定序缓冲器，它与指令单元 116 联合操作来支持无序的指令处理，也与整数和浮点寄存器 126 或 128 内的重命名缓冲器联合操作以避免指令结果对于特定寄存器的冲突。公用片上处理器（“COP”）和联合测试行动组（“JTAG”）单元 134 为进行边界扫描互连测试而提供一个连接到系统的串行接口。

图 1 所示的体系结构完全是为示意和解释本发明而给出的，并不意味着隐含有结构性的限制。本领域的熟练人员会认识到，许多变化都是有可能的。例如，处理器 100 可包括多个整数和浮点执行单元以提高处理的吞吐量。所有这类变化都在本发明的精神和范围之内。

现在参见图 1B，该图是一个按照本发明优选实施例的具有多个执行单元和多个完成表的处理器的高级框图。中央完成表 132 控制所有的指令退役。中央完成表 132 有一个指令队列，指令队列有固定数量的退役位置限制输出。在本发明中，向处理器添加了额外的完成表，每个执行单元都有一个指令完成表，用于完成中央完成表 132 没有完成的指令。

每个执行单元都负责完成其自己类型的指令。执行单元将完成信息广播给其它执行单元以及中央完成表。多个完成表的并行操作会使指令完成的速率更高。通过减少由中央完成表 132 处理的指令的数量，完成的速率得到提高，中央完成表 132 队列被清空的速度更快；

完成速率越高，完成表队列被清空的速度就越快，从而使得每周期的指令调度速率更高，因为重命名缓冲器也以更快的速度被释放。

在图 1A 中，描述了处理器主要部件的操作。在图 1B 中，处理器的操作基本上是相同的，其例外之处是增加了完成表 136、138、140 和 142。每个执行单元—整数单元 120、加载/存储单元 124、浮点单元 122 和系统单元 118—都有一个关联的完成表。如果重命名缓冲器 126 和 128 是开放的，则指令就从指令单元 116 发出，被某执行单元例如浮点单元 122 接收。个别的指令单元—此例中是浮点单元 122—确定指令的类型，结束指令并将结束信息发送到中央完成表 132 和浮点单元 122 关联的完成表 140。根据指令的类型，中央完成表 132 或完成表 140 将把该指令退役到特定结构的寄存器。

通过将通常由中央完成表 132 退役的指令转移到执行单元完成表 136、138、140 和 142，中央完成表 132 中的指令完成队列变得更加宽松。中央完成表 132 中指令的这种减少，提供了更高的总体吞吐量。

参见图 2，该图展示了一个按照本发明优选实施例的中央完成表指令队列 200。中央完成表指令队列 200 用于保存机器中所有指令的状态。指令可以相对于彼此不按顺序地完成，在处理器流水线中存在一条可能引起中断的先前指令的情况下，指令不得被允许完成。防止指令无序完成的方法，是由调度单元为中央完成表指令队列 200 中每个可能引起中断的条目施加一个“可中断位”给该指令。

在本发明中，探听逻辑 202 被应用到输入指令，诸如 204，以允许探听中央完成表指令队列 200，判定是否有任何老指令有中断位 206 置位。

输入指令 204 有多个部分，它们其中的作用是确定要为指令作什么、何时完成指令以及队列中是否有相关指令。图 2 中展示了对于本发明来说是重要的指令部分。例如，ID 208 是指令标识符，由它来识别队列中的特定指令。指令体在图示表中是以标题“指令”标识的。

“Int”（“可中断”）206 是在结束的指令中包含的一个“可中断位”，用于标识可能在指令流中引起中断的较老指令。处理器通过使用可中断位，就能不按指令之间的相互顺序来完成指令并仍然保持程序的顺序。

