

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 9/30 (2006.01)



[12] 发明专利申请公布说明书

[21] 申请号 200610088938.9

[43] 公开日 2008 年 1 月 30 日

[11] 公开号 CN 101114217A

[22] 申请日 2006.7.27

[21] 申请号 200610088938.9

[71] 申请人 中国科学院计算技术研究所

地址 100080 北京市海淀区中关村科学院南路 6 号

[72] 发明人 袁 楠 范东睿

[74] 专利代理机构 中科专利商标代理有限责任公司
代理人 朱进桂

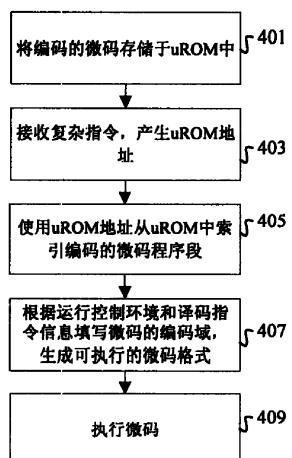
权利要求书 2 页 说明书 8 页 附图 3 页

[54] 发明名称

对复杂指令译码生成微码的译码装置和方法

[57] 摘要

公开了一种对复杂指令译码生成微码的译码装置，包括：初步译码模块，用于生成与输入的复杂指令相对应的、索引 uROM 中所存储的微码程序段的入口地址，并提供给 uROM 地址控制器；uROM 地址控制器，用于根据初步译码模块输入的用于索引 uROM 中所存储的微码程序段的入口地址，将与输入的复杂指令相对应的微码程序段的已编码微码从 uROM 中读取到微码解码器；uROM，用于存储与复杂指令相对应的微码程序段；以及微码解码器，根据输入的复杂指令信息和处理器运行时刻状态信息，对从 uROM 中读取的已编码微码进行解码。



1、一种对复杂指令译码生成微码的译码装置，包括：

初步译码模块，用于生成与输入的复杂指令相对应的、用于索引uROM中所存储的微码程序段的入口地址，并提供给uROM地址控制器；

uROM地址控制器，用于根据初步译码模块输入的用于索引uROM中所存储的微码程序段的入口地址，将与输入的复杂指令相对应的微码程序段中的已编码微码从uROM中读取到微码解码器；

uROM，用于存储与复杂指令相对应的微码程序段；以及

微码解码器，根据输入的复杂指令信息和处理器运行时刻状态信息，对从uROM中读取的已编码微码进行解码。

2、根据权利要求1所述的译码装置，其特征在于还包括：

暂停译码控制模块，用于检测能够改变处理器运行时刻状态信息的微码的译出或发射事件，暂停从uROM读取新微码，直到对所述能够改变处理器运行时刻状态信息的微码的操作完成为止。

3、根据权利要求1所述的译码装置，其特征在于所述已编码微码包括在操作码域、操作数域、操作数宽度域上经过了编码的微码。

4、根据权利要求3所述的译码装置，其特征在于所述输入的复杂指令信息和处理器运行状态信息包括用于解码在操作码域、操作数域、操作数宽度域上经过了编码的微码所需的操作信息。

5、一种包括根据权利要求1所述的对复杂指令译码生成微码的译码装置的通用中央处理器。

6、一种对复杂指令译码生成微码的译码方法，包括：

生成与输入的复杂指令相对应的、用于索引uROM中所存储的微码程序段的入口地址，并提供给uROM地址控制器；

uROM地址控制器根据生成的用于索引uROM中所存储的微码程序段的入口地址，将与输入的复杂指令相对应的微码程序段中的已编码微码从uROM中读取到微码解码器；

根据输入的复杂指令信息和处理器运行时刻状态信息，对从uROM

中读取的已编码微码进行解码。

7、根据权利要求6所述的译码方法，其特征在于还包括：

在检测到能够改变处理器运行时刻状态信息的微码的译出或发射事件时，暂停从uROM读取新微码，直到对所述能够改变处理器运行时刻状态信息的微码的操作完成为止。

8、根据权利要求6所述的译码方法，其特征在于所述已编码微码包括在操作码域、操作数域、操作数宽度域上经过了编码的微码。

9、根据权利要求8所述的译码方法，其特征在于所述输入的复杂指令信息和处理器运行状态信息包括用于解码在操作码域、操作数域、操作数宽度域上经过了编码的微码所需的操作信息。