转移待定位 BP 214 表示指令处于未分辨的转移路径。当转移得到分辨时，这个位被清除以表示指令不再是推测性的。TC 218 是一个由调度单元在调度单元探听中央完成表、确定已经有老指令将相同寄存器作为目标寄存器之后而设置的位。当 TC 218 位置位时，老指令必须有中央完成表来完成。等待位 216 是输入指令内的一个位，在发现某可中断指令诸如指令 208 时被置位。然后，通过将输入指令 204 的等待位 216 置位，使指令 ID 208 与输入指令 204 的等待 ID 212 被锁存。当老的、可中断指令 ID 208 完成时，它现在就能简单地被指令 204 探听。指令 ID 208 与指令 204 的等待 ID 212 进行比较，如果 ID 匹配，指令 204 的等待位 216 就被清除。然后，如果准备就绪，指令 204 就能被完成。

输入指令 204 简单地探听中央完成表指令队列 200 去寻找可中断指令 ID，本例中是指令 208。当老指令 208 被完成时，就将指令 208 的 ID 210 与完成队列中的指令 204 的等待 ID 212 比较。如果存在有匹配，则等待位 216 被清除，指令 204 就能被完成。

参见图 3，该图表示了一个按照本发明优选实施例的分布式完成逻辑的方法的高级流程图。处理从步骤 300 开始，该步骤表示调度单元向某执行单元发送一条指令。在调度单元向执行单元发送指令的同时，过程前进到步骤 302，该步骤表示一个指示指令调度到执行单元的信号被发送到中央完成单元。过程继续到步骤 304，该步骤表示执行单元将指令状态写到其本地完成表。

然后，过程继续到步骤 306，该步骤表示执行单元执行所接收的指令。过程然后前进到步骤 308，该步骤表示执行单元结束该指令。过程下一步前进到步骤 310，该步骤表示执行单元向中央完成表发送一个结束信号。该步骤更新中央完成表中的指令状态。同时，过程前进到步骤 312，该步骤表示结束信号也被发送到执行单元的完成表。过程继续到步骤 313，该步骤表示执行单元将结束的指令的结果写到其完成表。

过程下一步前进到步骤 314，该步骤表示执行单元检查关联的完成表以判定已经执行的指令的类型。然后，过程前进到步骤 316，该步骤表示执行单元为判定指令类型所作的三项检查之一。步骤 316 表示为判定指令是否处于推测性路径的检查。如果指令处于推测性路

径，过程就继续到步骤 317，该步骤表示执行单元等待推测性路径去分辨自身。过程然后返回步骤 316 继续，直到推测性路径被分辨出来。

如果指令不处于推测性路径，过程下一步就前进到步骤 318，该步骤表示执行单元检查在其完成表中是否有尚未完成但是可能引起中断的老指令（诸如加载或存储指令）。如果执行单元的完成表中的老指令可能引起中断，则过程就前进到步骤 322，该步骤表示中央完成表完成（退役）该结束的指令。中央完成表清除已完成指令的方法是将该指令委托给某个目标地址，或者因中断或错误的转移预测而丢弃该指令。当中央完成表完成指令时，它将完成信息广播给该执行单元以更新处理器中的所有完成表。完成了所结束的指令之后，过程下一步前进到步骤 324，该步骤表示中央完成表用完成信息来更新执行单元的各个完成表。如果老指令不会引起中断，过程就前进到步骤 320，该步骤表示执行单元检查老指令是否与该指令有相同的目标寄存器。如果是，过程就前进到上面讨论过的步骤 322 和 324。

过程下一步前进到步骤 326，该步骤表示执行单元更新其完成表和所有其它完成表，包括中央完成表。

过程然后前进到步骤 328，该步骤表示执行单元完成该指令。执行单元也将完成数据发送给所有其它执行单元和中央完成表。如果该指令能被执行单元的完成表退役，则执行单元就完成它。过程然后前进到步骤 330，该步骤表示执行单元更新适当的结构寄存器（architected register）并释放所有与该指令关联的重命名缓冲器。

过程在中央完成表与执行单元的各个完成表之间划分指令。通过调度时为每个指令增加一个转移待定位的方法，每个完成表都能处理来自不同路径的指令。转移待定位如果处于置位状态，则表示指令处于一个未分辨的转移路径。当转移被分辨出后，该位就被清除，表示指令不再是推测性的。通过将特定指令发送到本地完成表去完成而不是发送到中央完成表，减少了中央完成表必须跟踪和退役的指令的数量。这就释放了中央完成表指令队列的空间，从而提高了指令流的速度。

总之，可能被处理器内常规完成单元禁止的指令流，在本发明中可以通过分布式完成方案和多完成表而得到改善。多个完成表并行操

作，会使指令完成的速率更高。每个执行单元完成自己类型的指令并将完成信息广播给其它执行单元和中央完成表。由于由中央完成表处理的指令数量的减少，完成速率得到提高。中央完成表被清空的速度更快，提高了指令完成的速率。由于中央完成表队列以更快的速度被
5 清空，所以就获得了每周期指令调度速率更高的效果。

尽管本发明是结合优选实施例而具体展示和说明的，本领域的熟练人员会明白，在形式和内容上可以对其作出各种修改而不超出本发明的精神和范围。

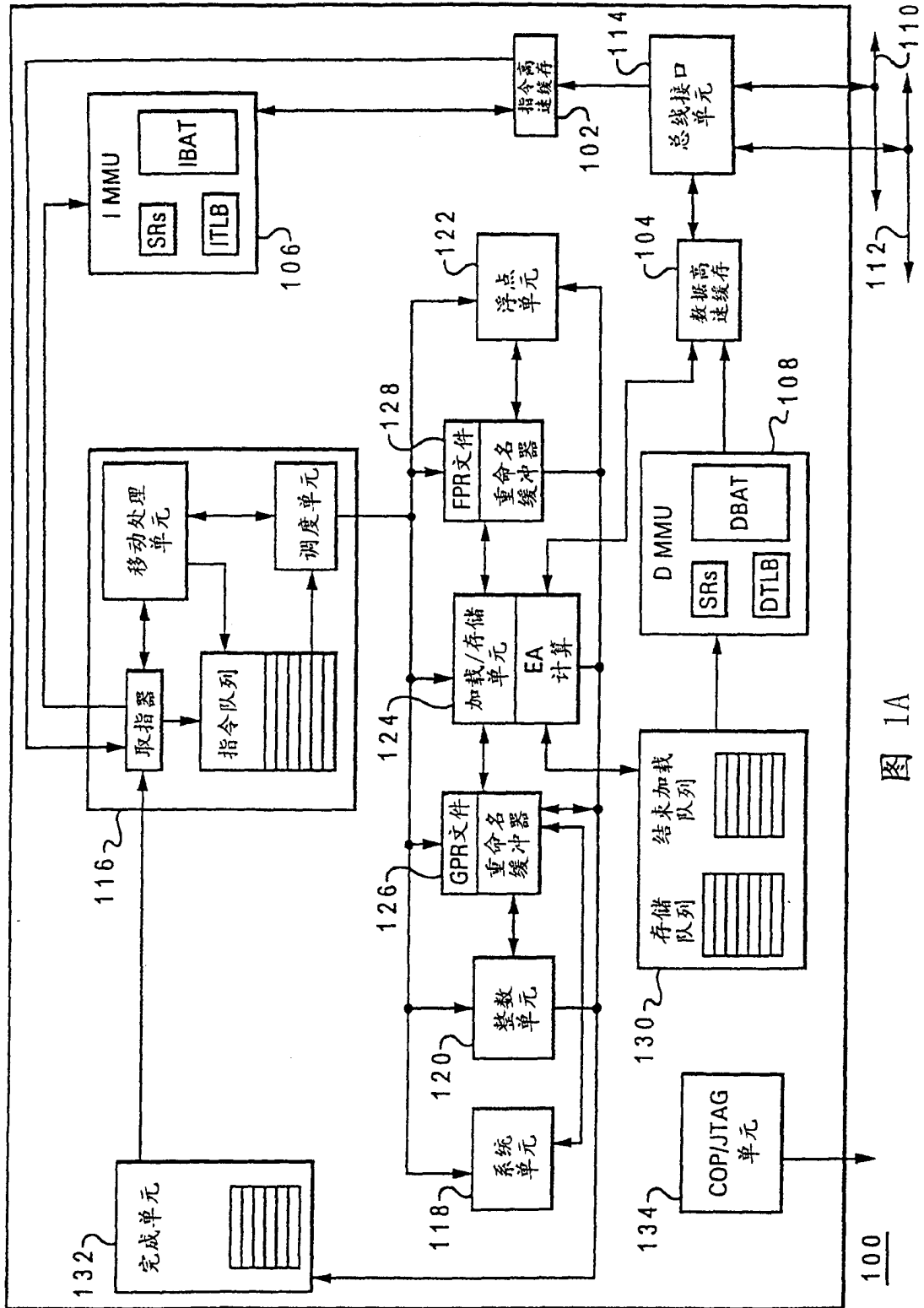


图 1A

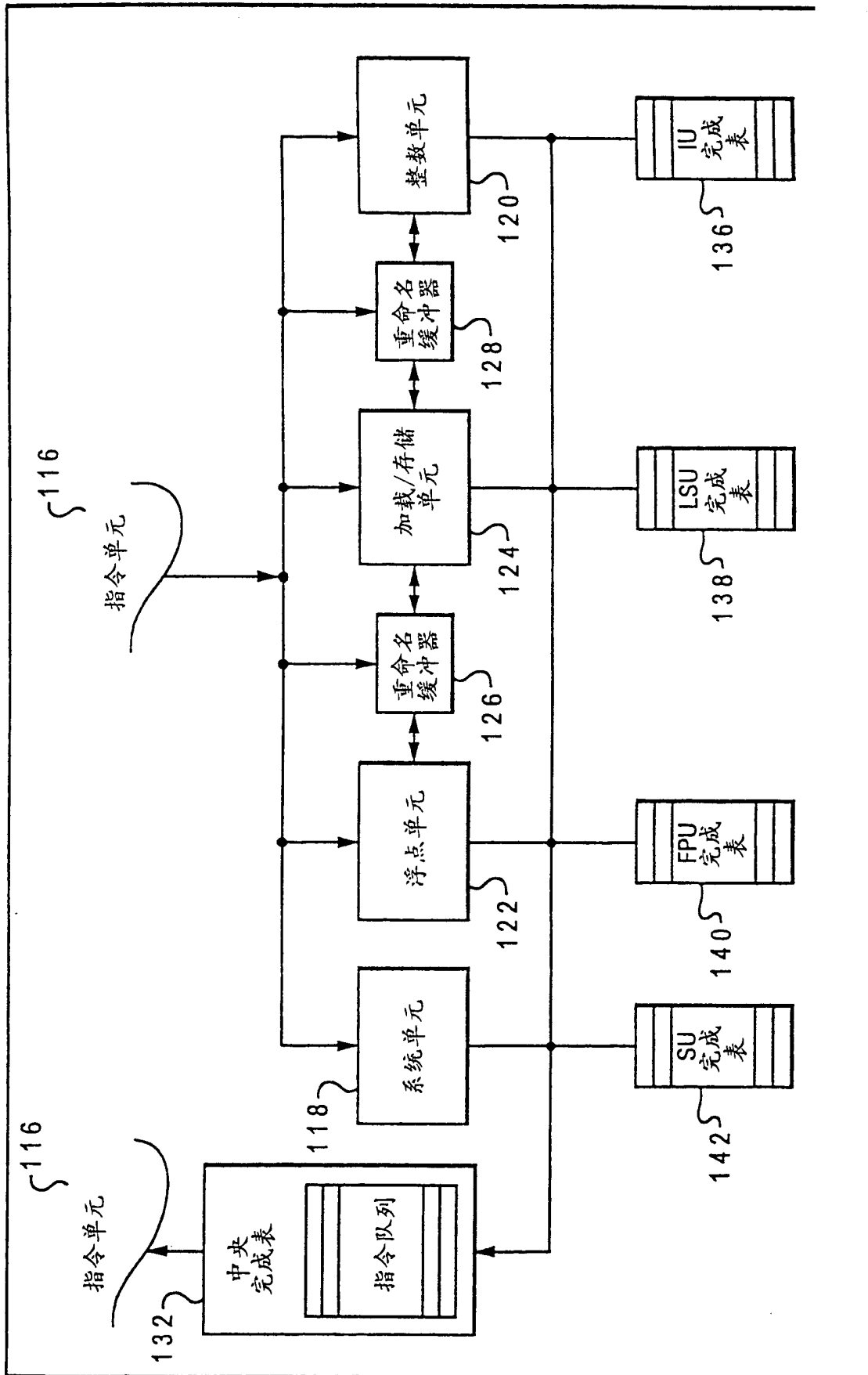