对复杂指令译码生成微码的译码装置和方法

技术领域

本发明涉及通用处理器，尤其涉及一种对复杂指令译码生成微码的译码装置和方法，通过暂停/恢复译码机制确保了译码的准确运行，且减小了用于存储微码程序段的ROM的容量。

背景技术

超标量通用中央处理器采用多发射的流水线结构获得高性能，划分为取指、译码、执行、写回、提交等流水级。在译码阶段，使用精简指令集系统的处理器通常由处理器中硬件电路直接译码指令，而使用复杂指令集系统的处理器可能需要将1条复杂指令译码为处理器可执行的2条或多条简单的指令，或者称之为微码。例如，在x86指令系统中，由于一部分指令实现了较为复杂的功能，因此x86处理器中需要使用微码程序段完成指令的功能。当复杂指令对应的微码程序段较长时，这些微码通常存储于处理器的ROM块中。

在通用中央处理器中，指令以二进制流的方式表示。在一些中央处理器系统上，指令的长度固定，而在另外一些系统上，指令的长度不相同，如x86指令系统，指令的长度可为1字节到15字节，其中操作码长度为1字节到2字节，其余字节指定了指令的寻址模式、操作数等细节。x86指令通用的构成如图1所示，可以看到，x86指令可能含有1~4字节可选的前缀，1~2字节的操作码，1字节可选的寻址模式字节(Mod/RM)，1字节可选的寻址地址字节(SIB)，1~4字节可选的地址偏移字节和1~4字节可选的立即数操作数字节。

x86指令的操作数域区分了不同指令的基本操作，受到前缀字节的影响。例如前缀字节可以改变指令的操作数宽度和地址宽度属性，改变内存访问段，或者指示处理器进行循环操作等。寻址模式字节和

寻址地址字节(仅在32位模式下使用)指定了寄存器操作数或者访问存储器的寻址模式所需要的操作数。可选的地址偏移字节指定了访问存储器相对于基地址的偏移量。可选的立即数操作数字节作为指令的立即数操作数参与运算。

x86指令集中部分指令完成的功能非常复杂，需要中央处理器多个连续的操作才能完成，这类指令称为复杂指令，1条复杂指令对应的连续的操作称为该指令的微码程序段。例如POPA指令，将ESP寄存器的值(堆栈指针)指定的存储器中堆栈的值装载至x86寄存器中，并在每一次装载过程中改变ESP寄存器值，这样的功能在处理器中使用连续的Load操作和加法运算完成。不同的x86指令由于实现功能的不同使用不同的微码程序段，对应的微码程序段通常存储于ROM中，通过读取ROM中的对应的微码程序段完成译码工作。

含有复杂指令的指令集系统的中央处理器接收指令后，将指令区分为简单指令和复杂指令，简单指令由硬件电路直接译码，复杂指令生成ROM中对应微码程序段的入口地址，由ROM控制电路通过地址索引ROM单元得到微码程序段。

同一条复杂指令可能由于指令操作数、处理器状态等不同使用不同的微码或微码程序段，需要将不同的情况下的微码都预先存储于ROM中。ROM容量的增加导致微码读出的延迟增大，不利于越来越高速的处理器的实现。同时，ROM容量的增加也使处理器成本提高。因此，减小ROM容量是复杂指令系统处理器设计者所遇到的挑战。

发明内容

本发明的目的是提出一种对复杂指令译码生成微码的译码装置和方法，通过暂停/恢复译码机制确保了译码的准确运行，且减小了用于存储已编码的微码程序段的ROM的容量。

根据本发明，提出了一种对复杂指令译码生成微码的译码装置，包括：初步译码模块，用于生成与输入的复杂指令相对应的、用于索引uROM中所存储的微码程序段的入口地址，并提供给uROM地址控制器；uROM地址控制器，用于根据初步译码模块输入的用于索引uROM中所存

储的微码程序段的入口地址，将与输入的复杂指令相对应的微码程序段中的已编码微码从uROM中读取到微码解码器；uROM，用于存储与复杂指令相对应的微码程序段；以及微码解码器，根据输入的复杂指令信息和处理器运行时刻状态信息，对从uROM中读取的已编码微码进行解码。

优选地，所述译码装置还包括：暂停译码控制模块，用于检测能够改变处理器运行时刻状态信息的微码的译出或发射事件，暂停从uROM读取新微码，直到对所述能够改变处理器运行时刻状态信息的微码的操作完成为止。

优选地，所述已编码微码包括在操作码域、操作数域、操作数宽度域上经过了编码的微码。

优选地，所述输入的复杂指令信息和处理器运行状态信息包括用于解码在操作码域、操作数域、操作数宽度域上经过了编码的微码所需的操作信息。

根据本发明，还提出了一种包括前述对复杂指令译码生成微码的译码装置的通用中央处理器。

根据本发明，还提出了一种对复杂指令译码生成微码的译码方法，包括：生成与输入的复杂指令相对应的、用于索引uROM中所存储的微码程序段的入口地址，并提供给uROM地址控制器；uROM地址控制器根据生成的用于索引uROM中所存储的微码程序段的入口地址，将与输入的复杂指令相对应的微码程序段中的已编码微码从uROM中读取到微码解码器；根据输入的复杂指令信息和处理器运行时刻状态信息，对从uROM中读取的已编码微码进行解码。

优选地，所述译码方法还包括：在检测到能够改变处理器运行时刻状态信息的微码的译出或发射事件时，暂停从uROM读取新微码，直到对所述能够改变处理器运行时刻状态信息的微码的操作完成为止。

附图说明

通过参考以下结合附图对所采用的优选实施例的详细描述，本发明的上述目的、优点和特征将变得显而易见，其中：

图1是示出了长度可变的x86指令的构成的示意图；

图2是示出了一种通用中央处理器的具体实现的方框图，该中央处理器包括根据本发明实施例的译码装置；

图3是示出了包括在图2的中央处理器中的根据本发明实施例的译码装置的构成的方框图；以及

图4是示出了根据本发明实施例的复杂指令译码方法的流程图。

具体实施方式

下面将参考附图来描述本发明的优选实施例。

图2是示出了一种通用中央处理器的具体实现的方框图，该中央处理器包括根据本发明实施例的译码装置。

如图2所示，所述中央处理器包括预转移预测模块201、寄存器堆203、指令缓存205、译码器207、指令调度模块209、执行功能部件211、译码/预取模块213、总线接口模块215和数据缓存217。

在典型的处理器中，程序运行时代码和数据存储在存储器中，总线接口模块215负责完成处理器对存储器的数据的装载和保存的请求。预译码/预取模块213从总线接口模块215接收指令，将指令装载入指令缓存205中的同时，对指令进行预译码。转移预测模块201根据预译码后的指令预测下一个指令基本块的起始地址，指示预译码/预取模块213预取预测的指令基本块。指令缓存205中的指令进入译码器207进行译码，变换为执行部件可执行的内部指令格式。简单指令可能译码为单条对应的指令，复杂指令可能译码为多条指令。译码后的指令进入指令调度模块209，指令调度模块209在指令准备好操作数后，发射指令至各个不同的执行功能部件211完成对应的功能。执行功能部件211可能需要从寄存器堆203存取数据，也可能存取数据缓存217缓存的存储器数据。当存取的数据无法在数据缓存217中找到时，数据缓存217请求总线接口模块215从存储器中取回该数据。