图 1B

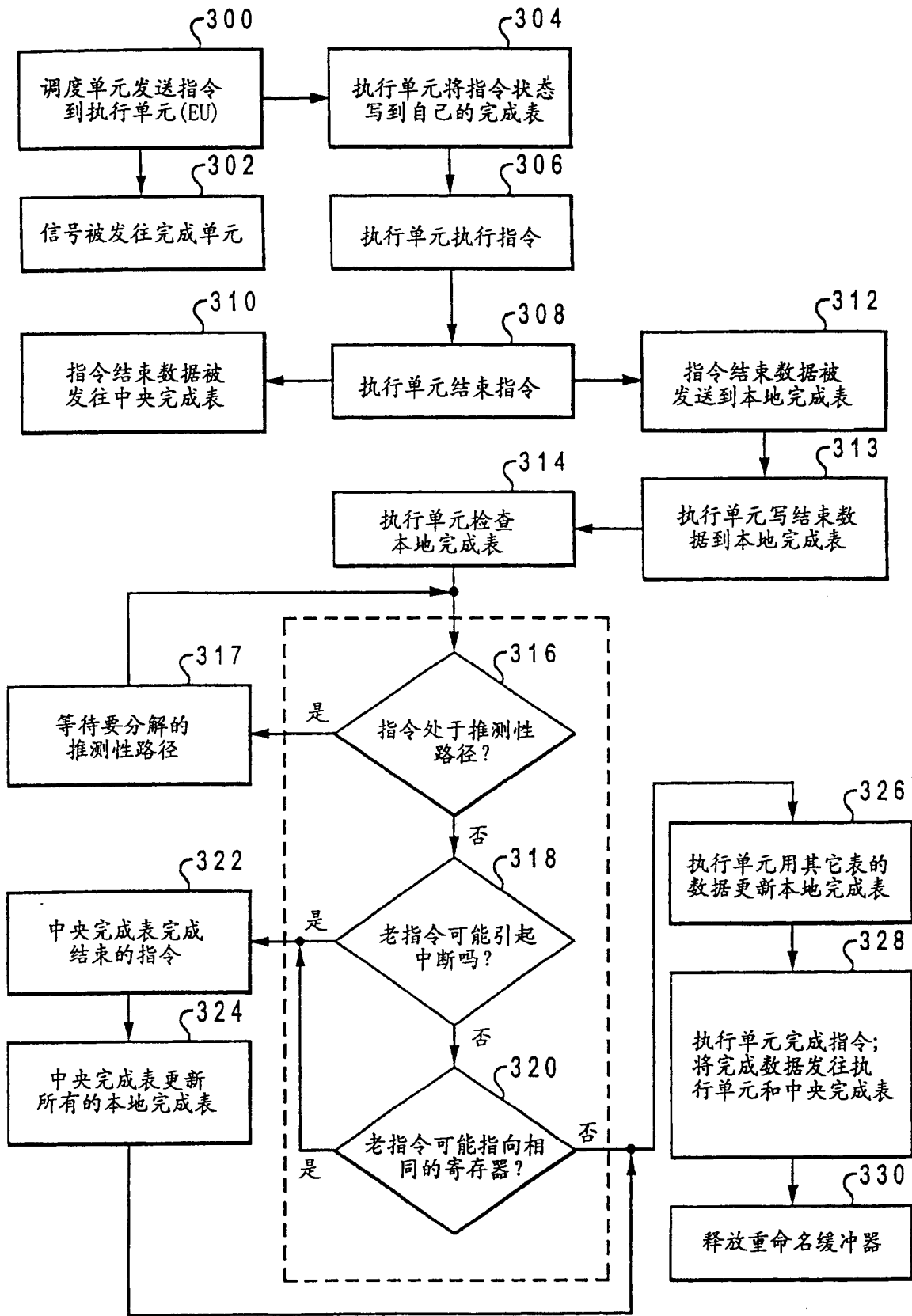


图 3

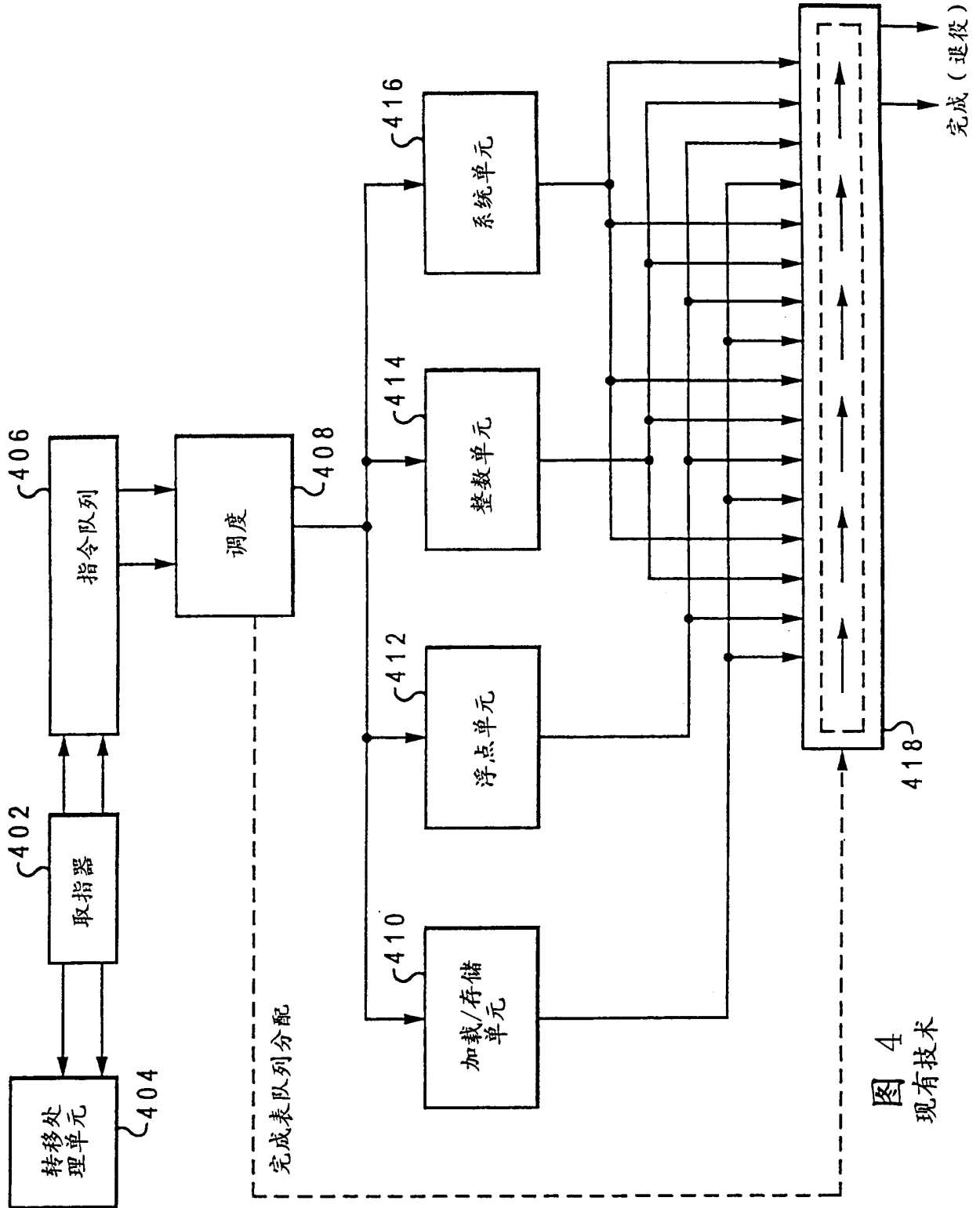


图 4
现有技术