下面将描述图3。图3是示出了包括在图2的中央处理器中的根据本发明实施例的译码装置（译码器207）的构成的方框图。

如图3所示，译码器207包括指令对齐模块301、初步译码模块303、

uROM地址控制器305、简单指令译码模块307、uROM（微程序只读存储器）309、暂停译码控制模块311、以及微码解码器313。

该译码器207接收含有指令的字节流，通过指令对齐模块301将不定长的指令划分边界，初步译码模块303将指令划分为简单指令和复杂指令，简单指令通过硬件组合电路（即，简单指令译码模块307）译码，复杂指令通过在uROM地址控制器305的控制下，从uROM中索引相应微码程序段完成译码工作。

初步译码模块303还为复杂指令生成提供用于索引uROM微码程序段的入口地址，送入uROM地址控制器305。uROM地址控制器305指示uROM中下一个时钟周期需要输出的微码在uROM中的入口地址。并根据该入口地址从uROM输出相应微码，编码的微码在微码解码器313根据译码的指令特点和当前处理器运行状态信息解码微码中的各个域，送至指令调度模块209。

由于可能影响微码解码器313的对编码的微码的解码结果，改变处理器运行控制状态的微码解码完成送出后，需要暂停译出微码的工作，并在处理器改变后恢复译出微码，这些工作是由暂停译码控制模块311来完成的。根据本发明的一个实施例，存储于uROM中的每条微码都包含1位的暂停位，修改处理器状态并影响微码解码器工作的微码该域置“1”，否则置“0”。

每个时钟周期译出的所有微码的暂停位送入暂停译码控制模块311。如果本时钟周期内输出包含暂停位为“1”的微码，则暂停译码控制模块311取消暂停位为“1”的微码后本时钟周期内所有输出的微码，并通知uROM地址控制器305将索引地址更改为暂停位为“1”的微码后一条微码在uROM中的入口地址，暂停位随着微码进入指令调度模块209。

此后，在暂停译码期间，暂停译码控制模块311控制uROM地址控制器305保持索引地址不变，并使uROM所有输出微码无效。仅当改变处理器运行控制状态的微码完成相应操作后，表明微码生成器接入的处理器运行控制状态信号已经改变完毕，指令调度模块209发送微码完成信号至暂停译码控制模块311，译码器207则根据暂停的uROM地址恢复

译出微码，暂停译码控制模块311继续监听输出微码的暂停位。暂停译码的机制用于处理器中控制寄存器的切换，例如浮点指令FRSTOR指令执行时将存储器中存储的浮点控制寄存器和浮点栈寄存器值导入处理器浮点控制器和浮点栈寄存器中，由于控制寄存器中浮点栈指针会影响浮点栈寄存器号到浮点逻辑寄存器号的映射，暂停译码机制保证了浮点栈指针修改后微码解码器能够根据浮点栈指针正确地完成浮点栈寄存器到浮点逻辑寄存器的映射。

图4是示出了根据本发明实施例的复杂指令译码方法的流程图。

同一条复杂指令可能由于指令操作数、处理器控制状态等不同使用不同的微码或微码程序段，需要将不同的情况下的微码都预先存储于uROM中。情况越多意味着uROM中需要预先存储的微码数量越多，然而这些微码程序段本身之间具备很大相似度。为了相同的微码程序段适用于不同的情况的同一条复杂指令译码甚至同一类复杂指令，存储于uROM中的所有微码的操作码域、操作数域、操作数宽度等域是经过编码的，表示不同解码方法。一个操作码的编码可能最后根据当前处理器控制运行状态解码为执行部件可识别的不同的操作码，一个操作数宽度域的编码可能根据译码指令的操作数宽度前缀解码为8位、16位、32位宽度等。例如，如下是一种根据本发明的操作数域8位的编码实施例(节选)。

操作数域编码	解码方式
0x01	将 x86 指令的 ModR/M 字节中的 RM 域指示的寄存器作为操作数
0x03	将 x86 指令操作码域中后 3 位指示的寄存器作为操作数
0x0e	将本条微码所包含的立即数域的值作为立即数操作数
0x0f	将本条微码所包含的立即数域的值(仍然是一个编码)指示的常数立即数作为立即数操作数，这样的常数立即数可能是本条 x86 指令的 EIP 值等。
0x11	将 EAX 寄存器作为操作数

0x20	将 1 个临时寄存器作为操作数
0x71	将 x86 浮点指令的 ModR/M 字节中 RM 域指示的浮点栈寄存器和当前浮点栈 TOP 指针运算后得到的逻辑寄存器作为操作数
0x81	将 CS 段选择控制寄存器或者影子寄存器作为操作数
0xa5	将 EFLAGS 控制寄存器作为操作数
0xc4	将 x86 指令的 ModR/M 字节中的 REG 域指示的 CR 寄存器作为操作数
.....

根据uROM索引地址uROM读电路输出编码的微码后，编码的微码进入微码解码器解码。如前所述，编码指定了微码解码器对微码域的解码方式。微码解码器根据编码从译码指令和当前处理器运行控制环境中收集信息，根据微码域的编码填写入微码各域中。例如微码中一个操作数的编码为0x01，微码解码器则会将x86指令的ModR/M字节中的RM域指示的寄存器作为操作数。又如微码的操作码域为0x3e，表示根据EFLAGS控制寄存器的串操作方向位DF位生成不同的加“1”或者减“1”操作。除此之外，微码使用的操作数宽度域、段寄存器域等都可以进行编码。

微码常用到立即数操作数，如移位操作需要立即数操作数指定移位的位数。当微码所使用的立即数的值固定不变时，可以直接将立即数的值写到uROM中微码的立即数域中。另外一些立即数操作数的值并不固定不变，例如POPA指令中连续的LOAD操作的偏移量，32位代码段下访问存储器的地址的间隔为4字节，16位代码段下访问存储器的地址的间隔为2字节。变化的立即数也由立即数域的编码实现，微码解码器能够根据处理器控制状态动态生成这些立即数。另外，需要读取控制寄存器值时，控制寄存器值也作为微码的立即数操作数参与运算，节省了很多读取控制寄存器或者将控制寄存器装载入功能部件的时钟周期数。例如发生例外时，进行代码段切换需要将发生例外的指令的EIP压栈，微码解码器直接将EIP值作为立即数操作数进行压栈操作，而不

用读取控制寄存器值后再进行压栈操作。在一个实施例中，这些直接作为立即数操作数写入微码的值还包括CPL寄存器值、下一条指令的EIP值、发生例外时的错误码ERROR-CODE寄存器值等。

如图4所示，在步骤401，将复杂指令对应的编码的微码程序段存储于ROM中，由于编码的通用性，通常一条复杂指令对应一个微码程序段。在步骤403，译码器207根据复杂指令的前缀、操作码等产生对应该复杂指令第一条微码的ROM地址。在步骤405，根据该ROM地址来索引微码程序段。在步骤407，译码器207中的微码解码器313根据复杂指令信息和处理器运行时刻状态信息将编码的微码解码，生成执行部件可运行的微码，送入指令调度模块209，其中所述复杂指令信息包括用于译码在操作码域、操作数域、操作数宽度、段寄存器域等上经过了编码的微码所需的操作信息。由于微码解码器对编码的微码的解码依赖于处理器运行时刻状态信息，因此当可能改变处理器运行时刻状态信息的微码译出或发射时，暂停译码控制模块311检测到这一事件，并暂停从ROM中索引新的微码和微码解码器的工作，指令调度部件209检测到这条可能改变处理器运行时刻状态信息的微码完成运算时，向暂停译码控制模块311送回恢复译码的信号，译码器207重新输出微码。由于暂停/恢复译码机制的保证了暂停/恢复译码时刻的前后的微码都能够看到正确的控制寄存器值，因此编码的微码可以指定将控制寄存器的值可以作为立即数操作数直接写入微码的操作数域中，参与运算。

尽管以上已经结合本发明的优选实施例示出了本发明，但是本领域的技术人员将会理解，在不脱离本发明的精神和范围的情况下，可以对本发明进行各种修改、替换和改变。因此，本发明不应由上述实施例来限定，而应由所附权利要求及其等价物来限定。

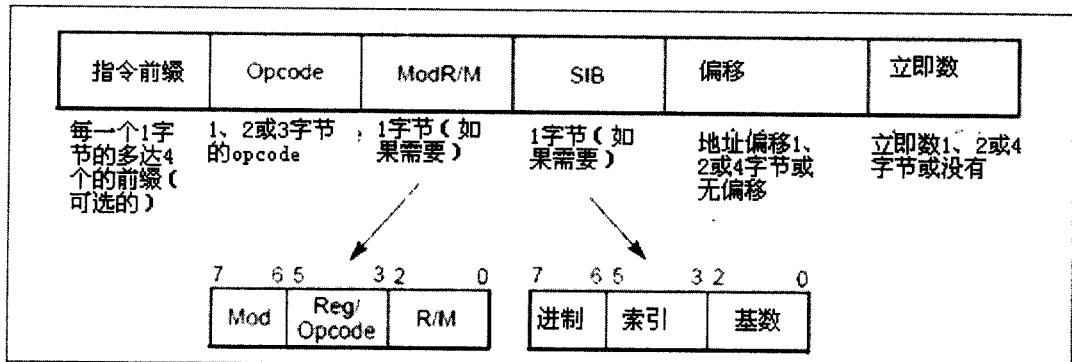


图 1

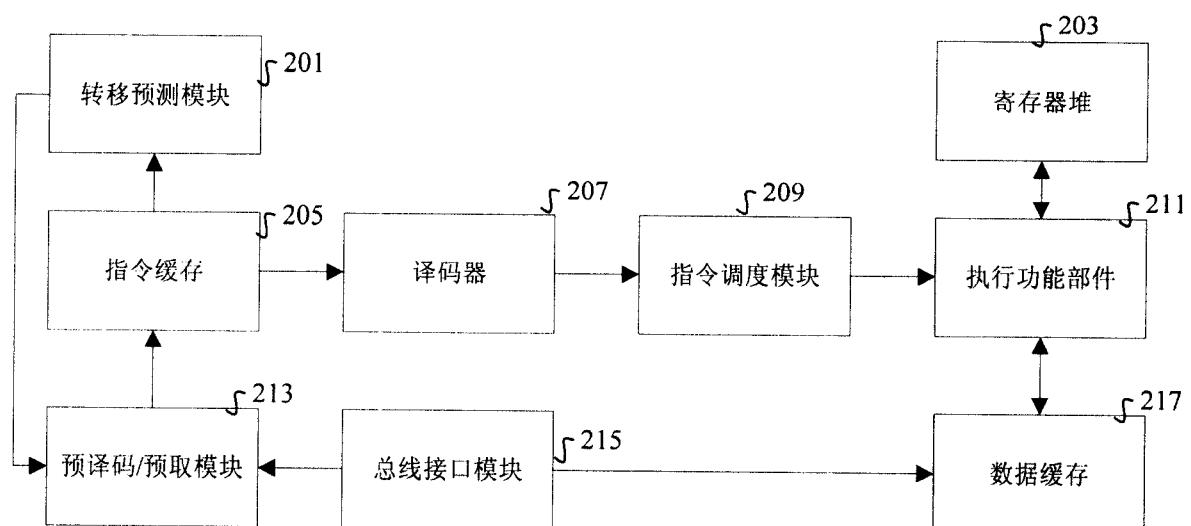


图 2

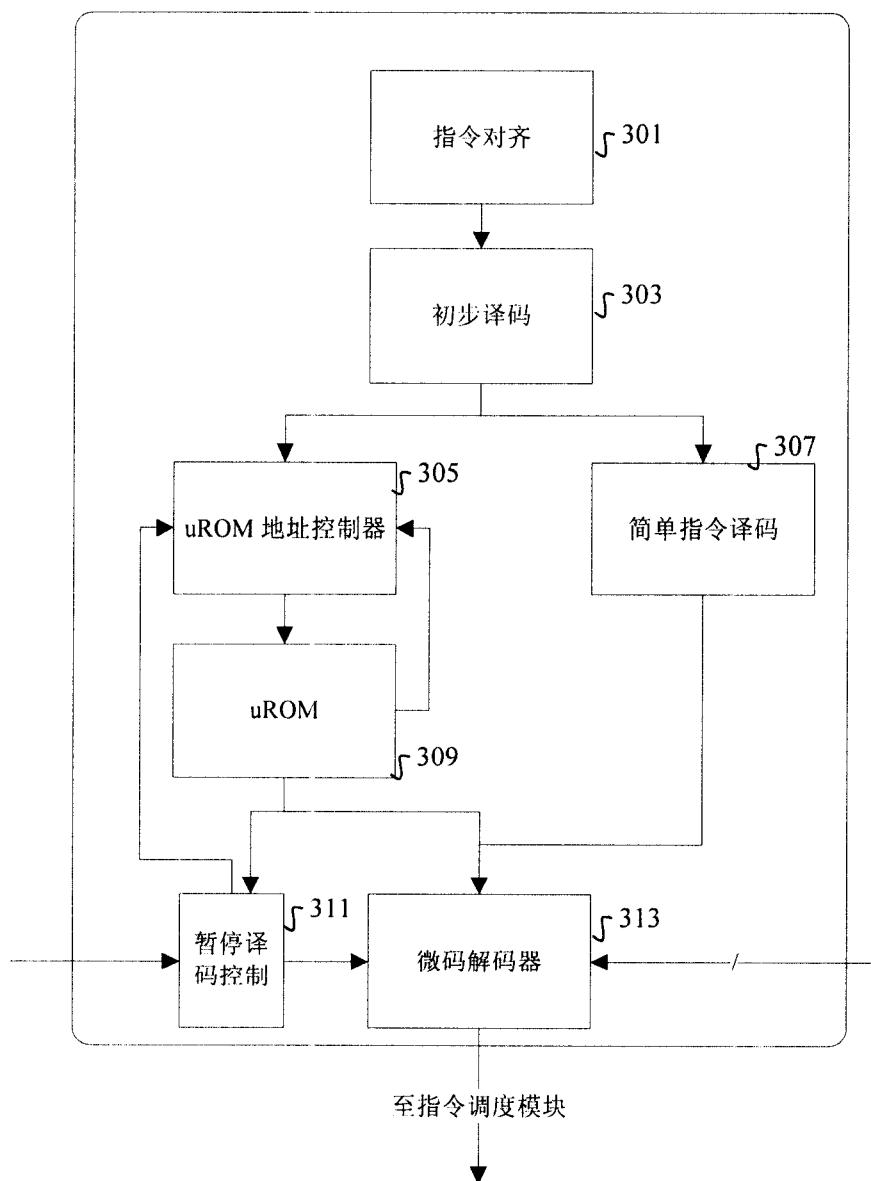


图 3

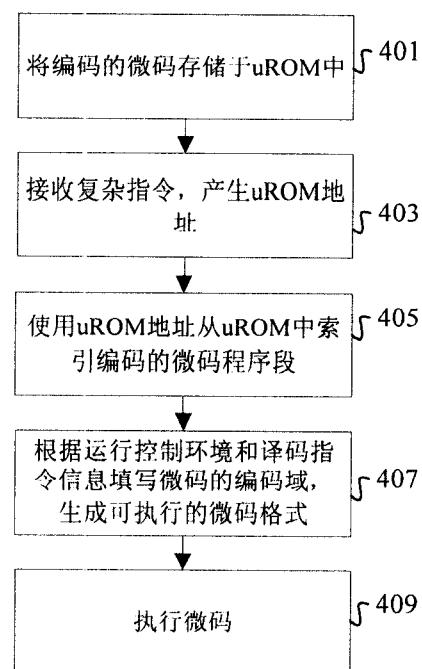


图 4